

MACK Stores Salesforce CRM Project

Phase 6: User Interface Development

Objective

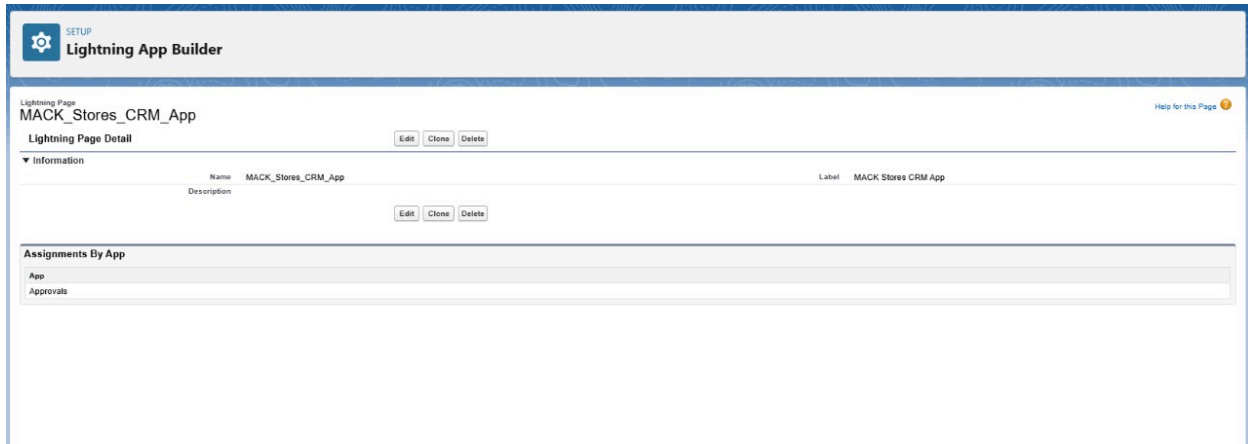
To design and implement a user-centric, responsive, and efficient UI using Salesforce Lightning Experience tools. This phase focuses on enhancing productivity, enabling seamless data access, and improving the overall user experience for Sales, Marketing, Store Managers, and Partner Users through the use of:

- Lightning App Builder (App, Record, and Home Pages)
 - Lightning Web Components (LWC)
 - Custom Tabs and Navigation
 - Utility Bar
 - Apex Integration
 - Navigation, Wire Service, and Events in LWC
-

1. Lightning App Builder – App Page

Steps :

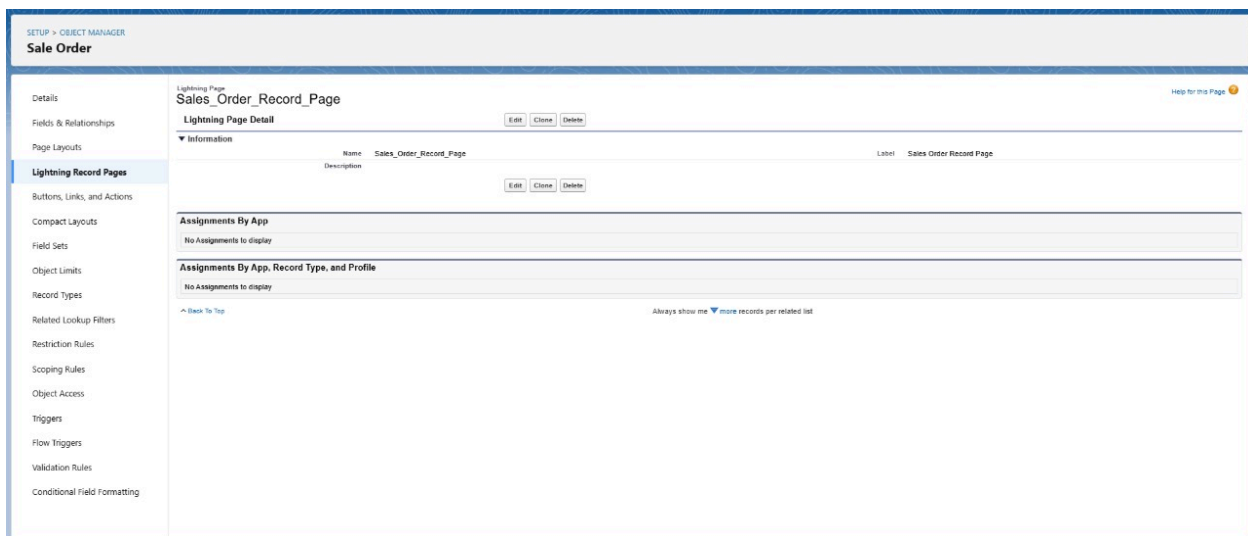
- Navigated to Setup → Lightning App Builder
- Created a new App Page:
 - Name: MACK Stores CRM App
 - Template: Three-Region Layout for modular flexibility
- Added Standard Components:
 - Reports
 - Recent Items
 - List Views (Sales Orders, Products)
- Embedded Custom LWCs:
 - **InventoryDashboard** (shows live inventory across stores)
 - **SalesOverview** (summarizes current period sales performance)
- Activated for all users



