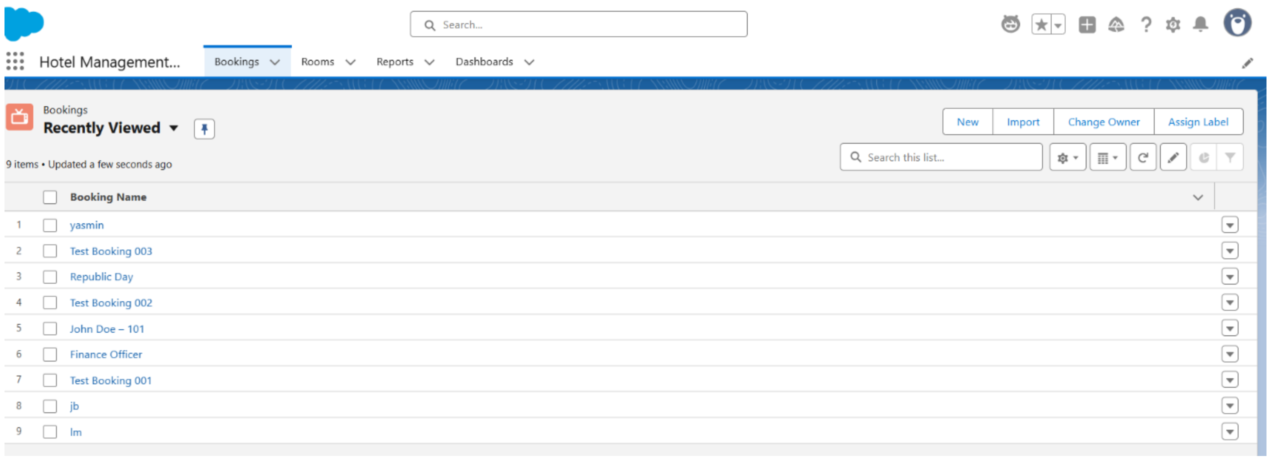
Phase 6: User Interface Development

Goal:
Build a user-friendly and interactive Salesforce Lightning interface for Fraud Detection, enabling Adjusters, Analysts, and Managers to access claims, view fraud scores, and take actions seamlessly.

---

1. **Lightning App Builder – Hotel Booking App**
2. **What was done:**
3. Created a new Lightning App named Hotel Booking App.
4. Added navigation items: Hotels, Rooms, Bookings, Reports, and Dashboards.
5. Created Custom Tabs for custom objects (Hotel__c, Room__c, Room_Booking__c) and added them to the app.
6. **Used:**
7. Provides a dedicated workspace for users to manage hotel information, room availability, and bookings without switching apps.
8. **Steps Followed:**
9. Setup → App Manager → New Lightning App.
10. Enter App Name: Hotel Booking App.
11. Added navigation items: Hotels, Rooms, Bookings, Reports, Dashboards.

12. Assigned the app to Customer, Receptionist, and Manager profiles.



Bookings
Recently Viewed

9 items • Updated a few seconds ago

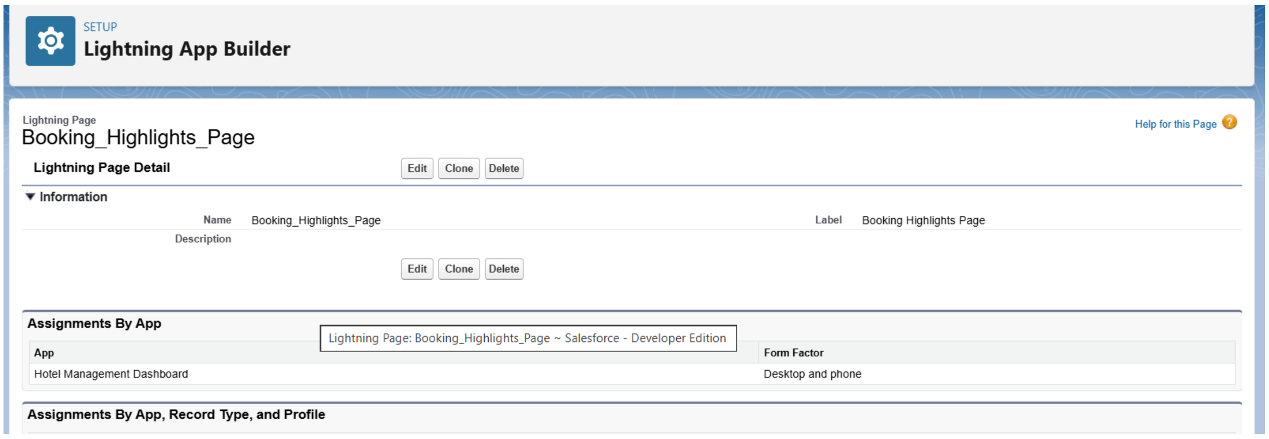
	Booking Name
1	yasmin
2	Test Booking 003
3	Republic Day
4	Test Booking 002
5	John Doe – 101
6	Finance Officer
7	Test Booking 001
8	jb
9	lm

2. Record Pages – Hotel & Room Page Customization

- ### What was done:
  - **Customized Hotel record page with components:**
    - Highlights Panel (Hotel Name, Location, Rooms Available, Base Price).
    - Record Detail (hotel details).
    - Related Lists (Rooms, Bookings).
    - Custom LWCs → hotelBooking, roomAvailabilityPanel.
- **Used:**
  - Provides a single view of hotel details and booking info for faster decision-making.
- **Steps Followed:**

1. Setup → Lightning App Builder → New Record Page.
2. Selected Hotel__c object.
3. Dragged required components into layout.
4. Added custom LWCs after deployment.
5. Activated the page for the Hotel Booking App.

Activated the page for the *Fraud Detection App*.


