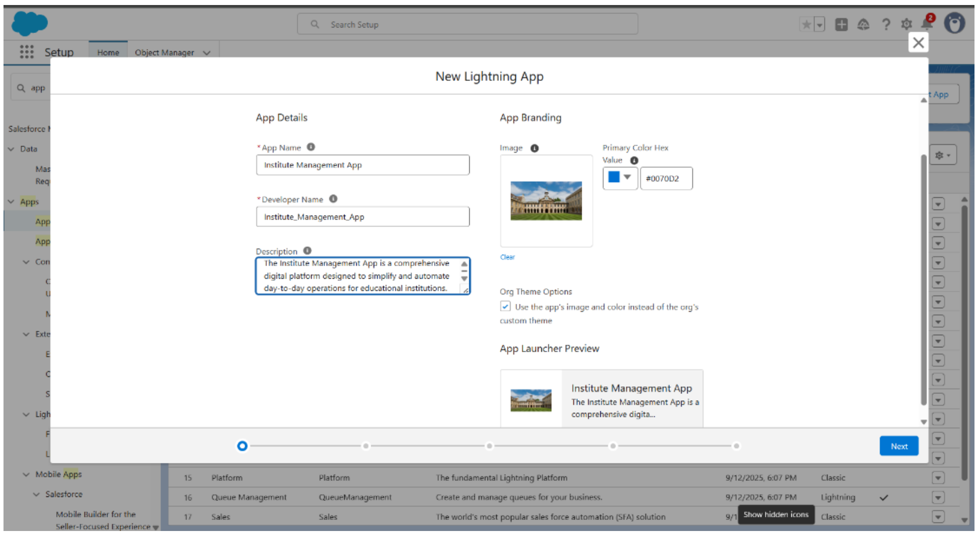
INSTITUTE MANAGEMENT SYSTEM

Phase 6: User Interface Development

A. Create the Lightning App

1. Setup → App Manager → New Lightning App.
2. Name: Institute Management. Add branding (logo/color).
3. Navigation Items: add custom object tabs (Students, Courses, Enrollments, Fee Payments) + Reports + Dashboards.
4. Utility Bar: click Add, include actions like New Enrollment, New Fee Payment, Recent Items, Notes.
5. Assign the app to Profiles (Admin, Faculty, Accountant). Save.



New Lightning App

App Details

*App Name: Institute Management App

*Developer Name: Institute_Management_App

Description: The Institute Management App is a comprehensive digital platform designed to simplify and automate day-to-day operations for educational institutions.

App Branding

Image

Primary Color Hex Value: #0070D2

Org Theme Options

✓ Use the app's image and color instead of the org's custom theme

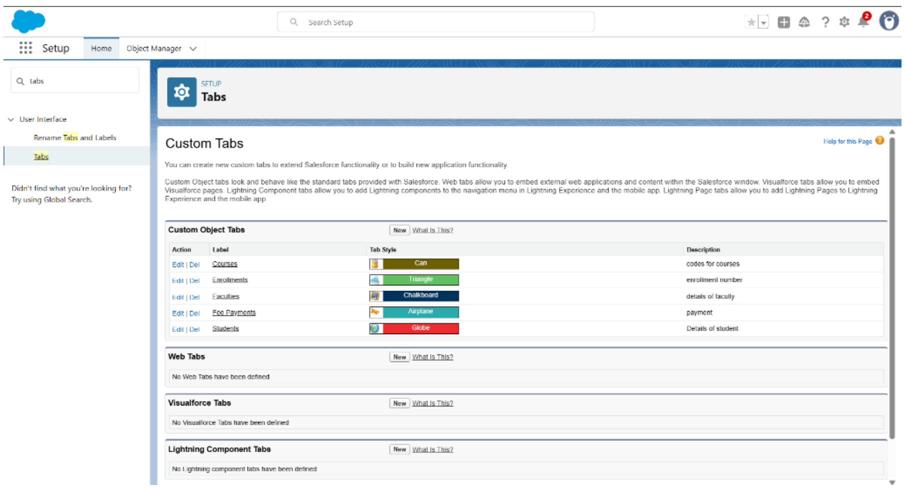
App Launcher Preview

Institute Management App
The Institute Management App is a comprehensive digita...

