Kathmandu University

Department of Computer Science and Engineering

Dhulikhel, Kavre

A Project Report
on
"Passive Growing App"

[Code No:COMP 116]

(For partial fulfillment of I Year/II Semester in Computer Science/Engineering)

Submitted by

Abhinav Bhatt (9)
Prakash Chaudhary(10)
Prisha Nepal (31)
Madhulika Yadav (57)
Praful Bhatt (61)

(Er.Subhadra Joshi)

Department of Computer Science and Engineering

Submission Date:2025-08-05

Bona fide Certificate

This project work on

“Passive Growing App”

is the bona fide work of

“Abhinav Bhatt

Prakash Chaudhary

Praful Bhatt

Prisha Nepal

Madhulika Yadav ”

who carried out the project work under my supervision.

Project Supervisor

_____

Er.Subhadra Joshi

Lecturer

Department of Computer Science and Engineering

Date:2025-8-5

Acknowledgement

We would like to express our heartfelt gratitude to the Department of Computer Science and Engineering (DOCSE) for providing us with the incredible opportunity to work on this project, “Passive Growing App." This project has expanded our technological horizons, improved our teamwork skills, and helped us better comprehend the tools and technology needed to create a fully effective recommendation website.

We would like to express our deepest gratitude to our supervisor, Er.Subhadra Joshi, for her invaluable guidance, unwavering support, and insightful feedback throughout the development of this project. Her mentorship has been instrumental in shaping our understanding of classroom management systems and refining our approach to problem-solving.

This project has greatly helped improve our understanding of the tools and technologies required to develop a fully functional recommendation website, broadened our technical horizons, and helped us develop teamwork skills.

We are also deeply thankful to our Project coordinator and supervisor, Mr. Suman Shrestha, for his unwavering support and valuable insights he has provided us with. His efforts have not been short of a guiding light for us.

Additionally, we extend our appreciation to all mentors and friends for their constant encouragement, support, and expert advice throughout the development process of “Passive Growing App” app. Their contributions have been instrumental in shaping the course of this wonderful journey. We sincerely look forward to their continued guidance in the future as well.

Abstract

Passive Growing App is a smart productivity system designed to help university students effectively manage their study schedules and enhance productivity by integrating a Smart Study Planner and Smart Reminder System. This project aims to assist students in organizing academic tasks, setting reminders, and tracking progress to foster better time management. Developed using C++ for core task management, the system will incorporate intelligent features such as personalized study suggestions based on priorities and deadlines. With a user-friendly interface and adaptive algorithms, Passive Growing App will serve as a digital assistant that aligns with individual learning patterns, helping reduce procrastination. Future enhancements may include mobile compatibility and machine learning integration for deeper productivity insights

Keywords*: Smart productivity system, study planner, reminder system, time management, C*

TABLE OF CONTENTS

List of Figures

Acronyms/Abbreviations

API – Application Program Interface

CSS - Cascading Style Sheet

HTTP - Hypertext Markup Language

RAM - Random Access Memory

REST - Representational State Transfer

RTX – Ray Tracing Texel eXtreme

SPA – Single Page Application

SQL – Structured Query Language

UI - User Interface

UX - User Experience

VRAM - VideoRandom Access Memory

Chapter 1 **Introduction**

1.1 Background

In recent years, universities have increasingly recognized the importance of productivity tools in helping students manage their academic workloads and achieve their learning goals. While existing tools like Google Calendar, Microsoft To-Do, and general reminder applications provide basic task scheduling and notifications, they often lack the personalized features needed to meet the unique demands of university students. These platforms do not integrate study planning, intelligent task prioritization, and adaptive reminders tailored to academic timelines, resulting in an unstructured approach to productivity.

At Kathmandu University (KU), students face challenges in balancing multiple courses, assignments, and extracurricular commitments. The absence of a dedicated productivity system often leads to poor time management, missed deadlines, and increased procrastination. General productivity tools are too generic and fail to provide the specific functionalities required for university-level study planning, such as dynamic reminders based on priorities, progress tracking, and smart scheduling.

Recent Developments:

- **Smart Productivity Tools:** Emerging platforms integrate basic scheduling with AI-driven features to improve task prioritization, but these tools are rarely designed with university students in mind.
- **Focus on Personalized Learning:** With the growth of adaptive learning systems, there is an increasing demand for tools that understand individual study habits and suggest optimal schedules.

- **Task Automation and Notifications:** Real-time reminders and automated rescheduling are becoming essential for students managing busy academic schedules.

Drawbacks of Existing Solutions:

- **Lack of Academic Focus:** Most existing tools cater to general users and fail to address the specific study needs of university students.
- **One-Size-Fits-All Approach:** Generic productivity tools lack adaptability and cannot create dynamic, personalized plans based on individual student requirements.
- **Limited Integration of Smart Features:** Current systems often do not incorporate intelligent suggestions or progress-based scheduling, resulting in inefficiencies.

To address these gaps, Passive Growing App is being developed as a smart productivity system specifically designed for university students. By combining a Smart Study Planner with a Smart Reminder System, Passive Growing App will provide features such as dynamic scheduling, intelligent reminders, and progress tracking. The system aims to create a personalized and adaptive productivity experience, helping students manage their academic workload efficiently and reduce procrastination.

1.2 Objectives

1. To develop Passive growing app, a platform that combines multiple productivity systems to help students manage their academic schedules efficiently.
2. To enable users to create and manage todo-lists, pomodoro timer, habit and productivity tracker each with their own profile.

3. To implement intelligent task prioritization features, enabling students to organize tasks based on deadlines, priorities, and study goals.
4. To provide progress tracking tools within Passive Growing app, allowing students to monitor their productivity and adjust study schedules accordingly.

1.3 Motivation and Significance

The motivation behind choosing this project arises from the increasing challenges university students face in managing their time, tasks, and academic responsibilities. Existing productivity tools like Google Calendar or Microsoft To-Do or web pomodoro apps provide basic task management and reminders, but they lack the personalization and intelligence required to meet the specific needs of students. As students ourselves, we recognized the absence of a dedicated platform that offers tailored study planning, dynamic reminders, and adaptive progress tracking to support students in balancing their academic workload effectively. This inspired the creation of Passive Growing App , a smart productivity system designed to help students organize, prioritize, and achieve their academic goals.

Passive Growing App addresses several limitations of existing solutions. Unlike generic task managers, Passive Growing App focuses specifically on student productivity by combining a Smart Study Planner with a Smart Reminder System. The platform adapts to individual study habits and provides dynamic scheduling, helping students minimize procrastination and manage their time effectively. By incorporating intelligent task prioritization and progress tracking, Passive Growing App ensures that students stay on top of deadlines and academic milestones.

The key features of Passive Growing App include:

- **To-do list feature** that allows students to create personalized todo lists based on their tasks, priorities, and deadlines.

- **Habit Tracker feature** that the user can log in to daily about their productivity on that day to track progress.
- **Progress Graphing feature** to help students monitor their productivity over each day of the month in detail.
- **User-Friendly Interface** to provide seamless navigation and accessibility for students of all technical levels.
- **Data Privacy and Security** to protect user information and ensure a secure environment for productivity management.

By offering these quite specialized features specifically, Passive Growing App distinguishes itself completely from existing productivity tools through its sharp focus on the truly special individual needs of university students. It provides a planned as well as flexible system, assisting learners in beating delays, refining how they use time, in addition to meeting learning targets well. Ultimately, Passive Growing App aims to empower students with many tools they need to succeed in their studies. It also helps them develop better productivity habits for the future.

Chapter 2 Related Works

Several tools and platforms have been developed to help users manage their time, tasks, and productivity. These tools often focus on task scheduling, reminders, and progress tracking but lack the personalized features needed to address the unique academic demands of university students. A review of existing systems highlights their strengths, limitations, and areas where Passive Growing App can provide improved functionality tailored for students.

1. **Google Calendar**
   Google Calendar is a widely used platform for scheduling tasks and events. It allows users to set deadlines, create reminders, and organize tasks by time slots. (Google, n.d.). However, Google Calendar does not offer intelligent task prioritization or adaptive reminders that can adjust based on a user's progress or changing priorities.
   Drawbacks:

- Lacks personalized task prioritization
- No progress tracking features
- Does not adapt to individual study habits

2. **Microsoft To-Do**
   Microsoft To-Do enables users to create task lists and set reminders for upcoming deadlines. (Microsoft, n.d.) While it is simple and user-friendly, it provides only static reminders and task management without incorporating progress tracking or dynamic rescheduling. The tool does not cater specifically to the study needs of university students.
   Drawbacks:

- Static task reminders without adaptation

- No dynamic rescheduling based on progress
- Limited tools for academic planning

3. **Pomofocus.io**

   Pomofocus.io is a web-based application designed to help users implement the Pomodoro Technique, a time management method developed by Francesco Cirillo in the late 1980s. The technique uses a timer to break work into intervals, traditionally 25 minutes in length, separated by short breaks. (Pomofocus, n.d.)

   Drawbacks:

- Only available while connected to the internet
- Only conventional pomodoro timer (25 minutes, 5 minutes and 15 minutes) available
- No advanced reminders

Chapter 3 Design and Implementation

The productivity software is developed with a user-centric as well as designer-centric approach of simplicity, usability, and seamless integration with modern backend services and database services. It provides users with tools to manage their daily tasks, track their habits, have study timers and break timers and stay organized.