2. Record Pages – Sales_Order__c

Steps :

- Customized Lightning Record Page for **Sales_Order__c**
- Added components:
 - Highlights Panel: **Status, Total_Amount__c, Payment_Status__c**
 - Record Detail
 - Related List – Single: **Order_Items__r, Approval_History__r**
 - Custom LWC: **DiscountCalculator** (calculates dynamic discounts based on business logic)

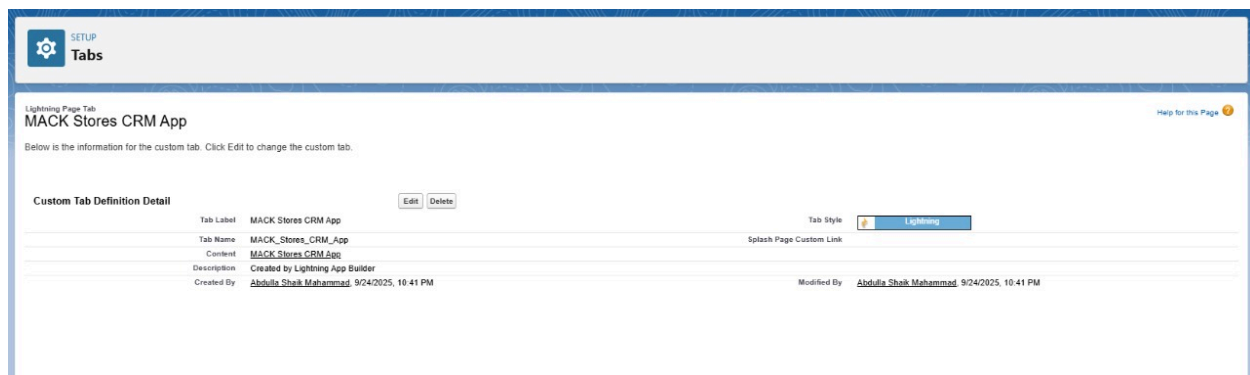


- Assigned record page to Sales Team Profile

3. Tabs & Navigation

Steps :

- Created Custom Tabs for:
 - **Loyalty_Program__c**
 - **Sales_Order__c**
 - **Product__c**
- Selected unique icons and colors for easy recognition
- Assigned tabs to:
 - **Sales**
 - **Marketing**
 - **Store Manager profiles**

