

# Fronted development project

## **Cook Book**

Team members and their roles:

1. Daniya Fathima. B -(TL)& contentvoiceover
2. Kathija Rayhana. A -Documentation
3. Faiza Fathima. I -Code execution
4. Aysha Mafazha. A. S - Demo video

# **Project Overview**

**Purpose:** Briefly describe the purpose and goals of the project.

**Features:** Highlight the key features and functionalities of the frontend.

# Architecture

## Component structure:

cookbook typically has a well-organized structure that helps readers easily find recipes, follow instructions, and enjoy the content. Here's a breakdown of the common components and structure of a cookbook;

1. Front Matter
2. General Information
3. Recipes Section (core part of the cookbook)
4. Organizational Structure
5. Back Matter

## **State management:**

When you say “state management of cookbook”, it can mean two different contexts:

1. If you mean for a school/home science project
2. If you mean in a software/app context

# Setup Instructions

## **Prerequisites:**

A prerequisite of a cookbook means the special benefits or advantages one gets from using or creating it. Here are some key prerequisites:

- 1. Knowledge at Hand**
- 2. Skill Development**
- 3. Time-Saving**
- 4. Consistency in Taste**
- 5. Variety of Dishes**

# Installation:

It can mean different things depending on the context;

1. In Cooking (kitchen use):

There is no “installation,” but you might mean arranging/organizing a cookbook in your kitchen for easy use, such as placing it on a stand, shelf, or digital app.

2. In IT / DevOps (Chef automation tool):

A cookbook is a set of configuration files and recipes used in the Chef automation tool.

# Folder Structure

## **Client:**

clients refer to the systems or nodes where the recipes from the cookbook are applied.

- 1. Chef Client (Node)**
- 2. Workstation**
- 3. Chef Server**
- 4. End Users (Indirect Clients)**

# Utilities:

1. Automation of Tasks
2. Consistency
3. Scalability
4. Reusability
5. Version Control
6. Flexibility & Modularity
7. Time Saving
8. Error Reduction
9. Environment Management



# Running the application

## **Frontend:**

the frontend of a Cookbook (in the Chef automation context), we usually mean the interface or visible part through which users interact with cookbooks.

- 1. Workstation (Developer/Admin Interface)**
- 2. Chef Manage (Web UI – if enabled)**
- 3. Chef Automate Dashboard**
- 4. Supermarket (Community Frontend)**

# Component Documentation

## Key components:

1. Title & Introduction
2. Table of Contents
3. Chapters/Sections
4. Recipe Layout (core component)
5. Visuals
6. Tips & Techniques Section
7. Index & Glossary
8. References & Acknowledgments

# Reusable components:

1. Recipe Card Template
2. Ingredient Lists
3. Measurement & Conversion Tables
4. Cooking Techniques Section
5. Visual Components
6. Nutritional Info Format
7. Tips & Substitutions Block
8. Glossary & Index
9. Design Layouts / Style Guide

# STATE MANAGEMENT

## Global State

The application uses [Context API / Redux / Zustand – adapt based on your project] to manage global state. This ensures data consistency across multiple components without requiring prop drilling.

Flow: Data is stored in a centralized state store. Components subscribe to relevant slices of the state and update automatically when changes occur.

Example: User authentication status and theme preferences are stored

Local state is managed within individual components using React's `useState` and `useReducer` hooks.

# Usage of cookbook

## Usage of a Cookbook

1. Learning new recipes – Beginners and experienced cooks use it to explore new dishes.
2. Step-by-step instructions – Provides clear guidance on ingredients, quantities, and cooking methods.
3. Reference material – Acts as a handy resource when trying a new cuisine or cooking style.
4. Inspiration – Encourages creativity by giving ideas for meals, flavors, and presentation.
5. Preservation of culture – Traditional cookbooks preserve family, regional, or cultural recipes.

# Styling

Frameworks Used: Tailwind CSS / Material UI / Styled Components / Sass.

- Utility-first classes for faster development.
- Responsive design across screen sizes.

Theming:

- Light/Dark mode toggle.
- Centralized theme variables: colors, typography, spacing.
- Easily extendable for brand customization.

# Testing

Unit Tests: RecipeCard, SearchBar, Buttons

Integration Tests: Recipe addition, deletion, favorite toggling

End-to-End Tests: Navigate through home, recipe details, add/edit recipes using Cypress or Playwright

Code Coverage:

- Reports generated with Jest
- Maintain >80% coverage on critical components



# Future Enchantments

Purpose :

- Show the roadmap for improving the Cookbook app.
- Demonstrates vision and planning to stakeholders.

Detailed Enhancements:

1. Recipe Ratings & Reviews:

- Users can rate recipes 1–5 stars.
- Add text reviews for better community engagement.

2. Advanced Search & Filters:

- Search by multiple ingredients, difficulty, or preparation time.
- Filter by dietary preferences (vegetarian, vegan, gluten-free).

3. Import/Export Recipes:

- Export recipes as PDF or JSON files.
- Import recipes from external sources or other users.



# Conclusion

- The Cookbook app provides a centralized, interactive recipe management tool
- Ensures ease of use, responsive design, and scalability
- Ready for future enhancements: ratings, reviews, sharing, multi-language support