**SportyShoes (an E-commerce Website).**

**This document contains sections for:**

* Project Description
* Core concepts used in project
* Flow of the Application.
* Project Users Stories : ( Agile and Scrum )
* Demonstrating the product capabilities, appearance, and user interaction.
* Conclusions

The code for this project is hosted at : <https://github.com/kanika880/sporty-shoes>

The project is developed by Kanika Gupta.

* 1. **Project Description:**

**Project objective:**

As a Full Stack Developer, complete the features of the application by planning the development of an e-commerce website sporty-shoes.

**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 e-commerce portal sportyshoes.com.

**In this application the users can:**

* Sign up by entering user id, name and password
* Sign in/ Sign out
* View the list of all products in the inventory.
* Search for a product by name.
* View the details of product by product id.
* View their own account details.
* Edit their account details.

For the above features to work, there will be an admin backend with the following features:

* An admin login page where the admin can change the password after login, if he wishes
* View the list of all signed up users.
* Search for users by their name.
* Add a new product.
* View the details of product by product id.
* Update the details of product.
* View the sorted list of all orders.

**The flow and features of the application:**

* Plan more than two sprints to complete the application
* List the core concepts and algorithms being used to complete this application
* Implement the appropriate concepts, such as exceptions, collections, and sorting techniques, spring boot, REST-API for source code optimization and increased performance

**You must use the following:**

* Eclipse/IntelliJ: An IDE to code for the application
* Java: A programming language to develop the web pages, databases, and others
* SQL: To create tables for admin, classes, students, and other specifics
* Git: To connect and push files from the local system to GitHub
* GitHub: To store the application code and track its versions
* Scrum: An efficient agile framework to deliver the product incrementally
* Search and Sort techniques: Data structures used for the project
* Specification document: Any open-source document or Google Docs

**The following requirements should be met:**

* The source code should be pushed to your GitHub repository. You need to document the steps and write the algorithms in it.
* The submission of your GitHub repository link is mandatory. In order to track your task, you need to share the link of the repository. You can add a section in your document.
* Document the process step-by-step starting from sprint planning to the product release.
* The application should not close, exit, or throw an exception if the user specifies an invalid input.
* You need to submit the final specification document which will include:
* Project and developer details
* Sprints planned and the tasks achieved in them
* Algorithms and flowcharts of the application
* Core concepts used in the project
* Links to the GitHub repository to verify the project completion

* 1. **Core concepts used in the project:**
* Used Java Language in Eclipse IDE.
* Spring-boot framework
* Collections framework
* Sorting
* Flow Control
* Recursion
* Exception Handling
* Streams API
* MYSQL
* JPA
  1. **Project Users Stories : ( Agile and Scrum )**

The project is planned to be completed in 3 sprints. Tasks assumed to be completed in the sprint are:

* Creating the flow of the application
* Initializing git repository to track changes as development progresses.
* Writing the Java program to fulfill the requirements of the project.
* Testing the Java program with different kinds of User input
* Pushing code to GitHub.

The goal of the company is to deliver a high-end quality product as early as possible.

**Sprint 1**

1) Adding or modifying product details by admin.

2) Observe the available products.

3) Observe orders.

4) Add/update product details.

**Sprint 2**

5) Options of choosing preferable product by user.

6) See available products in inventory.

**Sprint 3**

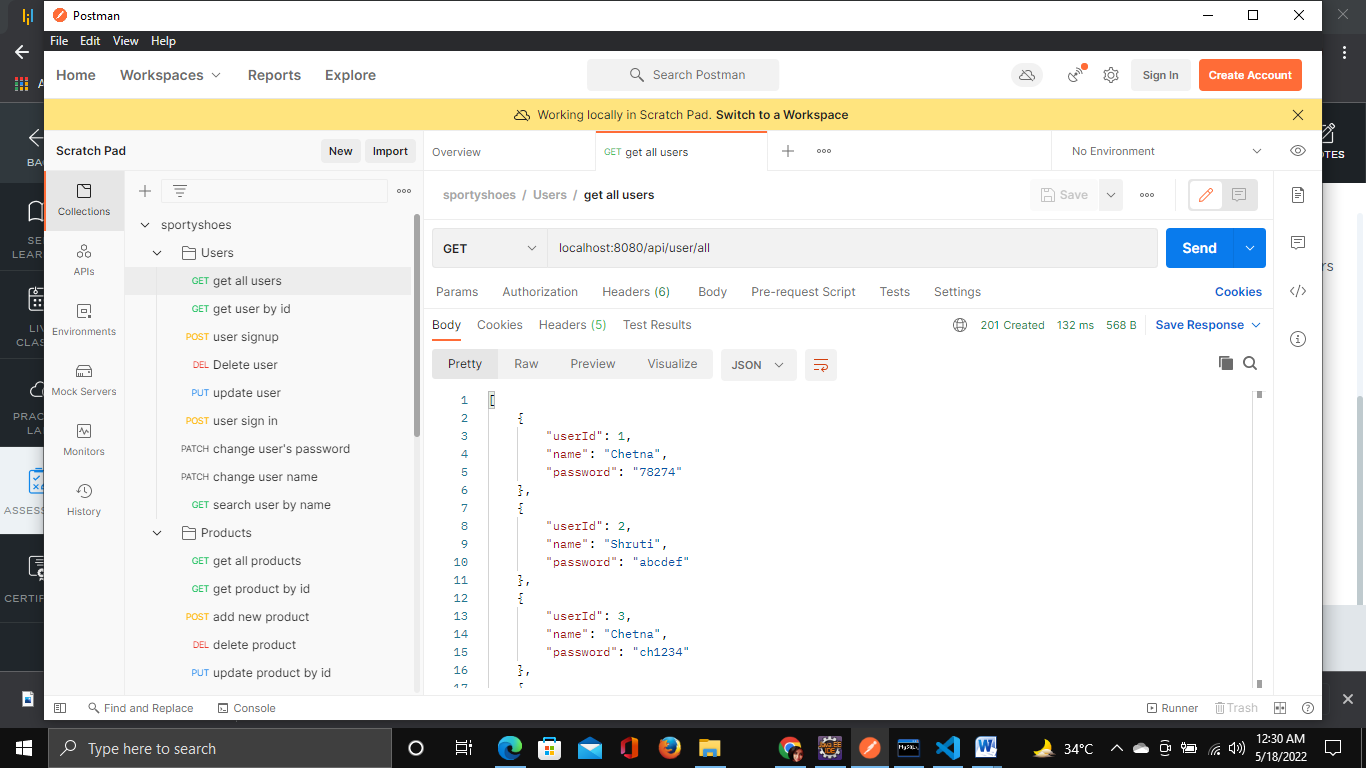
7) Storing user data

1. **Directory Structure / package:**

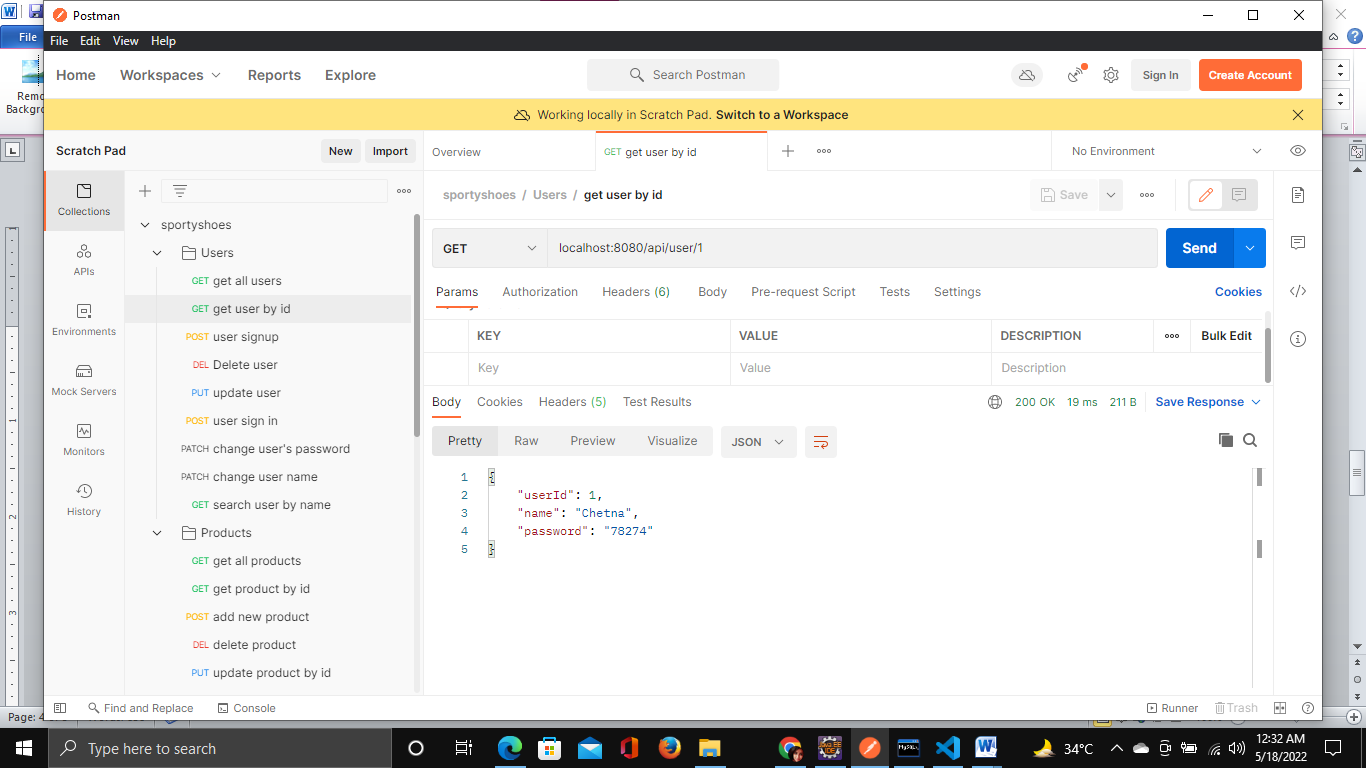
**Demonstrating the product capabilities, appearance, and user interactions**

