

# Project Definition and Proposal Outline

Crash Course — Day 1 Afternoon

**Your Angle**

## > Choosing your angle

---

You cannot respond to everything in the RFP with equal depth. A strong proposal is **specific**.

Consider:

- > A specific **region** or country (e.g., coastal Bangladesh, the Sahel, Mozambique)
- > A specific **hazard type** (floods, drought, extreme heat, cyclones)
- > A specific **population** (children under 5, elderly, displaced communities)
- > A specific **infrastructure gap** (health facilities, emergency shelters, road access)

Your angle determines your research questions, your data sources, and the shape of your proposal.

## > What are the risks?

---

Every proposal has risks. The strong ones **name them**.

- > What data might **not exist** for your chosen region?
- > What if the data exists but is **too coarse** (national, not district-level)?
- > What if two data sources use **different geographic boundaries**?
- > What assumptions are you making about **data quality**?

A proposal that acknowledges risks is more credible than one that pretends there are none.

## > Exercise: develop your angle

---

### Individual work – 20 minutes

Write down **1-2 possible angles** for your proposal response:

1. Which **region** or country?
2. Which **hazard type**?
3. Which **population** or **infrastructure gap**?
4. In one sentence, what would your proposal argue?

You will use these to form teams after the exercise.

## > **Team formation**

---

Find **1-2 other students** whose angle is compatible with yours.

- > You don't need identical angles, just compatible ones
- > A team focusing on "flood risk in East Africa" can accommodate different infrastructure focuses
- > Teams of **2-3 people**

Once your team is formed, agree on a **shared angle** that combines your perspectives.

**Setting Up**

## > Your workspace

---

You need three things:

1. **A GitHub repository** for your team's project
2. **A Codespace** launched from that repository
3. **An agentic CLI tool** installed inside the Codespace

If you set this up during the lectures, it should take a few minutes. If not, follow along and ask for help.

## > Exercise: set up your environment

---

### Individual – 20 minutes

1. One team member creates the repository and invites the others as collaborators
2. Each team member launches a Codespace from the shared repository
3. Install your chosen CLI tool:

### Gemini CLI:

```
npm install -g @google/gemini-cli
```

**Claude Code or OpenAI Codex:** follow the installation instructions for your tool.

1. Authenticate and verify it works: ask it a simple question.

# Building Your First Map

## > The goal

---

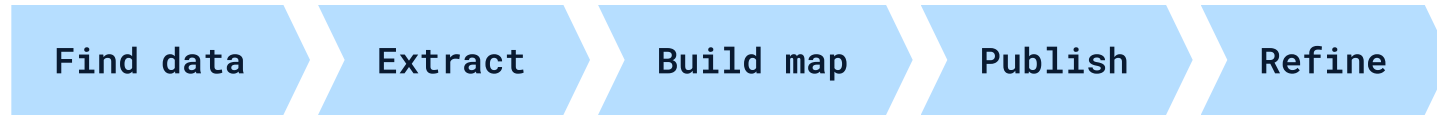
By the end of this afternoon, your team will have:

- > Used your AI tool to **find and retrieve** geospatial data relevant to your angle
- > Built a **Leaflet map** displaying that data
- > **Published it** on GitHub Pages
- > A live URL you can share

This is your first artifact. It will not be perfect. The point is to go through the full loop once.

## > The loop

---



Each cycle produces a **visible result** you can check and improve.

- > First pass: get something on the map
- > Second pass: improve the data or add a layer
- > Third pass: refine the presentation

The AI tool handles the technical work. You handle the **decisions**: what data, what to show, what the map should communicate.

## > Step 1: ask your AI tool to find data

---

Start by asking your agent what data exists for your angle.

"I'm working on a proposal about flood preparedness in [your region]. I need geospatial data I can put on a map. What open data sources exist? Compare them on coverage, format, and how to access them."

Review what the AI suggests. **You decide** which source to start with.

## > Step 2: extract and save

---

Ask your agent to write a script that retrieves the data and saves it to your repository.

"Write a Python script that fetches health facility locations in [your region] from [source] and saves the result as a GeoJSON file in data/. Add comments explaining each step. Do not run it yet."

**Read the script first.** Then ask the agent to run it.

A script is a **reproducible artifact**. If someone asks how you got this data, you show them the script.

## > Step 3: build a map

---

Ask your agent to create a Leaflet map from your data.

"Build a self-contained HTML page with a Leaflet map that displays the GeoJSON file in data/. The map should be centered on [your region] and show each point with a popup containing the name. Save it as index.html."

Open the HTML file in the Codespace preview to check it works.

## > Step 4: publish on GitHub Pages

---

Commit your files and enable GitHub Pages:

### 1. Commit and push:

```
git add .  
git commit -m "first map: [your data description]"  
git push
```

1. **Enable Pages:** Repository Settings → Pages → Source: main branch → Save

2. Your map is live at: [https://\[username\].github.io/\[repo-name\]/](https://[username].github.io/[repo-name]/)

## > Step 5: iterate

---

Your first map is a starting point. Now improve it:

- > **Add a layer:** ask the AI to find a second dataset (e.g., population density, flood zones, admin boundaries) and overlay it on the map
- > **Improve the popups:** add more fields from the data
- > **Refine the style:** colours, legend, zoom level
- > Each improvement: commit, push, check the live page

Every commit is a documented step in your process.

## > **Exercise: build and publish your map**

---

### **Team work — rest of the afternoon**

1. Use your AI tool to find geospatial data for your angle
2. Extract it with a documented script
3. Build a Leaflet map and publish it on GitHub Pages
4. Iterate: add layers, improve, push again

**Aim for at least two commits** showing progression.

When you're done (or when time is called), note down:

- > What data you found and what you couldn't find
- > What worked well with the AI tool and what didn't

**Wrap-Up**

## > Show your maps

---

Each team: **share your GitHub Pages URL** and tell us in 2 minutes:

1. What **data** did you put on the map?
2. What was the **hardest part**?
3. What would you **add next**?

## > What we produced today

---

### **Morning:**

- > A shared understanding of how to read an RFP strategically
- > The "show don't tell" insight: demonstrate capacity, don't just describe it

### **Afternoon:**

- > Teams with a defined angle
- > A working development environment
- > A published map with real data on GitHub Pages
- > The beginning of your artifact chain

## > Day 2 preview

---

Tomorrow we get serious about data collection:

- > **Research questions and horizon table** — what data do we actually need?
- > **Structured data retrieval** — moving from exploration to rigorous collection
- > **Documentation** — every step documented and reproducible

Come with your Codespace ready and your map URL handy.

**Questions?**