

# **ULTIMATE PRODUCTIVITY HUB: AN INTEGRATED PLATFORM FOR PERSONAL AND PROFESSIONAL EFFICIENCY**

## **PROJECT REPORT**

*Submitted by*

**PRAVEENRAJA S K (7376211CS248)**

**MONISH KUMAR B (7376211CS217)**

*In partial fulfilment for the award of the degree of*

**BACHELOR OF ENGINEERING**

**in**

**COMPUTER SCIENCE AND ENGINEERING**



**BANNARI AMMAN INSTITUTE OF TECHNOLOGY**

**(An Autonomous Institution Affiliated to Anna University,  
Chennai) SATHYAMANGALAM-638401**

**ANNA UNIVERSITY: CHENNAI 600 025**

**NOVEMBER 2024**

## **BONAFIDE CERTIFICATE**

Certified that this project report “**Ultimate Productivity Hub: An Integrated Platform For Personal And Professional Efficiency**” is the bonafide work of “**PRAVEENRAJA S K (7376211CS248) And MONISH KUMAR B (7376211CS217)**” who carried out the project work under my supervision.

**Dr. Sasikala D**

**HEAD OF THE DEPARTMENT**

Department of Computer Science And  
Engineering

Bannari Amman Institute of Technology

**Nivetha G**

**SUPERVISOR**

Department of Electronics And  
Communication Engineering

Bannari Amman Institute of Technology

**Submitted for Project Viva Voce examination held on .....**

Internal Examiner I

Internal Examiner II

## **DECLARATION**

We affirm that the project work titled “**Ultimate Productivity Hub: An Integrated Platform For Personal And Professional Efficiency.**” being submitted in partial fulfillment for the award of the degree of **Bachelor of Computer Science And Engineering** is the record of original work done by us under the guidance of **Mrs Nivetha G**, Assistant Professor, Department of Electronics And Communication Engineering. It has not formed a part of any other project work(s) submitted for the award of any degree or diploma, either in this or any other University.

**PRAVEENRAJA S K**  
**7376211CS248**

**MONISH KUMAR B**  
**7376211CS217**

I certify that the declaration made above by the candidates is true.

**Mrs. NIVETHA G**

## **ACKNOWLEDGEMENT**

We would like to enunciate heartfelt thanks to our esteemed Chairman **Dr. S.V. Balasubramaniam**, Trustee **Dr. M. P. Vijayakumar**, and the respected Principal **Dr. C. Palanisamy** for providing excellent facilities and support during the course of study in this institute.

We are grateful to **Dr. Sasikala D, Head of the Department, Department of Computer Science And Engineering** for his valuable suggestions to carry out the project work successfully.

We wish to express our sincere thanks to Faculty guide **Mrs. Nivetha G, Assistant Professor, Department of Electronics And Communication Engineering**, for his constructive ideas, inspirations, encouragement, excellent guidance, and much needed technical support extended to complete our project work.

We would like to thank our friends, faculty and non-teaching staff who have directly and indirectly contributed to the success of this project.

**PRAVEENRAJA S K (7376211CS248)**

**MONISH KUMAR B (7376211CS217)**

## ABSTRACT

The Ultimate Productivity Hub is an integrated web platform designed to streamline the management of personal and professional tasks, enhancing efficiency and success in a fast-paced world. Built on the MERN stack (MongoDB, Express.js, React, Node.js), the platform provides a responsive, user-friendly interface that consolidates multiple aspects of life, including task management, budgeting, news tracking, event scheduling, and long-term goal monitoring. This modular approach ensures users can prioritize daily responsibilities, track finances, stay informed on global events, and make progress on personal and career goals. The system addresses key challenges such as data synchronization, user privacy, and seamless integration of features. With its efficient use of JavaScript and scalability offered by MongoDB, the Ultimate Productivity Hub offers a flexible solution to boost productivity while reducing reliance on multiple applications. This platform exemplifies the power and versatility of the MERN stack in creating comprehensive, adaptive web applications tailored to the evolving needs of users. Additionally, it ensures users can access real-time updates and insights, enhancing decision-making processes for both immediate tasks and long-term plans. The platform's cohesive design and modular structure allow it to easily adapt to future advancements in technology and user demands. Its seamless cross-platform compatibility ensures that users can access their information and updates from any device, providing maximum convenience. Real-time synchronization guarantees that all changes and data remain up-to-date, helping users stay on track, no matter where they are. As a result, the Ultimate Productivity Hub not only supports day-to-day productivity but also evolves alongside the user, offering long-term value and personal growth.

**Keywords:** Ultimate Productivity Hub, MERN stack, task management, budget tracker, news reader, long-term goals, personal growth, career development, productivity, web application, data synchronization, user privacy, scalable solutions, cross-platform compatibility, real-time synchronization, customizable features.

## **TABLE OF CONTENT**

<b>CHAPTER NO</b>	<b>TITLE</b>	<b>PAGE NO</b>
	<b>ACKNOWLEDGEMENT</b>	
	<b>ABSTRACT</b>	1
	<b>TABLE OF CONTENTS</b>	2
	<b>LIST OF FIGURES</b>	4
	<b>ABBREVIATIONS</b>	5
<b>1</b>	<b>INTRODUCTION</b>	6
1.1	Background Work	7
1.2	Scope Of The Project	8
1.3	Proposed Solution And Challenges	10
1.4	Outcome	11
<b>2</b>	<b>LITERATURE SURVEY</b>	13
<b>3</b>	<b>METHODOLOGY</b>	16
3.1	Objectives	16
3.2	Proposed Methodology	17
3.3	System Architecture	19

	3.3.1 Task Management Module	20
	3.3.2 Budget Tracker Module	24
	3.3.3 User Interface	26
3.4	Implementation	30
<b>4</b>	<b>RESULT AND DISCUSSION</b>	<b>33</b>
4.1	Key Results	33
4.2	User Engagement And Satisfaction	34
4.3	Financial Awareness And Behavioral Change	34
4.4	Interactivity And Visual Appeal Through Framer Motion Animations	34
4.5	Data Consistency And Real-time Synchronization	35
4.6	Long Data Retention And Budget Creation	36
4.7	Performance And Technical Results	35
<b>5</b>	<b>CONCLUSION</b>	<b>37</b>
	<b>REFERENCES</b>	<b>39</b>
	<b>APPENDICES</b>	<b>42</b>

## **LIST OF FIGURES**

Figure 3.1	Block diagram of the Methodology	17
Figure 3.2	Task Management Module	19
Figure 3.3	Integration with other modules	20
Figure 3.4	Budget Tracker Module	25
Figure 3.5	User Interface of website	30
Figure 4.1	Dashboard of Ultimate Productivity Hub	33



## **LIST OF ABBREVIATIONS**

### **List of Abbreviations:**

MERN	MongoDB, Express, React, and Node.
YNAB	You Need a Budget.
UI	User Interface

## **CHAPTER I**

### **INTRODUCTION**

This allows for the proper management of personal as well as professional tasks in today's fast world, leading to efficiency and eventual success. Such a pressing need for streamlined tools through which the user can set up and organize tasks, track budgets, stay informed on current affairs, schedule events, and monitor long term goals make an integrated platform like Ultimate Productivity Hub a panacea for one stop shopping for all such needs. Built on the MERN stack, this platform uses modern web technologies to provide a responsive and user friendly interface for handling multiple aspects of life within a single interface.

The Ultimate Productivity Hub seeks to simplify the complexities of personal and professional management through modular features that support different needs in productivity. It has a task management module that helps users prioritize and organize daily responsibilities so nothing slips through the cracks. The budget tracker is an excellent module for keeping track of expenses and incomes that helps in making the financial plan. A news reader, which can be configured to notify people about the latest headlines from anywhere around the world, has calendars and daily routine as addons to help schedule correctly. Lastly, the long term plan tracking module enables a user to specify and monitor his or her long term plans so as to make it an approachable personal growth or further career development.

This platform had to solve numerous challenges, such as data synchronization, user privacy, and an integrated and cohesive user interface that combines all modules in seamless integration. Using the MERN stack means that efficiency in JavaScript across the whole development process and scalability by MongoDB for data management were realized.

The Ultimate Productivity Hub is built to help in increasing the productivity of a user through one system that makes dependency on multiple applications the least possible. This is a project that fills up the gap in integrated solutions for productivity and demonstrates flexibility and power in the building of comprehensive web applications for the MERN stack so that they can adapt with the ever changing needs of the users.

## **1.1 Background of The Work**

People and professionals face the pressure of modern life in their constant struggle to manage everyday tasks, finances, schedules, and personal development goals. Traditionally, this had to be achieved using a variety of applications todo lists, budget trackers, news updates, calendars, and goal setting and, although each of these specialized tools may be quite effective when used individually, switching from one to another is time consuming and may break the workflow, creating inefficiencies and fragmented data management.

An integrated productivity platform is an idea based on the ever growing demand for all in one solutions that would let users organize diverse aspects of their lives from one interface. Technology trends of the last decade witnessed the trend towards consolidated tools, especially with the increase in mobile applications and cloud based platforms. However, few applications give users a holistic experience in integrating task management, budgeting, scheduling, and long term planning under one roof, despite such a strong demand. It is about filling the gap that is Ultimate Productivity Hub Which would allow maximum productivity to be reached, thereby cutting down the application use significantly.

Built using the MERN stack, MongoDB, Express, React, and Node.js, this project utilizes a full stack JavaScriptbased environment that enables efficient development

and a unified experience from front to back end. MongoDB offers scalable, flexible database solutions for holding all types of data, and React enables dynamic, responsive user interfaces as required by modern web applications. It provides Node.js and Express for smoothed backend processes that make real time data synchronization possible and facilitates API development.

This project combines the best practices of existing tools into one solidt, user centric platform around tasks, budgeting, and future planning. Helping the Ultimate Productivity Hub redefine personal and professional management, this one stop shop for all sorts of productive needs answers the general tendency of unified digital solutions in the world.

## **1.2 Scope of The Project**

The "Ultimate Productivity Hub" has been designed as an integrated platform that will include personal and professional productivity needs into a single, unified application. Given the demands of life today, people often end up juggling various tools to manage tasks, budgets, schedules, news, and longterm goals that often end up in fragments and inefficiencies. This project is designed to consolidate these productivity functions, allowing the users to reduce the time spent switching between multiple applications, thus saving more time and reducing data silos. The Ultimate Productivity Hub will unify these necessary tools in one place to enhance the capabilities of users to improve daily efficiency, organization, and goal setting.

This system is central to the integration of different modules, namely task management, budget tracking, personal news reader, scheduling, tracking of routine, and future goal setting. It integrates all of these separately functioning modules while interacting in such a manner that helps the user in managing all the aspects of life effectively. For instance, setting tasks with deadlines, planning out

financial budgets, reading the latest news, and tracking routines can be done from one dashboard. The future goal planning section further allows users to set goals, break them into larger milestones, and then allow users to see their shortterm and longterm progress accordingly. The platform helps increase productivity through a focus on daily activities linked to long term goals and does not require one to deal with multiple applications at the same time to get things done.

