

LIKE UNIVERSITY

(University Management System)



Our motto

Explore, Learn, Thrive

MODULES

- University application form
- Student Creation
- Fees Payments
- Results
- Event & Events Registration
- Insights (Reports & Dashboards)

Profiles:

- 1) Management
- 2) Professor
- 3) CR

Objects:

1) Student

- Student name (Text)
 - First Name
 - Last Name
- Father/Guardian name (Text) (Company)
- Date of Birth (Date)
- Gender (Male, Female, Others)
- Department (Lookup)
- Email (Email)
- Phone (Phone)
- Address (Street, city, PIN code, state, Country)
- Student ID (auto)
- Year of join (Formula Field –TEXT(YEAR (TODAY ())))
- 10th board
- 10th score
- 12th board
- 12th score
- Sem 1 total subjects – Roll-up summary
- Sem 1 passed subjects – Roll-up summary
- Sem1 Actual CGPA total-Roll-up summary
- Sem 1 CGPA-formula
- Similarly above for sem2 – sem8

2) Professor

- Professor ID (auto)
- Professor Name
- Department (Lookup)
- Subject (lookup)
- Experience (Number)
- Email (Email)
- Phone (Phone)
- HOD(Checkbox) – Check if he is HOD

3) Department

- Department ID(Auto)
- Department Name(text)
- Fees Per Semester (Currency)

4) Marks

- Student id(master detail)
- Department (Formula Field)
- Semester(Picklist)
- Subject ID(Lookup)
- Subject Name(Formula)
- Subject Grade Points(Formula Field)
- Subject Grade(Formula)---will be created later
- Subject Marks(Number(3,1))

5) Fee payments

- Payment ID(Auto Num)
- Student ID(Master-Detail)
- -Student Name(Formula)
- -Department(Auto-populates--formula)
- Semester(Picklist)
- -Fees Amount(Autopopulate--formula)
- Paid(checkbox)
- Mode Of Payment (Picklist)

6) Lead

- Student name
- Father/Guardian name (Company)
- Date of Birth
- Gender(Male, Female, Others)
- Department(picklist)
 - CSE
 - ECE
 - EEE
 - MECH
 - CIVIL
- Email (Email)
- Phone (Phone)
- Address (Street, city, PIN code, state, Country)
- 10th board name(global picklist)
 - State Board
 - CBSE
 - ICSE
- 10th score
- 12th board name(global picklist)
- 12th score
- Admission Fees paid – checkbox
- Application Status (Picklist)

- New
- Pending
- Approved
- Rejected

7)Subject

- Subject Code
- Subject Name
- Department ID(Lookup)
- SEM(Picklist)

8)Event

- Contact_Organizer__c
- Event_Date_Time__c
- Event_Details__c
- Event_Status__c
- Event_Venue__c
- Name
- Organizer__c
- Registration_End_Date__c
- Registration_Start_Date__c
- Registration_Status__c
- specify_venue_if_others__c

9)Event Registration

- Event_Name__c
- Name
- Role_in_event__c – (Picklist)
 - Organizer
 - Volunteer
 - Participant
 - Guest Participant
- Student_ID__c

***Note** - Most of above fields have their respective Validations which are not documented here. Please choose according to project requirements.

Duplicate rule: for lead object

Using Email &phone

Action when creating- block

Action when editing – block

Formula Fields:

1) Student – Year of Join – YEAR(TODAY())

2) Student – Age - (TRUNC((TODAY() - Date_of_Birth__c) , 0))/365

3) Mark – CGPA –

IF(ISBLANK(Subject_Marks__c), VALUE("") ,
IF(Subject_Marks__c > 90 && Subject_Marks__c <= 100, 10,
IF(Subject_Marks__c > 80 && Subject_Marks__c <= 90, 9 ,
IF(Subject_Marks__c > 70 && Subject_Marks__c <= 80, 8 ,
IF(Subject_Marks__c > 60 && Subject_Marks__c <= 70, 7 ,
IF(Subject_Marks__c > 55 && Subject_Marks__c <= 60, 6 ,
IF(Subject_Marks__c >= 50 && Subject_Marks__c <= 55, 5 ,
IF(Subject_Marks__c >=40 && Subject_Marks__c <50, 4 , 0))))))

4)Mark – Grade –

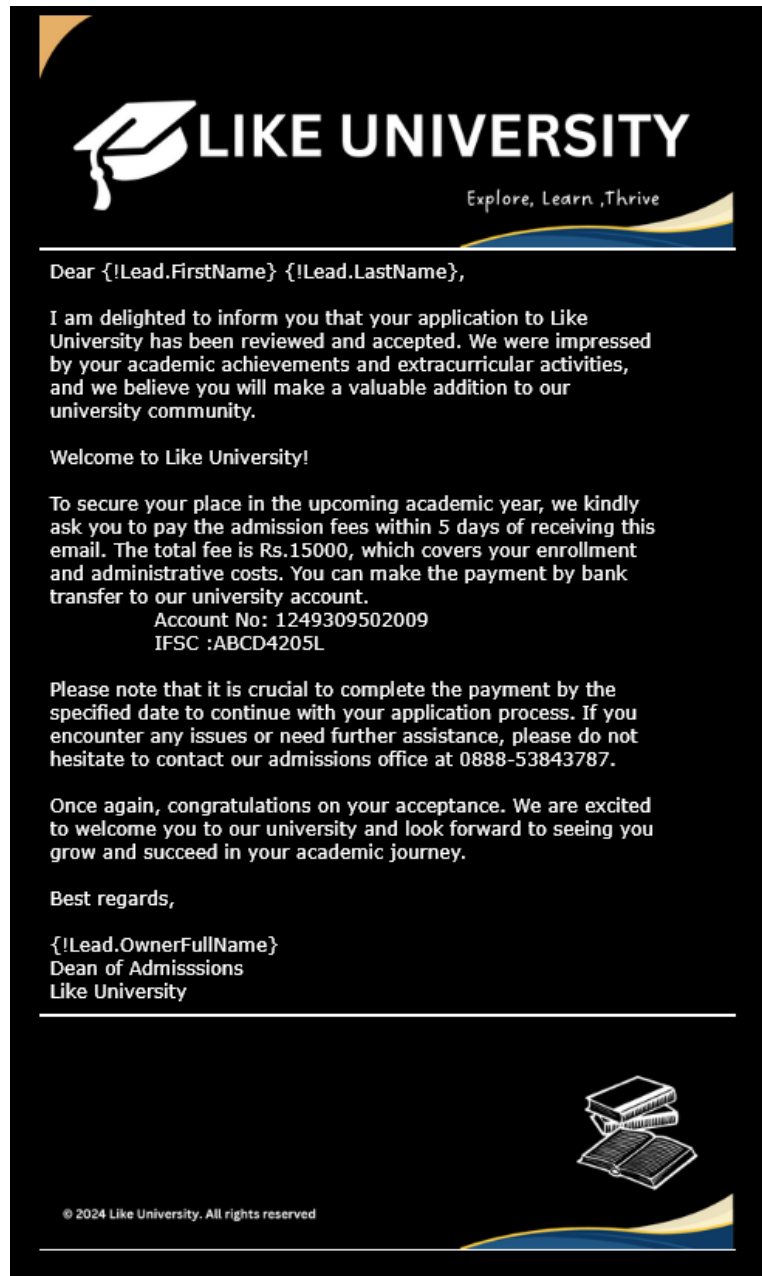
IF(ISBLANK(Subject_CGPA__c), 'N/A', CASE(Subject_CGPA__c , 10, 'O', 9,
'A+', 8, 'A', 7, 'B+', 6, 'B', 5, 'C', 4, 'P', 0, 'F', 'N/A'))

5)Student – CGPA –

IF(Sem_1_total_subjects__c = Sem_1_Passed_Subjects__c ,
Semester_1_Actual_CGPA__c / Sem_1_Passed_Subjects__c , 0)

