Employee Management system

Course Name: PG-DAC

Members:

S.NO.	Name	Roll No.
1.	AMIT MISHRA	200250120005
2.	GARIMA JOSHI	200250120031
3.	NIKHIL BHATI	200250120058
4.	PRANAV KUMAR ZADE	200250120069
5.	VARUN MALPOTRA	200250120105

Project Details

1. The Web based project “Employee Management System” is an attempt to stimulate the basic operations of a general management dashboard.
2. EMPMS is project comprising 3 technologies together giving it a dynamic build to perform the important operations in a management system.
3. 3 level of access is provided: Admin, Manager and Employee
4. The three technologies used are
    a. Angular as client side,
    b. Nodejs as server-side handling login
    c. J2EE (Spring-boot with JPA) in server side
5. The operations that can performed are
    a. Add Employee
    b. Select Employee
    c. Update Employee
    d. Add Project
    e. Add Team Members
    f. Manage Leave Requests
    g. Manage Resource requests
    h. Progress Dashboard
6. Structure of Client side is as:
    a. ADMIN
        i. Add Employee
        ii. Project
        iii. Notification
        iv. Leave Request
        v. Resource Request
        vi. Sidenav
    b. EMPLOYEE
        i. Dashboard
        ii. Teams
        iii. Resource Requests
        iv. Leave Requests
        v. Sidenav
    c. MANAGER
        i. Dashboard
        ii. Leave requests
        iii. Add project member
        iv. Raise Tickets
        v. Resource Requests
        vi. Sidenav
    d. TOPNAV
        i. Notification
        ii. Logout
        iii. Chat
    e. LOGIN

Scenarios:

1) Add Employee
    (a) The UI components of add employee when clicked on **add button** will call click event registered with function of **addemp**,
    (b) **addemp** function defined in **aemployee.component.ts** will first check whether given employee id is 3 digit or not if yes then it will pass the registration to the next it will check role of employee is manager then it will assign manager id and insert in manager table also.
    (c) When **insertEmp service** (registered in adminservice.service.ts) is invoked in the **addemp** function it will call the service function in spring boot and send data object via post mode through `http://localhost:6500/employees_B`
    (d) The service layer function of **insert employee by id** is called and all the data fields which are filled gets stored in database;
2) Add Project
    (a) The Client details in the **project.component.ts** gets inserted after that the project details can be inserted and free available managers can be assigned to particular projects. For all these functionalities there is a function registered **in project.component.ts** file
    (b) The submit button will call **addproject** function from **admin.service.ts** which will insert call insertproject service registered in adminservice.service.ts with the call of function from spring boot and send data object via post mode through `http://localhost:6500/project_B`

Scenarios where we got struct:

1) Session Handling:
    (a) Sessions are handled securely but got struct because server is hosted locally and we are getting only single instance of session.
2) Delete Function
    a) Due to foreign key Violations the delete functions is unable to delete the table entries from database.
    b) This problem can be overcome by creating a trigger which sequentially first delete all the foreign key entities and then delete primary key of entity table.
3) Chats registration
    a) Chat room registration poc is made successfully but cannot be applied in angular.
4) In WebRTC
    a) Successfully completed poc on WebRTC but because we are not using centralized media servers, so one client is connected to multiple other clients directly, which was talking too much bandwidth therefore video calling experience was not good.
5) We didn't know how to use observables; this issue is overcome by reading angular documentation RxJS.

Learnings:

1) Agular, JPA
2) Session Handling:
    a) How to apply sessions using session node – session modules.
3) Chats using Socket.io
    a) Learnt managing chat sessions with multiple instances.
4) Applying Design system in angular
5) How to use WebRTC which is a functionality of browser used for multimedia transfer and video calling.
6) Learnt how to communicate with components of angular.