

# • Smart Event Management & Ticketing System

## • Problem Statement

Event organizers face multiple challenges:

- Ticket sales are fragmented across different channels (online, offline, sponsors).
- No centralized system to manage seat allocation, VIP access, cancellations, and refunds.
- Sponsors don't get proper visibility into performance.
- Attendees lack a smooth digital experience (QR tickets, real-time updates, group discounts).

To solve this, we want a system to:

- Centralize event, ticket, and sponsor management.
- Automate approval workflows for VIP tickets and cancellations.
- Enable digital payments and QR-based entry passes.
- Provide real-time dashboards for ticket sales, sponsorship revenue, and attendee engagement.

## Phase 1: Problem Understanding & Industry Analysis

### 1 Requirement Gathering

- Capture event details (type, date, location, capacity).
- Manage ticket categories (General, VIP, Early Bird, Group Discounts).
- Automate seat allocation and group bookings.
- Track sponsorship deals and benefits.
- Send digital tickets/QR codes to attendees.
- Enable refunds and cancellations with approval workflows.
- Provide dashboards for ticket sales, revenue, and attendance.

### 2 Stakeholder Analysis

- Event Organizers: Need event setup, sponsor management, ticket monitoring.
- Attendees: Need smooth booking, payments, QR entry, and refund options.
- Sponsors: Need visibility into event reach audience size, and engagement.
- Administrators (CRM Managers): Ensure secure access, data integrity, and reporting.

### 3 Business Process Mapping

#### Current:

- Tickets sold via third-party sites or offline.
- Manual sponsor agreement tracking.
- Attendees face delays in receiving confirmation.
- No unified dashboard for revenue analysis.

#### Proposed:

- Centralized Event, Ticket, Sponsor objects.
- Automated VIP approvals and refund workflows.
- QR-code ticketing + calendar sync for attendees.
- Dashboards for organizers & sponsors to track revenue, seats, and engagement.

## 4 Industry-specific Use Case Analysis

- Concerts & Music Festivals ☐ VIP, General, Early Bird ticketing.
- Corporate Conferences ☐ Multi-track event scheduling, sponsor branding.
- Sports Events ☐ Seat allocation and bulk ticket sales.
- Weddings & Private Events ☐ Guest management with QR invitations.

## 5 AppExchange Exploration

- Explore apps for payment gateways (Razorpay).
- Explore apps for QR code generation.
- Check Event Management accelerators available on AppExchange and customize instead of reinventing everything.

## Phase 2: Org Setup & Configuration

- This phase focused on setting up and configuring the Salesforce Developer Org for the Smart Event Management Project. The goal was to establish a solid foundation for the CRM system, ensuring proper organizational structure, security, and access controls. The configuration was carried out entirely using Salesforce’s declarative (point-and-click) tools without coding.

### Steps Completed in Phase 2:

#### 1. Company Profile Setup

SETUP

Company Information

Company Information

Smart Event Management CRM

The organization's profile is below.

[User Licenses \(104\)](#) | [Permission Set Licenses \(104\)](#) | [Feature Licenses \(11\)](#) | [Usage-based Entitlements \(104\)](#)

Organization Detail

Edit

Organization Name	Smart Event Management CRM	Phone	
Primary Contact	OrgFarm EPIC	Fax	
Division		Default Locale	English (United States)
Address	India	Default Language	English
Fiscal Year Starts In	January	Default Time Zone	(GMT+05:30) India Standard Time (Asia/Kolkata)
Activate Multiple Currencies	<input type="checkbox"/>	Currency Locale	English (India) - INR
Enable Data Translation	<input type="checkbox"/>	Used Data Space	342 KB (7%) <a href="#">View</a>
Newsletter	<input checked="" type="checkbox"/>	Used File Space	17 KB (0%) <a href="#">View</a>
Admin Newsletter	<input checked="" type="checkbox"/>	API Requests, Last 24 Hours	14 (15,000 max)
Hide Notices About System Maintenance	<input type="checkbox"/>	Streaming API Events, Last 24 Hours	0 (10,000 max)
Hide Notices About System Downtime	<input type="checkbox"/>	Restricted Logins, Current Month	0 (0 max)
Locale Formats	ICU	Salesforce.com Organization ID	00DgL000008eZWN
		Organization Edition	Developer Edition
		Instance	CAN98

- Configured company information, business hours, and holidays.

#### 2. Business Hours

SETUP

Business Hours

Organization Business Hours

Select the days and hours that your support team is available. These hours, when associated with escalation rules, determine the times at which cases can escalate.

If you enter blank business hours for a day, that means your organization does not operate on that day.

Business Hours Detail

Edit

Business Hours Name

Smart Event Default Hours

Business Hours	Sunday	9:00 AM to 6:00 PM	Time Zone	(GMT+05:30) India Standard Time (Asia/Kolkata)
	Monday	9:00 AM to 6:00 PM	Default Business Hours	<input checked="" type="checkbox"/>
	Tuesday	9:00 AM to 6:00 PM		
	Wednesday	9:00 AM to 6:00 PM		
	Thursday	9:00 AM to 6:00 PM		
	Friday	9:00 AM to 6:00 PM		
	Saturday	9:00 AM to 6:00 PM		

Active

☒

Created By Samhitha.M.C 9/17/2025, 4:36 AM

Last Modified By Samhitha.M.C 9/17/2025, 4:36 AM

Edit

#### 3. Holidays

SETUP

Holidays

Holiday Detail

Holidays are dates and times at which business hours are suspended. These dates and times, when associated with business hours, suspend business hours.

Add or remove business hours to holidays to suspend business hours and escalation rules during the holidays.

