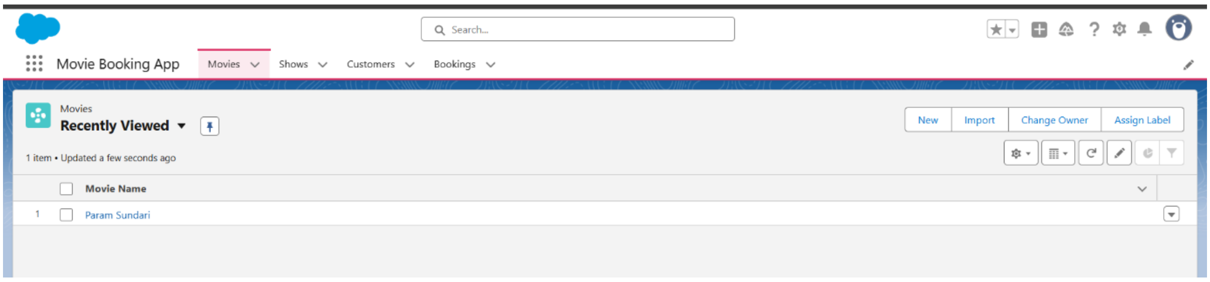
Phase 6: User Interface Development

- **Lightning App Builder:**

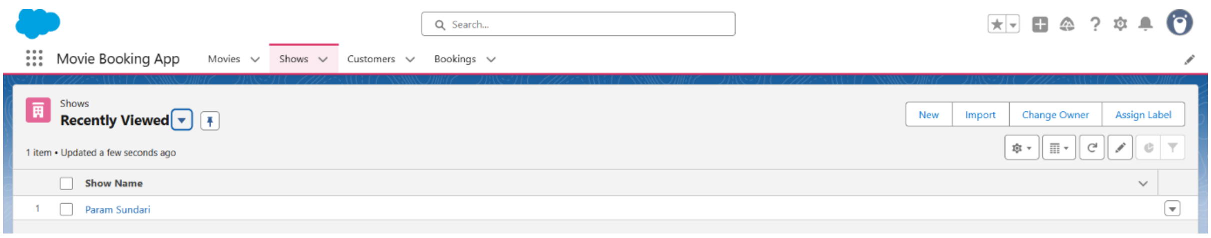
  By using the Type

  - Movie Booking App using Lightning Experience App Manager. This app is now visible in the App Launcher.
  - **Tabs Added:**
  - **Movie__c** → for movie details
  - **Show__c** → for show details
  - **Customer__c** → for customer information
  - **Booking__c** → for booking records



