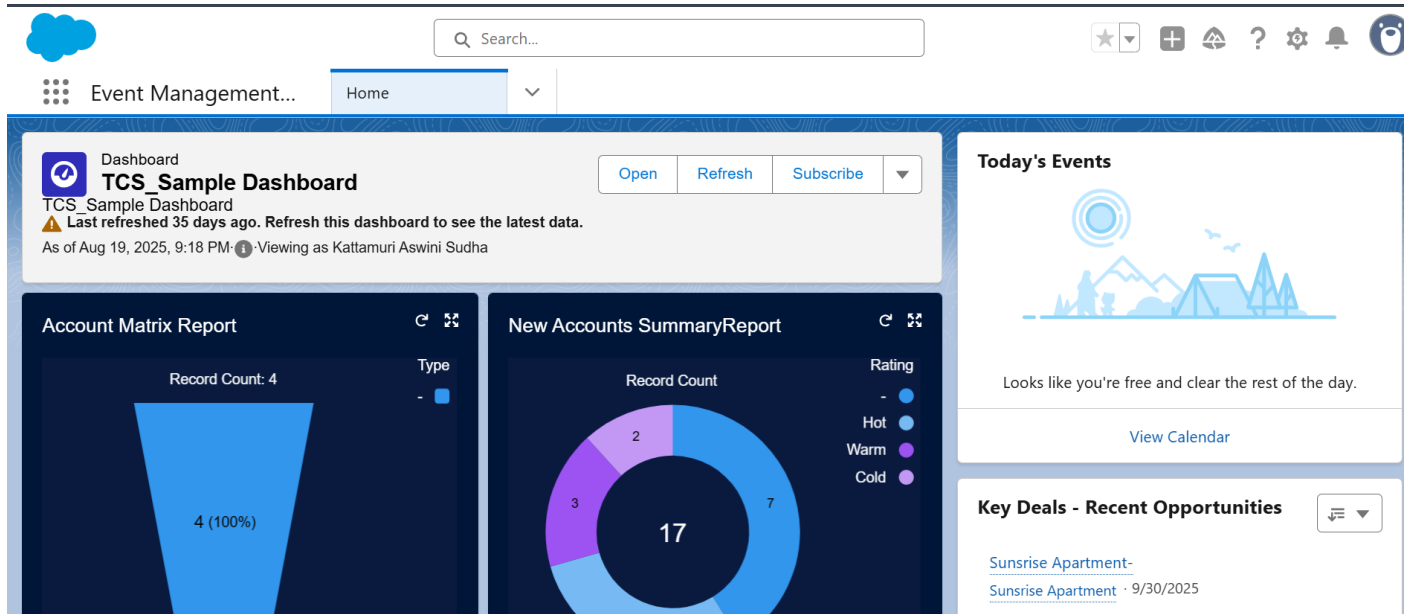


# Phase 6: User Interface Development

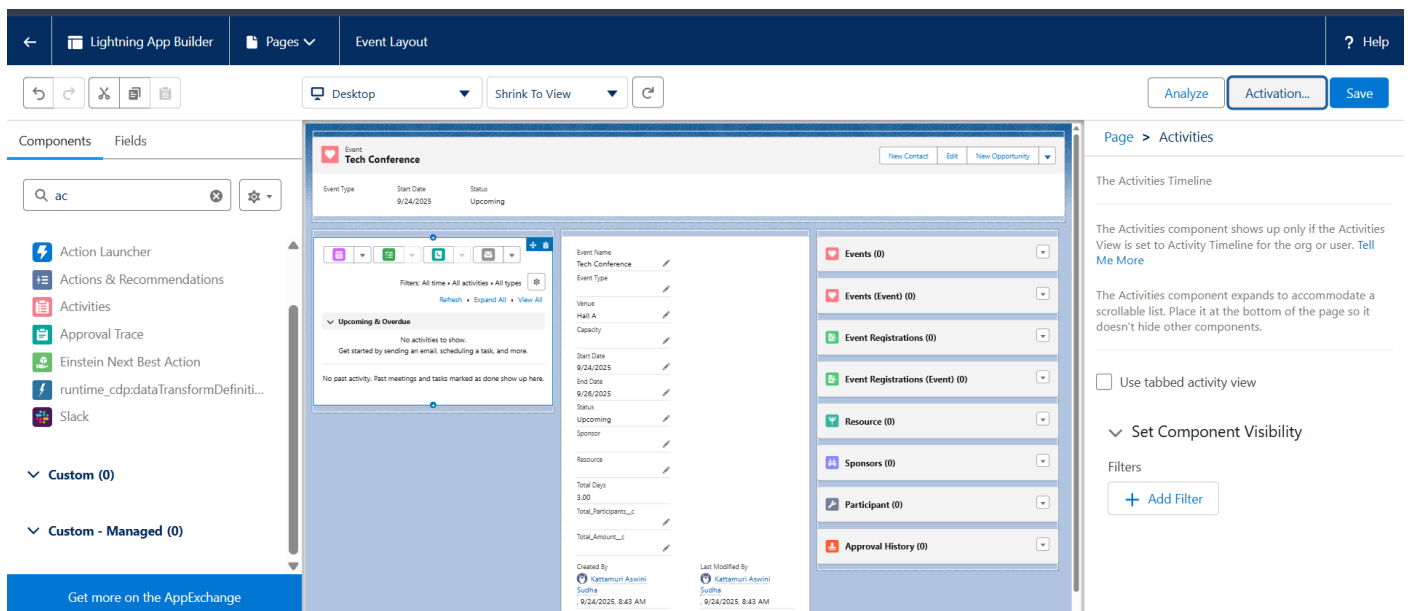
## Step 1: Lightning App Builder – Create the App

