# TASK MANAGEMENT WEB APPLICATION

Project submitted to the

SRM University – AP, Andhra Pradesh

for the partial fulfilment of the requirements to award the degree of

**Bachelor of Technology**

In

**Computer Science and Engineering**

**School of Engineering and Sciences**

Submitted by

**ASHOK ARYAL ANKSUSH KUMAR ADHIKARI**

**(AP23110011540) (AP23110011672)**

**MOHAMED YOUSSOUF KEITA JANGALA SAI ADITYA**

**(AP23110011680) (AP23110011619)**

****

Under the Guidance of

**Ms. K. Kavitha Rani**

**Lecturer, Department of CSE**

**SRM University–AP**

**Neerukonda, Mangalagiri, Guntur**

**Andhra Pradesh – 522 240**

**[April, 2025]**

**Certificate**

Date: \_\_\_\_\_\_\_\_\_

This is to certify that the work present in this Project entitled “**TodoApp**” has been carried out by **Ashok Aryal, Ankush Kumar Adhikari, Mohamed Youssouf Keita, Jangala Sai Aditya** under my/our supervision. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor of Technology/Master of Technology in **School of Engineering and Sciences**.

**Supervisor**

(Signature)

Prof. K. Kavitha Rani

Designation,

Affiliation.

i

**Acknowledgement**

The satisfaction that accompanies the successful completion of any task would be incomplete without introducing the people who made it possible and whose constant guidance and encouragement crowns all efforts with success.

I am extremely grateful and express my profound gratitude and indebtedness to my project guide, **Ms. K. Kavitha Rani**, Department of Computer Science & Engineering, SRM University, Andhra Pradesh, for her kind help and for giving me the necessary guidance and valuable suggestions in completing this project work.

ASHOK ARYAL(AP2311001540)

ANKUSH KUMAR ADHIKARI(AP23110011672)

MOHAMED YOUSSOUF KEITA(AP23110011680)

JANGALA SAI ADITYA(AP23110011619)

**ii**

**Table of Contents**

• **Certificate**……………………………………………………………………………….……… i

• **Acknowledgements**………………………………………………………..…………………... ii

• **Table of Contents**……………………………………………………………………………...iii

• **Abstract**…………………………………………….………………………………………… vii

**1. Introduction**…………………………....………………………......…………………………..1

1.1 Objectives.……….…………………………………………………………………… 1

1.2 Technology Stack ………………………….………………………………………… 1

1.3 System Architecture…………….…….……………………………………………… 1

**2. Methodology**………………………………………………………………………………….. 3

  2.1 Key Functionalities.…..……………………………………………………………… 3

  2.2 User Interface Design………….…………..……………….………………………… 3

**3. Implementation** ……...………………………………………………………………………. 5

3.1 Key Functionalities.…..……………………………………………………………… 3

3.2 User Interface Design...……………….……………………………………………… 3

3.3 Security and Configuration...…………….…………………………………………… 3

3.4 API Integration……………….....…………….…………………………………………… 3

3.5 Testing and Validation…….....…………….…………………………………………… 3

**4. Result and Analysis** ……………………….…………………………………………………. 7

**5. Conclusion** ………………..………………………………………………………………….. 9

**6. Future Enhancements**…….………………………………………………………………… 10

**7. References** …..………………………………………………………………………………. 11

**iii**

**Abstract**

This project report presents the development of "TodoApp", a web-based task management system built using Flask (Python), Jinja2, Bootstrap, HTML, and CSS. The application enables users to add, update, delete, and monitor tasks, while also offering analytics and RESTful API endpoints for extended functionality. This report elaborates on the application’s design, implementation, and future scope, providing a complete overview of the development process and the final product.

**Introduction**

In today’s fast-paced world, managing time and tasks efficiently is vital. With increasing responsibilities, a tool that helps users organize and manage their day-to-day activities is more necessary than ever. TodoApp is a lightweight, responsive, and feature-rich task manager that simplifies task organization and boosts productivity. It is built with simplicity in mind but is powerful enough to support filtering, statistics, and REST APIs.