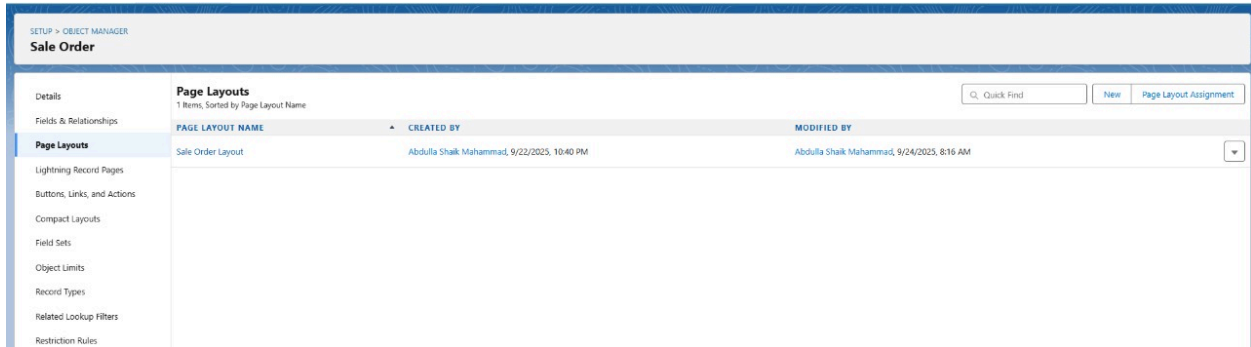


- Added these tabs to the MACK Stores CRM App Navigation Bar

4. Home Page Layouts

Steps :

- Created a custom Home Page for Sales Managers using Lightning App Builder
- Added:
 - **KPI Cards: Total Sales This Month, Open Orders**
 - **Charts/Reports: Inventory vs Demand, Top-Selling Products**
 - **Tasks & Events**
 - **Custom LWC: Top5Customers** (shows best-performing customers by sales)
- Set as default home page for Sales Manager profile



5. Utility Bar

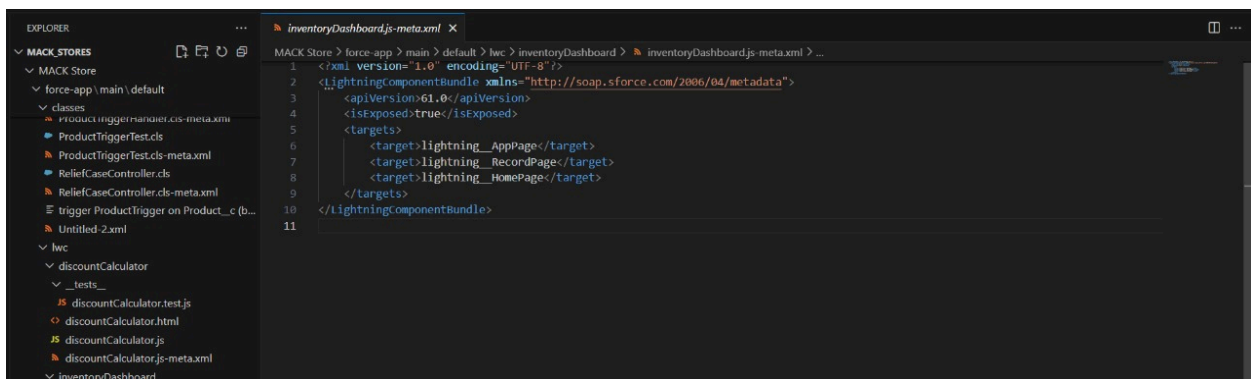
Steps :

- Edited MACK Stores CRM App via App Manager
- Added the following Utility Bar items:
 - Notes
 - History
 - Recent Items
 - QuickInventoryLookup (custom LWC)
- Set visibility for Sales and Store Managers

