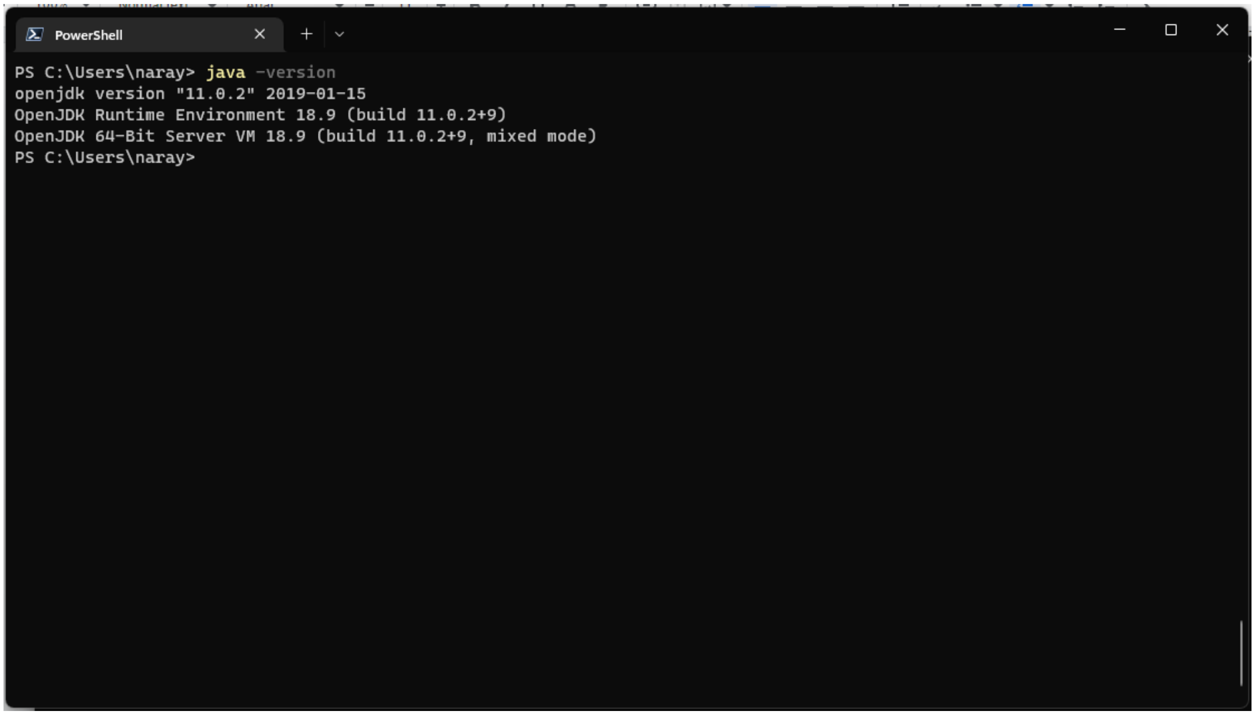
Name:Pavan Sainath Atukuri
U ID: 119397428
Secure Coding.

The project runs on spring boot. In this document, we'll see how to install the dependencies of the project one by one and we need Java to run the project.

- **Java installation**

Make sure java is installed on your machine. Java 11 is preferred. The project will also work on java 8. To check the type "java -version". It should output something like below.



```
PS C:\Users\naray> java -version
openjdk version "11.0.2" 2019-01-15
OpenJDK Runtime Environment 18.9 (build 11.0.2+9)
OpenJDK 64-Bit Server VM 18.9 (build 11.0.2+9, mixed mode)
PS C:\Users\naray>
```

