

## PROJECT TITLE : COLLEGE MANAGEMENT APPLICATION

DAY 1:

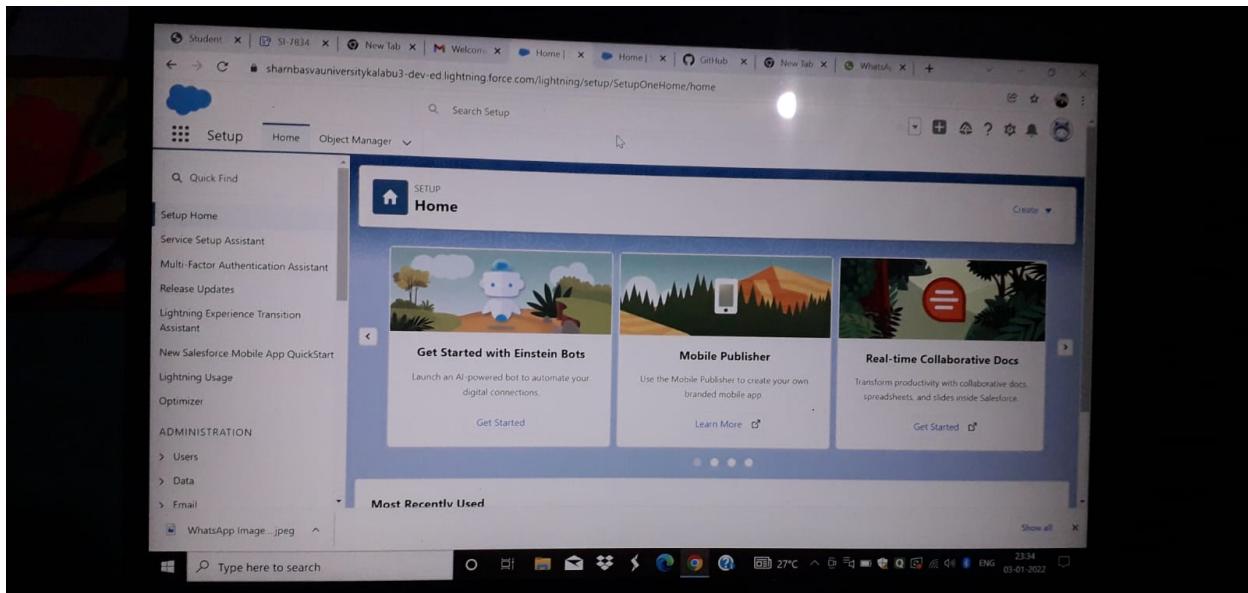
TOPIC : COLLEGE MANAGEMENT APPLICATION

CREATE YOUR SALESFORCE DEVELOPER ORG TO GET STARTED

- MILESTONE/ACTIVITIES :
- 1)Creating Developer Account. [ [developers.salesforce.com/](https://developers.salesforce.com/) ]
  - 2)Account Activation.
  - 3>Login To Your Salesforce Trailhead Account.  
[ <https://login.salesforce.com/> ]

DETAILED DESCRIPTION : In the first step I have created the salesforce developer account by following the given instruction such as , { FIRST NAME LAST NAME, E-MAIL 'ID and USER NAME} . After this I have signed up ,and I have received the mail , for verifying my developer account. This email process activates my salesforce developer account . so I have created new password for my salesforce developer a account. I have logged into my salesforce developer account my username and password.

SCREENSHOT : After successful logging into developer account I got this home page.



DAY : 2

TOPIC : COLLEGE MANAGEMENT APPLICATION

### CUSTOM OBJECT CREATION

MILESTONE/ACTIVITIES: 1)Custom Object Creation

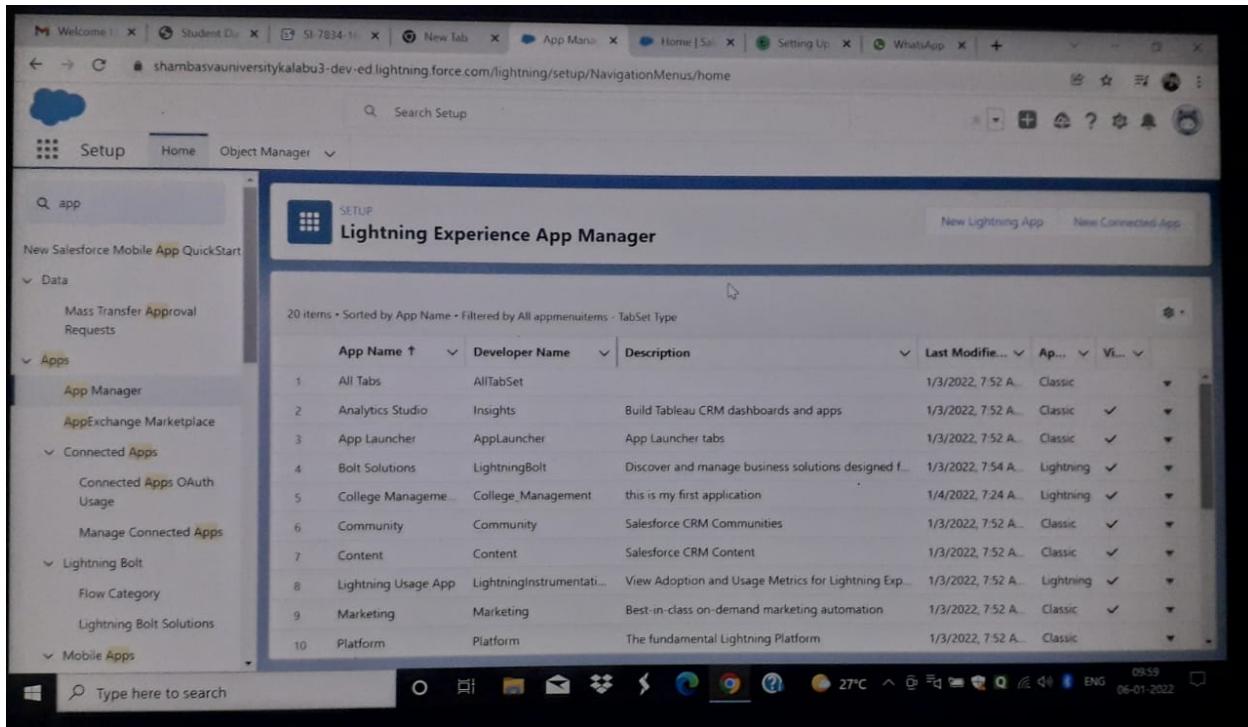
- 2)Create The Custom Objects For The Application
- 3)Create Fields On College Object
- 4)Create Fields On Application Form Object
- 5)Create Fields On Student Object
- 6)Create Fields On Subject Object

College Management Application

DETAILED DESCRIPTION :

- 1= First we have to login to our salesforce account, then we have to click on the setup gear icon. The home page will appear.
- 2 = We have to search for app manager.
- 3 = Now we have to click on new lightning app.
- 4 = Now we have create the College Management Application App.
- 5 = Enter the details as per the guidelines

Screenshot : After creating the app we can see the name in the app manager.



**TOPIC 1 = Create Object Creation : Create the custom object for the application.**

Create the following custom object for applications.

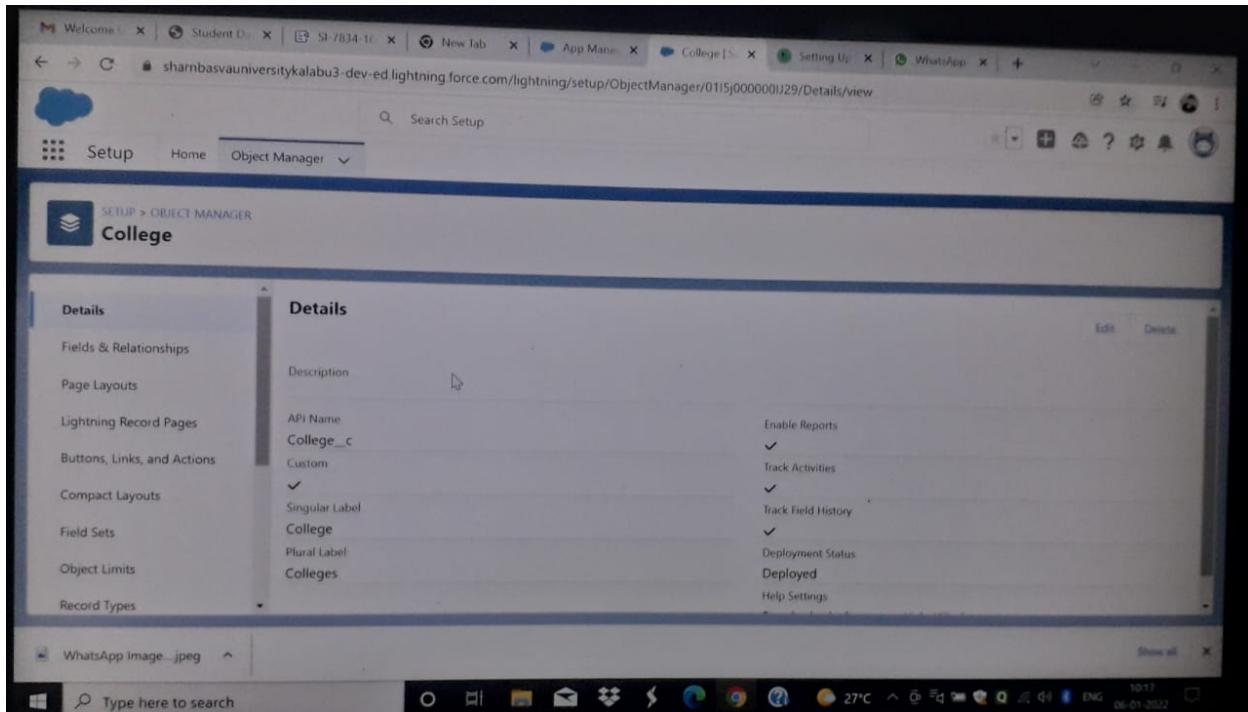
- 1)College
- 2)Application Form
- 3)Students
- 4)Subject

Now we have to create the custom object for College.

1 = Now we have to click on the object manager, then click on the new custom object creation.

2 = Enter the details as per the guidelines.

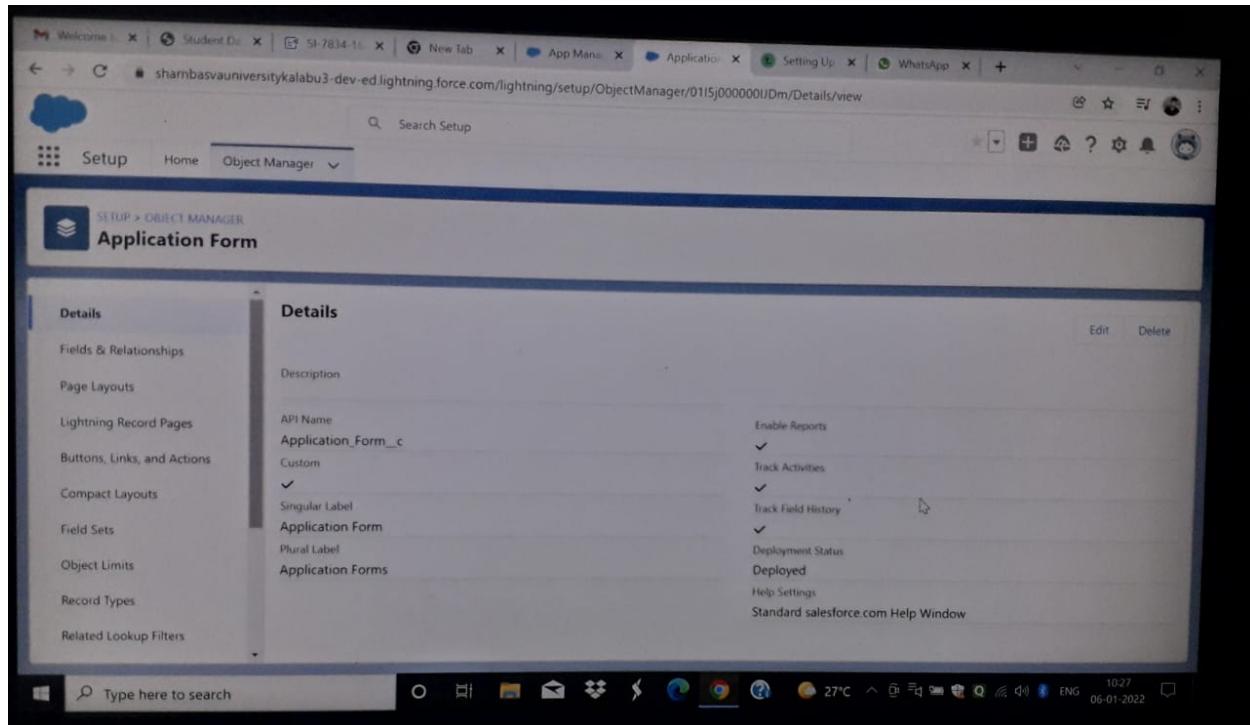
**Screenshot : After creating successfully we can see the College object in the object manager .**



Now we have to create the custom object for Application Form

- 1 = Now we have to click on the object manager, then click on the new custom object creation.
- 2 = Enter the details as per the guidelines.