SETUP
Lightning App Builder

Lightning Page
Booking_Highlights_Page

Help for this Page

Lightning Page Detail [Edit] [Clone] [Delete]

Information

Name	Booking_Highlights_Page	Label	Booking Highlights Page
Description			

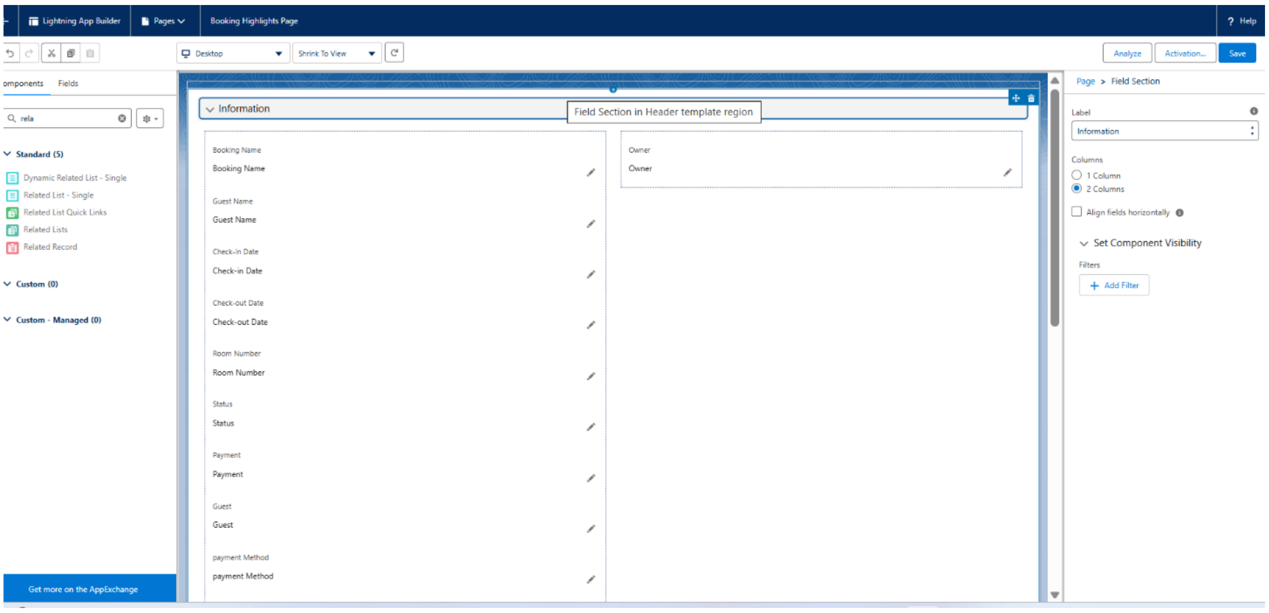
[Edit] [Clone] [Delete]

Assignments By App

Lightning Page: Booking_Highlights_Page ~ Salesforce - Developer Edition

App	Form Factor
Hotel Management Dashboard	Desktop and phone

Assignments By App, Record Type, and Profile


Lightning App Builder | Pages | Booking Highlights Page | Help

Desktop | Shrink To View | Analyze | Activation... | Save

Components | Fields

rela

Standard (5)
- Dynamic Related List - Single
- Related List - Single
- Related List Quick Links
- Related Lists
- Related Record

Custom (0)

Custom - Managed (0)

Get more on the AppExchange

Information | Field Section in Header template region

Booking Name	Owner
Guest Name	
Check-in Date	
Check-out Date	
Room Number	
Status	
Payment	
Guest	
payment Method	

Page > Field Section

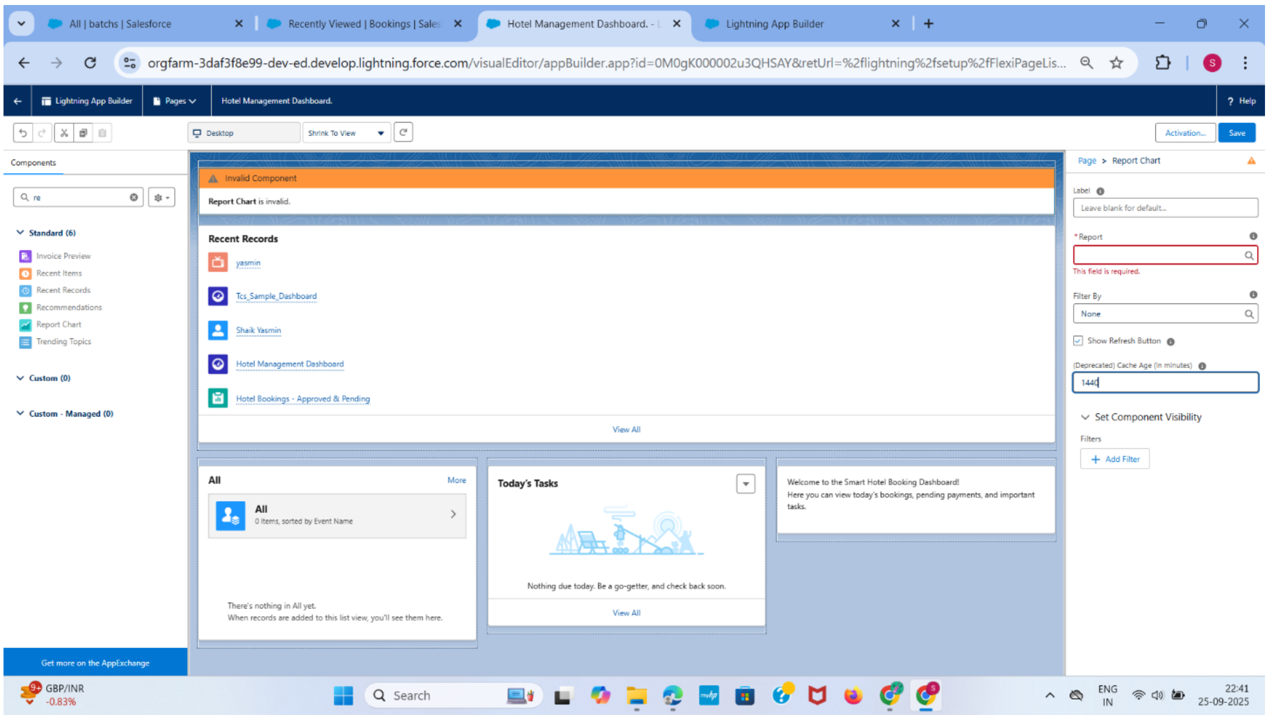
Label: Information

Columns: ○ 1 Column ● 2 Columns

☐ Align fields horizontally

Set Component Visibility

Filters: + Add Filter


All | batchs | Salesforce | Recently Viewed | Bookings | Sale | Hotel Management Dashboard. | Lightning App Builder

orgfarm-3daf3f8e99-dev-ed.develop.lightning.force.com/visualEditor/appBuilder.app?id=0M0gK000002u3QHSAY&retUrl=%2flightning%2fsetup%2fFlexiPageLis...

Lightning App Builder | Pages | Hotel Management Dashboard. | Help

Desktop | Shrink To View | Activation... | Save

Components

re

Standard (6)
- Invoice Preview
- Recent Items
- Recent Records
- Recommendations
- Report Chart
- Trending Topics

Custom (0)

Custom - Managed (0)

Get more on the AppExchange

Invalid Component

Report Chart is invalid.