If java is not installed. Follow the below steps

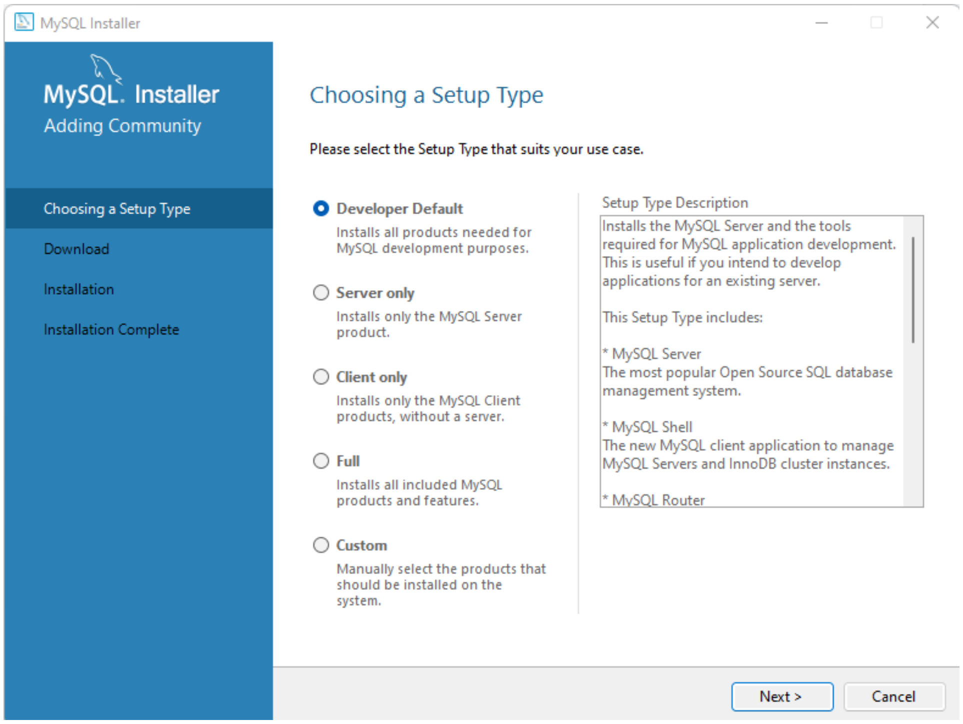
1. Download jdk.zip file using this link
2. Extract the zip anywhere in your system
3. Set JAVA_HOME and PATH variable using this link

- **MySQL installation**

1. Install mysql 8 version and mysql workbench to access the appication
    a. https://dev.mysql.com/downloads/installer/

Name:Pavan Sainath Atukuri
U ID: 119397428
Secure Coding.

b. Give default options in installation wizard



Choosing a Setup Type

Please select the Setup Type that suits your use case.

● **Developer Default**
Installs all products needed for MySQL development purposes.

○ **Server only**
Installs only the MySQL Server product.

○ **Client only**
Installs only the MySQL Client products, without a server.

○ **Full**
Installs all included MySQL products and features.

○ **Custom**
Manually select the products that should be installed on the system.

Setup Type Description

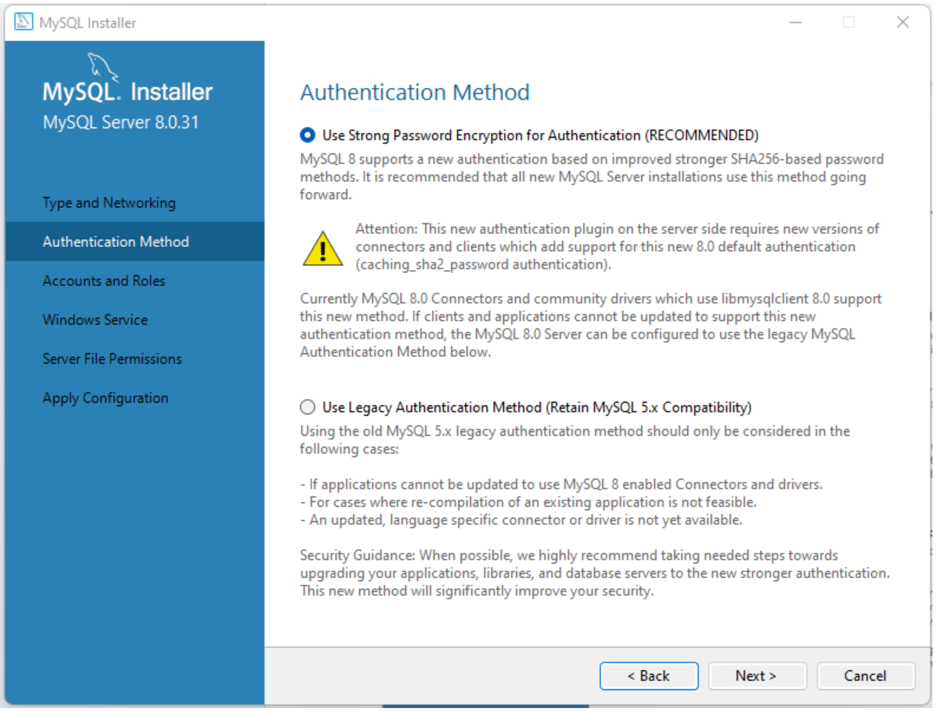
Installs the MySQL Server and the tools required for MySQL application development. This is useful if you intend to develop applications for an existing server.

