

Project Title: Leave Management System

Project Description:

This project is aimed at developing an online Leave Management System that is of importance to an organization. The Leave Management System (LMS) is an Internet based application that can be accessed throughout the organization. This system can be used to automate the workflow of leave applications and their approvals.

Functional Requirements:

- The System should have a provision for Employee/Manager to login.
- The System should allow the Employee/Manager to view the leave history details.
- System should allow the Employee/Manager to view the public Holidays list.
- System should allow the Manager to approve/reject the Leaves of the Employee.
- System should allow the Employee to withdraw the leave (which has not been approved).
- System should allow the Employee to cancel the leave (which has been already approved).
- System should have a provision for Manager to view the Employee Details & their Leaves

Entity class structure:

- **Employee** -> empld, firstName, lastName, job, mobile, email, managerId
 - **managerId** is the empld, indicates manager of an employee whom to report all activities.
 - **empld** is to be inputted by following first 4 letter of the company, last 2 digits of the year joined and 3 digits of code eg: **SUTH25001** where
 - **SUTH** for Sutherland, **25** for 2025 joined and **001** series.
- **Employee** table details should be stored along with their mangerId in the employee table after creation. Keep some sample employee records(with meaningful data) in the table for processing.
- **PublicHolidays** -> id (auto-increment), date, holidayDetails.
 - Keep some data in the table about the upcoming holiday details based on the date this task starts.
 - Public Holidays - only view operation is there.
- **LeaveRequest** -> id (auto-increment), empld, managerId, fromDate, toDate, leaveType (CASUAL/MEDICAL), numberOfDays, dateApplied, leaveStatus (APPLIED/APPROVED/REJECTED/CANCELLED/WITHDRAWN), managerId, remarks (reason for rejection is mandatory).
 - Number of days to be calculated based on fromDate and toDate.

- dateApplied is system date
- leave status should be APPLIED when the leave request submitted and it should be inserted automatically.
- Leave status should be updated as approved/rejected when verified by the concerned manager
- Leave status should be updated as CANCELLED/WITHDRAWN when employee cancel/withdraw the leave request.

Following are the endpoints of this application to be present:

- /addemployee
- /getemployeebyid
- /addholidaydetails
- /viewholidaylist
- /applyleaverequest
- /verifyleaverequest
- /checkleaverequeststatus
- /cancelleave
- /withdrawleave
- /viewleavehistory

- ** Identify the relevant exceptions to be handled.

Technologies to be used:

- Spring Boot Data JPA, Oracle, React JS

Expectations for submission:

- Application must contain the proper coding standards like class, method, data members, package names etc., layered architecture.
- Related exceptions must be handled wherever needed.
- Error and prompt messages need to be clearly defined wherever necessary.
- Proper alignment must be maintained for the UI
- Do not use the graphics/CSS in the UI
- Bootstrap can be allowed.
- **Do not copy the code from any AI tools.

Capstone Evaluation Parameters

Your project will be evaluated based on the following parameters:

- Understanding of requirements
- Implementation of layered architecture
- Adherence to coding standards
- Exception handling

- UI/UX design
- Overall application performance and execution
- Viva (oral evaluation)

XX