Hackathon Project Phases Template

Project Title:

“Audio To Image using Tranformers”

Team Name:

“Audio Visionary”

Team Members:

- N.K.Lakshmi Narasimha Murthy
- G. Jayasimha
- P. Sai charan
- K. Sai Shyam Sundar

Phase-1: Brainstorming & Ideation

Objective:

The primary objective of the given code is to create an **Audio-to-Image Conversion Pipeline** using deep learning models.

Key Points:

1. **Problem Statement:**

   In many real-world scenarios, converting audio into meaningful visual content can enhance understanding, creativity, and accessibility. This project aims to develop a system that takes an **audio file** (such as a voice recording or environmental sound) as input, transcribes it into **text**, and subsequently generates a corresponding **image** that visually represents the audio's content.

2. **Proposed Solution:**
   - **Audio-to-Text Conversion:** The code uses the **Wav2Vec2** model to transcribe audio input into text, ensuring accurate speech recognition even in noisy environments.
   - **Text-to-Image Generation:** The transcribed text is processed by the **Stable Diffusion** model to generate realistic and detailed images, with optional fine-tuning using **LoRA** or **DreamBooth** for improved visual quality.

3. **Target Users:**

   Content Creators & Artists: For generating visaudio stories, or spoken descriptions.

   Educators & Trainers: To create visual aids from lecture recordings .

   Digital Marketers: To convert audio campaigns or slogans into creative visual content.

4. **Expected Outcome:**
   - The code will convert audio files into accurate text descriptions.
   - Based on the transcribed text, it will generate detailed and realistic images.
   - The process should run smoothly, providing clear outputs for both text and images.
   - With optional fine-tuning, the image quality can be improved for better visual results.

Phase-2: Requirement Analysis

Objective:

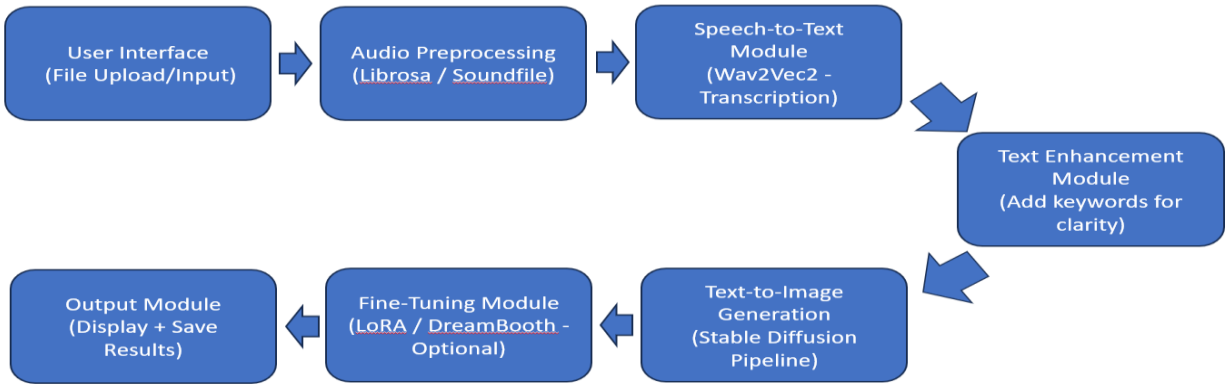
Define the technical and functional requirements for the Audio To Image.

Key Points:

1. **Technical Requirements:**
   - Programming Language: **Python**
   - Backend: Hugging Face Transformers
   - Frontend: **Streamlit Web Framework**
   - Database: **Not required initially (API-based queries)**
2. **Functional Requirements:**
   - **Audio Input Handling:** Accepts audio files in formats like WAV and MP3.
   - **Speech-to-Text Conversion:** Uses **Wav2Vec2** to transcribe audio to text.
   - **Text-to-Image Generation:** Uses **Stable Diffusion** to generate images.
   - **Fine-Tuning (Optional):** Supports LoRA for enhanced model performance.
3. **Constraints & Challenges:**
   - Requires **GPU** for efficient processing; CPU-only execution will be significantly slower.
   - Poor-quality audio with noise or low volume may result in inaccurate transcription.
   - Image generation relies on precise and descriptive text prompts for optimal results.

