Max Fit Event Management Application

Project report submitted in partial fulfilment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering

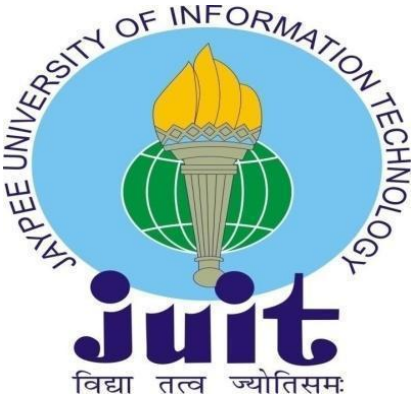
By

RIDHAM GODHA (191316)

UNDER THE SUPERVISION OF

Dr Abhilasha Sharma

to



Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology, Waknaghat, 173234, Himachal Pradesh, INDIA

CERTIFICATE

Candidate's Declaration

I hereby declare that the work presented in this report entitled Max Fit event management application in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering submitted in the Department of Computer Science and Engineering, the Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from July 2022 to May 2023 under the supervision of Dr Abhilasha Sharma (Assistant Professor , Department of Computer Science & Engineering and Information Technology).

I also authenticate that I have carried out the above-mentioned project work under the proficiency stream Cloud computing.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Ridham Godha,191206

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Dr Abhilasha Sharma
Assistant Professor
Department of Computer Science & Engineering and Information Technology

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: .....

Type of Document (Tick): PhD Thesis | M.Tech Dissertation/ Report | B.Tech Project Report | Paper

Name: _____ **Department:** _____ **Enrolment No** _____

Contact No. _____ **E-mail.** _____

Name of the Supervisor: _____

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): _____

_____

_____

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages =
- Total No. of Preliminary pages =
- Total No. of pages accommodate bibliography/references =

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at .....(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor) **Signature of HOD**

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

<table>
<tr><th>Copy Received on</th><th>Excluded</th><th>Similarity Index (%)</th><th colspan="2">Generated Plagiarism Report Details (Title, Abstract & Chapters)</th></tr>
<tr><td rowspan="2"></td><td rowspan="4">• All Preliminary Pages
• Bibliography/Images/Quotes
• 14 Words String</td><td rowspan="2"></td><td>Word Counts</td><td></td></tr>
<tr><td rowspan="2">Report Generated on</td><td>Character Counts</td><td></td></tr>
<tr><td rowspan="2"></td><td rowspan="2">Submission ID</td><td>Total Pages Scanned</td><td></td></tr>
<tr><td>File Size</td><td></td></tr>
</table>

Checked by
Name & Signature **Librarian**

.....

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com

ACKNOWLEDGEMENT

Firstly, I express our heartiest thanks and gratefulness to almighty God for his divine blessing to make it possible to complete the project work successfully. I am grateful and wish my profound indebtedness to Supervisor **Dr Abhilasha Sharma**, **Assistant Professor** , Department of CSE Jaypee University of Information Technology, Wakhnaghat. The deep knowledge & keen interest of my supervisor in the field of computer science helped me to carry out this project. Her endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, and reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

I would like to express my heartiest gratitude to **Dr Abhilasha Sharma, Department of CSE**, for her kind help to finish my project.

I would also generously welcome each one of those individuals who have helped me straightforwardly or in a roundabout way in making this project a win. In this unique situation, I want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated our undertaking.

Finally, I must acknowledge with due respect the constant support and patience of my parents.

Ridham Godha (191316)

TABLE OF CONTENT

Content Page No.

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Event management is a complex task that requires efficient planning, coordination, and execution. In this report, we present an event management application developed using Salesforce technology. The application is designed to streamline the process of managing events, attendees, speakers, and organisers. By leveraging the power of Salesforce, the application provides a comprehensive solution that is scalable and adaptable to the changing needs of the organisation.

The event management application developed using Salesforce technology is a comprehensive solution for managing events, attendees, speakers, and organisers. The system comprises several custom objects, including Locations, Event Organizers, Events, Attendees, Speakers, Event Attendees, Event-Speakers, and Error Logs. The Locations object stores the details of the venue where events are held. The Event Organizers object contains the information of the individuals or companies who organise the events. The Events object is used to manage the details of the events, such as date, time, venue, and organiser. The Attendee object stores the details of the individuals who register for the event, and the Speakers object contains the information of the individuals who deliver speeches at the event.

The Event Attendees object is a junction object that establishes a many-to-many relationship between the Events and Attendee objects, allowing the tracking of the attendees who registered for specific events. Similarly, the Event-Speakers object is another junction object that establishes a many-to-many relationship between the Events and Speakers objects, allowing the tracking of the speakers who deliver speeches at specific events.

The Error Logs object is used to log errors that occur during the execution of the application, providing an easy way to track and resolve issues.

Overall, the event management application developed using Salesforce technology provides a complete solution for managing events, attendees, speakers, and organisers. The application has been designed to be scalable and adaptable to the changing needs of the organisation. The use of custom objects in the application provides a flexible and extensible data model that can be easily modified to meet the requirements of different types of events.

CHAPTER 1 : INTRODUCTION

Event management is a challenging task that requires a lot of planning, coordination, and execution. With the advancement of technology, it has become easier to manage events efficiently. In this report, we present an event management application developed using Salesforce technology. The application provides a comprehensive solution for managing events, attendees, speakers, and organizers.

Salesforce is a cloud-based customer relationship management (CRM) platform that provides various tools for businesses to manage their customer interactions. The platform has gained immense popularity due to its flexibility, scalability, and ease of use. Salesforce technology is ideal for developing event management applications as it provides several benefits.

The Custom Objects which were used in the project are:

1) Locations: This object stores the details of the venue where events are held. It includes information such as the name, address, capacity, and facilities available at the venue.
2) Event Organizers: This object contains the information of the individuals or companies who organize the events. It includes details such as the name, contact information, and organization details.
3) Events: This object is used to manage the details of the events, such as date, time, venue, and organizer. It also includes information about the agenda, attendees, and speakers.
4) Attendees: This object stores the details of the individuals who register for the event. It includes information such as the name, email, contact information, and registration details.
5) Speakers: This object contains the information of the individuals who deliver speeches at the event. It includes details such as the name, contact information, and topic of the speech.
6) Event Attendees: This object is a junction object that establishes a many-to-many relationship between the Events and Attendee objects. It allows the tracking of the attendees who registered for specific events.
7) Event-Speakers: This object is another junction object that establishes a many-to-many

relationship between the Events and Speakers objects. It allows the tracking of the speakers who deliver speeches at specific events.

8) Error Logs: This object is used to log errors that occur during the execution of the application. It provides an easy way to track and resolve issues that may arise during the event management process.

The process involved several steps, including gathering requirements, design, Development, Testing, deployment, maintenance.

The aim of this project is to develop an event management application using Salesforce technology that provides a comprehensive solution for managing events, attendees, speakers, and organizers. The application aims to streamline the event management process by providing a centralized platform for managing all aspects of an event, from planning and coordination to execution and follow-up. The application also aims to improve the user experience for event managers and attendees by providing an intuitive and user-friendly interface and automating many of the manual tasks associated with event management. Overall, the aim of this project is to provide a robust and efficient event management solution that helps businesses to organize and manage successful events with ease

1.1 Introduction

In today's fast-paced business environment, event management has become an essential part of many organizations. To help businesses manage events more efficiently, we have developed an event management application using Salesforce technology. Salesforce is a powerful cloud-based platform that offers a wide range of tools and features for building custom applications, making it an ideal choice for event management.

The event management application includes several custom objects designed to meet the specific needs of event managers. These custom objects are:

1) Locations: This custom object stores the details of the venue where events are held. It includes fields such as the name, address, capacity, and facilities available at the venue.
2) Event Organizers: This custom object contains the information of the individuals or companies who organize the events. It includes fields such as the name, contact information, and organization details.

3) Events: This custom object is used to manage the details of the events, such as date, time, venue, and organizer. It also includes information about the agenda, attendees, and speakers.
4) Attendees: This custom object stores the details of the individuals who register for the event. It includes fields such as the name, email, contact information, and registration details.
5) Speakers: This custom object contains the information of the individuals who deliver speeches at the event. It includes fields such as the name, contact information, and topic of the speech.
6) Event Attendees: This custom object is a junction object that establishes a many-to-many relationship between the Events and Attendee objects. It allows the tracking of the attendees who registered for specific events.
7) Event-Speakers: This custom object is another junction object that establishes a many-to-many relationship between the Events and Speakers objects. It allows the tracking of the speakers who deliver speeches at specific events.
8) Error Logs: This custom object is used to log errors that occur during the execution of the application. It provides an easy way to track and resolve issues that may arise during the event management process.

Each custom object has been carefully designed to meet a specific need in the event management process. By using these custom objects, event managers can easily track all aspects of an event, from planning to execution and follow-up. For example, the MAX FIT object allows event managers to easily identify and manage the fitness centers or gyms where events are held, while the Attendees object makes it easy to keep track of all the individuals who register for an event. Additionally, the use of junction objects such as Event Attendees and Event-Speakers allows event managers to establish relationships between various objects and track attendance and participation in specific events.

The fields in each custom object have been carefully selected to capture the information that is needed to manage events effectively. For example, the Locations object includes fields such as the name, address, and capacity of the venue, while the Speakers object includes fields such as the name, contact information, and topic of the speech. These fields allow event managers to easily manage all aspects of an event, from scheduling to communication with speakers and attendees.

In addition to the custom objects, the event management application also includes several other features that make it a comprehensive solution for event management. For example, the application includes

workflows and automation rules that can be used to automate many of the manual tasks associated with event management, such as sending reminders to attendees or scheduling follow-up emails after an event. The application also includes reporting and analytics tools that allow event managers to track key metrics such as attendance rates, speaker ratings, and event feedback.

One of the key benefits of using Salesforce technology for event management is its scalability. As businesses grow and expand their events, the application can easily be customized to meet their changing needs. Additionally, Salesforce offers robust security features that ensure the safety and privacy of sensitive information such as attendee details and financial data.

Overall, the event management application developed using Salesforce technology and the custom objects mentioned above provides a powerful and flexible solution for businesses looking to streamline their event management process. With its user-friendly interface, automation features, and reporting tools, the application makes it easy for event managers to plan, execute, and follow up on successful events.

1.2 Problem Statement

Traditional event management processes can be manual, time-consuming, and error-prone. Organizing an event involves keeping track of a myriad of details, from scheduling the venue to managing speakers and attendees. Businesses are increasingly looking to technology solutions to streamline their event management processes, but off-the-shelf solutions may not be customizable enough to fit their unique needs, and custom development can be costly and time-consuming.

To address these challenges, the Event Management Application using Salesforce technology and the custom objects mentioned earlier provides a comprehensive, scalable, and customizable solution for event management. By using the application, event managers can easily track all aspects of an event, from planning to execution and follow-up, without having to rely on manual processes or generic solutions that do not fit their specific needs.

