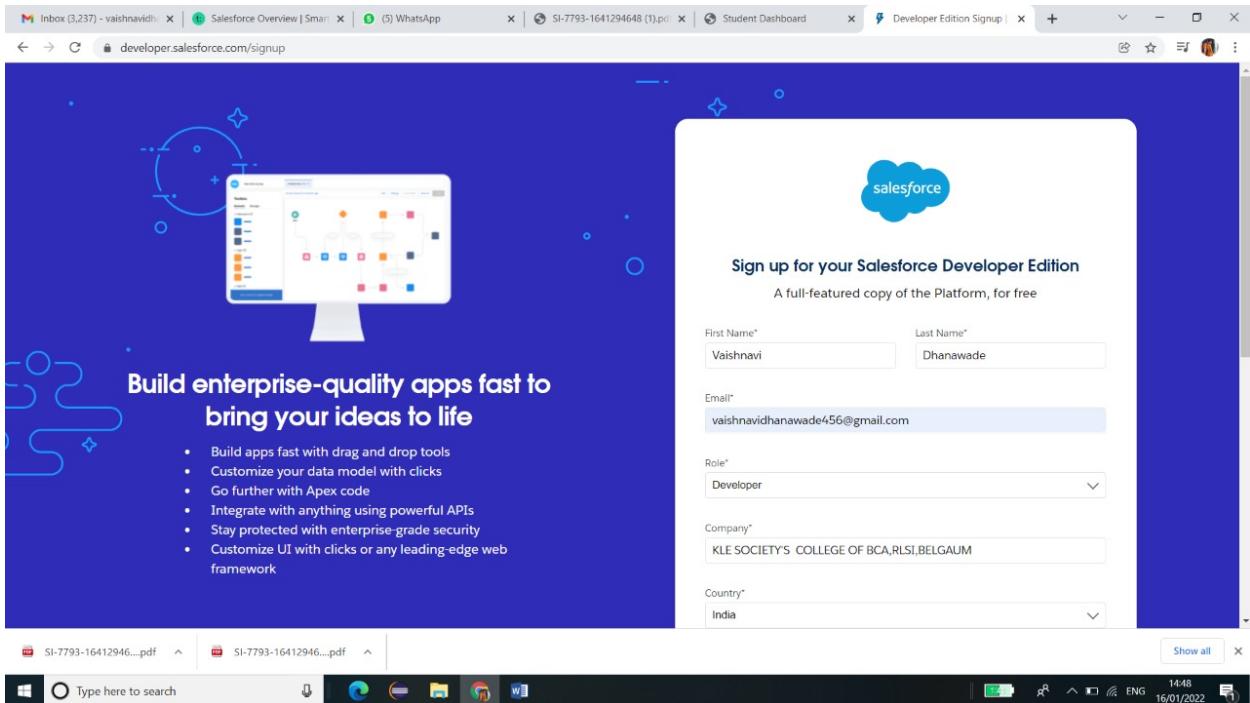
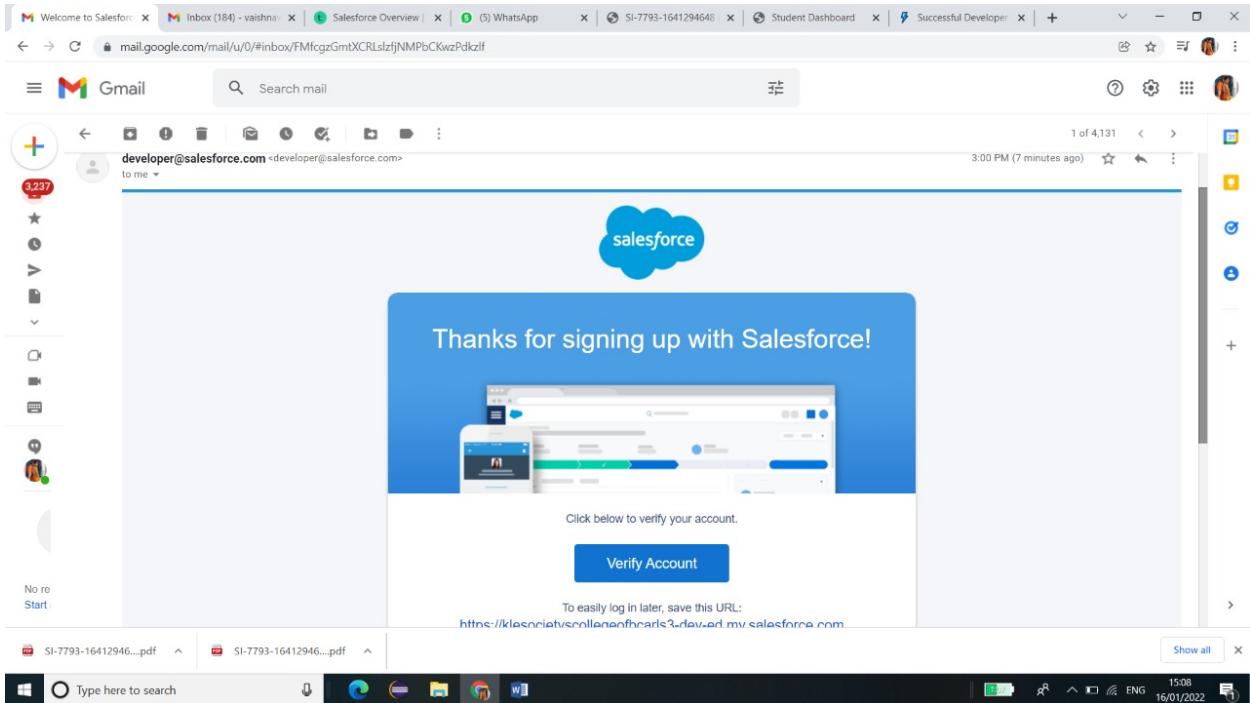


Day 1:

- To create College Management Application Project Created Salesforce Developer Org to get started



