**A**

**PROJECT REPORT**

**ON**

**TASK MANAGER USING DJANGO TECHNOLOGY**

**SUBMITTED BY**

**MS.SHIVAMMA KOGNYRE**

**UNDER THE GUIDANCE OF**

**PROF. P. D. JAWALKAR**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**P.N.E.S.P.’S**

**NAGESH KARAJAGI *ORCHID* COLLEGE OF ENGG. & TECH**

**SOLAPUR – 413 002**



**AFFILIATED TO**

**[DR.BABASAHEB AMBEDKAR TECHNOLOGICAL UNIVERSITY](https://dbatu.ac.in/) LONERE**

i

**CERTIFICATE**

This is to certify that the project entitled **TASK MANAGER USING DJANGO TECHNOLOGY** has

been completed by following students of B.E (CSE) class in a satisfactory manner under my guidance.

**SHIVAMMA KOGNURE**

The project is found to be complete in partial fulfilment for the award of Degree of Bachelor of Technology of Computer Science And Engineering of DBATU University , Lonere.

**(PROF. P.D. JAWALKAR) (Prof. V.V. BAG) (Dr. J. B. DAFEDAR)**

**PROJECT GUIDE HOD PRINCIPAL**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**P.N.E.S.P.’S**

**NAGESH KARAJAGI *ORCHID* COLLEGE OF ENGG. & TECH**

**SOLAPUR – 413 002**



**AFFILIATED TO**

**[DR.BABASAHEB AMBEDKAR TECHNOLOGICAL UNIVERSITY](https://dbatu.ac.in/) LONERE**

ii

**Code of Ethics for Plagiarism**

1. I/We know that plagiarism means taking and using the ideas, writings, works or inventions of another person as if they were one’s own. I/We know that plagiarism not only includes verbatim copying, but also the extensive use of another person’s ideas without proper acknowledgement (which includes the proper use of quotation marks). I/We know that plagiarism covers this sort of use of material found in textual sources and from the Internet.
2. I/We acknowledge and understand that plagiarism is wrong.
3. I/We understand that my research must be accurately referenced. I/We have followed the rules and convention concerning referencing, citation and the use of quotation as set out in the Departmental Guide.
4. This project work is my/our own work, or my/our group’s unique group project work. I/We acknowledge that copying someone else work or part of it, is wrong, and that submitting identical work to others constitutes a form of plagiarism.

5. I/We have not allowed, nor will I/We in the future allow, anyone to copy my/our work with the intention of passing it off as their own work.

……………………………………………………………

……………………………………………………………

……………………………………………………………

……………………………………………………………

(Name & Signature of the Student/s)

Place: NKOCET, Solapur

Date:

iii



|  |  |
| --- | --- |
|  | **DECLARATION** |

By the UG/PG (B.E./M.E.) Student

I/We hereby declare that the Report of the UG/P.G. Project Work entitled

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

Which is being submitted to the Solapur University, Solapur in Partial ful-fillment of the requirements for

the award of the Degree Bachelor/Master of ………………………………… in the department of

…………………………………………………………………..…. , is a bonafide report of the work carried

out by me/us. The material contained in this report has not been submitted to any University or Institution

for the award of any degree.

……………………………………………………………

……………………………………………………………

……………………………………………………………

……………………………………………………………

(Name & Signature of the Student/s)

Place: NKOCET, Solapur

Date:

iv

**PROJECT APPROVAL SHEET**

The project entitled **TASK MANAGER USING DJANGO TECHNOLOGY** submitted by the following students -

**SHIVAMMA KOGNURE**

is hereby approved in partial fulfilment for the award of Degree of Bachelor of Technology of Computer Science And Engineering of DBATU University , Lonere.

EXAMINERS

1. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**P.N.E.S.P.’S**

**NAGESH KARAJAGI *ORCHID* COLLEGE OF ENGG. & TECH**

**SOLAPUR – 413 002**



**AFFILIATED TO**

**[DR.BABASAHEB AMBEDKAR TECHNOLOGICAL UNIVERSITY](https://dbatu.ac.in/) LONERE**

v

**ABSTRACT**

The objective of this project is to design and implement a comprehensive Task Manager application using Django, a powerful web framework for Python. The Task Manager system provides users with a seamless and efficient way to manage their tasks, enabling them to create, update, and delete tasks as needed. Additionally, the application offers advanced features such as task prioritization, allowing users to categorize tasks based on their importance or urgency. Furthermore, the system incorporates the ability to set due dates for tasks, helping users stay organized and meet deadlines effectively. To enhance productivity, the Task Manager also includes reminders and notifications to ensure that users are aware of upcoming tasks. The implementation of the system utilizes Django's robust features and leverages the SQLite database for efficient data storage and retrieval. Through the development of this Task Manager application, users will experience improved task management capabilities, facilitating better organization and productivity in their personal or professional lives.

Furthermore, the Task Manager application incorporates a user-friendly interface that promotes ease of use and intuitive navigation. Users can effortlessly create new tasks, providing detailed descriptions and assigning relevant tags or categories to better organize their workflow. The application's intuitive design allows users to quickly access task details, update task statuses, and track progress. Additionally, the Task Manager offers search and filtering functionalities, empowering users to efficiently locate specific tasks based on various criteria such as due dates, priority levels, or assigned tags. The system also supports multiple user accounts, ensuring secure access and personalized task management for individual users or teams. With its robust features and user-centric design, the Task Manager application offers a comprehensive solution for effectively managing tasks, increasing productivity, and promoting efficient collaboration.

vi

**Acknowledgment**

I would like to express my sincere gratitude to our project guide, Prof. P. D. Jawalkar for his invaluable guidance, expertise, and unwavering support throughout the duration of this project. Her extensive knowledge, insightful feedback, and constant encouragement have played a pivotal role in shaping the development and success of the Task Manager application.

We would also like to extend our heartfelt appreciation to Principle J. B. Dafedar for his support and encouragement throughout the project. His guidance and valuable inputs have been instrumental in our project's progress and success.

Furthermore, we would like to thank our friends and colleagues who have provided encouragement, assistance, and valuable insights during the project's lifecycle.

