**PHASE 6 – User Interface Development**

In this phase, the focus is on designing a user-friendly interface for the Delivery, Orders, Customers, and Loyalty Program modules. Salesforce’s Lightning framework and Lightning Web Components (LWC) are used to create a modern, interactive UI.

**1. Lightning App Builder**

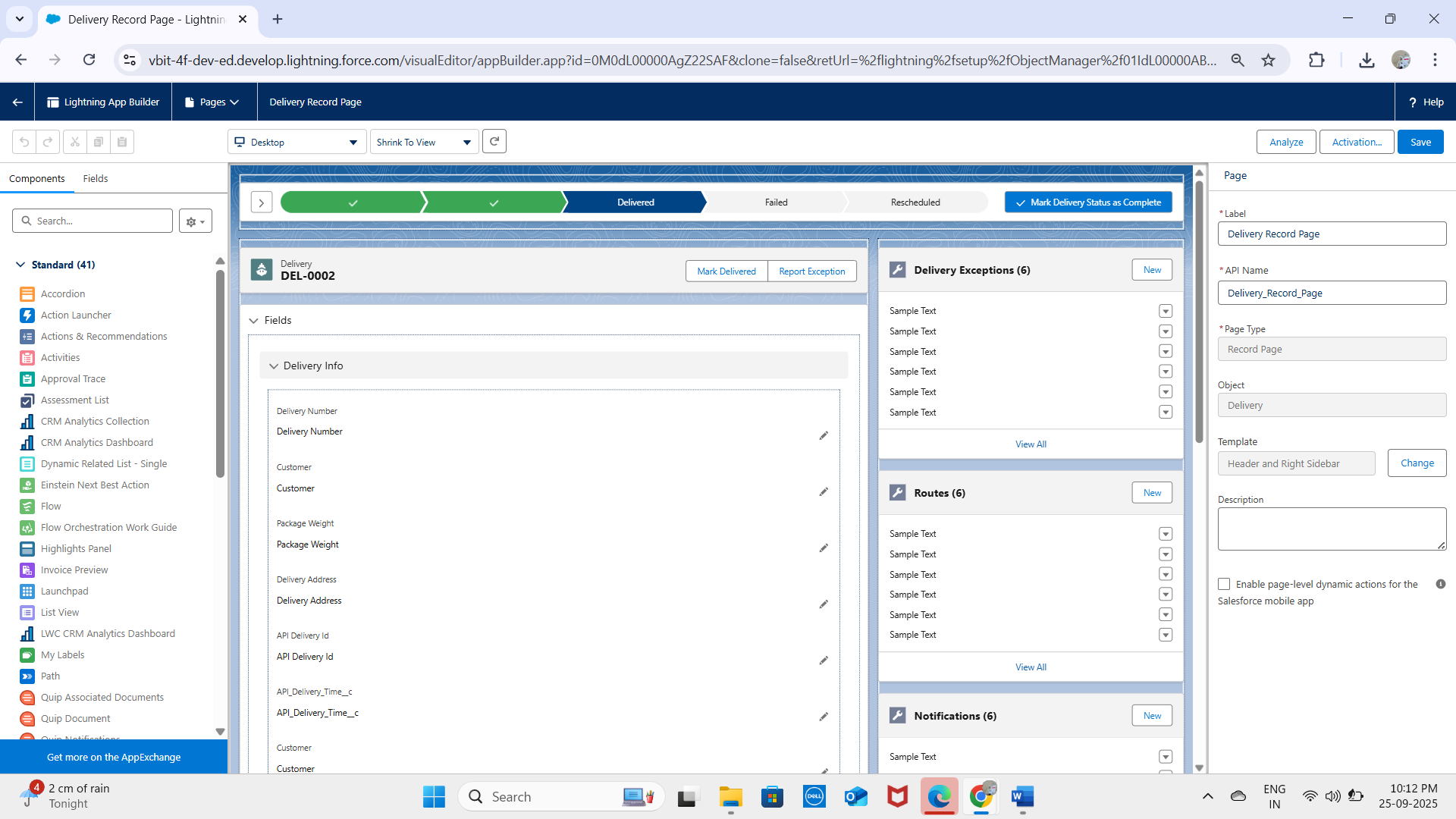
The Lightning App Builder was used to create custom pages for both Lightning Experience and the Salesforce mobile app.

* Drag-and-drop components to design pages without code.
* Example: Custom Order Management Dashboard displaying pending deliveries and loyalty points.

**2. Record Pages**

Record Pages were customized to show relevant details for each object:

* Delivery\_\_c Record Page: Shows Delivery Number, Status, Customer, and Scheduled Date.