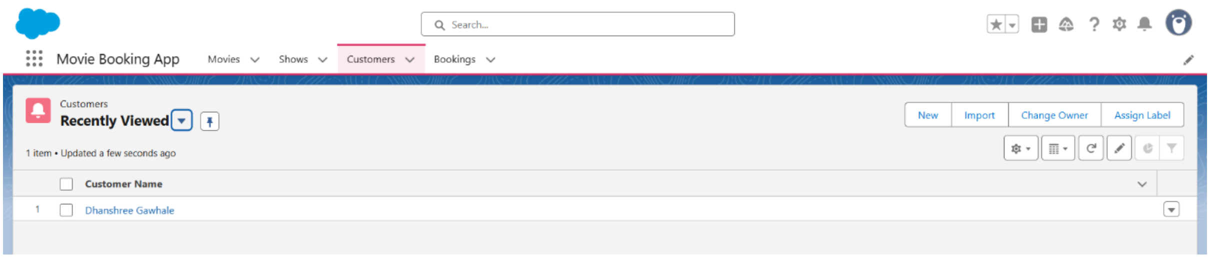
1 item • Updated a few seconds ago

	Movie Name
1	Param Sundari



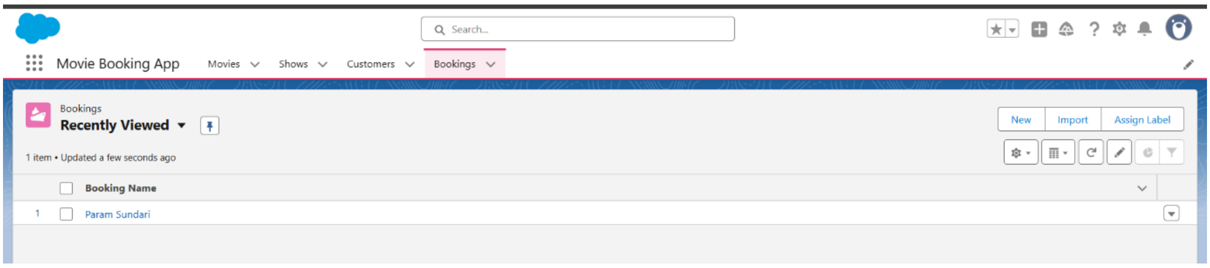
1 item • Updated a few seconds ago

	Show Name
1	Param Sundari



1 item • Updated a few seconds ago

	Customer Name
1	Dhanshree Gawhale

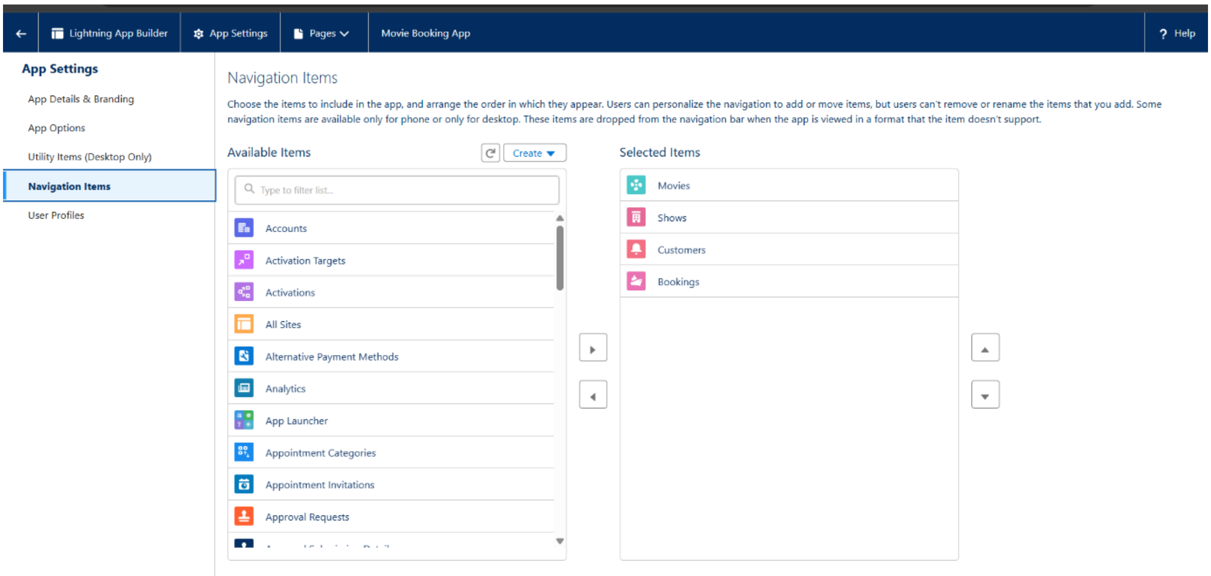


Bookings

Recently Viewed

1 item • Updated a few seconds ago

	Booking Name
1	Param Sundari



Navigation Items

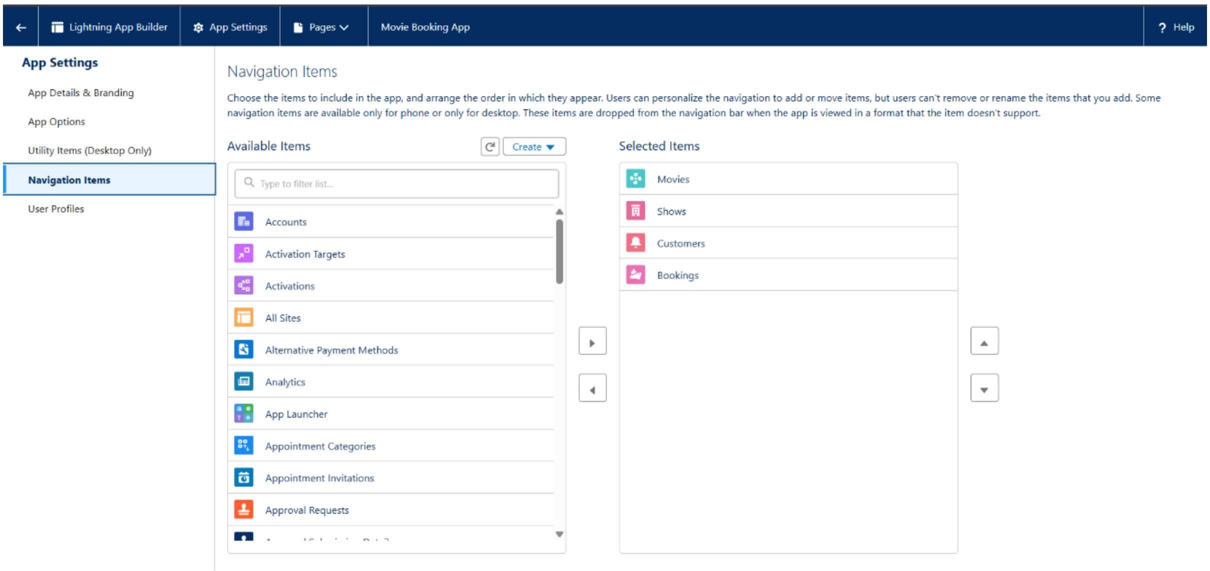
Choose the items to include in the app, and arrange the order in which they appear. Users can personalize the navigation to add or move items, but users can't remove or rename the items that you add. Some navigation items are available only for phone or only for desktop. These items are dropped from the navigation bar when the app is viewed in a format that the item doesn't support.

