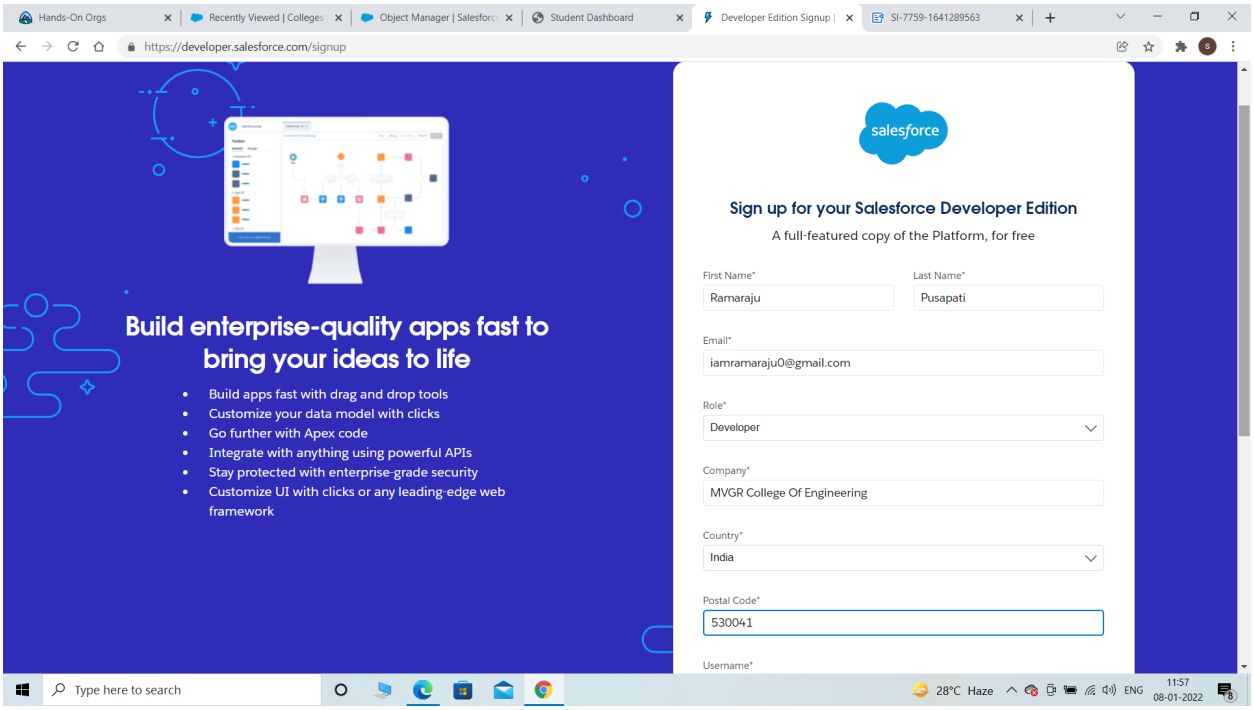
Day 1
Topic:Creating Salesforce development account and activating it.
Activity:
Task1:Creating a salesforce developer account
The following steps are involved in creating the salesforce developer account
Creating a developer org in salesforce.
1.Go to developers.salesforce.com/
2.Click on sign up.
3.On the sign up form, enter the following details :
4.First name & Last name
5.Email
6.Role : Developer
7.Company : College Name
8.County : India
9.Postal Code : pin code
10.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



Task2:verifying the account through email verification
You will be receiving a mail from salesforce developer org with a click on button "verify your email" click on it.After clicking on it you will be redirected to the salesforce login page where you will be asked your login credentials to proceed further.

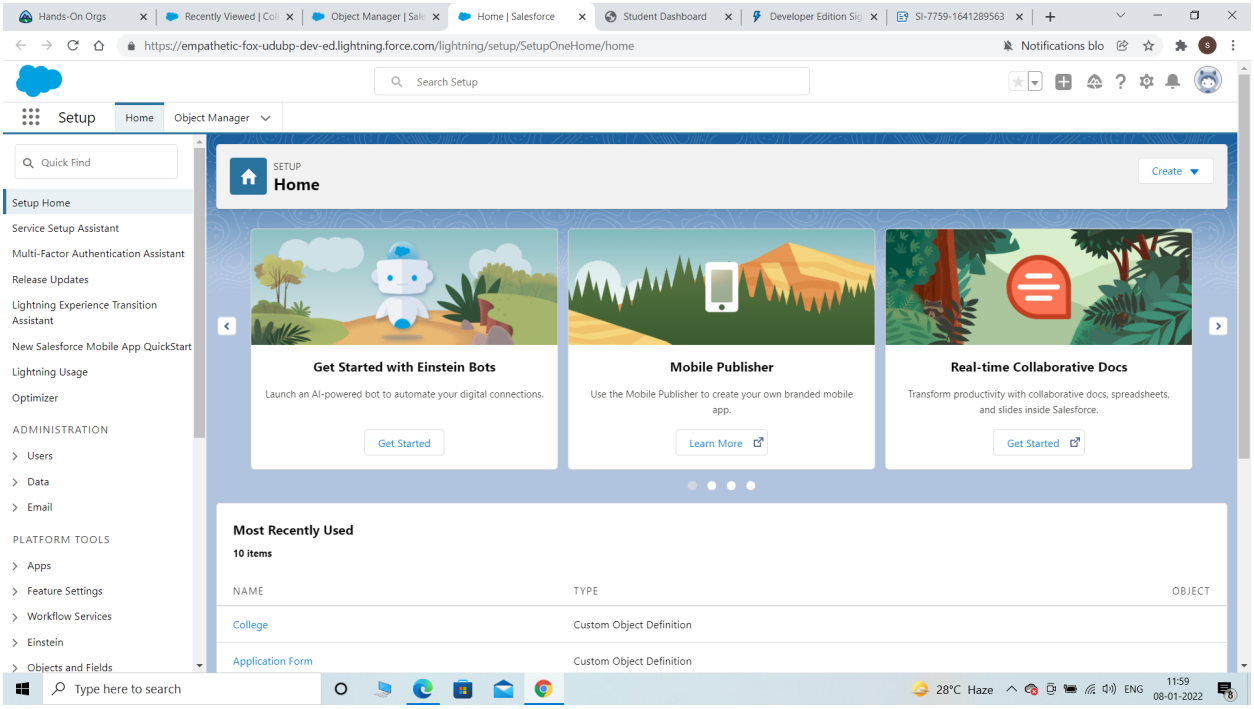
Task 3Login into your salesforce developer account

The steps that are needed to followed are:

1.Go to 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.



Day 2:

Topic:Costum object creation

Activities:

Task1:Create College management Application

The steps involved in this are

1.Click on the settings icon on the salesforce home page this will show two options

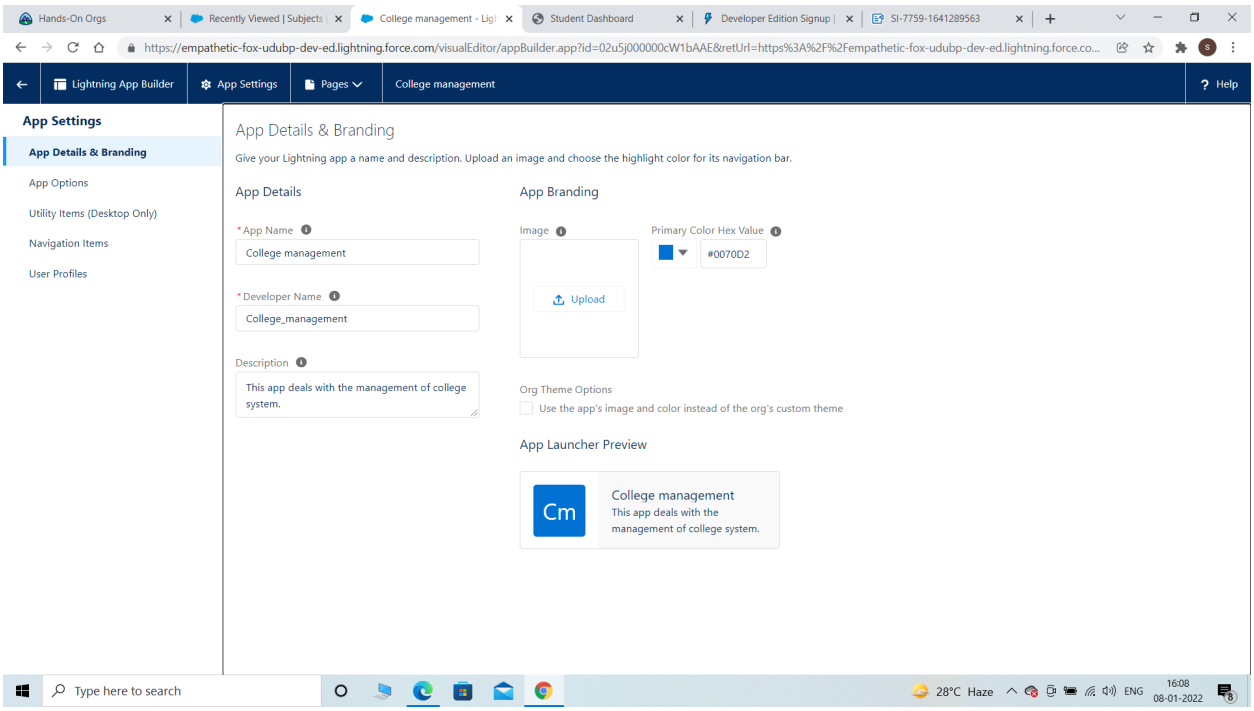
2.Click on setup

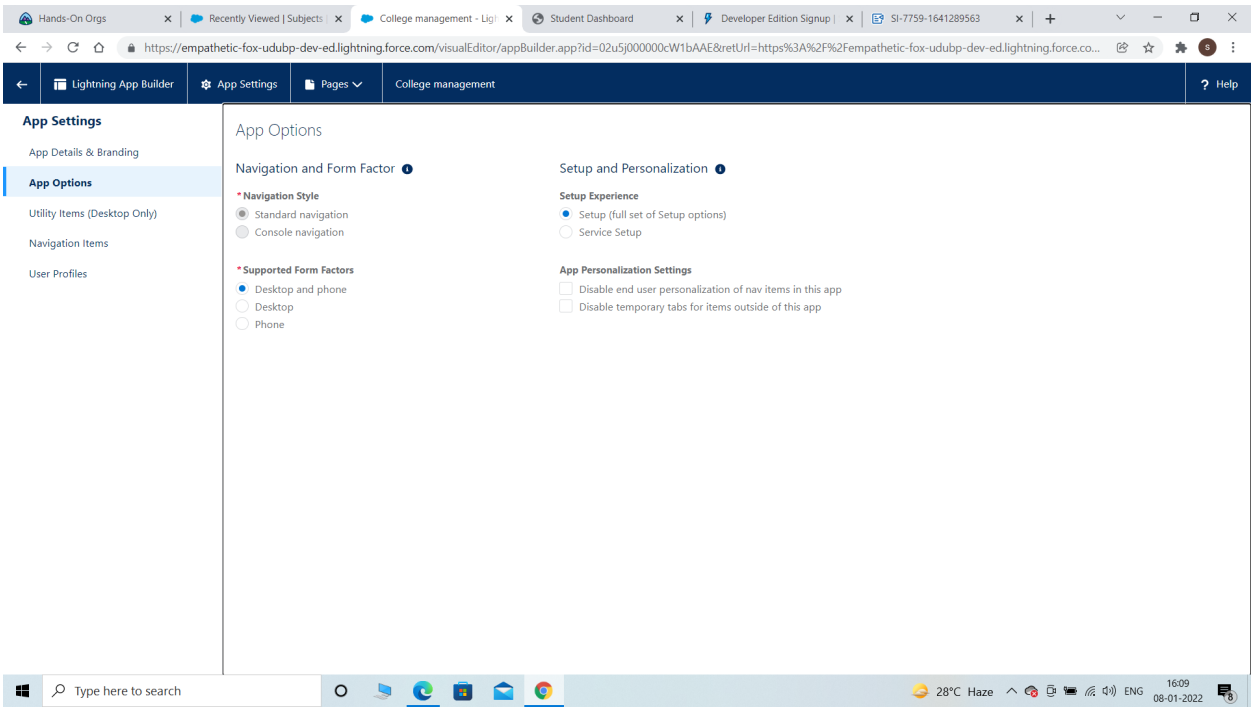
3.This will open the salesforce home page

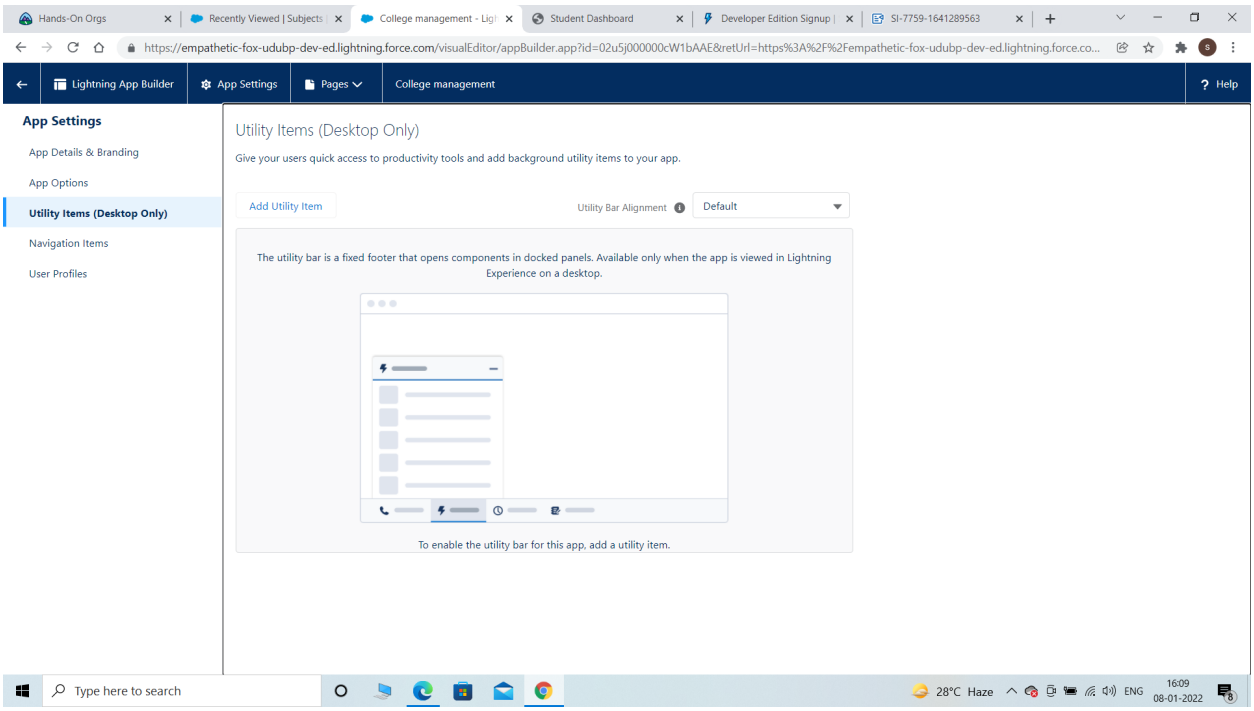
4.Now,search for app manager and click on it

