



HOTEL MANAGEMENT PROJECT USING LIGHTENING WEB COMPONENT (LWC)

2021





Table of Contents

PROJECT WORK: STORY 1	3
PROJECT WORK: STORY 2	14
PROJECT WORK: STORY 3	18
PROJECT WORK: STORY 4	21

Dhanashri Gade



PROJECT WORK: STORY 1

Executive Summary:

Thank you so much for giving us time for Meeting.

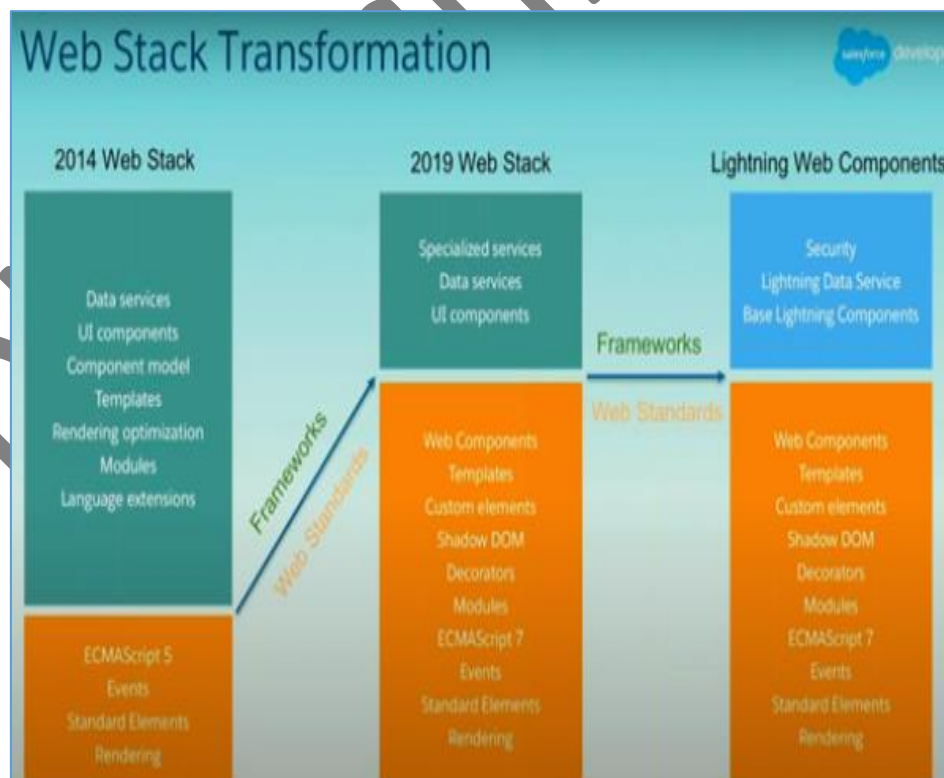
Intelogik has prepared solution as per your requirements.

Intelogik has provided proposed Solution for your Requirement.

SALESFORCE Platform with Lightning Web Component is Solution to your requirement.

LIGHTENING WEB COMPONENT:

- LWC is new project building pattern in Salesforce.






