

## Project Title : College Management Application

**Topic :** Created Salesforce Developer Org To Get Started

Milestone / Activities :

- a)Created Developer Account
- b)Account Activation
- c>Login To Your Salesforce Account

Detailed Description:

a)Created a developer org in salesforce.

Go to developers.salesforce.com Click on sign up

On the sign up form, enter the following details :

First name & Last name :

Email :

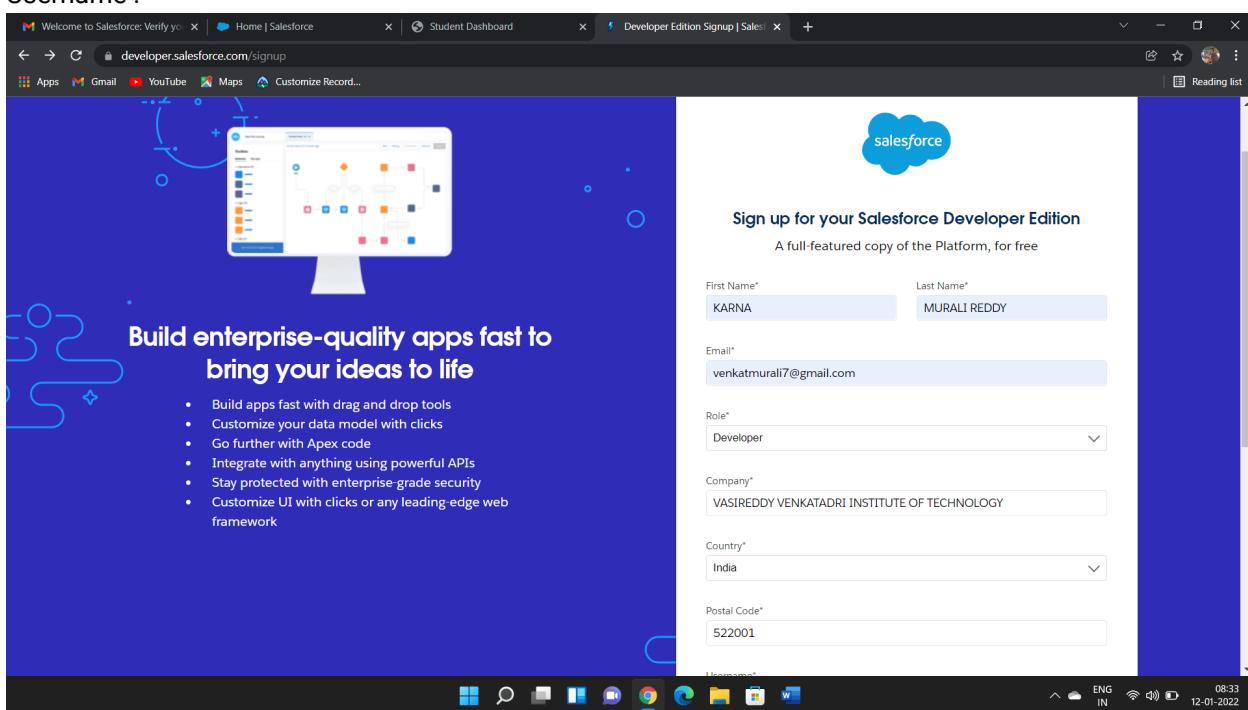
Role : Developer

Company : VVIT

Country : India

Postal Code :

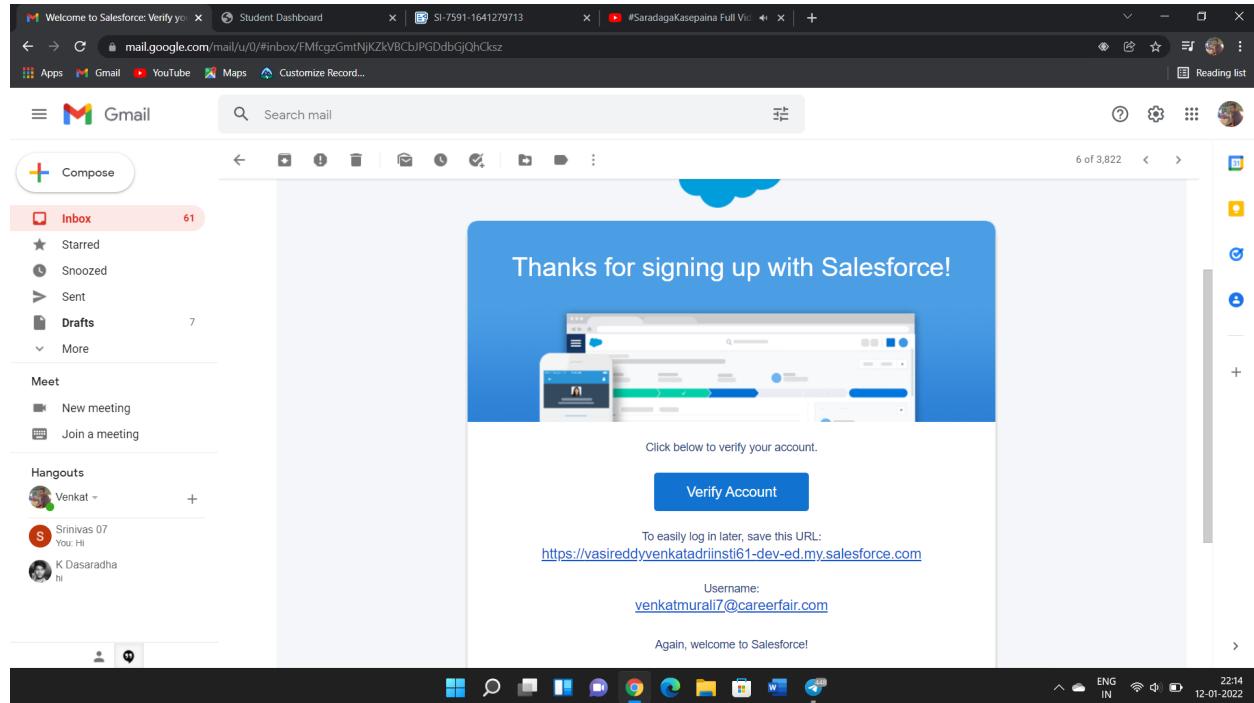
Username :



