

COLLEGE MANAGEMENT APPLICATION

Day 1 :

Topic : Creating Salesforce Developer Org To Get Started

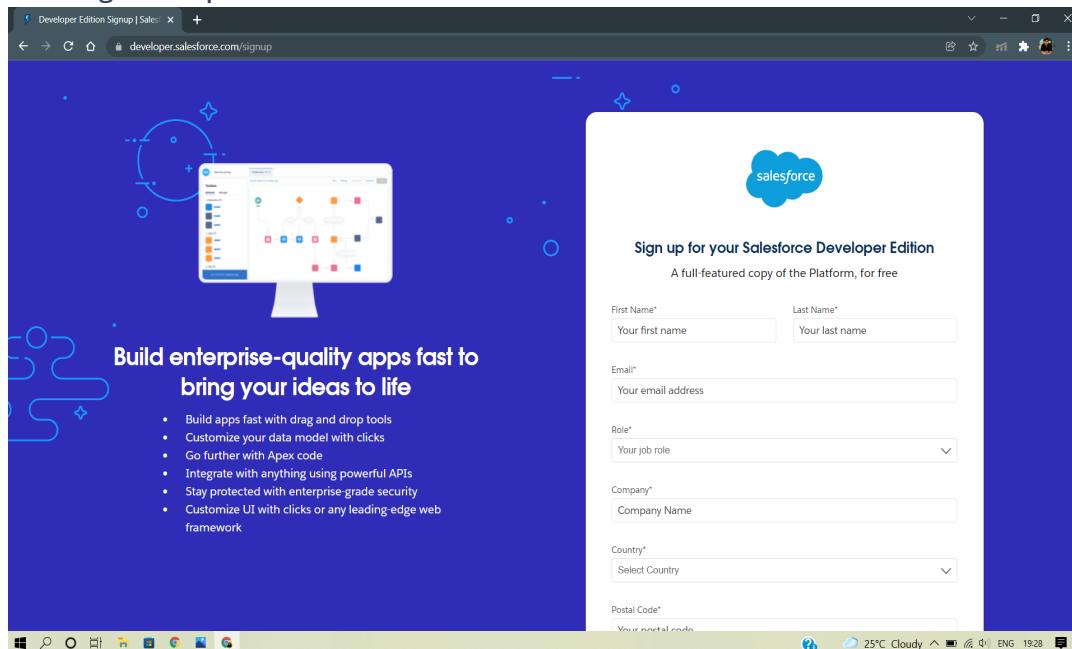
Milestone/Activity :Creating ,activating and loggin into salesforce developer account

Detailed Description :Creating a developer org in salesforce.

1. Go to developers.salesforce.com/
2. Click on sign up.
3. On the sign up form, enter the following details :
 1. First name & Last name
 2. Email
 3. Role : Developer
 4. Company : College Name
 5. County : India
 6. Postal Code : pin code
 7. Username : should be a combination of your name and company
This need not be an actual email id, you can give anything in the format :username@organization.com

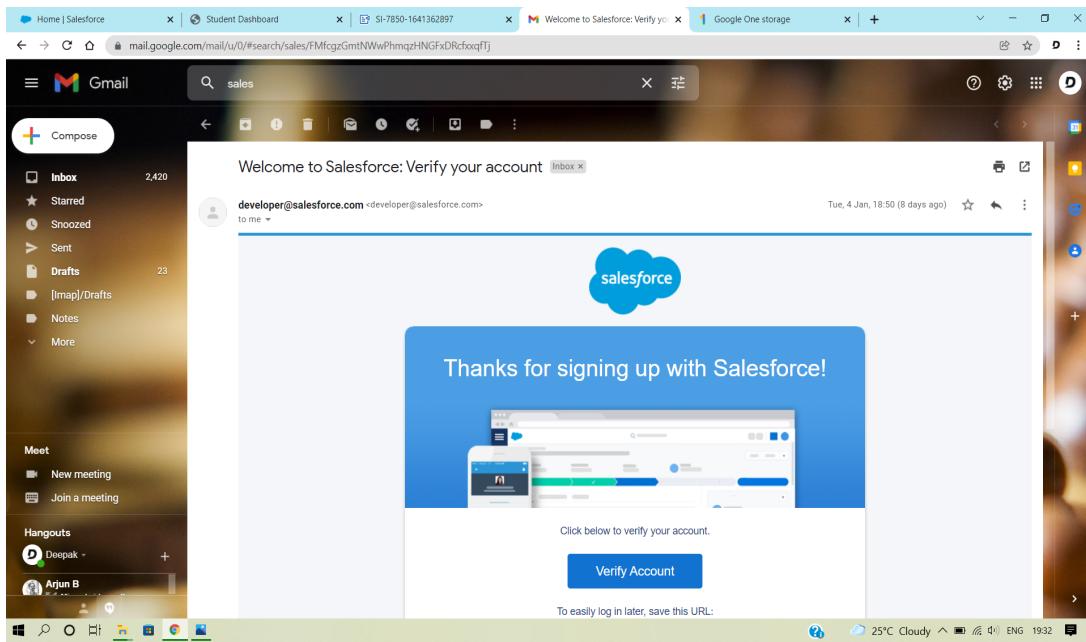
Screenshots :

Creating Developer Account



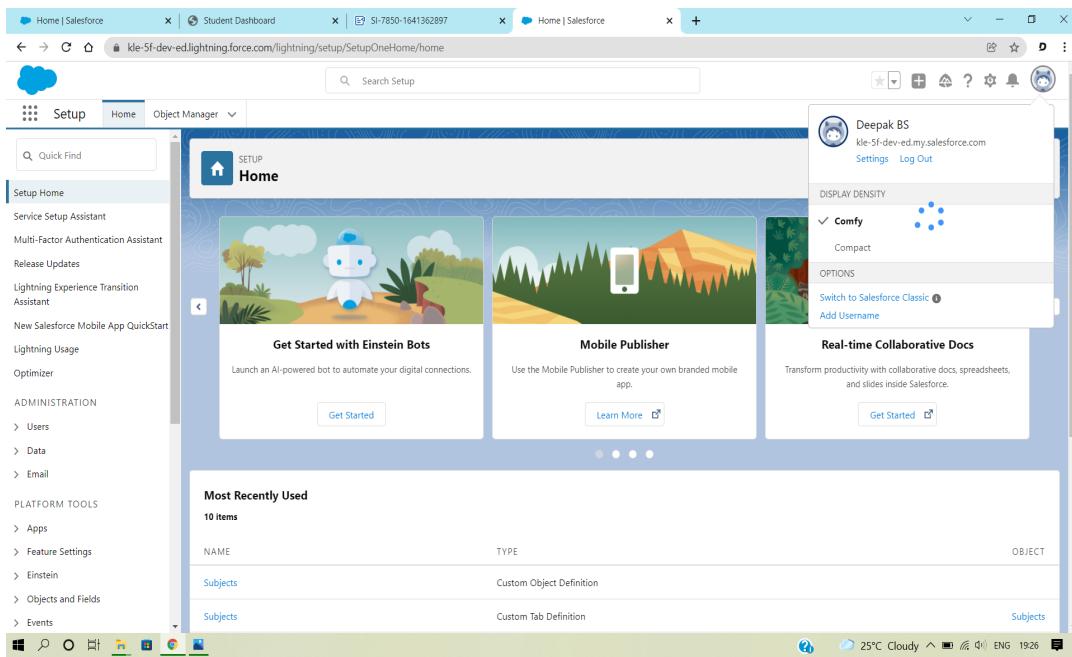
COLLEGE MANAGEMENT APPLICATION

Account Activation:



COLLEGE MANAGEMENT APPLICATION

logged in:



COLLEGE MANAGEMENT APPLICATION

Day 2 :

Topic :Custom Object Creation

Milestone/Activity : Create the following Custom Objects For the Application.

