



Faculty of Computer Science and Information Technology

***EDUCALENDAR: ENHANCING DAILY PLANNING FOR STUDENTS WITH NLP-
DRIVEN EVENT AUTOMATION***

Rebecca Lai Yee Enn

Bachelor of Computer Science with Honours

(Computational Science)

2025

**EDUCALENDAR: ENHANCING DAILY PLANNING FOR STUDENTS WITH NLP-DRIVEN
EVENT AUTOMATION**

REBECCA LAI YEE ENN

This project is submitted in partial fulfilment of the
requirements for the degree of
Bachelor of Computer Science with Honours
(Computational Science)

Faculty of Computer Science and Information Technology

UNIVERSITI MALAYSIA SARAWAK

2025

DECLARATION

I hereby declare that this final year project is my original work. I have not copied from any other student's work or any other sources except where due to reference or acknowledgement is not made explicitly in the text, nor has any of the part had been written by another person.



.....
(Rebecca Lai Yee Enn)

24/07/2025

.....
(Date Submitted)

Faculty of Computer Science and Information Technology University
Malaysia Sarawak

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to all those who have supported and guided me throughout the completion of this project.

First and foremost, I would like to extend my deepest thanks to my supervisor, Ts. Dr Sarah Flora anak Samson for her invaluable guidance, support, and encouragement. Her expertise and constructive feedback have been instrumental in the success of this project.

I would also like to thank my examiner, Dr Mohammad bin Hossin, for his support and for providing additional insights and suggestions throughout the course of this project.

I would like to acknowledge the Faculty of Computer Science and Information Technology at University of Malaysia Sarawak for providing the necessary resources and facilities, which enabled me to carry out my project effectively.

I would also like to express my gratitude to my friends for their support, collaboration, and friendship during the course of this project. Their feedback, discussions, and camaraderie were invaluable in refining my ideas.

A heartfelt thanks to my family for their unwavering love, patience, and encouragement. Their constant belief in me gave me the motivation to overcome challenges and complete this project.

Finally, I would like to thank everyone who contributed to this project, whether directly or indirectly, for their help, advice, or support.

ABSTRACT

Many students rely on digital calendars for managing academic and personal schedules, yet most still require repetitive manual entry of event detail such as event title, date, time, location, frequency, which is repetitive and prone to human errors. To address this, EduCalendar is developed as a smart, web-based calendar that leverages Natural Language Processing (NLP), specifically Named Entity Recognition (NER), to automatically extract key event details such as event name, date, time, location, and frequency from natural language input. Key features include natural language event creation, recurring event handling, visual task displays, and multi-tab navigation. This project focuses on gathering user requirements, training the NER model, and evaluating usability through functional and User Acceptance Testing (UAT). Built with SvelteKit, Firebase, spaCy, and deployed via Vercel and Hugging Face Spaces, EduCalendar achieved 80% extraction accuracy and received high user satisfaction, validating its potential as an intelligent productivity tool for students.

ABSTRACT

Ramai pelajar bergantung pada kalendar digital untuk mengurus jadual akademik dan peribadi, namun masih perlu memasukkan maklumat acara seperti tajuk, tarikh, masa, lokasi, dan kekerapan secara manual, berulang, dan mudah terdedah kepada kesilapan. Untuk mengatasi isu ini, EduCalendar dibangunkan sebagai kalendar pintar berasaskan web yang menggunakan Pemprosesan Bahasa Semula Jadi (NLP), khususnya Pengecaman Entiti Bernama (NER), untuk mengekstrak maklumat acara seperti nama acara, tarikh, masa, lokasi, dan kekerapan secara automatik daripada input bahasa semula jadi. Ciri utama termasuk penciptaan acara menggunakan bahasa semula jadi, pengurusan acara berulang, paparan tugas visual, dan navigasi berbilang tab. Projek ini menumpukan kepada pengumpulan keperluan pengguna, latihan model NER, dan penilaian kebolegunaan melalui ujian fungsional dan Ujian Penerimaan Pengguna (UAT). Dibina dengan SvelteKit, Firebase, dan spaCy, serta dideploy melalui Vercel dan Hugging Face Spaces, EduCalendar mencapai ketepatan pengekstrakan 80% dan mendapat kepuasan pengguna yang tinggi, membuktikan potensinya sebagai alat produktiviti pintar untuk pelajar.

TABLE OF CONTENTS

DECLARATION.....	i
ACKNOWLEDGEMENT	ii
ABSTRACT.....	iii
ABSTRACK	iv
TABLE OF CONTENTS	v
LIST OF FIGURES	ix
LIST OF TABLES	xii
Chapter 1 : Introduction	1
1.1 Background	1
1.2 Problem Statement	2
1.3 Objectives	3
1.4 Methodology	3
1.5 Project Scope	4
1.6 Significance of Project	4
1.7 Project Schedule.....	5
1.8 Expected Outcome	6
1.9 Project Report Outline	6
1.9.1 Chapter 1: Introduction	6
1.9.2 Chapter 2: Literature Review	7
1.9.3 Chapter 3: Requirement, Analysis and Design	7
1.9.4 Chapter 4: Implementation.....	7
1.9.5 Chapter 5: Testing	7
1.9.6 Chapter 6: Conclusion and Future Work.....	7
1.10 Summary	8
Chapter 2 : Literature Review	9
2.1 Introduction.....	9
2.2 Review on Existing Systems.....	9
2.2.1 Google Calendar	9
2.2.2 SkedPal	13
2.2.3 Fantastical	17
2.2.4 Comparison between Existing Systems	22

2.3 Review on Tools and Technologies.....	23
2.3.1 Front-end Development	23
2.3.1.1 Figma Prototype tool.....	23
2.3.1.2 SvelteKit Framework	24
2.3.1.3 HTML, CSS, JavaScript Programming language	24
2.3.2 Back-end Development.....	25
2.3.2.1 Python Programming language.....	25
2.3.2.2 Firebase	25
2.3.3 NLP model	26
2.3.3.1 NLP Techniques	26
2.3.3.2 Application of NER in Event Extraction.....	30
2.3.3.3 Utilized NLP Library for NER.....	31
2.3.3.4 Evaluation Metrics for NER Model	32
2.3.4 Software Development Tools	33
2.3.4.1 Visual Studio Code.....	34
2.4 Proposed System	34
2.5 Summary	35
Chapter 3 : Requirement, Analysis and Design	36
3.1 Introduction.....	36
3.2 Methodology Summary	36
3.3 System Planning.....	39
3.4 Requirement Elicitation and Analysis	40
3.4.1 Requirement Elicitation	40
3.4.1.1 Digital Calendar Usage	41
3.4.1.2 User Experience and AI integration in Digital Calendars	44
3.4.1.3 UI Preferences and Evaluation.....	47
3.4.2 Requirement Analysis	50
3.4.2.1 Functional Requirement	51
3.4.2.2 Software Requirement.....	52
3.4.2.3 Hardware Requirement	52
3.5 System Design	53
3.5.1 Logical Design	53
3.5.1.1 Class Diagram	54

3.5.1.2 Data Flow Diagram.....	54
3.5.2 Wireframe	64
3.5.2.1 Sign up/Login page	65
3.5.2.2 Event creation pop up	67
3.5.2.3 Homepage page.....	68
3.5.2.4 Calendar page.....	69
3.5.2.5 Courses page	70
3.6 Summary	70
Chapter 4 : Implementation	72
4.1 Introduction.....	72
4.2 Installation and Configuration	72
4.2.1 Visual Studio Code.....	72
4.2.2 Firebase Console.....	73
4.3 NLP NER dataset preparation.....	74
4.4 Custom NER Model.....	77
4.4.1 Data annotation	77
4.4.2 Custom NER Model Training and Validation	77
4.4.3 Custom NER Model Testing	79
4.4.4 Issues with Baseline Custom NER Model and Hyperparameter Tunning Attempts	80
4.4.5 Custom NER Model and Post-Processing	85
4.5 Proposed System's Web-based System's Interface	86
4.5.1 Sign-up/Login page.....	86
4.5.2 Event/Course creation Popup.....	87
4.5.3 Homepage	90
4.5.4 Calendar Page	93
4.5.5 Courses Page.....	95
4.5.6 Settings Popup	97
4.5.7 About EduCalendar Popup.....	98
4.7 Summary	99
Chapter 5 : Testing	100
5.1 Introduction.....	100
5.2 Functional Testing.....	100
5.2.1 Sign-Up Functionality Testing	100

5.2.2 Login Functionality Testing	101
5.2.3 Course Creation Testing	102
5.2.4 Course Details Editing Testing.....	104
5.2.5 Event Creation Testing	105
5.2.6 Event Detail Viewing Testing	106
5.2.7 Event Details Editing Testing.....	107
5.2.8 Event Sorting/Filtering Testing	108
5.2.9 Event Deletion Testing	109
5.2.10 Course Deletion Testing.....	109
5.2.11 Logout Functionality Testing	110
5.2.12 Forgot Password Functionality Testing.....	110
5.2.13 Change Password Functionality Testing	111
5.3 User Acceptance Test (UAT).....	111
5.3.1 Website Accessibility	112
5.3.2 (Task 1) Course Creation	114
5.3.3 (Task 2) Event Creation	114
5.3.4 System Acceptance	120
5.4 Summary	123
Chapter 6 : Conclusion and Future Work.....	124
6.1 Introduction.....	124
6.2 Achievement	124
6.3 Limitations and Constraints	125
6.4 Future Works.....	125
6.5 Conclusion	126
References.....	127
Appendix.....	130
Appendix A: Gantt Chart	130
Appendix B: Google Form Questionnaire	131
Appendix C: Google Form UAT Questionnaire.....	142
Appendix D: Full results of the UAT Questionnaire's (Task 2) Event Creation section.....	151

LIST OF FIGURES

Figure 1.1. Prototype Model (geeksforgeeks, 2025b).....	4
Figure 1.2. Gantt chart for FYP 1.	5
Figure 1.3 Gantt chart for FYP 2.	6
Figure 2.1. Google Calendar homepage.....	10
Figure 2.2. Google Calendar’s week view.	11
Figure 2.3. Google Calendar’s month view.	11
Figure 2.4. Google Calendar’s schedule view.....	12
Figure 2.5. Google Tasks in Google Calendar.	13
Figure 2.6. The homepage of SkedPal.	14
Figure 2.7. SkedPal’s list view.	15
Figure 2.8. SkedPal’s kanban board view.	15
Figure 2.9. SkedPal’s calendar’s week view.	15
Figure 2.10. SkedPal’s calendar’s month view.	16
Figure 2.11. UI of SkedPal during user text input for event creation.	16
Figure 2.12. The homepage of Fantastical.	18
Figure 2.13. Fantastical’s all tasks section.	19
Figure 2.14. Fantastical’s week view.	19
Figure 2.15. Fantastical’s quarter view.	20
Figure 2.16. UI of Fantastical during user text input for event creation.	21
Figure 2.17. Event extraction classification (Liu et al., 2021).	27
Figure 2.18. Representation of NER (Kaur et al., 2024).	28
Figure 2.19. Training spaCy model. (Nair et al., 2020).	30
Figure 2.20. System architecture pipeline of Shreyas et al. (2018)’s proposed system.	31
Figure 3.1. Prototyping Model.	36
Figure 3.2. Have you ever regularly used a digital calendar (e.g., Google Calendar, Apple Calendar, Outlook Calendar) to keep track of your academic responsibilities (assignments, projects, quizzes) and daily schedules?	41
Figure 3.3. If your answer in (4) is no, why didn't you use a digital calendar regularly?	42
Figure 3.4. If your answer in (4) is yes, what is the name of the digital calendar application?	43
Figure 3.5. If your answer in (4) is yes, how often did you use the digital calendar?	43
Figure 3.6. How important do you think it is to be able to keep track of your academic responsibilities (assignments, projects, quizzes) and daily schedules?	44
Figure 3.7. Do you find manually entering calendar events on digital calendars tedious or time-consuming?	45
Figure 3.8. If there is such an app that would be able to create events given some inputs in the digital calendar, do you think it will be beneficial to you?	45
Figure 3.9. Have you heard of AI integration in digital calendars?	46
Figure 3.10. In what way do you think AI integration in digital calendars would help you as a student?	46
Figure 3.11. The images shown in the questionnaire as visual reference for responders to answer question 13 and 14.....	47

Figure 3.12. Between the header section of Figure 1 and 2 (navigation between the home, calendar and course page are hidden within the menu icon), which type of header do you prefer?	47
Figure 3.13. Between the body section the homepage of Figure 1 and 2, which layout do you prefer?	48
Figure 3.14. The images shown in the questionnaire as visual reference for respondents to answer question 15.	48
Figure 3.15. Is the "Create Event" interface on both mobile (Figure 3 & 4) and desktop screen (Figure 5 & 6) visually clear and easy to understand?.....	49
Figure 3.16. The images shown in the questionnaire as visual reference for respondents to answer question 16.	49
Figure 3.17. Are the details displayed in an organized and easy-to-read format?	50
Figure 3.18. Class diagram of EduCalendar.	54
Figure 3.19. Context diagram of EduCalendar.	57
Figure 3.20. Level 0 DFD of EduCalendar.	58
Figure 3.21. Sign-up process.	59
Figure 3.22. Login process.....	59
Figure 3.23. Process input with NLP and create new event process process.	60
Figure 3.24. Manage event update and deletion process.	61
Figure 3.25. Create new course process.....	61
Figure 3.26. Create new performance item process.	62
Figure 3.27. Manage course/performance update and deletion process.	62
Figure 3.28. Process homepage/course/calendar view request process.	63
Figure 3.29. Sign up page desktop view.	65
Figure 3.30. Sign up page mobile view.....	65
Figure 3.31. Login page desktop view	65
Figure 3.32. Login page mobile view.	65
Figure 3.33. Create new event pop up desktop view.....	67
Figure 3.34. Create new event overlay mobile view.	67
Figure 3.35. UI of event creation pop up when it receives user input desktop view.....	67
Figure 3.36. UI of event creation overlay when it receives user input mobile view.	67
Figure 3.37. Homepage desktop view.	68
Figure 3.38. Homepage mobile view	68
Figure 3.39. Calendar page desktop view.	69
Figure 3.40. Calendar page mobile view.....	69
Figure 3.41. Courses page desktop view.....	70
Figure 3.42. Courses page mobile view.	70
Figure 4.1 Screenshot of VS Code for the proposed system's development.	73
Figure 4.2 Screenshot of the Cloud Firestore user interface	74
Figure 4.3 Template used in the event generation code.	75
Figure 4.4 Word cloud of the dataset.	76
Figure 4.5 Entity distribution in the dataset.	76
Figure 4.6 Model training and validation command.....	77
Figure 4.7 The output folder's contents.	78
Figure 4.8 The output of the training and validation command.....	78

Figure 4.9 Model testing and evaluation command.	79
Figure 4.10 The output of the testing and evaluation command.	79
Figure 4.11 Extra testing data that is manually created.	80
Figure 4.12 Sign-up page.	86
Figure 4.13 Login page.	87
Figure 4.14 Add new event tab section of the event/course creation popup.	87
Figure 4.15 UI of event creation pop up when it receives user input popup.	88
Figure 4.16 UI of event creation pop up when user click on the edit button.	88
Figure 4.17 Add new course tab section of the event/course creation popup.	89
Figure 4.18 Homepage on desktop (left) and on mobile (right).	90
Figure 4.19 Event details popup on desktop (left) and on mobile (right).	91
Figure 4.20 Edit event details popup on desktop (left) and on mobile (right).	91
Figure 4.21 Special events indicator on the calendar section, normal event(left), class time event (middle), recurring event (right).	92
Figure 4.22 Calendar page in default week view on desktop (left) and on mobile (right).	93
Figure 4.23 Calendar page in month view on desktop (left) and on mobile (right).	94
Figure 4.24 Calendar page in day view on desktop (left) and on mobile (right).	94
Figure 4.25 Calendar page in list view on desktop (left) and on mobile (right).	95
Figure 4.26 Courses page on desktop (left) and on mobile (right).	95
Figure 4.27 Edit course details popup on desktop (left) and on mobile (right).	96
Figure 4.28 Settings popup on desktop (left) and on mobile (right).	97
Figure 4.29 Change password UI.	98
Figure 4.30 About EduCalendar popup on desktop (left) and on mobile (right).	98
Figure 5.1 The device distribution used by the respondents.	112
Figure 5.2 EduCalendar’s website accessibility distribution of the respondents.	113
Figure 5.3 EduCalendar’s account creation success distribution of the respondents.	113
Figure 5.4 EduCalendar’s course creation success distribution of the respondents.	114
Figure 5.5 EduCalendar’s event extraction success distribution of the respondents.	115
Figure 5.6 The list of event prompts that EduCalendar failed to extract correctly.	115
Figure 5.7 EduCalendar’s overall system acceptance.	120
Figure 5.8 Other bugs in EduCalendar encountered by respondents.	121
Figure 5.9 Suggestions provided by the respondents that can improve EduCalendar.	122

LIST OF TABLES

Table 2.1. Comparison table of features between the reviewed existing digital calendars.	22
Table 2.2. The advantages and disadvantages of the reviewed existing digital calendars.	23
Table 2.3. Comparison between EduCalendar and the existing systems.	35
Table 3.1. Summary of the Prototype model methodology.....	39
Table 3.2. Features and their details of EduCalendar.....	51
Table 3.3. Software requirement of EduCalendar.	52
Table 3.4. Hardware requirement of EduCalendar.....	52
Table 3.5. The data store dictionary of EduCalendar.	56
Table 4.1 Extra testing on baseline model's performance with manually created testing data.	81
Table 4.2 Results of hyperparameter tuning.	85
Table 5.1 Valid User Sign-Up testing.....	101
Table 5.2 Invalid User Sign-Up testing.....	101
Table 5.3 Valid User Login testing.....	102
Table 5.4 Invalid User Login testing.....	102
Table 5.5 Successful course creation testing.....	103
Table 5.6 Unsuccessful course creation with missing course name testing.	104
Table 5.7 Unsuccessful course creation with missing course color testing.	104
Table 5.8 Course details edit testing.	105
Table 5.9 Successful course details edit testing.	106
Table 5.10 Unsuccessful course details edit testing.	106
Table 5.11 Event detail view testing.	107
Table 5.12 Event details edit testing.	107
Table 5.13 Event sorting testing.....	108
Table 5.14 Event filtered by date range testing.....	108
Table 5.15 Event filtered by event types testing.	109
Table 5.16 Event deletion testing.....	109
Table 5.17 Course deletion testing.....	109
Table 5.18 User logout testing.	110
Table 5.19 Forget user password testing.....	110
Table 5.20 Change user password testing.	111
Table 5.21 Analysis of the event prompts that are extracted incorrectly.....	119
Table 6.1 Summary of the project's objectives and achievements.....	125

Chapter 1 : Introduction

1.1 Background

In today's fast-paced world, managing time effectively is crucial for balancing professional responsibilities and personal commitments. Daily planners, whether digital or physical, serve as essential tools for organizing tasks, appointments, and events. With the technological advancements, most planners are now digital, for the convenience and reliability. However, manually inputting and managing these entries can be both time-consuming and prone to human errors, which creates friction between the app and the user, thus increasing the user's reluctance in managing their events and projects digitally.

Fortunately, recent advancements in Natural Language Processing (NLP) offer promising solutions to enhance the usability of daily planners. NLP, a subfield of artificial intelligence, enables computers to understand, interpret, and generate human language in a way that is both meaningful and useful (Stryker & Holdsworth, 2024). In NLP, Named Entity Recognition (NER) is one of its components where it identifies entities such as time, location, and event names in text (Hyder & Nafees, 2023). With NER, it will be able to identify the specific entities relevant to calendar events creation existing in the user's input. By integrating NLP's NER technique into daily planners, we can streamline the process of creating and managing events, making these tools more intuitive and efficient, as the users are only required to type on sentence to create a calendar event, as opposed to filling multiple fields and sieving through the options.

This project, EduCalendar, a web-based smart calendar, explores the integration of NLP's NER techniques into daily planners for students to automate the creation of events from natural language

inputs. For instance, users can input phrases like "Tea with friends every Wednesday at 2:30 PM for one and a half hours at Cafe Red Brick," and the system will automatically extract and structure the necessary details—such as the event name, time, duration, and location—into the planner. This not only simplifies the event creation process but also minimizes the risk of errors associated with manual entry. Besides, the user experience of using a daily planner is also greatly enhanced.

1.2 Problem Statement

Numerous individuals rely on digital calendaring tools such as Google Calendar, Outlook Calendar, and Apple Calendar for their daily scheduling and planning activities. However, most digital planners still require manual input of event details, which is cumbersome as students are already juggling between assignment deadlines, test and quizzes, and personal commitments (Ashana et al., 2021). To bridge this gap, there is a need for a system that leverages NLP's NER technique to interpret user input in everyday language automatically, allowing the calendar to automatically populate with relevant event information by recognizing entity categories like event name, time range, location, and frequency of the event in the sentences. This would enable students to focus on completing their tasks rather than spending time on manual data entry. By making calendars smarter through NLP, we can significantly enhance the user experience and streamline the scheduling process.

1.3 Objectives

This project aims to design and develop a web-based smart calendar that enables users to automate event creation through the application of NLP techniques. To achieve this goal, the following three objectives are outlined:

1. To review existing systems that are relevant and curate requirements for EduCalendar, a web-based smart calendar's design and functionalities.
2. To design and develop EduCalendar with the incorporation of NLP for students to plan their projects, assignments, tests, quizzes, and personal commitments.
3. To evaluate the EduCalendar's usability through functional testing and user acceptance testing.

1.4 Methodology

The methodology of the EduCalendar website will be based on Prototyping Model (PM). In this methodology, a proof of concept (PoC) is developed early in the process. This prototype serves to demonstrate the feasibility of an idea and to validate whether the proposed solution meets the desired requirements (Dhiman, 2024).

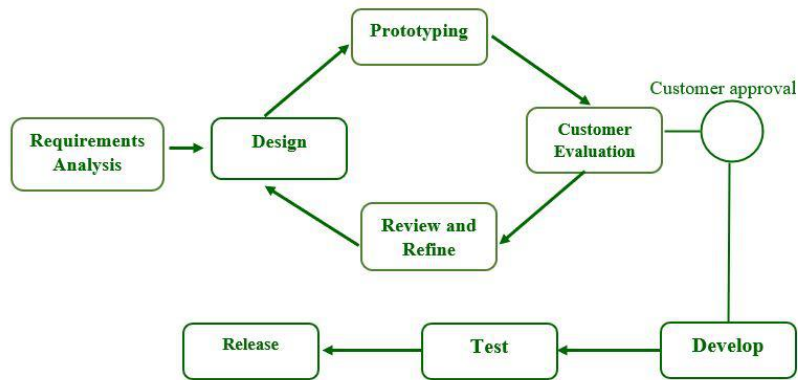


Figure 1.1. Prototype Model (geeksforgeeks, 2025b)

Figure 1.1. shows the process cycle of the prototyping development and iterative development of PM. There are eight phases in PM, namely Requirement Analysis, Design, Prototyping, Customer Evaluation, Review and Refine, Develop, Test, and Release. Additional elaboration of the methodology will be discussed in Chapter Three.

1.5 Project Scope

This project targets students who are balancing homework, schoolwork, quizzes, and tests. The solution is a web-based smart calendar designed to help manage their busy schedules. The entities recognized in the user's natural language input for calendar event creation would be limited to event name, date range, time range, location, and frequency only. Other entities would not be taken into consideration for the scope of user input for this system.

1.6 Significance of Project

By integrating NLP's Named Entity Recognition technique into daily planners, this project aims to enhance user experience and efficiency of students when using a daily planner. It addresses the need for a more intuitive way to manage time and commitments, thus providing users with a

powerful tool to organize their schedules more effectively. This integration has the potential to significantly reduce the time spent on managing daily tasks so the user can focus their energy on completing the actual tasks instead.

1.7 Project Schedule

This project utilizes Gantt chart to manage and track its schedule, tasks, and progress to ensure that the project is completed on time. The schedule is divided into two phases. The first phase is Final Year Project 1 (FYP 1), and the second phase is Final Year Project 2 (FYP 2). FYP 1 starts on 7th October 2024 and ends on 12th January 2024, as shown in Figure 1.2 below.

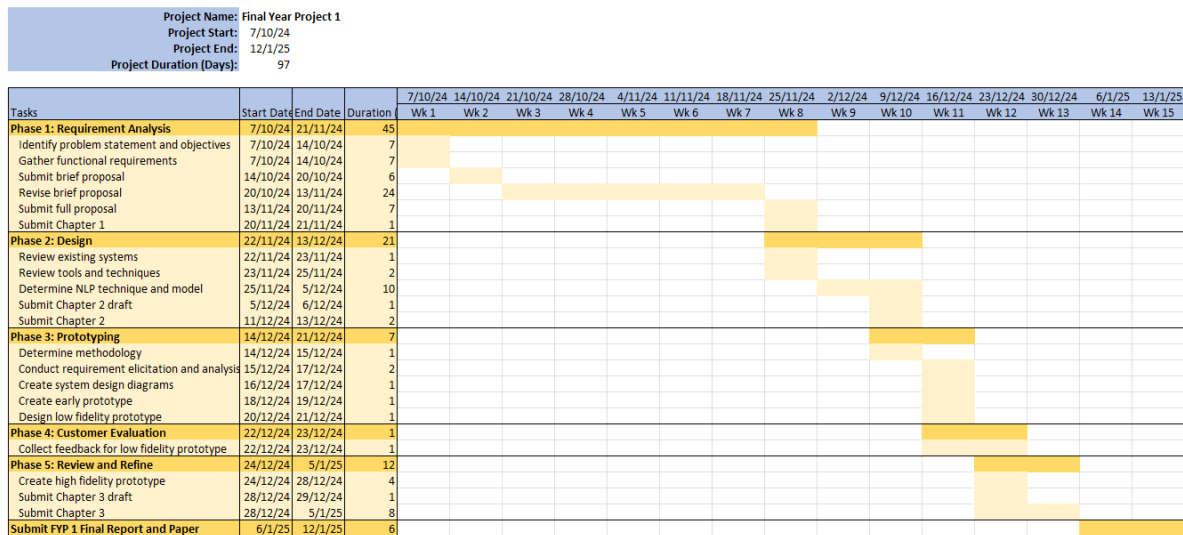


Figure 1.2. Gantt chart for FYP 1.

The second phase, FYP 2, starts on 17th March 2025 and ends on 28th July 2025, as shown in Figure 1.3 below.

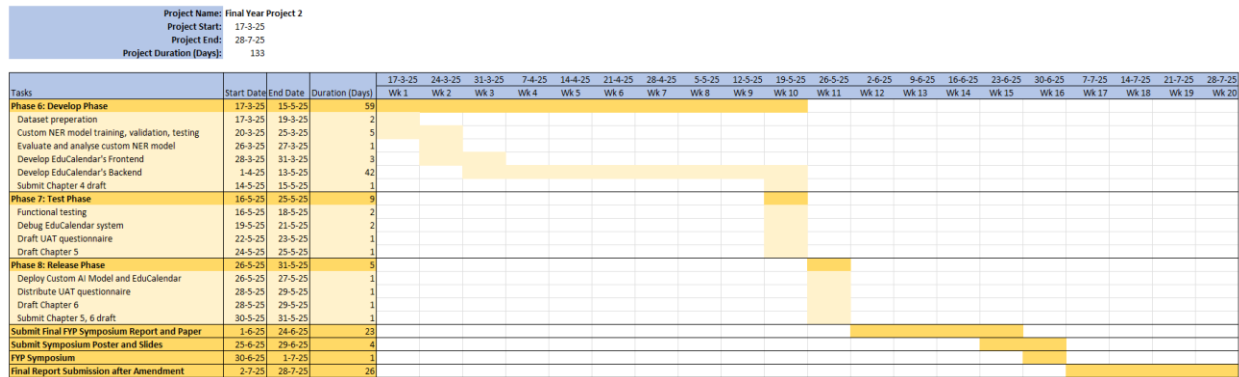


Figure 1.3 Gantt chart for FYP 2.

1.8 Expected Outcome

With EduCalendar, students will be able to use natural language input to accurately create calendar events on the system with minimal difficulties. Once created, these events can be easily viewed, edited, and deleted as needed.

1.9 Project Report Outline

This thesis will contain:

1.9.1 Chapter 1: Introduction

This chapter introduces the project by outlining its background, problem statement, objectives, methodology, scope, significance, schedule, and the project report's outline. It provides background information on the problem being addressed, the motivation behind the project, and the objectives that guide the research and development process.

1.9.2 Chapter 2: Literature Review

This chapter reviews existing systems relevant to the proposed system, summarizing the existing system's features, pros, and cons in the digital calendar ecosystem. It also determines the tools and technologies used to develop the proposed system.

1.9.3 Chapter 3: Requirement, Analysis and Design

This chapter details the functional and non-functional requirements of the system. It summarizes the methodology introduced in Chapter 1 and detailing the specific procedures and technologies involved in each phase of the model. Requirements of the proposed system is analyzed and collected using questionnaires. Besides, this chapter also make system design diagrams of the proposed system, which is divided into logical design and physical design.

1.9.4 Chapter 4: Implementation

This chapter describes the technical aspects of the project's implementation. It includes an overview of the development process, tools and technologies used, and the coding practices followed. Specific features or components of the system are discussed in detail.

1.9.5 Chapter 5: Testing

This chapter covers the testing strategies employed to ensure the system meets its requirements and functions correctly. It includes descriptions of testing methods like functional testing and user acceptance testing are conducted, and the results of the tests are documented.

1.9.6 Chapter 6: Conclusion and Future Work

This chapter summarizes the key findings and contributions of the proposed system, reflecting on its overall success in achieving its objectives, and the system's limitations and constraints. It

also discusses potential areas for future work, including enhancements or extensions of the proposed system.

1.10 Summary

In summary, Chapter One introduces the proposed system, EduCalendar, a smart digital calendar that allows students to automate event creation using NLP. The aim of this proposed system is to create a digital calendar that is tailored for students with the ability to create events using just natural language input.

Chapter 2 : Literature Review

2.1 Introduction

This chapter explores three existing systems with similar functionalities and capabilities to EduCalendar, a digital calendar with automated event creation. The features, pros, and cons of the existing systems are reviewed and subsequently summarized in a comparison table for a comprehensive high-level view. Thus, the best aspects will be selected to be included in the proposed system.

Besides that, the appropriate tools and technologies for web-based front-end and back-end development will be analyzed and chosen for the development of EduCalendar.

2.2 Review on Existing Systems

This sub-chapter looks into the existing systems that are relevant to the proposed system and deliver a quick summary of their main selling points, User Interface (UI), User Experience (UX), NLP integration, target audience, and other features. The systems featured here consists of web-based systems, ‘Google Calendar’ and ‘SkedPal’, and a software system, ‘Fantastical’. These digital calendars have the common features of adding, editing, viewing, and deleting events.

2.2.1 Google Calendar

Google calendar is one of the most widely used digital calendars as it provides the essentials of a basic digital calendar for free. This system is available in both mobile and web application, but this review will be focusing on its web-based counterpart.

The UI of Google Calendar is simple and clean, straight to the point. Figure 2.1 shows the default homepage features a navigation bar on the top, a side bar of Google Calendar's integration with other applications. The main functions of the calendar are contained in the middle of the page. It consists of a weekly view of the user's events, the monthly calendar of the current month, and the list of the user's created calendars.

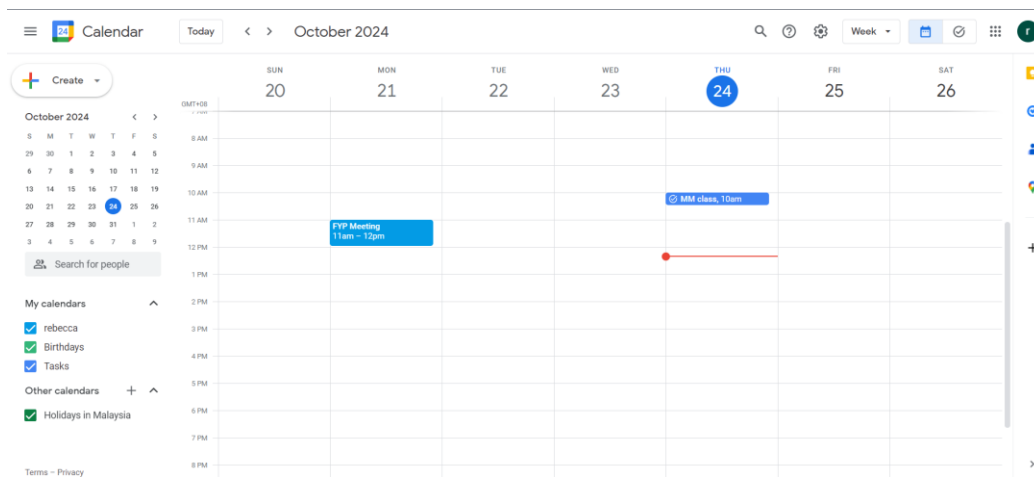


Figure 2.1. Google Calendar homepage.

For Google calendar's events or task view, it offers three types of layout patterns, which are timeline (Figure 2.2), grid (Figure 2.3) and list (Figure 2.4). The timeline layout is employed for both the day, week, and four days views, while the grid layout is utilized in the month and quarter views. Additionally, the list layout is featured in the schedule view.

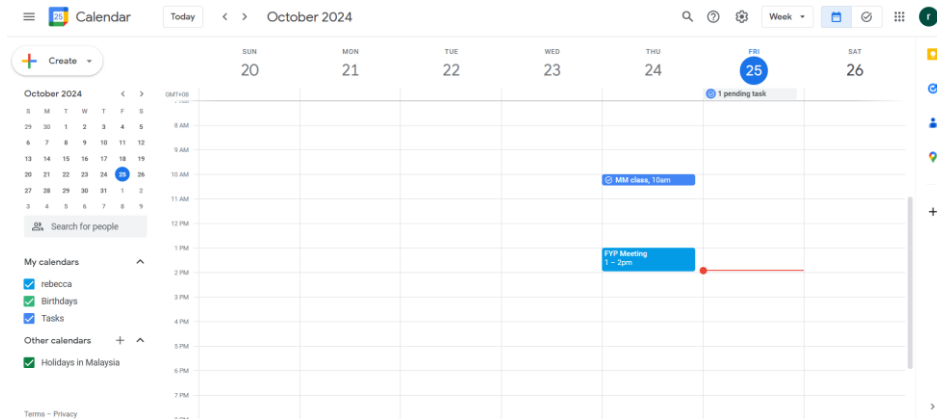


Figure 2.2. Google Calendar's week view.

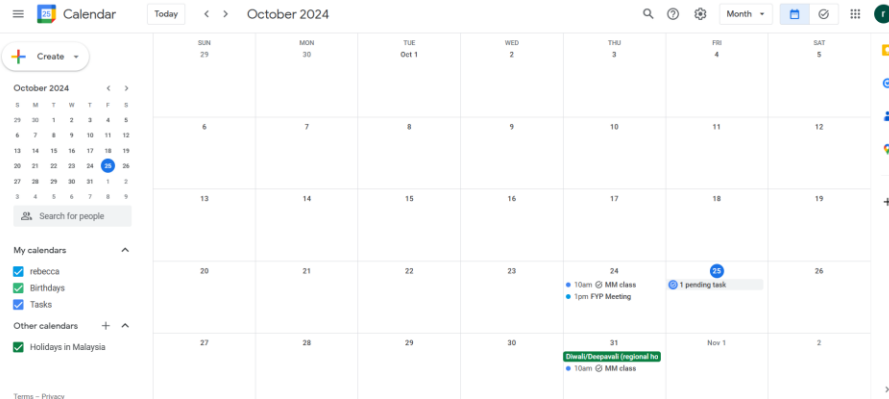


Figure 2.3. Google Calendar's month view.

