

**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)

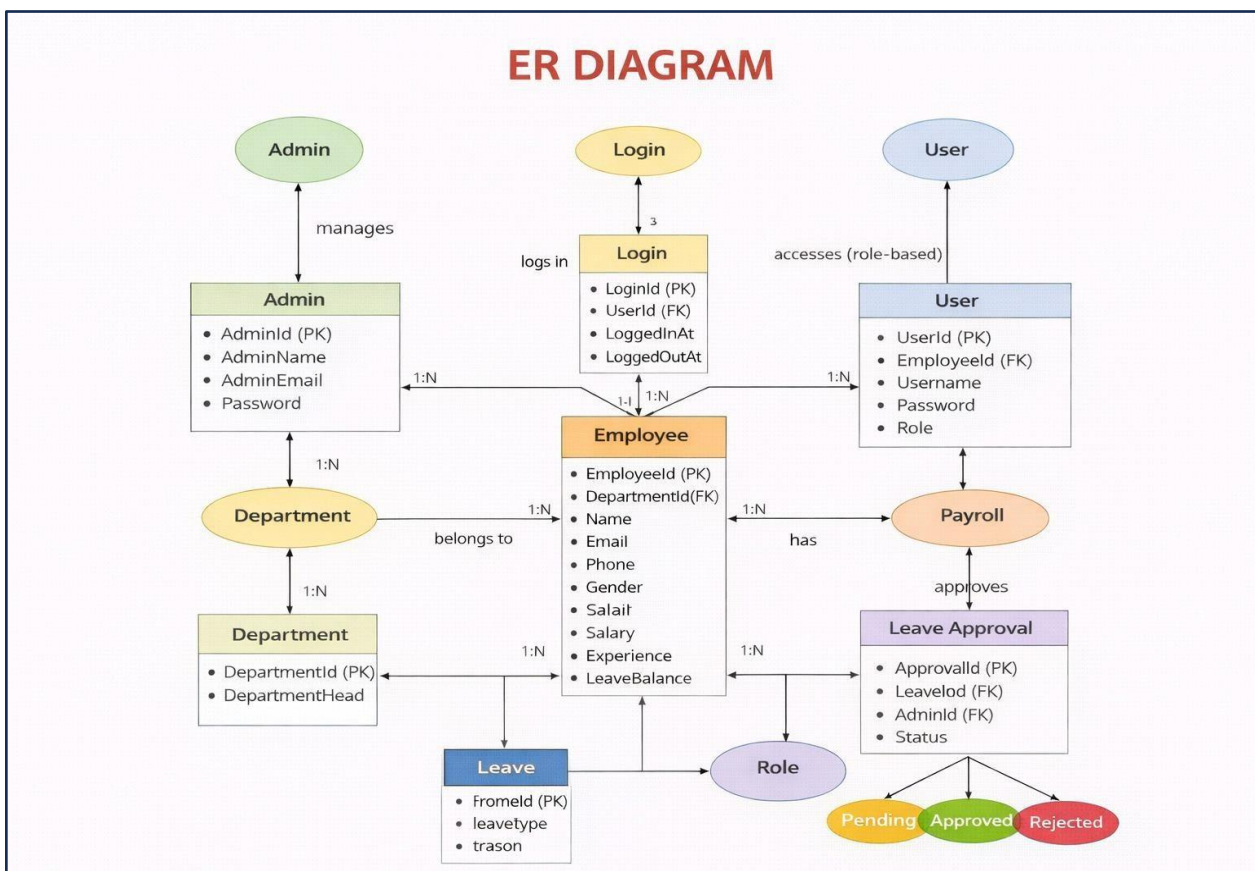
- **Purpose:** To allow administrators to oversee and manage all user accounts.
- **Key Functions:**
  - View a complete list of all system users.
  - Update user profiles (e.g., name, email, role).
  - Delete user accounts from the system.

### Module 3: Employee Leave Module

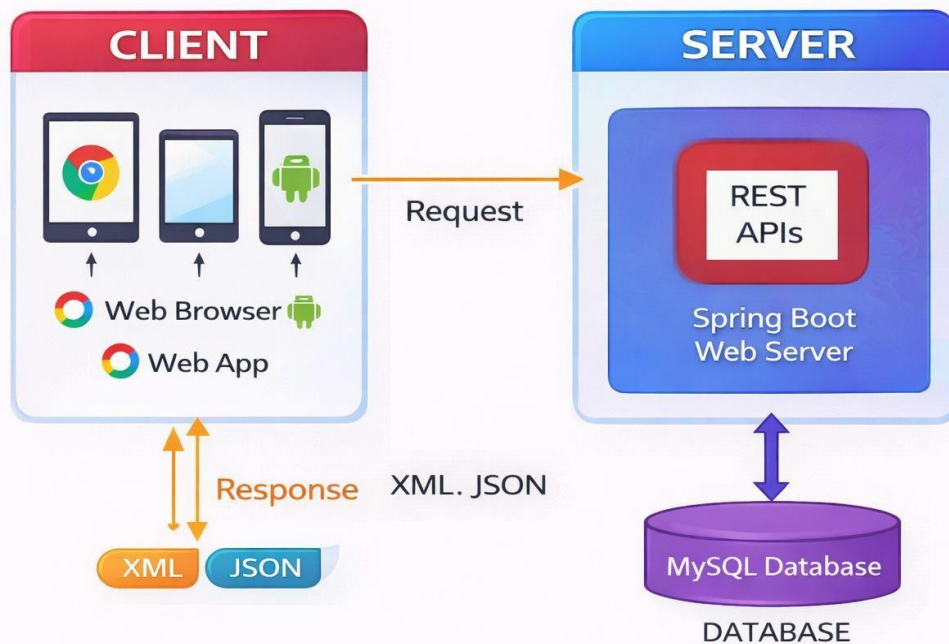
- **Purpose:** To provide employees with self-service leave management.
- **Key Functions:**
  - Submit a new leave application form.
  - View a personal dashboard listing all their own leave requests (history and current status).

### 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**.

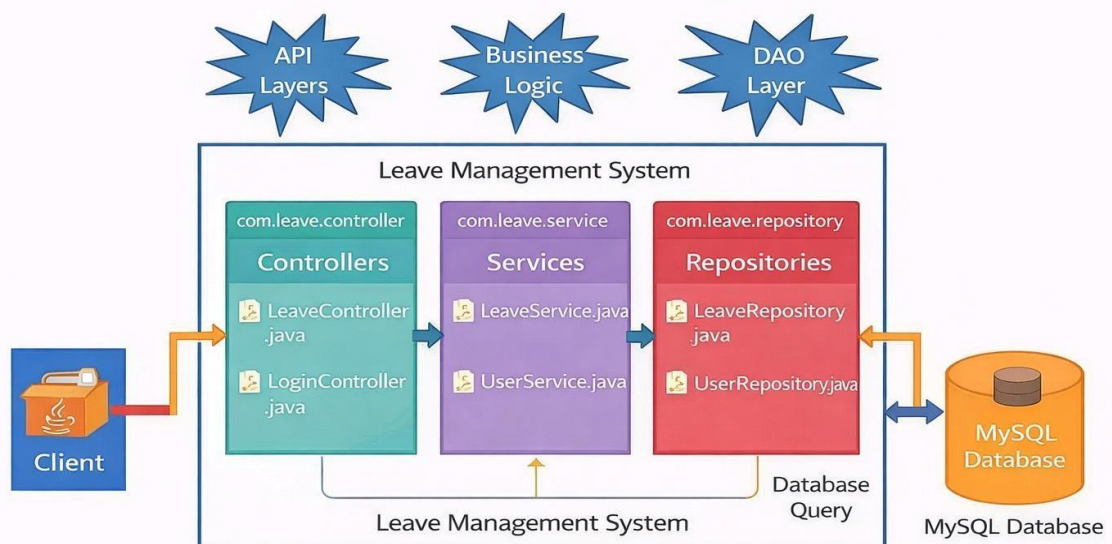


## CLIENT – SERVER ARCHITECTURE



## 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	Retrives all registered users
POST	http://localhost:8084/users/update	Updates user details
POST	http://localhost:8084/users/chang e-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	Retrives logged in employees leave

