# TICK TASK-TO DO LIST

PROJECT REPORT

submitted by:


| | |
|---|---|
| **AARAV R JITH** | **ADR22CS001** |
| **ABIN T ABEY** | **ADR22CS005** |
| **ANANDHU A K** | **ADR22CS013** |
| **ROHIT KRISHNAN** | **ADR22CS056** |


to

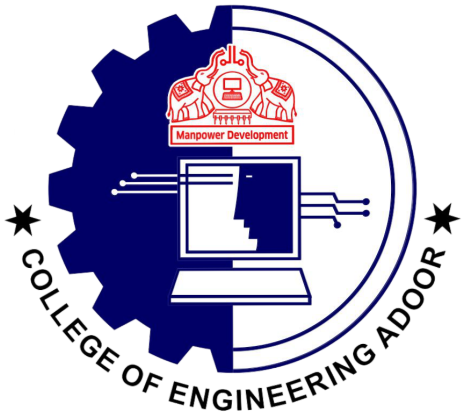
the APJ Abdul Kalam Technological University
in partial fulfillment of the requirements for the award of Degree

of

Bachelor of Technology
in
*Computer Science and Engineering*

Under the guidance of

**PROF. JAYALAKSHMI**

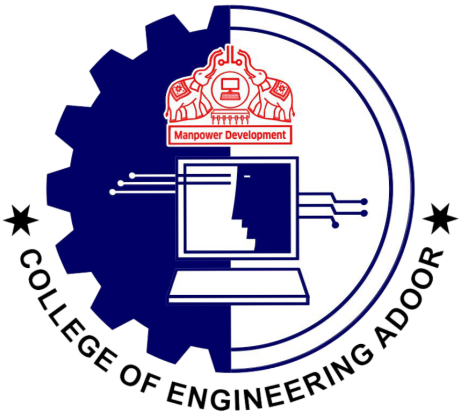




**Department of Computer Science and Engineering**
**College of Engineering Adoor**

**2023-2024**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**
**COLLEGE OF ENGINEERING, ADOOR**




# CERTIFICATE

This is to certify that the report entitled **A TICK TASK A TASK MANAGEMENT SYSTEM** submitted by **AARAV R JITH (ADR22CS001), ABIN T ABEY (ADR22CS005), ANANDHU A K (ADR22CS013),ROHIT KRISHNAN (ADR22CS056)** to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineering is a bonafide record of the project work carried out by them under my guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Countersigned by:

| **Project Guide** | **Project Coordinator** | **Head of the Dept** |
|---|---|---|
| Prof. Honey M | Prof. Prof Jayalakshmi | Dr. Vinod P R |

# DECLARATION

We undersigned hereby declare that the project report **A Tick task a task management system** submitted for partial fulfillment of the requirements for the award of the degree of Bachelor of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by us under supervision of **Prof Jayalakshmi**. This submission represents our ideas in our own words and where ideas or words of others have been included, we have adequately and accurately cited and referenced the original sources. We also declare that we have adhered to the ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed as the basis for the award of any degree, diploma, or similar title of any other University.

**Place:** Manakala, Adoor
**Date:** 02 / 04 / 25

**AARAV R JITH**
**ABIN T ABEY**
**ANANDHU A K**
**ROHIT KRISHNAN**

# Contents

# ACKNOWLEDGEMENT

We extend our profound gratitude to God almighty for His blessings. Our heartfelt thanks to **Dr. K Sunilkumar**, Principal, College of Engineering, Adoor, and **Prof. Vinod**, Head of Department, for their exceptional support throughout our journey. We reserve special appreciation for **Prof. JAYALAKSHMI** whose mentorship has been invaluable. Our gratitude extends to the dedicated faculty, supportive friends, and loving family members who stood by us. We also acknowledge the invaluable contributions of researchers in Task Management system. Lastly, we express our sincere thanks to everyone involved; your unwavering support has been the cornerstone of our success. We earnestly hope that our endeavors in Task Management Systems will contribute positively to the betterment of society.

With heartfelt thanks,
**AARAV R JITH**
**ABIN T ABEY**
**ANANDHU A K**
**ROHIT KRISHNAN**

# ABSTRACT

Tick Task is an innovative task management system designed to enhance productivity, streamline organization, and improve time management for individuals and teams. Unlike traditional to-do list applications, Tick Task integrates task categorization, priority setting, scheduling, and real-time collaboration to provide a flexible and user-friendly experience. The system incorporates reminders, progress tracking, and gamification elements to keep users motivated and engaged. Additionally, Tick Task aims to simplify team coordination by enabling seamless task sharing, assignment, and tracking, making it ideal for both personal and professional use. The project follows a structured development approach, with a focus on usability, efficiency, and adaptability. Future enhancements include AI-based task recommendations, mobile app development, and integration with third-party productivity tools. By offering a customizable, intuitive, and collaborative platform, Tick Task aspires to be a comprehensive and reliable solution for modern task management needs.

# LIST OF FIGURES

# Chapter 1

# INTRODUCTION

Tick task is a task management system simply a to do list which is simple yet effective tool for organizing tasks, managing time, and improving productivity. In today's fast-paced world, individuals and teams struggle with keeping track of their responsibilities efficiently. Tick Task is designed to solve this problem by providing a structured, user-friendly platform for task management. Our goal is to create an application that helps users plan, prioritize, and execute tasks seamlessly. Unlike traditional to-do lists, Tick Task offers real-time collaboration, task categorization, scheduling, and reminders to ensure users stay on track. It also includes motivational features like progress tracking and gamification to keep users engaged. This project focuses on enhancing productivity by allowing users to break tasks into smaller, manageable steps while ensuring ease of access across multiple devices. Whether for personal use, teamwork, or professional project management, Tick Task aims to be an intuitive and efficient solution for task organization

## 1.1 MOTIVATION

The current paradigm in 3D object security employs selective encryption techniques, allowing specific content segments to remain accessible while encrypting the rest. While this approach achieves a delicate balance between security and accessibility, it paves the way for a myriad of challenges. The absence of hierarchical decryption poses a critical drawback, leading to security vulnerabilities, limited scalability, intricate key management, and suboptimal system performance. Recognizing these deficiencies, our project endeavors to redefine the landscape by introducing a format-compliant encryption method. This innovative approach not only rectifies the lack of hierarchical decryption but also serves as a comprehensive solution, addressing security concerns, enhancing scalability, simplifying key management, and optimizing overall system performance. The identified issues extend beyond mere technicalities, encompassing a spectrum of operational inefficiencies. The lack of hierarchical decryption not only exposes vulnerabilities but also hampers scalability, making it challenging to adapt to evolving data demands. Complex key management further exacerbates the situation, hindering smooth operation and contributing to suboptimal system performance. Our project, driven by a commitment to excellence, seeks to eliminate these challenges, ushering in a new era of 3D object security that prioritizes hierarchical decryption, mitigates vulnerabilities, ensures scalability, simplifies key management, and optimizes overall performance with a focus on interoperability.

## 1.2 PROBLEM DESCRIPTION

Despite the availability of numerous task management applications, many lack essential features that cater to modern users. Key issues include: Lack of Personalization, Many apps do not allow users to customize task categories, priorities, or views according to their workflow. Ineffective Reminders, Existing applications often have basic notification systems that do not effectively remind users about important deadlines. Limited Collaboration, Some apps are designed for individuals rather than teams, making it difficult to assign and track shared tasks efficiently. No Motivation Features, Users often struggle with consistency, and traditional to-do lists do not include features like progress tracking or rewards to encourage productivity. Complex Interfaces, Some task management solutions have overcomplicated interfaces, making them difficult to navigate for new users.

## 1.3 SCOPE OF PROPOSED SYSTEM

The scope of Tick Task extends beyond simple task management by providing a comprehensive and flexible solution for individuals and teams to plan, organize, and track their tasks efficiently. The system is designed to cater to a wide range of users, from students managing assignments to professionals handling multiple projects and teams coordinating collaborative work. With features such as task categorization, priority setting, reminders, and collaboration tools, Tick Task ensures that users can tailor their workflow to match their specific needs.Tick Task also incorporates gamification elements and progress tracking, which serve to boost motivation and encourage consistent productivity. Another critical aspect of the system is its collaborative features, which allow multiple users to work together on shared tasks. Overall, Tick Task aims to be more than just a to-do list—it is a tool that helps users take control of their tasks, improve time management, and enhance their workflow while maintaining a user-friendly and intuitive experience.

## 1.4 OBJECTIVES

The main objective of Tick Task is to create a powerful yet simple task management system that enhances productivity and task organization. It allows users to categorize tasks, set deadlines, and track progress, ensuring better organization. By helping users prioritize important tasks and manage their schedules efficiently, the system improves overall time management. Motivational features such as progress tracking and reminders keep users engaged, fostering consistent productivity. Additionally, Tick Task facilitates seamless collaboration by enabling teams to share, assign, and track tasks efficiently. By achieving these objectives, Tick Task aims to be a reliable, flexible, and effective task management solution for both individuals and teams.

## 1.5 PROJECT SCHEDULE

| Sl.no | ACTIVITY | DEC | JAN | FEB | MAR | APR |
|---|---|---|---|---|---|---|
| 1 | LITERATURE REVIEW | X | X | | | |
| 2 | FEASIBILITY STUDY | | X | | | |
| 3 | SYSTEM DESIGN | | X | X | | |
| 4 | IMPLEMENTATION | | | X | X | |
| 5 | TESTING | | | | X | |
| 6 | PROJECT REPORT | | | | X | |

Table 1.1: Project Schedule

# Chapter 2

# LITERATURE REVIEW

### 1. DEVELOPMENT OF TO-DO LIST AND MONETARY MANAGEMENT SYSTEM

by
Bong Siaw Wee
Department of Electrical Engineering
Politeknik Mukah
Mukah, Sarawak, Malaysia

The study titled **Development of To-Do List and Monetary Management System** proposes an enhanced to-do list application that supports group collaboration and financial tracking. Unlike conventional to-do list apps that only offer basic task management, this system allows users to:

- Share and update task lists among team members.
- Track financial expenditures and income associated with specific tasks.
- Integrate task management with an interactive calendar for better scheduling.
- Access and update tasks across multiple devices using cloud storage.

The system is particularly useful for families, project teams, and study groups who require a shared and real-time task management solution.

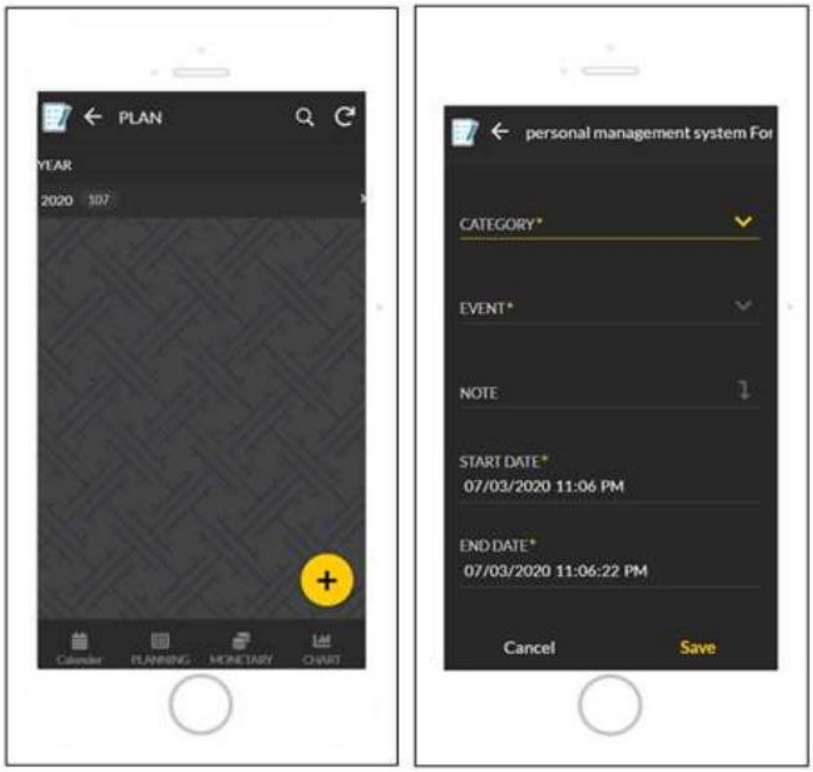



Figure 2.1: Press on the "+" button to add a new task to the "To-Do List" (Planning button)

## 2.1 METHODOLOGY

The application was developed using the **Rapid Application Development (RAD)** methodology, which focuses on quick prototyping and iterative improvements. The development process included:

1. Setting up a dedicated Google account for managing the cloud database.
2. Designing structured Google Sheets to store task and financial data.
3. Using AppSheet to create a graphical user interface for mobile and web access.
4. Enabling real-time updates and automatic synchronization across all devices.

This approach allowed for a flexible and easily customizable application that could be adapted to various user needs.

