



SALESFORCE  
PROJECTREADY



# College Management Application

## DAILY REPORT

**DAY1**

03-01-2022

## Create Your Salesforce Developer Org To Get Started

### STEP1:-

- WE NEED TO CREATE A DEVELOPER ORG THROUGH THE BELOW LINK.  
[developers.salesforce.com/](https://developers.salesforce.com/)

#### STEPS TO CREATE DEVELOPER ORG

1. Click on sign up.
2. On the sign up form, enter the following details :
  1. First name & Last name
  2. Email
  3. Role : Developer
  4. Company : College Name
  5. County : India
  6. Postal Code : pin code
  7. Username : should be a combination of your name and company

This need not be an actual email id, you can give anything in the format :  
[username@organization.com](mailto:username@organization.com)

**FINALLY**

Click on **sign up** after filling these.



**Sign up for your Salesforce Developer Edition**  
A full-featured copy of the Platform, for free

First Name\*

Last Name\*

Email\*

Role\*

Company\*

Country\*

Postal Code\*

Username\*

By registering, you confirm you have read and agree to the [Terms of Use](#) and the [Master Subscription Agreement](#) and to the storing and processing of your personal data by Salesforce as described in the [Privacy Statement](#), including use for marketing purposes.

**Sign me Up**

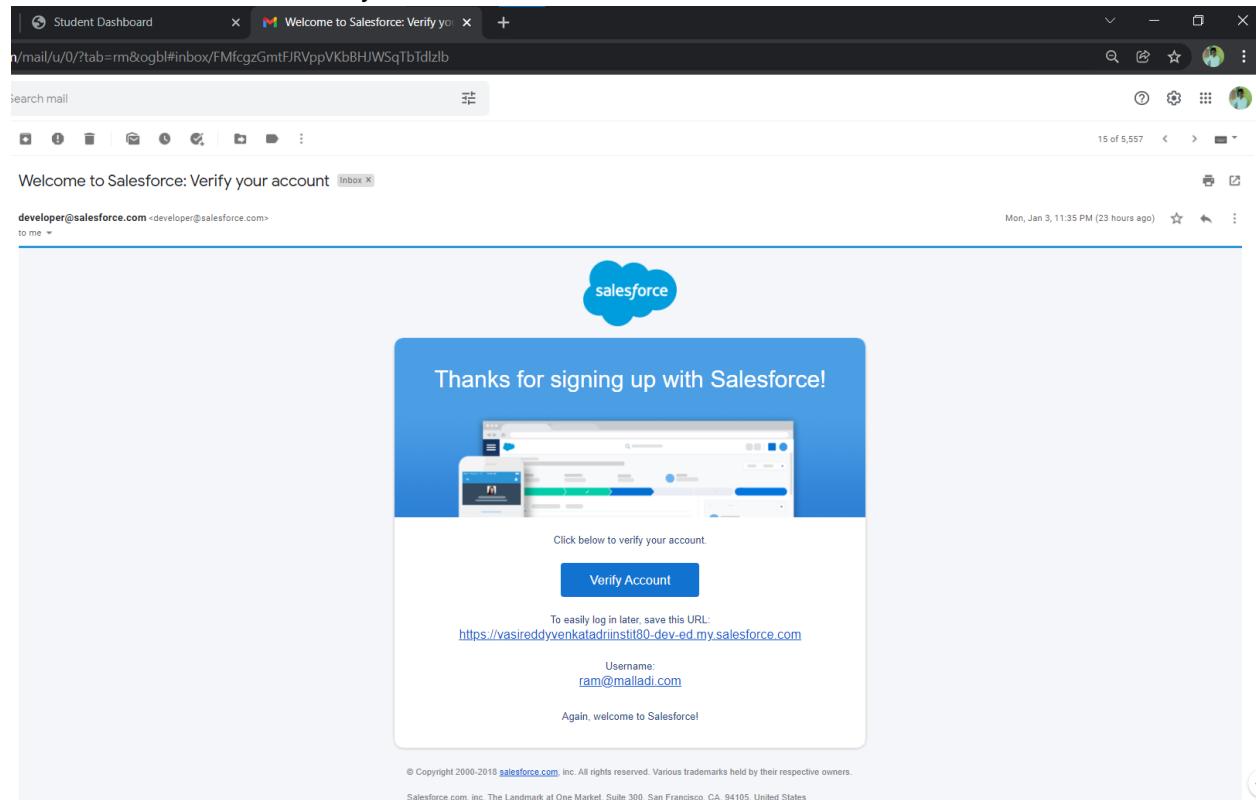
Already have a Salesforce Developer Environment?  
[Log in](#)

**STEP2:-**

- WE NEED TO ACTIVATE AND VERIFY DEVELOPER ORG WHICH WE CREATED .

## STEPS TO ACTIVATE YOUR ACCOUNT

A verification Mail is sent to your Mail ID.

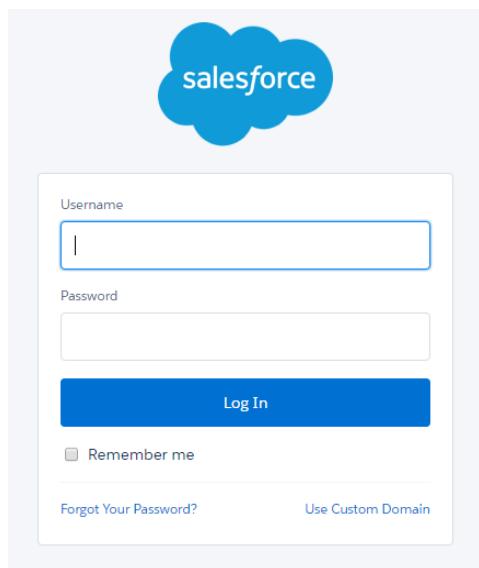


## **STEP3:-**

- LOGIN TO YOUR SALESFORCE ACCOUNT.

## STEPS TO OPEN SALESFORCE ACCOUNT

1. After activating and verifying your developer org.
2. Click on the salesforce developer login tab
3. Enter the username and password that you just created.
4. After entering username and password click login button
5. After login
6. This is the home page which you will see.



A screenshot of the Salesforce Setup Home screen. The top navigation bar shows tabs for "Setup", "Home", and "Object Manager". The main content area has a "SETUP" header and a "Home" section featuring three cards: "Get Started with Einstein Bots", "Mobile Publisher", and "Real-time Collaborative Docs". On the left, there's a sidebar with sections like "Setup Home", "Service Setup Assistant", and "ADMINISTRATION". On the right, a user profile for "Ram Malladi" is shown with options for "DISPLAY DENSITY" (set to "Comfy"), "OPTIONS", "Switch to Salesforce Classic", and "Add Username". The bottom status bar shows system information like battery level, signal strength, and date/time.