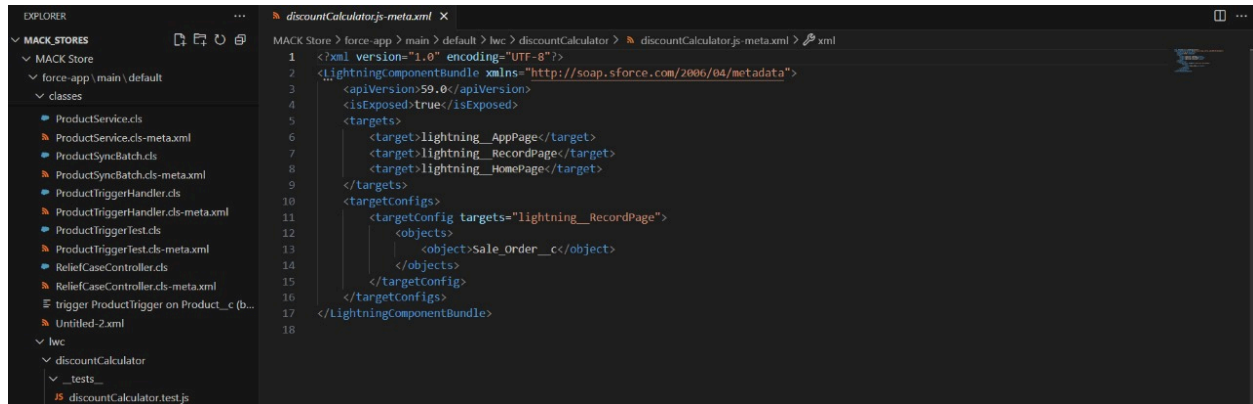
6. Lightning Web Components (LWC)

Developed LWCs:Deployment:

InventoryDashboard: Visualizes stock levels across locations



DiscountCalculator: Calculates discounts dynamically based on quantity and customer tier

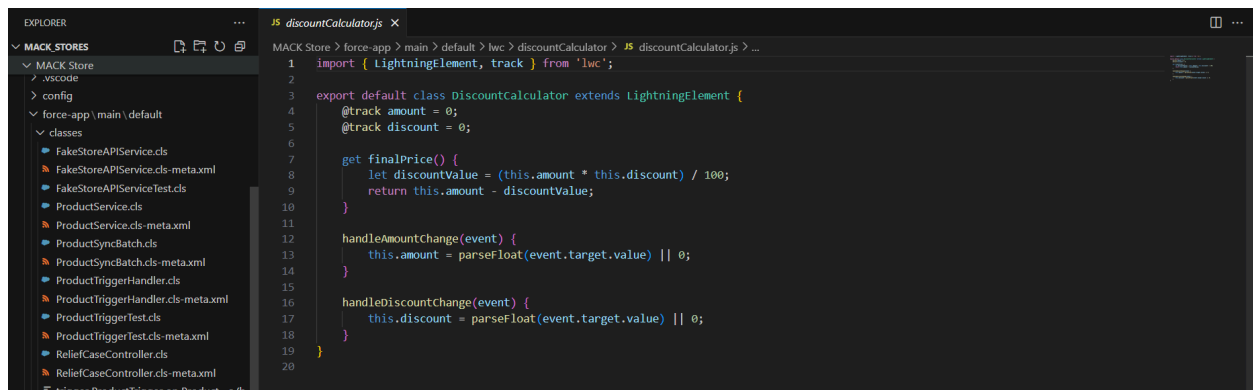


- Built in VS Code using Salesforce DX
- Exposed via **.js-meta.xml** for use in App, Record, and Home Pages
- All components follow responsive design for desktop and mobile

7. Apex Integration with LWC

Apex Classes Created:

- **InventoryController:** Fetch inventory data for products
- **LoyaltyController:** Create and manage loyalty program records
- **DiscountService:** Calculate discount based on rules