## 2.2 RESULTS AND DISCUSSION

The system was evaluated through a user study at Mukah Polytechnic, where students tested the application in a real-world scenario. The key findings include:

- Users found the app to be highly effective in managing group tasks.
- The integrated financial tracking feature helped users plan and monitor expenses.

- Real-time updates and cloud storage ensured seamless collaboration among team members.
- The interactive charts and tables provided clear insights into task progress and financial status.

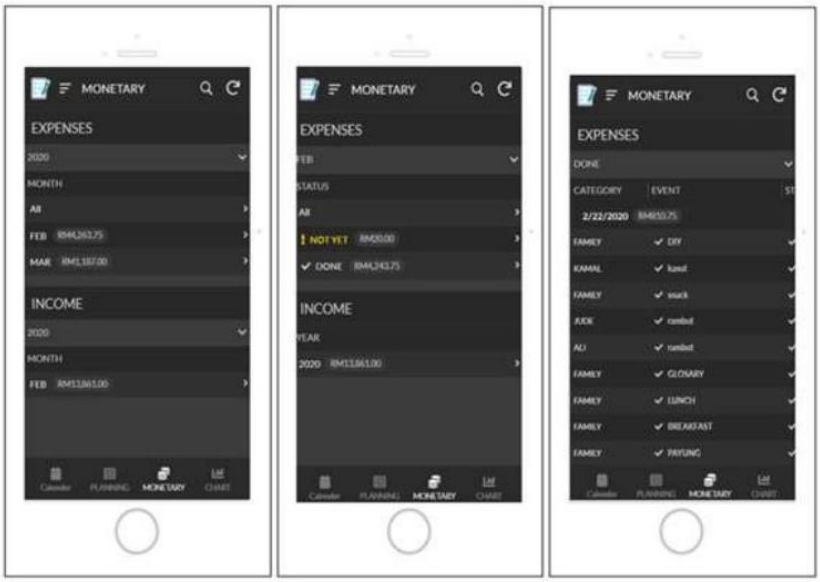



Figure 2.2: Task progress and financial tracking dashboard

## 2. TASKDO: A DAILY TASK RECOMMENDER SYSTEM

**by Dr. Mohammad Amin Kuhail , Nikhil sai ,Santosh Gurram**

The paper "TaskDo: A Daily Task Recommender System" by Kuhail and Gurram discusses the challenges individuals face in task accomplishment due to time constraints and inefficient planning. While many task management tools exist, they often lack guidance for optimal task management. To address this, the authors propose TaskDo, a task recommender system that suggests specific task types based on user history and other factors. The system also collects user feedback to improve its recommendations over time.

The introduction highlights that productivity is a common challenge for many, with employees working beyond scheduled hours and students struggling with organization. Effective planning can save time and improve task management. TaskDo aims to address the lack of tools that suggest specific tasks at specific times by recommending task types based on a user's history of completed tasks. The system considers variables like the day of the week, time of day, and task type (indoor/outdoor, intellectual/physical) to make recommendations. The related work section discusses existing task management applications like Wunderlist and Todoist, which help users organize tasks but do not provide task suggestions.

The authors also mention Timeful, an application that aimed to learn user habits and schedules to assist with planning, but it is no longer available. The paper also explores recommender systems, discussing collaborative filtering and content-based filtering, and hybrid models that combine both techniques to avoid issues like the cold start problem and over-specialization. The design section details TaskDo's system architecture, which includes modules for user tasks, task database, recommendations, and user feedback. The system categorizes tasks and considers dependent variables like day type and time of day. User feedback on task completion is collected to help the recommender system learn and provide better recommendations. The authors simulated a database to develop and evaluate the system.

The methodology section explains how TaskDo generates weekly plan recommendations by drilling down the Task DB, applying binary encoding, calculating correlations between task types and task completion satisfaction, and assigning rank scores to each task type. The two highest-ranked task types are then recommended for each day and time slot.

In the future work section, the authors discuss their plans to collect real-world data from volunteers to validate the system and make adjustments based on user feedback. They also aim to incorporate additional dependent variables like weather, task location, and task duration, and explore using machine learning classification algorithms to improve task recommendations

## 3. BRIDGING THE GAP BETWEEN TIME MANAGEMENT RESEARCH AND TASK MANAGEMENT APP DESIGN: A STUDY ON THE INTEGRATION OF PLANNING FALLACY MITIGATION STRATEGIES

by Yoana Ahmetoglu, Duncan P. Brumby and Anna L. Cox.

This study explores how personal task management (PTM) applications can better support time estimation accuracy by incorporating strategies from psychological research. The authors identify a major challenge: individuals often underestimate the time required to complete tasks due to a cognitive bias known as the planning fallacy. This leads to missed deadlines, increased

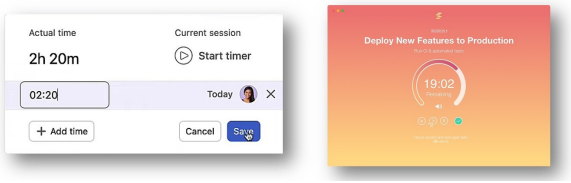



Figure 2.3: An illustration of time tracking in Asana (left) and pomodoro session in Serene (right)

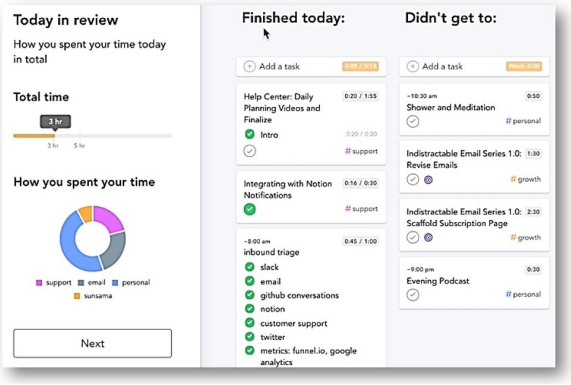



Figure 2.4: An illustration of time analytics in Sunsama's daily review ritual

stress, and reduced productivity. Despite the prevalence of PTM apps, few are designed with features that help users make accurate time estimates.

The research is structured into two key studies. Study 1 conducts a literature review to identify four strategies recommended in psychology for improving time estimation: 1.Duration Feedback – Providing users with regular and objective feedback on how long past tasks have taken to improve future estimates. 2. Distributional Data– Encouraging reliance on past data about similar tasks rather than intuitive judgment. 3. Task Breakdown – Breaking tasks into smaller, manageable steps to improve estimation accuracy. 4. Induced Neutrality – Encouraging a neutral, less optimistic perspective when estimating time. Study 2 then analyzes 47 popular PTM apps to assess how well they implement these strategies. The results show a major gap between theory and practice. While some apps provide time tracking and analytics (which could support duration feedback), they are often difficult to use or not designed for personal productivity. Features supporting task breakdown, like subtasks and templates, are common, but they do not always guide users to break down tasks in a way that reduces the planning fallacy. Distributional data and induced neutrality strategies were almost entirely absent from current apps.

The paper concludes that PTM app designers should incorporate features that explicitly address time estimation biases, such as proactive time tracking, AI-assisted task breakdowns, and visualization tools for reviewing past task durations. These improvements could significantly enhance productivity by helping users manage their time more realistically.

## 4. TOWEL: TOWARDS AN INTELLIGENT TO-DO LIST

by Kenneth Conley, James Carpenter

The paper outlines a comprehensive study focused on developing a robust and efficient task management system designed to improve productivity and streamline personal or team-based work. The authors begin by highlighting the increasing demand for task management solutions due to the complexity of modern work environments. They emphasize that effective task organization is crucial for ensuring deadlines are met and goals are achieved. The paper introduces a system designed to overcome common limitations seen in existing task management tools by improving user interface design, notification systems, and integration with third-party services.

The core architecture of the proposed system is built using a combination of modern web technologies to ensure scalability and responsiveness. The system's backend leverages a relational database for structured data management, while the frontend is designed to provide a seamless user experience. The authors describe key design principles such as simplicity, intuitive navigation, and dynamic task visualization. Emphasis is placed on ensuring that users can quickly add, modify, and organize tasks without facing a steep learning curve.

In addition to standard task creation and editing features, the system introduces advanced functionalities such as task prioritization, deadline reminders, and collaboration tools for team-based projects. A unique aspect discussed is the integration of progress-tracking metrics, which offer users insights into their productivity trends. The authors argue that these features collectively enhance task management efficiency and improve user satisfaction.

The paper further details the implementation process, describing the chosen development framework, database design, and API integration strategy. The system's architecture follows a modular approach, enabling developers to add new features easily. The authors provide code snippets and flowcharts to illustrate the system's functionality, ensuring readers gain a clear understanding of how the components interact to deliver a cohesive experience.

User evaluation played a vital role in validating the effectiveness of the proposed system. The authors conducted user trials where participants were asked to use the system for real-world task management. Feedback was collected on usability, feature effectiveness, and overall satisfaction. Results indicated that participants found the system intuitive and significantly improved their ability to track and manage tasks efficiently. The authors highlight key insights from the user study, such as the positive impact of reminder notifications and visual progress tracking.

## 5. ANALYSIS TESTING BLACK BOX AND WHITE BOX ON TO-DO LIST

by

Dede Irman Pirdaus, Rizki Apriva Hidayana

The paper examines the quality assurance of a web-based To-Do List application using black box and white box testing methods. As web applications become increasingly crucial for task management and productivity, ensuring their functionality and reliability is essential. This study focuses on testing the login and main pages of the application through structured test scenarios, aiming to identify system weaknesses and improve software quality.

The research begins by explaining the significance of To-Do List applications in organizing daily tasks and enhancing efficiency. These applications generally feature user-friendly interfaces with task categorization, reminders, and integrations with other tools. However, for any software to be effective, rigorous testing must be conducted. The study highlights that while software testing helps in identifying bugs, security vulnerabilities, and functional inconsistencies, it also comes with challenges, such as increased costs and time consumption.

The methodology follows a structured approach, starting with designing test scenarios and creating test cases based on expected user interactions. The process includes executing test cases and analyzing results to validate system performance. The paper employs two primary testing techniques:

1. Black Box Testing: This method evaluates the system from an external user perspective without examining the internal code structure. It focuses on input-output relationships, ensuring that the application responds correctly to user actions. The testing scenarios involve logging in with correct and incorrect credentials, validating error messages, adding and managing tasks, editing, deleting, and logging out. The results confirm that the application correctly processes user inputs and displays appropriate notifications when errors occur.

2.White Box Testing: This method involves an in-depth examination of the internal program structure, including the source code and logic flow. The study analyzes the application's control flow using flowcharts and flowgraphs to determine the number of independent logic paths. By calculating cyclomatic complexity, the researchers identify five key execution paths that need to be tested to ensure comprehensive coverage. The analysis of the login mechanism reveals code vulnerabilities and logical inefficiencies, which can be improved for better performance.

The findings suggest that both black box and white box testing methods effectively identify application flaws. While black box testing ensures the system meets user expectations, white box testing provides deeper insights into code functionality and logical consistency. The paper concludes that combining both methods enhances overall software reliability and security. Future improvements include refining error-handling mechanisms, optimizing system performance, and expanding test coverage to include additional features.

## 6. WHAT A TO-DO: STUDIES OF TASK MANAGEMENT TOWARDS THE DESIGN OF A PERSONAL TASK LIST MANAGER

by

Daniel G bobrow

The study concludes that effective task management is not hindered by poor prioritization but rather by the significant effort required to ensure that important tasks are completed despite distractions, interruptions, and competing demands. Contrary to the common assumption that individuals struggle with organizing their work, the research highlights that people develop highly personalized and sophisticated strategies to track, organize, and execute their tasks.

These strategies often involve a combination of tools, including traditional to-do lists, email inboxes, physical reminders, and digital calendars. The study underscores the adaptability of individuals in managing their workloads, demonstrating that task management is not a one-size-fits-all process but instead a dynamic practice influenced by contextual factors, social obligations, and personal work styles.

The findings emphasize that an effective task management system should not rigidly impose structure but should instead support and enhance users' existing behaviors. To this end, the researchers propose several key design principles for an optimal task management tool, including multiple task views, integration with scheduling tools, historical tracking of completed tasks, and prioritization based on social relationships. Recognizing the need for a system that aligns with real-world workflows, the authors developed TaskVista, a prototype task manager that incorporates these insights. TaskVista is designed as a lightweight, intuitive tool that sits on the user's desktop, allowing seamless task entry, prioritization, and collaboration. By enabling smart sorting of tasks based on urgency, social relationships, and deadlines, it reduces cognitive overload and helps users manage their workload more efficiently.

Ultimately, the study argues that the real challenge in task management lies in ensuring that critical tasks are completed despite changing circumstances and unexpected demands. Task management tools should not merely function as digital lists but should serve as adaptive, intelligent assistants that provide timely reminders, facilitate collaboration, and help users make better-informed decisions about their work. By learning from real-world task management practices, systems like TaskVista can help bridge the gap between how people naturally organize their tasks and the structured approach imposed by conventional task management applications. The research suggests that the future of task management lies in creating flexible, user-centered tools that integrate seamlessly into diverse work environments, making productivity a more effortless and intuitive process.

