A person touching a computer screen

Description automatically generated

**Project Name Submission Date**

**Prepared by**

Task Management

12/1/2024

Krishna Sai Desaboina

FINAL REPORT

## Abstract

This project proposal summarizes the development of a web-based Task Tracker System designed to make task management easy. The system will be able to create, update, and delete tasks. Besides that, users can set priorities, attach deadlines, and monitor task statuses, whether pending, in progress, or complete. The user interface will be designed using Python's Streamlit Library for the front end, while SQL will act as the back-end database to store details of the tasks. The project includes front-end state management and backend database interactions. In the future, this may be enhanced by user authentication, advanced filtering, sorting options, and notifications for upcoming task deadlines. This tool will help users stay organized and manage their tasks effectively.

## Project Design

Technologies

The programming languages used are Python for the development of the frontend part of this project.

DBMS: MySQL will be used for maintaining the Tasks and Project team data.

Web Development: Python's Streamlit Library has been used for the construction of the web-based interface.

Version Control: To maintain source code, to work in teams, Git version control systems were used.

IDEs / Text Editors: Used for coding and debugging of the front-end and MySQL Workbench for the Database management System.

Design an intuitive user interface where users can interface with the system. It is about saving user inputs, showing data, and guiding with easy navigation. Confirm user's input for integrity and accuracy of the data. Perform checks on the existence of fields, data format, and length to prevent improper data from being stored in the database. You should open the connection with the database management system and include query and transaction methods to persist and retrieve the information on the database. Create the method of creating, updating, and gathering records from relevant tables.

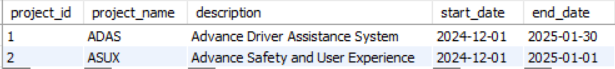
## Design method or functions:

Task Manager Database is created in MySQL Database and The Employee Details are filled accordingly with employee's personal information (user\_id, name, email, password).

A screenshot of a computer

Description automatically generated

The Project Details are filled in accordingly with Project Details (project\_id, project name, description, start\_date, end\_date)



The Team details are filled accordingly with Team details like (team\_id, team\_name)

A close-up of a screen

Description automatically generated etc...

Built a UI app with the Streamlit package of Python. The UI consists of three pages: Create, Update, Delete, Status.

**Create:**

This page is used to create tasks. The details entered on this page are Inserted into the Tasks table.

**Update:**

This page is used to Update tasks. The details entered on this page are updated in the Tasks table. If the given ID does not exist in the database then, it returns an error.

**Delete:**

This page is used to delete the task in the task table using the task\_ID. If the given ID does not exist in the database then, it returns an error.

**Status:**

This page is used to see the status of the task using the multiple joins of the tables: Tasks, Task\_Statuses, Task\_Priority, Projects, Team\_Projects, Teams, Task\_Assignees, Users.

### Sample Output:

### (13, 'a', 'a', datetime.date(2024, 12, 27), 'Completed', 'Low', 'ASUX', 'Testing', 'ram', '[ram@gmail.com](mailto:ram@gmail.com)')

## **Potential Tables:**

1. Users table: This table stores user details and is also related to tasks and comments, as users can comment on tasks.
2. Teams table: This represents a group of users linked to projects through the as relationship between teams and projects.
3. Projects table: It contains project details and is related to tasks.
4. Projects table: This has project details and it is linked with tasks.
5. Task Assignees table: Connects the tasks and users as multiple users can be assigned to multiple tasks.
6. Task Statuses table: This table stores possible statuses for tasks like "Pending", "In Progress", "Completed" etc.
7. Task Priorities table: It stores possible priorities for tasks with "Low", "Medium", "High" priority.
8. Comments table: This table stores user comments on tasks and then links users to tasks through user\_id and task\_id.
9. Team Projects table: It relates teams and projects.

## **Assumptions:**

The system is built on a centralized company’s administration configuration with a single database and a web-based interface that is only accessible to users with permission.

Security mechanisms like authentication, authorization, and data encryption will be implemented to ensure data privacy and protection.

The project will assume routine data backups and a disaster recovery plan to prevent data loss.

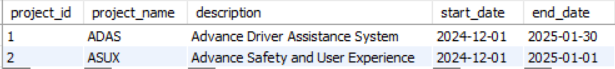
In order to gather requirements and confirm system functionality, the project implies collaboration with stakeholders and Higher-level professionals.

## **Project Dataset:**

The employee's personal information (user\_id, name, email, password), Project Details (project\_id, project name, description, start\_date, end\_date), Team details like (team\_id, team\_name), etc are all included in this dataset.

A screenshot of a computer

Description automatically generated



A close-up of a screen

Description automatically generated



**Introduction:**

Nowadays, times are tough and

individuals and companies to manage everything and keep being productive is no longer possible without usually proper operation. In other words, it can be a guide in handling personal tasks, meeting deadlines, and successfully working with teams that lead us to greater time management.

