

Event Management System

Project Report Submitted
In Partial Fulfillment of the Requirements
for The **Degree of**

MASTER OF COMPUTER APPLICATION

By

PRINCI GARG

University Roll no.: 1900290149072

Under the supervision of

Mr. Ankit Verma
(Assistant Professor)

KIET GROUP OF INSTITUTIONS, GHAZIABAD



to the

DEPARTMENT OF COMPUTER APPLICATIONS

**DR. A.P.J. ABDUL KALAM TECHNICAL UNIVERSITY
LUCKNOW**

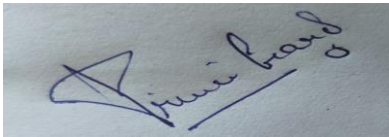
(Formerly Uttar Pradesh Technical University, Lucknow)

(July 2021)

DECLARATION

We hereby declare that the work presented in this report entitled **“EVENT MANAGEMENT SYSTEM”**, was carried out by US. We have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University or Institute. We have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, experiments, results, that are not my original contribution. We have used quotation marks to identify verbatim sentences and given credit to the original authors/sources.

We affirm that no portion of my work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, We shall be fully responsible and answerable.

A handwritten signature in blue ink, appearing to read 'Princi Garg', is shown on a light-colored background.

Name : Princi Garg

Roll. No. :1900290149072

Branch : Master of Computer Applications

INTERNSHIP CERTIFICATE



Cognizant GenC Internship - Early Graduation and Completion

1 message

<GenCProgram@cognizant.com>
To: princigarg7083@gmail.com

Tue, 20 Jul 2021 at 8:16 pm

Cognizant

Dear Princigarg,

Greetings from Cognizant !

Glad to let you know that we are graduating you early from the Internship program based on business requirements and prioritize your FTE onboarding process aligned to your performance in the Internship program.

Congratulations on completing your Internship at Cognizant during the period between
06-03-2021 and 15-06-2021.

Your overall performance during Internship has been Satisfactory.

We appreciate the passion and professionalism you've exhibited during the Internship. We take this opportunity to wish you all the best and looking forward to work with you soon.

Best Regards,
Human Resources – GenC Program

© 2021 Cognizant. All rights reserved.

This e-mail and any files transmitted with it are for the sole use of the intended recipient(s) and may contain confidential and privileged information. If you are not the intended recipient(s), please reply to the sender and destroy all copies of the original message. Any unauthorized review, use, disclosure, dissemination, forwarding, printing or copying of this email, and/or any action taken in reliance on the contents of this e-mail is strictly prohibited and may be unlawful. Where permitted by applicable law, this e-mail and other e-mail communications sent to and from Cognizant e-mail addresses may be monitored.

CERTIFICATE

Certified that **Princi Garg (1900290149072)** has carried out the project work presented in this project entitled “**EVENT MANAGEMENT SYSTEM**” for the award of **Master of Computer Application** from Dr. A.P.J Abdul Kalam Technical University, Lucknow under my supervision. The thesis embodies results of original work, and studies as are carried out by the student himself and the contents of the thesis do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Mr. Ankit Verma

Assistant Professor

Department of Computer Applications

KIET Group of Institutions, Ghaziabad

Signature of External Examiner

Signature of Internal Examiner

Dr. Ajay Kumar Shrivastava

Prof. & Head of Department KIET

Group of Institutions,

Ghaziabad

ACKNOWLEDGEMENT

Firstly, I thank **COGNIZANT.** for allowing me to complete my project successfully. I express my sincere gratitude to all those who initiated and helped me in the successful completion of this project. Sincere thanks and profound gratitude to my project Manager **Mr. Shibu** for their strict supervision, constant encouragement, inspiration and guidance, which ensure the worthiness of my work. Working under them was an enrich experience and my project Internal Supervisor **Ms. Sangeeta Arora** for helping in carrying out the project work and for many valuable and useful information and guidance while bringing out this project.

I am highly grateful to **Dr. Ajay Kumar Shrivastava, HOD** of **MCA KIET** Group of Institutions, Ghaziabad for providing this opportunity to carry out the Major Project at “**EVENT MANAGEMENT SYSTEM**”. **KIET** Group Of Institutions Ghaziabad for providing academic inputs, guidance & Encouragement throughout this period.

Finally, I express my indebtedness to all who have directly or indirectly contributed to the successful completion of my major project.

PRINCI GARG
(1900290149072)

Table of Content

Chapter 1 :Introduction

1.1 Overview	1
1.2 Motivation	2
1.3 Problem Definition.....	2
1.4 Objective.....	2

Chapter 2 : Literature Survey

2.1 Work Done In Related Field	3
2.2 Different Features of the Salesforce Platform.....	3

Chapter 3 : Workflow And Software Description

3.1 Workflow.....	6
3.2 Software Specifications.....	7
3.2.1 Programming / Scripting Languages.....	7
3.2.2 Lightning Component.....	7
3.2.5 SOQL.....	11
3.2.6 SOSL	12
3.2.7 Trigger.....	13
3.2.8 Salesforce Object.....	14
3.2.9 Tabs	15
3.2.10 Relationship.....	15

Chapter 4 : Why Should I Manage Events In Salesforce?

4.1 Type of Events	17
4.1.1 Multi-track.....	17
4.1.2 Conferences	18
4.1.3 RSVP Events	18
4.1.4 Training Course.....	19
4.1.5 Fundraising Events	19

Chapter 5 : Overview of Project

5.1 Entities Involved.....	21
5.1.1 Location.....	21
5.1.2 Event Organizer.....	21
5.1.3 Event.....	22
5.1.4 Attendees	22
5.1.5 Speaker	22
5.1.6 Event-Attendee.....	22
5.1.7 Event-Speaker	22

Chapter 6 : Project Description

Chapter 7 : Literature Review

CHAPTER-1

INTRODUCTION

1.1 OVERVIEW

In the past, storing the information was always a challenge for every organization. There is a lot of data that is being stored and processed by every organization. Every thing used to manual in those days but arrival of computers has changed everything. Organizations started storing data in their own local database or local servers. But organizations found it difficult to maintain the local database. The local databases were also not readily available from everywhere. Now cloud computing technologies are introduced. Cloud computing is the delivery of different services through the Internet. These resources include tools and applications like data storage, servers, databases, networking, and software. There are many companies that offer cloud computing. Salesforce is one of the companies that provides cloud computing tools. It majorly provides customer relationship management (CRM) service and also provides enterprise applications focused on customer service, marketing automation, analytics, and application development. Customer Relationship management (CRM) is a term that refers to practices, strategies and technologies that companies use to manage and analyze customer interactions and data throughout the customer lifecycle, with the goal of improving business relationships with customers, assisting in customer retention and driving sales growth. CRM systems are designed to analyze the customers data using different tools. This will in turn help the company to maintain grow the products based on customers interests. Salesforce provides services like platform as a service (PAAS) and software as a service (SAAS). So, we can use this platform for both developing a software and hosting a software. In this project we have developed an application which can manage event particularly for time being.

1.2 Motivation

This application deals with a lot of data and automation processes. So, we have decided to use cloud platform. Since Salesforce is one of the best cloud-based platforms in the world and provides a lot of powerful tools for analysing the data and provide automation tools.

1.3 Problem Definition

This application is for a company that is dealing with event. In today's era, all the events are organized live or through a virtual mode. Managing all the events in specific time, particular time is very vital. No conflict should be there.

1.4 Objective

The steps involved in making this project are:

- Design the database by creating objects (tables), fields, relationships between the objects.
- Create different Lightning components for user interface of the applications.
- Create Apex classes which act as an interface between UI components and Salesforce cloud.
- Create events that help in communication between the UI components.

CHAPTER-2

LITERATURE SURVEY

2.1 Work Done in Related Field:

Customer Relationship Management (CRM) has become one of the most dynamic technology topics of the millennium. According to Chen and Popovich (2003), CRM is not a concept that is really new but rather due to current development and advances in information and enterprise software technology, it has assumed practical importance. The root of CRM is relationship marketing, which has the objective of improving the long-term profitability of customers by moving away from product-centric marketing. Customer Relationship Management (CRM) has become one of the most dynamic technology topics of the millennium. According to Chen and Popovich (2003), CRM is not a concept that is really new but rather due to current development and advances in information and enterprise software technology, it has assumed practical importance. The root of CRM is relationship marketing, which has the objective of improving the long-term profitability of customers by moving away from product-centric marketing. That is when salesforce was introduced.

Salesforce started as a cloud-based solution for CRM. It involves managing all aspects of relationship between an organization and its customers. For example, the contact details of the customer, the deals that are in progress or already completed, the support requests from a customer or a new lead from a new customer. Beyond the customer related information, it also involves storing and managing the details of the people and the concerned department from the seller organization that is managing the customer's account and needs. This makes it easy to manage and enhance the relationship with the customer and hence better growth for the organization.