## 7. A REVIEW OF DAILY PRODUCTIVITY GROWTH USING TODO MANAGER

by Suyash Kejriwal ,Vaibhav Vishal,Aastha Gulati,Gaurav Gambhir

The paper presents a comprehensive overview of the To-Do Manager application designed to improve daily productivity through enhanced task management strategies. Similar to the FastDTW critique, the authors emphasize addressing inefficiencies in traditional task managers by introducing features that provide more precise and adaptable solutions.

The To-Do Manager introduces a novel approach that tracks task completion in phases, acknowledging that most tasks are not completed in one go. Unlike traditional to-do list applications that treat tasks as binary (either complete or incomplete), this system allows users to mark incremental progress. By tracking these partial completions, the application offers a more accurate representation of a user's progress towards their goals. This concept aligns with the critique of FastDTW's reliance on approximation, emphasizing instead the benefits of tracking partial progress for improved accuracy.

One of the standout features discussed in the paper is the Target Completion Indicator. This feature visually tracks progress using color-coded calendar markers, where green indicates successful achievement of the user's predefined goals and red denotes failure to meet targets. This visual representation mirrors the argument made in the FastDTW critique, where classic DTW's precise results are often more effective than approximate methods. By offering users a clear and immediate visual cue regarding their performance, the To-Do Manager encourages users to remain consistent in their productivity habits.

Furthermore, the paper underlines the significance of tracking task efficiency, which is akin to ensuring data accuracy in time series analysis. The system generates detailed statistical reports displaying completed tasks, pending tasks, and overall performance metrics. These reports allow users to analyze their daily, weekly, and monthly progress, promoting self-awareness and improved task management. This emphasis on precise performance tracking echoes the FastDTW critique's conclusion that accuracy and detailed tracking are critical to improving results in complex systems.

## 8. SCHEDULEME - SMART DIGITAL PERSONAL ASSISTANT FOR AUTOMATIC PRIORITY BASED TASK SCHEDULING AND TIME MANAGEMENT

By Liyanage A.N Madhushanka A.G.A.D, Uduwara U.M.L, W.M.S.D Jayarathne, Samanthi Siriwardana, Shyam Reyal, Mithsara W.K.M.

The paper presents a comprehensive overview of the To-Do Manager application designed to improve daily productivity through enhanced task management strategies. Similar to the FastDTW critique, the authors emphasize addressing inefficiencies in traditional task managers by introducing features that provide more precise and adaptable solutions.

The To-Do Manager introduces a novel approach that tracks task completion in phases, acknowledging that most tasks are not completed in one go. Unlike traditional to-do list applications that treat tasks as binary (either complete or incomplete), this system allows users to mark incremental progress. By tracking these partial completions, the application offers a more

accurate representation of a user's progress towards their goals. This concept aligns with the critique of FastDTW's reliance on approximation, emphasizing instead the benefits of tracking partial progress for improved accuracy.

One of the standout features discussed in the paper is the Target Completion Indicator. This feature visually tracks progress using color-coded calendar markers, where green indicates successful achievement of the user's predefined goals and red denotes failure to meet targets. This visual representation mirrors the argument made in the FastDTW critique, where classic DTW's precise results are often more effective than approximate methods. By offering users a clear and immediate visual cue regarding their performance, the To-Do Manager encourages users to remain consistent in their productivity habits.

Furthermore, the paper underlines the significance of tracking task efficiency, which is akin to ensuring data accuracy in time series analysis. The system generates detailed statistical reports displaying completed tasks, pending tasks, and overall performance metrics. These reports allow users to analyze their daily, weekly, and monthly progress, promoting self-awareness and improved task management. This emphasis on precise performance tracking echoes the FastDTW critique's conclusion that accuracy and detailed tracking are critical to improving results in complex systems.

## 9. GAMIFICATION-BASED TO-DO LIST MOBILE APPLICATION DEVELOPMENT

by Arvendoa, Cakra , Emny Harna Yossy

The paper presents the development of a mobile to-do list application that leveragesgamification elements to enhance user motivation and productivity. The primary goal of this application is to address common challenges users face in staying motivated to complete everyday tasks. By applying engaging game mechanics, the application encourages consistent user interaction and improved task management.

The application incorporates several core gamification features designed to foster user engagement. The point system rewards users with points for completing tasks, providing a tangible measure of progress. Each completed task earns the user a designated number of points based on its complexity and urgency. For example, simpler tasks such as "send an email" may earn fewer points, while complex or time-consuming tasks like "finish a project report" may award higher points. This structured points system encourages users to prioritize important tasks while still rewarding their progress on smaller, routine activities. Accumulated points can also be spent on in-app rewards, unlocking additional features or customization options, further motivating users to stay active.

The level system builds on the point system by introducing progression milestones. As users accumulate points, they advance to higher levels, each representing a significant achievement. Each new level may unlock unique rewards, such as profile upgrades, enhanced task management features, or themed visual effects within the app. This structured progression fosters a sense of achievement, encouraging users to continue completing tasks consistently. Moreover, by introducing challenging milestones, the application maintains user engagement by giving them new goals to strive for. This system mirrors traditional gaming mechanics, where progress tracking and incremental achievements keep players motivated to advance.

Additionally, the badge system offers virtual rewards for achieving specific milestones, adding an element of achievement recognition that motivates users to reach their goals. Badges are awarded for consistent task completion, meeting deadlines, or achieving streaks of productive days, reinforcing positive behaviors and establishing a sense of accomplishment.

To ensure a structured and organized development process, the paper outlines the use of the waterfall method. This method follows a sequential development process involving requirement analysis, design, implementation, testing, and maintenance. By completing each stage before progressing to the next, developers ensured a well-defined workflow, minimizing potential errors and improving overall project quality. The waterfall approach was particularly suited for this project as it provided clear milestones, helping the developers methodically implement each gamification feature.

The effectiveness of the application was evaluated through user testing, where participants provided feedback on its impact on their productivity and motivation. Test results indicated that users felt more encouraged to complete tasks due to the reward-driven nature of the system. The point, level, and badge systems collectively improved user engagement, with many participants highlighting these features as effective motivators. Positive feedback emphasized the application's ability to create a sense of accomplishment, making daily task management more enjoyable.

Overall, the paper concludes that integrating gamification into a mobile to-do list application effectively enhances user motivation and task completion. By combining structured development practices with engaging game mechanics, the application successfully addresses the challenges users face in maintaining consistent productivity.

## 10. HEYMITRA- A DAILY LIFE ASSISTANT WEB APPLICATION WITH TODO LIST AND TIME TRACKING ABILITY

by Er. Harsh Sharma Samarth Maheshwari, Sanjay Singla, Sagar Kumar, Shivam Chopra, Garvit Khandelwal

The research paper introduces HeyMitra, a web-based daily life assistant designed to improve student productivity by addressing the challenges of time management and study planning. With students facing increasing academic demands and distractions, HeyMitra aims to provide an effective solution for better organization and improved focus.

At the core of HeyMitra lies a sophisticated dynamic algorithm designed to intelligently generate personalized to-do lists. This algorithm considers multiple factors such as assignment deadlines, exam dates, task priority, and individual course schedules. By dynamically adjusting task priorities based on changing academic needs, HeyMitra provides students with a structured yet flexible overview of their responsibilities. This feature helps mitigate feelings of overwhelm and ensures students can focus on high-priority tasks while effectively managing their workload.

In addition to task management, HeyMitra integrates a powerful time tracking feature that enables students to monitor the duration of their study sessions and other activities. This functionality allows students to analyze how they allocate their time, identify inefficiencies, and make strategic adjustments to their schedules. By observing patterns in their study habits, students can pinpoint optimal periods for focused work and track their overall productivity. This

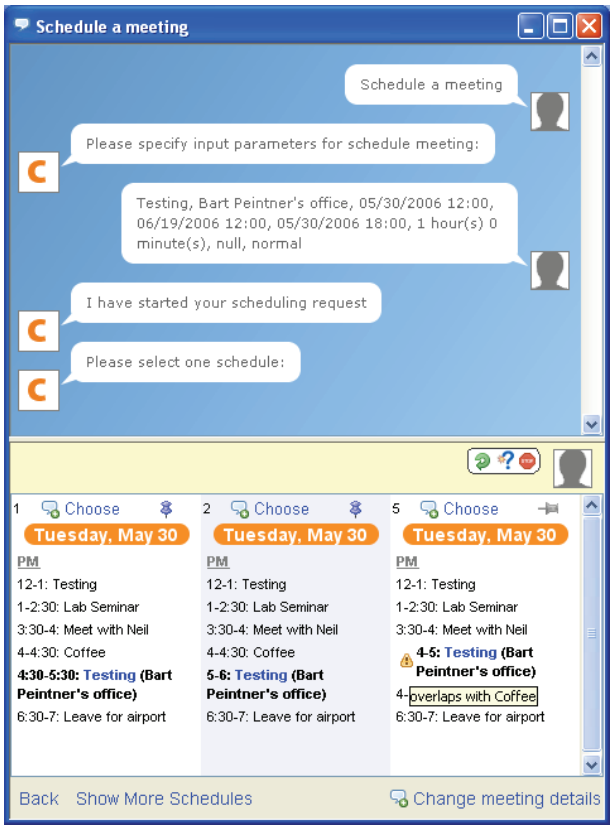



Figure 2.5: Chat with assistant on task management

data-driven insight empowers users to make informed decisions about their time management strategies.

Another essential feature of HeyMitra is its performance reporting system, which compiles detailed analytics on task completion rates, total study hours, and consistency in academic efforts. These reports provide students with objective insights into their strengths and weaknesses in time management. By identifying areas for improvement, students can refine their strategies and adopt more efficient study practices. The combination of data collection, analysis, and feedback establishes a continuous improvement loop that helps students develop productive habits over time.

The decision to develop HeyMitra as a web application offers distinct advantages in accessibility and integration. Since it is web-based, students can access the platform across multiple devices such as laptops, tablets, and smartphones. This ensures that users can engage with the platform seamlessly, whether they are at home, in the library, or on the go. The dynamic algorithm further ensures that HeyMitra adapts to real-time changes in schedules, priorities, and deadlines, providing students with an up-to-date and personalized task management experience.

By integrating dynamic to-do list generation, comprehensive time tracking, and insightful performance reporting, HeyMitra provides a proactive and intelligent digital assistant that actively supports students in managing their academic and personal responsibilities.

## 2.3 LITERATURE SURVEY OVERVIEW

| Sl No: | Title of the Paper | Authors | Technologies Used | Publication | Year | Observations |
|---|---|---|---|---|---|---|
| 1 | Towel Towards an Intelligent To Do List | Kenneth, James | Task list management systems, Task prioritization strategies | SRI International | 2004 | Task management practices and with software assistant. |
| 2 | What a To Do Studies of Task Management Towards the Design of a Personal Task List Manager | Victoria ,Brinda ,Nathaniel ,Peter ,Daniel G,Nicolas | Web based digital task management systems | CHI | 2004 | Using Effective task management prioritization strategies |
| 3 | TaskDo A Daily Task Recommender System | Dr.Mohammad ,Nikhil Sai ,Gurram | Content based recommender system | IEEE | 2019 | Optimal task scheduling for productivity |
| 4 | A Review Of Daily Productivity Growth Using Todo Manager | Suyash ,Vaibhav ,Aastha ,Gaurav | Online To do list applications, calendar based visualization | IRJMETS | 2020 | Features like progress tracking, prioritization, a calendar to display success visually |
| 5 | Development of To Do List and Monetary Management System | Diana ,Chen Wong , BongSiaw Wee | Google Sheets,Google Forms,AppSheet, Google Drive | IABEC | 2021 | Combines task and monetary management with group collaboration, customizable features |
| 6 | Gamification Based To Do List Mobile Application Development | Arvendoa Cakra,Emny Harna Yossy | Gamification frameworks,Task management systems,databases | IEEE (SQLite) | 2023 | Gamified to do list web app, Progress visualization |

| | | | | | | |
|---|---|---|---|---|---|---|
| 7 | ScheduleME Smart Digital Personal Assistant for Automatic Priority Based Task Scheduling and Time Management | Liyanag, Uduwara, Jayarathne, Samanth, Shyam, Mithsara | Personalized scheduling techniques, user centered interaction design | IEEE | 2021 | Efficient task management and a user friendly interface. |
| 8 | HeyMitra A Daily Life Assistant web application with Todo List and Time Tracking ability | Er.Harsh, Sharma, Samarth, Sanjay , Sagar , Shivam Chopra, Garvit Khandelwal | Data visualization tools | IEEE | 2024 | Academic planning with personalized to do lists |
| 9 | Bridging the Gap Between Time Management Research and Task Management App Design A Study on the Integration of Planning Fallacy Mitigation Strategies | Yoana, Duncan P. Brumby, Anna L.Cox | Time tracking tools, Analytics Dashboards | CHI | 2024 | Implement research based strategies effectively |
| 10 | Analysis Testing Black Box and White Box on Application To Do List Based Web App | Irman Pirdaus, Rizki Hidayana | Black box testing, white box testing, web based to do list application testing | IJMSC | 2024 | Black box and white box testing identified errors and weaknesses in a web based to do list application, improving its functionality and quality |

# Chapter 3

# SYSTEM DESIGN

The design of Tick Task – To-Do List is carefully structured to provide a seamless and efficient task management experience. The system is built to simplify task creation, assignment, tracking, and completion while ensuring that users can collaborate effectively in a well-organized and intuitive environment.

