

Leave Management System

Capstone Project Case Study

By

HARSHITA SRI R

Leave Management System

Introduction

Traditional leave management processes, often reliant on physical forms, scattered email chains, or error-prone spreadsheets, present significant operational inefficiencies and a notable lack of transparency for modern organizations. This manual and fragmented approach typically results in prolonged approval cycles, frequent miscommunication between staff and management, and considerable administrative overhead in tracking and calculating leave balances. Furthermore, the absence of a centralized, auditable record system complicates reporting and compliance.

The development of this system followed a structured methodology, beginning with a thorough analysis of user requirements and culminating in a functional prototype. Key considerations throughout the process included ensuring data security through role-based access control, designing an intuitive user experience to encourage adoption, and building a scalable architecture to accommodate future enhancements such as calendar integrations, automated balance accruals, and multi-level approval chains.

Abstract

This report details the development of a web-based Leave Management System (LMS) designed to replace inefficient, manual leave-tracking methods. The system provides a centralized digital platform featuring distinct interfaces for employees and administrators. Employees can seamlessly submit leave requests and track their status and balances, while administrators can review, approve, reject, or hold applications through a dedicated dashboard. By automating the entire workflow, the LMS eliminates paperwork, reduces errors, ensures policy compliance, and provides real-time transparency. The project demonstrates a functional solution that enhances operational efficiency, improves communication, and simplifies leave administration for modern organizations.

Project Objectives:

1. **Automate Core Workflow:** To digitalize the end-to-end leave application and approval process, eliminating reliance on physical forms, emails, or spreadsheets.
2. **Enhance User Experience:** To provide an intuitive interface for employees to easily submit leave requests and view their real-time status and balance history.
3. **Empower Administrative Control:** To equip managers/admins with a centralized dashboard to efficiently review, approve, reject, or hold pending leave applications and manage user accounts.
4. **Ensure Accuracy & Transparency:** To maintain an accurate, automated record of all leave transactions, providing clear audit trails and preventing disputes over balances and approvals.
5. **Establish a Scalable Foundation:** To build a modular system with a secure REST API backend that can be extended with future features like calendar integration, multi-level approvals, and detailed reporting.

Client Requirements

Based on initial stakeholder consultations and analysis of existing manual processes, the following functional and non-functional requirements were defined for the Leave Management System (LMS).

Functional Requirements (What the system must do)

1. User Authentication & Role Management:

- The system must allow new users to register with details (name, email, password, role).
- The system must allow registered users to log in and log out securely.
- The system must enforce two distinct user roles: **Employee** and **Manager (Admin)**.
- System functionality and data access must be determined by the user's role.

2. Employee Portal Requirements:

- An employee must be able to submit a new leave application, providing details such as leave type, start date, end date, and reason.
- An employee must be able to view a list of all their own past and current leave requests.
- An employee must be able to see the current status (e.g., Pending, Approved, Rejected) of each request.

3. Manager/Admin Portal Requirements:

- A manager must be able to view a dashboard listing **all** leave requests from all employees.
- A manager must have the ability to **approve** or **reject** any pending leave request.
- A manager must be able to view and manage all user accounts (view list, update details, delete users).

Technologies and Tools Used

- **Programming Language:** Java – used for backend logic and application development
- **Framework:** Spring Boot – used to build the web application with MVC architecture and embedded Tomcat server
- **Database Connectivity:** JDBC (Java Database Connectivity) – used to connect the application with the MySQL database and perform CRUD operations
- **Database:** MySQL – used as the relational database to store employee and user information
- **Frontend Technologies:** HTML and CSS – used to design and style the user interface
- **IDE / Development Tool:** Spring Tool Suite (STS) / Eclipse – used for coding, debugging, and project management
- **Server:** Embedded Apache Tomcat – used to deploy and run the web application
- **Build Tool:** Maven – used for dependency management and project build automation
- **Browser:** Google Chrome – used for testing and accessing the application

Software Requirements:

- Operating System: Windows / Linux
- Programming Language: Java (JDK 8 or above)
- Framework: Spring Boot
- Database: MySQL
- Database Connectivity: JDBC (Java Database Connectivity)
- Frontend Technologies: HTML, CSS
- IDE / Development Tool: Spring Tool Suite (STS) / Eclipse
- Build Tool: Maven
- Server: Embedded Apache Tomcat
- Web Browser: Google Chrome

System Modules

The Leave Management System is architecturally divided into four core modules, each handling a distinct set of functionalities.

Module 1: Authentication & Authorization Module

- **Purpose:** To manage user identity and control system access.
- **Key Functions:**
 - User Registration (for Employees and Managers).
 - User Login and Logout (Session Management).
 - Role-based permission enforcement across the application.

Module 2: User Management Module (Admin)

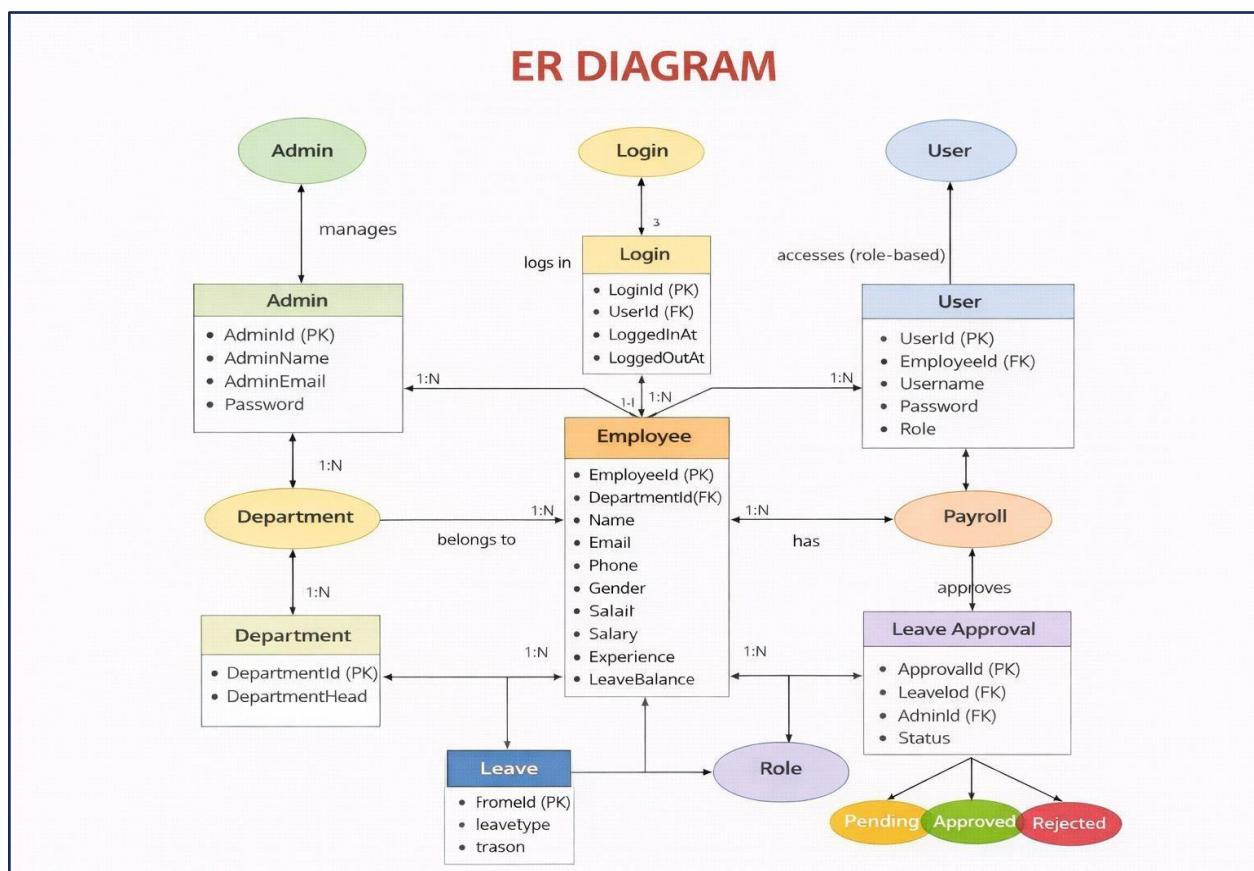
- **Purpose:** To allow administrators to oversee and manage all user accounts.
- **Key Functions:**
 - View a complete list of all system users.
 - Update user profiles (e.g., name, email, role).
 - Delete user accounts from the system.

Module 3: Employee Leave Module

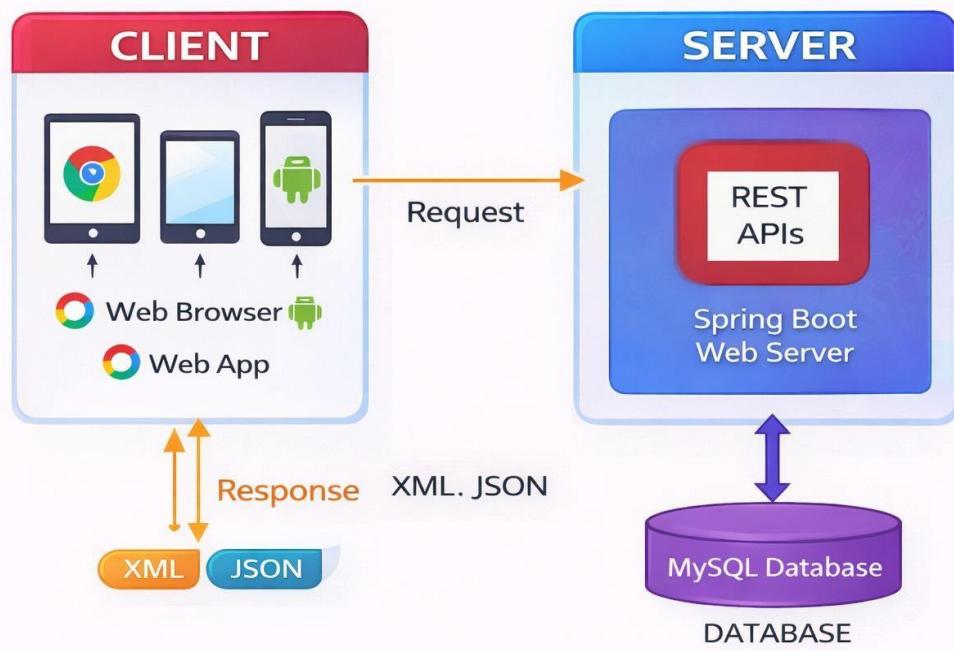
- **Purpose:** To provide employees with self-service leave management.
- **Key Functions:**
 - Submit a new leave application form.
 - View a personal dashboard listing all their own leave requests (history and current status).