15	Platform	Platform	The fundamental Lightning Platform	9/12/2025, 6:07 PM	Classic	
16	Queue Management	QueueManagement	Create and manage queues for your business.	9/12/2025, 6:07 PM	Lightning	✓
17	Sales	Sales	The world's most popular sales force automation (SFA) solution	9/1	Classic	

B. Create Tabs for Objects and LWC

1. Setup → Tabs → New → Custom Object Tab: create tabs for Student__c, Course__c, Enrollment__c, Fee_Payment__c.
2. For LWC components you want as standalone tabs: Setup → Tabs → Lightning Component Tabs → New → pick your LWC and label it (done after deploying LWC).


Setup Home Object Manager

tabs

User Interface

Rename Tabs and Labels

Tabs

Didn't find what you're looking for? Try using Global Search.

SETUP

Tabs

Custom Tabs

You can create new custom tabs to extend Salesforce functionality or to build new application functionality.

Custom Object tabs look and behave like the standard tabs provided with Salesforce. Web tabs allow you to embed external web applications and content within the Salesforce window. Visualforce tabs allow you to embed Visualforce pages. Lightning Component tabs allow you to add Lightning components to the navigation menu in Lightning Experience and the mobile app. Lightning Page tabs allow you to add Lightning Pages to Lightning Experience and the mobile app.

Custom Object Tabs

Action	Label	Tab Style	Description
Edit | Del	Courses	Can	codes for courses
Edit | Del	Enrollments	Triangle	enrollment number
Edit | Del	Faculties	Chalkboard	details of faculty
Edit | Del	Fee Payments	Airplane	payment
Edit | Del	Students	Globe	Details of student

Web Tabs

No Web Tabs have been defined

Visualforce Tabs

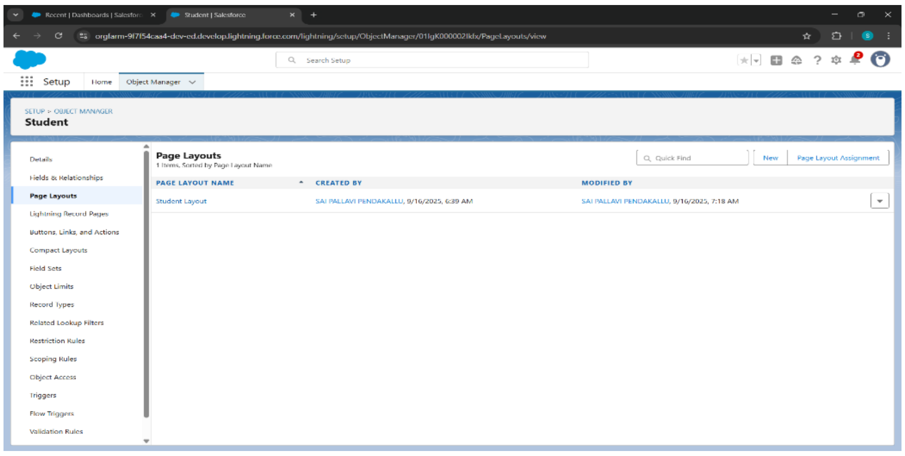
No Visualforce Tabs have been defined

Lightning Component Tabs

No Lightning component tabs have been defined

C. Page Layouts & Quick Actions

1. Setup → Object Manager → open Student__c → Page Layouts → Edit layout → drag Related Lists (Enrollments, Fee Payments). Save.


orgfarm-9f7f54caa4-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01IgK000002Ikfx/PageLayouts/view

