Student Attendance & Performance Tracker

Phase 6: User Interface Development

Overview

Phase 6 focuses on building an intuitive and user-friendly interface using Salesforce Lightning Experience. In this phase, declarative tools like Lightning App Builder and programmatic tools like Lightning Web Components (LWC) were used to design responsive pages, enhance navigation, and improve user productivity. The goal is to ensure that teachers, admins, and users can easily access, view, and manage student attendance and enrollment information.

6.1 Lightning App Builder

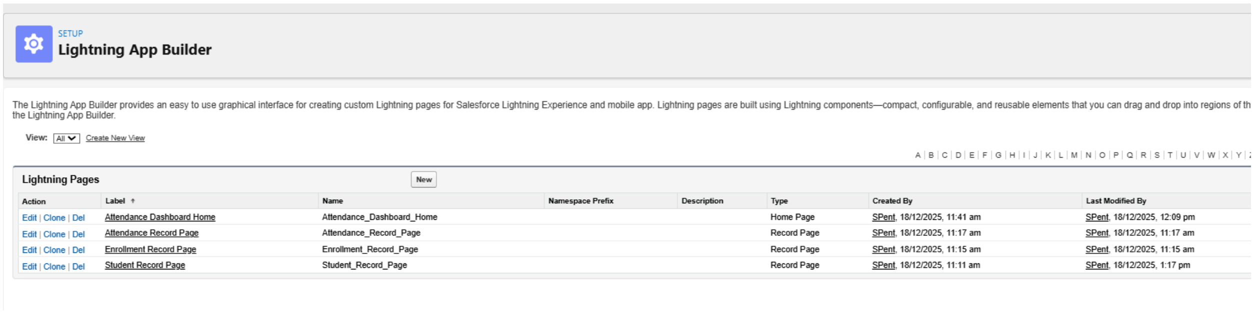
Use Case

Lightning App Builder is used to customize Salesforce pages without coding. In this project, it helps design record pages and home pages tailored for students, enrollments, and attendance tracking. This improves usability by displaying only relevant information to users.

Implementation

Lightning App Builder was used to:

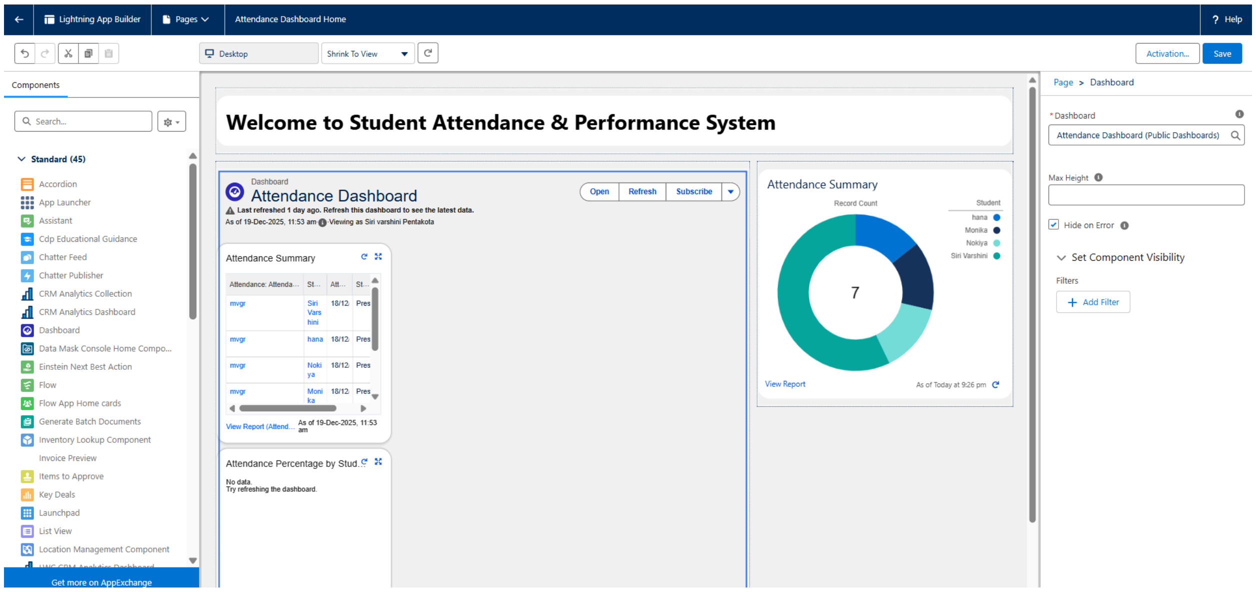
- Create custom Record Pages
- Design a custom Home Page
- Add components like Tabs, Dashboards, Related Lists, and Rich Text