Module 4: Manager Leave Dashboard Module

- **Purpose:** To provide managers with a control panel for overseeing the leave workflow.
- **Key Functions:**
 - View a centralized list of **all** leave requests from every employee.
 - Review pending requests and take action: **Approve** or **Reject**.

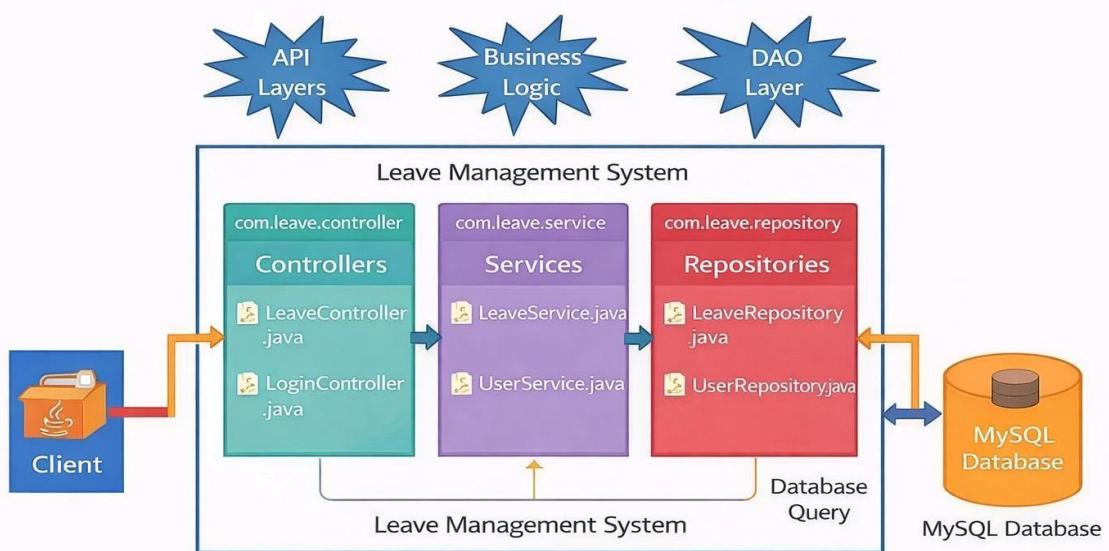


CLIENT – SERVER ARCHITECTURE



Spring Boot Architecture –

— LEAVE MANAGEMENT SYSTEM



Manager Module

The Manager Module is a secure, role-restricted control panel that empowers managers to oversee the entire leave ecosystem. It features a comprehensive User Management Dashboard to view, edit, and delete all system users, and a Leave Administration Dashboard to review every leave request, with direct action buttons to approve or reject applications. Additionally, it includes tools for configuring leave policies and generating organizational reports, all accessed through a centralized, intuitive interface that consolidates full administrative control.

Employee Module

The Employee Module is a self-service portal designed for staff to manage their leave simply and efficiently. It centers on a user-friendly Leave Application Form for submitting new requests and a personalized "My Leave" Dashboard to track the real-time status of all their applications. This focused interface provides employees with complete transparency and control over their own leave process

HTTP / HTTPS Request Methods

Login Module

HTTP Method	URL	Description
POST (Employee/Manager)	http://localhost:8084/auth/register	Registers a new user
POST	http://localhost:8084/auth/login	Authenticates user and starts session
GET	http://localhost:8084/auth/logout	Logout the current user

User Management Module

HTTP Method	URL	Description
GET	http://localhost:8084/users/all	Retrieves all registered users
POST	http://localhost:8084/users/update	Updates user details
POST	http://localhost:8084/users/change-password	Changes password for logged in users

Leave Management Module - Employee

HTTP Method	URL	Description
POST	http://localhost:8084/leave/apply	Employee applies for leave
GET	http://localhost:8080/leave/my leaves	Retrieves logged in employees leave

Leave Management Module - Manager

HTTP Method	URL	Description
GET	http://localhost:8084/leave/all	Retrieves all leave requests
GET	http://localhost:8084/leave/approve	Approves a leave requests

DATA DICTIONARY

DATABASE

MANAGER / EMPLOYEE DATABASE

The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Displays the following SQL code:

```

USE leave_db;
SELECT id, name, email, password, role
FROM users;
SELECT * FROM users;
SELECT * FROM LEAVES;
    
```
- Result Grid:** Shows a table with columns: id, name, email, password, and role. The data is as follows:

id	name	email	password	role
1	HARSHITA SRI R.	harshi@gmail.com	harshi	EMPLOYEE
2	aditi	aditi@gmail.com	aditi	MANAGER
6	HARSHITA SRI R.	harshitarsi.r@gmail.com	harshi	EMPLOYEE
7	ranjana	ranjana@gmail.com	ranjana	EMPLOYEE
8	Adithi L.	adithi@gmail.com	adithi	MANAGER
9	Vagisha D	vag@gmail.com	vaghi	EMPLOYEE
- Action Output:** Shows the history of actions taken in the database:

#	Time	Action	Message	Duration / Fetch
5	20:05:38	CREATE TABLE leaves (id INT AUTO_INCREMENT PRIMARY KEY, user_email VARCHAR(100) NOT NULL) ENGINE=InnoDB;	0 row(s) affected	0.078 sec
6	20:05:40	SHOW TABLES	2 row(s) returned	0.000 sec / 0.000 sec
7	20:05:43	DESC users	5 row(s) returned	0.000 sec / 0.000 sec
8	20:05:46	USE leave_db	0 row(s) affected	0.000 sec
9	20:05:48	SELECT id, name, email, password, role FROM users LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec
10	20:05:57	SELECT * FROM USERS LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec

EMPLOYEE LEAVE DATABASE

The screenshot shows the MySQL Workbench interface. In the top-left, the navigation pane includes sections for MANAGEMENT (Server Status, Client Connections, Users and Privileges, Status and System Variables, Data Export, Data Import/Restore), INSTANCE (Startup / Shutdown, Server Logs, Options File), and PERFORMANCE (Dashboard, Performance Reports, Performance Schema Setup). The central area displays a query editor with the following SQL code:

```
27  
28 •   SELECT id, name, email, password, role  
      FROM users;  
29  
30  
31 •   SELECT * FROM USERS;  
32 •   SELECT * FROM LEAVES;
```

Below the code, the Result Grid shows the following data:

ID	User Email	From Date	To Date	Leave Type	Reason	Status
1	ranu@gmail.com	2026-01-01	2025-12-02	CASUAL	Home Visit	PENDING
2	harsh@gmail.com	2026-01-17	2025-12-19	SICK	Dengue Fever	PENDING
3	NULL	NULL	NULL	NULL	NULL	NULL

The bottom section shows the Output tab with the following log entries:

Action	Time	Action	Message	Duration / Fetch
15	20:06:54	SELECT * FROM LEAVES LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
16	20:06:56	SELECT * FROM USERS LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec
17	20:06:58	SELECT id, name, email, password, role FROM users LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec
18	20:07:01	SELECT * FROM LEAVES LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
19	20:13:18	SELECT * FROM USERS LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec
20	20:17:43	SELECT * FROM LEAVES LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec

GET METHOD ON MANAGING LEAVES

The screenshot shows the Postman interface. The left sidebar displays a History section with several recent requests, including:

- GET http://localhost:8084/manage-leaves.html
- GET http://localhost:8084/manager/manage-users.html
- GET http://localhost:8084/manager/dashboard.html
- GET http://localhost:8084/register.html
- GET http://localhost:8084/login.html

The main panel shows a GET request to <http://localhost:8084/manager/manage-leaves.html>. The Request tab shows the method as GET and the URL as <http://localhost:8084/manager/manage-leaves.html>. The Headers tab shows the following headers:

- Content-Type: application/json
- Accept: */*

The Body tab shows the following JSON payload:

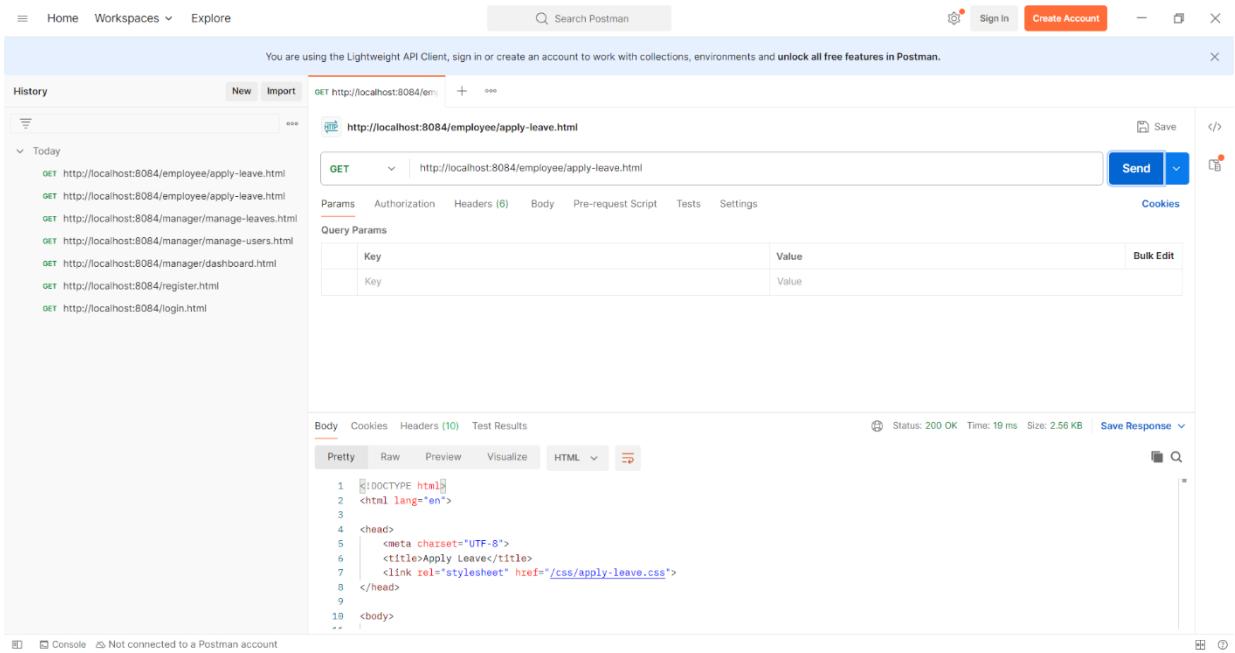
```
{ "leave_type": "SICK", "start_date": "2025-12-19", "end_date": "2025-12-20", "reason": "Dengue Fever", "status": "PENDING" }
```

The Response tab shows the following JSON response:

```
{ "id": 1, "user_email": "ranu@gmail.com", "from_date": "2026-01-01", "to_date": "2025-12-02", "leave_type": "CASUAL", "reason": "Home Visit", "status": "PENDING" }
```

The status bar at the bottom indicates: Status: 200 OK Time: 19 ms Size: 894 B Save Response.

POST METHOD ON APPLYING LEAVES



You are using the Lightweight API Client, sign in or create an account to work with collections, environments and unlock all free features in Postman.

History New Import

GET http://localhost:8084/employee/apply-leave.html

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value
Key	Value

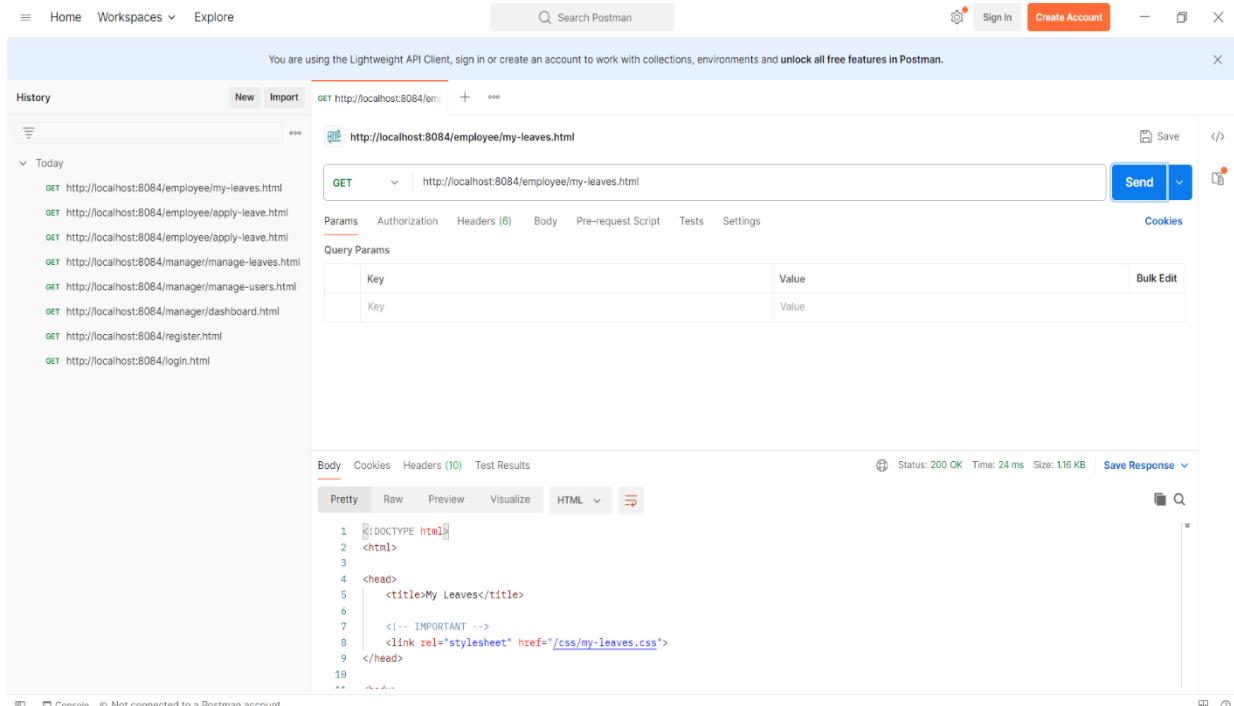
Body Cookies Headers (10) Test Results

Pretty Raw Preview Visualize HTML

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <title>Apply Leave</title>
7   <link rel="stylesheet" href="/css/apply-leave.css">
8 </head>
9
10 <body>
11
```

Status: 200 OK Time: 19 ms Size: 2.56 KB Save Response

GET METHOD ON MY LEAVES



You are using the Lightweight API Client, sign in or create an account to work with collections, environments and unlock all free features in Postman.

History New Import

GET http://localhost:8084/employee/my-leaves.html

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value
Key	Value

Body Cookies Headers (10) Test Results

Pretty Raw Preview Visualize HTML

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <title>My Leaves</title>
6
7   <!-- IMPORTANT -->
8   <link rel="stylesheet" href="/css/my-leaves.css">
9 </head>
10
11
```

