

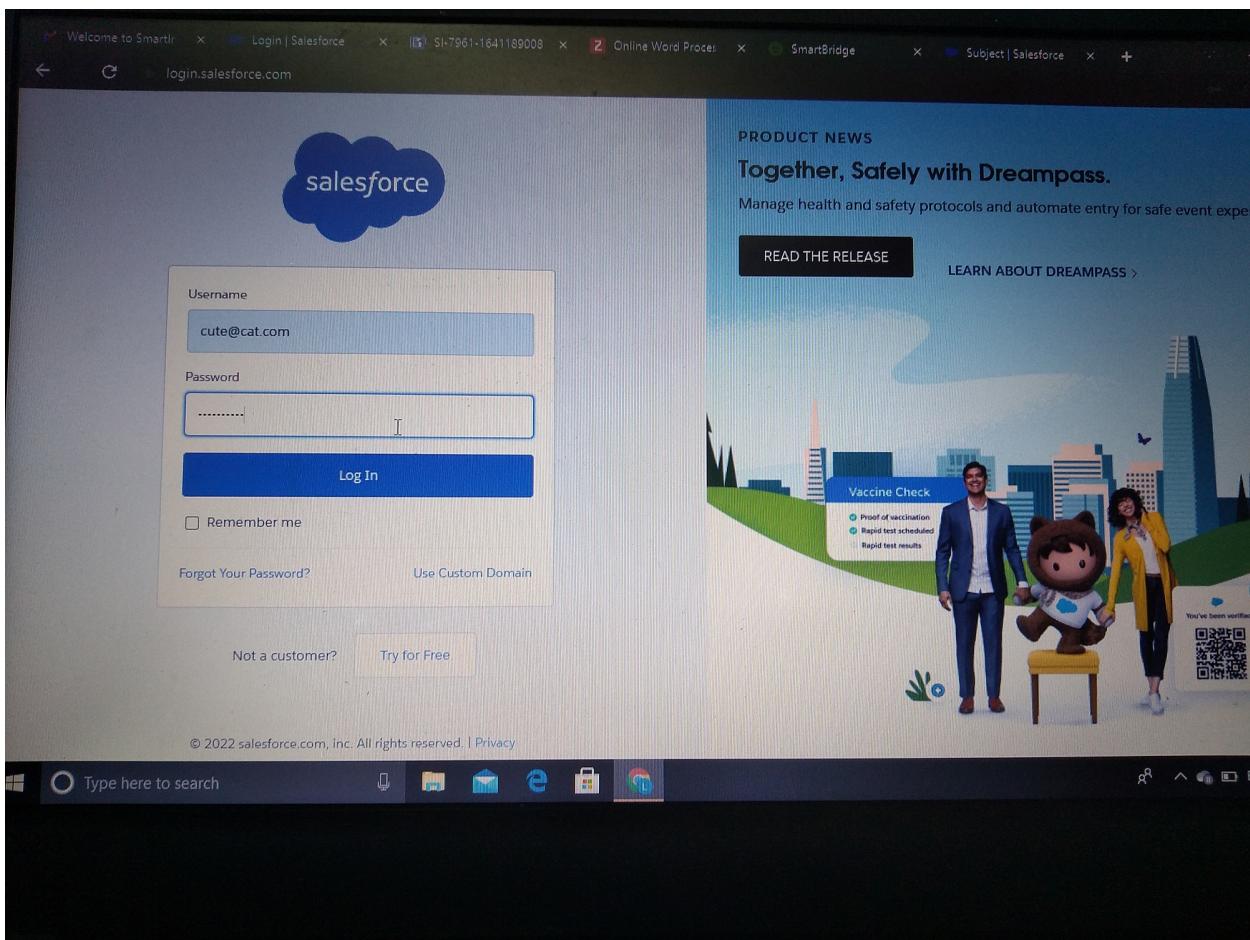
# COLLEGE MANAGEMENT APPLICATION

## Day 1: Create Your Salesforce Developer Org To Get Started

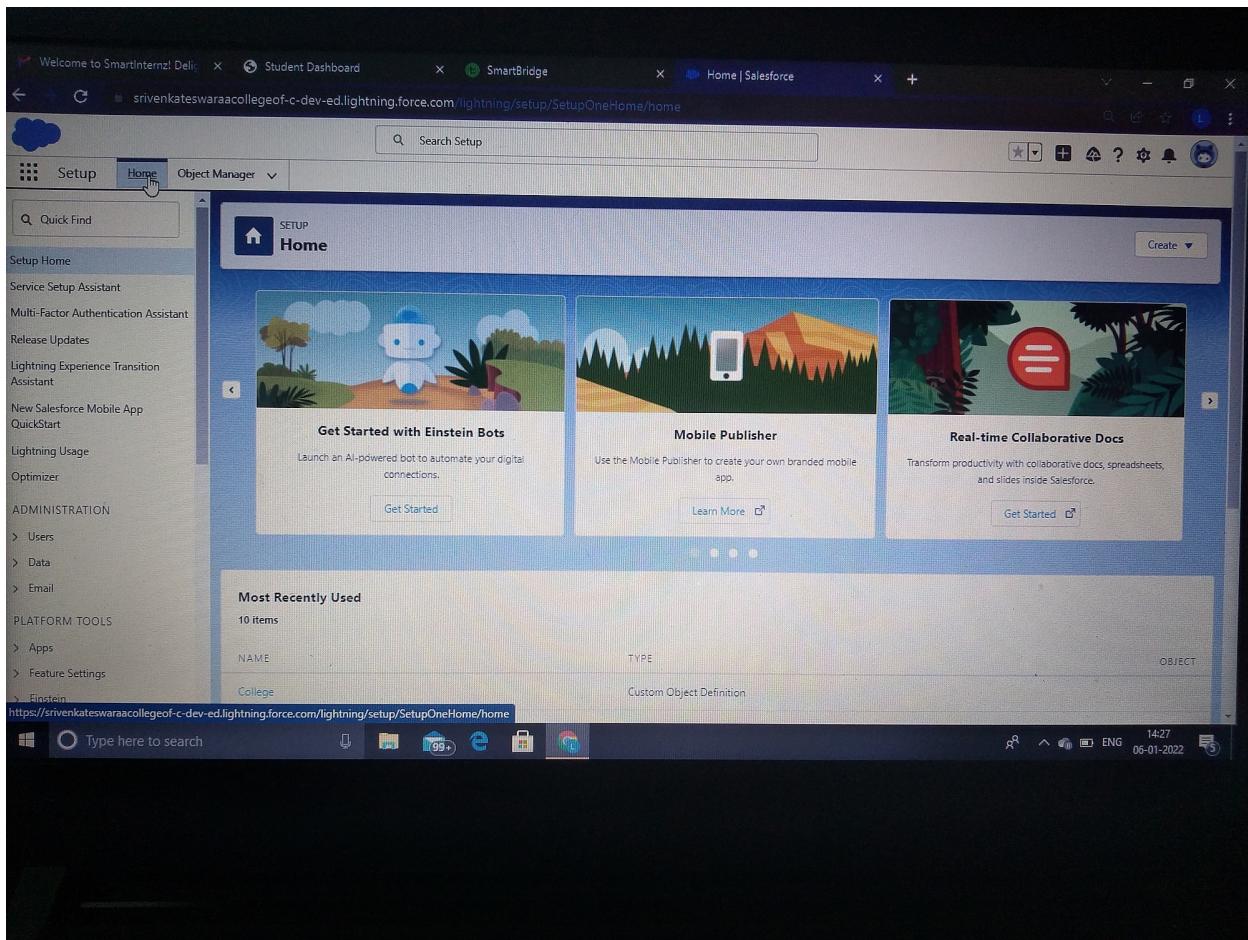
**Milestone Activities :** Created ,Activated , and Logged in.

