Angular Javascript Frontend Frameworks(Angular)

Submitted By:

Anush Jowin A - 2462044

Duggempudi Praveen Kumar Reddy - 2462066

Gutha Nihitha - 2463021

Darain Brit A – 2462060

- **Institution**: *Christ University*
- **Date of Submission**: *23/01/2026*

Project 12:

Travel Booking and Destination Explorer Application

Github: https://github.com/2326praveen/Travel-Booking-app

Deployment: https://travel-booking-app-2mzl.vercel.app/destinations

1. Introduction

The Travel Booking and Destination Explorer Application is a modern single-page web application (SPA) developed using Angular 21 and TypeScript. The application enables users to explore popular travel destinations, view detailed travel packages, and book trips through an intuitive and responsive interface.

The project demonstrates the practical implementation of Angular's ecosystem, including components, routing, services, dependency injection, reactive forms, RxJS observables, and Angular Material for UI design.

The final application provides a visually rich, user-friendly travel planning experience with booking management and responsive design.

2. Objectives

The main objectives of this project are:

- To design and build a travel booking application using Angular and TypeScript
- To understand Angular's component-based architecture
- To implement routing and navigation in a single-page application
- To use services and dependency injection for data management

- To implement reactive forms with validation for bookings
- To manage application state using RxJS Observables
- To design a modern and responsive UI using Angular Material

3. Scope of the Project

- Develop a single-page travel booking application using Angular
- Display travel destinations with detailed descriptions and ratings
- Show available travel packages with itineraries and pricing
- Implement a booking system with form validation and date selection
- Allow users to view and manage their bookings
- Design a responsive UI using Angular Material
- Use mock data and Local Storage for data persistence

4. Tools & Technologies Used

Tool/Technology	Purpose
Angular	Frontend framework
TypeScript	Strongly typed programming language
VS Code	Code editor
CSS	Styling and layout
JSON Server	Mock backend for songs, artists, and playlists
Node.js	Runtime environment
Angular CLI	Project scaffolding and build tool
Angular Material	UI components and theming

5. System Architecture

The application follows a **modular Angular architecture**:

- **Components** handle UI rendering and user interactions

- **Services** manage business logic and data operations
- **Routing module** enables navigation without page reloads
- **Models (interfaces)** define structured data
- **RxJS Observables** manage application state
- **Angular Material** ensures a consistent and responsive UI

6. Key Features

Feature	Description
Responsive UI Design	Adapts to different screen sizes using Angular Material
Destination Browsing	Displays destinations with images, descriptions, and ratings
Travel Package Viewing	Shows detailed packages with itineraries and pricing
Trip Booking	View and manage bookings in user dashboard
Booking Management	View and manage bookings in user dashboard

7. Angular Components Design

The application is divided into reusable components:

Destination List Component

- Displays all available travel destinations
- Shows destination images, descriptions, and ratings
- Lists popular activities
- Provides navigation to related travel packages

Package Detail Component

- Displays detailed travel package information
- Shows itinerary, inclusions, and exclusions

- Displays available travel dates and pricing
- Includes “Book Now” option

Booking Form Component

- Collects user details (name, email, phone)
- Date picker with availability constraints
- Number of travelers selection
- Real-time price calculation
- Form validation with feedback

User Dashboard Component

- Displays booking history
- Shows booking status (Confirmed, Pending, Cancelled)
- Allows cancellation of bookings
- Displays booking details

Navbar Component

- Provides navigation links (Destinations, Packages, Dashboard)
- Built using Angular Material Toolbar

8. Services and Dependency Injection

DestinationService

- Fetches destination data from mock JSON
- Provides destination filtering and retrieval

PackageService

- Manages travel package data
- Retrieves packages based on destination

BookingService

- Handles booking creation and cancellation
- Stores and retrieves bookings from LocalStorage

Services are injected using **Angular Dependency Injection** to ensure loose coupling and reusability.

