**Business Requirements Document (BRD)**

**Project Title:** Task Flow: Remote Team Project Management Web Application

**Date:** September 3, 2024

**Submitted By:** Abin Benoy (284752), Nihal Varma Balerao (284707)

**Problem Statement:**

The shift to remote work has brought about significant challenges in team coordination and task management. Teams distributed across various time zones often struggle with asynchronous communication, leading to delays in project timelines and misunderstandings about task responsibilities. Additionally, the lack of face-to-face interaction can result in a disconnect between team members, making it harder to maintain accountability and ensure that everyone is on the same page. Traditional task management tools may not provide the level of flexibility needed to accommodate different work styles and time zones, leaving gaps in communication and collaboration.

**1. Project Overview**

‘Task Flow’ is a web application is designed to improve collaboration among remote teams by providing real-time updates, adaptable task management, and progress tracking features. The application uses a modular microservices architecture for scalability and flexibility, enabling project managers and team members to efficiently track progress, set deadlines, and assign tasks. This application addresses challenges in asynchronous communication, ensuring all team members remain aligned and engaged.

**2. Business Objectives**

* **Enhance Remote Collaboration:** Provide a platform that fosters effective collaboration among remote teams by centralizing project and task management.
* **Increase Productivity:** Streamline management processes and improve overall team productivity by providing tools for real-time updates, task tracking, and accountability.
* **Support Flexibility:** Enable flexible work arrangements by allowing team members to work across different time zones and manage tasks asynchronously.

**3. Stakeholders**

* **Project Manager:** Oversees the project, creates tasks, assigns them to team members, tracks progress, and generates reports.
* **Team Members (Employees):** Executes tasks assigned by the project manager, and updates task status.

**4. Functional Requirements**

**4.1 User Management**

* **User Roles:** The application will support two user roles—Project Manager and Employee.
* **Account Creation:** Project Managers will create accounts for Employees, who will then receive login credentials.
* **Authentication:** Secure authentication mechanisms, including username and password, will be used to log in to the application.

**4.2 Project Management**

* **Project Creation:** Project Managers can create new projects, including attributes such as project name, description, start date, and end date.
* **Task Management:** Project Managers can create, assign, and manage tasks within a project. Task attributes include name, description, due date, priority, status, and assigned employee.
* **Task Summary:** Project Managers can view task summaries based on specific dates, showing how many tasks were completed and how many were overdue.

**4.3 Progress Tracking and Reporting**

* **Progress Tracking:** The application will track the progress of tasks and projects, allowing Project Managers to monitor the completion of tasks.
* **Reporting:** Project Managers can generate reports or view dashboards that visualize project status, including task completion rates and overdue tasks.

**5. Non-Functional Requirements**

**5.1 Security**

* **Data Protection:** User data, including authentication details, will be encrypted to ensure security.
* **Role-Based Access Control:** Access to features and data will be controlled based on user roles.

**5.2 Performance**

* **Response Time:** The application should provide quick response times, with page loads and data updates happening within a few seconds.

**5.3 Usability**

* **User Interface:** The application will have a user-friendly interface that is intuitive and easy to navigate.