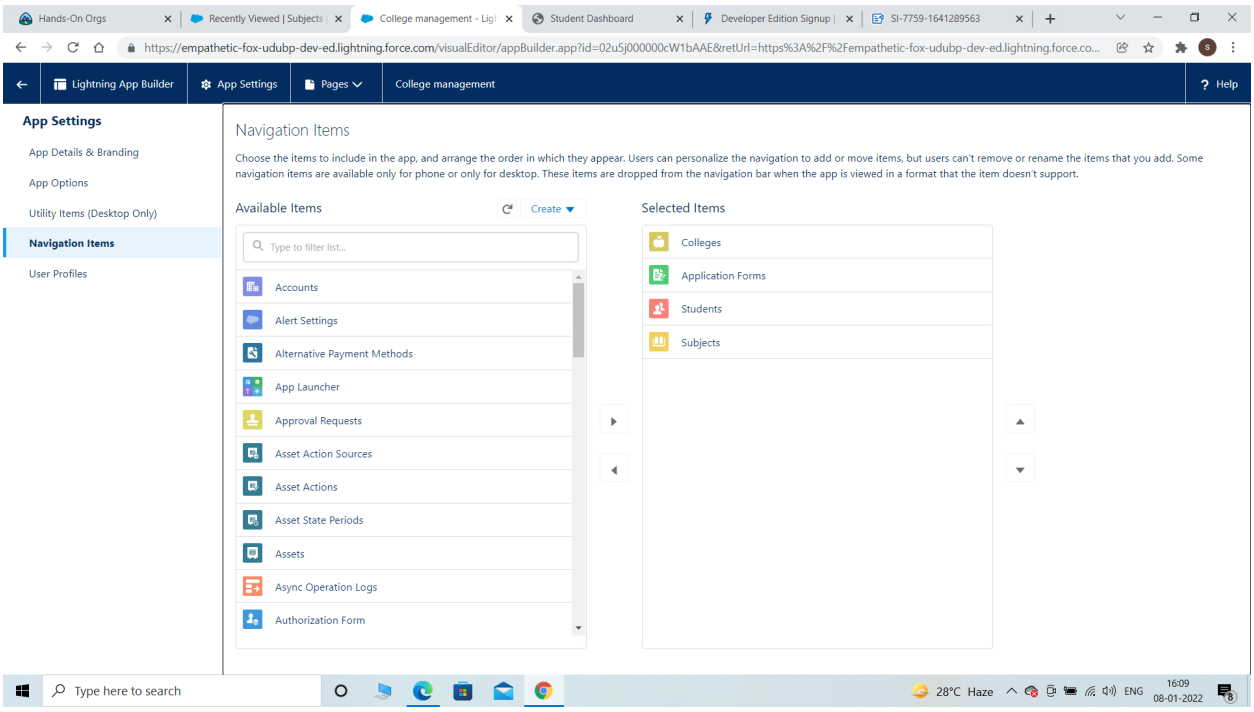
5.Click on new lightening app

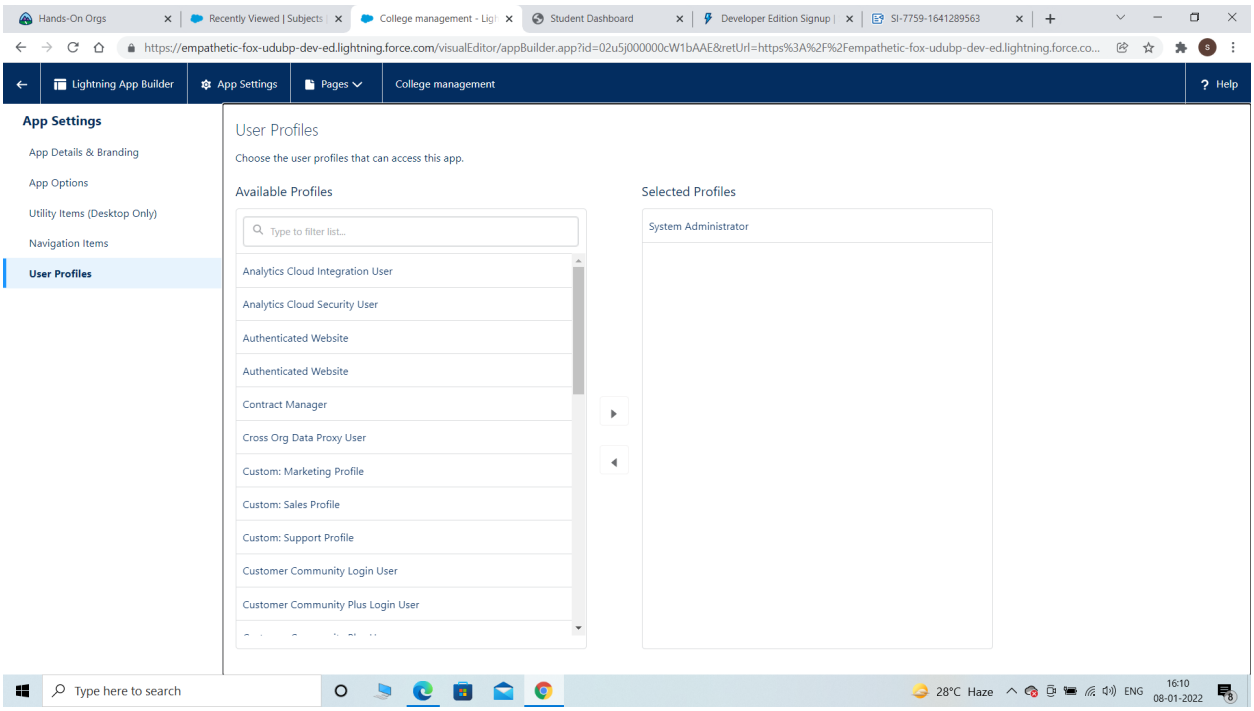
6.Fill the app details and click next,In the same way fill the app options,navigation items,user profiles and then click on save this creates the new lightening app called College management application


Hands-On Orgs
Recently Viewed | Subjects
College management - Ligh
Student Dashboard
Developer Edition Signup
SI-7759-1641289563
https://empathetic-fox-udubp-dev-ed.lightning.force.com/visualEditor/appBuilder.app?id=02u5j000000cW1bAAE&retUrl=https%3A%2F%2Fempathetic-fox-udubp-dev-ed.lightning.force.co...
Lightning App Builder
App Settings
Pages
College management
Help
App Settings
App Details & Branding
App Options
Utility Items (Desktop Only)
Navigation Items
User Profiles
App Details & Branding
Give your Lightning app a name and description. Upload an image and choose the highlight color for its navigation bar.
App Details
*App Name
College management
*Developer Name
College_management
Description
This app deals with the management of college system.
App Branding
Image
Upload
Primary Color Hex Value
#0070D2
Org Theme Options
Use the app's image and color instead of the org's custom theme
App Launcher Preview
Cm
College management
This app deals with the management of college system.
Type here to search
28°C Haze
ENG
16:08
08-01-2022


Hands-On Orgs
Recently Viewed | Subjects
College management - Ligh
Student Dashboard
Developer Edition Signup
SI-7759-1641289563
https://empathetic-fox-udubp-dev-ed.lightning.force.com/visualEditor/appBuilder.app?id=02u5j000000cW1bAAE&retUrl=https%3A%2F%2Fempathetic-fox-udubp-dev-ed.lightning.force.co...
Lightning App Builder
App Settings
Pages
College management
Help
App Settings
App Details & Branding
App Options
Utility Items (Desktop Only)
Navigation Items
User Profiles
App Options
Navigation and Form Factor
*Navigation Style
Standard navigation
Console navigation
*Supported Form Factors
Desktop and phone
Desktop
Phone
Setup and Personalization
Setup Experience
Setup (full set of Setup options)
Service Setup
App Personalization Settings
Disable end user personalization of nav items in this app
Disable temporary tabs for items outside of this app
Type here to search
28°C Haze
ENG
16:09
08-01-2022


Hands-On Orgs
Recently Viewed | Subjects
College management - Ligh
Student Dashboard
Developer Edition Signup
SI-7759-1641289563
https://empathetic-fox-udubp-dev-ed.lightning.force.com/visualEditor/appBuilder.app?id=02u5j000000cW1bAAE&retUrl=https%3A%2F%2Fempathetic-fox-udubp-dev-ed.lightning.force.co...
Lightning App Builder
App Settings
Pages
College management
Help
App Settings
App Details & Branding
App Options
Utility Items (Desktop Only)
Navigation Items
User Profiles
Utility Items (Desktop Only)
Give your users quick access to productivity tools and add background utility items to your app.
Add Utility Item
Utility Bar Alignment
Default
The utility bar is a fixed footer that opens components in docked panels. Available only when the app is viewed in Lightning Experience on a desktop.
To enable the utility bar for this app, add a utility item.
Type here to search
28°C Haze
ENG
16:09
08-01-2022


Hands-On Orgs
Recently Viewed | Subjects
College management - Ligh
Student Dashboard
Developer Edition Signup
SI-7759-1641289563
https://empathetic-fox-udubp-dev-ed.lightning.force.com/visualEditor/appBuilder.app?id=02u5j000000cW1bAAE&retUrl=https%3A%2F%2Fempathetic-fox-udubp-dev-ed.lightning.force.co...
Lightning App Builder
App Settings
Pages
College management
Help
App Settings
App Details & Branding
App Options
Utility Items (Desktop Only)
Navigation Items
User Profiles
Navigation Items
Choose the items to include in the app, and arrange the order in which they appear. Users can personalize the navigation to add or move items, but users can't remove or rename the items that you add. Some navigation items are available only for phone or only for desktop. These items are dropped from the navigation bar when the app is viewed in a format that the item doesn't support.
Available Items
Create
Type to filter list...
Accounts
Alert Settings
Alternative Payment Methods
App Launcher
Approval Requests
Asset Action Sources
Asset Actions
Asset State Periods
Assets
Async Operation Logs
Authorization Form
Selected Items
Colleges
Application Forms
Students
Subjects
Type here to search
28°C Haze
ENG
16:09
08-01-2022



Task2:Create the custom objects for the college management application

The steps involved in creating custom objects are

1.Click on setup

2.Click on object manager

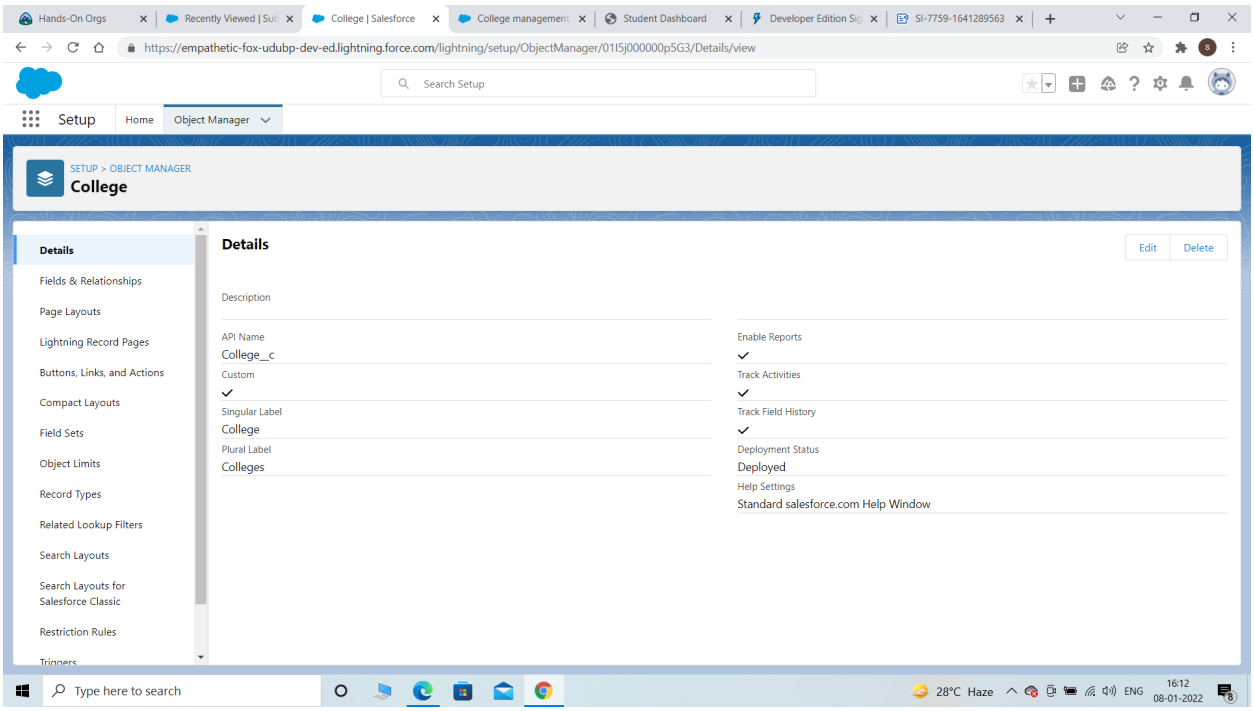
3.Click on create new custom object

4.Select the label and fill in the necessary details and click on save

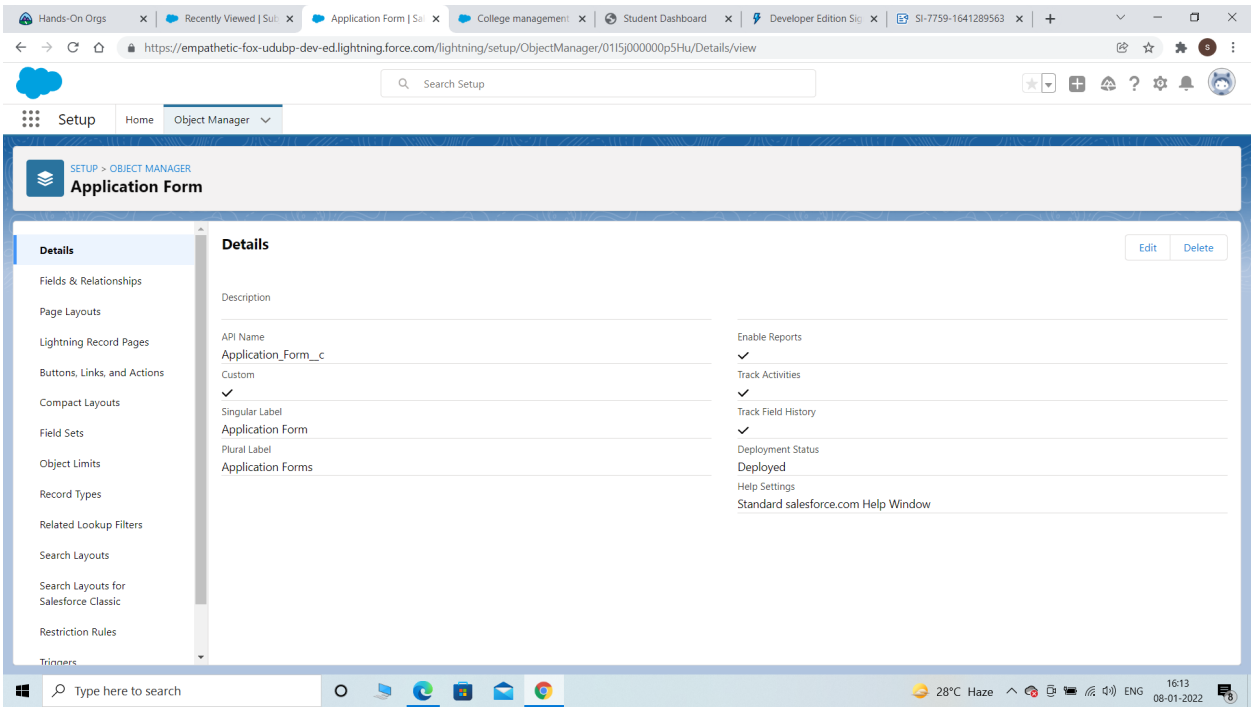
then select a icon for the object and click on save and next to create new objects.

