**Make an E-commerce Website for Sporty Shoes**

**Projection Description:**

As a Full Stack Developer, complete the features of the application by planning the development and pushing the source code to the GitHub repository. 

**Background of the problem statement:**

Sporty Shoes is a company that manufactures and sells sports shoes. They have a walk-in store, and now, they wish to launch their ecommerce portal sportyshoes.com.

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

You’re asked to develop a prototype of the application. It will be then presented to the relevant stakeholders for budget approval. Your manager has set up a meeting where you’re asked to do the following:

* Presenting the specification document which has the product’s capabilities, appearance, and user interactions.
* Setting up Git and GitHub account to store and track your enhancements of the prototype.
* Explaining the Java concepts used in the project.
* Discussing the generic features of the product.
* There will be an admin to manage the website. An administrator login will be required to access the admin page.

**The admin should be able to change his password if he wants, he should be able to:**

* Manage the products in the store including categorizing them.
* Browse the list of users who have signed up and be able to search user.
* See purchase reports filtered by date and category

**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 purchase shoes.
* As a user, I want to login using login page and enter the welcome page.
* As a user, I want to search for available shoes based on category, shoe type, color, date, season and price.
* As a user, I want to select a shoe and purchase it.
* As an admin, I want to login and add products.
* As an admin, I want to maintain the products.
* As an admin, I want to list available products and users.
* As a developer, I want to build restful webservice application to add, delete, update and delete admin users.
* As a developer, I want to build restful webservice application to add, delete, update, list and delete users.
* As a developer, I want to build restful webservice application to add, delete, update, list and delete shoe category.
* As a developer, I want to build restful webservice application to add, delete, update, list and delete products.
* As a developer, I want to build restful webservice application to search products by creation date, category, color, season, shoe type and price.
* As a developer, I want to build restful webservice application to add, delete, update, list and delete purchased items.
* As a developer, I want to build restful webservice application to list purchased items by category and date.

**Sprint 1 (Week 1)**

* As a developer, I want to build restful webservice application to add, delete, update and delete admin users.
* As a developer, I want to build restful webservice application to add, delete, update, list and delete users.
* As a developer, I want to build restful webservice application to add, delete, update, list and delete shoe category.
* As a developer, I want to build restful webservice application to add, delete, update, list and delete products.

**Sprint 2 (Week 2)**

* As a developer, I want to build restful webservice application to search products by creation date, category, color, season, shoe type and price.
* As a developer, I want to build restful webservice application to add, delete, update, list and delete purchased items.
* 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.

**Architecture diagram / flow chart:**

**User Operations,**

**Start**

**User page with login, create user and change password options are performed through rest API.**

**If user account is created, user logs into his account successfully and performs operations.**

**List of available sporty shoes are listed for purchase. Selection can be done by category, shoe type, color, season, price and creation date.**

**User purchases the shoes which is done as part of rest API post man tool.**

**End**

**Admin Operations,**

**Start**

**Admin page with login, create user and change password options are performed through rest API.**

**If admin account is created, admin logs into his account successfully and performs product maintenance operations.**

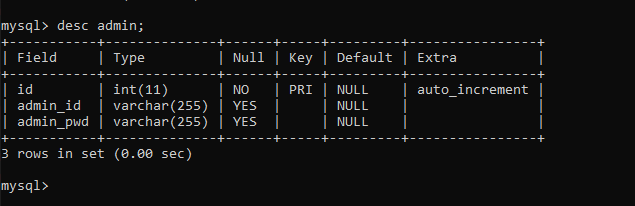
**Admin can add new products into system and maintain products through rest API postman tool.**

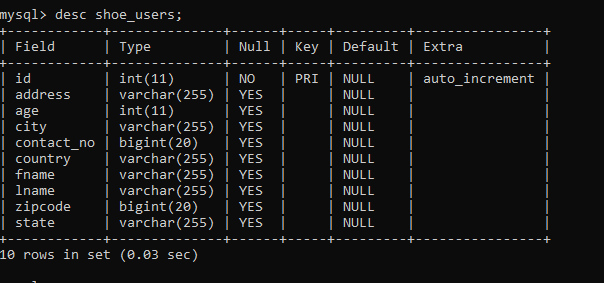
**List of users and available sporty shoes are listed for purchase. Selection can be done by category, shoe type, color, season, price and creation date.**

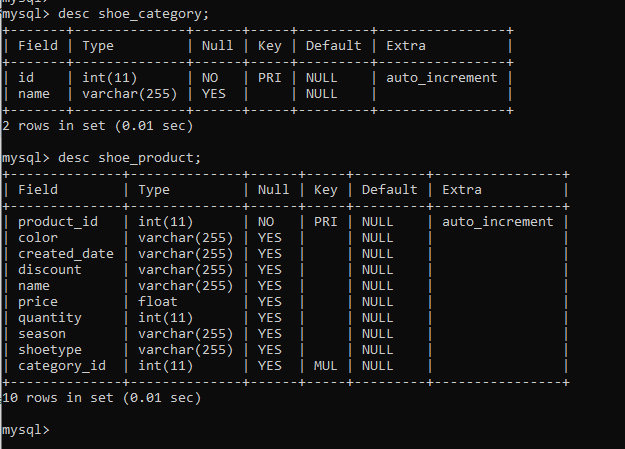
**End**

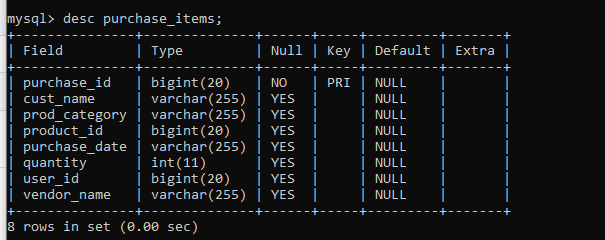
**Screenshots,**

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

****



****



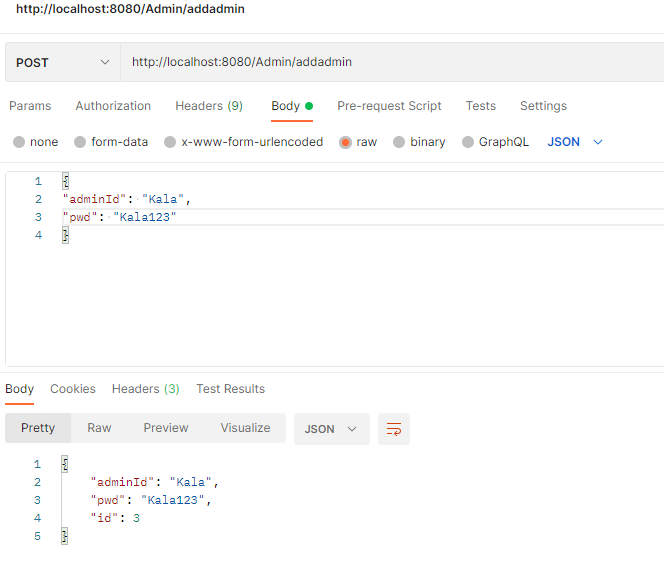
* Add admin in the system,

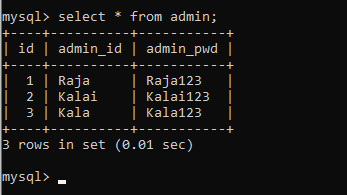
{

"adminId": "Kala",

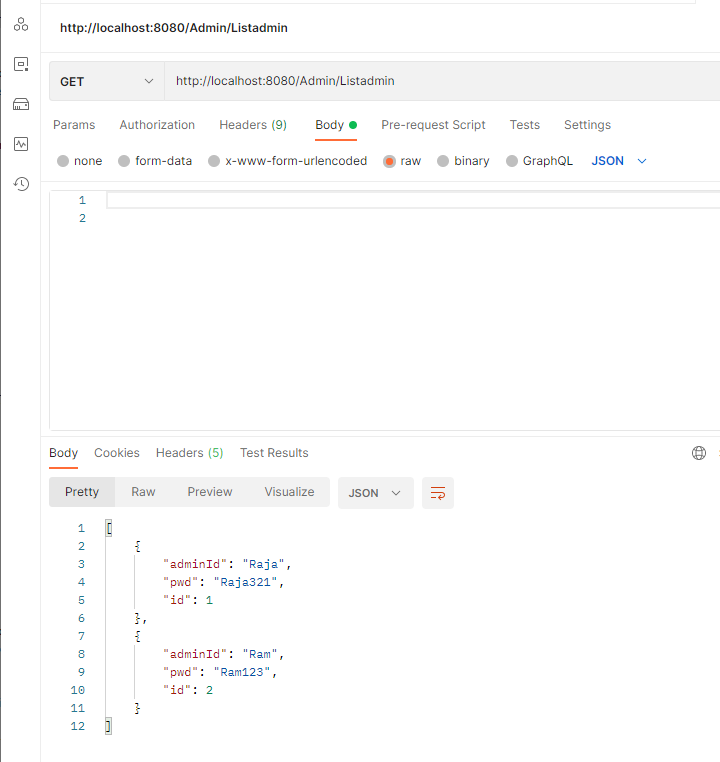
"pwd": "Kala123"