To achieve this, the system follows a systematic workflow that automates various aspects of task management, such as setting priorities, sending reminders, and monitoring task progress. The architecture is designed to handle individual and group tasks efficiently, ensuring that users can focus on their priorities without unnecessary complexity. The design also includes real-time notifications, role-based access, and an interactive dashboard to help users stay updated on pending and completed tasks.

Additionally, the use case diagram plays a crucial role in defining how different users interact with the system. Various roles, such as Admins, Task Creators, Task Assignees, and Regular Users, have different levels of access and permissions, ensuring that task-related activities are well-structured and organized. The system ensures that task assignments are clear, progress is monitored effectively, and completed tasks are reviewed for quality.

### 3.0.1 WORKFLOW DIAGRAM

The workflow diagram represents the step-by-step process of how a task moves through the system, from creation to completion.

### 3.0.2 WORKFLOW STEPS

1. **User Login & Authentication**
    - Users sign in with their credentials to access the system.
    - New users can register and create an account.
2. **Task Creation**
    - Users can create a new task by adding a title, description, due date, and priority level.
    - Tasks can be assigned to individuals or teams.

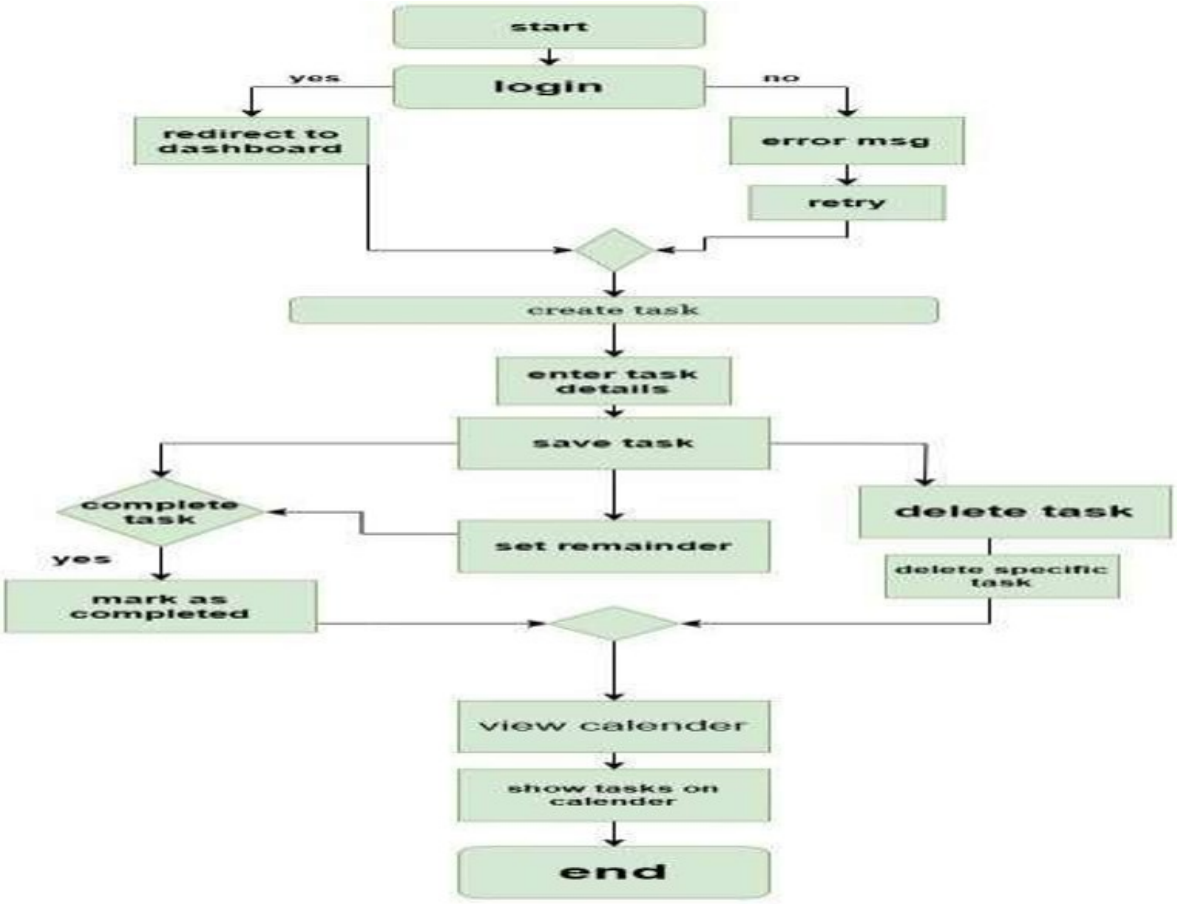



Figure 3.1: Workflow Diagram

3. **Task Assignment & Notifications**
    - Assigned users receive notifications about new tasks.
    - Tasks appear in their personal dashboard with status updates.
4. **Task Execution & Progress Updates**
    - Users update task status (e.g., ”In Progress”, ”On Hold”, ”Completed”).
    - Comments and attachments can be added for better collaboration.
5. **Reminders & Deadlines**
    - Automated reminders notify users of upcoming deadlines.
    - Urgent tasks are highlighted in the dashboard.
6. **Task Completion & Review**
    - Users mark the task as ”Completed” when finished.
    - Team members or supervisors review the task and provide feedback.
7. **Analytics & Reports**
    - Users can generate reports to track productivity.

- Statistics like completed vs. pending tasks and efficiency scores are displayed.

This structured workflow ensures tasks are properly assigned, monitored, and completed, leading to improved organization and productivity.

## 3.1 USE CASE DIAGRAM

A use case diagram visually represents how users interact with the Tick Task system. It provides a clear understanding of the system's functionalities by identifying the actors (users) and their associated use cases (actions they perform). This diagram helps define the system's scope and ensures that all essential user interactions are covered.

### 3.1.1 MAIN ACTORS AND THEIR ROLES

- **Admin**
  - Manages user accounts, roles, and permissions.
  - Configures system settings, ensuring smooth functionality.
  - Monitors overall task progress and performance metrics.
  - Generates system-wide reports to assess user engagement and efficiency.
- **Task Creator**
  - Responsible for creating new tasks in the system.
  - Assigns tasks to individuals or groups based on workload and priority.
  - Defines task details, including descriptions, due dates, priorities, and dependencies.
  - Sets reminders and notifications to ensure deadlines are met.
  - Modifies task parameters when necessary, including reassignment or deadline extensions.
- **Task Assignee**
  - Receives assigned tasks and updates their progress (e.g., "Not Started", "In Progress", "Completed").
  - Adds relevant comments, attachments, or links for better collaboration.
  - Communicates with the Task Creator or Admin for clarifications or modifications.
  - Marks tasks as complete once finished, triggering a review process.
- **Regular User / Viewer**
  - Has access to view tasks, schedules, and progress reports.
  - Monitors the status of assigned or group tasks.
  - Can generate basic reports on task completion and pending workloads.
  - Uses filtering and sorting options to prioritize tasks based on deadlines and urgency.

### 3.1.2 PRIMARY USE CASES

- **Login & User Authentication** – Users securely log in to the system using their credentials.
- **Create Task** – A Task Creator or Admin creates a new task, specifying its details.
- **Update Task** – Task Assignees modify the task status as they work on it and add notes or attachments.
- **Set Reminders & Notifications** – Users schedule automated reminders and receive alerts for deadlines.
- **View Task List & Dashboard** – Users access their task dashboard to check pending, in-progress, and completed tasks.
- **Complete Task & Review** – Task Assignees mark tasks as completed, triggering a review by the Task Creator or Admin.
- **Generate Reports & Analytics** – Users generate productivity reports based on task completion rates and efficiency metrics.

This use case diagram defines how different roles interact with the system and outlines the functionalities available to each user.

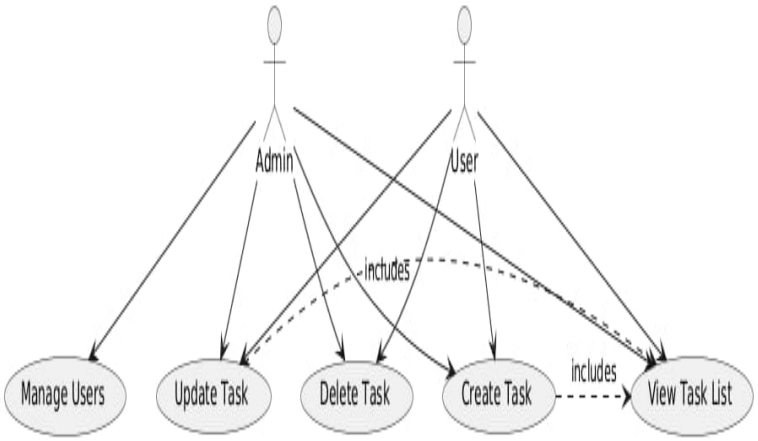



Figure 3.2: Use Case Diagram

# Chapter 4

# SYSTEM ARCHITECTURE

The system architecture of Tick Task is designed to be **scalable, efficient, and user-friendly**, ensuring seamless task management for individuals and teams. The system follows a **modular design**, consisting of multiple components that interact to provide a smooth and reliable experience. These components include the **frontend user interface, backend application logic, database, notification system, and third-party integrations**. Each component plays a crucial role in enabling **real-time task creation, assignment, tracking, collaboration, automated reminders, and analytics**.

### 4.0.1 ARCHITECTURAL OVERVIEW

The **Tick Task system** is built with the following key components:

- **User Interface (Frontend)** – Provides an intuitive and interactive platform for users.
- **Application Logic (Backend)** – Handles business logic, processes tasks, and ensures security.
- **Database (Data Storage)** – Stores all essential data, including tasks, users, and notifications.
- **Notification System** – Sends alerts and reminders for upcoming tasks and deadlines.
- **Third-Party Integrations** – Connects with external services like Google Calendar, Slack, and cloud storage for enhanced functionality.

Each of these components is **interconnected**, ensuring **real-time synchronization, security, and efficiency**.

### 4.0.2 USER INTERFACE(FRONTEND)

- Designed to be **user-friendly, responsive, and accessible** across multiple devices.
- Provides a **dashboard** displaying an overview of pending, ongoing, and completed tasks.
- Includes a **task management panel** where users can create, assign, and update tasks.
- Supports **reminders and notifications**, ensuring users stay informed about deadlines.
- Enables **collaborative tools** such as comments, file attachments, and shared task lists.
- Communicates with the backend via **secure API requests** for real-time updates.

### 4.0.3 APPLICATION LOGIC (BACKEND)

- Acts as the **core processing unit** of the system.
- Manages **user authentication and role-based access control** (Admin, Task Creator, Task Assignee, Viewer).
- Handles **task creation, assignment, updates, and completion tracking**.
- Implements **task dependencies** to ensure proper scheduling and avoid conflicts.
- Generates **real-time notifications and reminders** based on task deadlines.
- Provides **analytical insights and reporting features** to track user productivity.
- Built using **scalable technologies** to handle multiple users and large task volumes efficiently.

### 4.0.4 DATABASE (DATA STORAGE)

- Stores all critical data, including **user profiles, task details, task statuses, priorities, notifications, and analytics reports**.
- Uses **data encryption** to ensure security and prevent unauthorized access.
- Supports **real-time synchronization**, meaning any task update is instantly reflected across all devices.
- Implements **optimized queries** for fast data retrieval, even with high usage.
- Ensures **data consistency and reliability** for seamless system operation.

### 4.0.5 NOTIFICATION SYSTEM

- Designed to **keep users informed** with real-time alerts and reminders.
- Sends notifications through multiple channels, including:
  - **Email alerts** for task assignments and deadline reminders.
  - **Push notifications** for real-time task updates and urgent alerts.
  - **Dashboard alerts** for in-app reminders and overdue tasks.
- Dynamically triggers **alerts based on predefined rules**, such as upcoming deadlines and overdue tasks.
- Helps improve **user engagement and accountability** by ensuring tasks are completed on time.

### 4.0.6 DATA FLOW IN TICK TASK

The **data flow** within Tick Task follows a structured process to ensure **efficiency and seamless operations**:

1. **User logs in** – The system **authenticates credentials** and loads the user dashboard.
2. **Task is created** – The system **processes and stores** the task in the database, notifying assigned users.
3. **Task updates occur** – Users modify task status, and **changes are synchronized in real time**.
4. **Notifications are triggered** – The system sends reminders **based on due dates and priorities**.
5. **Tasks are marked as completed** – The system updates task status and **generates reports**.
6. **Reports and analytics are generated** – Users can view performance insights and **track efficiency**.

