

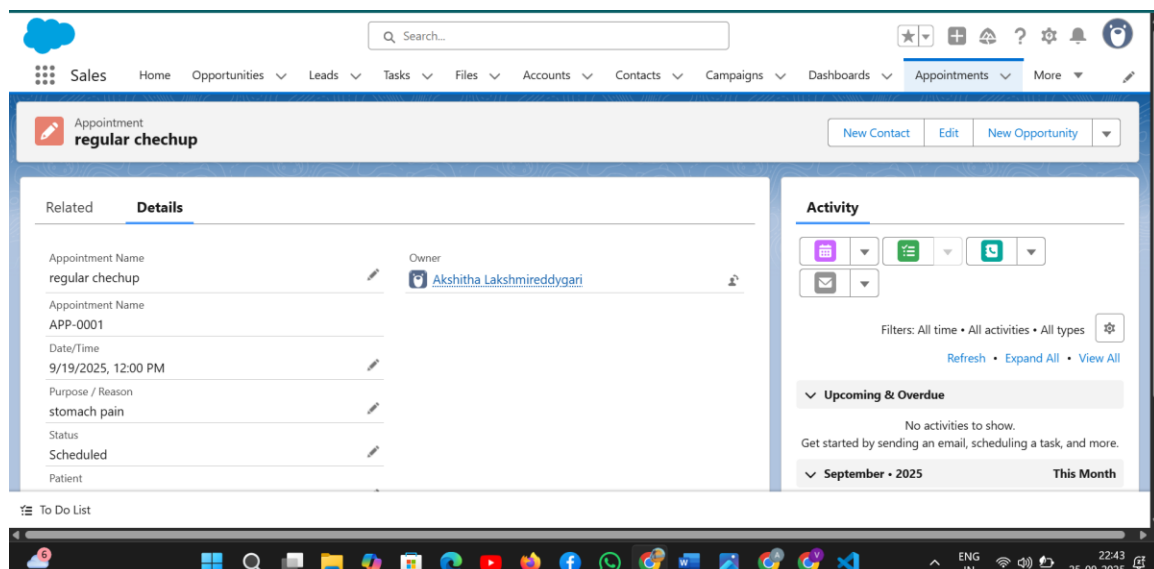
Phase 6: User Interface Development

Introduction:

This phase focuses on building an intuitive Lightning experience for different hospital roles (Doctor, Receptionist, Admin). I used Lightning App Builder, Record Pages, Tabs, Home Page Layouts, Utility Bar, a custom LWC and Apex integration to deliver a seamless UI.

1. Lightning App Builder

To provide different users with customized pages, I used Lightning App Builder to design Appointment – Doctor View and Appointment – Receptionist View record pages. Each page shows only the components relevant to that role.

