landing.html

<div class="introtext">

<h1>Intro text</h1>

<h3>

Lorem Ipsum is simply dummy text of the printing and typesetting industry.

Lorem Ipsum has been the industry's standard dummy text ever since the 1500s,

when an unknown printer took a galley of type and scrambled it to make a type specimen book.

It has survived not only five centuries, but also the leap into electronic typesetting,

remaining essentially unchanged.

</h3>

</div>

<div>

<button (click)="navpage()">Apply For Trainer</button>

</div>

landing.ts

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

import { Router } from "@angular/router";

@Component({

selector: 'app-landing-page',

templateUrl: './landing-page.component.html'

})

export class LandingPageComponent implements OnInit {

constructor(private router: Router) { }

ngOnInit() {

}

navpage() {

// Navigate to the specified route when button is clicked

this.router.navigate(['/place-fitness-trainer-appointment']);

}

}

Place.html

<h1>Place Fitness Trainer Appointment</h1>

<form [formGroup]="fitnessForm" (ngSubmit)="onSubmit()">

<div class="form-group">

<label for="firstname">First Name:</label>

<input type="text" class="form-control" id="firstname" formControlName="firstname" required>

</div>

<div class="form-group">

<label for="lastname">Last Name:</label>

<input type="text" class="form-control" id="lastname" formControlName="lastname" required>

</div>

<div class="form-group">

<label for="age">Age:</label>

<input type="number" class="form-control" id="age" formControlName="age" required>

</div>

<div class="form-group">

<label for="phonenumber">Phone Number:</label>

<input type="tel" class="form-control" id="phonenumber" formControlName="phonenumber" required>

</div>

<div class="form-group">

<label for="email">Email:</label>

<input type="email" class="form-control" id="email" formControlName="email" required pattern="[a-z0-9.\_%+-]+@[a-z0-9.-]+\.[a-z]{2,}$">

</div>

<div class="form-group">

<label for="streetaddress">Street Address:</label>

<input type="text" class="form-control" id="streetaddress" formControlName="streetaddress" required>

</div>

<div class="form-group">

<label for="city">City:</label>

<input type="text" class="form-control" id="city" formControlName="city" required>

</div>

<div class="form-group">

<label for="state">State:</label>

<input type="text" class="form-control" id="state" formControlName="state" required>

</div>

<div class="form-group">

<label for="country">Country:</label>

<input type="text" class="form-control" id="country" formControlName="country" required>

</div>

<div class="form-group">

<label for="pincode">Pincode:</label>

<input type="number" class="form-control" id="pincode" formControlName="pincode" required>

</div>

<button type="submit" class="btn btn-primary" [disabled]="fitnessForm.invalid">Submit</button>

</form>

Place.ts

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

import { FormBuilder, FormGroup, Validators } from '@angular/forms';

import {UserService} from '../\_services';

export class Fitness {

constructor(

public inr: number,

public paisa: number,

public streetaddress: string,

public city: string,

public state: string,

public country: string,

public pincode: number,

public phonenumber: number,

public email: string,

public firstname:string,

public lastname: string,

public age:number,

public trainerpreference: string,

public physiotherapist: string,

public packages: string

) { }

}

@Component({

selector: 'app-place-fitness-trainer-appointment',

templateUrl: './place-fitness-trainer-appointment.component.html'

})

export class PlaceFitnessTrainerAppointmentComponent implements OnInit {

fitnessForm: FormGroup;

constructor(private formBuilder: FormBuilder, private userService: UserService) { }

ngOnInit() {

this.fitnessForm = this.formBuilder.group({

inr: ['', Validators.required],

paisa: ['', Validators.required],

streetaddress: ['', Validators.required],

city: ['', Validators.required],

state: ['', Validators.required],

country: ['', Validators.required],

pincode: ['', Validators.required],

phonenumber: ['', Validators.required],

email: ['', [Validators.required, Validators.email]],

firstname: ['', Validators.required],

lastname: ['', Validators.required],

age: ['', Validators.required],

trainerpreference: [''],

physiotherapist: [''],

packages: ['']

});

}

onSubmit() {

this.fitnessForm.value;

console.log(

"LOG: LoginComponent -> onSubmit -> this.fitnessForm.value",

this.fitnessForm.value

);

if (this.fitnessForm.invalid) {

return;

}

const fitnessData = this.fitnessForm.value;

const fitness = new Fitness(

fitnessData.inr,

fitnessData.paisa,

fitnessData.streetaddress,

fitnessData.city,

fitnessData.state,

fitnessData.country,

fitnessData.pincode,

fitnessData.phonenumber,

fitnessData.email,

fitnessData.firstname,

fitnessData.lastname,

fitnessData.age,

fitnessData.trainerpreference,

fitnessData.physiotherapist,

fitnessData.packages

);

this.userService.postfitnessdata(fitness)

.subscribe(

response => {

console.log('Fitness data posted successfully:', response);

this.fitnessForm.reset();

},

error => {

console.error('Error posting fitness data:', error);

}

);

}

}

View.html

<h1>Appointments</h1>

<table class="table table-striped">

<thead>

<tr>

<th scope="col">Sl NO</th>

<th scope="col">Name</th>

<th scope="col">Address</th>

<th scope="col">City</th>

<th scope="col">Package</th>

<th scope="col">Trainer Preference</th>

<th scope="col">Phone</th>

</tr>

</thead>

<tbody>

</tbody>

</table>

View.ts

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

import { HttpClient } from '@angular/common/http';

interface Appointment {

id: number;

name: string;

address: string;

city: string;

package: string;

trainerPreference: string;

phone: string;

}

@Component({

selector: 'app-view-appointment',

templateUrl: './view-appointment.component.html',

})

export class ViewAppointmentComponent implements OnInit {

appointments: Appointment[] = [];

constructor(private http: HttpClient) {}

ngOnInit() {

this.fetchAppointments();

}

fetchAppointments() {

this.http.get<Appointment[]>('http://your-api-endpoint/appointments') // Replace with your actual API endpoint

.subscribe(data => {

this.appointments = data;

});

}

}