Interim Project Report

On

MindFlow – Task & Productivity Application

Submitted By:

Sajal Kanwal

12307408

Submitted To:

Mr. Aman Kumar

School of Computer Science and Engineering

Lovely Professional University

Phagwara, Punjab

Acknowledgement

The successful completion of this project would not have been possible without the invaluable support, guidance, and encouragement of several individuals. I would like to express my sincere gratitude to each of them for their contributions, assistance, and insights, which were instrumental in realizing the *MindFlow- Task & Productivity Application.*

First and foremost, I extend my heartfelt appreciation to my project advisor, *Mr. Aman Kumar*, whose expertise and patient guidance were indispensable throughout this journey. Their commitment to nurturing a thorough understanding of the subject matter provided a solid foundation upon which this project was built. Their constructive feedback, comprehensive review, and insightful suggestions were pivotal in refining this system and addressing challenges effectively.

I would also like to thank the faculty and staff of *School of Computer Science* for providing a supportive learning environment and access to the resources essential for this project's development. The facilities and academic resources provided by the institution played a crucial role in facilitating both the research and practical phases of this project.

A special note of gratitude goes to my colleagues and classmates who offered their encouragement and insights, engaging in fruitful discussions that often-sparked new ideas and improvements to the system. Their camaraderie and support fostered a collaborative atmosphere that enriched the development process.

Finally, I am immensely grateful to my family and friends for their understanding, patience, and encouragement throughout this endeavour. Their unwavering support has been a source of motivation and strength, allowing me to focus fully on the successful completion of this project.

To all those mentioned above and many others who contributed indirectly, I extend my deepest thanks. This project has been an enriching learning experience, made possible by your guidance and support.

Table of Contents

Introduction

In today's dynamic and fast-paced world, productivity tools have become indispensable for professionals and students alike, who navigate complex schedules and juggle multiple high-stakes responsibilities. Amid this landscape, the **MindFlow Task & Productivity Manager** emerges as a cutting-edge productivity application designed not just to organize, but to empower. Recognizing that efficiency, focus, and goal alignment are key drivers of success, MindFlow offers a robust, all-in-one solution that addresses the modern demands of personal and professional productivity.

MindFlow redefines time management by integrating essential productivity tools into a single, intuitive platform that doesn't just manage tasks—it optimizes them. The application includes a versatile suite of features, such as **task reminders, calendar management, note-taking, daily journaling, and productivity analytics**, each purpose-built to enhance workflow and productivity. These foundational tools are supplemented with specialized modules like the **To-Do List, Stopwatch, and Pomodoro Timer**—each crafted with precision to meet the unique needs of productivity-focused users. With each module intuitively structured, MindFlow eliminates friction, allowing users to focus on what truly matters.

This holistic design makes MindFlow an invaluable asset in overcoming the most pressing productivity challenges: structuring tasks effectively, setting priorities with clarity, and managing time with discipline. By providing a seamless user experience that enables users to track progress, define objectives, and celebrate milestones, MindFlow doesn't just assist—it drives users to reach their fullest potential. As a productivity solution that balances efficiency with an elegant, minimalistic interface, MindFlow aspires to be more than a tool. It aims to be a trusted companion that adapts to each user's unique workflow and goals.

MindFlow's modular architecture and professional-grade functionality position it as a powerful platform, with potential to rival leading productivity solutions. It's crafted to not only meet today's productivity needs but to anticipate the future, with a scalable design that can accommodate advanced features and integrations as the user's demands grow. For individuals and organizations striving to optimize performance and achieve excellence, MindFlow stands ready to transform productivity from a daily chore into a streamlined journey of progress, growth, and accomplishment.

Objectives and Scope of the Project

The *MindFlow Task & Productivity Manager* is designed with a clear set of goals: to create a comprehensive and accessible productivity application that integrates multiple functionalities into a unified platform. By focusing on task management, time tracking, and productivity enhancement, *MindFlow* aims to empower users to handle their personal and professional responsibilities efficiently and with ease.

1. Project Objectives:

- **Integrated Productivity Platform:** Develop an all-in-one productivity application that combines multiple essential tools—such as task management, time tracking, and productivity analytics—into a single, cohesive platform, simplifying the user's productivity workflow.
- **User-Friendly Interface Design:** Ensure that each module, including the To-Do list, Stopwatch, and Pomodoro Timer, features an intuitive and visually consistent interface. By enhancing usability and accessibility, *MindFlow* strives to deliver a seamless experience for users with varying levels of technical expertise.
- **Structured Time Management Techniques:** Implement robust time management features, such as the Pomodoro technique and Stopwatch timer, to help users manage their focus and energy effectively, ultimately reducing burnout and increasing productivity.
- **Data Security and User Privacy:** Establish a secure login system that protects user data, ensuring that personal information and productivity metrics remain confidential and accessible only to the user.
- **Incorporating User Feedback:** Continuously gather user feedback and insights to refine functionalities and improve the user experience in future versions, adapting the application to better meet user needs and preferences.

2. Scope of the Project:

The scope of *MindFlow* is extensive, covering the development of seven interconnected modules, each addressing a unique aspect of productivity. These modules provide core functionalities such as task management, calendar scheduling, time tracking, note-taking, journaling, and basic analytics. This cohesive suite of tools aims to support users in every stage of their productivity journey.

Key elements of the project scope include:

- **Module Development and Integration:** Each module, from task tracking to journaling, will be designed to seamlessly interact with others, creating a cohesive and integrated productivity experience. The current focus is on building these modules with reliability, usability, and user-focused design in mind.
- **Cross-Platform Compatibility:** Utilize Python's versatile libraries to ensure compatibility across different operating systems. This allows *MindFlow* to be accessible to a wide range of users, regardless of their device or OS.
- **Consistent Design Language:** Develop a unified and visually appealing design language across all modules, ensuring a cohesive look and feel that enhances the user experience and reinforces *MindFlow*'s professional aesthetic.
- **Foundation for Future Enhancements:** While the initial version provides fundamental productivity tools, future iterations could introduce advanced features, such as AI-driven personalization, enhanced data analytics, cloud sync for multi-device support, and collaboration capabilities for team-based productivity.