## Project Title : College Management Application

### b) Account Activation:

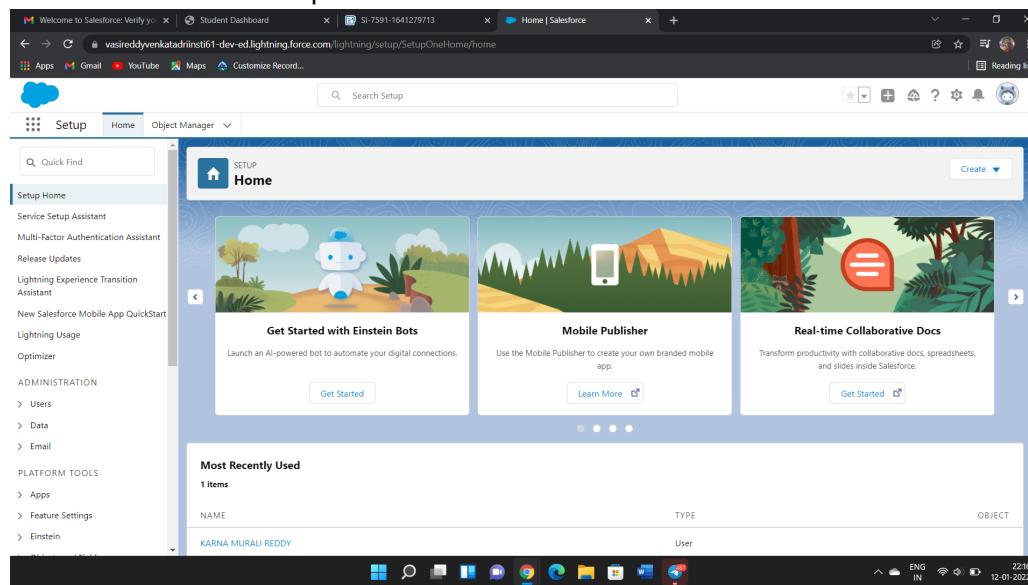
Go to the inbox of the email that we used while signing up. Click on the verify account to activate our account.



### c) Login To Your Salesforce Account

Go to salesforce.com and click on login.

Enter the username and password



that we just created

**Topic :** Custom Object Creation

**Milestone / Activities:**

- a) Create The Custom Objects For The Application
- b) Create Fields On College Object
- c) Create Fields On Application Form Object

**Detailed Description:**

► Created The Custom Objects For The Application

Created the following Custom Objects For the Application.

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

► College Object

Created the following fields on the 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		mit@maa.com mit@ccu.com mit@mum.com mit@del.com mit@blr.com mit@hyd.com
Capacity Of Students	Picklist		500-1500,1500-3000,3000-5000,5000-8000,8000-12000

## Project Title : College Management Application

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

- Header:** Search Setup, Home, Object Manager.
- Breadcrumb:** SETUP > OBJECT MANAGER College
- 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
  - Triggers
- Table:** Fields & Relationships
 

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)		✓

### Create Fields On Application Form Object

Create the following fields on the application form object.

Field Name	Data Type	Required	Values
Application Form ID	Auto number		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	

## Project Title : College Management Application

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

- Header: Search Setup, Home, Object Manager
- Breadcrumbs: SETUP > OBJECT MANAGER
- Page Title: Application Form
- Left sidebar (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.
- Main Content: Fields & Relationships table with 15 items, sorted by Field Label.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Address	Address_c	Text(255)		
Application Form ID	Application_Form_ID_c	Auto Number		
College	College_c	Master-Detail(College)	✓	
College fees	College_fees__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)		

## Day 3 :

### Topic : Custom Object Creation, Adding Business Logic To Application

#### Milestone / Activities:

- Create Fields On Student Object
- Create Fields On Subject Object
- Creating Global Picklist Value Sets
- Creating Field Dependencies
- Creating Validation Rules

#### Detailed Description:

##### ► Created Fields On Student Object

Create the following fields On the Subject object.

## Project Title : College Management Application

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 Setup interface with the following details:

- Header:** Includes a cloud icon, a search bar ("Search Setup"), and various navigation icons.
- Breadcrumbs:** SETUP > OBJECT MANAGER > Student
- Left Sidebar:** A navigation menu with items like 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, Restriction Rules, and Triggers.
- Fields & Relationships Page:**
  - Section Header:** Fields & Relationships (10 Items, Sorted by Field Label)
  - Table Headers:** FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, INDEXED
  - Table Data:**

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	CreatedBy	Lookup(User)		
Date Of Birth	Date_Of_Birth__c	Date		
Guardian Name	Guardian_Name__c	Text(30)		
Last Modified By	LastModifiedBy	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)	✓	
Phone	Phone__c	Phone		
Student Name	Name	Text(80)	✓	

### ➤ Create Fields On Subject Object

Created the following fields On the Subject object.

