A PROJECT ON

Employee Task Management System

SUBMITTED IN

PARTIAL FULFILLMENT OF THE REQUIREMENT

FOR THE COURSE OF DIPLOMA IN ADVANCED COMPUTING FROM CDAC

SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY

Hinjawadi

SUBMITTED BY:

Manthan Sarawade,

Raviraj Kale,

Shantanu Patil,

Swapnil Jha

UNDER THE GUIDENCE OF:

Mr. Nilesh Pawar

Faculty Member

Sunbeam Institute of Information Technology, Pune

ACKNOWLEDGEMENT

A project usually falls short of its expectation unless aided and guided by the right persons at the right time. We avail this opportunity to express our deep sense of gratitude towards Mr. Nitin Kudale (Center Coordinator, SIIT, Pune) and Mr. Yogesh Kolhe (Course Coordinator, SIIT, Pune).

We are deeply indebted and grateful to them for their guidance, encouragement, and deep concern for our project. Without their critical evaluation and suggestions at every stage of the project, this project could never have reached its present form.

Last but not the least, we thank the entire faculty and the staff members of Sunbeam Institute of Information Technology, Pune for their support.

Manthan Sarawade,

Raviraj Kale,

Shantanu Patil,

Swapnil Jha

0824 PG-DAC

SIIT

CERTIFICATE

This is to certify that the project work under the title 'Employee task Management System' is done by Manthan Sarawade, Raviraj Kale, Shantanu Patil, Swapnil Jha in partial fulfillment of the requirement for award of Diploma in Advanced Computing Course.

Mr. Nilesh Pawar

Project Guide

Mr. Yogesh Kolhe

Course Co-Coordinator

Date: 11-02-2025

Table of Contents:

ABSTRACT

In today's fast-paced work environment, organizations require an efficient system to manage employee tasks, track progress, and ensure timely completion of assignments. The Employee Task Management System is a web-based application designed to streamline task allocation, monitoring, and reporting within an organization.

This system provides role-based access for administrators, managers, and employees, allowing efficient task delegation and real-time tracking. Administrators can assign tasks to employees, set deadlines, and monitor progress, while employees can update their task status and submit reports. The system also features automated notifications, priority-based task management, and a structured dashboard for enhanced productivity.

Developed using React for the frontend and Spring Boot for the backend, the system follows the MVC architecture to ensure scalability, maintainability, and security. The database is managed using MySQL, ensuring structured data storage and retrieval.

The Employee Task Management System significantly improves task visibility, reduces manual tracking efforts, and enhances workplace efficiency by providing a transparent and organized task management solution.

INTRODUCTION

1.1Introduction:

The Employee Task Management System (ETMS) is designed to help organizations efficiently manage and track employee tasks. It provides a structured approach to task allocation, tracking, and verification, ensuring a streamlined workflow and improved productivity. This system allows administrators, managers, and employees to collaborate effectively, reducing miscommunication and missed deadlines.

1.2 Purpose:

This report presents the general use cases and data models of the Employee Task Management System. With these use cases and data models, this report aims to illustrate the system's processes and facilitate the project's development.

1.3 Problem Statement:

Task management is a crucial aspect of business operations. Employees and managers often struggle with organizing and tracking tasks efficiently. Without a structured task management system, productivity can decrease, deadlines can be missed, and miscommunication can occur. This report outlines the Employee Task Management System, which provides a structured approach to task allocation, tracking, and verification, thereby improving workflow within an organization.

1.4 The Scope of the Project:

This system will be a web-based application that enables users to manage tasks effectively. The system allows the admin to verify registered users, add managers, and assign projects. Managers can create and assign tasks to employees. Employees can view their assigned tasks and submit them for review. Managers will then verify tasks and update the status as either completed or rejected.

1.5 Aims & Objectives:

After reading this unit, learners will be able to:

- Understand the concept of Task Management.
- Study the different types of task assignments.
- Analyse the role of task tracking in business efficiency.
- Study the various types of Task Management structures.

1.6 Benefits of Employee Task Management System:

The potential benefits of a structured task management system include:

1. **Enhanced Productivity and Efficiency:** Streamlines task allocation, tracking, and verification, enabling employees to focus on their priorities and reducing delays.
2. **Improved Collaboration and Communication:** Facilitates clear, consistent communication among administrators, managers, and employees, ensuring everyone stays informed on task progress.
3. **Increased Accountability and Transparency:** Creates a clear record of task assignments, progress, and completions, making it easier to monitor performance and hold team members accountable.
4. **Optimized Workflow and Resource Management:** Provides real-time updates and progress tracking, allowing for efficient resource allocation and quick adjustments to project needs.

1.7 Overview of Document:

The document is divided into the overall description and functional requirements. The overall description outlines the basic use cases for the system's processes, while the functional requirements detail the necessary interactions to support these use cases.

Overall Description

2.1 Product Perspective:

A structured task management system enables employees and managers to track work progress, ensuring timely completion of projects. The system provides a user-friendly interface for task assignment, submission, and verification, improving efficiency and accountability.

2.2 Product Features:

Admin:

● Admin can log in to the system. ● Verify registered users. ● Add Managers. ● Assign Projects. ● Manage Users.

Manager:

● Managers can log in to the system. ● View assigned projects. ● Create and assign tasks to employees. ● Verify submitted tasks. ● Update task status as completed or rejected.

Employee:

● Employees can log in to the system. ● View assigned tasks. ● Submit completed tasks. ● View task status updates.

2.3 User Classes and Characteristics:

Admin:

- Has full system control.
- Verifies user registrations, assigns managers, and maintains system integrity.

Manager:

- Oversees task assignments and monitors project progress.
- Ensures task completion and updates task status.

Employee:

- Receives and completes assigned tasks.
- Submits work for verification and tracks status updates.

Requirements Specification:

3.1 Software Requirements:

- **Technology:** Java (J2SE, J2EE), Hibernate, Spring Boot
- **SDK:** AWS EC2, AWS S3 Standard Bucket
- **Web Technologies:** React, CSS, JavaScript, Material UI
- **Web Server:** Apache Tomcat 9.0
- **Java Version:** Java 17
- **Database:** MySQL 8.0
- **IDE:** Spring tool suite 4.

3.2 Hardware Requirements (Minimum):

- **Processor:** Intel Core i3 or equivalent
- **RAM:** 4GB
- **Storage:** 160GB HDD or SSD

3.3 Performance, Safety, and Security Requirements:

- The system should be accessible 24/7 with minimal downtime.
- Secure authentication mechanisms should be in place to protect user credentials.
- Data should be backed up regularly to prevent data loss.
- Only authorized personnel should access confidential data, ensuring cybersecurity compliance.

3.4 Software Quality Attributes:

- **Availability:** The system should be accessible across multiple operating systems.
- **Accessibility:** The software should be user-friendly and easily navigable.
- **Compatibility:** The system should support various browsers and devices.
- **Scalability:** The software should handle multiple concurrent users efficiently.
- **Maintainability:** System updates and maintenance should be simple and cost-effective.

SPRING BOOT: Java Spring Boot (Spring Boot) is a tool that makes developing web applications and micro-services with Spring Framework faster and easier through three core capabilities: Auto-configuration. An opinionated approach to configuration. The ability to create standalone applications.

MySQL: MySQL is an open source 'Relational Database Management System' in which all the data are stored in the form of tables. Each table is connected to some other table i.e. has a relationship with another table and this relationship is established through integrity constraints. These tables have columns that represent the attributes of an entity and there are rows of data for each column. This is called the database and is connected to the front end or user interface with the help of a controller. This is a fast and highly scalable database management System.

Non-Functional Requirements

3.3.1 Performance Requirements: The system should store all database records properly and remain available for use 24/7 through the server. The application should be user-friendly with an intuitive interface, ensuring all options are easily accessible for user convenience.