MindFlow is poised to evolve into a powerful productivity tool that helps users manage their time, tasks, and goals efficiently.

Application Tools

The MindFlow project leverages a carefully selected set of tools, libraries, and architectural components to create a robust, intuitive productivity application. This section details the specific purpose each technological component serves in MindFlow, explaining how they contribute to the app's functionality, user experience, and performance.

1. Programming Language: Python

Python is the core programming language for MindFlow, chosen for its readability, versatility, and extensive library support, which enables efficient development and prototyping. Python's role in MindFlow extends to facilitating modular architecture, allowing each feature (such as the To-Do list, Pomodoro Timer, and Stopwatch) to be developed independently. This modularity not only enhances maintainability but also simplifies future expansion or updates to the application's features. Python's robust error-handling capabilities are also integral to providing a seamless user experience, as they help prevent crashes or disruptions when handling unexpected inputs or managing data files. Moreover, Python's ecosystem offers a range of libraries that support essential functionalities, including Tkinter for GUI design, which allows for a streamlined,

visually appealing interface, and sqlite3 for handling local data storage, securing user tasks and productivity metrics within the app.

Python's Role in *MindFlow*:

- **Modularity**: Python's support for modular programming enables each feature in *MindFlow* (like the To-Do list, Pomodoro Timer, and Stopwatch) to be built independently, facilitating a highly maintainable codebase.
- **Error Handling**: Python's robust error-handling capabilities allow the application to manage exceptions effectively, providing a smoother user experience even in cases of unexpected data input or file access errors.
- **Extensive Libraries**: Python's ecosystem includes powerful libraries, especially for GUI design and data management, which significantly enhance *MindFlow's* core functionality and reduce the need for external dependencies.

2. Integrated Development Environments (IDEs)

Different IDEs were employed to streamline the coding, debugging, and testing phases of MindFlow, each tailored to specific needs in the development lifecycle:

- **PyCharm**: As the primary IDE for the project, PyCharm is used extensively for the core development of MindFlow's features. Its intelligent code completion, integrated version control, and powerful debugging tools streamline the creation of complex modules like the Pomodoro timer and productivity analytics. PyCharm's extensive support for Python modules allows the development team to rapidly test, iterate, and debug functionalities across MindFlow, ensuring high code quality and reliability.
- **Visual Studio Code**: Visual Studio Code is used as a flexible editor for quick adjustments and feature refinements within the application. With extensions for Python linting, real-time debugging, and Git integration, Visual Studio Code complements PyCharm by enabling rapid iteration on minor improvements, such as UI adjustments or code enhancements across modules. Its lightweight setup supports the efficient development of user interface elements and minor logic changes, making it especially useful for collaborative development and minor, quick-fix implementations in MindFlow.

3. Libraries and Packages

Several essential Python libraries power *MindFlow*, each serving a key role in the project's design, functionality, and user interface:

- **Tkinter**: Tkinter is the core library for *MindFlow's* graphical user interface. This standard GUI toolkit for Python provides the framework for creating windows, buttons, labels, and other interface elements. Tkinter's simple yet flexible structure enables *MindFlow* to deliver an intuitive layout while keeping the application lightweight and responsive.
- **ttkbootstrap**: An advanced theme extension for Tkinter, ttkbootstrap allows *MindFlow* to adopt modern, polished themes, significantly enhancing the visual appeal. By adding design consistency and usability, ttkbootstrap helps the application align with contemporary UI standards, making each module visually cohesive and user-friendly.
- **JSON**: Currently, *MindFlow* employs JSON files to manage data storage. JSON's text-based format makes it ideal for storing structured data, such as to-do lists, timer settings, and journal entries, without the need for complex data handling. JSON files also allow straightforward data retrieval and updates, supporting each module's functionality with minimal overhead.

JSON in Action in *MindFlow*:

Each module stores its specific data in a dedicated JSON file. For example, the To-Do module saves tasks in a file where each entry is a JSON object containing task properties such as title, due date, and completion status. The data is serialized to JSON format when saved, enabling easy loading and updating whenever the application initializes or changes are made.

Code Snippet from To-Do Module:

```
def load_tasks(self):
    try:
        with open("tasks.json", "r") as f:
            data = json.load(f)
            for task in data:
                text = task.get("text", "")
                status = task.get("status", "Pending")
                self.task_tree.insert("", "end", values=(text, status))
    except FileNotFoundError:
        pass
    except json.JSONDecodeError:
        messagebox.showerror("Error", "The tasks file is corrupted. Starting with an empty task list.")
```

This snippet highlights how JSON data is read into *MindFlow*, restoring previous states. Error handling ensures that corrupted or missing files do not crash the application, notifying users instead.

4. Integrated Modules

The core functionality of *MindFlow* is divided into several interlinked modules, each with a unique purpose. Below is a breakdown of each module and how it operates at a technical level.

- To-Do List Module: This module allows users to create, view, update, and delete tasks. The Listbox widget in Tkinter displays tasks, while a JSON file acts as a persistent data store. Task entries are saved as JSON objects with properties such as "name," "deadline," and "priority," allowing for straightforward data manipulation and filtering.
- Pomodoro Timer Module: The Pomodoro timer provides a time-blocking tool that helps users focus for set intervals. It uses Tkinter's Label and Button widgets to display countdowns and control the timer. The application records each completed Pomodoro session in a JSON file, capturing details like session duration and timestamp, which can later be analysed to improve user focus habits.
- Stopwatch Module: The Stopwatch allows users to track the time spent on specific tasks or activities. Tkinter's Label widget is used to display elapsed time, while the Button widget controls start, stop, and reset functions. This module is helpful for users who want to measure the time they invest in individual tasks.
- Notes and Journal Module (planned): This module will allow users to store daily reflections or notes related to their tasks and productivity. The current implementation plan is to save journal entries in JSON format, categorized by date. This module will provide a daily log that users can refer back to, helping them track their progress and motivation over time.