The objects that are needed to be created for college management application are

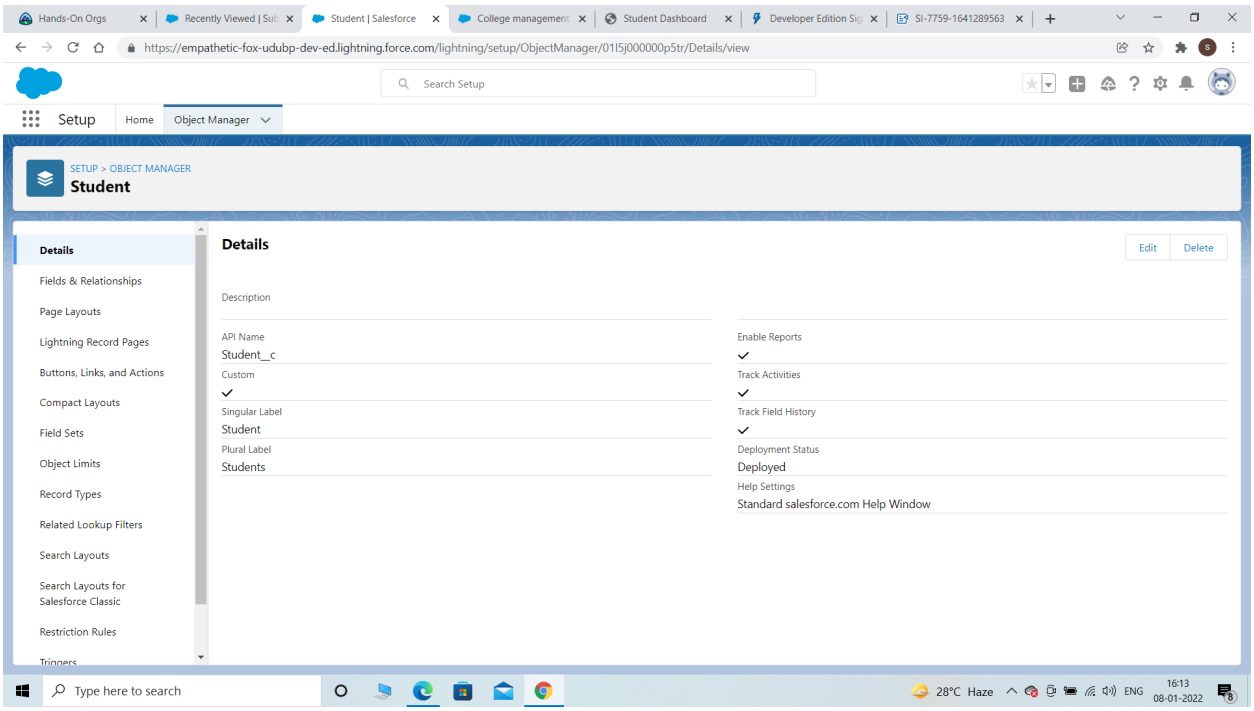
1.College



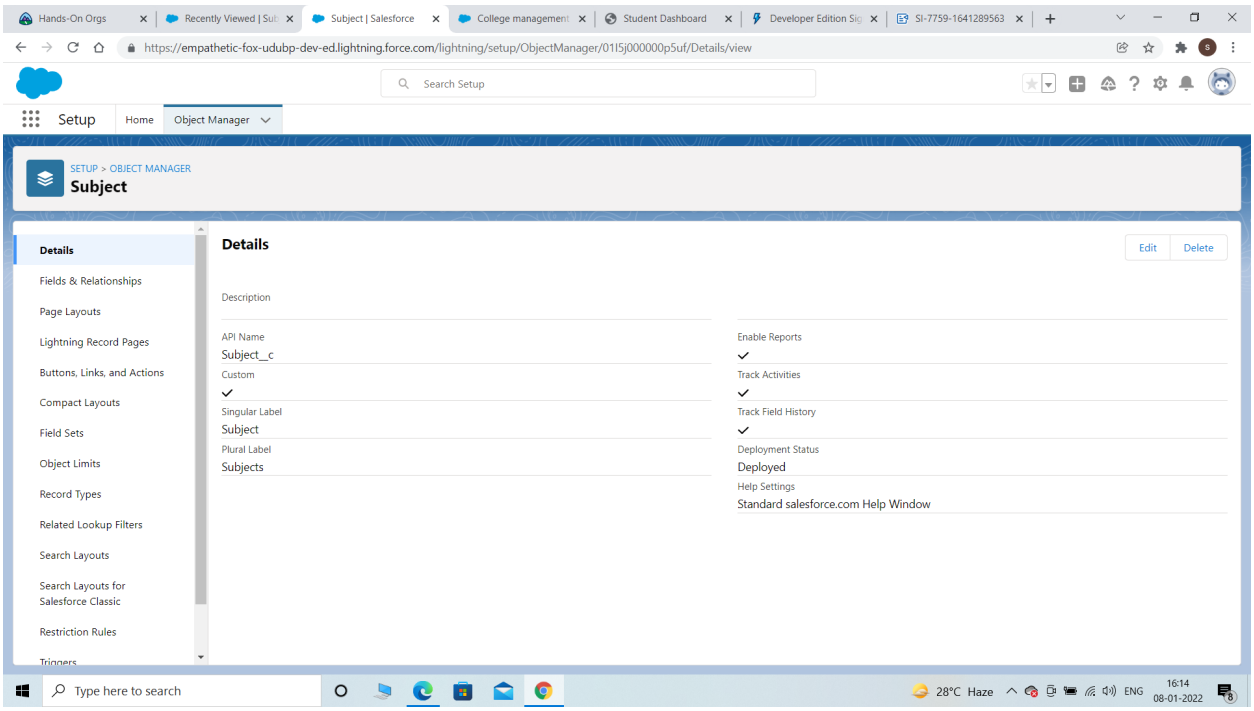
2.Application Form



3.Student



4.Subject



Task 3:Creating the fields on each of the above objects

The process to create fields is as follows

1.click on object manager

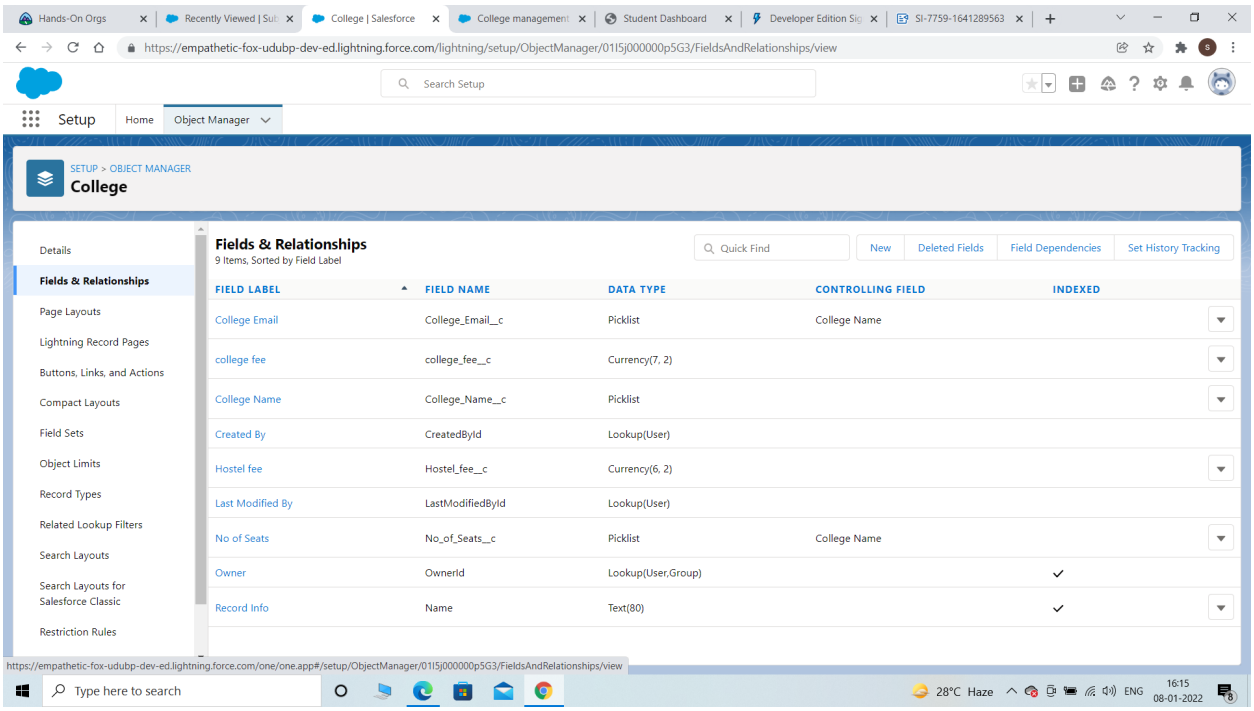
2.Search the required object in the search bar

3.click on the object

4.click on fields and relationships

5.Click on new
6.select the data type for the new field and click on next then enter the field label and save
The objects with their necessary fields that are required to be created for the college management application are:
1.College
The fields that are required in the college object are
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

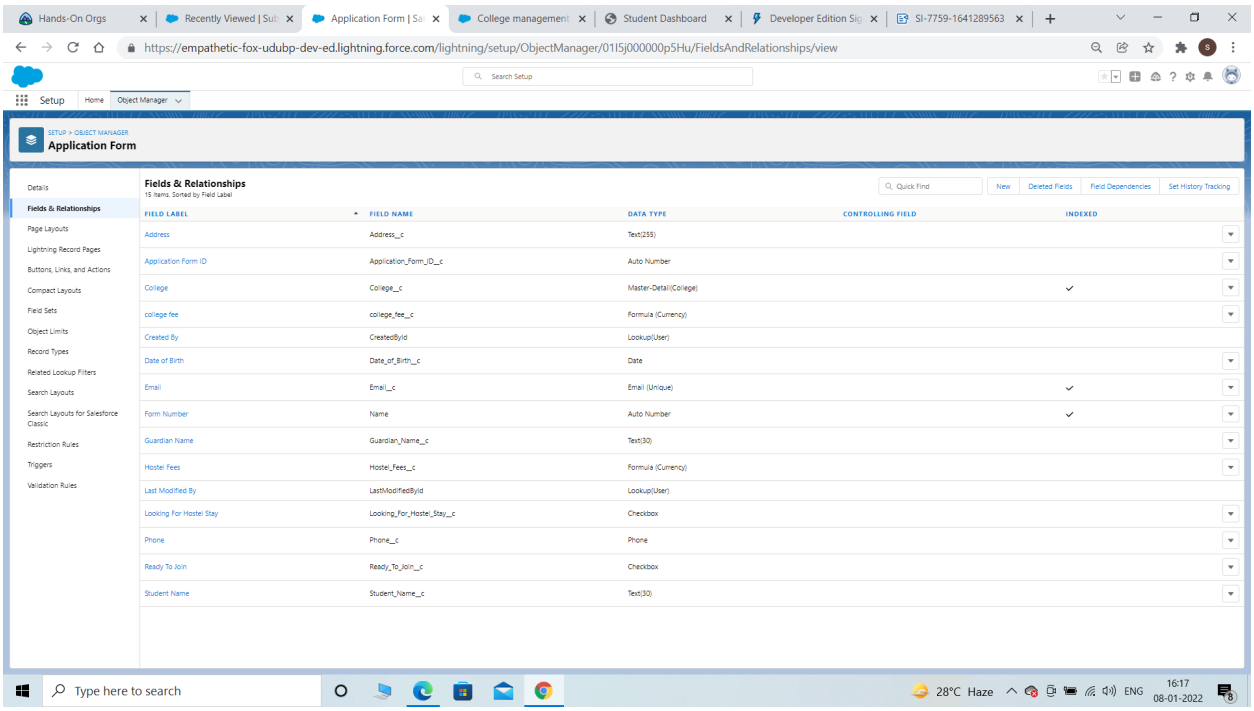


2.Application Form

The fields that are needed to be created on the application form object are:

Create the following fields on the application form object.

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		Yes	

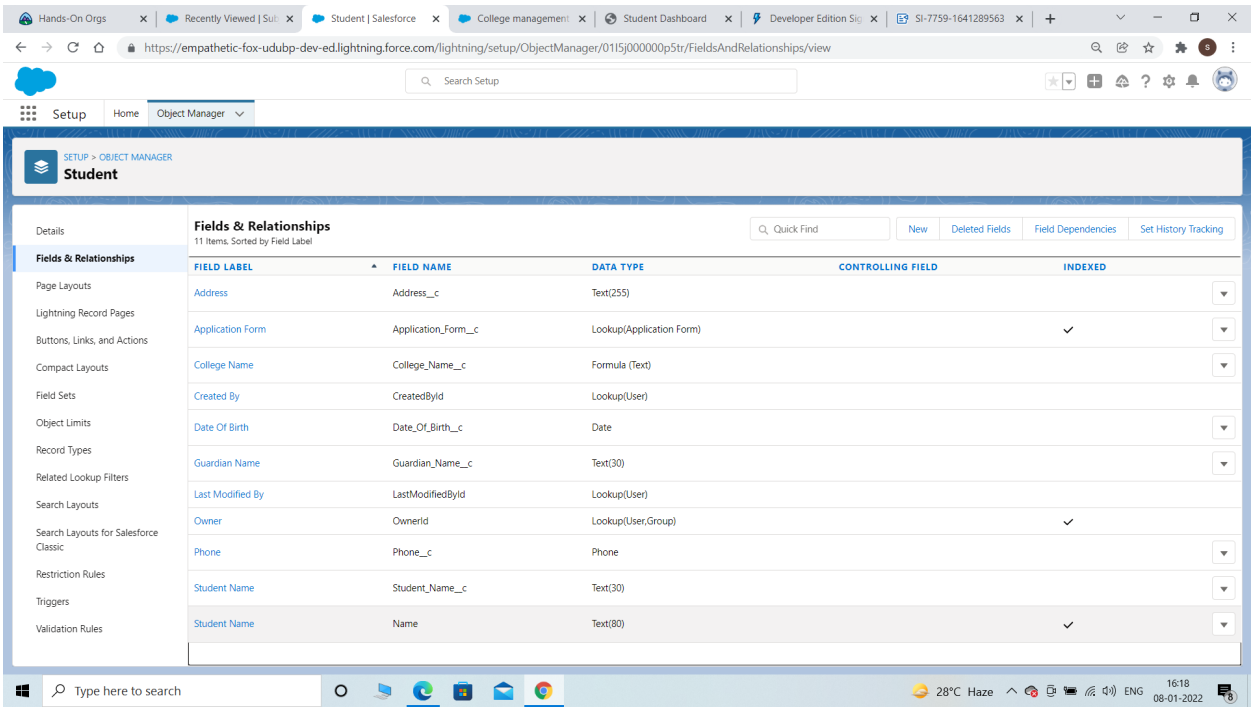


3.Student

The fields that are needed to be created on the student object are:

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	

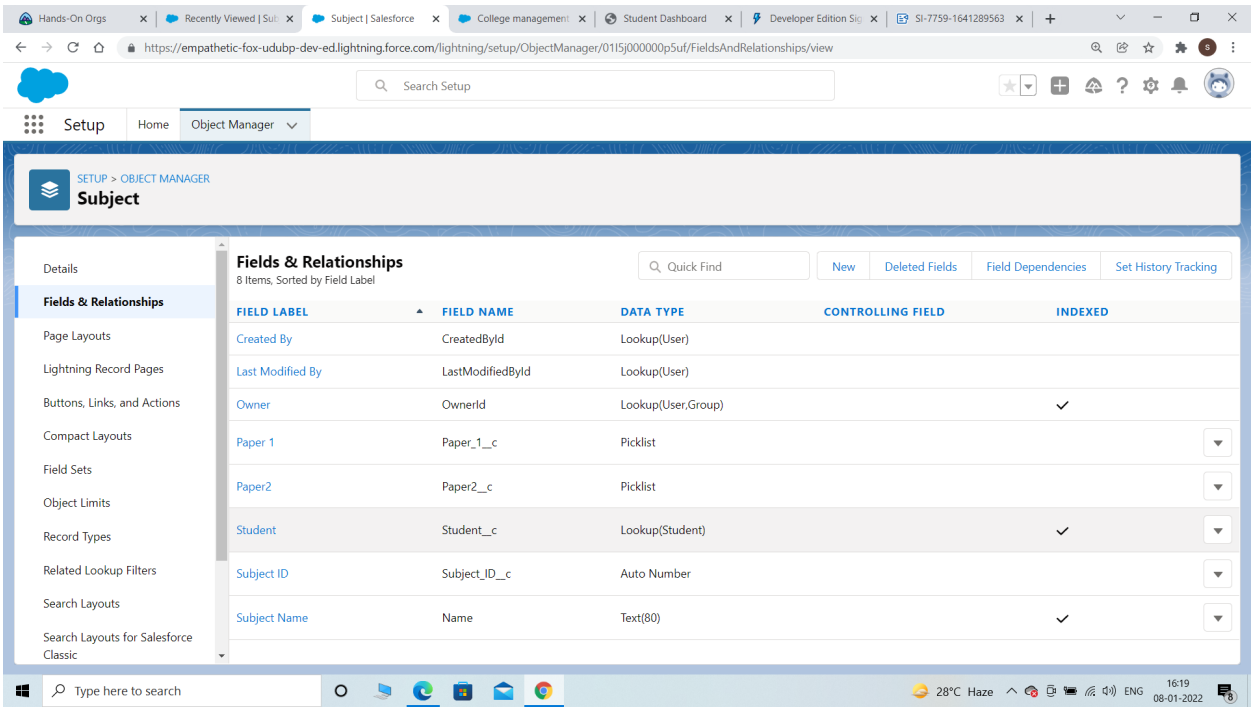


4.Subjects

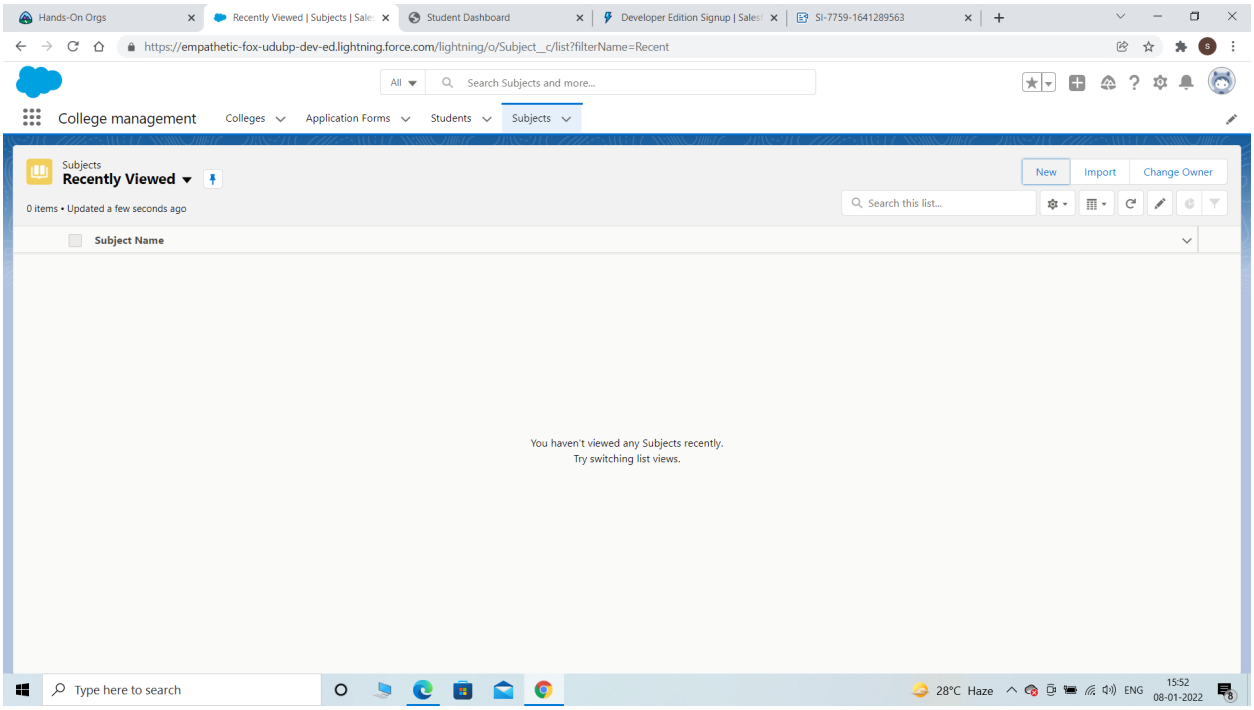
The fields that are required to be created on the subjects object are:

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)		



After creation of the objects and their necessary fields the college management application look as follows:



Day 3:

Topic:Adding Business Logic To Application

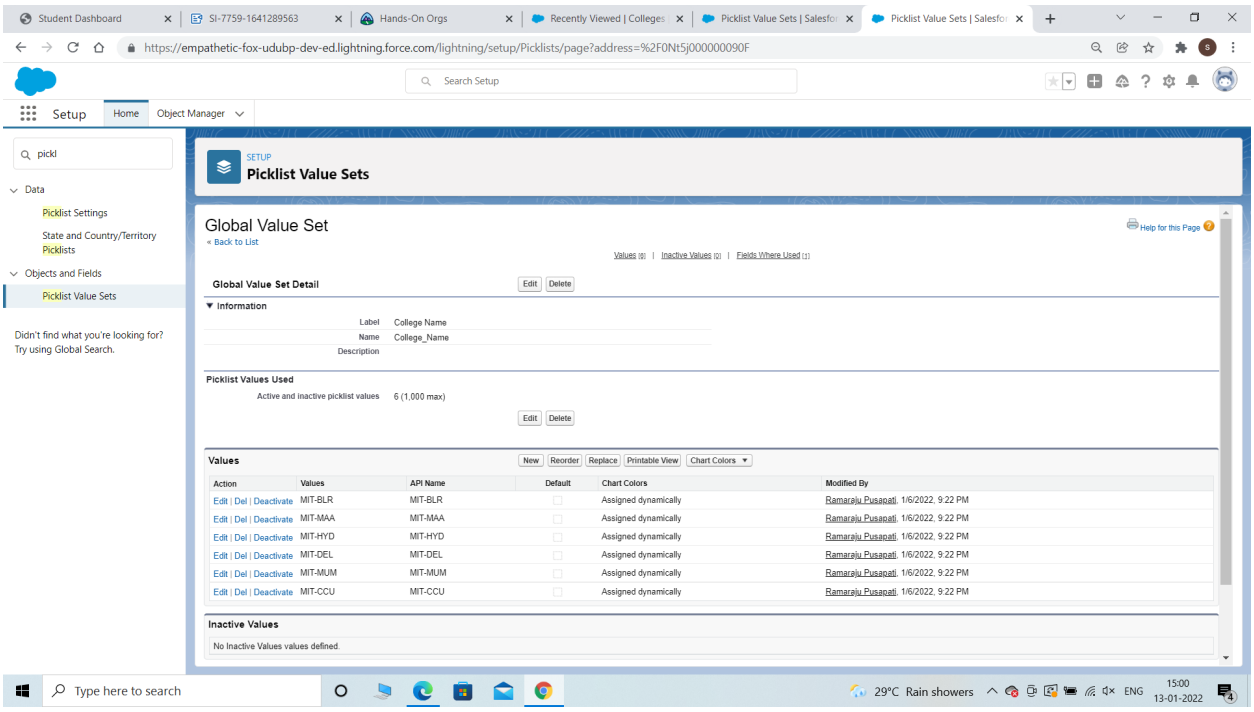
Task1:Creating global picklist value sets

The process for creating global picklist value sets is as follows:

1.Click on home in setup

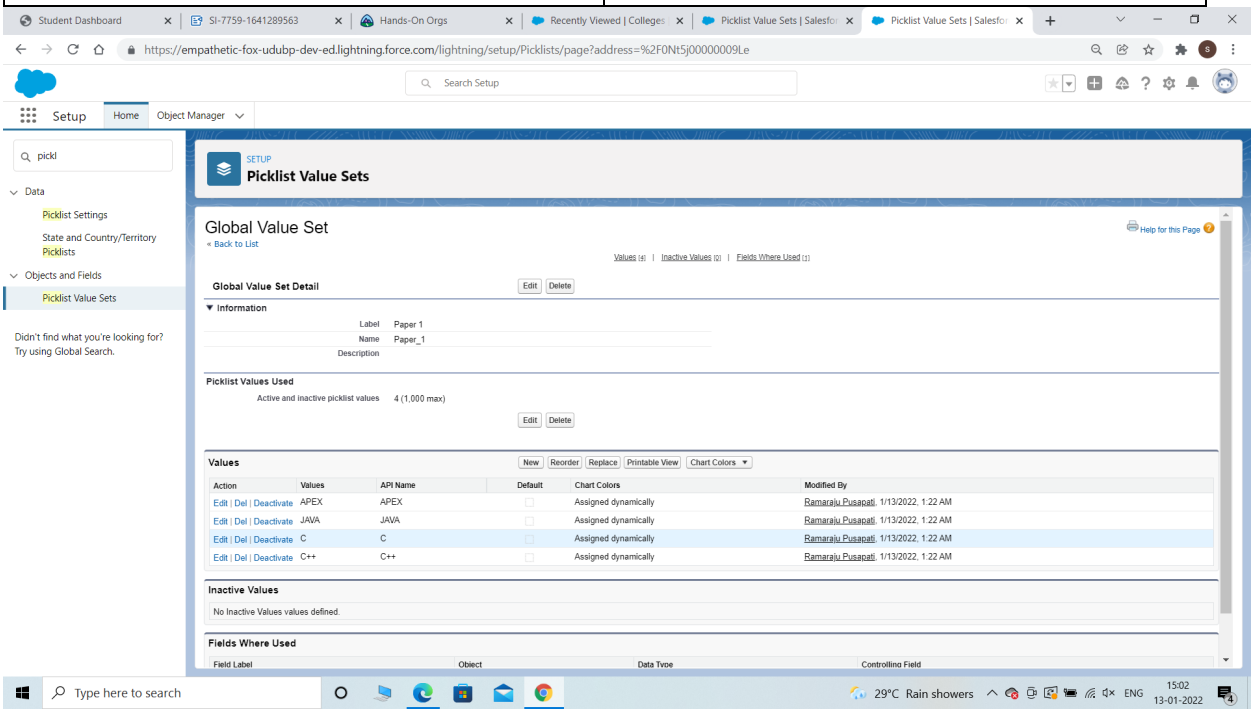
2.search for picklist value sets in the search bar and click on it
3.click on new
4.enter the label and values to be displayed in the picklist
5.click on save
6.while creating the field for picklist select global value sets option and select the required value set
1. Create the following global picklist value sets for the application.
 a)College

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



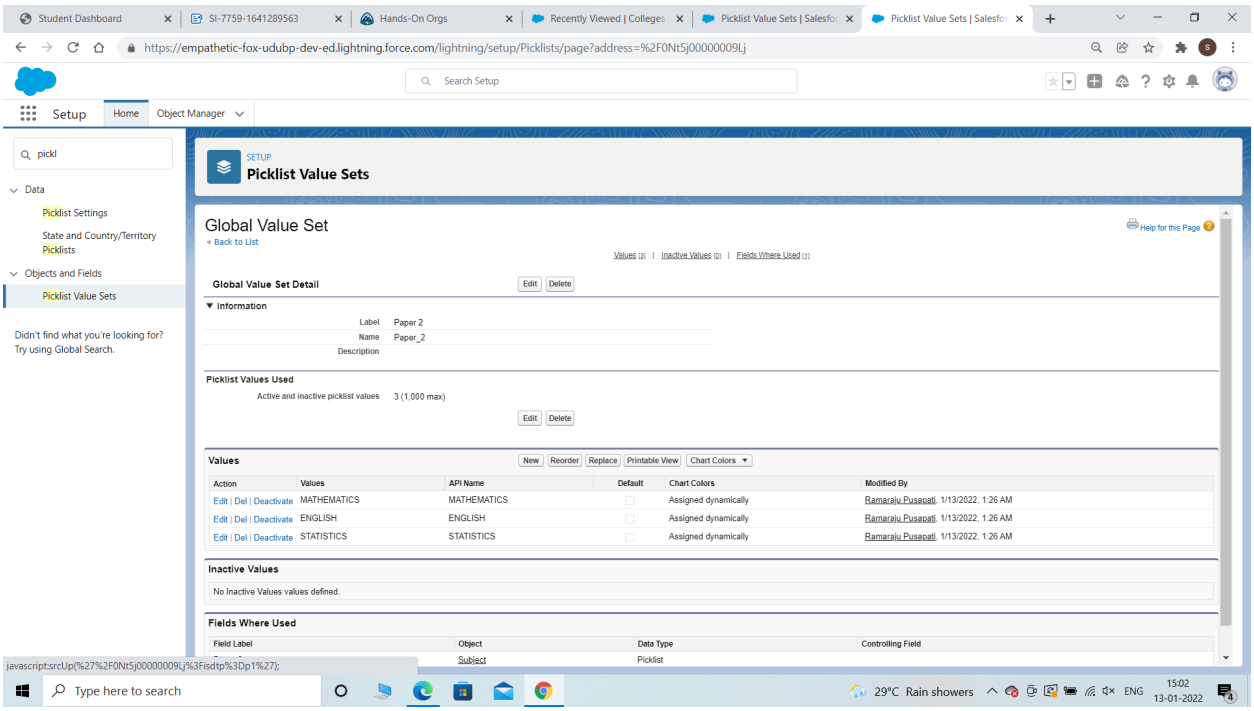
b)Paper1

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



c)Paper2

Picklist Value Name	Values
Paper2	MATHEMATICS ENGLISH STATISTICS



Task2:Creating Field Dependencies

The process to create field dependencies is as follows

1.Go to object manager

2.click on college object

3.click on fields and relationships

4.click on field dependencies

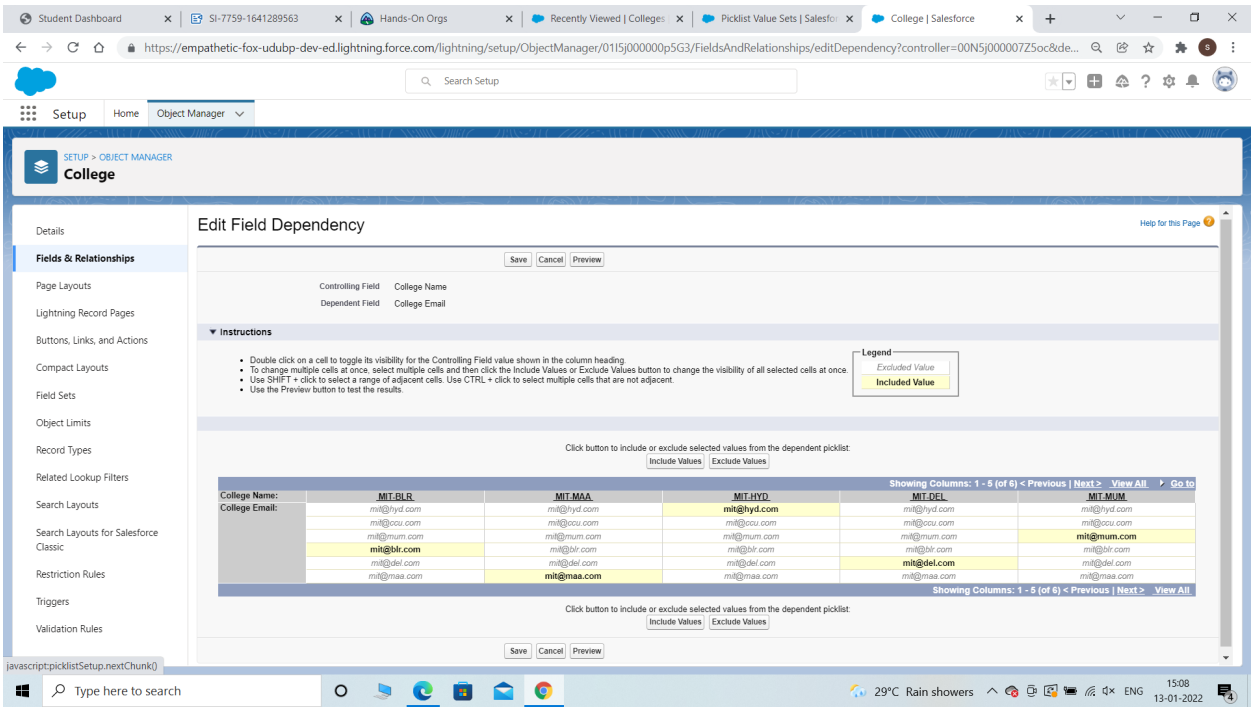
5.click on new

6.select the controlling field and the dependent field

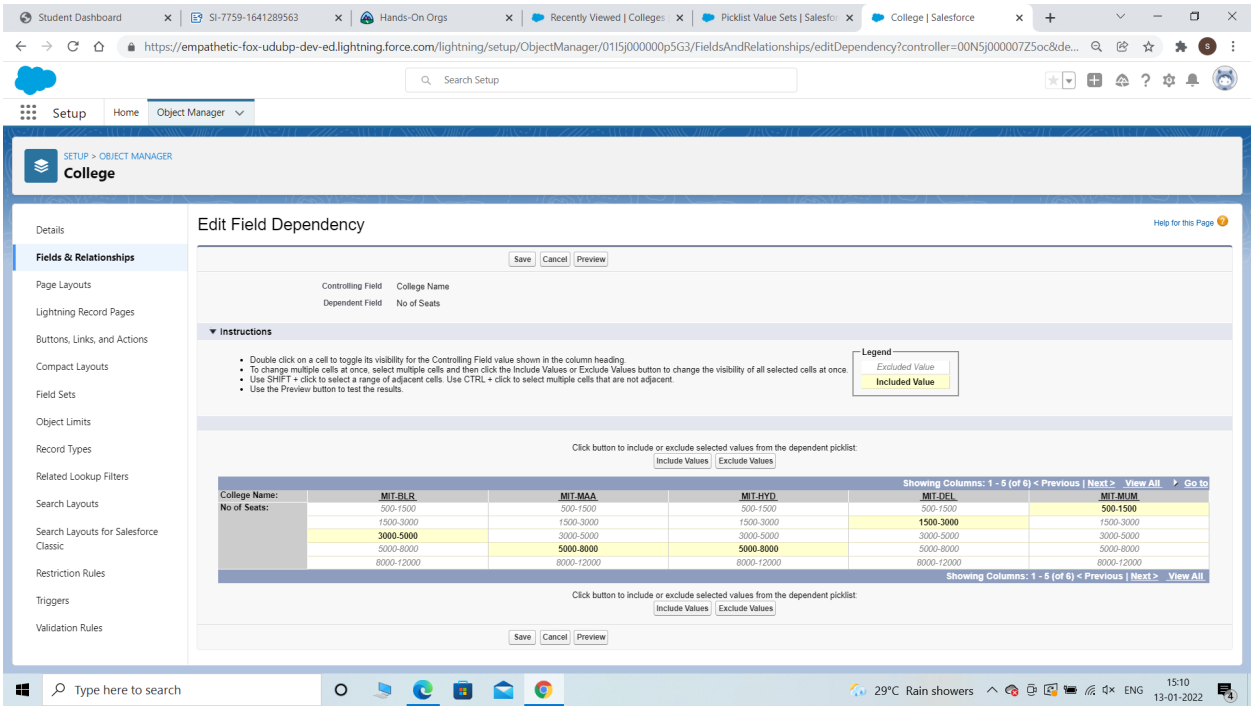
7.select the values to be included and save it

The field dependencies that we need to create are:

1)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.



