Hackathon Project Phases Template for the **Audio2Art** project.

Hackathon Project Phases Template

Project Title:

Audio2Art

Team Name:

Sound Vision

Team Members:

- Ega Sai Charan
- Kethapally Abhishek
- Mohammad Abdus Sami
- Naspuri Aditya
- Mandapalli Shanmukha sai Srinivas

Phase-1: Brainstorming & Ideation

Objective:

Develop an advanced AI-powered tool that converts **audio prompts into visual representations** using transformer models and generative AI. Audio2Art bridges auditory and visual experiences, enabling users to generate images from voice descriptions effortlessly.

Key Points:

1. **Problem Statement:**

- Many users struggle to translate their imagination or verbal descriptions into visuals.
- Artists, educators, and designers often need quick AI-generated visual content from spoken

ideas.

- Current text-to-image AI models require precise textual input, limiting accessibility for users who prefer voice commands.

2. Proposed Solution:

- **Audio2Art** is an AI-powered application that **processes voice descriptions and generates corresponding images** using deep learning and generative AI models.
- The system leverages **transformer-based models** for speech-to-text conversion and a **state-of-the-art image generation model** for visual rendering.
- Users can describe an image in their own words, and the AI will **interpret, process, and visualize** the description accurately.

3. Target Users:

- **Artists & Designers** – For quick concept visualizations.
- **Educators & Students** – For learning through AI-generated images.
- **Content Creators** – To generate visual assets from voice input.
- **Accessibility Users** – For individuals who prefer or require voice-based interactions.

4. Expected Outcome:

- A fully functional **AI-powered Audio-to-Image generation tool** that takes audio prompts and converts them into **high-quality visuals** in real time.
- A seamless **React/Next.js frontend** with a **Python-based backend**, optimized for **VS execution**.
- Enhanced user experience with **speech recognition, deep learning, and generative AI** to create compelling visuals effortlessly.

.

Phase-2: Requirement Analysis

Objective:

Define the technical and functional requirements for the **Audio2Art** application.

Key Points:

1. Technical Requirements:

- **Programming Language:** Python
- **Backend:** Transformer-based models for **speech-to-text processing** and **image generation**
- **Frontend: React/Next.js** for an interactive and intuitive UI

- **Execution Environment:** vs code(Internal Environment)
- **Database:** Not required initially (real-time AI model execution)

2. Functional Requirements:

- Convert **audio descriptions into text** using a speech recognition model.
- Process the text prompt using **generative AI** to create **visual representations**.
- Provide a **user-friendly interface** for voice input and image generation.
- Offer **customization options** (e.g., image style, resolution, and artistic filters).
- Ensure **fast response times** for real-time audio-to-image conversion.

3. Constraints & Challenges:

- Ensuring **accurate speech-to-text conversion** for diverse accents and languages.
- Handling **large model inference times** efficiently on Google Colab.
- Optimizing **image generation quality** while maintaining fast processing.
- Managing **cloud execution limits** and ensuring smooth user experience.

Phase-3: Project Design

Objective:

Develop the **architecture and user flow** of the **Audio2Art** application.

Key Points:

1. System Architecture:

- User provides an **audio description** via the UI.
- **Speech-to-text AI model** converts the voice input into a text prompt.
- The **text prompt** is processed using a **generative AI model** for image creation.
- The **generated image** is displayed on the frontend, with options for customization.

2. User Flow:

- **Step 1:** User speaks a description of the desired image (e.g., "A futuristic city at sunset with flying cars").
- **Step 2:** The backend processes the audio using **speech recognition** and converts it into text.
- **Step 3:** The AI **image generation model** creates a visual based on the text prompt.
- **Step 4:** The frontend displays the generated image with **download and customization options**.

