Building Product and Profile Page for TradeWithUs Platform

Overview

TradeWithUs is an app for B2B Traders. The trader could be a buyer or a seller. A seller can register and list their products on the platform, while a buyer can register and find sellers on the platform.

Goal

1. Come up with a backend to store and manage the profiles of traders and the product catalog. The scope of the backend is mentioned in the Backend Scope Section.
2. Design the frontend as per the designs provided in the Design section in this document.

Backend Scope

The backend service will power the Profile and Catalog pages of the application. It will be responsible for handling managing product and profile data and ensuring efficient data retrieval. You may create one single backend with clear separation of APIs and responsibilities across the two Profile and Catalog Services. You may create the data yourself using the APIs and provide the API call(or curl) commands in the README for the sake of running the application.

1. Catalog Management Service
    a. This service will be responsible for managing products and their attributes like name, images, descriptions
    b. Ensure data consistency through transactions where necessary.
    c. The service should expose the below APIs
        i. GET /product/{productId}
            1. This returns the product details
        ii. GET /product/all
            1. This returns all the products in the store
        iii. POST /product
            1. This will create the product in the data store and return the product id
        iv. PUT /product/{productId}
            1. This will update the product the data store

   d. Database design: You should come up with a basic database schema which may hold the below parameters(and additionally also not limited to below). One single table or collection(if using MongoDB) should also be fine.
      i. ProductId
      ii. ProductName
      iii. Origin
      iv. PackingDetails
      v. Forecast
      vi. Colour
      vii. CultivationType
      viii. Moisture
      ix. FormAndCut
      x. Image
2. Profile Management Service
   a. This service is responsible for managing the profile information of the traders
   b. Ensure data consistency through transactions where necessary.
   c. The service should expose the below APIs
      i. GET /profile/{profileId}
         1. This returns the profile details
      ii. GET /profile/all
         1. This returns all the profiles in the datastore
      iii. POST /profile
         1. This creates the profile in the data store and return the profile id
      iv. PUT /profile/{profileId}
         1. This updates the profile in the datastore
   d. Database design: You should come up with a basic database schema which may hold the below parameters(and additionally also not limited to below). One single table or collection(if using MongoDB) should also be fine.
      i. ProfileId
      ii. BusinessName
      iii. BusinessOverview
      iv. BusinessType
      v. Established
      vi. Address
      vii. Logo
      viii. Owner

Guidelines

- You may use either Java(with springboot) or Python to design the backend. CRUD APIs should be there
- You may use any of MongoDB, Postgresql or MySQL to store the data
- You may use either React Native Web or NextJS along with React for frontend development

- Front end should be responsive
- **You may use the company name, logo, product of your choice for the assignment. The designs provided are just for reference**
- Deadline: Submit the assignment within 1 week.
- For the demo, we can directly navigate to the profile or catalog page without the home screen route, like http://localhost/profile/{profileId} should show the profile page directly
- Provide setup instructions for both frontend and backend.
- Share the code via GitHub/Bitbucket and grant access for review.
- Attach video/screenshots for reference
- Include a README explaining:
  - How to run the application
  - Commands(curl or database inserts) to insert/update data

Evaluation Criteria:

- Both frontend and backend work in liaison
- Correct API implementation to manage and store profile and catalog
- Functionality: Displaying the correct product page and profile page
- Code quality: Maintainable, modular, and in line with OOP concepts(for backend)
- Documentation: Clear setup and usage instructions.
- **Bonus Points:**
  - **Login/Signup Flows implementation**
  - Ensure code is modular and extendable.
  - Ensure code is explainable

Designs

Designs for product and profile pages can be found at - https://www.figma.com/design/r1AEKIpgYvE9baVRk13pAX/Full-Stack-Assignment?node-id=0-1&p=f&t=ITh13et35ZxfZTXS-0