CRM Enterprise App Summary

Compiled from repository files only (no external sources)

What It Is

CRM Enterprise is a monorepo with an Angular client and an ASP.NET Core backend for CRM workflows, with additional supply-chain modules surfaced in the UI.
The client supports a mock API development mode, while the server includes JWT auth, Swagger, health checks, tenant middleware, and a SignalR presence hub.

Who It Is For

Primary users appear to be sales reps/managers and CRM administrators/ops (inferred from routes, permissions, and user-story/docs filenames).
Explicit persona statement: Not found in repo.

What It Does

- Dashboard plus CRM records: customers, contacts, leads, opportunities, and activities.
- Lead workflows include pipeline, import, create/edit, and convert routes.
- Decision Inbox includes inbox, approvals, AI reviews, policies/SLA, and history views.
- Role/permission-gated navigation and route access (auth + role guards).
- Settings for users, roles, permissions, security levels, workspace/tenant, workflows, notifications, and audit log.
- Public auth and onboarding flows: landing, login, change password, accept invite, supplier onboarding.
- Supply-chain pack navigation includes RFQ, suppliers, catalog/pricing, procurement, logistics, inventory, quality, and analytics.

How It Works (Repo-Evidence Architecture)

- Angular app boots with router + PrimeNG + HTTP interceptors (`app.config.ts`). Routes load a shell and feature pages; auth/role guards enforce permissions (`app.routes.ts`).
- Client data flow: browser -> Angular HTTP client -> interceptors. If `environment.useMockApi` is enabled, `/api` calls are served by `mock-api.interceptor.ts` + `mock-db.ts`; otherwise requests target the API (proxy config points `/api` to `http://localhost:5014`).
- API data flow: ASP.NET Core controllers -> Application + Infrastructure projects -> EF Core (`CrmDbContext`) -> SQL Server. Middleware/services include tenant resolution, JWT auth/authorization policies, Swagger (dev), health endpoints (`/health`, `/healthz`), and SignalR presence hub (`/api/hubs/presence`).

How To Run (Minimal)

- Frontend: install Node.js 20+ and npm 10+, then `cd client`, `npm install`, `npm start` (serves at `http://localhost:4200`).
- Backend: repo README says .NET 8, but API/Infrastructure projects currently target `net10.0`; use a compatible .NET SDK plus Docker for local SQL Server.
- Quick local API path from repo root: `./scripts/dev-api.sh` (starts `docker compose` SQL Server, waits for health, then runs `dotnet run` in `server/src/CRM.Enterprise.Api`).
- Alternative SQL-only step: `docker compose up -d sqlserver` (SQL Server exposed on `1433`); API config/connection string details are in server appsettings/README.

Repo Evidence Used

Files inspected to produce the summary on page 1

- /Users/yasserahmed/Desktop/Development Projects/CRM-Enterprise/README.md
- /Users/yasserahmed/Desktop/Development Projects/CRM-Enterprise/docker-compose.yml
- /Users/yasserahmed/Desktop/Development Projects/CRM-Enterprise/scripts/dev-api.sh
- /Users/yasserahmed/Desktop/Development Projects/CRM-Enterprise/client/package.json
- /Users/yasserahmed/Desktop/Development Projects/CRM-Enterprise/client/proxy.conf.json
- /Users/yasserahmed/Desktop/Development Projects/CRM-Enterprise/client/src/app/app.config.ts
- /Users/yasserahmed/Desktop/Development Projects/CRM-Enterprise/client/src/app/app.routes.ts
- /Users/yasserahmed/Desktop/Development Projects/CRM-Enterprise/client/src/app/layout/navigation/navigation.config.ts
- /Users/yasserahmed/Desktop/Development Projects/CRM-Enterprise/client/src/app/mocks/mock-api.interceptor.ts
- /Users/yasserahmed/Desktop/Development Projects/CRM-Enterprise/client/src/environments/environment.ts
- /Users/yasserahmed/Desktop/Development Projects/CRM-Enterprise/client/src/environments/environment.production.ts
- /Users/yasserahmed/Desktop/Development Projects/CRM-Enterprise/server/src/CRM.Enterprise.Api/Program.cs
- /Users/yasserahmed/Desktop/Development Projects/CRM-Enterprise/server/src/CRM.Enterprise.Api/CRM.Enterprise.Api.csproj
- /Users/yasserahmed/Desktop/Development Projects/CRM-Enterprise/server/src/CRM.Enterprise.Infrastructure/CRM.Enterprise.Infrastructure.csproj

Notes

- README states ASP.NET Core/.NET 8 backend, but inspected API and Infrastructure project files target net10.0.
- Poppler (`pdftoppm`) was not available in this environment, so visual PNG render QA could not be performed here.