

angular.json

```
{
  "$schema": "./node_modules/@angular/cli/lib/config/schema.json",
  "version": 1,
  "newProjectRoot": "projects",
  "projects": {
    "ProjectYoga": {
      "projectType": "application",
      "schematics": {
        "@schematics/angular:application": {
          "strict": true
        }
      },
      "root": "",
      "sourceRoot": "src",
      "prefix": "app",
      "architect": {
        "build": {
          "builder": "@angular-devkit/build-angular:browser",
          "options": {
            "outputPath": "dist/ProjectYoga",
            "index": "src/index.html",
            "main": "src/main.ts",
            "polyfills": "src/polyfills.ts",
            "tsConfig": "tsconfig.app.json",
            "assets": [
              "src/favicon.ico",
              "src/assets"
            ],
            "styles": [
              "node_modules/bootstrap/dist/css/bootstrap.css",
              "src/styles.css"
            ],
            "scripts": [
              "node_modules/bootstrap/dist/js/bootstrap.js"
            ]
          },
          "configurations": {
            "production": {
              "budgets": [
                {
                  "type": "initial",
                  "maximumWarning": "5mb",
                  "maximumError": "5mb"
                },
                {
                  "type": "anyComponentStyle",
                  "maximumWarning": "2kb",
                  "maximumError": "4kb"
                }
              ]
            },
            "fileReplacements": [
              {
                "replace": "src/environments/environment.ts",
                "with": "src/environments/environment.prod.ts"
              }
            ],
            "outputHashing": "all"
          },
          "development": {
            "buildOptimizer": false,
            "optimization": false,
```

```

        "vendorChunk": true,
        "extractLicenses": false,
        "sourceMap": true,
        "namedChunks": true
    },
    },
    "defaultConfiguration": "production"
},
"serve": {
    "builder": "@angular-devkit/build-angular:dev-server",
    "configurations": {
        "production": {
            "browserTarget": "ProjectYoga:build:production"
        },
        "development": {
            "browserTarget": "ProjectYoga:build:development"
        }
    },
    "defaultConfiguration": "development"
},
"extract-il8n": {
    "builder": "@angular-devkit/build-angular:extract-il8n",
    "options": {
        "browserTarget": "ProjectYoga:build"
    }
},
"test": {
    "builder": "@angular-devkit/build-angular:karma",
    "options": {
        "main": "src/test.ts",
        "polyfills": "src/polyfills.ts",
        "tsConfig": "tsconfig.spec.json",
        "karmaConfig": "karma.conf.js",
        "assets": [
            "src/favicon.ico",
            "src/assets"
        ],
        "styles": [
            "src/styles.css"
        ],
        "scripts": []
    }
}
}
}
},
"defaultProject": "ProjectYoga"
}

```

backend\app.js

```
const express = require("express");
const cors = require("cors");
const bodyparser = require("body-parser");
const mongodb = require("mongodb").MongoClient;
const app = express();
app.use(cors());
app.use(bodyparser.json());
var db;
mongodb.connect("mongodb://localhost:27017/yogausers?
retryWrites=true&w=majority", (error, result) => {
    if (error) {
        console.log("DB Not Connected");
    }
    else {
        db = result.db("yogausers");
        console.log("DB Connected");
    }
});
app.use((req, res, next) => {
    console.log("Middleware 1");
    next();
});
app.use("/home", (req, res, next) => {
    console.log("Middleware 2");
    next();
});
function verifyUser(req, res, next) {
    console.log("User Verified");
    next();
}
app.get("/", (req, res) => {
    console.log("Index Page");
    res.send("<h1>Welcome to Express</h1>");
});
app.get("/home", verifyUser, (req, res) => {
    console.log("Home Page");
    var data = { a: "hi", b: "hello" };
    res.json(data);
});
app.post("/register", (req, res) => {
    req.body._id = new Date().getTime();
    console.log(req.body);
    db.collection("users").insertOne(req.body, (error, data) => {
        if (error) {
            res.status(403).json("Error in Inserting Doc");
        }
        else {
            res.json("User Registered Successfully!");
        }
    })
});
app.post("/login", (req, res) => {
    console.log(req.body);
    db.collection("users").find(req.body, { projection: { _id: 1, Fname: 1,
lname: 1, userage: 1, gender: 1 } }).toArray((error, data) => {
        if (error) {
            res.status(403).json("Error");
        }
    })
});
```

```

        else {
            res.json(data);
        }
    });
});
app.get("/allusers", (req, res) => {
    db.collection("users").find(null, { projection: { userpass:
0 } }).toArray((error, data) => {
        if (error) {
            res.status(403).json("Error in Finding the Doc");
        }
        else {
            res.json(data);
        }
    });
});
app.get("/uemailcheck/:uemail", (req, res) => {
    console.log(req.params.uemail);
    db.collection("users").find({ uemail: req.params.uemail }, { projection:
{ _id: 1 } }).toArray((error, data) => {
        if (error) {
            res.status(403).json("Error in Finding the Doc");
        }
        else {
            res.json(data);
        }
    });
});
app.get("/uemailcheck/:uemail", (req, res) => {
    console.log(req.params.uemail);
    db.collection("users").find({ uemail: req.params.uemail }, { projection:
{ _id: 1 } }).toArray((error, data) => {
        if (error) {
            res.status(403).json("Error in Finding the Doc");
        }
        else {
            res.json(data);
        }
    });
});
app.get("/getuser/:userid", (req, res) => {
    console.log(req.params.userid);
    db.collection("users").find({ _id:
Number(req.params.userid) }).toArray((error, data) => {
        if (error) {
            res.status(403).json("Error in Finding the Doc");
        }
        else {
            res.json(data);
        }
    });
});
app.get("/profile/:userid", (req, res) => {
    console.log(req.params.userid);
    db.collection("users").find({ _id: Number(req.params.userid) },
{ projection: { userpass: 0 } }).toArray((error, data) => {
        if (error) {
            res.status(403).json("Error in Finding the Doc");
        }
        else {
            res.json(data);
        }
    });
});

```

```

    }
  });
});
app.put("/cart", (req, res) => {
  console.log(req.body);
  var condition = { _id: req.body._id };
  var newValues = { $set: { date:req.body.date, pack:req.body.pack, level:
req.body.level, isdietplan: req.body.isdietplan, istrain: req.body.istrain,
payment: req.body.payment, isbath: req.body.isbath , paid:req.body.paid,
coupon:req.body.coupon } }
  db.collection("users").update(condition, newValues, (error, data) => {
    if (error) {
      res.status(403).json("Error in Finding the Doc");
    }
    else {
      res.json("User data updated successfully");
    }
  })
});
app.put("/updateuser", (req, res) => {
  console.log(req.body);
  var condition = { _id: req.body._id };
  var newValues = { $set: { Fname: req.body.Fname, lname: req.body.lname,
usage: req.body.usage, uemail: req.body.uemail, level:req.body.level,
gender:req.body.gender, pack:req.body.pack ,isbath:req.body.isbath,
isdietplan:req.body.isdietplan, istrain:req.body.istrain,
date:req.body.date} }
  db.collection("users").update(condition, newValues, (error, data) => {
    if (error) {
      res.status(403).json("Error in Finding the Doc");
    }
    else {
      res.json("User data updated successfully");
    }
  })
});
app.delete("/deleteuser/:userid", (req, res) => {
  console.log(req.params);
  db.collection("users").deleteOne({ _id: Number(req.params.userid) },
(error, data) => {
    res.json("Users deleted successfully");
  });
});
app.get("/searchuser/:searchtxt?", (req, res) => {
  console.log(req.params.searchtxt);
  if (req.params.searchtxt != undefined) {
    var search = new RegExp(req.params.searchtxt, "i");
    var searchCond = { Fname: search };
  }
  else {
    var searchCond = null;
  }
  db.collection("users").find(searchCond).toArray((error, data) => {
    res.json(data);
  });
});
module.exports = app;

```

karma.conf.js

```
module.exports = function (config) {
  config.set({
    basePath: '',
    frameworks: ['jasmine', '@angular-devkit/build-angular'],
    plugins: [
      require('karma-jasmine'),
      require('karma-chrome-launcher'),
      require('karma-jasmine-html-reporter'),
      require('karma-coverage'),
      require('@angular-devkit/build-angular/plugins/karma')
    ],
    client: {
      jasmine: {
      },
      clearContext: false
    },
    jasmineHtmlReporter: {
      suppressAll: true
    },
    coverageReporter: {
      dir: require('path').join(__dirname, './coverage/ProjectYoga'),
      subdir: '.',
      reporters: [
        { type: 'html' },
        { type: 'text-summary' }
      ]
    },
    reporters: ['progress', 'kjhtml'],
    port: 9876,
    colors: true,
    logLevel: config.LOG_INFO,
    autoWatch: true,
    browsers: ['Chrome'],
    singleRun: false,
    restartOnFileChange: true
  });
};
```

package.json

```
{
  "name": "project-yoga",
  "version": "0.0.0",
  "scripts": {
    "ng": "ng",
    "start": "ng serve",
    "build": "ng build",
    "watch": "ng build --watch --configuration development",
    "test": "ng test"
  },
  "private": true,
  "dependencies": {
    "@angular/animations": "~12.2.0",
    "@angular/common": "~12.2.0",
    "@angular/compiler": "~12.2.0",
    "@angular/core": "~12.2.0",
    "@angular/forms": "~12.2.0",
    "@angular/platform-browser": "~12.2.0",
    "@angular/platform-browser-dynamic": "~12.2.0",
    "@angular/router": "~12.2.0",
    "body-parser": "^1.19.0",
    "bootstrap": "^5.1.3",
    "cors": "^2.8.5",
    "express": "^4.18.3",
    "git": "^0.1.5",
    "mangodb": "^1.0.0",
    "mongodb": "^3.7.4",
    "ngx-webstorage-service": "^4.1.0",
    "nodemon": "^2.0.14",
    "rxjs": "~6.6.0",
    "tslib": "^2.3.0",
    "zone.js": "~0.11.4"
  },
  "devDependencies": {
    "@angular-devkit/build-angular": "~12.2.9",
    "@angular/cli": "~12.2.9",
    "@angular/compiler-cli": "~12.2.0",
    "@types/jasmine": "~3.8.0",
    "@types/node": "^12.11.1",
    "jasmine-core": "~3.8.0",
    "karma": "~6.3.0",
    "karma-chrome-launcher": "~3.1.0",
    "karma-coverage": "~2.0.3",
    "karma-jasmine": "~4.0.0",
    "karma-jasmine-html-reporter": "~1.7.0",
    "typescript": "~4.3.5"
  }
}
```


README.md

```
# YOGA FITNESS CENTER WEB APPLICATION
#### 1. Single Page Application (Angular Routing) and Mobile Responsive
(Bootstrap).
#### 2. Tech Stack: MEAN Stack (MongoDB, Express.js, Angular, Node.js).
#### 3. The user has to register and login to buy packages , Admin user can
do member management.
### YogaFitnessCenter
This project is generated with [Angular CLI](https:
#### Development server
Run `ng serve` for a dev server. Navigate to `http://localhost:4200/`. The
app will automatically reload if you change any of the source files.
#### Code scaffolding
Run `ng generate component component-name` to generate a new component. You
can also use `ng generate directive|pipe|service|class|guard|interface|enum|
module`.
#### Build
Run `ng build` to build the project. The build artifacts will be stored in
the `dist/` directory.
#### Running unit tests
Run `ng test` to execute the unit tests via [Karma](https:
#### Running end-to-end tests
Run `ng e2e` to execute the end-to-end tests via a platform of your choice.
To use this command, you need to first add a package that implements end-to-
end testing capabilities.
#### Further help
To get more help on the Angular CLI use `ng help` or go check out the
[Angular CLI Overview and Command Reference](https:
# TO RUN THE YOGA FITNESS CENTER APPLICATION
## 1. Prerequisites
Before we begin, ensure that you have the following tools and knowledge:
#### 1. Node.js:
Install Node.js from the official website (https:
#### 2. npm (Node Package Manager):
npm comes bundled with Node.js and is used to manage packages and
dependencies for your project.
#### 3. MongoDB:
Install MongoDB from the official website (https:
#### 4. Angular CLI:
Install Angular CLI globally using npm. You can do this by running `npm
install -g @angular/cli`. Angular is a front-end framework for building
dynamic web applications.
Now that you have the necessary tools in place, let's get started with
building your MEAN stack application!
## 2. Install node modules
> npm install
## 3. Run Application
1. open command prompt and run below command
> npm run start
2. open another command prompt and run below command
> node server.js
```

server.js

```
const http=require("http");
const app =require("./backend/app");
const server =http.createServer(app);
server.listen(3000, ()=>{
  console.log("check 3000");
});
```