To demonstrate the product capabilities, below are the sub-sections configured to highlight api requests handling by postman for the project:

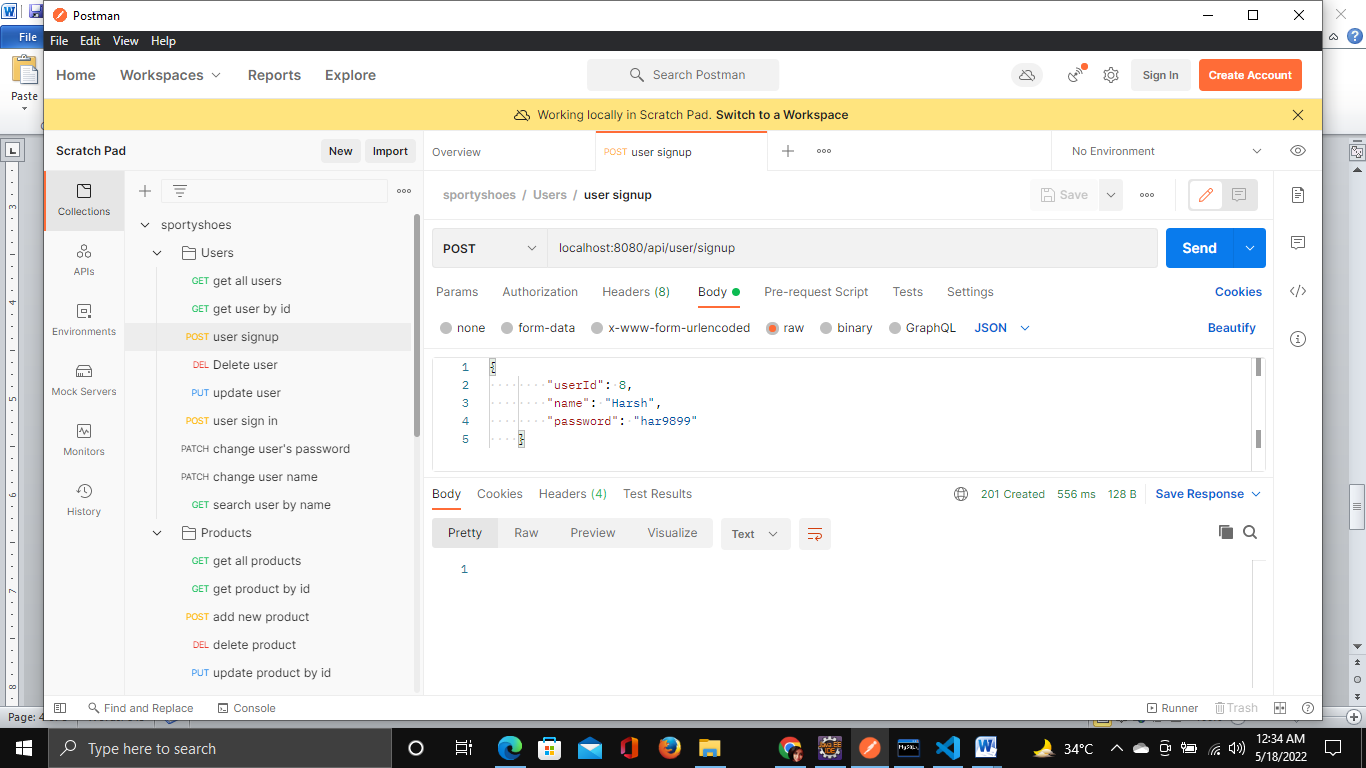
1. Getting all user details by admin



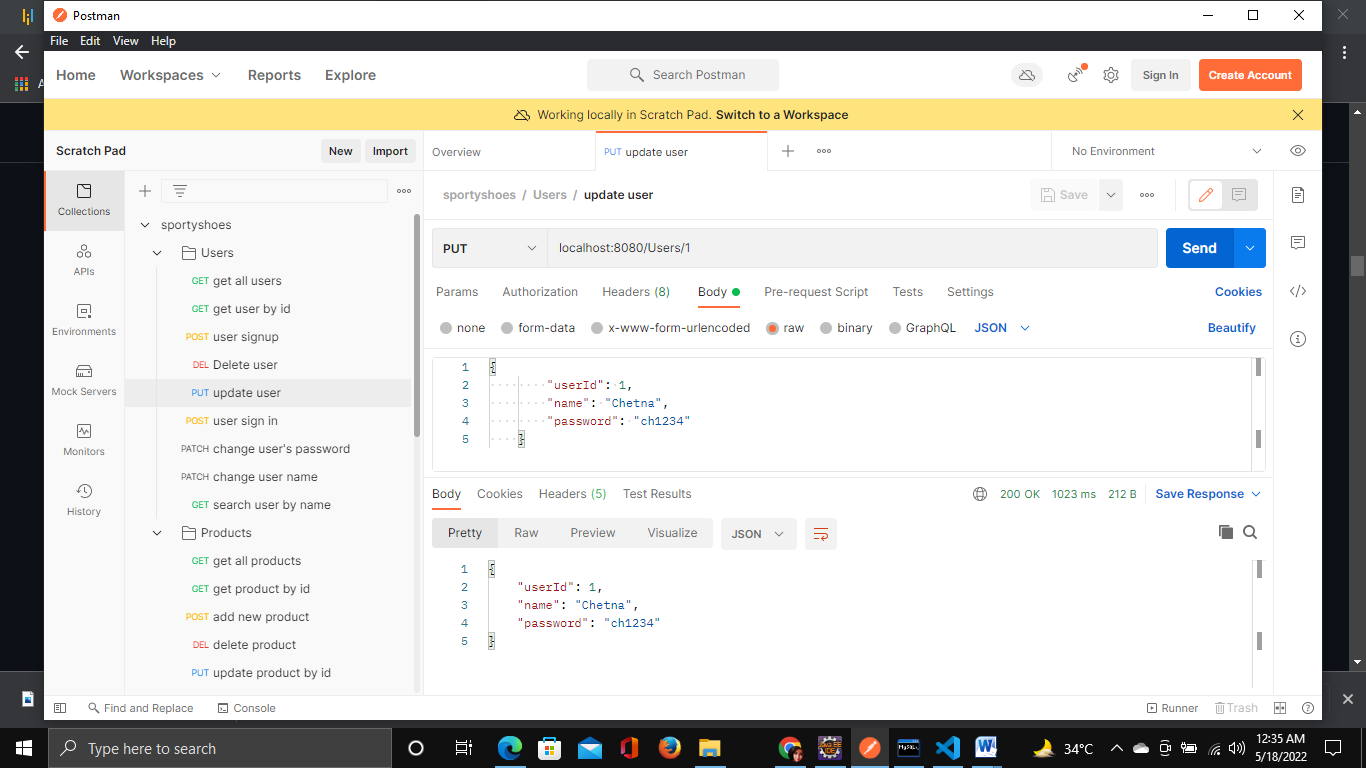
1. Getting user by user id.