# Chapter 5

# REQUIREMENT ANALYSIS

### 5.0.1 Functional Requirements

The functional requirements define the specific behaviors and features of the Tick Task application. These include:

- **User Authentication and Role Management**
  - Users must be able to register, log in, and log out securely.
  - The system should support different user roles, such as admin, task creator, and task assignee.
- **Task Management**
  - Users should be able to create, update, delete, and assign tasks.
  - Tasks should have attributes such as title, description, priority level, due date, and assigned user(s).
- **Task Tracking and Notifications**
  - Users should receive reminders and notifications for upcoming and overdue tasks.
  - The system should update task statuses (e.g., pending, in progress, completed) dynamically.
- **Collaboration and Communication**
  - Users should be able to comment on tasks and attach relevant documents.
  - Teams should have a shared workspace to view and manage group tasks.

  charts and graphs, for performance tracking.

### 5.0.2 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements define the overall system attributes such as performance, security, and usability. These include:

- **Performance Requirements**

  - The system should handle multiple concurrent users without performance degradation.
  - Tasks and notifications should be updated in real time with minimal delay.

- **Security Requirements**
  - User data and tasks should be securely stored with encryption.
  - Role-based access control should be implemented to restrict unauthorized actions.

- **Usability Requirements**
  - The system should have an intuitive and user-friendly interface.
  - It should be accessible on both desktop and mobile devices.

- **Scalability Requirements** -+
  - The system should be designed to accommodate future enhancements, such as AI-based task recommendations.
  - Cloud-based infrastructure should be used to support an increasing number of users.

# Chapter 6

# IMPLEMENTATION

## 6.1 LOGIN PAGE

Below is the HTML code for the TickTask login page:

Listing 6.1: TickTask Login Page Code

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width,
initial-scale=1.0">
<title>TickTask - Home</title>
<link rel="stylesheet" href="logstyle.css">
<link rel="preconnect" href="https://
fonts.googleapis.com">
<link rel="preconnect"
href="https://fonts.gstatic.com" crossorigin>
<link href=
"https://fonts.googleapis.com/css2?family=Dr+Sugiyama&family=

Mogra&display=swap" rel="stylesheet">
<script defer src="https://cdnjs.cloudflare.com
/ajax/libs/gsap/3.12.2/gsap.min.js"></script>
<script defer src="script.js"></script>
</head>
<body>
<div class="navbar">
<div class="logo">
<img src="https://github.com
/AaravR-Jith/ticktacklogin
/blob/main/ticktask-high-resolution-logo.png?raw=true"
alt="TickTaskLogo">
</div>
<div class="nav-links">
<a href="#">Get Started</a>
```

```
<a href="#">Made For</a>
<a href="#">About</a>
</div>
</div>

<div class="container">
<div class="left">
<span>Stay on Track with</span>
<span>TickTask!</span>
</div>
<div class="right">
<div class="cta">
<a href="login2.html" class="get-started">Get Started</a>
</div>
</div>
</div>

<div class="section">
<div class="image">
<img src="https://github.com/AaravR-Jith/ticktacklogin/blob/main/
1st.jpeg?raw=true" alt="Task■Dashboard">
</div>
<div class="desc">
<span>Be organized, be productive!</span>
<span>TickTask helps you manage tasks effortlessly,</span>
<span>from daily to-dos to big projects</span>
<span>all in one place.</span>
</div>
</div>

<div class="section">
<div class="image">
<img src="https://github.com/AaravR-Jith/ticktacklogin/
raw/refs/heads/main/team.webp" alt="Team■Collaboration">
</div>
<div class="desc">
<span>Work together</span>
<span>seamlessly with real-time</span>
<span>team collaboration tools.</span>
</div>
</div>

<div class="section">
<div class="image">
<img src="https://github.com/AaravR-Jith/ticktacklogin/
raw/refs/heads/main/customize.webp" alt="Customization">
</div>
<div class="desc">
```

```
<span>Personalize your workspace</span>
<span>with themes, dark mode,</span>
<span>and layouts.</span>
</div>
</div>

<div class="footer">
<a href="login.html" class="get-started">Get Started</a>
</div>
</body>
</html>
```

Listing 6.2: TickTask Login Page Code

```
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

body {
    background-color: #f8f9fa;
    color: #333;
    font-family: Arial, sans-serif;
}

.navbar {
    display: flex;
    align-items: center;
    justify-content: space-between;
    background: #fff;
    padding: 15px 50px;

}

.logo img {
    height: 100px;
    max-width: 100%;
}

.nav-links a {
    margin-left: 20px;
    text-decoration: none;
    font-weight: bold;
    color: #007bff;
    font-size: 18px;
}

.nav-links a:hover {
```

```
    color: #0056b3;
}

.container {
    display: flex;
    height: 80vh;
    padding: 50px;
    align-items: center;
    justify-content: space-between;
}

.left {
    flex: 1;
    display: flex;
    flex-direction: column;
    justify-content: center;
    text-align: center;
    font-size: 4rem;
    font-weight: bold;
    color: rgb(28, 4, 207);
    font-family: "Dr■Sugiyama", serif;;
}

.right {
    flex: 1;
    display: flex;
    justify-content: center;
}

.login-form {
    background: #fff;
    padding: 30px;
    box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
    border-radius: 10px;
    width: 100%;
    max-width: 350px;
}

.login-form h2 {
    text-align: center;
    margin-bottom: 20px;
}

.login-form input {
    width: 100%;
    padding: 10px;
    margin: 10px 0;
    border: 1px solid #ccc;
```

```
    border-radius: 5px;
}

.login-form button {
    width: 100%;
    padding: 10px;
    background: #007bff;
    color: #fff;
    font-weight: bold;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    font-size: 16px;

    box-shadow: 0 6px 15px rgba(0, 0, 0, 0.3);
}

.login-form button:hover {
    background: #00b327;
    transform: translateY(-3px);
}


.section {
    display: flex;
    align-items: center;
    padding: 50px;
    flex-wrap: wrap;
    box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
    border-radius: 20px;
    margin: 20px;
}

.section:nth-child(even) {
    flex-direction: row-reverse;
}

.desc {
    display: flex;
   flex-direction: column;
    color: #0f4994;
    font-size: 1.5rem;
    font-weight: bold;
    flex: 1;
    padding: 20px;
    font-family:  "Mogra", serif;
}
```

```
.image {
    flex: 1;
    padding: 20px;
    display: flex;
    align-items: center;
    justify-content: center;
}

.image img {
    width: 100%;
    height: auto;
   max-height: 350px;
    border-radius: 10px;
    object-fit: cover;
}


@media (max-width: 768px) {
    .container {
        flex-direction: column;
        text-align: center;
        height: auto;
    }

    .left {
        font-size: 2rem;
    }

    .section {
        flex-direction: column;
        text-align: center;
    }
}
.footer {
    display: flex;
    justify-content: center;
    align-items: center;
    padding: 50px;
    background: #fff;
}

.get-started {
    padding: 15px 40px;
    font-size: 20px;
    font-weight: bold;
    color: white;
    background: linear-gradient(90deg, #007bff, #0056b3);
    border: none;
```

```
    border-radius: 30px;
    cursor: pointer;
    box-shadow: 0 4px 10px rgba(0, 0, 0, 0.2);
}

.get-started:hover {
    transform: translateY(-3px);
    background: #0f0568;
}
\begin{lstlisting}[language=HTML, caption=TickTask Login Page Code]
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width,
initial-scale=1.0">
<title>Register & Login</title>
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/
libs/font-awesome/6.5.1/css/all.min.css">
<link rel="stylesheet" href="style.css">
<script type="module" src="firebaseauth.js"></script>
</head>

<body>
<div class="container" id="signup"
style="display:none;">
<h1 class="form-title">Register</h1>
<form method="post" action="">
<div id="signUpMessage"
class="messageDiv"
style="display:none;"></div>

<div class="input-group">
<i class="fas■fa-user"></i>
<input type="text" id="fName" placeholder="First.Name" required>
<label for="fName">First Name</label>
</div>

<div class="input-group">
<i class="fasfauser"></i>
<input type="text" id="lName" placeholder="Last.Name" required>
<label for="lName">Last Name</label>
</div>

<div class="input-group">
<i class="fas.fa-envelope"></i>
<input type="email" id="rEmail" placeholder="Email" required>
```

```
<label for="rEmail">Email</label>
</div>

<div class="input-group">
<i class="fas.fa-lock"></i>
<input type="password" id="rPassword" placeholder="Password" required>
<label for="rPassword">Password</label>
</div>

<button class="btn" id="submitSignUp">Sign Up</button>
</form>

<div class="links">
<p>Already have an account?</p>
<button id="signInButton" class="text-button">Sign In</button>
</div>
</div>

<div class="container" id="signIn">
<h1 class="form-title">Sign In</h1>
<form method="post" action="">
<div id="signInMessage" class="messageDiv" style="display:none;"></div>

<div class="input-group">
<i class="fas■fa-envelope"></i>
<input type="email" id="email" placeholder="Email" required>
<label for="email">Email</label>
</div>

<div class="input-group">
<i class="fas■fa-lock"></i>
<input type="password" id="password" placeholder="Password" required>
<label for="password">Password</label>
</div>

<p class="recover">
<a href="#">Recover Password</a>
</p>

<button class="btn" id="submitSignIn">Sign In</button>
</form>

<div class="links">
<p>Don't have an account yet?</p>
<button id="signUpButton" class="text-button">Sign Up</button>
</div>
</div>
```

```
<script src="script.js"></script>
</body>
</html>
```

Listing 6.3: authentication firebase

```
// Import the functions you need from the SDKs you need
import { initializeApp }
from "https://www.gstatic.com/firebasejs/11.5.0/firebase-app.js";
import {
getAuth,
createUserWithEmailAndPassword,
signInWithEmailAndPassword
} from
"https://www.gstatic.com
/firebasejs/11.5.0/firebase-auth.js";
import {
    getFirestore,
    setDoc,
    doc
} from
"https://www.gstatic.com/firebasejs/11.5.0/firebase-firestore.js";

const firebaseConfig = {
    apiKey: "AIzaSyDRbQKhLrE9bzHGYTewfrP-uG3_DR75Gfo",
    authDomain: "login-form-265f6.firebaseapp.com",
    projectId: "login-form-265f6",
    storageBucket: "login-form-265f6.appspot.com",
    messagingSenderId: "431116535291",
    appId: "1:431116535291:web:47ea2b4cd26dfc38c9775e"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
const auth = getAuth(app);
const db = getFirestore(app);

function showMessage(message, divId) {
    const messageDiv = document.getElementById(divId);
    messageDiv.style.display = "block";
    messageDiv.innerHTML = message;
    messageDiv.style.opacity = 1;
    setTimeout(function() {
        messageDiv.style.opacity = 0;
    }, 5000);
}

// Form toggle functionality
document.getElementById('signUpButton')
```

```
.addEventListener('click', function() {
    document.getElementById('signIn').style.display = 'none';
    document.getElementById('signup').style.display = 'block';
});

document.getElementById('signInButton').
addEventListener('click', function() {
    document.getElementById('signup').
    style.display = 'none';
    document.getElementById('signIn').
    style.display = 'block';
});

// Sign Up functionality
document.getElementById('submitSignUp').
addEventListener('click', (event) => {
    event.preventDefault();
    const email = document.getElementById('rEmail').value;
    const password = document.getElementById('rPassword').value;
    const firstName = document.getElementById('fName').value;
    const lastName = document.getElementById('lName').value;

    createUserWithEmailAndPassword(auth, email, password)
        .then((userCredential) => {
            const user = userCredential.user;
            const userData = {
                email: email,
                firstName: firstName,
                lastName: lastName
            };

showMessage('Account Created Successfully', 'signUpMessage');

            const docRef = doc(db, "users", user.uid);
            setDoc(docRef, userData)
                .then(() => {
                    window.location.href = 'index.html';
                })
                .catch((error) => {
      console.error("Error writing document", error);
    showMessage('Error saving user data', 'signUpMessage');
                });
        })
        .catch((error) => {
            const errorCode = error.code;
            if (errorCode == 'auth/email-already-in-use') {
 showMessage('Email address already exists!','signUpMessage');
       } else if (errorCode == 'auth/weakpassword'
```

```
        )showMessage
        ('Password should be at least 6 characters',
        'signUpMessage');
            } else {
 showMessage('Unable to create user:
 + error.message, 'signUpMessage');
            }
         });
});

