

Project Title: SALESFORCE PROJECT READY

Topic: Create Your Salesforce Developer Org To Get Started

Milestone / Activities:

1. Create Your Salesforce Developer Org To Get Started
2. Account Activation
3. Salesforce Login

Detailed Description:

Creating a developer org in salesforce.

1. Go to [developers.salesforce.com](https://developer.salesforce.com)
2. Click on sign up.
3. On the sign up form, enter the following details :
First name & Last name
Email
Role : Developer
Company : College Name
Country : India
Postal Code : pin code
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

Click on sign up after filling these.

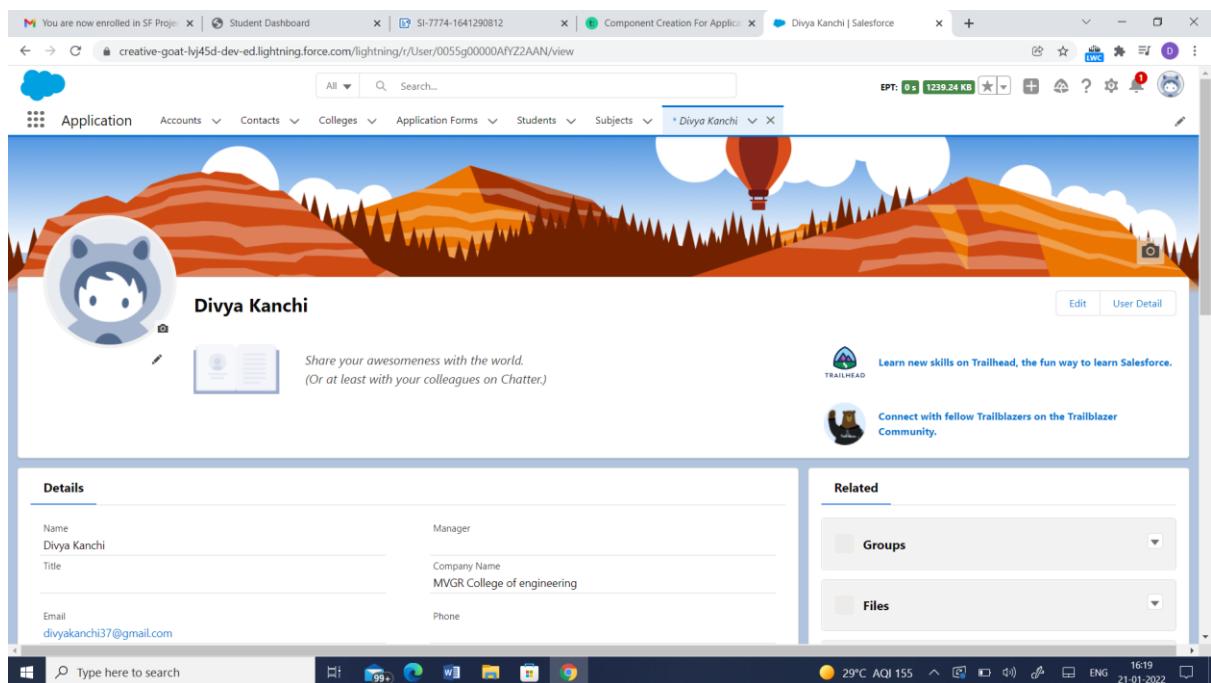
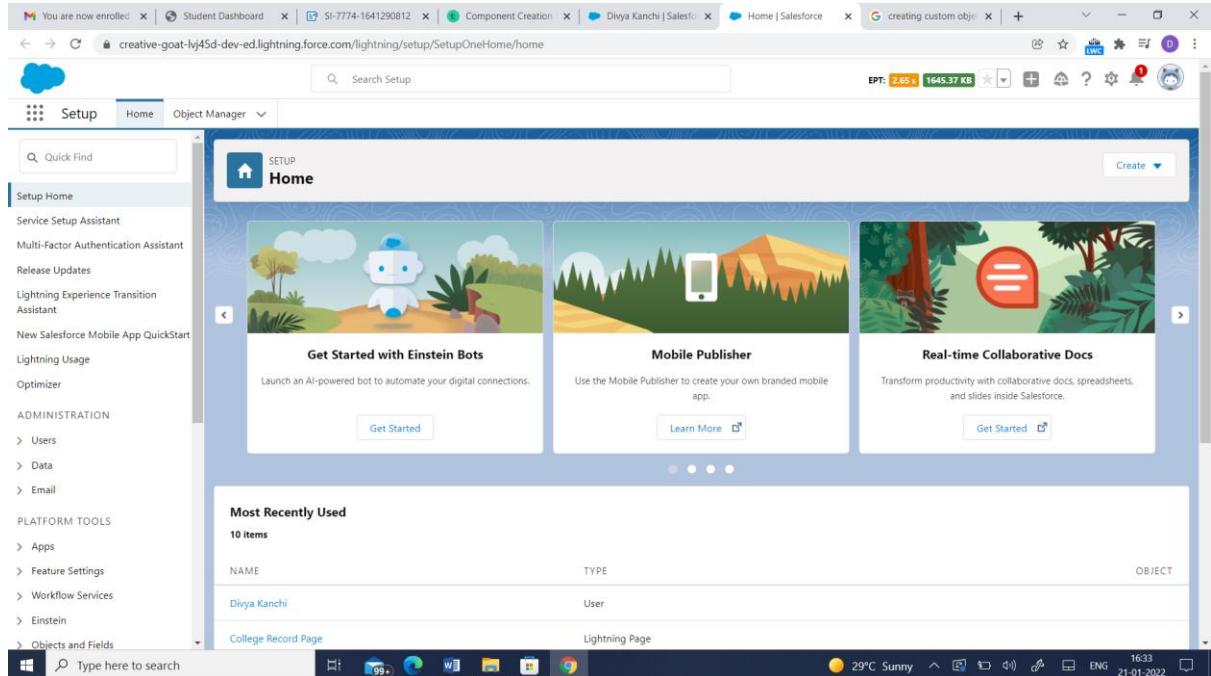
Account Activation :

Go to the inbox of the email that you used while signing up. Click on the verify account to activate your account. The email may take 5-10mins, as allocating a developer org will take time.

Login To Your Salesforce Account :

1. Go to [salesforce.com](https://www.salesforce.com) and click on login.
2. Enter the username and password that you just created.
3. After login this is the home page which you will see.

OUTCOME :



Topic: Custom Object Creation

- Milestone / Activities:**
1. Creation Of College Object
 2. Creation Of Application Form Object
 3. Creation Of Student Object
 4. Creation Of Subject Object

Detailed Description:

1.) Creation of College Object:

Log in to your Salesforce account. Click Setup at the upper-right corner. Under the Build section, click Create and select Objects. To create a custom object, click New Custom Object. Enter College in Label, Plural Label as Colleges, and Object Name.

2.) Creation of Application Form Object:

Log in to your Salesforce account. Click Setup at the upper-right corner. Under the Build section, click Create and select Objects. To create a custom object, click New Custom Object. Enter Application Form in Label, Plural Label as Application Forms, Object Name.