Our sincere gratitude goes out to all individuals who have supported us in various ways during the completion of this project. Their contributions have been instrumental in making this endeavor a success.

vii

**INDEX**

**Abstract** -------------------------------------------------------------------------------------------------------------------00

List of figures ………………….. ……………………………………………………………………..……..vi

List of figures ……………………………………………………………………………………..…………vii

**1. INTRODUCTION**

1.1 General Introduction………………………………………………………………………………...……00

1.2 Motivation………………………………………………………………………………………...……...00

1.3 Problem Statement……………………...………………………………………………………………...00

1.4 Summary……………………...…………………………………………………………………………..00

**2. LITERATURE REVIEW**……………………………..………………………………………………....00

2.1 Summary……………………...…………………………………………………………………………..00

**3. SYSTEM DESCRIPTION**

3.1 General Introduction……………………….…………………………………………………………..…00

3.2 Block Diagram/Flow Chart.……………………………………………………………………..……….00

3.3 Summary………………………………………………………………...……………………………..…00

**4. HARDWARE DESCRIPTION**

4.1. Server Infrastructure……………………….………………………………………………………..…00 4.2. Client Devices……………………….…………………………………………………………..………00 4.3. Additional Hardware Dependencies ………………………………………...…………………..………00 4.4.Summary……………………...…………………………………………………………………………..00

**5. SOFTWARE DESCRIPTION**

5.1 List of Software……………………….……………………………………...…………………..………00 5.2 Django……………………………….……………………….……………...…………………..……….00 5.3 Summary……………………...…………………………………………………………………………..00

**6. RESULTS AND DISCUSSION**

6.1 General Introduction ……………………….……………...…………………..…………………………00

6.2 Results Correlated with Problem Statement …………………..…………………………………………00

6.3 Discussion…. ………………………………………..……….……………...…………………..……….00

6.4 Summary…………………………………………………. ……………...…………………..………….00

**7. BUSINESS PLAN AND COMMERCIAL ASPECT**

7.1 Market Survey………………………………………..……….……………...…………………..………00

7.2 Feasible Analysis…………………………………………..……….……………...…………………..…00

7.3 Summary……………………………………………..……….……………...…………………..……….00

**8. CONCLUSION**

8.1 Application…………………………………………..……….……………...…………………..……….00

8.2Advantages and Disadvantage.……………………………..……….……………...…………………….00

8.3 Future Scope

8.4 Conclusion

**References**

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| TABLE NUMBER | TABLE NAME | PAGE NUMBER |
| 8.1.1 | TEST CASE |  |

**LIST OF FIGURE**

|  |  |  |
| --- | --- | --- |
| **FIG NUMBER** | **FIG NAME** | **PAGE NUMBER** |
| 3.1 | FLOW/UML DIAGRAM |  |
| 3.2 | USE CASE DIAGRAM |  |
| 8.1 | HOME\_PAGE |  |
| 8.2 | LOGIN\_PAGE |  |
| 8.3 | FORGOT\_PASSWORD |  |
| 8.4 | RESET\_PAGE |  |
| 8.5 | DASHBOARD |  |
| 8.6 | CREATE\_PROJECT |  |

**CHAPTER ONE**

**INTRODUCTION**

**1.1 General Introduction**

The "Task Manager Using Django" project aims to develop a web-based application that facilitates efficient task management. In today's fast-paced and demanding world, individuals and organizations often struggle to keep track of their tasks and effectively prioritize their work. The Task Manager application, built using Django, offers a comprehensive solution to address these challenges.

Django, a powerful web framework for Python, provides a solid foundation for developing robust and scalable applications. Leveraging Django's features and capabilities, the Task Manager application will enable users to create, update, and delete tasks seamlessly. Moreover, the application will incorporate essential functionalities such as setting task priorities, due dates, and reminders, empowering users to stay organized and meet deadlines effectively.

The Task Manager application will have a user-friendly interface, ensuring ease of use and intuitive navigation. Users will be able to create tasks with detailed descriptions, assign tags or categories for better organization, and easily access task details. The application will also support search and filtering options, enabling users to locate specific tasks based on various criteria such as due dates, priority levels, or assigned tags.

This project seeks to provide a practical and efficient task management solution that enhances productivity and promotes effective collaboration. By streamlining the task management process and providing users with a centralized platform to manage their tasks, the Task Manager application will contribute to improved organization and optimized workflow.

Overall, the Task Manager Using Django project aims to harness the power of Django's web development capabilities to create a user-friendly and feature-rich task management application. Through this project, users will experience enhanced productivity, better organization, and simplified task management processes.

**1.2 Motivation**

The development of the Task Manager application using Django is motivated by several factors that highlight the need for an efficient task management solution.

These motivations include:

1. Increasing Complexity of Tasks: In today's fast-paced world, individuals and organizations face an ever-growing volume of tasks that need to be managed effectively. The Task Manager application aims to simplify the task management process and provide users with a centralized platform to manage their tasks effectively.

2. Demands for Productivity and Efficiency: With increasing demands for productivity and efficiency, individuals and teams require tools that help them streamline their work processes. The Task Manager application, built on Django, offers advanced features and functionalities that enhance productivity and enable users to accomplish tasks in a more organized and systematic manner.

3. Collaborative Work Environments: In many professional settings, collaboration and teamwork are crucial for successful task completion. The Task Manager application addresses this by providing features that facilitate collaboration, such as assigning tasks to team members, sharing task updates, and enabling real-time communication within the application.

4. Mobile Workforce and Remote Collaboration: The rise of remote work and mobile workforces necessitates task management solutions that are accessible from anywhere, anytime. The Task Manager application, being web-based, allows users to access their tasks and manage them remotely, ensuring seamless collaboration and productivity even in distributed work environments.

5. Leveraging Django's Capabilities: Django's robust features and extensive ecosystem make it an ideal framework for building web applications. The Task Manager application leverages the power and flexibility of Django to provide a scalable, secure, and customizable task management solution.By addressing these motivations, the Task Manager application using Django aims to empower individuals and teams with an efficient and user-friendly tool for managing tasks, increasing productivity, and enabling seamless collaboration in today's dynamic work environments.

**1.3 Problem Statement**

The development of the Task Manager application using Django is driven by the need to address the limitations and challenges associated with traditional task management methods. The following problem statement outlines the key issues that the Task Manager application aims to overcome:

1. Lack of Organization: Traditional task management methods such as pen and paper or basic to-do lists often result in disorganization and difficulty in keeping track of tasks. Users may struggle to prioritize tasks effectively, leading to missed deadlines and decreased productivity.

2. Limited Collaboration: Collaborative work environments require efficient task management tools that enable seamless collaboration and coordination among team members. Existing methods may lack collaboration capabilities, hindering effective teamwork and hindering task progress tracking and communication.

3. Inefficient Task Prioritization: Without a structured system, individuals and teams may struggle to prioritize tasks based on their importance and urgency. This can lead to a lack of clarity in task execution and inefficient allocation of resources.

4. Accessibility and Mobility: In today's dynamic work environments, where remote work and mobile workforces are prevalent, it is crucial to have a task management solution that is accessible from anywhere and on various devices. Traditional methods may not offer the necessary flexibility and accessibility required for efficient task management.

5. Limited Task Tracking and Reporting: Monitoring task progress and generating reports on task completion can be challenging with conventional methods. The lack of comprehensive tracking and reporting functionalities makes it difficult to evaluate individual or team performance and identify areas for improvement.

By addressing these problems, the Task Manager application using Django aims to provide a solution that offers organized task management, enhanced collaboration, efficient task prioritization, accessibility across devices, and robust tracking and reporting capabilities.

**1.4 Summary**

The general introduction sets the stage by introducing the Task Manager Using Django project, highlighting its purpose and significance. The motivation section explains the driving factors behind the development of the application, addressing the need for efficient task management solutions in today's fast-paced and collaborative work environments. The problem statement identifies the limitations and challenges associated with traditional task management methods, emphasizing the need for an organized, collaborative, and accessible solution.

