Documentation: Geo Atlas

IT22122414 – Sembukuttiarachchi J S

This documentation provides an overview of how to use the Geo Atlas app, the API endpoints it uses, and the challenges faced during development.

Table of Contents

Overview

Geo Atlas is a web application that allows users to:

- Search for countries and view detailed information.
- Add and manage their favorite countries.
- Register, log in, and manage their accounts.

The app is built using React, Vite, and Firebase for backend services (Firestore and Authentication).

How to Use the App

1. Home Page

- The home page displays a search bar and a list of countries.
- Users can search for a country by name or scroll through the list.
- Clicking on a country redirects to the **Country Details** page.

2. Country Details Page

- Displays detailed information about the selected country, including:
- Name, population, region, subregion, capital, and flag.
- Users can add the country to their favorites by clicking the "Add to Favorites" button.

3. Favorites Management

- Users can view their favorite countries in the **Favorites** section.
- They can remove a country from their favorites by clicking the "Remove" button.

4. Authentication

- Users can register for an account using their email and password.
- After logging in, users can manage their session and access personalized features like favorites.

API Endpoints

The app uses Firebase Firestore as the backend. Below are the key API endpoints and their uses:

1. Get Favorites

- **Endpoint**: GET /favorites
- **Description**: Fetches the list of favorite countries for the authenticated user.

2. Add Favorite

- **Endpoint**: POST /favorites
- **Description**: Adds a country to the user's favorites.
- **Implementation**:

3. Remove Favorite

- **Endpoint**: DELETE /favorites
- **Description**: Removes a country from the user's favorites.

4. Authentication

- **Register**: Firebase Authentication handles user registration.
- **Login**: Firebase Authentication validates user credentials.
- **Logout**: Firebase Authentication ends the user session.

Challenges Faced

1. Firestore Security Rules

- **Problem**: Encountered Missing or insufficient permissions errors when accessing Firestore.
- **Solution**: Updated Firestore rules to ensure authenticated users can only access their own data:

```
rules_version = '2';
service cloud.firestore {
 match /databases/{database}/documents {
  match /favorites/{document} {
   allow read, write: if request.auth != null && request.auth.uid == resource.data.userId;
  }
 }
}
```

2. Handling Authentication State

- **Problem**: Managing user sessions and ensuring the app reacts to authentication state changes.
- **Solution**: Used Firebase's onAuthStateChanged to track user authentication status and update the UI accordingly.

3. Favorites Duplication

- **Problem**: Users could add the same country to their favorites multiple times.
- **Solution**: Added a check before adding a favorite to ensure it doesn't already exist:

```
const existsQuery = query(
 collection(db, "favorites"),
 where("userId", "==", userId),
 where("countryCode", "==", countryCode)
);
const existsSnapshot = await getDocs(existsQuery);
if (!existsSnapshot.empty) {
 return { success: false, message: "Already in favorites" };
}
```