

# Phase 6: User Interface Development (Revaa)

## Objective

To design and configure an intuitive and functional user interface in Salesforce for Revaa, enabling Admins and Investors to access property, transaction, and portfolio data easily. This phase focuses on Lightning App Builder, Record Pages, Tabs, LWC, and navigation enhancements.

The screenshot shows the 'App Settings' page in Salesforce. The left sidebar has a blue header 'App Settings' and a section 'App Details & Branding' which is selected. Below it are 'App Options', 'Utility Items (Desktop Only)', 'Navigation Items', and 'User Profiles'. The main content area is titled 'App Details & Branding' with the sub-section 'App Details'. It contains fields for 'App Name' (Revaa), 'Developer Name' (Aniket\_Kakade), and 'Description' (Smart Fractional Real Estate Investment). To the right is the 'App Branding' section, which includes an 'Image' field showing the Revaa logo, a 'Primary Color Hex Value' field (#72BCFD), and an 'Org Theme Options' checkbox (checked) with the label 'Use the app's image and color instead of the org's custom theme'. At the bottom is the 'App Launcher Preview' section, which shows a preview of the app icon and name.

## 1 Lightning App Builder

**Purpose:** Create apps, pages, and customize layouts to match user roles and workflows.

### Steps:

1. Go to **Setup → Lightning App Builder**.
2. Click **New → Select App Page / Record Page / Home Page** depending on requirement.
3. Enter a **Label:** Revaa Dashboard App
4. Choose **Page Template** → e.g., One Region or Header & Two Columns.

5. Drag components from **Components Panel** onto the canvas (e.g., **Record Detail**, **List View**, **Custom LWC**)
6. Click **Save** → **Activate** → Assign to **Org Default / App Default / App, Record Type, Profile**.

### Screenshot:

The screenshot shows the 'Lightning App Builder' interface under the 'SETUP' tab. The main title is 'Lightning App Builder'. Below it, 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.' A navigation bar at the top includes 'View: All' and 'Create New View'. A search bar below the navigation bar has the placeholder 'A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | Other | All'. The main content area is titled 'Lightning Pages' and contains a table with the following data:

Action	Label	Name	Namespace Prefix	Description	Type	Created By	Last Modified By
Edit   Clone   Del	Lightning Record Page	Lightning_Record_Page			Record Page	ani, 9/28/2025, 2:10 PM	ani, 9/28/2025, 2:10 PM
Edit   Clone   Del	Property Record Page	Property_Record_Page			Record Page	ani, 9/28/2025, 2:23 PM	ani, 9/28/2025, 3:06 PM
Edit   Clone   Del	Revaa Home Page	Revaa_Home_Page			Home Page	ani, 9/28/2025, 2:30 PM	ani, 9/28/2025, 3:06 PM
Edit   Clone   Del	Transaction Record Page	Transaction_Record_Page			Record Page	ani, 9/28/2025, 2:18 PM	ani, 9/28/2025, 2:19 PM

The screenshot shows the 'Revaa Home Page' component editor in the Lightning App Builder. The top navigation bar includes 'Lightning App Builder', 'Pages', 'Revaa Home Page', 'Activation...', and 'Save'. The left sidebar is titled 'Components' and lists 'Standard (43)' components: Accordion, App Launcher, Assistant, Cdp Metrics Overview, Chatter Feed, Chatter Publisher, CRM Analytics Collection, CRM Analytics Dashboard, Dashboard, Data Mask Console Home Component, Einstein Next Best Action, Flow, Flow App Home cards, Generate Batch Documents, and Inventory Lookup Component. Below this is a link 'Get more on the AppExchange'. The main workspace displays a 'Key Deals - Recent Opportunities' component. The right panel is titled 'Page' and contains the following configuration fields:

- Page**
  - \*Label:** Revaa Home Page
  - \*API Name:** Revaa\_Home\_Page
  - \*Page Type:** Home Page
  - Template:** Home page header two colu... (with a 'Change' button)
- Description:** (empty text area)