**Description :** Successfully Created The Salesforce Developer Org Account.

### Salesforce Login Page



## Salesforce Home Page



## Day 2: Custom Object Creation

**Milestone Activities :** Create the Custom Objects and the Fields for Individual Objects Like College, Application Form , Students and Subjects.

**Description:** Successfully Created the Custom Objects with Respected Fields and Relationships.

### Create Fields On College Object

The screenshot shows the Salesforce Object Manager interface for the 'College' object. The left sidebar lists various setup options like Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, etc. The main content area is titled 'Fields & Relationships' and displays a table of fields. The table has columns for FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The fields listed are:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Capacity Of Students	Capacity__Of_Students__c	Picklist	College Name	
College	College__c	Picklist		
College Fees	College_Fees__c	Currency(7, 2)		
College Name	College_Name__c	Picklist		
College Name	Name	Text(80)		
Created By	CreatedBy	Lookup(User)		
Email	Email__c	Picklist	College Name	
Hostel Fees	Hostel_Fees__c	Currency(6, 2)		
Last Modified By	LastModifiedBy	Lookup(User)		

## Create Fields On Application Form Object

The screenshot shows the Salesforce Object Manager interface for the 'Application Form' object. The left sidebar contains navigation links such as Details, Fields & Relationships (which is selected), Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, Search Layouts for Salesforce Classic, and Restriction Rules. The main content area is titled 'Fields & Relationships' and displays a table of 15 items, sorted by Field Label. The columns in the table are FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The fields listed are:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Address	Address__c	Text(255)		
Application Form ID	Application_Form_ID__c	Auto Number		
Application Form Name	Name	Text(80)		
College	College__c	Master-Detail(College)		
College Fees	College_Fees__c	Formula (Currency)		
Created By	CreatedBy	Lookup(User)		
Date of Birth	Date_of_Birth__c	Date		
Email	Email__c	Email (Unique)		
Guardian Name	Guardian_Name__c	Text(30)		

## Create Fields On Student Object

The screenshot shows the Salesforce Setup interface with the following details:

- Header:** Welcome to SmartInternz! Dev, Student Dashboard, SmartBridge, srivenkateswaraacollegeof-c-dev-ed.lightning.force.com lightning:/setup/ObjectManager/015|000000DNNnN/FieldsAndRelationships/view.
- Search Bar:** Search Setup
- Navigation:** Setup, Home, Object Manager
- Breadcrumbs:** SETUP > OBJECT MANAGER > Student
- Left Sidebar:** Details, Fields & Relationships (selected), Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, Search Layouts for Salesforce Classic, Restriction Rules.
- Table:** Fields & Relationships (11 Items, Sorted by Field Label)

Field Name	Label	Type
Application Form	Application_Form__c	Lookup(Application Form)
College Name	College_Name__c	Formula (Text)
Created By	CreatedById	Lookup(User)
Date Of Birth	DateOfBirth__c	Date
Guardian Name	Guardian_Name__c	Text(30)
Last Modified By	LastModifiedById	Lookup(User)
Owner	OwnerId	Lookup(User,Group)
Phone	Phone__c	Phone
Student Name	Student_Name__c	Text(30)
Student Name	Name	Text(30)
- System Status:** Type here to search, 99+, ENG, 06-01-2022, 14:55

## Create Fields On Subject Object

The screenshot shows the Salesforce Object Manager interface for the 'Subject' object. The left sidebar lists various setup options like Details, Page Layouts, Lightning Record Pages, etc. The main area displays the 'Fields & Relationships' section with 8 items. The table shows the following fields:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Paper 1	Paper_1__c	Picklist		
Paper2	Paper2__c	Picklist		
Student	Student__c	Lookup(Student)		✓
Subject ID	Subject_ID__c	Auto Number		
Subjects. Name	Name	Text(80)		✓

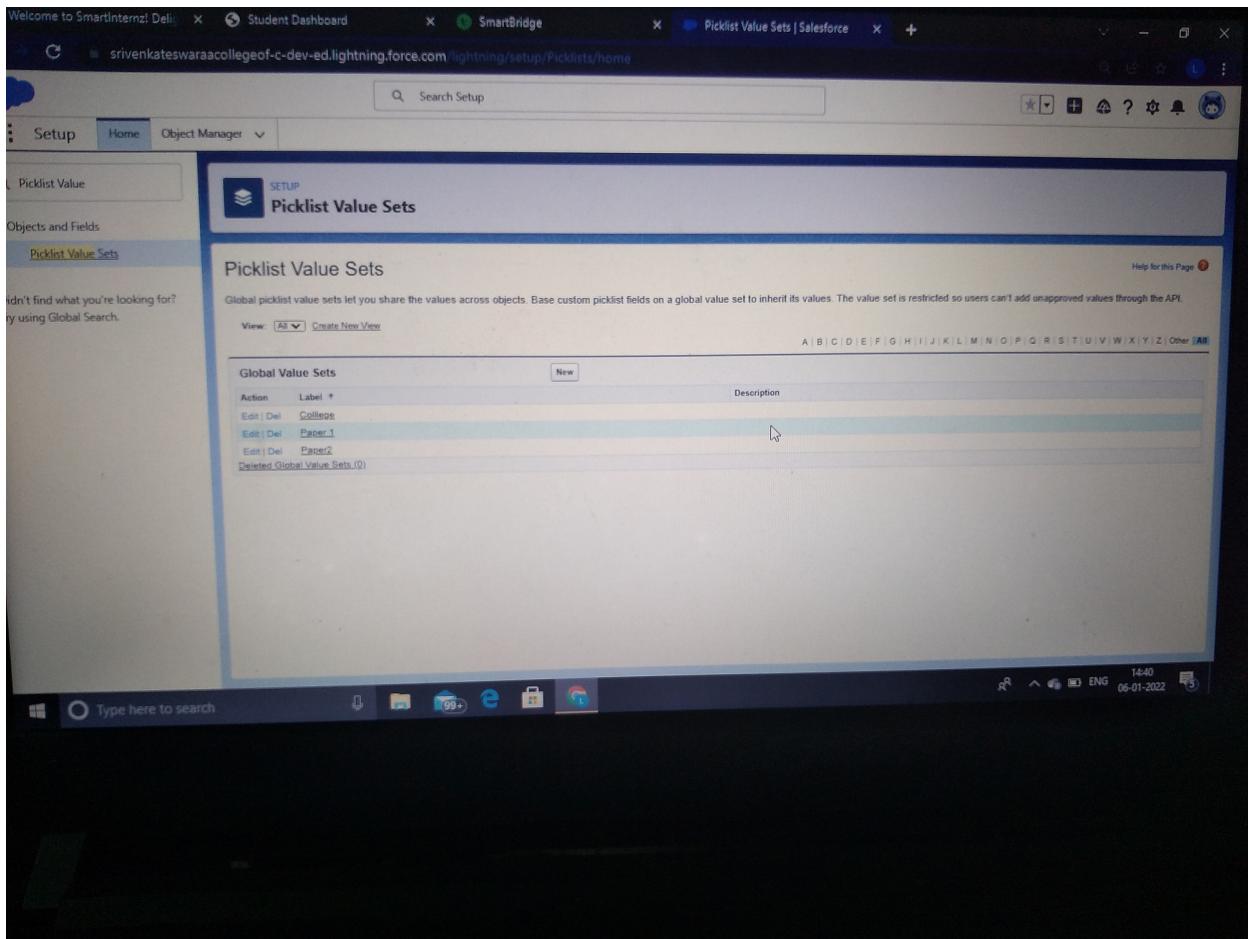
## Day 3: Adding Business Logic To Application