9. Angular Material and UI Design

The UI uses Angular Material components:

- **MatToolbar** – Navigation bar
- **MatCard** – Destination and package previews
- **MatDatepicker** – Travel date selection
- **MatFormField** – Booking form inputs
- **MatButton** – User actions
- **MatGridList** – Responsive layouts

Material theming ensures:

- Clean and modern design
- Fully responsive layouts
- Consistent styling across components

10. Challenges Faced & Solutions

Challenge	Solution
Managing booking state	Used centralized BookingService with RxJS
Form validation issues	Implemented Reactive Forms with validators
UI responsiveness issues	Applied Angular Material responsive layouts
Data persistence	Used LocalStorage for saving bookings

11. Outcome

- Fully functional Angular travel booking application
- Modular and scalable architecture

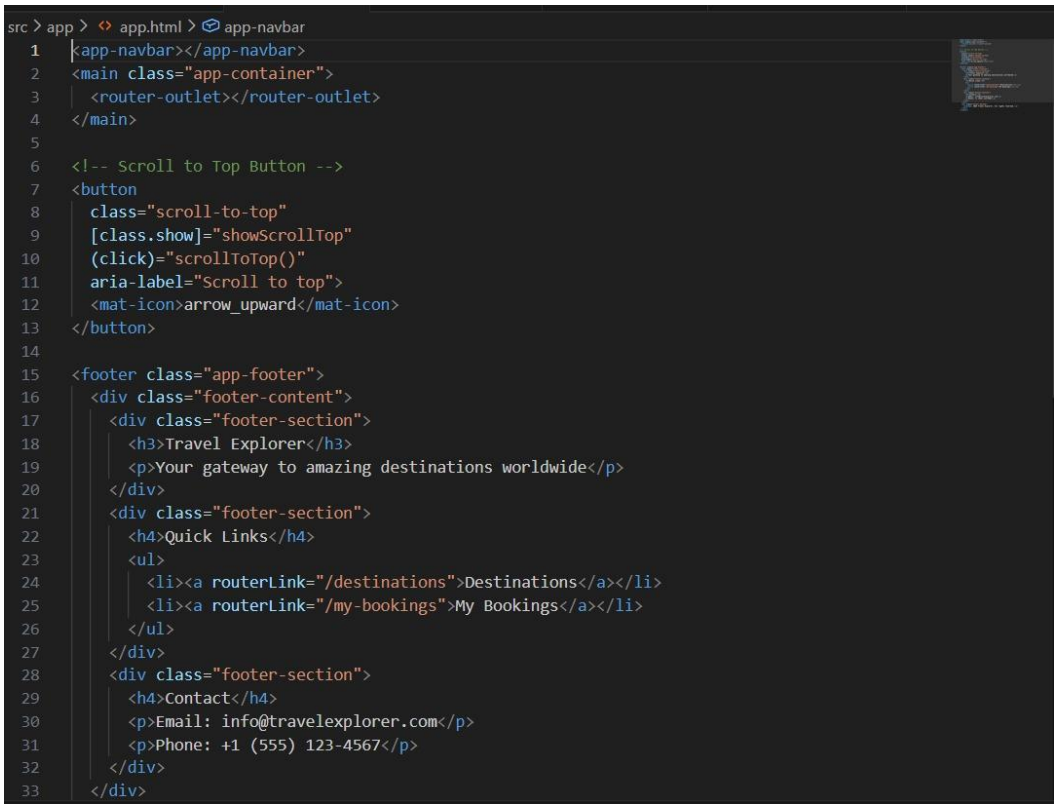
- Responsive UI built with Angular Material
- Booking system with validation and persistence
- Destination and package exploration features
- Unit testing using Vitest
- Well-documented project structure and setup

12. Future Enhancements

- Integration with a real backend and database
- User authentication and personalized accounts
- Online payment gateway integration
- Advanced search and filtering options
- Travel recommendations based on user preferences

