Python - Aptpath Internship Batch 2

Project Title: Task Manager Documentation

Project Statement and Outcomes:

The objective of this project is to develop a Task Manager application to help users organize, track, and manage their tasks efficiently. The software will offer an intuitive interface for users to add, view, update, and delete tasks. Users can categorize tasks into different project groups, set due dates, assign priority levels, and filter tasks based on various parameters. The main aim is to streamline task management, enabling users to achieve better productivity and efficiency in handling personal and professional tasks.

The outcome of this project will be a fully functional web application where users can create, edit, categorize, and manage tasks. The system will feature user authentication, ensuring secure access for multiple users. It will also provide a responsive interface built with Streamlit, enabling users to interact with the system on various devices.

Modules to be Implemented:

- User Authentication Module
- Task Creation and Management Module
- Task Categorization and Filtering Module
- Database and Data Storage Module
- UI and User Interaction Module
- Review, Bug Fixes, Documentation

Week-wise Module Implementation and High-level Requirements:

Week 1:

Module Implementation - User Authentication and Task Creation & Management

High-level Requirements:

- **Setup Development Environment:**
  - Install necessary libraries such as Streamlit, Pandas, SQLite, Bcrypt, and SessionState.
- **User Authentication:**
  - Implement a secure registration system allowing users to create an account.
  - Enable user login with secure password hashing using Bcrypt.
  - Implement session management so users stay logged in during their session.
- **Task Creation & Management:**
  - Create a form for adding new tasks with the following fields:
    - Task Name (mandatory)
    - Description
    - Priority (Low, Medium, High)
    - Deadline (date picker)
    - Status (Pending, In Progress, Completed)
    - Project (to categorize tasks into specific projects)
  - Once the user submits the form, save the task data to the database.
  - Build the UI to display all tasks in a tabular format using Streamlit's st.dataframe() for visualization.
- **Task Editing and Deletion:**
  - Allow users to edit task details directly from the task table.
  - Provide an option for users to delete tasks that are no longer needed.

- **Database and Storage:**

    - Use SQLite as the primary database for storing user and task data.
- **Create a table named tasks with the following structure:**
  CREATE TABLE tasks (id INTEGER PRIMARY KEY AUTOINCREMENT, task_name TEXT NOT NULL,description TEXT, priority TEXT, deadline DATE, status TEXT, project TEXT);
    - Store user data securely with hashed passwords for security.
- **UI Design:**
    - Design a user-friendly interface with navigation options for login, task management, and filtering.
    - Use Streamlit's sidebar for filtering options and task categorization.
- **Error Handling:**
    - Implement proper error messages for user authentication failures.
    - Handle input validation for task creation and editing to prevent submission of invalid data.

Week 2:

Module Implementation - Streamlit Documentation

High-level Requirements:

- **Comprehensive Documentation:**
    - Write comprehensive documentation for all modules and functionalities.
    - Explain user workflows for registration, login, task creation, editing, deletion, and filtering.
- **Technical Specifications:**
    - Provide technical details about the system's architecture, libraries used, and key configurations.
- **System Diagrams:**
    - Include diagrams such as flowcharts to represent system workflows and module interactions.
- **Code Documentation:**
    - Add in-line comments to the source code for better readability and understanding.
- **User Guide:**

    - Provide a user guide with step-by-step instructions on how to use the system, including screenshots where applicable.

Output:

- A login and registration system with secure password handling.
- Ability for users to add, view, edit, and delete tasks.
- A fully functional database for storing tasks and user information.
- A user-friendly interface with interactive components for task filtering and status updates.
- Comprehensive documentation for the system, including user guides and technical specifications.