

## Phase 6 – User Interface Development (Admin & Developer)

---

### 1. Lightning App Builder

Use Case: Create a custom app for shipment tracking with tabs, record pages, and utilities for logistics and customer teams.

Explanation: Lightning App Builder allows you to combine standard components, custom components, and flows into a unified interface.

Steps:

1. Setup → App Builder → New Lightning App
  2. Name: ShipmentTrack
  3. Include standard navigation items (Shipments, Inventory, Customers)
  4. Assign to profiles (Logistics Manager, Customer Service)
- Screenshot: Lightning App Builder canvas with ShipmentTrack app layout.
- 

### 2. Record Pages

Use Case: Customize Shipment record layout for clarity and efficiency.

Explanation: Create separate layouts for different profiles, including shipment details, customer info, feedback flows, and approval history.

Steps:

1. Setup → Object Manager → Shipment → Lightning Record Pages → New
  2. Add components: Details, Related Lists, Flows, Approval History
  3. Activate → Assign to profiles
- Screenshot: Record page editor with components arranged.
- 

### 3. Tabs

Use Case: Easy access to key objects like Shipments, Inventory, Customers, Warehouses.

Steps:

1. Setup → Tabs → New Tab → Select Object → Add to App
  2. Name tab (e.g., Shipments)
- Screenshot: App with new tabs in navigation menu.
- 

### 4. Home Page Layouts

Use Case: Dashboard for users to see shipment status, pending approvals, and alerts.

Steps:

1. Setup → Home Pages → New Lightning Home Page

2. Add components: List Views, Reports, Charts, Custom Notifications  
Screenshot: Home page layout in Lightning App Builder.
- 

## 5. Utility Bar

Use Case: Quick access to common actions like creating shipments, searching customers, or checking inventory.

Steps:

1. Setup → App Builder → Open App → Utility Bar → Add Utilities (e.g., “Create Shipment”, “Search Shipment”)

Screenshot: Utility bar configuration panel.

---

## 6. LWC (Lightning Web Components)

Use Case: Custom interactive components for shipment tracking and feedback.

Example: shipmentFeedback component to capture rating and comments.

Steps:

1. Create LWC: VS Code or Developer Console → lwc create shipmentFeedback
2. HTML + JS + CSS for UI
3. Deploy → Add to Lightning Record Page

Screenshot: LWC in Record Page with interactive form.

---

## 7. Apex with LWC

Use Case: Fetch or update shipment records using server-side logic.

Example: Retrieve shipment details for LWC dynamically.

@AuraEnabled(cacheable=true)

```
public static Shipment__c getShipmentDetails(Id shipmentId){  
    return [SELECT Id, Name, Status__c, Customer__c FROM Shipment__c WHERE Id = :shipmentId];  
}
```

---

## 8. Events in LWC

Use Case: Communication between parent and child components.

Example: Emit a custom event when feedback is submitted:

```
const feedbackEvent = new CustomEvent('submitted', {detail: this.feedback});  
this.dispatchEvent(feedbackEvent);
```

---

## 9. Wire Adapters

Use Case: Automatically fetch Salesforce data in a reactive way.

Example:

```
import { LightningElement, wire } from 'lwc';

import getShipmentDetails from '@salesforce/apex/ShipmentController.getShipmentDetails';

export default class ShipmentDetails extends LightningElement {
  @wire(getShipmentDetails, { shipmentId: '$recordId' }) shipment;
}
```

---

## 10. Imperative Apex Calls

Use Case: Call Apex methods on button click or specific action.

```
import { LightningElement } from 'lwc';

import updateShipmentStatus from
'@salesforce/apex/ShipmentController.updateShipmentStatus';

handleClick(){
  updateShipmentStatus({ shipmentId: this.recordId, status: 'Delivered' })
    .then(result => console.log('Status updated'))
    .catch(error => console.error(error));
}
```

---

## 11. Navigation Service

Use Case: Navigate programmatically between pages, e.g., after submitting feedback.

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

export default class ShipmentNav extends NavigationMixin(LightningElement){
  navigateToShipment(){
    this[NavigationMixin.Navigate]({
      type: 'standard__recordPage',
```

```
attributes: {  
    recordId: this.recordId,  
    objectApiName: 'Shipment__c',  
    actionName: 'view'  
}  
});  
}  
}
```