- Account activation
- Login to salesforce account



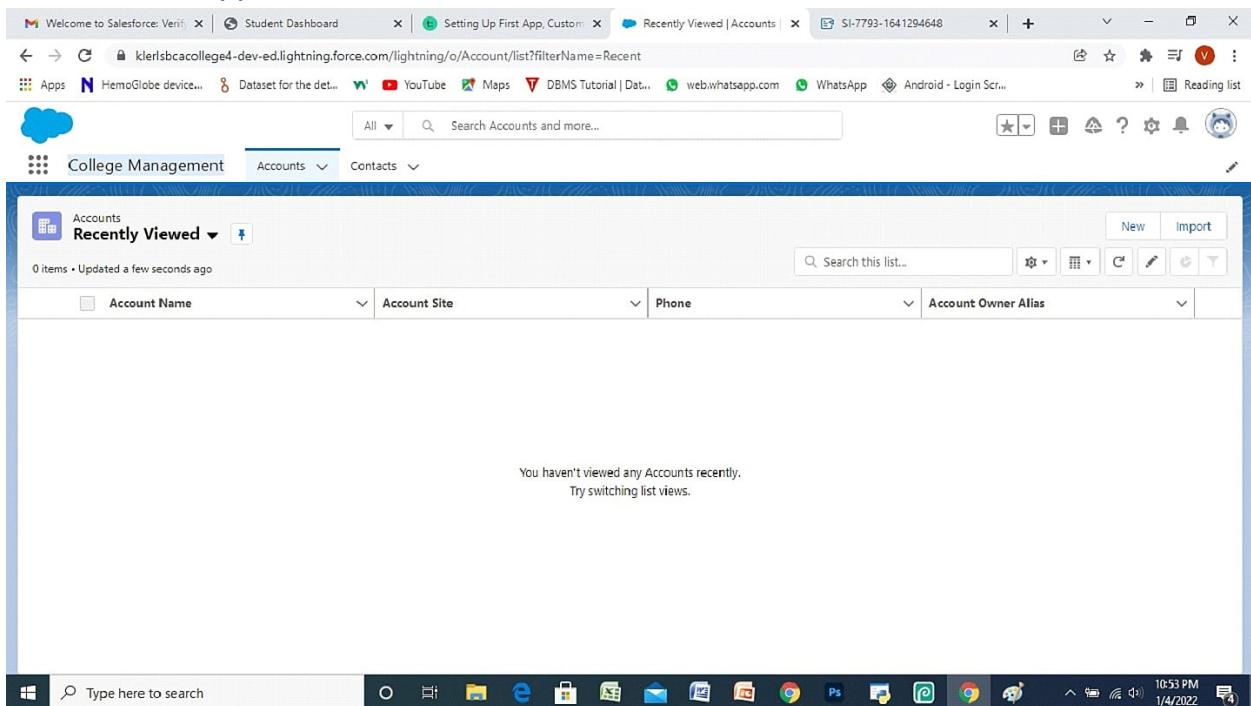
Additional things learned

4 Types of sandbox

1. developer sandbox
2. developer pro sandbox
3. partial copy sand box
4. full copy sandbox

Day 2:

- Custom object creation
- Create application called COLLEGE MANAGEMENT APPLICATION



Create the following Custom Objects For the Application.

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

- Create custom object called COLLEGE for application

Setup > OBJECT MANAGER

College

Details	
Fields & Relationships	
Page Layouts	
Lightning Record Pages	API Name College__c
Buttons, Links, and Actions	Custom ✓
Compact Layouts	Singular Label College
Field Sets	Plural Label Colleges
Object Limits	
Record Types	
Related Lookup Filters	
Search Layouts	
<input type="button" value="Edit"/> <input type="button" value="Delete"/>	

- Create fields for COLLEGE custom object
- Record info
- College Fees
- Hostel Fees
- College Name
- Email
- Capacity of students

Setup > OBJECT MANAGER

College

Fields & Relationships																																																								
Details	10 Items, Sorted by Field Label																																																							
Fields & Relationships	<table border="1"> <thead> <tr> <th>FIELD LABEL</th> <th>FIELD NAME</th> <th>DATA TYPE</th> <th>CONTROLLING FIELD</th> <th>INDEXED</th> </tr> </thead> <tbody> <tr> <td>Capacity of Students</td> <td>Capacity_of_Students__c</td> <td>Picklist</td> <td></td> <td></td> </tr> <tr> <td>College Fees</td> <td>College_Fees__c</td> <td>Currency(7, 2)</td> <td></td> <td></td> </tr> <tr> <td>College Name</td> <td>College_Name__c</td> <td>Picklist</td> <td></td> <td></td> </tr> <tr> <td>College Name</td> <td>Name</td> <td>Text(80)</td> <td>✓</td> <td></td> </tr> <tr> <td>Created By</td> <td>CreatedBy</td> <td>Lookup(User)</td> <td></td> <td></td> </tr> <tr> <td>Email</td> <td>Email__c</td> <td>Picklist</td> <td></td> <td></td> </tr> <tr> <td>Hostel Fees</td> <td>Hostel_Fees__c</td> <td>Currency(6, 2)</td> <td></td> <td></td> </tr> <tr> <td>Last Modified By</td> <td>LastModifiedBy</td> <td>Lookup(User)</td> <td></td> <td></td> </tr> <tr> <td>Owner</td> <td>OwnerId</td> <td>Lookup(User/Group)</td> <td>✓</td> <td></td> </tr> <tr> <td>Record info</td> <td>Record_info__c</td> <td>Text(10)</td> <td></td> <td></td> </tr> </tbody> </table>	FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED	Capacity of Students	Capacity_of_Students__c	Picklist			College Fees	College_Fees__c	Currency(7, 2)			College Name	College_Name__c	Picklist			College Name	Name	Text(80)	✓		Created By	CreatedBy	Lookup(User)			Email	Email__c	Picklist			Hostel Fees	Hostel_Fees__c	Currency(6, 2)			Last Modified By	LastModifiedBy	Lookup(User)			Owner	OwnerId	Lookup(User/Group)	✓		Record info	Record_info__c	Text(10)		
FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED																																																				
Capacity of Students	Capacity_of_Students__c	Picklist																																																						
College Fees	College_Fees__c	Currency(7, 2)																																																						
College Name	College_Name__c	Picklist																																																						
College Name	Name	Text(80)	✓																																																					
Created By	CreatedBy	Lookup(User)																																																						
Email	Email__c	Picklist																																																						
Hostel Fees	Hostel_Fees__c	Currency(6, 2)																																																						
Last Modified By	LastModifiedBy	Lookup(User)																																																						
Owner	OwnerId	Lookup(User/Group)	✓																																																					
Record info	Record_info__c	Text(10)																																																						
Page Layouts																																																								
Lightning Record Pages																																																								
Buttons, Links, and Actions																																																								
Compact Layouts																																																								
Field Sets																																																								
Object Limits																																																								
Record Types																																																								
Related Lookup Filters																																																								
Search Layouts																																																								
Search Layouts for Salesforce Classic																																																								
Restriction Rules																																																								
Triggers																																																								

- Create Fields on Application Form object
- Address
- College
- College Fees
- Hostel Fees
- Date of birth
- Email
- Guardian Name
- Looking for hostel stay
- Ready to join
- Student name
- Phone

Application record created

The screenshot shows a web browser window with multiple tabs open. The active tab displays an application form record titled "F-000002". The form contains the following data:

Field	Value
Application Form	F-000002
Address	belaGum
College	MIT-CCU
Date of birth	1/1/2022
Email	valish@ppp.com
Looking For Hostel Stay	<input type="checkbox"/>
Ready To Join	<input type="checkbox"/>
Phone	435789566
Created By	Vaishnavi Kadam, 1/9/2022, 7:04 AM
Last Modified By	Vaishnavi Kadam, 1/9/2022, 7:04 AM

Day 3:

- Create fields of Student
- Address
- Application Form
- College name
- Date of birth

- Guardian name
- Phone

SETUP > OBJECT MANAGER
Student

Fields & Relationships					
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		LastModifiedById	Lookup(User)		
Owner		OwnerId	Lookup(User,Group)	✓	
Phone		Phone_c	Phone		
Student Name		Name	Text(80)	✓	

- Create fields on Subject object
- Paper1
 - Paper2
 - Student

SETUP > OBJECT MANAGER
Subject

Fields & Relationships					
FIELD LABEL		FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By		CreatedBy	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 Name		Name	Auto Number	✓	