- Action: Create a Lightning App → name it Event Management App.



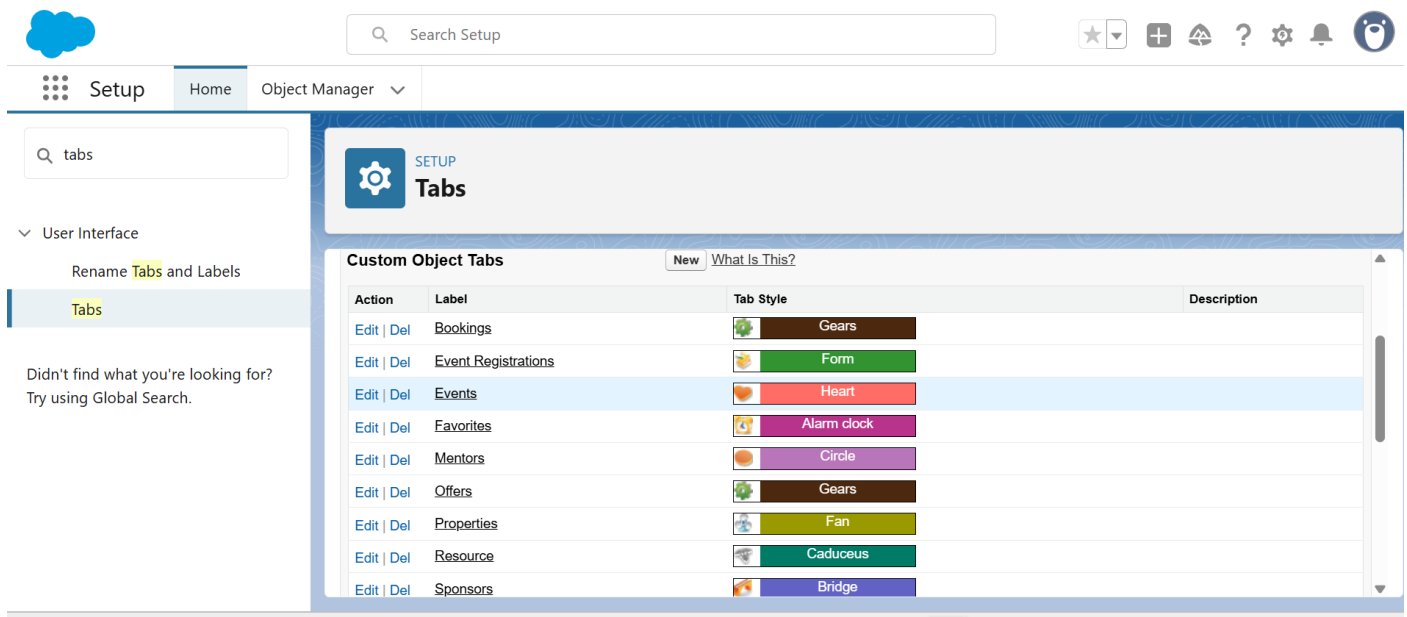
## Step 2: Record Page

- Event page:
  - Display a **list of Bookings** related to each Event.
  - Use **Related Lists** or **LWC datatable** to show bookings.
- Booking page:
  - Show **booking details**, including linked Event & Customer info.