SETUP
Lightning App Builder
View: All
Create New View
A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y
New

The Lightning App Builder provides an easy to use graphical interface for creating custom Lightning pages for Salesforce Lightning Experience and mobile app. Lightning pages are built using Lightning components—compact, configurable, and reusable elements that you can drag and drop into regions of th
the Lightning App Builder.

Lightning Pages

Action	Label ↑	Name	Namespace Prefix	Description	Type	Created By	Last Modified By
Edit | Clone | Del	Attendance Dashboard Home	Attendance_Dashboard_Home			Home Page	SPent, 18/12/2025, 11:41 am	SPent, 18/12/2025, 12:09 pm
Edit | Clone | Del	Attendance Record Page	Attendance_Record_Page			Record Page	SPent, 18/12/2025, 11:17 am	SPent, 18/12/2025, 11:17 am
Edit | Clone | Del	Enrollment Record Page	Enrollment_Record_Page			Record Page	SPent, 18/12/2025, 11:15 am	SPent, 18/12/2025, 11:15 am
Edit | Clone | Del	Student Record Page	Student_Record_Page			Record Page	SPent, 18/12/2025, 11:11 am	SPent, 18/12/2025, 1:17 pm



Welcome to Student Attendance & Performance System

Dashboard
Attendance Dashboard
Open | Refresh | Subscribe
Last refreshed 1 day ago. Refresh this dashboard to see the latest data.
As of 19-Dec-2025, 11:53 am · Viewing as Siri varshini Pentakota

Attendance Summary

Attendance: Attenda...	St...	Att...	St...
mvgr	Siri Varshini	18/12	Pres
mvgr	hana	18/12	Pres
mvgr	Nokiya	18/12	Pres
mvgr	Monika	18/12	Pres

View Report (Attend... As of 19-Dec-2025, 11:53 am

Attendance Percentage by Stud...
No data.
Try refreshing the dashboard.

Attendance Summary
Record Count: 7
Student: hana, Monika, Nokiya, Siri Varshini
View Report — As of Today at 9:26 pm

* Dashboard: Attendance Dashboard (Public Dashboards)
Max Height:
[✓] Hide on Error
Set Component Visibility
Filters
+ Add Filter

6.2 Record Pages

Use Case

Record Pages display all information related to a specific record. Custom record pages were created to organize data logically and reduce scrolling. Each object has its own optimized page layout.

