**Angular Assignment Documentation: Personalized Travel Planner**

**Project Overview**

The **Personalized Travel Planner** is a single-page web application designed to enable users to plan and manage their travel itineraries. It provides personalized destination recommendations, a budget management system, and an interactive itinerary planner. Built using Angular, the project follows modular architecture for scalability and maintainability.

**Features and Angular Concepts Used**

**1. Setup and Structure**

* **Modules:**
  + Core Module: Contains global services (e.g., AuthService) and components like the Navbar.
  + Shared Module: Reusable components (e.g., truncateDescription pipe) and directives (highlightFavorite).
  + Feature Modules: Separate modules for Destinations, Itineraries, Budgeting, and Authentication.
* **Angular Concepts:**
  + Lazy loading of feature modules.
  + Dependency Injection for services.
  + Routing with child routes for feature modules.

**2. User Authentication**

* **Functionality:**
  + Mock login system using a fake API (http://localhost:3000/signupUsersList).
  + Auth Guard restricts access to authenticated users.
* **Angular Concepts:**
  + Services (AuthService) with BehaviorSubject for managing login state.
  + Guards (AuthGuard) to protect routes.
  + Reactive forms for handling login and signup forms.

**3. Destinations Page**

* **Functionality:**
  + Displays a list of destinations with filtering by type, budget, and rating.
  + Users can review and rate destinations.
* **Angular Concepts:**
  + **HTTP Client:** Fetches destination data from a mock JSON API.
  + **Data Binding:** For dynamically rendering destination details.
  + **Pipes:** For filtering and formatting destination lists.
  + **Directives:** Custom directives to highlight favorite destinations.

**4. Create Itinerary Feature**

* **Functionality:**
  + Allows users to plan day-by-day activities using a form.
  + Drag-and-drop for reordering activities.
  + Validates activity dates within selected travel dates.
* **Angular Concepts:**
  + **Reactive Forms:** For dynamic and validated forms.
  + **DragDropModule:** Enables drag-and-drop functionality.
  + **Data Binding:** Reflects activity changes dynamically.
  + **Custom Validation:** Validates activity dates.

**5. Budget Management**

* **Functionality:**
  + Users can set budgets, add expenses, and categorize spending.
  + Dynamic budget tracking and visualization.
* **Angular Concepts:**
  + **Two-Way Binding:** Tracks budget and expenses.
  + **Custom Pipes:** Formats currency values.
  + **ng2-charts (optional):** Visualizes spending breakdown.

**6. Explore Page with Recommendations**

* **Functionality:**
  + Personalized travel recommendations based on user's selected destinations and travel history.
* **Angular Concepts:**
  + **Services:** Business logic for generating recommendations.
  + **Observable Streams:** Updates recommendations in real-time.
  + **Dynamic Routing:** Links users to recommended activities.

**7. Sharing Itinerary**

* **Functionality:**
  + Generates shareable links for itineraries.
  + Routes to shared itineraries using unique identifiers.
* **Angular Concepts:**
  + **Router:** Manages dynamic routes for shared itineraries.
  + **ActivatedRoute:** Reads query parameters for shared itinerary IDs.

**8. Custom Directives and Pipes**

* **Directives:**
  + highlightFavorite: Highlights user-favorite destinations.
* **Pipes:**
  + truncateDescription: Truncates long descriptions to improve UI consistency.
* **Angular Concepts:**
  + Structural directives for DOM manipulation.
  + Transform pipes for formatting.

**9. Authentication Guard**

* **Functionality:**
  + Restricts access to certain features unless authenticated.
* **Angular Concepts:**
  + Route Guards (CanActivate) to redirect unauthenticated users.
  + Router for navigation and redirection.

**10. Unit Testing**

* **Focus Areas:**
  + **Critical Components:** ItineraryPlannerComponent and BudgetTrackerComponent.
  + **Services:** Test business logic in AuthService.
  + **Custom Features:** Validate highlightFavorite directive and truncateDescription pipe.
* **Angular Concepts:**
  + Test utilities like TestBed for configuring tests.
  + HttpClientTestingModule for mocking HTTP requests.
  + SpyOn to mock and track method calls.

**Testing Approach**

**Unit Testing**

* Framework: Jasmine with Angular TestBed.
* Coverage Areas:
  + Form validation and state updates.
  + HTTP requests and response handling.
  + Component lifecycle hooks (e.g., ngOnInit).

**Bonus Features**

1. **Offline Mode:**
   * Store itineraries in localStorage or IndexedDB when offline.
   * Sync data with the server when online.
2. **Push Notifications:**
   * Use ServiceWorkerModule for notification reminders.
3. **Dark Mode:**
   * Implement using @angular/material's theming capabilities.