Hackathon Project Phases Template for the Transforming voice prompts into visual creations using transformers project.

Hackathon Project Phases Template

Project Title:

Transforming voice prompts into visual creations using transformers

Team Name:

Speak scene

Team Members:

- Md Minshaniya
- M Sreeja
- K Saniya Darja

Phase-1: Brainstorming & Ideation

Objective:

Develop an AI-powered tool that converts voice prompts into visual representations using OpenAI's Whisper, GPT-3.5, and DALL·E.

Key Points:

1. **Problem Statement:**
    - Manual image creation requires artistic skills and time.
    - Need for an AI-driven solution to automate speech-to-image conversion.

- Enhance accessibility for individuals with disabilities or creative constraints

2. **Proposed Solution:**
    - Use **Whisper** for voice transcription.
    - Employ **GPT-3.5** to refine and generate image descriptions.
    - Utilize **DALL·E 2** to create high-quality AI-generated visuals.

3. **Target Users:**
    - Content creators who need quick concept images.
    - Designers and artists for idea visualization.
    - Educators and students for interactive learning
    .

4. **Expected Outcome:**
    - A functional AI-powered tool capable of transforming speech into relevant, high-quality images.

Phase-2: Requirement Analysis

Objective:

Define technical and functional requirements for the AI-driven speech-to-image application.

Key Points:

1. **Technical Requirements:**

    - **Programming Language:** Python
    - **Backend:** Flask/FastAPI
    - **Frontend:** Gradio for a user-friendly interface
    - **AI Models:** OpenAI Whisper, GPT-3.5 Turbo, DALL·E 2
    - **Cloud Storage:** AWS/GCP
    - **Database:** PostgreSQL/Firebase
    -

2. **Functional Requirements:**

    - Speech-to-text conversion with Whisper.
    - Automatic summarization and optimization of text prompts with GPT-3.5.
    - Image generation using DALL·E 2.

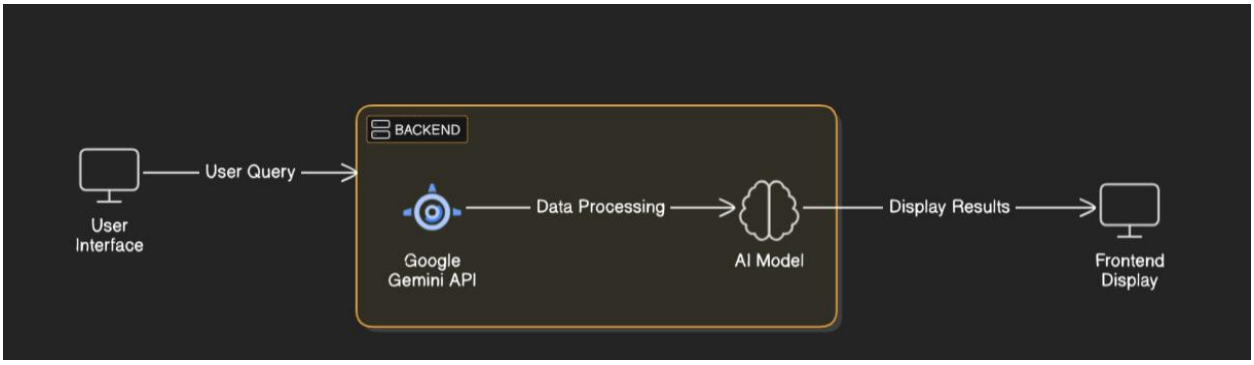
- User interface to display transcriptions and generated images.
- Secure data handling and encryption.

3. **Constraints & Challenges:**
    - Ensuring accurate interpretation of voice commands.
    - Balancing processing speed and image quality.
    - Handling complex and ambiguous prompts effectively.

Phase-3: Project Design

Objective:

Develop the architecture and user flow of the application.



Key Points:

1. **System Architecture:**

    - User inputs voice prompts.
    - Whisper AI transcribes the speech.
    - GPT-3.5 refines and optimizes the prompt.
    - DALL·E 2 generates an image.
    - Results displayed in the Gradio interface.

2 . User Flow:

- **Step 1:** User speaks into the microphone.
- **Step 2:** Whisper AI transcribes the audio into text.
- **Step 3:** GPT-3.5 processes and e**Step **nhances the prompt.
- **Step 4:** DALL·E 2 generates a relevant image.

- **Step 5:** User reviews and downloads the image.

.

3 . UI/UX Considerations:

- Simple, intuitive interface for easy use.
- Real-time preview of transcriptions and images.
- Dark and light mode support.

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	End of day1	K.Saniya	API key setup, dependencies	Working API integration
Sprint 1	UI Development	🟡 Medium	4 hours (Day 1)	End of Day 1	Md.Minshaniya	API response handling	Basic UI ready
Sprint 2	Speech-to-Text Implementation	🔴 High	3 hours (Day 2)	Mid-Day 2	MSreeja& K.Saniya	UI & API ready	Conversion of speech to text
Sprint 2	image Generation Pipeline .	🔴 High	3 hours (Day 2)	Mid-Day 2	Md.Minshaniya & M.Sreeja	Image generation complete	Working text-to-image model
Sprint 3	Testing & Debugging	🟡 Medium	1.5 hours (Day 2)	Mid-Day 2	QA Team	Complete project setup	Stable version
Sprint 3	Final demo preparation & 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** & dependencies.
(🔴 **High Priority)** Integrate **speech-to-text model**.
(🟡 **Medium Priority)** Design **UI** mockups.

Sprint 2 – Core Features & Debugging (Day 2)

(🔴 **High Priority)** Implement **transformer-based image generation**.
(🔴 **High Priority)** Test voice-to-image pipeline.

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 and and optimize the AI pipeline.

Key Points:

1. **Technology Stack Used:**
    - Frontend: Gradio
    - Backend: Flask/FastAPI
    - AI Models: Whisper, GPT-3.5, DALL·E 2
    - Deployment: Docker, AWS/GCP
2. **Development Process:**
    - Implement API authentication and request handling.
    - Develop speech-to-text and text-to-image pipelines.
    - Optimize application for performance and scalability.

3. **Challenges & Fixes:**
    - Challenge: Slow API response times → Fix: Implement caching.
    - Challenge: Limited API requests → Fix: Optimize calls and handle errors.

Phase-6: Functional & Performance Testing

Objective:

Ensure that the project works as expected.

Test Case ID	Category	Test Scenario	Expected Outcome	Status	Tester
TC-001	Functional Testing	Speech input recognition	Accurate transcription	✅ Passed	Tester 1
TC-002	Functional Testing	Image generation accuracy	Relevant AI-generated image.	✅ Passed	Tester 2
TC-003	Performance Testing	API response time under 500ms	API should return results quickly.	⚠ Needs Optimization	Tester 3
TC-004	Bug Fixes & Improvements	Fixed incorrect API responses.	Data accuracy should be improved.	✅ Fixed	Developer
TC-005	Final Validation	Ensure UI is responsive across devices.	UI should work on mobile & desktop.	✅Fixed	Tester 2
TC-006	Deployment Testing	Host the UI using Streamlit Sharing	UI 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**