Business Hours [0]

Holiday Detail

EditDelete

Holiday Name	NEW YEAR
Description	
Date and Time	1/1/2026 All Day
Recurring Holiday	Occurs every January 1 effective 1/1/2026
Created By	Samhitha M.C 9/17/2025, 4:48 AM

#### 4. User Setup

- Created multiple users with different roles (Event Manager, Sponsor, Attendee).

SETUP

Users

User Edit

John Event Manager

SaveSave & NewCancel

General Information

First NameJohn

Last NameEvent Manager

Aliasjeven

Emailjohneventmanager@gmail

Usernamejohneventmanager@smarte

NicknameUser17581994855334175

Title

Company

Department

RoleEvent Manager

User LicenseSalesforce

ProfileStandard User

Active☒

Marketing User☐

Offline User☐

Knowledge User☐

Flow User☐

Service Cloud User☐

- Event Manager

SETUP

Users

User Edit

Sarah Sponsor

SaveSave & NewCancel

General Information

First NameSarah

Last NameSponsor

Aliassspon

Emailsam+smarteventsponsor@t

Usernameam+sponsor@smartevent

NicknameUser175819973517988547

Title

RoleSponsor

User LicenseSalesforce

ProfileStandard User

Active☒

Marketing User☐

Offline User☐

Knowledge User☐

- Sponsor

SETUP

Users

User Edit

David Attendee

SaveSave & NewCancel

General Information

First NameDavid

Last NameAttendee

Aliasdatte

Emailsam+attendee@gmail.com

Usernameattendee@smartevent.com

NicknameUser175819995070990819

Title

Company

Department

Division

RoleAttendee

User LicenseSalesforce Platform

ProfileStandard Platform User

Active☒

Marketing User☐

Offline User☐

Knowledge User☐

Flow User☐

Service Cloud User☐

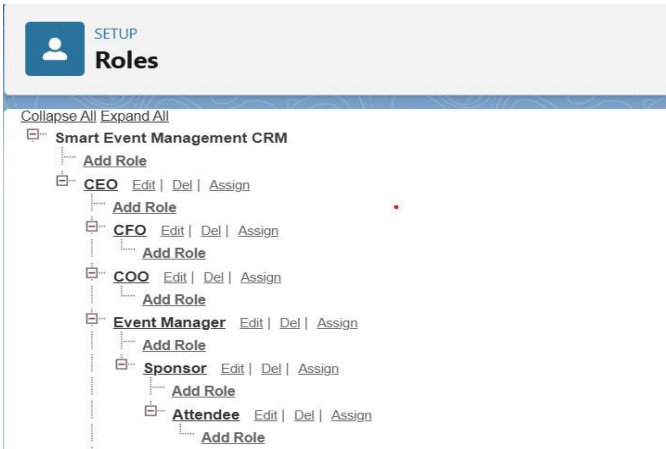
Site.com Contributor User☐

Site.com Publisher User☐

- Attendee

5. Roles

- Defined role hierarchy to manage data visibility across the organization.



6. Profiles

- Created and customized profiles by cloning standard profiles and adjusting object-level permissions.

The screenshot shows the 'Profiles' setup page in Salesforce. At the top, there's a 'SETUP' button and a 'Profiles' header. Below this, there's a 'Profile' section for 'Event Manager Profile'. The text says: 'Users with this profile have the permissions and page layouts listed below. Administrators can change a user's profile. If your organization uses Record Types, use the Edit links in the Record Type Settings section below to make one.' Below this text, there are links for 'Login IP Ranges [0]', 'Enabled Apex Class Access [0]', 'Enabled Visualforce Page Access [0]', 'Enabled External Credential Principal Access [0]', 'Enabled Custom Metadata Type Access [0]', 'Enabled Custom Setting Definition Access [0]', and 'Enabled Custom Permissions [0]'. Below the links, there's a 'Profile Detail' section with buttons for 'Edit', 'Clone', 'Delete', and 'View Users'. The 'Profile Detail' section contains a table with the following information:

Name	Event Manager Profile
User License	Salesforce
Description	
Created By	Samhitha M.C. 9/19/2025, 9:04 PM

- Event Manager Profile

The screenshot shows the 'Profiles' setup page in Salesforce. At the top, there's a 'SETUP' button and a 'Profiles' header. Below this, there's a 'Profile' section for 'Sponsor Profile'. The text says: 'Users with this profile have the permissions and page layouts listed below. Administrators can change a user's profile. If your organization uses Record Types, use the Edit links in the Record Type Settings section below to make one.' Below this text, there are links for 'Login IP Ranges [0]', 'Enabled Apex Class Access [0]', 'Enabled Visualforce Page Access [0]', 'Enabled External Credential Principal Access [0]', 'Enabled Custom Metadata Type Access [0]', 'Enabled Custom Setting Definition Access [0]', and 'Enabled Custom Permissions [0]'. Below the links, there's a 'Profile Detail' section with buttons for 'Edit', 'Clone', 'Delete', and 'View Users'. The 'Profile Detail' section contains a table with the following information:

Name	Sponsor Profile
User License	Salesforce
Description	
Created By	Samhitha M.C. 9/19/2025, 9:12 PM

- Sponsor Profile

SETUP

Profiles

Profile

Attendee Profile

Users with this profile have the permissions and page layouts listed below. Administrators can cha

If your organization uses Record Types, use the Edit links in the Record Type Settings section bel

Login IP Ranges [0]

 | 

Enabled Apex Class Access [0]

 | 

Enabled Visualforce Page Acces

Enabled External Credential Principal Access [0]

 | 

Enabled Custom Metadata Type Access [0]

 | 

Enabled Cu

Enabled Custom

Profile Detail

Edit

Clone

Delete

View Users

Name	Attendee Profile
User License	Salesforce
Description	
Created By	Samhitha M.C, 9/19/2025, 9:13 PM

- Attendee Profile

## 7. Login Access Policies

- Enabled admin login access for troubleshooting and testing.