Status: 200 OK Time: 24 ms Size: 1.16 KB Save Response

GET METHOD ON CHANGE PASSWORD

The screenshot shows the Postman interface. In the top navigation bar, there are links for Home, Workspaces, Explore, and a search bar labeled "Search Postman". On the right, there are buttons for "Sign In" and "Create Account". Below the header, a message says "You are using the Lightweight API Client, sign in or create an account to work with collections, environments and unlock all free features in Postman." The main area has a "History" sidebar on the left listing various API calls. The main workspace shows a "GET" request to "http://localhost:8084/employee/change-password.html". The "Params" tab is selected, showing a single query parameter "Key" with a value of "Value". A "Send" button is visible. Below the request, the "Body" tab shows the raw HTML response:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Change Password</title>
5 </head>
6 <body>
7
8   <h2>Change Password</h2>
9
10 </body>
11 </html>
```

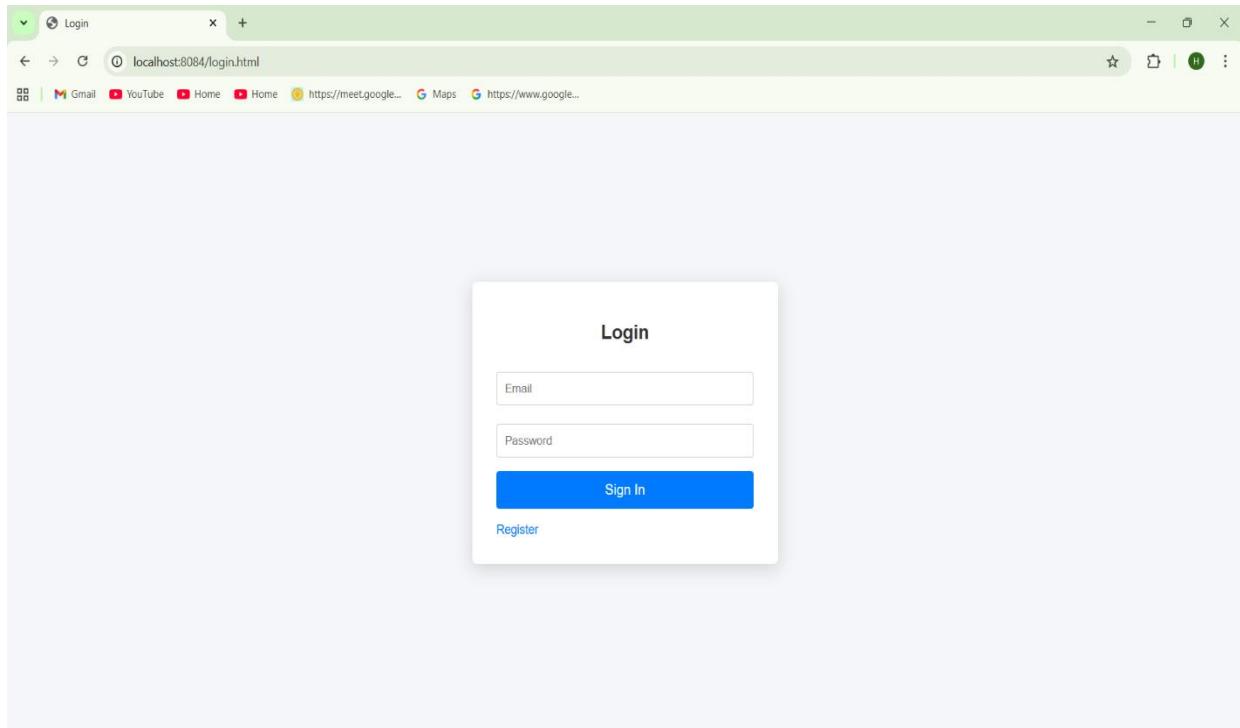
JUNITTEST CASES FOR MANAGER MODULE

The screenshot shows the Eclipse IDE interface with several open windows. The "Package Explorer" shows a Java project structure with classes like AuthController, LeaveController, my-leaves.html, my-leaves.js, UserController, and LeaveDAOTest. The "LeaveManagementApplicationTests [Runner: JUnit 5] (0.775 s)" window displays a green progress bar indicating successful execution. The "Outline" view shows a class hierarchy for com.example.leavemanagement, specifically focusing on the LeaveDAOTest class with a method named "testApplyLeave()". The "Code Editor" window contains the Java code for the "LeaveDAOTest" class, which includes imports for junit and assertions, and a test method "testApplyLeave()" that creates a new "Leave" object and asserts that the result of calling "LeaveDAO.applyLeave(leave)" is true, with the reason being "Leave should be applied successfully".

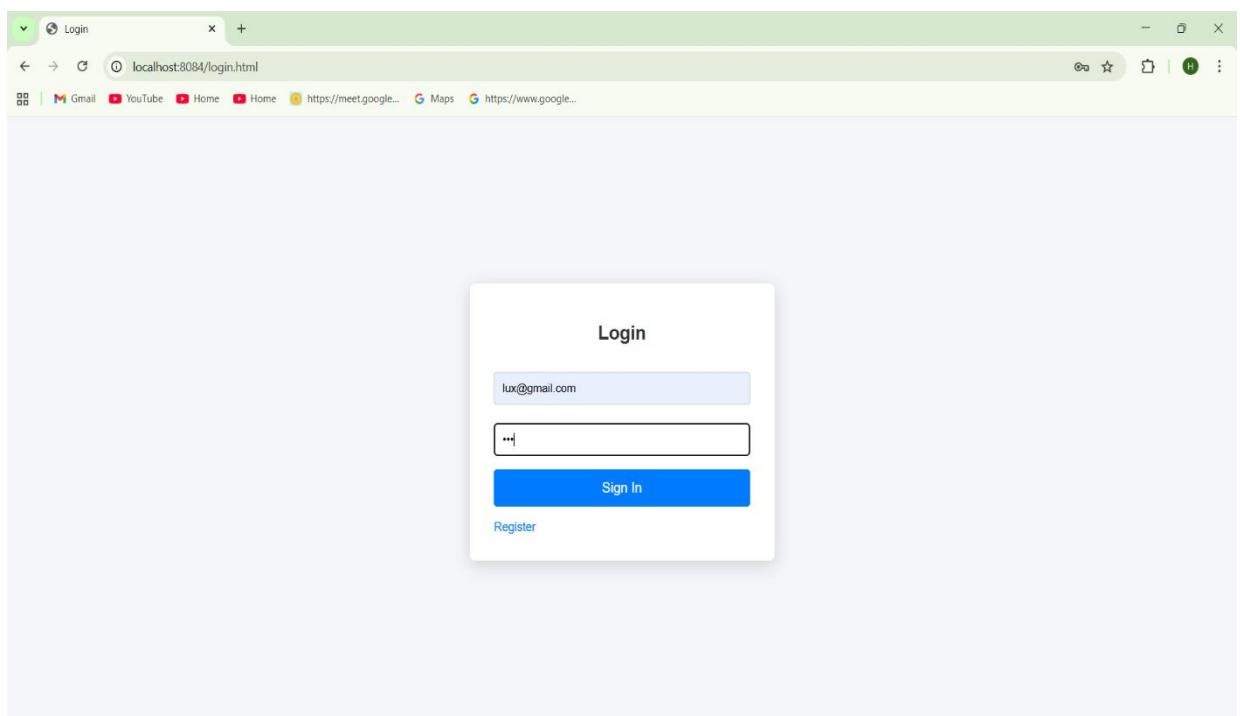
```
1 package com.example.leavemanagement;
2
3 import static org.junit.jupiter.api.Assertions.*;
4 import org.junit.jupiter.api.Test;
5
6 import com.example.leavemanagement.dao.LeaveDAO;
7 import com.example.leavemanagement.model.Leave;
8
9 public class LeaveDAOTest {
10
11     @Test
12     public void testApplyLeave() {
13
14         Leave leave = new Leave();
15         leave.setUserEmail("junituser@mail.com");
16         leave.setFromDate("2025-01-10");
17         leave.setToDate("2025-01-12");
18         leave.setLeaveType("CASUAL");
19         leave.setReason("JUnit testing");
20         leave.setStatus("PENDING");
21