INTELOGIK-INTERNSHIP PROGRAM

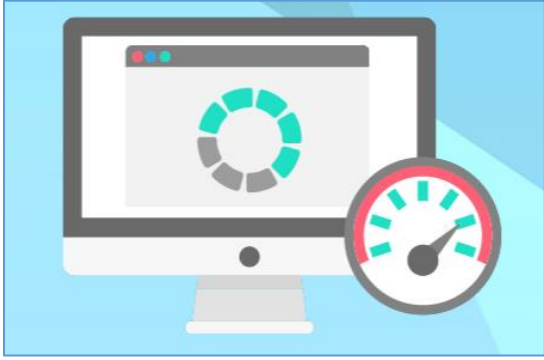
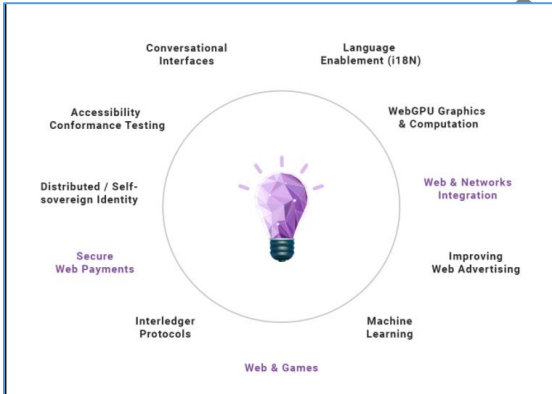
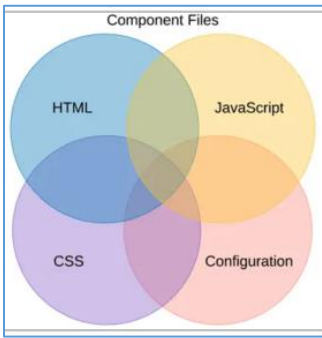
- LWC is way of creating Web Components in Lightning Environment.
- All modern Web Browsers are working on Web Standards and they are improving their performance every day.
- LWC also follows W3C Web Component standards to take advantage of modern technologies and native Browser's features to execute it as Fast with use of minimally Network Bandwidth.
- LWC are built on Latest Web Standard Features.
- Components created using LWC are better in performance than LAC.
- LWC apps are robust as they include modern Web Standards.

Dhanashri Gade

❖ BENEFITS OF USING LIGHTNING WEB COMPONENTS IN BUSINESS :

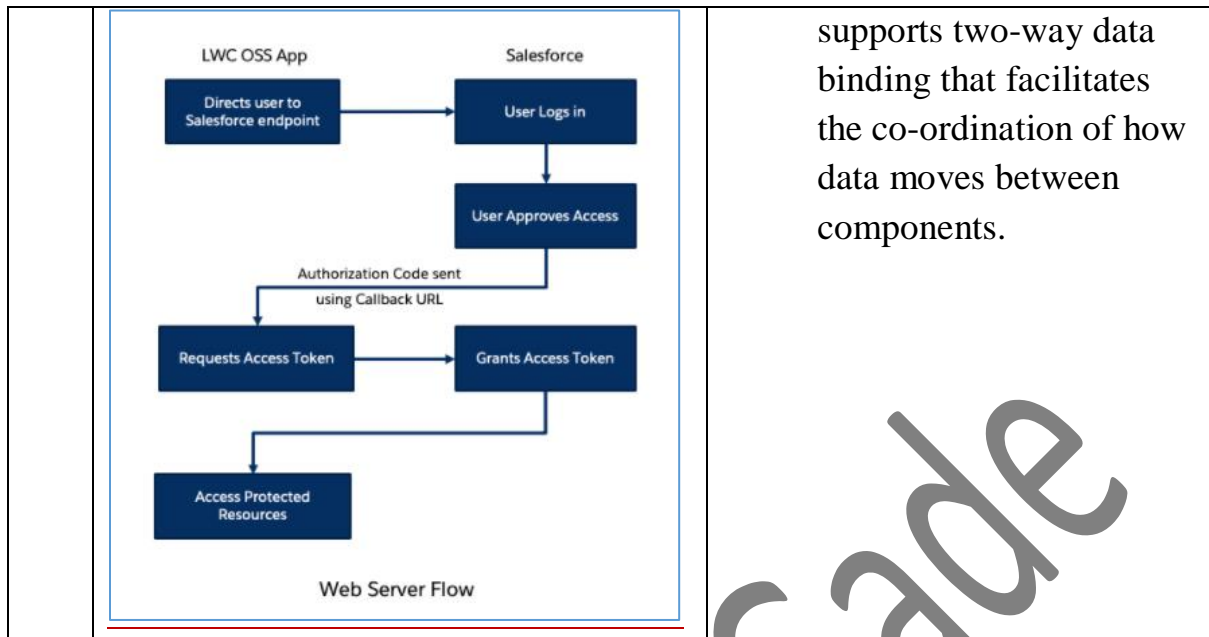


<u>SR NO</u>	<u>Benefits of using LWC in Business</u>	<u>Explanation</u>
1	<p><u>Performance Enhancements</u></p> 	<ul style="list-style-type: none"> Lightning Web Components are most likely to render faster and provide better performance leading to deliverability, as there is no added abstraction layer.

