

COLLEGE MANAGEMENT APPLICATION

I have created a new developer account with my own credentials and with required information asked such as

name, email, username, organization name etc...

later, i have activated my developer account, where i receive a mail after creation of the developer account, and then logged into the salesforce, with the new username that i have created by developer account.

CUSTOM OBJECT CREATION:

Here, we need to create the custom objects for the applications

College

Application form

Students

Subject.

(Go to the setup>>object manager>>click on create new object)

Next, we should create the fields for the custom objects as follows:

Go to the custom object to which you want to add the fields, select fields and relationships>>new and select the field type that you want to create like collegefee, college name, collegeemail and related field types to the objects college,applicationform,subject and the students.

Setup > Object Manager > College

Fields & Relationships

| FIELD LABEL | FIELD NAME | DATA TYPE | CONTROLLING FIELD | INDEXED |
|------------------|------------------|--------------------|-------------------|---------|
| collegeemail | collegeemail__c | Picklist | Collegename | |
| collegefee | collegefee__c | Currency(7, 2) | | |
| Collegename | Collegename__c | Picklist | | |
| CollegeName | Name | Text(80) | | |
| Created By | CreatedById | Lookup(User) | | |
| hostelfee | hostelfee__c | Currency(6, 2) | | |
| Last Modified By | LastModifiedById | Lookup(User) | | |
| noofseats | noofseats__c | Picklist | Collegename | |
| Owner | OwnerId | Lookup(User/Group) | | |

Setup > Object Manager > applicationform

Fields & Relationships

| FIELD LABEL | FIELD NAME | DATA TYPE | CONTROLLING FIELD | INDEXED |
|--------------|-----------------|------------------------|-------------------|---------|
| address | address__c | Text(255) | | |
| College | Collegee__c | Master-Detail(College) | | |
| collegefee | collegefee__c | Currency(7, 2) | | |
| Created By | CreatedById | Lookup(User) | | |
| dateofbirth | dateofbirth__c | Date | | |
| email | email__c | Email (Unique) | | |
| form Number | Name | Auto Number | | |
| formid | formid__c | Auto Number | | |
| guardianname | guardianname__c | Text(30) | | |

Setup > Object Manager > student

Fields & Relationships

| FIELD LABEL | FIELD NAME | DATA TYPE | CONTROLLING FIELD | INDEXED |
|------------------|--------------------|-------------------------|-------------------|---------|
| address | address__c | Text(255) | | |
| applicationform | applicationform__c | Lookup(applicationform) | | |
| Collegename | Collegename__c | Text(30) | | |
| Created By | CreatedById | Lookup(User) | | |
| dateofbirth | dateofbirth__c | Date | | |
| guardianname | guardianname__c | Text(20) | | |
| Last Modified By | LastModifiedById | Lookup(User) | | |
| Owner | OwnerId | Lookup(User/Group) | | |
| phone | phone__c | Phone | | |

Setup > Object Manager > student

Fields & Relationships

| FIELD LABEL | FIELD NAME | DATA TYPE | CONTROLLING FIELD | INDEXED |
|------------------|--------------------|-------------------------|-------------------|---------|
| address | address__c | Text(255) | | |
| applicationform | applicationform__c | Lookup(applicationform) | | |
| Collegename | Collegename__c | Text(30) | | |
| Created By | CreatedById | Lookup(User) | | |
| dateofbirth | dateofbirth__c | Date | | |
| guardianname | guardianname__c | Text(20) | | |
| Last Modified By | LastModifiedById | Lookup(User) | | |
| Owner | OwnerId | Lookup(User/Group) | | |
| phone | phone__c | Phone | | |

Setup > OBJECT MANAGER
subject

| 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 Name | Name | Text(80) | | ✓ |
| subjectid | subjectid__c | Auto Number | | |

As shown above, create different field types for the objects named college, application form, students and subjects.

