

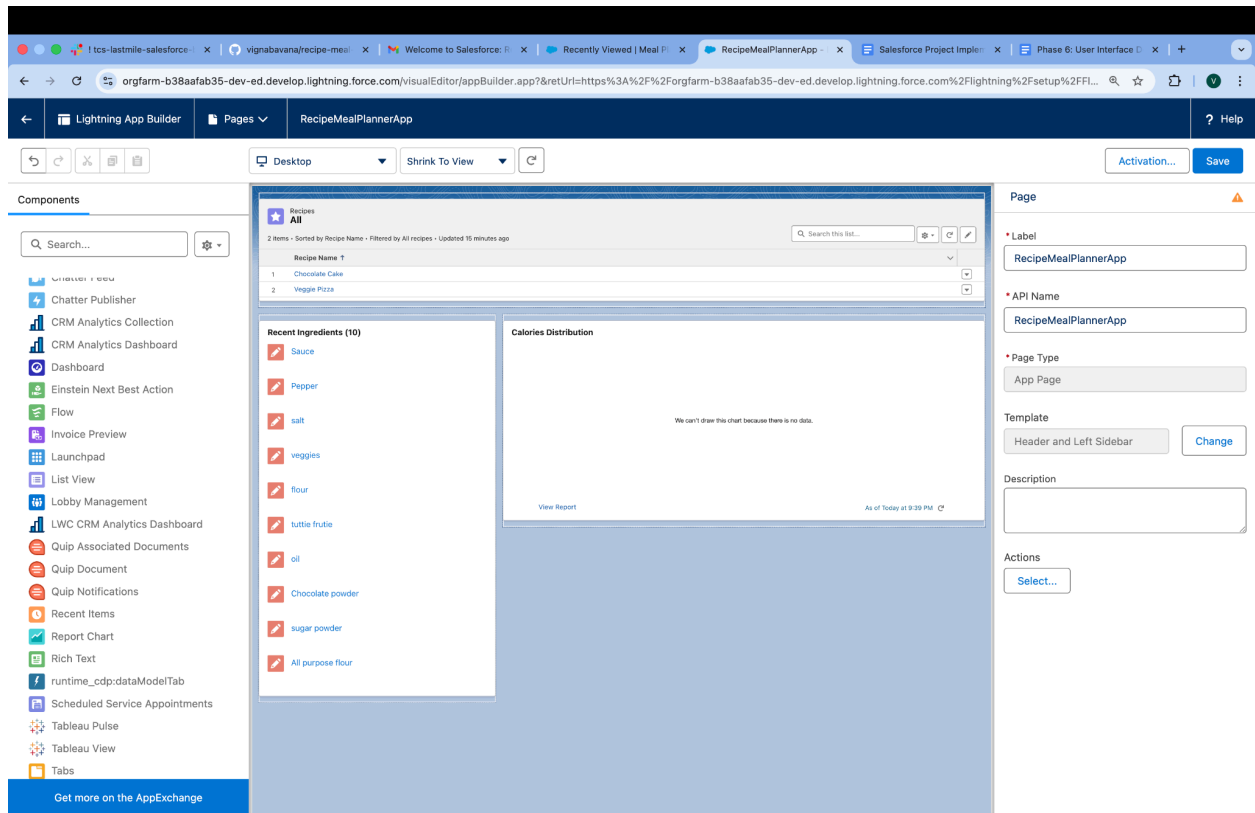
Phase 6: User Interface Development

- Lightning App Builder
- Record Pages
- Tabs
- Home Page Layouts
- Utility Bar
- LWC (Lightning Web Components)
- Apex with LWC
- Events in LWC
- Wire Adapters
- Imperative Apex Calls
- Navigation Service

