SMART SLA TRACKER: INTELLIGENT CUSTOMER COMPLAINT & ESCALATION MANAGEMENT
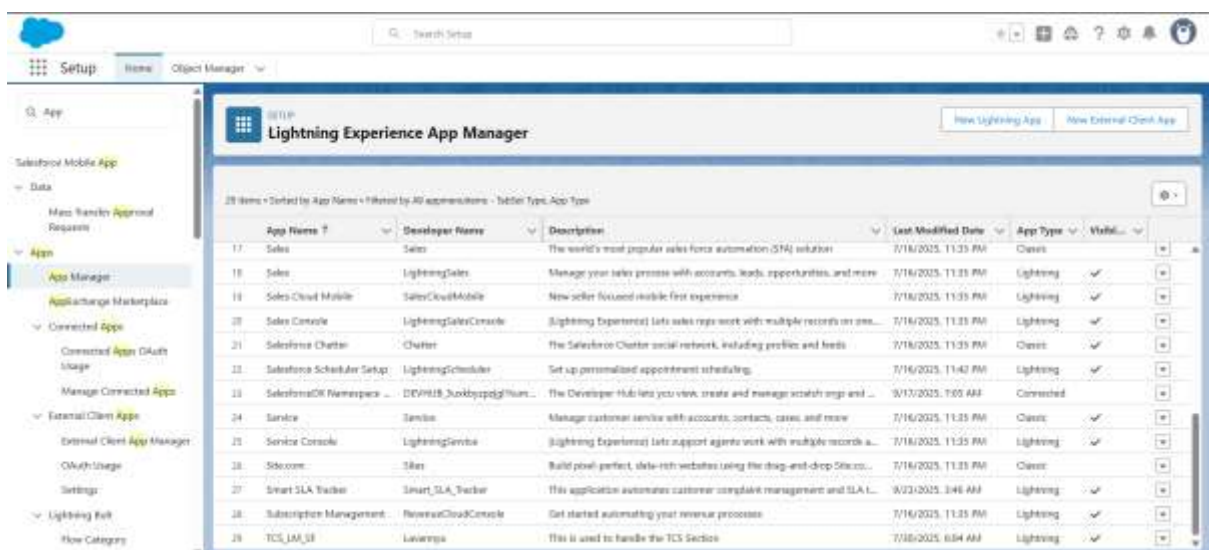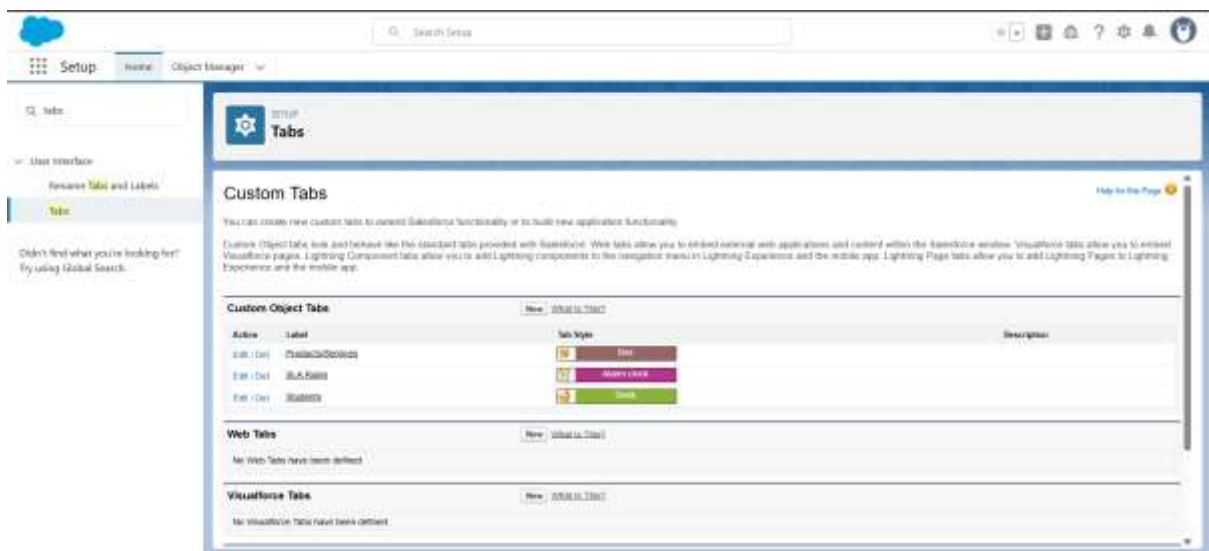
Phase 6: User Interface Development

6.1 Lightning App & Home Page:

Setup > App Manager > New Lightning App > App Name > as Smart SLA Tracker

• Create custom object Tabs > SLA Rule and Product/Service.