2.2 Different features of the Salesforce platform :

- **Contact Management:**

To view customer contact details, activity history, customer communications, and internal account discussions, etc. In short, it manages all the data pertaining to the contact with a customer.

- **Opportunity Management:**

It provides the details of the stage a deal is in, the products involved in the deal, the quotation for the deal etc. In short it manages all the data that helps in identifying, progressing and closing a deal.

- **Salesforce Engage:**

This feature is focused on making personalized contact with a customer for various campaigns designed by the marketing team. It also provides real time sales alerts based on the level of engagement with a customer.

- **Sales Collaboration:**

This feature helps in quickly finding experts who can help in closing a deal based on customer queries and feedback. In short, it helps in bringing in a collaborative effort to engage an entire team in the deal and make the deal happen.

- **Sales Performance Management:**

It provides a metric-based goal setting, and also continuous feedback and rewards and recognition for the sales team. This helps in enhancing the performance of the sales team.

- **Lead Management:**

This feature initiates and tracks the leads that are in progress. It also helps in continually optimizing campaigns across every channel.

- **Salesforce Mobile App:**

This is the mobile platform to carry out all the above activities on a mobile platform.

- **Workflow and Approvals:**

It is a visual design to automate the business processes. The interface provides simple drag and drop options to make this design. It helps in creating a flexible approval process with deal discounts and expense management etc.

- **Email Integration:**

Salesforce can integrate to an existing email platform. This helps in providing flexibility to the existing team with no additional learning curve.

- **Files Sync and Share:**

This feature provides the sales team the power to easily share various files, discuss them and update them as needed. Also receive alerts when something in the file changes.

- **Reports and Dashboards:**

Dashboards offer a real-time picture of the business at a glance. With this, anyone can create detailed reports which can be accessed from anywhere.

- **Sales Forecasting:**

This feature helps in getting a real time view of the forecast of a sales team. It provides multi-currency support and an in-line editing mode to manage the sales forecast well.

- **Developers Platform:**

It also provides developers platform to make apps with our own interface and objects.

CHAPTER – 3

WORK FLOW AND SOFTWARE DESCRIPTION

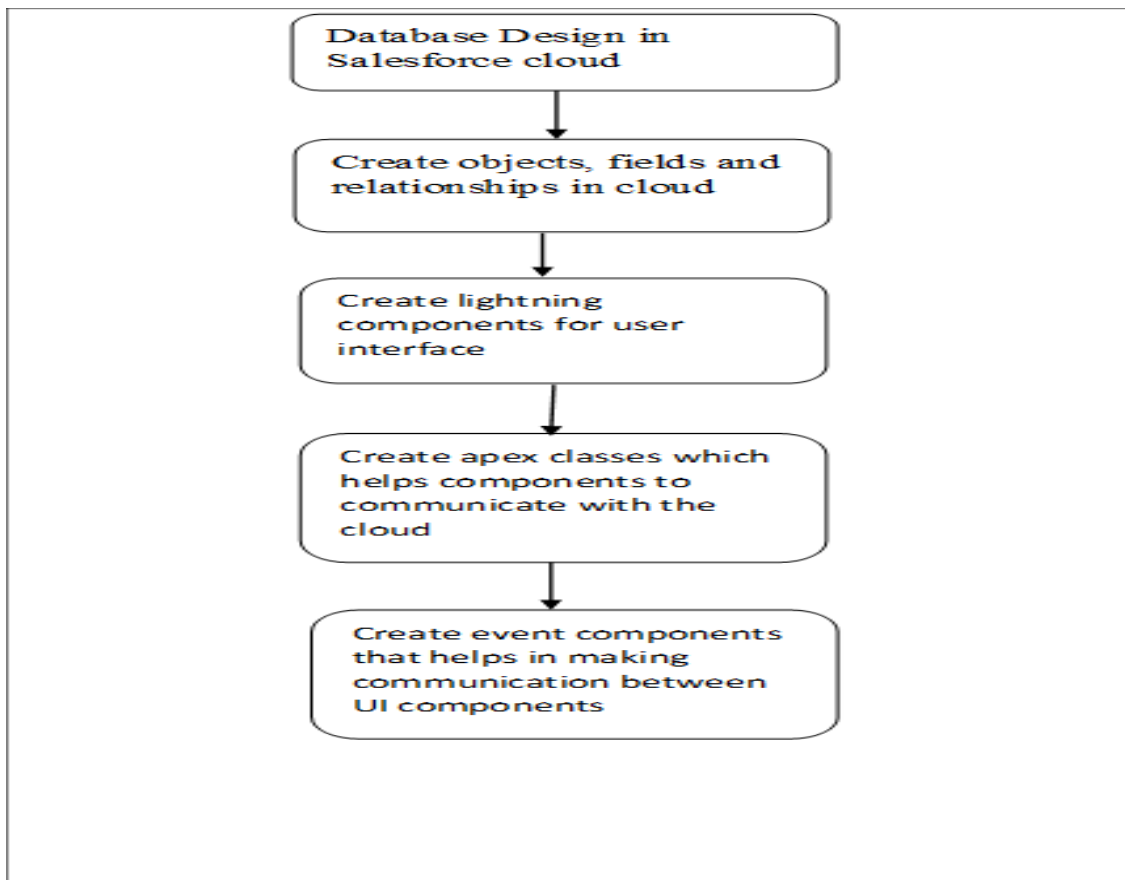


Fig: 3.1 Work flow

Firstly, we designed our database, tells the way in which data should be stored in the cloud visually. There is a tool called schema builder in salesforce which will help to design our database. Then we created object, fields for storing the data in the form of the tables (here tables are called objects). We will have to establish relationship between the objects as well in this step then we will develop UI of the application using lightning components. These components come as bundle (HTML,

CSS, JS). Then we developed apex classes which acts as an interface between the UI components and salesforce cloud for retrieval or manipulation of data in the cloud. Then there are events that will help UI components to communicate at last we have integrated and tested the components as an application.

3.2 Software Specifications:

3.2.1 Programming/Scripting Languages:

- **HTML:** HTML is a markup language which is used to design web pages.
- **CSS:** CSS is a style sheet language used in describing the presentation of a document written in a markup language such as HTML.
- **JavaScript:** JavaScript is a scripting language used to make web pages interactive.

3.2.2 Lightning Components:

Lightning components are the self-contained and reusable units of an app. They represent a reusable section of the UI, and can range in granularity from a single line of text to an entire app.

The framework includes a set of prebuilt components. For example, components that come with the Lightning Design System styling are available in the lightning namespace. These components are also known as the base Lightning components. You can assemble and configure components to form new components in an app. Components are rendered to produce HTML DOM elements within the browser.

A component can contain other components, as well as HTML, CSS, JavaScript, or any other Web-enabled code. This enables you to build apps with sophisticated UIs.

The lightning components comes as a bundle. The bundle includes:

Resource	Resource Name	Usage
Component or Application	sample.cmp or sample.app	The only required resource in a bundle. Contains markup for the component or app. Each bundle contains only one component or app resource.
CSS Styles	sample.css	Contains styles for the component.
Controller	sampleController.js	Contains client-side controller methods to handle events in the component.
Design	sample.design	File required for components used in Lightning App Builder, Lightning pages, Experience Builder, or Flow Builder.
Documentation	sample.auradoc	A description, sample code, and

		<p>one or multiple references to</p> <p>example components</p>
Renderer	sampleRenderer.js	Client-side renderer to override default rendering for a component.
Helper	sampleHelper.js	JavaScript functions that can be called from any JavaScript code in a component's bundle
SVG File	sample.svg	Custom icon resource for components used in the Lightning App Builder or Experience Builder.

Table 3.1 Lightning Components Bundle

3.2.3 Lightning Events

Event-driven programming is used in many languages and frameworks, such as JavaScript and Java Swing. The idea is that you write handlers that respond to interface events as they occur.

A component registers that it may fire an event in its markup. Events are fired from JavaScript controller actions that are typically triggered by a user interacting with the user interface.

There are two types of events in the framework:

- **Component events** are handled by the component itself or a component that instantiates or contains the component.
- **Application events** are handled by all components that are listening to the event. These events are essentially a traditional publish-subscribe model.

You write the handlers in JavaScript controller actions.

3.2.4 Apex Classes

Apex is like Java for Salesforce. It enables you to add and interact with data in the Lightning Platform persistence layer. It uses classes, data types, variables, and if-else statements. You can make it execute based on a condition, or have a block of code execute repeatedly.

