GENAI Hackathon Project

Project Title:

Audio2Art: Transforming Voice Prompts into Visual Creations using Transformers

Team Name:

404 Brain Not Found

Team Members:

- S.Shyam Sundar
- M.Venkat ramana
- Mohammad Rabbani Pasha
- CH.Manikya Kalyan
- G.Vivek Vardhan

Phase-1: Brainstorming & Ideation

Objective:

To develop an advanced AI system that seamlessly transforms verbal audio descriptions into high-quality visual artwork, bridging the gap between auditory input and visual output through innovative technology..

Key Points:

- **Core Technology**: Utilizes transformer models and cutting-edge AI to process and interpret audio descriptions with high accuracy.
- **User Experience**: Enables effortless creation of visual content through simple voice commands, eliminating technical barriers.
- **Versatility**: Adaptable across multiple use cases including creative production, accessibility tools, educational applications, and commercial design.
- **Innovation Focus**: Represents a novel approach to human-computer interaction by connecting auditory and visual domains in an intuitive way.
- **Customization Capabilities**: Provides tailored visual outputs based on specific verbal descriptions, giving users creative control.
- **Accessibility Benefits**: Makes visual creation accessible to those with limited technical skills or physical limitations that make traditional design tools challenging.
- **Market Differentiation**: Stands apart from existing visual generation tools through its specialized audio-to-visual conversion approach.
- **Scalability**: Designed to handle both simple requests and complex descriptive narratives with equal effectiveness.

Phase-2: Requirement Analysis

Objective:

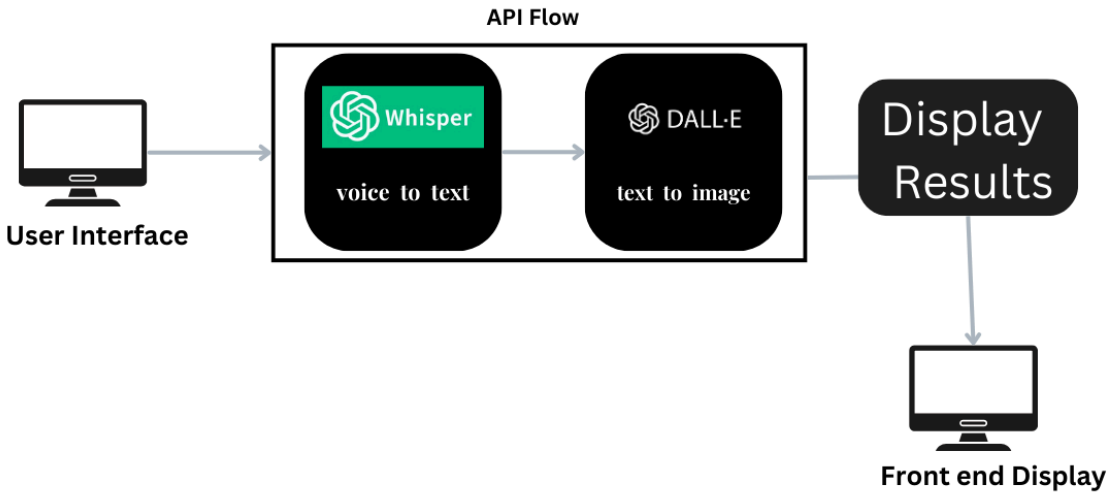
To identify, analyze, and document the comprehensive functional and technical requirements for the Audio2Art system, ensuring it effectively converts audio descriptions into high-quality visual representations while meeting user needs across diverse use cases.

Key Points:

- **System Architecture Requirements**
  - Define the AI and transformer model specifications needed for accurate audio-to-image conversion
  - Establish performance benchmarks for processing speed and output quality
  - Determine cloud infrastructure and computing resource requirements
- **Audio Processing Capabilities**
  - Identify requirements for multi-language support and dialect recognition
  - Define noise filtering and audio quality enhancement needs
  - Establish requirements for handling different audio input formats and durations
- **Visual Output Requirements**
  - Document image quality standards and resolution requirements
  - Define stylistic variation capabilities and customization options
- **User Experience Analysis**
  - Map the complete user journey from audio input to image generation
  - Identify accessibility requirements for diverse user groups
  - Define interface simplicity requirements to ensure "effortless" use
- **Use Case Mapping**
  - Document requirements for creative professional applications
  - Analyze educational implementation requirements
  - Define commercial application requirements and integration capabilities
- **Performance Metrics**
  - Define accuracy standards for audio-to-visual translation
  - Establish response time requirements for various complexity levels
  - Create quality assessment frameworks for evaluating generated images

Phase-3: Project Design

Objective:



To establish a robust, scalable, and efficient architectural framework for the Audio2Art system that effectively processes audio inputs and generates high-quality visual outputs while maintaining optimal performance and user experience.