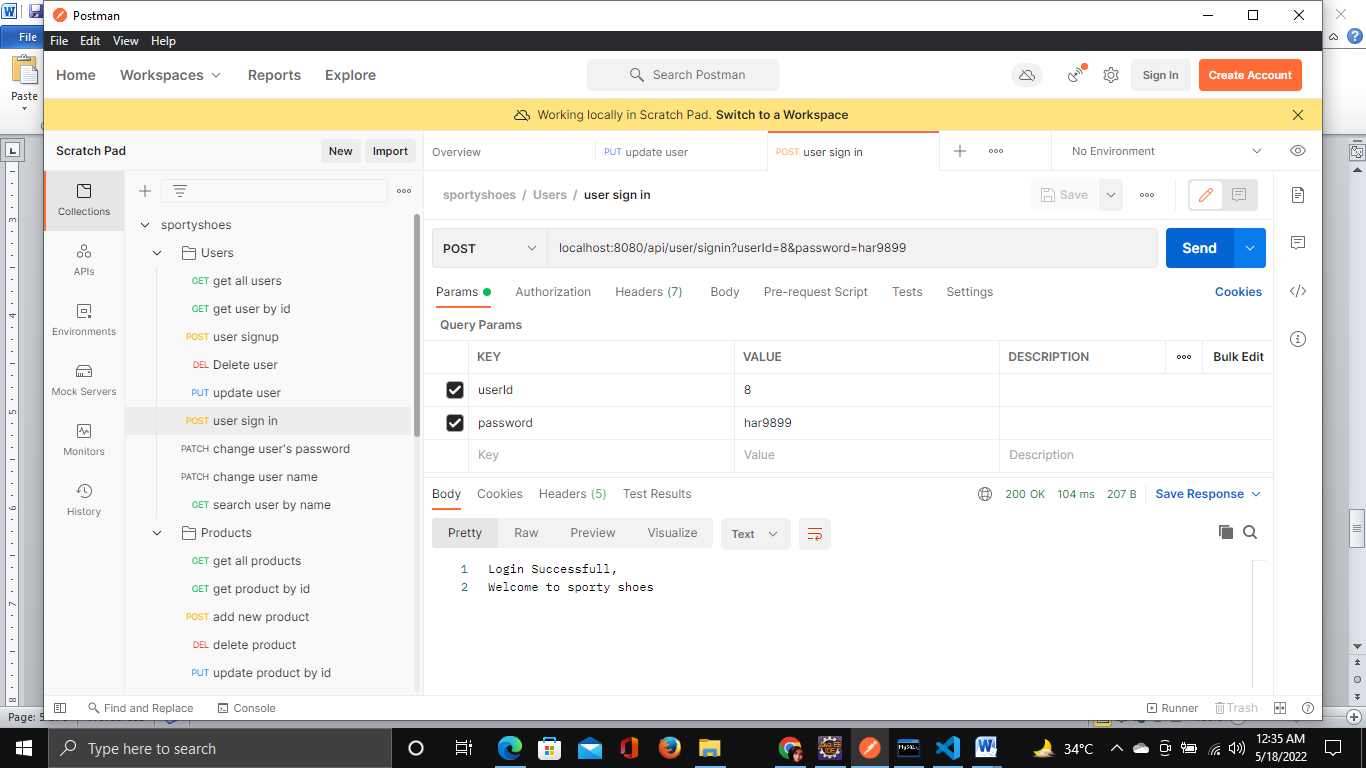
1. User sign up



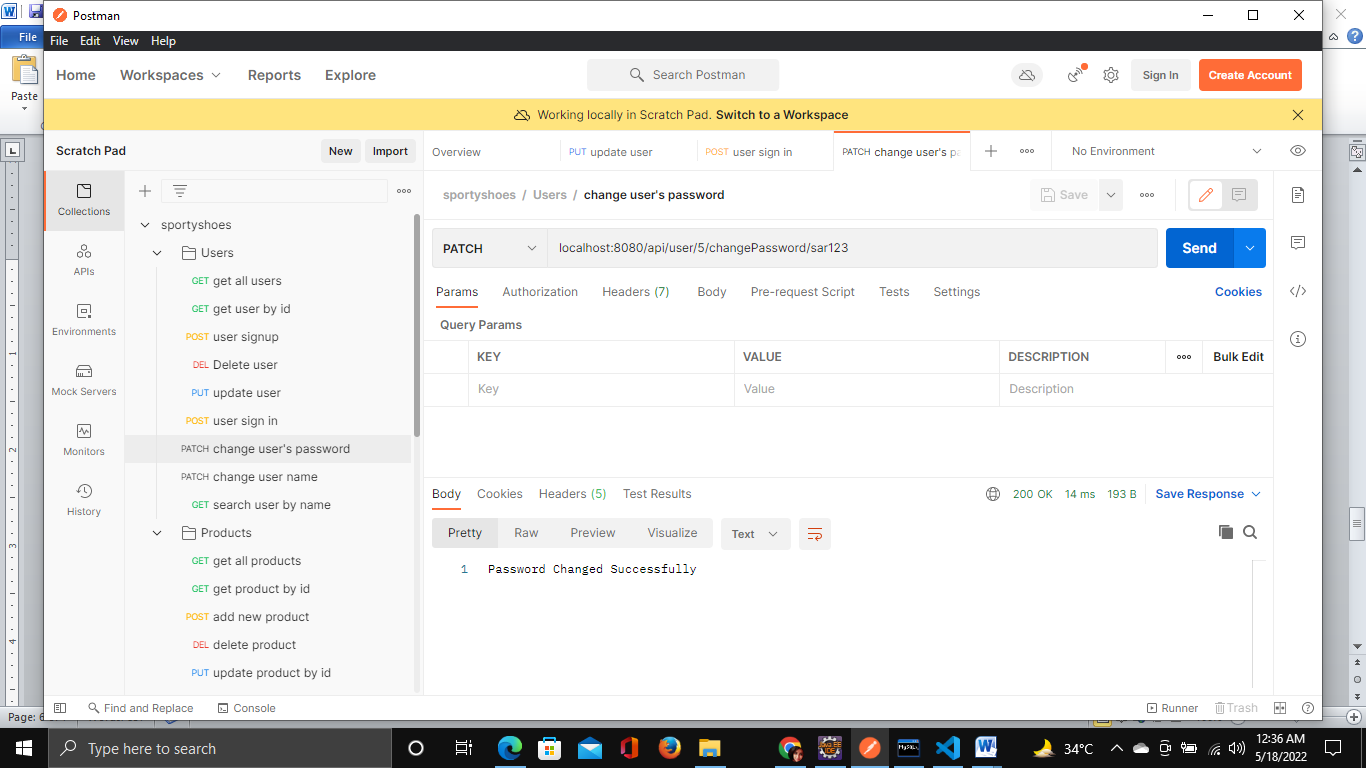
1. Update user details



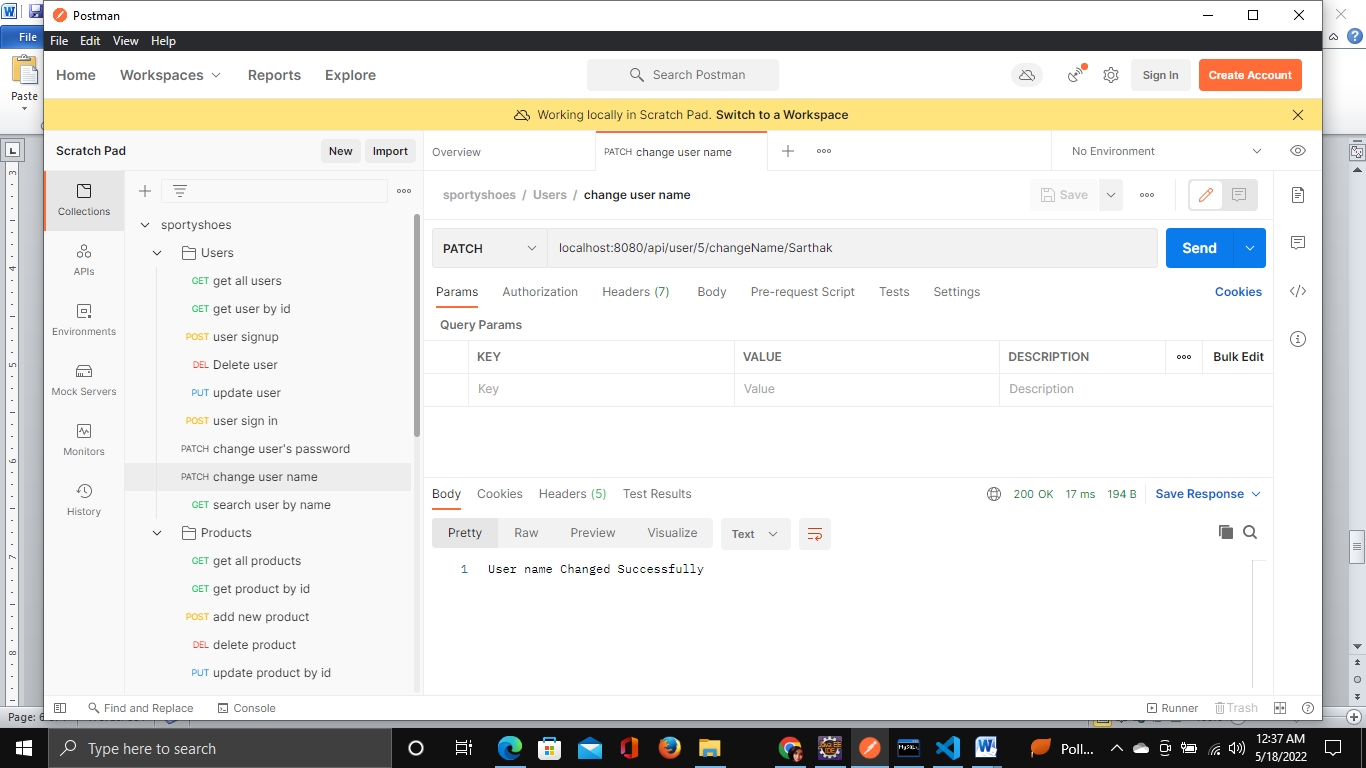
1. User sign in



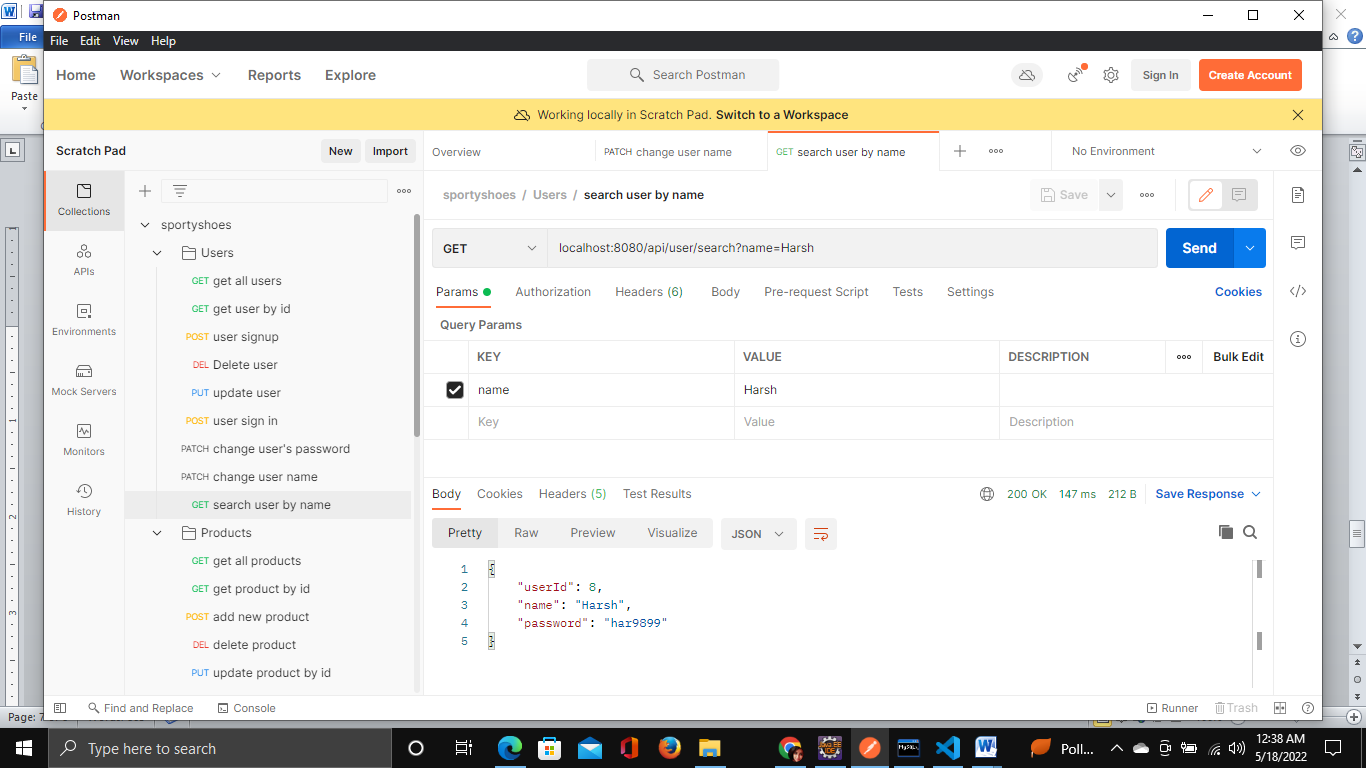
1. Update user password