SETUP

Login Access Policies

Login Access Policies

Control which support organizations your users can grant login access to.

Changes Saved

Manage Support Options

Save

Cancel

Setting	Enabled		
Administrators Can Log in as Any User	<input checked="" type="checkbox"/>		
Support Organization	Packages	Available to Users	Available to Administrators Only
Salesforce.com Support		<input checked="" type="radio"/>	<input type="radio"/>

Save

Cancel

## 8. Sharing Rules

- Created exceptions to OWD (e.g., Event Manager access to Attendees, Sponsors, and Tickets).

SETUP

Sharing Settings

No sharing rules specified.

Work Type Group Sharing Rules

New

Recalculate

Work Type Group Sharing Rules Help

No sharing rules specified.

Attendee Sharing Rules

New

Recalculate

Attendee Sharing Rules Help

Action	Criteria	Shared With	Access Level
<a href="#">Edit</a>   <a href="#">Del</a>	Owner in Role Attendee	Role Event Manager	Read Only

Event Sharing Rules

New

Recalculate

Event Sharing Rules Help

No sharing rules specified.

Sponsor Sharing Rules

New

Recalculate

Sponsor Sharing Rules Help

Action	Criteria	Shared With	Access Level
<a href="#">Edit</a>   <a href="#">Del</a>	Owner in Role Sponsor	Role Event Manager	Read/Write

Ticket Sharing Rules

New

Recalculate

Ticket Sharing Rules Help

Action	Criteria	Shared With	Access Level
<a href="#">Edit</a>   <a href="#">Del</a>	Owner in Role Attendee	Role Event Manager	Read Only

9. Organization wide Defaults(OWD)

- Configured baseline record access (Event = Public Read Only, Attendee/Sponsor/Ticket = Private).

Attendee	Private	Private	✓
Event	Public Read Only	Public Read Only	✓
Sponsor	Private	Private	✓

10. Permission Sets

- Assigned extra permissions without modifying base profiles.

Custom Object Permissions									
	Basic Access				Data Administration				
	Read	Create	Edit	Delete	View All Records	Modify All Records	View All Fields		
Attendees	✓	✓							
Events	✓	✓	✓	✓	✓	✓			

	Basic Access				Data Administration				
	Read	Create	Edit	Delete	View All Records	Modify All Records	View All Fields		
Sponsors	✓	✓	✓						

Phase 3: Data Modeling and Relationships

Goal: Build the core data structure to manage events, sponsors, attendee and tickets.

1. Standard and Custom Objects

- Account: To store information.
- Contact: To store contact details.

Custom Object:

SETUP > OBJECT MANAGER

Attendee

Details

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Restriction Rules

Scoping Rules

Details

Description

API Name  
Attendee\_c\_\_c

Custom  
✓

Singular Label  
Attendee

Plural Label  
Attendees

Enable Reports  
✓

Track Activities  
✓

Track Field History

Deployment Status  
Deployed

Help Settings  
Standard salesforce.com Help Window

i. Attendee

SETUP > OBJECT MANAGER  
**Event**

**Details**  
Fields & Relationships  
Page Layouts  
Lightning Record Pages  
Buttons, Links, and Actions  
Compact Layouts  
Field Sets  
Object Limits  
Record Types  
Related Lookup Filters

**Details**  
Description  
API Name  
Event\_c\_\_c  
Custom  
✓  
Singular Label  
Event  
Plural Label  
Events  
Enable Reports  
✓  
Track Activities  
✓  
Track Field History  
Deployment Status  
Deployed  
Help Settings  
Standard salesforce.com Help Window

ii.

Event

SETUP > OBJECT MANAGER  
**Sponsor**

**Details**  
Fields & Relationships  
Page Layouts  
Lightning Record Pages  
Buttons, Links, and Actions  
Compact Layouts  
Field Sets  
Object Limits  
Record Types  
Related Lookup Filters  
Restriction Rules

**Details**  
Description  
API Name  
Sponsor\_c\_\_c  
Custom  
✓  
Singular Label  
Sponsor  
Plural Label  
Sponsors  
Enable Reports  
✓  
Track Activities  
✓  
Track Field History  
Deployment Status  
Deployed  
Help Settings  
Standard salesforce.com Help Window

iii.

Sponsor

SETUP > OBJECT MANAGER  
**Ticket**

**Details**  
Fields & Relationships  
Page Layouts  
Lightning Record Pages  
Buttons, Links, and Actions  
Compact Layouts  
Field Sets  
Object Limits  
Record Types  
Related Lookup Filters  
Restriction Rules

**Details**  
Description  
API Name  
Ticket\_c\_\_c  
Custom  
✓  
Singular Label  
Ticket  
Plural Label  
Tickets  
Enable Reports  
✓  
Track Activities  
✓  
Track Field History  
Deployment Status  
Deployed  
Help Settings  
Standard salesforce.com Help Window

iv.

Ticket

## 2. Fields

Event\_Status\_\_c

- Purpose: Tracks the current state of the event lifecycle.
- Values: *Planning, Registration Open, Active (In Progress), Completed, Canceled*. (This field is crucial for automation and reporting.)