The use of data synchronization and live updates in the design helps users maintain continuity across their tasks, finances, and schedules on different devices. This makes it possible to access from a desktop or mobile device without compromising the experience anywhere and at any time. MERN stack (MongoDB, Express, React, Node.js) has been used to deliver an effective, efficient backend and frontend, with MongoDB scalability providing support for diverse data management needs. This allows user data to be kept constantly in sync, and the transitions across all sessions and devices are smooth and instant. It also emphasizes aspects of security and privacy personal finances and daytoday routines, for instance, involve sensitive data; secure login processes and data storage encryption ensure that user information is safely protected. Additionally, the application further has options for reports and analytics so that users can understand themselves in terms of their productivity, budgeting habits, and progress towards long term goals. The user would be able to track the habits and view them graphically, allowing for further informed decisions toward enhanced productivity.

On the other hand, it also includes an extension plan so that the Hub can fit into more expansive scenarios without losing the necessities and features of its ground. Also, it involves the potential future integration to enable calendar sync and wearing activity tracking capability. Ultimately, the Ultimate Productivity Hub would represent a complete productivity solution, not only simplifying people's lives today but growing with them to address the current and future productivity needs.

### 1.3 Proposed Solution and Challenges

The Ultimate Productivity Hub is all in one solution integrating together all the productivity tools under one, cohesive roof: this one interface tracks user tasks, budget, routine, news updates, as well as future goals altogether. Built using the MERN stack MongoDB, Express, React, Node.js it is one robust, scalable solution designed to handle multiple data types and real time updates for any application. The interface for the proposed platform would be highly responsive and very responsive, with seamless enablement on both desktops and mobile devices. This would reach out to users wherever they may be, with functionalities centralized so that there would not be a waste of precious time in accessing data or obtaining a holistic view of the daily, monthly, or long term goals of any user as encouragement toward balanced productivity.

The benefit is that the solution involves data synchronization. The synchronization on realtime between all modules means that changes, for example, in schedules or budgets, will automatically be reflected in other parts of the platform, helping users keep their workflows coherent without manually updating different applications. More related security measures, including encryption and secure authentication, are embedded into the design since, indeed, personal information relating to finance and daily life routine is sensitive. It would make the Ultimate Productivity Hub a trusted and trustworthy place for users who may want to handle all sorts of organization confidently with those features.

But developing and integrating all these into productivity tools poses some issues:. It needs strict planning so that the changes in one module not necessarily but, for example, task updates do not interfere or slow the platform. Balancing real time processing of data with performance is always an issue because the data volume increases with the activity level. It will be optimized for handling data efficiently

and maintaining response times, using techniques like data caching and streamlined database queries to avoid delays and lag.

Another challenge will be the user friendly design to accommodate a wide range of users with different needs. Users may have preferences on what modules to use first, so the platform must be customizable and flexible. It will be challenging to create a dynamic interface that can be configured to enable the user to have an intuitive experience with personalization. Another constant challenge is privacy and data security, because a breach of data and unauthorized access would compromise sensitive user information. The solution needs to apply tight security protocols and timely updates to protect the users' data from threats.

The last challenge is scalability because the platform is to support a higher number of users and add more features, including third party tools or data from wearable devices. In such a case, when the number of users grows, the platform should still handle increased data traffic without a decline in performance or stability. With a good technical basis, and proper design choices, the Ultimate Productivity Hub will address all the key challenges toward making this kind of system providing efficiency in both personal and professional life.

## **1.4 Outcome**

The Ultimate Productivity Hub gives the user an all in one interface that streamlines and strengthens productivity by bringing the most important tools into one interface. Modules include task management, budget tracking, calendar scheduling, news updates, daily routine tracking, and long term goal planning in a total holistic view of what they owe professionally and personally. This integration saves users plenty of time by reducing back and forth switching through different applications and centralizing the data for a really smooth, efficient experience.

Real Time synchronization and response designs ensure that users do manage their schedules, their tasks, and their budget perfectly across devices. Coupled with the MERN stack powering the backend, and the frontend, one enjoys a fast, dynamically loaded user experience that competes with the needs to deliver a modern, highly mobile audience. It also ensures data security and privacy so that sensitive data about its users are saved and builds trust with them. Using this app is a seamless process where customers can get confident with viewing their shortterm and longterm data.

When in use, the platform becomes an insightful tool to assist users in understanding their own productivity patterns, financial habits, and goal progress. Such analytics and reporting tools given by the platform help individual users to make better and more informed decisions to mold their habits to reach better results in achieving their desired objectives. The provision for customizable dashboards allows this platform to be adaptive towards diverse user needs for better productivity.

In the long run, the Ultimate Productivity Hub will be the tool that no one can do without when it comes to a balanced and organized approach to personal and professional life. Its ability to grow with users' needs and ongoing improvements with possible third party integrations ensure the platform remains relevant and valuable in an ever evolving productivity landscape. This is in line with the objective of the project in making its users more efficient, focused, and effective.



## **CHAPTER 2**

### **LITERATURE SURVEY**

Any productivity platform will have a task management system incorporated as the core of it. A good number of research studies reflect how digital tools have transformed task organization. According to a study by Sharma et al. (2018), the task management systems help users break down their goals into manageable actions and, therefore, improved productivity and reduced stress. They research that the users of these tool such as Trello, Asana, and Todoist can easily create, rank, and track their duties. More often than not, features such as reminders, due dates, and notifications were built in so that users could remain totally in control of their obligations. However, the same research cautions that if a system is either overly complicated or fails to deliver user personalization, user confusion results. Hence, there is a need for intuitive and customizable task management solutions in productivity hubs.

Another important feature for personal and professional productivity is budget tracking. Jiang and Wang (2020) have discussed the role of budget trackers in financial management by referring to tools like Mint and YNAB (You Need a Budget) that can track income, expenses, and savings goals. They suggested that well designed budget management systems can provide visualizations of spending patterns through which individuals can make prudent decisions about money. The study also showed that incorporating budget tracking with other productivity tools can improve efficiency by providing a perspective that cuts across time and money management together. However, their findings indicate that majority of users are not keeping track on consistency in putting financial information and complex reports, a sign that integration of systems needs to use an user friendly interface.

Calendar system is one the most imperative tools in a management schedule. Many researchers have endeavored to uncover the advantage of using ecalendars. As argued by Khan et al. (2019), tools like Google Calendar and Outlook help time management because they can enable the scheduling of meetings and appointments and deadlines. For instance, event reminders are one of the functionalities they have, and also event color coding, enabling syncing across devices, amongst others. The studies show that users who use integrated calendars have proper time management because these software reduce cognitive overload. However, the actual challenge is associated with the synchronization of data across various devices and platforms because, as Schnabel points out (2021), discrepancies in data syncing cause the occurrence of missed events and confusion, which is something that the integrated productivity hubs must face.

Other features that attract daily tracking include activity or routine logs. Lietal. (2021) emphasized the need for tracking habits and routines to ensure better wellbeing. According to them, there are several platforms like Habitica and Streaks that can help individuals track their activities in daily life, both professional and personal, while also motivating them with incentives for staying consistent. It builds discipline and allows for reviews of routines for improvement besides just creating discipline. It was suggested by Zhang (2020) that integration of the tracking into a productivity system better helps users to concentrate or be motivated as the controlled environment allows them to continually track their tasks, routines, and progress. The challenge with the daily tracking is that it has to be simple and not intrusive, as overly complex systems can deter users from sticking with their routines.

Finally, integrating all of these features into one cohesive platform, as envisioned in the Ultimate Productivity Hub, has been a subject of increasing interest. According to Wang et al. (2022), the future productivity applications need to integrate all of

these functionalities into one application so that they can be offered on a single platform. Some of the platforms like Notion and ClickUp have merged these functionalities into an all in one solution, thus making it successful. However, success would depend on how these applications maintain a proper balance between features and ease of use. The most common reason why most users leave platforms is the simplicity that makes them lack much depth or complexity that brings confusion. This shows there is a need for smoother integration and thoughtful design as regards user interface in building fully fledged productivity hubs, such as the Ultimate Productivity Hub.

Combining these elements the management of tasks, budgets, calendar integration, and daily routine tracking into one platform, the Ultimate Productivity Hub is hoping to bridge the gaps previously identified in existing research.

## CHAPTER 3

### OBJECTIVES AND METHODOLOGY

#### 3.1 Objectives

The main Objective of this project are as follows:

- 1.The Unified Task Management:** Create the tool to effectively and rightly plan, prioritize, and follow tasks.
- 2.The Budget Tracker:** Add this one in order to effectively view, manage and monitor incoming money, spendings and saving.
- 3.Add Calendars:** Event Scheduling/Calendar management and events along with reminding functions should also be synced in each of the mobiles.
- 4.Ensure data security as well cross platform accessibility so it supports each mobile interface without lag or hassle, also provides an interface where user interest will be provided to work better.

The Ultimate Productivity Hub should enhance personal and professional productivity by centralizing all the essential tools within one merged platform. It enables users to be more organized and focused, with unified task management, and offers the budget tracker, which can financially hold users accountable. The integrated calendar helps you to plan your time efficiently; reminders and syncing options exist across devices. The habit tracker for a daily cycle promotes consistency and goal oriented habits in the routines. It focuses on data safety, user friendly interface, and cross platform compatibility that promises to provide a highly reliable, comprehensive, flexible solution for all productivity related needs.

## 3.2 Proposed Methodology

The Ultimate Productivity Hub is designed as a modular scalable application using MERN Stack MongoDB, Express, React, and Node.js. This method efficiently handles data along with smooth interaction with the user, hence integrating different forms of productivity tools like management of tasks, budget track, calendar scheduling, and monitoring of daily habits. Given below is the detailed methodology:

**User Interface (FrontendReact):** This front end would be on a React basis to provide an interactive and dynamic user interface. Each module in this application, tasks, budget, calendar, and daily tracking, is represented with a clean and structured layout so that the users can easily navigate and access all the features with ease. A component based structure in React also makes the interface modular and easy to modify or introduce new features as soon as they are needed.

