Project: Hotel Booking Application

1. Setup Angular Project

1. Ensure you have Angular CLI installed:

   ```bash
   npm install -g @angular/cli
   ```

2. Create a new Angular project:

   ```bash
   ng new hotel-booking-app
   ```

3. Navigate to the project directory:

   ```bash
   cd hotel-booking-app
   ```

4. Serve the application:

   ```bash
   ng serve
   ```

2. Project Structure

```
hotel-booking-app/
│
├── src/
│   ├── app/
│   │   ├── components/
│   │   │   ├── hotel-list/
│   │   │   │   ├── hotel-list.component.html
│   │   │   │   ├── hotel-list.component.ts
│   │   │   │   ├── hotel-list.component.css
│   │   │   ├── hotel-details/
│   │   │   │   ├── hotel-details.component.html
│   │   │   │   ├── hotel-details.component.ts
│   │   │   │   ├── hotel-details.component.css
│   │   │   ├── booking-form/
│   │   │   │   ├── booking-form.component.html
│   │   │   │   ├── booking-form.component.ts
│   │   │   │   ├── booking-form.component.css
│   │   ├── services/
│   │   │   ├── hotel.service.ts
│   │   ├── models/
```

```
│   │   │   ├── hotel.model.ts
│   │   ├── app.component.html
│   │   ├── app.component.ts
│   │   ├── app-routing.module.ts
│   │   ├── app.module.ts
│   │
│   └── ...
```

3. Create Hotel Model

Create a model for hotel data.

hotel.model.ts

```typescript
export interface Hotel {
  id: number;
  name: string;
  location: string;
  price: number;
  rating: number;
  description: string;
```

}
```

#### 4. Create Hotel Service

Create a service to handle HTTP requests to fetch hotel data.

**hotel.service.ts**

```typescript
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { Observable } from 'rxjs';
import { Hotel } from '../models/hotel.model';

@Injectable({
  providedIn: 'root'
})
export class HotelService {
  private apiUrl = 'https://api.example.com/hotels'; // Replace with actual API endpoint

```
  constructor(private http: HttpClient) {}

  getHotels(): Observable<Hotel[]> {
    return this.http.get<Hotel[]>(this.apiUrl);
  }

  getHotelById(id: number): Observable<Hotel> {
    return this.http.get<Hotel>(`${this.apiUrl}/${id}`);
  }
}
```

5. Create Components

Create components for listing hotels, displaying hotel details, and booking form.

hotel-list.component.ts

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

```
import { HotelService } from
'../../services/hotel.service';
import { Hotel } from '../../models/hotel.model';

@Component({
  selector: 'app-hotel-list',
  templateUrl: './hotel-list.component.html',
  styleUrls: ['./hotel-list.component.css']
})
export class HotelListComponent implements OnInit
{
  hotels: Hotel[] = [];

  constructor(private hotelService: HotelService) {}

  ngOnInit(): void {
    this.hotelService.getHotels().subscribe((data:
Hotel[]) => {
      this.hotels = data;
    });
  }
```

}
```

**hotel-list.component.html**

```html
<div class="hotel-list">
  <div *ngFor="let hotel of hotels" class="hotel-item">
    <h3>{{ hotel.name }}</h3>
    <p>{{ hotel.location }}</p>
    <p>\${{ hotel.price }} per night</p>
    <p>Rating: {{ hotel.rating }}</p>
    <button routerLink="/hotel/{{ hotel.id }}">View Details</button>
  </div>
</div>
```

hotel-details.component.ts

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

```
import { ActivatedRoute } from '@angular/router';
import { HotelService } from
'../../services/hotel.service';
import { Hotel } from '../../models/hotel.model';

@Component({
  selector: 'app-hotel-details',
  templateUrl: './hotel-details.component.html',
  styleUrls: ['./hotel-details.component.css']
})
export class HotelDetailsComponent implements
OnInit {
  hotel: Hotel | undefined;

  constructor(
    private route: ActivatedRoute,
    private hotelService: HotelService
  ) {}

  ngOnInit(): void {
```

```
    const id =
Number(this.route.snapshot.paramMap.get('id'));

this.hotelService.getHotelById(id).subscribe((data:
Hotel) => {
      this.hotel = data;
    });
  }
}
```

hotel-details.component.html