## 2 Record Pages

**Purpose:** Customize layouts for Property, Investor, and Transaction records.

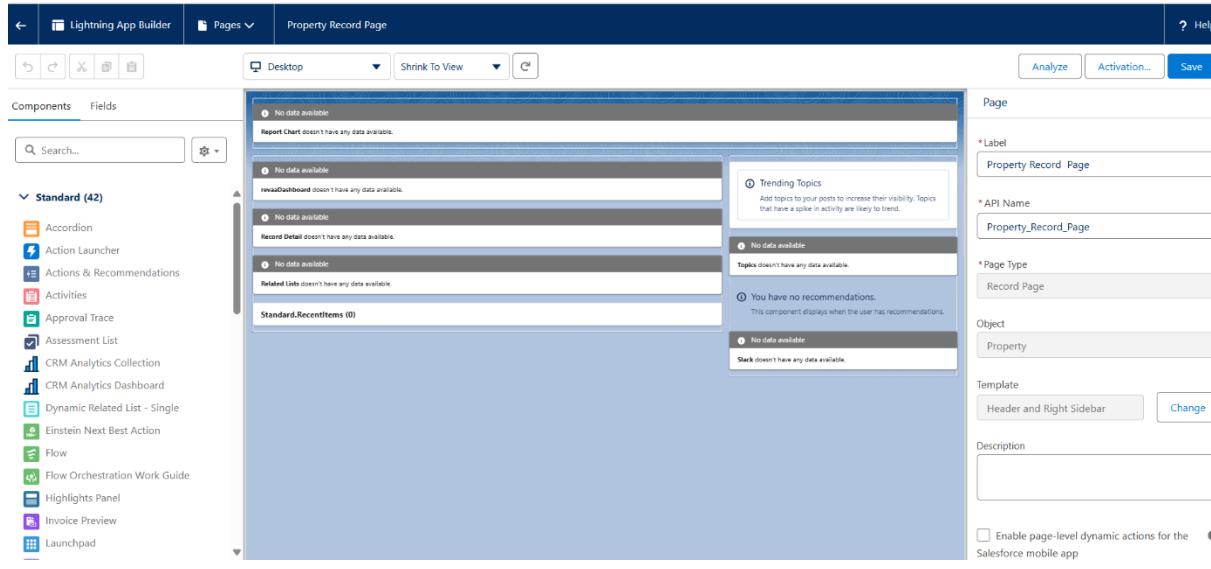
**Steps:**

1. Navigate to **Setup** → **Object Manager** → [Select Object, e.g., **Property**] → **Lightning Record Pages**.
2. Click **New** → Choose **Record Page** → Name it **Property Record Page**.
3. Select **Template** → Drag the following components:

- **Record Detail** (to show Property fields)
- **Related Lists** (Transactions, Investment Lots, Profit Distribution)
- **Custom LWC** (RevaaDashboard for visual insights)

#### 4. Save & Activate → Assign for App, Profile, Record Type.

##### Screenshot:



## 3 Tabs

**Purpose:** Make navigation easier across custom objects and reports.

##### Steps:

1. Setup → **App Manager** → Open Revaa Dashboard App → **Edit**
2. Add **Tabs**:
  - Property
  - Investor
  - Transaction
  - Investment Lot
  - Profit Distribution
  - Reports & Dashboards
3. Arrange tabs in logical order.
4. Save & Finish.

##### Screenshot:

## Custom Tabs

Help [f](#)

You can create new custom tabs to extend Salesforce functionality or to build new application functionality.

Custom Object tabs look and behave like the standard tabs provided with Salesforce. Web tabs allow you to embed external web applications and content within the Salesforce window. Visualforce tabs allow you to embed Visualforce pages. Lightning Component tabs allow you to add Lightning components to the navigation menu in Lightning Experience and the mobile app. Lightning Page tabs allow you to add Lightning Pages Experience and the mobile app.