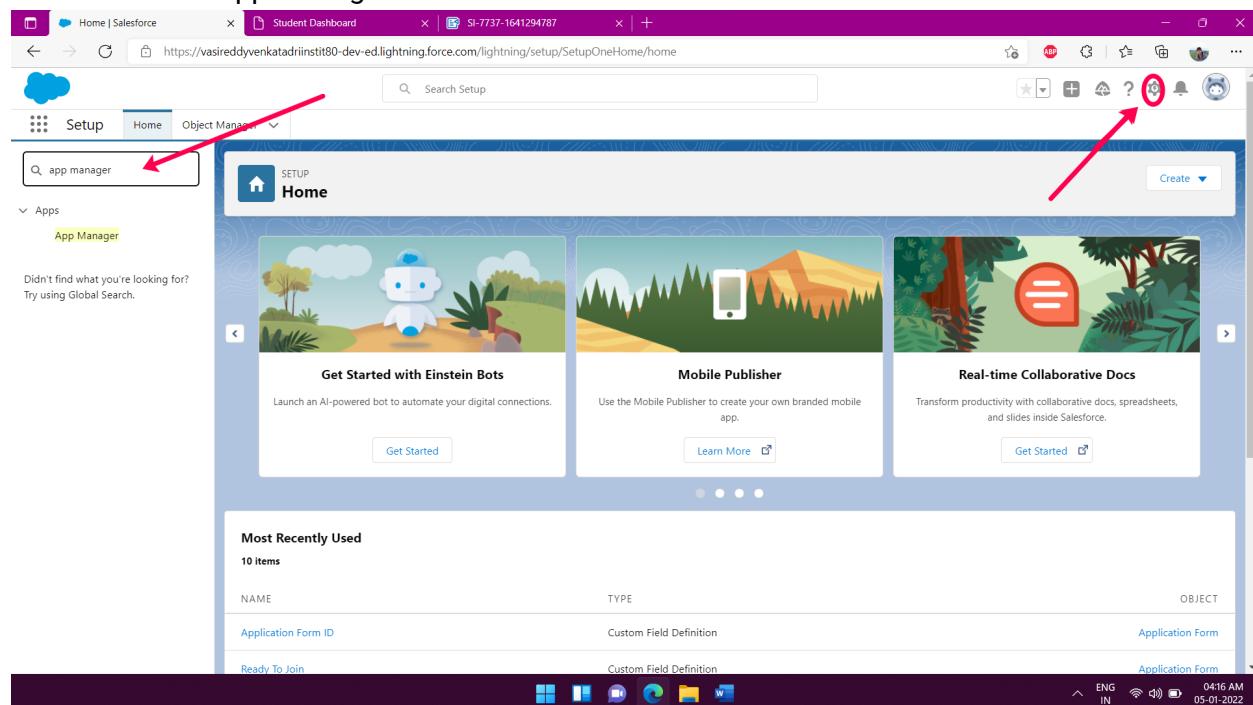
# Custom Object Creation

## STEP1:-

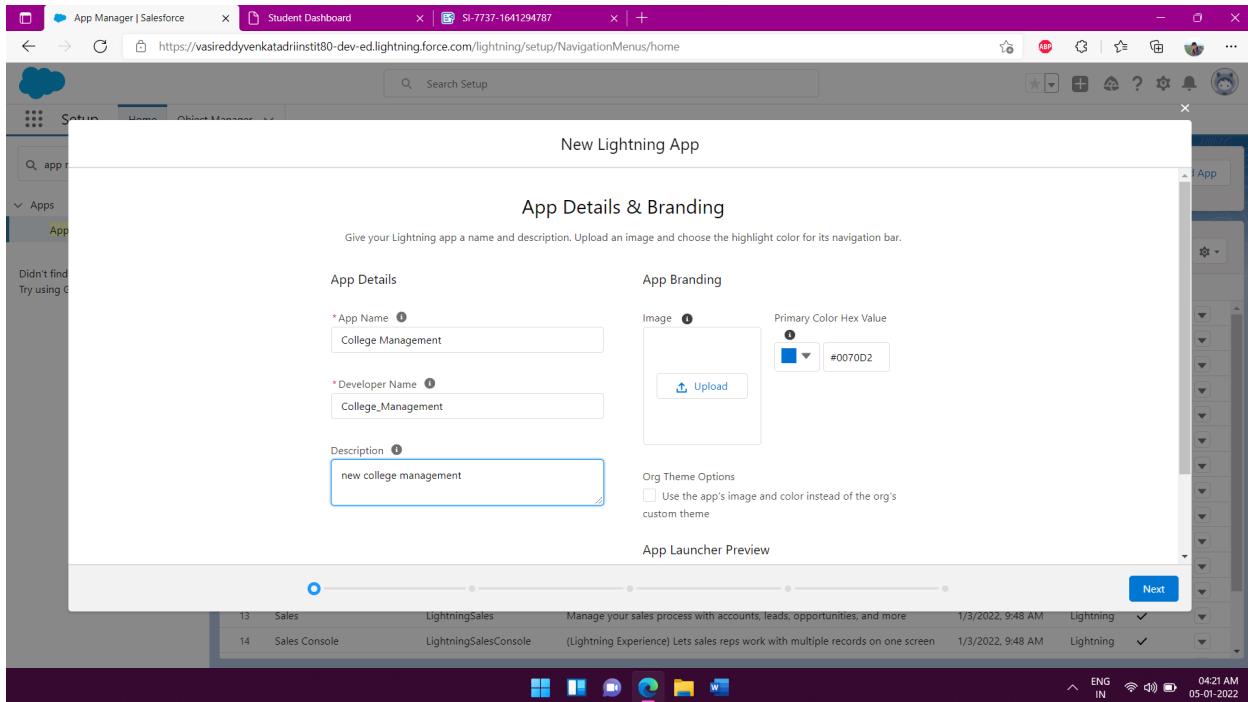
- WE NEED TO CREATE COLLEGE MANAGEMENT APP IN APP LAUNCHER

### STEPS TO CREATE APP

- Login to your salesforce developer org
- Go to the setup at the top side where appear as gear symbol
- Click on the setup menu
- In Quick Find box press App Manager
- Click on App manager



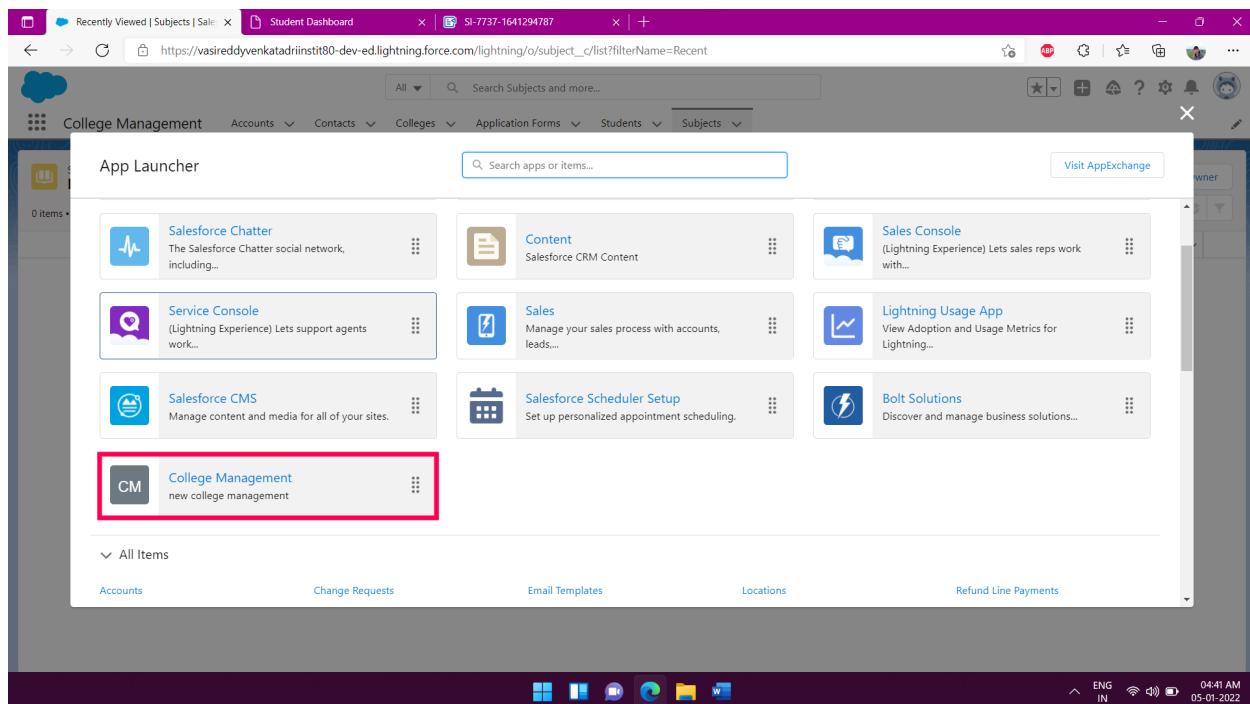
- After clicking it you will be redirected to the Lightning Experience App Manager
- Click New Lightning App