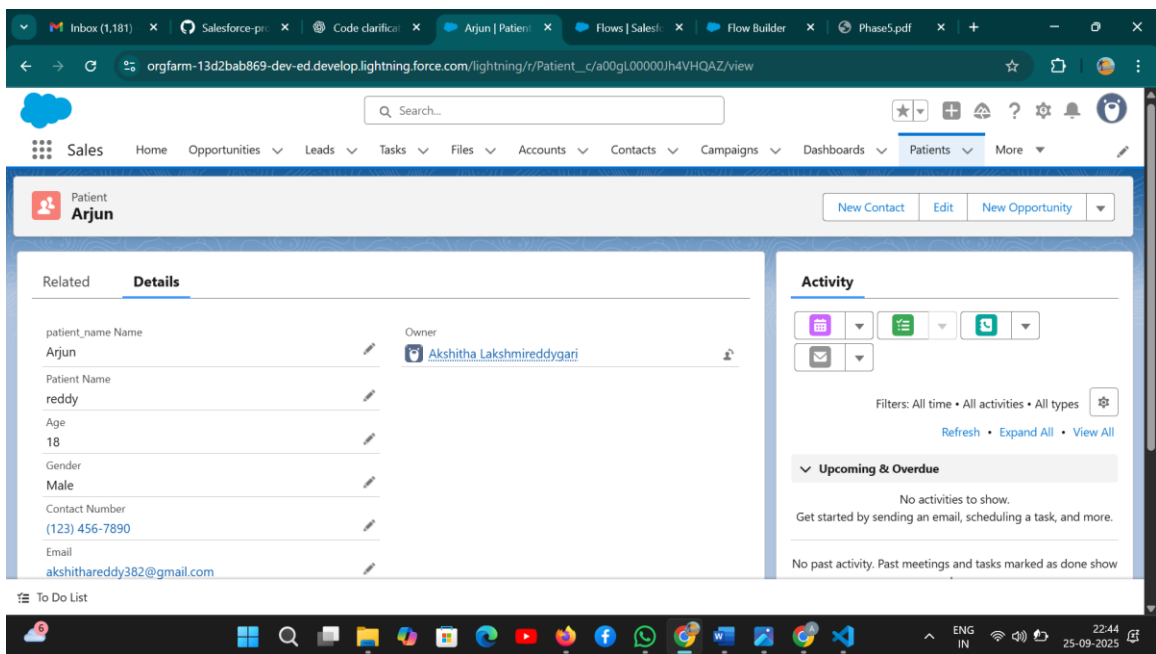
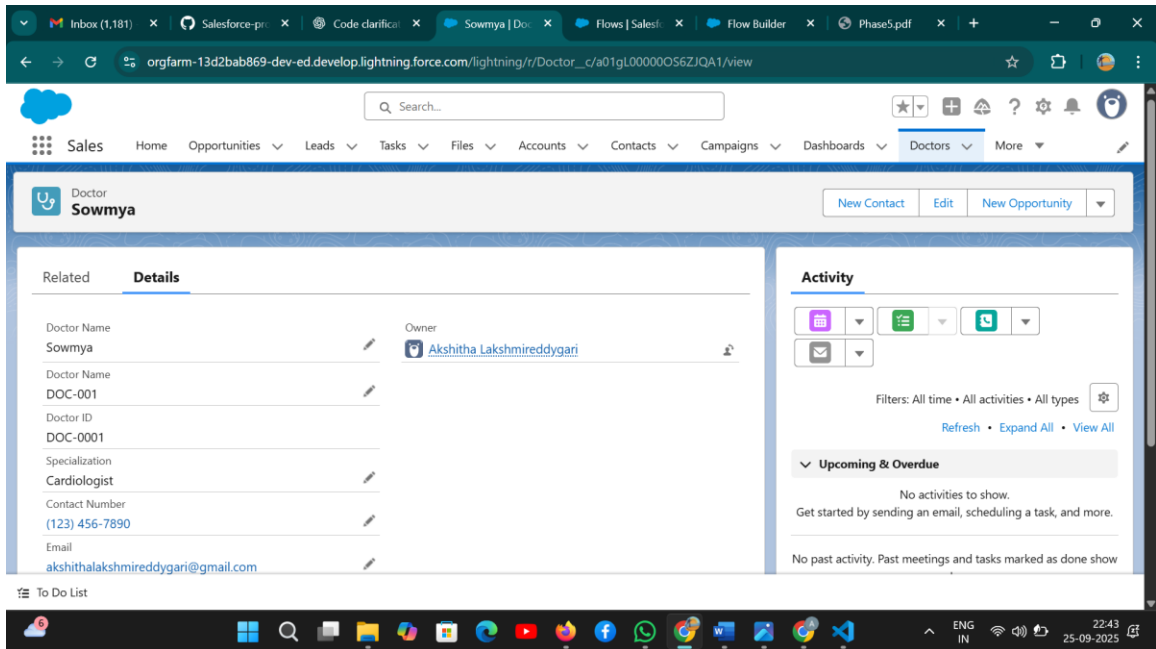


2. Record Pages (Doctor & Receptionist Views)

Doctors need quick access to upcoming appointments and related Medical Cases.

