Documentation and User Guide

Project Overview

This Flask-based web application processes text data to generate multiple-choice questions (MCQs) using Google Generative AI. Users can upload text, PDF, or DOCX files or provide a URL to fetch text. The application generates MCQs with correct and plausible distractors, presents them in a readable format, and provides options to export the data in CSV, JSON, or TXT formats.

Features

- **Text Processing:** Extract text from various file formats (TXT, PDF, DOCX) or fetch text from a URL.
- **MCQ Generation:** Generate MCQs from the provided text using Google Generative AI.
- **Export Options:** Export generated MCQs in CSV, JSON, or TXT formats.
- **User Interface:** A simple web interface for file upload, text input, and export options.

Prerequisites

- **Python** (>=3.7)
- **Flask**
- **Pandas**
- **PyPDF2**
- **python-docx**
- **requests**
- **google-generativeai**
- **A Google API Key** for Google Generative AI

Installation

1. **Create a Virtual Environment:**

   ```
   python -m venv venv
   ```

2. **Activate the Virtual Environment:**
   - **Windows:**

     ```
     venv\Scripts\activate
     ```

   - **macOS/Linux:**

     ```
     source venv/bin/activate
     ```

3. **Install Required Packages:**

   ```
   bash
   Copy code
   pip install -r requirements.txt
   ```

4. **Set Environment Variables:** Ensure you have a `.env` file or set the environment variable `GOOGLE_API_KEY` with your Google Generative AI API key.

File Structure

- **app.py:** Main application script with Flask routes and logic.
- **templates/index.html:** Upload and text input form.
- **templates/mcqs.html:** Display generated MCQs in a table format.
- **requirements.txt:** List of required Python packages.

Usage

1. **Start the Flask Application:**

   ```
   bash
   Copy code
   python app.py
   ```

2. **Access the Application:** Open a web browser and navigate to http://127.0.0.1:5000.
3. **Upload a File or Provide a URL:**
   - **File Upload:** Select a TXT, PDF, or DOCX file.
   - **URL Input:** Enter a URL to fetch text from.
4. **Generate MCQs:**
   - Enter details for **Source**, **Topic**, and **Difficulty**.
   - Click the "Upload" button to process the file or URL and generate MCQs.
5. **View Generated MCQs:**
   - MCQs will be displayed in a table format on the results page.
6. **Export MCQs:**
   - Choose a format (CSV, JSON, TXT) and click the "Export" button to download the MCQs.

API Endpoints

- **/**: Displays the main page with upload and input options.
- **/upload**: Handles file uploads and URL processing, generates MCQs, and renders results.
- **/export**: Exports MCQs in the selected format.

Error Handling

- **Unsupported File Format:** If an unsupported file format is uploaded, the user will receive a "Unsupported file format" error.
- **No Text Data Provided:** If no text is extracted from the uploaded file or URL, a "No text data provided" error will be shown.
- **Invalid JSON Data:** If invalid JSON data is provided for export, an "Invalid JSON data" error will be displayed.

Example

Uploading a TXT file:

1. Select a .txt file containing text.
2. Enter "Book", "Literature", and "Easy" for Source, Topic, and Difficulty, respectively.
3. Click "Upload".
4. View the generated MCQs in the table format.
5. Export MCQs as CSV.

Providing a URL:

1. Enter a URL pointing to a text-based resource.
2. Enter "Website", "Technology", and "Medium" for Source, Topic, and Difficulty, respectively.
3. Click "Upload".
4. View and export the generated MCQs.

Troubleshooting

- **API Key Issues:** Ensure the GOOGLE_API_KEY environment variable is set correctly.
- **File Parsing Errors:** Verify that the file formats are supported and not corrupted.
- **Note:** Please use a short file to upload otherwise your google api limit will be exhausted.