Data Types and Variables:

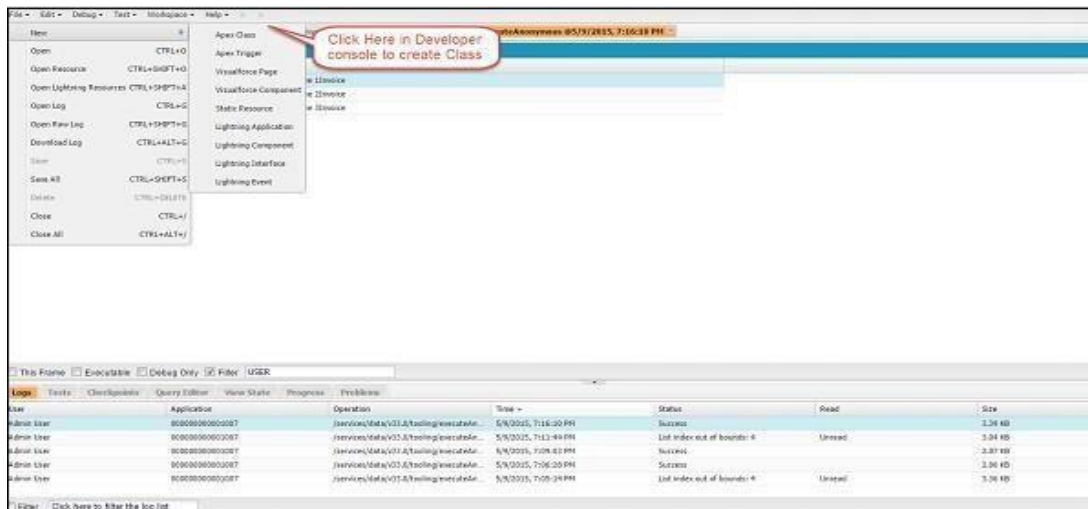
Apex uses data types, variables, and related language constructs such as enums, constants, expressions, operators, and assignment statements.

Control Flow Statements:

Apex provides if-else statements, switch statements, and loops to control the flow of code execution. Statements are generally executed line by line; in the order they appear. With control flow statements, you can make Apex code execute based on a certain condition, or have a block of code execute repeatedly.

Working with Data in Apex:

You can add and interact with data in the Lightning Platform persistence layer. The sObject data type is the main data type that holds data objects. You'll use Data Manipulation Language (DML) to work with data, and use query languages to retrieve data, such as the (), among other things.



3.2.5 Salesforce Object Query Language (SOQL)

Use the Salesforce Object Query Language (SOQL) to search your organization's Salesforce data for specific information. SOQL is similar to the SELECT statement in the widely used Structured Query Language (SQL) but is designed specifically for Salesforce data.

Similar to the SELECT command in Structured Query Language (SQL), SOQL allows you to specify the source object (such as Account), a list of fields to retrieve, and conditions for selecting rows in the source object.

Use SOQL when you know which objects the data resides in, and you want to:

- Retrieve data from a single object or from multiple objects that are related to one another.
- Count the number of records that meet specified criteria.
- Sort results as part of the query.
- Retrieve data from number, date, or checkbox fields.

The screenshot shows the Salesforce Query Editor interface. At the top, a blue header bar displays 'Query Results - Total Rows: 3'. Below this is a table with three rows of data: 'Avi Green', 'Ashley James', and 'Lauren Boyle'. To the right of the table, a callout box labeled 'Step 4: Results show here, including the number of rows' points to the table. Below the table is a 'Query Grid' with buttons: 'Save Rows', 'Insert Row', 'Delete Row', 'Refresh Grid', 'Access in Salesforce: Create New', and 'Open'. Below the grid is a tabbed interface with 'Logs', 'Tests', 'Checkpoints', 'Query Editor' (selected), 'View State', 'Progress', and 'Problems'. The 'Query Editor' tab shows a SQL query: 'SELECT name FROM contact WHERE recipe__c IN (select ID from recipe__c where name = 'pavlova')'. A callout box labeled 'Step 2: Type your query syntax here' points to the query text. Below the query is a yellow box with the text 'Any query errors will appear here...'. At the bottom left, there is an 'Execute' button and a checkbox for 'Use Tooling API'. A callout box labeled 'Step 3: Click this to execute the SOQL query' points to the 'Execute' button. On the right side, there is a 'History' panel with a list of executed queries.

Step 1: Click this tab to pull up the Query Editor

Step 2: Type your query syntax here

Step 3: Click this to execute the SOQL query

Step 4: Results show here, including the number of rows

3.2.6 Salesforce Object Search Language (SOSL):

Use the Salesforce Object Search Language (SOSL) to construct text-based search queries against the search index.

When building efficient SOSL queries, create filters that are selective. By default, SOSL queries scan all entities. The search engine looks for matches to the search term across a maximum of 2,000 records. Sharing is applied after the result set is returned from the search stack. If your filters are not selective and cause search term matches of more than 2000 records, there is a possibility of running into search crowding.

Use SOSL when you don't know which object or field the data resides in, and you want to:

- Retrieve data for a specific term that you know exists within a field. Because SOSL can tokenize multiple terms within a field and build a search index from this, SOSL searches are faster and can return more relevant results.

- Retrieve multiple objects and fields efficiently where the objects might or might not be related to one another.
- Retrieve data for a particular division in an organization using the divisions feature

FIND {Crisis} IN ALL FIELDS RETURNING Contact(FirstName, LastName, Phone, Email, Title)

Search Results - Total Rows: 3

Contact (3)

FirstName	LastName	Phone	Email	Title
Palma	Sunrise	(554)623-1212	psunrise@trailhead.c...	Control Engineer - Sp...
Vega	North	(416)556-1312	vnorth@trailhead.com	Control Engineer - Sp...
Quentin	Foam	(415)555-1212	qfoam@trailhead.com	Control Engineer - Sp...

Access in Salesforce: [Create New](#) [Open Detail Page](#) [Edit Page](#)

Logs Tests Checkpoints **Query Editor** View State Progress Problems

FIND {Crisis} IN ALL FIELDS RETURNING Contact(FirstName,LastName,Phone,Email,Title)

Any query errors will appear here...

Execute Query Plan ☐ Use Tooling API

History

- Executed
- SELECT FIRSTNAME FR...
- SELECT DEPARTMENT F...
- SELECT NAME, PHONE, ...

3.2.7 Trigger

A trigger is the piece of code that executed before and after a record is Inserted/Updated/Deleted from the force.com database. Apex can be invoked through the use of triggers. A Trigger is a functional action which gets on particular events. Triggers will happen before records entering into the database and while going out of the database. Each event is the firing point.

- Onclick
- Submit.

A Trigger is Apex code that execute before or after the following types of operations.

- Insert
- Update
- Delete
- Undelete

We can have a trigger run before an Object's records are inserted into the database, after records have been deleted, or even after a record is restored from the Recycle BIN.

3.2.8 Salesforce Object

Objects are database tables that allow us to store data specific to the organization in salesforce.

Two type of objects in Salesforce. They are.

1. Standard Objects: The objects provided by salesforce.com is called standard objects.

Examples of standard objects are accounts, contacts, opportunities, Leads, products, campaigns, cases, users, contracts, Report, and dashboards, etc.

2. Custom Object: The objects created by us are called custom objects. Custom objects store information that is unique and important to your organization. Custom objects are the heart of any application. Custom objects provide a structure for sharing data.

Custom objects have properties such as

- Custom fields.
- Relationship to other objects.
- Page Layouts.
- A custom user interface tab

An object is a home for your data. Think of an object as a form you'd use to collect and store crucial info on a specific topic that is relevant and actionable for your business. Standard objects are **a set of objects** that come already set up for you in Sales Cloud.

Salesforce objects are database tables that permit you to store the data specific to an organization. It **supplies information that is unique and essential to their organization**. It is the heart of any application. It provides a structure for sharing data.



3.2.9 Tabs :

A tab is a User interface to create records for an object and to view the records in objects.

In salesforce we three types of tabs

1. Custom object Tab
2. Web Tab
3. Visualforce Tab

The path to creating Custom tabs.

Setup -> Build -> Create -> Tab -> click on New tab and enter the details to complete the tab creation process.

3.2.10 Relationship :

An object relationship in Salesforce is **a two-way association between two objects.**

Relationships are created by creating custom relationship fields on an object. This is done so that when users view records, they can also see and access related dataThere are several different types of relationships in Salesforce.

3.2.11 OWD :

Organization Wide Defaults(OWD) in salesforce is **the baseline level of access that the most restricted user should have**. Organizational Wide Defaults are used to restrict access. You grant access through other means like(sharing rules, Role Hierarchy, Sales Teams and Account teams, manual sharing, Apex Sharing).