Available Items

- Accounts
- Activation Targets
- Activations
- All Sites
- Alternative Payment Methods
- Analytics
- App Launcher
- Appointment Categories
- Appointment Invitations
- Approval Requests

Selected Items

- Movies
- Shows
- Customers
- Bookings


Navigation Items

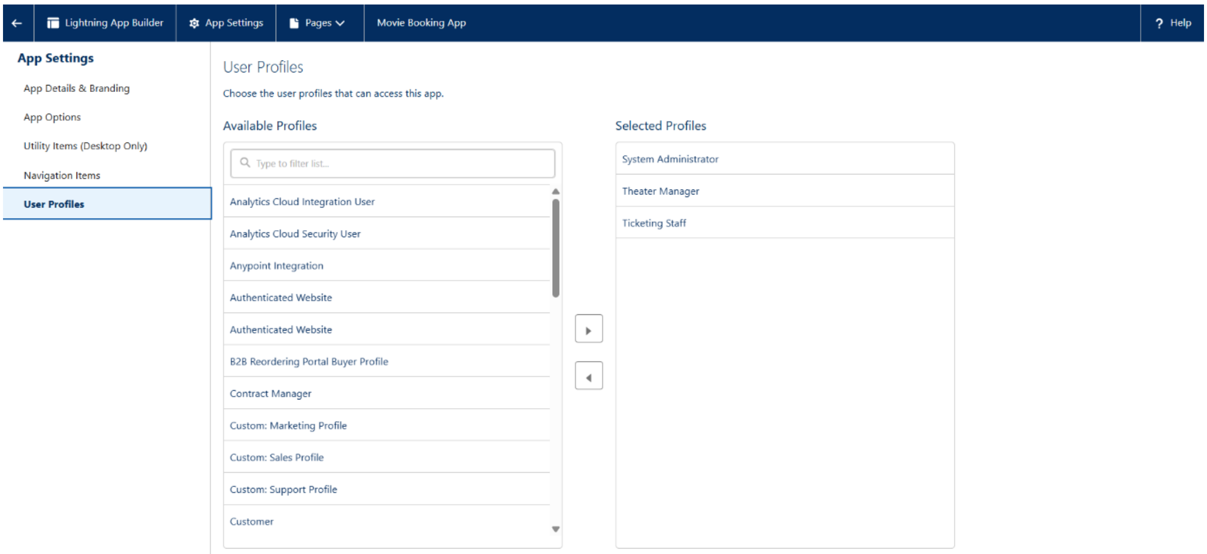
Choose the items to include in the app, and arrange the order in which they appear. Users can personalize the navigation to add or move items, but users can't remove or rename the items that you add. Some navigation items are available only for phone or only for desktop. These items are dropped from the navigation bar when the app is viewed in a format that the item doesn't support.

