Project Documentation

Social Content Engine

The **Social Content Engine** is a modular backend system designed to automate the creation of content tailored for various social media platforms. Initially focused on TikTok, the engine is designed with flexibility in mind, allowing it to be extended to other platforms in the future.

Overview

The Social Content Engine provides a range of functionalities, including:

- **Document Processing**: Extracts text from uploaded documents (e.g., PDFs).
- **Content Generation**: Uses AI models, such as OpenAI's GPT, to generate engaging content based on extracted text.
- **Audio Conversion**: Converts generated text to speech.
- **Video Creation**: Overlays generated content onto video clips and synchronizes captions with audio.
- **Call-to-Action Integration**: Adds a custom call-to-action at the end of each video.
- **API Interface**: A Flask-based API that interacts with the social content engine.
- **Streamlit Frontend**: A simple web interface for uploading documents, managing content, and generating videos.

Folder Structure

The project is organized into the following directories and files:

```
tiktokce/
├── frontend/
│   └── app.py
├── LICENSE
├── requirements.txt
├── tests/
│   ├── test_content_generation.py
│   ├── test_audio_conversion.py
│   ├── test_document_processing.py
│   ├── test_video_creation.py
│   ├── __init__.py
│   └── test_api.py
├── MANIFEST.in
├── README.md
├── setup.py
├── .gitignore
├── api/
```

```
│   ├── __init__.py
│   ├── main.py
│   └── routes.py
├── data/
│   ├── video_library/
│   └── sample_documents/
├── src/
│   ├── config/
│   │   └── __init__.py
│   ├── social_content_engine/
│   │   ├── cta_integration/
│   │   │   ├── __init__.py
│   │   │   └── add_cta.py
│   │   ├── content_generation/
│   │   │   ├── __init__.py
│   │   │   └── ai_content_generator.py
│   │   ├── audio_conversion/
│   │   │   ├── __init__.py
│   │   │   └── text_to_speech.py
│   │   ├── video_creation/
│   │   │   ├── video_library/
│   │   │   │   ├── __init__.py
│   │   │   │   └── manage_videos.py
│   │   │   ├── video_overlay.py
│   │   │   ├── __init__.py
│   │   ├── document_processing/
│   │   │   ├── __init__.py
│   │   │   └── pdf_processor.py
│   │   └── main.py
│   └── utils/
│       ├── logger.py
│       └── decorators.py
```

Directory and File Descriptions

- **frontend/**: Contains the Streamlit-based frontend application.
  - `app.py`: The main file for the frontend application.
- **LICENSE**: The license under which the project is distributed.
- **requirements.txt**: Lists the Python packages required for the project.
- **tests/**: Contains test scripts for various modules in the project.
  - `test_content_generation.py`: Tests for content generation.
  - `test_audio_conversion.py`: Tests for audio conversion.
  - `test_document_processing.py`: Tests for document processing.
  - `test_video_creation.py`: Tests for video creation.
  - `__init__.py`: Initialization file for the tests package.
  - `test_api.py`: Tests for the API functionality.
- **MANIFEST.in**: Specifies files to include in the source distribution.
- **README.md**: This documentation file, providing an overview of the project.
- **setup.py**: Setup script for installing the package.
- **.gitignore**: Specifies files and directories to be ignored by Git.
- **api/**: Contains the API files for interacting with the backend engine.
  - `__init__.py`: Initialization file for the API package.
  - `main.py`: The main entry point for the API.
  - `routes.py`: Defines the API routes and endpoints.
- **data/**: Stores video libraries and sample documents for testing.

    - `video_library/`: Directory for storing video assets.
    - `sample_documents/`: Directory for storing sample documents.
- **src/**: The core source code for the Social Content Engine.
    - `config/`: Configuration settings for the project.
        - `__init__.py`: Initialization file for the config package.
    - `social_content_engine/`: Contains all the engine's processing modules.
        - `cta_integration/`: Module for adding call-to-actions.
            - `__init__.py`: Initialization file for the cta_integration package.
            - `add_cta.py`: Handles adding call-to-actions to videos.
        - `content_generation/`: Module for generating content.
            - `__init__.py`: Initialization file for the content_generation package.
            - `ai_content_generator.py`: Uses AI to generate content.
        - `audio_conversion/`: Module for converting text to speech.
            - `__init__.py`: Initialization file for the audio_conversion package.
            - `text_to_speech.py`: Converts text into audio files.
        - `video_creation/`: Module for creating videos from content.
            - `video_library/`: Manages video assets.
                - `__init__.py`: Initialization file for the video_library package.
                - `manage_videos.py`: Handles video library management.
            - `video_overlay.py`: Overlays content and audio onto videos.
            - `__init__.py`: Initialization file for the video_creation package.
        - `document_processing/`: Module for processing documents.
            - `__init__.py`: Initialization file for the document_processing package.
            - `pdf_processor.py`: Extracts text from PDF documents.
        - `main.py`: The main interface for the Social Content Engine.
    - `utils/`: Utility functions and helpers.
        - `logger.py`: Configures and manages logging for the project.
        - `decorators.py`: Provides reusable decorators for the project.

Installation and Setup

1. Install Dependencies

Ensure you have Python installed. Then, install the required dependencies:

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

2. Run the API

Start the API by running:

```
python api/main.py
```

3. Run the Streamlit Frontend

To launch the frontend interface, run:

```
streamlit run frontend/app.py
```

4. Access Swagger Documentation

Swagger UI for the API is available at http://localhost:5000/apidocs.

Contribution

Feel free to fork this repository, make updates, and submit pull requests. All contributions are welcome!

License

This project is licensed under the terms specified in the LICENSE file.