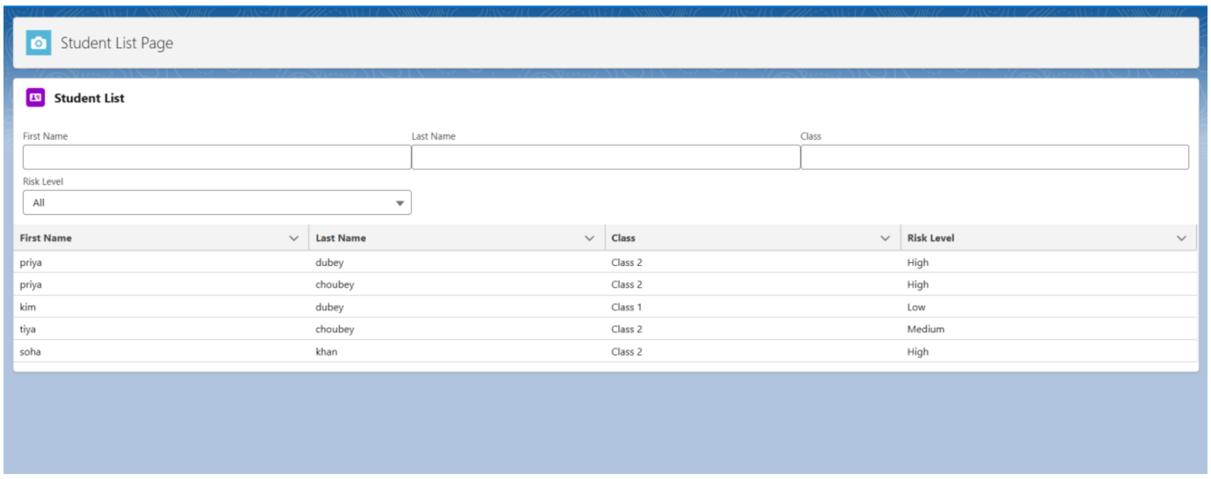
Phase 6: User Interface Development

1. Lightning App Builder

- Built Student List App Page showing all students with some fields and Risk Level filter.



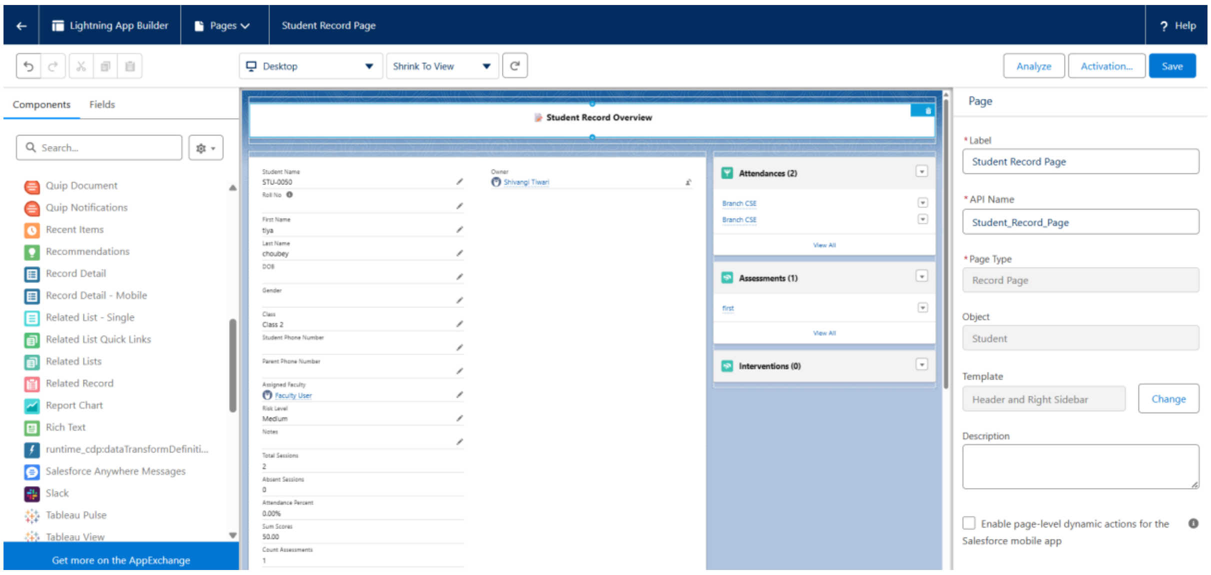
First Name	Last Name	Class

Risk Level: All

First Name	Last Name	Class	Risk Level
priya	dubey	Class 2	High
priya	choubey	Class 2	High
kim	dubey	Class 1	Low
tiya	choubey	Class 2	Medium
soha	khan	Class 2	High

2. Record Pages

- Built Student Record Page for 360° view with related lists.
- Custom layout for Student__c showing key fields: Name, Class, Risk Level.
- Related lists: Assessments, Attendance, Interventions.



Student Name: STU-0050
Owner: Shivangi Tiwari
Roll No:
First Name: tiya
Last Name: choubey
DOB:
Gender:
Class: Class 2
Student Phone Number:
Parent Phone Number:
Assigned Faculty: Faculty User
Risk Level: Medium
Notes:
Total Sessions: 2
Absent Sessions: 0
Attendance Percent: 0.00%
Sum Scores: 50.00
Count Assessments: 1

