



Prompt:

Crea un proyecto backend en Next.js 14 (App Router) llamado "chatbot-next" con estas características:

1. Estructura de carpetas:

`/lib/db.ts` → conexión a PostgreSQL usando pg y DATABASE_URL de `.env.local`
`/lib/huggingface.ts` → servicio para llamar a la API de Hugging Face (modelo: `mistralai/Mistral-7B-Instruct-v0.2`) usando token de `.env.local`
`/app/api/chat/route.ts` → endpoint POST `/api/chat`

Posible Ejemplo de la estructura de la tabla donde almacenar el historial de conversaciones:

....

```
CREATE TABLE IF NOT EXISTS chat_history (  
  id SERIAL PRIMARY KEY,  
  user_message TEXT NOT NULL,  
  ai_response TEXT NOT NULL,  
  created_at TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP  
);
```

....

2. Funcionalidad del endpoint POST `/api/chat`:

- Recibe `{ message }` en JSON.
- Llama a Hugging Face con el modelo indicado.
- Devuelve `{ message, response }`.
- Guarda ambos en PostgreSQL en la tabla ``chat_history`` con timestamp.

3. Archivo `.env.local` de ejemplo:

```
DATABASE_URL=postgresql://postgres:postgres@localhost:5432/chatbot  
HUGGINGFACE_TOKEN=TOKEN
```

4. Ejemplo de uso del modelo via API HuggingFace:

```
....  
import { InferenceClient } from "@huggingface/inference";  
  
const client = new InferenceClient(process.env.HF_TOKEN);  
  
const chatCompletion = await client.chatCompletion({  
  provider: "featherless-ai",  
  model: "mistralai/Mistral-7B-Instruct-v0.2",  
  messages: [  
    {  
      role: "user",  
      content: "What is the capital of France?",  
    },  
  ],  
});  
  
console.log(chatCompletion.choices[0].message);  
....
```

5. Documentación en README.md:

- Explicación de la estructura.
- Cómo levantar PostgreSQL y el backend en Docker.
- Ejemplo de uso del endpoint con curl.
- Notas sobre la tabla chat_history.

6. Consideraciones:

- Comentarios explicativos en cada archivo.
- Backend únicamente, sin frontend ni UI.
- Docker para PostgreSQL y backend.
- Uso de TypeScript.

Genera todo el código listo para copiar y pegar, incluyendo README.md completo.

Uso:

Copialo y pegalo en firebase studio, lovable o alguna IA similar