Final Project — Proposal

One to two pages. Copy the headings below into your own document, answer each one, and submit it on Canvas as a PDF.

A proposal isn't a promise. It's your best guess, written down so that you, your pod and we can see what you're betting on, and where it might go wrong. Everything in it is allowed to change once you start talking to people. A proposal that is specific and turns out wrong is far more useful than one that is vague and can't.

1. Your project in one sentence

What it is, and who it's for.

> *Example:* "A tool that helps student coaches of the 5C club sports teams turn their notes from practice into a training plan for each player."

2. The problem, and who has it

- **Who** has this problem? Be specific: "first-year students who commute", not "students".
- **What do they do now** to deal with it, and what's hard or annoying about that?
- **How do you know?** Say whether it's your own experience, someone you've already talked to, or a guess you still need to check.

3. A first point-of-view statement

Fill in the blanks:

> *[user]* needs *[need]* because *[insight]*.

The need is a verb, not a feature ("needs to know which machines are free", not "needs an app"). The insight is the part that surprised you, or that you're least sure of.

Then write one or two "How might we …?" questions that open up the problem without naming a solution.

4. A better world, for whom?

Who is better off if this works, and how? One or two sentences on how it fits the College's vision of STEM for a Better World.

5. What the large language model (LLM) does

- What is the one thing your product needs that only a model can do?
- What goes into the model (the user's text, a photo, data from your API), and what comes back?
- If you took the model out, what would be left? If the answer is "most of the product", look for a better job for the model.
- What happens when the model is wrong? Who would notice, and what would it cost them?

6. The new application programming interface (API)

- Which API will you use, and what for?
- Does it need a key, does it have a free tier, and what are its rate limits?
- **Show that it works:** make one real request (with `curl`, or from a short script) and paste the first few lines of the response.
- If it falls through, what's your backup?

7. A sketch

A photo of a hand-drawn sketch of the main screen: what a person sees, and what they do there. Pencil on paper is perfect: rough is the point.

8. The people you'll talk to

Name at least three people who have the problem (or describe them, if you'd rather not use names), and how you'll reach each one. Write the three questions you most want to ask them.

9. Your stack

The course stack (FastAPI, SQLModel, Postgres, a plain HTML, CSS and JavaScript page, Docker on Railway), or something else? If something else, say what, and why.

10. Your biggest risk

What's the one thing most likely to sink this project: the problem isn't real, the API doesn't give you what you need, the model can't do the job well enough, or something else? How will you find out as early as possible?

Check yourself

Before you submit, check that your proposal says how you'll meet each requirement in the project brief:

- ☐ **It's for someone:** a real person or group other than you (sections 2 and 8)
- ☐ **A frontend:** a page a person uses for the main task (section 7)
- ☐ **A database:** what your app stores, and reads back
- ☐ **Publicly deployed:** where it will run (section 9)
- ☐ **It calls an LLM:** a job only the model can do (section 5)
- ☐ **It uses a new API:** one real response, pasted in (section 6)
- ☐ **Built with AI:** you'll build it with an AI coding agent, in a repo with a `CLAUDE.md`