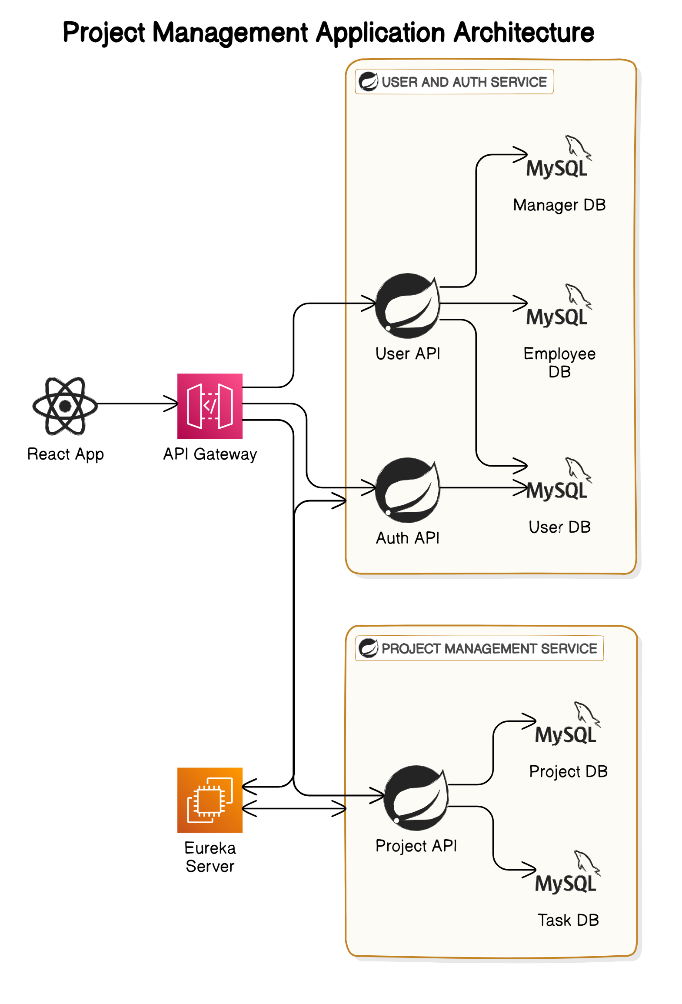
**5.4 Reliability**

* **Uptime:** The application should be available 99.9% of the time, with minimal downtime for maintenance and updates.
* **Error Handling:** The application will include robust error handling to ensure stability and reliability.