SETUP > OBJECT MANAGER

Student

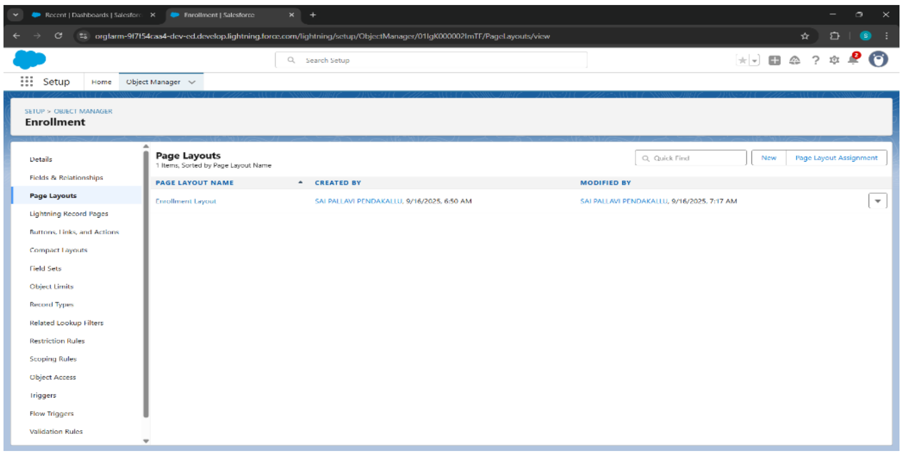
Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Restriction Rules, Scoping Rules, Object Access, Triggers, Flow Triggers, Validation Rules

Page Layouts

1 Items, Sorted by Page Layout Name

PAGE LAYOUT NAME	CREATED BY	MODIFIED BY
Student Layout	SAI PALLAVI PENDAKALLU, 9/16/2025, 6:39 AM	SAI PALLAVI PENDAKALLU, 9/16/2025, 7:18 AM

2. Setup → Object Manager → Enrollment__c → Page Layouts → add Submit for Approval or custom quick actions if needed.


orgfarm-9f7f54caa4-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01IgK000002ImTF/PageLayouts/view

SETUP > OBJECT MANAGER

Enrollment

Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Restriction Rules, Scoping Rules, Object Access, Triggers, Flow Triggers, Validation Rules

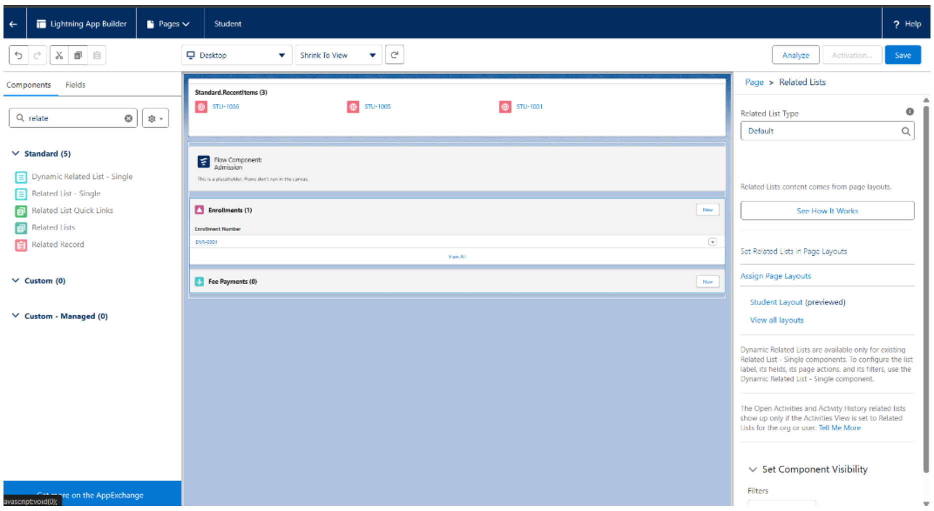
Page Layouts

1 Items, Sorted by Page Layout Name

PAGE LAYOUT NAME	CREATED BY	MODIFIED BY
Enrollment Layout	SAI PALLAVI PENDAKALLU, 9/16/2025, 6:50 AM	SAI PALLAVI PENDAKALLU, 9/16/2025, 7:17 AM

D. Build Record Pages (Lightning App Builder)

1. App Launcher → open App Builder: Setup → Lightning App Builder → New → select Record Page.
2. Choose object (e.g., Student) → choose template (Header & Right Sidebar) → name (Student Record Page).
3. Drag standard components and your custom LWC (once deployed) onto the page: Related lists, Highlights, Quick Actions.
4. Click Activation → make it the org default for the object or assign by app/profile/record type. Save & Activate.