Setup > Tabs > "Custom Object Tabs" > New.

1. Give it a App Name (Smart SLA Tracker) and Choose a logo

2. Choose App options.

3. Add Utility Items

4. In the Navigation Items, add the following tabs : Home, Cases, SLA Rules, Products/Services
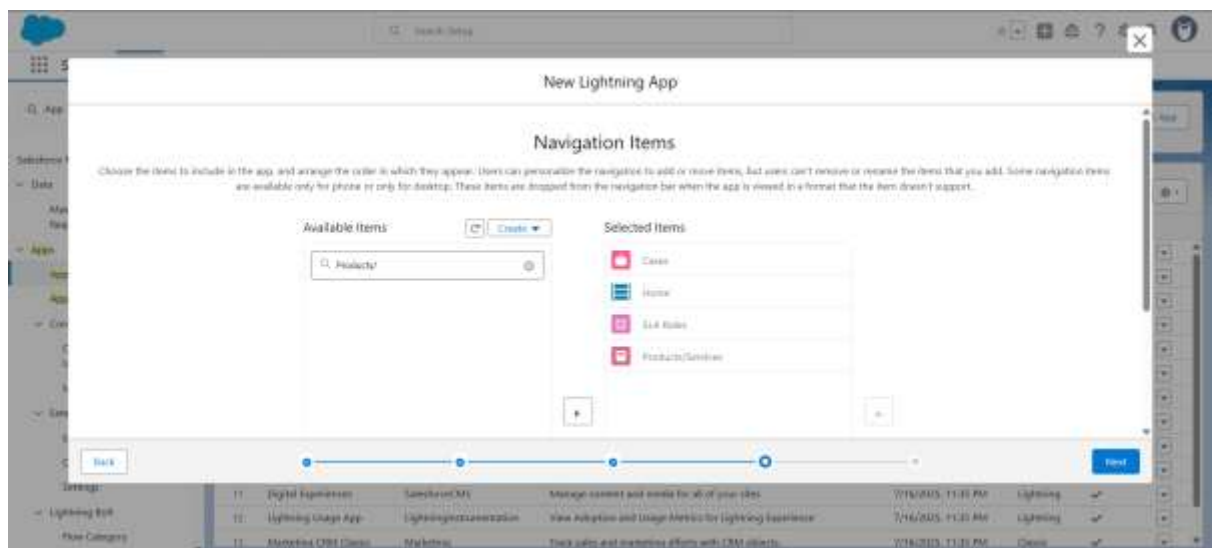
6.2 Customize the Home Page:

App Launcher > Smart SLA Tracker > Gear icon > Edit page

Add and Configure Components:
1. Assistant
2. Report Chart
3. Recent Items
4. Today's Events and Tasks
5. Rich Text

• Save and activate:

Save > activate > App and Profile > Smart SLA Tracker > SLA Support Agent and SLA Support Manager > save

