**Phase 6: User Interface Development**

1. **Lightning App Builder**

The Lightning App Builder is a point-and-click tool in Salesforce used to create custom pages for both Lightning Experience and the Salesforce mobile app. It allows administrators and developers to drag and drop components onto pages, configure their properties, and design highly customized layouts without needing to write code.

1. **Record Pages**

Record Pages are customized layouts in Salesforce that display the details of a single record, such as an Account, Contact, or Opportunity. They allow developers to add standard, custom, or Lightning components to enhance the user experience by showing only the most relevant information for each user profile or business scenario.

1. **Tabs**

Tabs are navigation elements that provide users with quick access to different objects, web pages, or custom Lightning components within Salesforce. They improve usability by organizing information and ensuring that frequently used objects (like Accounts or Leads) are easily accessible.

1. **Home Page Layouts**

The Home Page Layout defines what users see when they log in to Salesforce. It can include reports, dashboards, news, tasks, and other key components. By customizing the home page layout, organizations can ensure that users see the most important insights and actions immediately, improving productivity.

1. **Utility Bar**

The Utility Bar is a fixed panel at the bottom of the Lightning Experience interface. It provides quick access to productivity tools such as Notes, History, or custom components. Since it is always visible, users can access important utilities without navigating away from their current page.

1. **Lightning Web Components (LWC)**

Lightning Web Components are Salesforce’s modern programming model for building web interfaces. Based on core web standards (JavaScript, HTML, and CSS), LWCs are lightweight, fast, and reusable across applications. They enable developers to create highly interactive and dynamic user experiences within Salesforce.

1. **Apex with LWC**

LWC can communicate with the server-side business logic written in Apex. By calling Apex methods, LWCs can perform operations such as retrieving records, updating data, or executing custom logic. This integration ensures a seamless connection between the frontend UI and backend Salesforce data.

1. **Events in LWC**

Events in LWCs are used for communication between components. For example, a child component can send data to its parent using custom events. This event-driven architecture ensures that components remain modular, reusable, and loosely coupled while still being able to interact effectively.

1. **Wire Adapters**

Wire Adapters are a declarative way to read Salesforce data in LWCs. By using the @wire decorator, developers can automatically retrieve Salesforce records, lists, or metadata without writing explicit Apex code. Wire adapters are reactive, meaning data updates automatically when changes occur.

1. **Imperative Apex Calls**

Imperative Apex Calls are programmatic calls made from LWCs to Apex methods. Unlike wire adapters, imperative calls provide greater flexibility by allowing developers to pass parameters dynamically and handle logic conditionally. They are commonly used for scenarios requiring complex processing or on-demand data retrieval.

1. **Navigation Service**

The Navigation Service in Lightning allows LWCs to navigate users to different pages within Salesforce, such as record pages, list views, or external URLs. Instead of hardcoding links, developers use the Navigation Service API to provide a consistent and seamless navigation experience across the platform.