



Salesforce PropertyHub

By:- Sushmita Katariya

Phase 6 – User Interface Development

Lightning App Builder

Purpose: Build and customize Salesforce pages with drag-and-drop ease.

Key Features:

- Supports App Pages, Record Pages, and Home Pages.
- Allows placement of standard and custom components (LWCs, charts, related lists).
- Dynamic visibility for components based on user roles or record types.

Business Impact:

- Accelerates deployment of UI changes.
- Reduces dependency on developers for non-technical modifications.

A screenshot of the Lightning App Builder interface. The top navigation bar includes back, forward, search, and help buttons, along with tabs for "Lightning App Builder", "Pages", and "Real Estate Dashboard". Below the navigation is a toolbar with icons for desktop view, shrink-to-view, and refresh. The main area is divided into two sections: "Components" on the left and "Page" configuration on the right. The "Components" section shows a list of available components like Accordion, Chatter Feed, and CRM Analytics Collection. The "Page" configuration section shows a preview of a "Property Information" page with details for a "Luxury Villa" and a "2BHK Apartment in Pune". To the right of the preview, there's a "Page" configuration panel with fields for "Label" (Real Estate Dashboard), "API Name" (Real_Estate_Dashboard), "Page Type" (App Page), "Template" (Header and Two Regions), "Description" (Property management dashboard for agents), and "Actions". A "Save" button is located at the top right of the configuration panel.

Record Pages

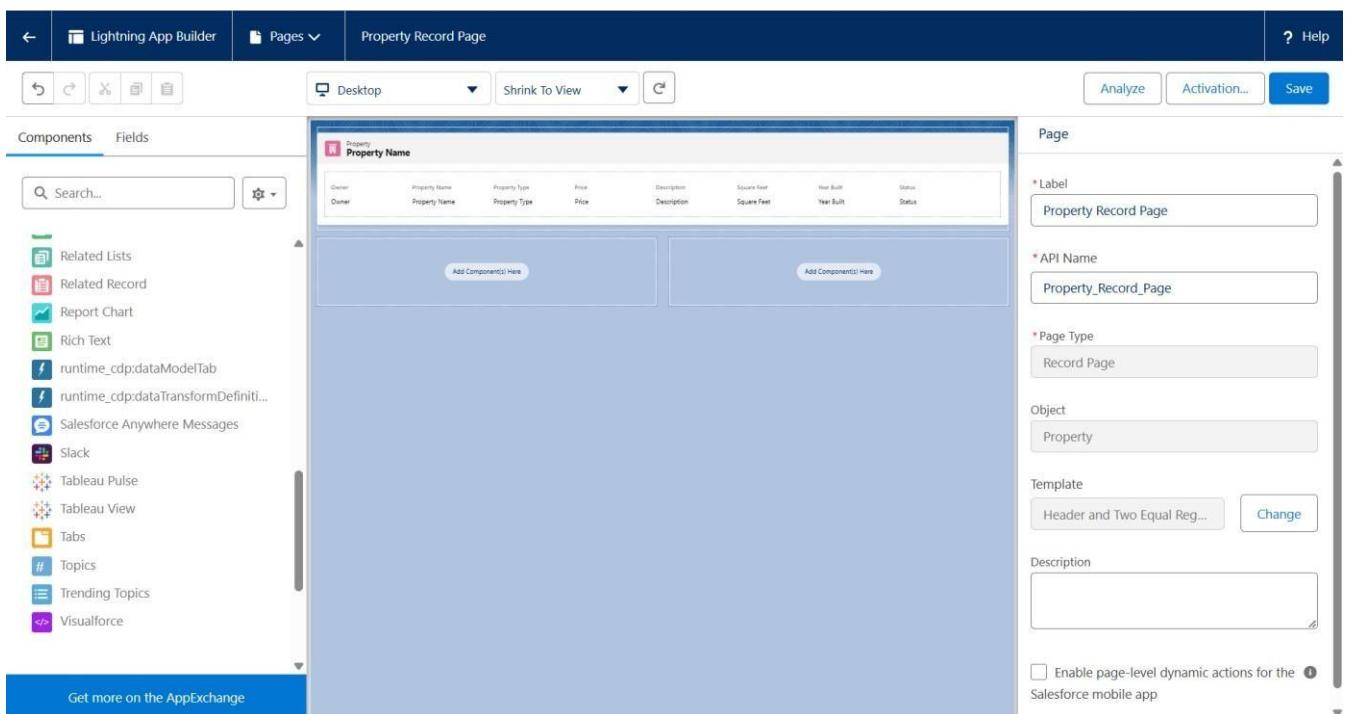
Purpose: Enhance record-level user experience by customizing layouts.

