**Displaying User Feedback**

**Projection Description:**

Create a Spring Boot project that will capture user feedback using a REST endpoint. The REST resource will take in parameters using HTTP POST. The feedback data will be then added to a database table.

**Background of the problem statement:**

As a part of developing an ecommerce web application, a REST resource is needed to capture user feedback. Feedback data will be received from third-party apps and websites. The data will be sent to the REST API which will collect feedback from various sources.

**The website needs to have the following features:**

● Create a MySQL table named feedback for storing feedback data  
● An entity class Feedback should be made with annotations to link it with the feedback table  
● A repository class should then map the entity class to the CrudRepository interface.  
● Create a REST controller class to create the REST endpoint. It should take in parameters using the POST protocol  
● Data received in the REST controller will be then saved into the database  
● Create a test form in HTML to submit data to the REST endpoint to ensure it’s working  
● The step-by-step process involved in completing this task should be documented.

**Project Users Stories**

As a full stack developer, I want to develop Sporty shoes ecommerce website which list out products for purchase,

* As a user, I want to website to enter feedback user interface screen.
* As a user, I want to submit the feedback which gets stored into the system.
* As a developer, I want to develop a website with text field which accepts the feedback from the users.
* As a developer, I want to create a POJO bean feedback which links the text box with feedback and database column.
* As a developer, I want to build controller which gets the feedback value from website and stores to the database.
* As a developer, I want to build restful webservice application to add, delete, update, list and delete feedback.
* As a developer, I want to build restful webservice application to list purchased items by category and date.

**Sprint 1 (Week 1)**

* As a user, I want to website to enter feedback user interface screen.
* As a user, I want to submit the feedback which gets stored into the system.
* As a developer, I want to develop a website with text field which accepts the feedback from the users.
* As a developer, I want to create a POJO bean feedback which links the text box with feedback and database column.
* As a developer, I want to build controller which gets the feedback value from website and stores to the database.
* As a developer, I want to build restful webservice application to add, delete, update, list and delete feedback.
* As a developer, I want to build restful webservice application to list purchased items by category and date.
* Testing the Java program with different kinds of User input.
* Initializing git repository to track changes as development progresses.
* Pushing code to GitHub.
* Creating this specification document with application details, appearance, and user interactions.

## **Core Java concepts used in project**

* + - Spring JPA concepts for database related operations,
    - Spring boot for rest API concepts.
    - JSP for webpage creation.
    - Spring MVC.

**Architecture diagram / flow chart:**

**User Operations,**

**Start**

**User enters the webpage and enters the feedback and submits the form.**

**Upon submit , feedback gets stored into the database table.**

**End**

**Admin Operations,**

**Start**

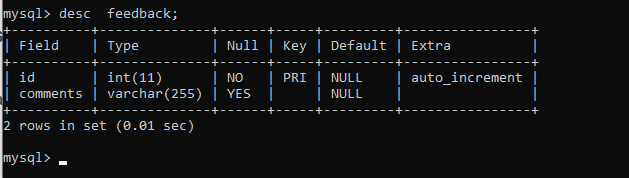
**Admin can add , delete , list and update feedback with postman tool with rest API services.**

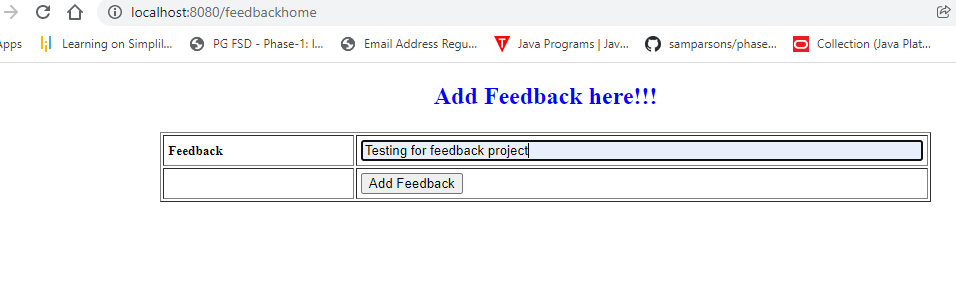
**Database gets updated based on the operations performed on the postman tool.**

