

Meta University Eng Project Plan Template

Fill in blanks (enclosed by brackets []) and remove red text as you work through writing your project plan. Your project plan should be a living document and can be changed as you progress through the internship. Make sure to work on this document together with your manager to get feedback, as well as ensuring your project meets the requirements and expectations in the [Project Guide](#).

[Project Name]

Intern: Eric Lin

Intern Manager: Yao Li

Intern Director: [Name]

Peer(s): Xiaoying Li, Silvia Wang

GitHub Repository Link: [Link]

Overview

[Provide a brief description of what your project is about and what problems it aims to solve.]

As a competitive programmer, I try to find new ways to improve my question solving skills. However, while there are websites that list questions and their ELO ratings, there is no website that actively suggests questions, content topics, or resources to practice on. My project aims to solve this problem.

- Category: Mental fitness/stats tool?
- Story: The main goal is to provide the user with suggested problems at their estimated ELO rating, maybe with targeted question topics.
- Market: Competitive programmers aiming to improve
- Habit: For people actively trying to improve, probably daily.
- Scope: The initial scope is to suggest problems based on the ELO rating that the competitive site has. Later on, I want to calculate the estimated rating based on the user's submissions/number of submissions (will have to look at the ELO rating algorithm).

Product Spec

Based on the app description, this section goes into more detail about what the app should do, and what functionalities it must provide to the users.

User Stories

User stories are actions that the user should be able to perform in your app.

First, focus and identify functionality that is required for your MVP (Minimum Viable Product) that conforms to all the project requirements and expectations. Make sure your technical challenges are part of your MVP.

You should also identify optional / nice-to-have functionalities that would be done as stretch goals during MU Week 8 and 9. Remember, *technical challenges should not be optional features*, they must be code complete before the end of Week 8!

Required

- As a competitive programmer:
 - User can see all questions from a site
 - Users can see upcoming contests (with their current timezone and links to contest signup)
 - User can filter questions by rating and question topic tags
 - User can log in
 - User can see their own profile along with their stats below (with a graph library for visualization)
 - User can see their latest submissions
 - User can see their submission stats (how many questions solved within an ELO range)
 - User can see their question content solve stats (percentage of questions solved with certain content tags)
 - User can see the percentage of questions with certain tags in their ELO range.
 - User can see their daily, weekly, and monthly submission count

- Users can see a community board of suggested resources to study with
 - Users can filter resources by tag that can be set by resource author
 - Once logged in, users can upvote/downvote contents
 - Higher rated resources will be placed higher

Optional

- Support multiple competitive sites (required is codeforces)
-

Screen Archetypes

[Describe the different screens that, together, compose the full experience of your app. You can leverage anything you want, such as diagrams and mocks.]

[Using diagrams you can also describe how navigation and presentation of these screens will work on a high-level.]

[These are just high-level representations though. Don't spend too much time building mocks.]

Data Model

[Describe the data you're going to need to back your application. This can include database models (like tables), or external data you'll require from some API.]

Server Endpoints

[Describe the endpoints that your application is going to consume from your server. If you're using REST, then you'll probably want to include the method (GET/POST/etc) and the expected parameters (query parameters, body parameters, etc.)]

Navigation

Project Requirements

[Based on the [Project Guide](#), describe how your project is going to be fulfilling each of the base project requirements.]

Technical Challenges

For your project, you should demonstrate that you can apply what you've learned so far and expand on that knowledge to write code and implement features that go beyond the scope of the projects you worked on during CodePath.

Based on the general idea and direction of your project requirements, your intern manager will create at least two (2) Technical Challenges for you. This section is all about explaining what they are and how you're planning to tackle them - you'll work together with your manager to fill it out.

Technical Challenge #1 - [Name/Small Description]

What

What problem are you solving, and what parts go beyond what you learned in CodePath?

How

Explain in words how you'll solve this problem.

You're encouraged to expand on this section with pseudo-code, links to external frameworks, architecture / design diagrams, anything that you can use to explain this to others!

Technical Challenge #2

What

How

Database Integration

[Describe what you are using for database storage. For example, Parse, MongoDB, Sequelize, etc.]

External APIs

[Describe at least one external API you're using for your project. For example, Google Maps, Spoonacular, OpenWeather, etc.]

Authentication

[Describe how user authentication is handled for your project, including logging in and signing up. Also describe any kind of cookie / session management you're doing and how you're implementing it, and how this affects navigation between different screens by the same user.]

Visuals and Interactions

[Provide details on how your app is fulfilling the following UI craft requirements, and how these are technically accomplished.]

- Interesting Cursor Interaction
- UI Component with Custom Visual Styling
- Loading State

Timeline

Project execution will start in Week 4 of MU. Based on the previously defined requirements, user stories and technical challenges, use the following table to scope out and plan a timeline for deliverables over Week 4 - 9. You can be as detailed as you need, ranging from simply mentioning the user stories, or dividing them into sub-tasks.

You are free to modify the table, add / remove rows or columns, whatever fits your style! The important thing here is that you focus and prioritize certain aspects of your project so you don't get behind and are ready to deliver the MVP - remember your required features should be code complete before the end of Week 8, including both technical challenges!

We also encourage you to leverage project tracking tools such as GitHub Issues or Meta's internal Tasks / GSD tooling to keep manage individual units of work.

MU Week	Project Week	Focus	User Stories
4	1	Focus on the components that will serve as the skeleton of your project. You will probably be using most of what you learned in CodePath to set up things like the client and server repositories, initial routing, login / registration, creating a database with object models, etc.	Example: <ul style="list-style-type: none">- User can login- User can create an account- [Optional] User passwords are encrypted in the database for security
5	2	Week 5 and 6 should be where you focus on the specific requirements of your project.	Example: <ul style="list-style-type: none">- User can create / edit / delete posts- User can chat with other users in real-time (e.g. technical challenge)
6	3	By this point, you should be getting started with your technical challenges as well.	
7	4	You should focus on finishing your MVP and core requirements. By this point, you should be done with at least one of your technical challenges.	
8	5	Continue work on finishing touches and stretch goals for your MVP. By this point, your core functionality and both TAPs should all be in place. It is also a good point to start working on stretch goals that could further expand on the functionality	

		<p>(and technical complexity) of your project.</p> <p>This week you also have to submit your self-review, make sure you allocate enough time for this alongside your final submission for your project!</p>	
9	6	<p>It's time to show others what you have built! Work on a presentation and demo that you will present to other interns to showcase your work. You are also free to continue polishing and expanding on your project!</p>	
10	7	<p>For this week, we have a bunch of extra activities prepared to give you a quick dive of what it is to work at Meta. You will find activities around using internal tools and frameworks, and even committing code to our internal repositories.</p>	