ADDING BUSINESS LOGIC APPLICATION:

Create the global picklist value set, (go to setup>>home tab>>and search as picklist value set) for the college and the subjects which include paper1 and paper2

Setup > Data > Picklist Value Sets

| Action | Values | API Name | Default | Chart Colors | Modified By |
|-------------------------|-----------|-----------|--------------------------|----------------------|---|
| Edit Del Deactivate | lendi-bir | lendi-bir | <input type="checkbox"/> | Assigned dynamically | Amrutha Neelamsetty, 1/8/2022, 10:15 PM |
| Edit Del Deactivate | lendi-maa | lendi-maa | <input type="checkbox"/> | Assigned dynamically | Amrutha Neelamsetty, 1/8/2022, 10:15 PM |
| Edit Del Deactivate | lendi-hyd | lendi-hyd | <input type="checkbox"/> | Assigned dynamically | Amrutha Neelamsetty, 1/8/2022, 10:15 PM |
| Edit Del Deactivate | lendi-del | lendi-del | <input type="checkbox"/> | Assigned dynamically | Amrutha Neelamsetty, 1/8/2022, 10:15 PM |
| Edit Del Deactivate | lendi-mum | lendi-mum | <input type="checkbox"/> | Assigned dynamically | Amrutha Neelamsetty, 1/8/2022, 10:15 PM |
| Edit Del Deactivate | lendi-ccu | lendi-ccu | <input type="checkbox"/> | Assigned dynamically | Amrutha Neelamsetty, 1/8/2022, 10:15 PM |

The screenshot shows the Salesforce Setup interface with the URL <https://lendiinstituteofengineer-bc-dev-ed.lightning.force.com/lightning/setup/Picklists/page?address=%2F0Nt5j000000onuW>. The left sidebar is expanded, showing sections like Data, Feature Settings, Objects and Fields, and a search bar. The main content area is titled "Picklist Value Sets". It displays an "Information" section with Label "paper1", Name "paper1", and Description. Below it is a "Picklist Values Used" section showing 4 active and inactive values: apex, java, c, and c++. A "Values" table lists these with their API names and modified by Amrutha Neelamsetty on 1/13/2022 at 2:04 AM. There are also sections for "Inactive Values" (none defined) and "Fields Where Used" (no records displayed).

This screenshot shows the same Salesforce Setup interface and URL as the first one. The main content area is titled "Picklist Value Sets" and displays a "Global Value Set Detail" section. It shows an "Information" section with Label "paper2", Name "paper2", and Description. Below it is a "Picklist Values Used" section showing 3 active and inactive values: maths, statistics, and english. A "Values" table lists these with their API names and modified by Amrutha Neelamsetty on 1/13/2022 at 2:05 AM. There are also sections for "Inactive Values" (none defined) and "Fields Where Used" (no records displayed).

After, this we need to create field dependencies,
such that the college name and email
,college name and no:of seats is depend on the college name, so that when
we select the name of the college then the related email of the college is

automatically updated without choosing the email of that particular college, similarly for the no: of seats in that college that are available, As shown in the below figure the field dependencies are set.

Next, the validation rules are created for college name and record in collage object and on application form object, to stop any modification on the application form once student record is created.

The screenshot shows two screenshots of the Salesforce Setup interface for the 'Collegee' object.

Screenshot 1: Collegee Field Dependencies

This page allows you to define dependencies between fields (e.g., dependent picklists). It shows one dependency entry:

| Action | Controlling Field | Dependent Field | Modified By |
|------------|-------------------|---------------------------|--|
| Edit Del | Collegename | collegeemail noofseats | Amrutha.Neelamsetty, 1/12/2022, 10:44 AM Amrutha.Neelamsetty, 1/12/2022, 10:46 AM |

Screenshot 2: Validation Rules

This page lists validation rules for the 'Collegee' object. There is one rule named 'collegename_similar':

| RULE NAME | ERROR LOCATION | ERROR MESSAGE | ACTIVE | MODIFIED BY |
|---------------------|----------------|-------------------|--------|---|
| collegename_similar | Top of Page | enter collegename | ✓ | Amrutha.Neelamsetty, 1/13/2022, 2:25 AM |

The screenshot shows the Salesforce Object Manager interface. On the left, a sidebar lists various setup categories like Details, Fields & Relationships, Page Layouts, etc. The main content area is titled "applicationform Validation Rule". It displays the validation rule detail with the following configuration:

- Validation Rule Detail**
- Rule Name:** student_record_created
- Error Condition Formula:**

```
AND( readyjoin__c == true,
    OR(
        ISCHANGED( address__c ),
        ISCHANGED( College__c ),
        ISCHANGED( City__c ),
        ISCHANGED( email__c ),
        ISCHANGED( guardianname__c ),
        ISCHANGED( phone__c )
    )
)
```
- Active:** Yes
- Error Message:** no errors
- Description:** Created By Amrutha Neelamsetty, 1/13/2022, 2:30 AM
- Modified By:** Amrutha Neelamsetty, 1/13/2022, 2:30 AM
- Error Location:** Top of Page

Next, the students record is created, there are two steps in creating the student records by the process builder and the flow builder.

The screenshot shows the Process Builder interface. The process is titled "Process Builder - application form process". The workflow starts with a "START" step, followed by an "applicationform" step. This is followed by a decision diamond labeled "ready to join is checked". If the condition is TRUE, it leads to an "IMMEDIATE ACTIONS" section containing "create a student rec...". If the condition is FALSE, it leads to another decision diamond labeled "+ Add Criteria". If this second condition is TRUE, it leads to another "IMMEDIATE ACTIONS" section containing "+ Add Action". Both paths eventually lead to a "STOP" step.

Here, in the process builder, click on the add object block, choose the object as application form(when the object is created)
Add criteria block, set the conditions of the criteria as application form equals , type as boolean and value is true, so now it checks the condition is working or not, Likewise add the criteria of different application fields like college fee, hostel fee.

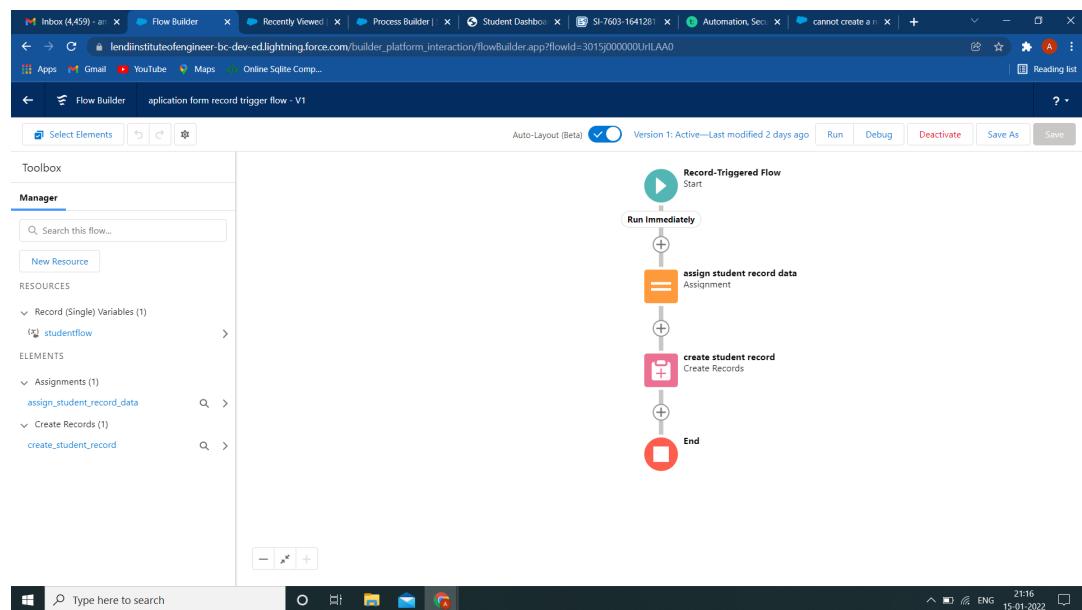
If the criteria is true, then the next action set the student record

So now when the details are filled in the application form then the student related field is created.

FLOW BUILDER:

Go to the flow builder, click on get records from the left panel and give it a name and select the object as student.

Now set the conditions, next connect the start block to the get records block and then add a loop from the left panel and add a new assignment as in the process builder like variable,operator and the value for different student name, college fee, nbbhostel fee etc.and debug and run after the connections are given, now the tabular form record type is obtained.