## Project Title : College Management Application

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 for the 'Subject' object. The left sidebar lists various setup options like Page Layouts, Lightning Record Pages, etc. The main area displays the 'Fields & Relationships' section with the following details:

- Fields & Relationships**: 8 items, Sorted by Field Label
- FIELD LABEL**, **FIELD NAME**, **DATA TYPE**, **CONTROLLING FIELD**, **INDEXED**
- Created By: CreatedByld, Lookup(User),
- Last Modified By: LastModifiedByld, Lookup(User),
- Owner: OwnerId, Lookup(User,Group), ✓
- Paper1: Paper1\_c, Picklist,
- Paper2: Paper2\_c, Picklist,
- Student: Student\_c, Lookup(Student), ✓
- Subject ID: Subject\_ID\_c, Auto Number,
- Subject Name: Name, Text(80), ✓

### ► Creating Global Picklist Value Sets

1. Create the following global picklist value sets for the application.
- a)College

## Project Title : College Management Application

Picklist Value Name	Values
College	MIT-HYD MIT-BLR MIT-MUM MIT-MAA MIT-DEL MIT-CCU

The screenshot shows the Salesforce Setup interface with the 'Picklist Value Sets' page open. The left sidebar navigation includes 'Data' (Picklist Settings, State and Country/Territory, Picklists), 'Feature Settings' (Topics, Topic Assignment Triggers, Topic Triggers, Topics for Objects), and 'Objects and Fields' (Picklist Value Sets). The 'Picklist Value Sets' option is selected. The main content area displays a 'Global Value Set' named 'College Name'. The 'Information' section shows the Label as 'College Name', Name as 'College\_Name', and Description as blank. The 'Values Used' section shows 6 active and inactive picklist values. A table lists these values with columns for Action, Values, API Name, Default, Chart Colors, and Modified By. All values were modified by 'Durga Prasad Avula' on 1/4/2022 at 11:14 PM.

Action	Values	API Name	Default	Chart Colors	Modified By
Edit   Del   Deactivate	MIT-BLR	MIT-BLR	<input type="checkbox"/>	Assigned dynamically	Durga Prasad Avula, 1/4/2022, 11:14 PM
Edit   Del   Deactivate	MIT-MAA	MIT-MAA	<input type="checkbox"/>	Assigned dynamically	Durga Prasad Avula, 1/4/2022, 11:14 PM
Edit   Del   Deactivate	MIT-HYD	MIT-HYD	<input type="checkbox"/>	Assigned dynamically	Durga Prasad Avula, 1/4/2022, 11:14 PM
Edit   Del   Deactivate	MIT-DEL	MIT-DEL	<input type="checkbox"/>	Assigned dynamically	Durga Prasad Avula, 1/4/2022, 11:14 PM
Edit   Del   Deactivate	MIT-MUM	MIT-MUM	<input type="checkbox"/>	Assigned dynamically	Durga Prasad Avula, 1/4/2022, 11:14 PM
Edit   Del   Deactivate	MIT-CCU	MIT-CCU	<input type="checkbox"/>	Assigned dynamically	Durga Prasad Avula, 1/4/2022, 11:14 PM

b)Paper1

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

## Project Title : College Management Application

**Picklist Value Sets**

**Global Value Set**

**Global Value Set Detail**

Action	Values	API Name	Default	Chart Colors	Modified By
Edit   Del   Deactivate	Apex	Apex	<input type="checkbox"/>	Assigned dynamically	Durga Prasad Avula, 1/5/2022, 10:53 PM
Edit   Del   Deactivate	Java	Java	<input type="checkbox"/>	Assigned dynamically	Durga Prasad Avula, 1/5/2022, 10:53 PM
Edit   Del   Deactivate	C	C	<input type="checkbox"/>	Assigned dynamically	Durga Prasad Avula, 1/5/2022, 10:53 PM
Edit   Del   Deactivate	C++	C++	<input type="checkbox"/>	Assigned dynamically	Durga Prasad Avula, 1/5/2022, 10:53 PM

**Values**

**Inactive Values**

c) Paper2

Picklist Value Name	Values
Paper2	MATHEMATICS ENGLISH STATISTICS

**Picklist Value Sets**

**Global Value Set**

**Global Value Set Detail**

Action	Values	API Name	Default	Chart Colors	Modified By
Edit   Del   Deactivate	MATHEMATICS	MATHEMATICS	<input type="checkbox"/>	Assigned dynamically	Durga Prasad Avula, 1/6/2022, 3:57 AM
Edit   Del   Deactivate	STATISTICS	STATISTICS	<input type="checkbox"/>	Assigned dynamically	Durga Prasad Avula, 1/6/2022, 3:58 AM
Edit   Del   Deactivate	ENGLISH	ENGLISH	<input type="checkbox"/>	Assigned dynamically	Durga Prasad Avula, 1/6/2022, 3:58 AM

**Values**

**Inactive Values**

No Inactive Values values defined.

<https://vvti62-dev-ed.my.salesforce.com/one/app#/alohaRedirect/0Nt5j000000CkX?sdtp=p1>

## Project Title : College Management Application

### ► Creating Field Dependencies

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

The screenshot shows the Salesforce Setup interface for creating field dependencies on the 'College' object. The left sidebar is titled 'Fields & Relationships' and lists various customization options. The main content area is titled 'College Field Dependencies' and displays a table of existing dependencies:

Action	Controlling Field	Dependent Field	Modified By
Edit   Del	College Name	College Email	Durga.Prasad.Ayula, 1/6/2022, 7:12 PM
Edit   Del	College Name	No Of Seats	Durga.Prasad.Ayula, 1/5/2022, 4:17 AM

## Project Title : College Management Application

The screenshots show the 'Edit Field Dependency' configuration for the 'College' object in Salesforce.