// Sign In functionality
document.getElementById('submitSignIn')
.addEventListener('click', (event) => {
event.preventDefault();
const email = document.getElementById('email').value;
const password = document.getElementById('password').value;

 signInWithEmailAndPassword(auth, email, password)
        .then((userCredential) => {
            // Signed in
            window.location.href = 'index.html';
        })
        .catch((error) => {
         const errorCode = error.code;
    if (errorCode == 'auth/invalid-credential') {
 showMessage('Invalid email or password', 'signInMessage');
            } else {
     showMessage('Error signing in: '
     + error.message, 'signInMessage');
            }
        });
});
const signIn=document.getElementById('submitSignIn');
signIn.addEventListener('click',(event)=>{
    event.preventDefault();
    const email=document.getElementById('email').value;
    const password = document.getElementById('password').value;
    const auth=getAuth();

    signInWithEmailAndPassword(auth,email,password)
    .then((userCredential)=>{
        showMessage('login is successfull','signInMessage');
        const user=userCredential.user;
        localStorage.setItem('loggedInUserId',user.id);
        window.location.href='index.html'

    })
    .catch((error)=>{
```

```
        const errorCode=error.code;
        if(errorCode=='auth/invalid-credential '){
            showMessage('Incorrect email or password','signInMessage');
        }
        else{
            showMessage('Account does not exist','signInMessage');
        }
    })
})
\section{DASHBOARD}


\begin{lstlisting}[language=HTML, caption=dashboard]
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
<meta name="viewport" content="width=device-width, initial-scale=1.0"
/>
<title>Tasky Appln</title>

<!-- custom css -->
<link rel="stylesheet" href="styles.css" />

<!-- CSS only for bootstrap -->
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/
bootstrap.min.css"
rel="stylesheet"
integrity="sha384-rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1RWuH61DGLwZ
JEdK2Kadq2F9CUG65"
crossorigin="anonymous"
/>

<!-- font-awesome icons -->
<link
rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/
6.2.1/css/all.min.css"
integrity="sha512-MV7K8+y+gLIBoVD59lQIYicR65iaqukzvf/nwasF0nqhPay5w/
9lJmVM2hMDcnK1OnMGCdVK+iQrJ7lzPJQd1w=="
crossorigin="anonymous"
referrerpolicy="no-referrer"
/>
</head>

<body onload="loadInitialData()">
```

```
<!-- Btn which should be displayed on smaller screen -->
<div class="add__new__btn__mobile__only d-md-none">
<button
class="btn btn-primary d-flex align-items-center gap-2"
type="submit"
data-bs-toggle="modal"
data-bs-target="#addNewModal"
      >
<i class="fas fa-plus"></i>Add New
</button>
</div>

<!-- too add new modal on click of add an item btn -->
<div
class="modal fade"
id="addNewModal"
tabindex="-1"
aria-labelledby="addNewModalLabel"
aria-hidden="true"
>
<div class="modal-dialog modal-dialog-centered">
<div class="modal-content">
<div class="modal-header">
<h5 class="modal-title" id="addNewModalLabel">Add New Task</h5>
<button
type="button"
class="btn-close"
data-bs-dismiss="modal"
aria-label="Close"
></button>
</div>
<div class="modal-body">
<form onsubmit="return false">
<div class="mb-3">
<label for="imageUrl" class="form-label">Image URL</label>
<input
type="url"
class="form-control"
id="imageUrl"
placeholder="http://emample.com/image.png"
/>
</div>
<div class="mb-3">
<label for="taskTitle" class="form-label">Task Title</label>
<input
 type="text"
class="form-control"
id="taskTitle"
```

```
placeholder="Learn Web Dev"
required
/>
</div>
<div class="mb-3">
<label for="tags" class="form-label">Task Type</label>
<input
type="text"
class="form-control"
id="tags"
placeholder="Work..."
required
/>
</div>
<div class="mb-3">
<label for="taskDescription" class="form-label"
>Task Description</label
>
<textarea
class="form-control"
id="taskDescription"
placeholder="Explainsomethingabout.it..."
rows="4"
required
></textarea>
</div>
button
type="button"
class="btn.btn-secondary"
data-bs-dismiss="modal"
>
Close
</button>
<button
type="submit"
class="btn btn-primary"
onclick="handleSubmit()"
>
Save changes
</button>
</form>
</div>
</div>
</div>
</div>

!-- Modal for UI Display On Home pg -->
<div
```

```
class="modalfade"
id="showTask"
tabindex="-1"
aria-labelledby="showTaskLabel"
aria-hidden="true"
>
<div class="modal-dialog modal-lg">
<div class="modal-content">
<div class="modal-header">
<h1 class="modal-title" id="showTaskLabel">Task Details</h1>
<button
type="button"
class="btn-close"
data-bs-dismiss="modal"
aria-label="Close"
></button>
</div>
<div class="modal-body task__modal__body">
<!-- <p>js txt</p> -->
</div>
<div class="modal-footer">
<button
type="button"
class="btn.btn-secondary"
data-bs-dismiss="modal"
>
Close
</button>
</div>
</div>
</div>
</div>

<!-- Nav bar -->
<nav class="navbar navbar-expand-md bg-light">
<div class="container-fluid">
<a class="navbar-brand" href="#">Tasky</a>
<button
class="navbar-toggler"
type="button"
data-bs-toggle="collapse"
data-bs-target="#navbarSupportedContent"
aria-controls="navbarSupportedContent"
aria-expanded="false"
aria-label="Toggle navigation"
>
<span class="navbar-toggler-icon"></span>
</button>
```

```
<div class="collapse navbar-collapse" id="navbarSupportedContent">
<ul class="navbar-nav me-auto mb-2 mb-lg-0">
<li class="nav-item">
<a class="nav-link active" aria-current="page" href="#">Home</a>
</li>
</ul>

<!-- <form class="d-flex" role="search"> -->

<button
class="btn.btn-primary.d-flex.align-items-center.gap-2"
type="submit"
data-bs-toggle="modal"
data-bs-target="#addNewModal"
>
<i class="fas fa-plus"></i>
Add an item
</button>
<!-- </form> -->
</div>
</div>
</nav>

<!-- entire body - search i/p with display cards -->
<div class="container">
<!-- Content here -->
<section class="mt-4">
<div class="rowjustify-content-center">
<div class="col-md-6">
<div class="input-groupflex-nowrapshadow-lg rounded">
<input
type="search"
class="form-control"
placeholder="Search_Task"
aria-label="Search_Task"
aria-describedby="addon-wrapping"
oninput="searchTask.apply(this, arguments)"
/>
<span class="input-group-text" id="addon-wrapper">
<i class="fas fa-search"></i>
</span>
</div>
</div>
</div>
</section>

<section class="mt-4">
<div class="rowtask__contents">
```

```
<!-- <h1>This is the place where we will get all the cards</h1> -->
</div>
</section>
</div>

<!-- scripts -->
<script type="text/javascript" src="index.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/
popper.min.js" integrity="sha384oBqDVmMz9ATKxIep9tiCxS/
Z9fNfEXiDAYTujMAeBAsjFuCZSmKbSSUnQlmh/jp3"
        crossorigin="anonymous"
    ></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/
bootstrap.min.js"
integrity="sha384-cuYeSxntonz0PPNlHhBs68uyIAVpIIOZZ5JqeqvYYIcEL727ksk
C66kF92t6Xl2V"
        crossorigin="anonymous"
    ></script>
  </body>
</html>
```

Listing 6.4: dashboard - js

```
 <!DOCTYPE html>
<html lang="en">

<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>TickTask - To-Do List</title>
<link rel="stylesheet" href="styles.css">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/
font-awesome/6.4.0/css/all.min.css">
</head>

<body>
<div class="mobile-menu-btn">
<i class="fas fa-bars"></i>
</div>

<div class="container">
<aside class="sidebar">
<div class="profile">
<div class="profile-edit">
<img src="assets/profile.png" alt="Profile" class="profile-img"
id="profile-pic">
<button class="edit-profile-btn"><i class="fas fa-camera"></i></button>
<input type="file" id="profile-upload" accept="image/*" style="display:
```

```
none;">
</div>
<h3 id="profile-name">ayan</h3>
<p id="profile-email">ayan@email.com</p>
</div>
<nav>
<ul>
 <li class="active" data-section="tasks"><i class="fas fa-tasks"></i>
 Tasks</li>
<li data-section="calendar"><i class="fas fa-calendar-alt"></i>
Calendar</li>
<li data-section="analytics"><i class="fas fa-chart-pie"></i>
Analytics</li>
 </ul>
 </nav>
<div class="upgrade-banner">
<p>Upgrade to <strong>Pro</strong> for more features!</p>
<button>Upgrade</button>
</div>
</aside>
<main class="content">
<div id="tasks-section" class="content-section active">
<header>
<h1>Today's Tasks</h1>
<div class="date-display">
<span id="current-date">Monday, June 10, 2024</span>
</div>
 </header>
<div class="task-input">
<input type="text" id="new-task" placeholder="Add a new task...">
<div class="task-options">
<select id="task-type">
<option value="work">Work</option>
<option value="personal">Personal</option>
<option value="medical">medical</option>
<option value="study">Study</option>
 </select>
<input type="date" id="task-date">
<select id="task-priority">
 <option value="low">Low</option>
<option value="medium" selected>Medium</option>
 <option value="high">High</option>
</select>
<button id="add-task"><i class="fas fa-plus"></i> Add</button>
</div>
</div>
 <div class="task-filters">
<button class="filter-btn active" data-filter="all">All</button>
```

```
<button class="filter-btn" data-filter="today">Today</button>
<button class="filter-btn" data-filter="high">High Priority</button>
</div>
<div class="task-list" id="task-list">
</div>
</div>

<div id="calendar-section" class="content-section">
<h2><i class="fas fa-calendar-alt"></i> Calendar</h2>
<div class="calendar" id="calendar">
</div>
</div>

<div id="analytics-section" class="content-section">
<h2><i class="fas fa-chart-pie"></i> Analytics</h2>
<div class="analytics-content">
<div class="stats-container">
<div class="stat-card">
<h3>Total Tasks</h3>
<p id="total-tasks">0</p>
</div>
<div class="stat-card">
<h3>Completed</h3>
<p id="completed-tasks">0</p>
</div>
<div class="stat-card">
<h3>Pending</h3>
<p id="pending-tasks">0</p>
</div>
</div>
<div class="chart-container">
<canvas id="tasks-chart"></canvas>
</div>
</div>
</div>
</main>
</div>

<script src="scriptindex.js"></script>
<script>
    document.addEventListener('DOMContentLoaded', function() {
const profileUpload = document.getElementById('profile-upload');
const profilePic = document.getElementById('profile-pic');
const editProfileBtn = document.querySelector('.edit-profile-btn');
 const profileName = document.getElementById('profile-name');
 const profileEmail = document.getElementById('profile-email');

editProfileBtn.addEventListener('click', () => profileUpload.click());
```

```
 profileUpload.addEventListener('change', function(e) {
if (e.target.files.length) {
const reader = new FileReader();
reader.onload = function(event) {
profilePic.src = event.target.result;
localStorage.setItem('profilePic', event.target.result);
};
reader.readAsDataURL(e.target.files[0]);
}
});

if (localStorage.getItem('profilePic')) {
profilePic.src = localStorage.getItem('profilePic');
}
if (localStorage.getItem('profileName')) {
profileName.textContent = localStorage.getItem('profileName');
}
if (localStorage.getItem('profileEmail')) {
profileEmail.textContent = localStorage.getItem('profileEmail');
}

const sidebar = document.querySelector('.sidebar');
const mobileMenuBtn = document.querySelector('.mobile-menu-btn');

mobileMenuBtn.addEventListener('click', function() {
sidebar.classList.toggle('active');
});

const navItems = document.querySelectorAll('nav ul li');
const contentSections = document.querySelectorAll('.content-section');

navItems.forEach(item => {
item.addEventListener('click', function() {
navItems.forEach(navItem => navItem.classList.remove('active'));
contentSections.forEach(section
=section.classList.remove('active'));

this.classList.add('active');
const sectionId = `${this.dataset.section}-section`;
document.getElementById(sectionId).classList.add('active');

            if (window.innerWidth <= 768) {
              sidebar.classList.remove('active');
            }

            if (this.dataset.section === 'calendar') {
              generateCalendar();
```

```
    } else if (this.dataset.section === 'analytics') {
      updateAnalytics();
    }
  });
});const currentDate = new Date();const options = { weekday:

'long',year: 'numeric', month:'long', day: 'numeric' };

document.getElementById('currentdate').
textContent=currentDate.toLocaleDateString('en-US', options);

let tasks = JSON.parse(localStorage.getItem('tasks')) || [];

const taskInput = document.getElementById('new-task');
const taskType = document.getElementById('task-type');
const taskDate = document.getElementById('task-date');
const taskPriority = document.getElementById('task-priority');
const addTaskBtn = document.getElementById('add-task');
const taskList = document.getElementById('task-list');
const filterButtons = document.querySelectorAll('.filter-btn');
const calendar = document.getElementById('calendar');

renderTasks();
generateCalendar();

addTaskBtn.addEventListener('click', addTask);
taskInput.addEventListener('keypress', function(e) {
  if (e.key === 'Enter') addTask();
});

filterButtons.forEach(button => {
  button.addEventListener('click', function() {
    filterButtons.forEach(btn => btn.classList.remove('active'));
    this.classList.add('active');
    renderTasks(this.dataset.filter);
  });
});

function addTask() {
  const title = taskInput.value.trim();
  if (!title) return;

  const newTask = {
    id: Date.now(),
    title,
    type: taskType.value,
    date: taskDate.value || formatDate(currentDate),
    priority: taskPriority.value,
```