The software is all built with a modular design that enables each module, such as authentication, connection to database or plotting of graphs to operate independently while task management and habit tracking operate consistently. This design allows for easy expansion, allowing you to add new features or improvements without compromising existing features. The frontend built using Qt between devices such as user data time synchronization, user data to-do lists, and habit logs offers a responsive, cross-platform user interface that makes the application accessible on various operating systems such as Windows, macOS, and Linux. Important data such as authenticated email which also carries the authentication state is effectively shared between different parts of the application, improving both performance and user experience.

The application is designed to be extensible and future-proof with a focus on user needs and technological advancements. Push notifications or analytics can be added as features such as offline support, customizable user options, and integration with additional services such as separate backend servers can be added.

3.1 System Requirement Specifications

3.1.1 Software Specifications

- **C++**: C++ is the backbone of this productivity application, providing the performance and flexibility needed to build a robust and efficient system. It is used to define the application's structure, logic, and behavior, leveraging object-oriented programming principles like encapsulation and modularity. C++

handles everything from user authentication and data management to UI interactions, ensuring the application runs smoothly across different platforms. Its integration with the Qt framework and Firebase services makes it a powerful choice for developing feature-rich, cross-platform applications.

- **Qt**: Qt is a flexible framework used by this program to build the user interface and oversee application logic. It has a wide-ranging array of libraries with which to create reactive and attractive user interfaces, in addition to utilities by which to manage events, signals, and slots. Qt's cross-platform capabilities ascertain the application functions without problems on Windows, macOS, along with Linux. Within this program, Qt is used carefully for the detailed design of certain pages. Qt is also used with precision for the specific management of navigation among those pages with QStackedWidget.

- **Firebase**: Firebase serves as the backend for this application, providing essential services like user authentication and real-time data storage. Firebase Authentication handles user login and registration, while Firebase Firestore or Realtime Database stores user-specific data such as tasks and habit logs. The application communicates with Firebase using REST API calls, ensuring secure and efficient data synchronization. Firebase's scalability and real-time capabilities make it an ideal choice for managing user data in this productivity tool.

- **REST API**: REST API is the communication bridge between the application and Firebase. It allows the program to send HTTP requests (e.g., POST, GET, DELETE) to Firebase endpoints for tasks like user authentication, data storage, and retrieval. In this application, REST API calls are implemented using Qt's

QNetworkAccessManager, which handles network communication. This ensures that user data is securely transmitted and synchronized with Firebase, enabling real-time updates and seamless integration with the backend.

