**Audio2Art: Transforming Voice Prompts Into Visual Creations Using Transformers**

**DONE BY:  
SANSKRITI SAJLAL  
MULLAI AGHALYA  
HARIDHRA LAXMI KUMAR**

## ***Problem Statement***

In today’s increasingly AI-driven world, the boundary between creativity and technology is rapidly dissolving. Artists, educators, and therapists alike are seeking intuitive tools that can bridge the gap between what we say and what we see. Audio2Art answers this need by empowering users to transform voice prompts into vivid, AI-generated images in seconds. By combining the latest advances in speech recognition and image generation, this project demonstrates the immense potential of multimodal artificial intelligence.

## ***Requirements***

### **Functional Requirements**

1. Users can upload .wav audio files via a web-based interface.
2. The system must transcribe the audio into a text prompt using a pre-trained Wav2Vec2 model.
3. The transcribed text must be passed into a Stable Diffusion model to generate a corresponding image.
4. The generated image and text prompt are displayed in the UI.
5. Users must have the ability to download the generated image.

### **Non-Functional Requirements**

1. Transcription and image generation should be completed within 30 seconds.
2. The system should utilize available GPU resources for faster execution.
3. The interface must be simple, responsive, and mobile-friendly.
4. The system should be easily deployable from a Google Colab environment.
5. Public access should be available via LocalTunnel for external testing.

## ***User Stories***

**​​Artist**

As an artist, I want to instantly turn spontaneous thoughts and emotions into visual art by simply speaking them aloud,  
so that I can capture fleeting inspiration and spark new creative directions with minimal effort.

### **High School Teacher**

As a high school teacher, I want my students to narrate historical events and see them visualized in real time,  
so that learning becomes more immersive, engaging, and memorable through visual storytelling.

### **Therapist**

As a therapist, I want my clients to express their emotions verbally and see them transformed into art,  
so that they can explore their feelings in a non-verbal, visual way—supporting deeper emotional understanding and healing.

### **Developer**

As a developer, I want to build and deploy a robust audio-to-image application powered by multimodal AI,  
so that I can showcase cutting-edge technology and inspire creative real-world applications of AI.

### **Student (University)**

As a university student, I want to convert my spoken research ideas into visual concepts using Audio2Art,  
so that I can better understand abstract topics and present my ideas more effectively in academic settings.

## 

## 

## 

## 

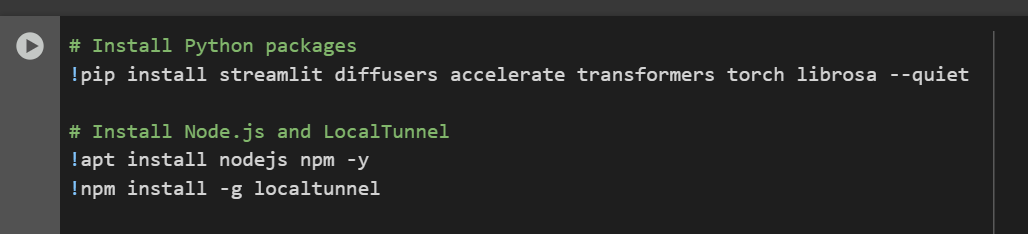
## ***Project Planning and Scheduling***

| **Phase** | **Task** | **Timeline** |
| --- | --- | --- |
| Phase 1: Setup | Install packages and configure the Colab environment | Day 1 |
| Phase 2: Model Building | Create ImageModel.py to transcribe and generate images | Day 2 |
| Phase 3: UI Development | Build Streamlit UI in app.py | Day 3 |
| Phase 4: Integration | Integrate UI with model and logic flow | Day 4 |
| Phase 5: Deployment | Set up LocalTunnel for external testing | Day 5 |
| Phase 6: Testing | Conduct end-to-end testing and UI refinements | Day 6 |

## 

## *Code*

### **Dependency Installation**

  
Part 1:

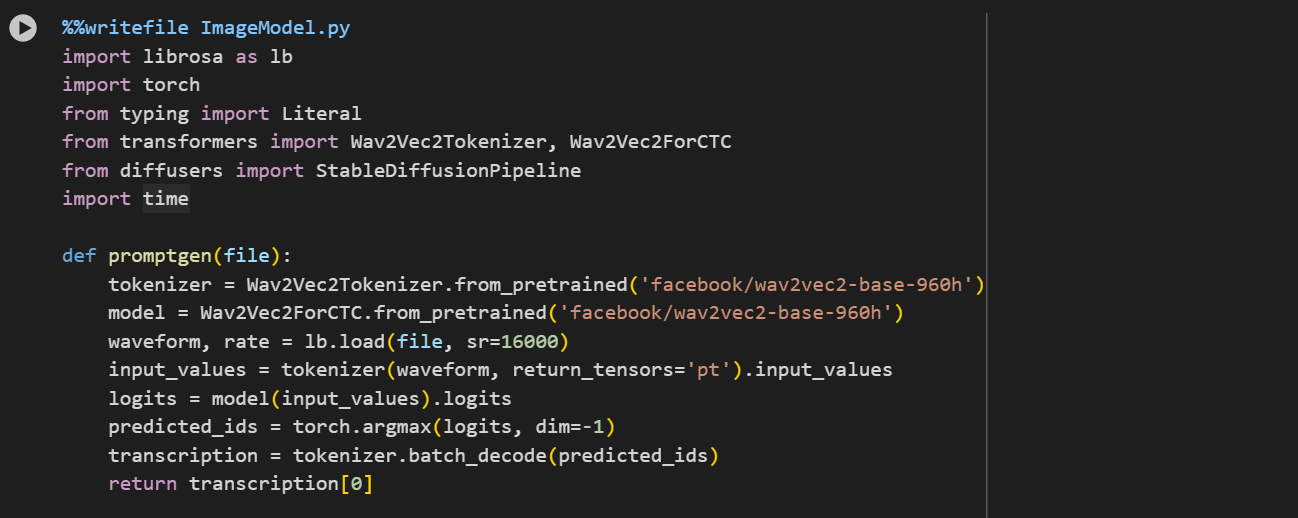
Installs Python libraries:

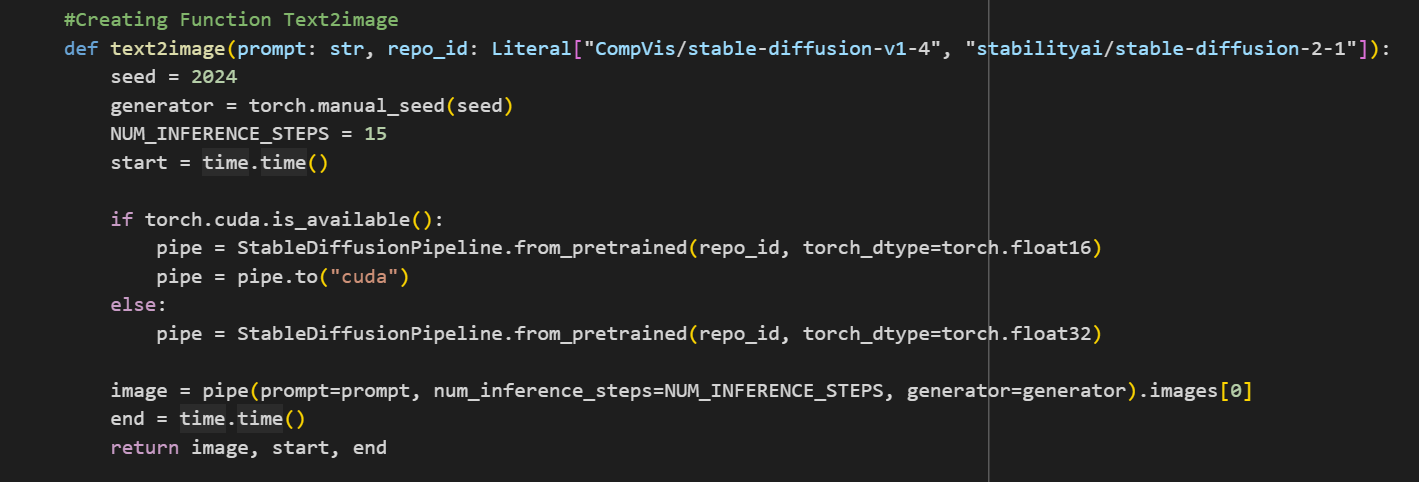
* streamlit is for building the web UI.
* diffusers is for using the Stable Diffusion model.
* accelerate helps manage different device types (CPU/GPU).
* transformers is used for Wav2Vec2 speech recognition.
* torch is the PyTorch deep learning framework.
* librosa is used to process audio files.