Attendances (2)
- Branch CSE
- Branch CSE

Assessments (1)
- first

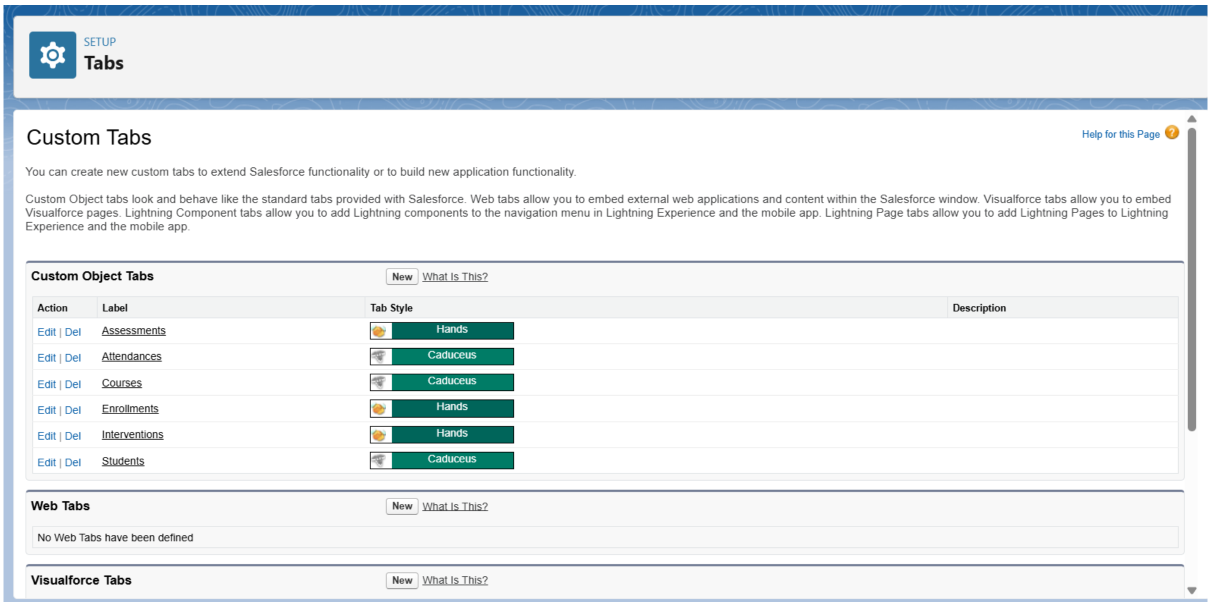
Interventions (0)

Page
- * Label: Student Record Page
- * API Name: Student_Record_Page
- * Page Type: Record Page
- Object: Student
- Template: Header and Right Sidebar
- Description:
- [ ] Enable page-level dynamic actions for the Salesforce mobile app

3. Tabs

- Added Student, Course, Assessment, Attendance, Intervention, Enrollment tabs.

- Faculty can easily navigate between objects.



SETUP
Tabs

Custom Tabs

Help for this Page

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.

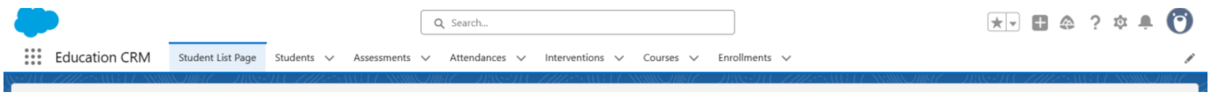
Custom Object Tabs New What Is This?

Action	Label	Tab Style	Description
Edit | Del	Assessments	Hands	
Edit | Del	Attendances	Caduceus	
Edit | Del	Courses	Caduceus	
Edit | Del	Enrollments	Hands	
Edit | Del	Interventions	Hands	
Edit | Del	Students	Caduceus	

Web Tabs New What Is This?

No Web Tabs have been defined

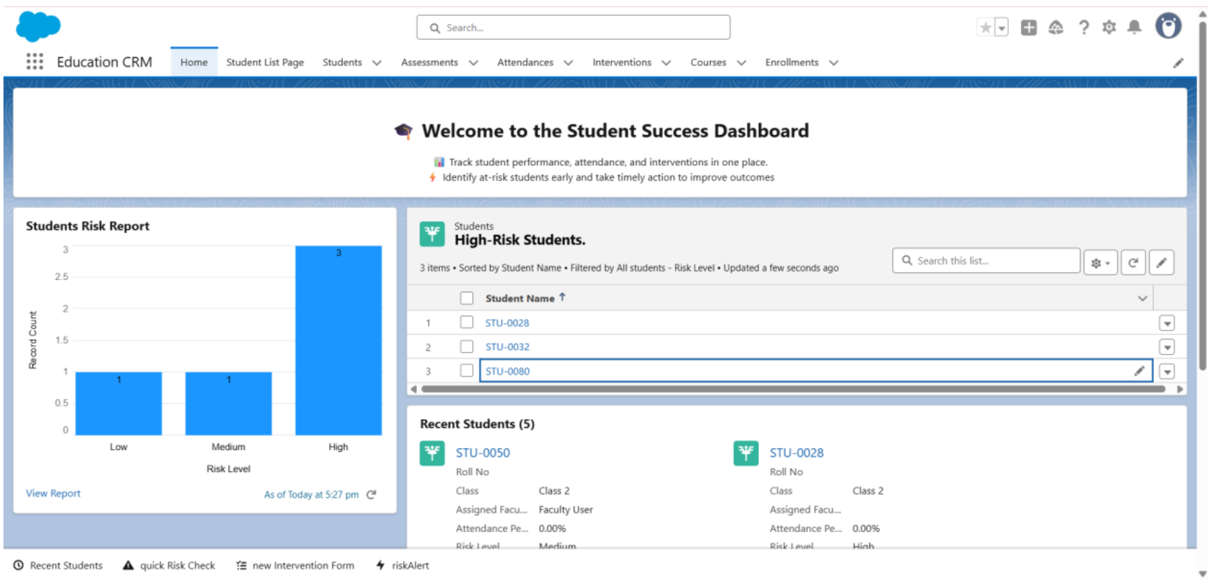
Visualforce Tabs New What Is This?



4. Home Page Layouts

Created a Student Success Dashboard home page :-

- Added Risk Level Report Chart to visualize High/Medium/Low students.
- Added High-Risk Students List which displays students needing immediate attention for timely interventions.
- Added Recent Items which provides easy access to recently viewed records and common tasks.



Welcome to the Student Success Dashboard

Track student performance, attendance, and interventions in one place.
Identify at-risk students early and take timely action to improve outcomes

Students
High-Risk Students.