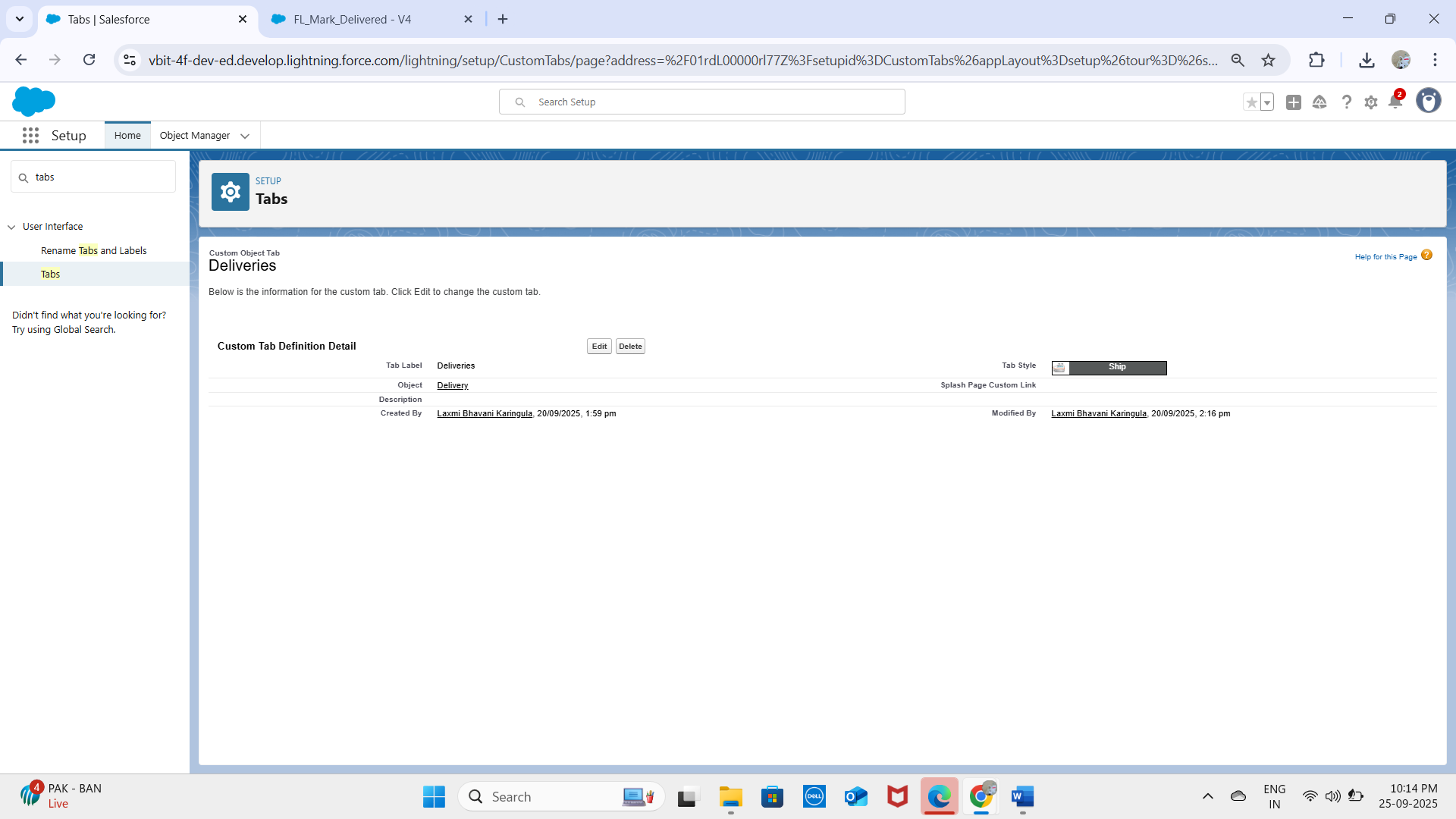


* Components like charts and related lists were added to enhance user experience.

**3. Tabs**

Custom tabs were created to improve navigation:

* Deliveries Tab – Access Delivery records.



**4. Home Page Layouts**

Custom home pages were designed for different profiles:

* Display key reports like Daily Deliveries, Pending Orders, and Loyalty Points Summary.
* Include quick links for creating new orders or deliveries.

**5. Utility Bar**

Configured the Utility Bar for quick access to:

* Notes for customer interactions.
* Recent deliveries and orders.
* Quick access to Loyalty Program components.

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

LWCs were used to create interactive UI components:

* Delivery Tracker LWC – Shows real-time delivery status.
* Order Summary LWC – Displays total orders and revenue dynamically

**7. Apex with LWC**

LWCs communicate with Apex classes to perform backend operations:

* Retrieve delivery and order records dynamically.
* Update loyalty points when orders are completed.
* Example: getCustomerLoyaltyPoints Apex method called from LWC.

**8. Events in LWC**

Custom events were implemented for component communication:

* Child-to-parent events to notify updates.
* Example: When an order is completed in Order Summary LWC, the Loyalty Points LWC is updated automatically.

**9. Wire Adapters**

Used wire adapters for reactive data fetching:

**@wire(getRecord, { recordId: '$recordId', fields: [NAME\_FIELD, STATUS\_FIELD] })**

**10. Imperative Apex Calls**

For conditional or dynamic operations, imperative Apex calls were used:

getCustomerLoyaltyPoints({ customerId: this.customerId })

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

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

* Example: Fetch loyalty points only when an order is completed.

**11. Navigation Service**

Implemented the Navigation Service to navigate users seamlessly:

* Navigate from Orders List to Order Detail Page.
* Navigate from Customer Record to Loyalty Program Page.
* Provides consistent navigation without hardcoding URLs.

**✅ Outcome:**  
The user interface for the project now provides:

* A modern, interactive Lightning Experience.
* Easy navigation between Deliveries, Orders, Customers, and Loyalty modules.
* Real-time updates and notifications for users.
* Seamless integration between frontend LWCs and backend Apex logic.