**Searching for a Specific User and Updating the User Information**

**Project objective:**

Create a Spring MVC web application that will retrieve users based on their user ID. The retrieved user data will then be edited in a form and updated in the database. The entire database processing has to be done using Hibernate. Front pages will be made in JSP.

**code**

**Pom.xml**

<project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4\_0\_0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>0001</groupId>

<artifactId>SpecificUserandUpdatingtheUserInformation</artifactId>

<packaging>war</packaging>

<version>0.0.1-SNAPSHOT</version>

<name>SpecificUserandUpdatingtheUserInformation Maven Webapp</name>

<url>http://maven.apache.org</url>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>4.2.9.RELEASE</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-jdbc -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-jdbc</artifactId>

<version>4.2.9.RELEASE</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-orm -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-orm</artifactId>

<version>4.2.9.RELEASE</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-tx -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-tx</artifactId>

<version>4.2.9.RELEASE</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-webmvc -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>4.2.9.RELEASE</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-core -->

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>3.6.9.Final</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-entitymanager -->

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-entitymanager</artifactId>

<version>3.6.9.Final</version>

</dependency>

<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>8.0.19</version>

</dependency>

<!-- https://mvnrepository.com/artifact/commons-dbcp/commons-dbcp -->

<dependency>

<groupId>commons-dbcp</groupId>

<artifactId>commons-dbcp</artifactId>

<version>1.4</version>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<configuration>

<source>1.8</source>

<target>1.8</target>

</configuration>

</plugin>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-war-plugin</artifactId>

<version>3.3.1</version>

</plugin>

</plugins>

<finalName>SpecificUserandUpdatingtheUserInformation</finalName>

</build>

</project>

Config.java

package com.example;

import org.springframework.context.annotation.ComponentScan;

import org.springframework.context.annotation.Configuration;

@Configuration

@ComponentScan(basePackages = {"com.example"})

public class config {

}

**Intializer.java**

**package** com.example;

**import** org.springframework.web.servlet.support.AbstractAnnotationConfigDispatcherServletInitializer;

**public** **class** intializer **extends** AbstractAnnotationConfigDispatcherServletInitializer {

@Override

**protected** Class<?>[] getRootConfigClasses() {

// **TODO** Auto-generated method stub

**return** **null**;

}

@Override

**protected** Class<?>[] getServletConfigClasses() {

// **TODO** Auto-generated method stub

**return** **new** Class[] {config.**class**};

}

@Override

**protected** String[] getServletMappings() {

// **TODO** Auto-generated method stub

**return** **new** String[] {"/"};

}

}

**Employee.java**

package com.example;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

import javax.persistence.Table;

@Entity

@Table

public class Employee {

@Id

@GeneratedValue

private int eid;

private String ename;

private String email;

public int getEid() {

return eid;

}

public void setEid(int eid) {

this.eid = eid;

}

public String getEname() {

return ename;

}

public void setEname(String ename) {

this.ename = ename;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

**}**

**EmployeeDAo.java**

package com.example;

import com.example.Employee;

import java.util.List;

import org.springframework.orm.hibernate3.HibernateTemplate;

import com.sun.net.httpserver.Request;

public class EmployeeDao {

private HibernateTemplate temp;

public void setTemp(HibernateTemplate temp) {

this.temp = temp;

}

//search

public List<Employee> getallemployees(){

String sql="from Employee";

return (List<Employee>) temp.find(sql);

}

//update

public int update(Employee e) {

temp.update(e);

return 1;

}

}

**EmployeeContoller.java**

import java.io.IOException;

import java.io.PrintWriter;

import java.util.List;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.servlet.ModelAndView;

@Controller

public class EmployeeContoller {

public static String id;

@RequestMapping("/search")

public ModelAndView search(HttpServletRequest request, HttpServletResponse response) {

ModelAndView mv=new ModelAndView();

ApplicationContext ac=new ClassPathXmlApplicationContext("spring.xml");

Employee e=ac.getBean(Employee.class);

EmployeeDao edao=ac.getBean(EmployeeDao.class);

List<Employee> elist=edao.getallemployees();

System.out.println(elist);

mv.setViewName("display.jsp");

mv.addObject("list", elist);

id=request.getParameter("eid");

return mv;

}

@RequestMapping("/update")

public ModelAndView update(HttpServletRequest request, HttpServletResponse response) {

ModelAndView mv=new ModelAndView();

ApplicationContext ac=new ClassPathXmlApplicationContext("spring.xml");

Employee e=ac.getBean(Employee.class);