3. UI/UX Considerations:

- **Minimalist, intuitive interface** with a clear call to action for voice input.
- **Real-time audio processing feedback** to show the converted text before image generation.
- **Customization controls** for adjusting image style, resolution, and artistic filters.
- **Dark & light mode** options for enhanced accessibility and 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 & Model Integration	🔴High	6 hours (Day 1)	End of Day 1	**Mandapalli Shanmuka Sai Srinivas**	Python, Speech-to-Text & Image Gen Models	AI models integrated and working
Sprint 1	Frontend UI Development	🟡 Medium	2 hours (Day 1)	End of Day 1	**Kethapally Abhishek**	API response format finalized	Basic UI with audio input & display setup
Sprint 2	Audio Processing & Image Generation	🔴High	3 hours (Day 2)	Mid-Day 2	**Ega Sai Charan**	Speech-to-Text & Image Gen models ready	Functional pipeline for audio-to-image
Sprint 2	Error Handling & Debugging	🔴High	1.5 hours (Day 2)	Mid-Day 2	**Mohammad Abdus Sami & Naspuri Aditya**	API logs, UI inputs	Stable and optimized AI pipeline
Sprint 3	Testing & UI Enhancements	🟡 Medium	1.5 hours (Day 2)	Mid-Day 2	Naspuri Aditya	API response, UI layout completed	Responsive UI, smoother 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 – Setup & Integration (Day 1)

- 🔴**(High Priority)** Set up the development environment & install dependencies.
- 🔴 **(High Priority)** Integrate **speech-to-text AI** and **image generation model**.
- 🟡 **(Medium Priority)** Build a basic UI with **audio input fields** and **image display section**.

Sprint 2 – Core Features & Debugging (Day 2)

- 🔴 **(High Priority)** Implement **audio processing pipeline** (convert speech to text & generate images).
- 🔴 **(High Priority)** Debug AI processing errors & optimize model response times.

Sprint 3 – Testing, Enhancements & Submission (Day 2)

- 🟡 **(Medium Priority)** Test AI model outputs, refine UI, & fix UI-related bugs.
- 🟢 **(Low Priority)** Final **demo preparation & project deployment**.

Phase-5: Project Development

Objective:

Implement the **core features** of the **Audio2Art** application.

Key Points:

1. Technology Stack Used:

- **Frontend:** React/Next.js
- **Backend:** Python-based **Speech-to-Text & Image Generation Models**
- **Execution Environment: VS Code (Local Development with Virtual Environment)**
- **Programming Language:** Python

2. Development Process:

- Implement **speech recognition model** for **audio-to-text conversion**.
- Integrate **generative AI model** to convert text descriptions into images.
- Develop a **user-friendly UI** with **audio input and image display functionality**.

- Optimize **real-time processing** for seamless user experience.
- Utilize **Git** for version control and team collaboration.

3. Challenges & Fixes:

- **Challenge:** Slow image generation times.
  - **Fix:** Optimize model inference using **local GPU acceleration (CUDA/TensorFlow optimizations)**.
- **Challenge:** Inconsistent speech recognition for different accents.
  - **Fix:** Fine-tune speech-to-text preprocessing to improve accuracy.
- **Challenge:** High computational cost of AI model execution.
  - **Fix:** Implement **batch processing and result caching** to minimize redundant computations.
- **Challenge:** Managing dependencies across different team members' setups.
  - **Fix:** Use **virtual environments (venv or conda) and a requirements.txt file** for consistent installations.

Phase-6: Functional & Performance Testing

Objective:

Ensure that the **Audio2Art** application works as expected and delivers optimal performance.

Test Case ID	Category	Test Scenario	Expected Outcome	Status	Tester
TC-001	Functional Testing	Convert audio description: "A sunset over the ocean with dolphins."	A relevant image should be generated.	✅ Passed	Ega Sai Charan
TC-002	Functional Testing	Convert complex audio input: "A futuristic city with neon lights."	AI should interpret and generate the image correctly.	❌ **Failed - Inaccurate image generation**	Kethapally Abhishek

TC-003	Performance Testing	AI model should generate an image in under 5 seconds.	Image should be generated quickly.	❌ **Failed - Processing too slow**	Mohammad Abdus Sami
TC-004	Bug Fixes & Improvements	Fix inaccuracies in speech-to-text conversion.	Improved transcription accuracy.	✅ Fixed	Naspuri Adithya
TC-005	Final Validation	Ensure UI is responsive across devices.	UI should work on mobile & desktop.	❌ Failed - UI b on mobile Tester	Mandapalli Shanmukha Sai Srinivas
TC-006	Deployment Testing	Post the web.	App should be accessible online and version-controlled.	🚀 **Deployed on Git**	Entire Team

Final Submission

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