Key Features:

- Tabs, sections, related lists, and component visibility rules.
- Supports embedding LWCs for dynamic data.

Business Impact:

- Improves workflow efficiency by displaying only relevant information.
- Reduces clicks and time spent searching for data.



Tabs

Purpose: Organize and simplify navigation within Salesforce apps.

Key Features:

- Create tabs for objects, Visualforce pages, or Lightning pages.
- Supports role-based visibility and default landing tabs.

Business Impact:

- Streamlines navigation, improving user efficiency.
- Enhances adoption by providing intuitive access to key objects.

SETUP

Tabs

Custom Tabs

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 to Lightning Experience and the mobile app.

Custom Object Tabs		New	What Is This?
Action	Label	Tab Style	Description
Edit Del	Commissions		Stack of Cash
Edit Del	Properties		Building
Edit Del	Property.Visits		People

Web Tabs		New	What Is This?
No Web Tabs have been defined			

Visualforce Tabs		New	What Is This?
No Visualforce Tabs have been defined			

Lightning Component Tabs		New	What Is This?

Home Page Layouts

Purpose: Deliver centralized dashboards with actionable insights.

Key Features:

- Components: reports, charts, tasks, and notifications.
- Supports role-specific layouts.

Business Impact:

- Users access critical KPIs immediately.
- Improves decision-making by highlighting actionable data.

Lightning App Builder

Pages

Home Page

Activation... **Save**

Components

Search...

Standard (43)

- Accordion
- App Launcher
- Assistant
- Cdp Metrics Overview
- Chatter Feed
- Chatter Publisher
- CRM Analytics Collection
- CRM Analytics Dashboard
- Dashboard
- Data Mask Console Home Compo...
- Einstein Next Best Action
- Flow
- Flow App Home cards
- Generate Batch Documents

Assistant

Nothing needs your attention right now. Check back later.

Today's Events

Looks like you're free and clear the rest of the day.

Today's Events

Looks like you're free and clear the rest of the day.

Sample Flow Report: Screen Flows

We can't draw this chart because there is no data.

Page

*** Label** **Home Page**

*** API Name** **Home_Page**

*** Page Type** **Home Page**

Template **Home Template One Region** **Change**

Description

Get more on the AppExchange

Utility Bar

Purpose: Provide quick access to frequently used tools.

Key Features:

- Bottom bar with tools like recent records, calculators, chat, or reports.
- Accessible throughout the app.

Business Impact:

- Improves multitasking and operational efficiency.
- Reduces time spent switching between pages.

The screenshot shows the Salesforce Lightning App Builder interface. The top navigation bar includes 'Lightning App Builder', 'App Settings', 'Pages', 'Salesforce PropertyHub', and a 'Help' link. On the left, a sidebar titled 'App Settings' contains links for 'App Details & Branding', 'App Options', 'Utility Items (Desktop Only)' (which is selected and highlighted in blue), 'Navigation Items', and 'User Profiles'. The main content area is titled 'Utility Items (Desktop Only)' and describes how to give users quick access to productivity tools. It features a 'Recent Items' section with a list of items like 'Report Chart', 'Notes', 'To Do List', 'History', 'Phone', 'History', 'List View', and 'Flow'. To the right of this list are 'Utility Item Properties' and 'Component Properties' sections, each with various input fields and dropdown menus for configuration.

Lightning Web Components (LWC)

Purpose: Build modern, reusable, and high-performance components.

Key Features:

- Modular and lightweight.
- Reactive properties and data binding.
- Can integrate with Apex, events, and navigation services.

Business Impact:

- Enhances user experience with dynamic, interactive elements.
- Improves scalability and maintainability of UI components.