## Step 3: Tabs

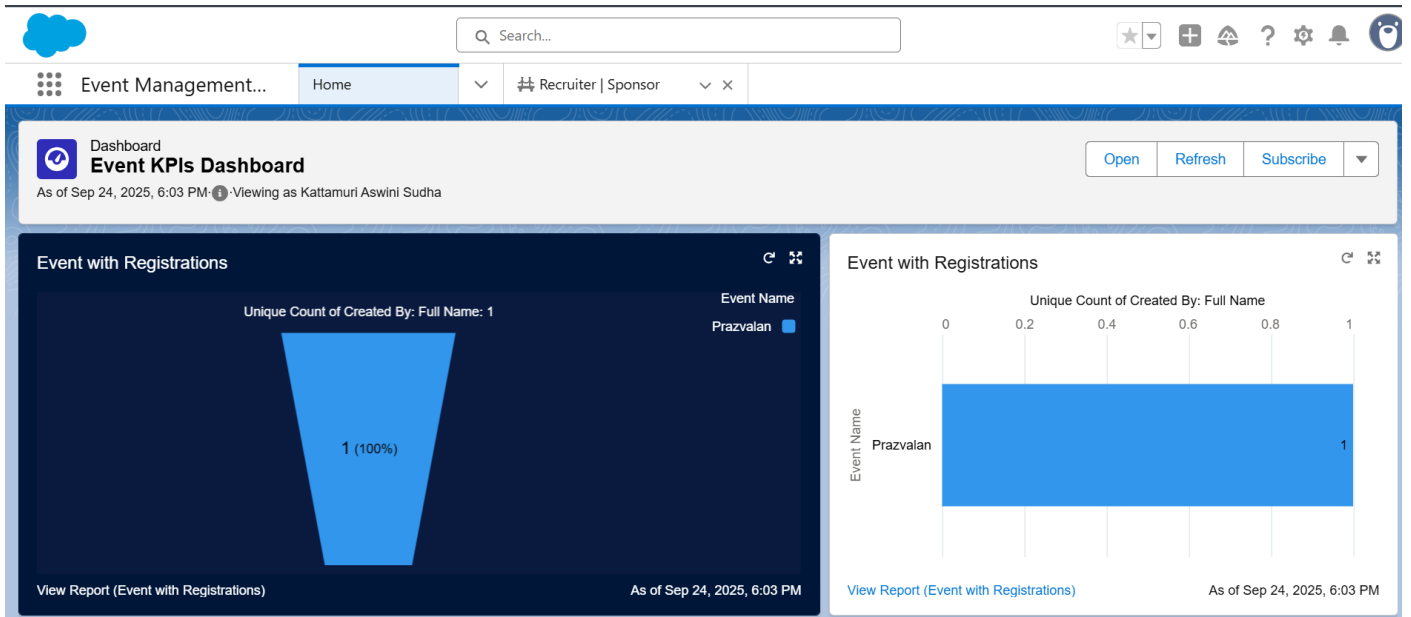
- Add these tabs to the app:
  1. **Events** → list all events.
  2. **Bookings** → list all bookings.
- Users can easily switch between events & bookings.



## Step 4: Home Page Layouts

### Add a dashboard:

- Show events summary (upcoming events, total bookings, etc.).
- Can use Reports & Dashboards or LWC chart components.



## Step 5 : Utility Bar

### Add quick action:

- Example: New Booking button.
- Users can create a booking from any page without leaving it.

The screenshot shows the 'Lightning App Builder' interface for configuring the 'Event Management Home Page'. The top navigation bar includes 'Lightning App Builder', 'Pages', 'Event Management Home Page', and 'Help'. The main workspace displays a preview of the dashboard layout, which includes the 'Event KPIs Dashboard' and a table of event registrations.

