**AI Document Analysis System Documentation**

**Introduction**

The AI Document Analysis System is designed to extract, analyze, and interpret textual content from PDFs using state-of-the-art natural language processing (NLP) models. This project provides users with the ability to upload PDF documents and query them for relevant information. The system leverages SQLite for user authentication and document management, ensuring secure and efficient data storage.

**Key Features:**

* **User Authentication:** Secure sign-up and login functionality with SQLite.
* **PDF Processing:** Extracts text from uploaded PDF documents.
* **AI-Powered Query System:** Uses transformer-based models to answer user queries based on document content.
* **Admin Dashboard:** Allows administrators to view user records and uploaded PDFs.

**Installation & Setup**

**Prerequisites:**

Ensure you have the following installed:

* Python 3.8+
* pip (Python package manager)
* SQLite3
* Required dependencies:

pip install gradio transformers sqlite3 PyPDF2 python-dotenv

**Environment Setup:**

1. Clone the repository:
2. git clone
3. cd
4. Set up environment variables in a .env file:
5. Run the application:
6. python <appname>.py

**Architecture Overview**

**System Components:**

1. **Frontend (Gradio UI):** User-friendly web interface.
2. **Backend (Flask/Python):** Handles authentication, document processing, and AI queries.
3. **Database (SQLite3):** Stores user credentials and document records.
4. **AI Engine (Transformers):** Utilizes NLP models for document question-answering.

**Data Flow:**

1. User logs in or signs up.
2. Uploads a PDF file.
3. AI extracts and processes text.
4. User submits a query.
5. AI model generates a response.
6. Response is displayed in the UI.

**Usage Guide**

**User Authentication:**

1. **Sign Up:** Enter email and password.
2. **Login:** Access the chatbot after authentication.

**Uploading a PDF:**

* Click on the "Upload PDF" button.
* Select a PDF file.
* Wait for text extraction to complete.

**Querying the Document:**

* Enter a question related to the uploaded document.
* Click "Submit" to receive an AI-generated answer.

**Model Details**

**AI Model Used:**

* **Transformer Model:** distilbert-base-cased-distilled-squad
* **Functionality:** Performs question-answering tasks based on extracted text.
* **Limitations:** Works best on structured documents with clear question-answer pairs.

**Examples & Tutorials**

**Example Query:**

from transformers import pipeline

qa\_model = pipeline("question-answering", model="distilbert-base-cased-distilled-squad")

context = "The AI Document Analysis System is designed to extract, analyze, and interpret textual content from PDFs."

question = "What is the purpose of the system?"

response = qa\_model(question=question, context=context)

print(response["answer"])

**Troubleshooting & FAQs**

**Common Issues:**

1. **Database Not Found:** Ensure users.db exists and is accessible.
2. **PDF Extraction Fails:** Ensure the PDF is not encrypted or damaged.
3. **Incorrect Model Predictions:** Use clearer, more specific queries.

**Contributing Guidelines**

**How to Contribute:**

* Fork the repository and create a new branch.
* Implement your changes and submit a pull request.
* Follow coding standards and include relevant documentation.

**License & References**

* **License:** ISC License
* **References:**
  + [Transformers Library](https://huggingface.co/transformers/)
  + [Gradio Documentation](https://gradio.app/docs/)
  + [SQLite Documentation](https://sqlite.org/docs.html)