Available Items

- Accounts
- Activation Targets
- Activations
- All Sites
- Alternative Payment Methods
- Analytics
- App Launcher
- Appointment Categories
- Appointment Invitations
- Approval Requests

Selected Items

- Movies
- Shows
- Customers
- Bookings



User Profiles

Choose the user profiles that can access this app.

Available Profiles

Type to filter list...

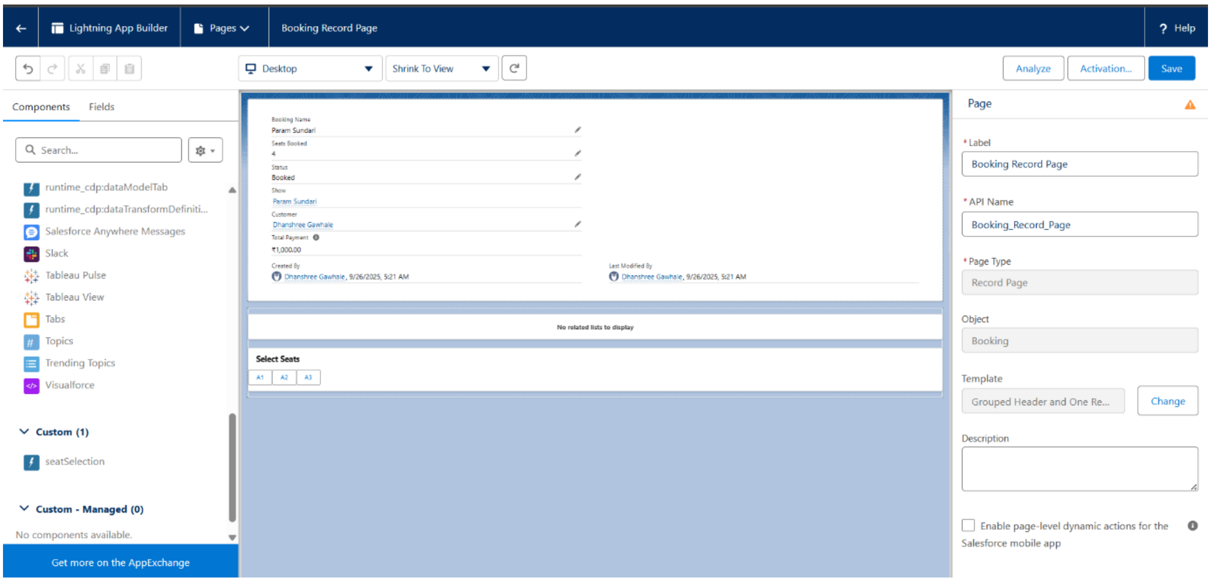
- Analytics Cloud Integration User
- Analytics Cloud Security User
- Anypoint Integration
- Authenticated Website
- Authenticated Website
- B2B Reordering Portal Buyer Profile
- Contract Manager
- Custom: Marketing Profile
- Custom: Sales Profile
- Custom: Support Profile
- Customer

Selected Profiles

- System Administrator
- Theater Manager
- Ticketing Staff

❖ Record pages

- Defines different types of records for the same object.
- Controls page layouts, picklist values, and business processes.
- Example: “Regular Booking” vs “Premium Booking” in a Movie Booking App.