Implemented Record Pages

- Student Record Page
- Enrollment Record Page
- Attendance Record Page

A. Student Record Page

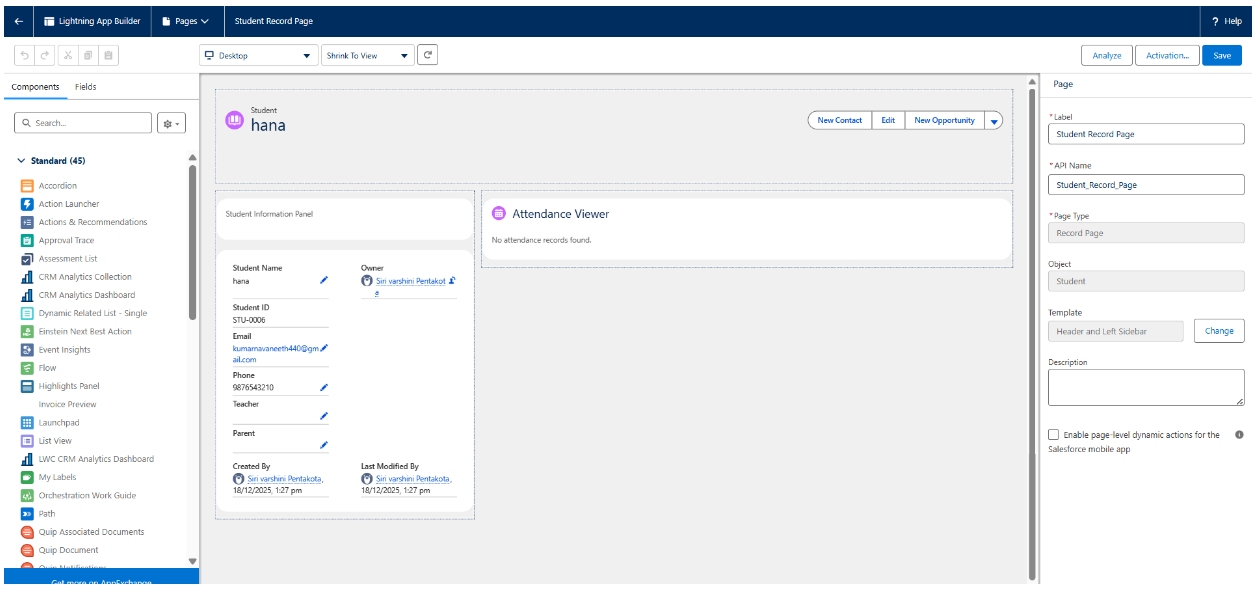
Description

The Student Record Page displays student details, enrollments, attendance, and results in a tab-based layout. This allows teachers to quickly access all student-related data from one screen.

