MYCAB (TAXI BOOKING SYSTEM)

1. Background

In today’s traffic world, taxi plays a major role when a person wants to move from one place to another. Here in Rwanda transportation of people through taxi(cab) it’s not easy to get one when you need it so that’s why mycab will help you when you need it you have to book it only.

In my taxi Booking System(mycab), people can easily book a taxi. Users have to pick-up point & destination, date & time. Users will get a list of All the bookings date & time in order to know when my car is available.

3 functional and 2 non functional requirement

Functional requirement

View private booking

Book for private cab

manage booking (view booking and cancel booking)

book mycab (normal booking)

Non functional

accessibility: mycab web app can be accessible at any device pc, phones and tablets

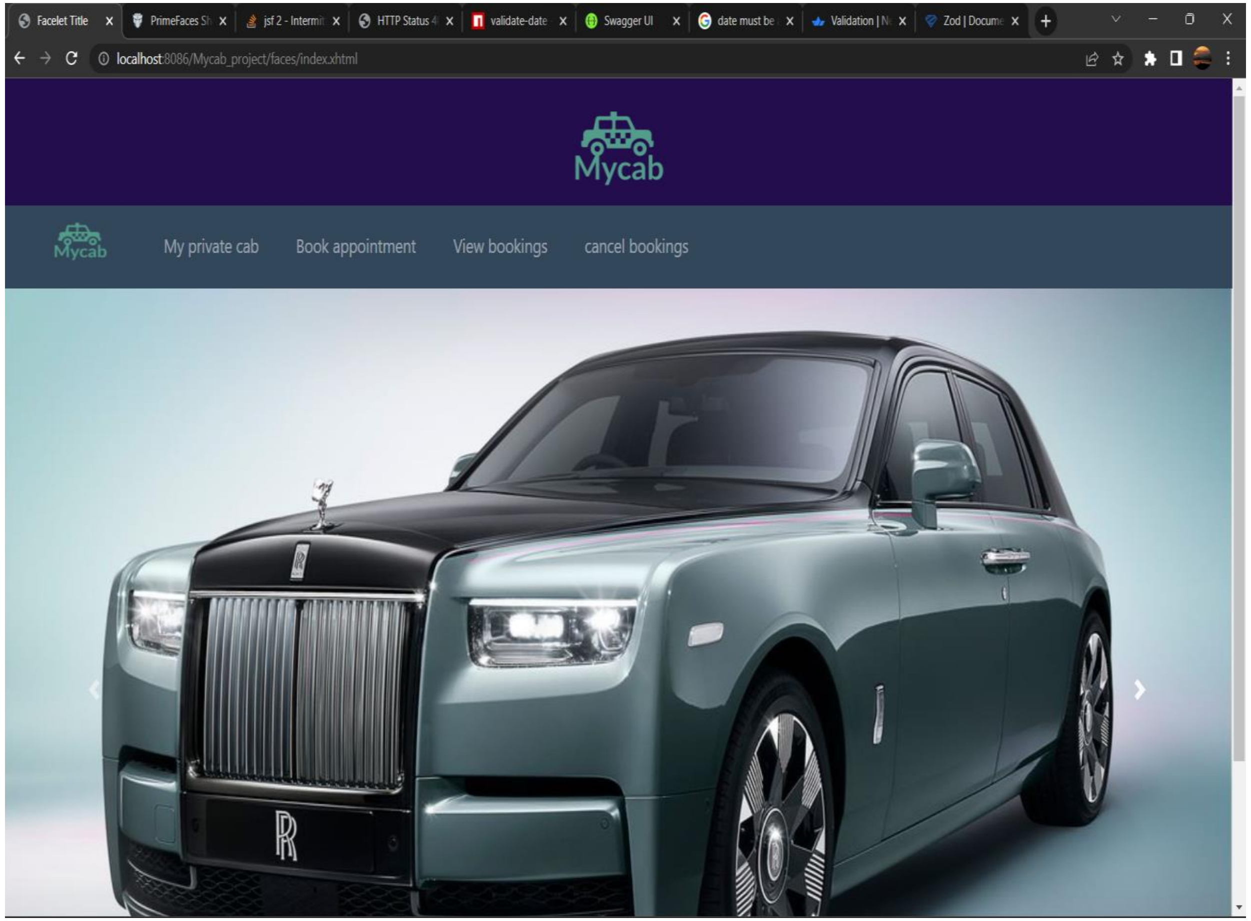
performance: fast response time no request take longer than 5 seconds

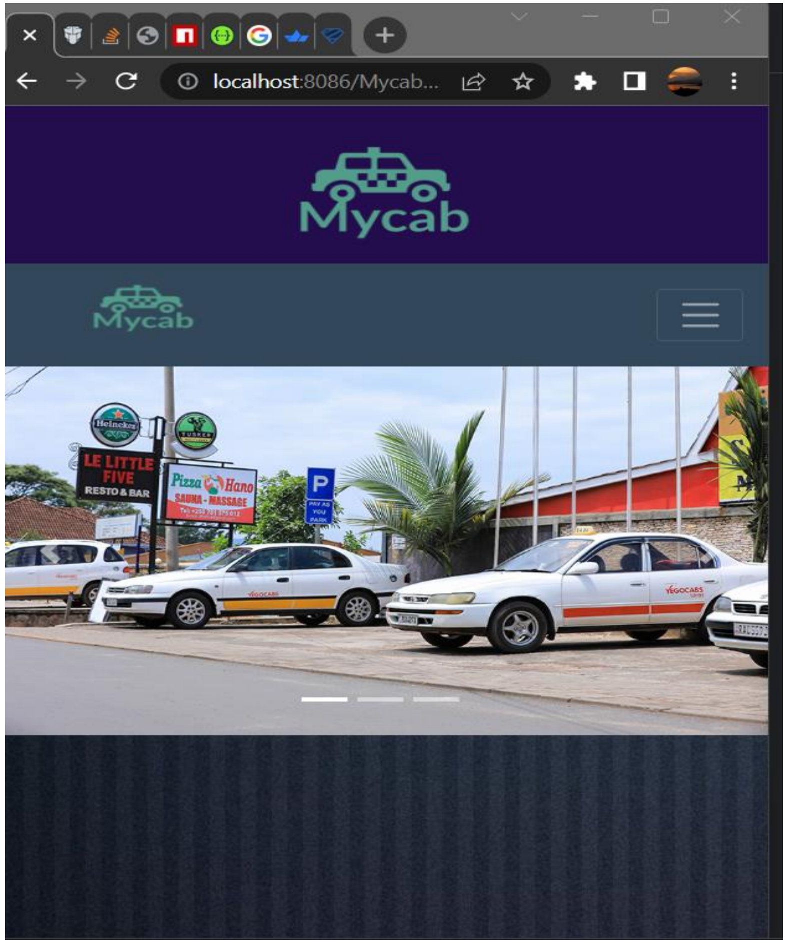
user interface requirement

intuitive: nice and easy to use

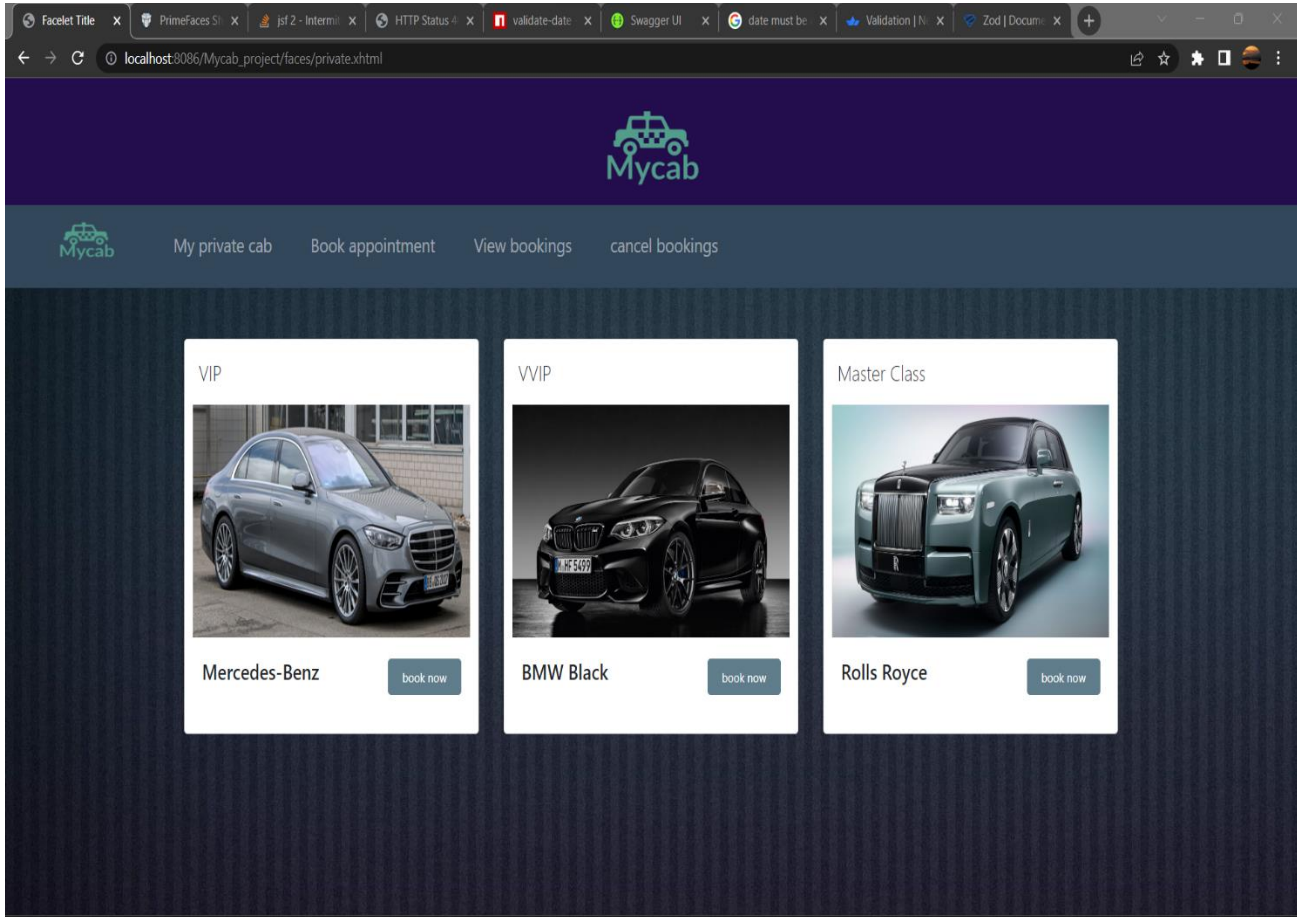
responsive its look good on different device by using(bootstrap)

