SPRINGBOOT ANGULAR PROJECT

RAILWAY RESERVATION SYSTEM

Project Member:

Ms.Anugiraha
Ms.Deepa
Ms.Gopika
Ms.Kamaleeswari
Ms.Sivaranjani
Ms.Vinodhini

Guided By :

Prof. Indrakka Mali

Title :

- Introduction
- Objective
- Technology to be Used
- Proposed System
- Snapshots
- Diagrams
- Advantages
- Conclusion

ABSTRACT :

- The Indian Railways (IR) carries about 5.5 lakhs passengers in reserved accommodation every day. The Computerized Passenger Reservation System (PRS) facilities the booking and cancellation of tickets from any of the 4000 terminals (i.e. PRS booking window all over the countries). These tickets can be booked or cancelled for journeys commencing in any part of India and ending in any other part, with travel time as long as 72hours and distance up to several thousand kilometers.

- In the given project we will be developing a website which will help users to find train details, book and cancel tickets and the exact charge of their tickets to the desired destination.

- With the help of online booking people can book their tickets online through internet, sitting in their home by a single click of mouse. Using this website people can easily get their tickets done within minutes.

Introduction:

- This web application is developed to manage bookings of trains using a single medium. Admin who is the main user in this application has the responsibility of adding trains, seat capacity, the fare for the train, source, and destination. The user is another role in this application who will book tickets using this application.

- This Project has all the necessary functionality from adding, searching, and deleting trains from the system to booking tickets.
- Online railway reservation Project is proposed by Backend and frontend. This project is designed to automate the process of booking train tickets and managing the whole process online.

- The Railway Reservation system is built to support all the functionality related to the process of booking, trains, tracking, and managing. We will discuss each aspect of this project in detail for enhanced understanding.

Objective :

- The advancement of the digital era has led to online booking systems because of obvious reasons. As a user, we want our train booking also should be automated.

- The main idea is to implement a proper process to system.

- In our existing system contains a many operations like user registration, user search, chart etc..

- The organization can maintain computerized records without redundant entries registration, student search, fees, details of the student etc..

Software & Hardware used:

Software Requirements :

- Operating System : Windows 7 or Windows 10
- Language : Java
- IDE : Spring Boot, Postman

- Backend : Microsoft MySQL server

- Server : Tomcat 8.5

- Frontend : visual studio code, bootstrap, Angular, CSS, HTML.

Hardware Requirements :

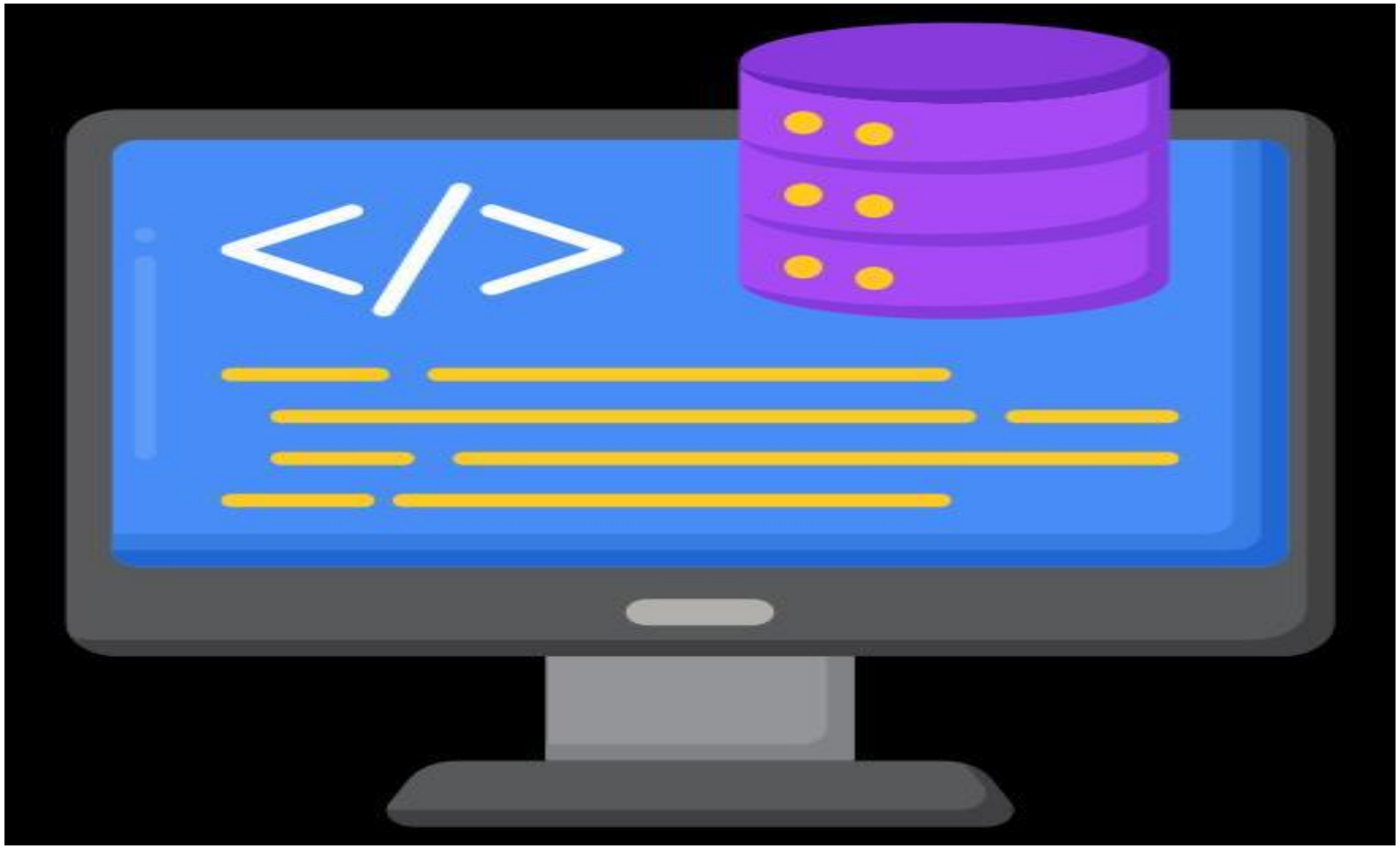
- CPU : Intel Pentium IV processor
- RAM : 512 MB or above
- Hard Disk : 2 GB hard disk space or minimum

Proposed System:

- The Railway Reservation System facilitates the passengers to enquire about the trains available on the basis of source and destination, Booking and Cancellation of tickets, enquire about the status of the booked ticket, etc. The aim of case study is to design and develop a database maintaining the records of different trains, train status, and passengers.

- This is a web-related application that permits us to approach the entire knowledge regarding to the passengers (user), train.

Back-End :



- Core Java

- Spring Boot
- Spring Data JPA
- Hibernate
  - One to many relation
  - Many to one relation
- Spring Boot Web
- MySQL Database

About Back-End:

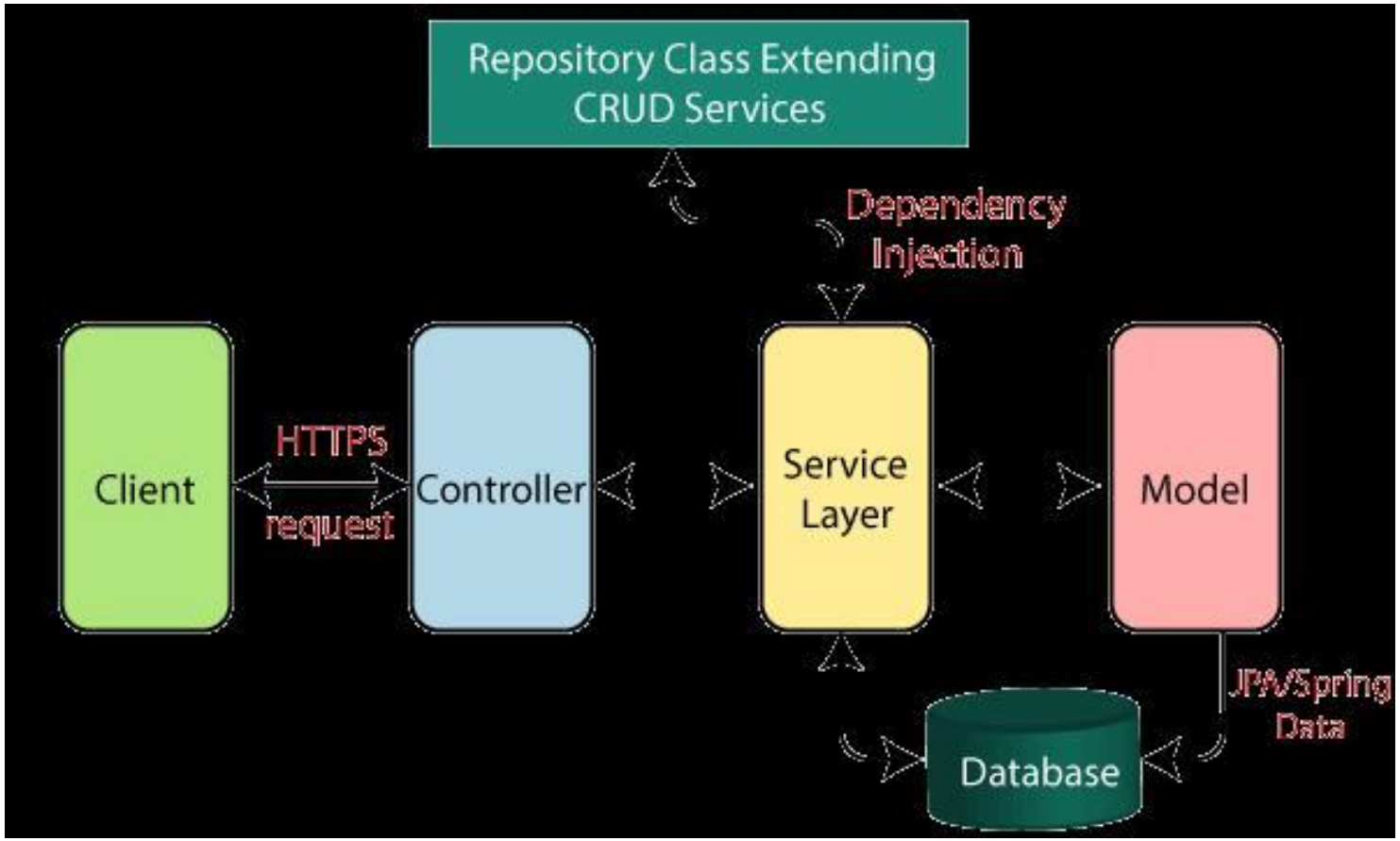
- There are mainly four operations will be performed by the user and admin Ex. Insert, update, retrieve/fetch, delete.

