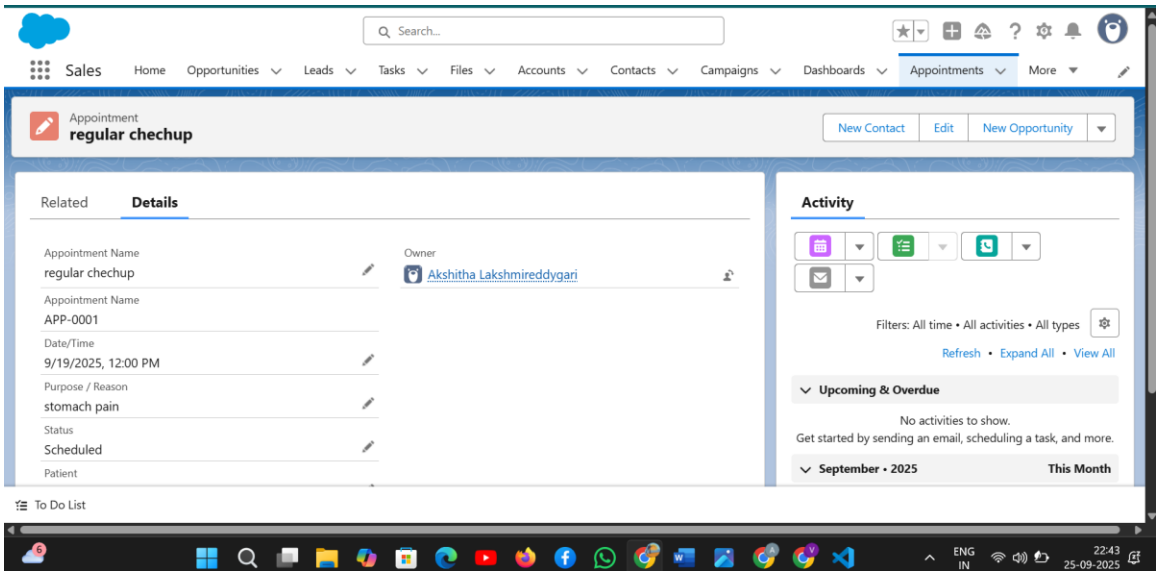
Phase 6: User Interface Development

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.



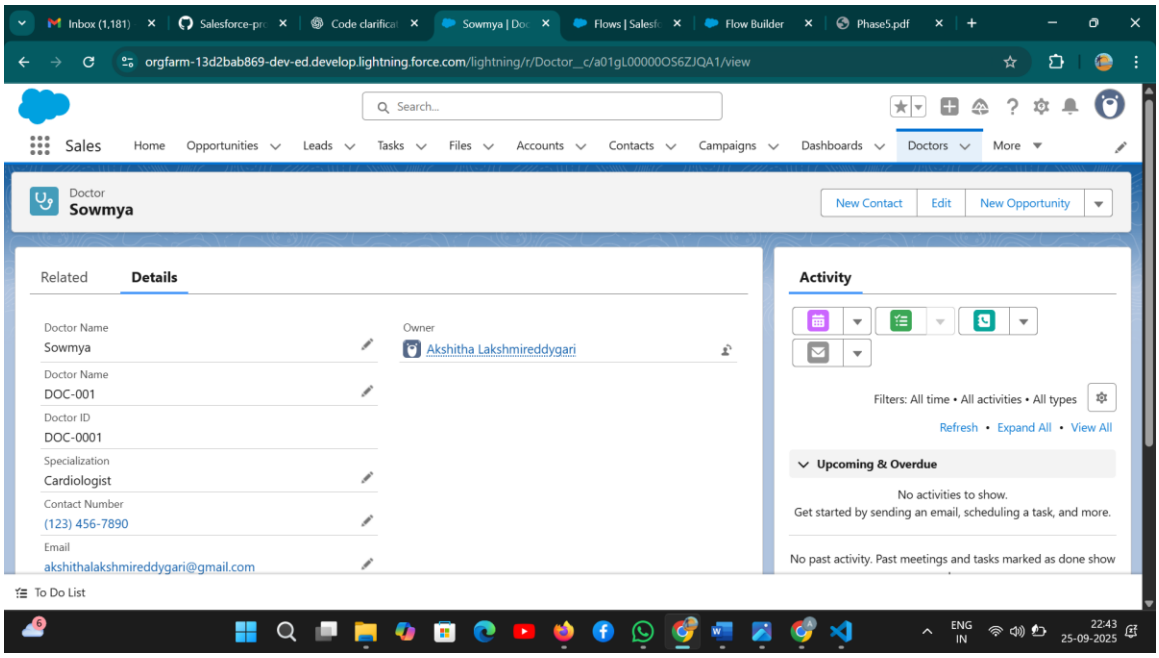
2. Record Pages (Doctor & Receptionist Views)

