Introduction (Slide - 3)

- Our project is about a Car Rental Company specializing in Renting Cars to customers.
- It is an Online system through which customers can view and get details of available cars, register personal details, view profiles and book cars.

Functionality (Slide - 4)

1. **Create and maintain Car records :**
   Admin can add new car details and also can edit and update existing car details maintaining an organized Car database.

2. **Search Car details :**
   The UI allows the users to search cars easily by Car Name, Car type, Color, Brand.

3. **Register and maintain Customer records.**
   The users can register and update their details in the Customer records.

4. **Prevent Duplicate Customer records :**

One Customer cannot Create two Customer records using the same details (Name, Email, Phone).

5. **Allows the customer to rent available cars:**
   Here comes the most important feature of the application. The customers can book cars directly from the homepage prior to their expected pick-up date.

6. **Prevent Duplicate Booking records :**

Two Customers cannot book the same car for the same date.

7. **Customers cannot give Present day as Booking date.**

8. **For Booking, ‘To-Date’ always greater than ‘From-Date’**

9. **Customers feedback on overall experience :**

Customers can provide feedback to the site based on their experience with the form of rating along with feedback messages.

10. **Report is generated based on the customer's feedback.**

The admin will be able to monitor customer satisfaction by the reports generated from the customer feedback ratings.

Data Modeling (Slide - 5)

In this slide we can see the schema architecture of our application.
Here we have 4 main objects which are: Rental customer, car, Booking and Feedback.

- The rental customer object is having the customer details such as customer id, email id, phone no. customer address.
- Car object is having Car details like Car id, car name, brand etc.
- The booking object is having the booking details like Booking id, Destination location, pickup location, From date, To date,
  Booking object also has Two lookup fields Car and Customer and those work as a link between booking and those two objects.
- Feedback object is having fields like feedback id, messages, ratings.
  This object also has a lookup field Customer which connects this object to customer object.

Profile (Slide - 6)

- **Admin :** We have created an admin profile by cloning the System Administrator. An admin has the permission to configure and customize the application. It can also access all the functionality.

- **Rental Customer :** We also have created a Rental customer by cloning the Standard User. A Customer will be able to book a car, give reviews and see others reviews, edit their own records but their functionality is limited

  In this project Admin will be able to Create, Read , Edit and Delete the records but Rental Customer has the permission to read records only.

Page layout (Slide -7)

Page layout is used to modify the User Interface. In this Project we have created two page layouts one for the Admin and one for the User(Rental Customer) on Car Object .

Car Layout for Admin - As we can see that in this layout we can see all the fields available in Car Object.

Car Layout for Customer - This is the layout that we have created for the rental customer. A customer will be able to see fields like Car Id, Car Name,Image, Car Type, Color, Brand, Seats, Price per Day, Fuel Type. Here a Customer would not be able to see fields like Created by, owner, Last Modified by.

Validation Rules(Slide -8,9)

We know that the Validation rule gets executed when we are saving a record that we have created or edited. In this project we have created 3 validation Rules.

- The very first validation rule is on Booking object to check if 'To-day' is greater than 'From Date' as while booking the final date must always be greater than the date we are booking the car. So we have written this formula `From_Date__c > To_Date__c.` As we can see in the ppt that If this formula becomes true as we are trying to save a new record then an error message will be displayed at 'To-Date' field location and the record will not be saved. But if we enter the correct dates and formula becomes false then the record will be saved.
- The second validation Rule that we have created is also on Booking object to check if Validation rule to check 'From date' should be more than Today. Like for example if we are booking a car Today then From date should be tomorrow or any other dates except Today. So we have written a formula `From_Date__c <= TODAY().` As we can see in this ppt that if this formula becomes true error message will be displayed on 'From-Date' field location and the record will not be saved.
- The third validation rule that we have created is on Rental Customer object to make Rental Customer's Phone and Email mandatory.If the email and phone field is left empty then a record will not be saved unless and until one fill the fields. So we have written a formula `ISBLANK( Email__c ) || ISBLANK( Phone_Number__c )` using ISBLANK function. As we can see in the ppt if any of the fields or both fields is left empty then error message will be displayed on top of the page and records will not be saved.

Flows (10)

We know that Flow is an automation tool provided by Salesforce which can be used to perform various tasks like, Sending an Email, Posting a chatter, Sending the custom Notifications.

In this project we use record trigger flow to Send an Email to customers . When a customer successfully booked a car or created a record then this auto generated mail is triggered. In this mail body we also used dynamic subject which shows customers booking details,cars details. The email body's subject is changed according to customer details . We send this email to the customer registered email address fetching from customer profile.

Next we use schedule trigger to send reminder mail to the customers . Here we send reminder mail to the customers 1 days before the customers booking start date, For that we use

Time Source : Booking__c: From date
Offset Number : 1
Offset Options : Days before

In this email body we have also added dynamic subject and dynamic body based on customer details and customer booking details.