Whether it's managing personal tasks, reaching the deadline, or working with colleagues on projects, to make sure systems run smoothly, a mechanism to organize tasks properly can significantly influence time management and performances as well.

A Task Management System can be described as a tool that offers a means of adding, organizing, tracking, and sorting tasks according to the user's requirements. This system, as such, provides a simplified way of assigning tasks, evaluating the progress of a project, and keeping track of due dates, which empower users with the capabilities to concentrate on the most important ones.

In this project, the development of a Task Management System

based on SQL

(Structured Query Language) for data storage purposes and Streamlit

for the creation of an interactive, web-based user interface will be carried out. SQL functions as a basic store and access mechanism for task data, while Streamlit gives a really quick and user-friendly way to build and deploy the application.

The system permits operations such as definite task addition, update, task deletion, management of task status, priority, and deadlines, which are the basics that every user can have through. Besides, with solid SQL database features and straightforward Streamlit development, the Task Management System is thus a scalable and user-friendly tool for personal uses and team collaborations, alike, powered by.

The objective of this project is to bring an uncomplicated and seamless user experience with the system, which means simplicity and convenience and a smooth performance of the work to be done are crucial. This introduction of the Task Management System provides an initial foundation for understanding how combining SQL and Streamlit can with to an efficient and interactive tool that manages daily tasks and long projects set up efficiently.

# Challenges:

Some of the issues have been such as insufficient and frequent or outdated information shared, as well as the difficulty in the coordination that results from the relationships of tasks. Some of them use old techniques that are dull, even involving hands-on approach and that means that they are not as efficient in big groups. Also, the interfaces may be confusing or cluttered leading to improper utilization. Another issue is privacy because it is possible that private data is not well safeguarded. The multi-faceted approaches are helpful when attempting to solve the big picture because when solving the problems in a location where people have the facility and knowledge to perform the tasks, they are located.

# Motivation:

Some Task Management features that have been created

* + Task Scheduling
  + Deadlines for Every Task
  + Updating of Task details
  + Tacking the Status of Every Task
  + User Friendly UI

# Tables Created:

1. Users table: This table stores user details and is also related to tasks and comments, as users can comment on tasks.
2. Teams table: This represents a group of users linked to projects through the as relationship between teams and projects.
3. Projects table: It contains project details and is related to tasks.
4. Projects table: This has project details and it is linked with tasks.
5. Task Assignees table: Connects the tasks and users as multiple users can be assigned to multiple tasks.
6. Task Statuses table: This table stores possible statuses for tasks like "Pending", "In Progress", "Completed" etc.
7. Task Priorities table: It stores possible priorities for tasks with "Low", "Medium", "High" priority.
8. Comments table: This table stores user comments on tasks and then links users to tasks through user\_id and task\_id.
9. Team Projects table: It relates teams and projects.

# ER-Diagram:

We have created ER Diagram (Class activity diagram) using Lucid Chart as shown below

****

**Work Flow:**

A diagram of a process

Description automatically generated

**Create:**

This page is used to create tasks. The details entered on this page are Inserted into the Tasks table.

**Update:**

This page is used to Update tasks. The details entered on this page are updated in the Tasks table. If the given ID does not exist in the database then, it returns an error.

**Delete:**

This page is used to delete the task in the task table using the task\_ID. If the given ID does not exist in the database then, it returns an error.

**Status:**

This page is used to see the status of the task using the multiple joins of the tables: Tasks, Task\_Statuses, Task\_Priority, Projects, Team\_Projects, Teams, Task\_Assignees, Users.

### Sample Output:

### (13, 'a', 'a', datetime.date(2024, 12, 27), 'Completed', 'Low', 'ASUX', 'Testing', 'ram', '[ram@gmail.com](mailto:ram@gmail.com)')

# Tech Stack Model:

A close-up of a logo

Description automatically generated

**ScreenShots:**

A screenshot of a computer

Description automatically generated

## **Task Creation:**

## A screenshot of a computer Description automatically generated

## **Task Update:**

## A screenshot of a computer Description automatically generated

## **Task Status:**

## A screenshot of a computer Description automatically generated

## **Delete Task:**

## A screenshot of a computer Description automatically generated

## **Task Status:**

## A screenshot of a computer Description automatically generated

**Conclusion:**

A traditional Task Management System built using SQL in the back end and Streamlit as the front end is the one explained above. MySQL and Streamlit if you want your Task Management System to have a strong database you use MySQL for writing your SQL scripts. If you wish to develop interactive web apps you make use of Streamlit. You can fix a task management system with primary operations like task addition, update, and deletion, progress monitoring, and deadline supervision very quickly.