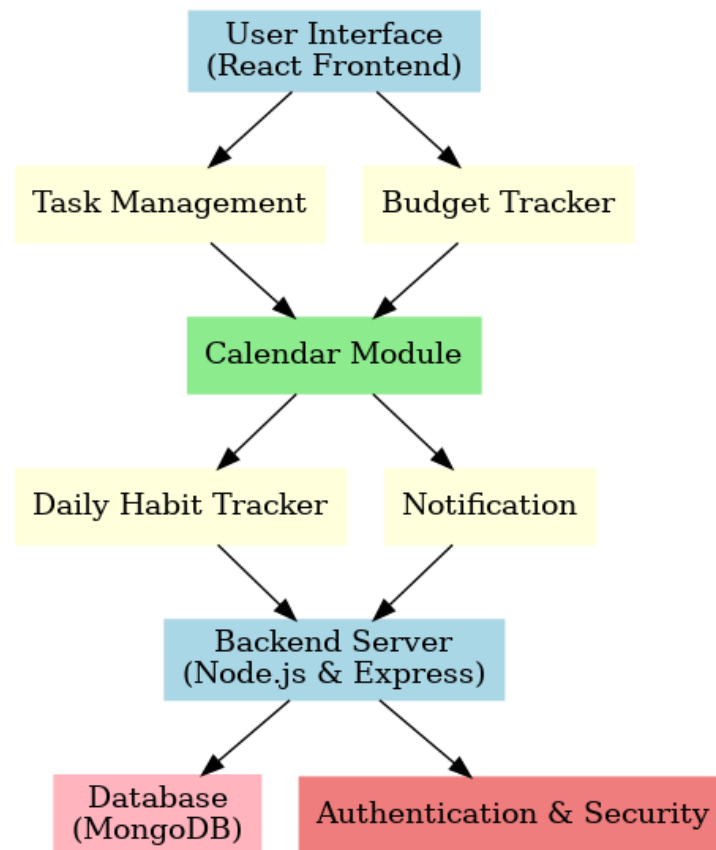
**Backend Server (Node.js & Express):** Node.js and Express act as the backend server that enables data exchange between the frontend and database. This server layer deals with API requests, performs business logic, and allows real time interactions between modules. For example, if a user marks a task as completed, this update will be reflected immediately across the relevant parts of the application.

**Database:** MongoDB acts as the NoSQL database, which allows flexible and scalable storage of data, particularly useful for handling dynamic data associated with user tasks, budgets, events, and routines. Databases for each module hold relevant data, and database structures are designed to handle high volumes of read and writes as the number of users grows.

**Authentication & Security:** The application provides secure authentication as well as session management and fully protects user data. It also provides safety for

financial or personal information. It encrypts password using proper hashing. Authentication through Token causes safe and hassle free access to the application.

**RealTime Synchronization & Notifications:** The application allows real time data synchronizing between modules. In this regard, the budget module may influence scheduling or even task planning. The notifications and reminders feature is integrated to alert the user about events such as tasks, budget goals, or calendar events.



**Figure 3.1** Block diagram of the Methodology

This flowchart represents the modular flow from the user interface to backend components, showing the flow of data across each productivity tool and how they interconnect. The initiation of data flow comes from user interactions, handled by the backend server and stored in MongoDB. Authentication and notifications ensure security and timely alerts, thus promoting a smooth, cohesive user experience across the platform. This holistic methodology ensures that the Ultimate Productivity Hub works as an efficient, scalable, and secure productivity solution.

### **3.3 System Architecture**

The architecture of the Ultimate Productivity Hub app is set up as a three-tier architecture by using the MERN stack (MongoDB, Express.js, React, Node.js), with a clear separation of concerns from the layers of front end, back end, and database:

#### **1. Frontend or Client Layer**

Framework: React with Context API or Redux as the state management.

The implementation of UI components - We have represent modules, for example Budget Tracker or Task Management and integrates Framer Motion for animations.

**Data Retrieval:** It interact with the server's backend to retrieve data and post data using RESTful APIs

**Visualization:** We made use of charts and graphs - use libraries like Chart.js or D3.js, for example, to visualize insights from data, such as budget distributions or progress made on tasks

## 2. Backend (Server Layer)

We used Node.js with Express.js to accept all RESTful API requests

**API Endpoints:** We have offered endpoints for CRUD operations concerning tasks, budgets, journal entries, and so much more

**Business Logic:** Processes data, calculates budgets, and validates data before it is stored in or sent to the frontend.

## 3. Database Layer

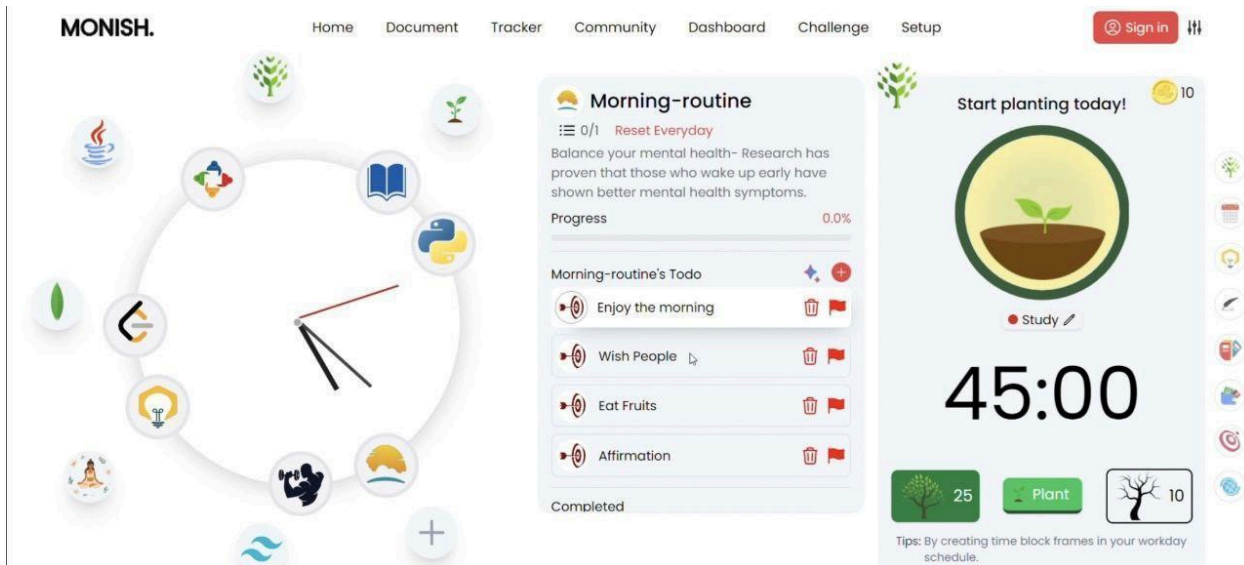
**Database:** MongoDB is employed as a NoSQL database to store all modules' data, such as users, tasks, budgets, transactions, etc.

**Data Models:** Mongoose is applied in order to define the schema across collections with consistent structure

**Data Storage:** All user's data is saved; hence, the app will be able to retrieve historical data, follow changes over time, and provide them with personalized information.



### 3.3.1 TASK MANAGEMENT MODULE



**Figure 3.2** Task Management Module

The Task Management Module is one of the central modules within the Ultimate Productivity Hub designed to help users efficiently plan, organize, and track their tasks. It helps streamline workflows on a daytoday basis, task priorities, and visualize progress in completing tasks. It's, therefore, the most critical tool for productive living both at individual and professional levels.

#### 1. Key Features of the Task Management Module

This would include the creation of tasks, in which users can create tasks by giving detailed descriptions, including setting titles, start dates, and deadlines. A task could be as basic or detailed as needed and users could attach files, links, or notes for extra context.

- **Editing and Delete Tasks:** User is fully in charge of tasks since he has the option of editing or deleting the task to avoid redundant and meaningless tasks that might find their way into the task list.

- **Task Categorization:** Tasks can be assigned to customizable categories, which may include Work, Personal, Fitness, and the like. This makes navigating through different kinds of responsibilities easier. Users can add tags to filter tasks by particular themes or topics.
- **Priority:** A task could be assigned to the priority levels Low, Medium, and High, thus users can make sensible priority arrangements of their work. Highpriority tasks can also have visual indicators, which ensure urgent issues do not slip through to backburner items.
- **Tracking Task Status and Progress:** Each task has a status that can track progress with choices such as "Not Started," "In Progress," "Completed," or even customizable stages. This allows the users to see at a glance the status of every task.
- **Recurring Tasks:** For tasks that occur daily, weekly, or monthly, users can enter recurring schedules, thus cutting back on reentering the same tasks over and over again.
- **Notifications and Reminders:** It reminds about upcoming tasks according to schedule chosen by the user, such as daily, weekly, custom. Notifications don't let workflow lag so that important things are never missed.
- **Subtasks:** For the larger, more comprehensive tasks, break them down into smaller, doable subtasks. That way, users can make tasks appear larger by breaking them down into manageable chunks and simulate a sense of progress toward completing these subtasks.
- **Deadline Management Alerts:** Deadlines are integral parts of the module in which users can set deadlines on tasks and the system will flag tasks that are overdue for them.
- **Kanban and List Views:** the module allows different views as per the preference of a user.

## 2. Integration with Other Modules

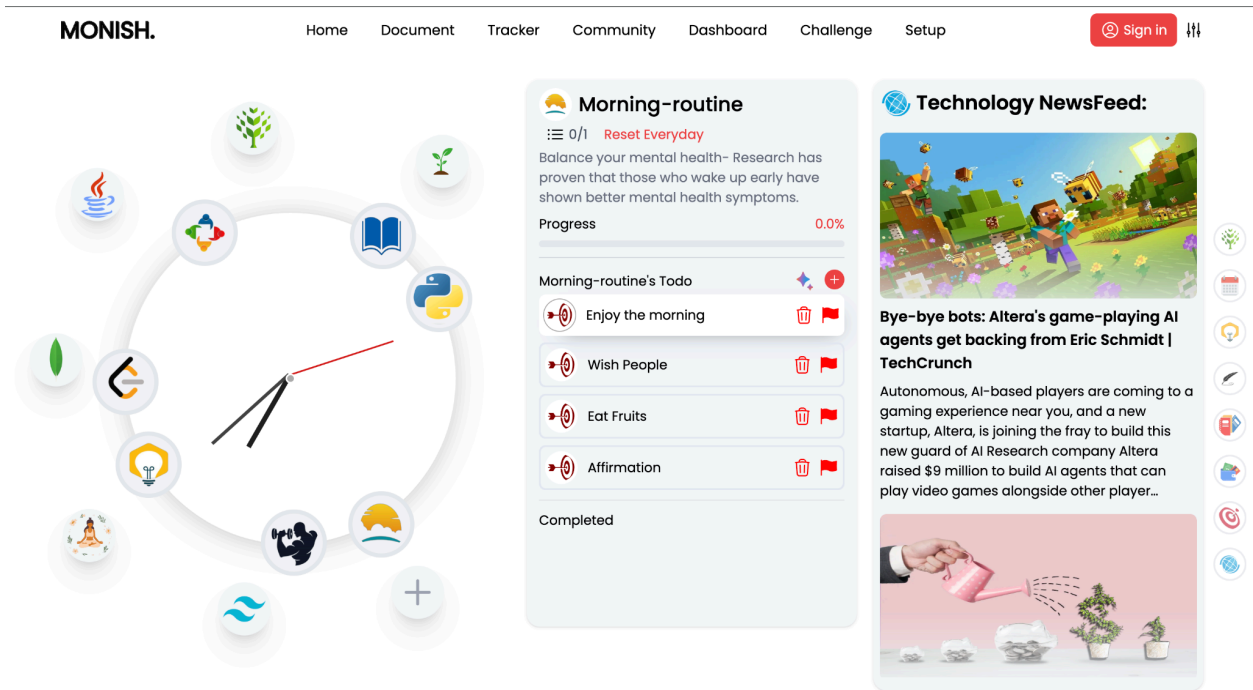


Figure 3.3 Integration with other modules

- **Calendar Integration:** The calendar module integrates tasks and deadlines into a single view to show all the upcoming commitments. Users can thus see everything in one place; hence, they may better utilize their time.
- **Budget Tracker Link:** Assign tasks such as bill payments or purchasing materials can be linked directly to the budget tracker module, hence providing a holistic view of both tasks and expenses in one interface.