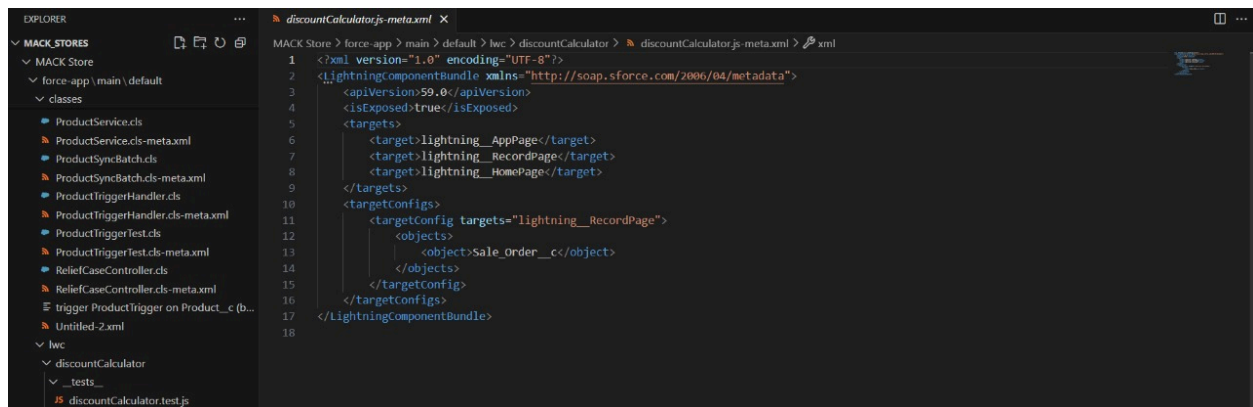


Usage Patterns:

- Wire Service for reactive UI updates (`@AuraEnabled(cacheable=true)`)
 - Imperative Calls for DML operations (e.g., create loyalty record)
 - Implemented error handling with `try-catch` and toast messages
-

8. LWC Events & Component Communication

- Used Custom Events to pass data from child to parent (e.g., `DiscountCalculator` → parent `OrderPage`)
- Used `@api`, `@track` and reactive variables for two-way binding
- Ensured event listeners are correctly set up in parent components



9. Wire Adapters

- Used `lightning/uiRecordApi` to fetch:
 - Current User Info
 - Record Details
 - Object Metadata
 - Example: `getRecord` with `USER_ID` to personalize dashboard views
 - Avoided unnecessary Apex by using wire adapters where possible
-

10. Imperative Apex Calls

Used for operations that require DML or conditional logic:

- import createLoyalty from '@salesforce/apex/LoyaltyController.createLoyalty';
 -
 - handleCreate() {
 - createLoyalty({ contactId: this.contactId })
 - .then(result => {
 - this.showToast('Success', 'Loyalty record created!', 'success');
 - })
 - .catch(error => {
 - this.showToast('Error', error.body.message, 'error');
 - });
 - }
 - All DML wrapped in server-side try/catch
 - UI shows relevant success/error messages via **ShowToastEvent**
-

11. Navigation Service

Implemented **NavigationMixin** to:

- Navigate to newly created records (Orders, Loyalty records)
 - Redirect users to specific tabs or apps
 - this[NavigationMixin.Navigate]({
 - type: 'standard__recordPage',
 - attributes: {
 - recordId: this.orderId,
 - objectApiName: 'Sales_Order__c',
 - actionName: 'view'
 - }
 - });
-

12. Testing & Validation

- Used Lightning App Builder Preview to validate UI layouts
- Tested Record and Home Pages across:
 - Sales
 - Store Manager
 - Marketing profiles
- Verified mobile responsiveness via Salesforce Mobile App

- **Used Debug Logs, Console Logs, and LWC Playground for troubleshooting**
 - **All Apex Classes and LWCs covered with unit tests and passed QA**
-

Outcome of Phase 6

- **MACK Stores CRM now offers a highly intuitive, role-based, and mobile-responsive user interface.**
 - **Users can access reports, dashboards, records, and tools in fewer clicks.**
 - **Custom LWCs and Apex services provide real-time data visibility and functionality.**
 - **Seamless navigation, event handling, and automation streamline workflows across departments.**
-