

Micro LWC Component how to create email survey form.

After Signing up for a developer edition org, start with milestone 1 that is creating an Apex class to handle the logic for sending emails. This class will be used by the LWC to trigger the email sending process.

What you'll learn

1. Building LWC component
2. Survey and Feedback tracking.

Milestone 1: Build Survey Form Lightning App

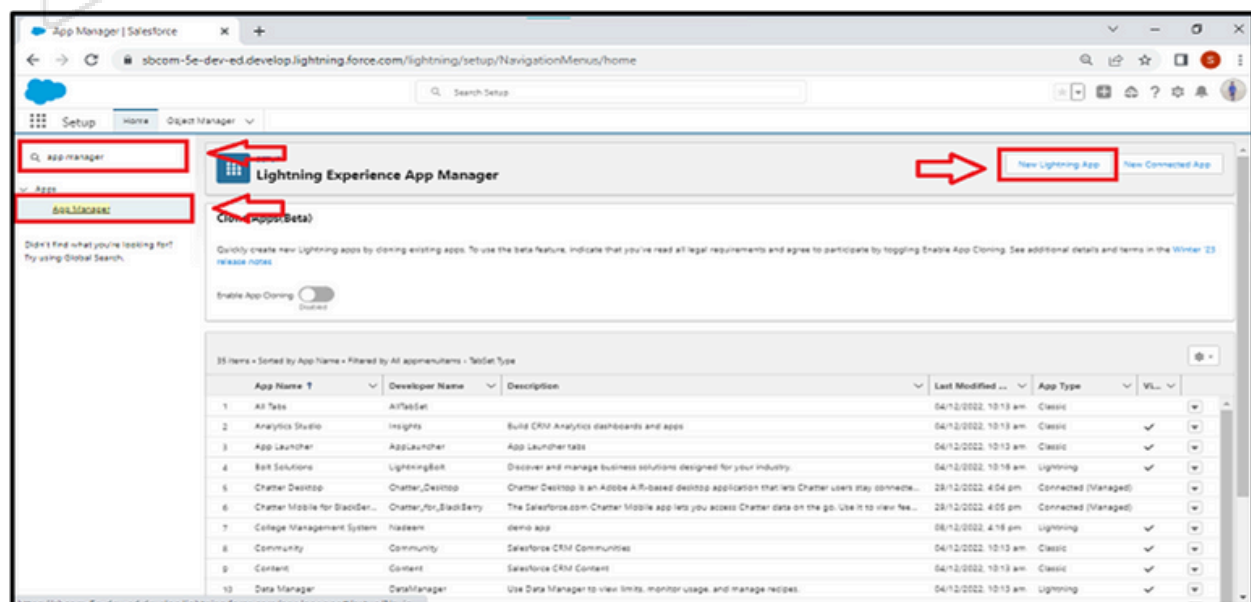
The app is built in Lightning Experience, depending on your organization's preferences and requirements. Below is a brief overview of how you might design such an app :

- Design a Lightning App using the Lightning App Builder.
- Include tabs for the required Custom Objects.
- Customize the app's color scheme and logo to give it a branded look.

Activity -1

Create the Lightning App

1. Go to the Gear Icon → Click on setup → click on the Home Button → Go to the quick find box and select the App Manager



2.Fill the app name as an in app details and branding →Next → (App option page) keep it as default → Next

3. Utility Items keep it as default → Next → (Add Navigation Items)(add custom tabs if needed) → Next → (Add User Profile) Add System Administrator, Salesforce platform user, Standard User → Next.

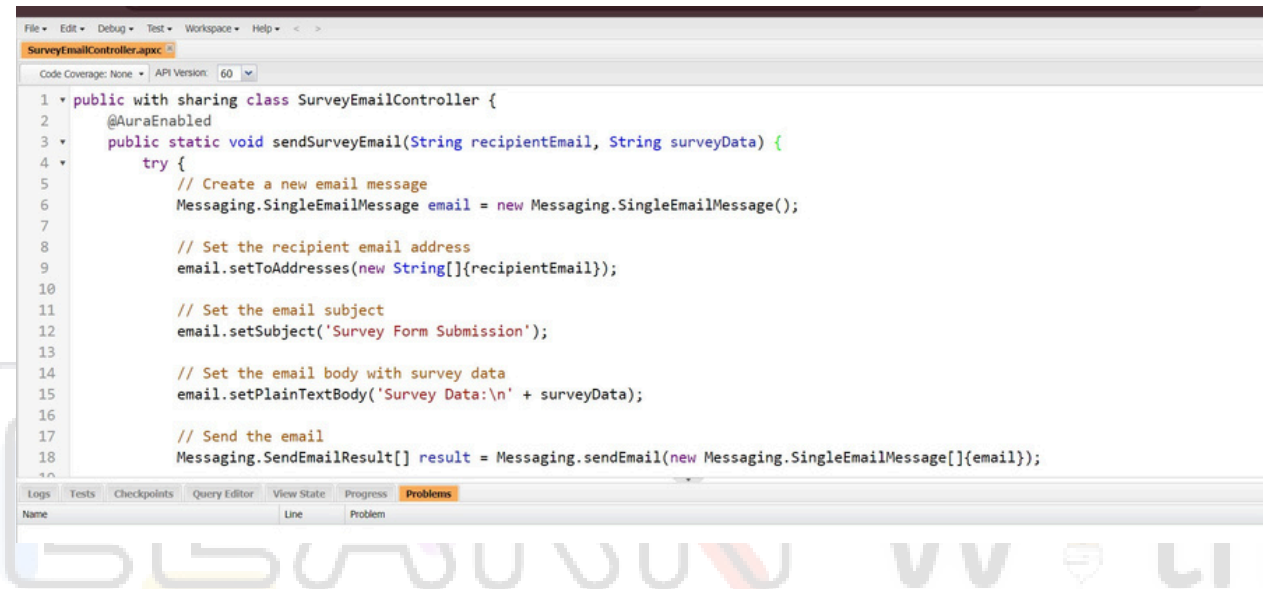
4. To Add Navigation Items:
Select the items from the search bar and move it using the arrow button → Next.
select all the tabs which you have created

5. To Add User Profiles:
Search profiles in search bar → click on the arrow button & select Standard user,
standard Platform user & System Admin Profile→ save & finish.

