**System Plan Document**

**Project**: Trip Nest

**Date:** June 3, 2025

**Author:** John Patrick Trinidad

**1. Project Overview**

* **Project Name:**
* **Description:**  
  A web app that allows users to explore tourist spots worldwide by filtering locations through country, province/state, and city. The app displays tourist spots as the main results and provides detailed info, weather, and maps for each spot.
* **Goals:**
  + Enable users to filter tourist spots by country, province/state, and city globally.
  + Display detailed tourist spot information, including images, description, and location.
  + Provide weather info for tourist spots.
  + Use APIs to get location data, tourist spots, and weather.

**2. Scope**

* **In Scope:**
  + Location filters: country, province/state, city.
  + Display filtered tourist spots list based on selected filters.
  + Tourist spot detail pages with info, images, weather, and map.
  + API integration with GeoDB Cities (for location data), travel APIs (Amadeus, Google Places, etc. for tourist spots), and OpenWeatherMap (weather).
  + Responsive design.
* **Out of Scope:**
  + Ticket or hotel booking.
  + User accounts (optional for future).
  + Offline use.

**3. Modules**

| **Module Name** | **Description** | **Key Features / Components** |
| --- | --- | --- |
| **Landing Page** | Welcome users with search and featured tourist spots | Search bar, featured tourist spots carousel |
| **Explore Page** | Filter tourist spots by country, province/state, city | Filters (country → province/state → city), tourist spot list/grid |
| **Tourist Spot Detail Page** | Detailed info for selected tourist spot | Description, photos, weather widget, map location, opening hours, reviews (if available) |
| **Favorites Page** (Optional) | User’s saved tourist spots | List of favorites, remove/edit favorites |
| **About Page** | App info and developer contact | Static info page |
| **Error Page** | Friendly error messages | 404, API errors, etc. |

**4. System Architecture**

* **Frontend:**
  + React.js with hooks for UI and state management.
  + Components for filters, tourist spot cards, detail views, weather, and map.
* **Backend:** (optional)
  + Node.js + Express for proxying API calls or business logic.
* **APIs Used:**
  + GeoDB Cities API (location filters)
  + Google Places API / Amadeus API / TripAdvisor API (tourist spots and images)
  + OpenWeatherMap API (weather info)
  + Map services (Leaflet, Mapbox, Google Maps)

**5. Features**

| **Feature** | **Description** | **Priority** |
| --- | --- | --- |
| Location Filtering | Filter tourist spots by country, province/state, and city | High |
| Tourist Spot Listing | Show tourist spots matching filter s  elections | High |
| Tourist Spot Details | Detailed info page per tourist spot with images, map, and weather | High |
| Image Handling | Display photos from APIs or fallback images | Medium |
| Map Integration | Show tourist spot location on an interactive map | Medium |
| Favorites/Wishlist | Save favorite tourist spots for easy access | Low |

**6. Data Flow**

* User selects country → filters by province/state → filters by city → app fetches and displays matching tourist spots → user clicks on a tourist spot → app displays detailed info (description, images, weather, map).

**7. UI/UX Design**

* Mobile-first, clean design
* Hierarchical filters: Country → Province/State → City
* List or card view of tourist spots as main result
* Detail page for tourist spots with image gallery, map, and weather widget

**8. Technical Requirements**

* React.js frontend
* Node.js + Express backend (optional)
* Axios or Fetch API for REST API calls
* Environment variables for API keys
* Responsive CSS (Tailwind, Bootstrap, or custom)