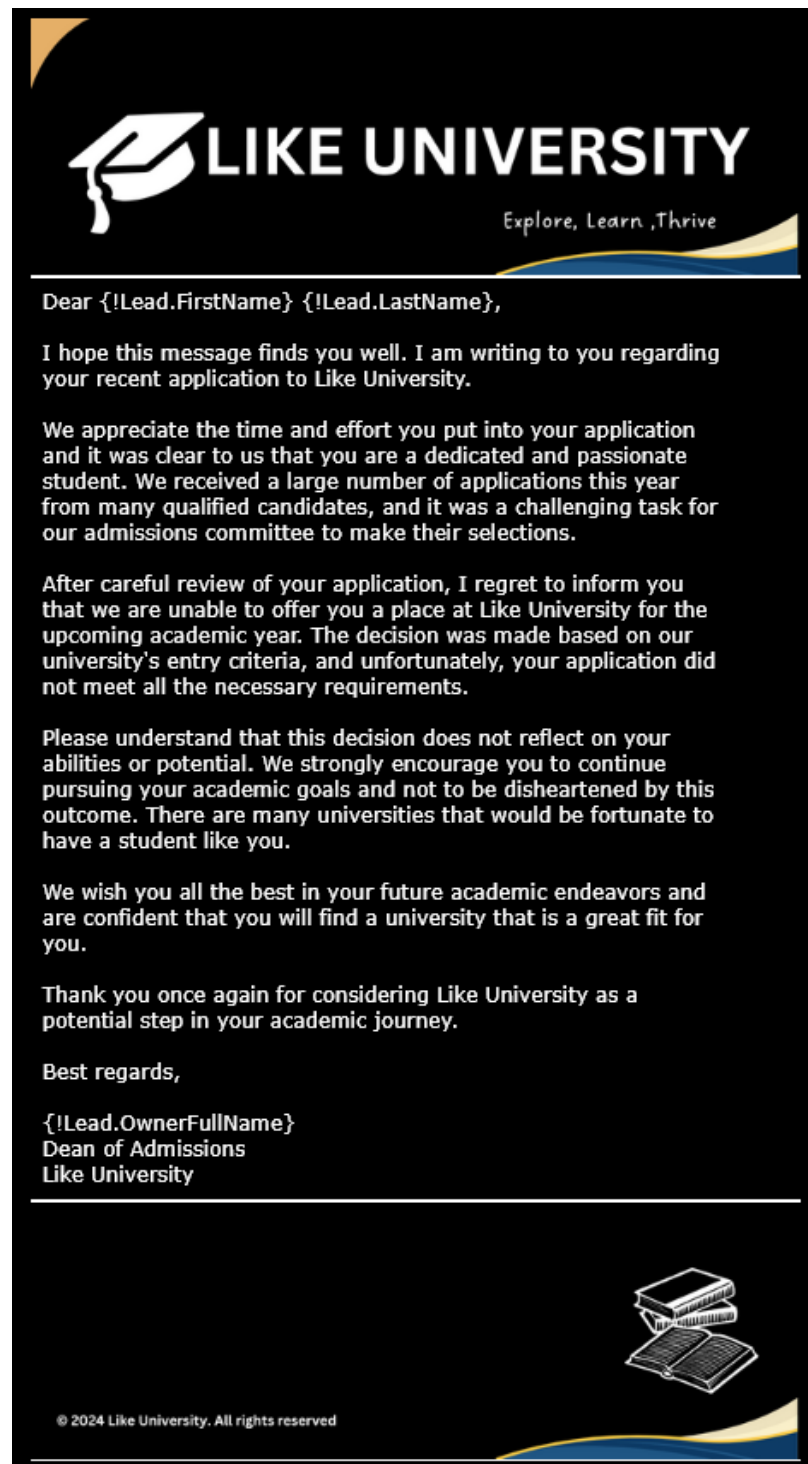
EMAIL ALERTS :

1)Accepting Mail:



On approval of the application the above email is sent.

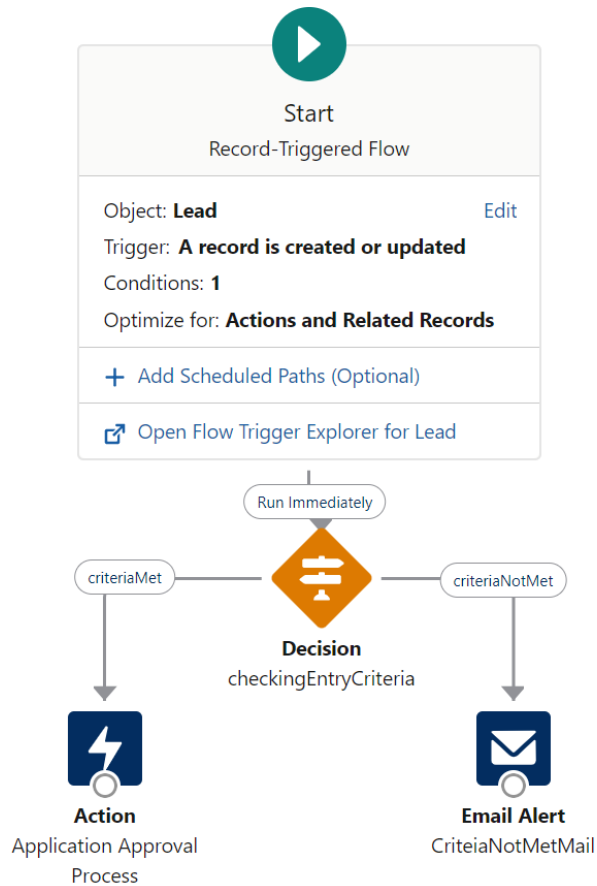
2)Rejecting Mail



On rejection of the application, this mail is sent.

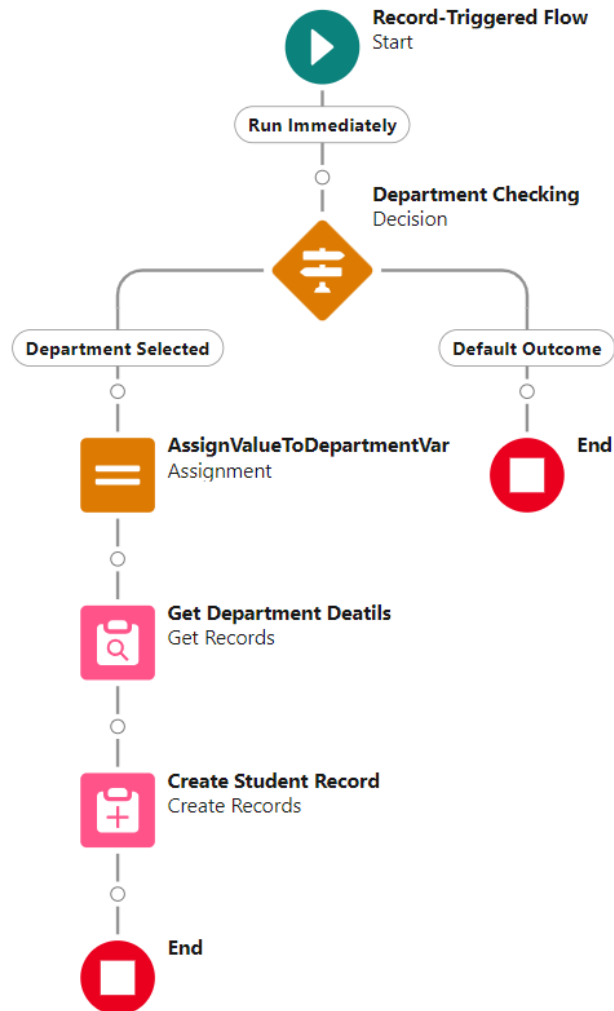
FLows:

1)Application Auto Approval Process:



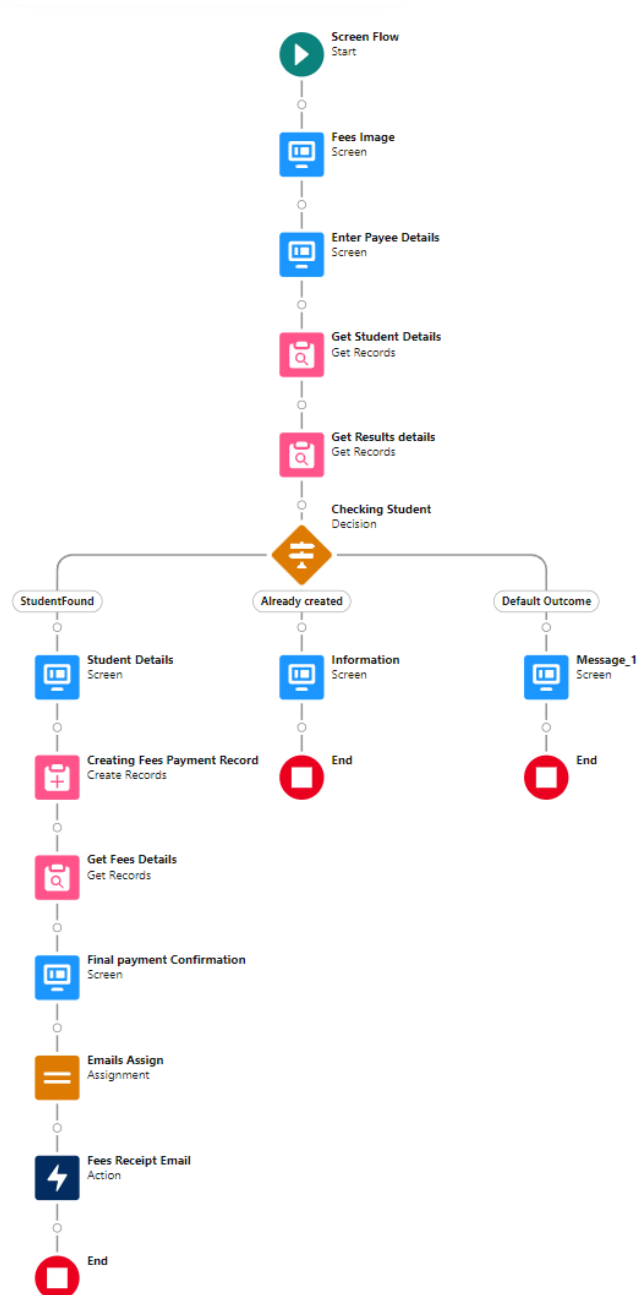
This is an auto approval process that sends application for management approval, if management approves application, then a mail alert is sent to pay application fees else an rejection mail is sent.