Milestone 2: SurveyEmailController Class

Activity- 1

Defining the logic for automating actions based on Apex class to handle the email notifications on the survey. It includes methods for sending notifications for these stages.



Apex Class :

```
public with sharing class SurveyEmailController {

    @AuraEnabled

    public static void sendSurveyEmail(String recipientEmail, String surveyData) {

        try{

            // Create a new email message

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


            // Set the recipient email address

            email.setToAddresses(new String[]{recipientEmail});


            // Set the email subject
```

```
email.setSubject('Survey Form Submission');
```

```
// Set the email body with survey data
```

```
email.setPlainTextBody('Survey Data:\n' + surveyData);
```

```
// Send the email
```

```
Messaging.SendEmailResult[] result = Messaging.sendEmail(new  
Messaging.SingleEmailMessage[]{email});
```

```
// Check the result of the email sending operation
```

```
if (result[0].success) {
```

```
    System.debug('Email sent successfully');
```

```
} else {
```

```
    System.debug('Error sending email: ' + result[0].errors[0].message);
```

```
    throw new AuraHandledException('Error sending email: ' +  
result[0].errors[0].message);
```

```
}
```

```
} catch (Exception e) {
```

```
    // Handle exceptions and provide appropriate error handling
```

```
    System.debug('Error sending email: ' + e.getMessage());
```

```
    throw new AuraHandledException('Error sending email: ' + e.getMessage());
```

```
}
```

}

}

Milestone 3 : SurveyForm LWC component

Activity- 1

Open a Workspace:

Open Visual Studio Code and create a new workspace or open an existing one. A workspace is a directory that contains your Salesforce projects.

Create a New Salesforce Project:

Open the Command Palette (Ctrl + Shift + P) and run the command "SFDX: Create Project".

Choose the project type.

For LWC development, choose "Standard" for most cases.

Authorize an Org:

Open the Command Palette and run the command "SFDX: Authorize an Org".

Log in to your Salesforce org.

Create a New Lightning Web Component:

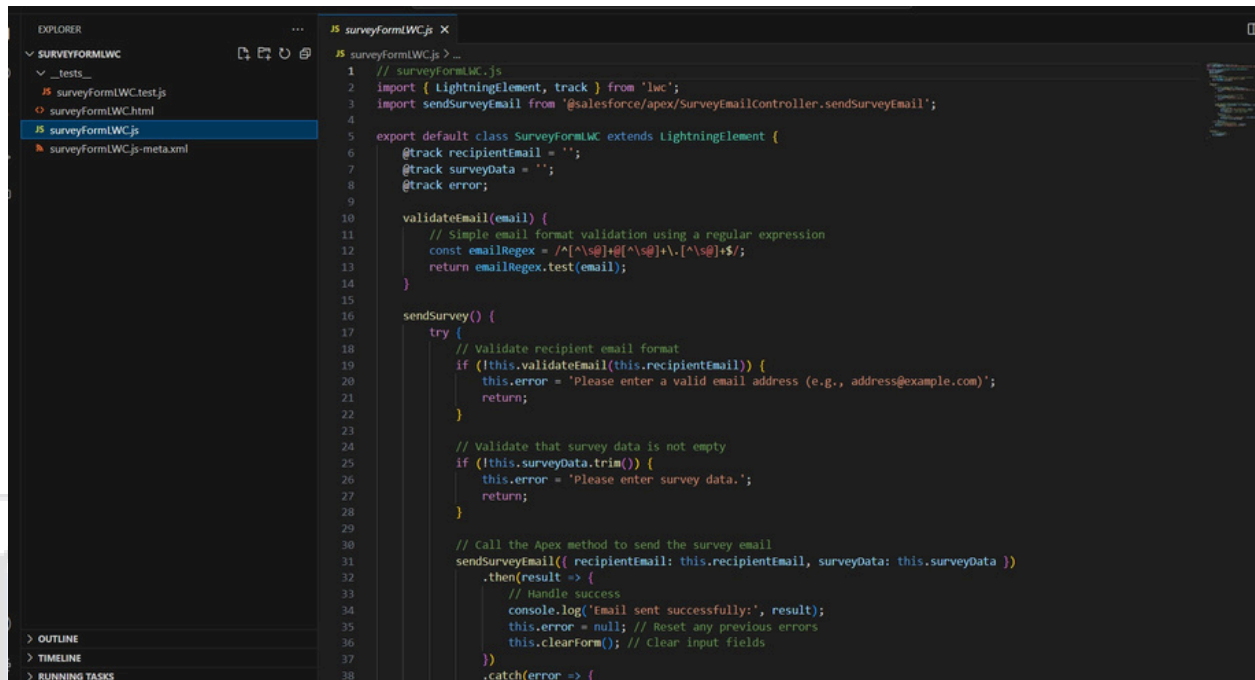
Open the Command Palette and run the command "SFDX: Create Lightning Web Component".

Enter a name for your component (Login) and choose the directory to save it.

Edit Your Lightning Web Component:

Open the newly created component in the force-app/main/default/lwc directory.

