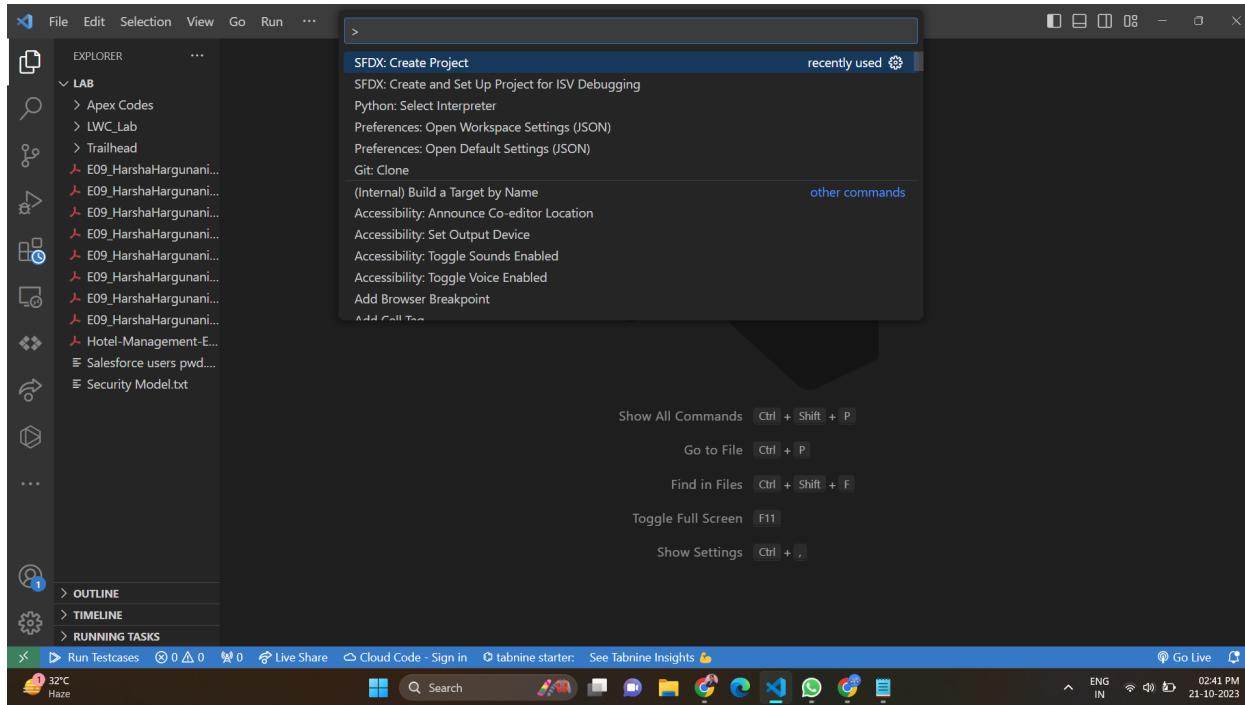


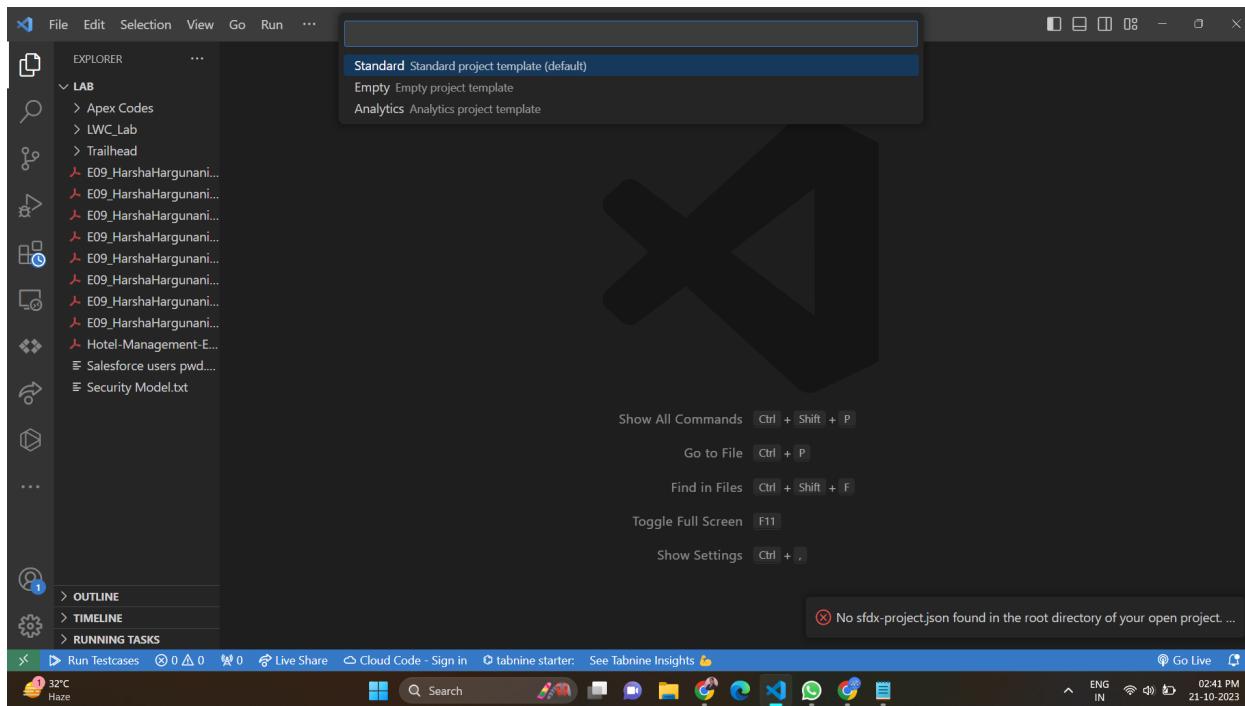
CRM Practical 8

Name. Harsha Hargunani
Batch. E1
Roll no. 09

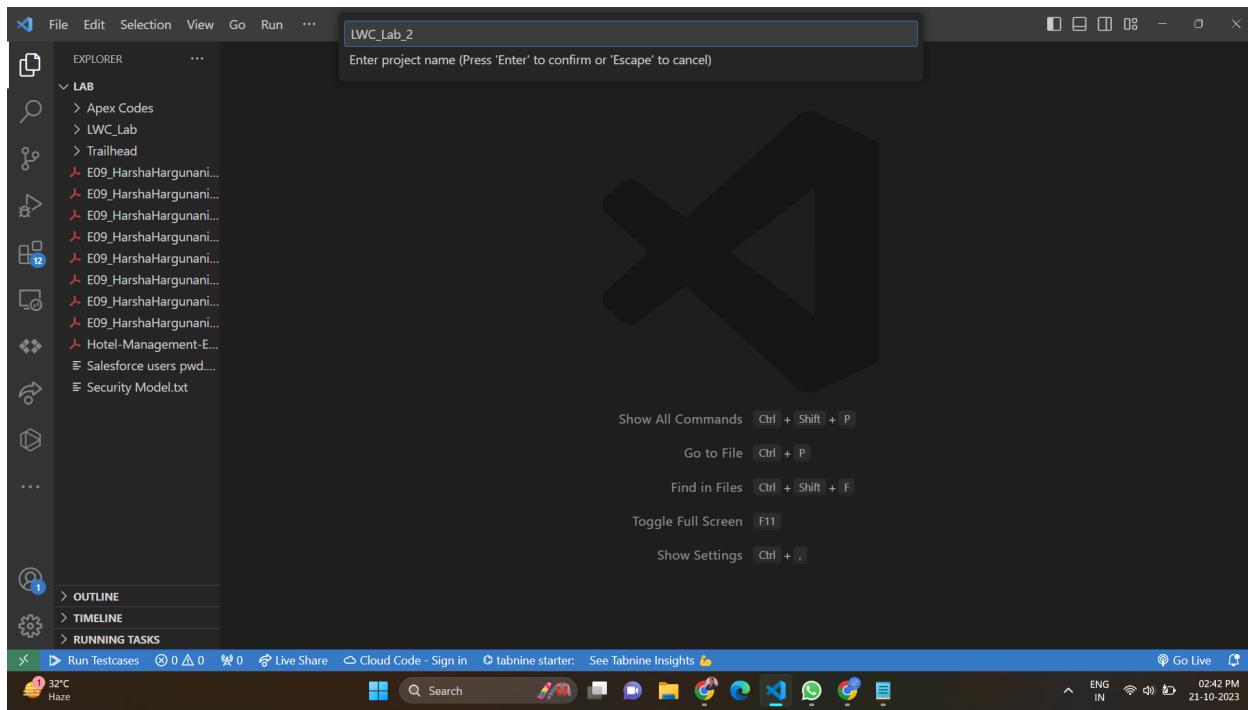
Step1: Open Command Palette by pressing Ctrl+Shift+P and type SFDX. Select Create new Project Option.