## Leave Management Module - Manager

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

# DATA DICTIONARY

## DATABASE

## MANAGER / EMPLOYEE DATABASE

The screenshot displays the MySQL Workbench interface for a local instance of MySQL 8.0. The left sidebar shows the 'MANAGEMENT' and 'INSTANCE' sections. The main window shows a SQL editor with the following queries:

```

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

```

The 'Result Grid' shows the output of the query 'SELECT \* FROM USERS'. It contains 9 rows of data:

	id	name	email	password	role
1					EMPLOYEE
2					EMPLOYEE
4					MANAGER
6					EMPLOYEE
7					EMPLOYEE
8					MANAGER
9					EMPLOYEE
9					EMPLOYEE
9					EMPLOYEE

The 'Output' section shows the execution log of the queries:

#	Time	Action	Message	Duration / Fetch
5	20:05:38	CREATE TABLE leaves ( id INT AUTO_INCREMENT PRIMARY KEY, user_email VARCHAR(100) NOT ...	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. The SQL editor contains the following query:

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

The Results Grid displays the following data:

#	id	User_email	from_date	to_date	leave_type	reason	status
1	1	ranju@gmail.com	2026-01-01	2025-12-02	CASUAL	Home Visit	PENDING
2	2	harsh@gmail.com	2026-01-17	2025-12-19	SICK	Dengue Fever	PENDING

The Output tab shows the following log:

#	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 request is a GET method to the URL `http://localhost:8084/manager/manage-leaves.html`. The response is a 200 OK status with a time of 19 ms and a size of 894 B. The response body is shown in the Pretty tab, displaying the following HTML structure:

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

# POST METHOD ON APPLYING LEAVES

Home Workspaces Explore Search Postman Sign In Create Account

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

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

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

Method: GET

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

Query Params: Key Value Bulk Edit

Body: Cookies Headers (10) Test Results Status: 200 OK Time: 19 ms Size: 2.56 KB Save Response

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>
```

# GET METHOD ON MY LEAVES

Home Workspaces Explore Search Postman Sign In Create Account

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

Today: GET http://localhost:8084/employee/my-leaves.html, GET http://localhost:8084/employee/apply-leave.html, GET http://localhost:8084/manager/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

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

Method: GET

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

Query Params: Key Value Bulk Edit

Body: Cookies Headers (10) Test Results Status: 200 OK Time: 24 ms Size: 1.16 KB Save Response

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
```

## GET METHOD ON CHANGE PASSWORD

The screenshot shows the Postman interface with a GET request to `http://localhost:8084/employee/change-password.html`. The response is an HTML document with the following structure:

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

The status bar indicates a 200 OK response with a time of 13 ms and a size of 768 B.

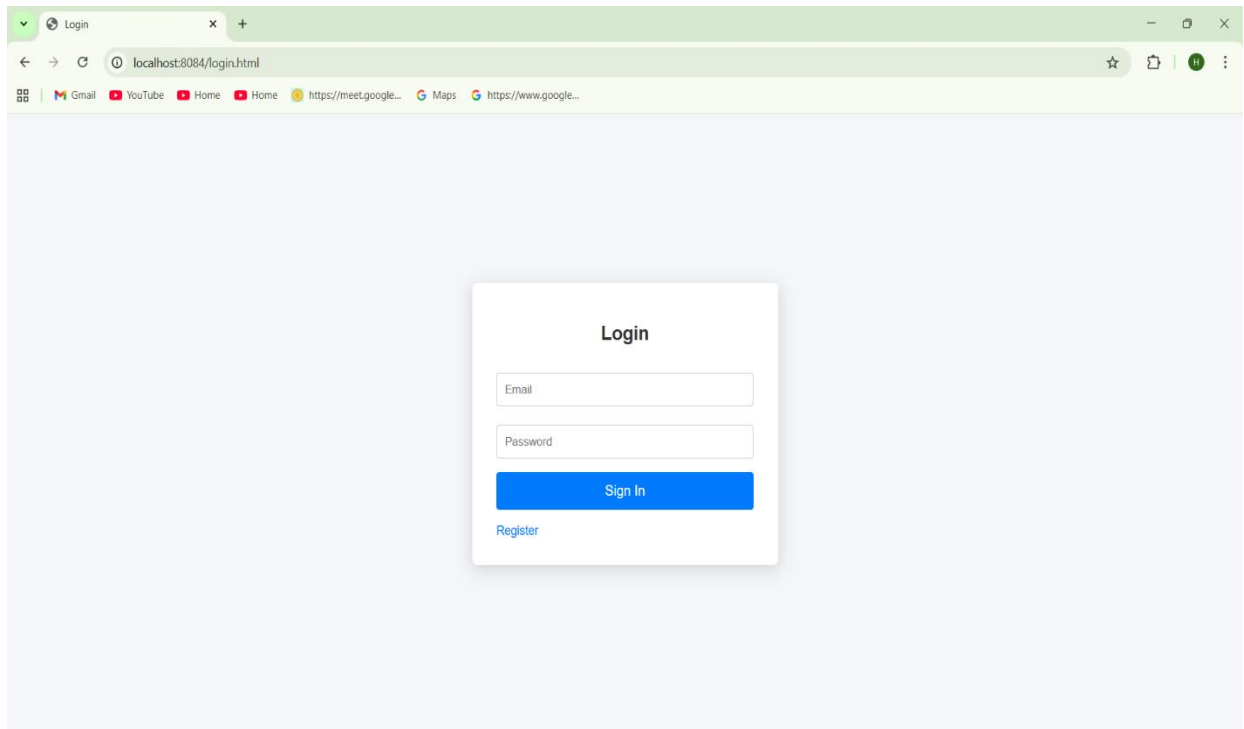
## JUNITTEST CASES FOR MANAGER MODULE

The screenshot shows the Eclipse IDE with a JUnit test case for the `LeaveDAO` interface. The test class is `LeaveDAOTest` and the test method is `testApplyLeave()`. The test method creates a `Leave` object, sets its properties, and calls the `applyLeave()` method of the `LeaveDAO` interface. The test is run successfully, as indicated by the green bar in the console.

```
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@gmail.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
28
29
30
```