- These operation will be performed using the spring boot framework, core java and spring boot web , spring data Jpa , and hibernate.

- For connectivity purpose we are using MySQL database. It is connected to the java and database.

Working Of Back-End :

- There are mainly three stages



- UIController - In Spring Boot, the train Controller class is responsible for processing incoming requests, preparing a model, and returning the view to be rendered as a response.

- Service - Components are the class file which contains @Service annotation.

  - ✓ These class files are used to write business logic in a different layer.

- Entity – The entities are the persistence object stores as a record entry database.

- ✓ In an application entity manages instances manage it.
- ✓ The set of entity classes represents the data contained within a single data store.

LIST OF ENTITIES AND ATTRIBUTES:

ENTITIES	ATTRIBUTES
Train	**trainId**

	trainName from_to day time capacity charge

User	**<u>userId</u>** userName gender date phNo adharNo

	train

- Repository - Repository is a specialization of @Component annotation which is used to indicate that the class provides the mechanism for storage, retrieval, update, delete and search operation on objects.

  - ✓ Repository is directly connected with the database and then it can return to the repository and response to the service and it can return to the controller and then response send to the client.

Advantages:

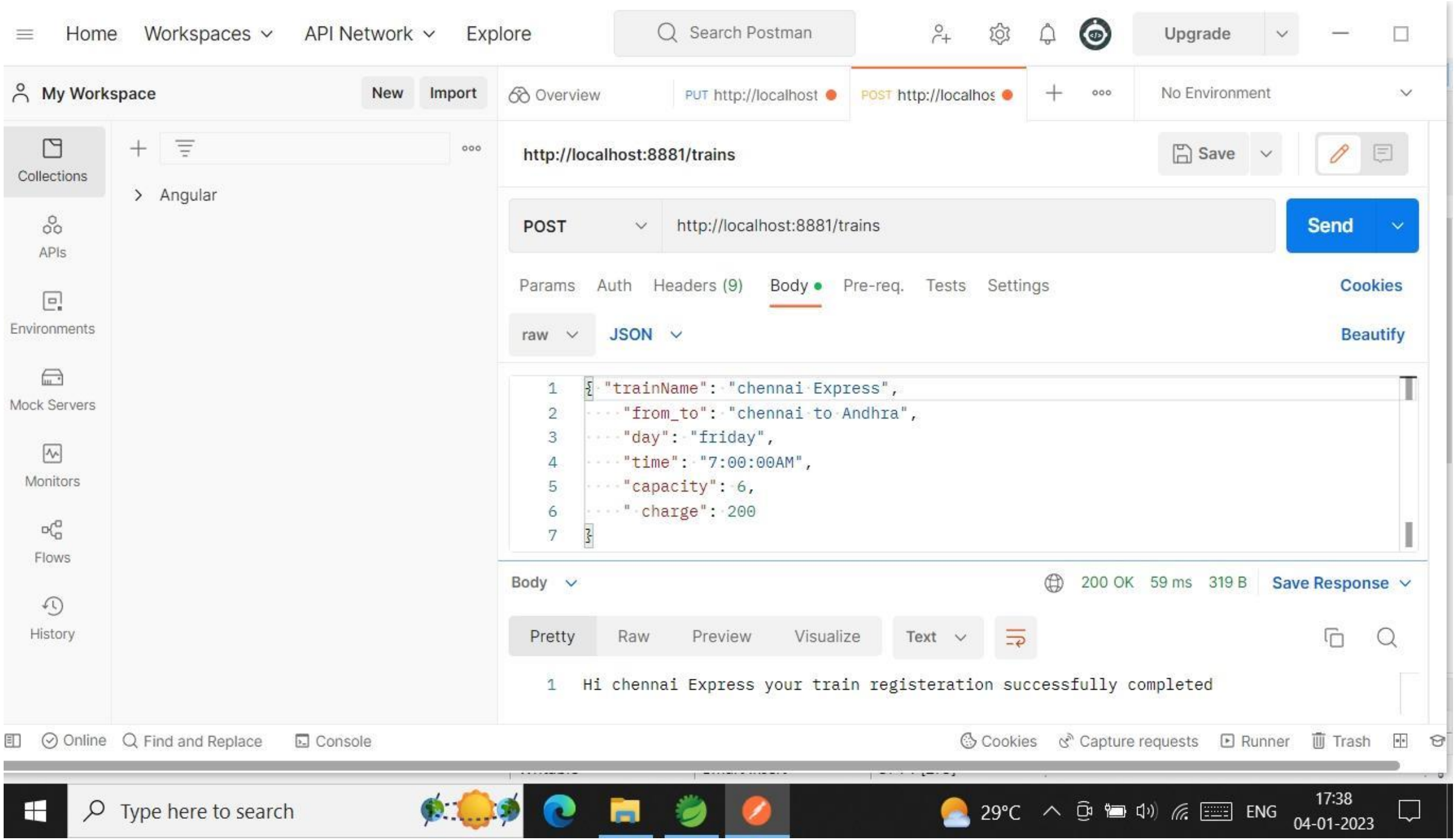
- Better and improved charge details
- Better user performance
- Drastic reduction in manual tasks
- Elimination of paper based tasks
- A reservation system increases user satisfaction.

- Makes the reservation smart and ready for the future.
- Resource utilization.
- Duplication of the user and train data is avoided.

Snap Shots of Back-end:

Api Images in Train:

Add train :


Home
Workspaces
API Network
Explore
Search Postman
Upgrade
My Workspace
New
Import
Overview
PUT http://localhost
POST http://localhos
No Environment
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
Angular
http://localhost:8881/trains
Save
POST
http://localhost:8881/trains
Send
Params
Auth
Headers (9)
Body
Pre-req.
Tests
Settings
Cookies
raw
JSON
Beautify
1 { "trainName": "chennai Express",
2 "from_to": "chennai to Andhra",
3 "day": "friday",
4 "time": "7:00:00AM",
5 "capacity": 6,
6 " charge": 200
7 }
Body
200 OK
59 ms
319 B
Save Response
Pretty
Raw
Preview
Visualize
Text
1 Hi chennai Express your train registeration successfully completed
Online
Find and Replace
Console
Cookies
Capture requests
Runner
Trash
Type here to search
29°C
ENG
17:38
04-01-2023

Delete train by Id :

Home Workspaces API Network Explore Search Postman Upgrade

You are offline. Your data can't be saved and may not be up to date. Until you're back online, you can work and save data locally by switching to your Scratch Pad from Settings.

My Workspace New Import

Overview PUT http://localhost DEL http://localhost No Environment

Collections APIs Environments Mock Servers Monitors

Angular

http://localhost:8881/trains/5 Save

DELETE http://localhost:8881/trains/5 Send

Params Auth Headers (7) Body Pre-req. Tests Settings Cookies

Body 200 OK 128 ms 269 B Save Response

Pretty Raw Preview Visualize Text

```
1  Train is deleted
```

Offline Find and Replace Console Cookies Capture requests Runner Trash

Type here to search 29°C ENG 17:56 04-01-2023

Get all train:

Home Workspaces API Network Explore Search Postman Upgrade

My Workspace New Import

Collections APIs Environments Mock Servers Monitors Flows History

Angular

Overview PUT http://localhost GET http://localhost No Environment

http://localhost:8881/findAll Save

GET http://localhost:8881/findAll Send

Params Auth Headers (9) Body Pre-req. Tests Settings Cookies

Body 200 OK 19 ms 922 B Save Response

Pretty Raw Preview Visualize JSON


```
    "trainId": 1,
    "trainName": "xyz express",
    "from_to": "chennai to madurai",
    "day": "sunday",
    "time": "5:00PM",
    "capacity": 4,
    "charge": 150.0
},
{
    "trainId": 4,
    "trainName": "zyx express",
    "from_to": "pune to chennai",
```


Online Find and Replace Console Cookies Capture requests Runner Trash

Type here to search 29°C ENG 17:44 04-01-2023

Get train by id:

Home Workspaces API Network Explore Search Postman Upgrade

You are offline. Your data can't be saved and may not be up to date. Until you're back online, you can work and save data locally by switching to your Scratch Pad from Settings.