22         boolean result = LeaveDAO.applyLeave(leave);
23
24         assertTrue(result, "Leave should be applied successfully");
25     }
26
27 }
```

PROJECT OUTPUTS

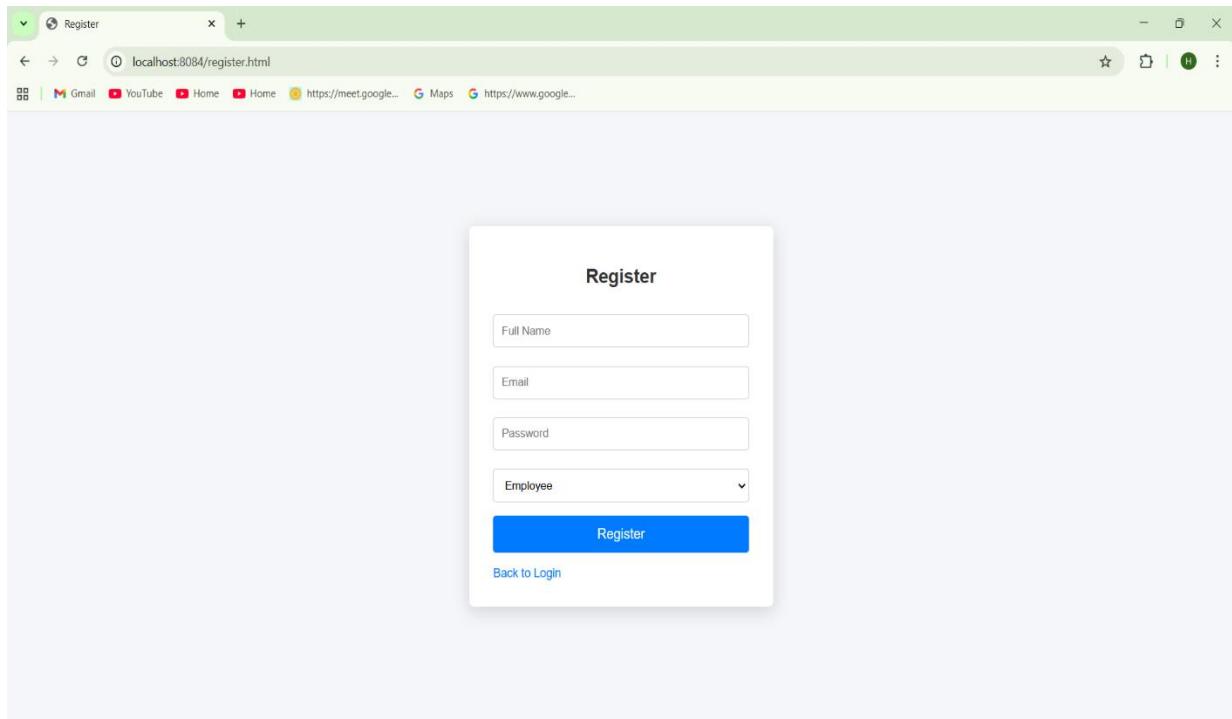
Login Page



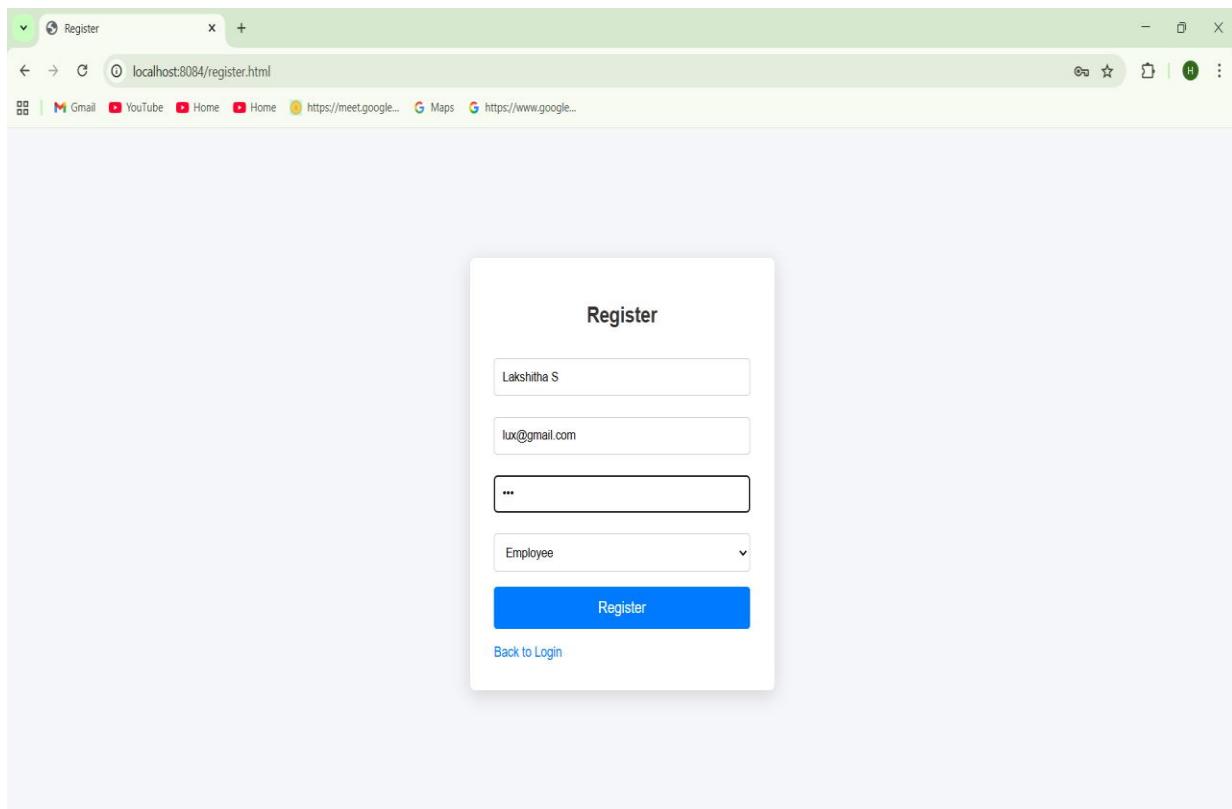
Employee's Login Page

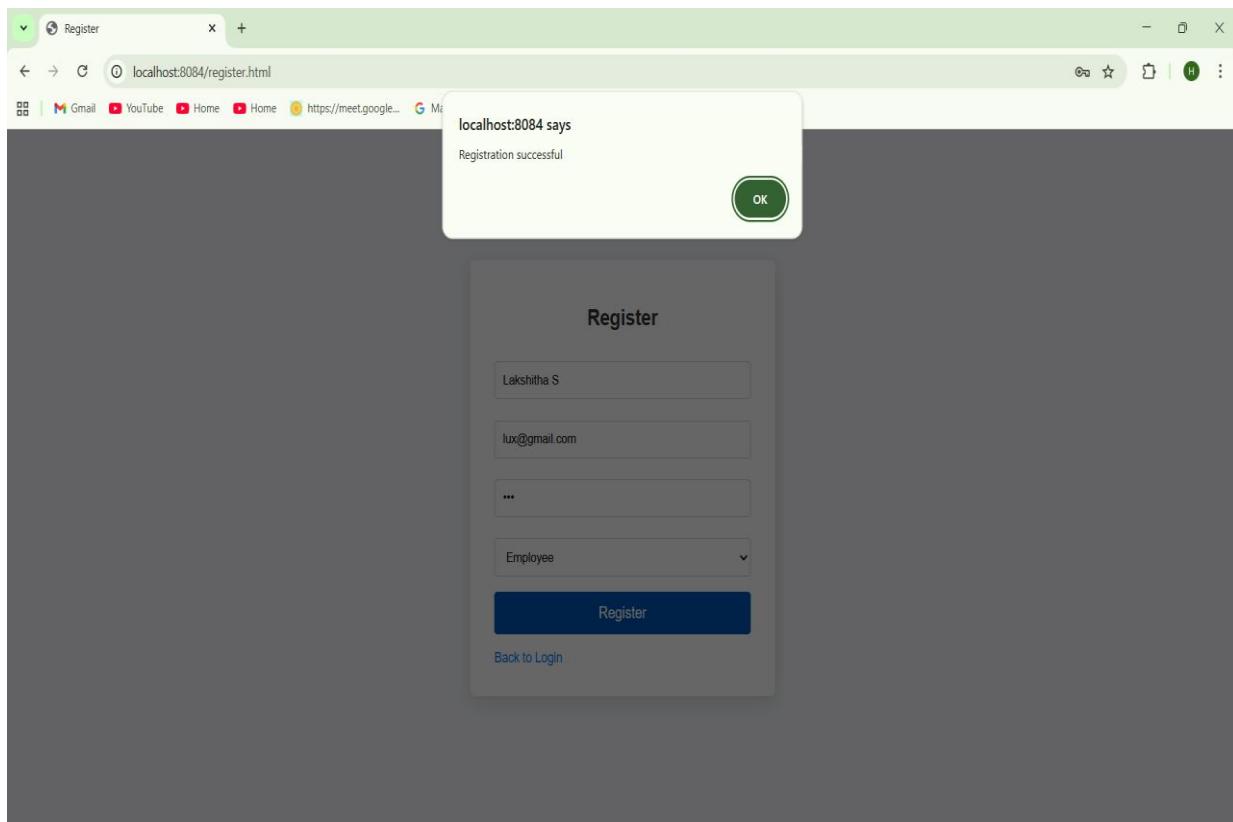


Registration Page

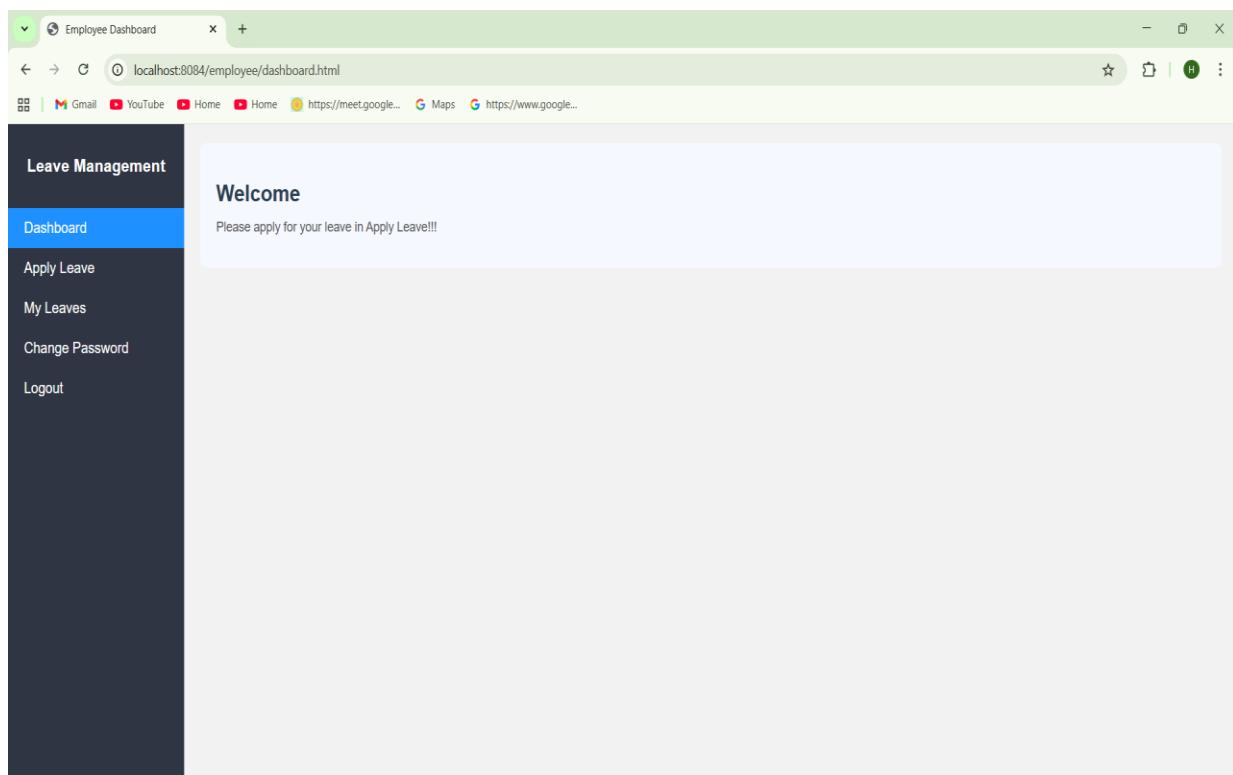


Employee's Register Page





Employee's Dashboard Page

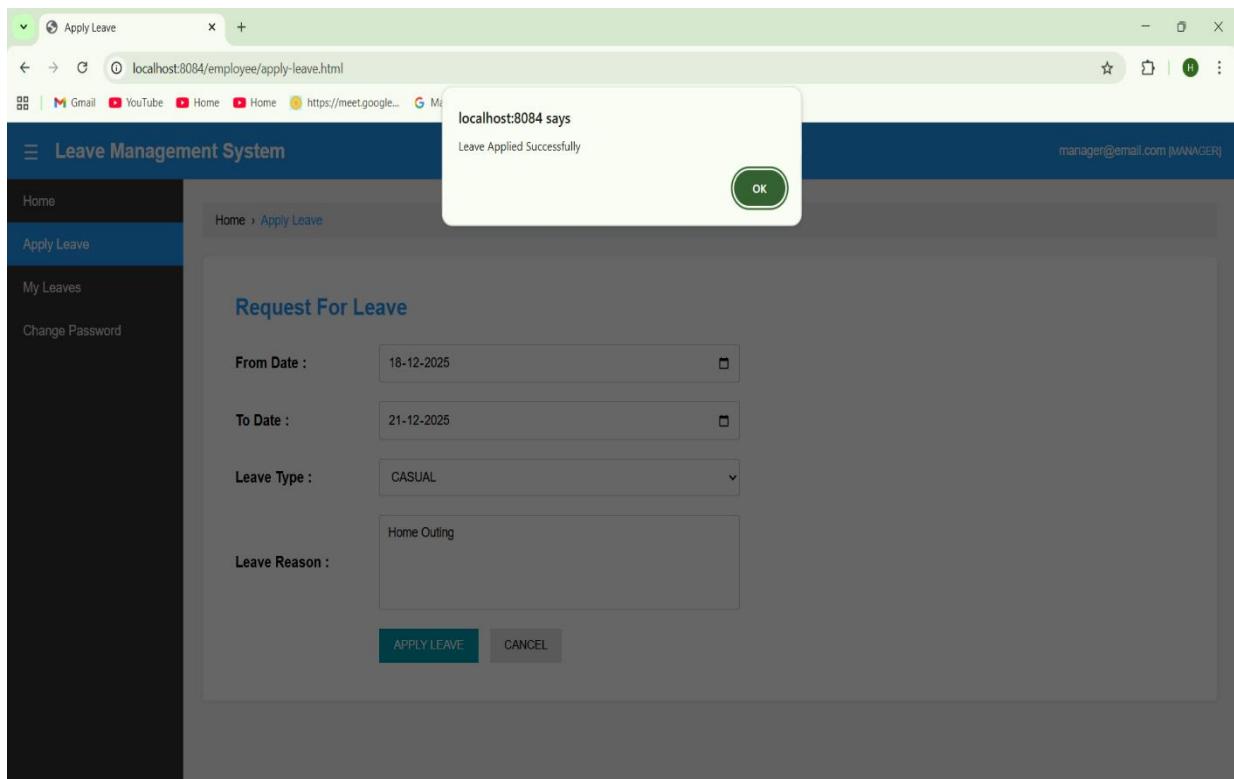


Request For Leave

The screenshot shows a web browser window titled "Apply Leave" with the URL "localhost:8084/employee/apply-leave.html". The page is part of the "Leave Management System" and is titled "Request For Leave". On the left, there is a sidebar with links: Home, Apply Leave (which is selected and highlighted in blue), My Leaves, and Change Password. The main content area shows a form with fields for "From Date" (dd-mm-yyyy), "To Date" (dd-mm-yyyy), "Leave Type" (a dropdown menu showing "SELECT"), and "Leave Reason" (a text area). Below the form are two buttons: "APPLY LEAVE" (in blue) and "CANCEL". The top right corner of the page shows the email "manager@email.com [MANAGER]".

Employee's Leave Request

The screenshot shows a web browser window titled "Apply Leave" with the URL "localhost:8084/employee/apply-leave.html". The page is part of the "Leave Management System" and is titled "Request For Leave". On the left, there is a sidebar with links: Home, Apply Leave (selected and highlighted in blue), My Leaves, and Change Password. The main content area shows a form with fields for "From Date" (18-12-2025), "To Date" (21-12-2025), "Leave Type" (CASUAL), and "Leave Reason" (Home Outing). Below the form are two buttons: "APPLY LEAVE" (in blue) and "CANCEL". The top right corner of the page shows the email "manager@email.com [MANAGER]".



Employee's - My Leaves Page

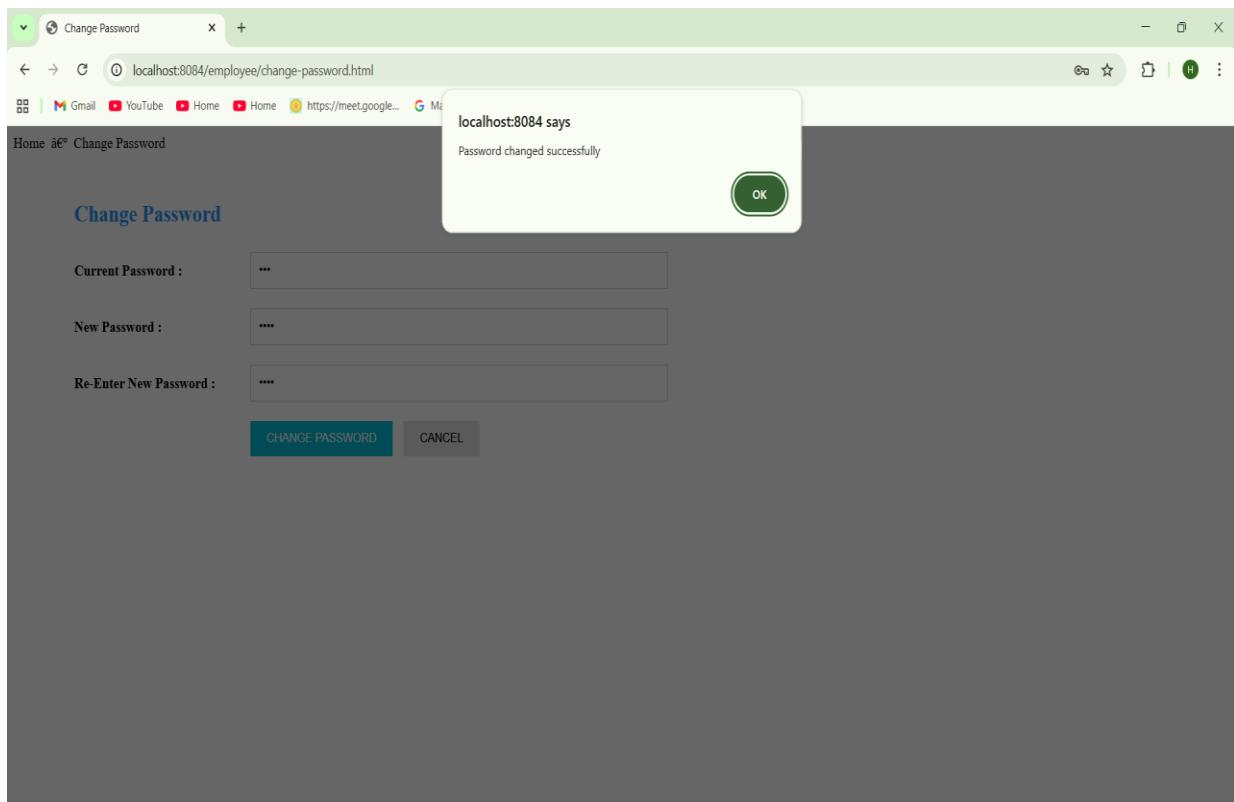
FROM DATE	TO DATE	LEAVE TYPE	REASON	STATUS
2025-12-18	2025-12-21	CASUAL	Home Outing	PENDING

Change Password Page

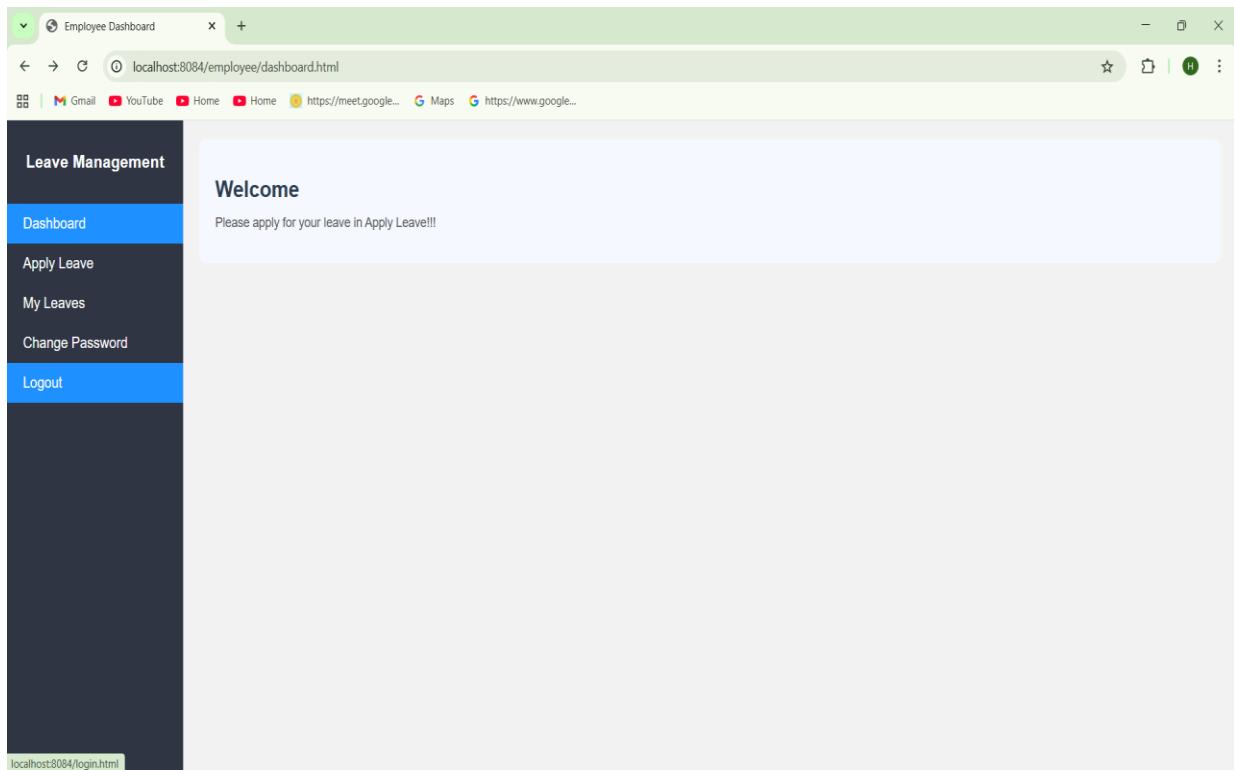
The screenshot shows a web browser window with the title bar "Change Password". The address bar displays "localhost:8084/employee/change-password.html". Below the address bar, there are several icons for various services like Gmail, YouTube, and Google Maps. The main content area is titled "Change Password". It contains three input fields: "Current Password:", "New Password:", and "Re-Enter New Password:". Below these fields are two buttons: a blue "CHANGE PASSWORD" button and a grey "CANCEL" button.

Changing New Password

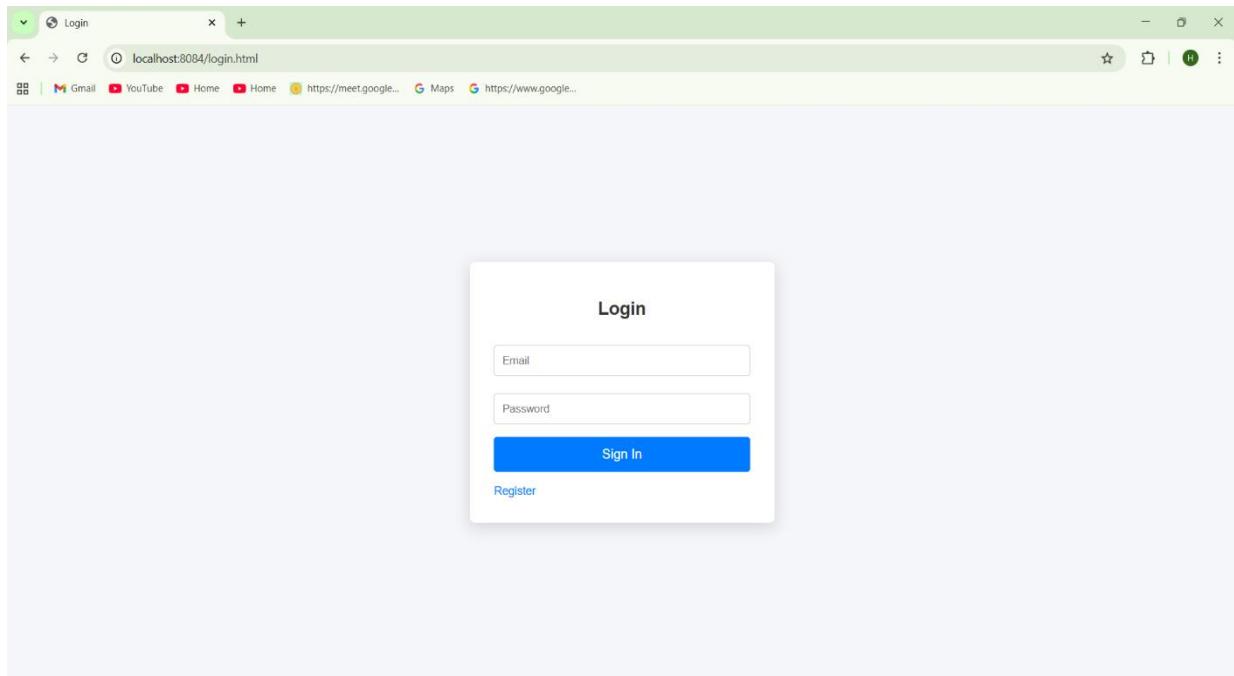