## PROJECT OUTPUTS

### Login Page



Login

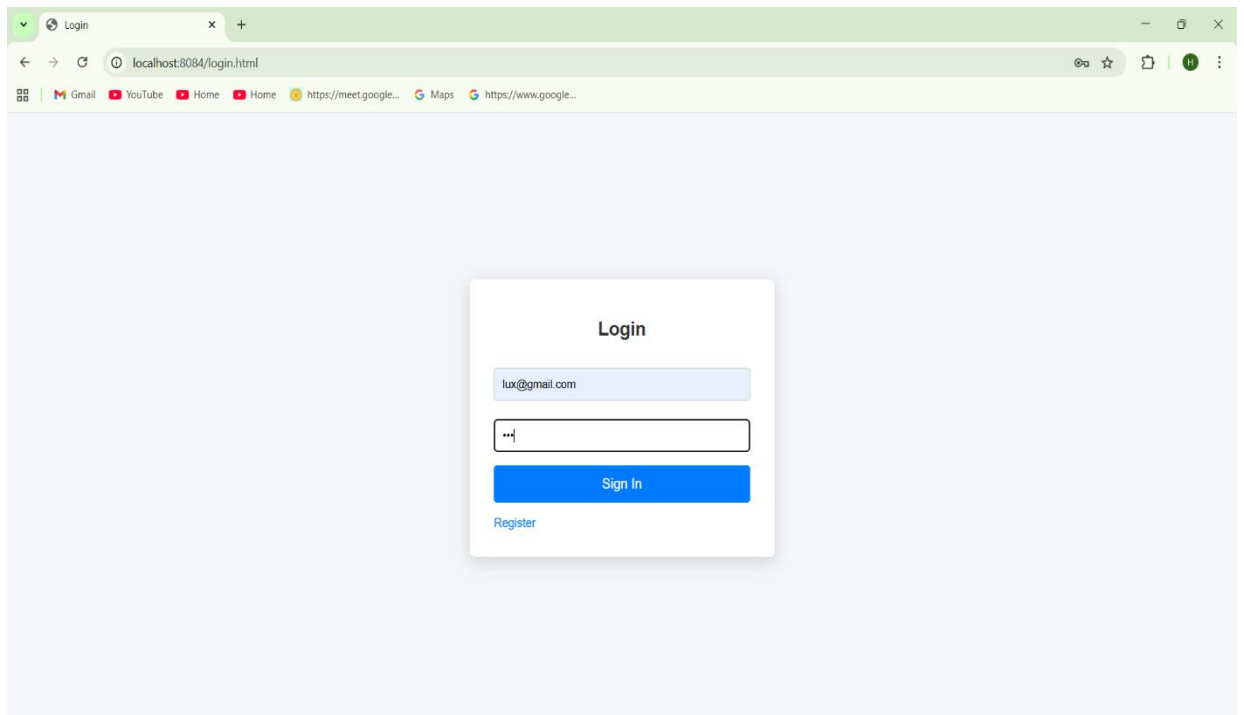
Email

Password

Sign In

[Register](#)

### Employee's Login Page



Login

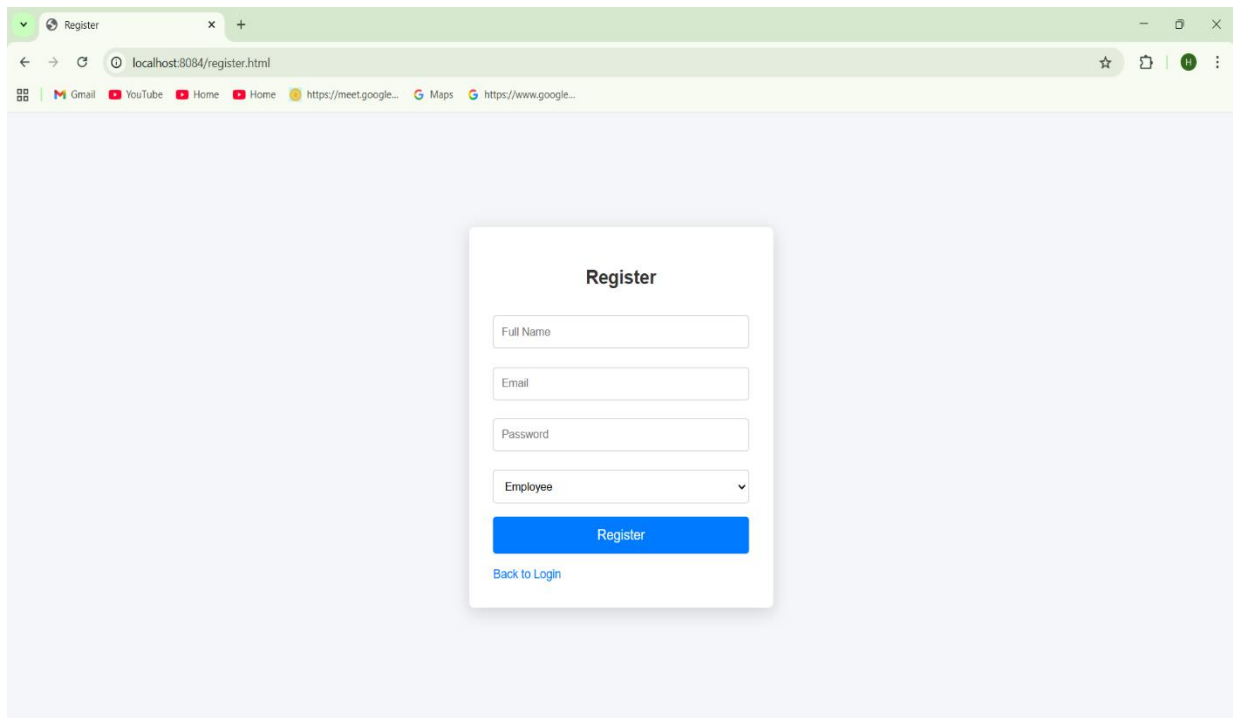
lux@gmail.com

\*\*\*

Sign In

[Register](#)

## Registration Page



A screenshot of a web browser window displaying a registration form. The browser's address bar shows 'localhost:8084/register.html'. The form is titled 'Register' and contains the following fields: 'Full Name', 'Email', 'Password', and a dropdown menu for 'Employee'. A blue 'Register' button is positioned below the dropdown, and a blue link 'Back to Login' is located at the bottom left of the form.

Register

Full Name

Email

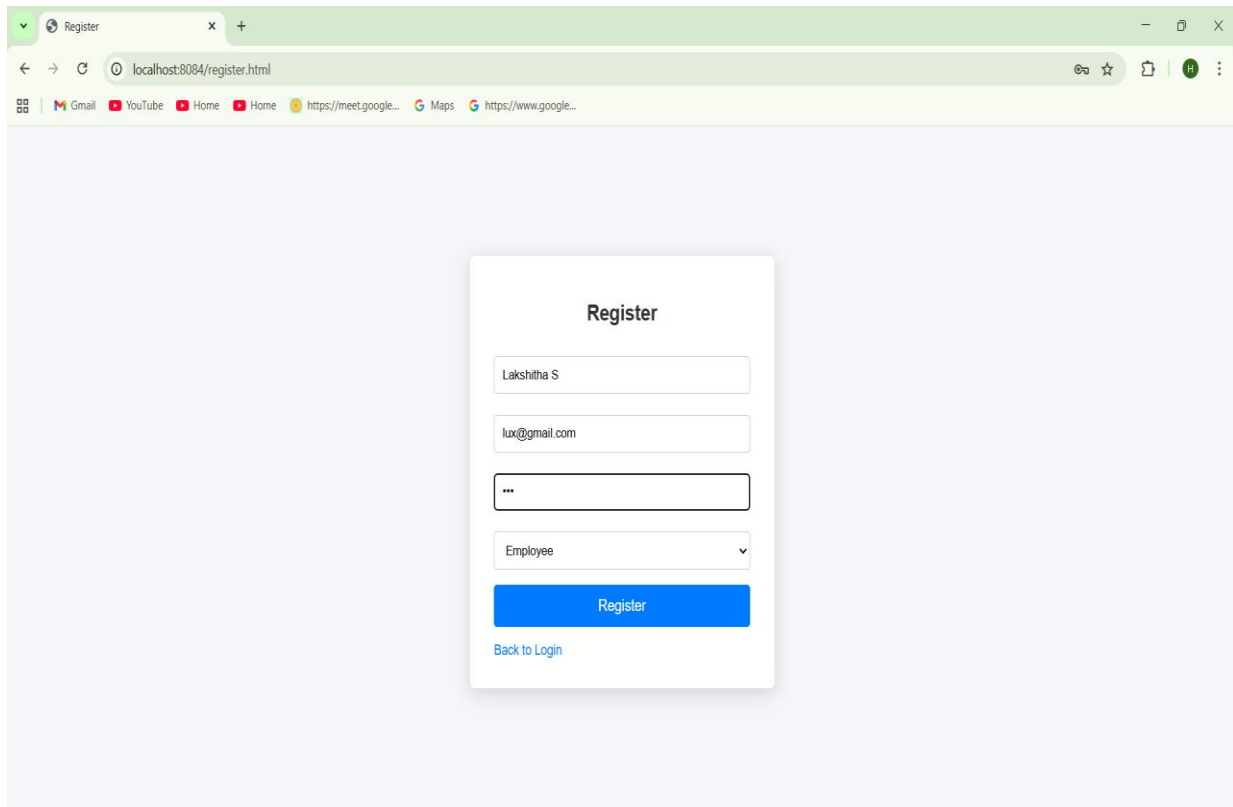
Password

Employee

Register

[Back to Login](#)