This is a placeholder. Flows don't run in the canvas.

Related Lists content comes from page layouts.

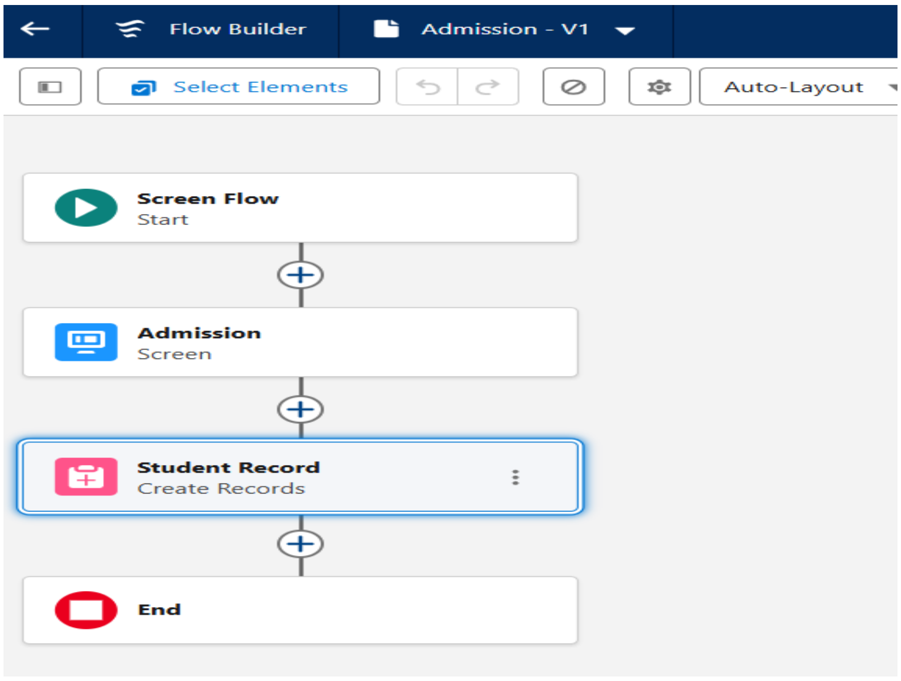
Dynamic Related Lists are available only for existing Related List - Single components. To configure the list label, its fields, its page actions, and its filters, use the Dynamic Related List - Single component.

The Open Activities and Activity History related lists show up only if the Activities View is set to Related Lists for the org or user. Tell Me More

E. Screen Flow – Student Admission Form

Purpose: Let admins create a new student easily via a guided form.

Steps:
1. Go to Setup → Flow → New Flow → Screen Flow.
2. Screen Element: Add fields for student details (Name, Email, Address, Course selection).
3. Create Records Element:
   - Object = Student
   - Map input fields to Student fields.
4. Connect elements → Save → Activate the Flow.