Recent Records
- yasmin
- Tcs_Sample_Dashboard
- Shaik Yasmin
- Hotel Management Dashboard
- Hotel Bookings - Approved & Pending

View All

All | More

All — 0 items, sorted by Event Name

There's nothing in All yet. When records are added to this list view, you'll see them here.

Today's Tasks

Nothing due today. Be a go-getter, and check back soon.

View All

Welcome to the Smart Hotel Booking Dashboard! Here you can view today's bookings, pending payments, and important tasks.

Page > Report Chart

Label: Leave blank for default...

* Report: This field is required.

Filter By: None

☑ Show Refresh Button

(Deprecated) Cache Age (in minutes): 1440

Set Component Visibility

Filters: + Add Filter

GBP/INR -0.83% | Search | ENG IN | 22:41 25-09-2025

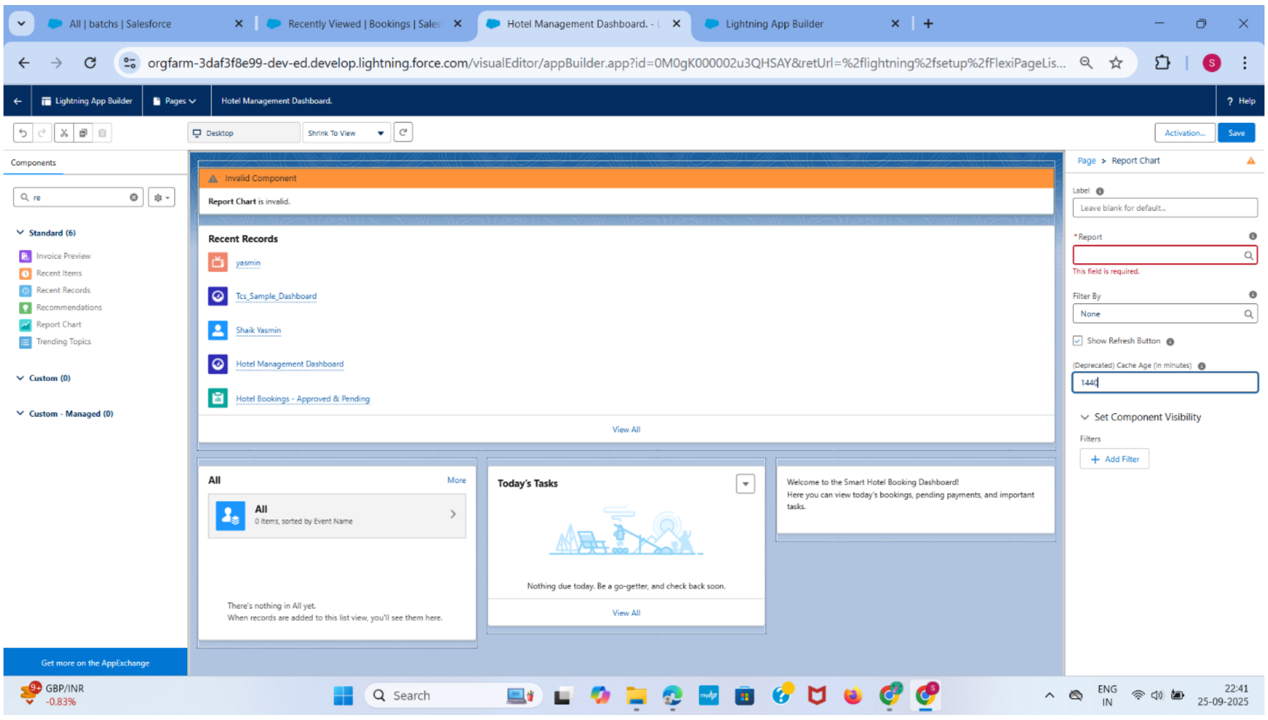
Home Page Layouts

- **What was done:**
  - Created a **Hotel Booking Home Page**.
  - Added components:
    - Report Chart → **Bookings Trends**.

    - Recent Items → Latest Bookings.
    - List View → **Available Rooms / Hotels**.

- **Used:**
    - Provides Managers a dashboard overview of hotel occupancy, popular rooms, and recent bookings.

- **Steps Followed:**
    1. Setup → Lightning App Builder → New Home Page.
    2. Added components: Report Chart, Recent Items, List View.
    3. Activated for **Hotel Booking App** and assigned profiles.


All | batchs | Salesforce
Recently Viewed | Bookings | Sales
Hotel Management Dashboard.
Lightning App Builder
orgfarm-3daf3f8e99-dev-ed.develop.lightning.force.com/visualEditor/appBuilder.app?id=0M0gK000002u3QHSAY&retUrl=%2flightning%2fsetup%2fFlexiPageLis...
Lightning App Builder
Pages
Hotel Management Dashboard.
Help
Desktop
Shrink To View
Activation...
Save
Components
re
Standard (6)
Invoice Preview
Recent Items
Recent Records
Recommendations
Report Chart
Trending Topics
Custom (0)
Custom - Managed (0)
Get more on the AppExchange
Page > Report Chart
GBP/INR -0.83%
Search
ENG IN
22:41 25-09-2025

Invalid Component

Report Chart is invalid.

Recent Records

- yasmin
- Tcs_Sample_Dashboard
- Shaik Yasmin
- Hotel Management Dashboard
- Hotel Bookings - Approved & Pending

View All

All More

All — 0 Items, sorted by Event Name

There's nothing in All yet.
When records are added to this list view, you'll see them here.

Today's Tasks

Nothing due today. Be a go-getter, and check back soon.

View All

Welcome to the Smart Hotel Booking Dashboard!
Here you can view today's bookings, pending payments, and important tasks.