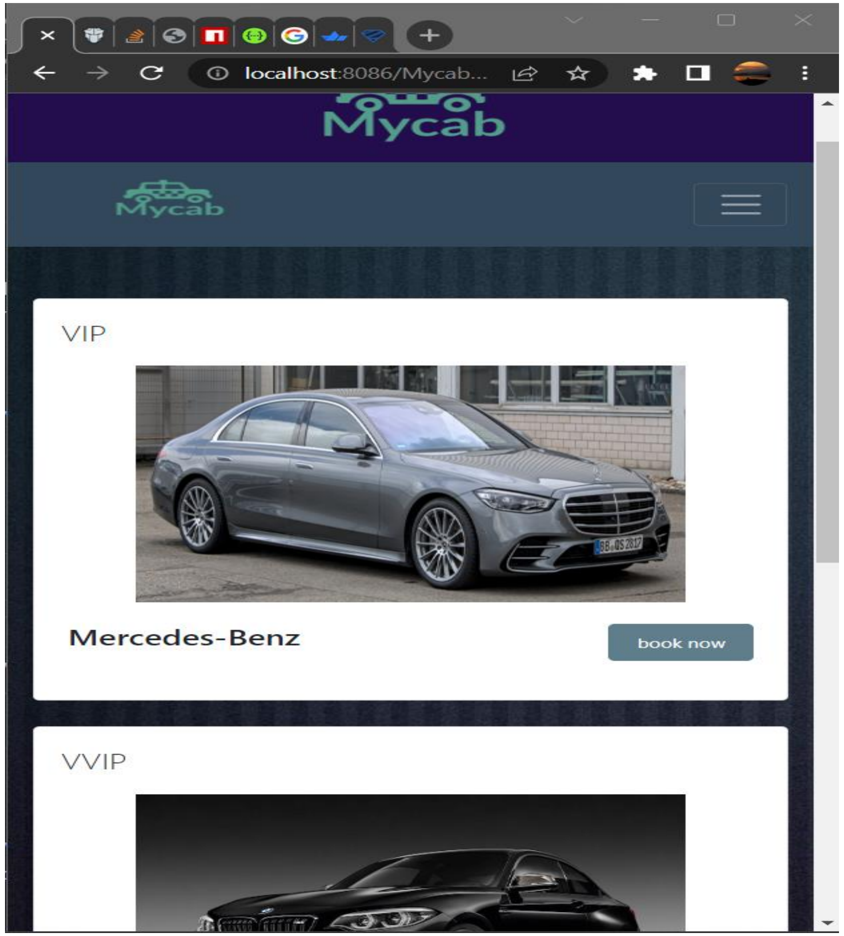
Mycab is responsive everywhere it look good on phone,laptop or on any other device



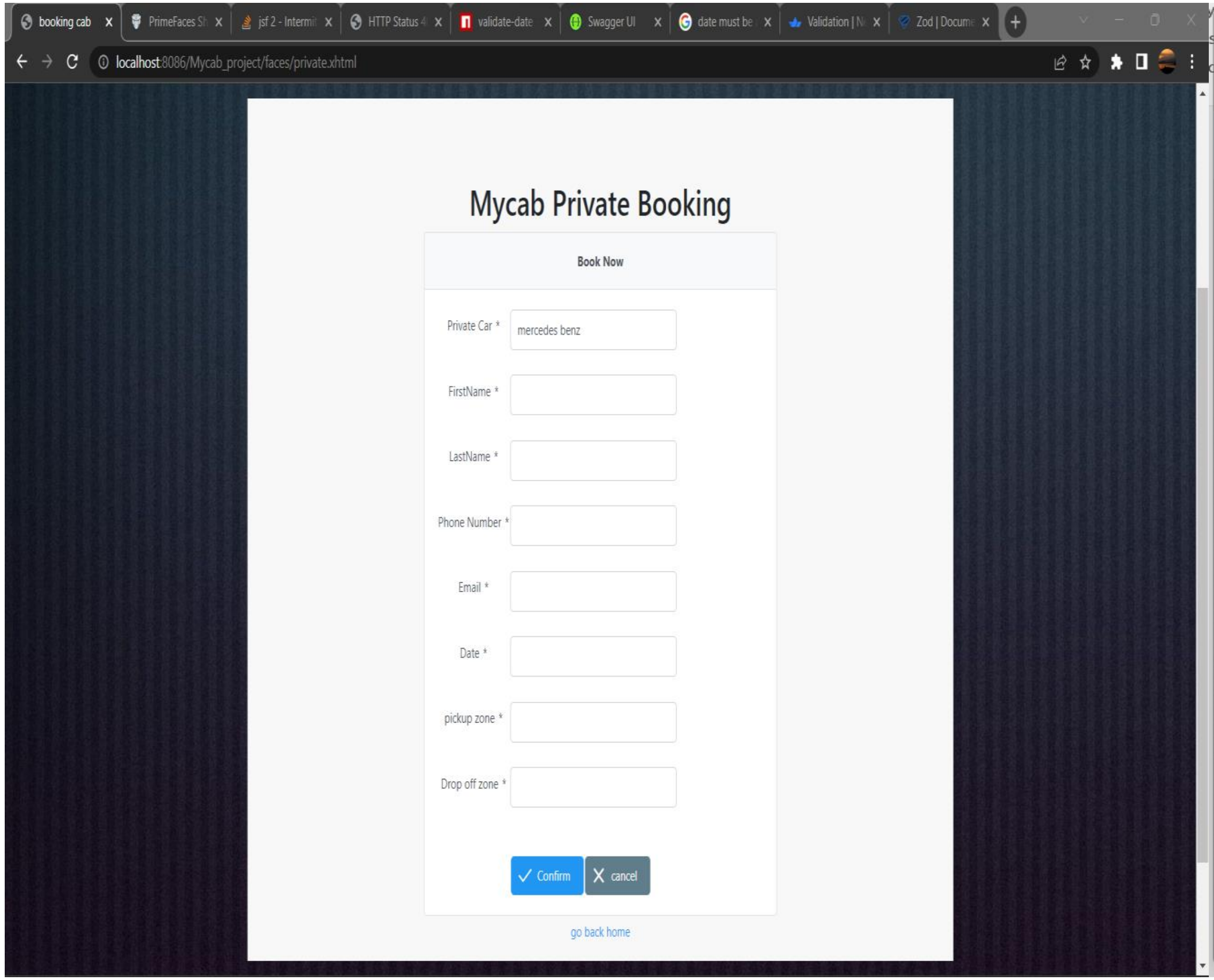


You can view and choose which private cab you want and book it





After choosing which private cab you want you will see private booking form with your car you choose on private car field which is readable only you can’t edit it



localhost:8086/Mycab_project/faces/private.xhtml

Mycab Private Booking

Book Now

Private Car * mercedes benz

FirstName *

LastName *

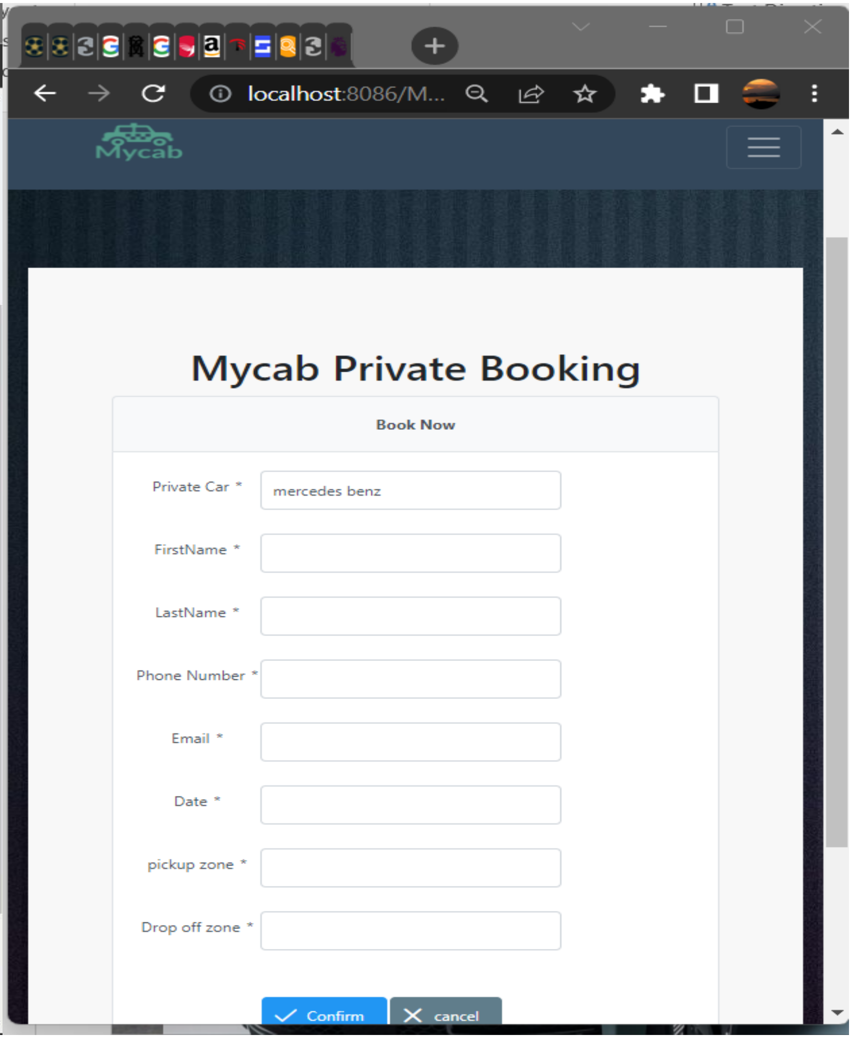
Phone Number *

Email *

Date *

pickup zone *

Drop off zone *



localhost:8086/M...

Mycab Private Booking

Book Now

Private Car * mercedes benz

FirstName *

LastName *

Phone Number *

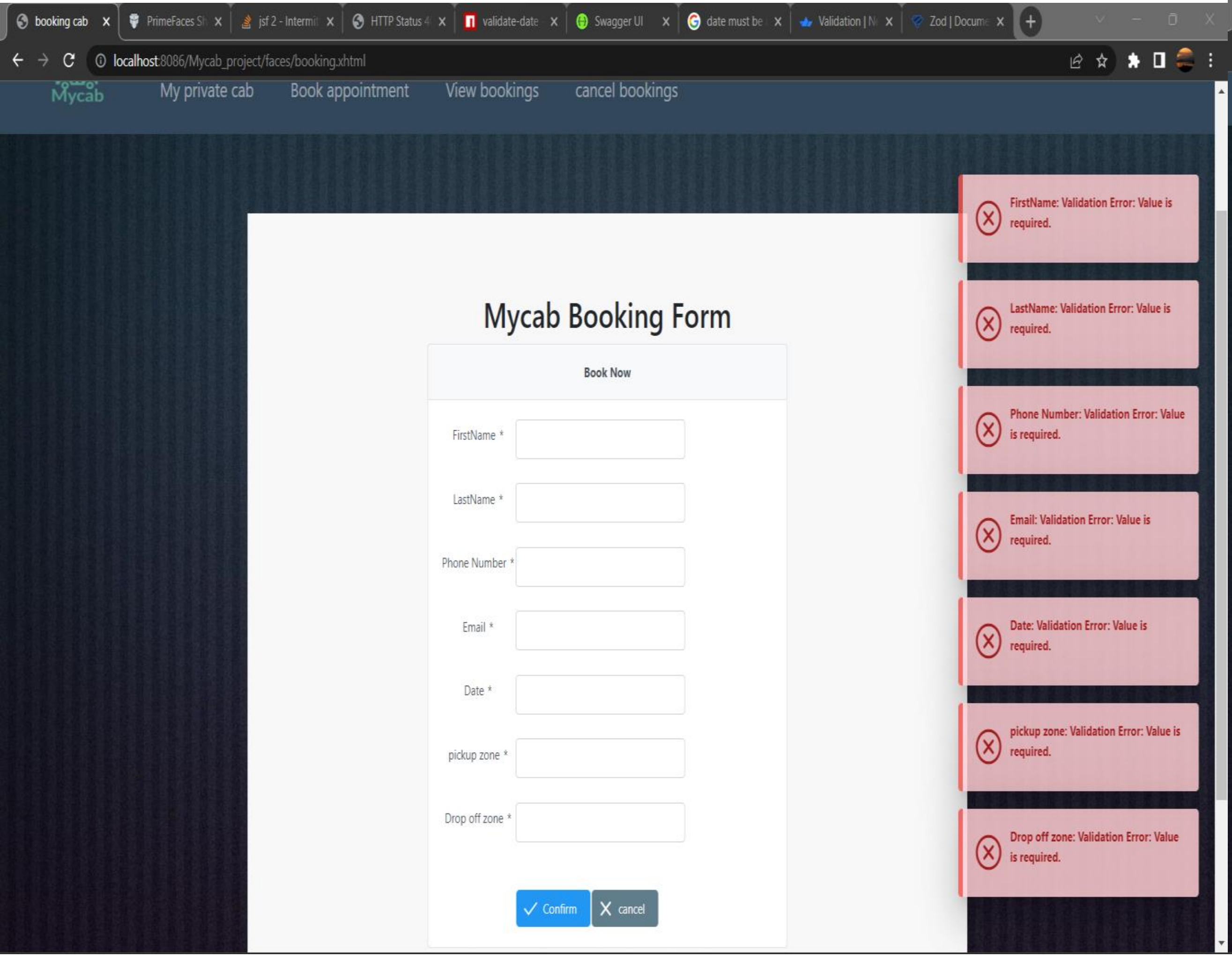
Email *

Date *

pickup zone *

Drop off zone *

The form has validation on it before you submit you have to meet with our validation rules example: you cant book in past date or phone number must be Rwandan, etc



Mycab Booking Form

Book Now

FirstName *	
LastName *	
Phone Number *	
Email *	
Date *	
pickup zone *	
Drop off zone *	

FirstName: Validation Error: Value is required.