3.3.2 Safety Requirements: All login credentials of administrators, managers, and employees should be securely stored using encryption. User records and task data should be backed up regularly across database servers. In case of a security breach or accidental deletion, a backup server should restore lost data.

3.3.3 Security Requirements: Authentication mechanisms should be in place to protect passwords and sensitive information. Only authorized users should access specific data, with access rights controlled by the administrator. The system should prevent unauthorized modifications or access to confidential records.

3.4.1 Software Quality Attributes:

3.4.1 Availability: The system should be accessible on multiple operating systems and support different hardware configurations.

3.4.2 Accessibility: The software will be accessible to administrators, managers, and employees with appropriate role-based access.

3.4.3 Compatibility: The software will support multiple platforms and browsers for ease of use.

3.4.4 Durability: The system will be stress-tested to ensure stable performance under concurrent user load and multiple task operations.

This report serves as a comprehensive guide to understanding the Employee Task Management System and its role in enhancing productivity and workflow efficiency within an organization.

3.4.1 Availability
The system should run on a variety of operating systems that support the JavaScript language. The system should run on a variety of hardware

3.4.2 Accessibility
The software will be accessible to admins, builders, and users.

3.4.3 Compatibility
The software will be compatible with multiple platforms.

3.4.4 Durability:
The software will be tested for working with multiple user

System Design

4.1 Use Case Diagram:

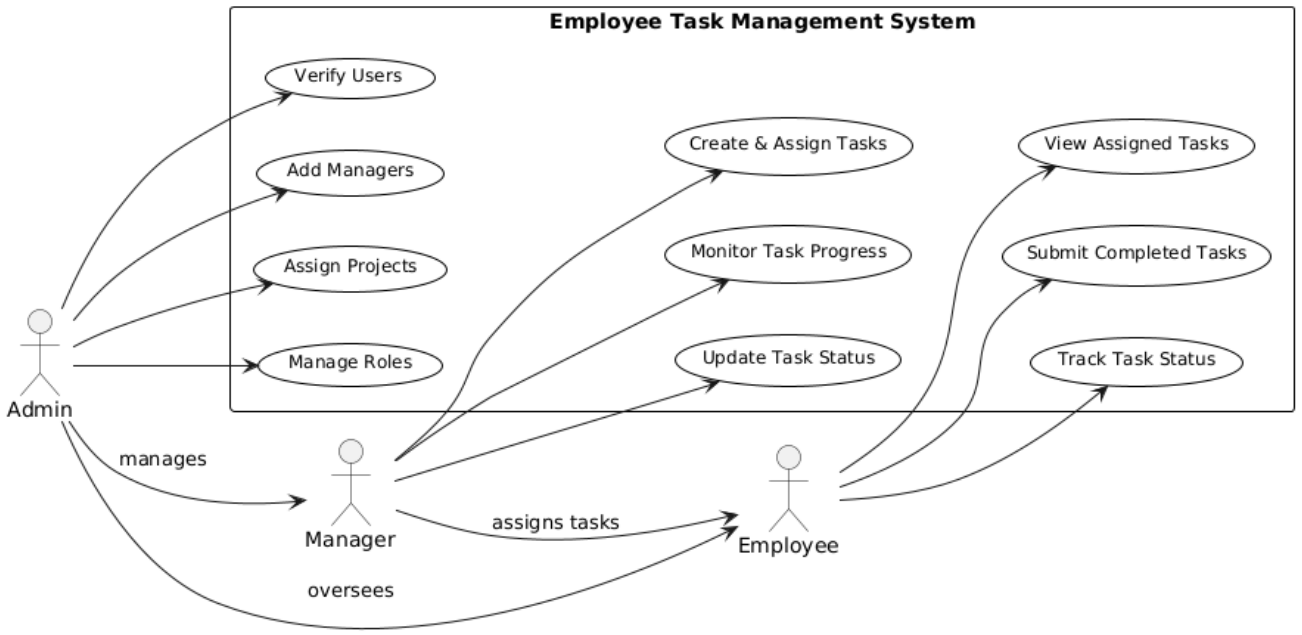


Figure 1.1: Use Case Diagram

4.1.1 Admin DFD