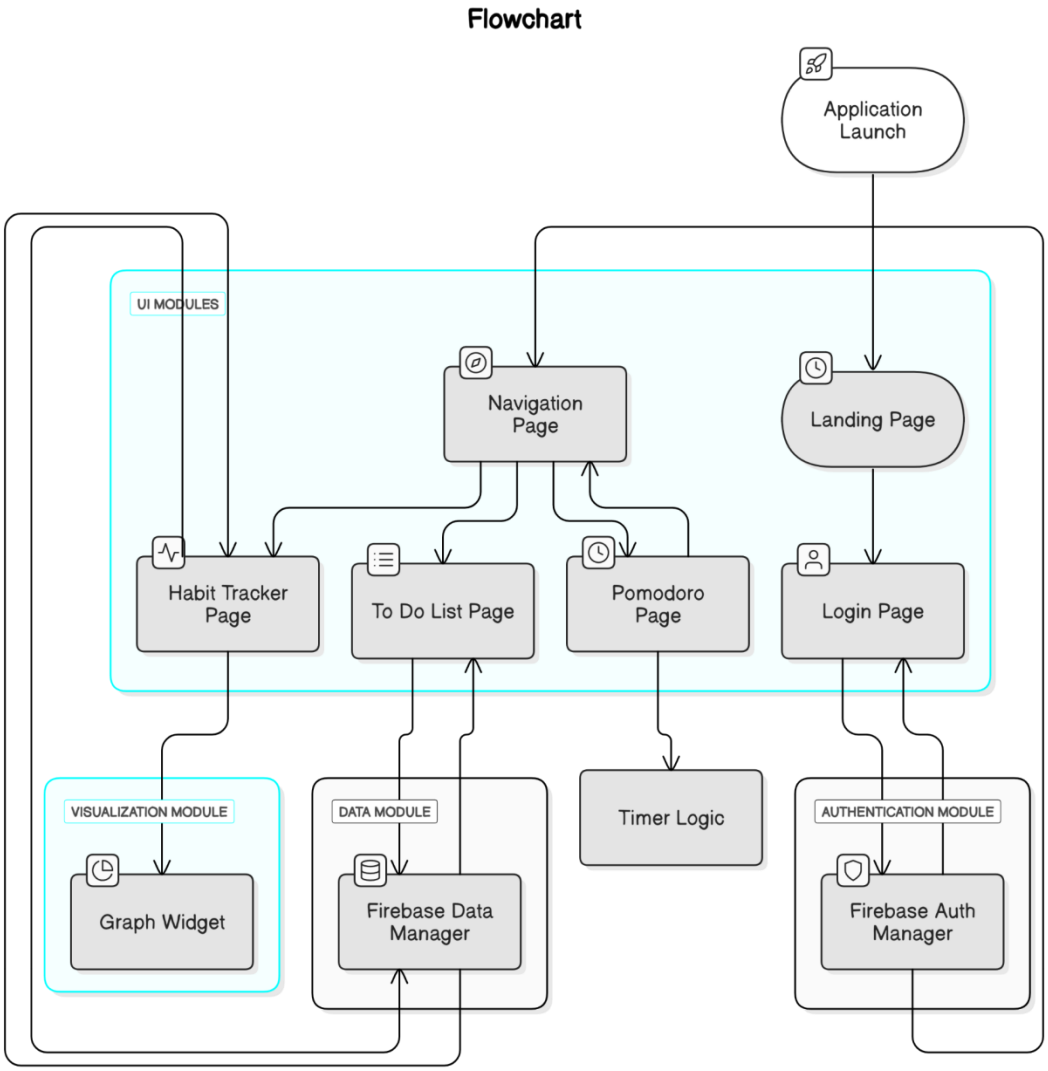


Figure 3.1.1 Flow Diagram For Integrated program

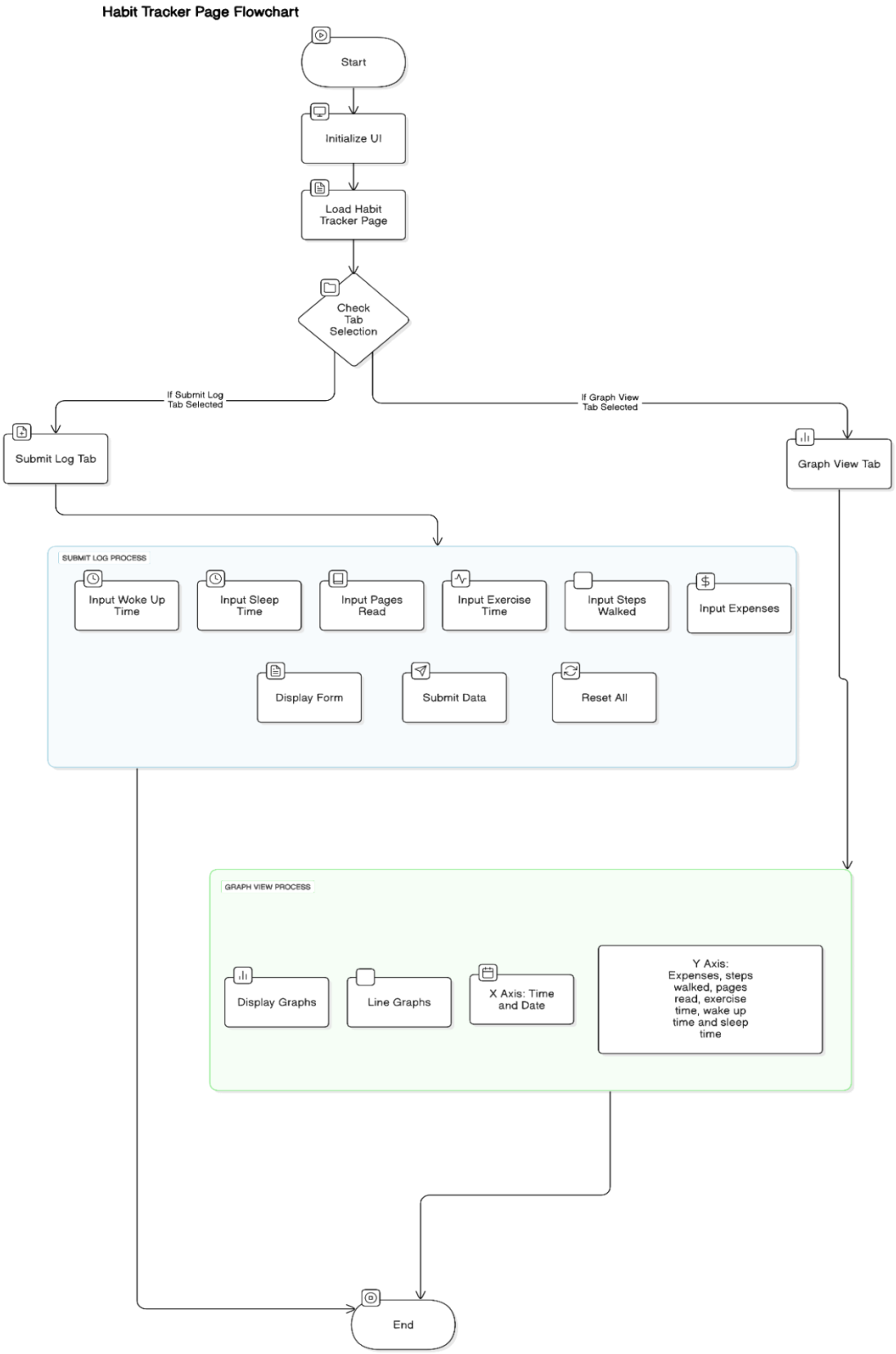


Figure 3.1.2 Flow Diagram For Habit Tracker Page

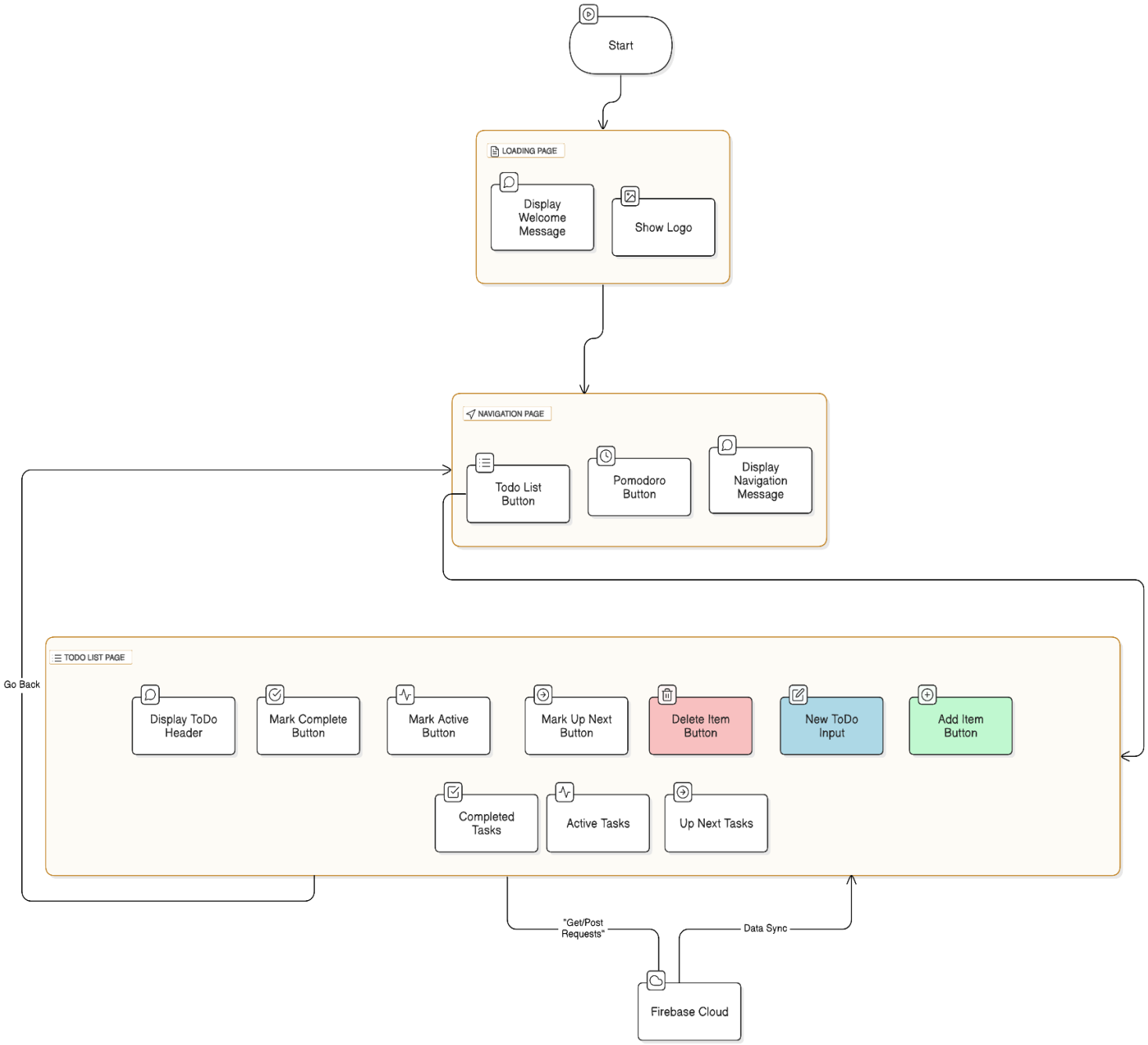


Figure 3.1.3 Flow Diagram For To-do List Page

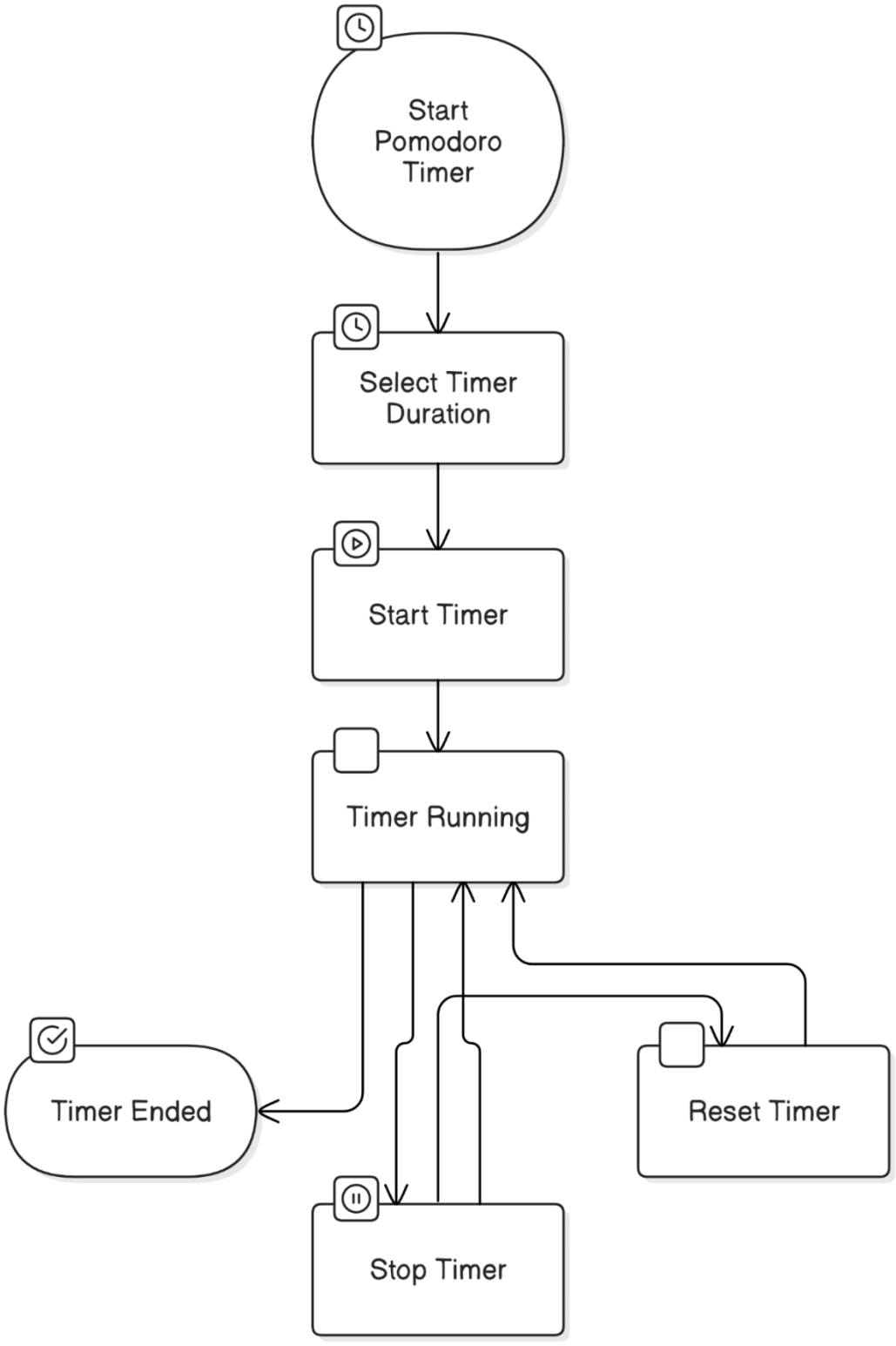


Figure 3.1.4 Flow Diagram For Pomodoro Page

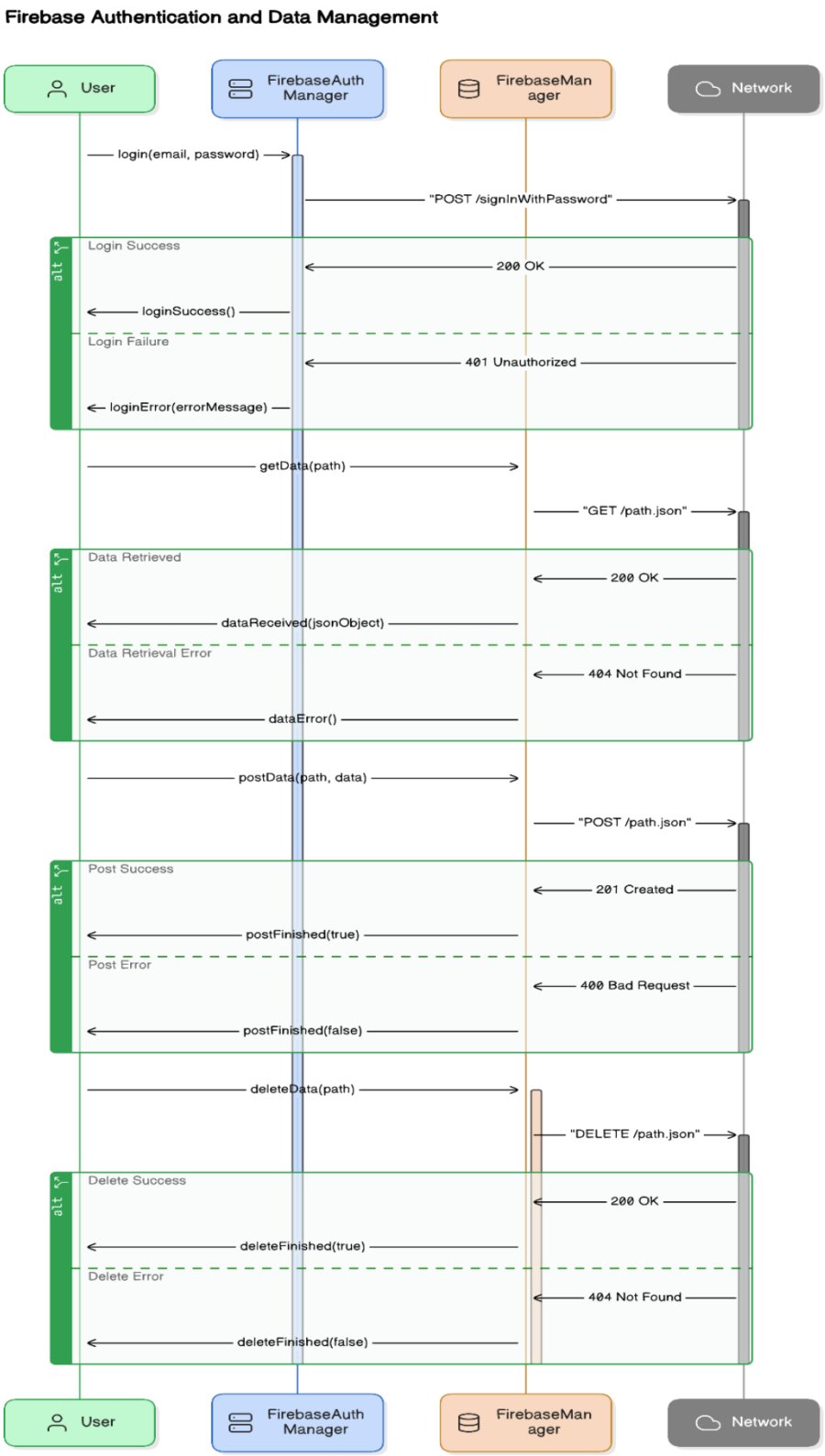


Figure 3.1.5 Flow Diagram For Database communication

3.1.2 Hardware Specifications

The main hardware devices used in the making of this project include Intel Core i7 13700HXprocessor with 16 cores and 24 threads, an Nvidia GForce RTX 4060 GPU, with 8GB of VRAM and 16GB of RAM.

Chapter 4 **Discussion on the achievements**

The project has successfully seen the completion of most of the objectives that we had set out to fulfill at the start of the project. We have successfully created a platform that combines multiple productivity systems to help students manage their academic schedules efficiently.

It enables users to create and manage todo-lists, pomodoro timer, habit and productivity tracker each with their own profile so that they can implement intelligent task prioritization features, enabling students to organize tasks based on deadlines, priorities, and study goals.

Throughout the development process, we became proficient in various technologies, including C++, QT, FireBase realtime database, Figma and UI design tools. We also deepened our understanding of how productivity apps work, how focus affects productivity and how data reflection can cause a positive feedback loop in students .