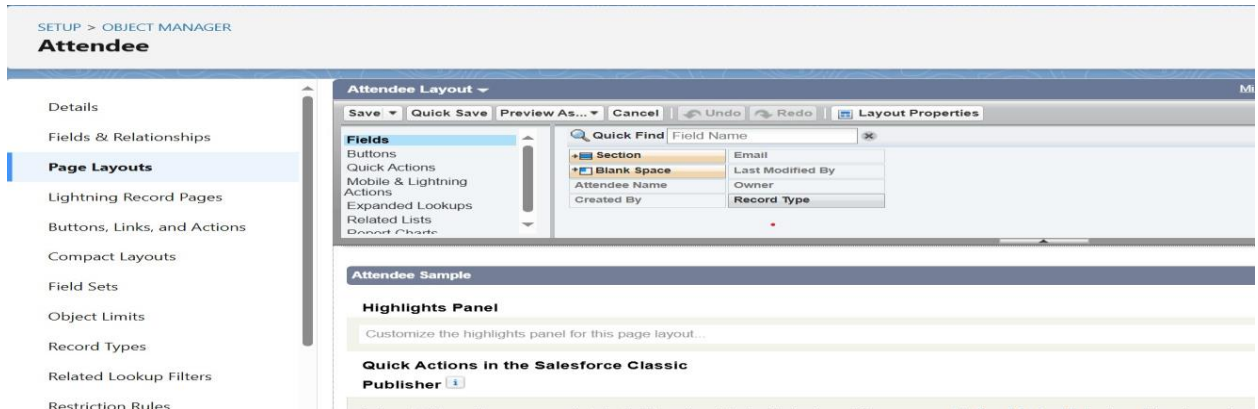
Ticket\_Type\_\_c

- Purpose: Tracks the specific access level purchased or assigned.
- Values: *General Admission, VIP, Speaker, Sponsor, Employee*. (Affects Page Layout visibility for related details, like seating assignment.)

### 3. Record Types

- Record Types allow you to offer different business processes values, and page layouts to different users based on their profile. They are often used on Objects like Lead, Opportunity, or Case, and custom objects.

### 4. Page Layout

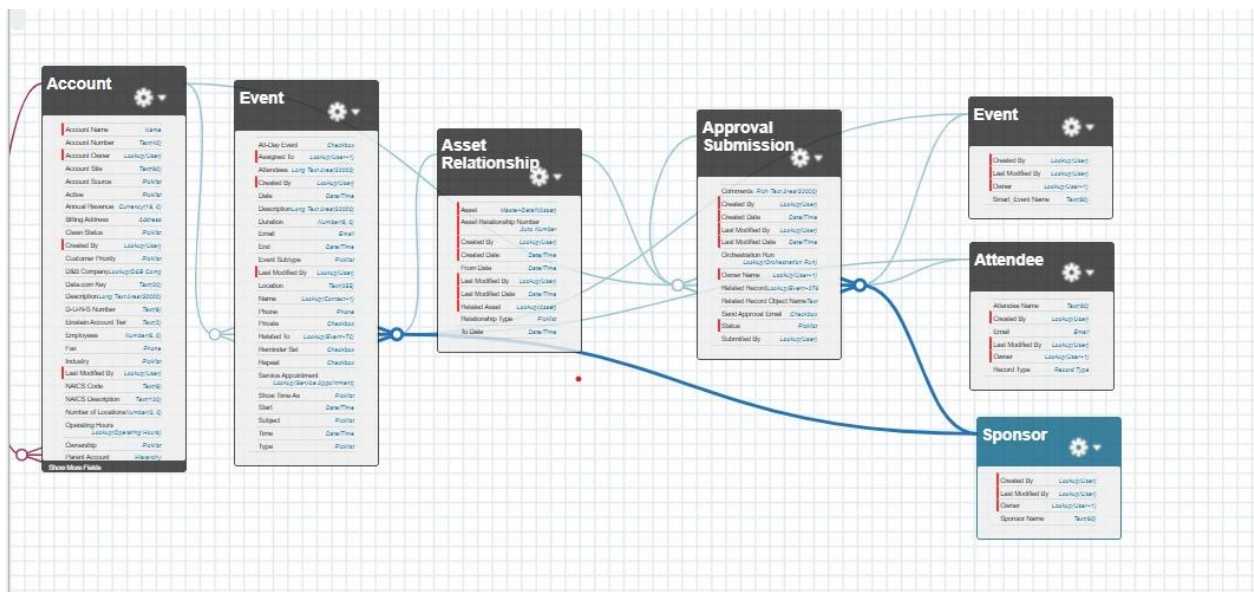


- Page Layouts are assigned to specific **Profiles** and **Record Types**.

### 5. Custom Layout

- The Project compact layout is configured to show the project name, event manager name, sponsors and tickets in list.

### 6. Schema builder





## Phase 4: Process Automation (Admin)

7. Validation Rules:

### Attendee Validation Rule

[Back to Attendee](#)

**Validation Rule Detail**

EditClone

Rule Name	Attendee_Email_Required	Active	<input checked="" type="checkbox"/>
Error Condition Formula	ISBLANK(Email__c)		
Error Message	Email is required for Attendees.	Error Location	Top of Page
Description			
Created By	Samhitha M.C, 9/24/2025, 6:19 AM	Modified By	Samhitha M.C, 9/24/2025, 6:19 AM

EditClone

- A validation rule has been created on the attendee. It prevents user from blank email Id.


8. Workflow Rules:

- This is a legacy automation tool. All new automations for this project are being built in Flow builder for better performance.

9. Process Builders:

- Used for all record triggered automations in this project.

