Practical Assignment -3

Name:Rana Aayurshi VijayKumar

Semester: 7th

Roll No.:39

Subject:Application development using FullStack(701)

Question 1:

Create Angular application with routing link assigned for following requirements:.

- Display integer, sting, date data in a component.

- Display array values in html list.

- Display array in combo.

- Take a combo with two values (textbox, textarea). Display textbox or textarea depending on selection of a combo.

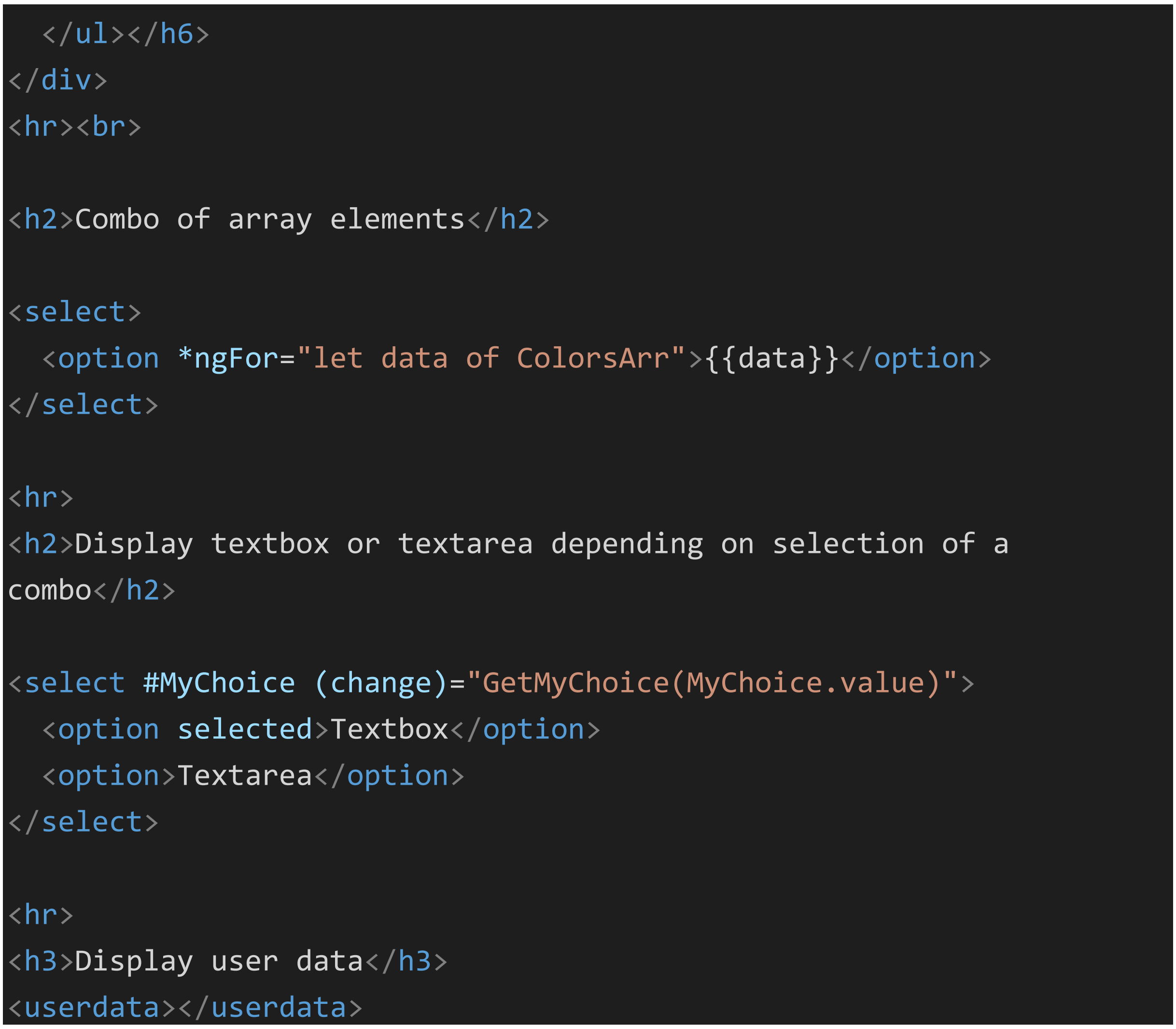
-Make another component for user input form with user’s personal details, take input and display data on submit.

Solution:

→Code:

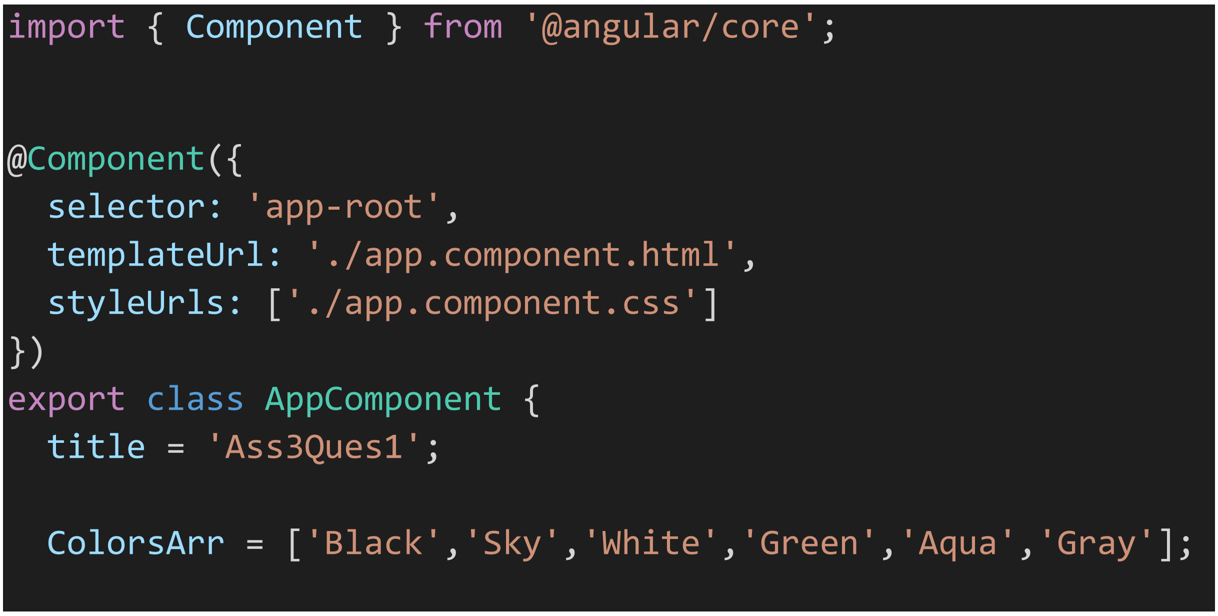
-app.component.html

```
<component1></component1>
<hr><br>
<div>
  <h2>List of Array elements</h2>
  <h6><ul *ngFor="let data of ColorsArr">
      <li>{{data}}</li>
```



```
  </ul></h6>
</div>
<hr><br>

<h2>Combo of array elements</h2>

<select>
  <option *ngFor="let data of ColorsArr">{{data}}</option>
</select>

<hr>
<h2>Display textbox or textarea depending on selection of a
combo</h2>

<select #MyChoice (change)="GetMyChoice(MyChoice.value)">
  <option selected>Textbox</option>
  <option>Textarea</option>
</select>

<hr>
<h3>Display user data</h3>
<userdata></userdata>
```

-app.component.ts:



```
import { Component } from '@angular/core';


@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'Ass3Ques1';

  ColorsArr = ['Black','Sky','White','Green','Aqua','Gray'];
  GetMyChoice(value:any)
  {
      alert(value);
  }

}
```

-comp1.Component.html:

```
<div>
    <h2>==>Integer,String and Date values : </h2>
      <h4>Integer value : {{intValue}}<br/>
      String value : {{strValue}}<br/>
      Date value : {{dateValue}}</h4>
</div>
```

-comp1.component.ts:

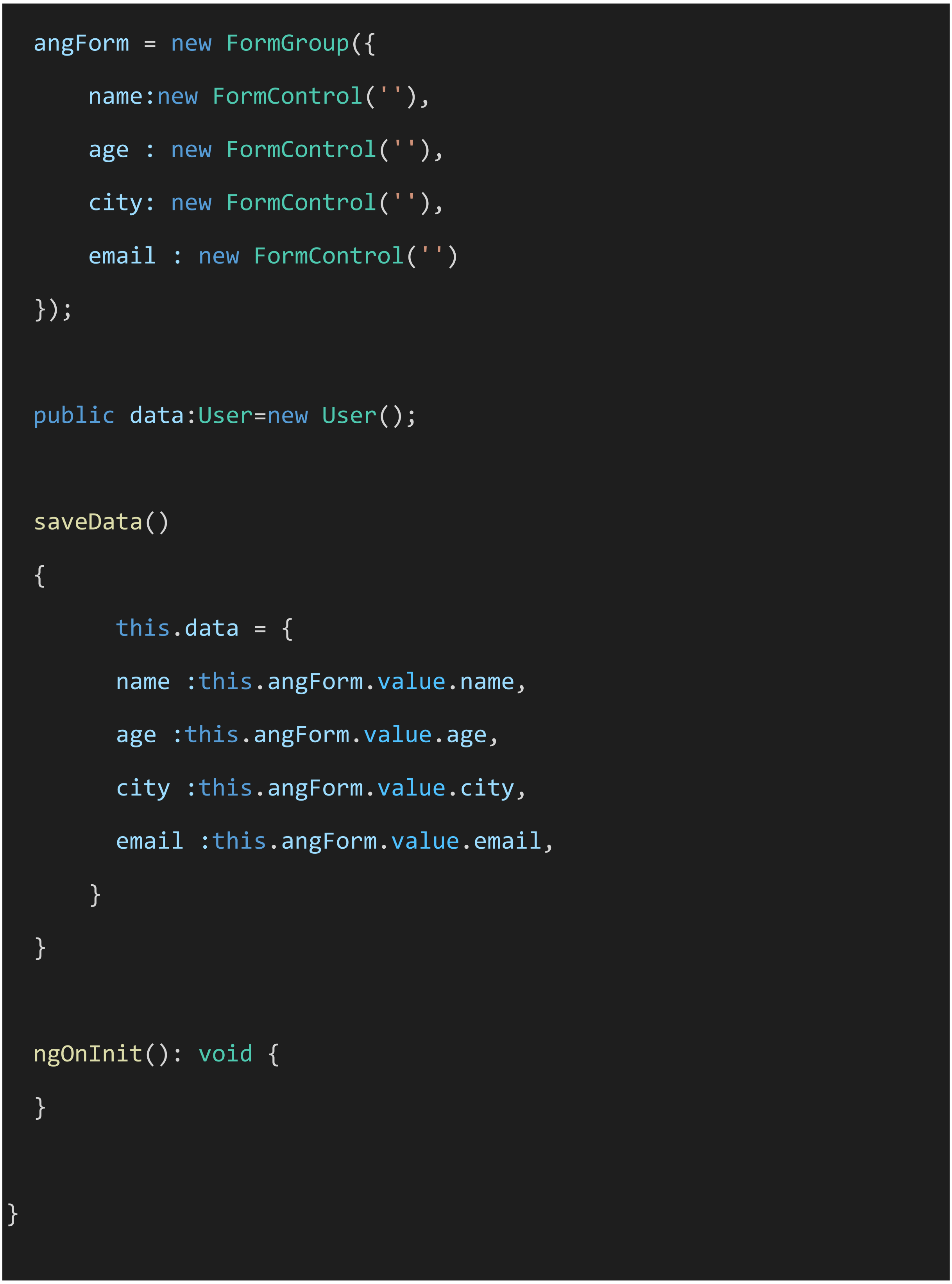
```
import { Component, OnInit } from '@angular/core';


@Component({
  selector: 'component1',
  templateUrl: './comp1.component.html',
  styleUrls: ['./comp1.component.css']
})
export class Comp1Component implements OnInit {

  constructor() { }

  ngOnInit(): void {
  }


  intValue:number = 176;

  strValue:string = "Aayurshi Rana";

  dateValue:Date = new Date();


}
```

-userdata.component.html

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <form method="post" [formGroup]="angForm" (submit)="saveData()">

        Name : <input type="text" formControlName="name"><br/>

        Age : <input type="number" formControlName="age"><br/>

        City : <input type="text" formControlName="city"><br/>

        Email : <input type="email" formControlName="email"><br/>
        <button>Submit</button>
    </form>

    <div>
        Name : {{data.name}}<br>
        Age : {{data.age}}<br>
        City : {{data.city}}<br>
        Email : {{data.email}}
    </div>

</body>
</html>
```

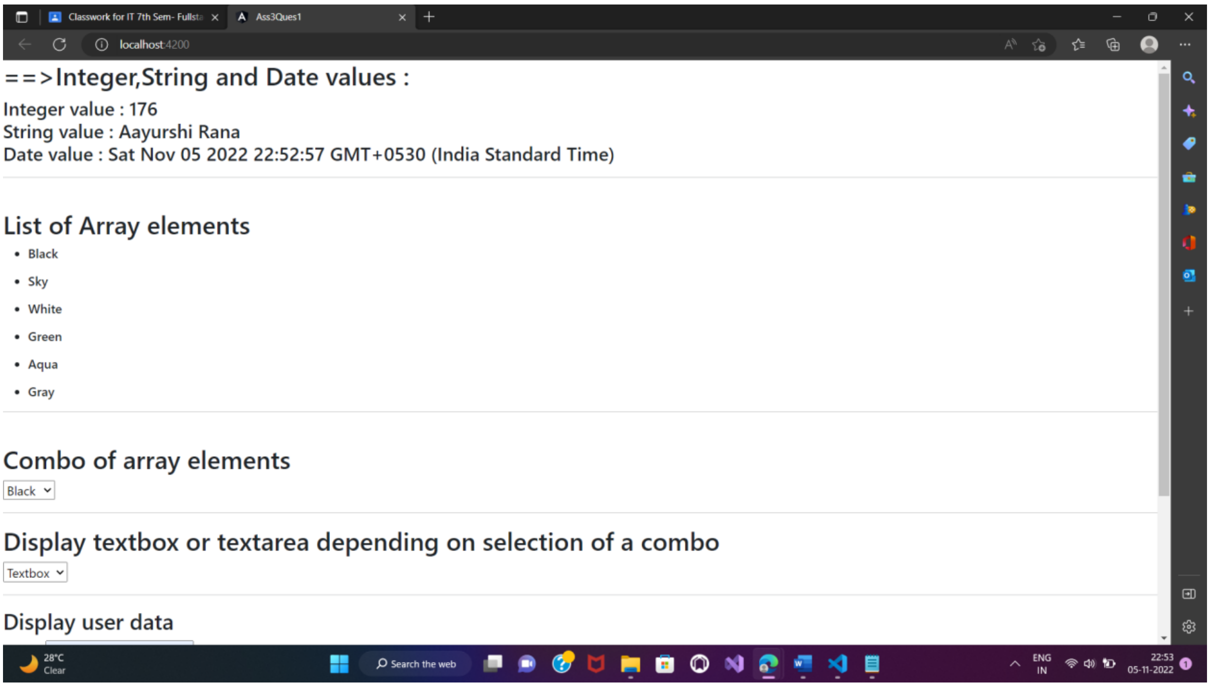
-userdata.component.ts

```
import { Component, OnInit } from '@angular/core';
import { FormGroup,FormControl} from '@angular/forms';
import { User } from '../Models/user';

@Component({
  selector: 'userdata',
  templateUrl: './userdata.component.html',
  styleUrls: ['./userdata.component.css']
})
export class UserdataComponent implements OnInit {

  constructor() { }

```



```
  angForm = new FormGroup({
      name:new FormControl(''),
      age : new FormControl(''),
      city: new FormControl(''),
      email : new FormControl('')
  });

  public data:User=new User();

  saveData()
  {
        this.data = {
        name :this.angForm.value.name,
        age :this.angForm.value.age,
        city :this.angForm.value.city,
        email :this.angForm.value.email,
      }
  }