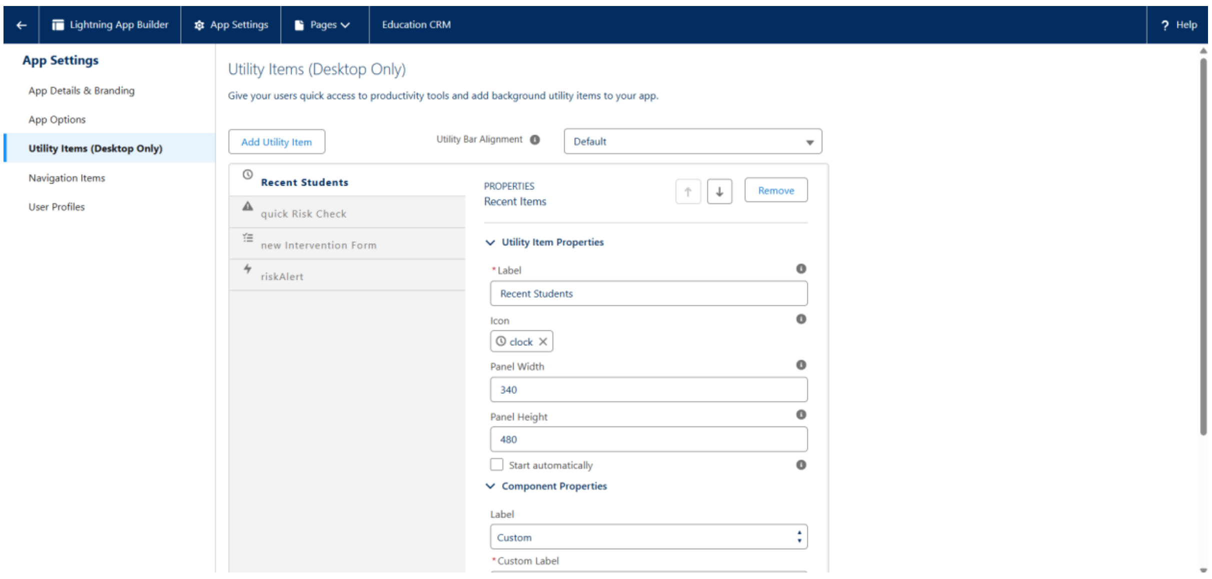
3 items • Sorted by Student Name • Filtered by All students - Risk Level • Updated a few seconds ago

	Student Name
1	STU-0028
2	STU-0032
3	STU-0080

Recent Students (5)

	STU-0050		STU-0028
Roll No		Roll No	
Class	Class 2	Class	Class 2
Assigned Facu...	Faculty User	Assigned Facu...	
Attendance Pe...	0.00%	Attendance Pe...	0.00%
Risk Level	Medium	Risk Level	High

5. Utility Bar

- Added Quick Risk Check LWC to view recent students' risk anytime.
- Added New Intervention Form LWC for quick creation of interventions.
- Added Risk Alert LWC to highlight high risk students.
- 

Utility Items (Desktop Only)

Give your users quick access to productivity tools and add background utility items to your app.

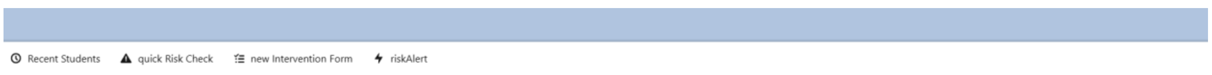
PROPERTIES
Recent Items

Utility Item Properties

*Label: Recent Students
Icon: clock
Panel Width: 340
Panel Height: 480
Start automatically: ☐

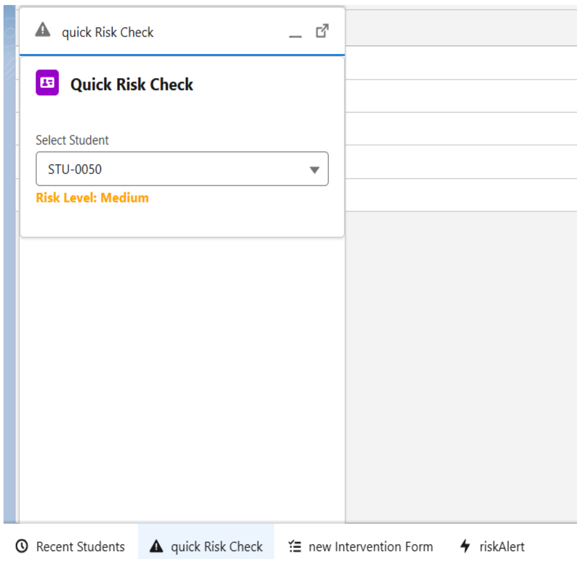
Component Properties

Label: Custom
*Custom Label



6. Lightning Web Components (LWC)

- studentList → list and filter students; click to navigate to record.
- quickRiskCheck & newInterventionForm → utility bar tools for faculty.
- riskAlert → provides the list of all students with high risk.

