



## Phase 6: User Interface Development

### 1. Lightning App Builder

- Created and configured Lightning pages
- Successfully used drag-and-drop interface
- Components properly placed and configured
- Pages activated and assigned to target audiences

The screenshot shows the Salesforce Lightning App Builder interface. At the top, there's a header with a gear icon labeled "SETUP" and "Lightning App Builder". Below the header, a message states: "The Lightning App Builder provides an easy to use graphical interface for creating custom Lightning pages for Salesforce Lightning Experience and mobile app. Lightning pages are built using Lightning components—compact, configurable, and reusable elements that you can drag and drop into regions of the page in the Lightning App Builder." There are buttons for "View: All" and "Create New View". A navigation bar at the top right includes letters from A to Z and an "All" button. The main area is titled "Lightning Pages" with a "New" button. It contains a table with columns: Action, Label (with an up arrow), Name, Namespace Prefix, Description, Type, Created By, and Last Modified By. A single row is shown: "Edit | Clone | Del" followed by "Carbon\_Emission\_Record\_Page" in the Name column, "Record Page" in the Type column, and "vra, 9/22/2025, 3:12 AM" in the Created By column.

### 2. Record Pages

- Carbon Emission Record Page created and deployed
- carbonFootprintTracker component integrated
- Page layout optimized for user experience
- Proper assignment to Salesforce application

The screenshot shows the Salesforce Object Manager for the "Carbon Emission" object. At the top, it says "SETUP > OBJECT MANAGER" and "Carbon Emission". On the left, there's a sidebar with links: Details, Fields & Relationships, Page Layouts, **Lightning Record Pages** (which is selected and highlighted in blue), Buttons, Links, and Actions, and Compact Layouts. The main content area is titled "Lightning Record Pages" with a sub-header "1 Items, Sorted by Label". It shows a table with columns: LABEL, ORG DEFAULT, APP DEFAULT, OTHER ASSIGNME..., and MODIFIED BY. One item is listed: "Carbon Emission Record Page" under the "LABEL" column, with "Vraddhi Valecha, 9/22/2025, 8:59 AM" under "MODIFIED BY". There are "New" and "View Page Assignments" buttons at the top right of the table.

### 3. Tabs

- Custom tab created for Carbon Emission object
- Tab successfully added to app navigation
- User access permissions properly configured
- Tab styling and properties optimized

The screenshot shows the 'Custom Object Tabs' section of the Salesforce Setup. At the top, there are 'New' and 'What Is This?' buttons. Below is a table with columns: Action, Label, Tab Style, and Description. A single row is present with 'Edit | Del' under Action, 'Carbon Emissions' under Label, 'Leaf' under Tab Style, and an empty Description field.

### 4. Home Page Layouts

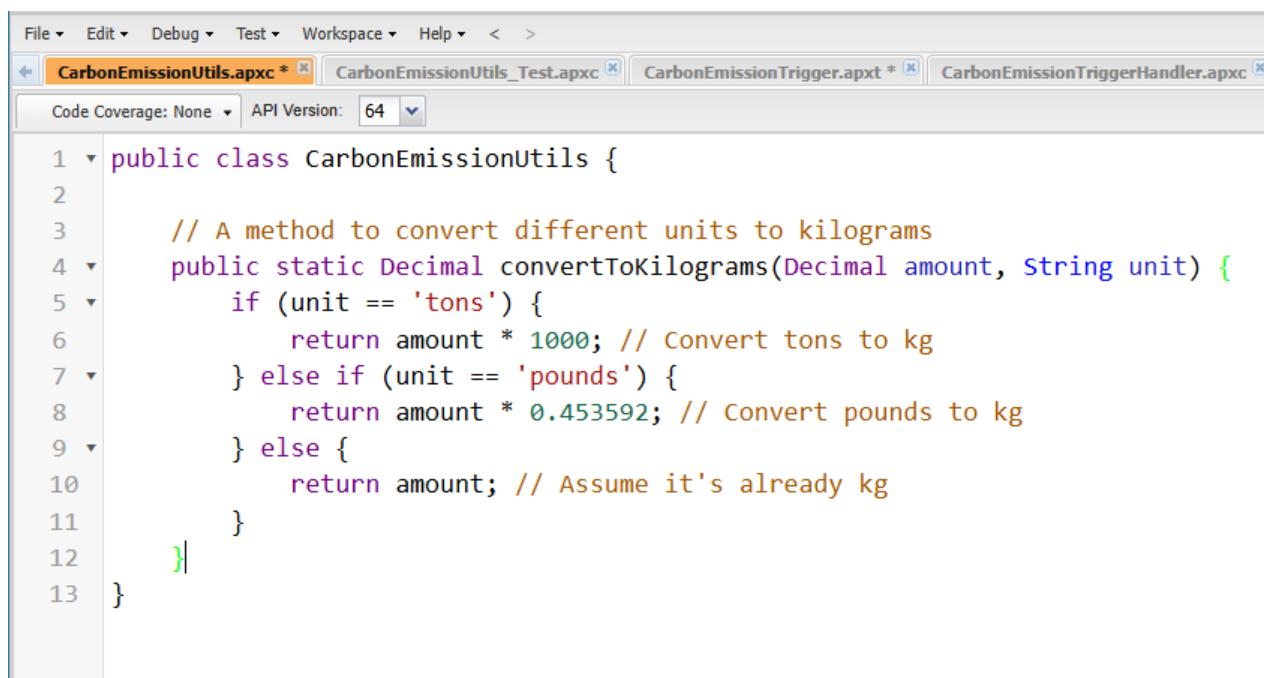
- Home page components successfully configured
- carbonFootprintTracker added to home layout
- Layout assignments to user profiles completed
- Component positioning and sizing optimized