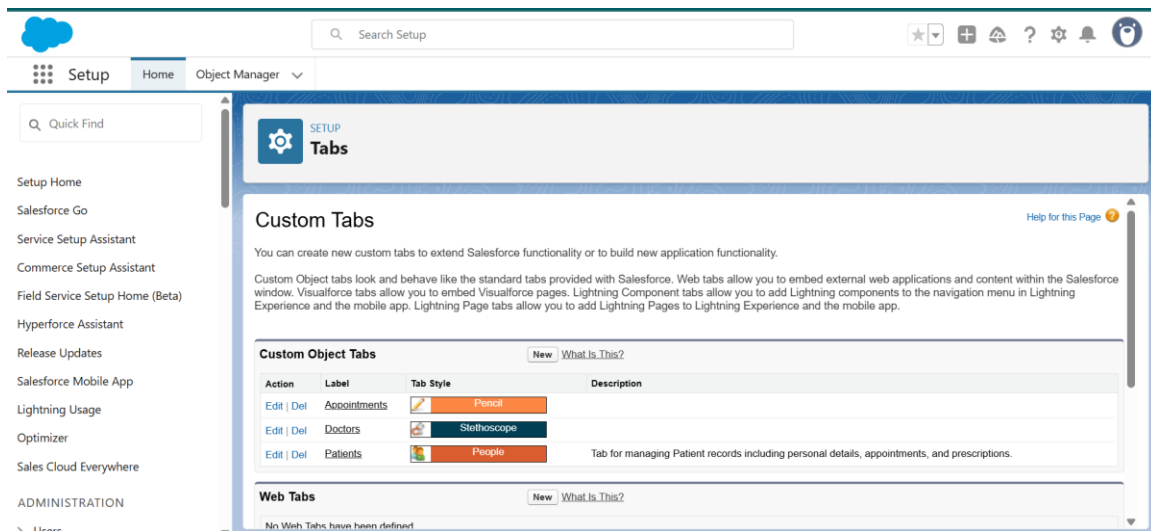
Receptionists need fast appointment entry. I created two record pages with appropriate components:

- Appointment – Doctor View: shows patient summary, related Medical Cases, and the upcoming appointments LWC.
- Appointment – Receptionist View: optimized for quick data entry and follow-up creation.
- Activated per profile/record type so each user sees the correct page.



3. Tabs

I exposed all major objects (Patients, Appointments, Medical Cases) as tabs in the MediCare Lightning app so staff can navigate quickly.



4. Home Page Layouts

I built a custom MediCare Home Page that displays a dashboard component (Upcoming Appointments, Recent Appointments & Medical Cases, Upcoming Appointments Report Chart) for a real-time view.

5. Utility Bar

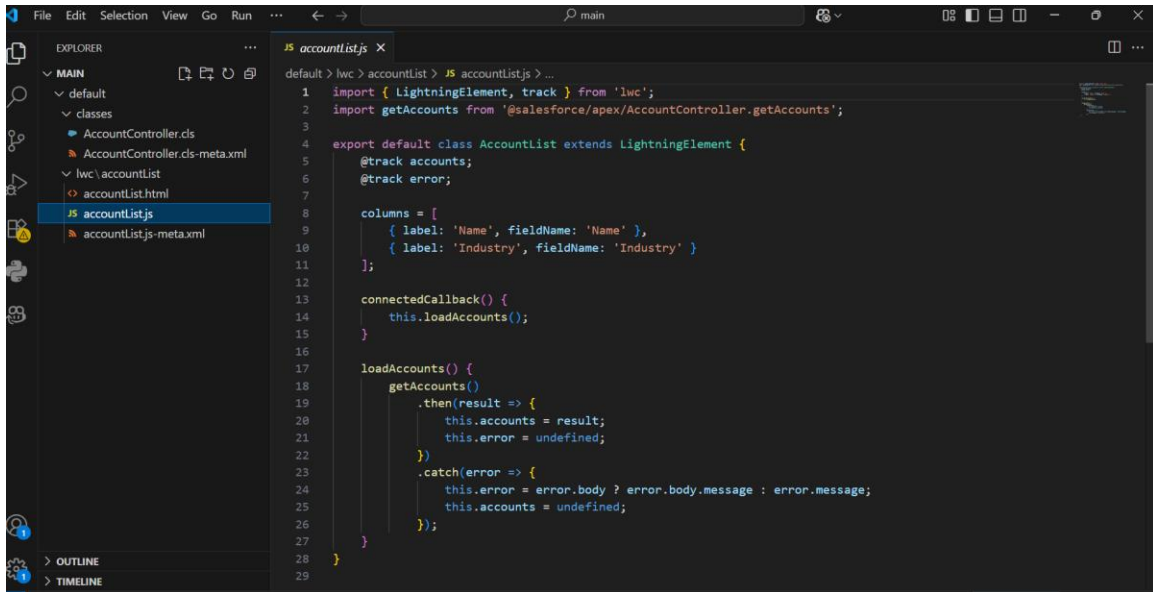
I added Reports/Dashboards and Quick Actions to the Utility Bar to give staff one-click access from anywhere in the app.

