

# SceneSense AI

Product Requirements Document (PRD)

Version: 1.0 | Status: Hackathon MVP + Next-Step Roadmap

**One-line summary:** Convert unstructured screenplay scenes into structured cinematic intent (emotion, narrative purpose, visual mood, camera style) with a confidence score to accelerate pre-production planning.

## 1. Background and context

Film and content teams spend significant time translating script text into visual decisions. This translation is often manual and inconsistent across roles (director, DoP, storyboard artist). SceneSense AI provides a fast, structured first-pass breakdown that improves alignment and speeds up planning.

## 2. Problem statement

Screenplay scenes are written as unstructured text. Creative intent (tone, mood, pacing, visual language) is implicit, leading to slow breakdowns, repeated discussions, and misalignment across team members.

### 2.1 Key pain points

- **Unstructured scenes:** Intent is not captured as explicit fields (emotion, mood, framing).
- **Slow pre-production:** Teams manually infer scene purpose and visuals for each scene.
- **Inconsistent interpretation:** Different stakeholders interpret the same scene differently.
- **Hard to reuse decisions:** No standardized output that can plug into shot lists or storyboards.
- **Costly misalignment:** Poor alignment increases delays and rework.

## 3. Goals and non-goals

### 3.1 Goals

- Generate consistent, structured cinematic intent from raw scene text.
- Help teams reach creative alignment faster during pre-production.
- Deliver a demo-ready tool that is reliable in offline hackathon settings.
- Ensure outputs are machine-readable (strict JSON) for future integrations.

### 3.2 Non-goals (MVP)

- Automatically generating full storyboards or a finalized shot list without human review.
- Replacing directors/cinematographers—this product provides guidance, not final decisions.
- Guaranteeing perfect creative understanding for every possible script style.

## 4. Users and primary use cases

SceneSense AI is designed for creators and film production teams who need fast, repeatable scene analysis.

User	Primary need	What they get
Director	Quick understanding of tone and purpose	Emotion + narrative purpose + camera direction baseline
Cinematographer (DoP)	Lighting and mood alignment	Visual mood + suggested framing/movement
Storyboard / Previz Artist	Speed up scene visualization	Structured cues that can map to shots
Film Student / Indie Creator	Learn + plan faster with limited resources	Instant breakdown + confidence indicator

## 4.1 Example usage scenario

A director pastes a tense confrontation scene. SceneSense AI returns a structured intent summary showing high tension, a foreshadowing purpose, dark visual mood, and close-up framing. The team uses this output as a starting point for a storyboard discussion and aligns faster.

## 5. Product scope (MVP)

- Scene input:** Text area to paste screenplay scene.
- Example scenes:** Dropdown to load pre-built sample scenes (tension/romance/action).
- Analyze action:** Button that runs the LLM-based intent extraction.
- Structured output:** Strict JSON fields (emotion, narrative purpose, visual mood, camera style, confidence).
- Confidence label:** High/Medium/Low badge derived from score.
- Validation and errors:** Empty/short scene checks; non-JSON fallback display.

## 6. Functional requirements

ID	Requirement	Priority
FR1	User can paste or type a screenplay scene into the input box	P0
FR2	User can load a sample scene from a dropdown	P1
FR3	User can trigger analysis using a single Analyze button	P0
FR4	System returns structured output in strict JSON format with required keys	P0
FR5	System displays JSON output clearly in the UI	P0
FR6	System computes confidence label (High/Medium/Low) and displays it	P1
FR7	System validates empty or too-short scenes and shows warnings	P0
FR8	System handles non-JSON model output gracefully (debug view)	P0

## 7. Non-functional requirements

- **Performance:** typical response within a few seconds for short scenes.
- **Reliability:** stable JSON outputs with low temperature inference.
- **Usability:** minimal steps; clear success and error messaging.
- **Portability:** runs locally on Windows; simple setup with requirements.txt.
- **Security:** API key stored in .env; not committed to version control.

## 8. UX requirements

- User can complete a full flow in under 30 seconds: paste → analyze → read output.
- JSON output is readable and copyable.
- Success message confirms completion; warnings are actionable.
- Confidence is visually obvious (High/Medium/Low).

## 9. Success metrics

- JSON parse success rate  $\geq 95\%$  during repeated demo runs.
- End-to-end demo completion under 30 seconds.
- Outputs vary logically across genres (romance vs action vs tension).
- No crash across 10 consecutive runs on a laptop.

## 10. Risks and mitigations

Risk	What can happen	Mitigation
WiFi / API latency	Slow response or demo interruption	Carry screenshots + pre-generated output; optional short recording
Non-JSON model output	Parsing fails and UI cannot render JSON	Strict prompt + fallback view for raw response
Weak input scene	Short scene produces generic output	Minimum length validation + user guidance
Scalability questions	Judges ask about production readiness	Roadmap: batch upload, caching, export, integrations

## 11. Roadmap (post-hackathon)

- **Multi-scene upload:** analyze an entire script or multiple scenes in one run.
- **Shot-list templates:** convert intent output into suggested shot patterns.
- **Export:** PDF/CSV export for sharing with production teams.

- **Role-based modes:** Director Mode vs Writer Mode outputs.
- **Integrations:** connect to storyboard/previz tools (future).