

# Technical Architecture: The Digital Observer (Full-Stack)

## 1. Core Technology Stack

### Frontend: The Interactive Layer

- **Next.js (App Router):** Foundational framework providing **Static Site Generation (SSG)** for the portfolio and **On-Demand Revalidation**. This keeps the 3D scene fast while allowing text/video updates to go live without a redeploy.
- **React Three Fiber (R3F):** Manages the 3D scene graph. Tapes and text are treated as reactive components that update when the backend data changes.
- **Drei:** Specifically for `useVideoTexture` (TV playback) and `Text` (rendering admin-edited titles as 3D physical meshes).
- **GSAP (GreenSock):** Handles mechanical timelines (cassette sliding) and the draggable physics of the 2D resume papers.

### Styling: The Hybrid Approach

- **Tailwind CSS:** For the Admin Dashboard, HUDs, and general UI layout.
- **Vanilla CSS / CSS Modules:** For precise "bespoke" styling of the Resume papers and the scroll-based **"Curtain Lift"** logic. Handles the `position: fixed` and `z-index` layering required for the parallax effects.

### Backend: Logic & Persistence

- **Node.js (Next.js API Routes):** Manages project updates, text configuration, and view statistics.
- **MongoDB:** Stores three core collections: `projects` (cassettes), `siteConfig` (UI text), and `interactionStats`.
- **Mongoose:** ODM for schema enforcement.
- **NextAuth.js:** Secures the admin portal.

## 2. System Design: Content-Driven Engine

### A. UI Text & Copy Management ( `siteConfig` )

Every string in the user interface is fetched from MongoDB. This allows you to edit the "voice" of the portfolio from the dashboard:

- **Section Titles:** `heroTitle` , `vcrSectionTitle` , `workbenchTitle` , `floorSectionTitle` .
- **Subtitles & Prompts:** `scrollPromptText` , `vcrInstructionText` , `emptyTvMessage` .
- **Resume Content:** `resumeJSON` containing all job titles, dates, and descriptions for the Workbench section.

## B. Project & Video Management ( `projects` )

The dashboard provides full CRUD for the VCR tapes:

- **Metadata:** Edit project names and GitHub/Live links.
- **Video Preview:** Swap the `.mp4` video that plays on the 3D TV when a tape is inserted.

## 3. Layout & Scroll Mechanics (Fixed & Overlap)

The portfolio utilizes a non-traditional scroll architecture to enhance the immersive feel.

- **Section 1 (Hero) - Fixed:** The Hero section is `position: fixed` at the top of the stack. It remains stationary while Section 2 scrolls up and overlaps it, effectively hiding the eyeball as the user "enters" the workshop.
- **Middle Sections (Projects/Resume) - Sliding:** These sections exist in the normal document flow but sit at a higher `z-index` . They slide over the Hero and eventually "lift" to reveal the final section.
- **Section 4 (Contact) - Fixed Background:** The Contact section is `position: fixed` at the very bottom of the stack ( `z-index: 1` ).
- **The "Curtain Lift":** As the user reaches the end of the Resume section, it uses a large `margin-bottom: 100vh` to physically slide up and out of the viewport, revealing the stationary Workshop Floor underneath.

## 4. Section-by-Section Visual & Content Breakdown

### Section 1: The Panopticon (Hero)

- **Visual Look:** A vast, high-contrast black void containing a single, hyper-realistic procedural eyeball.
- **Layout State: Fixed.** Does not scroll; gets covered by Section 2.
- **Contains:** Hero Title, Subtitle, and Scroll Prompt.

### Section 2: The VCR Station (Projects)

- **Visual Look:** Transition to a wooden desk with a retro TV and VCR.
- **Layout State: Overlapping.** Slides up over Section 1.

- **Contains:** Cassette stack, Interactive TV, and Launch Button.

### Section 3: The Workbench (Resume)

- **Visual Look:** Top-down view of the wooden desk with scattered A4 papers.
- **Layout State: Overlapping.** Connects seamlessly to Section 2.
- **Contains:** Draggable Resume papers (Profile, Experience, Skills).

### Section 4: The Workshop Floor (Contact)

- **Visual Look:** A high-detail concrete floor with scattered "evidence" items.
- **Layout State: Fixed.** Revealed only when Section 3 "lifts" away.
- **Contains:** Contact items (Polaroid, Envelope, PCB, Sticky Note) and Inspect Mode logic.

## 5. Component Architecture

`DynamicText.jsx`

- A reusable component that bridges the database and the UI.
- In 3D sections, it utilizes Drei's `<Text />` to ensure your admin-edited titles are physically part of the 3D world (lighting, shadows).

`VCRStation.jsx` **(Section 2)**

- `TV.jsx` : Maps the `videoUrl` from the inserted cassette to a `THREE.VideoTexture`.
- `Cassette.jsx` : Dynamically renders the label text based on the `projects` collection in MongoDB.

## 6. Tactile Feedback & State Responses

### A. The "VCR Clunk" (Mechanical Feedback)

- **Insertion:** When a tape reaches the sensor zone, its movement changes from "User Drag" to a scripted GSAP "Suck-in" animation.
- **LED Status:** The VCR mesh features a small procedural LED. It glows **Orange** during the "Suck-in" phase and turns **Solid Green** only when the video buffer is ready.

### B. TV "Tracking" Static (Loading State)

- **State: Buffering:** When a tape is inserted but the video hasn't loaded, the noise becomes "aggressive"—heavy horizontal lines and flickering "Tracking" text.

### C. Inspection Logic (Section 4)

- **Pick-up:** Items follow a curved path to the camera with a subtle "wobble" (spring physics).

## 7. Analytics & Interaction Stats

- Tracks specific user behavior: `TAPE_INSERTED` , `LAUNCH_CLICKED` , and `ITEM_INSPECTED` .
- Displays a summary of project popularity and contact clicks in the Admin Dashboard.

## 8. Production & Performance

- **On-Demand Revalidation:** Saving changes in the Admin Dashboard makes updates live instantly.
- **Conditional Render:** 3D loops are paused when a section is out of view to maintain 60FPS.