LastName: Validation Error: Value is required.

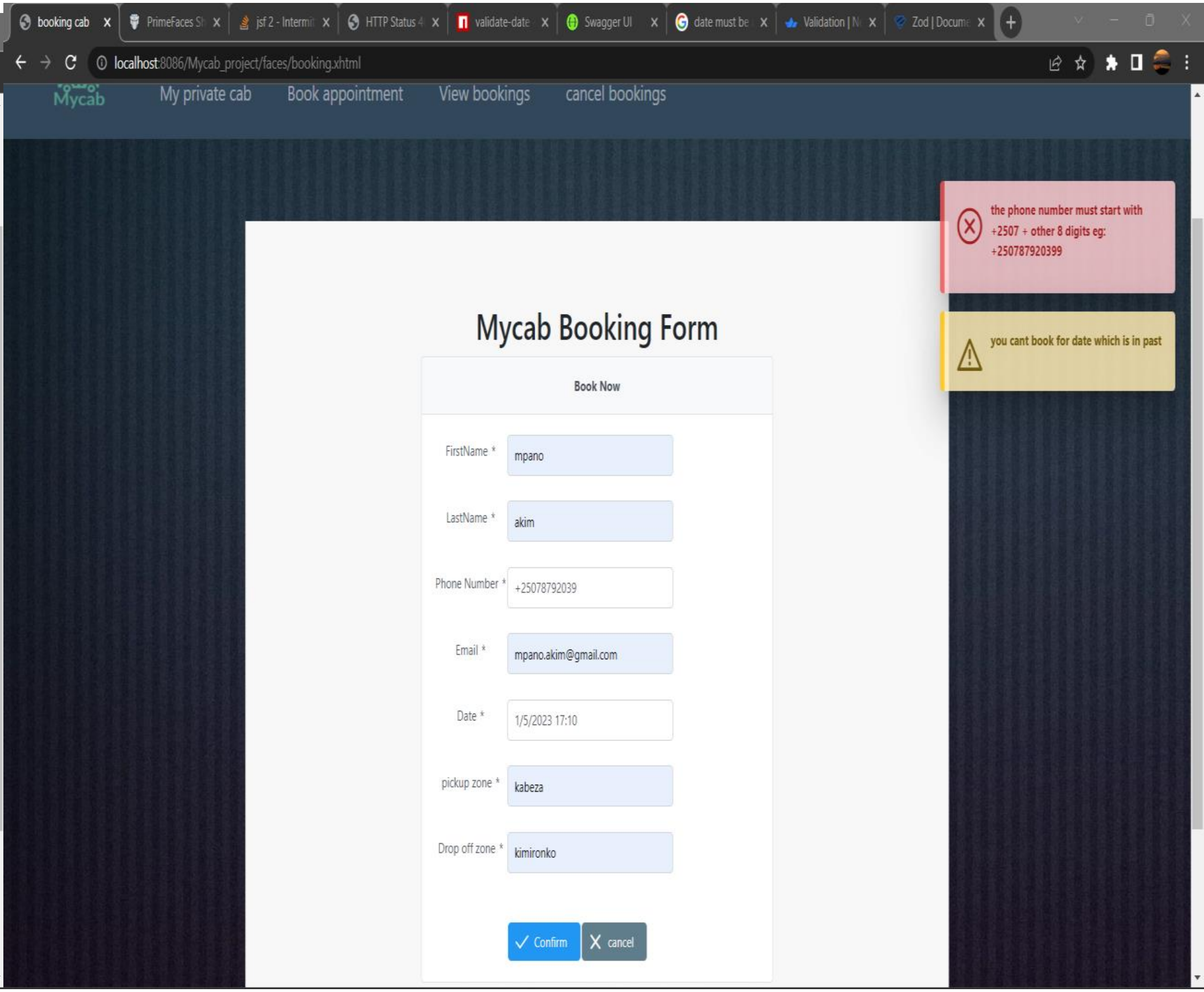
Phone Number: Validation Error: Value is required.

Email: Validation Error: Value is required.

Date: Validation Error: Value is required.

pickup zone: Validation Error: Value is required.

Drop off zone: Validation Error: Value is required.



Mycab Booking Form

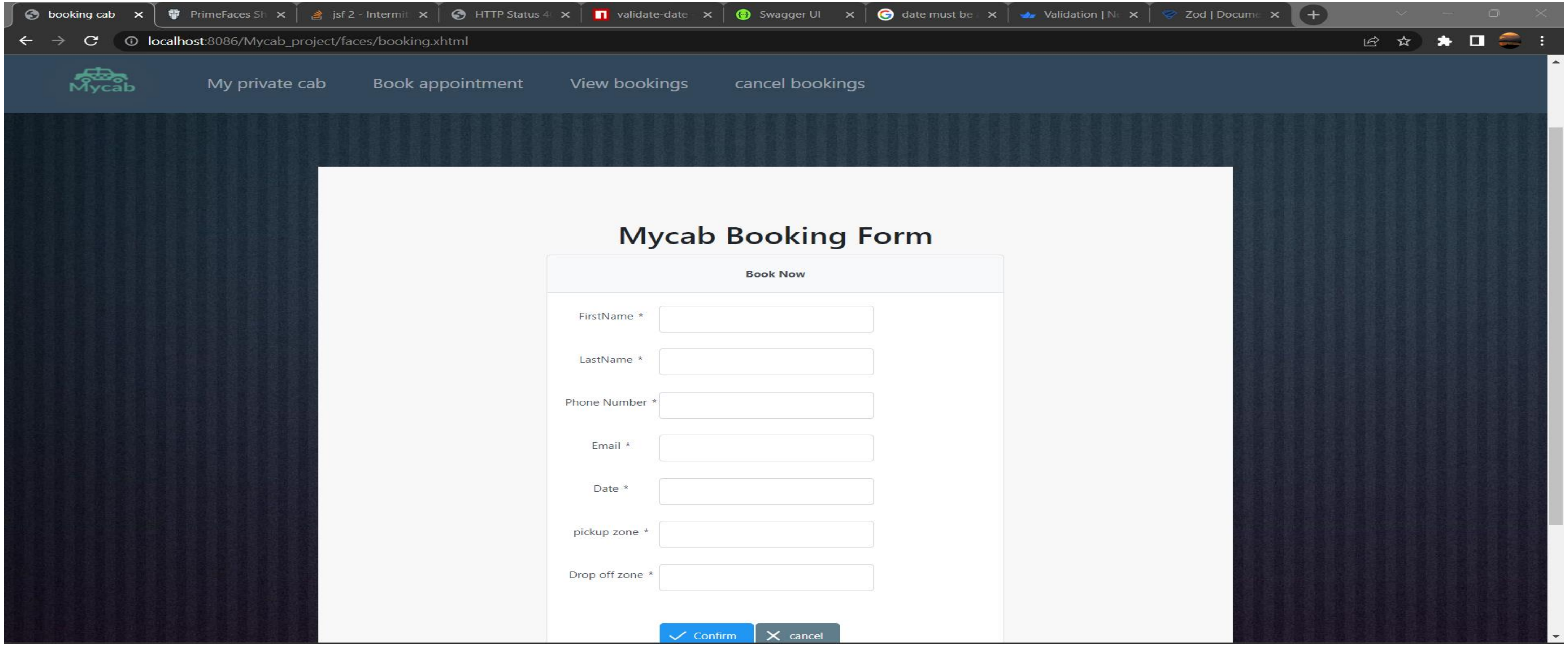
Book Now

FirstName *	mpano
LastName *	akim
Phone Number *	+25078792039
Email *	mpano.akim@gmail.com
Date *	1/5/2023 17:10
pickup zone *	kabeza
Drop off zone *	kimironko

the phone number must start with +2507 + other 8 digits eg: +250787920399

you cant book for date which is in past

You can also book normal booking with normal cab

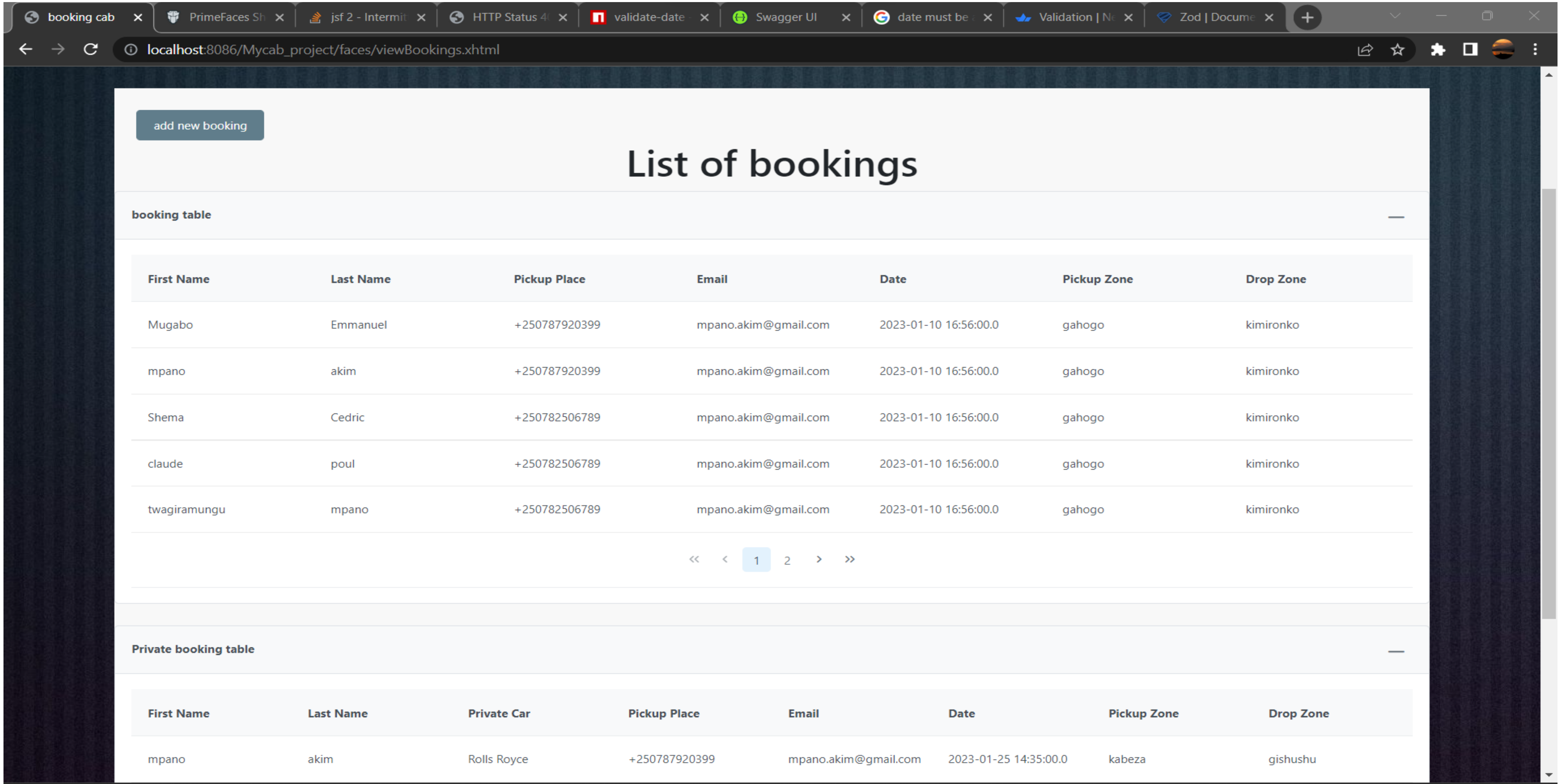


Mycab Booking Form

Book Now

FirstName *	
LastName *	
Phone Number *	
Email *	
Date *	
pickup zone *	
Drop off zone *	