Edit the HTML, JavaScript, and CSS files as needed.



```
1 // surveyFormLWC.js
2 import { LightningElement, track } from 'lwc';
3 import sendSurveyEmail from '@salesforce/apex/SurveyEmailController.sendSurveyEmail';
4
5 export default class SurveyFormLWC extends LightningElement {
6     @track recipientEmail = '';
7     @track surveyData = '';
8     @track error;
9
10    validateEmail(email) {
11        // Simple email format validation using a regular expression
12        const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
13        return emailRegex.test(email);
14    }
15
16    sendSurvey() {
17        try {
18            // Validate recipient email format
19            if (!this.validateEmail(this.recipientEmail)) {
20                this.error = 'Please enter a valid email address (e.g., address@example.com)';
21                return;
22            }
23
24            // Validate that survey data is not empty
25            if (!this.surveyData.trim()) {
26                this.error = 'Please enter survey data.';
27                return;
28            }
29
30            // Call the Apex method to send the survey email
31            sendSurveyEmail({ recipientEmail: this.recipientEmail, surveyData: this.surveyData })
32                .then(result => {
33                    // Handle success
34                    console.log('Email sent successfully:', result);
35                    this.error = null; // Reset any previous errors
36                    this.clearForm(); // Clear input fields
37                })
38                .catch(error => {
```

SurveyFormLWC (JavaScript) code:

```
//surveyFormLWC.js
import{LightningElement,track}from'lwc';
importsendSurveyEmailfrom
 '@salesforce/apex/SurveyEmailController.sendSurveyEmail';
```

```
export default class SurveyFormLWC extends LightningElement {
    @trackrecipientEmail="";
    @tracksurveyData="";
    receipientHandle(event){
        if(event.target.label=='RecipientEmail'){
            this.recipientEmail=event.target.value;
        }elseif(event.target.label=='SurveyData'){
            this.surveyData=event.target.value;
        }
    }
    sendSurvey(){
        //Validateinputdataifneeded
        console.log('email'+this.recipientEmail);
        console.log('survey'+this.surveyData);
        //CalltheApexmethodtosendthesurveyemail
```

```

    sendSurveyEmail({ recipientEmail: this.recipientEmail, surveyData:
this.surveyData })
    .then(result => {
        // Handle success
        console.log('Email sent successfully:', result);
    })
    .catch(error => {
        // Handle errors
        console.error('Error sending email:', error);
    });
}
}

```

```

1  <!-- surveyFormLWC.html -->
2  <template>
3      <lightning-card title="Survey Form">
4          <div class="slds-m-around_medium">
5              <label>Email:</label>
6              <lightning-input type="email" required value={recipientEmail} label="Recipient Email" onchange=
(handleEmailChange)></lightning-input>
7
8              <label>Survey Data:</label>
9              <textarea value={surveyData} label="Survey Data" onchange={handleSurveyDataChange}></textarea>
10
11             <lightning-button label="Send Survey" onclick={sendSurvey} disabled={isSendButtonDisabled}></lightning-button>
12
13             <!-- Add this section for error display -->
14             <template if:true={error}>
15                 <div class="slds-notify_container slds-is-relative">
16                     <div class="slds-notify slds-notify_toast slds-theme_error" role="alert">
17                         <span class="slds-assistive-text">Error</span>
18                         <span class="slds-icon_container slds-icon-utility-error slds-m-right_small slds-no-flex
slds-align-top" title="Description of icon when needed">
19                             <lightning-icon icon-name="utility:error" alternative-text="Error icon" variant="inverse"
size="small"></lightning-icon>
20                         </span>
21                         <div class="slds-notify_content">
22                             <h2 class="slds-text-heading_small">{error}</h2>
23                         </div>
24                     </div>
25                 </div>
26             </template>
27         </div>
28     </lightning-card>
29 </template>

```

SurveyFormLWC(html) code:

```
<!-- surveyFormLWC.html -->
```

```
<template>
```

```
<lightning-card title="Survey Form">
```

```
<div class="slds-m-around_medium">
```

```
<!-- Your survey form HTML goes here -->
```

<!-- Include fields for survey questions and input elements -->

<!-- Example: -->

<label>Email:</label>

<lightning-input type="email" value={recipientEmail} onchange={receptientHandle}
label="Recipient Email"></lightning-input>

<lightning-textarea name="input2" label="Survey Data" value="initial
value" onchange={receptientHandle}></lightning-textarea>