}

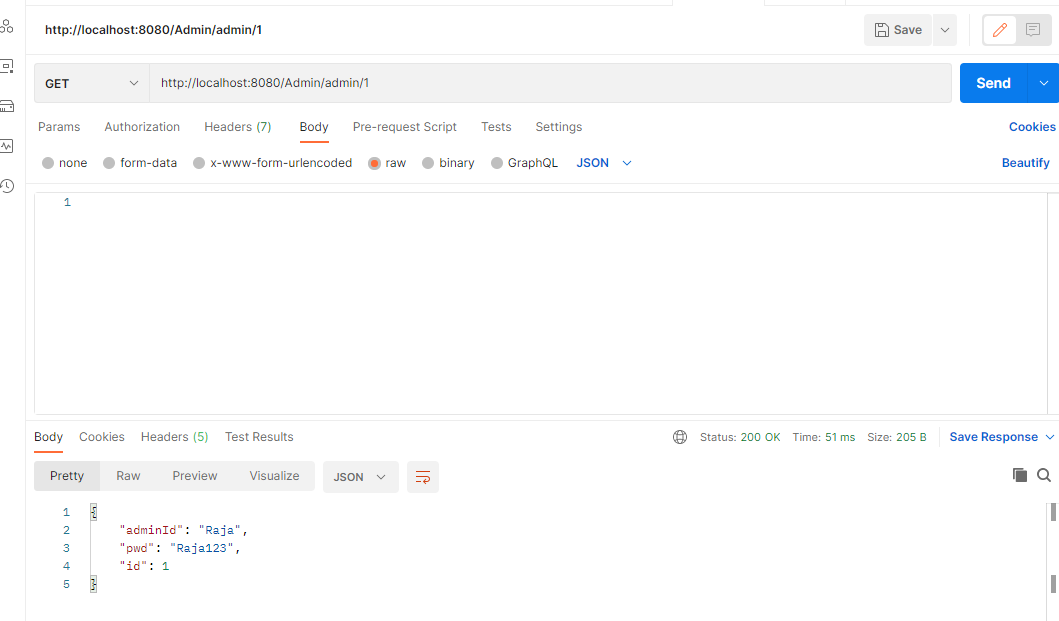




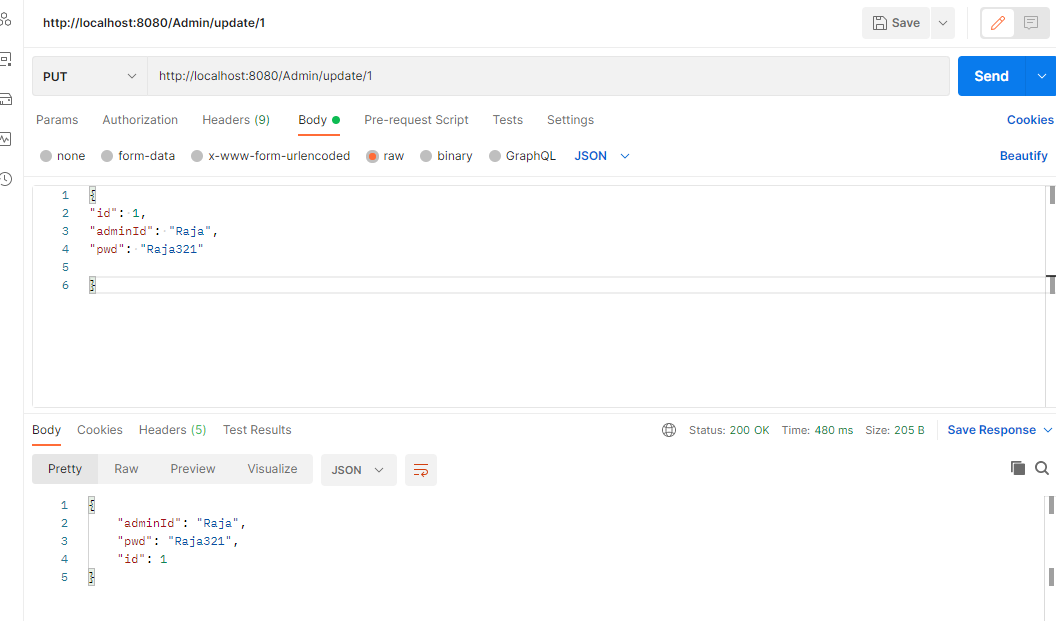
* List of admins registered in the system,

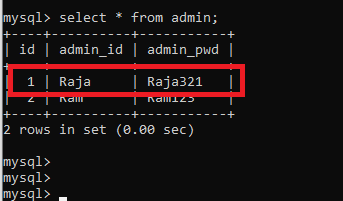


* Get specific admin in the system



* Update specific admin in the system



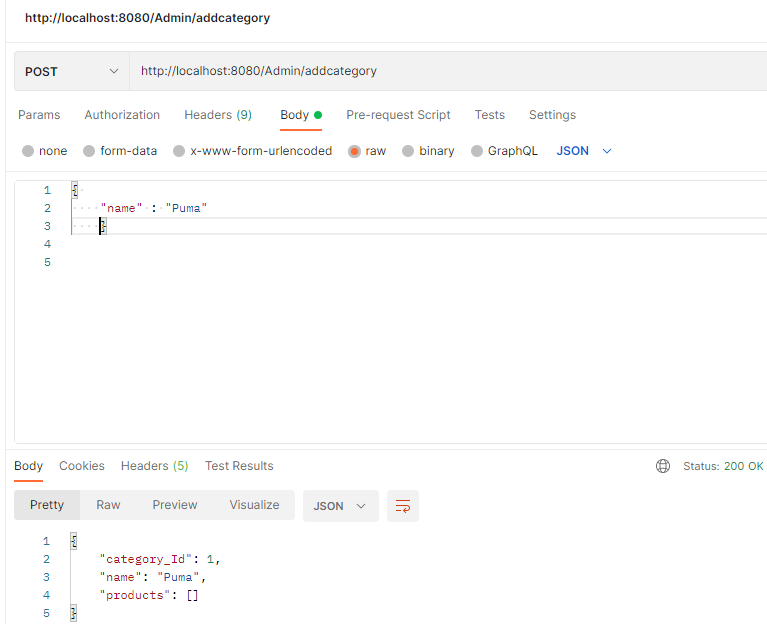


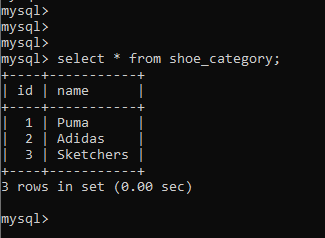
* Add category in the system

{

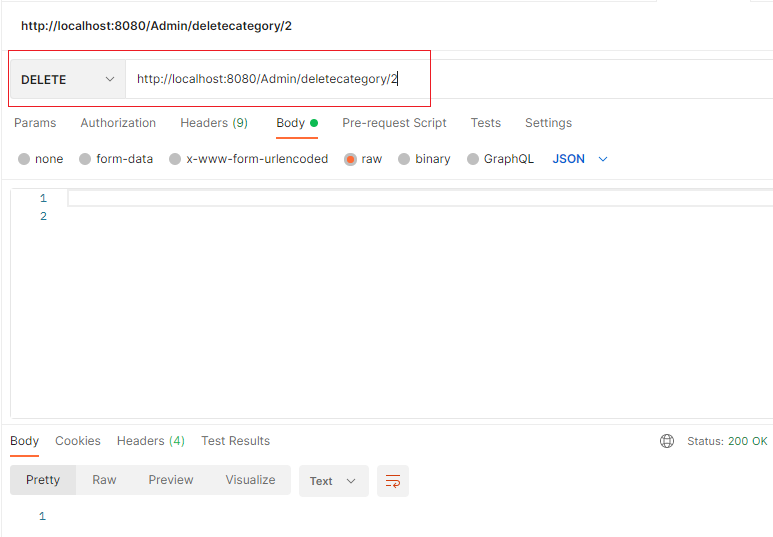
"name" : "Puma"