Map-based destination exploration

13.Sample Code



src > app > # app.css > :host

```css
:host {
  display: flex;
  flex-direction: column;
  min-height: 100vh;
}

.app-container {
  flex: 1;
  position: relative;
  min-height: calc(100vh - 64px - 200px);
  background: linear-gradient(rgba(255, 255, 255, 0.25), rgba(255, 255, 255, 0.3)),
              url('https://images.unsplash.com/photo-1488646953014-85cb44e25828?w=1920&q=90&au
}

.app-container router-outlet {
  display: block;
}

.app-footer {
  background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
  color: white;
  padding: 40px 20px 20px;
  margin-top: auto;
}

.footer-content {
  max-width: 1200px;
  margin: 0 auto;
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
  gap: 30px;
  margin-bottom: 20px;
```

rc > app > # app.css > :host

```css
:host {
  display: flex;
  flex-direction: column;
  min-height: 100vh;
}

.app-container {
  flex: 1;
  position: relative;
  min-height: calc(100vh - 64px - 200px);
  background: linear-gradient(rgba(255, 255, 255, 0.25), rgba(255, 255, 255, 0.3)),
              url('https://images.unsplash.com/photo-1488646953014-85cb44e25828?w=1920&q=90&au
}

.app-container router-outlet {
  display: block;
}

.app-footer {
  background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
  color: white;
  padding: 40px 20px 20px;
  margin-top: auto;
}

.footer-content {
  max-width: 1200px;
  margin: 0 auto;
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
  gap: 30px;
  margin-bottom: 20px;
}
```

src > app > TS app.config.ts > ...

```ts
import { ApplicationConfig } from '@angular/core';
import { provideRouter } from '@angular/router';
import { provideAnimationsAsync } from '@angular/platform-browser/animations/async';

import { routes } from './app.routes';

export const appConfig: ApplicationConfig = {
  providers: [
    provideRouter(routes),
    provideAnimationsAsync()
  ]
};

```

src > app > TS app.ts > ...

```ts
import { Component, signal, OnInit, HostListener } from '@angular/core';
import { RouterOutlet, RouterLink } from '@angular/router';
import { CommonModule } from '@angular/common';
import { MatIconModule } from '@angular/material/icon';
import { NavbarComponent } from './components/navbar/navbar';

@Component({
  selector: 'app-root',
  imports: [RouterOutlet, RouterLink, NavbarComponent, CommonModule, MatIconModule],
  templateUrl: './app.html',
  styleUrl: './app.css'
})
export class App implements OnInit {
  protected readonly title = signal('travel-booking-app');
  showScrollTop = false;

  ngOnInit(): void {
    // Initial check for scroll position
    this.checkScrollPosition();
  }

  @HostListener('window:scroll', [])
  onWindowScroll(): void {
    this.checkScrollPosition();
  }

  private checkScrollPosition(): void {
    this.showScrollTop = window.scrollY > 300;
  }

  scrollToTop(): void {
    window.scrollTo({
      top: 0,
```

src > app > app.html > app-navbar

```
<app-navbar></app-navbar>
<main class="app-container">
  <router-outlet></router-outlet>
</main>

<!-- Scroll to Top Button -->
<button
  class="scroll-to-top"
  [class.show]="showScrollTop"
  (click)="scrollToTop()"
  aria-label="Scroll to top">
  <mat-icon>arrow_upward</mat-icon>
</button>

<footer class="app-footer">
  <div class="footer-content">
    <div class="footer-section">
      <h3>Travel Explorer</h3>
      <p>Your gateway to amazing destinations worldwide</p>
    </div>
    <div class="footer-section">
      <h4>Quick Links</h4>
      <ul>
        <li><a routerLink="/destinations">Destinations</a></li>
        <li><a routerLink="/my-bookings">My Bookings</a></li>
      </ul>
    </div>
    <div class="footer-section">
      <h4>Contact</h4>
      <p>Email: info@travelexplorer.com</p>
      <p>Phone: +1 (555) 123-4567</p>
    </div>
  </div>
```

