# Meal App Project Documentation

## 1. Project Title

**Meal App** – A React Native application for browsing meals, viewing recipes, managing favorites, and user authentication.

## 2. Project Overview

The **Meal App** allows users to: - Search meals by name. - View random meals using “Surprise Me” feature. - See detailed recipe information including ingredients, steps, and YouTube tutorials. - Add/remove meals from favorites. - Register and login as a user. - View personal user details and logout.

The app uses **React Native**, **Redux Toolkit**, **AsyncStorage**, and **React Navigation**.

## 3. Project Objectives

* Provide a **user-friendly interface** for meal browsing and recipe exploration.
* Enable **user authentication** for personalized experience.
* Implement a **favorites system** that persists user preferences.
* Show **detailed recipe information** including category, area, ingredients, steps, and video links.
* Use modern **React Native development practices** with Redux for state management.

## 4. Technologies Used

| Category | Tools/Libraries |
| --- | --- |
| Frontend | React Native, JavaScript, JSX |
| Navigation | React Navigation (Stack & Tab) |
| State Management | Redux Toolkit, redux-persist |
| Storage | AsyncStorage |
| HTTP Requests | Axios |
| Icons | React Native Vector Icons (Ionicons) |
| Styling | React Native StyleSheet |

## 5. Project Architecture / Folder Structure

MLDB  
│── android  
│── ios  
│── node\_modules  
│── src  
│ ├── navigation  
│ │ ├── RootNavigator.tsx  
│ │ └── TabNavigator.tsx  
│ ├── redux  
│ │ ├── authSlice.ts  
│ │ ├── favoritesSlice.ts  
│ │ └── store.ts  
│ └── screens  
│ ├── Auth  
│ │ ├── LoginScreen.tsx  
│ │ └── RegisterScreen.tsx  
│ ├── FavoritesScreen.tsx  
│ ├── HomeScreen.tsx  
│ ├── RecipeDetailsScreen.tsx  
│ └── UserDetailsScreen.tsx  
│── App.tsx

## 6. Navigation Flow

**RootNavigator** handles authentication stack and main app flow. - Screens: Login, Register, HomeTabs (TabNavigator), RecipeDetails.

**TabNavigator** handles bottom tabs: - Home, Favorites, UserDetails.

## 7. Redux Setup

**authSlice.ts:** Manages users, currentUser, and login state. **favoritesSlice.ts:** Manages favorite meals with AsyncStorage persistence. **store.ts:** Combines reducers and sets up redux-persist.

## 8. Screens Description

**LoginScreen:** User login with validation.

**RegisterScreen:** Register new users, store in Redux, navigate to login.

**HomeScreen:** - Displays meals and random meal. - Search meals. - Toggle favorites.

**FavoritesScreen:** Show user’s favorite meals with remove option.

**RecipeDetailsScreen:** Detailed recipe info including ingredients, steps, category, area, and YouTube link.

**UserDetailsScreen:** Shows user info and allows logout.

## 9. Key Features & Functionalities

* Search meals by name.
* View random meal with “Surprise Me”.
* Favorite management with persistent storage.
* Detailed recipe instructions.
* User registration and login.
* Logout functionality.

## 10. User Authentication & Favorites Management

* Users register and login.
* Current user stored in Redux.
* Favorites are user-specific, saved in AsyncStorage, and persist across sessions.

## 11. API Integration & Data Handling

* Uses **TheMealDB API** for fetching meals.
* Axios used for HTTP requests.
* Ingredients and measures dynamically extracted.
* Instructions split into steps.
* YouTube links open in external browser.

## 12. Conclusion & Future Scope

* Provides a **complete meal exploration experience**.
* Future improvements:
  + Add categories filter.
  + Improve UI/UX with animations.
  + Support multiple languages.
  + Enable offline mode with cached data.

**End of Document**