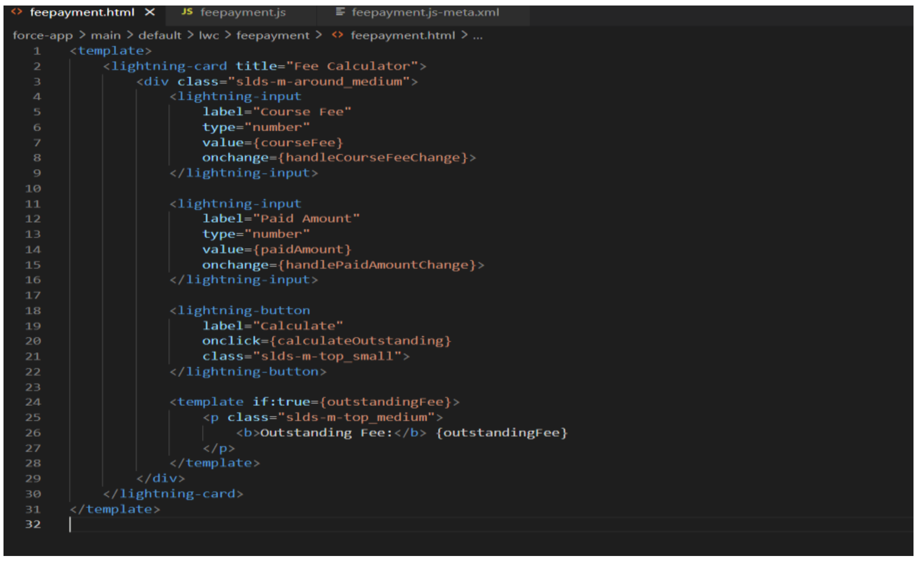


Lightning Web Component – Fee Calculator

Purpose: Allow quick calculation of fees (e.g., Course Fee – Paid Amount = Outstanding Fee).

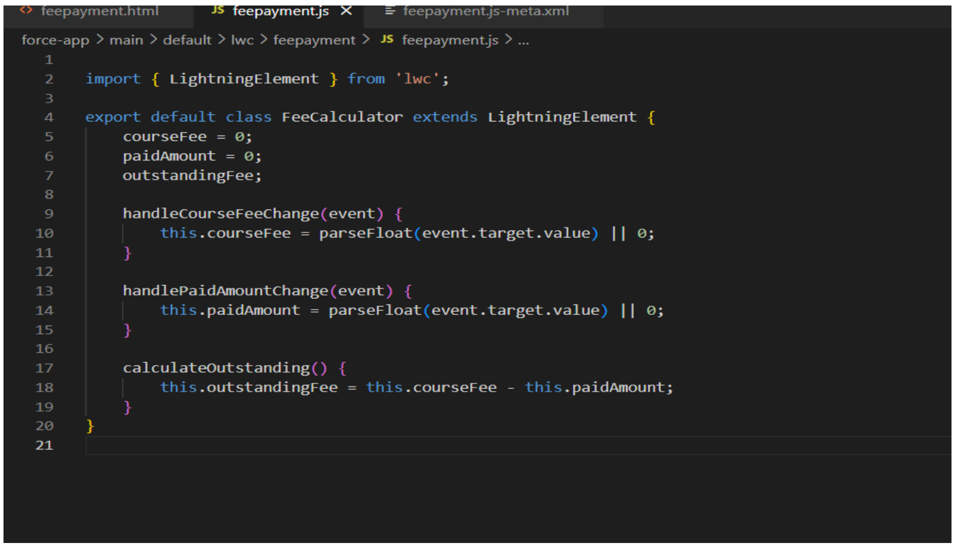
LWC Development Steps:

1. Create Project (in VS Code Command Palette):
   SFDX: Create Project with Manifest
2. Create LWC:
   SFDX: Create Lightning Web Component
   Name → feepayment
3. Edit Files:
   - feepayment.html → Create form (inputs for Course Fee & Paid Fee).


feepayment.html
feepayment.js
feepayment.js-meta.xml
force-app > main > default > lwc > feepayment > feepayment.html > ...

```html
<template>
    <lightning-card title="Fee Calculator">
        <div class="slds-m-around_medium">
            <lightning-input
                label="Course Fee"
                type="number"
                value={courseFee}
                onchange={handleCourseFeeChange}>
            </lightning-input>

            <lightning-input
                label="Paid Amount"
                type="number"
                value={paidAmount}
                onchange={handlePaidAmountChange}>
            </lightning-input>

            <lightning-button
                label="Calculate"
                onclick={calculateOutstanding}
                class="slds-m-top_small">
            </lightning-button>

            <template if:true={outstandingFee}>
                <p class="slds-m-top_medium">
                    <b>Outstanding Fee:</b> {outstandingFee}
                </p>
            </template>
        </div>
    </lightning-card>
</template>
```