Phase-3: Project Design

Objective:



Key Points:

1. **System Architecture:**

   - User uploads an audio file through the UI.
   - The backend processes the audio using Wav2Vec2 for transcription.
   - The transcribed text is sent to Stable Diffusion for image generation.
   - The generated image is displayed on the UI.

2. **User Flow:**

   - Step 1: User uploads an audio file.
   - Step 2: The backend transcribes audio to text using Wav2Vec2.
   - Step 3: The text is processed to generate an image using Stable Diffusion.
   - Step 4: The output image and transcribed text are displayed to the user.

3. **UI/UX Considerations:**

   - Simple, user-friendly interface with clear navigation.
   - Visual previews of the generated images for improved user experience.

Phase-4: Project Planning (Agile Methodologies)

Objective:

Break down development tasks for efficient completion.

Sprint	Task	Priority	Duration	Deadline	Assigned To	Dependencies	Expected Outcome
Sprint 1	Environment Setup & API Integration	High	6 hours (Day 1)	End of Day 1	Murthy	Google colab, Python, Streamlit setup	API connection established & working
Sprint 1	Frontend UI Development	Medium	2 hours (Day 1)	End of Day 1	Shyam	API response format finalized	Basic UI with input fields
Sprint 2	Implement audio to text and text to image functionalities	High	3 hours (Day 2)	Mid-Day 2	Sai Charan & Jayasimha	API response, UI elements ready	Search functionality with filters
Sprint 2	Error Handling & Debugging	High	1.5 hours (Day 2)	Mid-Day 2	Murthy & Sai Charan	API logs, UI inputs	Improved API stability
Sprint 3	Testing & UI Enhancements	Medium	1.5 hours (Day 2)	Mid-Day 2	Jayasimha & Shyam	API response, UI layout completed	Responsive UI, better user experience
Sprint 3	Final Presentation & Deployment	Low	1 hour (Day 2)	End of Day 2	Entire Team	Working prototype	Demo-ready project

Sprint Planning with Priorities

Sprint 1 - Environment Setup & Integration (Day 1)

- Setup Python environment and required libraries.
- Integrate Wav2Vec2 and Stable Diffusion models.
- Develop basic UI for file upload and display.

Sprint 2 – Core Features & Debugging (Day 2)

- Implement audio-to-text and text-to-image functionalities.
- Debug transcription errors and improve output quality.

Sprint 3 – Testing & Deployment (Day 3)

- Test model outputs for accuracy and performance.
- Finalize the UI and deploy the project for public use.

Phase-5: Project Development

Objective:

Implement core features of the Audio-to-Image Pipeline.

Key Points:

1. **Technology Stack Used:**
    - **Frontend:** Streamlit
    - **Backend:** Hugging Face Transformers
    - **Programming Language:** Python
2. **Development Process:**
    - Setup environment and dependencies.
    - Implement Wav2Vec2 and Stable Diffusion models.
    - Develop user interface for file uploads and image displays.
3. **Challenges & Fixes:**

    - **Challenge: Slow performance on CPU.**
      Fix: Recommend GPU usage for faster processing.

    - **Challenge: Inaccurate transcription for noisy audio.**
      Fix: Introduce noise reduction preprocessing steps.

Phase-6: Functional & Performance Testing

Objective:

Ensure that the Audio-to-Image Pipeline functions correctly under various conditions.

Test Case ID	Category	Test Scenario	Expected Outcome	Status	Tester
TC-001	Functional Testing	Transcribe clear audio file	Accurate text transcription	Passed	Tester 1
TC-002	Functional Testing	Transcribe noisy audio file	Text with minor errors	Needs Optimization	Tester 2

TC-003	Performance Testing	Generate image from text	Clear and detailed image	Passed	Tester 3
TC-004	Performance	Test processing speed on CPU	Slower but functional output	Passed	Tester 1
TC-005	Final Validation	Ensure UI responsiveness	UI works on both desktop & mobile	Passed	Tester 2
TC-006	Deployment Testing	Host the app using Streamlit Sharing	App should be accessible online.	Deployed	DevOps

---

Final Submission

1. **Project Report Based on the templates**
2. **Demo Video (3-5 Minutes)**
3. **GitHub/Code Repository Link**
4. **Presentation**