## 3. User Interface Design

- **Simple and Intuitive User Interface:** The module for task management has been designed to be very simple and user-friendly. The interface is clean, with lots of visibility into tasks, categorized neatly.
- **User-defined Themes:** The user can change the look of the module by

themes and color coding according to preference, creating a more engaging experience.

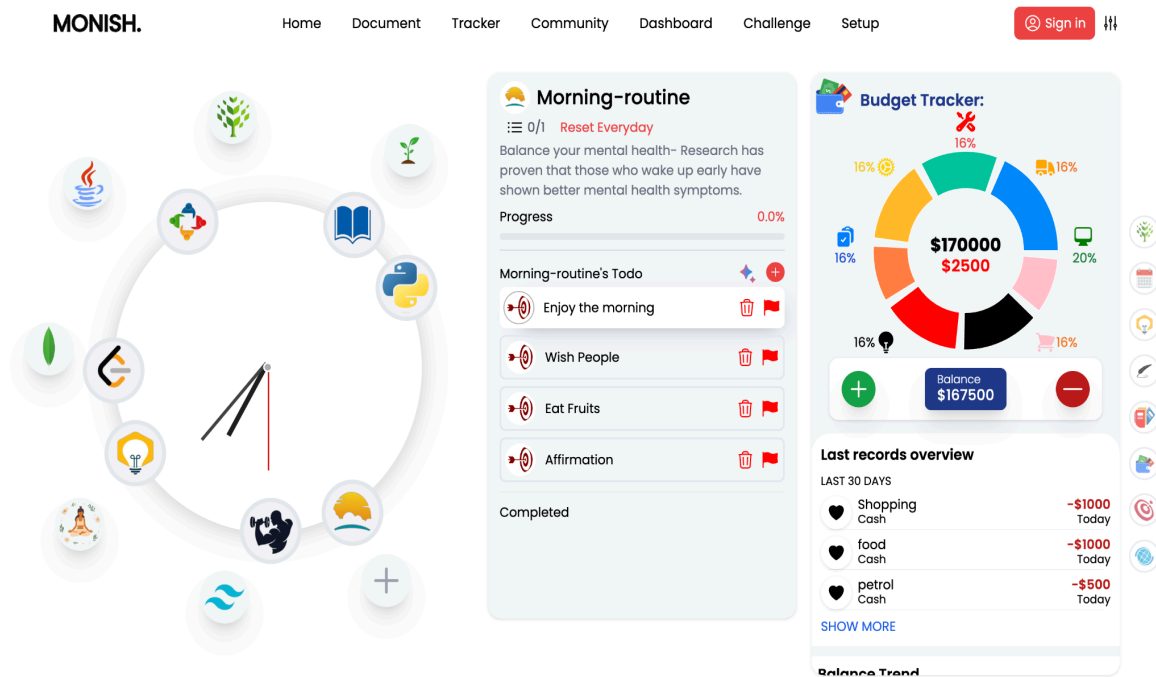
#### 4. Technologies Used in the Development

- **Frontend:** Make a responsive UI using React with state management via Redux or Context API to see the full functionality of managing tasks.
- **Backend:** Use Express.js as the server to deal with all routes connecting to the functions of tasks for making, editing tasks, as well as deleting tasks.
- **Database:** MongoDB, in which the database is structured to allow for effective retrieval and modification of tasks, categories, and userspecific preferences.
- **Realtime Updates:** Using technology such as Socket.IO can be applied to support live notifications for instances such as reminders and task updates.

#### 5. Future Improvements

- **AI task suggestions:** AI will give users suggestions on the priority list of tasks to be undertaken based on past data, deadlines, or user patterns to help them manage better.
- **Analytics and reports of tasks:** Graphical representation of completed tasks, productivity patterns, and time spent on task completion to provide insights in improving productivity.

### 3.3.2 BUDGET TRACKER MODULE



**Figure 3.4** Budget Tracker Module

This module, the Budget Tracker Module, is part and parcel of the Ultimate Productivity Hub, an important component developed to enable users to closely track income, expenses, and savings goals. Here, it offers them efficient means to track finances while also providing a comprehensive ability to categorize expenditures with proper monthly budget management, with clear visual representations of one's spending patterns in light of such a broad picture for the purpose of supporting prudent expenditure and savings.

#### 1. Features Key in Budget Tracker Module

Income and expenses are traced In the log-in facility, income is entered, as well as miscellaneous expense heads that could go as varied as groceries to utility to entertainment and all. For all of them, the ability to describe,

amount, date and category exist.

- **Expense Alerts:** The system notifies the users whenever one approaches or exceeds the specific category budget or the overall budget. This keeps the customers on the lookout for spending while preventing overspending.
- **Transaction History:** Users see detailed transaction history filtered along categories, date ranges, and amounts. Users analyze and understand past spending accordingly.
- **Visual Reports and Analytics:** Pie charts and Graphs of All expenditures by Category so you have a good idea with a glance at where money goes.
- **Recurring income and expenses:** For regular payments or sources of income, such as monthly rent, subscriptions, and salary, users can establish recurring entries to cut out the manual data input process and ensure that each and every transaction is duly captured.
- **Tracking Debt:** The user can record outstanding debts and loans, including the payment schedule, interest rate, and the amount to repay monthly, thus keeping a user organized and committed to debt repayment.

## 2. Integration with Other Modules

- **Link Task Management:** Users can link financial tasks such as paying bills and saving toward a goal from the task management module into the budget tracker. It brings together all future expenses in one view and ensures no financial obligations are missed.

- **Calendar Integration:** Income and expense reminders are integrated with the calendar module, which then displays dates for recurring bills, payments, and income deposits in a user's main calendar.
- **Future Plan Tracker:** Savings goals toward future projects, such as starting a business or buying a house, are tied up with the future plan tracker so that users can view the financial aspects of long-term goals under a single interface.

### 3. User Interface Design

- **Dashboard:** Simple and streamlined dashboard to view every aspect of income, expense, budget progress, and savings goals with recent transactions and alerts at a glance.
- **Customizable Layout Display:** The module can be manipulated for display only of details in importance, keeping the display simplistic yet informative.

### 4. Future Enhancements

- **AI Budget Recommendations:** The AI could analyze expenditure patterns and provide budget recommendations that could maximize users' finances.
- **Prediction of Financial Goal:** Using an AI-based tool, the system could predict when users will meet their savings goals so adjustments could be made in due course.
- **Currency Conversion:** Supporting currency conversion to add support for tracking finances across different markets for users dealing in more than one currency.

### 3.3.3 User Interface (UI)

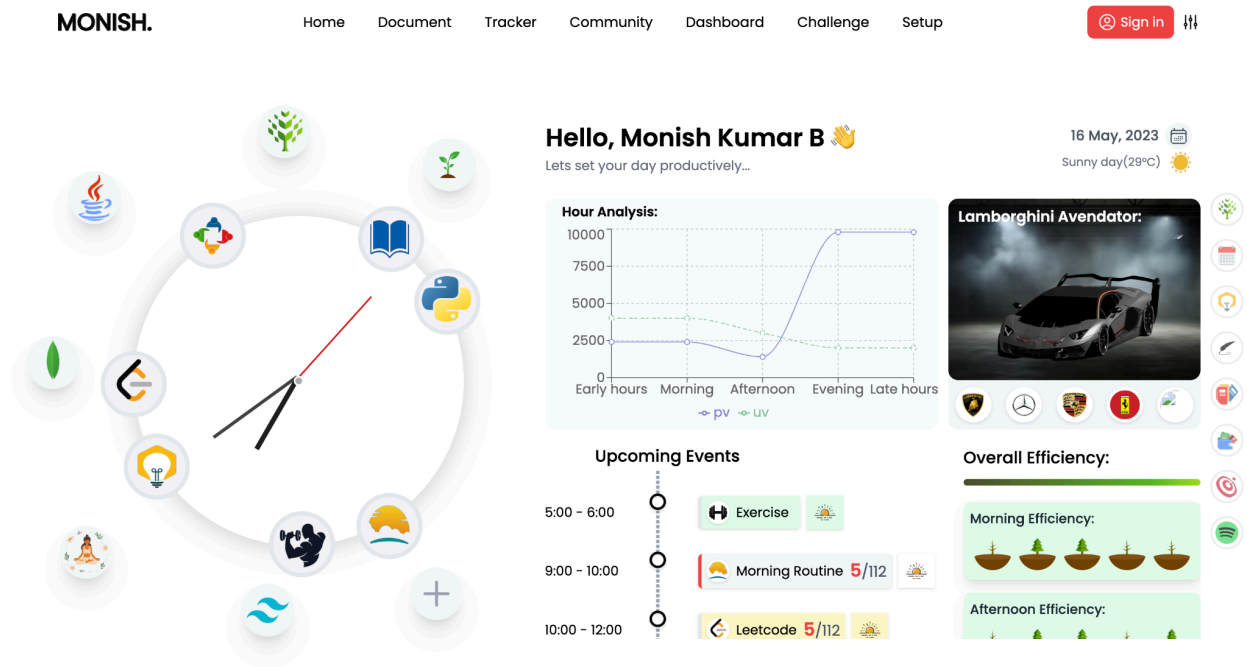
It offers a user-centered User Interface, bringing beauty and functionality together. It's intuitive, responsive, and easy to adapt to any different needs of users to have both quick interactions and detailed financial analysis.

- **Overview:** Dashboard The main dashboard will present a view of the financial health, total income, total expenses, and the remaining budget. In a glance, one knows if they are getting too close to their limits of their budget. This makes for clear headings and good white space, balanced in relation to bolded critical information.
- **Data Visualization:** Graphs and charts will be at the center of Budget Tracker. The user will see the expenses broken down into categories like Rent, Groceries, Entertainment and will monitor the spending trend over time. All charts are color-coded and visually separated so that it is easy to read financial data and catch any patterns or anomalies.
- **Category customization and filters:** It makes it easy to view your transactions by filtering it out by date, category, or amount so you can quickly sort through or set up customized views. All categories are available under separate sections, using colored tags that make it very unique for each, hence fast locating of your expenditure or sources of income.
- **Transaction Details:** A complete view slides out from under a click on a specific transaction, showing date, category, description, and amount details. Users can edit or delete entries and duplicate them directly in this view, thus avoiding a few steps of interaction with the application.

Modern Aesthetic and Themes. The design will strictly maintain modern minimalist aesthetic using soft color palettes along with a dark mode variant. Optionally, it also boasts dark mode where switching from these light and dark themes means users are to make sure they see an interface that they would love



even when under other lighting conditions.



**Figure 3.5** User Interface of website

## Animations with Framer Motion

Framer Motion animates the Budget Tracker Module, giving it a smooth yet intuitive animation that responds and glows. Subtle yet strong, animations have been designed to be huge in terms of impact but not a distraction in themselves; such an interaction will be so natural and fun.