6. Lightning Web Component (LWC) – “Upcoming Appointments”

- A custom LWC appointmentCalendar built and deployed.
- It queries all upcoming appointments and displays Patient, Doctor, Date/Time, Status, Duration in a datatable.
- Dropped into the Appointment – Doctor View page so the doctor can instantly see upcoming appointments.

7. Apex with LWC

The LWC calls the Apex class AppointmentController using a wire adapter to fetch upcoming appointments server-side. This ensures the doctor always sees real-time appointment data without page refresh.

A screenshot of the Visual Studio Code editor interface. The Explorer pane on the left shows a project structure with folders 'MAIN', 'default', and 'classes'. Under 'classes', there is a file 'AccountController.cls'. Under 'lwc/accountList', there are files 'accountList.html', 'accountList.js' (selected), and 'accountList.js-meta.xml'. The main editor area displays the content of 'accountList.js'. The code is a JavaScript file for a Lightning Web Component (LWC) that extends 'LightningElement'. It imports 'LightningElement' and 'track' from 'lwc', and 'getAccounts' from '@salesforce/apex/AccountController.getAccounts'. The component has two tracked properties: 'accounts' and 'error'. It defines a 'columns' array with two columns: 'Name' and 'Industry'. The 'connectedCallback' method calls 'loadAccounts()'. The 'loadAccounts' method calls 'getAccounts()' and uses 'then' and 'catch' to handle the response and errors, updating 'this.accounts' and 'this.error' respectively. The code is as follows:

```
1 import { LightningElement, track } from 'lwc';
2 import getAccounts from '@salesforce/apex/AccountController.getAccounts';
3
4 export default class AccountList extends LightningElement {
5   @track accounts;
6   @track error;
7
8   columns = [
9     { label: 'Name', fieldName: 'Name' },
10    { label: 'Industry', fieldName: 'Industry' }
11  ];
12
13  connectedCallback() {
14    this.loadAccounts();
15  }
16
17  loadAccounts() {
18    getAccounts()
19      .then(result => {
20        this.accounts = result;
21        this.error = undefined;
22      })
23      .catch(error => {
24        this.error = error.body ? error.body.message : error.message;
25        this.accounts = undefined;
26      });
27  }
28 }
29
```

8. Wire Adapters

I used a @wire adapter in the LWC to call the Apex method automatically and refresh when the data changes. This keeps the appointments list up-to-date without manual refresh.

9. Results / Observations

- Doctors now see a live list of upcoming appointments directly on their Appointment page.
- Receptionists can quickly create and manage appointments without leaving their view.
- The MediCare Home Page provides a real-time snapshot of the hospital's activities (appointments and cases).
- Utility Bar ensures quick access to reports, dashboards and actions from anywhere in the app.
- The LWC with Apex integration makes the UI dynamic and responsive, reducing page refreshes.