Lightning App Builder

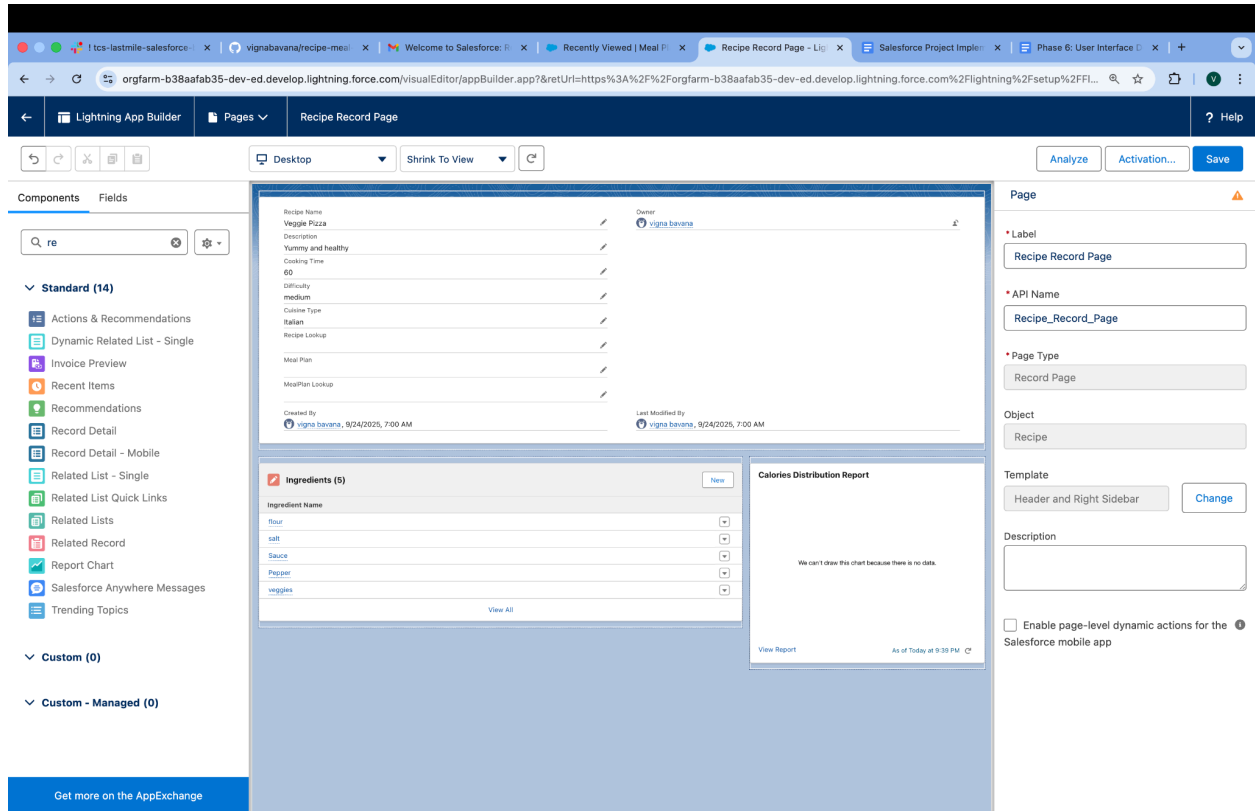
What is it?

- A **drag-and-drop tool** in Salesforce that allows you to build **custom pages** (Record Pages, Home Pages, and App Pages) without writing code.
- You can place **standard components, custom components, and third-party components** into your UI.



Record Pages

1. Created a custom Lightning Record Page for Recipe__c.
2. Top section: **List View** to display all recipes with inline editing enabled.
3. Bottom-left: **Recent Items** showing recent Ingredient__c records.
4. Bottom-right: **Report Chart** displaying calories distribution.
5. Related Lists configured with **Enhanced List** type to show Ingredients linked to each recipe.



Tabs

Purpose: Tabs in Salesforce help users navigate between objects, Lightning pages, and other resources easily. For the Recipe Meal Planner app, custom tabs were created for each object to manage and view records efficiently.

Custom Object Tabs:

- **Recipes** – Create and manage recipes.
- **Ingredients** – Manage ingredients used in recipes.
- **Meal Plans** – View and organize meal plans.
- **Quantities** – Track ingredient quantities.
- **Shopping Lists** – Manage shopping list items.
- **Lookup Tabs** – Recipe Lookup and Meal Plan Lookup for relationships.

- **Calories** – Track calorie details for ingredients or recipes.