## Employee's Register Page



A screenshot of a web browser window displaying the same registration form as above, but with pre-filled data. The 'Full Name' field contains 'Lakshitha S', the 'Email' field contains 'lux@gmail.com', and the 'Password' field contains three asterisks. The 'Employee' dropdown menu is also present. The blue 'Register' button and the 'Back to Login' link remain at the bottom.

Register

Lakshitha S

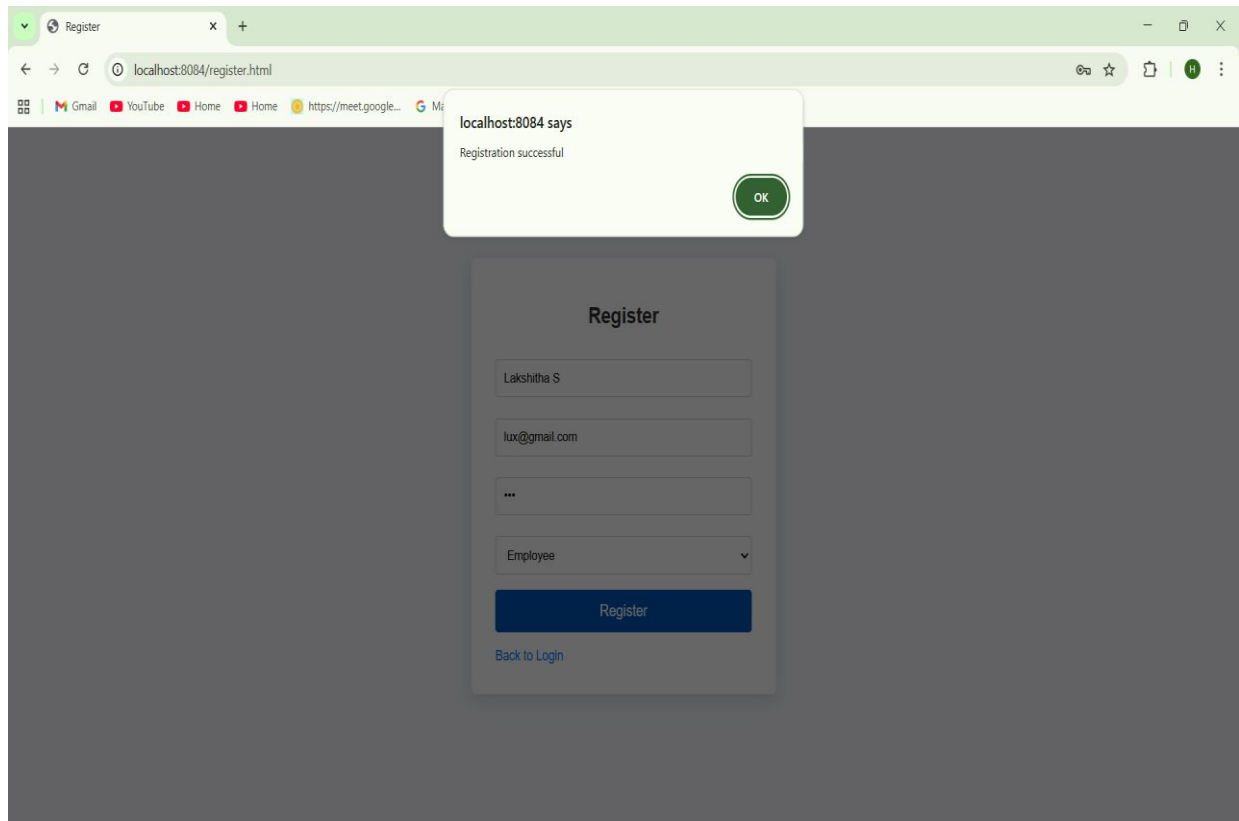
lux@gmail.com

\*\*\*

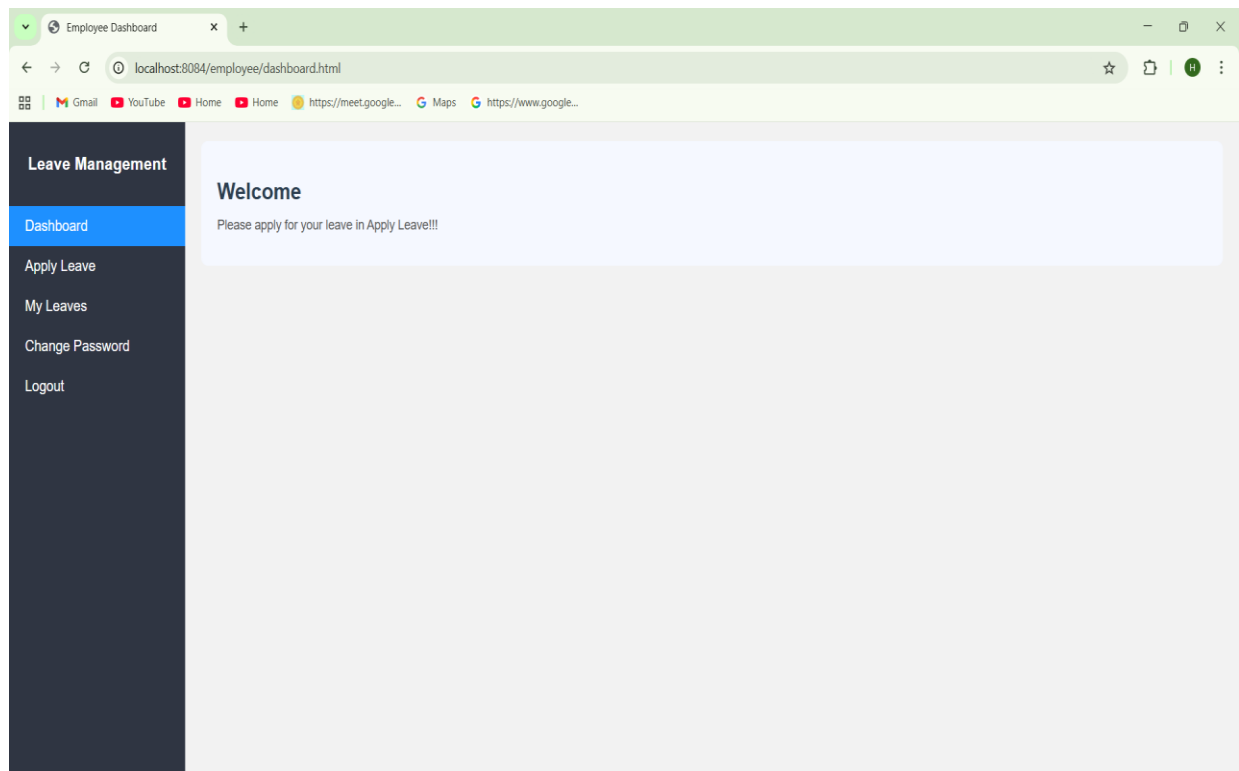
Employee

Register

[Back to Login](#)



