Employee Management System

(PSST06)

Team name: **Quad Squads**

Team code: **ST 04**

Project Overview

The Employee Management Project is a comprehensive system designed to streamline and automate various aspects of managing employees within an organization. It provides a centralized platform to efficiently handle employee information, department details, Employee-Department assignment and Employee Promotion. The project aims to enhance productivity, improve HR processes, and promote a seamless experience.

Problem Statement

Design and implement an Employee Management System with the following features and functionalities:

1. Employee Details: Store employee information in a database table and provide REST API endpoints for data entry and retrieval. The employee details should include attributes such as employee ID, name, email, contact number, date of joining, and years of experience.

2. Department Details: Maintain department information in a separate database table and expose REST API endpoints for CRUD operations on department data. The department details should include attributes such as department ID, name, location, and manager ID.

3. Employee-Department Assignment: Provide functionality to assign an employee to a department. Each employee can be assigned to only one department at a time. Implement REST API endpoints to facilitate this assignment process.

4. Employee Promotion: Include an option to promote eligible employees to manager positions. To be considered for a manager role, an employee must have a minimum of 5 years of experience. Use the employee's date of joining to validate this criteria. Implement REST API endpoints to handle the promotion functionality.

5. Git Repository and Code Sharing: Create a GitHub profile (if not already available) and push the code for the Employee Management System to the repository. Share the git URL with the organizers after pushing the code.

Solution Description

Database Design:

Created a relational database schema with tables for employee details and department details.

The employee have columns such as employee ID, name, email, contact number, date of joining promoted date.

The department table have columns such as department ID, name, location, and manager ID.

Established a one-to-many relationship between the department table and the employee table using the manager ID column.

REST API Endpoints:

Implemented REST API endpoints using a framework like Flask or Django to handle data entry and retrieval for employee details.

Created endpoints for CRUD operations on employee details, such as creating a new employee, retrieving employee information, updating employee details, and deleting an employee.

Implemented REST API endpoints for CRUD operations on department details, such as creating a new department, retrieving department information, updating department details, and deleting a department.

Employee-Department Assignment:

Implemented an endpoint to assign an employee to a department.

This endpoint takes the employee ID and department ID as input and update the manager ID column in the department table accordingly.

Ensured that each employee can be assigned to only one department at a time.

Employee Promotion:

Implemented an endpoint to handle employee promotions.

This endpoint validates the employee's years of experience by comparing the date of joining with the current date.

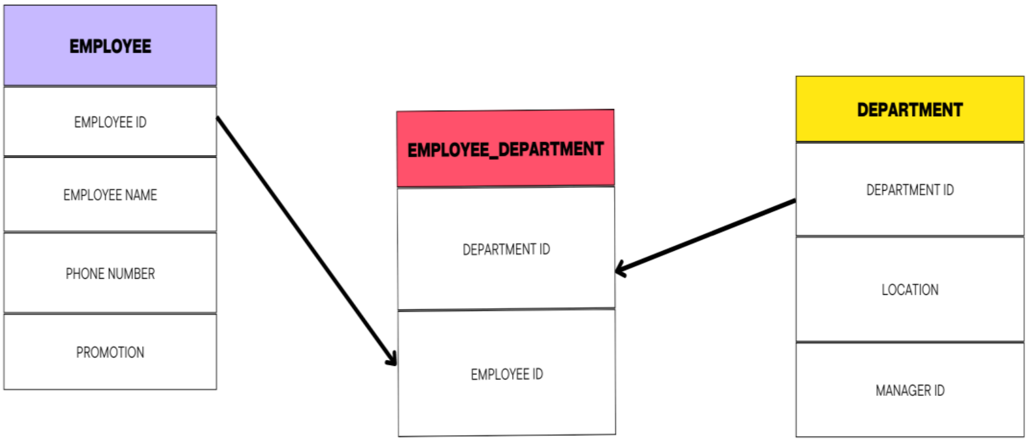
If the employee meets the promotion criteria (e.g., minimum 5 years of experience), update their role to manager in the employee table and the manager ID in the department table.