src\app\adduser\adduser.component.html

```
<div class="container" style="min-height: 560px; font-family: 'Big Shoulders
Stencil Display', cursive;
font-family: 'Londrina Solid', cursive;">
  <div class="row justify-content-center" style="padding-top: 4%;">
    <div class="col-lg-5">
      <div class="card shadow-lg border-0 rounded-lg mt-5">
        <div class="card-header " style="background-color: #c5c6c7;">
          <h2 style="text-align: center; font-family: 'Big Shoulders Stencil
Display', cursive;
font-family: 'Londrina Solid',
cursive;">Add members</h2>
          <h5>{{msg}}</h5>
          <div class="card-body shadow-lg" style="background-color:
#1F2833;">
            <form (submit)="addUser(f)" #f="ngForm">
              <div class="row">
                <div class="col">
                  <div class="form-floating mb-3">
                    <input class="form-control" id="inputfname"
#firstname="ngModel" name="Fname" ngModel type="text"
placeholder="fname" required minlength="3" />
                    <label for="fname">First Name</label>
                  </div>
                </div>
                <div class="col">
                  <div class="form-floating mb-3">
                    <input class="form-control" id="inputlname"
#lastname="ngModel" name="lname" type="text"
placeholder="lname" ngModel required />
                    <label for="lname">Last Name</label>
                  </div>
                </div>
              </div>
              <p class="alert alert-danger" *ngIf="firstname.errors?.required
&& firstname.touched">Please enter the
First Name</p>
              <p class="alert alert-danger"
*ngIf="firstname.errors?.minlength && firstname.touched">Please enter Min 3
characters </p>
              <div class="row">
                <div class="col">
                  <div class="form-floating mb-3">
                    <input class="form-control" id="age" type="number"
placeholder="age" name="userage" ngModel
#userage="ngModel" required />
                    <label for="age">Age</label>
                  </div>
                  <p class="alert alert-danger"
*ngIf="userage.errors?.required && userage.touched">Please enter the age
</p>
                </div>
                <div class="col">
                  <div class="form-check">
                    <input class="form-check-input" type="radio"
name="gender" value="male" id="flexRadioDefault1"
ngModel #usergender="ngModel" required>
                    <label class="form-check-label" for="flexRadioDefault1"
style="background-color: whitesmoke; border-radius:
```

```

3px; width: 50%; text-align: center; ">
        Male
    </label>
</div>
    <div class="form-check" style="margin-top: 10px; ">
        <input class="form-check-input" type="radio"
name="gender" value="female" id="flexRadioDefault2"
        ngModel #usergender="ngModel" required>
        <label class="form-check-label" for="flexRadioDefault2"
        style="background-color: whitesmoke; border-radius:
3px; width: 50%; text-align: center; ">
            Female
        </label>
    </div>
</div>
    <p class="alert alert-danger"
*ngIf="usergender.errors?.required && usergender.touched">Please click the
        gender </p>
</div>
    <div class="form-floating mb-3">
        <input class="form-control" id="inputphone" type="text"
placeholder="Phone Number" name="userphone"
        ngModel #userphone="ngModel" required pattern="^(0|[1-9]
[0-9]*)$" maxlength="10" />
        <label for="inputphone">Phone Number</label>
    </div>
    <p class="alert alert-danger" *ngIf="userphone.errors?.required
&& userphone.touched">Please enter the
        Phone number</p>
    <p class="alert alert-danger"
*ngIf="userphone.errors?.pattern">please enter the valid phone number</p>
    <div class="form-floating mb-3">
        <input class="form-control"
id="inputEmail" (blur)="uemailcheck(uemail.value)" name="uemail"
        type="email" ngModel #uemail="ngModel"
placeholder="name@example.com" required email />
        <label for="inputEmail">Email address</label>
    </div>
    <p class="alert alert-danger" *ngIf="uemail.errors?.required &&
uemail.touched">Please enter the Email</p>
    <p class="alert alert-danger" *ngIf="uemail.errors?.email &&
uemail.touched">Please enter the valid Email
    </p>
    <div class="form-floating mb-3">
        <input class="form-control" id="inputPassword"
type="password" placeholder="Password" name="userpass"
        ngModel #userpass="ngModel" required password
minlength="8" />
        <label for="inputPassword">Create Password</label>
    </div>
    <p class="alert alert-danger" *ngIf="userpass.errors?.required
&& userpass.touched">Please Create a
        password</p>
    <p class="alert alert-danger"
*ngIf="userpass.errors?.minlength">Please Create a strong password (min 8
        character) </p>
    <h4 style="color:; margin: 1vw; display: flex; justify-
content: center; ">Package</h4>
    <select class="form-select form-select-lg mb-3" name="pack"
ngModel aria-label=".form-select-lg example"
        placeholder="Select Package">

```

```

        <option selected>Select Package</option>
        <option value="Hatha Yoga">Hatha Yoga</option>
        <option value="Vinyasa Yoga">Vinyasa Yoga</option>
        <option value="Iyengar Yoga">Iyengar Yoga</option>
        <option value="Ashtanga Yoga">Ashtanga Yoga</option>
        <option value="Bikram Yoga">Bikram Yoga</option>
        <option value="Kundalini Yoga">Kundalini Yoga</option>
    </select>
    <h4 style="color: pink; margin: 1vw; display: flex; justify-
content: center;">Level</h4>
    <div class="form-check" style="display: inline-block; color:
#c5c6c7; margin-left: 2vw; ">
        <input class="form-check-input" type="radio" name="level"
#level="ngModel" id="gridRadios1"
        value="Beginner" checked ngModel >
        <label class="form-check-label" for="level">
            Beginner
        </label>
    </div>
    <div class="form-check" style="display: inline-block; color:
#c5c6c7; margin-left: 2vw; ">
        <input class="form-check-input" type="radio" name="level"
#level="ngModel" id="gridRadios2"
        value="Intermediate" ngModel>
        <label class="form-check-label" for="level">
            Intermediate
        </label>
    </div>
    <div class="form-check" style="display: inline-block; color:
#c5c6c7; margin-left: 2vw; ">
        <input class="form-check-input" type="radio" name="level"
#level="ngModel" id="gridRadios3"
        value="Advanced" ngModel>
        <label class="form-check-label" for="level">
            Advanced
        </label>
    </div>
    <div>
        <h4 style="color: pink; margin: 1vw; display: flex; justify-
content: center;">Adds on</h4>
        <div class="form-check" style="display: inline-block; color:
#c5c6c7; margin-left: 2vw; ">
            <input class="form-check-input" type="checkbox"
id="flexCheckDefault" ngModel value="Steambath"
            name="isbath">
            <label class="form-check-label" for="flexCheckDefault">
                Steambath
            </label>
        </div>
        <div class="form-check" style="display: inline-block; color:
#c5c6c7; margin-left: 2vw; ">
            <input class="form-check-input" type="checkbox" ngModel
value="DietPlan" name="isdietplan"
            id="flexCheckDefault">
            <label class="form-check-label" for="flexCheckDefault">
                DietPlan
            </label>
        </div>
        <div class="form-check" style="display: inline-block; color:
#c5c6c7; margin-left: 2vw; ">
            <input class="form-check-input" type="checkbox"

```

```

id="flexCheckDefault" ngModel value="PersonalTrainer"
    name="istrain">
    <label class="form-check-label" for="flexCheckDefault">
    PersonalTrainer
    </label>
  </div>
</div>
<div class="d-flex align-items-center justify-content-between
mt-4 mb-0">
    <button class="btn btn-info btn-block" value="Register"
type="submit"
    [disabled]="f.invalid || !useremailavail" style="text-
align: center; font-family: 'Big Shoulders Stencil Display', cursive;
font-family:
'Londrina Solid', cursive;background-color: pink!important; border-color:
pink !important;"> Register</button> <h5 style="color: pink;">{{msg}}</h5>
    </div>
  </form>
</div>
<div class="card-footer text-center py-3">
  <p class="card-text"><small class="text-muted">Yoga fitness
center</small></p>
  <button class="btn btn-info me-md-2" type="button" routerLink="/
manager" style="border:solid #1F2833; background-color: pink;margin:
1vw;">Back
  </button>
</div>
</div>
</div>
</div>
</div>
</div>
</div>

```

src/app/adduser/adduser.component.spec.ts

```
import { ComponentFixture, TestBed } from '@angular/core/testing';
import { AdduserComponent } from './adduser.component';
describe('AdduserComponent', () => {
  let component: AdduserComponent;
  let fixture: ComponentFixture<AdduserComponent>;
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [ AdduserComponent ]
    })
    .compileComponents();
  });
  beforeEach(() => {
    fixture = TestBed.createComponent(AdduserComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });
  it('should create', () => {
    expect(component).toBeTruthy();
  });
});
```

src\app\adduser\adduser.component.ts

```
import { Component, OnInit } from '@angular/core';
import { NgForm } from '@angular/forms';
import { UsersService } from '../users.service';
@Component({
  selector: 'app-adduser',
  templateUrl: './adduser.component.html'
})
export class AdduserComponent implements OnInit {
  msg:string;
  useremailavail=false;
  date = new Date();
  constructor(public userSer :UsersService) { }
  ngOnInit(): void {
  }
  addUser(form:NgForm){
    form.value.date = this.date;
    this.userSer.doUserRegistration(form.value).subscribe((data:string)=>{
      console.log(data);
      this.msg = data;
      form.reset();
    }, (error:any)=>{
      console.log(error);
      this.msg = "Something Went Wrong!!";
    });
  }
  uemailcheck(uemail:string)
  {
    this.userSer.uemailcheckAvail(uemail).subscribe((data:any[])=>{
      console.log(data);
      if(data.length==0)
      {
        this.msg=""
        this.useremailavail= true;
      }
      else{
        this.msg="Email already registered";
        this.useremailavail= false;
      }
    },(error:any)=>{
      console.log(error);
      this.msg="Something went wrong";
    })
  }
}
```


src/app/admin.guard.spec.ts

```
import { TestBed } from '@angular/core/testing';
import { AdminGuard } from '../admin.guard';
describe('AdminGuard', () => {
  let guard: AdminGuard;
  beforeEach(() => {
    TestBed.configureTestingModule({});
    guard = TestBed.inject(AdminGuard);
  });
  it('should be created', () => {
    expect(guard).toBeTruthy();
  });
});
```

src/app/admin.guard.ts

```
import { Injectable } from '@angular/core';
import { CanActivate, Router } from '@angular/router';
import { UsersService } from '../users.service';
@Injectable({
  providedIn: 'root'
})
export class AdminGuard implements CanActivate {
  user:string=localStorage.getItem("loggeduser");
  constructor(public userSer : UsersService, public myRouter : Router)
  {
  }
  canActivate(): boolean {
    if(!this.userSer.isAdmin())
    {
      this.myRouter.navigateByUrl("/");
    }
    return this.userSer.isAdmin();
  }
}
```

src\app\app-routing.module.ts

```
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { AuthGuard } from '../auth.guard';
import { AdminGuard } from '../admin.guard';
import { logGuard } from '../log.guard';
import { LoginComponent } from '../login/login.component';
import { MempackagesComponent } from '../mempackages/mempackages.component';
import { ProfileComponent } from '../profile/profile.component';
import { RegisterComponent } from '../register/register.component';
import { SlideComponent } from '../slide/slide.component';
import { NotfoundComponent } from '../notfound/notfound.component';
import { CartComponent } from '../cart/cart.component';
import { PaymentComponent } from '../payment/payment.component';
import { ManagerComponent } from '../manager/manager.component';
import { AdduserComponent } from '../adduser/adduser.component';
import { EdituserComponent } from '../edituser/edituser.component';
import { ContactComponent } from '../contact/contact.component';
const routes: Routes = [
  {path:"",component:SlideComponent},
  {path:"register", component:RegisterComponent,canActivate:[logGuard]},
  {path:"contact",component:ContactComponent},
  {path:"login", component:LoginComponent ,canActivate:[logGuard]},
  {path:"profile", component:ProfileComponent, canActivate:[AuthGuard]},
  {path:"packages", component:MempackagesComponent},
  {path:"cart", component:CartComponent, canActivate:[AuthGuard]},
  {path:"payment", component:PaymentComponent, canActivate:[AuthGuard]},
  {path:"manager", component:ManagerComponent, canActivate:[AdminGuard]},
  {path:"edituser/:userid",component:EdituserComponent,canActivate:
[AdminGuard]},
  {path:"adduser", component:AdduserComponent,canActivate:[AdminGuard]},
  {path:"**",component:NotfoundComponent}
];
@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```

src\app\app.component.css

```
.body{
  font-family: 'Big Shoulders Stencil Display', cursive; font-family:
'Londrina Solid', cursive;
}
```

src\app\app.component.html

```
<app-navmain></app-navmain>  
<router-outlet></router-outlet>  
<app-marq></app-marq>  
<app-footer1></app-footer1>
```

src\app\app.component.spec.ts