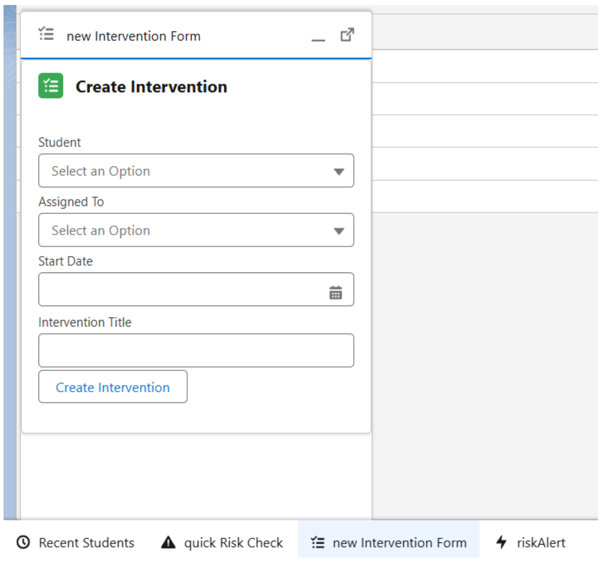


Quick Risk Check

Select Student

STU-0050

Risk Level: Medium



Create Intervention

Student

Select an Option

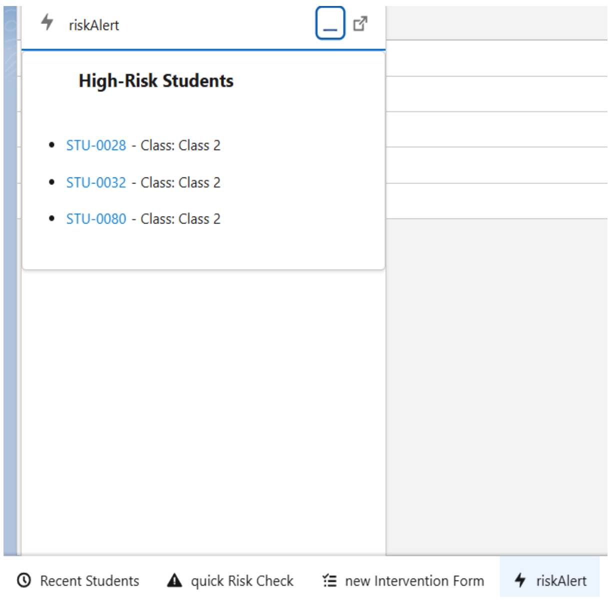
Assigned To

Select an Option

Start Date

Intervention Title

Create Intervention

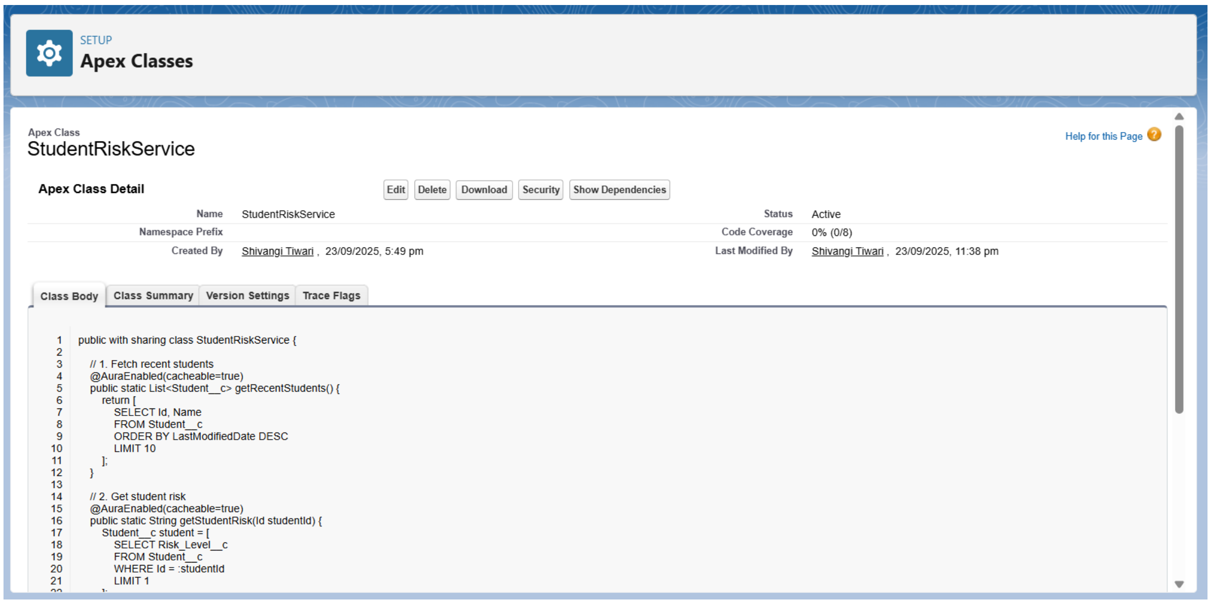


High-Risk Students

- STU-0028 - Class: Class 2
- STU-0032 - Class: Class 2
- STU-0080 - Class: Class 2

7. Apex with LWC

- Apex methods used for fetching students, filtering by risk, creating interventions-
  - getRecentStudents (StudentRiskService): Used in quickRiskCheck utility bar component to display recently viewed students for quick access.

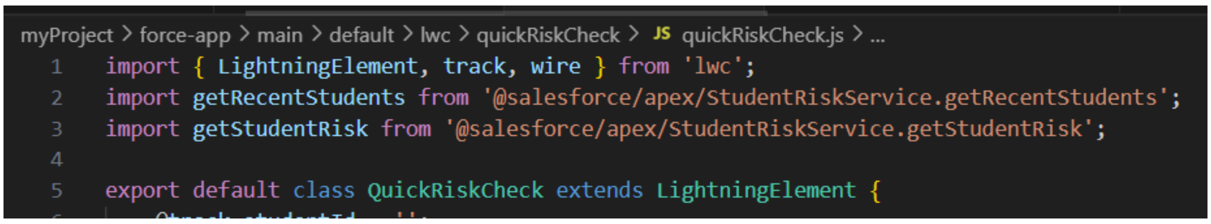


Apex Class
StudentRiskService

Apex Class Detail

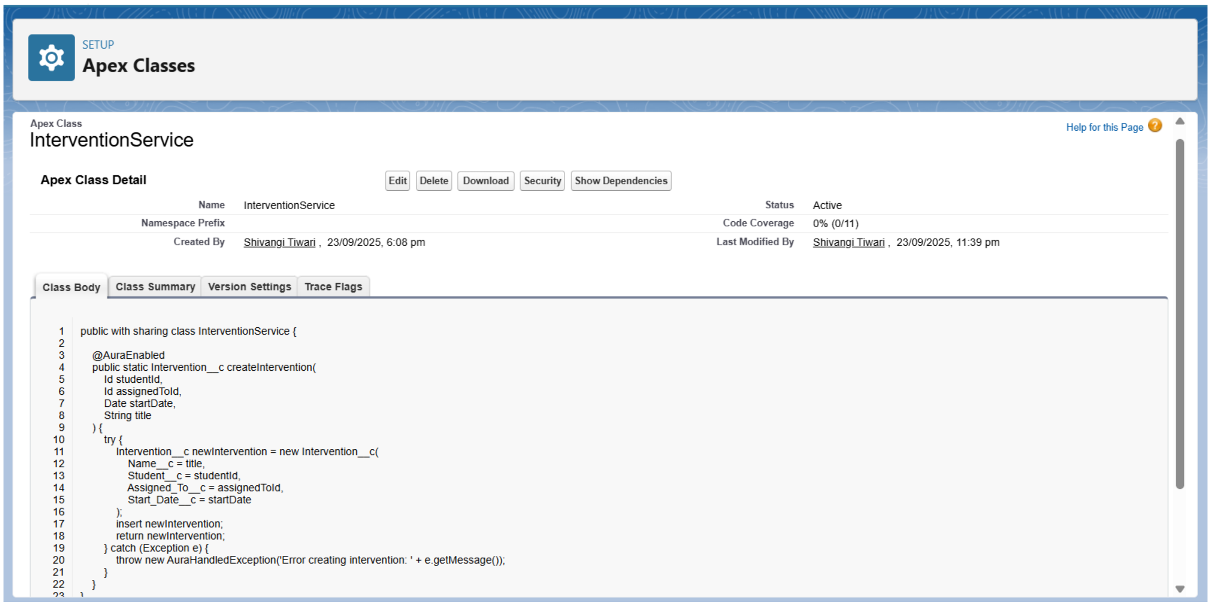
Name	StudentRiskService	Status	Active
Namespace Prefix		Code Coverage	0% (0/8)
Created By	Shivangi Tiwari , 23/09/2025, 5:49 pm	Last Modified By	Shivangi Tiwari , 23/09/2025, 11:38 pm

```
public with sharing class StudentRiskService {

  // 1. Fetch recent students
  @AuraEnabled(cacheable=true)
  public static List<Student__c> getRecentStudents() {
    return [
      SELECT Id, Name
      FROM Student__c
      ORDER BY LastModifiedDate DESC
      LIMIT 10
    ];
  }

  // 2. Get student risk
  @AuraEnabled(cacheable=true)
  public static String getStudentRisk(Id studentId) {
    Student__c student = [
      SELECT Risk_Level__c
      FROM Student__c
      WHERE Id = :studentId
      LIMIT 1
```