You can also view the list of booking which are normal and private booking tables



List of bookings

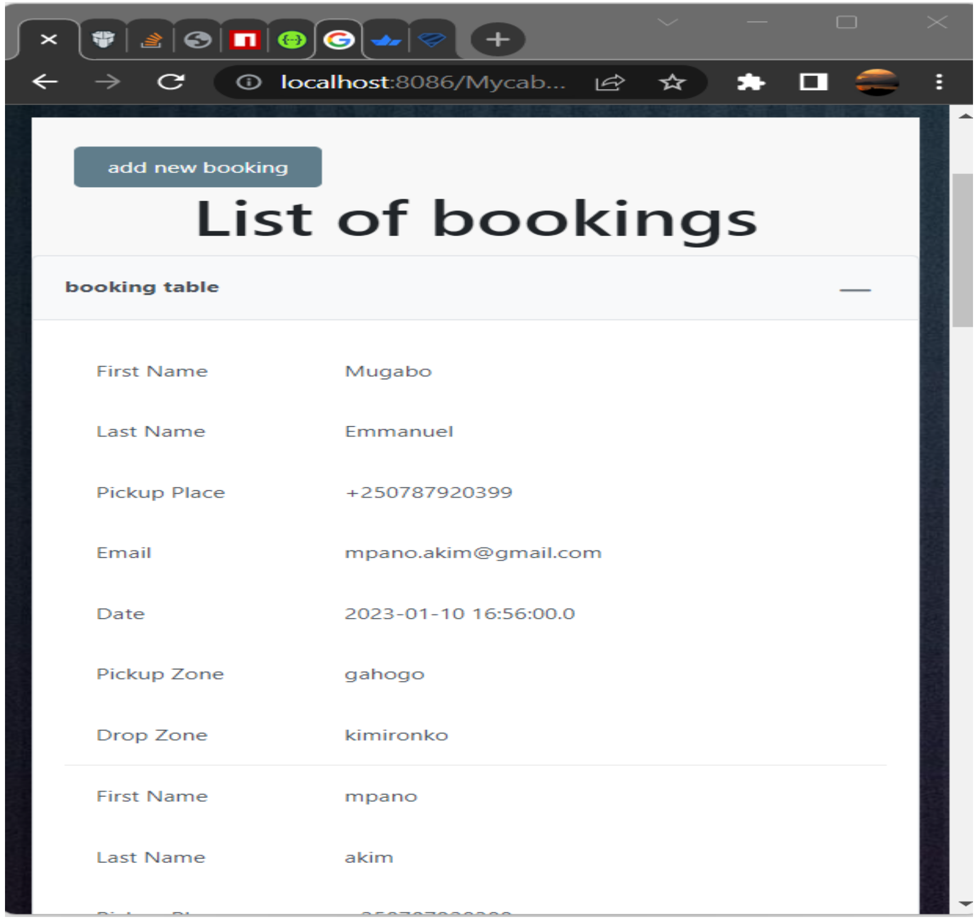
booking table

First Name	Last Name	Pickup Place	Email	Date	Pickup Zone	Drop Zone
Mugabo	Emmanuel	+250787920399	mpano.akim@gmail.com	2023-01-10 16:56:00.0	gahogo	kimironko
mpano	akim	+250787920399	mpano.akim@gmail.com	2023-01-10 16:56:00.0	gahogo	kimironko
Shema	Cedric	+250782506789	mpano.akim@gmail.com	2023-01-10 16:56:00.0	gahogo	kimironko
claude	poul	+250782506789	mpano.akim@gmail.com	2023-01-10 16:56:00.0	gahogo	kimironko
twagiramungu	mpano	+250782506789	mpano.akim@gmail.com	2023-01-10 16:56:00.0	gahogo	kimironko

Private booking table

First Name	Last Name	Private Car	Pickup Place	Email	Date	Pickup Zone	Drop Zone
mpano	akim	Rolls Royce	+250787920399	mpano.akim@gmail.com	2023-01-25 14:35:00.0	kabeza	gishushu

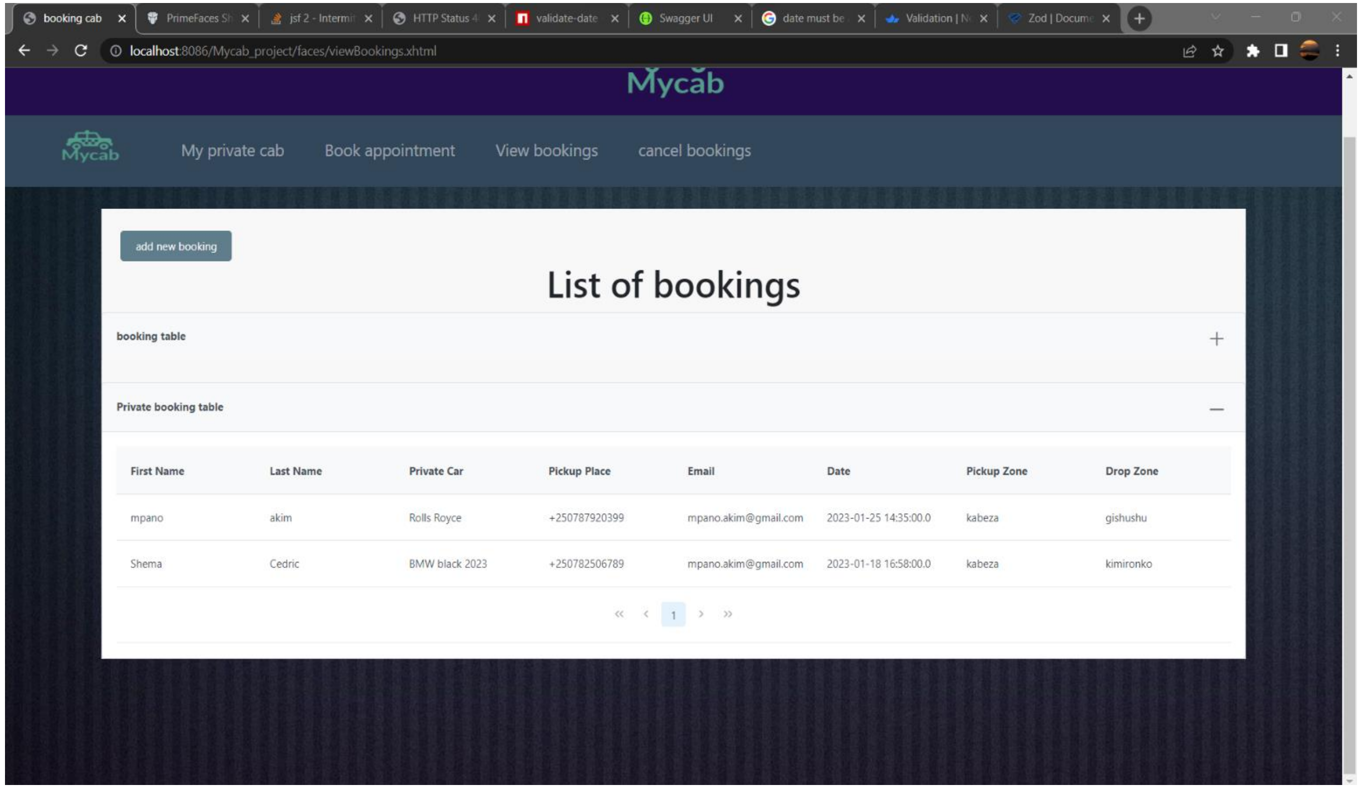
The table also is responsive you can view your data very well on or device



List of bookings

booking table

First Name	Mugabo
Last Name	Emmanuel
Pickup Place	+250787920399
Email	mpano.akim@gmail.com
Date	2023-01-10 16:56:00.0
Pickup Zone	gahogo
Drop Zone	kimironko
First Name	mpano
Last Name	akim



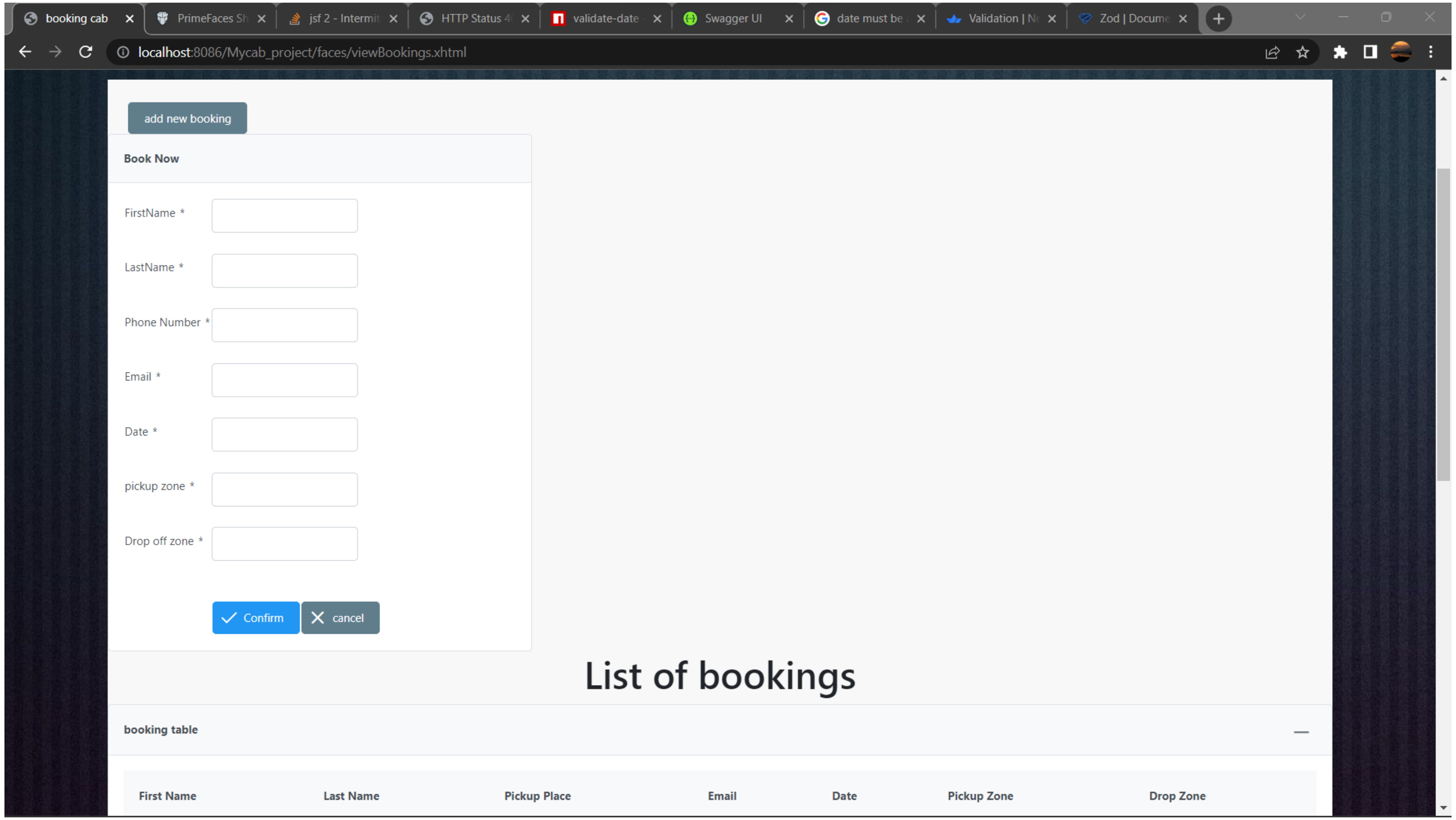
List of bookings

booking table

Private booking table

First Name	Last Name	Private Car	Pickup Place	Email	Date	Pickup Zone	Drop Zone
mpano	akim	Rolls Royce	+250787920399	mpano.akim@gmail.com	2023-01-25 14:35:00.0	kabeza	gishushu
Shema	Cedric	BMW black 2023	+250782506789	mpano.akim@gmail.com	2023-01-18 16:58:00.0	kabeza	kimironko