```
import { TestBed } from '@angular/core/testing';
import { RouterTestingModule } from '@angular/router/testing';
import { AppComponent } from './app.component';
describe('AppComponent', () => {
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      imports: [
        RouterTestingModule
      ],
      declarations: [
        AppComponent
      ],
    }).compileComponents();
  });
  it('should create the app', () => {
    const fixture = TestBed.createComponent(AppComponent);
    const app = fixture.componentInstance;
    expect(app).toBeTruthy();
  });
  it(`should have as title 'ProjectYoga'`, () => {
    const fixture = TestBed.createComponent(AppComponent);
    const app = fixture.componentInstance;
    expect(app.title).toEqual('ProjectYoga');
  });
  it('should render title', () => {
    const fixture = TestBed.createComponent(AppComponent);
    fixture.detectChanges();
    const compiled = fixture.nativeElement as HTMLElement;
    expect(compiled.querySelector('.content
span')?.textContent).toContain('ProjectYoga app is running!');
  });
});
```

src\app\app.component.ts

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

src\app\app.module.ts

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { FormsModule } from '@angular/forms';
import { AppRoutingModuleModule } from './app-routing.module';
import { HttpClientModule } from '@angular/common/http';
import { StorageServiceModule } from 'ngx-webstorage-service';
import { AppComponent } from './app.component';
import { NavmainComponent } from './navmain/navmain.component';
import { SlideComponent } from './slide/slide.component';
import { Footer1Component } from './footer1/footer1.component';
import { MarqComponent } from './marq/marq.component';
import { LoginComponent } from './login/login.component';
import { RegisterComponent } from './register/register.component';
import { ProfileComponent } from './profile/profile.component';
import { NotfoundComponent } from './notfound/notfound.component';
import { MempackagesComponent } from './mempackages/mempackages.component';
import { CartComponent } from './cart/cart.component';
import { PaymentComponent } from './payment/payment.component';
import { ManagerComponent } from './manager/manager.component';
import { AdduserComponent } from './adduser/adduser.component';
import { EdituserComponent } from './edituser/edituser.component';
import { ContactComponent } from './contact/contact.component';
@NgModule({
  declarations: [
    AppComponent,
    NavmainComponent,
    SlideComponent,
    Footer1Component,
    MarqComponent,
    LoginComponent,
    RegisterComponent,
    ProfileComponent,
    NotfoundComponent,
    MempackagesComponent,
    CartComponent,
    PaymentComponent,
    ManagerComponent,
    AdduserComponent,
    EdituserComponent,
    ContactComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModuleModule,
    FormsModule,
    HttpClientModule,
    StorageServiceModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```


src\app\auth.guard.spec.ts

```
import { TestBed } from '@angular/core/testing';
import { AuthGuard } from '../auth.guard';
describe('AuthGuard', () => {
  let guard: AuthGuard;
  beforeEach(() => {
    TestBed.configureTestingModule({});
    guard = TestBed.inject(AuthGuard);
  });
  it('should be created', () => {
    expect(guard).toBeTruthy();
  });
});
```

src\app\auth.guard.ts

```
import { Injectable } from '@angular/core';
import { CanActivate, Router } from '@angular/router';
import { UsersService } from '../users.service';
@Injectable({
  providedIn: 'root'
})
export class AuthGuard implements CanActivate {
  constructor(public userSer : UsersService, public myRouter : Router)
  {
  }
  canActivate(): boolean {
    if(!this.userSer.isLoggedIn())
    {
      this.myRouter.navigateByUrl("/login");
    }
    return this.userSer.isLoggedIn();
  }
}
```

src/app/cart/cart.component.html

```
<div style="font-family: 'Big Shoulders Stencil Display', cursive;
font-family: 'Londrina Solid', cursive; background-color: #c5c6c7 !important;
max-height:200vh; ">
  <div class="d-grid gap-2 d-md-flex justify-content-md-center">
    <button class="btn btn me-md-2" style=" margin: 1vw; color: pink !
important; background-color:#1F2833 ;">
      <h3>CART</h3>
    </button>
    <button class="btn btn-info me-md-2" type="button" routerLink="/
profile"
      style="border:solid #1F2833; background-color: pink;margin:
1vw;">Profile</button>
    <button class="btn btn-info me-md-2" type="button" routerLink="/
packages"
      style="border:solid #1F2833; background-color: pink;margin:
1vw;">Back
    </button>
  </div> <!-- nav bar-->
  <!-- Cart-->
  <hr style="height: 2px; margin: 1%;padding: 0;">
  <div class="row justify-content-center">
    <div class="col-lg-5">
      <div class="card shadow-lg border-0 rounded-lg" style=" border-
radius: 12px;">
        {{msg}}
        <div class="card-header" style="background-color: #c5c6c7;
border-radius: 12px;">
          <h2 style="text-align: center; font-family: 'Big
Shoulders Stencil Display', cursive;
                                font-family: 'Londrina
Solid', cursive;">Check Out</h2>
          <div class="card-body" style="background-color:
whitesmoke; border-radius:12px ;">
            <div *ngIf="isPackbuy">
              <h3 class="bg-info"
                style="display: flex; justify-content:
center; border-radius: 12px; padding:4px; background-color:pink !important ;">
                {{packname}}</h3>
            </div>
            <h5> {{packdesc}} </h5>
            <form (submit)="packCb(fo)" #fo="ngForm"
*ngIf="userdata">
              <fieldset class="form-group">
                <div class="row">
                  <legend class="col-form-label col-sm-2
pt-0">
                    <h5 class="bg-info"
                      style="display: flex; justify-
content: center; border-radius: 12px; padding:4px;background-color:pink !
important ;">
                      Level</h5>
                  </legend>
                  <div class="col-sm-10">
                    <div class="form-check">
                      <input class="form-check-input"
type="radio" name="level" #level="ngModel"
                      id="gridRadios1"
value="Beginner" checked ngModel required>
```

```

for="level">
    <label class="form-check-label"
        Beginner
    </label>
</div>
<div class="form-check">
    <input class="form-check-input"
        id="gridRadios2"
    type="radio" name="level" #level="ngModel"
    value="Intermediate" ngModel>
    <label class="form-check-label"
        Intermediate
    </label>
</div>
<div class="form-check">
    <input class="form-check-input"
        id="gridRadios3"
    type="radio" name="level" #level="ngModel"
    value="Advanced" ngModel>
    <label class="form-check-label"
        Advanced
    </label>
</div>
</div>
</div>
<hr style="margin: 1%; height: 2px;">
</fieldset>
<fieldset class="form-group">
    <div class="row">
        <legend class="col-form-label col-sm-2
pt-0">
            <h5 class="bg-info"
                style="display: flex; justify-
content: center; border-radius: 12px; padding: 4px; background-color: pink !
important ;">
                Add ons</h5>
            </legend>
            <div class="col-sm-10">
                <div class="form-check">
                    <input class="form-check-input"
                        name="isdietplan" ngModel
                    #addson1 (change)="diet($event)" ngModel>
                    <label class="form-check-label"
                        for="defaultCheck2">
                            Diet Plan (&#8377; 500)
                        </label>
                    </div>
                    <div class="form-check">
                        <input class="form-check-input"
                            type="checkbox" value="PersonalTrainer"
                            name="istrain" (change)="train($event)" ngModel>
                        <label class="form-check-label"
                            for="defaultCheck2">
                                Personal Trainer (&#8377;
2000)
                            </label>
                        </div>

```

```

                                <div class="form-check">
                                    <input class="form-check-input"
type="checkbox" value="Steambath"
name="isbath" (change)="bath($event)" ngModel>
                                <label class="form-check-label"
for="defaultCheck2">
                                    steambath (&#8377; 1000)
                                </label>
                                </div>
                            </div>
                        </div>
                        <hr style="margin: 1%; height: 2px;">
                    </fieldset>
                    <fieldset class="form-group">
                        <div class="row">
                            <legend class="col-form-label col-sm-2
pt-0">
                                <h5 class="bg-info"
                                    style="display: flex; justify-
content: center; border-radius: 12px; padding: 4px; background-color: pink !
important ;">
                                    Payment</h5>
                                </legend>
                                <div class="col-sm-10">
                                    <div class="form-check">
                                        <input class="form-check-input"
type="radio" name="payment" id="gridRadios1"
                                        value="Gpay" required ngModel
#payment="ngModel" [(ngModel)]="val" >
                                        <label class="form-check-label"
for="gridRadios1">
                                            Gpay
                                        </label>
                                    </div>
                                    <div class="form-check">
                                        <input class="form-check-input"
type="radio" name="payment" id="gridRadios2"
                                        value="Upi" ngModel
#payment="ngModel" ngModel="val" >
                                        <label class="form-check-label"
for="gridRadios2">
                                            Upi
                                        </label>
                                    </div>
                                    <div class="form-check">
                                        <input class="form-check-input"
type="radio" name="payment" id="gridRadios3"
                                        value="Network Banking"
ngModel #payment="ngModel" [(ngModel)]="val" >
                                        <label class="form-check-label"
for="gridRadios3">
                                            Network Banking
                                        </label>
                                    </div>
                                </div>
                            </div>
                        </div>
                    </fieldset>
                    <div class="row">
                        <div class="col">
                            <input type="text" class="form-control"

```

```

name="coupon" placeholder="Coupon Code"
[ (ngModel) ]="couponcode">
</div>
<div class="col">
<input type="button" class="btn btn-info"
style="background-color: pink !important ;"
Value="Apply" (click)="coupon($event)">
<p style="color: red; display: inline-
block; margin-left: 1vw;">{{codemsg}} {{offer}}</p><p style="color: green;
display: inline-block; margin-left: 1vw; background-color: pink !
important ;">{{appliedmsg}} </p>
</div>
</div>
<div style="margin-top: 1vh; display: flex;
justify-content: flex-end;">
<h3 style="margin: 1vh;"> <span class="text-
info" style="color: pink !important ;">Adds ons + package &nbsp; </span> Total:
{{pay}} </h3>
<button type="submit" class="btn btn-info btn-
lg" style="background-color: pink; border-color: pink;" value="Buy"
[disabled]="fo.invalid">Buy</button>
</div>
</form>
</div>
<div class="card-footer text-center py-3">
<p class="card-text"><small class="text-muted">Yoga
fitness center</small></p>
<div class="small"><a routerLink="/contact"
style="color: #1F2833;">Need Help?</a>
</div>
</div>
</div>
</div>
</div>
</div>

```

src/app/cart/cart.component.spec.ts

```
import { ComponentFixture, TestBed } from '@angular/core/testing';
import { CartComponent } from './cart.component';
describe('CartComponent', () => {
  let component: CartComponent;
  let fixture: ComponentFixture<CartComponent>;
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [ CartComponent ]
    })
    .compileComponents();
  });
  beforeEach(() => {
    fixture = TestBed.createComponent(CartComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });
  it('should create', () => {
    expect(component).toBeTruthy();
  });
});
```

src/app/cart/cart.component.ts

```
import { Component, OnInit } from '@angular/core';
import { PackService } from '../pack.service';
import { NgForm } from '@angular/forms';
import { Router } from '@angular/router';
import { UsersService } from '../users.service';
import { ActivatedRoute, Params } from '@angular/router';
@Component({
  selector: 'app-cart',
  templateUrl: './cart.component.html'
})
export class CartComponent implements OnInit {
  isPackbuy = true;
  code: string = "452dfg";
  pay = 2000;
  userdataa: { _id: number, Fname: string, lname: string, uemail: string,
  userage: Number, usergender: Number, gender: String, userpass: string,
  userphone: number }
  codemsg: string;
  couponcode: string = "";
  packn: string;
  isdiet = false;
  msg: string;
  userdata: any = localStorage.getItem("loggeduser");
  date = new Date();
  offer: string;
  appliedmsg: string;
  taxy: number;
  istax: boolean;
  isnottax: boolean;
  val:string;
  coup:string;
  packname = localStorage.getItem("packna");
  packdesc = localStorage.getItem("packds");
  constructor(public activeRoute: ActivatedRoute, public pac: PackService,
  public userSer: UsersService, public rooter: Router) { this.packn =
  pac.packname }
  ngOnInit(): void {
    this.activeRoute.params.subscribe((param: Params) => {
      console.log(param.userid);
    });
  }
  bath(event: any) {
    if (event.target.checked == true)
      if (this.couponcode == this.code) {
        this.pay += 500;
      }
      else {
        this.pay += 1000;
      }
    else {
      if (this.couponcode == this.code) {
        this.pay -= 500
      }
      else {
        this.pay -= 1000;
      }
    }
  }
}
```



```

train(event: any) {
  if (event.target.checked == true)
    if (this.couponcode == this.code) {
      this.pay += 1000;
    }
    else {
      this.pay += 2000;
    }
  else {
    if (this.couponcode == this.code) {
      this.pay -= 1000;
    }
    else {
      this.pay -= 2000;
    }
  }
}
diet(event: any) {
  if (event.target.checked == true)
    if (this.couponcode == this.code) {
      this.pay += 250;
    }
    else {
      this.pay += 500;
    }
  else {
    if (this.couponcode == this.code) {
      this.pay -= 250;
    }
    else {
      this.pay -= 500;
    }
  }
}
coupon(event: any) {
  if (this.couponcode == this.code) {
    event.target.disabled = true;
    this.pay *= 0.5;
    this.coup="Applied";
    this.appliedmsg = "Coupon Applied";
    this.codemsg = "50%";
  }
  else {
    this.codemsg = "Please Enter valid Code";
  }
}
iscoupon() {
  if (this.couponcode == this.code) {
    return true;
  }
  else {
    return false;
  }
}
packCb(form: NgForm) {
  console.log("Package Registered");
  console.log(form.value);
  var stringToConvert = localStorage.getItem("loggeduser");
  var numberValue = Number(stringToConvert);
  console.log(numberValue);
  form.value._id = numberValue;
}

```

