Take-Home Assignment: Smart B-Roll Inserter for UGC Videos

Overview

You are asked to build a system that automatically plans how B-roll clips should be inserted into an A-roll (talking-head / UGC) video.

This assignment is designed to test:

- Your research ability
- Your system design thinking
- Your ability to integrate multiple tools and APIs

Problem Statement

Given:

- One **A-roll video** (a person speaking to the camera)
- Multiple **B-roll video clips** (product shots, lifestyle shots, screen recordings, etc.)

Your task is to:

1. Analyze the A-roll video and understand *what is being said and when*
2. Analyze the B-roll clips and understand *what each clip represents*
3. Automatically decide **where and which B-roll clips should be inserted** into the A-roll video
4. Output a **structured timeline plan** that describes these insertions

Optionally, you may also generate the final stitched video.

Expected Outcome

Required Output

Your system must output a **timeline plan** in JSON format describing:

- At what timestamp B-roll should be inserted
- For how long
- Which B-roll clip is used
- Why that clip was chosen for that moment

Example (illustrative only):

```
{
  "insertions": [
    {
      "start_sec": 12.5,
      "duration_sec": 2.0,
      "broll_id": "broll_03",
      "confidence": 0.78,
      "reason": "Speaker mentions pouring coffee and texture"
    }
  ]
}
```

Optional Output (High Signal)

- A final rendered video using ffmpeg that overlays B-roll visuals while keeping A-roll audio intact

Technical Requirements

You shall use the following technologies:

- **Python**
- **ffmpeg**
- **React**
- **LLM API** (OpenAI or any comparable model) - reimbursable

You are free to choose specific libraries and services, but you must justify your choices.

Functional Requirements - Hints

1. A-Roll Understanding

- Extract the transcript of the A-roll video
- Transcript must include timestamps (sentence-level minimum)
- The transcript should be usable to reason about *what is being said at a given time*

2. B-Roll Understanding

For each B-roll clip, your system must create a text representation that describes its content.

This may come from:

- metadata or
- Automatically generated descriptions using vision or LLM techniques

3. Matching Logic

Your system must decide:

- Which moments in the A-roll are suitable for B-roll
- Which B-roll clip best matches each selected moment

Your logic must go beyond naive random insertion. At minimum, it should:

- Use semantic matching (e.g. embeddings)
- Avoid inserting B-roll too frequently
- Avoid inserting B-roll during critical speaking moments
- Prefer moments where visuals add value to what is being said

4. Timeline Planning

Generate a structured plan containing:

- A-roll duration
- Transcript segments
- B-roll insertions with timestamps and durations
- A brief explanation for each insertion

5. Frontend UI

Build a simple React interface that allows:

- Uploading A-roll and multiple B-roll clips
- Triggering plan generation
- Viewing the transcript with timestamps
- Viewing the proposed B-roll insertions

6. Video Rendering (Optional)

If implemented:

- Use ffmpeg to overlay B-roll visuals onto the A-roll
- A-roll audio must remain uninterrupted
- Focus on correctness over visual polish

Constraints

To keep the scope manageable:

- A-roll length: 30–90 seconds
- Number of B-roll clips: 6
- B-roll insertions: 3-6

What We Care About Most

This assignment is **not** about perfection.

We are evaluating:

- Your reasoning and trade-offs

- Your system design clarity
- Your ability to connect language understanding with media timelines

---

Deliverables

1. GitHub repository
2. README with:
3. Setup instructions
4. How to run backend and frontend
5. Required environment variables
6. Output artifacts:
7. Timeline plan JSON
8. Optional final rendered video

Time Expectations

This assignment is designed to take less than 2 days

Good luck, and we look forward to reviewing your approach.