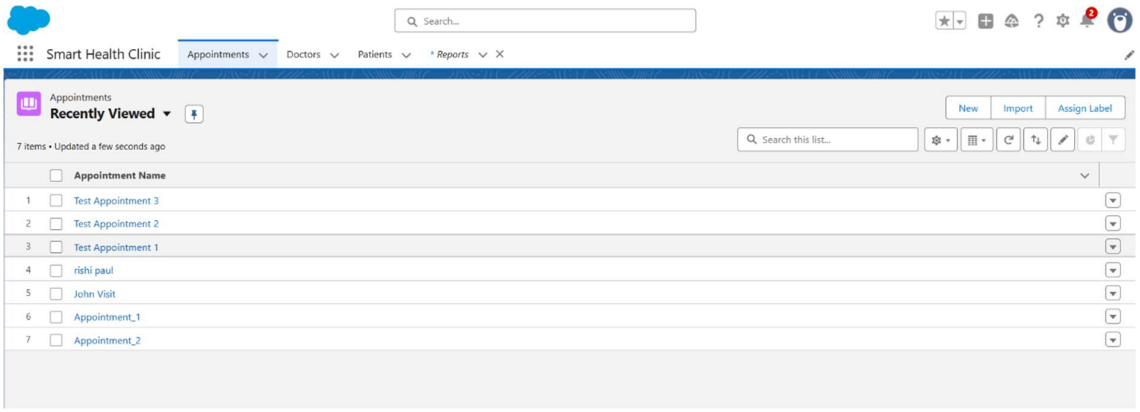
PHASE – 6 User Interface Development

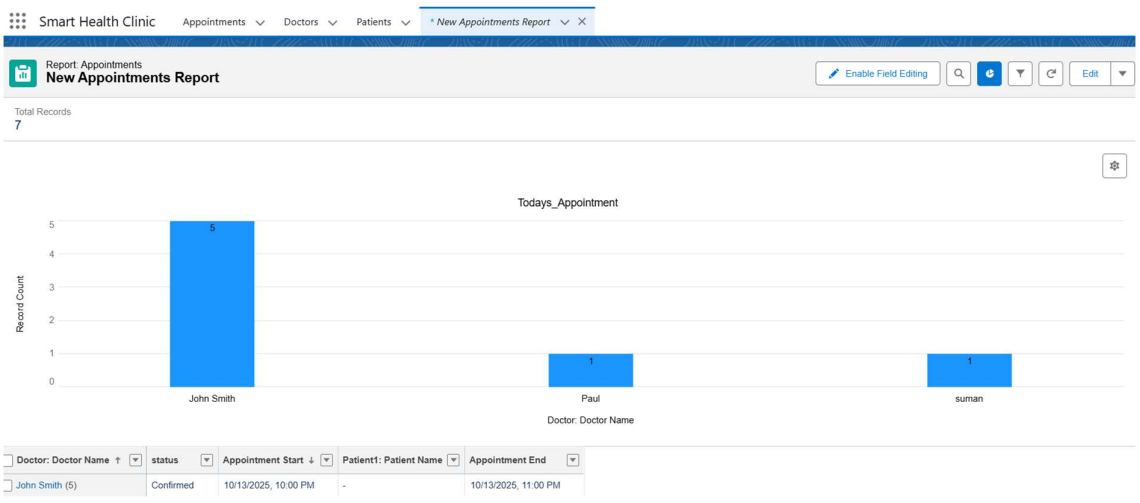
1.Lightning App Builder

The Lightning App Builder is a point-and-click tool in Salesforce used to create custom pages for Lightning Experience and the Salesforce mobile app. It allows admins to easily drag and drop components, arrange layouts, and build apps without writing code. You can use it to create record pages, home pages, and app pages tailored to specific user needs.



2.Record Pages

Record Pages define how a record (like an Account, Contact, or custom object) appears in Salesforce. With the Lightning App Builder, you can customize these pages by adding sections, fields, components, and related lists to make information more organized and accessible to users.



3. Tabs

Tabs in Salesforce provide easy access to objects and apps. You can create custom tabs for standard or custom objects, Visualforce pages, or Lightning pages. Tabs help users navigate quickly and improve usability by bringing frequently used features to the forefront.