**Milestone Activities :** Create Global Picklist Values,Field Dependencies and Validation Rule.

**Description :** Successfully Completed Global Picklist Values,Field Dependencies and Validation Rule.

Below are the Screenshot for the Creation of Global Picklist Values,Field Dependencies and Validation Rule.

### *Creating Global Picklist Value Sets*



## Creating Field Dependencies On College Object

The screenshot shows the Salesforce Setup interface for the 'College' object. The left sidebar is titled 'Fields & Relationships' and includes options like Details, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, Search Layouts for Salesforce Classic, and Restriction Rules. The main content area is titled 'College Field Dependencies' and contains a table of field dependencies:

Action	Controlling Field	Dependent Field	Modified By
Edit   Del	College Name	Email	Lisa Christinal J_ 1/4/2022, 11:59 PM
Edit   Del	College Name	Capacity Of Students	Lisa Christinal J_ 1/5/2022, 12:02 AM

## Creating Validation Rule On College Object

The screenshot shows the Salesforce Setup interface for the College object. The left sidebar lists various configuration options like Details, Fields & Relationships, Page Layouts, etc. The main content area displays the 'College Validation Rule' details. The validation rule is named 'College\_Name\_Similar' and is active. The error condition formula is 'TEXT(College\_\_Name\_\_c) <> Name'. The error message is 'College name and record info should have the same name'. The rule was created by Lisa.Christina.J. on 1/5/2022, 12:03 AM and modified by the same user on 1/5/2022, 12:03 AM.

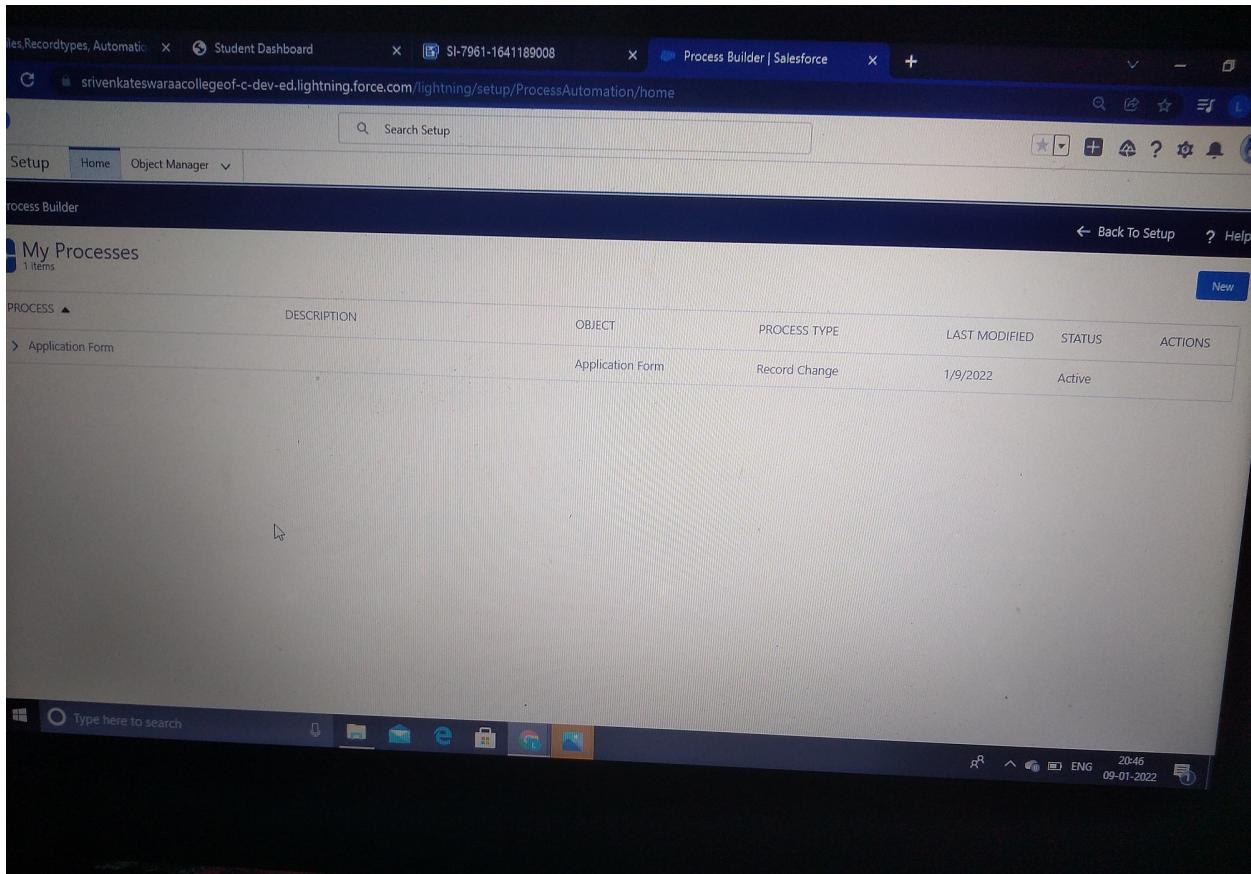
Rule Name	College_Name_Similar	Active
Error Condition Formula	TEXT(College__Name__c) <> Name	
Error Message	College name and record info should have the same name	Error Location
Description		Top of Page
Created By	Lisa.Christina.J. 1/5/2022, 12:03 AM	Modified By
		Lisa.Christina.J. 1/5/2022, 12:03 AM