## Employee's Dashboard Page



# Request For Leave

Apply Leave

localhost:8084/employee/apply-leave.html

manager@email.com [MANAGER]

Home

Apply Leave

My Leaves

Change Password

Home > Apply Leave

### Request For Leave

From Date :

To Date :

Leave Type :

Leave Reason :

APPLY LEAVE CANCEL

# Employee's Leave Request

Apply Leave

localhost:8084/employee/apply-leave.html

manager@email.com [MANAGER]

Home

Apply Leave

My Leaves

Change Password

Home > Apply Leave

### Request For Leave

From Date :

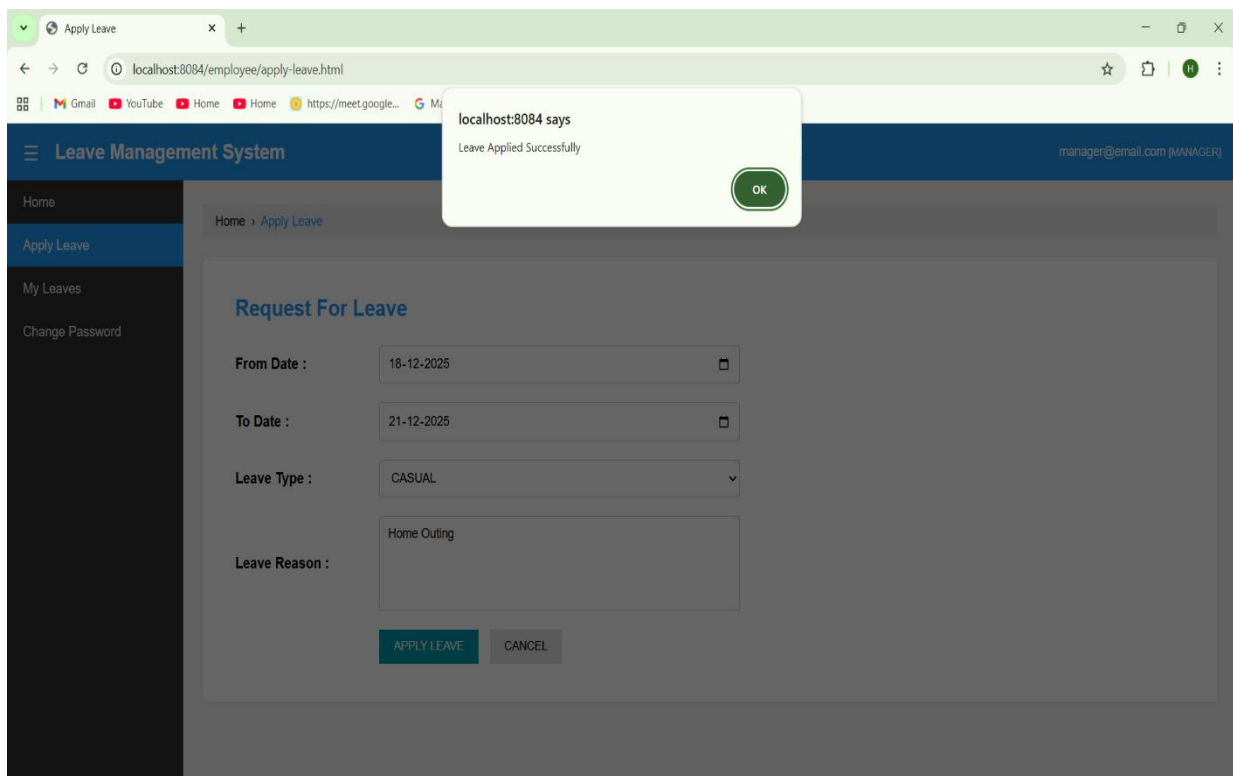
To Date :

Leave Type :

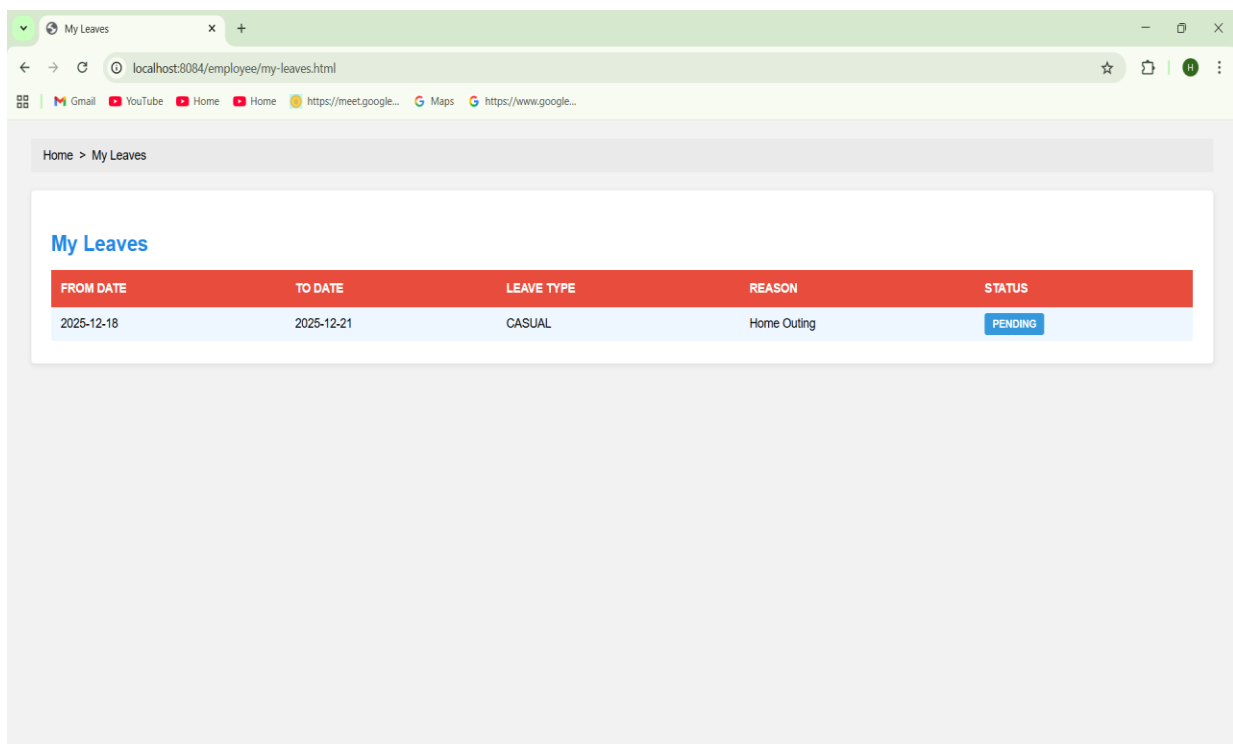
Leave Reason :

APPLY LEAVE CANCEL





## Employee's - My Leaves Page



# Change Password Page



Home » Change Password

## Change Password

Current Password :

New Password :

Re-Enter New Password :