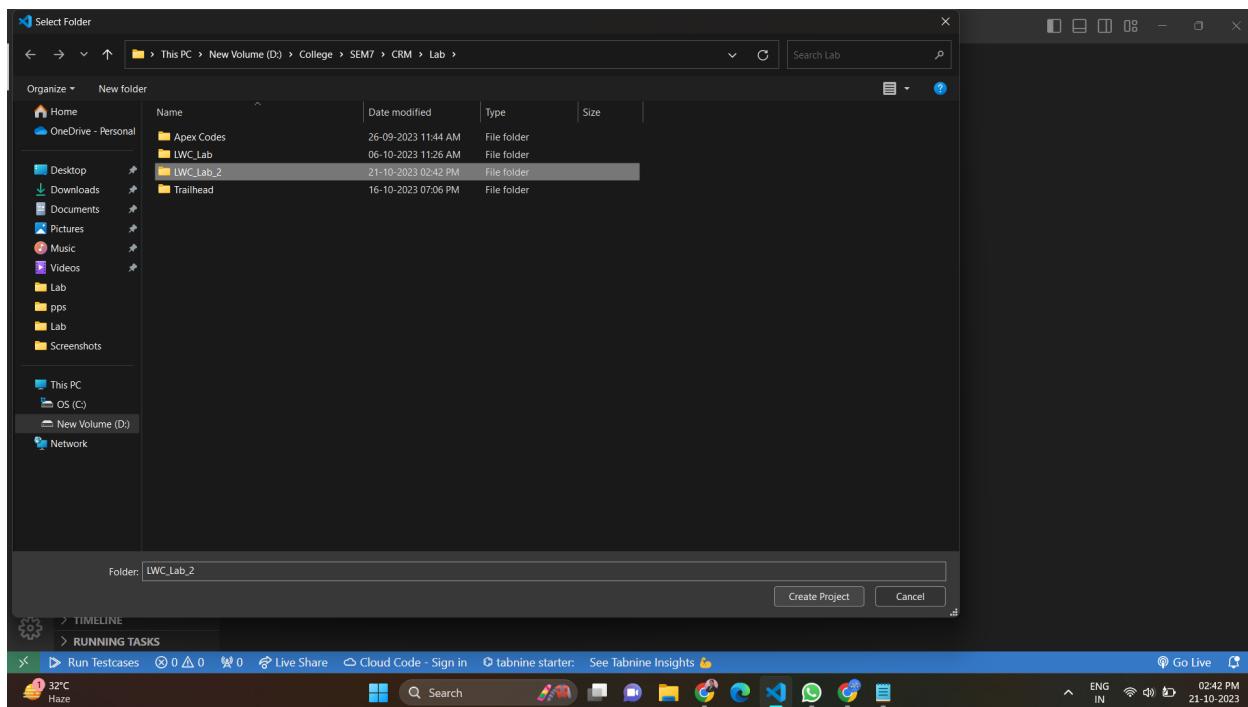
Step2: Choose Standard Project



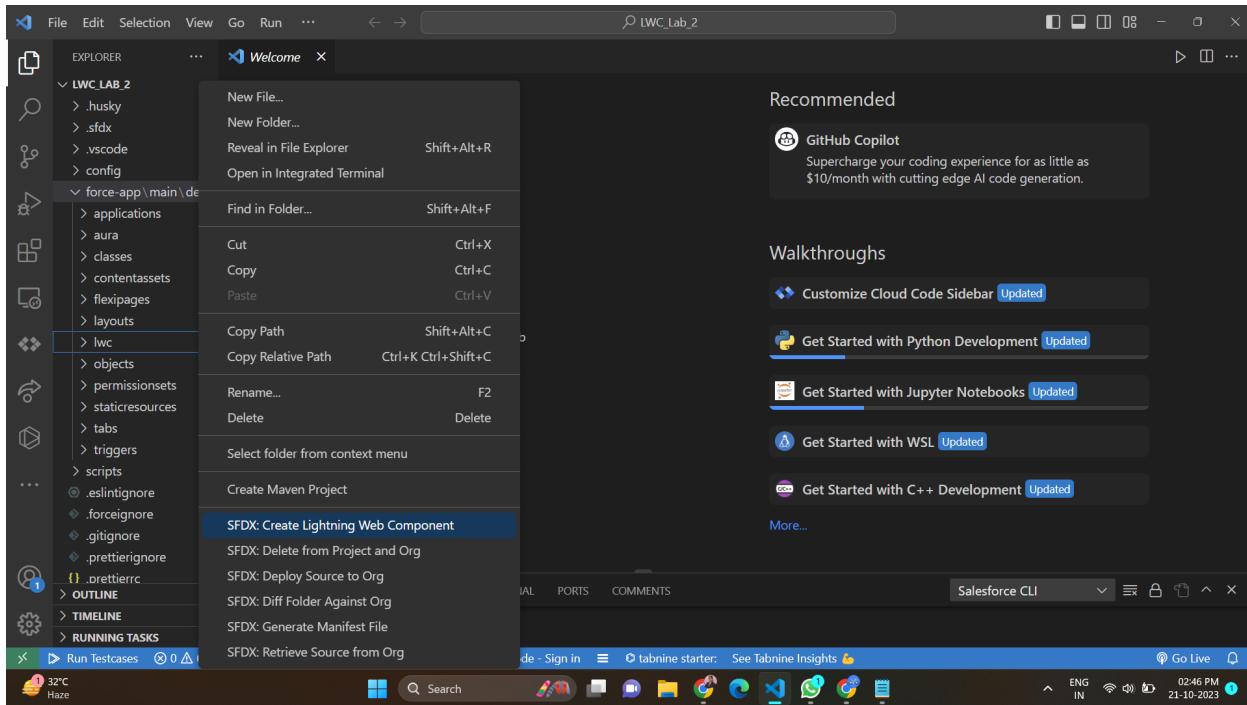
Step3: Enter Project Name - LWC_Lab_2



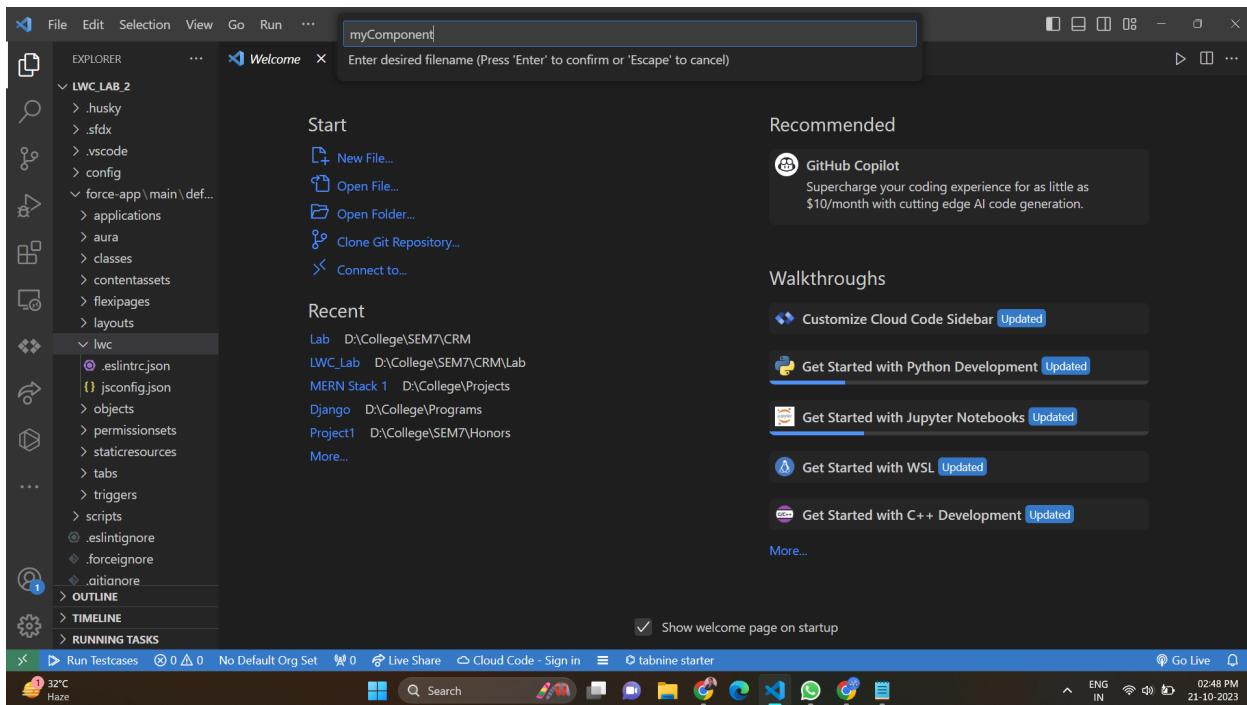
Step4: Select a folder to create the new project.



Step5: Under force-app\main\default\lwc, right click and select Create Lightning Web Component.



Step6: Enter name of the component - myComponent



Step7: Enter the code for html, js and xml files.

myComponent.html

```
<template>
  <lightning-card>
    <lightning-input data-formfield = "accountName" type = "text" label =
    "Enter Account Name" name="accName"></lightning-input>
    <lightning-input data-formfield = "accountDescription" type = "text" label
    = "Enter Description" name="accDescription"></lightning-input>
    <lightning-button variant="brand" label="Submit"
    onclick={saveButtonHandler}></lightning-button>
  </lightning-card>
</template>
```

myComponent.js

```
import { LightningElement } from 'lwc';

import createAccountRecord from
"@salesforce/apex/AccountProvider.createAccountRecord";

export default class MyComponent extends LightningElement {
  objAccount = { 'sobjectType' : 'Account'}
  saveButtonHandler(){

this.objAccount.Name=this.template.querySelector('lightning-input[data-formfield=
"accName"]').value;