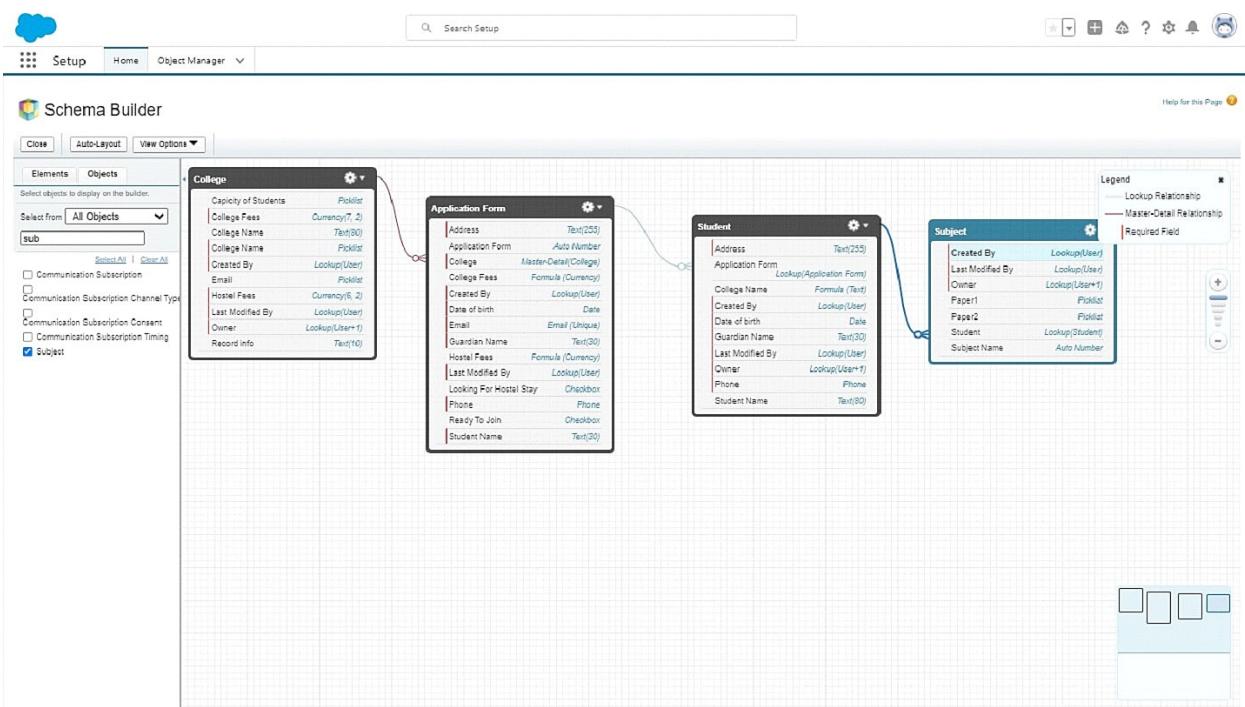
college management application display

The screenshot shows a web-based application interface for college management. At the top, there is a navigation bar with tabs for 'College Management', 'Colleges', 'Application Forms', 'Students', and 'Subjects'. Below the navigation bar is a search bar labeled 'Search Colleges and more...'. On the left, there is a sidebar titled 'Recently Viewed' which lists five colleges: MIT-CCU, MIT-MAA, MIT-MUM, MIT-HYD, and MIT-BLR. The main area of the screen displays a table with columns for 'College Name' and other details, though the content is mostly obscured by a large white box.

here the tabs are in lookup relationship

COLLEGE > APPLICATION FORM > STUDENT > SUBJECT

► Schema Builder



It is Pictorial representation schema diagram

► **Legend** it shows symbolic representation ,which symbol represents what.

Schema Builder

The Schema Builder interface displays two objects: **College** and **Application Form**. The **College** object has fields like Capacity of Students (Picklist), College Fees (Currency(7, 2)), College Name (Text(80)), etc. The **Application Form** object has fields like Address, Application Form (Master-Detail relationship with College), College Fees, Created By, Date of birth, Email, and Guardian Name. A legend box is overlaid on the **Application Form** object, defining symbols: a blue line for Lookup Relationship, a red line for Master-Detail Relationship, and a red square for Required Field.

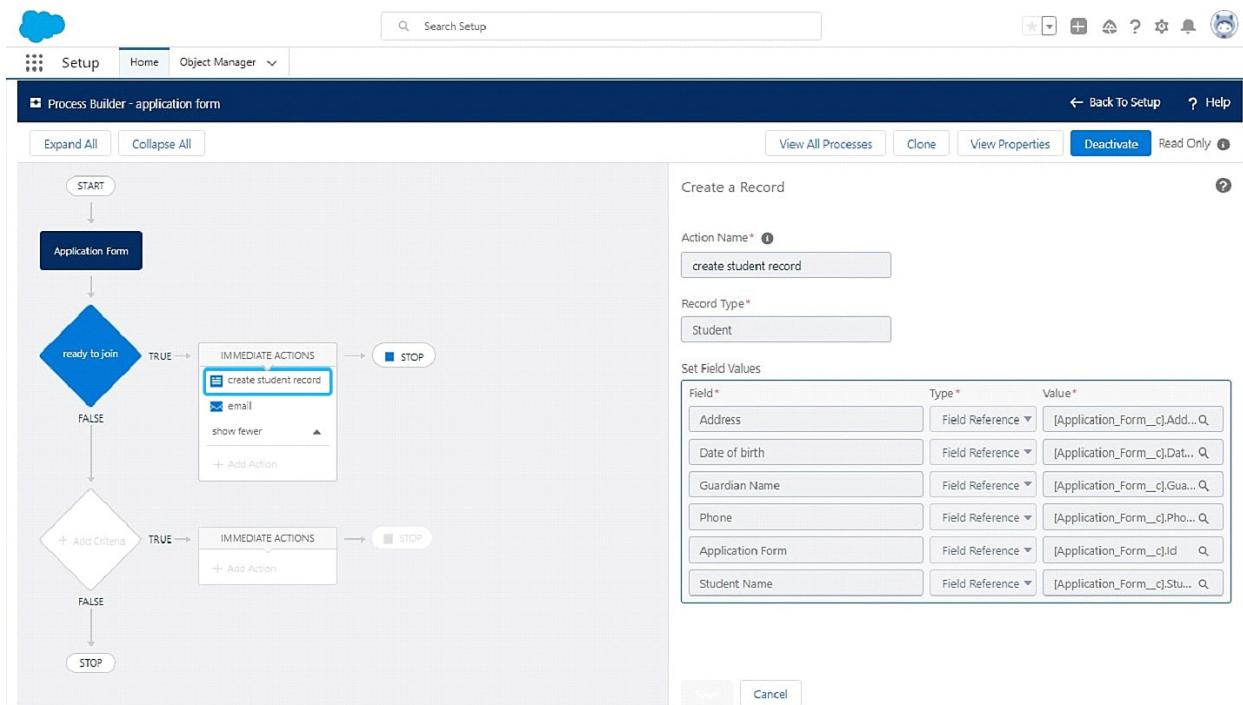
► **How to display list view fields**

1. Go to list view ,select all
2. Click list view control
3. Select fields to display
4. Select the fields that you want to display.