5. Version Control: Git

Git is used for version control throughout the *MindFlow* project, facilitating collaboration, tracking changes, and ensuring code stability. By employing Git, the development team maintains a clear history of modifications, simplifies code reviews, and efficiently manages feature branches, making the process of adding new modules and features smoother.

Benefits of Using Git in *MindFlow*:

- **Branching and Merging**: Developers can work on different modules or features in isolated branches, merging updates once thoroughly tested.

- **Code Backup**: Storing the code in a remote repository ensures redundancy and provides a backup of the project's progress.
- **Collaboration**: Git's capabilities allow multiple team members to contribute without conflict, essential for a multi-module application like *MindFlow*.

6. Current Data Storage Method: JSON

At present, *MindFlow* uses JSON files for data storage across its modules. This approach provides a lightweight, accessible format that supports easy read-write operations, suitable for handling structured data such as tasks, timer settings, and journal logs.

How JSON Supports *MindFlow's* Modules:

- **Task Management**: JSON objects store task attributes, making it easy to retrieve, update, and sort tasks within the To-Do module.
- **Pomodoro Timer Logs**: Each completed Pomodoro session is stored as a JSON entry, recording details like start and end times. This data allows users to review productivity patterns over time.

While JSON serves as an effective initial data storage solution, limitations include difficulty in handling complex relationships and lack of concurrency control. Thus, the project is set to transition to more structured data storage methods as it matures.

MindFlow's architecture, built on Python and bolstered by Tkinter, ttkbootstrap, JSON, and Git, establishes a strong foundation for an effective productivity application. The choice of tools reflects a balance between usability, modularity, and scalability, ensuring that *MindFlow* delivers a streamlined user experience while supporting future growth. As development progresses, enhancements to the data management layer will increase the application's capability to handle complex productivity tasks and securely manage user data. These structured upgrades will contribute to *MindFlow's* evolution into a comprehensive, reliable, and adaptable tool for improving productivity in both personal and professional contexts.

Project Design

The design of *MindFlow* is the core framework that underpins its functionality, user experience, and effectiveness as a productivity application. This section provides an in-depth view of the architectural elements, functional flow, and the role of each module in creating a cohesive, robust system that meets the demands of modern productivity needs. A meticulously crafted design ensures that *MindFlow* not only delivers seamless task management and time tracking but also provides a harmonious and engaging user experience.

1. Application Architecture

The architecture of *MindFlow* is modular, each component designed to work independently while interacting cohesively within the larger system. This modular approach enables the application to remain flexible, supporting scalability and the potential addition of future features. Key architectural principles include:

- **Modular Design**: Each core feature—To-Do List, Pomodoro Timer, Stopwatch, and Calendar—operates as a discrete module with its own logic, user interface, and data storage protocol. These modules can be developed, tested, and refined in isolation, contributing to a streamlined development process and enhancing overall reliability.
- **Centralized Data Handling**: A centralized data manager oversees data storage and retrieval, ensuring that each module accesses consistent information, whether it's user settings, task lists, or session history. This unifying component minimizes redundancy and streamlines data flow throughout the application.
- **Layered Design Pattern**: Adopting a layered design structure ensures a clear separation between the user interface, business logic, and data management layers. This separation simplifies code maintenance and allows for independent development of the application's functional and visual elements.

2. User Interface (UI) and User Experience (UX) Design

The UI/UX design is fundamental to *MindFlow's* identity, focusing on ease of use, visual appeal, and accessibility. A consistent design language is maintained across all modules, which makes navigation intuitive and ensures that the user can interact with the application without distraction.

- **Minimalist Aesthetic**: Using ttkbootstrap with Tkinter as the base, *MindFlow* employs a modern, minimalist design with a cohesive color scheme that aligns with professional standards. This minimalist approach reduces visual clutter, focusing attention on key elements to promote engagement and ease of use.

- **Interactive Elements**: Each module includes interactive elements like checkboxes, toggles, and calendar dates to streamline user input. Animations and visual cues guide users through each action, enhancing the UX and ensuring that interactions feel responsive and intuitive.
- **Accessibility Considerations**: The design prioritizes accessibility, using high-contrast colors for readability and providing large, clear buttons for ease of use. Keyboard shortcuts and tooltips are integrated to assist all user types, ensuring *MindFlow* remains inclusive and accessible.

3. Module Design

Each module within *MindFlow* is purpose-built to serve a distinct productivity function, contributing to a holistic system. Below is an in-depth look at each module and how they interrelate:

3.1. To-Do List Module:

- The **To-Do List** module in MindFlow is tailored for effective task management, enabling users to organize, prioritize, and complete tasks in a streamlined, intuitive interface. Designed to be both functional and accessible, the To-Do List module encourages users to break down their responsibilities into manageable items, helping them focus on daily, weekly, and long-term objectives.
- **Functional Flow**: This module allows users to add, edit, delete, and mark tasks as complete. Each task can be categorized by priority level, due date, and category tag, providing clarity and structure. The interface supports a "Today" view for urgent tasks and a broader "All Tasks" view, allowing users to stay organized and visualize upcoming deadlines. A sorting feature enables users to arrange tasks by priority, due date, or status, ensuring they can quickly access and focus on their most pressing items.
- **Data Integrity**: Tasks are stored in a JSON format that captures essential details like task name, description, due date, priority level, and completion status. This setup ensures that data is preserved reliably across sessions and can be quickly accessed by other modules if needed. JSON's adaptable nature also allows for future enhancements, such as subtasks, recurring tasks, or integration with calendar functions, ensuring the To-Do List module remains robust and scalable.
- **Value Proposition of To-Do List:** The To-Do List module anchors users' daily productivity by giving them clear control over their tasks and responsibilities. By simplifying task management and helping users track progress, it reduces overwhelm and enhances focus, embodying MindFlow's dedication to practical, user-centric solutions that empower users to achieve their goals with confidence.