**Components Panel:**

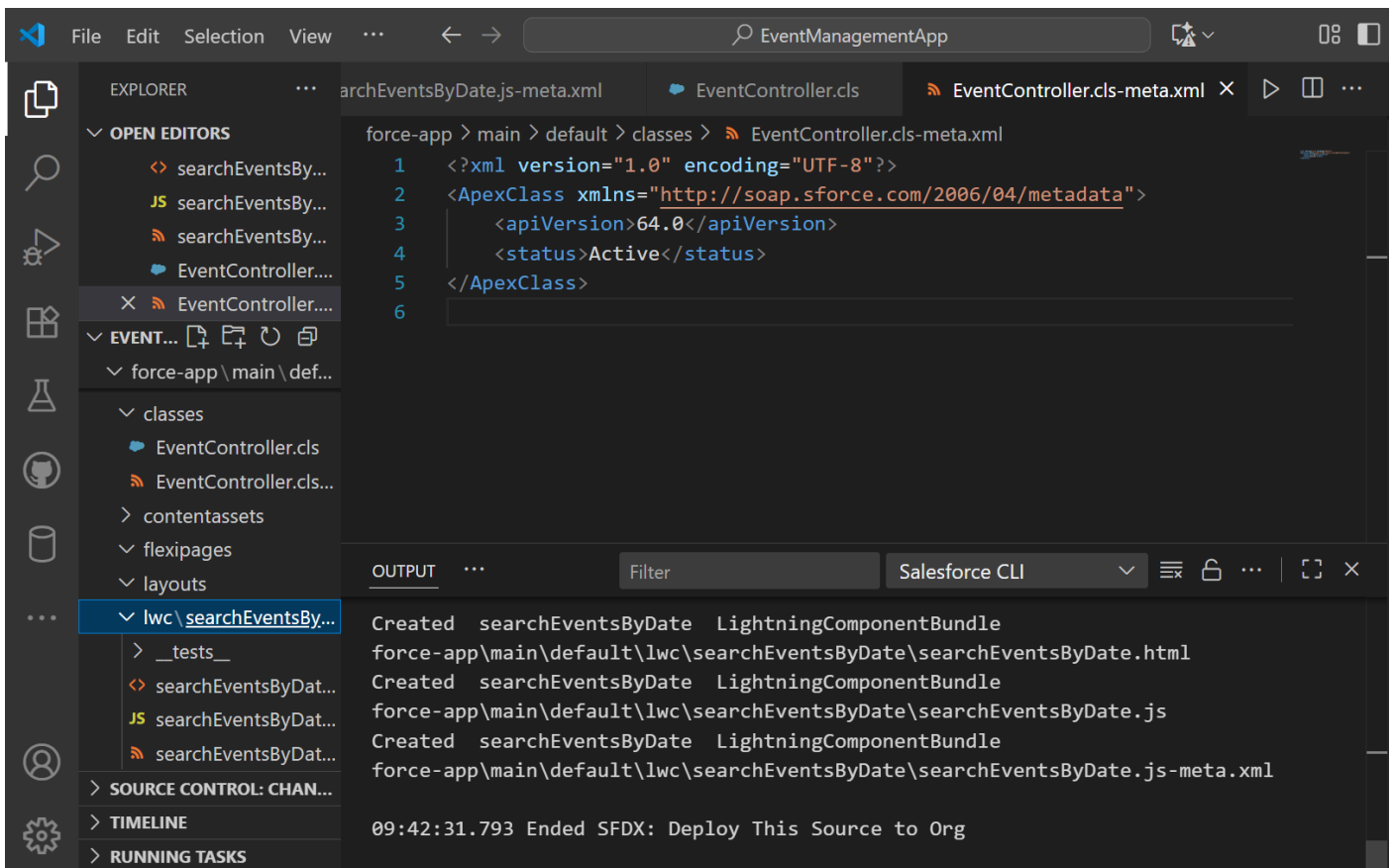
- Search:** Search...
- Standard (43):**
  - Accordion
  - App Launcher
  - Assistant
  - Cdp Metrics Overview
  - Chatter Feed
  - Chatter Publisher
  - CRM Analytics Collection
- Get more on the AppExchange**

**Page Configuration Panel:**

- Label:** Event Management Home Page
- API Name:** Event\_Management\_Home\_Page
- Page Type:** Home Page
- Template:** Header and Three Regions (Change)
- Description:**

