# Project Title

An Employee Leave Management System

# Project Outline

Develop an Employee Leave Management System where employees can apply for leaves, and their managers can approve/deny the leaves.

# Functional Requirements

1. Sign Up Page:
   1. Employees should be able to sign up using a sign-up form
   2. This form should take in a password that the employee can use to login later
   3. This form should generate an employee id for the employees
   4. The login page and the sign-up page are the only pages that should be visible to un authenticated users.
   5. The signup form should include relevant fields that would be required for the proper functioning of the application.
2. Admin Page
   1. There should be special user named admin whose credentials should be hard coded in the database
   2. The admin page should only be accessible by the admin user.
   3. Assign appropriate roles and access control for this feature.
   4. This page should feature a form which contains two fields: Manager ID and Employee ID
   5. This form should assign a manager to an employee.
3. Managers and Employees:
   1. Managers and Employees should both be employees.
   2. The Employee entity should have two fields.
      1. One for ManagerID
      2. Other for SubordinateIDs
   3. An employee can have one Manager but multiple subordinates
   4. The manager is assigned by the ManagerID and the subordinates using the SubordinateIDs
4. Apply for Leave Page:
   1. An employee can apply for leave on this page
   2. This page should have a form which will take a start date and end date for the leave.
   3. There should also be a field which takes in the reason for the leave.
5. Approve Leaves Page:
   1. On this page an employee should see the leaves requested by his/her subordinates.
   2. It should show the reason, start date, and end date for the leave.
   3. There should be buttons to approve/deny the request.
6. Requests Page:
   1. On this page an employee should see the list of leaves requested by the employee along with the status.
   2. If a leave has been approved by the manager this should show status as approved

# Creative Freedom

The functional requirements define abstract functionalities. There are multiple ways to implement the functionalities discussed above. This project gives you the creative freedom to choose your data structures for the project.

Anything not mentioned in the functional requirements, but necessary to develop the complete application can be added in by the student.

The student is supposed to make educated decisions on how to place different elements for different functionalities so as to display a better UI design.

# Marking Scheme

You will be marked on the following parameters:

1. The complete application should be developed using the Microservices Architectural Style. You will be scored on the domains that you select for your Microservices.
2. Package names should confirm to the Layered Architecture or Domain Driven Design. You will be scored on your choice of package names.
3. You will be given marks for following Industry Best Practices.
4. Your APIs should be RESTful and should implement proper mappings for CRUD Operations. You will be scored for proper mappings.
5. You should use Based Authentication and Authorization in your application. You will be scored for proper implementation.
6. You will be scored for proper implementation of Zuul Proxy and Eureka Discovery Service.
7. Finally, you will be scored for all the functionalities defined in this document.

# Submission

1. Create a new STS Workspace
2. Develop all the projects in this workspace
3. Compress the workspace folder as a .zip file
4. Rename the folder to **p2\_yourFullName.zip** (If you don’t follow this naming convention, then it will become hard to score your project)
5. Submit it to the appropriate person