The application interface shows the **Application Forms** list view. The table header includes columns for Application Form, Student Name, Date of birth, Email, College, College Fees, Hostel Fees, Guardian Name, and Phone. The table body contains two rows of data:

	Application Form	Student Name	Date of birth	Email	College	College Fees	Hostel Fees	Guardian Name	Phone
1	F-000002	vaishnavi k	1/1/2022	vaish@ggg.com	MIT-CCU	\$140,000.00	\$13,000.00	jeevan	435789566
2	F-000003	ross ja	2/13/2004	jack12@gmail.com	MIT-MUM	\$150,000.00	\$120,000.00	sona	123456754

► Process Automation



New record created

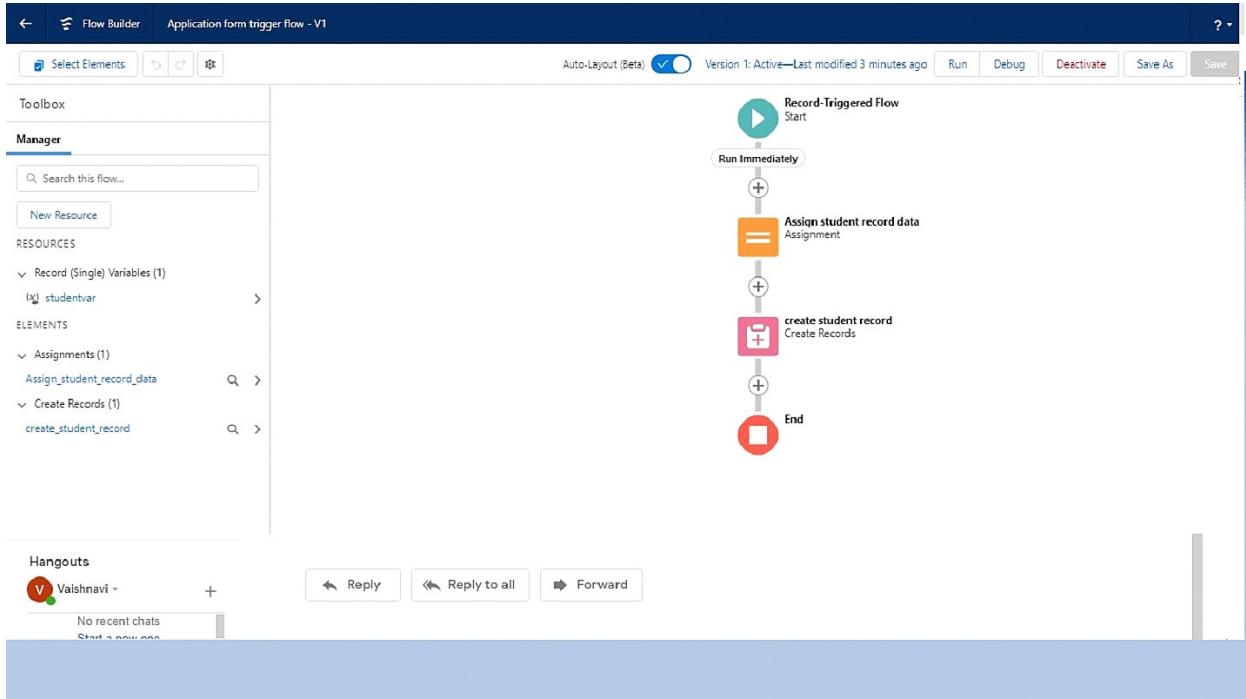
The screenshot shows the "College Management" application form details for record "F-000017". The "Details" tab is selected, displaying the following fields:

- Student Name: jaya
- Address: mumbai
- College: MIT-CCU
- Email: jaya@jaya.com
- Looking For Hostel Stay:
- Ready To Join:
- Created By: Vaishnavi Kadam, 1/14/2022, 12:55 AM
- Collega Fees: \$140,000.00
- Hostel Fees: \$13,000.00
- Guardian Name: father
- Date of birth: 1/8/1998
- Phone: (897) 117-3565
- Application Form: F-000015
- Last Modified By: Vaishnavi Kadam, 1/14/2022, 12:55 AM

The screenshot shows a web-based application interface for 'College Management'. At the top, there's a navigation bar with 'All', a search bar ('Search Application Forms and more...'), and various icons. Below the header, the page title is 'Application Form F-000017'. The main content area has tabs for 'Related' and 'Details'. Under 'Related', there's a section for 'Notes & Attachments (0)' with a file upload button and a placeholder 'Or drop files'. Another section for 'Students (1)' shows a single entry with 'Student Name' set to 'Jaya', a 'New' button, and a 'View All' link.

Day 4:

- Create student record using flow



► Assign student record data

The screenshot shows the 'Edit Assignment' screen in the Flow Builder. The title bar says 'Edit Assignment' and the sub-title is 'Assign student record data (Assign_student_record_data)'. The main area is titled 'Set Variable Values' with the sub-instruction 'Each variable is modified by the operator and value combination.' Below this, there are seven assignment rows:

- Variable: A_d studentvar > Address X Operator: Equals Value: A_d \$Record > Address X
- Variable: A_d studentvar > Guardian Name X Operator: Equals Value: A_d \$Record > Guardian Name X
- Variable: A_d studentvar > Phone X Operator: Equals Value: A_d \$Record > Phone X
- Variable: A_d studentvar > Application Form X Operator: Equals Value: A_d \$Record > Record ID X
- Variable: A_d studentvar > College Name X Operator: Equals Value: A_d \$Record > College X
- Variable: A_d studentvar > Date of birth X Operator: Equals Value: A_d \$Record > Date of birth X
- Variable: A_d studentvar > Student Name X Operator: Equals Value: A_d \$Record > Student Name X

At the bottom left is a '+ Add Assignment' button, and at the bottom right are 'Cancel' and 'Done' buttons.

► Record created using flow

The screenshot shows the 'Application Form' details for record F-000018 in the College Management application. The top navigation bar includes 'College Management', 'Colleges', 'Application Forms', 'Students', and 'Subjects'. The main view shows the 'Details' tab selected, displaying the following data:

Field	Value
Student Name	ganesh
Address	pune
College	MIT-BLR
Email	ganesh@gane.com
Looking For Hostel Stay	<input type="checkbox"/>
Ready To Join	<input checked="" type="checkbox"/>
Created By	Vaishnavi Kadam, 1/14/2022, 5:46 AM
College Fee	\$140,000.00
Hostel Fees	\$10,000.00
Guardian Name	mother
Date of Birth	11/20/2009
Phone	658755754
Application Form	F-000018
Last Modified By	Vaishnavi Kadam, 1/14/2022, 5:46 AM

At the top right of the application window are buttons for 'New Contact', 'Edit', and 'New Case'.

► Related data

The screenshot shows a Salesforce application form for 'Application Form F-000018'. The 'Related' tab is selected, displaying a list of 'Students (1)'. One student, 'ganesh', is listed with the name 'Student Name'. A 'Notes & Attachments (0)' section is also present.

- Learned about developer console
- Salesforce primitive datatypes : boolean ,string,date,time,id,integer ,double
- Collection: list,map,set

Day 5:

Query

The screenshot shows the Salesforce Developer Console with a query result grid and history panel.

Query Results - Total Rows: 10

Student_Name__c	Guardian_Name__c	Date_of_birth__c
vaishnavi k	jeevan	2022-01-01
ross ja	sona	2004-02-13
ross ja	father	2001-02-27
jj	jkh	2022-01-05
vini	mom	2001-03-22
w	jkh	2022-01-11
hh	hhhhh	2022-01-21
jim	hhhhh	2003-05-07
jaya	father	1998-01-08
ganesh	mother	2009-11-20

Access in Salesforce: Create New | Open Detail Page | Edit Page

History

```
select Student_Name__c,Guardian_Name__c,Date_of_birth__c from Application_Form__c
```

Any query errors will appear here...