Some aspects did not follow the goals and must be noted. We first planned to include a calendar for scheduling but ended up not finishing it because we had little time and some technical problems. We first wanted to add push alerts to the app; later, we thought that would distract users, so we chose to work on raising the quality of the app experience.

The major features of Passive Growing App include:

4.1 User Authentication:

An essential part of the application is the login and register feature, which is made to give users safe access to their accounts. Through REST API calls, it verifies user credentials, including password and email, using Firebase Authentication. The

application sends a request to Firebase when a user submits their information, and Firebase verifies it and, if successful, returns an authentication token. While the user's email is kept in a singleton class for global access across all pages, this token is used to allow access to the application's features. By limiting access to personal information like to-do lists and habit records to authorized users, the login and register procedure improves user experience and security.

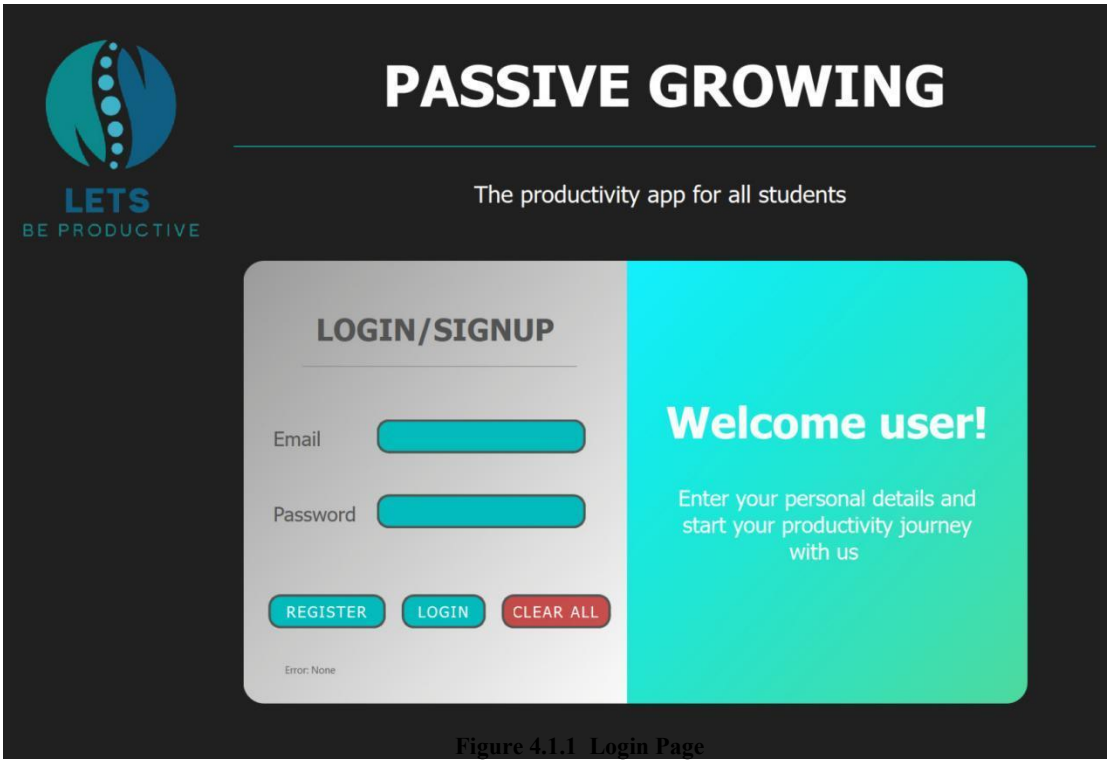


Figure 4.1.1 Login Page

4.2 Pomodoro Timer:

The pomodoro timer splits work times into fixed segments to help people focus and get tasks done. It follows set rules that assign 25 minutes of work then 5 minutes for a break. We have also allowed for other customizable time intervals as the users are all unique and have their own pomodoro preferences. It shows signals on screen plus sounds alerts when a work session or break nears its end while letting people run, stop

or set the timer again. Its core program that runs the timer with session states uses C++, the timer screen comes from Qt plus Qwidgets. This tool suits workers, pupils or every person aiming to work more by lowering interruptions urging careful work or easing even work routines.