<p><u>2</u></p>	<p><u>Faster loading of websites</u></p> 	<ul style="list-style-type: none"> Lightning Web Components is <u>faster in loading the developed components</u> and is a lightweight framework built using web standards.
<p><u>3</u></p>	<p><u>Improved web standards</u></p> 	<ul style="list-style-type: none"> Lightning Web Components has built-in browser security features from Web Components Standards, allowing out-of-box usage and fewer custom functions. With Lightning Web Components, the more we learn about web standards, the more we have the skills that can be used in other technologies.
<p><u>4.</u></p>	<p><u>Common Components</u></p> 	<ul style="list-style-type: none"> Using Lightning Web Components, it is now easy to write components that do not have a user interface and those components can be reused in other components that are



		more powerful than static tools.
5.	<p><u>Simple to learn</u></p>	<ul style="list-style-type: none">• Lightning Web Components are basically taking form through the native web standards in the browser.• Meaning there is no layer of specialized required like Aura Framework or any other framework, just JavaScript needs to be created.
6.	<p><u>Easier pathway for developers</u></p>	<ul style="list-style-type: none">• No additional framework is required to learn with a view to create Lightning Web Components and therefore the transition for developers is much easier.
7.	<p><u>Better security, testing and browser compatibility</u></p>	<ul style="list-style-type: none">• With Lightning Web Components, CSS (Cascading style sheets), Script and more the event range is better and more limited.• Each of these offers more consistency in the design of components.• Furthermore, Lightning Web Components

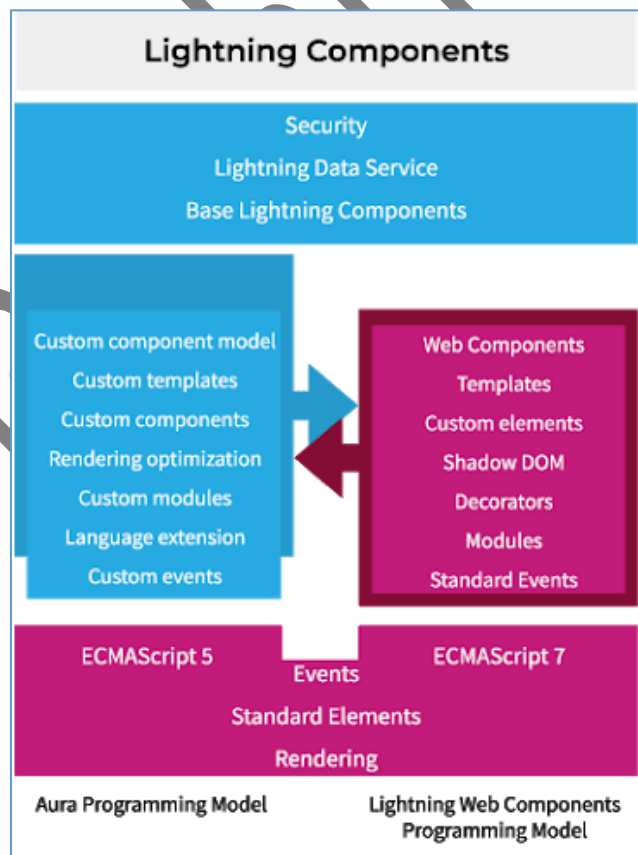




❖ **BENEFITS OF USING LIGHTNING WEB COMPONENTS (LWC) OVER LIGHTNING AURA COMPONENTS (LAC):**



Client has created few Components in LAC. Client is worried about those components whether those can be used in LWC as Project is shifting to LWC.





- Aura Components and LWC can co-exist and interoperate means can work in same project without any issue.
- Aura Components and LWC can co-exist on same page as well.
- Aura Components can include Lightning Web component but vice versa is not possible.
- Aura Components and LWC can share same Base Lightning features.
- Aura Components and LWC can share same underlying Service like User Interface API, Lightning Data Service.

