**Hackathon Project Phases Template** for the “ **audio transcription app using Open AI Whisper ”** project

Hackathon Project Phases Template

# Project Title:

Audio transcription app using OpenAI Whisper

# Team Name:

**(Hackaholics)**

# Team Members:

* Datla Shravya
* S.Vijay
* G.Archana
* D.Sai Kiran

# Phase-1: Brainstorming & Ideation

## Objective:

Develop an accurate and efficient audio transcription app using OpenAI Whisper, providing users with fast, reliable, and editable transcripts, while improving productivity and accessibility."

## Key Points:

1. **Problem Statement:** The challenge of accurately transcribing audio content in real-time from diverse sources like meetings, lectures, and podcasts, especially with background noise and multiple languages.
2. **Proposed Solution:** An AI-powered audio transcription app utilizing OpenAI Whisper to convert speech into text with real-time processing, noise filtering, and multi-language support.
3. **Target Users:** Professionals, students, content creators, and anyone in need of accurate, efficient transcription for various audio content, regardless of environment or language.
4. **Expected Outcome:** A precise, real-time transcription tool using OpenAI Whisper that supports multiple languages and environments, enhancing productivity and accessibility

# Phase-2: Requirement Analysis

## Objective: Identify the essential technical and functional requirements while analyzing potential constraints and challenges for developing the Audio Transcription App using OpenAI Whisper.

## Key Points:

1. **Technical Requirements:**
   * Programming Language: **Python**
   * Backend: **softwares FFmpeg , whisper model**
   * Frontend: **gradio Python library**
   * Database: **google colab**
2. **Functional Requirements:** Speech-to-text conversion, multi-language support, noise filtering, punctuation handling, and integration with various audio sources.

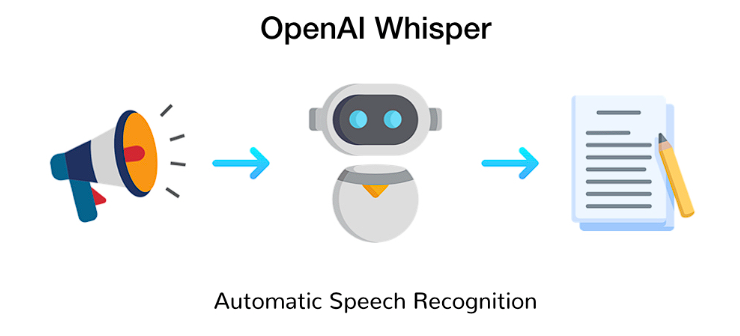
.

1. **Constraints & Challenges:** Handling noisy environments, processing large audio files efficiently, ensuring real-time accuracy, and optimizing for multiple languages.

# Phase-3: Project Design

## Objective:

Define the overall structure, user interaction, and interface design for the audio transcription systemusing OpenAI Whisper.



## Key Points:

1. **System Architecture:** A modular architecture with audio input processing, Whisper API integration, real-time transcription, data storage, and an interactive user interface
2. **.User Flow:** Users upload or record audio → System processes and transcribes using Whisper → Text output is displayed and can be edited, saved, or exported.
3. **UI/UX Considerations:** A clean, intuitive interface with easy audio upload, live transcription preview, multi-language selection, and accessibility features for a seamless user experience.

# Phase-4: Project Planning (Agile Methodologies)

## Objective:

Break down the project into sprints using Agile methodology, defining tasks, priorities, deadlines, and dependencies to ensure smooth execution within the 2-day deadline.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Task** | **Priority** | **Duration** | **Deadline** | **Assigned To** | **Dependencies** | **Expected Outcome** |
| Sprint 1 | Set up development environment | ✅  High | 4 hours | Day 1 | Developer | None | Ready environment for coding |
| Sprint 1 | Integrate OpenAI Whisper API | ✅  High | 6 hours | Day 1 | Developer | Dev setup complete | API integrated for transcription |
| Sprint 2 | Design basic UI layout | 🔹  Medium | 4 hours | Day 1 | UI/UX Team | None | Initial wireframe for user interaction |
| Sprint 2 | Implement audio input handling | ✅  High | 5 hours | Day 2 | Developer | API integration | Users can upload/record audio |
| Sprint 3 | Process and transcribe audio | ✅  High | 6 hours | Day 2 | Developer | Audio input setup | Real-time transcription enabled |
| Sprint 3 | Display and edit text output | 🔹  Medium | 4 hours | Day 2 | UI/UX Team | Transcription logic | Users can view and edit transcripts |

## Sprint Planning with Priorities

* **Sprint 1 (Day 1 - Setup & Integration)**
* ✅ High Priority: Development environment setup, Whisper API integration
* 🔹 Medium Priority: Basic UI layout design
* **Sprint 2 (Day 2 - Core Features & Testing)**
* ✅ High Priority: Audio input handling, transcription processing, testing
* 🔹 Medium Priority: UI improvements, text output display and editing

# Phase-5: Project Development

## Objective:

## Implement the core functionalities of the audio transcription system using OpenAI Whisper while addressing development challenges efficiently.

## Key Points:

1. **Technology Stack Used:**
   * **Frontend:**  Gradio Python library
   * **Backend:** Google colab
   * **Programming Language:** Python
2. **Development Process:**
   * **Data Collection :** Audio files
   * **Data Preprocessing:** whisper OpenAI’s model
   * **Feature Engineering:** speech (Audio ) recognition and convert to text
3. **Challenges & Fixes:**
   * **Challenge:** noisy audio, latency, and multi-language accuracy
   * **Fix:** Use AI models that continuously learn from new data and trends.

# Phase-6: Functional & Performance Testing

## Objective:

Ensure the audio transcription system functions correctly, performs efficiently, and delivers accurate results through rigorous testing.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Category** | **Test Scenario** | **Expected Outcome** | **Status** | **Tester** |
| TC-001 | Functional Testing | Upload an audio file for transcription | Audio file successfully uploaded | Pending | Tester A |
| TC-002 | Functional Testing | Real-time transcription using Whisper API | Accurate text output displayed | Pending | Tester A |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TC-003 | Functional Testing | Support for multiple languages | Transcription in selected language | Pending | Tester B |
| TC-004 | Performance Testing | Process long-duration audio files efficiently | Minimal delay,  accurate output | Pending | |  | | --- | |  |  |  | | --- | | Tester C | |
| TC-005 | Performance Testing | Handle noisy background in audio input | Noise reduction, clear transcription. | Pending | |  | | --- | |  |  |  | | --- | | Tester B | |
| TC-006 | |  | | --- | |  |  |  | | --- | | Functional  Testing | | |  | | --- | |  |  |  | | --- | | Export transcription  in different formats  (txt, doc) | | File exported successfully | Pending | |  | | --- | |  |  |  | | --- | | Tester A | |

# Final Submission

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