6.3 Building a Lightning Web Component (LWC):

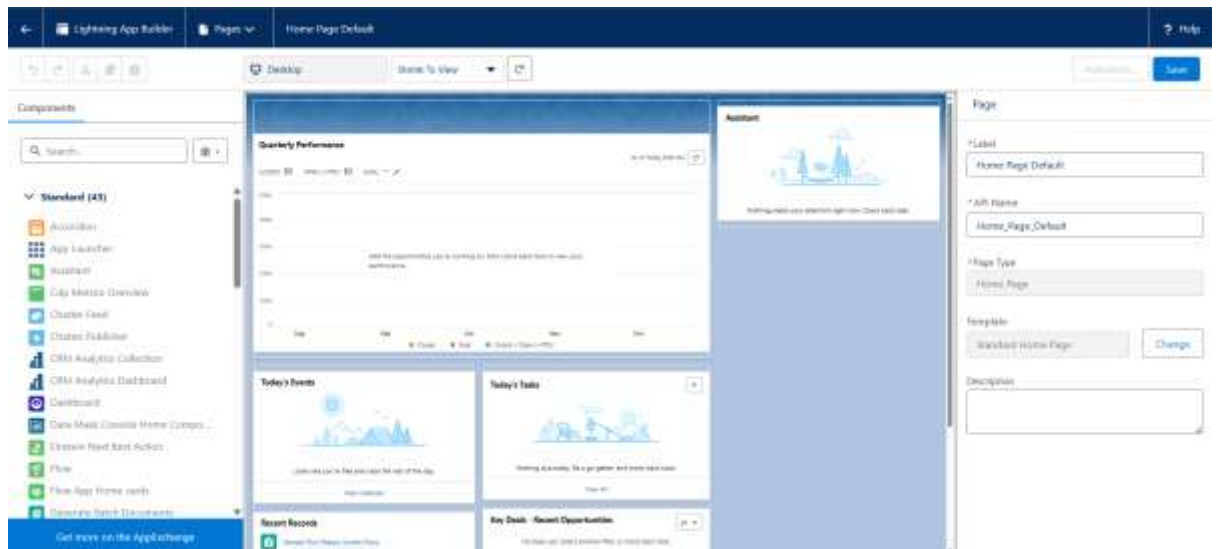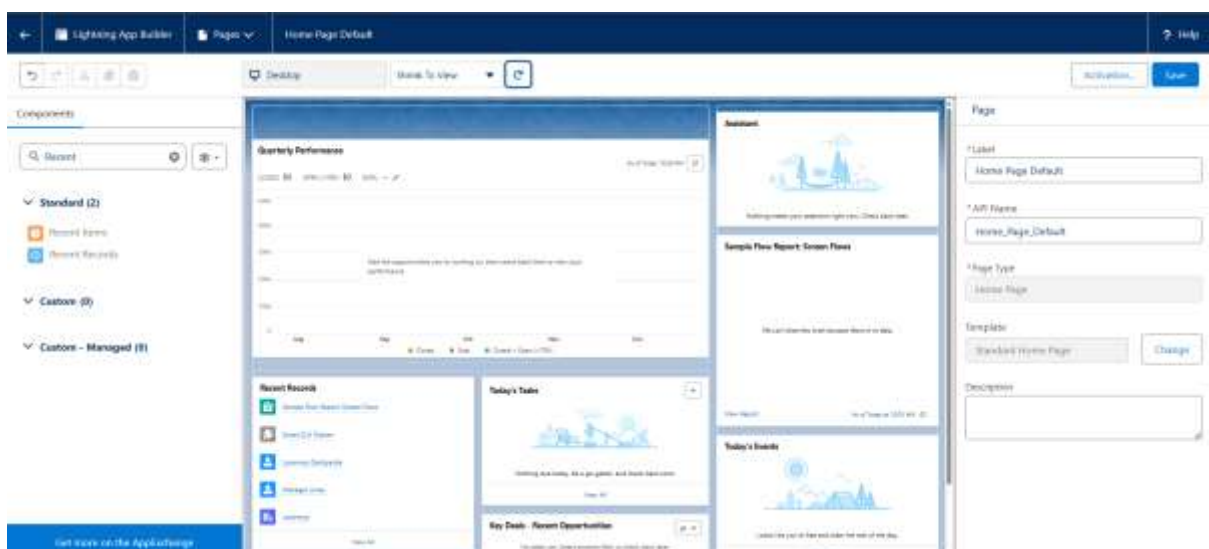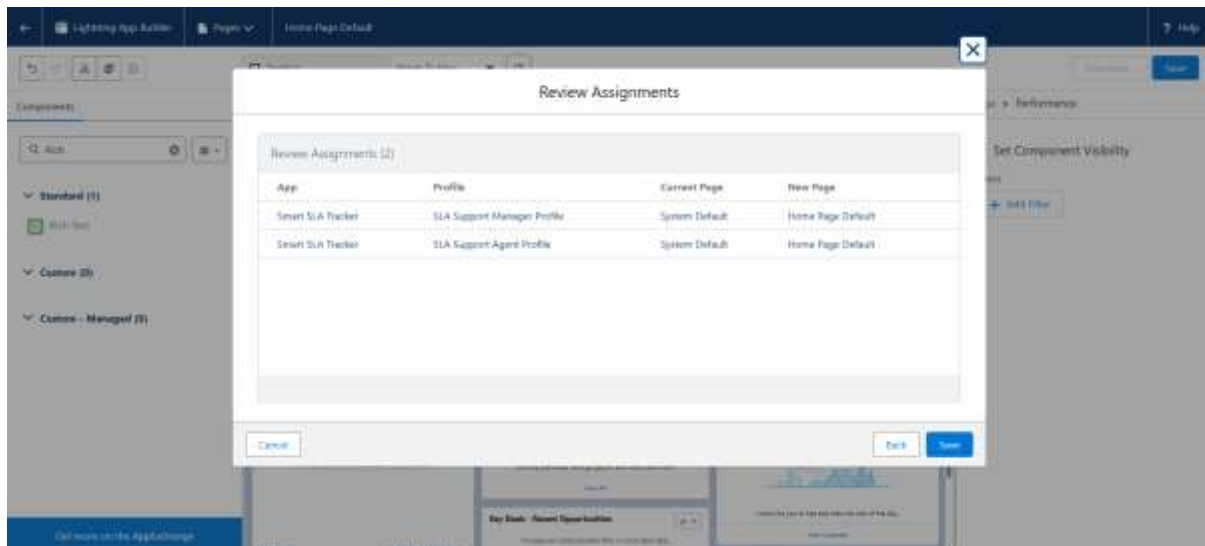**Vs Code > SFDX: Create Lightning Web Component > slaStatusIndicator > slaStatusIndicator.html, slaStatusIndicator.js, and slaStatusIndicator.js-meta.xml**