1. College
2. Application Form.
3. Students.
4. Subjects.

Detailed Description: Create The Custom Objects For The Application

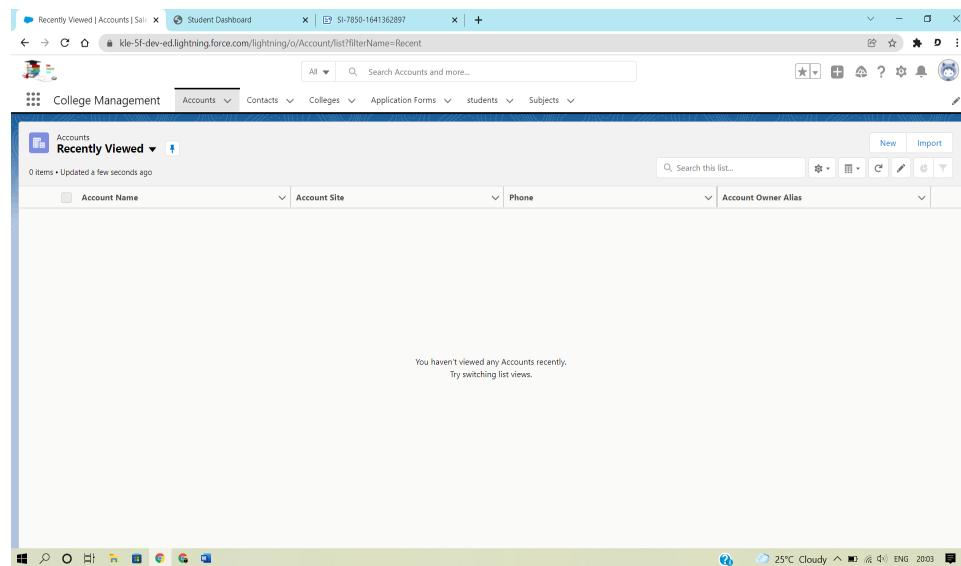
Create Fields On College Object

Create Fields On Application Form Object

Create Fields On Student Object

Application name is college management

1. CreatedThe Custom Objects For The Application



COLLEGE MANAGEMENT APPLICATION

2.created fields on college object:

Field Name	Data Type	Required	Values
Record Info	Text		
College Fees	Currency(7,2)	Yes	
Hostel Fees	Currency(6,2)	Yes	
College Name	Picklist(Refer Business Logic In Milestones)	Yes	
Email	Picklist		blr@mit.co.in hyd@mit.co.in mum@mit.co.in maa@mit.co.in ccu@mit.co.in del@mit.co.in
Capacity Of Students	Picklist		500-1000, 1000-2500, 2500-6000, 6000-10000

The screenshot shows the Salesforce Setup interface for the College object. The left sidebar lists various setup categories such as Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, and Triggers. The main content area is titled 'Fields & Relationships' and shows 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
college_email	college_email_c	Picklist	college_name	
college_fee	college_fee_c	Currency(7, 2)		
college_name	college_name_c	Picklist		
Created By	CreatedById	Lookup(User)		
hostel_fee	hostel_fee_c	Currency(6, 2)		
Last Modified By	LastModifiedById	Lookup(User)		
no_of_seats	no_of_seats_c	Picklist	college_name	
Owner	OwnerId	Lookup(User,Group)		✓
Record Info	Name	Text(80)		✓

COLLEGE MANAGEMENT APPLICATION

a. creating validation rule for college object:

Create a validation rule on the college object such that the college name and record info should have the same name.

TEXT(College_Name__c) <> Name.

The screenshot shows the Salesforce Object Manager interface. A validation rule named 'college_name_similar' has been created for the 'College' object. The rule checks if the 'College Name' field value does not match the record's 'Name'. It includes an error message: 'please use the college name in record info'. The rule is active and was created by 'Deeptak BS' on 1/10/2022 at 7:33 AM.

b. creating globalpicklist for email id field of college object

The screenshot shows the Salesforce Setup interface under 'Data' settings. A new 'Global Value Set' is being created for the 'college_name' field. The 'Values' section lists various college names like MIT-BLR, MIT-MAA, etc., each associated with an API name and modified by 'Deeptak BS' on 1/5/2022 at 6:54 AM. The 'Fields Where Used' section shows that this picklist is used for the 'college_name' field of the 'College' object.

COLLEGE MANAGEMENT APPLICATION

3.creating fields on application form object:

Create the following fields on the application form object.

Field Name	Data Type	Required	Values
Application Form ID	Autonumber		F-{000000} Starting Number=1
Address	Text(255)	Yes	
College	Master-Detail(College)	Yes	
College Fees	Formula(Currency)		
Hostel Fees	Formula(Currency)		
Date of Birth	Date	Yes	
Email	Email(Unique)	Yes	
Guardian Name	Text(30)	Yes	
Looking For Hostel Stay	Checkbox(default=Uncheck)		
Ready To Join	Checkbox(default=Uncheck)		
Student Name	Text(30)	Yes	
Phone	Phone	Yes	

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

- Setup > Object Manager**: Application Form
- Fields & Relationships**: Fields & Relationships
- 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, Restriction Rules, Triggers, Validation Rules.
- Fields & Relationships** table:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
College	College__c	Master-Detail(College)		✓
college fee	college_fee__c	Formula (Currency)		✓
Created By	CreatedById	Lookup(User)		✓
Date of Birth	Date_of_Birth__c	Date		✓
Email	Email__c	Email (Unique)		✓
Form Number	Name	Auto Number		✓
Guardian Name	Guardian_Name__c	Text(30)		✓
Hostel Fees	Hostel_Fees__c	Formula (Currency)		✓
Last Modified By	LastModifiedById	Lookup(User)		✓
Looking For Hostel Stay	Looking_For_Hostel_Stay__c	Checkbox		✓
Ready To Join	Ready_To_Join__c	Checkbox		✓
Student Name	Student_Name__c	Text(20)		✓

COLLEGE MANAGEMENT APPLICATION

4.creating field on Student object:

Create the following fields on the student object.

Field Name	Data Type	Required	Values
Student Name	Text		
Address	Text(255)	Yes	
Application Form	Lookup(Application Form)		
College Name	Formula(Text)		
Date Of Birth	Date	Yes	
Guardian Name	Text(30)	Yes	
Phone	Phone	Yes	

The screenshot shows the Salesforce Object Manager interface. The top navigation bar includes tabs for 'Setup', 'Home', and 'Object Manager'. The current view is 'student'. On the left, a sidebar lists various setup categories like 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, Triggers, and Validation Rules. The main content area is titled 'Fields & Relationships' and displays a table of 10 fields. The table columns 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 Area(255)		
Application Form	Application_Form_c	Lookup(Application Form)		
college Name	college_Name_c	Formula (Text)		
Created By	CreatedById	Lookup(User)		
Date of Birth	Date_of_Birth_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	Name	Text(80)		

COLLEGE MANAGEMENT APPLICATION

5.creating field on subject object:

Create the following fields On the Subject object.