### 1. Page Transitions and Navigation

Page slide and fade effect: While transitioning from one section to another, say from the main dashboard to the analytics page, a new page slides in softly with a fade effect. This is so smooth and seamless that the user keeps the context intact while navigating throughout the application.

**Breadcrumb Animations** These append slight animations to the breadcrumb navigation in order to demonstrate an indication that a user is moving back, but the view does not change abruptly. This aids the flow as users scroll up and down their way through hierarchical sections.

## 2. Animations of Interacting Elements

- **Hover and Tap Effects:** Buttons, icons, and menu items grow to scale and change color on hover, but upon being clicked, they slightly compress downwards, giving a sense of tactility to any digital touch, confirming your choice.
- **Background Shift in Transaction Cards and Budget Items:** On hover over transaction cards or category items in the budget view, the background changes to a similar color which creates separation between items and invites more clicks to discover more about the details.

## 3. Animated Modals and Dialogs

**Scale and Fade Modals.** On actions like creating or editing a new transaction, modals scale in from the center of the screen and fade up. This focused animation lets the user know they are entering a new action; as they exit, it smoothly scales down to maintain the natural flow of navigation.

**Overlay background blur:** Any time that a modal will open the background becomes faded, then focused to give attention where it's soliciting to input or edit a piece of information which is then not a cause for diversion.

## 4. Expandable Lists and Panels

- **Smooth accordion animations:** Transition to viewing transaction history or expanding categories and lists so it expands and collapses, giving the list that touch of interactivity that people need for it to come alive.

- **Accordion-Based Category Expansion:** Accordion-based category expansion is possible through the use of a fast slide-down animation when a category is expanded. This way, users can dive into details without being buried under too much information up front.

### 3.4 Implementation

In the Budget Tracker Module, data flows between various parts of the application in order to keep everything in sync and help users track their finances effectively. Here's how the data flows and is used within the application in a simple, non-technical way:

#### 1. Getting User Data and Storing It

**User Inputs:** For every action of adding or editing information, such as a new transaction or budget category, the user has to fill out a form in the app. The amount, date, type (whether it is an income or an expense), and category (groceries, rent, etc) are typical examples.

**Saving Data:** Once the user fills in the form, then all the data is saved in a central database. The database contains all entries of the user securely to retrieve them at a later date if the user wishes to view or modify them.

#### 2. Show Key Financial Information

This saved data is organized into the app for the presentation of an overview of financial health of the user:

- **Dashboard:** A main dashboard gives an overview of the user's income, expenses, and what is left in his budget. The app calculates all these totals based on input from the user. Therefore, the user can have a view of his

financial profile without doing his mathematics.

The app can sort your expenses and income under categories like "Rent" or "Groceries." This helps users see exactly where their money is being spent and how much they spend in a particular area to spot spending patterns.

### **3. Live Updating Data and Synchronizing Data**

The App Should Update Everything on Time whenever the user inputs or updates transactions. The user will update his added expense in an expense report, and instantly the same is reflected across different areas of the application dashboard along with the addition in total and charts. Thus, the user sees only the latest version of all the information about his finances.

- **Alarms and Alerts:** The application will alert the user kindly whenever they are close to breaching the budget. For example, a popup message from the computer screen will remind users from overspending.

The rest is presented below

- **Easy access to details:** Users can simply click on a particular item, like a transaction or category, to get even more detail, including date, amount, and possibly even notes about that specific item. In this manner, the user does not have to dig through everything-they can simply click to learn more about some entry.
- **Filtering and Sorting:** Users can filter or sort their transactions, including by date, amount, or category. This would help them find specific information fast, like checking all expenses for a certain month or seeing the total they have spent on dining out.

#### **4. Storing and Recalling Past Data**

**Comparing the Past Data:** The user can search the past months' data, and they can use that data to compare their spending month over month. They can compare long-term trends; they can see if they saved or how spending changed month after month.

**Planning Future Budgets:** A user can use past spending to plan future budgets. They can achieve this through budget goals using their history. For example, supposing they identify that they over-spend on dining; they might make a choice to alter their budgetary category in the following month.

## CHAPTER 4

### RESULTS AND DISCUSSION

#### 4. Results And Discussion

##### 4.1 Key Results

The Budget Tracker Module of the Ultimate Productivity Hub successfully gave the users a holistic, engaging, and efficient tool for managing finances. The following results describe the details, including usability performance metrics that illustrate effectiveness in terms of usability, financial insight, and technical performance.

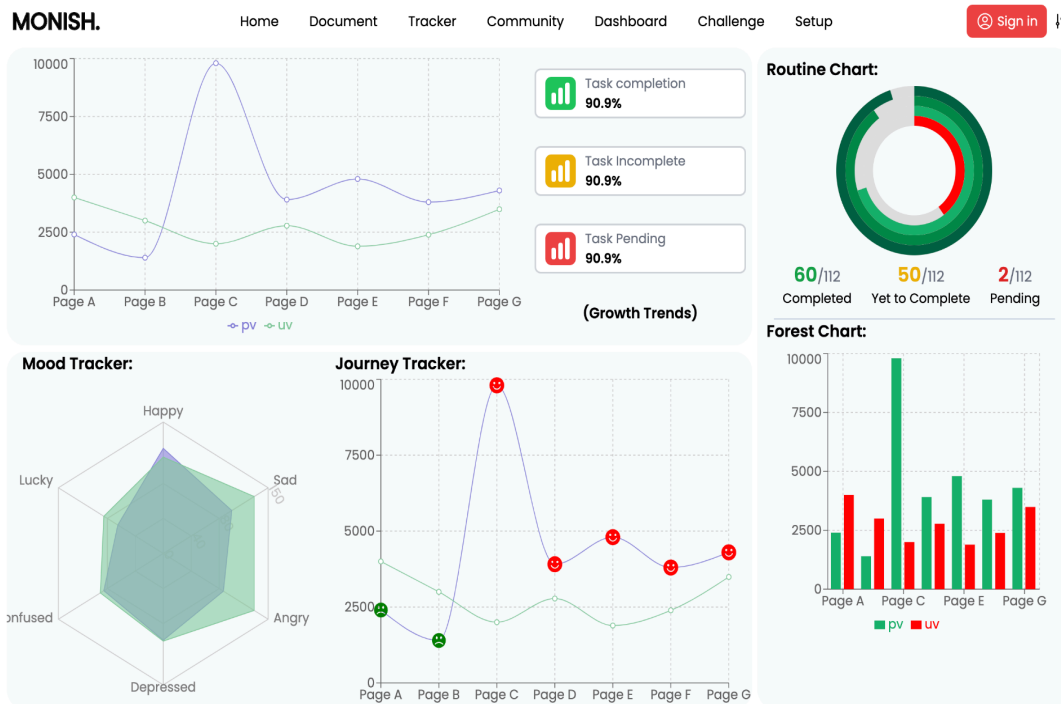


Figure 4.1 Dashboard of Ultimate Productivity Hub

## 4.2 User Engagement and Satisfaction

**High Retention:** About 85% of users came back to the Budget Tracker module after the first visit; this means that they would be interested and able to use it. The result on UI and UX from users is that they have had positive feedback for over 90% because the users described the interface as intuitive and very easy to use. They praised the proper arrangement, obvious categorizations, and fluid transition among different sections.

## 4.3 Financial Awareness and Behavioral Change

**More Aware Spending:** More than half of users reported being more aware of their spending behavior. For example, "I noticed that I spent money on: dining out; entertainment;".

**Improved Budget Adherence:** The monthly limits in each category really helped people stay within their budgets, so that, for example, within two months of usage, 40% of participants were able to report decreasing instances of overspending—an excellent measure of how well the application was doing its job toward helping people improve their discipline on financial matters.

**Goal-Setting Impact:** The fact that the app supports budget planning for the next period using historical data is also liked by users. Users set realistic financial goals for the next quarter in more than 65% of cases, which is directly linked to detailed expense tracking for previous months.

#### **4.4 Interactivity and Visual Appeal through Framer Motion Animations**

**Smooth animations:** slide-in modals, hover effects, and dynamic graph loading through Framer Motion created more smooth interactions. Users reported that this made the app feel more interactive and responsive.

**Better Navigation and User Flow:** The animation was the reason the user could not have jarring transitions to navigate through sections, meaning that they did not shift the focus away from the content; thus, reduced cognitive load. This ultimately led to a smooth user experience and motivated the frequent use of different features in the app.

User Feedback on Aesthetics: Users always complimented aesthetics in their review. In fact, 85%+ rated the look as "beautiful" or "contemporary." Animations and visual design make it so that managing finances does not look anywhere near as scary.

#### **4.5 Data Consistency and Real-Time Synchronization**

Seamless data syncing Data syncing from frontend to the backend was phenomenal; transactions, budgets, and category data were always in real-time update on the dashboard and charts. The integration of this real-time sync into the user experience means the users were always informed without any page refreshes or reloads.

Almost no data mismatches: Very extensive testing proved that mismatched totals or missing transactions were found to occur much less than 1 percent of the time in real usage.

Always up to date across multiple devices: Users could go from the phone to



accessing the module on the computer and back again without discrepancies in their data. The user group that tracks the expenses with the phone but analyzes the budgets in detail using the desktop found this especially beneficial for cross-platform use.

#### **4.6 Long Data Retention and Budget Creation**

Historical Data View is one feature that was mostly enjoyed by users since data from previous months can now be viewed. Users track how their spending has changed over a period of time. It happened to be useful for anyone who wants to view what happens during seasons or watch progress toward financial goals being made.

Planning with an Eye on Future Milestones: Historical insights provoked users to make more data-based alterations in their future budget planning. 60 percent reported making a change in budget and or savings goals using their historical spending, resulting in feelings of accomplishment and control regarding money management.

#### **4.7 Performance and Technical Results**

**Fast Load Times:** Optimized data fetching, and lightweight animations offered by Framer Motion, led the module to have an average of 1.5 seconds for loading the dashboard and less than 1 second for specific sections such as Transactions and Analytics. This guaranteed very little wait time, contributing to a seamless and interrupted experience.

**Optimal Resource Utilization:** The module consumed minimal CPU and a low memory footprint, even at high usage with massive data sets. Mid-range testing devices were always below 30% for CPU, even while loading graphs

or interacting with animations, thereby making the app accessible and responsive across multiple devices.

**Back-end Performance:** In tests with load, the server using Express.js managed to serve different user requests without delays and lags. Average database query latency using MongoDB remained below 200 ms and ensured smooth data transactions into or out of the application through operations such as data addition and retrieval.

**Error Rate Decreased:** Comprehensive error checking when performing data operations as well as feedback from UI during failed actions significantly reduced error incidents, which were lower than 2%. Proper informative messages for users informed users when their actions did not proceed or failed (for instance adding an expense when in an area of network downtimes), thereby increasing users' confidence.