<u>SR NO</u>	<u>LWC Over LAC in Business</u>	<u>Explanation</u>
<u>1</u>	<u>Site Loading Speed</u>	<ul style="list-style-type: none">• The Lightning web component is hugely faster than the Aura component in loading web pages.
<u>2</u>	<u>Easy Learning</u>	<ul style="list-style-type: none">• The Lightning web component script uses HTML & JavaScript language. That means tasks are more comfortable to do.
<u>3</u>	<u>Best performance</u>	<ul style="list-style-type: none">• As LWC builds on web components, it makes LWC extremely lightweight & efficient in memory management.• That's the reason LWC runs much faster than



		Aura lightning components.
4.	<u>Debugging made easy</u>	<ul style="list-style-type: none">• Debugging is significantly easier for LWC than ALC.• Different Salesforce discussion forums address & execute the step by step debugging topics very quickly.
5.	<u>Easy to ramp:</u>	<ul style="list-style-type: none">• The absence of a different framework enables developers to learn LWC more easily compared to LAC.
6.	<u>Standardized</u>	<ul style="list-style-type: none">• As LWC uses core web components, it offers everything needed for the blazing-fast performance of browsers.• It uses common coding languages like CSS, JavaScript & HTML.



7.	<u>Inbuilt browser security features</u>	<ul style="list-style-type: none">• LWC has inbuilt browser security features driving its out-of-the-box usage while Aura requires an event-driven programming model that is GUI dependent.• It focuses on actions like button press & clicks determining the flow of movement. The Lightning web components support these actions.
8.	<u>Data binding</u>	<ul style="list-style-type: none">• Data binding is a technique that binds data sources from the provider and consumer together and synchronizes them.• LWC supports two-way data binding that helps in coordinating how data moves between different components.
9.	<u>Unit testing Support</u>	<ul style="list-style-type: none">• LWC supports the unit testing framework over LAC



10.	<u>Versatility</u>	<ul style="list-style-type: none">• Full encapsulation is available with LWC over LAC
-----	--------------------	---

Dhanashri Gade



PROJECT WORK: STORY 2

Client wants to execute all business process on SALESFORCE Platform using LWC with following requirement of creating few tables and fields.

- Create App.
 1. Create “Hotel Management Project in LWC” app in salesforce org in Lightning environment.

(1).Table Details:

- Create Objects and Fields :

1. Object Name : Guest Master

Sr No.	Field Name	Field Type	Field Values(If Present)
1.	Guest Code	Standard Field→ Name →AutoNumber	
2.	Guest First Name	Text	
3.	Guest Last Name	Text	
4.	Guest Phone	Phone	
5.	Guest Email	Email	

Fields & Relationships		
8 Items, Sorted by Field Label		
FIELD LABEL	FIELD NAME	DATA TYPE
Created By	CreatedById	Lookup(User)
Guest Code	Name	Auto Number
Guest Email	Guest_Email__c	Email
Guest First Name	Guest_First_Name__c	Text(50)
Guest Last Name	Guest_Last_Name__c	Text(50)
Guest Phone	Guest_Phone__c	Phone
Last Modified By	LastModifiedById	Lookup(User)
Owner	OwnerId	Lookup(User,Group)



2. Object Name : Hotel Master

Sr No.	Field Name	Field Type	Field Values(If Present)
1.	Hotel Code	Standard Field→ Name → AutoNumber	
2.	Hotel City	Text	
3.	Hotel Type	Picklist	3 Star, 4 Star, 5 Star
4.	Per day cost	Currency	
5.	WiFi Availability	Checkbox	
6.	Cancellation Availability	Checkbox	
7.	Cash pay Availability	Checkbox	

Fields & Relationships

10 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE
Cancellation Availability	Cancellation_Availability__c	Checkbox
Cash pay Availability	Cash_pay_Availability__c	Checkbox
Created By	CreatedById	Lookup(User)
Hotel City	Hotel_City__c	Text(50)
Hotel Code	Name	Auto Number
Hotel Type	Hotel_Type__c	Picklist
Last Modified By	LastModifiedById	Lookup(User)
Owner	OwnerId	Lookup(User,Group)
Per day cost	Per_day_cost__c	Currency(17, 1)
WiFi Availability	WiFi_Availability__c	Checkbox



