

# Ollama AI Chat Application - Project Summary & Instructions

## What We Built

We created a professional AI chat application that runs locally on your computer and connects to Ollama. This gives you:

- Complete Privacy: All AI processing happens on your computer
- No Internet Required: Works offline once set up
- Professional Interface: Beautiful blue-black corporate design
- Multiple AI Models: Switch between different models easily
- Real-time Chat: Instant responses from your local AI

## Technologies & Tools Used

Core Technologies:

- Next.js 15
- TypeScript
- Tailwind CSS
- React

UI Components:

- shadcn/ui
- Radix UI
- Lucide React

AI Integration:

- Ollama
- REST API

Development Tools:

- VSCode
- Node.js & npm
- PowerShell

## Project Architecture

ollama-chat-app/

■ ■ ■ ■ app/

■ ■ ■ ■ api/

■ ■ ■ ■ chat/route.ts

■ ■ ■ ■ health/route.ts

■ ■ ■ ■ models/route.ts

■ ■ ■ ■ page.tsx

■ ■ ■ ■ layout.tsx

■ ■ ■ ■ globals.css

■ ■ ■ ■ components/ui/

■ ■ ■ ■ button.tsx

■ ■ ■ ■ input.tsx

■ ■ ■ ■ card.tsx

■ ■ ■ ■ select.tsx

■ ■ ■ ■ scroll-area.tsx

■ ■ ■ ■ badge.tsx

■ ■ ■ ■ lib/

■ ■ ■ ■ utils.ts

■ ■ ■ ■ package.json

## How It Works

1. Frontend (What You See):

- Beautiful Interface
- Chat Messages in bubbles

- Model Selector dropdown
  - Status Indicator (Ollama connected or not)
  - Suggestion Cards for prompts
2. Backend API Routes:
- /api/chat: Sends messages to Ollama, gets response
  - /api/health: Checks if Ollama is running
  - /api/models: Lists available AI models
3. Ollama Integration:
- Runs on localhost:11434
  - Supports llama3:latest (4.7GB), gemma:2b (1.7GB)
  - All local, no internet needed

## **Step-by-Step Setup Process**

### Phase 1: Prerequisites

- ✓ Installed Ollama (version 0.9.1)
- ✓ Downloaded AI models (llama3:latest, gemma:2b)
- ✓ Started Ollama server (ollama serve)

### Phase 2: Project Creation

- ✓ Created Next.js project with TypeScript and Tailwind
- ✓ Installed dependencies (UI libraries, icons, utilities)
- ✓ Set up project structure (folders and files)

### Phase 3: Backend Development

- ✓ Created API routes for chat, health check, and models
- ✓ Connected to Ollama via HTTP requests
- ✓ Added error handling and logging

### Phase 4: Frontend Development

- ✓ Built chat interface with React