MyCookbook Development Roadmap

**Notes to Self:**

* This application is slightly uncharted territory in a lot of ways… take 1 step at a time and focus on 1 component at a time, and try not to get overwhelmed by thinking about every component at once!!
* Past projects can be guides (my old React portfolio, old mycookbook PHP app, etc)
* DO NOT worry about the styling until functionality has been hammered out. Style isn’t an afterthought, but it won’t be the difficult part of creating this application…
* HAVE FUN with this!

1. ~~Develop this final project plan and have plan approved~~
   1. ~~Modify if needed to fit scope of class~~
2. Create wireframe for project
   1. Map out UI using tool like Figma
   2. Map out likely components needed in React
   3. Map out database elements
   4. Map out API routes and Lambda functions necessary to tie things together
3. Create a sandbox environment to experiment with the database and AWS design.
   1. Take notes on the setup for the actual setup
   2. Experiment with CRUD operations on the DynamoDB database
   3. Mess with other AWS services
   4. Mock out navigation and a header with authorization applied (What is in the header??? Depends on if a user is logged in!!)
   5. Mock out other nav bars
   6. Mess with the database from the user interface – get CRUD working with a workable database
   7. Prototype out the input forms for ingredients and instructions…
4. Create the initial Amplify application and begin to create the application
   1. Create the React application
   2. Add whatever basics are necessary to get started
5. Begin database development in AWS
   1. Create the DynamoDB database for the application
6. Set up user login/log out
   1. Use AWS Cognito to authenticate users for the application.
   2. Test user creation
   3. Test user login
7. Begin front end development
   1. Set up the frames for the basic components that will be needed for the application…
      1. Home page
      2. User Console page
         1. User categories menu
      3. User main recipe view/search page
      4. Single Recipe Page
      5. Create Recipe Page
      6. Update Recipe Page
      7. Delete Recipe Page
8. Set up create category
   1. Create necessary components to allow a user to create a category
   2. Create necessary components to allow a user to update a category
9. Set up recipe creation
   1. Create necessary components to allow a user to create a recipe
   2. Make sure new recipes properly display
10. Set up recipe updates
    1. Create components necessary to allow a user to update a recipe
11. Set up recipe/category deletes
    1. Create components necessary to allow a user to delete a recipe.
    2. Create components necessary to allow a user to delete a category. This may cascade to a categories recipes – TBD
12. Set up styles
    1. Set up basic styling to have the site be good enough to go live
13. Touch up any bugs and test the website
    1. Make sure everything runs smoothly for the MVP
14. Deploy the MVP using amplify
    1. Deploy the application
    2. Test the application after deployment
    3. Get a better URL to use for the website
15. Consider and plan for advanced feature development…..
    1. Custom colors/themes
    2. Add images to recipes
    3. Search engine optimization
    4. Print recipes
    5. Think about deploying to another AWS environment – with a different IAM user specifically for this. This will make this project more maintainable, and open to adding other devs 😊

Important Development Notes

**Steps I’ve Taken so Far (Copy these)**

* Created the database
* Added a Lambda function
* Added the API Gateway layer
  + Note: The API layer…. Can probably just be named “recipe”? I don’t know if there is going to need to be a separate function for every single CRUD operation. We shall see.

**DynamoDB Notes**

DynamoDB is schemaless…. This makes development in it a little bit interesting… and adds extra challenges! But it’s WAY more dynamic. There is actually pretty much no structure. Here is my schema:

**Partition Key: userID 🡨 Will eventually be taken from cognito/the user**

**Sort Key: recipeID 🡨 Generated by UUID**

The partition key and sort key form a composite primary key. They are both needed for a record to be entered.

Interestingly, there is NO auto increment feature for key fields!!! That means that one must be manually provided/used. The user ID will come from Cognito eventually… but for the recipes, there will need to be some way to automagically generate a unique ID

**Enter the UUID npm package**. **I’ll be using this to generate a recipeID**

**Instructions from GPT/UUID NPM:**

* npm install uuid
* Use the docs to use the tool, it looks super simple.

<https://www.npmjs.com/package/uuid> <-- This is the package from npm. It has an insane amount of downloads so I can only imagine it is totally safe.

**Integrating Lambda and API Gateway with the DynamoDB Storage**

d

Daily Workflows

**4/3/2025**

Here is a breakdown of the goals I have this evening, working from 6:30 – 10:30PM

* Get the DynamoDB database connected with the React application through Lambda and API gateway.
  + DO NOT worry about authorization first. Just worry about connecting the components together
  + Make CRUD operations, if possible
  + Touch on Authorization once that is all put together
  + Think about how to make sure a user is going to see only their particular recipes…. Stressing on this might be for another day though
* **Focus on the basic DynamoDB connection and CRUD operation setup.**
* Give some thought into the wireframes
  + Pages
  + Menus
  + Nav components