**🤖 Part 3: AI Integration (Curiosity + Optional Code)**

**✍️ Short Essay:**

If I build a **chat feature with AI**, I would choose **OpenAI's GPT model** (like GPT-4). It’s accurate, powerful, and easy to use through an API. Many apps like ChatGPT use it, so it's tested and reliable.

**✅ Why I choose this:**

* It's good at understanding and replying clearly.
* It supports streaming responses (like real chat).
* Easy to connect with using the OpenAI or Vercel AI SDK.

**🛠️ How to add it in Next.js app:**

1. I will create an **API route** in /app/api/chat/route.ts.
2. I’ll use **Vercel AI SDK** (or direct OpenAI API) to send the user's message to the model.
3. Then return the model’s response back to the frontend.

**💬 Handling input/output:**

* I will take user input from a <textarea> or <input> box.
* When the user presses "Send", the message is sent to the backend API.
* The backend talks to OpenAI and sends back the answer.

**🔁 For streaming (like real-time typing):**

* I’ll use **Server-Sent Events (SSE)** or Vercel’s streaming features.
* As the AI types each word, it will appear one by one in the chat (just like ChatGPT).
* This gives a better experience and feels fast.

✅ In short:  
I’ll use **OpenAI + Next.js + Vercel AI SDK**.  
Frontend sends message → Backend sends to OpenAI → AI reply comes back (streamed or full).

### 🌱 Part 4: Developer Mindset (5–10 minutes)

**🗣️ Short Answers:**

1. **Most exciting tool/stack I explored:**  
   I recently explored **Vercel AI SDK with Next.js 15**. It was exciting to see how easily I could build an AI chat app using streaming and server functions.
2. **How I approach debugging:**  
   First, I **read the error message** carefully. Then I **use console.log()** to check values step-by-step. If I’m still stuck, I search online (Stack Overflow or docs) and try small changes to test what works.
3. **Project I’m proud of:**  
   <https://github.com/JRajKumar24/library-management-system>  
   It’s a project where I combined backend API routes and AI to build a smart chat feature.