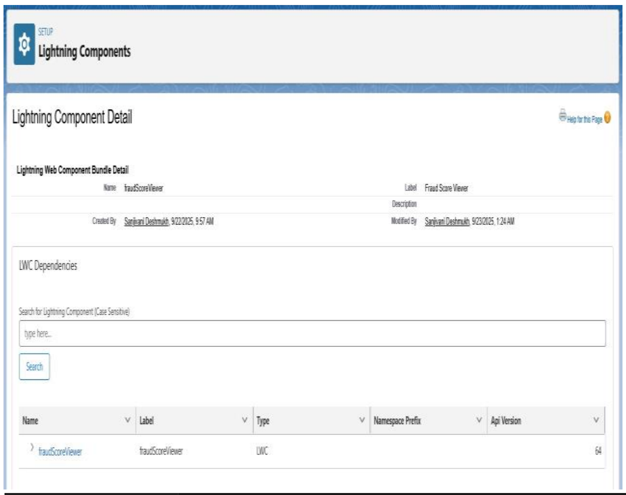
Label: Leave blank for default...
*Report:
This field is required.
Filter By: None
☑ Show Refresh Button
(Deprecated) Cache Age (in minutes): 1440
Set Component Visibility
Filters
+ Add Filter

4. 4. Lightning Web Components (LWCs)

Developed LWCs to extend UI functionality:

- (**a) hotelBooking**
    - Purpose: Allows users to select hotels, rooms, dates, and book rooms.
    - Usage: Added to Hotel Record Page or App Page.
    - Why: Provides interactive booking interface without leaving the page.
- **(b) roomAvailabilityPanel**
    - Purpose: Shows room availability for a selected hotel.
    - Usage: Added to record pages and home page list view.
    - Why: Helps users track available rooms in real time.
- **(c) bookingHistoryTable**
    - Purpose: Displays all bookings in a table view with filters.
    - Usage: Added to Home Page or Dashboard component.
    - Why: Allows Managers to monitor all bookings at a glance.

reports.


SETUP

Lightning Components

Lightning Component Detail

Help for this Page

Lightning Web Component Bundle Detail

Name	fraudScoreViewer	Label	Fraud Score Viewer
		Description	
Created By	Sanjivani Deshmukh, 9/22/2025, 9:57 AM	Modified By	Sanjivani Deshmukh, 9/23/2025, 1:24 AM

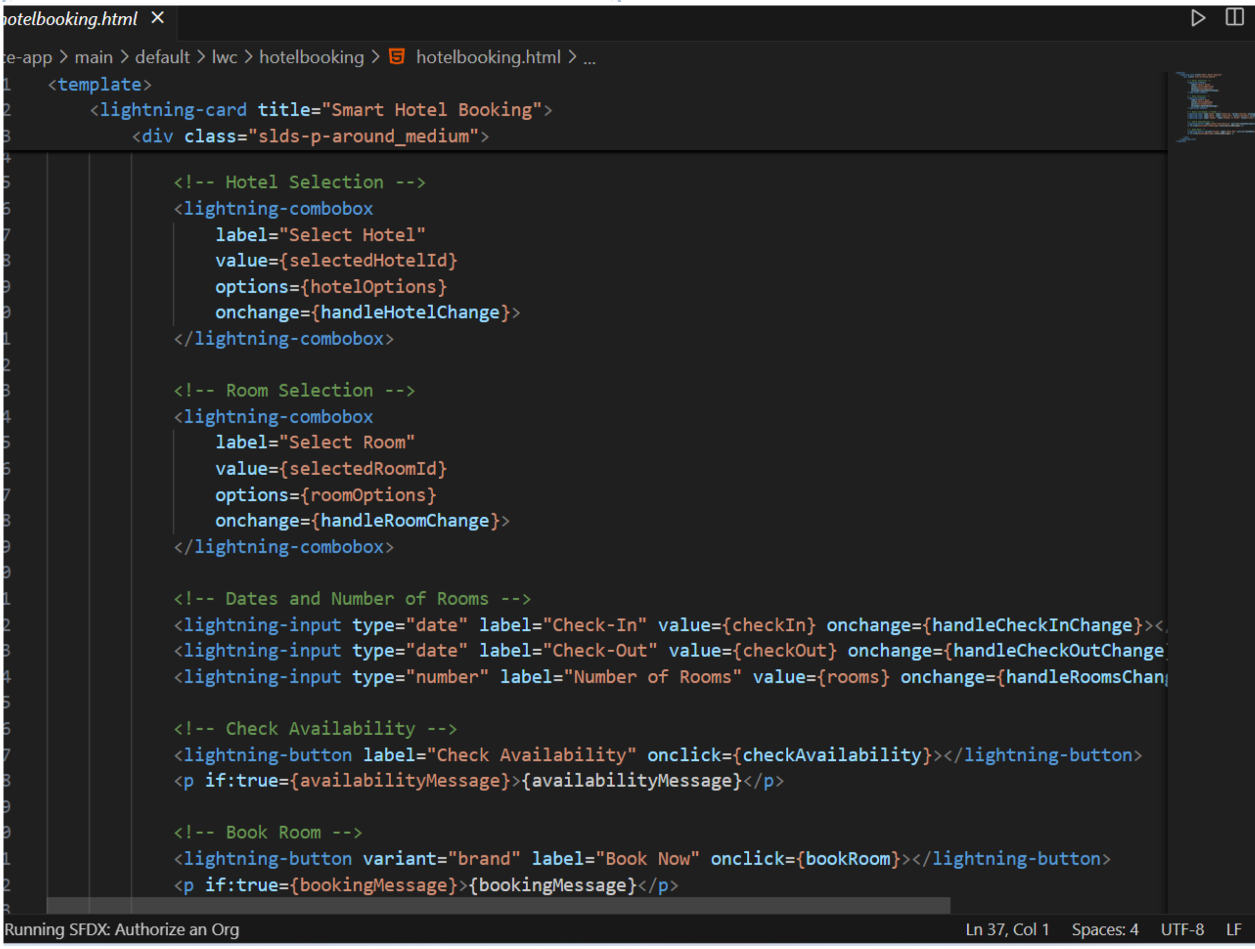
LWC Dependencies

Search for Lightning Component (Case Sensitive)

type here...