  ngOnInit(): void {
  }

}
```

->Output:



==>Integer,String and Date values :

Integer value : 176
String value : Aayurshi Rana
Date value : Sat Nov 05 2022 22:52:57 GMT+0530 (India Standard Time)

List of Array elements

- Black
- Sky
- White
- Green
- Aqua
- Gray

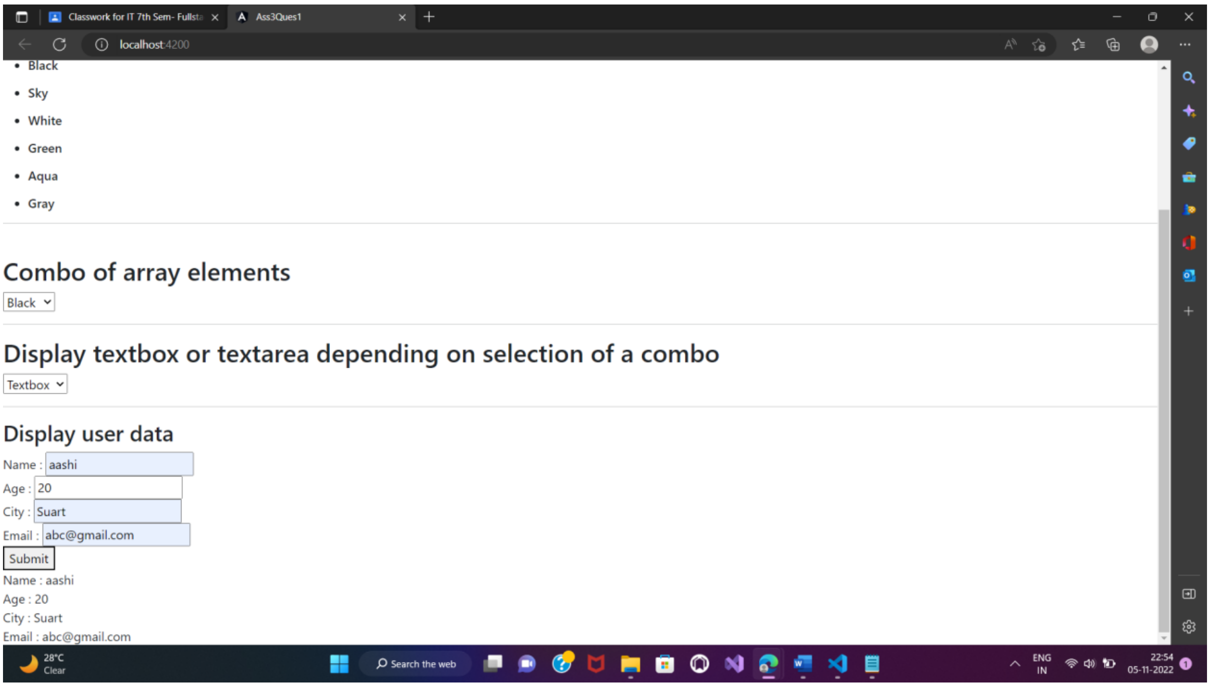
Combo of array elements

Black

Display textbox or textarea depending on selection of a combo

Textbox

Display user data



- Black
- Sky
- White
- Green
- Aqua
- Gray

Combo of array elements

Black

Display textbox or textarea depending on selection of a combo

Textbox

Display user data

Name : aashi
Age : 20
City : Suart
Email : abc@gmail.com
Submit

```
Name : aashi
Age : 20
City : Suart
Email : abc@gmail.com
```

Question 3:

Design and develop a module to manage vehicle data for a vehicle resale

agency considering following requirements.

- Design two collections

o (1) vehicle_category (c_id, category_name)

o (2) vehicle_data (_id, vehicle_brand, category_name, vehicle picture,

price, depreciation , number of years, total price)

- Insert vehicle

o upload a picture

o Load category data in a combo

o Establish relation between two collections

- List vehicle data along with category_name.

Solution:

Backend:

-app.js

```
const express = require("express");
const app = express();
const vehRoute = require("./Route/VehRoute");
const dbCon = require("./DbConnection/dbCon");
const cors = require("cors");

app.use(express.urlencoded({extended:false}));
app.use(express.json());
app.use(cors());
app.use("/",vehRoute);


const start = ()=>{
    try{
            dbCon("mongodb://localhost:27017/vehicleDb");
            var server = app.listen(8000,()=>{
                console.log("Server is listening on port " +
server.address().port);
            });
        }
    catch(err)
    {
        console.log("Error : " + err);
    }
};

start();
```

-mongoDb Connection:

```
const mongoose = require("mongoose");

const Connection = (url)=>{
    try{
        mongoose.connect(url);
        console.log("Mongodb connected successfully");
      }
    catch(err)
    {
        console.log("Can not connect mongoDb\n" + err);
    }
};

module.exports = Connection;
```

-Models:

-Vehicle:

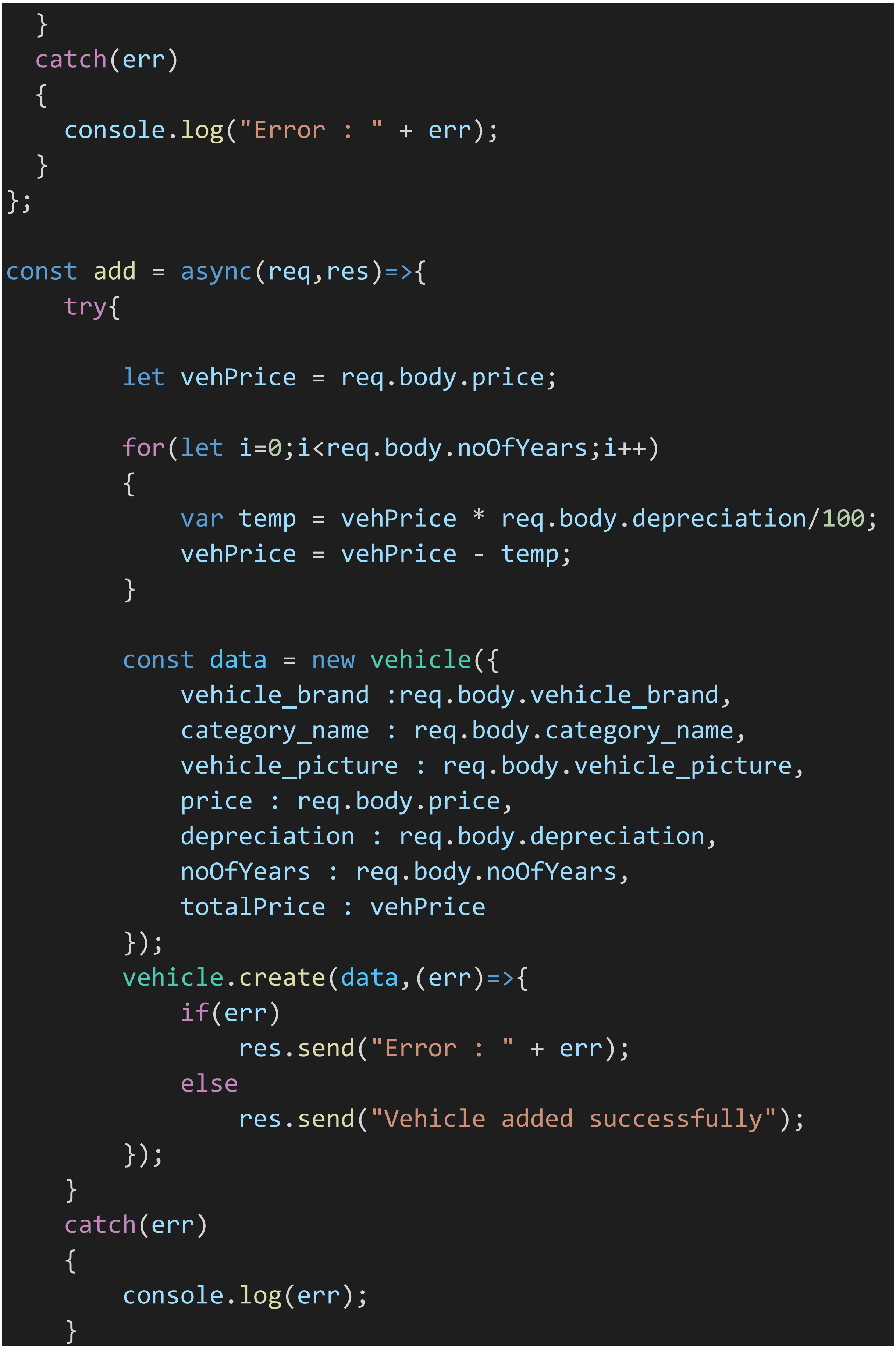
```
const mongoose = require("mongoose");

const vehSchema = new mongoose.Schema({
    vehicle_brand:{
        type : String
    },
    category_name : {
        type:String
    },
    vehicle_picture : {
        type:String
    },
    price : {
        type:Number
    },
    depreciation:{
        type : Number
    },
    noOfYears : {
        type:Number
    },
    totalPrice : {
        type:Number
    },
});

const vehModel = mongoose.model("Vehicles",vehSchema);

module.exports = vehModel;
```

-Category:

```
const mongoose = require("mongoose");

const categorySchema = new mongoose.Schema({
    category_name : {
        type:String
    }
});

const categoryModel = mongoose.model("Categories",categorySchema);
module.exports = categoryModel;
```

-Controller:

```
const vehicle = require("../Model/vehicle");
const category = require("../Model/category");

const display = async(req,res)=>{
  try{
        var data = vehicle.find({},(err,data)=>{
            if(err)
                console.log("Error : " + err);
            else
                console.log("display method called");
                res.send(data);
        })
```



```
  }
  catch(err)
  {
    console.log("Error : " + err);
  }
};

const add = async(req,res)=>{
    try{

        let vehPrice = req.body.price;

        for(let i=0;i<req.body.noOfYears;i++)
        {
            var temp = vehPrice * req.body.depreciation/100;
            vehPrice = vehPrice - temp;
        }

        const data = new vehicle({
            vehicle_brand :req.body.vehicle_brand,
            category_name : req.body.category_name,
            vehicle_picture : req.body.vehicle_picture,
            price : req.body.price,
            depreciation : req.body.depreciation,
            noOfYears : req.body.noOfYears,
            totalPrice : vehPrice
        });
        vehicle.create(data,(err)=>{
            if(err)
                res.send("Error : " + err);
            else
                res.send("Vehicle added successfully");
        });
    }
    catch(err)
    {
        console.log(err);
    }
```

```
};

const loadCategories = async(req,res)=>{
    try{
            var data = category.find({},(err,data)=>{
                if(err)
                    console.log("Error : " + err);
                else
                    res.send(data);
            })
    }
    catch(err)
    {
        console.log("Error : " + err);
    }
};

module.exports = {display,add,loadCategories}
```

-Route:

```
const {display,add,loadCategories} =
require("../Controller/vehController");
const vehRoute = require("express").Router();
const multer = require("multer");

var options = multer.diskStorage({
    destination : function(req,file,cb){
        cb(null,'./uploads');
    },
    filename : function(req,file,cb){
        cb(null,file.originalname );
    }
});

var upload = multer({storage:options});

vehRoute.route("/").get(display);
vehRoute.route("/add").post(upload.single("Image"),add).get(
loadCategories);

module.exports = vehRoute;
```

→FrontEnd:

Classes:

-vehicles.ts:

```
export class Vehicles {
    _id : string="";
    vehicle_brand: string="";
    category_name:string="";
    vehicle_picture :string="";
    price :number=0;
    depreciation:number=0;
    noOfYears:number=0;
    totalPrice : number=0;
}
```

-categories.ts:

```
export class Categories {
    _id:string="";
    category_name : string="";
}
```

-app.component.html:

```
<h3><a href="/add" class="link-primary" style="text-decoration:
none;">Add New</a></h3>
```

```
<router-outlet></router-outlet>
```

-service:

```
import { Injectable } from '@angular/core';
import {HttpClient} from '@angular/common/http';
import { Observable } from 'rxjs';
import{Vehicles} from '../Models/vehicles';
import{Categories} from '../Models/categories';

@Injectable({
  providedIn: 'root'
})
export class MySericeService {

  constructor(private http:HttpClient) { }

  getCategories():Observable<Categories[]>
  {
     var data =
this.http.get<Categories[]>('http://localhost:8000/add');
     return data;
  }

  getVehicles():Observable<Vehicles[]>
  {
      var data =
this.http.get<Vehicles[]>('http://localhost:8000');
      return data;
  }

  addVehicle(brand:any,category:any,pic:any,price:any,depr:any,ye
ars:any,total:any)
  {
      const data = {
        vehicle_brand:brand,
        category_name : category,
        vehicle_picture : pic,
        price : price,
        depreciation  :depr,
        noOfYears : years,
        totalPrice : total
      }
      console.log(data);
      this.http.post("http://localhost:8000/add",data).subscribe(
res=>console.log("Vehicle added successfully"));
  }
}
```

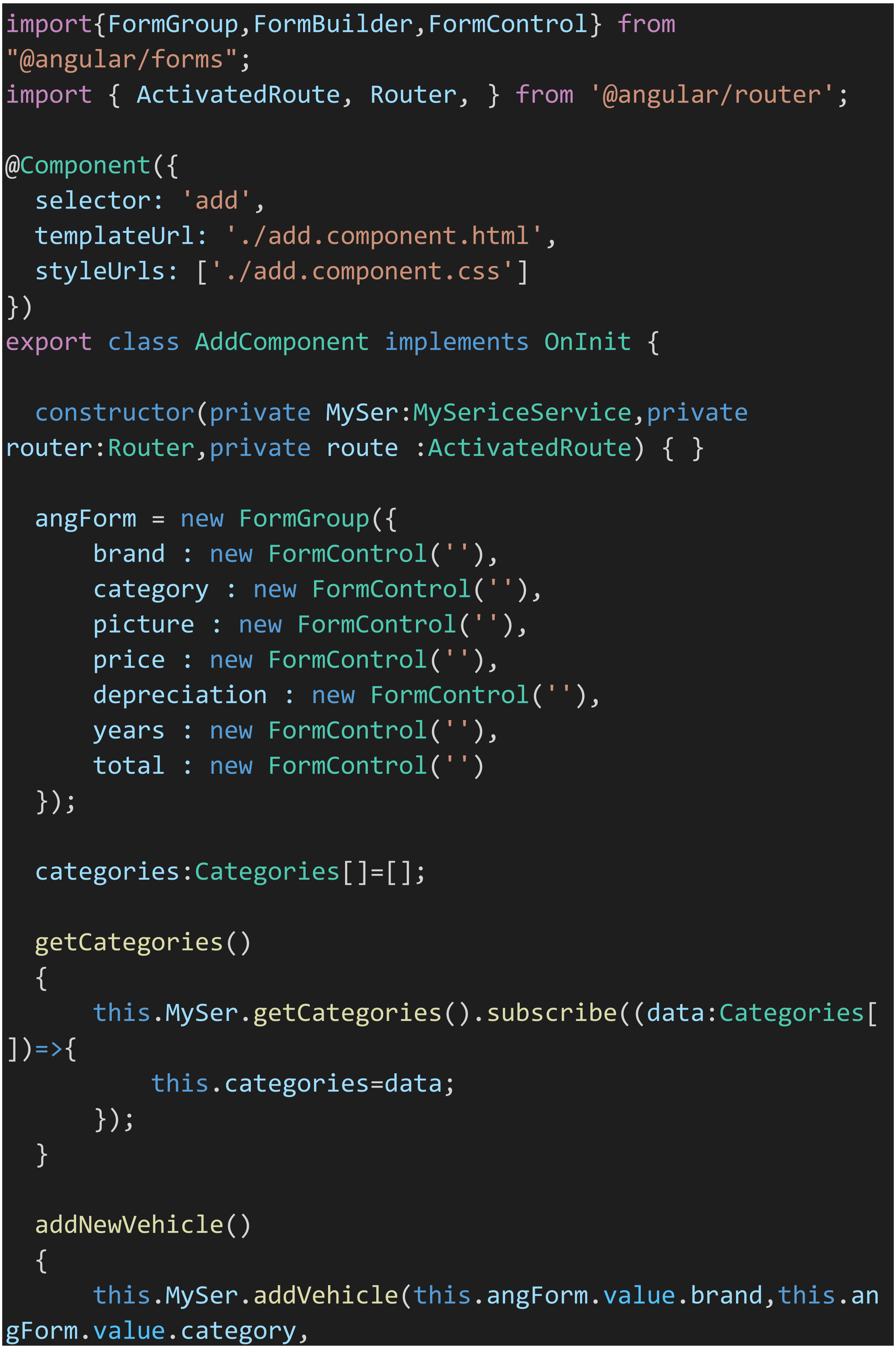
Add.component.html:

```
<html>
    <head>
        <title>Add new vehicle</title>
        <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/
bootstrap.min.css" rel="stylesheet">
    </head>
    <body>
        <form method="post" [formGroup]="angForm"
(submit)="addNewVehicle()" enctype="multipart/form-data">

            <div class="form-group col-md-6">
              <h6><label>Vehicle Brand</label></h6>
              <input type="text" class="form-control"
formControlName="brand"  name="vehicle_brand"
placeholder="Vehicle Brand">
            </div>

            <div class="form-group col-md-4">
              <h6><label>Vehicle category</label></h6>
              <select name="category_name"
formControlName="category" class="form-control">
                <option *ngFor="let cate of
categories">{{cate.category_name}}</option>
              </select>
            </div>