8. Fill the Columns given above
9. After filling the details click save.
10. Your new app is created

App Name ↑	Developer Name ↓	Description	Last Modified ...	App ...	Visi... ↓
All Tabs	AllTabSet	Build Tableau CRM dashboards and apps	1/3/2022, 9:48 AM	Classic	✓
Analytics Studio	Insights	Build Tableau CRM dashboards and apps	1/3/2022, 9:48 AM	Classic	✓
App Launcher	AppLauncher	App Launcher tabs	1/3/2022, 9:48 AM	Classic	✓
Bolt Solutions	LightningBolt	Discover and manage business solutions designed for your industry.	1/3/2022, 9:50 AM	Lightning	✓
College Management	College_Management	new college management	1/4/2022, 11:56 PM	Lightning	✓
Community	Community	Salesforce CRM Communities	1/3/2022, 9:48 AM	Classic	✓
Content	Content	Salesforce CRM Content	1/3/2022, 9:48 AM	Classic	✓
Lightning Usage App	LightningInstrumentation	View Adoption and Usage Metrics for Lightning Experience	1/3/2022, 9:48 AM	Lightning	✓
Marketing	Marketing	Best-in-class on-demand marketing automation	1/3/2022, 9:48 AM	Classic	✓
Platform	Platform	The fundamental Lightning Platform	1/3/2022, 9:48 AM	Classic	✓
Queue Management	QueueManagement	Create and manage queues for your business.	1/3/2022, 9:48 AM	Lightning	✓
Sales	Sales	The world's most popular sales force automation (SFA) solution	1/3/2022, 9:48 AM	Classic	✓
Sales	LightningSales	Manage your sales process with accounts, leads, opportunities, and more	1/3/2022, 9:48 AM	Lightning	✓
Sales Console	LightningSalesConsole	(Lightning Experience) Lets sales reps work with multiple records on one screen	1/3/2022, 9:48 AM	Lightning	✓
Calendar Chances	Chances	The Calendar Chances feature allows you to view and edit events and chances	1/3/2022, 9:48 AM	Classic	✓

11. To find the app you created go to the 9dots appearing at the top left corner
12. Here is the app you created.



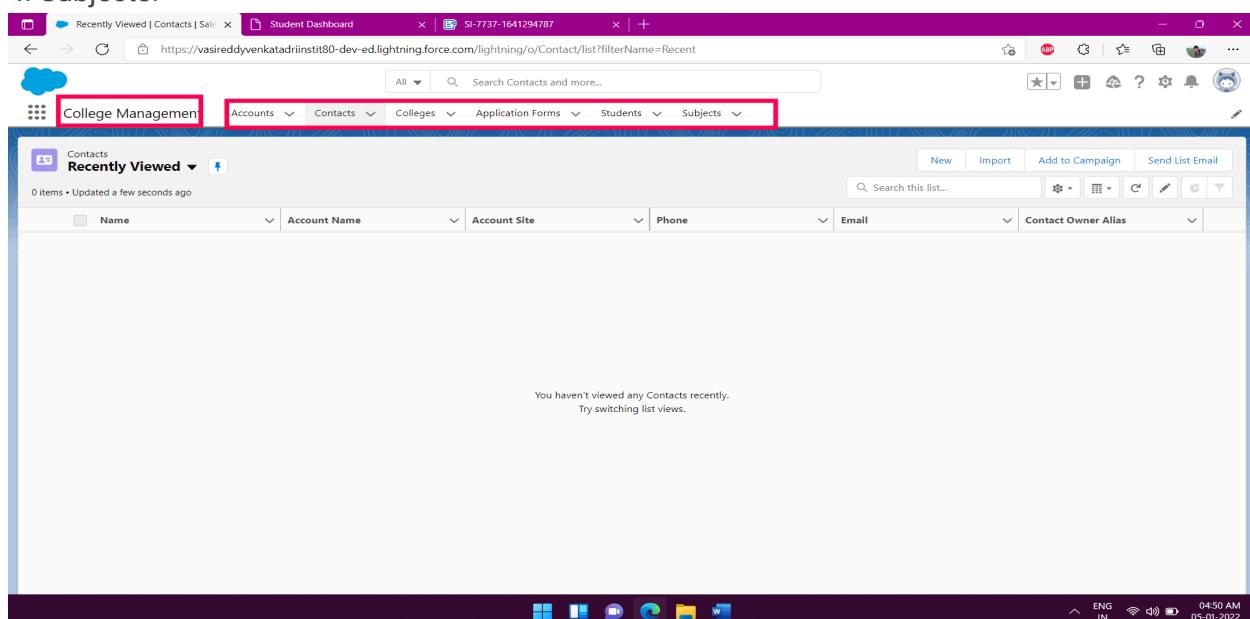
## STEP2:-

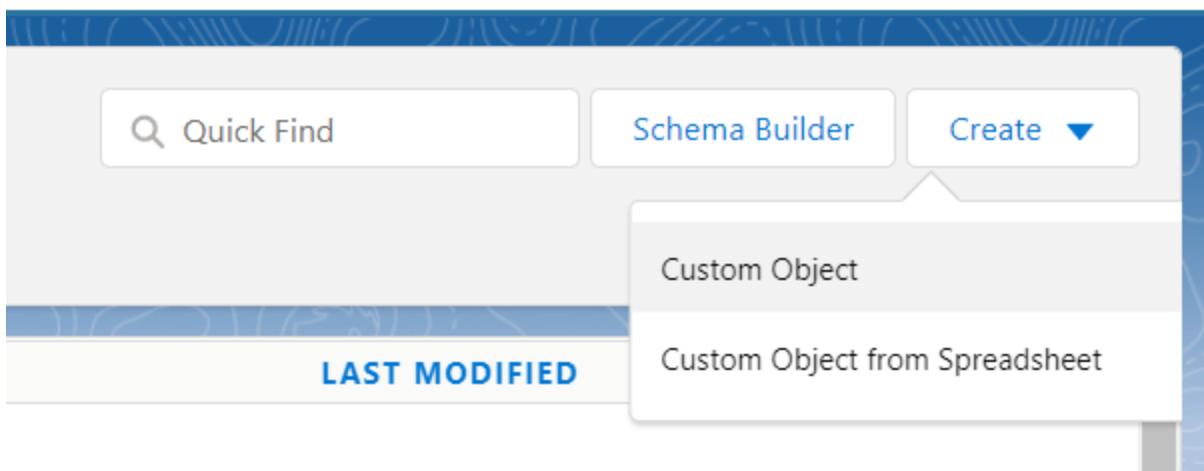
- NOW CREATE CUSTOM OBJECTS

Steps to Create the following Custom Objects for the College Management.

Custom Objects are:-

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





1. Go to the object manager
2. Click on create at the right side top of the screen
3. Click on Custom Object
4. Give the fields required
5. After giving the label

Enable the Optional features below

**Optional Features**

- Allow Reports
- Allow Activities
- Track Field History
- Allow in Chatter Groups
- Enable Licensing.

**Object Classification**

When these settings are enabled, this object is classified as an Enterprise Application object. When these settings are disabled, this object is classified as a Light Application object. [Learn more](#).

- Allow Sharing
- Allow Bulk API Access
- Allow Streaming API Access

**Deployment Status**

- In Development
- Deployed

**Search Status**

When this setting is enabled, your users can find records of this object type when they search. [Learn more](#).

- Allow Search

**Object Creation Options (Available only when custom object is first created)**

- Add Notes and Attachments related list to default page layout
- Launch New Custom Tab Wizard after saving this custom object

- i. Allow Reports
- ii. Allow Activities
- iii. Track Field History
- iv. Allow in Chatter Groups
- v. Allow Search
- vi. Add notes
- vii. Launch New Custom

### STEP3:-

- WE NEED TO CREATE FIELDS FOR THE CUSTOM OBJECTS

#### STEPS TO CREATE FIELDS ON OBJECTS

- Go to the custom Object you created
- On left side there will be Fields & Relations Column
- Click on it
- Click new to create new fields

The screenshot shows the Salesforce Setup interface for the Account object. On the left, a sidebar lists various setup options like Page Layouts, Lightning Record Pages, and Buttons, Links, and Actions. The main area is titled 'Fields & Relationships' and displays a table of existing fields. The table has columns for FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. A red arrow points to the 'New' button at the top right of the table header, which is used to create new fields.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Account Name	Name	Name		✓
Account Number	AccountNumber	Text(40)		
Account Owner	OwnerId	Lookup(User)		✓
Account Site	Site	Text(80)		
Account Source	AccountSource	Picklist		
Active	Active__c	Picklist		
Annual Revenue	AnnualRevenue	Currency(18, 0)		
Billing Address	BillingAddress	Address		
Clean Status	CleanStatus	Picklist		✓
Created By	CreatedBy	Lookup(User)		
Customer Priority	CustomerPriority__c	Picklist		

After clicking new you need to select the data type and field name.

Create the following fields on the college object.

Field Name	Data Type	Required	Values
Record Info	Text		
College Fees	Currency(7,2)	Yes	
Hostel Fees	Currency(6,2)	Yes	
College Name	Picklist(Refer Business Logic In Milestones)	Yes	

Email	Picklist		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

Create the following fields on the Application Form object.

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

Create the following fields on the student object.

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

Create the following fields On the Subject object.