Lightning Page Tabs:

- **RecipeMealPlannerApp** – A custom Lightning Page tab created using Lightning App Builder for a unified interface combining list views, reports, and recent items.

Notes:

- Web Tabs and Visualforce Tabs are not used.
- Lightning Component Tabs only include pre-installed managed components.
- All tabs are activated and visible to users in Lightning Experience and Mobile.

Outcome: Users can easily navigate between objects and pages, improving usability and access to meal planning data.

The screenshot shows the Salesforce Setup page for Custom Tabs. The left sidebar contains navigation links for Setup, Home, and Object Manager. The main content area is titled "Custom Tabs" and includes a search bar and a "Help for this Page" link. Below the title, there is a section for "Custom Object Tabs" with a table listing various tabs. The table has columns for Action, Label, Tab Style, and Description. The tabs listed are: Calories (Shield icon), Ingredients (Pencil icon), Meal Plan Lookup (Box icon), Meal Plans (Chalkboard icon), Quantities (Dice icon), Recipe Lookup (Apple icon), Recipes (Star icon), and Shopping Lists (Shopping Cart icon). Below the table, there are sections for "Web Tabs", "Visualforce Tabs", and "Lightning Component Tabs", all of which are currently empty.

Action	Label	Tab Style	Description
Edit Del	Calories	Shield	
Edit Del	Ingredients	Pencil	
Edit Del	Meal Plan Lookup	Box	
Edit Del	Meal Plans	Chalkboard	Tab for meal plan
Edit Del	Quantities	Dice	
Edit Del	Recipe Lookup	Apple	
Edit Del	Recipes	Star	Tab for Recipe object where users can create and manage recipes.
Edit Del	Shopping Lists	Shopping Cart	here is the tab for Shopping list items

Home Page Layouts

Home Page is customized using **Lightning App Builder** in Salesforce Lightning Experience.

Components such as **Recent Items**, **Report Charts**, and **Rich Text** are added to show key data like recent recipes, calories distribution, and instructions.

Each component can be configured with labels, number of records, and display properties.

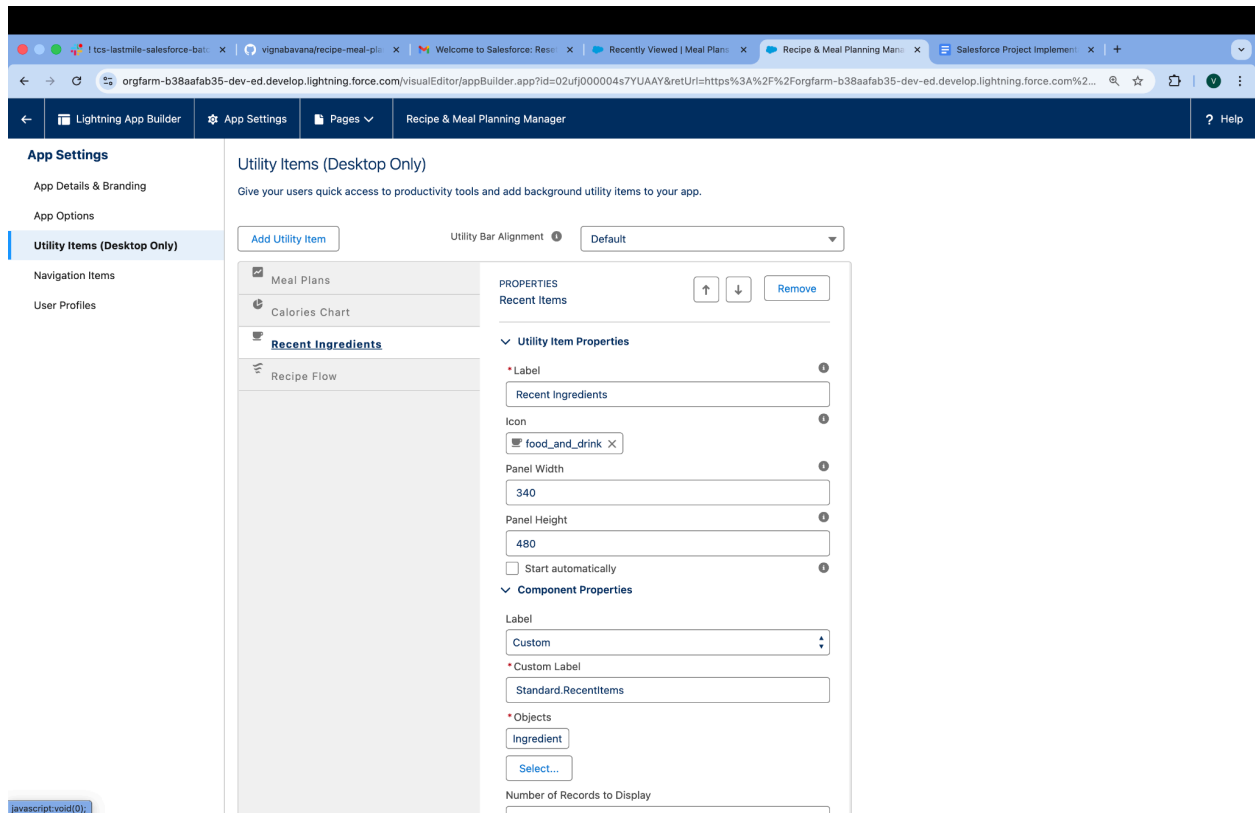
Page is activated and assigned to **profiles** to control visibility for users and admins.

