**DOCUMENTATION**

**Problem Statement:**

To 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.

**Working:**

As a part of developing an ecommerce web application, a REST resource is used to capture the user feedback. The data is sent to the REST API which collects feedback from various sources. An entity class Feedback is made with annotations to link it with the feedback table. A repository class maps the entity class to the CrudRepository interface. A REST controller class is created to create the REST endpoint. It takes in parameters using the POST protocol. Data received in the REST controller will be then saved into the database.

**Tools and Technologies used:**

1. Apache Tomcat as the web server
2. Spring Boot Application with Hibernate (that internally uses jdbc)
3. H2 Database () which is an embedded SQL database.
4. Spring Tool Suite 4: An IDE to code the application
5. Java: A programming language to develop the prototype
6. Git: To connect and push files from the local system to GitHub
7. GitHub: To store the application code and track its versions
8. Maven: To manage the project dependencies

**Steps to achieve the required task:**

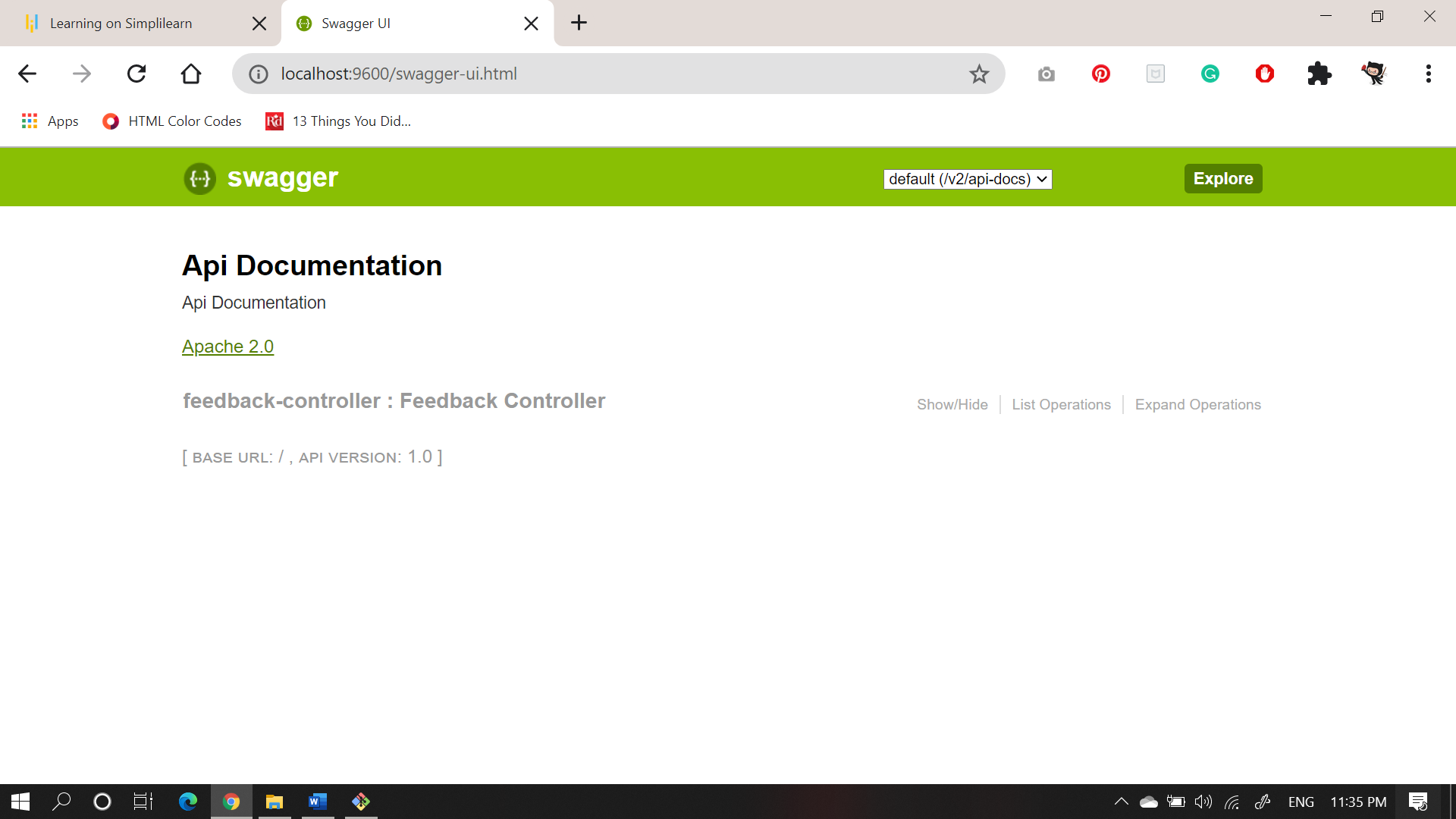
1. The user can interact with the application via a command line, without needing any other application except (Java to be installed in the system) to support it. For windows terminal, type the command:

**java -jar Spring\_Boot\_User\_Feedback-0.0.1-SNAPSHOT.jar**

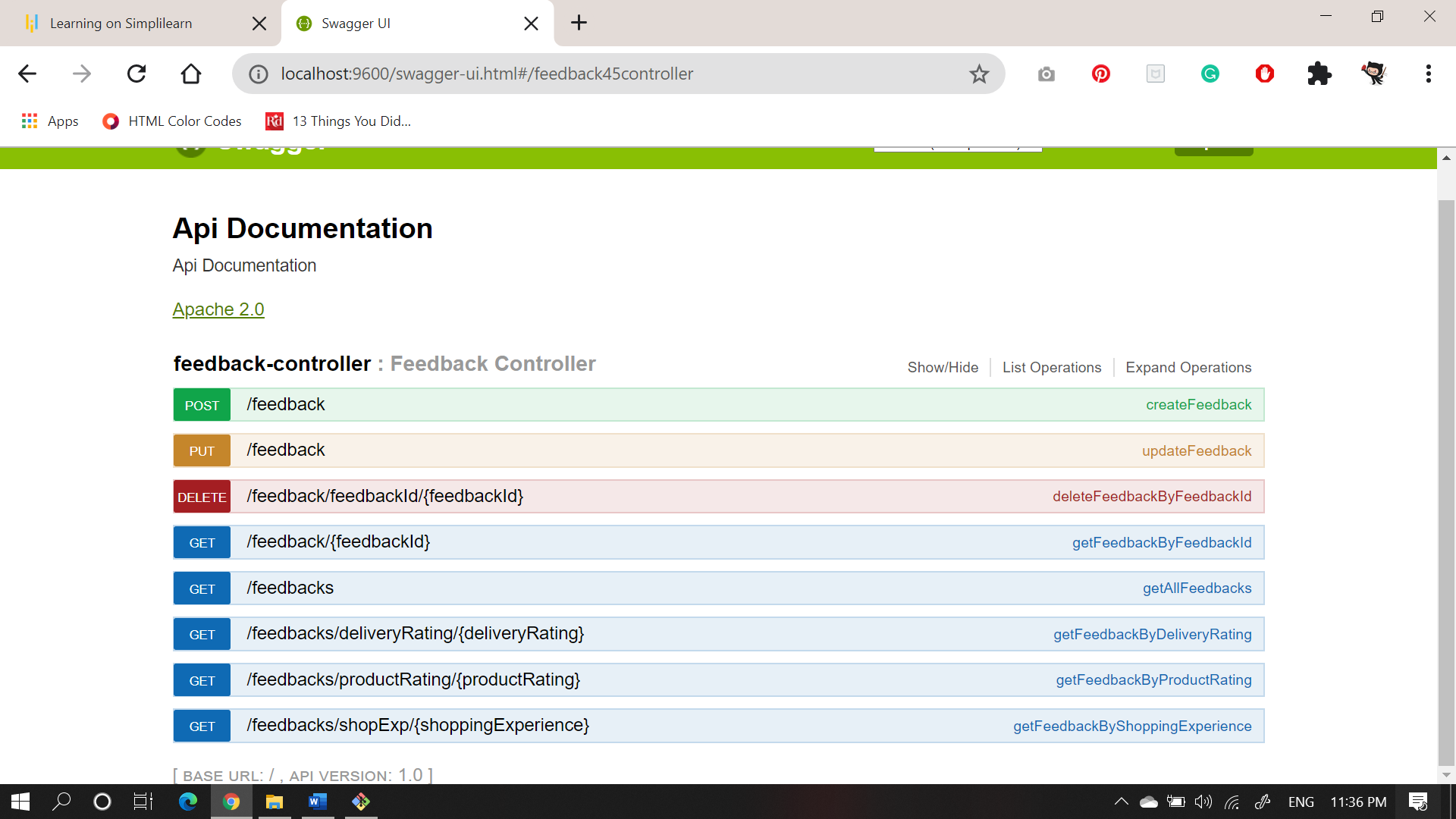
1. The Spring Boot Application starts successfully. Now For Documentation and testing the application, visit the url below:

[**http://localhost:9600/swagger-ui.html**](http://localhost:9600/swagger-ui.html)