Field Name	Data Type	Required	Values
Subject ID	AutoNumber		S-{000000} Starting Number=1
Paper 1	Picklist(Refer Business Logic Milestone)		
Paper2	Picklist(Refer Business logic Milestone)		
Student	Lookup(Student)		

The screenshot shows the Salesforce Object Manager interface. The left sidebar lists various setup categories like Details, Fields & Relationships, Page Layouts, etc. The main content area is titled 'Fields & Relationships' and displays a table of fields for the 'Subjects' object. The table includes columns for Field Label, Field Name, Data Type, Controlling Field, and Indexed status. The fields listed are:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedBy	Lookup(User)		
Last Modified By	LastModifiedBy	Lookup(User)		
Owner	OwnerId	Lookup(User/Group)		✓
Paper 1	Paper_1_c	Picklist		
Paper2	Paper2_c	Picklist		
Subject ID	Subject_ID_c	Auto Number		
Subjects Name	Name	Text(80)		✓

COLLEGE MANAGEMENT APPLICATION

a. creating global picklist value for subject object

b) Paper1

Picklist Value Name	Values
Paper 1	APEX JAVA C C++

c) Paper2

Picklist Value Name	Values
Paper2	MATHEMATICS ENGLISH STATISTICS

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

- Page Header:** Student Dashboard, Picklist Value Sets | Salesforce, Profiles, Recordtypes, Automation, Picklist Value Sets | Salesforce, Student Dashboard.
- Search Bar:** Search Setup.
- Left Navigation:** Setup, Home, Object Manager, Data (Picklist Settings, State and Country/Territory Picklists), Objects and Fields (Picklist Value Sets).
- Main Content:**
 - Section:** SETUP Picklist Value Sets
 - Section:** Picklist Value Sets
 - Description:** Global picklist value sets let you share the values across objects. Base custom picklist fields on a global value set to inherit its values. The value set is restricted so users can't add unapproved values through the API.
 - View:** All, Check New View.
 - Global Value Sets:** A table listing existing global value sets:

Action	Label	Description
Edit Del	college.name	
Edit Del	Paper1	
Edit Del	Paper2	
- Page Footer:** https://kle-5f-dev-ed.lightning.force.com/one/app#/setup/Picklists/home, 23°C Cloudy, ENG, 19:51.

COLLEGE MANAGEMENT APPLICATION

Day 3 :

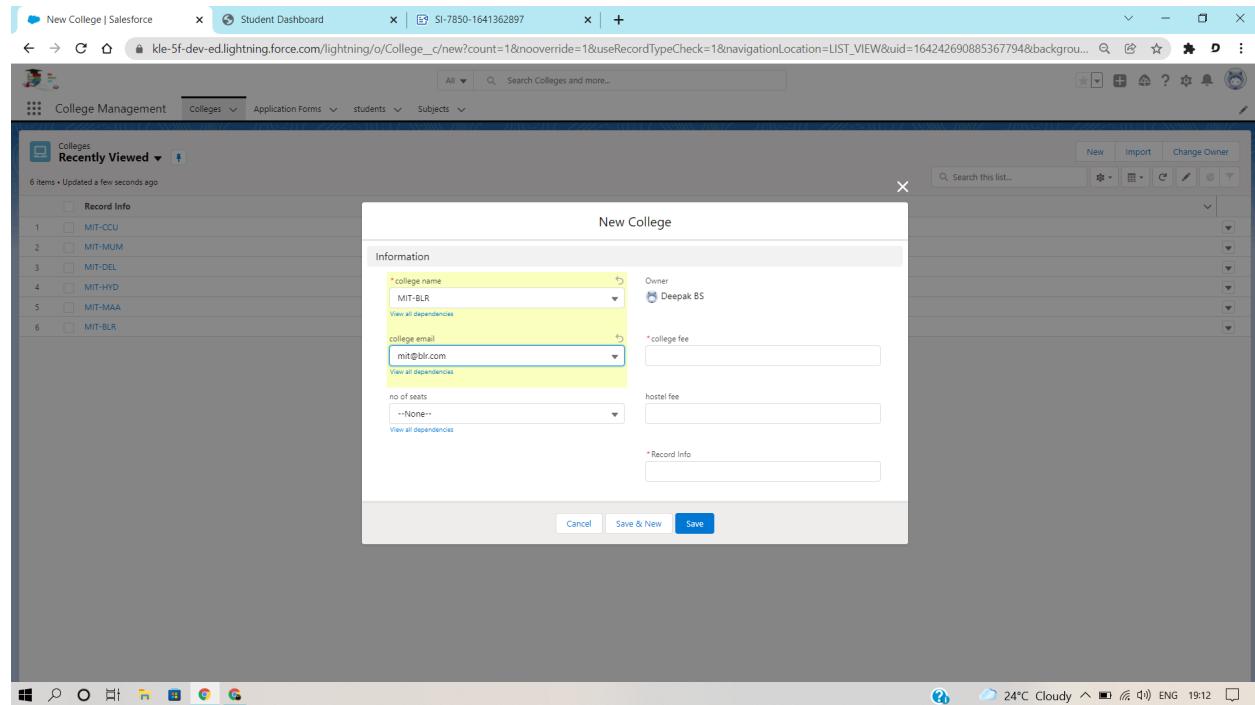
Topic : Adding Business to logic

Milestone/Activity : creating field dependencies

Created Field Dependencies

1. Created field dependency between college Name and Email, where the controlling field is college Name and dependent field is Email. Selected the email ids according to the college names.

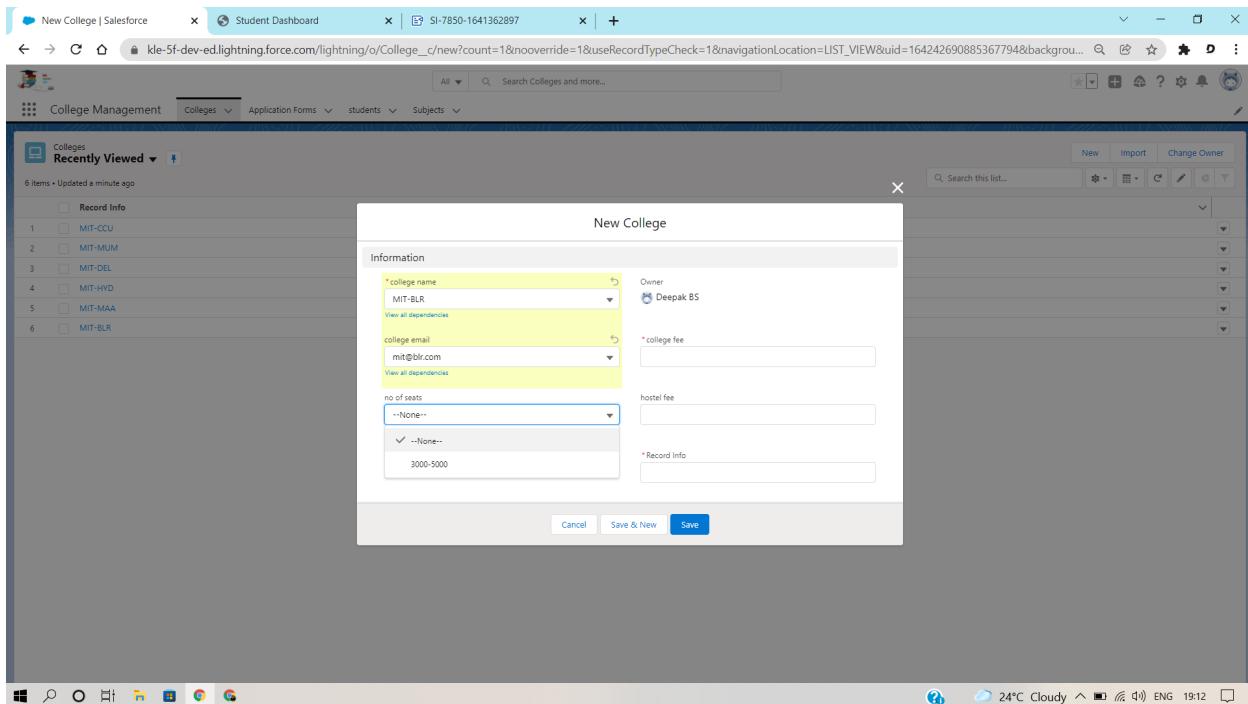
Screenshot :



2. Create field dependency between college Name and capacity of students, where the controlling field is college Name and dependent field is Capacity of Students. Select the values according to your wish.