            <div class="form-group col-md-6">
```

```
              <h6><label>Vehicle Picture</label></h6>
              <input type="file" class="form-control"
formControlName="picture" name="vehicle_picture">
            </div>

            <div class="form-group col-md-6">
              <h6><label>Price</label></h6>
              <input type="number" class="form-control"
formControlName="price" name="price" placeholder="Price">
            </div>

            <div class="form-group col-md-6">
              <h6><label>Depreciation</label></h6>
              <input type="number" class="form-control"
name="depreciation" formControlName="depreciation"
placeholder="Depreciation">
            </div>

            <div class="form-group col-md-6">
              <h6><label>No of years</label></h6>
              <input type="number" class="form-control"
name="noOfYears" placeholder="No of years"
formControlName="years">
            </div><br/>
            <button type="submit"  class="btn btn-
primary">Add Vehicle</button>
        </form>
    </body>
</html>
```

-add.component.ts:

```
import { Component, OnInit } from '@angular/core';
import { Vehicles } from '../Models/vehicles';
import { Categories } from '../Models/categories';
import { MySericeService } from '../services/my-
serice.service';
```



```
import{FormGroup,FormBuilder,FormControl} from
"@angular/forms";
import { ActivatedRoute, Router, } from '@angular/router';

@Component({
  selector: 'add',
  templateUrl: './add.component.html',
  styleUrls: ['./add.component.css']
})
export class AddComponent implements OnInit {

  constructor(private MySer:MySericeService,private
router:Router,private route :ActivatedRoute) { }

  angForm = new FormGroup({
      brand : new FormControl(''),
      category : new FormControl(''),
      picture : new FormControl(''),
      price : new FormControl(''),
      depreciation : new FormControl(''),
      years : new FormControl(''),
      total : new FormControl('')
  });

  categories:Categories[]=[];

  getCategories()
  {
      this.MySer.getCategories().subscribe((data:Categories[
])=>{
          this.categories=data;
      });
  }

  addNewVehicle()
  {
      this.MySer.addVehicle(this.angForm.value.brand,this.an
gForm.value.category,
        this.angForm.value.picture,this.angForm.value.price,
        this.angForm.value.depreciation,this.angForm.value.y
ears,this.angForm.value.total);
      this.router.navigate([""]);
    }

  ngOnInit(): void {
    this.getCategories();
  }

}
```

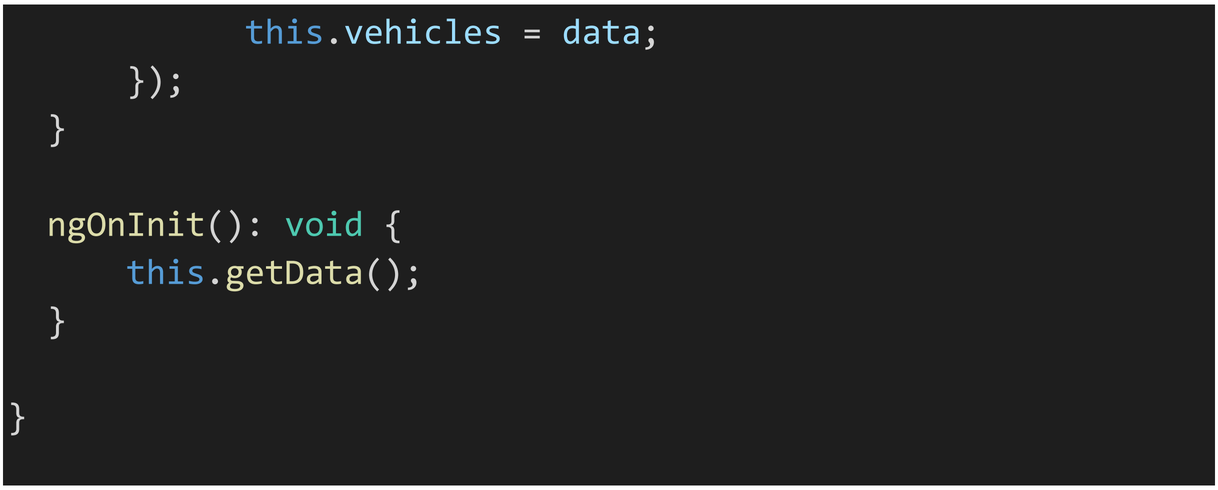
-home.component.html:

```
<html>
    <head>
        <title>Display Vehicles</title>
        <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/
bootstrap.min.css" rel="stylesheet">
    </head>

    <body>
        <table class="table table-striped table-bordered">
            <thead>
            <tr>
                <th>Vehicle brand</th>
                <th>Category name</th>
                <th>Picture</th>
                <th>Price</th>
                <th>Depreciation %</th>
                <th>No of years</th>
                <th>Total price</th>
            </tr>

            <tr *ngFor="let data of vehicles">
```

```
                <td>{{data.vehicle_brand}}</td>
                <td>{{data.category_name}}</td>
                <td><img src="{{data.vehicle_picture}}"
width="200"></td>
                <td>{{data.price}}</td>
                <td>{{data.depreciation}}</td>
                <td>{{data.noOfYears}}</td>
                <td>{{data.totalPrice}}</td>
            </tr>
            </thead>

        </table>
    </body>
</html>
```

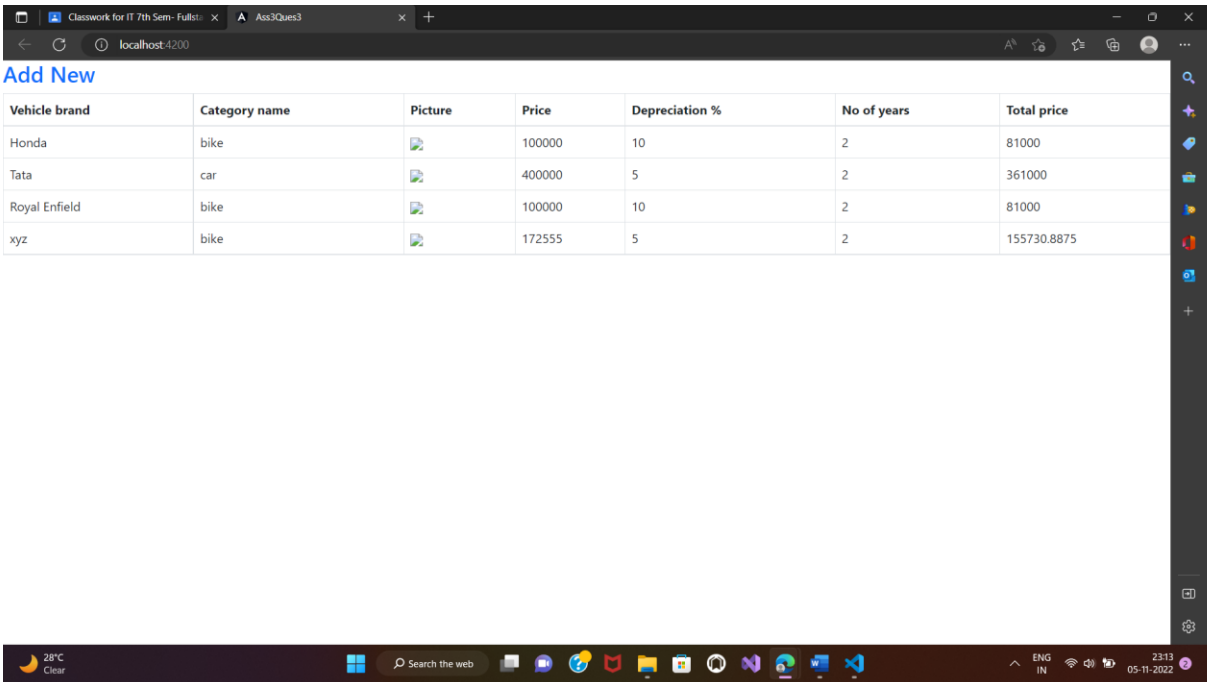
-home.component.ts:

```
import { Component, OnInit } from '@angular/core';
import { MySericeService } from '../services/my-
serice.service';
import { Vehicles } from '../Models/vehicles';