In summary, the Task Manager Using Django project aims to develop a web-based application that streamlines task management, enhances collaboration, and improves productivity. By leveraging Django's capabilities, the application provides users with a user-friendly interface, advanced features for task organization, efficient prioritization, and seamless collaboration. The project addresses the limitations of traditional methods by offering accessibility across devices, real-time tracking, and reporting functionalities. Overall, the Task Manager application aims to revolutionize task management, enabling individuals and teams to effectively manage tasks, boost productivity, and achieve their goals efficiently.

**CHAPTER TWO**

**LITERATURE REVIEW**

The literature review chapter aims to provide a comprehensive analysis of existing literature, research, and frameworks relevant to task management, web application development, and Django. It establishes a theoretical foundation and understanding of the field, identifies relevant theories, methodologies, and best practices, and explores the existing solutions and approaches in task management.

1. Task Management Principles:

Task management involves various principles that contribute to effective task organization and execution. The literature explores concepts such as task prioritization, time management, task dependency, and task allocation. It delves into different methods for prioritizing tasks based on urgency, importance, and impact on goals. Additionally, it examines theories and techniques related to time management, including the Eisenhower Matrix and Pomodoro Technique. The concept of task dependency is explored, considering how tasks are interrelated and the impact of dependencies on task sequencing and project management. Furthermore, strategies for efficient task allocation are discussed, taking into account factors such as individual skills, workload balancing, and team collaboration.

2. Web Application Development:

Web application development plays a vital role in the creation of the Task Manager application. The literature provides an overview of web development technologies, including HTML, CSS, JavaScript, and server-side programming languages such as Python. It explains the client-server architecture and the interaction between web servers and browsers in web application communication. The chapter also focuses on frameworks, particularly Django, highlighting their benefits in terms of rapid development, code reusability, and built-in security features. The features and advantages of Django, such as URL routing, templating engine, ORM, and admin interface, are discussed in detail. Additionally, the database management capabilities of Django, including support for various database systems and the use of migrations for schema updates, are explored. The chapter also addresses the security measures offered by Django, such as protection against common web vulnerabilities and built-in authentication and authorization mechanisms.

3. Existing Task Management Solutions:

The literature review examines popular task management applications, including Trello, Asana, and Todoist. These applications are analyzed in terms of their features, user interfaces, and underlying technologies/frameworks. Strengths and weaknesses are identified to gain insights into existing solutions and potential areas for improvement. By reviewing the strengths and weaknesses of these applications, the Task Manager application can incorporate desirable features and address potential shortcomings to provide a more comprehensive and user-friendly solution.

4. Research and Studies:

Relevant research papers, studies, and academic works related to task management and web application development are analyzed. These academic works contribute to the theoretical understanding and empirical evidence in the field. The methodologies used in research studies are reviewed, including user surveys, experiments, and case studies. The findings derived from these studies provide insights into user behavior, user experience (UX), and task management methodologies. The literature review evaluates the contributions made by these studies to the field of task management and informs the development of the Task Manager application by incorporating best practices and user-centered design principles.

The literature review chapter serves as a critical foundation for the subsequent chapters, guiding the development and implementation of the Task Manager application. It establishes the theoretical background, identifies relevant theories and methodologies, and provides insights into existing solutions and research in the field. This comprehensive analysis informs the decision-making process and ensures that the Task Manager application incorporates the latest advancements and best practices in task management and web application development.

**2.1 Summary**

The literature review chapter provides a comprehensive analysis of existing literature, research, and frameworks related to task management, web application development, and Django. It explores task management principles, including task prioritization, time management, task dependency, and task allocation. The chapter also focuses on web application development, discussing technologies such as HTML, CSS, JavaScript, and the advantages of using Django as a framework. Existing task management solutions, including Trello, Asana, and Todoist, are examined, highlighting their features and limitations. Additionally, the review incorporates research papers and studies that contribute to the theoretical understanding and empirical evidence in the field. Overall, the literature review serves as a foundation for the development of the Task Manager application, guiding the implementation of best practices and user-centered design principles.

