**Source code for user feedback:**

Index.jsp:

<html>

<head>

<style>

*.center* {

text-align: *center*;

align-items:*center*;

}

**a**{

text-decoration:*None*;

color:*black*;

}

</style>

</head>

<body>

<div class=*"center"*>

<h1>User Feedback</h1>

<h2 class=*"hello-title"*>Search for a User and Update Information</h2>

<br><br>

<form method=*"post"* action=*"update"*>

Enter id: <input type=*"text"* id=*"id"* name=*"id"* placeholder=*"Type here"* required>

<br>

<br>

<input type=*"submit"* value=*"Enter"* />

<button><a href=*"/feedback"*>View all feedback</a></button>

</form>

</div>

</body>

</html>

Application Properties:

spring.jpa.hibernate.ddl-auto=update

spring.datasource.url=jdbc:mysql://${MYSQL\_HOST:localhost}:3306/userdb

spring.datasource.username=root

spring.datasource.password=Mohan@4635

spring.mvc.view.prefix=/webapp/jsp/

spring.mvc.view.suffix=.jsp

server.port=8090

Feedback controller.java:

**package** com.example.UserFeedback.controllers;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.ui.ModelMap;

**import** org.springframework.web.bind.annotation.GetMapping;

**import** org.springframework.web.bind.annotation.PathVariable;

**import** org.springframework.web.bind.annotation.RequestParam;

**import** org.springframework.web.bind.annotation.ResponseBody;

**import** org.springframework.web.bind.annotation.RestController;

**import** com.example.UserFeedback.entities.Feedback;

**import** com.example.UserFeedback.repositories.FeedbackRepository;

**import** com.example.UserFeedback.services.FeedbackService;

@RestController

**public** **class** FeedbackController {

@Autowired

FeedbackService feedbackService;

@Autowired

FeedbackRepository feedbackRepository;

@GetMapping(value="/")

**public** String showIndexPage(ModelMap model){

**return** "<html>\n"

+ "<head>\n"

+ " <style>\n"

+ " .center {\n"

+ " text-align: center;\n"

+ " }\n"

+ " a {\n"

+ " text-decoration: None;\n"

+ " color:black;\n"

+ " }\n"

+ " #comment, #rating, #name {\n"

+ " padding: 16px;\n"

+ " margin:4px;\n"

+ " border-radius:4px;\n"

+ " }\n"

+ " \n"

+ " </style>\n"

+ "</head>\n"

+ "<body>\n"

+ " <div class=\"center\">\n"

+ " <h1>User Feedback Page</h1>\n"

+ " \n"

+ " \n"

+ " <br><br>\n"

+ " <form method=\"get\" action=\"update\">\n"

+ " <br><h3>Enter your feedback below:</h3>\n"

+ " <input type=\"text\" id=\"comment\" name=\"comment\" placeholder=\"Enter Comment Here...\" required>\n"

+ " <input type=\"number\" id=\"rating\" name=\"rating\" placeholder=\"Enter Rating Here...\" required>\n"

+ " <input type=\"text\" id=\"name\" name=\"name\" placeholder=\"Enter UserName Here...\" required> <br><br> \n"

+ " <input type=\"submit\" value=\"Submit\" /> &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp; \n"

+ " <button><a href=\"/feedback\">View all feedback</a></button>\n"

+ " </form>"

+ " </div>\n"

+ "</body>\n"

+ "</html>";

}

@GetMapping("/feedback")

**public** @ResponseBody String getAllFeedbacks() {

Iterable<Feedback> allFB = feedbackRepository.findAll();

**return** "<html>\n"

+ "<head>\n"

+ " <style>\n"

+ " .center {\n"

+ " text-align: center;\n"

+ " align-items: center;\n"

+ " }\n"

+ " \n"

+ " </style>\n"

+ "</head>\n"

+ "<body >\n"

+ " <div class=\"center\">\n"

+ "<h1>Feedback Table</h1>\n"

// + "<table border='1'>"+"<tr><th>"+"UserName"+"</th>"+"<tr><th>"+"Id"+"</th>"+"<tr><th>"+"Comments"+"</th>"+"<tr><th>"+"Rating"+"</th></tr>"

+ allFB.toString()

+ " </div>\n"

+ "</body>\n"

+ "</html>";

}

}

Rest update controller.java:

**package** com.example.UserFeedback.controllers;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.web.bind.annotation.GetMapping;

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

**import** org.springframework.web.bind.annotation.RequestMethod;

**import** org.springframework.web.bind.annotation.RequestParam;

**import** org.springframework.web.bind.annotation.RestController;

**import** com.example.UserFeedback.entities.Feedback;

**import** com.example.UserFeedback.repositories.FeedbackRepository;

**import** com.example.UserFeedback.services.FeedbackService;

@RestController

**public** **class** RestUpdateController

{

@Autowired

FeedbackService feedbackService;

@Autowired

FeedbackRepository feedbackRepository;