Using ajax by clicking on add new user on viewbookings you can see the form and book their



Book Now

FirstName *

LastName *

Phone Number *

Email *

Date *

pickup zone *

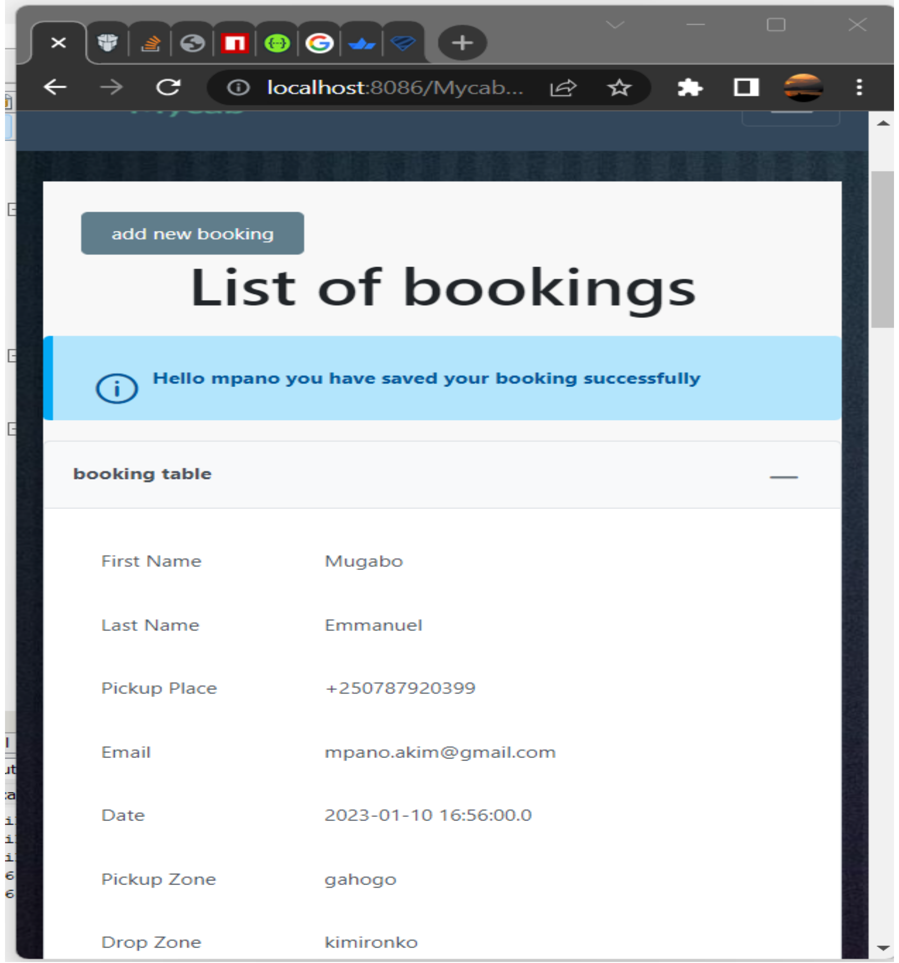
Drop off zone *

List of bookings

booking table

First Name	Last Name	Pickup Place	Email	Date	Pickup Zone	Drop Zone

If you meet all validation you can submit your form and you will be automatically redirect on viewbookings with message telling you that you have saved your booking and you will see it down on the list



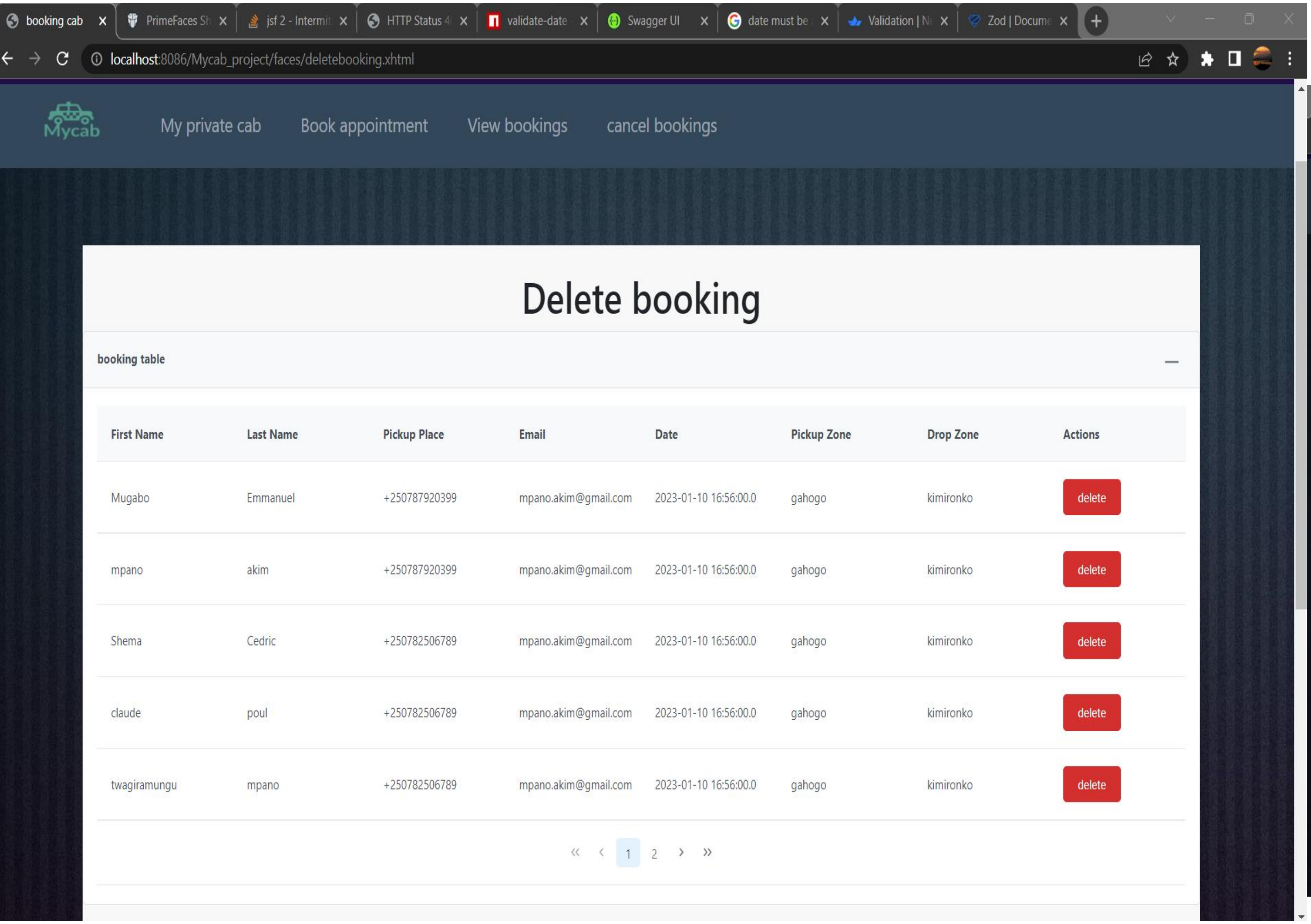
List of bookings

Hello mpano you have saved your booking successfully

booking table

First Name	Mugabo
Last Name	Emmanuel
Pickup Place	+250787920399
Email	mpano.akim@gmail.com
Date	2023-01-10 16:56:00.0
Pickup Zone	gahogo
Drop Zone	kimironko

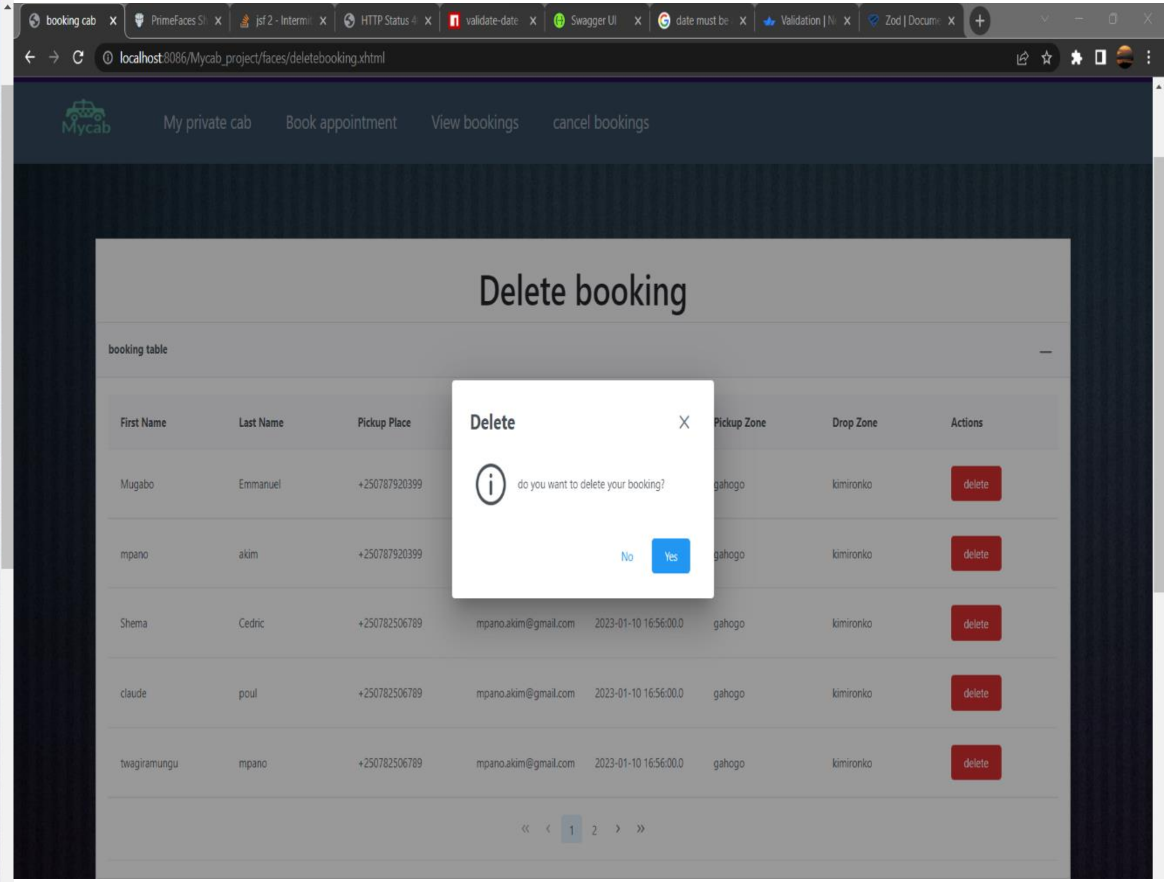
You can also click on cancel booking and see the table with all booking and action button for delete if you click it you can see confirmation dialog if you click yes it will delete it automatically