**CHAPTER 3**

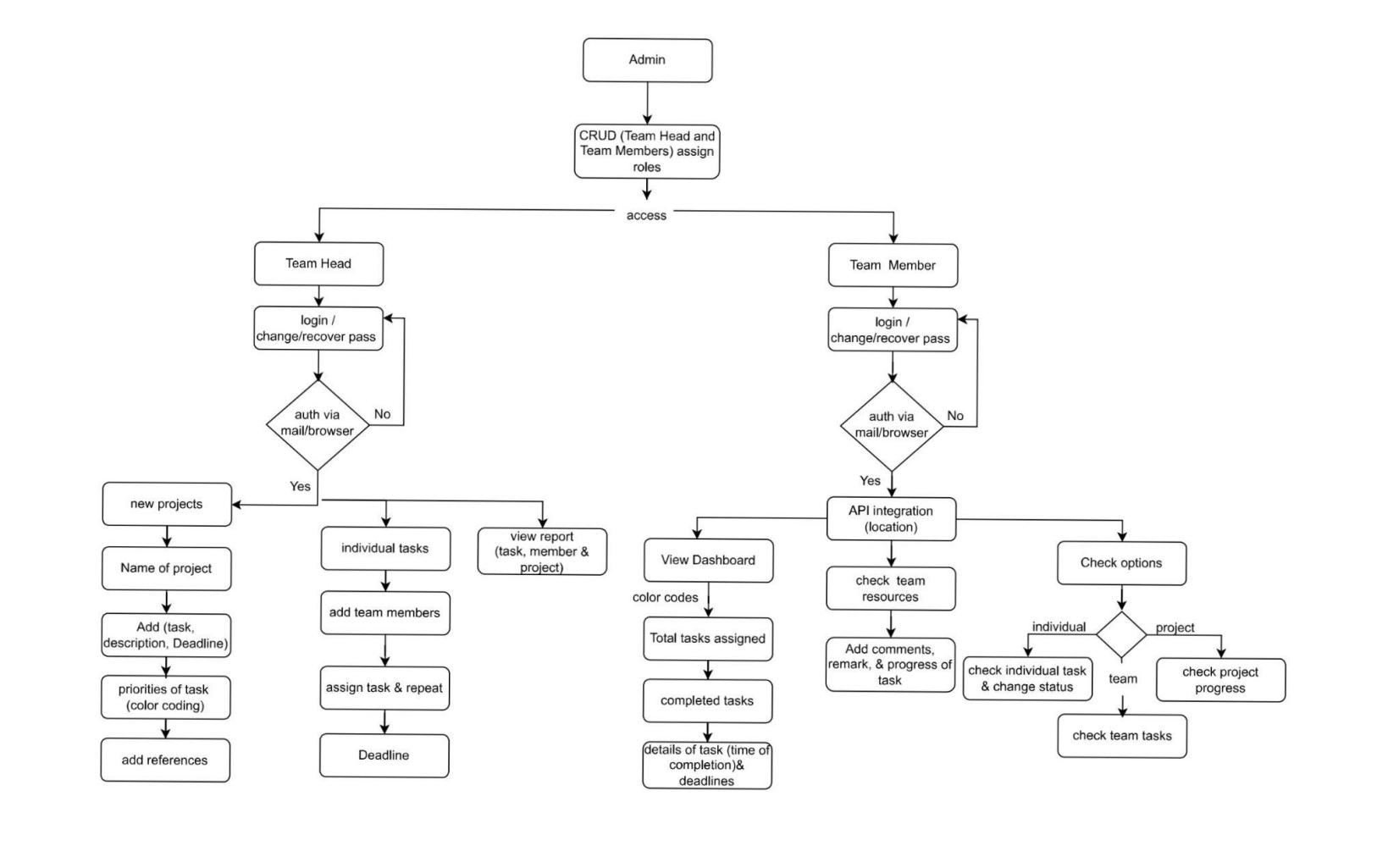
**SYSTEM DESCRIPTION**

**3.1 General Introduction**

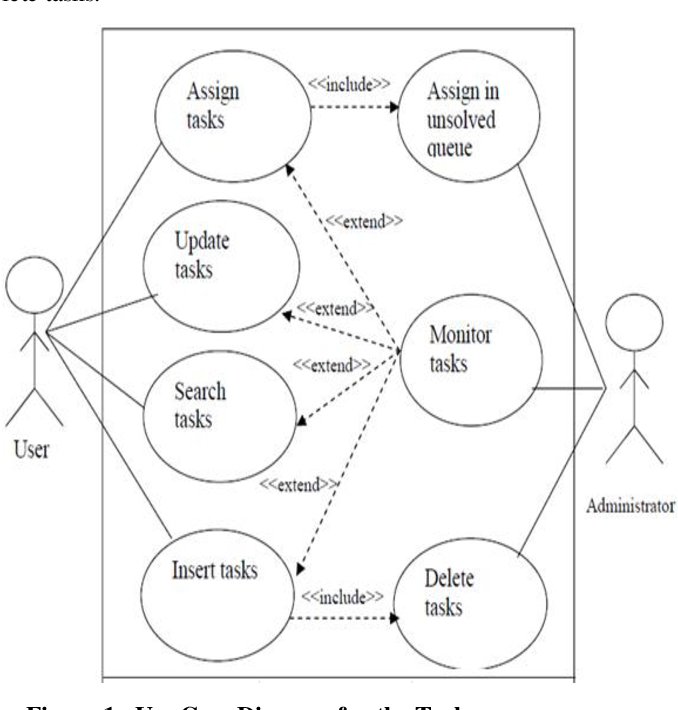
The system description section provides an overview of the Task Manager application built using Django. It includes a general introduction, a block diagram or flowchart illustrating the system architecture, a brief description of the building blocks/components, and a summary of the system.

**3.2 Block Diagram/Flow Chart and Brief Description**

The block diagram or flow chart visually represents the architecture and flow of the Task Manager application. It outlines the various components and their interactions within the system.



**FIG 3.1 FLOW/UML DIAGRAM**

****

**FIG 3.2 USE CASE DIAGRAM**

The Task Manager application comprises several building blocks that work together to provide efficient task management capabilities. These building blocks include:

1. User Interface: The user interface component is responsible for providing a user-friendly interface through which users can interact with the Task Manager application. It allows users to create, view, update, and delete tasks, set priorities and due dates, and manage task-related information.

2. Task Management Module: This module handles the core functionality of task management. It includes features such as task creation, task editing, task deletion, and task prioritization. It allows users to organize tasks, set reminders, and track task progress.

3. Database Management: The database management component utilizes a database system, such as SQLite, to store and retrieve task-related data. It handles the storage of tasks, user information, and other relevant data required for the Task Manager application.

4. User Authentication and Authorization: This component ensures secure access to the Task Manager application. It authenticates users, verifies their identities, and grants appropriate access levels based on user roles and permissions. User authentication and authorization mechanisms help protect sensitive data and maintain the integrity of the system.

5. Notifications and Reminders: The Task Manager application incorporates a notifications and reminders system to keep users informed about upcoming tasks, deadlines, and important updates. This component sends notifications and reminders to users via email, push notifications, or other communication channels.

**3.3 Summary**

In summary, the Task Manager application using Django is designed to provide an efficient and user-friendly solution for managing tasks. The system comprises various building blocks, including the user interface, task management module, database management, user authentication and authorization, and notifications and reminders. These components work together to enable users to create, organize, and track tasks, set priorities and due dates, and receive timely reminders. The system aims to streamline task management, improve productivity, and enhance collaboration among users.

**CHAPTER FOUR**

**HARDWARE DESCRIPTION**

The hardware description section provides an overview of the hardware requirements and components needed to run the Task Manager application built using Django. It includes information about the server infrastructure, client devices, and any additional hardware dependencies.

**4.1. Server Infrastructure:**

* Server: The Task Manager application requires a server to host the application and handle incoming requests from clients. The server can be a physical server or a cloud-based server.
* Processor: A suitable processor is required to handle the computational load of the Django application and database operations.
* Memory (RAM): Sufficient RAM is necessary to ensure smooth performance and handle concurrent user requests.
* Storage: Adequate storage capacity is needed to store the application code, database files, and any associated media files.
* Network Interface: The server should have a stable network connection to facilitate communication with client devices.

**4.2. Client Devices:**

* Personal Computers/Laptops: Users can access the Task Manager application using personal computers or laptops. These devices should meet the minimum system requirements for web browsing and running modern web applications.
* Mobile Devices: The Task Manager application may also be accessible through mobile devices such as smartphones and tablets. The application should be responsive and compatible with different screen sizes and operating systems.

**4.3. Additional Hardware Dependencies:**

* Networking Equipment: A reliable network infrastructure is required to ensure smooth communication between the server and client devices. This includes routers, switches, and network cables.
* Backup and Storage Devices: It is recommended to have backup and storage devices in place to ensure data redundancy and prevent data loss in case of hardware failures.

It is important to note that the specific hardware requirements may vary depending on the scale of the Task Manager application and the expected user load. It is advisable to conduct a thorough assessment of the application's requirements and consult with IT professionals or system administrators to determine the most suitable hardware setup.

In the next chapters, the software and system architecture components will be discussed in detail, complementing the hardware description to provide a comprehensive understanding of the Task Manager application.

**4.4. Summary**

The hardware description section provides an overview of the hardware requirements for running the Task Manager application. It includes the server infrastructure needed to host the application, such as the server itself, processor, memory (RAM), storage, and network interface. Client devices, including personal computers/laptops and mobile devices, are also discussed. Additionally, any additional hardware dependencies like networking equipment and backup/storage devices are mentioned. The specific hardware requirements may vary based on the application's scale and user load. Conducting a thorough assessment and consulting with IT professionals is recommended. Overall, this section sets the foundation for understanding the hardware components necessary to support the Task Manager application.

**CHAPTER FIVE**

**SOFTWARE DESCRIPTION**

**5.1 List of Software**

The Task Manager application built using Django relies on various software components to enable its functionality. The following is a list of software used in the development and operation of the application:

1. Django: Django is a high-level Python web framework that provides the foundation for building web applications. It offers a range of features and tools that simplify the development process, including URL routing, template engine, database ORM, and user authentication.

