Project Proposal

Submitted by : Zainab Bibi

Internship ID : TN/IN01/PY/018

Project Title :

PDF to Audio & MCQ Generator using Python

1. Project Overview

The proposed project is a Python-based desktop application that allows users to convert the textual content of PDF files into audio format and automatically generate multiple-choice questions (MCQs) from the same content. This dual-function tool enhances accessibility for auditory learners and also serves as an intelligent content analysis tool for educational purposes.

2. Problem Statement

In the digital learning environment, students and professionals often deal with lengthy PDF documents that can be hard to read in one sitting. Additionally, manually generating practice questions from academic documents is time-consuming. There is a need for an application that can both convert reading material to speech and extract potential questions for self-assessment.

3. Objectives

- To develop a user-friendly desktop application with a graphical interface.
- To enable users to convert PDF content to audio files using text-to-speech technology.
- To automate the generation of MCQs from the PDF content.
- To provide a clean, modern design and a responsive GUI using Tkinter.

4. Functional Requirements

- Users should be able to upload any .pdf file.
- The system should extract readable text from the PDF.
- The extracted text should be converted into audio and saved as an .mp3 file.
- The application should generate 3–5 MCQs automatically from the text.
- The user should be able to view MCQs within the application.

5. Non-Functional Requirements

- The GUI should be responsive and professionally designed.
- The application should handle large files by splitting text into manageable chunks.
- The application must validate input and provide appropriate feedback (e.g., file not selected, empty PDF).
- The text-to-speech service should require an active internet connection.

6. Tools and Technologies

- **Language:** Python 3.x

- **Libraries/Modules:**
    - Tkinter (GUI)
    - PyPDF2 (PDF text extraction)
    - gTTS (Google Text-to-Speech)
    - OS and Random (file handling and logic)
- **Platform:** Windows/Linux Desktop Environment
- **Optional Tools:** PyInstaller for creating standalone executable

7. System Architecture

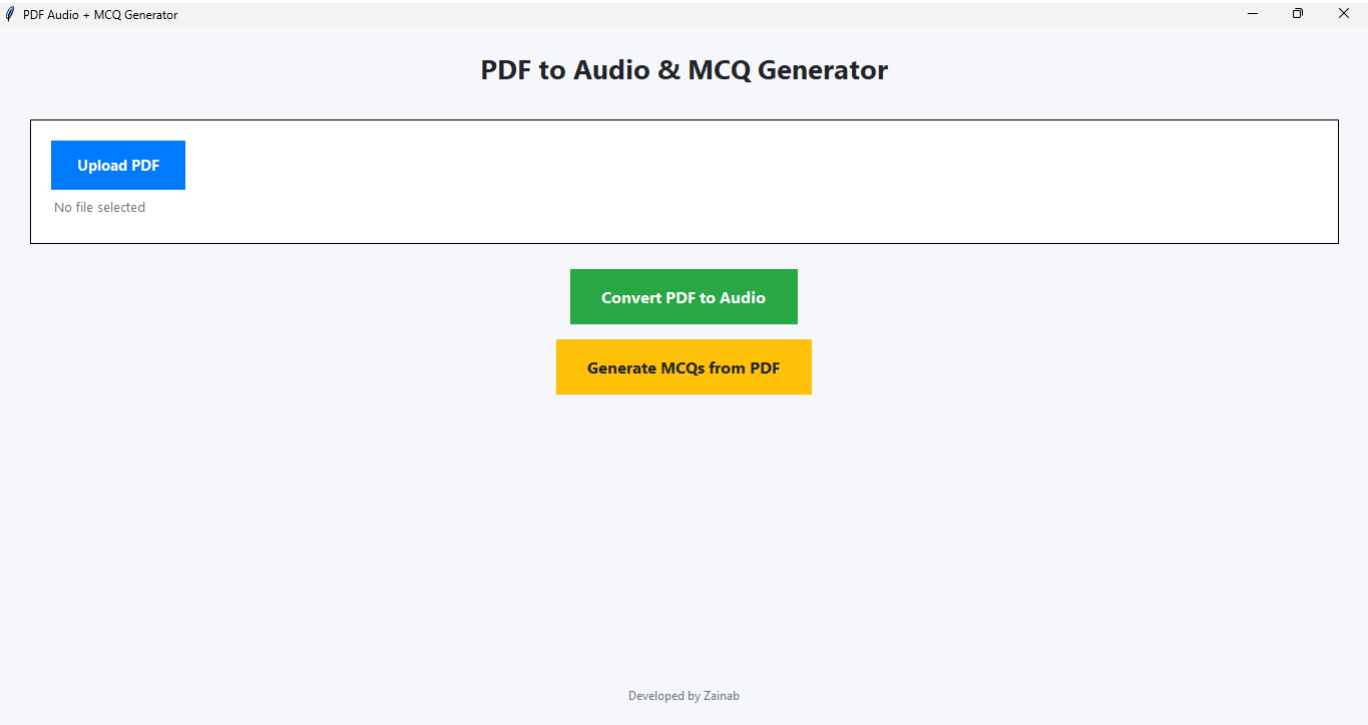
1. **Input Layer:**
    - PDF File is uploaded by the user.
2. **Processing Layer:**
    - Text is extracted using PyPDF2.
    - Text is sent to gTTS for audio conversion.
    - Text is analyzed and split into sentences to generate MCQs.
3. **Output Layer:**
    - MP3 file is saved.
    - MCQs are displayed in a new GUI window.

8. Use Cases

- Visually impaired users who prefer listening over reading.
- Students preparing for exams and wanting quick revision questions.
- Teachers looking to automate quiz generation from notes.
- Professionals who wish to listen to reports during commutes.

9. Expected Outcomes

- A working desktop application with dual functionality: text-to-speech and MCQ generation.
- A polished user interface that meets usability standards.
- Error-handling and support for different file sizes.



Conclusion

This project aims to provide an effective and practical tool for digital content conversion and evaluation. By combining PDF-to-audio functionality with automatic MCQ generation, this application will offer significant value in both academic and professional settings. It bridges the gap between accessibility and intelligent content assessment using simple yet powerful Python tools.