3.2. Pomodoro Timer Module:

- The **Pomodoro Timer** module is a productivity-enhancing tool based on the Pomodoro Technique, which promotes focused work intervals with scheduled breaks to prevent burnout and improve concentration. The Pomodoro Timer is designed for users aiming to enhance focus, manage time effectively, and maintain a balanced work routine.

- **Functional Flow**: The Pomodoro Timer features customizable work and break intervals, typically set to a 25-minute work session followed by a 5-minute break, with a longer break after several cycles. Users can adjust the interval settings to match their preferences. A notification or sound alert signals the beginning and end of each session, helping users stay on track and transition between focused work and rest seamlessly.

- **Data Synchronization**: Pomodoro sessions are stored in JSON format, recording details such as session start times, total number of sessions, and break intervals. This format enables flexible data storage and can be leveraged to generate insights into work patterns and productivity trends. Looking ahead, data collected in this module may support future analytical features, such as productivity graphs or trend analyses, to offer deeper insights into user habits.

- **Value Proposition of Pomodoro Timer:** The Pomodoro Timer embodies the MindFlow mission by combining structure and flexibility, allowing users to enhance focus while prioritizing regular mental breaks. By fostering a sustainable work routine, this module helps users maintain productivity over time, balancing intensive work sessions with restorative breaks, and creating an environment conducive to high-quality, consistent performance.

3.3. Stopwatch Module:

- The **Stopwatch** module in MindFlow is a dedicated time-tracking tool that empowers users to measure and manage the time spent on specific activities, promoting greater awareness and control over productivity. Designed for users who wish to track time-intensive tasks—whether for study sessions, project work, or exercise—the Stopwatch module offers precision and simplicity within an intuitive interface.

- **Functional Flow**: The Stopwatch provides essential functionalities: start, pause, resume, and reset, presented in a straightforward and user-friendly layout. Time is displayed in hours, minutes, and seconds, allowing users to monitor their progress with high accuracy. Additionally, users can label each tracked session, making it easier to identify and categorize activities. This feature supports detailed tracking and reflection, enabling users to analyse how time is distributed across various tasks.

- **Data Logging**: The Stopwatch module uses JSON for logging time data, capturing information such as start and end times, duration, and optional labels for each session. This lightweight and structured approach to data logging preserves session information across application usage and

allows for easy retrieval when users wish to review or analyse their time records. While JSON is sufficient for the current phase, future versions may incorporate a more advanced data storage solution, facilitating integration with other productivity modules for seamless data sharing and broader analytics capabilities.

- **Value Proposition of Stopwatch:** The Stopwatch module provides users with an accurate and accessible means of tracking time, promoting a mindful approach to time management. By helping users visualize and log time spent on each activity, this module aligns with MindFlow's broader mission of fostering productivity through well-structured tools, enabling users to refine their focus, analyse their efficiency, and take deliberate steps toward optimizing their daily routines.

3.4. NotesHub Module:

- The **NotesHub** module serves as a cornerstone in the MindFlow project, functioning as a comprehensive repository for capturing, organizing, and retrieving notes and ideas in real-time. Positioned as the project's information nucleus, NotesHub enables users to manage thoughts and data with both ease and flexibility. This module is designed not just as a simple note-taking feature but as a structured environment for brainstorming, meeting notes, class summaries, and task-specific documentation—offering users a centralized space to organize their ideas with precision.

- **Functional Flow**: NotesHub supports a rich-text editor that allows users to create and format notes with a range of styling options, such as bolding, italicizing, adding lists, and inserting links. These options ensure that critical information stands out and that notes are easy to navigate and reference. Users can tag and categorize each note, enhancing organizational clarity, while an efficient search function enables quick retrieval of specific notes by keywords, dates, or tags. Additionally, the module includes a pinning feature, allowing users to prioritize high-importance notes, making it ideal for ongoing projects or frequently referenced information.

- **Data Storage and Retrieval**: NotesHub employs a structured JSON format for data storage, providing a balance between flexibility and organization for user content. JSON's adaptability allows each note's data—such as title, content, tags, timestamps, and priority status—to be stored in an accessible manner. This format supports future enhancements, such as optimized search functionality and complex filtering, without compromising the existing data structure. As MindFlow progresses, the NotesHub module is designed for a seamless migration to more advanced database systems, facilitating larger-scale data management and cloud integration to ensure continued scalability and reliability.

- **Value Proposition of NotesHub:** NotesHub's role as the project's central information hub is pivotal to MindFlow's objective of delivering a professional-grade productivity solution. By providing a thoughtfully designed, versatile module, NotesHub transforms note-taking into a dynamic and structured experience. Users can organize their knowledge base, access critical details swiftly, and manage complex projects with ease. This module reflects MindFlow's

commitment to creating an impactful, reliable productivity platform that addresses the multifaceted demands of today's users.

4. Data Management

Currently, *MindFlow* employs JSON for data storage across modules, but as the application scales, more structured and reliable storage solutions will be implemented. Each data structure is designed to simplify data access and manipulation within the modules while ensuring data integrity and user privacy.

- **JSON as a Transitional Solution**: JSON is lightweight and ideal for early-stage development. It simplifies serialization and deserialization, allowing for straightforward data retrieval and updates across modules. However, JSON's limitations in scalability and concurrency highlight the need for a more advanced system.
- **Future Database Implementation**: As the user base grows and demands increase, transitioning to a relational or NoSQL database (such as SQLite or MongoDB) will offer improved data integrity, support for concurrent access, and optimized query performance. A database will also simplify data recovery, backup, and syncing processes, supporting cross-device compatibility.