SOQL :Salesforce object query

Object Query example for Application form

- SOSL: salesforce object search language
- Collection
- Apex

➤ Two types of apex classes

1. Sync
2. Async

➤ Types of async apex:

1. Future method
2. batch class
3. queueable class
4. scheduler class

Day 6:

➤ Create a batch apex for application form

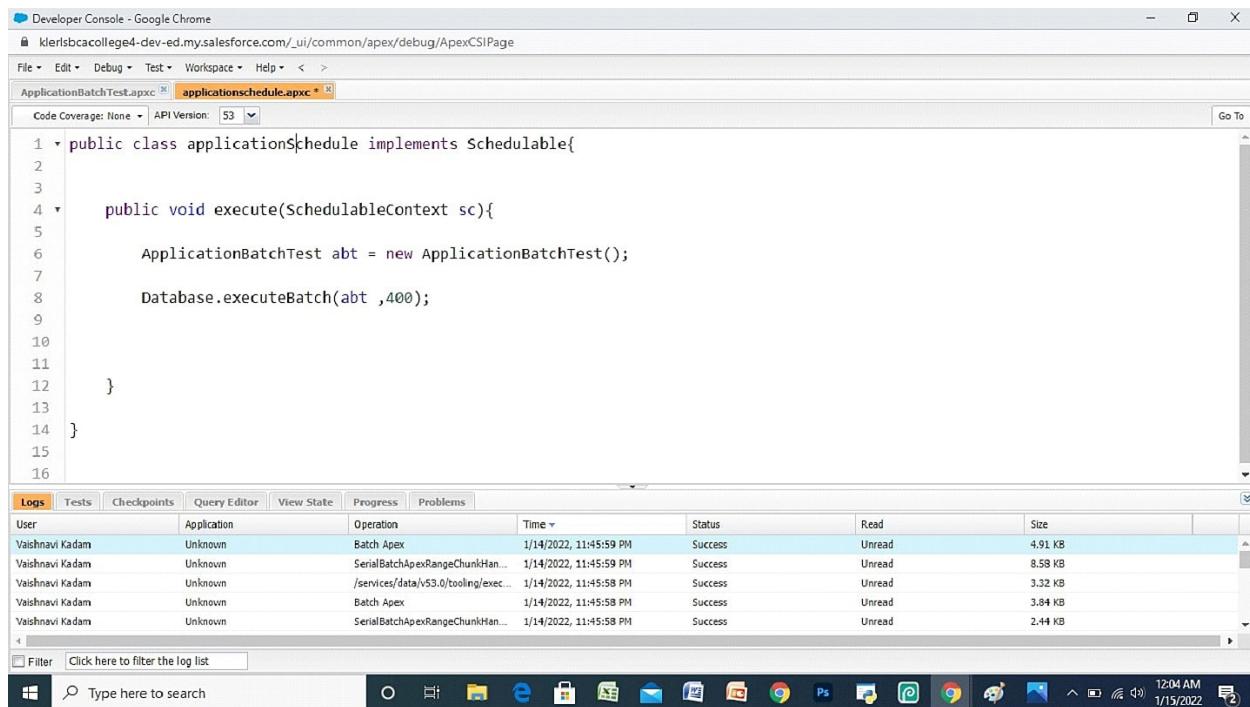
The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is https://klerlsbcacollege4-dev-ed.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage. The tabs at the top are ApplicationBatchTest.apxc and applicationSchedule.apxic. The code editor contains the following Apex class:

```
1 - public class ApplicationBatchTest implements Database.Batchable<sObject>, Database.Stateful {
2
3     //start(), execute(). finish()
4     public Integer totalForms = 0; // total no of application form
5     public Integer totalConvertedForms = 0; // total no of students
6     public Database.QueryLocator start(Database.BatchableContext bc){
7         // gathers the data for you
8         String applicationQuery = 'Select id, Name, Ready_To_Join__c From Application_Form__c';
9         return Database.getQueryLocator(applicationQuery);
10    }
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 admin)
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 per today : '+totalConvertedForms);
26        String[] emailAddess = new String[]{vaishnavik052@gmail.com};
27        mail.setToAddresses(emailAddess);
28        Messaging.sendEmail(new Messaging.SingleEmailMessage[]{ mail });
29    }
30 }
```

The Log tab at the bottom shows the following log entries:

User	Appliation	Operator	Time	Status	Read	Size
Vaishnavi Kadam	Unknown	Batch Apex	1/1/2022, 11:15:59 PM	Success	Unread	4.91 KB
Vaishnavi Kadam	Unknown	SerialBatchApexRangeChunkHandler	1/1/2022, 11:15:59 PM	Success	Unread	0.58 KB
Vaishnavi Kadam	Unknown	/server/data/v31.0/tools/executearmymous/	1/1/2022, 11:15:59 PM	Success	Unread	3.12 KB
Vaishnavi Kadam	Unknown	Batch Apex	1/1/2022, 11:15:59 PM	Success	Unread	3.04 KB

► Create a scheduler class



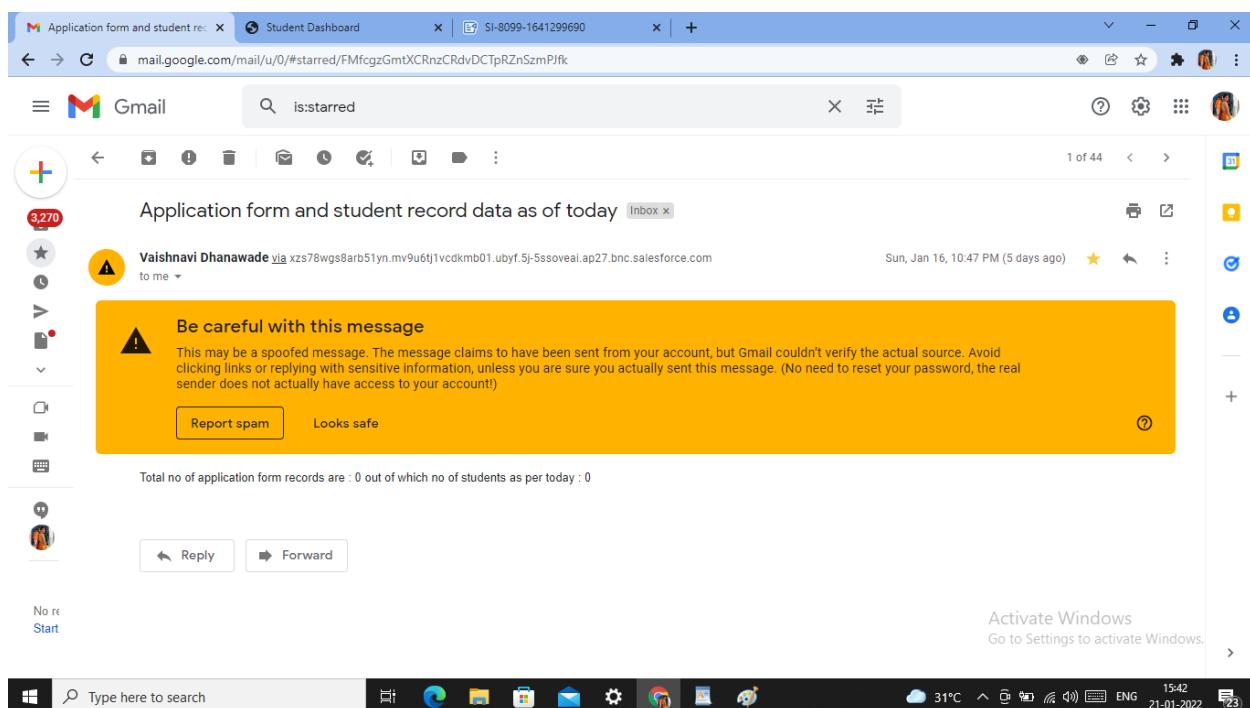
```
1 public class applicationSchedule implements Schedulable{
2
3     public void execute(SchedulableContext sc){
4
5         ApplicationBatchTest abt = new ApplicationBatchTest();
6
7         Database.executeBatch(abt ,400);
8
9
10    }
11
12 }
13
14 }
15
16 }
```