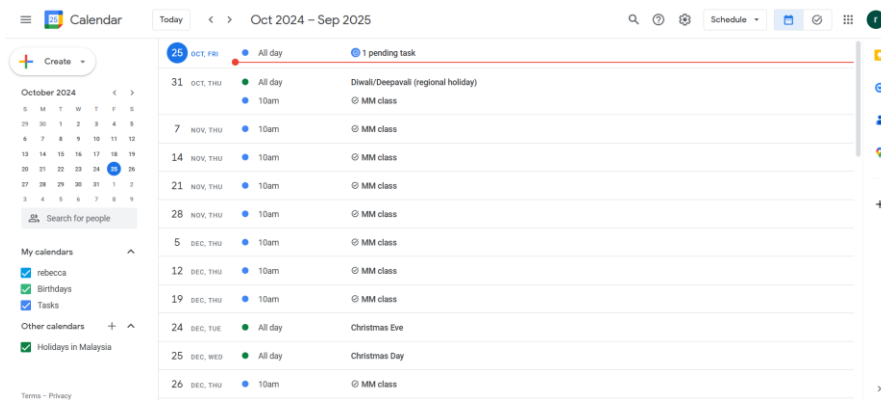


Figure 2.4. Google Calendar's schedule view.

As for the user experience of Google Calendar, the system's learning curve is minimal, as it follows established design patterns of a calendar, which is already familiar to most users. The main features are focused on the middle of the page. Navigation to a different view (from weekly view to monthly view) is contained in a drop-down box, which is easily accessible on the system's top navigation bar.

Unfortunately, Google Calendar does not provide NLP features. Should the user wish to integrate that feature, they will have to install third-party extensions.

Google Calendar's target audience is the general public. For business purposes, Google Calendar offered paid add-on services with business specific capabilities.

Other than the basic features of digital calendars, Google Calendar offers a lot of integration within both its own Google ecosystem and third-party systems. The integrations with Google Keep, Google Tasks (Figure 2.5), Google Contacts, Google Maps, and other third-party add-ons can be accessed on the right sidebar. Google Calendar’s integration with Google Tasks is the most notable one as it has another separate section on the navigation bar. The tasks recorded here will also show up on the timeline view in Google Calendar.

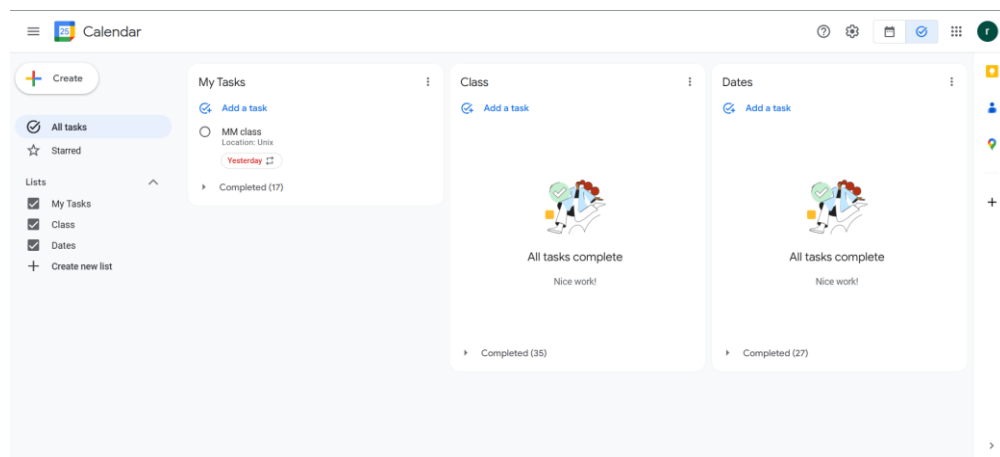


Figure 2.5. Google Tasks in Google Calendar.

In addition to integration, users can differentiate between their events—such as work and personal events—by creating separate calendars. This allows for color-coded events as a visual cue, and users can easily toggle their visibility by checking or unchecking the calendars in the list at the bottom left corner.

2.2.2 SkedPal

SkedPal is an AI-powered scheduling platform that utilizes time blocking to optimize the user’s daily time planning. It intelligently analyzes your tasks, deadlines, priorities, and preferences to create personalized schedules that adapt to the user’s workflow. Unfortunately,

SkedPal is a monthly subscription system that offers a 14-day trial for users to test out their product and decide if they want to commit into paying for the system (SkedPal, n.d.).

The UI of SkedPal is simple and clean, straight to the point. The default homepage, as shown in Figure 2.6, features a main navigation bar on its left side, which can be hidden and unhidden. The main functions of the calendar are focused on the right side, where its contents are centered for enhanced readability.

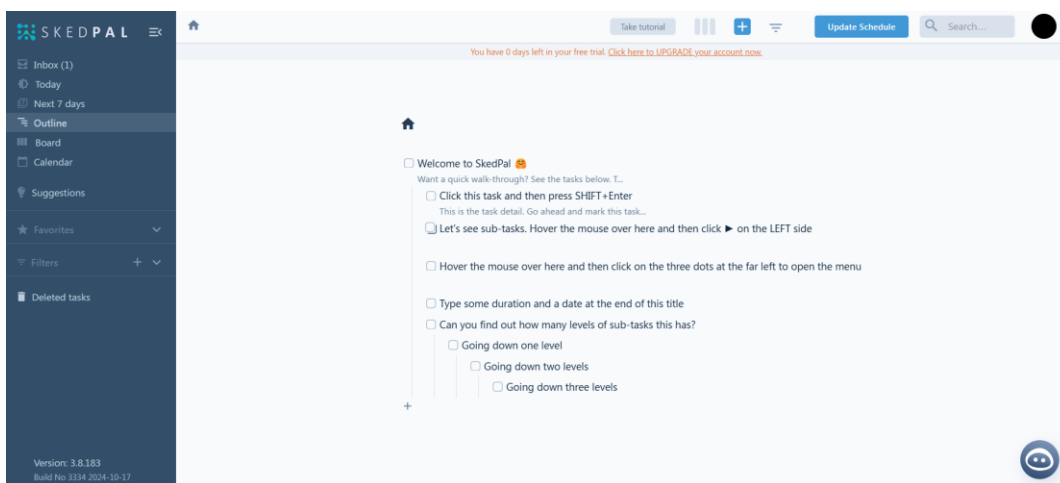


Figure 2.6. The homepage of SkedPal.

In SkedPal, the calendar's events or task view can be displayed in three different layouts, which are list (Figure 2.7), kanban board (Figure 2.8), timeline (Figure 2.9), and grid view. The list layout is utilized in displaying inbox, today, next seven days, and outline. The kanban board is employed in the board view, and the timeline and grid view are used when displaying the calendar's day, week, and month view.

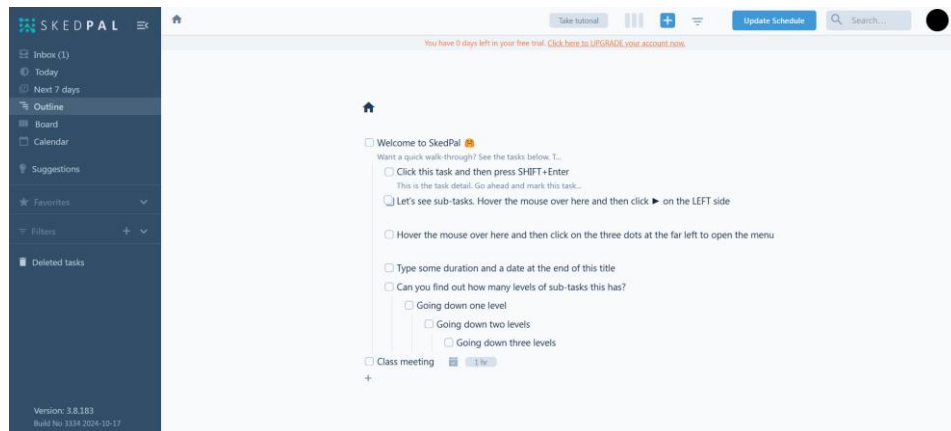


Figure 2.7. SkedPal's list view.

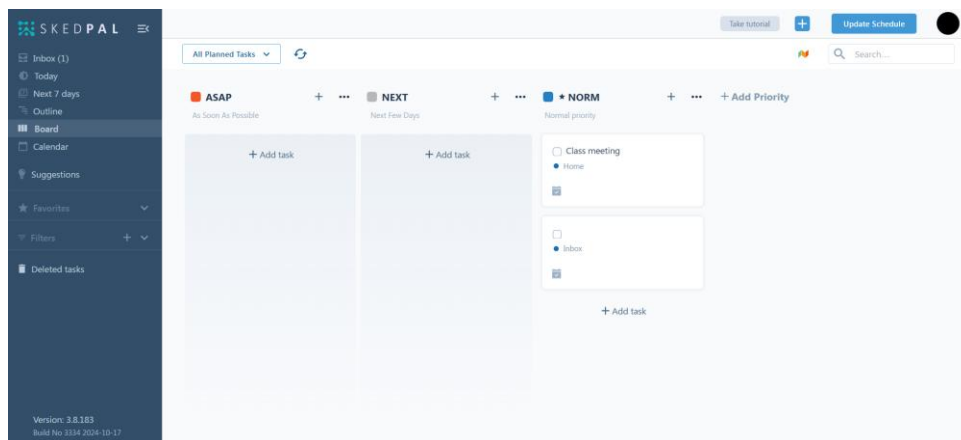


Figure 2.8. SkedPal's kanban board view.

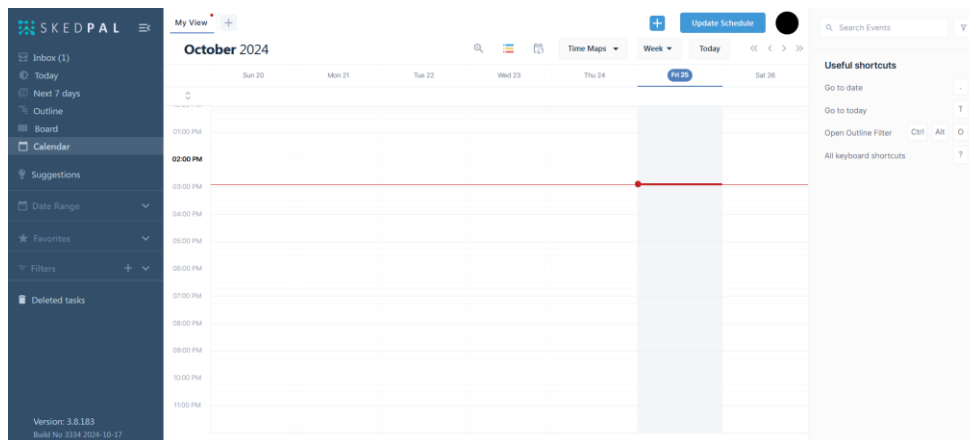


Figure 2.9. SkedPal's calendar's week view.

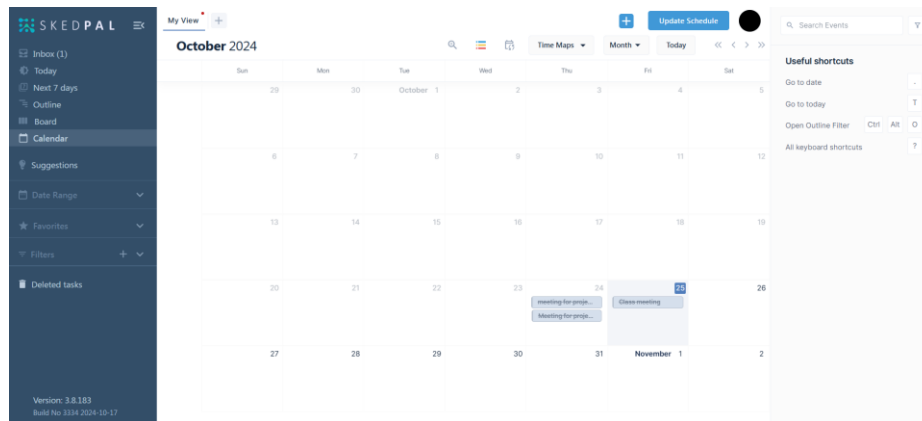


Figure 2.10. SkedPal's calendar's month view.

SkedPal's user experience requires a steeper learning curve, as it provides more advanced features like time blocking, automated task prioritization, and AI scheduling. Even though they provide a step-by-step tutorial for users to get acclimatized to the planner, the time and effort required might create a lot of frustration for the user, as they are already busy with their own tasks. The information organization is also a little confusing due to its multiple choices in layout options on the main navigation bar.

The NLP feature of calendar event generation is provided by SkedPal and it is prompt-based, as shown in Figure 2.11. The detected keywords or entities are highlighted as a feedback mechanism to communicate to the user on which terms are utilized, so the user can make changes accordingly. The key demographic of SkedPal is the general public.

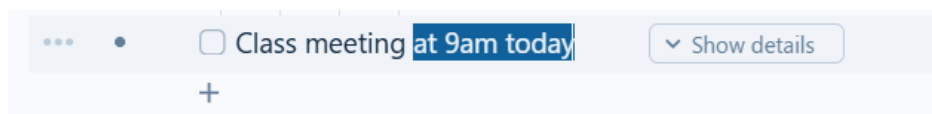


Figure 2.11. UI of SkedPal during user text input for event creation.

SkedPal offers features like time blocking, automated task prioritization, and AI scheduling suggestions. The time blocking feature allows users to schedule tasks effectively. Automated task prioritization occurs mainly in the board section, where each task's priority is determined by its

relative position on the board and within its column. SkedPal's AI-powered scheduling automatically arranges tasks based on the user's availability, eliminating the need for users to manually specify time slots for each task. Additionally, SkedPal enables users to group tasks together, so that all tasks within the same group inherit the group's personalized parameters, such as timing and tags.

2.2.3 Fantastical

Fantastical is a freemium app that primarily serves as an interface for users to access and manage their existing digital calendar accounts, such as Google Calendar, Outlook Calendar, or Apple Calendar. It consolidates all of the users' events from different calendar sources and the to-do list from the native IOS tasks app. Despite not providing a web-based system like the previous two systems, Fantastical is chosen as one of this project's review candidates due to its NLP text to event creation feature. This review will focus on their tablet application.

The UI of Fantastical is simple and straight to the point. It is however more cluttered compared to Google Calendar. Figure 2.12 shows the default homepage of Fantastical, which consists of two columns, the overview section on the left, and the visualization of the tasks and events on the right. On the overview section, there is a mini calendar of the current month on top, and a list of upcoming events under it. Next to the overview section displays the events and tasks in terms of daily, weekly, monthly, quarterly, or yearly views. There is also a section for a complete list of to-do tasks.

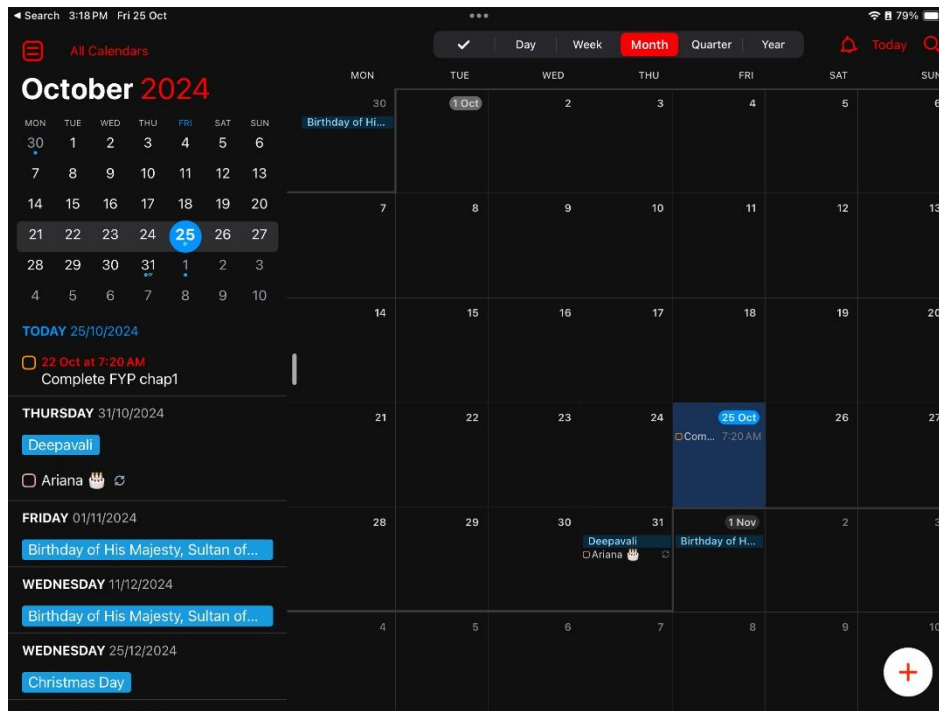


Figure 2.12. The homepage of Fantastical.

For Fantastical's events or task view it comes in three different display formats, which are list (Figure 2.13), timeline (Figure 2.14), and grid (Figure 2.13). The list layout is used in the upcoming events and the to-do list, timeline layout in the daily and weekly view, and finally, grid layout in monthly and quarterly view.

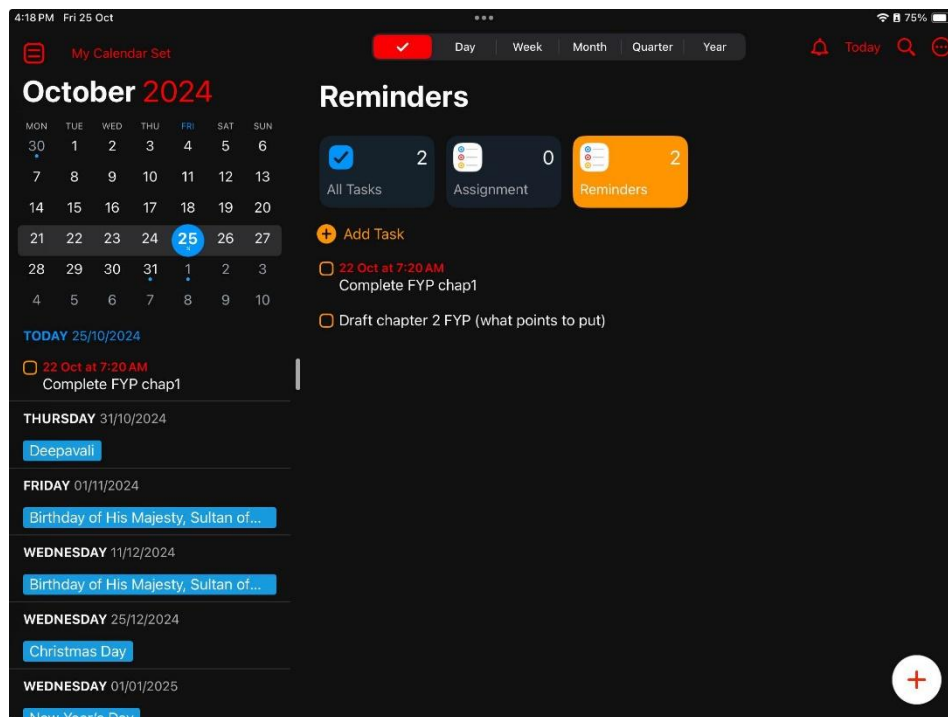


Figure 2.13. Fantastical's all tasks section.

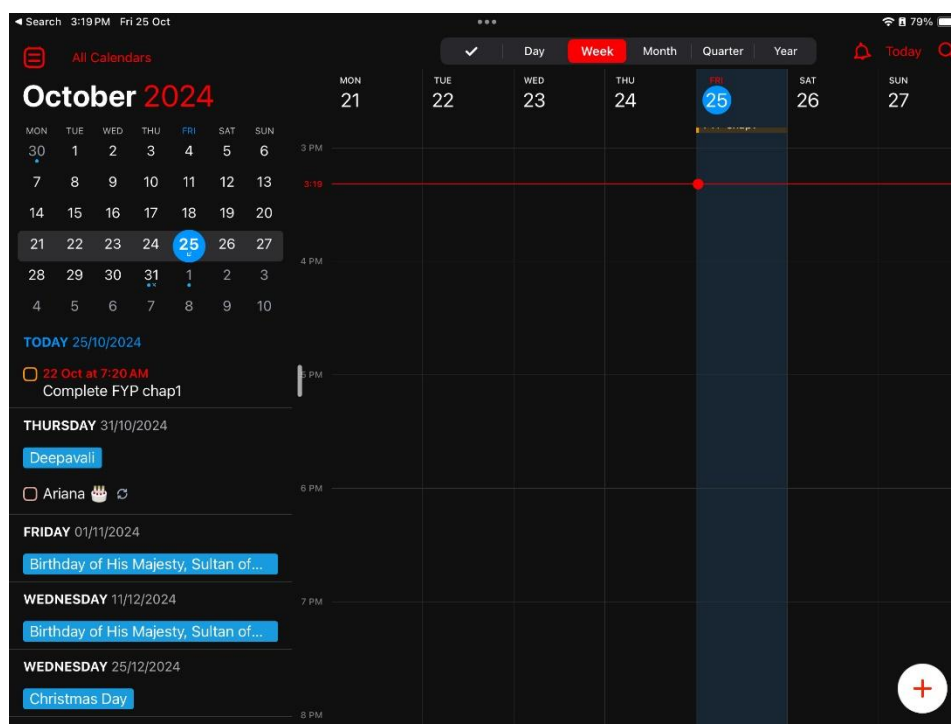


Figure 2.14. Fantastical's week view.

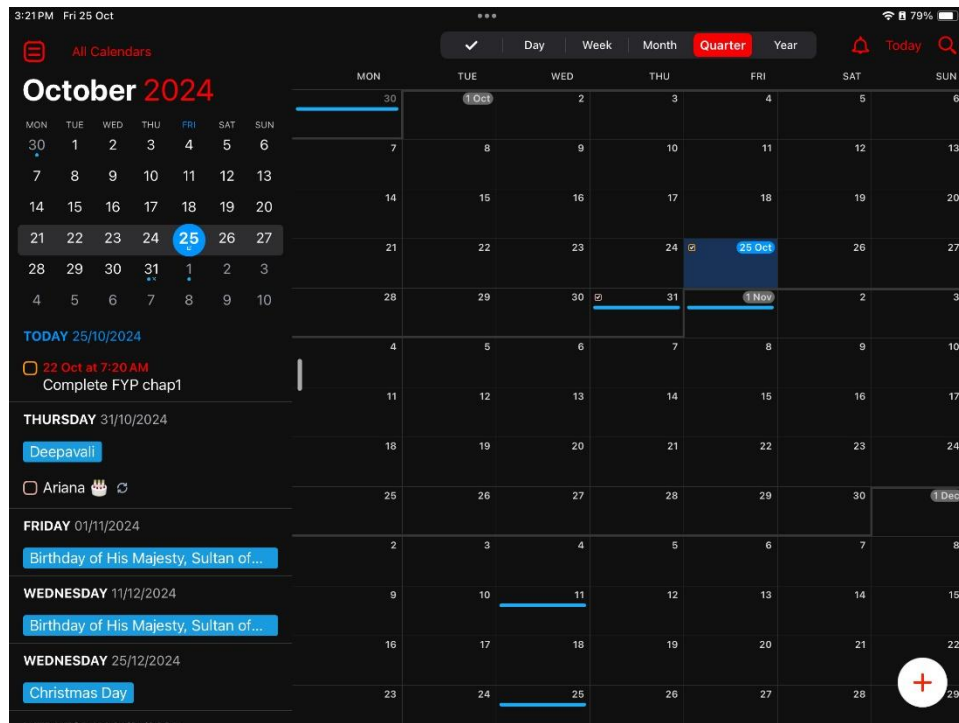


Figure 2.15. Fantastical's quarter view.

The system's learning curve is minimal, as it follows established design patterns of a calendar, which is already familiar to most users. The overview section and detailed view section's size can be customized by dragging the grey slider in the middle of the sections. The overview section can be completely hidden this way, should the user want to have a bigger view of the detailed view section.

Similar to SkedPal, the NLP calendar event creation is also provided by Fantastical and is also prompt-based (Figure 2.16). As the user is typing, the system will visualize the input text as a temporary event on the calendar. This visual feedback allows user to spot for potential double-booking or amend input errors.

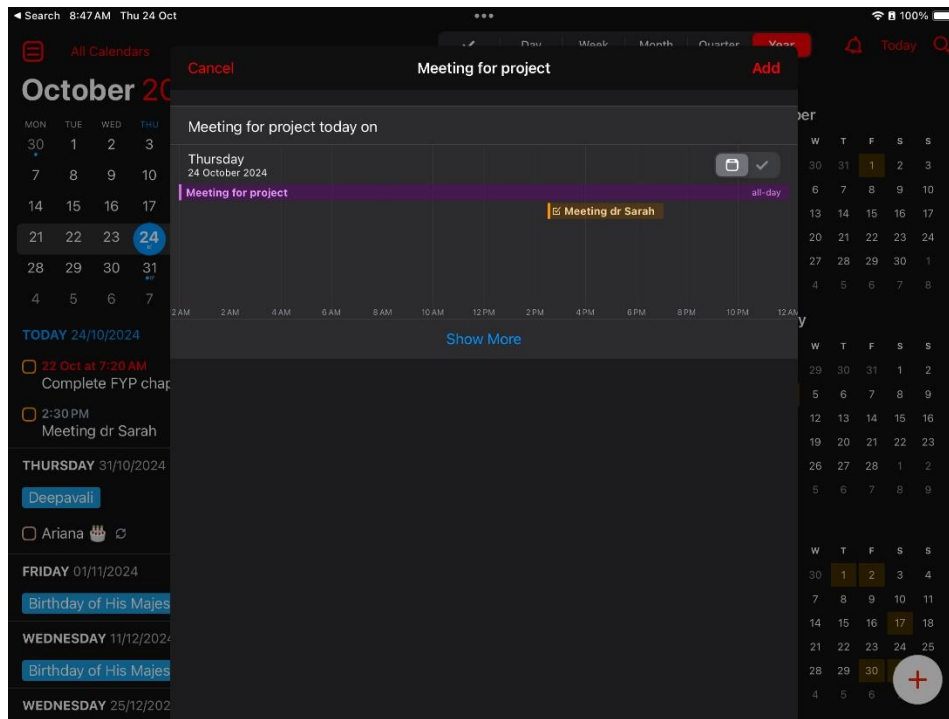


Figure 2.16. UI of Fantastical during user text input for event creation.

Fantastical is ideal for individuals who want to consolidate their calendar in one place, create events, set reminders, and easily share their schedule with others for personal use.

Aside from the basic digital calendar features, Fantastical enables users to create task and calendar templates for frequently used event formats, which streamlines the scheduling process. This feature is particularly helpful for recurring events, allowing users to save time and maintain consistency. Additionally, Fantastical offers the option to categorize calendar activities using calendar sets (a premium feature), which allows users to easily switch between different sets of events. This functionality not only reduces visual clutter but also enhances organization, making it simpler to focus on specific aspects of their schedules, whether for work, personal activities, or special projects.

2.2.4 Comparison between Existing Systems

Existing System Features	Google Calendar	SkedPal	Fantastical
UX	Minimal learning curve.	Steep learning curve, requires more effort to understand and fully utilize its features.	Minimal learning curve.
Event/Tasks view	Timeline, Grid, List	List, Kanban board, Grid	Timeline, Grid, List
NLP Event Creation Integration	X	✓	✓
To-do list integration	✓	✓	✓
Tagging system	X	✓	X

Table 2.1. Comparison table of features between the reviewed existing digital calendars.

Table 2.1 presents a comparison table based on features, which are: NLP Integration, event/tasks view, to-do list integration, and tagging system.

Although Google Calendar and Fantastical do not include a tagging system, they effectively differentiate events by allowing users to create separate calendars. This approach serves a functionally similar purpose to tagging, enabling users to categorize their events—such as work, personal, or social activities—while providing a clear visual distinction. By using different calendars, users can easily manage their schedules and focus on specific types of events as needed.

System	Advantages	Disadvantages
Google Calendar	- Free service for users who only require its basic features. - Clean and uncluttered interface.	- Does not offer in-house NLP integrations.
SkedPal	- Offers NLP integration for event creation. - Offers automated task scheduling.	- Requires steep learning curve. - Requires monthly subscription.

Fantastical	- Free service for users who only require its basic features. - Offers NLP integration for event creation.	- Only available on IOS platforms
-------------	---	-----------------------------------

Table 2.2. The advantages and disadvantages of the reviewed existing digital calendars.

Table 2.2 evaluates the pros and cons of the reviewed existing systems, in the context of its capabilities of being a digital calendar for students.

2.3 Review on Tools and Technologies

This sub-chapter explores the relevant tools and technologies for the proposed system, EduCalendar.

2.3.1 Front-end Development

Front-end development is the part of web development that focuses on the visual and interactive aspects of a website or application. It encompasses the visual and interactive elements that users engage with, shaping how users perceive and navigate a digital product. The tools that will be used for front-end development are Figma, for early prototyping, SvelteKit framework and HTML, CSS, JavaScript programming language for development.

2.3.1.1 Figma Prototype tool

Figma started off as a web-based design tool used primarily for UI/UX design. It's public release dates back to September 2016 (Figma, n.d.), around a year after its invite exclusive preview program (Josh, 2015). It allows designers to create, prototype, and collaborate on interface designs in real time. Figma's capabilities include creating interactive prototypes, enabling the user to simulate user flows and test designs before development. It supports the creation and maintenance of design systems, ensuring consistency across projects with reusable components and styles. With

automatic version control, Figma saves versions of the user's files, making it easy to track changes and revert to previous iterations. Additionally, its cloud-based nature ensures cross-platform compatibility, allowing team members to access and collaborate on projects from any operating system.

2.3.1.2 SvelteKit Framework

Svelte free and open-source software that is released under the MIT license (Svelte, n.d.-a). It is a modern JavaScript framework used for building web applications. Unlike other popular frameworks like React, Vue, or Angular, Svelte takes a different approach to web development by moving much of the work traditionally done in the browser to compile time (Svelte, n.d.-a). is the official framework for building full-featured web applications with **Svelte**, extending it with features like routing, server-side rendering, and API endpoints using a simple file-based structure (Svelte, n.d.-b). While Svelte handles the component logic and reactivity, SvelteKit provides the tools to turn those components into complete, production-ready apps with less setup and better performance.

2.3.1.3 HTML, CSS, JavaScript Programming language

Hyper Text Markup Language (HTML), Cascading Style Sheets (CSS), and JavaScript are the most basic programming languages used when doing web development. HTML is the standard language used to create the structure and backbone of web pages. It defines elements like headings, paragraphs, images, links, and other content. CSS is used to control the presentation, formatting, and layout of the HTML elements, such as colors, fonts, spacing, and positioning to make the web page more pleasing to the human eyes. JavaScript (JS) is a programming language that adds interactivity and dynamic behavior to web pages, allowing for actions like responding to user

clicks, validating forms, or updating content without reloading the page. Together, HTML provides the structure, CSS styles it, and JavaScript adds functionality, enabling the creation of fully interactive and visually appealing web applications (Kingsley, 2021).

2.3.2 Back-end Development

Back-end development is the part of web development that focuses on the server-side part of a website or application. It focuses on the web page's logic, database interactions, and server configuration. The tools that will be used for back-end development Python programming language and Firebase's Cloud Firestore service for the database.

2.3.2.1 Python Programming language

Python is a high-level, interpreted programming language known for its simplicity, readability, and versatility created by Guido van Rossum, released in 1991 (w3schools, n.d.). It is popular due to its simple syntax and community support that actively provides resources, forums, and open-source libraries. Python is used to develop EduCalendar due to it providing open-source libraries, such as spaCy and NLTK, that is used to handle NLP operations.

2.3.2.2 Firebase

Firebase is a platform developed by Google that helps developers to build, manage, and scale applications more easily by providing ready-to-use backend services. It supports multiple platforms, including Android, iOS, web, and Unity, and reduces the need for server-side programming by offering tools like authentication, hosting, messaging, and real-time databases (geeksforgeeks, 2025a). One of its key services is Cloud Firestore, a scalable NoSQL document database that allows developers to store, sync, and query data efficiently across users and devices (geeksforgeeks, 2025a). Firestore organizes data into collections and documents, supports real-

time updates, and is designed to work seamlessly on a global scale, making it ideal for modern app development.

2.3.3 NLP model

The two existing systems, SkedPal and Fantastical, which have NLP integration for event automation, did not clearly specify the NLP techniques used to develop these features. Hence, efforts were made to identify suitable NLP techniques for implementing this feature in the proposed system. The motivation behind these efforts stems from the need to ensure accurate and efficient automation of tasks, based on the user inputs. The selection of NLP techniques is focused on their ability to handle short text efficiently and effectively.

2.3.3.1 NLP Techniques

NLP is the intersect of AI and linguistics, where it aids machines to understand human written text, expressions, and statements (Kulkarni et al., 2023). According to Kulkarni (2023), by utilizing the power of NLP in machines and digital interfaces, it can drastically improve user experience and work efficiency. In NLP, event extraction it's a subfield (Liu et al., 2021), in order to carry out event extraction from natural language text, there are several NLP techniques that can be utilized.

Event extraction is another NLP technique aims to identify and extract event-related information from unstructured text (Singh, 2018). This involves detecting events described in natural language, and be defined as a specific occurrence that involves actions or activities, often associated with specific entities (people, organizations, locations, etc.), times, and other contextual information (Xiang & Wang, 2019). For example, a simple event might include a meeting, a conference, or a birthday.

As shown in Figure 2.17, here are two types of event extraction, closed-domain event extraction and open-domain event extraction (Liu et al., 2021; Xiang & Wang, 2019).

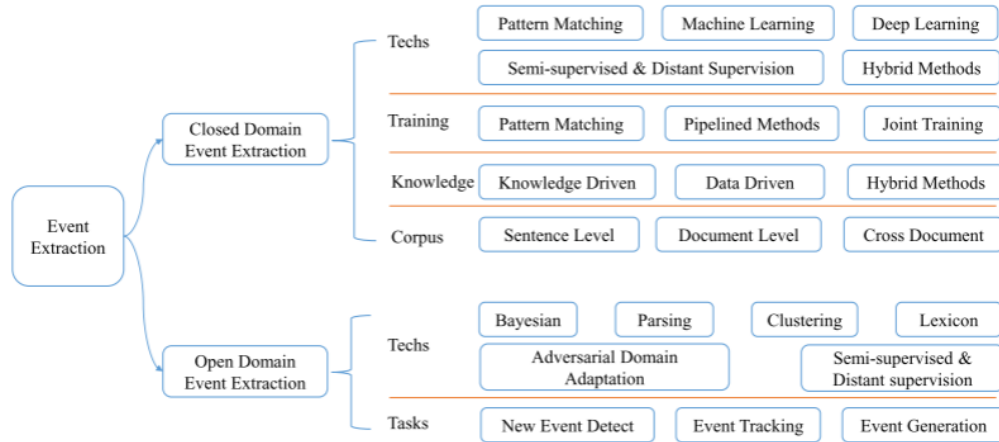


Figure 2.17. Event extraction classification (Liu et al., 2021).

Closed-domain extraction refers to identifying and extracting entities or information from a predefined, limited set of categories or topics. The system is trained on a specific domain with fixed categories, and it operates under the assumption that the data it encounters will fall within those predefined boundaries (Liu et al., 2021; Xiang & Wang, 2019). Open-domain extraction is more general and does not limit itself to specific, predefined categories. It seeks to identify new or unexpected entities or information from a wide array of possible topics and categories (Liu et al., 2021; Xiang & Wang, 2019).

In the context of building EduCalendar’s event creation automation feature, closed-domain event extraction is more suitable. This is due to the existence of predetermined entities (event title, event start time, event end time, etc.) required to create a calendar event. Further detecting new entities would be redundant and unnecessary as it might overwhelm the user with information overloading.