2) Creating an Student from Lead automatically on admission fees payment



On admission fees payment marked as paid, the Application(Lead) is converted to student automatically.

3) Fees Payment Screen Flow



This screen flow is used by management to create a record on fees payment of a student and sent an email to student.

Working this screenflow

Fees Payment Screen Flow

* Enter Student ID

LU-000070

* Semester

SEM-1

* Email

abcd@gmail.com

[Previous](#)

[View Details](#)

Fees Payment Screen Flow

Student Details

Student ID : LU-000070

Student Name : Rishi Kumar

Department : Electronics & Communication Engineering

Semester : SEM-1

Fees to be paid : 50,000

[Previous](#)

[Next](#)

Fees Payment Screen Flow

Payment Successful.

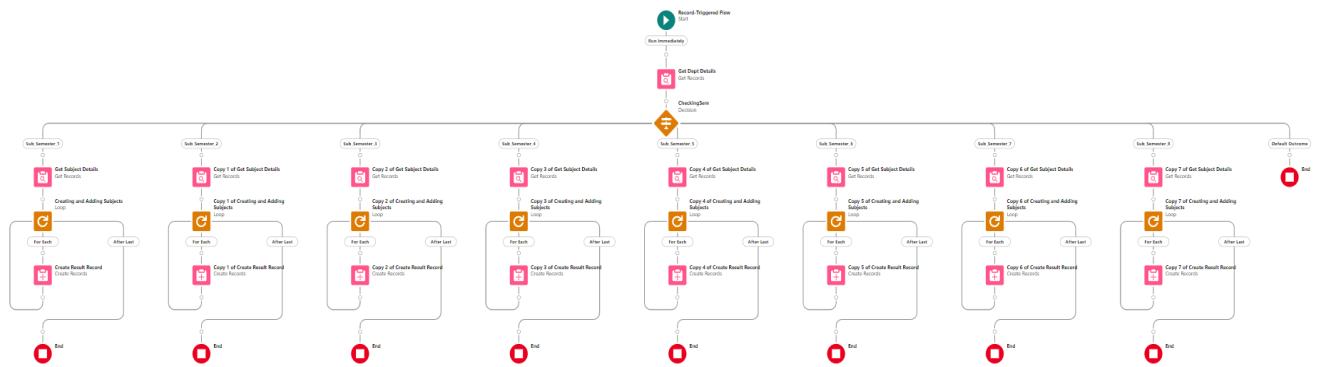
For more details, verify your mail or contact Computer Section.

Thank you.

[Previous](#)

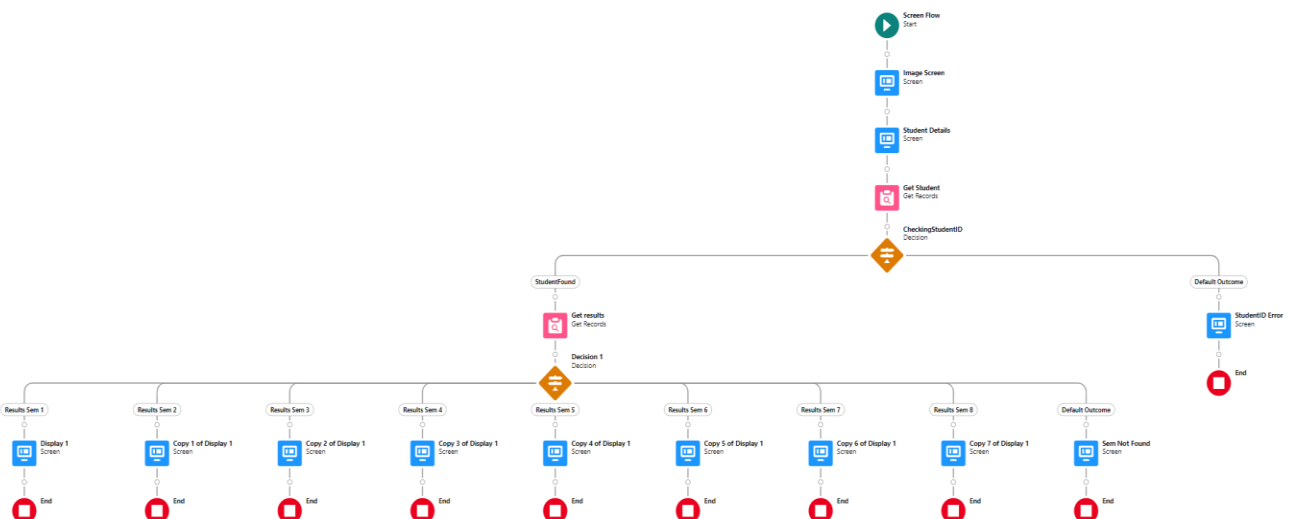
[Next](#)

4) Auto-creating results based on fees payment for semester



This is used for automatically creating results for each subject of semester for which student has paid the fees. For example, If student pays fee for semester-1, the all the subjects of the respective semester is created & based on student department.

5) Check Results Screen Flow



This is used to check results of student.

Working of above screen flow

Get Result Screen Flow

Enter Student ID

LU-000001

Semester

SEM-1

[Previous](#)

[Show Result](#)

Get Result Screen Flow

RESULT

STUDENT ID : LU-000001

STUDENT NAME : Vinod Kumar

DEPARTMENT : Computer Science & Engineering

SEMESTER : SEM-1

CGPA : 8.86

GRADE : A

[Previous](#)

[Finish](#)

Approval Process Used in Flow

Approval Processes

Lead: Student Application Approval

[« Back to Approval Process List](#)

Help for this Page ?

Process Definition Detail

EditCloneDeactivate

Process Name

Student Application Approval

Active

☒

Unique Name

Student_Application_Approval

Next Automated Approver Determined By

Description

Entry Criteria

Record Editability

Administrator OR Current Approver

Allow Submitters to Recall Approval Requests

☐

Approval Assignment Email Template

Initial Submitters

Lead Owner

Created By

Vinod Kumar Reddy, 12/04/2024, 12:11 pm

Modified By

Vinod Kumar Reddy, 26/05/2024, 10:54 pm

Initial Submission Actions

Add ExistingAdd New

Action	Type	Description
	Record Lock	Lock the record from being edited
Edit Remove	Field Update	Application Status Pending

Approval Steps

Show Actions | Edit

Action	Step Number	Name	Description	Criteria	Assigned Approver	Reject Behavior
	1	Step 1			User: Aravind Krishna	Final Rejection

Final Approval Actions

Add ExistingAdd New

Action	Type	Description
Edit	Record Lock	Unlock the record for editing
Edit Remove	Email Alert	University Acceptance Mail Alert
Edit Remove	Field Update	Approval Date Udt
Edit Remove	Field Update	Application Status Approved

Final Rejection Actions

Add ExistingAdd New

Action	Type	Description
Edit	Record Lock	Unlock the record for editing
Edit Remove	Field Update	Application Status Rejected
Edit Remove	Email Alert	University Rejection Mail Alert

Recall Actions

Add ExistingAdd New

Action	Type	Description
	Record Lock	Unlock the record for editing

This is an approval process that is used for application approval.