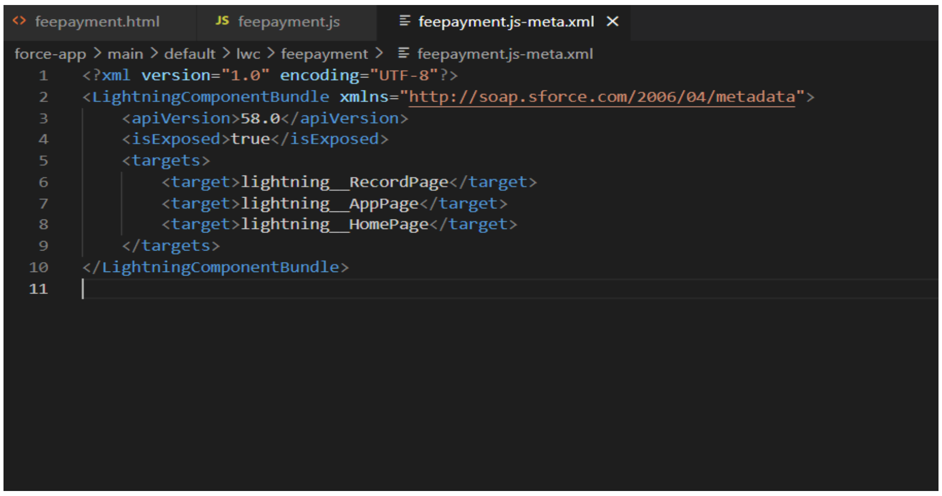
feepayment.js → Logic to calculate outstanding fee.



force-app > main > default > lwc > feepayment > feepayment.js > ...

```javascript

import { LightningElement } from 'lwc';

export default class FeeCalculator extends LightningElement {
    courseFee = 0;
    paidAmount = 0;
    outstandingFee;

    handleCourseFeeChange(event) {
        this.courseFee = parseFloat(event.target.value) || 0;
    }

    handlePaidAmountChange(event) {
        this.paidAmount = parseFloat(event.target.value) || 0;
    }

    calculateOutstanding() {
        this.outstandingFee = this.courseFee - this.paidAmount;
    }
}
```

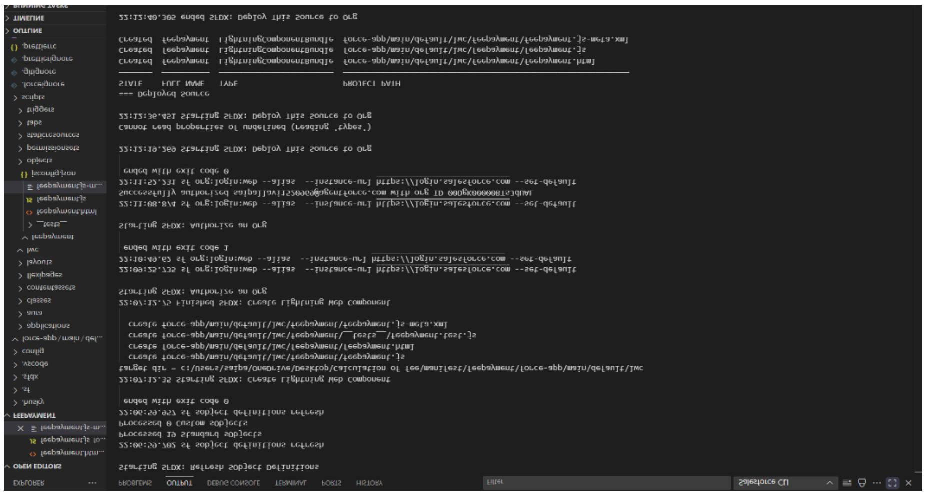
feepayment.js-meta.xml → Expose component to Record Page.



