SOFTWARE REQUIREMENT SPECIFICATION FOR TASK DASHBOARD

Name	Jeevika S
Roll Number	7376222AD147
Seat Number	11
Project ID	7
Project Title	Task Dashboard
Stack	Python Stack

STACK COMPONENTS

Frontend	○ HTML ○ CSS ○ JS
Backend	○ Flask ○ Django
Database	○ SQLite ○ MySQL ○ PostgreSQL
Task Scheduling	○ APScheduler

Authentication & Authorization	○ Flask-login ○ Django Auth System
Deployment	○ Cloud platform

PROBLEM STATEMENT

Organizations and groups frequently battle with monitoring assignments, cutoff times, and undertaking status. The absence of a unified and outwardly natural dashboard can prompt missed cutoff times, wasteful undertaking prioritization, and trouble in project the executives. The Undertaking Dashboard means to address these difficulties by giving a far-reaching answer for task checking and detailing.

INTRODUCTION

The Task Dashboard is an electronic application that gives an incorporated stage to task the board and checking. It takes care of both regulatory and client level access, considering proficient following and the board of errands inside an association or venture.

PROJECT SCOPE

A feature-rich web application developed with the Python stack is the Task Dashboard. A User Dashboard that enables users to monitor the progress of activities they have been allocated and an Admin Dashboard for managing users, projects, and task assignments are among its primary features. Task creation, assignment, and smooth task progress tracking from pending to done are all made possible by this application. Strong analytics and reporting features offer information on team output, job completion rates, and project status. With the use of databases, deployment techniques, and Python frameworks, Task Dashboard provides an effective and scalable task management solution that helps businesses improve workflows and teamwork.

PROJECT OVERVIEW

A web application running on Python, the Task Dashboard offers a centralised platform for task administration and monitoring. The backend functionality, database integration, and RESTful API development will be handled by the project using a Python web framework, such as Flask or Django. To create a user experience that is easy to use, the frontend will be constructed with HTML, CSS, and JavaScript and will incorporate a contemporary framework such as React, Angular, or Vue.js. Important functionalities of the programme will include an Admin Dashboard for managing users and projects, a User Dashboard for task tracking, assignment and creation of tasks, progress updates, and extensive reporting and analytics capabilities. Docker containers will be used in the deployment to guarantee scalable and consistent delivery across many environments.

PROJECT FLOW

Conditions Collecting and Analysing:

To comprehend the intended features and functions of the Task Dashboard, collect requirements from stakeholders, including administrators and ordinary users.

Database Design:

Create the database design needed to hold task-specific data, such as task specifications, user data, and task status.

Backend Development:

Create the database design needed to hold task-specific data, such as task specifications, user data, and task status.

Backend Development:

Use the web framework of choice (Flask or Django) to implement the backend, manage CRUD (Create, Read, Update, Delete) tasks, and interface with the database.

Frontend Development:

Using HTML, CSS, and JavaScript, create the user interface. Then, using the frontend framework of your choice, integrate it with the backend.

Authentication and Authorization:

To guarantee safe dashboard access, put user authentication and authorization procedures in place.

Task Scheduling:

Include the ability to schedule tasks so that they are automatically updated, and so that notifications are sent when tasks are past due.

Reporting and Analytics:

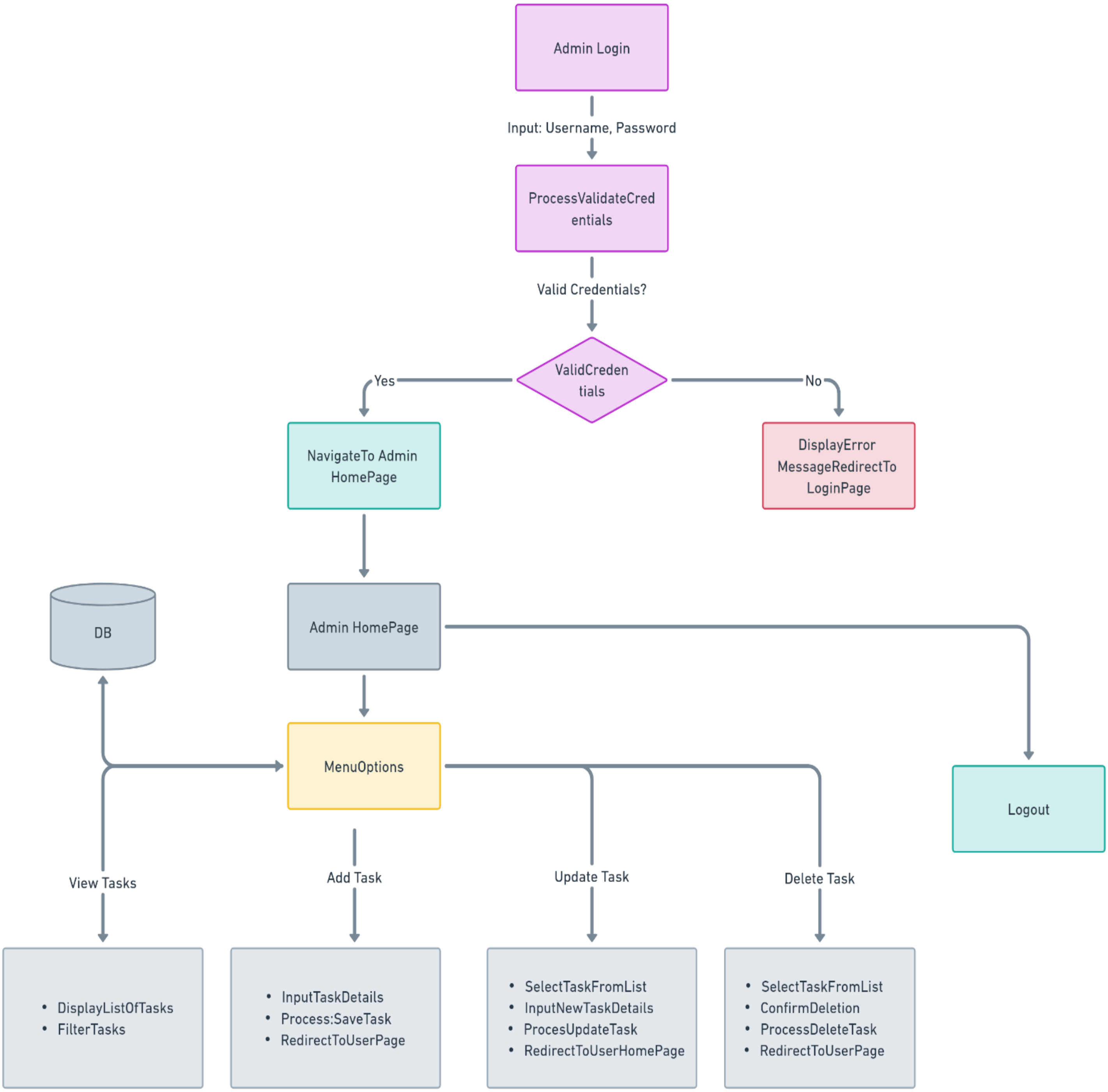
Task management and team productivity can be gained by administrators through the implementation of reporting and analytics services.

Testing and Deployment:

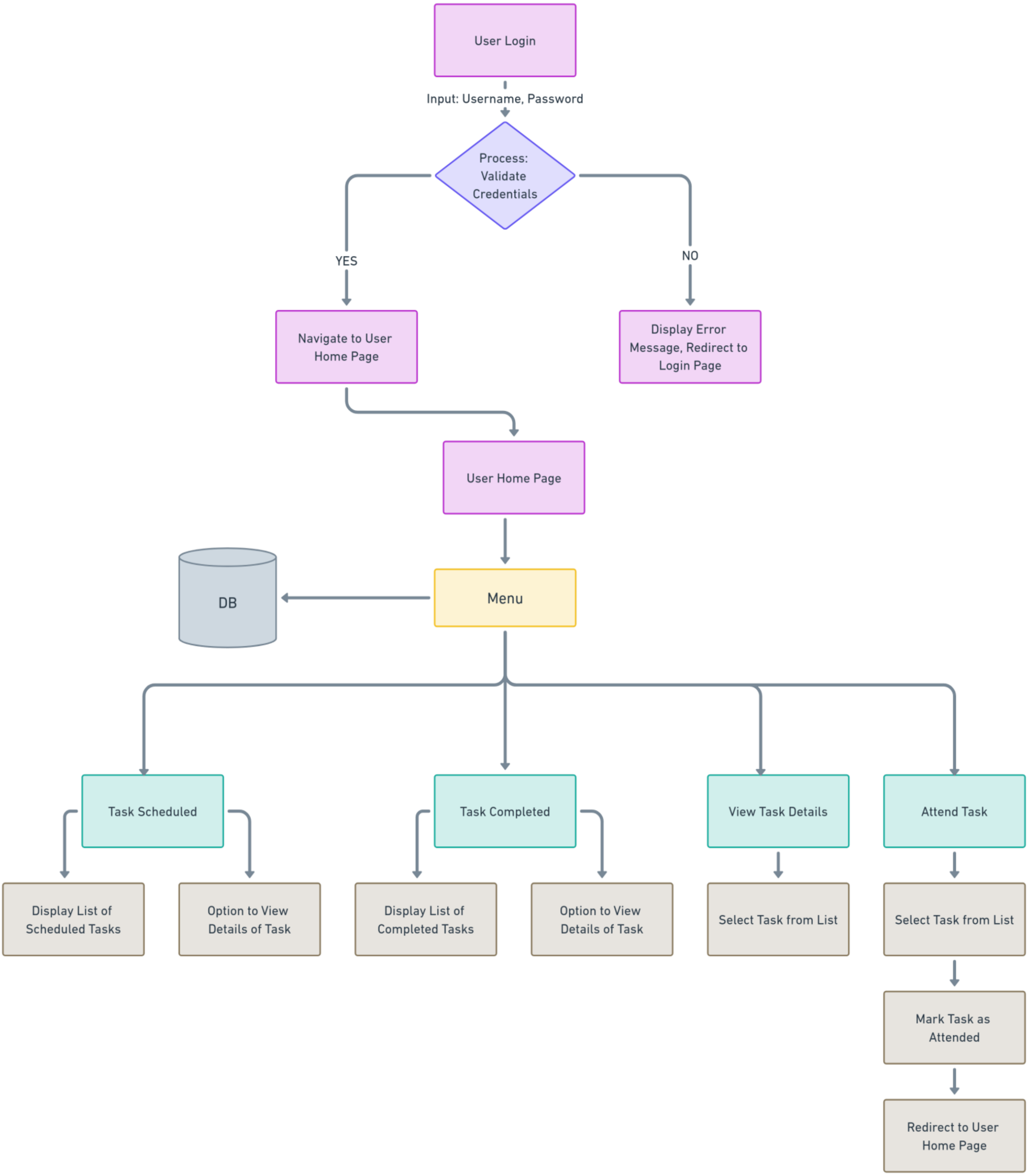
Make sure the application is fully tested, fix any bugs, and then move the Task Dashboard to the designated cloud computing platform or hosting environment.

FLOWCHART

Admin's page


Admin Login
Input: Username, Password
ProcessValidateCredentials
Valid Credentials?
ValidCredentials
Yes
No
NavigateTo Admin HomePage
DisplayError MessageRedirectTo LoginPage
DB
Admin HomePage
MenuOptions
Logout
View Tasks
Add Task
Update Task
Delete Task
• DisplayListOfTasks
• FilterTasks
• InputTaskDetails
• Process:SaveTask
• RedirectToUserPage
• SelectTaskFromList
• InputNewTaskDetails
• ProcesUpdateTask
• RedirectToUserHomePage
• SelectTaskFromList
• ConfirmDeletion
• ProcessDeleteTask
• RedirectToUserPage

User’s Page


User Login
Input: Username, Password
Process: Validate Credentials
YES
NO
Navigate to User Home Page
Display Error Message, Redirect to Login Page
User Home Page
Menu
DB
Task Scheduled
Task Completed
View Task Details
Attend Task
Display List of Scheduled Tasks
Option to View Details of Task
Display List of Completed Tasks
Option to View Details of Task
Select Task from List
Select Task from List
Mark Task as Attended
Redirect to User Home Page