10. Approval Process:

 **SETUP**  
**Approval Processes**

### Approval Processes

## Event: Event\_Approval

[« Back to Approval Process List](#)

**Process Definition Detail**

EditCloneDeleteActivate

Process Name	Event_Approval
Unique Name	Event_Approval
Description	
Entry Criteria	
Record Editability	Administrator <b>ONLY</b> AI
Approval Assignment Email Template	<u>Welcome_to_Smart_Event</u>
Initial Submitters	Event Owner
Created By	Samhitha M.C, 9/25/2025, 10:06 AM

- Approval process is required for this project as it requires more team members

11. Flow Builder

Flow Builder

Send Welcome Email - V1

Your automation was activated.

Select Elements

Auto-Layout

Last saved on 9/25/2025, 10:26 PM

Active

Run

Debug

View Tests

Save As New Version

Save

De

Record-Triggered Flow

Start

Object: **Attendee**

Trigger: **A record is created**

Optimize for: **Actions and Related Recor...**

+ Add Scheduled Paths (Optional)

Open Flow Trigger Explorer for Attend...

Run Immediately

Send Welcome Email

Action

End

Welcome Email to Attendee

\* Label

Send Welcome Email

\* API Name

Send\_Welcome\_Email

Description

Use values from earlier in the flow to set the inputs for the "Welcome Email to Attendee" core ac use its outputs later in the flow, store them in variables.

Set Input Values

A<sub>3</sub> \* Record ID

\$Record → Id

Show advanced options

- Record-Triggered Flow: Runs automatically when a attendee is created . This single flow is the cover of the project’s automation.

12. Email alerts

SETUP

Email Alerts

Email Alert

Welcome Email to Attendee

Rules Using This Email Alert [0] | Approval Processes Using This Email Alert [0]

Email Alert Detail

Edit

Delete

Clone

Description	Welcome Email to Attendee
Unique Name	Welcome_Email_to_Attendee
From Email Address	Current User's email address
Recipients	User: David Attendee
Additional Emails	
Created By	Samhitha M.C, 9/25/2025, 9:44 AM

Edit

Delete

Clone

SETUP

Classic Email Templates

Text Email Template

Welcome\_to\_Smart\_Event

Preview your email template below.

Email Template Detail

Edit

Delete

Clone

Email Templates from Salesforce	Unfiled Public Classic Email Templates
Email Template Name	Welcome_to_Smart_Event
Template Unique Name	Welcome_to_Smart_Event
Encoding	Unicode (UTF-8)
Author	Samhitha M.C [Change]
Description	
Created By	Samhitha M.C, 9/25/2025, 9:37 AM

Edit

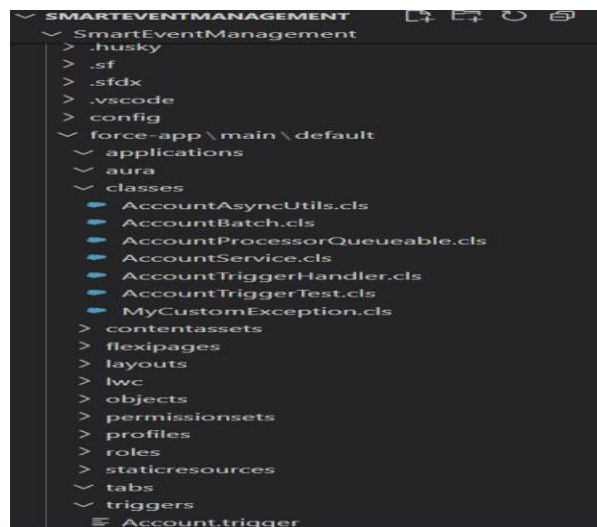
Delete

Clone

## Phase 5: Apex Programming (Developer)

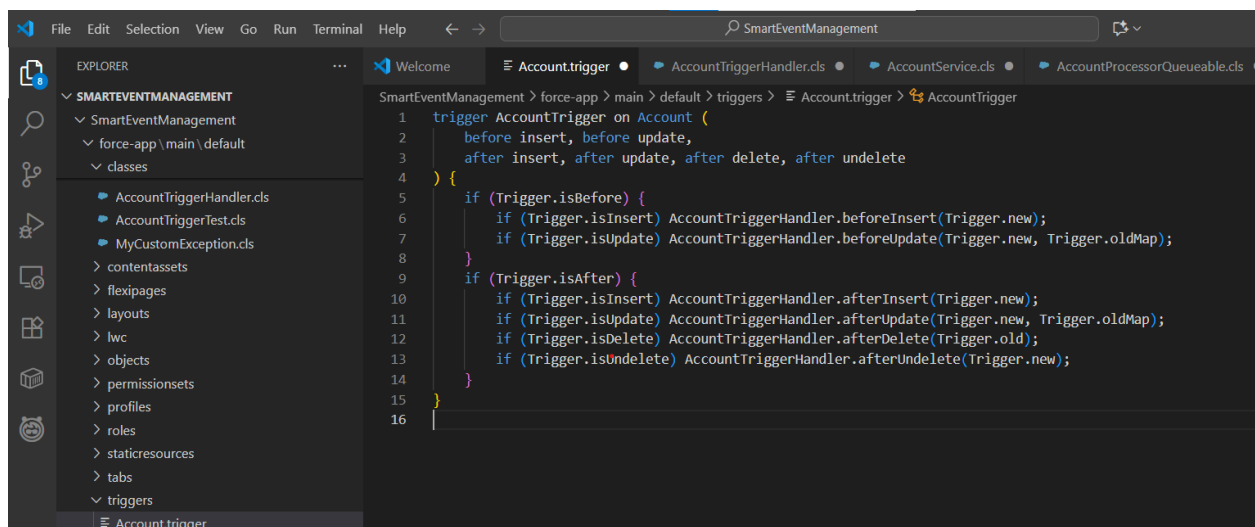
### 1. Classes and Objects

- A Class is a blueprint for an object. It defines variables (attributes) and methods (actions).
- AccountAsyncUtils.cls: Likely a utility class with helper methods, possibly for asynchronous operations.
- AccountBatch.cls: A Batch Apex class, a blueprint for a job object that processes large data sets.
- AccountProcessorQueueable.cls: A Queueable Apex class, a blueprint for an asynchronous job object.
- AccountService.cls: A service layer class that holds the main business logic for the Account object.
- AccountTriggerHandler.cls: The class that contains the bulkified logic for the Account trigger.
- AccountTriggerTest.cls: A test class (ends with Test.cls) used to verify the functionality of other Account-related classes.
- MyCustomException.cls: A blueprint for a custom error object that can be thrown and caught in your code.