EmployeeDao edao=ac.getBean(EmployeeDao.class);

e.setEname(request.getParameter("ename"));

e.setEmail(request.getParameter("eemail"));

e.setEid(Integer.parseInt(request.getParameter("eid")));

int row=edao.update(e);

if(row>0) {

mv.setViewName("index.jsp");

}

return mv;

}

}

**spring.xml**

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<beans xmlns = *"http://www.springframework.org/schema/beans"*

xmlns:xsi = *"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation = *"http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans-3.0.xsd"*>

<bean id=*"employee"* class=*"com.example.Employee"*></bean>

<bean id=*"dao"* class=*"com.example.EmployeeDao"*>

<property name=*"temp"* ref=*"hibernatetemp"* ></property>

</bean>

<!-- HibernateTemplate with session factory-->

<bean id=*"hibernatetemp"* class=*"org.springframework.orm.hibernate3.HibernateTemplate"*>

<property name=*"sessionFactory"* ref=*"sefref"*></property>

</bean>

<!-- session factory with the datasource -->

<bean id=*"sefref"* class=*"org.springframework.orm.hibernate3.annotation.AnnotationSessionFactoryBean"*>

<property name=*"dataSource"* ref=*"ds"*></property>

<property name=*"packagesToScan"* value=*"com.example"*></property>

<property name=*"hibernateProperties"*>

<props>

<prop key=*"hibernate.dialect"*> org.hibernate.dialect.MySQLDialect</prop>

<prop key=*"hibernate.hbm2ddl.auto"*>update</prop>

<prop key=*"hibernate.show\_sql"*>true</prop>

</props>

</property>

</bean>

<bean id=*"ds"* class=*"org.apache.commons.dbcp.BasicDataSource"*>

<property name=*"driverClassName"* value=*"com.mysql.cj.jdbc.Driver"*></property>

<property name=*"url"* value=*"jdbc:mysql://localhost:3306/springdemo"*></property>

<property name=*"username"* value=*"root"*></property>

<property name=*"password"* value=*"root"*></property>

</bean>

</beans>

**Index.jsp**

<html>

<body>

<h2>Employee Search</h2>

<form action=*"search"*>

Employee Id <input type=*"number"* name=*"eid"*><br>

<input type=*"submit"* value=*"search"*>

</form>

</body>

</html>

**display.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<%@ page import=*"java.util.\*"* %>

<%@ page import=*"com.example.\*"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<%

List<Employee> list=(List<Employee>)request.getAttribute("list");

**int** i=0;

%>

<table border=*"1"*>

<tr><th>Employee id</th><th>Employee name</th><th>Employee email</th><th>Edit</th></tr>

<%**for**(Employee e:list){

**if** (e.getEid()==Integer.parseInt(EmployeeContoller.id)){ i+=1; %>

<tr><td><%=e.getEid() %></td><td><%=e.getEname() %></td><td><%=e.getEmail() %></td><td><a href=*"update.jsp?id=*<%=e.getEid() %>*"*>Edit</a></td></tr>

<%}

}%>

</table>

<% **if**(i==0) {%>

<jsp:forward page=*"failure.jsp"*></jsp:forward>

<% } %>

</body>

</html>

**failure.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<h1>invalid id</h1>

</body>

</html>

**update.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<h1>update</h1>

<form action =*"update"*>

<input type=*"hidden"* name=*"eid"* value=*"*<%= request.getParameter("id") %>*"*>