Lightning Home Page replaces classic Home Page Layouts in modern Salesforce apps.

Utility Bar

The Utility Bar provides quick access to essential tools within the Recipe Meal Planner App. For this project, it includes:

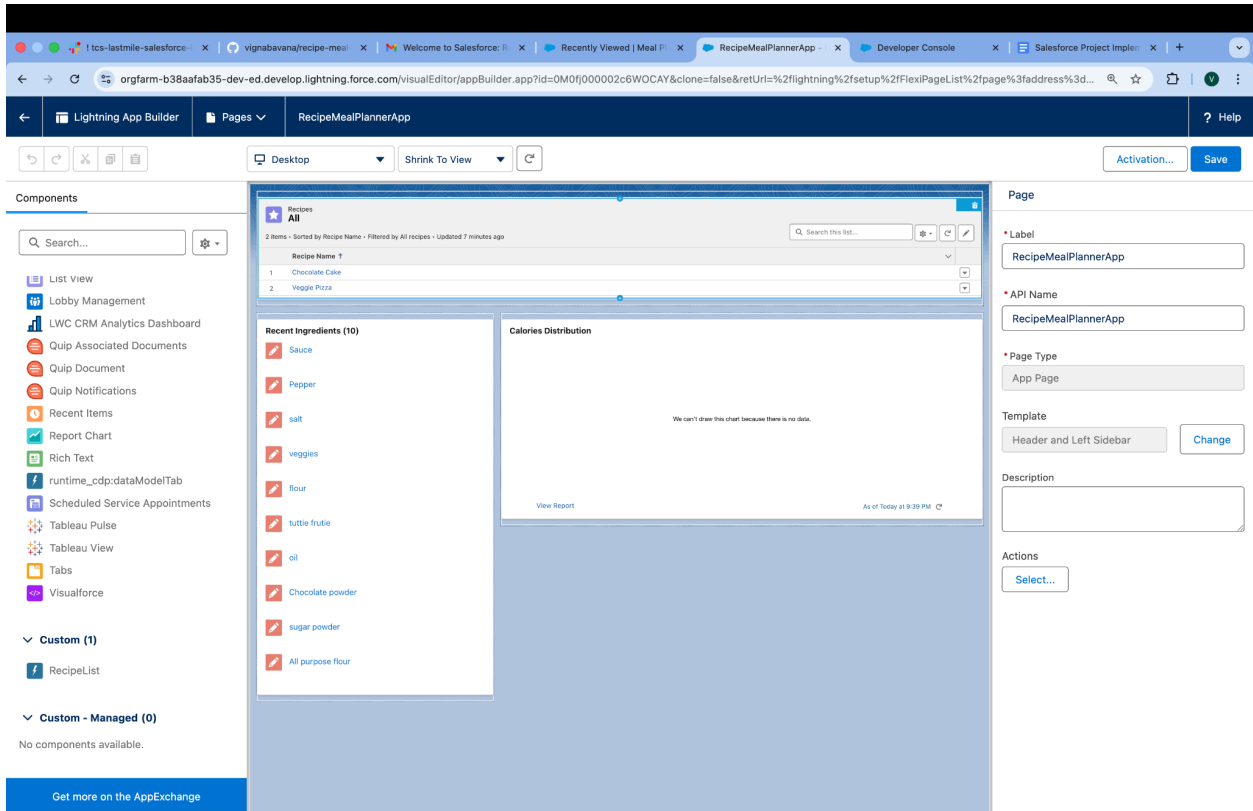
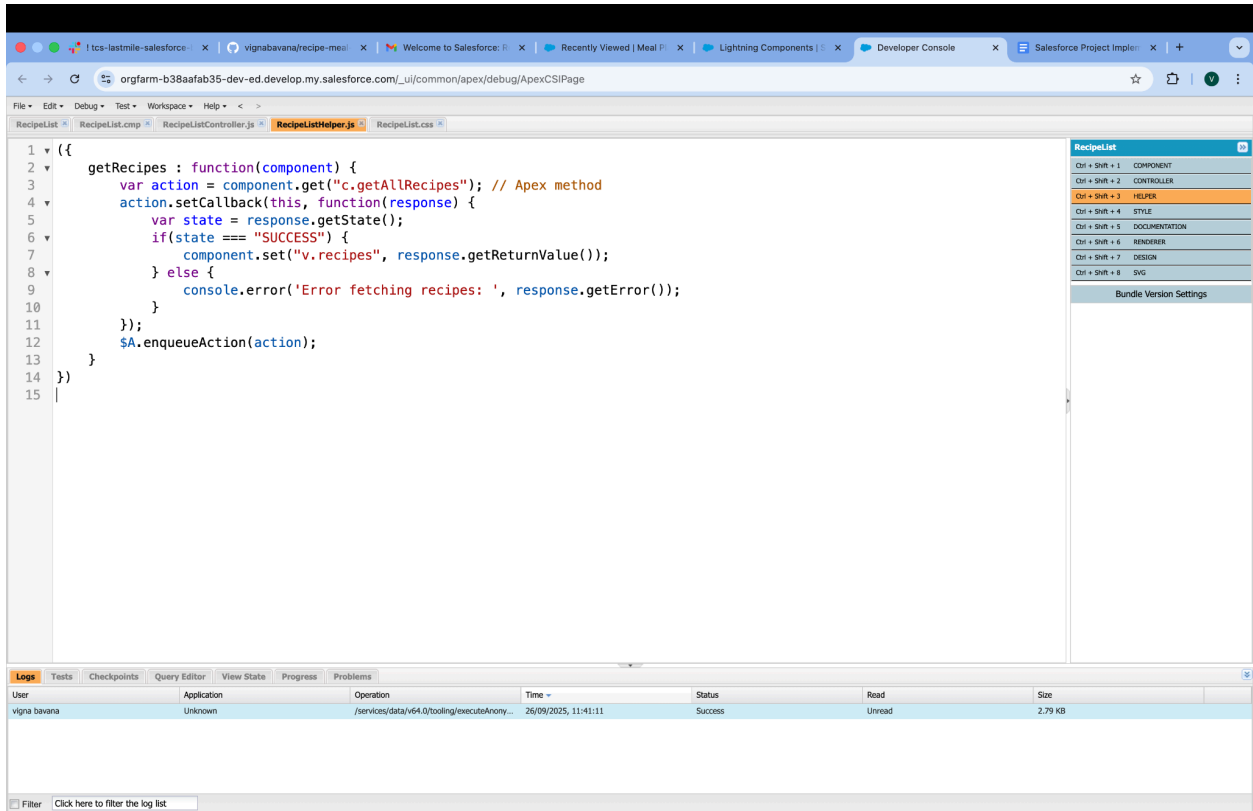
1. **Add Recipe** – Launches a flow to quickly create new recipes.
2. **Recent Ingredients** – Displays the most recently added ingredients for easy reference.
3. **Calories Chart** – Shows a report chart of calories distribution across recipes.
4. **Meal Plans Overview** – Provides a quick view of all meal plans in the system.
5. **Help** – Offers short instructions or guidance for using the app.



