**Project Specification Document**

Name : Bar Harush

Phone number : 0542562988

**Project Overview**

This project is a web application designed to display Starbucks locations on a map. The application uses OpenLayers for rendering the map and Turf.js for geospatial operations. The project is containerized using Docker to ensure consistency across different environments.

**Project Structure**

תמונה שמכילה טקסט, צילום מסך, גופן, מספר

התיאור נוצר באופן אוטומטיThe project is organized as follows:

**Functionality**

**Data Sources**

* **Starbucks Locations**:
  + URL:https://raw.githubusercontent.com/mmcloughlin/starbucks/master/locations.json
* **Countries Boundaries**:
  + URL: https://raw.githubusercontent.com/datasets/geo-countries/master/data/countries.geojson
* **Countries Names and Codes**:
  + URL:https://gist.githubusercontent.com/keeguon/2310008/raw/bdc2ce1c1e3f28f9cab5b4393c7549f38361be4e/countries.json

**Features**

1. **Map Initialization**:
   * Uses OpenLayers to initialize the map with OpenStreetMap as the base layer.
   * Centers the map at coordinates [0, 0] with a zoom level of 2.
2. **Data Fetching and Parsing**:
   * Fetches and parses country names, boundaries, and Starbucks locations.
   * Decodes HTML entities and fixes JSON format issues as needed.
3. **Displaying Starbucks Locations**:
   * Adds Starbucks locations to the map.
   * Filters locations based on the selected country.
4. **Handling No Locations**:
   * Displays a message on the map if no Starbucks locations are found in the selected country.
5. **Dynamic Updates**:
   * Updates the list of countries and the dropdown menu based on fetched data.
   * Reacts to user selections and updates the map accordingly.

**Design and Image Selection**

The design of the application was intentionally kept simple to align with Starbucks' branding and user expectations. The color scheme includes Starbucks' signature green (#005127 for the header and #007a33 for headings) combined with neutral tones like white and light gray (#f4f4f4) to create a clean and professional look.

The layout is minimalistic, ensuring a user-friendly experience:

* **Header**: A green background with a centered logo.
* **Controls**: A white background with rounded corners and a subtle shadow for a polished appearance.
* **Map**: Takes up 60% of the viewport height for optimal visibility.
* **Country List**: Styled with a card-like appearance, featuring a white background and rounded corners to keep the interface consistent and inviting.

Images used on the map are specifically selected to be easily recognizable and relevant to the Starbucks brand. The overall design focuses on clarity, readability, and an intuitive user interface.

**Initial Issue**

The first issue encountered in the project was understanding the differences between ISO\_A2 and ISO\_A3 codes and how they affected the data’s behavior. ISO\_A2 codes are 2-letter country codes (e.g., "US" for the United States), while ISO\_A3 codes are 3-letter codes (e.g., "USA" for the United States). The project’s data was based entirely on ISO\_A2 codes.

**Solution**: After realizing that the expected results were not being achieved, time was invested in learning the differences between ISO\_A2 and ISO\_A3 codes. It was discovered that all the relevant data was based on ISO\_A2, the 2-letter country codes. ISO\_A2 is widely used for defining country codes in various datasets, making it a standard choice for geospatial data processing. Understanding this distinction allowed for the correct filtering and handling of country-based data, ensuring that Starbucks locations were displayed correctly on the map.

**Docker Configuration**

* **Dockerfile**: Defines how to build the Docker image for the project.
* **docker-compose.yml**: Manages the configuration and orchestration of Docker containers.
* **my-server-image.tar**: The built Docker image for the server.

**Issue Resolution**

The issue with displaying Starbucks locations on the map in Docker has been resolved. All data is now correctly displayed and integrated within the Docker environment.

**Code Overview**

Here is a summary of the provided JavaScript code:

* **Data Fetching Functions**:
  + fetchCountries(): Fetches and processes country data.
  + fetch(locationsUrl): Fetches and processes Starbucks location data.
  + fetch(countriesGeoJsonUrl): Fetches and processes countries GeoJSON data.
* **Map Functions**:
  + clearLayers(): Clears all non-base map layers.
  + addNoLocationsMessage(): Adds a message to the map if no Starbucks locations are found.
  + addLocationsToMap(): Adds Starbucks locations to the map and centers the map based on the locations.
  + removeExtraQuotes(): Removes extra quotes from country codes.
  + calculateCenter(): Calculates the average location of the Starbucks points to center the map.
* **Event Handling**:
  + Handles changes in the country selection dropdown and updates the map accordingly.

**Screenshot**

Local(without docker)

תמונה שמכילה טקסט, צילום מסך, מפה

התיאור נוצר באופן אוטומטי

When I choose country (with Starbucks)

תמונה שמכילה טקסט, צילום מסך, מפה

התיאור נוצר באופן אוטומטי

When I choose country (without Starbucks)

תמונה שמכילה טקסט, צילום מסך, מפה

התיאור נוצר באופן אוטומטי

With docker



תמונה שמכילה טקסט, תוכנה, תכונות מולטימדיה, מולטימדיה

התיאור נוצר באופן אוטומטי

תמונה שמכילה טקסט, תוכנה, סמל מחשב, מערכת הפעלה

התיאור נוצר באופן אוטומטי

תמונה שמכילה טקסט, תוכנה, סמל מחשב, דף אינטרנט

התיאור נוצר באופן אוטומטי

תמונה שמכילה טקסט, צילום מסך, מפה

התיאור נוצר באופן אוטומטי

תמונה שמכילה טקסט, מפה, צילום מסך, אטלס

התיאור נוצר באופן אוטומטי