3.) Creation of Student Object:

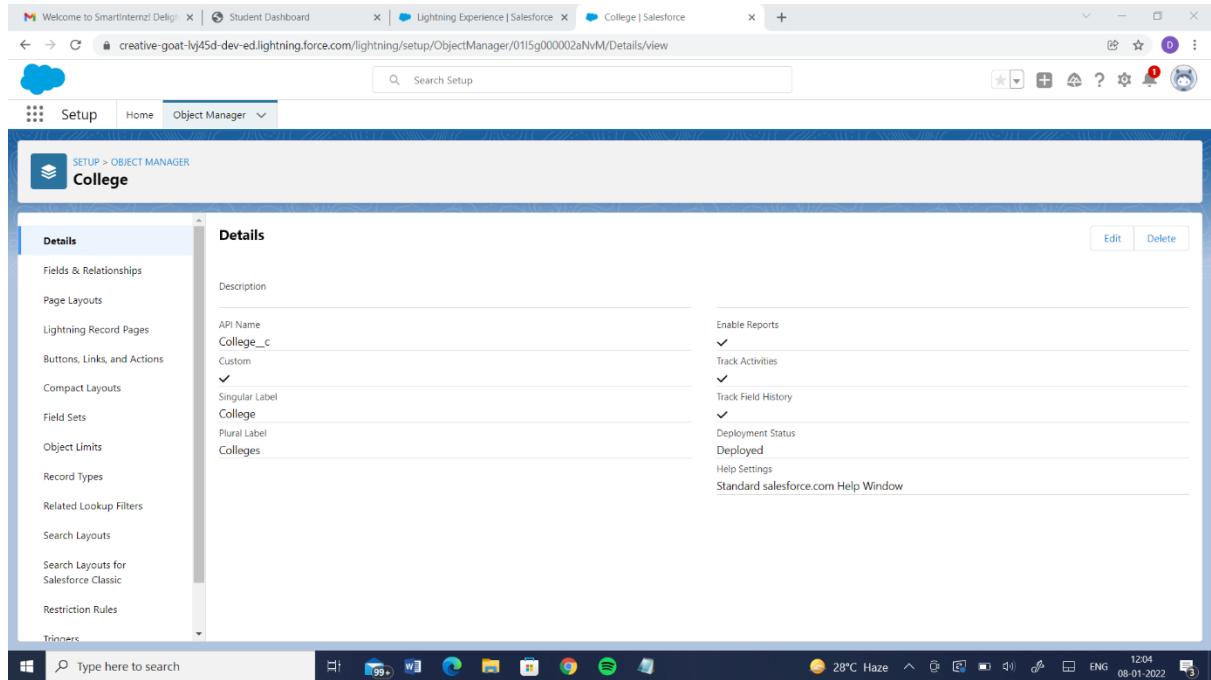
Log in to your Salesforce account. Click Setup at the upper-right corner. Under the Build section, click Create and select Objects. To create a custom object, click New Custom Object. Enter Student in Label, Plural Label as Students, Object Name.

4.) Creation of Subject Object:

Log in to your Salesforce account. Click Setup at the upper-right corner. Under the Build section, click Create and select Objects. To create a custom object, click New Custom Object. Enter Subject in Label, Plural Label as Subjects, Object Name.

OUTCOME :

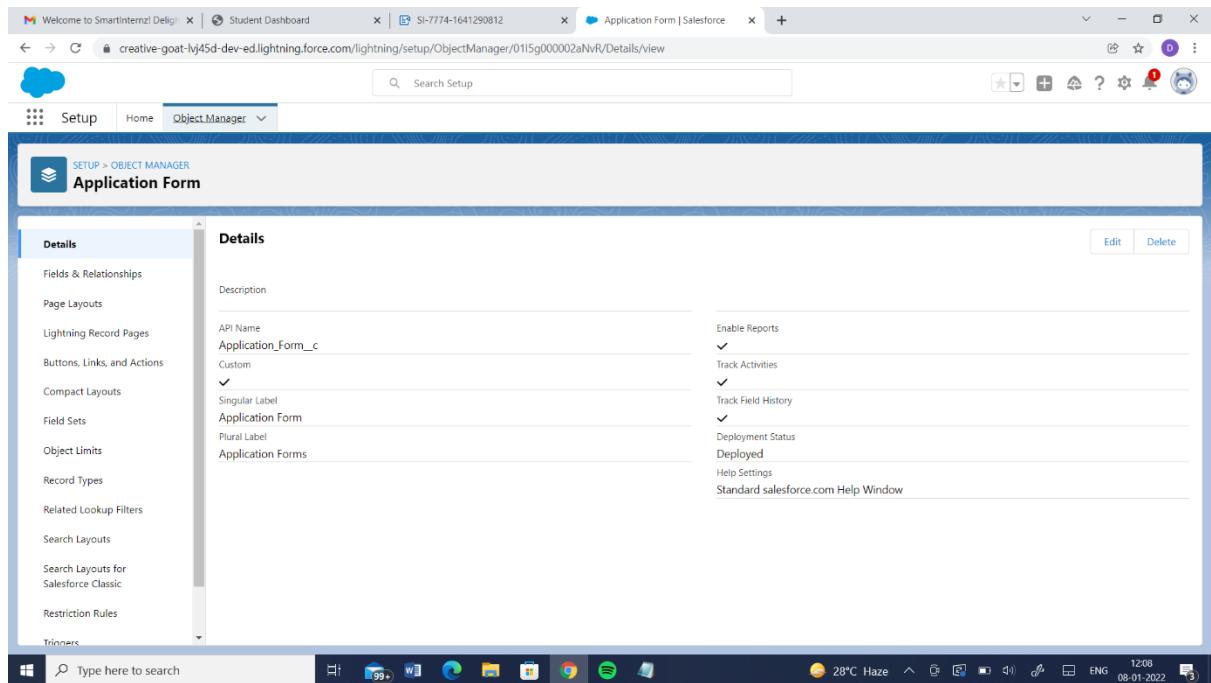
College Object:



The screenshot shows the Salesforce Setup interface for the 'College' object. The left sidebar lists various configuration options: 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, Restriction Rules, and Triggers. The main 'Details' tab is selected, displaying the following information:

Setting	Value
Description	
API Name	College__c
Custom	✓
Singular Label	College
Plural Label	Colleges
Enable Reports	✓
Track Activities	✓
Track Field History	✓
Deployment Status	Deployed
Help Settings	Standard salesforce.com Help Window

Application Form Object:



The screenshot shows the Salesforce Setup interface for the 'Application Form' object. The left sidebar lists various configuration options: 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, Restriction Rules, and Triggers. The main 'Details' tab is selected, displaying the following information:

Setting	Value
Description	
API Name	Application_Form__c
Custom	✓
Singular Label	Application Form
Plural Label	Application Forms
Enable Reports	✓
Track Activities	✓
Track Field History	✓
Deployment Status	Deployed
Help Settings	Standard salesforce.com Help Window

Student Object:

The screenshot shows the Salesforce Object Manager interface for the 'Students' object. The left sidebar lists various configuration options: 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 panel displays the 'Details' section for the 'Students' object. It includes fields for Description, API Name (Students__c), Singular Label (Students), Plural Label (Students), and several checkboxes for reporting and tracking features like Enable Reports, Track Activities, Track Field History, Deployment Status (Deployed), Help Settings, and Standard salesforce.com Help Window. At the bottom right of the main panel are 'Edit' and 'Delete' buttons.

Subject Object:

The screenshot shows the Salesforce Object Manager interface for the 'Subject' object. The left sidebar lists the same configuration options as the Student object. The main panel displays the 'Details' section for the 'Subject' object. It includes fields for Description, API Name (Subject__c), Singular Label (Subject), Plural Label (Subjects), and the same set of reporting and tracking checkboxes as the Student object. At the bottom right of the main panel are 'Edit' and 'Delete' buttons.