<lightning-button label="Send Survey" onclick={sendSurvey}></lightning-button>

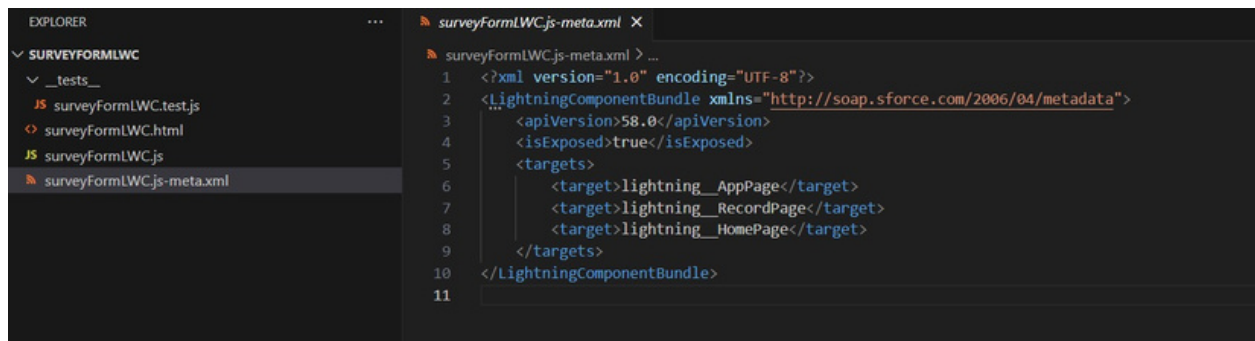
</div>

</lightning-card>

</template>

</lightning-card>

</template>



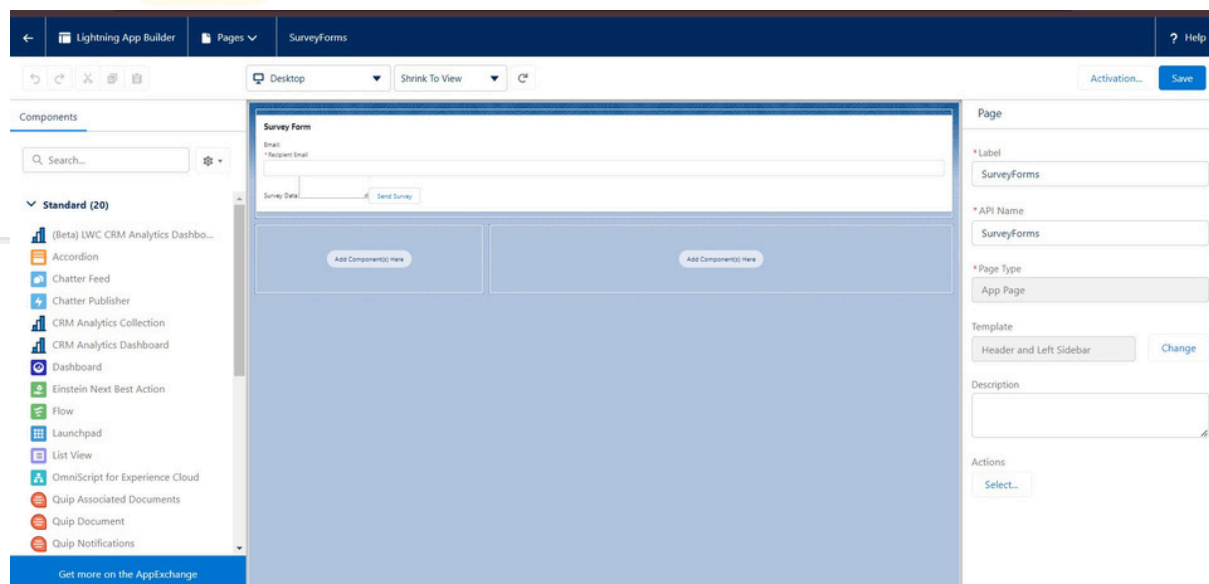
SurveyFormLWC(Xml) code:

```
<?xml version="1.0" encoding="UTF-8"?>
```

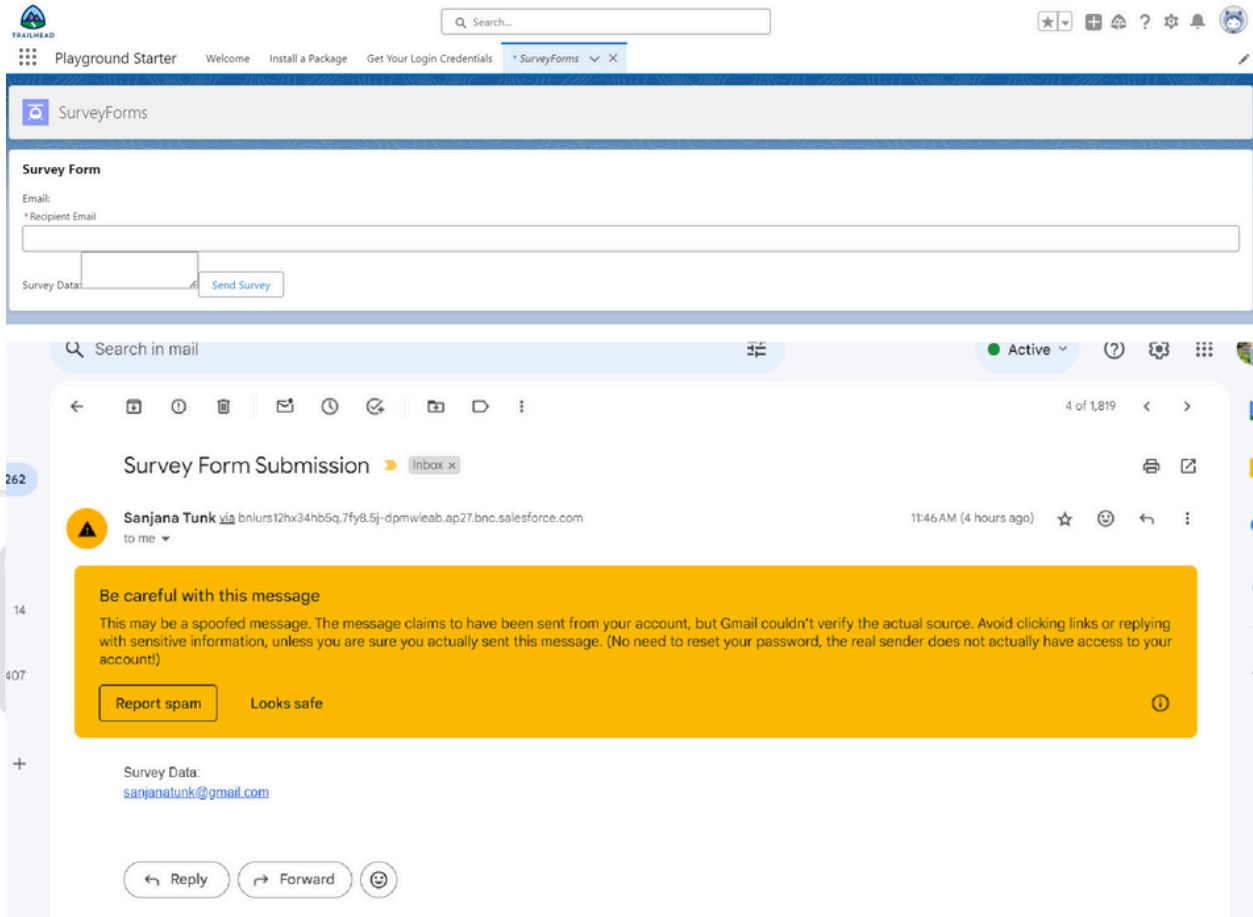
```
<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
  <apiVersion>58.0</apiVersion>
  <isExposed>true</isExposed>
  <targets>
    <target>lightning__AppPage</target>
    <target>lightning__RecordPage</target>
    <target>lightning__HomePage</target>
  </targets>
</LightningComponentBundle>
```

Deploy the source code.