2. Python: Python is a widely-used programming language that serves as the primary language for developing the Task Manager application. It provides a rich ecosystem of libraries and frameworks, making it well-suited for web application development.

3. HTML, CSS, and JavaScript: These are the core web technologies used for creating the user interface and defining the presentation and behavior of the Task Manager application. HTML provides the structure, CSS controls the styling, and JavaScript handles client-side interactivity.

4. SQLite: SQLite is a lightweight and embedded relational database management system used to store and retrieve task-related data in the Task Manager application. It is easy to set up, requires minimal configuration, and integrates seamlessly with Django.

5. Web Browsers: Web browsers such as Chrome, Firefox, Safari, and Edge are essential software for accessing and using the Task Manager application. The application is designed to be compatible with modern web browsers and responsive across different devices.

**5.2 Django**

Django is a powerful and versatile web framework that serves as the backbone of the Task Manager application. It provides a robust set of tools and features that streamline the development process and ensure efficient task management. Key features of Django include:

- URL Routing: Django's URL routing mechanism maps URLs to specific views, allowing for organized and structured handling of incoming requests.

- Template Engine: Django's built-in template engine enables the creation of dynamic and reusable templates, simplifying the generation of HTML content and enhancing the user interface.

- Database ORM: Django's Object-Relational Mapping (ORM) abstracts database interactions, providing a convenient and Pythonic way to interact with the underlying database system, such as SQLite.

- User Authentication and Authorization: Django includes built-in functionality for user authentication and authorization, ensuring secure access to the Task Manager application and protecting user data.

**5.3 Summary**

The Task Manager application relies on a variety of software components to enable its functionality. The list of software includes Django as the primary web framework, Python as the programming language, HTML, CSS, and JavaScript for the user interface, SQLite as the database management system, and web browsers for accessing the application. Django, in particular, offers essential features such as URL routing, a template engine, database ORM, and user authentication and authorization. These software components work together to create a robust and user-friendly task management application.

**CHAPTER SIX**

**RESULTS AND DISCUSSION**

**6.1 General Introduction**

The Results and Discussion section presents the outcomes of implementing the Task Manager application using Django. It provides an overview of the achieved results and discusses their correlation with the identified problem statement. The section also includes a summary of the key findings and insights gained from the results and their implications.

**6.2 Results Correlated with Problem Statement**

The results obtained from the implementation of the Task Manager application address the identified problem statement, which is to provide an efficient and user-friendly solution for task management. The following are the key results achieved:

1. User-Friendly Interface: The Task Manager application offers a user-friendly interface that allows users to easily create, edit, and organize tasks. The interface incorporates intuitive design principles, making it straightforward for users to navigate and interact with the application.

2. Task Management Functionality: The implemented application successfully provides essential task management functionalities. Users can create tasks, set priorities, assign due dates, and track task progress. The application allows for seamless task updates and deletions, providing flexibility in managing tasks.

3. Efficient Task Organization: The Task Manager application enables users to organize tasks efficiently. They can categorize tasks into different lists or categories, set task dependencies, and establish task relationships. This feature enhances task organization and aids in managing complex projects effectively.

4. Notifications and Reminders: The application incorporates a notifications and reminders system, which helps users stay informed about upcoming tasks, deadlines, and important updates. Users receive timely notifications, reducing the chances of missing crucial deadlines and ensuring task completion within the desired time frame.

5. Secure User Authentication: The implemented user authentication system ensures secure access to the Task Manager application. It validates user credentials, protects sensitive data, and prevents unauthorized access, thereby maintaining the confidentiality and integrity of user information.

**6.3 Discussion**

The results obtained from implementing the Task Manager application indicate a successful achievement of the objectives set out in the problem statement. The application provides a user-friendly and efficient solution for task management, addressing the identified challenges. The intuitive user interface simplifies task creation and organization, aiding users in managing their tasks effectively. The notifications and reminders system enhances productivity by keeping users informed about upcoming deadlines and important updates.

The secure user authentication mechanism adds a layer of protection, ensuring that only authorized individuals can access the application and the associated task data. Overall, the Task Manager application successfully addresses the problem statement by streamlining task management processes and improving productivity.

During the implementation and testing phase, several considerations and limitations were identified. These include the need for further optimization to enhance application performance, scalability to handle larger user bases, and potential integration with third-party services for enhanced functionality.

**6.4 Summary**

In summary, the implementation of the Task Manager application using Django has resulted in a successful solution for efficient task management. The application offers a user-friendly interface, essential task management functionalities, efficient task organization, notifications and reminders, and secure user authentication. The achieved results correlate with the identified problem statement, addressing the challenges associated with task management. The discussion highlights the strengths and limitations of the implementation and identifies areas for further improvement.

In the subsequent chapters, the evaluation of the implemented system, user feedback, and recommendations for future enhancements will be discussed, providing a comprehensive analysis of the Task Manager application's performance and potential areas for further development.

**CHAPTER SEVEN**

**BUSINESS PLAN AND COMMERCIAL ASPECT**

**7.1 Market Survey**

A market survey was conducted to assess the potential demand and viability of the Task Manager application in the market. The survey involved gathering information about the target audience, their task management needs, existing solutions available, and the competitive landscape. The key findings from the market survey are as follows:

1. Market Demand: The market survey revealed a growing demand for task management solutions, particularly among individuals and businesses seeking to improve their productivity and organizational efficiency. Task management applications are widely used across various industries and sectors, indicating a substantial market potential.

2. Existing Solutions: The survey identified several existing task management applications available in the market. These applications vary in terms of features, pricing models, and target audiences. However, there is still room for innovative and user-friendly solutions that cater to specific user requirements.

3. Competitor Analysis: The market survey included a thorough analysis of competing task management applications. This analysis focused on their strengths, weaknesses, pricing strategies, user feedback, and market share. Understanding the competition helped identify opportunities for differentiation and market positioning.

**7.2 Feasibility Analysis**

A feasibility analysis was conducted to assess the commercial viability and sustainability of the Task Manager application. The analysis considered various factors, including technical feasibility, market feasibility, financial feasibility, and operational feasibility. The key findings from the feasibility analysis are as follows:

1. Technical Feasibility: The technical feasibility analysis concluded that the Task Manager application is technically feasible to develop and implement using Django and related technologies. The availability of skilled developers, adequate infrastructure, and the compatibility of software components supported the technical feasibility.

2. Market Feasibility: The market feasibility analysis indicated a favorable market environment for the Task Manager application. The growing demand for task management solutions, coupled with the identified gaps in existing applications, suggests a viable market opportunity. Effective marketing strategies and targeted positioning can help capture a significant market share.

3. Financial Feasibility: The financial feasibility analysis involved assessing the investment required for application development, marketing, operations, and potential revenue streams. It considered factors such as development costs, pricing strategies, subscription models, and potential return on investment. The analysis concluded that the Task Manager application has promising financial viability, subject to effective revenue generation and cost management.

4. Operational Feasibility: The operational feasibility analysis evaluated the resources, skills, and processes required to operate and maintain the Task Manager application. It assessed factors such as team composition, infrastructure, customer support, and scalability. The analysis indicated that the application can be efficiently operated and scaled to accommodate growing user demands.

