

### Add New Career Task

### My Career Tasks

test1

HighSkill Development2/27/2025

### Motivational Message Board

When you feel like quitting, remember why you started.

2/10/2025, 6:52:38 PM

The way a team plays as a whole determines its success.

2/10/2025, 6:52:38 PM

Page 1 of 201

# CookBook-Meal Master

<https://github.com/Jocelynmie/CookBook>

# Project Objectives

---

## **Streamline Meal Planning Process**

- Create an intuitive platform that simplifies the entire cooking workflow from recipe selection to grocery shopping
- Reduce time spent planning meals and preparing shopping lists

## **Automate Shopping List Generation**

- Develop a smart system that automatically compiles ingredients from selected recipes
- Consolidate duplicate ingredients and standardize units for efficient shopping

## **Build a Comprehensive Recipe Management System**

- Provide tools for users to browse, create, and organize recipes with detailed information
- Support visual content (images) to enhance the recipe browsing experience

## **Improve Grocery Shopping Efficiency**

- Generate organized shopping lists that can be easily referenced while shopping
- Provide quantity and measurement information to ensure accurate purchasing

# What you are most proud of

**Smart Ingredient Consolidation:** The `generateShoppingList` function in `mealPlan.js` intelligently combines duplicate ingredients across multiple recipes, adding up quantities when the name and unit match. This solves a common frustration in meal planning where the same ingredient appears multiple times.

**One-Click Shopping List Addition:** The implementation in `RecipeCard.jsx` that allows users to add all ingredients from a recipe to the shopping list with a single click creates a seamless experience between browsing recipes and preparing to shop.

**Responsive User Feedback:** Throughout the application, there are thoughtful status updates and notifications that keep users informed about their actions (like adding recipes to the shopping list or submitting suggestions).

# What was the hardest part

**Ingredient Normalization:** Creating an algorithm that could intelligently identify and combine the same ingredients across different recipes proved difficult. The code in the `generateShoppingList` function in `mealPlan.js` required careful implementation to handle variations in how ingredients might be entered.

**Unit Handling:** Managing different units of measurement was particularly challenging. The system needed to recognize when ingredients were the same but had different units (e.g., "3 tablespoons of olive oil" vs. "45ml olive oil") while avoiding incorrect combinations.

**Asynchronous State Management:** Coordinating the shopping list state between recipe additions and removals required careful handling of asynchronous operations. Ensuring the shopping list always reflected the current state of selected recipes without race conditions was technically demanding.

# Technologies Used

## Backend

- **Node.js:** JavaScript runtime for the server environment
- **Express.js:** Web application framework for building the API endpoints
- **MongoDB:** NoSQL database for storing recipes, meal plans, and user suggestions
- **MongoDB Driver:** For connecting and interacting with the MongoDB database

## Frontend

- **React:** Core library for building the user interface components
- **React Router:** For navigation between different pages (HomePage, Recipes Page, Shopping List Page, Suggestions)
- **CSS:** Custom stylesheets for component styling
- **Fetch API:** For making HTTP requests to the backend services

# Conclusion

**MealMaster represents a thoughtful solution to a common everyday challenge: bridging the gap between recipe collections and grocery shopping. By focusing on the automatic generation of shopping lists from selected recipes, the application delivers genuine utility that simplifies meal planning and preparation.**

**The project successfully demonstrates the practical application of modern web technologies to create a seamless user experience. The React frontend provides an intuitive interface for recipe management, while the Express backend efficiently handles data operations with MongoDB. The component-based architecture ensures maintainability and extensibility as the application grows.**