This Setup Type includes:

* MySQL Server
The most popular Open Source SQL database management system.

* MySQL Shell
The new MySQL client application to manage MySQL Servers and InnoDB cluster instances.

* MySQL Router



Authentication Method

● Use Strong Password Encryption for Authentication (RECOMMENDED)

MySQL 8 supports a new authentication based on improved stronger SHA256-based password methods. It is recommended that all new MySQL Server installations use this method going forward.

Attention: This new authentication plugin on the server side requires new versions of connectors and clients which add support for this new 8.0 default authentication (caching_sha2_password authentication).

Currently MySQL 8.0 Connectors and community drivers which use libmysqlclient 8.0 support this new method. If clients and applications cannot be updated to support this new authentication method, the MySQL 8.0 Server can be configured to use the legacy MySQL Authentication Method below.

○ Use Legacy Authentication Method (Retain MySQL 5.x Compatibility)

Using the old MySQL 5.x legacy authentication method should only be considered in the following cases:

- If applications cannot be updated to use MySQL 8 enabled Connectors and drivers.
- For cases where re-compilation of an existing application is not feasible.
- An updated, language specific connector or driver is not yet available.

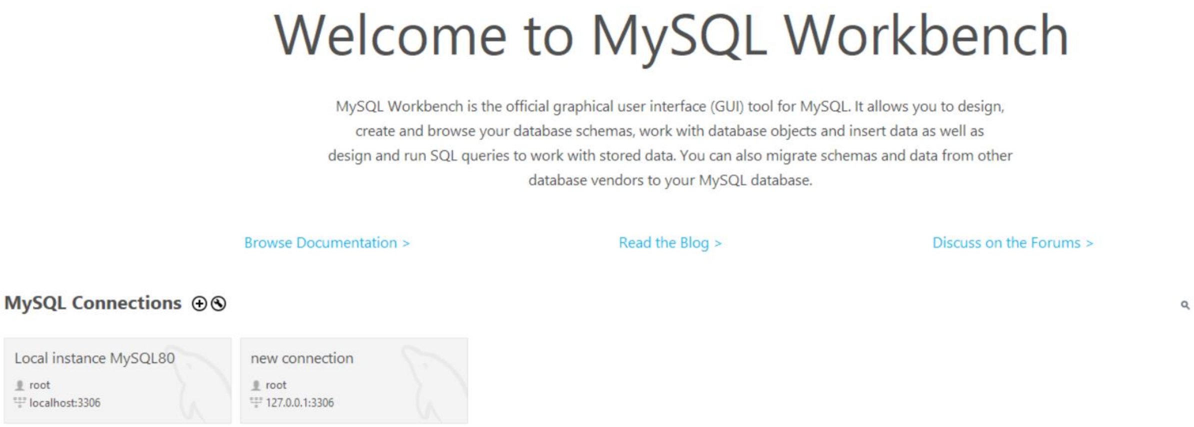
Security Guidance: When possible, we highly recommend taking needed steps towards upgrading your applications, libraries, and database servers to the new stronger authentication. This new method will significantly improve your security.