C.R.U.D ACCESS ON PROFILE

Management:

	Read	Create	Edit	Delete
Departments	✓	✓	✓	✓
Fees Payments	✓	✓	✓	✓
Professors	✓	✓	✓	✓
Result	✓	✓	✓	✓
Student	✓	✓	✓	✓
Subjects	✓	✓	✓	✓
Leads	✓	✓	✓	✓
Events	✓	✓	✓	✓
Event Registration	✓	✓	✓	✓

Professor:

	Read	Create	Edit	Delete
Departments	✓	✗	✗	✗
Fees Payments	✗	✗	✗	✗
Professors	✓	✗	✗	✗
Result	✓	✗	✓	✗
Student	✓	✗	✗	✗
Subjects	✓	✗	✗	✗
Event	✓	✗	✓	✗
Event Registration	✓	✓	✓	✗

CR:

	Read	Create	Edit	Delete
Departments	✗	✗	✗	✗
Fees Payments	✗	✗	✗	✗
Professors	✗	✗	✗	✗
Result	✗	✗	✗	✗
Student	✗	✗	✗	✗
Subjects	✗	✗	✗	✗
Leads	✗	✗	✗	✗
Events	✓	✗	✗	✗
Event Registration	✓	✓	✗	✗

Student:

Student is a guest user he can check his/her results and payments from college website (no login needed). (Built using Experience Builder & LWC)

WEB-TO-LEAD FORM AS APPLICATION FROM ALONG WITH VS CODE VALIDATIONS:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <!-- <meta name="viewport" content="width=, initial-scale=1.0"> -->
  <title>Application | Like University</title>
  <link rel="icon" type="image/x-icon" href=".\logo_no_bg_favicon_32.png">
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.min.css"
  integrity="sha384-
xOolHFLEh07PJGoPkLv1IbcEPTNtaed2xpHsD9ESMhqIYd0nLMwNLD69Npy4HI+N"
crossorigin="anonymous">
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/js/bootstrap.bundle.min.js"
  integrity="sha384-
Fy6S3B9q64WdZWQUiU+q4/2Lc9npb8tCaSX9FK7E8HnRr0Jz8D6OP9dO5Vg3Q9ct"
  crossorigin="anonymous"> </script>
  <script src="https://cdn.jsdelivr.net/npm/jquery@3.5.1/dist/jquery.slim.min.js"
  integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
  crossorigin="anonymous"> </script>
  <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"
  integrity="sha384-
9/reFTGAW83EW2RDu2S0VKAizap3H66lZ81PoYIFhbGU+6BZp6G7niu735Sk7lN"
  crossorigin="anonymous"> </script>
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/js/bootstrap.min.js"
  integrity="sha384-
+sLIOodYLS7CIrQpBjl+C7nPvqq+FbNUBDunl/OZv93DB7Ln/533i8e/mZXLi/P+"
  crossorigin="anonymous"> </script>
  <link href='https://cdnjs.cloudflare.com/ajax/libs/bootstrap-
datepicker/1.9.0/css/bootstrap-datepicker.min.css'
rel='stylesheet' type='text/css'>
  <script src='https://cdnjs.cloudflare.com/ajax/libs/bootstrap-
datepicker/1.9.0/js/bootstrap-datepicker.min.js'
type='text/javascript'> </script>
```



```
<style>
  #header {
    height: 100px;
    width: auto;
    background-image: url(".\\Untitled design.png");
    background-position: center;
    background-repeat: no-repeat;
    background-size: cover;
  }

  .marquee {
    height: 50px;
    overflow: hidden;
    position: relative;
    background: #fefefe;
    color: red;
    font-weight: 700;
    font-size: small;
    border: 1px solid #4a4a4a;
  }

  .marquee p {
    position: absolute;
    width: 100%;
    height: 100%;
    margin: 0;
    line-height: 50px;
    text-align: center;
    -moz-transform: translateX(100%);
    -webkit-transform: translateX(100%);
    transform: translateX(100%);
    -moz-animation: scroll-left 2s linear infinite;
    -webkit-animation: scroll-left 2s linear infinite;
    animation: scroll-left 20s linear infinite;
  }

  @-moz-keyframes scroll-left {
    0% {
      -moz-transform: translateX(100%);
    }

    100% {
      -moz-transform: translateX(-100%);
    }
  }

  @-webkit-keyframes scroll-left {
    0% {
```

```

        -webkit-transform: translateX(100%);
    }

    100% {
        -webkit-transform: translateX(-100%);
    }
}

@keyframes scroll-left {
    0% {
        -moz-transform: translateX(100%);
        -webkit-transform: translateX(100%);
        transform: translateX(100%);
    }

    100% {
        -moz-transform: translateX(-100%);
        -webkit-transform: translateX(-100%);
        transform: translateX(-100%);
    }
}

.form-group.required .control-label:after {
    color: #d00;
    content: "*";
    position: absolute;
    margin-left: 3px;
    top: 4px;
}

label {
    font-weight: 700;
    font-size: 14px;
    color: black;
}

#formbody {
    height: 1100px;
    width: auto;
    background-image: url(".\\vadim-sherbakov-d6ebY-faOO0-unsplash.jpg");
    background-position: center;
    background-size: cover;
}

#formbody>#cont {
    width: 50%;
    height: auto;
    background-color: rgb(241, 241, 241);
    opacity: 0.9;
}

```

```
}
```

```
#footer {  
  height: 50px;  
  width: auto;  
  background-color: black;  
  color: white;  
  font-size: 14px;  
}
```

```
input[type="text"],  
textarea {  
  background-color: white;  
  opacity: 1;  
}
```

```
.error {  
  font-size: 16px;  
  margin: 0;  
  padding: 0;  
  color: red;  
}
```

```
.btn-group-vertical a {  
  border: 2px solid white;  
  background-color: transparent;  
  color: white;  
  padding: 5px;  
  margin: 3px;  
  width: 180px;  
  border-radius: 5px;  
  transition-duration: 0.4s;  
}
```

```
.btn-group-vertical a:hover {  
  background-color: white;  
  color: black;  
}
```

```
.datepicker {  
  padding: 20px;  
  border-radius: 5px;  
  background-color: #f8f9fa;  
  color: #495057;  
  box-shadow: 0 0.5rem 1rem rgba(0, 0, 0, 0.15);  
}
```

```

.datepicker table tr td,
.datepicker table tr th {
    border-radius: 20px;
    height: 32px;
    width: 32px;
    font-size: 0.875rem;
}

.datepicker table tr td.active,
.datepicker table tr td.active:hover,
.datepicker table tr td.active.disabled,
.datepicker table tr td.active.disabled:hover {
    background-color: #007bff;
    color: #fff;
}

.datepicker .datepicker-switch,
.datepicker .prev,
.datepicker .next {
    font-size: 1.25rem;
}

.datepicker .datepicker-switch:hover,
.datepicker .prev:hover,
.datepicker .next:hover {
    background-color: #e9ecef;
}

.datepicker .datepicker-switch {
    width: 220px;
}
</style>
<script>
$(document).ready(function () {
    $('#00NGA00000UJCNw').datepicker({
        format: 'dd/mm/yyyy',
        changeMonth: true,
        changeYear: true,
        autoclose: true,
        todayHighlight: true,
        clearBtn: true
    });
});

function formSubmit() {
    var dateOfBirth =
document.getElementById("00NGA00000UJCNw").value.split("/");

```

```

var modDOB = `${dateOfBirth[1]}/${dateOfBirth[0]}/${dateOfBirth[2]}`
var tenScore = document.getElementById("00NGA00000UJCOV").value;
var interScore = document.getElementById("00NGA00000UJCOK").value;
var declarationCheck = document.getElementById("gridCheck").checked;
var dobDate = new Date(modDOB).getTime();
var nowDate = new Date().getTime();
var time_diff = nowDate - dobDate;
var days_diff = time_diff / (1000 * 24 * 60 * 60);
//console.log(days_diff < (17 * 365));
if (days_diff < (17 * 365)) {
    document.getElementById("ageError").innerHTML = "<small>Your age
should be above 17.</small>";
    document.getElementById("00NGA00000UJCNw").focus();
    return false;
}
if (parseFloat(tenScore) < 7 || parseFloat(tenScore) > 10) {
    document.getElementById("tenError").innerHTML = "<small>10th Score
should be between 7 and 10.</small>";
    document.getElementById("00NGA00000UJCOV").focus();
    return false;
}
if (parseFloat(interScore) < 7 || parseFloat(interScore) > 10) {
    document.getElementById("interError").innerHTML = "<small>12th Score
should be between 7 and 10.</small>";
    document.getElementById("00NGA00000UJCOK").focus();
    return false;
}
if (declarationCheck == false) {
    document.getElementById("checkError").innerHTML = "<small>Check
this declation.</small>";
    document.getElementById("gridCheck").focus();
    document.getElementById("checkError").animate([ { opacity: 0 }, {
opacity: 1 } ], {
        duration: 200, // milliseconds
        iterations: 5, // repeat indefinitely
    });
    return false;
}
}
</script>
</head>

<body>
    <div class="container-fluid">
        <div id="header" class="d-flex justify-content-end align-items-center">
            <div class="btn-group-vertical mr-3" role="group" aria-label="Basic
example">
                <a class="btn btn-primary" href="https://sunrichglobal-dev-
ed.develop.my.site.com/management"

```

