
One Cursor agent is a mistake

We build web apps with four. Here's why splitting the work beats one do-everything prompt.

Start in planning mode, not code

Before a single line gets written, one agent does nothing but plan. It maps the app, the data flow, and the edge cases. This alone cut our rework.

Agent 1: The Planner

- ▶ Its only job is architecture and sequencing. No code. It writes the spec the other agents follow, so nothing gets built blind.

Agent 2: Backend

- ▶ Owns data models, APIs, and logic. It reads the planner's spec instead of guessing. Cleaner structure, fewer surprises later.

Agent 3: Landing & Frontend

- ▶ Handles the interface and the landing experience only. Keeping it separate from backend means neither one drifts into the other's mess.

Agent 4: QA

- ▶ Reviews everything the others produce. A dedicated critic catches what the builder can't see. This is the agent most people skip — and it's the one that saves you.

Why separation actually works

One agent holding four jobs loses context and cuts corners. Narrow scope per agent means sharper output and easier debugging.

“

AI plans and builds fast, but you own the final call. We never ship a client feature without a human reading the QA agent's notes first.

Where it still needs you

Your turn

Are you still running one agent for the whole build, or have you split the roles? Curious what setup works for you — tell me in the comments.



Muhammad Yasir Azeem

The technical co-founder for non-technical founders

programmox.com