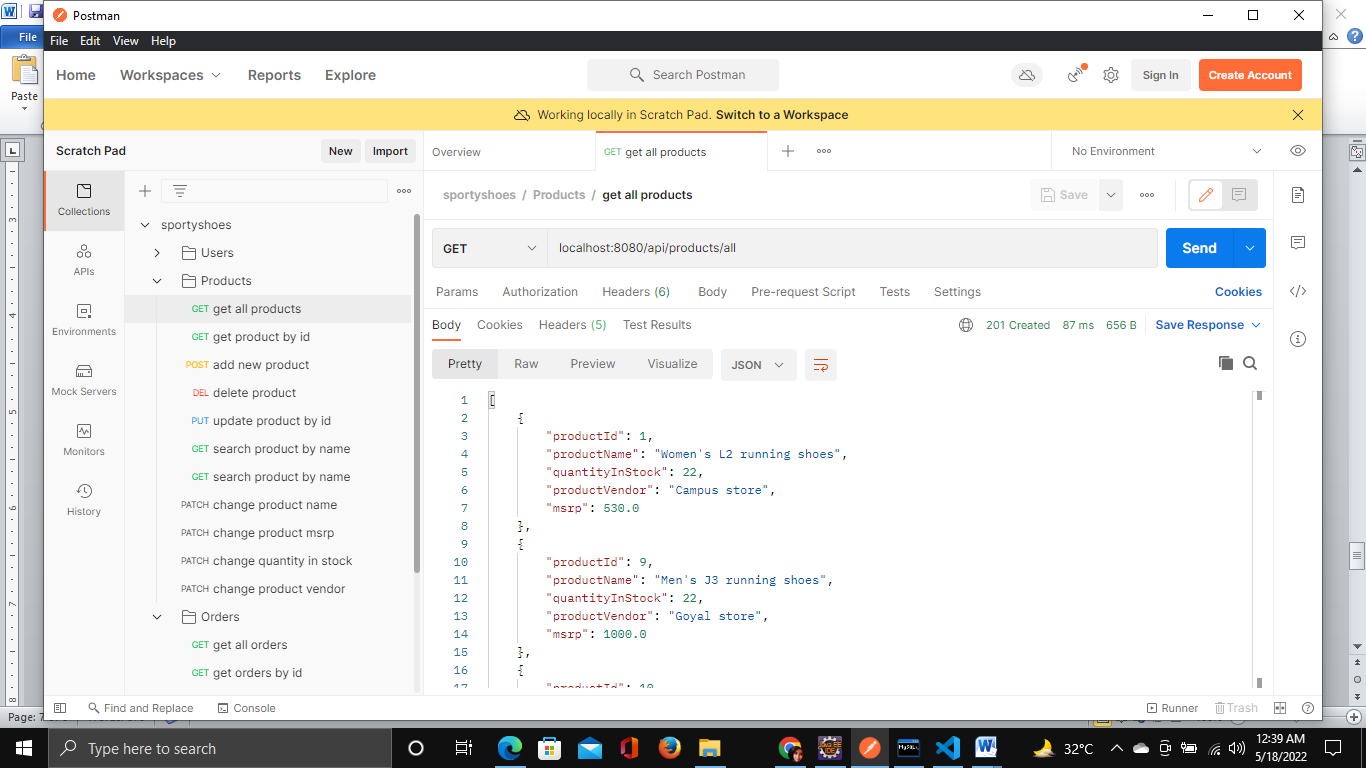
1. Update user name



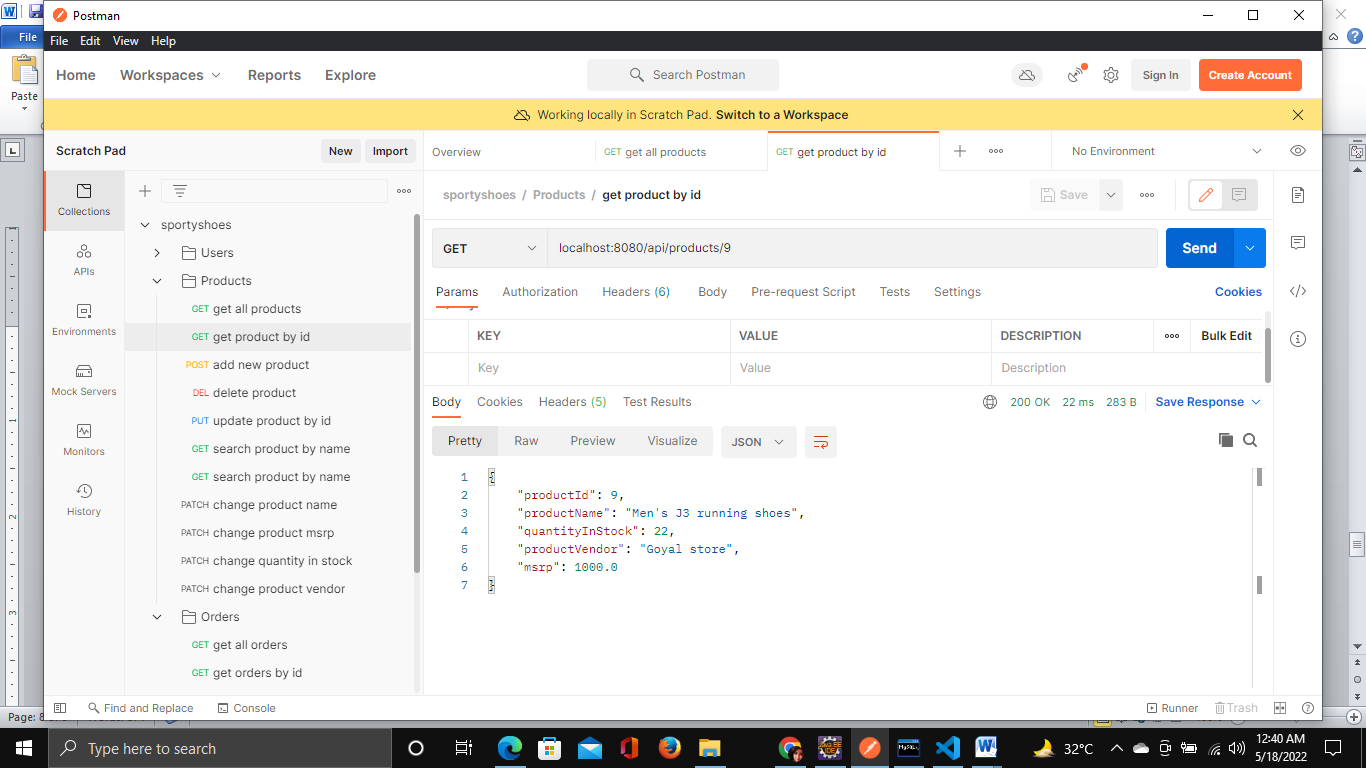
1. Search user by name



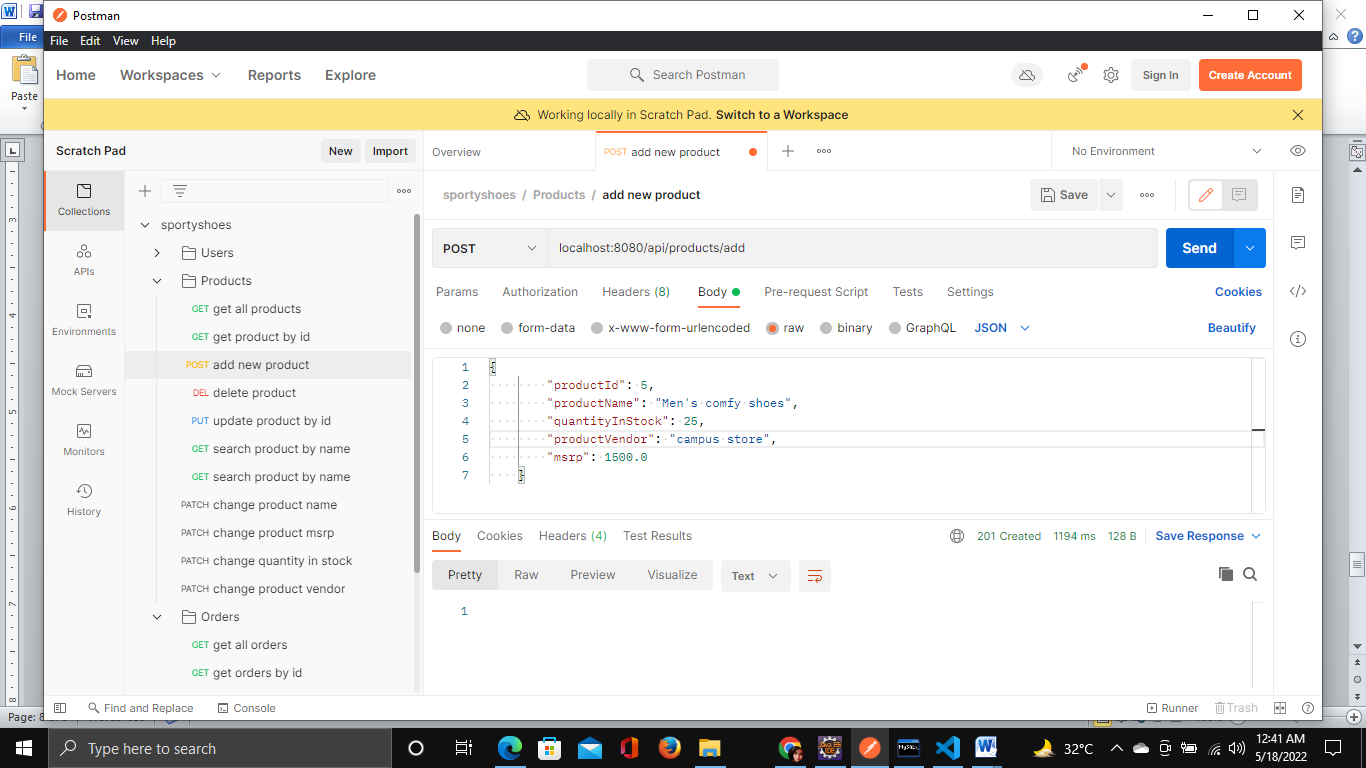
1. Get product details



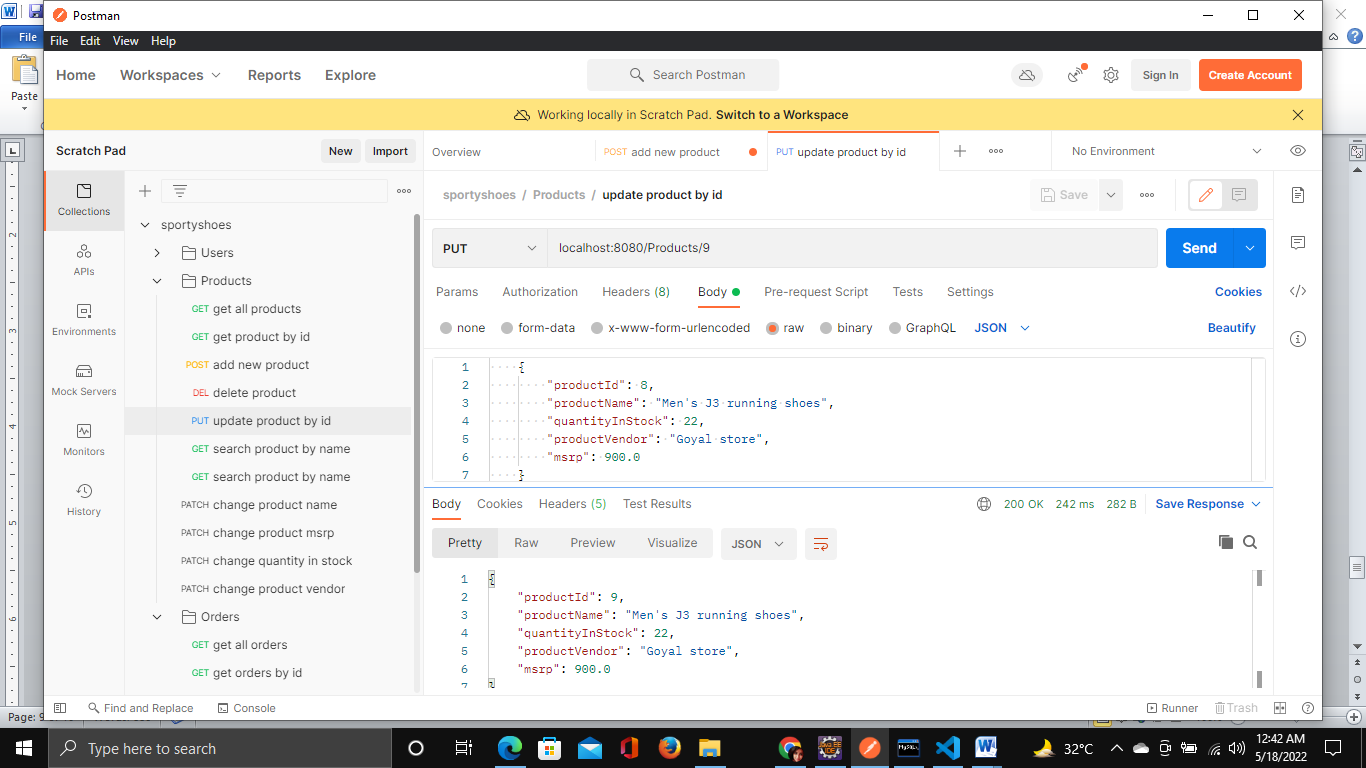