this.objAccount.Type=this.template.querySelector('lightning-input[data-formfield=
"accDescription"]').value;
  createAccountRecord({objAcc:this.objAccount})
  .then((result) =>{
    this.result=result;
    this.error=undefined;
  }).catch((error) =>{
    this.error = error;
    this.result = undefined;
  });
}
```

```
}
```

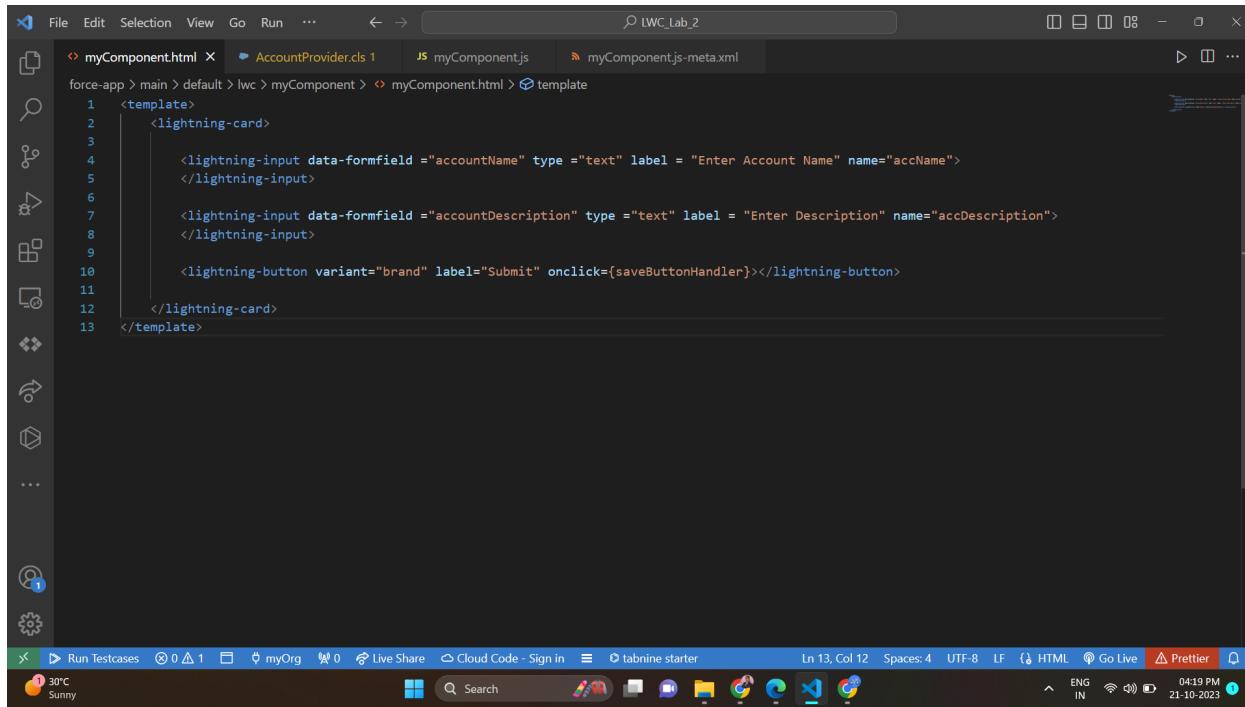
myComponent.js-meta.xml

```
<?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__RecordAction</target>
        <target>lightning__RecordPage</target>
    </targets>

</LightningComponentBundle>
```

Screenshots -



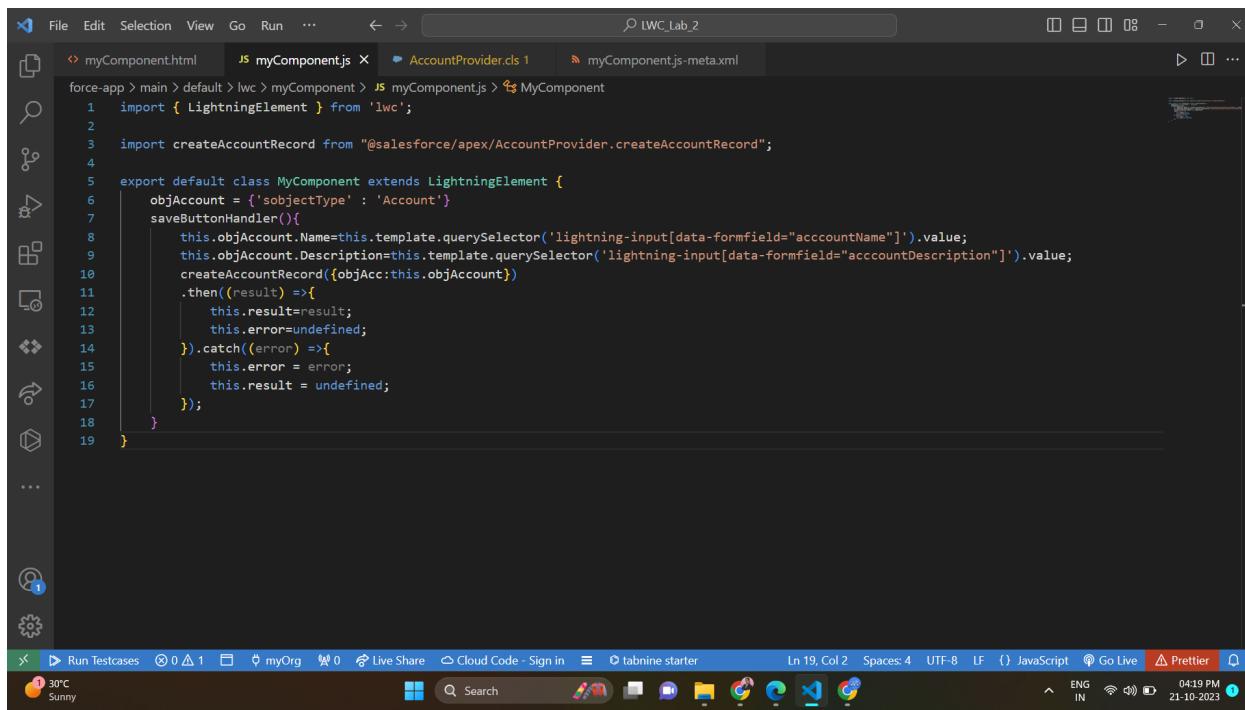
The screenshot shows the Lightning Dev Hub interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** LWC_Lab_2
- Project Structure:** force-app > main > default > lwc > myComponent > myComponent.html > template
- Code Editor:** myComponent.js-meta.xml (highlighted)
- Code Content:**

```
<?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__RecordAction</target>
        <target>lightning__RecordPage</target>
    </targets>

</LightningComponentBundle>
```
- Toolbars and Icons:** Includes icons for file operations, search, and various developer tools.
- Bottom Status Bar:** Shows weather (30°C Sunny), search bar, and system status (ENG IN, 04:19 PM, 21-10-2023).