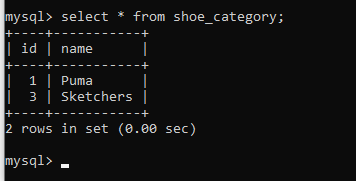
}



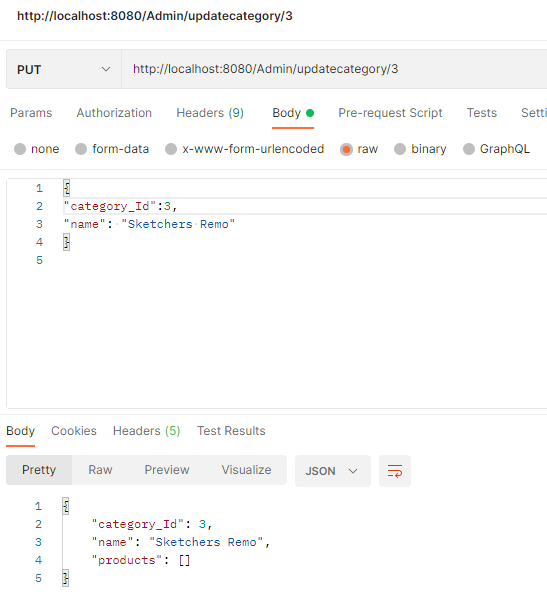


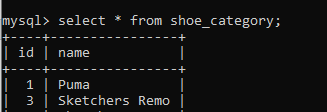
* Delete category in the system



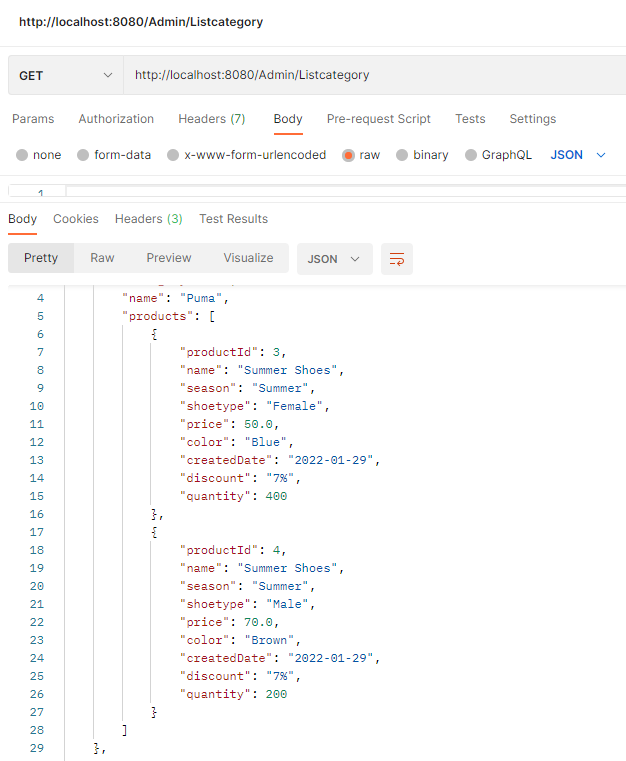


* Update category in the system



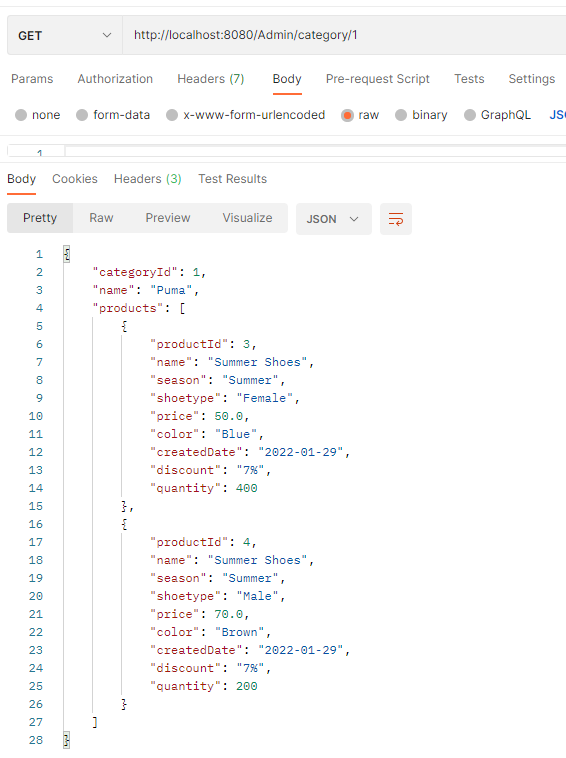


* List categories in the system





* Get specific category in the system



* Add product in the system

{

"category": {"category\_Id":2},

"color": "White",

"discount": "10%",

"name": "Rain Shoes",

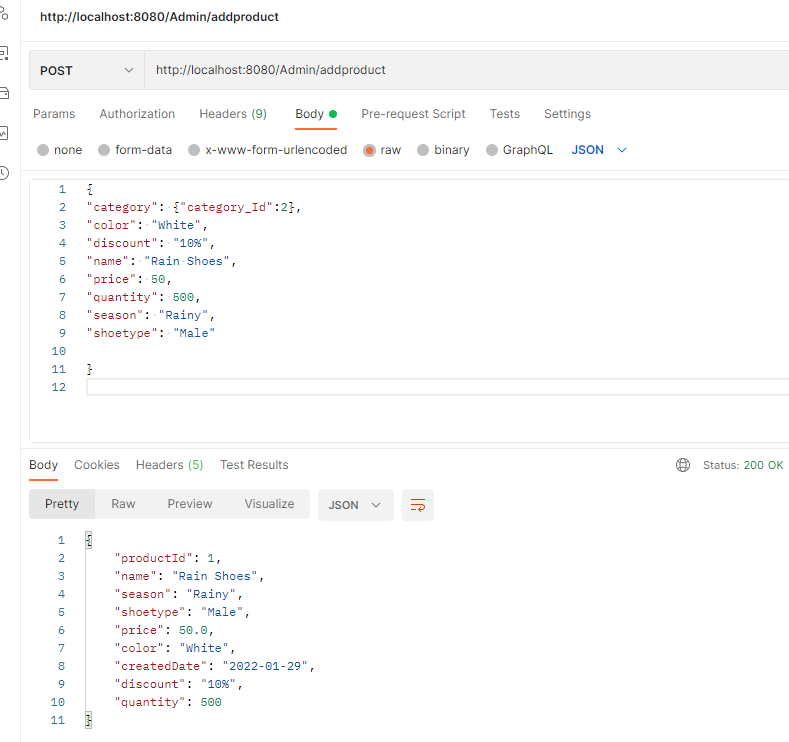
"price": 50,

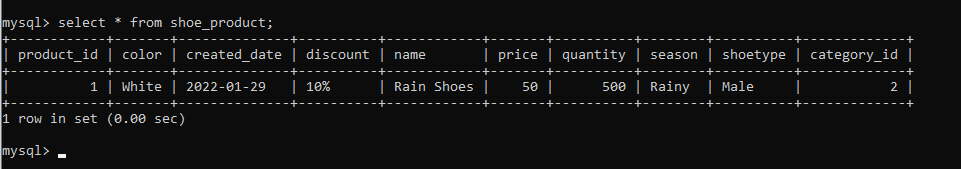
"quantity": 500,

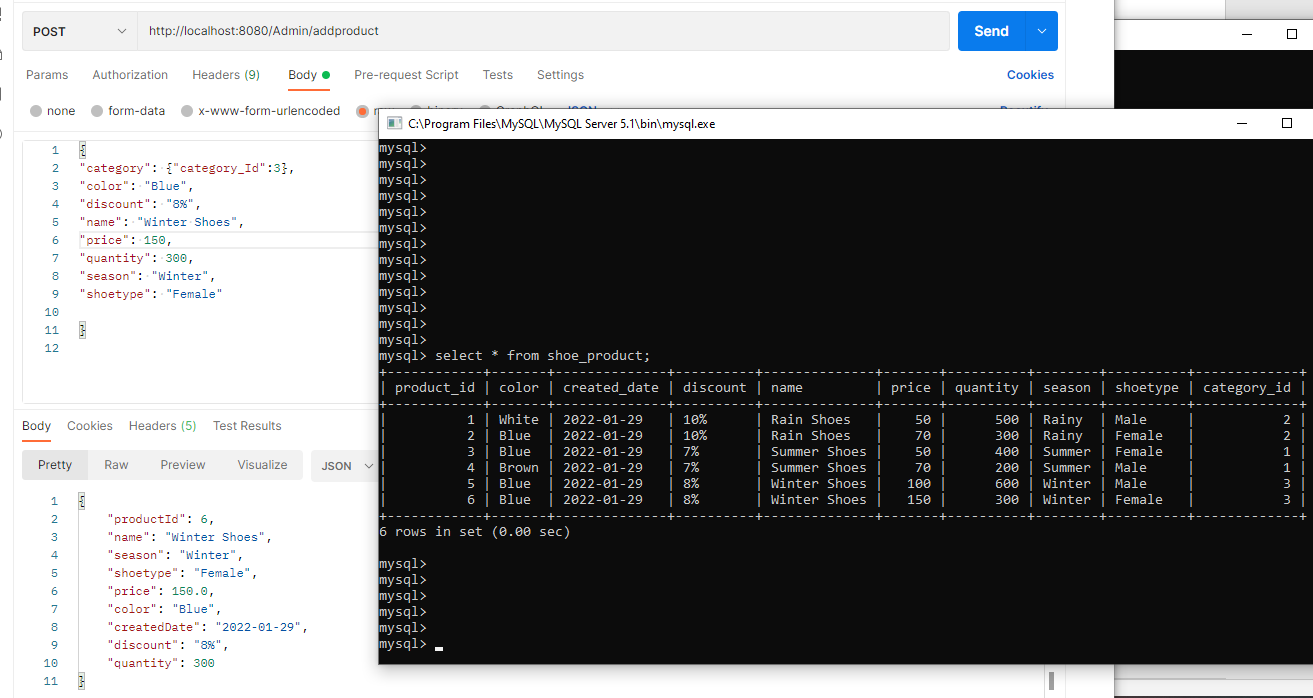
"season": "Rainy",

"shoetype": "Male"