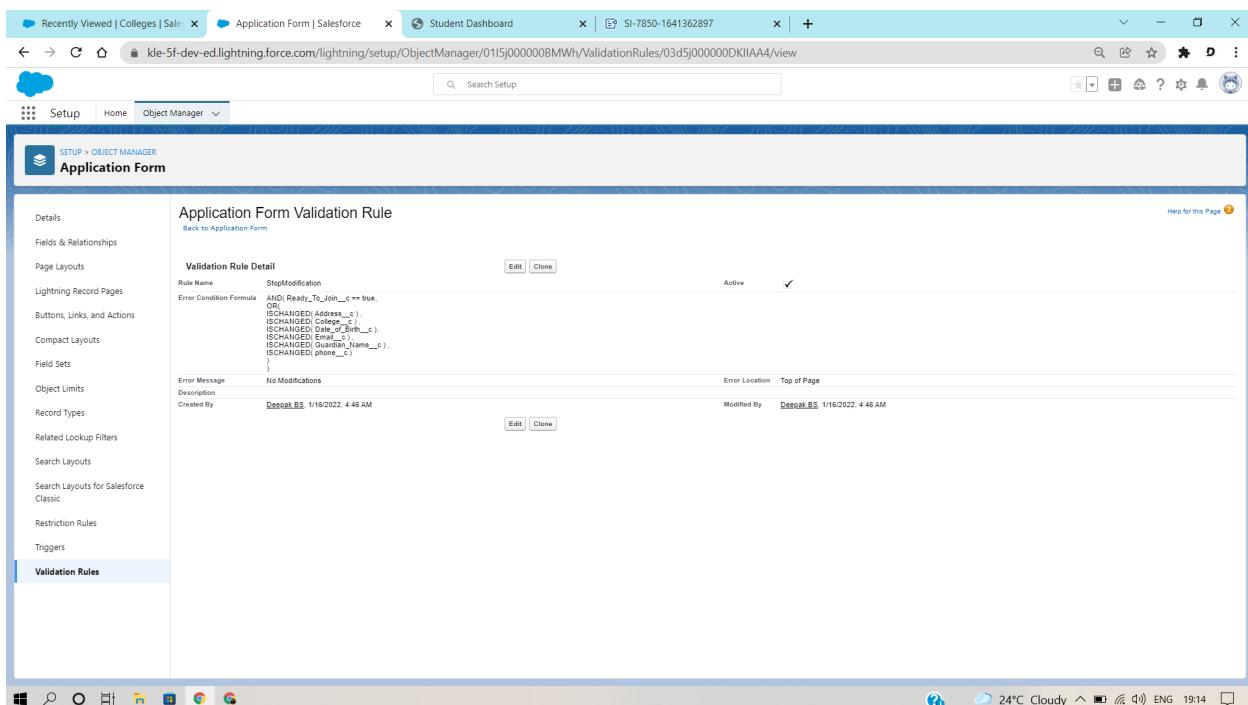
COLLEGE MANAGEMENT APPLICATION

Screenshot :



creating a validation rule on the application form object to stop any modification on the application form once a student record is created.

Screenshot :



COLLEGE MANAGEMENT APPLICATION

Day 4 :

Topic :

Process Automation and Flow

Milestone/Activity :

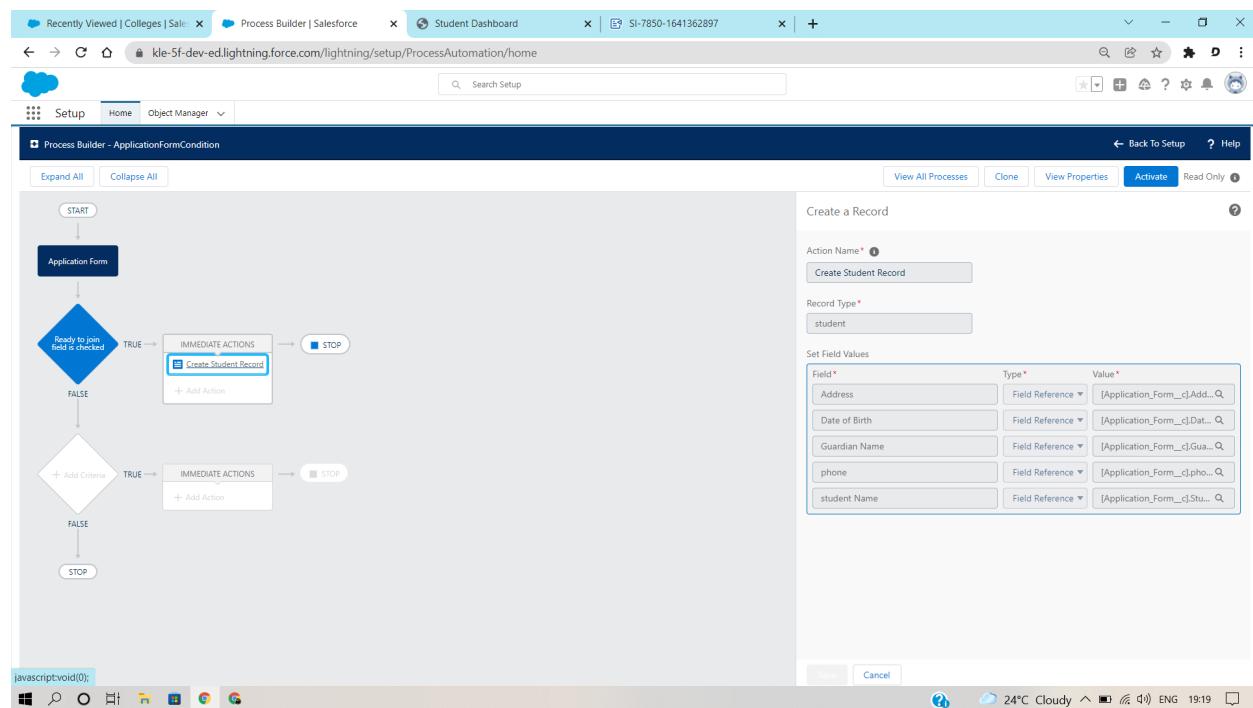
Process Automation

Created The Student Record Using Flow

Detailed Description : .Created pocess automation for application form

Created an automation process such that when the "ready to join" field is checked on the application form object we need to create the student record automatically with the information specified in the application form record.

2. Go to Setup -> select "Process Builder" from quick find. Create a Process Builder on the "Application Form" object with a condition as "When a record change". And select "When a record is created or edited".
 - . In the diamond shape box(called nodes), select the criteria which trigger the Process builder to fire. In our example, it is "When Ready to Join field is checked."