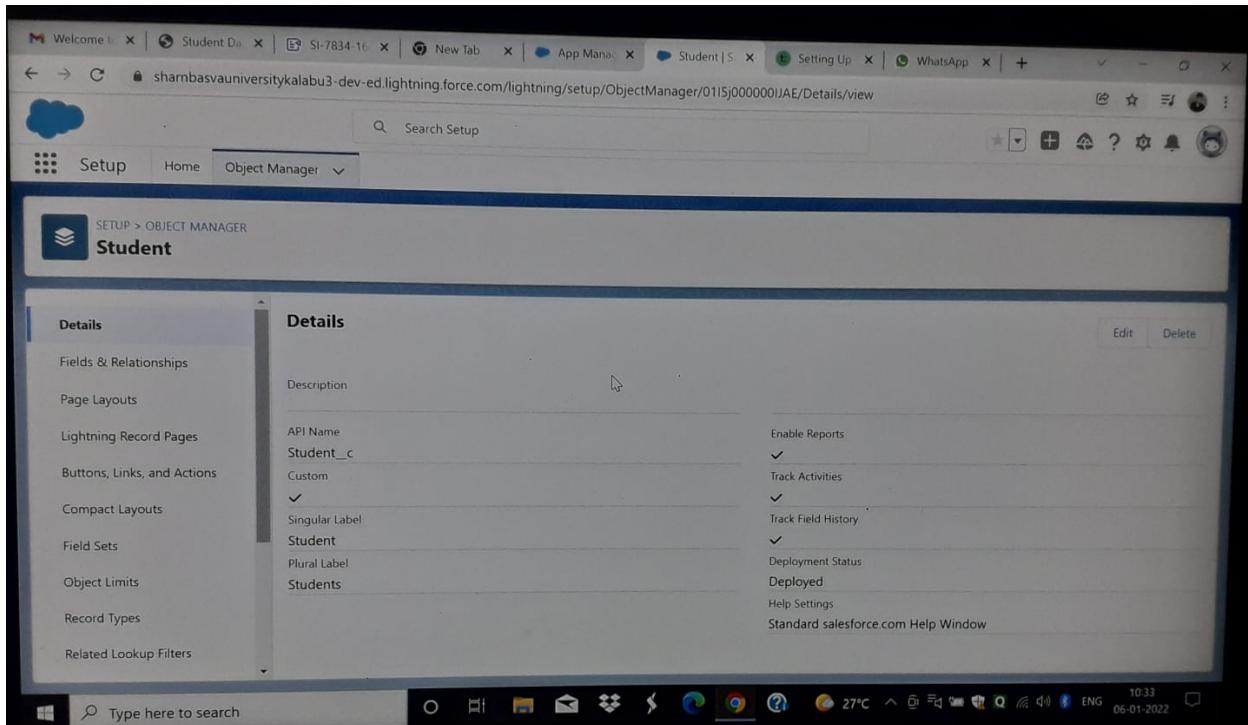
Screenshot : After creating successfully we can see the Application form object in the object manager.



Now we have to create the custom object for Students

- 1 = Now we have to click on the object manager, then click on the new custom object creation.
- 2 = Enter the details as per the guidelines.

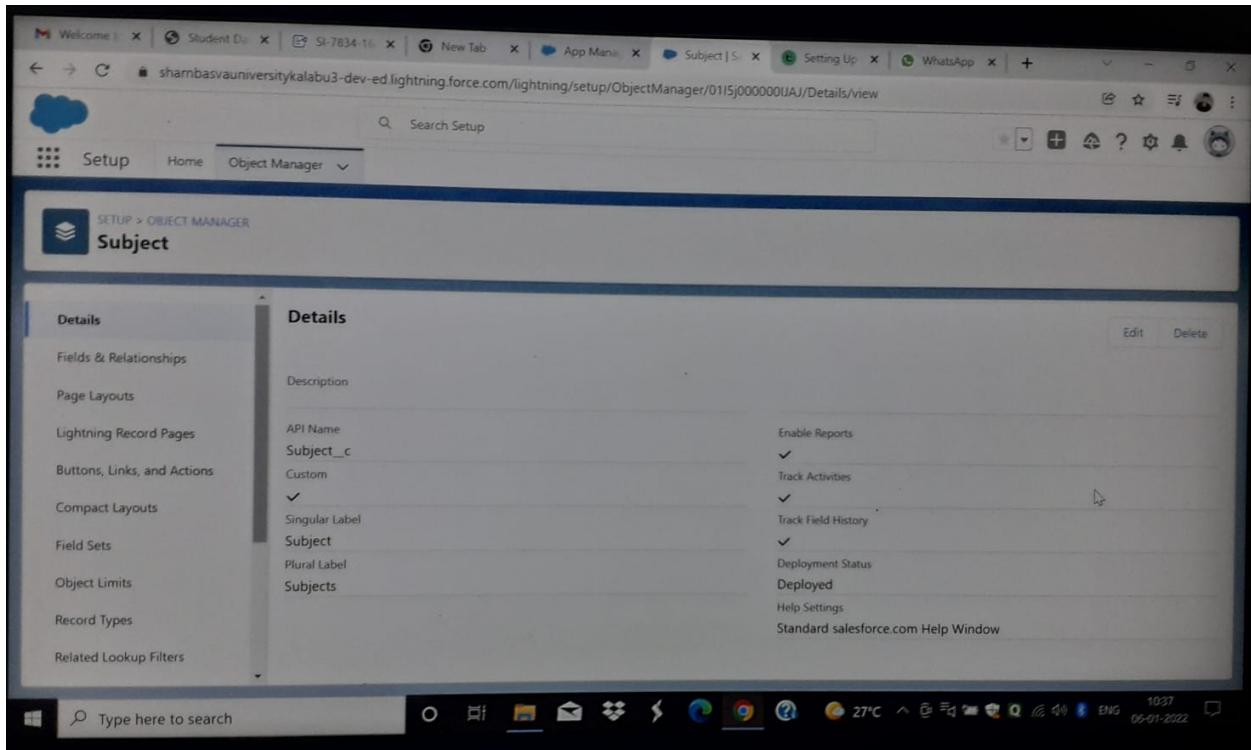
Screenshot : After creating successfully we can see the Student object in the object manager.



Now we have to create the custom object for Subjects

- 1 = Now we have to click on the object manager, then click on the new custom object creation.
- 2 = Enter the details as per the guidelines.

Screenshot : After creating successfully we can see the subject object in the object manager .



## TOPIC 2 : Create Fields On College Object

Now we have to create the field and relationship for the College name .

- 1 = Now we have to click on the college object , then we have click on the fields and relationships.
- 2 = Now we have to click on new option to create field name as college name.
- 3 = Enter the details as per the guidelines.
- 4 = After creating successfully we can see the field name in the field and relationship in the college object

Now we have to create the field and relationship for the College Email

- 1 = Now we have to click on the college object , then we have click on the fields and relationships.
- 2 = Now we have to click on new option to create field name as college Email.
- 3 = Enter the details as per the guidelines.
- 4 = After creating successfully we can see the field name in the field and relationship in the college object

Now we have to create the filed and relationship for the College fees

- 1 = Now we have to click on the college object , then we have click on the fields and relationships.
- 2 = Now we have to click on new option to create field name as college fees.
- 3 = Enter the details as per the guidelines.
- 4 = After creating successfully we can see the field name in the filed and relationship in the college object

Now we have to create the filed and relationship for the Hostel fees

- 1 = Now we have to click on the college object , then we have click on the fields and relationships.
- 2 = Now we have to click on new option to create field name as hostel fees.
- 3 = Enter the details as per the guidelines.
- 4 = After creating successfully we can see the field name in the filed and relationship in the college object

Now we have to create the filed and relationship for the number of seats

- 1 = Now we have to click on the college object , then we have click on the fields and relationships.
- 2 = Now we have to click on new option to create field name as number of seats.
- 3 = Enter the details as per the guidelines.
- 4 = After creating successfully we can see the field name in the filed and relationship in the college object

Now we have to create the filed and relationship for the record info

- 1 = Now we have to click on the college object , then we have click on the fields and relationships.
- 2 = Now we have to click on new option to create field name as record info.
- 3 = Enter the details as per the guidelines.
- 4 = After creating successfully we can see the field name in the filed and relationship in the college object

Screenshot : After creating all the fields and relationships in the College object we can see the

output.

The screenshot shows the Salesforce Object Manager Fields & Relationships page for the 'College' object. The page lists nine fields, each with its field label, name, data type, controlling field, and indexed status. The fields 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	

Screen shot : output

The screenshot shows the Salesforce Setup interface with the URL [sharmbasvauniversitykalabu3-dev-ed.lightning.force.com/lightning/setup/ObjectManager/01ISj000000J29/FieldsAndRelationships/view](https://sharmbasvauniversitykalabu3-dev-ed.lightning.force.com/lightning/setup/ObjectManager/01ISj000000J29/FieldsAndRelationships/view). The page title is "College". The main content area is titled "Fields & Relationships" and lists 9 items, sorted by Field Label. The columns show the field name, its internal name, and its type. The fields listed are:

Field	Internal Name	Type
College Fee	College_fee__c	Currency(10, 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
Owner	OwnerId	Lookup(User,Group)
Record Info	Name	Text(80)

### DAY 3:

#### TOPIC : COLLEGE MANAGEMENT APPLICATION

#### ADDING BUSINESS LOGIC TO APPLICATION

MILESTONE/ACTIVITIES: 1)Creating Global Value picklist Sets

- 2)Creating Field Dependencies
- 3)Creating Validation Rules
- 4)Process Automation
- 5)Create The Student Record Using Flow

Business logic is the brain of the application; it makes sense of the data and responds to button clicks in the UI.

A controller in the Model-View-Controller architecture forms the bridge between a model data and a display.

we can write the business logic in apex code as well as in declarative syntax.

## Creating Global Picklist Value Sets.

1=We have to create the global picklist values for the application

a) Picklist value name = college

Values = MIT-HYD

MIT-BLR

MIT-MUM

MIT-MAA

MIT-DEL

MIT-CCU

1=First we have to login to our salesforce developer account.

2=Then we have to click on set up icon, the home page we appear.

3=Now we have to search for picklist value sets in the search bar,then enter it.

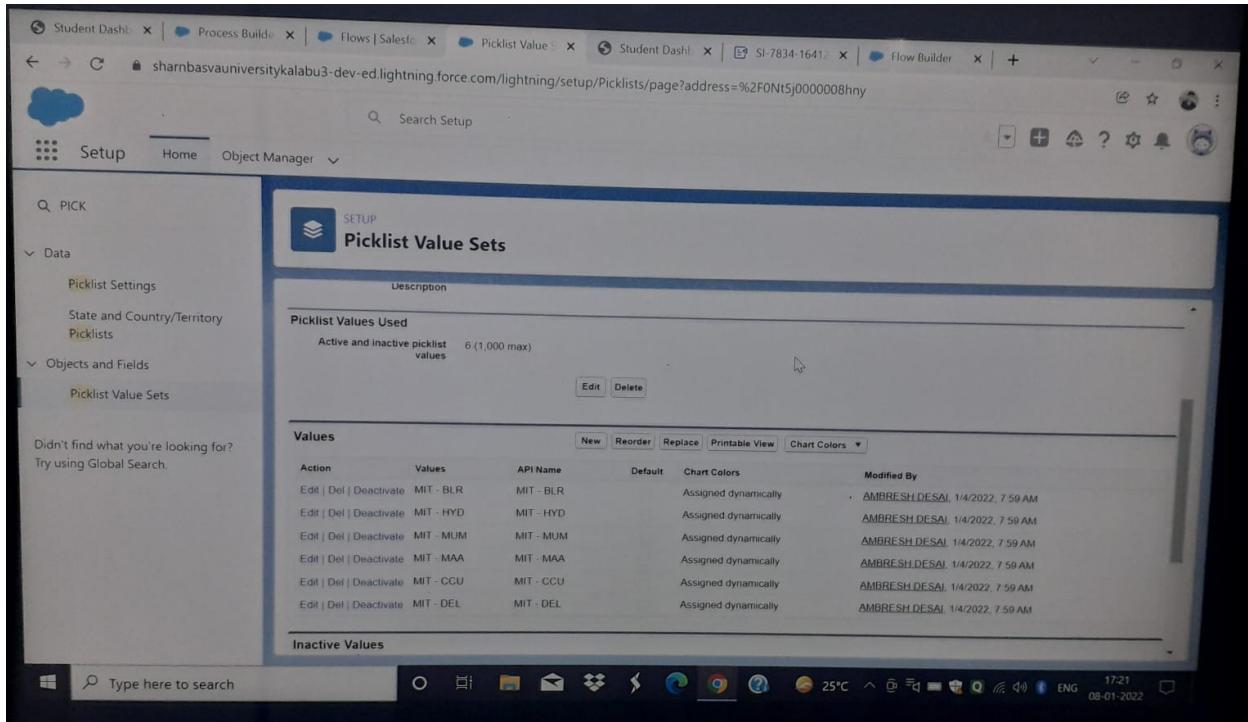
4=Now we have to click on the new option to create the global picklist values for college .