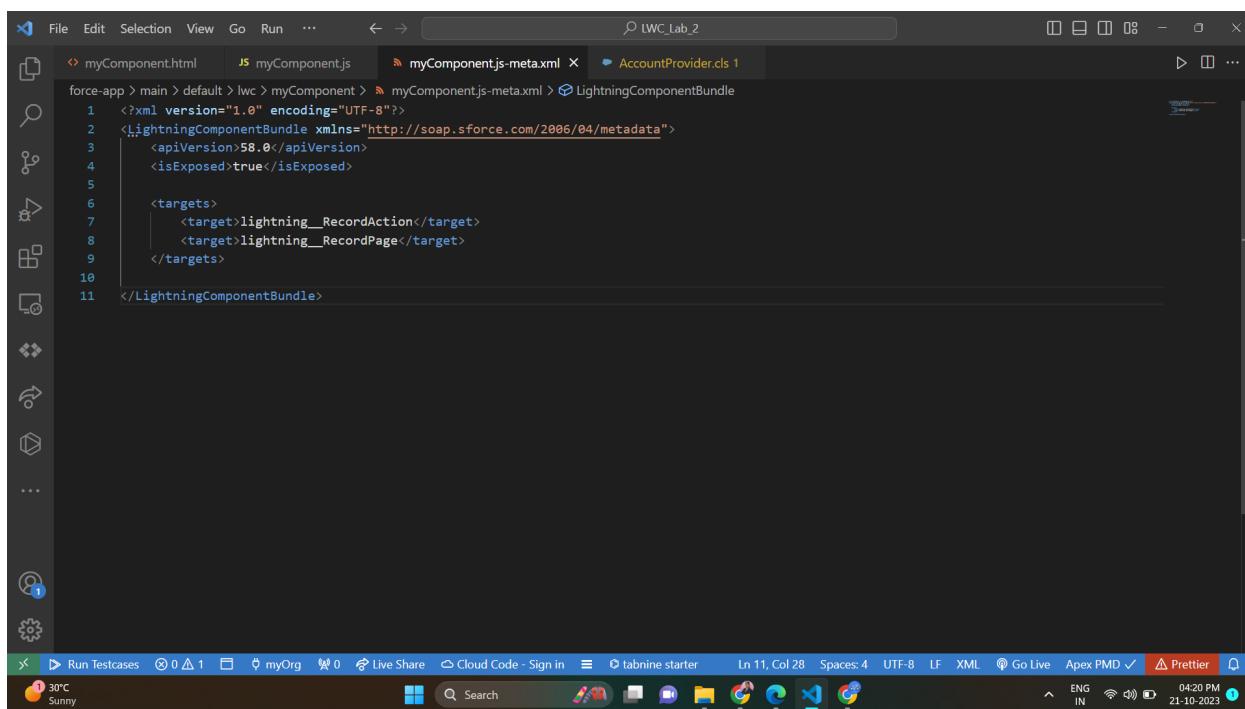
File Edit Selection View Go Run ... ← → ⌘ LWC_Lab_2

myComponent.html myComponent.js AccountProvider.cls 1 myComponent.js-meta.xml

```
force-app > main > default > lwc > myComponent > myComponent.js > MyComponent
1 import { LightningElement } from 'lwc';
2
3 import createAccountRecord from '@salesforce/apex/AccountProvider.createAccountRecord';
4
5 export default class MyComponent extends LightningElement {
6     objAccount = {'sobjectType': 'Account'}
7     saveButtonHandler(){
8         this.objAccount.Name=this.template.querySelector('lightning-input[data-formfield="accconutName"]').value;
9         this.objAccount.Description=this.template.querySelector('lightning-input[data-formfield="accconutDescription"]').value;
10        createAccountRecord({objAcc:this.objAccount})
11        .then((result) =>{
12            this.result=result;
13            this.error=undefined;
14        }).catch((error) =>{
15            this.error = error;
16            this.result = undefined;
17        });
18    }
19 }
```

Run Testcases 30°C Sunny Live Share Cloud Code - Sign in tabnine starter JavaScript Go Live Prettier

30°C Sunny 04:19 PM 21-10-2023



File Edit Selection View Go Run ... ← → ⌘ LWC_Lab_2

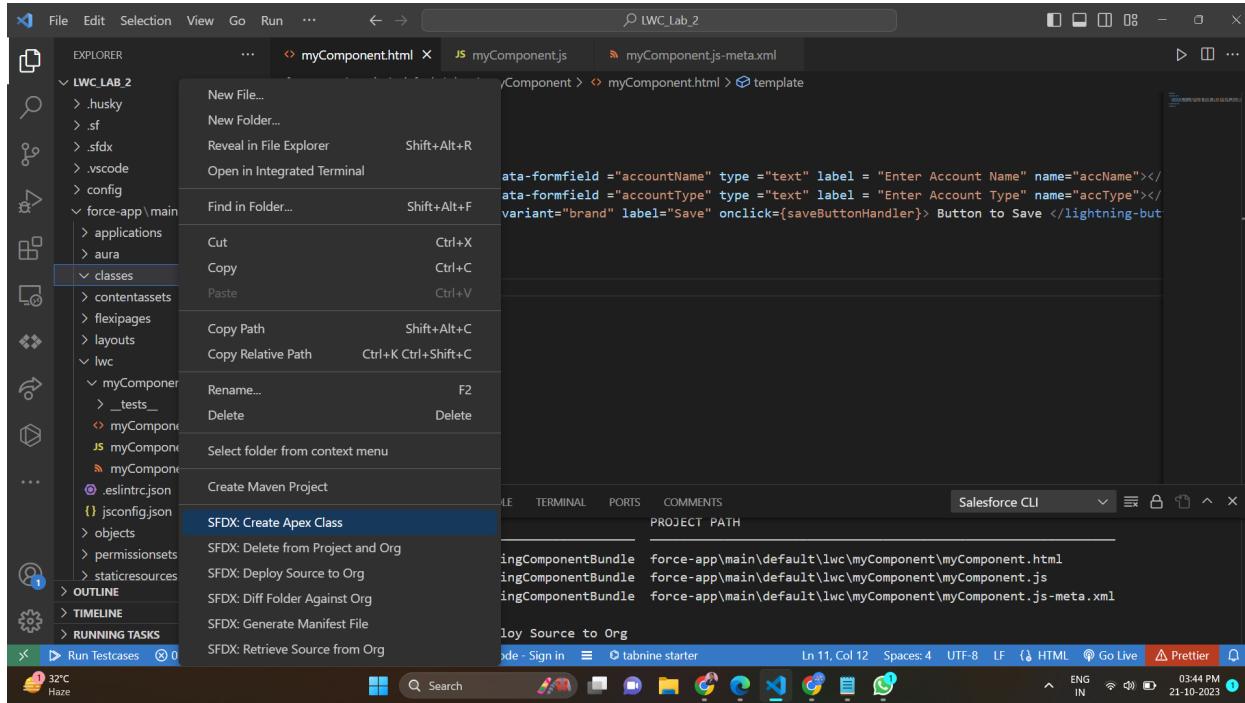
myComponent.html myComponent.js myComponent.js-meta.xml AccountProvider.cls 1

```
force-app > main > default > lwc > myComponent > myComponent.js-meta.xml > LightningComponentBundle
1 <?xml version="1.0" encoding="UTF-8"?>
2 <LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
3     <apiVersion>58.0</apiVersion>
4     <isExposed>true</isExposed>
5
6     <targets>
7         <target>lightning__RecordAction</target>
8         <target>lightning__RecordPage</target>
9     </targets>
10
11 </LightningComponentBundle>
```