3. Object Name : Payment Master

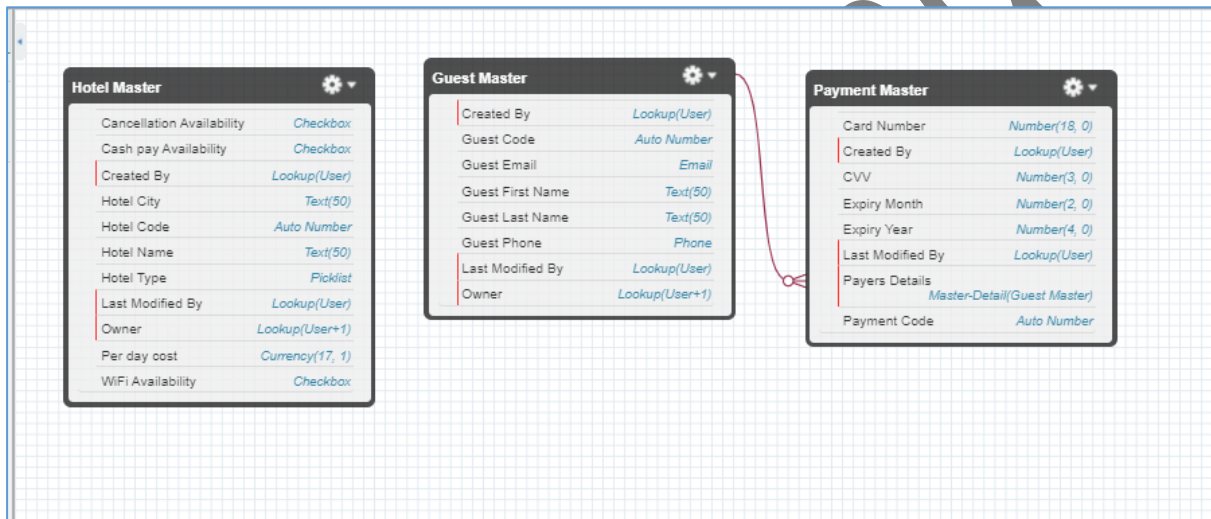
Sr No.	Field Name	Field Type	Field Values(If Present)
1.	Payment Code	Standard Field→ Name → AutoNumber	
2.	Payers Details	Relationship to Guest Master	
3.	Card Number	Number	
4.	Expiry Month	Number(Length : 2)	
5.	Expiry Year	Number(Length : 4)	
6.	CVV	Number(Length : 3)	

Fields & Relationships			Q Quick
9 Items, Sorted by Field Label			
FIELD LABEL	FIELD NAME	DATA TYPE	
Card Number	Card_Number__c	Number(18, 0)	
Created By	CreatedById	Lookup(User)	
CVV	CVV__c	Number(3, 0)	
Expiry Month	Expiry_Month__c	Number(2, 0)	
Expiry Year	Expiry_Year__c	Number(4, 0)	
Last Modified By	LastModifiedById	Lookup(User)	
Owner	OwnerId	Lookup(User,Group)	
Payers Details	Payers_Details__c	Lookup(Guest Master)	
Payment Code	Name	Auto Number	



(2).Relationship Between Objects:

Sr No.	Type of Relationship	Parent Object	Child Object	Field Name
1.	Master Detail	Guest Master	Payment Master	Payers Details





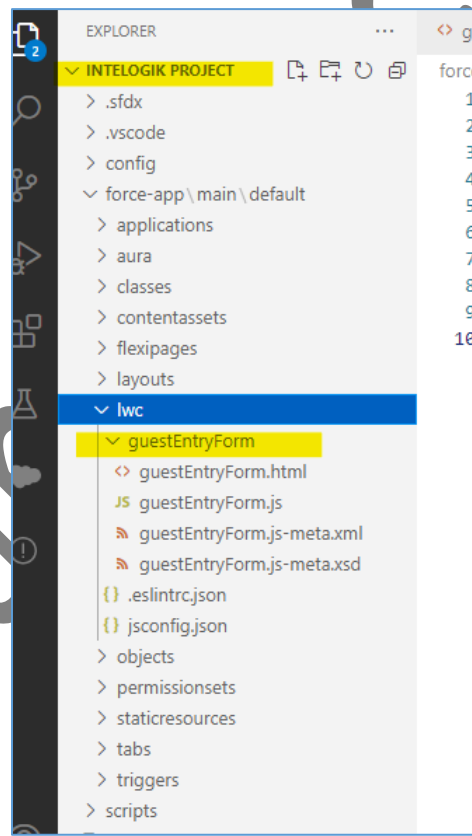
PROJECT WORK: STORY 3

Client is asking for “Guest Entry Form” Component in LWC.

