

---

# Hackathon Project Phases Template

## Project Title:

Audio2Art

## Team Name:

Golden Dawn

## Team Members:

- T.Sohan Roy
- Ayesha Afreen
- A.Renu Thrisha
- G.Rohith Kumar
- G.Shivaji Ganesh

---

## Phase-1: Brainstorming & Ideation

### Objective:

Audio2Art aims to develop an AI-powered system that converts audio prompts into stunning visual representations using advanced transformer models. This project enhances creativity by seamlessly bridging auditory and visual experiences, enabling effortless image generation from voice descriptions.

### Key Points:

#### 1. Problem Statement:

- Many users struggle to translate their creative ideas into visual representations due to a lack of artistic skills or time constraints.
- Traditional image generation methods require manual effort, limiting accessibility and efficiency.

## 2. Proposed Solution:

- An AI-powered application using **Flask API and deep learning models (Diffusers, Transformers)** to generate artistic visuals from audio inputs.
- The app transforms **audio queries into artistic images**, allowing users to explore creative AI-generated visuals based on sound inputs.

## 3. Target Users:

- **Artists and designers** looking for quick visual concept generation..
- **Content creators and storytellers** who need instant illustrations based on descriptions.
- **Individuals with limited artistic skills** seeking an intuitive way to create digital art.

## 4. Expected Outcome:

- Audio2Art will **enable seamless conversion of voice descriptions into high-quality images**, streamlining creative workflows. It will enhance artistic expression, making visual creation more accessible and efficient.

---

# Phase-2: Requirement Analysis

## Objective:

Define the technical and functional requirements for the Audio2Art Website.

## Key Points:

### 1. Technical Requirements:

- Programming Language: **Python**
- Backend: **Flask API**
- Frontend: **Flask-based UI**
- Machine Learning Frameworks: **PyTorch, Transformers, Diffusers**
- Image Processing: **PIL, NumPy**
- CORS Handling: **Flask-CORS**
- Database: **Not required initially (API-based queries)**

### 2. Functional Requirements:

- Process and **transform audio inputs into artistic visual** outputs.
- Utilize **deep learning models for audio-to-image** generation.
- Provide an **interactive web-based interface** for user input and results display.

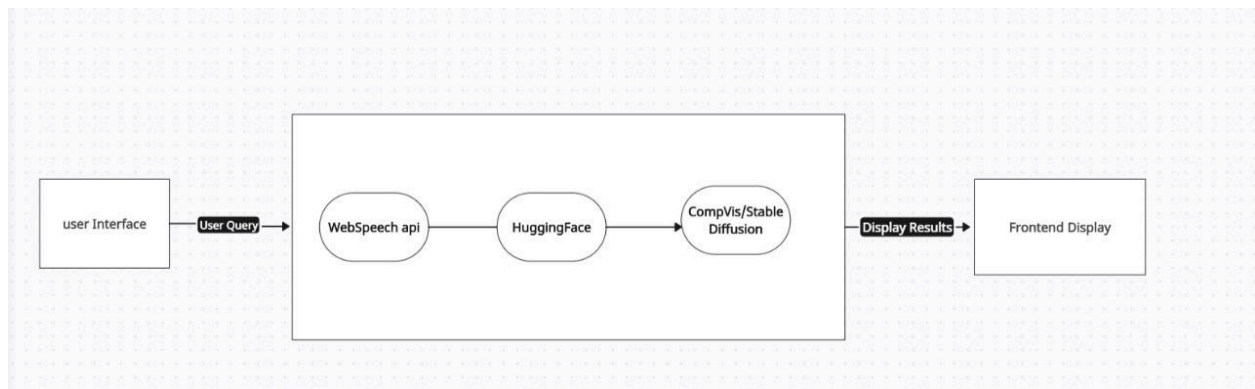
- Ensure real-time processing and smooth user experience.
3. **Constraints & Challenges:**

- **Managing computational resources** for real-time inference.
  - **Optimizing API calls** to prevent latency issues.
  - **Handling cross-origin requests** efficiently using Flask-CORS.
- 

## Phase-3: Project Design

### Objective:

Develop the architecture and user flow of the application.