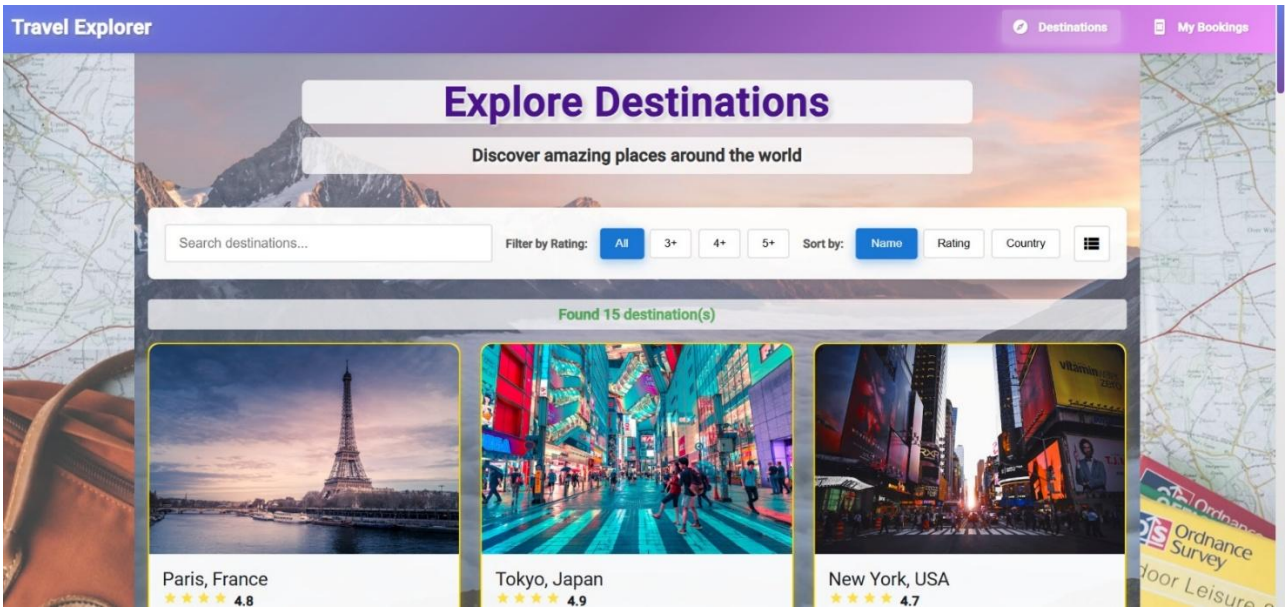
src > app > app.routes.ts > ...

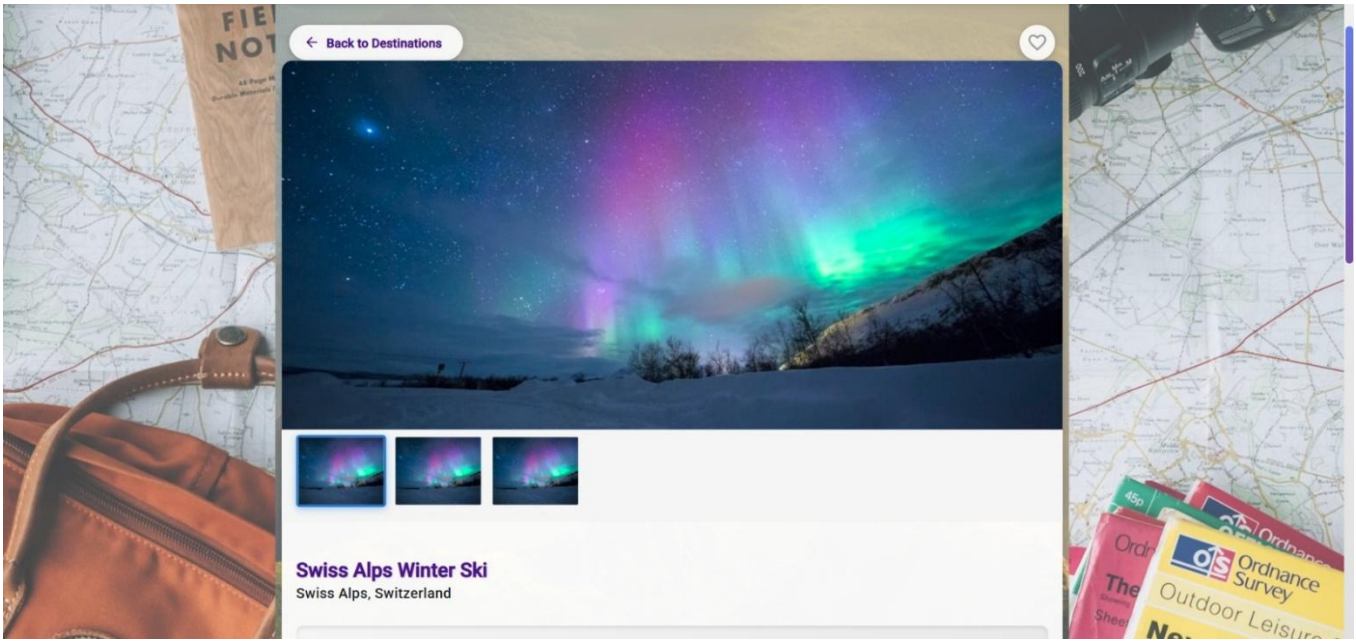
```
import { Routes } from '@angular/router';
import { DestinationListComponent } from './components/destination-list/destination-list';
import { PackageDetailComponent } from './components/package-detail/package-detail';
import { BookingFormComponent } from './components/booking-form/booking-form';
import { UserDashboardComponent } from './components/user-dashboard/user-dashboard';

export const routes: Routes = [
  { path: '', redirectTo: '/destinations', pathMatch: 'full' },
  { path: 'destinations', component: DestinationListComponent },
  { path: 'package/:id', component: PackageDetailComponent },
  { path: 'booking/:packageId', component: BookingFormComponent },
  { path: 'my-bookings', component: UserDashboardComponent }
];