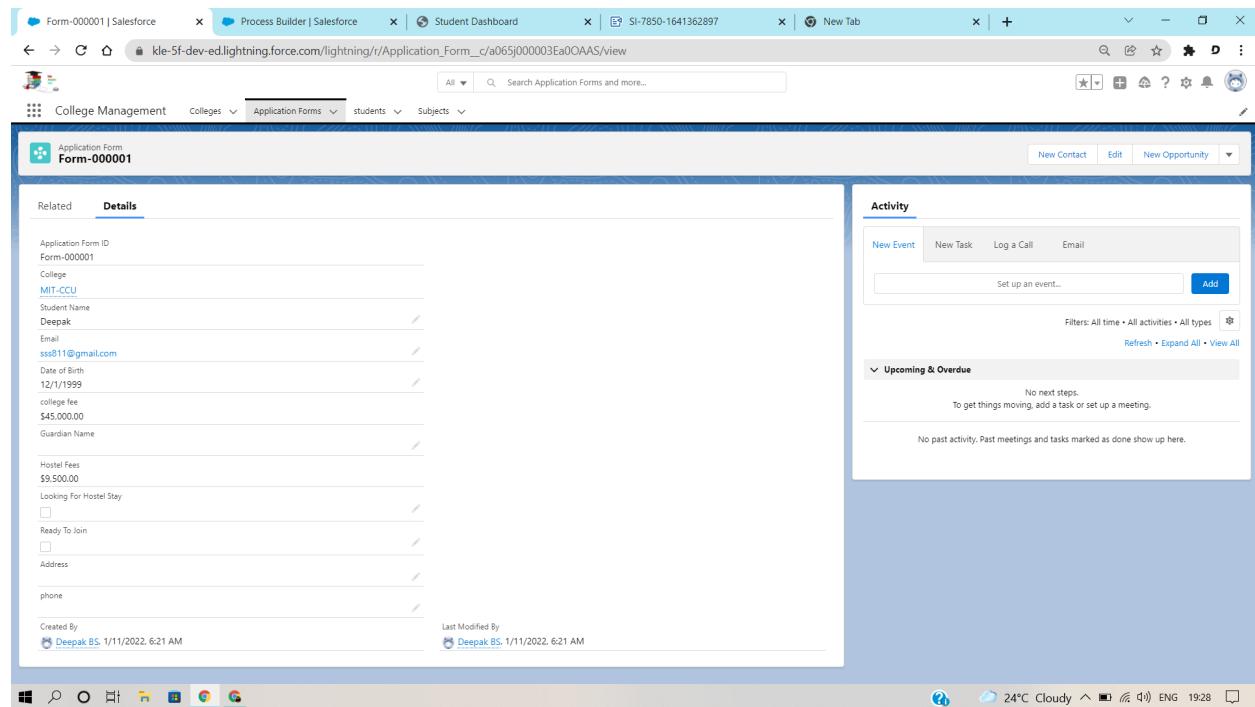


COLLEGE MANAGEMENT APPLICATION

Created The Student Record Using Flow:

- . First deactivated the process builder which we created earlier.
- . Now search for flows and select new flow ->record triggered flow
- . 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.
- . Now create a variable named student in the resource section.
- . Added Assignment
- .Finally add create records.

Screnshot :added assignment



COLLEGE MANAGEMENT APPLICATION

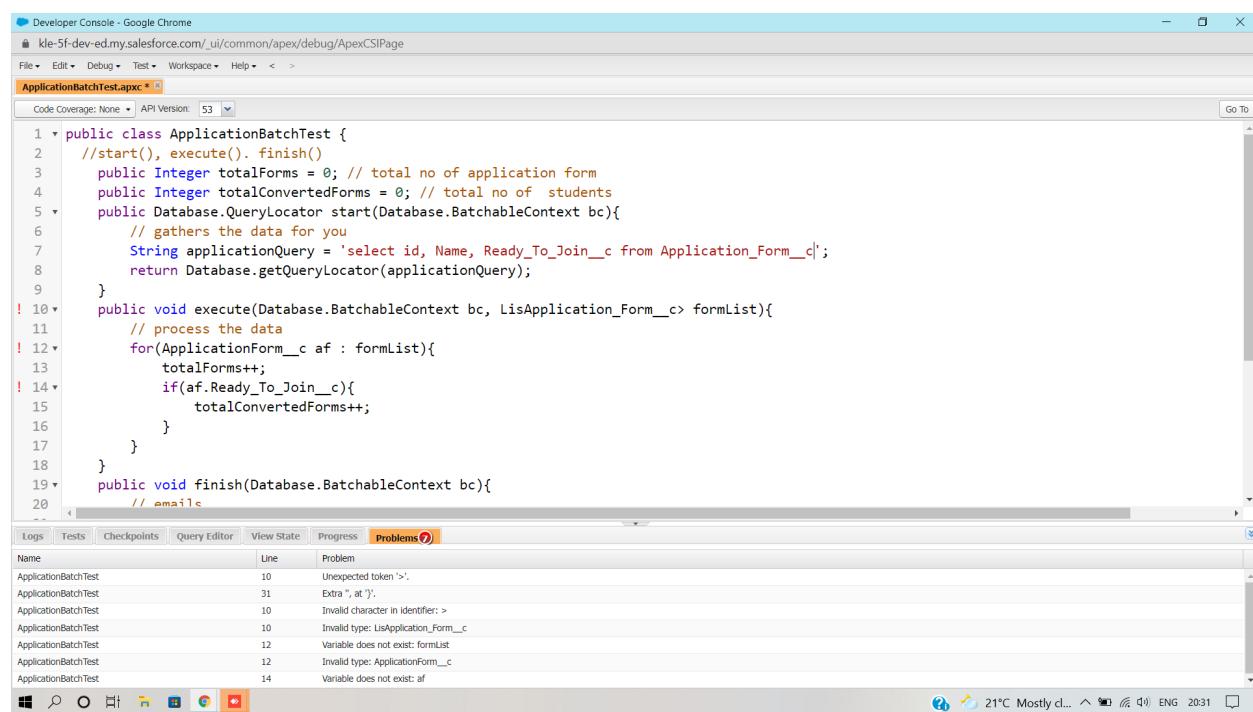
Day 5 :

Topic: Batch Apex

Milestone/Activity: Batch apex

Detailed Description :