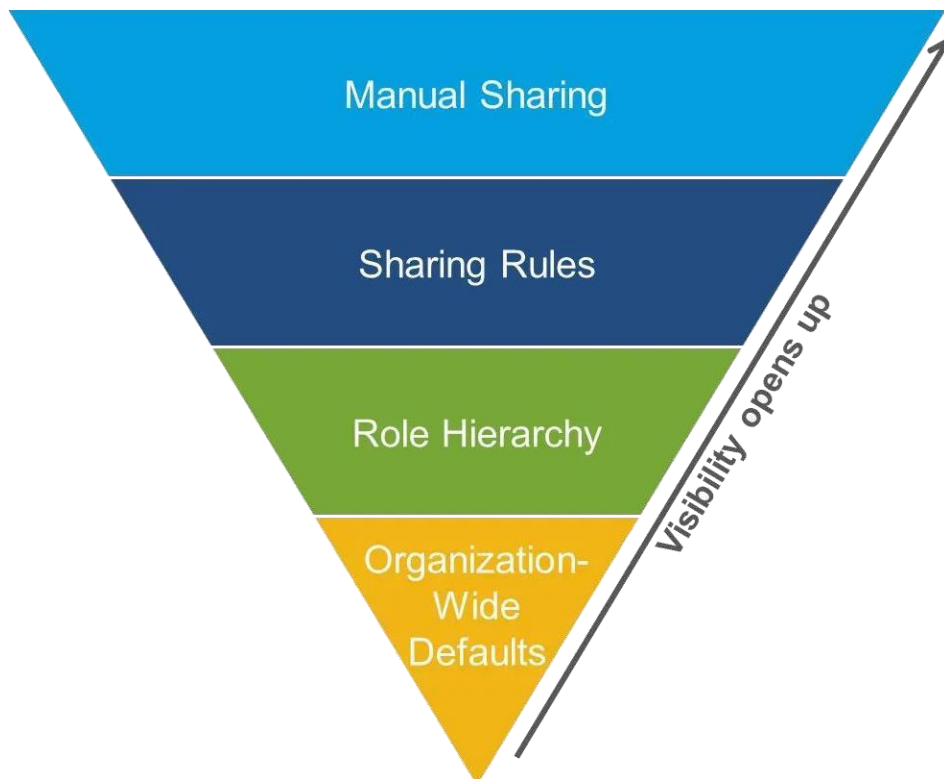


Figure : OWD in Salesforce

Object permissions determine the baseline level of access for all the records in an object. Org-wide defaults modify those permissions for records a users doesn't own. Org-wide sharing settings can be set separately for each type of object.

CHAPTER-4

Why Should I Manage Events in Salesforce?

An event management system within Salesforce allows you to access all relevant data in one centralized location—your database!

You'll be able to manage all event-related tasks without leaving Salesforce, providing you with a seamless and comprehensive event-planning experience.

4.1 TYPES OF EVENTS:

4.1.1 Multi-Track or Multi-Session Events:

If your event offers multiple tracks or sessions, you can use your Salesforce event management app to ensure that guests can easily register for exactly what they want—nothing more, and certainly nothing less!

For example, if you're a university offering multi-session courses, your app can enable students to register for all events within the series at one time.



Figure : Multi-Track or Multi-Session Events

4.1.2 Conferences :

Organizations like professional associations often plan and host conferences for their members to come together and have enriching experiences. Use your app to schedule speakers, vendors, and exhibitors for your conference. Then, allow your guests to manage their schedules depending on which tracks or sessions appeal to them.



Figure : Conferences

4.1.3 RSVP Events :

If your event requires specific registration and has a set guest list, your event management software can easily automate the process.

You can use your Salesforce CRM and event tools to segment your database and design the most strategic guest list for your event.

Send out invitations to specific individuals or create a designated list and then file your received RSVPs within your database. With a Salesforce app, you can customize your invitation process without the risk of losing data!



Figure : RSVP Events

4.1.4 Training Courses :

Training courses are often utilized to help constituents increase their knowledge. Employers might offer training courses to new employees or a nonprofit may offer them for their volunteers.

Whether you're offering training courses to your students, staff, or supporters, you can easily manage registrations within your database. Guests can choose to sign up for a one-time session, certain events within a series, or an entire sequence at once.



Figure : Training Courses

4.1.5 Fundraising Events :

Fundraising events are an engaging and profitable way to raise money for nonprofits, universities, churches, and other organizations.

When you plan your fundraisers within your Salesforce CRM, you'll automatically have access to all the data you need to plan an event that appeals to supporters

CHAPTER-5

OVERVIEW OF PROJECT

The project that we are going to implement is an Event Management project and we will be using the salesforce internal and community as well. This event management project can be for any company or event however we are managing the event for a fitness company called MAXFIT.

This project includes the following topics

- Salesforce Admin
 - Objects & Relationships
 - Sharing Rules, OWD & Roles
- Community
- Salesforce Development
 - Apex Trigger
 - Apex Batch Apex
 - Future Method
 - Integration (Both Apex REST & REST API)
- Lightning Web Component - This Covers most of the lightning web component including
 - Events
 - Custom Lookup
 - Calling Apex in bot wire & imperative apex
 - Using Web Component in Community
 - Navigation & Toast Events
- Reusable Error Handling Framework
- Salesforce Community - For the project, we are using Customer Community

MAXFIT:



Event Management (MAX FIT) MAX FIT is looking for an event management software which will help them to manage their events very effectively along with the attends & location information. As a developer you have been asked to develop the solution using Salesforce Platform. Here are the complete information of the project

5.1 Entities(Objects) Involved :

5.1.1 Location (Address Book)

1. Street (Text)
2. City (Text)
3. State (Text)
4. Postal Code (Text)
5. Country (PickList)
6. Landmark (Text)
7. Verified (checkbox)

5.1.2 Event Organizer

1. Name (Standard field)
2. Email (Email)
3. Alternative Email (Email)
4. Phone (Phone)
5. Alternative Phone (Phone)
6. Address (Lookup - Location)

5.1.3 Event

1. Event # (Standard Auto Number)
2. Name (Text)
3. Status (Created, Published, In Progress, Completed, Post Poned, Cancelled) - Picklist
4. Organizer (Lookup - Event Organizer)
5. Start Date/Time
6. End Date/Time Max Seats (Number)
7. # People Attending (Rollup Summary Field)
8. Remaining Seats (Formula field - “Max Seats - # People Attending“)
9. Lookup (Location)
10. Location Verified (Formula Location r.Verified c)
11. Live? (Checkbox)
12. Recurring? (Checkbox)
13. Event Type (In-Person, Virtual) - Picklist
14. Frequency (Daily, Weekly) - Picklist

5.1.4 Attendees

1. Name (Standard)
2. Email (Email)
3. Phone (Phone)
4. Company Name (Text)
5. Address (Lookup - Location)

5.1.5 Speaker

1. Name (Standard Name Field)
2. Email (Email)
3. Phone (Phone)
4. Company (Text)

5.1.5 Event-Attendee

1. Event (M-D Event)
2. Attendee (M-D Attendee)

5.1.6 Event-Speaker

1. Event (M-D Event)
2. Speaker(M-D Speaker)

5.1.7 Error Log

1. Log Date/Tile (Store the Date & Time details when the Error Occurred)
2. Log Details (Stores the Complete Stack Track about the Problem) - Text Area Long
3. Process Name (The Name of Either Apex Class or Apex Trigger) - Text
4. Log # (Auto Generated Number)

During the discussion between the business & client, total 15 milestone has been decided which as follows

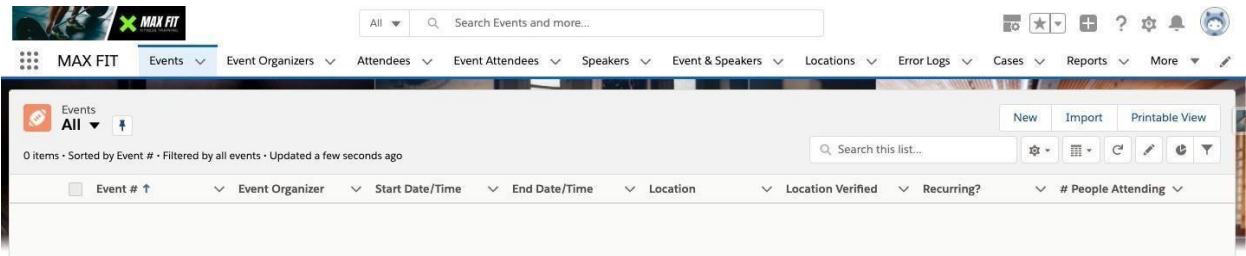
Before we start - Please do not forgot to watch the below playlist by SFDC Panther for salesforce development

1. [Back To Basic](#)
2. [LWC](#)

5.2 Milestone 1 (Object Setup)

#1 Complete the Object Structure, Choose the correct Data Type for each field.

#2 Create a Lightning Application & name it “**MAX FIT**” Use a logo of your choice.