The screenshot displays the Salesforce Home page with several key components:

- Quarterly Performance:** A chart showing revenue trends from August to December. It includes a legend for 'Closed' (orange), 'Goal' (green), and 'Closed + Open (> 70%)' (blue).
- Today's Events:** A section showing a calendar view with a note: "Looks like you're free and clear the rest of the day."
- Today's Tasks:** A section showing a task list with a note: "Nothing due today. Be a go-getter, and check back soon."
- Recent Records:** A list of recent activity including a sample flow report and users Vradhhi Valecha and Test User1.
- Key Deals - Recent Opportunities:** A section showing a deal list with a note: "No deals yet. Select another filter or check back later."
- Assistant:** A sidebar with a nature-themed icon and the message: "Nothing needs your attention right now. Check back later."

## 5. Utility Bar

- Utility bar configured in App Manager
- Component successfully integrated into utility interface
- Utility properties and behavior settings applied
- App published with utility bar enhancements

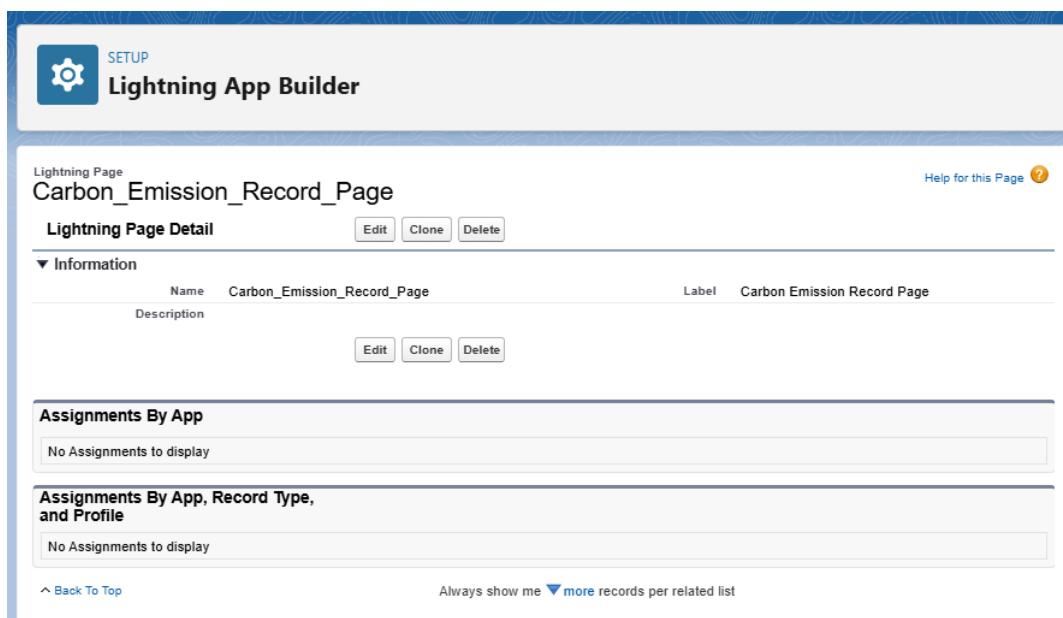


The screenshot shows the App Manager interface with the following tabs: File, Edit, Debug, Test, Workspace, Help, <, >. The active tab is CarbonEmissionUtils.apxc. Below the tabs, there are dropdowns for Code Coverage (None) and API Version (64). The main area contains the following Java code:

```
1 public class CarbonEmissionUtils {  
2  
3     // A method to convert different units to kilograms  
4     public static Decimal convertToKilograms(Decimal amount, String unit) {  
5         if (unit == 'tons') {  
6             return amount * 1000; // Convert tons to kg  
7         } else if (unit == 'pounds') {  
8             return amount * 0.453592; // Convert pounds to kg  
9         } else {  
10            return amount; // Assume it's already kg  
11        }  
12    }  
13}
```