```
form.value.paid= this.pay;
form.value.coupon=this.coup;
form.value.date = this.date;
form.value.pack = this.packn;
this.userSer.addpackages(form.value).subscribe((data: string) => {
    this.msg = data;
    form.reset();
}, (error: any) => {
    this.msg = "something went wrong";
})
this.rooter.navigateByUrl("/payment")
}
```

src\app\contact\contact.component.html

```
<div style="font-family: 'Big Shoulders Stencil Display', cursive;
font-family: 'Londrina Solid', cursive; background-color: #c5c6c7; min-
height:80vh; border-radius: 12px; padding-bottom: 2vh; ">
  <div class="d-grid gap-2 d-md-flex justify-content-md-center">
    <button class="btn btn-info me-md-2" type="button" routerLink="/"
      style="border:solid #1F2833; background-color: pink;margin:
1vw;">Home</button>
    <button class="btn btn-info me-md-2" type="button" routerLink="/"
packages"
      style="border:solid #1F2833; background-color: pink;margin:
1vw;">Package
    </button>
  </div>
  <div class="container-fluid">
    <div class="card shadow-lg" style="background-color: #c5c6c7;">
      <br>
      <div class="card-body">
        <h5 class="card-title " style="color: #1F2833;">Address</h5>
        <p style="color: #1F2833;" class="card-text">Powai,Andhrer East, Maharashtra.</p><hr>
        <h5 class="card-title " style="color: #1F2833;">Contact Us</h5>
        <a class="card-text" >  YogaFitnessCenter@gmail.com <br>  9999999999
        </a>
        <p class="card-text"><small class="text-muted">Last updated on
06/11/2021</small></p>
      </div>
    </div>
  </div>
</div>
```

src/app/contact/contact.component.spec.ts

```
import { ComponentFixture, TestBed } from '@angular/core/testing';
import { ContactComponent } from './contact.component';
describe('ContactComponent', () => {
  let component: ContactComponent;
  let fixture: ComponentFixture<ContactComponent>;
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [ ContactComponent ]
    })
    .compileComponents();
  });
  beforeEach(() => {
    fixture = TestBed.createComponent(ContactComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });
  it('should create', () => {
    expect(component).toBeTruthy();
  });
});
```

src\app\contact\contact.component.ts

```
import { Component, OnInit } from '@angular/core';
@Component({
  selector: 'app-contact',
  templateUrl: './contact.component.html'
})
export class ContactComponent implements OnInit {
  constructor() { }
  ngOnInit(): void {
  }
}
```

src\app\edituser\edituser.component.html

```
<div class="container" style="min-height: 560px; font-family: 'Big Shoulders
Stencil Display', cursive;
font-family: 'Londrina Solid', cursive;">
  <div class="row justify-content-center" style="padding-top: 4%;">
    <div class="col-lg-5">
      <div class="card shadow-lg border-0 rounded-lg mt-5">
        <div class="card-header " style="background-color: #c5c6c7;">
          <h2 style="text-align: center; font-family: 'Big Shoulders Stencil
Display', cursive;
font-family: 'Londrina Solid',
cursive;">Edit Member</h2>
        <div class="card-body shadow-lg" style="background-color:
#1F2833;">
          <form #f="ngForm" (submit)="editUser(f)" *ngIf="userdata">
            <div class="row">
              <div class="col">
                <div class="form-floating mb-3">
                  <input class="form-control" id="inputfname"
#firstname="ngModel" name="Fname" [ngModel]= "userdata.Fname" type="text"
placeholder="fname" required minlength="3" />
                  <label for="fname">First Name</label>
                </div>
              </div>
              <div class="col">
                <div class="form-floating mb-3">
                  <input class="form-control" id="inputlname"
#lastname="ngModel" name="lname" type="text" placeholder="lname" [ngModel]=
"userdata.lname" required />
                  <label for="lname">Last Name</label>
                </div>
              </div>
            </div>
            <p class="alert alert-danger" *ngIf="firstname.errors?.required
&& firstname.touched">Please enter the First Name</p>
            <p class="alert alert-danger"
*ngIf="firstname.errors?.minlength && firstname.touched">Please enter Min 3
characters </p>
            <div class="row">
              <div class="col">
                <div class="form-floating mb-3">
                  <input class="form-control" id="age" type="number"
placeholder="age" name="userage" [ngModel]= "userdata.userage"
#userage="ngModel" required />
                  <label for="age">Age</label>
                </div>
                <p class="alert alert-danger"
*ngIf="userage.errors?.required && userage.touched" >Please enter the age </p>
              </div>
              <div class="col">
                <div class="form-check">
                  <input class="form-check-input" type="radio"
name="gender" value="male" id="flexRadioDefault1" [ngModel]=
"userdata.gender" #usergender="ngModel" required >
                  <label class="form-check-label" for="flexRadioDefault1"
style="background-color: whitesmoke; border-radius:
3px; width: 50%; text-align: center;">
                    Male
                  </label>
                </div>
              </div>
            </div>
          </form>
        </div>
      </div>
    </div>
  </div>
```

```

        </div>
        <div class="form-check" style="margin-top: 10px;">
            <input class="form-check-input" type="radio"
name="gender" value="female" id="flexRadioDefault2" [ngModel]=
"userdata.gender" #usergender="ngModel" required >
            <label class="form-check-label" for="flexRadioDefault2"
style="background-color: whitesmoke; border-radius:
3px; width: 50%; text-align: center;">
                Female
            </label>
        </div>
    </div>
    <p class="alert alert-danger"
*ngIf="usergender.errors?.required && usergender.touched" >Please click the
gender </p>
</div>
<div class="form-floating mb-3">
    <input class="form-control" id="inputphone" type="text"
placeholder="Phone Number" name="userphone" [ngModel]= "userdata.userphone"
#userphone="ngModel" required pattern="^(0|[1-9][0-9]*)$" maxlength="10"/>
    <label for="inputphone">Phone Number</label>
</div>
<p class="alert alert-danger" *ngIf="userphone.errors?.required
&& userphone.touched">Please enter the Phone number</p>
<p class="alert alert-danger"
*ngIf="userphone.errors?.pattern">please enter the valid phone number</p>
<div class="form-floating mb-3">
    <input class="form-control" id="inputEmail" name="uemail"
type="email" #uemail="ngModel" [ngModel]=
"userdata.uemail" placeholder="name@example.com" required email />
    <label for="inputEmail">Email address</label>
</div>
<p class="alert alert-danger" *ngIf="uemail.errors?.required &&
uemail.touched">Please enter the Email</p>
<p class="alert alert-danger" *ngIf="uemail.errors?.email &&
uemail.touched">Please enter the valid Email
</p>
<div class="form-floating mb-3">
    <input class="form-control" id="inputPassword"
type="password" placeholder="Password" name="userpass"
[ngModel]= "userdata.userpass" #userpass="ngModel" required
password minlength="8" />
    <label for="inputPassword">Create Password</label>
</div>
<p class="alert alert-danger" *ngIf="userpass.errors?.required
&& userpass.touched">Please Create a
password</p>
<p class="alert alert-danger"
*ngIf="userpass.errors?.minlength">Please Create a strong password (min 8
character) </p>
<h4 style="color: pink; margin: 1vw; display: flex; justify-
content: center;">Package</h4>
<select class="form-select form-select-lg mb-3"
name="pack" [ngModel]= "userdata.pack" #pack="ngModel" aria-label=".form-
select-lg example"
placeholder="Select Package">
    <option selected>Select Package</option>
    <option value="Hatha Yoga">Hatha Yoga</option>
    <option value="Vinyasa Yoga">Vinyasa Yoga</option>
    <option value="Iyengar Yoga">Iyengar Yoga</option>
    <option value="Ashtanga Yoga">Ashtanga Yoga</option>

```

```

        <option value="Bikram Yoga">Bikram Yoga</option>
        <option value="Kundalini Yoga">Kundalini Yoga</option>
    </select>
    <h4 style="color: pink; margin: 1vw; display: flex; justify-
content: center;">Level</h4>
    <div class="form-check" style="display: inline-block; color:
#c5c6c7; margin-left: 2vw; ">
        <input class="form-check-input" type="radio"
name="level" [ngModel]= "userdata.level" #level="ngModel" id="gridRadios1"
        value="Beginner" checked ngModel >
        <label class="form-check-label" for="level">
            Beginner
        </label>
    </div>
    <div class="form-check" style="display: inline-block; color:
#c5c6c7; margin-left: 2vw; ">
        <input class="form-check-input" type="radio"
name="level" [ngModel]= "userdata.level" #level="ngModel" id="gridRadios2"
        value="Intermediate" ngModel>
        <label class="form-check-label" for="level">
            Intermediate
        </label>
    </div>
    <div class="form-check" style="display: inline-block; color:
#c5c6c7; margin-left: 2vw; ">
        <input class="form-check-input" type="radio"
name="level" [ngModel]= "userdata.level" #level="ngModel" id="gridRadios3"
        value="Advanced" ngModel>
        <label class="form-check-label" for="level">
            Advanced
        </label>
    </div>
    <div>
        <h4 style="color: pink; margin: 1vw; display: flex; justify-
content: center;">Adds on</h4>
        <div class="form-check" style="display: inline-block; color:
#c5c6c7; margin-left: 2vw; ">
            <input class="form-check-input" type="checkbox"
id="flexCheckDefault" [ngModel]= "userdata.isbath" value="Steambath"
            name="isbath">
            <label class="form-check-label" for="flexCheckDefault">
                Steambath
            </label>
        </div>
        <div class="form-check" style="display: inline-block; color:
#c5c6c7; margin-left: 2vw; ">
            <input class="form-check-input"
type="checkbox" [ngModel]= "userdata.isdietplan" value="DietPlan"
name="isdietplan"
            id="flexCheckDefault">
            <label class="form-check-label" for="flexCheckDefault">
                DietPlan
            </label>
        </div>
        <div class="form-check" style="display: inline-block; color:
#c5c6c7; margin-left: 2vw; ">
            <input class="form-check-input" type="checkbox"
id="flexCheckDefault" [ngModel]= "userdata.istrain" value="PersonalTrainer"
            name="istrain">
            <label class="form-check-label" for="flexCheckDefault">
                PersonalTrainer

```



```

        </label>
      </div>
    </div>
    <div class="d-flex align-items-center justify-content-between
mt-4 mb-0">
      <button class="btn btn-info btn-block" value="Register"
type="submit" [disabled]="f.invalid" style="background-color: pink !important;
center; font-family: 'Big Shoulders Stencil Display', cursive;
                                font-family:
'Londrina Solid', cursive;"> Register</button><h5 style="color: pink;">{{msg}}
</h5>
    </div>
  </form>
</div>
  <div class="card-footer text-center py-3">
    <p class="card-text"><small class="text-muted">Yoga fitness
center</small></p>
    <button class="btn btn-info me-md-2" type="button" routerLink="/
manager" style="border: solid #1F2833; margin: 1vw; background-color: pink !
important;">Back
    </button>
  </div>
</div>
</div>
</div>
</div>
</div>

```

src/app/edituser/edituser.component.spec.ts

```
import { ComponentFixture, TestBed } from '@angular/core/testing';
import { EdituserComponent } from './edituser.component';
describe('EdituserComponent', () => {
  let component: EdituserComponent;
  let fixture: ComponentFixture<EdituserComponent>;
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [ EdituserComponent ]
    })
    .compileComponents();
  });
  beforeEach(() => {
    fixture = TestBed.createComponent(EdituserComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });
  it('should create', () => {
    expect(component).toBeTruthy();
  });
});
```

src\app\edituser\edituser.component.ts

```
import { Component, OnInit } from '@angular/core';
import { NgForm } from '@angular/forms';
import { ActivatedRoute, Params } from '@angular/router';
import { UsersService } from '../users.service';
@Component({
  selector: 'app-edituser',
  templateUrl: './edituser.component.html'
})
export class EdituserComponent implements OnInit {
  msg: string;
  date = new Date();
  userdata: { _id: number, Fname: string, lname: string, uemail:string,
  usage:Number, usergender:Number, gender:String, userpass:string,
  userphone:number ,level:string;
  isbath:boolean,istrain:boolean,isdietplan:boolean,pack:string }
  constructor(public activeRoute: ActivatedRoute, public userSer:
  UsersService) { }
  ngOnInit(): void {
    this.activeRoute.params.subscribe((param: Params) => {
      console.log(param);
      if (param.userid) {
        this.userSer.getSingleUserData(param.userid).subscribe((data: any[]))
=> {
          console.log(data);
          this.userdata = data[0];
        }, (error: any) => {
          console.log(error);
        });
      }
    });
  }
  editUser(form: NgForm) {
    form.value._id = this.userdata._id
    form.value.date=this.date;
    console.log(form.value);
    this.userSer.editsingleUserdata(form.value).subscribe((data: string) => {
      this.msg = data;
    }, (error: any) => {
      this.msg = "something went wrong";
    });
  }
}
```

src\app\footer1\footer1.component.html