**Screenshot 1 (Controlling Field: College Name, Dependent Field: Email):**

- Controlling Field:** College Name
- Dependent Field:** Email
- Instructions:**
  - Double click on a cell to toggle its visibility for the Controlling Field value shown in the column heading.
  - To change multiple cells at once, select multiple cells and then click the Include Values or Exclude Values button to change the visibility of all selected cells at once.
  - Use SHIFT + click to select a range of adjacent cells. Use CTRL + click to select multiple cells that are not adjacent.
  - Use the Preview button to test the results.
- Legend:**
  - Excluded Value (Grey)
  - Included Value (Yellow)
- Table Data:**

College Name	MIT-BLR	MIT-HYD	MIT-MUM	MIT-MAA	MIT-CCU	MIT-DEL
Email:	b1@mit.co.in	b2@mit.co.in	b3@mit.co.in	b4@mit.co.in	b5@mit.co.in	b6@mit.co.in
	hyd@mit.co.in	hyd@mit.co.in	hyd@mit.co.in	hyd@mit.co.in	hyd@mit.co.in	hyd@mit.co.in
	mum@mit.co.in	mum@mit.co.in	mum@mit.co.in	mum@mit.co.in	mum@mit.co.in	mum@mit.co.in
	maa@mit.co.in	maa@mit.co.in	maa@mit.co.in	maa@mit.co.in	maa@mit.co.in	maa@mit.co.in
	ccu@mit.co.in	ccu@mit.co.in	ccu@mit.co.in	ccu@mit.co.in	ccu@mit.co.in	ccu@mit.co.in
	del@mit.co.in	del@mit.co.in	del@mit.co.in	del@mit.co.in	del@mit.co.in	del@mit.co.in

**Screenshot 2 (Controlling Field: Capacity Of Students, Dependent Field: Capacity Of Students):**

- Controlling Field:** Capacity Of Students
- Dependent Field:** Capacity Of Students
- Instructions:**
  - Double click on a cell to toggle its visibility for the Controlling Field value shown in the column heading.
  - To change multiple cells at once, select multiple cells and then click the Include Values or Exclude Values button to change the visibility of all selected cells at once.
  - Use SHIFT + click to select a range of adjacent cells. Use CTRL + click to select multiple cells that are not adjacent.
  - Use the Preview button to test the results.
- Legend:**
  - Excluded Value (Grey)
  - Included Value (Yellow)
- Table Data:**

College Name	MIT-BLR	MIT-HYD	MIT-MUM	MIT-MAA	MIT-CCU	MIT-DEL
Capacity Of Students:	500-1000	500-1000	500-1000	500-1000	500-1000	500-1000
	1000-2500	1000-2500	1000-2500	1000-2500	1000-2500	1000-2500
	2500-4000	2500-4000	2500-4000	2500-4000	2500-4000	2500-4000
	6000-10000	6000-10000	6000-10000	6000-10000	6000-10000	6000-10000

### ➤ Creating Validation Rules

1. 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`.

## Project Title : College Management Application

The screenshot shows the Salesforce Setup interface for the 'College' object. On the left, a sidebar lists various configuration options: Details, Fields & Relationships, 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 Validation Rule' and displays the 'Validation Rule Detail' for a rule named 'College\_Name\_Similar'. The rule is active and has the formula `TEXT(College__Name__c) <> Name`. The error message is 'Please Use The College Name in Record info'. The 'Error Location' is set to 'Top of Page'. The 'Created By' field shows 'Durga Prasad Avula' with a timestamp of '1/5/2022, 6:33 AM', and the 'Modified By' field also shows 'Durga Prasad Avula' with the same timestamp. There are 'Edit' and 'Clone' buttons at the top right of the detail section.

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

AND( Ready\_To\_Join\_\_c == true,  
OR(  
ISCHANGED( Address\_\_c ),  
ISCHANGED( College\_\_c ),  
ISCHANGED( Date\_Of\_Birth\_\_c ),  
ISCHANGED( Email\_\_c ),  
ISCHANGED( Guardian\_Name\_\_c ),  
ISCHANGED( Phone\_\_c )  
)  
)

## Project Title : College Management Application

The screenshot shows the Salesforce Setup interface for the 'Application Form' object. On the left, a sidebar lists various setup categories like Details, Fields & Relationships, Page Layouts, etc. The main content area is titled 'Application Form Validation Rule' and displays the configuration for a specific rule named 'ReadyToJoin'. The rule's formula is: AND( Ready\_To\_Join\_\_c == true, OR( ISCHANGED(Address\_\_c), ISCHANGED(College\_\_c), ISCHANGED(Date\_of\_Birth\_\_c), ISCHANGED(Email\_\_c), ISCHANGED(Phone\_\_c) ) ). It has an error message 'False information' and is active. The 'Created By' field shows 'Durga.Parsad' with a timestamp of 1/15/2022, 9:29 PM, and the 'Modified By' field also shows 'Durga.Parsad' with the same timestamp.

## Day 4 :

### Topic : Adding Business Logic To Application

#### Milestone / Activities:

- a) Process Automation
- b) Create The Student Record Using Flow

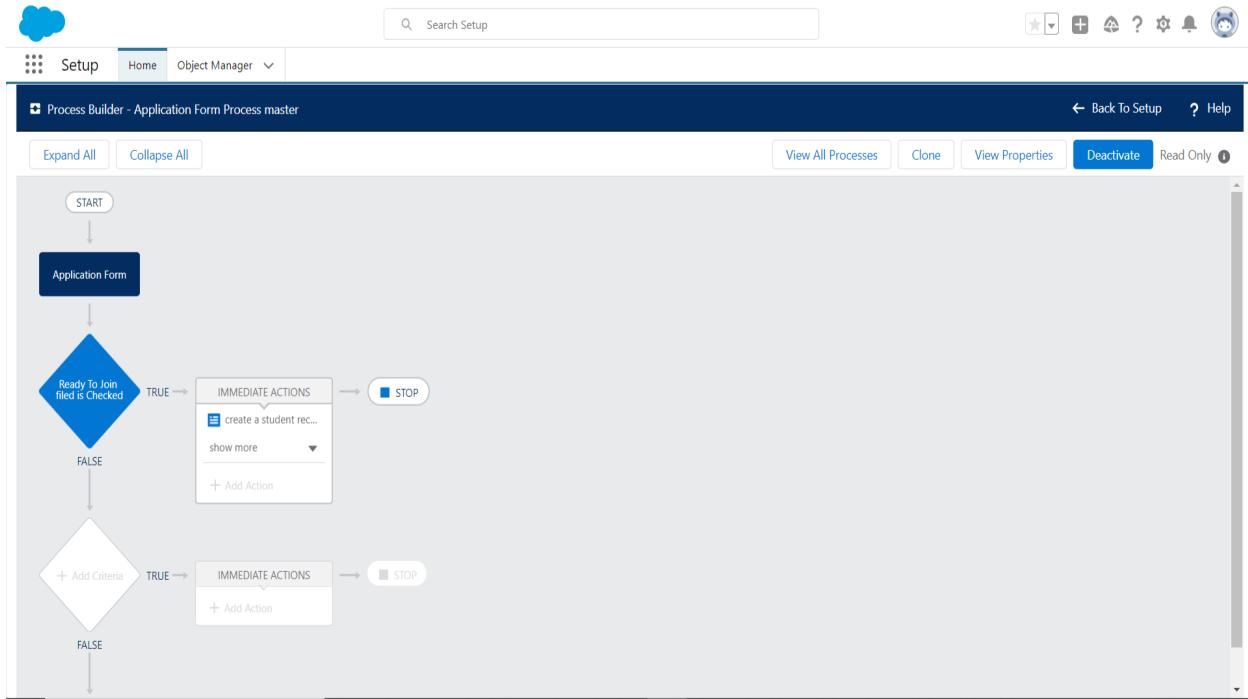
#### Detailed Description:

##### ► Process Automation

1. 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".
  - a. 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."
  - b. Once the node is setup, click on the adjacent box called "Immediate action".

## Project Title : College Management Application

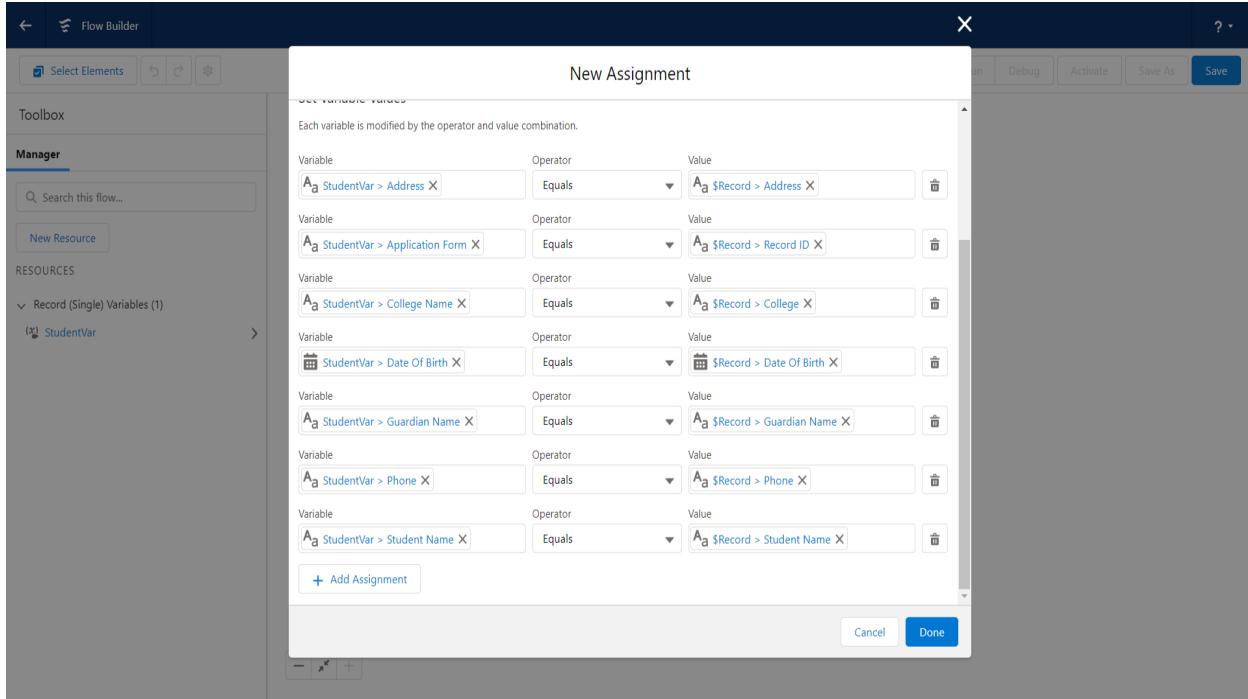
And select create a record on the student object. Please follow the below screenshot.



### ► Create The Student Record Using Flow

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.
5. Now add the assignment as follows.

## Project Title : College Management Application



## Day 5

**Topic:** Create a batch apex form

**Milestone \ Activity:** Created

**Description:**

1.From the Developer console created a new apex class and enter the following code

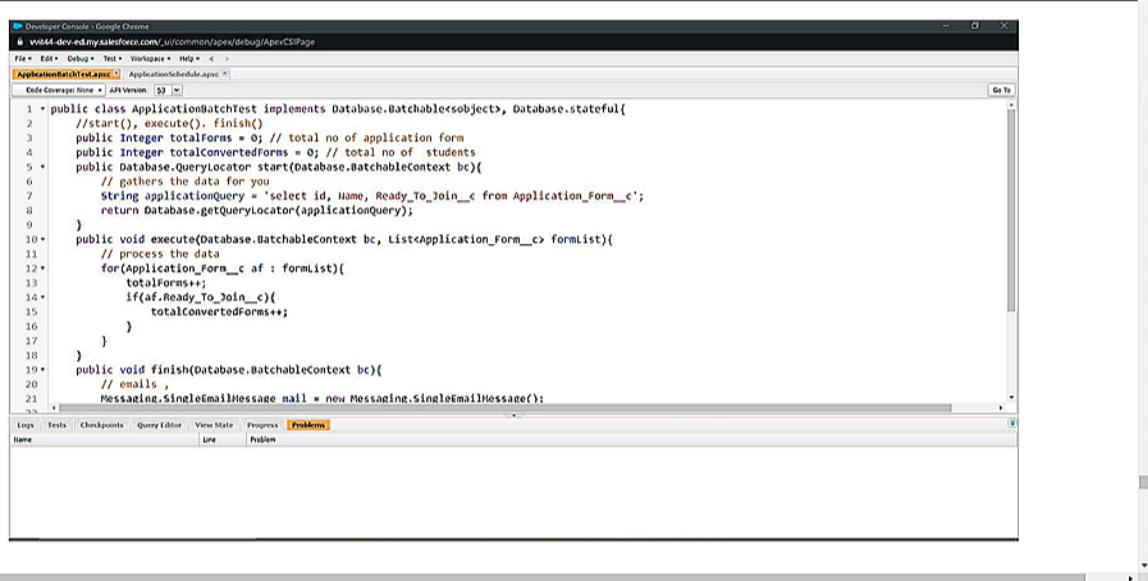
Syntax Of BatchApex

```
Public class ApplicationBatchTest implements Database.Batchable<sObject>{
```

```
//start(), execute(). finish()
public Integer totalForms = 0; // total no of application form
public Integer totalConvertedForms = 0; // total no of students
public Database.QueryLocator start(Database.BatchableContext bc){
    // gathers the data for you
    String applicationQuery = 'select id, Name, Ready_To_Join__c from
ApplicationForm__c';
    return Database.getQueryLocator(applicationQuery);
}
```