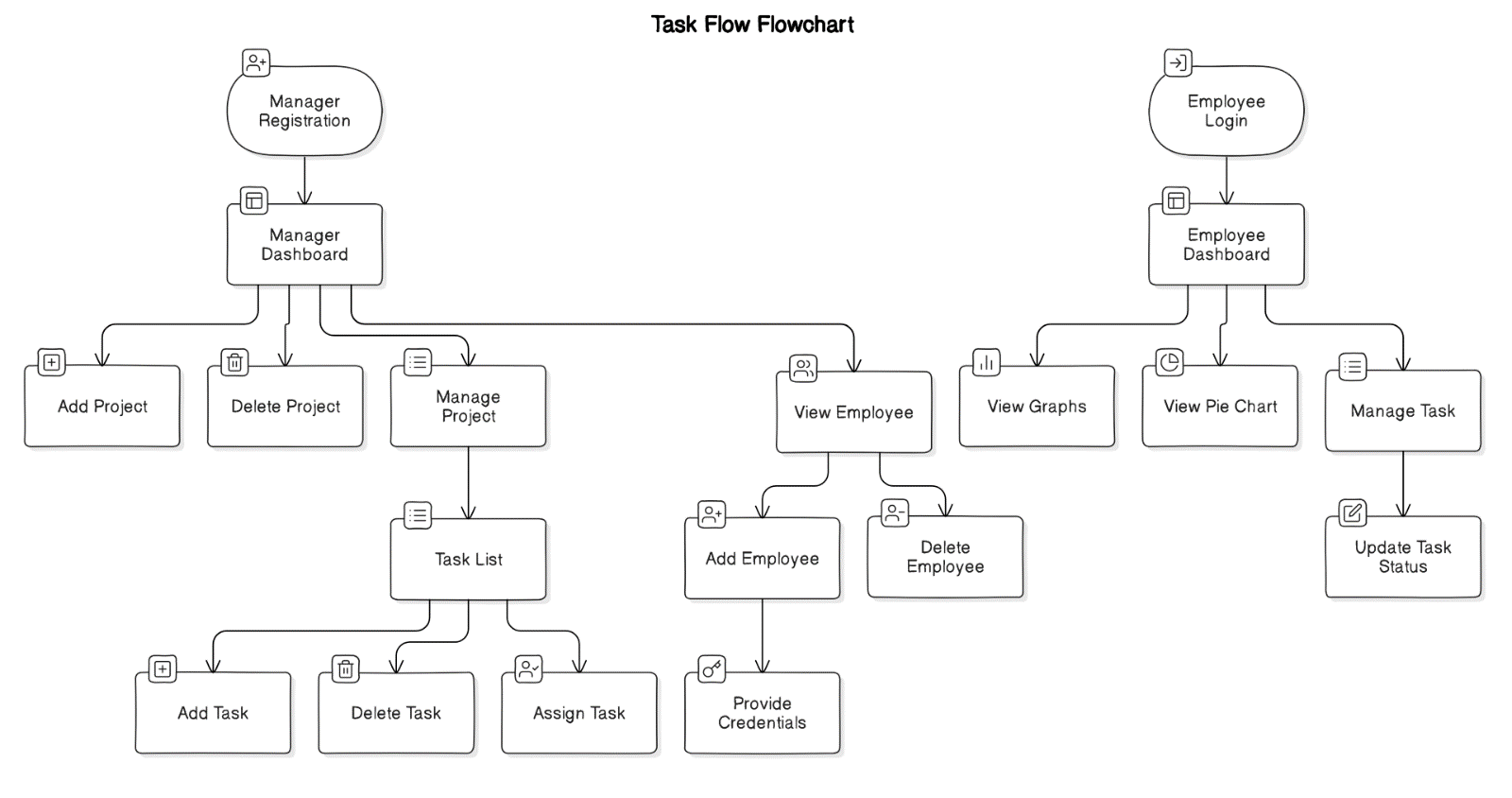
**6. Project Architecture**

The project architecture follows a microservices approach with a front end built using React, and a back end developed with Spring Boot. The system leverages two Eureka server-based microservices to handle core functionalities such as user management, project/task management. The database is MySQL, chosen for its efficiency in managing data.

* Front End: React.js
* Back End: Spring Boot with RESTful APIs.
* Microservices: Registered via Eureka servers for service discovery and load balancing.
* Database: MySQL for efficient data storage and retrieval.



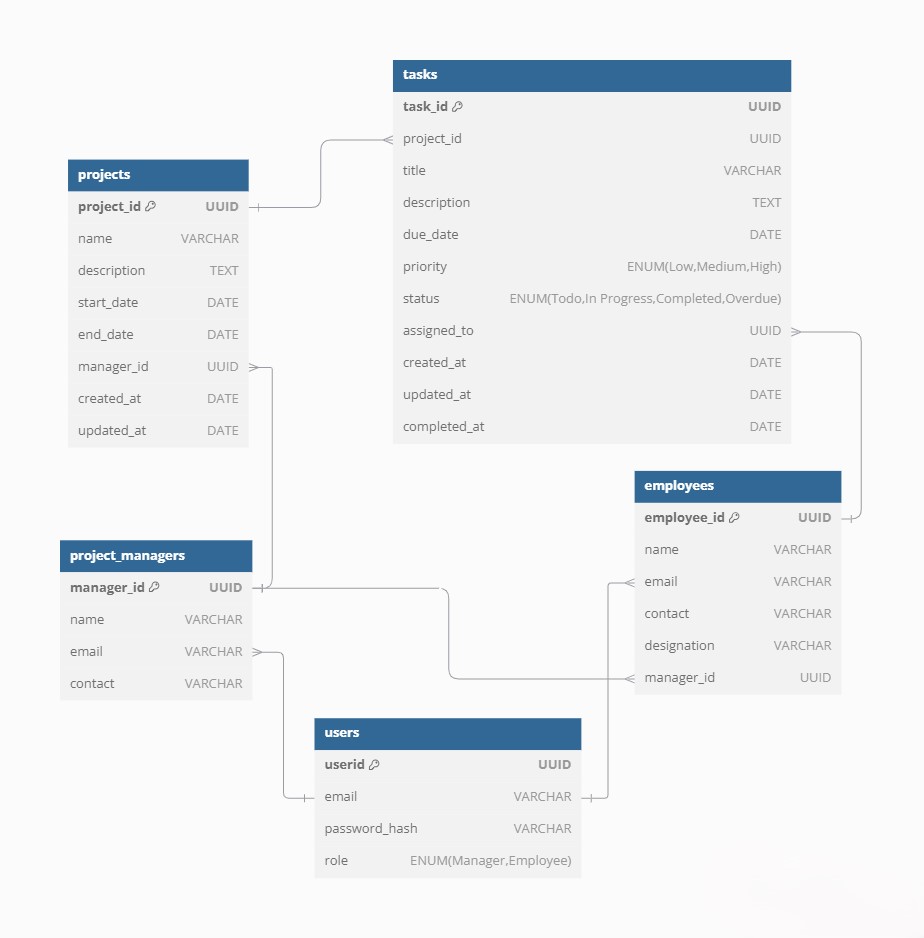
**7. Application Workflow**



**8. Database Schema Overview**

The application will include the following tables in its database schema:

* **Project Managers:** Stores details of project managers, including username and password for authentication.
* **Employees:** Stores details of employees, including username and password for authentication, and the project manager they report to.
* **Projects:** Contains information about projects and the associated project manager.
* **Tasks:** Stores task details, including assignment and status updates.



**9. Assumptions**

* Users will have access to stable internet connections.
* Project Managers and Employees will receive appropriate training on how to use the application.
* The application will be used by remote teams operating across different time zones.

**10. Constraints**

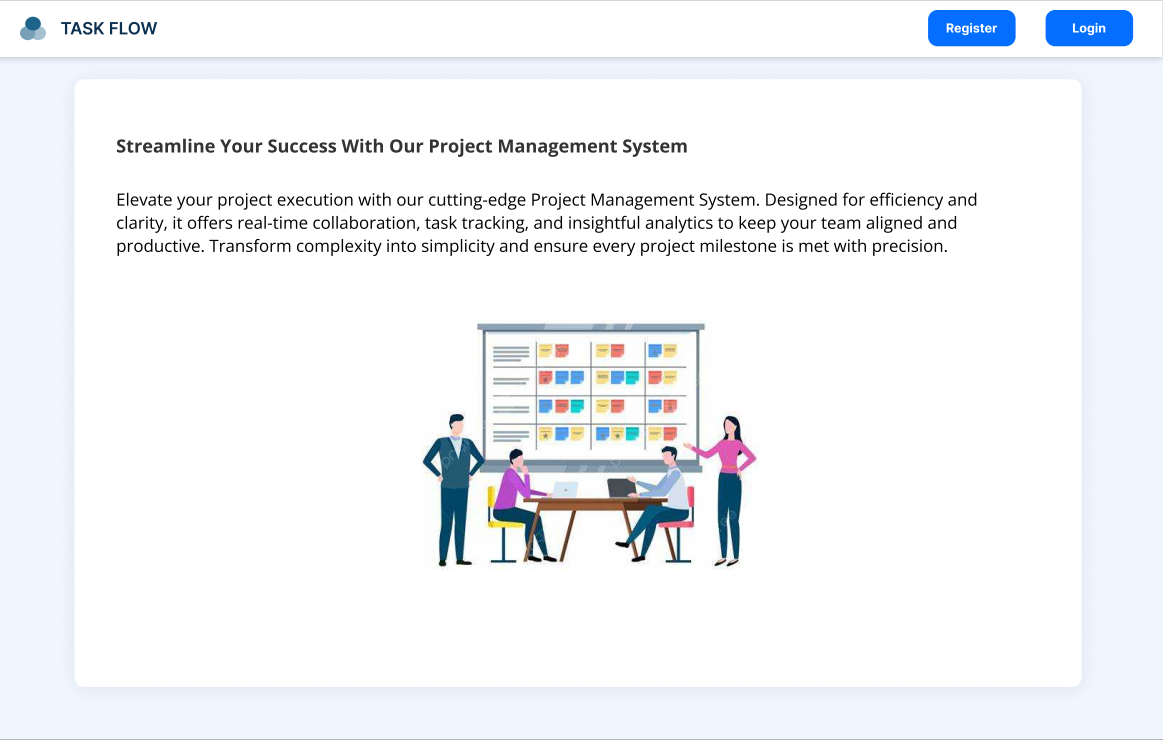
* Development and deployment timelines must be met to ensure the application is available for use within the specified project deadlines.

