TURKISH TECHNOLOGY

Digital Lab

Technical Assessment: Flight Menu Web Application

As part of Turkish Airlines' commitment to enhancing passenger experience, we aim to provide travelers with an easy-to-use digital solution for viewing and interacting with in-flight meal menus. This application will allow passengers to independently explore meal options, inquire about specific dietary needs in their preferred language, and receive personalized responses. By digitizing menu information and supporting multilingual interaction, this app will contribute to a more inclusive, convenient, and enjoyable journey for our passengers.

Objective:

Build a web application using a **full-stack Next.js** framework that allows users to:

1. Capture or upload images of meal menus from Turkish Airlines flights. You can use the example menus provided.
2. Extract menu information from the image using either OCR (Optical Character Recognition) or Large Language Models (LLMs) with vision capabilities.
3. Allow users to interact with the menu through a chat interface, where they can ask specific dietary questions (e.g., gluten-free options), in their preferred language.

Requirements:

- **Image Upload and Menu Extraction**:
  - Allow users to upload or capture an image of a meal menu.
  - Use either an OCR solution or an LLM with vision capabilities to extract text from the menu image.
- **Menu Processing and Display**:
  - Parse and structure the menu data to recognize individual items and their descriptions.
  - Display detected menu items in a card-based UI for easy viewing and selection.
- **Interactive Chat Interface**:
  - Build a chat interface where users can:
    - Ask questions about menu items, including specific dietary inquiries (e.g., "What are gluten-free options?").

        - Receive answers based on menu content.
    - Support multilingual queries: Users can ask questions in various languages, and the app should detect the language and respond accordingly.
- **Deployment**:
    - Deploy the application to a platform of your choice (e.g., Vercel, Heroku, AWS, Azure) and provide a testable URL.

Deliverables:

1. **Source Code**:
    - Provide a link to the GitHub repository containing the source code.
2. **Documentation**:
    - Explain the architecture, major components, and libraries used.
    - Include setup instructions and a guide on how to use the application.
3. **Sample URL**:
    - A deployed version of the app with a public URL for testing.

Evaluation Criteria:

- **Functionality**:
    - Accurate extraction and parsing of menu items.
    - Card-based UI effectively displaying menu items for user interaction.
    - Chat interface that answers user questions based on menu data.
- **Multilingual Support**:
    - Ability to process and respond in multiple languages.
- **Deployment**:
    - Successfully deployed application with a functional demo URL.
- **Code Quality and Documentation**:
    - Well-organized, readable code and detailed documentation.

Additional Notes:

If time allows, feel free to think beyond the instructions to enhance the user experience (UX). Ideas could include alternative approaches, additional filtering options, a more interactive UI, or other features that make the app more user-friendly and engaging.

Don’t reinvent the wheel; you are encouraged to use any AI platforms and sample repositories you can find.