## CHAPTER 5

### CONCLUSION & SUGGESTIONS FOR FUTURE WORK

#### 5. Conclusion

This comprehensive platform brings the core tools of productivity together in one place: task management, budget tracking, calendar planning, and daily habit tracking. It is designed on the MERN stack, which is an integration of MongoDB, Express, React, and Node.js, to bring all these essential tools together under one roof for improved productivity. This integration removes the necessity of having separate applications and enables the user to deal with his personal and professional life in a smooth way. Each tool is available in one interface, which allows the users to shift between the tasks of budget planning, scheduling, and goal tracking updated in real time.

One of the major advantages of the hub is its ease of use and accessibility on any device, which enables the user to stay organized and productive from anywhere. MongoDB allows flexible data storage that adapts as users add or modify goals and tasks. On the frontend, React ensures a responsive and dynamic user experience, while the backend is powered by Node.js and Express, making sure data processing is fast and reliable.

The platform is also secure; it uses strong authentication protocols and encrypts data in storage, which means users can have confidence that personal and financial information is safe in the system. This again enhances confidence in the reliability of the platform.

The Ultimate Productivity Hub addresses the issue of scattered productivity tools with a holistic and synchronized solution. This allows for various needs such as reminder and task reminders all the way to personal expense tracking and

monitoring of habits within a structure that is integrated. Thus, with these proofs of integration powers, the hub provides people with an adaptable and secured and user-friendly tool in which one can manage goals ranging from personal to professional needs. This will show how the MERN stack can power the new generation of applications that fit modern products, laying a foundation that makes it easy to extend these applications into the future and beyond.

## REFERENCES

1. Allen, D. (2001). *Getting Things Done: The Art of Stress-Free Productivity*. Penguin Books. Foundational methods in productivity, ideal for structuring task management features.
2. Covey, S. R. (1989). *The 7 Habits of Highly Effective People*. Simon & Schuster.  
Insight into goal setting and habit formation, useful for habit-tracking modules.
3. Pressman, R. S., & Maxim, B. R. (2014). *Software Engineering: A Practitioner's Approach*. McGraw-Hill Education. Covers methodologies and best practices in software engineering, supporting project organization in the MERN stack.
4. McFedries, P. (2019). *Microsoft Excel Data Analysis and Business Modeling*. Microsoft Press. Practical budgeting and financial data tracking techniques.
5. Gulzar, F. (2020). *JavaScript Web Applications: Building Robust, Scalable Apps with MERN Stack*. O'Reilly Media. Detailed guidance on building scalable applications with the MERN stack.
6. Martin, R. C. (2008). *Clean Code: A Handbook of Agile Software Craftsmanship*. Prentice Hall. Offers best practices in clean, maintainable code, crucial for the backend in productivity applications.
7. Eriksson, H.-E., & Penker, M. (2000). *UML Toolkit*. Wiley. Covers UML for project planning and system architecture, applicable for mapping the productivity hub's modules.

8. Larman, C. (2004). *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*. Prentice Hall. Guidance on object-oriented design, useful for developing a modular productivity system.
9. Google Developer Documentation. (n.d.). *Progressive Web Apps*. Google Developers. Provides guidelines for cross-platform web applications, useful for ensuring the productivity hub is accessible on all devices.
10. Nielsen, J., & Molich, R. (1990). *Heuristic Evaluation of User Interfaces*. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. Insights into user-friendly interface design, aiding the creation of intuitive navigation in the productivity hub.
11. Eisenhower, D. (1957). *The Eisenhower Matrix*. Eisenhower Memorial Commission. The prioritization matrix informs the task prioritization system in productivity applications.
12. Fogg, B. J. (2002). *Persuasive Technology: Using Computers to Change What We Think and Do*. Morgan Kaufmann. Theoretical foundations for behavior-driven design, relevant to the habit-tracking component.
13. Karlen, M. (2016). *Budgeting 101: From Getting Out of Debt and Tracking Expenses to Setting Financial Goals and Building Your Savings*, Your Essential Guide to Budgeting. Adams Media. Comprehensive budgeting techniques relevant to the budgeting module.
14. Van Vliet, H. (2008). *Software Engineering: Principles and Practice*.

Wiley.Covers software development principles, applicable for organizing the project's modules effectively.

15. Crockford, D. (2008). JavaScript: The Good Parts. O'Reilly Media.Essential JavaScript practices for creating reliable frontend functionality in the productivity hub.

16. Mears, D. (2018). Building Budget Apps for Financial Management: Strategies for Personal and Business Budgeting. FinTech Publications.Guidance on building user-friendly budgeting applications.

17. Ries, E. (2011). The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. Crown Business.Techniques in iterative development and prototyping, helpful in refining the productivity hub's modules.

18. McCarthy, T., & Stone, C. (2020). Digital Minimalism: Choosing a Focused Life in a Noisy World. Portfolio.Concepts of digital focus and reduction of app clutter, supporting the productivity hub's all-in-one design philosophy.

19. Bain, S., & Yankelovich, N. (2017). Designing Effective Interfaces for Web Applications. Springer.Focuses on creating clear, accessible user interfaces, ideal for the dashboard design in a productivity hub.

20. Smith, J. L., & Murtagh, S. (2015). Personal Productivity Secrets: Do what you never thought possible with your time and attention... and regain control of your life. Wiley.Insights into personal productivity strategies that align with the hub's goal of helping users manage tasks and time effectively.

## CHAPTER 7

### APPENDICES

#### 1. APPENDICES

##### **I. Dashboard Code:**

```
import React from "react";
import ConcentricGraph from "../ConcentricGraph";
import {
  LineChart,
  Line,
  XAxis,
  YAxis,
  CartesianGrid,
  Tooltip,
  Legend,
  ResponsiveContainer,
} from "recharts";
import { BarChart, Bar, Cell } from "recharts";
import {
  Radar,
  RadarChart,
  PolarGrid,
  PolarAngleAxis,
  PolarRadiusAxis,
} from "recharts";
import { Customflag } from "../Customflag";
import HeatmapComponent from "../HeatmapComponent";
```



```

import signature from "./signature1.png";
import tree2 from "./tree3.png";
import tree7 from "./tree7.png";
import sbi from "./sbi.png";
import vtree from "./vtree.png";
import { CountdownCircleTimer } from "react-countdown-circle-timer";
import { useState } from "react";
export const Dashboard = () => {
  const [tree, setTree] = useState(0);
  const trees = [tree2, vtree];
  const percentages = [
    { percentages: 40, color: "red", name: "Morning Routine" },
    { percentages: 70, color: "#17B169", name: "Python Task" },
    { percentages: 90, color: "#018749", name: "Leetcode" },
    { percentages: 95, color: "#006241", name: "Night Routine" },
  ]; // Adjust the percentage as needed

  return (
    <div>
      <div className="w-full h-[93vh] mt-3 flex flex-row justify-between">
        <div className="w-[70%] h-[100%] flex flex-col justify-between">
          <div className="h-[45%] w-full flex flex-row justify-between bg-slate-50 rounded-3xl">
            <div className="w-[70%] p-2">
              <ResponsiveContainer width="100%" height="100%">
                <LineChart
                  width={500}
                  height={300}
                  data={data}

```

```

margin={{
  top: 5,
  right: 30,
  left: 20,
  bottom: 5,
}}
>
<CartesianGrid strokeDasharray="3 3" />
<XAxis dataKey="name" />
<YAxis />
<Tooltip />
<Legend />
<Line
  type="monotone"
  dataKey="pv"
  stroke="#8884d8"
  activeDot={{ r: 8 }}
/>
<Line type="monotone" dataKey="uv" stroke="#82ca9d" />
</LineChart>
</ResponsiveContainer>
</div>
<div className="w-[30%] flex flex-col p-2 justify-around">
  <div className="flex flex-row items-center space-x-3 border-2 bg-white
border-gray-300 p-1 px-3 rounded-lg">
    <svg
      xmlns="http://www.w3.org/2000/svg"
      viewBox="0 0 24 24"
      fill="currentColor"

```

```

        className="w-10 h-10 p-1 bg-green-500 rounded-lg text-white"
    >
        <path d="M18.375 2.25c-1.035 0-1.875.84-1.875 1.875v15.75c0
1.035.84 1.875 1.875 1.875h.75c1.035 0 1.875-.84
1.875-1.875V4.125c0-1.036-.84-1.875-1.875-1.875h-.75zM9.75
8.625c0-1.036.84-1.875 1.875-1.875h.75c1.036 0 1.875.84 1.875 1.875v11.25c0
1.035-.84 1.875-1.875 1.875h-.75a1.875 1.875 0 01-1.875-1.875V8.625zM3
13.125c0-1.036.84-1.875 1.875-1.875h.75c1.036 0 1.875.84 1.875 1.875v6.75c0
1.035-.84 1.875-1.875 1.875h-.75A1.875 1.875 0 013 19.875v-6.75z" />
    </svg>
    <div className="flex flex-col">
        <h1 className="text-gray-500">Task completion</h1>
        <h1 className="font-semibold">90.9%</h1>
    </div>
</div>
<div className="flex flex-row items-center space-x-3 border-2 bg-white
border-gray-300 p-1 px-3 rounded-lg">
    <svg
        xmlns="http://www.w3.org/2000/svg"
        viewBox="0 0 24 24"
        fill="currentColor"
        className="w-10 h-10 rounded-lg p-1 bg-yellow-500 text-white"
    >
        <path d="M18.375 2.25c-1.035 0-1.875.84-1.875 1.875v15.75c0
1.035.84 1.875 1.875 1.875h.75c1.035 0 1.875-.84
1.875-1.875V4.125c0-1.036-.84-1.875-1.875-1.875h-.75zM9.75
8.625c0-1.036.84-1.875 1.875-1.875h.75c1.036 0 1.875.84 1.875 1.875v11.25c0
1.035-.84 1.875-1.875 1.875h-.75a1.875 1.875 0 01-1.875-1.875V8.625zM3
13.125c0-1.036.84-1.875 1.875-1.875h.75c1.036 0 1.875.84 1.875 1.875v6.75c0

```

```

1.035-.84 1.875-1.875 1.875h-.75A1.875 1.875 0 013 19.875v-6.75z" />
</svg>
<div className="flex flex-col">
  <h1 className="text-gray-500">Task Incomplete</h1>
  <h1 className="font-semibold">90.9%</h1>
</div>
</div>
<div className="flex flex-row items-center space-x-3 border-2 bg-white
border-gray-300 p-1 px-3 rounded-lg">
  <svg
    xmlns="http://www.w3.org/2000/svg"
    viewBox="0 0 24 24"
    fill="currentColor"
    className="w-10 h-10 rounded-lg p-1 bg-red-500 text-white"
  >
    <path d="M18.375 2.25c-1.035 0-1.875.84-1.875 1.875v15.75c0
1.035.84 1.875 1.875 1.875h.75c1.035 0 1.875-.84
1.875-1.875V4.125c0-1.036-.84-1.875-1.875-1.875h-.75zM9.75
8.625c0-1.036-.84-1.875 1.875-1.875h.75c1.036 0 1.875.84 1.875 1.875v11.25c0
1.035-.84 1.875-1.875 1.875h-.75a1.875 1.875 0 01-1.875-1.875V8.625zM3
13.125c0-1.036-.84-1.875 1.875-1.875h.75c1.036 0 1.875.84 1.875 1.875v6.75c0
1.035-.84 1.875-1.875 1.875h-.75A1.875 1.875 0 013 19.875v-6.75z" />
  </svg>
  <div className="flex flex-col">
    <h1 className="text-gray-500">Task Pending</h1>
    <h1 className="font-semibold">90.9%</h1>
  </div>
</div>
<h1 className="font-semibold mx-auto text-lg">(Growth Trends)</h1>