```
            completed: false,
            createdAt: new Date().toISOString()
          };

          tasks.unshift(newTask);
          saveTasks();
          renderTasks();
          generateCalendar();

          taskInput.value = '';
          taskDate.value = '';
        }

        function renderTasks(filter = 'all') {
         taskList.innerHTML = '';

       let filteredTasks = [...tasks];

        if (filter === 'today') {
        const today = formatDate(currentDate);
       filteredTasks = tasks.filter(task => task.date === today);
          } else if (filter === 'high') {
filteredTasks = tasks.filter(task => task.priority === 'high');
          }

       if (filteredTasks.length === 0) {
        taskList.innerHTML = '<p class="no-tasks">No tasks found</p>';
            return;
          }

          filteredTasks.forEach(task => {
            const taskElement = document.createElement('div');
       taskElement.className = `task-item ${task.completed
       'completed' :''}`  taskElement.innerHTML = `
              <div class="task-info">
                <div class="task-title">${task.title}</div>
                <div class="task-meta">
    <span class="task-type ${task.type}">${task.type}</span>
  <span>${formatDisplayDate(task.date)}</span>
   <span class="task-priority ${task.priority}">${task.priority}</span>
                </div>
              </div>
        <div class="task-actions">
          <button class="complete-btn" data-id="${task.id}">
                  <i class="fas fa-check"></i>
        </button>
              <button class="delete-btn" data-id="${task.id}">
                <i class="fas fa-trash"></i>
```

```
            </button>
          </div>
        `;
        taskList.appendChild(taskElement);
      });

      document.querySelectorAll('.complete-btn').forEach(button => {
        button.addEventListener('click', toggleComplete);
      });

      document.querySelectorAll('.delete-btn').forEach(button => {
        button.addEventListener('click', deleteTask);
      });
    }

    function toggleComplete(e) {
      const taskId = parseInt(e.currentTarget.dataset.id);
      const taskIndex = tasks.findIndex(task => task.id === taskId);

if (taskIndex !== -1) {
tasks[taskIndex].completed = !tasks[taskIndex].completed;
saveTasks();
renderTasks(document.querySelector('.filter-btn.active').dataset.filter);
}
}
function deleteTask(e) {
const taskId = parseInt(e.currentTarget.dataset.id);
tasks = tasks.filter(task => task.id !== taskId);
saveTasks();
renderTasks(document.querySelector('.filter-btn.active').dataset.filter);
generateCalendar();
updateAnalytics();
}
function saveTasks() {
localStorage.setItem('tasks', JSON.stringify(tasks));
    }
function generateCalendar() {
calendar.innerHTML = '';
const days = ['Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat'];
 days.forEach(day => {
onst header = document.createElement('div');
header.className = 'calendar-header';
header.textContent = day;
calendar.appendChild(header);
});

const year = currentDate.getFullYear();
const month = currentDate.getMonth();
```

```
onst firstDay = new Date(year, month, 1).getDay();
const daysInMonth = new Date(year, month + 1, 0).getDate();

for (let i = 0; i < firstDay; i++) {
const emptyDay = document.createElement('div');
emptyDay.className = 'calendar-day empty';
calendar.appendChild(emptyDay);

for (let day = 1; day <= daysInMonth; day++) {
const dateStr = `${year}-${String(month + 1).padStart(2,'0')}-
${String(day).padStart(2, '0')}`;

const dayElement = document.createElement('div');
dayElement.className = 'calendar-day';
if (day === currentDate.getDate() && month

===currentDate.getMonth() && year ===
currentDate.getFullYear()) {

dayElement.classList.add('today');
          }


const hasTasks = tasks.some(task => task.date === dateStr);
if (hasTasks) dayElement.classList.add('has-tasks');
 dayElement.innerHTML = `
      <div>${day}</div>
        `;
  calendar.appendChild(dayElement);
 }
  }

function updateAnalytics() {
const totalTasks = tasks.length;
const completedTasks = tasks.filter
(task => task.completed).length;
const pendingTasks = totalTasks - completedTasks;


document.getElementById('total-tasks').textContent = totalTasks;
 document.getElementById('completed-tasks').
 textContent = completedTasks;
document.getElementById('pending-tasks').
textContent pendingTasks;
      }

      function formatDate(date) {
        const year = date.getFullYear();
```

```
        const month = String(date.getMonth() + 1).padStart(2, '0');
        const day = String(date.getDate()).padStart(2, '0');
        return `${year}-${month}-${day}`;
      }

      function formatDisplayDate(dateStr) {
const date = new Date(dateStr);
return date.toLocaleDateString('en-US',
{ month: 'short', day: 'numeric' });
      }
    });
 </script>
 <style>
    * {
      margin: 0;
      padding: 0;
      box-sizing: border-box;
      font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    }

    body {
      background-color: #f5f7fa;
      color: #333;
    }

    .container {
      display: flex;
      min-height: 100vh;
    }

    .mobile-menu-btn {
      display: none;
      position: fixed;
      top: 15px;
      left: 15px;
      z-index: 1000;
      background: #2c3e50;
      color: white;
      width: 40px;
      height: 40px;
      border-radius: 50%;
      display: flex;
      align-items: center;
      justify-content: center;
      cursor: pointer;
      box-shadow: 0 2px 5px rgba(0,0,0,0.2);
    }
```

```
.sidebar {
  width: 250px;
  background: #2c3e50;
  color: white;
  padding: 20px;
  display: flex;
  flex-direction: column;
  transition: transform 0.3s ease;
}

.profile {
  text-align: center;
  margin-bottom: 30px;
}

.profile-edit {
  position: relative;
  display: inline-block;
}

.profile-img {
  width: 80px;
  height: 80px;
  border-radius: 50%;
  object-fit: cover;
  margin-bottom: 10px;
  border: 3px solid #3498db;
}

.edit-profile-btn {
  position: absolute;
  bottom: 5px;
  right: 5px;
  background: #3498db;
  color: white;
  border: none;
  width: 30px;
  height: 30px;
  border-radius: 50%;
  cursor: pointer;
  display: flex;
  align-items: center;
  justify-content: center;
}

.sidebar nav ul {
  list-style: none;
}
```

```
.sidebar nav ul li {
  padding: 12px 15px;
  margin: 5px 0;
  border-radius: 5px;
  cursor: pointer;
  display: flex;
  align-items: center;
  gap: 10px;
}

.sidebar nav ul li:hover {
  background: #34495e;
}

.sidebar nav ul li.active {
  background: #3498db;
}

.sidebar nav ul li i {
  width: 20px;
  text-align: center;
}

.upgrade-banner {
  margin-top: auto;
  background: rgba(255, 255, 255, 0.1);
  padding: 15px;
  border-radius: 8px;
  text-align: center;
}

.upgrade-banner button {
  background: #f39c12;
  color: white;
  border: none;
  padding: 8px 15px;
  border-radius: 5px;
  margin-top: 10px;
  cursor: pointer;
  width: 100%;
}

.content {
  flex: 1;
  padding: 30px;
}
```

```
.content-section {
  display: none;
}

.content-section.active {
  display: block;
}

header {
  margin-bottom: 30px;
}

header h1 {
  font-size: 28px;
  color: #2c3e50;
}

.date-display {
  color: #7f8c8d;
  font-size: 14px;
  margin-top: 5px;
}

.task-input {
  margin-bottom: 25px;
}

.task-input input[type="text"] {
  width: 100%;
  padding: 12px 15px;
  border: 1px solid #ddd;
  border-radius: 5px;
  font-size: 16px;
  margin-bottom: 10px;
}

.task-options {
  display: flex;
  gap: 10px;
}

.task-options select,
.task-options input[type="date"] {
  padding: 10px;
  border: 1px solid #ddd;
  border-radius: 5px;
}
```

```
.task-options button {
  padding: 10px 15px;
  background: #3498db;
  color: white;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}

.task-filters {
  display: flex;
  gap: 10px;
  margin-bottom: 20px;
}

.filter-btn {
  padding: 8px 15px;
  background: #ecf0f1;
  border: none;
  border-radius: 20px;
  cursor: pointer;
}

.filter-btn.active {
  background: #3498db;
  color: white;
}

.task-list {
  display: flex;
  flex-direction: column;
  gap: 10px;
}

.task-item {
  background: white;
  padding: 15px;
  border-radius: 8px;
  box-shadow: 0 2px 5px rgba(0,0,0,0.1);
  display: flex;
  justify-content: space-between;
  align-items: center;
  transition: all 0.3s ease;
}

.task-item:hover {
  transform: translateY(-2px);
  box-shadow: 0 4px 8px rgba(0,0,0,0.1);
```

```
}

.task-info {
  flex: 1;
}

.task-title {
  font-weight: 600;
  margin-bottom: 5px;
}

.task-meta {
  display: flex;
  gap: 15px;
  font-size: 13px;
  color: #7f8c8d;
}

.task-type {
  padding: 3px 8px;
  border-radius: 10px;
  font-size: 12px;
}

.task-type.work {
  background: #e3f2fd;
  color: #1976d2;
}

.task-type.personal {
  background: #e8f5e9;
  color: #388e3c;
}

.task-type.shopping {
  background: #fff3e0;
  color: #f57c00;
}

.task-priority {
  font-weight: bold;
}

.task-priority.high {
  color: #e53935;
}

.task-priority.medium {
```

```
  color: #fb8c00;
}

.task-priority.low {
  color: #43a047;
}

.task-actions {
  display: flex;
  gap: 10px;
}

.task-actions button {
  background: none;
  border: none;
  cursor: pointer;
  color: #7f8c8d;
  font-size: 16px;
}

.task-actions button:hover {
  color: #3498db;
}

.completed {
  text-decoration: line-through;
  opacity: 0.7;
}

.calendar-section {
  margin-top: 40px;
}

.calendar {
  display: grid;
  grid-template-columns: repeat(7, 1fr);
  gap: 10px;
  margin-top: 15px;
}

.calendar-header {
  font-weight: bold;
  text-align: center;
  padding: 10px;
}

.calendar-day {
  background: white;
```

```
  padding: 10px;
  border-radius: 5px;
  text-align: center;
  min-height: 60px;
  box-shadow: 0 1px 3px rgba(0,0,0,0.1);
}

.calendar-day.empty {
  visibility: hidden;
}

.calendar-day.today {
  background: #e3f2fd;
  font-weight: bold;
}

.calendar-day.has-tasks::after {
  content: "";
  display: block;
  width: 6px;
  height: 6px;
  background: #3498db;
  border-radius: 50%;
  margin: 5px auto 0;
}

.analytics-content {
  margin-top: 20px;
}

.stats-container {
  display: flex;
  gap: 20px;
  margin-bottom: 30px;
}

.stat-card {
  flex: 1;
  background: white;
  padding: 20px;
  border-radius: 8px;
  box-shadow: 0 2px 5px rgba(0,0,0,0.1);
  text-align: center;
}

.stat-card h3 {
  color: #7f8c8d;
  font-size: 16px;
```

```
        margin-bottom: 10px;
    }

    .stat-card p {
        font-size: 24px;
        font-weight: bold;
        color: #2c3e50;
    }

    .chart-container {
        background: white;
        padding: 20px;
        border-radius: 8px;
        box-shadow: 0 2px 5px rgba(0,0,0,0.1);
        height: 300px;
    }

    @media (max-width: 768px) {
        .mobile-menu-btn {
            display: flex;
        }

        .sidebar {
            position: fixed;
            top: 0;
            left: 0;
            height: 100%;
            z-index: 999;
            transform: translateX(-100%);
        }