LWC (Lightning Web Components)

Lightning Web Components (LWC) in Recipe Meal Planner:

- **Purpose:** Dynamically display recipes, ingredients, and meal plan info.
- **Structure:** HTML for UI, JS for logic, Apex Controller for server data, and Meta for metadata & visibility.
- **Example:** recipeList LWC shows all recipes in a Lightning Card, fetching data from Recipe__c.
- **Benefits:** Fast, reusable, and fully compatible with Salesforce Lightning Experience.



Apex with LWC

Objective: Display Recipe data (Name, ID, and Meal Plan Name) dynamically on a Lightning Page.

Apex Controller: RecipeController with a @AuraEnabled method getRecipes() to fetch records from Recipe__c.

Lightning Web Component: recipeList created using Developer Console.

- **HTML:** Template to iterate and display recipe details.
- **JS:** Calls Apex method using @wire to fetch data and track reactive updates.
- **Meta XML:** Exposes components on App, Record, and Home Pages.

Integration: Component added to RecipeMealPlannerApp Lightning Page via App Builder.

Outcome: Recipes are displayed live with auto-fetching capability, reusable across pages.



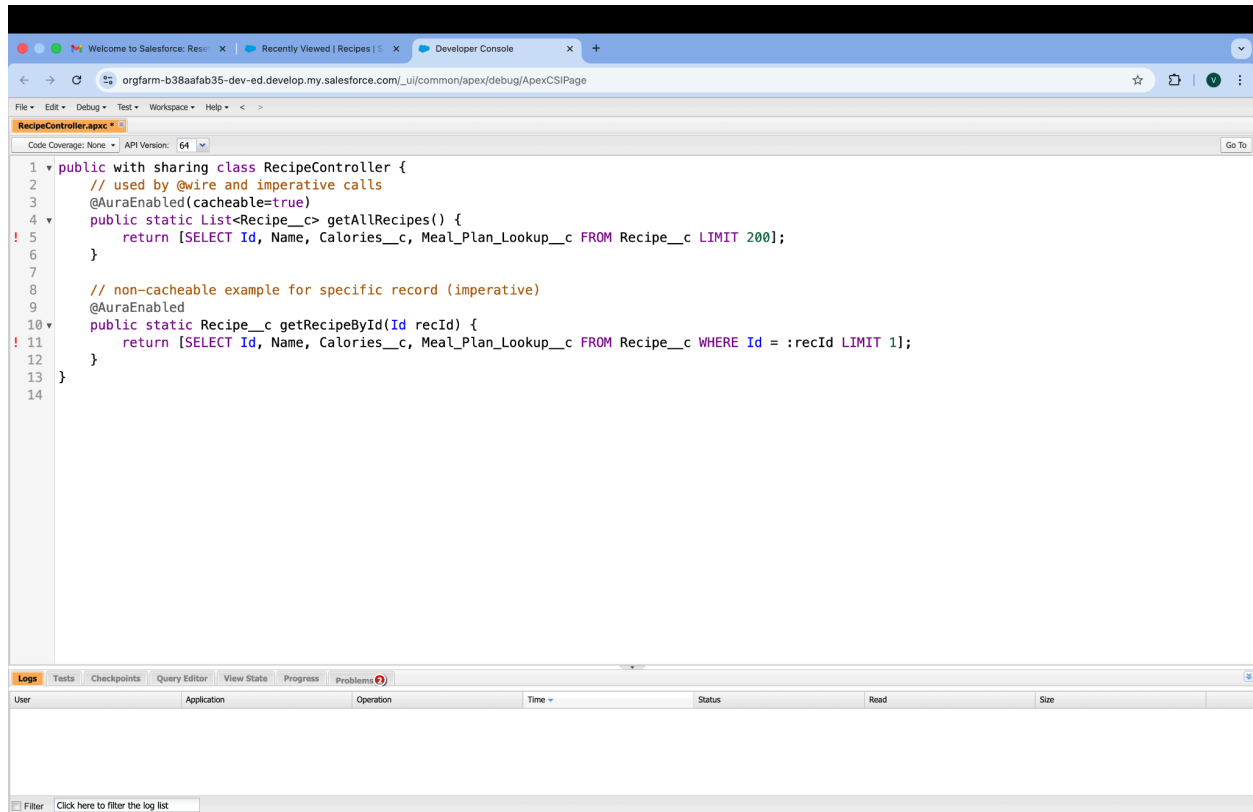
Events in LWC

Events are how components **communicate** with each other in LWC.

- **Child → Parent Communication:** Child component **dispatches an event**, parent listens to it.
- **Parent → Child Communication:** Parent passes data via **@api properties** (not events).

- **Event Types:**

- CustomEvent – to send custom data.
- Lightning events – for platform things like navigation.

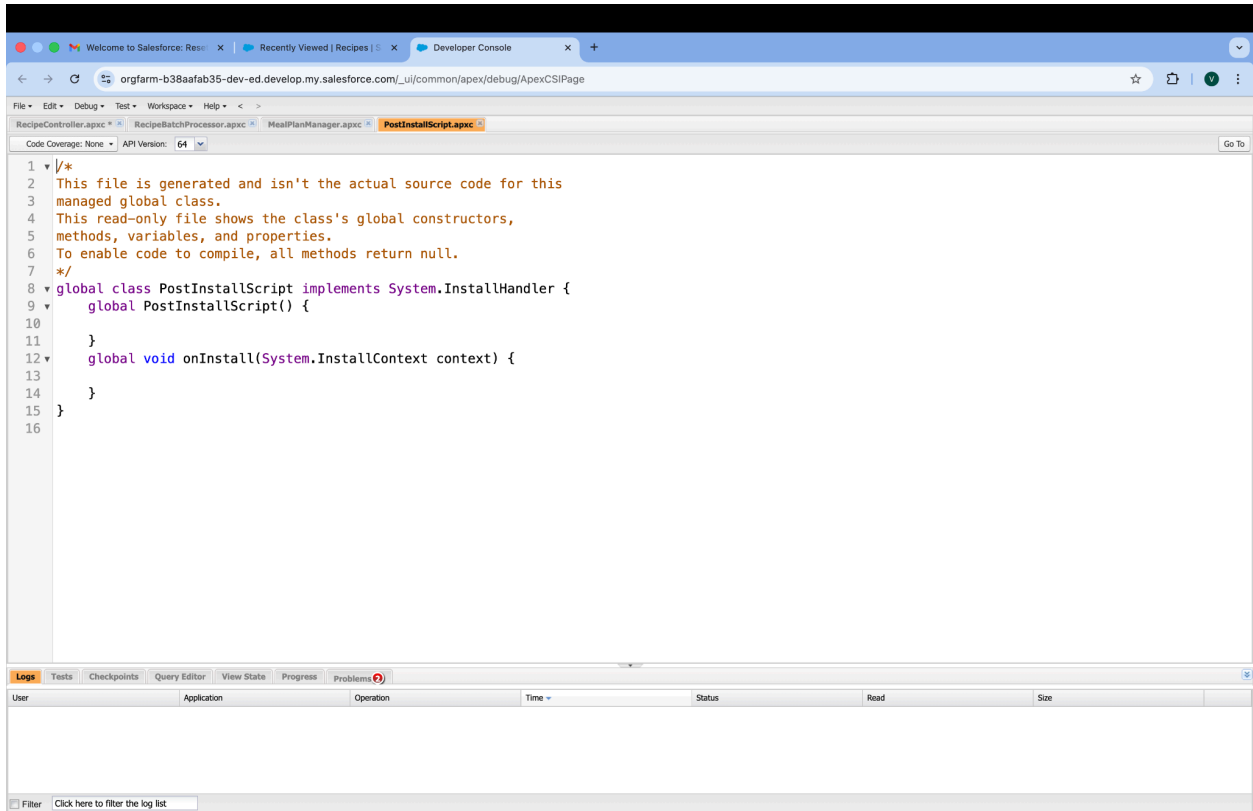


Wire Adapters

Used to **fetch Salesforce data reactively**.