Run Testcases 30°C Sunny Live Share Cloud Code - Sign in tabnine starter Ln 11, Col 28 Spaces: 4 UTF-8 LF XML Go Live Apex PMD Prettier

30°C Sunny 04:20 PM 21-10-2023

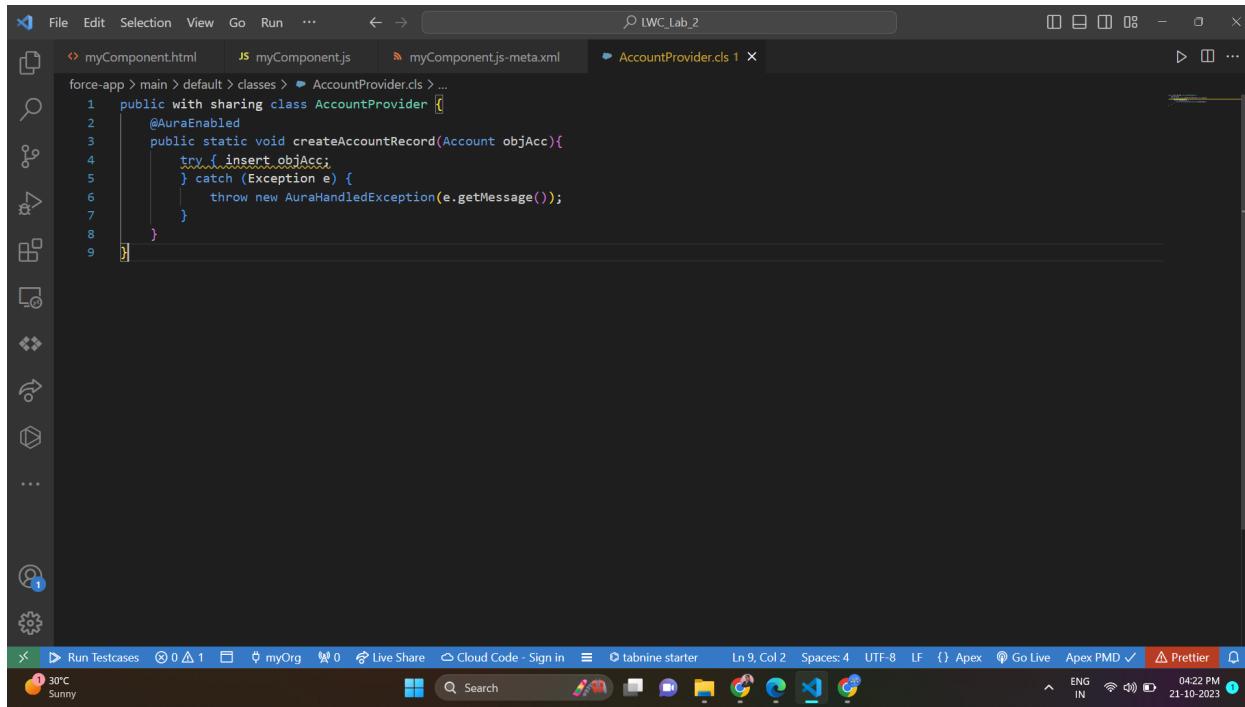
Step8: Under classes, right click and create a new apex class titled AccountProvider and enter the code.



AccoutProvider.cls

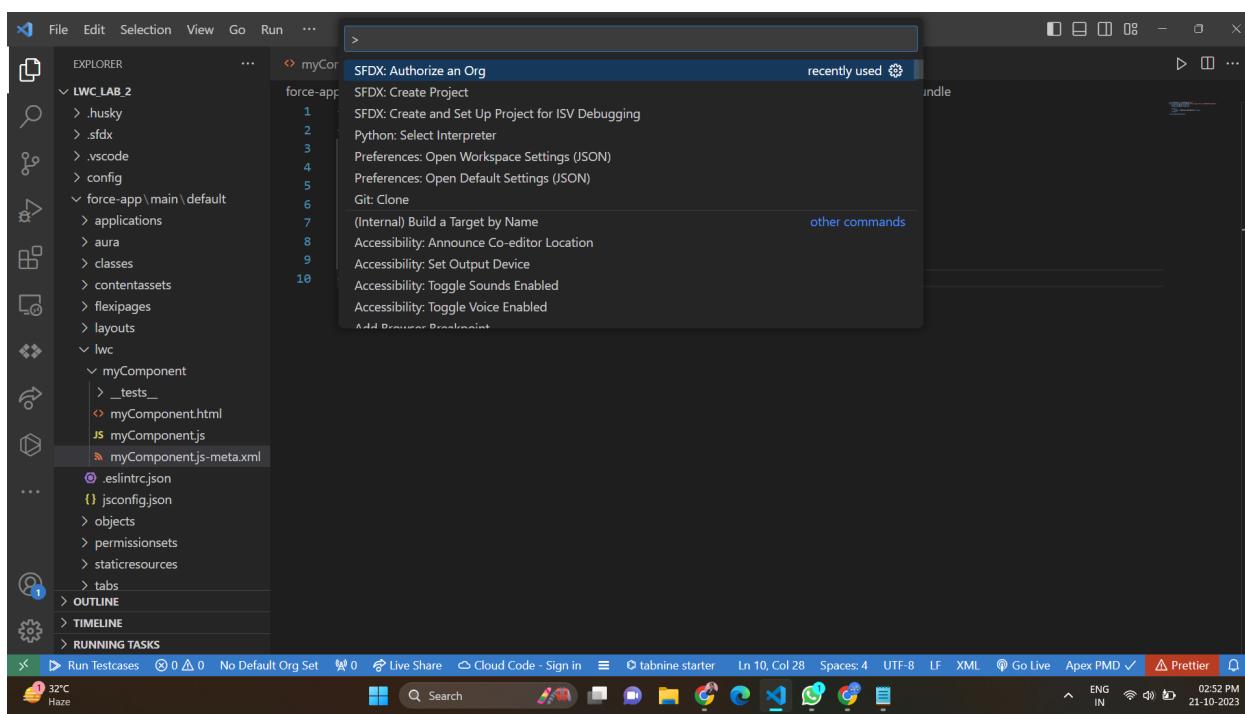
```
public with sharing class AccountProvider {
    @AuraEnabled
    public static void createAccountRecord(Account objAcc){
        try { insert objAcc;
        } catch (Exception e) {
            throw new AuraHandledException(e.getMessage());
        }
    }
}
```

