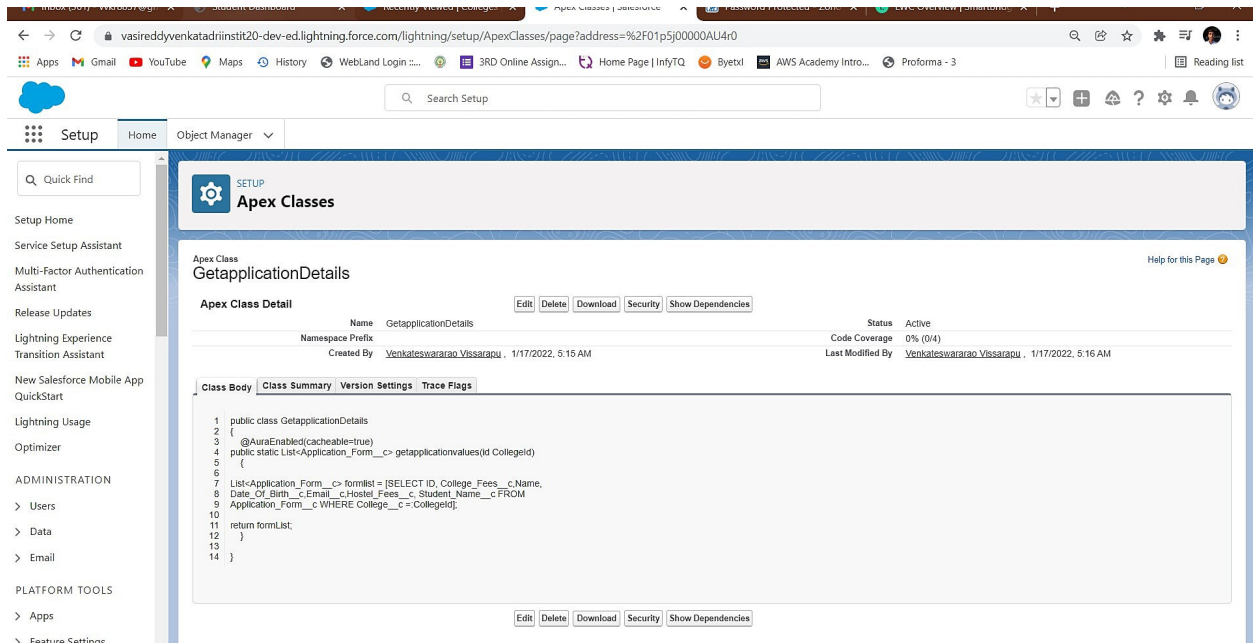


Day : 8

Topic : Lightning Web Components

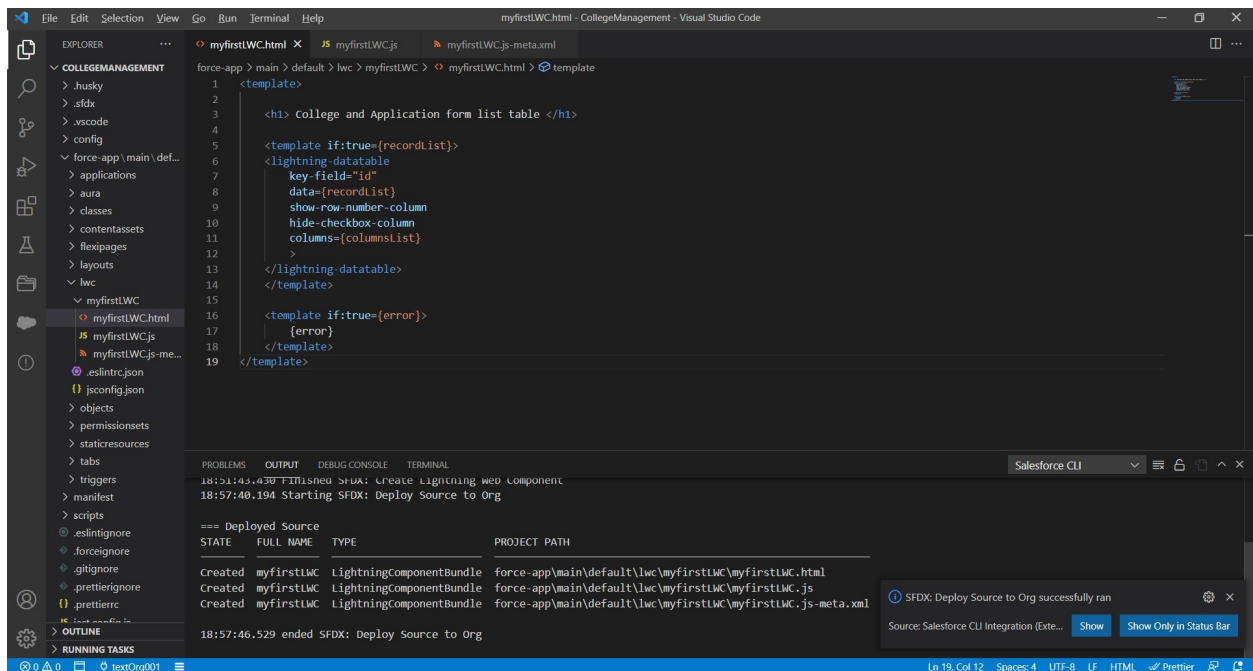
Activites : 1)Create College DataTable Component( APEX CLASS)