**11. Data Requirements**

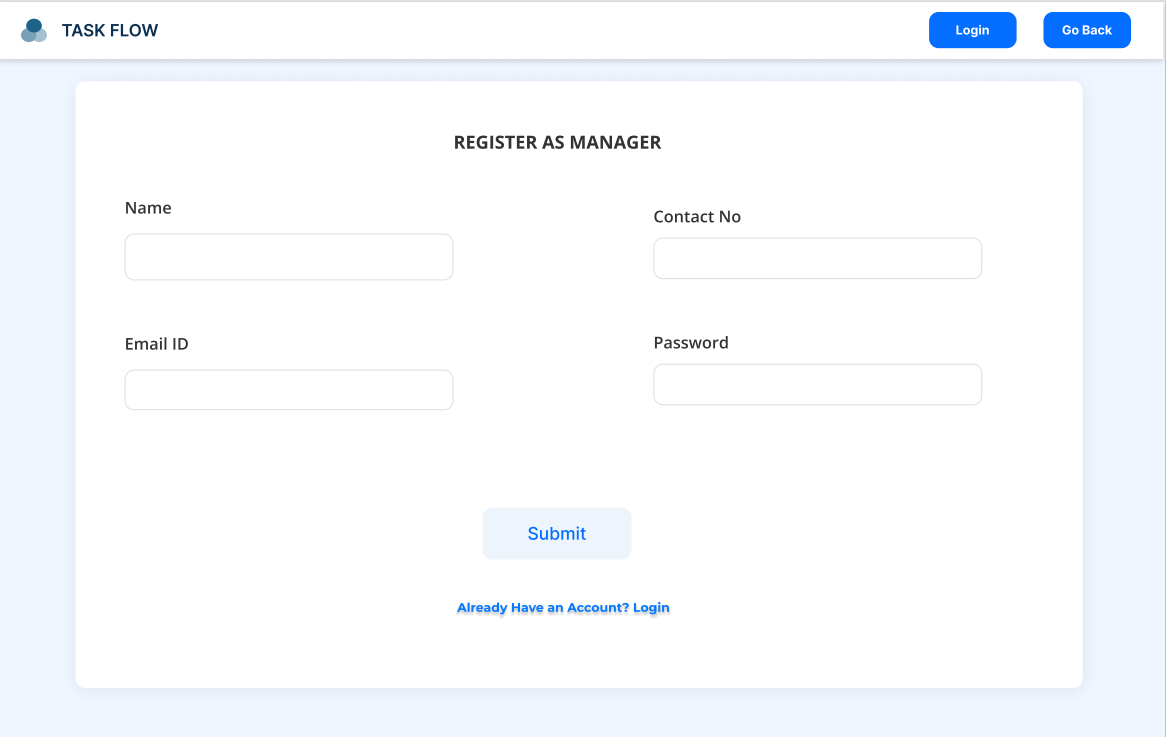
* **Project Data:** Information related to projects, including project name, description, start date, end date, and assigned manager.
* **Task Data:** Details related to tasks, including name, description, due date, priority, status and assigned employee.
* **User Data:** Information related to users, including username, password, role (Project Manager or Employee), and email.