## Step 6: LWC – Search Events by Date

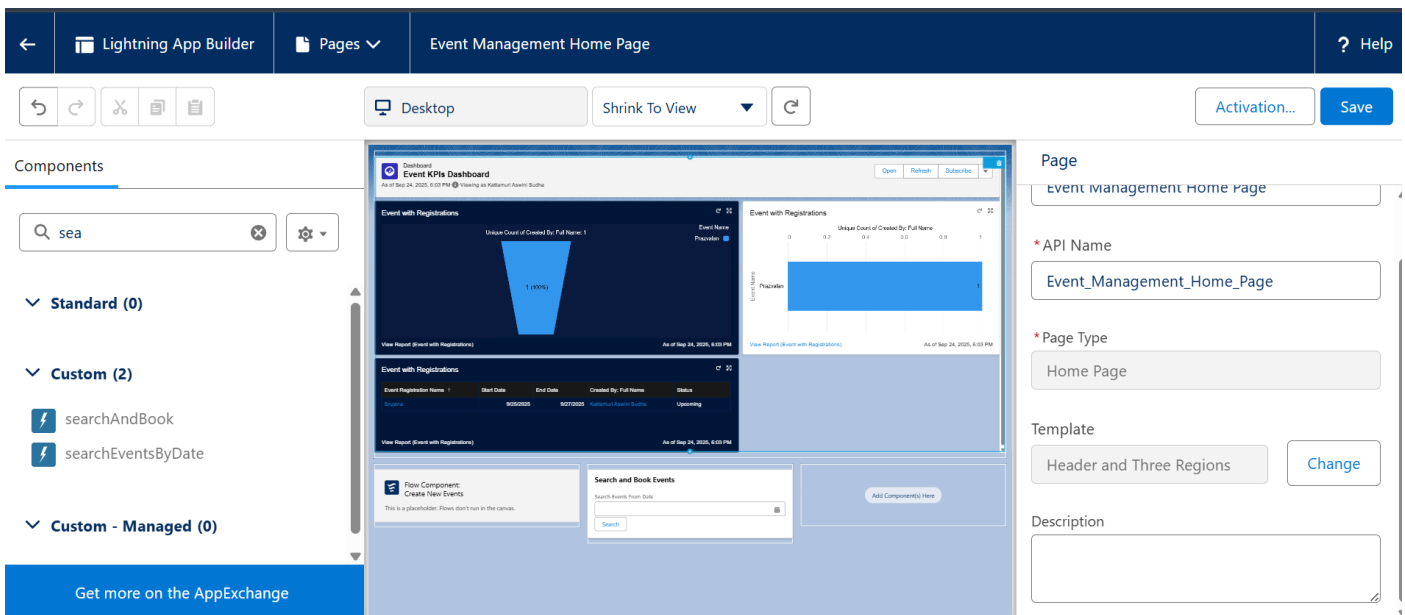
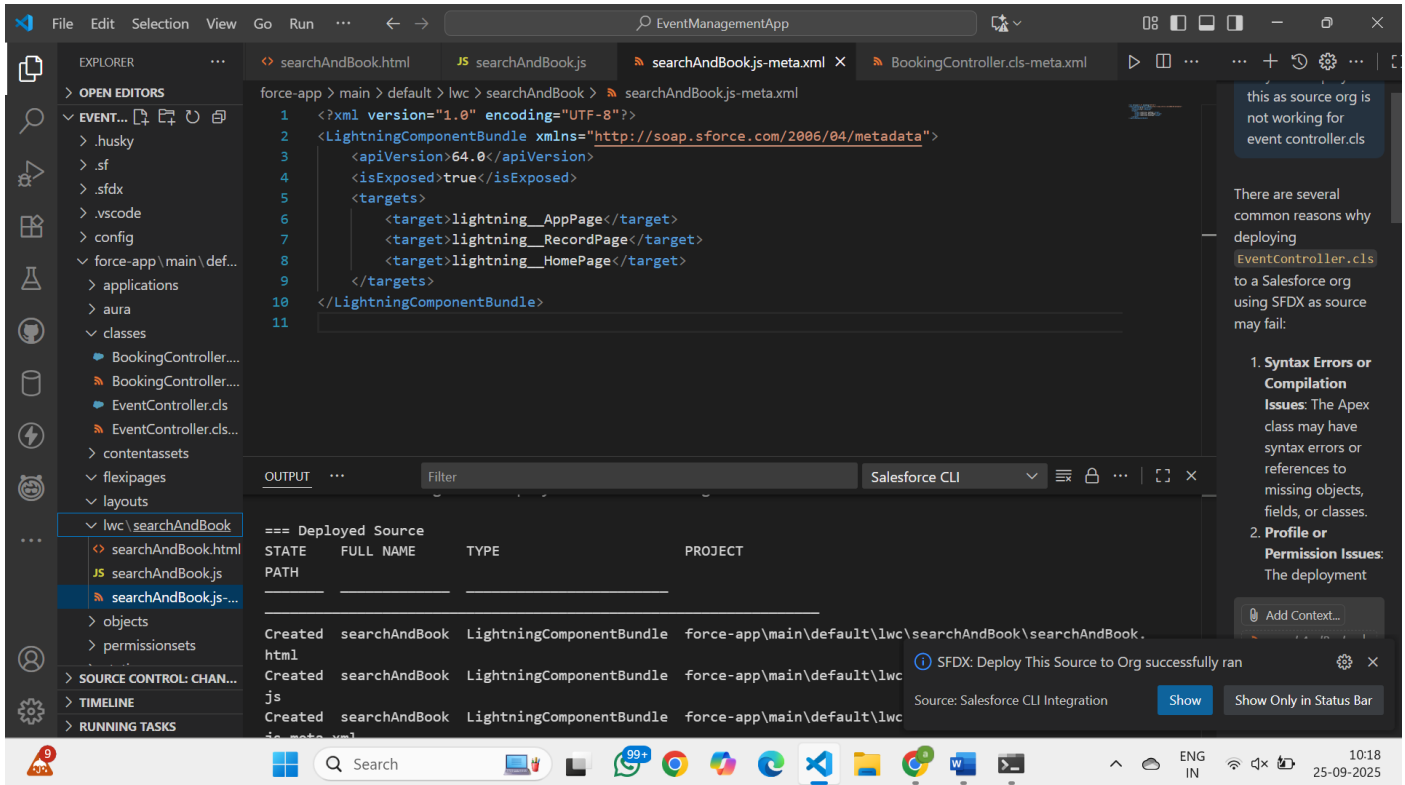
- **Component:** Search Events by date.
- **Functionality:**
  - Input start date & end date.
  - Show upcoming events in datatable.
- **Purpose:** Users can quickly find events to book.