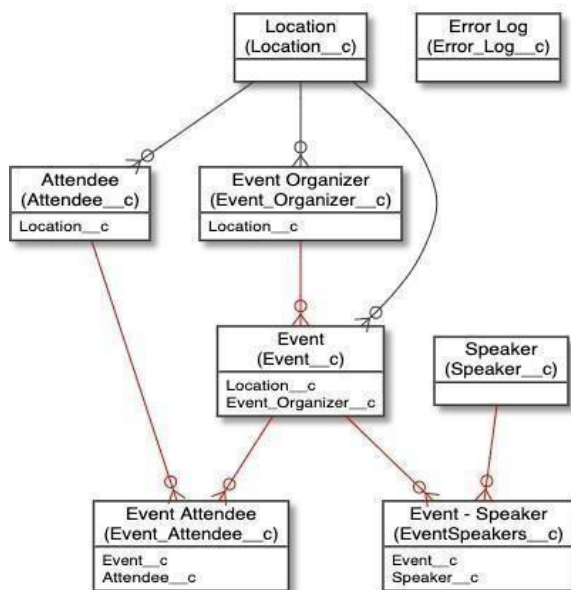


Must Have - Each Field must have a valid description & help text After Creating the Object/Fields & Type your app will look like below

You can get all the object details and can use Lightning Environment to create the object in a very fast way. Note to remember is use the Correct Datatype for each field.

Here is the Entity Relationship Diagram

Salesforce Entity-Relationship Diagram



Milestone 1.1-

Validation Rule Setup

1 - Validation Rule on Event Object

1.1- If Recurring? checkbox is checked then user must need to fill Frequency field & If checkbox is unchecked then User can not select Frequency field.

1.2- If Virtual is Selected as Value for Event Type field then Prevent User to Select Location on Event Record.

1.3- End Date/Time must be at-least 1 day ahead of Start Date/Time (If there is a value in End Date/Time field)

1.4- If Event Type field value is In-Person then user must need to select Location on Event Record.

2 - Event Attendee Object

Attendee can only be associated with the Event whose End Date is in future & Event Live Checkbox is checked and Event is accepting the Attendees (means Remaining Seats field value is not 0)

3 - Event Speaker Object

Speaker can only be associated with the Event whose End Date is in future & Event Live Checkbox is checked

Milestone 1.2 -

Duplicate Rule Setup

1 Speaker Object

User can not create duplicate speaker record with the same Email & Phone

2 Attendee Object

User can not Create duplicate attendee with the Same Name, Email & Phone

3 Event Organizer Object

Apply the same rule as Speaker object for not to having duplicate Organizer in the System

Milestone 1.3

Profile, User, OWD & Role Setup

Profile Setup -

Create 3 Profiles (Clone from Standard User), the name of the profile will be Event Organizer, Event Attendee & Speaker.

User Setup -

Create users for the testing purpose

Role Setup -

1. Create Roles (Organizer & Attendee & Speaker). All roles must report to CEO
2. Role Hierarchy

- CEO
- Organizer
 - Attendee
 - Speaker

3. OWD Setup -

Check the table for OWD below and make the changes

Object Permission Set-up :- Please provide the objects & fields level permission at the Profile level as per the table below.

Note:

C : Created

R : Read

E : Edit

D : Delete

X : No Access

Organization Wide Default

Object Name	<i>Organization Wide Default</i>
<i>Event</i>	Public Read Only
<i>Event - Organizer</i>	Public Read Only
<i>Speaker</i>	Private & Create a Sharing Rule to share the Speakers with Organizers (Role)
<i>Attendee</i>	Private & Create a Sharing Rule to share the Attendee with Organizers (Role)
<i>Location</i>	Public Read Only
<i>Event - Speaker</i>	Public Read Only
<i>Event - Attendees</i>	Public Read Only

Sharing Rule Setup:- As per the business requirement, we need to share every Speaker & Attendee record with Organizer Role. For this, you need to setup 2 Sharing Rules as per the below details.

- Speaker Object - Create a Sharing Rule which will share the Speaker records with the Role Organizer. And the permission should be Read/Edit.
- Attendee Object - Create a Sharing Rule which will share the Attendee records with the Role Organizer. And the permission should be Read/Edit.

5.3 Milestone 2

Apex Class Development : -

Develop a reusable Apex Class which contains a method to insert the Error log Object records. This method must contain the parameters to get the dynamic details of the fields (Log Date/Tile, Log Details & Process Name)

5.4 Milestone 3

Trigger Development (Event - Speaker Object) :-

Develop a Trigger on Event - Speaker object which will throw an error if the Speaker Selected on Event - Speaker Record already have an Event against his name. i.e - For a Speaker there will be only one event at a time. Reject Duplicate Bookings

- In which object the Trigger will be (Event - Speaker)
- What are the events (before insert , before update)

Output - Check the duplicate bookings and throw the error

5.4 Milestone 4

Trigger Development :-

Develop an Apex Trigger on Event Attendee Object (Whenever a New Record gets created) to send the email to Attendee saying that registration has been confirmed . In the email include the basic details of attendee like name, email, phone. Also, send an email to attendee. Use below format for email

Subject :-

Pass for the “ Event name Here”

Email Body :-

Dear Attendee Name,

Thank you for registering for “ Event Name Here” which will be Organized on “ Event Date Here ” & will be held in “Event Location Here”. We are excited to have you, see you in the event.

Find the Google Map Location for the Event Here. ([https:// www.google.com/maps/place/](https://www.google.com/maps/place/)) +

Add your Location Here Thanks,

“Organizer Name Here”

Must Have - Bulk record handling along with the Exception handling. Use of Handler/Helper Class

5.5 Milestone 5

Develop Unit Test Case :-

Develop the Unit Test for the above Apex Trigger(s) & The code coverage must be at-least 90% for Trigger & Handler/Helper class.

Must Have -

1. At-least 90% code coverage
2. Use of @TestSetup annotation
3. Use of Test.startTest & Test.stopTest
4. Use of Asserts Methods
5. Test must be a test case for negative test case.
6. All test cases must be successfully passed.
7. Use of TestUtility Class (optional)

5.6 Milestone 6**Develop Apex Batch :-**

Write an Apex Batch which should purge(delete) all the event records which are more than 2 months old & have been Organized. Hint - Use End Date of the Event & live? checkbox of Event Record.

In the finish method of the batch apex send an email to “sfdcpanther@gmail.com” email address saying that execution of the batch has been processed.

1. A handler class must be used for the batch
2. Send Email from the finish method as outlined in the requirement

5.7 Milestone 7

Develop Unit Test - Develop the unit test for the above batch with at-least 95% code coverage.

Must Have -

1. Unit Test must be tested for at-least 199 records
2. Code coverage must be 85%

5.8 Milestone 8

Make Call-out to SmartyStreets API :-

SmartyStreets API is used to verify the address. In this milestone you need to consume <https://smartystreets.com/products/apis/us-street-api>.

Requirement - The business ask is to verify the address of the Location Object. Use the above API to verify the address & update “**Location Verified**” field on Location Object.

Must Have - Use of Error Handling and code-reusability

5.9

MileStone 9

Milestone 9.1

- a. Setup VS Code for Non-Scratch Org into your Machine and Connect with Salesforce.
- b. Retrieve the Salesforce Metadata From Salesforce to VS Code
- c. Deploy code to VS code.

MileStone 9.2

Create a Custom Lookup in Lightning Web Component.

Prerequisite for Custom Lookup in LWC

Milestone 9.3

Add Event to Salesforce: -

Before we start create a new RichText Area field on event object and Label it as Event Detail.

Develop a Lightning Web Component which will display the Form Input for the Event to take the input from the user and save the Event Record. The Input Form should contain the below fields from Event object.

- a Name (Custom Field)
- b. Organizer (Lookup)
- c. Start Date
- d. End Date
- e. Max Attendees
- f. Lookup (Location)
- g. Event Detail Rich Text Field

NOTE :-

- 1. We will be using Lightning ui record API to create the record that means no ApexClass is required.*
- 2. After Creating the Record, user should be redirect to event detail page.*

5.10

MileStone 10

Event Detail Component : -

Develop a Lightning Web component to display the event details. This Component will contains some additional details like Speaker Details, Location Details & All Attendees who are attending the Event. So basically this component will have multiple tabs like below. Clicking on Each Tab it should open the detail related to Event.

Note: - This component will be used to display the Event Details.

Event Details	Speaker Details	Location Details	Attendees Details

Part 1 -

1. Event Details - LDS
2. Speakers - Data-table

Part 2 -

3. Location Details
4. Attendees Details

5.11 Milestone 11

Develop Event List Component : -

Develop a Lightning Web Component which display all the events in a DataTable which are Live and on clicking on Event Name page it should redirect to Event Detail Page. Event that are displayed in data-table that StartDate of the event record should be either equals to today or greater.

Must Have: - Search Functionality to search the event based on the name, start date & location.

We will implement this component in 2 part and details are as below.

Part 1 - Develop the component and Display the record in the data-table

Part 2 - Implement the search functionality

5.12 Milestone 12

attendeeEvents Component :-

Develop a Lightning web component which will sit inside the detail page of the Attendee object and display the details of the Event that attendee has attended or registered.