The MAX FIT custom object allows event managers to identify and manage fitness centers or

gyms where events are held, while the Attendees object makes it easy to keep track of individuals who register for an event. Additionally, junction objects such as Event Attendees and Event-Speakers help establish relationships between various objects and track attendance and participation in specific events.

The Event Management Application addresses pain points associated with traditional event management processes by providing a powerful and flexible solution tailored to the specific needs of individual businesses. By using the application, event managers can save time, reduce errors, and deliver a better attendee experience.

1.3 Objective

The primary objective of the MAX FIT project is to establish an efficient event management process, including creating, identifying, and implementing appropriate control measures. In the MAX FIT event management process, business owners can create events for their clients, with some configurations being critical while others are minor. If the owner of an event wants to cancel it, they can remove it from the event object, and all corresponding configurations will be automatically deleted. The goal of this process is to categorize events and define appropriate control measures. Attendees who register for an event will receive an email alert about it.

The second objective of the MAX FIT Event Management process is to program events in ways such as transferring client information. For instance, clients can register for one or more events that are taking place in MAX FIT. These actions indicate changes to configuration items or services that are classified as events. The event management process aims to program this type of flow and manage configurations and information in an IT service provider. Additionally, there is a product inventory of MAX FIT Gym that can be suggested by event owners to clients. After the event, clients can purchase the gym's protein products offered by MAX FIT Gym.

Overall, the event management application developed using Salesforce technology and the custom objects mentioned above provides a powerful and flexible solution for businesses looking to streamline their event management process. With its user-friendly interface, automation features, and reporting tools, the application makes it easy for event managers to

plan, execute, and follow up on successful events.

1.4 Methodology

The methodology of the Event Management Application using Salesforce technology follow the following phases:

1. Requirement gathering: The development team collects information on the requirements for the Event Management Application. This includes understanding the necessary features and functionalities and any constraints or limitations.
2. Design: In this phase the team designs the technical requirements like how many custom objects do we need, what are their types, what is the relationship between those objects and the overall flow of data.
3. Development: In this phase we writes the code for the application, including custom Apex code and Visualforce pages, and integrates it with Salesforce
4. Testing: The team conducts functional testing, integration testing, and performance testing to ensure that the application meets the specified requirements and functions as intended.
5. Deployment: After testing and approval, the application is deployed to the production environment. This involves setting up the necessary infrastructure and configuration.
6. Maintenance: The development team maintains the application after deployment by providing ongoing support and troubleshooting, as well as ensuring that it remains up-to-date and functional.

1.5 Organisation

This project report is divided into five distinct chapters, each of which is explained below:

Chapter 1: It includes a brief overview of our project, the problem statement that served as the foundation for our project's objective, our main project goals, and the methodology or approach that will be used to complete our study, which was to develop a event management project which manages the events, its corresponding speakers and its corresponding attendees.

Chapter 2: The history of music genre categorization is covered in this chapter using data from standardised books, journals, transactions, websites, and other sources. Machine learning, neural networks, and an introduction to a few distinct types of neural networks are the first topics covered. This chapter outlines how several neural network types, datasets, and accuracy evaluation approaches have been used by various academics to tackle music genre categorization model challenges, with variable degrees of success. It also explains the technique or strategy we decide to employ.

Chapter 3: This chapter's main objective is to describe in detail that how we are creating an event management application. We are describing in detail the following steps like: Gathering the requirements and understanding the necessary features and functionalities and any constraints or limitations serve as a starting point, followed by the discussion on technical difficulties and then the development of the requirements.

The custom objects that was used and the factors, including the software and hardware, that will be used to build the mode are mostly covered.

Chapter 4: This chapter provides the tests performed and the corresponding data gathered at various stages of the project to give us a clear picture about the validation which were applied in the projects and what are the constraints and limitations which are there in the application. After the tests we will get to know which methods we need to use like whether we need to use Apex triggers or flow etc in our application.

Chapter 5: This report summarises the study conducted on the Event Management Application and presents the results obtained from the model developed at the end of the project. Additionally, it identifies areas of the application that could be further strengthened or expanded upon through additional research and development. The report also highlights the creative ideas, innovations, and labor that went into the development and testing of the application, as well as the insights gained from the results obtained. The study showcases the capabilities and benefits of the Event Management Application, and the report highlights the importance of this technology in modern event management and its potential to transform the industry.

Chapter-2: LITERATURE SURVEY

Eventbrite: Eventbrite is a widely used platform that allows users to create, promote, and sell tickets for events. It offers features such as event registration, ticketing, attendee management, and event marketing tools.

Cvent: Cvent is an all-in-one event management platform that offers comprehensive solutions for event planning, registration, onsite management, and analytics. It provides features like event website creation, attendee management, agenda building, and event reporting.

Eventzilla: Eventzilla is an online event registration and ticketing platform. It offers features such as event registration forms, payment processing, email notifications, attendee management, and event analytics.

RegOnline: RegOnline, a part of Cvent, is an event registration and management software. It provides tools for event registration, attendee management, email communication, and reporting.

Aventri: Aventri (formerly etouches) is a cloud-based event management software that offers end-to-end solutions for event planning, registration, onsite management, and post-event analysis. It includes features like event website creation, registration forms, attendee management, and event reporting.

Bizzabo: Bizzabo is an event management platform that helps users create, manage, and promote events. It offers features like event registration, networking tools, event marketing, and analytics.

Chapter 3: SYSTEM DESIGN & DEVELOPMENT

The overall aim of this study is to design and develop an event management application that can efficiently manage different types of events. To achieve this goal, we will take a step-by-step approach that involves analyzing the current requirements, selecting the appropriate technology, custom object, and database.

After extensive research on the most suitable technology for developing this application, we have concluded that Salesforce technology is the best choice. This is due to its several benefits, including its ability to offer a comprehensive platform for developing robust and scalable applications, its user-friendly interface, and its ability to integrate with other applications easily.

Moreover, we have identified custom objects like Events, Event Organizers, Attendees, Event Attendees, Speakers, Locations, and Error Logs that will help us achieve our goal of efficiently managing events. These objects have been designed with custom fields that capture all the necessary details and information required to ensure the smooth running of events.

Figure 13 depicts the developer org view after custom objects and fields were created in Salesforce.

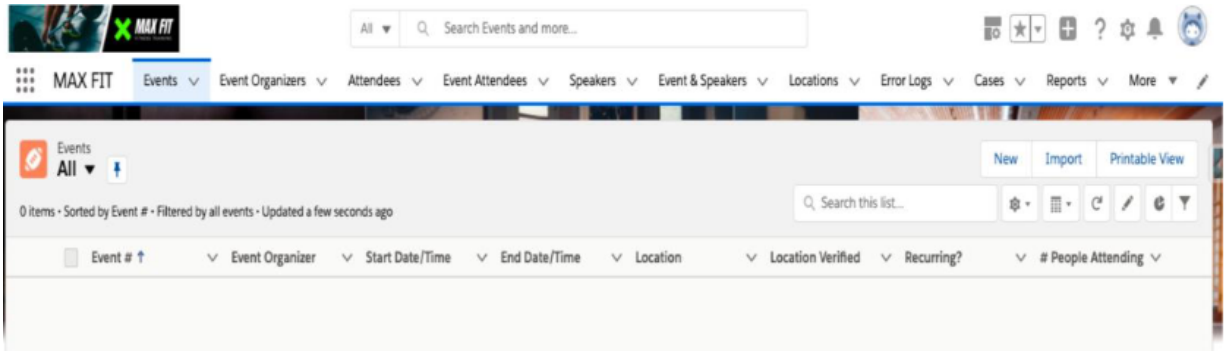


Figure 1: After Creating the Objects/Fields
App View

-The successful implementation of an event management application using Salesforce technology can bring numerous benefits :

Firstly, it can help event organizers to streamline their event management process and improve the overall attendee experience. The application can provide them with a centralized platform to manage all aspects of the event, including registrations, speaker and attendee management, scheduling, and feedback collection. This can reduce the manual effort and errors, and improve the efficiency and accuracy of event management.

Secondly, the application can benefit the attendees by providing them with an easy and intuitive platform to register for events, view schedules, and interact with other attendees and speakers. This can enhance their engagement and satisfaction with the event, leading to increased attendance and positive word-of-mouth marketing.

Thirdly, the application can benefit the society by enabling the hosting of more successful events, including conferences, seminars, workshops, and charity events. These events can bring together people from diverse backgrounds and industries, promote knowledge sharing and networking, and contribute to the overall growth and development of the society.

Overall, an event management application using Salesforce technology can bring significant benefits to the market and society by improving the event management process, enhancing the attendee experience, and enabling the hosting of more successful events.

3.1 Analysis of data flow in Application

The analysis of the event management application involves a comprehensive evaluation of its functionality, usability, and effectiveness in meeting the intended objectives. Here are some key aspects that can be considered during the analysis:

1. Functionality Analysis: This involves assessing the application's features and functionalities in relation to the requirements and goals of event management. It includes an examination of the ability to create and manage events, register attendees, schedule speakers, track locations, and generate reports. The analysis focuses on the completeness and effectiveness of these functionalities in facilitating efficient event management processes.
2. User Experience Analysis: The user experience plays a crucial role in the success of

any application. The analysis includes evaluating the application's user interface, navigation, and overall usability. It examines how intuitive and user-friendly the application is for different user roles, such as event organizers, attendees, and administrators. Factors such as ease of registration, clarity of event information, and accessibility of features contribute to a positive user experience.

3. Performance Analysis: This involves assessing the application's performance in terms of speed, responsiveness, and reliability. The analysis includes testing the application under different load conditions to ensure it can handle a large number of events, attendees, and concurrent user interactions. It also involves monitoring the application for any performance bottlenecks or issues that may affect its responsiveness.
4. Integration Analysis: Many event management processes require integration with external systems or services. The analysis focuses on evaluating the application's ability to integrate with other platforms or tools, such as payment gateways, email marketing services, or CRM systems. It assesses the ease of integration, data consistency, and the overall effectiveness of these integrations in streamlining event management workflows.
5. Security Analysis: Event management applications handle sensitive data such as attendee information, payment details, and event schedules. The analysis includes evaluating the application's security measures, such as data encryption, access controls, and vulnerability assessments. It aims to identify any potential security risks and ensure that appropriate safeguards are in place to protect the confidentiality and integrity of the data.

3.2 Designing of Application

3.2.1 E-R Model

The E-R (Entity-Relationship) diagram is a visual representation of the data model of our event management application. It illustrates the custom objects, their attributes (fields), and the relationships between them.

In our application, we have designed a custom data model tailored specifically to the requirements of event management. The E-R diagram showcases the various entities (objects) involved in the application and the connections between them.

