PDF Summarizer App

Overview

The PDF Summarizer App allows users to upload a PDF and get a summary generated using a transformer-based model from Hugging Face.The frontend is built using **Streamlit**.

Tech Stack

- **Frontend**: Streamlit
- **NLP Model**: Hugging Face Transformers
- **Text Extraction**: `PyMuPDF (fitz)` or `PyPDF2`
- **Model**: `facebook/bart-large-cnn`, `t5-base`, etc.

To Run the Application

How It Works

This tool helps users upload a PDF and get a short summary of its content. When a user uploads a PDF file through the Streamlit interface, the app reads and extracts all the text from the file using a library called PyMuPDF. If the text is too long, it breaks it into smaller parts so that the summarization model can handle it easily.

Each part is then sent to a pre-trained AI model (like BART or T5 from Hugging Face) that reads the text and creates a short summary. After all parts are summarized, the app combines them into one final summary and shows it to the user. This way, users can quickly understand long PDFs without reading the whole thing.

Issues Faced & How They Were Solved

Issue	Solution
`RuntimeError: At least one of TensorFlow 2.0 or PyTorch should be installed.`	Installed PyTorch via `pip install torch`.
`ValueError: Your currently installed version of Keras is Keras 3, but this is not yet supported in Transformers.`	Used PyTorch-only models (e.g., `facebook/bart-large-cnn`) instead of TensorFlow-based ones.
`ModuleNotFoundError: No module named 'fitz'`	Installed PyMuPDF with `pip install PyMuPDF`.
Large PDFs caused `Token limit exceeded` errors	Chunked text into smaller segments (~1000 tokens) before summarizing.
Model inference was slow	Used lighter models like `t5-small` or summarized in chunks.

✅ Key Learnings

- Learned how to integrate **Hugging Face transformers** with custom inputs (PDF text).
- Understood trade-offs between **model size vs. speed vs. quality**.
- Successfully built a **clean UI in Streamlit**