**12. Figma Designs (Wire Framework)**

Home Page

****

Register Manager Page

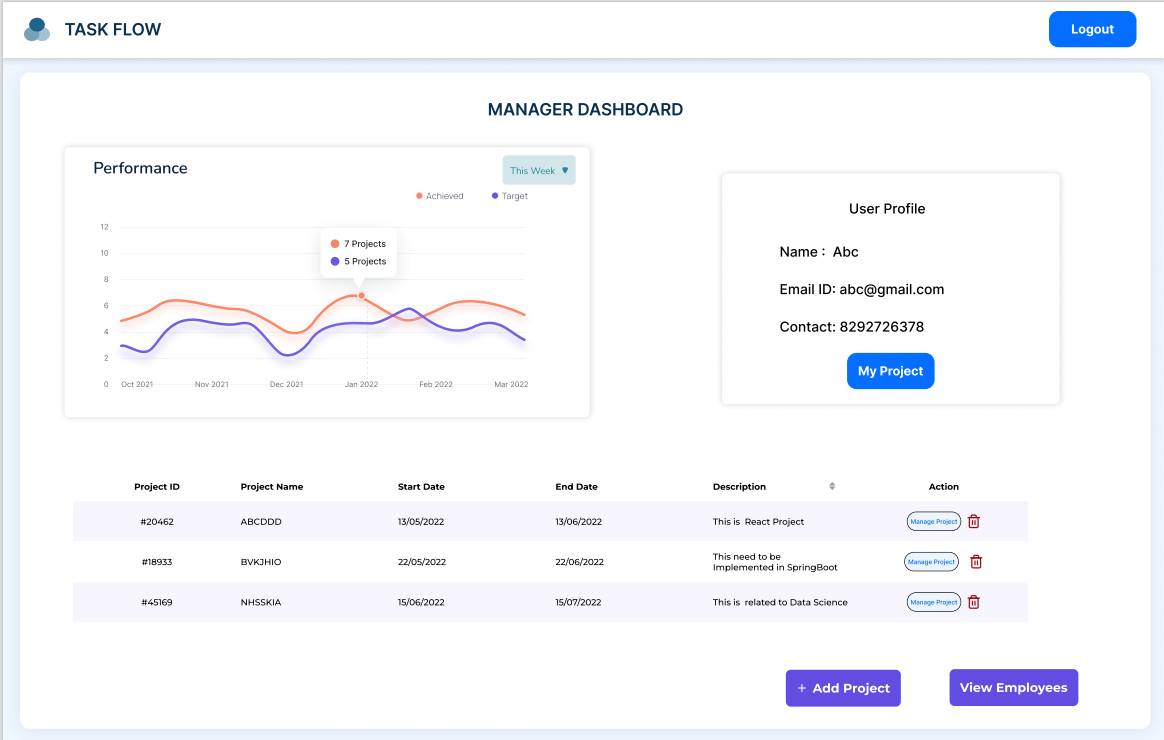
****

Login Page

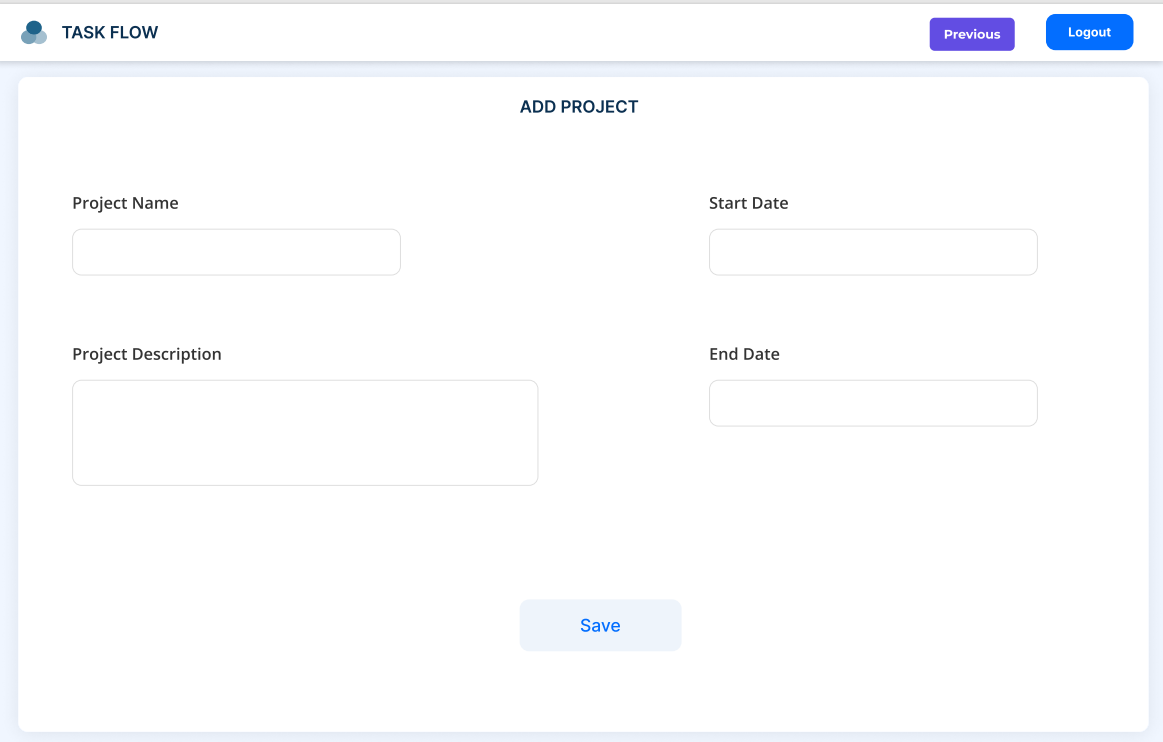
A screenshot of a login form

Description automatically generated

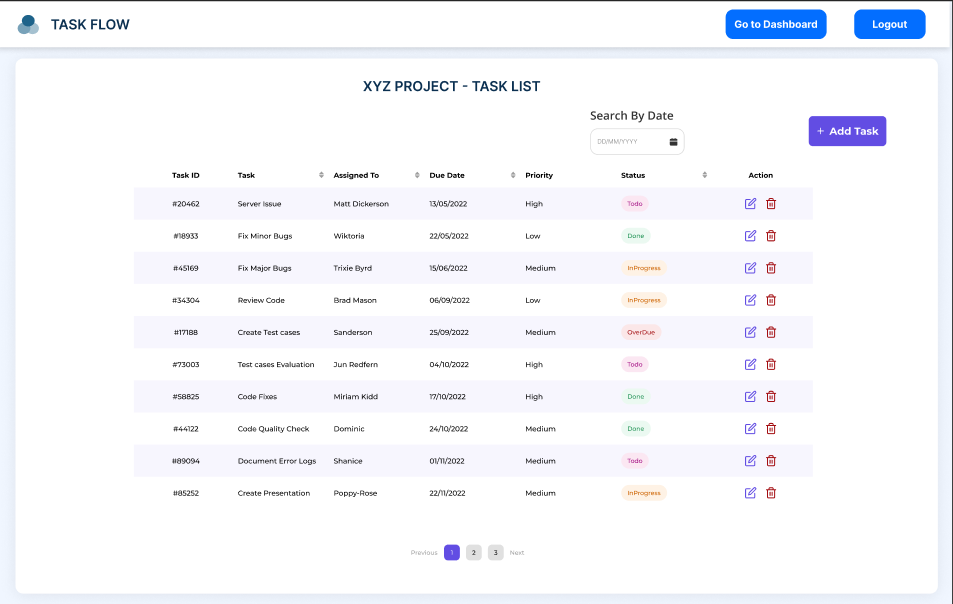