myProject > force-app > main > default > lwc > quickRiskCheck > JS quickRiskCheck.js > ...

```
import { LightningElement, track, wire } from 'lwc';
import getRecentStudents from '@salesforce/apex/StudentRiskService.getRecentStudents';
import getStudentRisk from '@salesforce/apex/StudentRiskService.getStudentRisk';

export default class QuickRiskCheck extends LightningElement {
```

- createIntervention (InterventionService): Used in newInterventionForm to create a new Intervention record when faculty submits the form.

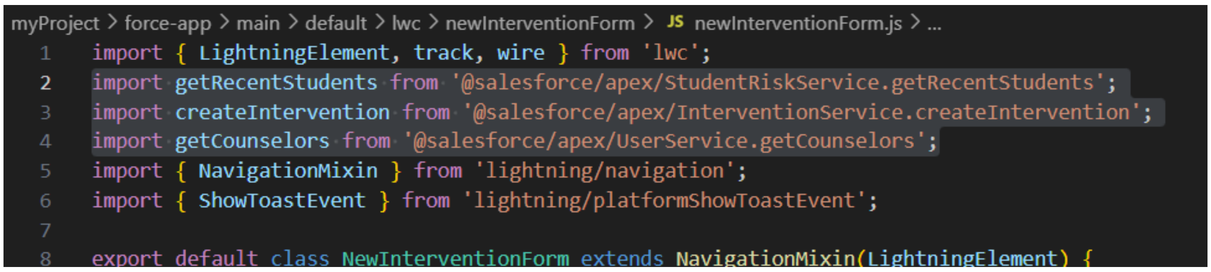


Apex Class
InterventionService

Apex Class Detail

Name	InterventionService	Status	Active
Namespace Prefix		Code Coverage	0% (0/11)
Created By	Shivangi Tiwari , 23/09/2025, 6:08 pm	Last Modified By	Shivangi Tiwari , 23/09/2025, 11:39 pm

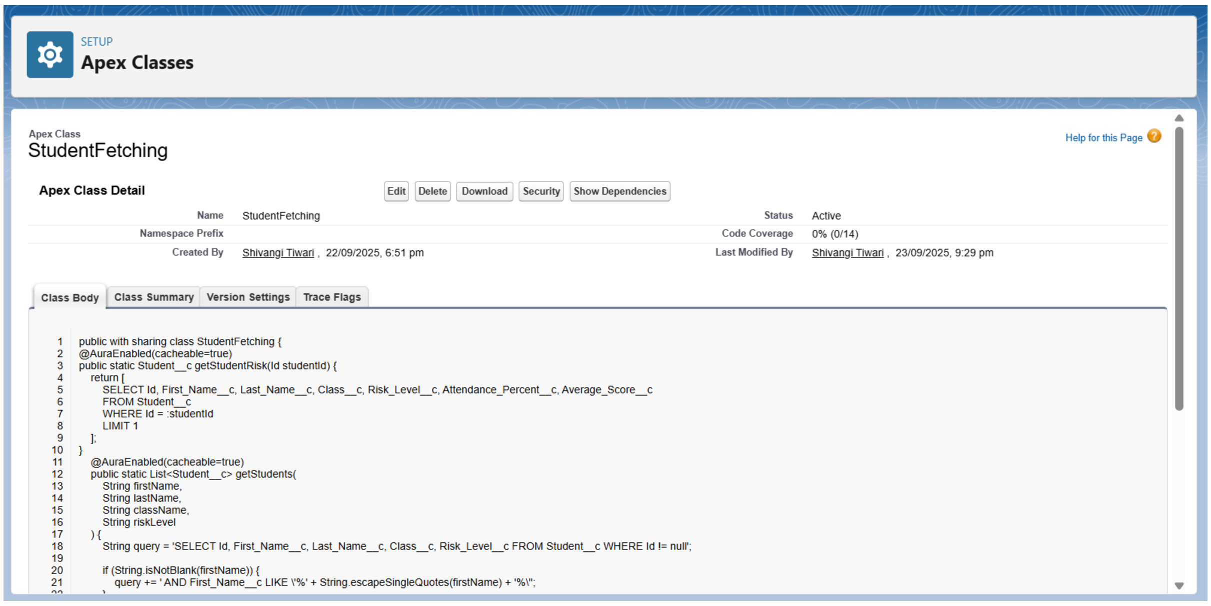
```
public with sharing class InterventionService {

    @AuraEnabled
    public static Intervention__c createIntervention(
        Id studentId,
        Id assignedToId,
        Date startDate,
        String title
    ) {
        try {
            Intervention__c newIntervention = new Intervention__c(
                Name__c = title,
                Student__c = studentId,
                Assigned_To__c = assignedToId,
                Start_Date__c = startDate
            );
            insert newIntervention;
            return newIntervention;
        } catch (Exception e) {
            throw new AuraHandledException('Error creating intervention: ' + e.getMessage());
        }
    }
```



myProject > force-app > main > default > lwc > newInterventionForm > JS newInterventionForm.js > ...

```
import { LightningElement, track, wire } from 'lwc';
import getRecentStudents from '@salesforce/apex/StudentRiskService.getRecentStudents';
import createIntervention from '@salesforce/apex/InterventionService.createIntervention';
import getCounselors from '@salesforce/apex/UserService.getCounselors';
import { NavigationMixin } from 'lightning/navigation';
import { ShowToastEvent } from 'lightning/platformShowToastEvent';

export default class NewInterventionForm extends NavigationMixin(LightningElement) {
```

- getStudents(StudentFetching): Used in studentList to fetch student records.