The component must have 2 sections

- 1 - Upcoming Events: - Display all the event that attendee has registered
- 2 - Past Events - Display all the Event that the user was part of.

Like the Event Detail Component & Event List Component, we will also implement in 2 part. Find the details below.

Part 1 - Implement the Upcoming Event functionality and use lightning accordion to have two separate sections

Part 2 - Implement the Past Events functionality

5.13 Milestone 13

Modify Event Detail Component Add New Speaker Button on Event Speaker Tab and New Attendee on Event Attendees Tab to associate the Speakers and attendees to an event.

Note- We are going to use Lightning Navigation to open the create record wizard for the same. For Example, to create the Event Speaker record we can use below code.

5.14 Milestone 14

Expose Salesforce Object - Develop & Apex Rest to send the Event Details to 3rd party in JSON format.

Note -

Only the event which are live and start date is greater than today should be exposed. You only need to develop the class. Need not to worry about 3rd party.

5.15 Milestone 15

1. Modify the Speaker Object
 - a. Create a URL field (Profile URL)
 - b. Create a Rich Text Area Field (About Me)
2. Add speakerTile Lightning Web Component

5.16 Milestone 16

1. Identify and Setup the Community
 - a. Identify the community for the Project
 - b. Enable the Community
 - c. Create the Community

Setup Login & Registration Page

- d. Create a Pick-list field on User Object
 - i. You Are ? (Organizer , Attendee)
 - ii. Default “Attendee” value for Pick-list

Setup Community Profiles

- e. Add Necessary Tabs in Community for navigation
- f. Clone “**Customer Community Plus User**” Profile and Create 2 new profiles

- i. MaxFit Organizer
- ii. MaxFit Attendee
- g. Provide the Object Level Permission same as Organizer & Attendee Profile
- h. Profile the Field Level Access to all 2 profiles that we have created.

5.17 Milestone 17

Make some cosmetic changes in community.

5.18 Milestone 18

1. Add Event Detail Component in Community
 - a. Modify eventDetails Component
 - b. Create a Custom Page for Event Object
2. Add attendeesEvent component in Community
 - a. Modify attendeesEvents Component
 - b. Create a Custom Page for Attendee Object
2. Make “**New**” button visible in Community
3. Add Image URL to CSP Sites
4. Add Logo for Community
5. Provide field Level Permission to Community Guest User Profile

CHAPTER – 6

PROJECT DESCRIPTION

CLASSES ::

1. AttendeeEventsService.cls

```
public with sharing class AttendeeEventsService {  
    @AuraEnabled  
    public static List<Event_Attendee_c> upcomingEvets(String attendeeId) {  
        List<Event_Attendee_c> eventLists = [  
            SELECT  
                Id,  
                Name,  
                Event_c,  
                Event_r.Location_c,  
                Event_r.Location_r.Name,  
                Event_r.Event_Organizer_c,  
                Event_r.Event_Organizer_r.Name,  
                Event_r.Name_c,  
                Event_r.Start_DateTime_c  
            FROM Event_Attendee_c  
            WHERE  
                Attendee_c = :attendeeId  
                AND Event_r.Start_DateTime_c >= today  
                AND Event_r.Live_c = true  
            WITH SECURITY_ENFORCED  
        ];
```

```

return eventLists;
}

```

@AuraEnabled

```

public static List<Event_Attendee_c> pastEvents(String attendeeId) {
    List<Event_Attendee_c> eventLists = [
        SELECT
            Id,
            Name,
            Event_c,
            Event_r.Location_c,
            Event_r.Location_r.Name,
            Event_r.Event_Organizer_c,
            Event_r.Event_Organizer_r.Name,
            Event_r.Name_c,
            Event_r.Start_DateTime_c
        FROM Event_Attendee_c
        WHERE Attendee_c = :attendeeId AND Event_r.Start_DateTime_c < today
        WITH SECURITY_ENFORCED
    ];
    return eventLists;
}
}

```

2. CustomSearchController.cls

```

public with sharing class CustomSearchController {

    @AuraEnabled
    public static String searchRecords(String objName, String fieldName, String searchKey){
        String searchKeyword = searchKey + '*!';
    }
}

```

```

        String returningQuery = objName+' ( Id, '+fieldName+')';
        String query = 'FIND :searchKeyword IN ALL FIELDS RETURNING '+returningQuery+'
LIMIT 2000';
        List<List<sObject>> subjectList = Search.query(Query);
        return JSON.serialize(subjectList);
    }
}

```

3.DeleteEventBatch

```

public class DeleteEventBatch implements Database.Batchable<sObject> {

    public Database.QueryLocator start(Database.BatchableContext batchContext){
        String Query = 'Select Id, Name From Event_c Where Live_c = true AND End_Date_Time_c
< LAST_N_MONTHS:2';
        return Database.getQueryLocator(Query);
    }

    public void execute(Database.BatchableContext batchContext, List<Event_c> eventList ){
        if(!eventList.isEmpty()){
            List<Database.DeleteResult> deleteResult = Database.delete(eventList);
            for(Database.DeleteResult dr : deleteResult){
                if(!dr.isSuccess()){
                    List<Database.Error> errors = dr.getErrors();
                    String errorInString = String.join(errors, ',');
                    TransactionLogHandler.doHandleExceptionWithError(errorInString,
'DeleteEventBatch');
                }
            }
        }
    }
}

```

```

public void finish(Database.BatchableContext batchContext){
    Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage();

    mail.setSubject('Batch Processing Completed');
    List<String> toAddress = new List<String>();
    toAddress.add('sfdcpanther@gmail.com');
    mail.setToAddresses( toAddress );
    mail.setSenderDisplayName('SFDCPanther');

    mail.setHtmlBody('The DeleteEventBatch Batch Procoessing has been Completed');
    List<Messaging.SingleEmailMessage> emails = new List<Messaging.SingleEmailMessage>
{mail};
    Messaging.sendEmail(emails);
}
}

```

4.DeleteEventBatchTest

```

@Test
public class DeleteEventBatchTest {

    @testSetup
    public static void testSetupData(){
        Event_Organizer_c org = new Event_Organizer_c (
            Name = 'Amit Singh',
            Phone_c = '9807654321',
            Email_c = 'sfdcpanther@gmail.com.dev1',
            Alternative_Phone_c = '9807654321',
            Alternative_Email_c ='sfdcpanther@gmail.com.dev1'

```

```

    );
    insert org;
    List<Event_c> eventList = new List<Event_c>();
    for(Integer i=0; i<=200; i++){
        Event_c event = new Event_c(
            Name_c = 'MAX FIT Campaign #' + i + 1,
            Event_Organizer_c = org.Id,
            Event_Type_c = 'Virtual',
            Frequency_c = 'Weekly',
            Max_Seats_c = 199,
            Recurring_c = true,
            Live_c = true,
            Start_DateTime_c = System.now().addMonths(-4),
            End_Date_Time_c = System.now().addDays(3).addMonths(-4)
        );
        eventList.add(event);
    }
    insert eventList;
}

@Test
static void sendDeleteEventTest(){
    Test.startTest();

    String jobId = Database.executeBatch(new DeleteEventBatch(), 250);

    Test.stopTest();

}
}

```


5.EventAttendeeTriggerHandler

```
public class EventAttendeeTriggerHandler {

    public static void sendConfirmationEmail(List<Event_Attendee_c> newRecordList ) {

        Set<Id> attendeesIdsSet = new Set<Id>();
        Set<Id> eventIdsSet = new Set<Id>();

        for(Event_Attendee_c ea : newRecordList){
            attendeesIdsSet.add(ea.Attendee_c);
            eventIdsSet.add(ea.Event_c);
        }

        Map<Id,Attendee_c> attendeeMap = new Map<Id,Attendee_c>(
            [Select Id, Name, Email_c From Attendee_c WHERE Id IN : attendeesIdsSet]
        );

        Map<Id, Event_c > eventMap = new Map<Id, Event_c > ( [Select Id, Name_c,
Start_DateTime_c, Event_Organizer_c , Event_Organizer_r.Name,
                                Location_c , Location_r.Name, Location_r.City_c,
                                Location_r.State_c, Location_r.Country_c,
                                Location_r.Postal_Code_c, Location r_Street_c
                                FROM Event_c WHERE ID IN: eventIdsSet]
        );

        List<Messaging.SingleEmailMessage> emailList = new List<Messaging.SingleEmailMessage>();

        for(Event_Attendee_c ea : newRecordList){

            Attendee_c att = attendeeMap.get(ea.Attendee_c);
```

```

Event_c evt = eventMap.get(ea.Event_c);

Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage();

mail.setSubject('Pass for the '+evt.Name_c);
List<String> toAddress = new List<String>();
toAddress.add(att.Email_c);
mail.setToAddresses( toAddress );
mail.setSenderDisplayName(evt.Event_Organizer_r.Name);

String locaton = 'https://www.google.com/maps/place/'+evt.Location_r.Street_c+'
'+evt.Location_r.City_c+' '
    +evt.Location_r.State_c+' '+
    evt.Location_r.Country_c+' '+evt.Location_r.Postal_Code_c;
String hrefForLocation = '<a href="'+locaton+'"'+'target="_blank">Here</a>';
String emailBody = 'Dear ' + att.Name + ',<br/><br/>'+
    'Thank you for registering for '+evt.Name_c+' which will be Organized on '+
    evt.Start_DateTime_c+' & will be held in '+evt.Location_r.Name
    +'.<br/>We are excited to have you,'+
    'see you in the event. <br/>'+
    'Find the Google Map Location for the Event '+hrefForLocation+'.<br/><br/><br/>'+
    'Thanks,<br/>'+evt.Event_Organizer_r.Name;

mail.setHtmlBody(emailBody);
emailList.add(mail);

}

try{

List<Messaging.SendEmailResult> results = Messaging.sendEmail(emailList, false);
for(Messaging.SendEmailResult email : results){