1. Get product details by id



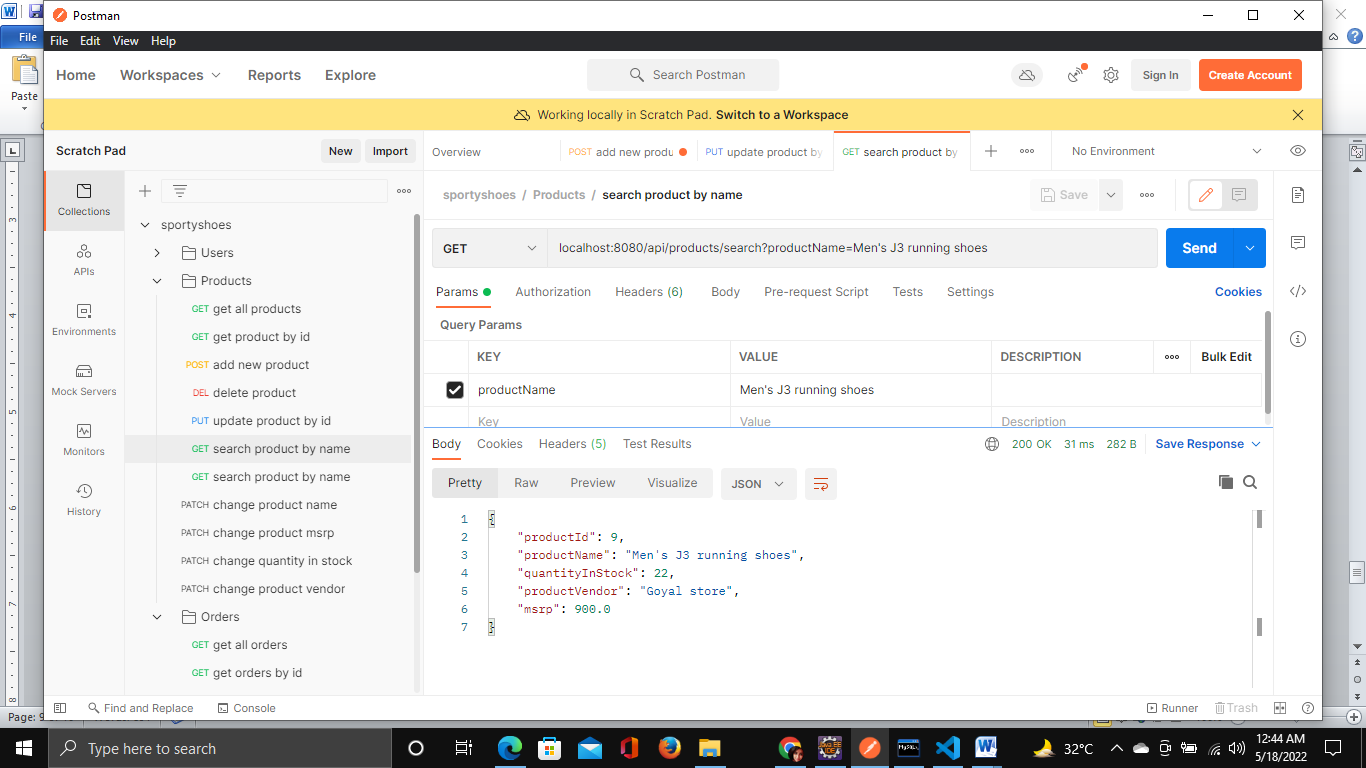
1. Add new product in inventory



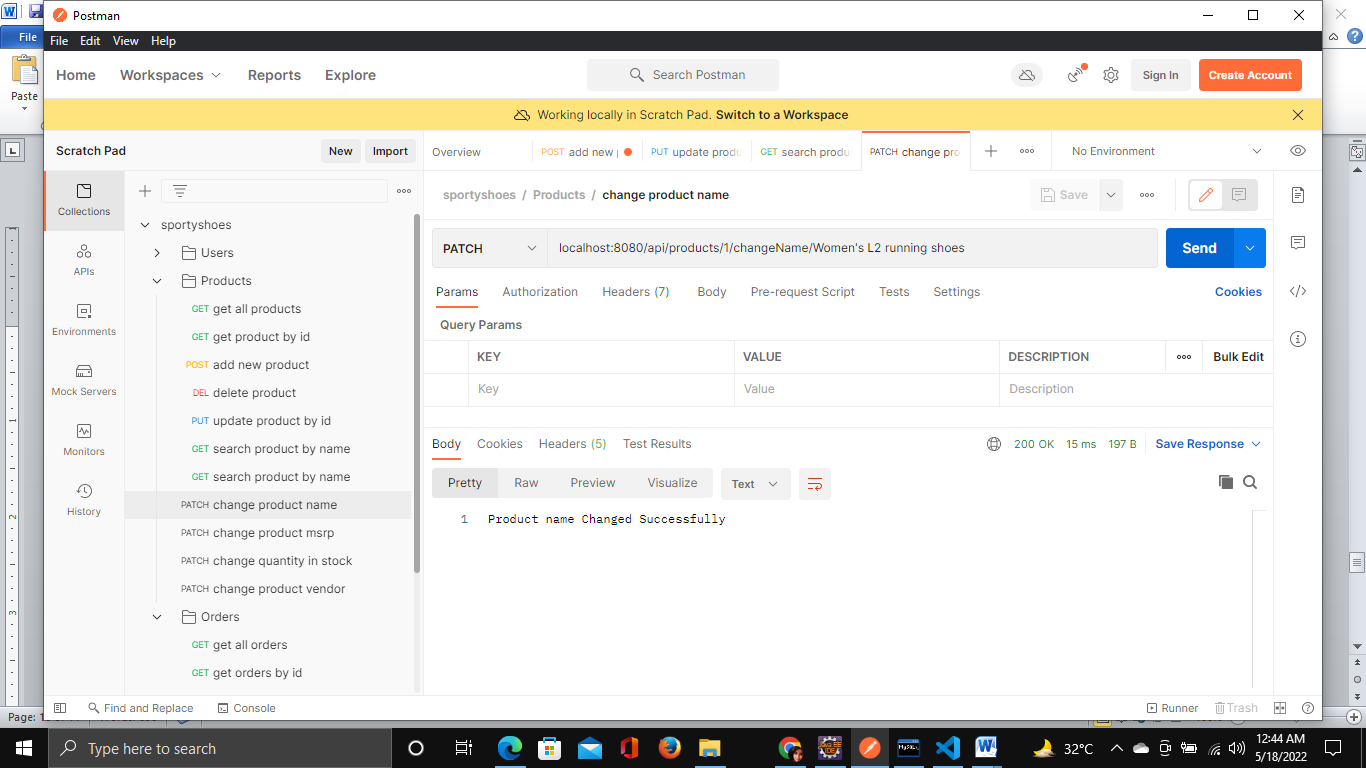
1. Update product by id



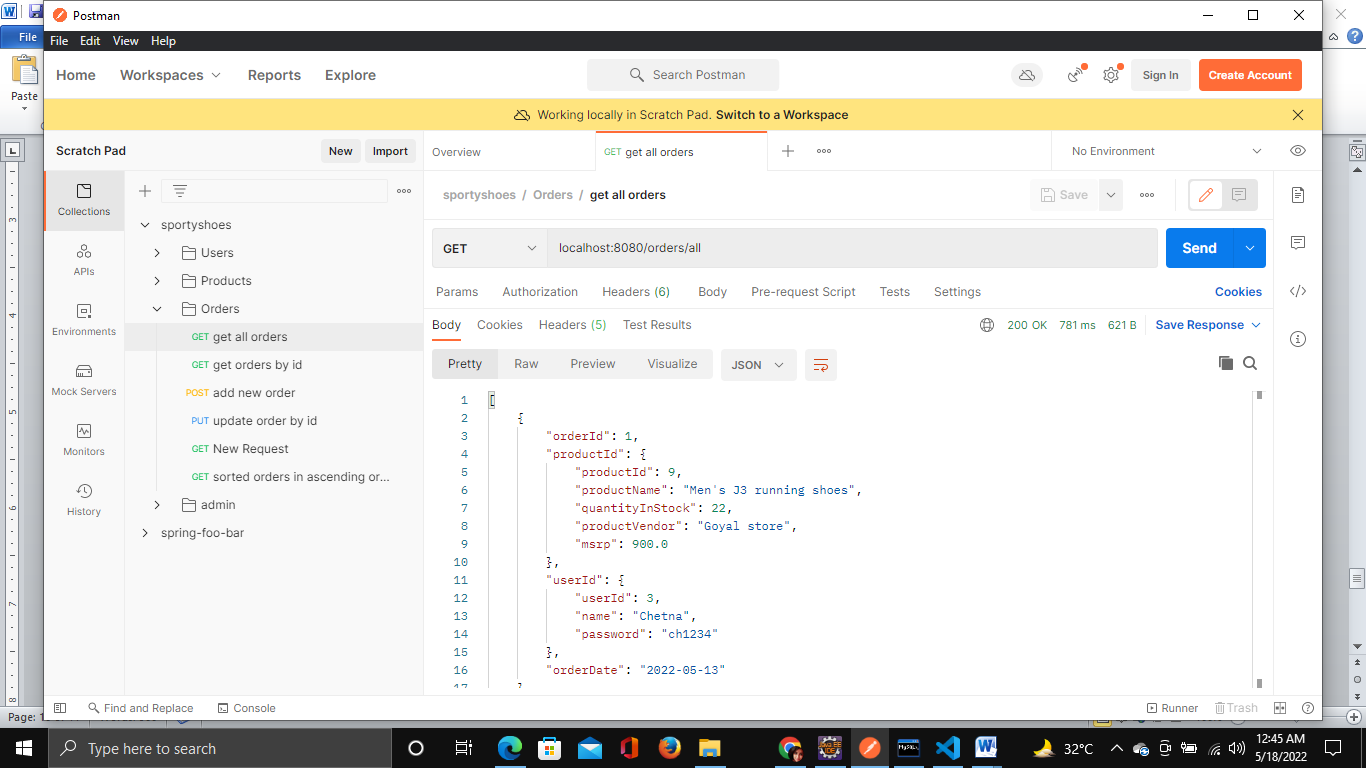
1. Search product by name