5. Security and Privacy Design

Ensuring data security and user privacy is paramount in *MindFlow's* design. Key design features include:

- **User Authentication**: *MindFlow* implements a secure login system to ensure only authorized access. Credentials and sensitive data are stored securely, with future plans to implement encryption for added protection.
- **Data Encryption**: As part of the planned transition to a database, data encryption protocols will be introduced to secure personal information and usage history. This ensures compliance with data protection standards and fosters user trust.
- **Error Handling and Data Validation**: The design incorporates extensive error handling and data validation to safeguard data accuracy and prevent corrupt data from entering the system. This reduces the risk of system crashes or data loss, contributing to a stable and reliable user experience.

The design of *MindFlow* reflects a commitment to providing users with an integrated, efficient, and user-friendly productivity solution. Each component of the design—from the modular architecture and user-centric interface to secure data management and privacy features—contributes to an application that adapts to users' evolving needs. As *MindFlow* continues to develop, future enhancements will ensure scalability, improved data management, and heightened security, solidifying its role as a comprehensive productivity tool in both personal

and professional settings. Through its intricate and thoughtfully constructed design, *MindFlow* demonstrates its potential to become an indispensable resource for productivity management.

Flowchart

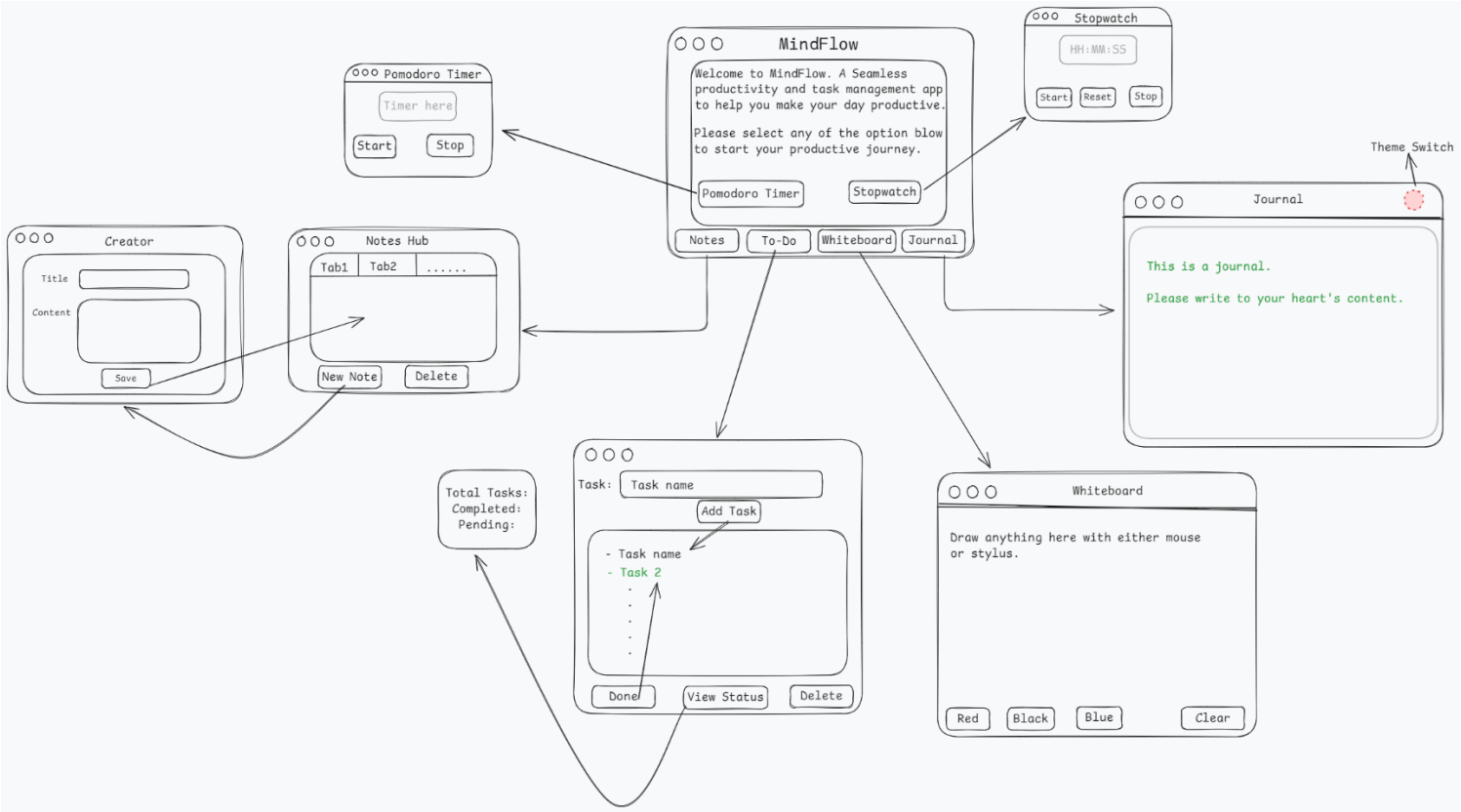

Figure 1: MindFlow – Task & Productivity Application's Blueprint

Screenshots of Application

Authentication

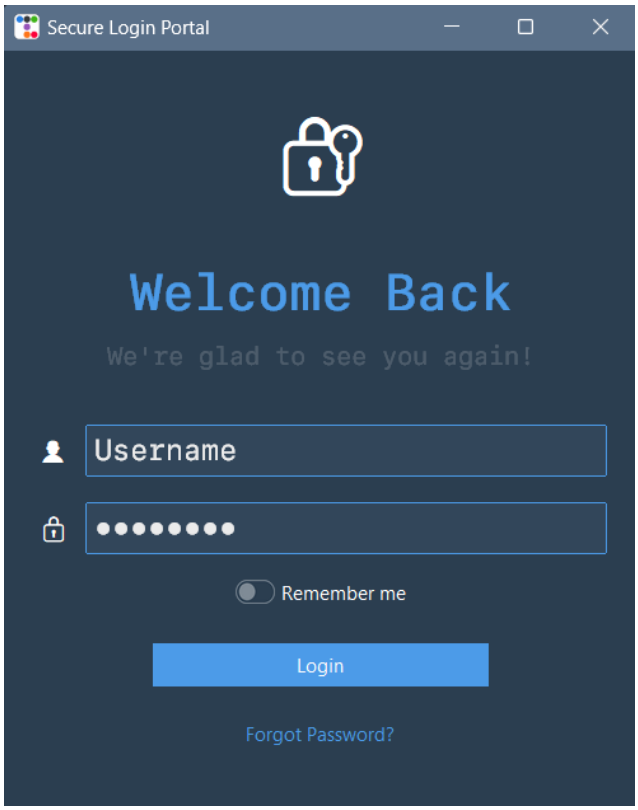

Figure 2: Login Window: This is the first module user interacts with and adds their valid credentials to further access and interact with the application.

After successful login

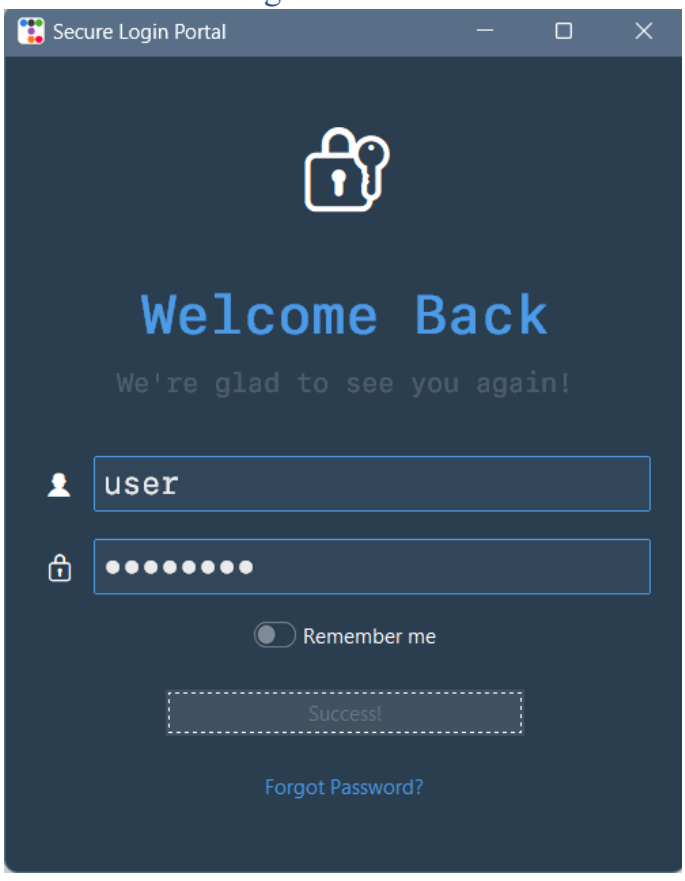

Figure 2.1: Successful Validation Window: After the credential are authenticated, the user is routed to the landing page.