```html
<div *ngIf="hotel">
  <h2>{{ hotel.name }}</h2>
  <p>{{ hotel.location }}</p>
  <p>\${{ hotel.price }} per night</p>
  <p>Rating: {{ hotel.rating }}</p>
  <p>{{ hotel.description }}</p>
  <button routerLink="/booking/{{ hotel.id }}">Book
Now</button>
</div>
```

```

**booking-form.component.ts**

```typescript
import { Component, OnInit } from '@angular/core';
import { FormBuilder, FormGroup, Validators } from '@angular/forms';
import { ActivatedRoute } from '@angular/router';

@Component({
  selector: 'app-booking-form',
  templateUrl: './booking-form.component.html',
  styleUrls: ['./booking-form.component.css']
})
export class BookingFormComponent implements OnInit {
  bookingForm: FormGroup;

  constructor(
    private fb: FormBuilder,
    private route: ActivatedRoute
```
```

```
  ) {
    this.bookingForm = this.fb.group({
      name: ['', Validators.required],
      email: ['', [Validators.required, Validators.email]],
      checkInDate: ['', Validators.required],
      checkOutDate: ['', Validators.required],
      guests: [1, [Validators.required,
Validators.min(1)]]
    });
  }

  ngOnInit(): void {}

  onSubmit(): void {
    if (this.bookingForm.valid) {
      console.log('Booking Successful',
this.bookingForm.value);
      alert('Booking Successful');
      this.bookingForm.reset();
    } else {
      alert('Please fill in all required fields');
```

```
  }
 }
}
```

booking-form.component.html

```html
<form [formGroup]="bookingForm" (ngSubmit)="onSubmit()">
 <div>
  <label for="name">Name</label>
  <input id="name" formControlName="name" type="text" />
 </div>
 <div>
  <label for="email">Email</label>
  <input id="email" formControlName="email" type="email" />
 </div>
 <div>
  <label for="checkInDate">Check-in Date</label>
```

```
    <input id="checkInDate" formControlName="checkInDate" type="date" />
  </div>
  <div>
    <label for="checkOutDate">Check-out Date</label>
    <input id="checkOutDate" formControlName="checkOutDate" type="date" />
  </div>
  <div>
    <label for="guests">Number of Guests</label>
    <input id="guests" formControlName="guests" type="number" />
  </div>
  <button type="submit">Book Now</button>
</form>
```

6. Configure Routing

Set up routing for the application.

app-routing.module.ts

```typescript
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { HotelListComponent } from './components/hotel-list/hotel-list.component';
import { HotelDetailsComponent } from './components/hotel-details/hotel-details.component';
import { BookingFormComponent } from './components/booking-form/booking-form.component';

const routes: Routes = [
  { path: '', component: HotelListComponent },
  { path: 'hotel/:id', component: HotelDetailsComponent },
  { path: 'booking/:id', component: BookingFormComponent },
];

@NgModule({
  imports: [RouterModule.forRoot(routes)],
```

```
  exports: [RouterModule]
})
export class AppRoutingModule { }
```

7. Update App Module

Ensure the app module imports necessary modules and declares components.

app.module.ts

```typescript
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { FormsModule, ReactiveFormsModule } from '@angular/forms';
import { HttpClientModule } from '@angular/common/http';

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
```

```
import { HotelListComponent } from
'./components/hotel-list/hotel-list.component';
import { HotelDetailsComponent } from
'./components/hotel-details/hotel-
details.component';
import { BookingFormComponent } from
'./components/booking-form/booking-
form.component';


@NgModule({
  declarations: [
    AppComponent,
    HotelListComponent,
    HotelDetailsComponent,
    Booking


FormComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    FormsModule,
```

```
    ReactiveFormsModule,
    HttpClientModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

8. Run the Application

Start the Angular development server and navigate to `http://localhost:4200` to see your hotel booking application in action.

```bash
ng serve
```