Custom Object Tabs		
Action	Label	Tab Style
Edit   Del	<a href="#">Investment Lots</a>	Bell
Edit   Del	<a href="#">Investors</a>	People
Edit   Del	<a href="#">Profit Distributions</a>	Stack of Cash
Edit   Del	<a href="#">Properties</a>	Building
Edit   Del	<a href="#">Transactions</a>	Safe

The screenshot shows the Lightning App Builder interface. At the top, there's a dark header bar with navigation icons and the text "Lightning App Builder", "App Settings", "Pages", and "Revaa". Below this is a sidebar with sections for "App Settings" (App Details & Branding, App Options, Utility Items (Desktop Only)) and "Navigation Items" (User Profiles). The main area is titled "Navigation Items" and contains two lists: "Available Items" and "Selected Items". The "Available Items" list includes various Salesforce objects and features like Activation Targets, Activations, All Sites, Alternative Payment Methods, Analytics, App Launcher, Appointment Categories, Appointment Invitations, Approval Requests, Approval Submission Details, and Approval Submissions. The "Selected Items" list contains "Accounts", "Contacts", "Reports", "Dashboards", "Investment Lots", "Investors", "Properties", "Transactions", and "Profit Distributions". Arrows between the two lists indicate the process of dragging items from the available list to the selected list.

## 4 Home Page Layouts

**Purpose:** Provide quick insights, metrics, and shortcuts.

**Steps:**

1. Setup → **Lightning App Builder** → **Home Page** → New
2. Drag components:
  - o **Dashboard** (Investor Portfolio / Property Investment Status)
  - o **Report Chart** (Profit Distribution)
  - o **Custom LWC** (quick investment actions or stats)

### 3. Activate → Assign to **Org Default / Profiles**

---

## 5 Utility Bar

**Purpose:** Add quick actions accessible from any page.

**Steps:**

1. Setup → **App Manager** → **Reva Dashboard App** → **Edit** → **Utility Bar**
2. Add items:
  - **Quick Create Transaction**
  - **Search Investor**
  - **Notifications / Alerts**
3. Configure width, icon, and label for each item.
4. Save → Done.

**Screenshot:**

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

- Header:** Includes "Lightning App Builder", "App Settings", "Pages", and "Revaa".
- Left Sidebar (App Settings):**
  - Utility Items (Desktop Only)** (selected)
  - Navigation Items
  - User Profiles
- Utility Items (Desktop Only) Screen:**
  - Section Header:** Utility Items (Desktop Only)  
Give your users quick access to productivity tools and add background utility items to your app.
  - Add Utility Item** button
  - Utility Bar Alignment:** Default
  - History Item:** To Do List, Notes, Chatter Feed
  - Properties Panel:**
    - History** (Label: History, Icon: clock, Panel Width: 340, Panel Height: 480, Start automatically checked)

## 6 Lightning Web Components (LWC)

**Purpose:** Add custom UI functionality like dashboards, charts, and interactive elements.

**Steps:**

1. Create LWC in VS Code:
2. `sfdx force:lightning:component:create --type lwc --componentname RevaaDashboard --outputdir force-app/main/default/lwc`
3. Files created:
  - o `RevaaDashboard.html` → UI template
  - o `RevaaDashboard.js` → Logic, Apex calls, and data handling
  - o `RevaaDashboard.js-meta.xml` → Metadata configuration (exposure to record pages)
4. Example configuration in `js-meta.xml`:
5. `<targets>`
6.   `<target>lightning__RecordPage</target>`
7.   `<target>lightning__AppPage</target>`
8.   `<target>lightning__HomePage</target>`
9. `</targets>`
10. Deploy LWC to Salesforce:
  - o Right-click component folder → **SFDX: Deploy Source to Org**
11. Add LWC to record page or app page via **Lightning App Builder**.

