Hackathon Project Phases Template for the **AutoSage App** project.

Hackathon Project Phases Template

Project Title:

Audio2Art

Team Name:

codemasterzz

Team Members:

- UL Sivani
- M Sreeja
- R Srija
- CH Varsha
- P Samyuktha

Phase-1: Brainstorming & Ideation

Objective:

develop a website that converts audio into images using user's input voice prompts , creating visual representations based on audio features.

Key Points:

1. **Problem Statement:**

- inaccurate audio-image mapping due to noisy inputand challenges in generating meaningful visuals that properly represent sound characteristics.

2. **Proposed Solution:**

- the website allows the user to modify the text incase it was not detected properly

3. **Target Users:**

- Content Creators & Designers
- Educators & Students
- Marketing & Branding Agencies

4. **Expected Outcome:**

a website that would be able togenerate prompt images based on user's audio input

---

Phase-2: Requirement Analysis

Objective:

Define the technical and functional requirements for the audio to image generator

Key Points:

1. **Technical Requirements:**

audio-text:wav2vec2
text-audio:stable diffusion and transformers
librosa to convert audio files
transformer architecture:pytorch

2. **Functional Requirements:**

1.Accept voice input and convert it into text.
2.Process text and generate real-time images based on prompts.

3. **Constraints & Challenges:**

ensuring lo-latency real time generation
handling different accents in voice input

Phase-3: Project Design

Objective:

Develop the architecture and user flow of the application.

Key Points:

1.System Architecture :

- Frontend: audio-text: transcribe audio
- Flask Backend:text to image
- AI Model: hugging face stable diffusion for ai generation

user flow:

- Voice Input- Speaks into the microphone to provide a prompt.
- Processing– Audio converts to text, then text generates an image.
- Image Display*– Generated image appears in real time.

UI/UX Considerations:

UI:Clean design, easy voice input, real-time image display, responsive layout.

UX: Smooth processing, multilingual support, feedback options, loading indicators.

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	sivani	token ,python, streamlit	token connection established & working Basic UI
Sprint 1	Frontend UI Development	Medium	2 hours (Day 1)	End of Day 1	sreeja		with input fields
Sprint 2	Vehicle Search & Comparison	High	3 hours (Day 2)	Mid-Day 2	srija	token response, UI elements ready	Search functionality with filters
Sprint 2	Error Handling & Debugging	High	1.5 hours (Day 2)	Mid-Day 2	varsha	token logs, UI inputs	Improved token stability
Sprint 3	Testing & UI Enhancements	Medium	1.5 hours (Day 2)	Mid-Day 2	samyuktha	token response, UI layout completed	Responsive UI, better user experience
Sprint 3	Final Presentation & Deployment	Low	1 hour (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 **Google Gemini API**.
(🟡 **Medium Priority)** Build a **basic UI with input fields**.

Sprint 2 – Core Features & Debugging (Day 2)

(🔴 **High Priority)** Implement **search & comparison functionalities**.
(🔴 **High Priority)** Debug API issues & handle **errors in queries**.

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 Audio to art website.

Key Points:

1. **Technology Stack Used:**

   Frontend: Streamlit
   Backend: hugging face stable diffusion
   Programming Language: Python

2. **Development Process:**

   ○ Implement :hugging face token
   . ○ Optimize **search queries for performance and relevance**.

3. **Challenges & Fixes:**

   t image generated
   ○ **Challenge:** Delayed API response times.
   Fix: Implement **caching** to store frequently queried results.
   ○ **Challenge:** Limited API calls per minute.
   Fix: Optimize queries to fetch **only necessary data**.

Phase-6: Functional & Performance Testing

Objective:

ensure that Audio to art website works as expected

Test Case ID	Category	Test scenario	Expected Outcome	Status	
TC-001	Functional Testing	mountains	exact image generated.	✅ Passed	
TC-002	Functional Testing	family enjoying vacation on beach	exact image generated	✅ Passed	

TC-003	Performance Testing	API response time under 500ms	API should return results quickly.	⚠ Needs Optimization	
TC-004	Bug Fixes & Improvements	Fixed incorrect API responses.	Data accuracy should be improved.	✅ Fixed	
TC-005	Final Validation	Ensure UI is responsive across devices.	UI should work on mobile & desktop.	❌ Failed - UI broken on mobile	T
TC-006	Deployment Testing	Host the app using Streamlit Sharing	App should be accessible online.	🚀 Deployed	

Final Submission

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