*𝐂𝐨𝐨𝐤𝐁𝐨𝐨𝐤:*

𝒀𝒐𝒖𝒓 𝑽𝒊𝒓𝒕𝒖𝒂𝒍 𝑲𝒊𝒕𝒄𝒉𝒆𝒏

𝑨𝒔𝒔𝒊𝒔𝒕𝒂𝒏𝒕

# 𝐏𝐫𝐨𝐣𝐞𝐜𝐭 𝐓𝐢𝐭𝐥𝐞

***Cookbook: Your Virtual Kitchen Assistant***

## *Team Members*

***• Devapreethi – Frontend Developer  
• Dharshini – UI/UX Designer  
• Fasila – State Management & API Integration  
• Fayiza – Documentation & Testing  
• Hajira Banu – Project Lead***

# 𝐏𝐮𝐫𝐩𝐨𝐬𝐞

***The purpose of Cookbook: Your Virtual Kitchen Assistant is to help users discover, manage, and organize their cooking journey through a digital assistant. The application allows users to search for recipes, save favorites, create shopping lists, and customize their own digital cookbook.***

## *Features*

***• Search recipes by name, cuisine, or ingredients  
• Step-by-step cooking instructions with timers  
• User-friendly dashboard for saved recipes  
• Personalized shopping list generator  
• Dark/Light theme support  
• Responsive and mobile-friendly design***

# 𝐀𝐫𝐜𝐡𝐢𝐭𝐞𝐜𝐭𝐮𝐫𝐞

## *Component Structure*

***App.js – Root component, contains routes and global state providers  
Navbar – Handles navigation between pages  
HomePage – Displays featured recipes and search bar  
RecipeList – Shows search results or category-based recipes  
RecipeDetails – Full recipe with ingredients, steps, and images  
Favorites – User’s saved recipes  
ShoppingList – Auto-generated shopping list from selected recipes  
ThemeToggle – Switch between light/dark mode***

## *State Management*

***Redux Toolkit for global state (favorites, shopping list, user preferences)  
Local State within components (search inputs, modal visibility)***

## *Routing*

***Implemented using React Router v6  
/ → HomePage  
/recipes → RecipeList  
/recipes/:id → RecipeDetails  
/favorites → Favorites  
/shopping-list → ShoppingList***

# *Setup Instructions*

## *Prerequisites*

***Node.js (>=16.x)  
npm or yarn***

## *Installation*

***1. Clone the repository:  
 git clone https://github.com/your-username/cookbook-virtual-assistant.git  
 cd cookbook-virtual-assistant  
  
2. Install dependencies:  
 npm install  
  
3. Set up environment variables in .env:  
 REACT\_APP\_API\_KEY=your\_api\_key***

𝐅𝐨𝐥𝐝𝐞𝐫 𝐒𝐭𝐫𝐮𝐜𝐭𝐮𝐫𝐞:

***cookbook-virtual-assistant/  
│── public/ # Static assets  
│── src/  
│ ├── assets/ # Images, icons, etc.  
│ ├── components/ # Reusable UI components (Navbar, ThemeToggle, etc.)  
│ ├── pages/ # Page-level components (HomePage, RecipeList, etc.)  
│ ├── redux/ # State slices and store  
│ ├── hooks/ # Custom hooks  
│ ├── utils/ # Helper functions  
│ └── App.js # Main app file  
│── package.json***

𝐑𝐮𝐧𝐧𝐢𝐧𝐠 𝐭𝐡𝐞 𝐚𝐩𝐩𝐢𝐥𝐜𝐚𝐭𝐢𝐨𝐧

***Start the frontend server locally:  
npm start***

## 𝐂𝐨𝐦𝐩𝐨𝐧𝐞𝐧𝐭 𝐃𝐨𝐜𝐮𝐦𝐞𝐧𝐭𝐚𝐭𝐢𝐨𝐧:

## *Key Components*

***RecipeDetails  
- Props: id (string)  
- Purpose: Fetches and displays recipe details including ingredients, steps, and nutritional information.  
  
ShoppingList  
- Props: recipes (array of recipe objects)  
- Purpose: Generates a shopping list based on selected recipes.***

## *Reusable Components*

***Button – Custom styled button with variants  
Modal – For displaying popups (e.g., recipe preview)  
SearchBar – Input field with debounce for searching recipes***

## 𝐒𝐭𝐚𝐭𝐞 𝐌𝐚𝐧𝐚𝐠𝐞𝐦𝐞𝐧𝐭: *Global State*

***Favorites Slice: Stores user’s saved recipes  
Shopping Slice: Handles shopping list data  
Theme Slice: Stores user’s theme preference***

## *Local State*

***Used for small UI interactions such as toggling modals, handling form inputs, etc.***

# User Interface

***HomePage: Search bar + featured recipes carousel  
RecipeDetails Page: Recipe image, ingredients, instructions, and 'Add to Favorites' button  
ShoppingList Page: Auto-generated shopping list with checkboxes  
Favorites Page: Grid view of saved recipes***

## 𝐒𝐭𝐲𝐥𝐢𝐧𝐠:

## *CSS Frameworks/Libraries*

***Tailwind CSS for utility-first styling  
Styled Components for theme management***

## *Theming*

***Light/Dark theme toggle with persistent preference stored in Redux and localStorage***

## 𝐓𝐞𝐬𝐭𝐢𝐧𝐠:

## *Testing Strategy*

***Unit Tests: Component-level tests using Jest  
Integration Tests: Data fetching & UI rendering using React Testing Library  
End-to-End Tests: User flows tested with Cypress***

## *Code Coverage*

***Coverage ensured with Jest + Istanbul reports***

# Screenshot or demo

# 

# 

# 𝐊𝐧𝐨𝐰𝐧 𝐈𝐬𝐬𝐮𝐞𝐬:

# *- API response delay may affect recipe loading.*

# *- Dark mode not fully optimized for some pages.*

# *- No offline support yet.*

# 𝐅𝐮𝐭𝐮𝐫𝐞 𝐄𝐧𝐡𝐚𝐧𝐜𝐞𝐦𝐞𝐧𝐭𝐬:

# *- Voice-enabled cooking assistant.*

# *- AI-based meal planner.*

# *- Social sharing of recipes.*

# *- Offline recipe access with PWA support.*

# *- Integration with smart kitchen devices.*