enter new name <input type=*"text"* name=*"ename"* >

enter new email <input type=*"text"* name=*"eemail"* >

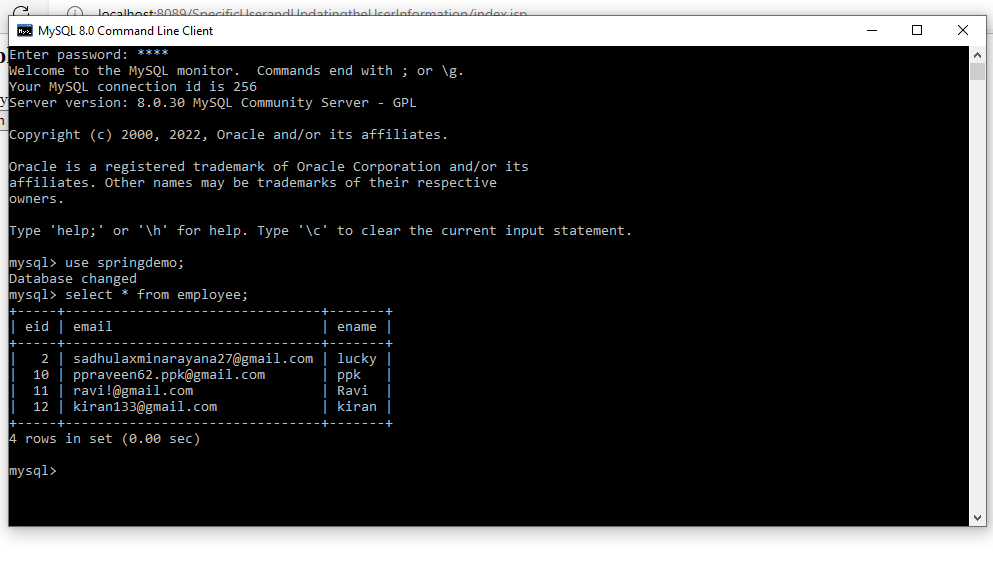
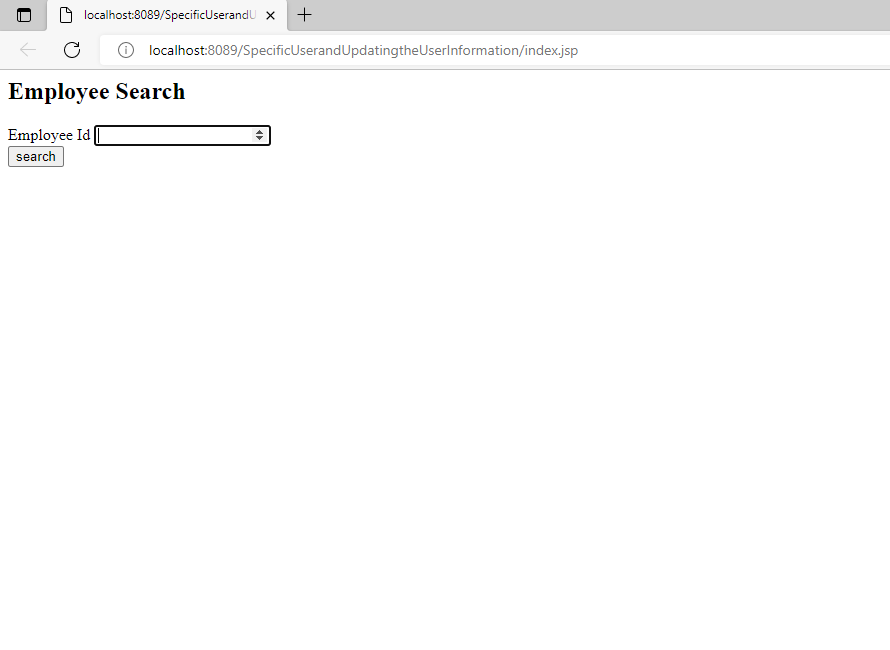
<input type =*"submit"* value=*"ok"*>

</form>

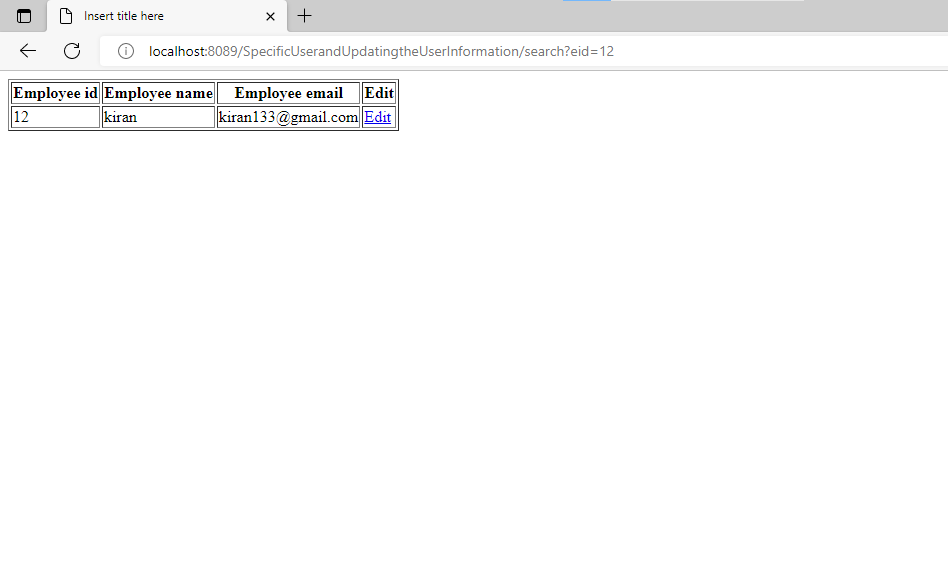
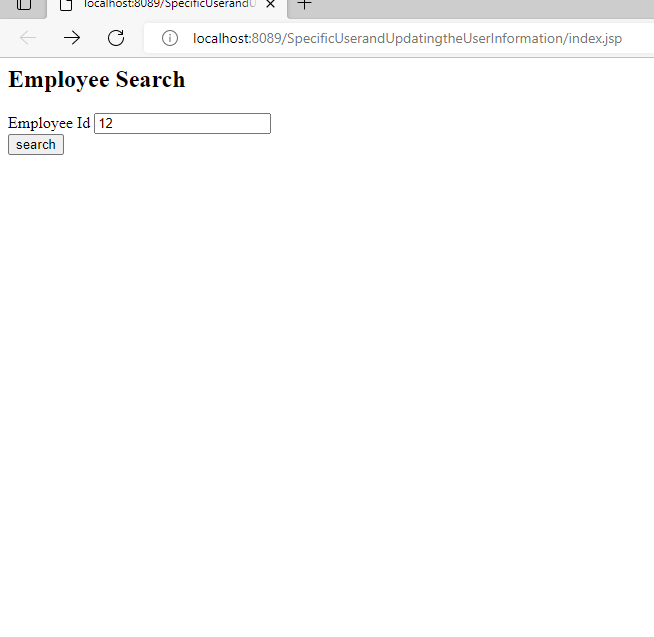
</body>