Manager Dashboard Page



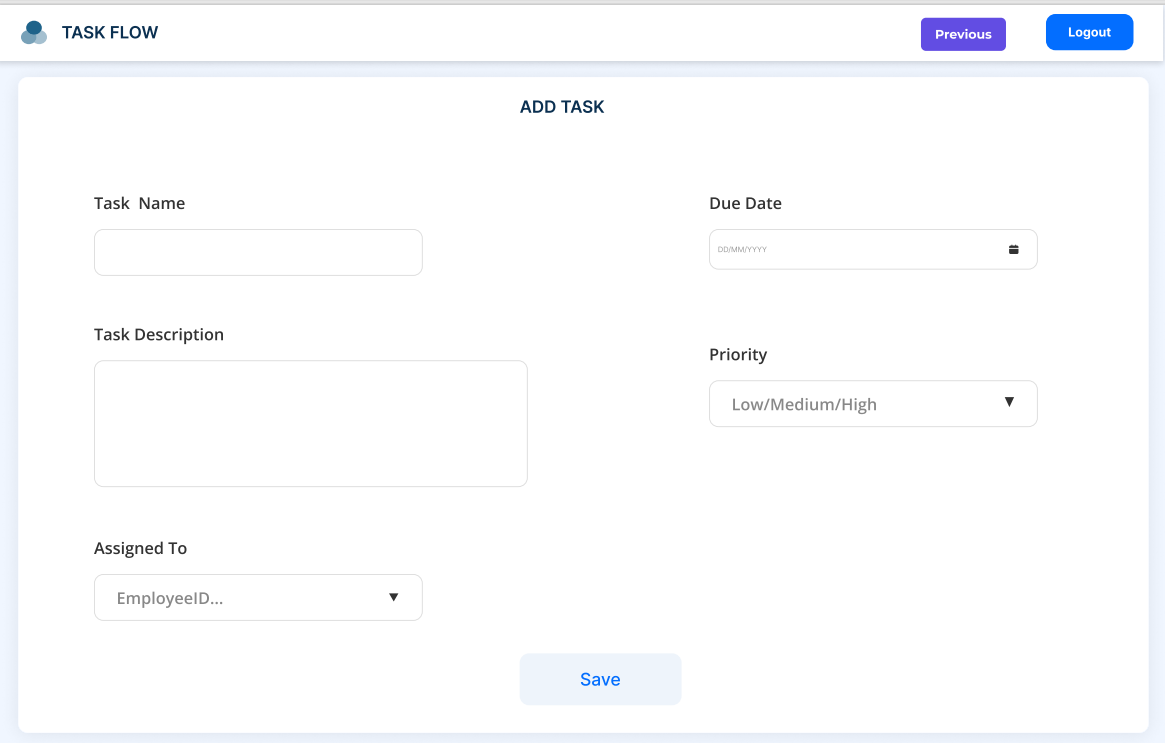
Add Project Page



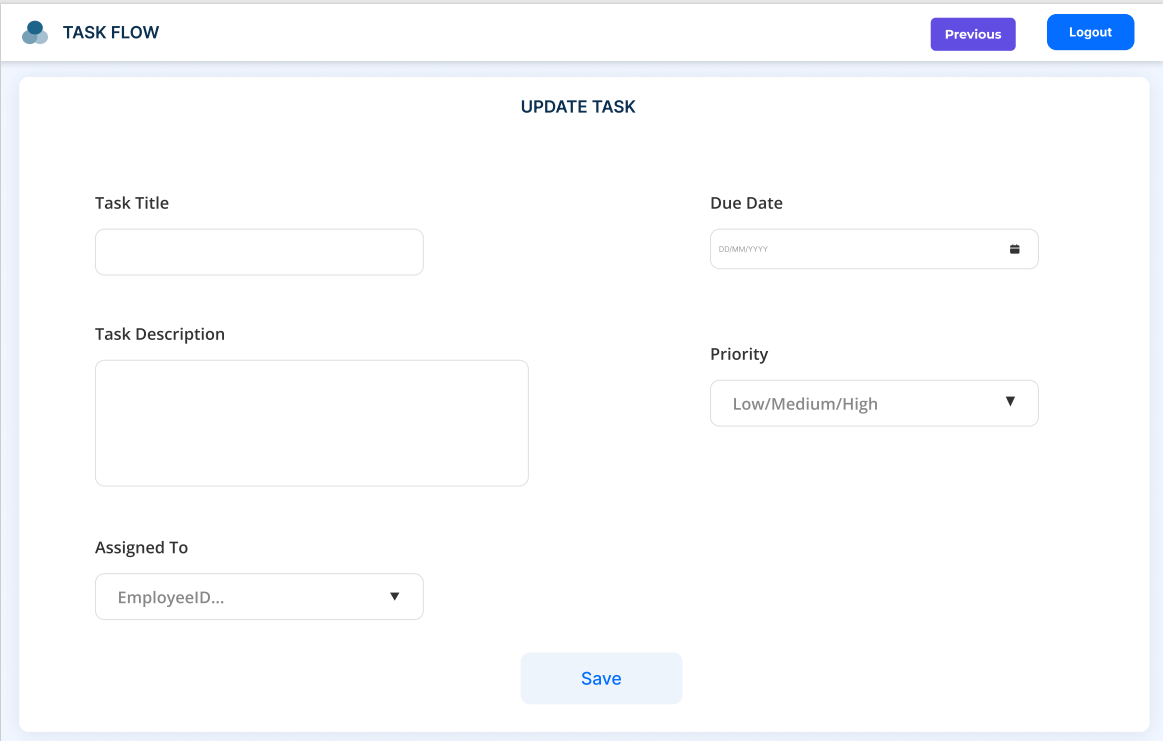
View Tasks page



Add Task Page



Update Task Page



Employee List Page

A screenshot of a computer

Description automatically generated

Add/Register Employee Page

A screenshot of a computer

Description automatically generated

Employee Dashboard Page

A screenshot of a computer

Description automatically generated

Task List for Employee

A screenshot of a project

Description automatically generated

Update Task Status