In the context of Apex programming, an Object is an instance created from a class. These objects do not exist as files in your directory; they are created and used at runtime within the code

### 2. Apex Triggers (before/after insert/update/delete)



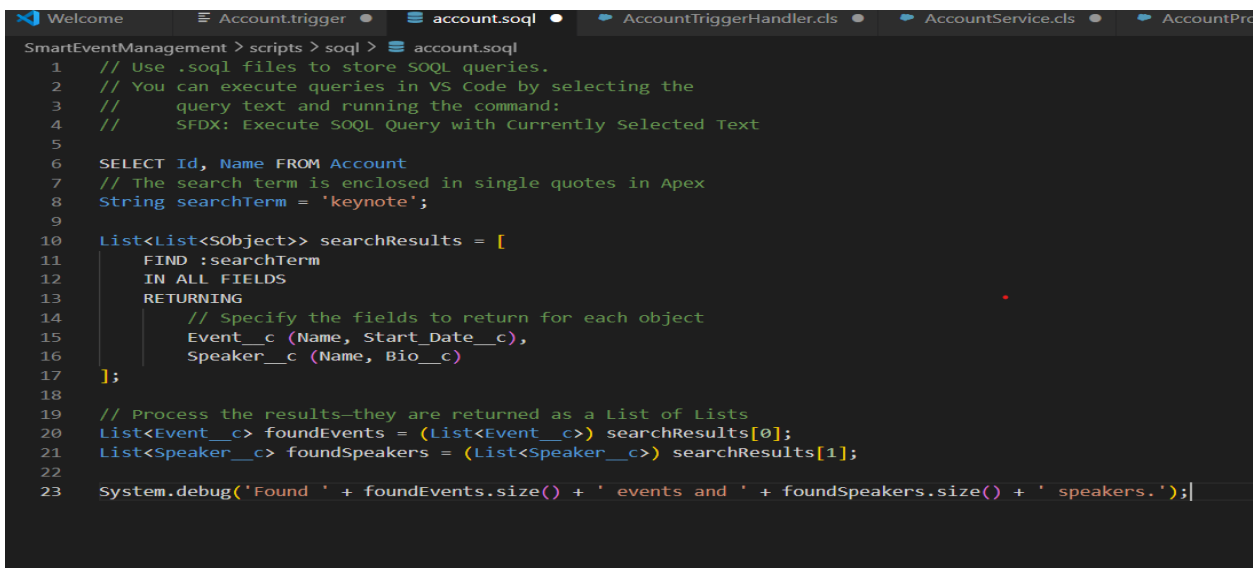
- Apex Triggers are code blocks that execute before or after a Data Manipulation Language (DML) event (insert, update, delete, undelete) occurs on a Salesforce object record.

### 3. Trigger Design Pattern

- A trigger framework or pattern is a way to structure your Apex code to make it scalable, maintainable

### 4. SOQL and SOSL

- SOQL (Salesforce Object Query Language): Used to retrieve records from a *single* object or multiple related objects based on criteria. It's similar to SELECT in SQL.
- SOSL (Salesforce Object Search Language): Used to search for text across *multiple* objects and fields simultaneously. It's better for text-based searches when you don't know the exact object.



```
1 // Use .soql files to store SOQL queries.
2 // You can execute queries in VS Code by selecting the
3 // query text and running the command:
4 // SFDX: Execute SOQL Query with Currently Selected Text
5
6 SELECT Id, Name FROM Account
7 // The search term is enclosed in single quotes in Apex
8 String searchTerm = 'keynote';
9
10 List<List<SObject>> searchResults = [
11     FIND :searchTerm
12     IN ALL FIELDS
13     RETURNING
14         // Specify the fields to return for each object
15         Event__c (Name, Start_Date__c),
16         Speaker__c (Name, Bio__c)
17 ];
18
19 // Process the results—they are returned as a List of Lists
20 List<Event__c> foundEvents = (List<Event__c>) searchResults[0];
21 List<Speaker__c> foundSpeakers = (List<Speaker__c>) searchResults[1];
22
23 System.debug('Found ' + foundEvents.size() + ' events and ' + foundSpeakers.size() + ' speakers.');
```

### 5. Collections: List, Set, Map

- Used a Set<String> in trigger handler to efficiently store keys.

### 6. Control Statement

- The trigger handler uses if statements to iterate and check the conditions.

### 7. Asynchronous Processing (Batch, Queueable, Future Methods)

- Scheduled Apex is a form of asynchronous processing. The use of Batch or Queueable Apex isn't explicitly mentioned as a required step, but they are often the next logical step if the logic within your scheduled job needs to process millions of records.

### 8. Exception Handling

- Using try, catch, and finally blocks to gracefully handle runtime errors (exceptions) without halting the entire transaction or showing a cryptic error message to the user.

