Aptiva Login ? Frontend, Backend, Database Report

Frontend - Built landing/chat UI in webapp/index.html plus assets/app.js and app.css with hero, chat transcript, conversation dropdown, system switch, login controls, and lease drafts sidebar. - Chat UX streams responses with SSE, optimistic placeholders, markdown rendering, and inline cards. Preferences banner hidden from transcript, new chat button disables until backend responds. - Lease workflow UI lists drafts via /api/lease/drafts, allows refresh/download with auth headers, and emphasizes latest status. - Auth controls cover signup/login/guest/Google placeholder, memory sharing toggle, and forgot-password form hitting /api/auth/forgot-password. - Performance tweaks lazy-load conversation bodies, remove blocking overlays, and streamline CSS transitions.

Backend - FastAPI server/app.py exposes auth, chat, conversation, preference, and lease routes while serving static UI. - Chat endpoints stream LangGraph output, manage conversations, and enforce guardrails via get_current_user and sanitize/guard_rail_check. - Auth endpoints create hashed users, sessions, guest identities, logout, and forgot-password tokens stored in Supabase. - Lease endpoints generate PDFs through system1/lease_drafter.py, save base64 documents, and expose list/detail/download APIs. - Error handling includes retry wrapper around Supabase, SSE fallbacks, sanitized filenames, and PII-safe logging.

Database - Supabase/Postgres via storage/supabase_store.py with tables users, sessions, conversations, messages, preferences, lease_drafts, password_resets; RLS locks rows to user_id. - Preferences stored per user (not per chat) and rehydrated into new conversations; lease drafts filtered by user and conversation. - Performance improvements fetch only titles for sidebar, paginate-ready queries, and retries for transient errors.

Testing & Deployment 1. pip install -r requirements.txt 2. Configure .env with OpenAI, Apify, Supabase URL + service key 3. Run uvicorn server.app:app --host 0.0.0.0 --port 8000 4. Visit http://localhost:8000, exercise chat + lease flow 5. Push updates on branch database (git push origin database)