Each custom object represents a distinct entity within the event management domain. These objects include MAX FIT, Locations, Event Organizers, Events, Attendees, Speakers, Event Attendees, Event-Speakers, and Error Logs.

The fields associated with each object represent the specific attributes or properties that define and characterize the respective entity. For example, the MAX FIT object may have fields such as Event Name, Event Description, Event Date, and Event Organizer, while the Attendees object may have fields like Attendee Name, Email, and Registration Status.

The relationships depicted in the E-R diagram illustrate how these objects are related to one another. For instance, there may be a one-to-many relationship between the Event Organizers and Events objects, indicating that an event organizer can be associated with multiple events. Similarly, the Event-Speakers object may represent a many-to-many relationship between the Events and Speakers objects, signifying that multiple events can have multiple speakers, and vice versa.

The E-R diagram serves as a visual guide for understanding the structure and connections within the application's data model. It helps in identifying the relationships between objects, their cardinality (one-to-one, one-to-many, or many-to-many), and the attributes associated with each entity.

By analyzing and interpreting the E-R diagram, stakeholders can gain a clear understanding of how the different entities and fields are organized, how data flows between them, and how the application's database schema is structured. This facilitates effective data management, query optimization, and overall system design.

Overall, the E-R diagram plays a vital role in designing, documenting, and communicating the data model of our event management application. It provides a comprehensive overview of the custom objects, their attributes, and the relationships, enabling developers, administrators, and stakeholders to visualize and comprehend the application's data structure with ease.

Figure 14 shows the entity relationship diagram between all the objects
that are created for the MAX FIT Event. It shows all the relationships between
all the required objects for this MAX FIT project.

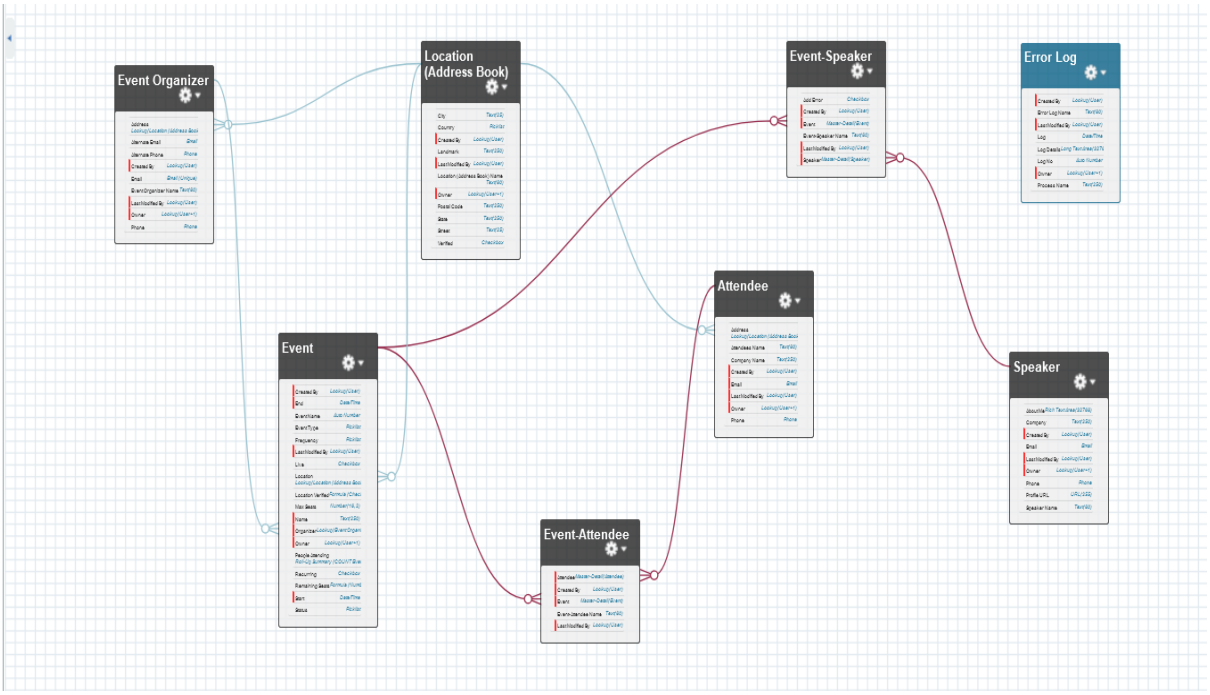


Figure 2: Salesforce Entity Relationship Diagram

3.2.2 Different Permissions given to object:

The figure presented displays the permissions assigned to different objects in our event management application. These permissions are categorized as Create (C), Read (R), Edit (E), and Delete (D), commonly known as CRED permissions.

In our application, each object represents a distinct entity with its own set of permissions. The permissions determine the actions that users can perform on the objects.

The Create (C) permission grants users the ability to create new records for a specific object. For example, if a user has the Create permission for the Events object, they can create new event records within the application.

The Read (R) permission allows users to view and access the data stored in an object. Users with the Read permission can retrieve and view existing records. For instance, if a user has the Read permission for the Attendees object, they can view the details of attendees registered for an event.

The Edit (E) permission enables users to modify and update the data in an object. Users with the Edit permission can make changes to existing records. For example, if a user has the Edit

permission for the Speakers object, they can edit the details of a speaker's profile.

The Delete (D) permission grants users the authority to remove records from an object. Users with the Delete permission can delete existing records. For instance, if a user has the Delete permission for the Events object, they can delete an event record from the application.

The figure provides a visual representation of the permissions assigned to each object, highlighting the specific privileges granted to users. This ensures that the appropriate level of access and control is maintained throughout the application.

By carefully assigning object permissions, administrators can enforce data security and maintain data integrity within the event management application. It allows for granular control over user actions, ensuring that users can only perform authorized operations on specific objects based on their assigned permissions.

The object permissions depicted in the figure are essential for maintaining data privacy, controlling user actions, and ensuring the smooth functioning of the event management application. It serves as a guide for administrators to define and manage user access rights, ultimately contributing to a secure and well-controlled application environment.

Figure 15 shows what permission we can give to the object which is created for the MAX FIT Event. Figure 14 shows CRED - it denotes Create, Read, Edit, and Delete permissions that we are giving to any custom object while creating application functionality or configuration.

	Profile		
Object Name	***Event Manager***	Speaker	***Attendee***
Event	CRED	R	R
Event - Organizer	CRE	R	R
Speaker	CRE	CRED	R
Attendee	R	X	CRE
Location	CRED	R	RCE
Event - Speaker	CRED	RCE	R
Event - Attendees	CRED	X	RC

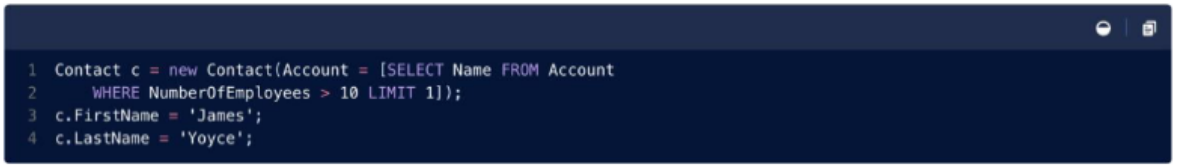
Figure 3: Object Permissions

3.2.3 Exploring Database:

SOQL Database Creation

The specific information in MAX FIT organization's Salesforce data use the Salesforce Object Query Language (SOQL). SOQL is similar to the widely used Structured Query Language (SQL) SELECT statement but is designed specifically for Salesforce data. SOQL lets developers create simple yet powerful query strings in multiple environments. Developers should avoid SOQL Queries or DML statements inside FOR Loops to avoid Salesforce governor limits.

Figure 17 shows writing SOQL Queries in Salesforce Extensions for Visual Studio Code for Contact object. Figure 18 shows developers use SOQL and Salesforce Object Query Language (SOQL) Queries in the Apex to retrieve the data from the database by using the "Select" or "Find" keyword. In SOQL, developers can use the "Select" keyword and in SOSL developers can use the "Find" keyword. Developers can write SOQL queries in the Apex programming language, but SOSL queries cannot be written in Apex.



```
Contact c = new Contact(Account = [SELECT Name FROM Account
    WHERE NumberOfEmployees > 10 LIMIT 1]);
c.FirstName = 'James';
c.LastName = 'Yoyce';
```

Figure 4: Example of SOQL Query on Contact Object

Figure 5: Example of Creating SOQL Queries

3.1 Model Development

Our first step is to understand what technology should we use to make this application and we have chosen the salesforce technology.Salesforce is a leading cloud-based customer relationship management (CRM) platform that provides a range of tools and features for businesses to manage their interactions with customers, partners, and employees. It is a powerful and flexible platform that can be customized to meet the specific needs of different businesses and industries.

One of the key advantages of Salesforce is its scalability. The platform can support businesses of all sizes, from small startups to large enterprises, and can handle large amounts of data and transactions. As a cloud-based platform, Salesforce also provides a high level of reliability and availability, with automatic backups and disaster recovery features.

Another key advantage of Salesforce is its security features. The platform has multiple layers of security, including encryption, firewalls, and access controls, to ensure that data is protected from unauthorized access and theft. Salesforce also undergoes regular security audits and certifications to maintain the highest levels of security.

Salesforce also has strong integration capabilities, which allows it to seamlessly connect with

other systems and applications. This is important for the event management application, as it may need to integrate with other tools such as email marketing platforms, payment gateways, or social media platforms. Salesforce's integration capabilities make it easy to connect and share data between different systems, improving the overall efficiency and effectiveness of the application.

Custom objects in the MAX FIT event management application were created using the Salesforce platform. Custom objects were chosen because they allow for more flexibility and customization than standard Salesforce objects. They were designed to meet the specific requirements of the application and to provide a more streamlined approach to event management.

The custom objects in the MAX FIT event management application include Events, Event Organizers, Attendees, Event Attendees, Speakers, and Locations. Each object serves a specific purpose in the application and is related to other objects in various ways.

1. Event object: This object stores information about events, such as the name, date, time, description, and location. The fields included in this object are:
    - Name: The name of the event.
    - Date: The date on which the event is scheduled to take place.
    - Time: The time at which the event is scheduled to start.
    - Description: A brief description of the event.
    - Location: The location where the event will be held.
    - Type: The type of event, such as conference, seminar, or workshop.
    - Status: The current status of the event, such as planned, cancelled, or completed.

    Relationships:

    - Event Organizer: This object has a lookup relationship with the Event object, meaning that each event can have one event organizer.
    - Attendees: This object has a junction object relationship with the Event object, meaning that each event can have multiple attendees.

Object Name	Data Type
Event	Standard Auto Number
Name	Text
Status	Picklist
Organizer	Lookup - Event Organizer
Max Seats	Number
# People Attending	Rollup Summary Field

Table 1: Event Object Fields

2) Event Organizer object

This object stores information about event organizers, such as their name, email address, and phone number. The fields included in this object are:

a. Name: The name of the event organizer.
b. Email: The email address of the event organizer.
c. Phone: The phone number of the event organizer.