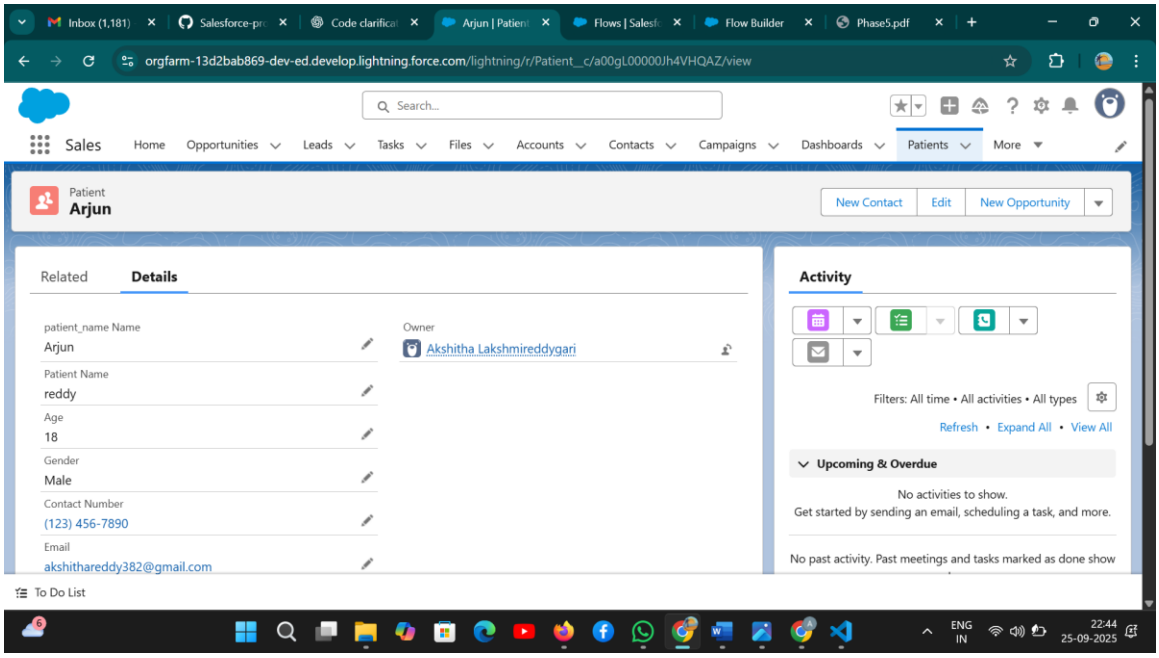
Doctors need quick access to upcoming appointments and related Medical Cases. Receptionists need fast appointment entry. I created two record pages with appropriate components:

• 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.