The screenshot shows the Salesforce Setup interface. The left sidebar contains navigation links like 'Setup Home', 'Service Setup Assistant', 'Multi-Factor Authentication Assistant', 'Release Updates', 'Lightning Experience Transition Assistant', 'New Salesforce Mobile App QuickStart', 'Lightning Usage', 'Optimizer', and 'ADMINISTRATION' (with sub-links for Users, Data, and Email). Below these are 'PLATFORM TOOLS' (with sub-links for Apps and Feature Settings). The main content area is titled 'Apex Classes' and shows details for the class 'GetApplicationDetails'. It includes buttons for 'Edit', 'Delete', 'Download', 'Security', and 'Show Dependencies'. The class is active, created by Venkateswararao Vissarapu on 1/17/2022 at 5:15 AM, with 0% code coverage. The 'Class Body' tab is selected, showing the following Apex code:

```
1 public class GetApplicationDetails
2 {
3     @AuraEnabled(cacheable=true)
4     public static List<Application_Form__c> getApplicationvalues(Id CollegeId)
5     {
6
7         List<Application_Form__c> formList = [SELECT ID, College_Fees__c, Name,
8         Date_Of_Birth__c, Email__c, Hostel_Fees__c, Student_Name__c FROM
9         Application_Form__c WHERE College__c =:CollegeId];
10
11         return formList;
12     }
13 }
14 }
```

2)Create College DataTable Component (HTML FILE)



The screenshot shows the Visual Studio Code editor with the file 'myfirstLWC.html' open. The file contains the following HTML template:

```
1 <template>
2
3     <h1> College and Application form list table </h1>
4
5     <template if:true={recordList}>
6         <lightning-datatable
7             key-field="id"
8             data={recordList}
9             show-row-number-column
10            hide-checkbox-column
11            columns={columnsList}
12        >
13        </lightning-datatable>
14    </template>
15
16    <template if:true={error}>
17        {error}
18    </template>
19 </template>
```

The Explorer sidebar on the left shows the project structure for 'COLLEGE MANAGEMENT', including files like 'myfirstLWC.html', 'myfirstLWC.js', and 'myfirstLWC.js-meta.xml'. The bottom of the screen shows the 'PROBLEMS' and 'OUTPUT' panels, with a message indicating that the source was successfully deployed to the org.