Screenshot -

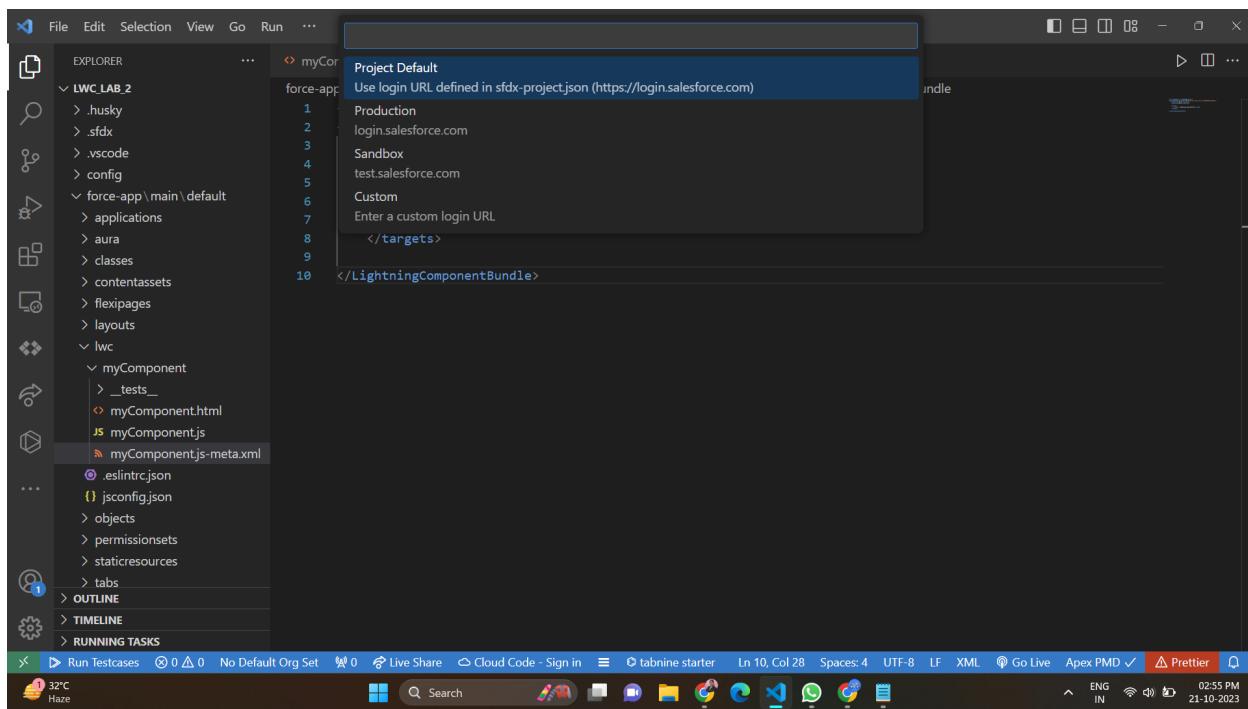


```
force-app > main > default > classes > AccountProvider.cls > ...
1  public with sharing class AccountProvider {
2      @AuraEnabled
3      public static void createAccountRecord(Account objAcc){
4          try{ insert objAcc;
5          } catch (Exception e) {
6              throw new AuraHandledException(e.getMessage());
7          }
8      }
9  }
```

Step9: Open Command Palette and type SFDX. Select Authorize an Org.



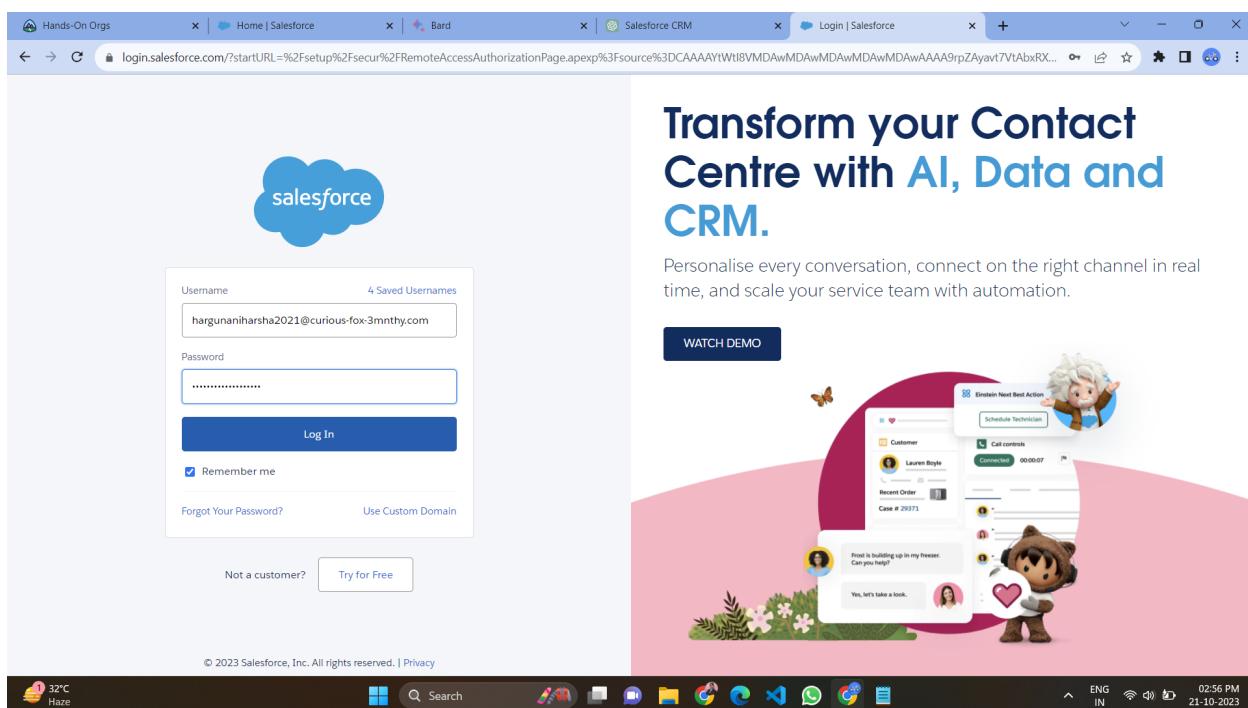
Step10: Select Project Default

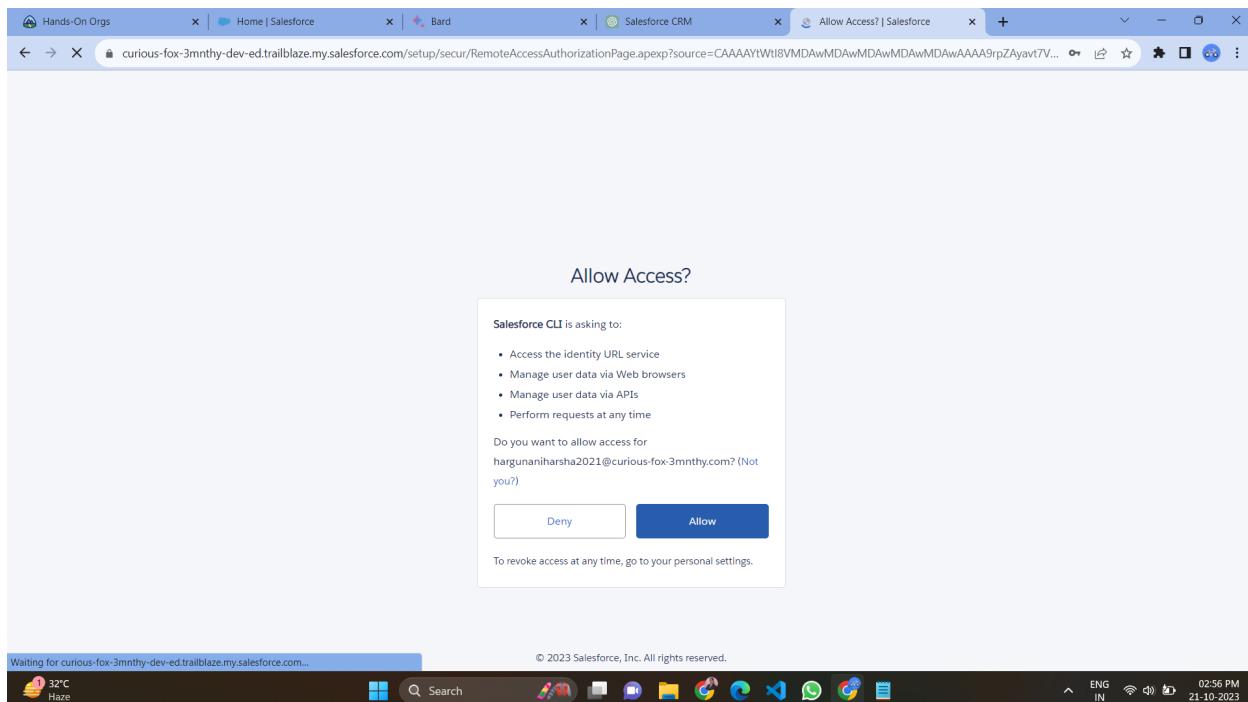


The screenshot shows the VS Code interface with the 'Project Default' dropdown open. The dropdown menu lists several options: Production, login.salesforce.com, Sandbox, test.salesforce.com, Custom, and Enter a custom login URL. The 'Custom' option is currently selected. The code editor shows a component definition with a target attribute set to 'Custom'.

