ByteSize Sage AI Hackathon - Master the Attention Economy with Gen AI

Theme: AI + Multimodal Processing + Content Intelligence

Problem Statement:

Mentors, educators, and creators produce hours of high-value, long-form video content (lectures, podcasts and workshops). However, modern audiences consume information in 60-second bursts. Valuable "golden nuggets" of wisdom are often buried in 60-minute videos making them inaccessible to the average viewer.

Common challenges include:

- Sifting through hours of footage to find "viral" or high-impact moments manually.
- Converting horizontal (16:9) video to vertical (9:16) while keeping the speaker centered. (Optional)
- Manually transcribing and timing "karaoke-style" captions to keep viewers engaged. (Optional)

To solve this, we aim to build PulsePoint AI, a platform where users upload long-form videos and the app uses GenAI and Multimodal models to automatically:

- **Identify "Emotional Peaks":** Detect high-energy or profound moments using audio spikes and sentiment analysis.
- **Smart-Crop to Vertical:** Use computer vision to track the speaker's face and ensure they stay in the frame for TikTok/Reels. (Optional)
- **Generate Dynamic Captions:** Create high-contrast, timed overlays and catchy "hook" headlines to stop the scroll. (Optional)

This tool will turn a single session into a week's worth of viral marketing content, making high-level education snackable and shareable.

Submission Requirement:

- Users must submit a link to their public GitHub repository on the Unstop portal.
- **Mandatory:** Include a screen recording video of your live working project in the README of your GitHub repository.

Resources for Help (Optional for reference):

1. Core Engine: Multimodal & Gen AI APIs

You will need an API that can "understand" video or text transcripts.

- **Google Gemini 1.5 Flash:**
  - **Why:** It has a massive context window (1M+ tokens), meaning you can upload an entire long video or its transcript for analysis. Excellent at finding "the best parts."
  - **Get Started:** Google AI Studio
- **OpenAI Whisper:**
  - **Why:** The industry standard for converting video audio into text with high-accuracy timestamps.
  - **Docs:** OpenAI Speech-to-Text

2. Video & Audio Processing

Since this involves video editing, these Python libraries are your best friends:

- **MoviePy:** For cutting, cropping, and exporting video clips.
  - MoviePy Documentation
- **Librosa:** For analyzing audio "loudness" to find where the mentor is speaking most passionately.
  - Librosa Guide
- **MediaPipe:** For "Face Detection" to help with smart-cropping to vertical.
  - MediaPipe Solutions

3. Backend: Python-Based API

You need a server to handle the video processing and communicate with the AI.

- **FastAPI:** High performance and very easy to document.
  - FastAPI First Steps
- **Flask:** Simple and great for quick hackathon prototypes.
  - Flask Hello World

4. Frontend & Styling

- **Pico.css:** Drop this link into your HTML for instant and beautiful styling without writing a single line of CSS.
  - Pico.css Website

- **Streamlit:** If you want to skip HTML/CSS entirely, use Streamlit to build a video-processing web app entirely in Python.
    - Streamlit Docs

Sample Input / Output of your Web Tool:

You need to build a web application which takes a video as input - either google drive link or upload capability. On clicking a button, Output 3-5 reels should be generated. You need to test your tool using below mentioned Input Video Link -

Input Video Link:

Input video for ByteSize Hackathon.mp4

Expected Output:

Links or Downloadable files of 3-5 reels generated from Input video.