

Capstone Project Brief

A pre-filled example. Adapt it to your own build, then publish it.

Shola's Tech Notes · sholastechnotes.com

Example brief. Swap in your own project. Everything below is illustrative, written for a fictional career switcher moving from a customer support role into cloud and AI. Keep the four headings; replace the rest.

Project: Support Desk Assistant (a small multi-agent customer service helper, built as a capstone)

1. Problem statement

In my old support role, a large share of tickets were the same few questions (order status, returns policy, password resets) waiting in a queue for a person. I wanted to see whether a small team of AI agents could answer the routine ones with a source, and hand everything else to a human with the context already gathered.

- **Who it is for:** the support inbox of a small online shop (a made-up one, with test data).
- **What good looks like:** an illustrative target, not a result: routine questions answered with the help article cited, anything uncertain passed to a person with a summary.

2. Architecture

- **Router agent:** reads each message and decides: policy question, order lookup, or human.
- **Knowledge agent:** answers policy questions from a handful of help articles using retrieval, and always cites the article it used.
- **Order agent:** calls a mock order-status API (one serverless function over test data). Read-only access, nothing else.
- **Handoff agent:** when confidence is low or the customer is unhappy, writes a three-line summary for a person and stops.
- **Guardrails:** no refunds, no account changes, every routing decision logged so it can be reviewed.
- **Example stack:** a hosted LLM API, serverless functions, a small vector store, a basic web chat page, infrastructure as code.
- **Key trade-off:** four small agents instead of one big prompt, so each piece can be tested on its own. The cost is more moving parts to wire up and monitor.

3. What I learned

- **Routing is the hard part.** When an answer was wrong, it was usually because the router sent the message to the wrong agent. Test the router first.
- **A handoff is a feature.** Knowing when to stop and pass to a person matters more than answering everything.
- **You need a test set early.** A short list of sample questions with the route I expected made every change measurable.
- **Least privilege applies to agents too.** Giving the order agent read-only access removed a whole class of risk.
- **Writing it up exposed gaps.** Explaining each decision in plain English showed me which parts I had copied rather than understood.

4. Next steps

- **Evaluate on every change:** run the sample questions automatically before each deploy.
- **Track cost per conversation,** so I can explain the running cost in an interview.
- **Publish:** open-source the repo with a README and diagram, then a short LinkedIn post covering the problem, the architecture and one trade-off.

Make it yours: keep the four headings, swap the project, and cut anything you cannot explain out loud. Rough and public beats polished and private.

Reply to Tuesday's email with SHIP and a one-line summary of your brief. I will tell you the one thing I would tighten before you publish it.