1. *How will your project benefit from React?*

What are the main differences between vanilla JavaScript and React?

• Component-Based Architecture: React lets us break the UI into reusable, isolated components (like RecipeCard, Navbar, FilterBar), making the code more modular and maintainable.

• Declarative UI: React makes it easy to create interactive UIs. We describe how the UI should look at any given time, and React takes care of updating the DOM when state changes.

• State Management: Built-in useState, useEffect, and context APIs help manage state efficiently across the app (e.g., favorite recipes, filters).

• React Router: Easy navigation between views like Home, Recipe Detail, Favorites, Upload Page, etc.

• Ecosystem Support: React has a huge ecosystem (component libraries, hooks, dev tools) to speed up development.

What will React allow you to do that would have been difficult with vanilla JavaScript?

• Real-time UI updates for things like filtering recipes, favoriting/unfavoriting recipes.

• Managing app-wide state (e.g., user login, shared favorites) using context or state management libraries.

• Seamless routing between pages without full reloads.

• Easier integration with backend APIs and handling of asynchronous data. If you are evolving an existing JavaScript project, how will it change with React?

• Refactor UI into components (e.g., RecipeCard, RecipeList, Header, etc.)

• Introduce React Router to handle navigation. • Use state hooks instead of manually managing DOM state.

• Replace imperative logic with declarative data binding.

1. *What will the MVP of your project look like?*

What are the core features that your project must have to be useful to someone?

• Recipe Listing, Filtering/Category Selection, Upload Recipe, Favorites Functionality, and Recipe Images Upload & Preview.

What are some extra features you might add later if you have time?

• We will add recipes and drinks that don’t contain the big 9 allergies, Dark Mode Toggle, Social Sharing, and Tag System.

1. *Expected List of React Components in the MVP.*

A brief list of the custom React Components you will need to develop for your application.

* App, Header, RecipeList, RecipeCard, RecipeDetail, FavoritesPage, FilterBar, LoginForm, and UploadRecipeForm.

Brief justifications for including these components.

* Main wrapper that handles routing and global state/context, Navigation bar with links (Home, Favorites, Upload, Login/Logout), Displays a list of recipe cards and full recipe view with ingredients, instructions, name, and pictures. Component to select, filter, add/delete recipe.

1. *What is your production roadmap?*

What are the main things that you plan to complete each week to finish your project on time?

Week 1:

• Basic idea of app

• Create basic UI layout

• Build public Recipe List Page

• Set up Sign Up / Login

Week 2:

• Add filtering functionality

• Build RecipeDetail and FavoritesPage

• Implement adding/removing favorites

• Begin UploadRecipeForm

Week 3:

• Finalize recipe upload

• Add Private Routes for authenticated pages

Week 4:

• Polish UI/UX

• Test all features + fix bugs

• Prepare demo and documentation

What are the potential "roadblocks" that you anticipate in the next few weeks, and how to you plan to deal with them?

* Getting enough recipes, authentication complexity, state issues, limited time frame, file uploads, and form validation.

If you are working on this as a group project, how are you dividing the workload? Who is responsible for what?

* We are equally working on everything together even if we are working on individual task. For example: Auth system, Upload Recipe, Private Routes, RecipeList, RecipeDetail, FavoritesPage, Filters, Search, Styling, Responsive design.