## Project Title : College Management Application

```
public void execute(Database.BatchableContext bc, List<ApplicationForm__c>
formList){      // process the data
    for(ApplicationForm__c af : formList){
        totalForms++;
        if(af.Ready_To_Join__c){
            totalConvertedForms++;
        }
    }
}
public void finish(Database.BatchableContext bc){
    // emails ,
    Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage();
    // address, subject, content( data to sent to admins)
    mail.setSubject(' Application form and student record data as of today ');
    mail.setPlainTextBody(' Total no of application form records are : '+totalForms+
    ' out of which no of students as per today : '+totalConvertedForms);
    String[] emailAddess = new String[
    {'harshavardhanakaredla2001@gmail.com'}];
    mail.setToAddresses(emailAddess);
    Messaging.sendEmail(new Messaging.SingleEmailMessage[]{ mail } );
}
}
```



The screenshot shows the Salesforce Developer Console interface. The top bar includes tabs for 'File', 'Edit', 'Debug', 'Test', 'Workspace', 'Help', and 'Go To'. The main area displays the Apex code for 'ApplicationBatchTest'. The code implements the Database.Batchable interface, performing a query to gather data and then processing it to count converted forms. It also sends an email to a specific address. The bottom navigation bar features tabs for 'Logs', 'Tests', 'Checkpoints', 'Query Editor', 'View State', 'Progress', and 'Problems', with 'Problems' currently selected.

```
1 * public class ApplicationBatchTest implements Database.Batchable<sobject>, Database.stateful{
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(Application_Form__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        Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage();
22    }
}
```

**Topic:** Create a schedular class

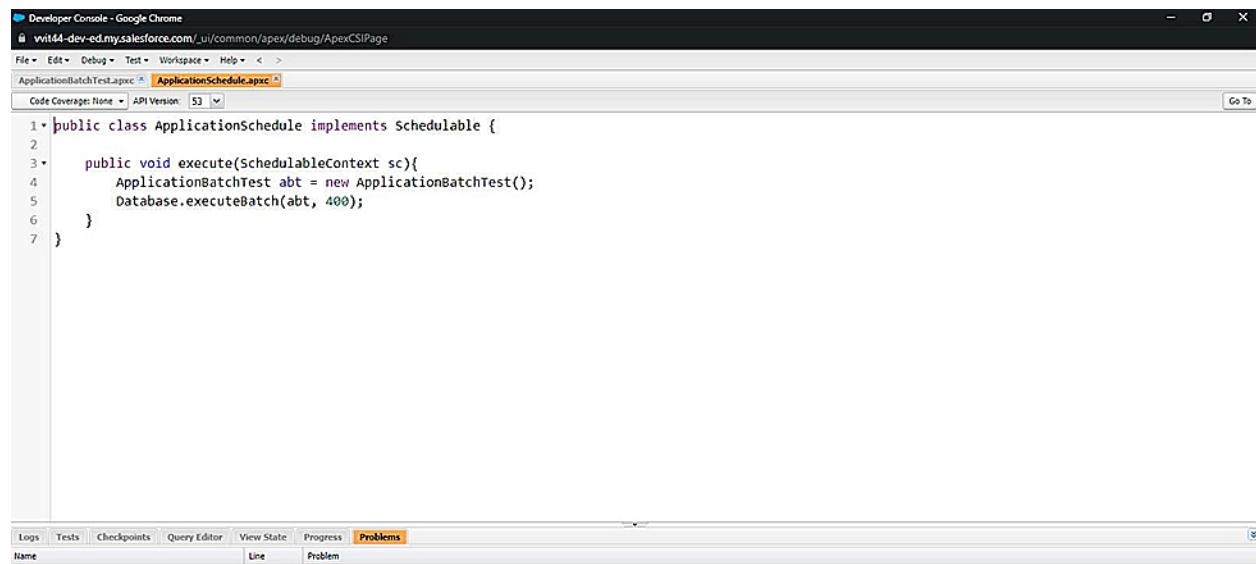
**Milestone \ Activity:** Created

**Description:**

1. From the developer console create a new apex class and enter the following code.

```
public class application schedule implements Schedulable{  
  
    public void execute(SchedulableContext sc){  
        ApplicationBatchTest abt = new ApplicationBatchTest();  
        Database.executeBatch(abt, 400); // 200 to 2000  
  
    }  
}
```

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



The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is [vrit44-dev-ed.my.salesforce.com/\\_ui/common/apex/debug/ApexCSIPage](http://vrit44-dev-ed.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage). The tab bar shows "ApplicationBatchTest.apxc" and "ApplicationSchedule.apxc". The code editor displays the following Apex code:

```
1 *public class ApplicationSchedule implements Schedulable {  
2  
3     public void execute(SchedulableContext sc){  
4         ApplicationBatchTest abt = new ApplicationBatchTest();  
5         Database.executeBatch(abt, 400);  
6     }  
7 }
```

The "Problems" tab in the bottom navigation bar is selected. There are no problems listed.