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



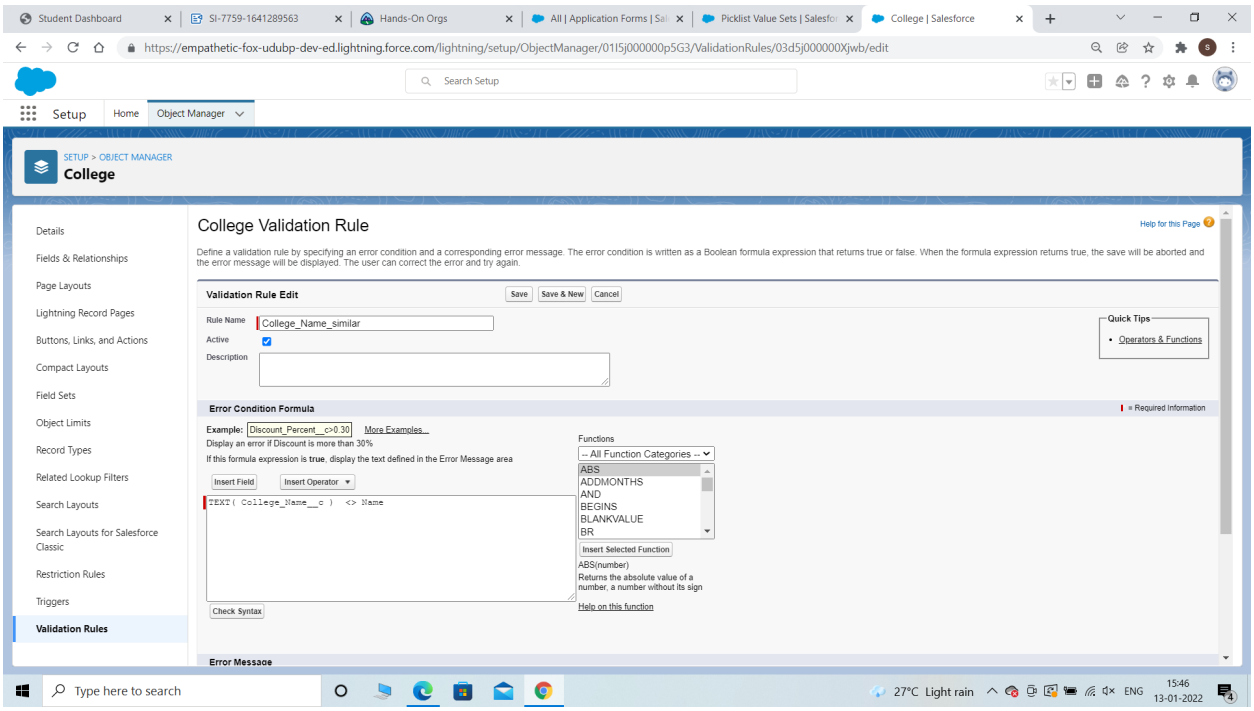
Task3:Creating validation rules on objects

The steps required to follow for creation of validation rules are:

1.Click on object manager

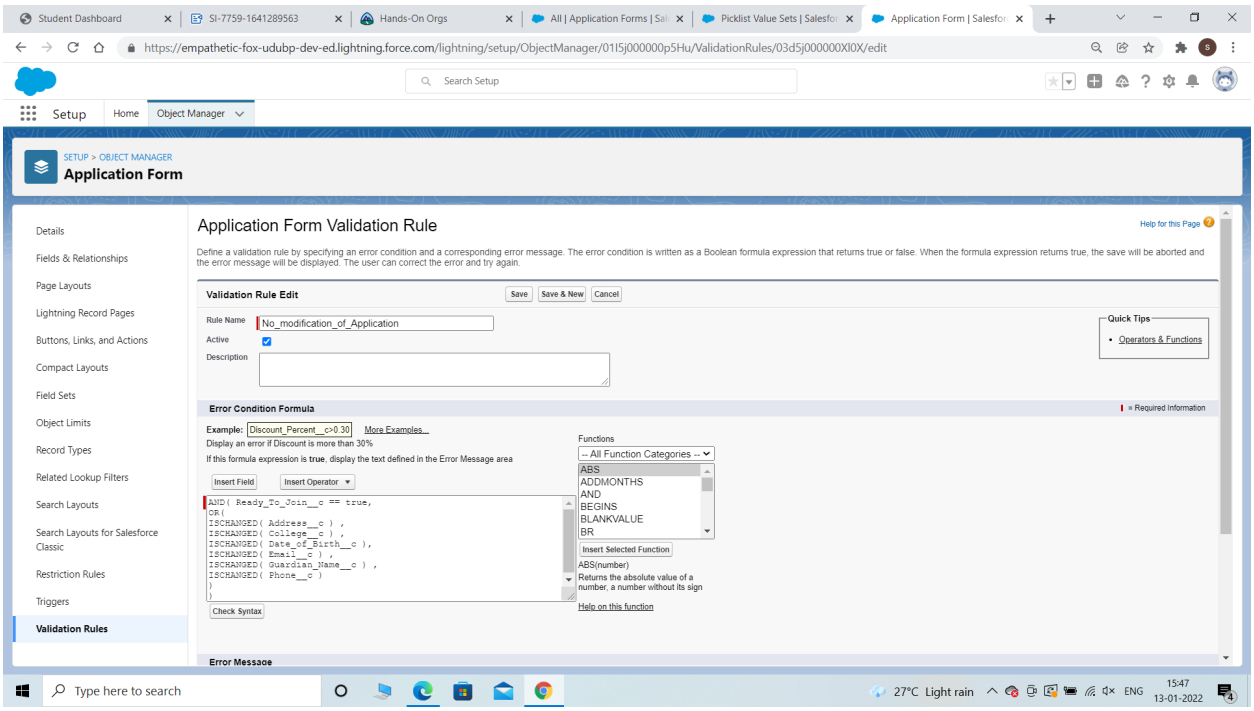
2.go to the required object
3.click on validation rules
4.click on new
5.add the rule and save it
The objects on which the validation rules have to be created are:
1) Create a validation rule on the college object such that the college name and record info should have the same name.

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( Date_Of_Birth__c ),
ISCHANGED( Email__c ) ,
ISCHANGED( Guardian_Name__c ) ,
ISCHANGED( Phone__c )
)
)
```



Task4:Process Automation

Steps required for process Automation are:

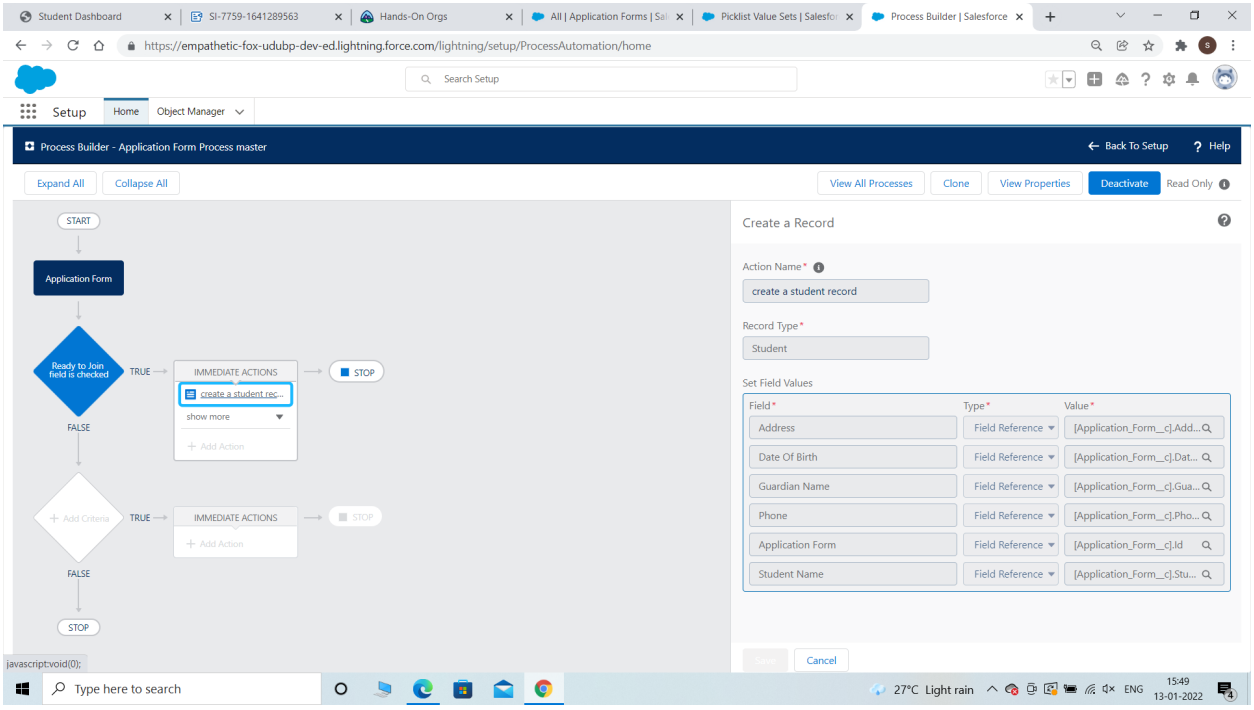
1.click on the home

2.search for process builder

3.click on new

4.enter the necessary fields

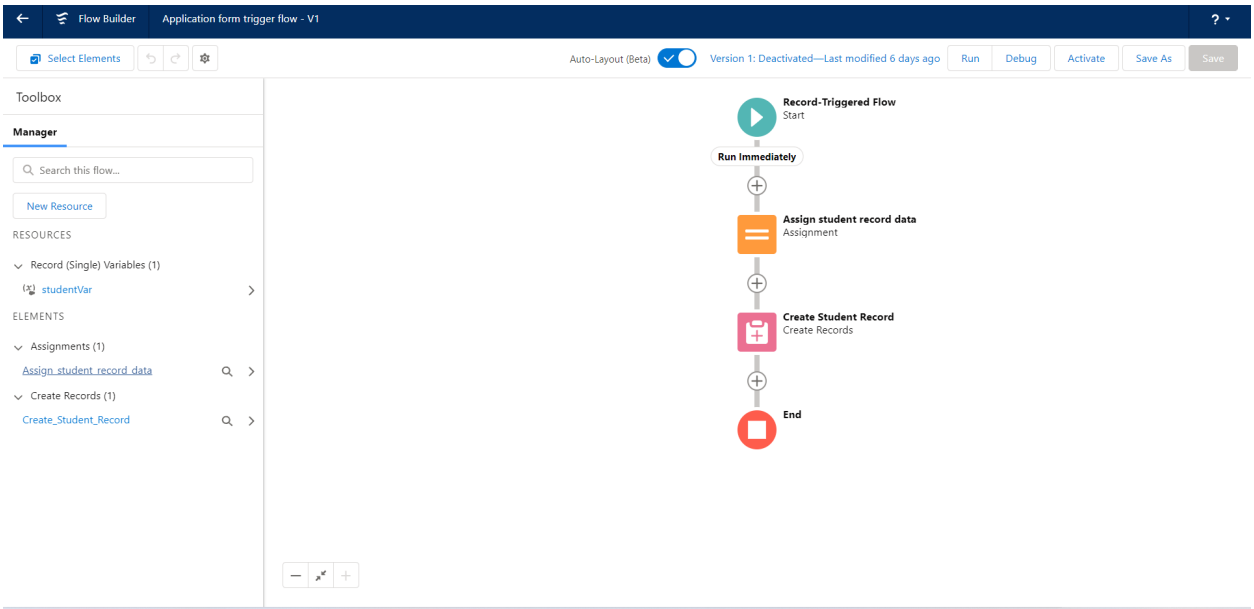
5.click on activate then save

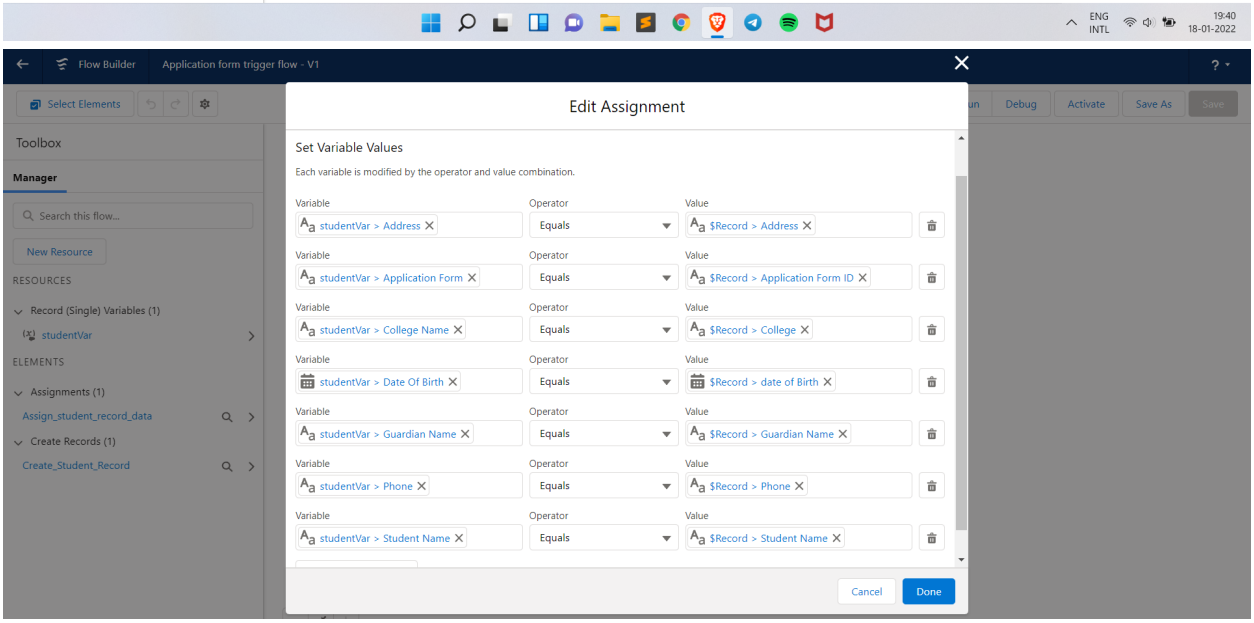


Task5:Create the student record using flow

The steps required to build the flow are as follows:

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.





Day 6

Topic: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
   }
}
```

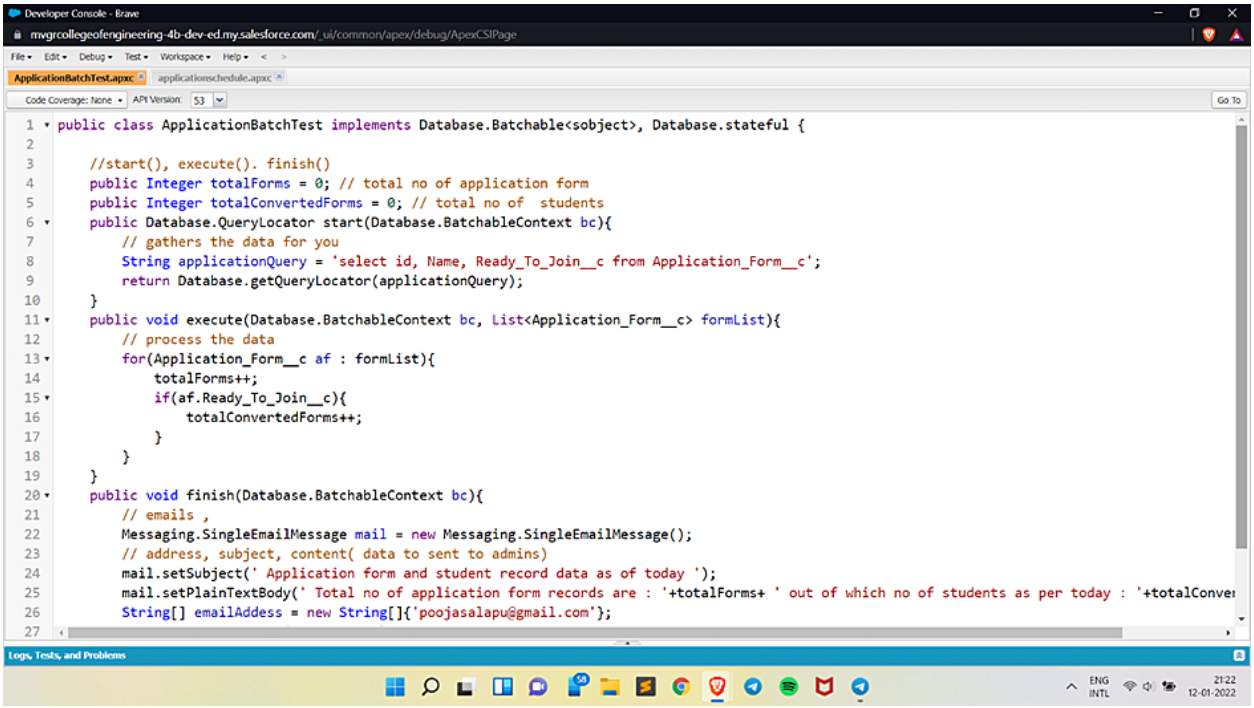
Task1:Create a batch apex for Application form

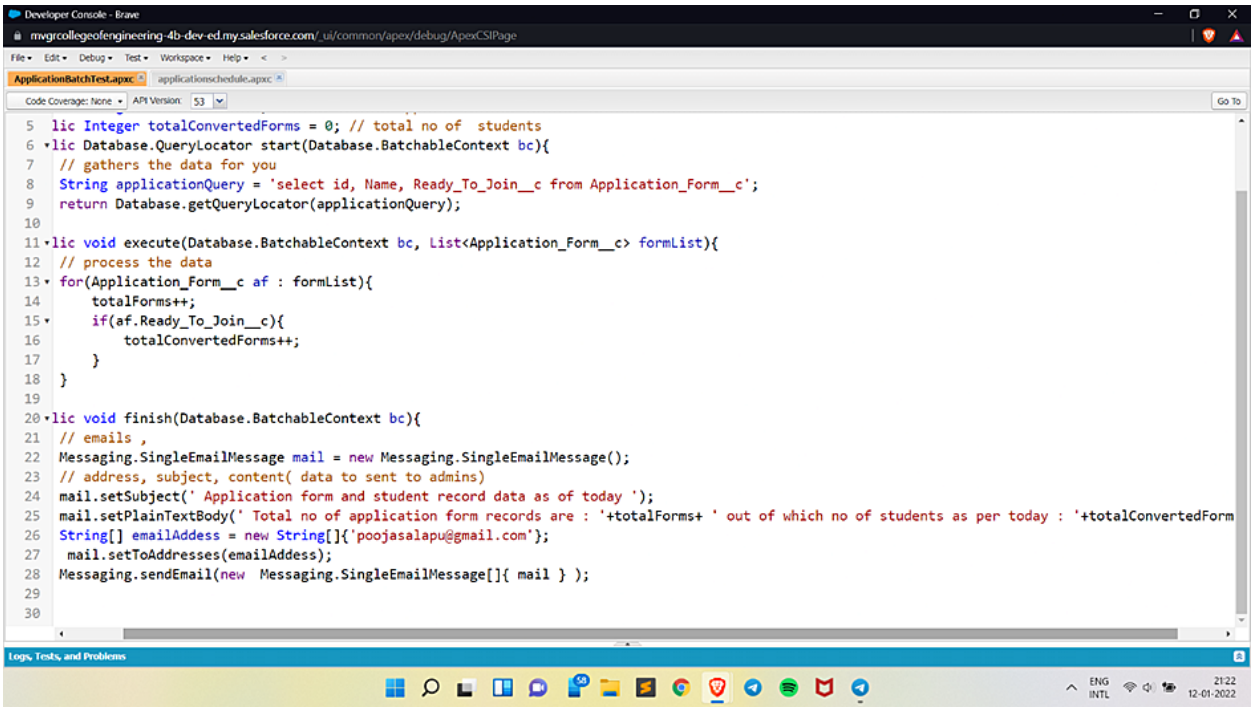
For this you need to create a apex class and do the following