slaStatusIndicator.html

```
force-app > main > default > lwc > slaStatusIndicator > slaStatusIndicator.html > ...
   1   <template>
   2       <lightning-card title="SLA Status">
   3           <div class="slds-p-around_medium" style={statusStyle}>
   4               <p>SLA Due Date: {slaDueDate}</p>
   5           </div>
   6       </lightning-card>
   7   </template>
   8
```

slaStatusIndicator.js

slaStatusIndicator.js-meta.xml

- **Deploy the code to org**
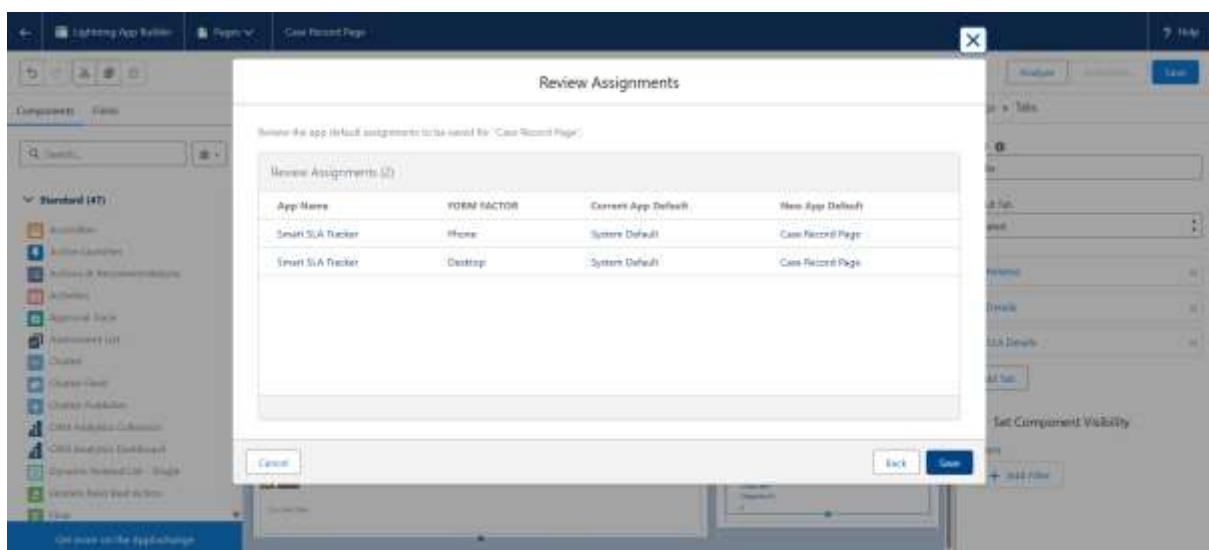
6.4 Customize the Case Record Page:

App Launcher > Smart SLA Tracker > Case > Any object > Gear icon > Edit page

1. Add the Highlights Panel
2. Add a new Tab Component
3. Add the Custom LWC > **SLA Details**
4. Add Field: **SLA Due Date**
5. caseMetrics
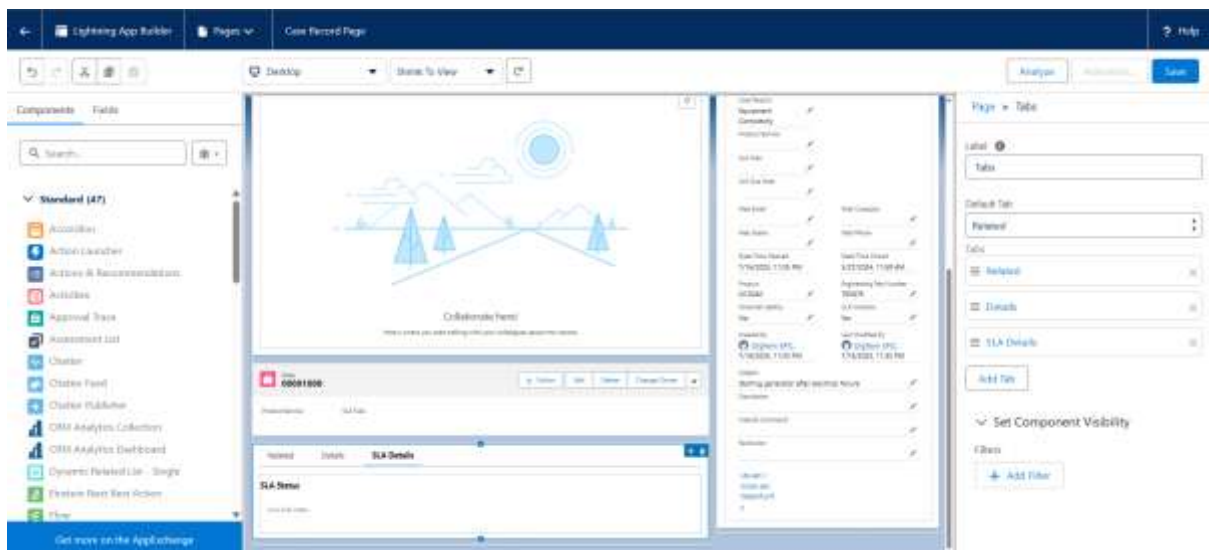
Save and Activate:

Choose App Default as Smart SLA Tracker

6.5 The Lightning Web Component (LWC):

CaseDataService.cls

```apex
public with sharing class CaseDataService {
    @AuraEnabled(cacheable=true)
    public static Integer getHighPriorityCaseCount() {
        return [SELECT count() FROM Case WHERE Priority = 'High'];
    }
}
```

- **caseMetrics**

caseMetrics.html

```html
<template>
    <lightning-card title="Case Metrics">
        <div class="slds-p-around_medium">
            <lightning-button label="Get Case Metrics" onclick={getCaseCount}></lightning-button>
            <p>High Priority Cases: {highPriorityCount}</p>
        </div>
    </lightning-card>
</template>
```

caseMetrics.js

```javascript
import { LightningElement, track } from 'lwc';
import getHighPriorityCaseCount from '@salesforce/apex/CaseDataService.getHighPriorityCaseCount';

export default class CaseMetrics extends LightningElement {
    @track highPriorityCount = 0;

    getCaseCount() {
        getHighPriorityCaseCount()
            .then(result => {
                this.highPriorityCount = result;
            })
            .catch(error => {
                console.error(error);
            });
    }
}
```

caseMetrics.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
    <apiVersion>55.0</apiVersion>
    <isExposed>true</isExposed>
    <targets>
        <target>lightning__HomePage</target>
    </targets>
</LightningComponentBundle>
```

Deployment :


```
Created  CaseDataService  ApexClass  force-app\main\default\classes\CaseDataService.cls-meta.xml

15:39:16.615 Ended SFDX: Deploy This Source to Org
15:39:28.272 Starting SFDX: Deploy This Source to Org

--- Deployed Source
STATE     FULL NAME     TYPE                     PROJECT PATH

Created   caseMetrics   LightningComponentBundle  force-app\main\default\lwc\caseMetrics\caseMetrics.html
Created   caseMetrics   LightningComponentBundle  force-app\main\default\lwc\caseMetrics\caseMetrics.js
Created   caseMetrics   LightningComponentBundle  force-app\main\default\lwc\caseMetrics\caseMetrics.js-meta.xml

15:39:30.811 Ended SFDX: Deploy This Source to Org
```

6.6 Add a Utility Bar:

History, notes