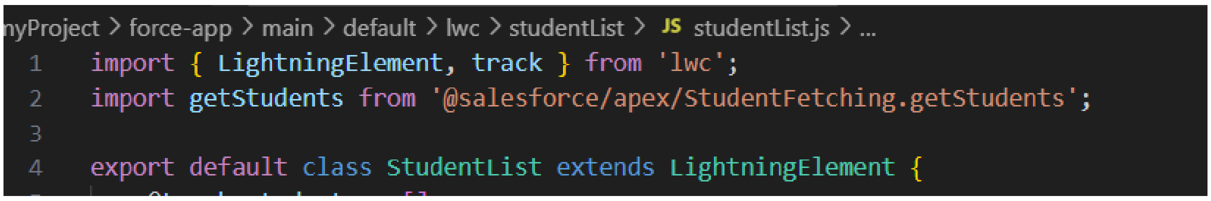


Apex Class
StudentFetching

Apex Class Detail

Name	StudentFetching	Status	Active
Namespace Prefix		Code Coverage	0% (0/14)
Created By	Shivangi Tiwari , 22/09/2025, 6:51 pm	Last Modified By	Shivangi Tiwari , 23/09/2025, 9:29 pm

```
public with sharing class StudentFetching {
@AuraEnabled(cacheable=true)
public static Student__c getStudentRisk(Id studentId) {
    return [
        SELECT Id, First_Name__c, Last_Name__c, Class__c, Risk_Level__c, Attendance_Percent__c, Average_Score__c
        FROM Student__c
        WHERE Id = :studentId
        LIMIT 1
    ];
}
    @AuraEnabled(cacheable=true)
    public static List<Student__c> getStudents(
        String firstName,
        String lastName,
        String className,
        String riskLevel
    ) {
        String query = 'SELECT Id, First_Name__c, Last_Name__c, Class__c, Risk_Level__c FROM Student__c WHERE Id != null';

        if (String.isNotBlank(firstName)) {
            query += ' AND First_Name__c LIKE \'%' + String.escapeSingleQuotes(firstName) + '%\'';
```

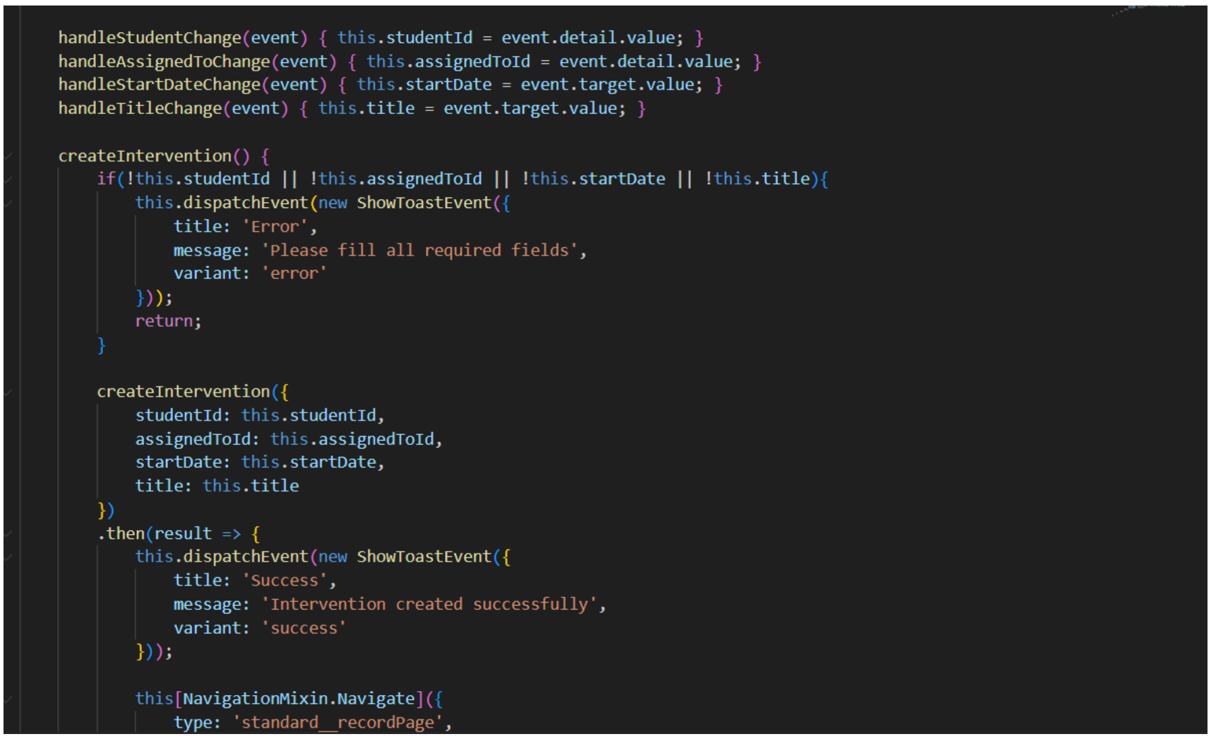

nyProject > force-app > main > default > lwc > studentList > JS studentList.js > ...

```
import { LightningElement, track } from 'lwc';
import getStudents from '@salesforce/apex/StudentFetching.getStudents';

export default class StudentList extends LightningElement {
```

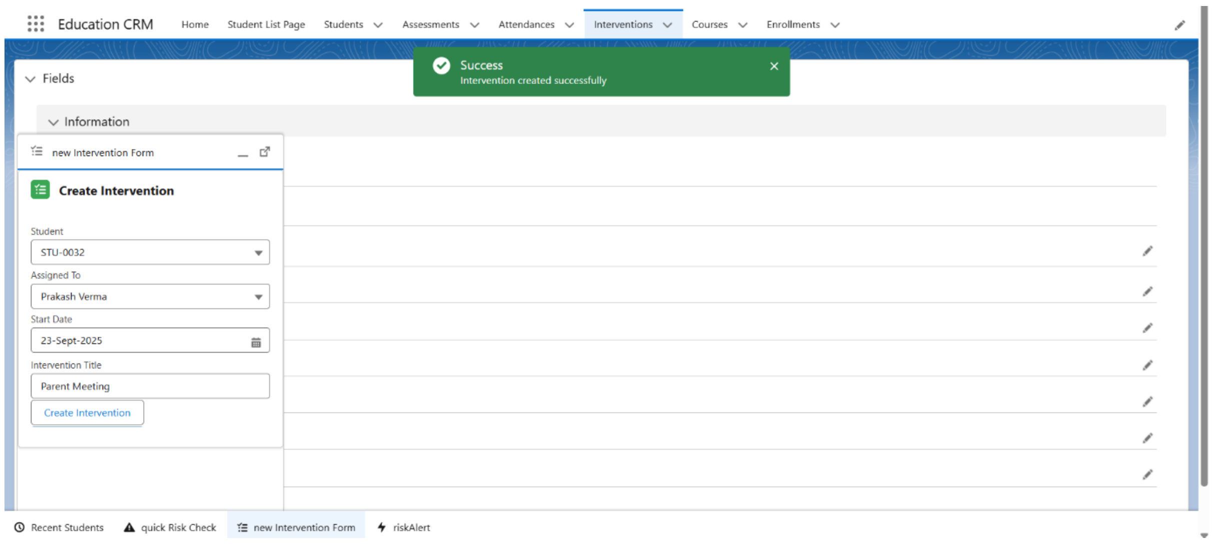
- Used @AuraEnabled for LWC integration.

