

The most complex app I've ever built,
I built in 3.5 days.

This is how I LIFT



THE OLD WAY

I'm looking at [[X DOCS LINK]]. What does this mean? [QUOTE SOMETHING DIRECTLY]

Example 01

Senior Pair Programmer / React router versions explained

EA i'm following this intro to react router tutorial:
<https://reactrouter.com/tutorials/address-book>
what does this mean?
URL segments, layouts, and data are more often than not coupled (tripled?) together. We can see it in this app already:
URL SegmentComponentData/ <App> list of contacts
contacts/:contactId <Contact> individual contact
Because of this natural coupling, React Router has data conventions to get data into your route components easily.

Great question! Let me visualize this for you because it's a really important concept.

The "Triples" Relationship - Think of it like a restaurant:

Imagine each URL in your app is like a **different section of a restaurant**, and they naturally need three things together:

Restaurant Section (URL)	→	Staff Needed (Component)	→	Menu/Info They Need (Data)
-----------------------------	---	-----------------------------	---	-------------------------------

Here's the pattern visualized:

/ (lobby)	→ <App>	→ [List of all tables]
/table/5	→ <TableView>	→ [Info about table 5]
/table/5/order/3	→ <OrderView>	→ [Details of order 3]

In your address book app:

URL	Component	Data
/	→ <App>	→ All contacts (the list)
/contacts/123	→ <Contact>	→ Contact #123's details

Why this matters - The "Coupling" concept:

Think of these three things as **inseparable teammates**:

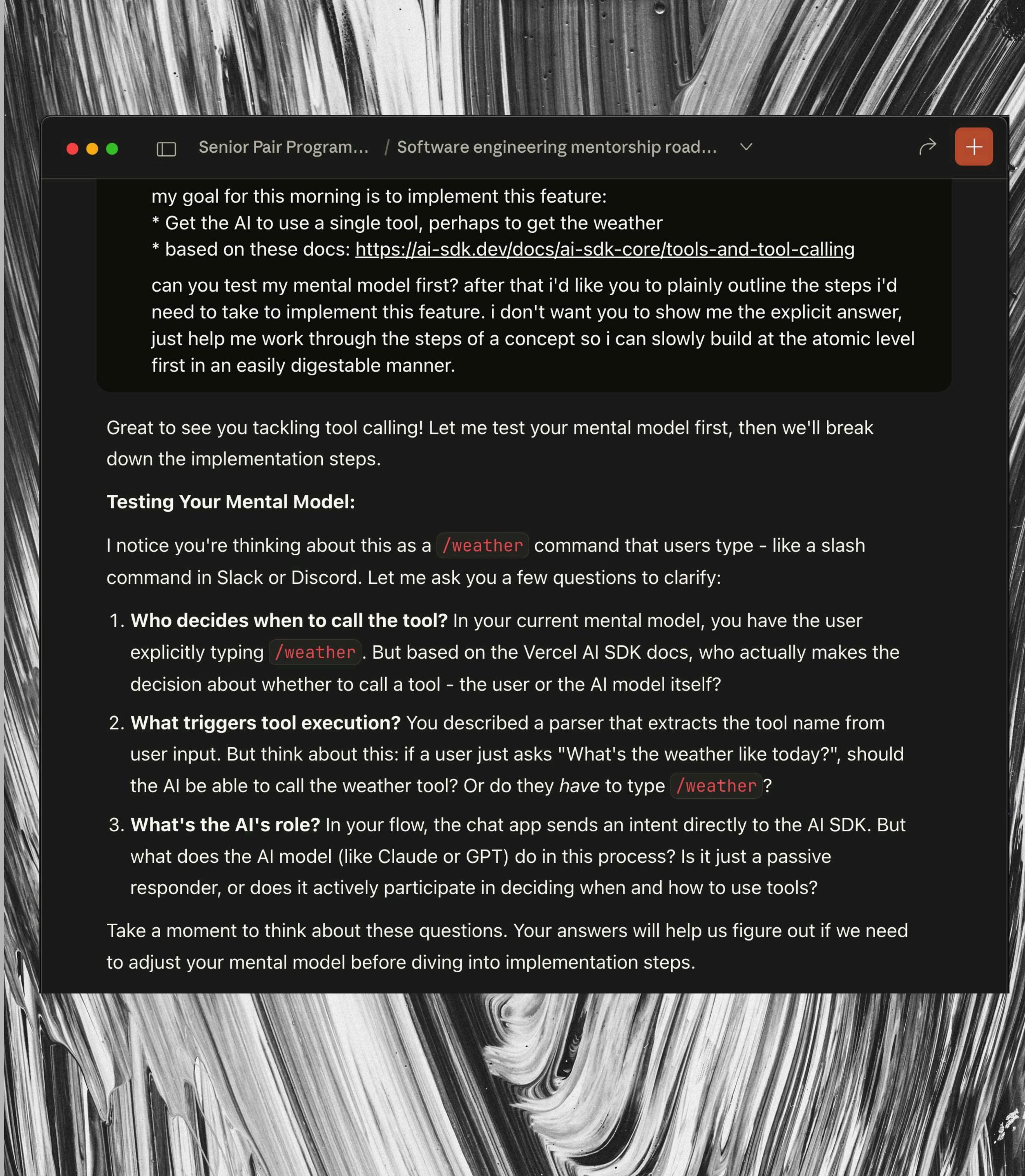
↙ They're connected!

URL: /contacts/5
↓
Component: <Contact> ← needs to show contact 5
↓
Data: { name: "Sarah", phone: "555-1234" }

THE SHIFT

I've read [X] docs [[LINK TO DOCS]], my goal is [X], before we implement, can you test my mental model?

Here's what I think is happening: [YOUR UNDERSTANDING]. Is this correct? If not, explain why—avoid code examples, focus on concepts.



THE OUTPUT

You're right about [X]and [Y], but you're treating [Z] as critical when it's just an example. The actual critical piece is [W].

Example 03

Senior Pair Programmer / React AI chat development journey

EA if my mental model is correct, we would receive the chat in `chat.tsx`, convert it in `chat.tsx` then pass it to `ai.ts`

[CONCEPT]

Good thinking, but let's trace through where the conversion actually needs to happen.

Think about it this way:

- `chat.tsx` uses `useChat()` which **creates and manages** UI messages
- `useChat()` automatically sends those messages to your server action
- Your `ai.ts` action **receives** those UI messages and needs to pass them to `streamText()`

The mismatch happens on the server side - you're receiving UI messages but `streamText()` needs Model messages.

Here's a key question: Can you change what format `useChat()` sends messages in?

No - `useChat()` is a library hook that always sends UI messages. That's its job.

So where does the conversion need to happen?

Look at your action in `ai.ts`:

```
typescript
const { messages } = await request.json() // UI messages arrive here
const result = streamText({
  model: google("gemini-2.5-flash"),
  messages // Wrong format being passed here!
})
```

The conversion needs to happen **between** receiving the messages and passing them to `streamText()`.

Does that make sense? The client (`chat.tsx`) doesn't know or care about AI model formats - that's the server's (`ai.ts`) responsibility.

THE RESULTS

With this approach, I implemented:



React Router
Supabase DB
Drizzle ORM



Better Auth
Vercel AI SDK

I DIDN'T JUST BUILD
AN APP—
I LEARNED HOW TO
LEARN 10X FASTER.

KNOWLEDGE GAINED
COMPOUNDS FOREVER.