Search

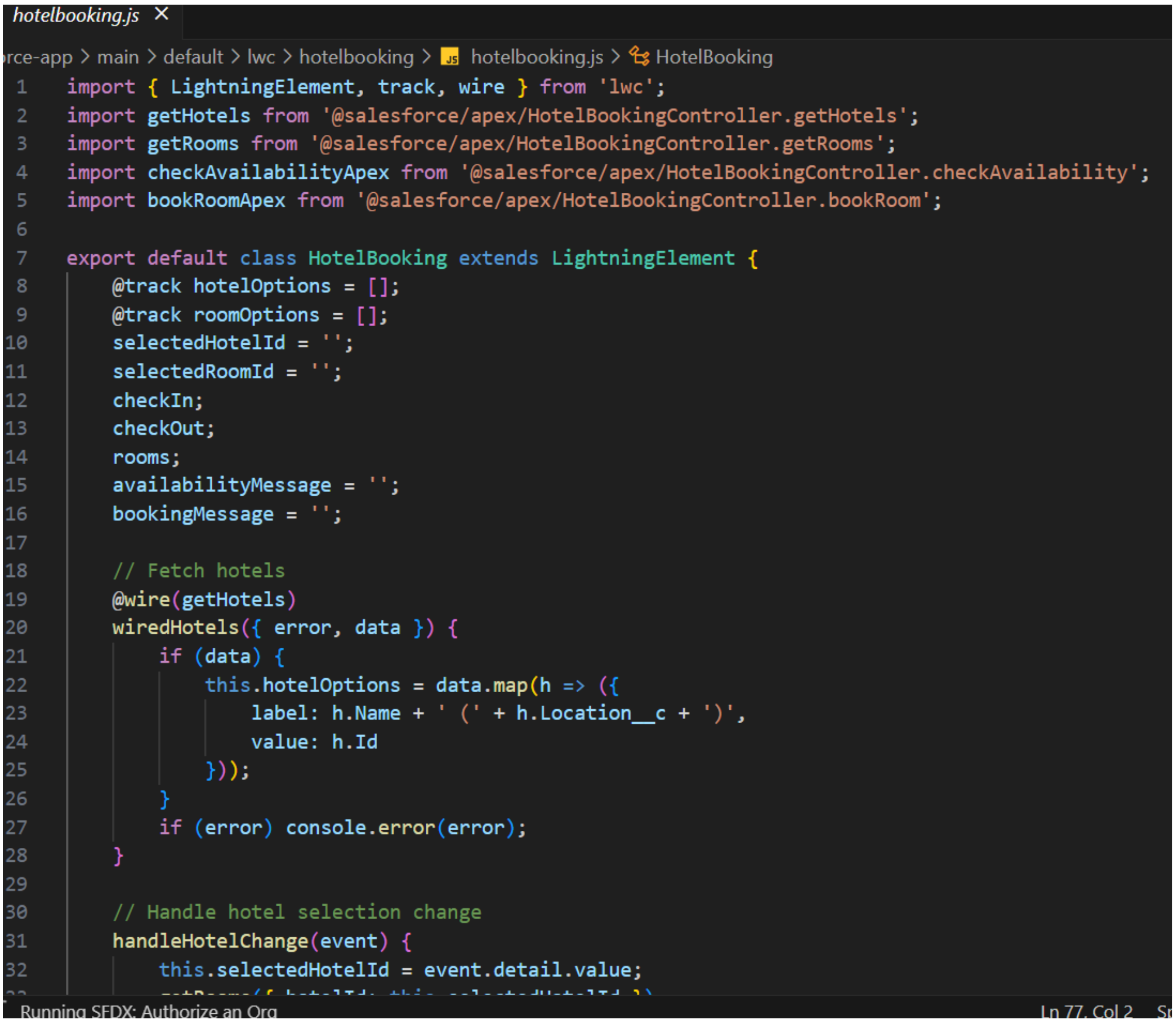
Name	Label	Type	Namespace Prefix	Api Version
fraudScoreViewer	fraudScoreViewer	LWC		64


otelbooking.html

e-app > main > default > lwc > hotelbooking > hotelbooking.html > ...

```html
<template>
    <lightning-card title="Smart Hotel Booking">
        <div class="slds-p-around_medium">

            <!-- Hotel Selection -->
            <lightning-combobox
                label="Select Hotel"
                value={selectedHotelId}
                options={hotelOptions}
                onchange={handleHotelChange}>
            </lightning-combobox>

            <!-- Room Selection -->
            <lightning-combobox
                label="Select Room"
                value={selectedRoomId}
                options={roomOptions}
                onchange={handleRoomChange}>
            </lightning-combobox>

            <!-- Dates and Number of Rooms -->
            <lightning-input type="date" label="Check-In" value={checkIn} onchange={handleCheckInChange}><
            <lightning-input type="date" label="Check-Out" value={checkOut} onchange={handleCheckOutChange
            <lightning-input type="number" label="Number of Rooms" value={rooms} onchange={handleRoomsChan

            <!-- Check Availability -->
            <lightning-button label="Check Availability" onclick={checkAvailability}></lightning-button>
            <p if:true={availabilityMessage}>{availabilityMessage}</p>

            <!-- Book Room -->
            <lightning-button variant="brand" label="Book Now" onclick={bookRoom}></lightning-button>
            <p if:true={bookingMessage}>{bookingMessage}</p>
```