### Key Points:

1. **System Architecture:**

- Users input an **audio query** or **text query** via the UI.
- Query is processed using **Flask API** and deep learning models (**Diffusers, Transformers**).
- **AI model** generates and processes **artistic visual outputs**.
- The frontend displays the generated artwork based on the audio input.

2. **User Flow:**

- Step 1: The user provides a **voice description using the Web Speech API**, which converts the spoken input into text..
- Step 2: The text input is processed using the **Hugging Face API**, which interprets the description and generates a structured prompt for image creation..
- Step 3: The **CompVis/Stable Diffusion model** generates a high-quality image based on the processed prompt.

- Step 4: The generated image is displayed on the **UI**, allowing users to view and refine their artwork.








### 3. UI/UX Considerations:

- **Minimalist, user-friendly interface** for seamless navigation.
- **Voice prompt & Text Prompt** for better 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	Sohan	Hugging Face Api, Web speech	API connection established & working
Sprint 1	Frontend UI Development	 Medium	2 hours (Day 1)	End of Day 1	Sohan	API response format finalized	Basic UI with input fields
Sprint 2	Speech Recognition	 High	1.5 hours (Day 2)	Mid-Day 2	Shivaji & Rohith	API response	Accurate text from voice prompt
Sprint 2	Image generation	 High	1.5 hours (Day 2)	Mid-Day 2	Ayesha & Thrisha	API response	Good quality images from text
Sprint 2	Error Handling & Debugging	 High	1.5 hours (Day 2)	Mid-Day 2	Sohan	API logs, UI inputs	Improved API stability
Sprint 3	Testing & UI Enhancements	 Medium	1.5 hours (Day 2)	Mid-Day 2	Sohan	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 – Setup & Integration (Day 1)

- (● High Priority) Set up the **environment** & install dependencies.
- (● High Priority) Integrate **Flask API** with **Diffusers & Transformers**.
- (● Medium Priority) Build a basic UI with input fields for **audio/text input**.

## Sprint 2 – Core Features & Debugging (Day 2)

- (● High Priority) Implement audio-to-art transformation logic.
- (● High Priority) Debug **model inference issues** & handle **Flask API errors**.

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

- (● Medium Priority) Test API responses, refine UI, & fix UI bugs.
- (● Low Priority) Final **demo preparation & deployment**.

---

# Phase-5: Project Development

## Objective:

Implement core features of the Audio2Art Website.

## Key Points:

### 1. Technology Stack Used:

- **Frontend: Flask-based UI**
- **Backend:** Flask API, Deep Learning Models (Diffusers, Transformers)
- **Programming Language:** Python

### 2. Development Process:

- Implement **API key authentication** and **Flask API integration**.
- Develop **audio-to-art transformation logic** using AI models.
- **Optimize inference speed** for real-time art generation.

### 3. Challenges & Fixes:

- **Challenge:** Delayed AI inference times.  
**Fix:** Implement model optimization techniques like **quantization** and **caching**.

- **Challenge:** High resource consumption during image generation.  
**Fix:** Use efficient **batch processing** and **GPU acceleration**.

---

## Phase-6: Functional & Performance Testing

### Objective:

Ensure that the Audio2Art App works as expected.

Test Case ID	Category	Test Scenario	Expected Outcome	Status	Tester
TC-001	Functional Testing	Query "Image of a Lion with pink hair"	A Lion with pink hair.	✅ Passed	Shivaji
TC-002	Functional Testing	Query "Virat Kohli holding a bouquet"	Image of Virat Kohli holding a bouquet	✅ Passed	Rohith
TC-003	Performance Testing	API response time within few seconds	API should return results quickly.	⚠ Needs Optimization	Ayesha
TC-004	Bug Fixes & Improvements	Fixed incorrect API responses.	Data accuracy should be improved.	✅ Fixed	Sohan
TC-005	Final Validation	Ensure UI is responsive across devices.	UI should work on mobile & desktop.	❌ Failed - UI broken on mobile	Thrisha
TC-006	Deployment Testing	Host the app using Local Tunnel Sharing	App should be accessible online.	🚀 Deployed	Local Tunnel

---

## Final Submission

1. **Project Report Based on the templates**
2. **Demo Video (3-5 Minutes) :**
3. **GitHub/Code Repository Link : <https://github.com/sohanroy676/Audio2Art.git>**
4. **Presentation**