## 9. Test Classes

```
@isTest
private class AccountTriggerTest {
    @isTest static void testBeforeInsertAndQueueableAndBatch() {
        // Create test data
        List<Account> accts = new List<Account>();
        for (Integer i=0; i<5; i++) {
            accts.add(new Account(Name = ' TestAcct ' + i + ' ')); // spaces to be trimmed
        }

        Test.startTest();
        insert accts; // triggers run, queueable enqueued
        // run batch within test context
        Database.executeBatch(new AccountBatch(), 100);
        Test.stopTest();

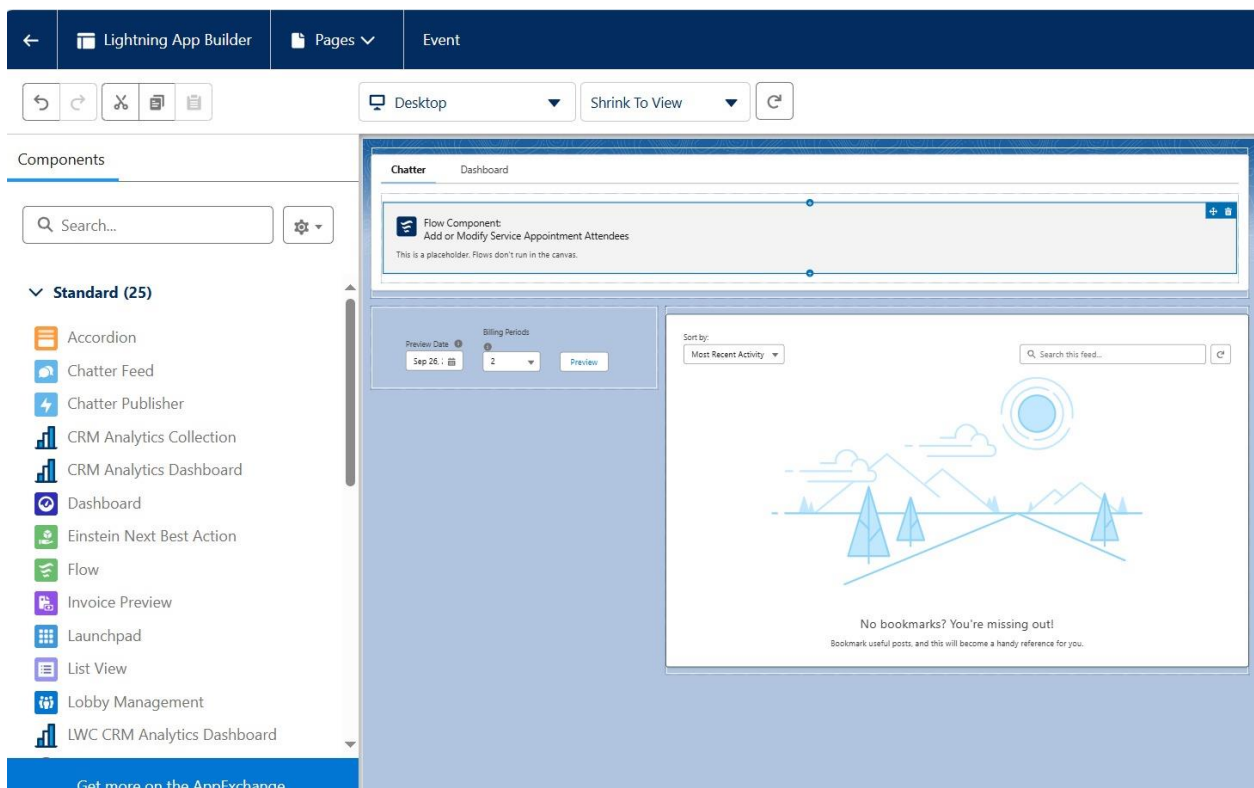
        // Collect inserted ids
        List<Id> ids = new List<Id>();
        for (Account a : accts) ids.add(a.Id);

        // Verify names were trimmed
        for (Account a : [SELECT Name, Description FROM Account WHERE Id IN :ids]) {
            System.assertEquals(a.Name, a.Name.trim(), 'Name should be trimmed by trigger/service');
            // Description should include 'processed' because queueable updated it during Test.stopTest()
            System.assert(a.Description != null && a.Description.contains('processed'), 'Queueable should have updated Description');
        }
    }
}
```

## Phase 6: User Interface Development

### 1 . Lightning App Builder

- Used the lightning app builder to create app named “Smart Event Management”



Lightning App Builder
App Settings
Pages
Smart\_Event\_Management
? Help

App Settings
Back

App Details & Branding
App Options
Utility Items (Desktop Only)
Navigation Items
User Profiles

App Details & Branding
Give your Lightning app a name and description. Upload an image and choose the highlight color for its navigation bar.

App Details
\* App Name
Smart\_Event\_Management
\* Developer Name
Smart\_Event\_Management
Description
To handle events .

App Branding
Image
Primary Color Hex Value
#0070D2
Org Theme Options
☒ Use the app's image and color instead of the org's custom theme
App Launcher Preview

Sm
Smart\_Event\_Management
To handle events .

## 2. Record Pages

- Event and Booking record pages can be customized to display important event details like participants, venue, and schedule.

## 3. Tabs

- Custom tabs can be created for modules like Events, Bookings, Sponsors to allow easy navigation inside the Smart Event Manager app.
- Tabs help segment functionalities so that users can quickly access required features like event reports or registration details.

## 4. Home Page Layouts

Home
Chatter
Campaigns
Leads
Accounts
Contacts
Opportunities
Forecasts
Contracts
Orders
Cases
Solutions
Products
Reports
Dashboards

Create New...