```

        role="button">Management Login</a>
        <a class="btn btn-primary" href="https://sunrichglobal-dev-
ed.develop.my.site.com/professors"
        role="button">Professor Login</a>
    </div>
</div>
<div class="marquee my-1">
    <p>Embark on your journey of knowledge, our gates are wide open. Seize
the moment, and submit your
    application today.</p>
</div>
<div id="formbody" class="d-flex justify-content-center align-items-center">
    <div id="cont" class="p-3 rounded font-weight-normal">
        <div>
            <p class="h2 text-center">APPLICATION FORM</p>
        </div>
        <div class="info p-1">
            <p><span style="color:#d00;font-weight: 600;">*</span> =
required</p>
        </div>
        <form action=" "
            method="POST" onsubmit="return formSubmit();"
            autocomplete="off">
            <input type="hidden" name="oid" value="00DGA000008YS98">
            <input type="hidden" name="retURL"
value="https://likeuniv.github.io/likeUniversityApplicationForm/thankyou.html">
            <div class="form-row">
                <div class="form-group col-md-6 required">
                    <label for="first_name" class="control-label">First
Name</label>
                    <input id="first_name" maxlength="40" name="first_name"
size="20" type="text"
                        class="form-control" pattern="[a-zA-Z ]+" required />
                </div>
                <div class="form-group col-md-6 required">
                    <label for="last_name" class="control-label">Last Name</label>
                    <input id="last_name" maxlength="80" name="last_name"
size="20" type="text"
                        class="form-control" pattern="[a-zA-Z ]+" required />
                </div>
            </div>
            <div class="form-row">
                <div class="form-group col-md-6 required">
                    <label for="company" class="control-label">Father/Guardian
name</label>
                    <input id="company" maxlength="40" name="company"
size="20" type="text" class="form-control"
                        pattern="[a-zA-Z ]+" />
                </div>
            </div>
        </div>
    </div>

```

```

    </div>
    <div class="form-group col-md-6 required">
      <label class="control-label">Gender</label>
      <select id="00NGA00000UJInK" name="00NGA00000UJInK"
title="Gender" class="form-control"
      required>
        <option value="">--None--</option>
        <option value="Male">Male</option>
        <option value="Female">Female</option>
        <option value="Others">Others</option>
      </select>
    </div>
  </div>

  <div class="form-row">
    <div class="form-group col-md-6 required">
      <label class="control-label">Date of Birth</label>
      <span class="dateInput dateOnlyInput"><input
id="00NGA00000UJCNw" name="00NGA00000UJCNw"
      size="12" type="text" class="form-control"
placeholder="dd/mm/yyyy"
      required /></span>
      <div id="ageError" class="error">

    </div>
  </div>
  <div class="form-group col-md-6 required">
    <label class="control-label">Department</label>
    <select id="00NGA00000UJCO1" name="00NGA00000UJCO1"
title="Department" class="form-control">
      <option value="">--None--</option>
      <option value="CSE">CSE</option>
      <option value="ECE">ECE</option>
      <option value="EEE">EEE</option>
      <option value="MECH">MECH</option>
      <option value="CIVIL">CIVIL</option>
    </select>
  </div>
</div>

<div class="form-row">
  <div class="form-group col-md-6 required">
    <label for="mobile" class="control-label">Mobile</label>
    <input id="mobile" maxlength="40" name="mobile" size="20"
type="text" class="form-control"
      pattern="[6-9]{1}[0-9]{9}" placeholder="0000000000"
required />
  </div>
  <div class="form-group col-md-6 required">

```

```

        <label for="email" class="control-label">Email</label>
        <input id="email" maxlength="80" name="email" size="20"
type="text" class="form-control"
        pattern="[a-z0-9._%+\-]+@[a-z0-9.\-]+\.[a-z]{2,}$"
placeholder="joe@testmail.com"
        required />
    </div>
</div>

<div class="form-row">
    <div class="form-group col-md-6 required">
        <label class="control-label">10th Board Name</label>
        <select id="00NGA00000UJCOQ" name="00NGA00000UJCOQ"
title="10th Board Name"
        class="form-control" required>
            <option value="">--None--</option>
            <option value="State Board">State Board</option>
            <option value="CBSE">CBSE</option>
            <option value="ICSE">ICSE</option>
        </select>
    </div>
    <div class="form-group col-md-6 required">
        <label class="control-label">10th Score</label>
        <input id="00NGA00000UJCOV" name="00NGA00000UJCOV"
size="20" type="text"
        class="form-control" placeholder="10.00"
        pattern="^(10(\.0{1,2})?|[0-9](\.[0-9]{1,2})?)$" required />
        <div id="tenError" class="error">

    </div>
</div>
</div>

<div class="form-row">
    <div class="form-group col-md-6 required">
        <label class="control-label">12th Board Name</label>
        <select id="00NGA00000UJCOf" name="00NGA00000UJCOf"
title="12th Board Name"
        class="form-control" required>
            <option value="">--None--</option>
            <option value="State Board">State Board</option>
            <option value="CBSE">CBSE</option>
            <option value="ICSE">ICSE</option>
        </select>
    </div>
    <div class="form-group col-md-6 required">
        <label class="control-label">12th Score</label>
        <input id="00NGA00000UJCOK" name="00NGA00000UJCOK"
size="20" type="text"

```



```

        class="form-control" placeholder="10.00"
        pattern="^(10(\.0{1,2})?|[0-9](\.[0-9]{1,2})?)$" required />
<div id="interError" class="error">

    </div>
</div>
</div>
<div class="form-group required">
    <label>Student Address</label>
    <textarea id="00NGA00000UJN7a" name="00NGA00000UJN7a"
rows="4" type="text"
        class="form-control"></textarea><br>
    </div>
    <div class="form-group">
        <div class="form-check">
            <input class="form-check-input" type="checkbox"
id="gridCheck">
                <label class="form-check-label" for="gridCheck">
                    <small>I hereby declare that the information provided in this
application is true and
                        correct to the best of my knowledge and belief. I
understand that any false or
                        misleading information may lead to the rejection of my
application or cancellation
                        of my admission.
                    </small>
                </label>
            <div id="checkError" class="error">

                </div>
            </div>
            <input type="submit" name="submit" class="btn btn-success">
        </form>
    </div>
</div>
<div id="footer" class="mt-1">
    <p class="copyright ml-2">&copy; 2024 Like University. All rights
reserved.</p>
</div>
</body>

</html>

```

SALESFORCE DEVELOPMENT

1) Trigger to create student from lead on admission fees payment.

LeadToStudentCreate.aptx:

```
trigger LeadToStudentCreate on Lead (after insert,after update) {  
    if(Trigger.isAfter && (Trigger.isInsert || Trigger.isUpdate)){  
        LeadToStudentHandler.createStudent(Trigger.new);  
    }  
}
```

LeadToStudentHandler.apxc:

```
public class LeadToStudentHandler {  
    public static void createStudent(List<Lead> apps) {  
        Set<String> leadDepartments = new Set<String>();  
  
        for (Lead app : apps) {  
            if (app.Admission_Fee_Paid__c) {  
                leadDepartments.add(app.Department__c);  
            }  
        }  
  
        Map<String, Id> departmentMap = new Map<String, Id>();  
  
        for (Department__c dept : [SELECT Id, Name FROM Department__c WHERE Name IN  
:leadDepartments]) {  
            departmentMap.put(dept.Name, dept.Id);  
        }  
  
        List<Student__c> studentList = new List<Student__c>();  
  
        for (Lead Id : apps) {  
            if (Id.Admission_Fee_Paid__c) {
```

```

Id deptId = departmentMap.get(Id.Department__c);
if (deptId != null) {
    Student__c s = new Student__c();
    s.First_Name__c = Id.FirstName;
    s.Last_Name__c = Id.LastName;
    s.Father_Guardian_Name__c = Id.Company;
    s.Phone__c = Id.MobilePhone;
    s.Email__c = Id.Email;
    s.Student_Address__c = Id.Student_Address__c;
    s.Gender__c = Id.Gender__c;
    s.X10th_Board_Name__c = Id.X10th_Board_Name__c;
    s.X10th_Score__c = Id.X10th_Score__c;
    s.X12th_Board_Name__c = Id.X12th_Board_Name__c;
    s.X12th_Score__c = Id.X12th_Score__c;
    s.Department_ID__c = deptId;
    s.Date_of_Birth__c = Id.Date_of_Birth__c;
    studentList.add(s);

} else {
    System.debug('No matching department found for Lead Department: ' +
Id.Department__c);
}
}
}

```

```

if (!studentList.isEmpty()) {
    try {
        insert studentList;
    } catch (DmlException e) {
        System.debug('Error inserting students: ' + e.getMessage());
    }
}

```

```

    }
}
}
}

```

2) Trigger to create all subjects when fees is paid for particular semester.

FeesPaidToResult.apxt:

```

trigger FeesPaidToResult on Fees_Payment__c (after insert,after update) {
    if(Trigger.isAfter && (Trigger.isUpdate || Trigger.isInsert)){
        FeesPaidToSemHandler.createResults(Trigger.new);
    }
}

```

FeesPaidToResultHandler.apxc:

```

public class FeesPaidToResultHandler {
    public static void createResults(List<Fees_Payment__c> feesList){

        Set<Id> studentIds = new Set<Id>();

        for (Fees_Payment__c fees : feesList) {
            if(fees.Paid__c == true){
                studentIds.add(fees.Student_ID__c);
            }
        }

        Map<Id, Student__c> studentMap = new Map<Id, Student__c>([SELECT Id,
        Department_ID__c FROM Student__c WHERE Id IN :studentIds]);
    }
}

```

```

Map<Id, Set<String>> deptSemesterMap = new Map<Id, Set<String>>();
for (Fees_Payment__c fees : feesList) {
    Student__c student = studentMap.get(fees.Student_ID__c);
    if (student != null) {
        if (!deptSemesterMap.containsKey(student.Department_ID__c)) {
            deptSemesterMap.put(student.Department_ID__c, new Set<String>());
        }
        deptSemesterMap.get(student.Department_ID__c).add(fees.Semester__c);
    }
}

```