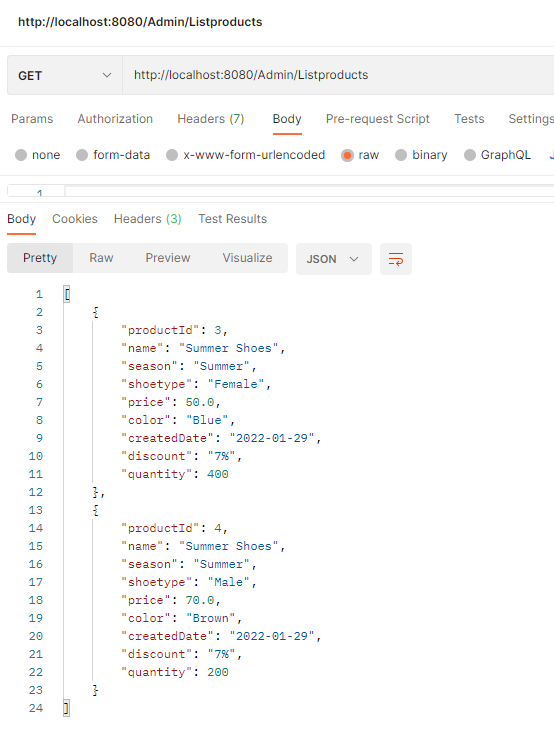
}



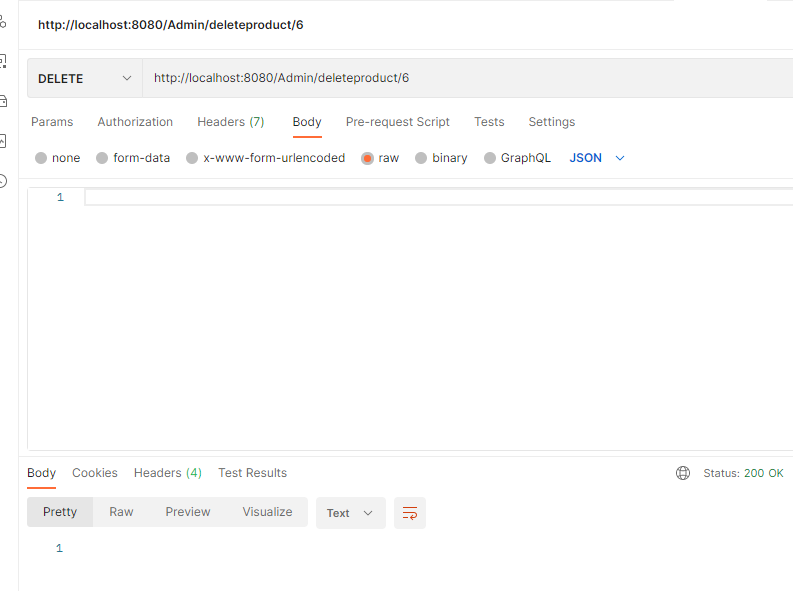


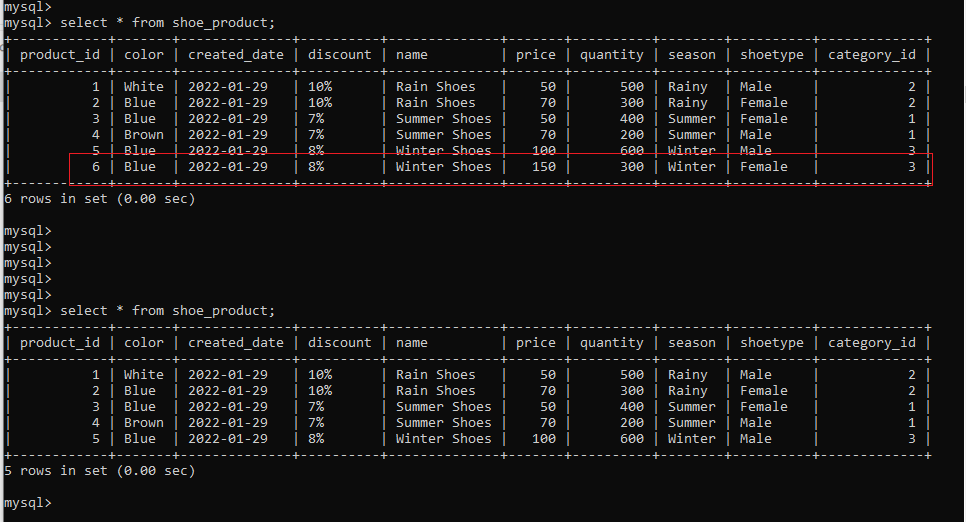


* List products in the system



* Delete product in the system





* Update product in the system

{

"category": {"category\_Id":2},

"color": "White",

"discount": "8%",

"name": "Rain Shoes",

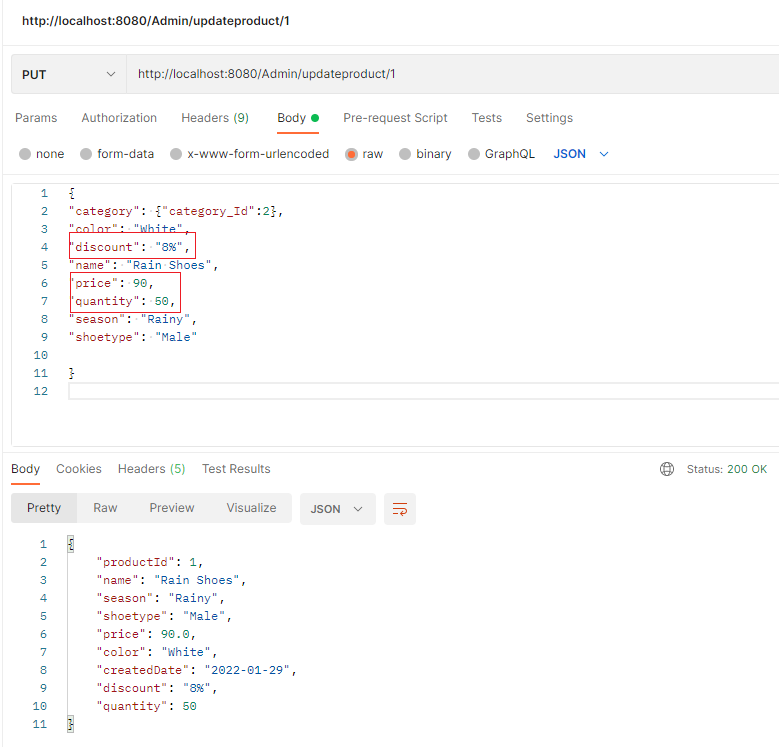
"price": 90,

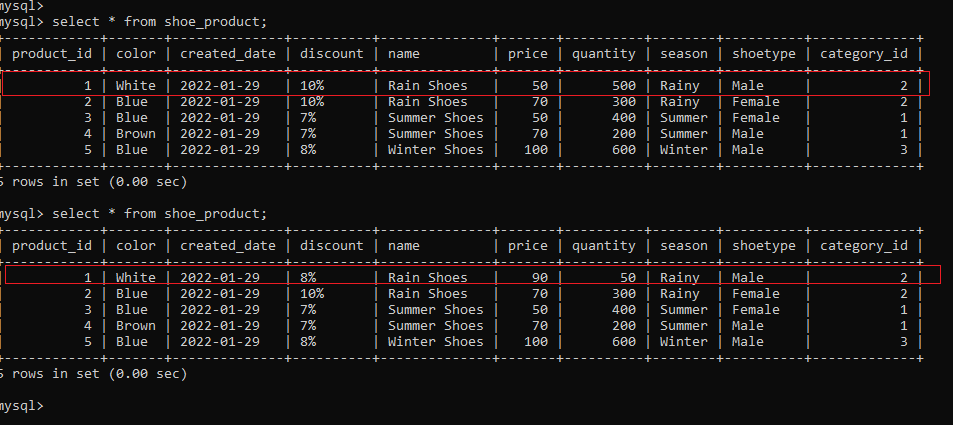
"quantity": 50,

"season": "Rainy",

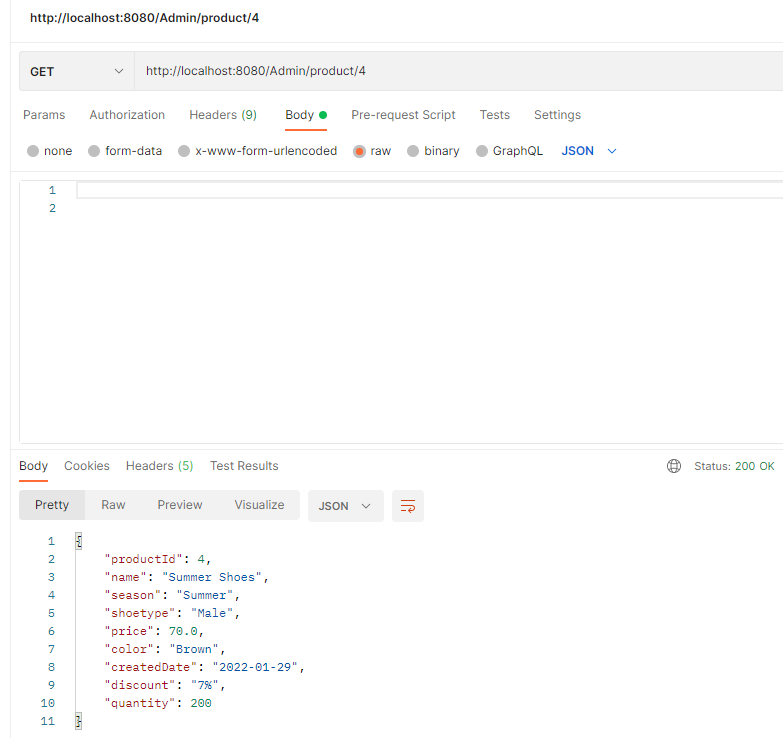
"shoetype": "Male"