2.

Name:Pavan Sainath Atukuri
U ID: 119397428
Secure Coding.

• MySQL Workbench configuration

1. After installation, open the MySQL workbench and follow the prompts and please give user name as root and password and please make a note of these credentials as we need to use the credential later.Please give below credentials as username and pasword
   Username:root
   Password:Maryland@77@
2. Open the Mysql and it looks like the below image



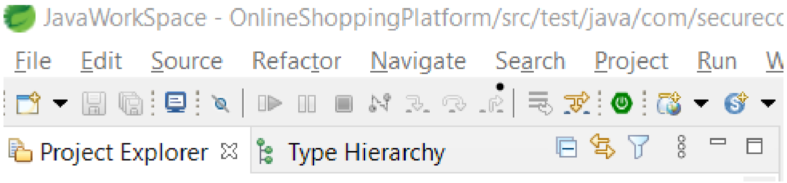
Welcome to MySQL Workbench

MySQL Workbench is the official graphical user interface (GUI) tool for MySQL. It allows you to design, create and browse your database schemas, work with database objects and insert data as well as design and run SQL queries to work with stored data. You can also migrate schemas and data from other database vendors to your MySQL database.

3. Nowclick on the instance and login and create Schema and give name as "`secure_online_shoppingplatform`"

4. Install Spring Tool Suite(STS) using the below link
   https://spring.io/tools
   please follow the prompts and install STS IDE.

5. Now open the STS and follow the below steps to upload the Java project from the file systems to the STS IDE.
6. Click on File->Open Projects from file system.



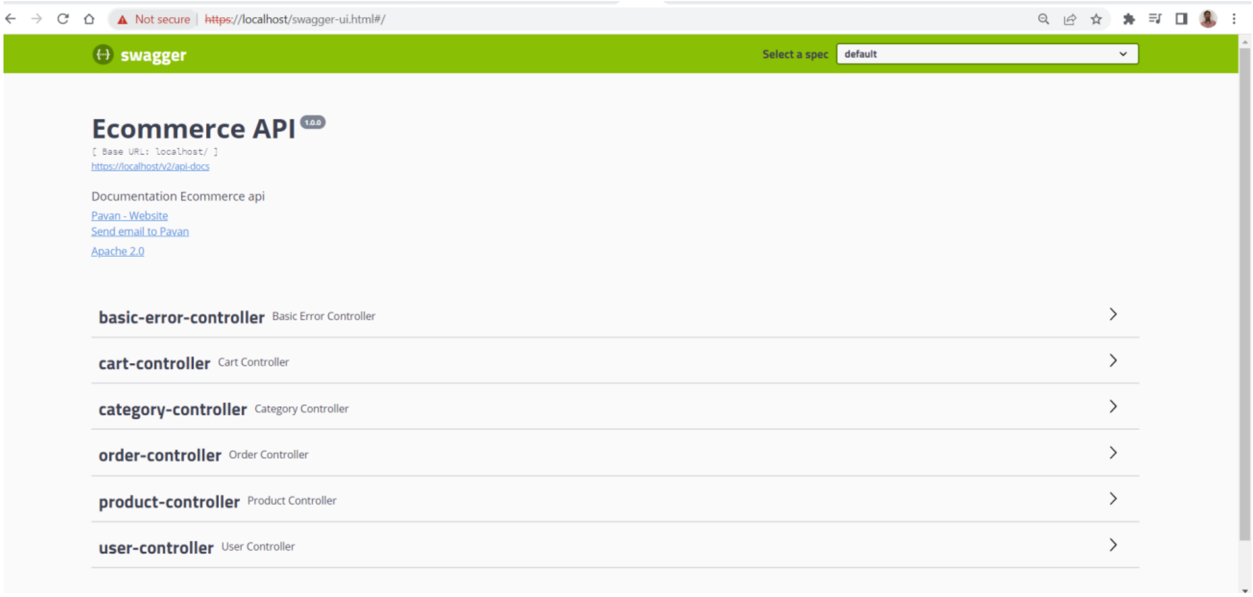
7.

