We will focus on the **Logic and Security** of the code itself. Here is the documentation and prototype structure for the **Penuel Unified Prototype**, built from scratch with a "Security-First" developer mindset.

---

1. Local Security Architecture (The "Safe Lab")

Even without a VPS, we must code as if the system is live to prevent "Systemic Security Risks."

- **Environment Shadowing:** We will use a .env.local file. This is a local text file that holds your "Mock" API keys. React is configured to never include this file in a public build, so your secrets stay on your machine.
- **Zero-Trust Components:** We will write "Validation Logic" inside every React component. If a user tries to enter a negative amount in the **Car Wash** package selector, the code will reject it before it even tries to process a payment.
- **Sanitized Data Binding:** We will strictly use React's {curly brace} syntax for rendering. This ensures that even if we "mock" data from a text file, it can never execute malicious scripts.

---

2. Project Logic Structure

We are building a **Multi-Tenant Prototype**. This means one codebase that "switches" its personality based on which branch you are testing.

Folder Hierarchy

Plaintext

```
/penuel-unified-prototype
├── /public          # Static assets (Logos for Plaza & Stopover)
├── /src
│   ├── /components     # Reusable UI (Buttons, Cards, Nav)
│   ├── /context       # The "Empire" Brain (Handles the Business Switcher)
│   ├── /data          # Local JSON files (Replacing the live database for now)
│   │   ├── retail.json     # Data from your retail.txt
│   │   └── services.json   # Data from your services.txt
│   ├── App.js         # The Root Controller
│   └── index.js       # The Entry Point
├── .env.local         # WHERE SECRETS HIDE (Local only)
```

```
└── package.json       # Project Manifest
```

---

3. The "Business Switcher" Logic (The Prototype Core)

This is how we handle the **Plaza vs. Stopover** logic without needing a complex backend.

Logic Layer	Purpose	Security Benefit
Theme Toggler	A simple JS function that changes the CSS variables (Gold vs. Green).	No hardcoded styles; easy to audit.
Data Filter	If activeBusiness === 'Stopover', the system only loads data from services.json.	Prevents "Data Leaking" between business units.
Mock Gateway	A function that simulates an M-Pesa STK push. It "waits" for 3 seconds then returns a "Success" code.	Allows us to build the "Success/Fail" UI screens without spending a cent on transaction fees.
