LocalSentinel.ai — VS Code Demo Plan (shareable one-pager)

What we're shipping (MVP)

- **VS Code extension** that runs a **full repo scan on command** (no real-time yet).
- **Static tools first → AI second**: rules catch common issues; AI **triages, explains, and proposes fixes**.
- **Output:** a clean **R/Y/G report** inside VS Code with:
  - **Plain-English repo summary** (what this project does, for new eyes).
  - **Findings** (file + line, why it matters).
  - **Proposed solutions** split into **separate, copy-ready prompts** per issue category.

Model choice (local, reliable, code-savvy)

- **Primary: Code Llama 7B Instruct** (quantized, e.g., q4 via `llama.cpp`)
  - Best balance of **code understanding** vs **16 GB RAM** constraint on Windows ARM64.
- **Fallbacks (if needed):**
  - **Llama-3.1 8B Instruct** (quantized) if it loads comfortably.
  - **Phi-3 Mini (3.8B)** for extra-low RAM headroom (weaker, but snappy).
- **Role:** explain code, **triage rule hits**, write **plain-English fixes** and **repo overview**.
  (No autonomous code-gen; we don't let it write large patches by itself.)

Pipeline (end-to-end)

1. **User clicks “Scan repo”** in VS Code.
2. **Static sweep (fast, local):**
    - **Semgrep** (multi-lang vulns/patterns)
    - **Bandit** (Python security checks)
    - **Light regex checks** (obvious keys, debug flags, permissive CORS, etc.)
3. **AI pass (Code Llama 7B):**
    - **Rank severity (R/Y/G)** and de-noise false positives.
    - **Explain** each issue in plain English.
    - **Propose fixes** as **short, separate prompts** per issue (ready to paste to ChatGPT/Claude CLI if we want auto-patching later).
    - **Summarize the repo** (what it is, main components, data flows) for new stakeholders.
4. **Report render in VS Code:** score, sections, jump-to-code, copy buttons for each fix.

Report structure (inside VS Code)

- **Header:** Project name, **overall score (0–100)**, scan time, “local-only” badge.
- **“New to this repo?”** — 3-5 sentence **plain-English overview**.
- **Findings (grouped):**
    - 🟥 **Critical** (backdoors, secrets, auth bypass)
    - 🟨 **Warning** (unsafe defaults, weak crypto, outdated deps)

  - 🟩 **Info** (hygiene, nits)
    Each finding shows:
  - **File:line** + **why it matters** (1–2 sentences)
  - **Snippet** (few lines)
  - **Proposed solution(s): separate, copy-ready prompts** (one per fix).
- **Next steps:** short checklist (apply fixes, re-scan, optional PDF export later).

Why this wins the demo

- **Works offline** on the Snapdragon X laptop (privacy, “edge” box checked).
- **Clear value fast:** catches at least one **planted backdoor** + offers **immediate, understandable fixes**.
- **Developer-friendly:** zero config, runs where devs live (VS Code), crisp report for PMs/partners.

Scope trims (to go fast)

- **No real-time linting** (scan-on-demand only).
- **No dynamic sandboxing** (static + AI only).
- **Optional** OSV/dependency CVE check (include if time allows).

Roles & 24-hour build plan

T0–T6 (Engine first):

- Wire **Semgrep/Bandit** → JSON.
- Run **Code Llama 7B** locally; prompt templates for **triage/explain/fix**.
- CLI prints **Markdown** report.

T6–T12 (VS Code shell):

- Button/command: **“LocalSentinel: Scan Repo”**.
- Spawn engine, capture Markdown → **webview**.
- Click-to-open file/line.

T12–T18 (Polish):

- Severity icons, score, copy buttons per fix.
- Add **repo overview** section (AI).
- Seed **demo repo** with a backdoor; tune prompts for crisp output.

T18–T24 (Harden & rehearse):

- Error states, basic settings (include/exclude folders).
- README (install, run, offline note).
- Rehearse **backdoor find** → **fix** flow.

Hand-off checklist (what each person does)

- **AI/Model:** get **Code Llama 7B q4** running; finalize 3 prompts: **triage**, **explain**, **fix**.
- **Rules:** minimal **Semgrep** pack + **Bandit** config; add 6–8 **regex checks** (secrets, debug, `exec`, permissive CORS).
- **Ext/UI:** VS Code command, webview, Markdown theming, **jump-to-code**.

- **Demo:** prep **vuln repo** + script: scan → show 🟥 → apply fix (manual or one-click) → re-scan.
- **Docs:** README (MIT, offline, how to run), quick GIF/screenshot.

Stretch (only if time remains)

- **One-click patch** for the demo backdoor (apply minimal diff).
- **CVE/deps** quick check (OSV) with offline cache.
- **Export HTML/PDF** of report.

Sound-bite for judges / teammates

> “LocalSentinel is a one-click **offline** code audit in VS Code. It runs **static checks**, then a local **code-savvy model** summarizes the repo, **flags real risks**, and gives **copy-ready fixes** per issue. In 60 seconds, we find a hidden backdoor, explain it, and show how to fix it — without any code leaving the laptop.”