**Day 6 :** Lightening Web Component

**Topic :** Create College Data Table Component

**Milestone \ Activity :** Created

**Description :**

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 .

Create an Apex Class With The Required Functionality

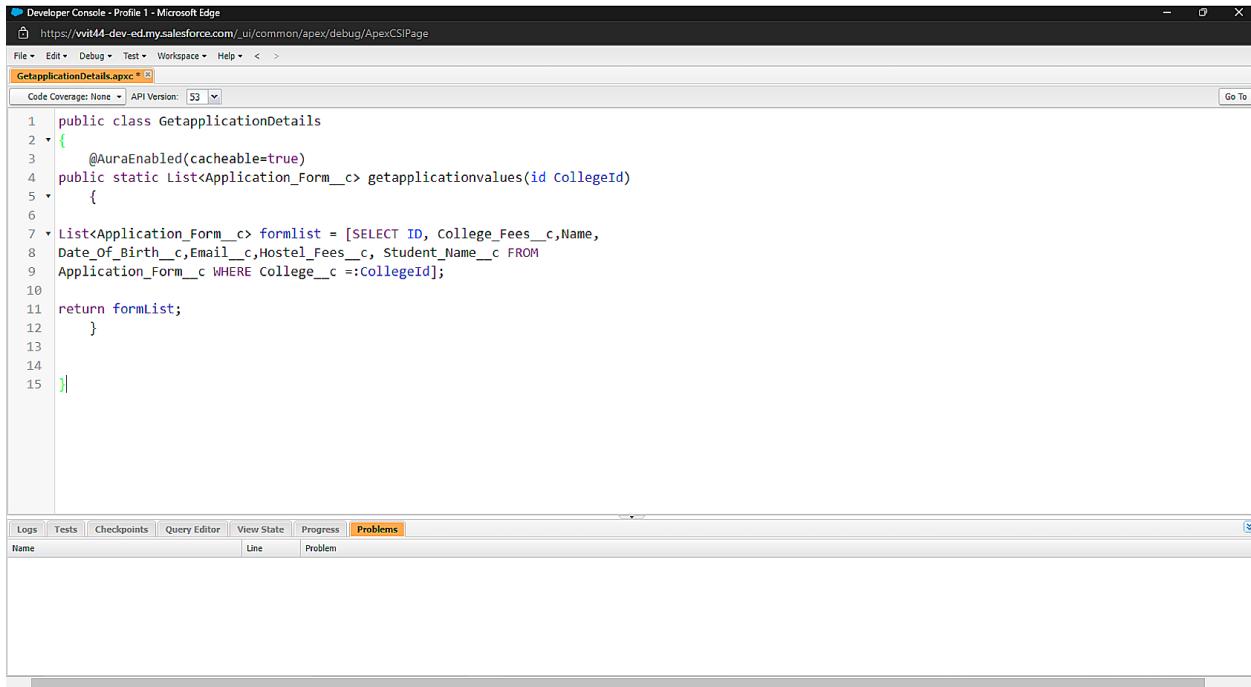
```
public class GetapplicationDetails
{
    @AuraEnabled(cacheable=true)
    public static List<ApplicationForm__c> getapplicationvalues(id CollegelId)
    {

        List<ApplicationForm__c> formlist = [SELECT ID, College_Fees__c, Name,
        Date_of_Birth__c, Email__c, Hostel_Fees__c, Student_Name__c FROM
        ApplicationForm__c WHERE College__c =:CollegelId];

        return formList;
    }

}
```

## Project Title : College Management Application



The screenshot shows the Salesforce Developer Console in Microsoft Edge. The URL is https://vit44-dev-ed.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage. The tab title is "GetapplicationDetails.apex". The code editor contains 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
15 }
```

The bottom navigation bar shows tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Problems tab is selected.

**Topic :** Create College DataTable Component (HTML FILE)

**Milestone \ Activity :** Created

**Description :**

HTML:

```
<template>
```

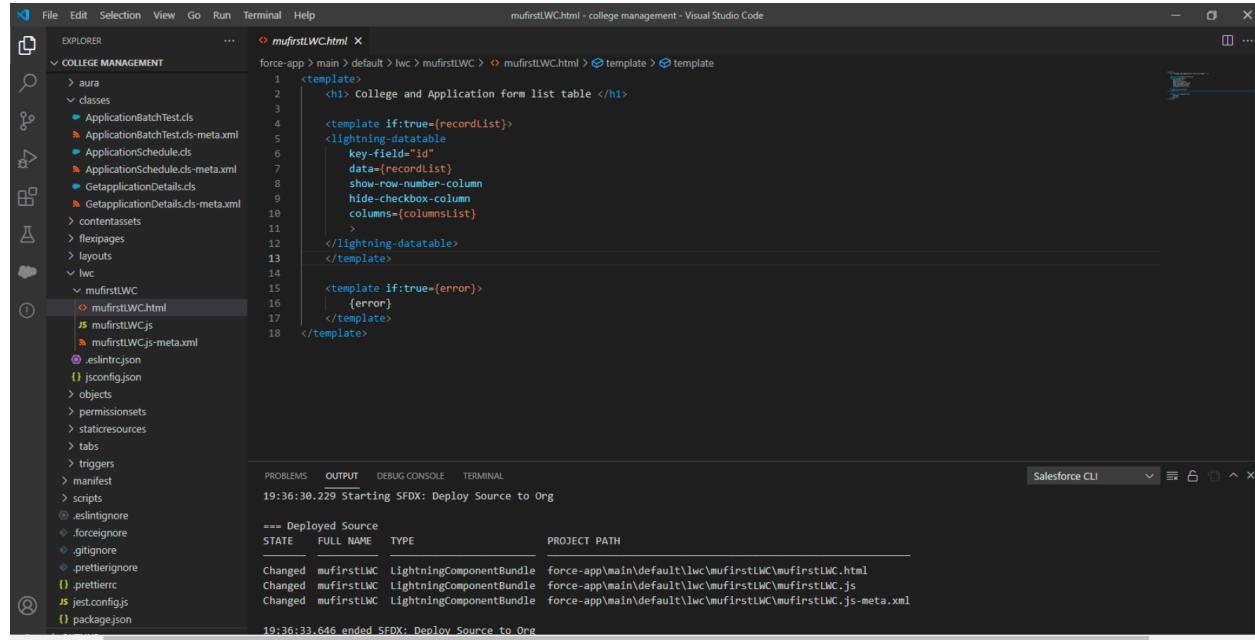
```
    <h1> College and Application form list table </h1>
```