Named Entity Recognition (NER) is the task of identifying and classifying named entities (NEs) in text into predefined categories (Azar et al., 2021; Lai, 2022; Ushio & Camacho-Collados, 2022). These entities typically refer to real-world objects such as names of people, organizations, locations, dates, etc. The operation of NER systems that highlight and group the named entities in a sentence. “Droupadi Murmu”, a person (PER) is highlighted in red, “New Delhi”, the location (LOC) is highlighted in blue, the organization (ORG) is “Bharat Heavy Electricals Limited” which is highlighted in purple, and other named entity (O) is highlighted in green. Numerous kinds of proper nouns, common names, and other linguistic constructions known as named entities can be recognized and arranged using this method (Kaur et al., 2024). As NER extracts keywords or entities from the input statement and determines the meaning of those entities, sentences, and their relationships, it is crucial to the semantic portion of NLP (Shelar et al., 2020). Thus, any important fields in information management, including semantic annotation, question answering, ontology population, and opinion mining, are based on NER (Singh, 2018).



Figure 2.18. Representation of NER (Kaur et al., 2024).

NER approaches can be generally categorized into three types: hybrid, rule-based, and machine learning (IBM, n.d.). According to IBM (n.d.), rule-based approaches create grammatical rules to identify entities based on structural and grammatical features, however it is not time-efficient and may not generalize well to new data. Machine learning approaches train models on

labeled datasets with algorithms like conditional random fields and maximum entropy, using techniques spanning from traditional methods (decision trees and support vector machines) to advanced deep learning techniques (Recurrent Neural Networks (RNNs) and transformers). This approach generalizes better to new data, but require a large amount of labeled training data and computationally expensive. Hybrid approaches combines both rule-based and machine learning methods together to leverage their strengths, by using grammatical rules for common entities and machine learning for those which are more complex.

Named Entity Linking (NEL) goes a step beyond Named Entity Recognition. Once NER identifies an entity in text, NEL associates that entity with a unique reference in a knowledge base or an external entity catalog (Lai, 2022; Singh, 2018). References to entities in natural language text are often ambiguous, as a single entity may have multiple mentions, and a single mention may refer to several different entities. For an example, the word ‘apple’ can refer to the tech company or the fruit. Therefore, Entity Linking is the process of mapping named entity mentions to corresponding entries in a structured Knowledge Base (Singh, 2018).

Part-of-Speech (POS) tagging is the task of identifying and labeling each word in a sentence with its corresponding grammatical category (Singh, 2018). These categories represent the syntactic role of the word in the sentence (e.g. a noun, verb, adjective, etc.).

NLP NER technique is chosen as it can identify and classify entities such as dates, times, people, locations, and other relevant components of the sentence, which is the core to extract specific structured information from natural language text. The text that will be processed using NLP is also short and contains one event per input only, which makes directly identifying entities

without any relational linking sufficient. To make NER more suitable for students, extra emphasis is put on recognizing student-specific keywords like ‘assignment’, ‘test’, quiz’, and ‘due date’.

2.3.3.2 Application of NER in Event Extraction

Event extraction using NER has been applied in use cases similar to EduCalendar. Nair et al. (2020) and the team employed NER to extract event information from the user's Gmail inbox in a real-time environment to eliminate the time spent manually opening emails and remove the risk of human error or oversight. They utilized Support Vector Machine (SVM), a binary classifier machine learning technique to train a model to classify emails to mails that contain events and those that does not to filter out irrelevant emails. For event extraction with NER, they manually annotated their training dataset with tags like DATE, TIME, VENUE, and LINK. The dataset follows a BIO tagging scheme, a widely used tagging format in named entity recognition. They used spaCy’s existing English model as a base to further train the model to suit their use case’s needs. The extracted data is later validated and filtered using Selector and Validator module. The accuracy of their spaCy NER model is 83%, where 83 out of 100 mails. In the 17 mails, 12 of the mails got the venue wrong. Figure 2.19 below shows the training spaCy model.

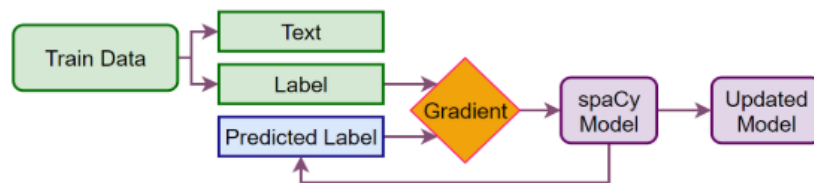


Figure 2.19. Training spaCy model. (Nair et al., 2020).

Shreyas et al. (2018) and the team also implemented event extraction NER from emails to update event in calendar. The team aims to develop a browser extension that detects events in emails and extract the events into the user’s calendar. Their process is visualized in Figure 2.20.

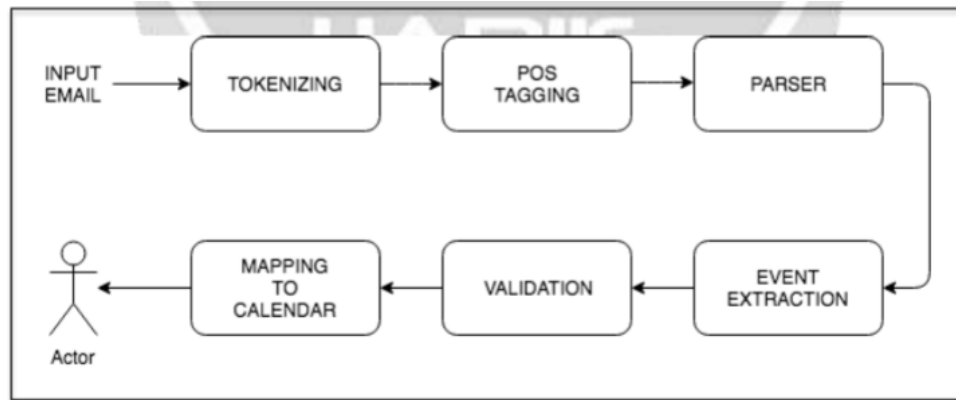


Figure 2.20. System architecture pipeline of Shreyas et al. (2018)'s proposed system.

After retrieving the contents of email, they tokenized the contents using NLTK library's `nltk.tokenize` function, they further narrow down the relevant keywords using POS tagging and date/time string parser. During event extraction, they divided it into four sections, which are subject extraction, location extraction, date extraction, and time extraction as not all emails with an event contain all of the details. Lastly, the extracted event is stored in JSON format and will then be mapped to Google Calendar using its API. The results of their test cases demonstrate generally high accuracy across various scenarios: 93.3% (28 out of 30 test cases) when all event details are provided, 90% (27 out of 30 test cases) when some details are missing, and 90% (9 out of 10 test cases) for handling incorrect input. However, the accuracy of their proposed system drops to 75% (15 out of 20 test cases) when the messages are written informally.

2.3.3.3 Utilized NLP Library for NER

SpaCy is a an open-source, industrial-strength NLP library designed specifically for advanced text processing (Jugran et al., 2021; Nair et al., 2020). Its focus is primarily on natural language processing and is capable of handling large volumes of text (spaCy, n.d.-a).

NLTK is another open-source NLP library in Python which is widely used for educational purposes as well as for research in computational linguistics and NLP (Shreyas et al., 2018). It is mainly used by academic users like students, educators, and researches to write papers and run benchmarks (NLTK, n.d.).

The spaCy python NLP library has been chosen over the NLTK python NLP library as SpaCy is more production-oriented compared to NLTK, which is more research and academic focused. SpaCy’s production-focused philosophy made it to be built with efficiency in mind. Thus, it is faster and requires lower memory usage. SpaCy’s training pipeline utilizes Deep Learning model, specifically neural networks to power NER (spaCy, n.d.), their training process is shown in Figure 2.19.

2.3.3.4 Evaluation Metrics for NER Model

Model performance is essential to ensure accurate extraction of entities from text. Among the most commonly used evaluation metrics are precision, recall, and F1-score, which are from classification theory and are widely adapted in NLP tasks (Microsoft Learn, 2024).

Precision is defined as the proportion of named entities identified by the model that are correctly labeled. It is calculated by dividing the number of true positive predictions (TP) by the sum of true positives and false positives (FP) (Esuli & Sebastiani, 2010), shown in Equation (1) below:

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

Recall measures the model’s ability to retrieve all relevant named entities. It is computed as the number of true positives divided by the sum of true positives and false negatives (FN) (Esuli & Sebastiani, 2010), shown in Equation (2) below:

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

F1-score is the harmonic mean of precision and recall, which offers a balanced measure when both false positives and false negatives are concerned (Microsoft Learn, 2024), shown in Equation (3) below:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (3)$$

In NER, evaluation often uses exact match evaluation, where an entity prediction is only considered correct if it gets both the start and the end of the entity exactly right, and also correctly labels the entity (Esuli & Sebastiani, 2010). This is why precision, recall, and F1-score are used as these metrics ignore the large number of irrelevant words and only judge the model based on how well it finds the important entities (Derczynski, 2016). A model with high precision but low recall finds only a few entities, but most of them are correct, while the reverse finds many entities, but also makes more mistakes. F1-score provides a balance between these two, giving a more complete picture of the model’s performance (Microsoft Learn, 2024).

2.3.4 Software Development Tools

A software development tool is a program that connects input data sources with applications or delivers output to end users (Rathod, n.d.). It encompasses various applications used by developers, including code editors, libraries, debuggers, and performance analysis tools

(Rathod, n.d.). In this project, the software development tool used to develop the proposed system is Visual Studio Code.

2.3.4.1 Visual Studio Code

Visual Studio Code (VS Code) is a lightweight, open-source code editor developed by Microsoft based off the Electron framework (Visual Studio Code, 2024). It's widely used by developers for various programming tasks due to its wide support for multiple programming languages and intuitive extensions, which are provided by both official sources and its community of users, and its availability across operating systems like Windows, macOS, and Linux (Visual Studio Code, 2024). Key features of VS Code include intelligent code editing, integrated terminal, debugging, and Git integration which allows for streamlined version control.

2.4 Proposed System

To develop the proposed system, EduCalendar, the best and suitable features from the reviewed existing systems will be incorporated together with some academic specific attributes. For front-end development, Figma will be used for early prototype creation, while Svelte framework and HTML, CSS, and JavaScript language will be used for front-end code development. Python programming language, MySQL for database, and NLP NER technique will be used for back-end development. The web-based code development will be completed using VS Code. The features comparison of EduCalendar and the existing systems are tabulated below in Table 2.3.

Features	Google Calendar		SkedPal		Fantastical		EduCalendar	
Event/Task Views	Timeline, List	Grid	List, board	Kanban Grid	Timeline, List	Grid	Timeline, List	Grid
NLP Integration	X		✓		✓		✓	
Tagging system	X		✓		X		✓	

Academic performance dashboard	X	X	X	✓
--------------------------------	---	---	---	---

Table 2.3. Comparison between EduCalendar and the existing systems.

The NLP model in the proposed system will be incorporated in the event creation section, and aims to enhance SkedPal’s approach by refining post-processing after extracting event details. In SkedPal, the NLP model sometimes leaves extra spaces in event titles, such as ‘class meeting with SV at library,’ where a double space appears between ‘SV’ and ‘at.’ This issue arises from user inputs like ‘class meeting with SV next Monday at library.’ Additionally, as shown in the example above, SkedPal’s model occasionally fails to detect locations accurately. While Fantastical’s NLP model is more robust, its availability is limited to the iOS ecosystem.

None of the three existing systems reviewed are designed specifically for students, which is why they lack student-centric features such as academic performance tracking and countdown dashboards for assignments and tests. Since the proposed system is specifically tailored for students, there is a need for features that address the unique needs of students, so most of their academic related information are focused in one system.

2.5 Summary

Chapter Two reviewed and described three existing systems with relevant features and capabilities in the digital calendar ecosystem. The existing systems’ key features, pros and cons are compared and evaluated in depth in table format. This chapter also reviewed the tools and technologies that will be utilized to develop EduCalendar. In conclusion, the proposed system will be able to improve daily planning for students by automating event creation with NLP.

Chapter 3 : Requirement, Analysis and Design

3.1 Introduction

This chapter explains the methodology implemented to develop EduCalendar, the proposed system. As mentioned in Chapter 1, the Prototyping Model (PM) is adopted to check the feasibility of the idea and to validate whether the proposed solution meets the desired requirements early on.

3.2 Methodology Summary

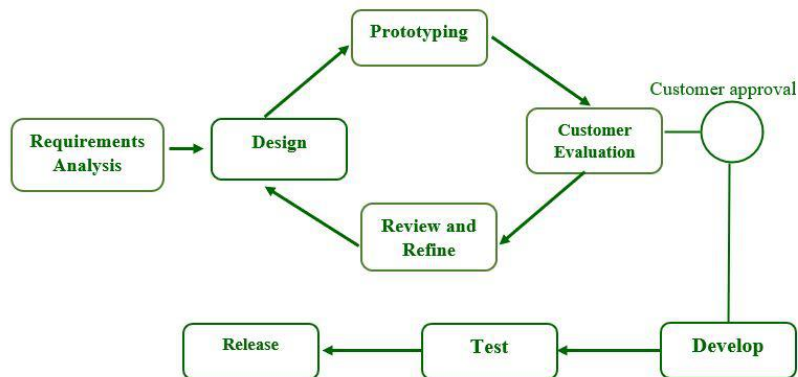


Figure 3.1. Prototyping Model.

This project consists of eight phases, which are Requirement Analysis, Design, Prototyping, Customer Evaluation, Review and Refine, Develop, Test, and Release.

The requirement analysis phase involves gathering and analyzing user requirements through observations and reading. The goal is to identify EduCalendar's improvements, functionalities, objectives and constraints. The information is documented to outline the scope, functionality, and performance criteria of the project, providing a solid foundation for the subsequent phases.

In the Design phase, the focus shifts to developing a rough blueprint for EduCalendar. The gathered requirements are translated into design specifications that include NLP functionalities, enabling users to interact with the calendar using conversational language. Architectural diagrams that detail the communication between the NLP components and EduCalendar is created for a high-level view of how EduCalendar works. Data models and user interface designs are also outlined in this phase. Additionally, technical constraints are defined to ensure the design aligns with the project's goals.

The Prototyping phase involves building an early prototype of the system to explore and validate EduCalendar's design concepts. During this phase, a wireframe that includes key features and functionalities, which is the NLP integration, is prototyped using Figma, and is then presented to students, who are the main target audience of this system. Feedback is collected to identify areas for improvement, ensuring that it aligns closely with user expectations and project requirements.

The customer evaluation phase is where preliminary feedback is obtained from the students regarding the prototype to gather their insights. The feedback is collected and analyzed to identify any issues or areas requiring enhancement. This phase provides valuable input that helps in refining the design of EduCalendar and making necessary adjustments before moving forward. When a satisfactory prototype iteration is created, the project will move forward to the review and refine phase.

During the Review and Refine phase, the focus is on refining EduCalendar's design and prototype based on the feedback received. This involves reviewing the feedback, identifying necessary changes, and updating both the design documents and the prototype to address any issues. If needed, the prototyping phase may be revisited to ensure that the refined design aligns well with

user expectations. The result is an improved prototype and design that are better suited to meet user needs.

The Develop phase is where the final system of EduCalendar is built based on the refined design. During this phase, the design is translated into code and integrated into the system. Development best practices and coding standards are followed to ensure quality. Unit testing and integration testing are performed throughout development to verify functionality and address any issues early on. This phase produces in a fully developed system that incorporates feedback and meets the design specifications.

In the Test phase, the developed EduCalendar system undergoes testing to ensure it is accurate, reliable, functional, and meets quality standards. Functional and user acceptance tests are conducted to identify and fix bugs or issues on both the developer's side and the user's (students) side. The goal is to validate that the system performs as expected and meets all requirements. This phase ensures that the product is ready for deployment.

Finally, the Release phase involves deploying the final system of EduCalendar to students. The system is rolled out to production environments, and support and training are provided to users as needed. The result is a live product that is accessible to users.

Table 3.1 shoes the summary of each phase.

Phases	Procedures	Technologies
Requirement analysis	System planning - Documentation	Microsoft Word
	Requirement Elicitation - Review on existing systems - Questionnaire	Google form Microsoft Word

	Requirement Analysis - Functional requirement - Software requirement - Hardware requirement	
Design	System design - Logical design - o Class Diagram - o Data Flow Diagram (DFD)	Draw.io Visual Paradigm
Prototyping	Wireframe - Mobile interface - Desktop interface	Figma software
Customer evaluation	Feedback on the user interface design from university students.	Google form
Review and refine	Amend early wireframe, any changes related to the UI and UX will be made in this phase	Figma software
Develop	- Front-end development - Backend development - NLP integration	SvelteKit HTML, CSS, JS Python Firebase Visual Studio Code
Test	- Functional test - User acceptance test (UAT)	Microsoft Word Google form
Release	System Deployment - Deploy custom NER model and system - Documentation	Hugging Face Space Vercel Microsoft Word

Table 3.1. Summary of the Prototype model methodology.

3.3 System Planning

System planning is the initial phase in the software development life cycle. It involves defining the overall goals, objectives, and scope of the project to ensure its success. The main aim

is to create a solid foundation by understanding the project's requirements, constraints, and feasibility before actual development begins.

3.4 Requirement Elicitation and Analysis

The system requirements and expectations are gathered during this phase using questionnaires. The results of requirement elicitation will be used as requirement analysis to determine the functional, software, and hardware requirements.

3.4.1 Requirement Elicitation

A questionnaire approach is utilized for the proposed system's requirement elicitation section. Google form is used to create a questionnaire of 16 questions which four sections, User demographics (Question 1 to 3), Digital Calendar Usage (Questions 4 to 7), User Experience and AI integration in Digital Calendars (Questions 8 to 12), and UI Preferences and Evaluation (Questions 13 to 16). The questionnaire is distributed online to 23 respondents who are currently an active university student. The findings of the last three sections are discussed and analyzed. The data gathered are evaluated and presented in the form of a pie chart.

3.4.1.1 Digital Calendar Usage

4. Have you ever regularly used a digital calendar (e.g., Google Calendar, Apple Calendar, Outlook Calendar) to keep track of your academic responsibilities (assignments, projects, quizzes) and daily schedules?

23 responses

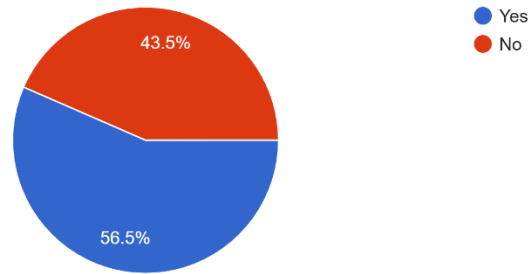


Figure 3.2. Have you ever regularly used a digital calendar (e.g., Google Calendar, Apple Calendar, Outlook Calendar) to keep track of your academic responsibilities (assignments, projects, quizzes) and daily schedules?

Figure 3.2 shows that more than half of the respondents (56.5%, 13 respondents) are regular users of digital calendar applications, and the rest (43.5%, 10 respondents) does not. The respondents are then divided into two sections, those who regularly uses a digital calendar, and those who does not.

5. If your answer in (4) is no, why didn't you use a digital calendar regularly?

10 responses

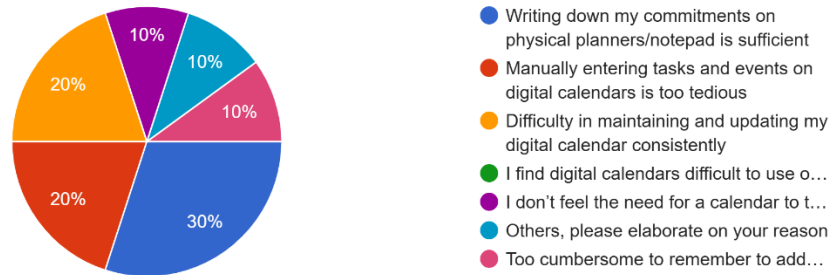


Figure 3.3. If your answer in (4) is no, why didn't you use a digital calendar regularly?

On Figure 3.3, for the respondents who answered no on the Question 4, their reason for not committing to digital calendars are quite split. The majority of them (30%, three respondents) thinks that writing down their commitments physically on planners and notepads is sufficient. 10% (1 respondent) expressed that “I don’t feel the need for a calendar to track my responsibilities”.

During the initial deployment of the questionnaire, the option "Others, please elaborate on your reason" was included under Question 5 but inadvertently did not allow respondents to provide written input, which caused it to have one respondent (10%) picking it without being able to specify their reason. This issue was identified and subsequently corrected, and the “Too cumbersome to remember to add events consistently” (10%) option is a respondent provided answer with the other option.

6. If your answer in (4) is yes, what is the name of the digital calendar application?
13 responses

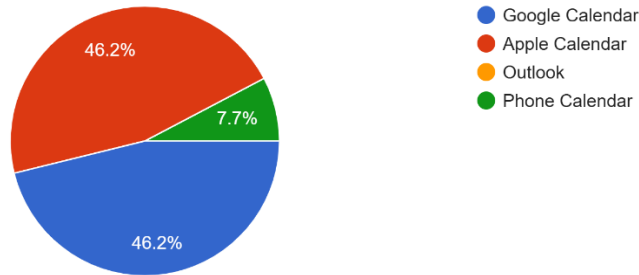


Figure 3.4. If your answer in (4) is yes, what is the name of the digital calendar application?

For respondents who answered yes in Question 4, they are directed to Question 6 to identify the digital calendars they are actively using and are subsequently familiar with it. Figure 3.4 shows that the main digital calendars used are split into Google Calendar (46.2%, six respondents) and Apple Calendar (46.2%, six respondents). The remaining 7.7% (one respondent) defaults to the digital calendar their phone provides. No one out of the 13 responders used Outlook calendar as their go to digital calendar option.

7. If your answer in (4) is yes, how often did you use the digital calendar?
13 responses

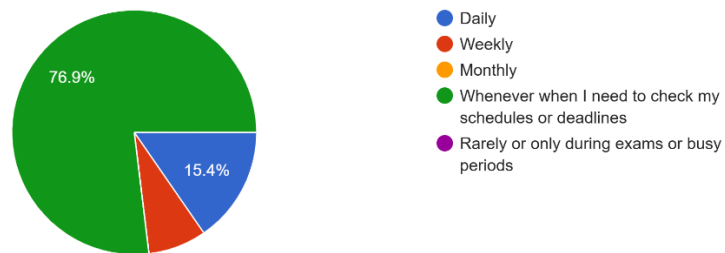


Figure 3.5. If your answer in (4) is yes, how often did you use the digital calendar?

For respondents who answered yes in Question 4, they are also directed to Question 7 to observe their digital calendar usage frequency. Figure 3.5 displays that the greater part of the

respondents (76.9%, 10 respondents) only open the digital calendar whenever they need to check their schedules or deadlines, followed by daily check-ins (15.4%, two respondents), and weekly check-ins (7.7%, one respondent). No respondent picked the “Monthly” option.

3.4.1.2 User Experience and AI integration in Digital Calendars

8. How important do you think it is to be able to keep track of your academic responsibilities (assignments, projects, quizzes) and daily schedules?
23 responses

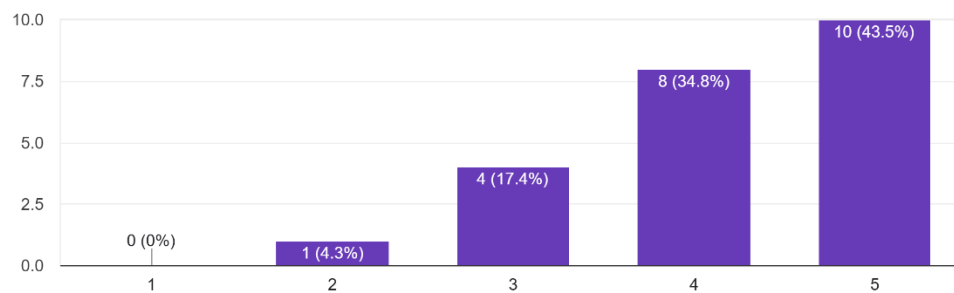


Figure 3.6. How important do you think it is to be able to keep track of your academic responsibilities (assignments, projects, quizzes) and daily schedules?

All respondents are then guided to this question to study how the responders think of the importance of keeping track of their academic responsibilities and daily schedules out of the scale of one to five. In Figure 3.6, most of the respondents are in consensus that it is very important (43.5%, 10 responders). Eight respondents (34.8%) think it's important and four respondents (17.4%) thinks its moderately important. one respondent (4.3%) think it is not very important, and no one think that it is not important at all.

9. Do you find manually entering calendar events on digital calendars tedious or time-consuming?
23 responses

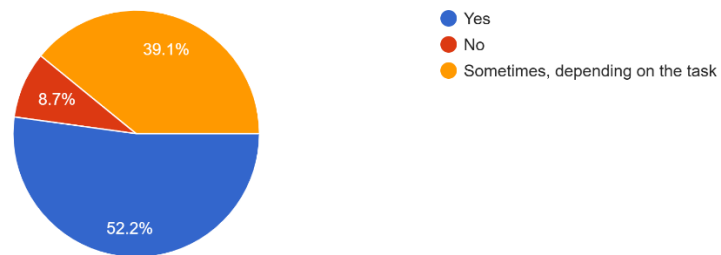


Figure 3.7. Do you find manually entering calendar events on digital calendars tedious or time-consuming?

Figure 3.7 depicts that the majority of the respondents (52.2%, 12 respondents) agreed on the notion that manually entering calendar events on digital calendars is tedious and time-consuming, 39.1% (9 respondents) only find the situation to be cumbersome depending on the task. Only two respondents (8.7%) do not agree.

10. If there is such an app that would be able to create events given some inputs in the digital calendar, do you think it will be beneficial to you?
23 responses

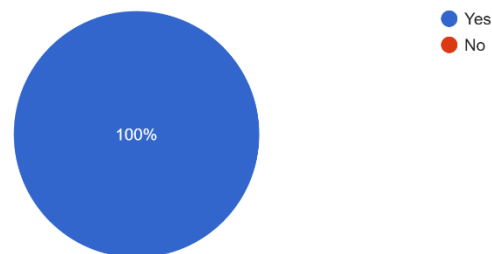


Figure 3.8. If there is such an app that would be able to create events given some inputs in the digital calendar, do you think it will be beneficial to you?

Figure 3.8 demonstrates unanimous agreement among respondents (100%, 22 respondents) regarding the potential usefulness of an app that can create events in a digital calendar based on inputs, indicating that all respondents believe such an app would be beneficial to them.

11. Have you heard of AI integration in digital calendars?
23 responses

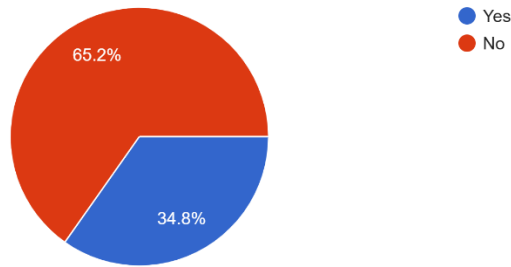


Figure 3.9. Have you heard of AI integration in digital calendars?

As shown in Figure 3.9, the majority of respondents (65.2%, or 15 individuals) reported that they were unaware of AI integration in digital calendars. In contrast, only 34.8% (8 respondents) indicated that they were informed about this topic.

12. In what way do you think AI integration in digital calendars would help you as a student?
23 responses

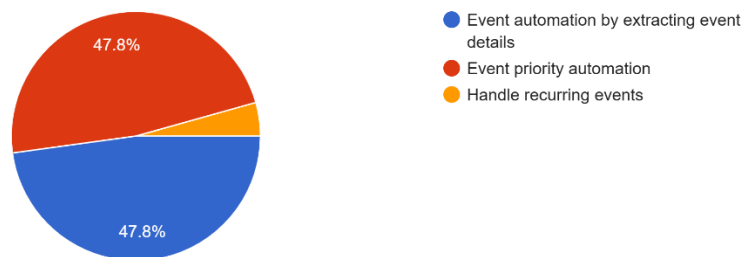


Figure 3.10. In what way do you think AI integration in digital calendars would help you as a student?

Figure 3.10 illustrates respondents' preferences regarding the potential benefits of AI integration in digital calendars. An equal proportion of respondents (47.8%, 11 respondents) selected "Event priority automation" and "Event automation by extracting event details" as the most helpful features. A smaller segment (4.3%, one respondent) chose "Handle recurring events."

3.4.1.3 UI Preferences and Evaluation

The **Figure 1** and **Figure 2** shown below are 2 different types of UI arrangement on a digital calendar web system on a mobile screen. Answer question 13 and 14 based on the 2 figures shown below.



Figure 3.11. The images shown in the questionnaire as visual reference for responders to answer question 13 and 14.

13. Between the header section of Figure 1 and 2 (navigation between the home, calendar and course page are hidden within the menu icon), which type of header do you prefer?
23 responses

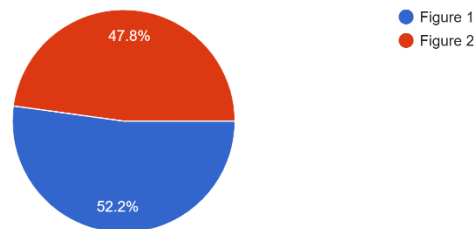


Figure 3.12. Between the header section of Figure 1 and 2 (navigation between the home, calendar and course page are hidden within the menu icon), which type of header do you prefer?

Figure 3.11 displays the images provided in the questionnaire as visual references to assist respondents in answering questions 13 and 14. Figure 3.12 presents the results of respondents' choices for question 13. The data reveals that Figure 1, selected by 52.2% of respondents (12 respondents), holds a slight edge over Figure 2, chosen by 47.8% of respondents (11 respondents).

14. Between the body section the homepage of Figure 1 and 2, which layout do you prefer?
23 responses

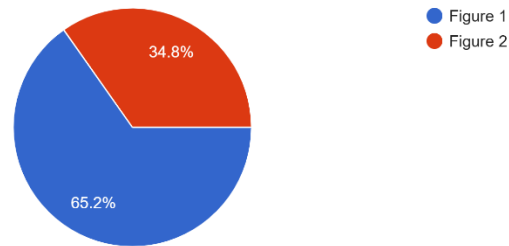


Figure 3.13. Between the body section the homepage of Figure 1 and 2, which layout do you prefer?

For the results of question 14, Figure 3.13 shows that most of the respondents (65.2%, 15 respondents) prefer the homepage section of Figure 1, and the rest (34.8%, eight respondents) choose Figure 2.

For mobile screen, **Figure 3** shows the UI when the users click on the plus icon on the bottom right of the screen (as shown in Figure 1) to create a new event, and **Figure 4** shows how the event details are presented after users enter an event prompt in the textbox.

For desktop screen's interface, is shown in **Figure 5** and **6**.

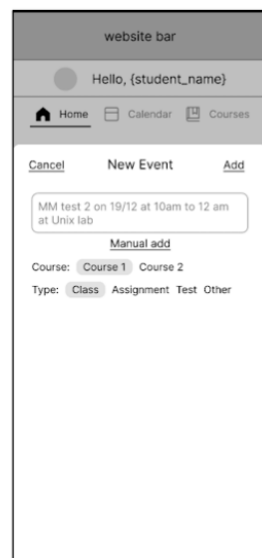


Figure 3

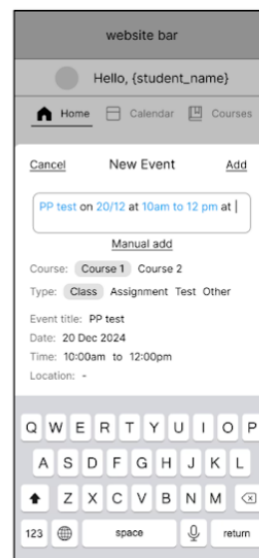


Figure 4

Figure 3.14. The images shown in the questionnaire as visual reference for respondents to answer question 15.

15. Is the "Create Event" interface on both mobile (Figure 3 & 4) and desktop screen (Figure 5 & 6) visually clear and easy to understand?
23 responses

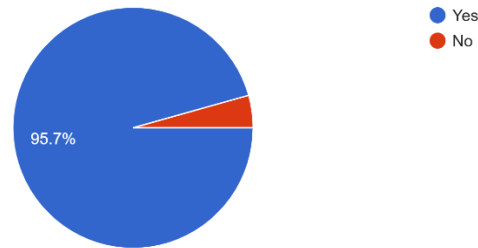


Figure 3.15. Is the "Create Event" interface on both mobile (Figure 3 & 4) and desktop screen (Figure 5 & 6) visually clear and easy to understand?

Figure 3.14 displays the images provided in the questionnaire as visual references to assist respondents in answering questions 15. Figure 3.15 presents the results of respondents' choices for question 15. The data reveals that 95.7% of respondents (22 respondents) think that the create event interface is visually clear and easy to understand, with only one respondent (4.3%) disagreeing.

Figure 7 shows the courses page of a digital calendar web system on a mobile screen, and **Figure 8** is the courses page on a desktop screen

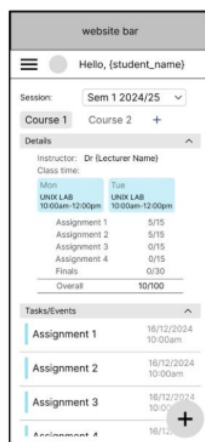


Figure 7

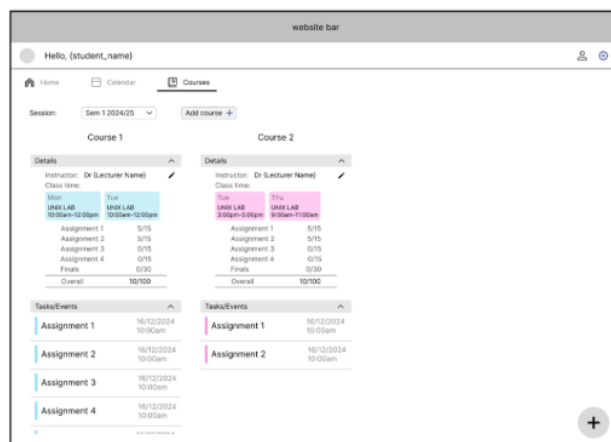


Figure 8

Figure 3.16. The images shown in the questionnaire as visual reference for respondents to answer question 16.

16. Are the details displayed in an organized and easy-to-read format?
23 responses

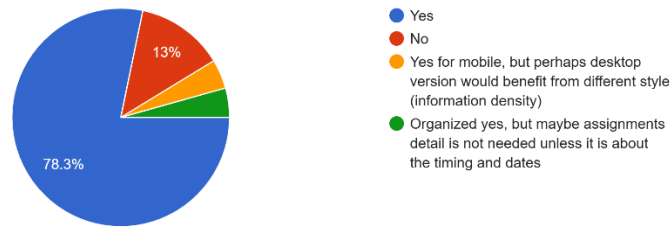


Figure 3.17. Are the details displayed in an organized and easy-to-read format?

Figure 3.16 shows that most respondents (78.3%, 18 respondents) find the details organized and easy to read. However, 13% disagree, citing dissatisfaction with the format. A small portion (4.3%, one respondent) believe the mobile version is fine but suggest the desktop version could use a different style to improve information density. Another 4.3% (1 respondent) feel assignment details are unnecessary unless focused on timing and dates.

3.4.2 Requirement Analysis

EduCalendar's user requirements and specifications are collected from requirement elicitation through questionnaires and the review of the existing similar systems in Chapter 2. The information gathered are analyzed and are used in determining the functional requirement, system requirement, and hardware requirement.

3.4.2.1 Functional Requirement

The targeted users of EduCalendar are students. The features of EduCalendar are determined based on the results of the questionnaires in requirement elicitation and the review of the existing similar systems in Chapter 2. Table 3.2 shows the features and their specific details.

Features	Details
Sign up/Login	Mandatory for all users to create an account to use the features of this system.
NLP integration for event automation	Event creation is automated by implementing NLP NER technique. Users will have to input a short sentence like ‘Maths test on 19/11 from 10am to 12pm at UNIX lab’, and the system will detect entities such as event title, date range, time range, location, and frequency. Users will then have to choose the course related to the event, and the type of event (Class, Assignment, Test, Other). If the system is unable to accurately capture the entities of the user input, users can opt for manually inputting the event details. The manual input is added as a fail-safe option
Event view, update, and deletion	The events can be viewed in list and timeline form. Users can update event details or delete events as they please.
Course add, view, update, and deletion	To add events, it is necessary to link it under a course. Courses can be added, viewed, updated and deleted according to the user’s means. Each course has details such as: course instructor name, class slot, and grades. The class slots are added through creating an event for the course with event type ‘Class’.
Add, view, update, and delete grades for each course.	Grades for each course’s assignments and tests can be added, viewed, updated, and deleted according to the user’s means. The grades are viewed under the course details section.
Sort event based on date or course	Events can be sorted based on date or course on the homepage.
Filter event based on time range	Events can be filtered based on time ranges such as ‘All’, ‘This week’, and ‘This month’.