Technologies Used

- Django
- REST-API Framework
- Python
- HTML

Architecture and Design

ENTITY RELATIONSHIP DIAGRAM



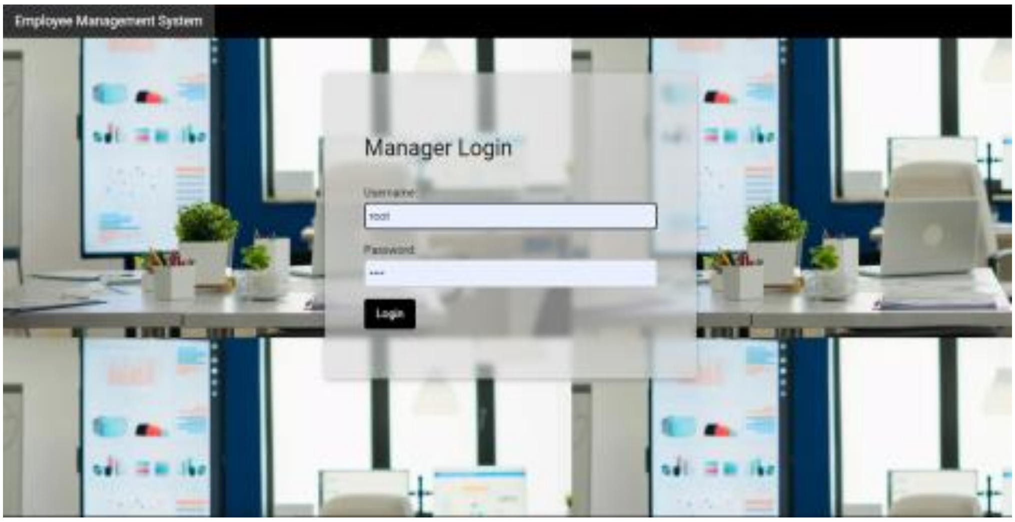
Implementation Details

- **Technology Stack:**
  Backend Framework: Django
  Database: SQLite3
  Frontend: HTML, CSS
- **System Architecture:**
  The Employee Management System follows a client-server architecture. The Django framework serves as the backend server, handling requests and responses, while the frontend is built using HTML and CSS for user interface rendering.
- **Database Design:**
  Database Management System: SQLite3
- **Tables:**
  Design and implement database tables to store employee information, attendance records, leave requests, performance evaluations, and other relevant data. Ensure appropriate relationships and indexes for efficient data retrieval and updates.
- **User Authentication and Authorization:**
  Implement user authentication and authorization mechanisms using Django's built-in authentication system. This ensures secure access to the system and restricts user actions based on their roles and permissions.
- **API Design and Development:**
  Build RESTful APIs using Django's REST framework to expose functionality for creating, reading, updating, and deleting employee information, attendance records, leave requests, etc. Design the API endpoints and data structures to align with the system's requirements and ensure consistency.
- **Employee Information Management:**
  Develop functionalities to handle employee information, including CRUD (Create, Read, Update, Delete) operations. Implement APIs for adding new employees, retrieving employee details, updating employee information, and deleting employee records.

User Interface/User Experience

The Employee Management System features a well-designed User Interface (UI) built using HTML and CSS, prioritizing usability and visual aesthetics. The UI incorporates a clean and intuitive layout, allowing users to navigate seamlessly through the system's functionalities. Attention was given to organizing information effectively, providing clear and concise labels, and maintaining consistency across the interface.