Installs nodejs and npm to enable the use of localtunnel.

Installs localtunnel globally to share the app via a public link.

### **Model Pipeline – ImageModel.py**





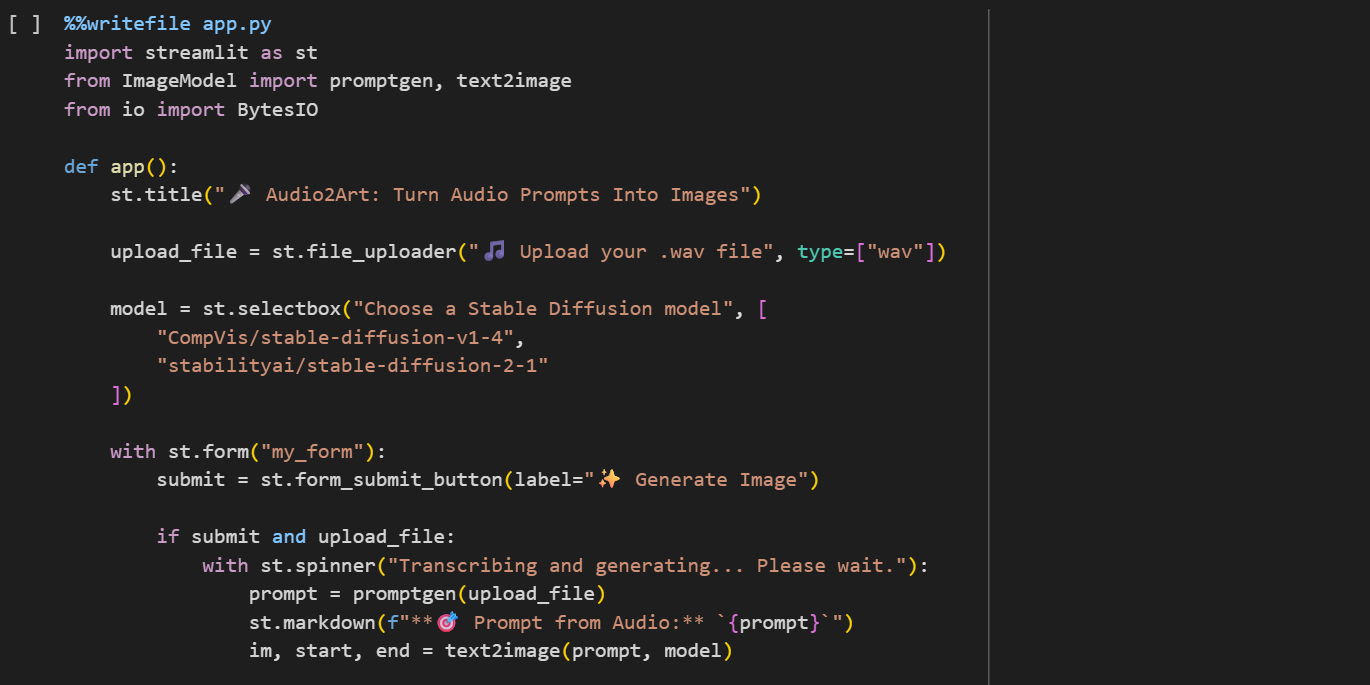
Part 2:

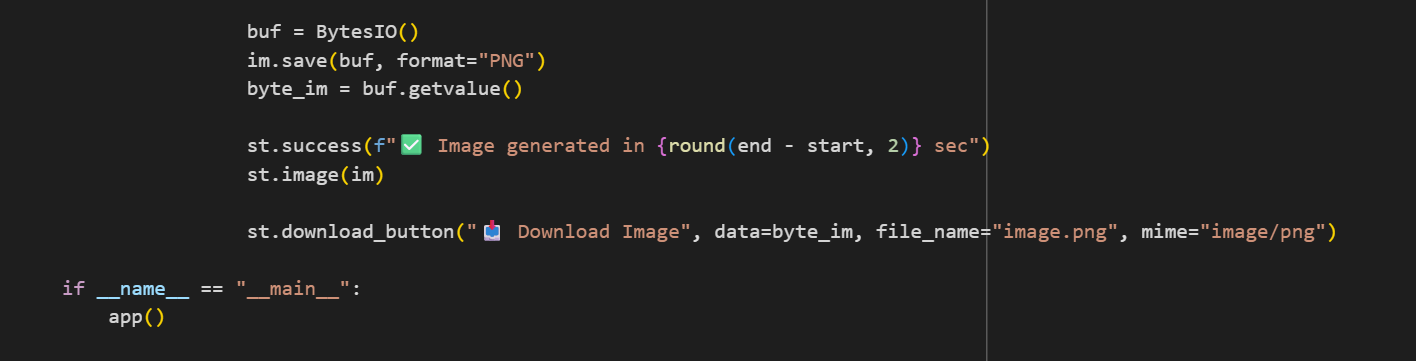
This file defines the core model logic.

promptgen() takes a .wav file, transcribes it using the Wav2Vec2 model, and returns text.

text2image() takes the transcribed text and a model choice, then uses Stable Diffusion to generate an image.

**Web UI – app.py**



  
Part 3:

Creates a user-friendly web interface.

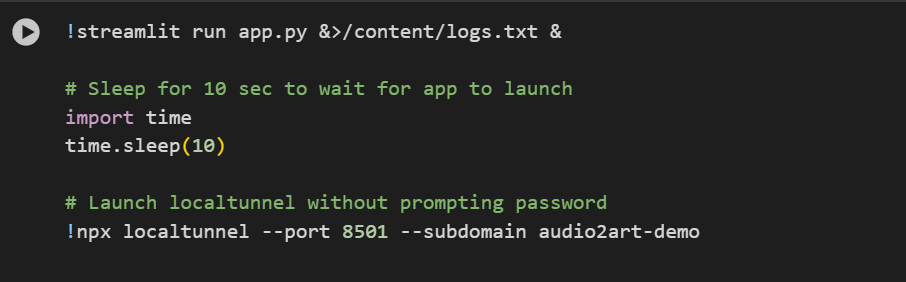
Allows the user to upload a .wav file.

Lets the user choose one of two Stable Diffusion models.

On form submission, transcribes the audio and generates an image.

Displays and offers the image for download.

### **Launch and Share Application**



Part 4:

Runs the Streamlit app in the background.