APEX CLASS(FOR APPLICATION FORM):

Go to setup and from the dropdown box click on the developer console, where we write our code(all our apex codes)

So, apex contains start(), execute and finish methods

Write the code in the apex as shown in figure below, here we given the code that if any student wants to join then send the email.

run the code and check whether the apex code is working fine or not.

The screenshot shows the Salesforce Developer Console interface. The top window displays the code for the `ApplicationBatchTest` 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, readytojoin__c from applicationform__c';
9         return Database.getQueryLocator(applicationQuery);
10    }
11    public void execute(Database.BatchableContext bc, List<applicationform__c> formList){
12        // process the data
13        for(applicationform__c af : formList){
14            totalForms++;
15            if(af.readytojoin__c){
16                totalConvertedForms++;
17            }
18        }
19    }
20 }
```

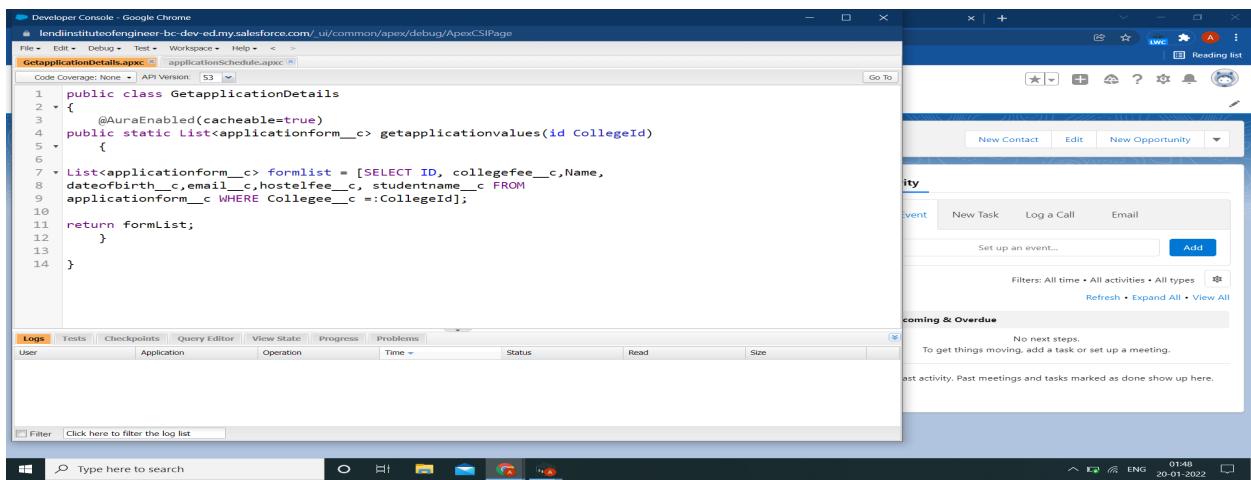
Below the code editor is a log table showing API activity:

| User | Operation | Time | Status | Read | Size |
|---------------------|--|------------------------|---------|--------|---------|
| Amrutha Neelamsetty | Batch Apex | 1/15/2022, 11:23:25 PM | Success | Unread | 4.93 KB |
| Amrutha Neelamsetty | Batch Apex | 1/15/2022, 11:23:24 PM | Success | Unread | 3.85 KB |
| Amrutha Neelamsetty | SerialBatchApexRangeChunkHandler | 1/15/2022, 11:23:24 PM | Success | Unread | 4.2 KB |
| Amrutha Neelamsetty | SerialBatchApexRangeChunkHandler | 1/15/2022, 11:23:24 PM | Success | Unread | 2.45 KB |
| Amrutha Neelamsetty | /services/data/v53.0/tooling/executeA... | 1/15/2022, 11:23:24 PM | Success | Unread | 3.48 KB |
| Amrutha Neelamsetty | /aura | 1/15/2022, 11:16:52 PM | Success | Unread | 2.13 KB |

The bottom part of the screenshot shows a browser window with the Gmail inbox open, displaying an email from "Amrutha Neelamsetty" with a yellow warning bar about potential spoofing.

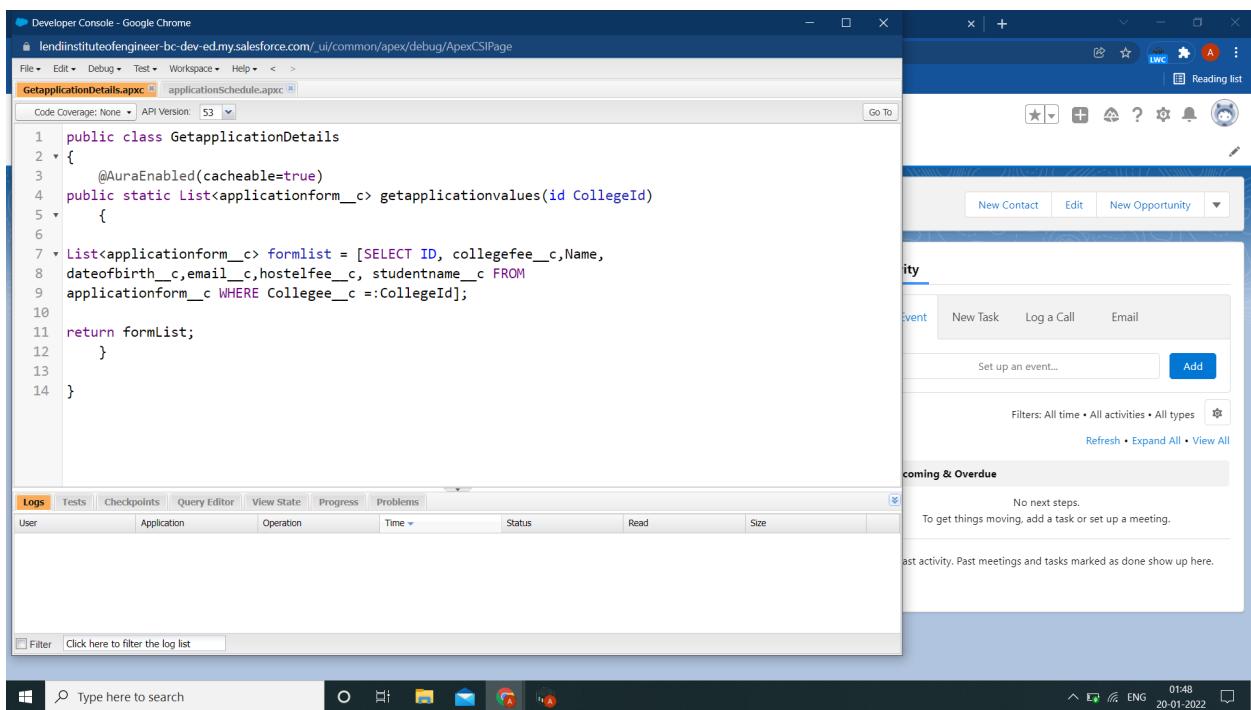
APEX (CREATE A SCHEDULAR CLASS)

Now, here write a apex code so that u can schedule the jobs that are available, we can schedule the job according to our convinent slots of date time.



