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: Amaka Amanda Okoro
Intern Manager: David Olaniran
Intern Director: Nicole Cheng
Peer(s): Dave Oyekeye Aditi Anshu
GitHub Repository Link: [Link]

Overview

- Category: Beauty and Personal Care
- Story: Users sign up, log in, and create a personalized skincare profile by answering a series of questions about their skin type, concerns, and preferences. The system uses this information to provide tailored product recommendations from various brands and retailers. Users can also input their sustainability and vegan preferences to filter out products that do not meet their standards.
- Market: Individuals seeking personalized skincare advice, particularly those with specific skin concerns or preferences (e.g., acne-prone, sensitive skin). This platform can cater to a wide range of users, including Young adults, Individuals with specific skin concerns,
- Habit: Weekly(to explore new product recommendations), Monthly(to reassess their skin concerns)
- Scope:
    - Integrating with at least one API from a reputable skincare brand or retailer

    - Developing a comprehensive profiling system to gather user data and provide accurate recommendations
    - Creating a user-friendly interface for users to input their preferences and view personalized product suggestions
    - Implement Filtering

Product Spec

User Stories

Required

- User can login
- User can create an account
- User can view their personalized skincare profile
- User can edit their skincare profile information (e.g., skin type, concerns, preferences)
- User can receive tailored product recommendations based on their profile
- User can filter product recommendations by sustainability and vegan preferences
- User can view a list of recommended products with details (e.g., product name, brand, price, rating)

Optional

- User can upload photos of their skin to help with profiling (e.g., acne tracking, skin tone analysis)
- User can track their skincare routine and progress over time
  User can set reminders for product application or replacement
- User can see notifications of actions made by their friends
- User can view reviews and ratings from other users for recommended products
- User can edit their profile information
- User can receive notifications when new products are added that match their profile preferences

Screen Archetypes

ie. wireframes

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

1. Welcome/landing page
    - "Sign Up" and "Log In" options
    - App branding
    - Navigation: Directs user to authentication flow.

2. Authentication Screens

Sign Up

- Input fields for email, password, and optional name
- Optional social login (Google/Apple)

Log In

- Email/password input
- "Forgot Password?" link

3. Profile Questionnaire

Series of form questions about:

- Skin type (oily, dry, combination)
- Skin concerns (acne, sensitivity, hyperpigmentation)
- Preferences (fragrance-free, cruelty-free)
- Sustainability/vegan filters

Navigation: Upon completion, directs to Profile Summary and Recommendation screen.

4. Skincare Profile Summary

- Displays user's saved skin profile with edit capabilities.
- Features:
    - Overview of all profile attributes

    - “Edit Profile” button
    - Option to upload photo for tracking or skin tone matching (optional feature)

5. Product Recommendations Screen

- Provides curated skincare suggestions based on user profile.
- Features:
    - Scrollable product cards with:
        - Product image
        - Name, brand, price, ingredients preview
        - Tags (e.g., vegan, sustainable, fragrance-free)
    - Filter bar (toggle for vegan/sustainable preferences)
    - "Add to Favorites" or "Save" functionality

6. Product Detail Screen

- Gives more in-depth info on a selected product.
- Features:
    - Full description, ingredients list, benefits
    - Reviews & ratings from other users
    - Retailer links to purchase
    - “Back to Recommendations” navigation

[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.]

We will integrate with at least one reputable skincare product API to fetch product information. All external product data will be normalized and stored in the products table for easy querying and filtering based on user profile inputs.

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

Technical Challenge #1 - Personalized Recommendation Engine

What

Develop a personalized recommendation engine that can accurately suggest skincare products based on user profiles. This goes beyond basic filtering and requires implementing machine learning algorithms to analyze user data and preferences.

How

- Data Collection: Gather user data through profile inputs, including skin type, concerns, and preferences.
- Algorithm Selection: Choose a suitable machine learning algorithm (e.g., collaborative filtering, content-based filtering) to analyze the data.
- Model Training: Train the model using a dataset of skincare products and user preferences.

- Integration: Integrate the recommendation engine into the application, ensuring it updates recommendations in real-time as user profiles change.
- Testing and Optimization: Continuously test and optimize the model for accuracy and performance.

Technical Challenge #2- Real-time Skin Analysis

What

Implement a feature that allows users to upload photos of their skin for real-time analysis. This feature should provide insights into skin conditions and suggest products accordingly.

How

- Image Processing: Use computer vision techniques to process and analyze uploaded images.
- Feature Extraction: Extract relevant features from the images, such as skin tone, texture, and blemishes.
- Analysis and Feedback: Develop algorithms to assess skin conditions and provide feedback or recommendations.
- User Interface: Design an intuitive interface for users to upload images and view analysis results.
- Security and Privacy: Ensure that all image data is handled securely and in compliance with privacy regulations.

[Optional] Technical Challenge #3 - Sustainable and Vegan Product Filtering

What

Implement a feature that allows users to filter skincare product recommendations based on sustainability and vegan preferences. This requires integrating with external APIs or databases to retrieve product information and applying complex filtering logic.

How

- API Integration: Integrate with external APIs (e.g. Environmental Working Group, Leaping Bunny) to retrieve data on sustainable and vegan skincare products.
- Data Modeling: Design a data model to store and manage product information, including sustainability and vegan certifications.

- Filtering Logic: Develop complex filtering logic to match user preferences with product data, ensuring accurate and relevant recommendations.
- User Interface: Design an intuitive interface for users to select their sustainability and vegan preferences and view filtered product recommendations.
- Performance Optimization: Optimize the filtering process to ensure fast and efficient results, even with large datasets.

Database Integration

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

We are using MongoDB this will help with interacting with the database in a Node.js environment. MongoDB allows for good document storage and for storing user profiles and skincare preferences.

External APIs

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

Ulta or Sephora unofficial product APIs would be useful for accessing up-to-date product listings, prices, brand details, and ratings.

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.]

User authentication will be handled by the use of firebase authentication. Firebase's built in session persistence ensures that users remain logged in across page reloads unless they explicitly log out.

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.	