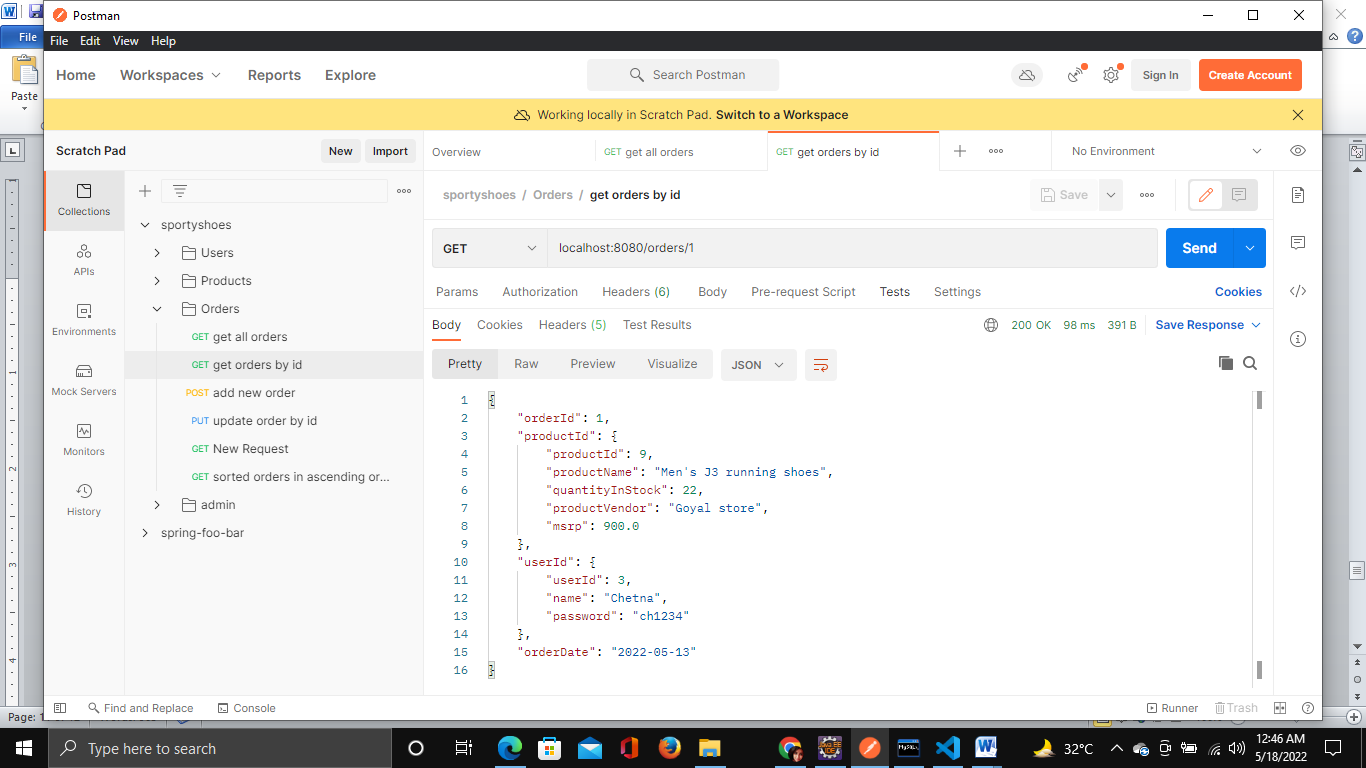
1. Update product details



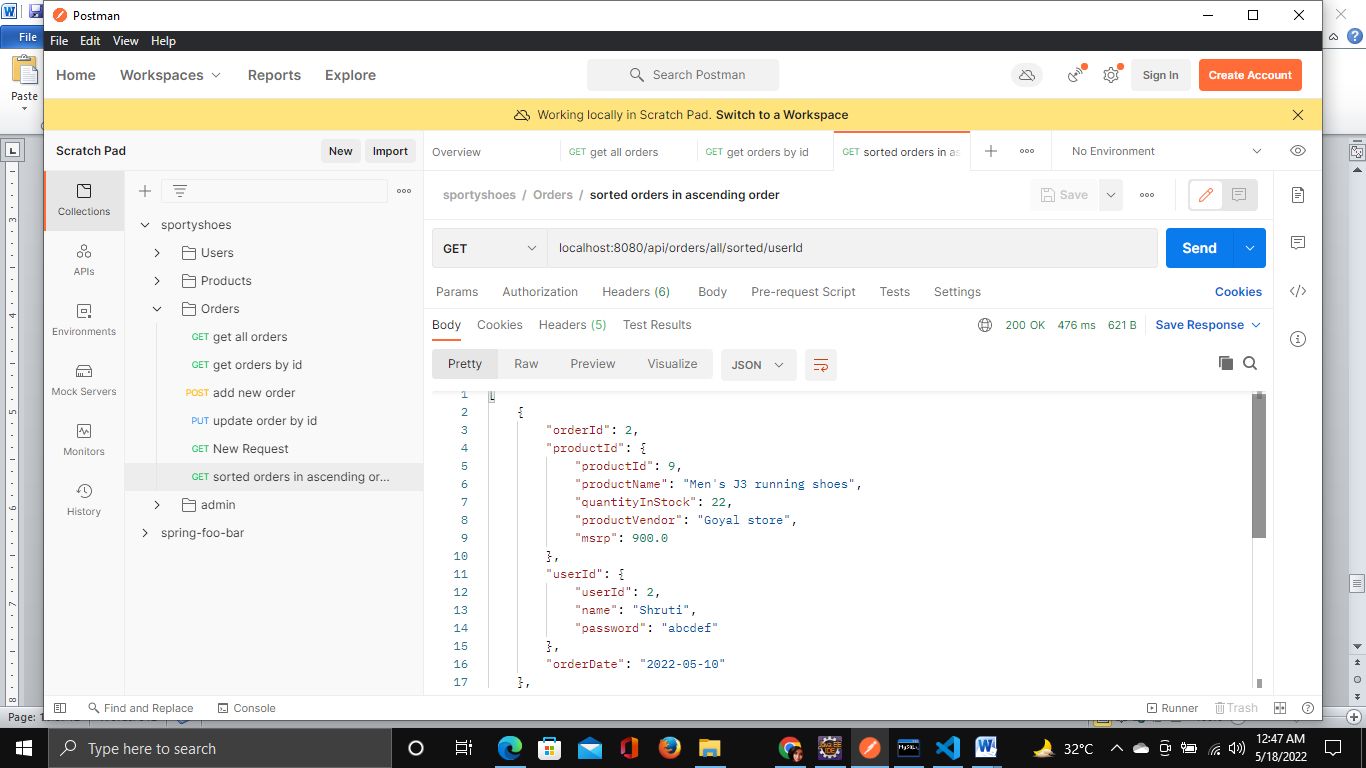
1. Get all order details



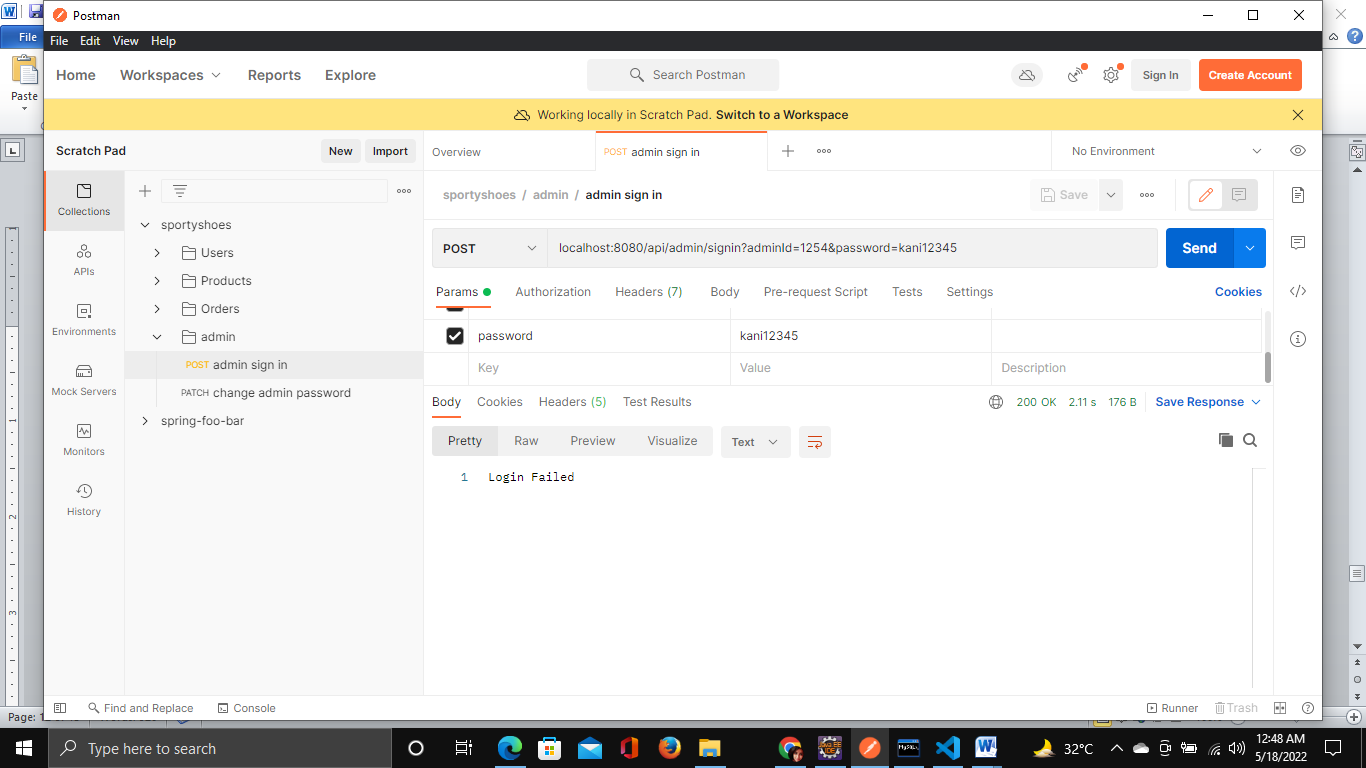
1. Get order details by order id