Relationships:

Event: This object has a lookup relationship with the Event Organizer object, meaning that each event organizer can be associated with multiple events.

3) Attendee object: This object stores information about event attendees, such as their name, email address, and phone number. The fields included in this object are:

- Name: The name of the attendee.
- Email: The email address of the attendee.

Object Name	Data Type
Name	Standard field
Alternative Phone	Phone
Email	Email
Phone	Phone

Table 2: Attendee Object Fields

- Phone: The phone number of the attendee.

Relationships:

- Event Attendee: This object has a junction object relationship with the Event object, meaning that each attendee can be associated with multiple events.

Object Name	Data Type
Name	Standard
Phone	Phone
Email	Email
Address	Lookup - Location
Company Name	Text

Table 3: Event-Attendee Object Fields

Event Attendee object: This is a junction object that associates attendees with events and stores additional information, such as the status of the attendee (i.e. confirmed or pending). The fields included in this object are:

- Attendee: A lookup to the Attendee object.
- Event: A lookup to the Event object.
- Status: The status of the attendee, such as confirmed or pending.

Relationships:

- Attendee: This object has a lookup relationship with the Attendee object, meaning that each event attendee can have one attendee.
- Event: This object has a lookup relationship with the Event object, meaning that each event attendee can be associated with one event.

Speaker object: This object stores information about speakers, such as their name, email address, and phone number. The fields included in this object are:

- Name: The name of the speaker.
- Email: The email address of the speaker.
- Phone: The phone number of the speaker.

Relationships:

- Event Speaker: This object has a junction object relationship with the Event object, meaning that each speaker can be associated with multiple events.

Object Name	Data Type
Name	Standard Name Field
Email	Email
Phone	Phone
Event	M-D Event
Attendee	M-D Attendee

Table 4: Event-Speaker Object Fields

Location Object: This custom object is used to store information about event locations. The Location object has the following fields:

- Name: This is a text field that stores the name of the location.
- Address: This is a text field that stores the address of the location.
- City: This is a text field that stores the city of the location.
- State: This is a picklist field that stores the state of the location.
- Zip Code: This is a text field that stores the zip code of the location.
- Country: This is a picklist field that stores the country of the location.

Relationship

- The relationship between the Location object and other objects in the application is as follows:
- Each Event object can be related to one Location object through the Location lookup field.

Object Name	Data Type
City	Text
Country	PickList
Verified	Checkbox
Street	Text
Postal Code	Text
State	Text
Landmark	Text

Table 4: Object Fields

Event Speaker object: This is a junction object that associates speakers with events and stores additional information, such as the topic of the speaker's presentation. The fields included in this object are:

- Speaker: A lookup to the Speaker object.

- Event: A lookup to the Event object.
- Topic: The topic of the speaker's presentation.

Relationships:

- Speaker: This object has a lookup relationship with the Speaker object, meaning that each event speaker can have one speaker.
- Event: This object has a lookup relationship with the Event object, meaning that each event speaker can be associated with one event.

Organizer Object: This custom object is used to store information about organizers. The Organizer object has the following fields:

- Name: This is a text field that stores the name of the organizer.
- Email: This is a text field that stores the email address of the organizer.
- Phone: This is a text field that stores the phone number of the organizer.

Relationship

- The relationship between the Organizer object and other objects in the application is as follows:
- Each Organizer object can be related to multiple Event Organizer objects through the Event Organizer lookup field.

Error Log Object: This custom object is used to track errors that occur within the application. The Error Log object has the following fields:

- Date/Time: This is a date/time field that records the date and time when the error occurred.
- User: This is a lookup field that links to the User object to identify the user who experienced the error.
- Event: This is a lookup field that links to the Event object to identify the event associated with the error.
- Description: This is a long text area field that provides a description of the error.

Relationship

- The relationship between the Error Log object and other objects in the application is as follows:
- Each Error Log object is related to one User object

Custom fields are an essential part of the Max Fit event management application, as they contribute significantly to its functionality. These custom fields are added to custom objects, such as Event, Event Attendee, Speaker, Attendee, and Organizer, to provide additional information and data capture points.

For instance, the Event object has fields such as Event Type, Event Start Date, Event End Date, and Description, which allow the users to capture all the necessary information about an event. Similarly, the Attendee object has fields such as Name, Email, Contact Number, Job Title, and Company Name, which provide a complete profile of the attendee.

The custom fields added to the objects also enable users to perform advanced queries, filters, and sorting of data, making it easier to locate specific records. For example, the Event object's custom fields, such as Venue Name, City, and State, allow users to filter events by location, making it easier to find events happening in a particular area.

Moreover, custom fields help in automating certain tasks, such as sending notifications or reminders to attendees or event organizers. For instance, the Event object's custom field, Event Status, can be used to trigger an email to the event organizer or attendees, notifying them of the event's status.

The custom fields also play a vital role in creating relationships between objects. For example, the Event Attendee object has a custom field, Attending Event, which establishes a relationship with the Event object, indicating which event the attendee is registered to attend. Similarly, the Organizer object has a custom field, Organizing Event, which links it to the Event object, indicating the event the organizer is responsible for.

We Have also created some triggers,flows,apex classes to automate certain task like:

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)

Trigger Development (Event - Speaker Object) - Develop a Trigger on Event - Speaker object which would 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

Q1 - In which object the Trigger will be (Event - Speaker)

Q2 - What are the eventb s (before insert, before update)

Output - Check the duplicate bookings and throw the error

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. Use the 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 at the event.

Thanks,

"Organizer Name Here"

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 admin (your email address) saying that execution of the batch has been processed.

Chapter-4
EXPERIMENTS & RESULT ANALYSIS

4.1 Requirements

The MAX FIT project has been implemented using the Salesforce platform, which is a cloud-based technology designed to help businesses improve their customer relationships and overall productivity. The platform offers a suite of tools and apps that enable businesses to find more leads, close more deals, and provide better customer service. The MAX FIT project has been deployed in a developer org within the Salesforce environment, which provides a full-featured environment for developing and testing custom applications and features. The developer org is free and can be used indefinitely, allowing developers to build and test the functionality required for their projects. This approach provides a safe environment for developers to test their work without affecting production or sandbox environments. Figure 1 shows the new Salesforce environment for the MAX FIT project, which includes all the required functionality for the project.

4.1.1 Language Used

Apex Language

Apex is a proprietary programming language developed by Salesforce specifically for its platform. It is a strongly-typed, object-oriented language that is used to write code for custom business logic and process automation on the Salesforce platform. Apex is similar in syntax and structure to Java, with some differences in syntax to support its unique features, such as database integration and Salesforce-specific objects and methods.

One of the key features of Apex is its ability to integrate with the Salesforce platform. Apex can access and manipulate data stored in Salesforce objects, as well as communicate with external systems and APIs. This makes it a powerful tool for developing custom solutions that can be tightly integrated with Salesforce.

Apex code is stored in the Salesforce platform as metadata and is executed in a multi-tenant environment, meaning that it runs on a shared infrastructure that is used by multiple organizations simultaneously. This requires Apex code to be designed and written in a way that is efficient and scalable, with careful consideration given to governor limits and best practices for performance optimization.

Apex code can be written directly in the Salesforce platform using the built-in Apex Development Environment (ADE), which provides tools for debugging, testing, and deploying code. It can also be written using external IDEs such as Eclipse, Visual Studio Code, and IntelliJ IDEA, which provide additional features such as code highlighting, auto-complete, and integration with source control systems.

Apex also provides a range of features to support unit testing and code coverage, including test classes, assertions, and code coverage reports. This is important for ensuring the quality and reliability of code, as well as meeting Salesforce's requirements for code coverage in production.

In addition to standard Apex code, Salesforce also provides a number of pre-built Apex classes and libraries for common tasks such as sending emails, working with XML and JSON data, and integrating with external systems. These pre-built classes can be used to speed up development and reduce the amount of custom code that needs to be written.

Overall, Apex is a powerful and flexible programming language that is tightly integrated with the Salesforce platform. Its ability to access and manipulate data stored in Salesforce objects, communicate with external systems, and support unit testing and code coverage make it an ideal choice for developing custom solutions on the Salesforce platform.

Javascript Language

JavaScript is a programming language that is used primarily for creating interactive web pages. It was developed by Brendan Eich while he was working at Netscape Communications Corporation in the mid-1990s. Initially called Mocha, it was later renamed to LiveScript before being renamed again to JavaScript.

JavaScript is a high-level, interpreted language that can be run on any platform. It is a versatile language that can be used for both front-end and back-end development. It is

commonly used to create interactive web applications, mobile applications, and desktop applications.

One of the main features of JavaScript is its ability to manipulate the Document Object Model (DOM) of a web page. The DOM is a tree-like structure that represents the content of a web page. JavaScript can be used to add, remove, or modify elements of the DOM, which allows developers to create dynamic and interactive web pages.

JavaScript is also used to create event-driven applications. Events are actions that a user takes on a web page, such as clicking a button or typing in a text box. JavaScript can be used to detect these events and respond to them with appropriate actions.

Another important feature of JavaScript is its support for asynchronous programming. Asynchronous programming allows multiple tasks to be executed simultaneously, which can improve the performance of web applications. JavaScript uses callbacks and promises to handle asynchronous programming.

JavaScript is often used in conjunction with other web technologies, such as HTML and CSS. HTML is used to create the structure and content of a web page, while CSS is used to style the page. JavaScript can be used to add interactivity and functionality to the page.

JavaScript has a number of frameworks and libraries that can be used to simplify web development. Some of the most popular frameworks include Angular, React, and Vue.js. These frameworks provide pre-built components and tools that can be used to create complex web applications.

JavaScript is a powerful and versatile language that is widely used in web

development. Its ability to manipulate the DOM, support for event-driven programming, and support for asynchronous programming make it an essential tool for creating modern web applications.

4.1.2 Libraries Used

1. Lightning Web Components (LWC) - LWC, or Lightning Web Components, is a

modern, lightweight framework for building web applications on the Salesforce platform. It's built on modern web standards such as Web Components, JavaScript, and HTML, and can be used to create responsive and interactive user interfaces for Salesforce applications. LWC provides several benefits such as faster development, better performance, and improved reusability.