# Apex with LWC – Create Booking

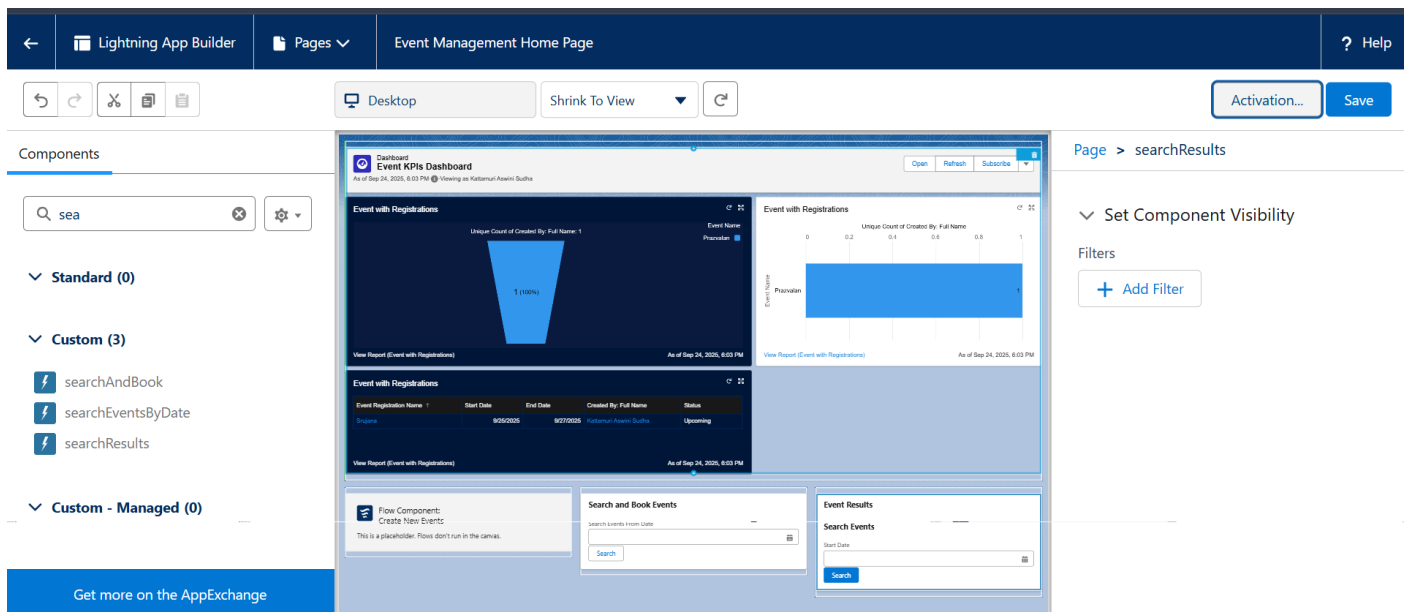
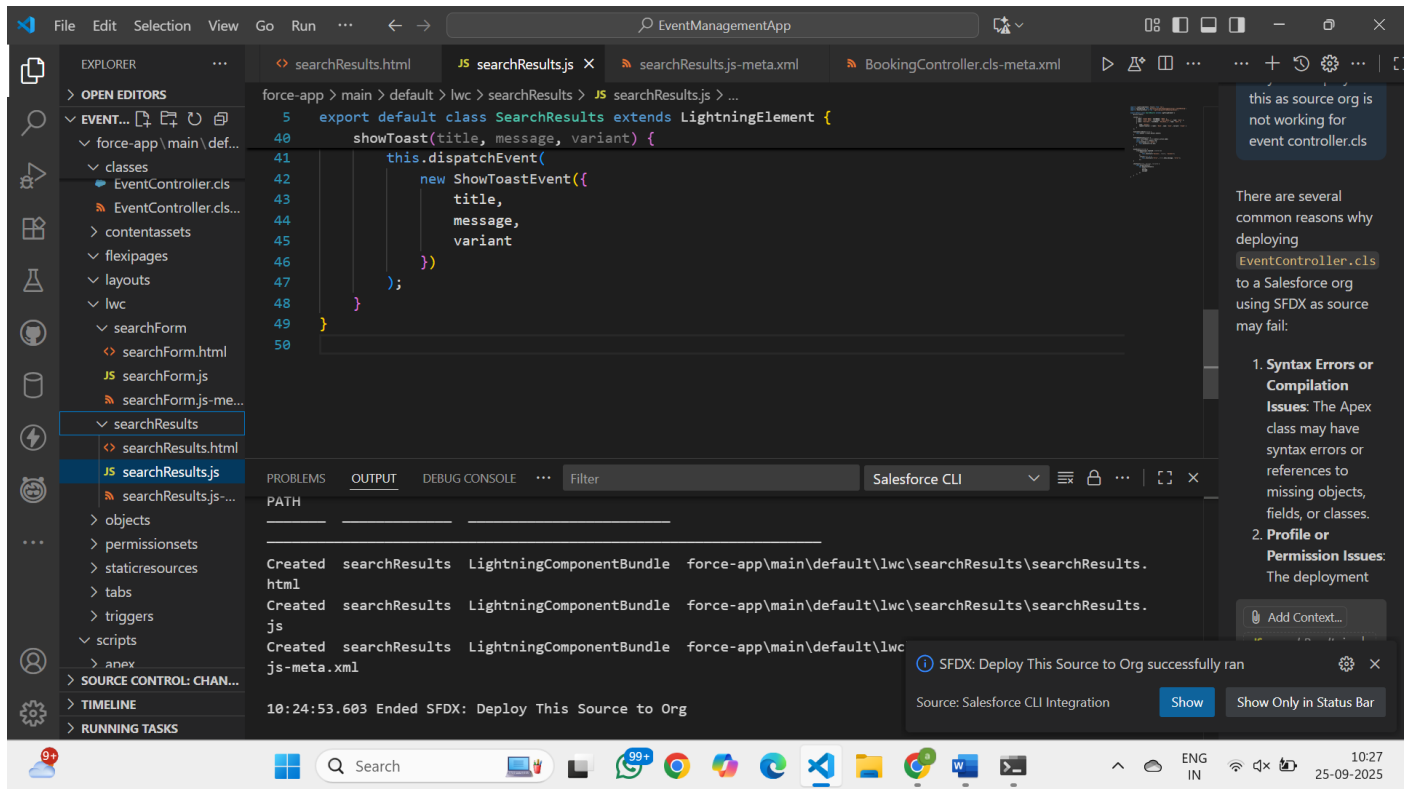
- Imperative Apex call:

- LWC calls Apex when user clicks Book Now.
- Apex inserts a new Booking record and returns its Id



## Step 8: Events in LWC

- **Child → Parent communication:**
  - Child component: search form.
  - Parent component: receives results & displays in datatable.
- **Purpose: Keep UI modular and reactive.**



## Step 9: Wire Adapters

**Function:** Fetch upcoming events automatically from Apex.

The screenshot shows the Visual Studio Code editor with the following components:

- Explorer:** Shows the project structure with folders for 'force-app/main/default' and 'lwc/upcomingEvents'. The 'EventController.cls' file is selected.
- Editor:** Displays the Apex class `EventController` with the following code:

```
public with sharing class EventController {
    @AuraEnabled(cacheable=true)
    public static List<Event__c> getUpcomingEvents() {
        return [
            SELECT Id, Name, Start_Date__c, End_Date__c
            FROM Event__c
            WHERE Start_Date__c >= TODAY
            ORDER BY Start_Date__c ASC
        ];
    }
}
```
- Problems:** Shows a message: "this as source org is not working for event controller.cls".
- Output:** Shows the deployment path for the component: `force-app/main/default/lwc/upcomingEvents/upcomingEvents.html`.
- Debug Console:** Shows the deployment status: "SFDX: Deploy This Source to Org successfully ran".
- Terminal:** Shows the deployment command: `10:32:07.778 Ended SFDX: Deploy This Source to Org`.