## Day 4: Process Automation

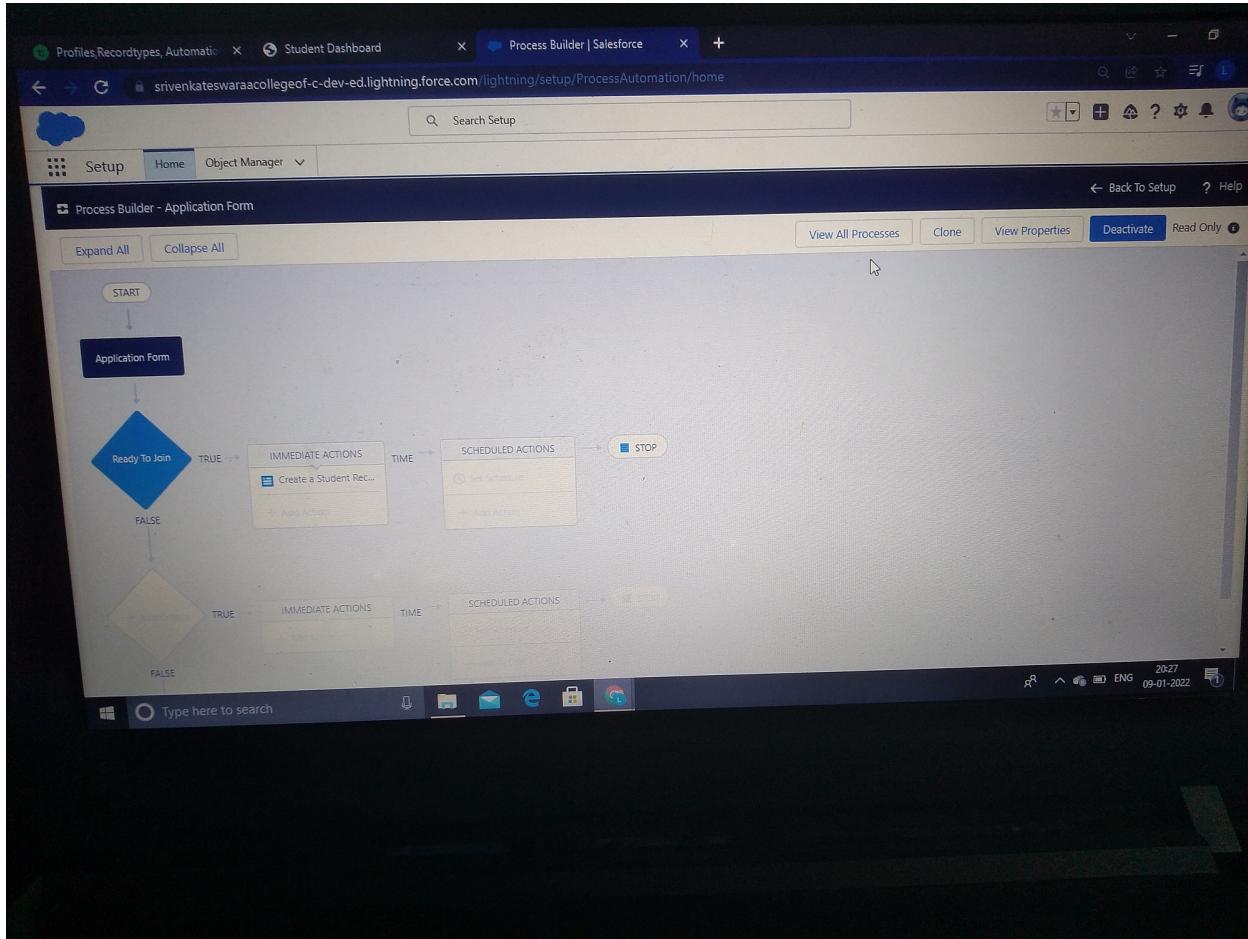
**Milestone Activities :** Create a Process Builder For Application Form .

**Description :** Successfully Created the Application Form Using Process Builder.

### *Creating Process Builder For Application Form*



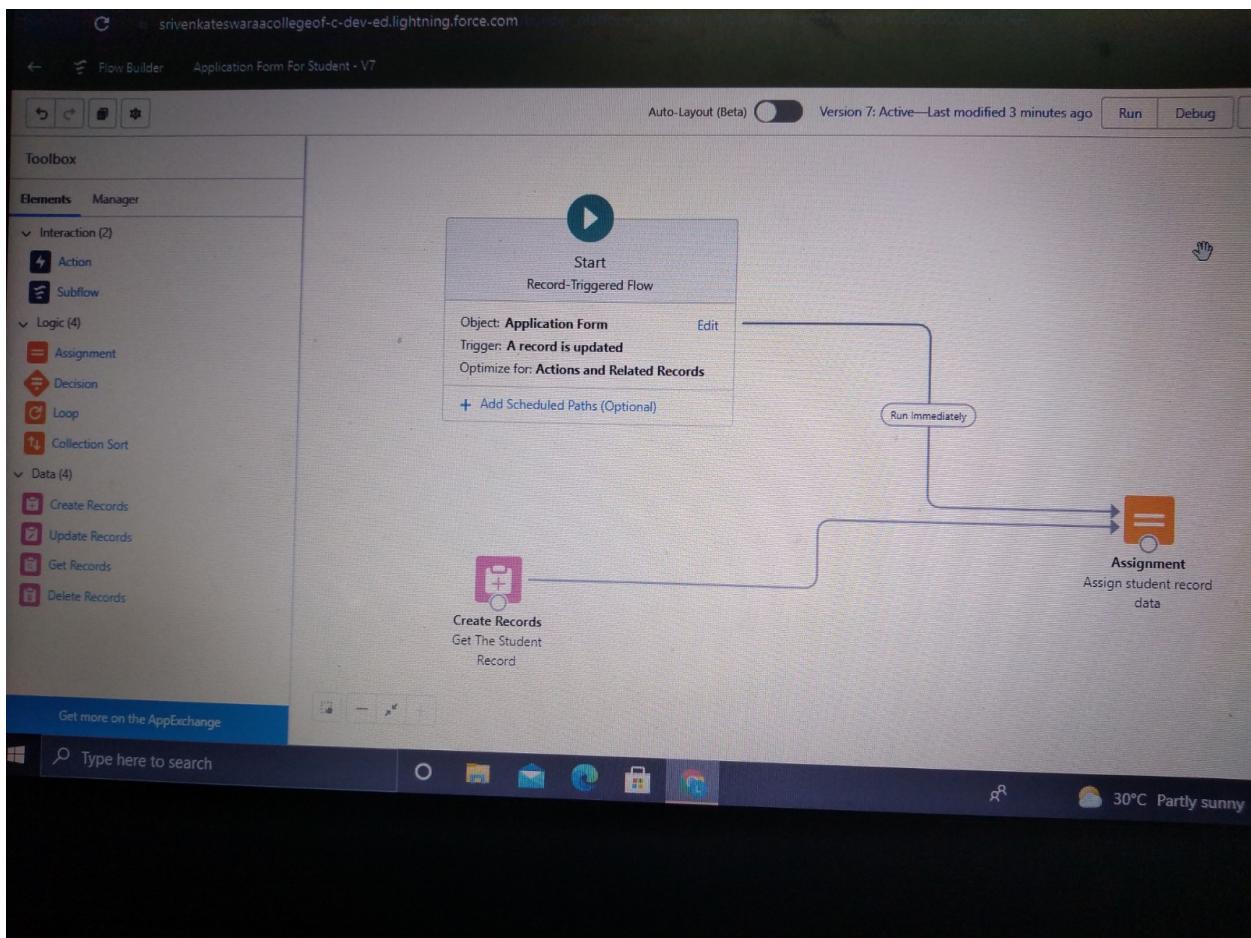
The Below Image Show that the Adding criteria as Ready To Join and Immediate Action as Create a Student Record for the Application Form Using Process Builder.