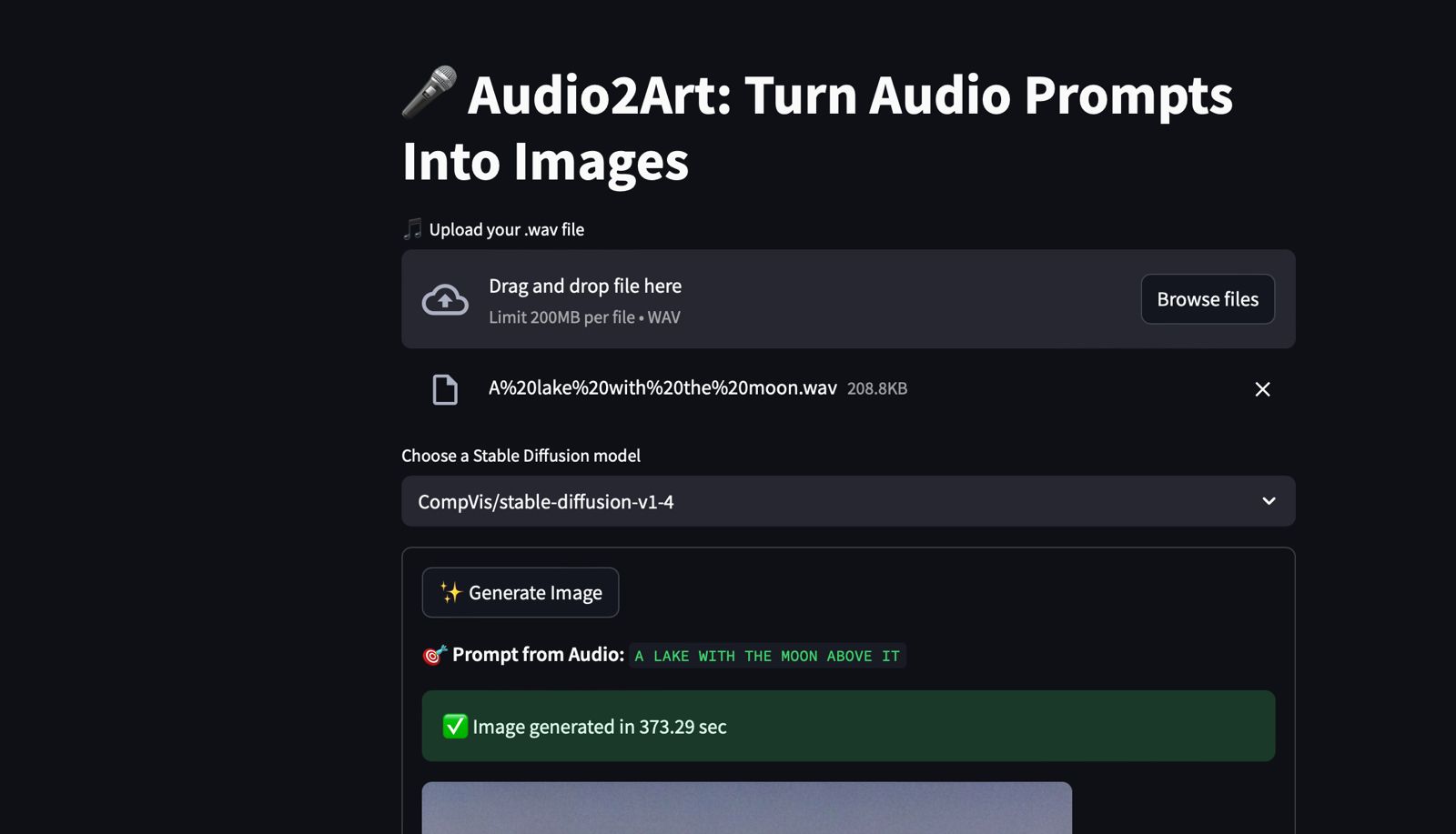
Waits 10 seconds to give it time to start.

Uses localtunnel to make the app accessible on the web at a custom subdomain (audio2art-demo).

## *Testing*

* Audio Input Validation: Tested .wav files of varying audio quality, including background noise scenarios. Transcription maintained high accuracy with minimal delay.
* Prompt Quality Check: Reviewed outputs from the Wav2Vec2 model to ensure the generated text accurately reflected user input.
* Image Generation Accuracy: Validated the consistency and thematic relevance of images generated from prompts using two Stable Diffusion model variants.
* End-to-End Timing: Confirmed that the full process (upload → transcribe → generate → display) completed in under 30 seconds on GPU.
* User Experience: Interface tested for intuitive flow, with positive feedback on usability and simplicity.

**OUTPUT SCREENSHOTS:**

****

## 

## *Advantages and Disadvantages*

### **Advantages**

* Voice-to-image creation offers a novel and accessible creative experience.
* Utilizes high-performing pre-trained transformer models.
* Simple deployment using Google Colab and LocalTunnel.
* Responsive UI with real-time feedback and image preview.
* Download functionality for easy image saving.

### **Disadvantages**

* Dependent on internet access and Colab GPU availability.
* Limited support for only .wav audio format.
* LocalTunnel can have intermittent availability or subdomain conflicts.
* Performance is affected on systems without CUDA-enabled GPUs.

## *Conclusion*

Audio2Art is more than just an application, it’s a creative catalyst. By combining the elegance of human language with the generative power of AI, it opens new frontiers in expression. Whether you're sketching out your next masterpiece, enriching classroom lessons, or facilitating self-reflection, Audio2Art proves that the future of creativity can begin with just your voice.

This project showcases a successful integration of modern transformer-based architectures for both natural language and image processing, all wrapped into a streamlined user experience. It stands as a strong example of practical multimodal AI in action.