</html>

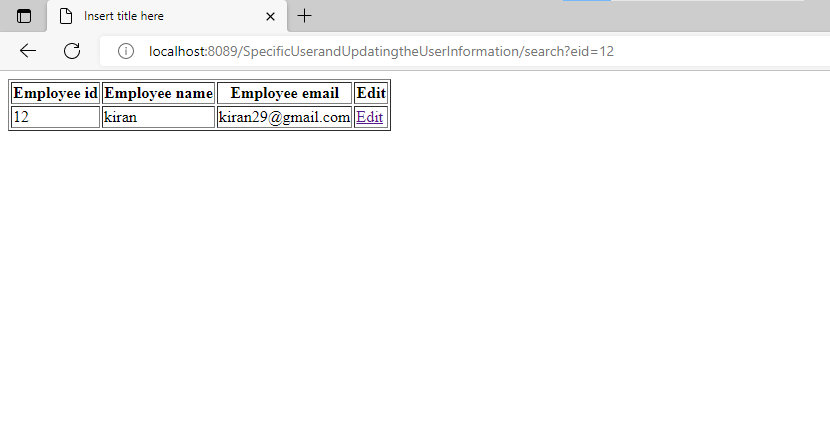
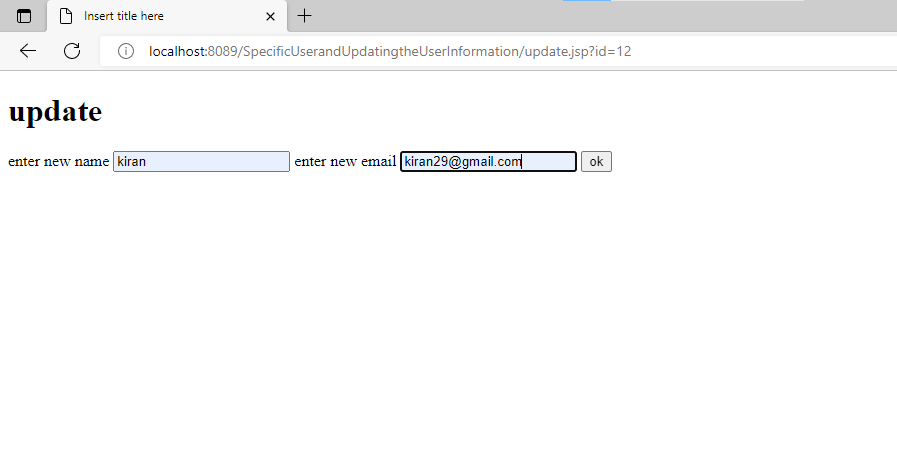
**Create a JSP page to take in a user ID**