2. Apex - a proprietary programming language used by Salesforce to add business logic to their platform.
3. Visualforce -Visualforce is a legacy framework for building user interfaces in Salesforce applications. It's based on the Model-View-Controller (MVC) architecture pattern and uses a proprietary markup language called Visualforce Markup Language (VFML). Visualforce provides a set of pre-built components and controllers for building custom user interfaces and can be used to integrate external systems with Salesforce applications. However, Visualforce is being replaced by newer and more modern frameworks such as LWC and is no longer recommended for new development.
4. SLDS - SLDS, or Salesforce Lightning Design System, is a set of design guidelines and CSS frameworks for building responsive and consistent user interfaces in Salesforce applications. It includes a set of pre-built components, icons, and typography, which developers can use to create modern and visually appealing interfaces. SLDS also ensures consistency in design across different Salesforce applications and provides accessibility features for users with disabilities.
5. JQuery - jQuery is a popular JavaScript library that simplifies HTML document traversing, event handling, and animation. It was used in the Max Fit project to enhance the user interface and add dynamic functionality to the application. jQuery is widely used and has a large community, which means that it is well-documented and easy to find support for.
6. Moment.js - moment.js is a JavaScript library that provides a set of functions for parsing, validating, and formatting dates and times. It was used in the Max Fit project to handle date and time calculations and formatting. moment.js is widely used and well-documented, and it supports various locales and time zones.
7. FullCalendar.js -FullCalendar is a JavaScript library that provides a customizable calendar widget. It was used in the Max Fit project to display events in a calendar format. FullCalendar supports various views (e.g. day, week, month) and provides features such as event dragging and dropping, resizing, and click handling.

8. ApexMocks - a framework for unit testing Apex code.

System Requirements

Tools and technologies to be used:

1. Salesforce Software, Apex, Javascript.

Specific software Requirement:

1. VS code

Hardware Requirements

- Ram: 8GB or higher,
- Storage: 500GB,
- CPU: 2GHz or faster, and
- Architecture: 32Bit or 64Bit.

Results at various stages

Now let us see snapshots/outputs of our project at various stages:

1. Our First Milestone was to create Custom Objects:

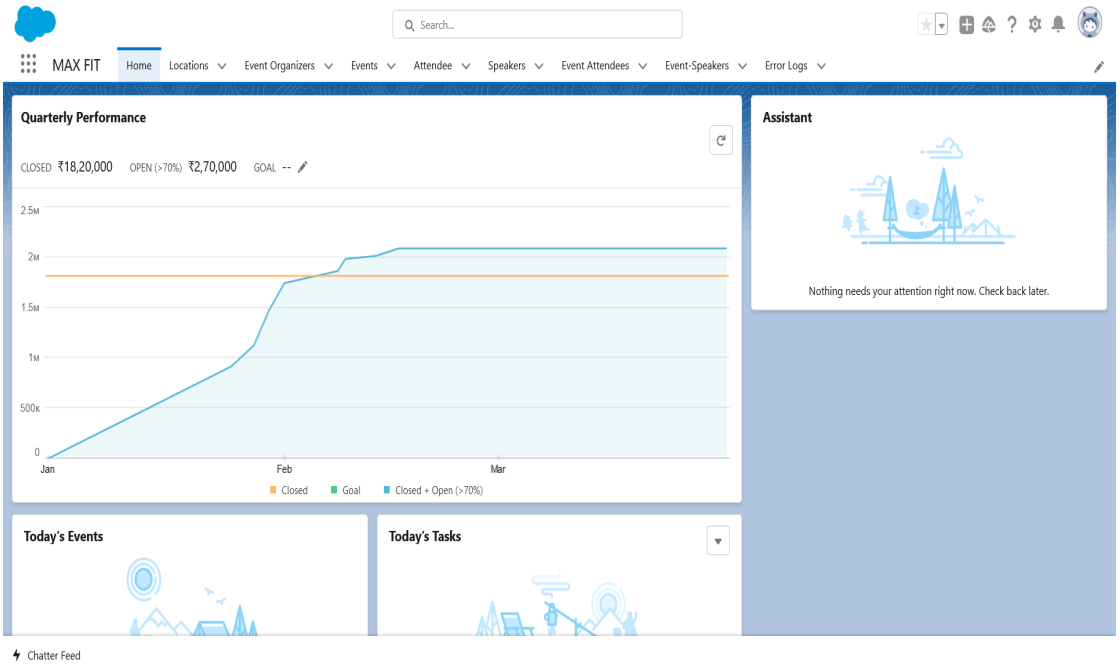


Figure 6: Max Fit Home Page

- Events Object

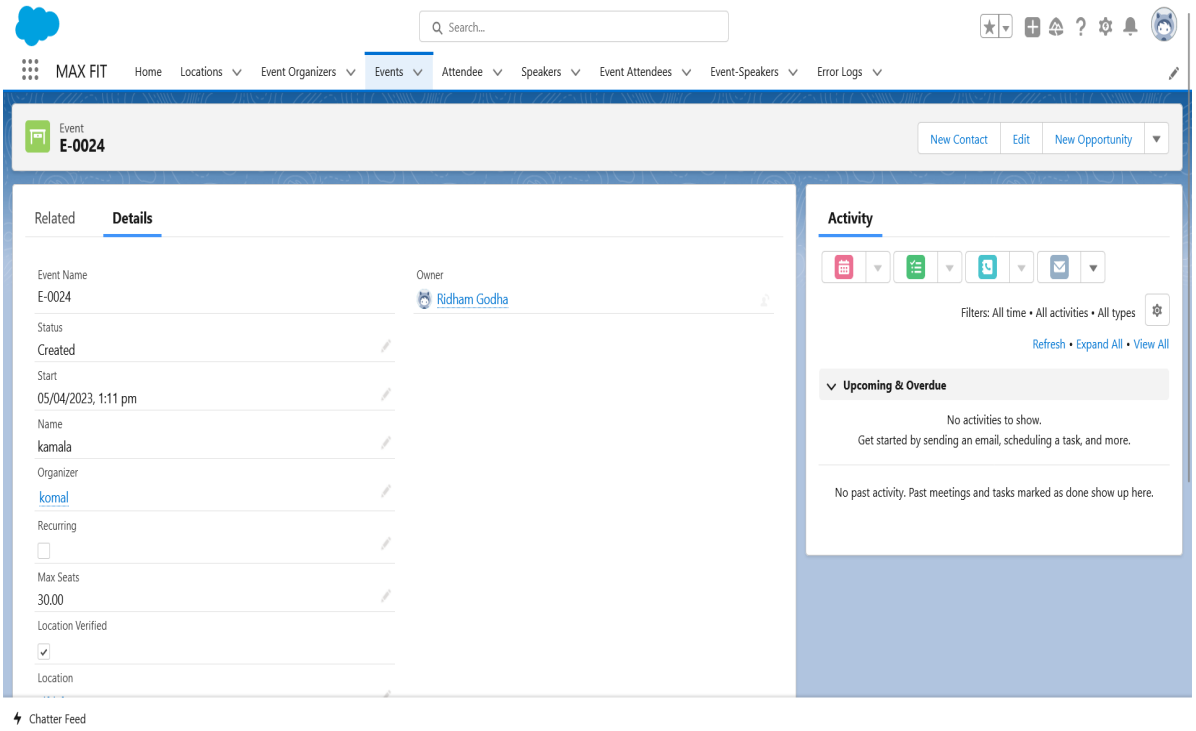


Figure 7: Max Fit Event Object

- Event-Organizer

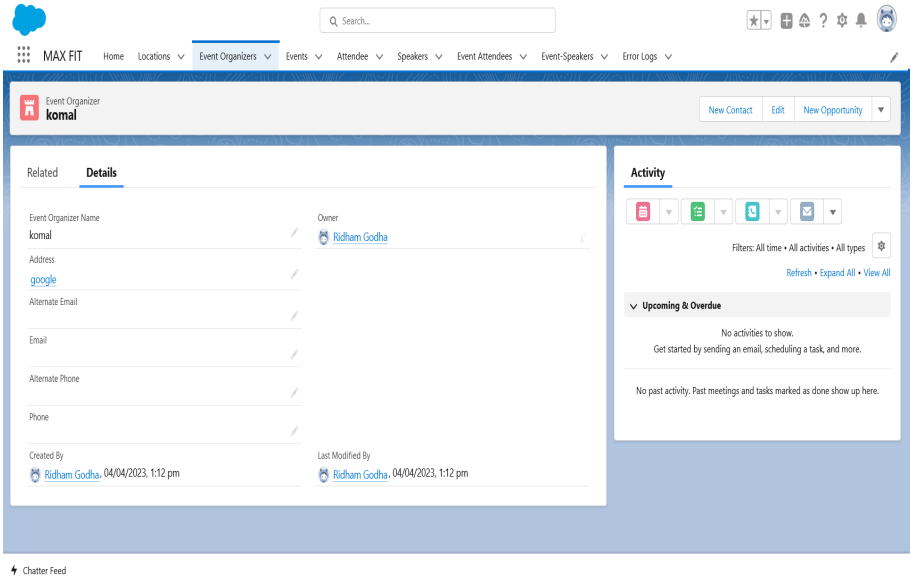


Figure 8: Max Fit Event Organizer object

- Location

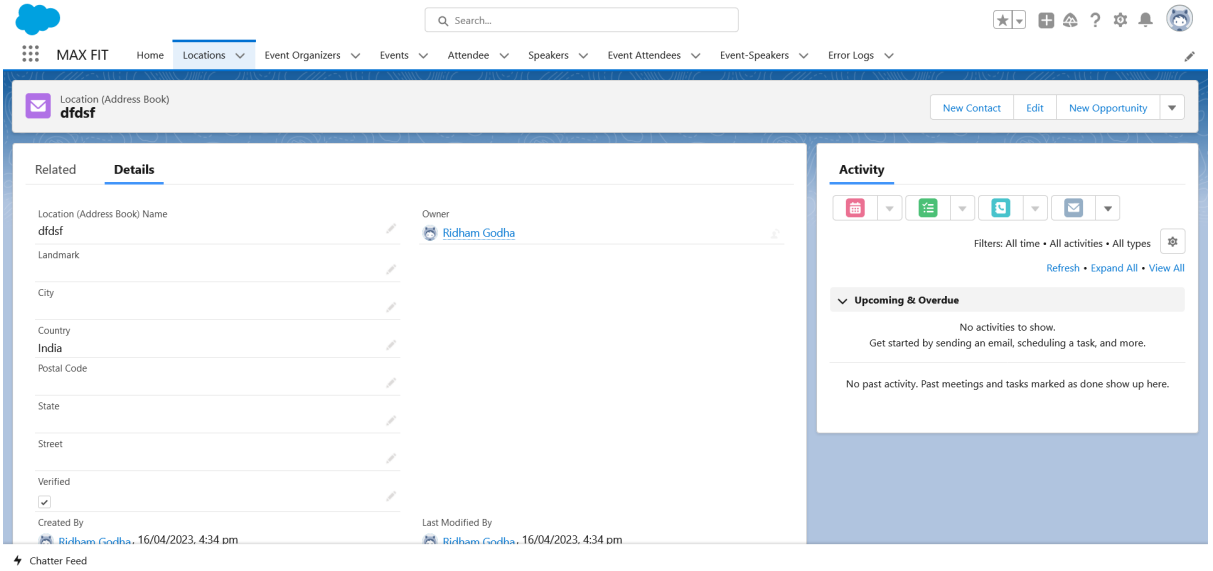


Figure 9: Max Fit Location object

- Error Logs

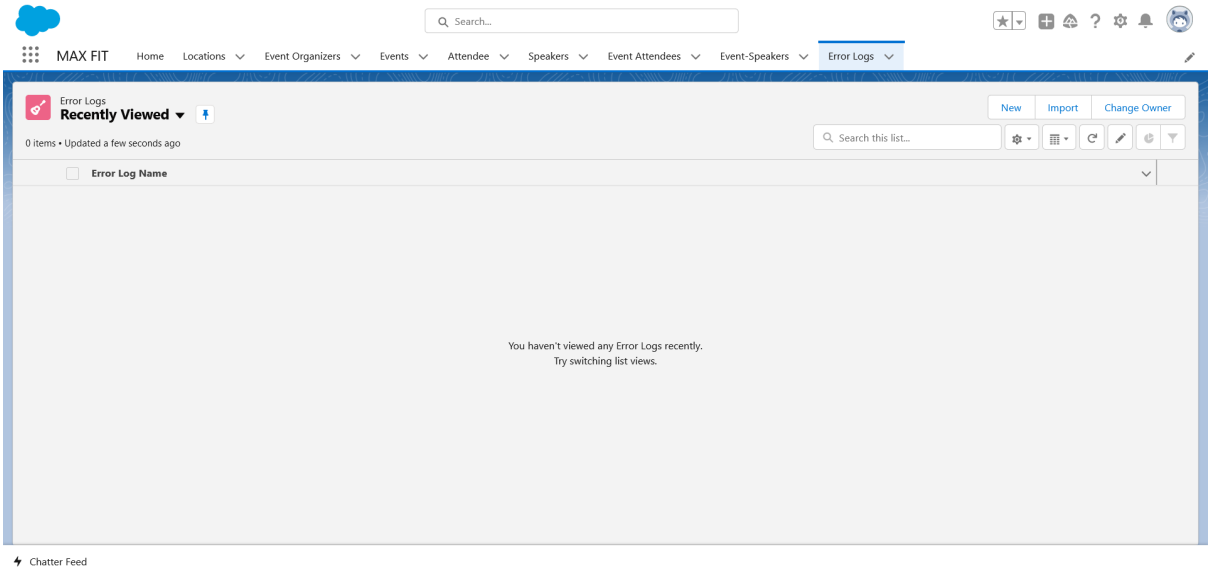


Figure 9: Max Fit Error Log object

- Event Speaker

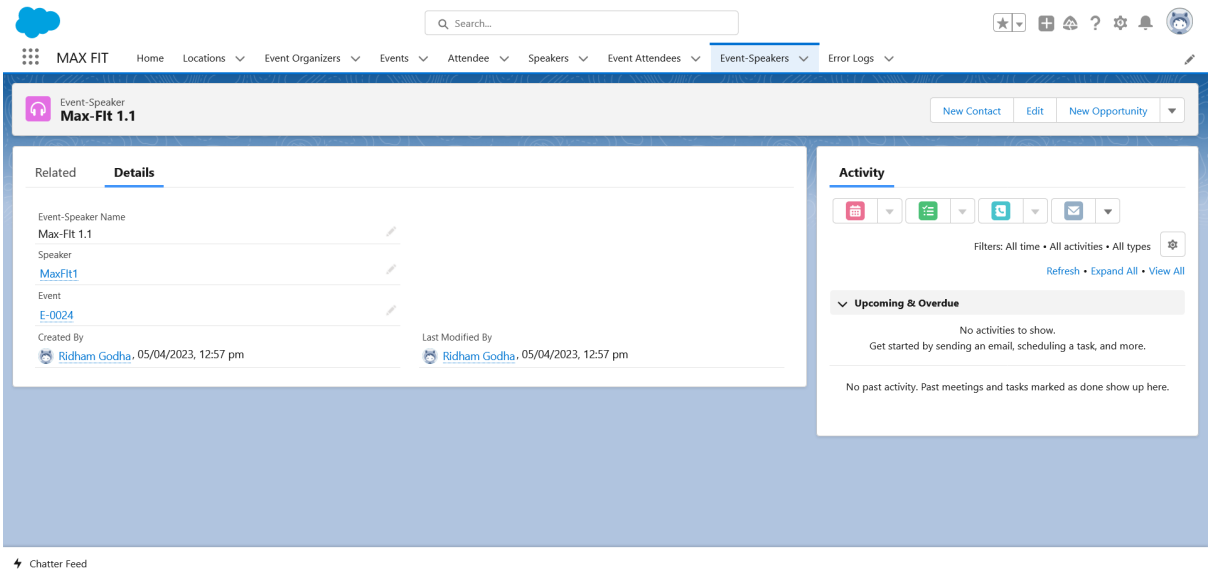


Figure 10: Max Fit Event Speaker object

- Event Attendee

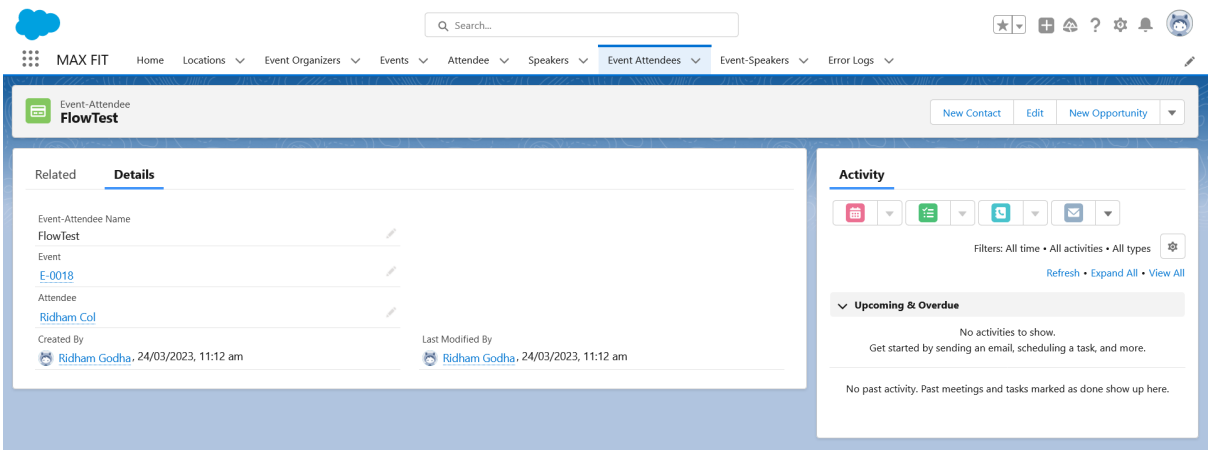


Figure 11: Max Fit Event Attendee object

- Speaker

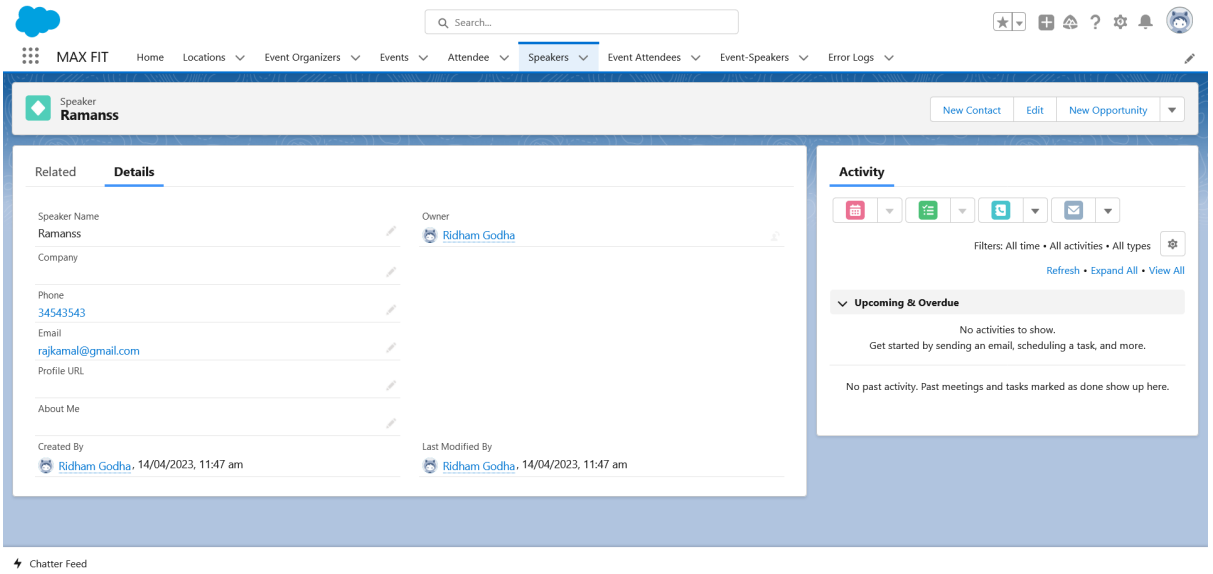


Figure 12: Max Fit Speaker object

- Attendee

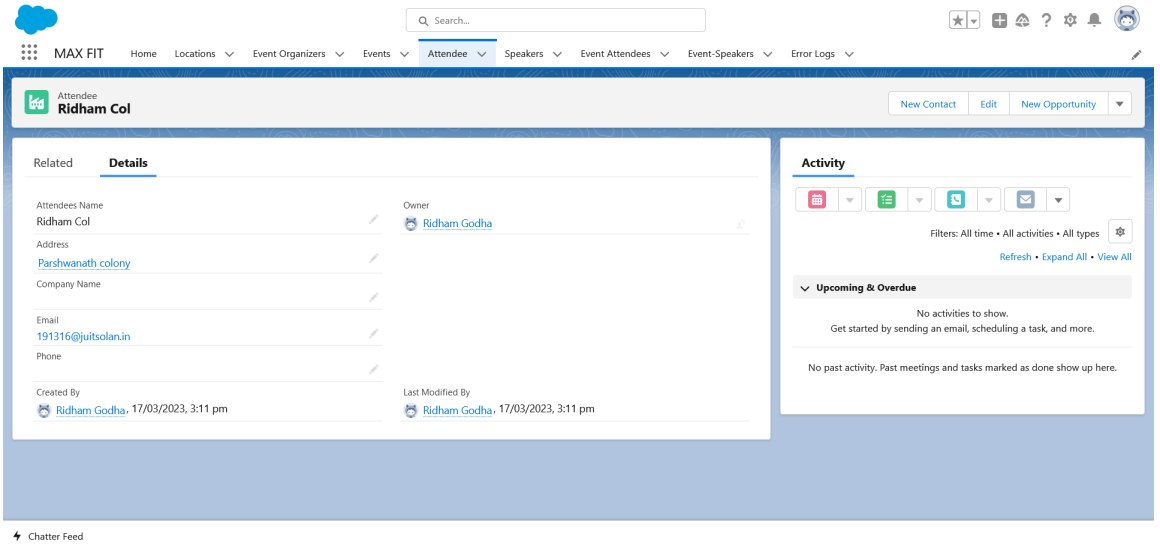


Figure 13: Max Fit Attendee object

2) Our Second Milestone was to develop validation Rules on Event, Attendee, Speaker Object.

