Chatbot Platform – Design & Architecture

1. Overview

The Chatbot Platform allows users to create projects, manage prompts, and chat with an AI bot. It consists of a React frontend and a FastAPI backend with JWT-based authentication.

2. Technology Stack

Layer	Technology
Frontend	React (Vite), TypeScript, Tailwind CSS
Backend	FastAPI, Python, SQLAlchemy, PostgreSQL
Auth	JWT, HTTP Bearer
API	REST API endpoints
Database	PostgreSQL
Deployment	Frontend: Vercel/Netlify, Backend: Uvicorn/FastAPI server

3. Frontend Design

3.1 Pages

- Login/Register: Authentication, JWT saved in localStorage.
- Projects: Lists user’s projects; select project to chat.
- Chat: Chat interface with user and bot messages, send/receive messages.

3.2 Components

- Navbar.tsx – Top navigation bar
- Sidebar.tsx – Sidebar menu (Projects, Settings, Logout)
- Button.tsx – Reusable button component
- Input.tsx – Reusable input field
- Chat.tsx – Chat container with messages, input field, and send button

3.3 Data Flow

1. User logs in → JWT stored in localStorage.
2. Projects fetched → user selects project → project ID stored in localStorage.
3. Messages sent via fetch POST → backend validates JWT → responds → messages rendered in chat.

3.4 Frontend Architecture Diagram

[React App]
  ├─ Pages: Login/Register, Projects, Chat

├─ Components: Navbar, Sidebar, Button, Input
└─ Services/API: Fetch calls to backend with JWT

4. Backend Design

4.1 Database Models

- User: id, name, email, password_hash
- Project: id, name, description, user_id
- Prompt: id, text, project_id

4.2 API Endpoints

Endpoint	Method	Purpose
/auth/register	POST	User registration
/auth/login	POST	User login, returns JWT
/projects	GET	Fetch user projects
/chat	POST	Send chat message to AI bot

4.3 Chat Flow

[User sends message] → [React Chat Component]
↓
[Fetch POST /chat] → [Backend FastAPI] → [JWT validation]
↓
[Forward to AI API] → [Response received] → [Return to frontend]
↓
[React updates chat messages]

5. Security

- JWT-based authentication for protected routes
- Passwords hashed with bcrypt (FastAPI backend)
- Authorization header checked on every request

6. Summary

This architecture ensures:
- Clear separation of concerns (frontend vs backend)
- Secure authentication
- Scalable design for multiple projects and chat interactions
- Easy maintainability using reusable React components