Hackathon Project Phases Template that ensures students can complete it efficiently while covering all six phases. The template is structured to capture essential information without being time-consuming.

Hackathon Project Phases Template

Project Title: Audio transcription app using Open AI Whisper

Team Name: BUG BOUNTY

Team Members:

- ALLI CHIRANJEEVI
- YENKATHALA SHIVA PRASAD
- DAVATH VENKATESHAM

Phase-1: Brainstorming & Ideation

Objective:

- Identify the need for an accurate and efficient speech-to-text transcription system.
- Explore the capabilities of OpenAI Whisper for transcription.

Key Points:

1. **Problem Statement**:
    - Many individuals and businesses struggle with converting spoken words into text accurately and efficiently. Existing solutions often lack support for multiple languages or have high costs.

2. **Proposed Solution**:

    - Our project utilizes OpenAI's Whisper model to transcribe speech into text with high accuracy, supporting multiple languages and real-time processing.

3. **Target Users:**

    - Students, journalists, researchers, professionals, and individuals needing an efficient transcription tool.

4. **Expected Outcome:**

    - A functional application that can take audio input, transcribe it into text, and allow users to save or export the transcriptions as PDFs.

Phase-2: Requirement Analysis

Objective:

- Identify and document the technical and functional requirements for implementing speech-to-text transcription.
- Assess potential challenges and limitations.

Key Points:

1. **Technical Requirements:**

    - Programming Language: Python
    - Frameworks: OpenAI Whisper, PyPDF2 (for PDF handling), Flask/Django (for web integration)
    - Libraries: NumPy, Torch, ffmpeg, OpenAI API

2. **Functional Requirements:**

    - Upload audio files or record live audio
    - Process and transcribe speech to text
    - Support for multiple languages

- Export transcriptions as PDF

3. **Constraints & Challenges:**

   - Large model size requiring optimization
   - Handling noisy audio inputs effectively
   - Ensuring quick processing time

Phase-3: Project Design

Objective:

- Define the architecture and workflow of the system.

- Design an intuitive and user-friendly interface for transcription and PDF export.

Key Points:

1. System Architecture Diagram:
   - User uploads/records audio → Whisper model processes speech → Text output generated → Option to edit/save as PDF
2. User Flow:
   - Upload/record → Process → Edit (optional) → Save/export
3. UI/UX Considerations:
   - Simple and intuitive interface with drag-and-drop functionality for audio files.

Phase-4: Project Planning (Agile Methodologies)

Objective:

- Develop an organized workflow for efficient project execution.

- Assign roles and responsibilities using Agile methodologies.

Key Points:

1. **Sprint Planning:**
    - Sprint 1: Setting up Whisper model and basic transcription
    - Sprint 2: Implementing PDF export functionality
    - Sprint 3: UI development and testing
    - Sprint 4: Final testing and deployment
2. **Task Allocation:**
    - Member 1: Backend development (Whisper integration)
    - Member 2: Frontend development (UI/UX)
    - Member 3: Testing and debugging
3. **Timeline & Milestones:**
    - Week 1: Setup and initial development
    - Week 2: Feature implementation
    - Week 3: Testing and improvements
    - Week 4: Final deployment

Phase-5: Project Development

Objective:

- Implement the speech-to-text transcription system and integrate PDF export functionality.
- Ensure smooth interaction between frontend and backend components.

Key Points:

1. **Technology Stack Used:**
    - Python, OpenAI Whisper, Flask/Django, PyPDF2

2. **Development Process:**
    - Set up Whisper API → Implement audio input processing → Transcription → PDF export → UI Integration
3. **Challenges & Fixes:**
    - Model speed optimization → Used batching and pre-processing techniques
    - Background noise handling → Applied noise reduction filters

---

Phase-6: Functional & Performance Testing

Objective:

- Validate the accuracy, speed, and reliability of the transcription system.
- Identify and fix bugs for improved performance.

Key Points:

1. **Test Cases Executed:**
    - Short vs. long audio files
    - Different accents and languages
    - Noisy vs. clear audio
    - PDF export verification
2. **Bug Fixes & Improvements:**
    - Addressed delays in processing
    - Improved UI responsiveness
3. **Final Validation:**
    - The system meets initial requirements and performs well.
4. **Deployment:**
    - Hosted on a cloud-based platform (e.g., AWS, Heroku) or local server for demonstration.

Final Submission

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