Topic: **Custom Object Creation**

- Milestone / Activities:**
1. Create Fields On College Object
 2. Create Fields On Application Form Object
 3. Create Fields On Subject Object
 4. Create Fields On Student Object

Detailed Description:**1. Create Fields On College Object**

Click the gear icon. and select Setup. This launches Setup in a new tab. Click the Object Manager tab. From the list of objects in the dropdown, click Suggestion. Click the Fields & Relationships section. Click New.

Creating the following fields:

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

2. Create Fields On Application Form Object

Click the gear icon. and select Setup. This launches Setup in a new tab. Click the Object Manager tab. From the list of objects in the dropdown, click Suggestion. Click the Fields & Relationships section. Click New.

Creating the following fields:

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	

3. Create Fields On Student Object

Click the gear icon. and select Setup. This launches Setup in a new tab. Click the Object Manager tab. From the list of objects in the dropdown, click Suggestion. Click the Fields & Relationships section. Click New.

Creating the following fields:

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	

4. Create Fields On Subject Object

Click the gear icon. and select Setup. This launches Setup in a new tab. Click the Object Manager tab. From the list of objects in the dropdown, click Suggestion. Click the Fields & Relationships section. Click New.

Creating the following fields:

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)		

OUTCOME:

College Object:

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

- Setup Tab:** The user is currently in the Setup tab.
- Object Manager:** Under the Object Manager section, the "College" object is selected.
- Fields & Relationships:** The main table displays the following fields:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Capacity Of Students	Capacity__Of_Students__c	Picklist	College Name	
College Fees	College_Fees__c	Currency(6, 2)		
College Name	College_Name__c	Picklist		
College Name	Name	Text(80)		
Created By	CreatedById	Lookup(User)		
Email	Email__c	Picklist	College Name	
Hostel Fees	Hostel_Fees__c	Currency(6, 2)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User.Group)		
Record Info	Record_Info__c	Text(20)		

Application Form Object:

The screenshot shows the Salesforce setup interface for the Application Form object. The left sidebar includes options like 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, Restriction Rules, and Triggers. The main content area is titled 'Fields & Relationships' and lists 15 items. The fields are:

Created By	CreatedById	Lookup(User)
date of Birth	date_of_Birth__c	Date
Email	Email__c	Email (Unique)
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
Phone	Phone__c	Phone
Ready To Join	Ready_To_Join__c	Checkbox
Student Name	Student_Name__c	Text(30)

The bottom of the screen shows a taskbar with various icons and system status information.

Student Object:

The screenshot shows the Salesforce setup interface for the Student object. The left sidebar includes options like 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, Restriction Rules, and Triggers. The main content area is titled 'Fields & Relationships' and lists 10 items. The fields are:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Address	Address__c	Text(255)		
Application Form	Application_Form__c	Lookup(Application Form)		
College Name	College_Name__c	Formula (Text)		
Created By	CreatedById	Lookup(User)		
Date Of Birth	DateOfBirth__c	Date		
Guardian Name	Guardian_Name__c	Text(30)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		
Phone	Phone__c	Phone		
Student Name	Name	Text(80)		

The bottom of the screen shows a taskbar with various icons and system status information.

Subject Object:

The screenshot shows the Salesforce Setup interface for the 'Subject' object. The left sidebar lists various setup categories like Page Layouts, Lightning Record Pages, and Field Sets. The main content area is titled 'Fields & Relationships' and displays a table of fields. The table has columns for FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The indexed column contains checkmarks for most fields.

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

Topic:

Adding Business Logic To Application

Milestone / Activities: 1. Creating Global Picklist Value Set

2. Creating Field Dependencies

3. Creating Validation Rules

4. Process Automation

1. Creating Global Picklist Value Sets:

Detailed Description:

A.) College

From Setup, enter Picklist in the Quick Find box, then select Picklist Value Sets. Next to Global Value Sets, click New. Enter College as label for the global value set. This name appears in Setup, and when users create a picklist based on this global value set. To tell users what these values are for, can give specific description of the global value set. Next we enter the values, one per line.

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

Optionally choose to sort the values alphabetically or to use the first value in the list as the default value, or both. If you select both options, Click Save.

b.) Paper 1

From Setup, enter Picklist in the Quick Find box, then select Picklist Value Sets. Next to Global Value Sets, click New. Enter Paper 1 as label for the global value set. This name appears in Setup, and when users create a picklist based on this global value set. To tell users what these values are for, can give specific description of the global value set. Next we enter the values, one per line.

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

Optionally choose to sort the values alphabetically or to use the first value in the list as the default value, or both. If you select both options, Click Save.

c.) Paper 2

From Setup, enter Picklist in the Quick Find box, then select Picklist Value Sets. Next to Global Value Sets, click New. Enter Paper 2 as label for the global value set. This name appears in Setup, and when users create a picklist based on this global value set. To tell users what these values are for, can give specific description of the global value set. Next we enter the values, one per line.

Picklist Value Name	Values
Paper 2	MATHEMATICS ENGLISH STATISTICS

Optionally choose to sort the values alphabetically or to use the first value in the list as the default value, or both. If you select both options, Click Save.

Outcome:

List of Global Picklist Value Sets created:

The screenshot shows the Salesforce Setup interface with the 'Picklist Value Sets' page open. The left sidebar shows various setup categories like Service Setup Assistant, Release Updates, and Administration. The main area displays a table of Global Value Sets with columns for Action, Label, and Description. One entry is visible: 'College' with a description of 'College Name'. A 'New' button is at the top right of the table.

College :

The screenshot shows the details of a Global Value Set named 'College'. The 'Global Value Set Detail' section shows the label 'College Name' and the name 'College_Name'. The 'Picklist Values Used' section indicates there are 6 active/inactive values. The 'Values' section lists the following data:

Action	Values	API Name	Default	Chart Colors	Modified By
Edit Del Deactivate	MIT-HYD	MIT-HYD	<input type="checkbox"/>	Assigned dynamically	Divya Kanchi, 1/7/2022, 11:51 PM
Edit Del Deactivate	MIT-BLR	MIT-BLR	<input type="checkbox"/>	Assigned dynamically	Divya Kanchi, 1/7/2022, 11:51 PM
Edit Del Deactivate	MIT-MUM	MIT-MUM	<input type="checkbox"/>	Assigned dynamically	Divya Kanchi, 1/7/2022, 11:51 PM
Edit Del Deactivate	MIT-MAA	MIT-MAA	<input type="checkbox"/>	Assigned dynamically	Divya Kanchi, 1/7/2022, 11:51 PM
Edit Del Deactivate	MIT-DEL	MIT-DEL	<input type="checkbox"/>	Assigned dynamically	Divya Kanchi, 1/7/2022, 11:51 PM
Edit Del Deactivate	MIT-CCU	MIT-CCU	<input type="checkbox"/>	Assigned dynamically	Divya Kanchi, 1/7/2022, 11:51 PM

Paper 1:

The screenshot shows the Salesforce Setup interface. The left sidebar includes links for Setup Home, Service Setup Assistant, Multi-Factor Authentication Assistant, Release Updates, Lightning Experience Transition Assistant, New Salesforce Mobile App QuickStart, Lighting Usage, Optimizer, Administration (Users, Data, Email), Platform Tools (Apps, Feature Settings, Workflow Services, Einstein), and Objects and Fields. The main content area displays the 'Picklist Value Sets' page under 'Global Value Set'. A new value set is being created with the following details:

Label	Name	Description
Paper_1	Paper_1	

The 'Values' section lists four active picklist values:

Action	Values	API Name	Default	Chart Colors	Modified By
Edit Del Deactivate	APEX	APEX	<input type="checkbox"/>	Assigned dynamically	Davya Kanchi 1/7/2022, 11:55 PM
Edit Del Deactivate	JAVA	JAVA	<input type="checkbox"/>	Assigned dynamically	Davya Kanchi 1/7/2022, 11:55 PM
Edit Del Deactivate	C	C	<input type="checkbox"/>	Assigned dynamically	Davya Kanchi 1/7/2022, 11:55 PM
Edit Del Deactivate	C++	C++	<input type="checkbox"/>	Assigned dynamically	Davya Kanchi 1/7/2022, 11:55 PM

The 'Inactive Values' section is empty.

