SceneSense AI

PBS + Solution (Team Share Version)

1. Quick overview

SceneSense AI is a lightweight tool that turns a screenplay scene (plain text) into a structured “cinematic intent” summary. It extracts: (1) emotion, (2) narrative purpose, (3) visual mood, (4) suggested camera style, and (5) a confidence score. The goal is to help directors and teams make faster, clearer pre-production decisions.

2. PBS — Problem Breakdown Structure

2.1 What problem are we solving?

In early film production (pre-production), teams repeatedly discuss tone, mood, and shot ideas based on unstructured script text. This causes slow planning, inconsistent interpretation, and avoidable rework.

2.2 Core pain points (broken down)

- **Unstructured input:** Scripts are plain text. Key creative intent is implicit, not labeled.
- **Slow manual breakdown:** Directors and teams manually infer mood, lighting, and framing scene by scene.
- **Inconsistent interpretation:** Different team members read the same scene differently (writer vs director vs cinematographer).
- **No structured output for tools:** Storyboarding, shot-listing, and previz tools work better with structured fields.
- **Higher risk of miscommunication:** Misalignment leads to reshoots, delays, and budget overruns—especially for smaller teams.

2.3 Who experiences this problem?

- **Directors:** Need fast clarity on emotion/mood and shot direction.
- **Cinematographers:** Need lighting + framing cues aligned with story intent.
- **Storyboard artists / previz teams:** Need structured guidance to sketch shots quickly.
- **Film students & indie creators:** Often lack time/resources for deep breakdowns.

2.4 Why this matters (impact)

- Cuts pre-production discussion time for scene breakdowns.
- Reduces creative misalignment across roles.
- Gives teams a consistent baseline for planning (storyboards, shot lists, lighting).
- Enables quicker iteration: rewrite scene → re-run → compare intent.

3. Solution

3.1 What we built (MVP)

We built a Streamlit web app (runs locally) where a user pastes a screenplay scene and clicks Analyze. The app calls an LLM (via Groq) and returns a strict JSON object describing the scene’s cinematic intent.

3.2 Input → Output (simple)

Input: A screenplay scene (plain text)

Output: Valid JSON with these fields:

- **emotion** — primary emotional tone (e.g., tension, anxiety, romance)
- **narrative_purpose** — why this scene exists in the story (setup, reveal, conflict, etc.)
- **visual_mood** — lighting/atmosphere guidance (dark, warm, neon, foggy, etc.)
- **camera_style** — suggested framing/movement (close-ups, handheld, push-in, wide establishing)
- **confidence** — score (0–1) representing how confident the model is

Example output JSON

```
{
  "emotion": "tension",
  "narrative_purpose": "build anticipation or foreshadowing",
  "visual_mood": "dark, foreboding, isolated; rain and metallic echoes",
  "camera_style": "static close-ups on micro-reactions; slow push-in to heighten dread",
  "confidence": 0.80
}
```

3.3 How it works (one clear flow)

- **Step 1:** User selects an example scene (optional) or pastes their own scene.
- **Step 2:** App validates input length to avoid weak analysis.
- **Step 3:** App builds a structured prompt that forces strict JSON output.
- **Step 4:** Groq LLM generates the cinematic intent JSON.
- **Step 5:** App parses JSON and displays it + confidence label (High/Medium/Low).

3.4 Why this is strong for a hackathon demo

- Instant and visual: paste scene → click → structured result.
- Clear differentiation across genres (romance vs action vs tension).
- Runs locally: reliable for offline hackathon environments.
- Structured JSON output: easy to integrate later with storyboarding / shot-list tools.

3.5 Scope (MVP vs next)

MVP (now): single-scene analysis to cinematic intent JSON.

Next (post-hackathon): multi-scene upload, batch analysis, shot-list templates, export to PDF/CSV, and integrations with previz/storyboard tools.