Delete booking

booking table

First Name	Last Name	Pickup Place	Email	Date	Pickup Zone	Drop Zone	Actions
Mugabo	Emmanuel	+250787920399	mpano.akim@gmail.com	2023-01-10 16:56:00.0	gahogo	kimironko	delete
mpano	akim	+250787920399	mpano.akim@gmail.com	2023-01-10 16:56:00.0	gahogo	kimironko	delete
Shema	Cedric	+250782506789	mpano.akim@gmail.com	2023-01-10 16:56:00.0	gahogo	kimironko	delete
claude	poul	+250782506789	mpano.akim@gmail.com	2023-01-10 16:56:00.0	gahogo	kimironko	delete
twagiramungu	mpano	+250782506789	mpano.akim@gmail.com	2023-01-10 16:56:00.0	gahogo	kimironko	delete



Delete booking

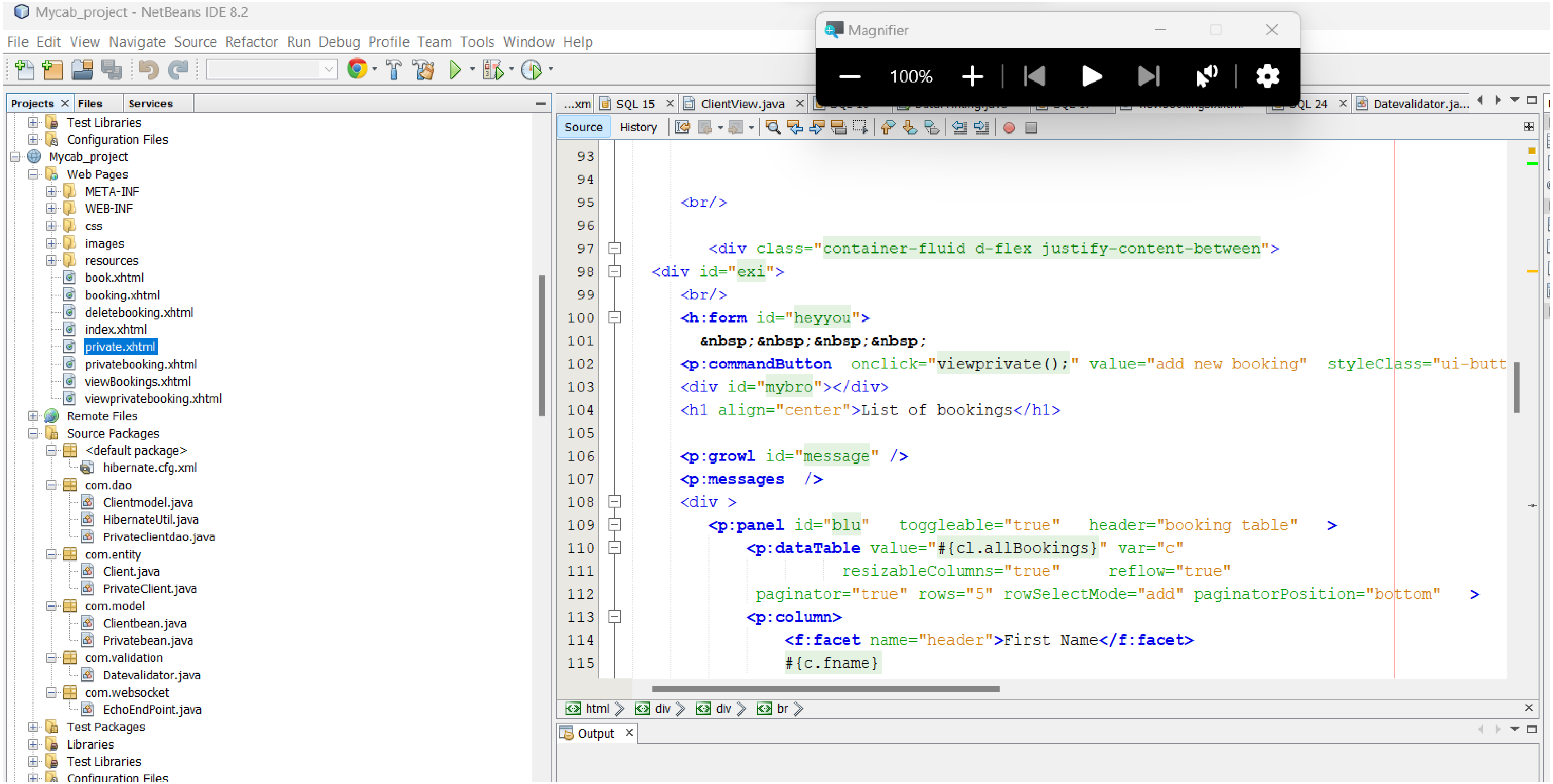
booking table

First Name	Last Name	Pickup Place			Pickup Zone	Drop Zone	Actions
Mugabo	Emmanuel	+250787920399			gahogo	kimironko	delete
mpano	akim	+250787920399			gahogo	kimironko	delete
Shema	Cedric	+250782506789	mpano.akim@gmail.com	2023-01-10 16:56:00.0	gahogo	kimironko	delete
claude	poul	+250782506789	mpano.akim@gmail.com	2023-01-10 16:56:00.0	gahogo	kimironko	delete
twagiramungu	mpano	+250782506789	mpano.akim@gmail.com	2023-01-10 16:56:00.0	gahogo	kimironko	delete

Delete

do you want to delete your booking?

This is the structure of mycab project

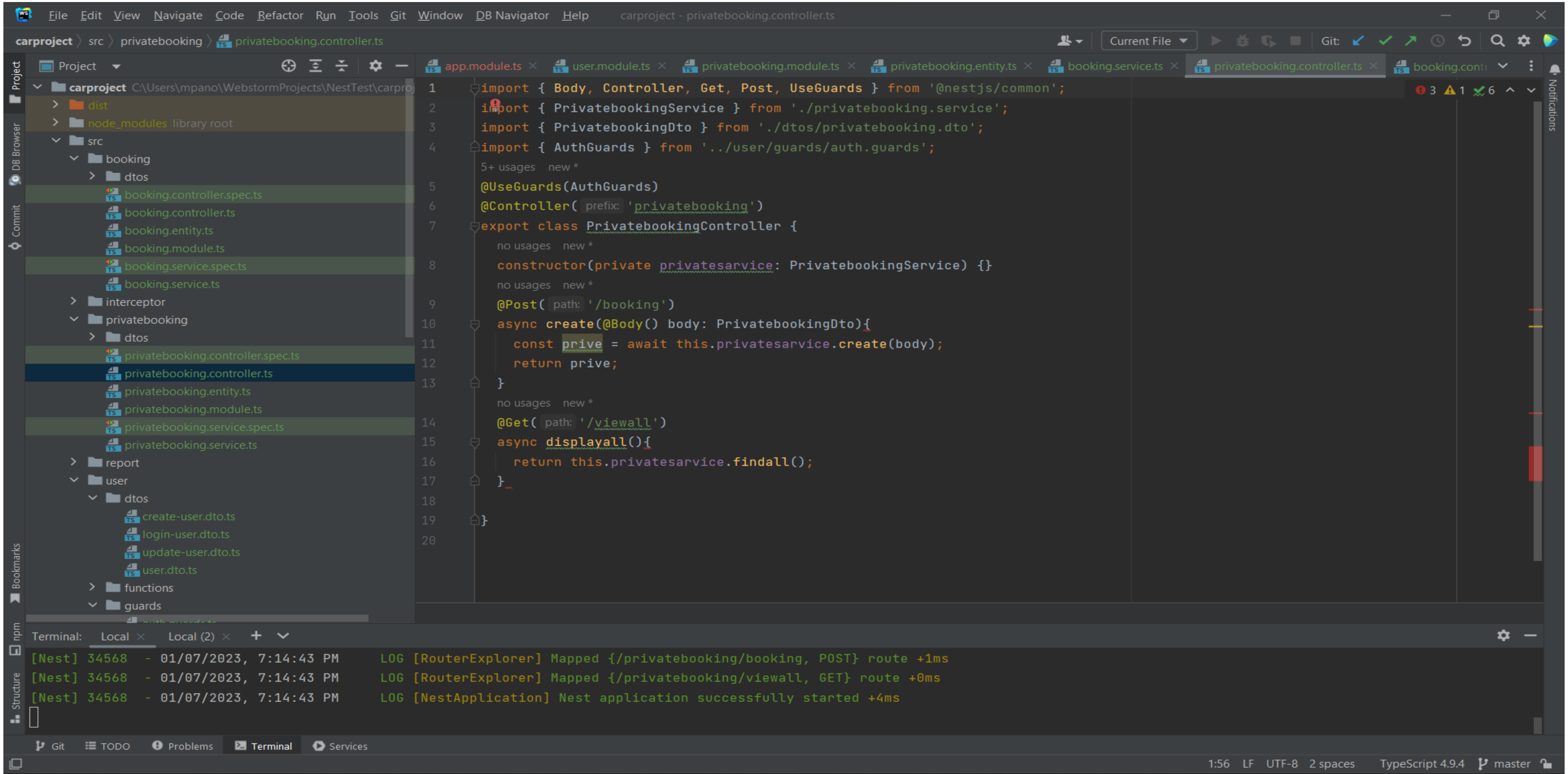


```
Test Libraries
Configuration Files
Mycab_project
  Web Pages
    META-INF
    WEB-INF
    css
    images
    resources
    book.xhtml
    booking.xhtml
    deletebooking.xhtml
    index.xhtml
    private.xhtml
    privatebooking.xhtml
    viewBookings.xhtml
    viewprivatebooking.xhtml
  Remote Files
  Source Packages
    <default package>
      hibernate.cfg.xml
    com.dao
      Clientmodel.java
      HibernateUtil.java
      Privateclientdao.java
    com.entity
      Client.java
      PrivateClient.java
    com.model
      Clientbean.java
      Privatebean.java
    com.validation
      Datevalidator.java
    com.websocket
      EchoEndPoint.java
  Test Packages
  Libraries
  Test Libraries
  Configuration Files
```

```


      <br/>

        <div class="container-fluid d-flex justify-content-between">
    <div id="exi">
      <br/>
      <h:form id="heyyou">
       &nbsp;&nbsp;&nbsp;&nbsp;
      <p:commandButton  onclick="viewprivate();" value="add new booking"  styleClass="ui-butt
      <div id="mybro"></div>
      <h1 align="center">List of bookings</h1>

      <p:growl id="message" />
      <p:messages  />
      <div >
        <p:panel id="blu"   toggleable="true"   header="booking table"   >
          <p:dataTable value="#{cl.allBookings}" var="c"
               resizableColumns="true"     reflow="true"
           paginator="true" rows="5" rowSelectMode="add" paginatorPosition="bottom"   >
          <p:column>
            <f:facet name="header">First Name</f:facet>
            #{c.fname}
```

Using nestjs which is a server side node.js framework I have build mycab backend application I tried to work on many features of my application like users Authentication and authorization where I have admin and normal users which are client. Client are not allowed to delete an user or booking and other things like validation of data all field or input has it’s own validation and also you cant perform any request without login except signup and login other wise you will get 403 forbidden request

This is the structure of mycab backend project in nestjs and an example of one of my controller class privatebooking with two decorators **Get and Post**

