

aDMe Assignment

September 2024

This doc consists of an assignment question and the instructions to do and submit it.

How to Submit

Once done :

- Create a GitHub repository (don't use **aDMe/Truflect/advertyzement** in the repo name or anywhere in repo).
- Share the link with us
- This repository should contain a readme file describing the method you have used to solve the problem.
- Time taken to complete this assignment.

Let us know if you have any doubts regarding the assignment.

Note :

- These assignments are designed purely for the evaluations purpose and any part of your submission won't be used in the actual product we are working on.
- Using the **advertyzement**, **Adme** or any related name in the repo or in the code will lead to direct rejection.

Assignment

Build a Food Product Explorer using OpenFoodFacts API

Objective: Create a web application that allows users to search, filter, and view detailed information about food products using the OpenFoodFacts API. The user should be able to filter products by category, and sort them based on various criteria (e.g., product name, ingredients, nutritional score).

Requirements:

Technologies:

- **Front-end:** ReactJS (or Next.js for server-side rendering if needed)
 - **Styling:** CSS or any CSS framework (Bootstrap, TailwindCSS)
 - **API Integration:** Use the OpenFoodFacts API to fetch and display product data
 - **Optional:** State management (e.g., Redux or React Context API) for managing global states (filters, cart, etc.)
-

Features:

1. Homepage:

- Display a list of food products fetched from the OpenFoodFacts API.
- Each product should display key information like:
 - Product name
 - Image
 - Category
 - Ingredients (if available)
 - Nutrition Grade (A, B, C, D, E)

- Users should be able to paginate through the product list (using **infinite scroll** or **load more** functionality).

2. Search Functionality:

- Add a search bar on the homepage where users can search for food products by name.
- The search should filter the product list based on the search query.

3. Barcode Search Functionality:

- Add a search bar on the homepage where users can search for food products by Barcode.

4. Category Filter:

- Include a dropdown or a side filter where users can filter products by category (e.g., beverages, dairy, snacks).
- Fetch the list of categories from the OpenFoodFacts API.

5. Sort Functionality:

- Allow users to sort the product list by:
 - Product name (A-Z, Z-A)
 - Nutrition grade (ascending/descending)

6. Product Detail Page:

- When a user clicks on a product, they should be redirected to a product detail page.
- The product detail page should display:
 - Product image
 - Full list of ingredients
 - Nutritional values (energy, fat, carbs, proteins, etc.)
 - Labels (vegan, gluten-free, etc.)

7. Responsive Design:

- Ensure the design is fully responsive and works well on both mobile and desktop screens.
-

API Documentation:

- Base URL: <https://world.openfoodfacts.org/>
- Sample Endpoints:
 - Get products by category:
<https://world.openfoodfacts.org/category/{category}.json>
 - Search products by name:
https://world.openfoodfacts.org/cgi/search.pl?search_terms={name}&json=true
 - Get product details by barcode:
<https://world.openfoodfacts.org/api/v0/product/{barcode}.json>
- Example Query:
 - <https://world.openfoodfacts.org/api/v0/product/737628064502.json> (retrieves detailed product information for a specific product by barcode)

Note : <https://world.openfoodfacts.org/> is an external website maintained by open food facts (an french non profit organization), if server not responding then wait for some time before trying again

Evaluation Criteria:

1. **Code Quality:** Clean, readable, and modular code structure. Make sure to use best practices such as component-based architecture in React.
2. **API Integration:** Effective use of the OpenFoodFacts API to fetch and display data.
3. **UI/UX:** A well-designed, user-friendly interface with responsive design.
4. **Functionality:** Correct implementation of the search, filtering, and sorting features.
5. **Pagination:** Infinite scrolling or load-more functionality should be smooth and performant.
6. **Bonus:** Implementation of cart functionality and state management.