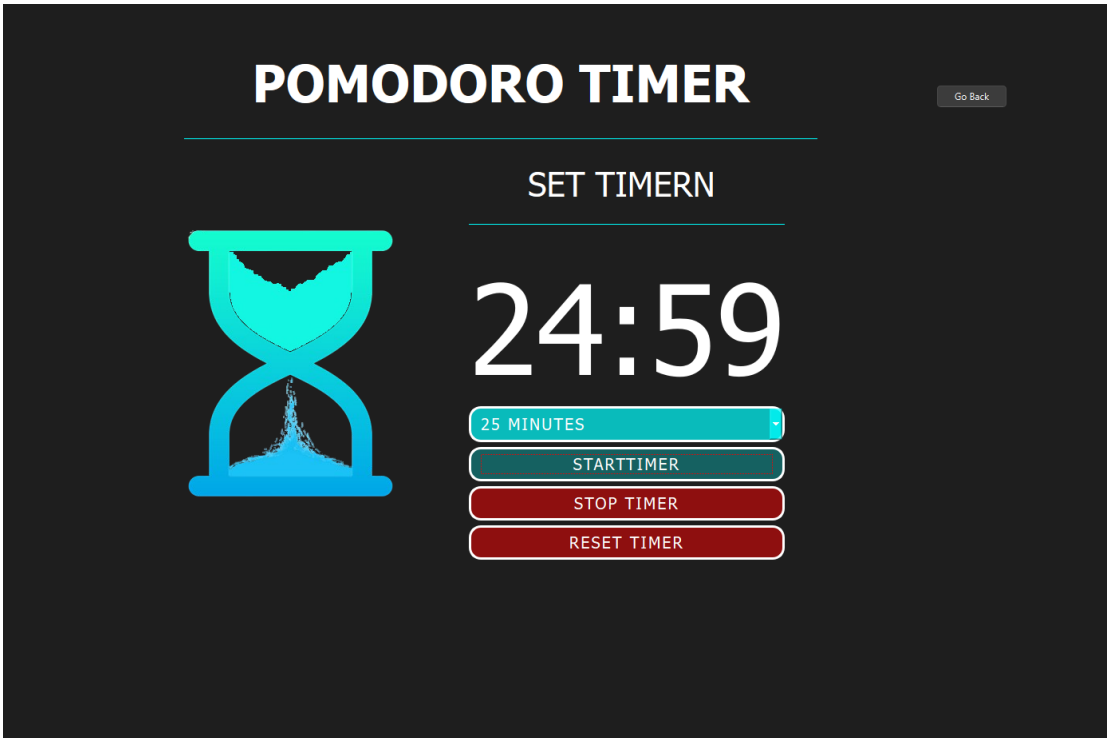


Figure 4.1.2 Pomodoro Page

4.3 To-do List:

The To-Do List tool helps manage daily tasks. Users can add tasks with a simple input box, which appears in a scrolling list. Users may mark tasks as done or remove them giving choice in handling tasks. Users can also decide what tag to give to the tasks (either active, completed or up next) and the tasks will be displayed in the UI accordingly. We used Firebase Realtime Database to keep tasks updated in real time, so users reach their to-do lists from any device. We used Qt for the clear user interface and C++ for the backend logic to boost productivity by guiding users to remain

organized and direct toward their aims. REST API calls send messages to Firebase making the to-do list a consistent and active resource.

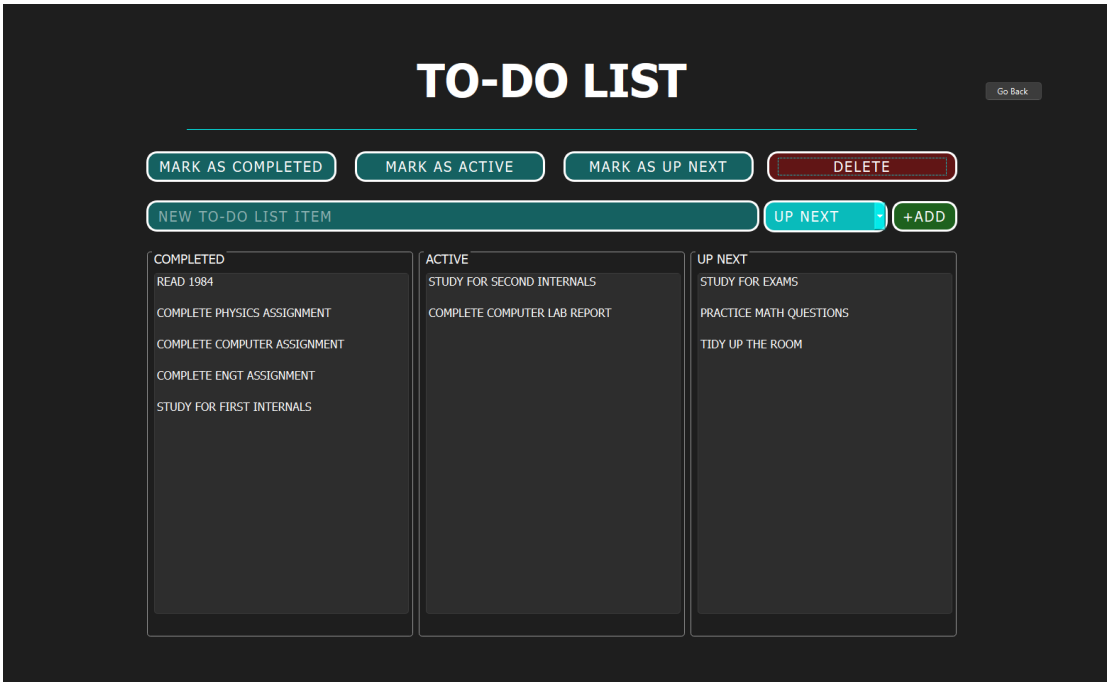


Figure 4.1.3 To-Do List Page

4.4 Habit Tracker Logging and Graphing:

By recording daily tasks such as wake times, sleep times, exercise duration, step counts, pages read. The Habit Tracker tool helps users build and keep healthy routines. Users fill in data on a clear form for quick synchronization, the tool stores it in Firebase Firestore or Realtime Database. A graphic displays progress over time by drawing line charts with Qt Charts. This picture assists a user in checking regularity noting patterns boosting drive to meet goals. The Habit Tracker improves self-control while building lasting habits. We used Qt so design stays simple employing C++ to save data and perform work.