**7.3 Summary**

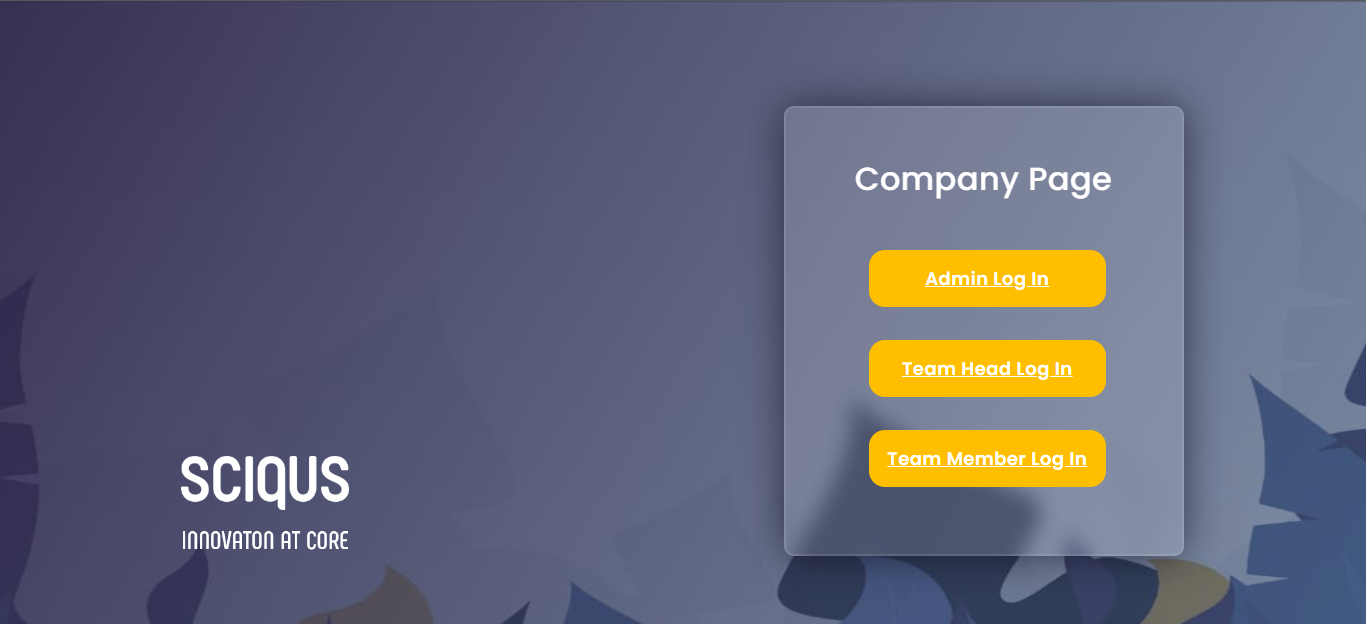
The business plan and commercial aspect of the Task Manager application consider the market survey findings and feasibility analysis. The market survey revealed a favorable market demand for task management solutions, with opportunities for differentiation and positioning. The feasibility analysis demonstrated the technical, market, financial, and operational feasibility of the application.

Based on the findings, the Task Manager application holds promising commercial potential. Effective marketing strategies, continuous product enhancements, and attentive customer support will be crucial for capturing and retaining a significant market share. The subsequent chapters will discuss the marketing and revenue generation strategies, as well as the operational aspects, to ensure the successful launch and sustainable growth of the Task Manager application.