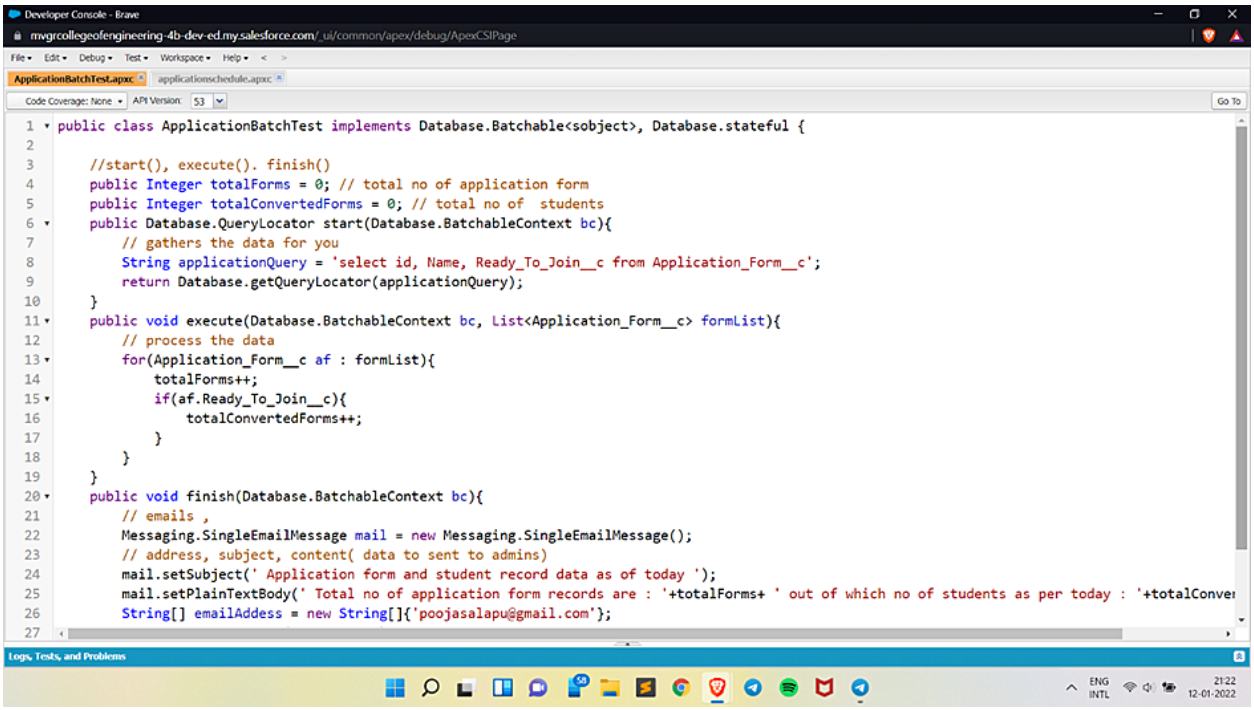
1. From the developer console create a new apex class and enter the following code.

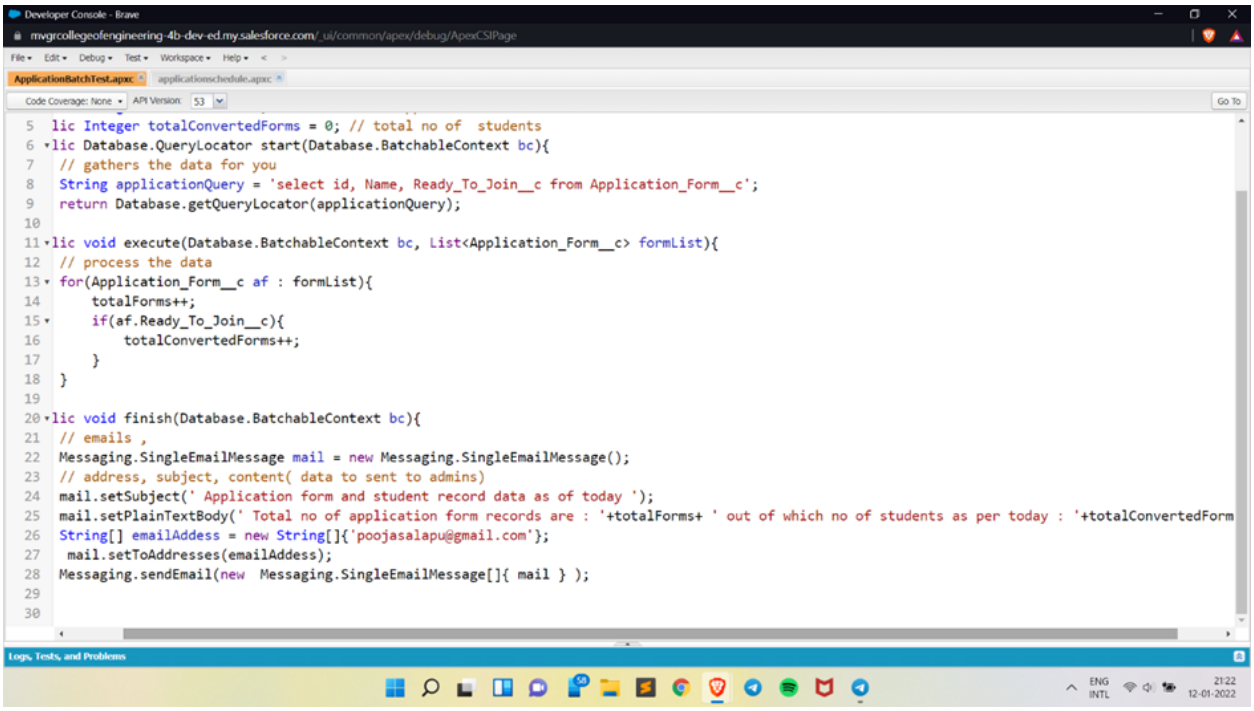
```
Public class ApplicationBatchTest implements Database.Batchable<sObject>{

   //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 } );
    }
}
```




Developer Console - Brave
mvgrcollegeofengineering-4b-dev-ed.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
ApplicationBatchTest.apxc
applicationschedule.apxc
Code Coverage: None
API Version: 53
Go To
lic Integer totalConvertedForms = 0; // total no of students
lic Database.QueryLocator start(Database.BatchableContext bc){
// gathers the data for you
String applicationQuery = 'select id, Name, Ready_To_Join__c from Application_Form__c';
return Database.getQueryLocator(applicationQuery);
lic void execute(Database.BatchableContext bc, List<Application_Form__c> formList){
// process the data
for(Application_Form__c af : formList){
totalForms++;
if(af.Ready_To_Join__c){
totalConvertedForms++;
}
}
lic 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 : '+totalConvertedForm
String[] emailAddess = new String[]{'poojasalapu@gmail.com'};
mail.setToAddresses(emailAddess);
Messaging.sendEmail(new Messaging.SingleEmailMessage[]{ mail });
Logs, Tests, and Problems
ENG
INTL
21:22
12-01-2022


Developer Console - Brave
mvgrcollegeofengineering-4b-dev-ed.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
ApplicationBatchTest.apxc
applicationschedule.apxc
Code Coverage: None
API Version: 53
Go To
public class ApplicationBatchTest implements Database.Batchable<sobject>, Database.stateful {
//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 Application_Form__c';
return Database.getQueryLocator(applicationQuery);
}
public void execute(Database.BatchableContext bc, List<Application_Form__c> formList){
// process the data
for(Application_Form__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 : '+totalConver
String[] emailAddess = new String[]{'poojasalapu@gmail.com'};
Logs, Tests, and Problems
ENG
INTL
21:22
12-01-2022

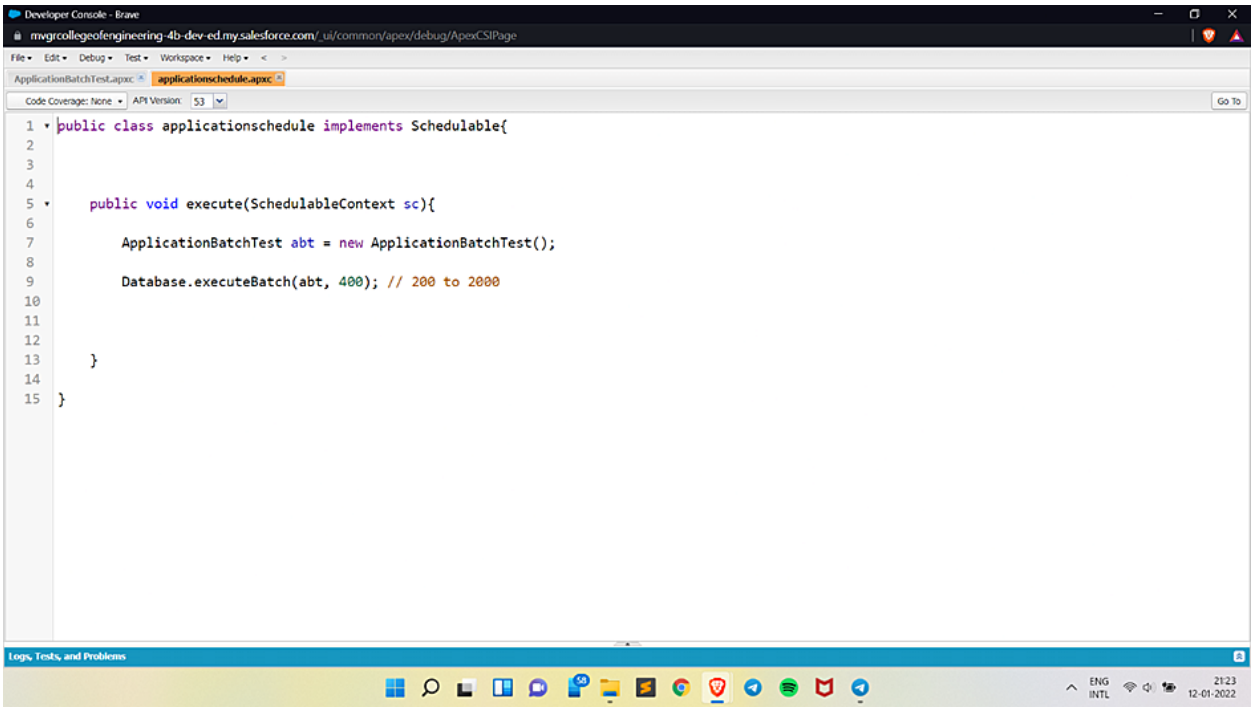


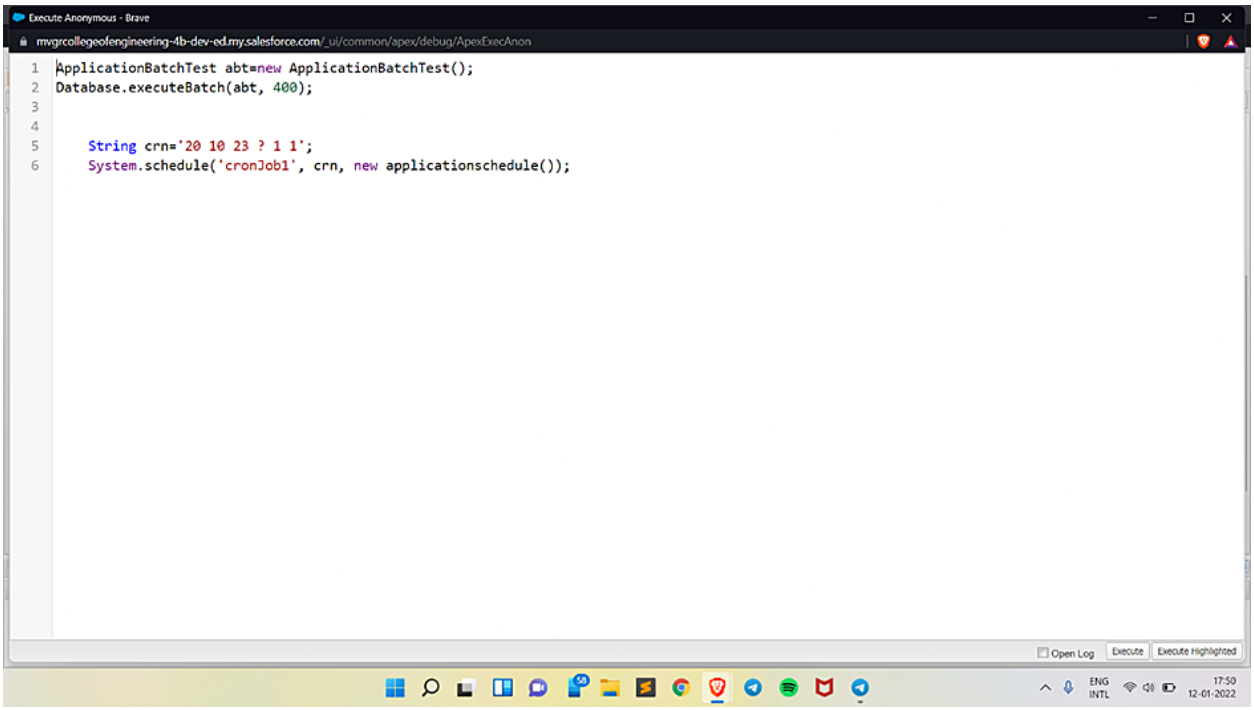
Task2:Create a scheduler class

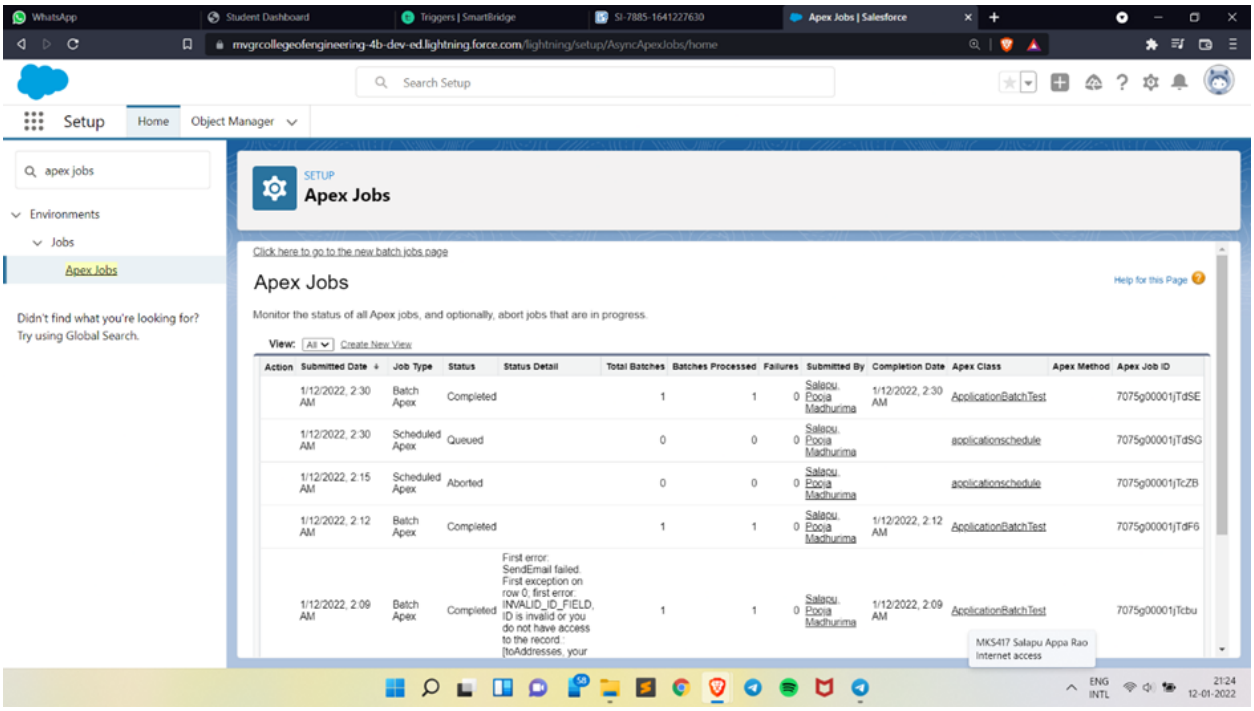
1. From the developer console create a new apex class and enter the following code.