## Day 5: Create The Student Record Using Flow

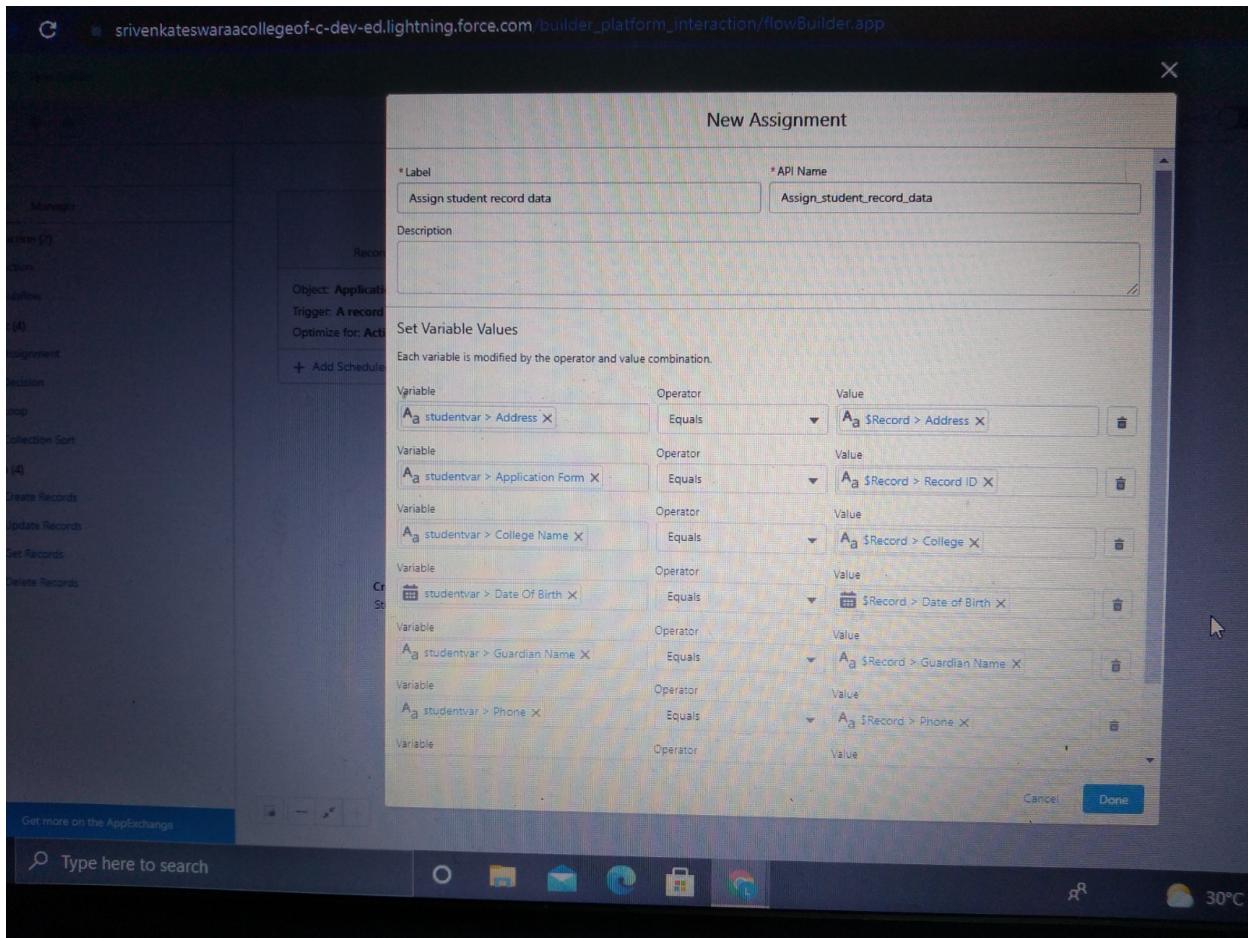
### Milestone Activities :

1. First deactivate the process builder which we created earlier
2. Now search for flows and select new flow ->record triggered flow.
3. For object select application form, in configure trigger select when the record is updated for entry criteria select as ready to join equals to true.
4. Now create a variable named student in the resource section.



The Above Image Represent the Record Triggered flow with Create Record and Assignment Elements.