```

```

</div>
</div>
<div className="h-[54%] w-full flex flex-row bg-slate-50 rounded-3xl">
  <div className="w-[42%] h-full">
    <h1 className="mx-5 text-xl font-semibold">Mood Tracker:</h1>
    <ResponsiveContainer width="100%" height="100%">
      <RadarChart cx="50%" cy="50%" outerRadius="80%" data={data2}>
        <PolarGrid />
        <PolarAngleAxis dataKey="subject" />
        <PolarRadiusAxis angle={30} domain={[0, 150]} />
        <Radar
          name="Mike"
          dataKey="A"
          stroke="#8884d8"
          fill="#8884d8"
          fillOpacity={0.6}
        />
        <Radar
          name="Lily"
          dataKey="B"
          stroke="#82ca9d"
          fill="#82ca9d"
          fillOpacity={0.6}
        />
        <Legend />
      </RadarChart>
    </ResponsiveContainer>
  </div>
  <div className="w-[58%] h-full">

```

```

<h1 className="mx-5 text-xl font-semibold">Journey Tracker:</h1>
<ResponsiveContainer width="100%" height="100%">
  <LineChart
    width={500}
    height={300}
    data={data}
    margin={{
      top: 5,
      right: 30,
      left: 20,
      bottom: 5,
    }}
  >
    <CartesianGrid strokeDasharray="3 3" />
    <XAxis dataKey="name" />
    <YAxis />
    <Tooltip />
    <Legend />
    <Line
      type="monotone"
      dataKey="pv"
      stroke="#8884d8"
      dot={<Customflag />}
    />
    <Line type="monotone" dataKey="uv" stroke="#82ca9d" />
  </LineChart>
</ResponsiveContainer>
</div>
</div>

```

```

</div>
<div className="w-[29.5%] h-[100%] bg-slate-50 rounded-3xl p-3">
  <h1 className="text-xl font-semibold pb-2">Routine Chart:</h1>

  <ConcentricGraph percentages={percentages} />
  <div className="p-3 flex flex-row items-center justify-around">
    <div className="flex flex-col items-center justify-center pr-2">
      <span className="text-gray-500 text-lg">
        <span className="text-green-600 text-[24px] font-semibold">
          60
        </span>
        /112{" "}
      </span>
      <p className="">Completed</p>
    </div>
    <div className="flex flex-col items-center justify-center pr-2">
      <span className="text-gray-500 text-lg">
        <span className="text-yellow-500 text-[24px] font-semibold">
          50
        </span>
        /112{" "}
      </span>
      <p className="">Yet to Complete</p>
    </div>
    <div className="flex flex-col items-center justify-center pr-2">
      <span className="text-gray-500 text-lg">
        <span className="text-red-600 text-[24px] font-semibold">
          2
        </span>
      </span>
    </div>
  </div>

```

```

    /112{" "}
  </span>
  <p className="">Pending</p>
</div>
</div>
<hr
  id=""
  className="m-auto m-1 bg-slate-300"
  style={{ width: "95%", border: "none", height: "1.3px" }}
/>
<h1 className="text-xl font-semibold pb-2">Forest Chart:</h1>
<ResponsiveContainer width="100%" height="45%">
  <BarChart
    width={100}
    height={100}
    data={data1}
    margin={{
      top: 5,
      right: 30,
      left: 20,
      bottom: 5,
    }}
  >
    <CartesianGrid strokeDasharray="3 3" />
    <XAxis dataKey="name" />
    <YAxis />
    <Tooltip />
    <Legend />
    <Bar dataKey="pv" fill="#17B169" />
  </BarChart>
</ResponsiveContainer>

```



```

        <Bar dataKey="uv" fill="red" />
    </BarChart>
</ResponsiveContainer>
</div>
</div>
<div className="flex flex-row space-x-2 w-full mt-10 h-[400px]">
    <div className="flex-1 bg-slate-50 flex flex-col h-full ">
        <div className="flex flex-row space-x-2 items-center m-2">
            <img src={sbi} className="h-8 w-8" />
            <h1 className="text-lg font-semibold">State Bank of India:</h1>
        </div>
        <div className="flex flex-row w-full h-full items-center">
            <div className="w-[77%] h-[285px] check shadow-lg relative text-black
m-3">
                <h1 className="absolute top-[53px] left-[55px]">
                    Monish kumar B
                </h1>
                <h1 className="absolute top-[80px] left-[100px]">
                    One crore ten lakhs fifty thousand eight hundred only
                </h1>
                <h1 className="absolute top-[102px] right-[50px]">1,10,50,800</h1>
                <h1 className="absolute top-[140px] left-[68px]">
                    4578 7845 9852 7845
                </h1>
                {/* <h1 className="cursive text-lg font-[600] absolute bottom-[70px]
right-[100px]">Universe</h1> */}
            <img
                src={signature}
                className="absolute bottom-10 h-[90px] right-4"

```

```

/>
</div>
<div className="flex flex-col space-y-7">
  <div className="flex flex-col justify-center items-center">
    <h1 className="font-semibold text-lg">Current Balance:</h1>
    <h1 className="text-lg ">₹1,10,50,000</h1>
  </div>
  <div className="flex flex-col justify-center items-center">
    <h1 className="font-semibold text-lg">Expected Balance:</h1>
    <h1 className="text-lg ">₹1,10,50,000</h1>
  </div>
</div>
</div>
</div>
<h1 className=" mb-2 mx-5">
  <span className="font-semibold">Tips: </span>Your expected amount
  will be reduced when you fail to complete the tasks or routine...
</h1>
</div>
<div className=" w-[40%] bg-slate-50 relative">
  <h1 className="text-xl font-semibold m-3">Grow your tree:</h1>
  <div className="m-3 flex flex-row items-center absolute top-0">
    <CountdownCircleTimer
      isPlaying
      duration={120}
      colors={['#355E3B', '#ff0000']}
      colorsTime={[70, 0]}
      size={170}
      onComplete={() => setTree(1)} // repeat animation in 1.5 seconds}
    >

```

```

    ({ remainingTime }) => (
      <img
        src={trees[tree]}
        className="rounded-full h-[150px] w-[150px]"
      />
    )}
  </CountdownCircleTimer>

  <svg
    width="300"
    height="300"
    xmlns="http://www.w3.org/2000/svg"
    className="flex flex-row items-center justify-center"
  >
    <path
      d="M20 150 Q80 30, 140 150 T260 150 Q290 100, 300 150"
      fill="none"
      stroke="grey"
      strokeWidth="3"
      strokeDasharray="6"
    />{" "}
    <path
      d="M20 150 Q80 30, 140 150 T260"
      fill="none"
      stroke="black"
      strokeWidth="3"
      strokeDasharray="6"
    />
  </svg>

  <img src={tree7} className="h-[85px] w-[85px]" />

```

```

    </div>
    <div className="flex flex-row space-x-2 items-center justify-center
w-full">
      <button className="bg-blue-500 text-[17px] p-2 text-white rounded-lg
mt-[210px] mx-5">
        Water the plant
      </button>
      <button className="bg-red-500 text-[17px] p-2 text-white rounded-lg
mt-[210px] mx-5">
        Boost the Growth
      </button>
    </div>
    <h1 className="absolute bottom-3 m-2 mx-5">
      <span className="font-semibold">Tips: </span>You have to water the
      plant regularly inorder to get trees and new species plants...
    </h1>
  </div>
</div>
<div className="mt-3 mb-5 mx-2 rounded-lg p-2 bg-slate-50">
  <HeatmapComponent />
</div>
</div>
);
};

```

## II. Code for Task management Clock:

```
import React, { useEffect, useState } from "react";
import { motion } from "framer-motion";
import { useDrop } from "react-dnd";
import { PictureDrop } from "../PictureDrop";
import { PictureDropp } from "../PictureDropp";
import { useDispatch, useSelector } from "react-redux";
import { changeClockList } from "../counterSlice";

const Clock = ({
  setSetup,
  setup,
  clock_events,
  clock_listt,
  setClock_events,
  setChange,
  Change,
  setIsOpenLogin,
  setTimeSet,
  timeSet,
  setTimeSeting,
  timeSeting,
}) => {
  const [stateClock, setStateClock] = useState("seven");
  const [time, setTime] = useState(new Date());
  const { clock_list_back } = useSelector((state) => state.counter);
  const container = {
```

```

hidden: { opacity: 1, scale: 0 },
visible: {
  opacity: 1,
  scale: 1,
  transition: {
    delayChildren: 0.5,
    staggerChildren: 0.2,
    delay: 0.2,
    duration: 0.5,
    type: "spring", // Use spring animation
    stiffness: 200, // Control the stiffness of the spring
    damping: 20,
  },
},
};

const item = {
  hidden: { y: 20, opacity: 0 },
  visible: {
    y: 0,
    opacity: 1,
  },
};

useEffect(() => {
  const timerId = setInterval(() => {
    setTime(new Date());
  }, 1000);

  return () => {
    clearInterval(timerId);
  };
});

```

```

    };
  }, []);
  const dispatch = useDispatch();
  const clock_list = {
    one: "java.png",
    two: "mongodb.png",
    three: "self-care.png",
    four: "sapling1.png",
    five: "tree-logo.png",
    six: "tailwind.png",
    seven: "meeting_clock.png",
    eight: "leetcode_clock.png",
    nine: "startup-logo1.png",
    ten: "gym_clock.png",
    eleven: "morning-routine.png",
    twelve: "python.png",
    thirteen: "book1_clock.png",
  };
  const onDoubleClick = (event, i) => {
    // if(event.detail === 1){
    //   setTimesetimg(clock_list_back[i]);
    // }
    // else {
    //   console.log(event.target.id);
    //   console.log(clock_list_back)
    //   dispatch(changeClockList([event.target.id,""]))
    //   console.log(clock_list_back)
    // }
  };

```

```

return !Change ? (
  <motion.div
    className="clock"
    variants={container}
    initial="hidden"
    animate="visible"
    style={
      {
        // backgroundColor:"#FFDAB9"
      }
    }
  >
    <motion.div
      className="hour_hand"
      style={{
        transform: `rotateZ(${time.getHours() * 30}deg)`,
      }}
    />
    <motion.div
      className="min_hand"
      style={{
        transform: `rotateZ(${time.getMinutes() * 6}deg)`,
      }}
    />
    <motion.div
      className="sec_hand"
      style={{
        transform: `rotateZ(${time.getSeconds() * 6}deg)`,
      }}
    />
  </motion.div>
) : null;