Table 3.2. Features and their details of EduCalendar.

3.4.2.2 Software Requirement

The software requirement summary for the development of the system is shown in Table 3.3 below.

Requirements	Technologies
Web Development tool	Visual Studio Code
Prototype Tool	Figma software
Web framework	Svelte framework
Programming languages	HTML CSS JavaScript Python
Database	MySQL
NLP library	spaCy
Diagramming tool	Visual Paradigm Draw.io
Survey Administration Tool	Google Forms
Project Planning tool	Microsoft Excel
Documentation	Microsoft Word

Table 3.3. Software requirement of EduCalendar.

3.4.2.3 Hardware Requirement

The hardware requirements for the development of EduCalendar is detailed in Table 3.4 below.

Requirements	Minimum	Recommended
Operating System	Windows 9/10 (64 bit)	Windows 11 (64 bit)
Processor	2.6 GHz	Higher than 2.6 GHz
Memory	4GB RAM	8GB RAM
Storage	2 GB available space	10 GB available space
Monitor	1280 x 800 screen resolution	1920 x 1200 screen resolution

Table 3.4. Hardware requirement of EduCalendar.

3.5 System Design

The system design includes building database relationships, creating the system's data flows, and designing the user interfaces. The proposed system design has two sections, logical design and user interface design's wireframe.

3.5.1 Logical Design

The logical design of EduCalendar's NoSQL database format is presented in Class Diagram form.

3.5.1.1 Class Diagram

The Class diagram is illustrated in Figure 3.18 below.

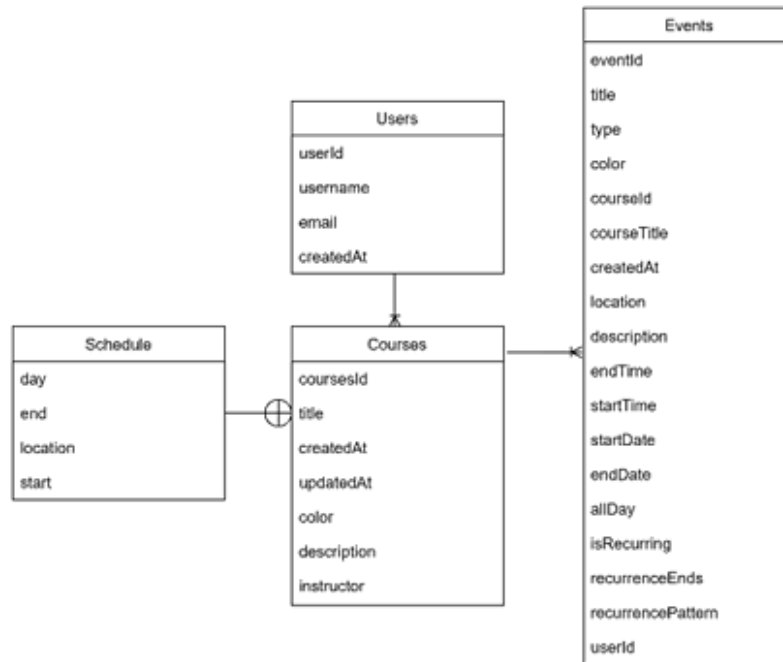


Figure 3.18. Class diagram of EduCalendar.

Figure 3.18 shows the class diagram of EduCalendar, featuring the entities Event, Course, Performance, and User. The Event entity represents calendar events, Course categorizes each event, Performance tracks graded items within courses, and User stores account data.

3.5.1.2 Data Flow Diagram

The data store dictionary of the DFD is detailed in Table 3.5 below.

Data Store ID	Data Store Name	Data Store Description	Data Store Element				
			Name	Description	Base /Derived	Type of data	Default value
D1	Event	Keep event details	Event ID	An event's unique identifier	Base	Char/Alphanumeric	-

			Event Title	The event's title	Base	Char/Alphanumeric	-
			User ID	The user's unique identifier	Base	Char/Alphanumeric	-
			Event Creation Date	The event's creation date	Base	Char/Alphanumeric	-
			Event start date	The event's start date	Base	Char/Alphanumeric	-
			Event end date	The event's end date	Base	Char/Alphanumeric	-
			Event start time	The event's start time	Base	Char/Alphanumeric	-
			Event end time	The event's end time	Base	Char/Alphanumeric	-
			Event location	The event's location	Base	Char/Alphanumeric	-
			Event recurrence	Recurring event indication	Base	Boolean	-
			Event recurrence pattern	Recurring event's pattern	Base	Char/Alphanumeric	-
			Event course tag	The course that the event is under	Base	Char/Alphanumeric	-
			Course ID	A course's unique identifier	Base	Char/Alphanumeric	-
			Event description	The prompt used to generate the event	Base	Char/Alphanumeric	-
			Event color	The course color that the event is under	Base	Char/Alphanumeric	-
			Event type tag	The type of an event (Assignment, Test, Other)	Base	Char/Alphanumeric	-

			Event All Day	All day event indication	Base	Boolean	-
D2	Course	Keep course's events and performance metrics	Course ID	A course's unique identifier	Base	Char/Alphanumeric	-
			User ID	The user's unique identifier	Base	Char/Alphanumeric	-
			Course Title	The course's name	Base	Char/Alphanumeric	-
			Course Color	The course's color	Base	Char/Alphanumeric	-
			Course Creation Date	The course's creation date	Base	Char/Alphanumeric	-
			Course Description	The course's description	Base	Char/Alphanumeric	-
			Course Schedule	The course's schedule	Base	Char/Alphanumeric	-
			Course Instructor	The course's instructor name	Base	Char/Alphanumeric	-
D3	User	Keep user login details	User ID	The user's unique identifier	Base	Char/Alphanumeric	-
			Username	The user's display name	Base	Char/Alphanumeric	-
			User email	The user's email	Base	Char/Alphanumeric	-

Table 3.5. The data store dictionary of EduCalendar.

The context diagram of the DFD is shown in Figure 3.18 below.

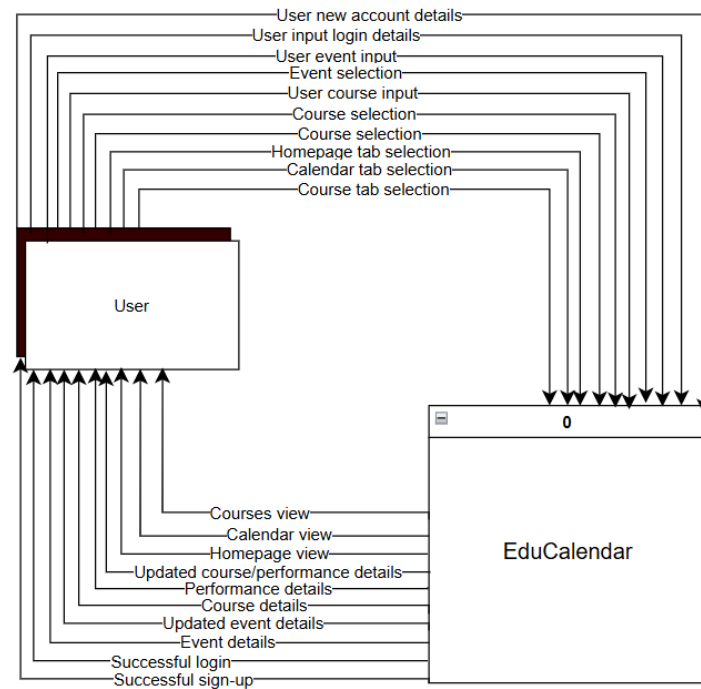


Figure 3.19. Context diagram of EduCalendar.

The context diagram of the proposed system illustrates that the primary external entity interacting with the system is the user, specifically university students, who form the target audience of EduCalendar. The data flows include user inputs for login, sign-up, event creation, course enrollment, and performance details, as well as tab selections. In response, the system generates outputs to the user's inputs and selections accordingly.

The level 0 of DFD is shown in Figure 3.19 below.

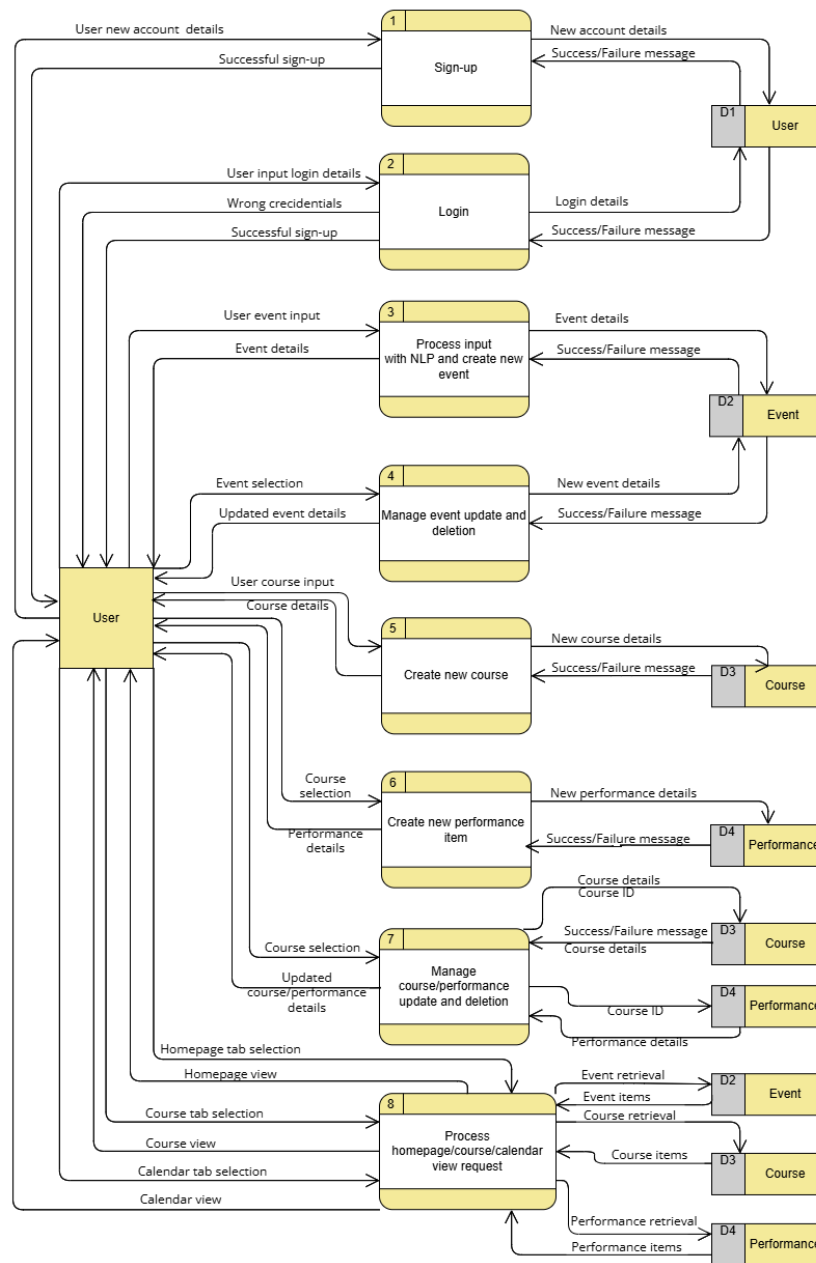


Figure 3.20. Level 0 DFD of EduCalendar.

The level 0 DFD shows that the proposed system has eight main processes and four data stores. Process one and two covers the user sign up and login process, process 3 covers the NLP

integration in the proposed system, and the rest of the process manages the creation of a new course and performance item, and how the event, course, and performance items are updated and deleted.

The level 1 DFDs are shown below in Figure 3.20 to 3.27.

Figure 3.21 shows the level 1 DFD of the sign-up process.

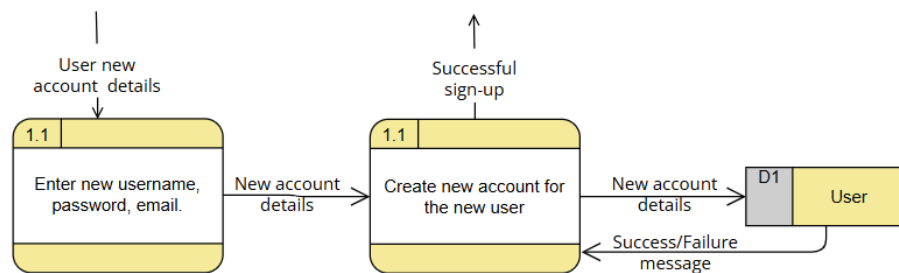


Figure 3.21. Sign-up process.

The users are required to create an account to be able to use the system so the system can keep track of the events, courses, and other information of the user when they use the system. To sign-up, the new users are prompted to enter their email, username, password, and password confirmation. Error messages will appear if the email entered does not have the structure of an email, and if the password and password confirmation does not match.

Figure 3.22 shows the level 1 DFD of the login process.

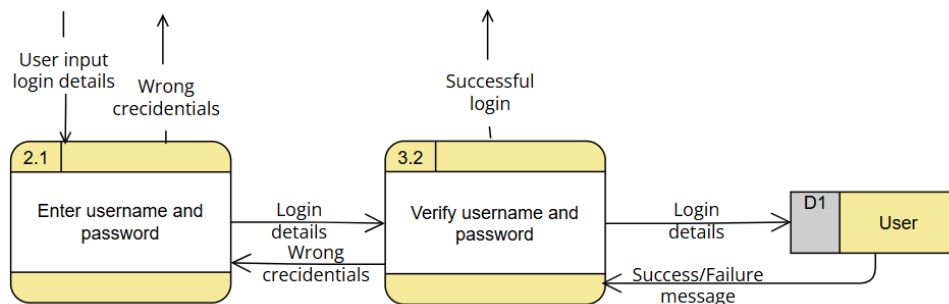


Figure 3.22. Login process.

For users who has already made an account on EduCalendar, they will be prompted to enter their username and password to login to the system. If the username and password does not match, an error message will appear and prompt the user to reenter the correct username or password. When the user successfully logs in, they will be directed to the homepage of EduCalendar.

Figure 3.23 shows the level 1 DFD of the process input with NLP and create new event process.

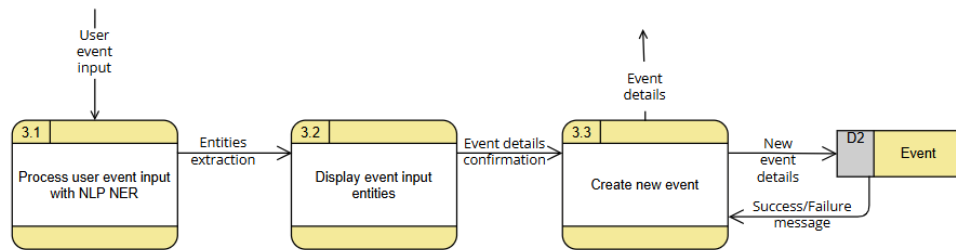


Figure 3.23. Process input with NLP and create new event process process.

The NLP NER information extraction happens when users are creating a new event. Users are prompted to enter a short sentence describing the event that they wished to be created, for an example ‘Math presentation on 2/1/2025 at UNIX Lab at 10am’, and the system will extract entities such as event title, date, time, and location from the short sentence input, and display the categorized extracted information to the users so they are able to confirm if the information are categorized correctly. If it is correct, user will confirm the event creation and a new event will be added into the event database.

Figure 3.24 shows the level 1 DFD of the manage event update and deletion process.

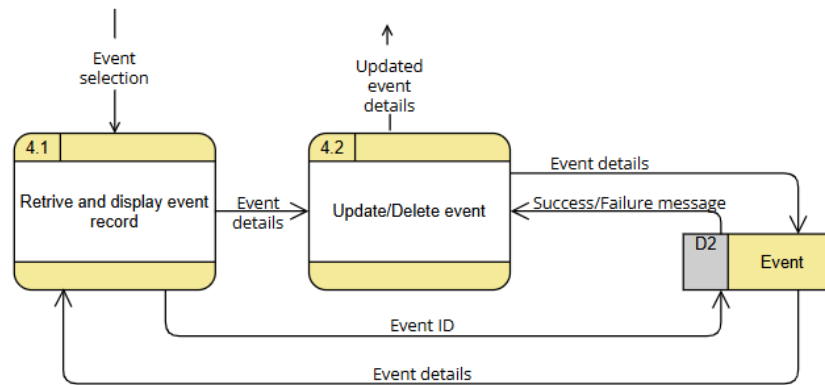


Figure 3.24. Manage event update and deletion process.

The manage events event update and deletion process manages the update and delete operations done on existing events. Users will have to select the specific event that they wish to edit or delete before being given the choice to update or delete the event. The updates on the selected event will be reflected in the database.

Figure 3.25 shows the level 1 DFD of the create new course process.

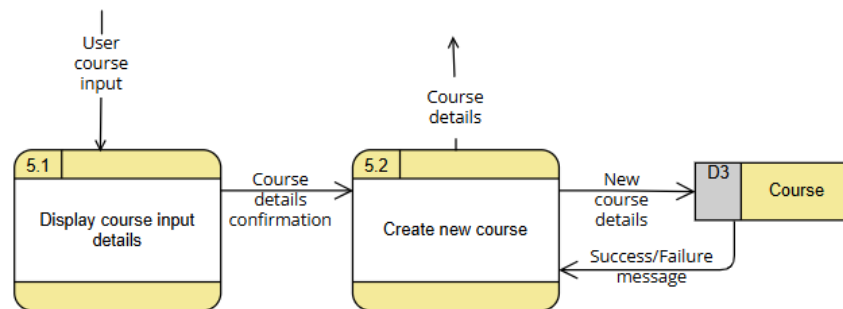


Figure 3.25. Create new course process.

The create new course process handles the creation of processes. When they use the add new course feature, they will be prompted to fill in course details like the course name and the course instructor. A new course will be successfully created after.

Figure 3.25 shows the level 1 DFD of the create new performance item process.

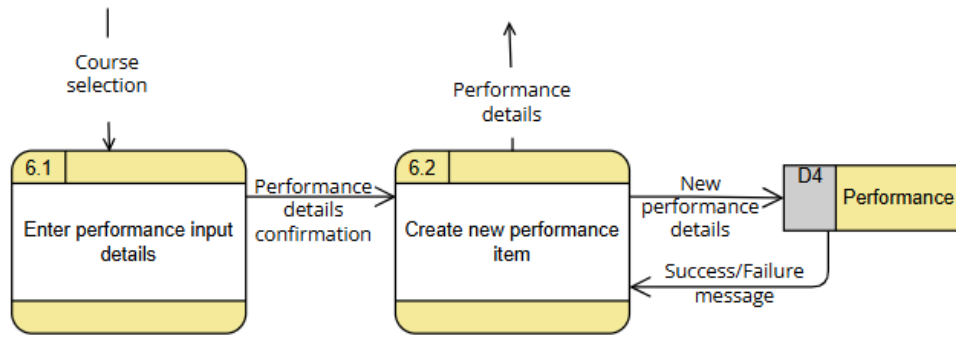


Figure 3.26. Create new performance item process.

Create new performance item process is almost the same as the create new course process, except that users will have to navigate to an existing course first, and then use the add new performance item feature. The newly added performance item will be displayed and added into the performance data store.

Figure 3.27 shows the level 1 DFD of the manage course/performance update and deletion process.

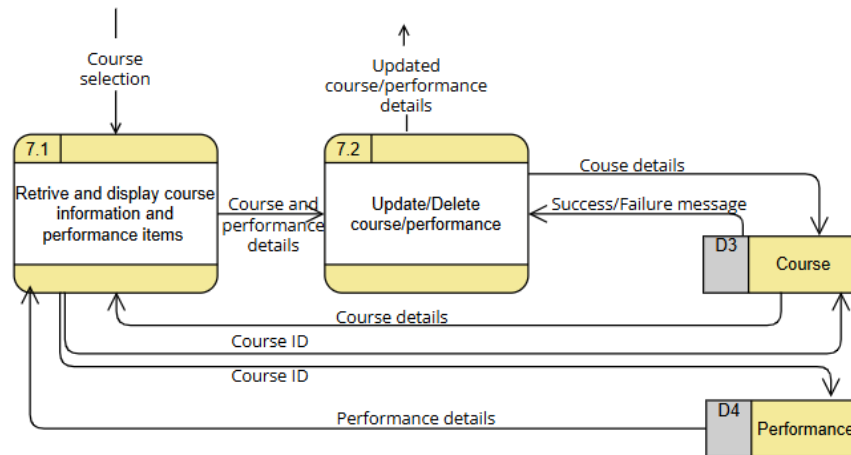


Figure 3.27. Manage course/performance update and deletion process.

Manage course/performance process handles the update and delete operations on courses and performance items. Users will have to select the course or course of the performance item that

they want to update or delete, and then they will be given the choice to update or delete a specific course or a performance item.

Figure 3.28 shows the level 1 DFD of the process homepage/course/calendar view request process.

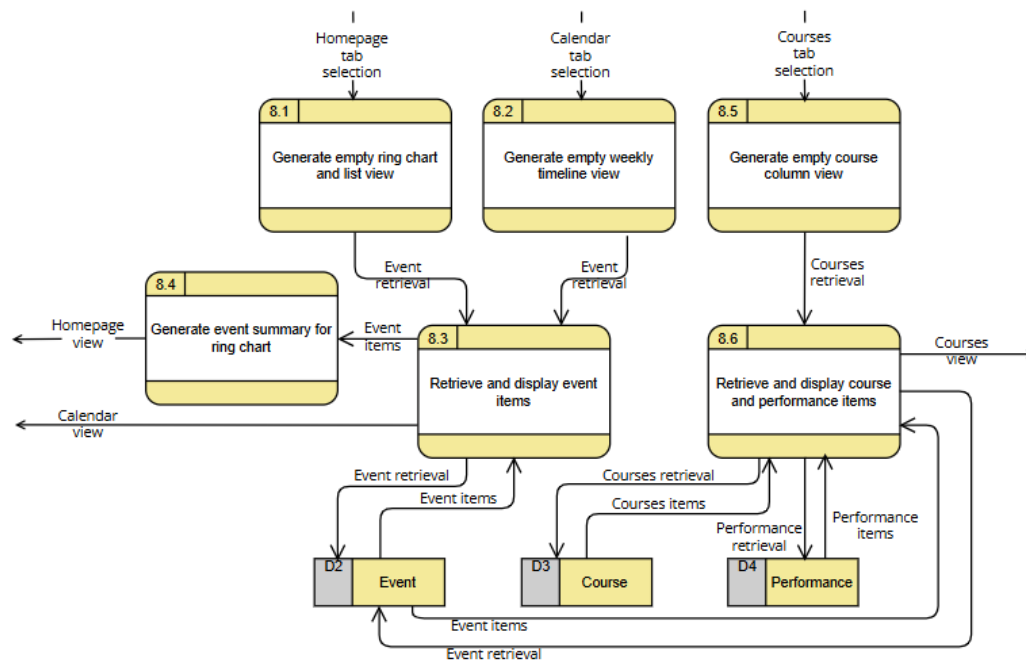


Figure 3.28. Process homepage/course/calendar view request process.

The process homepage/calendar/courses view request process handles when the users are switching between the homepage, calendar and courses pages. If users navigated to the homepage page, the system will retrieve all the events made by the user and display it in list form, ring chart form as a quick summary of the numbers of event left, and in a weekly timeline form. If the users navigated to the calendar page, the system will also retrieve event information and generate a timeline view with the weekly timeline view as its default. Lastly, if the user navigated to the courses tab, the system will retrieve course, event, and performance information and categorize the information based on the courses the event and performance items are related to.

3.5.2 Wireframe

The user interface design's wireframe of EduCalendar is visualized using a prototyping tool called Figma. As EduCalendar is a web system, both the mobile interface and desktop interfaced are designed for ease of use, ensuring a seamless and intuitive user experience across devices. The mobile interface is optimized for quick access and navigation on smaller screens, while the desktop interface is designed to leverage larger screen real estate for enhanced functionality and detailed views. Both interfaces maintain a consistent design language, ensuring familiarity and usability for users switching between devices.

3.5.2.1 Sign up/Login page

website bar



Welcome to EduCalendar!

I'm new here I have an account

Email

Username

Password

Confirm Password

Sign up

Figure 3.29. Sign up page desktop view.

website bar



Welcome to EduCalendar!

I'm new here I have an account

Email

Username

Password

Confirm Password

Sign up

Figure 3.30. Sign up page mobile view.

website bar



Welcome to EduCalendar!

I'm new here I have an account

Username

Password

Login

Figure 3.31. Login page desktop view

website bar



Welcome to EduCalendar!

I'm new here I have an account

Username

Password

Login

Figure 3.32. Login page mobile view.

Figure 3.28 and 3.29 shows the user interface of the sign-up page, while Figure 3.30 and 3.31 shows the login page. For sign-up the users are required to provide their email, create a username, create a strong password and confirm their password by typing it again. If the email does not follow the standard structure of an email, an error message will appear and the user will

be prompted to retype their email. An error message will also appear if the user chooses a username that has already been taken by another user, and they will also be prompted to type another username. For login, the user's username and password should match, or they are not allowed to log in to their account.

3.5.2.2 Event creation pop up

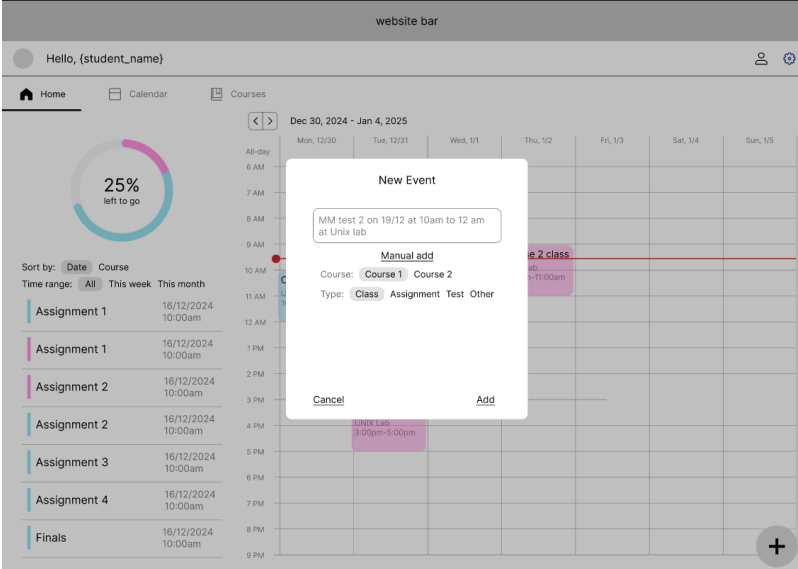


Figure 3.33. Create new event pop up desktop view.

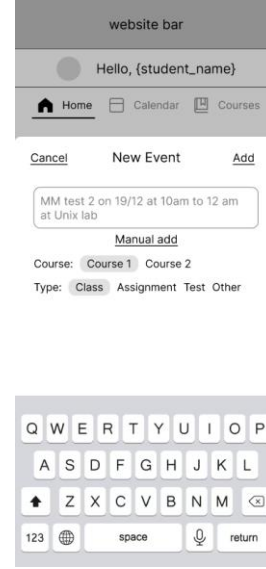


Figure 3.34. Create new event overlay mobile view.

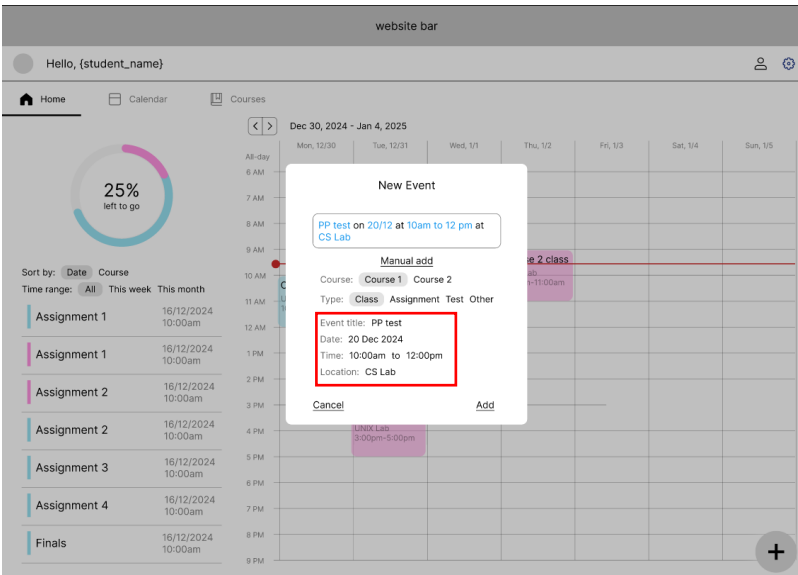


Figure 3.35. UI of event creation pop up when it receives user input desktop view.

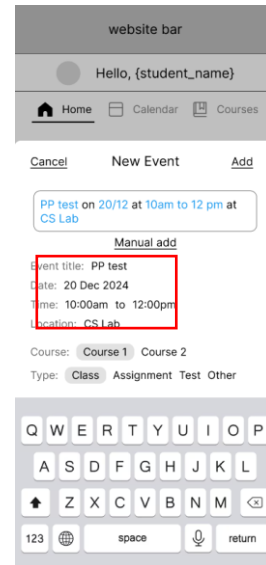


Figure 3.36. UI of event creation overlay when it receives user input mobile view.

Figure 3.32 and 3.33 shows the user interface when the user clicked the plus button at the bottom right screen. A pop up will appear, allowing the user to choose between using NLP integration to automate event creation, or create a new event by manually entering the event details. As the user type in a simple sentence like ‘PP test on 20/12 at 10am to 12pm at CS Lab’, the recognized entities will be highlighted, and displayed in their respective entity categories as outlined in the red rectangles as shown in Figure 3.34 and Figure 3.35.

After filling out the main event details, users will have to choose the course category related to the event, and the type of event, namely ‘Class’, ‘Assignment’, ‘Test’, and ‘Other’.

3.5.2.3 Homepage page

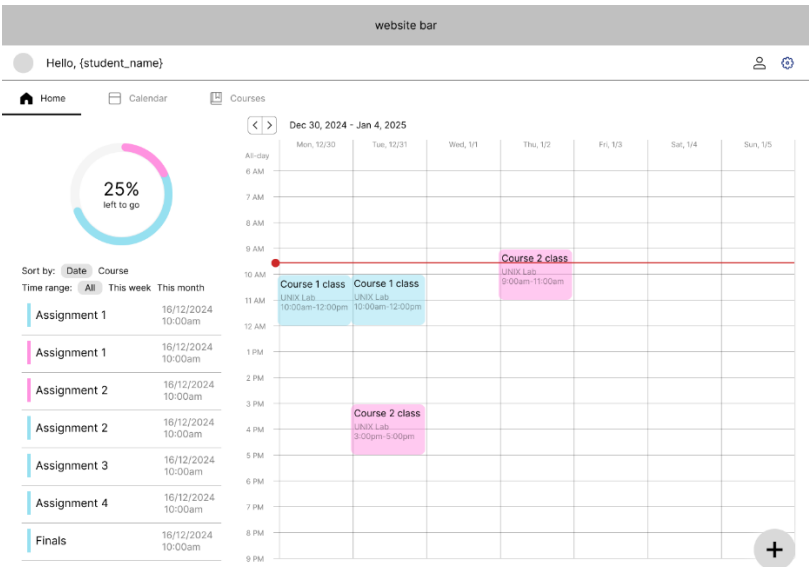


Figure 3.37. Homepage desktop view.

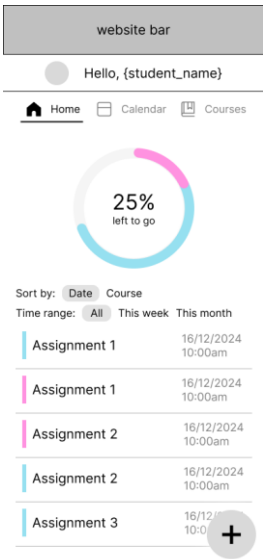


Figure 3.38. Homepage mobile view

Figure 3.36 displays the homepage in desktop and homepage view. For desktop view, the homepage is split into two sections, a quick summary of the events due soon in ring chart form and list display, and a timeline view of the current week’s events. The timeline view is absent from the mobile view as shown in Figure 3.37 due to the limited space available on a mobile screen.

3.5.2.4 Calendar page

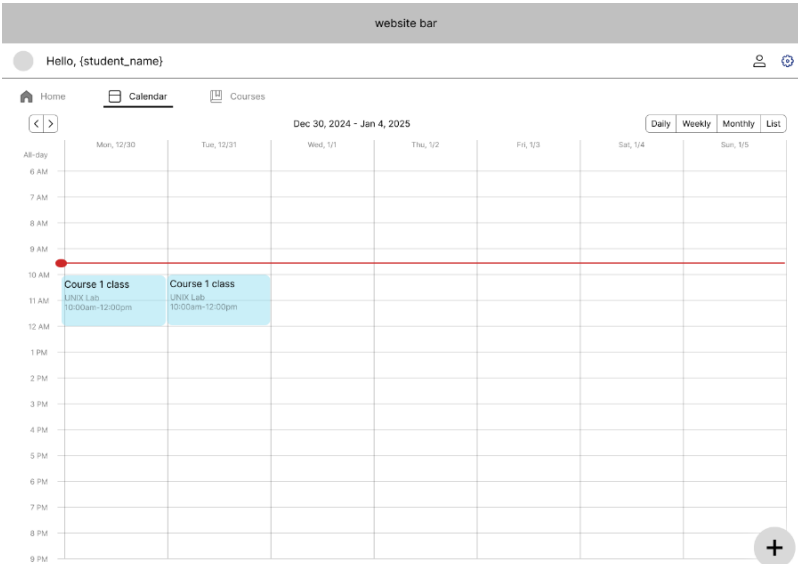


Figure 3.39. Calendar page desktop view.

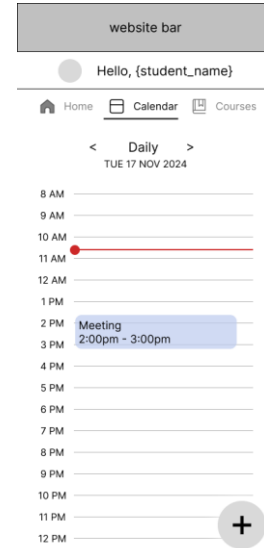


Figure 3.40. Calendar page mobile view.

Figure 3.38 and 3.39 shows the calendar page in both desktop and mobile view. Users are able to switch between daily, weekly, monthly, and list view.

3.5.2.5 Courses page

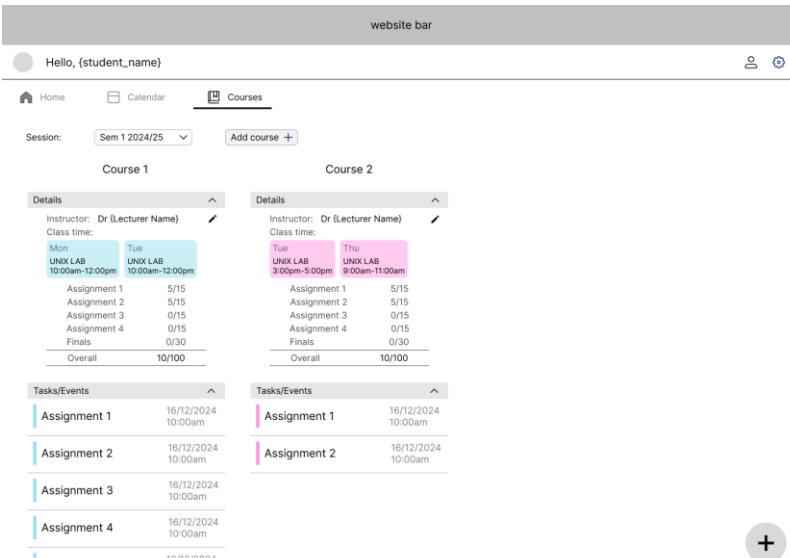


Figure 3.41. Courses page desktop view.

