NEUROCRAM — The Exam Intelligence Console

Rewrite the science of last-minute studying.

Concept

Exams are stressful, chaotic, and often unpredictable. Students don't fail because they don't study, they fail because they study *blindly*.

Your task is to create **NeuroCram**, a single-screen "Exam Intelligence Console" that analyzes a student's exam week and *auto-orchestrates* their study priorities, focus windows, weak spots, and revision strategies.

The interface should feel like a mission-control dashboard, blending exam pressure, performance analytics, and calm clarity. Make it visually stunning but functionally simple.

Problem Statement

Build a high-impact, interactive Exam Intelligence Console that takes in a set of exam details (subjects, remaining days, difficulty, student's self-rated confidence, etc) and outputs a smart, optimized, dynamic revision plan.

The system should:

- Present all insights on one immersive screen (like a command center).
- Include a bold hero section + CTA ("Generate Plan").
- Display 2–3 intelligent secondary modules (e.g., Stress Level Prediction, Productivity Zones, Question Pattern Emulator and any other you'd like to).
- Provide micro-animations & without clutter.

This is NOT a regular study planner. It should feel like an AI system preparing someone for a mission-critical exam week.

Requirements & Constraints

Category | Details

Core Concept: Build a single-screen exam command console with: a hero, one CTA, and 2–3 intelligent modules showing exam insights. Inputs can be mock data or user input.
Visual Polish: Choose at least two animated or reactive backgrounds. You can add lottie animations and include high-impact transitions and UI elements
Code Quality: Component-driven structure (React/Vue), clean file architecture, reusable

modules. Tailwind or Styled Components recommended.
Innovation: Add unique intelligence layers like a "CramHeat Map" showing subjects sorted by urgency, or "BrainEnergy Gauge" predicting burnout windows.
Clarity: Despite aesthetic richness, the interface must feel *calm*, minimal, and sharply prioritized.

Deliverables

1. Single-screen UI hosted on GitHub.
2. A 5–7 minute demo explaining:
    - Design thought process
    - Component structure
    - Animation implementation
    - How the "intelligence modules" work

Submission Reminders

- Include a public GitHub repo link.
- Include a Loom video walkthrough.

Tips

- Prioritize clarity over noise. Mission-control interfaces should feel powerful, not messy.
- Use GPU-friendly animations.
- Keep typography bold and concise.
- Add Easter eggs if you want (e.g. "Exam Survival Probability").