Components Used

- Highlights Panel
- Tabs Component

- Record Detail
- Related Lists (Enrollment, Attendance, Results)
- Rich Text



Student
hana

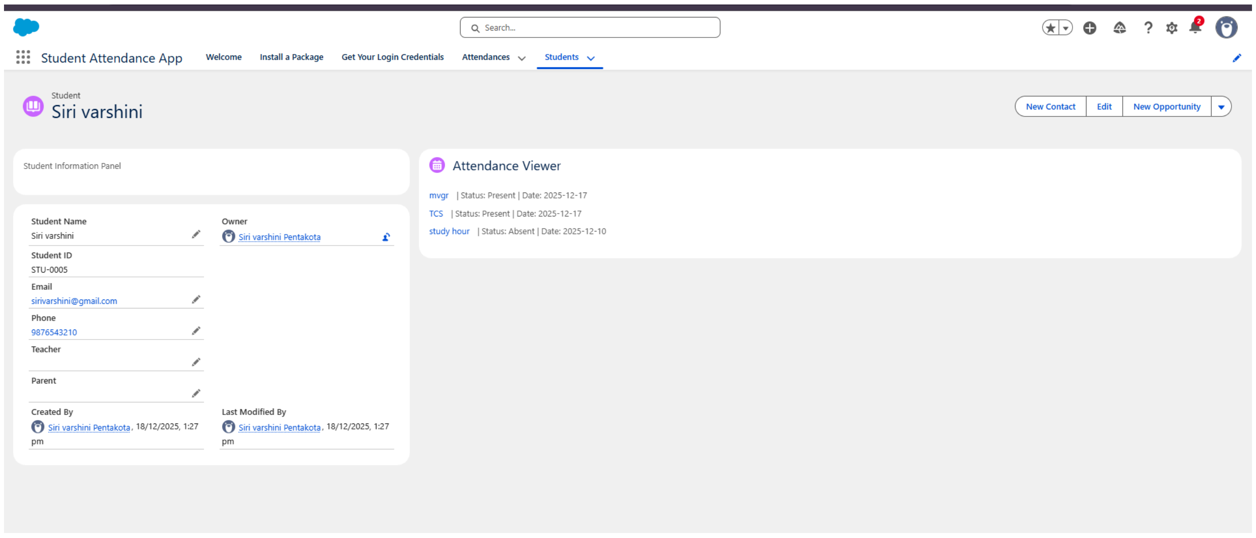
Student Information Panel

Attendance Viewer

No attendance records found.

Student Name: hana
Owner: Siri varshini Pentakota
Student ID: STU-0006
Email: kumarnavaneeth440@gmail.com
Phone: 9876543210
Teacher:
Parent:
Created By: Siri varshini Pentakota, 18/12/2025, 1:27 pm
Last Modified By: Siri varshini Pentakota, 18/12/2025, 1:27 pm

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



Student
Siri varshini

Student Information Panel

Attendance Viewer

mvgr | Status: Present | Date: 2025-12-17
TCS | Status: Present | Date: 2025-12-17
study hour | Status: Absent | Date: 2025-12-10

Student Name: Siri varshini
Owner: Siri varshini Pentakota
Student ID: STU-0005
Email: sirivarshini@gmail.com
Phone: 9876543210
Teacher:
Parent:
Created By: Siri varshini Pentakota, 18/12/2025, 1:27 pm
Last Modified By: Siri varshini Pentakota, 18/12/2025, 1:27 pm

---

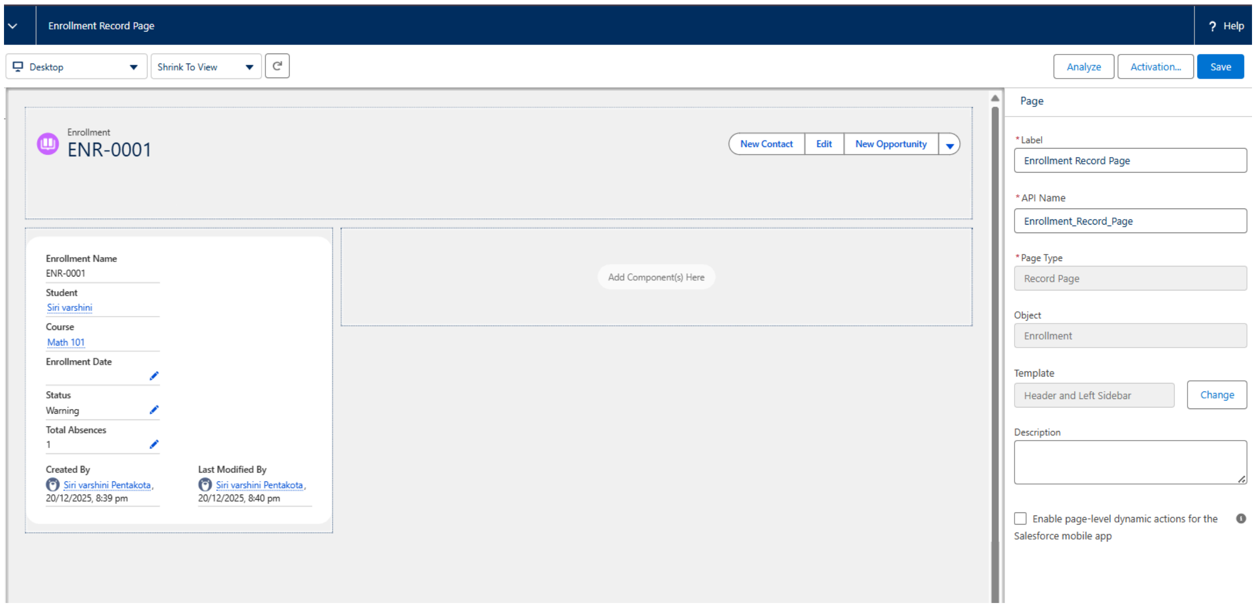
B. Enrollment Record Page

Description

The Enrollment Record Page shows enrollment details, associated student, course details, and attendance summary. It helps admins monitor student participation and enrollment status.