        .sidebar.active {
            transform: translateX(0);
        }
        .container {
            flex-direction: column;
        }
        .content {
            padding: 20px;
            padding-top: 70px;
        .task-options {
            flex-wrap: wrap;
        }
        .stats-container {
            flex-direction: column;}}
    </style>
</body>
</html>
```

## 6.2 RESULTS AND DISCUSSION

The TickTask To-Do List Management System was developed to enhance task organization and productivity. This section presents the results achieved and provides a discussion on system performance, usability, and feedback.

### 6.2.1 RESULTS

The following key features were successfully implemented:

- **User Authentication:** The system supports secure login functionality with error handling for invalid credentials.
- **Task Management:** Users can create, edit, mark tasks as complete, and delete tasks seamlessly.
- **User Interface:** The design incorporates an intuitive layout with clear navigation and visual appeal.
- **Customization:** Users can personalize their workspace with themes and layout options.

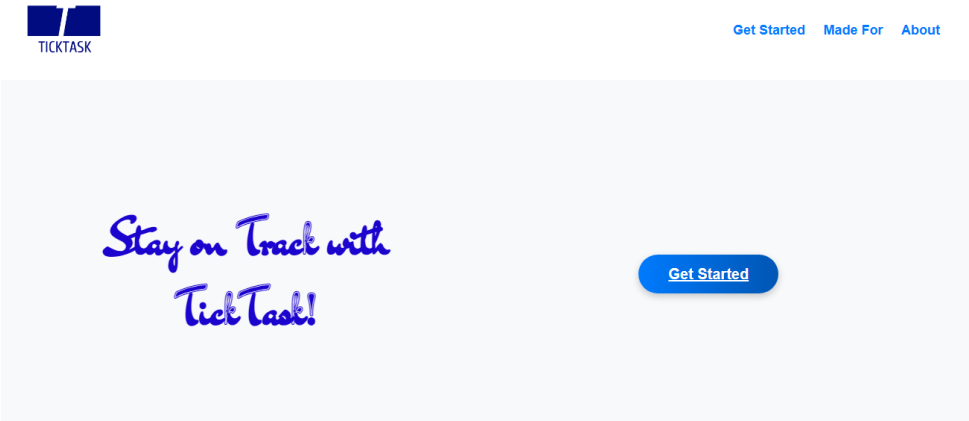



Figure 6.1: TickTask home Interface

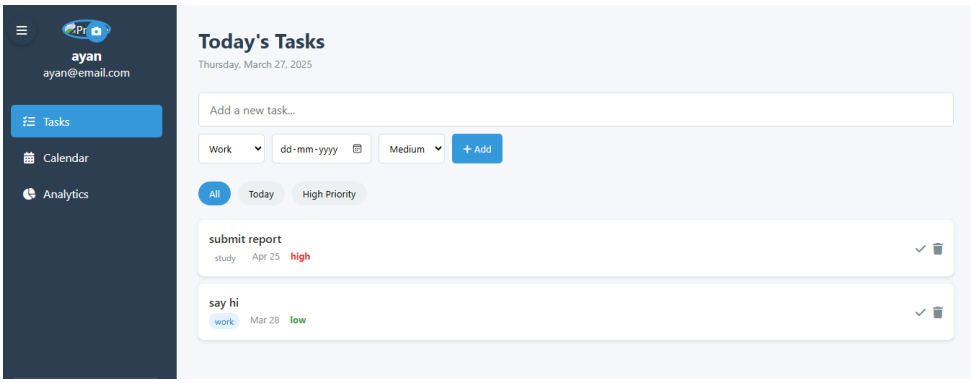



Figure 6.2: TickTask Dashboard Interface

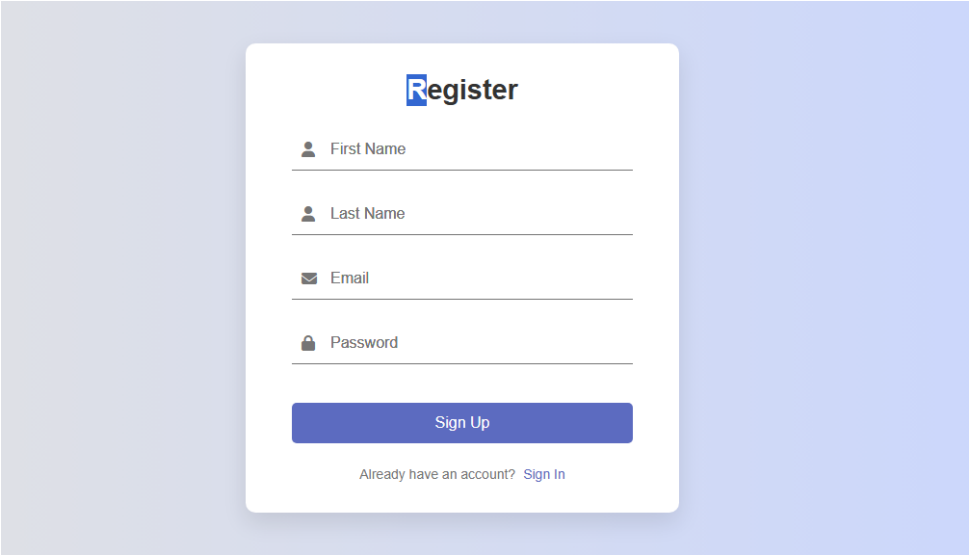



Figure 6.3: sign up

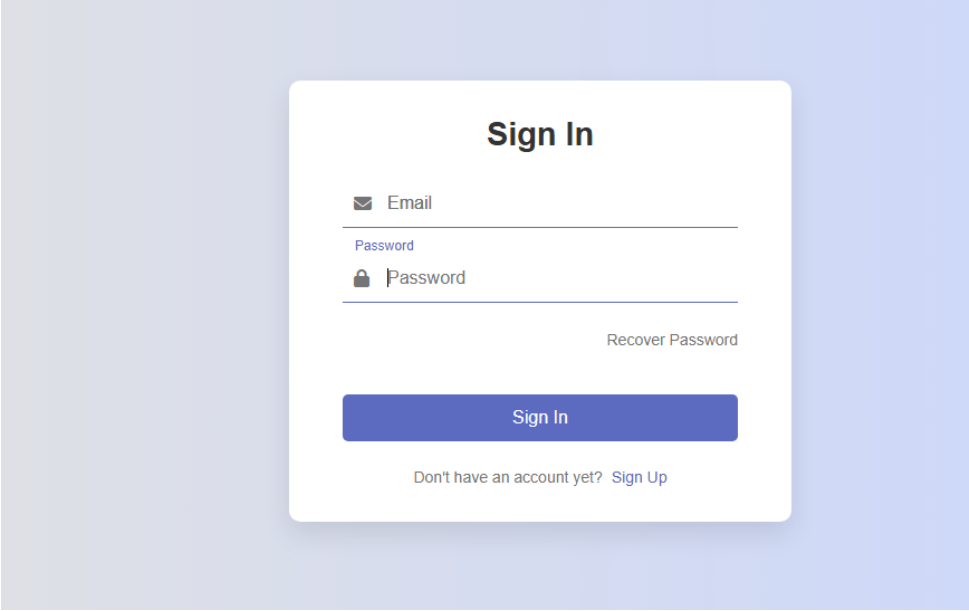



Figure 6.4: sign in

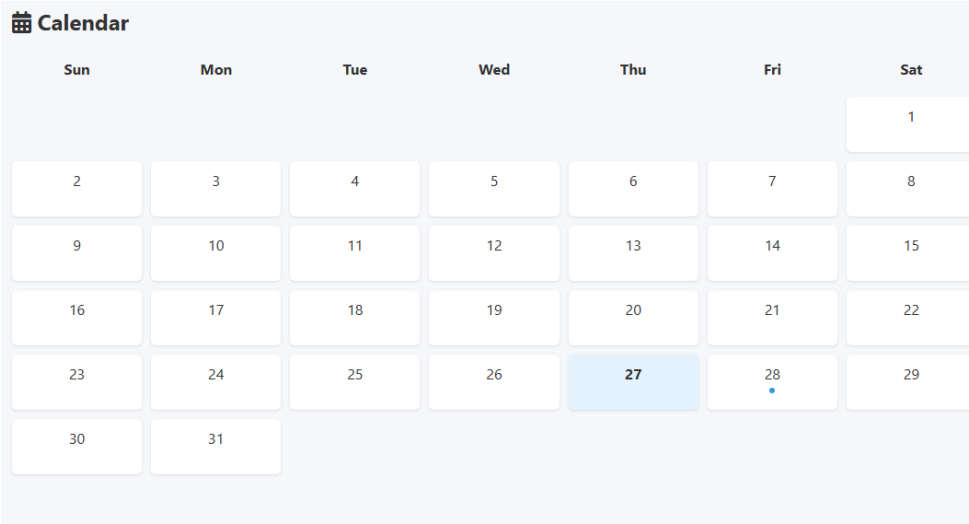



Figure 6.5: TickTask calender

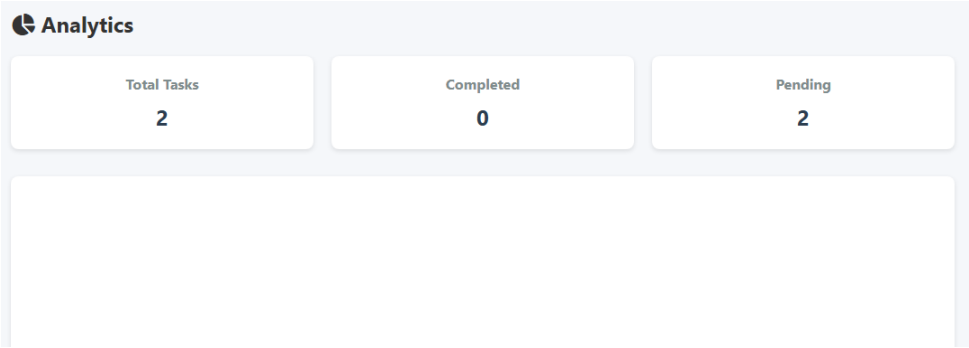



Figure 6.6: analytics

### 6.2.2 DISCUSSION

The development of TickTask presented several key insights:

- **Efficiency:** The use of efficient JavaScript frameworks like GSAP enhanced user experience with smooth animations and fast load times.
- **User Feedback:** Initial user testing indicated that users appreciated the intuitive interface, particularly the drag-and-drop functionality for task management.
- **Challenges:** Integrating real-time collaboration posed challenges, particularly in maintaining data consistency. Implementing WebSocket technology helped resolve this issue.
- **Future Enhancements:** Upcoming features may include calendar integration, reminders, and improved analytics for task tracking.

The system has proven effective for both individual and team-based task management, enhancing productivity and simplifying project organization.

# Chapter 7

# CONCLUSION

The Tick Task - To-Do List application has been developed with the primary objective of enhancing task management, organization, and collaboration for individuals and teams. In today's fast-paced world, efficient time management is essential, and traditional to-do lists often lack the necessary features for real-time collaboration and advanced tracking. Tick Task bridges this gap by incorporating **task creation, assignment, tracking, notifications, and collaboration tools, providing a structured approach to improving productivity. The application ensures that tasks are managed efficiently, deadlines are met, and accountability is maintained among users. A key highlight of Tick Task is its user-friendly interface, which allows users to interact

with the system seamlessly. The integration of real-time updates, role-based access control, and automated notifications ensures that users remain informed about task progress at all times. These features help eliminate confusion in task delegation and ensure that responsibilities are clearly assigned and tracked. Furthermore, task prioritization and categorization allow users to focus on high-priority assignments, reducing inefficiencies and improving overall workflow. In conclusion, the Tick Task - To-Do List system serves as a comprehensive and reliable task

management solution, providing an efficient, intuitive, and collaborative platform for users. The structured approach taken in designing the system ensures its effectiveness in enhancing task organization and teamwork. Future work will focus on optimizing system performance, improving the user experience, and integrating additional collaborative and analytical features. By continuously evolving to meet the needs of users, Tick Task aims to be a leading productivity tool that supports seamless task management and efficiency enhancement.

# REFERENCES


[1] S. Caballé, F. Xhafa, and L. Barolli, "Using mobile devices to support online collaborative learning," Mobile Information Systems, vol. 6, no.1, pp. 27-47, 2010.

[2] Y. Chen and D.Z. Wang, "Knowledge expansion over probabilistic knowledge bases," in Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data, 2014.

[3] M. Craglia, M. F. Goodchild, A. Annoni, G. Camara, M. F. Gould, W. Kuhn, D. Mark, L. Masser, D. Maguire, and S. Liang, "Next-generation digital earth," 2008.

[4] M. S. Hossain, C. Xu, Y. Li, A. K. Pathan, J. Bilbao, W. Zeng, and A. El-Saddik, "Impact of Next-Generation Mobile Technologies on IoT-Cloud Convergence," IEEE Communications Magazine, vol. 55, no.1, pp. 18-19, 2017.

[5] M. Miller, *Cloud computing: Web-based applications that change the way you work and collaborate online*, Que Publishing, 2008.

[6] E. Ramadani, A. Memeti, F. Imeri, N. Sejfuli-Ramadani, and F. Idrizi, "Social media-based app organizing daily events," in 2017 6th Mediterranean Conference on Embedded Computing (MECO), IEEE, 2017.

[7] M. S. Ahmad, N. E. Musa, R. Nadarajah, R. Hassan, and N. E. Othman, "Comparison between Android and iOS Operating System in terms of security," in 2013 8th International Conference on Information Technology in Asia (CITA), IEEE, 2013.

[8] N. Yorke-Smith, S. Saadati, K. L. Myers, and D. N. Morley, "The design of a proactive personal agent for task management," International Journal on Artificial Intelligence Tools, vol. 21, no. 01, pp. 1250004, 2012.

[9] X. Cheng, L. Fang, L. Yang, and S. Cui, "Mobile big data: The fuel for data-driven wireless," IEEE Internet of Things Journal, vol. 4, no. 5, pp. 1489-1516, 2017.