**Objectives**

The main goals of the TodoApp are:

* To provide an intuitive and easy-to-use interface for managing tasks.
* To allow users to add, delete, and mark tasks as complete or incomplete.
* To display real-time analytics on task status.
* To provide RESTful APIs for integration with other tools or automation scripts.
* To ensure the design is responsive and visually appealing across devices.

**Technology Stack**

* Backend: Python using Flask microframework
* Templating Engine: Jinja2
* Frontend: HTML5, CSS3, and Bootstrap 5
* Database: SQLite via SQL Alchemy ORM
* Configuration: env and config.py for environment and app settings

**System Architecture**

TodoApp is structured in a clean, modular MVC-like architecture:

* Models: Handled using SQL Alchemy ORM to define the schema for tasks.
* Views: Handled using Jinja2 templating for dynamic rendering of HTML pages.
* Controllers (Routes): Managed via Flask route decorators that process user input and API calls.
* APIs: RESTful endpoints allow external services to interact with tasks (e.g., create, update, get stats).

**Key Functionalities**

Task Management

* Add new tasks with title and priority.
* Mark tasks as completed.
* Delete completed or all tasks.

Filtering and Sorting

* Filter tasks based on completion status.
* View all, pending, or completed tasks.

Analytics Dashboard

* Visual progress bar showing the percentage of tasks completed.
* Textual and graphical insights into task distribution.

REST API Endpoints

* /Api/todos: Get all tasks.
* /Api/todos/<id>/complete: Mark task complete.
* /Api/todos/filter: Filter tasks.
* /Api/stats: Get task completion stats.

**User Interface Design**

The front-end design is clean, modern, and user-centric. Leveraging Bootstrap 5, the layout adjusts dynamically to different screen sizes. Key features include:

* Clear call-to-action buttons.
* Visual progress indicators.
* Priority-based styling using custom CSS classes from enhanced-features.css.
* Hover effects and accessibility improvements.

****

**Security and Configuration**

Security is considered by separating configuration values from the codebase using .env files and config.py. This includes:

* Secret keys for session security.
* Debug mode control.
* Dynamic database URI setup for development and deployment environments.

**API Integration**

TodoApp exposes a RESTful API for external integration. These endpoints enable:

* Task creation via external clients.
* Status checking and updates.
* Retrieval of statistical data (task completion rates).

The endpoints are JSON-compatible and designed for ease of use with frontend frameworks or third-party tools like Postman.

**Testing and Validation**

Manual testing was conducted via:

* Browser-based testing for UI flows and visual accuracy.
* Postman for RESTful API testing.
* Form validation using HTML attributes and Python back-end logic.
* Error handling for bad inputs and task id mismatches.

**Result And Analysis**

The project achieved all its major objectives:

* Responsive UI for task management.
* Working REST API for CRUD and statistics.
* Dynamic statistics and filtering.
* Modular and clean code architecture for easy expansion.

**Future Enhancements**

To expand the application’s capabilities, the following features can be introduced:

* User authentication for personalized task management.
* Task editing and due date reminders.
* Email/SMS notifications for upcoming deadlines.
* Mobile-first redesign using modern JavaScript frameworks like React or Vue.js.
* Dark mode toggle for better user experience.

**Conclusion**

TodoApp proves that even a lightweight web application can deliver professional, meaningful, and extensible task management. It incorporates essential software development principles like modularity, API design, responsive UIs, and secure configurations. The app is a solid foundation that can be scaled for real-world deployment or integrated into larger systems.

**References**

* **Flask Documentation**
* [**SQL Alchemy Documentation**](https://www.sqlalchemy.org/)
* [**Bootstrap 5 Documentation**](https://getbootstrap.com/)
* [**Python-dotenv**](https://pypi.org/project/python-dotenv/)
* **Jinja2 Templating Engine**