**Frontend-First Incremental Roadmap (Web)**

**Principles:**

1. **Small, incremental steps** — each AI task builds one visible piece of the UI.
2. **Immediate feedback** — every step produces a functional view with dummy data.
3. **Dummy data first** — real backend comes later. Replaceable without breaking the frontend.
4. **Shared logic for future reuse** — prepare components so mobile can reuse them.
5. **Permissions / roles considered** — UI elements reflect access, even with dummy data.

**Step 0: Project Bootstrapping**

* Initialize Next.js + TypeScript project.
* Install Tailwind CSS, React Context API, Axios, Socket.IO client.
* Create a **minimal folder structure**:
* /components
* /pages
* /context
* /theme
* /data
* /utils
* Make sure dummy data and its integration can be removed easily later.

**Step 1: Shared Theme & Global Context**

* Create **design tokens** (colors, spacing, fonts).
* Configure Tailwind CSS to use them.
* Create a **UserContext** with dummy user + roles/permissions.
* Test dummy data in a small “Preview” component to confirm UI changes with different roles.

**Step 2: Minimal Layout + Navigation**

* Implement **MainLayout** component (header/footer/sidebar).
* Add **Next.js routing** with placeholder pages (Dashboard, Profile, Content).
* Make sure navigation **updates the visible page** using dummy content.

**Step 3: Core Shared Components (Dummy Data)**

* Build a small set of components:
  + Buttons (role-aware)
  + Input fields
  + Cards/List items
  + Modals / Alerts
* Wrap logic in **shared component files**. Styling only in .web.tsx.
* Use dummy data to display **functional UI**.

**Step 4: Basic Pages with Dummy Data**

* Render Dashboard, Profile, Content pages using components from Step 3.
* Display lists, cards, modals — **fully interactive** with dummy data.
* Verify permissions: buttons/modules hidden for unauthorized roles.

**Step 5: Real-Time Simulation (Dummy Events)**

* Add Socket.IO client listening to dummy events.
* Update lists/cards/UI automatically when dummy events occur.
* Ensure UI responds correctly **without backend yet**.

**Step 6: Iterative Review**

* Test pages, components, real-time updates.
* Adjust UI, functionality, or component design.
* Fix any issues **before moving to backend or mobile**.

**Step 7: Prepare for Backend Integration**

* Ensure dummy data is **isolated** (easy removal).
* Confirm all API calls are type-safe and structured for backend.
* Keep shared logic reusable for mobile.