```
<div class="container-fluid" style="background-color:#1F2833;">
  <footer class="d-flex flex-wrap justify-content-between align-items-
center py-3 border-top" style="background-color:#1F2833;">
    <p class="col-md-4 mb-0 text-muted">&copy; 2024 YogaFitnessCenter, GPM
College</p>
    <a routerLink="/" class="col-md-4 d-flex align-items-center justify-
content-center mb-3 mb-md-0 me-md-auto link-dark text-decoration-none">
      
    </a>
    <ul class="nav col-md-4 justify-content-end" >
      <li class="nav-item"><a routerLink="/" class="nav-link px-2 text-
muted" style="color: pink !important; font-size: 100%; font-family: 'Big
Shoulders Stencil Display', cursive; font-family: 'Londrina Solid',
cursive;">Home</a></li>
      <li class="nav-item"><a routerLink="packages/" class="nav-link px-2
text-muted" style="color: pink !important; font-size: 100%; font-family: 'Big
Shoulders Stencil Display', cursive; font-family: 'Londrina Solid',
cursive;">Package</a></li>
      <li class="nav-item"><a routerLink="contact/" class="nav-link px-2
text-muted" style="color: pink !important; font-size: 100%; font-family: 'Big
Shoulders Stencil Display', cursive; font-family: 'Londrina Solid',
cursive;">Contact Us</a></li>
    </ul>
  </footer>
</div>
```

src\app\footer1\footer1.component.spec.ts

```
import { ComponentFixture, TestBed } from '@angular/core/testing';
import { Footer1Component } from './footer1.component';
describe('Footer1Component', () => {
  let component: Footer1Component;
  let fixture: ComponentFixture<Footer1Component>;
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [ Footer1Component ]
    })
    .compileComponents();
  });
  beforeEach(() => {
    fixture = TestBed.createComponent(Footer1Component);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });
  it('should create', () => {
    expect(component).toBeTruthy();
  });
});
```

src\app\footer1\footer1.component.ts

```
import { Component, OnInit } from '@angular/core';
@Component({
  selector: 'app-footer1',
  templateUrl: './footer1.component.html'
})
export class Footer1Component implements OnInit {
  constructor() { }
  ngOnInit(): void {
  }
}
```

src/app/log.guard.spec.ts

```
import { TestBed } from '@angular/core/testing';
import { LogGuard } from '../log.guard';
describe('LogGuard', () => {
  let guard: LogGuard;
  beforeEach(() => {
    TestBed.configureTestingModule({});
    guard = TestBed.inject(LogGuard);
  });
  it('should be created', () => {
    expect(guard).toBeTruthy();
  });
});
```

src\app\log.guard.ts

```
import { Injectable } from '@angular/core';
import { CanActivate, Router } from '@angular/router';
import { UsersService } from '../users.service';
@Injectable({
  providedIn: 'root'
})
export class logGuard implements CanActivate {
  constructor(public userSer : UsersService, public myRouter : Router)
  {
  }
  canActivate(): boolean {
    if(!this.userSer.isLoggedIn())
    {
      this.myRouter.navigateByUrl("/profile");
    }
    return !this.userSer.isLoggedIn();
  }
}
```


src\app\login\login.component.html

```
<div class="container" style="min-height: 560px; font-family: 'Big Shoulders
Stencil Display', cursive;
font-family: 'Londrina Solid', cursive;">
  <div class="row justify-content-center" style="padding-top: 4%;">
    <div class="col-lg-5">
      <div class="card shadow-lg border-0 rounded-lg mt-5">
        <div class="card-header" style="background-color: #c5c6c7;">
          <h2 style="text-align: center;">Login</h2>
          <div class="card-body" style="background-color:
#1F2833;"><p class="alert alert-danger" *ngIf="msg" style="align-self:
center;">{{msg}}</p>
          <form #f="ngForm" (submit)="doLogin(f)">
            <div class="form-floating mb-3">
              <input class="form-control" id="inputEmail"
type="email" name="uemail"
              placeholder="name@example.com" ngModel/>
              <label for="inputEmail">Email address</label>
            </div>
            <div class="form-floating mb-3">
              <input class="form-control"
id="inputPassword" type="password" placeholder="Password" name="userpass"
ngModel/>
              <label for="inputPassword">Password</label>
            </div>
            <div class="d-flex align-items-center justify-
content-between mt-4 mb-0">
              <button type="submit" class="btn btn-info btn-
block" name="upassword" style="text-align: center;background-color: pink!
important; border-color: pink !important;" >Login</button>
              <!-- routerLink="/profile"-->
            </div>
          </form>
        </div>
        <div class="card-footer text-center py-3">
          <div class="small" ><a routerLink="/register"
style="color: #1F2833;">Need an account? Sign up!</a></div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

src\app\login\login.component.spec.ts

```
import { ComponentFixture, TestBed } from '@angular/core/testing';
import { LoginComponent } from '../login.component';
describe('LoginComponent', () => {
  let component: LoginComponent;
  let fixture: ComponentFixture<LoginComponent>;
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [ LoginComponent ]
    })
    .compileComponents();
  });
  beforeEach(() => {
    fixture = TestBed.createComponent(LoginComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });
  it('should create', () => {
    expect(component).toBeTruthy();
  });
});
```

src\app\login\login.component.ts

```
import { Component, OnInit } from '@angular/core';
import { NgForm } from '@angular/forms';
import { ActivatedRoute, Router } from '@angular/router';
import { UsersService } from '../users.service';
@Component({
  selector: 'app-login',
  templateUrl: './login.component.html'
})
export class LoginComponent implements OnInit {
  msg : string;
  constructor(public userSer : UsersService, public myRouter : Router,public
  activeRoute: ActivatedRoute) { }
  ngOnInit(): void {
  }
  doLogin(form:NgForm)
  {
    console.log(form.value);
    this.userSer.doUserLogin(form.value).subscribe((data:any[])=>{
      console.log(data);
      if(data.length==0)
      {
        this.msg = "Invalid Login";
      }
      else {
        localStorage.setItem("loggeduser", data[0]._id);
        localStorage.setItem("loggedusername", data[0].Fname);
        localStorage.setItem("loggedusernamee", data[0].lname);
        localStorage.setItem("loggedusergen", data[0].gender);
        localStorage.setItem("loggeduserage", data[0].userage);
        this.myRouter.navigateByUrl("/profile");
      }
      if(data[0]._id==1711611304560)
      {
        localStorage.setItem("loggedadmin", data[0]._id);
      }
    }, (error:any)=>{
      console.log(error);
      this.msg = "Something went wrong";
    });
  }
}
```

src\app\manager\manager.component.html

```
<div
  style="background-color: whitesmoke; min-height:80vh; font-family: 'Big
Shoulders Stencil Display', cursive;
font-family: 'Londrina Solid', cursive; background-color: #c5c6c7; border-
radius: 12px; padding-left: 2vh; padding-right: 2vh;">
  <div class="container-fluid">
    <div class="row" style="margin-top: 30px;">
      <div class="col-sm-4"></div>
      <div class="col-sm-4"><button class="btn btn me-md-2" style="
margin: 1vw; color: pink; background-color:#1F2833 ;">
        <h3>Member Management</h3>
      </button></div>
      <div class="col-sm-4 " >
        <div style="color:#c5c6c7; margin-top: 10vh;">
          Serach: <input type="text" #search
(keyup)="doSearch(search.value)"> &nbsp;
          <a routerLink="/adduser">
            
            </a>
          </div>
          <div *ngIf="msg"><p class="alert alert-success">{{msg}}</p></div>
        </div>
      </div>
    </div>
    <div class="row">
      <div class="col"></div>
      <div class="col">
        <table class="table table-striped shadow-lg container-fluid">
          <thead>
            <tr style="color: #1F2833;">
              <th scope="col">#</th>
              <th scope="col">Name</th>
              <th scope="col">Email</th>
              <th scope="col">Phone</th>
              <th scope="col">gender</th>
              <th scope="col">age</th>
              <th scope="col">package</th>
              <th scope="col">Level</th>
              <th scope="col">Steam bath</th>
              <th scope="col">Diet plan</th>
              <th scope="col">Personal Trainer</th>
              <th scope="col">Date of Purchase</th>
              <th scope="col">Paid</th>
              <th scope="col">Coupon</th>
              <th scope="col">Operation</th>
            </tr>
          </thead>
          <tbody>
            <tr *ngFor="let user of users; index as i;">
              <th scope="row">{{i+1}}</th>
              <td>{{user.Fname}} {{user.lname}}</td>
              <td>{{user.uemail}}</td>
              <td>{{user.userphone}}</td>
              <td>{{user.gender}}</td>
              <td>{{user.userage}}</td>
              <td>{{user.pack}}</td>
            </tr>
          </tbody>
        </table>
      </div>
    </div>
  </div>
```

```

        <td>{{user.level}}</td>
        <td>{{user.isbath}}</td>
        <td>{{user.isdietplan}}</td>
        <td>{{user.istrain}}</td>
        <td>{{user.date|slice :0:10}}</td>
        <td>{{user.paid}}</td>
        <td>{{user.coupon}}</td>
        <td style="display: flex; margin: 1vh; background-
color:#c5c6c7">
            &nbsp;
        </td>
    </tr>
</tbody>
</table>
<p *ngIf="users.length==0"> No Records Found</p>
</div>
<div class="col"></div>
</div>
</div>
</div>

```

src/app/manager/manager.component.spec.ts

```
import { ComponentFixture, TestBed } from '@angular/core/testing';
import { ManagerComponent } from './manager.component';
describe('ManagerComponent', () => {
  let component: ManagerComponent;
  let fixture: ComponentFixture<ManagerComponent>;
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [ ManagerComponent ]
    })
    .compileComponents();
  });
  beforeEach(() => {
    fixture = TestBed.createComponent(ManagerComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });
  it('should create', () => {
    expect(component).toBeTruthy();
  });
});
```

src\app\manager\manager.component.ts

```
import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { UsersService } from '../users.service';
@Component({
  selector: 'app-manager',
  templateUrl: './manager.component.html'
})
export class ManagerComponent implements OnInit {
  msg: string;
  users: any[] = [];
  constructor(public userSer: UsersService) { }
  ngOnInit(): void {
    this.userSer.getAllUsers().subscribe((data: any[]) => {
      console.log(data);
      this.users = data;
    }, (error: any) => {
      this.msg = "Something went wrong"
    });
  }
  deleteUser(userId: number) {
    if (confirm("Do you want to delete this user?")) {
      this.userSer.deleteUserData(userId).subscribe((data: string) => {
        this.msg = data;
        var index = this.users.findIndex((obj) => {
          return obj._id == userId;
        });
        this.users.splice(index, 1);
      }, (error: any) => {
        this.msg = "something went wrong";
      });
    }
  }
  doSearch(search: string) {
    this.userSer.searchUsers(search).subscribe((data: any[]) => {
      console.log(data);
      this.users=data;
    }, (error: any) => {
      console.log(error);
    });
  }
}
```

```
src\app\marq\marq.component.html
```

```
<div style="z-index:0;">
  <marquee direction="left" height="10%" style="background-color:
#1F2833 ;" >
    <p style="color:#daa520; border-radius: 5px;"><strong> NOW GET Yoga
fitness center PACKAGES AT 50% OFF !!! Coupon code: 452dfg </strong></p>
  </marquee>
</div>
```


src/app/marq/marq.component.spec.ts

```
import { ComponentFixture, TestBed } from '@angular/core/testing';
import { MarqComponent } from './marq.component';
describe('MarqComponent', () => {
  let component: MarqComponent;
  let fixture: ComponentFixture<MarqComponent>;
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [ MarqComponent ]
    })
    .compileComponents();
  });
  beforeEach(() => {
    fixture = TestBed.createComponent(MarqComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });
  it('should create', () => {
    expect(component).toBeTruthy();
  });
});
```

src\app\marq\marq.component.ts

```
import { Component, OnInit } from '@angular/core';
@Component({
  selector: 'app-marq',
  templateUrl: './marq.component.html'
})
export class MarqComponent implements OnInit {
  constructor() { }
  ngOnInit(): void {
  }
}
```