Once user will enter data and click on save then, Data should be committed to respective Table.

Limited data should be entered in table.

LWC Component (Name) : guestEntryForm



Three kind of files are created in LWC component:

1)HTML 2)XML 3)JS

**LWC Component (HTML) : guestEntryForm.html**

```
<> guestEntryForm.html X  guestEntryForm.js-meta.xml  JS guestEntryForm.js
force-app > main > default > lwc > guestEntryForm > <> guestEntryForm.html > template
1  <template>
2      <lightning-card title = "Guest Entry Form">
3          <div class = "slds-p-horizontal_small"></div>
4          <lightning-layout>
5              <lightning-layout-item>
6                  <lightning-record-form
7                      object-api-name={guestInputApiName}
8                      fields={guestFieldList}
9                      onSuccess={guestMasterHandleUpdate}
10                     columns = "2">
11                  </lightning-record-form>
12              </lightning-layout-item>
13          </lightning-layout>
14      </lightning-card>
15  </template>
16
```

LWC Component (js) : guestEntryForm.js

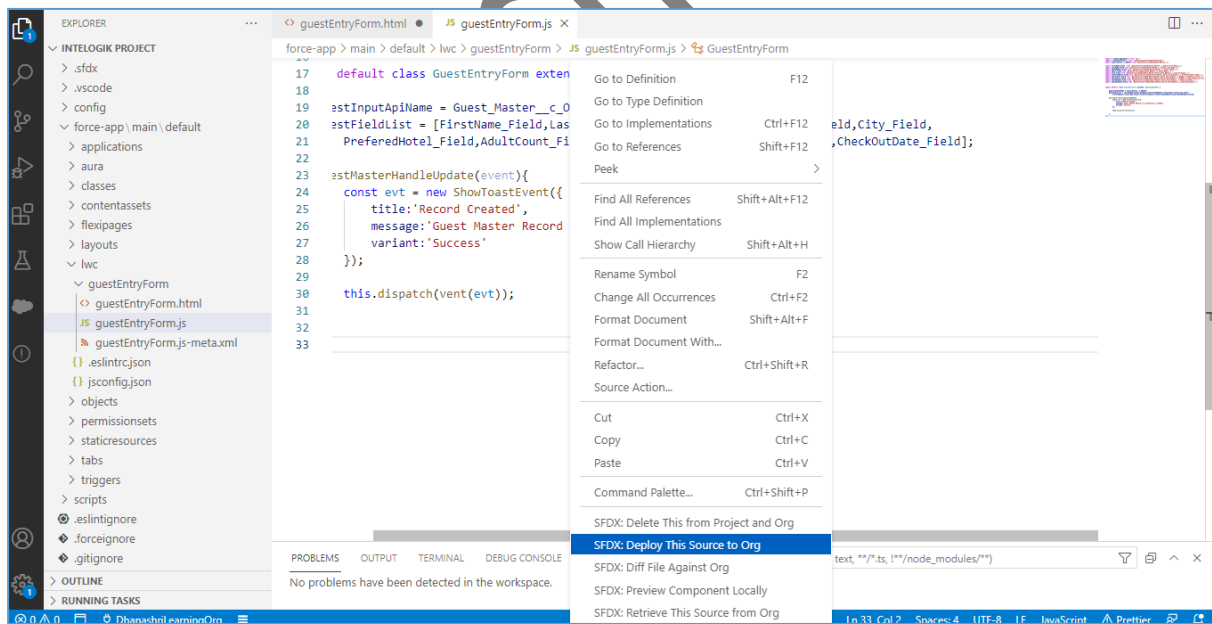
```
<> guestEntryForm.html  guestEntryForm.js-meta.xml  JS guestEntryForm.js X
force-app > main > default > lwc > guestEntryForm > JS guestEntryForm.js > ...
1  import { LightningElement } from 'lwc';
2  import { ShowToastEvent } from 'lightning/platformShowToastEvent';
3  import Guest_Master_Entry_Form_Detail__c_OBJECT from '@salesforce/schema/Guest_Master_Entry_Form_Detail__c';
4
5  import FirstName_Field from '@salesforce/schema/Guest_Master_Entry_Form_Detail__c.Guest_First_Name__c';
6  import LastName_Field from '@salesforce/schema/Guest_Master_Entry_Form_Detail__c.Guest_Last_Name__c';
7  import PhoneNumber_Field from '@salesforce/schema/Guest_Master_Entry_Form_Detail__c.Guest_Phone__c';
8  import Email_Field from '@salesforce/schema/Guest_Master_Entry_Form_Detail__c.Guest_Email__c';
9  import City_Field from '@salesforce/schema/Guest_Master_Entry_Form_Detail__c.City_Of_Hotel__c';
10 import PreferredHotel_Field from '@salesforce/schema/Guest_Master_Entry_Form_Detail__c.Preferred_Hotel_Type__c';
11 import AdultCount_Field from '@salesforce/schema/Guest_Master_Entry_Form_Detail__c.Number_of_Adults_12_yrs__c';
12 import ChildCount_Field from '@salesforce/schema/Guest_Master_Entry_Form_Detail__c.Number_of_Children_0_12_Yrs__c';
13 import CheckInDate_Field from '@salesforce/schema/Guest_Master_Entry_Form_Detail__c.Check_In_Date__c';
14 import CheckOutDate_Field from '@salesforce/schema/Guest_Master_Entry_Form_Detail__c.Check_Out_Date__c';
15
16
17 export default class GuestEntryForm extends LightningElement {
18
19     guestInputApiName = Guest_Master_Entry_Form_Detail__c_OBJECT;
20     guestFieldList = [FirstName_Field,LastName_Field,PhoneNumber_Field,Email_Field,City_Field,
21         PreferredHotel_Field,AdultCount_Field,ChildCount_Field,CheckInDate_Field,CheckOutDate_Field];
22
23     guestMasterHandleUpdate(event){
24         const evt = new ShowToastEvent({
25             title:'Record Created',
26             message:'Guest Master Record is successfully created',
27             variant:'Success'
28         });
29
30         this.dispatchEvent(evt);
31     }
32 }
```



