Automated Data Extraction and Invoice Management

Develop a React application for Swipe that automates the extraction, processing, and management of invoices data from various file formats. The app should organize the extracted data into three main sections: Invoices, Products, and Customers, and synchronize changes in real-time using Redux for consistent updates across tabs.

Assignment Requirements:

1. File Uploads, AI-Powered Data Extraction, and Organization

- Accept different types of Input files including:
- Excel files: Contains transaction details (serial number, net/total amount, customer info).
- PDF/Images: Invoices with customer and item details, totals, tax, etc.
- Implement a generic AI-based extraction solution for all file types (Excel, PDF, images) to identify and organize relevant data into the appropriate tabs (Invoices, Products, Customers).
- Dataset link: assignment_test_cases

2. React App Structure with Tabs

- **Invoices Tab**: Table with columns Serial Number, Customer Name, product name, qty, tax, Total Amount and Date. (All required columns) - extra information is up to you
- **Products Tab**: Display a table with the following columns: Name, Quantity, Unit Price, Tax, Price with Tax (all required). The Discount column is optional but can be included for added detail.
- **Customers Tab**: Display a table with the following required columns: Customer Name, Phone Number, and Total Purchase Amount. Additional fields can be added at your discretion for more comprehensive customer data.

- Bonus points for additional data fields in any field.

3. Centralized State Management: Implement **React Redux** to maintain a centralized state for consistent data handling across the app.

- Ensure that any changes made in the **Products** or **Customers** tabs are automatically reflected in the **Invoices** tab.
- The application should dynamically update values across all tabs in real time. For example, if the **Product Name** is updated in the **Products** tab, the change should be instantly visible in the corresponding entry within the **Invoices** tab

4.Validation and Error Handling

- **Data Validation**: Validate all extracted data for completeness and accuracy.
- **User Feedback**: Provide clear feedback for unsupported file formats and extraction errors.
- **Handling Missing Fields**: Highlight missing fields and prompt users to complete them in a user-friendly way.

5. Code Quality and Documentation

- Maintain modular, well-documented code following React/Redux best practices.
- Include documentation on the AI data extraction feature.
- Mention the test cases solved with Screenshots/ videos.

AI Test Cases:

Test cases - > assignment_test_cases

- Case-1: **Invoice pdfs**
- Case-2: **Invoice pdf + Images.**
- Case-3: **Excel File**
- Case-4: **Excel Files**
- Case-5: **All Types of Files.**

The goal is to **extract Invoice, Product and Customer details and update in the respective tabs** for each of the above test cases using a **generic AI based solution.**

If any required information is missing from the file(which is true for at least one of the above cases), you must handle that case and the missing information should be highlighted in the app. You must try to successfully solve as many cases as possible.

Resource:

We recommend using Google Gemini in solving this assignment:

- Explore document processing capabilities with the Gemini API | Google AI for Developers
- Explore vision capabilities with the Gemini API | Google AI for Developers

Though you may implement any solution of your choice and solve the test cases.

Submission Instructions:

1. GitHub Repository: Push code to a public GitHub repo.

2. Deployment: Deploy the app on a free platform (e.g., Vercel, Netlify).

3. Submission: Complete this form: https://forms.gle/e8vGCsickX5nJJuUA

We look forward to seeing your approach to this real-world project!