Name:Pavan Sainath Atukuri
U ID: 119397428
Secure Coding.

7.Now the Right click on the project and Click on Maven->Update Project->and checkbox Force Update snapshots
8. Now Right Click on the project, Maven-> Install
9. Now again right click on the project, Run as “Spring boot application”
10.No open Google chrome and click on the below link

https://localhost/swagger-ui.html#/

11. You should be able to see a screen like below.



Ecommerce API 1.0.0
[Base URL: localhost/]
https://localhost/v2/api-docs

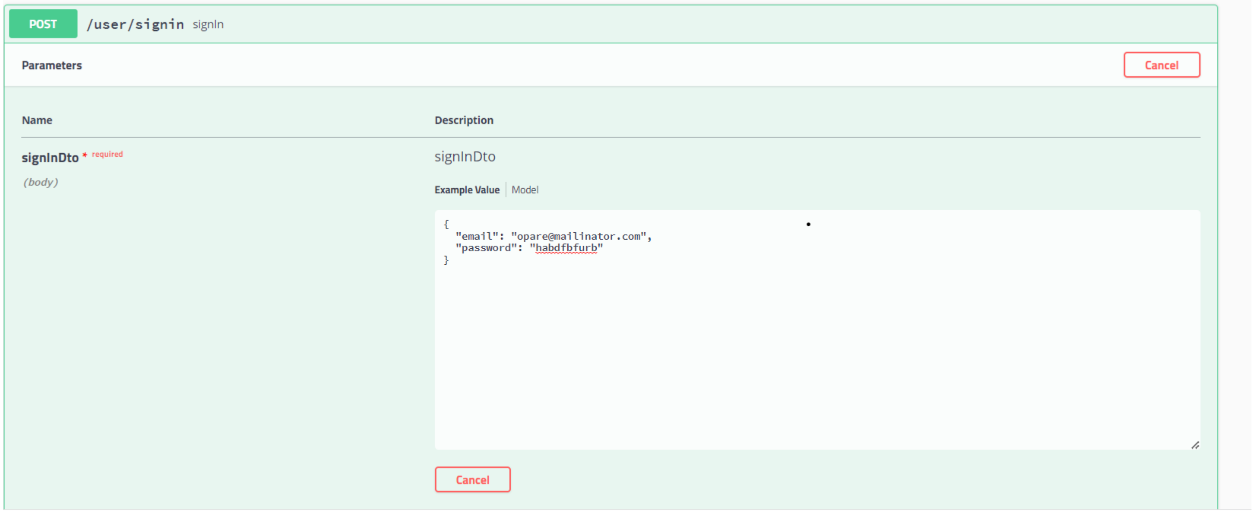
Documentation Ecommerce api
Pavan - Website
Send email to Pavan
Apache 2.0

basic-error-controller Basic Error Controller
cart-controller Cart Controller
category-controller Category Controller
order-controller Order Controller
product-controller Product Controller
user-controller User Controller

Using the REST End Points of the application using Swagger

Name:Pavan Sainath Atukuri
U ID: 119397428
Secure Coding.

- Click on the User Controller and then click on the tryout and give the credentials and click on execute.