5=Enter the filed name as college name, then move to next step.

6=Now enter the global picklist values for the college .

#### SCREENSHOOT : OUTPUT

## SCREENSHOOT : OUTPUT



7=Then save it.

We have to create the global picklist values for the application

b) Picklist value name = Paper 1

Values = APEX

JAVA

C

C++

1=Now we have to click on set up icon, the home page we appear.

2=Now we have to search for picklist value sets in the search bar,then enter it.

3=Now we have to click on the new option to create the global picklist values for Paper 1 .

4=Enter the filed name as college name, then move to next step.

5=Now enter the global picklist values for thePaper 1 .

SCREENSHOOT : OUTPUT

The screenshot shows the Salesforce Setup interface with the URL [https://sharnbasvauniversitykalabu3-dev-ed.lightning.force.com/lightning/setup/Picklists/page?address=%2F0Nt\\$0000008itN](https://sharnbasvauniversitykalabu3-dev-ed.lightning.force.com/lightning/setup/Picklists/page?address=%2F0Nt$0000008itN). The left sidebar is open with 'Data' selected, showing 'Picklist Settings' and 'State and Country/Territory Picklists'. Under 'Objects and Fields', 'Picklist Value Sets' is selected. The main content area is titled 'SETUP Picklist Value Sets'. It displays a table of 'Values' with columns: Action, Values, API Name, Default, Chart Colors, and Modified By. The table contains four rows: APEX (API Name: APEX), JAVA (API Name: JAVA), C (API Name: C), and C++ (API Name: C++). All values are marked as 'Assigned dynamically' and modified by 'AMBRESH DESAI' at 1/6/2022, 6:59 PM. Below the table, there is a section for 'Inactive Values' which states 'No Inactive Values values defined'. At the bottom, there is a 'Fields Where Used' section and a link to 'https://sharnbasvauniversitykalabu3-dev-ed.my.salesforce.com/one/one.app...'. The bottom right corner shows system status: 25°C, ENG, 08-01-2022, 17:43.

We have to create the global picklist values for the application

C) Picklist value name = Paper 2

Values = MATHEMATICS

ENGLISH

STATISTICS

1=Now we have to click on set up icon, the home page we appear.

2=Now we have to search for picklist value sets in the search bar,then enter it.

3=Now we have to click on the new option to create the global picklist values for Paper 2 .

4=Enter the filed name as Paper 2 name, then move to next step.

5=Now enter the global picklist values for the Paper 2.

SCREENSHOT : OUTPUT

The screenshot shows the Salesforce Setup interface with the URL [sharnbasvauniversitykalabu3-dev-ed.lightning.force.com/lightning/setup/Picklists/page?address=%2F0Nt5j0000008itS](https://sharnbasvauniversitykalabu3-dev-ed.lightning.force.com/lightning/setup/Picklists/page?address=%2F0Nt5j0000008itS). The left sidebar is expanded to show 'Data' and 'Objects and Fields'. Under 'Objects and Fields', 'Picklist Value Sets' is selected. The main content area is titled 'Picklist Value Sets' and displays a table of values. The table has columns: Action, Values, API Name, Default, Chart Colors, and Modified By. There are three rows: MATHEMATICS, ENGLISH, and STATISTICS, all modified by AMBRESH DESAI on 1/6/2022 at 7:00 PM. Below the table, sections for 'Inactive Values' (No records to display) and 'Fields Where Used' (No records to display) are shown. At the bottom right, there is a message: 'Always show me ▾ more records per related list'.

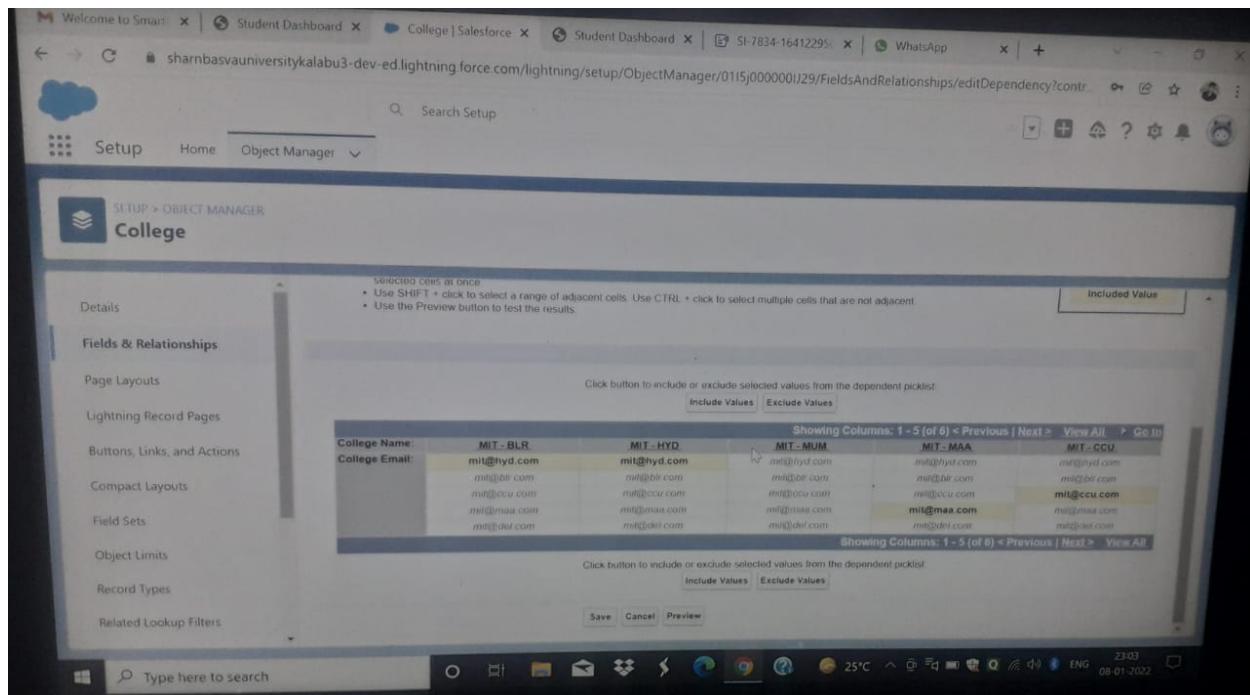
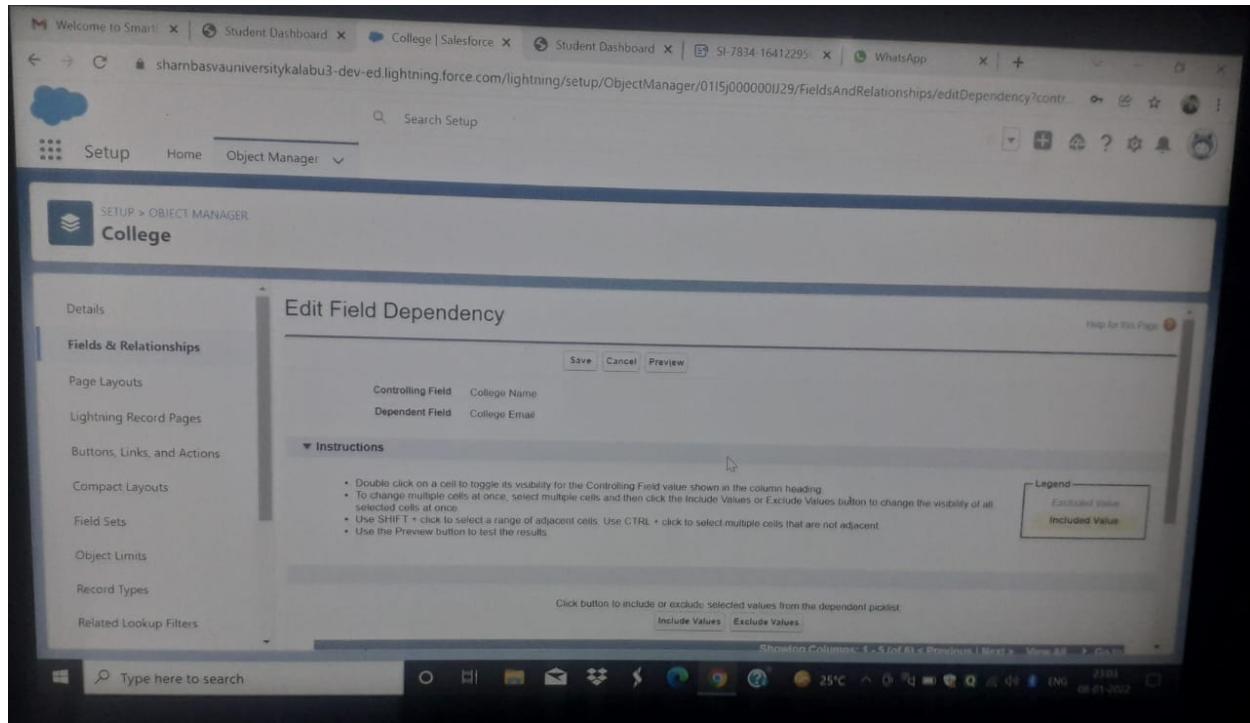
Action	Values	API Name	Default	Chart Colors	Modified By
Edit   Del   Deactivate	MATHEMATICS	MATHEMATICS	Assigned dynamically		AMBRESH DESAI, 1/6/2022, 7:00 PM
Edit   Del   Deactivate	ENGLISH	ENGLISH	Assigned dynamically		AMBRESH DESAI, 1/6/2022, 7:00 PM
Edit   Del   Deactivate	STATISTICS	STATISTICS	Assigned dynamically		AMBRESH DESAI, 1/6/2022, 7:00 PM

## Creating Field Dependencies

Now we have to create the filed dependencies,between college name and email where the controlling field is college name and dependent field is email, select the email id's according to college names.

- 1 = Now we have to click on the college object , then we have click on the field dependencies.
- 2 = Now we have to click on new option to create field name as college name and department field as college email.
- 3 = Enter the details as per the guidelines.
- 4 = After creating successfully we can see the field dependencies in the college filed dependencies.

SCREENSHOT : OUTPUT

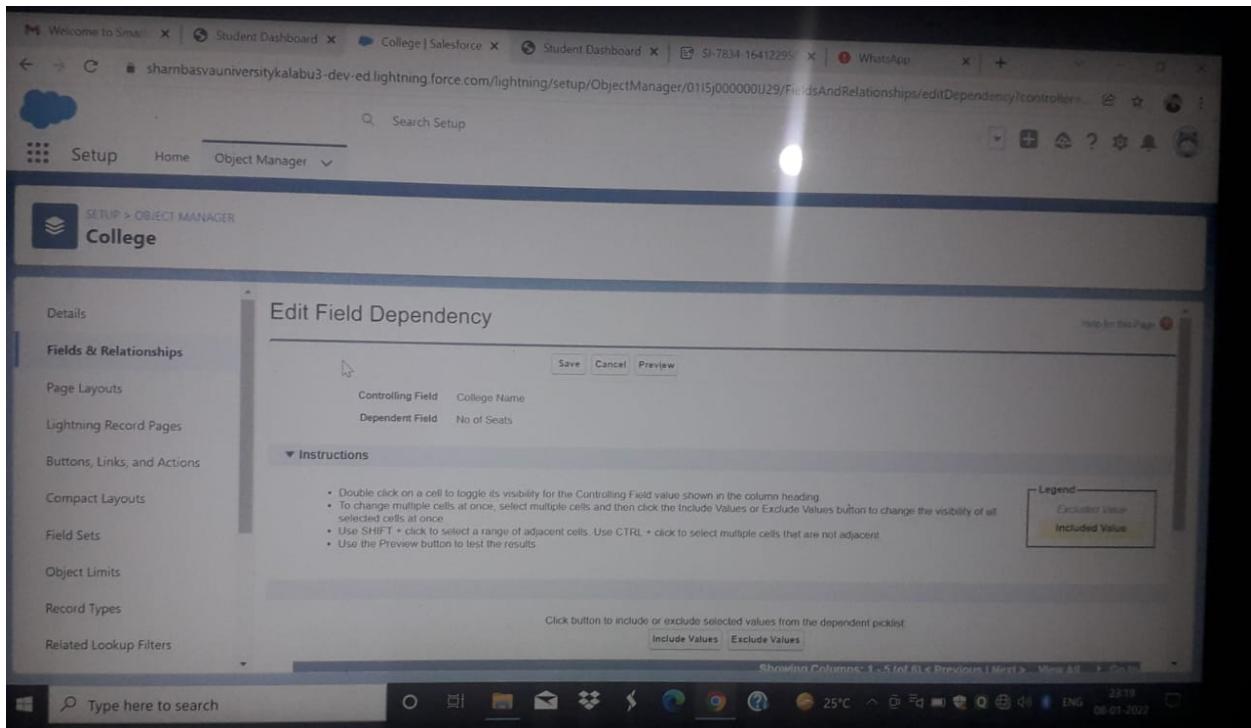


Now we have to create the field dependencies, between college name and capacity of students where the controlling field is college name and dependent field is capacity of students

select the values according to your wish.