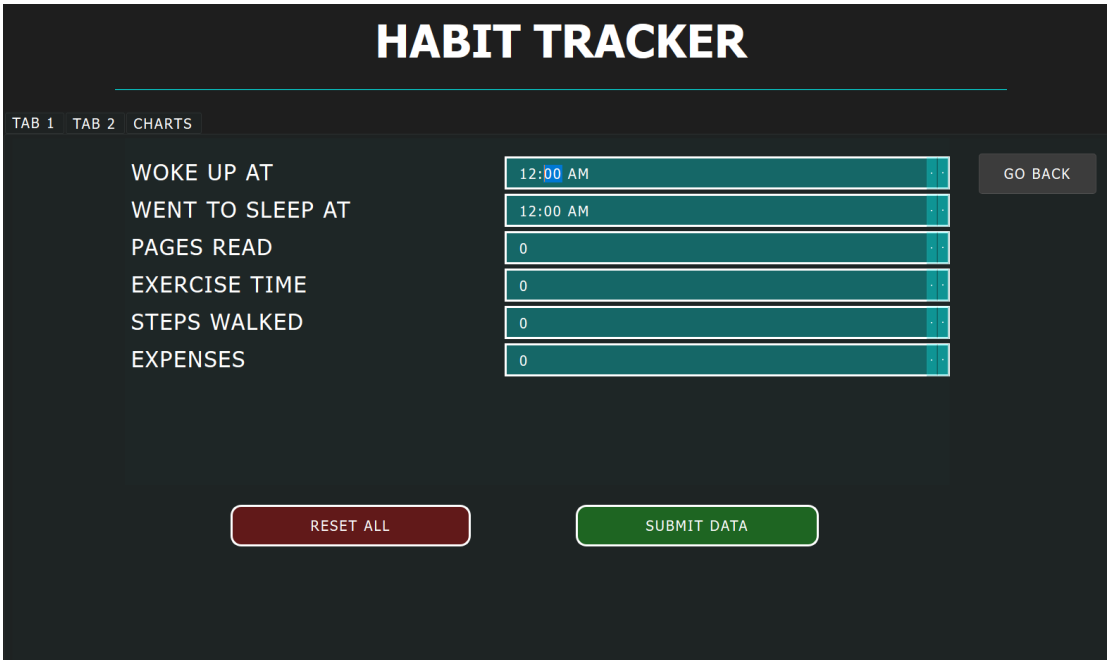


Figure 4.1.5 Habit Tracker Page

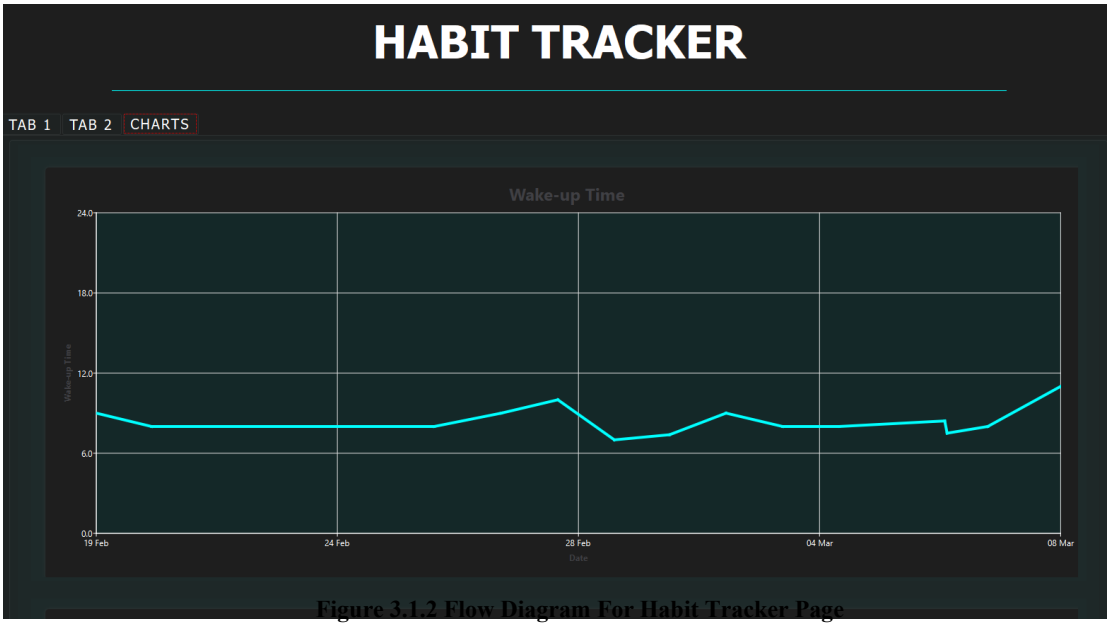


Figure 3.1.2 Flow Diagram For Habit Tracker Page

Figure 4.1.6 Pomodoro Page

Chapter 5 Conclusion and Recommendation

The productivity application successfully achieved its primary objectives, delivering a comprehensive platform that integrates task management, habit tracking, and time management tools to help users stay organized and productive.Passive Growing App made a good run as a productivity application that accomplished its core goals with a complete web application support including task management, habit-tracking, and time management tools all in one place to help users be organized and productive. Each component, such as the To-Do List as well as Habit Tracker, can function separately while also still blending smoothly in with the remaining application because of the modular architecture. The application offers a strong basis regarding productivity management. It still lacks third-party integrations and thorough analytics, so it has sufficient space intended for future improvements.

Although the application successfully accomplishes its primary objectives, time restrictions and the difficulty of scaling such functionality prevented the implementation of more sophisticated features like collaborative analytics for habit modification and integration with third-party productivity applications and the push notification and calendar features. In spite of this, the program offers a strong and intuitive experience for time management, habit tracking, and task management.

The project's scalable and effective system is the consequence of using Firebase for authentication and data storage, Qt for the cross-platform user interface, and C++ for the backend code. Each component, such as the To-Do List as well as Habit Tracker, can function separately while also still blending smoothly in with the remaining application because of the modular architecture. The application offers a strong basis regarding productivity management. It still lacks third-party integrations and thorough analytics, so it has sufficient space intended for future improvements. All things considered, the initiative effectively achieves its objectives by providing users with a dependable and entertaining platform to organize their work, develop wholesome routines, and boost productivity.

5.1 Limitations

The productivity application has two major limitations that could be addressed in future updates. First, it lacks offline functionality, meaning users cannot access or update their tasks and habits without an active internet connection. This dependency on Firebase for real-time data synchronization restricts usability in environments with poor or no connectivity. Second, the application does not support advanced analytics or collaborative features. While it provides basic logging and graphing for habits, it lacks deeper insights like trend analysis or personalized recommendations. Additionally, the absence of collaboration tools, such as shared to-do lists or group habit tracking, limits its utility for teams or families. Addressing these limitations could significantly enhance the app's functionality and user experience.

5.2 Future Enhancements

In order to provide offline support, the app will be able to use local storage such as SQLite or Qt's local storage APIs. So, they could view and manage reading and recording their tasks and habits without having to connect to the Internet. After revival of internet connection data could be synchronized with Firebase. Thus, we can make smooth transitions between on and offline modes. Likewise, adding a caching mechanism could help the app cache some of the fresh data and serve them locally to offer better performance and reliability when the connection is weak. We can also include offline and online mode as the user choice to include. For analytics, the machine learning libraries can be included in the application which will help to provide personalized analyses like habit trend analysis, predictive recommendations, and productivity scores. Things like enabling partnership through shared to-do lists and integrating group habit tracking or community challenges. Using Firebase's real-time database, these features allow users to collaborate without having to refresh.

Moreover, it would also be great to integrate third-party APIs (e.g., Google Calendar, Slack, or fitness trackers) to enable users to synchronize data across platforms and form a holistic productivity ecosystem. These are advanced features that can be done at a later stage.

References

Google. (n.d.). Google Calendar [Web application]. Retrieved from https://calendar.google.com

Atlassian. (n.d.). Trello [Web application]. Retrieved from https://trello.com

Microsoft. (n.d.). Microsoft To Do [Web application]. Retrieved from https://to-do.microsoft.com

Pomofocus. (n.d.). Pomofocus.io [Web application]. Retrieved from https://pomofocus.io

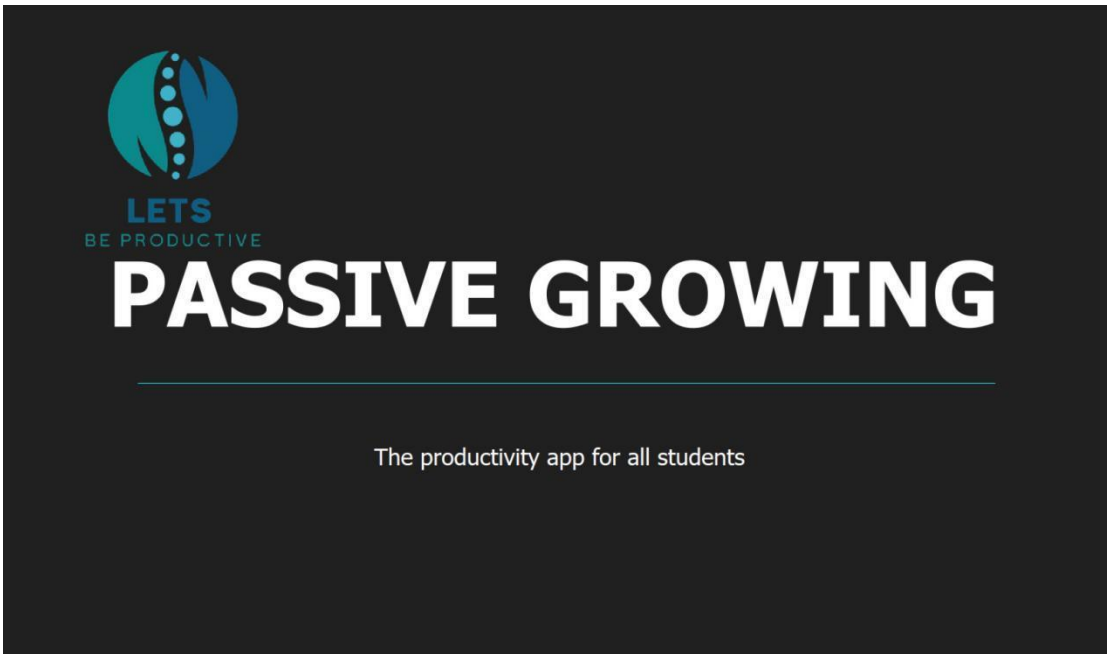
Bibliography

Kizilcec, R. F., & Halawa, S. (2021). The effect of tracking and reflecting on study habits on study behavior and grades. ResearchGate. https://www.researchgate.net/publication/352801064_The_Effect_of_Tracking_and_Reflecting_on_Study_Habits_on_Study_Behavior_and_Grades

Bellotti, V., Dalal, B., Good, N., Flynn, P., Bobrow, D. G., & Ducheneaut, N. (2004). What a to-do: Studies of task management towards the design of a personal task list manager. ResearchGate. https://www.researchgate.net/publication/221518959_What_a_to-do_studies_of_task_management_towards_the_design_of_a_personal_task_list_manager

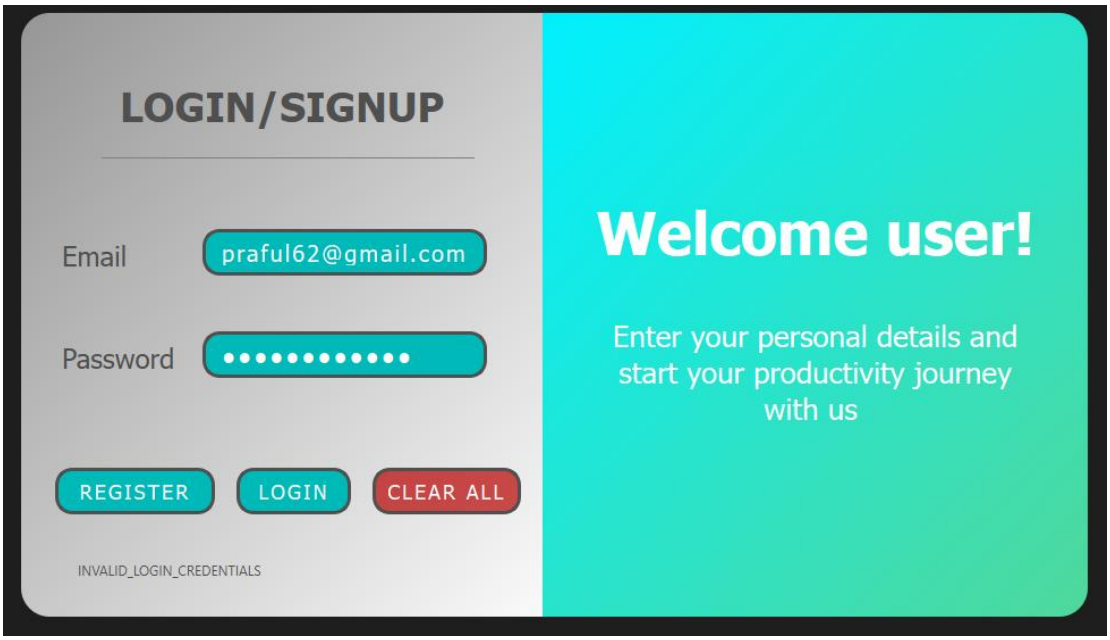
Kumar, A., & Sharma, R. (2020). A study on the impact of task management tools on productivity. International Research Journal of Modernization in Engineering, Technology, and Science (IRJMETS), 2(12), 5480. https://www.irjmets.com/uploadedfiles/paper/volume2/issue_12._december_2020/5480/1628083219.pdf