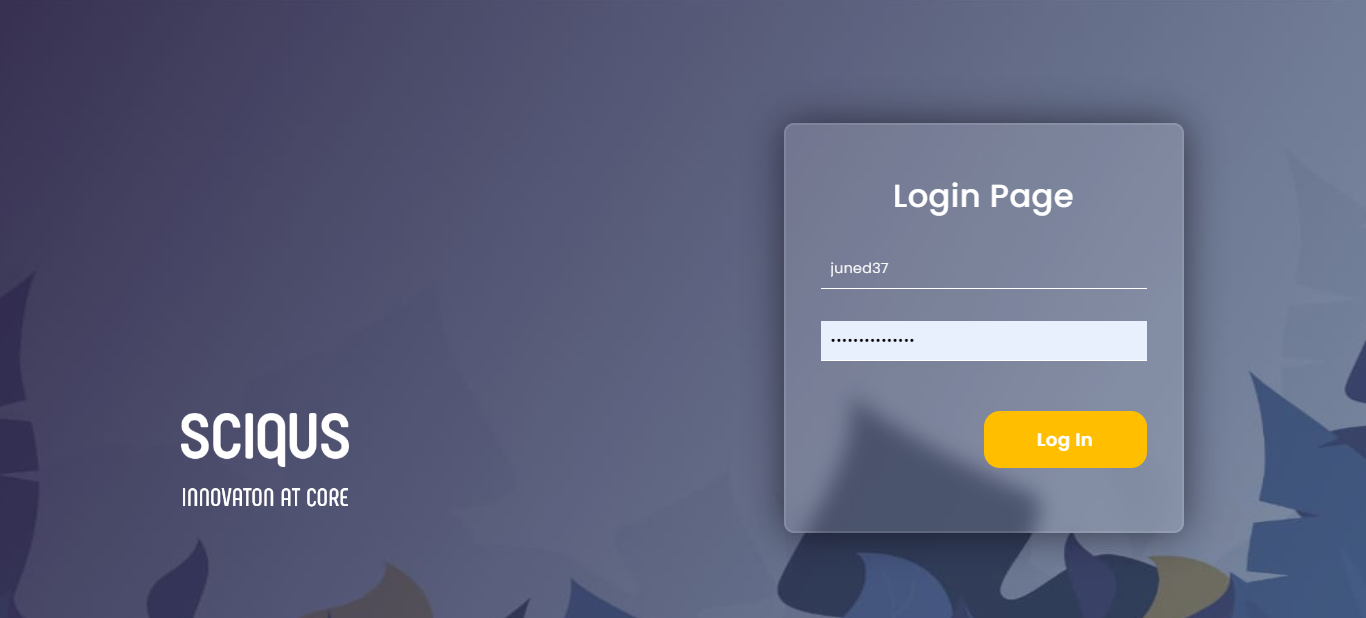
**CHAPTER EIGHT**

**CONCLUSION**

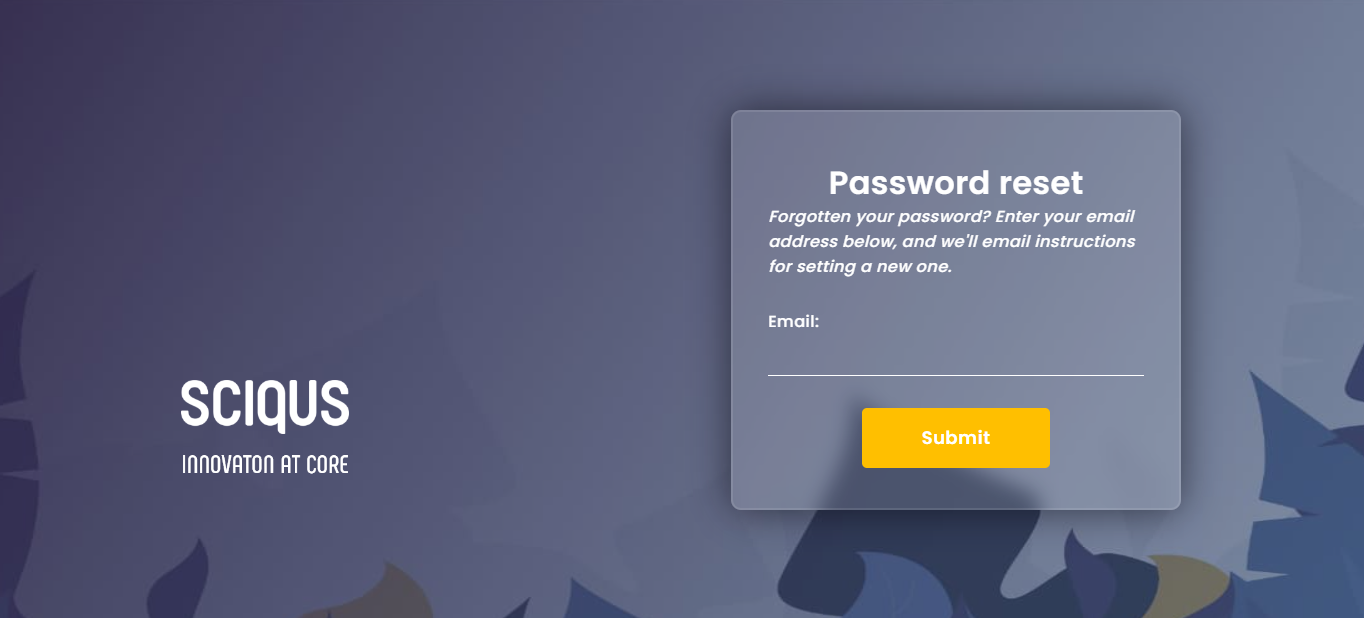
**8.1 Code , Implementation And Testing**