- 1 = Now we have to click on the college object , then we have click on the field dependencies.
- 2 = Now we have to click on new option to create field name as college name and department field as no of seats.
- 3 = Enter the details as per the guidelines.
- 4 = After creating successfully we can see the field dependencies in the college filed dependencies.

#### SCREENSHOT : OUTPUT



The screenshot shows the Salesforce Object Manager interface for the 'College' object. On the left, a sidebar lists various setup options like Details, Fields & Relationships, Page Layouts, etc. The main area displays a grid of field dependencies. A tooltip at the top right of the grid provides instructions: 'Selected cells at once' and '• Use SHIFT + click to select a range of adjacent cells. Use CTRL + click to select multiple cells that are not adjacent.' Below the grid, there are buttons for 'Include Values' and 'Exclude Values'. At the bottom of the grid, there are 'Save', 'Cancel', and 'Preview' buttons.

The screenshot shows the 'College Field Dependencies' page. The sidebar on the left is identical to the previous screenshot. The main content area is titled 'College Field Dependencies' and includes a sub-header 'Back to Custom Object: College'. It states, 'This page allows you to define dependencies between fields (e.g., dependent picklists).'. Below this, a table titled 'Field Dependencies' lists two entries:

Action	Controlling Field	Dependent Field	Modified By
Edit   Del	College Name	College Email	AMBRESH DESAI, 1/4/2022, 8:57 AM
Edit   Del	College Name	No of Seats	AMBRESH DESAI, 1/4/2022, 9:02 AM

## Creating Validation Rules

create a validation rule on the college object such that college name and record info should

have same name.

TEXT(College\_Name\_\_c)<>Name.

1 = Now we have to click on the college object , then we have click on the validation rules.

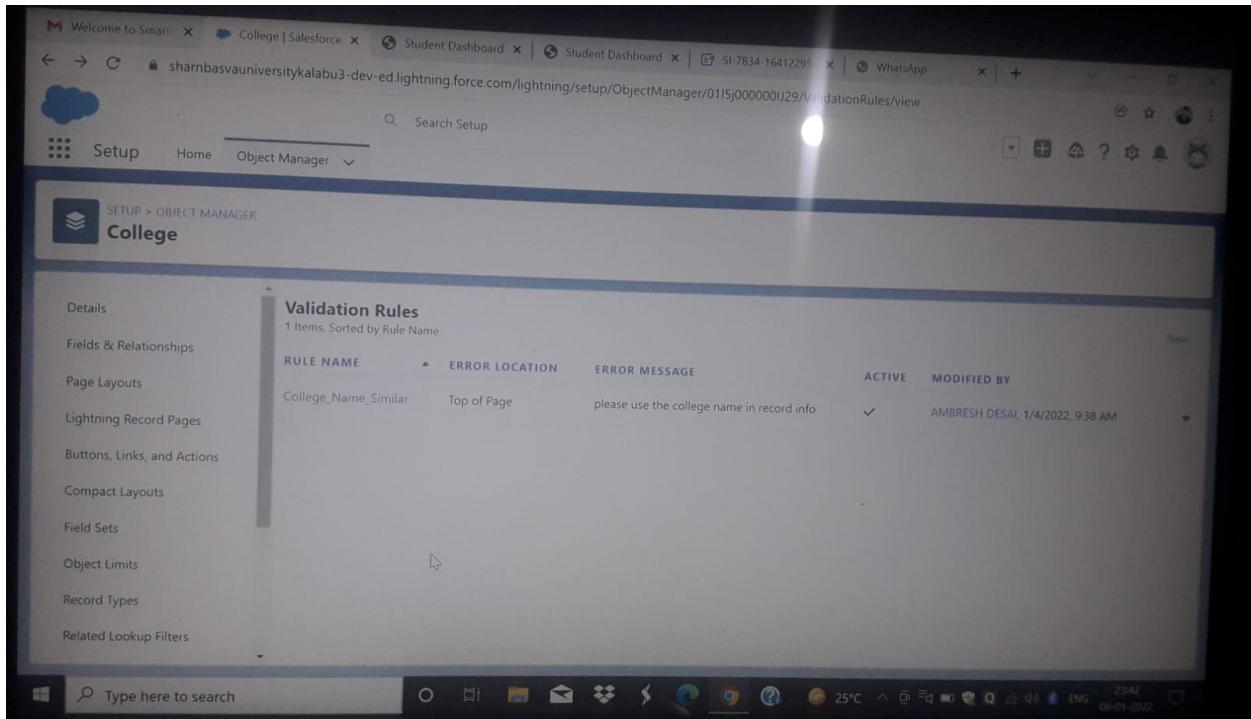
2 = Now we have to click on new option to create new validation name as college name and record info as name.

3 = Enter the details as per the guidelines.

4 = After creating successfully we can see the validation rule in the college validation file.

#### SCREENSHOT : OUTPUT

The screenshot shows the Salesforce Setup interface for the 'College' object. On the left, there's a sidebar with various tabs like Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, and Related Lookup Filters. The main content area is titled 'College Validation Rule' and shows the 'Validation Rule Detail' for a rule named 'College\_Name\_Similar'. The rule formula is 'TEXT(College\_Name\_\_c) <> Name'. It has an active status and is located at the top of the page. The 'Error Message' field contains the placeholder 'please use the college name in record info'. The 'Created By' field shows 'AMBRESH DESAI' with a creation date of '1/4/2022, 9:35 AM'. The 'Modified By' field also shows 'AMBRESH DESAI' with a modification date of '1/4/2022, 9:38 AM'. There are 'Edit' and 'Clone' buttons at the top right of the detail view. The browser address bar shows the URL for the validation rule setup page.



## Creating Validation Rules

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

- 1 = Now we have to click on the application object , then we have click on the validation rules.
- 2 = Now we have to click on new option to create new validation name as application form.
- 3 = Enter the details as per the guidelines.
- 4 = After creating successfully we can see the validation rule in the application form validation filed.

SCREENSHOT : OUTPUT

The screenshot shows the 'Application Form Validation Rule' detail page in the Salesforce Setup interface. The page title is 'Application Form Validation Rule'. The 'Rule Name' is 'Application\_Form\_Similar'. The 'Error Condition Formula' is:

```
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)  
)
```

The 'Error Message' is 'To stop any modification on the application form once a student record is created'. The 'Error Location' is 'Top of Page'. The rule is 'Active'. The 'Created By' is 'AMBRESH DESAI, 1/7/2022, 10:27 PM' and 'Modified By' is 'AMBRESH DESAI, 1/7/2022, 10:27 PM'. The page includes a sidebar with links like Details, Fields & Relationships, Page Layouts, etc.

The screenshot shows the 'Validation Rules' list page in the Salesforce Setup interface. The table has columns: RULE NAME, ERROR LOCATION, ERROR MESSAGE, ACTIVE, and MODIFIED BY. There is one item listed:

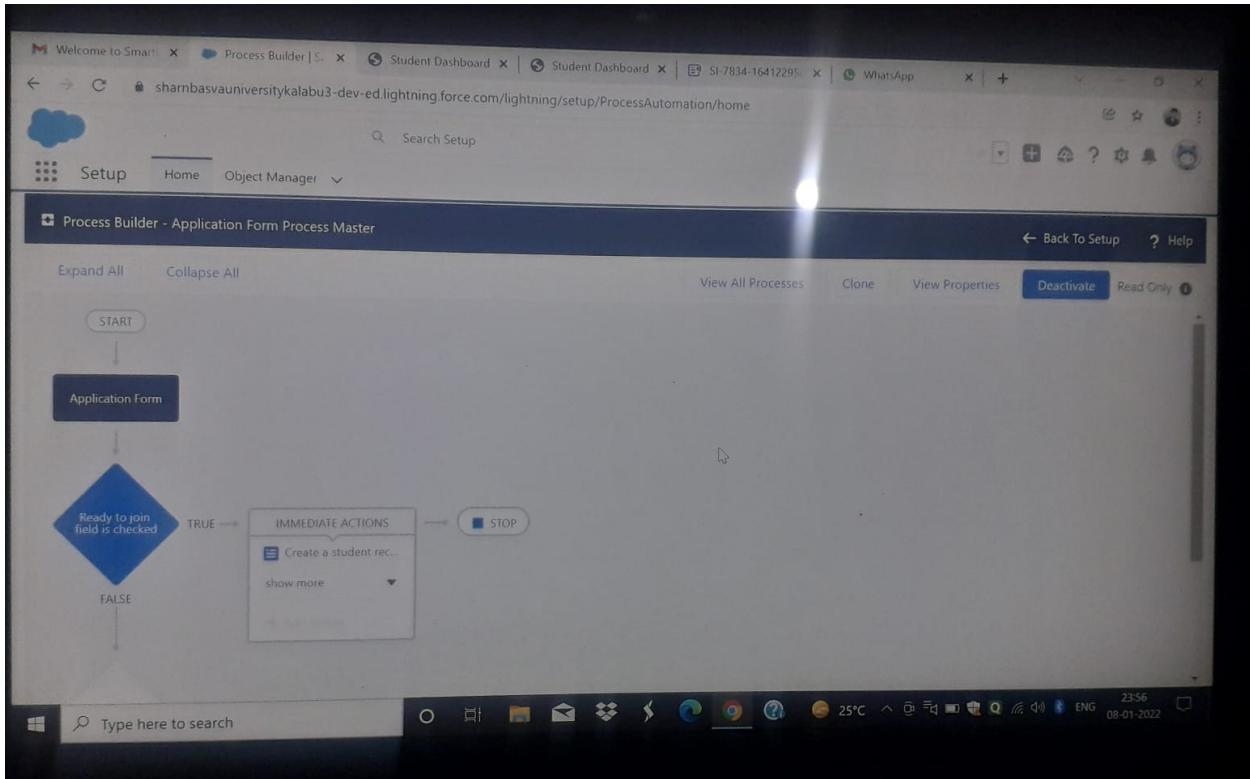
RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
Application_Form_Similar	Top of Page	To stop any modification on the application form once a student record is created	✓	AMBRESH DESAI, 1/7/2022, 10:27 PM

