

# The Pragmatist: AI Chat Interface

## Objective

Create a full-stack AI chat interface that demonstrates your ability to work with modern web technologies and integrate with AI APIs.

## Important Notes

1. Cost: You should not need to pay anything to complete this task. OpenRouter provides free access to several AI models.
2. Flexibility: Beyond the required tech stack listed below, you are free to use any additional tools, libraries, or approaches you prefer.
3. Time: We expect this to take approximately 2-4 hours, but take the time you need to produce quality work.

## Frontend Requirements

1. React Router v7 SPA: Implement as a single-page application with client-side routing
2. UI Library: Mantine or ShadCN (your choice)

## Backend Requirements

1. A PostgreSQL Database
  - a. Define a simple schema with at minimum a chats table and a messages table
2. Drizzle ORM

# AI Integration

OpenRouter ([openrouter.ai](https://openrouter.ai))

- Create a free account and obtain an API key
- Use one of the free models available (e.g., meta-llama/llama-3.2-3b-instruct:free, google/gemma-2-9b-it:free, or similar)
- Check [openrouter.ai/models](https://openrouter.ai/models) and filter by "Free" for current options

## Core Functionality

1. Chat UI
  - a. A clean interface for displaying conversation history
  - b. Input field for composing messages
  - c. Visual distinction between user and AI messages
2. Message Exchange
  - a. Users can send messages and receive AI responses
  - b. Messages should be persisted to the database
3. API Implementation
  - a. Design the API as you see fit. We recommend considering HTTP streaming for AI responses and tRPC for type-safe communication, but you may use any approach you prefer.

## Deliverables

1. A GitHub repository containing your code
2. A README.md with setup instructions
3. Brief notes on any architectural decisions or trade-offs you made

## Questions?

If you have any questions or need clarification, please don't hesitate to reach out.