This screenshot is identical to the one above, but it shows that the "New Password:" and "Re-Enter New Password:" fields now contain the value "....". All other elements, including the "Current Password:" field which still contains "..." and the "CHANGE PASSWORD" and "CANCEL" buttons, remain the same.



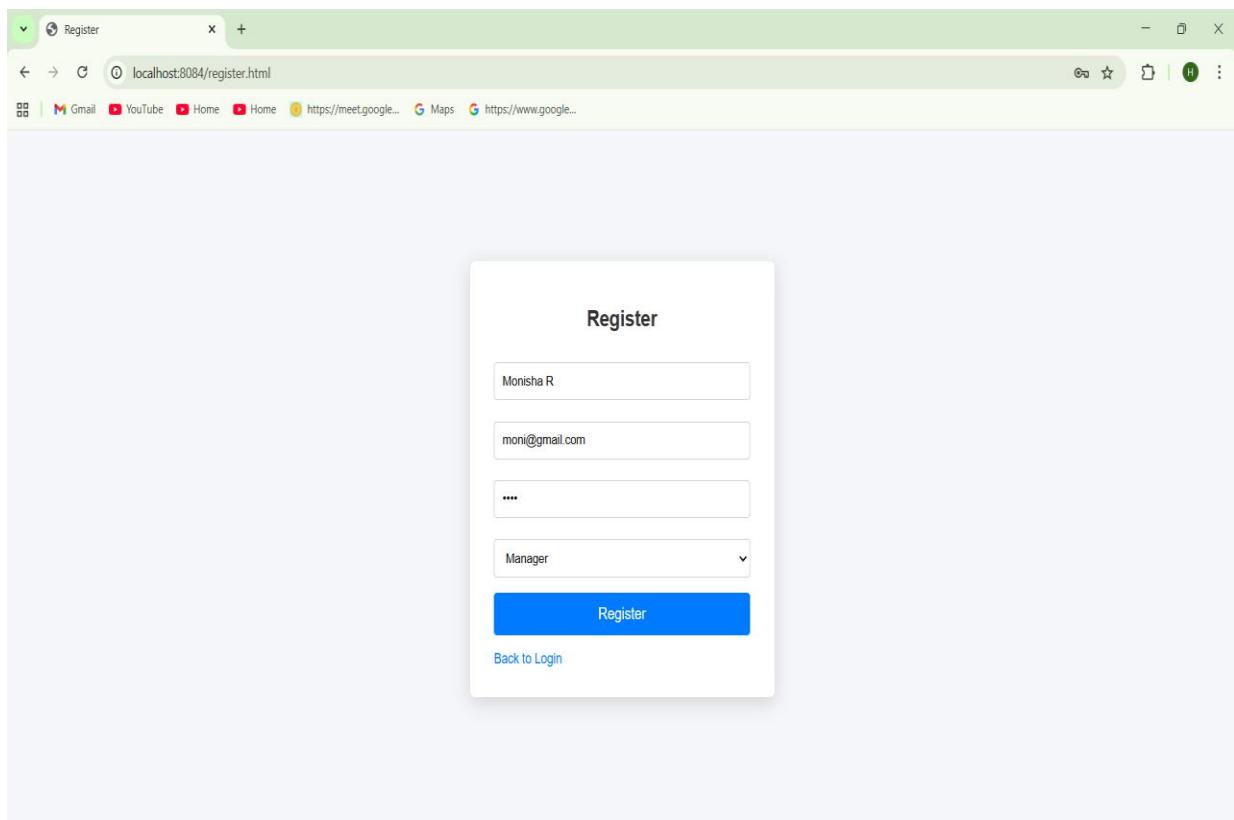
Before Logout

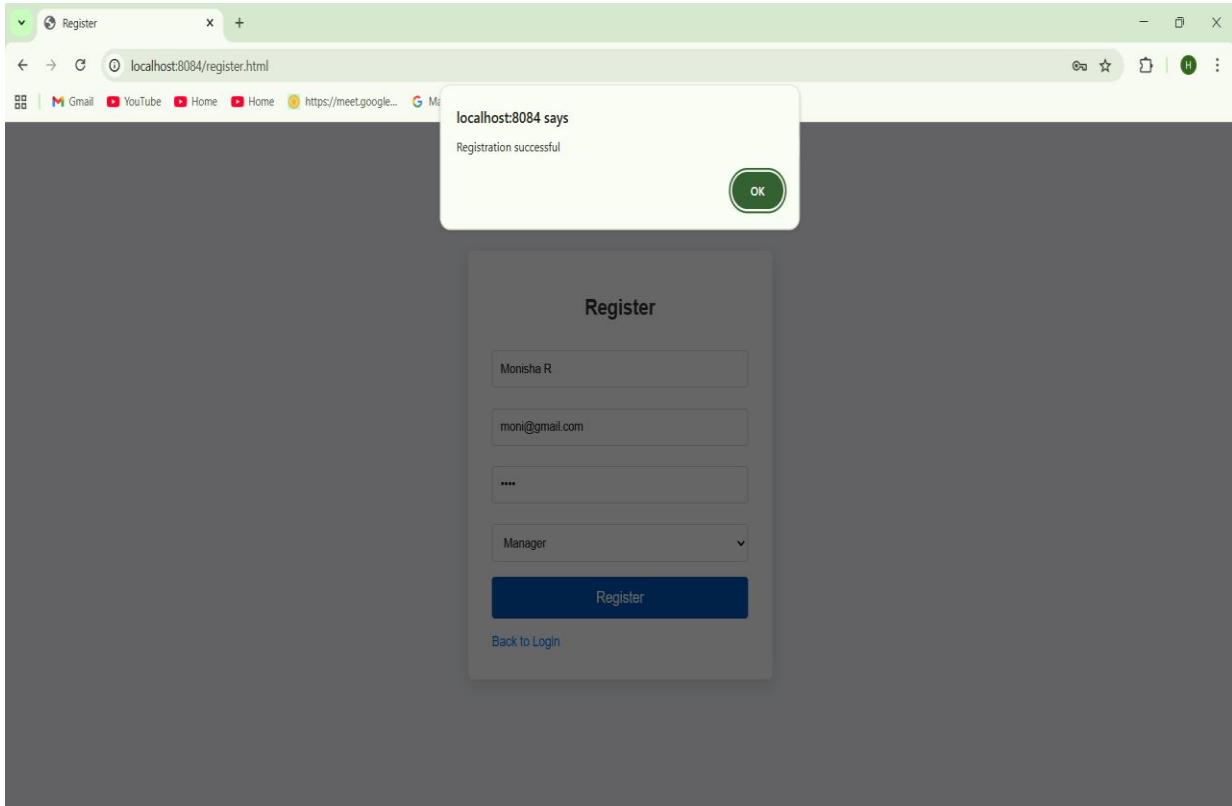


Logged Out

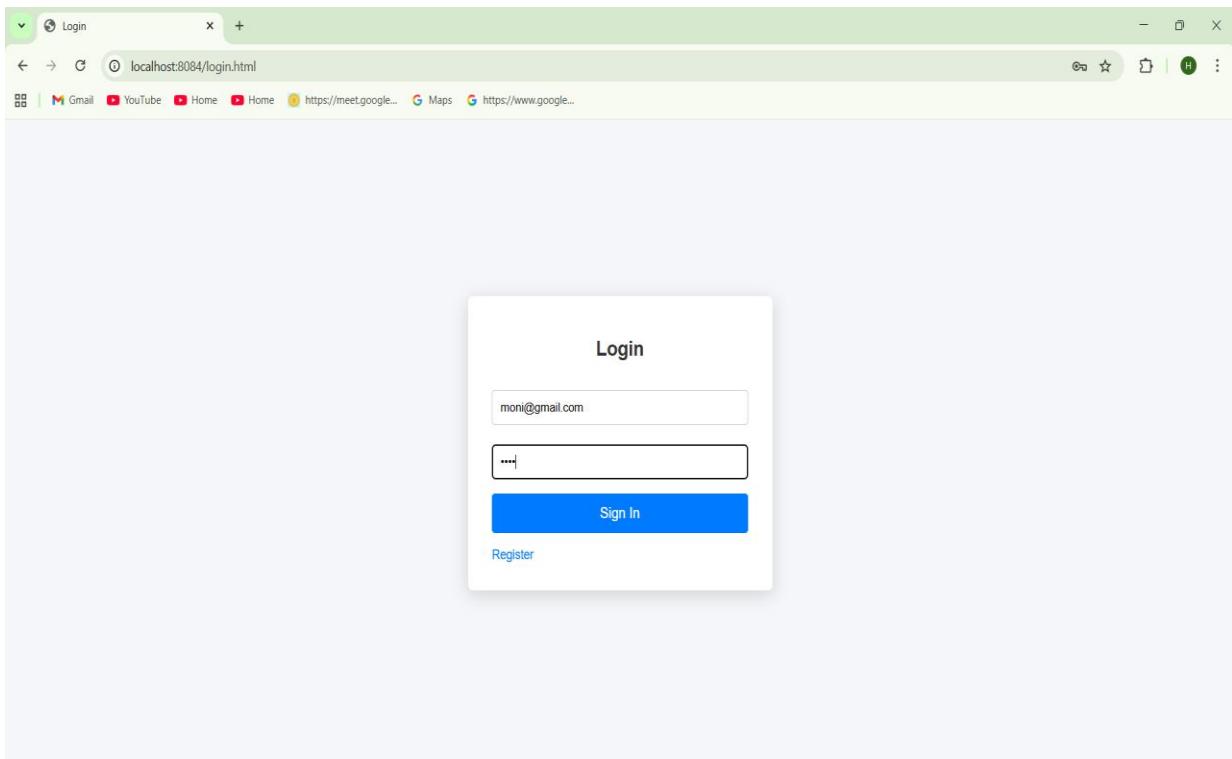


Manager's Register Page





Manager's Login Page



Manager's Home Page

The screenshot shows a web browser window titled "Manager Dashboard" at the URL "localhost:8084/manager/dashboard.html". The page is titled "Leave Management System" and displays a calendar for December 2025. The calendar shows days from 30 to 6. The days from 14 to 18 are highlighted in yellow, indicating a selected period. On the left, a sidebar menu for "Leave System" includes links for Home, Manage Users, Manage Leaves, Apply Leave, My Leaves, Change Password, and Logout. The top right corner shows the email "manager@email.com [MANAGER]".

Manage Users Page

The screenshot shows a web browser window titled "Manage Users" at the URL "localhost:8084/manager/manage-users.html". The page is titled "Leave Management System" and displays a list of users under the heading "Manage Users". The table has columns for NAME, E-MAIL, ROLE, and ACTION. The users listed are Project Manager (manager@email.com, MANAGER), harini s (harini@gmail.com, EMPLOYEE), and Harshita Sri R (harshi@gmail.com, EMPLOYEE). The top right corner shows the email "manager@email.com [MANAGER]".

NAME	E-MAIL	ROLE	ACTION
Project Manager	manager@email.com	MANAGER	
harini s	harini@gmail.com	EMPLOYEE	
Harshita Sri R	harshi@gmail.com	EMPLOYEE	

Manage Leaves Page

Employee	From	To	Type	Reason	Status	Action
aishu1@gmail.com	2025-12-18	2025-12-18	CASUAL	Home tour	REJECTED	<button>Approve</button> <button>Reject</button>
ranju1@gmail.com	2025-12-19	2025-12-21	SICK	Dengue Fever	ACCEPTED	<button>Approve</button> <button>Reject</button>
aditi1@gmail.com	2026-01-05	2026-01-10	CASUAL	Family Occasion	ACCEPTED	<button>Approve</button> <button>Reject</button>
harshita@gmail.com	2025-12-23	2025-12-23	CASUAL	Casual Leave	REJECTED	<button>Approve</button> <button>Reject</button>
swarna1@gmail.com	2025-12-18	2025-12-18	CASUAL	Casual Leave	ACCEPTED	<button>Approve</button> <button>Reject</button>
ravi@gmail.com	2026-01-02	2026-01-04	CASUAL	Home Visit	ACCEPTED	<button>Approve</button> <button>Reject</button>
aditi@gmail.com	2025-12-17	2025-12-26	CASUAL	Home Outing	PENDING	<button>Approve</button> <button>Reject</button>