The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is https://klerisbcacollege4-dev-ed.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage. The page title is "Developer Console - Google Chrome". The tabs at the top include File, Edit, Debug, Test, Workspace, Help, and Go To. The active tab is "applicationschedule.apxc". The code editor shows the Apex class definition. Below the code is a log table with the following data:

User	Application	Operation	Time	Status	Read	Size
Vaishnavi Kadam	Unknown	Batch Apex	1/14/2022, 11:45:59 PM	Success	Unread	4.91 KB
Vaishnavi Kadam	Unknown	SerialBatchApexRangeChunkHan...	1/14/2022, 11:45:59 PM	Success	Unread	8.58 KB
Vaishnavi Kadam	Unknown	/services/data/v53.0/tooling/exec...	1/14/2022, 11:45:58 PM	Success	Unread	3.32 KB
Vaishnavi Kadam	Unknown	Batch Apex	1/14/2022, 11:45:58 PM	Success	Unread	3.84 KB
Vaishnavi Kadam	Unknown	SerialBatchApexRangeChunkHan...	1/14/2022, 11:45:58 PM	Success	Unread	2.44 KB

At the bottom of the developer console, there is a search bar and a taskbar with various icons.

► Email I received



The screenshot shows a Gmail inbox. The email from Vaishnavi Dhanawade has a yellow warning box over it. The box contains the text: "Be careful with this message. This may be a spoofed message. The message claims to have been sent from your account, but Gmail couldn't verify the actual source. Avoid clicking links or replying with sensitive information, unless you are sure you actually sent this message. (No need to reset your password, the real sender does not actually have access to your account!)" There are two buttons at the bottom of the box: "Report spam" and "Looks safe". The email details are: Subject: Application form and student record data as of today, From: Vaishnavi Dhanawade via xzs78wgs8arb51yn.mv9u6tj1vcdkmb01.ubyf.5j-5ssoveal.ap27.bnc.salesforce.com, To: me, Date: Sun, Jan 16, 10:47 PM (5 days ago). At the bottom of the email view, it says "Total no of application form records are : 0 out of which no of students as per today : 0". Below the email, there are "Reply" and "Forward" buttons. The Windows taskbar at the bottom shows the date as 21-01-2022 and the time as 15:42.

DAY 7:

1. Apex Trigger

- > Two types of triggers
Before trigger and After trigger
- > Operations performed :- insert ,update ,delete ,undelete

2. Trigger context variables

- > trigger.new, trigger.old ,trigger.new map ,trigger.old map.

- Learned about how to delete trigger and use of custom label in trigger.
- Performed real time scenario : which do not allow to delete application form record as it has student record associated with it.
- Custom label : Where we store some static data/text.

3. Best practices for Apex and Test classes

- > Bulkification :whatever code we write should be capable of handling everything.
- > Should have collection variable whenever we write class,trigger.
- > Never write DML statements inside for loop
- > If you want to add DML statements than use list, map it will not hit any governor limits.
- > Can use Multiple for() loop but cannot use nested for loop

4. Test class: it is automate script for a developer to test his/her logic

- trigger minimal coverage =0%/1%
- apex minimal coverage should be more than 75%

5. - @istest

- @testup
- @test.start test();
- @test.stop test();

6. System.assertEquals(actual,expected);

DAY 8:

Introduction to LWC (lightning web component)

- > What is LWC:It is programming model to develop salesforce lightning components.
- > Why LWC
- > Benefits of LWC
- > Web stack transformation

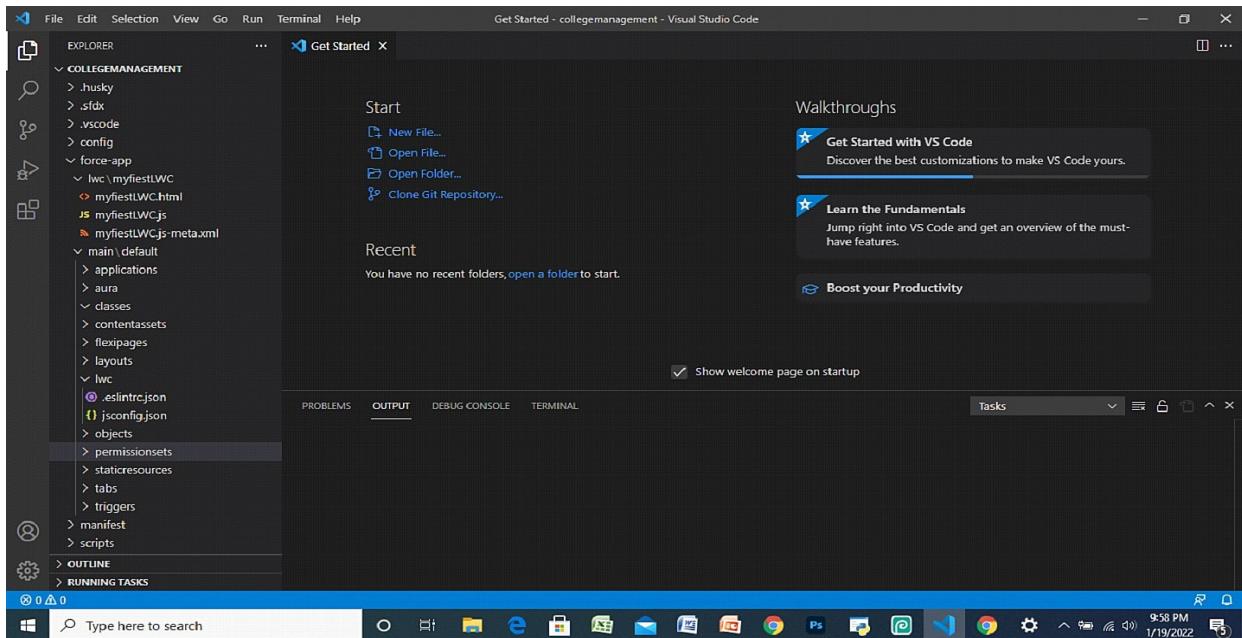
> LWC file contains main 3 files

- html ,javascript(js),meta-xml and one out of the box css file.