```

List<Subject__c> subjectList = [SELECT Id, Name, Department_Id__c, Semester__c
                                FROM Subject__c
                                WHERE Department_id__c IN :deptSemesterMap.keySet()];

```

```

Map<String, List<Subject__c>> subjectMap = new Map<String,
List<Subject__c>>();
for (Subject__c sub : subjectList) {
    String key = sub.Department_id__c + '_' + sub.Semester__c;
    if (!subjectMap.containsKey(key)) {
        subjectMap.put(key, new List<Subject__c>());
    }
    subjectMap.get(key).add(sub);
}
List<Result__c> resList = new List<Result__c>();

```

```

for (Fees_Payment__c fees : feesList) {
    Student__c student = studentMap.get(fees.Student_ID__c);
    if (student != null) {
        String key = student.Department_id__c + '_' + fees.Semester__c;
        List<Subject__c> subjects = subjectMap.get(key);
    }
}

```

```

        if (subjects != null) {
            for (Subject__c sub : subjects) {
                Result__c res = new Result__c();
                res.Student_ID__c = student.Id;
                res.Semester__c = fees.Semester__c;
                res.Subject_Code__c = sub.Id;
                resList.add(res);
            }
        }
    }
}

if (!resList.isEmpty()) {
    try {
        insert resList;
    } catch (DmlException e) {
        System.debug('Failed to insert results: ' + e.getMessage());
    }
}

}

} {

public static void createResults(List<Fees_Payment__c> feesList){

    Set<Id> studentIds = new Set<Id>();

    for (Fees_Payment__c fees : feesList) {
        if(fees.Paid__c == true){
            studentIds.add(fees.Student_ID__c);
        }
    }
}

```

```
Map<Id, Student__c> studentMap = new Map<Id, Student__c>([SELECT Id,
Department_ID__c FROM Student__c WHERE Id IN :studentIds]);
```

```
Map<Id, Set<String>> deptSemesterMap = new Map<Id, Set<String>>();
for (Fees_Payment__c fees : feesList) {
    Student__c student = studentMap.get(fees.Student_ID__c);
    if (student != null) {
        if (!deptSemesterMap.containsKey(student.Department_ID__c)) {
            deptSemesterMap.put(student.Department_ID__c, new Set<String>());
        }
        deptSemesterMap.get(student.Department_ID__c).add(fees.Semester__c);
    }
}
```

```
List<Subject__c> subjectList = [SELECT Id, Name, Department_Id__c, Semester__c
                                FROM Subject__c
                                WHERE Department_id__c IN :deptSemesterMap.keySet()];
```

```
Map<String, List<Subject__c>> subjectMap = new Map<String,
List<Subject__c>>();
```

```
for (Subject__c sub : subjectList) {
    String key = sub.Department_id__c + '_' + sub.Semester__c;
    if (!subjectMap.containsKey(key)) {
        subjectMap.put(key, new List<Subject__c>());
    }
    subjectMap.get(key).add(sub);
}
```

```
List<Result__c> resList = new List<Result__c>();
```

```
for (Fees_Payment__c fees : feesList) {
    Student__c student = studentMap.get(fees.Student_ID__c);
```

```

if (student != null) {
    String key = student.Department_id__c + '_' + fees.Semester__c;
    List<Subject__c> subjects = subjectMap.get(key);
    if (subjects != null) {
        for (Subject__c sub : subjects) {
            Result__c res = new Result__c();
            res.Student_ID__c = student.Id;
            res.Semester__c = fees.Semester__c;
            res.Subject_Code__c = sub.Id;
            resList.add(res);
        }
    }
}

if (!resList.isEmpty()) {
    try {
        insert resList;
    } catch (DmlException e) {
        System.debug('Failed to insert results: ' + e.getMessage());
    }
}
}

```

3) Scheduled apex class that rejects application status in lead if admission fees not paid within mentioned timeframe.

RejectingUnacceptingApplications.apxc:

```
public class RejectingUnacceptingApplications implements Schedulable {  
    public void execute(SchedulableContext sc) {  
        List<Lead> rejectionList = [SELECT Id,firstname, Admission_Fee_Paid__c,  
Days_Left_for_Rejection__c, Application_Status__c FROM Lead WHERE  
Application_Status__c = 'Approved' AND Admission_Fee_Paid__c = false AND  
Days_Left_for_Rejection__c = 0];  
        for(Lead Id : rejectionList){  
            Id.Application_Status__c = 'Rejected';  
        }  
  
        try{  
            update rejectionList;  
        }catch(Exception e){  
            System.debug(e);  
        }  
    }  
}  
  
//Run following commands  
/*  
RejectingUnacceptingApplications rua = new RejectingUnacceptingApplications();  
String cronExp = '0 0 0 * 5,6,7 ?';  
ID jobId = System.schedule('RejectApplications', cronExp , rua);  
*/
```

Lightning Web Components(LWC):

1)Results Check LWC component:

ApexControllerClass

```
public with sharing class FetchResult {
    public FetchResult() {

    }

    @AuraEnabled(cacheable=true)
    public static List<Result__c> getStudent(String studentID,String dob,String
semester){
        List<String> dateSplit = dob.split('/');
        String modifiedDate = dateSplit.get(2)+'-'+dateSplit.get(1)+'-
'+dateSplit.get(0);
        Date bdate = Date.valueOf(modifiedDate);

        List<Student__c> s = [select Id,First_Name__c from student__c WHERE
Name=:studentID AND date_of_birth__c=:bdate LIMIT 1];

        List<Result__c> res=[select
Subject_Name__c,Department__c,Subject_CGPA__c,Subject_Grade__c,
Student_ID__r.Name,Student_ID__r.First_Name__c, Student_ID__r.Last_Name__c from
result__c where Student_ID__c=:s.get(0).Id AND Semester__c=:semester];

        return res;
    }
}
```

resultCheck.html

```
<template>
    <lightning-card title="RESULTS" icon-name="standard:search">
        <div class="slds-m-around_medium" style="font-family: 'Segoe UI', Tahoma,
Geneva, Verdana, sans-serif;">
            <div class="slds-grid slds-gutters">
                <div class="slds-col">
                    <lightning-input label="Enter Student ID :" value={sid}
onchange={sidChange} required
placeholder="LU-XXXXXX"></lightning-input>
                </div>
                <div class="slds-col">
                    <lightning-input label="Enter DOB :" value={dob}
onchange={dobChange} required
```

```

placeholder="dd/mm/yyyy"></lightning-input>
    </div>
</div>
</div>
<div class="slds-m-around_medium">
    <div class="slds-grid slds-gutters">
        <div class="slds-col slds-size_4-of-12"></div>
        <div class="slds-col slds-size_4-of-12">
            <lightning-combobox label="Select semester" value={semVal}
options={sems} onchange={semChange}
required></lightning-combobox>
        </div>
        <div class="slds-col slds-size_4-of-12"></div>
    </div>
</div>
<!-- <div class="slds-grid slds-gutters">
    <div class="slds-col slds-size_1-of-4">
    </div>
    <div class="slds-col slds-size_1-of-4">
        <button class="slds-button slds-button_success"
onclick={getResults}>Get Results</button>
    </div>
    <div class="slds-col slds-size_1-of-4">
        <button class="slds-button slds-button_destructive"
onclick={clear}>Clear</button>
    </div>
    <div class="slds-col slds-size_1-of-4">
    </div>
</div> -->
<div class="slds-m-top_medium">
    <div class="slds-align_absolute-center">
        <button class="slds-button slds-button_success slds-m-
right_medium" onclick={getResults}>Get Results</button>
        <button class="slds-button slds-button_destructive slds-m-
left_medium" onclick={clear}>Clear</button>
    </div>
</div>
</div>

<template lwc:if={disp}>
    <div class="slds-box slds-m-around_medium"
style="font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-
serif;">

```

```

        <div class="slds-align_absolute-center slds-m-top_medium slds-m-
bottom_medium">
            <span class="slds-text-color_success ">****Records
Found****</span>
        </div>
        <div>
            <dl class="slds-list_horizontal slds-wrap">
                <dt class="slds-item_label slds-text-color_weak slds-
truncate" title="First Label">Student ID
                </dt>
                <dd class="slds-item_detail slds-truncate"
title="Description for first label">{studentId}</dd>
                <dt class="slds-item_label slds-text-color_weak slds-
truncate" title="Second Label">Student Name
                </dt>
                <dd class="slds-item_detail slds-truncate"
title="Description for second label">{studentName}
                </dd>
                <dt class="slds-item_label slds-text-color_weak slds-
truncate" title="Second Label">Department
                </dt>
                <dd class="slds-item_detail slds-truncate"
title="Description for second label">{studDept}</dd>
                <dt class="slds-item_label slds-text-color_weak slds-
truncate" title="Second Label">Semester
                </dt>
                <dd class="slds-item_detail slds-truncate"
title="Description for second label">{semVal}</dd>
            </dl>
        </div>
        <table>
            <thead>
                <tr>
                    <th>SUBJECT</th>
                    <th>CGPA</th>
                    <th>GRADE</th>
                </tr>
            </thead>
            <tbody>
                <template for:each={marksDetails} for:item="md">
                    <tr key={md.Id}>
                        <td>{md.Subject_Name__c}</td>
                        <td>{md.Subject_CGPA__c}</td>
                        <td>{md.Subject_Grade__c}</td>
                    </tr>
                </template>
            </tbody>
        </table>
    </div>

```