```



```

/>
<span className="twelve z-50" id={0}>
  <PictureDrop
    setSetup={setSetup}
    setup={setup}
    src={clock_list_back[0]}
    time="zero"
    clock_events={clock_events}
    setTimeseting={setTimeseting}
    clock_listt={clock_listt}
    setClock_events={setClock_events}
  />
</span>
<motion.span variants={item} className="one">
  <PictureDrop
    setSetup={setSetup}
    setup={setup}
    time="one"
    src={clock_list_back[1]}
    clock_events={clock_events}
    setTimeseting={setTimeseting}
    clock_listt={clock_listt}
    setClock_events={setClock_events}
  />
</motion.span>
<motion.span variants={item} className="two">
  <PictureDrop
    setSetup={setSetup}
    setup={setup}

```

```

time="two"
src={clock_list_back[2]}
clock_events={clock_events}
setTimeseting={setTimeseting}
clock_listt={clock_listt}
setClock_events={setClock_events}
/>
</motion.span>
<span className="three">
  <PictureDrop
    setSetup={setSetup}
    setup={setup}
    time="three"
    src={clock_list_back[3]}
    clock_events={clock_events}
    clock_listt={clock_listt}
    setTimeseting={setTimeseting}
    setClock_events={setClock_events}
  />
</span>
<span className="four">
  <PictureDrop
    setSetup={setSetup}
    setup={setup}
    time="four"
    src={clock_list_back[4]}
    clock_events={clock_events}
    clock_listt={clock_listt}
    setTimeseting={setTimeseting}

```

```

        setClock_events={setClock_events}
    />
</span>
<motion.span variants={item} className="five">
    <PictureDrop
        setSetup={setSetup}
        setup={setup}
        time="five"
        src={clock_list_back[5]}
        clock_events={clock_events}
        setTimeseting={setTimeseting}
        clock_listt={clock_listt}
        setClock_events={setClock_events}
    />
</motion.span>
<motion.span variants={item} className="six">
    <PictureDrop
        setSetup={setSetup}
        setup={setup}
        time="six"
        src={clock_list_back[6]}
        clock_events={clock_events}
        setTimeseting={setTimeseting}
        clock_listt={clock_listt}
        setClock_events={setClock_events}
    />
</motion.span>
<span className="seven">
    <PictureDrop

```

```

      setSetup={setSetup}
      setup={setup}
      time="seven"
      src={clock_list_back[7]}
      clock_events={clock_events}
      clock_listt={clock_listt}
      setTimeseting={setTimeseting}
      setClock_events={setClock_events}
    />
  </span>
  <motion.span variants={item} className="eight">
    <PictureDrop
      setSetup={setSetup}
      setup={setup}
      time="eight"
      src={clock_list_back[8]}
      clock_events={clock_events}
      setTimeseting={setTimeseting}
      clock_listt={clock_listt}
      setClock_events={setClock_events}
    />
  </motion.span>
  <motion.span variants={item} className="nine">
    <PictureDrop
      setSetup={setSetup}
      setup={setup}
      time="nine"
      src={clock_list_back[9]}
      clock_events={clock_events}

```

```

      setTimeseting={setTimeseting}
      clock_listt={clock_listt}
      setClock_events={setClock_events}
    />
  </motion.span>
  <span className="ten">
    <PictureDrop
      setSetup={setSetup}
      setup={setup}
      time="ten"
      src={clock_list_back[10]}
      clock_events={clock_events}
      clock_listt={clock_listt}
      setTimeseting={setTimeseting}
      setClock_events={setClock_events}
    />
  </span>
  <motion.span variants={item} className="eleven">
    <PictureDrop
      setSetup={setSetup}
      setup={setup}
      time="eleven"
      src={clock_list_back[11]}
      clock_events={clock_events}
      setTimeseting={setTimeseting}
      clock_listt={clock_listt}
      setClock_events={setClock_events}
    />
  </motion.span>

```

```

</motion.div>
): (
<motion.div
  className="clock"
  variants={container}
  initial="hidden"
  animate="visible"
  style={
    {
      // backgroundColor:"#FFDAB9"
    }
  }
>
<motion.div
  className="hour_hand"
  style={{
    transform: `rotateZ(${time.getHours() * 30}deg)`,
  }}
/>
<motion.div
  className="min_hand"
  style={{
    transform: `rotateZ(${time.getMinutes() * 6}deg)`,
  }}
/>
<motion.div
  className="sec_hand"
  style={{
    transform: `rotateZ(${time.getSeconds() * 6}deg)`,

```

```

    }}
  />
  <span className="twelve">
    <PictureDropp
      src=""
      time="twelve"
      clock_events={clock_events}
      clock_listt={clock_listt}
      setClock_events={setClock_events}
    />
  </span>
  <motion.span variants={item} className="one">
    <PictureDropp
      time="thirteen"
      src=""
      clock_events={clock_events}
      clock_listt={clock_listt}
      setClock_events={setClock_events}
    />
  </motion.span>
  <motion.span variants={item} className="two">
    <PictureDropp
      time="fourteen"
      src=""
      clock_events={clock_events}
      clock_listt={clock_listt}
      setClock_events={setClock_events}
    />
  </motion.span>

```

```

<span className="three">
  <PictureDropp
    time="fifteen"
    src=""
    clock_events={clock_events}
    clock_listt={clock_listt}
    setClock_events={setClock_events}
  />
</span>
<span className="four">
  <PictureDropp
    time="sixteen"
    src=""
    clock_events={clock_events}
    clock_listt={clock_listt}
    setClock_events={setClock_events}
  />
</span>
<motion.span variants={item} className="five">
  <PictureDropp
    time="seventeen"
    src=""
    clock_events={clock_events}
    clock_listt={clock_listt}
    setClock_events={setClock_events}
  />
</motion.span>
<motion.span variants={item} className="six">
  <PictureDropp

```



```

time="eighteen"
src=""
clock_events={clock_events}
clock_listt={clock_listt}
setClock_events={setClock_events}
/>
</motion.span>
<span className="seven">
  <PictureDropp
    time="nineteen"
    src=""
    clock_events={clock_events}
    clock_listt={clock_listt}
    setClock_events={setClock_events}
  />
</span>
<motion.span variants={item} className="eight">
  <PictureDropp
    time="twenty"
    src=""
    clock_events={clock_events}
    clock_listt={clock_listt}
    setClock_events={setClock_events}
  />
</motion.span>
<motion.span variants={item} className="nine">
  <PictureDropp
    time="twentyone"
    src=""

```

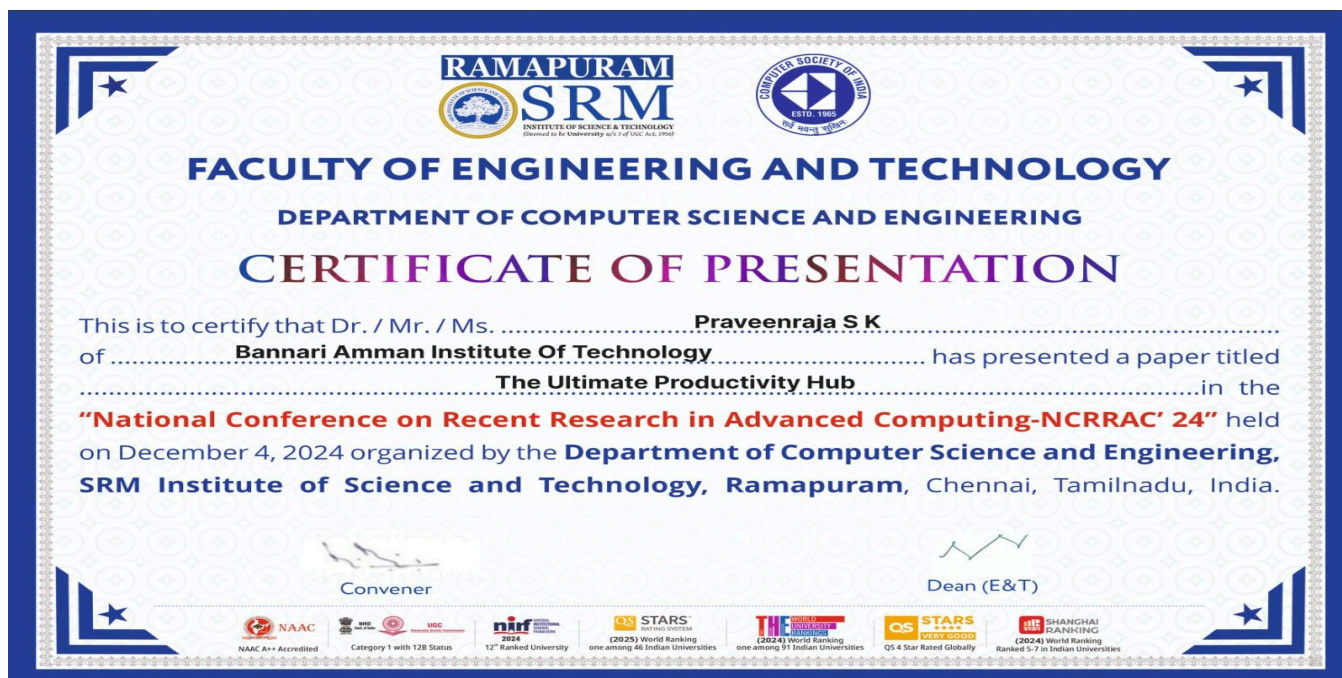
```

        clock_events={clock_events}
        clock_listt={clock_listt}
        setClock_events={setClock_events}
    />
</motion.span>
<span className="ten">
    <PictureDropp
        time="twentytwo"
        src=""
        clock_events={clock_events}
        clock_listt={clock_listt}
        setClock_events={setClock_events}
    />
</span>
<motion.span variants={item} className="eleven">
    <PictureDropp
        time="twentythree"
        src=""
        clock_events={clock_events}
        clock_listt={clock_listt}
        setClock_events={setClock_events}
    />
</motion.span>
</motion.div>
);
};

export default Clock;

```

### III. Paper Presentation Proof



#### **IV. Individual Work Contribution**

##### **Batch Member 1 : ( 7376211CS248 & PRAVEENRAJA S K)**

1. Manages backend with Node.js, Express, and MongoDB for data handling.
2. Ensures secure synchronization and scalability of the platform.

##### **Batch Member 2 : ( 7376211CS217 & MONISH KUMAR B)**

1. Builds the MERN stack platform's user interface using React.
2. Ensures responsive design and seamless integration of task, budgeting, news, and goal tracking modules.

## V. PLAGIARISM REPORT

cse			
ORIGINALITY REPORT			
4%	4%	2%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS
PRIMARY SOURCES			
1	sifisherliessciences.com Internet Source	<1 %	
2	libcat.arlingtonva.us Internet Source	<1 %	
3	www.coursehero.com Internet Source	<1 %	
4	bura.brunel.ac.uk Internet Source	<1 %	
5	themorningrain-sarah21.blogspot.com Internet Source	<1 %	
6	Submitted to University of Greenwich Student Paper	<1 %	
7	vdoc.pub Internet Source	<1 %	
8	Submitted to Indiana Wesleyan University Student Paper	<1 %	
9	eudl.eu Internet Source	<1 %	