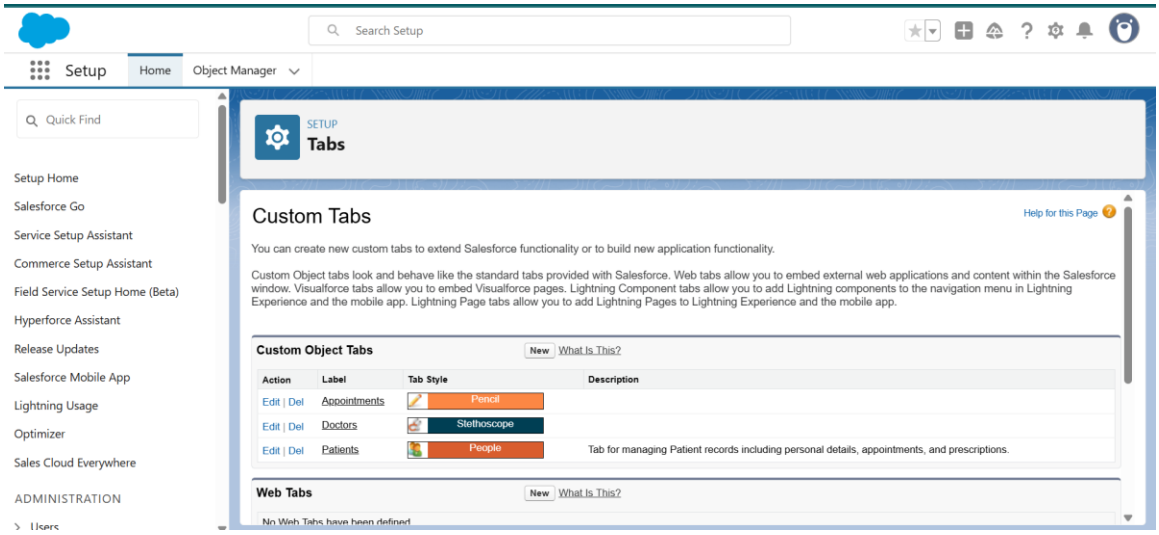
• Activated per profile/record type so each user sees the correct page.





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

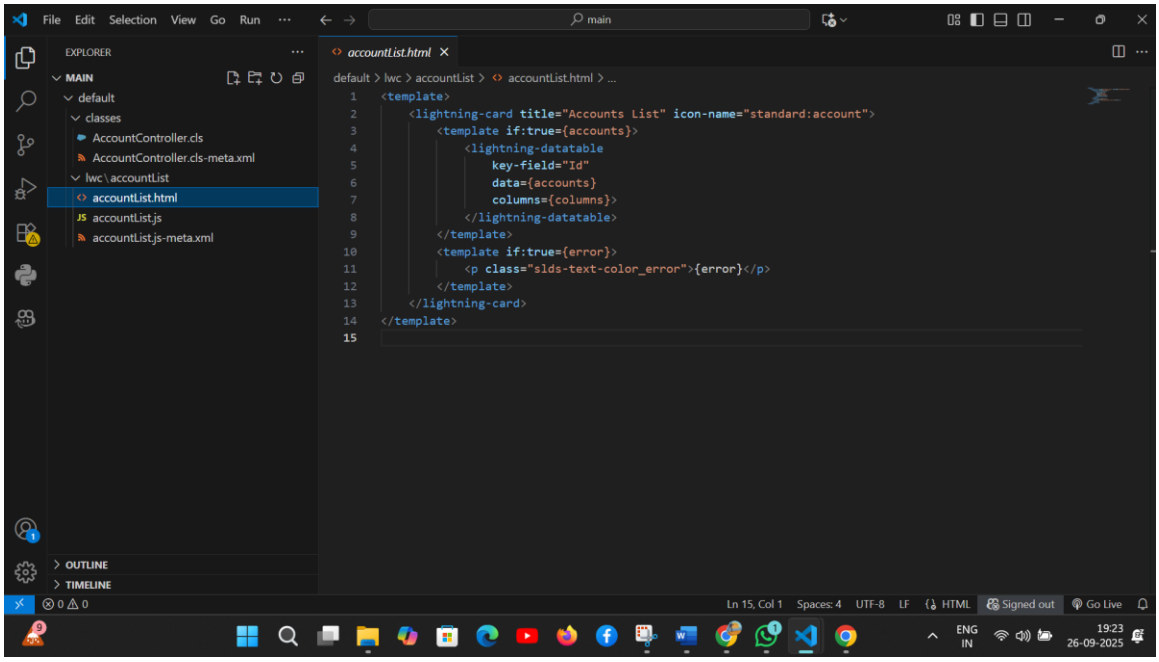
I added Reports/Dashboards and Quick Actions to the Utility Bar to give staff one-click access from anywhere in the app.