My Workspace New Import

Overview PUT http://localhost GET http://localhost No Environment

Collections APIs Environments Mock Servers Monitors

Angular

http://localhost:8881/trains/1 Save

GET http://localhost:8881/trains/1 Send

Params Auth Headers (7) Body Pre-req. Tests Settings Cookies

Body 200 OK 24 ms 384 B Save Response

Pretty Raw Preview Visualize JSON


```
[
    {
        "trainId": 1,
        "trainName": "xyz express",
        "from_to": "chennai to madurai",
        "day": "sunday",
        "time": "5:00PM",
        "capacity": 4,
        "charge": 150.0
    }
]
```


Connecting... Find and Replace Console Cookies Capture requests Runner Trash

Type here to search 29°C ENG 17:54 04-01-2023

Update train:

Home Workspaces API Network Explore Search Postman Upgrade

My Workspace New Import

Overview PUT http://localhost PUT http://localhost No Environment

Collections APIs Environments Mock Servers Monitors Flows History

Angular

http://localhost:8881/Trains/6 Save

PUT http://localhost:8881/Trains/6 Send

Params Auth Headers (9) Body Pre-req. Tests Settings Cookies

raw JSON Beautify

```
1  {
2      "trainId": 6,
3      "trainName": "abcundo xpress",
4      "from_to": "madhurai_to_thirunelveli",
5      "day": "saturday",
6      "time": "01:00:00 am",
7      "capacity": 150,
8      "charge": 100.0
   }
```

Body 200 OK 119 ms 280 B Save Response

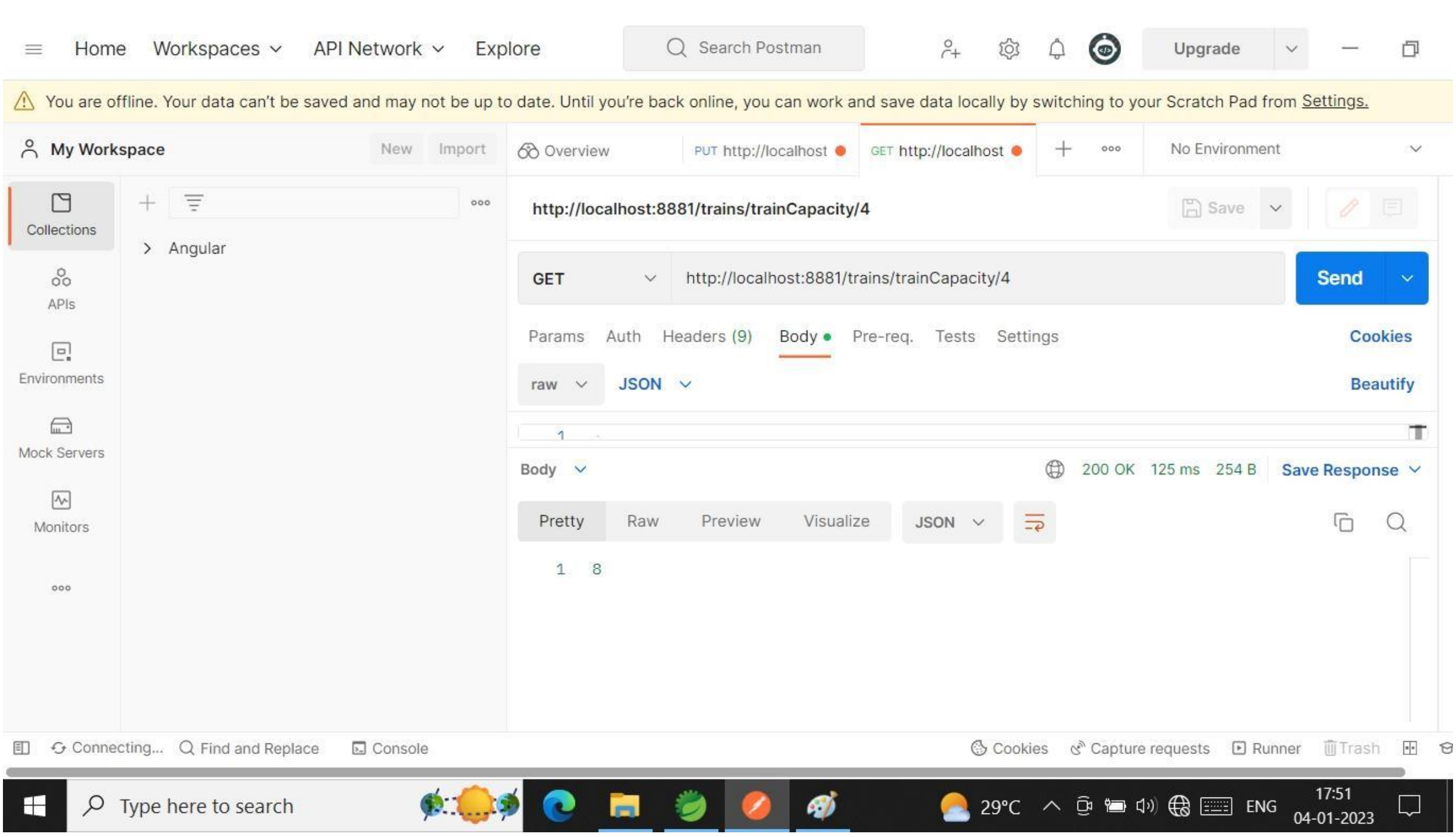
Pretty Raw Preview Visualize Text

```
1  abcundo xpress TrainUpdated
```

Online Find and Replace Console Cookies Capture requests Runner Trash

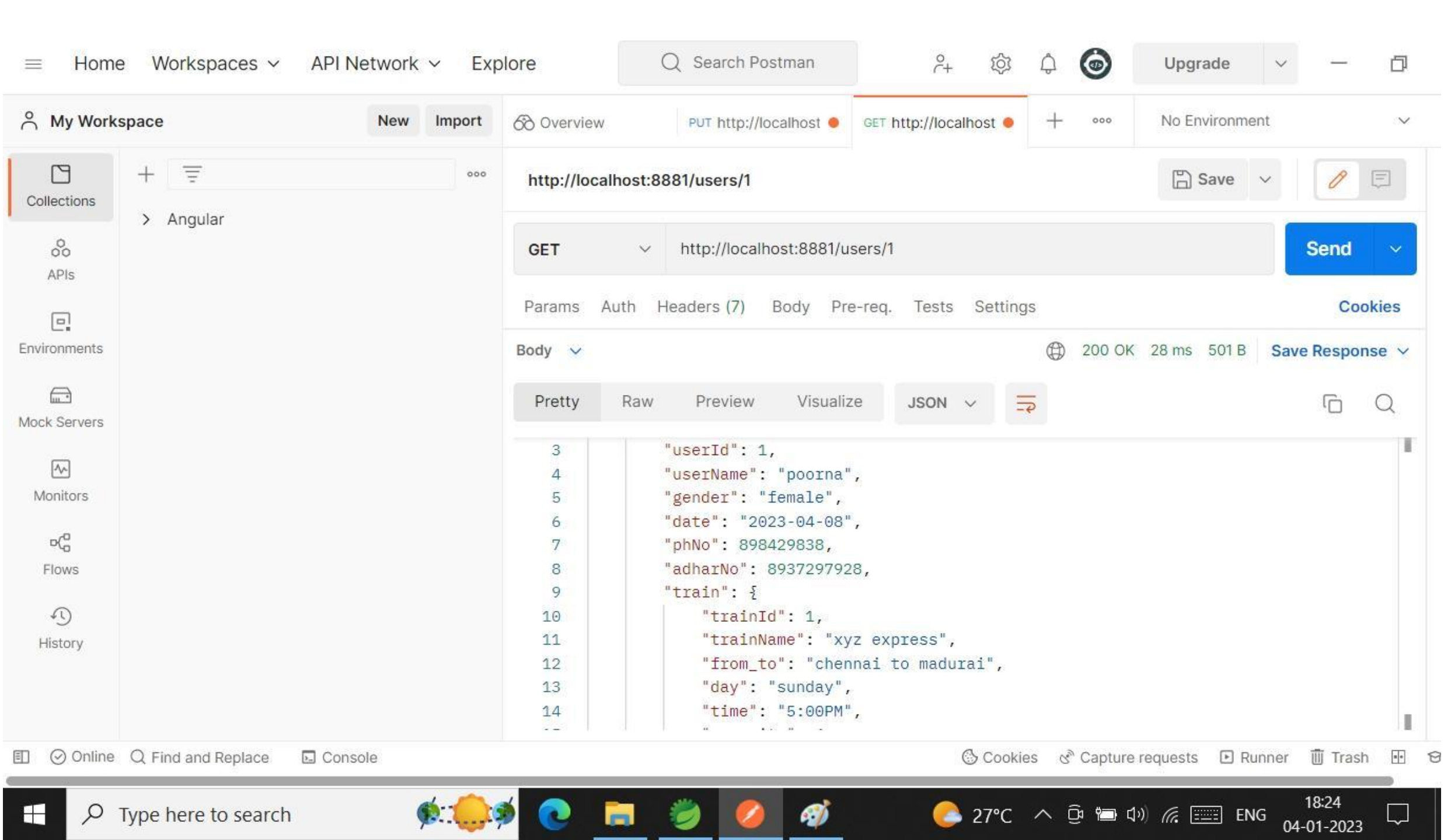
Type here to search 29°C ENG 17:48 04-01-2023

