Portfolio Internal Architecture Documentation (Version 2026/1.0)

Overview

This document serves as an exhaustive reference for the internal architecture, behavior, state management, and administrative constraints of the AdityaPatil-2026 personal portfolio project built on React + Vite + Firebase.

System Architecture

1. The Core Render Cycle

The application relies heavily on conditional rendering via React's useState hooks. Instead of a multi-page routing framework (like react-router-dom), the application utilizes a Single Page Application (SPA) construct.

- **activeTab State:** Dictates which widget or CategoryView mounts dynamically in the central <main> frame of App.jsx.
- **AnimatePresence:** The framer-motion package wraps these mounting instances, granting an exit animation out of the DOM prior to the next module rendering in. This avoids jarring layout shifts during navigation.

2. Component Structure

- App.jsx: The monolithic controller. It houses universal UI components (Navigation, Brand Logo, Theme Toggle, Contact Dock, Flying Banner) and injects the isAdmin boolean drill-down prop.
- TechnicalArsenal.jsx: Renders the left-column skills mapping.
- ProfessionalJourney.jsx: Renders the bottom map logic over time periods.
- CategoryView.jsx: A heavily modular generic component that expects an overarching ThemeMap color styling and an ID property. This renders individual domain projects inside specific dynamic tabs (i.e. "Software Engineering", "AI").
- AddCategoryModal.jsx: Administrator-only modal allowing custom Domain headers to be pushed onto the navigation bar dynamically.

Global State & Persistence (Firebase)

The application previously managed dynamic changes via browser localStorage. To ensure persistence across all devices, it now leverages **Google Firebase Firestore**.

When an admin mutates data on the web app, those alterations invoke setDoc() commands targeted directly at the adityapatil-2026 Firebase project. Any initial render fires an asynchronous useEffect hook utilizing getDoc() to pre-fill the React states.

Database Keys Schema Reference:

- **portfolio/arsenal**: Technical Arsenal categories and skill arrays.
- **portfolio/journey**: Professional timeline data.
- **portfolio/custom_categories**: Dynamic Navigation domains manually injected.
- **portfolio/resume_link**: The dynamic string linking to the user's downloadable Resume PDF.

- **portfolio/category_{STORAGE_KEY}**: Contains the complex arrays of custom project dictionaries built inside any given domain view.

Administrator Capabilities

Admin capabilities are restricted solely to users whose email successfully parses against the VITE_ADMIN_EMAILS environment variable. The JWT is procured via @react-oauth/google.

Authorized Functions Include:

1. **Curriculum Vitae Alteration:** Clicking the **Edit3** pencil icon modifies the customResume string pointing directly to a newly hosted drive link.
2. **Category Creation:** Appending a new Domain (with an associated color scheme) to the top navigation header via the generic AddCategoryModal.
3. **Data Modification/Deletion:** Direct inline mutation of specific project cards, chronological journey blocks, or nested arrays of "skills" within the technical arsenal.

Failure to match the verified environmental email variables throws an automatic "Unauthorized Google Account" blockage and aborts the .then() chain.

Design Patterns & UI Theming

The project adheres strictily to "Glassmorphism" inside a Light/Dark matter palette.

Dark/Light Engine:
A custom ThemeToggle mechanism taps into window.matchMedia('(prefers-color-scheme: dark)') to assess the baseline OS preference. It toggles a literal dark CSS class directly onto the global window.document.documentElement. TailwindCSS manages all visual shifts utilizing .dark:[util] classes globally.

CSS Grid Framework:
The primary content container utilizes a robust grid-cols-1 md:grid-cols-12 wrapper. This enables intricate desktop layouts (for example: Arsenal taking 3 columns, Hero taking 5, Dock taking 4) while effortlessly collapsing everything down to single-stack columns on mobile viewports.

Generated for Internal Documentation Management (2026).