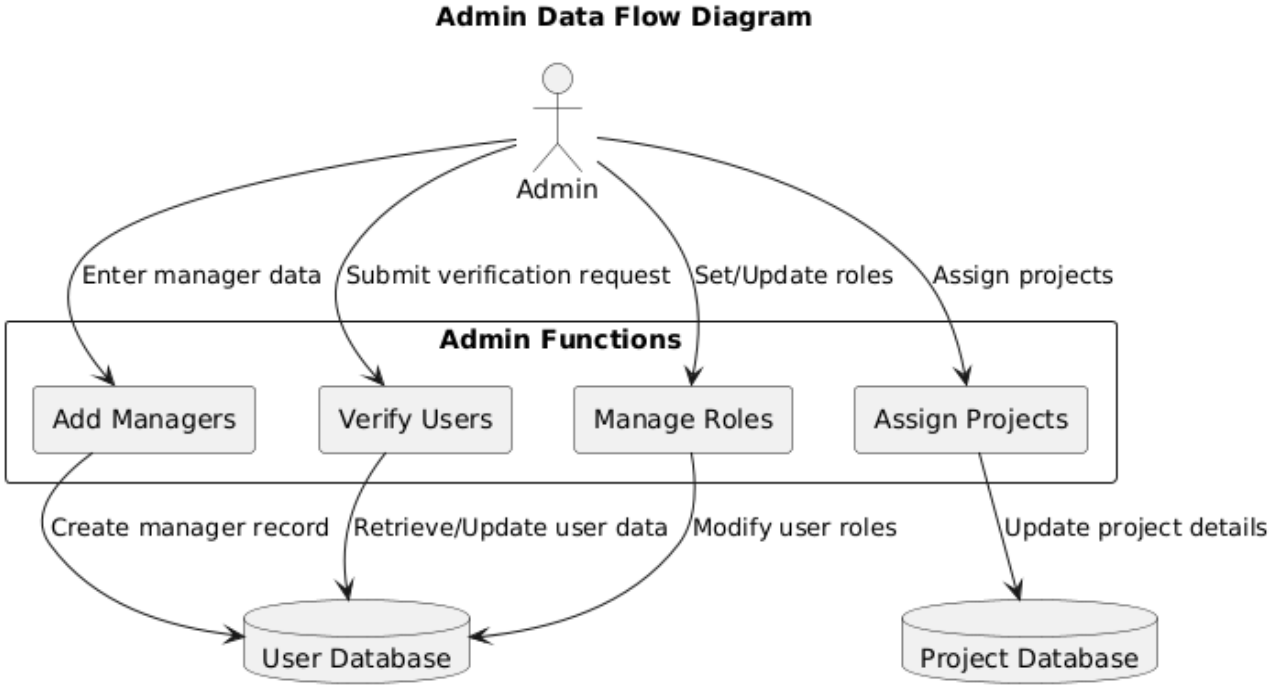


Fig 1.2 Admin DFD

4.1.2 Manager DFD

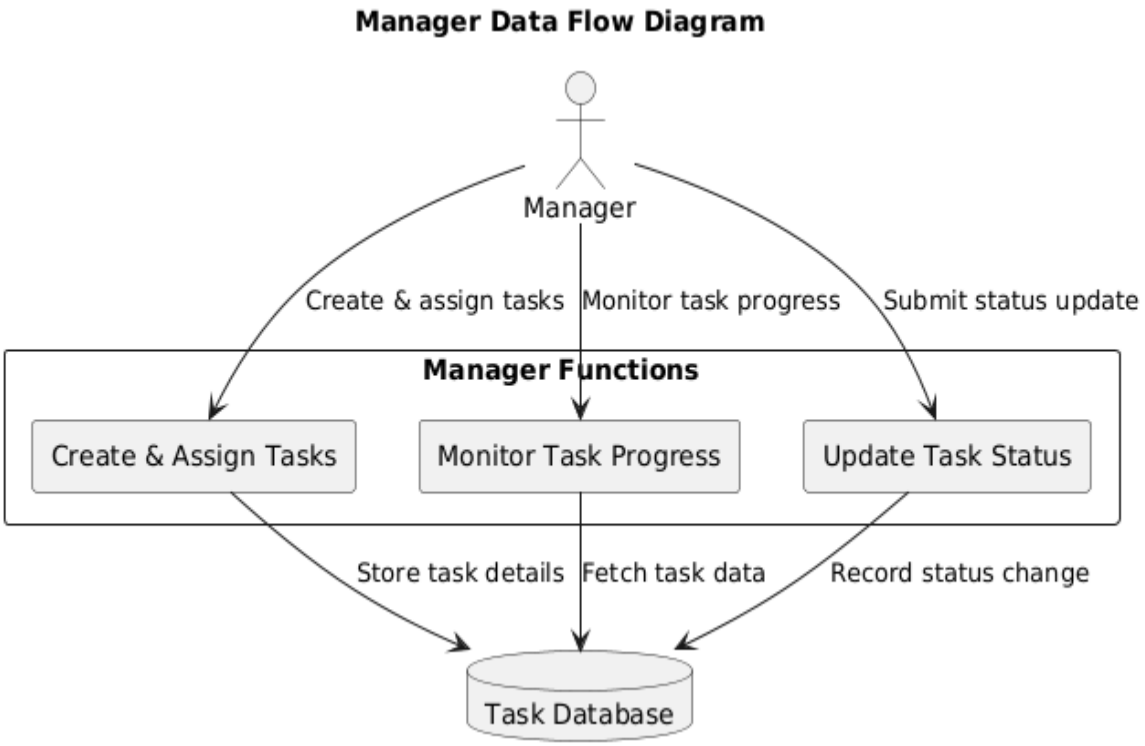


Fig 1.3 Manager DFD

4.2 Employee Data Flow

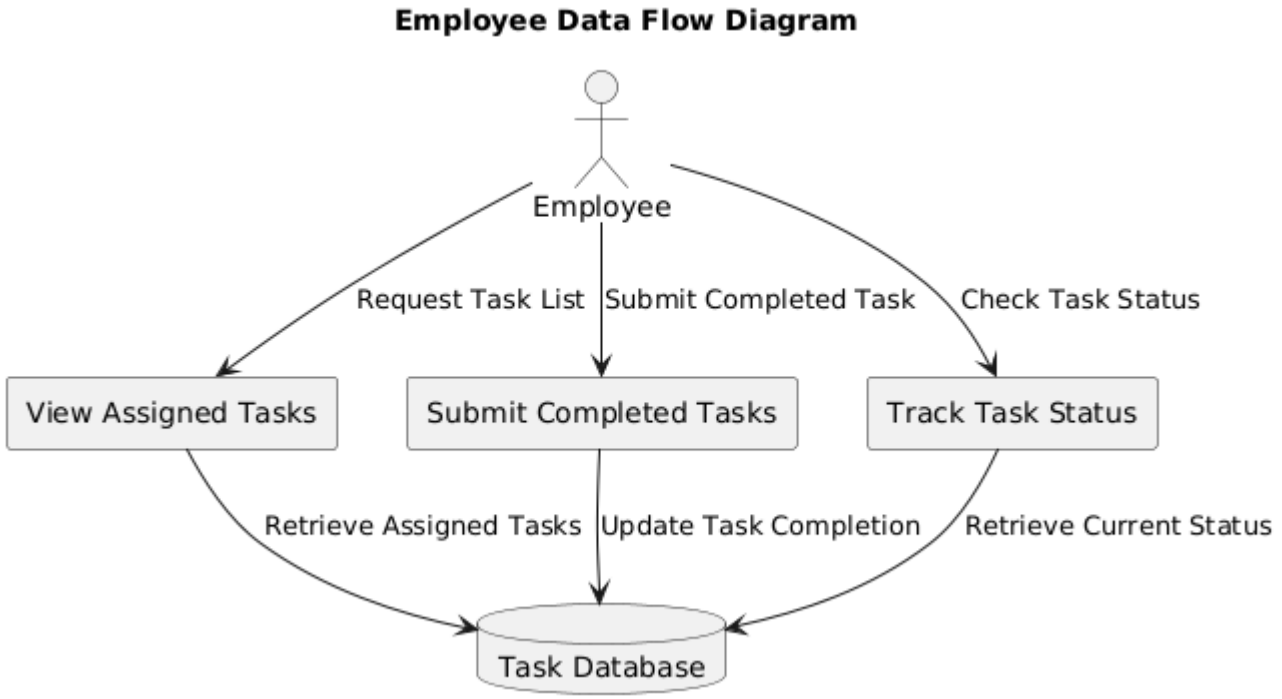


Fig 1.4 Employee DFD

4.3 Class Diagram

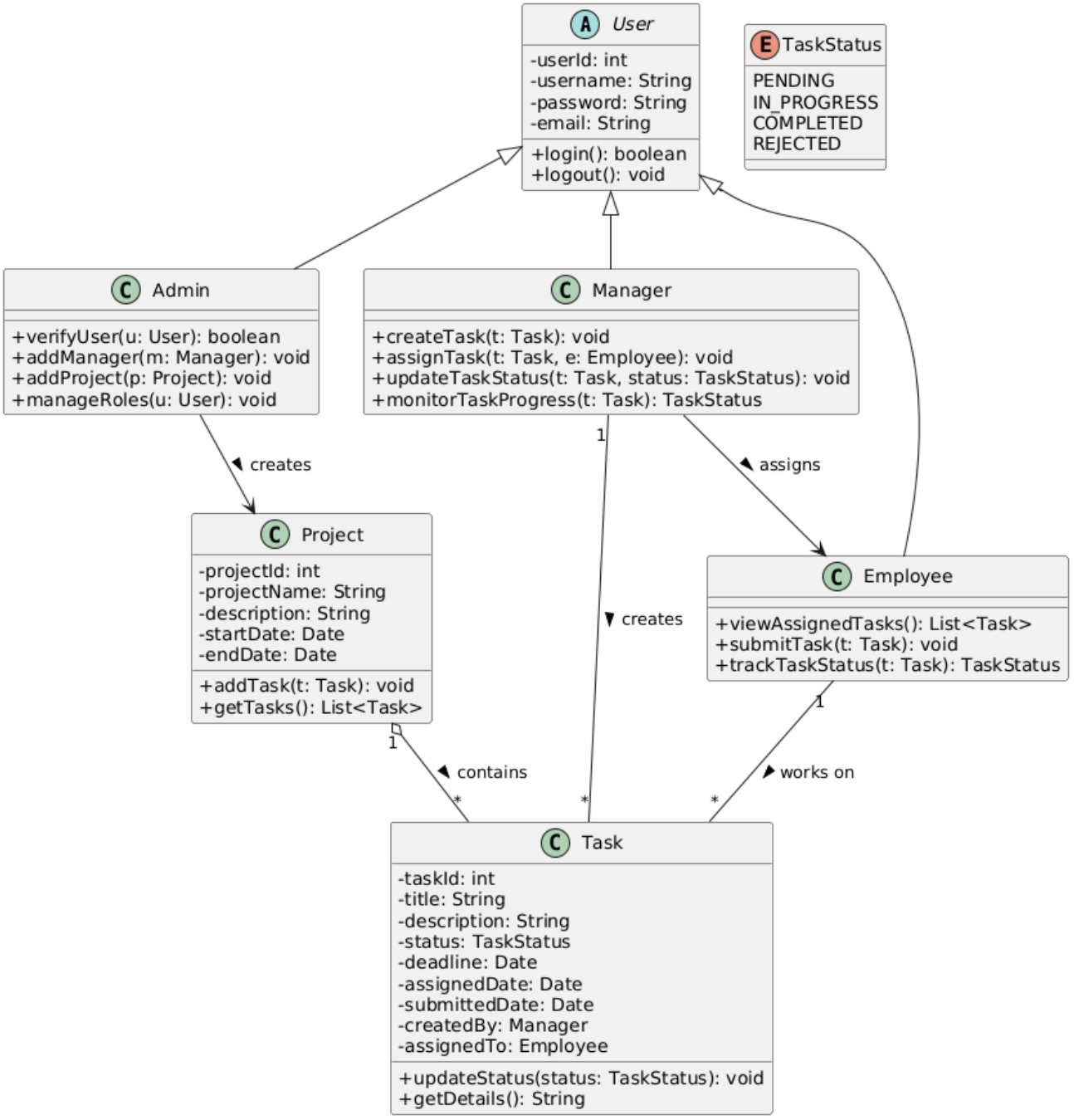


Fig 3: Class Diagram

4.4 ER Diagram

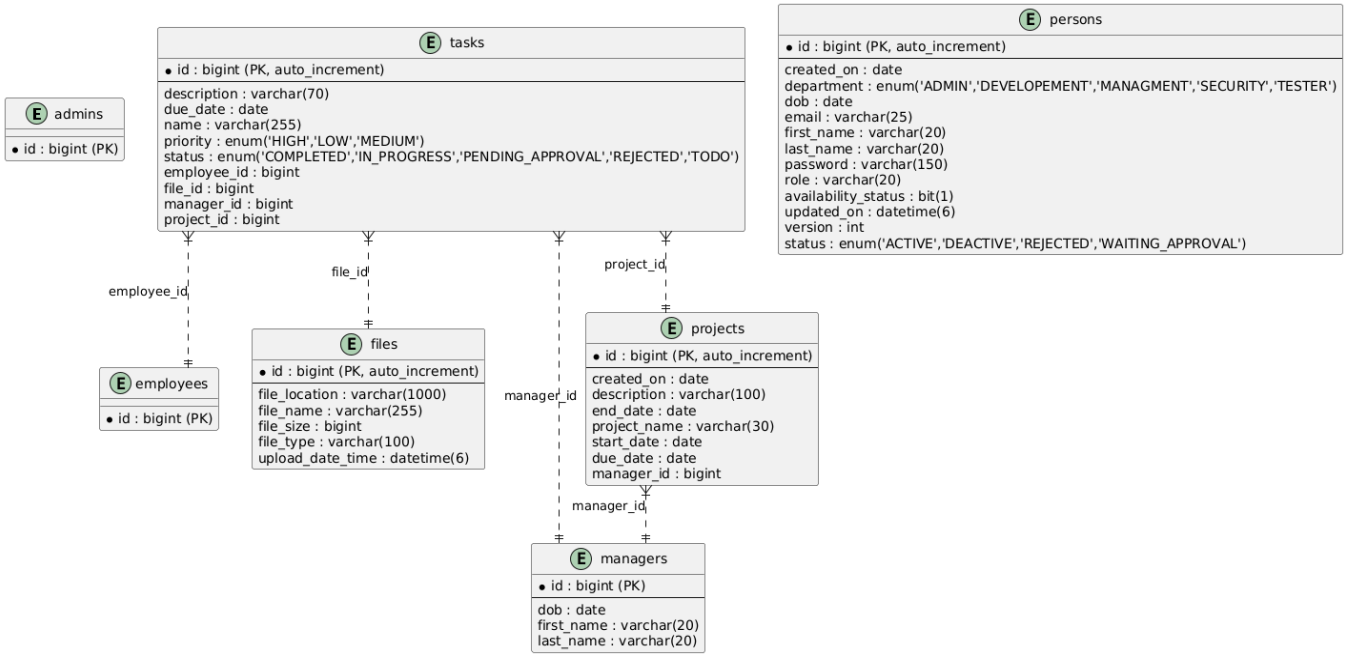


Fig 4: ER Diagram

4.1 Sequence Diagram

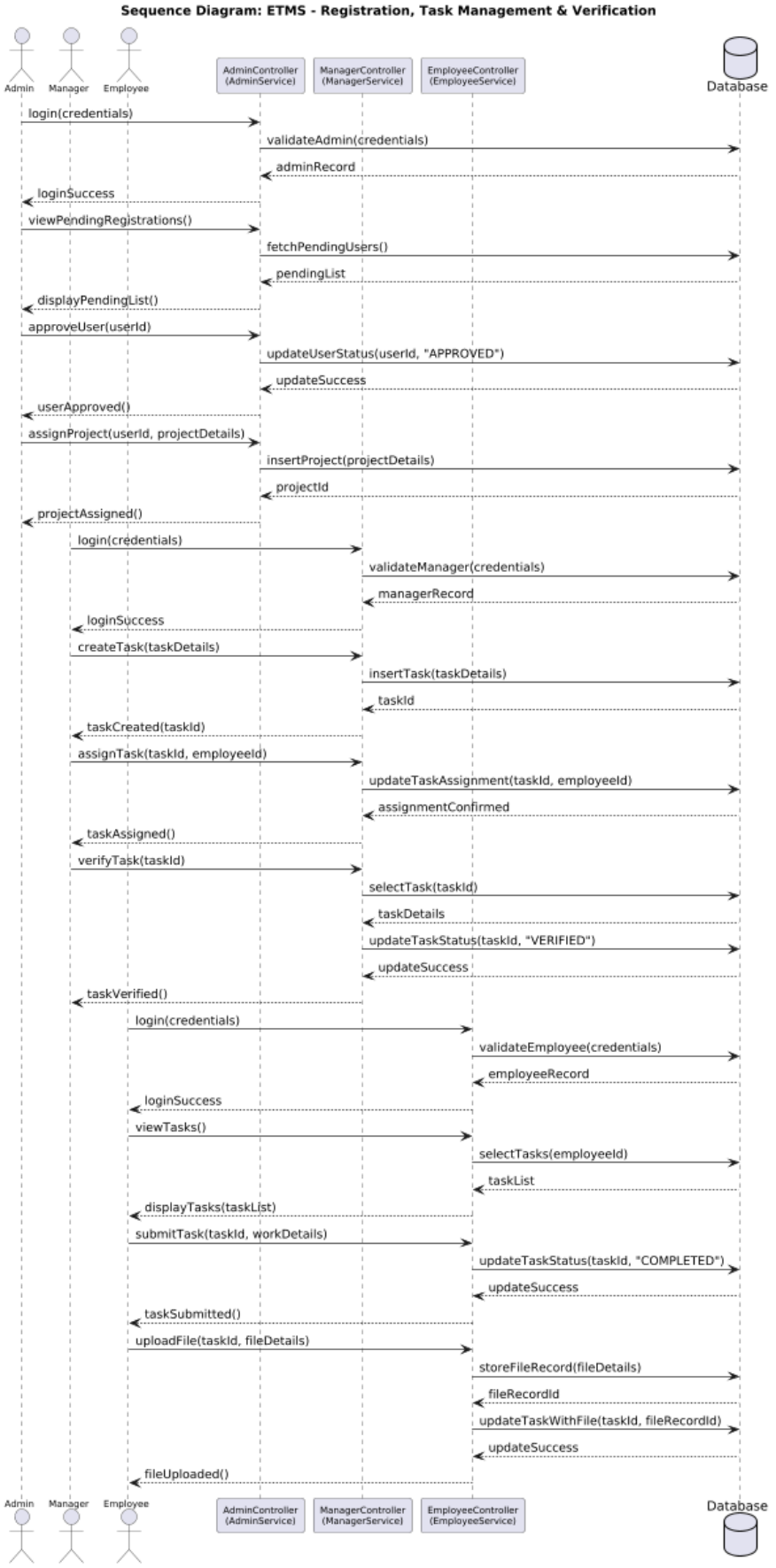


Fig 5: Sequence Diagram

Database Design

Table 1: persons

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
created_on	date	YES		NULL	
department	enum('ADMIN','DEVELOPEMENT','MANAGMENT','SECURITY','TESTER')	YES		NULL	
dob	date	YES		NULL	
email	varchar(25)	YES	UNI	NULL	
first_name	varchar(20)	YES		NULL	
last_name	varchar(20)	YES		NULL	
password	varchar(150)	NO		NULL	
role	varchar(20)	YES		NULL	
availability_status	bit(1)	YES		NULL	
updated_on	datetime(6)	YES		NULL	
version	int	NO		NULL	
status	enum('ACTIVE','DEACTIVE','REJECTED','WAITING_APPROVAL')	YES		NULL	

Table 2: Admin

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	

1 row in set (0.00 sec)

Table 3: Manager

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	
dob	date	YES		NULL	
first_name	varchar(20)	YES		NULL	
last_name	varchar(20)	YES		NULL	

Table 4: Employees

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	

Table 5: Projects

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
created_on	date	YES		NULL	
description	varchar(100)	YES		NULL	
end_date	date	YES		NULL	
project_name	varchar(30)	NO		NULL	
start_date	date	YES		NULL	
manager_id	bigint	NO	MUL	NULL	
due_date	date	YES		NULL	

Table 6: Tasks

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
description	varchar(70)	YES		NULL	
due_date	date	YES		NULL	
name	varchar(255)	NO		NULL	
priority	enum('HIGH','LOW','MEDIUM')	YES		NULL	