# Changing New Password



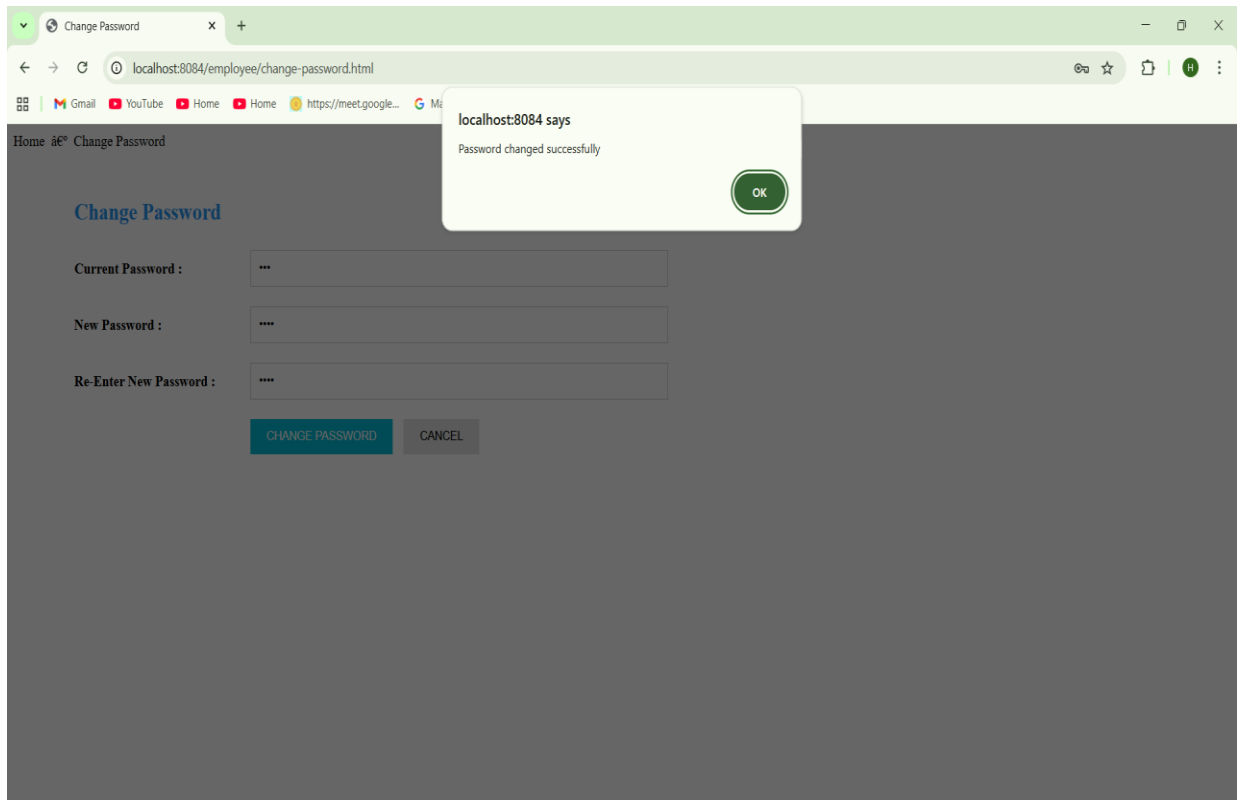
Home » Change Password

## Change Password

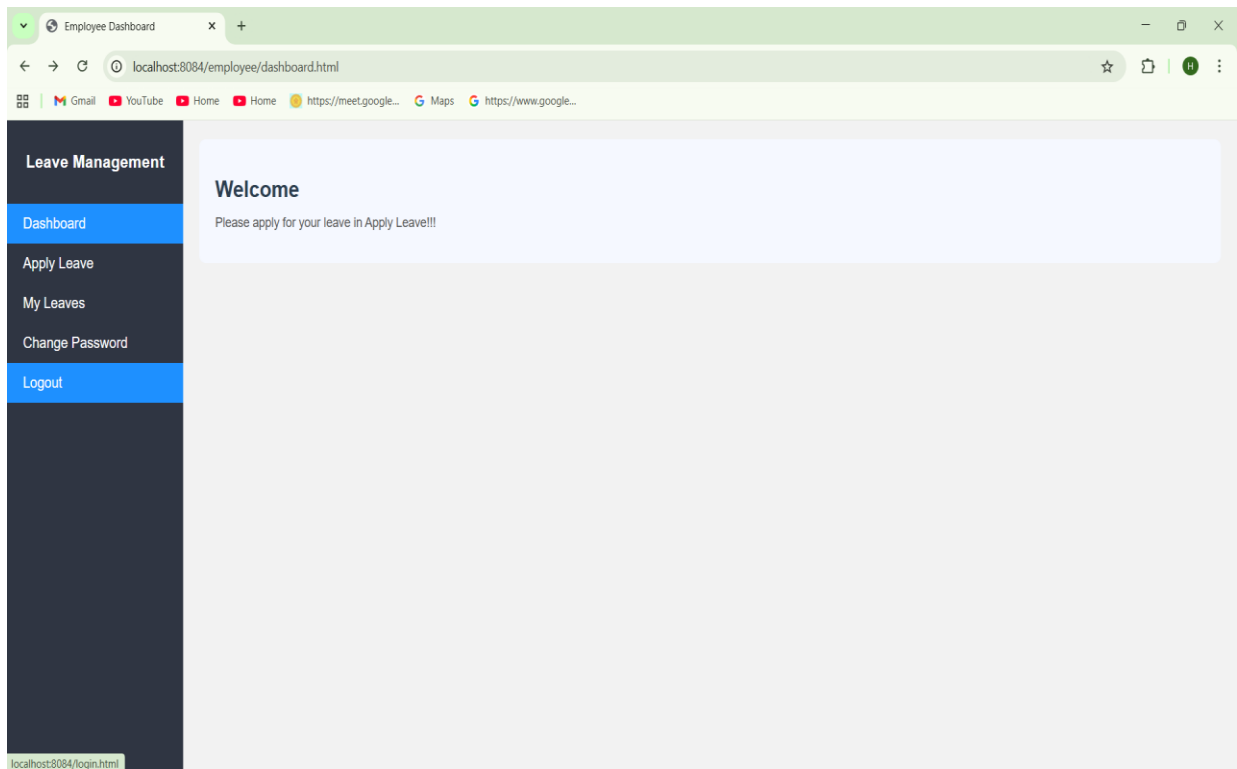
Current Password :

New Password :

Re-Enter New Password :



## Before Logout



## Logged Out

Login

Email

Password

Sign In

[Register](#)