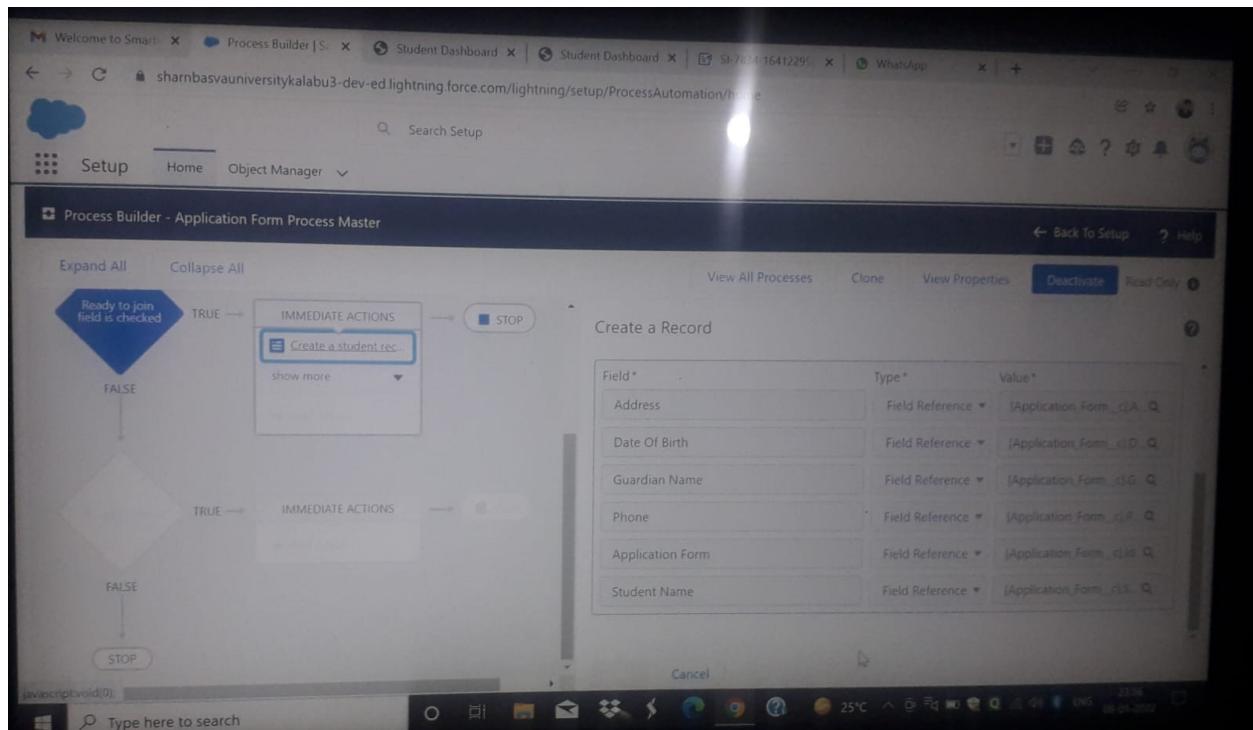
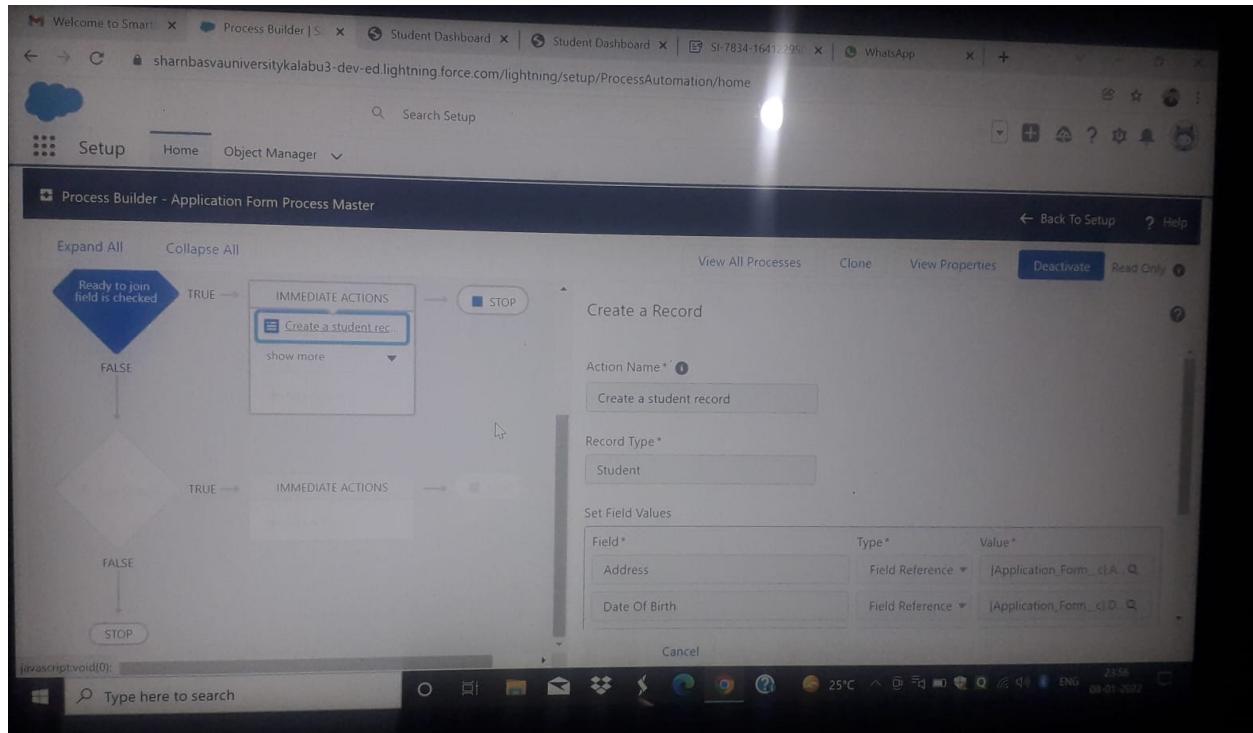
The sidebar on the left lists various setup categories: Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, and Related Lookup Filters.

## Process Automation

- 1 = Now we have to click on the process builder , then we have click on the new icon.
- 2 = Now we have to click on new option to create new process builder name as application form process master.
- 3 = Enter the details as per the guidelines.
- 4 = After creating successfully we can see the output .

### SCREENSHOT : OUTPUT

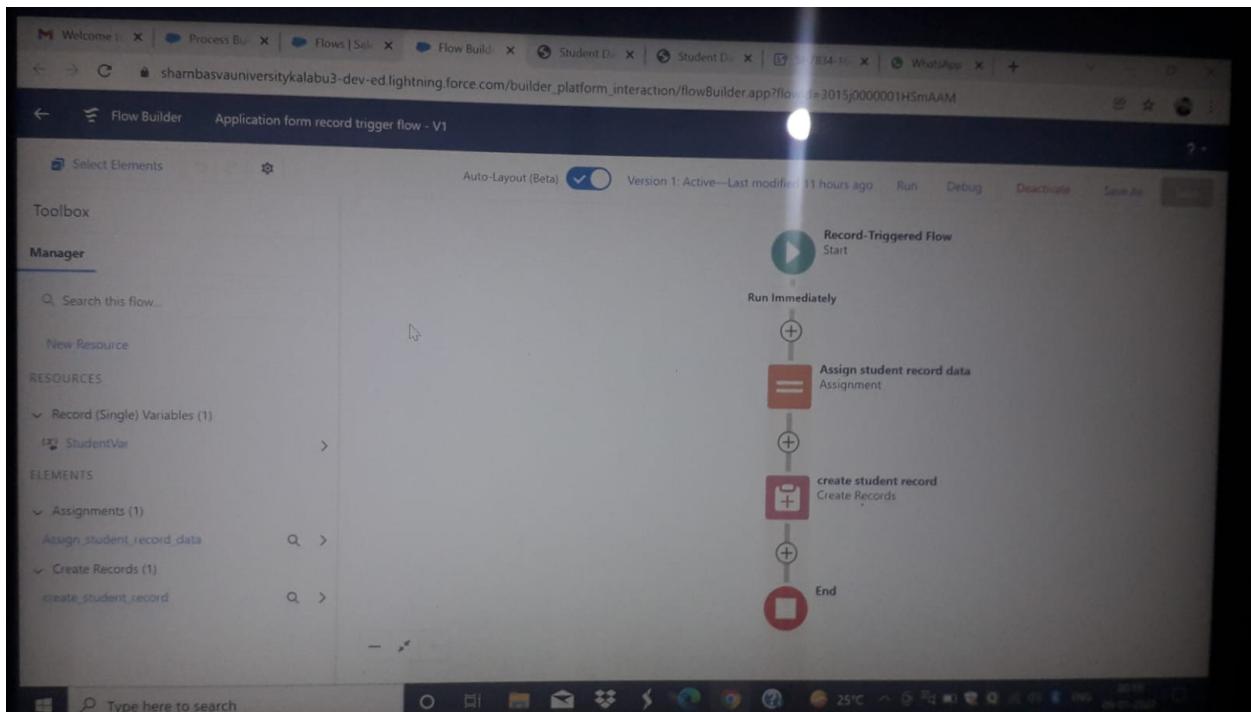




## Create The Student Record Using Flow

- 1 = Now we have to click on the process builder , then we have deactivate it.
- 2 = Now we have to click on new option to create new process flow name as application form record trigger.
- 3 = Enter the details as per the guidelines.
- 4 = After creating successfully we can see the output .

SCREENSHOT : OUTPUT



Then save it.

DAY 4:

TOPIC : COLLEGE MANAGEMENT APPLICATION

MILESTONE/ACTIVITIES : Automation ,Security, Sharing, Overview Of Apex.

Automation Tool : 1)Work Flow

- 2)Process Builder
- 3)Flow Builder

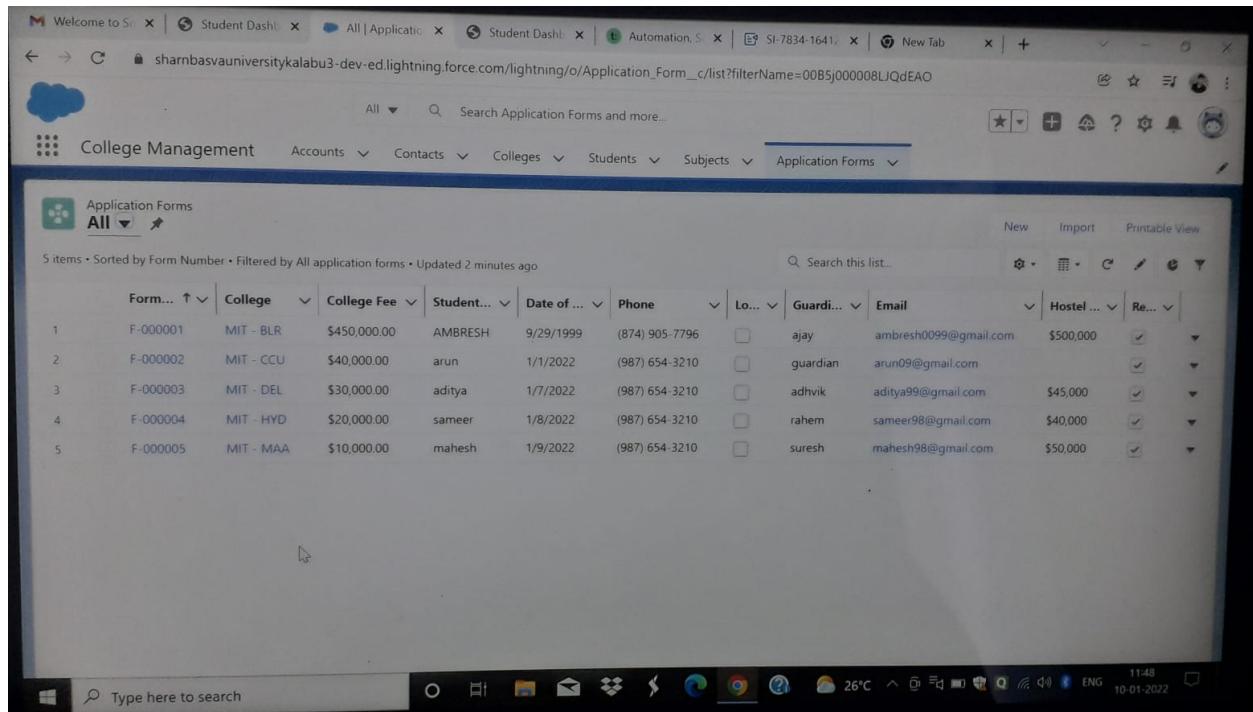
- 1 = Now we have to click on the process builder , then we have deactivate it.
- 2 = Now we have to click on the college management application, then select application form.
- 3 = Enter the details as per the guidelines.
- 3 = Now we have to select one application form and, enter the college name, student name, email id, date of birth , address, phone, guardian name,

and select ready to join.

4 = Then save it.

5 = After creating successfully we can see the output .