status	enum('COMPLETED','IN_PROGRESS','PENDING_APPROVAL','REJECTED','TODO')	YES		NULL	
employee_id	bigint	YES	MUL	NULL	
file_id	bigint	YES	UNI	NULL	
manager_id	bigint	YES	MUL	NULL	
project_id	bigint	YES	MUL	NULL	

Table 7: Files

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
file_location	varchar(1000)	YES		NULL	
file_name	varchar(255)	YES		NULL	
file_size	bigint	YES		NULL	
file_type	varchar(100)	YES		NULL	
upload_date_time	datetime(6)	YES		NULL	

CODING STANDARDS IMPLEMENTED

Naming and Capitalization

Below summarizes the naming recommendations for identifiers in Pascal casing is used mainly (i.e. capitalize first letter of each word) with camel casing (capitalize each word except for used in certain circumstances.

Identifier	Case	Examples	Additional Notes
Class	Pascal	Admin, Manager, Employee, Task, Project, ManagerController, ManagerService, File, Person	Classes should be nouns based on real-world objects. Do not use underscores or type prefixes.
Method	Camel	login, logout, createTask, assignTask, updateTaskStatus, viewTasks, submitTask, trackTaskStatus, uploadFile, updateEmployee	Methods should use verbs or verb phrases.
Parameter	Camel	username, password, email, taskId, projectId, deadline, adminId, employeeId, fileId	Use descriptive names that clearly indicate the purpose of the parameter.
Interface	Pascal with "I" prefix	IManagerService, IAdminService, IEmployeeService, IFileService, IProjectService, ITaskService	Interface names should be prefixed with "I". Do not use underscores.
Annotation	Pascal	RestController, RequestMapping, Autowired, Service	Use the '@' symbol at the beginning (e.g., @RestController).
DTOs	Camel	LoginDTO, TaskResponseDTO, ProjectResponseDTO, AdminDTO, EmployeeDTO, ManagerDTO, FileDTO	DTOs are used to transfer data between system layers.