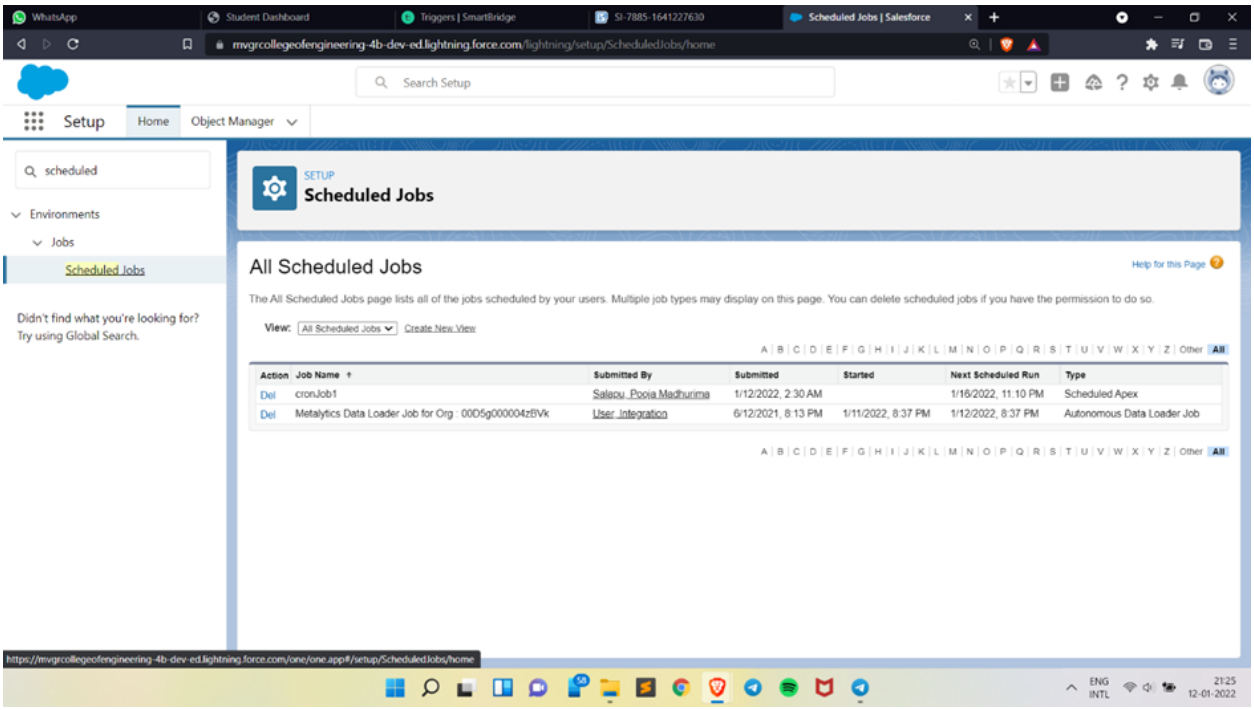
```
public class applicationschedule implements Schedulable{

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

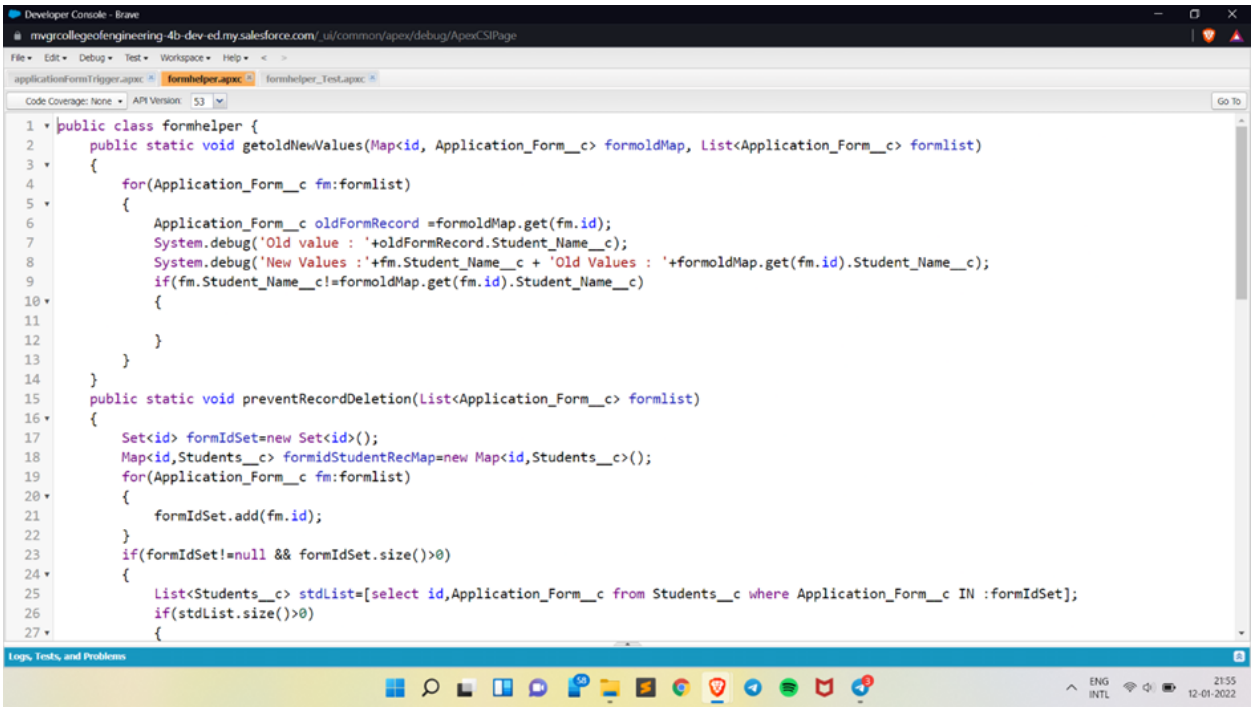
From setup search for apex class and click on schedule jobs and fill the details as per your requirements.


Developer Console - Brave
mvgrcollegeofengineering-4b-dev-ed.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
ApplicationBatchTest.apxc
applicationschedule.apxc
Code Coverage: None
API Version: 53
Go To
1 public class applicationschedule implements Schedulable{
2
3
4
5 public void execute(SchedulableContext sc){
6
7 ApplicationBatchTest abt = new ApplicationBatchTest();
8
9 Database.executeBatch(abt, 400); // 200 to 2000
10
11
12
13 }
14
15 }
Logs, Tests, and Problems
ENG
INTL
21:23
12-01-2022

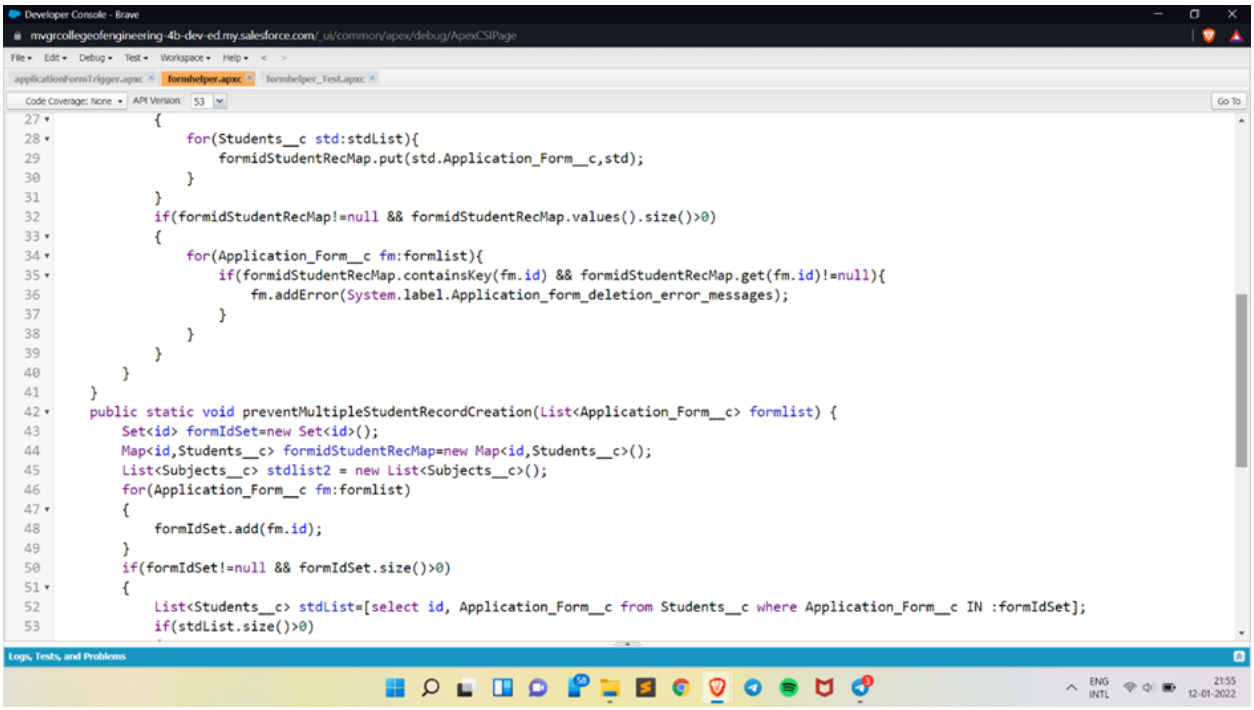

Execute Anonymous - Brave
mvgrcollegeofengineering-4b-dev-ed.my.salesforce.com/_ui/common/apex/debug/ApexExecAnon
1 ApplicationBatchTest abt=new ApplicationBatchTest();
2 Database.executeBatch(abt, 400);
3
4
5 String crn='20 10 23 ? 1 1';
6 System.schedule('cronJob1', crn, new applicationschedule());
Open Log
Execute
Execute Highlighted
ENG
INTL
17:50
12-01-2022


Apex Jobs | Salesforce
mvgrcollegeofengineering-4b-dev-ed.lightning.force.com/lightning/setup/AsyncApexJobs/home
Search Setup
Setup
Home
Object Manager
apex jobs
Environments
Jobs
Apex Jobs
Didn't find what you're looking for?
Try using Global Search.
SETUP
Apex Jobs
Click here to go to the new batch jobs page
Apex Jobs
Help for this Page
Monitor the status of all Apex jobs, and optionally, abort jobs that are in progress.
View: All
Create New View
Action
Submitted Date
Job Type
Status
Status Detail
Total Batches
Batches Processed
Failures
Submitted By
Completion Date
Apex Class
Apex Method
Apex Job ID
1/12/2022, 2:30 AM
Batch Apex
Completed
1
1
0
Salapu, Pooja Madhurima
1/12/2022, 2:30 AM
ApplicationBatchTest
7075g00001jTdSE
1/12/2022, 2:30 AM
Scheduled Apex
Queued
0
0
0
Salapu, Pooja Madhurima
applicationschedule
7075g00001jTdSG
1/12/2022, 2:15 AM
Scheduled Apex
Aborted
0
0
0
Salapu, Pooja Madhurima
applicationschedule
7075g00001jTcZB
1/12/2022, 2:12 AM
Batch Apex
Completed
1
1
0
Salapu, Pooja Madhurima
1/12/2022, 2:12 AM
ApplicationBatchTest
7075g00001jTdF6
1/12/2022, 2:09 AM
Batch Apex
Completed
First error: SendEmail failed. First exception on row 0; first error: INVALID_ID_FIELD, ID is invalid or you do not have access to the record: [toAddresses, your
1
1
0
Salapu, Pooja Madhurima
1/12/2022, 2:09 AM
ApplicationBatchTest
7075g00001jTcbu
MKS417 Salapu Appa Rao
Internet access
ENG
INTL
21:24
12-01-2022

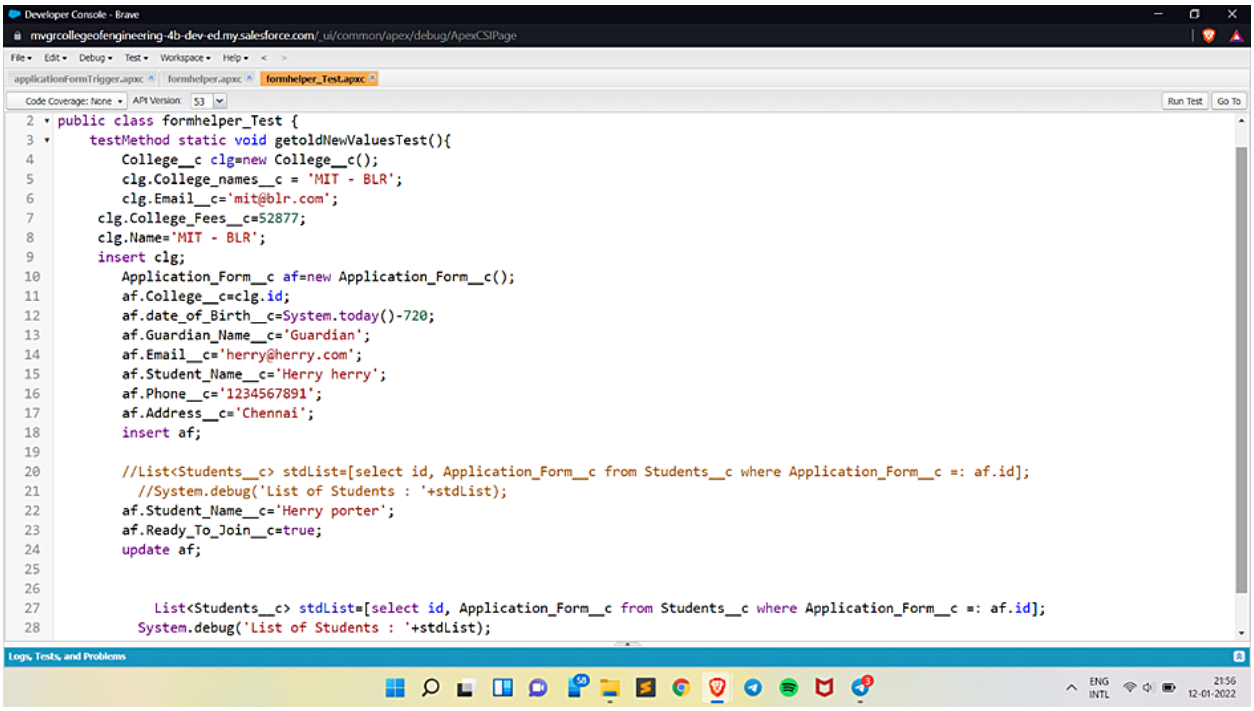

Scheduled Jobs | Salesforce
mvgrcollegeofengineering-4b-dev-ed.lightning.force.com/lightning/setup/ScheduledJobs/home
Search Setup
Setup
Home
Object Manager
scheduled
Environments
Jobs
Scheduled Jobs
Didn't find what you're looking for?
Try using Global Search.
SETUP
Scheduled Jobs
All Scheduled Jobs
Help for this Page
The All Scheduled Jobs page lists all of the jobs scheduled by your users. Multiple job types may display on this page. You can delete scheduled jobs if you have the permission to do so.
View: All Scheduled Jobs
Create New View
A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | Other | All
Action
Job Name
Submitted By
Submitted
Started
Next Scheduled Run
Type
Del
cronJob1
Salapu, Pooja Madhurima
1/12/2022, 2:30 AM
1/16/2022, 11:10 PM
Scheduled Apex
Del
Metalytics Data Loader Job for Org : 00D5g000004zBVk
User, Integration
6/12/2021, 8:13 PM
1/11/2022, 8:37 PM
1/12/2022, 8:37 PM
Autonomous Data Loader Job
https://mvgrcollegeofengineering-4b-dev-ed.lightning.force.com/one/one.app#/setup/ScheduledJobs/home
ENG
INTL
21:25
12-01-2022




```apex
public class formhelper {
    public static void getoldNewValues(Map<id, Application_Form__c> formoldMap, List<Application_Form__c> formlist)
    {
        for(Application_Form__c fm:formlist)
        {
            Application_Form__c oldFormRecord =formoldMap.get(fm.id);
            System.debug('Old value : '+oldFormRecord.Student_Name__c);
            System.debug('New Values :'+fm.Student_Name__c + 'Old Values : '+formoldMap.get(fm.id).Student_Name__c);
            if(fm.Student_Name__c!=formoldMap.get(fm.id).Student_Name__c)
            {

            }
        }
    }
    public static void preventRecordDeletion(List<Application_Form__c> formlist)
    {
        Set<id> formIdSet=new Set<id>();
        Map<id,Students__c> formidStudentRecMap=new Map<id,Students__c>();
        for(Application_Form__c fm:formlist)
        {
            formIdSet.add(fm.id);
        }
        if(formIdSet!=null && formIdSet.size()>0)
        {
            List<Students__c> stdList=[select id,Application_Form__c from Students__c where Application_Form__c IN :formIdSet];
            if(stdList.size()>0)
            {
```