Drag the custom component on the App page



Save and activate.



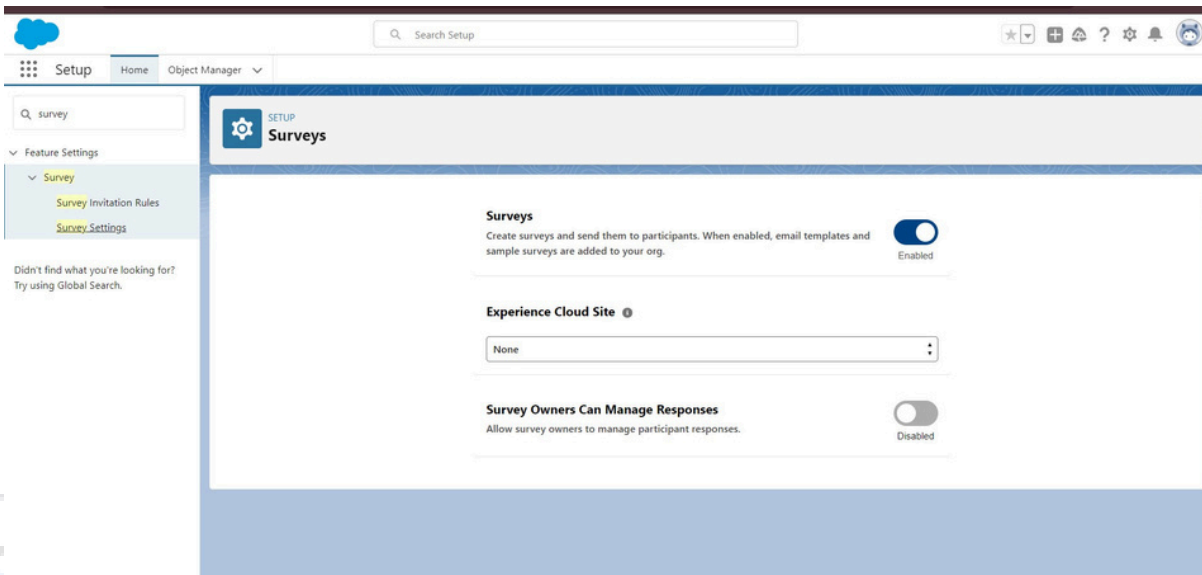
Milestone 4 : SurveyFormLink

Activity -1

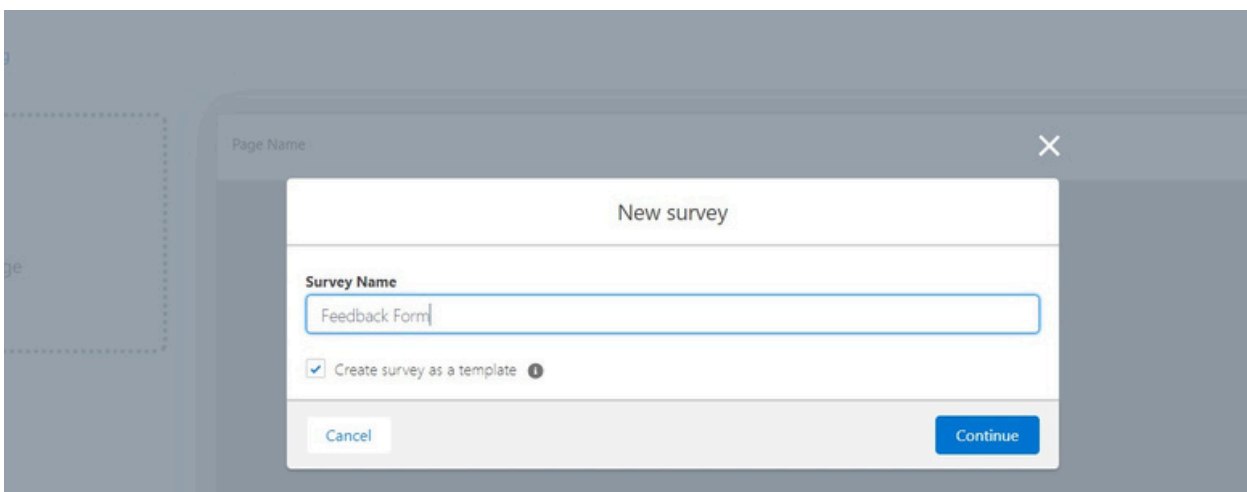
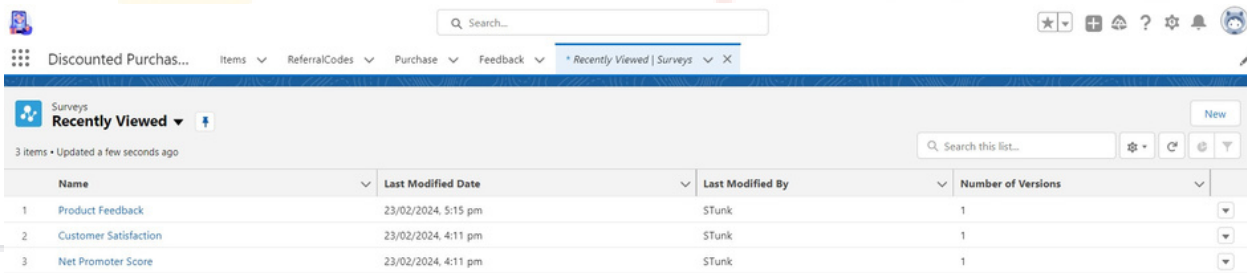
Create a Form for Survey