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

The screenshot shows the Salesforce Object Manager interface. The URL in the browser is <https://vasireddyvenkatadriinstit80-dev-ed.lightning.force.com/lightning/setup/ObjectManager/01l5j000000hYDh/Details/view>. The page title is "College". The main content area displays the "Details" tab for the College object. The API Name is listed as "College\_\_c". The "Enable Reports" checkbox is checked. Other settings shown include "Track Activities", "Track Field History", "Deployment Status" (set to "Deployed"), and "Help Settings" (set to "Standard salesforce.com Help Window"). 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.

Screenshot of the Salesforce Setup interface showing the Object Manager for the 'College' object.

The left sidebar shows the following navigation:

- 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

The main content area displays the 'Fields & Relationships' section with 10 items, sorted by Field Label. The table includes columns for FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED.

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(7, 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(250)		

Screenshot of the Salesforce Lightning interface showing the 'New College' record creation screen.

The top navigation bar shows the following tabs:

- New College | Salesforce
- subject | Salesforce
- Student Dashboard
- SI-7737-1641294787

The left sidebar shows the 'College Management' tab selected, with 'Recently Viewed' and 'College Name' listed.

The main content area displays the 'New College' form with the following fields:

- Information:
  - College Name:
  - Record Info:
  - College Fees:
  - College Name (dropdown):
    - None-
  - Email (dropdown):
    - None-
  - Capacity Of Students (dropdown):
    - None-
  - Hostel Fees:
- Buttons at the bottom: Cancel, Save & New, Save

The status bar at the bottom indicates: ENG IN 05:51 AM 05-01-2022

Screenshot of the Salesforce Setup interface showing the Object Manager for the 'Application Form' object.

The left sidebar shows the following sections:

- 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

The main 'Details' tab displays the following configuration details:

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

System status bar at the bottom right: ENG IN 05:52 AM 05-01-2022

Screenshot of the Salesforce Setup interface showing the 'Fields & Relationships' section for the 'Application Form' object.

The left sidebar shows the following sections:

- 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

The main table displays the fields and relationships:

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

System status bar at the bottom right: ENG IN 05:53 AM 05-01-2022

New Application Form | Salesforce | subject | Salesforce | Student Dashboard | SI-7737-1641294787

[https://vasireddyvenkatadriinstit80-dev-ed.lightning.force.com/lightning/o/Application\\_Form\\_\\_c/new?count=4&nooverride=1&useRecordTypeCheck=1&navi...](https://vasireddyvenkatadriinstit80-dev-ed.lightning.force.com/lightning/o/Application_Form__c/new?count=4&nooverride=1&useRecordTypeCheck=1&navi...)

College Management Accounts Contacts Colleges

Application Forms Recently Viewed 0 items • Updated a few seconds ago Application Form Name

### New Application Form

Information

\*Application Form Name

Application Form ID

\*Address

\*College  Search Colleges...

\*Date Of Birth

\*Email

\*Guardian Name

Looking For Hostel Stay

Ready To Join

\*Student Name

\*Phone

ENG IN 05:58 AM 05-01-2022

Recently Viewed | Contacts | Sales | Student | Salesforce | Student Dashboard | SI-7737-1641294787

<https://vasireddyvenkatadriinstit80-dev-ed.lightning.force.com/lightning/setup/ObjectManager/0115j000000hYkd/Details/view>

Setup Home Object Manager

### Student

SETUP > OBJECT MANAGER

**Details**

Fields & Relationships	Description	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
Page Layouts	API Name	Enable Reports
Lightning Record Pages	Student__c	✓
Buttons, Links, and Actions	Custom	Track Activities
Compact Layouts	Singular Label	✓
Field Sets	Student	Track Field History
Object Limits	Plural Label	Deployment Status
Record Types	Students	Deployed
Related Lookup Filters		Help Settings
Search Layouts		Standard salesforce.com Help Window
Search Layouts for Salesforce Classic		
Restriction Rules		

ENG IN 05:55 AM 05-01-2022

Salesforce Setup - Student | Salesforce

Recently Viewed | Contacts | Sales | Student | Salesforce | Student Dashboard | SI-7737-1641294787 | +

https://vasireddyvenkatadriinstit80-dev-ed.lightning.force.com/lightning/setup/ObjectManager/0115j000000hYkd/FieldsAndRelationships/view

Setup Home Object Manager

Search Setup

SETUP > OBJECT MANAGER

## Student

Details Fields & Relationships

Fields & Relationships

10 Items, Sorted by Field Label

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

Q, Quick Find New Deleted Fields Field Dependencies Set History Tracking

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

ENG IN 05:55 AM 05-01-2022

New Student | Salesforce | subject | Salesforce | Student Dashboard | SI-7737-1641294787 | +

https://vasireddyvenkatadriinstit80-dev-ed.lightning.force.com/lightning/o/Student\_\_c/new?count=6&nooverride=1&useRecordTypeCheck=1&navigationLoc...

Cloud

College Management Accounts Contacts Colleges Application Forms Students Subjects

All Search Students and more...

Students Recently Viewed

0 items • Updated a few seconds ago

Student Name

New Student

Information

\* Student Name: Ram Malladi  
Owner: Ram Malladi

\* Address:

\* Application Form: Search Application Forms...

\* Date Of Birth:

\* Guardian Name:

\* Phone:

Cancel Save & New Save

ENG IN 05:59 AM 05-01-2022

Recently Viewed | Contacts | Sales | subject | Salesforce | Student Dashboard | SI-7737-1641294787 | +

https://vasireddyvenkatadriinstit80-dev-ed.lightning.force.com/lightning/setup/ObjectManager/015j000000hYki/Details/view

Setup Home Object Manager

Search Setup

SETUP > OBJECT MANAGER  
**subject**

**Details**

Description

API Name: subject\_c  
Custom: ✓  
Singular Label: subject  
Plural Label: Subjects

Enable Reports: ✓  
Track Activities: ✓  
Track Field History: ✓  
Deployment Status: Deployed  
Help Settings: Standard salesforce.com Help Window

Edit Delete

https://vasireddyvenkatadriinstit80-dev-ed.lightning.force.com/one/one.app#/setup/ObjectManager/015j000000hYki/CompactLayouts/view

New subject | Salesforce | subject | Salesforce | Student Dashboard | SI-7737-1641294787 | +

https://vasireddyvenkatadriinstit80-dev-ed.lightning.force.com/lightning/o/subject\_c/new?count=7&nooverride=1&useRecordTypeCheck=1&navigationLoc...

ENG IN 05:56 AM 05-01-2022

College Management Accounts Contacts Colleges Application Forms Students Subjects

All Search Subjects and more...

Subjects Recently Viewed ▾

0 items • Updated a few seconds ago

New Import Change Owner

New subject

Information

\*subject Name:

Owner: Ram Malladi

subject ID:

Student: Search Students...

Paper1: -None--

Paper2: -None--

Cancel Save & New Save

ENG IN 06:00 AM 05-01-2022

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)		✓
Paper1	Paper1__c	Picklist		▼
Paper2	Paper2__c	Picklist		▼
Student	Student__c	Lookup(Student)		▼
subject ID	subject_ID__c	Auto Number		▼
subject Name	Name	Text(80)		▼