```
public class GetapplicationDetails {
    @AuraEnabled(cacheable=true)
    public static List<applicationform__c> getapplicationvalues(id CollegeId) {
        List<applicationform__c> formlist = [SELECT ID, collegefee__c,Name,
dateofbirth__c,email__c,hostelfee__c, studentname__c FROM
applicationform__c WHERE Collegee__c =:CollegeId];
        return formList;
    }
}
```

This below figure shows the apex code of appliation details of the student in the form of the list.



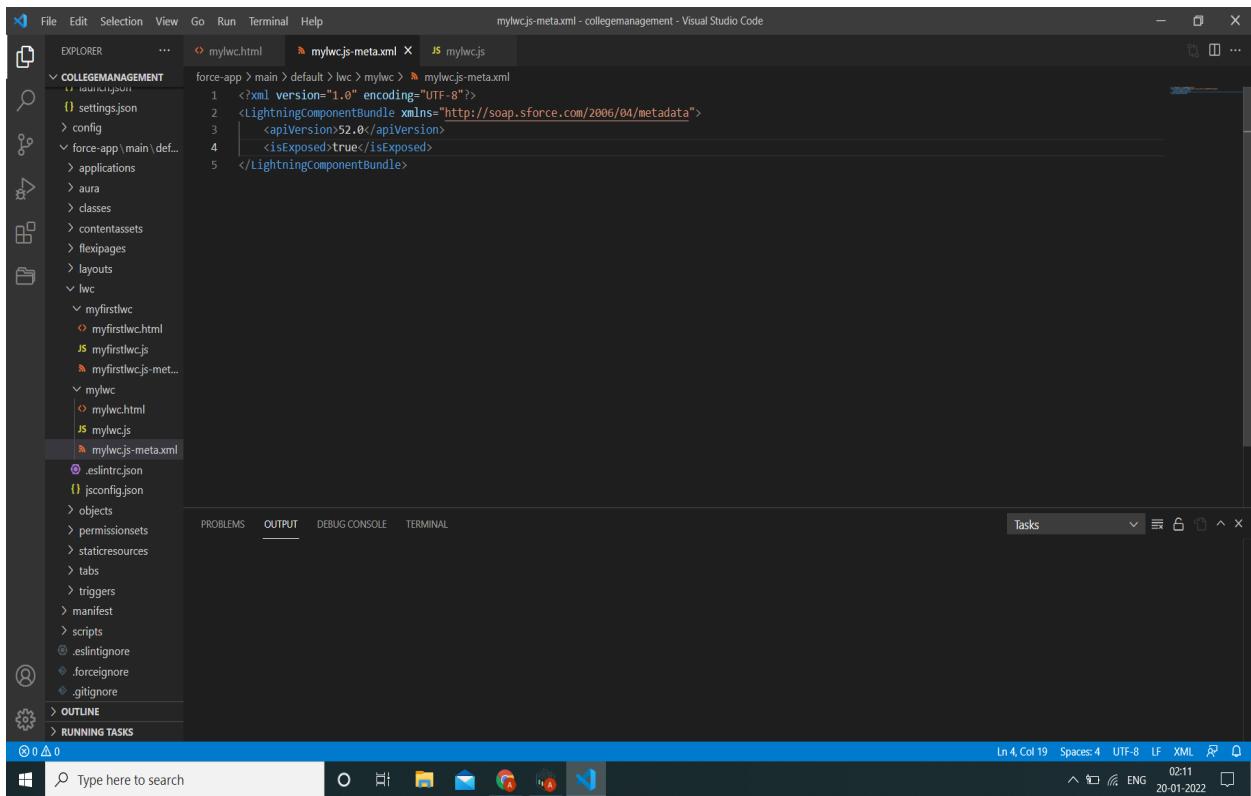
```
public class GetapplicationDetails {
    @AuraEnabled(cacheable=true)
    public static List<applicationform__c> getapplicationvalues(id CollegeId) {
        List<applicationform__c> formlist = [SELECT ID, collegefee__c,Name,
dateofbirth__c,email__c,hostelfee__c, studentname__c FROM
applicationform__c WHERE Collegee__c =:CollegeId];
        return formList;
    }
}
```

k ffff

LIGHTNING WEB COMPONENTS:

As, shown above create a college data table component by the apex class, Download, the visual studio code and install it in ur windows and also install project manager, salesforce extension pack and the salesforce cli. Now create new file and authorize it to an org, and then create a new lwc component, so when u create then autoatically creates javascript,html and meta file. and deploy that to org.

Now download salesforce lwc editor and pin to the crome extension



So, in the lwc component the new lwc file created in the visual studio code appears in the left panel.

we can choose the javascript,html and the meta files to write the data

The below figures shows the codes written in lwc editor:

Salesforce LWC Editor (Beta)

Search Apex or Component... collegedatable.html collegedatable.js collegedatable.css collegedatable.js-meta.xml

collegedatable > collegedatable.js

