YachtSicher GmbH

Executive Summary

I have developed an automated solution to streamline the processing of yacht insurance inquiries for YachtSicher GmbH. The solution leverages advanced technologies to extract information from unstructured PDF documents and presents it in an interactive, user-friendly web interface.

Solution Overview

The solution is a web application built with **Flask**, a lightweight Python web framework. It integrates **OpenAI's GPT-4** language model to extract data from PDF files containing yacht insurance inquiries. The extracted information is displayed in a dynamic and visually appealing table on the front-end, enhanced with **Bootstrap** for styling and **DataTables** for interactivity.

Key Components

1. **PDF Text Extraction**: Utilizes the pdfplumber library to extract text content from uploaded PDF files.
2. **Data Extraction with GPT-4**:
    - **Initial Extraction**: Sends the extracted text to GPT-4 with a carefully crafted prompt to extract specified fields such as Yacht Model, Owner's Name, and more.
    - **Handling Missing Data**: Identifies any missing fields and prompts GPT-4 to fill in these gaps by web search on reputable sources, ensuring data completeness.
3. **Web Application Backend**:
    - Built with **Flask**, handling file uploads, processing, and routing.
    - Incorporates secure practices by handling API keys via environment variables and managing uploaded files appropriately.
4. **Interactive Front-End**:
    - **Templates**: Uses HTML templates with **Jinja2** for dynamic content rendering.
    - **Styling and Responsiveness**: Implements **Bootstrap** to ensure the application is mobile-friendly and visually consistent.
    - **Data Presentation**: Employs **DataTables**, a plug-in for jQuery, to enhance the HTML table with features like sorting, searching, and pagination.
5. **Data Transformation**:

- Transposes the data so that each field is displayed as a row and each document's data as a column, improving readability and comparison across multiple inquiries.

Workflow

1. **User Interaction**:
   - Users access the web application and upload one or more PDF files containing yacht insurance inquiries.
2. **Backend Processing**:
   - The application saves the uploaded PDFs securely in a designated directory.
   - Extracts text from each PDF using pdfplumber.
   - Processes the extracted text with GPT-4 to obtain structured data.
   - Handles any missing information by re-invoking GPT-4 with a focused prompt.
   - Cleans up by removing the uploaded files after processing.
3. **Data Compilation**:
   - Collects all extracted data into a Pandas DataFrame.
   - Prepares the data for presentation in the front-end.
4. **Front-End Rendering**:
   - The Flask application renders the index.html template.
   - Injects the data table into the template, which is styled with Bootstrap classes.
   - Enhances the table with DataTables for improved user interaction, allowing features like horizontal scrolling to accommodate wide tables.
5. **User Experience**:
   - Users view the extracted data in a clear, organized table.
   - The interactive table allows users to navigate and review the information efficiently.