@Component({
  selector: 'display',
  templateUrl: './home.component.html',
  styleUrls: ['./home.component.css']
})
export class HomeComponent implements OnInit {

  constructor(private MySer : MySericeService) { }

  vehicles:Vehicles[]=[];

  getData()
  {
      this.MySer.getVehicles().subscribe((data:Vehicles[])=>
{
            console.log(data);
```



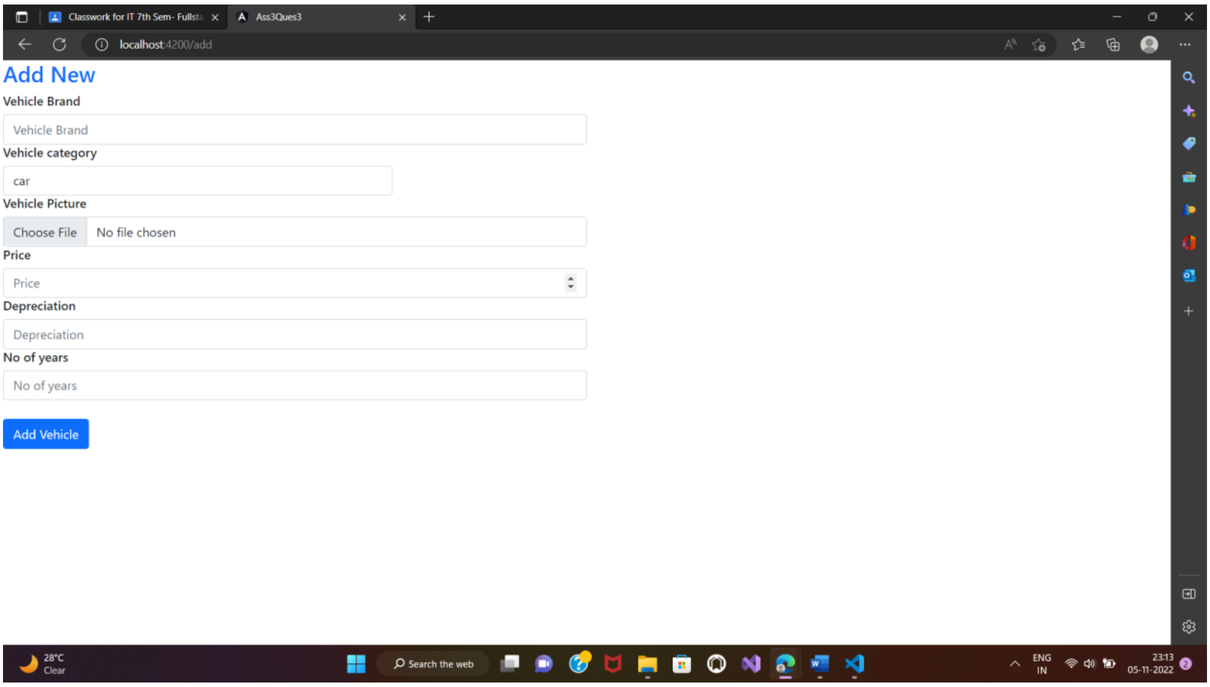
```
            this.vehicles = data;
      });
  }

  ngOnInit(): void {
      this.getData();
  }

}
```

-Output:



Add New

Vehicle brand	Category name	Picture	Price	Depreciation %	No of years	Total price
Honda	bike		100000	10	2	81000
Tata	car		400000	5	2	361000
Royal Enfield	bike		100000	10	2	81000
xyz	bike		172555	5	2	155730.8875



Add New

Vehicle Brand

Vehicle Brand

Vehicle category

car

Vehicle Picture

Choose File No file chosen

Price

Price

Depreciation

Depreciation

No of years

No of years

Add Vehicle

Question 2:

Design and develop shopping cart system.

Backend - Products (CRUD) API in Express.

Frontend

- Call CRUD API

- Display products in grid (3xn)

Solution:

Backend:

-app.js:

```
const express = require("express");
const app = express();
const dbCon = require("./MongoDb/dbCon");
const ShopRoute = require("./Route/ShoppingRoute");

app.use(express.urlencoded({extended:false}));
app.use(express.json());
app.use("/",ShopRoute);

const start = ()=>{
    try{
        dbCon("mongodb://localhost:27017/Shopping");
        const server = app.listen(8000,()=>{
            console.log("Server is listening on port : " +
server.address().port);
        });
    }
    catch(err)
    {
        console.log(err);
    }
}
start();
```

-mongoDb connection:

```
const mongoose  = require("mongoose");

const connection = (url)=>{
    try{
        mongoose.connect(url);
        console.log("mongoDb Connected...");
    }
    catch(err)
    {
        console.log(err);
    }
};

module.exports = connection;
```

-Models

-user:

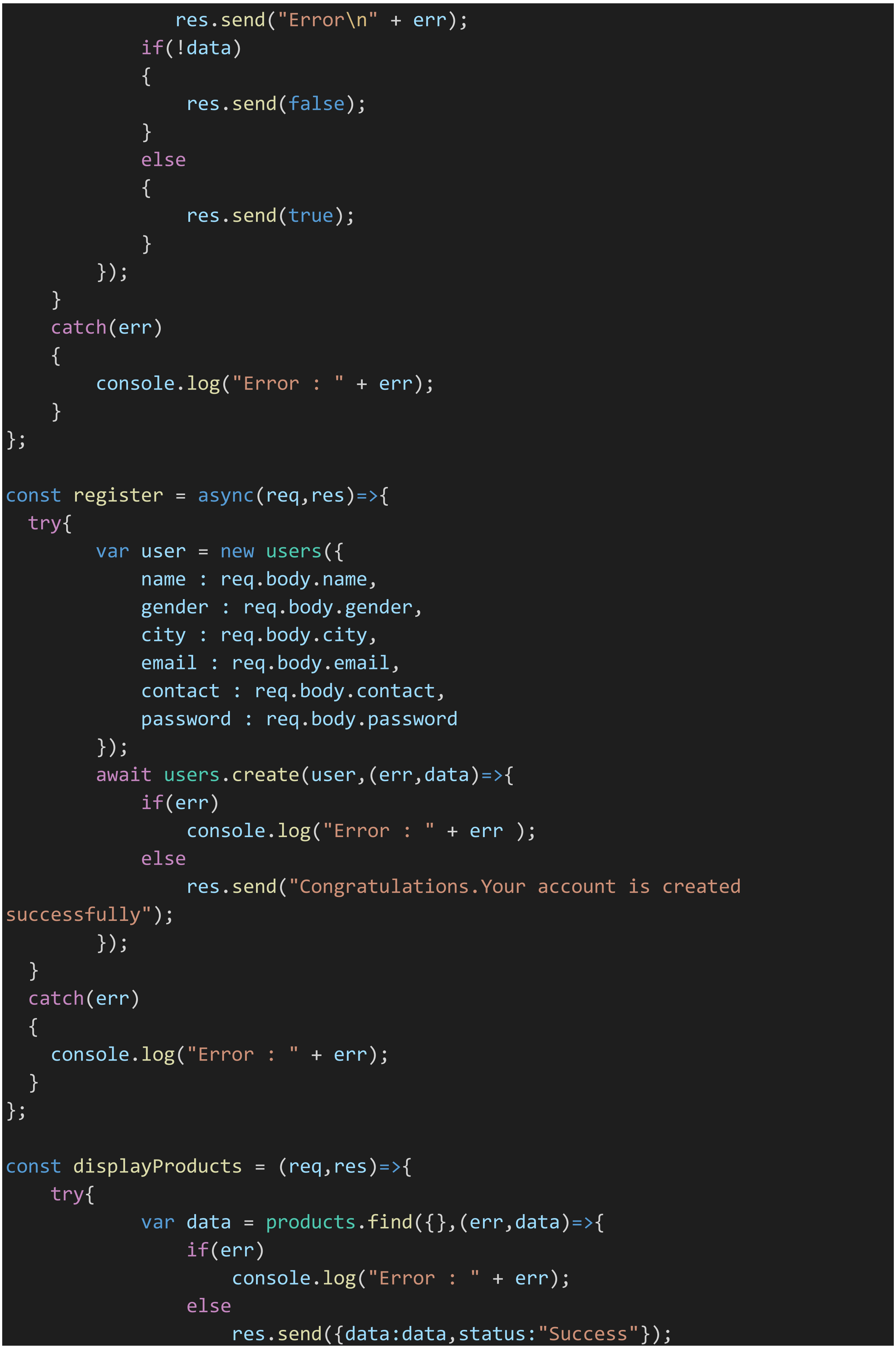
```
const mongoose = require("mongoose");

const userSchema = mongoose.Schema({
    name :{
        type:String
    },
    gender : {
        type : String
    },
    city : {
        type :String
    },
    email : {
        type :String
    },
    contact : {
        type :Number
    },
    password : {
        type :String
    }
});