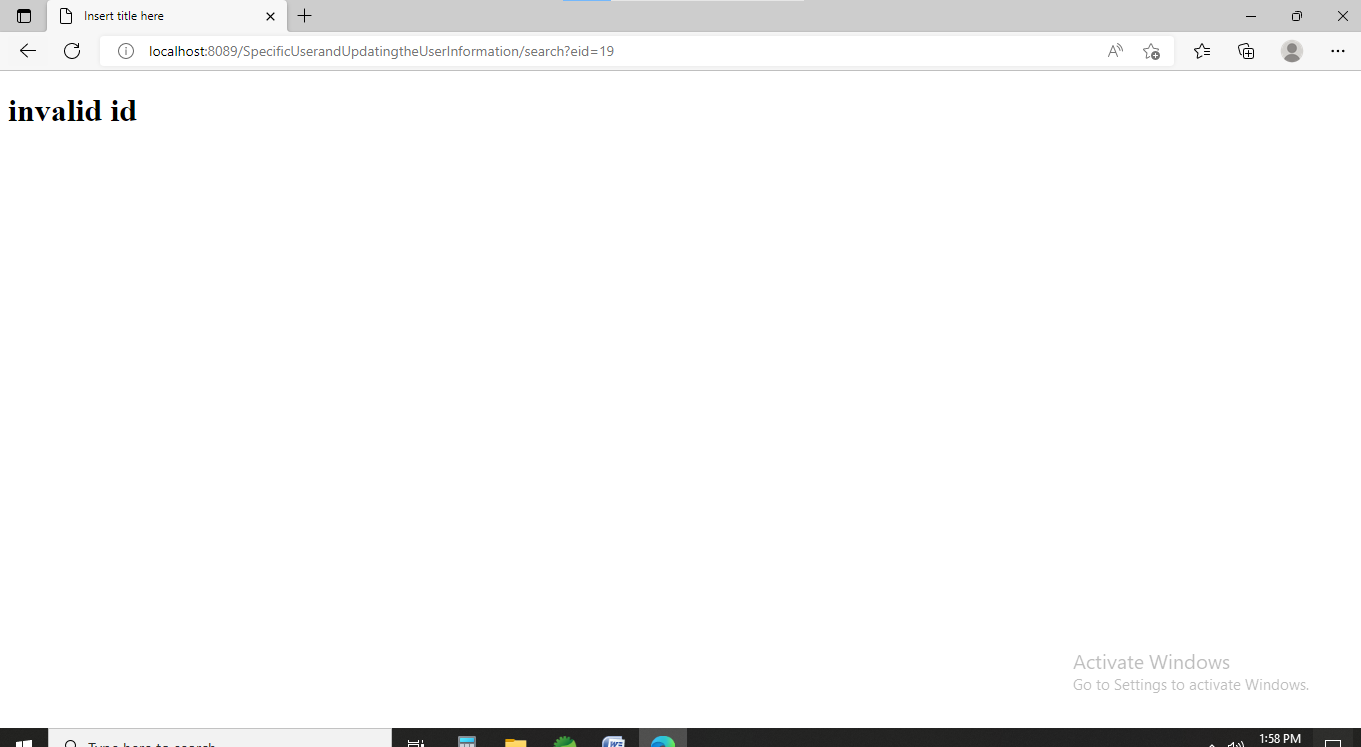
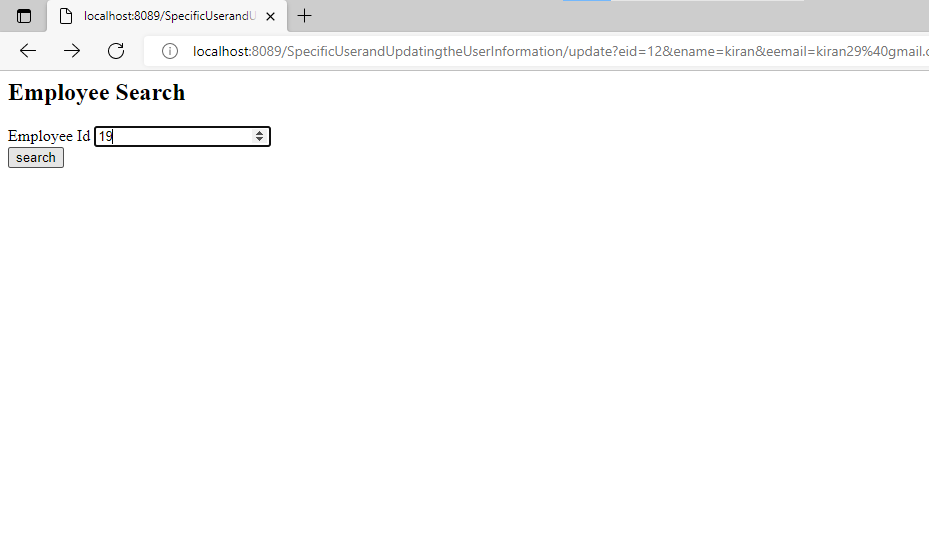
**If it is valid, it will retrieve user details from the database and show an edit form in JSP**



**Once the edit form is submitted, the controller will update the details in the database.**



**If it is invalid, it will display a JSP page with an error message.**



**Pushing code to github**