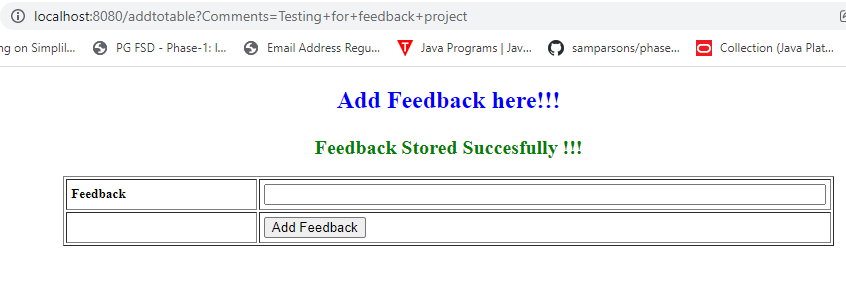
**End**

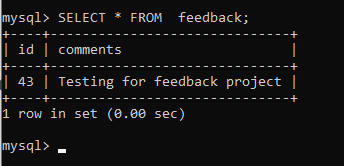
**Screenshots,**

* Start the spring boot and below database tables gets generated automatically,

****





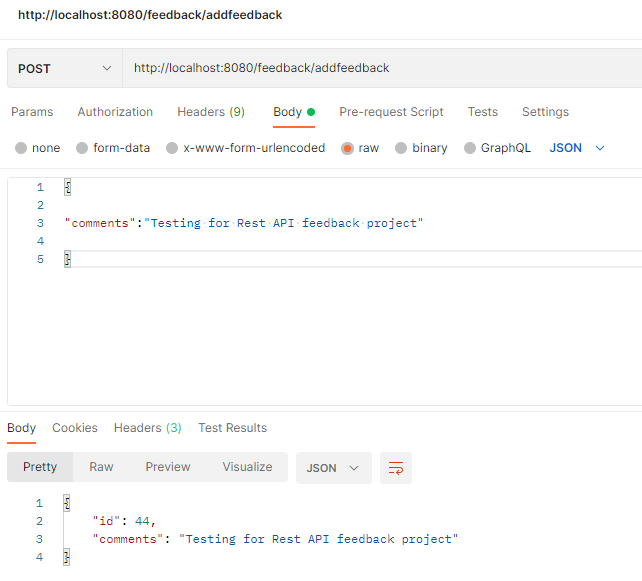


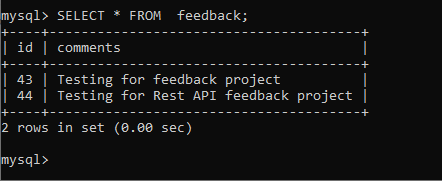
* Add admin in the system,

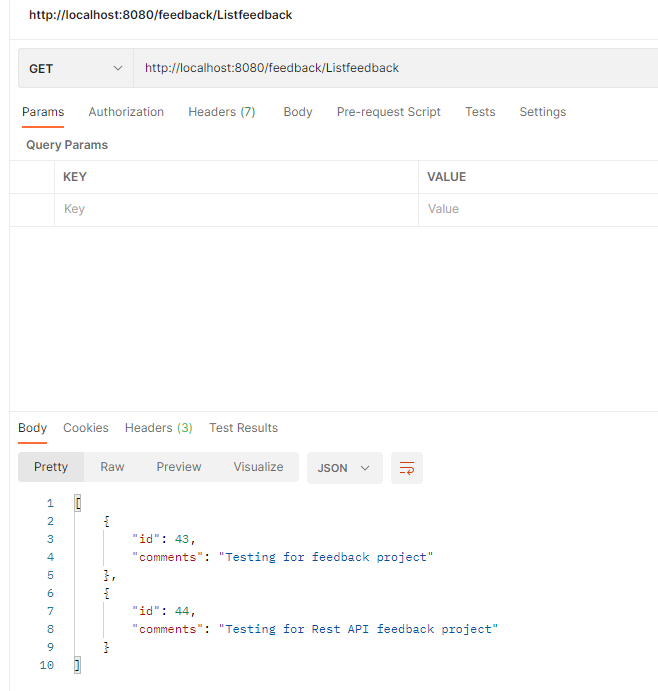
{

"comments":"Testing for Rest API feedback project"

}







## **Pushing the code to GitHub repository**

* Open your command prompt and navigate to the folder where you have created your files.

**cd <folder path>**

* Initialize repository using the following command:

**git init**

* Add all the files to your git repository using the following command:

**git add .**

* Commit the changes using the following command:

**git commit -m <commit message>**

* Push the files to the folder you initially created using the following command:

**git push -u origin main**

**Source code for the project available under below repository,**

<https://github.com/S-KAVITHA/Phase3-Practice-Projects>