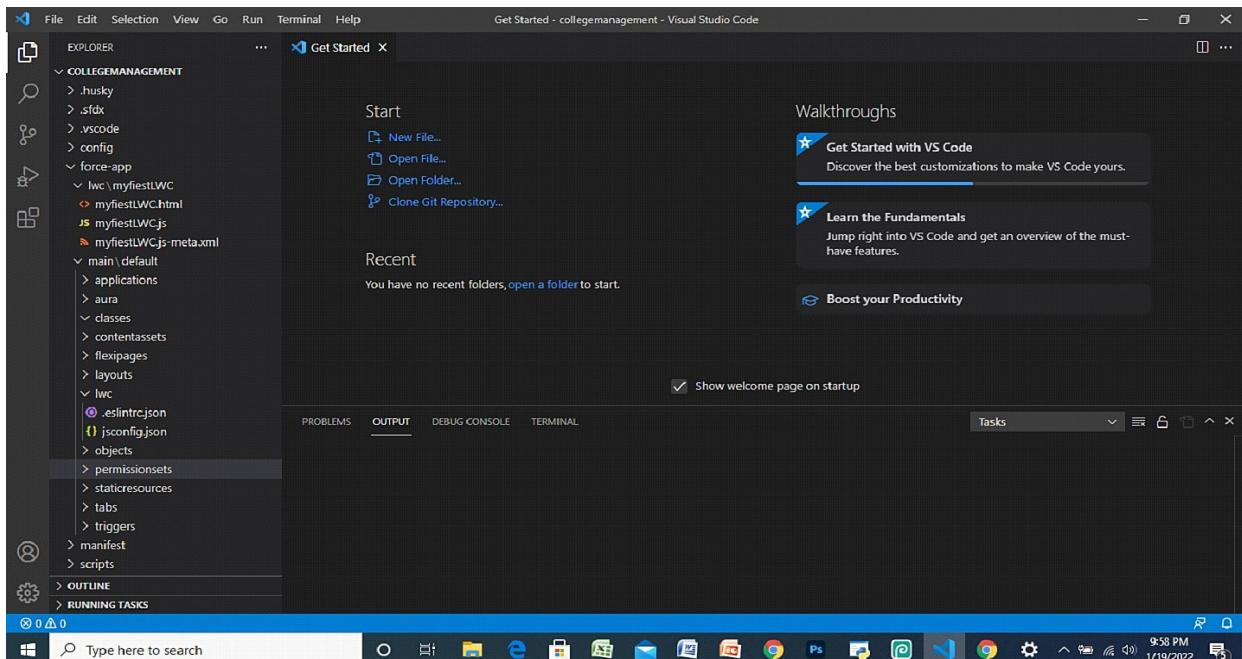
> Introduction to javascript

- js functions

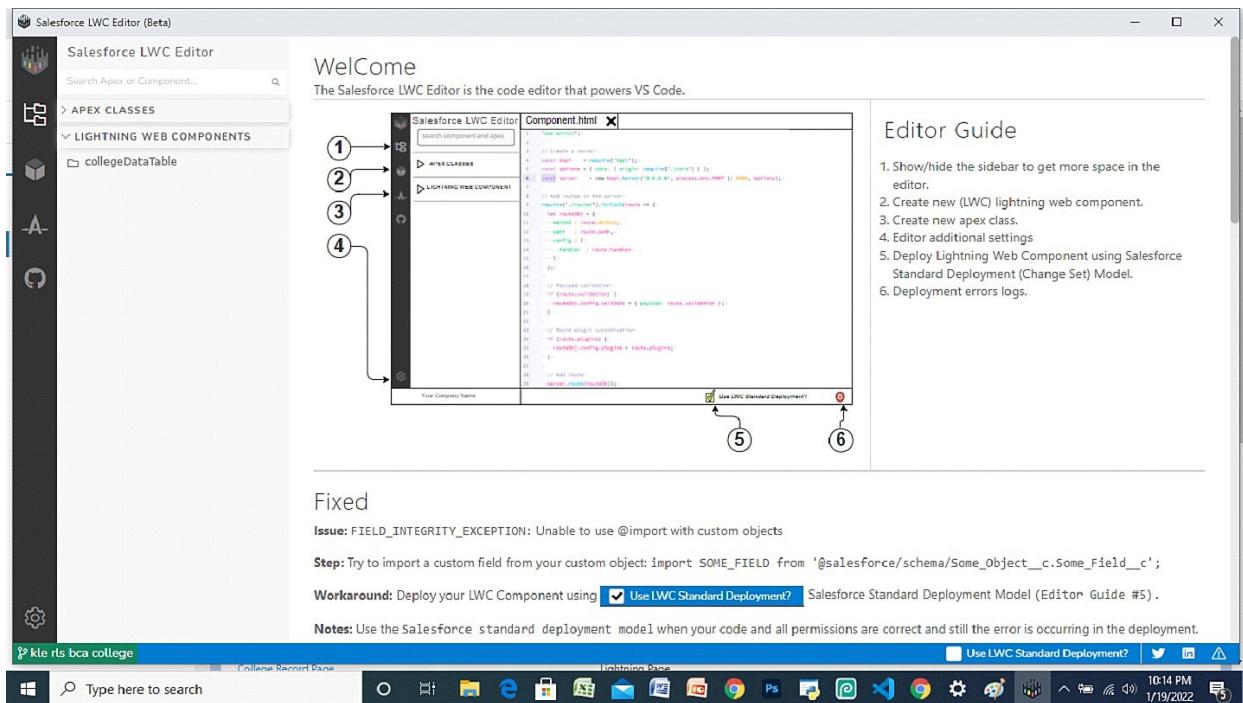
> Installation of vs.code



> In vs.code download Salesforce extension pack



> Installation of Salesforce LWC editor



Fixed

Issue: FIELD_INTEGRITY_EXCEPTION: Unable to use @import with custom objects

Step: Try to import a custom field from your custom object: import SOME_FIELD from '@salesforce/schema/Some_Object__c.Some_Field__c';

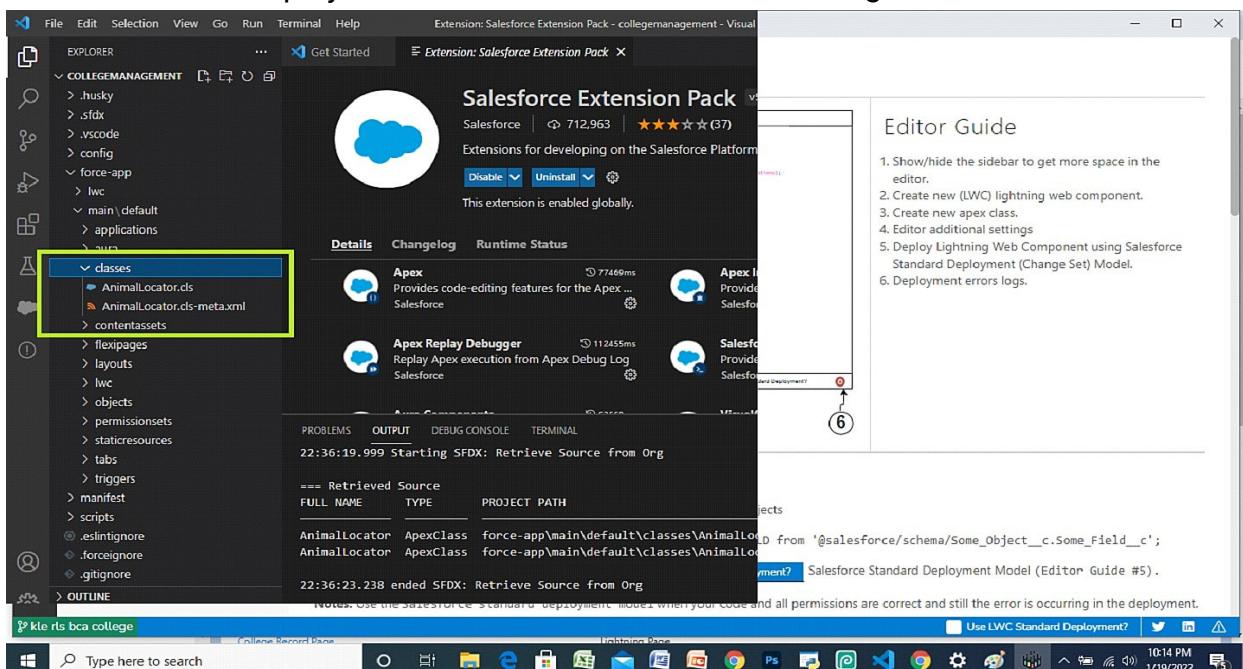
Workaround: Deploy your LWC Component using Use LWC Standard Deployment? Salesforce Standard Deployment Model (Editor Guide #5).

Notes: Use the Salesforce standard deployment model when your code and all permissions are correct and still the error is occurring in the deployment.

> Worked with date time and javascript.