```

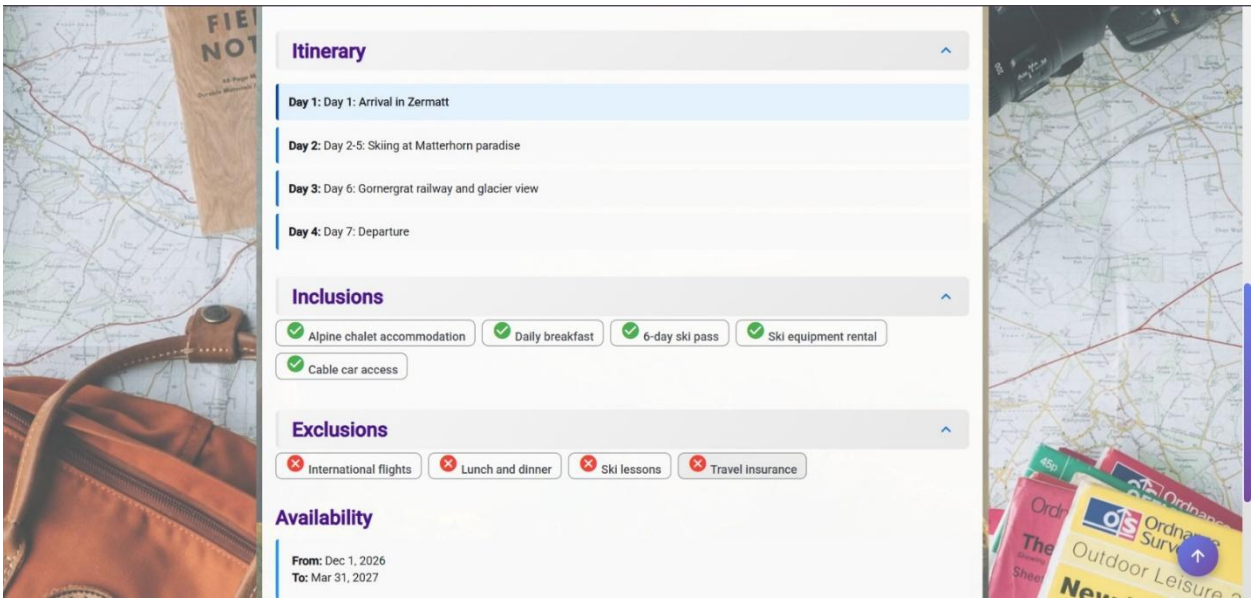
src > app > TS app.spec.ts > ...