5. Now add the assignment as follows.



**Description :** Successfully Created the Student Record Using Flow.

## Day 6: Creating a Batch Apex for Application Form

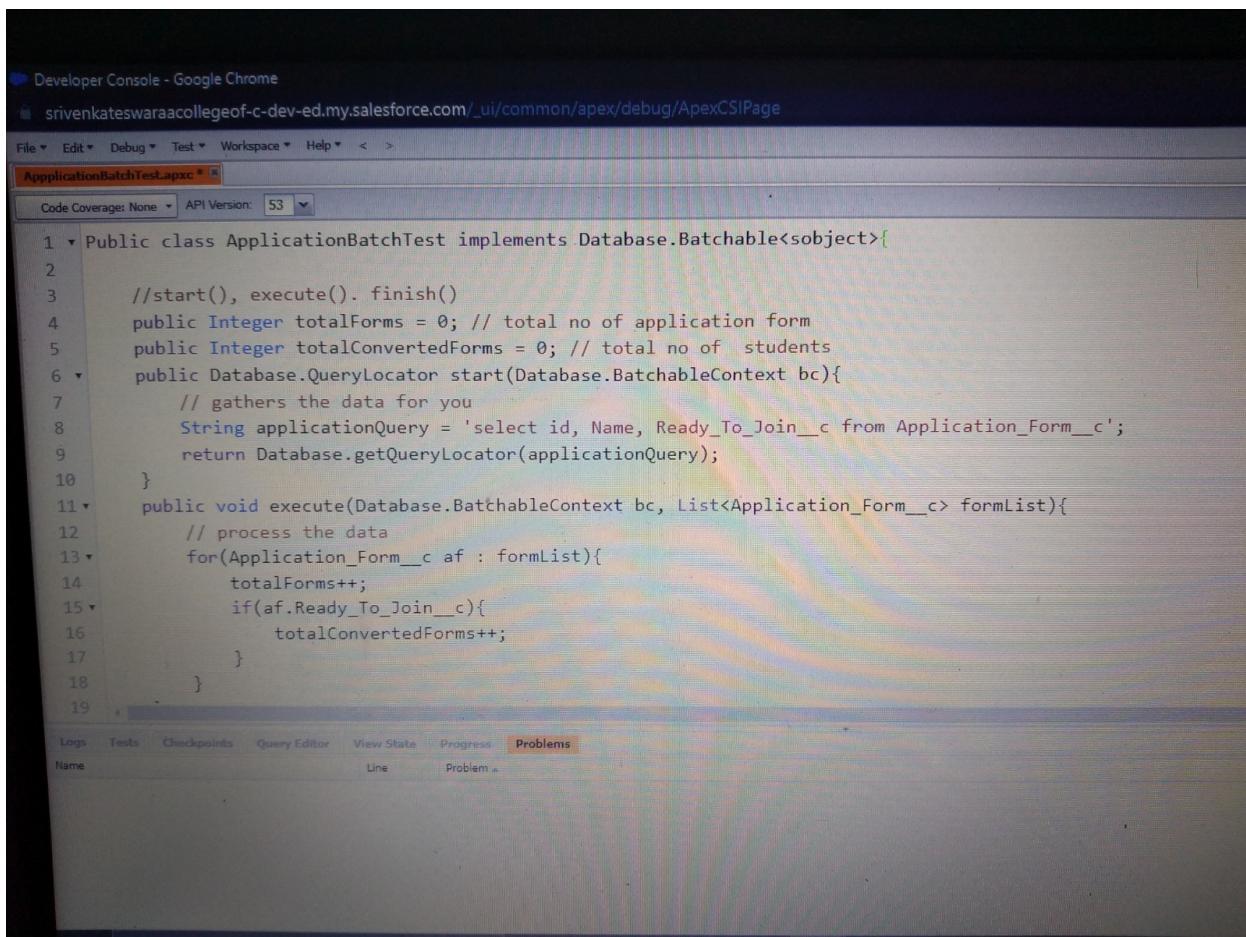
### Milestone Activities :

- Create a BatchApex for Application Form.
- Create a Schedular Class.

### Description :

Batches Apex is used to run large Jobs that would exceed normal processing limits. And the BathApex for Application Form and Schedular Class was Successfully Created .

### *Create A BatchApex For Application Form*



The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is srivenkateswaraacollegeof-c-dev-ed.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage. The tab title is ApplicationBatchTest.apxc. The code editor contains the following Apex class:

```
1 Public class ApplicationBatchTest implements Database.Batchable<sobject>{
2
3     //start(), execute(). finish()
4     public Integer totalForms = 0; // total no of application form
5     public Integer totalConvertedForms = 0; // total no of students
6     public Database.QueryLocator start(Database.BatchableContext bc){
7         // gathers the data for you
8         String applicationQuery = 'select id, Name, Ready_To_Join__c from Application_Form__c';
9         return Database.getQueryLocator(applicationQuery);
10    }
11    public void execute(Database.BatchableContext bc, List<Application_Form__c> formList){
12        // process the data
13        for(Application_Form__c af : formList){
14            totalForms++;
15            if(af.Ready_To_Join__c){
16                totalConvertedForms++;
17            }
18        }
19    }
}
```

The code editor has syntax highlighting for Apex. Below the code, there are tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Problems tab is selected, showing a single entry: "Line 19 Problem".

## Create A Schedular Class

The screenshot shows the Salesforce Developer Console interface. The title bar indicates it's running in Google Chrome on a Salesforce developer instance. The main area displays the code for a scheduled Apex class:

```
1 public class applicationschedule implements Schedulable{
2
3
4
5     public void execute(SchedulableContext sc){
6
7         ApplicationBatchTest abt = new ApplicationBatchTest();
8
9         Database.executeBatch(abt, 400); // 200 to 2000
10
11    }
12
13 }
14
15 }
```

Below the code editor, there are tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Problems tab is currently selected, showing a single entry:

Name	Type
Line	Problem

## Day 7: Creating Lightning Web Components

### Milestone Activities :

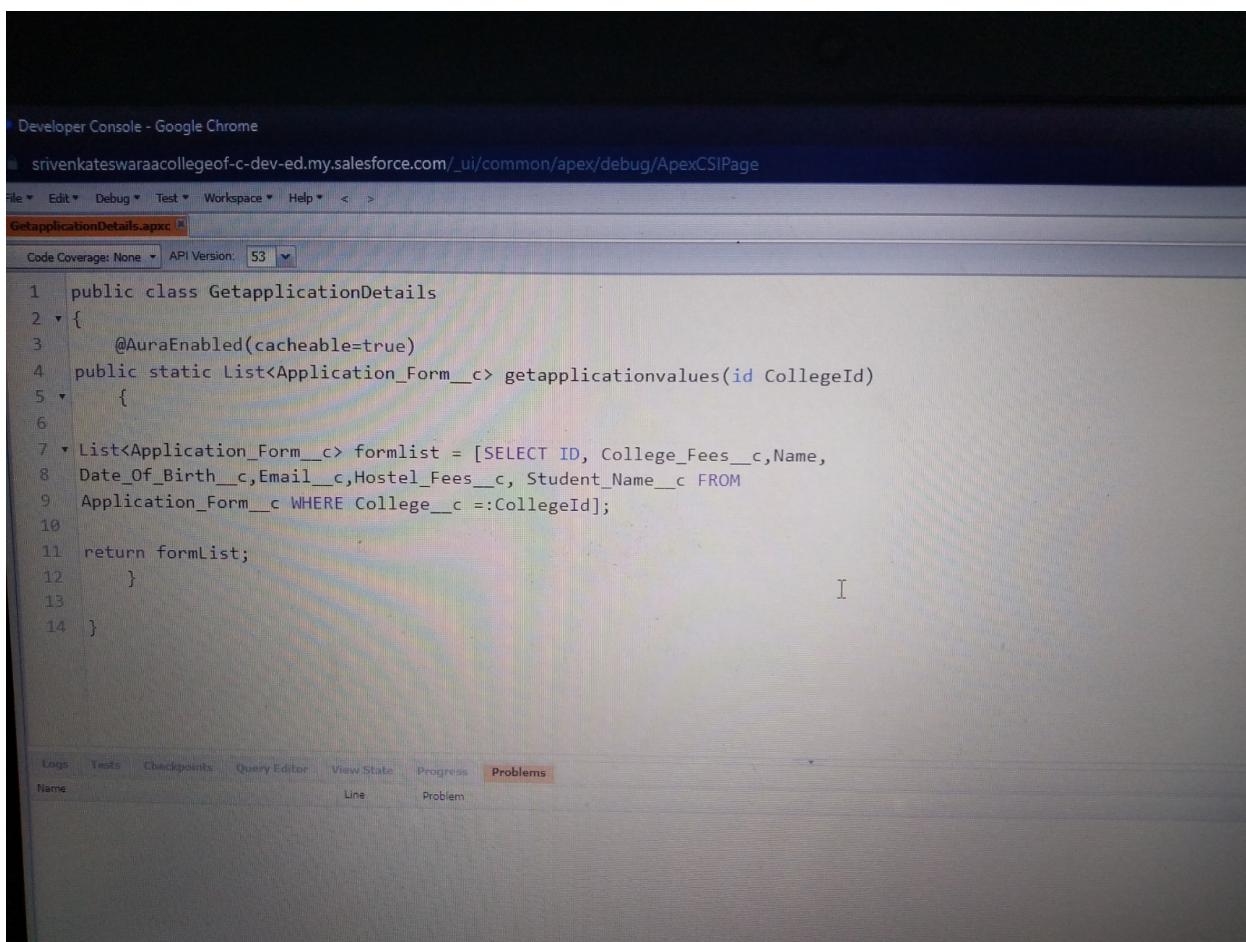
Create College DataTable Component Using

