Task Management System

**Submitted By-**

**Vivek Raj (2215002003)**

**Yash Kumar Agrawal (2215002023)**

**Yashica Bhardwaj (2215002043)**

**Overview**

A Task Management System (TMS) is an application designed to help users organize and manage their tasks efficiently. This project will utilize C++ for the application logic and MySQL for data storage.

**Objective**

The primary objective of this project is to develop a robust task management system that allows users to:

* Create, update, and delete tasks.
* Categorize tasks.
* Set deadlines and priorities.
* View tasks in different formats (e.g., by date, priority).
* Track task completion status**.**

**Features**

1. **User Authentication:**

* Register and log in users.
* Password encryption for secure storage.

1. **Task Management:**

* **Create Tasks:** Add new tasks with descriptions, deadlines, priorities, and categories.
* **Update Tasks:** Modify task details, such as status, deadlines, and priority.
* **Delete Tasks:** Remove tasks that are no longer needed.
* **View Tasks:** Display tasks in a list or calendar format, with filtering options (e.g., by date, priority, category).

1. **Task Categorization:**

* Assign categories or tags to tasks for better organization.

1. **Deadline and Priority Management:**

* Set deadlines and priorities for tasks.
* View tasks sorted by deadlines or priorities.

1. **Search and Filter:**

* Search tasks by keywords.
* Filter tasks based on different criteria (e.g., status, date range).

1. **Reporting:**

* Generate reports or summaries of tasks based on various criteria (e.g., completed tasks, overdue tasks).

Working

1. **Database Design:**
   * **Tables:**
     + Users: Store user information (user\_id, username, password\_hash).
     + Tasks: Store task details (task\_id, user\_id, description, deadline, priority, category, status).
     + Categories: Store task categories (category\_id, name).
   * **Relationships:**
     + One-to-many relationship between Users and Tasks.
     + One-to-many relationship between Categories and Tasks.
2. **C++ Application:**
   * **Database Connectivity:**
     + Use MySQL Connector/C++ to interface with the MySQL database.
     + Perform CRUD operations (Create, Read, Update, Delete) on the database.
   * **User Interface:**
     + Command-line interface (CLI) or a simple graphical user interface (GUI) using libraries such as Qt or wxWidgets.
     + Provide options for user login, task management, and report generation.
   * **Business Logic:**
     + Implement functions for task creation, updating, deletion, and retrieval.
     + Handle user authentication and authorization.
     + Implement search and filter functionalities.
3. **Interaction Flow:**
   * Users log in to the system.
   * They can create, view, update, or delete tasks.
   * Tasks are stored in the MySQL database.
   * Users can generate reports or search/filter tasks based on their needs.

**Key Points**

1. **Database Design:**
   * Ensure normalization to avoid data redundancy.
   * Design efficient queries to handle large datasets.
2. **Security:**
   * Use hashing algorithms for storing passwords securely.
   * Implement proper authentication and authorization checks.
3. **Error Handling:**
   * Implement robust error handling for database operations and user inputs.
4. **Performance:**
   * Optimize database queries and C++ code to handle large numbers of tasks efficiently.
5. **User Experience:**
   * Design an intuitive user interface.
   * Provide clear feedback and error messages to the user.
6. **Testing:**Perform thorough testing, including unit tests for individual components and integration tests for the overall system.