const userModel = mongoose.model("users",userSchema);
module.exports = userModel;
```

-product:

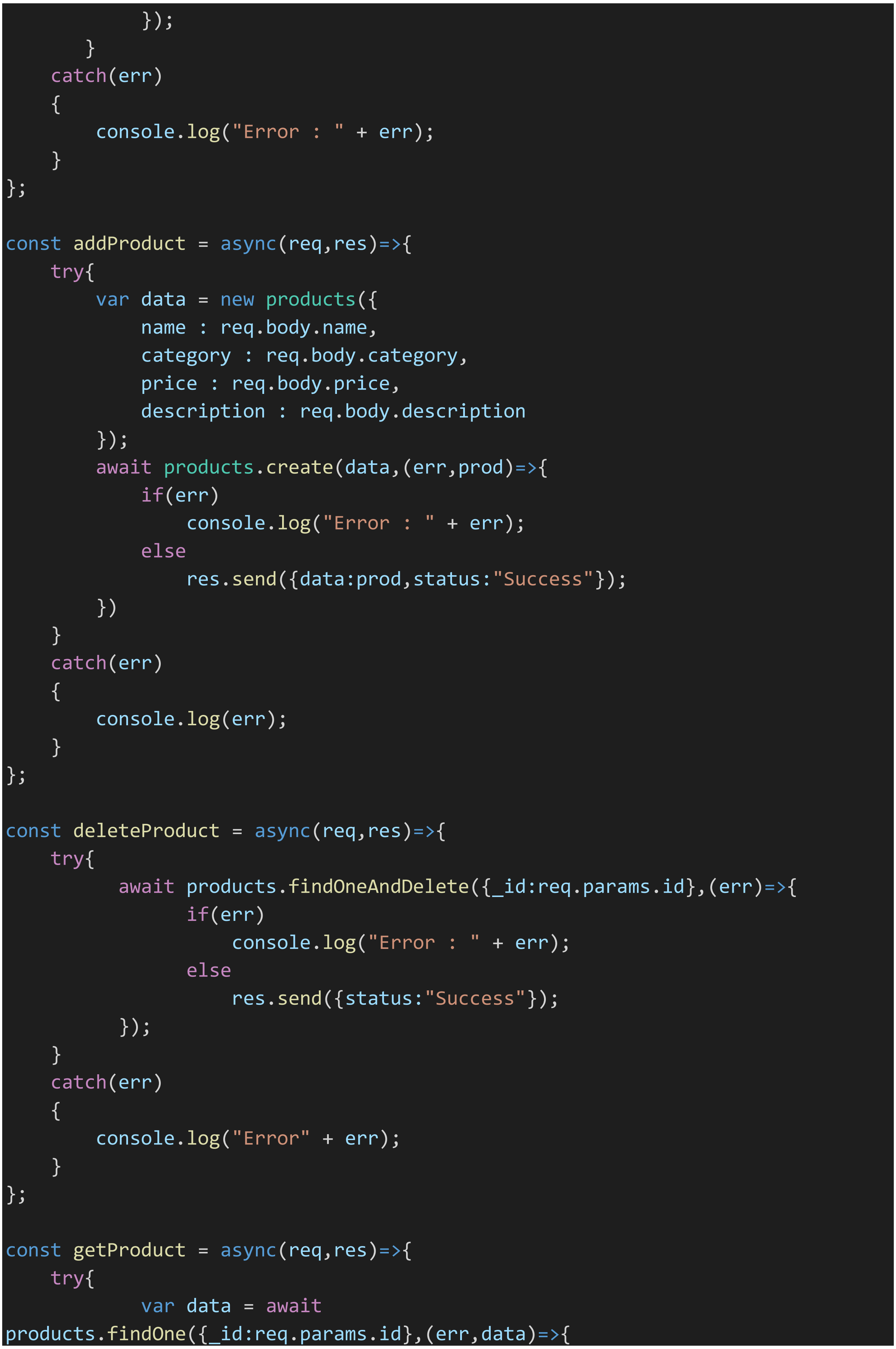
```
const mongoose = require("mongoose");

const prodSchema = mongoose.Schema({
    name :{
        type:String
    },
    category : {
        type : String
    },
    price : {
        type : Number
    },
    description : {
        type :String
    }
});

const prodModel = mongoose.model("products",prodSchema);
module.exports = prodModel;
```

-Controller:

```
const users = require("../Models/UserModel");
const products = require("../Models/ProductModel");

const login = async(req,res)=>{
    try{
        await
users.findOne({"email":req.body.email,"password":req.body.password},(err,data)
=>{
            if(err)
```



```
               res.send("Error\n" + err);
            if(!data)
            {
                res.send(false);
            }
            else
            {
                res.send(true);
            }
        });
    }
    catch(err)
    {
        console.log("Error : " + err);
    }
};

const register = async(req,res)=>{
  try{
        var user = new users({
            name : req.body.name,
            gender : req.body.gender,
            city : req.body.city,
            email : req.body.email,
            contact : req.body.contact,
            password : req.body.password
        });
        await users.create(user,(err,data)=>{
            if(err)
                console.log("Error : " + err );
            else
                res.send("Congratulations.Your account is created 
successfully");
        });
  }
  catch(err)
  {
    console.log("Error : " + err);
  }
};

const displayProducts = (req,res)=>{
    try{
            var data = products.find({},(err,data)=>{
                if(err)
                    console.log("Error : " + err);
                else
                    res.send({data:data,status:"Success"});
```



```
            });
       }
    catch(err)
    {
        console.log("Error : " + err);
    }
};

const addProduct = async(req,res)=>{
    try{
        var data = new products({
            name : req.body.name,
            category : req.body.category,
            price : req.body.price,
            description : req.body.description
        });
        await products.create(data,(err,prod)=>{
            if(err)
                console.log("Error : " + err);
            else
                res.send({data:prod,status:"Success"});
        })
    }
    catch(err)
    {
        console.log(err);
    }
};

const deleteProduct = async(req,res)=>{
    try{
          await products.findOneAndDelete({_id:req.params.id},(err)=>{
                if(err)
                    console.log("Error : " + err);
                else
                    res.send({status:"Success"});
          });
    }
    catch(err)
    {
        console.log("Error" + err);
    }
};

const getProduct = async(req,res)=>{
    try{
            var data = await
products.findOne({_id:req.params.id},(err,data)=>{
```

```
            if(err)
               res.send("Error\n" + err);
            if(err)
            {
                res.send("Product not found....");
            }
            else
            {
                res.send({data:data,message:"Success"});
            }
        });
    }
    catch(err)
    {
        console.log("Error : " + err);
    }
};

const updateProduct = async(req,res)=>{
    try{
        await
products.findOneAndUpdate({_id:req.params.id},req.body,{new:true},(err,data)=>
{
                if(err)
                    console.log("Error : " + err);
                else
                    res.send({data:data,status:"Success"});
        });
    }
    catch(err)
    {
        console.log("Error : " + err);
    }
};
module.exports =
{login,register,displayProducts,addProduct,deleteProduct,getProduct,updateProd
uct};
```

-Route:

```
const ShopRoute = require("express").Router();
const
{login,register,displayProducts,addProduct,deleteProduct,getProduct,updateProd
uct} = require("../Controller/Shopping");
```

```
ShopRoute.route("/add").post(register);
ShopRoute.route("/login").post(login);
ShopRoute.route("/").get(displayProducts);
ShopRoute.route("/addProduct").post(addProduct);
ShopRoute.route("/:id").get(getProduct).delete(deleteProduct).put(updateProduct);