```

        </template>
    </tbody>
</table>
<div>
    <dl class="slds-list_horizontal slds-wrap slds-m-
around_small">
        <dt class="slds-item_label slds-text-color_weak slds-
truncate" title="First Label">
            Overall CGPA</dt>
        <dd class="slds-item_detail slds-truncate"
title="Description for first label">{sumOfCg}
        </dd>
        <dt class="slds-item_label slds-text-color_weak slds-
truncate" title="Second Label">
            Overall Grade</dt>
        <dd class="slds-item_detail slds-truncate"
title="Description for second label">
            {getGrade}</dd>
    </dl>
</div>
    <div class="slds-text-align_center slds-text-color_destructive">-
-----Thank you-----</div>

</div>
</template>
<template lwc:if={errorDisp}>
    <div class="slds-box slds-m-around_medium">
        <div>
            <div>
                <div>
                    <p class="slds-text-color_error slds-text-
align_center slds-m-around_medium">!!!!No Data
                        Found!!!!</p>
                    <p class="slds-text-color_error slds-text-
align_center slds-text-heading_large">Based on
                        details you entered no data avaiable. Please
provide with correct details.</p>
                </div>
            </div>
            <div class="slds-m-around_large">
                Note : Make sure data of birth of format
"dd/mm/yyyy".
            </div>
        </div>
    </div>
</div>
</div>
</div>

```

```

    </template>
    <template lwc:if={errorDetails}>
        <div class="slds-box slds-m-around_medium"
            style="font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-
serif;">
            <div class="errorMsg">
                <div class="slds-align_absolute-center">
                    <p class="slds-text-color_error">Provide all necessary
data.</p>
                </div>
            </div>
        </div>
    </template>
</lightning-card>
</template>

```

resultCheck.js

```

import { LightningElement, track, wire } from 'lwc';
import fetchRes from '@salesforce/apex/FetchResult.getStudent';

//import getContacts from '@salesforce/apex/ContactController.getContacts';

export default class ResultCheck extends LightningElement {
    @track sid='';
    @track dob='';
    @track semVal='';
    @track sumOfCg=0;
    @track marksDetails;
    @track getGrade;
    @track filtArray;
    @track disp = false;
    @track errorDisp = false;
    @track errorDetails = false;
    @track studentName;
    @track studentId;
    @track studDept;
    @track passGrade = 'greenSpan';

    get sems(){
        return([
            { label: 'SEMESTER-1', value: 'SEM-1' },
            { label: 'SEMESTER-2', value: 'SEM-2' },
            { label: 'SEMESTER-3', value: 'SEM-3' },
            { label: 'SEMESTER-4', value: 'SEM-4' },

```

```

        { label: 'SEMESTER-5', value: 'SEM-5' },
        { label: 'SEMESTER-6', value: 'SEM-6' },
        { label: 'SEMESTER-7', value: 'SEM-7' },
        { label: 'SEMESTER-8', value: 'SEM-8' },
    ]);
}

sidChange(event){
    this.sid = event.target.value;
}

dobChange(event){
    this.dob = event.target.value;
}

semChange(event){
    this.semVal = event.target.value;
}

getResults(){
    //alert('Start');
    if(!(this.sid===' '||this.dob===' '||this.semVal===' ')){
        //alert(this.sid+' | '+this.dob+' | '+this.semVal);
        fetchRes({studentID : this.sid, dob: this.dob,semester :
this.semVal}).then((result)=>{
            if(result.length>0){
                this.errorDetails = false;
                this.errorDisp=false;
                this.disp=true;
                //console.log(result);
                this.marksDetails=result;
                this.calculateTotalCGPA();
                this.sumOfCg=(this.sumOfCg/this.marksDetails.length).toFixed(
2);

                this.gettingGrade();
                this.studentName= result[0].Student_ID__r.First_Name__c + ' '
+ result[0].Student_ID__r.Last_Name__c;
                this.studentId = result[0].Student_ID__r.Name;
                this.studDept = result[0].Department__c;
                //alert(this.disp);
            }else{
                this.errorDetails = false;
                this.disp=false;
            }
        });
    }
}

```

```

        this.errorDisp=true;
    }
}).catch((error)=>{
    this.errorDetails = false;
    this.disp=false;
    this.errorDisp=true;
    //console.log(error);
});
}else{
    this.disp=false;
    this.errorDisp=false;
    this.errorDetails = true;
}
//alert('End');
}

calculateTotalCGPA() {
    this.filtArray = this.marksDetails.filter(r=>r.Subject_CGPA__c>4);
    //console.log(this.filtArray);
    if(this.filtArray.length==this.marksDetails.length){
        this.sumOfCg = this.marksDetails.reduce((total, result) => total +
result.Subject_CGPA__c, 0);
    }else{
        this.sumOfCg=0;
        this.passGrade = 'redSpan';
    }
}

gettingGrade(){
    if(this.sumOfCg==10){
        this.getGrade='O';
    }else if(this.sumOfCg>=9 && this.sumOfCg<10){
        this.getGrade='A+';
    }else if(this.sumOfCg>=8 && this.sumOfCg<9){
        this.getGrade='A';
    }else if(this.sumOfCg>=7 && this.sumOfCg<8){
        this.getGrade='B+';
    }else if(this.sumOfCg>=6 && this.sumOfCg<7){
        this.getGrade='B';
    }else if(this.sumOfCg>=5 && this.sumOfCg<6){
        this.getGrade='C';
    }else if(this.sumOfCg>=4 && this.sumOfCg<5){
        this.getGrade='P';
    }else if(this.sumOfCg<4){
        this.getGrade='F';
    }
}

```



```

    }

    /*
    IF( Semester_2_CGPA__c = 10 , 'O',
    IF(Semester_2_CGPA__c>=9 && Semester_2_CGPA__c<10, 'A+' ,
    IF(Semester_2_CGPA__c>=8 && Semester_2_CGPA__c<9, 'A' ,
    IF(Semester_2_CGPA__c>=7 && Semester_2_CGPA__c<8, 'B+' ,
    IF(Semester_2_CGPA__c>=6 && Semester_2_CGPA__c<7, 'C' ,
    IF(Semester_2_CGPA__c>=5 && Semester_2_CGPA__c<6, 'c' ,
    IF(Semester_2_CGPA__c>=4 && Semester_2_CGPA__c<5, 'P' , 'F'))))))))
    */
}

clear(){
    this.sid='';
    this.dob='';
    this.semVal='';
    this.disp=false;
    this.errorDisp=false;
    this.errorDetails=false;
}

}

```

resultCheckt.css

```

table {
    background: #FFFFFF;
    border-radius: 0.25em;
    border-collapse: collapse;
    margin: 10px 0px;
}

th {
    border-bottom: 1px solid #DDDDDD;
    color: #333333;
    font-size: 0.85em;
    font-weight: 600;
    padding: 0.5em 1em;
    text-align: left;
    border: 1px solid #DDDDDD; /* Add this line */
}

td {

```

```
        color: #333333;
        font-weight: 400;
        padding: 0.65em 1em;
        border: 1px solid #DDDDDD; /* Add this line */
    }

    .disabled td {
        color: #999999;
    }

    tbody tr {
        transition: background 0.25s ease;
    }

    tbody tr:hover {
        background: #F5F5F5;
    }

    .greenSpan{
        color: rgb(19, 176, 19);
        font-weight: 700;
    }

    .redSpan{
        color: red;
        font-weight: 700;
    }

    dt{
        font-weight: 700;
    }

    .errorMsg{
        font-size: large;
        font-weight: 500;
    }
```

RESULTS

* Enter Student ID :

LU-000001

* Enter DOB :

08/04/2005

* Select semester

SEMESTER-1

Get Results

Clear

****Records Found****

Student ID LU-000001
Student Name Vinod Kumar
Department CSE
Semester SEM-1

SUBJECT	CGPA	GRADE
English	9	A+
Applied Physics	9	A+
Mathematics I	10	O
Computers and Information Technology	7	B+
Engineering Drawing Practice	10	O
Semiconductor Devices and Circuits	9	A+
C & Data Structures	8	A

Overall CGPA 8.86
Overall Grade A

-----Thank you-----

2)Fees Payments check LWC component:

ApexControllerClass