APPENDIX

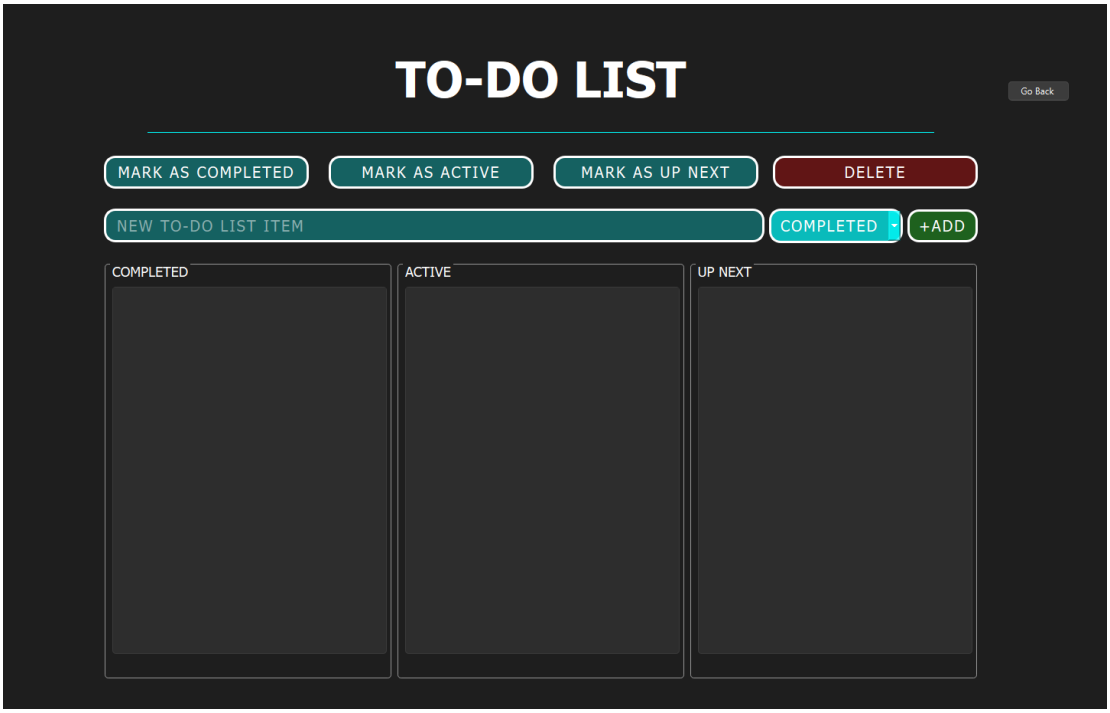


A 1. Landing Page UI

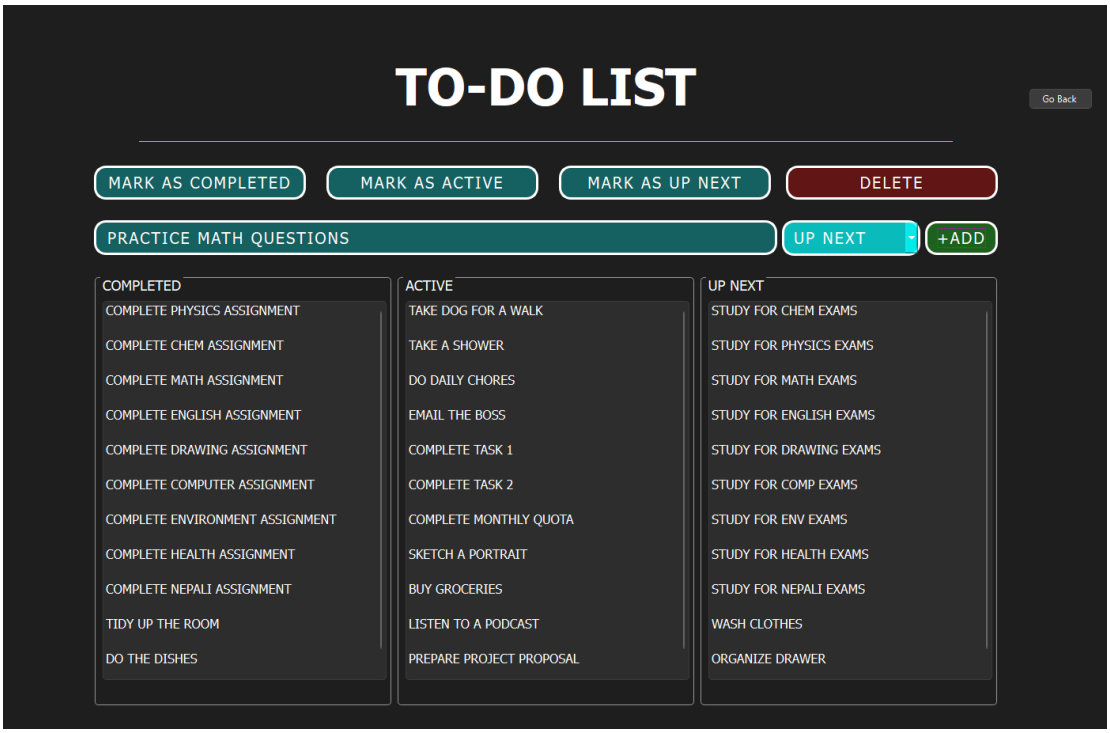
A 2. Authentication Page UI



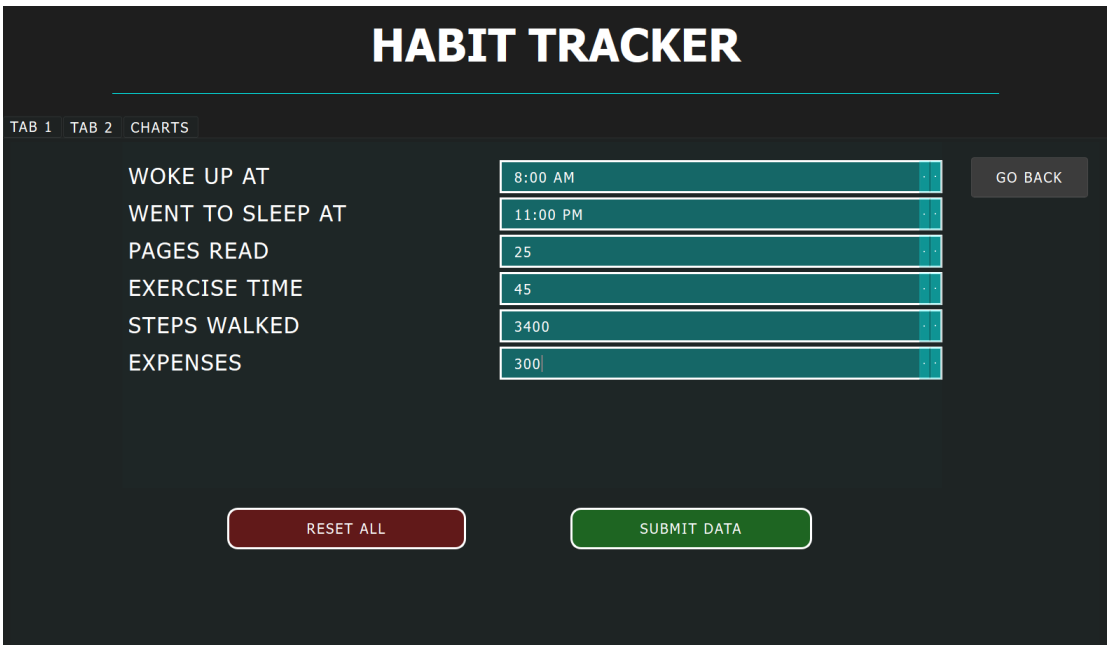
A 3. Sign Up Error Invalid Login Credentials