Landing Page

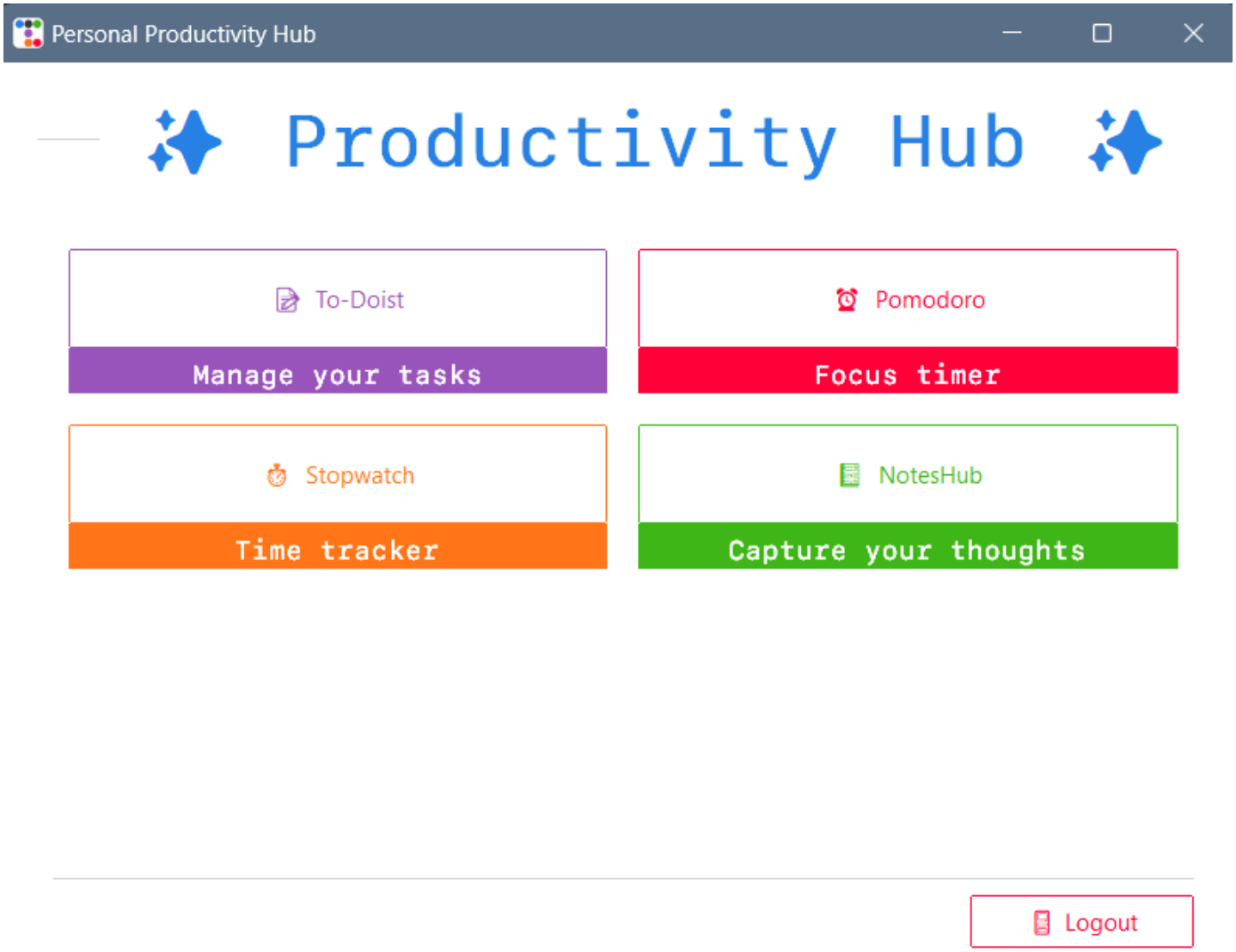

Figure 3: Landing Window: The main routing section of the application which handles all the integrated modules of the application along with their tooltips.

To-Do List

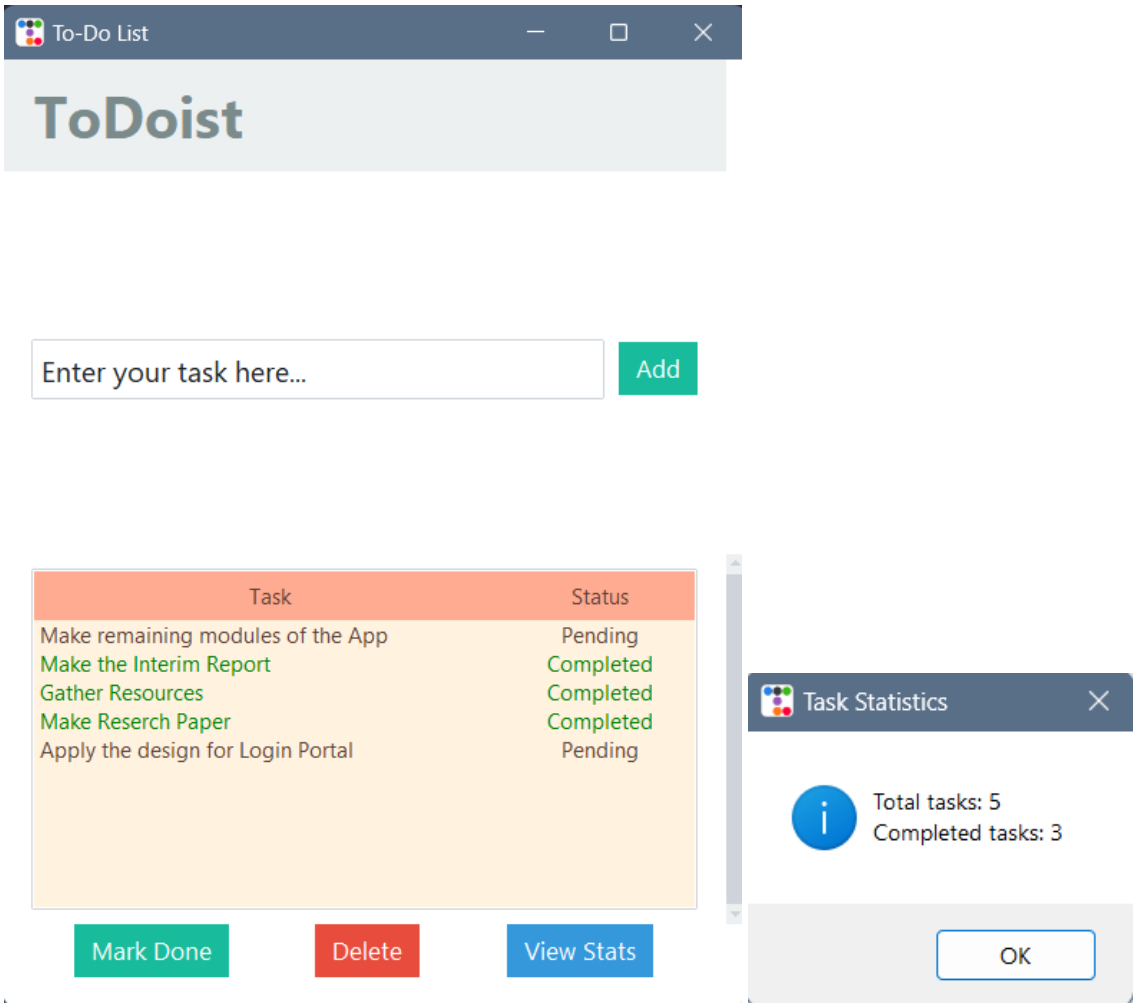

Figure 4: ToDoist: To-do List module to mange the tasks and keep track of the statistics of the Total, Completed and Pending tasks.