```
public with sharing class feesPayCheck {  
    public feesPayCheck() {  
  
    }  
  
    @AuraEnabled(cacheable=true)  
    public static List<Fees_Payment__c> getFeesPayments(String studId){  
        Student__c slist = [SELECT Id,Name FROM Student__c WHERE Name=:studId  
LIMIT 1];  
        Id sid = slist.Id;
```

```

        List<Fees_Payment__c> feesList = [SELECT Id, Name, CreatedDate,
Mode_of_payment__c, Semester__c,
Fees_Amount__c, Student_ID__r.Name, Student_ID__r.First_Name__c,
Student_ID__r.Last_Name__c, Student_ID__r.Department_ID__r.Name, Student_ID__r.Depa
rtment_ID__r.Fees_per_Semester__c FROM Fees_Payment__c WHERE Student_ID__c=:sid];

        return feesList;
    }
}

```

feesCheck.html

```

<template>
    <lightning-card title="Fees Payments Check" icon-name="utility:money">

        <div class="slds-m-around_medium">
            <div class="slds-grid slds-gutters">
                <div class="slds-col slds-size_1-of-3">

                    </div>
                    <div class="slds-col slds-size_1-of-3">
                        <lightning-input label="Enter Student ID :" value={sid}
onchange={sidChange} required
                        placeholder="LU-XXXXXX"></lightning-input>
                    </div>
                    <div class="slds-col slds-size_1-of-3">

                    </div>
                </div>
            </div>
            <div class="slds-grid slds-gutters">
                <div class="slds-col slds-size_1-of-3">
                    </div>
                <div class="slds-col slds-size_1-of-3">
                    <button class="slds-button slds-button_success slds-m-
right_large" onclick={getPayments}>Check
                        Payments</button>
                    <button class="slds-button slds-button_destructive slds-m-
left_xx-large" onclick={clear}>Clear
                        All</button>
                </div>
                <div class="slds-col slds-size_1-of-3">
                    </div>
            </div>
        </div>
    </template>

```

```

        <template lwc:if={disp}>
            <div class="slds-box slds-m-around_medium"
                style="font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-
serif;">
                <div class="slds-align_absolute-center slds-m-top_medium slds-m-
bottom_medium">
                    <span class="slds-text-color_success ">****Records
Found****</span>
                </div>
                <div>
                    <dl class="slds-list_horizontal slds-wrap">
                        <dt class="slds-item_label slds-text-color_weak slds-
truncate" title="First Label">Student ID
                        </dt>
                        <dd class="slds-item_detail slds-truncate"
title="Description for first label">{studentId}</dd>
                        <dt class="slds-item_label slds-text-color_weak slds-
truncate" title="Second Label">Student Name
                        </dt>
                        <dd class="slds-item_detail slds-truncate"
title="Description for second label">{studentName}
                        </dd>
                        <dt class="slds-item_label slds-text-color_weak slds-
truncate" title="Second Label">Department
                        </dt>
                        <dd class="slds-item_detail slds-truncate"
title="Description for second label">{studDept}</dd>
                        <!-- <dt class="slds-item_label slds-text-color_weak
slds-truncate" title="Second Label">Semester
                        </dt>
                        <dd class="slds-item_detail slds-truncate"
title="Description for second label">{semVal}</dd> -->
                    </dl>
                </div>
                <table>
                    <thead>
                        <tr>
                            <th>Payment ID</th>
                            <th>Semester</th>
                            <th>Fees Paid</th>
                            <th>Payment Mode</th>
                            <th>Paid Date</th>
                        </tr>
                    </thead>
                    <tbody>

```

```

        <template for:each={paymentRes} for:item="md">
            <tr key={md.Id}>
                <td>{md.Name}</td>
                <td>{md.Semester__c}</td>
                <td>{md.Fees_Amount__c}</td>
                <td>{md.Mode_of_payment__c}</td>
                <td>{md.CreatedDate}</td>
            </tr>
        </template>
    </tbody>
</table>
<div>
    <dl class="slds-list_horizontal slds-wrap slds-m-
around_small">
        <dt class="slds-item_label slds-text-color_weak slds-
truncate" title="First Label">
            Total Course Fees</dt>
        <dd class="slds-item_detail slds-truncate"
title="Description for first label">{totalFees}
        </dd>
        <dt class="slds-item_label slds-text-color_weak slds-
truncate" title="Second Label">
            Amount Paid</dt>
        <dd class="slds-item_detail slds-truncate"
title="Description for second label">
            {feesPaid}</dd>
        <dt class="slds-item_label slds-text-color_weak slds-
truncate" title="Second Label">
            Due Amount</dt>
        <dd class="slds-item_detail slds-truncate"
title="Description for second label">{feesDue}</dd>
    </dl>
</div>
    <div class="slds-text-align_center slds-text-color_destructive">-
-----Thank you-----</div>
</div>
</template>
<template lwc:if={errorDisp}>
    <div class="slds-box slds-m-around_medium"
style="font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-
serif;">
        <div>
            <div>
                <div>

```

```

                <p class="slds-text-color_error slds-text-align_center slds-m-around_medium">!!!!No Data Found!!!!</p>
                <p class="slds-text-color_error slds-text-align_center slds-text-heading_large">Based on details you entered no data available. Please provide with correct details.</p>
            </div>

        </div>
    </div>
</template>
<template lwc:if={errorDetails}>
    <div class="slds-box slds-m-around_medium" style="font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;">
        <div class="errorMsg">
            <div class="slds-align_absolute-center">
                <p class="slds-text-color_error">Provide all necessary
data.</p>
            </div>
        </div>
    </div>
</template>
</lightning-card>
</template>

```

feesCheck.js

```

import { LightningElement, track } from 'lwc';
import getFeesPayments from '@salesforce/apex/feesPayCheck.getFeesPayments';

export default class FeePayCheck extends LightningElement {
    @track sid='';
    @track paymentRes;
    @track studentId;
    @track studentName;
    @track studDept;
    @track disp=false;
    @track errorDisp=false;
    @track errorDetails=false;
    @track feesPaid;
    @track feesDue;
    @track totalFees;

```

```

sidChange(event){
    this.sid = event.target.value;
}

getPayments(){
    //alert('Started'+this.sid)
    if(!(this.sid==='')){
        //alert(this.sid);
        getFeesPayments({studId : this.sid}).then((result)=>{
            if(result.length>0){
                //alert('inside res');
                this.disp=true;
                this.errorDetails = false;
                this.errorDisp=false;
                this.paymentRes=result;
                this.studentId=result[0].Student_ID__r.Name;
                this.studentName=result[0].Student_ID__r.First_Name__c + ' '
+ result[0].Student_ID__r.Last_Name__c;
                this.studDept=result[0].Student_ID__r.Department_ID__r.Name;
                //console.log(result);
                this.totalFees = (result[0].Fees_Amount__c)*8;
                this.calculatateFeesPaid();
                this.feesDue = this.totalFees - this.feesPaid;
                //console.log(this.totalFees);
                //console.log(this.feesPaid);
                //console.log(this.feesDue);

            }else{
                //alert('inside empty field');
                this.errorDetails = false;
                this.disp=false;
                this.errorDisp=true;
            }
        }).catch((error)=>{
            //alert('inside error start');
            this.errorDetails = false;
            this.disp=false;
            this.errorDisp=true;
            //console.log(error);
            //alert('inside error end');
        });
    }else{
        this.disp=false;
    }
}

```



```

        this.errorDisp=false;
        this.errorDetails = true;
    }
}

calculatateFeesPaid(){
    this.feesPaid = this.paymentRes.reduce((total,res)=>
total+res.Fees_Amount__c,0);
}

clear(){
    this.sid='';
    this.disp=false;
    this.errorDisp=false;
    this.errorDetails=false;
}
}

```

feesCheck.css

```

table {
    background: #FFFFFF;
    border-radius: 0.25em;
    border-collapse: collapse;
    margin: 10px 0px;
}

th {
    border-bottom: 1px solid #DDDDDD;
    color: #333333;
    font-size: 0.85em;
    font-weight: 600;
    padding: 0.5em 1em;
    text-align: left;
    border: 1px solid #DDDDDD; /* Add this line */
}

td {
    color: #333333;
    font-weight: 400;
    padding: 0.65em 1em;
}

```

```

        border: 1px solid #DDDDDD; /* Add this line */
    }

.disabled td {
    color: #999999;
}

tbody tr {
    transition: background 0.25s ease;
}

tbody tr:hover {
    background: #F5F5F5;
}

```

Fees Payments Check

*Enter Student ID :

****Records Found****

Student ID LU-000001
 Student Name Vinod Kumar
 Department CSE

Payment ID	Semester	Fees Paid	Payment Mode	Paid Date
PAY-000001	SEM-1	55000	IMPS	2024-04-14T04:24:57.000Z

Total Course Fees 440000
 Amount Paid 55000
 Due Amount 385000

-----Thank you-----

Note: Use above components in experience builder and adjust the permissions such that students(guest users) are able to check their results and fees payment.

THE END