From the developer console created a new apex class and enter the following code



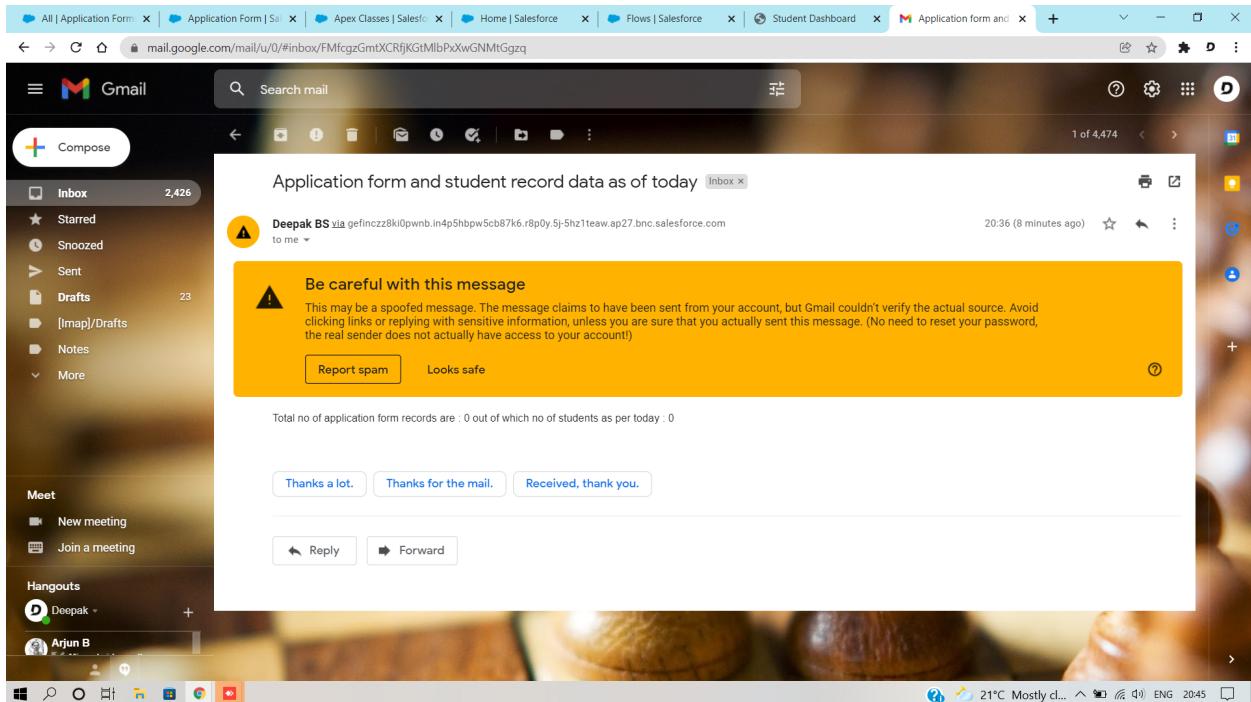
The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is kle-5f-dev-ed.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage. The tab title is "ApplicationBatchTest.apxc". The code editor contains the following Apex class:

```
1 public class ApplicationBatchTest {
2     //start(), execute(). finish()
3     public Integer totalForms = 0; // total no of application form
4     public Integer totalConvertedForms = 0; // total no of students
5     public Database.QueryLocator start(Database.BatchableContext bc){
6         // gathers the data for you
7         String applicationQuery = 'select id, Name, Ready_To_Join__c from Application_Form__c';
8         return Database.getQueryLocator(applicationQuery);
9     }
10    public void execute(Database.BatchableContext bc, List<Application_Form__c> formList){
11        // process the data
12        for(ApplicationForm__c af : formList){
13            totalForms++;
14            if(af.Ready_To_Join__c){
15                totalConvertedForms++;
16            }
17        }
18    }
19    public void finish(Database.BatchableContext bc){
20        // emails
21    }
}
```

The code has several syntax errors, indicated by red highlights and error messages in the "problems" tab:

Name	Line	Problem
ApplicationBatchTest	10	Unexpected token '>'.
ApplicationBatchTest	31	Extra ", at '}'.
ApplicationBatchTest	10	Invalid character in identifier: >
ApplicationBatchTest	10	Invalid type: List<Application_Form__c>
ApplicationBatchTest	12	Variable does not exist: formList
ApplicationBatchTest	12	Invalid type: ApplicationForm__c
ApplicationBatchTest	14	Variable does not exist: af

COLLEGE MANAGEMENT APPLICATION



A Schedular Class :

From the developer console created a new apex class as applicationschedule

Screenshot :

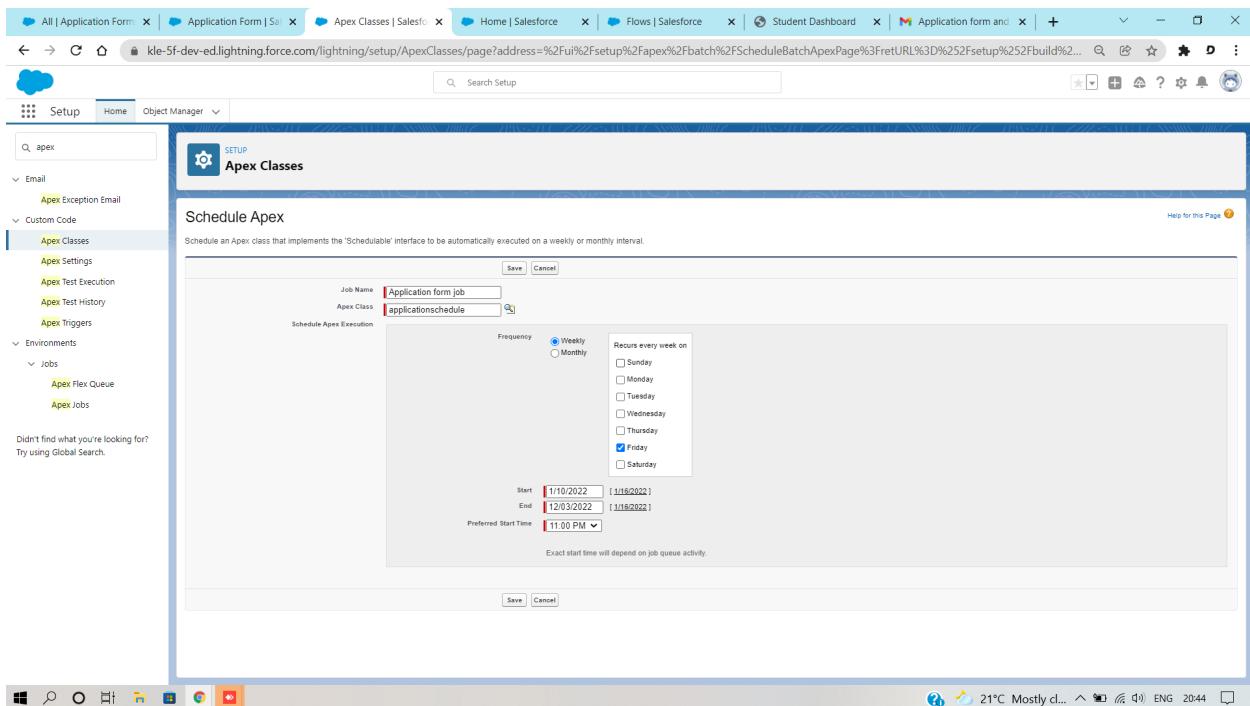
```
1 Enter your search term
2
3 public class applicationschedule implements Schedulable{
4
5
6     public void execute(SchedulableContext sc){
7
8         ApplicationBatchTest abt = new ApplicationBatchTest();
9
10        Database.executeBatch(abt, 400); // 200 to 2000
11
12
13
14    }
15
16 }
17 }
```

The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is 'klessnjalingappcollege2-dev-ed.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage'. The code editor window displays the 'applicationschedule.apxc' file. The code itself is a simple Apex class named 'applicationschedule' that implements the 'Schedulable' interface. It contains a single method 'execute' that creates an instance of 'ApplicationBatchTest' and executes a batch job with a size of 400. The developer console interface includes tabs for 'Logs, Tests, and Problems' and a search bar at the bottom.

COLLEGE MANAGEMENT APPLICATION

From setup search for apex class and click on schedule jobs and fill the details as per the requirements.