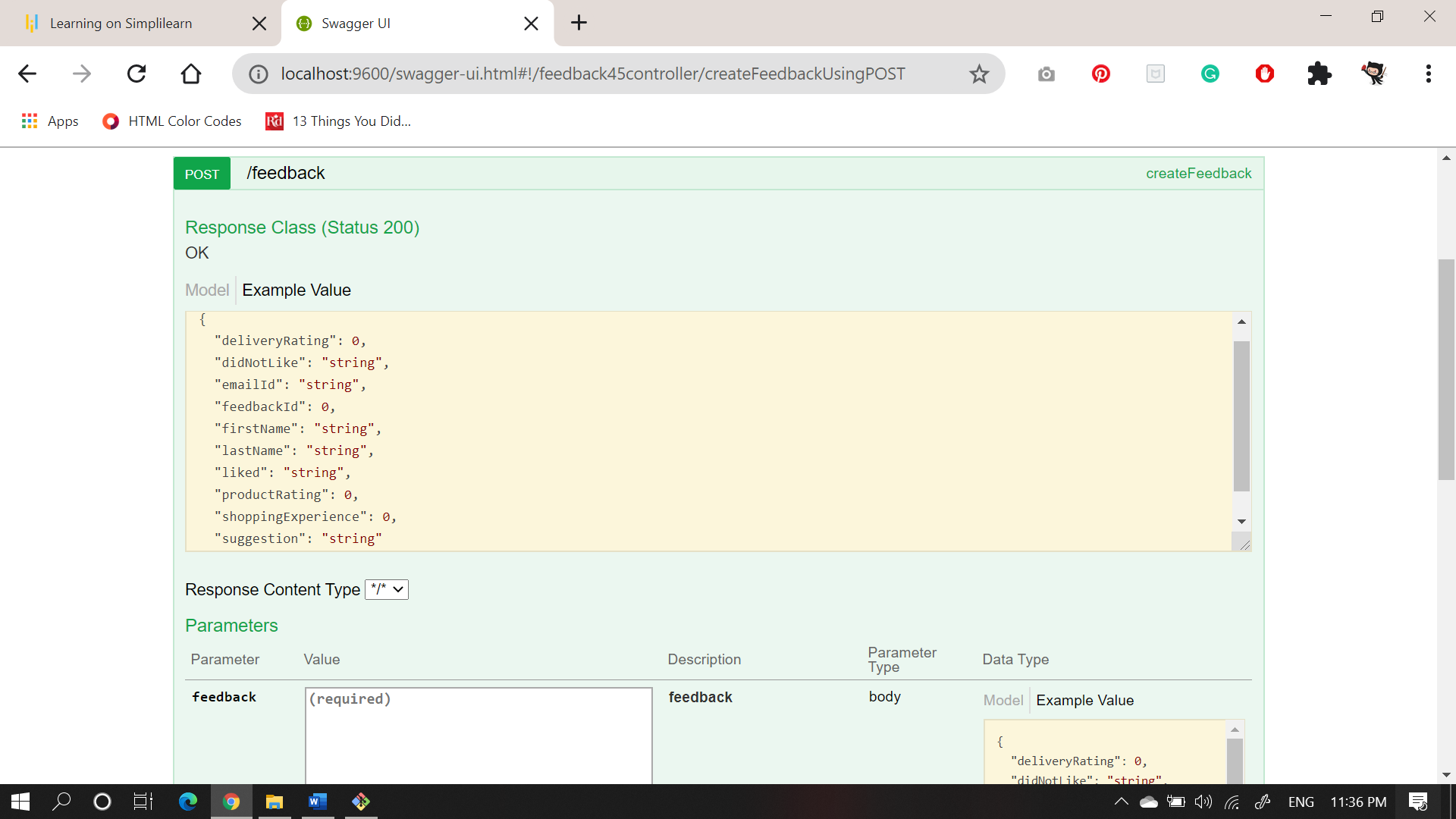
1. Swagger UI allows us to visualize and interact with the API’s resources without having any of the implementation logic in place. It’s automatically generated from our OpenAPI (formerly known as Swagger) Specification, with the visual documentation making it easy for back end implementation and client side consumption.



**Click on show/hide to get the Endpoints of the Application**



**Click on each method to view the documentation. Click on the block below Model: Example Value to get the format of data to be entered in “required” section. After filling the values click on “Try it out” to see the result.**



**Links to access the project:**

1. **For Project jar file**

[**https://github.com/anmolpanjwani/TrainingPhase2andPhase3Project/tree/master/Spring\_Boot\_User\_Feedback-0.0.1-SNAPSHOT.jar**](https://github.com/anmolpanjwani/TrainingPhase2andPhase3Project/tree/master/Spring_Boot_User_Feedback-0.0.1-SNAPSHOT.jar)

1. **For Source Code**

[**https://github.com/anmolpanjwani/TrainingPhase2andPhase3Project/tree/master/Spring\_Boot\_User\_Feedback**](https://github.com/anmolpanjwani/TrainingPhase2andPhase3Project/tree/master/Spring_Boot_User_Feedback)

1. **For Images**

[**https://github.com/anmolpanjwani/TrainingPhase2andPhase3Project/tree/master/Images**](https://github.com/anmolpanjwani/TrainingPhase2andPhase3Project/tree/master/Images)