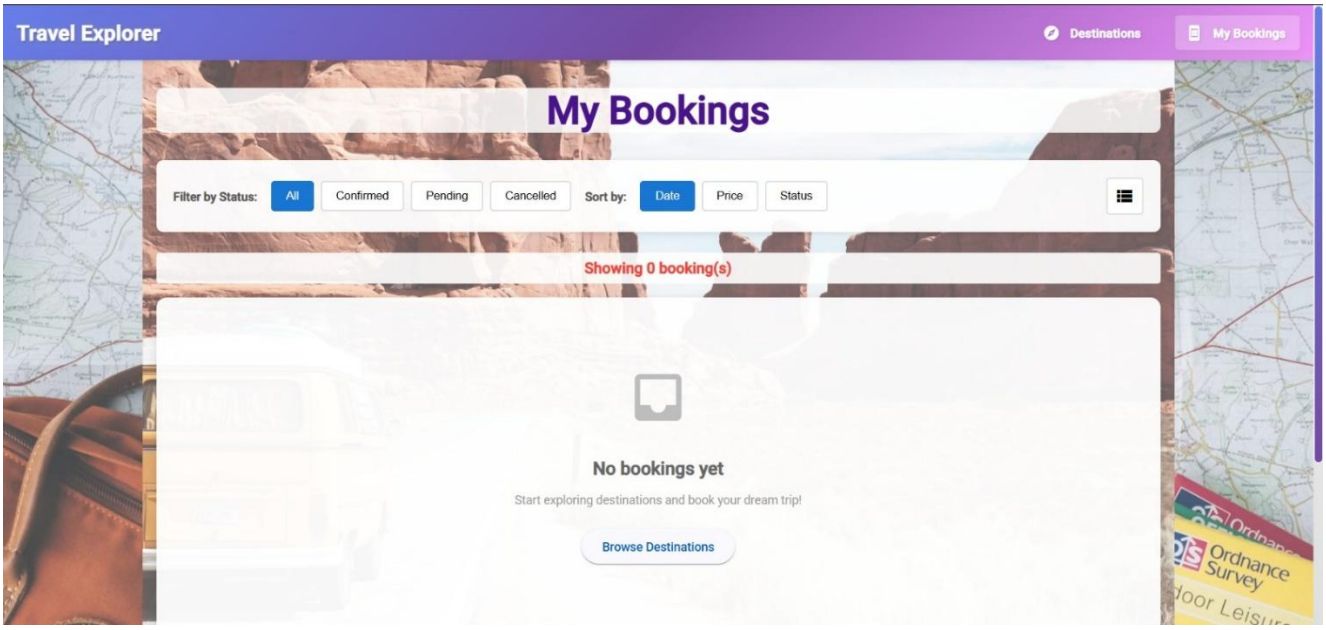
```ts
import { TestBed } from '@angular/core/testing';
import { App } from './app';

describe('App', () => {
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      imports: [App],
    }).compileComponents();
  });

  it('should create the app', () => {
    const fixture = TestBed.createComponent(App);
    const app = fixture.componentInstance;
    expect(app).toBeTruthy();
  });

  it('should render title', async () => {
    const fixture = TestBed.createComponent(App);
    await fixture.whenStable();
    const compiled = fixture.nativeElement as HTMLElement;
    expect(compiled.querySelector('h1')?.textContent).toContain('Hello, travel-booking-app');
  });
});

