Displaying User Feedback Write-Up

1. Configure the database: Create a MySQL database named "userfeedback" and define a table named "feedback" with the required columns (id, comments, rating, user).

2. Set up the project structure: Create a Maven project with the appropriate dependencies and directory structure.

3. Define the Feedback entity: Create the `Feedback` class with the required fields and annotations (`@Entity`, `@Id`, `@GeneratedValue`, `@Column`, etc.).

4. Create the FeedbackRepository: Define the `FeedbackRepository` interface, extending `CrudRepository`, to handle database operations for the Feedback entity.

5. Implement the FeedbackService: Create the `FeedbackService` class with methods for retrieving all feedback and adding new feedback. Autowire the `FeedbackRepository` for data access.

6. Implement the FeedbackController: Create the `FeedbackController` class with REST endpoints for retrieving all feedback and adding new feedback. Autowire the `FeedbackService` for handling feedback operations.

7. Create the TestFormController: Define the `TestFormController` class with endpoints for displaying a test form and submitting the form data. Autowire the `FeedbackService` for adding new feedback.

8. Set up JSP views: Create the required JSP files (index.jsp, post.jsp, testform.jsp) for rendering the web pages.

9. Configure application properties: Set the necessary configuration properties in the `application.properties` file, including database connection details, view configurations, and logging levels.

10. Run the application: Run the `FeedbackApplication` class to start the Spring Boot application.

11. Test the application: Access the provided URLs (e.g., http://localhost:8080/test_form) to interact with the application, fill in the form, submit feedback, and view the feedback data.

Note: Make sure you have the required dependencies added to the project, such as Spring Boot starters, MySQL connector, Spring Data JPA, Spring Web, and others mentioned in the `pom.xml` file.