The screenshot shows the Lightning App Builder interface. On the left, there's a sidebar titled 'Components' with sections for 'Available Components' (Search bar, Filter dropdown), 'Custom (3)' (propertyCard, propertyList, scheduleVisit), and 'Custom - Managed (0)' (No components available). Below the sidebar is a button 'Get more on the AppExchange'. The main area displays a 'Property Information' card with details for a Luxury Villa (Price: \$1,00,000, Status: Available, Type: Villa) and a 'Available Properties' section showing another villa and an apartment. To the right, the 'Real Estate Dashboard' page is being configured with fields for 'Page Label' (Real Estate Dashboard), 'API Name' (Real_Estate_Dashboard), 'Page Type' (App Page), 'Template' (Header and Two Regions), 'Description' (Property management dashboard for agents), and 'Actions' (Select...).

Apex Integration with LWC

Purpose: Connect UI components with backend logic for dynamic data operations.

Key Features:

- Imperative Apex calls for user-triggered operations.
- Wire adapters for reactive data fetching.

Business Impact:

- Enables real-time, data-driven UI interactions.
- Reduces manual data handling and improves reliability.

The screenshot shows the Salesforce PropertyHub Apex Classes page. At the top, there's a navigation bar with links for Accounts, Contacts, Dashboards, Customers, Leads, People, Paused Flows, Home, and More. The main content area shows the 'Apex Class Detail' for 'PropertyService'. The class details include:

Name	PropertyService	Status	Active
Namespace Prefix		Code Coverage	0% (0/28)
Created By	Sushmita Katariya , 9/13/2025, 9:29 AM	Last Modified By	Sushmita Katariya , 9/14/2025, 12:10 PM

 Below this, there are tabs for 'Class Body', 'Class Summary', 'Version Settings', and 'Trace Flags'. The 'Class Body' tab is selected, displaying the following Apex code:


```

1 public class PropertyService {
2
3     public static List<Property__c> getAvailableProperties(String city, Decimal minPrice, Decimal maxPrice, Integer bedrooms) {
4         String query = 'SELECT Id, Name, Address__c, City__c, Price__c, Bedrooms__c, Bathrooms__c, '+
5             'Square_Foot__c, Property_Type__c, Status__c, Listing_Date__c '+
6             'FROM Property__c '+
7             'WHERE Status__c = \'Available\'';
8
9     List<String> conditions = new List<String>();
10
11    if (String.isNotBlank(city)) {
12        conditions.add('City__c = :city');
13    }
14    if (minPrice != null) {
15        conditions.add('Price__c >= :minPrice');
16    }
17    if (maxPrice != null) {
18        conditions.add('Price__c <= :maxPrice');
19    }
}
    
```

Events in LWC

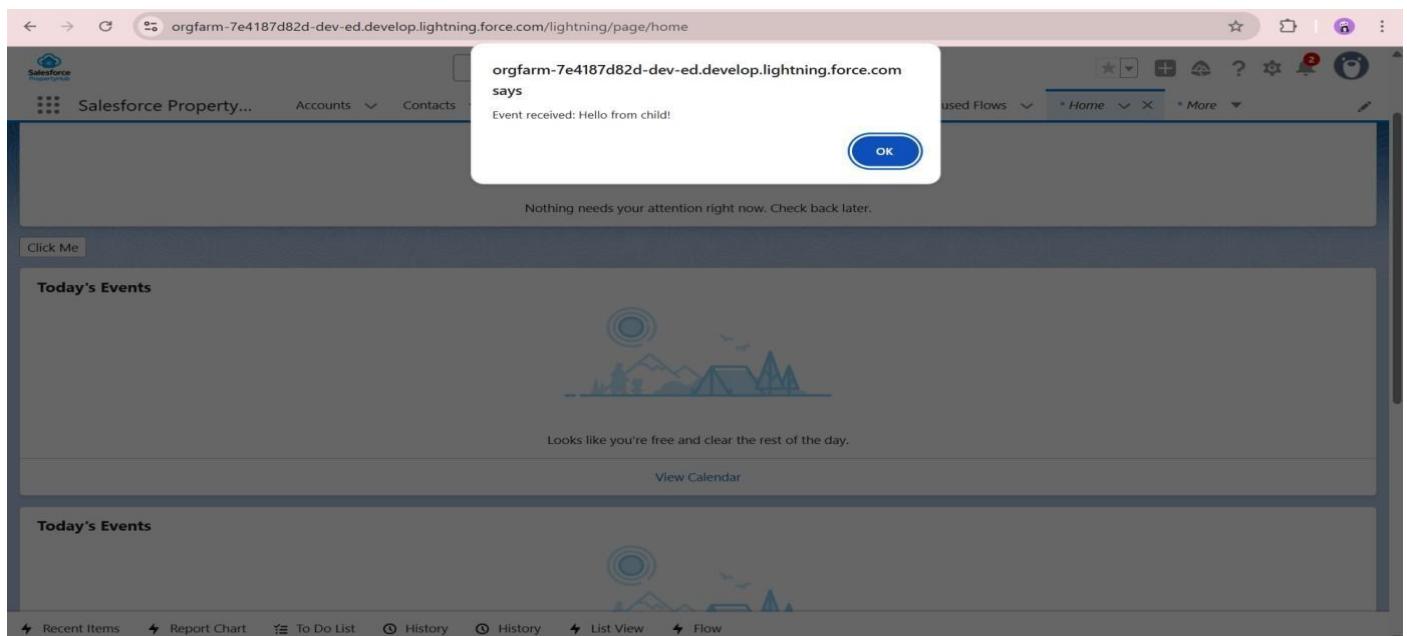
Purpose: Enable component-to-component communication.

