AI Video Summarizer App - Setup Instructions

Prerequisites

This application requires Python 3.8 – 3.11 (Python 3.9+ is recommended for optimal performance).

Installation Guide

1. Install Required Dependencies

Create a `requirements.txt` file in your project directory with the following content:

```
Flask
openai
moviepy
textblob
wordcloud
matplotlib
sentence-transformers
faiss-cpu
numpy
pillow
requests
```

Install all dependencies using:

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

Windows Users: If you encounter issues with `faiss`, try one of these alternatives:

```bash
pip install faiss-cpu --extra-index-url https://pypi.ngc.nvidia.com
```

Or use a specific version:

```bash
pip install faiss-cpu==1.7.4
```

2. FFmpeg Installation

FFmpeg is required for audio file conversion and processing.

Windows Installation:

1. Download FFmpeg from: https://www.gyan.dev/ffmpeg/builds/ffmpeg-release-essentials.zip
2. Extract the downloaded zip file
3. Add the FFmpeg binary path (e.g., `C:\ffmpeg\bin`) to your system PATH environment variable
4. Alternatively, use absolute paths in your code:

```python
os.system(r'"C:\ffmpeg\bin\ffmpeg.exe" -i temp_audio.wav -ac 1 -ar 16000 -y compressed_audio.wa
```

macOS Installation:

```bash
brew install ffmpeg
```

Linux Installation:

```bash
sudo apt update
sudo apt install ffmpeg
```

3. Database Initialization

The application uses SQLite for data storage. The database (`video_analysis.db`) and required tables will be created automatically on first run when `init_db()` is called.

The database stores:

- Video transcripts
- Generated summaries
- Sentiment analysis and emotion detection results
- Extracted keywords
- Embedding vectors for semantic search
- Timestamps and metadata

4. OpenAI API Configuration

Method 1: Direct API Key Replace the placeholder in your `app.py` file:

```python
client = openai.OpenAI(api_key="YOUR_ACTUAL_API_KEY_HERE")
```

Method 2: Environment Variable (Recommended) Use environment variables for better security:

```python
import os
client = openai.OpenAI(api_key=os.getenv("OPENAI_API_KEY"))
```

Set your API key in the environment:

Linux/macOS:

```bash
export OPENAI_API_KEY=sk-your-api-key-here
```

Windows:

```bash
set OPENAI_API_KEY=sk-your-api-key-here
```

5. Directory Structure Setup

Create the required directories for file storage:

```bash
mkdir uploads
mkdir static
```

These directories serve the following purposes:

- `uploads/`: Stores uploaded video files
- `static/`: Contains saved audio files and generated word cloud images

6. Running the Application

Start the Flask development server:

```bash
python app.py
```

The application will be available at: http://127.0.0.1:5000

7. Voice Q&A Feature

The application includes voice-based question answering functionality using JavaScript MediaRecorder API. This feature:

- Captures microphone input through the browser
- Sends audio data to the `/ask-audio` endpoint
- Requires modern browser support (Chrome/Edge recommended)

Important Requirements:

- Enable microphone permissions in your browser
- Use HTTPS when deploying to production (required for microphone access)
- Ensure your browser supports the MediaRecorder API

Troubleshooting

Common Issues

FFmpeg Not Found:

- Verify FFmpeg is properly installed and added to PATH
- Test by running `ffmpeg -version` in your terminal

Database Errors:

- Ensure write permissions in the project directory
- Check if `video_analysis.db` is created successfully

API Rate Limits:

- Monitor your OpenAI API usage
- Implement appropriate error handling for rate limit responses

Browser Compatibility:

- Use Chrome or Edge for best voice feature support
- Enable microphone permissions when prompted

Security Considerations

- Never commit API keys to version control
- Use environment variables for sensitive configuration
- Implement proper input validation for file uploads
- Consider implementing user authentication for production use

Support

For additional support or troubleshooting, refer to the documentation of individual components:

- Flask Documentation
- OpenAI API Documentation
- FFmpeg Documentation