```apex
            {
                for(Students__c std:stdList){
                    formidStudentRecMap.put(std.Application_Form__c,std);
                }
            }
            if(formidStudentRecMap!=null && formidStudentRecMap.values().size()>0)
            {
                for(Application_Form__c fm:formlist){
                    if(formidStudentRecMap.containsKey(fm.id) && formidStudentRecMap.get(fm.id)!=null){
                        fm.addError(System.label.Application_form_deletion_error_messages);
                    }
                }
            }
        }
    }
    public static void preventMultipleStudentRecordCreation(List<Application_Form__c> formlist) {
        Set<id> formIdSet=new Set<id>();
        Map<id,Students__c> formidStudentRecMap=new Map<id,Students__c>();
        List<Subjects__c> stdlist2 = new List<Subjects__c>();
        for(Application_Form__c fm:formlist)
        {
            formIdSet.add(fm.id);
        }
        if(formIdSet!=null && formIdSet.size()>0)
        {
            List<Students__c> stdList=[select id, Application_Form__c from Students__c where Application_Form__c IN :formIdSet];
            if(stdList.size()>0)
```



Day5:

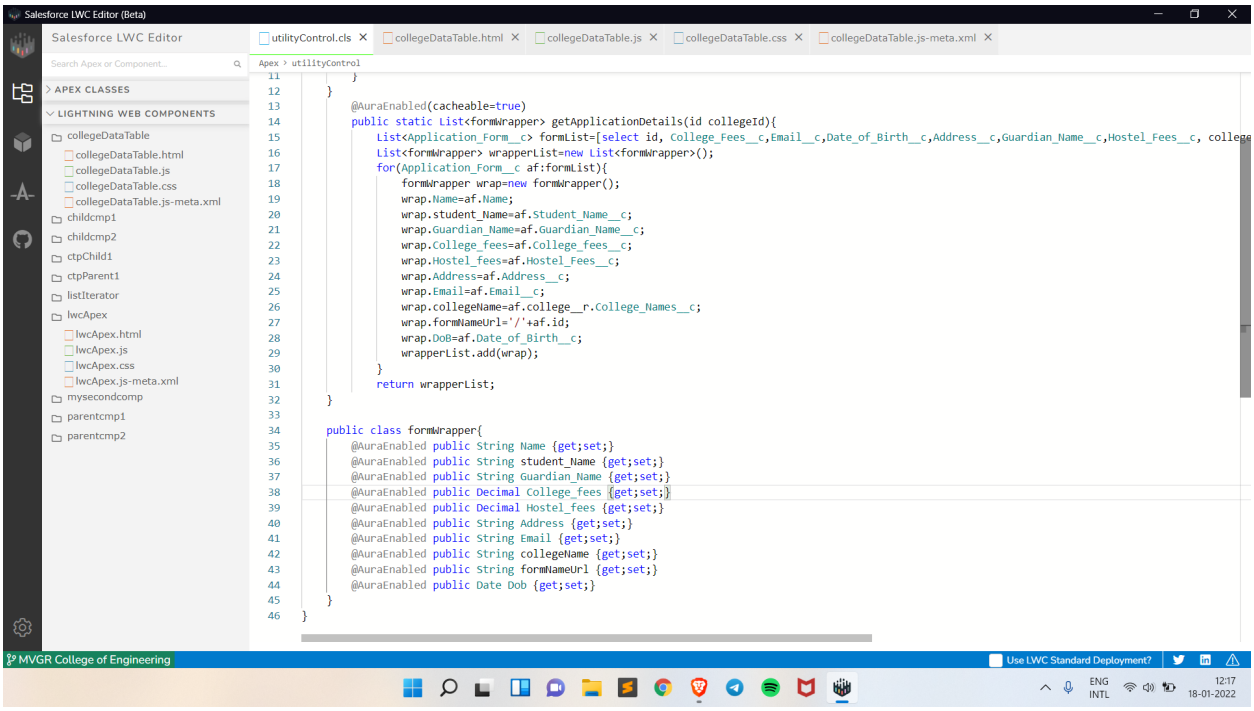
Topic:Lightening web components

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 .

Task1:Create college data table component(APEX CLASS)

Create an Apex Class With The Required Functionality

```
public class GetapplicationDetails
{
    @AuraEnabled(cacheable=true)
public static List<ApplicationForm__c> getapplicationvalues(id
CollegeId)
    {
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 =:CollegeId];
return formList;
    }
}
```



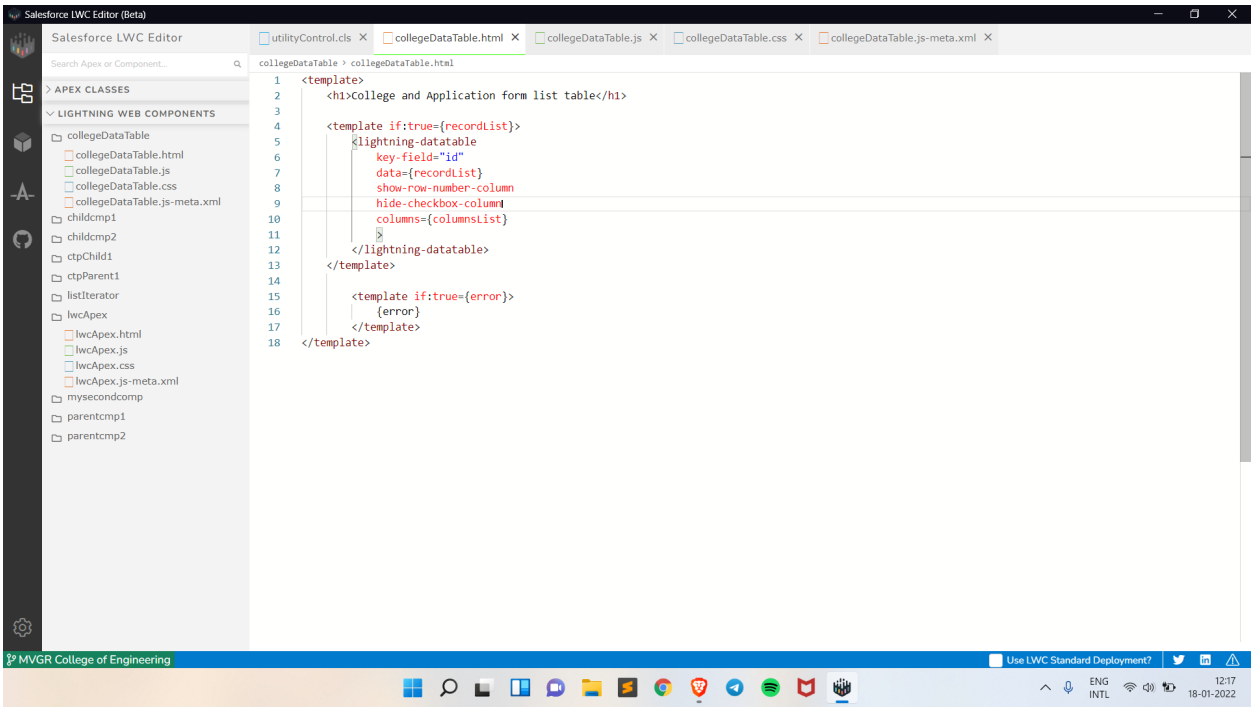

```
    }
}
    @AuraEnabled(cacheable=true)
    public static List<formWrapper> getApplicationDetails(id collegeId){
        List<Application_Form__c> formList=[select id, College_Fees__c,Email__c,Date_of_Birth__c,Address__c,Guardian_Name__c,Hostel_Fees__c, college
        List<formWrapper> wrapperList=new List<formWrapper>();
        for(Application_Form__c af:formList){
            formWrapper wrap=new formWrapper();
            wrap.Name=af.Name;
            wrap.student_Name=af.Student_Name__c;
            wrap.Guardian_Name=af.Guardian_Name__c;
            wrap.College_fees=af.College_fees__c;
            wrap.Hostel_fees=af.Hostel_Fees__c;
            wrap.Address=af.Address__c;
            wrap.Email=af.Email__c;
            wrap.collegeName=af.college__r.College_Names__c;
            wrap.formNameUrl='/'+af.id;
            wrap.DoB=af.Date_of_Birth__c;
            wrapperList.add(wrap);
        }
        return wrapperList;
    }

    public class formWrapper{
        @AuraEnabled public String Name {get;set;}
        @AuraEnabled public String student_Name {get;set;}
        @AuraEnabled public String Guardian_Name {get;set;}
        @AuraEnabled public Decimal College_fees {get;set;}
        @AuraEnabled public Decimal Hostel_fees {get;set;}
        @AuraEnabled public String Address {get;set;}
        @AuraEnabled public String Email {get;set;}
        @AuraEnabled public String collegeName {get;set;}
        @AuraEnabled public String formNameUrl {get;set;}
        @AuraEnabled public Date Dob {get;set;}
    }
}
```

Task2:Create college data table component(HTML CLASS)

Create the component with the following files.

HTML:

```
<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>
```



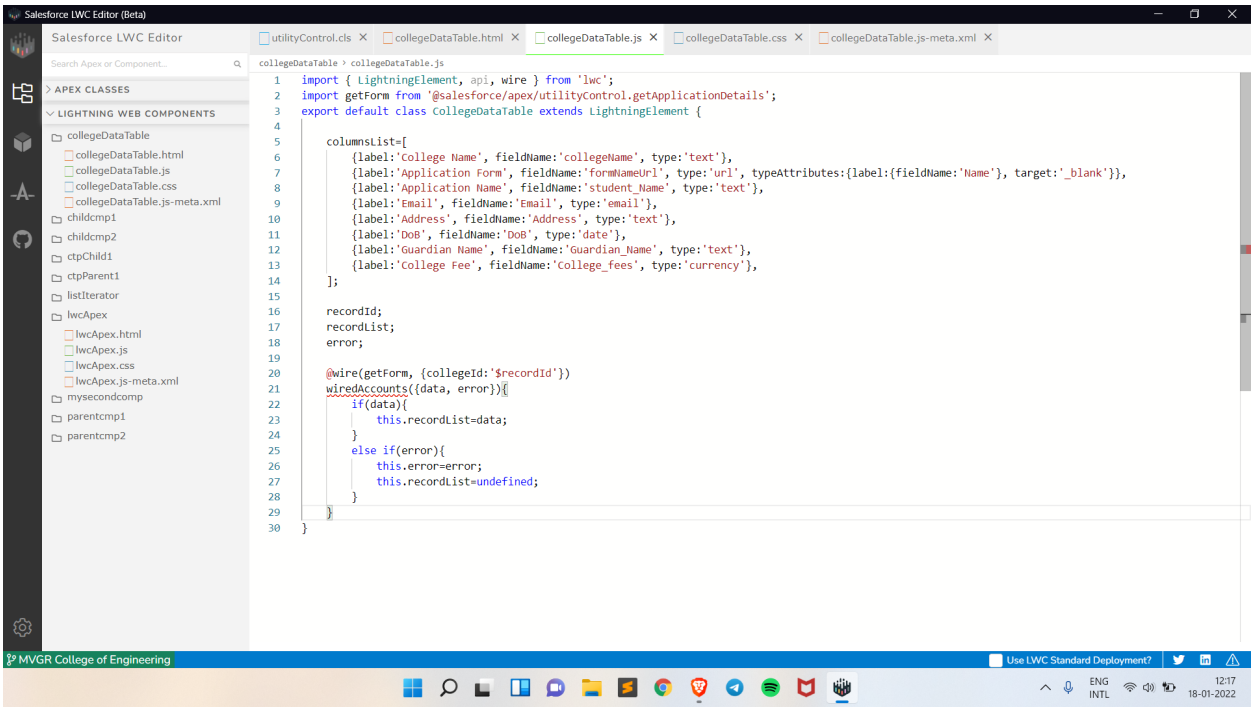
Task3:

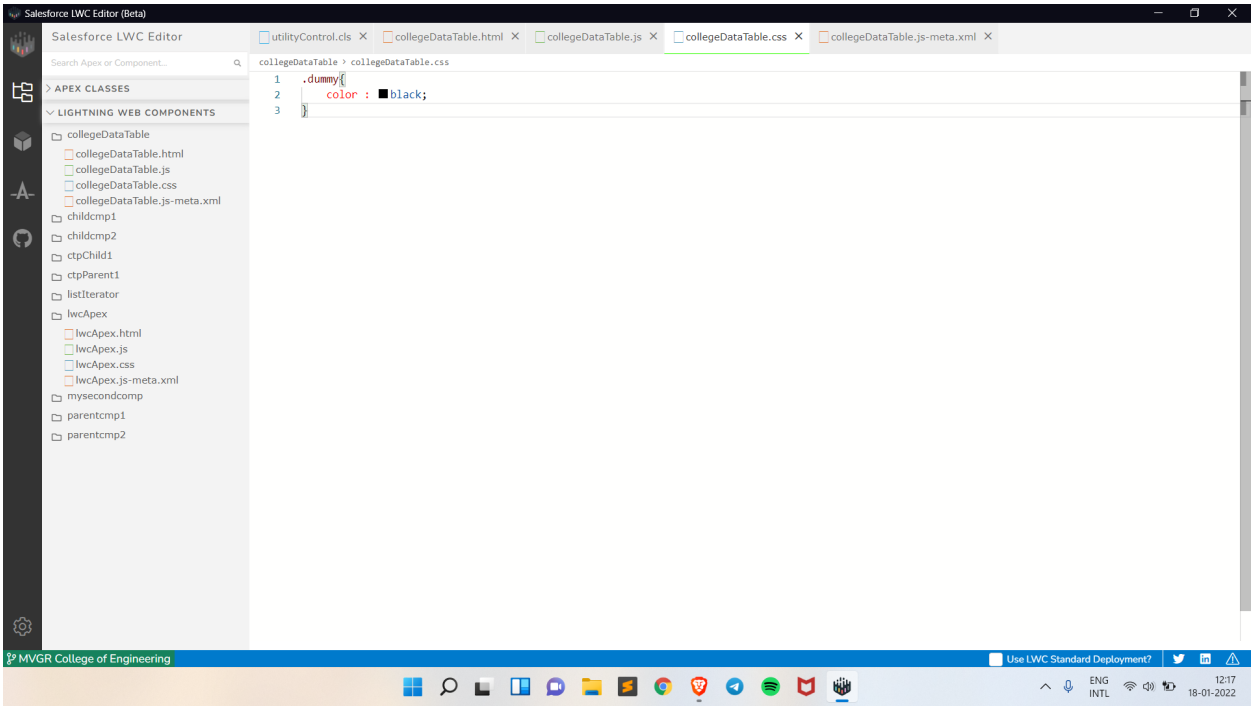
Topic:Create college data table component(SCRIPT CLASS)

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;
```

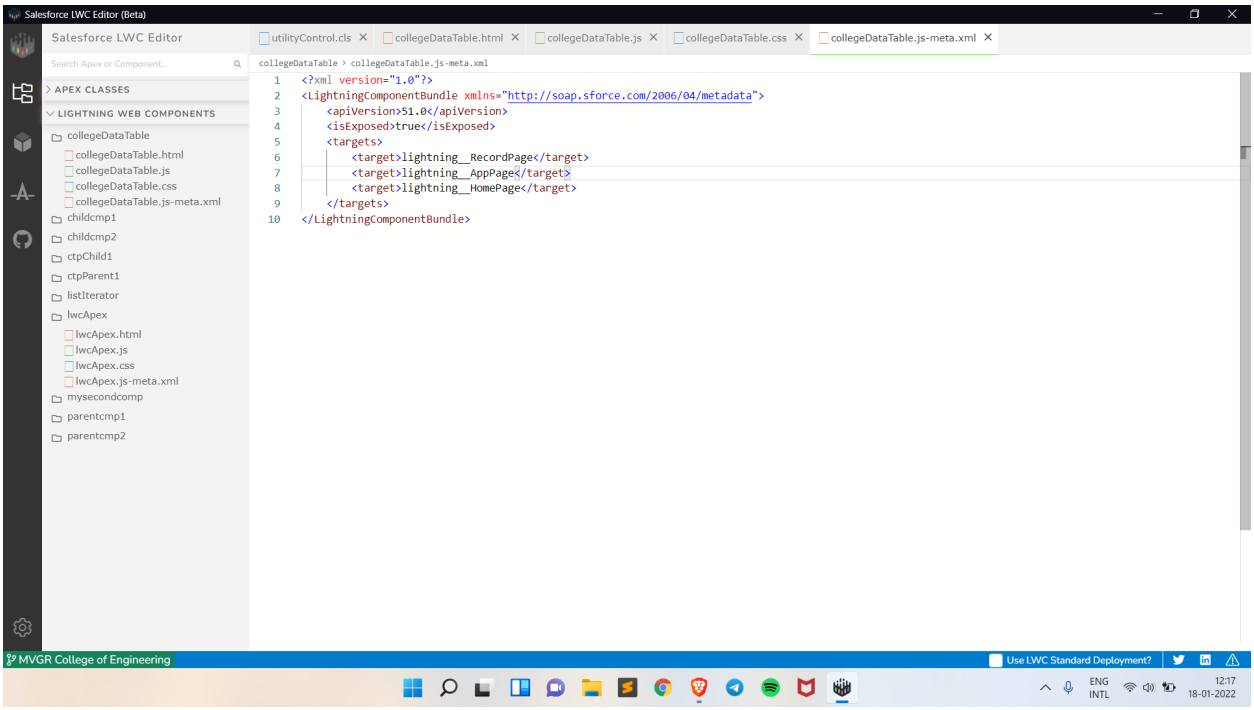
```
        }
        else if(error){
            this.error = error;
            this.recordList = undefined;
        }
    }
}
```





Task4:
Topic:Create college data table component(META CLASS)
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>
```



CONCLUSION:
In this project we got to learn the various steps of a college management application development with the help of various milestones like creating objects creating fields on the objects,defining the dependencies between the fields ,creating validation rules,Custom object creation,adding business logic to the application,creating apex class the lightening web components.