Phase 6: User Interface Development

1. Lightning App Builder

- Create a **custom Lightning App** called **Smart Property Portal**.

- Include key navigation tabs: **Leads, Properties, Visits, Deals, Bookings, Dashboards**.

- Add a **Utility Bar** for quick actions:

    o "Schedule Visit" button

    o "Convert Lead to Deal" shortcut

    o "Upload Agreement" action

2. Record Pages

- **Property Record Page:**

    o Display Location, Price, Features, Status (Available/Sold).

    o Related List for **Property Visits**.

    o Quick Action: "Schedule Visit."

- **Lead Record Page:**

    o Show Lead Score, Assigned Agent, Territory.

    o Related List → Past Interactions (emails, calls).

    o Quick Action: "Convert Lead to Deal."

- **Deal Record Page:**

    o Stages: Enquiry → Negotiation → Offer → Closed Won/Lost.

    o Related Property & Booking details.

    o Quick Action: "Generate Booking."

- **Booking Page:**

    o Payment details + Uploaded Agreements.

    o Approval Status.

3. Home Page Layouts

- **Sales Agent Home:**
    - List of Today's Visits.
    - Pending Lead Follow-ups.
    - Quick property search.

- **Sales Manager Home:**
    - Sales Performance Dashboard.
    - Team Leaderboard (Top Agents by Deals Closed).
    - List of Approvals Pending.

4. Utility Bar Features

- **Quick Visit Scheduler:** Opens a screen flow to book a property visit.

- **Lead Qualification Panel:** Shows real-time lead scoring.

- **Document Upload:** Allows managers/agents to attach property brochures or agreements.

5. Lightning Web Components (LWC)

Property Search Component

- Filters by: Price, Location, Property Type.

- Displays results in a grid view with images & key fields.

- Agents can click a property → open Property Record.

Visit Calendar Component

- Calendar view of all scheduled visits.

- Color-coded: Pending (blue), Completed (green), Missed (red).

- Drag & Drop rescheduling for agents.

Deal Progress Tracker

- Visual Kanban for deals → move from Enquiry → Negotiation → Offer → Closure.

- Shows % probability of closing at each stage.

Booking Document Upload

- Allows uploading of booking agreements.

- Uses Salesforce Files API to attach securely to records.

6. Apex with LWC Integration

- **Wire Adapters:** Fetch property records dynamically.

- **Imperative Apex Calls:** Query available properties in real time.

- **Events in LWC:** Fire events when a visit is booked → notify other components (like calendar).

- **Navigation Service:** Redirect users from search results → property record page.

7. Events in LWC

Events are used for **communication between components**.

- **Use Case 1 – Property Visit Booking**

  - When a user books a property visit in one LWC, a **custom event** fires → the Visit Calendar component updates automatically.

- **Use Case 2 – Deal Stage Update**

  - Dragging a deal card from "Negotiation" → "Offer" stage triggers an event → refreshes the Deal Progress Tracker.

8. Wire Adapters

Wire Adapters fetch Salesforce data declaratively.

- **Use Case – Property Search**

  - Use @wire(getRecord, { recordId: '$recordId', fields: [...] }) to fetch property details dynamically.

  - Example: Load **Price, Location, Status** whenever an agent opens a property record.

9. Imperative Apex Calls

Imperative calls are used when we need **on-demand queries or actions**.

- **Use Case – Advanced Property Search**

    o When an agent enters a price range & location → make an Apex call:

    o searchProperties({ minPrice: 2000000, maxPrice: 5000000, city: 'Hyderabad' })

    o .then(result => { this.propertyList = result; })

    o .catch(error => { this.error = error; });

    o Returns filtered property list → displayed instantly in LWC grid.

- **Use Case – Visit Confirmation**

    o On clicking "Confirm Visit," call an Apex method to insert a **Property Visit record** in Salesforce.

10. Navigation Service

Allows smooth navigation between Salesforce pages or records.

- **Use Case – Property Search Results → Property Record**

    o After selecting a property, navigate directly to its record page:

    o this[NavigationMixin.Navigate]({

    o   type: 'standard__recordPage',

    o   attributes: {

    o   recordId: propertyId,

    o   objectApiName: 'Property__c',

    o   actionName: 'view'

    o   }

    o });

- **Use Case – Deal Tracker → Booking Page**

    o When a deal is closed, auto-navigate the agent to **Booking Record creation page**.