Tabs Used

- Enrollment Details
- Student
- Course
- Attendance Summary



Enrollment
ENR-0001

Enrollment Name
ENR-0001
Student
Siri varshini
Course
Math 101
Enrollment Date
Status
Warning
Total Absences
1
Created By
Siri varshini Pentakota, 20/12/2025, 8:39 pm
Last Modified By
Siri varshini Pentakota, 20/12/2025, 8:40 pm

Page
* Label: Enrollment Record Page
* API Name: Enrollment_Record_Page
* Page Type: Record Page
Object: Enrollment
Template: Header and Left Sidebar
Description
Enable page-level dynamic actions for the Salesforce mobile app

---

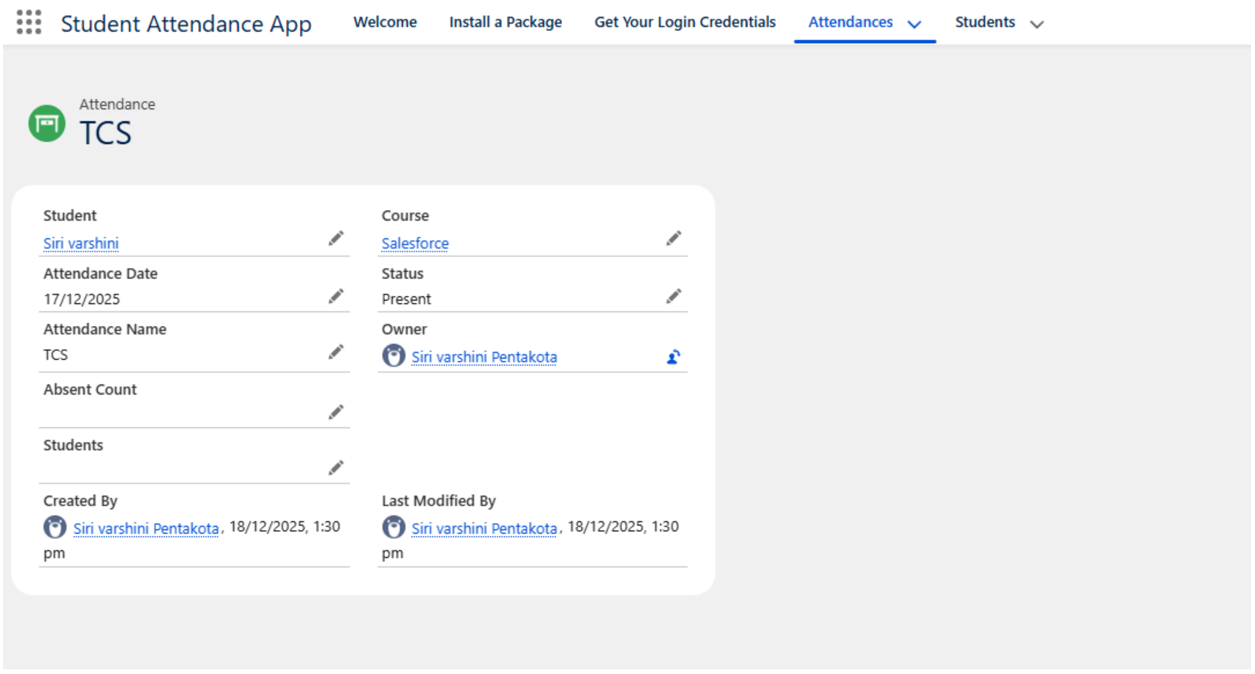
C. Attendance Record Page

Description

This page is designed for marking and reviewing attendance. It shows attendance date, status, linked student, and enrollment details.