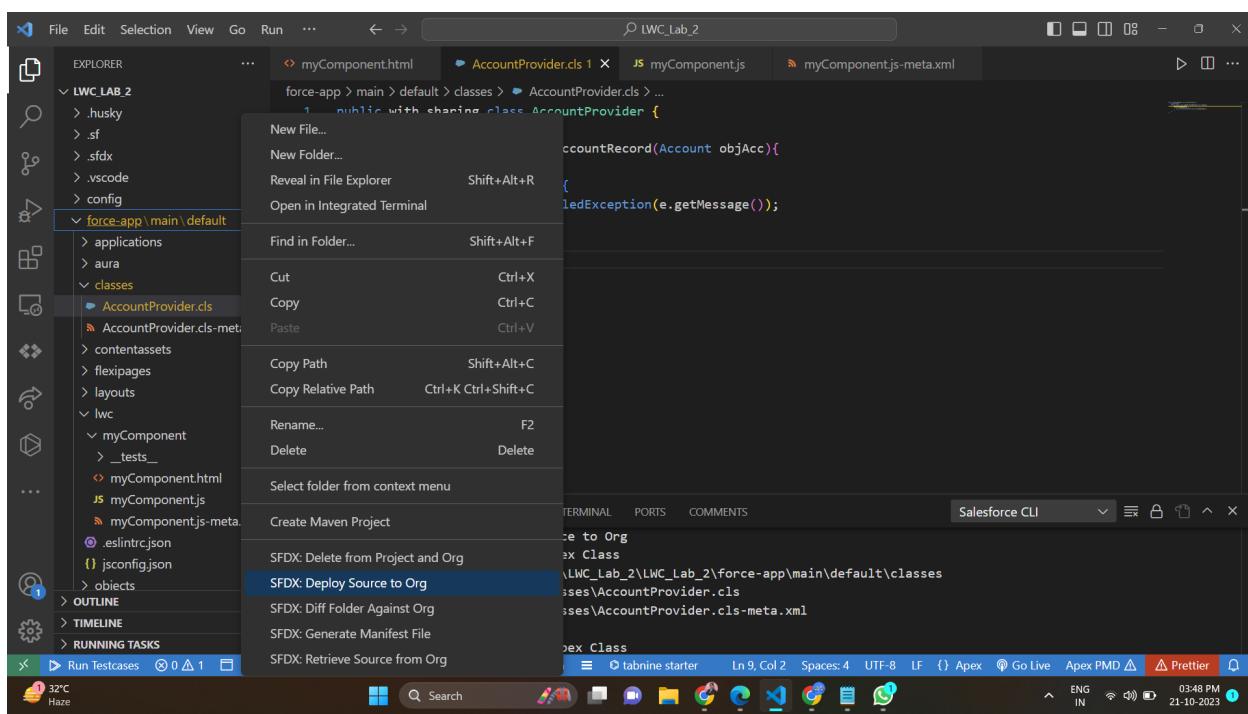
```
<!--targets-->
</LightningComponentBundle>
```

Step11: Enter Username and Password. Login and Allow Access.





Step12: Navigate back to vsCode and deploy the force-app\main\default folder.



```

1  public with sharing class AccountProvider {
2      @AuraEnabled
3      public static void createAccountRecord(Account objAcc){
4          try { insert objAcc;
5          } catch (Exception e) {
6              throw new AuraHandledException(e.getMessage());
7          }
8      }
9  }

```

The Explorer sidebar shows the project structure under LWC LAB_2, including force-app/main/default/classes/AccountProvider.cls and its metadata files.

The Problems panel shows the following changes:

STATE	FULL NAME	TYPE	PROJECT PATH
Changed	AccountProvider	ApexClass	force-app/main/default/classes/AccountProvider.cls
Changed	AccountProvider	ApexClass	force-app/main/default/classes/AccountProvider.cls-meta.xml
Changed	myComponent	LightningComponentBundle	force-app/main/default/lwc/myComponent/myComponent.html
Changed	myComponent	LightningComponentBundle	force-app/main/default/lwc/myComponent/myComponent.js
Changed	myComponent	LightningComponentBundle	force-app/main/default/lwc/myComponent/myComponent.js-meta.xml

Bottom status bar: 15:48:23.157 ended SFDX: Deploy Source to Org

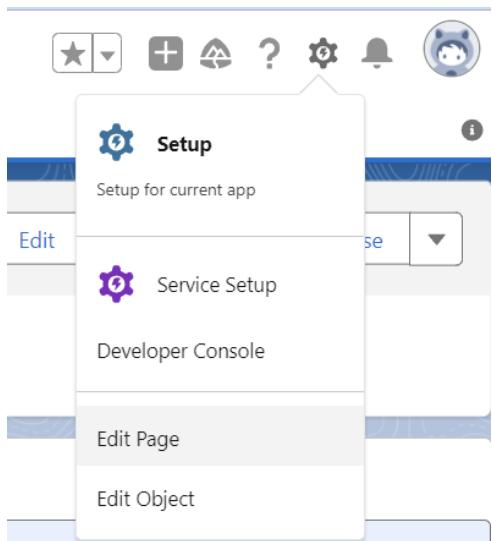
Step13: Login to the Salesforce Org and open the Account Tab in Apps.

Recently Viewed

Account Name	Account Site	Phone	Account Owner Alias
Demo Account 1		(336) 222-7000	HHarg
Burlington Textiles Corp of America			HHarg
Demo Account 2			HHarg

Bottom status bar: 04:27 PM 21-10-2023

Step14: Open any record and from setup click on Edit Page.



Step15: Builder gets open. Look for “myComponent” under Custom Components.