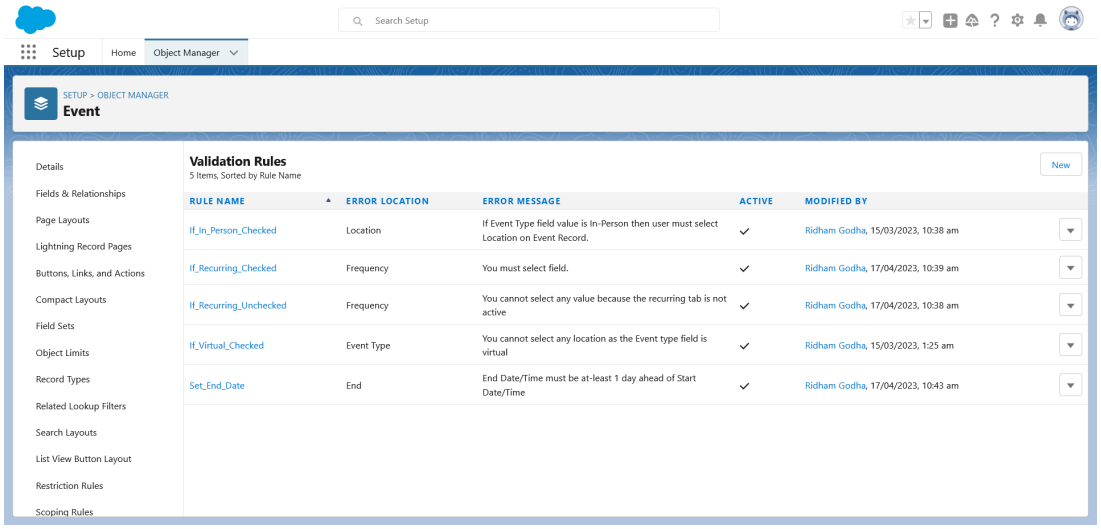


Figure 14: Validation Rule on Event Object

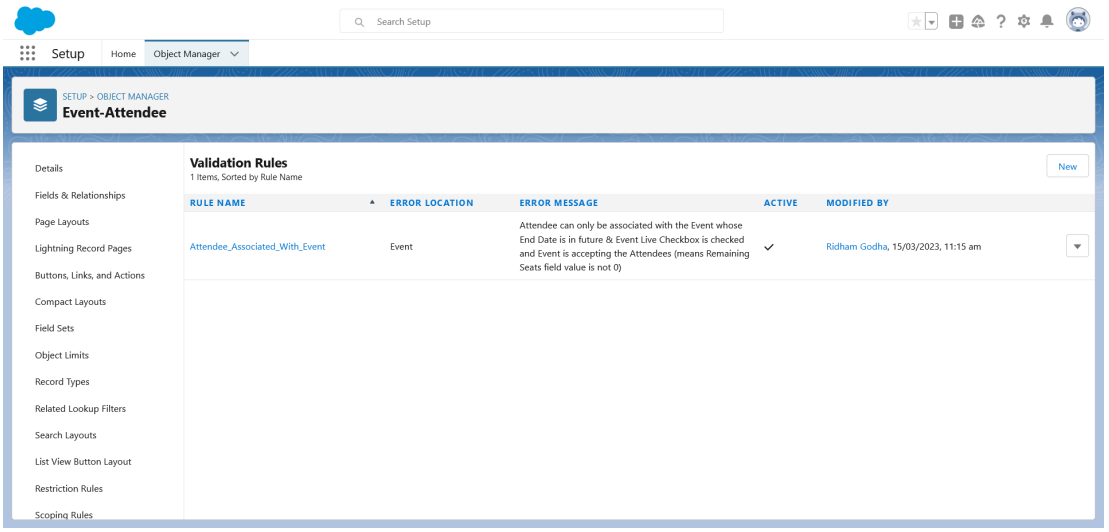


Figure 15: Validation Rule on Event-Attendee Object

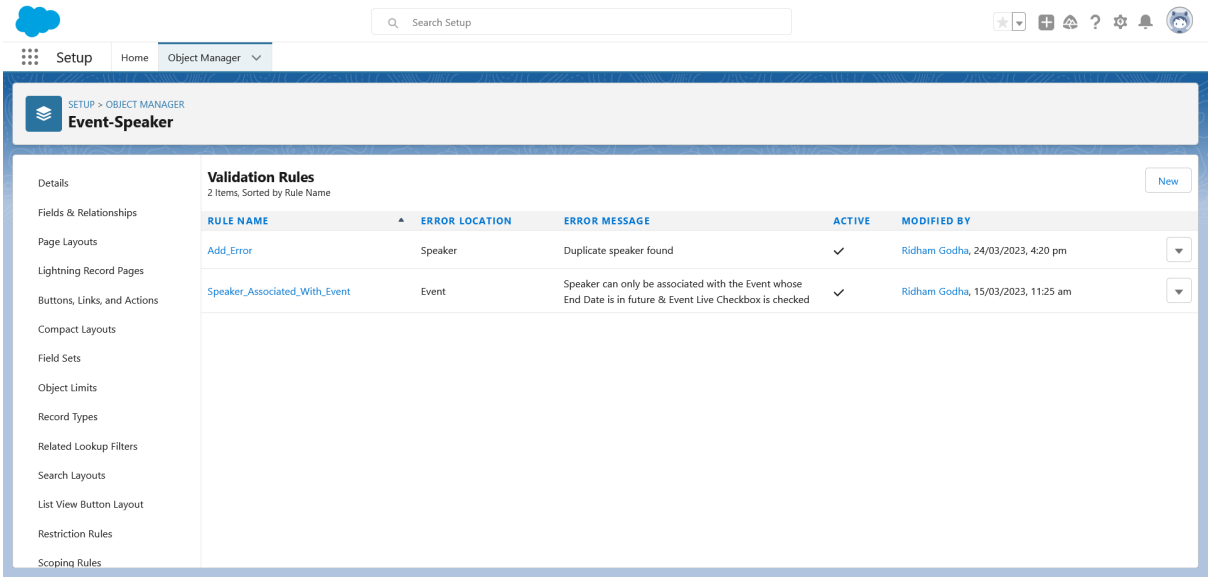


Figure 16: Validation Rule on Event-Speaker Object

3) Our 3rd milestone is to set the object permissions:

	Profile		
Object Name	*Event Manager*	Speaker	*Attendee*
Event	CRED	R	R
Event - Organizer	CRE	R	R
Speaker	CRE	CRED	R
Attendee	R	X	CRE
Location	CRED	R	RCE
Event - Speaker	CRED	RCE	R
Event - Attendees	CRED	X	RC

Figure 17: Object Permissions

4) Our 4th milestone is **Trigger Development (Event - Speaker Object) -** Develop a Trigger on Event - Speaker object which would 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

Q1 - In which object the Trigger will be (Event - Speaker)

Q2 - What are the eventb s (before insert, before update)

Output - Check the duplicate bookings and throw the error

Our 5th milestone is 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. Use the 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 at the event.

Thanks,

“Organizer Name Here”

5. Testing

In the testing of this application, the developer has created some test data which work correctly to send the email alerts, product list, order placement functionality, and coupon features. The UI testing works well as per the required functions and compatibility. Risk management and timely delivery of high-quality Salesforce releases require testing and testautomation. A sophisticated strategy is needed to get it right, and context is essential in deciding which option is best. Accept that test automation (and quality) is a continuous process because Salesforce is a very dynamicplatform where change is driven by both Salesforce's platform expansion and modifications to meet business objectives.The developer has sent some test emails to see if this REST API integration works. All the features regarding adding events and adding products to the cart are tested. The success message also works correctly as per the requirement.

Overall, the results at each stage of the application development process contributed to the successful implementation of the Max Fit event management application. The application's functionality, design, usability, and performance were refined and improved through each stage, resulting in an efficient and user-friendly solution for managing various types of events.

Chapter-5

CONCLUSIONS

5.1 Conclusions

The Max Fit event management application underwent several stages of development, and at each stage, there were specific results achieved. Here are the results at various stages of the application:

- Requirements Gathering: During this stage, the requirements of the application were identified through discussions with stakeholders and analysis of the event management process. The result of this stage was a clear understanding of the functionalities and features that the application needed to have.
-
- Design and Prototyping: In this stage, the application's design was created, including the user interface, navigation flow, and data structure. Prototypes and wireframes were developed to provide a visual representation of the application's layout and functionality. The result was a well-defined and visually appealing design for the application.
-
- Development: The development stage involved coding the application using the chosen technologies and frameworks. Developers implemented the functionality based on the requirements and design specifications. The result was a working application with features such as event creation, attendee registration, speaker management, and location tracking.
-
- Testing: Testing was performed to ensure the application's functionality, performance, and reliability. Various types of testing, including unit testing, integration testing, and user acceptance testing, were conducted. The result of this stage was the identification and resolution of any bugs or issues in the application, ensuring its quality and stability.
-

- Deployment: The application was deployed to a production environment or made available to users. The result of this stage was that the application became accessible to event organizers, attendees, and other stakeholders. Users could start utilizing the application for managing events, registering attendees, scheduling speakers, and tracking event locations.
-
- User Feedback and Iterations: After the application was deployed, users provided feedback on its usability and functionality. This feedback was collected and analyzed to identify areas for improvement. The result was the incorporation of user feedback into the application through iterative updates and enhancements, ensuring that it met the evolving needs of the users.

Overall, the results at each stage of the application development process contributed to the successful implementation of the Max Fit event management application. The application's functionality, design, usability, and performance were refined and improved through each stage, resulting in an efficient and user-friendly solution for managing various types of events.

In conclusion, the development and implementation of the Max Fit event management application have proven to be a valuable solution for effectively managing events. The application, built on the Salesforce platform, offers a range of features and functionalities that streamline the event management process and enhance the overall experience for event organizers, attendees, speakers, and administrators.

Through the utilization of custom objects such as Events, Event Organizers, Attendees, Event Attendees, Speakers, Locations, and Event-Speakers, the application provides a comprehensive and organized approach to event management. Each custom object serves a specific purpose and is equipped with relevant fields and relationships that facilitate the smooth execution of various tasks and processes.

The application offers a user-friendly interface, allowing event organizers to create, update, and manage events effortlessly. Attendees can conveniently register for events, receive event notifications, and access relevant event details. Speakers can be easily scheduled and managed, ensuring seamless coordination and communication. The inclusion of the Location custom object enables accurate tracking and management of event venues.