Samhitha M C
Friday September 26, 2025

Hide Feed

Post
File
New Event
More

Share an update, @mention someone...
Share

Sort By: Most Recent Activity

There are no updates.

Recent Items

Samhitha M C
Send Welcome Email
Send Welcome Email
Send Welcome Email
Sarah Sponsor
John Event Manager
David Attendee

Recycle Bin

Messages and Alerts

Custom Links

Recommendations
More

Download Salesforce
Run your business from your phone with Salesforce.
Skip >

David Attendee
Joined in the last month
Follow

Sarah Sponsor
Joined in the last month
Follow

John Event Manager
Joined in the last month
Follow

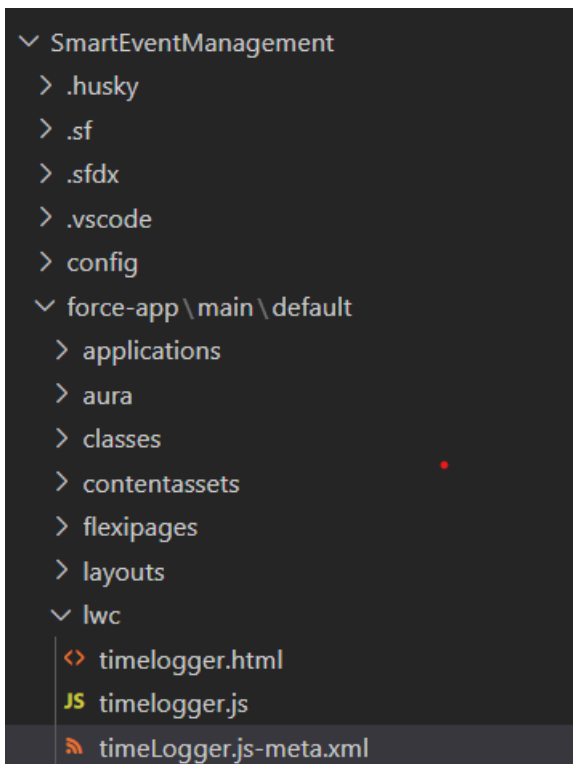
Integration User
Joined in the last month
Follow

## 5. Utility Bar

- The utility bar can include a Recent Items component so event managers can quickly access frequently used records like active events.

## 6. LWC

```
SmartEventManager > force-app > main > default > lwc > <> timellogger.html > ...
1  <template>
2    <lightning-card title="Log Billable Time">
3      <div class="slds-p-around_medium">
4        <lightning-input label="Hours" type="number" value={hours} onchange={handleHoursChange}></lightning-input>
5        <lightning-textarea label="Description" value={description} onchange={handleDescChange}></lightning-textarea>
6        <div class="slds-m-top_small">
7          <lightning-button label="Log Time" onclick={handleLogTime} variant="brand"></lightning-button>
8        </div>
9      </div>
10   </lightning-card>
11 </template>
12
```



- v. File directory which includes LWC file

- LWCs provide reusable UI blocks that ensure faster load times and responsive design for event operations.

## 7. Apex with LWC

- Apex classes can handle backend logic like saving registrations, calculating total revenue, or generating participant certificates.

## 8. Events in LWC

- Custom events can notify parent components, e.g., once an attendee has been successfully registered.

## 9. Imperative Apex calls

- Imperative calls can be used for on-demand actions like booking an event ticket when the “Register” button is clicked.
- They allow error handling and confirmation messages (e.g., “Registration Successful”) directly in the UI.

## 10 . Navigate service

- Navigating service is not used in this component , but a future enhancement could be to add a button that navigates service can redirect users to a list of all events, specific event records, or reports after completing an action.

# Phase 7: Integration and External Access

- For the **Smart Event Management Project**, we are not implementing Phase 7 at this stage because the current scope is focused on building core features. External integrations and advanced access configurations can be added in future phases when the application requires connections to outside systems.

### 1. Named Credentials

- Provide a secure way to store authentication details (like usernames, passwords, tokens) for calling external systems.
- Simplify integration by avoiding hardcoding credentials inside Apex code.

### 2. External Services

- Allow Salesforce to directly consume APIs from external systems using a declarative setup.
- Useful for connecting event management with third-party apps like payment gateways or ticketing platforms.

### 3. Web Services (REST/SOAP)

- Salesforce can expose its data as REST or SOAP services for other systems to consume.
- It can also consume external services for tasks like syncing event attendee data from partner systems.

### 4. Callouts

- Apex callouts allow Salesforce to send HTTP requests to external services (e.g., sending attendee info to an email service).
- Callouts are essential for real-time communication with APIs like payment processing or SMS notifications.

### 5. Platform Events

- Used for event-driven architectures, enabling apps inside and outside Salesforce to communicate asynchronously.
- For example, an “Event Registered” platform event could notify other systems about new registrations instantly.

### 6. Change Data Capture (CDC)

- Publishes real-time events whenever data changes in Salesforce (e.g., a new booking is created).
- Helps external systems stay synchronized without the need for constant API polling.

### 7. Salesforce Connect

- Allows Salesforce to access data stored in external databases without importing it into Salesforce.
- Ideal for cases where event data is maintained in another system but needs to be viewed in Salesforce

### 8. API Limits

- Understanding and optimizing API usage is crucial for integrations that involve high volumes of transactions.



#### 9. OAuth & Authentication

- OAuth provides a secure way for Salesforce to authenticate with external services.
- It ensures only authorized systems and users can access sensitive event-related data.

#### 10. Remote Site Settings

- Before making a callout, the external service URL must be whitelisted in Salesforce Remote Site Settings.
- This acts as a safeguard against unauthorized or accidental callouts to untrusted domains