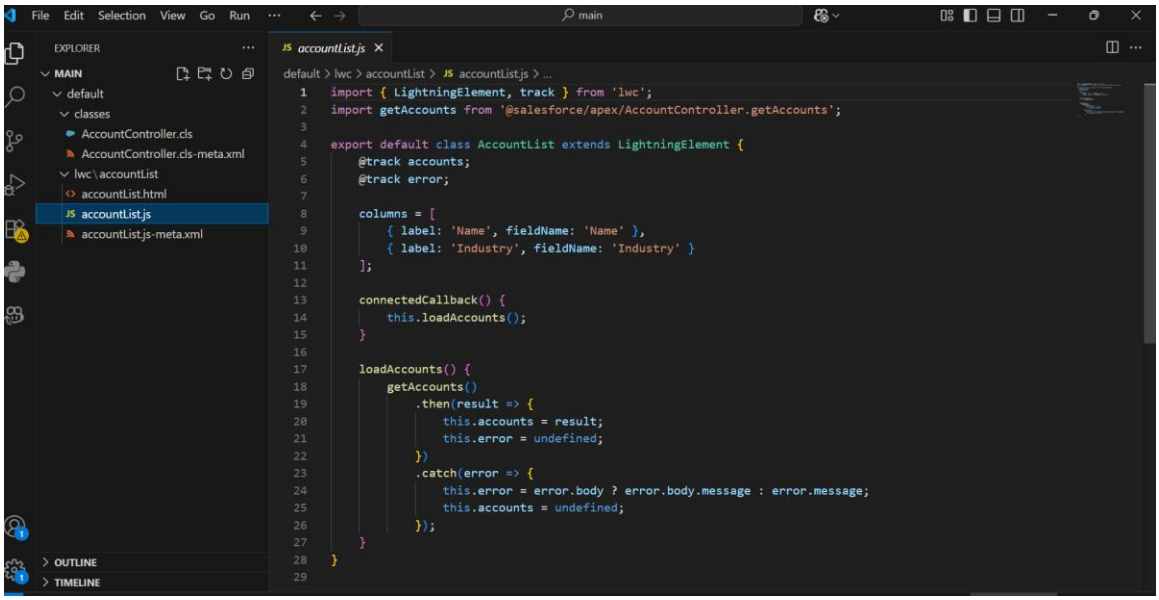
6. Lightning Web Component (LWC) – "Upcoming Appointments"