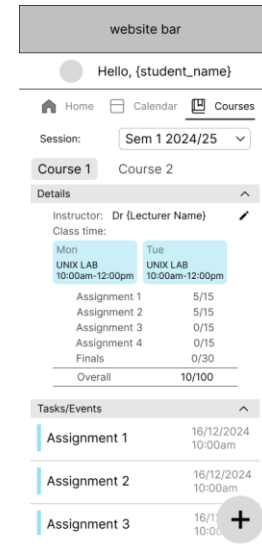


Figure 3.42. Courses page mobile view.

Figure 3.40 and 3.41 shows the courses page in both desktop and mobile view. The details of each course are displayed in columns. Each course has two sections, details section with information like instructor name, class time, and grades, and the task/events section where it displays the events that is created under that specific course. Both sections can be hide or unhide by the user. To add or edit the instructor's name or grades, users will have to click on the pen icon.

3.6 Summary

Chapter 3 describes the Prototyping Model used in the web development of EduCalendar. This chapter is divided into two parts, the first part being requirement elicitation and analysis, and the second part covering the proposed system's system design. Questionnaires has been given out using Google Forms to conduct requirement elicitation. Requirement analysis is carried out using the results of the questionnaires and the findings are utilized to determine EduCalendar's

functional, software, and hardware requirements. The logical design of the proposed system is presented through the use of Entity Relationship Diagram and Data Flow Diagram. Lastly, the wireframe of the system's user interface is designed using Figma.

Chapter 4 : Implementation

4.1 Introduction

This chapter goes through the training of the custom NLP NER model and implementation of the digital calendar web-based application with NLP NER integration in event creation. To meet the proposed system's expected outcome, the training and development will be made based on the decisions and results from Chapter 3 Requirement, Analysis, and Design. This chapter will also cover the installation details, dataset preparation, custom NER model training, validation, and testing, and in-depth explanations of the proposed system's interfaces' functionalities.

4.2 Installation and Configuration

The software development tools and environments utilized for the implementation are Visual Studio Code, Firebase Console. The specifics of the tools and environments is discussed in the following section.

4.2.1 Visual Studio Code

Visual Studio Code (VS Code) is a lightweight, open-source code editor developed by Microsoft, widely used by developers for various programming tasks due to its wide support for multiple programming languages and intuitive extensions. For this project's proposed system's development, version 1.99.3 is used alongside related plugins like Svelte for VS Code, Tailwind CSS IntelliSense, and Prettier - Code formatter.

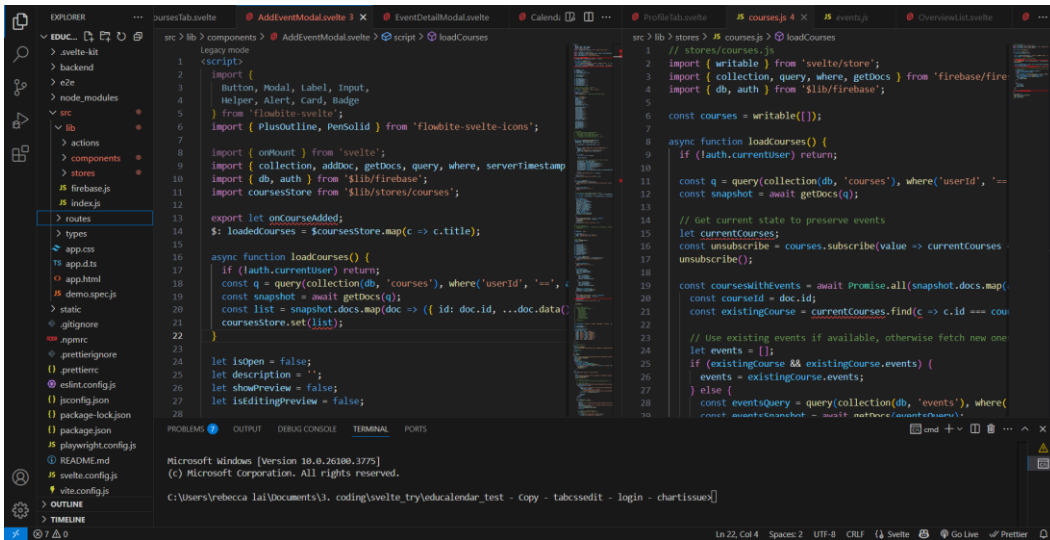


Figure 4.1 Screenshot of VS Code for the proposed system's development.

Figure 4.1 shows the screenshot of the user interface of VS Code during the development of the proposed system. The left side of the user interface shows the list of folders of EduCalendar's project folders and files, and the right side is the code editor interface. VS Code also has a built-in terminal interface on the bottom of the screen for running build scripts and debugging.

4.2.2 Firebase Console

Firebase was used to provide backend services such as user authentication and data storage through Cloud Firestore. The Firebase Console was configured early in the project lifecycle, enabling seamless integration with frontend components. Firebase allowed real-time synchronization of user-created events and course data, making it ideal for this calendar-based application where the data stored have simple hierarchical structuring.

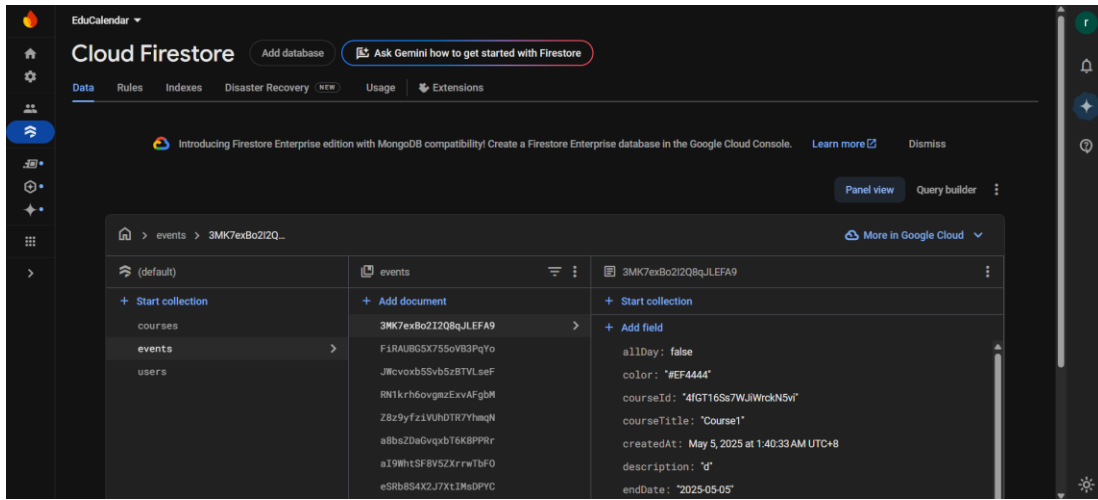


Figure 4.2 Screenshot of the Cloud Firestore user interface

Figure 4.2 shows the screenshot of the Firestore user interface. The data stored is accessed on the panel view, which is on the middle of the screen. From there, the values of each data field can be edited or deleted directly by hovering on a specific data field.

4.3 NLP NER dataset preparation

An early dataset with 100 events is created and annotated manually, but results of the custom trained model is unsatisfactory and unstable due to its limited dataset used for training. Thus, in order to create a dataset that is with sufficient sizing, consistent, and controlled diversity between frequency of rare patterns and entity types balance, another dataset of 1000 events containing entities `event_name`, `date_range`, `time_range`, `loc`, and `freq` is generated dynamically using Python. The dataset generation's code uses a template-based approach to cover realistic, human-like event phrases. The core script includes main components like templates, entity pools, variant and template manager, and validation and leaning.

```

templates = [
    "{event} at {location} on {date} at {time}",
    "{event} {time} on {date} at {location}",
    "{event} every {frequency} at {time} in {location}",
    "{event} {date} {time} at {location}",
    "{event} at {time} {date} in {location}",
    "{event} from {date_range} at {time} in {location}",
    "{event} {time} {frequency} at {location}"
]

```

Figure 4.3 Template used in the event generation code.

The template consisting of a set of 10 unique sentence templates is shown in Figure 4.3 above. The sentences are structured in a formal way with proper grammar.

Entities such as event_name ({event} in template) and loc ({location} in template) are loaded from external lists event_name.txt and loc.txt. The items in the text files are sourced from the early dataset's event_name and loc's entity values to add to the variety of the event name and location name. Date_range generation includes only formal format (e.g., Jan 4, 23/04/2025). Templates included both {date} and {date_range} placeholders to add variety in sentence structure and make the generated text sound more natural. Time_range ({time} in template) included 12- and 24-hour formats and durations like for '1 hour 30 mins'. The freq entity ({frequency} in template) covered periodicity such as every Tuesday and every weekday.

Entity diversity is also managed to ensure even usage across different formatting styles, like for dates there are formats like slash format (23/04/2024) and month format (Jan 4). The template usage is also tracked with a template manager to avoid repetition and ensure a balanced sentence structure distribution. Lastly, the generated examples are validated to ensure that the entity span positions are correct, no overlapping annotations, and no duplicated sentences.

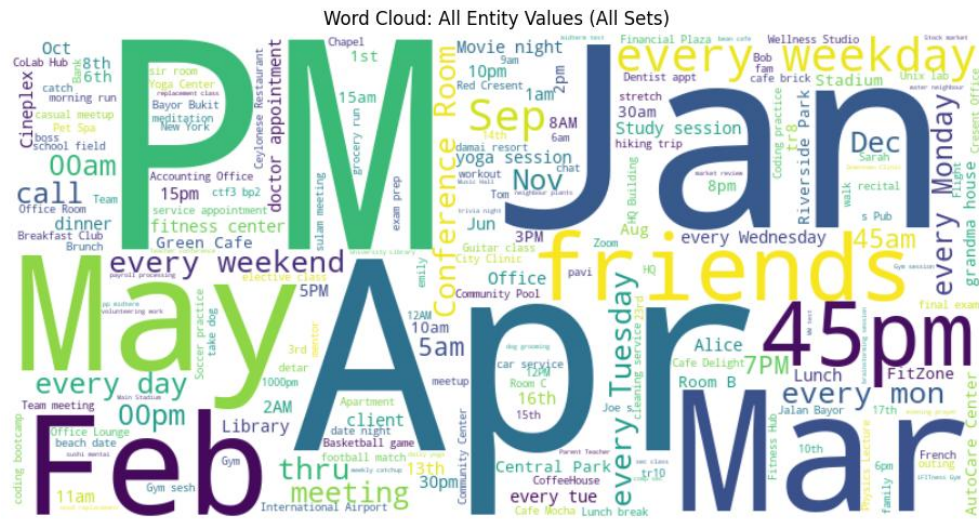


Figure 4.4 Word cloud of the dataset.

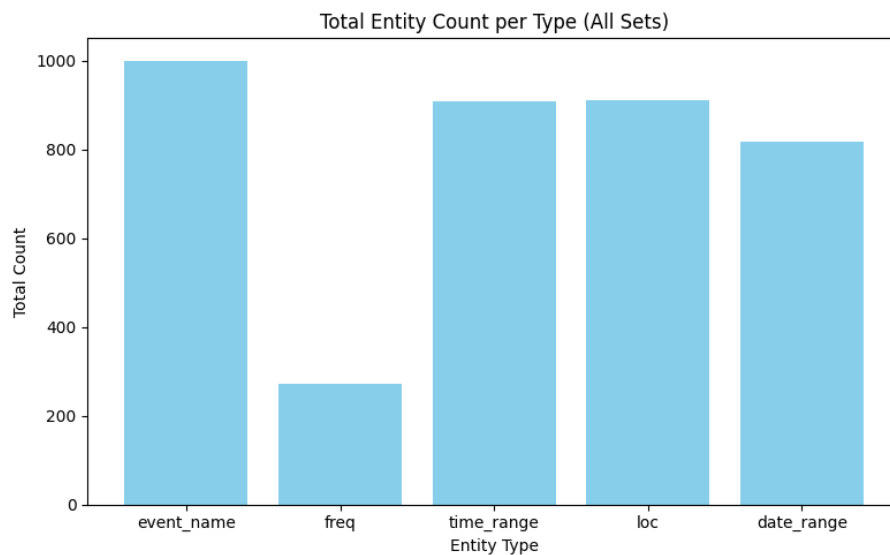


Figure 4.5 Entity distribution in the dataset.

The word cloud and entity distribution visualization are shown in Figure 4.4 and Figure 4.5. According to Figure 4.4, the word cloud shows high-frequency terms like months (e.g., Jan, Feb, May) and time-related expressions like PM and 5PM, indicating strong coverage of time and date entities in the dataset. Event-related terms such as meeting and friends also appear frequently. However, recurrence expressions like every Tuesday or every weekday appear smaller and less

frequent, reflecting the low representation of the freq entity. In Figure 4.5, event_name has the highest count of 1000, as all event items must have an event name. The freq entity has the lowest count as most events are of single occurrence.

4.4 Custom NER Model

4.4.1 Data annotation

The final dataset consisted of 1000 event sentences annotated with five custom entity types: event_name, loc, date_range, time_range, and freq. The annotations were generated programmatically using a custom Python script that assigned entity spans based on known insertion points from sentence templates. This automated method ensured consistency, reduced human error, and allowed for scalable dataset expansion. The annotated examples were then split into training (80%), validation (10%), and testing (10%) subsets and exported in the .spacy binary format for use with spaCy's training pipeline.

4.4.2 Custom NER Model Training and Validation

```
python -m spacy train config.cfg --output ./output --  
paths.train ./training/train.spacy --paths.dev ./training/val.spacy
```

Figure 4.6 Model training and validation command.

Model training and validation is done using the command in Figure 4.X, where it runs spaCy's training module using Python. It accepts a configuration file, a training file and validation file in spacy format, and produces an output folder. A config.cfg file defines the NLP pipeline, model architecture, optimizer settings, and training hyperparameters, the default config.cfg file is used as the baseline for the model performance. The output folder's contents are shown in Figure 4.X below.

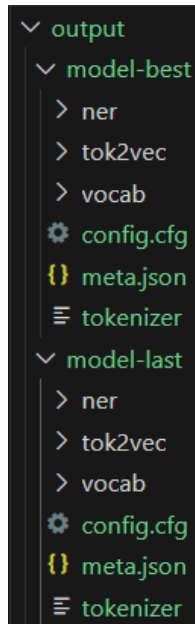


Figure 4.7 The output folder's contents.

```

===== Training pipeline =====
i Pipeline: ['tok2vec', 'ner']
i Initial learn rate: 0.001

```

E	#	LOSS TOK2VEC	LOSS NER	ENTS_F	ENTS_P	ENTS_R	SCORE
0	0	0.00	73.95	8.57	8.62	8.53	0.09
2	200	383.88	4984.34	96.12	96.12	96.12	0.96
5	400	147.45	354.82	98.97	98.97	98.97	0.99
8	600	94.66	158.34	98.19	98.19	98.19	0.98
12	800	136.60	57.69	99.22	99.22	99.22	0.99
17	1000	40.83	24.40	98.97	98.97	98.97	0.99
24	1200	108.35	65.26	99.48	99.48	99.48	0.99
31	1400	65.39	27.06	98.71	98.71	98.71	0.99
41	1600	291.82	82.61	98.97	98.97	98.97	0.99
53	1800	152.97	60.20	98.97	98.97	98.97	0.99
67	2000	120.62	37.57	99.48	99.48	99.48	0.99
84	2200	0.00	0.00	99.22	99.22	99.22	0.99
104	2400	61.61	23.74	98.97	98.97	98.97	0.99
124	2600	242.22	62.92	98.97	98.97	98.97	0.99
144	2800	50.05	9.10	99.48	99.48	99.48	0.99

```

✓ Saved pipeline to output directory
output\model-last

```

Figure 4.8 The output of the training and validation command.

After running the command in Figure 4.7, spacy will print a training log that summarizes the model's learning progress at regular intervals, which is shown in Figure 4.8. The model starts off with poor performance as it was just starting off. By step 200, F-score (ENTS_F) jumps to 96.12%, and the performance stabilizes around 98.97% f-score by step 400. The loss values (LOSS NER) also decrease sharply, which shows that the model is learning and optimizing. On later

epochs, the loss values continue to decrease and the F-score, precision (ENTS_P), and recall (ENTS_R) all stay near 98.97% - 99.48%. The training ends on epoch 144 with F-score, precision, and recall at 99.48%.

4.4.3 Custom NER Model Testing

```
python -m spacy evaluate output/model-last ./training/test.spacy
```

Figure 4.9 Model testing and evaluation command.

```
===== Results =====
TOK      100.00
NER P    99.74
NER R    99.74
NER F    99.74
SPEED    12212

===== NER (per type) =====

```

	P	R	F
event_name	100.00	100.00	100.00
freq	100.00	100.00	100.00
time_range	100.00	100.00	100.00
loc	100.00	100.00	100.00
date_range	98.84	98.84	98.84

Figure 4.10 The output of the testing and evaluation command.

Model evaluation and testing is carried out using the command in Figure 4.9, it evaluates the performance of the output model by using the testing data stored in spacy format to check how well the model performs on unseen data. After running the command, it prints the overall and per-entity-type metrics, which is shown in Figure 4.10. According to the screenshot, the baseline model performs very well, with precision (NER P), recall (NER R), and F1 (NER F) score of 99.74%, perfect tokenization accuracy (TOK), with the speed of 12212 tokens per second. As for the baseline model's per-entity-type metrics, it is able to perfectly identify event_name, freq, time_range, and loc, with slightly lower performance (98.84%) on date_range.

4.4.4 Issues with Baseline Custom NER Model and Hyperparameter Tunning Attempts

To evaluate the generalization and stability of the custom NER model beyond the initial training and test sets, an additional manually crafted dataset of 10 event sentences was used. These examples were not part of the training data and were selected to test both general recognition (first six examples) and the model's consistency when faced with minor variations in input structure (last four examples). The 10 extra testing examples are shown in Figure 4.11 below.

```
realworld_texts = [  
    "double check chap4 10/5 3pm-4pm",  
    "outing with friends at cafe brick at 2pm for 30 minutes on 31/10/2024",  
    "meeting at office on 01/11 at 3pm for 1 hour",  
    "kor class at 8:30 2 hours at tuition center",  
    "pp midterm test on 12nov at CS Lab at 10am to 1pm",  
    "math test on 16/5/2025 on 11am to 12pm at stadium",  
  
    "lunch date at alfresco at Feb 2 at 11pm",  
    "lunch date at alfresco Feb 11 11pm",  
    "lunch date at alfresco on 11Feb at 11pm",  
    "lunch date at alfresco on 11 Feb at 11pm",  
]
```

Figure 4.11 Extra testing data that is manually created.

Each prediction is manually reviewed and compared against expected entity recognition for event_name, loc, date_range, time_range, and freq. The wrongly labelled entities and its values are documented in Table 4.1 below.

#	Expected Entity	Model Output	Error
1	event_name: "double check chap4" date_range: "5/10" time_range: "3pm-4pm"	event_name: "double check chap4" time_range: "3pm-4pm"	Missed "5/10" as date_range.
2	event_name: " outing with friends" loc: "cafe brick" date_range: "2pm for 30 minutes" time_range: " 31/10/2024"	event_name: " outing with friends" loc: "cafe brick" time_range: "2pm for 30 minutes", "31/10/2024"	"2pm for 30 minutes" wrongly labeled as time_range
3	event_name: "meeting" loc: "office"	event_name: "meeting" loc: "office"	"01/11" mislabeled as time_range

	date_range: "01/11" time_range: "3pm for 1 hour"	time_range: "01/11", "3pm for 1 hour"	instead of date_range
4	event_name: "kor class" time_range: "8:30", "2 hours" loc: "tuition center"	event_name: "kor class" time_range: "8:30" date_range: "2 hours" loc: "tuition center"	"2 hours" mislabeled as date_range instead of time_range
5	event_name: "pp midterm test" date_range: "12nov" loc: "CS Lab" time_range: "10am to 1pm"	event_name: "pp midterm test" loc: "CS Lab" time_range: "12nov", "10am to 1pm"	"12nov" mislabeled as time_range instead of date_range
6	event_name: "math test" date_range: "16/5/2025" time_range: "11am to 12pm" loc: "stadium"	event_name: "math test" date_range: "16/5/2025" time_range: "11am to 12pm" loc: "stadium"	No error
7	event_name: "lunch date" loc: "alfresco" date_range: "Feb 2" time_range: "11pm"	event_name: "lunch date" loc: "alfresco" date_range: "Feb 2" time_range: "11pm"	No error
8	event_name: "lunch date" loc: "alfresco" date_range: "Feb 11" time_range: "11pm"	event_name: "lunch date" loc: "alfresco" date_range: "Feb 11" time_range: "11pm"	No error
9	event_name: "lunch date" loc: "alfresco" date_range: "11Feb" time_range: "11pm"	event_name: "lunch date" loc: "alfresco" time_range: "11Feb", "11pm"	"11Feb" mislabeled as time_range instead of date_range
10	event_name: "lunch date" loc: "alfresco" date_range: "11 Feb" time_range: "11pm"	event_name: "lunch date" loc: "alfresco" date_range: "11 Feb" time_range: "11pm"	No error

Table 4.1 Extra testing on baseline model's performance with manually created testing data.

According to the errors in Table 4.1, the baseline model has trouble identifying entities when the event item does not have stop words like 'at' or 'on' as shown in the first examples' testing result. Besides, it also sometimes misclassifies dates with slashes like 5/10 as time values. While it is sensitive to format variants like '11 Feb' and '11Feb', for the last four examples, it manages to get a three out of four correct, which shows that the model is quite consistent.

To address the issues in the baseline model, hyperparameter tuning was carried out. The edited parameter, training, validation, and testing results, and the results of testing on extra testing data is tabulated in Table 4.2 below:

Hyperparameter change	Training, Validation, and Testing Results	Results of testing on extra test data compared to baseline model
Dropout 0.1 -> 0.2	<pre> ===== Training pipeline ===== i Pipeline: ['tok2vec', 'ner'] i Initial learn rate: 0.001 E # LOSS TOK2VEC LOSS NER ENTS_F ENTS_P ENTS_R SCORE ----- 0 0 0.00 73.95 9.10 8.69 9.56 0.09 2 200 347.17 5564.44 96.36 96.87 95.87 0.96 5 400 365.19 780.11 99.22 99.22 99.22 0.99 8 600 142.10 283.83 98.06 98.19 97.93 0.98 12 800 285.65 215.79 98.19 98.19 98.19 0.98 17 1000 146.05 122.14 98.58 98.70 98.45 0.99 24 1200 166.67 90.23 98.97 98.97 98.97 0.99 31 1400 171.27 76.67 99.22 99.22 99.22 0.99 41 1600 309.01 86.31 99.48 99.48 99.48 0.99 53 1800 308.70 101.44 98.58 98.70 98.45 0.99 67 2000 111.41 32.22 99.48 99.48 99.48 0.99 84 2200 124.99 31.47 99.61 99.74 99.48 1.00 104 2400 312.51 59.75 98.84 98.96 98.71 0.99 124 2600 62.74 8.16 99.22 99.22 99.22 0.99 144 2800 134.67 24.37 99.22 99.22 99.22 0.99 164 3000 169.26 32.28 99.22 99.22 99.22 0.99 184 3200 511.03 74.04 99.22 99.22 99.22 0.99 204 3400 227.55 43.49 98.45 98.45 98.45 0.98 224 3600 82.17 12.33 98.97 98.97 98.97 0.99 244 3800 152.22 20.53 99.74 99.74 99.74 1.00 264 4000 12.31 1.36 99.22 99.22 99.22 0.99 284 4200 215.83 25.79 98.97 98.97 98.97 0.99 304 4400 258.03 35.66 99.22 99.22 99.22 0.99 324 4600 125.63 17.31 99.48 99.48 99.48 0.99 344 4800 280.58 27.99 98.45 98.45 98.45 0.98 364 5000 422.43 42.65 99.22 99.22 99.22 0.99 384 5200 265.19 42.34 99.48 99.48 99.48 0.99 404 5400 60.91 5.95 99.74 99.74 99.74 1.00 ✓ Saved pipeline to output directory output_dropout\model-last </pre>	Performs worse with similar errors. It gets more confused between entities time_range and date_range.

	<pre> ===== Results ===== TOK 100.00 NER P 99.48 NER R 99.48 NER F 99.48 SPEED 18367 ===== NER (per type) ===== P R F event_name 100.00 100.00 100.00 freq 100.00 100.00 100.00 time_range 100.00 100.00 100.00 loc 100.00 100.00 100.00 date_range 97.67 97.67 97.67 </pre>	
Hidden_width 64 -> 128	<pre> ===== Training pipeline ===== i Pipeline: ['tok2vec', 'ner'] i Initial learn rate: 0.001 E # LOSS TOK2VEC LOSS NER ENTS_F ENTS_P ENTS_R SCORE ----- 0 0 0.00 73.95 10.25 8.87 12.14 0.10 2 200 460.64 4042.37 96.50 96.88 96.12 0.96 5 400 310.91 389.83 98.97 98.97 98.97 0.99 8 600 311.69 269.74 99.48 99.48 99.48 0.99 12 800 150.16 80.39 99.22 99.22 99.22 0.99 17 1000 216.95 93.21 99.22 99.22 99.22 0.99 24 1200 206.21 63.26 98.45 98.45 98.45 0.98 31 1400 383.57 70.69 98.97 98.97 98.97 0.99 41 1600 511.37 91.07 98.84 98.96 98.71 0.99 53 1800 39.92 6.96 97.93 97.93 97.93 0.98 67 2000 165.50 25.87 98.45 98.45 98.45 0.98 84 2200 99.64 17.79 99.22 99.22 99.22 0.99 ✓ Saved pipeline to output directory output_hiddenwidth\model1-last ===== Results ===== TOK 100.00 NER P 99.74 NER R 99.74 NER F 99.74 SPEED 17851 ===== NER (per type) ===== P R F event_name 100.00 100.00 100.00 freq 100.00 100.00 100.00 time_range 100.00 100.00 100.00 loc 100.00 100.00 100.00 date_range 98.84 98.84 98.84 </pre>	Successfully labelled “8:30 2 hours” as time_range, but still retains same errors as mislabeling of date_range as time_range.

Learn_rate 0.001 -> 0.0005	<pre> ===== Training pipeline ===== i Pipeline: ['tok2vec', 'ner'] i Initial learn rate: 0.0005 E # LOSS TOK2VEC LOSS NER ENTS_F ENTS_P ENTS_R SCORE ----- 0 0 0.00 73.95 8.57 8.62 8.53 0.09 2 200 214.56 6928.25 93.39 93.75 93.02 0.93 5 400 190.28 631.35 98.45 98.45 98.45 0.98 8 600 79.52 195.67 98.71 98.71 98.71 0.99 12 800 55.48 71.34 98.19 98.19 98.19 0.98 17 1000 52.50 48.83 98.97 98.97 98.97 0.99 24 1200 50.51 36.21 98.97 98.97 98.97 0.99 31 1400 19.27 10.50 98.45 98.45 98.45 0.98 41 1600 20.75 8.75 99.22 99.22 99.22 0.99 53 1800 10.57 3.64 98.71 98.71 98.71 0.99 67 2000 54.49 16.70 99.22 99.22 99.22 0.99 84 2200 113.15 27.99 98.97 98.97 98.97 0.99 104 2400 20.23 7.43 98.45 98.45 98.45 0.98 124 2600 157.74 38.04 98.97 98.97 98.97 0.99 144 2800 108.23 24.34 99.22 99.22 99.22 0.99 164 3000 0.24 0.05 99.22 99.22 99.22 0.99 184 3200 8.59 3.02 99.22 99.22 99.22 0.99 ✓ Saved pipeline to output directory output learnrate\model-last ===== Results ===== TOK 100.00 NER P 99.22 NER R 99.22 NER F 99.22 SPEED 19390 ===== NER (per type) ===== P R F event_name 100.00 100.00 100.00 freq 100.00 100.00 100.00 time_range 100.00 100.00 100.00 loc 100.00 100.00 100.00 date_range 96.51 96.51 96.51 </pre>	Performs slightly worse, model now incorrectly labels event_name sometimes, and retains same errors as mislabeling of date_range as time_range.
Patience 1600 -> 500	<pre> ===== Training pipeline ===== i Pipeline: ['tok2vec', 'ner'] i Initial learn rate: 0.001 E # LOSS TOK2VEC LOSS NER ENTS_F ENTS_P ENTS_R SCORE ----- 0 0 0.00 73.95 8.57 8.62 8.53 0.09 2 200 383.88 4984.34 96.12 96.12 96.12 0.96 5 400 147.45 354.82 98.97 98.97 98.97 0.99 8 600 94.66 158.34 98.19 98.19 98.19 0.98 12 800 136.60 57.69 99.22 99.22 99.22 0.99 17 1000 40.83 24.40 98.97 98.97 98.97 0.99 24 1200 108.35 65.26 99.48 99.48 99.48 0.99 31 1400 65.39 27.06 98.71 98.71 98.71 0.99 41 1600 291.82 82.61 98.97 98.97 98.97 0.99 ✓ Saved pipeline to output directory output patience\model-last </pre>	Successfully labelled “8:30 2 hours” as time_range, mislabeled “alfresco” as event_name in the example eight, and retains same errors as mislabeling of date_range as time_range.

===== Results =====			
TOK	100.00		
NER P	99.22		
NER R	99.22		
NER F	99.22		
SPEED	17512		
===== NER (per type) =====			
	P	R	F
event_name	100.00	100.00	100.00
freq	100.00	100.00	100.00
time_range	98.85	100.00	99.42
loc	100.00	100.00	100.00
date_range	97.65	96.51	97.08

Table 4.2 Results of hyperparameter tuning.

According to the results in Table 4.2, while some hyperparameter tuning led to minor improvements, such as correctly labeling compound time expressions like “8:30 2 hours”, none of the adjustments fully resolved the core issues observed in the baseline model. Increasing dropout and lowering learn_rate led to reduced performance and new errors, such as misclassifying event_name. Increasing hidden_width and reducing patience helped in certain edge cases but failed to consistently correct the frequent confusion between date_range and time_range. This suggests that the model errors are more likely rooted in the dataset rather than unsuitable hyperparameters.

4.4.5 Custom NER Model and Post-Processing

Post-processing will be utilized in two ways, to address the misclassifications between date_range and time_range issue, and to parse natural language date and time expressions into structured, machine-readable formats.

Due to time constraints, the issue of misclassifications between date_range and time_range will be handled in post processing using regular expressions (regex). The regex patterns will check for slashes and month names (e.g. nov, feb) in time_range entity, and duration words like hours in

date_range entity. If slashes or month names are detected in time_range, it will change the wrongly labeled time_range to date_range. Conversely, if terms like "hour" or "minutes" appear in a span labeled as date_range, suggesting a duration rather than a date, the label is reassigned as time_range. This rule-based correction improves labeling consistency for overlapping or ambiguous cases.

To further standardize extracted entity values, the dateparser Python library is integrated to convert loosely formatted, human-readable date expressions—such as “12nov”, “next Monday”, or “in 3 days”—into structured date objects. This conversion enables accurate storage, filtering, and event scheduling within the EduCalendar system, as the parsed dates can be directly used in backend logic or calendar visualizations.

4.5 Proposed System’s Web-based System’s Interface

4.5.1 Sign-up/Login page

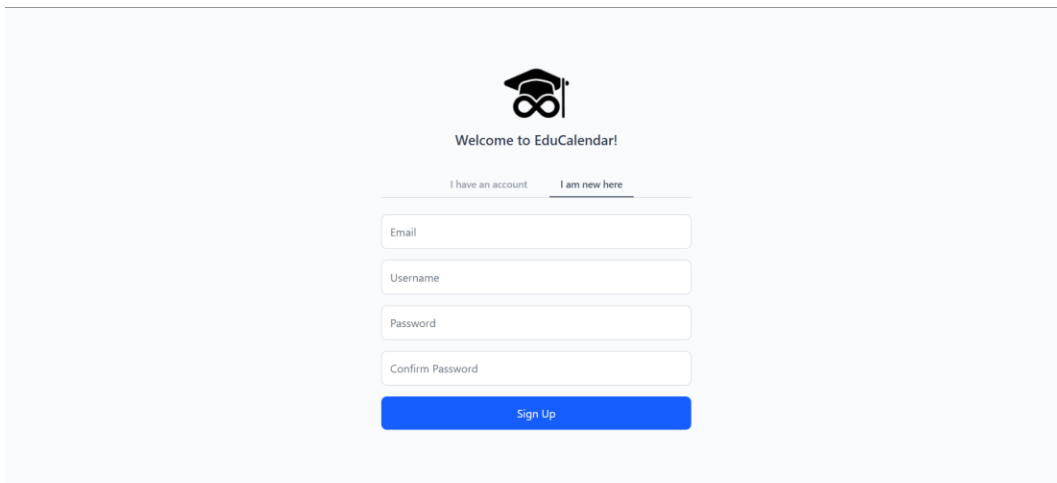
The image shows a web-based sign-up page for EduCalendar. At the top center is a logo featuring a graduation cap above an infinity symbol. Below the logo, the text "Welcome to EduCalendar!" is displayed. Underneath, there are two links: "I have an account" and "I am new here", with the latter being underlined. The main form consists of four input fields stacked vertically: "Email", "Username", "Password", and "Confirm Password". At the bottom of the form is a blue button labeled "Sign Up". The entire form is centered on a light gray background.

Figure 4.12 Sign-up page.

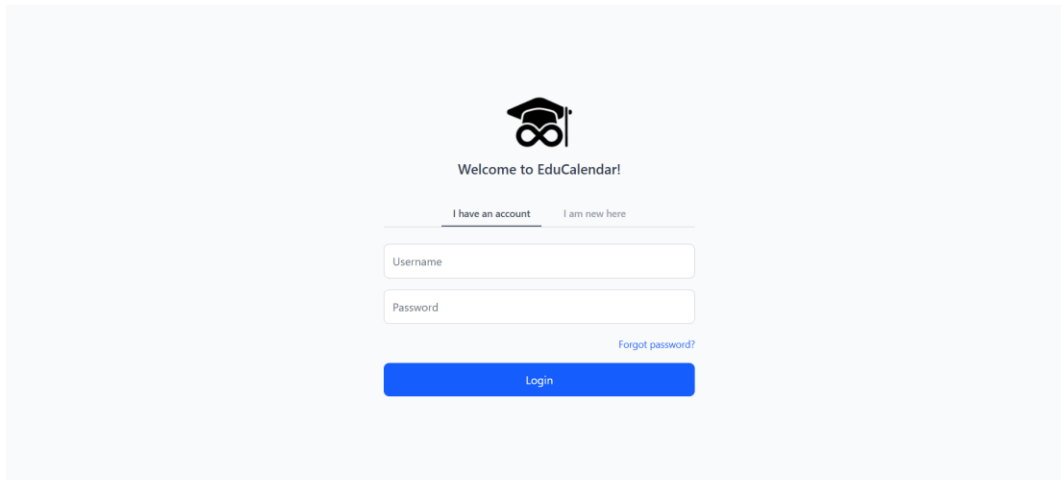


Figure 4.13 Login page.

Figure 4.12 shows the sign-up page and Figure 4.13 shows the login page. To sign-up into the web application, users will have to enter a valid email address, a unique username, and set up a password that must be longer than five characters. If the users already have an account registered prior, users can login directly with their username and password. Should the users forget their passwords, they can click on the forgot password link above the login button.