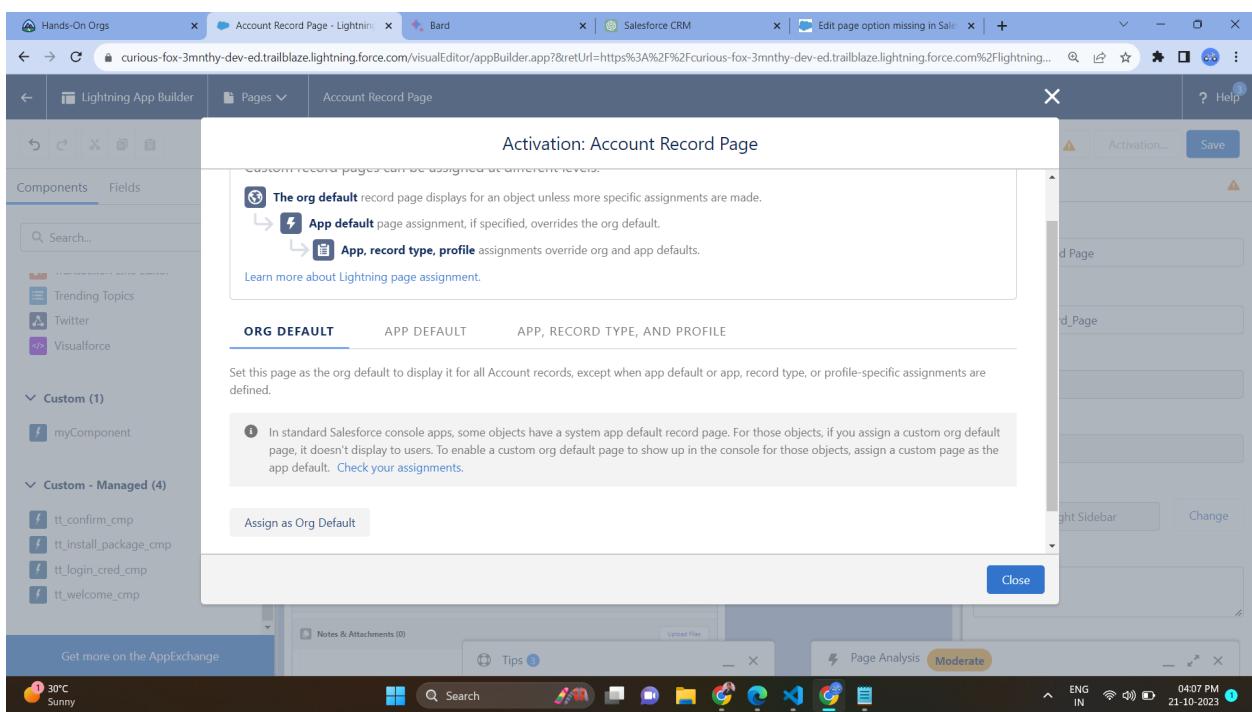
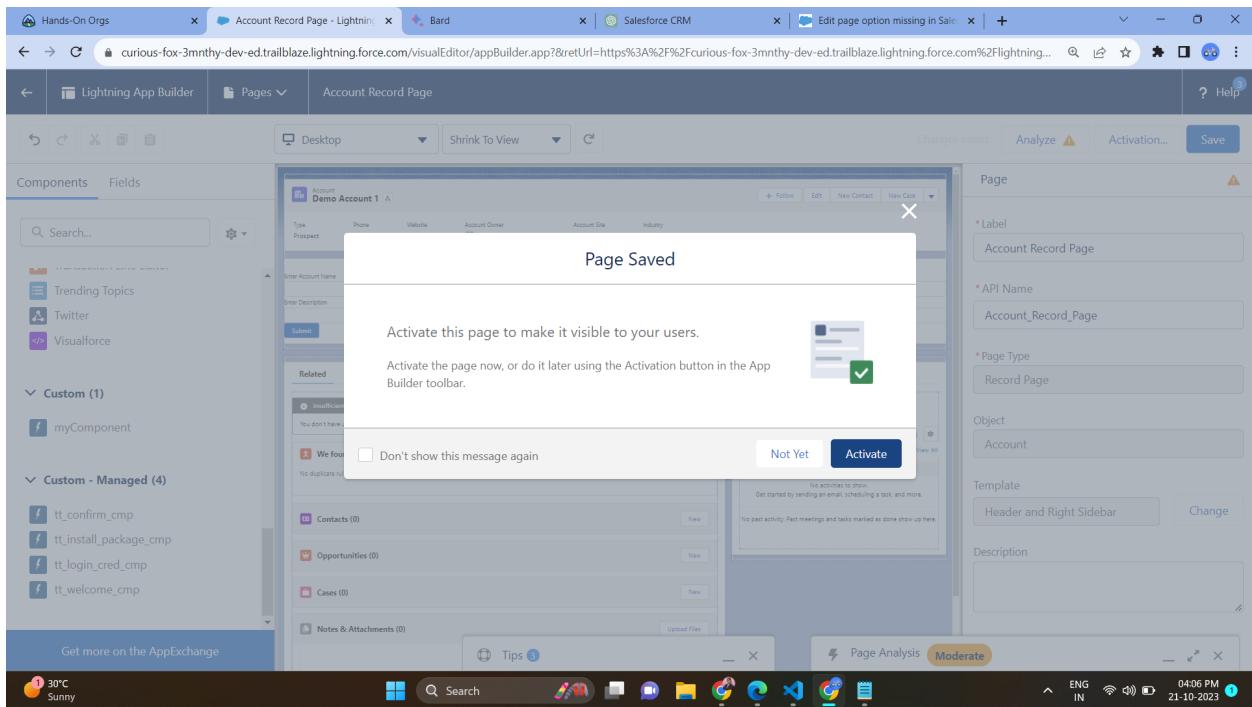
A screenshot of the Lightning App Builder interface. The left sidebar shows sections for 'Components' (with a search bar) and 'Fields'. Under 'Custom' components, there is one entry: 'myComponent'. Under 'Custom - Managed', there are four entries: 'tt_confirm_cmp', 'tt_install_package_cmp', 'tt_login_cred_cmp', and 'tt_welcome_cmp'. The main workspace displays an 'Account' record page for 'Demo Account 1'. On the right, the 'Page' configuration panel is open, showing fields for 'Label' (set to 'Account Record Page'), 'API Name' (set to 'Account_Record_Page'), 'Page Type' (set to 'Record Page'), 'Object' (set to 'Account'), 'Template' (set to 'Header and Right Sidebar'), and a 'Description' field. A 'Tips' panel is also visible at the bottom right.

Step16: Drag and drop the component in the page.

The screenshot shows the Salesforce App Builder interface. At the top, there are three buttons: 'Desktop' (selected), 'Shrink To View', and a refresh icon. Below these is a header bar with the title 'myComponent' and a small icon. The main content area displays an 'Account' record for 'Demo Account 1'. The account details include: Type (Prospect), Phone, Website, Account Owner (Harsha Hargunani), Account Site, and Industry. Below the details, there are tabs for 'Related', 'Details', and 'News'. The 'Related' tab is selected. A message box states 'Insufficient permissions' and 'You don't have user access to view this component.' Another message box below it says 'We found no potential duplicates of this Account.' and 'No duplicate rules are activated. Activate duplicate rules to identify potential duplicate records.'

The screenshot shows a web browser window with multiple tabs open. The active tab is titled 'Account Record Page'. The page content is identical to the one in the App Builder screenshot above, showing the 'Demo Account 1' record with the same fields and related components. The browser's address bar shows the URL: 'curious-fox-3mnthy-dev-ed.trailblaze.lightning.force.com/visualEditor/appBuilder.app?id=sfa_Account_rec_l&recordId=0015g00001Rc3PPAAZ&cloneable=true&retUrl=https%3A%2F%...'. The bottom of the screen shows a taskbar with various icons and system status information, including weather (30°C Sunny), search, and system notifications.

Step17: Click on Save and then Activate the new page.



Step18: Now the form is visible on a record's page.

The screenshot shows the Salesforce Lightning Experience interface. The top navigation bar includes tabs for 'Hands-On Orgs', 'Demo Account 1 | Account | Sales', 'Bard', 'Salesforce CRM', and 'Edit page option missing in Sales'. The main content area is an 'Account' page for 'Demo Account 1'. The page has fields for 'Type' (Prospect), 'Phone', 'Website', 'Account Owner' (Harsha Hargunani), 'Account Site', and 'Industry'. Below these fields is a large input area labeled 'Enter Account Name' containing 'Harsha Hargunani'. Another input area labeled 'Enter Description' contains the text 'Hi there! LWC Practical 8 performed!'. A 'Submit' button is located at the bottom left of this area. To the right of the input area is a sidebar titled 'Activity' which lists various activity types like Tasks, Events, and Chatter. The bottom of the screen shows a taskbar with icons for Weather (30°C Sunny), Search, and several application shortcuts, along with system status indicators for language (ENG IN), battery (04:07 PM), and date (21-10-2023).

This screenshot is nearly identical to the one above, showing the same 'Account' page for 'Demo Account 1'. The 'Enter Account Name' field now contains 'Harsha Hargunani' and the 'Enter Description' field contains 'Hi there! LWC Practical 8 performed!'. The 'Submit' button is visible. The 'Activity' sidebar and the system taskbar at the bottom are also present. The URL in the browser address bar is identical to the previous screenshot: 'curious-fox-3mnthy-dev-ed.trailblaze.lightning.force.com/lightning/r/Account/0015g00001Rc3PPAAZ/view'.