src\app\mempackages\mempackages.component.html

```
<div
  style="font-family: 'Big Shoulders Stencil Display', cursive;
font-family: 'Londrina Solid', cursive; background-color: #c5c6c7; border-
radius: 12px; padding-left: 2vh; padding-right: 2vh;">
  <div class="d-grid gap-2 d-md-flex justify-content-md-center">
    <button class="btn btn-info me-md-2" type="button" routerLink="/"
      style="border:solid #1F2833;background-color: pink; margin: 1vw;">Home
    </button>
    <button class="btn btn-info me-md-2" type="button" routerLink="/profile"
      style="border:solid #1F2833;background-color: pink; margin:
1vw;">Profile
    </button>
    <button class="btn btn me-md-2" style=" margin: 1vw; color: pink;
background-color:#1F2833 ;">
      <h3>Package</h3>
    </button>
  </div>
  <hr style="height: 2px; margin: 1%;padding: 0;">
  <div class="row row-cols-1 row-cols-md-3 g-1">
    <div class="col">
      <div class="card shadow-lg" style="margin: 2vh;">
        
        <div class="card-body">
          <div style="background-color: #1F2833; color: pink; border-radius:
5px;">
            <h4 class="card-title" style=" text-align:
center;">{{pac.HathaYogaThen }}</h4><!-- Hatha Yoga-Card 1-->
          </div>
          <p class="card-text">{{pac.HathaYogaS}}</p>
          <div style=" color: #1F2833; display: flex; justify-content: flex-
end; border-radius: 5px; "
            <h5 class="card-title bg-info" style=" text-align: center;
padding: .4vh; border-radius: 4px;background-color: pink !important;">
Price :&#8377;2000</h5>
          </div>
        </div>
        <button type="button" class="btn btn-info" style="background-color:
pink; border-color: pink;"
          (click)="pac.HathaYogaSBuy()">BUY</button>
        </div>
      </div>
    <div class="col">
      <div class="card shadow-lg" style="margin: 2vh;">
        
        <div class="card-body">
          <div style="background-color: #1F2833; color: pink; border-radius:
4px;">
            <h4 class="card-title" style=" text-align:
center;">{{pac.VinyasaYogaM}}</h4> <!--Weigthlifting-card 2-->
          </div>
          <p class="card-text">{{pac.VinyasaYogaNa}}</p>
          <div style=" color: #1F2833; display: flex; justify-content: flex-
end; border-radius: 5px; "
            <h5 class="card-tittle bg-info" style=" text-align: center;
padding: .4vh; border-radius: 4px;background-color: pink !important;">
```

```

Price :&#8377;2000</h5>
    </div>
</div>
    <button type="button" class="btn btn-info"
        (click)="pac.VinyasaYogaMBuy()" style="background-color: pink;
border-color: pink;">BUY</button>
    </div>
</div>
<div class="col">
    <div class="card shadow-lg" style="margin: 2vh;">
        
        <div class="card-body">
            <div style="background-color: #1F2833; color: pink; border-radius:
5px;">
                <h4 class="card-title" style=" text-align:
center;">{{pac.IyengarYoga}}</h4>
                <!--Iyengar Yoga --card3 -->
            </div>
            <p class="card-text">{{pac.IyengarYo}}</p>
            <div style=" color: #1F2833; display: flex; justify-content: flex-
end; border-radius: 5px; ">
                <h5 class="card-title bg-info" style=" text-align: center;
padding: .4vh; border-radius: 4px;background-color: pink !important;">
Price :&#8377;2000</h5>
            </div>
            </div><button type="button" class="btn btn-info"
                (click)="pac.IyengarYogaBuy()" style="background-color: pink;
border-color: pink;">BUY</button>
            </div>
        </div>
    <div class="col">
        <div class="card shadow-lg" style="margin: 2vh;">
            
            <div class="card-body">
                <div style="background-color: #1F2833; color: pink; border-radius:
5px;">
                    <h4 class="card-title" style=" text-align:
center;">{{pac.AshtangaYoga}}</h4>
                    <!--Kick boxing -- card4-->
                </div>
                <p class="card-text">{{pac.AshtangaYo}}</p>
                <div style=" color: #1F2833; display: flex; justify-content: flex-
end; border-radius: 5px; ">
                    <h5 class="card-title bg-info" style=" text-align: center;
padding: .4vh; border-radius: 4px;background-color: pink !important;">
Price :&#8377;2000</h5>
                </div>
                </div><button type="button" class="btn btn-info"
                    (click)="pac.AshtangaYogaBuy()" style="background-color: pink;
border-color: pink;">BUY</button>
                </div>
            </div>
        <div class="col">
            <div class="card shadow-lg" style="margin: 2vh;">
                
                <div class="card-body">

```

```

    <div style="background-color: #1F2833; color: pink; border-radius:
5px;">
        <h4 class="card-title" style=" text-align:
center;">{{pac.BikramYoga}}</h4><!-- Bikram Yoga -- card5-->
    </div>
        <p class="card-text">{{pac.BikramYo}}</p>
        <div style=" color: #1F2833; display: flex; justify-content: flex-
end; border-radius: 5px; ">
            <h5 class="card-title bg-info" style=" text-align: center;
padding: .4vh; border-radius: 4px;background-color: pink !important;">
Price :&#8377;2000</h5>
            </div>
            </div><button type="button" class="btn btn-info"
(click)="pac.BikramYogaBuy()" style="background-color: pink; border-
color: pink;">BUY</button>
        </div>
    </div>
    <div class="col">
        <div class="card shadow-lg" style="margin: 2vh;">
            
            <div class="card-body">
                <div style="background-color: #1F2833; color: pink; border-radius:
5px;">
                    <h4 class="card-title" style=" text-align:
center;">{{pac.KundaliniYoga}}</h4>
                    <!--sport--card5-->
                </div>
                <p class="card-text">
                    {{pac.KundaliniYo}} </p>
                <div style=" color: #1F2833; display: flex; justify-content: flex-
end; border-radius: 5px; ">
                    <h5 class="card-title bg-info" style=" text-align: center;
padding: .4vh; border-radius: 4px;background-color: pink !important;">
Price :&#8377;2000</h5>
                    </div>
                    </div><button type="button" class="btn btn-info"
(click)="pac.KundaliniYogaBuy()" style="background-color: pink;
border-color: pink; border-color: pink;">BUY</button>
                </div>
            </div>
        </div>
    </div>
</div>

```

src/app/mempackages/mempackages.component.spec.ts

```
import { ComponentFixture, TestBed } from '@angular/core/testing';
import { MempackagesComponent } from './mempackages.component';
describe('MempackagesComponent', () => {
  let component: MempackagesComponent;
  let fixture: ComponentFixture<MempackagesComponent>;
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [ MempackagesComponent ]
    })
    .compileComponents();
  });
  beforeEach(() => {
    fixture = TestBed.createComponent(MempackagesComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });
  it('should create', () => {
    expect(component).toBeTruthy();
  });
});
```

src\app\mempackages\mempackages.component.ts

```
import { Component, OnInit } from '@angular/core';
import { PackService } from '../pack.service';
@Component({
  selector: 'app-mempackages',
  templateUrl: './mempackages.component.html'
})
export class MempackagesComponent implements OnInit {
  useriid:any= localStorage.getItem("loggeduser");
  constructor(public pac:PackService) {
  }
  ngOnInit(): void {console.log(this.useriid);
  }
}
```

```
src\app\navmain\navmain.component.css
```

```
.nav-item :hover {  
    background-color: black;  
}  
.nav-item button {  
    border-bottom-color: pink !important;  
    border-right-color: pink !important;  
    border-top-color: pink !important;  
    border-left-color: pink !important;  
}  
.nav-item {  
    border-bottom-color: pink !important;  
}
```


src\app\navmain\navmain.component.html

```
<div style=" font-family: 'Big Shoulders Stencil Display', cursive;
font-family: 'Londrina Solid', cursive;">
  <nav class="navbar navbar-expand-lg navbar-light "
    style="background-color: #1F2833; border-top: 4px solid pink; z-
index: 1;">
    <div class="container-fluid">
      <a class="navbar-brand" routerLink="">
        
      </a>
      <a class="navbar-brand" routerLink="" style="color: pink;">
        <h1 style="font-family: 'Big Shoulders Stencil Display',
cursive;">Yoga Fitness center</h1>
      </a>
      <button class="navbar-toggler " type="button" style="color:
pink;" data-bs-toggle="collapse"
        data-bs-target="#navbarSupportedContent" aria-
controls="navbarSupportedContent" aria-expanded="false"
        aria-label="Toggle navigation" style="background-color:
pink;">
        <span class="navbar-toggler-icon "></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarSupportedContent"
style="margin-left: 40%;">
        <ul class="navbar-nav me-auto mb-lg-0">
          <li class="nav-item" style="margin-right: 1vh;">
            <h3 style="color: pink" *ngIf="userSer.isLoggedIn()">
Welcome! &nbsp;   {{userSer.name}} </h3>
          </li>
          <li class="nav-item" style="margin-right: 1vh;">
            <button class="btn btn-outline-info" routerLink="/"
style="color: pink; font-size: 100%; font-family: 'Big Shoulders Stencil
Display', cursive; font-family: 'Londrina Solid', cursive;">Home</button>
          </li>
          <li class="nav-item" style="margin-right: 1vh;">
            <button class="btn btn-outline-info" routerLink="/"
packages" style="color: pink; font-size: 100%; font-family: 'Big Shoulders
Stencil Display', cursive; font-family: 'Londrina Solid', cursive;">Packages</
button>
          </li>
          <li class="nav-item" style="margin-right: 1vh;">
            <button class="btn btn-outline-info" routerLink="/"
register" style="color: pink; font-size: 100%; font-family: 'Big Shoulders
Stencil Display', cursive; font-family: 'Londrina Solid', cursive;" *ngIf="!
userSer.isLoggedIn()">Register</button>
          </li>
          <li class="nav-item" style="margin-right: 1vh;">
            <button class="btn btn-outline-info" routerLink="/"
login" style="color: pink; font-size: 100%; font-family: 'Big Shoulders
Stencil Display', cursive; font-family: 'Londrina Solid', cursive;" *ngIf="!
userSer.isLoggedIn()">Login</button>
          </li>
          <li class="nav-item" style="margin-right: 1vh;">
            <button class="btn btn-outline-info" routerLink="/"
profile" style="color: pink; font-size: 100%; font-family: 'Big Shoulders
Stencil Display', cursive; font-family: 'Londrina Solid', cursive;"
*ngIf="userSer.isLoggedIn()">Profile</button>
          </li>
        </ul>
      </div>
    </div>
  </nav>
</div>
```

```

        <li class="nav-item" style="margin-right: 1vh;">
            <button class="btn btn-outline-info" routerLink="/
manager" style="color: pink; font-size: 100%; font-family: 'Big Shoulders
Stencil Display', cursive; font-family: 'Londrina Solid', cursive;"
*ngIf="userSer.isAdmin()">manager</button>
        </li>
        <li class="nav-item" style="margin-right: 1vh;">
            <button class="btn btn-outline-info" routerLink="/
register" style="color: pink; font-size: 100%; font-family: 'Big Shoulders
Stencil Display', cursive; font-family: 'Londrina Solid', cursive;"
*ngIf="userSer.isLoggedIn()"
                (click)="doLogout()">logout</button>
        </li>
    </ul>
</div>
</div>
</nav>
</div>