Running SFDX: Authorize an Org   Ln 37, Col 1   Spaces: 4   UTF-8   LF

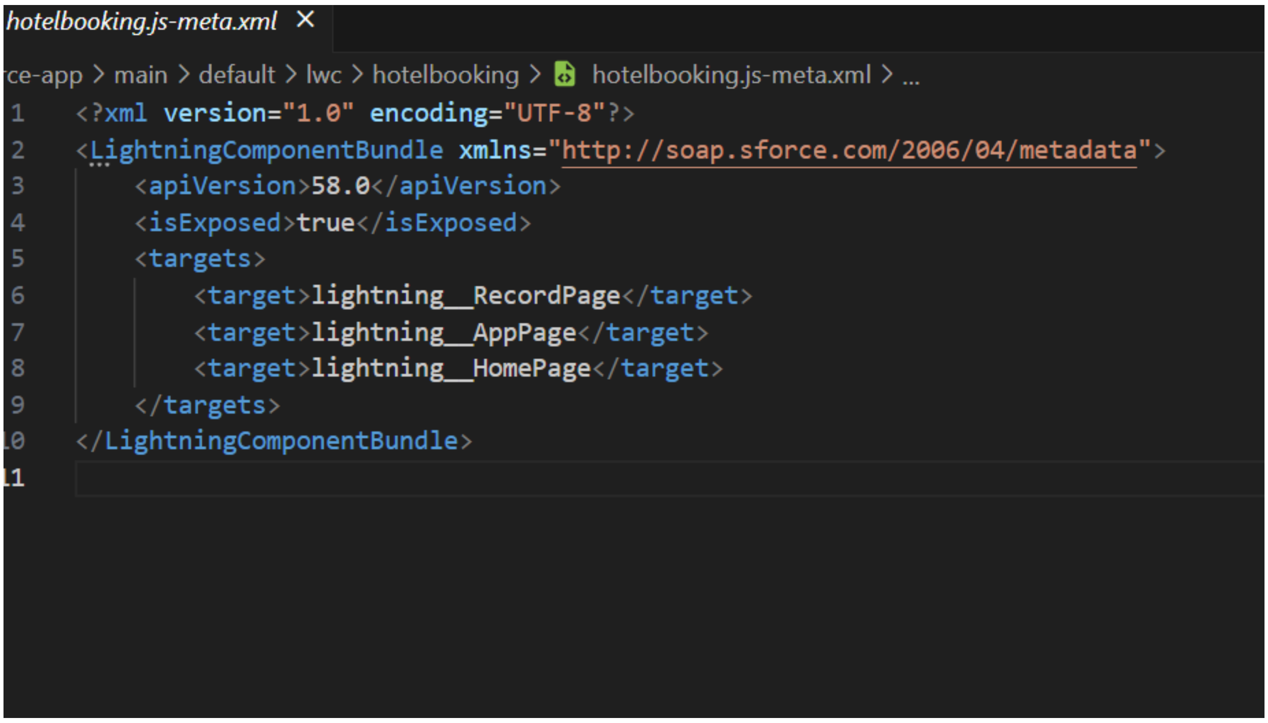


hotelbooking.js ✕

rce-app > main > default > lwc > hotelbooking > hotelbooking.js > HotelBooking

```javascript
import { LightningElement, track, wire } from 'lwc';
import getHotels from '@salesforce/apex/HotelBookingController.getHotels';
import getRooms from '@salesforce/apex/HotelBookingController.getRooms';
import checkAvailabilityApex from '@salesforce/apex/HotelBookingController.checkAvailability';
import bookRoomApex from '@salesforce/apex/HotelBookingController.bookRoom';

export default class HotelBooking extends LightningElement {
    @track hotelOptions = [];
    @track roomOptions = [];
    selectedHotelId = '';
    selectedRoomId = '';
    checkIn;
    checkOut;
    rooms;
    availabilityMessage = '';
    bookingMessage = '';

    // Fetch hotels
    @wire(getHotels)
    wiredHotels({ error, data }) {
        if (data) {
            this.hotelOptions = data.map(h => ({
                label: h.Name + ' (' + h.Location__c + ')',
                value: h.Id
            }));
        }
        if (error) console.error(error);
    }

    // Handle hotel selection change
    handleHotelChange(event) {
        this.selectedHotelId = event.detail.value;
```

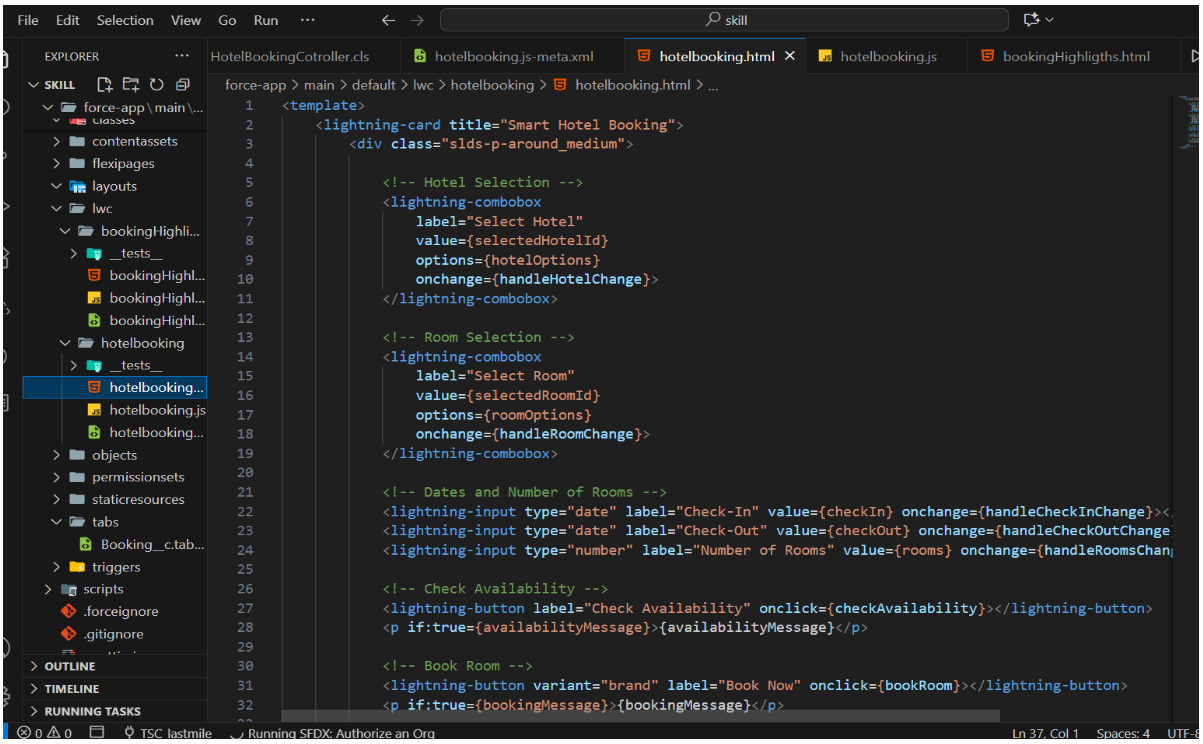
Running SFDX: Authorize an Org   Ln 77, Col 2



hotelbooking.js-meta.xml ✕

ce-app > main > default > lwc > hotelbooking > hotelbooking.js-meta.xml > ...

