Background

A 1-2 paragraph description of your project, providing the necessary context for someone unfamiliar with your project to understand what it does.

Functionality & MVPs

Fill in this template:

In { project name }, users will be able to:

{ 4-6 core features or functionalites of your project }

{ feature }

{ feature }

{ feature }

{ feature }

In addition, this project will include:

{ 2-4 other aspects of your project, including instructions & README }

{ other aspect of project }

{ other aspect of project }

Wireframes

Draw the layout of your project: the major sections of the game view, the chart and its legend, the game controls and about me links, etc. This doesn't have to be a fully detailed mockup; it can just be labeled boxes showing the layout of your project. See the Sample Proposal for an example.

Wireframe.cc is a great tool for quickly creating wireframes.

Technologies, Libraries, APIs

What technologies, libraries, and APIs will your project use? If you're building a game, you might use native browser technology like the Canvas API, or you might use a library like three.js. If you're doing data visualization, you might use d3 for rendering charts, and an API to fetch data.

If you're still deciding between a few different libraries or APIs, you can list the ones you're considering, and the potential benefits and drawbacks of each.

If your project needs a backend (see below), list that here as well.

Implementation Timeline

Here you will detail what you hope to get done each day for this project. It serves as a roadmap for the upcoming week. You'll have the following days to work on this project:

Friday Afternoon & Weekend

Monday

Tuesday

Wednesday

Thursday Morning

It can be quite hard to estimate how long parts of your project might take. Don't worry if you end up straying from this timeline; the goal is to have a plan in place for what you'll do & in what order, and to have a tentative pace.

You should also be aware that presentations will be Thursday afternoon, and deploying your project to GitHub Pages or Heroku will take some time Thursday morning, so don't plan much for that morning.

Bonus Features (Optional)

Backend

Some projects might need a lightweight backend.

Common Reasons for Needing a Backend:

1. Hiding API Keys

If you're using an API that requires an API key, you don't want that public on GitHub. But if your project is frontend only, there's nowhere to keep that API hidden. You'll need a backend, where you can hide your API key as an environmental variable on your server.

2. CORS Proxy

In addition, you might run into issues with an API where your AJAX requests are blocked by your browser's CORS policy. To summarize this policy: if you make an AJAX request to a website you're not currently on, your browser will default to blocking this request, unless the website's server explicitly allows requests from other websites.

There are good, security-related reasons for this policy, but it can also be a pain: some APIs that were meant for public usage haven't update their servers to provide these explicit permissions. To get around this, you need to make the request from a server, not a browser.

What To Do

You'll need a very lightweight backend written in JavaScript. After following the regular setup instructions, follow the server setup instructions; everything is explained there.

When it comes time to deploy your project as a live site, you won't be hosting your project on GitHub Pages. Instead, you'll use Heroku. A guide for deploying on Heroku is available here.