```

src/app/navmain/navmain.component.spec.ts

```
import { ComponentFixture, TestBed } from '@angular/core/testing';
import { NavmainComponent } from './navmain.component';
describe('NavmainComponent', () => {
  let component: NavmainComponent;
  let fixture: ComponentFixture<NavmainComponent>;
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [ NavmainComponent ]
    })
    .compileComponents();
  });
  beforeEach(() => {
    fixture = TestBed.createComponent(NavmainComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });
  it('should create', () => {
    expect(component).toBeTruthy();
  });
});
```

src\app\navmain\navmain.component.ts

```
import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { UsersService } from '../users.service';
import { ActivatedRoute } from '@angular/router';
@Component({
  selector: 'app-navmain',
  templateUrl: './navmain.component.html',
  styleUrls: ['./navmain.component.css']
})
export class NavmainComponent implements OnInit {
  name: string;
  constructor(public userSer: UsersService, public activeRoute:
  ActivatedRoute, public myRouter: Router) { }
  ngOnInit(): void {
  }
  doLogout() {
    localStorage.clear();
    this.myRouter.navigateByUrl("/");
  }
}
```

src\app\notfound\notfound.component.html

```
<div class="cen" style="height: 50vh; " >
  <div class="alert alert-info" role="alert" style="margin-top: 20vh;">
    <h5 style="text-align: center;"> Page NOT FOUND!</h5>
  </div>
</div>
```

src/app/notfound/notfound.component.spec.ts

```
import { ComponentFixture, TestBed } from '@angular/core/testing';
import { NotfoundComponent } from './notfound.component';
describe('NotfoundComponent', () => {
  let component: NotfoundComponent;
  let fixture: ComponentFixture<NotfoundComponent>;
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [ NotfoundComponent ]
    })
    .compileComponents();
  });
  beforeEach(() => {
    fixture = TestBed.createComponent(NotfoundComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });
  it('should create', () => {
    expect(component).toBeTruthy();
  });
});
```

src\app\notfound\notfound.component.ts

```
import { Component, OnInit } from '@angular/core';
@Component({
  selector: 'app-notfound',
  templateUrl: './notfound.component.html'
})
export class NotfoundComponent implements OnInit {
  constructor() { }
  ngOnInit(): void {
  }
}
```

src/app/pack.service.spec.ts

```
import { TestBed } from '@angular/core/testing';
import { PackService } from './pack.service';
describe('PackService', () => {
  let service: PackService;
  beforeEach(() => {
    TestBed.configureTestingModule({});
    service = TestBed.inject(PackService);
  });
  it('should be created', () => {
    expect(service).toBeTruthy();
  });
});
```


src/app/pack.service.ts

```
import { Injectable } from '@angular/core';
import { HttpClient, HttpClientModule } from '@angular/common/http';
import { Router } from '@angular/router';
@Injectable({
  providedIn: 'root'
})
export class PackService {
  packname="";
  packdesc="";
  accountStatus="Inactive";
  HathaYogaThen ="Hatha Yoga";
  HathaYogaS ="Hatha Yoga, considered a gentler form of Yoga, comprises of
basic introductory poses in which you're required to hold each pose for a few
breaths.It's important to be empty stomach to practice yoga, trying remaining
quite and silent while practicing it.";
  VinyasaYogaM="Vinyasa Yoga";
  VinyasaYogaNa="Vinyasa Yoga helps in developing a flow in your body while
practicing Yoga by linking movement with breath. It can lead to rise in your
heart rate as the pace is bound to increase with each pose. It focuses on
synchronizing the breath with a continuous flow of postures improving our
blood circulation.";
  IyengarYoga="Iyengar Yoga";
  IyengarYo="The poses are for longer periods as they help you align your
body posture in the right way. It's therapeutic for several conditions and
helps in cultivating strength and flexibility. Toned muscles, pain relief
from different joints of the body."
  AshtangaYoga="Ashtanga Yoga";
  AshtangaYo="Ashtanga Yoga is only for those looking for a challenging and
orderly approach to yoga.It means - eight limbs or branches of yoga which are
- Yama - moral codes, Niyama -self-purification and study, Asana - posture,
Pranayama - breath control. "
  KundaliniYoga="Kundalini Yoga";
  KundaliniYo=" Kundalini Yoga includes performing repetitive physical
exercises coupled with intense breath work while chanting and meditating. It
releases the untapped energy which resides within you, providing a sense of
self-awareness. It's also a blend of bhakti yoga.";
  BikramYo="Bikram Yoga helps in increasing blood flow, and helps in
improving posture, balance, and mobility.Practice this series of 26 poses and
2 breathing exercises in a room heated to around 105 degrees and 40% humidity
and get drenched in sweat.";
  BikramYoga="Bikram Yoga";
  stbath=1000;
  pt=2000;
  dplan=500;
  name=localStorage.getItem("loggedusername");
  namee=localStorage.getItem("loggedusernamee");
  gender=localStorage.getItem("loggedusergen");
  age=localStorage.getItem("loggeduserage");
  constructor(public http:HttpClient,public router: Router) {
    console.log("service created");
  }
  HathaYogaSBuy(){
    this.packname=this.HathaYogaThen ;
    localStorage.setItem("packna", this.packname);
    this.packdesc=this.HathaYogaS;
    localStorage.setItem("packds",this.packdesc);
    this.router.navigateByUrl("/cart")
  }
}
```

```

VinyasaYogaMbuy(){
    this.packname=this.VinyasaYogaM;
    localStorage.setItem("packna", this.packname);
    this.packdesc=this.VinyasaYogaNa;
    localStorage.setItem("packds",this.packdesc);
    this.router.navigateByUrl("/cart")
}
IyengarYogaBuy(){
    this.packname=this.IyengarYo;
    localStorage.setItem("packna", this.packname);
    this.packdesc=this.IyengarYoga;
    localStorage.setItem("packds",this.packdesc);
    this.router.navigateByUrl("/cart")
}
AshtangaYogaBuy(){
    this.packname=this.AshtangaYo;
    localStorage.setItem("packna", this.packname);
    this.packdesc=this.AshtangaYoga;
    localStorage.setItem("packds",this.packdesc);
    this.router.navigateByUrl("/cart")
}
BikramYogaBuy(){
    this.packname=this.BikramYoga;
    localStorage.setItem("packna", this.packname);
    this.packdesc=this.BikramYo;
    localStorage.setItem("packds",this.packdesc);
    this.router.navigateByUrl("/cart")
}
KundaliniYogaBuy(){
    this.packname=this.KundaliniYoga;
    localStorage.setItem("packna", this.packname);
    this.packdesc=this.KundaliniYo;
    localStorage.setItem("packds",this.packdesc);
    this.router.navigateByUrl("/cart")
}
}

```

src\app\payment\payment.component.html

```
<div class="container" style="font-family: 'Big Shoulders Stencil Display',
cursive;
font-family: 'Londrina Solid', cursive; background-color: #c5c6c7; max-
height:200vh; padding-bottom: 20vh;">
<div class="d-grid gap-2 d-md-flex justify-content-md-center">
  <button class="btn btn-info me-md-2" type="button" routerLink="/profile"
    style="border:solid #1F2833;background-color: pink; margin:
1vw;">Profile</button>
  <button class="btn btn-info me-md-2" type="button" routerLink="/packages"
    style="border:solid #1F2833;background-color: pink; margin:
1vw;">Packages
  </button>
</div> <!-- nav bar-->
<hr style="height: 2px; margin: 1%;padding: 0;">
  <div class="card text-center shadow-lg" style="margin: 4vw; background-
color: #c5c6c7;">
    <div class="card-header" >
      {{packname}} has been purchased
    </div>
    <div class="card-body" style="background-color:whitesmoke;">
      <h2 class="card-title text-success">Payment Successfull</h2>
      <p class="card-text">Thanks for choosing Yoga fitness center</p>
      <a routerLink="/profile" class="btn btn-info" style="background-
color: pink;
        border-bottom-color: pink !important;
        border-right-color: pink !important;
        border-top-color: pink !important;
        border-left-color: pink !important;
        ">Lets Go &raquo;</a>
    </div>
    <div class="card-footer text-muted">
      Yoga fitness center
    </div>
  </div>
</div>
```

src/app/payment/payment.component.spec.ts

```
import { ComponentFixture, TestBed } from '@angular/core/testing';
import { PaymentComponent } from './payment.component';
describe('PaymentComponent', () => {
  let component: PaymentComponent;
  let fixture: ComponentFixture<PaymentComponent>;
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [ PaymentComponent ]
    })
    .compileComponents();
  });
  beforeEach(() => {
    fixture = TestBed.createComponent(PaymentComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });
  it('should create', () => {
    expect(component).toBeTruthy();
  });
});
```

src\app\payment\payment.component.ts

```
import { Component, OnInit } from '@angular/core';
import { PackService } from '../pack.service';
@Component({
  selector: 'app-payment',
  templateUrl: './payment.component.html'
})
export class PaymentComponent implements OnInit {
  packname=localStorage.getItem("packna");
  constructor( public pac: PackService) { }
  ngOnInit(): void {
  }
}
```

src\app\profile\profile.component.html

```
<div class="container" style="font-family: 'Big Shoulders Stencil Display',
cursive;
font-family: 'Londrina Solid', cursive; background-color: #c5c6c7; max-
height:300vh; border-radius: 12px;">
  <div class="d-grid gap-2 d-md-flex justify-content-md-center">
    <button class="btn btn-info me-md-2" type="button" routerLink="/"
style="border:solid #1F2833;background-color:pink; margin: 1vw;">Home
    </button>
    <button class="btn btn me-md-2" style=" margin: 1vw; color: pink;
background-color:#1F2833 ;">
      <h3>Profile</h3>
    </button>
    <button class="btn btn-info me-md-2" type="button" routerLink="/
packages"
style="border:solid #1F2833;background-color: pink; margin:
1vw;">Package
    </button>
  </div>
  <hr style="height: 2px; margin: 1%;padding: 0; ">
  <div style="padding-bottom: 2vh;">
    <div style="display: flex; flex-flow: row wrap;">
      <div class="card mb-3 shadow-lg"
style="width: 18rem; margin: 0vw 1vw 1vw 1vw; background-
color: #c5c6c7; border-radius: 12px;">
        <div style="background-color:rgb(255, 255, 255); margin:
1vh; border-radius: 12px;">
          
        </div>
        <div class="card-body " *ngIf="userdata">
          <h5 class="card-title">Name : {{userdata.Fname}}
{{userdata.lname}}</h5>
          <hr>
          <p class="card-text">Age: {{userdata.userage}}<br>
Gender: {{userdata.gender}} <br>
Expertise level : {{userdata.level}}</p>
          <div class="d-grid gap-2 col-6 mx-auto" style="margin:
1vh;">
            <button class="btn btn-info" type="button"
routerLink="/packages"
style="border:solid #1F2833;background-
color:pink">Buy
            Package</button>
          </div>
        </div>
      </div>
      <div class="small"><a routerLink="/contact" style="color:
#1F2833; display: flex; justify-content: center; padding-bottom: 1vh;">Need
Help?</a>
    </div>
  </div>
  <div style="display: flex; flex-direction: column;" >
    <div class="card " style="background-color: #c5c6c7; max-
width: 120vh; border-radius: 12px;">
      <div>
        
    </div>
    <div class="card-body shadow-lg" style="background-color:
#c5c6c7; border-radius: 12px;">
        <h4 class="card-title" *ngIf="userdata.pack">Package:
{{userdata.pack}}-{{userdata.level}}</h4> <h4 class="card-title" *ngIf="!
userdata.pack">Package: 7days Free Trial</h4>
        <hr>
        <div *ngIf="userdata">
            <span>Adds on:</span>&nbsp;<span
*ngIf="addbath()"></span> &nbsp;<span
*ngIf="addtrain()"></span> &nbsp;<span
*ngIf="adddiet()"></span>
            <hr>
            <p class="card-text">Purchased on: {{userdata.date |
slice :0:10 }} <br></p>
        </div>
        <p class="card-text"><small class="text-muted">Yoga
fitness center</small></p>
    </div>
    <div class="container-fluid"
style="display: flex; flex-wrap: wrap; justify-content:
center; background-color: #c5c6c7; border-radius: 12px; padding: 0%; margin-
top: 0.5vw; ">
        <iframe max-width="1vh" style="margin:0.1vh; border-
radius: 12px;"
src="https://www.youtube.com/embed/FdyhENXyIQ4"
title="YouTube video player" frameborder="0"
allow="accelerometer; autoplay; clipboard-write;
encrypted-media; gyroscope; picture-in-picture"
allowfullscreen></iframe>
        <iframe max-width="1vh" style="margin:0.1vh; border-
radius: 12px;"
src="https://www.youtube.com/embed/brjAjq4zEIE"
title="YouTube video player" frameborder="0"
allow="accelerometer; ?autoplay=1; clipboard-write;
encrypted-media; gyroscope; picture-in-picture"
allowfullscreen></iframe>
        <iframe max-width="1vh" style="margin:0.1vh; border-
radius: 12px;"
src="https://www.youtube.com/embed/cMfChJLqma4"
title="YouTube video player" frameborder="0"
allow="accelerometer; autoplay; clipboard-write;
encrypted-media; gyroscope; picture-in-picture"
allowfullscreen></iframe>
    </div>
</div>
</div>
</div>
</div>