4. Home Page Layouts

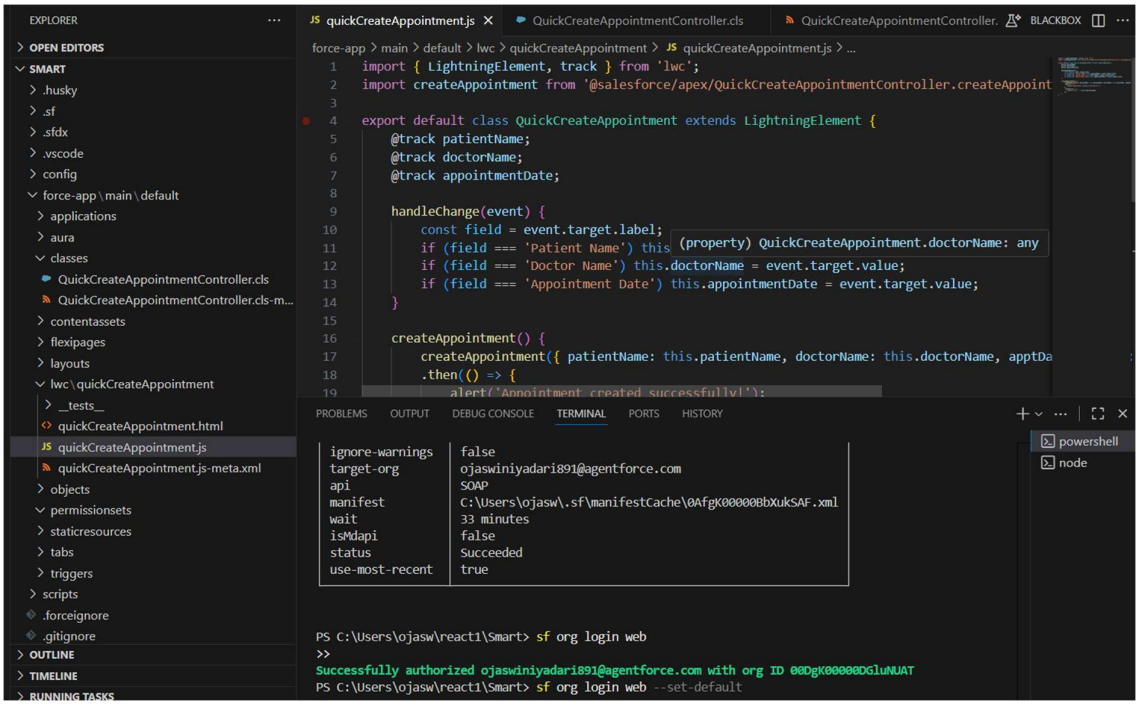
Home Page Layouts give users a personalized dashboard view with important information like tasks, recent records, and dashboards. Using the Lightning App Builder, admins can create customized home pages for different profiles so each user sees what's most relevant to their role.

5. Utility Bar

The Utility Bar appears at the bottom of Lightning apps and provides quick access to common tools such as Notes, History, or custom Lightning components. It helps improve productivity by allowing users to perform actions without leaving their current workspace.

6. LWC (Lightning Web Components)

Lightning Web Components (LWC) is a modern web development framework used in Salesforce to build fast, reusable, and dynamic UI components. It uses standard web technologies like JavaScript, HTML, and CSS, making it easy to create interactive user interfaces.



7. Apex with LWC

Apex can be used with LWC to perform backend logic like data retrieval or updates. By integrating Apex methods with LWCs, developers can connect the user interface with Salesforce data and business logic to create powerful, data-driven applications.

8. Events in LWC

Events in LWC enable communication between components. For example, a child component can send data to a parent component using custom events. This event-driven architecture makes the user interface dynamic and interactive.

9. Wire Adapters

Wire Adapters in LWC are used to get Salesforce data automatically using declarative syntax. They simplify the process of fetching records or lists from Salesforce without writing Apex code, ensuring data updates automatically when changes occur.

10. Imperative Apex Calls

Imperative Apex Calls allow developers to manually invoke Apex methods from JavaScript in LWC. This approach is useful when you need more control over when and how data is fetched or updated, such as after a button click or user input.

11. Navigation Service

The Navigation Service in LWC helps navigate users between different pages or records within Salesforce. Developers can programmatically redirect users to record pages, lists, or external URLs, creating a seamless and guided experience in the app.