

Product Technical Specification

WHY DO WE NEED A PROJECT SPECIFICATION?

There are two reasons why we require new projects to create a specification. First, to ensure that every project member has the same vision for the project. Second, it motivates project teams to create a plan about how to tackle the project without the complication of the actual code. **Describe the “what” of the project, not the “how”!**

Product Specification Requirements:

- **A high-level summary** of the product you are building, keep it concise
- **The problem** (can be a problem that has already been solved) that your product is solving
 - What is the problem?
 - Why do we have this problem?
 - How can we help?
 - Who will benefit? Who are your product's intended users?
- **Team Breakdown**
 - Who is the project manager or point of contact?
 - How will the team be divided by role (for ex. front-end/back-end/design)?
 - Who will be on each subteam/subgroup?
- Your **Minimum Viable Product (MVP)**
 - List features that your product should have. Be specific!
 - What are the main deliverables of your product you want to accomplish first at **minimum**? (**MVP deliverables, high priority**)
 - What are some deliverables you hope to accomplish **after** your MVP? (**Post-MVP deliverables, lower priority**)
 - Describe these features in detail on how they will behave, how you expect them to be used.
 - What are some indicators to show you've completed each feature? How can you test for these indicators?
- **Design** (Technical and Non-Technical)
 - What is the high-level architecture of your product? Provide a diagram of your tech stack and how they connect.
 - If your product has a **front-end** user interface:

- Use Figma or any wireframing tool to visualize your product. Include a publicly viewable link to your wireframe.
- View of product (Refer to the wireframe)
 - How does each page or view of the product look?
 - How does the user navigate between views?
- For each feature's user journey,
 - What is the user journey like for each feature?
 - How does the product look when activating the feature? During usage? After usage?
- If your product will require a **back-end** to provide services to the front-end:
 - What will the front-end be requesting from the backend?
 - Will it need to interact with a database?
- If your product requires a **database** for information storage:
 - What type of database is it? Relational database? NoSQL?
 - What is your database schema/structure?
- If your product requires **other services**, how will they connect to your product and provide support for its' features?
- **REGARDLESS** of your stack, how do you plan to **test your code at each level of the stack**? What potential testing frameworks/suites can your team use to show that each feature is functional and bug-free?
- **(If applicable) REGARDLESS** of your stack, how will your team prioritizing privacy, security, and accessibility in your project?
- **General timeline**
 - High-level timeline of the development process
 - When do you anticipate finishing your product? Each section of features?
 - Take into account how long your learning phase will be, because that takes up a significant portion of time.