4.5.2 Event/Course creation Popup

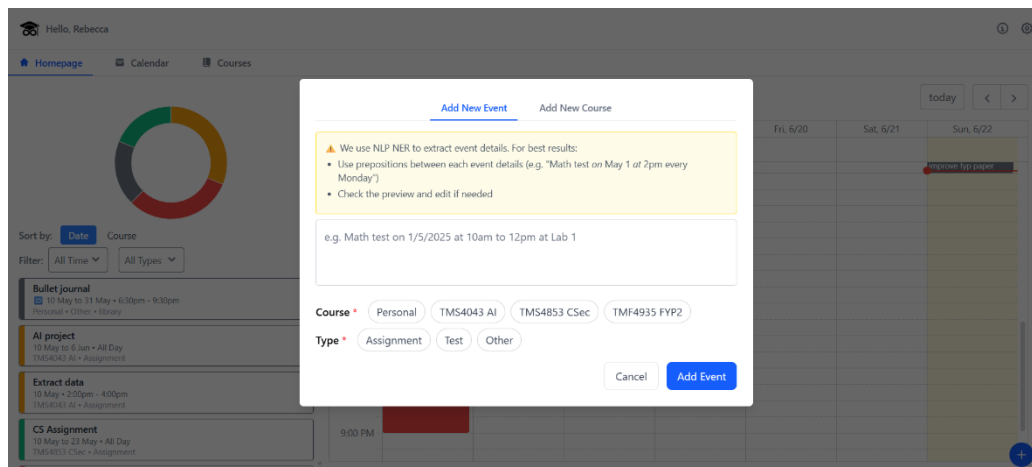


Figure 4.14 Add new event tab section of the event/course creation popup.

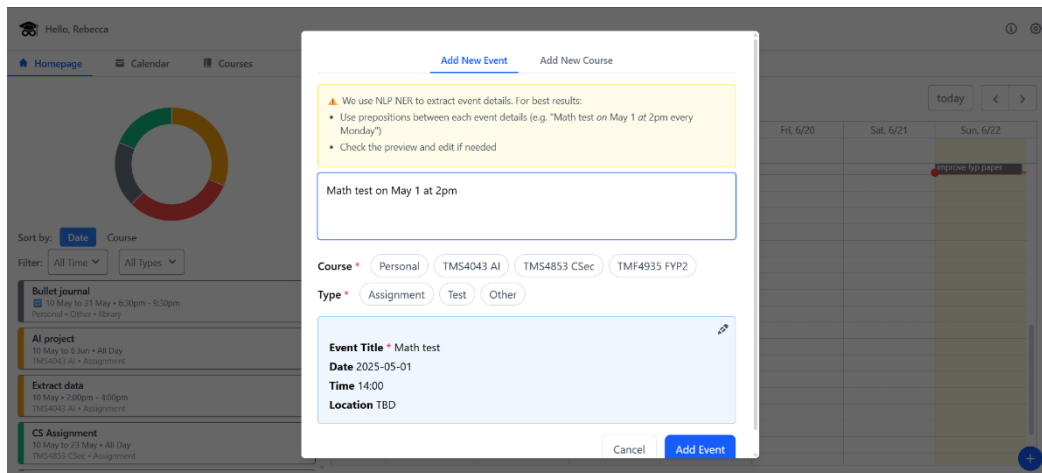


Figure 4.15 UI of event creation pop up when it receives user input popup.

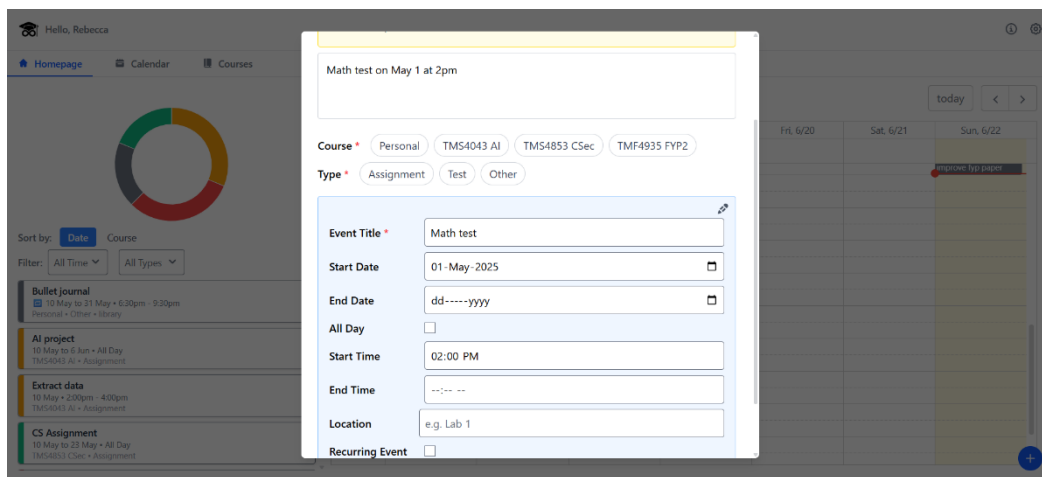


Figure 4.16 UI of event creation pop up when user click on the edit button.

Figure 4.14 shows the add new event tab section of the event/course creation. When the user just started, the course selection in the add new event tab will be empty, and they are required to add courses first before they can add any new events.

When the user wants to add new event, they will select the 'Add New Event' tab. By default, there will be a text area for the user to enter their event input, course selection for the corresponding event, and the type of the event. As the user type in their event sentence input, an event details preview will appear in a light blue box below the event type selection showing the results of the

custom NLP NER's event extraction's output, and will update as the user continue typing (Figure 4.15). Should the user find the result of the NER event detail extraction to be unsatisfactory, they can switch to manual event creation by clicking on the edit icon, which is on the event details preview box's top right. The UI of the manual event creation is shown in Figure 4.16.

The image shows a web application interface with a sidebar on the left and a main content area. The sidebar includes a user greeting 'Hello, Rebecca', navigation links for 'Homepage', 'Calendar', and 'Courses', a donut chart, and a list of tasks: 'Bullet journal' (10 May to 21 May), 'AI project' (10 May to 6 Jun), 'Extract data' (10 May to 20 Jun), and 'CS Assignment' (10 May to 29 May). The main content area displays a calendar for June 2022. Overlaid on this is a 'Add New Course' popup window. The popup has two tabs: 'Add New Event' and 'Add New Course', with the latter being active. It contains the following fields: 'Course Name' (mandatory, indicated by a red asterisk), 'Color' (with a dropdown menu), 'Instructor', 'Class Slot 1' (with a 'Day' dropdown), 'Class Slot 2' (with a 'Day' dropdown), 'Start' and 'End' time pickers for both slots, 'Location' pickers for both slots, and a 'Description' text area. At the bottom right of the popup are 'Cancel' and 'Save Course' buttons.

Figure 4.17 Add new course tab section of the event/course creation popup.

Figure 4.17 shows the add new course tab section of the event/course creation popup. The mandatory fields are course name and color; the rest can be added on post creation.

4.5.3 Homepage

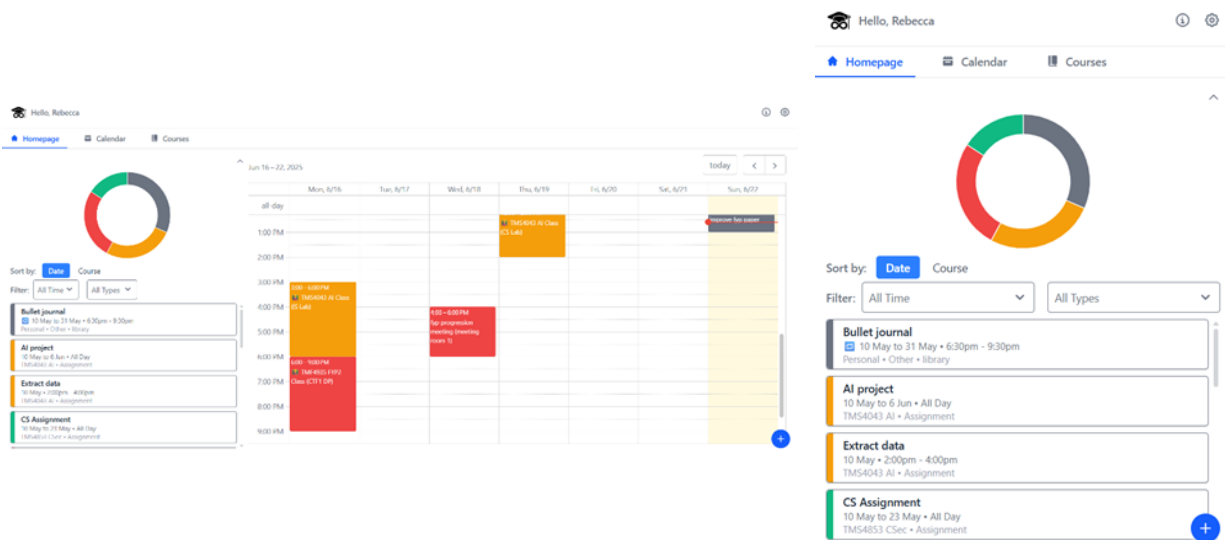


Figure 4.18 Homepage on desktop (left) and on mobile (right).

After logging in, the user will be directed to Homepage as shown in Figure 4.18. Homepage serves as the landing page for users, which shows a quick overview of the user's existing events. Desktop users will have two sections on their Homepage, the events overview on the left and a calendar view on the right. The mobile users' homepage will only display the events overview section due to the limited horizontal space on mobile devices. The events overview has two sections, the donut chart displaying the event distribution for each course, and the events list. The donut chart is collapsible, it can be hidden to leave more space for events list. The events list displays all of the user's events, and can be sorted by date or course, and filtered by time (today, the current week or month) and by the event's type (assignment, test, or other). Each event item can be viewed, edited, and deleted by pressing on the event cards, which will open a popup displaying the event details in full, as shown in Figure 4.19 below.

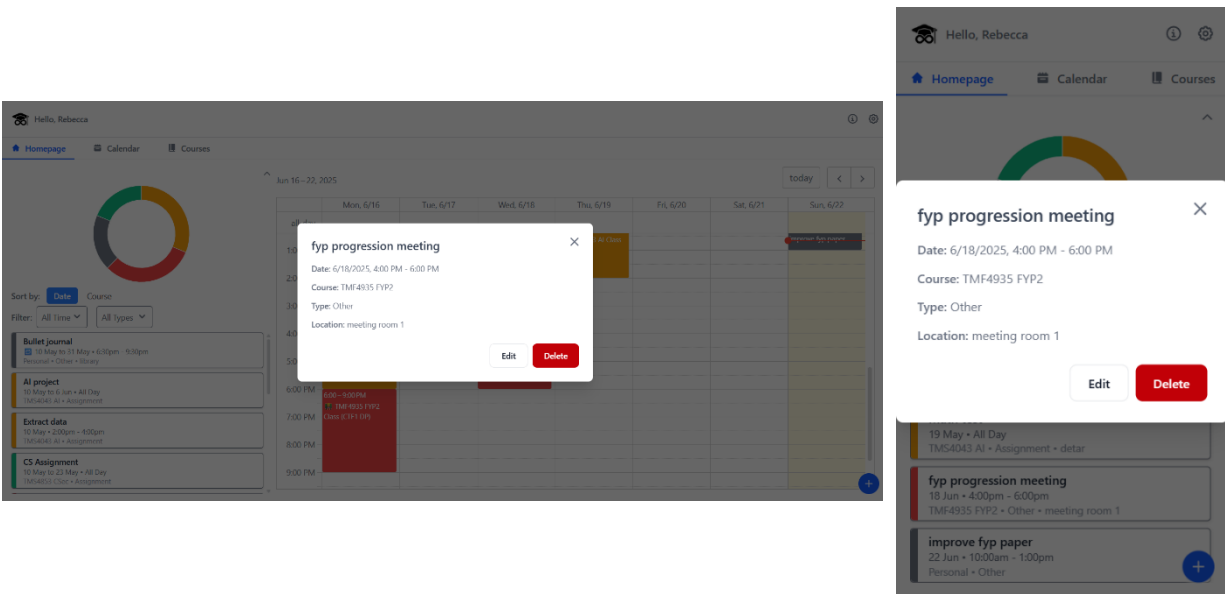


Figure 4.19 Event details popup on desktop (left) and on mobile (right).

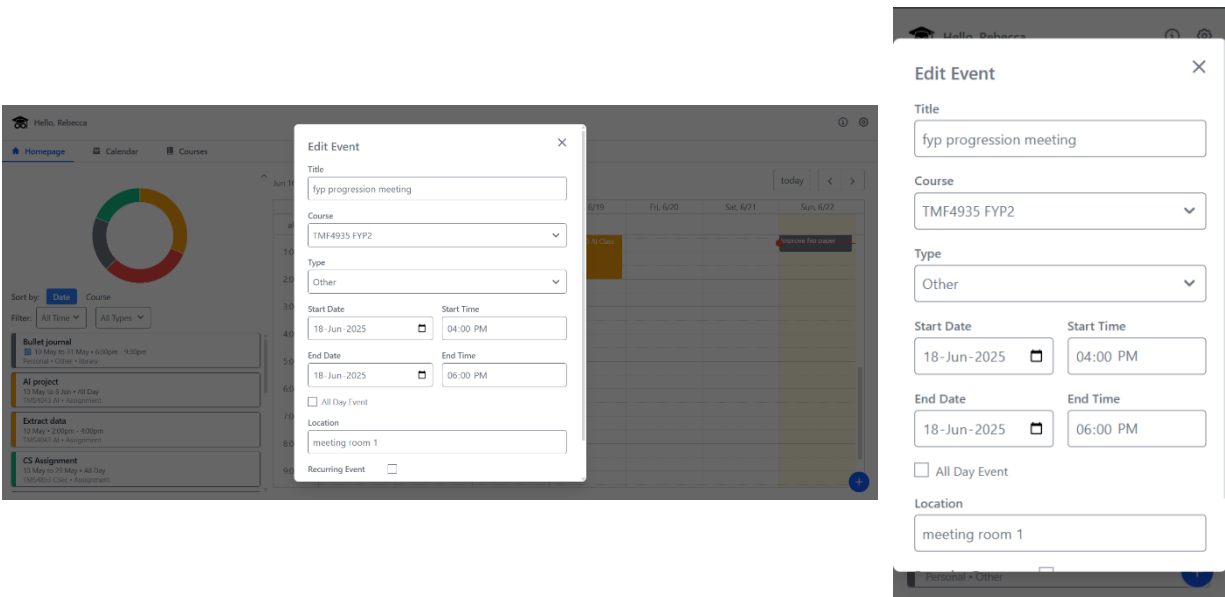


Figure 4.20 Edit event details popup on desktop (left) and on mobile (right).

The event details popup has two buttons below, edit button and delete button. The UI of the edit event details popout is shown in Figure 4.20. From here, user can choose which fields to edit. If the event chosen by the user is an event about their courses' class time, the button displayed

will be 'Edit Schedule' and 'Delete' button instead. The 'Edit Schedule' button will direct the user to edit course details when clicked. The edit course details popup will be shown and explained in the Courses Page section.

The calendar section in Homepage is in week view that starts the week with Monday. The current day is indicated by the light-yellow column, and the current time is indicated by a red line with a red circle on the line's left edge on the current day column. Users can navigate through the calendar by clicking the arrow buttons located at the top right corner to move to the next or previous week. To return to the current date, users can click the 'Today' button, which is positioned to the left of the navigation arrows. The event items on the calendar can be clicked, and will display the event details popup. It has the same functionality as the event items on overview section's event list.

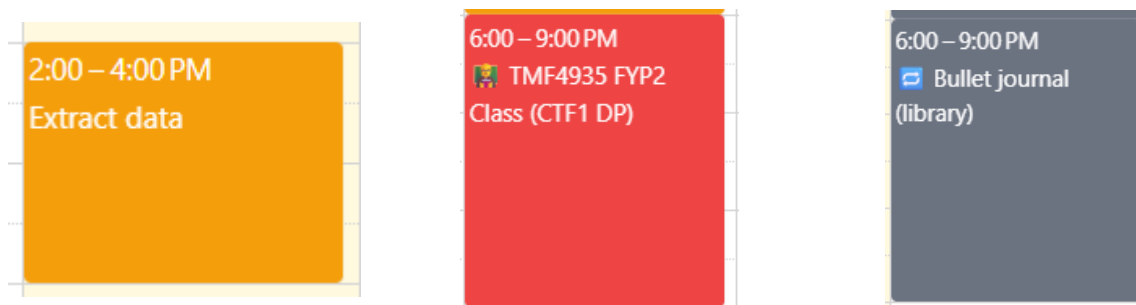


Figure 4.21 Special events indicator on the calendar section, normal event(left), class time event (middle), recurring event (right).

Figure 4.21 shows how different event looks like on the calendar section. Emojis are added on class time event and recurring event to visually differentiate between how the event data is handled. Class time event's dependency is linked to courses, meaning it is created in the 'Add new course' tab section of the Event/Course creation popup, and edited via Edit course popup. A

recurrence event is dependent on each other. For an example, when recurrence event A that repeats on every Saturday from 10 to 24 May 2024, if you click on the recurrence event item on 10 May 2025 (Saturday) to edit the event details, the rest of the recurrence event A's event items on the calendar will share the same updated event details. Class time event items are only generated on the calendar after their creation date.

4.5.4 Calendar Page

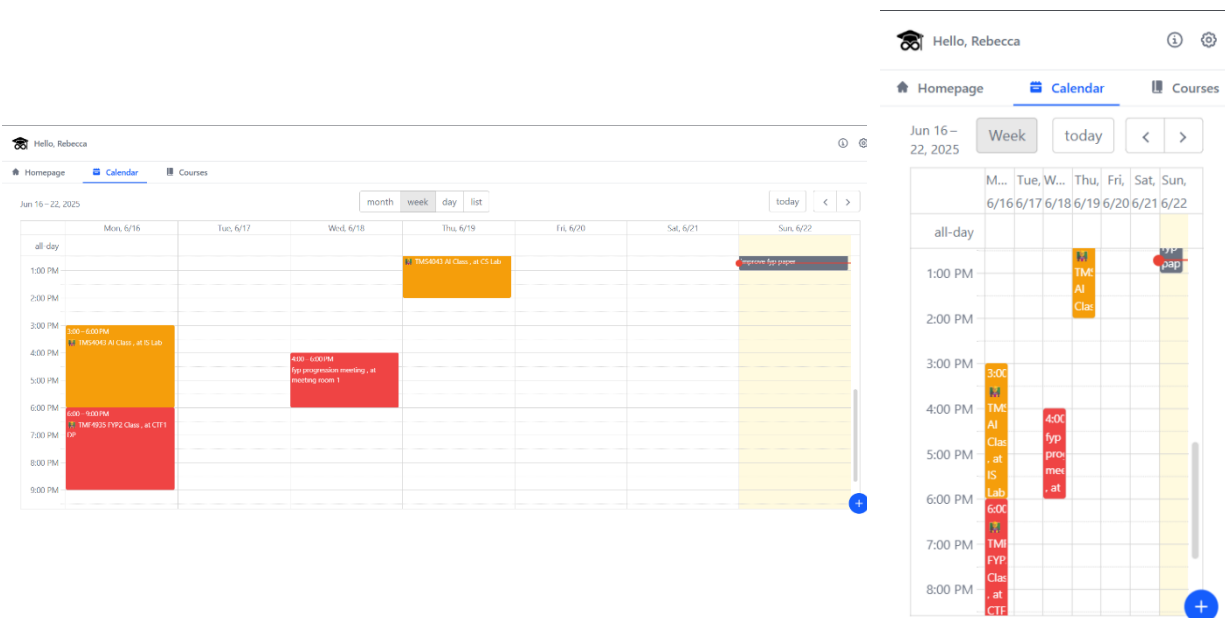


Figure 4.22 Calendar page in default week view on desktop (left) and on mobile (right).

When users click on the calendar tab, they will be directed to the Calendar page (Figure 4.22). The calendar view is built using an open-sourced calendar library [vkurko/calendar](https://github.com/vkurko/calendar) which is available on GitHub. The calendar in this page has the same functionality as Homepage's calendar section, with added view options such as week (default, Figure 4.22), month (Figure 4.23), day (Figure 4.24), and list (Figure 4.25).

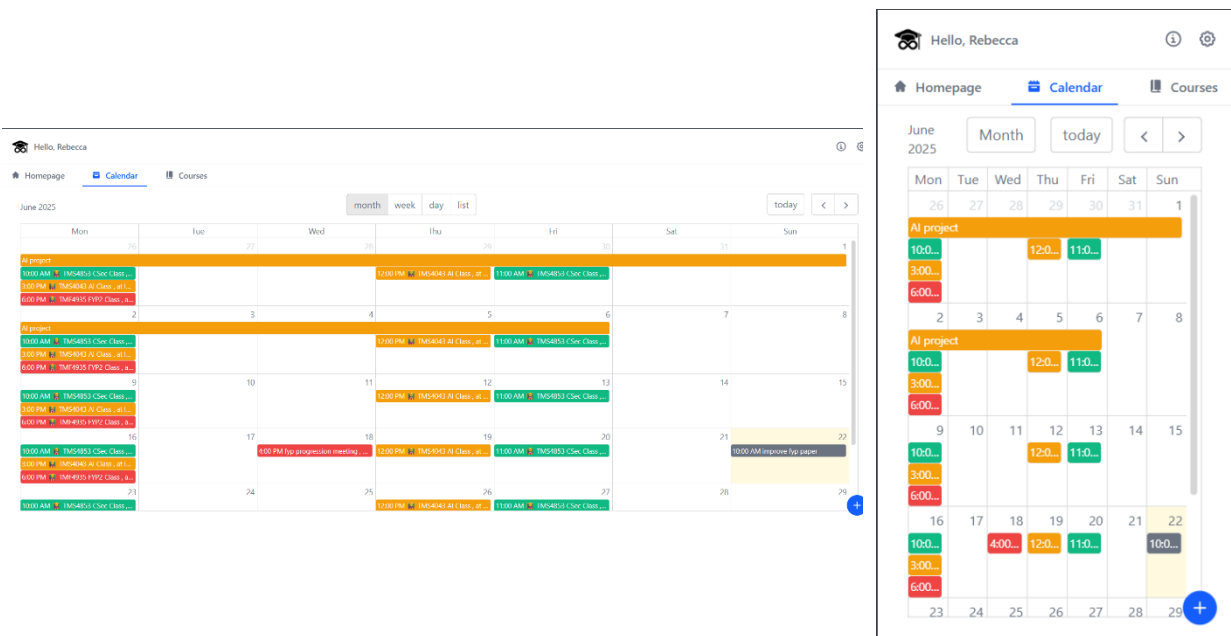


Figure 4.23 Calendar page in month view on desktop (left) and on mobile (right).

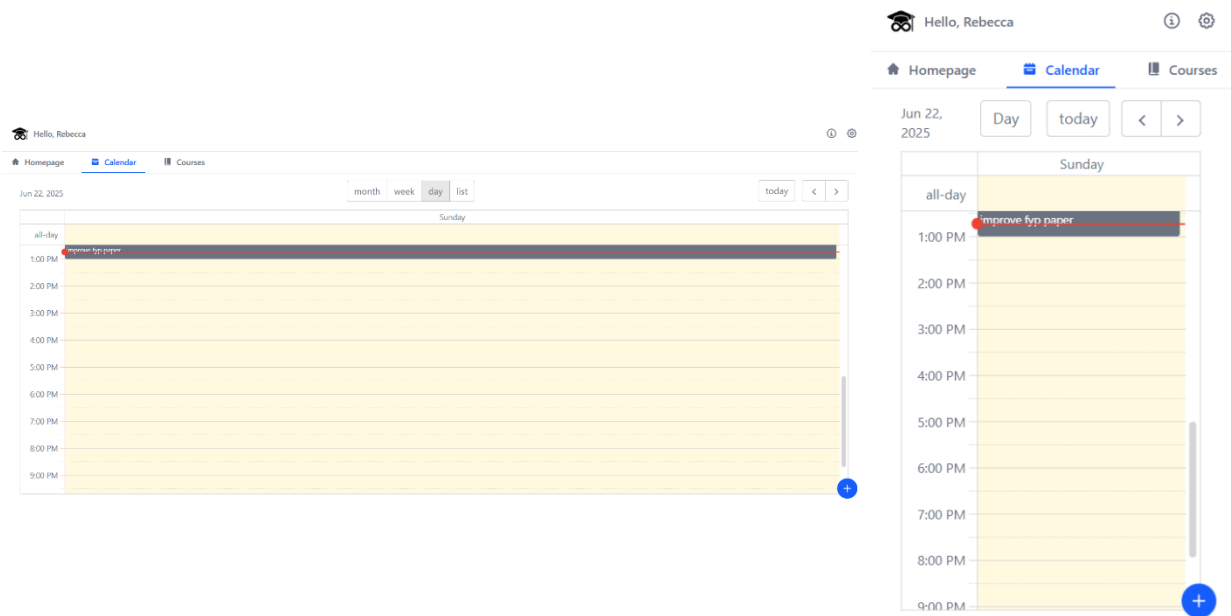


Figure 4.24 Calendar page in day view on desktop (left) and on mobile (right).

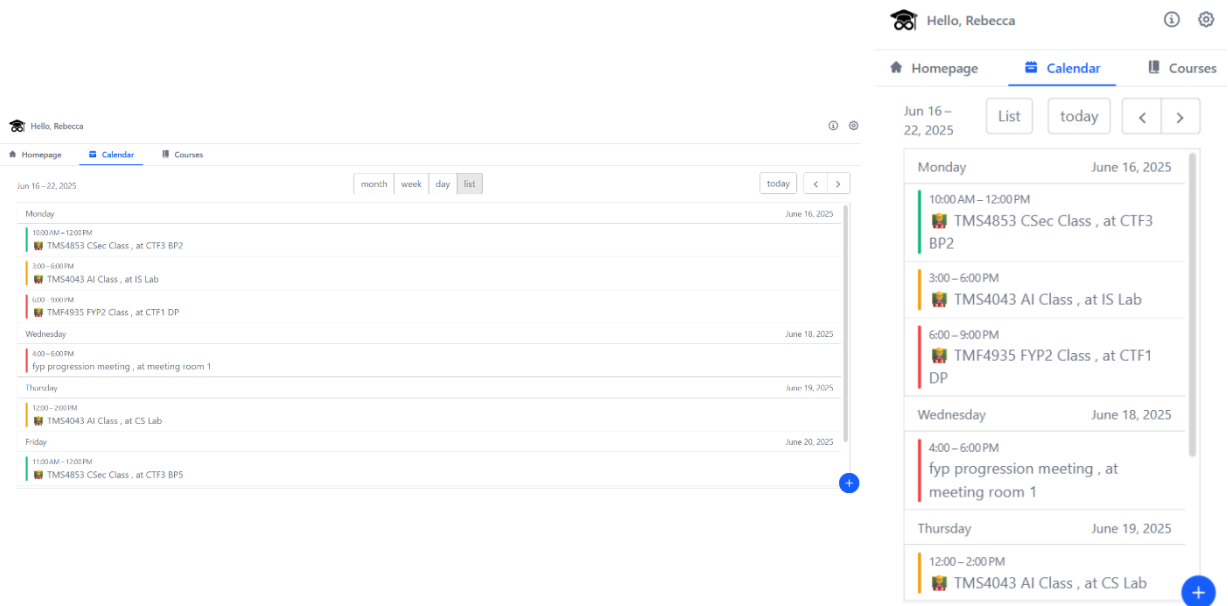


Figure 4.25 Calendar page in list view on desktop (left) and on mobile (right).

4.5.5 Courses Page

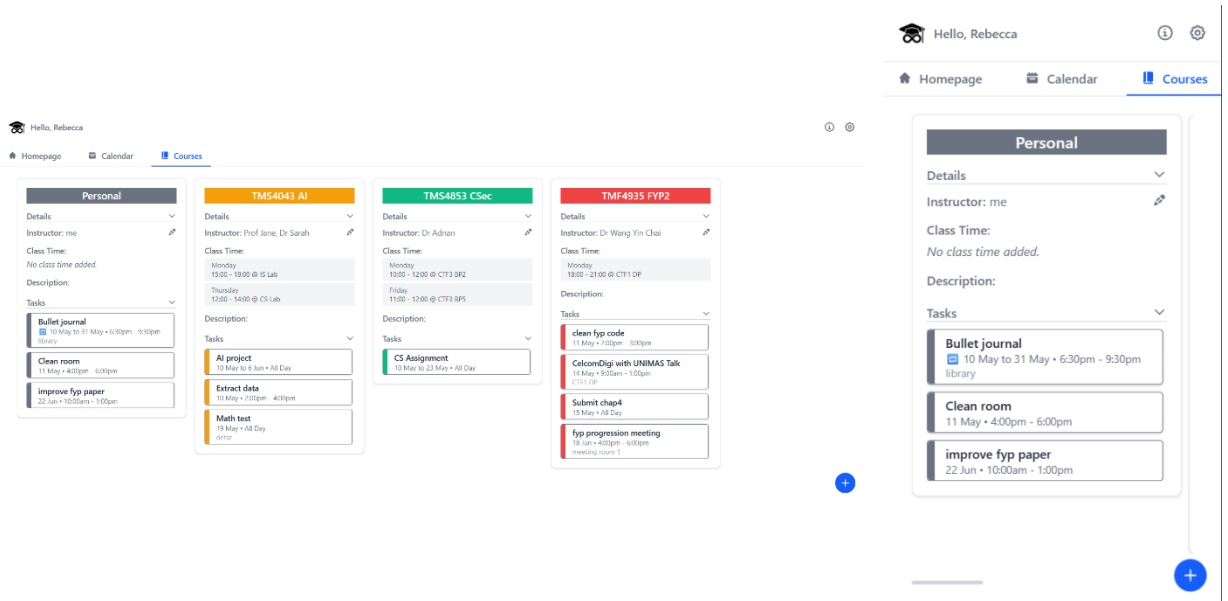


Figure 4.26 Courses page on desktop (left) and on mobile (right).

Figure 4.26 shows the Courses page, which is accessed by clicking on the courses tab. This page displays the user's courses. Due to the limited horizontal screen space on mobile devices, not all course cards are immediately visible, requiring users to swipe left or right to view additional content. Each course cards has two sections, details section and tasks section. Details section has three fields, instructor, class time, and description. To edit a course's details section, click on the pencil icon, and an edit course details popup will appear (Figure 4.27).

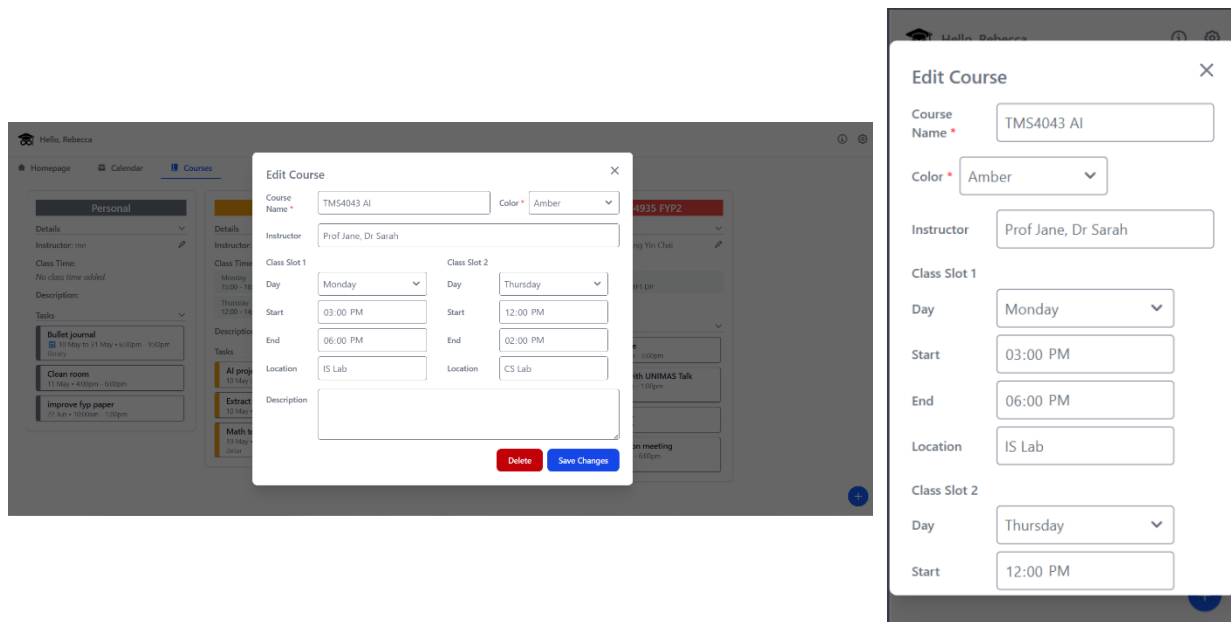


Figure 4.27 Edit course details popup on desktop (left) and on mobile (right).

4.5.6 Settings Popup

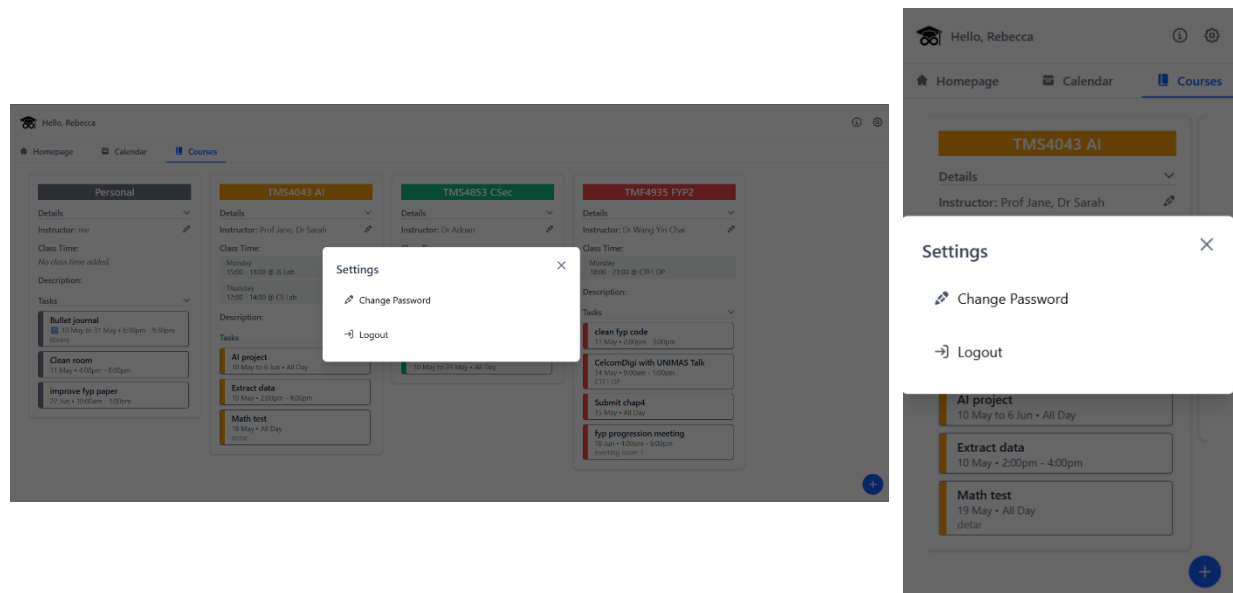


Figure 4.28 Settings popup on desktop (left) and on mobile (right).

If the user wishes to change their password or logout of their current account, click on the settings icon on the webpage's top right corner, and a settings popup will appear (Figure 4.28). The user will be directed to EduCalendar's Login/Sign-up page if they click the 'Logout' button. When the user clicks on change password, the popup will change to the UI in Figure 4.29 below. Should the user want to proceed with password change, click on the 'Send password reset email' button and Firebase will send an email to the user for them to securely change their password.