Key Features:

- Custom events (child → parent).
- Lightning Message Service for cross-component communication.

Business Impact:

- Supports modular design.
- Allows multiple components to react to user actions dynamically.



Wire Adapters Purpose: Reactive data fetching in LWCs. Key Features:

- Automatic updates when data changes in Salesforce.
- Can connect to standard objects or Apex methods.

Business Impact:

- Reduces coding complexity.
- Ensures UI always displays fresh and accurate data.

The screenshot shows a Salesforce Lightning Web Component (LWC) interface. At the top, there's a navigation bar with links for Accounts, Contacts, Dashboards, Customers, Leads, People, Paused Flows, Home, and More. A search bar is also present. The main content area displays a list of properties under the heading "List of Properties". Each item in the list includes the property name, status (Available), and price (\$). A "Click Me" button is located at the bottom left of the list.

Property Name	Status	Price (\$)
Greenwood Villa	Available	\$750000
Sunset Apartments	Available	\$550000
Grand Hotels & Resorts Ltd	Available	\$3500000
3BHK Apartment Gurgaon	Available	\$6500000
Luxury Villa Delhi	Available	\$8500000

Imperative Apex Calls

Purpose: Execute Apex methods manually from LWC for complex operations.

Business Impact:

- Full control over backend logic.
- Enables dynamic actions triggered by users, such as property searches or updates.

Navigation Service

Purpose: Programmatically navigate between Salesforce pages.

Key Features:

- Navigate to record pages, lists, objects, or external URLs.
- Works with buttons, links, or programmatic events.

Business Impact:

- Simplifies workflow and reduces clicks.
- Enhances user efficiency and experience.

The screenshot shows the Salesforce Home page. At the top, there is a search bar and a navigation bar with links for Accounts, Contacts, Dashboards, Customers, Leads, People, Paused Flows, Home, More, and a user icon. Below the navigation bar, there is a section titled "Navigation Example" with two buttons: "Go to Record Page" and "Go to Salesforce Website". The main content area features a section titled "Today's Events" with a blue background and a small graphic of a sun and mountains. Below this, a message says "Looks like you're free and clear the rest of the day." and a "View Calendar" button. Further down, there is a section titled "Sample Flow Report: Screen Flows". At the bottom of the page, there is a footer with links for Recent Items, Report Chart, To Do List, History, List View, and Flow.

→ **Next Steps (Phase 7 – Integration & External Access)**

1. **External System Integration** – Configure Named Credentials & Remote Site Settings to securely connect Salesforce with APIs (e.g., Twilio, Google, property listing platforms).
2. **API Callouts & Services** – Build Apex callouts and register external services for property sync and SMS notifications.
3. **Real-time Connectivity** – Implement Platform Events, Change Data Capture, and Salesforce Connect to exchange data with portals/websites instantly.