### 3) Create College DataTable Component(JAVA SCRIPT FILE)

The screenshot shows the Visual Studio Code interface with the Explorer, Search, and Run and Debug panels. The Explorer panel shows the project structure for 'myfirstLWC'. The Search panel shows the results of a search for 'myfirstLWC.js'. The Run and Debug panel shows the output of the SFDX command 'sfdx force:deploy:source:run', which successfully deployed the source to the Org.

```
force-app > main > default > lwc > myfirstLWC > JS myfirstLWC.js > myfirstLWC.js-meta.xml > columnsList
1 import { LightningElement, api, wire } from 'lwc';
2 import getApplicationValues
3 from '@salesforce/apex/GetApplicationDetails.getApplicationValues';
4 export default class CollegeDataTable extends LightningElement {
5   columnList = [
6     {label: 'Application Form', fieldName: 'Name', type: 'text' },
7     {label: 'College Fees', fieldName: 'College_Fees_c', type: 'currency' },
8     {label: 'Date Of Birth', fieldName: 'Date_of_Birth_c', type: 'date' },
9     {label: 'Email', fieldName: 'Email_c', type: 'email' },
10    {label: 'Hostel Fees', fieldName: 'Hostel_Fees_c', type: 'currency' },
11    {label: 'StudentName', fieldName: 'Student_Name_c', type: 'text' }
12  ];
13
14  @api recordId;
15  recordList;
16  error;
17
18  @wire(getApplicationValues, {collegeId: '$recordId'})
19  wiredCollegeData({data, error}){
20    if(data){
21      this.recordList = data;
22    }
23    else if(error){
24      this.error = error;
25    }
26  }
27}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

18:59:40.529 ERROR SFDX: Deploy Source to Org

18:59:25.92 Starting SFDX: Deploy Source to Org

=== Deployed Source

STATE	FULL NAME	TYPE	PROJECT PATH
Changed	myfirstLWC	LightningComponentBundle	force-app/main/default/lwc/myfirstLWC/myfirstLWC.html
Changed	myfirstLWC	LightningComponentBundle	force-app/main/default/lwc/myfirstLWC/myfirstLWC.js
Changed	myfirstLWC	LightningComponentBundle	force-app/main/default/lwc/myfirstLWC/myfirstLWC.js-meta.xml

18:59:30.984 ended SFDX: Deploy Source to Org

### 4)Create College DataTable Component(META FILE)

The screenshot shows the Visual Studio Code interface with the Explorer, Search, and Run and Debug panels. The Explorer panel shows the project structure for 'myfirstLWC'. The Search panel shows the results of a search for 'myfirstLWC.js-meta.xml'. The Run and Debug panel shows the output of the SFDX command 'sfdx force:deploy:source:run', which successfully deployed the source to the Org.

```
force-app > main > default > lwc > myfirstLWC > myfirstLWC.js-meta.xml > LightningComponentBundle
1 <?xml version="1.0"?>
2 <LightningComponentBundle
3   xmlns="http://soap.sforce.com/2006/04/metadata">
4   <apiVersion>51.0</apiVersion>
5   <isExposed>true</isExposed>
6   <targets>
7     <target>lightning_RecordPage</target>
8     <target>lightning_AppPage</target>
9     <target>lightning_HomePage</target>
10  </targets>
11 </LightningComponentBundle>
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

19:00:30.984 ERROR SFDX: Deploy Source to Org

19:00:54.645 Starting SFDX: Deploy Source to Org

=== Deployed Source

STATE	FULL NAME	TYPE	PROJECT PATH
Changed	myfirstLWC	LightningComponentBundle	force-app/main/default/lwc/myfirstLWC/myfirstLWC.html
Changed	myfirstLWC	LightningComponentBundle	force-app/main/default/lwc/myfirstLWC/myfirstLWC.js
Changed	myfirstLWC	LightningComponentBundle	force-app/main/default/lwc/myfirstLWC/myfirstLWC.js-meta.xml

19:00:59.787 ended SFDX: Deploy Source to Org

SFDX: Deploy Source to Org successfully ran

Source: Salesforce CLI Integration (Ext...) Show

The screenshot displays the Salesforce Lightning Components Setup interface. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The left sidebar menu lists various setup areas, with 'Lightning Components' selected. The main content area, titled 'Lightning Components', provides an overview of the framework, stating that it is a compact, configurable, and reusable element used for building apps and custom pages. It mentions two development models: Aura (the original) and Lightning Web Components (LWC, the newer model). Below the text is a table listing components. The table has columns for Action, Name, Label, Type, Namespace Prefix, and Api Version. A single component, 'myfirstLWC', is listed with a Type of 'LWC' and an Api Version of '51.0'.

Action	Name	Label	Type	Namespace Prefix	Api Version
Del	myfirstLWC	myfirstLWC	LWC		51.0