## Screenshot:

```
force-app > main > default > lwc > revaaDashboard > JS revaaDashboard.js U ...
1 import { LightningElement, api, wire, track } from 'lwc';
2 import { getRecord, updateRecord } from 'lightning/uiRecordApi';
3 import PRICE_FIELD from '@salesforce/schema/Property__c.Price__c';
4 import AVAILABLE_LOTS_FIELD from '@salesforce/schema/Property__c.Available_Lots__c';
5 import ID_FIELD from '@salesforce/schema/Property__c.Id';
6
7 export default class RevaaDashboard extends LightningElement {
8     @api recordId;
9     @track message = '';
10
11     @wire(getRecord, { recordId: '$recordId', fields: [PRICE_FIELD, AVAILABLE_LOTS_FIELD] })
12     property;
13
14     handlePurchase() {
15         if(this.property.data.fields.Available_Lots__c.value > 0){
16             const fields = {};
17             fields[ID_FIELD.fieldApiName] = this.recordId;
18             fields[AVAILABLE_LOTS_FIELD.fieldApiName] = this.property.data.fields.Available_Lots__c.value - 1;
19
20             updateRecord({ fields })
21             .then(() => {
22                 this.message = 'Lot Purchased Successfully!';
23             })
24         }
25     }
26
27     updateRecord(fields) {
28         return new Promise((resolve, reject) => {
29             updateRecord(fields)
30                 .then(resolve)
31                 .catch(reject);
32         });
33     }
34
35     @track lotsPurchased = 0;
36
37     @track message;
38 }

```

TERMINAL

```
PS C:\Users\anike\Revaa> cd force-app/main/default/lwc/RevaaDashboard/
PS C:\Users\anike\Revaa> sf org login web -a RevaaDev
Successfully authorized aniketkakade030549e@agentforce.com with org ID 00Dg00000BHzKBUAL
PS C:\Users\anike\Revaa> sf org open -u RevaaDev
Warning: The "-u" flag has been deprecated. Use "--target-org | -o" instead.
Opening org 00Dg00000BHzKBUAL as user aniketkakade030549e@agentforce.com
Waiting to resolve the Lightning Experience-enabled custom domain.... done
PS C:\Users\anike\Revaa>
```

## 7 Apex Integration with LWC

**Purpose:** Fetch or manipulate Salesforce data dynamically in LWC.

**Steps:**

Create Apex class:

```
public with sharing class RevaaDashboardController {
    @AuraEnabled(cacheable=true)
    public static List<Transaction__c> getTransactionsByInvestor(Id investorId){
        return [SELECT Id, Lots_Purchased__c, Property__r.Name FROM Transaction__c
WHERE Investor__c=:investorId];
    }
}
```

Import Apex in LWC JS:

```
import getTransactionsByInvestor from
'@salesforce/apex/RevaaDashboardController.getTransactionsByInvestor';
```

**Screenshot:** {Apex class in Salesforce Developer Console or VS Code}

## 8 Events in LWC

**Purpose:** Handle communication between components (child → parent).

**Example:**

```
// Dispatch event on lot selection
```

```
const selectedEvent = new CustomEvent('lotselect', { detail: this.selectedLot });

this.dispatchEvent(selectedEvent);
```

**Screenshot:** {Event usage code snippet in LWC JS file}

---

## 9 Wire Adapters & Imperative Apex Calls

**Wire Adapter Example:**

```
@wire(getTransactionsByInvestor, { investorId: '$recordId' })

transactions;
```

**Imperative Apex Call Example:**

```
getTransactionsByInvestor({ investorId: this.recordId })

.then(result => { this.transactions = result; })

.catch(error => { console.error(error); });
```

**Screenshot:** {Wire vs Imperative calls in LWC}

---

## 10 Navigation Service

**Purpose:** Enable navigation between objects, record pages, or URLs from LWC.

**Example:**

```
import { NavigationMixin } from 'lightning/navigation';
```

```
export default class RevaaDashboard extends NavigationMixin(LightningElement) {

    navigateToPropertyPage(propertyId){
        this[NavigationMixin.Navigate]({
            type: 'standard__recordPage',
            attributes: {
```

```
        recordId: propertyId,  
        objectApiName: 'Property__c',  
        actionName: 'view'  
    }  
});  
}  
}
```

---

## Expected Outcomes of Phase 6

- Intuitive, user-friendly interface for Admins and Investors.
  - LWC dashboards show real-time investment and profit data.
  - Custom record pages provide all related lists and quick actions.
  - Lightning App and utility bar enable efficient navigation and operations.
-