```

```

        System.debug(email.isSuccess());
        if(!email.isSuccess()){
            List<Messaging.SendEmailError> errors = email.getErrors();
            TransactionLogHandler.doHandleExceptionWithError(JSON.serialize(errors),
'EventAttendeeTriggerHandler');
        }
    }
} catch(System.Exception ex){
    TransactionLogHandler.doHandleException(ex, 'EventAttendeeTriggerHandler');
}
}
}

```

6.EventAttendeeTriggerTest

```

@isTest
public class EventAttendeeTriggerTest {

    @testSetup
    public static void setupData(){

        Event_Organizer_c org = new Event_Organizer_c (
            Name = 'Amit Singh',
            Phone_c = '9807654321',
            Email_c = 'sfdcpanther@gmail.com.dev1',
            Alternative_Phone_c = '9807654321',
            Alternative_Email_c ='sfdcpanther@gmail.com.dev1'
        );
        insert org;
    }
}

```

```

Event_c event = new Event_c(
    Name_c = 'MAX FIT Campaign',
    Event_Organizer_c = org.Id,
    Event_Type_c = 'Virtual',
    Frequency_c = 'Weekly',
    Max_Seats_c = 199,
    Recurring_c = true,
    Live_c = true,
    Start_DateTime_c = System.now(),
    End_Date_Time_c = System.now().addDays(3)
);
insert event;
Attendee_c att = new Attendee_c(
    Name = 'Akash Jain',
    Email_c = 'akash.jain@gmail.com',
    Phone_c = '9807654321'
);
insert att;

Event_Attendee_c evtAtt = new Event_Attendee_c(Event_c = event.Id, Attendee_c_ = att.Id);
insert evtAtt;
}

@isTest
static void sendEmailTest(){
    Test.startTest();
    try{
        integer i = 10/0;
    }catch(System.Exception ex ){

    }
}

```

```
        Test.stopTest();  
    }  
}
```

7 Custom Objects :

- ❖ Attendee c
- ❖ Customer c
- ❖ Error_Log c
- ❖ EventSpeakers c
- ❖ Event_Attendee c
- ❖ Event_Organizer c
- ❖ Event c
- ❖ Location c
- ❖ Speaker c

8 Trigger :

8.1 EventAttendeeTrigger.trigger

```

trigger EventAttendeeTrigger on Event_Attendee_c (after insert) {

    if(Trigger.isAfter && Trigger.isInsert){

        EventAttendeeTriggerHandler.sendConfirmationEmail(Trigger.New);

    }

}

```

8.2 EventSpeakerTrigger.trigger

```

trigger EventSpeakerTrigger on EventSpeakers_c (before insert, before update ) {

    // Step 1 - Get the speaker id & event id

    // Step 2 - SOQL on Event to get the Start Date and Put them into a Map

    // Step 3 - SOQL on Event - Spekaer to get the Related Speaker along with the Event Start Date

    // Step 4 - Check the Conditions and throw the Error


    //Step 1 - Start

    Set<Id> speakerIdsSet = new Set<Id>();

    Set<Id> eventIdsSet = new Set<Id>();


    for( EventSpeakers_c es : Trigger.New ){

        speakerIdsSet.add(es.Speaker_c);

        eventIdsSet.add(es.Event_c);

    }
}

```

```
//Step 1 - End
```

```
/*
```

```
* 10 Event Records
```

```
* 1 (EventId) K -- DateTime ( Event Start_DateTime c ) V
```

```
*/
```

```
// Step 2 Start
```

```
Map<Id, DateTime> requestedEvents = new Map<Id, DateTime>();
```

```
List<Event_c> relatedEventList = [Select Id, Start_DateTime_c From Event_c
```

```
Where Id IN : eventIdsSet];
```

```
for(Event_c evt : relatedEventList ){
```

```
    requestedEvents.put(evt.Id, evt.Start_DateTime_c);
```

```
}
```

```
// Step 2 End
```

```
// Step 3 - Start
```

```
List<EventSpeakers_c> relatedEventSpeakerList = [ SELECT Id, Event_c, Speaker_c,
```

```
Event_r.Start_DateTime_c
```

```
From EventSpeakers_c
```

```
WHERE Speaker_c IN : speakerIdsSet];
```

```
// Step 3 - End
```

```
// Step 4 - Start
```

```

for( EventSpeakers_c es : Trigger.New ){ // - Salesforce Geek

    DateTime bookingTime = requestedEvents.get(es.Event_c);

    // DateTime for that event which is associated with this new Event-Speaker Record


    for(EventSpeakers_c es1 : relatedEventSpeakerList) {

        // Amit Singh == Salesforce Geek => false

        // Amit Choudhary == Salesforce Geek => false

        // Salesforce Geek == Salesforce Geek => true

        if(es1.Speaker_c == es.Speaker_c && es1.Event_r.Start_DateTime_c == bookingTime ){

            es.Speaker_c.addError('The speaker is already booked at that time');

            es.addError('The speaker is already booked at that time');

        }

    }

}

// Step 4 - End

}

```


CHAPTER – 7

LITERATURE REVIEW

7.1 ABSTRACT :

Online Event management system is used to manage all the activity related to event. In any event many service providers work simultaneously and it is very hard to manage these providers. It is also important for event organizer that he has all the contacts details of these service providers so that he can contact them any time to plan an event at given time. To manage all these activity we have developed this software. Online event management system is an salesforce project that serves the functionality of an event manager. The system allow registered user login and new user are allowed to register on the application. The system helps in the management of events, users and the aspects related to them. The project provides most of the basic functionality required for an event type, the system then allows the user to select date and time of event, place and the event equipment. All the data is logged in the database and the user is given a receipt number for his booking [\[10\]](#). The data is then send to administrator (website owner) and they may interact with the client as per his requirement. Event management is a strong and fast growing profession with a rather low level of standardization. Often we take event management as a part of project management, but we have to consider that event management has very specific concepts and issues, and needs further developed methods and tools. We classify events, we compare project management and event management, we reconsider standards in both areas, and discuss perspectives for a stronger standardization of event management in the future. Now - a – days Salesforce is considering the best CRM [\[12\]](#).

7.2 Keywords :

User,

Admin,

Events,

Book,
Event Management,
Event Satisfaction [9],
Database.

7.3 Introduction :

Event management system is used to manage all the activity related to event. In any event many service providers work simultaneously and it is very hard to manage these providers. It is also important for event organizer that he has all the contacts details of these service providers so that he can contact them any time to plan an event at given time. To manage all these activity we have developed this software. event management is a problem because it requires much time [13]

This is an online event management system, software project that serves the functionality of an event manager. The project provides most of the basic functionality required for an event .It allows the user to select from list of event types. Events Management System is very helpful for events. This application being as a platform to know the events, to apply for the events

To understand use of this application consider the flow of actions happening, by this application user can register the students, after registering, user can login, after login, event details including name ,contact, address, venue of the event, date, event conducting time, cost of events etc. After receiving SMS student can register through application.

This system can be implemented in hotels, clubs for booking events. The system can also be used as software to promote the entire booking places. The user gets all the resources at a single place instead of wondering round for these. This system is effective and saves time and cost of the user. This study develops a theory explaining how an vendor(event manager) can regain control over its attendee(user).[4]

The most effective thing is that it is based on salesforce . This can be access from anywhere in the world .[1]

7.4 GUI :

The main thing is increase in flexibility and adaptiveness for the functions that the salesforce performs. During disruptions such as the COVID-19 pandemic,doing all the work in an online mode is vital. [\[2\]](#)

This system can be implemented in hotels, clubs for booking events and many more . The system can also be used as software to promote the entire booking places. The user gets all the resources at a single place instead of wondering round for these [\[3\]](#). This system is effectiveand saves time and cost of the user.

Admin view, updates, delete customer and vendor's records . Admin view update booking record, verify email and messages, receipt mail. If any vendors rating will constantly not good then admin can remove vendor.