**DAY3**

05-01-2022

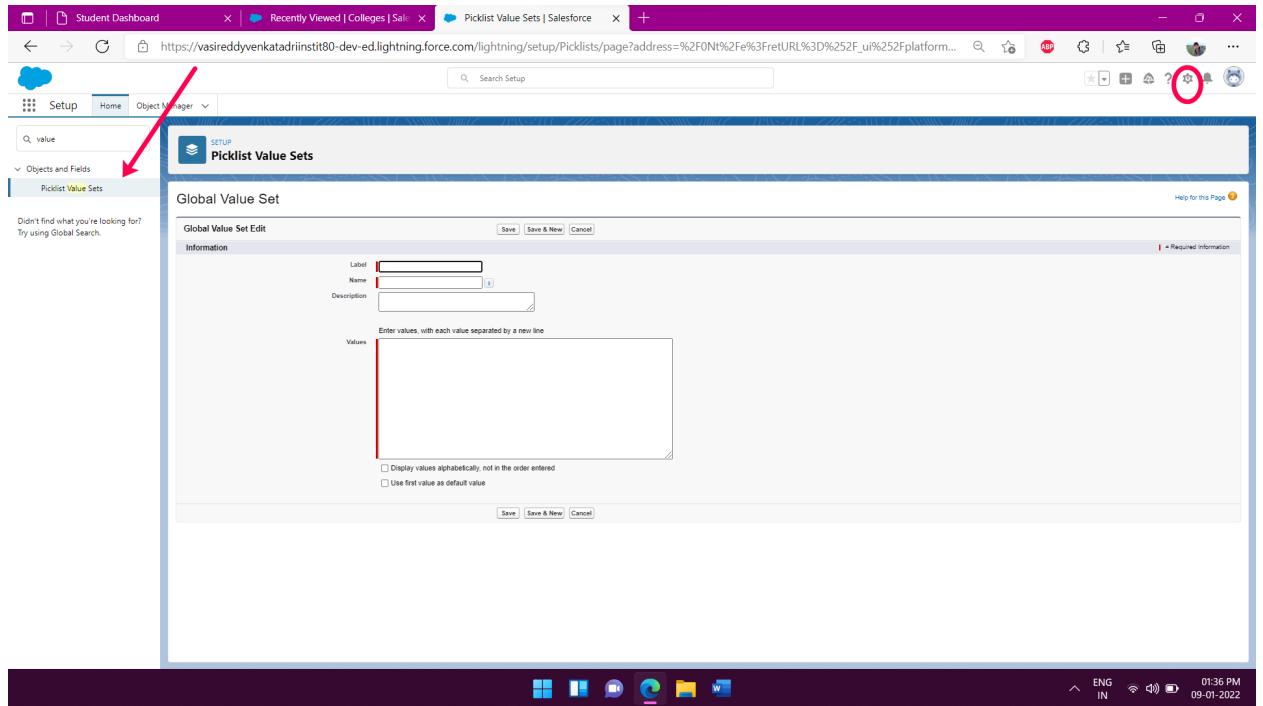
## Create Business Logic To Application

### STEP1:-

- WE NEED TO CREATE GLOBAL PICKLIST VALUE SETS

#### STEPS TO CREATE GLOBAL PICKLIST VALUE SETS

1. Go to the setup
2. Go to HOME
3. Search Picklist Values sets in Quick Find
4. Click on Picklist Values Sets



1. Create the following global picklist value sets for the application.

a)College

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

b)Paper1

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

c)Paper2

Picklist Value Name	Values
Paper2	MATHEMATICS ENGLISH STATISTICS

**Screenshot 1 (Top): Global Value Set - Paper2**

The screenshot shows the Salesforce Setup interface for creating a Global Value Set named "Paper2". The "Values" section contains three entries: MATHEMATICS, ENGLISH, and STATISTICS.

**Screenshot 2 (Bottom): Global Value Set - Paper1**

The screenshot shows the Salesforce Setup interface for creating a Global Value Set named "Paper1". The "Values" section contains four entries: APEX, JAVA, C, and C++.

**Global Value Set**

**Global Value Set Detail**

**Picklist Values Used**

Action	Values	API Name	Default	Modified By
Edit   Del   Deactivate	MATHEMATICS	MATHEMATICS	<input type="checkbox"/>	Ram Matali 1/9/2022, 12:06 AM
Edit   Del   Deactivate	ENGLISH	ENGLISH	<input type="checkbox"/>	Ram Matali 1/9/2022, 12:06 AM
Edit   Del   Deactivate	STATISTICS	STATISTICS	<input type="checkbox"/>	Ram Matali 1/9/2022, 12:06 AM

**Values**

**Inactive Values**

**Fields Where Used**

## STEP2:-

- WE NEED TO CREATE FIELD DEPENDENCIES

### STEPS TO CREATE FIELD DEPENDENCY

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

**College 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	Ram Matali 1/5/2022, 2:12 PM
Edit   Del	College Name	Capacity Of Students	Ram Matali 1/5/2022, 2:14 PM

**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.

Click button to include or exclude selected values from the dependent picklist:

College Name:	MIT-HYD	MIT-BLR	MIT-MUM	MIT-MAA	MIT-DEL
Email:	bir@mit.co.in hyd@mit.co.in	hyd@mit.co.in	hyd@mit.co.in mum@mit.co.in	bir@mit.co.in hyd@mit.co.in mum@mit.co.in maa@mit.co.in	bir@mit.co.in hyd@mit.co.in mum@mit.co.in maa@mit.co.in ccu@mit.co.in
	mum@mit.co.in	mum@mit.co.in	mum@mit.co.in	ccu@mit.co.in	ccu@mit.co.in
	maa@mit.co.in	maa@mit.co.in	maa@mit.co.in	dei@mit.co.in	dei@mit.co.in
	ccu@mit.co.in	ccu@mit.co.in	ccu@mit.co.in		
	dei@mit.co.in	dei@mit.co.in	dei@mit.co.in		

Showing Columns: 1 - 5 (of 6) < Previous | Next > [View All](#) [Go to](#)

Click button to include or exclude selected values from the dependent picklist:

**Save** **Cancel** **Preview**

**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.

Click button to include or exclude selected values from the dependent picklist:

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

Showing Columns: 1 - 5 (of 6) < Previous | Next > [View All](#) [Go to](#)

Click button to include or exclude selected values from the dependent picklist:

**Save** **Cancel** **Preview**

### STEP3:-

- WE NEED TO CREATE VALIDATION RULE

#### STEPS TO CREATE VALIDATION RULES

- Go to the Object Manager
- Search for college object

### 3. Click on College Object

### 4. Search for Validation Rules

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

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

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

```
AND(                               Ready_To_Join__c           ==           true,  
OR(  
ISCHANGED(                         Address__c             )           ,  
ISCHANGED(                         College__c            )           ,  
ISCHANGED(                         DateOfBirth__c       ),  
ISCHANGED(                         Email__c              )           ,  
ISCHANGED(                         GuardianName__c     )           ,  
ISCHANGED(                         Phone__c              )           )  
)  
)
```

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

- Page Layout:** College Validation Rule
- Validation Rule Detail:**
  - Rule Name:** Rule1
  - Error Condition Formula:** TEXT(College\_Name\_\_c) <> Name
  - Error Message:** please check your college name
  - Description:** (empty)
  - Created By:** Ram Malladi | 1/5/2022, 2:21 PM
  - Modified By:** Ram Malladi | 1/5/2022, 2:21 PM
- Left Sidebar:** Includes links for 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.
- Header:** Shows the URL vasireddyvenkatadninst80-dev-ed.lightning.force.com/lightning/setup/ObjectManager/011j000000hYDh/ValidationRules/03d5j000000suBAAY/view, the Salesforce logo, and various navigation icons.
- Bottom:** Shows the Windows taskbar with the Start button, a search bar, and several pinned application icons.