To create a custom object, follow these steps :

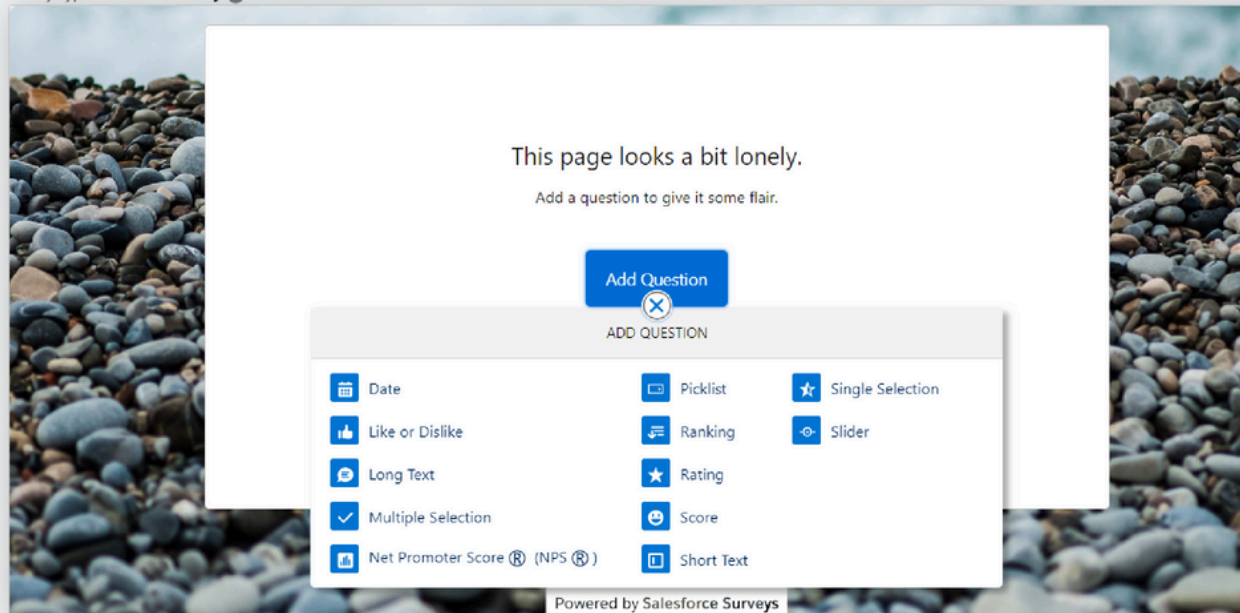
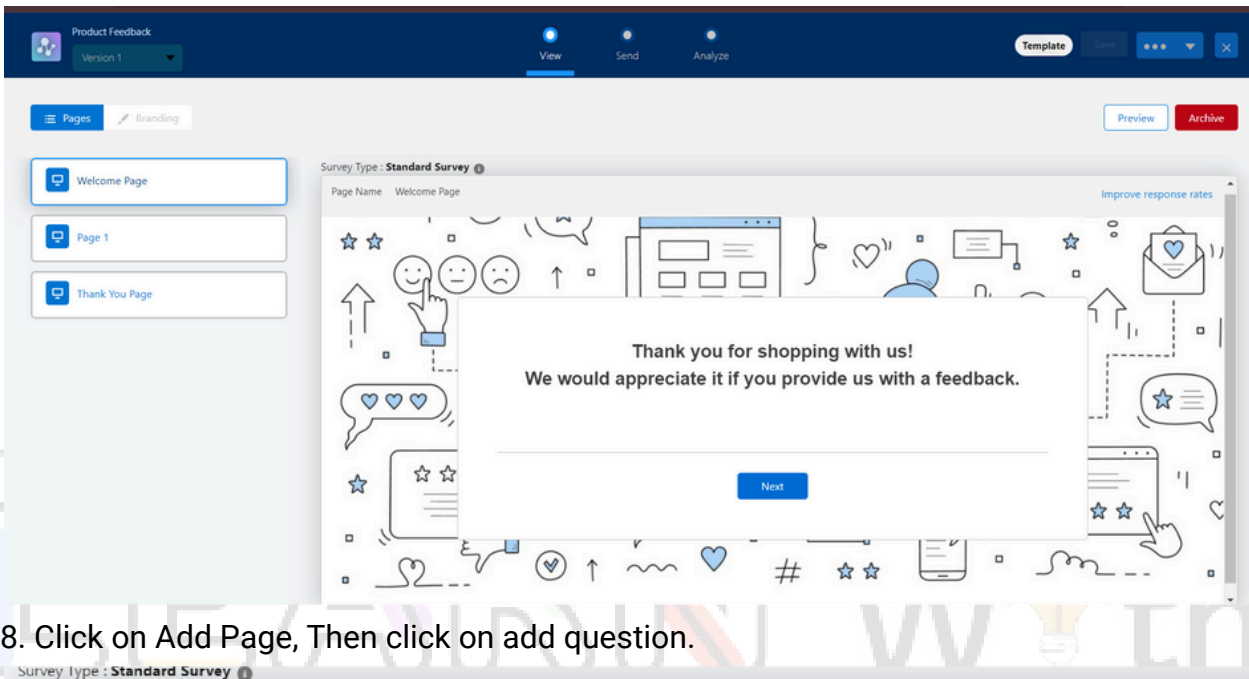
1. Click on the Gear Icon → From setup.
2. Click quick find, Survey settings.