Exception Class	Pascal with "Exception" suffix	ResourceNotFoundException, InvalidTaskException, FileUploadException, UnauthorizedAccessException	Exception class names must clearly indicate the type of error.

Comments:

Comment each type, each non-public type member, and each region declaration. Use end-line comments only on variable declaration lines. End-line comments are comments that follow code on a single line. Separate comments from comment delimiters (apostrophe) or // with one space. Begin the comment text with an uppercase letter. End the comment with a period. Explain the code; do not repeat

TEST REPORT

Another group called Linux did the testing and the report of the testing is given hereunder.

GENERAL TESTING:

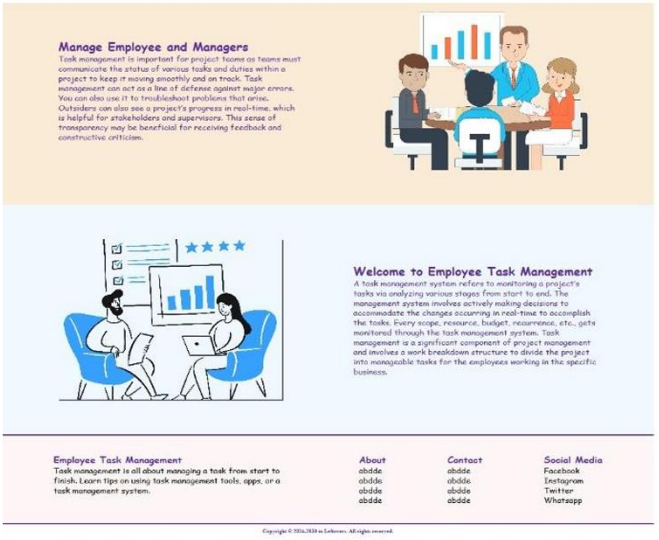
SR-NO	TEST CASE	EXPECTED RESULT	ACTUAL RESULT	ERROR MESSAGE
1	Sign Up Page	Sign Up page displays with registration form	Sign Up page OK	None
2	Successful Registration	User registers successfully and receives confirmation	Registration OK	None
3	Sign In Page	Sign In page displays with login fields	Sign In page OK	None
4	Successful Sign In	User is directed to home/dashboard based on role	Home page displayed	None
5	Task Creation (Manager)	Manager can create a new task with valid details	Task created	None
6	Task Assignment (Manager)	Manager assigns task to Employee successfully	Task assigned	None
7	View Assigned Tasks (Employee)	Employee sees list of assigned tasks	Tasks listed	None
8	Submit Task (Employee)	Employee submits completed task with file upload and receives confirmation	Task submitted	None
9	Task Verification (Manager)	Manager verifies submitted task; status updated to "Verified"	Task verified	None
10	Project Assignment (Admin)	Admin assigns project to Manager for department	Project assigned	None
11	View Project List (Admin)	Admin can view list of projects	Projects displayed	None
12	Logout Functionality	User logs out and is redirected to the login page	Logout successful	None

