Deployment Guide

This guide provides step-by-step instructions to deploy the AI Podcast web app on your local PC, after deployment you should be able to access the web app via the browser.

If you have already deployed this before, please refer to last section for simplified instructions to start the webapp.

Prerequisites

```
Please make sure you have the following environment installed:

- Node.js
- Python 3.11
- Azure functions core tools
```

First steps

```
- Open your terminal, and use git clone to download the repository.

- Download the Required_Files.zip and unzip it, we will need it later in this guide.
- https://aipodcaststorage.blob.core.windows.net/deploy-required-files/Required_Files.zip

- Create a .env file under path "System-Engineering-Automated-Podcast\functionApp\EpisodesGe
- Fill in the following variables with your own API keys

    STABILITY_API_KEY : StabilityAI API (used for image generation)

    ibm_api : IBM Watson API key (used for text to speech conversion)

    ibm_url : IBM Watson url (used for text to speech conversion)

    elevenlabs_api : ElevenLabs API (used for text to speech conversion)

    AZURE_STORAGE_CONNECTION_STRING : The connection string from Azure Storage Account
    AZURE_STORAGE_CONTAINER_NAME : The container name of the blob storage

    COSMOS_CONNECTION_STRING : The connection string from Azure Cosmos DB
    Example : "mongodb://<COSMOSDB_USER>:<COSMOSDB_PASSWORD>@<COSMOSDB_USER>.mongo.cosmos.az

    COSMOSDB_USER : USERNAME from Azure Cosmos DB
    COSMOSDB_PASSWORD : PASSWORD from Azure Cosmos DB
    COSMOSDB_DBNAME : Database name from Azure Cosmos DB
    COSMOSDB_HOST : HOST from Azure Cosmos DB
    COSMOSDB_PORT : PORT from Azure Cosmos DB
    COSMOSDB_ARGS : "ssl=true&replicaSet=globaldb&retrywrites=false&maxIdleTimeMS=120000&app
    COSMOSDB_COLLECTION = The collection name of the target mongoDB

- Create a .env file under path "System-Engineering-Automated-Podcast\server"
- Fill in the following variables with your own API keys

    COSMOSDB_USER : USERNAME from Azure Cosmos DB
    COSMOSDB_PASSWORD : PASSWORD from Azure Cosmos DB
    COSMOSDB_DBNAME : Database name from Azure Cosmos DB
    COSMOSDB_HOST : HOST from Azure Cosmos DB
    COSMOSDB_PORT : PORT from Azure Cosmos DB
    COSMOSDB_ARGS : "ssl=true&replicaSet=globaldb&retrywrites=false&maxIdleTimeMS=120000&app

- Download the local language model file from the following link:
- https://gpt4all.io/models/gguf/mistral-7b-openorca.gguf2.Q4_0.gguf

- Put the downloaded .gguf file in folder "System-Engineering-Automated-Podcast\functionApp\
```

Build the client files

```
- Run command in terminal:

    - "cd System-Engineering-Automated-Podcast\client"
    - "npm install"
    - "npm run build"

- Copy all the files from "System-Engineering-Automated-Podcast\client\build" to "System-Eng
```

Start the server

```
- Run command in terminal:

    - "cd .."
    - "cd System-Engineering-Automated-Podcast\server"
    - "npm install"
    - "npm start"
```

The website can be accessed from browser via URL “localhost:8080”, you can view and play the episodes from database now.

Start the function app

The server is now up and running, however in order to run the episode generation tasks, you’ll need the function app.

Create and activate the virtual environment

```
- For Windows:
    - Open a new PowerShell window

    - Run the following commands:

        - "cd System-Engineering-Automated-Podcast"
        - "python -m venv functionApp"
        - "cd functionApp"
        - ".\Scripts\Activate.ps1"

- For MacOS/Linux

    - Open a new terminal

    - Run the following commands:

        - "cd System-Engineering-Automated-Podcast"
        - "python -m venv functionApp"
        - "cd functionApp"
        - "source bin/activate"
```

Install requirements

```
- Run command in terminal:
    - "pip install azure-functions requests"
```

Create a local function app

```
- Run command in terminal:
    - "func init EpisodesGen --worker-runtime python -m v1 "
```

Create a new function

```
- Run command in terminal:

    - "cd EpisodesGen"
    - "func new --name GenerateEpisode --language python --template "HTTP trigger" --authlev

- When prompted whether to overwrite, choose "No"
```

Copy files and install the requirements

```
- Copy "ffmpeg.exe" and "ffprobe.exe" from Required_Files to "System-Engineering-Automated-P

- Create an empty folder called "tmp" under "EpisodesGen" folder

- Run command in terminal:
    - "pip install -r requirements.txt"
```

Start the function App

```
- Run command in terminal:

    - "func start"
```

Now the function app is also up and running, you can go back to the browser to generate new episodes.

If you have deployed before and want to start the webapp again

Here is the simplified commands to quickly start the webapp:

```
- In the terminal, change directory to "\server"

- Run "npm start"

- In the terminal, change directory to "\functionApp\EpisodesGen"

- Run "func start"
```

Now the web app is available to access!