ranju@gmail.com	2025-12-19	2025-12-19	CASUAL	Personal Outing	ACCEPTED	<button>Approve</button> <button>Reject</button>
lux@gmail.com	2025-12-18	2025-12-21	CASUAL	Home Outing	PENDING	<button>Approve</button> <button>Reject</button>

Manager – Approve/Reject Leaves

Employee	From	To	Type	Reason	Status	Action
aishu1@gmail.com	2025-12-18	2025-12-18	CASUAL	Home tour	REJECTED	<button>Approve</button> <button>Reject</button>
ranju1@gmail.com	2025-12-19	2025-12-21	SICK	Dengue Fever	ACCEPTED	<button>Approve</button> <button>Reject</button>
aditi1@gmail.com	2026-01-05	2026-01-10	CASUAL	Family Occasion	ACCEPTED	<button>Approve</button> <button>Reject</button>
harshita@gmail.com	2025-12-23	2025-12-23	CASUAL	Casual Leave	REJECTED	<button>Approve</button> <button>Reject</button>
swarna1@gmail.com	2025-12-18	2025-12-18	CASUAL	Casual Leave	ACCEPTED	<button>Approve</button> <button>Reject</button>
ravi@gmail.com	2026-01-02	2026-01-04	CASUAL	Home Visit	ACCEPTED	<button>Approve</button> <button>Reject</button>
aditi@gmail.com	2025-12-17	2025-12-26	CASUAL	Home Outing	PENDING	<button>Approve</button> <button>Reject</button>
ranju@gmail.com	2025-12-19	2025-12-19	CASUAL	Personal Outing	ACCEPTED	<button>Approve</button> <button>Reject</button>
lux@gmail.com	2025-12-18	2025-12-21	CASUAL	Home Outing	ACCEPTED	<button>Approve</button> <button>Reject</button>

Leave Pending Status in Calender

The screenshot shows a web-based leave management system. On the left is a dark sidebar with a navigation menu:

- Leave System
- Home
- Manage Users
- Manage Leaves
- Apply Leave
- My Leaves
- Change Password
- Logout

The main content area is titled "Leave Management System" and shows the date "December 2025". At the top of the calendar, there are filter buttons: "Accepted", "Rejected", "Pending" (which is checked), and "Accepted". The calendar grid displays dates from 30 November to 6 December. A blue horizontal bar spans from 17 to 20 December, with the email address "adhi@gmail.com" written on it. The days 17, 18, 19, and 20 December are highlighted in yellow.

Leave Approved Status in Calender

This screenshot shows the same leave management system interface, but with different filter settings. The "Accepted" button is checked, while "Pending" is unchecked. The calendar for December 2025 is displayed. A green horizontal bar spans from 17 to 20 December, containing the email addresses "lux@gmail.com", "swarna1@gmail.com", and "ranju1@gmail.com". The days 17, 18, 19, and 20 December are highlighted in green. The days 30 November, 1, 2, 3, 4, 5, and 6 December are shown in their original light gray color.

Leave Rejected Status in Calender

Manager Dashboard

