PII Guardian Frontend

The PII Guardian frontend is a marketing and demonstration site built with **Next.js (App Router)** and **TypeScript**. It serves as a public “shop window” for the PII Guardian API, providing a live redaction demo, feature highlights, and links to coverage and compliance docs. The app is fully containerized and designed for easy deployment and rebranding.

---

1) Technology Stack

Category	Technology	Version*	Purpose
Framework	Next.js (App Router)	*see `package.json`	Server + client components, routing, metadata
Language	TypeScript	*see `package.json`	Static typing
Styling	Tailwind CSS v4	*see `package.json`	Utility-first styling; Typography plugin for legal/comp
Animations	Framer Motion	*see `package.json`	Page and section animations
Icons	Lucide React	*see `package.json`	SVG icons
Packaging	Docker	—	Reproducible builds & deploys

* Use the exact versions from `package.json` for due diligence.

---

2) Project Structure

```
frontend/
├─ public/
│  ├─ coverage/                    # HTML coverage report (entry: index.html)
│  └─ compliance/                  # Compliance PDFs/HTML (served directly)
├─ src/
│  ├─ app/
│  │  ├─ layout.tsx               # Root layout (fonts, backgrounds, metadata)
│  │  ├─ Template.tsx             # Framer Motion page transitions
│  │  ├─ page.tsx                 # Home (server component) -> renders client
│  │  ├─ HomePageClient.tsx       # Interactive demo + “Deep Dive” buttons
│  │  ├─ globals.css              # Tailwind base + global utilities
│  │  ├─ compliance/
│  │  │  └─ page.tsx              # In-site Compliance Docs Portal (/compliance)
│  │  └─ legal/
│  │     ├─ layout.tsx            # Prose layout
│  │     ├─ privacy/page.tsx      # Privacy (includes telemetry disclaimer)
│  │     ├─ security/page.tsx
│  │     └─ terms/page.tsx
│  └─ components/
│     └─ animations/
│        ├─ FadeSection.tsx
│        └─ StickyNavWrapper.tsx
├─ Dockerfile
├─ next.config.ts
├─ tailwind.config.ts
└─ tsconfig.json
```

---

3) Core Components & Logic

3.1 Rendering Model

- `src/app/page.tsx` is a **Server Component** that sets metadata and renders the interactive client.
- `src/app/HomePageClient.tsx` is a **Client Component** (`"use client"`), responsible for state, user input, and API communication.

3.2 Live Demo API Interaction

- **Endpoints:**
  - `GET /v1/scan/rule-packs` (populate rule pack options)
  - `POST /v1/scan/redact` (perform redaction)
- **Config:** `NEXT_PUBLIC_API_BASE_URL` defines the backend base URL.
- **Auth:** `X-API-Key` header uses `NEXT_PUBLIC_DEMO_API_KEY`.
- **Rule Packs:** The UI lets users pick one or more packs from a curated list; the default is **`DEMO_PACK_STRICT`** (not `"DEMO_PACK"`).
- **Errors:** Non-2xx responses are parsed (if JSON) and shown in the UI; aborts are handled gracefully.

3.3 UX & Animations

- **Page transitions:** `Template.tsx` uses Framer Motion for subtle route transitions.
- **Sticky header:** `StickyNavWrapper.tsx` applies a blurred, semi-transparent header on scroll.
- **Section reveals:** `FadeSection.tsx` fades content into view using Intersection Observer + motion variants.

---

4) Configuration (Environment)

These are read at build/runtime (container or host env):

Env Var	Purpose	Used In
`NEXT_PUBLIC_API_BASE_URL`	Backend base URL	`HomePageClient.tsx`, Dockerfile
`NEXT_PUBLIC_DEMO_API_KEY`	Demo API key for public UI	`HomePageClient.tsx`
`NEXT_PUBLIC_APP_NAME`	Branding (header/footer)	`HomePageClient.tsx`
`NEXT_PUBLIC_CONTACT_EMAIL`	Footer + Privacy contact mailto	`HomePageClient.tsx`, `privacy/p`
`NEXT_PUBLIC_COVERAGE_URL`	Link to coverage report entry	`HomePageClient.tsx` (defaults to
`NEXT_PUBLIC_COMPLIANCE_PACK_URL`	Optional direct link to a key PDF	`HomePageClient.tsx` trust badge
`NEXT_PUBLIC_COMPLIANCE_PORTAL_URL`	Compliance Portal link	`HomePageClient.tsx` (defaults to

> Note: Variables are injected by Next.js at build time (public-prefixed). There is **no `docker-compose.yml` requirement**; the **`Dockerfile`** accepts build args or you can set envs at runtime.

---

5) Styling

- **Tailwind v4** powers the UI; `globals.css` includes base directives, a few global utilities (e.g., `.btn-cta`), and scrollbar tweaks.

- **Typography plugin** is applied to legal/compliance sections for readable prose.
- **Fonts:** Inter via `next/font`.
- No bespoke design system; classes are co-located in components for ease of resale/rebranding.

---

6) Deep Dive & Compliance

- **Pytests Coverage Report:** The "Deep Dive" button links to `NEXT_PUBLIC_COVERAGE_URL` (default **`/coverage/index.html`**).
- **Compliance Docs Portal:** The second button opens **`/compliance`**, a portal that auto-lists **`public/compliance/*.pdf`** and **`.html`** documents as cards (served directly, typically `noindex` via headers).
- **Acquire link:** Third button points to your Acquire.com listing.

---

7) Privacy & Telemetry

The Privacy page explicitly documents telemetry:

- We collect **usage metadata only** (tenant/API key, endpoint, timestamp, status, latency, request size).
- **We do not collect request content** (no text/files/PII stored).
- Users can opt out by self-hosting or disabling telemetry.
- Contact email is driven by `NEXT_PUBLIC_CONTACT_EMAIL`.

---

8) Build & Deployment

Local Dev

```
npm install
npm run dev
# http://localhost:3000
```

Production (Docker)

- **Multi-stage `Dockerfile`:** builder compiles Next.js; runner serves the app (`next start`).
- **Env injection:** set `NEXT_PUBLIC_*` at build (as build args) or runtime (as env).
- **Example:**

```
docker build -t pii-guardian-frontend .
docker run -p 3000:3000 \
  -e NEXT_PUBLIC_API_BASE_URL="http://localhost:8000" \
  -e NEXT_PUBLIC_DEMO_API_KEY="demo-key" \
  pii-guardian-frontend
```

---

9) Roadmap Ideas (optional for buyers)

- **Admin UI** for tenants, API keys, and custom rules.
- **Interactive API explorer** for additional endpoints.
- **Component extraction** to smaller units if the app grows (e.g., `HeroSection`, `DemoBox`, `FeatureGrid`).