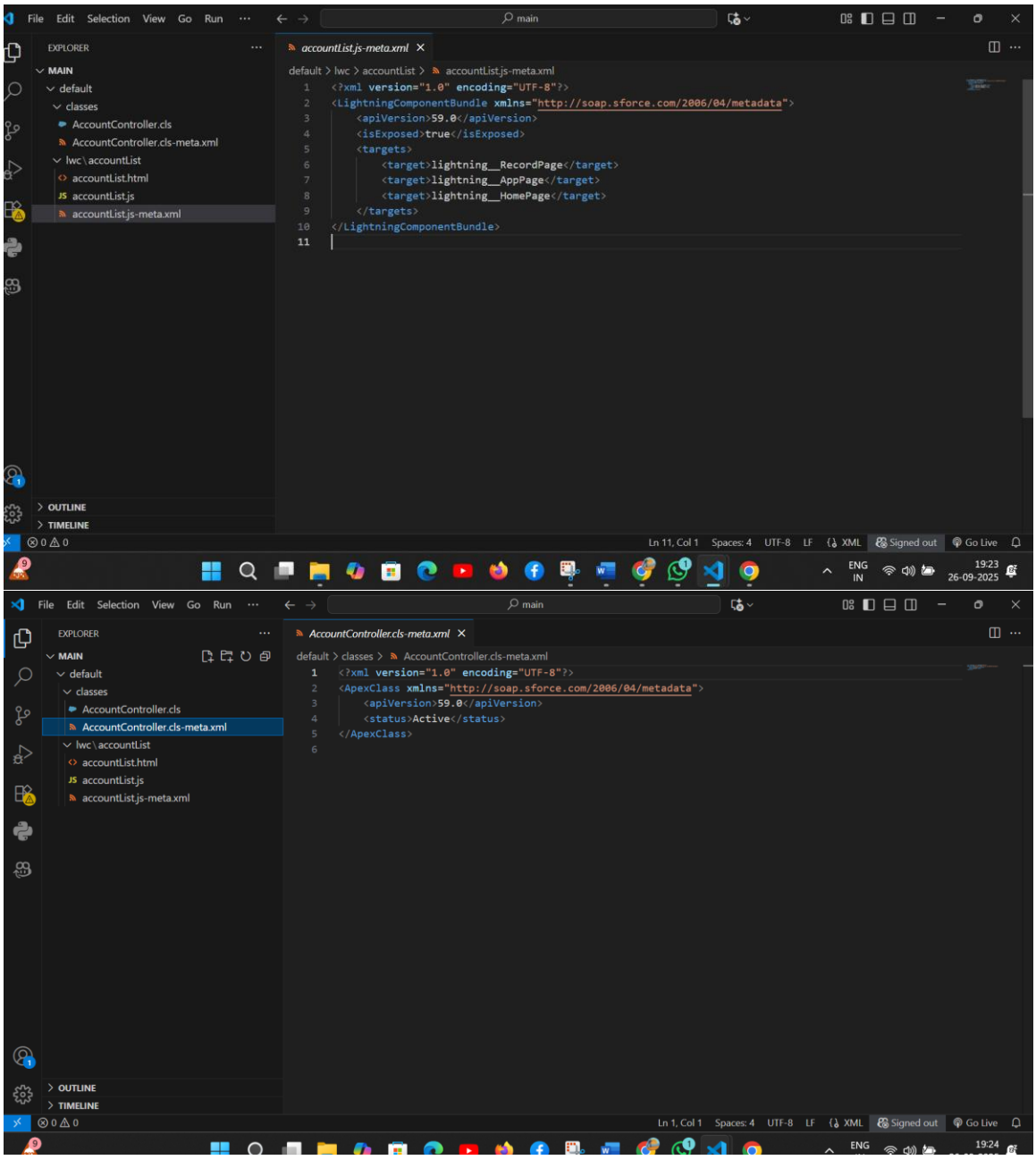
• A custom LWC appointmentCalendar built and deployed.
• It queries all upcoming appointments and displays Patient, Doctor, Date/Time, Status, Duration in a datatable.
• Dropped into the Appointment – Doctor View page so the doctor can instantly see upcoming appointments.

7. Apex with LWC

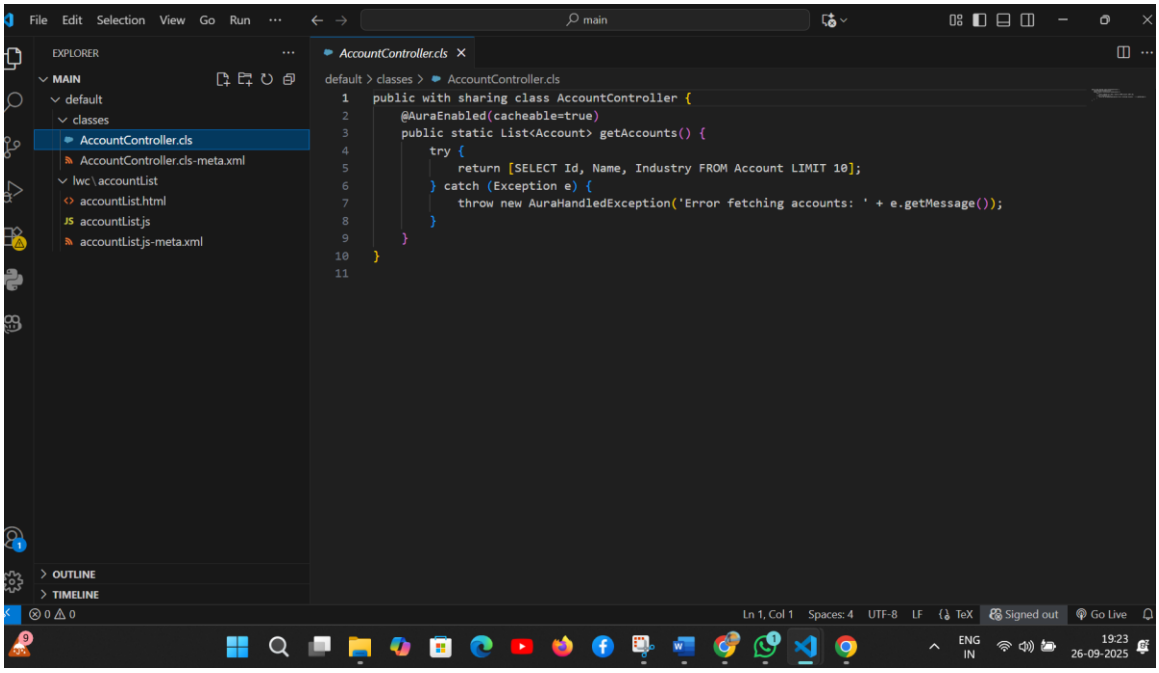
The LWC calls the Apex class AppointmentController using a wire adapter to fetch upcoming appointments server-side. This ensures the doctor always sees real-time appointment data without page refresh.


