Hackathon Project Phases Template

Project Title: Audio2Art

Team Name: Auragraph

Team Members: Karli Sri Varshini
Janga Divya
Janumpally Himasree
Kaparthi Jahnavi

Phase-1: Brainstorming & Ideation

Problem Statement:

The challenge is to find creative ways to turn sounds, like music or voices, into visual art. The goal is to use tools like software and AI to create art that is engaging and meaningful. The main problem is figuring out how to effectively transform sound into visuals that are both interesting and easy for the audience to connect with

Proposed Solution:

The proposed system will use software and AI to turn sound into visual art. It will analyze the audio's features, like rhythm and pitch, and create visuals that match the sound. The system will allow users to see dynamic images or animations that change with the audio, creating an interactive and engaging experience. The goal is to make it easy for people to experience the connection between sound and visuals in a fun and creative way.

Target Users:

Musicians and Sound Engineers: Visually representing their music or audio work. **Tech Artists and Designers:** Exploring new creative ways to combine sound and visuals. **Enthusiasts:** Interested in AI and generative art technology. **Educators and Students:** Using the system for learning and creative projects.

Expected Outcome:

To create an engaging, interactive experience that transforms audio into meaningful, dynamic visual art.

Phase-2: Requirement Analysis

Objective:

Define the technical and functional requirements for the Auragraph.

Key Points:

Technical Requirements:

Audio Processing Software: Tools to analyze sound features (e.g., pitch, rhythm).
Generative Art Tools: Software to create visuals based on sound (e.g., Processing).
AI/ML Algorithms: To generate visuals that match the audio in real-time.
Rendering Engine: For displaying visuals synchronized with audio (e.g., Unity).
Interactive Interface: Allow users to interact with the visuals.
Hardware: Devices capable of processing audio and visuals in real-time.

Functional Requirements:

Analyze audio input to extract key features (e.g., rhythm, pitch, frequency).
Generate dynamic visuals based on real-time audio data.
Sync visuals to audio for an immersive experience.
Allow user interaction with visuals (e.g., customization or control).
Support different types of audios (music, voice, nature sounds).
Enable real-time rendering without lag or delays.
Provide an easy-to-use interface for non-technical
users.

Phase-3: Project Design

System Architecture:

- **Audio Input**
  - Input sound through microphone, file, or other sources.
- **Audio Processing**
  - Analyze audio features (e.g., pitch, rhythm, frequency).
- **Visual Generation**
  - Generate visuals based on analyzed audio features.
- **Synchronization**
  - Sync visuals with the audio in real-time.

Phase-4: Project Planning (Agile Methodologies)

Sprint	Task	Priority	Duration	Deadline	Assigned To	Dependencies	Expected Outcome
Sprint 1	Environment Setup & Model Selection	High	6 hours (Day 1)	End of Day 1	Jahnavi	Google Colab, Transformer Models	Model selected & Colab environment ready
Sprint 1	Data Preprocessing & Feature Extraction	Medium	3 hours (Day 1)	End of Day 1	Divya	Raw audio dataset	Cleaned & feature-extracted dataset
Sprint 2	Model Training & Fine-tuning	High	4 hours (Day 2)	Mid-Day 2	Jahnavi	Pre-processed data, Colab setup	Model trained with initial accuracy
Sprint 2	Error Handling & Debugging	High	2 hours (Day 2)	Mid-Day 2	Varshini & Hima	Training logs, loss functions	Improved model stability
Sprint 3	UI & Output Image Generation	Medium	3 hours (Day 2)	Mid-Day 2	Divya	Model outputs, API integration	User interface ready with image output
Sprint 3	Final Testing & Deployment	Low	2 hours (Day 2)	End of Day 2	Entire Team	Trained model, UI setup	Demo-ready project with user input support

Phase-5: Project Development

Objective:

Implement core features of the Auragraph.

Key Points:

Technology Stack Used:

Audio Processing: Librosa, PyDub
Frontend: React, Angular
Backend: Node.js, Python (Flask/Django)
Database: MongoDB, PostgreSQL
Cloud Storage: AWS, Google Cloud

Development Process:

- Planning
- Design
- Coding
- Testing
- Deployment
- Maintenance

Phase-6: Functional & Performance Testing

Objective:

Ensure that the Audio2Art project functions as expected.

Test Case ID	Category	Test Scenario	Expected Outcome	Status	Tester
TC-001	Functional Testing	Convert "Hello World" audio to an image	Image generated with clear representation	✅ Passed	Tester 1
TC-002	Functional Testing	Convert noisy speech to an image	Output image maintains clarity	✅ Passed	Tester 2
TC-003	Performance Testing	Model inference time under 3 seconds	Image should be generated quickly	⚠️ Needs Optimization	Tester 3
TC-004	Bug Fixes & Improvements	Handle accents and different speech tones	Model should work accurately	✅ Fixed	Developer
TC-005	Final Validation	Ensure UI works across devices	UI should be responsive	❌ Failed - UI broken on mobile	Tester 2
TC-006	Deployment Testing	Deploy app using Streamlit	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