PROJECT MANAGEMENT RELATED STATISTICS

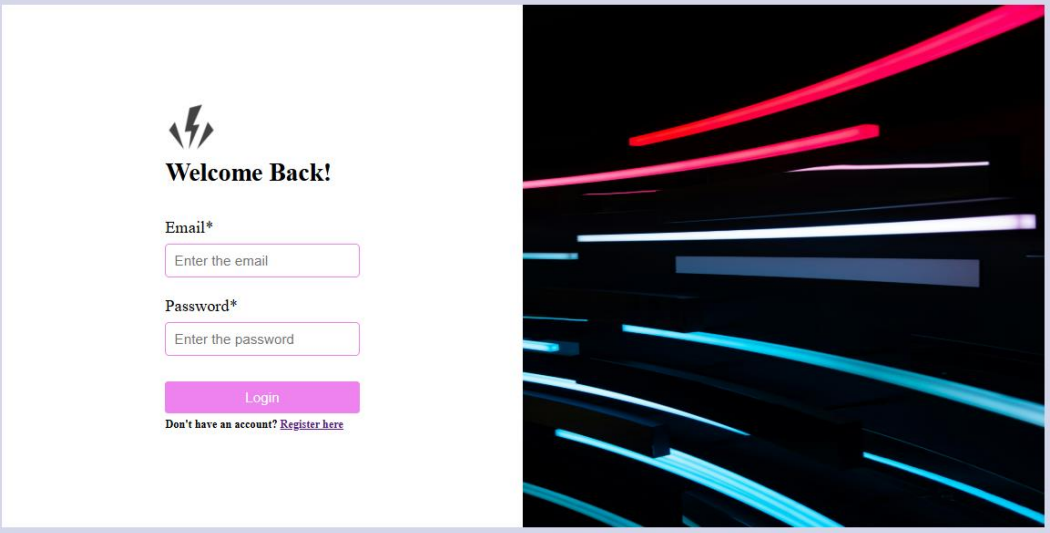
DATE	TASK PERFORMED	TASK PHASE	ADDITIONAL NOTES
November 11, 2024	Task Assignment: Requirements Gathering	Planning	Employee met with the client representative to gather requirements.
November 17, 2024	Task Assignment: SRS Document Preparation	Analysis	Initial SRS document drafted for client review.
November 30, 2024	Task Assignment: Database Design	Design	Database schema designed and finalized.
December 5, 2024	Task Assignment: UML Diagrams Creation	Design & Development	Use cases, class diagrams, and interface prototypes prepared.
December 16, 2024	Task Assignment: Code Module Development	Development	Module coding started; initial components developed.
December 17, 2024	Task Assignment: Code Module Testing	Development	Unit testing in progress on code modules.
December 18, 2024	Task Assignment: Web Application Implementation	Implementation	Offline application functionalities implemented.
December 19, 2024	Task Assignment: Window Application Implementation	Implementation	Setup for window-based application started.
December 21, 2024	Task Assignment: Integration & Testing	Testing	Integration testing initiated; unit tests ongoing.
December 28, 2024	Task Assignment: Final Validation	Quality Assurance	Final validations performed; all functionalities verified.
February 5, 2024	Task Assignment: Team Review and Feedback	Review	Task reviewed by team leads; feedback provided for improvements.
February 7, 2024	Task Assignment: Error Rectification	Debugging	Identified issues corrected; task updated.
February 9, 2024	Task Assignment: Final Adjustments	Finalization	Final modifications implemented; task marked complete.
February 10, 2024	Task Assignment: Project Submission	Submission	Task and project submitted to management for final approval.

UI Screenshots

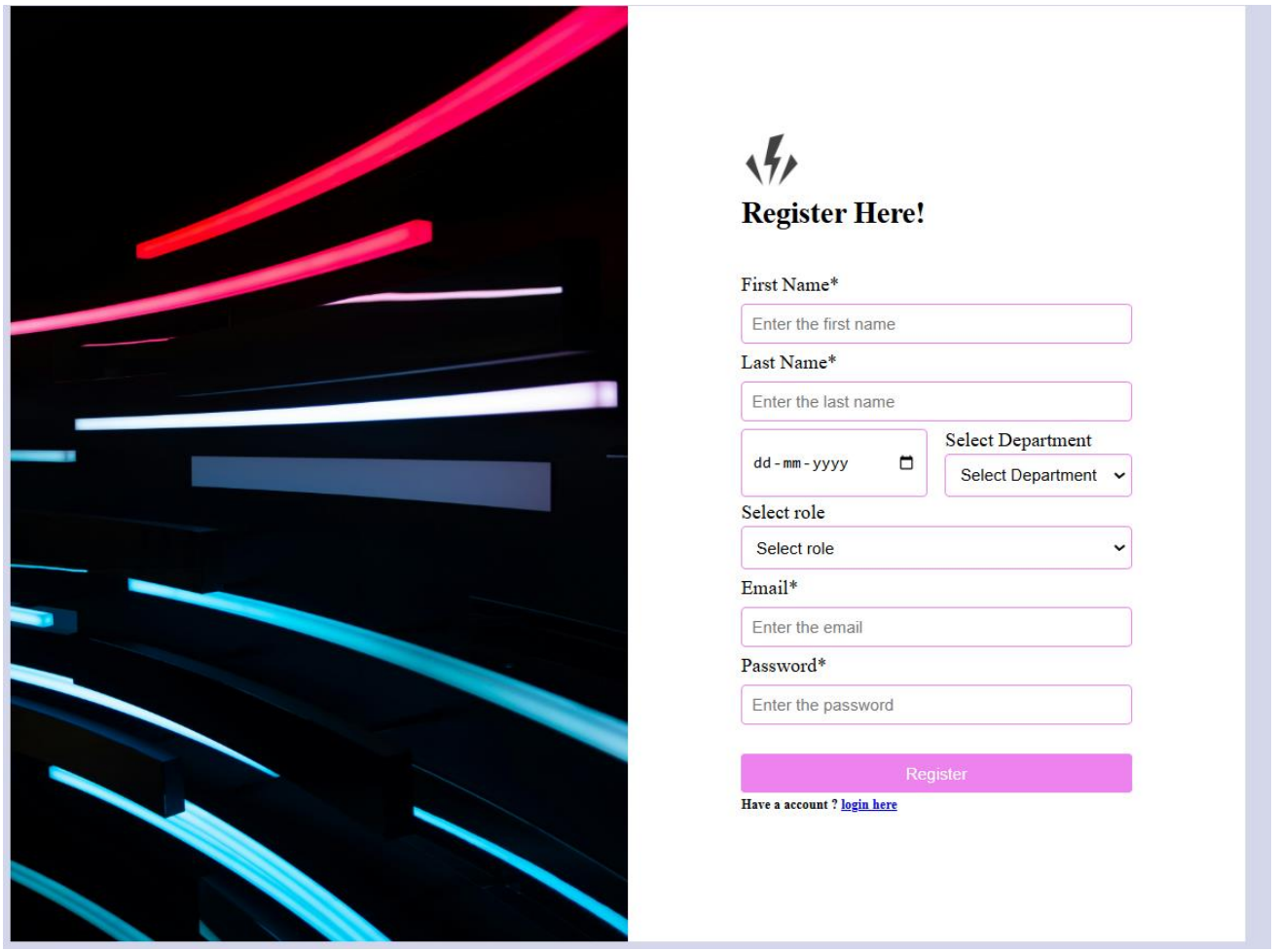
Home Page :