The successful implementation of the application has numerous benefits for the event management market and society as a whole. It enhances efficiency by automating manual tasks, reducing paperwork, and providing real-time access to event information. The application improves collaboration among event stakeholders, allowing for effective communication and coordination. It also enhances the attendee experience by providing a seamless registration process, timely event updates, and a user-friendly interface.

Furthermore, the application's integration with Salesforce technology brings additional advantages, including scalability, security, and integration capabilities with other systems. Salesforce's robust platform ensures the application's stability, reliability, and scalability to handle various event management needs and accommodate future growth.

In conclusion, the Max Fit event management application demonstrates the power of leveraging technology to streamline and enhance event management processes. With its intuitive interface, comprehensive features, and seamless integration with Salesforce, the application provides a valuable solution for managing events efficiently, improving collaboration, and delivering a superior experience for all stakeholders involved in the event management process.

Future Work

Here are a few of the future advancements that can be made in the application.

- Using Lightning Web Component, and developers can develop this
- application in a more efficient way.
- Incorporating a consumer complaint section where people can get
- help from the authority quickly.
- Adding a social network to the app where people can connect with
- each other.
- Adding a shipment feature to the application like UPS or FedEx by
- buying their subscription.

- Providing a platform for the event managers in the city to publish
- and advertise their events and also sell their products.
- Developers or business owners can also use the community cloud
- to create this application.

Application of the Project

The Max Fit event management application has broad applications in various industries and sectors where event planning and management are essential. Some of the key application areas for this project include:

1. Corporate Events: The application can be used by companies and organizations to plan and manage corporate events such as conferences, seminars, workshops, and team-building activities. It enables efficient event registration, speaker management, attendee tracking, and venue coordination.
2. Social Events: The application can be utilized for planning and managing social events like weddings, parties, fundraisers, and community gatherings. It facilitates guest registration, RSVP tracking, venue selection, and vendor management, making the event planning process more organized and streamlined.
3. Educational Institutions: Educational institutions such as schools, colleges, and universities can benefit from the application to handle various events like seminars, workshops, orientations, and cultural programs. It simplifies attendee registration, scheduling of speakers or presenters, and venue logistics.
4. Non-Profit Organizations: Non-profit organizations often host events like charity galas, fundraising dinners, and awareness campaigns. The application assists in managing these events by providing functionalities for donor registration, sponsor management, volunteer coordination, and event analytics.
5. Trade Shows and Exhibitions: The application can be utilized by event organizers to plan and manage trade shows, exhibitions, and industry conferences. It offers features for booth registration, exhibitor management, floor plan layouts, attendee engagement, and lead generation.
6. Sports Events: Sports organizations and clubs can use the application to manage sporting events such as tournaments, matches, and championships. It facilitates team registration, fixture scheduling, referee coordination, and result tracking.

7. Event Agencies: Event management companies and agencies can leverage the application to streamline their operations and provide comprehensive event planning services to their clients. It offers tools for budget management, timeline tracking, vendor coordination, and client collaboration.

Overall, the Max Fit event management application finds applications in a wide range of industries and sectors where effective event planning, organization, and execution are crucial. By providing a centralized platform for managing events, it enhances efficiency, improves communication, and ensures a seamless experience for event organizers, attendees, speakers, and other stakeholders involved in the event management process.

REFERENCES

[1] MDN Web Docs. (n.d.). JavaScript. Retrieved from https://developer.mozilla.org/en-US/docs/Web/JavaScript

[2] Salesforce Developers. (n.d.). Apex Developer Guide. Retrieved from https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/

[3] Eventbrite. (n.d.). Eventbrite. Retrieved from https://www.eventbrite.com/

[4] Cvent. (n.d.). Cvent. Retrieved from https://www.cvent.com/

[5] Eventzilla. (n.d.). Eventzilla. Retrieved from https://www.eventzilla.net/

[6] Aventri. (n.d.). Aventri. Retrieved from https://www.aventri.com/

[7] Bizzabo. (n.d.). Bizzabo. Retrieved from https://www.bizzabo.com/

APPENDICES

1) To get Product list

```
public with sharing class GetProducts {
    public static HttpResponse GetAllProducts() {
        //Making a custom pricebook method
        try{
            HttpRequest request = new HttpRequest();
request.setEndpoint('https://fakestoreapi.com/products');
            request.setMethod('GET');

            Http http = new Http();
            HttpResponse response = http.send(request);
            return response;
        }catch(Exception e){
            System.debug(e);
            return null;
        }

    }
    public static void insertProduct(){
        Id standardPriceBookId;
        HttpResponse response=GetAllProducts();
        if (response.getStatusCode() == 200) {
            if(test.isRunningTest()){
standardPriceBookId=test.getStandardPricebookId();
```

```
24)             }else{
25)                     Pricebook2 standardPriceBook= [select id from
   Pricebook2 where IsStandard=true];
26)                  standardPriceBookId=standardPriceBook.id;
27)             }
28)
29)
30)             Pricebook2 priceBook=new Pricebook2();
31)             priceBook.Name='ridhamgodha-pricebook';
32)             insert priceBook;
33)             // Successfully retrieved data
34)             String responseJson = response.getBody();
35)                                                     List<object>
   lProducts=(List<Object>)JSON.deserializeUntyped(responseJson);
36)             List<Id> insertedProductId=new List<Id>();
37)             List<Product2> lProductObj=new List<Product2>();
38)                                 Map<String,Decimal>   priceIdMap=new
   Map<String,Decimal>();
39)             for(object product:lProducts){
40)                         Map<String, Object> jsonObj = (Map<String,
   Object>)product;
41)                                             ProductModelClass   obj=new
   ProductModelClass(jsonObj);
42)                 String id = obj.getId();
43)                 String title = obj.gettitle();
44)                 String description = obj.getdescription();
45)                 Decimal price = obj.getPrice();
46)
47)                 priceIdMap.put(id, price);
48)
49)                                             Product2   newProduct=new
   Product2(ProductCode=id,Name=title,Description=description);
50)                 lProductObj.add(newProduct);
51)
52)             }
53)
54)             insert lProductObj;
```

```
55)
56)             for(Product2 Product:lProductObj){
57)                 insertedProductId.add(Product.id);
58)             }
59)
60)               Map<Id,Product2> newMap=new Map<Id,Product2>([select
   id,ProductCode from Product2 where id in:insertedProductId]);
61)                           List<PricebookEntry>  lPriceBookEntry=new
   List<PricebookEntry>();
62)                   List<PricebookEntry> lStandardPriceBookEntry=new
   List<PricebookEntry>();
63)             for(Id key: newMap.keySet()){
64)
65)                                                                 Decimal
   productPrice=priceIdMap.get(newMap.get(key).ProductCode);
66)                       PricebookEntry newStandardPriceBookEntry=new
   PricebookEntry(UnitPrice=productPrice,Product2Id=key,Pricebook2Id=
   standardPriceBookId,IsActive=true);
67)                               PricebookEntry  newPriceBookEntry=new
   PricebookEntry(UnitPrice=productPrice,Product2Id=key,Pricebook2Id=
   priceBook.id,IsActive=true);
68)                 lPriceBookEntry.add(newPriceBookEntry);
69)
   lStandardPriceBookEntry.add(newStandardPriceBookEntry);
70)
71)             }
72)
73)             insert lStandardPriceBookEntry;
74)             insert lPriceBookEntry;
75)
76)         } else {
77)             // Handle error response
78)                       String  error  =  'Failed  with  status:  '  +
   response.getStatusCode() + ' ' + response.getStatus();
79)             System.debug(error);
80)         }
81)     }
```

```
82) }
```

2) To get Product Test

```
@isTest
private class GetProductsTest {
    @isTest static void testCallout() {
        // Set mock callout class



        Test.startTest();
                            Test.setMock(HttpCalloutMock.class,    new
GetProductsCalloutMock());

        // Call method to test.
        // This causes a fake response to be sent
        // from the class that implements HttpCalloutMock.
        HttpResponse res = GetProducts.GetAllProducts();

        // Verify response received contains fake values
        String contentType = res.getHeader('Content-Type');
        System.assert(contentType == 'application/json');
        String actualValue = res.getBody();
                    String   expectedValue   =   '[{"id":1,"title":"Hello
there","Description":"Hello                    I                    am
mike","price":109.5},{"id":2,"title":"Hello                        there
ridham","Description":"Hello I am ridham","price":110.4}]';
        System.assertEquals(actualValue, expectedValue);
        System.assertEquals(200, res.getStatusCode());
```

```
        GetProducts.insertProduct();
                  Product2  product=[select  Name  from  Product2  where
ProductCode='1'];
        System.assertEquals('Hello there', product.Name);
        Test.stopTest();

    }
}
```

3) Post Speaker End Point

```
@RestResource(urlMapping='/Speakers/*')

global with sharing class PostSpeakerEndPoint {
    @HttpPost
    global static Id doPost(String name, String email,
        String phone, String company) {
            try{

                Speaker__c speakerRecord = new Speaker__c(
                Name=name,
                Email__c=email,
                Phone__c=phone,
                Company__c=company);

                insert speakerRecord;
```

```
                Id speakerId=speakerRecord.id;
                RestContext.response.statusCode=200;
                // system.debug(speakerId);
                return speakerId;

            }catch(Exception e){
                RestContext.response.statusCode=400;
                // system.debug('hell');
                return null;
                // return 'fail';
            }


    }
}
```

4) Speaker Registration page extension

```
public class SpeakerRegistrationPageExtension {

    private ApexPages.StandardController controller;
    public Speaker__c speaker{get;set;}
    public Event_Speaker__c eventSpeaker{get;set;}
    // public String Event__c.Organizer__r.Address__c {get;set;}
    public SpeakerRegistrationPageExtension(ApexPages.StandardController
controller) {
        this.controller = controller;
        speaker=new Speaker__c();
```

```
        eventSpeaker=new Event_Speaker__c();



    }
    public void s() {
        Savepoint sp=Database.setSavepoint();
        try{
            insert speaker;
            eventSpeaker.Speaker__c=speaker.id;
            insert eventSpeaker;


        }catch(Exception e){
            System.debug(e.getMessage());
            Database.rollback(sp);
        }
    }
}
```

Ridm

ORIGINALITY REPORT

15%	11%	0%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	scholarworks.lib.csusb.edu Internet Source	7%
2	vocal.media Internet Source	1%
3	Submitted to MAHSA University Student Paper	<1%
4	Submitted to John F. Kennedy High School Student Paper	<1%
5	Submitted to University of Greenwich Student Paper	<1%
6	Submitted to Southern New Hampshire University - Continuing Education Student Paper	<1%
7	Submitted to Dhirubhai Ambani Institute of Information and Communication Student Paper	<1%
8	Submitted to McMaster University Student Paper	<1%
9	Submitted to University of Northampton	