}

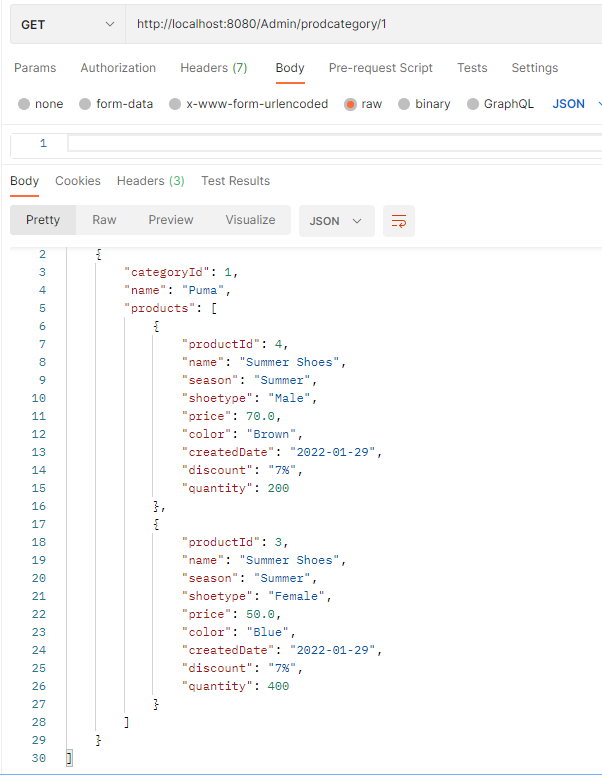




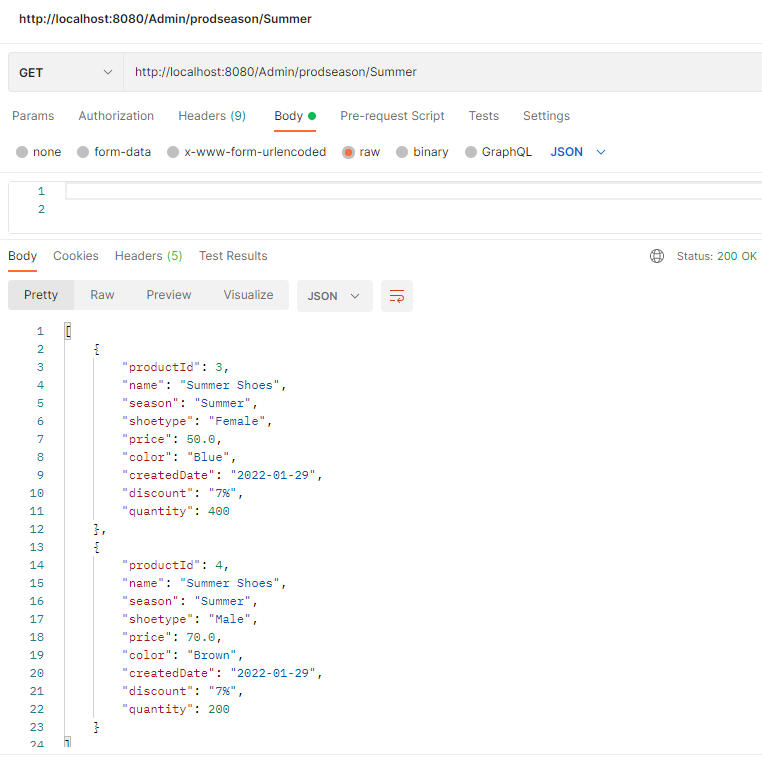
* Get specific product in the system



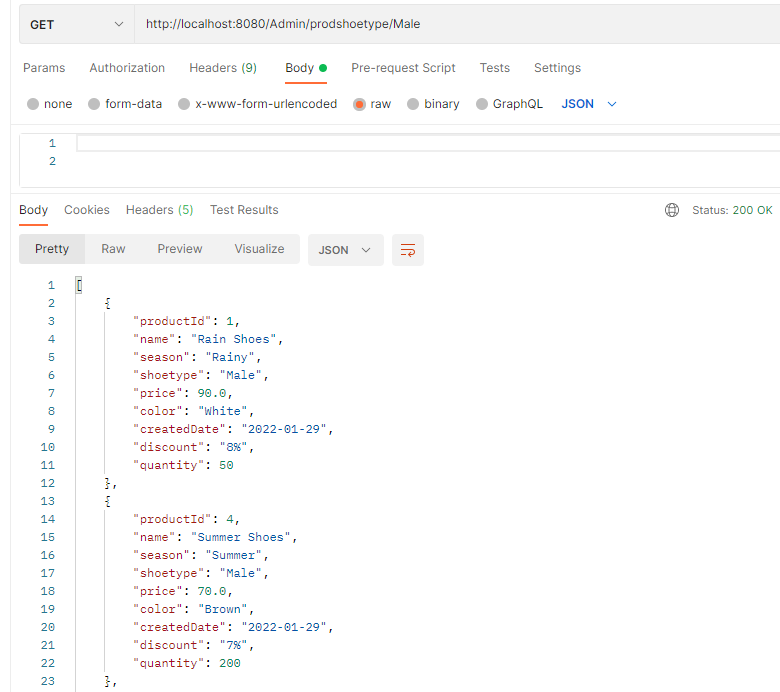
* Get product by category in the system



* Get product by season in the system

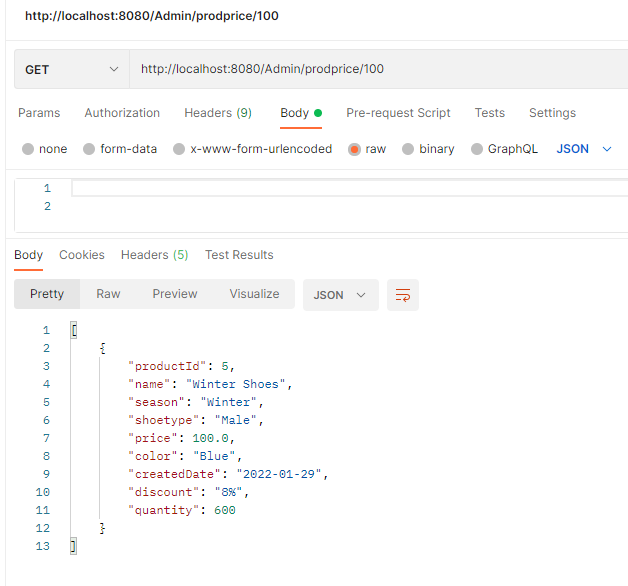


* Get product by shoe type in the system

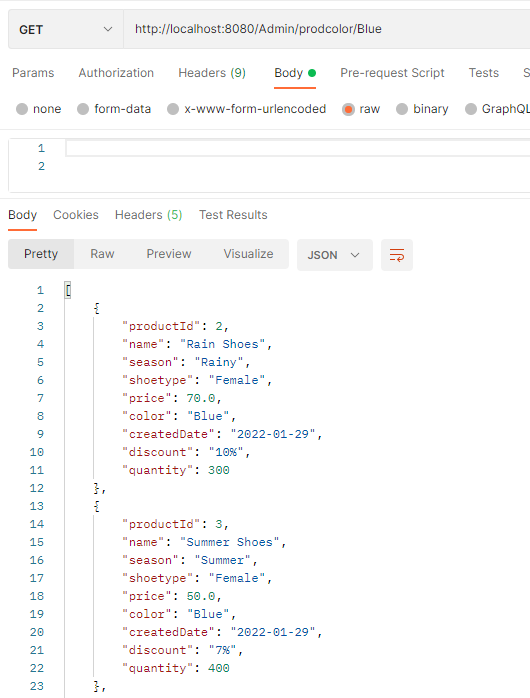




* Get product by price in the system

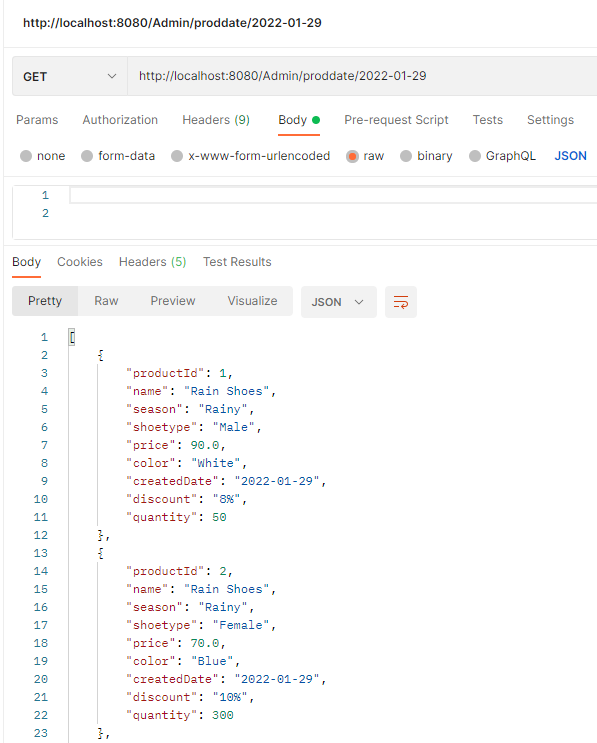


* Get product by color in the system





* Get product by creation date in the system







* Add users in the system

{

"fname": "Raja",

"lname": "Kumaran",

"address": "415 Hopestone Terrace",