localhost:8084/manager/dashboard.html

Leave Management System

manager@email.com [MANAGER]

Home

Pending Accepted Rejected

Pending Accepted Rejected

December 2025

today < >

Sun	Mon	Tue	Wed	Thu	Fri	Sat
30	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18 aishu1@gmail.com	19	20

Pending/Approved/Rejected Status in Calender

Manager Dashboard

localhost:8084/manager/dashboard.html

Leave Management System

manager@email.com [MANAGER]

Home

Pending Accepted Rejected

Pending Accepted Rejected

December 2025

today < >

Sun	Mon	Tue	Wed	Thu	Fri	Sat
30	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18 adit@gmail.com lux@gmail.com aishu1@gmail.com ranju@gmail.com swarna1@gmail.com ranju@gmail.com	19	20

Manager Dashboard

localhost:8084/manager/dashboard.html

Apply Leave

My Leaves

Change Password

Logout

December 2025

Sun	Mon	Tue	Wed	Thu	Fri	Sat
30	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27

aditi@gmail.com

lux@gmail.com

aishu1@gmail.com ranju1@gmail.com

swarna1@gmail.com ranju@gmail.com

aditi@gmail.com

harshita@gmail.com

today < >

This screenshot shows the Manager Dashboard for December 2025. The sidebar on the left includes links for Apply Leave, My Leaves, Change Password, and Logout. The main area displays a monthly calendar from December 30 to January 6. Several leave requests are marked with colored bars: a blue bar for December 17 (aditi@gmail.com), a green bar for December 18 (lux@gmail.com), a red bar for December 19 (aishu1@gmail.com, ranju1@gmail.com), a green bar for December 20 (swarna1@gmail.com, ranju@gmail.com), a long blue bar spanning December 21 to 25 (aditi@gmail.com), and a red bar for December 22 (harshita@gmail.com). Navigation buttons for today, previous month, and next month are at the top right.

Manager Dashboard

localhost:8084/manager/dashboard.html

Leave System

Home

Manage Users

Manage Leaves

Apply Leave

My Leaves

Change Password

Logout

Leave Management System

manager@email.com [MANAGER]

Leave Management System

Home

Pending Accepted Rejected

Pending Accepted Rejected

January 2026

Sun	Mon	Tue	Wed	Thu	Fri	Sat
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17

ravi@gmail.com

aditi1@gmail.com

today < >

This screenshot shows the Leave Management System for January 2026. The sidebar on the left includes links for Home, Manage Users, Manage Leaves, Apply Leave, My Leaves, Change Password, and Logout. The main area displays a monthly calendar from January 1 to 17. A single leave request is marked with a green bar for January 1 (ravi@gmail.com). Navigation buttons for today, previous month, and next month are at the top right.