8. Events in LWC

- When faculty fills all details in newInterventionForm and clicks "Create Intervention", a custom event is triggered.
- The event calls the createIntervention Apex method to create a new Intervention record in Salesforce.
- After successful creation, the LWC displays a "Intervention created successfully" message and can notify other components to refresh if needed.



```
handleStudentChange(event) { this.studentId = event.detail.value; }
handleAssignedToChange(event) { this.assignedToId = event.detail.value; }
handleStartDateChange(event) { this.startDate = event.target.value; }
handleTitleChange(event) { this.title = event.target.value; }

createIntervention() {
    if(!this.studentId || !this.assignedToId || !this.startDate || !this.title){
        this.dispatchEvent(new ShowToastEvent({
            title: 'Error',
            message: 'Please fill all required fields',
            variant: 'error'
        }));
        return;
    }

    createIntervention({
        studentId: this.studentId,
        assignedToId: this.assignedToId,
        startDate: this.startDate,
        title: this.title
    })
    .then(result => {
        this.dispatchEvent(new ShowToastEvent({
            title: 'Success',
            message: 'Intervention created successfully',
            variant: 'success'
        }));

        this[NavigationMixin.Navigate]({
            type: 'standard__recordPage',
```



Success
Intervention created successfully

new Intervention Form

Create Intervention

Student
STU-0032

Assigned To
Prakash Verma

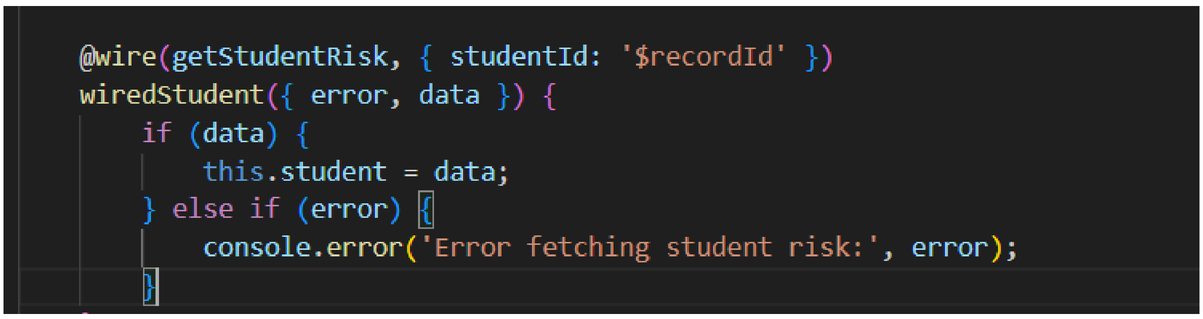
Start Date
23-Sept-2025

Intervention Title
Parent Meeting

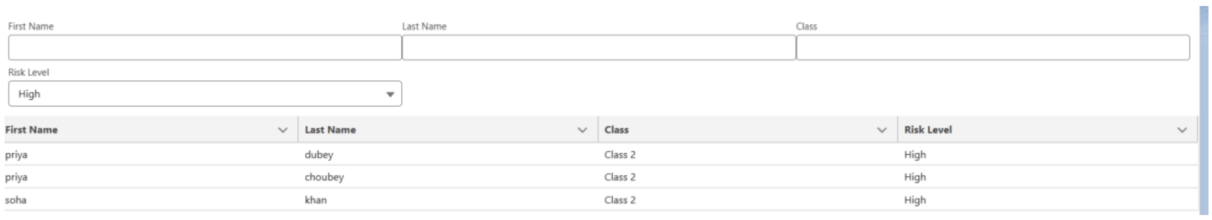
Create Intervention

9. Wire Adapters

- Used in studentList LWC to automatically fetch student records from Salesforce based on filters.
- uses a @wire Apex method to fetch students automatically based on the selected risk level.



```
@wire(getStudentRisk, { studentId: '$recordId' })
wiredStudent({ error, data }) {
    if (data) {
        this.student = data;
    } else if (error) {
        console.error('Error fetching student risk:', error);
    }
```



First Name

Last Name

Class

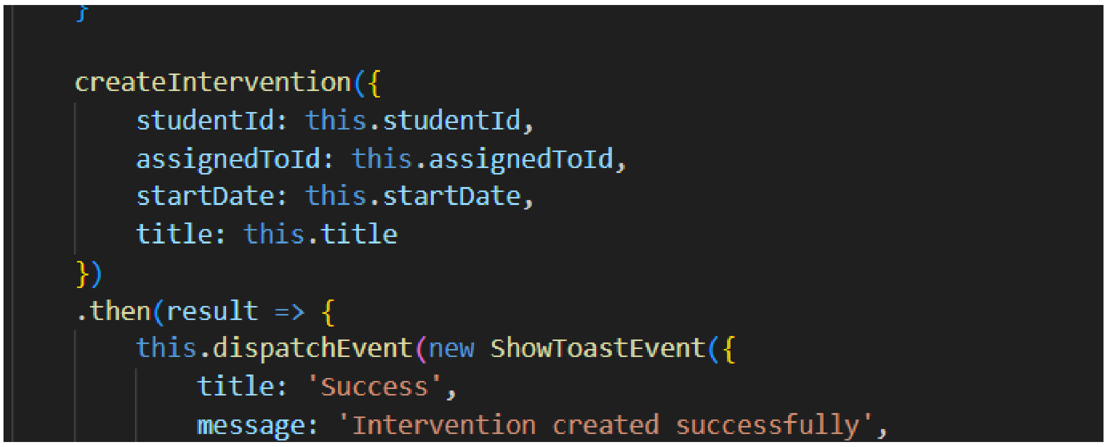
Risk Level
High

First Name	Last Name	Class	Risk Level
priya	dubey	Class 2	High
priya	choubey	Class 2	High
soha	khan	Class 2	High

10. Imperative Apex Calls

- Create Intervention called imperatively when user submits form.