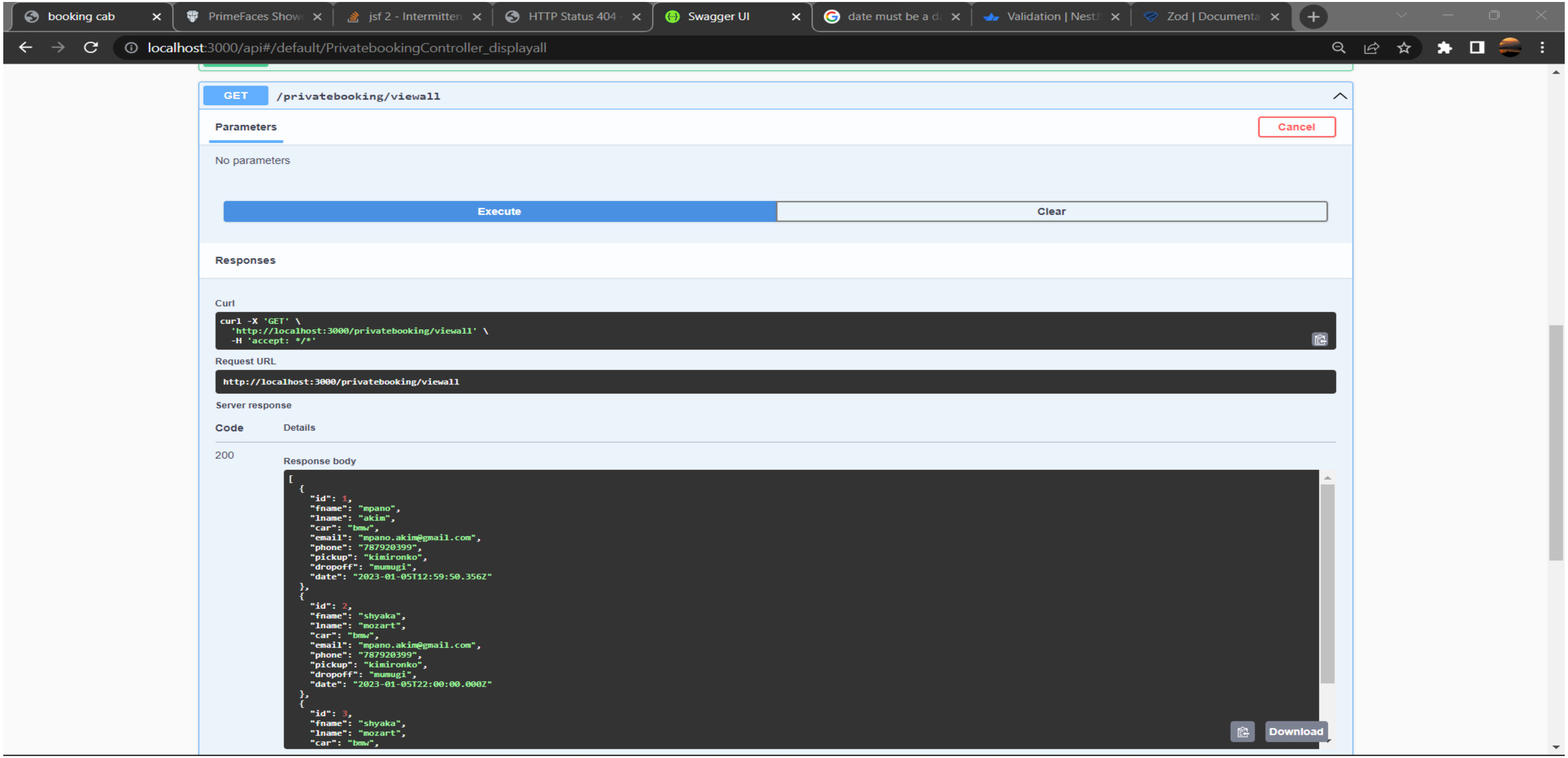


File Edit View Navigate Code Refactor Run Tools Git Window DB Navigator Help
carproject - privatebooking.controller.ts
carproject 〉 src 〉 privatebooking 〉 privatebooking.controller.ts
Current File
Git:
Project
app.module.ts
user.module.ts
privatebooking.module.ts
privatebooking.entity.ts
booking.service.ts
privatebooking.controller.ts
booking.contr
carproject C:\Users\mpano\WebstormProjects\NestTest\carproj
dist
node_modules library root
src
booking
dtos
booking.controller.spec.ts
booking.controller.ts
booking.entity.ts
booking.module.ts
booking.service.spec.ts
booking.service.ts
interceptor
privatebooking
dtos
privatebooking.controller.spec.ts
privatebooking.controller.ts
privatebooking.entity.ts
privatebooking.module.ts
privatebooking.service.spec.ts
privatebooking.service.ts
report
user
dtos
create-user.dto.ts
login-user.dto.ts
update-user.dto.ts
user.dto.ts
functions
guards
5+ usages new *
prefix:
no usages new *
path:
Terminal: Local Local (2)
Git TODO Problems Terminal Services
1:56 LF UTF-8 2 spaces TypeScript 4.9.4 master

```
import { Body, Controller, Get, Post, UseGuards } from '@nestjs/common';
import { PrivatebookingService } from './privatebooking.service';
import { PrivatebookingDto } from './dtos/privatebooking.dto';
import { AuthGuards } from '../user/guards/auth.guards';

@UseGuards(AuthGuards)
@Controller('privatebooking')
export class PrivatebookingController {
  constructor(private privatesarvice: PrivatebookingService) {}
  @Post('/booking')
  async create(@Body() body: PrivatebookingDto){
    const prive = await this.privatesarvice.create(body);
    return prive;
  }
  @Get('/viewall')
  async displayall(){
    return this.privatesarvice.findall();
  }

}
```

```
[Nest] 34568  - 01/07/2023, 7:14:43 PM     LOG [RouterExplorer] Mapped {/privatebooking/booking, POST} route +1ms
[Nest] 34568  - 01/07/2023, 7:14:43 PM     LOG [RouterExplorer] Mapped {/privatebooking/viewall, GET} route +0ms
[Nest] 34568  - 01/07/2023, 7:14:43 PM     LOG [NestApplication] Nest application successfully started +4ms
```

This is view all in table private booking using **swagger UI** which is an api client like postman



GET /privatebooking/viewall

No parameters

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:3000/privatebooking/viewall' \
  -H 'accept: */*'
```

Request URL

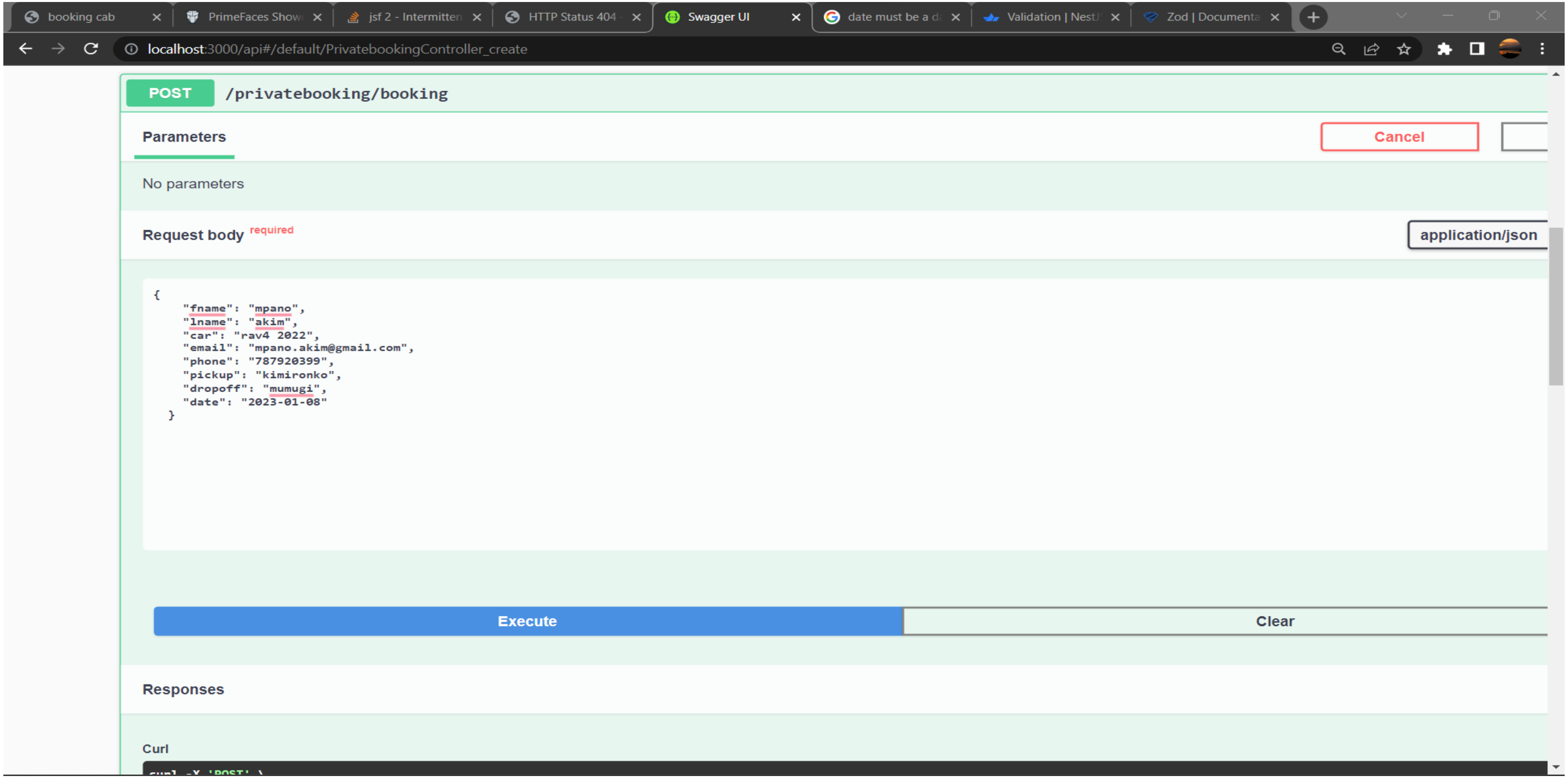
```
http://localhost:3000/privatebooking/viewall
```

Server response

Code	Details
200	Response body

```
[
  {
    "id": 1,
    "fname": "mpano",
    "lname": "akim",
    "car": "bmw",
    "email": "mpano.akim@gmail.com",
    "phone": "787920399",
    "pickup": "kimironko",
    "dropoff": "mumugi",
    "date": "2023-01-05T12:59:50.356Z"
  },
  {
    "id": 2,
    "fname": "shyaka",
    "lname": "mozart",
    "car": "bmw",
    "email": "mpano.akim@gmail.com",
    "phone": "787920399",
    "pickup": "kimironko",
    "dropoff": "mumugi",
    "date": "2023-01-05T22:00:00.000Z"
  },
  {
    "id": 3,
    "fname": "shyaka",
    "lname": "mozart",
    "car": "bmw",
```

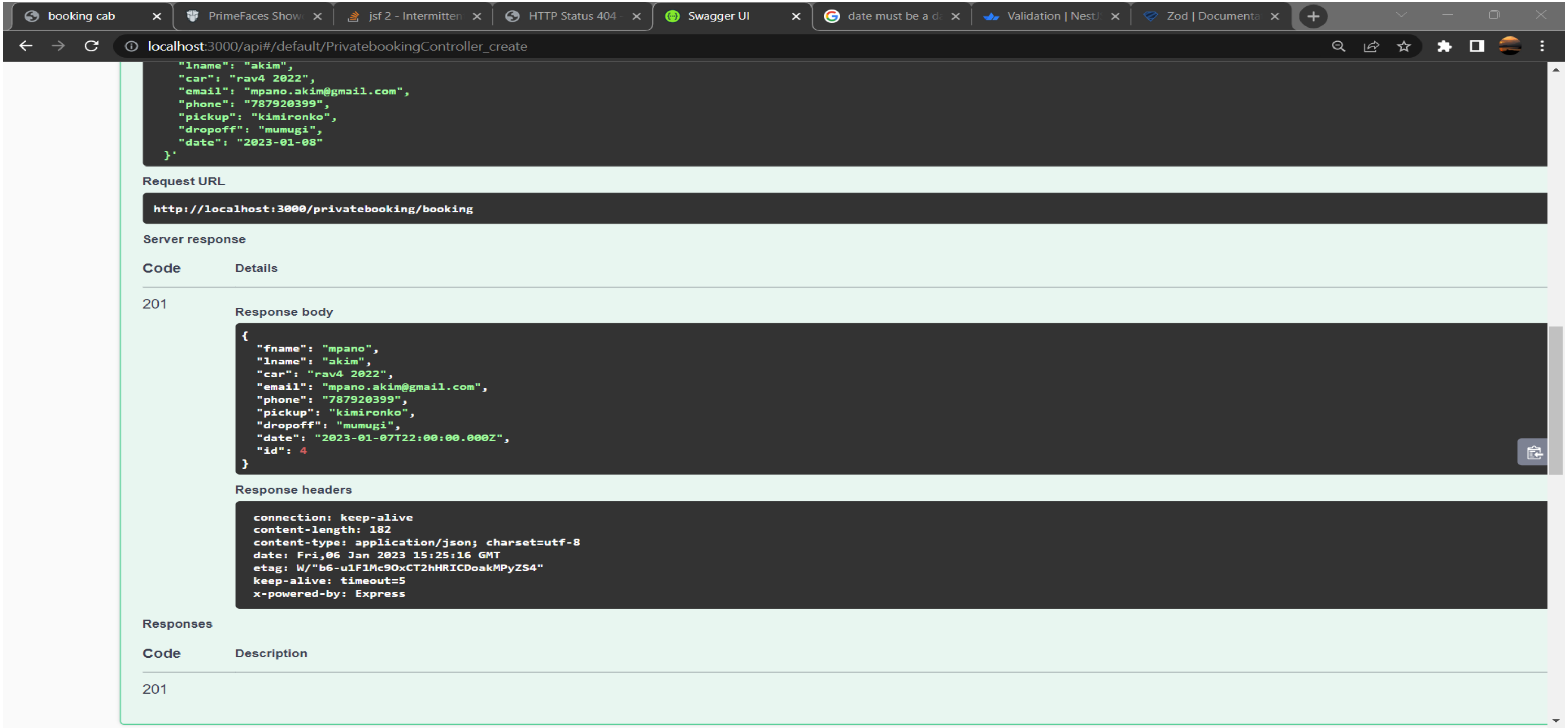
You can also save booking into database (am using postgres) to entering data which we specify in dto class is you enter other one which we don’t specify those one will be use less because it wii not save it into database