Booking Name
Param Sundari

Seats Booked
4

Status
Booked

Show
Param Sundari

Customer
Dhanshree Gawhale

Total Payment
₹1,000.00

Created By
Dhanshree Gawhale, 9/26/2025, 5:21 AM

Last Modified By
Dhanshree Gawhale, 9/26/2025, 5:21 AM

No related lists to display

Select Seats

A1 A2 A3

Page

* Label
Booking Record Page

* API Name
Booking_Record_Page

* Page Type
Record Page

Object
Booking

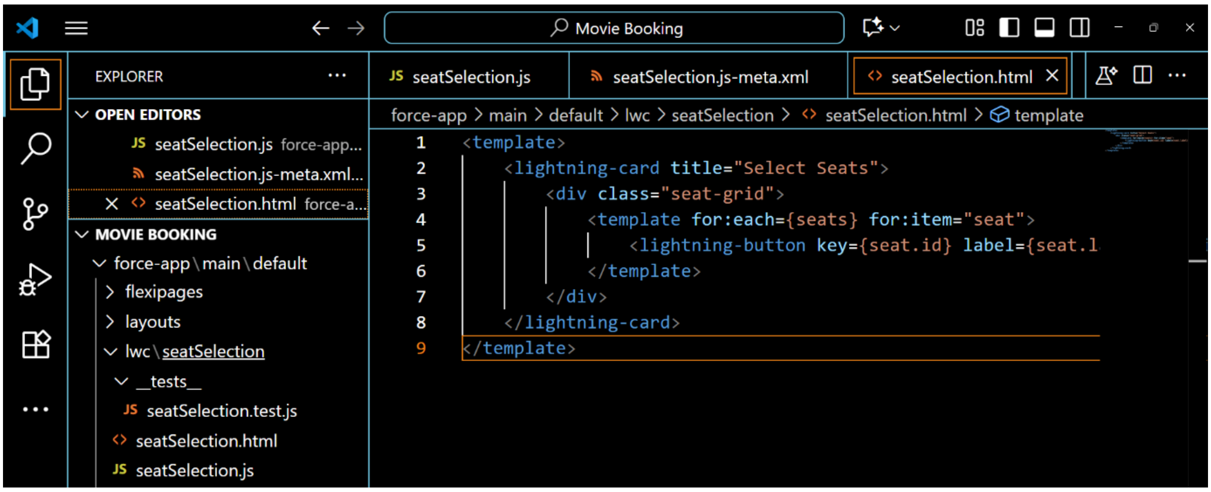
Template
Grouped Header and One Re...

Description

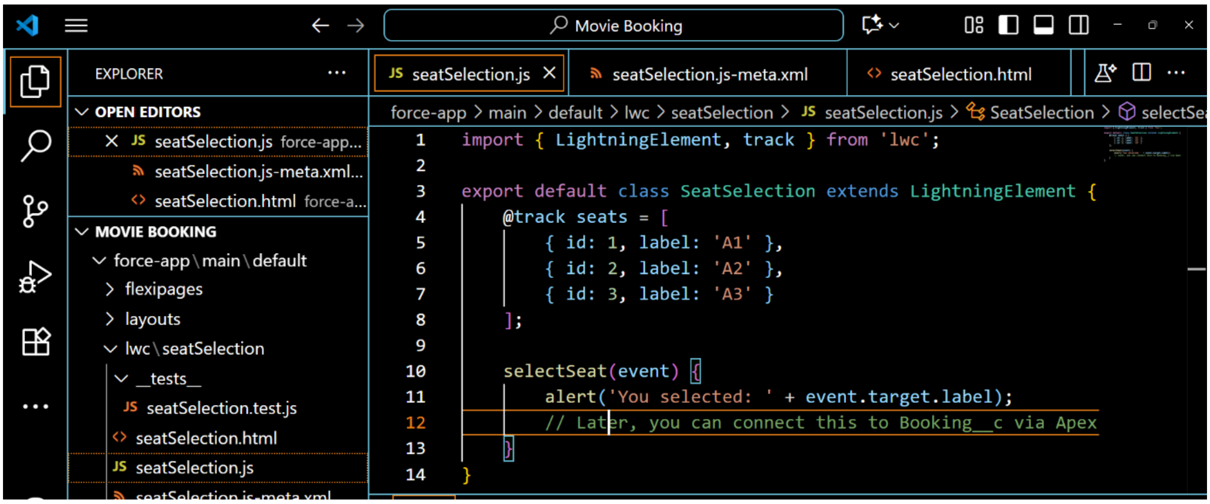
□ Enable page-level dynamic actions for the Salesforce mobile app

❖ Lightning Web Component (LWC)

- Custom, reusable component built with HTML, JavaScript, and CSS.
- Adds interactive features on pages, like seat selection.
- Can be placed on Record Pages, Apps, or Lightning Pages