"city": "Chantilly",

"country": "USA",

"state": "Virginia",

"zipcode": 62256,

"contact\_no": 3018953654,

"age": 50

}

{

    "fname": "Kalai",

    "lname": "Vani",

    "address": "125 Springhurst Terrace",

    "city": "Washington",

    "zipcode": 45879,

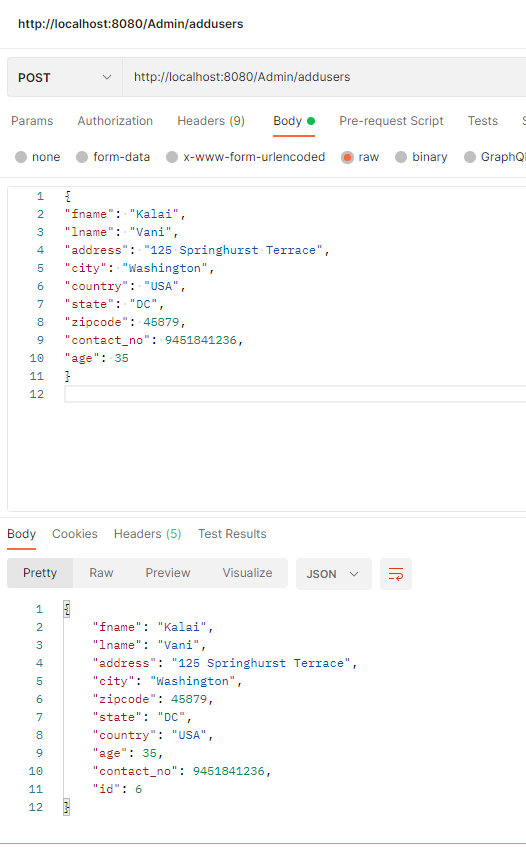
    "state": "DC",

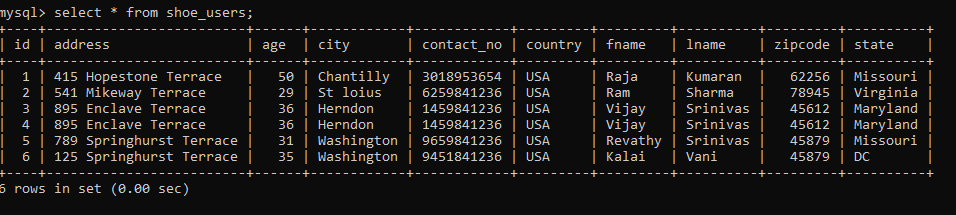
    "country": "USA",

    "age": 35,

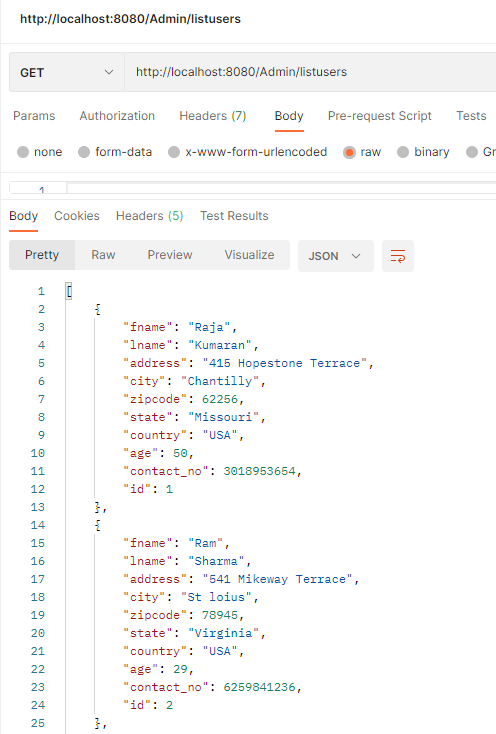
    "contact\_no": 9451841236,

 }





* List users in the system







* Update users in the system

{

"id":1,

"fname": "Raja",

"lname": "Kumaran",

"address": "415 Hopestone Terrace",

"city": "Chantilly",

"country": "USA",