force-app > main > default > lwc > feepayment > feepayment.js-meta.xml

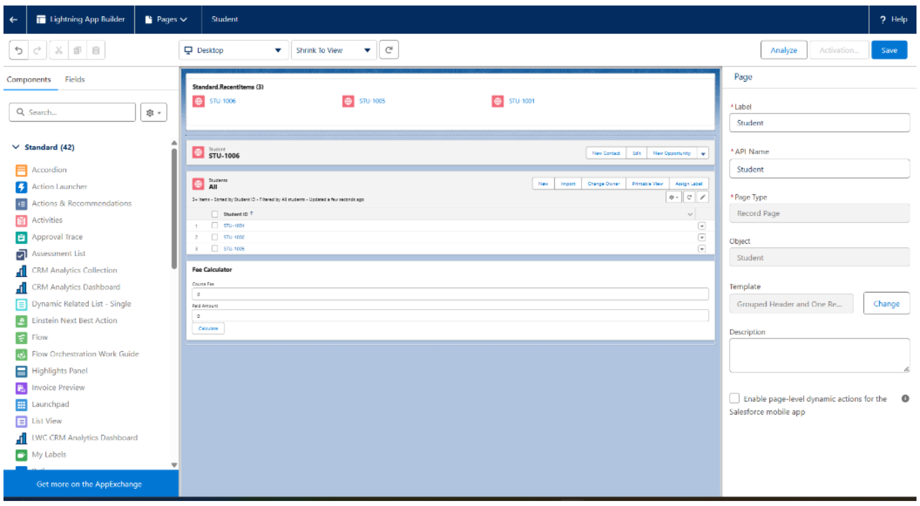
```xml
<?xml version="1.0" encoding="UTF-8"?>
<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
    <apiVersion>58.0</apiVersion>
    <isExposed>true</isExposed>
    <targets>
        <target>lightning__RecordPage</target>
        <target>lightning__AppPage</target>
        <target>lightning__HomePage</target>
    </targets>
</LightningComponentBundle>
```

Deploy Component:
  sf project deploy start --metadata LightningComponentBundle:feepayment -o



```
22:12:40.302 ended SFDX: Deploy this source to org

Created   feepayment   LightningComponentBundle   force-app/main/default/lwc/feepayment/feepayment.js-meta.xml
Created   feepayment   LightningComponentBundle   force-app/main/default/lwc/feepayment/feepayment.js
Created   feepayment   LightningComponentBundle   force-app/main/default/lwc/feepayment/feepayment.html
───────   ──────────   ────────────────────────   ──────────────────────────────────────────────────────────
STATE     FULL NAME    TYPE                       PROJECT PATH
=== Deployed Source

22:12:30.451 Starting SFDX: Deploy this source to org
Cannot read properties of undefined (reading 'types')

22:12:19.269 Starting SFDX: Deploy this source to org

 ended with exit code 0
22:11:52.231 sf org:login:web --alias  --instance-url https:\\login.salesforce.com --set-default
Successfully authorized saipallavi152066@agentforce.com with org ID 00Dgk00000BT5JduAI
22:11:08.874 sf org:login:web --alias  --instance-url https:\\login.salesforce.com --set-default

Starting SFDX: Authorize an Org

 ended with exit code 1
22:10:49.62 sf org:login:web --alias  --instance-url https:\\login.salesforce.com --set-default
22:09:25.735 sf org:login:web --alias  --instance-url https:\\login.salesforce.com --set-default

Starting SFDX: Authorize an Org
22:07:12.75 Finished SFDX: Create Lightning Web Component

  create force-app/main/default/lwc/feepayment/feepayment.js-meta.xml
  create force-app/main/default/lwc/feepayment/__tests__/feepayment.test.js
  create force-app/main/default/lwc/feepayment/feepayment.html
  create force-app/main/default/lwc/feepayment/feepayment.js
target dir = c:/Users/saipa/OneDrive/Desktop/Calculation of fee/manifest/feepayment/force-app/main/default/lwc
22:07:12.35 Starting SFDX: Create Lightning Web Component

 ended with exit code 0
22:06:59.957 sf sobject definitions refresh
Processed 0 Custom sObjects
Processed 19 Standard sObjects
22:06:59.702 sf sobject definitions refresh

Starting SFDX: Refresh SObject Definitions
```

Add to Lightning Page:
- Go to Lightning App Builder → Drag feepayment onto Student Page.



Label: Student
API Name: Student
Page Type: Record Page
Object: Student
Template: Grouped Header and One Re...
Description:

Results:

- Student Page: Displays student info and related records in a tabbed layout.
- Screen Flow: Enables guided student admission.
- Fee Calculator LWC: Interactive tool for fee management.
- Overall Outcome: Clean, intuitive, and functional UI for managing student-related data in Salesforce.