**DAY4**

06-01-2022

# Process Automation

## STEP1:-

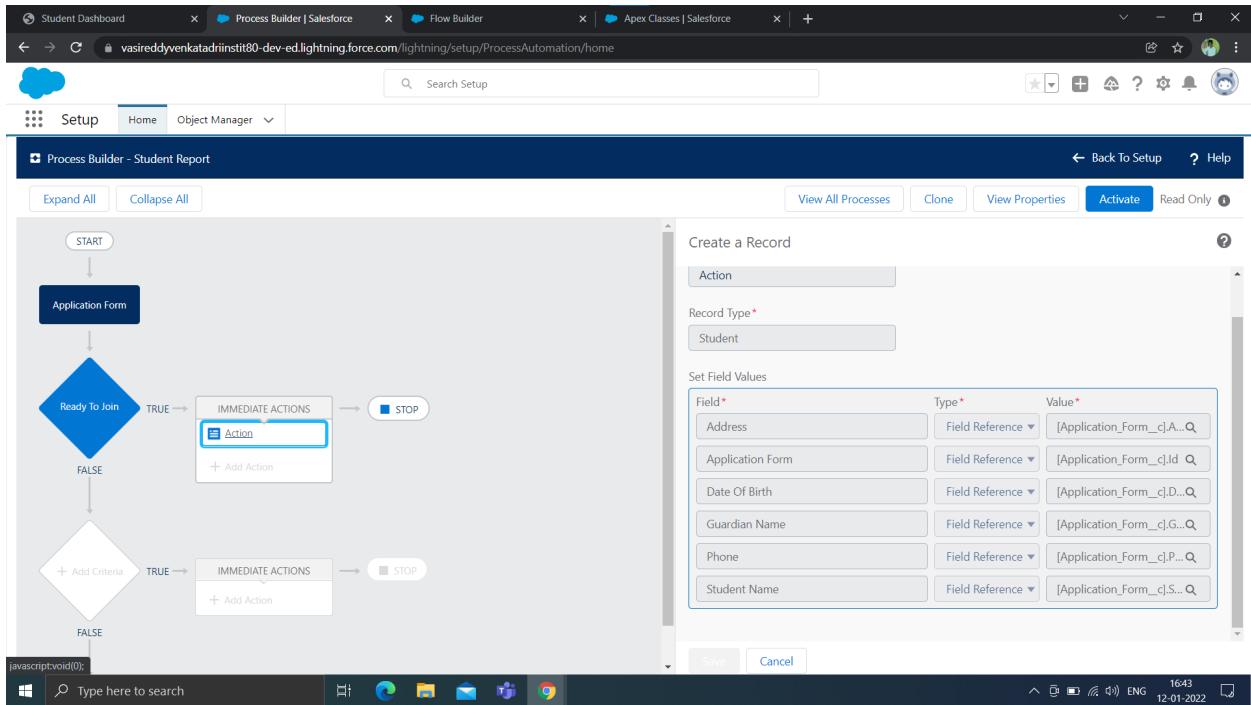
- WE NEED TO CREATE PROCESS AUTOMATION

### STEPS TO CREATE PROCESS AUTOMATION

1. OPEN SETUP
2. CLICK ON HOME
3. ENTER PROCESS BUILDER
4. CLICK NEW

1. 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.
2. Go to Setup -> select "Process Builder" from quick find. Create a Process Builder on the "Application Form" object with a condition as "When a record change". And select "When a record is created or edited".
  - a. In the diamond shape box(called nodes), select the criteria which trigger the Process builder to fire. In our example, it is "When Ready to Join field is checked."
  - b. Once the node is setup, click on the adjacent box called "Immediate action".

And select create a record on the student object



**DAY5**

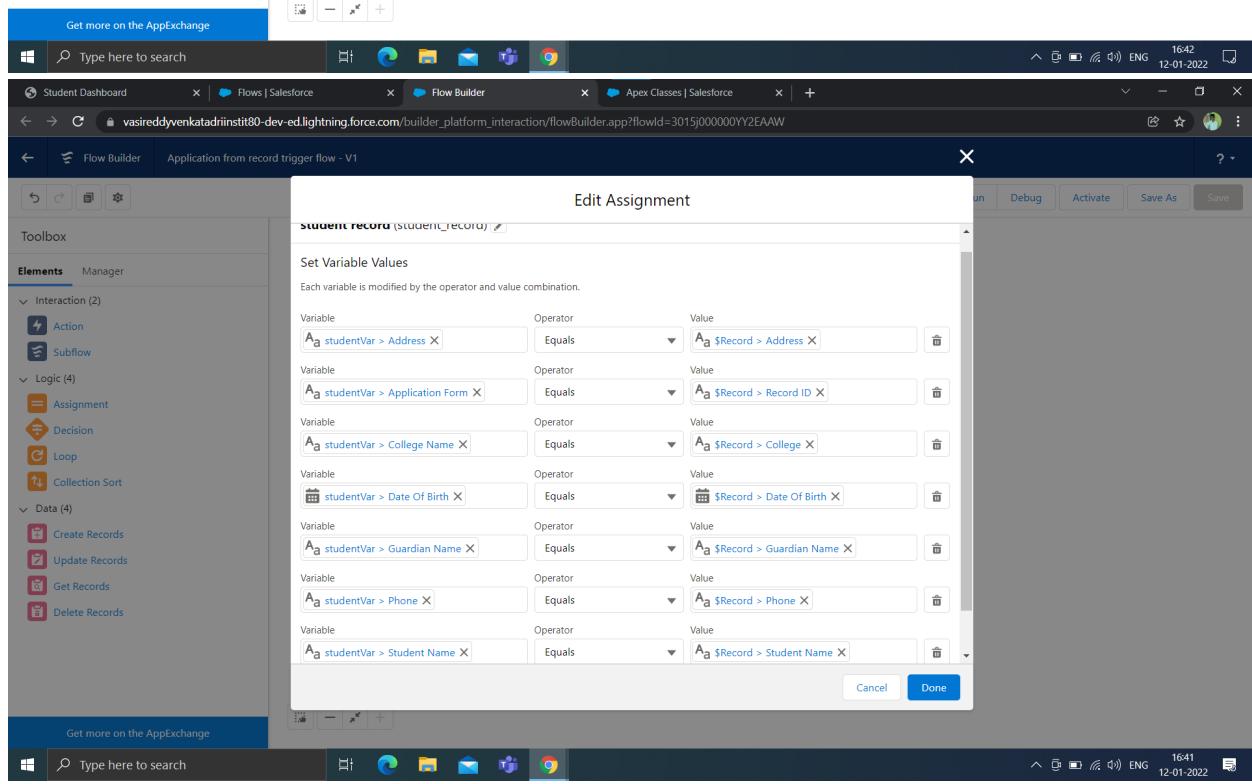
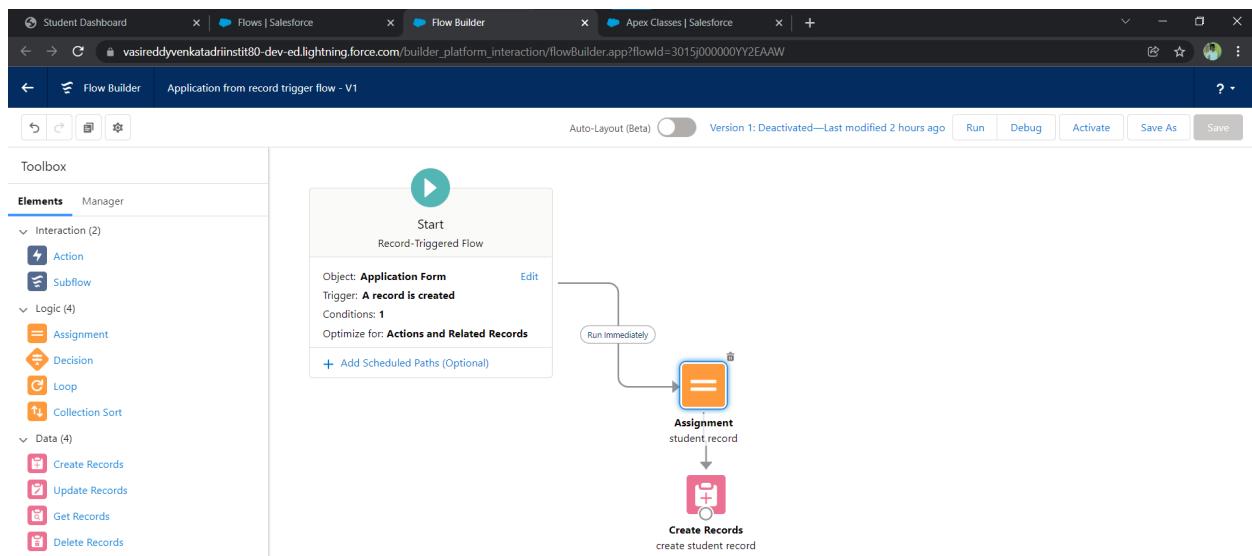
07-01-2022