Train capacity by Id:

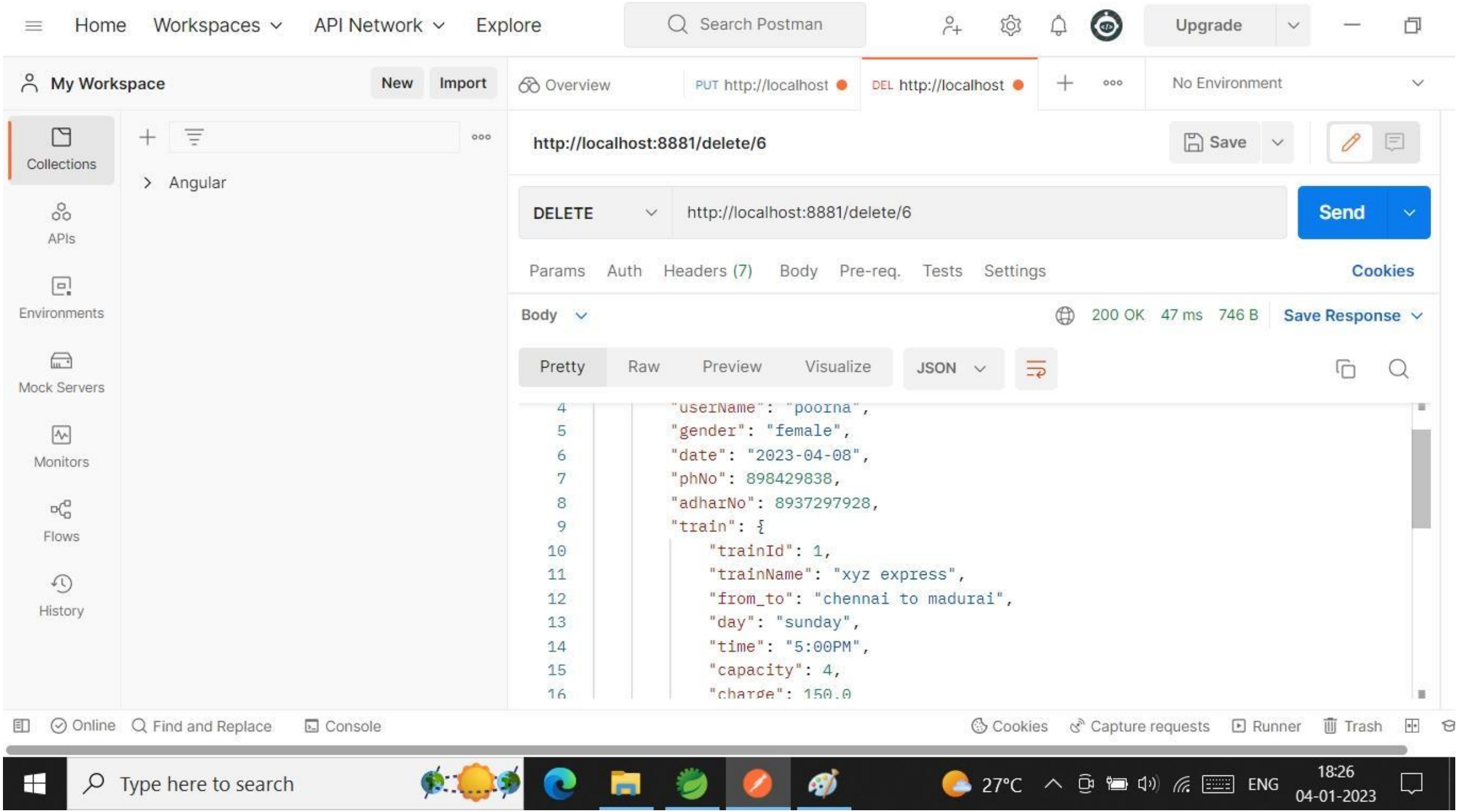

Home
Workspaces
API Network
Explore
Search Postman
Upgrade
You are offline. Your data can't be saved and may not be up to date. Until you're back online, you can work and save data locally by switching to your Scratch Pad from Settings.
My Workspace
New
Import
Overview
PUT http://localhost
GET http://localhost
No Environment
Collections
APIs
Environments
Mock Servers
Monitors
Angular
http://localhost:8881/trains/trainCapacity/4
Save
GET
http://localhost:8881/trains/trainCapacity/4
Send
Params
Auth
Headers (9)
Body
Pre-req.
Tests
Settings
Cookies
raw
JSON
Beautify
1
Body
200 OK
125 ms
254 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
1 8
Connecting...
Find and Replace
Console
Cookies
Capture requests
Runner
Trash
Type here to search
29°C
ENG
17:51
04-01-2023

Api Images in User:

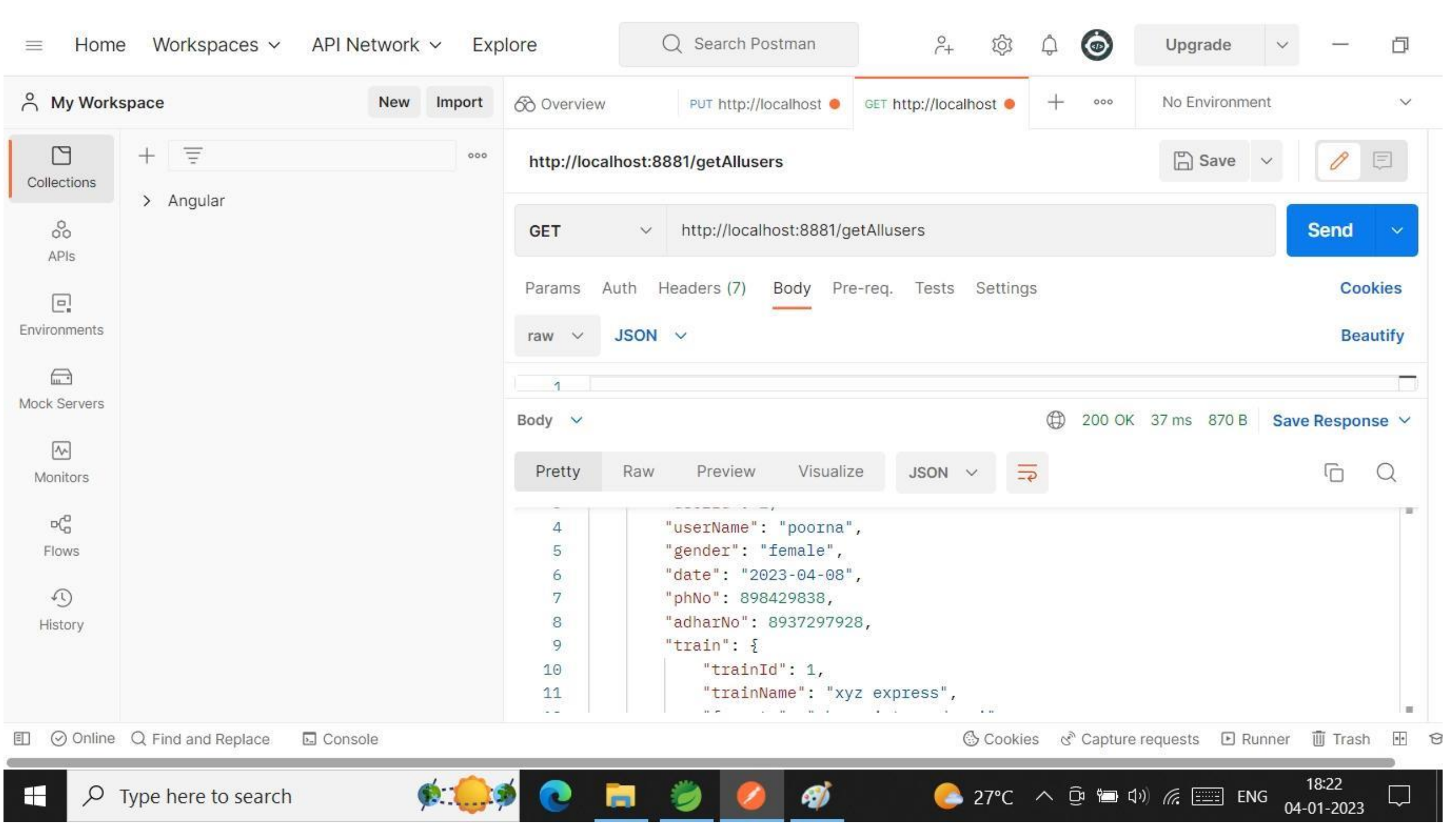
Get user by id:


Home
Workspaces
API Network
Explore
Search Postman
Upgrade
My Workspace
New
Import
Overview
PUT http://localhost
GET http://localhost
No Environment
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
Angular
http://localhost:8881/users/1
Save
GET
http://localhost:8881/users/1
Send
Params
Auth
Headers (7)
Body
Pre-req.
Tests
Settings
Cookies
Body
200 OK
28 ms
501 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
"userId": 1,
"userName": "poorna",
"gender": "female",
"date": "2023-04-08",
"phNo": 898429838,
"adharNo": 8937297928,
"train": {
"trainId": 1,
"trainName": "xyz express",
"from_to": "chennai to madurai",
"day": "sunday",
"time": "5:00PM",
Online
Find and Replace
Console
Cookies
Capture requests
Runner
Trash
Type here to search
27°C
ENG
18:24
04-01-2023

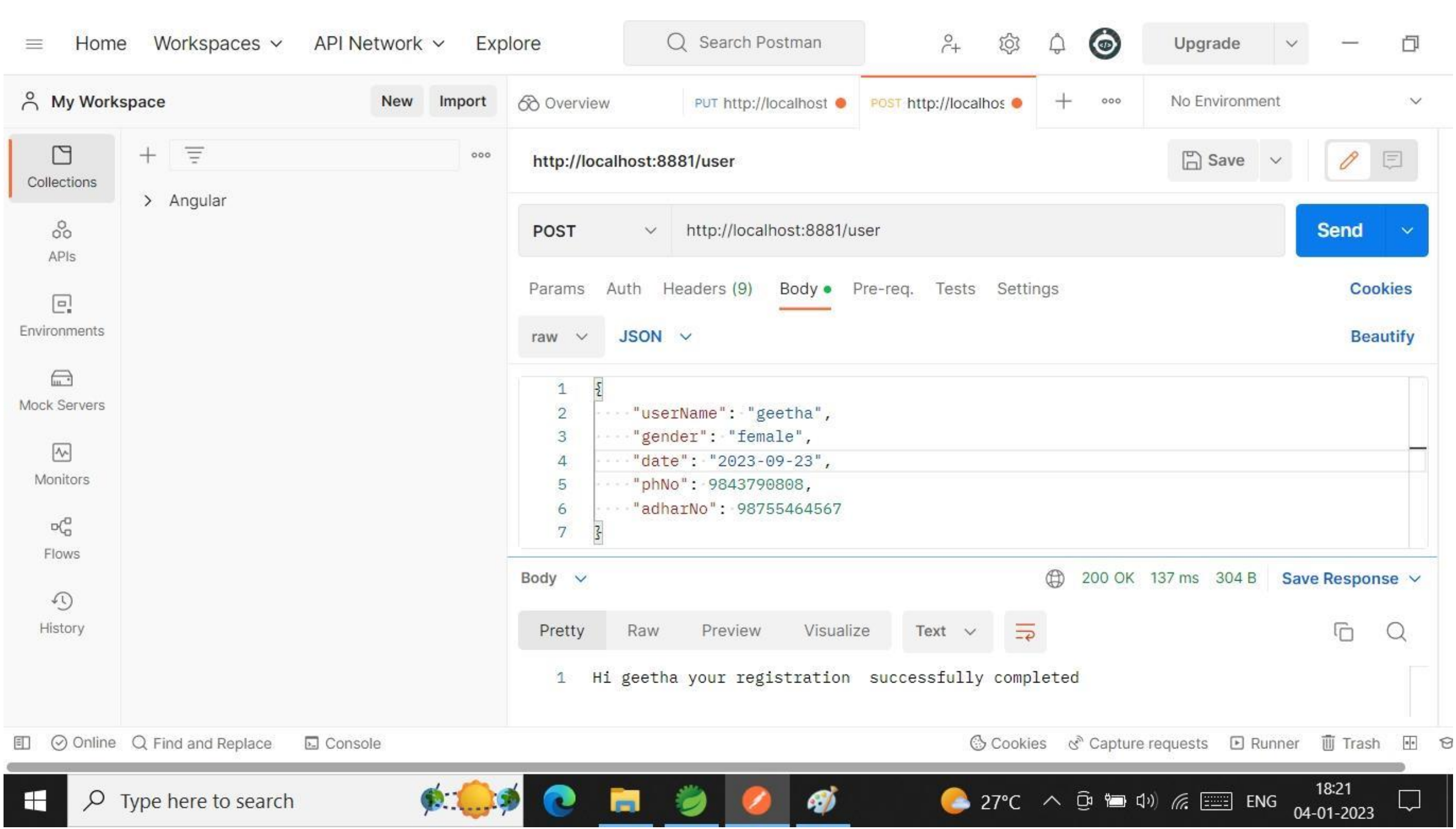
Delete user by id:


Home
Workspaces
API Network
Explore
Search Postman
Upgrade
My Workspace
New
Import
Overview
PUT http://localhost
DEL http://localhost
No Environment
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
Angular
http://localhost:8881/delete/6
Save
DELETE
http://localhost:8881/delete/6
Send
Params
Auth
Headers (7)
Body
Pre-req.
Tests
Settings
Cookies
Body
200 OK
47 ms
746 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
"userName": "poorna",
"gender": "female",
"date": "2023-04-08",
"phNo": 898429838,
"adharNo": 8937297928,
"train": {
"trainId": 1,
"trainName": "xyz express",
"from_to": "chennai to madurai",
"day": "sunday",
"time": "5:00PM",
"capacity": 4,
"charge": 150.0
Online
Find and Replace
Console
Cookies
Capture requests
Runner
Trash
Type here to search
27°C
ENG
18:26
04-01-2023

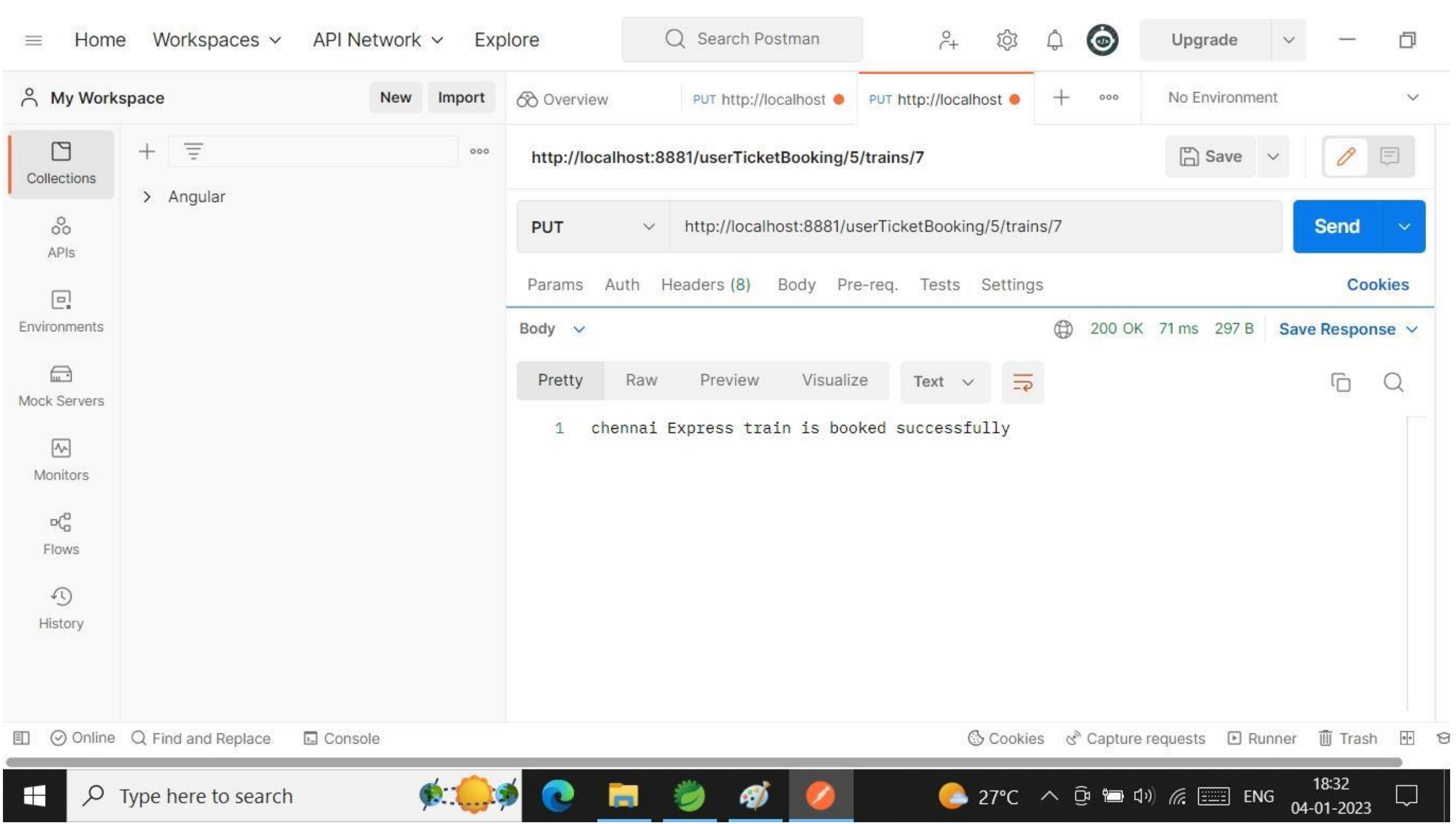
Get all user:


Home
Workspaces
API Network
Explore
Search Postman
Upgrade
My Workspace
New
Import
Overview
PUT http://localhost
GET http://localhost
No Environment
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
Angular
http://localhost:8881/getAllusers
Save
GET
http://localhost:8881/getAllusers
Send
Params
Auth
Headers (7)
Body
Pre-req.
Tests
Settings
Cookies
raw
JSON
Beautify
Body
200 OK
37 ms
870 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
"userName": "poorna",
"gender": "female",
"date": "2023-04-08",
"phNo": 898429838,
"adharNo": 8937297928,
"train": {
"trainId": 1,
"trainName": "xyz express",
Online
Find and Replace
Console
Cookies
Capture requests
Runner
Trash
Type here to search
27°C
ENG
18:22
04-01-2023

User:


Home
Workspaces
API Network
Explore
Search Postman
Upgrade
My Workspace
New
Import
Overview
PUT http://localhost
POST http://localhos
No Environment
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
Angular
http://localhost:8881/user
Save
POST
http://localhost:8881/user
Send
Params
Auth
Headers (9)
Body
Pre-req.
Tests
Settings
Cookies
raw
JSON
Beautify
1 {
2 "userName": "geetha",
3 "gender": "female",
4 "date": "2023-09-23",
5 "phNo": 9843790808,
6 "adharNo": 98755464567
7 }
Body
200 OK
137 ms
304 B
Save Response
Pretty
Raw
Preview
Visualize
Text
1 Hi geetha your registration successfully completed
Online
Find and Replace
Console
Cookies
Capture requests
Runner
Trash
Type here to search
27°C
ENG
18:21
04-01-2023

Ticket booking by user id and train id:

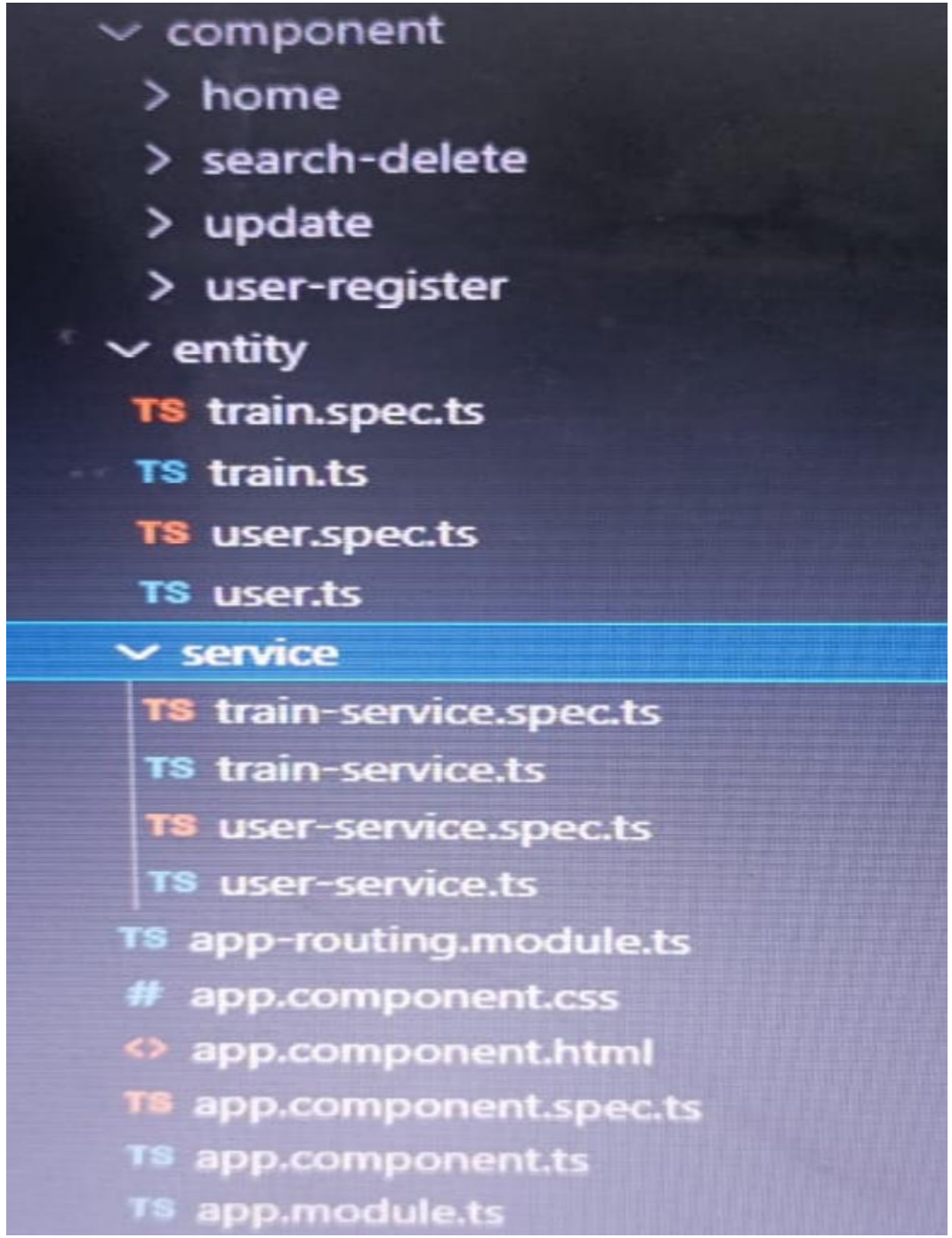

Home
Workspaces
API Network
Explore
Search Postman
Upgrade
My Workspace
New
Import
Overview
PUT http://localhost
PUT http://localhost
No Environment
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
Angular
http://localhost:8881/userTicketBooking/5/trains/7
Save
PUT
http://localhost:8881/userTicketBooking/5/trains/7
Send
Params
Auth
Headers (8)
Body
Pre-req.
Tests
Settings
Cookies
Body
200 OK
71 ms
297 B
Save Response
Pretty
Raw
Preview
Visualize
Text
1 chennai Express train is booked successfully
Online
Find and Replace
Console
Cookies
Capture requests
Runner
Trash
Type here to search
27°C
ENG
18:32
04-01-2023

FRONT-END:

- Angular
- Bootstrap
- CSS
- Typescript

About Front-End:

- The front end implemented by 4 components, 2 entity, 2 service.


component
home
search-delete
update
user-register
entity
train.spec.ts
train.ts
user.spec.ts
user.ts
service
train-service.spec.ts
train-service.ts
user-service.spec.ts
user-service.ts
app-routing.module.ts
app.component.css
app.component.html
app.component.spec.ts
app.component.ts
app.module.ts

Working Of Front-End :

- Railway Reservation system which implements the MVC(Model-View-Control)Architecture. As View, Angular-8 with routing is used . For the backend, Spring Boot v2.1.6 is used.

- Java persistence Api (JPA) is used to write the Business Logic. REST Api’s are Written to communicate between server ports (Angular->port:4200, Spring Boot ->port:8080, MySQL -> port:3306)

The project consist of a Login and Register page when user navigate to localhost:4200. On successful login or registration as a