```

1 import { LightningElement, api, wire } from 'lwc';
2 import getApplicationValues from '@salesforce/apex/GetapplicationDetails.getapplicationvalues';
3 export default class Collegedatable extends LightningElement {
4   columnsList = [
5     {label : 'Application Form', fieldName : 'Name', type:'text' },
6     {label : 'College Fee', fieldName : 'collegefee_c', type:'currency' },
7     {label : 'Date of Birth', fieldName : 'dateofbirth_c', type:'date' },
8     {label : 'Email', fieldName : 'email_c', type:'email' },
9     {label : 'Hostel fees', fieldName : 'hostelfee_c', type:'Currency' },
10    {label : 'Student Name', fieldName : 'studentname_c', type:'text' }
11  ];
12
13  @api recordId;
14  recordList;
15  error;
16
17  @wire(getApplicationValues, {CollegelId : '$recordId'})
18  wiredCollegoData({data, error}){
19    if(data){
20      this.recordList = data;
21    }
22  }
23  else if(error){
24
25  }

```

Result Panel

```

1 collegedatable.html successfully deployed source 1/19/2022 11:21:47 PM
2 collegedatable.js successfully deployed source 1/19/2022 11:21:47 PM
3 collegedatable.css successfully deployed source 1/19/2022 11:21:47 PM
4 collegedatable.js-meta.xml successfully deployed source 1/19/2022 11:21:47 PM

```

Lendi Institute of engineering and technology Type here to search Use LWC Standard Deployment? 23:41 19-01-2022

Salesforce LWC Editor (Beta)

Search Apex or Component... collegedatable.html collegedatable.js collegedatable.css collegedatable.js-meta.xml

collegedatable > collegedatable.js-meta.xml

```

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

```

Result Panel

```

1 collegedatable.html successfully deployed source 1/19/2022 11:21:47 PM
2 collegedatable.js successfully deployed source 1/19/2022 11:21:47 PM
3 collegedatable.css successfully deployed source 1/19/2022 11:21:47 PM
4 collegedatable.js-meta.xml successfully deployed source 1/19/2022 11:21:47 PM

```

Lendi Institute of engineering technology Type here to search Use LWC Standard Deployment? 23:41 19-01-2022

Salesforce LWC Editor (Beta)

Search Apex or Component... collegedatable.html collegedatable.js collegedatable.css collegedatable.js-meta.xml

collegedatable > collegedatable.html

```

<template>
1   <h1> College and Application form list table </h1>
2   <template if:true={recordList}>
3     <lightning-dataTable
4       key-field="id"
5       data={recordList}
6       show-row-number-column
7       hide-checkbox-column
8       columns={columnsList}
9     >
10    </lightning-dataTable>
11  </template>
12  <template if:true={error}>
13    | (error)
14  </template>
15 </template>

```

Result Panel

```

1 collegedatable.html successfully deployed source 1/19/2022 11:21:47 PM
2 collegedatable.js successfully deployed source 1/19/2022 11:21:47 PM
3 collegedatable.css successfully deployed source 1/19/2022 11:21:47 PM
4 collegedatable.js-meta.xml successfully deployed source 1/19/2022 11:21:47 PM

```

Lendi Institute of engineering technology Type here to search Use LWC Standard Deployment? 23:41 19-01-2022

So, write the codes and deploy the codes,

Now go to the sales>>and create the tab of datatable and add the college table to it. then the required college table component is created by using the lightning web component as shown in the below figure.

The screenshot displays a Salesforce Lightning Web Component (LWC) application titled "College Management". The main content area features a "Datatable" section showing a list of students with columns for Application Form, College Fees, Date Of Birth, Email, Hostel Fees, and StudentName. One row is visible, showing F-000002, \$32,000.00, Jan 14, 2004, srate@instry.com, 12000, and abhi. Below the datatable is a "Related" section and a "Details" section. To the right is an "Activity" sidebar with tabs for New Event, New Task, Log a Call, and Email. It includes a button to "Add" an event and a note about upcoming and overdue tasks. The bottom of the screen shows a Windows taskbar with various icons and a system tray.

| Application Form | College Fees | Date Of Birth | Email | Hostel Fees | StudentName |
|------------------|--------------|---------------|------------------|-------------|-------------|
| 1 F-000002 | \$32,000.00 | Jan 14, 2004 | srate@instry.com | 12000 | abhi |

So, likewise the college management application is done.