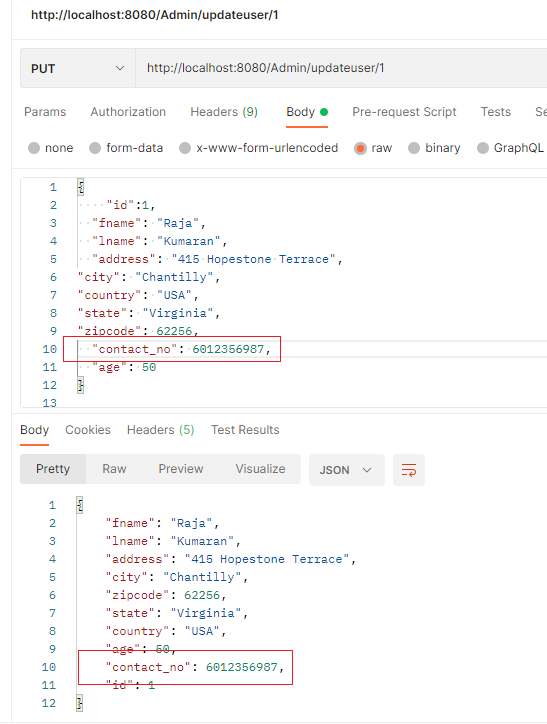
"state": "Virginia",

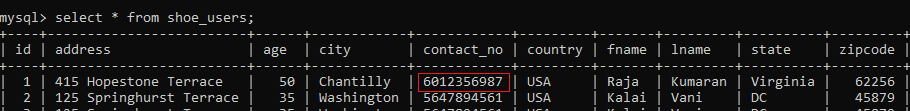
"zipcode": 62256,

"contact\_no": 6012356987,

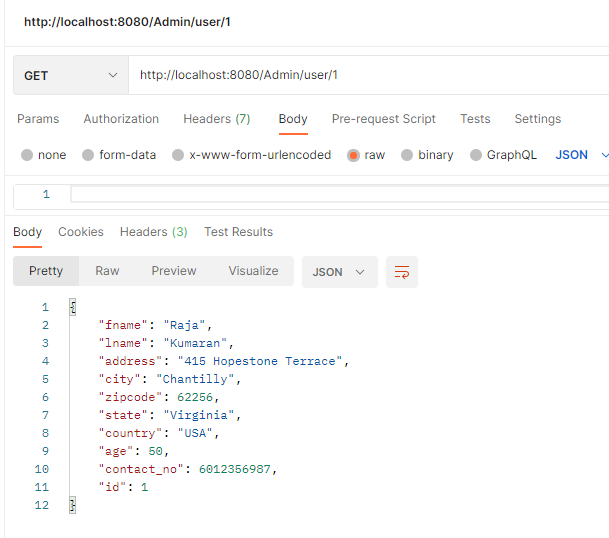
"age": 50

}

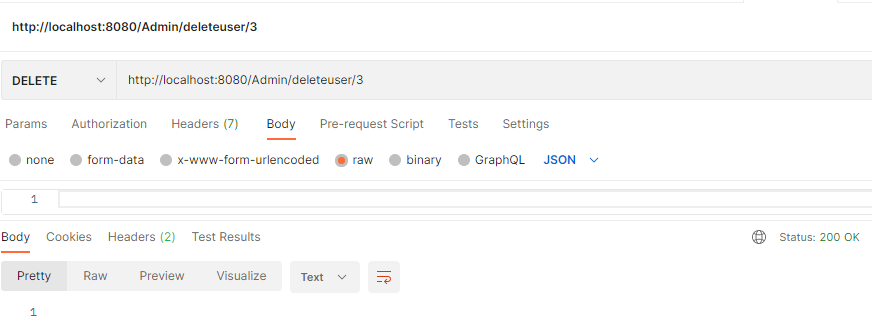


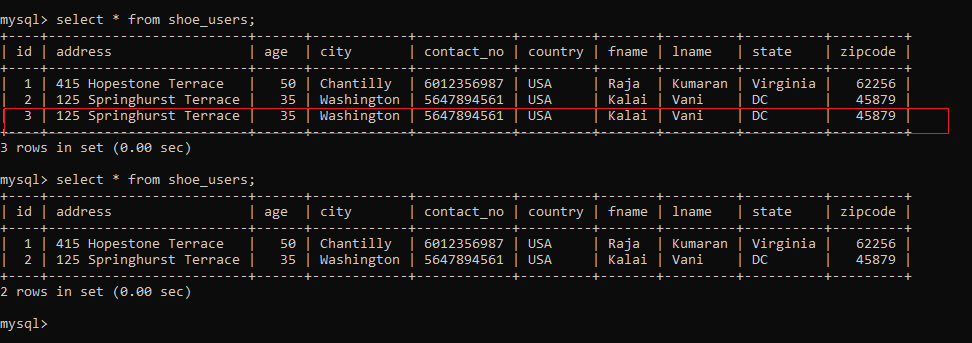


* Get specific user in the system



* Delete users in the system





* Add purchase items in the system

{

"custName":"Kalai Vani",

"prodCategory":1,

"productId":1,

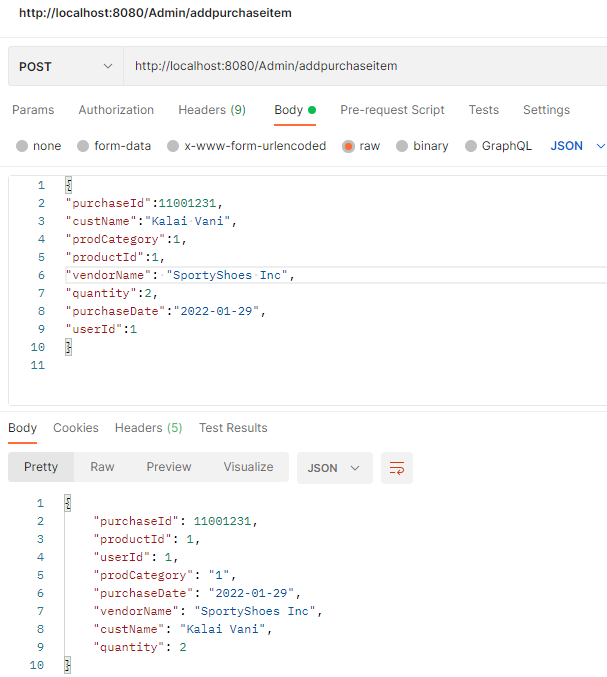
"quantity":75,

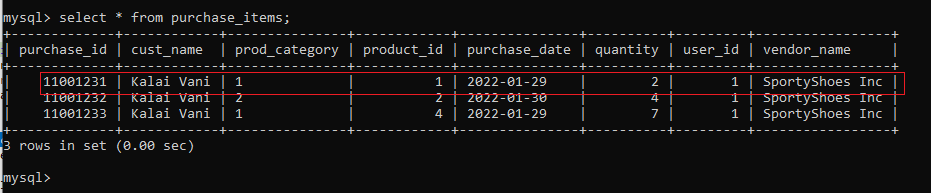
"purchaseDate":"2022-01-29",

"userId":1,

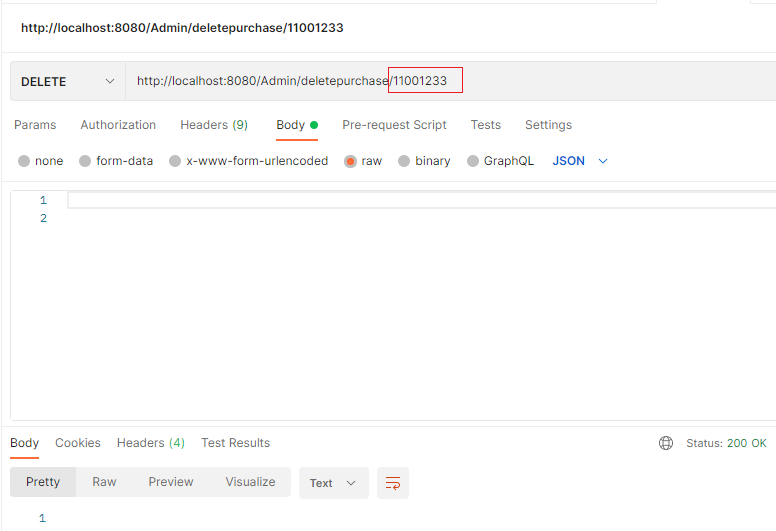
"vendorName":“SportyShoes Inc”