SCREENSHOT : OUTPUT



The screenshot shows a web-based application titled "College Management". The main menu includes "College Management", "Accounts", "Contacts", "Colleges", "Students", "Subjects", and "Application Forms". The "Application Forms" tab is selected. A sub-menu for "Application Forms" shows "All" selected. The main content area displays a table of 5 items, sorted by Form Number. The columns include: Form..., College, College Fee, Student..., Date of ..., Phone, Lo..., Guardi..., Email, Hostel..., and Re... . The data in the table is as follows:

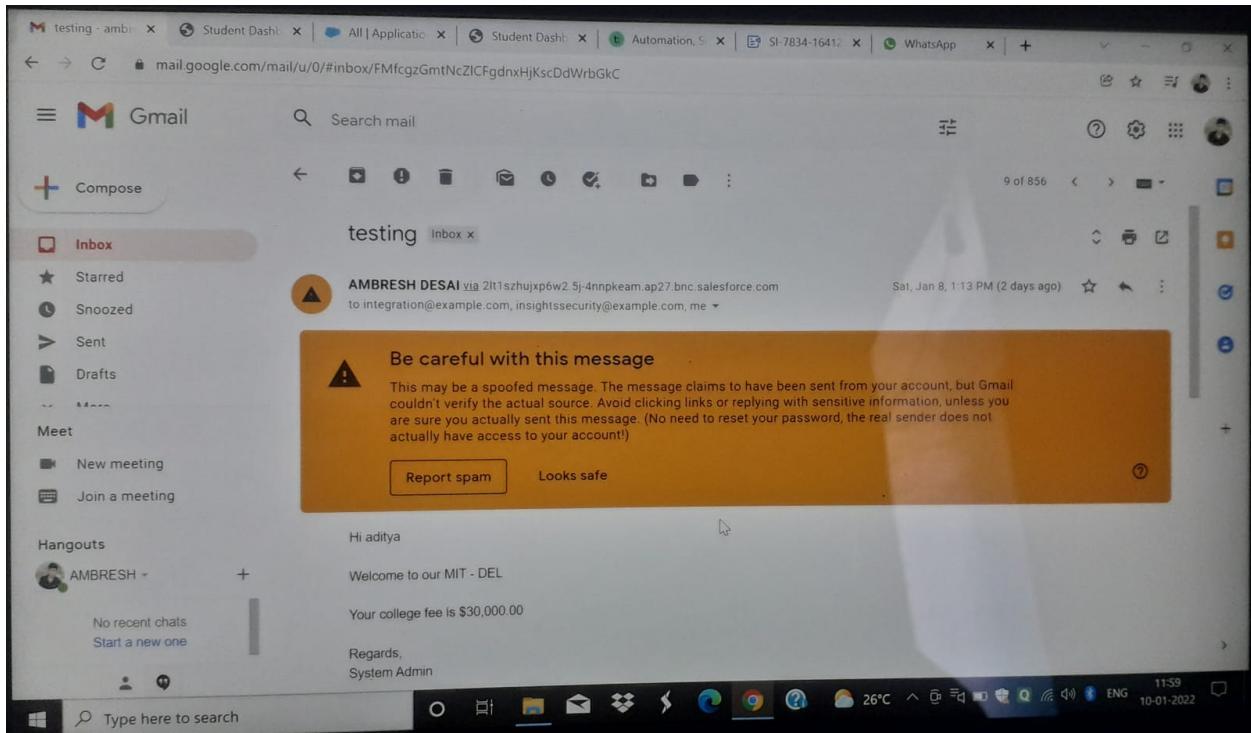
	Form...	College	College Fee	Student...	Date of ...	Phone	Lo...	Guardi...	Email	Hostel...	Re...
1	F-000001	MIT - BLR	\$450,000.00	AMBRESH	9/29/1999	(874) 905-7796	<input type="checkbox"/>	ajay	ambresh0099@gmail.com	\$500,000	<input checked="" type="checkbox"/>
2	F-000002	MIT - CCU	\$40,000.00	arun	1/1/2022	(987) 654-3210	<input type="checkbox"/>	guardian	arun09@gmail.com		<input checked="" type="checkbox"/>
3	F-000003	MIT - DEL	\$30,000.00	aditya	1/7/2022	(987) 654-3210	<input type="checkbox"/>	adhvik	aditya99@gmail.com	\$45,000	<input checked="" type="checkbox"/>
4	F-000004	MIT - HYD	\$20,000.00	sameer	1/8/2022	(987) 654-3210	<input type="checkbox"/>	rahem	sameer98@gmail.com	\$40,000	<input checked="" type="checkbox"/>
5	F-000005	MIT - MAA	\$10,000.00	mahesh	1/9/2022	(987) 654-3210	<input type="checkbox"/>	suresh	mahesh98@gmail.com	\$50,000	<input checked="" type="checkbox"/>

6 = Now we have to select one student details and we should attach that student details in related column.

7 = After saving the student name in related column, activate the process builder.

7 = Then we will receive a mail .

SCREENSHOT : OUTPUT



- 1 = Now we have to create the workflow rule.
- 2 = Now we have to search for workflow in the setup.
- 3 = Now select new workflow rule then enter.
- 4 = select the object application form,
- 5 = Enter the field name as Application Form Work flow Rule .
- 6 = Then enter the conditions as ready to join , equals ,true.
- 7 = Then save it.

SCREENSHOT : OUTPUT

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

- Page Header:** testing - ambri | All | Application | Student Dashb | Automation, S | Workflow Rule | SI-7834-16412 | WhatsApp
- Page Title:** Workflow Rules
- Left Sidebar:** Q workflow, Process Automation (Workflow Actions: Email Alerts, Field Updates, Outbound Messages, Send Actions, Tasks, Workflow Rules), Environments (Monitoring: Time-Based Workflow).
- Content Area:**
  - Section:** All Workflow Rules
  - Description:** Configure your organization's workflow by creating workflow rules. Each workflow rule consists of:
    - Criteria that cause the workflow rule to run.
    - Immediate actions that execute when a record matches the criteria. For example, Salesforce can automatically send an email that notifies the account team when a new high-value opportunity is created.
    - Time-dependent actions that queue when a record matches the criteria, and execute according to time triggers. For example, Salesforce can automatically send an email reminder to the account team if a high-value opportunity is still open ten days before the close date.
  - View:** All Workflow Rules, Create New View
  - Table:** A list of workflow rules with columns: Action (Edit | Del | Activate), Rule Name (Application Form WF), Description, Object (Application Form), and Status (Active). The table includes navigation links A through Z and a link to Other.
- Bottom:** Type here to search, system tray with icons for weather (26°C), battery, signal, and date/time (10-01-2022, 12:18).

The screenshot shows the details of a specific Workflow Rule named "Application Form WF".

**Workflow Rule Detail:**

Rule Name	Object
Application Form WF	Application Form

**Evaluation Criteria:** Evaluate the rule when a record is created, and any time it's edited to subsequently meet criteria.

**Description:** Application Form: Ready To Join EQUALS True

**Created By:** AMBRESH DESAI, 1/9/2022, 10:40 PM    **Modified By:** AMBRESH DESAI, 1/9/2022, 10:40 PM

**Workflow Actions:** Edit

**Immediate Workflow Actions:** No workflow actions have been added.

**Bottom:** Type here to search, system tray with icons for weather (26°C), battery, signal, and date/time (10-01-2022, 12:18).

DAY 5 :

TOPIC : COLLEGE MANAGEMENT APPLICATION

MILESTONE/ACTIVITIES : Primitive DataTypes, Collections

Automation Tool : 1)Application form object

2)college object

3)student object

For Application Form

1 = Now we have to click on the setup , then we have to select developer console in it.

2 = Now we have to click on the file folder and select apex class .

3 = Enter the details as per the guidelines, for application form.

3 = Now we have to enter the query of application form.

Student\_Name\_\_c

Ready\_To\_Join\_\_c

Date\_of\_Birth\_\_c

Guardian\_Name\_\_c

College\_Fee\_\_c

Address\_\_c.

For College object

1 = Now we have to click on the setup , then we have to select developer console in it.

2 = Now we have to click on the file folder and select apex class .

3 = Enter the details as per the guidelines, for college object.

3 = Now we have to enter the query of college object.

Name

College\_Name\_\_c

College\_Fee\_\_c

College\_Email\_\_c

CreatedById

## For Student object

- 1 = Now we have to click on the setup , then we have to select developer console in it.
- 2 = Now we have to click on the file folder and select apex class .
- 3 = Enter the details as per the guidelines, for student object.
- 3 = Now we have to enter the query of student object.

Student\_Name\_\_c

Application\_Form\_\_c

DateOfBirth\_\_c

College\_Name\_\_c

Guardian\_Name\_\_c

Address\_\_c

## SCREENSHOT : OUTPUT

The screenshot shows the Salesforce Developer Console interface. At the top, there are tabs for 'Logs', 'Tests', 'Checkpoints', 'Query Editor' (which is selected), 'View State', 'Progress', and 'Problems'. Below the tabs, a query is displayed:

```
select Student_Name__c, Guardian_Name__c, Date_of_Birth__c, college_r.College_Name__c, college_r.No_of_Seats__c, college_r.College_Email__c from Application_Form__c
```

The 'Query Results - Total Rows: 6' section displays the following data:

Student_Name__c	Guardian_Name__c	Date_of_Birth__c	College__r.College_Name__c	College__r.No_of_Seats__c	College__r.College_Email__c
Rounak	rounat	1999-01-26	MIT - HYD	5000-8000	mit@hyd.com
Likhita	Guardian	2002-12-29	MIT - CUJ	5000-8000	mit@cuja.com
Namaz	namaz	2006-01-15	MIT - BLR	5000-8000	mit@cuja.com
Vishwanath	Vash	2005-01-04	MIT - CUJ	5000-8000	mit@cuja.com
Viney	delhi guardian	2006-01-04	MIT - DEL	1500-3000	mit@deli.com
Rakesh	Rakesh Roshan	2000-01-01	MIT - BLR		mit@blr.com

Below the results, an 'Enter Apex Code' window is open, showing the following code:

```
1 List<Application_Form__c> aftList = [select Student_Name__c, Guardian_Name__c, Date_of_Birth__c from Application_Form__c];
2 
3 // debug statements
4 System.debug()
5
```

## DAY 6 :

### TOPIC : COLLEGE MANAGEMENT APPLICATION

#### MILESTONE/ACTIVITIES : Batch Apex

- 1)Create a batch appex for application form
- 2)Create a schedular class

## For Application Form

- 1 = Now we have to click on the setup , then we have to select developer console in it.
- 2 = Now we have to click on the file folder and select apex class .
- 3 = Enter the details as per the guidelines, for application form.
- 3 = Now we have to enter the code of application form.

## Create A Batchapex For Application Form

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

```
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);
    }
    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[]{'your email address'};  
mail.setToAddresses(emailAddess);  
Messaging.sendEmail(new Messaging.SingleEmailMessage[]{ mail } );  
}  
}
```

## SCREENSHOT : OUTPUT

```
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    }
21 }
```

The screenshot shows a Salesforce developer console window titled "sharnbasvauniversitykalabu3-dev-ed.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage". The active tab is "ApplicationBatchTest.apxc". The code editor contains the following Apex code:

```
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 }  
20 public void finish(Database.BatchableContext bc){  
21     // emails ,  
22     Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage();  
23     // address, subject, content( data to sent to admins)  
24     mail.setSubject(' Application form and student record data as of today ');  
25     mail.setPlainTextBody(' Total no of application form records are : '+totalForms+ ' out of which no of students as  
26     String[] emailAddess = new String[]{testambresh0099testmill@gmail.com};  
27     mail.setToAddresses(emailAddess);  
28     Messaging.sendEmail(new Messaging.SingleEmailMessage[]{ mail } );  
29 }  
30 }
```

The status bar at the bottom shows "Logs, Tests, and Problems" and a search bar. The system tray indicates it's 15:29 on 14-01-2022.

The screenshot shows a Salesforce developer console window titled "sharnbasvauniversitykalabu3-dev-ed.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage". The active tab is "ApplicationBatchTest.apxc". The code editor contains the same Apex code as the first screenshot:

```
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 }  
20 public void finish(Database.BatchableContext bc){  
21     // emails ,  
22     Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage();  
23     // address, subject, content( data to sent to admins)  
24 }
```

The interface is identical to the first screenshot, including the tabs at the bottom labeled "Logs", "Tests", "Checkpoints", "Query Editor", "View State", "Progress", and "Problems". The status bar at the bottom shows "Logs, Tests, and Problems" and a search bar. The system tray indicates it's 15:29 on 14-01-2022.

### For Schedular class

- 1 = Now we have to click on the setup , then we have to select developer console in it.
- 2 = Now we have to click on the file folder and select apex class .
- 3 = Enter the details as per the guidelines, for schedular class.
- 3 = Now we have to enter the code of schedular class.

