Frontend One-Page (React + Vite)

Stack

1. React
2. Vite
3. React Router
4. Axios

Main folder: `/projectRoot/frontend`

Structure

Key directories/files:

1. `src/pages` for main screens (home, auth, cases, evidence, board, judiciary, rewards, payments).
2. `src/components` for shared UI (layout, protected route, reusable parts).
3. `src/context/AuthContext` for auth state and user role context.
4. `src/api/client` for Axios instance and API communication.
5. `src/App.jsx` for route definitions.
6. 'src/context/ThemeProvider' provides Theme(light/dark) for the entire app.

Main UI Concepts

1. English UI as requested.
2. Modular dashboard: modules shown based on user roles/permissions.
3. Page-level workflow UI for each project section:

- cases and complaints
- crime scene reports
- evidence
- detective board
- interrogation/trial
- rewards

- payments

Routing

Public routes:

1. `/`
2. `/login`
3. `/register`

Protected routes:

1. `/dashboard`
2. `/cases`
3. `/evidence`
4. `/board`
5. `/reports`
6. `/judiciary`
7. `/rewards`
8. `/payments`
9. `/admin-rbac`

API Integration

1. Frontend communicates with Django APIs under `/api/...` .
2. JWT token is used for authorized requests.
3. User-facing error messages are extracted and shown from backend responses.

Build and Run

Dev mode:

```
cd /projectRoot/frontend
npm install
npm run dev -- --host
```

Production build:

```
npm run build
```

With Docker:

```
cd /projectRoot
docker compose up
```

Notes

1. UI behavior is tightly tied to permissions from backend.
2. Multi-role users see combined capabilities.
3. Current setup is optimized for dev speed and workflow testing.