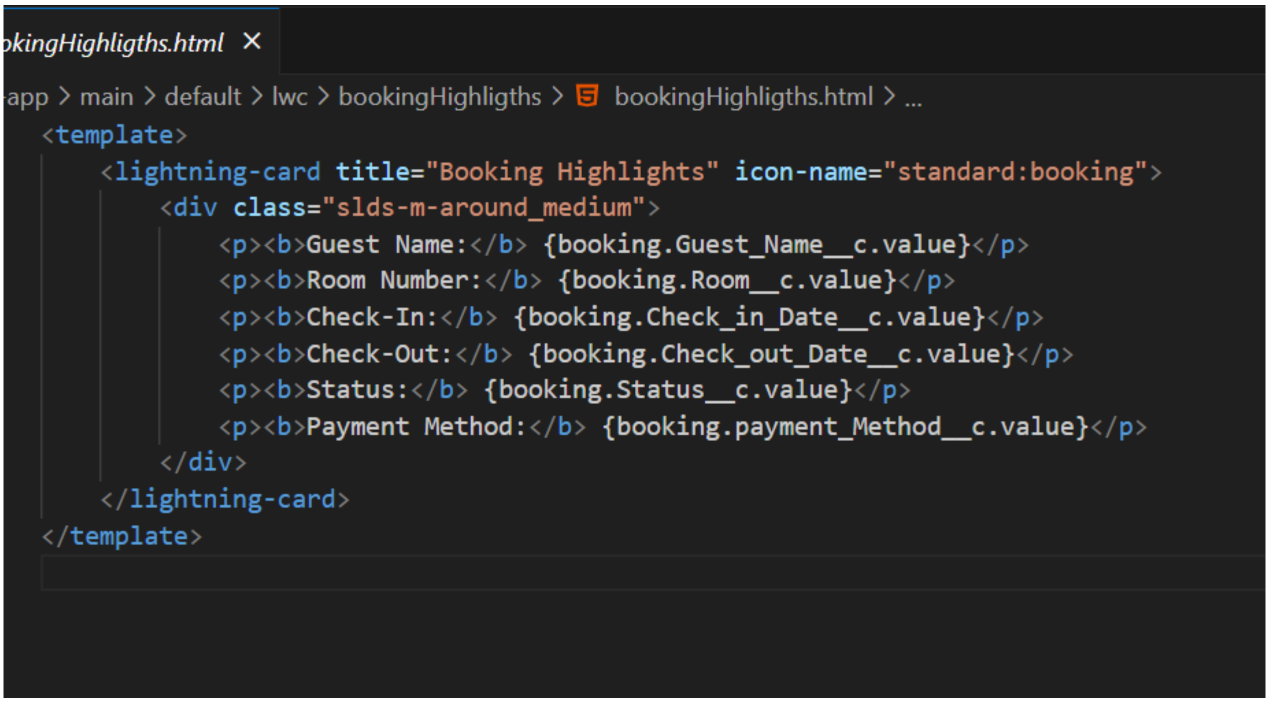
```xml
<?xml version="1.0" encoding="UTF-8"?>
<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
    <apiVersion>58.0</apiVersion>
    <isExposed>true</isExposed>
    <targets>
        <target>lightning__RecordPage</target>
        <target>lightning__AppPage</target>
        <target>lightning__HomePage</target>
    </targets>
</LightningComponentBundle>
```



```
<template>
    <lightning-card title="Smart Hotel Booking">
        <div class="slds-p-around_medium">

            <!-- Hotel Selection -->
            <lightning-combobox
                label="Select Hotel"
                value={selectedHotelId}
                options={hotelOptions}
                onchange={handleHotelChange}>
            </lightning-combobox>

            <!-- Room Selection -->
            <lightning-combobox
                label="Select Room"
                value={selectedRoomId}
                options={roomOptions}
                onchange={handleRoomChange}>
            </lightning-combobox>

            <!-- Dates and Number of Rooms -->
            <lightning-input type="date" label="Check-In" value={checkIn} onchange={handleCheckInChange}><
            <lightning-input type="date" label="Check-Out" value={checkOut} onchange={handleCheckOutChange
            <lightning-input type="number" label="Number of Rooms" value={rooms} onchange={handleRoomsChan

            <!-- Check Availability -->
            <lightning-button label="Check Availability" onclick={checkAvailability}></lightning-button>
            <p if:true={availabilityMessage}>{availabilityMessage}</p>

            <!-- Book Room -->
            <lightning-button variant="brand" label="Book Now" onclick={bookRoom}></lightning-button>
            <p if:true={bookingMessage}>{bookingMessage}</p>
```

❖ **bookingHighlights:**

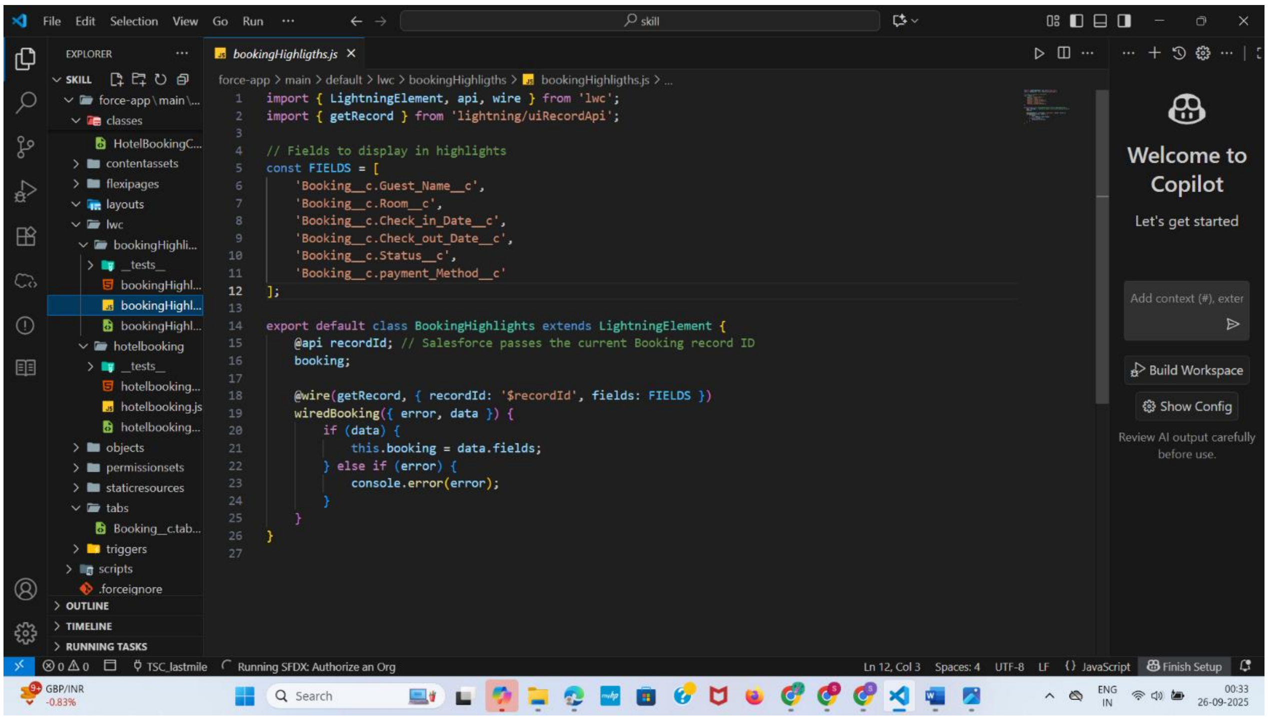
- **Purpose:** Shows key booking metrics such as total bookings, upcoming bookings, and fully booked hotels.
- **Usage:** Added to Home Page, App Page, or Hotel record page**.**
- **Why:** Provides Managers and staff with quick insights into hotel occupancy and booking trends without navigating multiple pages.

- 

kingHighligths.html ×

app > main > default > lwc > bookingHighligths > bookingHighligths.html > ...