```
{
    "fname": "mpano",
    "lname": "akim",
    "car": "rav4 2022",
    "email": "mpano.akim@gmail.com",
    "phone": "787920399",
    "pickup": "kimironko",
    "dropoff": "mumugi",
    "date": "2023-01-08"
  }
```

```
curl -X 'POST' \
```

If your data satisfy our validation you will see 201 response and return the user you just save in database other wise you will see 404 and 401 for bad request



```
   "lname": "akim",
   "car": "rav4 2022",
   "email": "mpano.akim@gmail.com",
   "phone": "787920399",
   "pickup": "kimironko",
   "dropoff": "mumugi",
   "date": "2023-01-08"
 }'
```

Request URL

```
http://localhost:3000/privatebooking/booking
```

Server response

Code	Details
201	Response body

```
{
  "fname": "mpano",
  "lname": "akim",
  "car": "rav4 2022",
  "email": "mpano.akim@gmail.com",
  "phone": "787920399",
  "pickup": "kimironko",
  "dropoff": "mumugi",
  "date": "2023-01-07T22:00:00.000Z",
  "id": 4
}
```

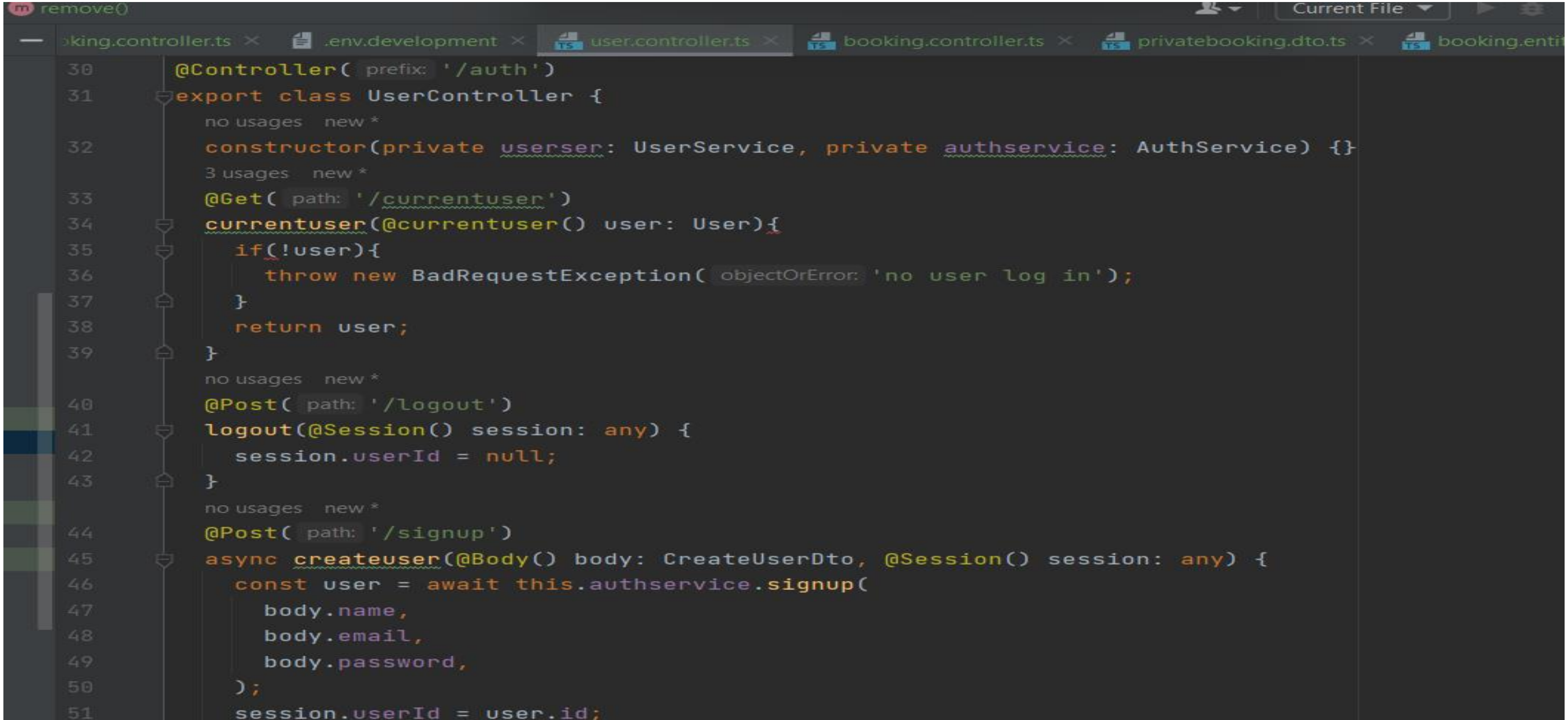
Response headers

```
 connection: keep-alive
 content-length: 182
 content-type: application/json; charset=utf-8
 date: Fri,06 Jan 2023 15:25:16 GMT
 etag: W/"b6-u1F1Mc9OxCT2hHRICDoakMPyZS4"
 keep-alive: timeout=5
 x-powered-by: Express
```

Responses

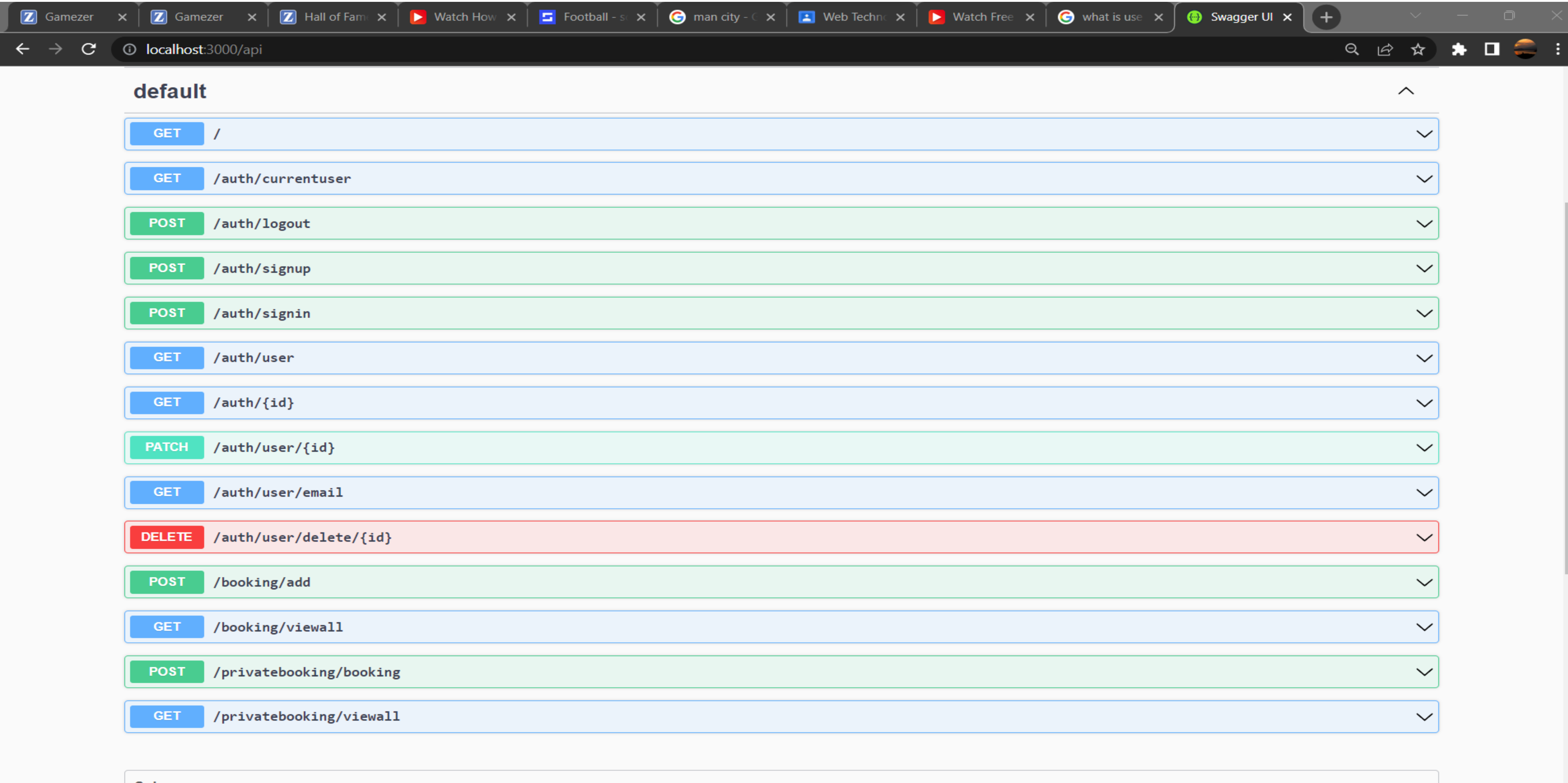
Code	Description
201	
]

Using typeORM it create for you repository class and what you have to do is creating service class to access your repository and controller and other class follow the structure of nestjs. this is my user controller class



```
@Controller( prefix: '/auth')
export class UserController {
  no usages   new *
  constructor(private userser: UserService, private authservice: AuthService) {}
  3 usages   new *
  @Get( path: '/currentuser')
  currentuser(@currentuser() user: User){
    if(!user){
      throw new BadRequestException( objectOrError: 'no user log in');
    }
    return user;
  }
  no usages   new *
  @Post( path: '/logout')
  logout(@Session() session: any) {
    session.userId = null;
  }
  no usages   new *
  @Post( path: '/signup')
  async createuser(@Body() body: CreateUserDto, @Session() session: any) {
    const user = await this.authservice.signup(
      body.name,
      body.email,
      body.password,
    );
    session.userId = user.id;
```

This is all the request you can perform(send)



default

GET	/
GET	/auth/currentuser
POST	/auth/logout
POST	/auth/signup
POST	/auth/signin
GET	/auth/user
GET	/auth/{id}
PATCH	/auth/user/{id}
GET	/auth/user/email
DELETE	/auth/user/delete/{id}
POST	/booking/add
GET	/booking/viewall
POST	/privatebooking/booking
GET	/privatebooking/viewall