```

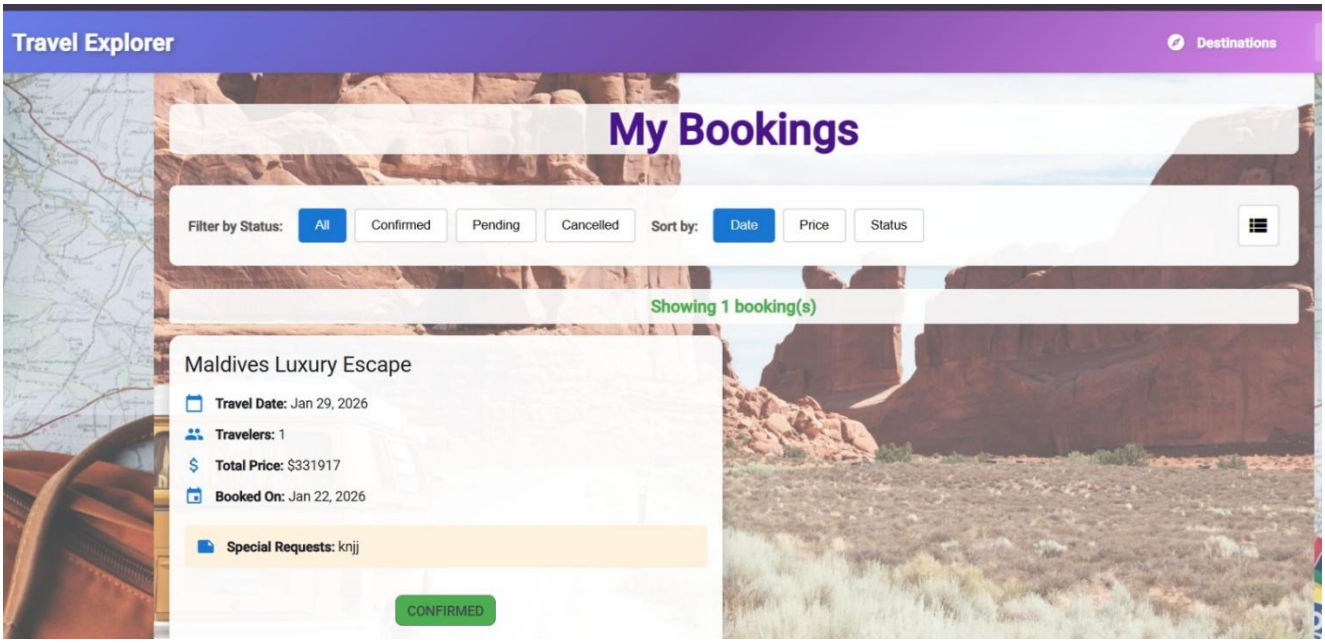
14. Output




Back to Destinations
Swiss Alps Winter Ski
Swiss Alps, Switzerland


Itinerary
Day 1: Day 1: Arrival in Zermatt
Day 2: Day 2-5: Skiing at Matterhorn paradise
Day 3: Day 6: Gornergrat railway and glacier view
Day 4: Day 7: Departure
Inclusions
Alpine chalet accommodation
Daily breakfast
6-day ski pass
Ski equipment rental
Cable car access
Exclusions
International flights
Lunch and dinner
Ski lessons
Travel insurance
Availability
From: Dec 1, 2026
To: Mar 31, 2027


Travel Explorer
Destinations
My Bookings
My Bookings
Filter by Status:
All
Confirmed
Pending
Cancelled
Sort by:
Date
Price
Status
Showing 0 booking(s)
No bookings yet
Start exploring destinations and book your dream trip!
Browse Destinations
Ordnance Survey


Travel Explorer
Destinations
My Bookings
Filter by Status:
All
Confirmed
Pending
Cancelled
Sort by:
Date
Price
Status
Showing 1 booking(s)
Maldives Luxury Escape
Travel Date: Jan 29, 2026
Travelers: 1
Total Price: $331917
Booked On: Jan 22, 2026
Special Requests: knjj
CONFIRMED

15. Conclusion

The Travel Booking and Destination Explorer Application successfully demonstrates the effective use of Angular 21 and TypeScript to develop a scalable, modular, and user-friendly single-page application. Throughout this project, core Angular concepts such as component-based architecture, routing, services, dependency injection, reactive forms, RxJS observables, and Angular Material were implemented to create a responsive and interactive travel booking platform.

The application provides users with a seamless experience for exploring destinations, viewing travel packages, and booking trips, while maintaining clean code structure and efficient state management. This project serves as a strong practical foundation for building real-world Angular applications in the travel and tourism domain.

16. References

- L&T LMS – https://learn.lntedutech.com/Landing/MyCourse
- Angular Official Documentation – https://angular.io/docs
- Angular Material Documentation – https://material.angular.io
- TypeScript Documentation – https://www.typescriptlang.org/docs