Paper 2:

The screenshot shows the Salesforce Setup interface, identical to Paper 1, but with a different Global Value Set configuration. A new value set is being created with the following details:

Label	Name	Description
Paper_2	Paper_2	

The 'Values' section lists three active picklist values:

Action	Values	API Name	Default	Chart Colors	Modified By
Edit Del Deactivate	MATHEMATICS	MATHEMATICS	<input type="checkbox"/>	Assigned dynamically	Davya Kanchi 1/7/2022, 11:57 PM
Edit Del Deactivate	ENGLISH	ENGLISH	<input type="checkbox"/>	Assigned dynamically	Davya Kanchi 1/7/2022, 11:57 PM
Edit Del Deactivate	STATISTICS	STATISTICS	<input type="checkbox"/>	Assigned dynamically	Davya Kanchi 1/7/2022, 11:57 PM

The 'Inactive Values' section states: "No Inactive Values values defined."

2. Creating Field Dependencies

A.) Create 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.

Detailed Description:

In the College object in the Object Manager, select Fields & Relationships. Click Field Dependencies. Followed by Click New. Then we select the **College Name** as the Controlling Field and **Email** as the Dependent Field. Click Continue. Select the appropriate Email in each column by double-clicking them.

College Name	Email
MIT-HYD	hyd@mit.co.in
MIT-BLR	blr@mit.co.in
MIT-MUM	mum@mit.co.in
MIT-MAA	maa@mit.co.in
MIT-DEL	del@mit.co.in
MIT-CCU	ccu@mit.co.in

Click Preview, then test the dependency by selecting different colleges and viewing the different emails available for each college. Click Close to close the preview window. Click Save.

B.) 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.

Detailed Description:

In the College object in the Object Manager, select Fields & Relationships. Click Field Dependencies. Followed by Click New. Then we select the **College Name** as the Controlling Field and **Capacity of Students** as the Dependent Field. Click Continue. Select the appropriate Capacity of Students in each column by double-clicking them.

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

Click Preview, then test the dependency by selecting different colleges and viewing the range of capacity available for each college. Click Close to close the preview window. Click Save.

Outcome :

Field Dependences between College Name and Email:

The screenshot shows the Salesforce setup interface for creating a field dependency. The left sidebar is titled 'Object Manager' and lists various object types like 'College'. The main area is titled 'Fields & Relationships' under 'College'. A modal window is open with the following details:

- 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)
- Preview Area:** Shows a grid of college names and their corresponding email addresses. The grid has columns for 'College Name' and 'Email'. The 'Email' column for each college has several rows of email addresses. The first row for each college is highlighted in yellow ('Included Value').
- Buttons:** Save, Cancel, Preview

Field Dependencies between College Name and Capacity of Students:

Edit Field Dependency

Controlling Field: College Name
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 (Yellow), Included Value (Blue)

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

College Field Dependencies

This page allows you to define dependencies between fields (e.g., dependent picklists).

Action	Controlling Field	Dependent Field	Modified By
Edit Del	College Name	Email	Divya.Kanchi, 1/8/2022, 4:53 AM
Edit Del	College Name	Capacity Of Students	Divya.Kanchi, 1/8/2022, 4:57 AM

3. Creating Validation Rules

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

Detailed Description:

In the College Object, click Validation Rules that is located on the left sidebar. Click New.

Enter the following properties for your validation rule:

- a. Rule Name: College_Name_Validation
- b. Error Condition Formula:

TEXT(College_Name__c) <> Name

- c. Error Message: College name and Record info must be same.

To check your formula for errors, click Check Syntax. Click Save to finish.

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

Detailed Description:

In the Application From Object, click Validation Rules that is located on the left sidebar. Click New.

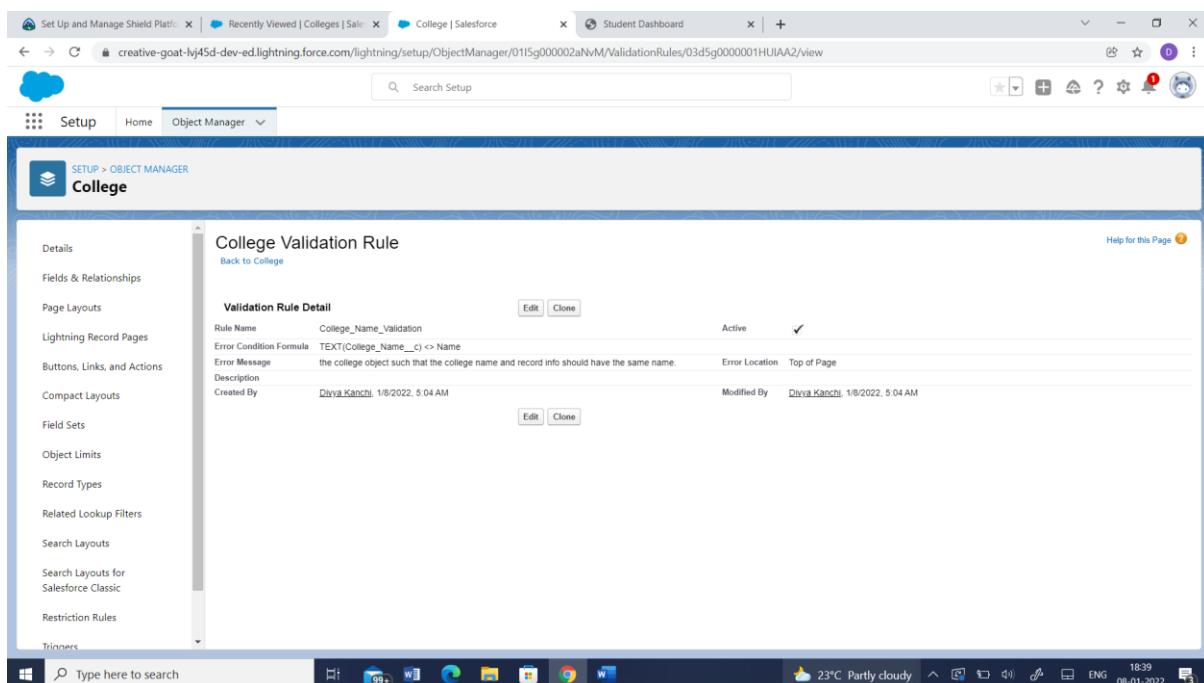
Enter the following properties for your validation rule:

- a. Rule Name: Application_Form_Validation
- b. Error Condition Formula:

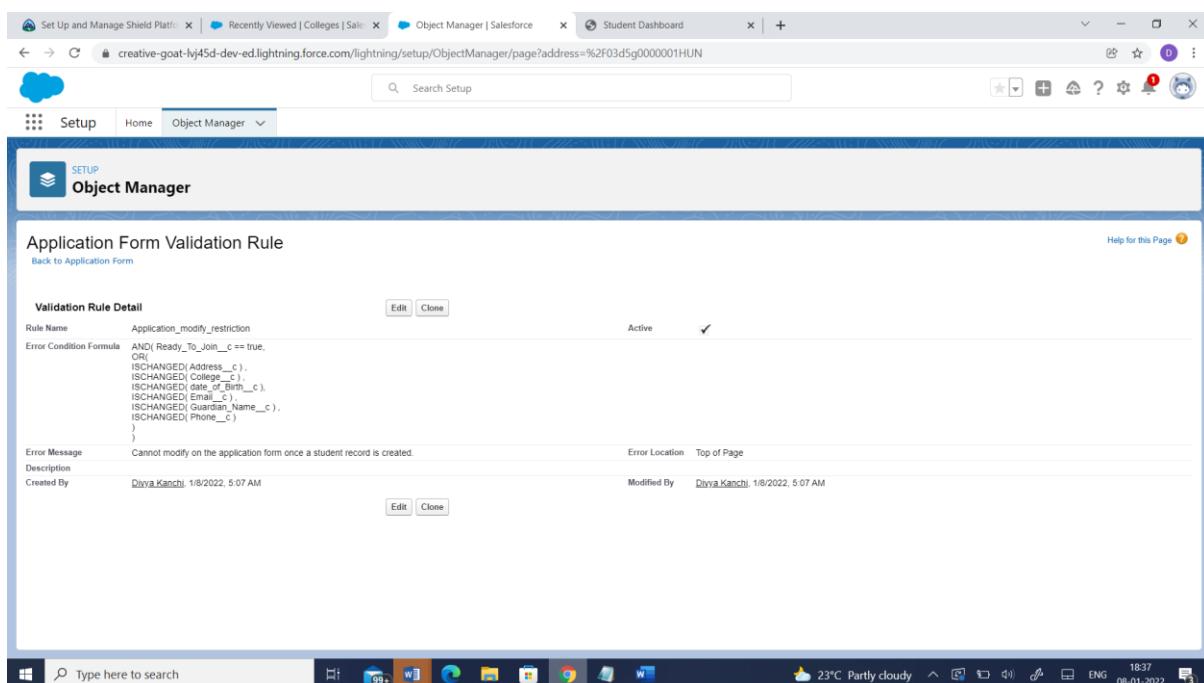
AND(Ready_To_Join__c == true ,
OR
ISCHANGED(Address__c) ,
ISCHANGED(College__c) ,
ISCHANGED(Date_Of_Birth__c) ,
ISCHANGED(Email__c) ,
ISCHANGED(Guardian_Name__c) ,
ISCHANGED(Phone__c)
)
)

To check your formula for errors, click Check Syntax. Click Save to finish.

Outcome:



The screenshot shows the Salesforce Setup interface for the 'College' object. On the left, a sidebar lists various configuration options like Details, Fields & Relationships, Page Layouts, and Triggers. The main content area displays the 'College Validation Rule' detail page. The rule is named 'College_Name_Validation' and is active. The error condition formula is 'TEXT(College_Name__c) <> Name'. The error message is 'the college object such that the college name and record info should have the same name.' The rule was created by 'Divya_Kanchi' on 1/8/2022 at 5:04 AM and modified by the same user on the same date and time.



The screenshot shows the Salesforce Setup interface for the 'Object Manager' section. The main content area displays the 'Application Form Validation Rule' detail page. The rule is named 'Application_modify_restriction' and is active. The error condition formula is 'AND(Ready_To_Join__c == true, ISCHANGED(Address__c), ISCHANGED(College__c), ISCHANGED(date_of_Birth__c), ISCHANGED(first_name__c), ISCHANGED(Guardian_Name__c), ISCHANGED(Phone__c))'. The error message is 'Cannot modify on the application form once a student record is created.' The rule was created by 'Divya_Kanchi' on 1/8/2022 at 5:07 AM and modified by the same user on the same date and time.

4. Process Automation

A.) Create 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.

Detailed Description:

Initially 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”. And select create a record on the student object.

OutCome:

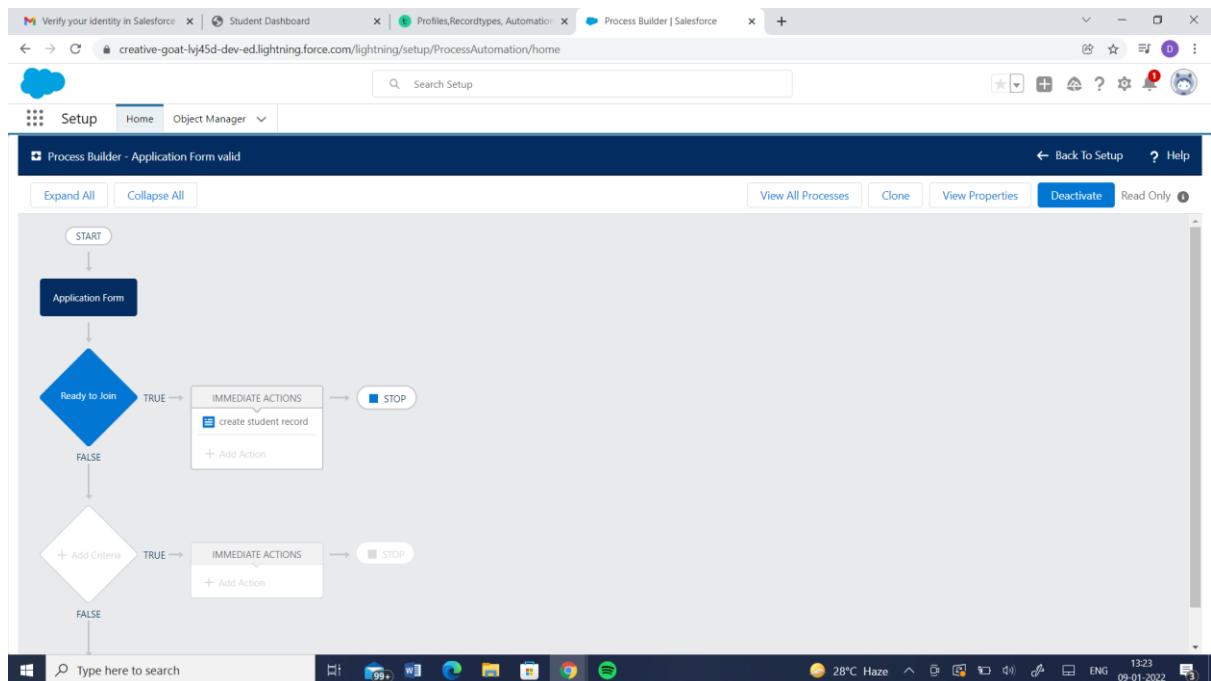
New Process : Application Form Validation

The screenshot shows the Salesforce Process Builder interface. The browser tab is titled 'Process Builder | Salesforce'. The main page displays a table titled 'My Processes' with one entry:

PROCESS	DESCRIPTION	OBJECT	PROCESS TYPE	LAST MODIFIED	STATUS	ACTIONS
> Application Form valid		Application Form	Record Change	1/8/2022	Active	<button>New</button>

The status bar at the bottom shows system information: 27°C Haze, ENG, 1358, 09-01-2022, and a battery icon.