## Manager's Register Page

Register

Monisha R

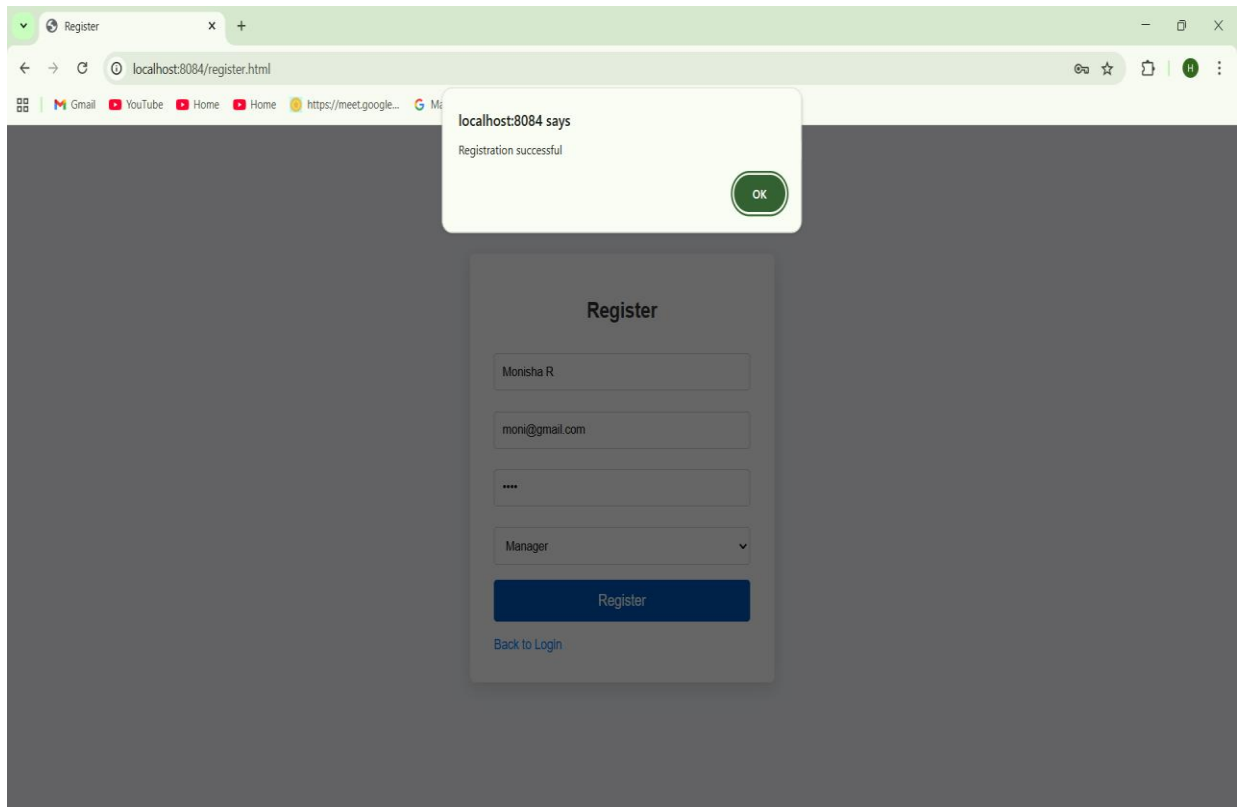
moni@gmail.com

\*\*\*\*

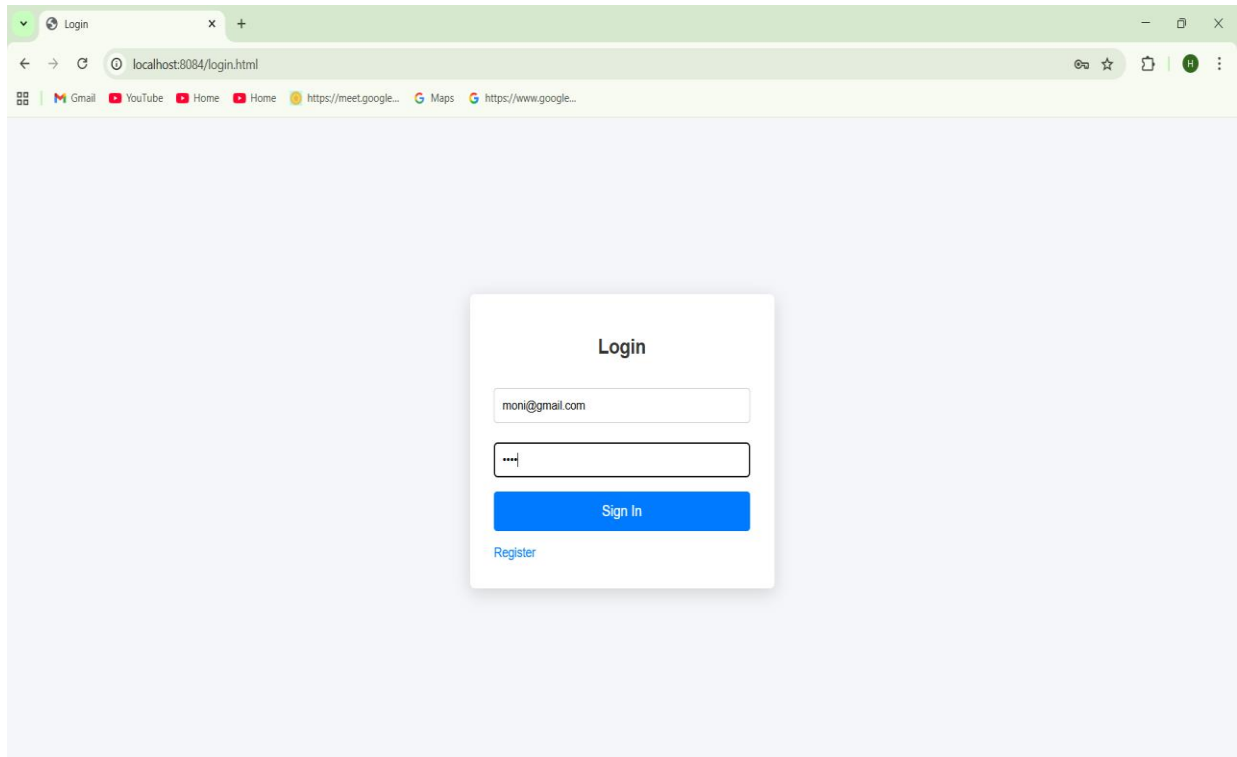
Manager

Register

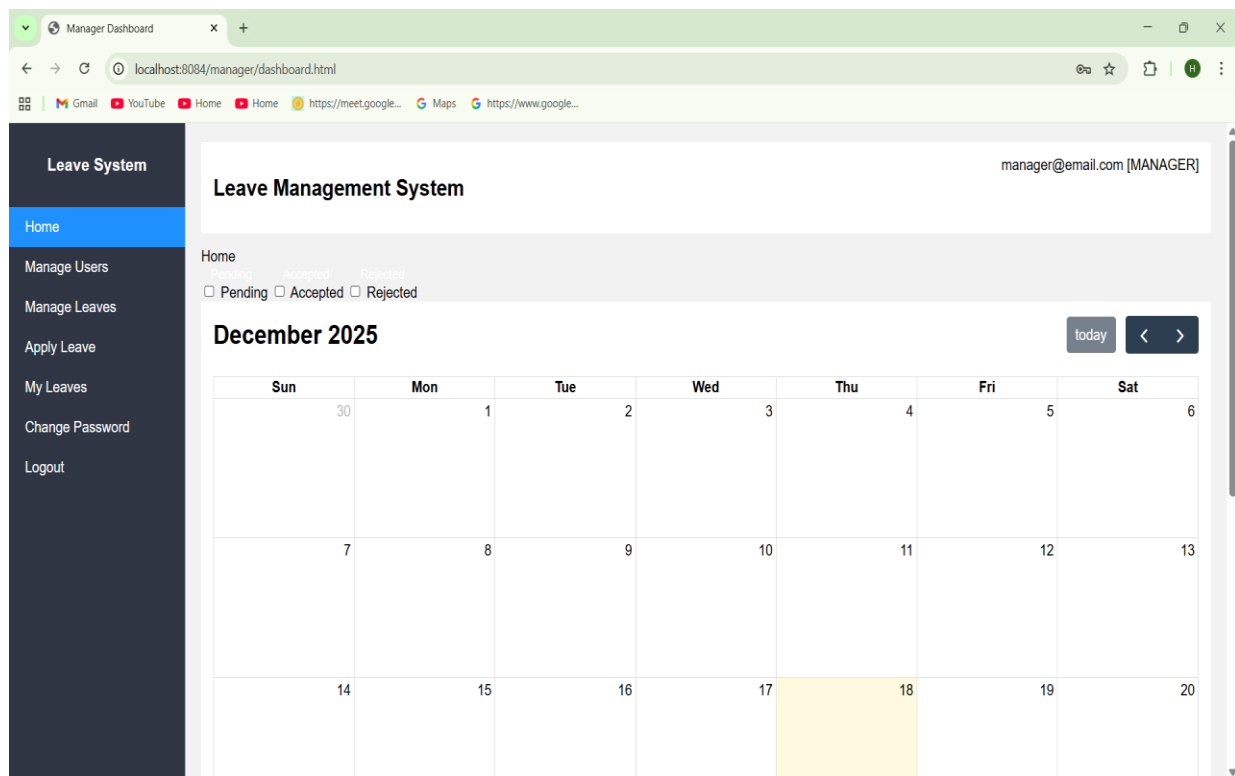
[Back to Login](#)



## Manager's Login Page



# Manager's Home Page



The screenshot shows a web browser window with the address bar displaying 'localhost:8084/manager/dashboard.html'. The page title is 'Manager Dashboard'. The main content area is titled 'Leave Management System' and shows the user 'manager@email.com [MANAGER]'. The left sidebar contains a menu with 'Leave System' and 'Home' (selected). The main content area has a 'Home' section with filters for 'Pending', 'Accepted', and 'Rejected'. Below this is a calendar for 'December 2025' with a 'today' button and navigation arrows. The calendar grid shows dates from 30 to 20, with the 18th highlighted in yellow.