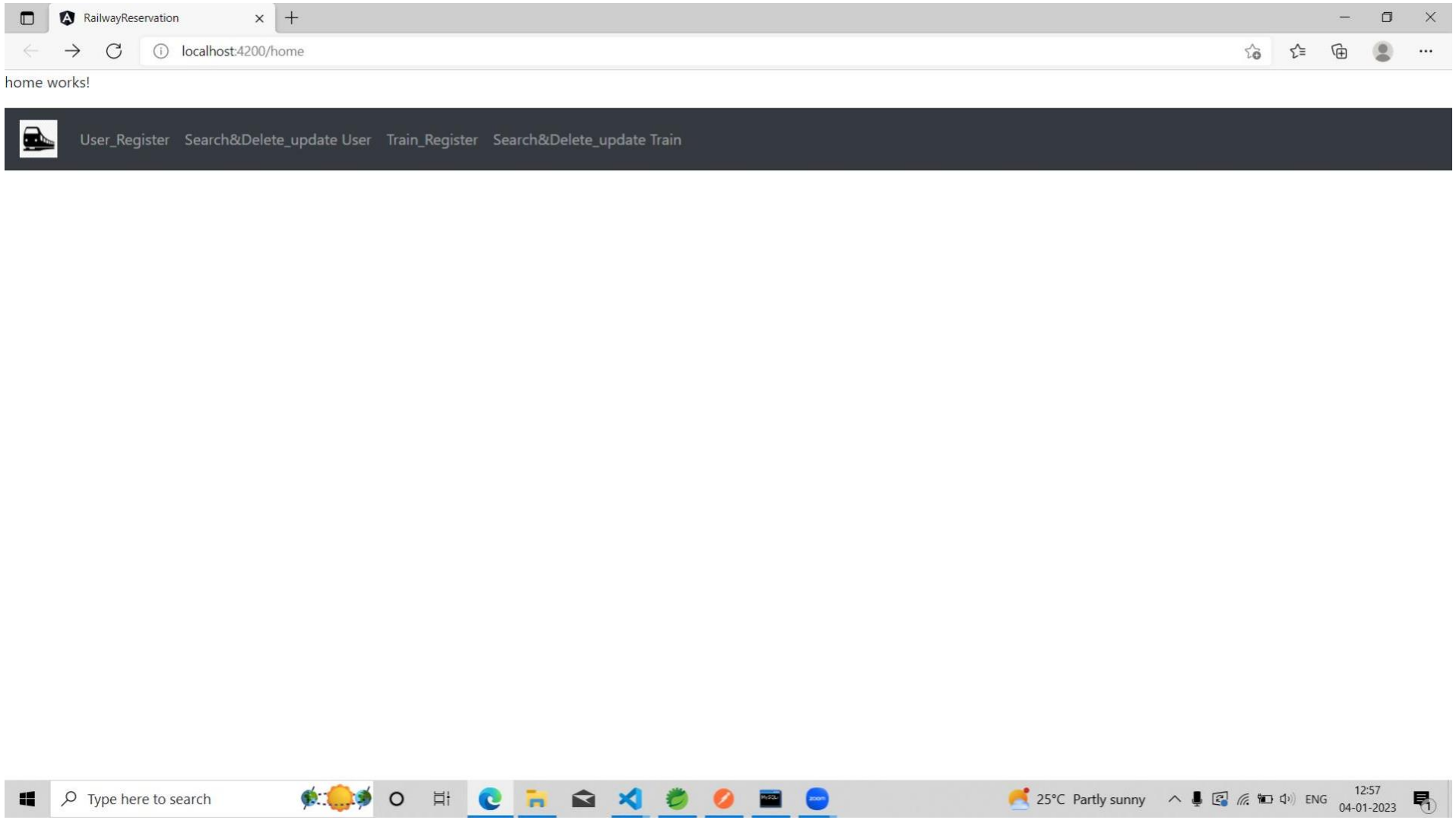
user, a welcome page all the user detail and option to booking your ticket.

Advantages Of Front-End:

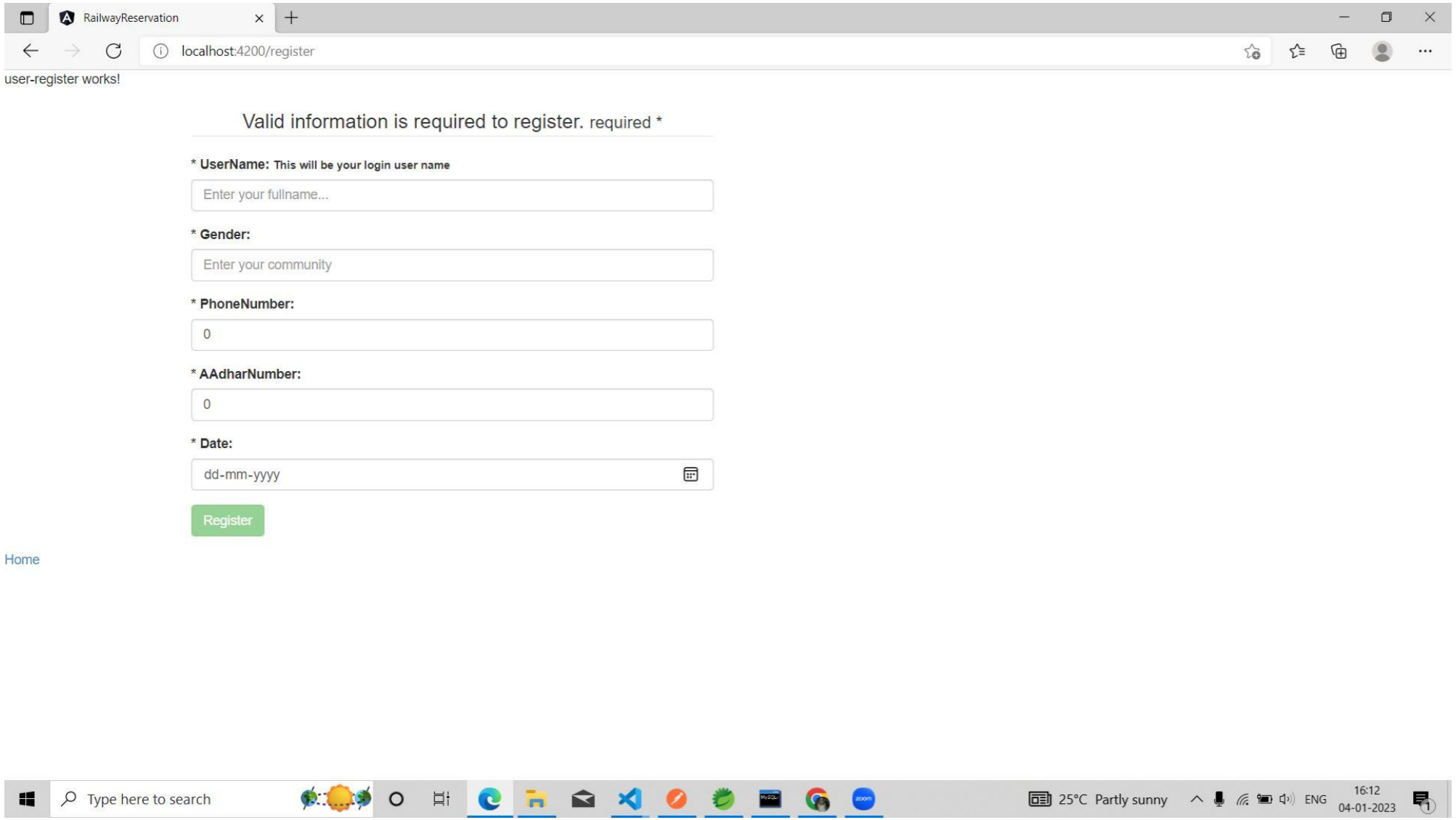
- High quality application.
- Improved speed and performance.
- Faster development process.
- Effective cross-platform development

Snapshots of Front-end:

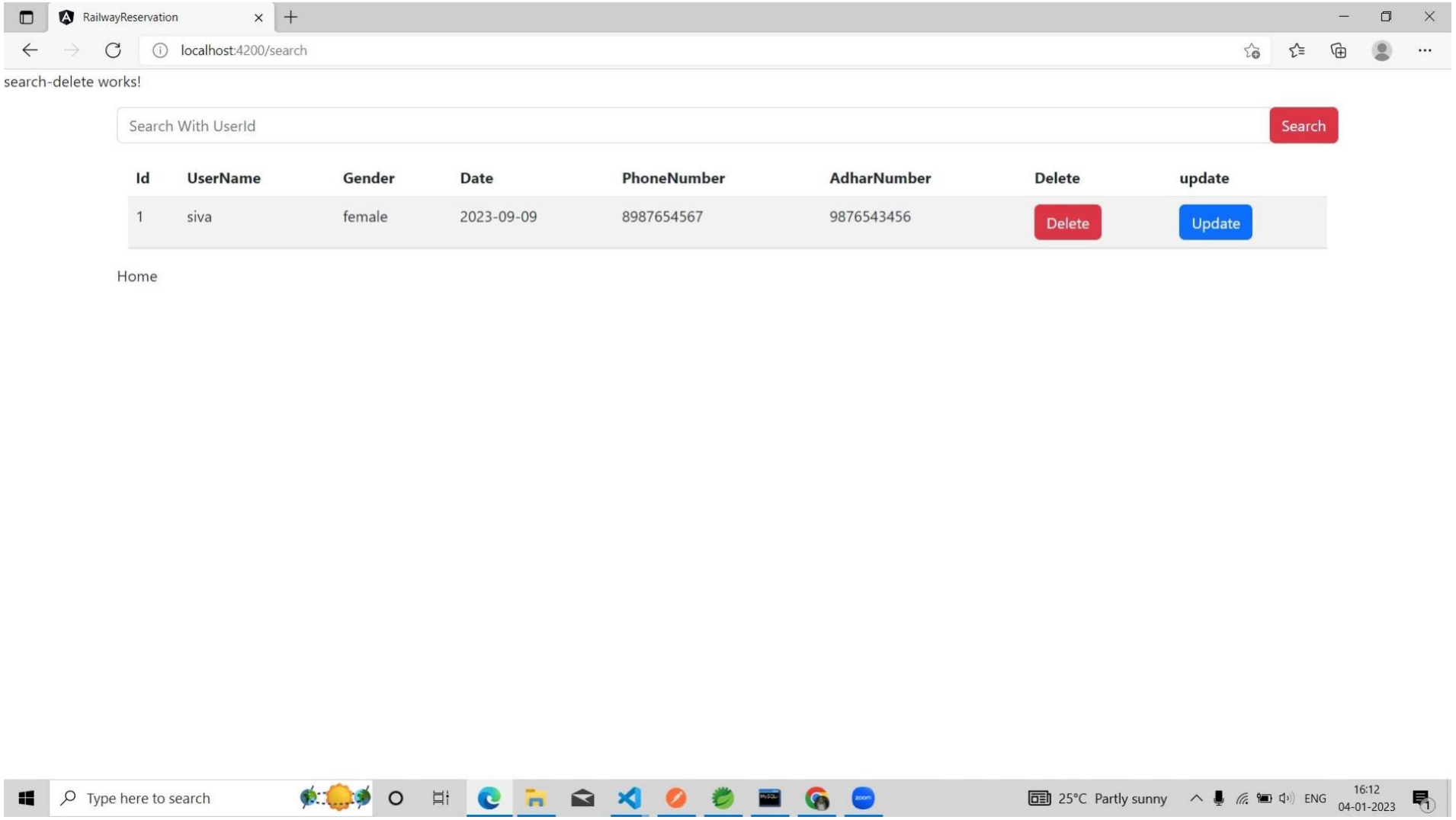
Home page:



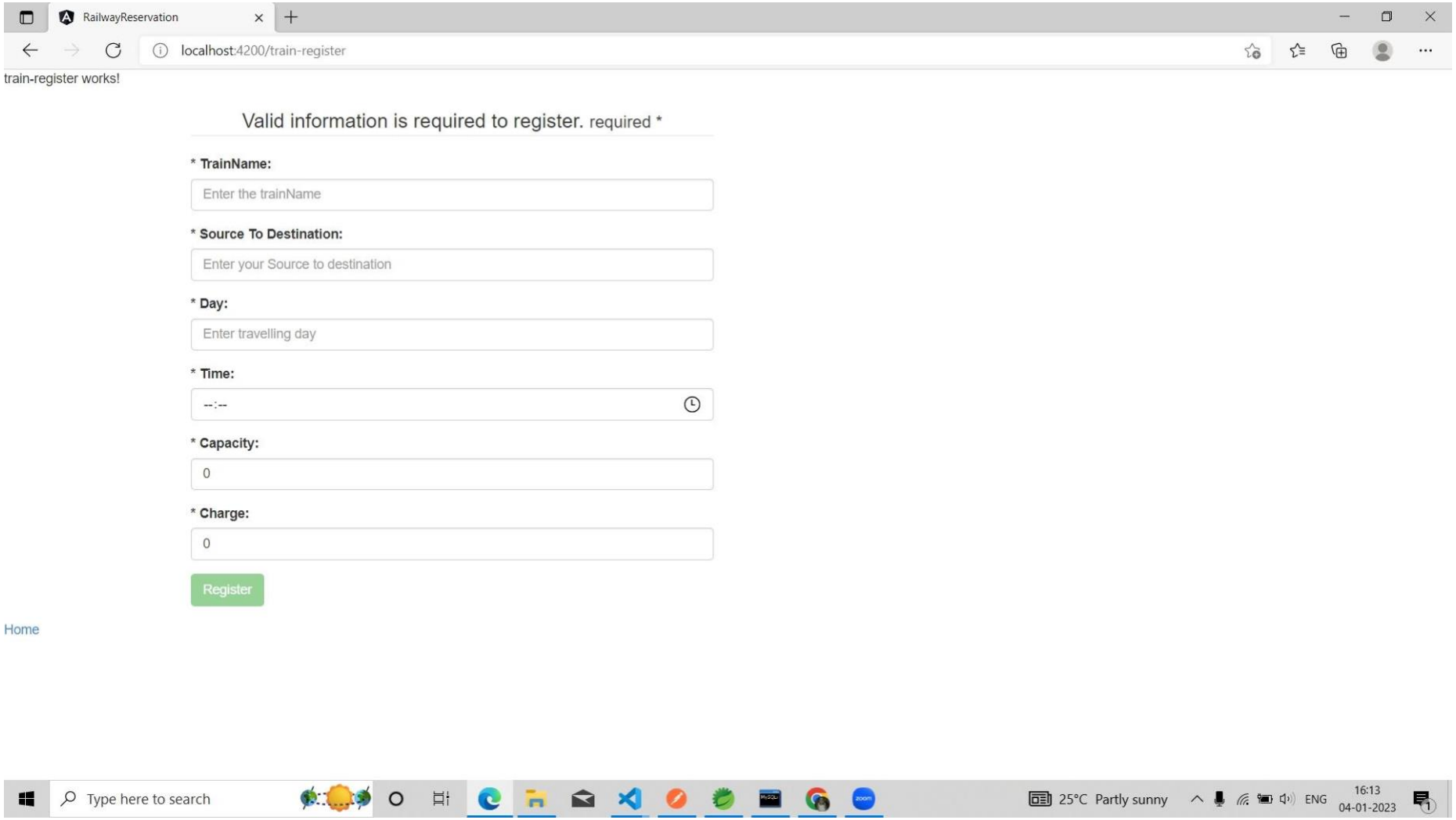
User register page:



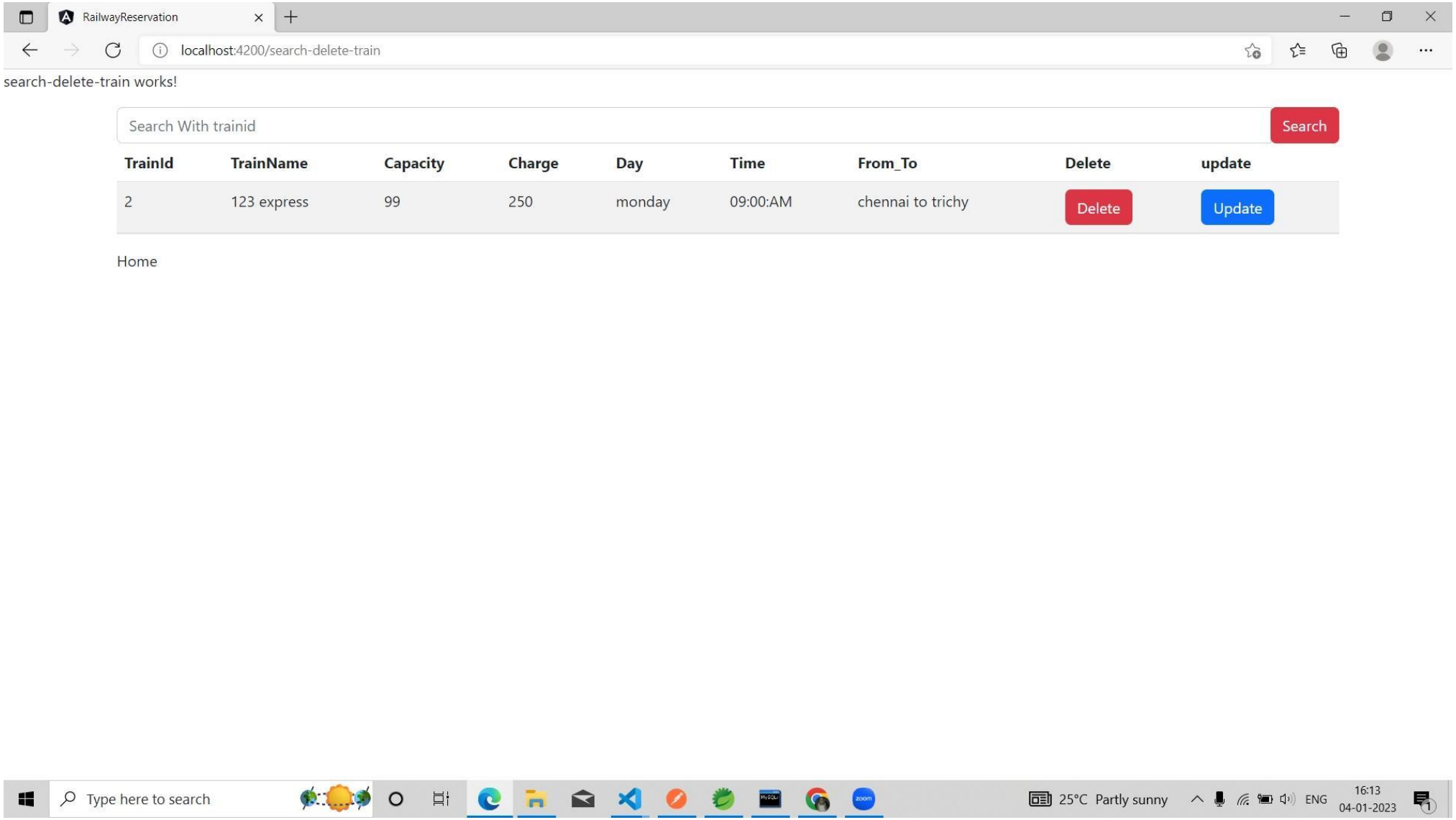
User search-delete_ update page:



Train register page:



Train search- delete_ update:


RailwayReservation
localhost:4200/search-delete-train
search-delete-train works!
Search With trainid
Search
TrainId
TrainName
Capacity
Charge
Day
Time
From_To
Delete
update
2
123 express
99
250
monday
09:00:AM
chennai to trichy
Delete
Update
Home
Type here to search
25°C Partly sunny
ENG
16:13
04-01-2023

Conclusion :

- In our project Railway reservation system we have stored all the information about the Trains scheduled and the users booking tickets and even status of trains, seats etc.

- This data base is helpful for the applications which facilitate passengers to book the train tickets and check the details of trains and their status from their place itself it avoids inconveniences of going to railway station for each and every query they get.

- We had considered the most important requirements only, many more features and details can be added to our project in order to obtain even more user friendly applications.

- These applications are already in progress and in future they can be upgraded and may become part of amazing technology.