Image Prompts:

Image Prompts:

Front Shot:

I want the angle of the Pagani huyuara .

Rear Shot:

I want the Zonda R's Engine to be exposed, give me an image of that

Hood lifted Engine Exposed, back angle with scissor doors open take a reference from upper image and generate for blue car exposed image i want

Video Prompt:

Video Prompt:

A photorealistic, continuous cinematic shot of a matte carbon fiber Pagani Zonda R in a high-contrast dark studio. The camera begins at a low, aggressive front-center angle, highlighting the gold wheels and aerodynamic canards. Without cutting, the camera performs a slow, sensual, sweeping orbit around the vehicle's flank. It transitions smoothly from the front three-quarter view, gliding past the side profile to showcase the glossy carbon fiber weave and the intricate side details, eventually arriving at a dramatic low rear angle, framing the massive wing and central exhaust pipes. Once established in this rear position, the rear clamshell cover lifts away vertically out of frame. Immediately following this, the exposed AMG V12 engine undergoes a slow-motion 'technical exploded view' animation. The cylinder heads, exhaust manifolds, and suspension components slowly float outwards from the center block with rigid mechanical precision. There is zero morphing or warping; every part remains solid and textured as they expand to reveal the internal engineering. The lighting remains dramatic and moody, emphasizing the metallic glint of the floating engine parts against the dark background

Tab 3

Website One Shot:

One-Shot Prompt: Build "Pagani Zonda R" Scrollytelling Showcase

Role: You are a Senior Creative Frontend Engineer specializing in Awwwards-winning "Scrollytelling" websites.
Objective: Build a complete, production-ready, single-page luxury car showcase for the **Pagani Zonda R**. The core mechanic is a scroll-controlled image sequence (video-like) that acts as the background, with a "HUD-style" text overlay that transitions strictly based on the scroll progress.
Tech Stack:

- **Framework**: Next.js 14+ (App Router, TypeScript).
- **Styling**: Tailwind CSS v4 (using `@theme` variables).
- **Animation**: Framer Motion (for all entry/exit animations).
- **Core Logic**: HTML5 Canvas + `framer-motion` `useScroll` for the image sequence.

Assets Provided:

- I have a folder of **240 frames** (images) of a rotating car in `public/images/zonda-sequence/` named
- 1.jpg to
- 240.jpg.

---

1. Project Setup & Configuration

System:

- Initialize a Next.js app.
- Install: `npm install framer-motion clsx tailwind-merge`.
- **Fonts**: Use `Orbitron` (headings) and `Rajdhani` (body/UI) from Google Fonts via `next/font/google`.

Styling (
app/globals.css):

- Use Tailwind v4 syntax.
- Define these colors:
  - `pagani-black`: `#1a1a1a`
  - `pagani-gold`: `#D4AF37`
  - `bright-gold`: `#FFD700`
  - `carbon-gray`: `#2a2a2a`
- Global styles: Background `#1a1a1a`, Text `white`, Selection `pagani-gold`.

2. Architecture & Logic (Critical)

We need a **Master Scroll Architecture** to prevent desync.

- app/page.tsx: Owns the `useScroll` hook attached to a `600vh` tall container. It passes the `scrollYProgress` MotionValue down to children.
- ZondaScrollCanvas: A "dumb" component that simply renders the correct frame (1-240) based on the prop `scrollYProgress`. Must use `window.devicePixelRatio` to ensure 4K sharpness (no grainy canvas).
- ZondaExperience: A "dumb" HUD overlay that transitions content (Hero -> Design -> Engine) based on the same `scrollYProgress`.

3. Component Specifications

A.

components/ZondaScrollCanvas.tsx

- **Props**: `{ scrollYProgress: MotionValue<number>, totalFrames: number, imageFolderPath: string }`.
- **Logic**:
  - Transform `scrollYProgress` (0 to 1) -> `frameIndex` (0 to 239).
  - Use a `<canvas>` element.
  - In `useEffect`, load all images first.
  - Draw the image using `ctx.drawImage` with `object-fit: contain` logic.
  - **CRITICAL**: Scale the context by `devicePixelRatio` to support Retina/4K displays.

B.

components/ZondaExperience.tsx (The HUD)

- **Props**: `{ scrollYProgress: MotionValue<number> }`.
- **Layout**: Absolute positioning `inset-0`, `pointer-events-none` (so user scrolls the page, not the div).
- **Phases**:
  1. **0% - 33% (Hero)**: Show "Pagani Zonda R", Price (€1.5M), "Inquire Now".
  2. **33% - 66% (Design)**: Fade out Hero, Fade in "Design" title + "Carbon Fiber Monocoque" text.
  3. **66% - 100% (Engine)**: Fade out Design, Fade in "Engine" title + Right-aligned Specs (V12, 750HP).
- **UI**: Premium sci-fi look. Thin borders, uppercase tracking, gold accents.

C.

components/Navbar.tsx

- Fixed at top (`z-50`).
- Logo left, "INQUIRE" button right.

- Glassmorphism background on scroll.

D.

app/page.tsx (Orchestrator)

- Structure:
- <main className="bg-pagani-black">
- <Navbar />
- {/* SCROLL SEQUENCE (Locked for 600vh) */}
- <section ref={containerRef} className="h-[600vh] relative">
-
- <ZondaScrollCanvas z-index-0 />
- <ZondaExperience z-index-10 />
-
- </section>
- {/* REST OF SITE (Scrolls naturally after sequence) */}
-
- <SpecsGrid />
- <Features />
- <Footer />
-
- </main>

4. Implementation Steps

Please generate the code for these exact files in this order:

1. app/layout.tsx (Fonts)
2. app/globals.css (Tailwind Theme)
3. data/carData.ts (The text content)
4. components/Navbar.tsx
5. components/ZondaScrollCanvas.tsx (**Remember High-DPI fix**)
6. components/ZondaExperience.tsx (**Remember Scroll Sync**)
7. app/page.tsx (**The Orchestrator**)

Tone: Premium, Technical, Italian Luxury. **Constraint**: Do not allow the user to scroll past the car sequence until it finishes playing (enforced by the sticky 600vh container).