@GetMapping("/update")

**public** String getEmployeeByName(@RequestParam("comment") String comment, @RequestParam("rating") **int** rating, @RequestParam("name") String name) {

Feedback f = **new** Feedback(count()+1, comment, rating, name);

feedbackRepository.save(f);

**return** "<html>\n"

+ "<head>\n"

+ " <style>\n"

+ " .center {\n"

+ " text-align: center;\n"

+ " }\n"

+ " \n"

+ " </style>\n"

+ "</head>\n"

+ "<body>\n"

+ " <div class=\"center\">\n"

// + " <h1>User Feedback Page</h1>\n"

+ " \n"

+ " <h2 class=\"hello-title\">Successfully Added Your Feedback</h2>\n"

+ " \n"

+ " <button><a href=\"/feedback\">Click here to view all feedback</a></button>\n"

+ " </div>\n"

+ "</body>\n"

+ "</html>";

}

**public** Integer count() {

**int** i = 1;

**while**(feedbackRepository.existsById(i))

i++;

**return** i;

}

}

Feedback.java:

**package** com.example.UserFeedback.entities;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.GenerationType;

**import** javax.persistence.Id;

**import** lombok.Data;

@Entity

@Data

**public** **class** Feedback {

@Id

@GeneratedValue(strategy=GenerationType.***AUTO***)

**private** Integer id;

**private** String comments;

**private** **int** rating;

**private** String user;

**public** Feedback()

{

}

**public** Feedback(Integer id, String comments, **int** rating, String user)

{

**this**.id = id;

**this**.comments = comments;

**this**.rating = rating;

**this**.user = user;

}

@Override

**public** String toString()

{

// return "<br><h3>" + user + " [" + id + "]" + " commented:</h3><h4>\"" + comments + "\" and rated: " + rating + "</h4><br>";

**return**(

"<style>\n" +

"table {\n" +

" font-family: Arial, Helvetica, sans-serif;\n" +

" border-collapse: collapse;\n" +

" width: 100%;\n" +

"}\n" +

"\n" +

"table td, table th {\n" +

" border: 1px solid #ddd;\n" +

" padding: 8px;\n" +

"}\n" +

"\n" +

"table tr:nth-child(even){background-color: #f2f2f2;}\n" +

"\n" +

"table tr:hover {background-color: #ddd;}\n" +

"\n" +

"table th {\n" +

" padding-top: 12px;\n" +

" padding-bottom: 12px;\n" +

" text-align: left;\n" +

" background-color: #16E2F5;\n" +

" color: white;\n" +

"}\n" +

"</style>"

+ "<table border='2' style='margin-left: auto;\n" +

" margin-right: auto;width:50%;'>"

+"<tr>"

+"<th> UserName </th>"

+"<th> Id </th>"

+"<th> Comments </th>"

+"<th> Rating </th>"

+"</tr>"

+"<td>"+user+"</td>"

+"<td>"+id+"</td>"

+"<td>"+comments+ "</td>"

+"<td>" +rating+ "</td>"

+"</tr>"

+"</table>");

}

}

Feedback repository.java:

**package** com.example.UserFeedback.repositories;

**import** org.springframework.data.repository.CrudRepository;

**import** com.example.UserFeedback.entities.\*;

**public** **interface** FeedbackRepository **extends** CrudRepository<Feedback, Integer>{

}

Feedback service.java:

**package** com.example.UserFeedback.services;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.stereotype.Service;

**import** com.example.UserFeedback.entities.Feedback;

**import** com.example.UserFeedback.repositories.\*;

@Service

**public** **class** FeedbackService {

@Autowired

**private** FeedbackRepository feedbackRepository;

**public** Iterable<Feedback> GetAllFeedback() {

**return** feedbackRepository.findAll();

}

}

Userfeedback Application.java:

**package** com.example.UserFeedback;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

**public** **class** UserFeedbackApplication {

**public** **static** **void** main(String[] args) {

SpringApplication.*run*(UserFeedbackApplication.**class**, args);

}

}

Pom.xml:

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

<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 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>2.4.3</version>

<relativePath/> <!-- lookup parent from repository -->

</parent>

<groupId>com.example</groupId>

<artifactId>UserFeedback</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>UserFeedback</name>

<description>Demo project for Spring Boot</description>

<properties>

<java.version>1.8</java.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>mysql</groupId>

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

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.projectlombok</groupId>

<artifactId>lombok</artifactId>

<optional>true</optional>

</dependency>

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>jstl</artifactId>

<version>1.2</version>

</dependency>

<dependency>

<groupId>org.apache.tomcat.embed</groupId>

<artifactId>tomcat-embed-jasper</artifactId>

<scope>provided</scope>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-devtools</artifactId>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

<configuration>

<excludes>

<exclude>

<groupId>org.projectlombok</groupId>

<artifactId>lombok</artifactId>

</exclude>

</excludes>

</configuration>

</plugin>

</plugins>

</build>

</project>