1. **CC :Calculator**

Step 1: Create Apex Class

In Developer Console:

1. Open Developer Console
2. Go to File > New > Apex Class
3. Name it CalculatorController

public class CalculatorController {

public Decimal number1 { get; set; }

public Decimal number2 { get; set; }

public String operation { get; set; }

public String result { get; set; }

public void calculate() {

try {

if (operation == 'Add') {

result = String.valueOf(number1 + number2);

} else if (operation == 'Subtract') {

result = String.valueOf(number1 - number2);

} else if (operation == 'Multiply') {

result = String.valueOf(number1 \* number2);

} else if (operation == 'Divide') {

if (number2 != 0) {

result = String.valueOf(number1 / number2);

} else {

result = 'Error: Cannot divide by zero';

}

}

} catch (Exception e) {

result = 'Error: ' + e.getMessage();

}

}

}

**Step 2: Create Visualforce Page**

1. Go to Setup → Search Visualforce Pages
2. Click **New**
3. Name: CalculatorPage
4. Paste this code:

<apex:page controller="CalculatorController">

<h1>Mathematical Calculator</h1>

<apex:form>

<apex:pageBlock title="Calculator">

<apex:pageBlockSection>

<apex:inputText value="{!number1}" label="Number 1"/>

<apex:inputText value="{!number2}" label="Number 2"/>

<apex:selectList value="{!operation}" size="1" label="Operation">

<apex:selectOption itemValue="Add" itemLabel="Add"/>

<apex:selectOption itemValue="Subtract" itemLabel="Subtract"/>

<apex:selectOption itemValue="Multiply" itemLabel="Multiply"/>

<apex:selectOption itemValue="Divide" itemLabel="Divide"/>

</apex:selectList>

<apex:commandButton value="Calculate" action="{!calculate}"/>

<apex:outputText value="Result: {!result}" style="font-weight:bold;"/>

</apex:pageBlockSection>

</apex:pageBlock>

</apex:form>

</apex:page>

Then click on save ->preview

1. **Student Mark Sheet Generator App**

**Step 1: Create Custom Objects**

Go to Setup → Object Manager and create:

🧑‍🎓 Student\_\_c (Custom Object)

* Fields:
  + Roll\_Number\_\_c → Text (Required)

📝 Mark\_\_c (Custom Object)

* Fields:
  + Student\_\_c → Lookup → Student\_\_c
  + Subject\_\_c → Text
  + Marks\_Obtained\_\_c → Number (3,0)

**Step 2: Add Sample Data**

Go to App Launcher → Student\_\_c tab:

1. Add a few student records.
2. Go to Mark\_\_c tab:
   * Add 3-5 records for each student with:
     + Subject name (e.g., Math, English)
     + Marks (e.g., 80)
     + Link each to a student using the Lookup

**Step 3: Apex Class — MarksheetGenerator**

Go to Developer Console → File → New → Apex Class, and name it MarksheetGenerator:

apex

CopyEdit

public class MarksheetGenerator {

public class SubjectMark {

public String subject { get; set; }

public Decimal mark { get; set; }

public SubjectMark(String subject, Decimal mark) {

this.subject = subject;

this.mark = mark;

}

}

public class Marksheet {

public String studentName { get; set; }

public String rollNumber { get; set; }

public List<SubjectMark> subjectMarks { get; set; }

public Decimal total { get; set; }

public Decimal percentage { get; set; }

public Marksheet() {

subjectMarks = new List<SubjectMark>();

}

}

public static Marksheet generateMarksheet(Id studentId) {

if (studentId == null) return null;

Student\_\_c student = [

SELECT Name, Roll\_Number\_\_c

FROM Student\_\_c

WHERE Id = :studentId

LIMIT 1

];

List<Mark\_\_c> marksList = [

SELECT Subject\_\_c, Marks\_Obtained\_\_c

FROM Mark\_\_c

WHERE Student\_\_c = :studentId

];

Marksheet sheet = new Marksheet();

sheet.studentName = student.Name;

sheet.rollNumber = student.Roll\_Number\_\_c;

Decimal totalMarks = 0;

for (Mark\_\_c mark : marksList) {

sheet.subjectMarks.add(new SubjectMark(mark.Subject\_\_c, mark.Marks\_Obtained\_\_c));

totalMarks += mark.Marks\_Obtained\_\_c;

}

sheet.total = totalMarks;

sheet.percentage = marksList.size() > 0 ? totalMarks / marksList.size() : 0;

return sheet;

}

}

**Step 4: Apex Controller — StudentMarksheetController**

Create another Apex class named StudentMarksheetController:

apex

CopyEdit

public class StudentMarksheetController {

public Id studentId { get; set; }

public MarksheetGenerator.Marksheet marksheet { get; set; }

public StudentMarksheetController() {

studentId = ApexPages.currentPage().getParameters().get('studentId');

if (studentId != null) {

marksheet = MarksheetGenerator.generateMarksheet(studentId);

}

}

}

**Step 5: Create Visualforce Page**

Go to Setup → Visualforce Pages → New, name it StudentMarksheetPage:

xml

CopyEdit

<apex:page controller="StudentMarksheetController">

<h1>Student Mark Sheet</h1>

<apex:form>

<apex:pageBlock title="Mark Sheet">

<apex:pageBlockSection title="Student Info">

<apex:outputText value="Name: {!marksheet.studentName}" /><br/>

<apex:outputText value="Roll No: {!marksheet.rollNumber}" />

</apex:pageBlockSection>

<apex:pageBlockSection title="Subjects & Marks">

<apex:dataTable value="{!marksheet.subjectMarks}" var="item">

<apex:column value="{!item.subject}" headerValue="Subject" />

<apex:column value="{!item.mark}" headerValue="Marks" />

</apex:dataTable>

</apex:pageBlockSection>

<apex:pageBlockSection title="Total & Percentage">

<apex:outputText value="Total: {!marksheet.total}" /><br/>

<apex:outputText value="Percentage: {!marksheet.percentage}%" />

</apex:pageBlockSection>

</apex:pageBlock>

</apex:form>

</apex:page>

**Step 6: Test the Page**

1. Copy a valid Student ID (from URL of a Student record).
2. Open Visualforce page like this:

bash

CopyEdit

/apex/StudentMarksheetPage?studentId=a02XXXXXXXXXXXX

1. You will see the full marksheet for that student.