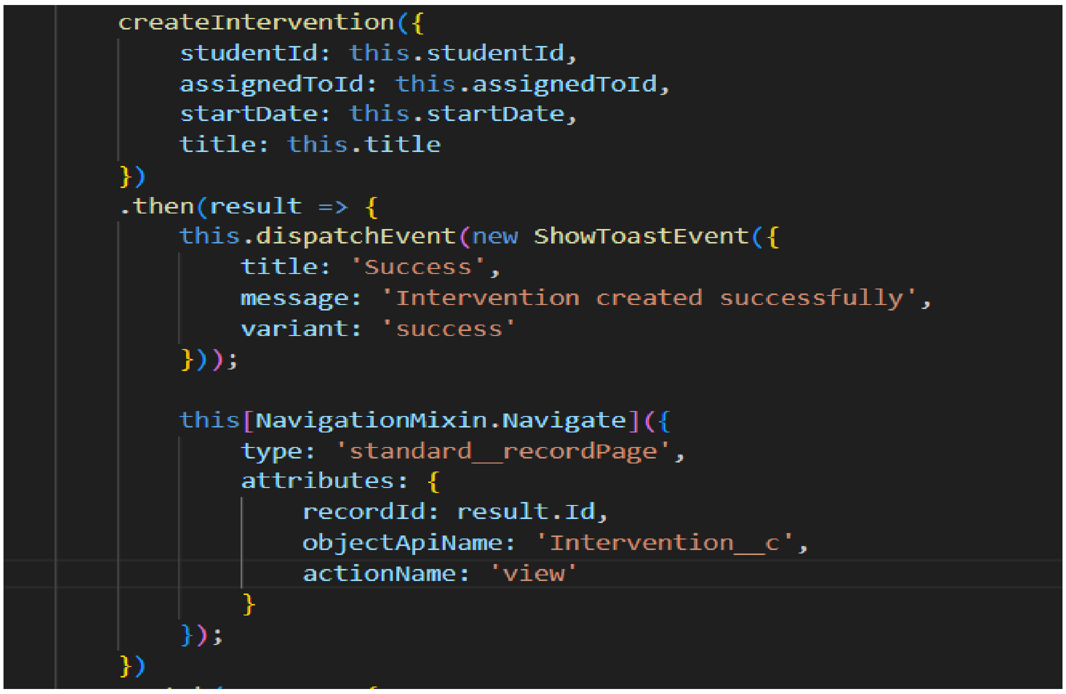
- After the intervention is created, a “Intervention created successfully” toast message is displayed to confirm the action.



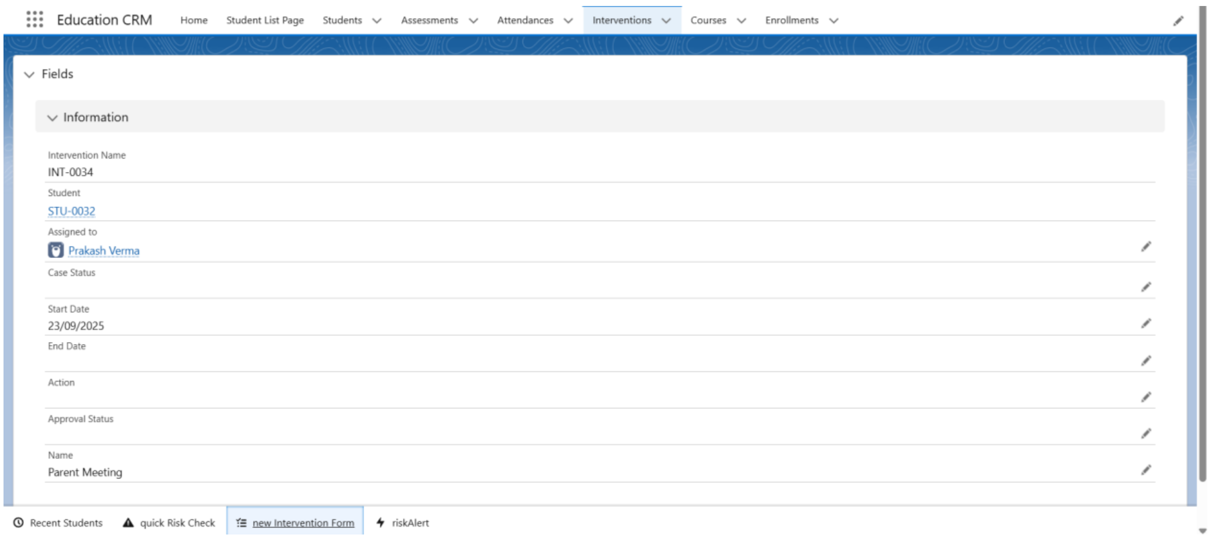
```
}

createIntervention({
    studentId: this.studentId,
    assignedToId: this.assignedToId,
    startDate: this.startDate,
    title: this.title
})
.then(result => {
    this.dispatchEvent(new ShowToastEvent({
        title: 'Success',
        message: 'Intervention created successfully',
```

11. Navigation Service

- After creating an intervention, the Navigation Service (NavigationMixin) opens the relevant Student or Intervention record page.
- Automatically redirects users after actions like creating an intervention, saving time and streamlining tasks.



```
createIntervention({
    studentId: this.studentId,
    assignedToId: this.assignedToId,
    startDate: this.startDate,
    title: this.title
})
.then(result => {
    this.dispatchEvent(new ShowToastEvent({
        title: 'Success',
        message: 'Intervention created successfully',
        variant: 'success'
    }));

    this[NavigationMixin.Navigate]({
        type: 'standard__recordPage',
        attributes: {
            recordId: result.Id,
            objectApiName: 'Intervention__c',
            actionName: 'view'
        }
    });
})
```



Intervention Name	INT-0034
Student	STU-0032
Assigned to	Prakash Verma
Case Status	
Start Date	23/09/2025
End Date	
Action	
Approval Status	
Name	Parent Meeting