Tabs Used

- Attendance Details
- Student
- Enrollment



Attendance
TCS

Student: Siri varshini	Course: Salesforce
Attendance Date: 17/12/2025	Status: Present
Attendance Name: TCS	Owner: Siri varshini Pentakota
Absent Count:	
Students:	
Created By: Siri varshini Pentakota, 18/12/2025, 1:30 pm	Last Modified By: Siri varshini Pentakota, 18/12/2025, 1:30 pm

---

6.3 Tabs

Use Case

Tabs are used to organize content into logical sections. This improves readability and user experience by avoiding cluttered pages.

Implementation

Tabs were added to:

- Separate Student Details, Attendance, Enrollments, and Results
- Improve navigation speed

6.4 Home Page Layouts

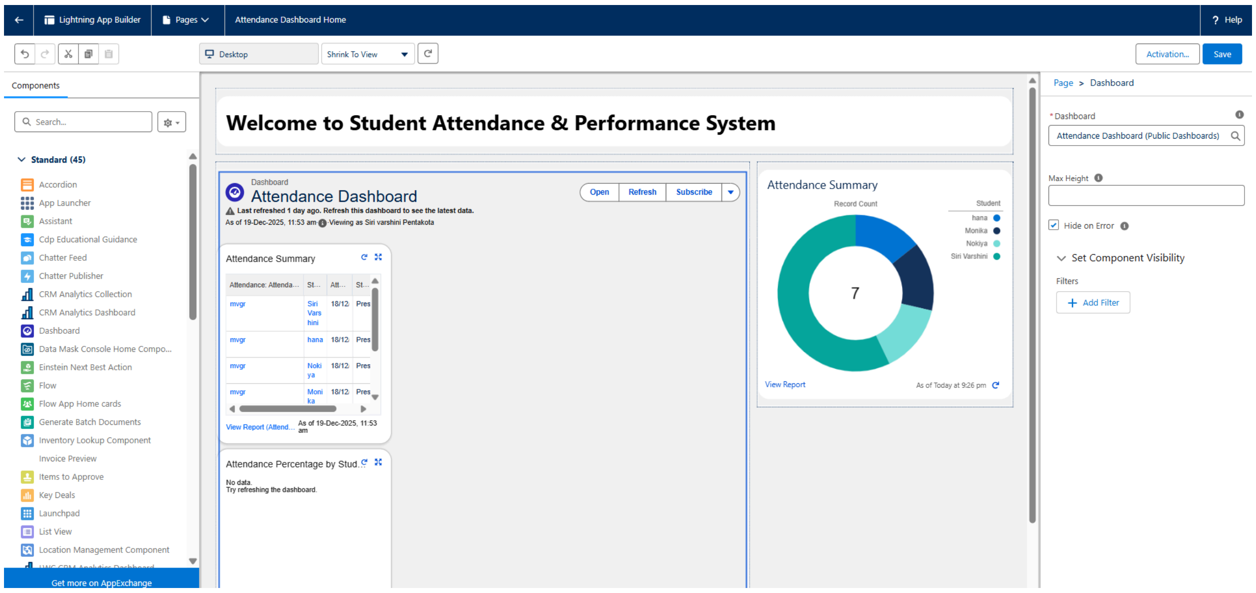
Use Case

The Home Page provides a dashboard view for users when they log in. It displays key reports and dashboards related to attendance and enrollment.