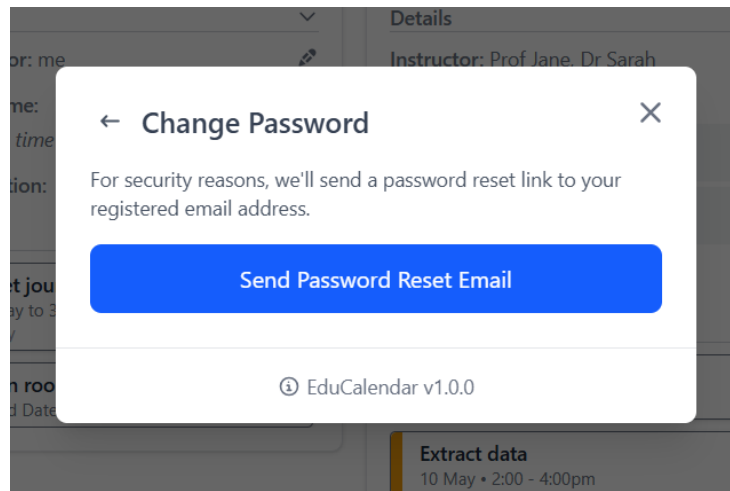


Figure 4.29 Change password UI.

4.5.7 About EduCalendar Popup

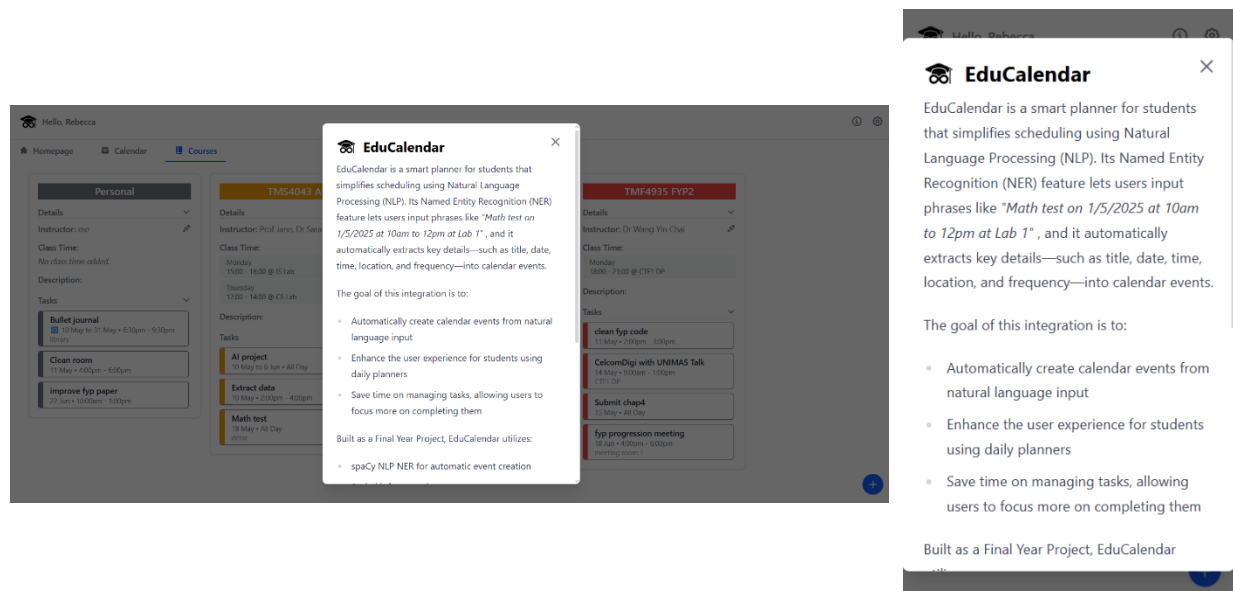


Figure 4.30 About EduCalendar popup on desktop (left) and on mobile (right).

Figure 4.30 above shows the about EduCalendar popup. This popup is accessed by clicking on the information icon next to the settings icon on the page's top right. The aim, main features, technologies used, and the contact details of the developer and the supervisor are listed on this popup for those who are interested in learning more about the EduCalendar system.

4.7 Summary

This chapter details the training of the custom NLP NER model and implementation of the digital calendar web-based application with NLP NER integration in event creation. The custom NLP NER model is trained using a dataset of 1000 event items generated dynamically using Python, which is subsequently split into 80-10-10 for training, validation, and testing. The custom model is then integrated into a web-based system built with Svelte framework and Firebase.

Chapter 5 : Testing

5.1 Introduction

This chapter outlines the testing phase of EduCalendar, the proposed digital calendar system. Testing was conducted to evaluate the system's functionality, reliability, and compliance with the specified requirements, ensuring its correct operation in real-world scenarios. Functional testing and User Acceptance Testing (UAT), was carried out to validate individual components and assess overall user satisfaction. Any bugs or issues identified during testing were either resolved immediately or scheduled for future fixes, depending on their severity and complexity.

5.2 Functional Testing

To ensure the system's reliability, functional testing is carried out to test individual features of the system. Each core functionality, such as user authentication, course management, and event handling, is evaluated to verify if it behaves as expected under normal and edge-case scenarios. This testing approach focuses on simulating real user interactions to confirm that the system responds correctly to valid inputs and gracefully handles invalid or unexpected inputs. The results of functional testing help validate that the EduCalendar system meets its functional requirements and delivers a seamless user experience.

5.2.1 Sign-Up Functionality Testing

Test Case ID	SU-001
Title	Valid User Sign-Up
Description	Verify that a user can successfully create an account using valid email, username, and matching passwords
Precondition	User is not logged in; the email and username have not been registered before
Test Steps	1. Navigate to the 'I am new here'/Sign-Up page. 2. Enter a valid email. 3. Enter a unique username.

	4. Enter a password that is at least six characters and confirm it. 5. Click the “Sign Up” button.
Test Input	Email: 81036@siswa.unimas.my Username: Rebecca Password: Rebecca81036 Confirm password: Rebecca81036
Expected Result	New account created; user directed to the ‘I have an account’/Login Page
Actual Result	New account created; user directed to the ‘I have an account’/Login Page
Status	Pass

Table 5.1 Valid User Sign-Up testing.

Test Case ID	SU-002
Title	Invalid User Sign-Up
Description	Verify that a user can cannot create an account using registered email
Precondition	User is not logged in; the email and username have been registered before
Test Steps	1. Navigate to the ‘I am new here’/Sign-Up page. 2. Enter a registered email. 3. Enter a unique username. 4. Enter a password that is at least six characters and confirm it. 5. Click the “Sign Up” button.
Test Input	Email: 81036@siswa.unimas.my Username: Rebecca2 Password: Rebecca810362 Confirm password: Rebecca810362
Expected Result	New account not created; error message ‘Firebase: Error (auth/email-already-in-use).’
Actual Result	New account not created; error message ‘Firebase: Error (auth/email-already-in-use).’
Status	Pass

Table 5.2 Invalid User Sign-Up testing.

5.2.2 Login Functionality Testing

Test Case ID	LI-001
Title	Valid User Login
Description	Ensure user can log in with correct credentials
Precondition	User is not logged in; has registered an EduCalendar account
Test Steps	1. Navigate to login page.

	2. Enter correct username and password. 3. Click 'Login'.
Test Input	Username: Rebecca Password: Rebecca81036
Expected Result	User directed to the EduCalendar's Homepage
Actual Result	User directed to the EduCalendar's Homepage
Status	Pass

Table 5.3 Valid User Login testing.

Test Case ID	LI-002
Title	Invalid User Login
Description	Ensure user cannot log in with wrong password
Precondition	User is not logged in; has registered an EduCalendar account
Test Steps	1. Navigate to login page. 2. Enter correct username and incorrect password. 3. Click 'Login'.
Test Input	Username: Rebecca Password: wrongpassword
Expected Result	Error message 'Firebase: Error (auth/invalid-credential).'
Actual Result	Error message 'Firebase: Error (auth/invalid-credential).'
Status	Pass

Table 5.4 Invalid User Login testing.

5.2.3 Course Creation Testing

Test Case ID	CC-001
Title	Successful course creation
Description	Create course with All Fields, verify successful course creation
Precondition	User is logged in
Test Steps	1. Open Add Event modal by pressing the button on the webpage's bottom left. 2. Switch to 'Add New Course' tab. 3. Fill in all fields. 4. Click 'Save Course'.
Test Input	Course name: Test course Color: Pink Instructor: Test Instructor

	Class slot 1 day: Tuesday Class slot 1 start: 9:00 AM Class slot 1 end: 10:00 AM Class slot 1 location: Lab1 Class slot 2 day: Wednesday Class slot 2 start: 9:00 AM Class slot 2 end: 11:00 AM Class slot 2 location: Lab2 Description: This course is made for testing
Expected Result	New course successfully created; a new course card is made in the Courses Tab with the correct details; new course's class time event is viewable on the calendar and appears after the course creation date.
Actual Result	New course successfully created; a new course card is made in the Courses Tab with the correct details; new course's class time event is viewable on the calendar and appears after the course creation date.
Status	Pass

Table 5.5 Successful course creation testing.

Test Case ID	CC-002
Title	Unsuccessful course creation with missing course name
Description	Create course with missing course name
Precondition	User is logged in
Test Steps	1. Open Add Event modal by pressing the button on the webpage's bottom left. 2. Switch to 'Add New Course' tab. 3. Fill in all fields except the course name field. 4. Click 'Save Course'.
Test Input	Course name: (empty) Color: Pink Instructor: Test Instructor Class slot 1 day: Tuesday Class slot 1 start: 9:00 AM Class slot 1 end: 10:00 AM Class slot 1 location: Lab1 Class slot 2 day: Wednesday Class slot 2 start: 9:00 AM Class slot 2 end: 11:00 AM Class slot 2 location: Lab2 Description: This course is made for testing
Expected Result	Error message 'Course name and color are required.'
Actual Result	Error message 'Course name and color are required.'

Status	Pass
--------	------

Table 5.6 Unsuccessful course creation with missing course name testing.

Test Case ID	CC-003
Title	Unsuccessful course creation with missing course color
Description	Create course with missing course color
Precondition	User is logged in
Test Steps	1. Open Add Event modal by pressing the button on the webpage's bottom left. 2. Switch to 'Add New Course' tab. 3. Fill in all fields except the color field. 4. Click 'Save Course'.
Test Input	Course name: Test course Color: (empty) Instructor: Test Instructor Class slot 1 day: Tuesday Class slot 1 start: 9:00 AM Class slot 1 end: 10:00 AM Class slot 1 location: Lab1 Class slot 2 day: Wednesday Class slot 2 start: 9:00 AM Class slot 2 end: 11:00 AM Class slot 2 location: Lab2 Description: This course is made for testing
Expected Result	Error message 'Course name and color are required.'
Actual Result	Error message 'Course name and color are required.'
Status	Pass

Table 5.7 Unsuccessful course creation with missing course color testing.

5.2.4 Course Details Editing Testing

Test Case ID	CDE-001
Title	Course details edit
Description	Edit course details and ensure the new course details is saved and updated
Precondition	User is logged in, course created
Test Steps	1. Open Courses Tab. 2. Click edit icon. 3. Change course details. 4. Click 'Save Changes'.

Test Input	Course name: New course name Color: Violet Instructor: New test Instructor Class slot 1 day: Tuesday Class slot 1 start: 9:30 AM Class slot 1 end: 10:30 AM Class slot 1 location: New Lab1 Class slot 2 day: Wednesday Class slot 2 start: 10:00 AM Class slot 2 end: 1:00 PM Class slot 2 location: New Lab2 Description: This course is made for testing, updated
Expected Result	Course details successfully updated; the course card's details in Courses Tab are updated with the correct details; course's class time event updated on the calendar
Actual Result	Course details successfully updated; the course card's details in Courses Tab are updated with the correct details; course's class time event updated on the calendar
Status	Pass

Table 5.8 Course details edit testing.

5.2.5 Event Creation Testing

Test Case ID	EC-001
Title	Successful course details edit
Description	Create events with course and event type selected, has event title, verify successful event creation
Precondition	User is logged in, at least one course created
Test Steps	1. Open Add Event modal by pressing the button on the webpage's bottom left. 2. Input event prompt in text area. 3. Select the course the event is associated to. 4. Select the event type the event is associated to. 5. Click 'Save Event'.
Test Input	Event prompt: Test event on 29 may at 2pm – 4pm on Webex • Extracted event title: Test event • Extracted date: 2025-05-29 • Extracted time: 14:00 to 16:00 • Extracted location: Webex Course: New course name Type: Other

Expected Result	New event created; event added on the overview list and donut chart in Homepage Tab, and the corresponding course card's tasks section in Courses Tab; event is viewable on the calendar
Actual Result	New event created; event added on the overview list and donut chart in Homepage Tab, and the corresponding course card's tasks section in Courses Tab; event is viewable on the calendar
Status	Pass

Table 5.9 Successful course details edit testing.

Test Case ID	EC-002
Title	Unsuccessful course details edit
Description	Create events without course selected, verify unsuccessful event creation
Precondition	User is logged in, at least one course created
Test Steps	1. Open Add Event modal by pressing the button on the webpage's bottom left. 2. Input event prompt in text area. 3. Select the event type the event is associated to. 4. Click 'Save Event'.
Test Input	Event prompt: Test event on 29 may at 2pm – 4pm on Webex • Extracted event title: Test event • Extracted date: 2025-05-29 • Extracted time: 14:00 to 16:00 • Extracted location: Webex Course: (empty) Type: Other
Expected Result	Error message 'Please select a course, type, and ensure the event has a title'
Actual Result	Error message 'Please select a course, type, and ensure the event has a title'
Status	Pass

Table 5.10 Unsuccessful course details edit testing.

5.2.6 Event Detail Viewing Testing

Test Case ID	EDV-001
Title	Event detail view
Description	Verify that specific event details are viewable from overview list
Precondition	User is logged in, event created
Test Steps	1. Open Homepage/Calendar/Courses Tab. 2. Click on an event card. 3. Click on close event details popup's 'X' icon.

Test Input	Click on ‘Edited test event’ event card created prior
Expected Result	The event details popup appears, the event details in the popup matches the event card that was clicked
Actual Result	The event details popup appears, the event details in the popup matches the event card that was clicked
Status	Pass

Table 5.11 Event detail view testing.

5.2.7 Event Details Editing Testing

Test Case ID	EDE-001
Title	Event details edit
Description	Edit event details and ensure the new event details is saved and updated
Precondition	User is logged in, event created
Test Steps	1. Click on an event card with details that requires editing. 2. Click edit button. 3. Update event details. 4. Click ‘Save Changes’.
Test Input	Event title: Edited test event Course: New course name Type: Assignment Start date: 28 May 2025 End date: 28 May 2025 Start date: 1:00 PM End date: 3:00 PM All day event checkbox: unchecked Location: Edited location Recurring event checkbox: unchecked
Expected Result	Event details successfully updated; the event card’s details are updated with the correct details; event details updated on the calendar
Actual Result	Event details successfully updated; the event card’s details are updated with the correct details; event details updated on the calendar
Status	Pass

Table 5.12 Event details edit testing.

5.2.8 Event Sorting/Filtering Testing

Test Case ID	ESF-001
Title	Event sorting
Description	Events sorted by Course correctly
Precondition	User is logged in, events created with different associated course
Test Steps	1. Open Homepage Tab. 2. Click on 'Course' button in sort by option.
Test Input	2 courses created with three events each
Expected Result	Events from the same courses are listed next to each other
Actual Result	Events from the same courses are listed next to each other
Status	Pass

Table 5.13 Event sorting testing.

Test Case ID	ESF-002
Title	Event filtered by date range (All time, Today, This week, This month)
Description	Events filtered by 'This week' correctly
Precondition	User is logged in, events that happens in different weeks are created
Test Steps	1. Open Homepage Tab. 2. Select 'This week' from the dropdown list in filter option.
Test Input	2 events with dates within the current week and 1 event that is not within the current week.
Expected Result	Only the 2 events with dates within the current week appears on the list
Actual Result	Only the 2 events with dates within the current week appears on the list
Status	Pass

Table 5.14 Event filtered by date range testing.

Test Case ID	ESF-003
Title	Event filtered by event types (All types, Assignments, Tests, Others)
Description	Events filtered by 'Assignment' correctly
Precondition	User is logged in, events with different event types are created
Test Steps	1. Open Homepage Tab. 2. Select 'Assignment' from the dropdown list in filter option.
Test Input	1 event with event type 'Assignment' and 1 event with another event type

Expected Result	Only the event with event type ‘Assignment’ appears on the list
Actual Result	Only the event with event type ‘Assignment’ appears on the list
Status	Pass

Table 5.15 Event filtered by event types testing.

5.2.9 Event Deletion Testing

Test Case ID	ED-001
Title	Event deletion
Description	Verify event deleted successfully
Precondition	User is logged in, event created
Test Steps	1. Click on an event card with details that requires deletion. 2. Click delete button.
Test Input	Click on ‘Edited test event’ event card created prior
Expected Result	Event deleted; does not appear on the overview list, calendar and associated course card’s task section
Actual Result	Event deleted; does not appear on the overview list, calendar and associated course card’s task section
Status	Pass

Table 5.16 Event deletion testing.

5.2.10 Course Deletion Testing

Test Case ID	CD-001
Title	Course deletion
Description	Verify course deleted successfully
Precondition	User is logged in, course created
Test Steps	1. Click on the course card that requires deletion’s edit icon. 2. Click delete button.
Test Input	Click on ‘New course name’ course card created prior
Expected Result	Course deleted; does not appear on the courses list in Courses Tab
Actual Result	Course deleted; does not appear on the courses list in Courses Tab
Status	Pass

Table 5.17 Course deletion testing.

5.2.11 Logout Functionality Testing

Test Case ID	LO-001
Title	User logout
Description	Verify user logout successfully
Precondition	User is logged in
Test Steps	1. Click on the gear icon on the webpage's top right. 2. Click logout button.
Test Input	-
Expected Result	User log out successfully; user directed to login page
Actual Result	User log out successfully; user directed to login page
Status	Pass

Table 5.18 User logout testing.

5.2.12 Forgot Password Functionality Testing

Test Case ID	FP-001
Title	Forget user password
Description	Verify forget user password functionality
Precondition	User not logged in, username registered, on the login page
Test Steps	1. Click on the 'Forgot password?' link. 2. Enter existing username. 3. Click 'Send Reset Link' button. 4. Locate email title 'Reset your password for educalendar-6e3a1' in registered username's email inbox. 5. Click on the link to reset password. 6. Input new password. 7. Click save button. 8. Go back to EduCalendar's login page. 9. Login with new password.
Test Input	New password: newpassword81036
Expected Result	'Password reset link sent' message shown after clicking Send Reset Link' button; email with title 'Reset your password for educalendar-6e3a1' is sent to the registered username's email; user successfully login with new password
Actual Result	'Password reset link sent' message shown after clicking Send Reset Link' button; email with title 'Reset your password for educalendar-6e3a1' is sent to the registered username's email; user successfully login with new password
Status	Pass

Table 5.19 Forget user password testing.

5.2.13 Change Password Functionality Testing

Test Case ID	CP-001
Title	Change user password
Description	Verify change user password functionality
Precondition	User logged in
Test Steps	1. Click on the gear icon on the webpage's top right. 2. Click 'Change Password' button. 3. Click on 'Send Password Reset Email' button. 4. Locate email title 'Reset your password for educaledar-6e3a1' in registered username's email inbox. 5. Click on the link to reset password. 6. Input new password. 7. Click save button. 8. Click logout button. 9. Login with new password.
Test Input	New password: 2newpassword81036
Expected Result	Email with title 'Reset your password for educaledar-6e3a1' is sent to the registered username's email; user successfully login with new password
Actual Result	Email with title 'Reset your password for educaledar-6e3a1' is sent to the registered username's email; user successfully login with new password
Status	Pass

Table 5.20 Change user password testing.

5.3 User Acceptance Test (UAT)

A questionnaire approach is utilized to carry out the UAT for EduCalendar. Google form is used to create a questionnaire of 28 questions and four sections, User Demographics, Website Accessibility, Tasks and System Acceptance. The questionnaire is distributed online to 14 responses who are currently an active university student. The findings of the last three sections are discussed and analyzed.

5.3.1 Website Accessibility

Are you accessing the EduCalendar website on your mobile phone or PC/laptop?

14 responses

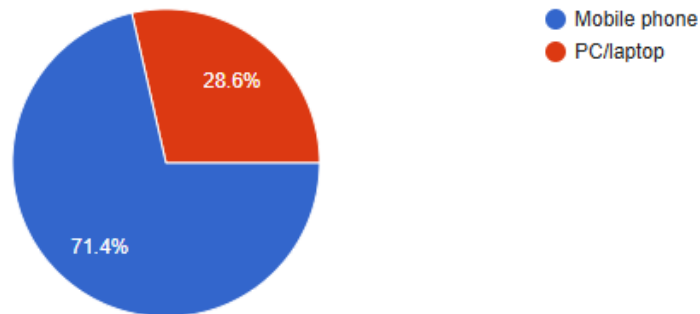


Figure 5.1 The device distribution used by the respondents.

Figure 5.1 shows that the majority of the respondents (71.4%, 10 respondents) uses their mobile phone to access EduCalendar, and only four respondents (28.6%) access the proposed system on their PC or laptop. This result suggests that most users prefer accessing scheduling tools on-the-go.

Are you able to access the EduCalendar website?

14 responses

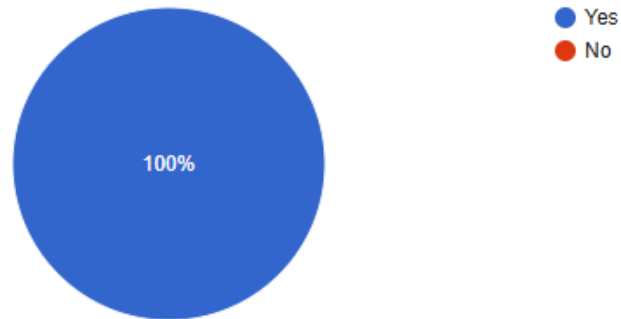


Figure 5.2 EduCalendar's website accessibility distribution of the respondents.

Figure 5.2 shows that all 14 respondents (100%) were able to access EduCalendar successfully. This indicates that the system is deployed successfully and correctly on Vercel, and is accessible on both mobile phone and PC or laptop.

Are you able to create an account?

14 responses

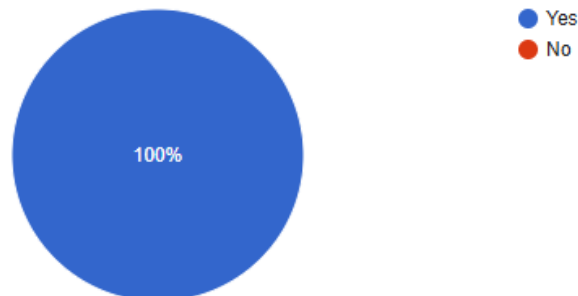


Figure 5.3 EduCalendar's account creation success distribution of the respondents.

Figure 5.3 illustrates that all 14 respondents (100%) were able to successfully create an account on EduCalendar. This indicates that the sign-up functionality worked reliably across all test users, regardless of the device or platform used.

5.3.2 (Task 1) Course Creation



Figure 5.4 EduCalendar's course creation success distribution of the respondents.

Figure 5.4 shows that all 14 respondents (100%) were able to create courses without encountering any issues. This result demonstrates that the course creation feature in EduCalendar is fully functional and user-friendly.

5.3.3 (Task 2) Event Creation

For this section, users are tasked to create five event prompts by using the NLP NER event extraction functionality of EduCalendar, which means a total of 70 event prompts has been created. For each event prompt, the user's input, extraction accuracy, and error details (if any) were collected. The responses are exported into a CSV file from Google Forms and is further processed in Excel to generate new data visualizations for a more comprehensive view. The full result is available in Appendix D.

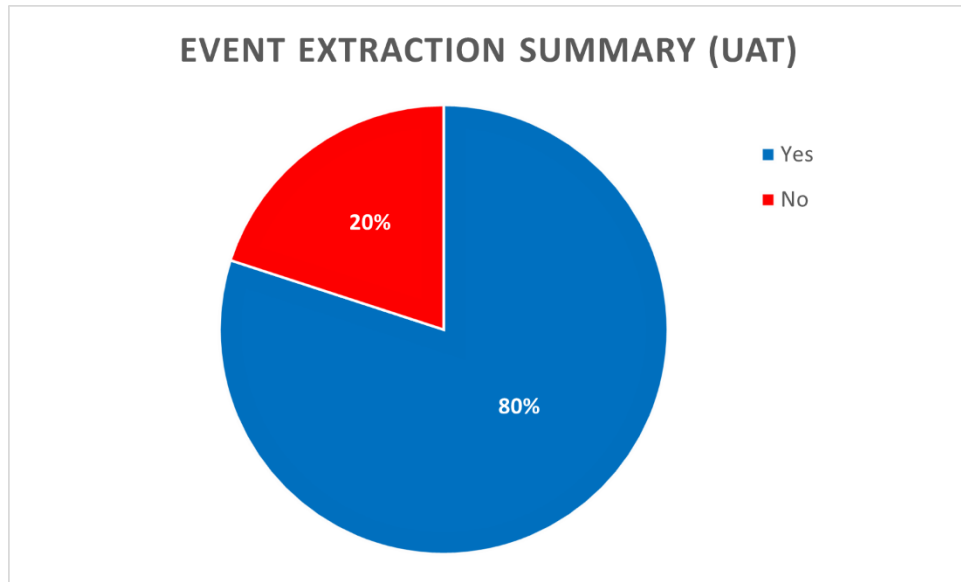


Figure 5.5 EduCalendar's event extraction success distribution of the respondents.

The pie chart in Figure 5.5 shows that over 80% of the event prompts (56 event prompts) are extracted successfully by EduCalendar's event extraction with NLP NER, while the remaining 20% (14 event prompts) failed. This result suggests that the custom NER model performs reasonably well for most cases but still encounters limitations with certain input structures. Figure 5.6 below details the list of event prompts that were not extracted correctly. The extraction errors are further analyzed in Table 5.21. The event prompt 'Quiz at dewan 115utiara at 1600 hrs' is excluded from the analysis as the event is extracted correctly upon rechecking.

Prompt	Correctly extracted?	If no, why (respondent index)	Respondent index
Assignment on June 7 at 9am to June 15 10am	No	The Ai didn't recognise and extract the event title	2
French Assessment 1 on this Wednesday	No	The system outputs 'this Wednesday' as a location	5
French Oral Assessment tomorrow	No	Nothing was extracted. Only 'French' as the title	5
AI tutorial submission next week	No	'AI tutorial submission next week' extracted as Title only	5
AI project presentation timeslot (330pm)	No	Time (because of typo word time slot)	6
Quiz at dewan mutiara at 1600 hrs	No		6
random thing on 27-05-2025 at 2pm.	No	it gives start date 27/5 and end date 26/5	10
Physics test on every friday from 1pm to 4pm	No	Wrong time	11
On May 24, report assignment submit at 8pm	No	Event Title *	11
Test 1 on 29/05/2025 from 10am to 12pm at CS Lab	No	The date input cannot be 29/05/2025, have to be 29/5/202	12
Test 1 on 29/5/2025 from 10am to 12pm at CS Lab	No	The event title cannot save with digit like 'Test 1'	12
work on May 27 at 9am to 6pm every monday to friday	No	remove to friday	13
Math test on 5/26 at Warren Lecture Hall 5pm	No	Event title: Hall	14
Lab Report due on 5/28 during lab	No	Location: 5/28 during lab	14

Figure 5.6 The list of event prompts that EduCalendar failed to extract correctly.

#	Event Prompt	Expected Output	Actual Output	Explanation
1	Assignment on June 7 at 9am to June 15 10am	event_name: Assignment date_range: June 7 to June 15 time_range: 9am to 10am loc: freq: parsed_start_date: 2025-05-31 parsed_end_date: 2025-06-15 parsed_start_time: 9:00 parsed_end_time: 10:00 recurrence_pattern:	event_name: date_range: 9am to June 15 time_range: 10am loc: freq: parsed_start_date: 2025-05-31 parsed_end_date: 2025-06-15 parsed_start_time: 10:00 parsed_end_time: recurrence_pattern:	Date and time details are mixed together, causing confusion when extracting date and time entities
2	French Assessment 1 on this Wednesday	event_name: French Assessment 1 date_range: this Wednesday time_range: loc: freq: parsed_start_date: 2025-05-28 parsed_end_date: parsed_start_time: parsed_end_time: recurrence_pattern:	event_name: French Assessment 1 date_range: time_range: loc: this Wednesday freq: parsed_start_date: parsed_end_date: parsed_start_time: parsed_end_time: recurrence_pattern:	Custom model's training data did not account for relative time references
3	French Oral Assessment tomorrow	event_name: French Oral Assessment date_range: tomorrow time_range: loc: freq: parsed_start_date: 2025-05-27 parsed_end_date: parsed_start_time: parsed_end_time: recurrence_pattern:	event_name: French Oral Assessment date_range: Oral Assessment tomorrow time_range: loc: freq: parsed_start_date: parsed_end_date: parsed_start_time: parsed_end_time: recurrence_pattern:	Custom model's training data did not account for relative time references
4	AI tutorial submission next week	event_name: AI tutorial submission date_range: next week time_range: loc:	event_name: AI tutorial submission next week date_range: time_range: loc:	Custom model's training data did not account for

		freq: parsed_start_date: parsed_end_date: parsed_start_time: parsed_end_time: recurrence_pattern:	freq: parsed_start_date: parsed_end_date: parsed_start_time: parsed_end_time: recurrence_pattern:	relative time references, 'next week' time reference is too vague
5	AI project presentation timeslot (330pm)	event_name: AI project presentation timeslot date_range: time_range: 330pm loc: freq: parsed_start_date: parsed_end_date: parsed_start_time: 15:30 parsed_end_time: recurrence_pattern:	event_name: AI project presentation timeslot (date_range: time_range: 330pm loc: freq: parsed_start_date: parsed_end_date: parsed_start_time: 15:30 parsed_end_time: recurrence_pattern:	'timeslot' has typo
6	random thing on 27- 05-2025 at 2pm.	Event_name: random thing date_range: 27-05-2025 time_range: 2pm. Loc: freq: parsed_start_date: 2025-05- 27 parsed_end_date: parsed_start_time: 14:00 parsed_end_time: recurrence_pattern:	event_name: random thing date_range: 27-05- time_range: 2pm. Loc: freq: parsed_start_date: 2025-05- 27 parsed_end_date: 2025-05-31 parsed_start_time: 14:00 parsed_end_time: recurrence_pattern:	Confusion in model's entity boundaries when dash is used in dates, post processing uses dash to split multiple date entries like '19/5 – 20/5'
7	Physics test on every Friday from 1pm to 4pm	event_name: Physics test date_range: time_range: from 1pm to 4pm loc: freq: every Friday parsed_start_date: parsed_end_date: parsed_start_time: parsed_end_time: recurrence_pattern: Friday	event_name: Physics test date_range: from 1pm to 4pm time_range: loc: freq: every Friday parsed_start_date: 2025-05- 31 parsed_end_date: 2025-05-31 parsed_start_time: parsed_end_time: recurrence_pattern: Friday	Mislabeling time_range as date_range

8	On May 24, report assignment submit at 8pm	event_name: report assignment submit date_range: May 24 time_range: 8pm loc: freq: parsed_start_date: 2025-05-24 parsed_end_date: parsed_start_time: 20:00 parsed_end_time: recurrence_pattern:	event_name: date_range: May 24 time_range: 8pm loc: , report assignment submit freq: parsed_start_date: 2025-05-24 parsed_end_date: parsed_start_time: 20:00 parsed_end_time: recurrence_pattern:	Mislabeling event_name as loc
9	Test 1 on 29/05/2025 from 10am to 12pm at CS Lab	event_name: Test 1 date_range: 29/05/2025 time_range: from 10am to 12pm loc: CS Lab freq:	event_name: Test date_range: 29/05/2025 time_range: from 10am to 12pm loc: CS Lab freq:	29/05/2025 is correctly extracted on developer's end. Model's training data lacks examples of event names containing numbers
10	Test 1 on 29/5/2025 from 10am to 12pm at CS Lab event_name: Test	parsed_start_date: 2025-05-29 parsed_end_date: parsed_start_time: 10:00 parsed_end_time: 12:00 recurrence_pattern:	parsed_start_date: 2025-05-29 parsed_end_date: parsed_start_time: 10:00 parsed_end_time: 12:00 recurrence_pattern:	
11	work on May 27 at 9am to 6pm every Monday to Friday	event_name: work date_range: May 27 time_range: 9am to 6pm loc: freq: every Monday parsed_start_date: 2025-05-27 parsed_end_date: parsed_start_time: 09:00 parsed_end_time: 18:00 recurrence_pattern: every Monday to Friday	event_name: to Friday date_range: May 27 time_range: 9am to 6pm loc: freq: every Monday parsed_start_date: 2025-05-27 parsed_end_date: parsed_start_time: 09:00 parsed_end_time: 18:00 recurrence_pattern: Monday	Model's training data lacks the frequency pattern of "X to Y", causing confusion
12	Math test on 5/26 at Warren Lecture Hall 5pm	event_name: Math test date_range: 5/26 time_range: 5pm loc: Warren Lecture Hall freq:	event_name: Hall date_range: 5/26 time_range: 5pm loc: Warren Lecture freq:	Including a preposition between 'Warren Lecture

		parsed_start_date: 2025-05-26 parsed_end_date: parsed_start_time: 17:00 parsed_end_time: recurrence_pattern:	parsed_start_date: 2025-05-26 parsed_end_date: parsed_start_time: 17:00 parsed_end_time: recurrence_pattern:	Hall' and '5pm' helps the model extract the entities correctly. Model relies on syntactic cues to distinguish between entities
13	Lab Report due on 5/28 during lab	event_name: Lab Report due date_range: 5/28 time_range: loc: lab freq: parsed_start_date: parsed_end_date: parsed_start_time: parsed_end_time: recurrence_pattern:	event_name: Lab Report due date_range: time_range: loc: 5/28 during lab freq: parsed_start_date: parsed_end_date: parsed_start_time: parsed_end_time: recurrence_pattern:	Model is not used to the preposition 'during'. The model extracts correctly when 'during' is changed to 'at'

Table 5.21 Analysis of the event prompts that are extracted incorrectly.

From Table 5.21, the custom NLP NER model does not properly recognize relative time terms (e.g. 'tomorrow', 'this Wednesday'), date format with dashes (e.g. 25-05-2025), frequency format of 'X to Y' (e.g. every Monday to Friday), and event names containing numbers. This is due to the lack of examples of similar structure and format in the custom model's training data. The custom model also relies heavily on prepositions like 'on' and 'at', and fails to recognize 'during' as this term is not represented in the training data.

5.3.4 System Acceptance

Please indicate how much you agree with the following statements using a scale from 1 to 5 (1 = Strongly Disagree, 5 = Strongly Agree):

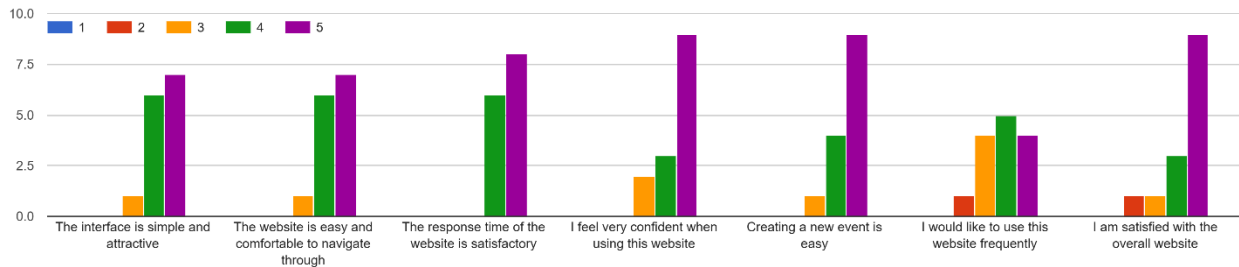


Figure 5.7 EduCalendar's overall system acceptance.