Movie Booking
EXPLORER
OPEN EDITORS
seatSelection.js force-app...
seatSelection.js-meta.xml...
seatSelection.html force-a...
MOVIE BOOKING
force-app \ main \ default
flexipages
layouts
lwc \ seatSelection
__tests__
seatSelection.test.js
seatSelection.html
seatSelection.js
seatSelection.js
seatSelection.js-meta.xml
seatSelection.html
force-app > main > default > lwc > seatSelection > seatSelection.html > template

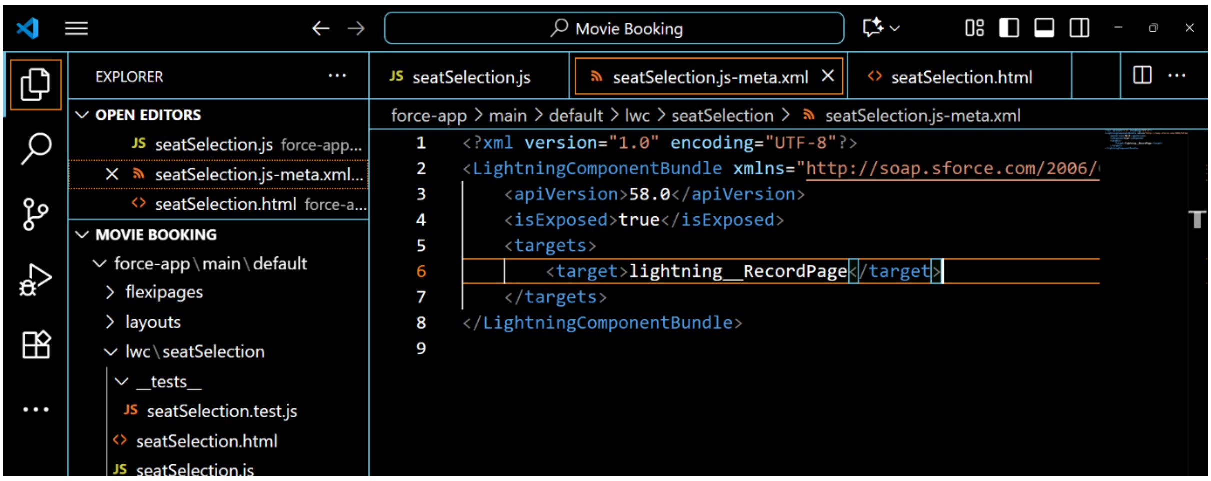
```
<template>
    <lightning-card title="Select Seats">
        <div class="seat-grid">
            <template for:each={seats} for:item="seat">
                <lightning-button key={seat.id} label={seat.l
            </template>
        </div>
    </lightning-card>
</template>
```


Movie Booking
EXPLORER
OPEN EDITORS
seatSelection.js force-app...
seatSelection.js-meta.xml...
seatSelection.html force-a...
MOVIE BOOKING
force-app \ main \ default
flexipages
layouts
lwc \ seatSelection
__tests__
seatSelection.test.js
seatSelection.html
seatSelection.js
seatSelection.js-meta.xml
seatSelection.js
seatSelection.js-meta.xml
seatSelection.html
force-app > main > default > lwc > seatSelection > seatSelection.js > SeatSelection > selectSe

```
import { LightningElement, track } from 'lwc';

export default class SeatSelection extends LightningElement {
    @track seats = [
        { id: 1, label: 'A1' },
        { id: 2, label: 'A2' },
        { id: 3, label: 'A3' }
    ];

    selectSeat(event) {
        alert('You selected: ' + event.target.label);
        // Later, you can connect this to Booking__c via Apex
    }
}
```


Movie Booking
EXPLORER
OPEN EDITORS
seatSelection.js force-app...
seatSelection.js-meta.xml...
seatSelection.html force-a...
MOVIE BOOKING
force-app \ main \ default
flexipages
layouts
lwc \ seatSelection
__tests__
seatSelection.test.js
seatSelection.html
seatSelection.js
seatSelection.js
seatSelection.js-meta.xml
seatSelection.html
force-app > main > default > lwc > seatSelection > seatSelection.js-meta.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<LightningComponentBundle xmlns="http://soap.sforce.com/2006/(
    <apiVersion>58.0</apiVersion>
    <isExposed>true</isExposed>
    <targets>
        <target>lightning__RecordPage</target>
    </targets>
</LightningComponentBundle>
```