Displaying User Feedback

Writeup

1. Start
2. Set up a Spring Boot project with the necessary dependencies for building a web application and interacting with a MySQL database.
3. Create a **Feedback** entity class with fields for **id**, **username**, **email**, and **comment**. Annotate the class with **@Entity** and add appropriate annotations to mark the **id** field as the primary key.
4. Create a **FeedbackRepository** interface that extends **JpaRepository** to handle CRUD operations for the **Feedback** entity.
5. Create a **FeedbackController** class and annotate it with **@RestController** to define the REST API endpoints.
6. In the **FeedbackController**, inject the **FeedbackRepository** dependency.
7. Implement the **saveFeedback** method in the **FeedbackController** to handle the POST request for saving feedback. Use the **@PostMapping** annotation and receive the feedback details as parameters annotated with **@RequestParam**.
8. Inside the **saveFeedback** method, create a new **Feedback** object, set the received parameters, and save it using the **feedbackRepository.save()** method.
9. Return a success message as a response from the **saveFeedback** method by redirecting to **success.html**.

10. Implement the **getAllFeedbacks** method in the **FeedbackController** to handle the GET request for retrieving all feedbacks. Use the **@GetMapping** annotation.
11. Inside the **getAllFeedbacks** method, use the **feedbackRepository.findAll()** method to fetch all feedbacks from the database and return them as a **list.**
12. Create an HTML form (**index.html**) with input fields for **username**, **email**, and **comment**. Set the form's action to **/feedback** and method to **POST**.
13. In **success.html,** display **success message,** provide options to **return to Home**(index.html) and **show all feedbacks.**
14. Run the Spring Boot application and access the form page (**index.html**) using a web browser.
15. Enter the feedback details and submit the form.
16. The form data will be sent to the **/feedback** endpoint, and the **saveFeedback** method will be invoked. The feedback will be saved to the database, and a success message will be returned.
17. Upon successful submission, the user will be redirected to the **success.html** page, where they will see the success message.
18. End