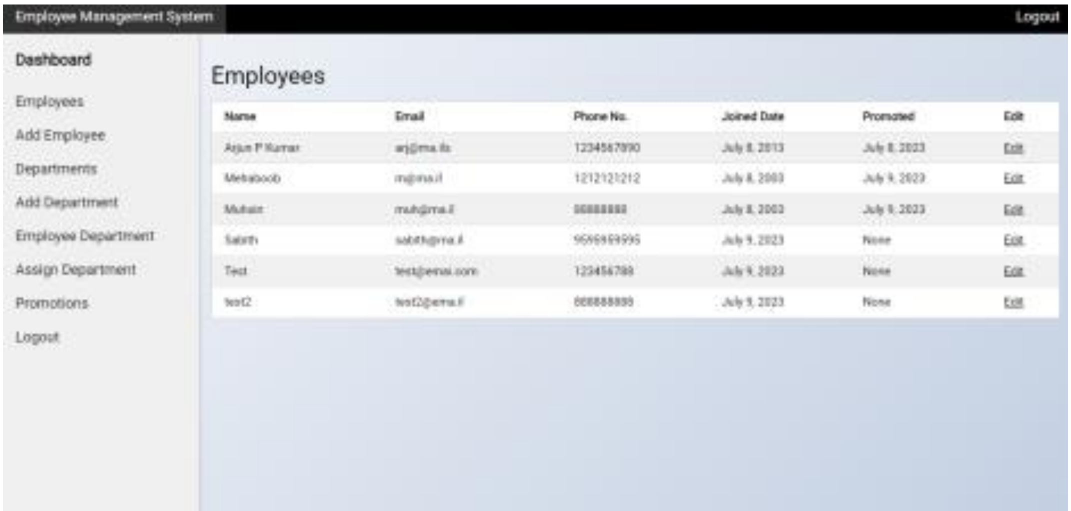
DEMO RESULT



Username: root

Password: ****

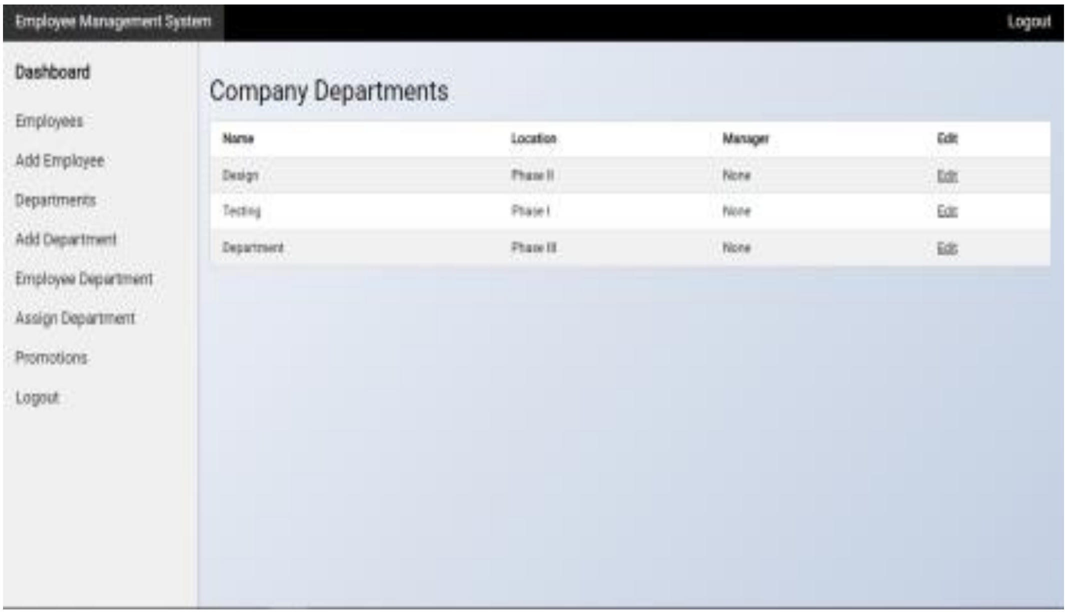
Manager login page



Employees

Name	Email	Phone No.	Joined Date	Promoted	Edit
Arjun P Kumar	arj@ma.ils	1234567890	July 8, 2013	July 8, 2023	Edit
Mehaboob	m@ma.il	1212121212	July 8, 2003	July 9, 2023	Edit
Muhsin	muh@ma.il	88888888	July 8, 2003	July 9, 2023	Edit
Sabith	sabith@ma.il	9696969996	July 9, 2023	None	Edit
Test	test@emai.com	123456789	July 9, 2023	None	Edit
test2	test2@ema.il	888888888	July 9, 2023	None	Edit

