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

**Hackathon Project Phases Template**

**Project Title: Audio2Image**

**AutoSage App Using Gemini Flash**

**Team Name:Dreamhack**

**Team Members:**

* Ch.Dedeepya
* K.Jyothsna
* D.Sowmya
* Ch.Meghana

**Phase-1: Brainstorming & Ideation**

**Objective:**

Develop an AI-powered system that converts real-time or recorded audio into relevant images using deep learning and transformer models.

1. **Problem Statement:**
   * Many applications require visual representation of speech or sound for accessibility, education, and creative purposes.
   * There is a need for a system that can generate meaningful images from spoken words, making audio more interpretable and engaging.
2. **Proposed Solution:**
   * An AI-driven system that transcribes audio, analyzes its meaning, and generates images using a transformer-based model.
   * The system integrates speech recognition, NLP, and image generation models to create accurate visual representations of spoken words.
3. **Target Users:**

**Content creators and educators who want to visualize spoken concepts.**

**Accessibility tools for individuals with hearing impairments.**

**Creative applications in storytelling and media production.**

**4.Expected Outcome:**

A functional AI system that takes audio input and generates corresponding images with high accuracy and relevance.

**Phase-2: Requirement Analysis**

**Objective:**

Define the technical and functional requirements for the Audio-to-Image system.

**Key Points:**

1. **Technical Requirements:**
   * Programming Language: **Python**

* Speech Recognition: Google Speech-to-Text API / Whisper AI
* NLP Processing: Transformers (BERT/GPT-based)
* Image Generation: Stable Diffusion / DALL-E / Custom GAN Model
* Backend: Flask/FastAPI
* Frontend: React / Streamlit
* Database: Not required initially (API-based model calls)

**2.Functional Requirements:**

* Convert speech/audio input into text.
* Process transcribed text to extract key themes.
* Generate images based on extracted themes using AI models.
* Provide an intuitive UI for users to interact with the system.

**3.Constraints & Challenges:**

* Ensuring accurate speech recognition for different accents and noise levels.
* Efficient text-to-image mapping for meaningful image generation.
* Optimizing AI model performance to reduce processing time.

**Phase-3: Project Design**

**Objective:**

Develop the architecture and user flow of the application.



**Key Points:**

1. **System Architecture:**

* User inputs audio via microphone or uploads a file.
* Speech-to-text model transcribes the input.
* NLP module extracts relevant themes.
* AI image generation model creates visuals based on extracted text.
* The frontend displays the generated images.

**2. User Flow:**

* + Step 1: User provides an audio input.
  + Step 2: The system processes the audio into text.
  + Step 3: Extracted text is analyzed for key concepts.
  + Step 4: An AI model generates images based on the concepts.
  + Step 5: The final image is displayed to the user.

1. **UI/UX Considerations:**
   * **Simple, intuitive interface for audio upload and playback.**
   * **Display transcription before image generation.**
   * **Options to refine image generation based on user preference**

**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 | 2 hours (Day 1) | Mid of Day 1 | Jyothsna | API Keys, Model Setup | API connection established & working |
| Sprint 2 | Frontend UI Development | 🟡 Medium | 2 hours (Day 1) | Mid of Day 1 | Meghana | API response format finalized | Basic UI with input fields |
| Sprint 3 | Audio Processing & Transcription | 🔴 High | 3 hours (Day 2) | End of Day 1 | Dedeepya | API response, UI elements ready | AI-generated images displayed |
| Sprint 3 | Error Handling & Debugging | 🔴 High | 3 hours (Day 2) | End of Day 1 | Sowmya | Logs & bug reports | Improved API stability |
| Sprint 4 | Image Generation Model Integration | 🟡 Medium | 2 hours (Day 2) | Mid-Day 2 | Entire team | Text processing module ready | Responsive UI, better user experience |
| Sprint 4 | Final Presentation & Deployment | 🟢 Low | 1 hour (Day 2) | Mid 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.  
 **(🟡 Medium Priority)** Build a **basic UI with input fields**.

**Sprint 2 – Core Features & Debugging (Day 1)**

**(🔴 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-Image system.

**Key Points:**

1. **Technology Stack Used:**
   * **Frontend:** Streamlit
   * **Backend:** Flask
   * **Programming Language:** Python
2. **Development Process:**
   * Implement audio processing and transcription.
   * Develop NLP pipeline for theme extraction..
   * Integrate AI-based image generation models.
3. **Challenges & Fixes:**
   * **Challenge** **Noisy audio affects transcription accuracy.**
   * **Fix:** Implement noise reduction and improve speech model.
   * **Challenge:** Generated images may not always match audio themes
   * **Fix:** Fine-tune prompt engineering for better accuracy

**Phase-6: Functional & Performance Testing**

**Objective:**

Ensure that the AutoSage App works as expected.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Category** | **Test Scenario** | **Expected Outcome** | **Status** | **Tester** |
| TC-001 | Functional Testing | Convert spoken sentence to text | Correct text output | ✅ Passed | Jyothsna |
| TC-002 | Functional Testing | Generate image from text | Relevant image output | ✅ Passed | Meghana |
| TC-003 | Performance Testing | Process audio & generate image in under 2s | API should return results quickly. | ✅ Fixed | Sowmya |
| TC-004 | Bug Fixes & Improvements | Fix UI alignment issues | System responds quickly | ✅ Fixed | Dedeepya |
| TC-005 | Final Validation | Ensure UI is responsive across devices. | UI should work on mobile & desktop. | ✅ Fixed | Sowmya |
| TC-006 | Deployment Testing | Host the app using Streamlit Sharing | App should be accessible online. | 🚀 Deployed | Meghana |

**Final Submission**

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