**EMPLOYEE REGISTRATION PROJECT-DOCUMENTATION**

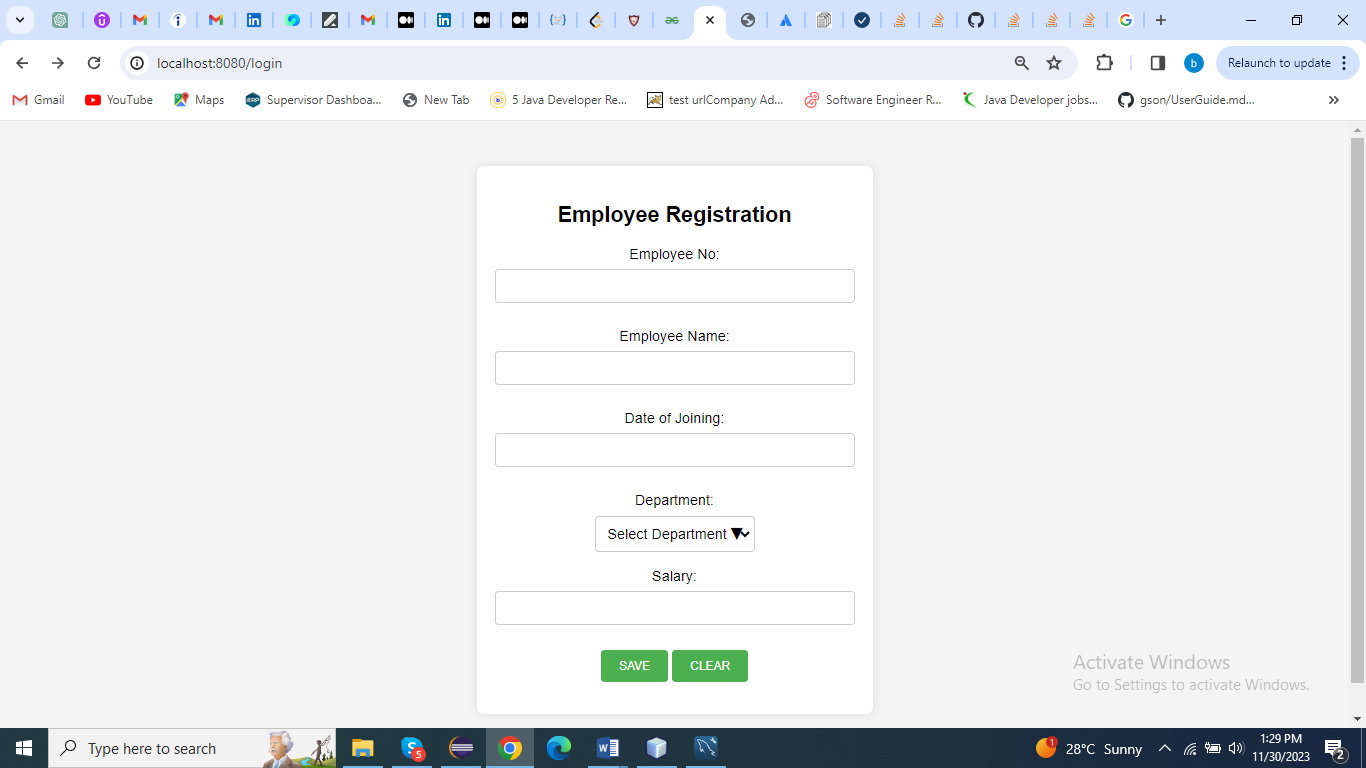
The Employee Registration Project is a web application designed to facilitate the registration of employees within an organization. The project utilizes Spring Boot for backend development, JSP for frontend, and REST APIs for communication between the two.

**Technologies Used**

* Spring Boot for back end development
* Java Server Pages(JSP) with CSS styling for frontend
* RESTful APIs
* Java In memory database-H2

**Communication Flow**

* User interacts with JSP based registration form
* JSP sends Restful requests to spring boot back end
* Spring boot process the request and interacts with the database
* Responses are sent to jsp for rendering



H2 Database

Repository

Controller

Service

**Setup and Installation**

Registration-form/

|-- src/

| |-- main/

| |-- java/

| |-- com.employee.registrationform/

| |-- Advice/

| |-- controller/

| |-- service/

| |-- repository/

| |-- Entity/

| |-- Exception

| |-- resources/

| |-- application.properties

| |-- data.sql

| |-- webapp/

| |-- WEB-INF/

| |-- jsp/

| |-- registration.jsp

| |-- success.jsp

|-- pom.xml

**Dependencies**

* Spring Boot Starter web
* Spring Boot Starter Data JPA
* Spring Boot DevTools
* Spring Boot Starter Validation
* Tomcat Ember Jasper(for JSP)
* H2(for h2 in memory database)
* Jstl (jsp)

**Data base Design**

Tables

* Employee

Columns:id,employee\_no,employee\_name,date\_of\_joining,department,salary

* Department

Columns:id,code,departmentName

**Rest API Documentation**

**Save Registration details**

Request

* Method:post
* URL: /addRegistration
* Body: {

"empNo": "46",

"empName": "Name of Employee",

"joiningDate": "2023-12-12",

"department": 1,

"salary": 400.00

}

* Response:201

**Search Employee By Employee number**

Request

* Method:get
* URL: searchEmployeeByNumber/EmployeeNo/{empNo}
* Response:200

**Search Employee By Employee name**

Request

* Method:get
* URL: searchEmployeeByName/EmployeeName/{empName}
* Response:200

**Get Employee Details By Id**

Request

* Method:get
* URL: searchEmployeeById/EmployeeId/{id}
* Response:200

**Get All Employee Details**

Request

* Method:get
* URL: fetchAllEmployees
* Response:200

**Delete Employee By Id**

Request

* Method:delete
* URL: deleteEmployeeById/id/{id}

**Update Employee By Id**

Request

* Method:put
* URL: updateEmployeeDetails
* Response:200
* Body: {

"empNo": "46",

"empName": "Name of Employee",

"joiningDate": "2023-12-12",

"department": 1,

"salary": 400.00

}

**Search Employee By Department**

Request

* Method:get
* URL: searchEmployeeByDepartment/departmentId/{id}
* Response:200