Meta

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: [Emmanuel Mokua]
Intern Manager: [Pablo Moreno Tamez]
Intern Director: [Name]
Peer(s): [Nick Ostrowski, David Flores]
GitHub Repository Link: [Link]

Overview

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

- Category: [Finance]
- Story: [An all-in-one financial expenses tracker for college students to help them keep track of their part-time income, monthly expenses and current small investments]
- Market: [Full-time college students]
- Habit: [Users will use the app daily to help them make better financial decisions and optimize their daily spending]
- Scope: [The initial scope of the app is expenses and income related to common college student activities]
- Problem Statement: [More than 70% of college students report feeling stressed about their finances, and nearly half of the respondents say they are not confident in their ability to manage their money.]
- Solutions Brainstorm: [

1. An app that helps students find part-time remote employment opportunities.
2. An app that connects students with financial resources to educate them on how to best manage their finances in college.
3. An app that helps students keep track of all their expenses and income throughout the entire school year.
4. An app that gamifies student expenditure in comparison to their friends to help them minimize expenses that they would normally not incur.
5. An app that helps students split their expenses with other college students who use similar products online.
   ]

- Features Brainstorm: [
  1. Home Page.
  2. Login Page.
  3. User Profile Page.
  4. Expenses Dashboard.
  5. Income Dashboard.
  6. Sidebar.
  7. Onboarding Page…on first login
     ]
- Stretch Features: [
  1. Portfolio Watchlist (stock market api).
  2. Embed financial resources from outside sources.
  3. AI Assistant (Trained on financial information relevant to college students).
  4. Animated Web App (...using JS GSAP library).
     ]

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

Small example for Facebook app:

- User can login
- User can create an account
- User can create / edit / delete posts
- User can view a feed of posts
- User can add / remove other users as friends
- User can like posts
- User can see their profile

Optional

- User can create multi-media (photos, videos) posts
- User can see notifications of actions made by their friends
- User can edit their profile information
- User can see their friends' profile
- User can comment on posts
- User passwords are encrypted in the database for security

User Roles

College students who are seeking help with managing their finances in a better way.

User Personas

1. “Sarah”: Sarah is a sophomore studying biology at Cornell University. She works part time at the campus library and receives financial aid to cover her tuition. She uses her phone frequently throughout the day to check her account balance and wants an easy way to track her spending.
2. “John”: John is a junior studying computer science living off-campus. He has a part time job at a local grocery store and also earns some money through freelance coding projects. He wants a way to keep track of his various income sources and to ensure that he stays within his budget for each month.

User Stories

- As a college student, I want to be able to easily input my income sources so that I can keep track of my earnings throughout the school year.
- As a college student, I want to be able to categorize my expenses so that I can see where my money is going and identify areas to cut back on spending.
- As a college student, I want to receive notifications when I go over budget in a particular category so that I can adjust my spending accordingly.
- As a college student, I want to be able to set financial goals for the semester/year so that I can work towards saving for larger purchases or paying off debt.
- As a college student, I want to be able to connect the app to my bank accounts so that my transactions are automatically updated and I don't have to manually input every purchase.
- As a college student, I want to be able to compare my spending habits to other students like me so that I can see if I am overspending in certain areas.
- As a college student, I want to be able to access my financial data from multiple devices so that I can stay up-to-date on my spending whether I'm using my phone or laptop.
- As a college student, I want to be able to track my small investments so that I can see how they are performing and make informed decisions about buying or selling.
- As a college student, I want to be able to connect with financial resources to learn more about managing my money effectively.
- As a college student, I want to be able to find part-time remote employment opportunities to supplement my income and gain work experience.

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: - 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: - 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 (and technical complexity) of your project. 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!	
9	6	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!	
10	7	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.	