Process flow:



New Application Form Creation:

Student Details

- Student Name: Hans vihan
- Address: Visakhapatnam
- Application Form: TA-00001
- Date Of Birth: 1/25/2000
- Guardian Name: Hans
- Phone: (908) 790-8710
- College Name: MIT-MUM

Last Modified By
Divya Kanchi, 1/9/2022, 12:18 AM

Activity

Upcoming & Overdue

No next steps.
To get things moving, add a task or set up a meeting.

No past activity. Past meetings and tasks marked as done show up here.

Student Record Created when Ready to join value is true:

The screenshot shows a Salesforce Lightning interface for a student record. The top navigation bar includes tabs for 'Verify your identity in Salesforce', 'Student Dashboard', 'Profiles, Recordtypes, Automation', 'College | Salesforce', and 'TA-00001 | Salesforce'. The main page header displays 'creative-goat-lvj45d-dev-ed.lightning.force.com/lightning/r/Application_Form_c/a025g000006VWlAAW/view'. The page title is 'Application Form TA-00001'. The left sidebar has sections for 'Related' (Notes & Attachments, Students), 'Details' (with a note about Hans vihan), and 'Activity' (New Event, New Task, Log a Call, Email). The bottom of the screen shows the Windows taskbar with various pinned icons and system status.

Topic: **Batch Apex**

- Milestone / Activities:**
- 1) From the developer console create a new apex class and enter the following code.
 - 2) From the developer console create a new apex class and enter the following code.

Detailed Description:

Batch Apex is used to run large jobs (think thousands or millions of records!) that would exceed normal processing limits. Using Batch Apex, you can process records asynchronously in batches (hence the name, “Batch Apex”) to stay within platform limits. If you have a lot of records to process, for example, data cleansing or archiving, Batch Apex is probably your best solution.

Syntax of BatchApex:

```
public class MyBatchClass implements Database.Batchable<sObject> {  
    public (Database.QueryLocator | Iterable<sObject>) start(Database.BatchableContext bc) {  
        // collect the batches of records or objects to be passed to execute  
    }  
    public void execute(Database.BatchableContext bc, List<P> records){  
        // process each batch of records  
    }  
    public void finish(Database.BatchableContext bc){  
        // execute any post-processing operations  
    }  
}
```

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

Program Implementation:

```
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 } );
    }
}
```

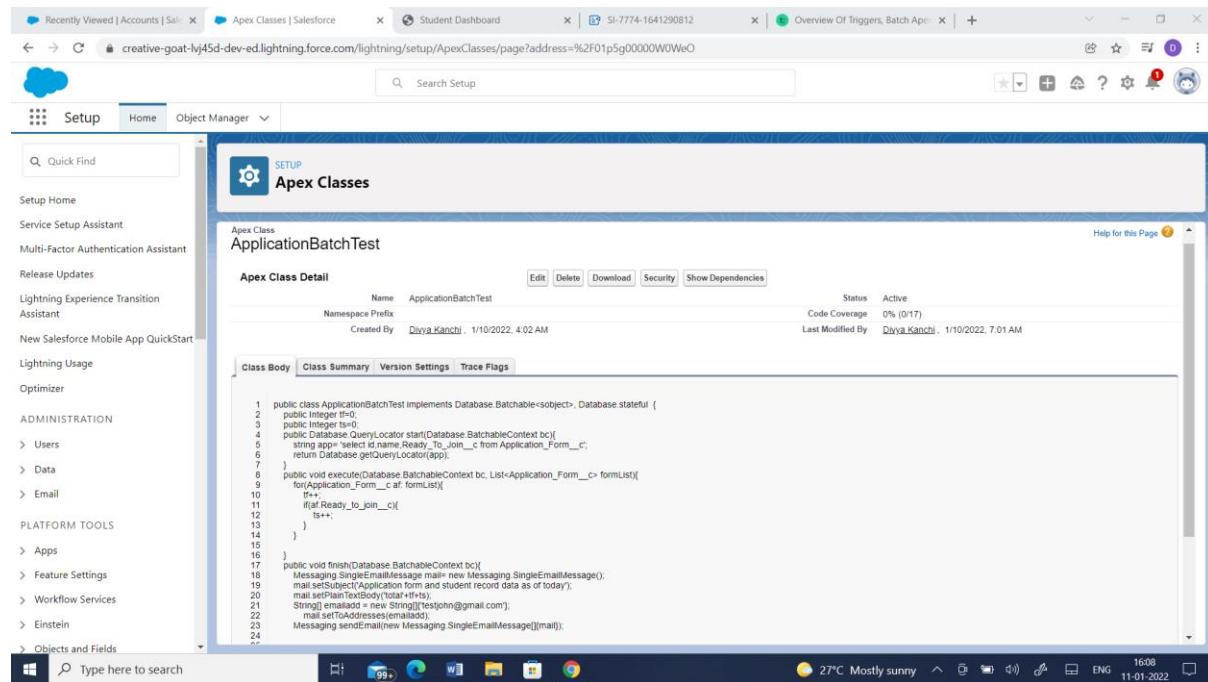
2) From the developer console create a new apex class and enter the following code.

Program Implementation:

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

After program implementation, from apex class we need to schedule job and fill the details as per our requirements.

Outcome:



The screenshot shows the Salesforce Setup interface with the 'Apex Classes' tab selected. The main content area displays the details for the Apex Class 'applicationschedule'. The class is named 'applicationschedule', has a namespace prefix, and was created by 'Divya Kanthi' on 1/11/2022 at 1:37 AM. It is active and has 0% code coverage. The 'Class Body' tab is selected, showing the following Apex code:

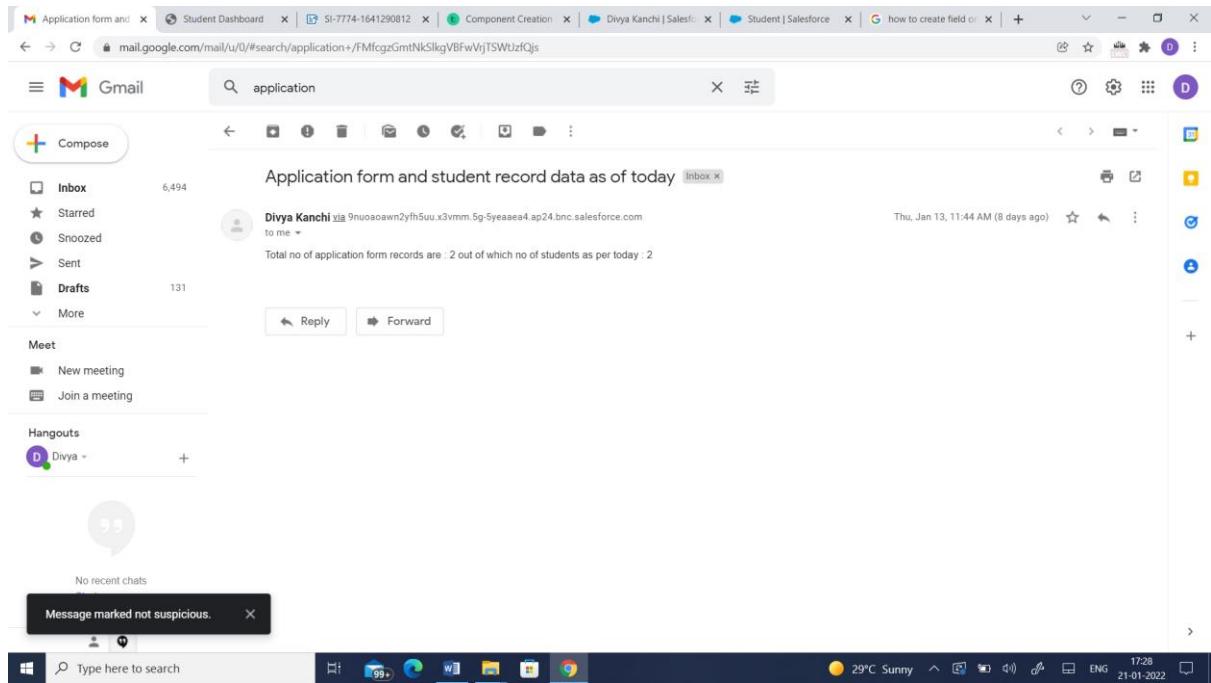
```
1 public class applicationschedule implements Schedulable{  
2       
3       
4     public void execute(SchedulableContext sc){  
5         ApplicationBatchTest abt = new ApplicationBatchTest();  
6         Database.executeBatch(abt, 400); // 200 to 2000  
7     }  
8       
9 }  
10   
11   
12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

Below the code, there are edit, delete, download, security, and show dependencies buttons.

The screenshot shows the Salesforce Setup interface with the 'Apex Jobs' tab selected. The main content area displays the 'Apex Jobs' page, which includes a heading, a link to the new batch jobs page, and a table listing scheduled Apex jobs. The table has columns for Action, Submitted Date, Job Type, Status, Status Detail, Total Batches, Batches Processed, Failures, Submitted By, Completion Date, Apex Class, Apex Method, and Apex Job ID. Two entries are listed:

Action	Submitted Date	Job Type	Status	Status Detail	Total Batches	Batches Processed	Failures	Submitted By	Completion Date	Apex Class	Apex Method	Apex Job ID
1/11/2022, 1:48 AM	Batch Apex	Completed	1	1	Kanchi_Divya	1/11/2022, 1:48 AM	ApplicationBatchTest	7075g00001j7Bom				
1/11/2022, 1:39 AM	Scheduled Apex	Queued	0	0	Kanchi_Divya	applicationschedule	7075g00001j7AWZ					

On the left sidebar, under the 'Email' section, the 'Apex Jobs' item is highlighted. At the bottom of the page, there is a search bar and a note about using Global Search.



Topic:

Lightning Web Components

Milestone / Activities:

- 1) Create College DataTable Component(APEX CLASS)
- 2) Create College DataTable Component (HTML FILE)
- 3) Create College DataTable Component(JAVA SCRIPT FILE)
- 4) Create College DataTable Component(META FILE)

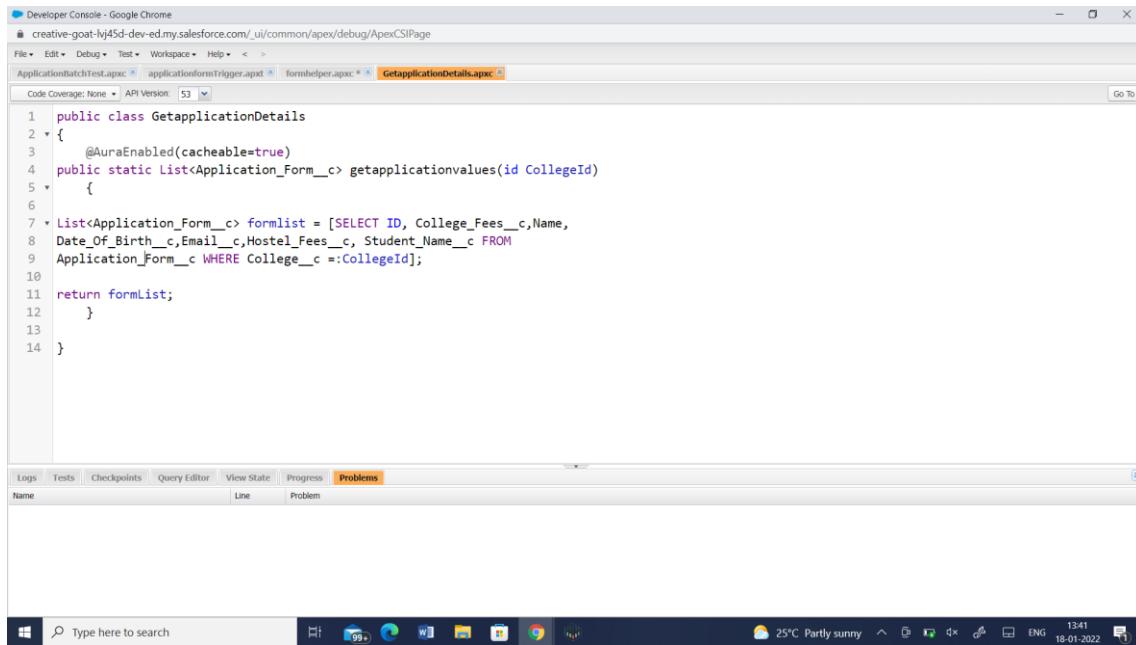
Detailed Description:

- Improved performance of the component as most of the code is recognized by the native web browser engine and web stack
- Ability to compose applications using smaller chunks of code since the crucial elements that are required to create a component is part of the native web browser engine and web stack
- Increase in the robustness of the applications built using LWCs as they are inclusive of the said modern web standards.
- Parallel interoperability and feasibility to use both Lightning Web Components and Aura components together in the applications with no visible differentiation to the end-users

For creation of 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.

OUTCOME:

College DataTable Component(APEX CLASS)



The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is `creative-goat-lj45d-dev-ed.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage`. The tabs at the top include ApplicationBatchTest.apxc, applicationformTrigger.apxc, formhelper.apxc, and GetapplicationDetails.apxc. The code editor displays the following Apex class:

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

The bottom of the screen shows the Windows taskbar with various icons and the system tray indicating it's 25°C, Partly sunny, and the date is 18-01-2022.

College DataTable Component(HTML CLASS)

The screenshot shows the Salesforce LWC Editor interface. The left sidebar lists components under 'LIGHTNING WEB COMPONENTS' including 'datatable', 'listIterator', 'lwcApex', and 'parent'. The main editor area contains the following HTML 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>
```

The 'Result Panel' at the bottom shows deployment logs:

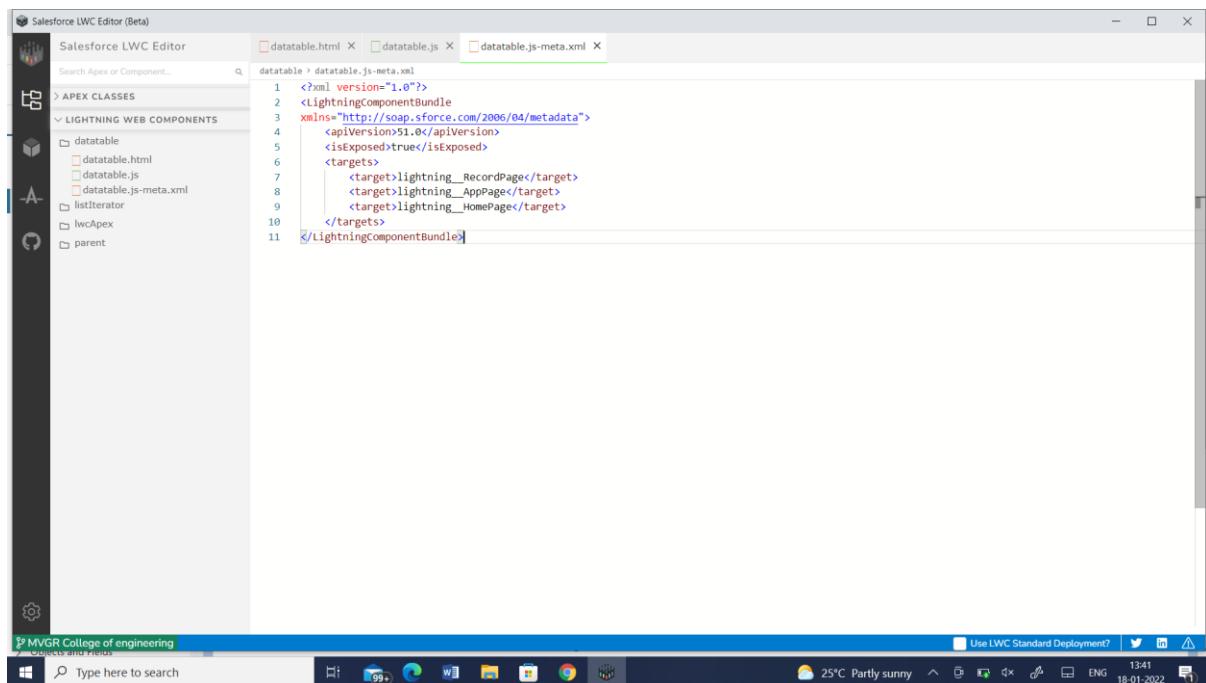
```
1 datatable.html successfully deployed source 1/18/2022 1:37:57 PM
2 datatable.js successfully deployed source 1/18/2022 1:37:57 PM
3 datatable.js-meta.xml successfully deployed source 1/18/2022 1:37:57 PM
```

College DataTable Component(JAVA SCRIPT CLASS)

The screenshot shows the Salesforce LWC Editor interface. The left sidebar lists components under 'LIGHTNING WEB COMPONENTS' including 'datatable', 'listIterator', 'lwcApex', and 'parent'. The main editor area contains the following JavaScript code:

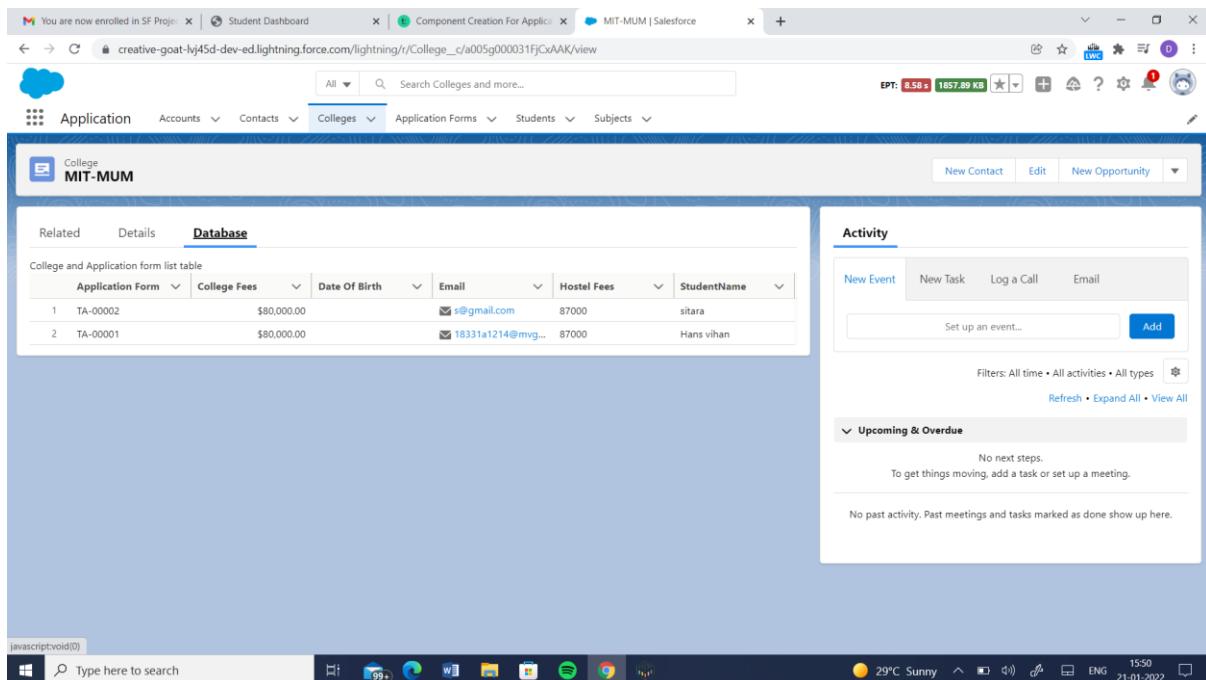
```
import { LightningElement, api, wire } from 'lwc';
import getapplicationvalues from '@salesforce/apex/GetapplicationDetails.getapplicationvalues';
export default class collegedatatable extends LightningElement {
  columnslist = [
    {label : 'Application Form' , fieldName : 'Name', type:'text' },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

College DataTable Component(META CLASS)



The screenshot shows the Salesforce LWC Editor (Beta) interface. On the left, there's a sidebar with icons for Apex Classes, Lightning Web Components, and various components like datatable, listIterator, lwcApex, and parent. The main area displays the contents of the `datatable.js-meta.xml` file:

```
1 <?xml version="1.0"?>
2 <LightningComponentBundle
3 xmlns="http://soap.sforce.com/2006/04/metadata">
4   <apiVersion>51.0</apiVersion>
5   <isExposed>true</isExposed>
6   <targets>
7     <target>lightning__RecordPage</target>
8     <target>lightning__AppPage</target>
9     <target>lightning__HomePage</target>
10   </targets>
11 </LightningComponentBundle>
```



The screenshot shows the Salesforce Lightning Experience. At the top, there are tabs for "Student Dashboard" and "Component Creation For Application". The main content area has a header "MIT-MUM | Salesforce" and a search bar "Search Colleges and more...". Below the header, there are navigation links for "Application", "Accounts", "Contacts", "Colleges", "Application Forms", "Students", and "Subjects". The main content area displays a "Database" table for "College and Application form list table". The table has columns: Application Form, College Fees, Date Of Birth, Email, Hostel Fees, and StudentName. There are two rows of data:

	Application Form	College Fees	Date Of Birth	Email	Hostel Fees	StudentName
1	TA-00002	\$80,000.00		s@gmail.com	87000	sitara
2	TA-00001	\$80,000.00		18331a1214@mvg...	87000	Hans vihan

To the right of the table, there's an "Activity" section with buttons for "New Event", "New Task", "Log a Call", and "Email". It also includes a "Set up an event..." input field and a "Filters" dropdown. Below the activity section, there's a "Upcoming & Overdue" section with a message: "No next steps. To get things moving, add a task or set up a meeting." and a note: "No past activity. Past meetings and tasks marked as done show up here."