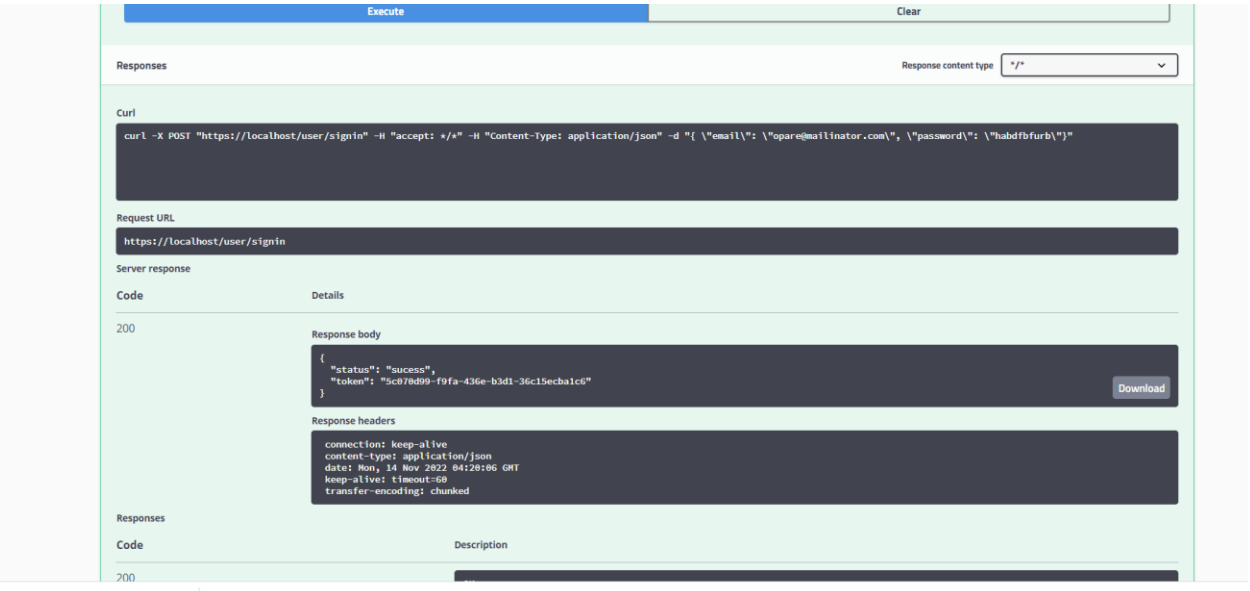
```
{
  "email": "opare@mailinator.com",
  "password": "habdfbfurb"
}
```

This will return the response body of the Sign API and please be sure once you are signed in you will get a token as below and this will be necessary for all the subsequent requests and API calls.
Please Note: For all the subsequent calls we need to add token in the header or in the request body to get the required API response.

Swagger will prompt you to enter the token wherever required either as a Request header or the Request param

Token will appear as in the below screenshot

Name:Pavan Sainath Atukuri
U ID: 119397428
Secure Coding.



Responses

Response content type */*

Curl

```
curl -X POST "https://localhost/user/signin" -H "accept: */*" -H "Content-Type: application/json" -d "{ \"email\": \"opare@mailinator.com\", \"password\": \"habdfbfurb\"}"
```

Request URL

```
https://localhost/user/signin
```

Server response

Code Details

200

Response body

```
{
  "status": "sucess",
  "token": "5c070d99-f9fa-436e-b3d1-36c15ecba1c6"
}
```

Response headers

```
 connection: keep-alive
 content-type: application/json
 date: Mon, 14 Nov 2022 04:20:06 GMT
 keep-alive: timeout=60
 transfer-encoding: chunked
```