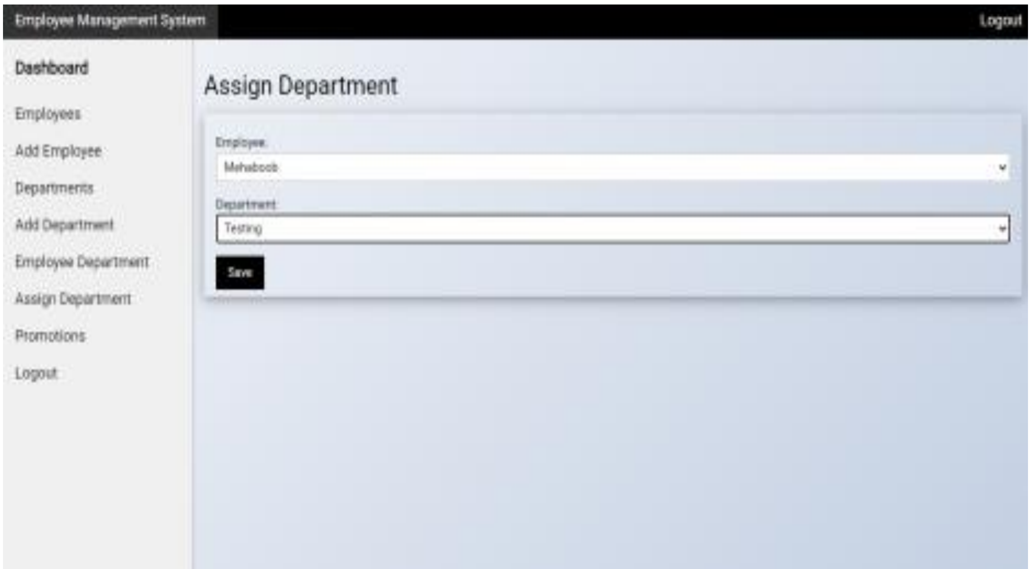
Employee details



Company Departments

Name	Location	Manager	Edit
Design	Phase II	None	Edit
Testing	Phase I	None	Edit
Department	Phase III	None	Edit

Department details

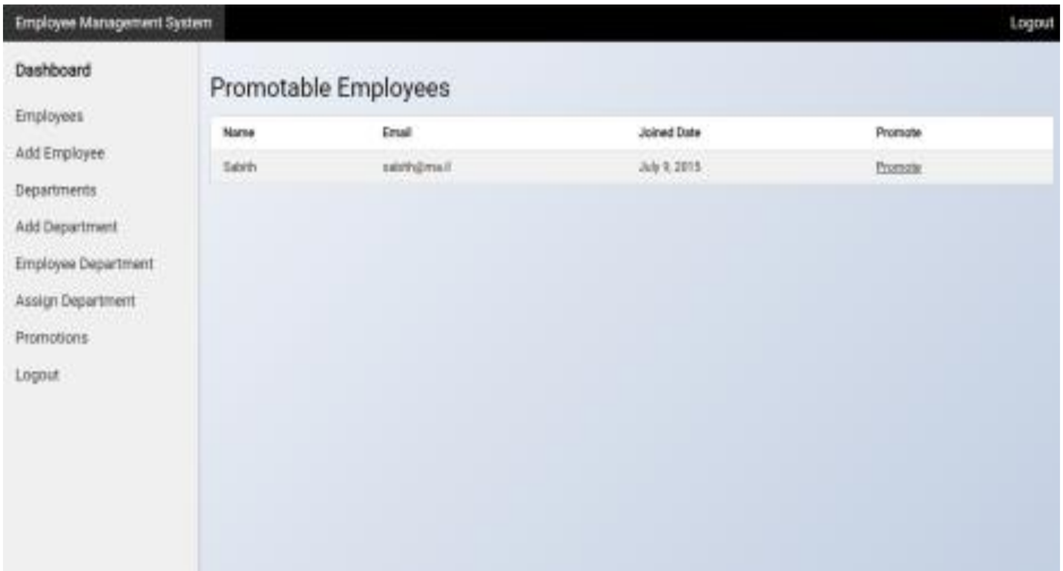


Assign Department

Employee: Mehaboob

Department: Testing

Assigning departments



Promotable Employees

Name	Email	Joined Date	Promote
Sabith	sabith@mail	July 9, 2015	Promote

Employee promotion

Conclusion

In conclusion, the Employee Management System has been successfully developed to streamline and automate various aspects of managing employees within our organization. The system's comprehensive functionalities, coupled with its intuitive user interface and seamless user experience, make it a valuable tool for managers and administrators alike. The implementation of the Employee Management System using Django, REST API, SQLite3, HTML, and CSS has proved to be reliable and efficient. The chosen technologies provided a solid foundation for the system, ensuring security, scalability, and ease of maintenance. As we move forward, the Employee Management System will continue to undergo regular updates and enhancements to adapt to evolving organizational needs. The project serves as a strong foundation for further development and optimization, ensuring that organization remains equipped with a robust and efficient solution for managing our valuable workforce.

Contact Information

Sabith B S Kumar : sabithbskumar@gmail.com

Mohammed Mehaboob S S : mohdmehaboob225@gmail.con

Muhammen Muhsin M S : muhammedmuhsin967@gmail.com

Arjun P Kumar : arjunpkumar15@gmail.com