Registration: In the users and vendors registration form, the users and vendors has to enter the users and vendors name, address, email identification and the phone number. The users and vendors details are allowed to store in the centralized database with an automatic generated event id. Event id can be generated automatically [\[7\]](#).

Check Availability: In the check availability module user check the availability of the halls and then book the hall for their event.

Salespeople must be adopted to work successfully in potentially frustrating situations also [\[11\]](#).

Customer: In the customer module customer doing own registration , login , search event view , update , delete own profile, select event date and time, select event , select place, select equipment, book event, view confirmation mail , change password. Search is optional they can search event without login. After booking vendors provide notification like book successfully and then logout.

Vendors: Vendors register, login, view, insert, update, delete own profile, confirm booking of event for customer. Vendors manage events, manage places, manage equipment, manage food, and Send notification of event booking to customer. They can see the reviews provided by customer after organizing event successfully.

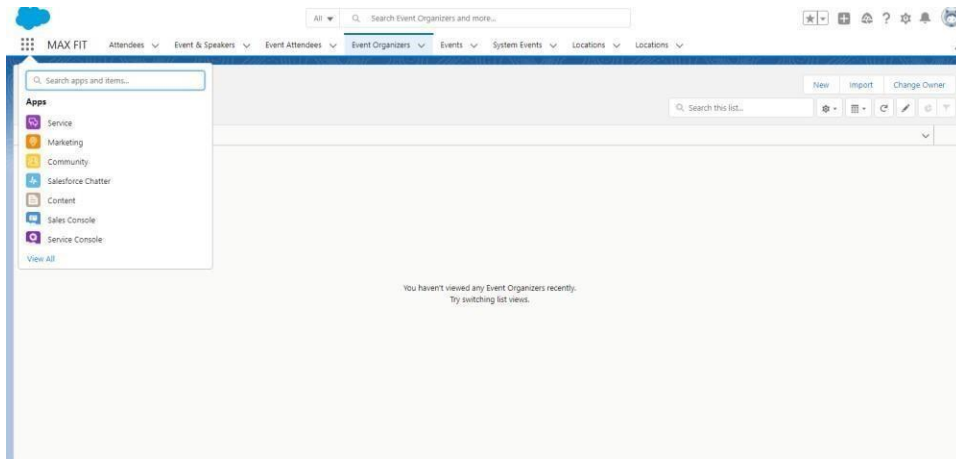


FIGURE : Event Management System Overview

In Event Management System there are many tabs which is shown through this diagram :

Attendees , Event and Speakers , Event Attendee , Events , Events Organizers , System Events , Locations.

FIGURE : Create New Event

If any person want to create event so they have to registered on new Event .

In which they have to filled some details like Name of Event , Location. Start Date/ Time , Recurring Max Seats , Live, Location , Status, Event Organizer, End Date/ Time, Event Type, Frequency, Event Detail. Various validation are used in this :

1. Recurring? checkbox is checked then user must need to fill Frequency field & Ifcheckbox is unchecked then User can not select Frequency field.
2. If Virtual is Selected as Value for Event Type field then Prevent User to Select Location on Event Record.
3. End Date/Time must be at-least 1 day ahead of Start Date/Time (If there is a value in End Date/Time field)
4. If Event Type field value is In-Person then user must need to select Location on Event Record.

The screenshot shows a web form titled "New Event Organizer". Below the title is a section labeled "Information". This section contains two columns of input fields. The left column has three fields: "* Name" (containing "ab"), "* Phone" (containing "78978778778"), and "Alternative Phone" (containing "9808064553"). The right column has three fields: "* Email" (containing "ab@gmail.com"), "Alternative Email" (empty), and "Location" (a search input with the placeholder "Search Locations..."). At the bottom of the form, there are three buttons: "Cancel", "Save & New", and "Save".

FIGURE : Event Organizer

In Event Organizer organizer have to fill this information

Name , Phone no , Alternative Phone No , Email , Alternative Email , Location.

FIGURE : New Event Attendee

In this Attendee will fill all the details like Event(they will choose that event) , Attendee .

Various validation are already used in this :

1. Attendee can only be associated with the Event whose End Date is in future & Event Live Checkbox is checked and Event is accepting the Attendees (means Remaining Seats field value is not 0)

FIGURE : New System Event

attention has been focused on various aspects of salesforce turnover in recent years[\[14\]](#)

7.5 Results and Discussions:

The event management software was developed with proper planning and guidance. Iterative waterfall model will be used during the development of this project. Planning at each stage will be done properly. The design phase will be fully designed as per protocol. Unit testing of each module and sub-module will be performed. After that the modules and sub-modules will be integrated and an integrated testing will be performed. The project is meant to serve the managing purpose for all kind of events with a complete and easy approach.

This software examine a set of salesforce management factors that contribute to successful new product [\[6\]](#)

Event management is the application of project management to the creation and development of small and/or large-scale personal or corporate events such as festivals, conferences, ceremonies, weddings, formal parties, concerts, or conventions. It involves studying the brand, identifying its target audience, devising the event concept, and coordinating the technical aspects before actually launching the event.

It is very helpful because large numbers of small organizations and prosumers have shifted away from managing data on their own devices and now they are heavily reliant on service-providers for both storage and processing of their data [\[15\]](#).

This paper considers the impacts of event management system which is based on salesforce on a large company.[\[5\]](#)

7.6 Conclusion :

Event Management System is user friendly and cost effective system, it is customized with activities related to event management life-cycle. It provides a new edge to management industry.

Our project is only a humble venture to satisfy the needs to manage their project work. Several user friendly coding have also adopted. This package shall prove to be a powerful package in satisfy all requirements of the user. The objective of software planning is to provide a frame work that enable the manager to make reasonable estimate made within a limited time frame at the beginning of the software project and should be update regularly as the project regularly.

To enhance salesforce performance, a manufacturer's control and a retailer's control must recoup for each other's weaknesses [8].

At the end it is concluded that we have made effort on following points...

A description of background and context of the project and its relation to work already done in the area

Made statement of the aims and objectives of the project. # The description of the purpose, scope and applicability.

We define the project on which we are working in project.

We describe the requirement specifications of the system and actions that can be done on these things.

We designed user interface and security issues related to system.

Finally the system is implemented and tested according to the test cases

7.7 REFERENCES

1. Rahul, G., and M. Suresh. "Modelling of factors influencing agility of salesforce in management responsibility orientation of retail stores: A TISM approach." *Materials Today*:
2. *Proceedings* (2021).
3. Sharma, Arun, Deva Rangarajan, and Bert Paesbrugghe. "Increasing resilience by creating an adaptive salesforce." *Industrial Marketing Management* 88 (2020): 238-246.
4. Mason, Mike. "Overcoming the risks of privileged user abuse in Salesforce." *Network Security* 2018, no. 8 (2018): 6-8.
5. Kim, Stephen K., and Yeon Sung Jung. "Regaining control of salesforce." *Industrial Marketing Management* 73 (2018): 84-98.
6. Cronin, B., and E. Davenport. "Laptops and the marketing information chain: The benefits of salesforce automation." *International journal of information management* 10, no. 4 (1990): 278-
7. 287.
8. Fraenkel, Stefan, Darek M. Haftor, and Natallia Pashkevich. "Salesforce management factors for successful new product launch." *Journal of Business Research* 69, no. 11 (2016): 5053-5058.
9. Dai, Yue, and Xiuli Chao. "Salesforce contract design and inventory planning with asymmetric risk-averse sales agents." *Operations Research Letters* 41, no. 1 (2013): 86-91.
10. Kim, Stephen K., and Amrit Tiwana. "Dual Control of Salesforce in Partially Integrated Channels." *Journal of Retailing* 93, no. 3 (2017): 304-316.
11. Miao, C. Fred, and Kenneth R. Evans. "Motivating industrial salesforce with sales control systems: An interactive perspective." *Journal of Business Research* 67, no. 6 (2014): 1233-1242.
12. Grace, David, and Tom Pointon. "Marketing research through the salesforce." *Industrial Marketing Management* 9, no. 1 (1980): 53-58.

13. Dubinsky, Alan J., and Mary E. Lippitt. "Techniques that reduce salesforce frustration." *Industrial Marketing Management* 9, no. 2 (1980): 159-166.
14. Munson, J. Michael, and W. Austin Spivey. "Salesforce selection that meets federal regulation and management needs." *Industrial Marketing Management* 9, no. 1 (1980): 11-21.
15. Dubinsky, Alan J. "Recruiting college students for the salesforce." *Industrial Marketing Management* 9, no. 1 (1980): 37-45.
16. Sager, Jeffrey K., Charles M. Futrell, and Rajan Varadarajan. "Exploring salesperson turnover: A causal model." *Journal of Business Research* 18, no. 4 (1989): 303-326.
17. Clarke, Roger. "Can small users recover from the cloud?." *Computer law & security review* 33, no. 6 (2017): 754-767.