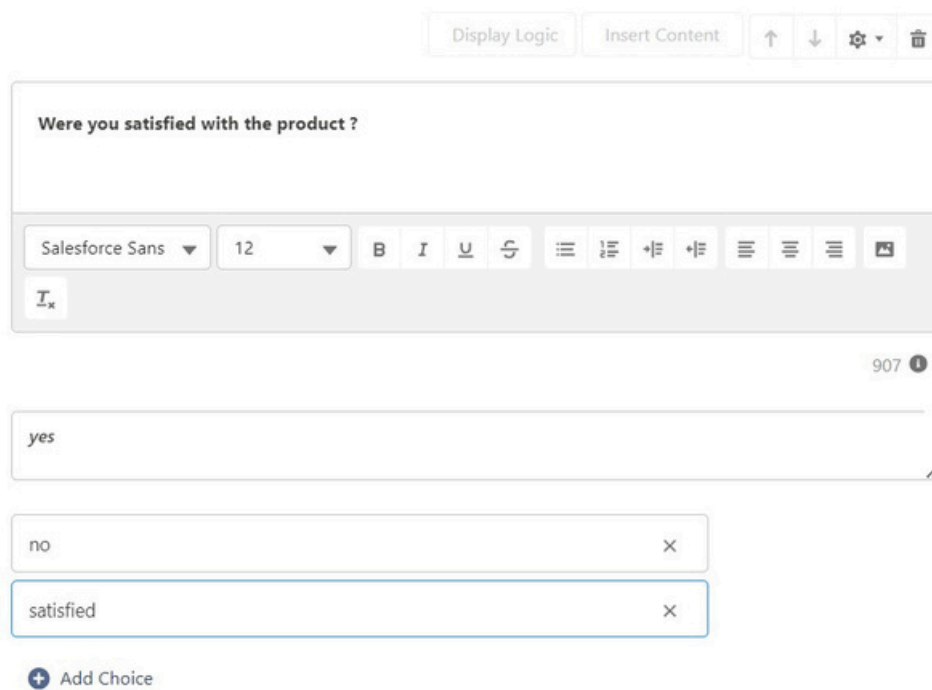
3. Click the App Launcher.
4. Search for "Survey".
5. Click on New to create one.



6. Fill the Survey Name as "Feedback Form".
7. Click on Create survey as a Template, then continue.

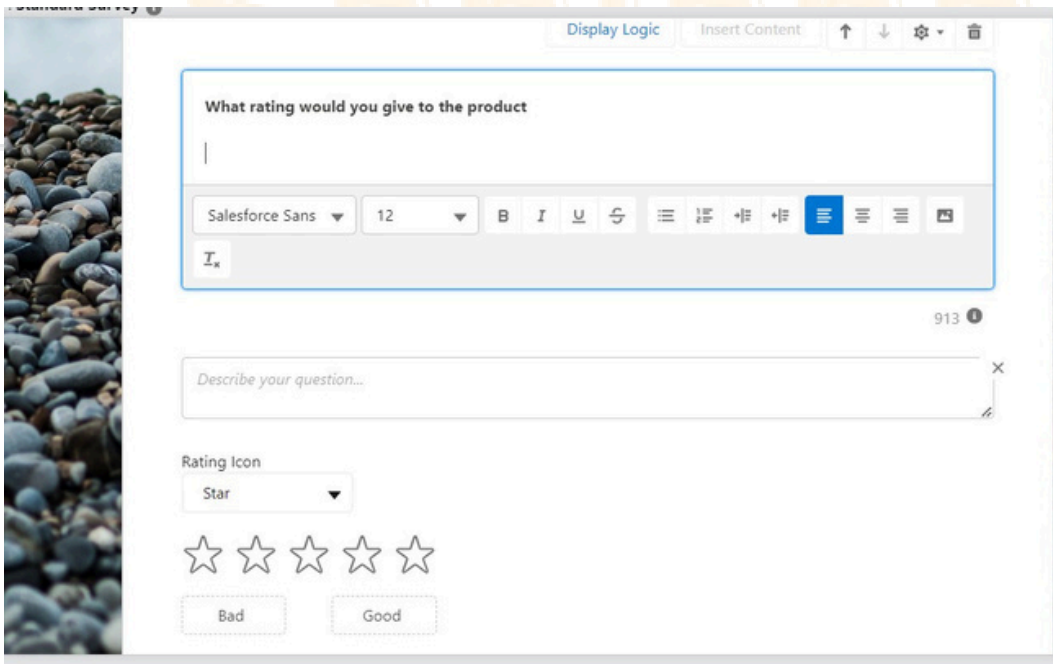


9. Choose multiple selection then enter your desired question



The screenshot shows a survey editor interface. At the top, there are buttons for "Display Logic" and "Insert Content", along with icons for up, down, settings, and delete. The main question area contains the text "Were you satisfied with the product ?". Below the question, there is a rich text editor with a font dropdown set to "Salesforce Sans", a size dropdown set to "12", and various formatting icons (bold, italic, underline, link, bulleted list, numbered list, indent, outdent, align left, align center, align right, and image). A text input field below the editor contains the word "yes". Below this, there are two more text input fields, one containing "no" and another containing "satisfied", each with a close button (X). At the bottom, there is a button labeled "Add Choice".

10. Click Add question, Select Rating, define the question



The screenshot shows a survey editor interface for a rating question. At the top, there are buttons for "Display Logic" and "Insert Content", along with icons for up, down, settings, and delete. The main question area contains the text "What rating would you give to the product". Below the question, there is a rich text editor with a font dropdown set to "Salesforce Sans", a size dropdown set to "12", and various formatting icons (bold, italic, underline, link, bulleted list, numbered list, indent, outdent, align left, align center, align right, and image). A text input field below the editor contains the text "Describe your question...". Below this, there is a "Rating Icon" dropdown set to "Star". Below the dropdown, there are five star icons. At the bottom, there are two buttons labeled "Bad" and "Good".

11. Click Add question, Select Long Text, define the question.

Please share your shopping experience with us, What could we do to make it better?

Enter answer...