module.exports = ShopRoute;
```

Frontend:

Classes:

Users.ts:

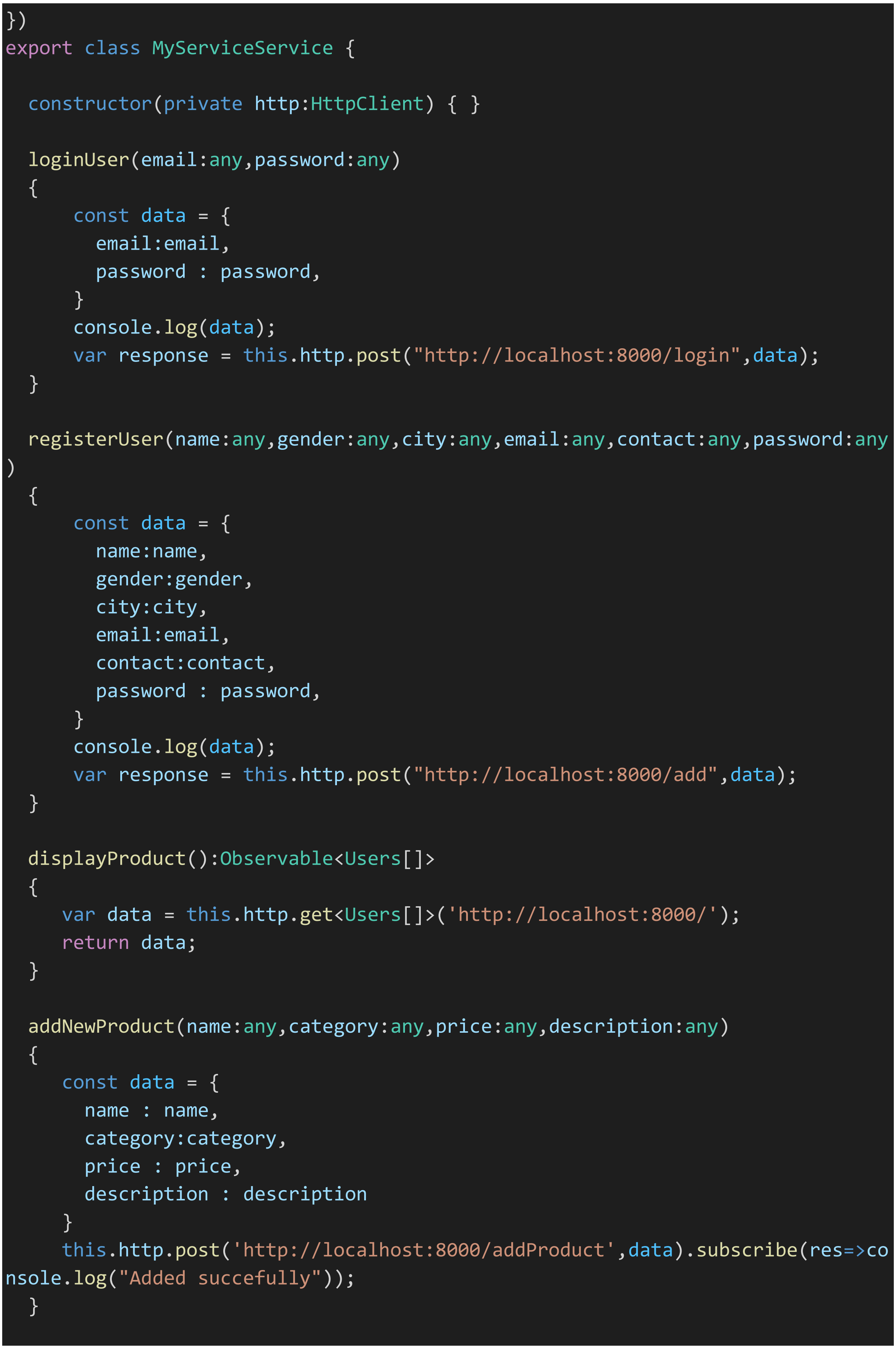
```
export class Users {
    name : string="";
    gender : string="";
    city : string="";
    email : string = "";
    contact : number=0;
    password : string="";
}
```

Products.ts:

```
export class Products {
    name :string="";
    category : string="";
    price : number=0;
    description : string="";
}
```

Service:

```
import { Injectable } from '@angular/core';
import {HttpClient} from '@angular/common/http';
import { Observable } from 'rxjs';
import { Users } from '../Models/users';
import { Products } from '../Models/products';

@Injectable({
  providedIn: 'root'
```



```
})
export class MyServiceService {

  constructor(private http:HttpClient) { }

  loginUser(email:any,password:any)
  {
      const data = {
        email:email,
        password : password,
      }
      console.log(data);
      var response = this.http.post("http://localhost:8000/login",data);
  }

  registerUser(name:any,gender:any,city:any,email:any,contact:any,password:any
)
  {
      const data = {
        name:name,
        gender:gender,
        city:city,
        email:email,
        contact:contact,
        password : password,
      }
      console.log(data);
      var response = this.http.post("http://localhost:8000/add",data);
  }

  displayProduct():Observable<Users[]>
  {
     var data = this.http.get<Users[]>('http://localhost:8000/');
     return data;
  }

  addNewProduct(name:any,category:any,price:any,description:any)
  {
     const data = {
       name : name,
       category:category,
       price : price,
       description : description
     }
     this.http.post('http://localhost:8000/addProduct',data).subscribe(res=>co
nsole.log("Added succefully"));
  }
  deleteProduct(id:any)
  {
     var url = "http://localhost:8000/" + id;
     return this.http.delete(url);
  }

  getOneProduct(id:any):Observable<Products>
  {
     var url = "http://localhost:8000/" + id;
     return this.http.get<Products>(url);
  }

  updateStud(id:any,name:any,category:any,price:any,description:any)
  {
     var updStud = {
      name : name,
      category:category,
      price : price,
      description : description
     };

     var url = "http://localhost:8000/" + id;
     return this.http.put(url,updStud).subscribe(res=>{
       console.log("Record updated successfully");
     });
  }
}
```

-login.component.html

```
<html>
    <head>
        <title>Login</title>
    </head>
    <body>
        <section class="Form my-4 mx-5">
            <div class="container">
                <div class="row  no-gutters">
                    <div class="col-lg-5">
                        <img src="assets/HomeImg.jpeg" alt="Please Wait..."
class="img-fluid">
                    </div>
                    <div class="col-lg-7 px-5 pt-5" >
                        <form [formGroup]="angForm">

                            <div class="form-row">
                                <div class="col-lg-7">
                                    <h1>Login</h1>
                                </div>
                            </div>

                            <div class="form-row">
                                <div class="col-lg-7">
                                    <input type="text" placeholder="Email Address"
formControlName="email" class="form-control">
                                </div>
                            </div>

                            <div class="form-row">
                                <div class="col-lg-7">
                                    <input type="password" placeholder="*****"
formControlName="password" class="form-control">
                                </div>
                            </div>

                            <div class="form-row">
                                <div class="col-lg-7">
                                    <input type="submit" value="Login" class="btn1 btn
btn-primary my-">
                                </div>
                            </div>

                            <p>Don't have an account??<br/>
                            <a href="#" class="link-primary">Register here</a></p>
                        </form>
                    </div>
                </div>
            </div>
        </section>
    </body>
</html>
```

-registration.component.html:

```
<html>
    <head>
        <title>Login</title>
    </head>
    <body>
        <section class="Form my-4 mx-5 ">
            <div class="container">
                <div class="row  no-gutters">
                    <div class="col-lg-7 px-5 pt-5" >
                        <form [formGroup]="angForm">

                            <div class="form-row">
                                <div class="col-lg-7">
                                    <h1>Create new account</h1>
                                </div>
                            </div>
                            <div class="form-row">
                                <div class="col-lg-7 ">
                                    <label>Name</label>
                                    <input type="text" placeholder="Name"
formControlName="name" class="form-control">
                                </div>
                            </div>

                            <div class="form-row ">
                                <div class="col-lg-7">
                                    <label>Gender</label><br>
                                    <input type="radio" value="male"
formControlName="gender" name="gender">Male
                                    <input type="radio" value="female"
formControlName="gender" name="gender">Female
                                    <input type="radio" value="other"
formControlName="gender" name="gender">Other
                                </div>
                            </div>

                            <div class="form-row ">
                                <div class="col-lg-7">
                                    <label>City</label>
                                    <input type="text" placeholder="City"
formControlName="city" class="form-control">
                                </div>
                            </div>

                            <div class="form-row">
                                <div class="col-lg-7">
                                    <label>Email</label>
                                    <input type="email" placeholder="Email ID"
formControlName="email" class="form-control">
                                </div>
                            </div>

                            <div class="form-row ">
                                <div class="col-lg-7">
                                    <label>Contact Number</label>
                                    <input type="number" placeholder="Contact Number"
formControlName="contact" class="form-control">
                                </div>
                            </div>

                            <div class="form-row">
                                <div class="col-lg-7">
                                    <label>Password</label>
                                    <input type="password" placeholder="*****"
formControlName="password" class="form-control">
                                </div>
                            </div>

                            <div class="form-row">
                                <div class="col-lg-7">
```

```
                                    <input type="submit" value="Login" class="btn1 btn
btn-primary my-">
                                </div>
                            </div>
                            <br/>

                        </form>
                    </div>
                    <div class="col-lg-5">
                        <img src="assets/HomeImg.jpeg" alt="Please Wait..."
class="img-fluid">
                    </div>
                </div>
            </div>
        </section>

    </body>
</html>
```

//39 Aayurshi Rana//