- APEX CLASS
- HTML FILE
- JAVA SCRIPT FILE
- META FILE

### Description :

Lightning Web Components (LWC) is a stack of modern lightweight frameworks built on the latest web standards. It is a DOM (Document Object Model), element created through reusable code and is used to generate a dynamic interface without using JavaScript or building a Library.

### ***Create College DataTable Component Using APEX CLASS***



The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is [srivenkateswaracollegeof-c-dev-ed.my.salesforce.com/\\_ui/common/apex/debug/ApexCSIPage](https://srivenkateswaracollegeof-c-dev-ed.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage). The page displays the code for the `GetapplicationDetails.apxc` class. The code is as follows:

```
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 }
```

The developer console also includes tabs for Log, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Problems tab is currently selected.

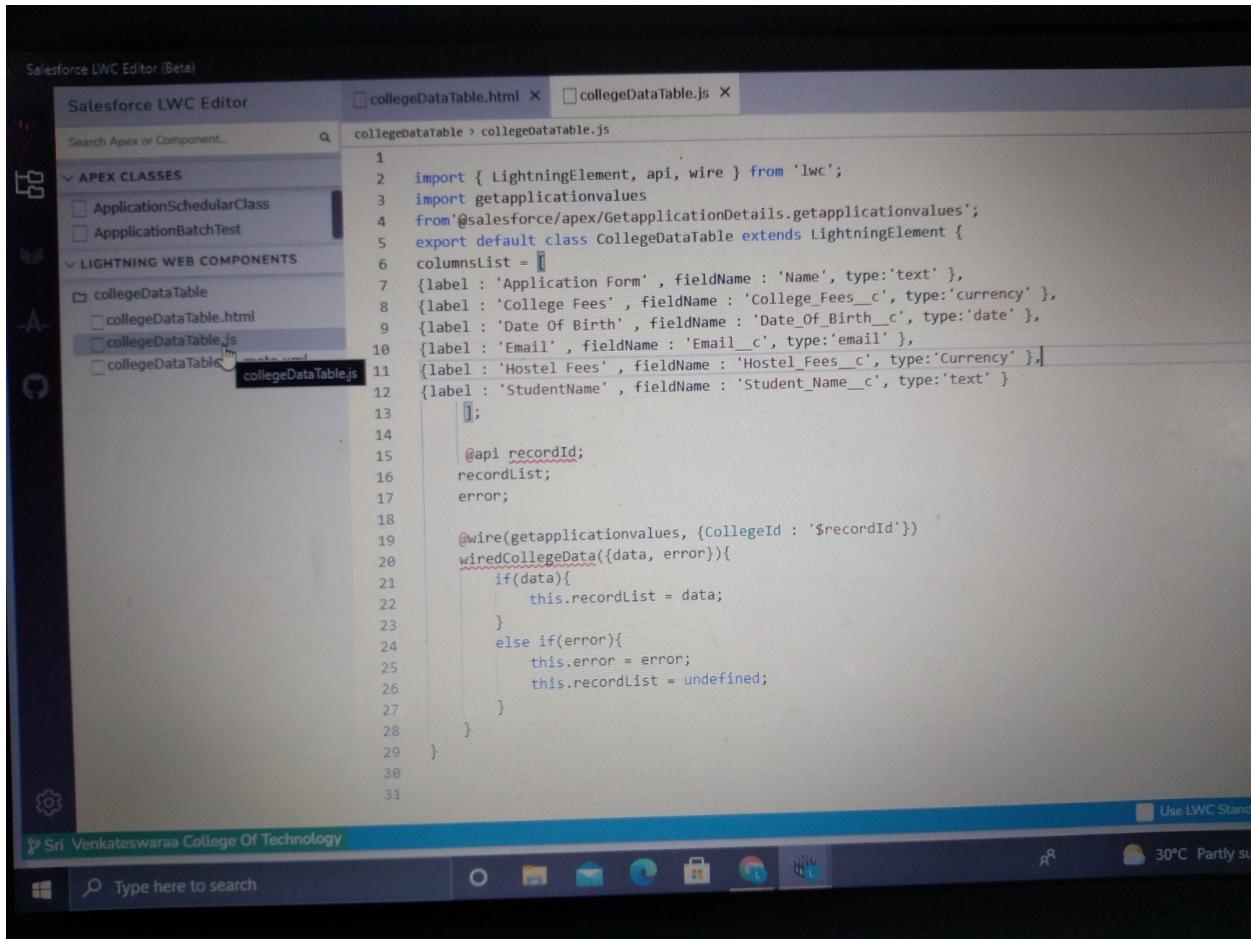
## Create College DataTable Component Using HTML FILE

The screenshot shows the Salesforce LWC Editor (Beta) interface. On the left, there's a sidebar with options like 'APEX CLASSES' and 'LIGHTNING WEB COMPONENTS'. Under 'LIGHTNING WEB COMPONENTS', a component named 'collegeDataTable' is selected. The main area displays the code for 'collegeDataTable.html'.

```
<template>
    .
    .
    <h1> College and Application form list table </h1>
    <template if:true={recordList}>
        <lightning-datatable
            key-field="id"
            data={recordList}
            show-row-number-column
            hide-checkbox-column
            columns={columnsList}
        >
    </lightning-datatable>
</template>
<template if:true={error}>
    {error}
</template>
```

Below the code, the 'Result Panel' shows the message: "CollegeDataTable component has been successfully created". The status bar at the bottom right indicates the date as 1/19/2022.