Flows (11)

We use another record trigger flow to Send an Email to the customer. We used four conditions when customers edit details of booking (Such as From_date, To_date, Pickup_Location or Destination) then this auto generated mail is triggered. In this mail customers can show his updated booking information. This email body is also dynamic which contains customers updated information. This email is also sent to the customer registered email address fetching from customer profile.

LWC(12)

Lightning web components are custom HTML elements built using HTML and modern JavaScript and CSS.
It allows the developer to build custom components of their own choice.
In our project we've used custom LWC component named carCantroller which is having two functionalities :

1. Custom data-table to show Car details
2. Custom modal (pop-up) for booking

Here in this slide we can see snippets of vscode with the files which are needed to build the component.

Carcantroller.html: this is the html file which has the tags of custom data table and the modal for the booking.

Carcantroller.css: this is the css file responsible for the custom design of the data table.

Carcantroller.js: this is the Modern javascript file which is responsible for all backend connections to the SOQL and custom functionality.

Import functionality declared in a module eg-lwc(the core module), use the import statement.
To allow other code to use functionality in a module, use the export statement.

carController.js-meta : XML file that defines the metadata configuration values for the component eg-

Components Label
Availability
Configuration Attributes
Builder Attributes

CarController: (Slide - 13)

Apex Class CarController can be imported into JavaScript Classes by Lightning web Components using import.
After which, CarController can be called into the components as functions either via using the wire service (@wire) or imperatively wherein the Apex Method should be marked with @AuraEnabled(cacheable=true).

To get this data the component will wire an Apex method. The Apex method makes a SOQL query that returns a list of Cars in the Home page.

To return list of Cars We use a SOQL query to select certain records Car Name,Car Type ,Colour,Brand,ImgURL,Price from Car with limit 10.

Duplicate Customer: (Slide - 14)

Apex triggers enable you to perform custom actions before or after events to record in Salesforce, such as insertions, updates, or deletions.
We have created an Apex Trigger to prevent duplicate customer profiles when a customer has identical Email or Contact Number on both profiles.

For that purpose, we have created a SOQL query to select certain records like email and phone which checks if email and phone matches with any other existing customer's email and phone. If it gets a match, it will show an error that 'Customer is already exist'.

Duplicate Booking: (Slide - 15)

We have also created an Apex Trigger to prevent duplicate booking when a car has identical booking details such as From Date and To date on both booking.

Here we create a SOQL query that selects car, from_date and to_date from booking and checks whether booking duration clashes with any existing booking duration with the same car. If it conflicts, it gets an error 'Car is already booked' under the Car field and 'This date is already booked' under the ' From Date' field of 'New Booking' pop-up.

Delete Trigger(Slide-16)

In the next part of BookingDuplicate trigger class, we check that when a user wants to delete his or her booking,then customers can't delete it before one day of booking start date or on the starting day of booking. For that we check the difference between the user's system date. If the difference is one or system date is the booking start date then it shows an error pop-up showing 'Cannot delete Booking before 1 day or the Same day'. And the booking can not be deleted.

Reports (Slide - 17)

Salesforce report provides us a visual representation of the sales analytics and gives access to key business data insights which helps in making better decisions.

Reports are displayed in Salesforce in rows and columns and can be filtered according to a particular object.We can group the data together and also display them in the form of graphical chart.

Let's say we just want to see people who gave the feedback after their car rental experiences.Thus, we can predict which experience gets the higher rating and which needs to be improved.

Future Scope & Conclusion (Slide - 18)

- Query raise & Support

  A support system allows you to efficiently provide service to customers, partners, and internal employees by assigning a ticket (or case number) to every service inquiry. You can use it to track the issue to its resolution whether you are the customer, partner, service agent, manager, or even the CEO.

- Payment gateway integration

  Even though Salesforce Commerce Cloud (SFCC) is a powerful platform for digital commerce,many things like payments or tax calculations can't be performed without specific integrations. The integrations enhance SFCC functionality and provide those services to end customers. Integrating payment services with SFCC is enabled using Salesforce LINK Cartridges. This blog will tell you how to build a LINK cartridge for a payment integration and offer your product to enterprise-level companies.

- UI improvement

  If we are building a custom web or mobile app and need a user interface that lets users view, create, edit, and delete Salesforce records—**a user interface that looks and behaves like Salesforce**—UI API is the way to go.

- Chatbot development

  A chatbot is **an application that simulates human conversation, either aloud or via text message**. Instead of having a conversation with a person, like a sales rep or support agent, a customer can have a conversation with a computer. Whether through typing or talking, a chatbot can connect with a customer.in future we can build a chatbot in this platform which will assist the customers using the application more easily.

- Increase Automation to improve efficiency.

Conclusion (Slide - 19)

The web based car rental system has offered an advantage to both customers as well as Car Rental Company to efficiently and effectively manage the business and satisfies customers' need at the click of a button.

Our aim was to provide a very user-friendly Car Rental experience