Key Points:

Microservice Architecture

- Implement a decoupled microservice design pattern for flexibility and independent scaling
- Separate core functionalities into distinct services: Audio Processing, Text Transformation, Image Generation, User Management

Data Flow Pipeline

- Audio Capture → Speech Recognition → Text Processing → Prompt Engineering → Image Generation → Post-Processing → Delivery
- Design for fault tolerance with retry mechanisms and graceful degradation

AI Model Infrastructure

- Implement dual-phase transformer model approach: audio-to-text model followed by text-to-image model
- Utilize specialized audio processing neural networks for feature extraction and noise reduction
- Deploy advanced diffusion models for high-quality image generation with style customization

Ability to Download and share

- Implementation of download button for downloading the image into the system
- Usage of media share button for sharing the image directly through media apps

User Experience Flow

- Web/Mobile Interface → Audio Capture → Processing Feedback Loop → Refinement Options → Final Output
- Implement real-time progress indicators with estimated completion times

Security Architecture

- Implement end-to-end encryption for audio data in transit and at rest
- Design privacy-preserving processing pipelines with data minimization principles
- Establish comprehensive access controls and audit logging for all system interactions

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	1 hours (Day 1)	End of Day 1	Shyam	OpenAI API Key,Python	API connection established & working
Sprint 1	Jupyter notebook development	🔴High	6 hours (Day 1)	End of Day 1	Shyam, Rabbani	API response format finalized	Basic UI with input fields
Sprint 2	Model setup ,UI Development	🔴 High	3 hours (Day 2)	Mid-Day 2	Rabbani, Venkat, Shyam	API response, UI elements ready	Search functionality with filters
Sprint 2	Error Handling & Debugging	🔴 High	1.5 hours (Day 2)	Mid-Day 2	Shyam, Kalyan, Rabbani	API logs, UI inputs	Improved API stability
Sprint 3	Testing & UI Enhancements	🟡 Medium	1.5 hours (Day 2)	Mid-Day 2	Vivek	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 **OpenAI API**.
(🟡 **Medium Priority)** Build a **basic UI with input fields**.

Sprint 2 – Core Features & Debugging (Day 2)

(🔴 **High Priority)** Implement **Development of both back and front end**.
(🔴 **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 AutoSage App.

Key Points:

User Input & Processing

- **Voice Input & Recognition:** WebRTC, FastAPI, DeepSpeech, Whisper.
- **Text Preprocessing:** NLTK, SpaCy, TensorFlow NLP.

AI Processing & Image Generation

- **Text-to-Image Models:** Stable Diffusion, DALL·E, GANs, CLIP.
- **Style Customization:** OpenCV, PyTorch, StyleGAN.

Backend & Cloud Infrastructure

- **Model Hosting:** AWS, Google Cloud, TensorFlow Serving, Kubernetes.
- **Database & Storage:** PostgreSQL, MongoDB, Firebase.
- **API Services:** FastAPI, Flask, GraphQL.

Frontend & User Interface

- **Web & Mobile App:** React.js (Web), Flutter (Mobile).

- **Real-time Processing:** WebSockets, Redis Cache.

Deployment & Security

- **Scalability:** Docker, Kubernetes, NGINX.
- **Security:** OAuth 2.0, JWT, AES Encryption

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	Query "generate a picture of dog"	Picture of dog.	✅ Passed	Vivek
TC-002	Functional Testing	Query "city with neon lights"	Picture of city with neon lights	✅ Passed	
TC-003	Performance Testing	API response time under 20s	API should return results quickly.	⚠ Needs Optimization	Venkat
TC-004	Bug Fixes & Improvements	Fixed incorrect API responses.	Data accuracy should be improved.	✅ Fixed	Rabbani
TC-005	Final Validation	Ensure UI is responsive across devices.	UI should work on mobile & desktop.	✅ Passed	kalyan
TC-006	deployment	Runs on local host	Performs without issues on the local host	🚀 Deployed	DevOps

---

Final Submission

1. **Project Report Based on the templates**
2. **GitHub/Code Repository Link**
3. **Presentation**