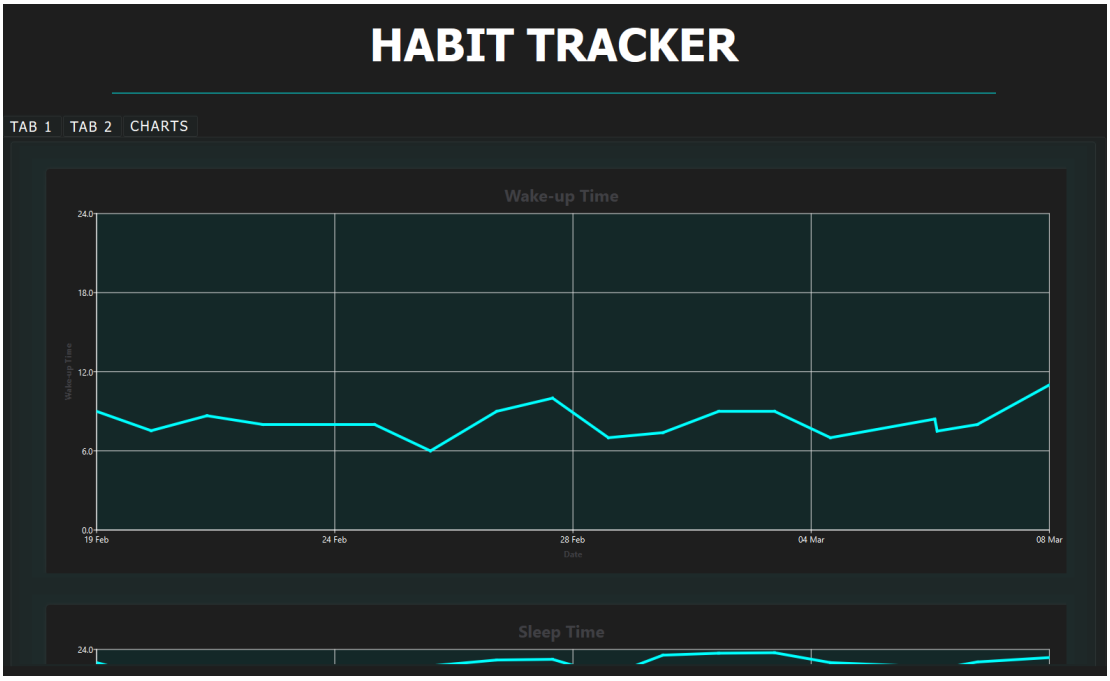
A 4. To Do List Empty UI



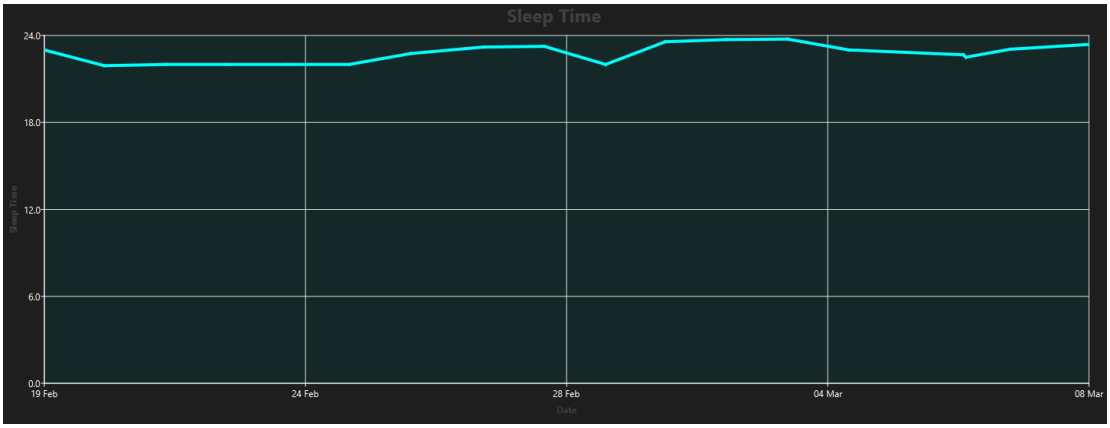
A 5. To Do List Complete UI



A 6. Habit Tracker UI

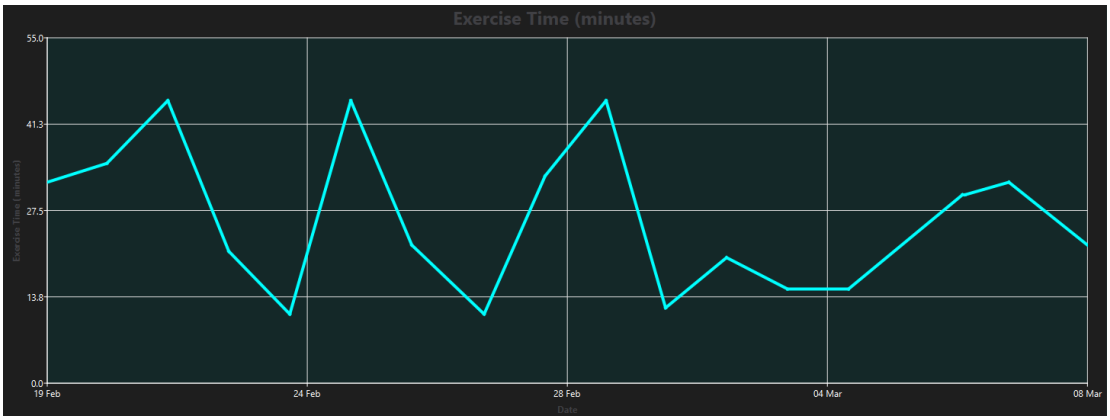


A 7. Habit Tracker Page Graph UI



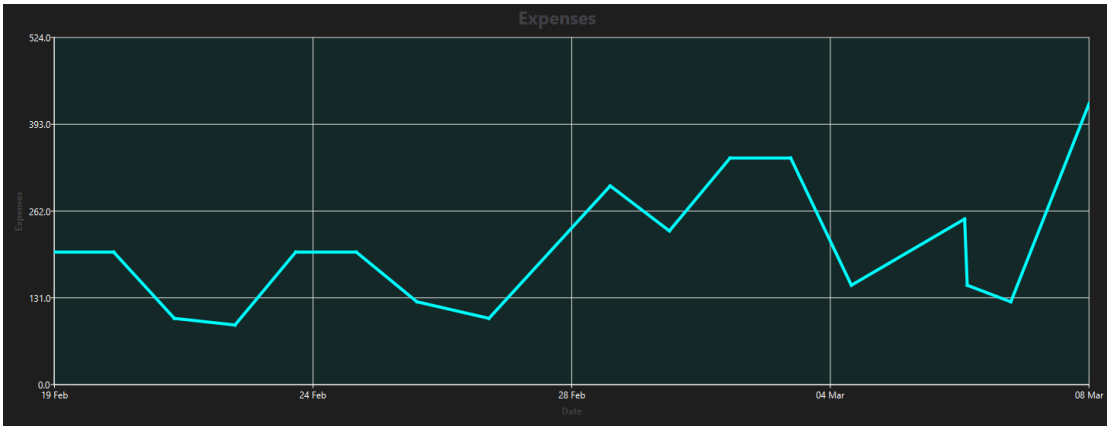
A 8. Sleep Time Graph

A 9. Pages Read Graph

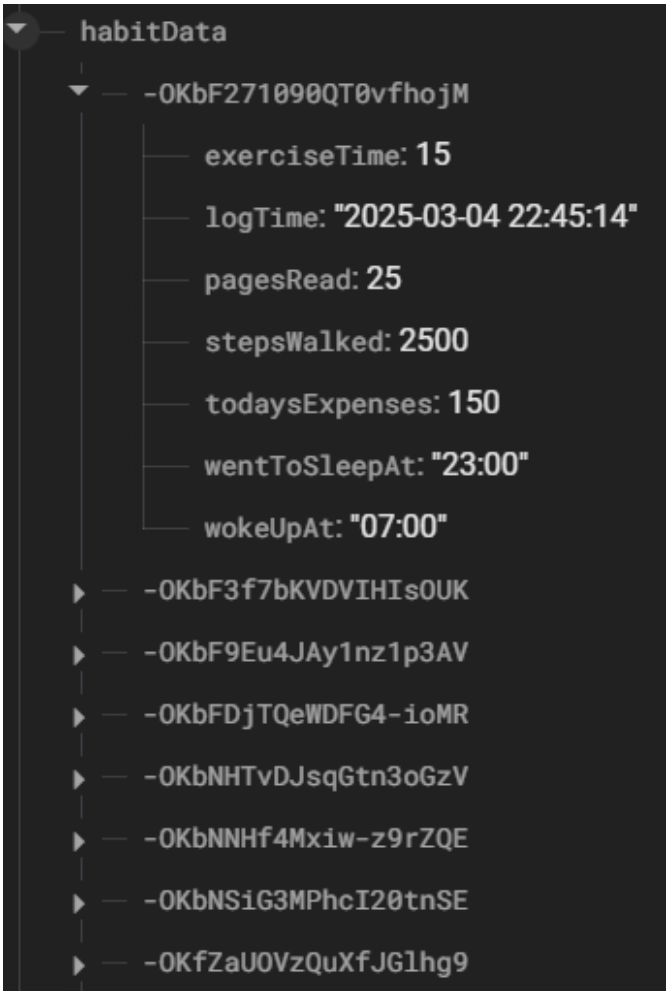


A 10. Exercise Time Graph

A 11. Steps Walked Graph



A 12. Expenses Graph



A 13. Habit Tracker Data Database Schema

A 14. To-Do List Data Database Schema