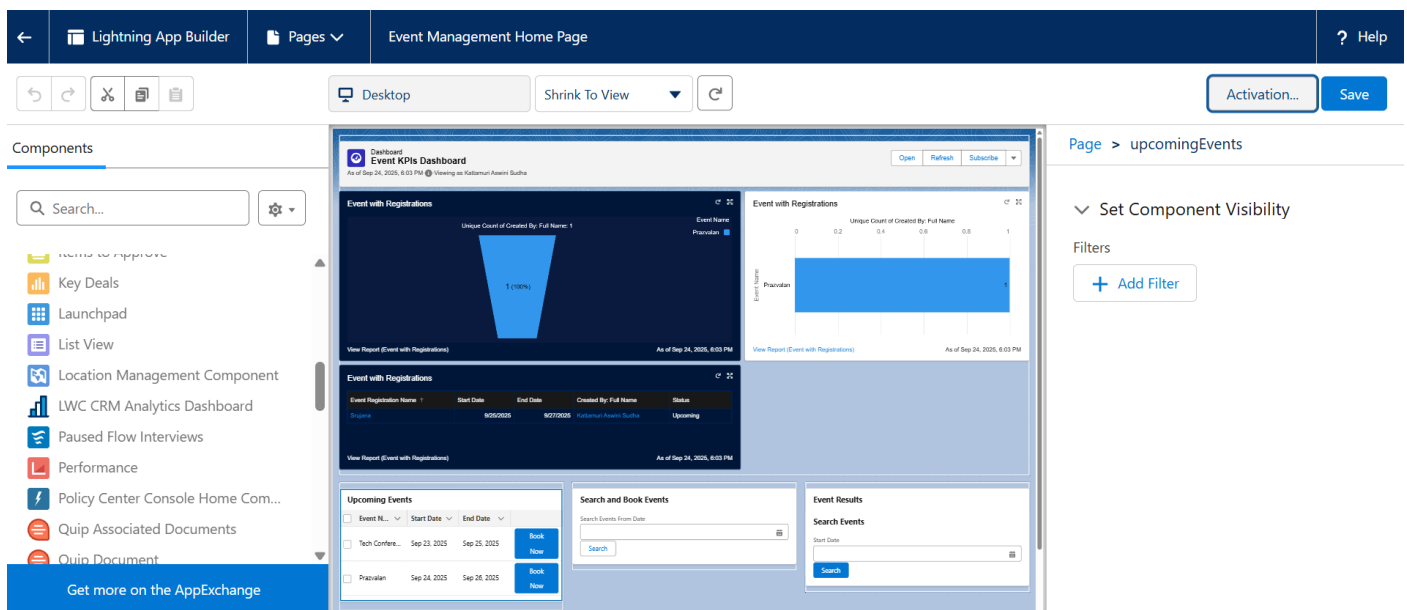
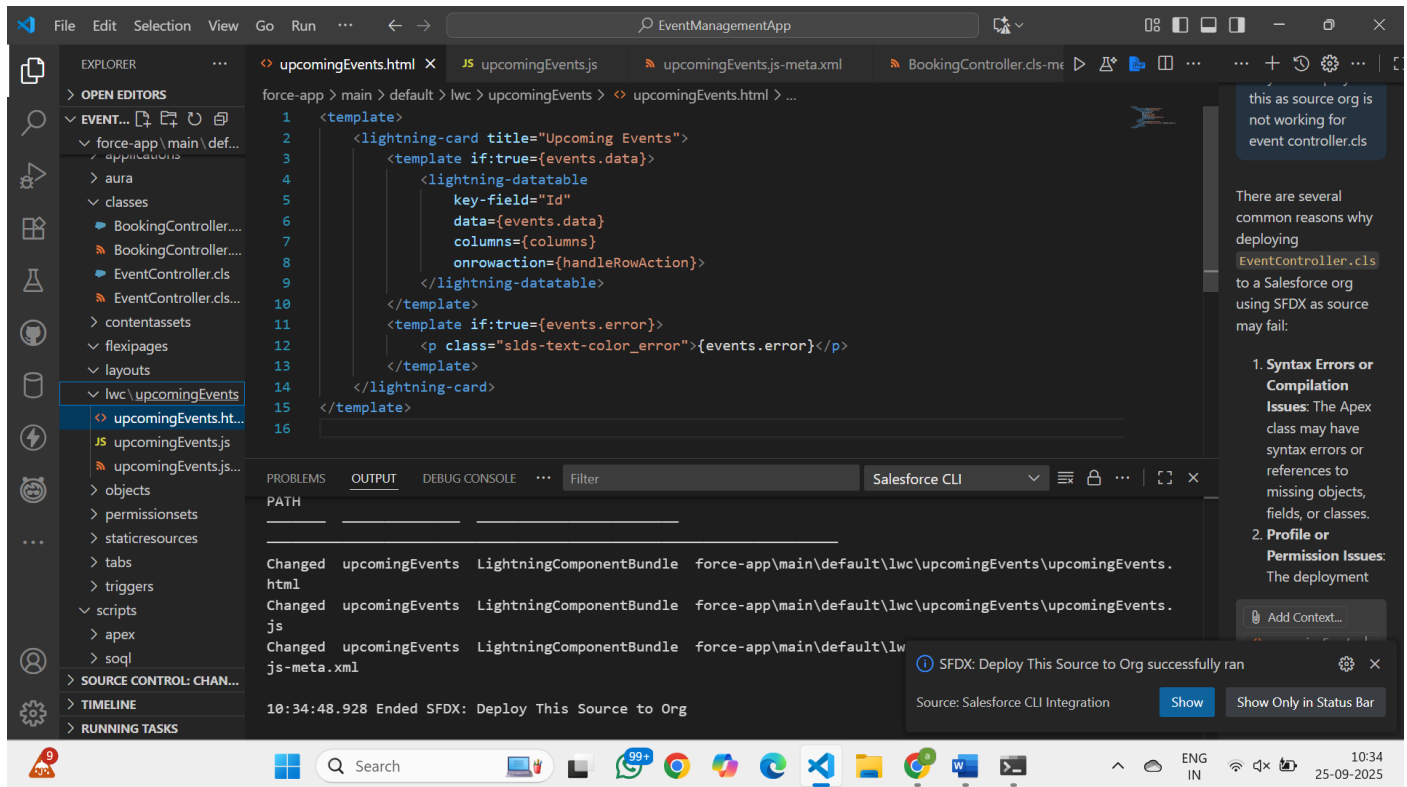
The screenshot shows the Lightning App Builder interface for the 'Event Management Home Page'. The page layout includes:

- Header:** 'Lightning App Builder' and 'Event Management Home Page'.
- Components:** A list of components on the left, including 'Standard (2)' (Inventory Lookup Component, setup\_cdp:cdpEducationalGuidance) and 'Custom (1)' (upcomingEvents).
- Page:** The main content area showing the 'upcomingEvents' component. It displays a dashboard with two charts: 'Event with Registrations' and 'Event with Registrations'. The charts show a unique count of created by full name, with a value of 1 (100%).
- Filters:** A section on the right titled 'Set Component Visibility' with a 'Filters' section and an 'Add Filter' button.

## Step 10: Imperative Apex Calls

🔖 Used when action needed, like booking an event.

🔖 Called on button click, not automatically.





**Purpose:** After booking, navigate to Booking record page.

**Purpose:** After booking, navigate to Booking record page.