File Edit Selection View Go Run
main
EXPLORER
MAIN
default
classes
AccountController.cls
AccountController.cls-meta.xml
lwc\accountList
accountList.html
accountList.js
accountList.js-meta.xml
accountList.html
default > lwc > accountList > accountList.html > ...
<template>
<lightning-card title="Accounts List" icon-name="standard:account">
<template if:true={accounts}>
<lightning-datatable
key-field="Id"
data={accounts}
columns={columns}>
</lightning-datatable>
</template>
<template if:true={error}>
{error}
</template>
</lightning-card>
</template>
OUTLINE
TIMELINE
Ln 15, Col 1 Spaces: 4 UTF-8 LF HTML Signed out Go Live
ENG IN
19:23 26-09-2025


File Edit Selection View Go Run
main
EXPLORER
MAIN
default
classes
AccountController.cls
AccountController.cls-meta.xml
lwc\accountList
accountList.html
accountList.js
accountList.js-meta.xml
accountList.js
default > lwc > accountList > accountList.js > ...
import { LightningElement, track } from 'lwc';
import getAccounts from '@salesforce/apex/AccountController.getAccounts';
export default class AccountList extends LightningElement {
@track accounts;
@track error;
columns = [
{ label: 'Name', fieldName: 'Name' },
{ label: 'Industry', fieldName: 'Industry' }
];
connectedCallback() {
this.loadAccounts();
}
loadAccounts() {
getAccounts()
.then(result => {
this.accounts = result;
this.error = undefined;
})
.catch(error => {
this.error = error.body ? error.body.message : error.message;
this.accounts = undefined;
});
}
}
OUTLINE
TIMELINE


File Edit Selection View Go Run
main
EXPLORER
MAIN
default
classes
AccountController.cls
AccountController.cls-meta.xml
lwc\accountList
accountList.html
accountList.js
accountList.js-meta.xml
accountList.js-meta.xml
default > lwc > accountList > accountList.js-meta.xml
<?xml version="1.0" encoding="UTF-8"?>
<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
<apiVersion>59.0</apiVersion>
<isExposed>true</isExposed>
<targets>
<target>lightning__RecordPage</target>
<target>lightning__AppPage</target>
<target>lightning__HomePage</target>
</targets>
</LightningComponentBundle>
OUTLINE
TIMELINE
Ln 11, Col 1 Spaces: 4 UTF-8 LF XML Signed out Go Live
ENG IN
19:23 26-09-2025

File Edit Selection View Go Run
main
EXPLORER
MAIN
default
classes
AccountController.cls
AccountController.cls-meta.xml
lwc\accountList
accountList.html
accountList.js
accountList.js-meta.xml
AccountController.cls-meta.xml
default > classes > AccountController.cls-meta.xml
<?xml version="1.0" encoding="UTF-8"?>
<ApexClass xmlns="http://soap.sforce.com/2006/04/metadata">
<apiVersion>59.0</apiVersion>
<status>Active</status>
</ApexClass>
OUTLINE
TIMELINE
Ln 1, Col 1 Spaces: 4 UTF-8 LF XML Signed out Go Live
ENG
19:24



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.