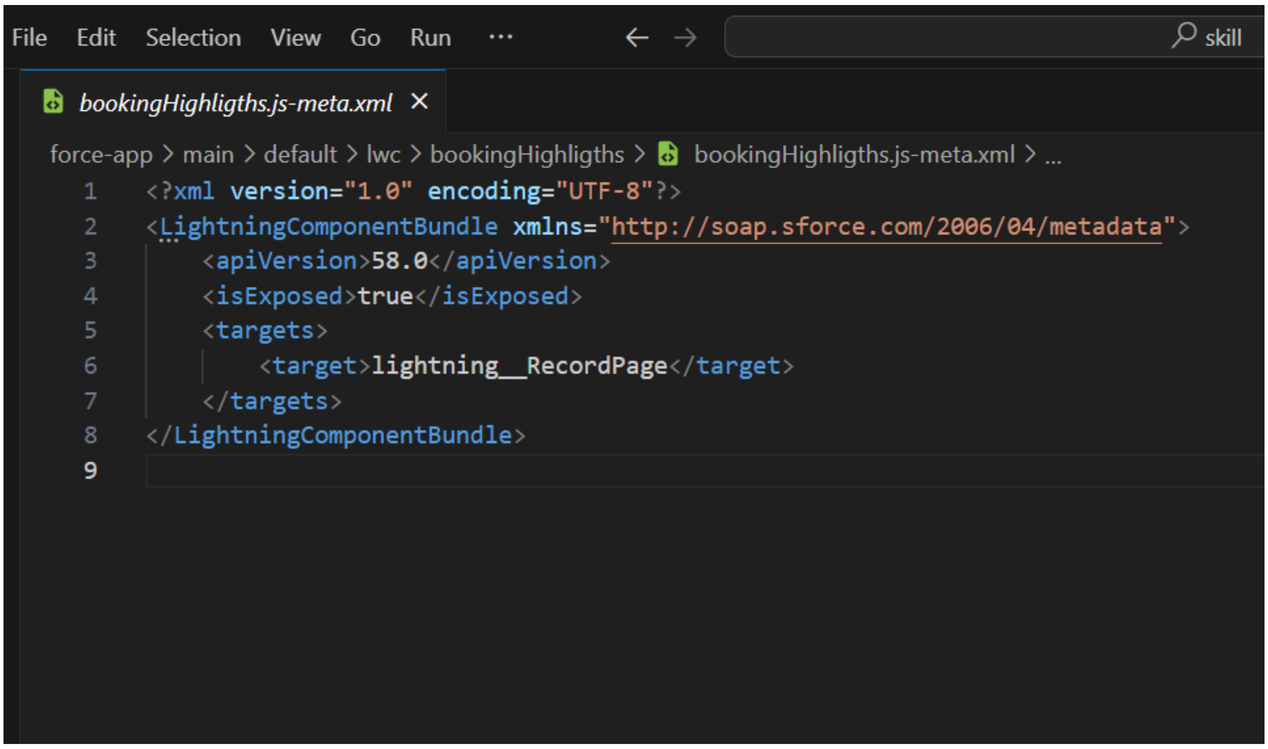
```html
<template>
    <lightning-card title="Booking Highlights" icon-name="standard:booking">
        <div class="slds-m-around_medium">
            <p><b>Guest Name:</b> {booking.Guest_Name__c.value}</p>
            <p><b>Room Number:</b> {booking.Room__c.value}</p>
            <p><b>Check-In:</b> {booking.Check_in_Date__c.value}</p>
            <p><b>Check-Out:</b> {booking.Check_out_Date__c.value}</p>
            <p><b>Status:</b> {booking.Status__c.value}</p>
            <p><b>Payment Method:</b> {booking.payment_Method__c.value}</p>
        </div>
    </lightning-card>
</template>
```



bookingHighligths.js ×

force-app > main > default > lwc > bookingHighligths > bookingHighligths.js > ...

```javascript
import { LightningElement, api, wire } from 'lwc';
import { getRecord } from 'lightning/uiRecordApi';

// Fields to display in highlights
const FIELDS = [
    'Booking__c.Guest_Name__c',
    'Booking__c.Room__c',
    'Booking__c.Check_in_Date__c',
    'Booking__c.Check_out_Date__c',
    'Booking__c.Status__c',
    'Booking__c.payment_Method__c'
];

export default class BookingHighlights extends LightningElement {
    @api recordId; // Salesforce passes the current Booking record ID
    booking;

    @wire(getRecord, { recordId: '$recordId', fields: FIELDS })
    wiredBooking({ error, data }) {
        if (data) {
            this.booking = data.fields;
        } else if (error) {
            console.error(error);
        }
    }
}
```


File
Edit
Selection
View
Go
Run
skill
bookingHighligths.js-meta.xml
force-app > main > default > lwc > bookingHighligths > bookingHighligths.js-meta.xml > ...

```xml
<?xml version="1.0" encoding="UTF-8"?>
<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
    <apiVersion>58.0</apiVersion>
    <isExposed>true</isExposed>
    <targets>
        <target>lightning__RecordPage</target>
    </targets>
</LightningComponentBundle>
```

5. Apex with LWC Integration

- **What was done:**
    - Created Apex Controllers (HotelBookingController) to fetch hotels, rooms, and booking data, and perform booking operations.
    - Granted Apex Class Access to **Customer, Receptionist, and Manager profiles**.
- **Used:**
    - LWCs cannot directly query complex data or perform business logic beyond LDS; Apex handles queries and operations.
- **Steps Followed:**
    1. Setup → Apex Classes → Created controllers.
    2. Setup → Profiles → Added Apex Class Access.

---

6. Events in LWC

- **Usage:**

  - Used custom events (roombooked) in hotelBooking to notify parent components or refresh availability panels.
- **Why:**
  - Provides real-time refresh of room availability after a booking without reloading the page.

---

7. Wire Adapters & Imperative Calls

- **Wire Adapters:**
  - Used @wire with Apex methods getHotels and getRooms.
  - Why: Automatically refreshes UI when data changes (e.g., new hotels or rooms added).
- **Imperative Calls:**
  - Used for actions like checkAvailability and bookRoom.
  - **Why**: Offers flexibility to trigger operations only after user action.

---

8. Navigation Service

- **Usage:**
  - Used NavigationMixin.Navigate to navigate from booking tables or hotel lists to record pages (Hotel, Room, Booking).
- **Why:**
  - Improves usability by allowing users to jump directly to record details without searching manually**.**

---

9. Permissions & Profiles

- **What was done:**

    - Granted Apex Class Access and LWC access to Customer, Receptionist, and Manager profiles.
    - Mapped Lightning Record Pages and Home Pages to Hotel Booking App.
- **Why:**
    - Ensures only authorized users can perform bookings or view sensitive hotel/booking information.
