

## **Innovation in Recipe Finder Ionic App**

Student: G00439376

### **Overview**

The Ionic App Recipe Finder application satisfies the project specification and incorporates several additional features that extend functionality, improve user usability, and enhance robustness. The innovations focus on value additions in user experience, defensive programming, state management, and clean application architecture aligned with Ionic and Angular best practices from lectures and framework documentation and patterns.

### **Innovation Highlights**

#### **1. Improved User Feedback and Responsiveness**

- Implemented a loading spinner and status message during asynchronous API calls.
- Prevented empty ingredient searches to avoid unnecessary API requests and confusing results.

#### **Benefit:**

Provides clear visual feedback during API calls, improves perceived performance, and prevents repeated user actions of clicking buttons unnecessarily.

#### **2. Enhanced Recipe Presentation**

- Integrated ingredient-specific images using the Spoonacular CDN.
- Dynamically rendered ingredient measurements using separate JSON paths for Metric and US units.
- Formatted cooking instructions into clear, numbered steps with a fallback message when instructions are unavailable or not provided.

#### **Benefit:**

Improves readability and overall usability of the recipe details page beyond the base specification.

#### **3. Enhanced Favourites Management**

- Added a “Clear All Favourites” feature to allow efficient bulk management of stored recipes.
- Implemented a user friendly ‘no favourites’ message with iconography when no favourites exist.

**Benefit:**

Provides better user control and communication, avoiding blank or confusing screens.

**4. Defensive Programming and Robust Data Handling**

- Used optional chaining (?.) and nullish coalescing (??) to safely handle incomplete or unexpected API responses.
- Sanitised user input using encodeURIComponent() to prevent malformed API requests caused by spaces or special characters.

**Benefit:**

Prevents runtime crashes and improves application stability when interacting with external APIs.

**5. Clean Architecture with Shared Services**

- Centralised persistent storage logic (favourites, measurement preferences, selected recipe) into a shared data service.
- Centralised all HTTP communication into a dedicated HTTP service.
- Ensured separation of concerns between UI components and data logic.

**Benefit:**

Improves maintainability, scalability, and code readability while reducing duplication.

**6. Improved Navigation and Lifecycle Awareness**

- Implemented Ionic back navigation using ion-back-button for consistent user flow.
- Applied Ionic lifecycle hooks (ionViewWillEnter) on pages where data may change after navigation and to stop a refresh on page where the data is stale.

**Benefit:**

Ensures up-to-date data is always displayed.