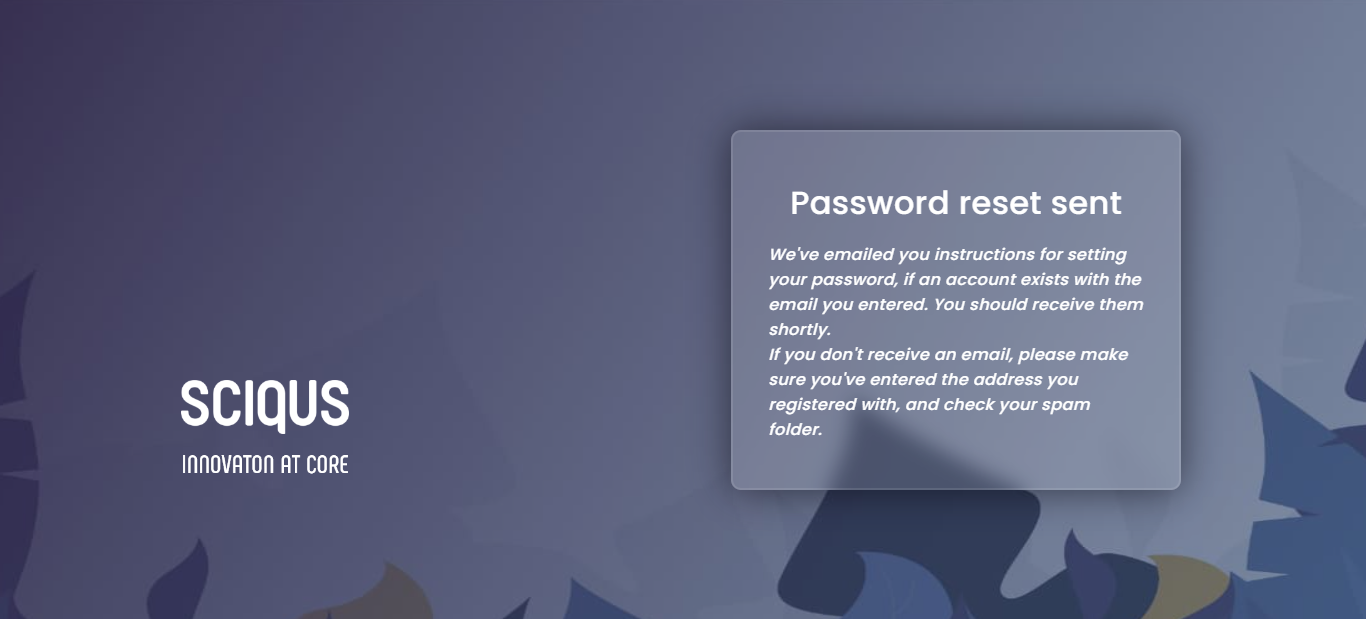
**FIG 8.1 HOME\_PAGE**



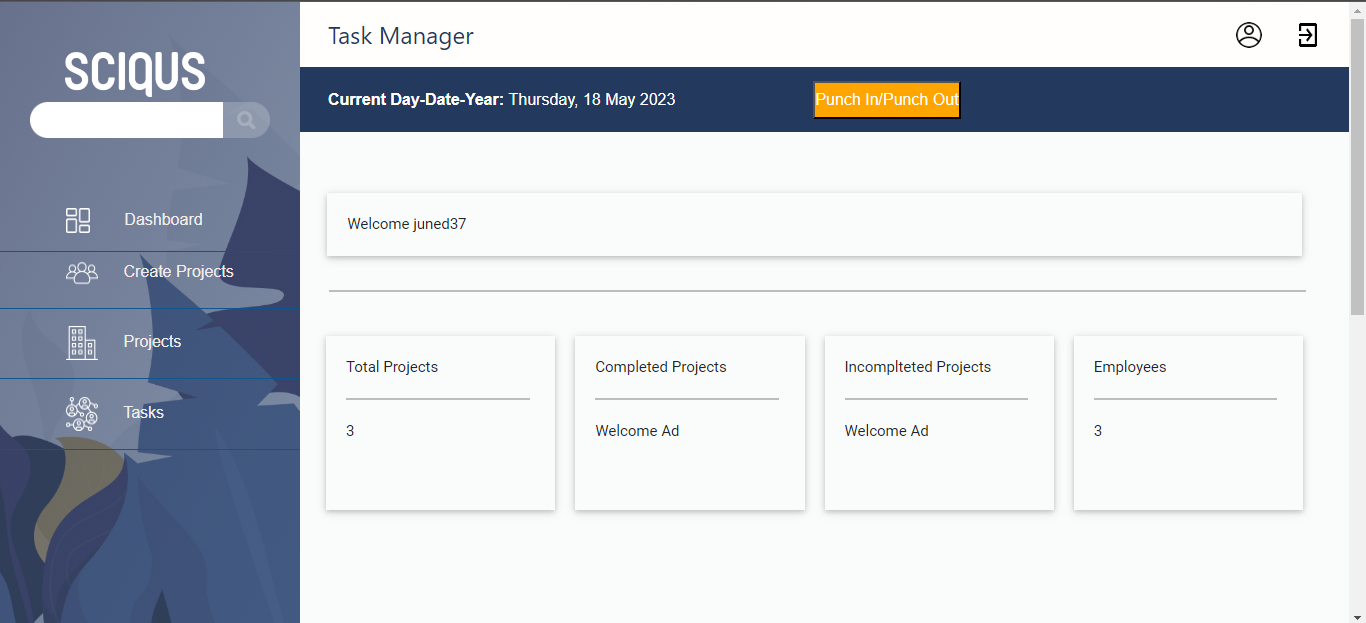
**FIG 8.2 LOGIN PAGE**



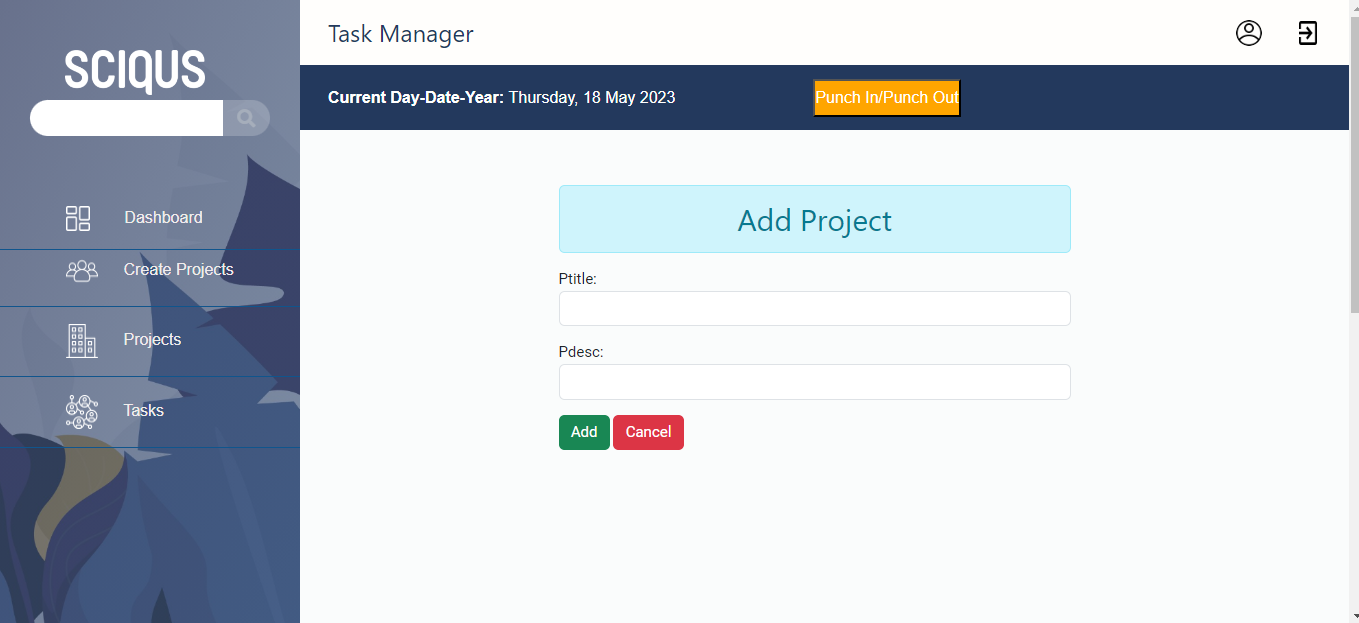
**FIG 8.3 FORGOT\_PASSWORD**



**FIG 8.4 RESET\_PAGE**



**FIG 8.5 DASHBOARD**



**FIG 8.6 CREATE\_PROJECT**

**8.1.1 Test Case**

|  |  |
| --- | --- |
| TEST ID | 01 |
| Test Case Summary | User Registration Testing |
| Related Requirement | 1.Register valid data to allfields  2.Check the data and register Test Procedure |
| Test Procedure | Register data send to Database |
| Test Data | If any invalid Data or not filled , it will be resending touser |
| Expected Result | After all Verification and Validation , can login to their account |
| Actual Result | After all Verification and Validation , can login to their account |
| Status | **Success** |
| Test Environment | Windows/Linux |
| Test Case | **PASS** |

**TABLE 8.1.1 TEST CASE**

**8.2 Applications**

The Task Manager application developed using Django offers a comprehensive and user-friendly solution for efficient task management. It caters to the needs of individuals and businesses across various industries who seek to improve their productivity and organizational efficiency. The application's key features, including task creation, prioritization, organization, notifications, and user authentication, make it suitable for a wide range of applications, such as:

1. Personal Task Management: Individuals can utilize the Task Manager application to effectively manage their personal tasks, to-do lists, and deadlines. It helps them stay organized, track progress, and ensure timely completion of tasks.

2. Project Management: The application can be employed for project management purposes, allowing teams to collaborate, assign tasks, monitor progress, and meet project deadlines. It facilitates efficient communication and coordination among team members.

3. Business Task Management: Businesses of all sizes can benefit from the Task Manager application to streamline their task management processes. It enhances workflow efficiency, reduces errors, and promotes effective task delegation.

**8.3 Advantages and Disadvantages**

Advantages:

- User-friendly interface that simplifies task management processes.

- Essential features such as task creation, prioritization, and organization.

- Notifications and reminders system to stay informed about upcoming tasks and deadlines.

- Secure user authentication to protect sensitive data.

- Scalability to accommodate growing user bases and increasing task volumes.

- Compatibility with multiple devices and web browsers for easy access.

Disadvantages:

- Potential performance limitations when dealing with a large number of tasks or concurrent users.

- The need for continuous updates and enhancements to meet evolving user requirements.

- Reliance on internet connectivity for accessing the application.

**8.3 Future Scope**

The Task Manager application has significant potential for future enhancements and expansions. Some areas for future scope and improvement include:

1. Integration with Third-Party Services: Integration with popular productivity tools, calendar applications, or project management platforms can enhance the application's functionality and provide users with seamless synchronization of tasks and deadlines.

2. Mobile Application Development: Developing a dedicated mobile application for the Task Manager can enhance accessibility and convenience, allowing users to manage tasks on the go.

3. Data Analytics and Insights: Implementing data analytics capabilities can provide users with valuable insights into their task management patterns, productivity trends, and areas for improvement.

4. Collaboration and Team Management Features: Introducing advanced collaboration features, such as task assignment, real-time updates, and team communication tools, can cater to the needs of larger teams and complex projects.

**8.4 Conclusion**

In conclusion, the Task Manager application developed using Django offers a user-friendly and efficient solution for task management. The application addresses the identified problem statement by providing essential features, intuitive interface, and secure authentication. The results and discussion demonstrate the successful achievement of the objectives, as the application streamlines task management processes and improves productivity.

While the application offers several advantages, it is important to consider its limitations and scope for future enhancements. By continuously refining and expanding its features, integrating with third-party services, and addressing user feedback, the Task Manager application can establish itself as a reliable and versatile tool for individuals and businesses seeking effective task management solutions.