### Create A Schedular Class

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

```
public class applicationschedule 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..

SCREENSHOT : OUTPUT

The screenshot shows a browser window with the URL [sharnbasvauniversitykalabu3-dev-ed.my.salesforce.com/\\_ui/common/apex/debug/ApexCSIPage](sharnbasvauniversitykalabu3-dev-ed.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage). The page displays a code editor for an Apex class named `applicationschedule.apxc`. The code implements the `Schedulable` interface and contains a single method `execute` that creates and executes a batch test. Below the code editor is a navigation bar with tabs: Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Problems tab is currently selected. At the bottom of the screen is a Windows taskbar with various icons and system status information.

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

The screenshot shows a browser window with multiple tabs open, including 'Welcome to', 'Student Da', 'SI-7834-16', 'Overview C', 'New Tab', 'Object Ma', 'Developer', and 'WhatsApp'. The main content area is a code editor for an Apex trigger named 'applicationFormTrigger.aptx'. The code is as follows:

```
1 trigger applicationFormTrigger on Application_Form__c (before insert, after insert, before update, after update) {
2
3     // Context variables : trigger.new, trigger.old, trigger.newMap, trigger.oldMap
4     /*object - Application_Form__c
5     trigger.new trigger.old = list of Application_Form__c
6     trigger.new or trigger.old = List<Application_Form__c>
7     trigger.newMap, trigger.oldMap = Map<id, Application_Form__c>
8
9     when you do a before insert, you dont have trigger.old
10    when you do after insert operation, both trigger.old and trigger.new is same.
11
12    in edit operation , your trigger.old is different then trigger.new
13}
```

Below the code editor, there are tabs for 'Logs', 'Tests', 'Checkpoints', 'Query Editor', 'View State', 'Progress', and 'Problems'. The 'Problems' tab is selected. At the bottom of the screen, the Windows taskbar is visible with icons for File Explorer, Mail, Task View, Edge, Google Chrome, and others, along with system status like battery level, temperature (21°C), and date/time (14-01-2022, 15:37).

The screenshot shows the Salesforce Developer Console interface. The top navigation bar includes tabs for Welcome, Student Data, Overview, New Tab, Object Manager, Developer, and WhatsApp. Below the tabs, the URL is sharnbasvauniversitykalabu3-dev-ed.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage. The main area displays the code for the apex class 'formhelper1.apxc'. The code is as follows:

```
1 public class formhelper1 {
2
3     public static void getoldnewvalues(Map<id, Application_Form__c> formoldMap, List<Application_Form__c> formlist ){
4
5         for(Application_Form__c fm : formlist){
6             Application_Form__c oldFormRecord = formoldMap.get(fm.id);
7             System.debug(' old values :'+oldFormRecord.Student_Name__c);
8             System.debug(' new values : '+fm.Student_Name__c+' old values :'+formoldMap.get(fm.id).Student_Name__c);
9         }
10    }
11 }
```

Below the code editor, there are tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Problems tab is selected. At the bottom of the screen, the Windows taskbar is visible with various icons and the system tray showing the date and time.

## DAY 7:

### TOPIC : COLLEGE MANAGEMENT APPLICATION

#### MILESTONE/ACTIVITIES:

- 1) Triggers

Now we have to update the formhelper1

1 = Now we have to click on the setup , then we have to select developer console in it.

2 = Now we have to click on the file folder and select apex class .

3 = Enter the details as per the guidelines, formhelper1.

3 = Now we have to enter the code of formhelper1.

SCREENSENAPSHOT : OUTPUT

Developer Console - Google Chrome  
gmailcomtest-4e-dev-ed.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage

File • Edit • Debug • Test • Workspace • Help •

applicationFormTrigger.apql • Formhelper.Lambda •

Code Coverage: None • API Version: 51.0

```
1 * public class formhelper1 {
2 *
3 *     public static void getoldNewValues(Map<id, Application_Form__c> formoldMap, List<Application_Form__c> formlist){
4 *         for(Application_Form__c fm : formlist){
5 *
6 *             Application_Form__c oldFormRecord = formoldMap.get(fm.id);
7 *             System.debug(' old value : '+oldFormRecord.Student_Name__c);
8 *             System.debug(' new values : '+fm.Student_Name__c+' old values : '+formoldMap.get(fm.id).Student_Name__c);
9 *
10 *             if(fm.Student_Name__c != formoldMap.get(fm.id).Student_Name__c){
11 *                 // do some operation
12 *             }
13 *         }
14 *     }
15 * }
16 * public static void preventRecordDeletion(List<Application_Form__c> formlist){
17 *
18 *     Set<id> formIdSet = new Set<id>();
19 *     for(Application_Form__c fm : formlist){
20 *         formIdSet.add(fm.id);
21 *     }
22 *     if(formIdSet != null && formIdSet.size() > 0)
23 * }
24 }
```

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

- Title Bar:** gmailcomtest-4e-dev-ed.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage
- Toolbars:** File, Edit, Debug, Test, Workspace, Help.
- Tabs:** applicationFormTrigger.apql, FormHelper Lapis, Code Coverage, Notes, API Version: 53.
- Code Area:** The code is for a static method named preventRecordDeletion. It uses a Map to store student records and checks if an application form has a student record associated with it before allowing deletion.

```
11.     if(fm.Student_Name__c != formIdMap.get(fm.id).Student_Name__c){
12.         // do some operation
13.     }
14. }
15.
16. public static void preventRecordDeletion(List<Application_Form__c> formlist){
17.
18.     Set<id> formIdSet = new Set<id>();
19.     // map to store , key as application_form id and value as student record.
20.     Map<id, Student__c> formidStudentRecMap = new Map<id, Student__c>(); // map of application form id and student record.
21.     for(Application_Form__c fm : formlist){
22.         formIdSet.add(fm.id);
23.     }
24.     if(formIdSet != null && formIdSet.size() > 0){
25.         List<Student__c> stdList = [select id, Application_Form__c from Student__c where Application_Form__c IN :formIdSet];
26.         if(stdList.size() > 0){
27.             for(Student__c std : stdList){
28.                 formidStudentRecMap.put(std.Application_Form__c, std); // for Map put() to add data into map
29.             }
30.         }
31.         if(formidStudentRecMap != null && formidStudentRecMap.values().size() > 0){
32.             for(Application_Form__c fm : formlist){
33.                 if(formidStudentRecMap.containsKey(fm.id) && formidStudentRecMap.get(fm.id) != null){
34.                     fm.addError(' you are not allowed to delete the application form record as it has a student associated with it. ');
35.                 }
36.             }
37.         }
38.     }
39. }
40.
41. }
```

- Log, Tests, and Problems:** Log, Tests, and Problems tabs are visible at the bottom.

The screenshot shows the Developer Console interface with the following details:

- Title Bar:** Developer Console - Google Chrome, gmailcomtest-4e-dev-ed.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage
- Toolbars:** File, Edit, Debug, Test, Workspace, Help.
- Logs:** Log executed by anonymous user on 01/11/2022, 5:28:14 PM, Student\_\_c@5:24 PM, Student\_\_c@5:24 PM.
- Code Coverage:** Code Coverage, Notes, API Version: 53.
- Code Area:** The same preventRecordDeletion method is shown, identical to the one in the IDE.

```
14. }
15.
16. public static void preventRecordDeletion(List<Application_Form__c> formlist){
17.
18.     Set<id> formIdSet = new Set<id>();
19.     // map to store , key as application_form id and value as student record.
20.     Map<id, Student__c> formidStudentRecMap = new Map<id, Student__c>(); // map of application form id and student record.
21.     for(Application_Form__c fm : formlist){
22.         formIdSet.add(fm.id);
23.     }
24.     if(formIdSet != null && formIdSet.size() > 0){
25.         List<Student__c> stdList = [select id from Student__c where Application_Form__c IN :formIdSet];
26.         if(stdList.size() > 0){
27.             for(Student__c std : stdList){
28.                 formidStudentRecMap.put(std.Application_Form__c, std);
29.             }
30.         }
31.         if(formidStudentRecMap != null && formidStudentRecMap.values().size() > 0){
32.             for(Application_Form__c fm : formlist){
33.                 formIdSet.add(fm.id);
34.             }
35.         }
36.     }
37. }
38. }
```

- Query Editor:** A query is being run in the Query Editor tab: select id, name, application\_form\_\_c from Student\_\_c where Application\_Form\_\_c = '0011|00000065MJAAT'.

DAY 8:

TOPIC : COLLEGE MANAGEMENT APPLICATION

MILESTONE/ACTIVITIES:

- 1)Introduction to LWC
- 1)What is LWC?
- 2)Why LWC?
- 3)Advantages of LWC
- 4)Familiar with LWC coding
- 5)Software needed for LWC
- 6)Familiar with Javascript
- 7>Create your first component
- 8>Data binding
- 9)Applying CSS
- 10)Conditional Rendering
- 11)Looping And Iteration
- 12)Javascript Properties
- 13)Life Cycle Hooks
- 14)Event Communication
- 15)Working with salesforce data
- 16)Introduction to LDS
- 17)Apex class with LWC
- 18)Navigation services
- 19>Data table services

What is LWC?

- 1)LWC is a new programming model to develop salesforce lightning components.
- 2)IT'S a UI framework that is built using native html and modern javascript.

Benefits of LWC?

- 1)More standards.
- 2)Less proprietary.
- 3)Common Component model.
- 4)Transferable skills.
- 5)Better Performance.

\*Now we have to download LWC in our salesforce org.

\*We have to download visual studio code in our windows.

Introduction to Javascript :

- 1)Javascript is a cross-platform, object oriented programming language used to make webpages interative.
- 2)Every component should have javascript files.
- 3>To develop lightning components, use the latest version of javascript.

\*Now we have to download salesforce extension pack.

DAY 9:

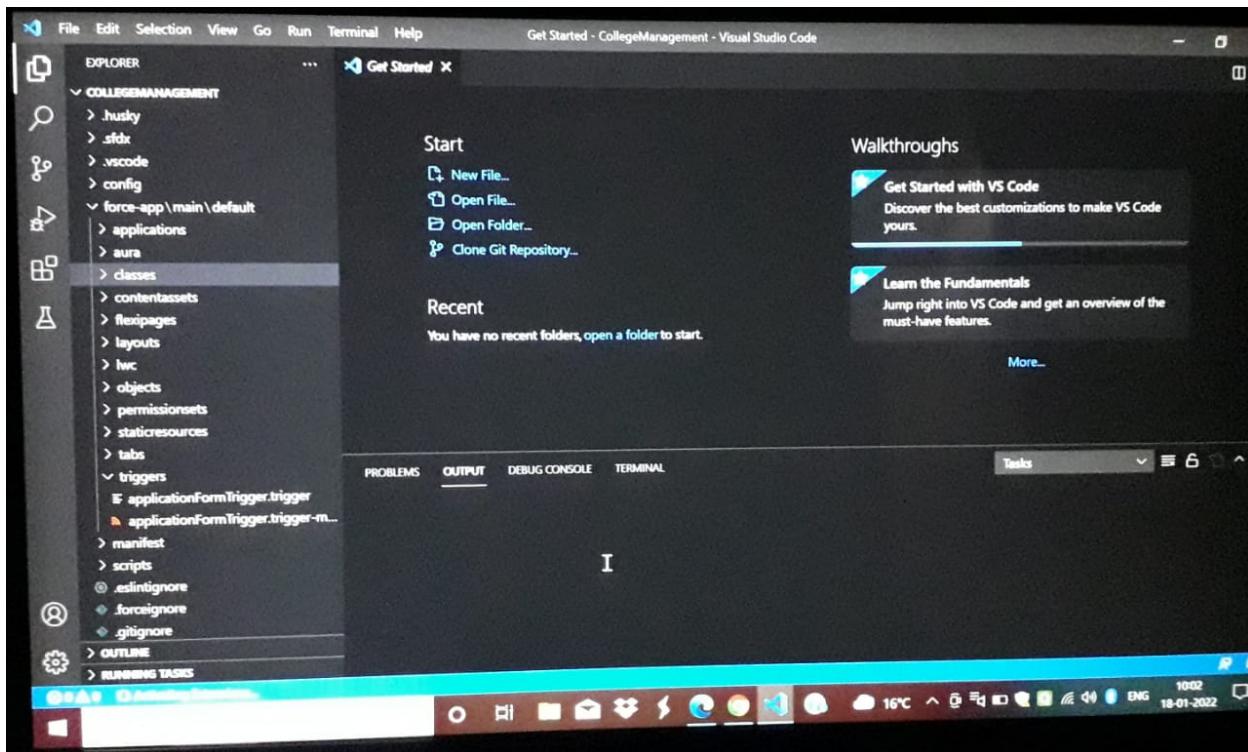
TOPIC : COLLEGE MANAGEMENT APPLICATION

MILESTONE/ACTIVITIES:

- 1)We Have To Install Visual Studio Code.
- 2)We Have To Install Salesforce Extension Pack.
- 3)We Have To Install Salesforce CLI Pack.
- 4)We have To Install Salesforce LWC Editor.

1)We Have To Install Visual Studio Code.

SCREENSHOT : OUTPUT



```
C:\Users\HP> Error: Run sfdx help for a list of available commands.
C:\Users\HP>sfdx
Salesforce CLI

VERSION
  sfdx-cli/7.133.0 win32-x64 node-v16.13.1

USAGE
  $ sfdx [COMMAND]

TOPICS
  alias    manage username aliases
  auth     authorize an org for use with the Salesforce CLI
  config   configure the Salesforce CLI
  force    tools for the Salesforce developer
  info     access cli info from the command line
  plugins  add/remove/create CLI plug-ins

COMMANDS
  autocomplete  display autocomplete installation instructions
  commands      list all the commands
  help          display help for sfdx
  plugins       list installed plugins
  update        update the sfdx CLI
  which         show which plugin a command is in

C:\Users\HP>
```

2) We Have To Install Salesforce Extension Pack.

SCREENSHOT : OUTPUT

The screenshot shows a Windows desktop environment with Visual Studio Code open. The code editor interface includes:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Explorer Panel:** Shows a project structure for "COLLEGEMANAGEMENT" containing files like .husky, .sfdx, .vscode, config, force-app, manifest, scripts, and various ignore files (.eslintignore, .forceignore, .gitignore, .prettierignore, .prettierrc, jest.config.js, package.json, README.md, sfdx-project.json).
- Start Panel:** Buttons for New File..., Open File..., and Open Folder...; a message stating "You have no recent folders, open a folder to start."
- Walkthroughs:** "Get Started with VS Code" and "Learn the Fundamentals" sections.
- Terminal:** Displays the output of SFDX commands:

```
Starting SFDX: Refresh SObject Definitions
08:01:37.498 sfdx:object:definitions refresh
Processed 16 Standard sObjects
08:01:37.685 sfdx:object:definitions refresh
ended with exit code 0
```
- Bottom Status Bar:** Shows system icons for battery, signal, and date/time (18-01-2022).

\*Now we have to install or download the apex classes,  
and apex triggers from our salesforce org.

- 1) Application Batch test.
- 2) Application Schedule.
- 3) Form Helper 1.
- 4) My first class.

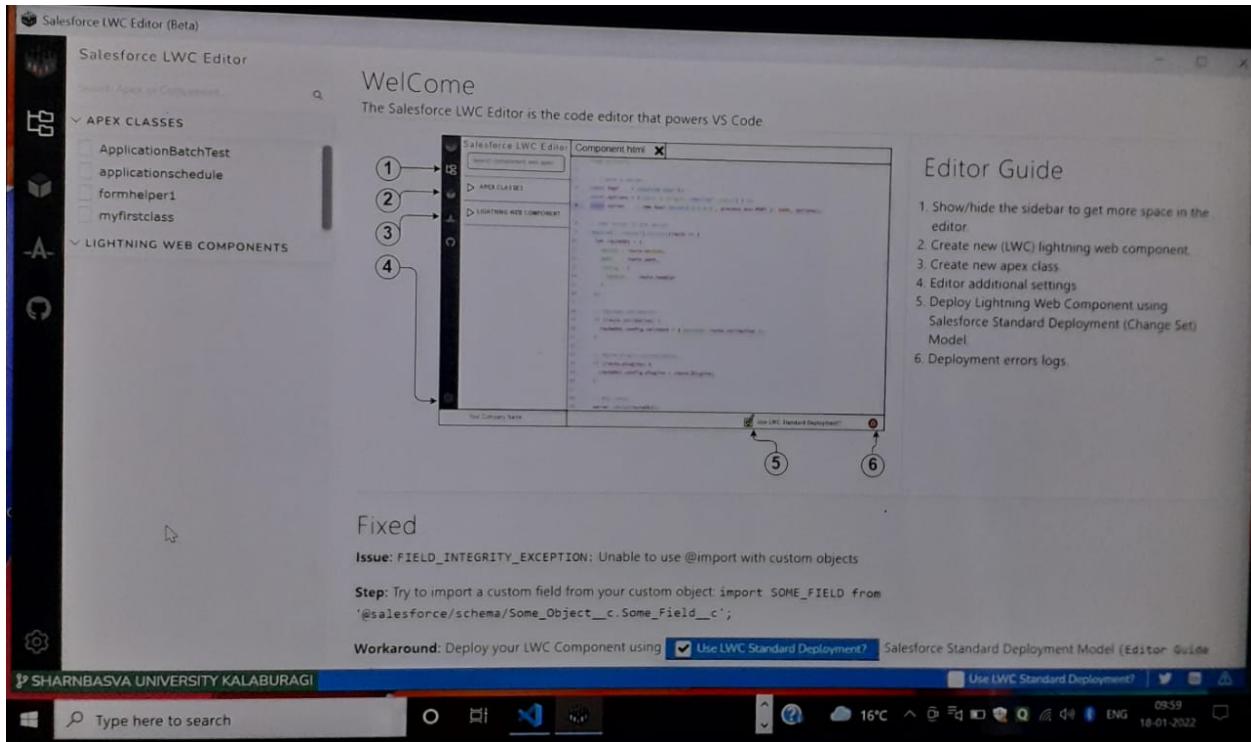
The screenshot shows the Visual Studio Code interface with the following details:

- File Path:** force-app/main/default/classes/applicationschedule.cls
- Code Content:**

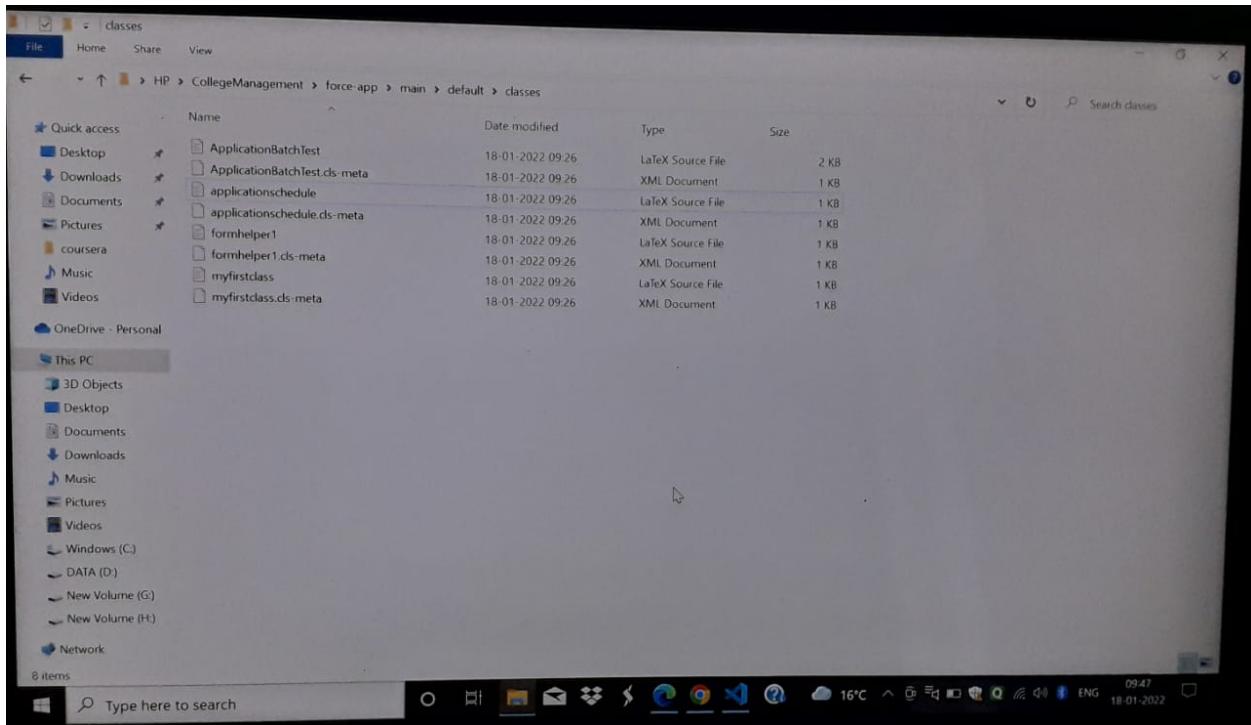
```
1  public class applicationschedule implements Schedulable{
2
3      public void execute(SchedulableContext sc){
4
5          ApplicationBatchTest abt = new ApplicationBatchTest();
6
7          Database.executeBatch(abt, 400); // 200 to 2000
8
9      }
10
11 }
```

- Explorer View:** Shows the project structure under "COLLEGEMANAGEMENT". The "classes" folder contains "ApplicationBatchTest.cls", "ApplicationBatchTest.cls-meta.xml", and "applicationschedule.cls".
- Bottom Status Bar:** Displays system information including battery level, temperature (16°C), and date/time (18-01-2022).

4)We have To Install Salesforce LWC Editor.



- 1) Now we have to select the project folder from our internal pc.
- 2) Then name it as college management folder.
- 3) We can see the apex classes and apex triggers.



## DAY 10:

### TOPIC : COLLEGE MANAGEMENT APPLICATION

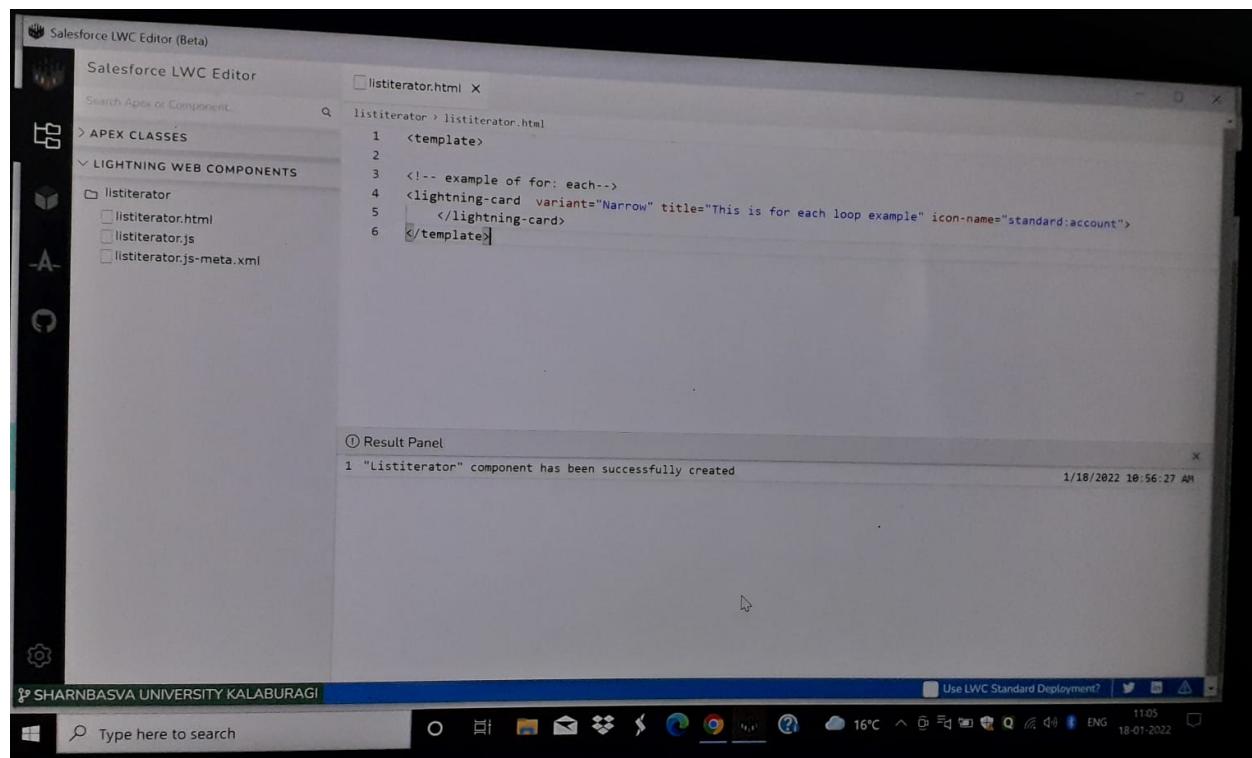
MILESTONE/ACTIVITIES: Lightning Web Components.

- 1)Create College Data Table Component (Apex Class).
- 2)Create College Data Table Component (HTML File).
- 3)Create College Data Table Component (Java script File).
- 4)Create College Data Table Component(Meta File).

#### 1)Create College Data Table Component (Apex Class)

```
public static List<ApplicationForm__c> getapplicationvalues(id CollegelId)
{
```

```
List<ApplicationForm_c> formlist = [SELECT ID, College_Fees_c.Name,  
Date_Of_Birth_c,Emailc,Hostel_Feesc, Student_Name_c FROM  
ApplicationForm_c WHERE College_c =:CollegeId];  
  
return formList;  
}  
  
}
```



2)Create College Data Table Component (HTML File).

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

3)Create College Data Table Component (Java script File).

```
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;
error;

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

```

4)Create College Data Table Component(Meta File).

```

<?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>

```

## SCREENSHOT : OUTPUT

The screenshot shows the Salesforce LWC Editor (Beta) interface. On the left, the file tree displays components under 'APELLATION WEB COMPONENTS' and 'APEX CLASSES'. The 'collegeDataTable' component is selected, showing its files: collegeDataTable.html, collegeDataTable.js, collegeDataTable.css, collegeDataTable.js-meta.xml, and listIterator. The 'listIterator' file is open, displaying the following Apex code:

```
1  public class utilityControl{
2      {
3          @AuraEnabled(cacheable=true)
4          public static List<Application_Form__c> getapplicationvalues(id CollegeId)
5          {
6              List<ApplicationForm__c> formlist = [SELECT ID, College_Fees__c,Name,
7 Date_of_Birth__c,Email__c,Hostel_Fees__c, Student_Name__c FROM
8 Application_Form__c WHERE College__c =:CollegeId];
9
10         return formList;
11     }
12
13
14 }
15
```

In the bottom right corner of the code editor, there is a message: "1 'CollegeDataTable' component has been successfully created". The system status bar at the bottom indicates the date and time: 1/18/2022 11:18:38 AM.

## SCREENSHOT : OUTPUT