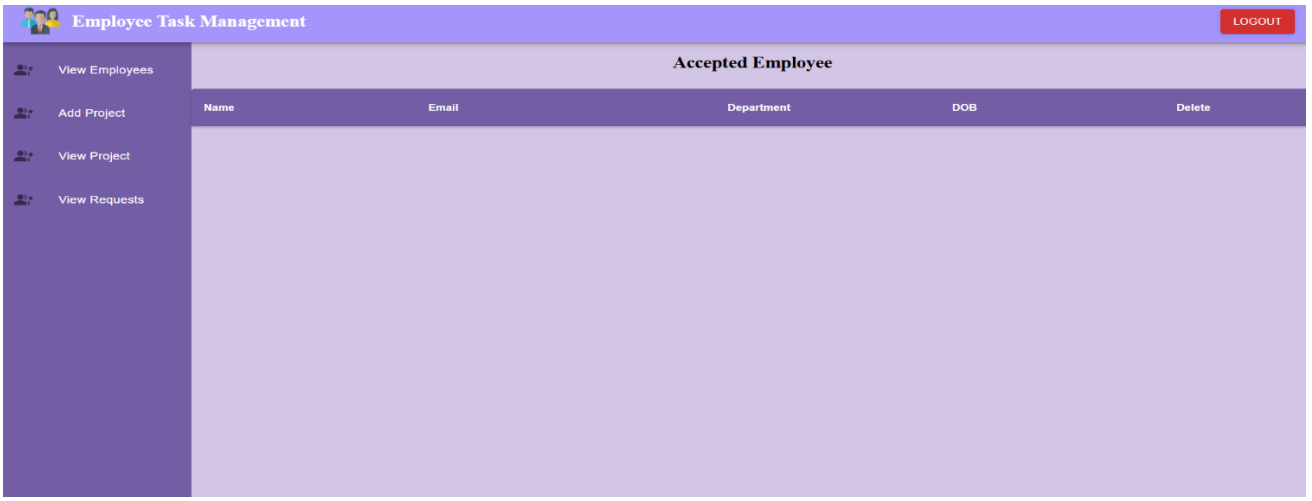
Login Page:



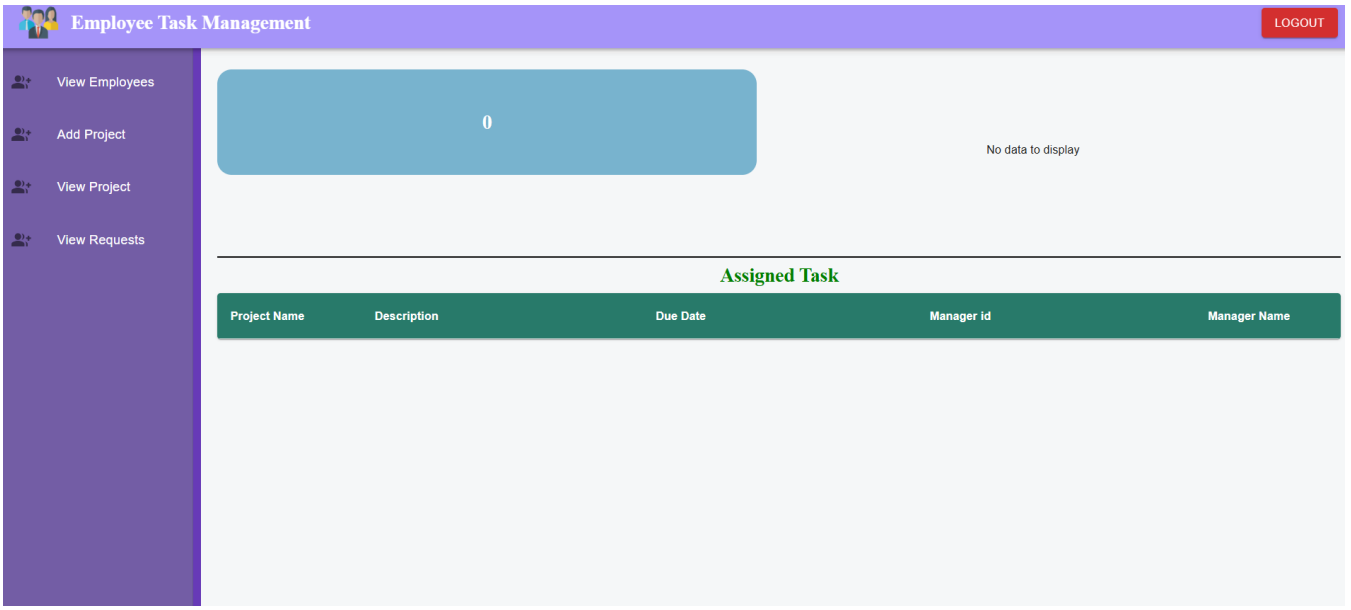
Register Page:



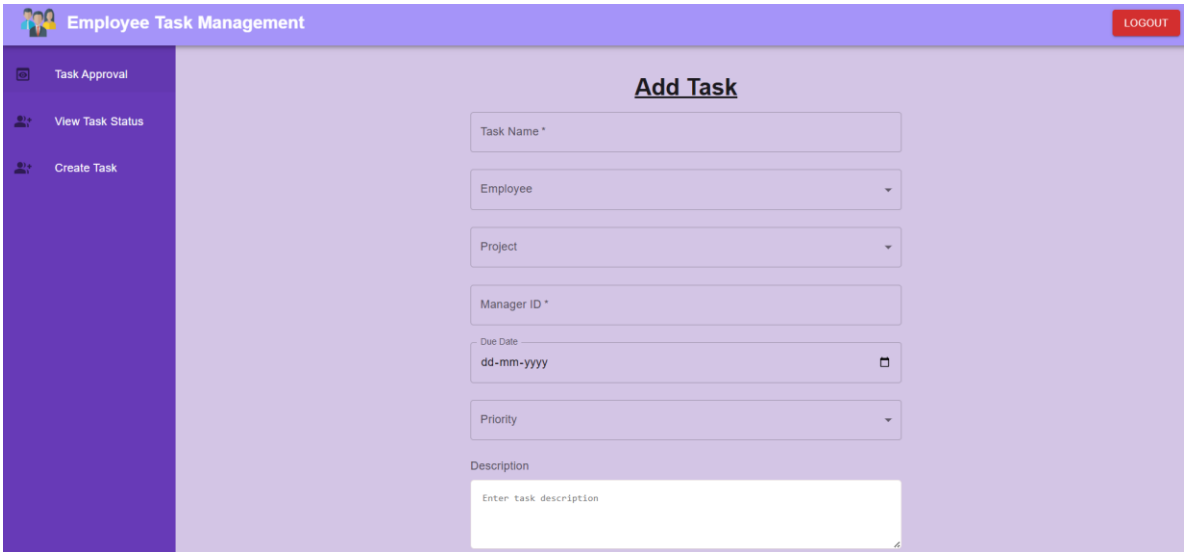
Admin Pages:

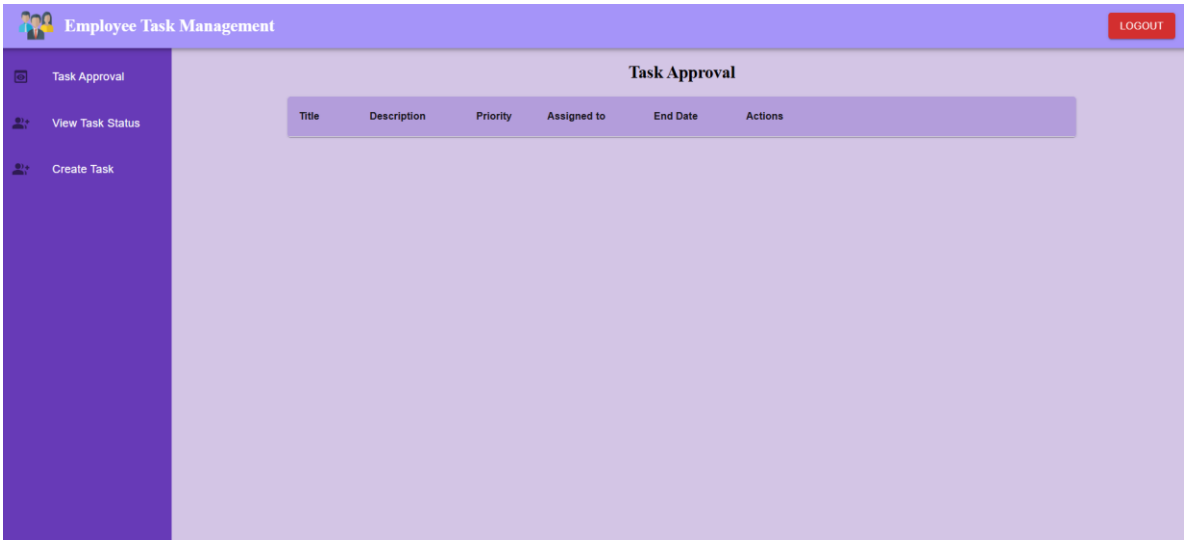


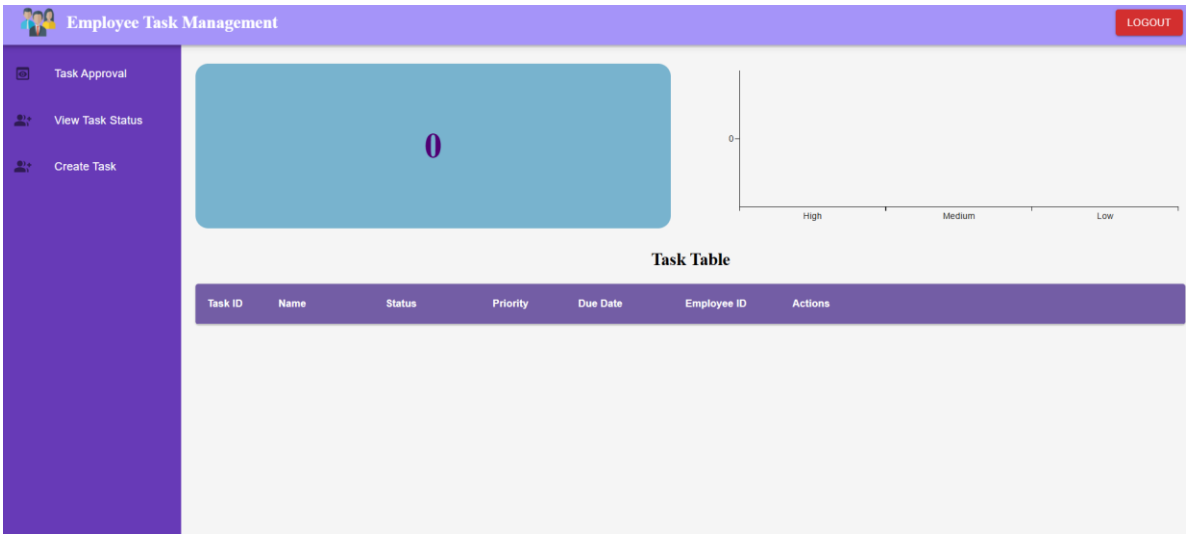

Employee Task Management
LOGOUT
View Employees
Add Project
View Project
View Requests
View Employee
Name
Email
Department
DOB
Accept
Reject


Employee Task Management
LOGOUT
View Employees
Add Project
View Project
View Requests
0
No data to display
Assigned Task
Project Name
Description
Due Date
Manager id
Manager Name

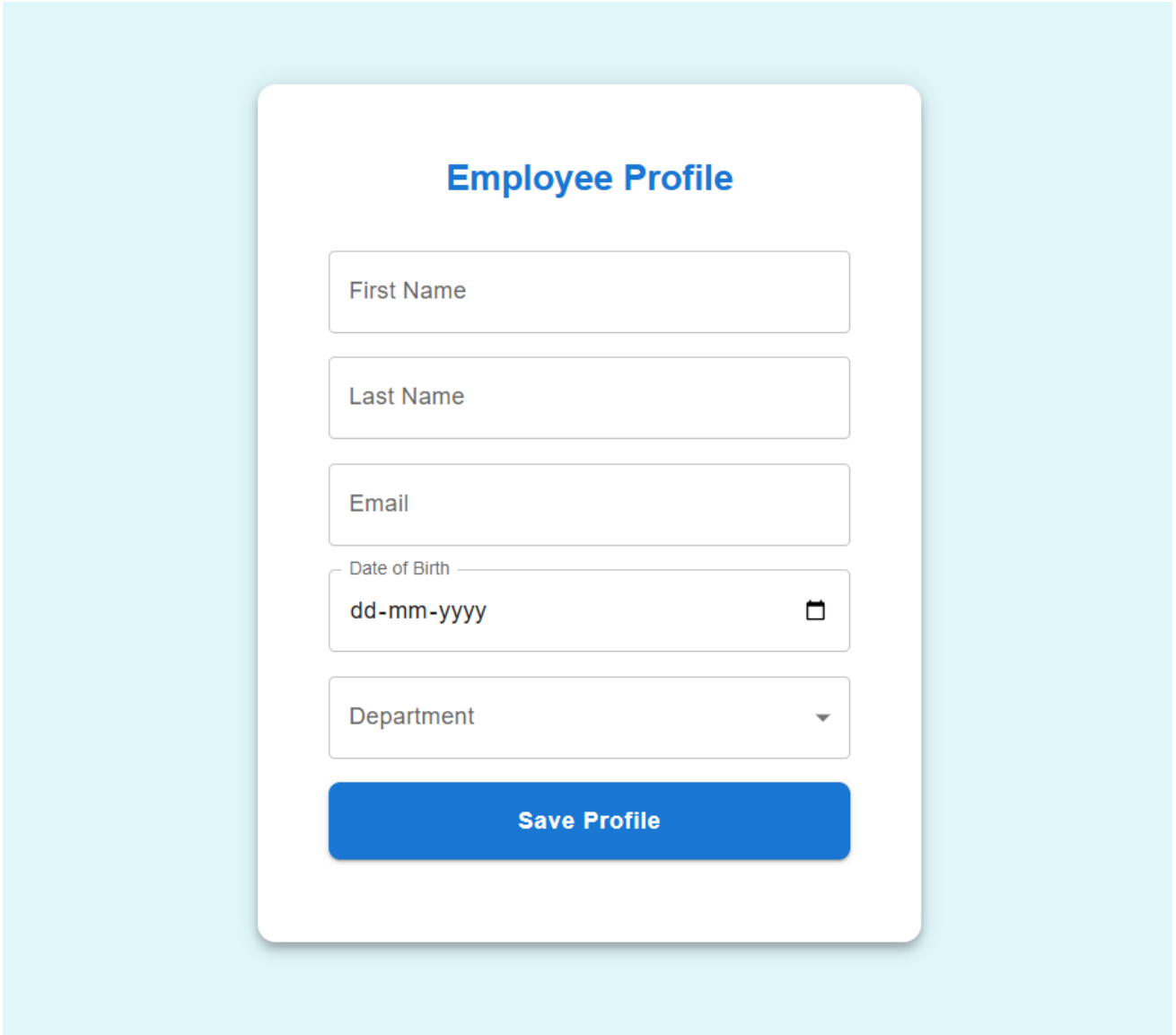
Manager Pages:







Employee Pages:



REFERENCES

1. **Spring Boot Documentation**
   URL: https://spring.io/projects/spring-boot

2. **React.js Documentation**
   URL: https://reactjs.org/docs/getting-started.html

3. **Redux Documentation**
   URL: https://redux.js.org

4. **Java Programming Language**
   URL: https://www.oracle.com/java/

5. **MySQL Workbench Documentation**
   URL: https://dev.mysql.com/doc/workbench/en/

6. **Spring Boot with React and Redux**
   URL: https://www.baeldung.com/spring-boot-react-and-redux

7. **Java Persistence API (JPA) Documentation**
   URL: https://www.eclipse.org/eclipselink/documentation/2.7/

8. **Swagger Documentation for Spring Boot**
   URL: https://springdoc.org/

9. **MDN Web Docs**
   URL: https://developer.mozilla.org/

10. **React Redux Integration Guide**
    URL: https://react-redux.js.org