## Create The Student Record Using Flow

- WE NEED TO CREATE A FLOW

### STEPS TO CREATE A FLOW

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

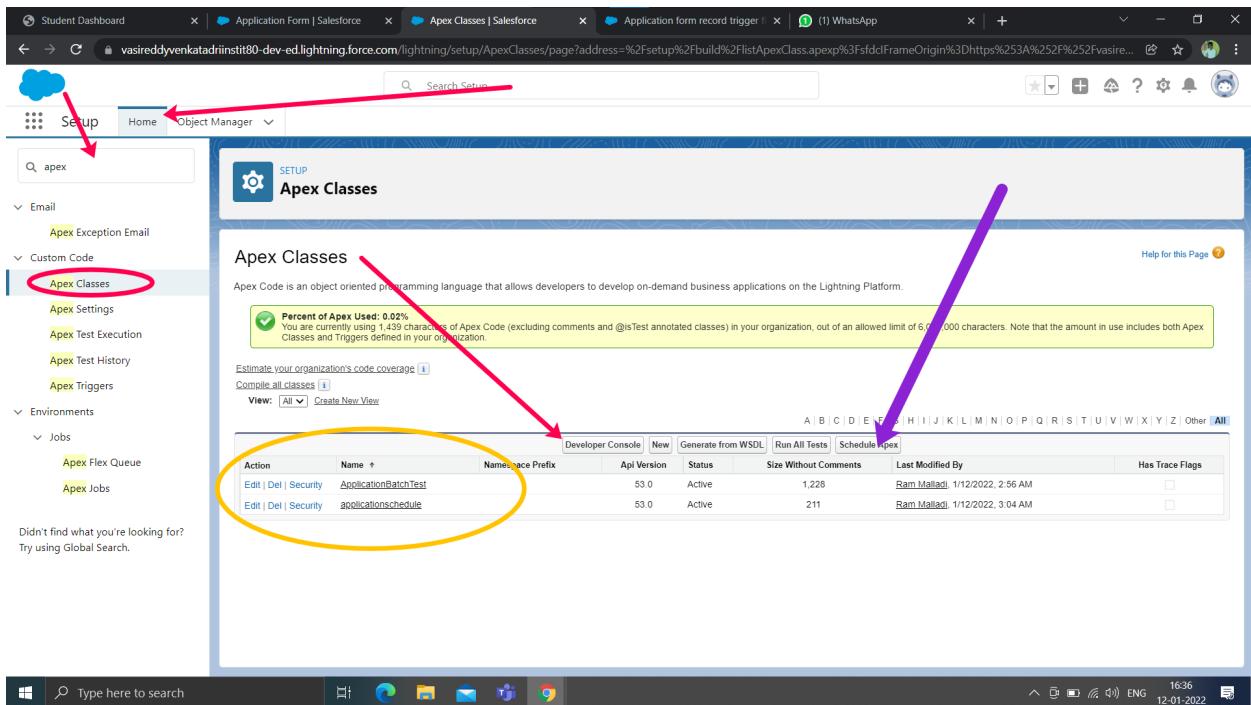


## Batch Apex

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  
    }  
}
```



## Create A Batch apex 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 });
}
}

```

## THE NEWBatch Apex

```

public Integer totalForms=0;
public Integer totalConvertedForms=0;
public Database.QueryLocator start(Database.BatchableContext bc){
    String applicationQuery='select id, Name, Ready_To_Join_c from Application_Form__c';
    return Database.getQueryLocator(applicationQuery);
}
public void execute(Database.BatchableContext bc, List<Application_Form__c> formList){
    for(Application_Form__c af : formList){
        totalForms++;
        if(af.Ready_To_Join__c){
            totalConvertedForms++;
        }
    }
}
public void finish(Database.BatchableContext bc){
    Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage();
    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[]{'m*****@gmail.com'};
    mail.setToAddresses(emailAddess);
    Messaging.sendEmail(new Messaging.SingleEmailMessage[]{ mail });
}
}

```

```

1 Public class ApplicationBatchTest implements Database.Batchable<sobject>,Database.stateful{
2     public Integer totalForms=0;
3     public Integer totalConvertedForms=0;
4     public Database.QueryLocator start(Database.BatchableContext bc){
5         String applicationQuery='select id, Name, Ready_To_Join__c from Application_Form__c';
6         return Database.getQueryLocator(applicationQuery);
7     }
8     public void execute(Database.BatchableContext bc, List<Application_Form__c> formList){
9         for(Application_Form__c af : formList){
10             totalForms++;
11             if(af.Ready_To_Join__c){
12                 totalConvertedForms++;
13             }
14         }
15     }
16     public void finish(Database.BatchableContext bc){
17         Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage();
18         mail.setSubject(' Application form and student record data as of today ');
19         mail.setPlainTextBody(' Total no of application form records are : '+totalForms+ ' out of which no of st
20         String[] emailAddress = new String[]{malladiseetharam0@gmail.com};
21         mail.setToAddresses(emailAddress);
22         Messaging.sendEmail(new Messaging.SingleEmailMessage[]{ mail });
23     }
24 }

```

**DAY7**

11-01-2022

## Create A Scheduler 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.

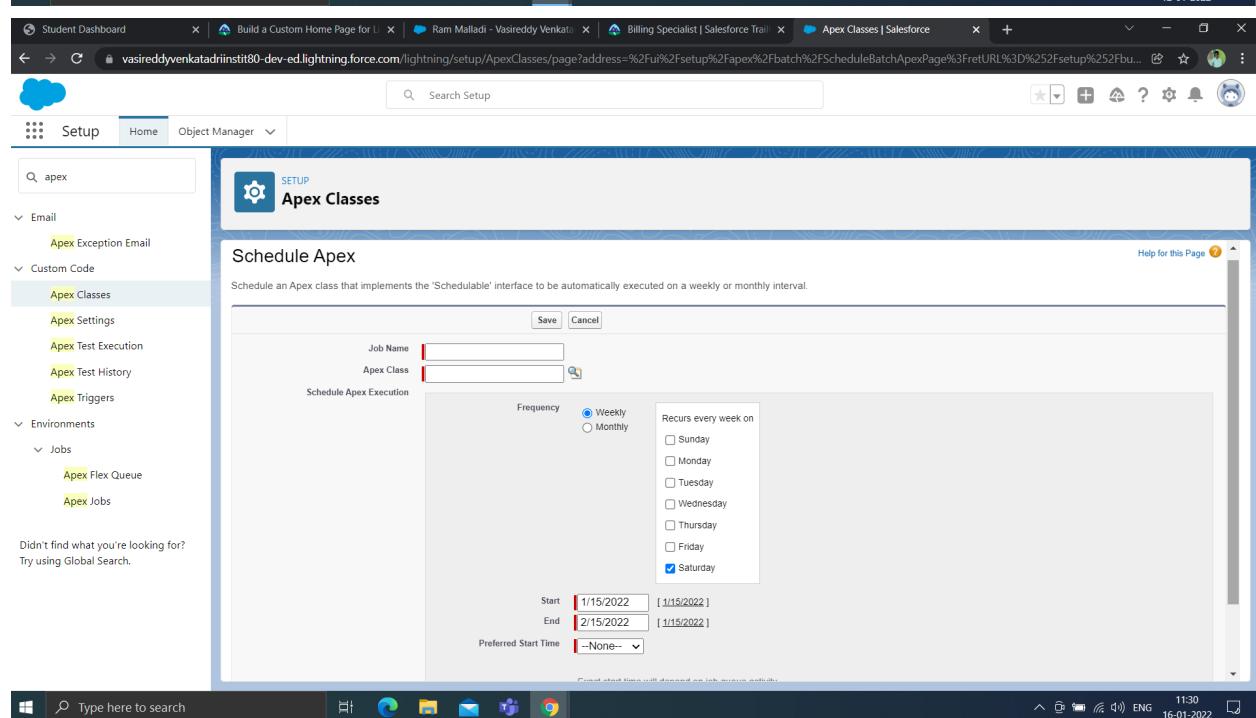
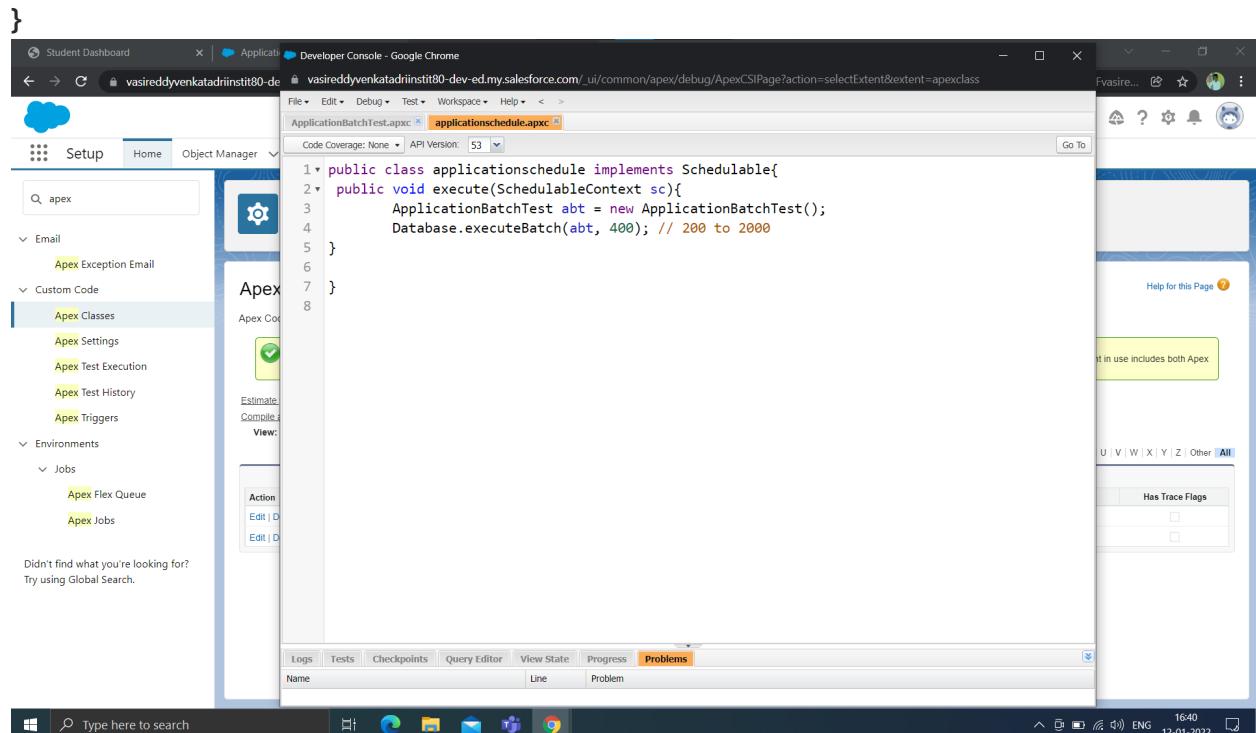
## MODIFIED CODE