## 6. LWC (Lightning Web Components)

- carbonFootprintTracker component fully developed
- All component files (.js, .html, .css, .xml) created
- Business logic and user interface implemented
- Component successfully deployed and tested

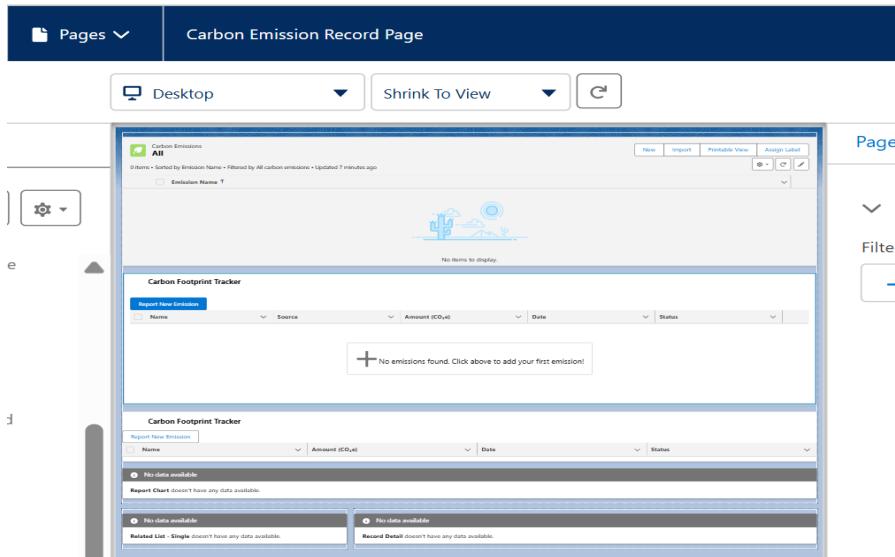


The screenshot shows the Lightning App Builder interface with the following sections:

- SETUP** button and **Lightning App Builder** title.
- Lightning Page**: **Carbon\_Emission\_Record\_Page**.  
Edit | Clone | Delete
- Information** section:

Name	Carbon_Emission_Record_Page	Label	Carbon Emission Record Page
Description			

  
Edit | Clone | Delete
- Assignments By App**: No Assignments to display.
- Assignments By App, Record Type, and Profile**: No Assignments to display.
- [^ Back To Top](#) | Always show me **more** records per related list



## 7. Apex with LWC

- CarbonEmissionController Apex class developed
- @AuraEnabled methods properly implemented
- Successful integration between Apex and LWC
- Data flow and error handling verified

```
// use the createEmission method that exists in your Apex
const result = await createEmission({
    type: this.emissionSource, // Maps to Source__c in Apex
    amount: parseFloat(this.amount),
    emissionDate: this.emissionDate,
});
```

## 8. Events in LWC

- Custom event system implemented
- Event dispatching and handling configured
- Component communication established
- Event data passing successfully tested
- **bubbles: true** → Event bubbles up to parent components.
- **composed: true** → Event crosses shadow DOM boundaries.

```
// 🔍 EVENTS IN LWC: Custom event dispatchers
dispatchEmissionCreated(emissionId) {
    this.dispatchEvent(new CustomEvent('emissioncreated', {
        detail: {
            emissionId: emissionId,
            source: this.emissionSource,
            amount: this.amount,
            timestamp: new Date().toISOString()
        },
        bubbles: true,
        composed: true
    }));
}
```

## 9. Wire Adapters

- @wire decorators properly implemented
- Reactive data loading configured
- Automatic data refresh functionality working
- Wire adapter best practices applied

```
// ✅ wire adapter for emissions data
@wire(getEmissions)
emissions;
```

## 10. Imperative Apex Calls

- Imperative Apex methods successfully implemented
- Async/await patterns properly used
- Error handling and user feedback configured
- Data modification operations verified

## 11. Navigation Service

- NavigationMixin successfully integrated
- Record page navigation implemented
- List view and object page navigation working
- External URL navigation configured

```
// Navigation methods
viewRecord(recordId) {
    this[NavigationMixin.Navigate]({
        type: 'standard__recordPage',
        attributes: {
            recordId: recordId,
            objectApiName: 'Carbon_Emission__c',
            actionName: 'view'
        }
    });
}
```