```

src/app/profile/profile.component.spec.ts

```
import { ComponentFixture, TestBed } from '@angular/core/testing';
import { ProfileComponent } from './profile.component';
describe('ProfileComponent', () => {
  let component: ProfileComponent;
  let fixture: ComponentFixture<ProfileComponent>;
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [ ProfileComponent ]
    })
    .compileComponents();
  });
  beforeEach(() => {
    fixture = TestBed.createComponent(ProfileComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });
  it('should create', () => {
    expect(component).toBeTruthy();
  });
});
```


src\app\profile\profile.component.ts

```
import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { PackService } from '../pack.service';
import { UsersService } from '../users.service';
import { ActivatedRoute, Params } from '@angular/router';
@Component({
  selector: 'app-profile',
  templateUrl: './profile.component.html'
})
export class ProfileComponent implements OnInit {
  msg: string;
  userdata: { _id: number, Fname: string, lname: string, uemail: string,
  usage: Number, usergender: Number, gender: String, userpass: string,
  userphone: number, level: string, pack: string, istrain: boolean, isdietplan:
  boolean, isbath: boolean; date: string; }
  useriid: any = localStorage.getItem("loggeduser");
  constructor(public userSer: UsersService, public activeRoute:
  ActivatedRoute, public pac: PackService, public myRouter: Router) {
  }
  ngOnInit(): void {
    console.log(this.useriid);
    this.userSer.getloggedUserData(this.useriid).subscribe((data: any[]) => {
      console.log(data);
      this.userdata = data[0];
    }, (error: any) => {
      console.log(error);
    });
  }
  addbath()
  {
    if(this.userdata.isbath==true){
      return true
    }
    else{
      return false
    }
  }
  addtrain()
  {
    if(this.userdata.istrain==true){
      return true
    }
    else{
      return false
    }
  }
  adddiet()
  {
    if(this.userdata.isdietplan==true){
      return true
    }
    else{
      return false
    }
  }
}
```

src\app\slide\slide.component.html

```
<div id="carouselExampleCaptions" class="carousel slide" data-bs-
ride="carousel">
  <div class="carousel-indicators">
    <button type="button" data-bs-target="#carouselExampleCaptions" data-bs-
slide-to="0" class="active" aria-current="true" aria-label="Slide 1"></button>
    <button type="button" data-bs-target="#carouselExampleCaptions" data-bs-
slide-to="1" aria-label="Slide 2"></button>
    <button type="button" data-bs-target="#carouselExampleCaptions" data-bs-
slide-to="2" aria-label="Slide 3"></button>
  </div>
  <div class="carousel-inner">
    <div class="carousel-item active">
      
      <div class="carousel-caption d-none d-md-block">
        <h1 style="color: rgb(255, 243, 246); font-family: 'Big Shoulders
Stencil Display', cursive;
font-family: 'Londrina Solid', cursive; margin-bottom: 20%;">
Welcome to <br> <span style="font-family: 'Big Shoulders Stencil Display',
cursive;">Yoga fitness center</span></h1>
      </div>
    </div>
    <div class="carousel-item">
      
      <div class="carousel-caption d-none d-md-block">
        <h2 style="color: pink; background-color: #1F2833; margin-left:
14vw; border-radius: 5px; margin-right: 14vw; font-family: 'Big Shoulders
Stencil Display', cursive;
font-family: 'Londrina Solid', cursive;">THE BEST FITNESS CENTER ON
THE PLANET</h2>
        <p style="color: #1F2833;background-color: pink; border-radius:
5px;"><strong>
We believe in providing a high-quality fitness experience at an
affordable cost. Read more about our membership options to decide which one
works best for you.</strong></p>
      </div>
    </div>
    <div class="carousel-item">
      
      <div class="carousel-caption d-none d-md-block">
        <h2 style="color: pink; font-family: 'Big Shoulders Stencil
Display', cursive;
font-family: 'Londrina Solid', cursive; background-color: #1F2833;
margin-left: 14vw; border-radius: 5px; margin-right: 14vw;">YOUR FITNESS IS
ESSENTIAL™</h2>
        <p style="color: #1F2833;background-color: pink; border-radius:
5px;"><strong>Making your overall physical and mental wellness a priority!
will melt away stress, decrease anxiety, and increase your overall positive
energy levels, and you'll feel healthy and strong throughout the year!</
strong></p>
      </div>
    </div>
  </div>
  <button class="carousel-control-prev" type="button" data-bs-
target="#carouselExampleCaptions" data-bs-slide="prev">
    <span class="carousel-control-prev-icon" aria-hidden="true"></span>
    <span class="visually-hidden">Previous</span>
  </button>
```

```

        <button class="carousel-control-next" type="button" data-bs-
target="#carouselExampleCaptions" data-bs-slide="next">
        <span class="carousel-control-next-icon" aria-hidden="true"></span>
        <span class="visually-hidden">Next</span>
    </button>
</div>
<div style="z-index:0;" *ngIf="!userSer.isLoggedIn()">
    <marquee direction="left" style="background-color: #1F2833 ; margin:
0%; padding: 0%;" >
        <p style="color:goldenrod; border-radius: 5px;"><strong>REGISTER NOW
AND GET 7 DAYS FREE TRIAL !</strong></p>
    </marquee>
</div>
<div class=" p-5 rounded" style="background-color: #1F2833;">
    <div class="col-sm-8 mx-auto" style="background-color: #1F2833">
        <h1 style="color: pink; font-family: 'Big Shoulders Stencil Display',
cursive;
font-family: 'Londrina Solid', cursive;">ABOUT Yoga fitness center</h1>
        <p style="color: #c5c6c7;">
            Whether you're a first-time user or a veteran member, we're here to
provide a environment in which anyone - and everyone - can be comfortable.
            No matter what you're looking for in a fitness center, we've got a
membership option made for you. All Planet Fitness members enjoy unlimited
access to their home club and the support of our friendly, knowledgeable
staff anytime you need it.</p>
        <p>
            <a class="btn btn-info" routerLink="/login" role="button"
style="background-color: pink;
border-bottom-color: pink !important;
border-right-color: pink !important;
border-top-color: pink !important;
border-left-color: pink !important;">Lets Go &raquo;</a>
        </p>
    </div>
</div>
<div style="background-color: #1F2833;" class="abus">
    <!--image-->
    <div class="card-group" style="background-color: #1F2833; margin: 1vw;" >
        <div class="card" style="background-color: #1F2833; margin: 1vh; border-
radius: 12px;">
            
            <div class="card-body" style="background-color: #1F2833;">
                <h5 class="card-title" style="font-family: 'Big Shoulders Stencil
Display', cursive;
font-family: 'Londrina Solid', cursive; color: pink; display: flex;
justify-content: center;"> Perfect for women </h5>
                <p class="card-text" style="color: #c5c6c7; display: flex; justify-
content: center;">We have both unisex and women only facilities with female
coaches.</p>
            </div>
        </div>
        <div class="card" style="background-color: #1F2833; margin: 1vh; border-
radius: 12px;">
            
            <div class="card-body" style="background-color: #1F2833;">
                <h5 class="card-title" style="font-family: 'Big Shoulders Stencil
Display', cursive;
font-family: 'Londrina Solid', cursive; color: pink; display: flex;
justify-content: center;">Outdoor Programs</h5>

```

```

        <p class="card-text" style="color: #c5c6c7; display: flex; justify-
content: center;">We conduct outdoor programs every week , at free of cost.</
p>
    </div>
</div>
<div class="card" style="background-color: #1F2833; margin: 1vh; border-
radius: 12px; ">
    
    <div class="card-body" style="background-color: #1F2833; border-
radius: 12px;">
        <h5 class="card-title" style="font-family: 'Big Shoulders Stencil
Display', cursive;
        font-family: 'Londrina Solid', cursive; color: pink; display: flex;
justify-content: center;">Personal Training Program</h5>
        <p class="card-text" style="color: #c5c6c7; display: flex; justify-
content: center;">We have best coaches from each field to train you the
best .</p>
    </div>
</div>
</div>
<!-->
</div>

```

src/app/slide/slide.component.spec.ts

```
import { ComponentFixture, TestBed } from '@angular/core/testing';
import { SlideComponent } from './slide.component';
describe('SlideComponent', () => {
  let component: SlideComponent;
  let fixture: ComponentFixture<SlideComponent>;
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [ SlideComponent ]
    })
    .compileComponents();
  });
  beforeEach(() => {
    fixture = TestBed.createComponent(SlideComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });
  it('should create', () => {
    expect(component).toBeTruthy();
  });
});
```

src\app\slide\slide.component.ts

```
import { Component, OnInit } from '@angular/core';
import { UsersService } from '../users.service';
@Component({
  selector: 'app-slide',
  templateUrl: './slide.component.html'
})
export class SlideComponent implements OnInit {
  constructor(public userSer : UsersService) { }
  ngOnInit(): void {
  }
}
```

src\app\users.service.spec.ts

```
import { TestBed } from '@angular/core/testing';
import { UsersService } from '../users.service';
describe('UsersService', () => {
  let service: UsersService;
  beforeEach(() => {
    TestBed.configureTestingModule({});
    service = TestBed.inject(UsersService);
  });
  it('should be created', () => {
    expect(service).toBeTruthy();
  });
});
```

src\app\users.service.ts

```
import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';
@Injectable({
  providedIn: 'root'
})
export class UsersService {
  ad:string= '1635362908540';
  name:string="";
  constructor(public http : HttpClient,) {
  }
  doUserRegistration(data:any)
  {
    return this.http.post<string>("http:
  }
  doUserLogin(data:any)
  {
    return this.http.post<any[]>("http://localhost:3000/login", data);
  }
  isAdmin(){
    return !!localStorage.getItem("loggedadmin");
  }
  isLoggedIn()
  {
    this.name=localStorage.getItem("loggedusername");
    return !!localStorage.getItem("loggeduser");
  }
  getAllUsers()
  {
    return this.http.get<any[]>("http://localhost:3000/allusers");
  }
  getloggedUserData(userid:string)
  {
    return this.http.get<any[]>("http://localhost:3000/profile/"+userid);
  }
  uemailcheckAvail(uemail:string)
  {
    return this.http.get<any[]>("http://localhost:3000/uemailcheck/"+uemail);
  }
  uuemailcheckAvail(uemail:string)
  {
    return this.http.get<any[]>("http://localhost:3000/uuemailcheck/"+uemail);
  }
  getSingleUserData(userid:string)
  {
    return this.http.get<any[]>("http://localhost:3000/getuser/"+userid);
  }
  editsingleUserdata(data:any)
  {
    return this.http.put<string>("http://localhost:3000/updateuser/",data);
  }
  addpackages(data:any)
  {
    return this.http.put<string>("http://localhost:3000/cart/",data);
  }
  deleteUserData(userid:number)
  {
    return this.http.delete<string>("http://localhost:3000/
deleteuser/"+userid);
  }
}
```



```
    }  
    searchUsers(searchtxt:string)  
    {  
        return this.http.get<any[]>("http://localhost:3000/  
searchuser/"+searchtxt);  
    }  
}
```

```
src\environments\environment.prod.ts
```

```
export const environment = {  
  production: true  
};
```

```
src\environments\environment.ts
```

```
export const environment = {  
  production: false  
};
```

src\index.html

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>YogaFitnessCenter</title>
  <base href="/">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="favicon.ico">
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link href="https://fonts.googleapis.com/css2?family=Big+Shoulders+Stencil+D
isplay:wght@900&family=Londrina+Solid:wght@300;400&display=swap"
rel="stylesheet">
</head>
<body style="background-color: #1F2833;">
  <app-root></app-root>
</body>
</html>
```

src/main.ts

```
import { enableProdMode } from '@angular/core';
import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';
import { AppModule } from './app/app.module';
import { environment } from './environments/environment';
if (environment.production) {
  enableProdMode();
}
platformBrowserDynamic().bootstrapModule(AppModule)
  .catch(err => console.error(err));
```

src\polyfills.ts

```
import 'zone.js';
```

src\styles.css

```
.html{
  background-color: #1F2833;
  font-family: 'Big Shoulders Stencil Display', cursive;
font-family: 'Londrina Solid', cursive;
}
```

src/test.ts

```
import 'zone.js/testing';
import { TestBed } from '@angular/core/testing';
import {
  BrowserDynamicTestingModule,
  platformBrowserDynamicTesting
} from '@angular/platform-browser-dynamic/testing';
declare const require: {
  context(path: string, deep?: boolean, filter?: RegExp): {
    keys(): string[];
    <T>(id: string): T;
  };
};
getTestBed().initTestEnvironment(
  BrowserDynamicTestingModule,
  platformBrowserDynamicTesting(),
  { teardown: { destroyAfterEach: true }},
);
const context = require.context('./', true, /\.spec\.ts$/);
context.keys().map(context);
```


tsconfig.app.json

```
{
  "extends": "../tsconfig.json",
  "compilerOptions": {
    "outDir": "../out-tsc/app",
    "types": []
  },
  "files": [
    "src/main.ts",
    "src/polyfills.ts"
  ],
  "include": [
    "src/**/*.d.ts"
  ]
}
```

tsconfig.json

```
{
  "compileOnSave": false,
  "compilerOptions": {
    "baseUrl": "./",
    "outDir": "./dist/out-tsc",
    "forceConsistentCasingInFileNames": true,
    "strict": true,
    "strictNullChecks": false,
    "strictPropertyInitialization": false,
    "noImplicitReturns": true,
    "noFallthroughCasesInSwitch": true,
    "sourceMap": true,
    "declaration": false,
    "downlevelIteration": true,
    "experimentalDecorators": true,
    "moduleResolution": "node",
    "importHelpers": true,
    "target": "es2017",
    "module": "es2020",
    "lib": [
      "es2018",
      "dom"
    ]
  },
  "angularCompilerOptions": {
    "enableI18nLegacyMessageIdFormat": false,
    "strictInjectionParameters": true,
    "strictInputAccessModifiers": true,
    "strictTemplates": true
  }
}
```

tsconfig.spec.json

```
{
  "extends": "../tsconfig.json",
  "compilerOptions": {
    "outDir": "../out-tsc/spec",
    "types": [
      "jasmine"
    ]
  },
  "files": [
    "src/test.ts",
    "src/polyfills.ts"
  ],
  "include": [
    "src/**/*.spec.ts",
    "src/**/*.d.ts"
  ]
}
```