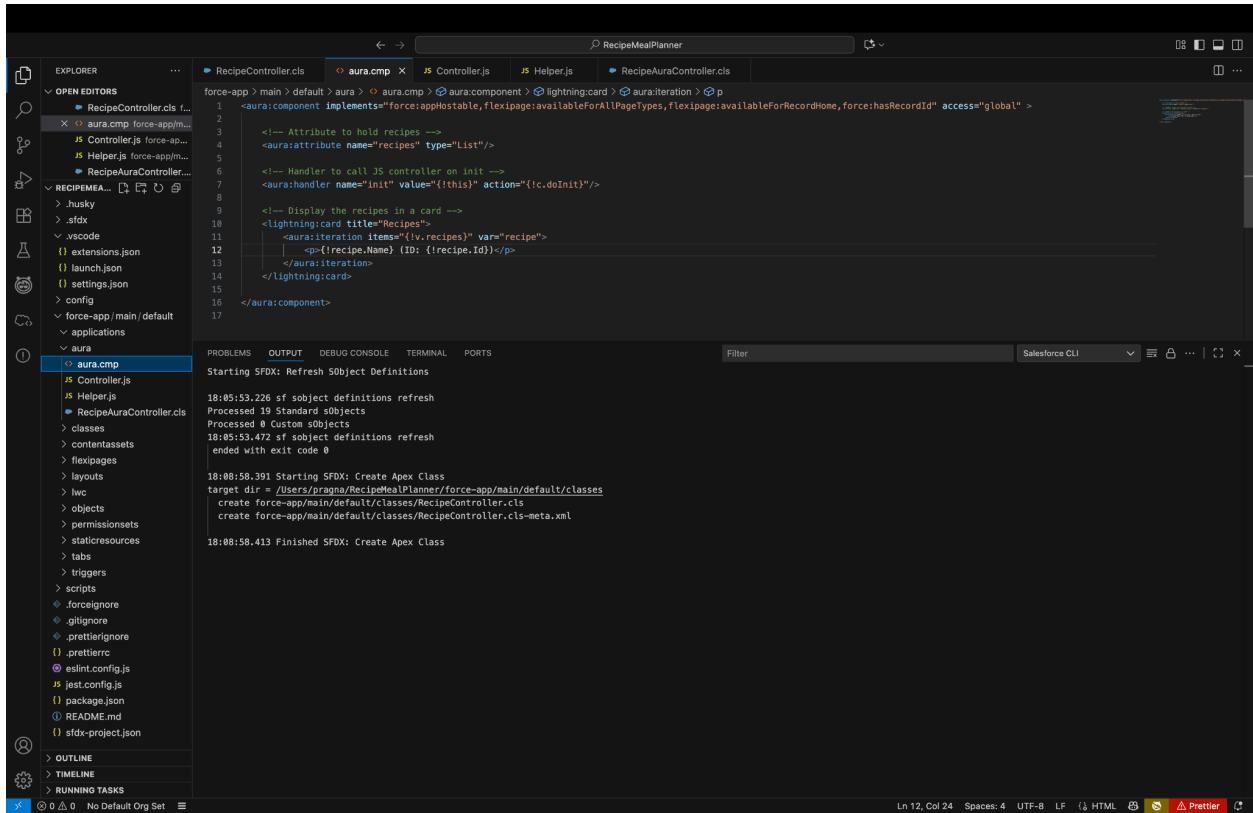
Decorator: @wire automatically fetches data and refreshes when dependencies change.

Based on **Lightning Data Service (LDS)** or **Apex meth**



Imperative Apex Calls

- Unlike wire, this is **explicit** – you call Apex in response to a user action (like button click).
- Syntax: `import method from '@salesforce/apex/ClassName.methodName'`



Navigation Service

Lets you **navigate users programmatically** inside Salesforce.

Provided via `lightning/navigation`.

Supports navigation to records, objects, web pages, apps, and external URLs.