Stopwatch

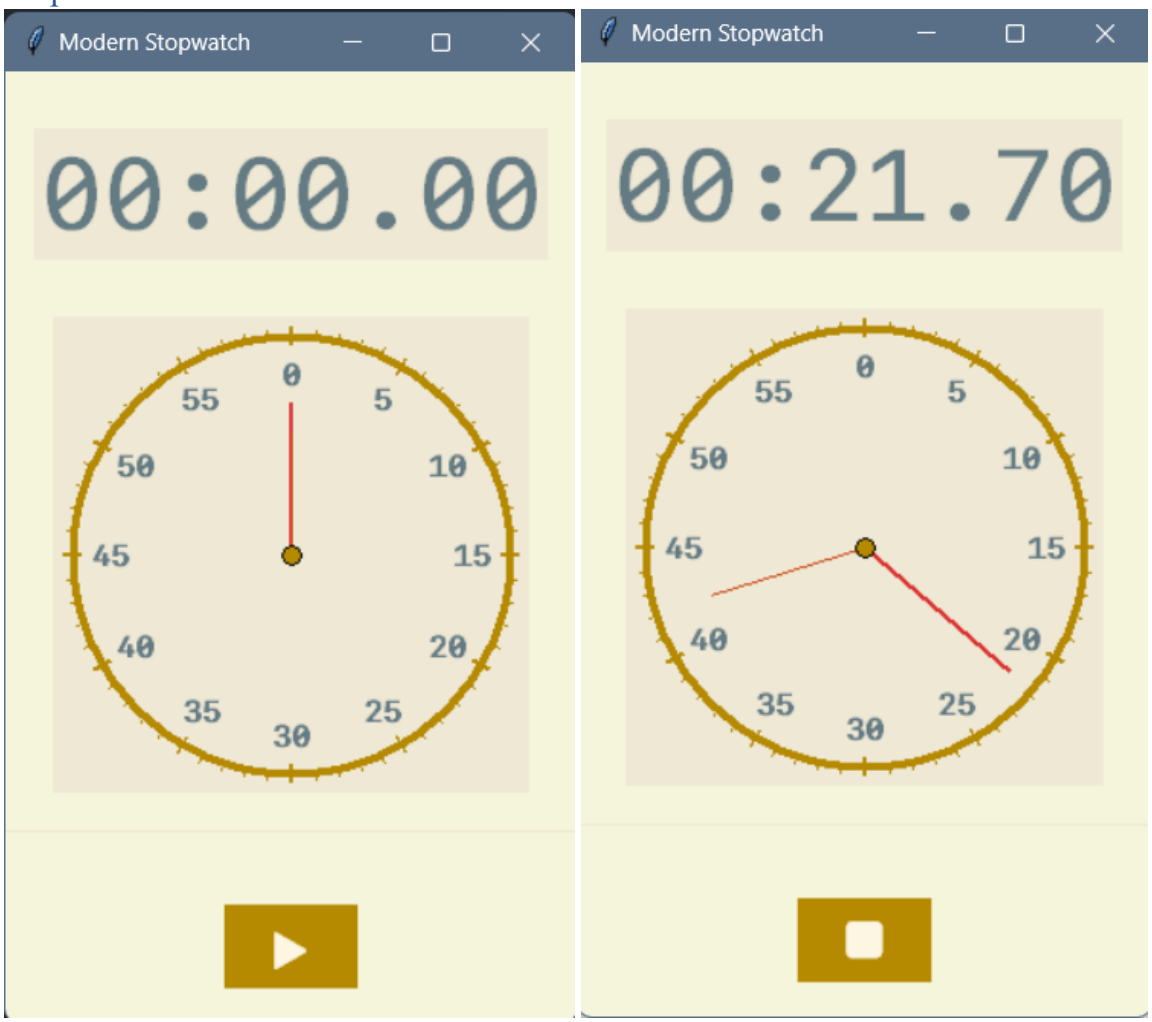

Figure 5: Stopwatch: Stopwatch module to keep track of the timed sessions and tasks.

Pomodoro Timer

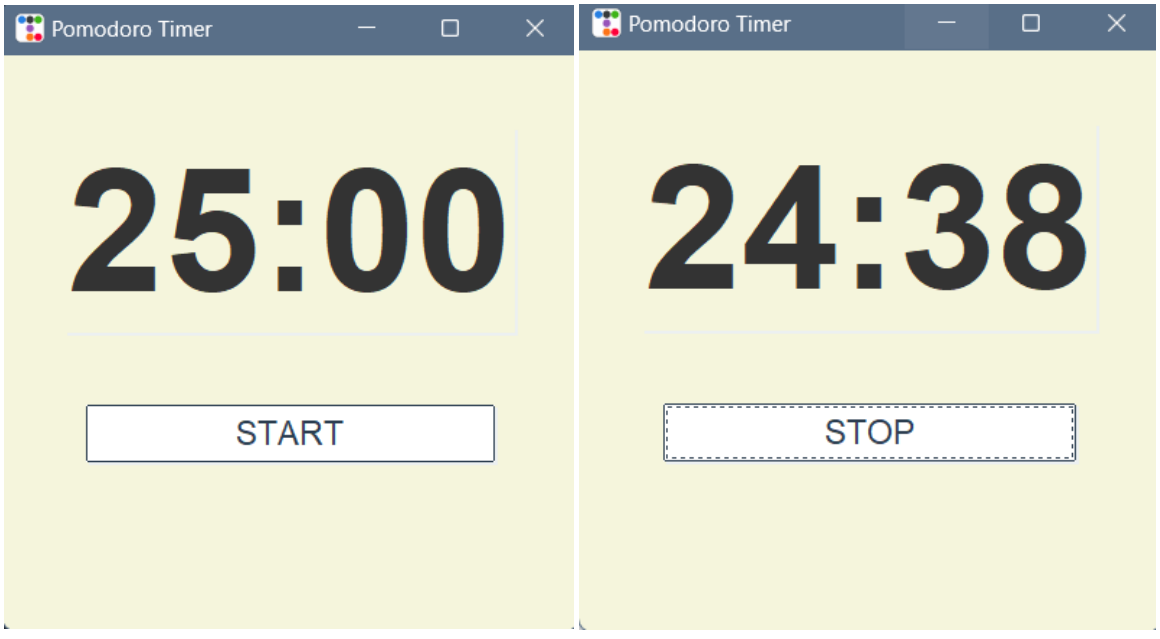

Figure 6: Pomodoro Timer: A focus as well as relief timer to keep the respective sessions consistent.