Figure 5.7 shows the 5-point Likert scale of the respondents' overall system acceptance; the responses are as follows:

- 1) The interface is simple and attractive
 - a. 7 respondents strongly agree, six respondents agree, one respondent is neutral.
- 2) The website is easy and comfortable to navigate through
 - a. 7 respondents strongly agree, six respondents agree, one respondent is neutral.
- 3) The response time of the website is satisfactory
 - a. 8 respondents strongly agree, six respondents agree
- 4) I feel very confident when using this website
 - a. 9 respondents strongly agree, three respondents agree, two respondents are neutral
- 5) Creating a new event is easy
 - a. 9 respondents strongly agree, four respondents agree, one respondent is neutral
- 6) I would like to use this website frequently
 - a. 4 respondents strongly agree, five respondents agree, four respondent is neutral, one respondent disagrees

7) I am satisfied with the overall website

- a. 9 respondents strongly agree, three respondents agree, one respondent is neutral, one respondent disagrees

Overall, there is a high agreement on visual design, navigation, and system performance. This indicates that the respondents generally found EduCalendar’s interface to be aesthetically pleasing, user friendly, and fast and responsive. A strong majority (9 respondents) strongly agree that they feel confident when using EduCalendar and that the NLP NER custom model is good enough that they felt creating a new event is easy. When asked on continued usage and overall satisfaction, the 14 respondents gave mixed feedback. The presence of lower scores indicates that there may be some reservations or areas for improvement in terms of long-term engagement.

If encountered any other errors when using EduCalendar, please list down here.

3 responses

Dates were automatically changed to one day before initial setting in home page. For example, the prompt stated an exam at 3pm on 5/29, but the home page changed it to 5/28. Times have all been changed to "all day". User location is California, USA

cannot create event with everyday (eg. every monday to friday)

Website not using the correct extracted info and model not extracting info correctly

Figure 5.8 Other bugs in EduCalendar encountered by respondents.

Figure 5.8 shows a collection of additional bugs encountered by respondents while using EduCalendar. One user reported that event dates appeared one day earlier on the homepage compared to the original input. This issue, along with another report stating that the website did not use the correct extracted information, suggests a likely time zone mismatch between the client (user's browser) and the server or database (e.g., UTC vs. local time). Another respondent noted that recurring event creation failed for inputs such as “every Monday to Friday”. This limitation

stems from the custom NLP NER model's inability to correctly extract multi-day recurrence patterns, which were not sufficiently represented in the training data.

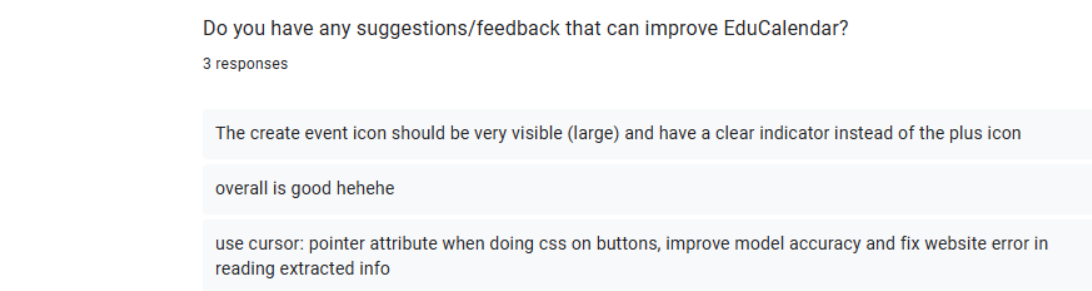


Figure 5.9 Suggestions provided by the respondents that can improve EduCalendar.

Figure 5.9 highlights the respondents' suggestions for improvements for EduCalendar. One respondent expressed general satisfaction with the system, stating that the overall experience was good, and there the other two respondents noted on the UI visibility and interactivity of the system. These suggestions indicate that while the system is functional, minor refinements to the user interface could significantly improve the overall user experience.

5.4 Summary

This chapter presented the testing phase of EduCalendar, covering both functional testing and User Acceptance Testing (UAT). All core features, which including sign-up, login, course and event handling, were successfully tested. UAT results showed positive feedback on usability and accessibility, but also highlighted issues such as incorrect date display and NLP extraction errors with relative time expressions and frequency ranges. Improvements were made to the UI and postprocessing logic to address these issues. Overall, the system was found to be functional, reliable, and user-friendly, with some areas identified for future enhancement.

Chapter 6 : Conclusion and Future Work

6.1 Introduction

This chapter provides a reflection on the overall project outcomes, including the extent to which the project objectives were achieved. It also outlines the system's limitations and constraints encountered during development and testing. Lastly, it presents potential improvements and future enhancements for both the EduCalendar system and its custom NLP NER model.

6.2 Achievement

There are a total of three objectives that has been set for this project as guidelines. Table 6.1 below summarizes the project objectives stated in Chapter 1 and the achievements throughout this entire project.

Objective	Achievement
To review existing systems that are relevant and curate requirements for EduCalendar, a web-based smart calendar's design and functionalities.	Three relevant existing systems, Google Calendar, SkedPal, and Fantastical, has been reviewed and analyzed in Chapter 2. Based on the analysis of these systems and the results from a Google Form questionnaire used for requirement elicitation in Chapter 3, the functional and non-functional requirements for EduCalendar were curated.
To design and develop EduCalendar for students to plan their projects, assignments, tests, quizzes, and personal commitments.	The wireframe of EduCalendar has been designed in Chapter 3. The development of the EduCalendar website and training of the custom NLP NER model has been done in Chapter 4.
To evaluate the EduCalendar's usability through functional testing and user acceptance testing.	EduCalendar's usability is evaluated via functional testing and User Acceptance Test (UAT) in Chapter 5 to ensure the system is functional and reliable on both the developer's and the user's side.

--	--

Table 6.1 Summary of the project's objectives and achievements.

6.3 Limitations and Constraints

EduCalendar has a few limitations and constraints as follows:

1. Custom NLP NER Model

- Requires clear and formal languages with prepositions (e.g. at, on) to ensure correct entity extraction.
- Cannot process multiple events from a single event prompt.
- Weak handling of frequency patterns like 'every Monday to Friday'.
- Relative time expressions (e.g. 'this Friday', 'tomorrow') has to be handled using post processing due to it not being considered when constructing the test data.
- Struggles with event titles that are very short or contains numbers.

2. EduCalendar website

- No integration with external digital calendars (e.g., Google Calendar, Apple Calendar).
- No multi-user collaboration or sharing.
- No notification/reminder system for upcoming events.
- Lacks to-do feature that allows users to tick off completed events.

6.4 Future Works

The limitations present opportunities for future improvements to increase system flexibility, intelligence, and adoption:

1. Custom NLP NER Model

- Train the model with informal and varied sentence structures to handle natural user input.
- Introduce logic to support multiple events per sentence.
- Improve handling of relative time references.
- Improve handling of frequency recognition for range expressions like “every Monday to Friday”

2. EduCalendar website

- Integration with third-party systems (e.g., Google Calendar, Apple Calendar, university systems, emails).
- Enable collaborative features, allowing users to share calendars or events with peers.
- Add reminder/notification features for upcoming tasks and deadlines.
- Add to-do feature that allows users to tick off completed events.
- Introduce AI-based suggestions for scheduling based on workload and free time.

6.5 Conclusion

In conclusion, this project successfully developed EduCalendar, a web-based smart calendar integrated with an NLP NER model that simplifies event creation through natural language input. The system achieved its core objectives, providing students with a more intuitive way to manage their schedules. While the NLP model and the EduCalendar system showed strong usability and functionality, certain limitations remain in flexibility, integration, and language processing. With further refinements and feature expansions, EduCalendar has the potential to become an academic scheduling and planning tool that bridges the gap between AI and student productivity.

References

- Ashana, L., Samanthi, S., Malithi, M., & Shyam, R. (2021). Design-Thinking and UCD Combination for Designing Effective Time Management Assistant Mobile App. *RoCHI - International Conference on Human-Computer Interaction*, 111–118. <https://doi.org/10.37789/rochi.2021.1.1.17>
- Azar, A. S., Dastan, H. M., Dindar, M. A., Hajar, M. Y., Ibrahim, M. I., Naaman, O., Nareen O. M. Salim, Shakir, F. K., Siddeeq, Y. A., & Zryan, N. R. (2021). Review on Natural Language Processing Based on Different Techniques. *Asian Journal of Research in Computer Science*, 10(1). <https://doi.org/10.9734/ajrcos/2021/v10i130231>
- Derczynski, L. (2016). *Complementarity, F-score, and NLP Evaluation*. <https://aclanthology.org/L16-1040/>
- Dhiman, K. L. (2024). *Reassessment of Software Development Life Cycle Models*. <https://journal.ijresm.com/index.php/ijresm/article/view/2929>
- Esuli, A., & Sebastiani, F. (2010). *Evaluating Information Extraction*. https://www.researchgate.net/publication/225269752_Evaluating_Information_Extraction
- Figma. (n.d.). *Release Notes*. Figma. Retrieved November 15, 2024, from <https://www.figma.com/release-notes/>
- geeksforgeeks. (2025a, February 18). *Firestore - Introduction*. Geeksforgeeks. <https://www.geeksforgeeks.org/firebase/firebase-introduction/>
- geeksforgeeks. (2025b, April 12). *Prototyping Model - Software Engineering*. Geeksforgeeks. <https://www.geeksforgeeks.org/software-engineering-prototyping-model/>
- Hyder, M., & Nafees, G. M. (2023). HARNESSING DEEP LEARNING AND NATURAL LANGUAGE PROCESSING (NLP). In *ENGINEERING THE FUTURE: MACHINE LEARNING AND DATA SCIENCE IN PRACTICE*. NOBLE SCIENCE PRESS. <https://doi.org/10.52458/9789388996747.nsp2023.eb.ch-02>
- IBM. (n.d.). *What is named entity recognition?* IBM.
- Josh, C. (2015, December 3). *23-Year-Old's Design Collaboration Tool Figma Launches With \$14M To Fight Adobe*. TechCrunch. <https://techcrunch.com/2015/12/03/figma-vs-goliath/>
- Jugran, S., Kumar, A., Tyagi, B. S., & Anand, V. (2021). Extractive Automatic Text Summarization using SpaCy in Python NLP. *2021 International Conference on Advance Computing and Innovative Technologies in Engineering, ICACITE 2021*, 582–585. <https://doi.org/10.1109/ICACITE51222.2021.9404712>
- Kaur, N., Saha, A., Swami, M., Singh, M., & Dalal, R. (2024). Bert-Ner: A Transformer-Based Approach For Named Entity Recognition. *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 1–7. <https://doi.org/10.1109/ICCCNT61001.2024.10724703>

- Kingsley, U. (2021, August 10). *Learn Web Development Basics – HTML, CSS, and JavaScript Explained for Beginners*. FreeCodeCamp. <https://www.freecodecamp.org/news/html-css-and-javascript-explained-for-beginners/>
- Kulkarni, R., Varade, Dr. P., & Yellalwar, Dr. R. G. (2023). NLP and Human-Computer Interaction: Enhancing User Experience through Language Technology. *International Journal for Research in Applied Science and Engineering Technology*, 11(10), 1393–1396. <https://doi.org/10.22214/ijraset.2023.56219>
- Lai, V. D. (2022). *Event Extraction: A Survey*. <http://arxiv.org/abs/2210.03419>
- Liu, J., Min, L., & Huang, X. (2021). *An overview of event extraction and its applications*. <http://arxiv.org/abs/2111.03212>
- Microsoft Learn. (2024, November 21). *Evaluation metrics for custom named entity recognition models*. Microsoft.
- Nair, A. M., Justus, A. A., Ramesh, A., & Rajan, B. (2020). Event Extraction from Emails. In *International Journal of Computer Applications* (Vol. 176, Issue 41).
- NLTK. (n.d.). *NLTK*. NLTK. Retrieved November 15, 2024, from <https://www.nltk.org/>
- Rathod, A. (n.d.). *What is Software Tool?* GeeksforGeeks. Retrieved November 15, 2024, from <https://www.geeksforgeeks.org/what-is-software-tool/>
- Shelar, H., Kaur, G., Heda, N., & Agrawal, P. (2020). Named Entity Recognition Approaches and Their Comparison for Custom NER Model. *Science and Technology Libraries*, 39(3), 324–337. <https://doi.org/10.1080/0194262X.2020.1759479>
- Shreyas, A., Priyanka, G., Pearl, M., & Swapnal, S. (2018). EVENT INFORMATION EXTRACTION FROM E-MAIL AND UPDATING EVENT IN CALENDAR. *International Journal of Advance Research and Innovative Ideas in Education*, 4(3). https://ijariie.com/AdminUploadPdf/EVENT_INFORMATION_EXTRACTION_FROM_E_MAIL_AND_UPDATING_EVENT_IN_CALENDAR_ijariie8590.pdf?srsltid=AfmBOookjPE2NfDLL6-qkm3Vxzixd1-MsgaOkz1ib3OGglqgE60Bokyx
- Singh, S. (2018). *Natural Language Processing for Information Extraction*. <http://arxiv.org/abs/1807.02383>
- SkedPal. (n.d.). *Pricing*. SkedPal. Retrieved November 15, 2024, from <https://skedpal.com/pricing/>
- spaCy. (n.d.-a). *Facts & Figures*. SpaCy. Retrieved November 16, 2024, from <https://spacy.io/usage/facts-figures>
- spaCy. (n.d.-b). *Training Pipelines & Models*. SpaCy. Retrieved February 17, 2025, from <https://spacy.io/usage/training>
- Stryker, C., & Holdsworth, J. (2024, August 11). *What is NLP?* IBM. <https://www.ibm.com/think/topics/natural-language-processing>

Svelte. (n.d.-a). *Svelte*. Svelte. Retrieved November 15, 2024, from <https://svelte.dev/>

Svelte. (n.d.-b). *What is SvelteKit?* Retrieved June 22, 2025, from <https://svelte.dev/tutorial/kit/introducing-sveltekit>

Ushio, A., & Camacho-Collados, J. (2022). *T-NER: An All-Round Python Library for Transformer-based Named Entity Recognition*. 53–62. <https://doi.org/https://doi.org/10.48550/arXiv.2209.12616>

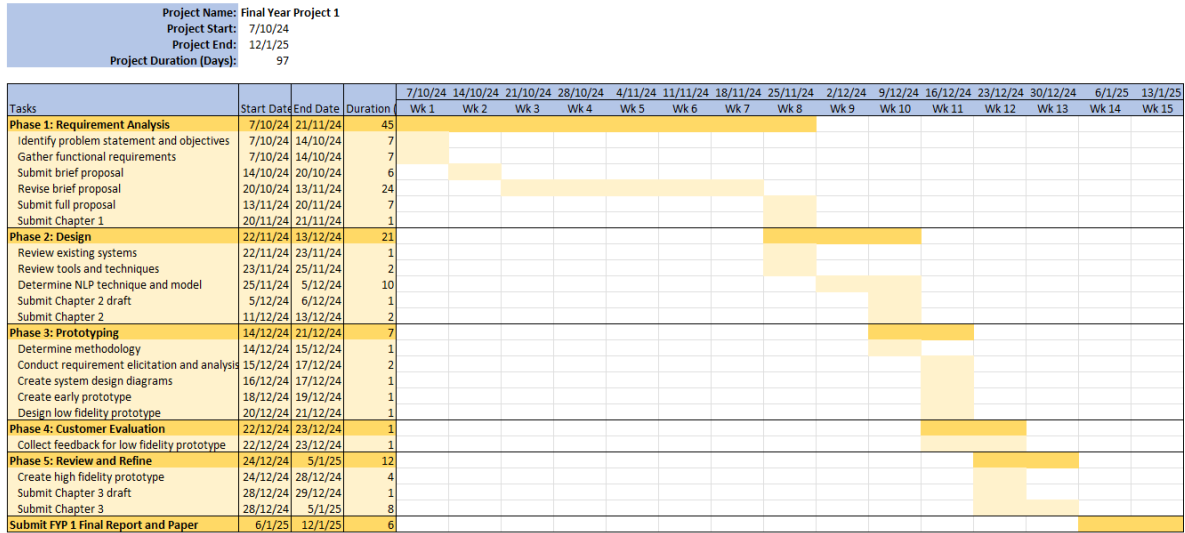
Visual Studio Code. (2024, October 29). *Why did we build Visual Studio Code*. Visual Studio Code. <https://code.visualstudio.com/docs/editor/whyvscodetext=At%20its%20heart%2C%20Visual%20Studio,selection%2C%20snippets%2C%20and%20more>

w3schools. (n.d.). *Python Introduction*. W3schools. Retrieved November 15, 2024, from https://www.w3schools.com/python/python_intro.asp

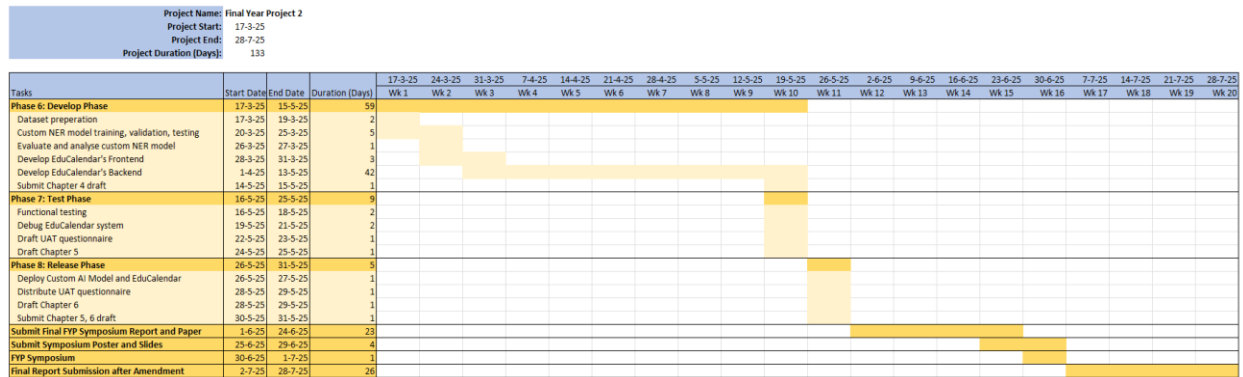
Xiang, W., & Wang, B. (2019). A Survey of Event Extraction from Text. *IEEE Access*, 7, 173111–173137. <https://doi.org/10.1109/ACCESS.2019.2956831>

Appendix

Appendix A: Gantt Chart



Project Schedule for FYP1



Project Schedule for FYP2

Appendix B: Google Form Questionnaire

FYP Questionnaire: Enhancing Daily Planning for Students with NLP-Driven Event Automation

Hi, I'm Rebecca Lai, a final-year student at UNIMAS, studying Computational Science at the Faculty of Computer Science & Information Technology (FCSIT).

I am currently developing a **web-based digital calendar system with AI integration**, specifically using **Natural Language Processing (NLP)**, which allows machines to interpret text and speech data, to **automate event creation**. This feature allows users to input phrases like *"Tea with friends every Wednesday at 2:30 PM for one and a half hours at Student Pavilion,"* and the system will automatically extract and organize the event details—such as the event name, date, time, duration, and location—into the calendar.

The goal of this integration is to:

- Enhance the user experience for students using daily planners.
- Save time on managing tasks, allowing users to focus more on completing them.

This survey includes **3 sections (User Demographics, Digital Calendar Usage, and AI Integration in Digital Calendars)** and will take only 2–3 minutes to complete. Your feedback is crucial in helping me design a user-friendly and efficient system.

If you have any questions or suggestions, feel free to contact me at 81036@siswa.unimas.my.

Thank you for your participation ❤️

User Demographics

1. How old are you? *

- ☐ 18
- ☐ 19
- ☐ 20
- ☐ 21
- ☐ 22
- ☐ 23
- ☐ 24 and above

2. What is your gender? *

- ☐ Female
- ☐ Male

3. What is your current major in university? *

Your answer _____

Digital Calendar Usage

4. Have you ever regularly used a digital calendar (e.g., Google Calendar, Apple Calendar, Outlook Calendar) to keep track of your academic responsibilities (assignments, projects, quizzes) and daily schedules? *

☐ Yes

☐ No

5. If your answer in (4) is no, why didn't you use a digital calendar regularly? *

☐ Writing down my commitments on physical planners/notepad is sufficient

☐ Manually entering tasks and events on digital calendars is too tedious

☐ Difficulty in maintaining and updating my digital calendar consistently

☐ I find digital calendars difficult to use or navigate

☐ I don't feel the need for a calendar to track my responsibilities.

☐ Other: _____

6. If your answer in (4) is yes, what is the name of the digital calendar application? *

☐ Google Calendar

☐ Apple Calendar

☐ Outlook

☐ Other: _____

7. If your answer in (4) is yes, how often did you use the digital calendar? *

☐ Daily

☐ Weekly

☐ Monthly

☐ Whenever when I need to check my schedules or deadlines

☐ Rarely or only during exams or busy periods

User Experience and AI Integration in Digital Calendars

8. How important do you think it is to be able to keep track of your academic responsibilities (assignments, projects, quizzes) and daily schedules? *

	1	2	3	4	5	
Not important	☐	☐	☐	☐	☐	Very important

9. Do you find manually entering calendar events on digital calendars tedious or time-consuming? *

- ☐ Yes
- ☐ No
- ☐ Sometimes, depending on the task

10. If there is such an app that would be able to create events given some inputs *
in the digital calendar, do you think it will be beneficial to you?

☐ Yes

☐ No

11. Have you heard of AI integration in digital calendars? *

☐ Yes

☐ No

12. In what way do you think AI integration in digital calendars would help you as *
a student?

☐ Event automation by extracting event details

☐ Event priority automation

☐ Handle recurring events

UI Preferences and Evaluation

The **Figure 1** and **Figure 2** shown below are 2 different types of UI arrangement on a digital calendar web system on a mobile screen. Answer question 13 and 14 based on the 2 figures shown below.



13. Between the header section of Figure 1 and 2 (navigation between the home, calendar and course page are hidden within the menu icon), which type of header do you prefer? *

- ☐ Figure 1
- ☐ Figure 2

14. Between the body section the homepage of Figure 1 and 2, which layout do you prefer? *

☐ Figure 1

☐ Figure 2

For **mobile screen**, **Figure 3** shows the UI when the users click on the plus icon on the bottom right of the screen (as shown in Figure 1) to create a new event, and **Figure 4** shows how the event details are presented after users enter an event prompt in the textbox.

For **desktop screen's** interface, is shown in **Figure 5** and **6**.

The screenshot shows a mobile app interface for creating a new event. At the top is a 'website bar' with a grey header. Below it is a grey bar with a circular profile icon and the text 'Hello, {student_name}'. A navigation bar contains three icons: a house for 'Home', a calendar for 'Calendar', and a book for 'Courses'. The main content area has three buttons: 'Cancel', 'New Event', and 'Add'. Below these is a text input field containing 'MM test 2 on 19/12 at 10am to 12 am at Unix lab'. Under the input field is a 'Manual add' link. Below that are two rows of selection options: 'Course: Course 1 Course 2' and 'Type: Class Assignment Test Other'. The bottom of the screen is empty.

Figure 3

The screenshot shows the same mobile app interface as Figure 3, but with the event details filled in. The text input field now contains 'PP test on 20/12 at 10am to 12 pm at |'. The 'Manual add' link is still present. The 'Course' and 'Type' selection options remain the same. Below the selection options, the event details are displayed: 'Event title: PP test', 'Date: 20 Dec 2024', 'Time: 10:00am to 12:00pm', and 'Location: -'. A virtual keyboard is visible at the bottom of the screen, showing the letters QWERTYUIOP, ASDFGHJKL, ZXCVBNM, and the space, return, and 123 keys.

Figure 4

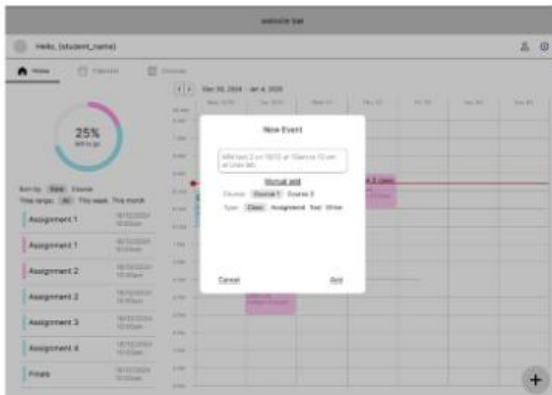


Figure 5

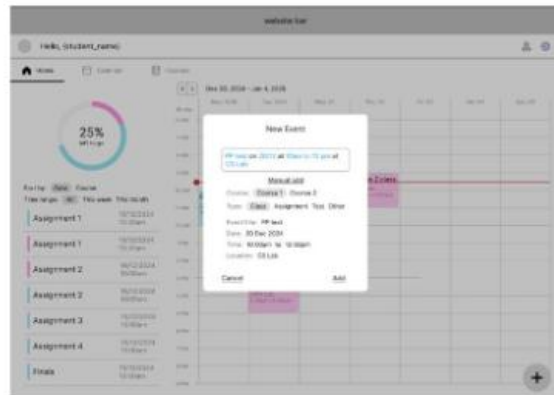


Figure 6

15. Is the "Create Event" interface on both mobile (Figure 3 & 4) and desktop screen (Figure 5 & 6) visually clear and easy to understand? *

☐ Yes

☐ No

☐ Other: _____

Figure 7 shows the courses page of a digital calendar web system on a mobile screen, and **Figure 8** is the courses page on a desktop screen

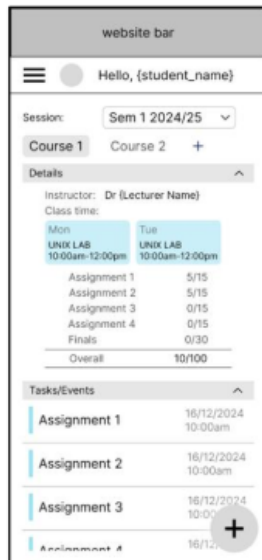


Figure 7

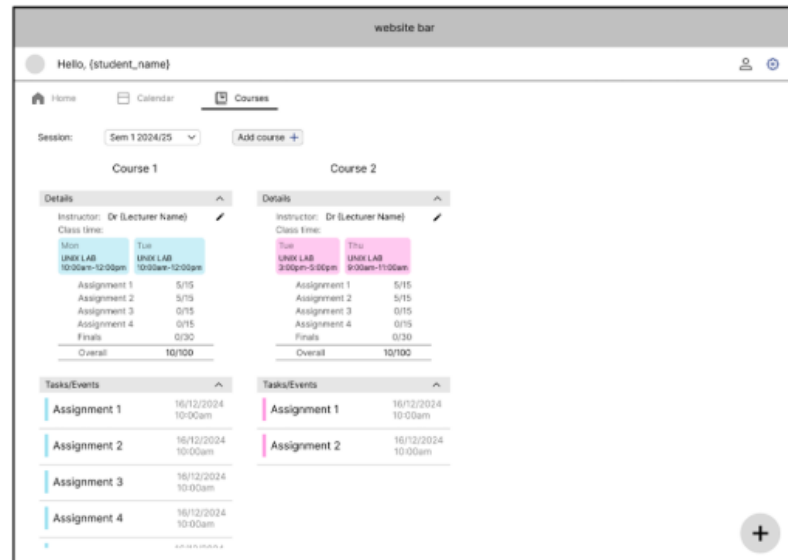


Figure 8

16. Are the details displayed in an organized and easy-to-read format? *

☐ Yes

☐ No

☐ Other: _____

Appendix C: Google Form UAT Questionnaire

FYP2 EduCalendar User Acceptance Test

Hi, I'm Rebecca Lai, a final-year student at UNIMAS, studying Computational Science at the Faculty of Computer Science & Information Technology (FCSIT).

I am currently developing a **EduCalendar, web-based digital calendar system with AI integration**, specifically using **Natural Language Processing (NLP)**, which allows machines to interpret text and speech data, to **automate event creation**. This feature allows users to input phrases like *"Tea with friends every Wednesday at 2:30 PM for one and a half hours at Student Pavilion,"* and the system will automatically extract and organize the event details—such as the event name, date, time, duration, and location—into the calendar.

The goal of this integration is to:

- Enhance the user experience for students using daily planners.
- Save time on managing tasks, allowing users to focus more on completing them.

The EduCalendar website can be access at this link: [EduCalendar website](#)

This survey includes **4 sections (User Demographics, Website Accessibility, Tasks and System Acceptance)** and will take only 5-10 minutes to complete. Your feedback is crucial in helping me in testing the accessibility and user acceptability of the EduCalendar system.

If you have any questions or suggestions, feel free to contact me at **81036@siswa.unimas.my**.

Thank you for your participation ❤️

Section 1: User Demographics

How old are you? *

- ☐ 18
- ☐ 19
- ☐ 20
- ☐ 21
- ☐ 22
- ☐ 23
- ☐ 24 and above

What is your gender? *

- ☐ Female
- ☐ Male

What is your current major in university? *

Your answer _____

Section 2: Website Accessibility

Are you accessing the EduCalendar website on your mobile phone or PC/laptop? *

- ☐ Mobile phone
- ☐ PC/laptop

Are you able to access the EduCalendar website? *

- ☐ Yes
- ☐ No

If no, why? What is the browser you are using to access the website?

Your answer _____

Are you able to create an account? *

- ☐ Yes
- ☐ No

If no, why?

Your answer

Section 3: (Task 1) Course Creation

Create **two new courses** of your choice.

Were you able to create courses without any issues? *

☐ Yes

☐ No

If no, briefly describe the issue(s):

Your answer

Section 3: (Task 2) Course Creation

Create **five events** using the text area input according to your writing preferences (e.g., "Science test on 15/06/2025 at 10am at Block B")

Enter your event prompt #1. *

Your answer

Was the event prompt #1 extracted correctly? *

☐ Yes

☐ No

If prompt #1 was not extracted correctly, what was incorrect or missing in the extracted data?

Your answer

Enter your event prompt #2. *

Your answer _____

Was the event prompt #2 extracted correctly? *

☐ Yes

☐ No

If prompt #2 was not extracted correctly, what was incorrect or missing in the extracted data?

Your answer _____

Enter your event prompt #3. *

Your answer _____

Was the event prompt #3 extracted correctly? *

☐ Yes

☐ No

If prompt #3 was not extracted correctly, what was incorrect or missing in the extracted data?

Your answer

Enter your event prompt #4. *

Your answer

Was the event prompt #4 extracted correctly? *

☐ Yes

☐ No

If prompt #4 was not extracted correctly, what was incorrect or missing in the extracted data?

Your answer

Enter your event prompt #5. *

Your answer

Was the event prompt #5 extracted correctly? *

☐ Yes

☐ No

If prompt #5 was not extracted correctly, what was incorrect or missing in the extracted data?

Your answer

Section 4: System Acceptance

Please indicate how much you agree with the following statements using a scale * from 1 to 5 (1 = Strongly Disagree, 5 = Strongly Agree):

	1	2	3	4	5
The interface is simple and attractive	☐	☐	☐	☐	☐
The website is easy and comfortable to navigate through	☐	☐	☐	☐	☐
The response time of the website is satisfactory	☐	☐	☐	☐	☐

I feel very
confident when
using this
website

☐☐☐☐☐

Creating a new
event is easy

☐☐☐☐☐

I would like to
use this
website
frequently

☐☐☐☐☐

I am satisfied
with the overall
website

☐☐☐☐☐

If encountered any other errors when using EduCalendar, please list down here.

Your answer

Do you have any suggestions/feedback that can improve EduCalendar?

Your answer

Appendix D: Full results of the UAT Questionnaire's (Task 2) Event Creation section

1	Prompt	Correctly extracted?	If no, why (respondent index)	Respondent index
2	PR test on 1/6/2025 at 10am at building A	Yes		1
3	Japanese test on 10/6/2025 at 7am at building C	Yes		1
4	International relationship test on 15/6/2025 at 10am at b	Yes		1
5	Public media test on 23/6/2025 at 3pm at building A	Yes		1
6	World history test on 27/6/2025 at 9am at building E	Yes		1
7	Quiz on June 6 at 9am in IT lab2	Yes		2
8	Quiz 2 on 15/06/2025 at 10am at IT lab 1	Yes		2
9	Assignment on June 7 at 9am to June 15 10am	No	The Ai didn't recognise and extract the event title	2
10	Quick Test on every Monday at IT lab	Yes		2
11	Essay writing on June 9 at 5pm in Hall 3	Yes		2
12	computing test on 15/6/2025 at 11am	Yes		3
13	mathematics test on 15/6/2025 at 11am	Yes		3
14	mathematics test on 15/6/2025 at 11am	Yes		3
15	mathematics test on 15/6/2025 at 11am	Yes		3
16	mathematics test on 15/6/2025 at 11am	Yes		3
17	quiz AI	Yes		4
18	fyp due date	Yes		4
19	gawai	Yes		4
20	balik sibu	Yes		4
21	project presentation	Yes		4
22	French Assessment 1 on this Wednesday	No	The system outputs 'this Wednesday' as a location	5
23	French Oral Assessment for the 29/5 - 31/5	Yes		5
24	French Oral Assessment tomorrow	No	Nothing was extracted. Only 'French' as the title	5
25	AI final exam on 2 June	Yes		5
26	AI tutorial submission next week	No	'AI tutorial submission next week' extracted as Title only	5
27	Math test at 4pm	Yes		6
28	AI project presentation timeslot (330pm)	No	Time (because of typo word time slot)	6
29	Quiz at dewan mutiara at 1600 hrs	No		6
30	Project submission via eleap (5pm)	Yes		6
31	Assignment submission at 9pm (ctf3)	Yes		6
32	AI Quiz	Yes		7
33	AI Class	Yes		7
34	Midterm Exam	Yes		7
35	Presentation	Yes		7
36	Quiz	Yes		7
37	Midterm CS	Yes		8
38	Quiz AI	Yes		8
39	Assignment CS	Yes		8
40	Assignment AI 1	Yes		8
41	Assignment AI 2	Yes		8
42	Accounting test on 30/5/2025 at 1pm at Room 100	Yes		9
43	Accounting assignment due on 1/6/2025 at 1pm	Yes		9
44	Financial risk management test on 5/6/2025 at 1pm at Ro	Yes		9
45	Financial risk management presentation on 29/5/2025 at	Yes		9
46	Financial risk management assignment discussion on 1/6/	Yes		9
47	Quiz 2 on May 27 12pm at eleap.	Yes		10
48	FYP chapter 5 and 6 submission on 31 May at 5 pm.	Yes		10
49	random thing on 27-05-2025 at 2pm.	No	it gives start date 27/5 and end date 26/5	10
50	test event every monday at 14:00	Yes		10
51	MCQ Quiz on May 27 at 3pm in Exam Hall.	Yes		10
52	Math test on 28 May at 8pm in every week	Yes		11
53	Physics test on every friday from 1pm to 4pm	No	Wrong time	11
54	Biology assignment on May 31 at 12am	Yes		11

55	Coding task assignment on May 27 at 6pm	Yes		11
56	On May 24, report assignment submit at 8pm	No	Event Title *	11
57	Test 1 on 29/05/2025 from 10am to 12pm at CS Lab	No	The date input cannot be 29/05/2025, have to be 29/5/202	12
58	Test 1 on 29/5/2025 from 10am to 12pm at CS Lab	No	The event title cannot save with digit like "Test 1"	12
59	Math test, CS Lab, 10am - 12 pm	Yes		12
60	Security assignment meeting every Monday from 9am - 11am	Yes		12
61	meeting with SV from 1300 - 1500 at office	Yes		12
62	big data test on June 1 at 2pm	Yes		13
63	assignment due on June 12 at 12pm	Yes		13
64	smoke test on May 30 at 1pm to 3pm	Yes		13
65	work on May 27 at 9am to 6pm every monday to friday	No	remove to friday	13
66	lunch with rebecca on May 28 at 12pm	Yes		13
67	Final exam in BILD1 at 2pm on 5/30	Yes		14
68	Chem final at 3pm in Galbraith Hall on 5/29	Yes		14
69	Astrophysics project due by 9pm on 5/30	Yes		14
70	Math test on 5/26 at Warren Lecture Hall 5pm	No	Event title: Hall	14
71	Lab Report due on 5/28 during lab	No	Location: 5/28 during lab	14