## Create College DataTable Component Using JAVA SCRIPT FILE

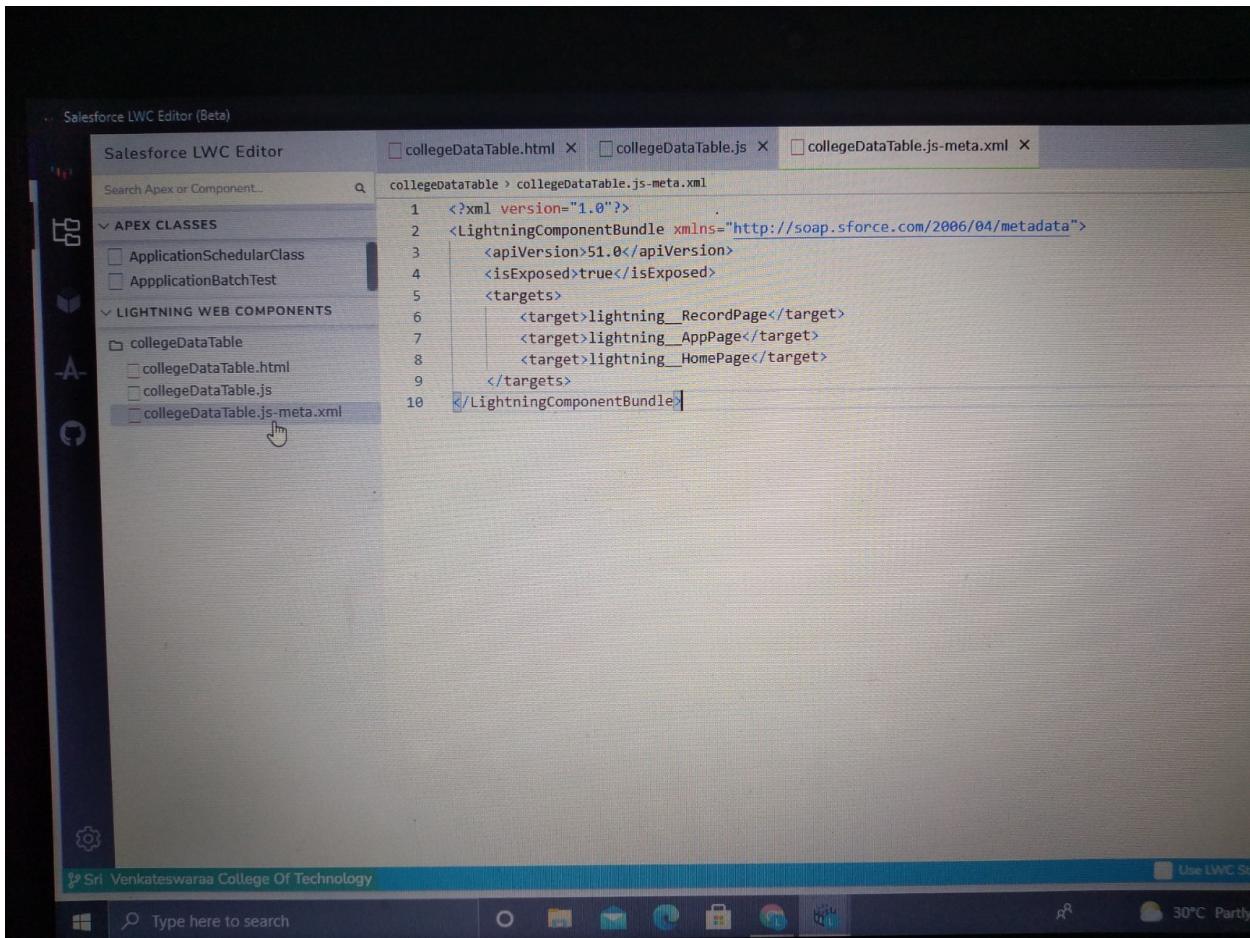


The screenshot shows the Salesforce LWC Editor (Beta) interface. The left sidebar lists 'APEX CLASSES' and 'LIGHTNING WEB COMPONENTS'. Under 'LIGHTNING WEB COMPONENTS', there is a folder named 'collegeDataTable' containing 'collegeDataTable.html' and 'collegeDataTable.js'. The 'collegeDataTable.js' file is open in the main editor area, displaying the following code:

```
1 import { LightningElement, api, wire } from 'lwc';
2 import getapplicationvalues
3 from '@salesforce/apex/GetapplicationDetails.getapplicationvalues';
4
5 export default class CollegeDataTable extends LightningElement {
6   columnsList = [
7     {label : 'Application Form' , fieldName : 'Name', type:'text' },
8     {label : 'College Fees' , fieldName : 'College_Fees__c', type:'currency' },
9     {label : 'Date Of Birth' , fieldName : 'Date_Of_Birth__c', type:'date' },
10    {Label : 'Email' , fieldName : 'Email__c', type:'email' },
11    {label : 'Hostel Fees' , fieldName : 'Hostel_Fees__c', type:'Currency' },
12    {label : 'StudentName' , fieldName : 'Student_Name__c', type:'text' }
13  ];
14
15  @api recordId;
16  recordlist;
17  error;
18
19  @wire(getapplicationvalues, {CollegeId : '$recordId'})
20  wiredCollegeData({data, error}){
21    if(data){
22      this.recordList = data;
23    }
24    else if(error){
25      this.error = error;
26      this.recordList = undefined;
27    }
28  }
29
30
31 }
```

The code defines a Lightning Web Component (LWC) named 'CollegeDataTable'. It imports 'LightningElement', 'api', and 'wire' from 'lwc'. It also imports 'getapplicationvalues' from the apex class 'GetapplicationDetails'. The component has a property 'columnsList' which contains an array of column definitions. It also has properties for 'recordId', 'recordlist', and 'error'. The component uses the 'wire' decorator to call the apex method 'getapplicationvalues' with the parameter 'CollegeId' set to the value of 'recordId'. The returned data is assigned to 'recordList', and any errors are assigned to 'error'.

## Create College DataTable Component Using META FILE



The screenshot shows the Salesforce LWC Editor (Beta) interface. The left sidebar lists 'APEX CLASSES' and 'LIGHTNING WEB COMPONENTS'. Under 'LIGHTNING WEB COMPONENTS', there is a folder 'collegeDataTable' containing 'collegeDataTable.html', 'collegeDataTable.js', and 'collegeDataTable.js-meta.xml'. The 'collegeDataTable.js-meta.xml' file is currently open in the main editor area. The code in the editor is:

```
<?xml version="1.0"?>
<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
<apiVersion>51.0</apiVersion>
<isExposed>true</isExposed>
<targets>
<target>lightning__RecordPage</target>
<target>lightning__AppPage</target>
<target>lightning__HomePage</target>
</targets>
</LightningComponentBundle>
```