Responses

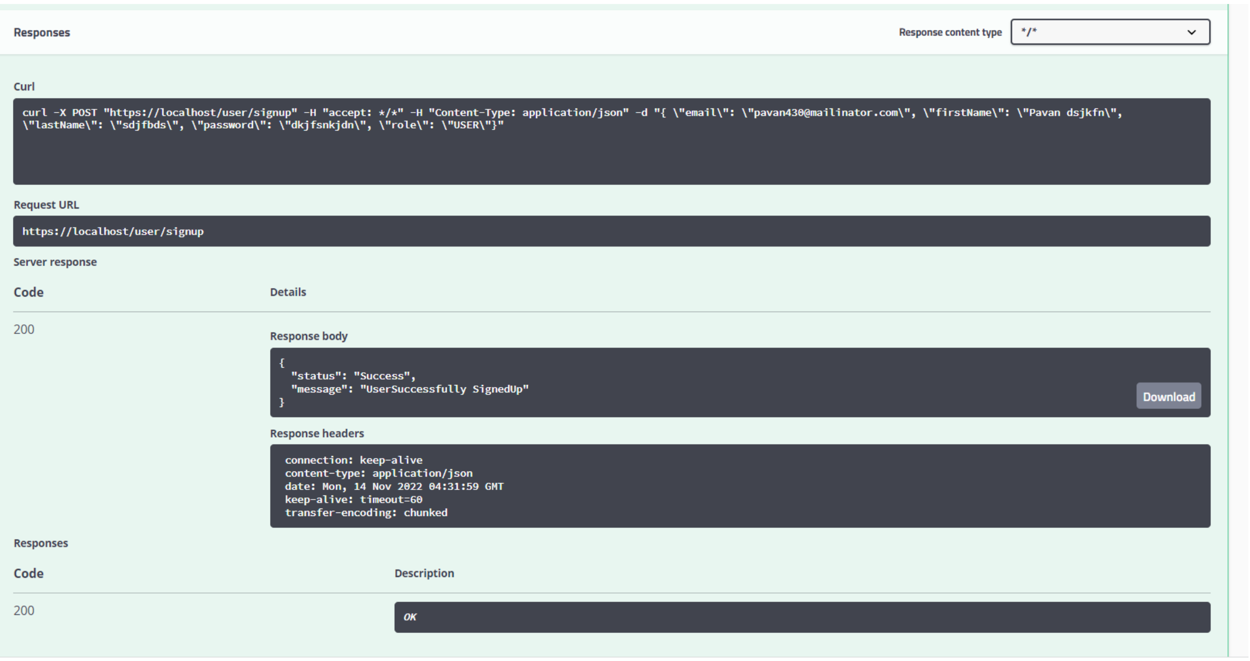
Code Description

200

This token is act as a Authorization token of the user and will be expired after 60 mins and after that we cannot make any API call. We need to login again to get the new token to and perform any action.

Signup:

Once we enter the all the required credentials as in below screenshot,user is signup done and he will get a remail to confirm his account in 15 mins or else his account will not be enabled to login or perform any other action.



Responses

Response content type */*

Curl

```
curl -X POST "https://localhost/user/signup" -H "accept: */*" -H "Content-Type: application/json" -d "{ \"email\": \"pavan430@mailinator.com\", \"firstName\": \"Pavan dsjkfn\",
\"lastName\": \"sdjfbds\", \"password\": \"dkjfsnkjdn\", \"role\": \"USER\"}"
```

Request URL

```
https://localhost/user/signup
```

Server response

Code Details

200

Response body

```
{
  "status": "Success",
  "message": "UserSuccessfully SignedUp"
}
```

Response headers

```
 connection: keep-alive
 content-type: application/json
 date: Mon, 14 Nov 2022 04:31:59 GMT
 keep-alive: timeout=60
 transfer-encoding: chunked
```

Responses

Code	Description
200	*OK*

Name:Pavan Sainath Atukuri
U ID: 119397428
Secure Coding.

We cannot register as a ADMIN as only one admin can be allowed to the system from backend.

```
insert into users (email,first_name,is_confirmed,last_name,password,role) values
("admin@secureplatform.com",
"admin",1, "admin","admin123","ADMIN")
```

So, there is no more admin allowed to register in to the register.

Admin only has certain privileges and he can perform certain actions as mentioned in the system design document.
Accordingly Sellers are allowed to certain API end points using their authorities.
Below are the few endpoints where users can be accessed according to theier roles and authorities
/user/signin->All users
/user/signup/-?All users
user/logout->all users
user/viewevenue-> only Admin
/user/viewAccount-> All users
/order/add-All users.