Screenshot :



COLLEGE MANAGEMENT APPLICATION

Day 6 :

Topic : Lightning Web Components

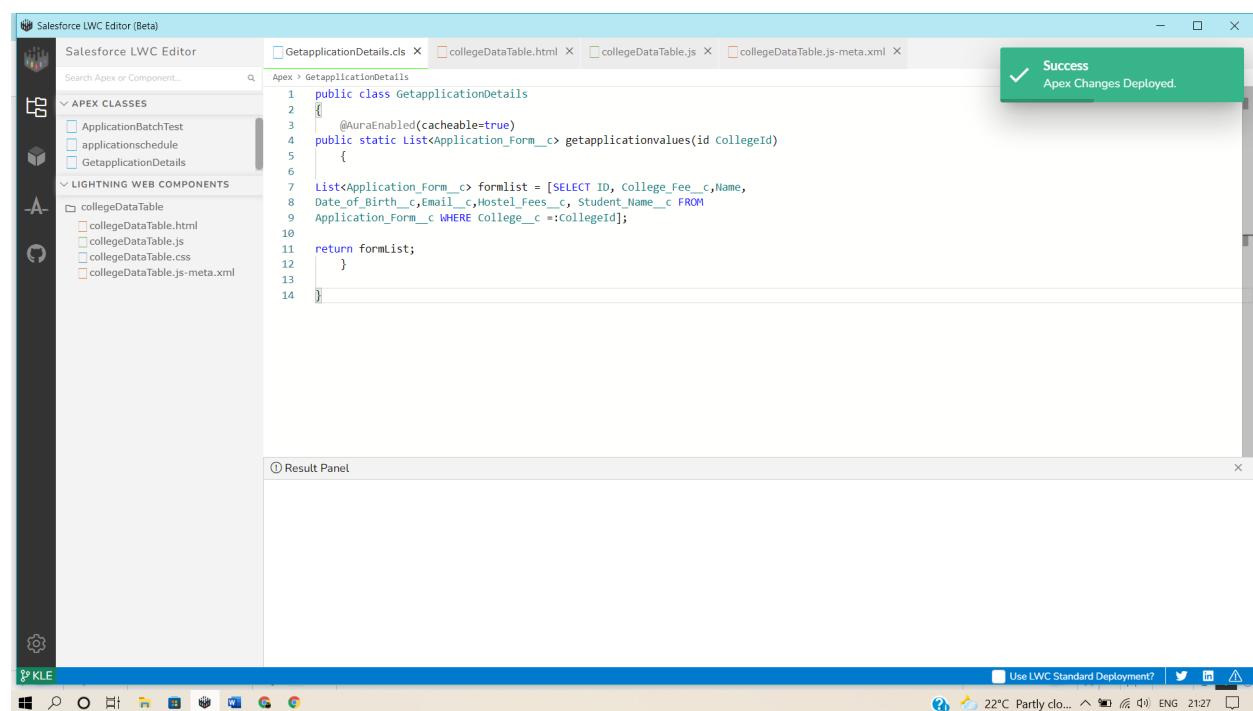
Milestone/Activity : Created College DataTable Component(APEX CLASS)
Created College DataTable Component (HTML FILE)
Created College DataTable Component(JAVA SCRIPT FILE)
Created College DataTable Component(META FILE)

Detailed Discription :

Created College DataTable Component(APEX CLASS)

Here we are going to create the college data table, for this we need to create an apex class, from which we are going to retrieve the data and Html, javascript files for UI .

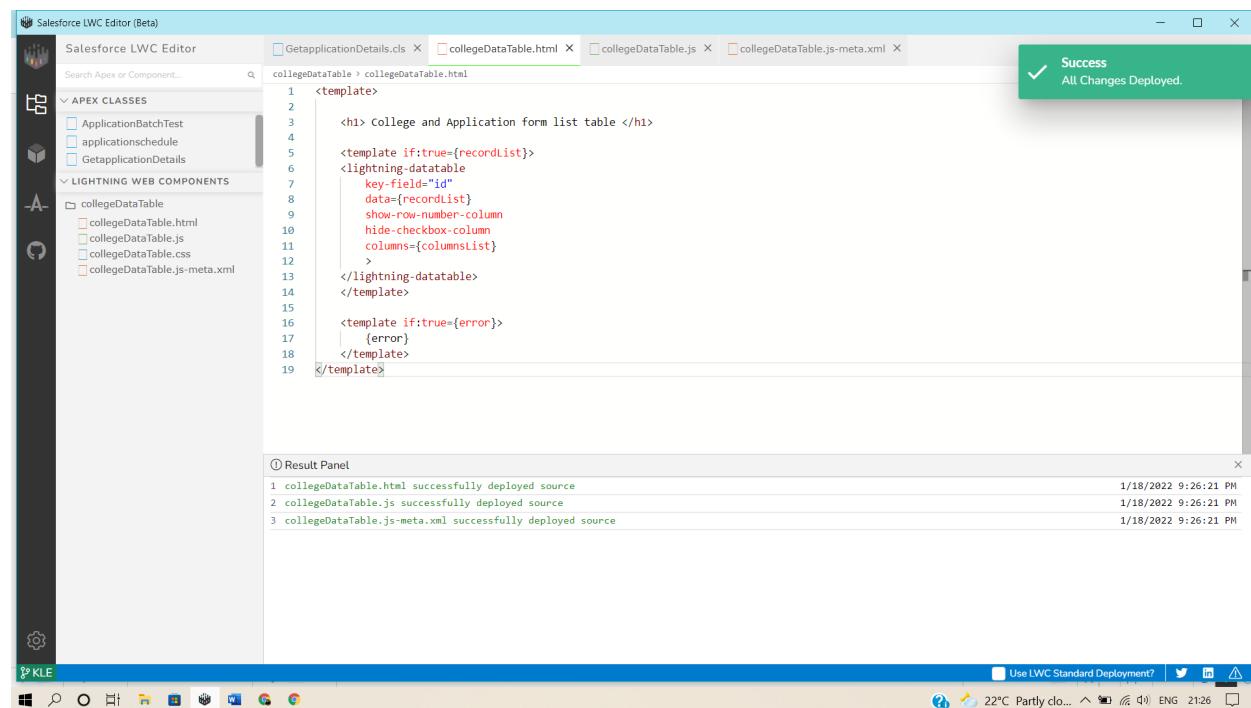
Screenshot :



COLLEGE MANAGEMENT APPLICATION

Created College DataTable Component (HTML FILE)

Screenshot :



The screenshot shows the Salesforce LWC Editor (Beta) interface. The sidebar on the left lists 'APEX CLASSES' and 'LIGHTNING WEB COMPONENTS'. Under 'LIGHTNING WEB COMPONENTS', there is a folder 'collegeDataTable' containing 'collegeDataTable.html', 'collegeDataTable.js', 'collegeDataTable.css', and 'collegeDataTable.js-meta.xml'. The main area is the code editor for 'collegeDataTable.html', which contains the following LWC template code:

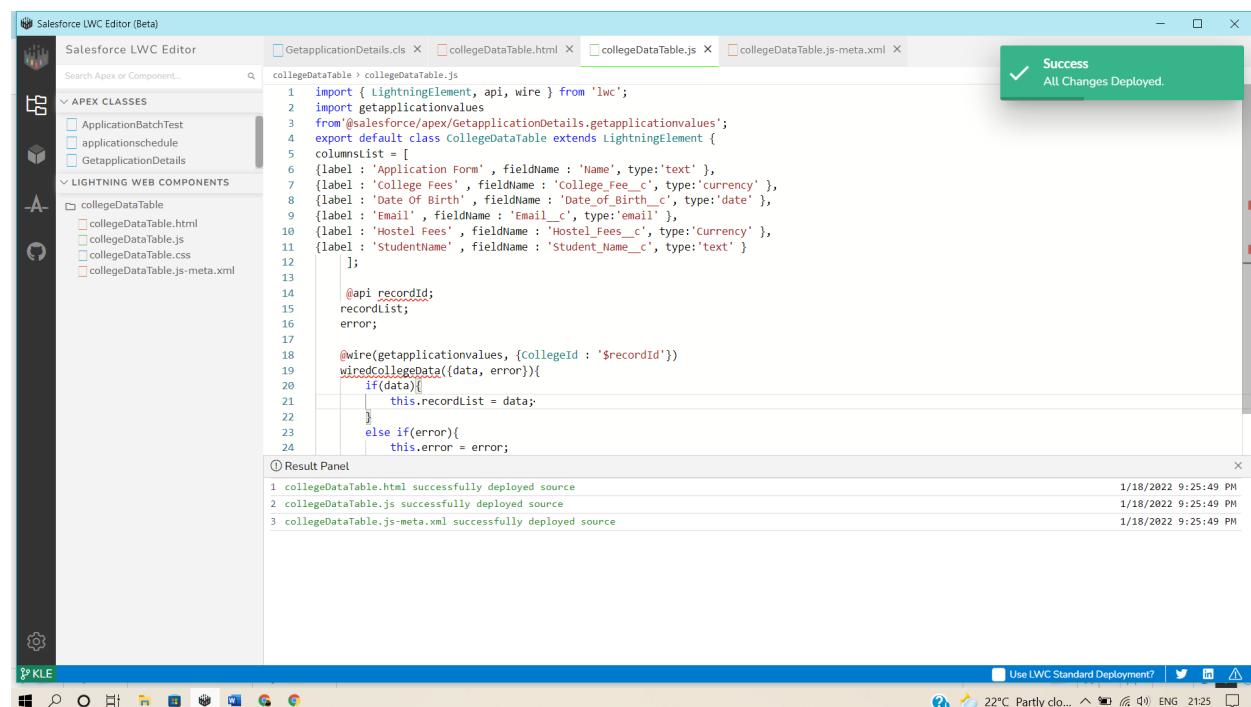
```
<template>
    <h1> College and Application form list table </h1>
    <template if:true={recordList}>
        <lightning-datatable
            key-field="id"
            data={recordList}
            show-row-number-column
            hide-checkbox-column
            columns={columnsList}
      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

Below the code editor is the 'Result Panel' which displays the deployment status of three files:

File	Status	Date
collegeDataTable.html	successfully deployed source	1/18/2022 9:26:21 PM
collegeDataTable.js	successfully deployed source	1/18/2022 9:26:21 PM
collegeDataTable.js-meta.xml	successfully deployed source	1/18/2022 9:26:21 PM

Created College DataTable Component(JAVA SCRIPT FILE)

Screenshot :



The screenshot shows the Salesforce LWC Editor (Beta) interface. The sidebar on the left lists 'APEX CLASSES' and 'LIGHTNING WEB COMPONENTS'. Under 'LIGHTNING WEB COMPONENTS', there is a folder 'collegeDataTable' containing 'collegeDataTable.html', 'collegeDataTable.js', 'collegeDataTable.css', and 'collegeDataTable.js-meta.xml'. The main area is the code editor for 'collegeDataTable.js', which contains the following LWC JavaScript code:

```
import { LightningElement, api, wire } from 'lwc';
import getapplicationvalues;
import {salesforce/apex/GetapplicationDetails.getapplicationvalues'};
export default class CollegeDataTable extends lightningElement {
    columnsList = [
        {label : 'Application Form' , fieldName : 'Name', type:'text' },
        {label : 'College Fees' , fieldName : 'College_Fee__c', type:'currency' },
        {label : 'Date Of Birth' , fieldName : 'Date_of_Birth__c', type:'date' },
        {label : 'Email' , fieldName : 'Email__c', type:'email' },
        {label : 'Hostel Fees' , fieldName : 'Hostel_Fees__c', type:'Currency' },
        {label : 'StudentName' , fieldName : 'Student_Name__c', type:'text' }
    ];
    @api recordId;
    recordList;
    error;
    @wire(getapplicationvalues, {CollegeId : '$recordId'})
    wiredcollegeData({data, error}){
        if(data){
            this.recordList = data;
        }
        else if(error){
            this.error = error;
        }
    }
}
```

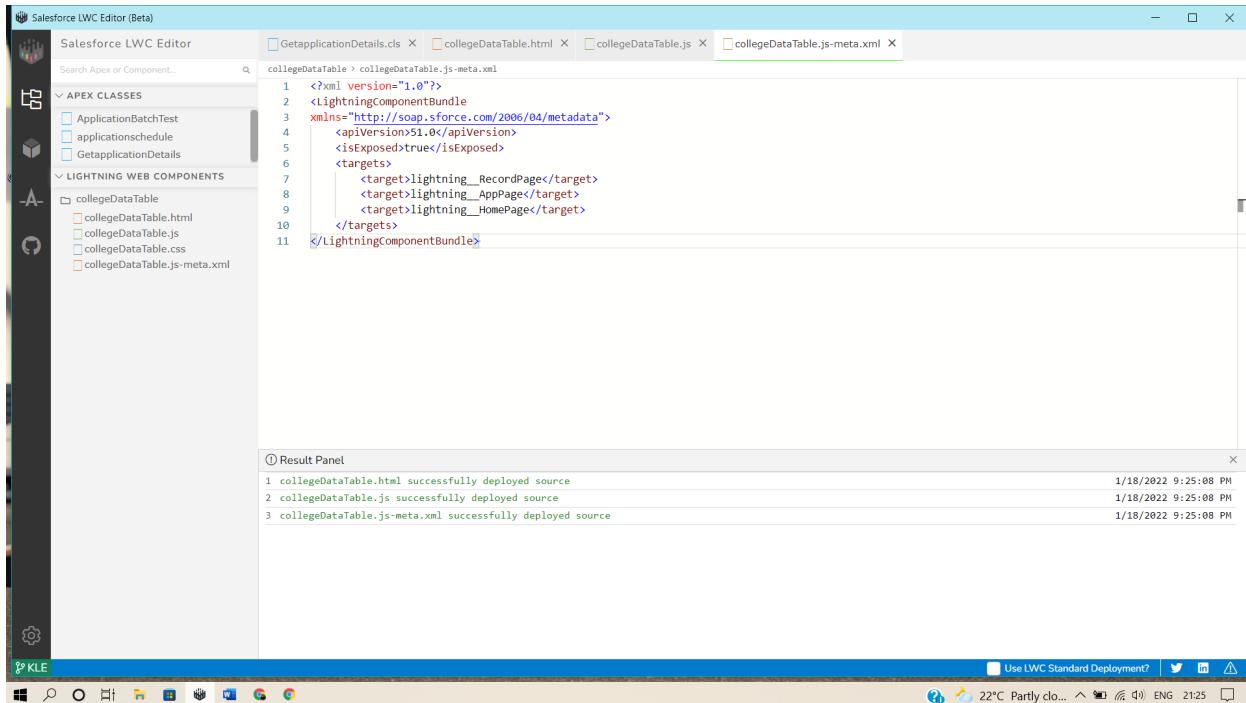
Below the code editor is the 'Result Panel' which displays the deployment status of three files:

File	Status	Date
collegeDataTable.html	successfully deployed source	1/18/2022 9:25:49 PM
collegeDataTable.js	successfully deployed source	1/18/2022 9:25:49 PM
collegeDataTable.js-meta.xml	successfully deployed source	1/18/2022 9:25:49 PM

COLLEGE MANAGEMENT APPLICATION

Created College DataTable Component(META FILE)

Screenshot :



Screenshot :

Final Output

