**Displaying User Feedback**

**Source code**

**1. UserFeedbackApplication.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);

System.out.println("Running...");

}

}

**2.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>";

}

}

**3. FeedbackController.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"

+ " \n"

+ " </style>\n"

+ "</head>\n"

+ "<body style=\"background-color:lightpink;\">\n"

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

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

+ " \n"

+ " <h2 class=\"hello-title\">Send User Feedback</h2>\n"

+ " \n"

+ " <a href=\"/feedback\">View all feedback</a>\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=\"Comment Here\" required>\n"

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

+ " <input type=\"text\" id=\"name\" name=\"name\" placeholder=\"Name Here\" required> \n"

+ " <input type=\"submit\" value=\"Enter\" />\n"

+ " </form>"

+ " </div>\n"

+ "</body>\n"

+ "</html>";

}

@GetMapping("/feedback")

public @ResponseBody String getAllFeedbacks() {

// This returns a JSON or XML with the Feedbacks

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

return "<html>\n"

+ "<head>\n"

+ " <style>\n"

+ " .center {\n"

+ " text-align: center;\n"

+ " }\n"

+ " \n"

+ " </style>\n"

+ "</head>\n"

+ "<body style=\"background-color:lightpink;\">\n"

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

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

+ allFB.toString()

+ " </div>\n"

+ "</body>\n"

+ "</html>";

}

}

**4. RestUpdateController.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 style=\"background-color:lightpink;\">\n"

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

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

+ " \n"

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

+ " \n"

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

+ " </div>\n"

+ "</body>\n"

+ "</html>";

}

public Integer count() {

int i = 1;

while(feedbackRepository.existsById(i))

i++;

return i;

}

}

**5. FeedbackRepository.java**

package com.example.UserFeedback.repositories;

import org.springframework.data.repository.CrudRepository;

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

public interface FeedbackRepository extends CrudRepository<Feedback, Integer>{

}

**6. FeedbackService.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();

}

}

**8.application.properties**

spring.jpa.hibernate.ddl-auto=update

spring.datasource.url=jdbc:mysql://localhost:3306/feed

spring.datasource.username=root

spring.datasource.password=Root@1999

server.port=8080

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

spring.mvc.view.suffix=.jsp

server.port=8080

**9.index.html**

<!DOCTYPE html>

<html>

<head>

<script src=*"testform.js"*>

</script>

</head>

<body>

<h2>Feedback Test Form</h2>

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

<!-- <form onsubmit="SubmitTestForm()"> -->

<label for=*"user"*>Name:</label><br>

<input type=*"text"* id=*"user"* name=*"user"* required><br>

<label for=*"rating"*>Rating:</label><br>

<input type=*"text"* id=*"rating"* name=*"rating"* required><br>

<label for=*"comment"*>Comment:</label><br>

<input type=*"text"* id=*"comment"* name=*"comment"* required><br><br>

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

</form>

</body>

</html>

**10.view.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>View</title>

</head>

<body>

</body>

</html>

**11.testform.js**

**function** SubmitTestForm() {

//TODO: gather fields from form

//TODO: Jsonify form fields

//TODO: Call postFormDataAsJson to http://localhost:8090/your/endpoint

alert("The form was submitted");

}

/\*\*

\* Helper function for POSTing data as JSON with fetch.

\*

\* **@param** {Object} options

\* **@param** {string} options.url - URL to POST data to

\* **@param** {FormData} options.formData - `FormData` instance

\* **@return** {Object} - Response body from URL that was POSTed to

\*/

**async** **function** postFormDataAsJson({ url, formData }) {

/\*\*

\* We can't pass the `FormData` instance directly to `fetch`

\* as that will cause it to automatically format the request

\* body as "multipart" and set the `Content-Type` request header

\* to `multipart/form-data`. We want to send the request body

\* as JSON, so we're converting it to a plain object and then

\* into a JSON string.

\*

\* **@see** https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods/POST

\* **@see** https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\_Objects/Object/fromEntries

\* **@see** https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\_Objects/JSON/stringify

\*/

**const** plainFormData = Object.fromEntries(formData.entries());

**const** formDataJsonString = JSON.stringify(plainFormData);

**const** fetchOptions = {

/\*\*

\* The default method for a request with fetch is GET,

\* so we must tell it to use the POST HTTP method.

\*/

method: "POST",

/\*\*

\* These headers will be added to the request and tell

\* the API that the request body is JSON and that we can

\* accept JSON responses.

\*/

headers: {

"Content-Type": "application/json",

"Accept": "application/json"

},

/\*\*

\* The body of our POST request is the JSON string that

\* we created above.

\*/

body: formDataJsonString,

};

**const** response = await fetch(url, fetchOptions);

if (!response.ok) {

**const** errorMessage = await response.text();

throw new Error(errorMessage);

}

return response.json();

}