```
    <template if:true={recordList}>
        <lightning-datable
            key-field="id"
            data={recordList}
            show-row-number-column
            hide-checkbox-column
            columns={columnsList}
        >
    </lightning-datable>
</template>
```

```
<template if:true={error}>
```

## Project Title : College Management Application

```
{error}
</template>
</template>
```



```
<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>
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

19:36:30.229 Starting SFDX: Deploy Source to Org

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

19:36:33.646 ended SFDX: Deploy Source to Org

**Topic :**Create College DataTable Component(JAVA SCRIPT FILE)

**Milestone \ Activity :** Created

**Description :**

```
import { LightningElement, api, wire } from 'lwc';
import getapplicationvalues
from '@salesforce/apex/GetapplicationDetails.getapplicationvalues';
export default class CollegeDataTable extends LightningElement {
    columnsList = [
        {label : 'Application Form' , fieldName : 'Name', type:'text' },
        {label : 'College Fees' , fieldName : 'College_Fees__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;
```

## Project Title : College Management Application

error;

```
@wire(getapplicationvalues, {CollegeId : '$recordId'})  
wiredCollegeData({data, error}){  
    if(data){  
        this.recordList = data;  
    }  
    else if(error){  
        this.error = error;  
        this.recordList = undefined;  
    }  
}  
}
```

```
File Edit Selection View Go Run Terminal Help mufirstLWC.js - college management - Visual Studio Code  
EXPLORER COLLEGE MANAGEMENT  
> aura  
> classes  
  ▷ ApplicationBatchTest.cls  
  ▷ ApplicationBatchTest.cls-meta.xml  
  ▷ ApplicationSchedule.cls  
  ▷ ApplicationSchedule.cls-meta.xml  
  ▷ GetapplicationDetails.cls  
  ▷ GetapplicationDetails.cls-meta.xml  
> contentassets  
> flexpages  
> layouts  
  ▷ lwc  
    ▷ mufirstLWC  
      ▷ mufirstLWC.html  
      ▷ mufirstLWC.js  
      ▷ mufirstLWC.js-meta.xml  
    ▷ .eslintrc.json  
    ▷ jconfig.json  
  > objects  
  > permissionsets  
  > staticresources  
  > tabs  
  > triggers  
  > manifest  
  > scripts  
  ▷ .eslinignore  
  ▷ .forceignore  
  ▷ .gitignore  
  ▷ .prettierignore  
  ▷ .prettierrc  
  ▷ jest.config.js  
  ▷ package.json  
> OUTLINE  
> RUNNING TASKS  
File Edit Selection View Go Run Terminal Help mufirstLWC.js - college management - Visual Studio Code  
COLLEGE MANAGEMENT  
force-app > main > default > lwc > mufirstLWC > mufirstLWC.js > CollegeDataTable > recordId  
1 import { LightningElement, api, wire } from 'lwc';  
2 import getapplicationvalues  
3 from '@salesforce/apex/GetapplicationDetails.getapplicationvalues';  
4 export default class CollegeDataTable extends LightningElement {  
5     columsList = [  
6         {label : 'Application Form', fieldName : 'Name', type:'text' },  
7         {label : 'College Fees', fieldName : 'College_Fees__c', type:'currency' },  
8         {label : 'Date Of Birth', fieldName : 'DateOfBirth__c', type:'date' },  
9         {label : 'Email', fieldName : 'Email__c', type:'email' },  
10        {label : 'Hostel Fees', fieldName : 'Hostel_Fees__c', type:'Currency' },  
11        {label : 'Student Name', 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            this.recordList = undefined;  
26        }  
27    }  
28  
29    PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL  
19:30:33.046 ended SFDX: Deploy Source to Org  
19:37:12.27 Starting SFDX: Deploy Source to Org  
==== Deployed Source  
STATE FULL NAME TYPE PROJECT PATH  
Changed mufirstLWC LightningComponentBundle force-app/main/default/lwc/mufirstLWC/mufirstLWC.html  
Changed mufirstLWC LightningComponentBundle force-app/main/default/lwc/mufirstLWC/mufirstLWC.js  
Changed mufirstLWC LightningComponentBundle force-app/main/default/lwc/mufirstLWC/mufirstLWC.js-meta.xml  
19:37:15.415 ended SFDX: Deploy Source to Org  
Salesforce CLI  
In 14, Col 20 Spaces: 4 UTF-8 LF () JavaScript □ Prettier 
```

**Topic:**Create College DataTable Component(META FILE)

**Milestone \ Activity :** Created

**Description :**

```
<?xml version="1.0"?>  
<LightningComponentBundle  
xmlns="http://soap.sforce.com/2006/04/metadata">  
    <apiVersion>51.0</apiVersion>
```

## Project Title : College Management Application

```
<isExposed>true</isExposed>
<targets>
    <target>lightning__RecordPage</target>
    <target>lightning__AppPage</target>
    <target>lightning__HomePage</target>
</targets>
</LightningComponentBundle>
```

```
<?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>
```

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

19:37:55.432 Starting SFDX: Deploy Source to Org  
19:37:58.326 ended SFDX: Deploy Source to Org