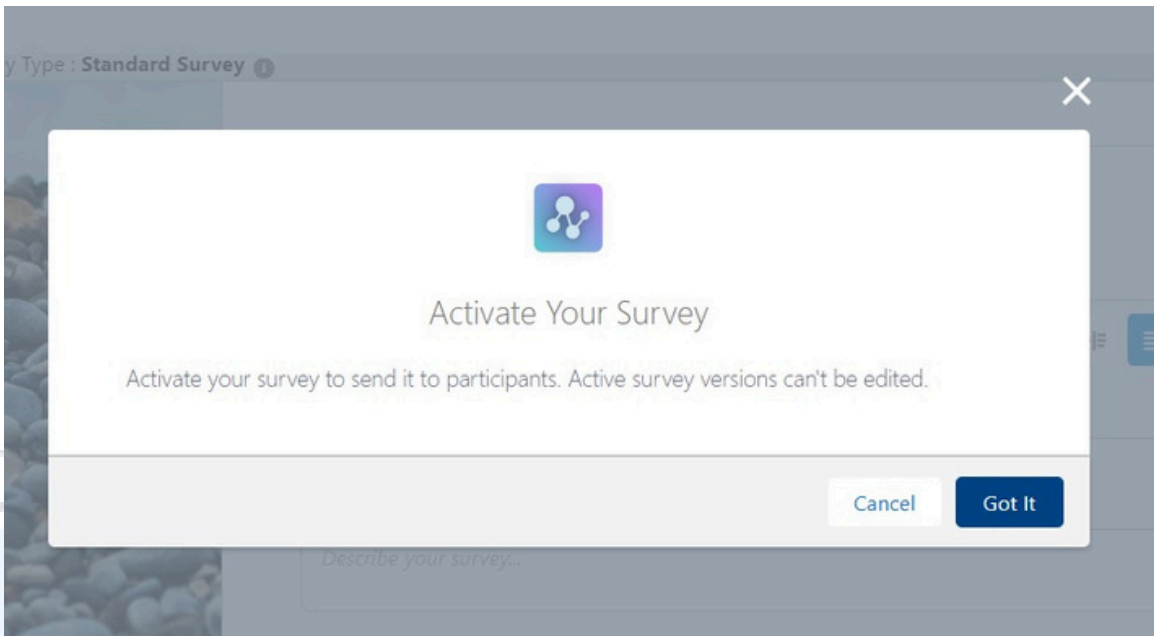
Previous

Finish

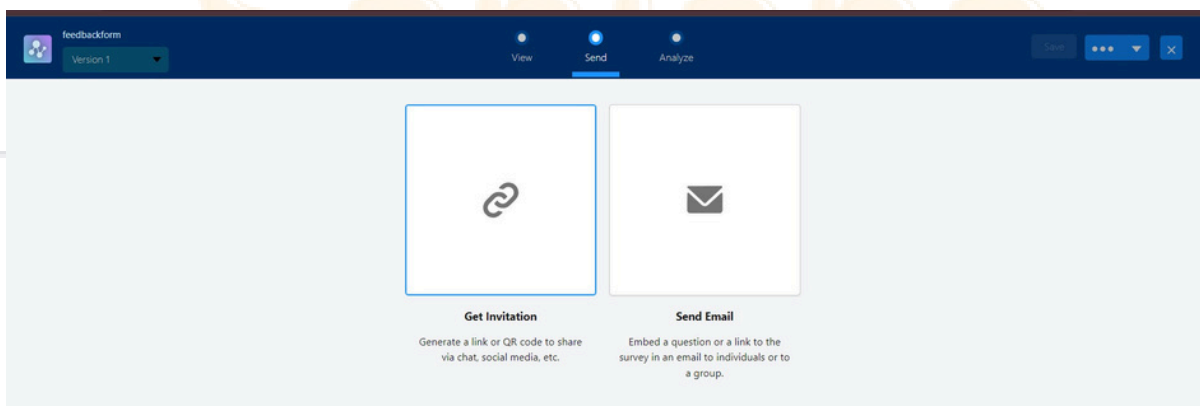
12. Click on Add Page, Then add a thankyou message.

The screenshot shows a survey editor interface. At the top right, there are 'Preview' and 'Activate' buttons. The survey title is 'Standard Survey'. The main content area shows a 'Thank You for your feedback.' message being added. Below the message, there is a rich text editor with a toolbar containing options for font (Salesforce Sans), size (12), bold (B), italic (I), underline (U), strikethrough (ABC), bulleted list, numbered list, link, unlink, indent, outdent, and image. A text area below the toolbar contains the text 'Enter additional information...'. The page number '948' is displayed at the bottom right of the editor.

13. Activate the survey.



14. Click Send, You can either copy the link or send it through mail. And then click on Analyze to check the response received.



Send Email

Embed a survey link or a question in an email and get a preview before you send it. Pro Tip: Embedding a question improves response rates.

Send Email

Add recipients to send survey. Pro Tip: Select list of recipients to mass send surveys.

Send to Individuals

Send the invitation to individuals by manually entering email addresses.

Send to List

Send survey invitations to recipients on contact and lead lists and members of a campaign.

Previous

Next

View

Send

Analyze

Get Survey Link

Participants in Your Company

Participants Outside Your Company

Send this link to survey participants within your company.

https://sbcom-be-dev-ed.develop.my.salesforce.com/survey/runtimeApp?invitationId=0KIS000000fyCo&surveyName=feedbackform&UUID=292d6778-d3c8-4b1d-a6f3-fbb55c017357

☐ Auto-Expires

☐ Anonymize responses

☐ Let participants see their responses

Download QR Code

Copy Link

feedbackform

Version 1

View

Send

Analyze

Survey Metrics

Export Responses

Refresh

Complete or In Progress Responses

0

We can't draw this chart because there is no data.

Response Dashboard

Select Survey Page

Page 1

Showing 2 of 2 questions

Were you satisfied with the product ?

What rating would you give to the product.

Sit tight! No one has responded to this question yet.

Sit tight! No one has responded to this question yet.