img_module/

This module contains the code and materials related to image generation.
It includes a standalone script for testing image generation (demo.py), a GUI tool for interacting with the image generation backend (generate_img_GUI.py), and a FastAPI-based backend service for generating images (img_generate_backend.py).
These components demonstrate the exploration and implementation of the image generation functionality in this project.

- **comparison_results/**
  Output directory for images generated by generate_img_compare.py.
  It saves the results of comparing Stable Diffusion, Janus, and DALL-E models.

- **shoes_images/**
  Output directory for images generated by demo.py, storing example outputs related to generating shoe-themed images using Stable Diffusion.

- **demo.py**
  A standalone Python script that loads a Stable Diffusion model through Hugging Face's diffusers library, automatically generates images based on preset prompts (related to shoes), and saves the output into shoes_images/.

- **generate_img_compare.py**
  A script for comparing different models (Stable Diffusion, Janus, DALL-E).
  It uses fixed prompts to generate images through each model, places the generated images side-by-side for visual comparison, and saves the result to comparison_results/.

- **generate_img_GUI.py**
  A simple Tkinter GUI application.
  It connects to the image generation backend (img_generate_backend.py) through HTTP requests, allowing users to input prompts and view generated images within a pop-up window.

- **img_generate_backend.py**
  A FastAPI backend service for image generation.
  It loads the Janus Pro 7B model at startup, listens for requests at the /generate-image/ endpoint, accepts prompts, generates images, saves them locally, and returns them as Base64-encoded images to the frontend.

mind_module/

This module contains code and materials related to mind map generation and display.

- **tmp_dir/**
  Temporary directory for storing intermediate files during mind map generation, including Markdown text files and rendered HTML files.

- **generate_mind.py**
  Defines the MindmapGenerator class.
  It receives structured text, saves it as Markdown, converts it into HTML using markmap-cli, and then uses Playwright to render the HTML as a PNG image.

- **show_png_frame.py**
  Defines the ZoomPanImageViewer class.
  It is a custom Tkinter component based on CTkFrame, supporting panning and zooming of PNG images using mouse dragging and scroll wheel.

test_script/

This directory contains independent testing scripts for different core system functionalities. These scripts are designed to verify whether different modules of the system — such as text generation, image generation, and TTS — are functioning correctly. They form an important part of the project's development and debugging process.

- **test_generate_image.py**
  A testing script for the image generation module.
  It can either call the backend API or directly load models to test generating images from text prompts.

- **test_generate_text.py**
  A testing script for the text generation module.
  It sends prompts to the text generation backend and checks the returned response.

- **test_output_image.png**
  A sample image generated during testing, mainly used for manual inspection of image quality.

- **test_output_tts.wav**
  A sample audio file generated during TTS testing, mainly used for manual evaluation of speech synthesis quality.

- **test_tts.py**
  A testing script for the TTS module.
  It sends text input, synthesizes speech, and plays or saves the resulting audio for evaluation.

Other Main Files and Directories

- **tmp_dir/** (at the root directory)
  General-purpose temporary directory, used during platform runtime to save generated temporary files (such as recorded audio WAV files for speech recognition, or cached online search results).

- **tmp_img/**
  Temporary directory used to save images returned by the /generate-image/ API. Images are stored locally and then loaded into the chat interface for display.

- **backend.py**
  The main backend service script for the entire platform, implemented with FastAPI. It exposes three core service APIs:

  - /generate-text/ : Calls the Qwen model for structured text generation.
  - /generate-image/ : Calls the Janus model for image generation.
  - /tts/ : Calls the FastPitch TTS model for text-to-speech synthesis.

All models are loaded locally to ensure privacy and low-latency inference.

- **main_GUI.py**
  The main frontend script of the platform, implemented using CustomTkinter.
  It includes the following main functionalities:

  - Captures voice input, performs speech recognition using Wav2Vec2, and converts speech to text.
  - Accepts text input for AI interactions.
  - Parses and extracts mind map instructions from AI responses, uses MindmapGenerator to generate mind maps, and displays them with ZoomPanImageViewer.
  - Detects and extracts image prompts from AI responses, calls the backend /generate-image/ API asynchronously, and displays generated images.
  - Sends AI responses sentence-by-sentence to the /tts/ API for text-to-speech, immediately playing back synthesized audio.
  - Integrates Google Search to enhance user queries when enabled.
  - Implements a multithreaded design to keep the GUI responsive, avoiding interface freezing during backend inference.