Software Requirement Specification for Template Task

Name	Vethashri M S
Roll no	7376222BT213
Seat no	351
Project ID	8
Problem Statement	Template Task

1. Problem Statement:

The goal is to create a template task management system that includes an admin login page, the ability to create tasks with descriptions, store tasks in a table, prevent repeated tasks, and show the completion status of tasks.

2. Project Flow:

Admin Login Page:

- Create an Admins table with columns such as Admin ID, Username, and Password to store admin user information.
- Upon successful login, admins will have access to the task management system.

Create Task with Description:

- Create a Tasks table with columns such as Task ID, Description, and Completed to store task information.
- Assign a unique Task ID to each task to ensure its identification.

Store Tasks in Table:

- When a task is created, insert the task details into the Tasks table. Each task should have a unique Task ID and a description associated with it.
- The Completed column can be initially set to false to indicate that the task is not yet completed.

Prevent Repeated Tasks:

- To prevent repeated tasks, you can implement a check before inserting a new task into the Tasks table.
- If a task with the same description is found, display a message indicating that the task is repeated and prevent its insertion.

Show Task Completion Status:

- To show the completion status of tasks, you can query the Tasks table and retrieve the Completed column value for each task.
- Display the task completion status, such as "Completed" or "Not Completed," based on the value of the Completed column.

3. Functional Requirements:

Admin Login Page:

- Create an admin login page with the following fields:
    i. Username
    ii. Password
- Validate the username and password using a database or a secure authentication method.
- Redirect the admin to the dashboard after successful login.

Create Task:

- Create a form to allow admins to create new tasks with the following fields:
    i. Task Title
    ii. Task Description
    iii. Due Date
- Validate the input data to ensure it is not empty and meets the required format.
- Store the task data in a database table.

Task Table:

- Create a database table to store task data with the following columns:
    i. Task ID (unique identifier)
    ii. Task Title

iii. Task Description

iv. Due Date

v. Completion Status (boolean or enum)

Prevent Repeated Tasks:

Implement a check to prevent duplicate tasks from being created by checking for existing tasks with the same title and description.

Task Completion Status:

- Allow admins to mark tasks as completed by updating the completion status in the database table.
- Display the completion status of each task in the dashboard or task list.

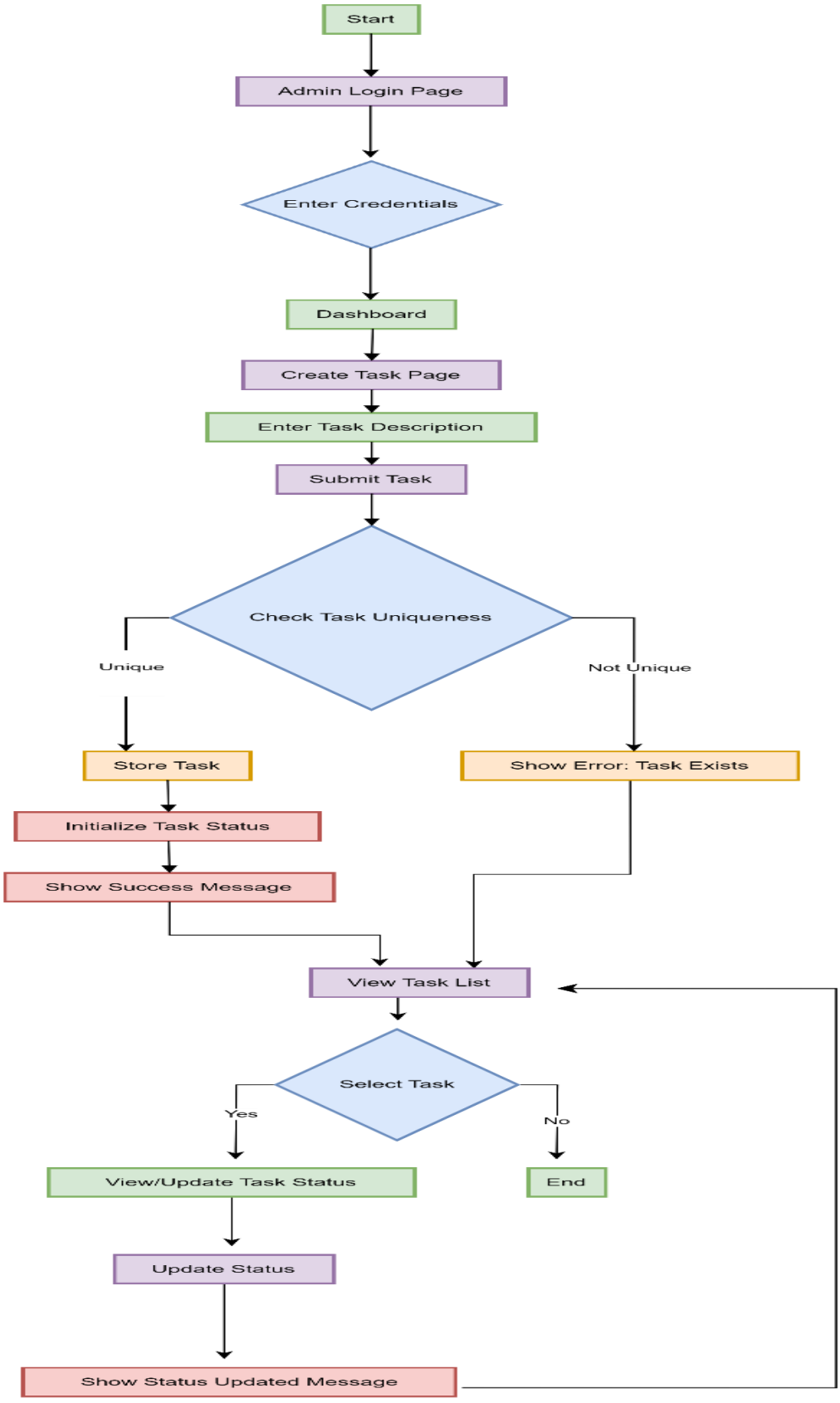
4. Non-Functional Requirements:

- **Security:** Ensure that all user input data is validated and sanitized to prevent SQL injection attacks and other security vulnerabilities.
- **Usability:** Design the admin login page and task creation form to be user-friendly and intuitive.
- **Performance:** Ensure that the system can handle a large number of tasks and users without significant performance degradation.

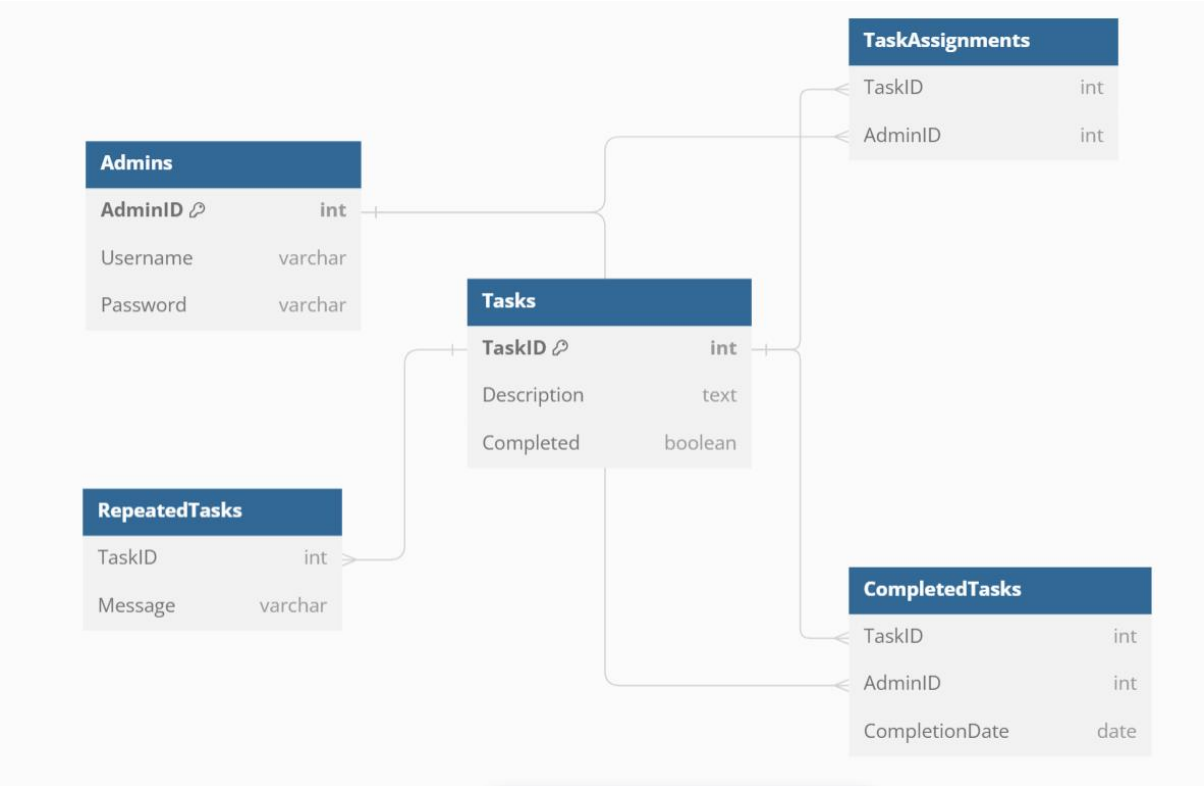
5.Technical Requirement:

Front End	• HTML • CSS • JS
Back End	• Python • Django(Python Web)
Database	• PostgreSQL • MySQL
API	• OpenAPI • SOAP APIs • REST Ful API

5.Flowchart:


Start
Admin Login Page
Enter Credentials
Dashboard
Create Task Page
Enter Task Description
Submit Task
Check Task Uniqueness
Unique
Not Unique
Store Task
Show Error: Task Exists
Initialize Task Status
Show Success Message
View Task List
Select Task
Yes
No
View/Update Task Status
End
Update Status
Show Status Updated Message

6. ER Diagram:


TaskAssignments
TaskID int
AdminID int
Admins
AdminID int
Username varchar
Password varchar
Tasks
TaskID int
Description text
Completed boolean
RepeatedTasks
TaskID int
Message varchar
CompletedTasks
TaskID int
AdminID int
CompletionDate date