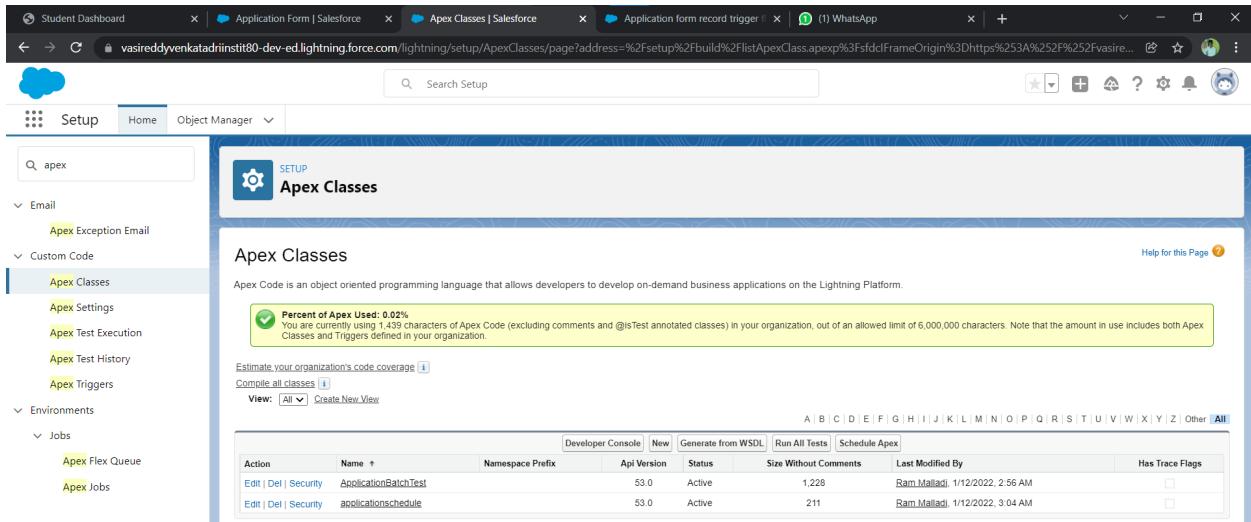
```
public class applicationschedule implements Schedulable{
```

```

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

```





**DAY8**

12-01-2022

## Lightning Web Components

Lightning Web Components (LWC) is a stack of modern lightweight frameworks built on the latest web standards. It is a DOM (Document Object Model), element created through reusable code and is used to generate a dynamic interface without using JavaScript or building a Library. This feasibility makes it quick and seamless, saving the developers a ton of time and effort on the Web Stack. Let's look at some of its remarkable features:

- 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

## Create College DataTable Component( APEX CLASS)

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

Create an Apex Class With The Required Functionality

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

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

        return formList;
    }

}
```

## MODIFIED CODE

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

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

        return formList;
    }

}
```

The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is `vasireddyvenkatadrinstit80-dev-ed.my.salesforce.com/_ui/common/apex/debug/ApexCIPage`. The tab title is `GetapplicationDetails.apxc`. The code editor contains 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 }
```

Below the code editor, there is a navigation bar with tabs: Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Problems tab is selected, showing an empty list.

## Create College DataTable Component (HTML FILE)

Create the component with the following files.

HTML:

```
<template>
```

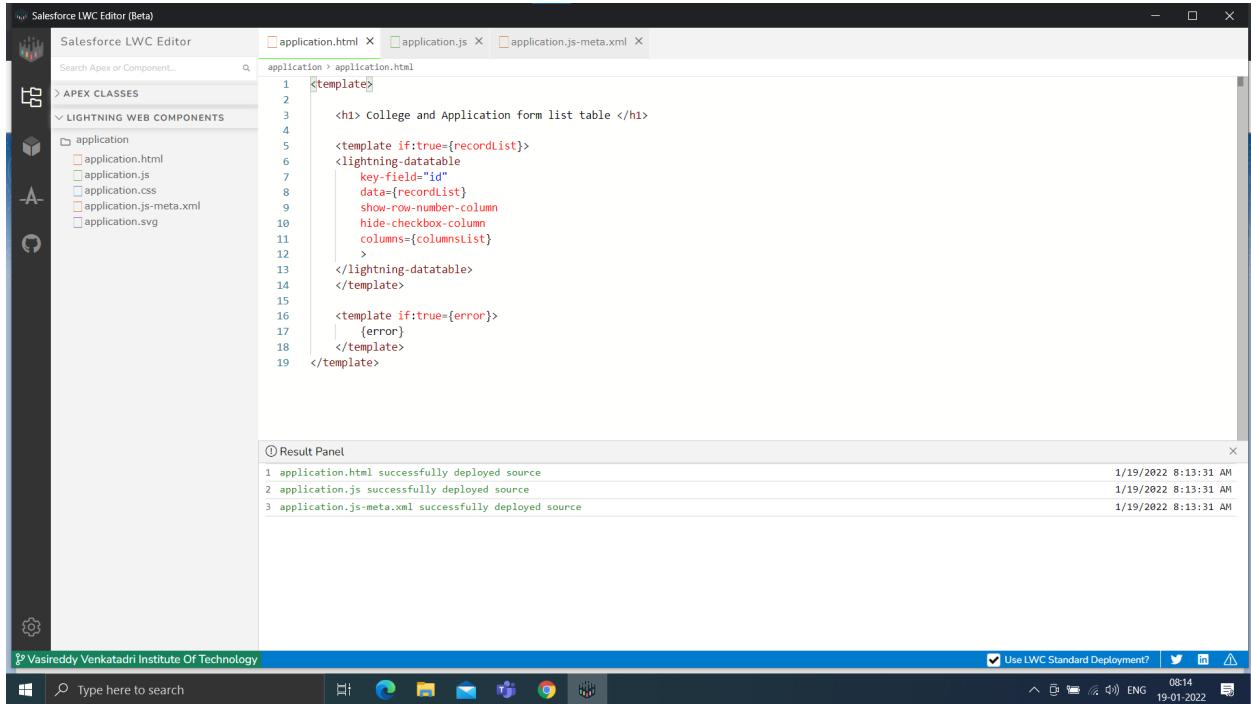
```
<h1> College and Application form list table </h1>
```

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

```

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

```



## Create College DataTable Component(JAVA SCRIPT FILE)

After the creation of the Html file create the following js 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;
    }
}
}

```

The screenshot shows the Salesforce LWC Editor (Beta) interface. The left sidebar displays the project structure under 'APEX CLASSES' and 'LIGHTNING WEB COMPONENTS'. The main editor area shows the contents of 'application.js'. The deployment history panel at the bottom shows three successful deployments:

Deployment	Date	Time
1 application.html	1/19/2022	8:13:31 AM
2 application.js	1/19/2022	8:13:31 AM
3 application.js-meta.xml	1/19/2022	8:13:31 AM

# Create College DataTable Component(META FILE)

Change the Meta File as follows:

```
<?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>
```