LWC Component (xml) : guestEntryForm.xml

```
guestEntryForm.html X guestEntryForm.js-meta.xml X JS guestEntryForm.js
force-app > main > default > lwc > guestEntryForm > guestEntryForm.js-meta.xml > LightningComponentBundle
1  <?xml version="1.0" encoding="UTF-8"?>
2  <LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
3      <apiVersion>50.0</apiVersion>
4      <isExposed>true</isExposed>
5      <targets>
6          <target>lightning__AppPage</target>
7          <target>lightning__RecordPage</target>
8          <target>lightning__HomePage</target>
9      </targets>
10 </LightningComponentBundle>
```

Deploy to SFDC org from VSCode





Create new Lightning App page in Lightning app Builder and add to Hotel Management App.

Guest Entry Form

Guest First Name

Guest Last Name

Guest Phone

Guest Email

City Of Hotel

Preferred Hotel Type

Number of Adults(12+ yrs)

Number of Children(0-12 Yrs)

Check In Date

Check Out Date

Cancel Save

PROJECT WORK: STORY 4

Client is asking for “Hotel Listing” Component in LWC.

Depending on data entered by User in “Guest Entry Form” Component, Page should show List of Suitable Hotels with all possible Hotel Specifications like Wi-Fi Availability Etc. depending on Hotel City and Hotel Type.

Before Every Hotel there should be Checkbox which User will select along with Hotel Specifications

User should click on ‘SELECT’ Button after selecting any Hotel from List.



LWC Component (xml) : showHotelList.xml

```
guestEntryForm.html × guestEntryForm.js-meta.xml × JS guestEntryForm.js
force-app > main > default > lwc > guestEntryForm > guestEntryForm.js-meta.xml > LightningComponentBundle
1  <?xml version="1.0" encoding="UTF-8"?>
2  <LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
3      <apiVersion>50.0</apiVersion>
4      <isExposed>true</isExposed>
5      <targets>
6          <target>lightning__AppPage</target>
7          <target>lightning__RecordPage</target>
8          <target>lightning__HomePage</target>
9      </targets>
10 </LightningComponentBundle>
```