DAY 9:

- > Installed Salesforce CLI
- > Create Lightning web component
- > learned how to deploy and retrieve sources in salesforce org



- > Map and set in javascript
- > Map :it stores data in terms of key and value.
- > set : set of values without key and it only contains unique values.
- > map and set functions.
- > Operators in js
- > Difference between ==and ===
- > Strings in javascript
- > Objects in javascript
- > Functions in javascript
- > Arrow function ,console statements.
- > Example on LWC component.

DAY 10:

Lightning web component

- > Create College data table component (Apex 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 Data. The main area has tabs for 'CollegeDataTable.cls', 'collegeDataTable.html', 'collegeDataTable.js', and 'collegeDataTable.js-meta'. The code editor displays the following Apex class:

```

1  public class CollegeDataTable
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 }

```

A green success message at the top right says 'Success Apex Changes Deployed.'

- > Create College data table (html file)

Salesforce LWC Editor (Beta)

Search Apex or Component...

APEX CLASSES

- AnimalLocator
- AnimalLocatorMock
- AnimalLocatorTest
- ApplicationBatchTest
- applicationSchedule
- CollegeDataTable
- DataManager_Activity
- DataManager_ActivityTest
- DataManager_CleanUp
- DataManager_CleanUpTest
- DataManager_controller
- DataManager_controllerTest
- DataManager_Dataflow

LIGHTNING WEB COMPONENTS

- collegeDataTable
 - collegeDataTable.html
 - collegeDataTable.js
 - collegeDataTable.css
 - collegeDataTable.js-meta.xml

```

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

```

Success
All Changes Deployed.

Use LWC Standard Deployment? Twitter LinkedIn 10:52 PM 1/19/2022

> Create College data table (js file)

Salesforce LWC Editor (Beta)

Search Apex or Component...

APEX CLASSES

- AnimalLocator
- AnimalLocatorMock
- AnimalLocatorTest
- ApplicationBatchTest
- applicationSchedule
- CollegeDataTable
- DataManager_Activity
- DataManager_ActivityTest
- DataManager_CleanUp
- DataManager_CleanUpTest
- DataManager_controller
- DataManager_controllerTest
- DataManager_Dataflow

LIGHTNING WEB COMPONENTS

- collegeDataTable
 - collegeDataTable.html
 - collegeDataTable.js
 - collegeDataTable.css
 - collegeDataTable.js-meta.xml

```

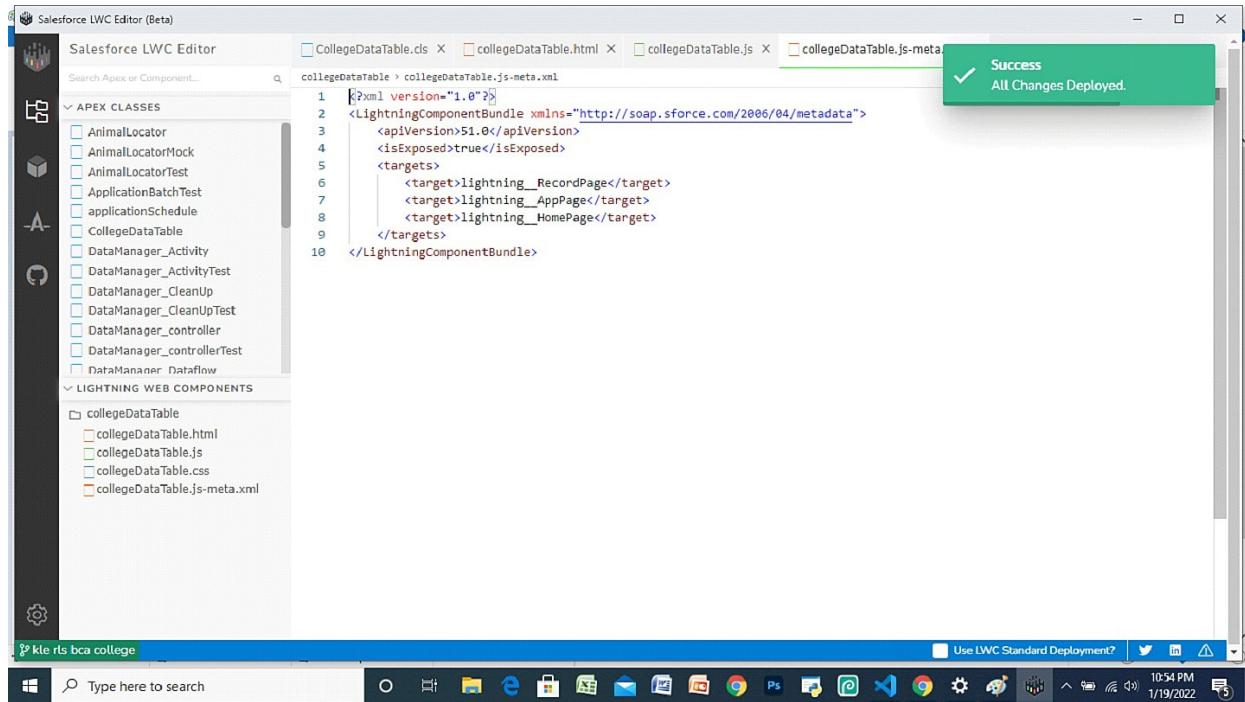
import { LightningElement, api, wire } from 'lwc';
import getapplicationvalues from '@salesforce/apex/GetapplicationDetails.getapplicationvalues';
export default class CollegeDataTable extends LightningElement {
  columnsList = [
    {label : 'Application Form' , fieldName : 'Name', type:'text' },
 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

```

Success
All Changes Deployed.

Use LWC Standard Deployment? Twitter LinkedIn 10:53 PM 1/19/2022

>Create College data table (meta-xml file)



> Output for College data table

The screenshot shows a web browser window with multiple tabs open. The main content area displays a 'College Management' application. At the top, there's a navigation bar with 'College' and 'MIT-CCU'. Below it is a search bar and a toolbar with icons for New Contact, Edit, and New Case. The main area features a 'Datatable' section with columns for Application Form, College Fees, Date of birth, Email, Hostel Fees, and Student Name. The data table contains 6 rows of student information. To the right of the datatable is an 'Activity' sidebar with sections for 'New Event', 'New Task', and 'Log a Call'. It also includes a 'Filters' dropdown and a 'Upcoming & Overdue' section.

Application Form	College Fees	Date of birth	Email	Hostel Fees	Student Name
1 F-000002	\$140,000.00	Dec 31, 2021	vaish@ggg.com	13000	vaishnavi k
2 F-000004	\$140,000.00	Feb 26, 2001	gfrdrfj@gmail.com	13000	ross ja
3 F-000009	\$140,000.00	Jan 4, 2022	dfrgfgt@gmail.c...	13000	jj
4 F-000015	\$140,000.00	Jan 10, 2022	ccccccccc@ggg.c...	13000	w
5 F-000016	\$140,000.00	May 6, 2003	gg@gmail.com	13000	jim
6 F-000017	\$140,000.00	Jan 7, 1998	jaya@jaya.com	13000	jaya

> Communication between parent to child and child to parent

DAY 11:

Life Cycle Hooks in LWC

- > Constructor(), Connectedcallback(),
 - constructor: fires when component instance is created.
 - flows from parents to child component.
 - structure of constructor().
- > render(), rendercallback()
- > disconnectedcallback(), errorcallback()

- > Toast messages
 - Indicates/inform user that they are writing something wrong
- > Four types of Toast messages
 - Success (in green colour)
 - Warning (in orange/yellow colour)
 - info (in grey colour)
 - error (in red colour)
- > sample examples on toast messages

- > Doubt clearing session