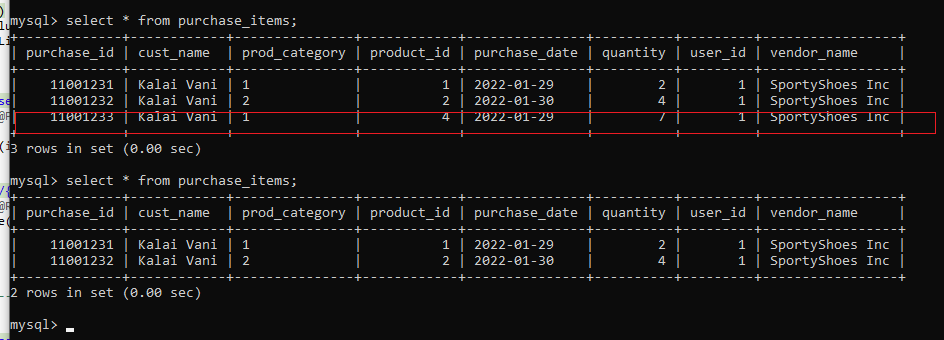
}



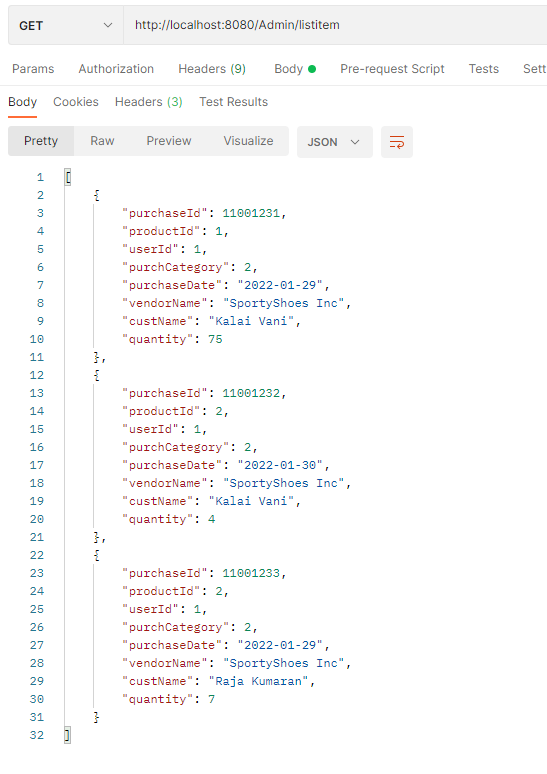


* Delete purchase items in the system

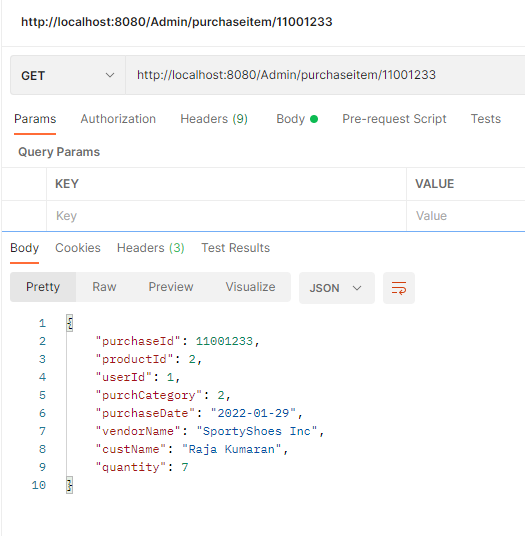




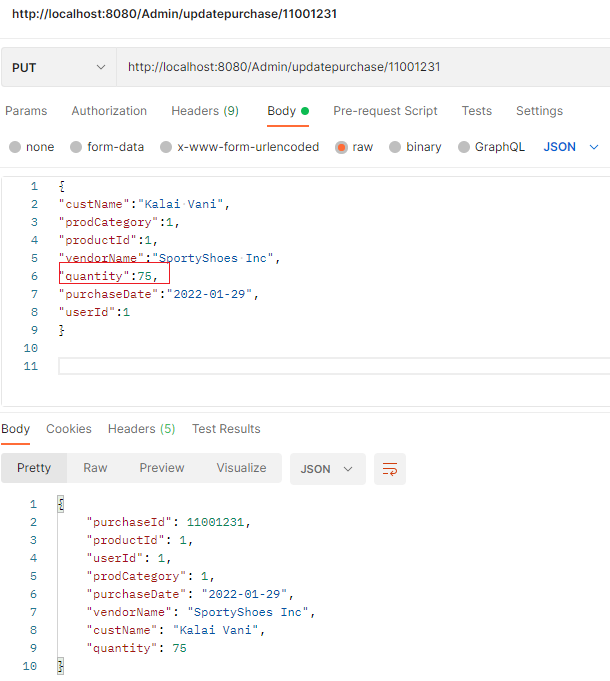
* List purchase items in the system

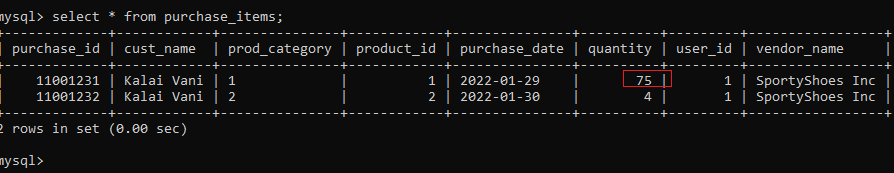


* Get specific purchase item in the system

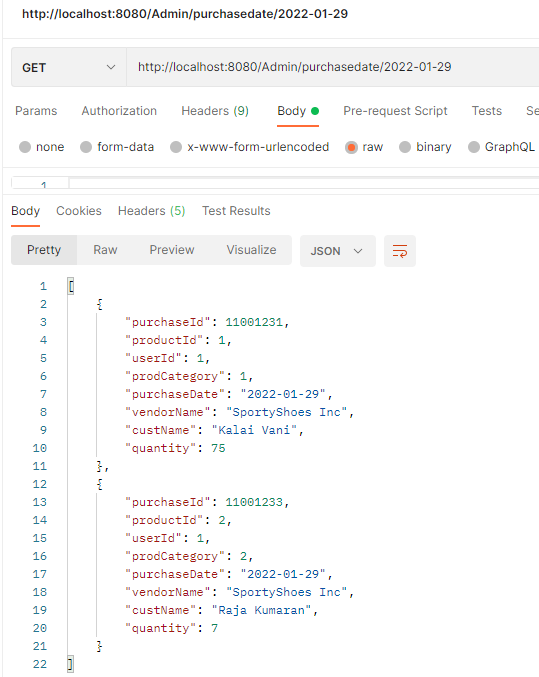


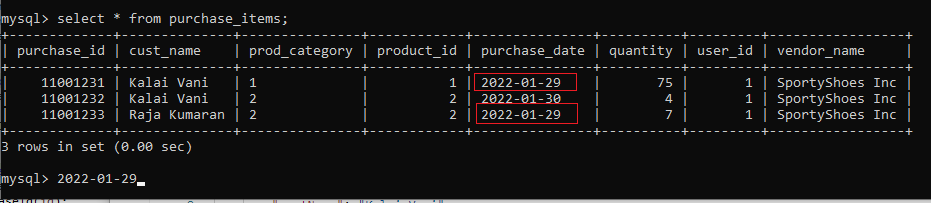
* Update purchase items in the system



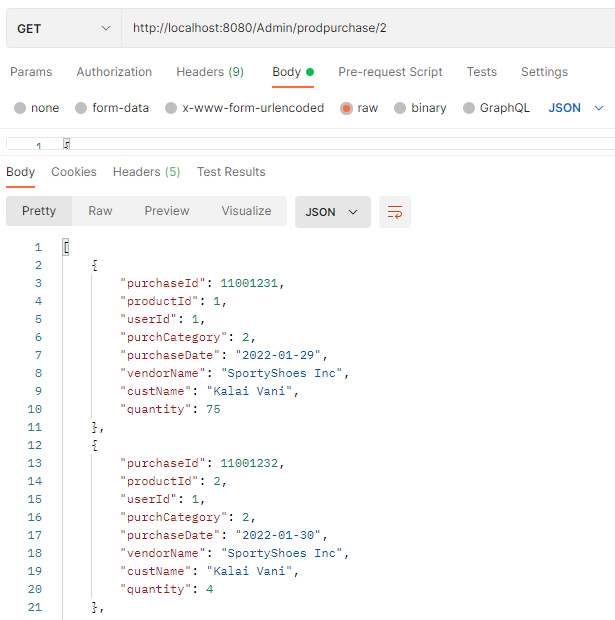


* Get purchase items by purchase date in the system



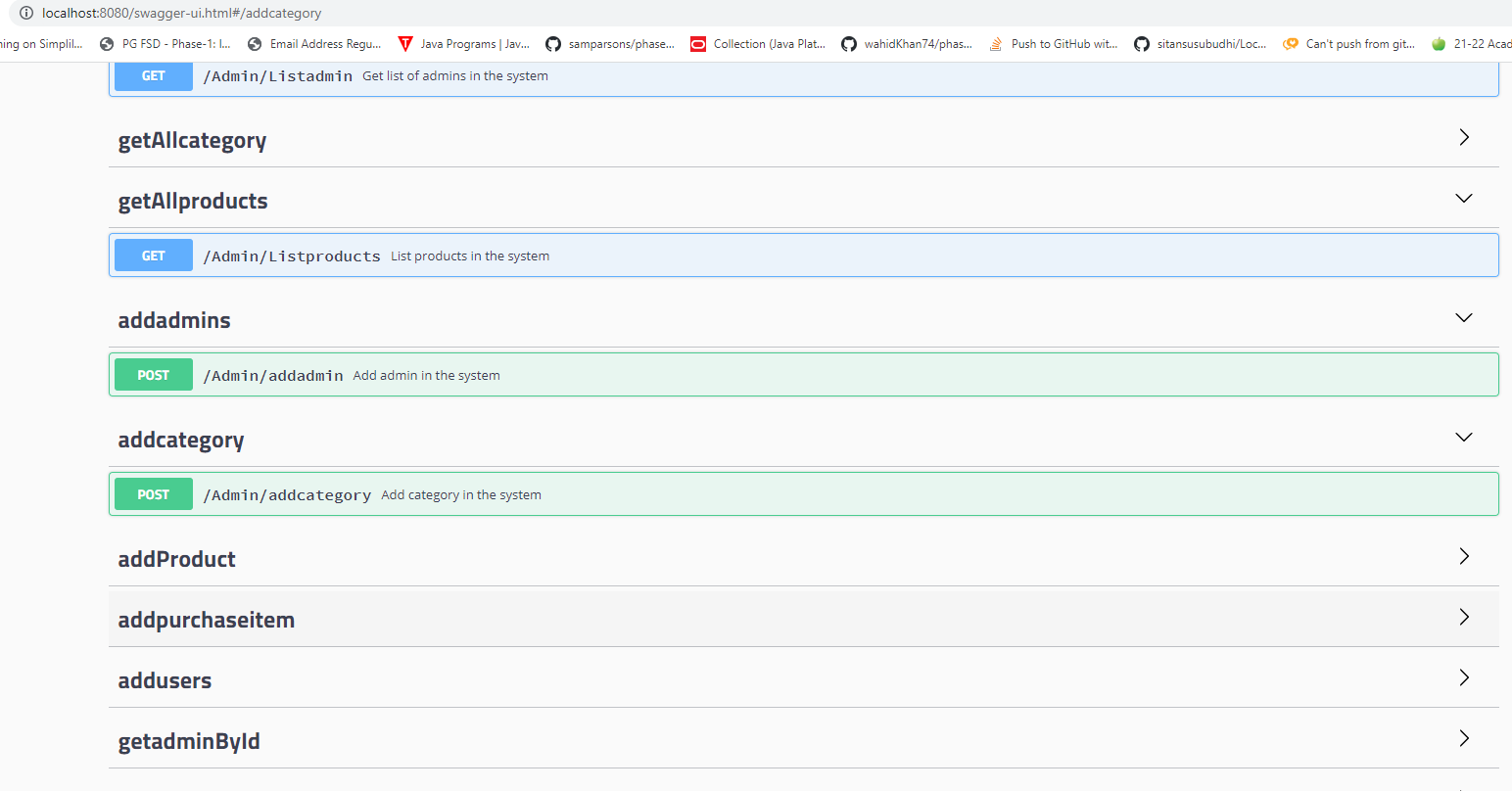


* Get purchase items by category in the system





Swagger tool,



## **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-SportyShoes-Project**](https://github.com/S-KAVITHA/Phase3-SportyShoes-Project)