

Documentation: WorldView

IT22111906 – Marambe R T

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

Table of Contents

- Overview
- How to Use the App
- API Endpoints
- Challenges Faced

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. 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.

2. 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.

3. 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

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.

3. Remove Favorite

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

Challenges Faced

1. Vite and Testing Setup

- **Problem:** Encountered `jest is not defined` error when trying to run tests with Jest.
Solution: Realized that Jest isn't natively supported in Vite environments, so switched to Vitest, which is designed to work seamlessly with Vite.
- **Problem:** `jest.fn()` was undefined after switching test runners.
Solution: Replaced all instances of `jest.fn()` with `vi.fn()`, the equivalent mocking function provided by Vitest.
- **Problem:** Test environment lacked DOM support needed for rendering React components.
Solution: Configured environment: `'jsdom'` in the Vitest setup to simulate a browser environment.
- **Problem:** Needed proper Vite configuration to enable testing.
Solution: Updated `vite.config.ts` using `defineConfig({ test: { ... } })` to define test globals, environment, and setup files.
- **Problem:** Initial unfamiliarity with Vitest syntax and configuration.
Solution: Read Vitest documentation and explored examples to adapt quickly and set up working test cases.