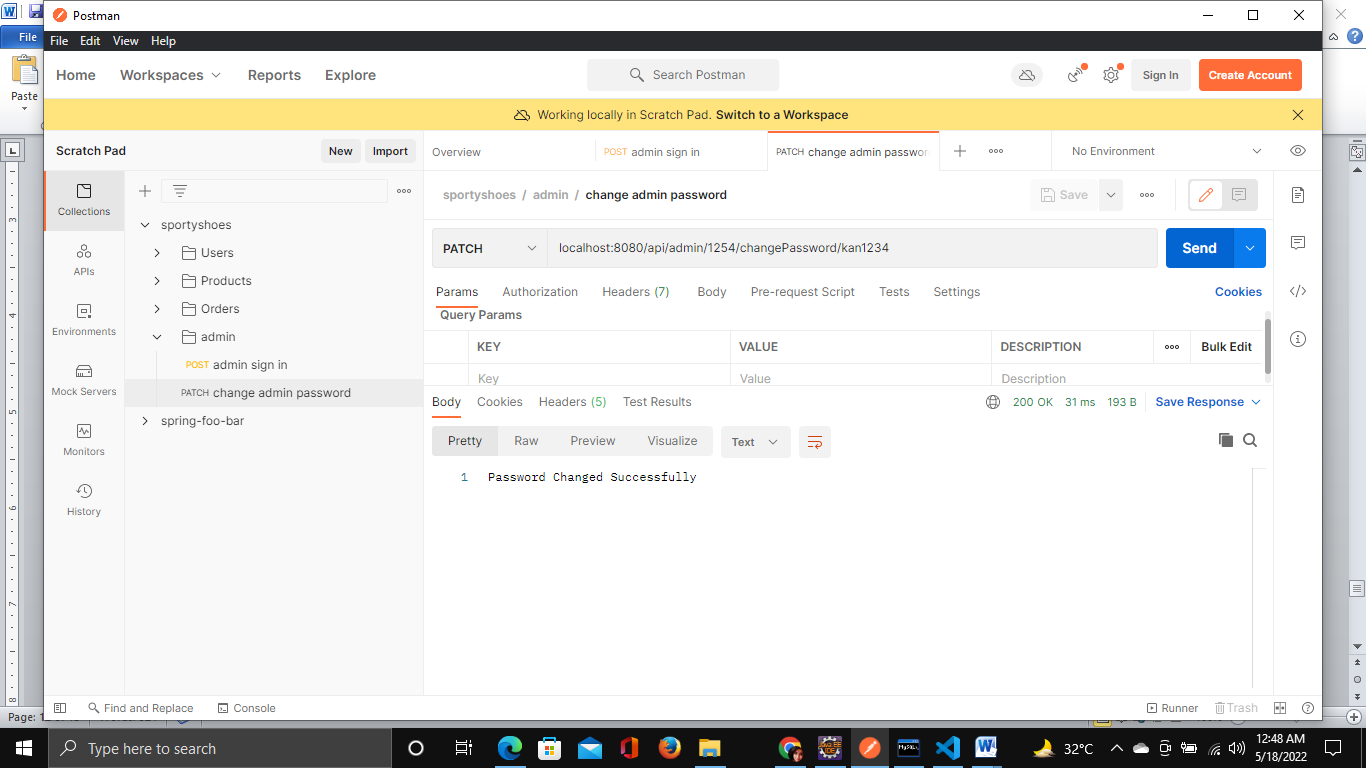
1. View order details sorted in ascending order



1. Admin sign in



1. Update admin password



**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 master**

## **Conclusions**

In the program an application has been developed in three sprints. This application handle the data of the Sporty shoes (An E-Commerce website). All the data about products are visible to user. The admin can login through an admin id and password and can manipulate the data.