Manager Dashboard

localhost:8084/manager/dashboard.html

manager@email.com [MANAGER]

## Leave Management System

Home

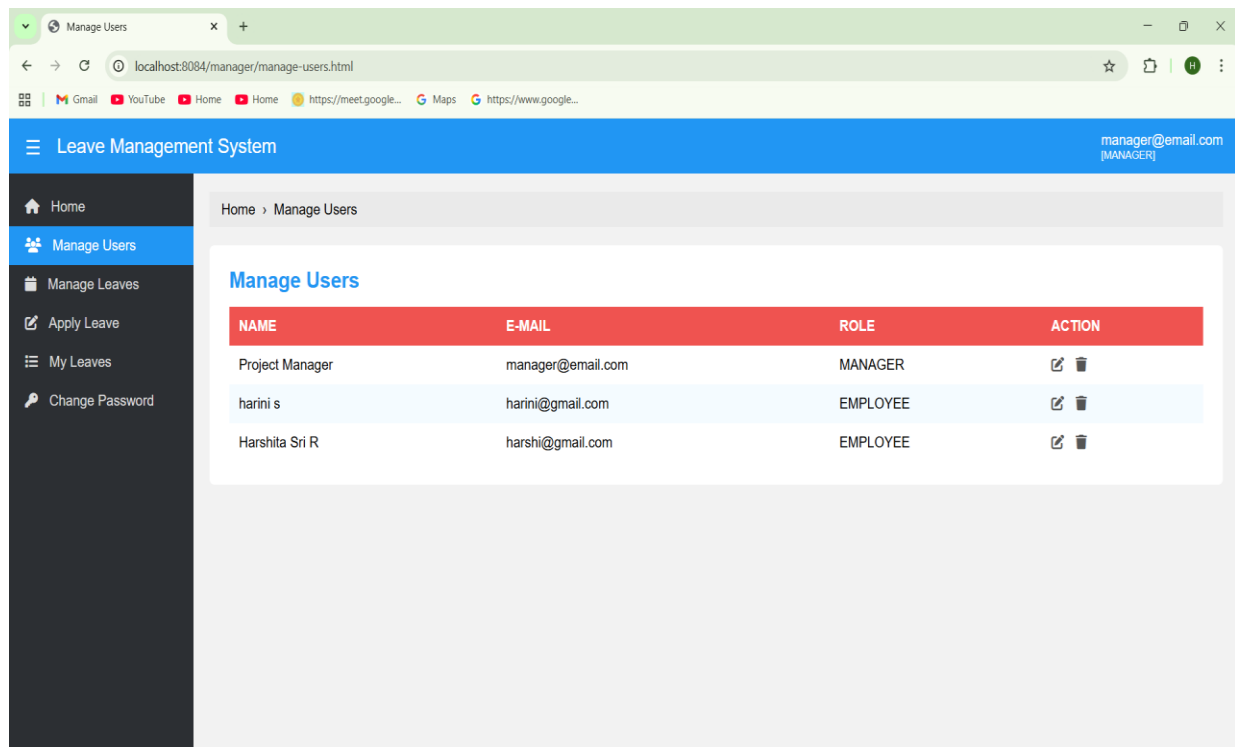
☐ Pending ☐ Accepted ☐ Rejected

### 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	19	20

# Manage Users Page



The screenshot shows a web browser window with the address bar displaying 'localhost:8084/manager/manage-users.html'. The page title is 'Manage Users'. The left sidebar contains a menu with 'Home', 'Manage Users' (selected), 'Manage Leaves', 'Apply Leave', 'My Leaves', and 'Change Password'. The main content area has a 'Home > Manage Users' breadcrumb. Below this is a 'Manage Users' section with a table listing users: Project Manager (manager@email.com, MANAGER), harini s (harini@gmail.com, EMPLOYEE), and Harshita Sri R (harshi@gmail.com, EMPLOYEE). Each user has an edit icon and a delete icon.

Manage Users

localhost:8084/manager/manage-users.html

manager@email.com [MANAGER]

## Manage Users

Home > Manage Users

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

Manage Leaves

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

## Manager – Approve/Reject Leaves

Manage Leaves

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

## Leave Pending Status in Calender

The screenshot shows the 'Leave Management System' interface. The left sidebar contains a 'Leave System' menu with options: Home, Manage Users, Manage Leaves, Apply Leave, My Leaves, Change Password, and Logout. The main content area is titled 'Leave Management System' and shows the user 'manager@email.com [MANAGER]'. Below the title, there are tabs for 'Pending', 'Accepted', and 'Rejected', with 'Pending' selected. A calendar for 'December 2025' is displayed, showing dates from 30 to 6. A blue bar highlights the dates 17 and 18, with the email 'aditi@gmail.com' written on it, indicating a pending leave request.

## Leave Approved Status in Calender

The screenshot shows the 'Leave Management System' interface with the 'Accepted' tab selected. The calendar for 'December 2025' displays approved leave requests for the dates 17 and 18. The requests are listed as follows:

Date	Requester
17	lux@gmail.com
17	swarna1@gmail.com
18	ranju1@gmail.com
18	ranju@gmail.com



## Leave Rejected Status in Calender

The screenshot shows a web browser window with the URL `localhost:8084/manager/dashboard.html`. The page title is "Leave Management System" and the user is logged in as "manager@email.com [MANAGER]". The left sidebar contains a "Leave System" menu with options: Home, Manage Users, Manage Leaves, Apply Leave, My Leaves, Change Password, and Logout. The main content area shows a calendar for December 2025. The calendar has a header with "Home", "Pending", "Accepted", and "Rejected" filters, with "Rejected" selected. The calendar grid shows dates from 30 to 20. A red box with the text "aishu1@gmail.com" is visible on the 18th, indicating a rejected leave request.

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

The screenshot shows the same web browser window as the previous one, but with different filter settings. The "Pending", "Accepted", and "Rejected" filters are all selected. The calendar grid shows dates from 30 to 20. On the 18th, there are multiple colored boxes representing different leave requests: a blue box for "aditi@gmail.com", a green box for "lux@gmail.com", a red box for "aishu1@gmail.com", and a green box for "swarna1@gmail.com". The 19th and 20th also show leave requests for "ranju1@gmail.com" and "ranju@gmail.com" respectively.

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 aditi@gmail.com lux@gmail.com aishu1@gmail.com swarna1@gmail.com	19 ranju1@gmail.com	20 ranju@gmail.com

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
			<div>aditi@gmail.com</div> <div>lux@gmail.com</div> <div>ashu1@gmail.com</div> <div>swarna1@gmail.com</div> <div>ranju1@gmail.com</div> <div>ranju@gmail.com</div>			
21	22	23	24	25	26	27
<div>aditi@gmail.com</div>				<div>harshita@gmail.com</div>		

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]

Home

Pending
Accepted
Rejected

☒ Pending
☒ Accepted
☒ Rejected

## January 2026

Sun	Mon	Tue	Wed	Thu	Fri	Sat
28	29	30	31	1	2	3
				<div>ravi@gmail.com</div>		
4	5	6	7	8	9	10
<div>aditi1@gmail.com</div>						
11	12	13	14	15	16	17