Project Document: MCQ Generator from PDF

1. Introduction

- **Project Overview**: The MCQ Generator from PDF is a web application designed to extract multiple-choice questions (MCQs) from uploaded PDF files and generate them into a structured CSV format.

- **Objectives**:

  - Provide a user-friendly interface for uploading PDF files.

  - Automatically extract MCQs from the PDF content.

  - Generate MCQs in a structured format suitable for educational purposes.

- **Audience**: Teachers, educators, and educational institutions seeking to automate the process of creating quizzes and tests.

2. Technologies Used

- **Programming Languages**: Python

- **Frameworks and Libraries**: Streamlit, Pandas, PyMuPDF, Langchain, Python-dotenv

- **APIs**: OpenAI API (for text generation)

3. Directory Structure

- **File Structure**:

  - `app.py`: Main Streamlit application script handling user interface and interaction.

  - `mcq_extractor.py`: Module for PDF text extraction and MCQ generation.

  - `requirements.txt`: List of project dependencies.

4. Installation Instructions

- **Prerequisites**: Python 3.x, pip package manager

- **Setup Instructions**:

  1. Clone the repository: `git clone https://github.com/your-repo/mcq_generator.git`

  2. Navigate to the project directory: `cd mcq_generator`

  3. Install dependencies: `pip install -r requirements.txt`

  4. Create a `.env` file and add your OpenAI API key.

5. Usage

- **Running the Application**: Execute `streamlit run app.py` in the terminal.

- **Functionality**:

  - Upload a PDF file containing educational content.

  - Specify the number of MCQs, subject, and tone for quiz generation.

  - View generated MCQs in a table format.

  - Download the generated MCQs as a CSV file.

6. Detailed Components

- **app.py**: Handles the Streamlit user interface and integrates with `mcq_extractor.py` for MCQ generation.

- **mcq_extractor.py**: Utilizes PyMuPDF for PDF text extraction and Langchain/OpenAI for MCQ generation based on extracted content.

7. Code Snippets

- **Example Code Snippet**: (Insert relevant code snippets from `app.py` and `mcq_extractor.py`)

8. Project Dependencies

- **List of Dependencies**: Refer to `requirements.txt` for a complete list of Python packages and versions.

9. Testing

- **Testing Strategy**: Manual testing performed for user interface interactions and automated unit testing for critical functions.

- **Tools Used**: Built-in testing frameworks for Python.

10. Limitations and Future Enhancements

- **Current Limitations**: PDFs with complex layouts may not extract MCQs accurately.

- **Future Enhancements**: Implement machine learning models for improved MCQ extraction and natural language understanding.

11. Contributors

- **Project Team**: Ganesh Jagadeesan (Developer)

12. Conclusion

- **Summary**: The MCQ Generator from PDF simplifies the process of creating educational quizzes and assessments from PDF documents, enhancing efficiency for educators and learners alike.

- **Acknowledgments**: Special thanks to OpenAI for their API support and the Streamlit community for their user-friendly framework.

Appendices

- **Additional Resources**: Links to project repository https://github.com/Ganlak/mcqgenerator.git.