Implementation

A custom Home Page named **“Attendance Dashboard Home”** was created using:

- Dashboard Component
- Report Chart
- Rich Text Welcome Message



Welcome to Student Attendance & Performance System

Dashboard
Attendance Dashboard
Last refreshed 1 day ago. Refresh this dashboard to see the latest data.
As of 19-Dec-2025, 11:53 am · Viewing as Siri varshini Pentakota

Attendance Summary

Attendance: Attenda...	St...	Att...	St...
mvgr	Siri Varshini	18/12	Pres
mvgr	hana	18/12	Pres
mvgr	Nokiya	18/12	Pres
mvgr	Monika	18/12	Pres

View Report (Attend... As of 19-Dec-2025, 11:53 am

Attendance Percentage by Stud...
No data.
Try refreshing the dashboard.

Attendance Summary
As of Today at 9:26 pm

* Dashboard: Attendance Dashboard (Public Dashboards)
Max Height:
[✓] Hide on Error
Set Component Visibility
Filters
Add Filter

6.5 Utility Bar

Use Case

The Utility Bar provides quick access to tools without leaving the current page. It improves productivity for teachers and admins.

Utility Items Added

- Recent Items
- Notes
- Reports

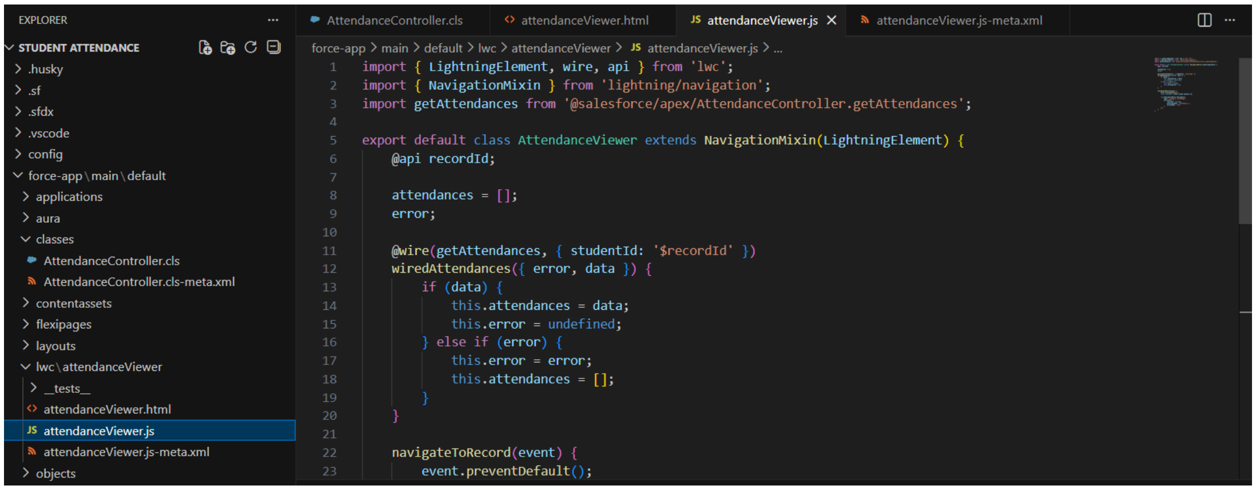
6.6 Lightning Web Components (LWC)

Use Case

LWCs were developed to create dynamic and interactive UI components that cannot be achieved using standard Salesforce components.

Implemented LWC

- **attendanceViewer**: Displays student attendance records dynamicallly



```javascript
import { LightningElement, wire, api } from 'lwc';
import { NavigationMixin } from 'lightning/navigation';
import getAttendances from '@salesforce/apex/AttendanceController.getAttendances';

export default class AttendanceViewer extends NavigationMixin(LightningElement) {
    @api recordId;

    attendances = [];
    error;

    @wire(getAttendances, { studentId: '$recordId' })
    wiredAttendances({ error, data }) {
        if (data) {
            this.attendances = data;
            this.error = undefined;
        } else if (error) {
            this.error = error;
            this.attendances = [];
        }
    }

    navigateToRecord(event) {
        event.preventDefault();
```

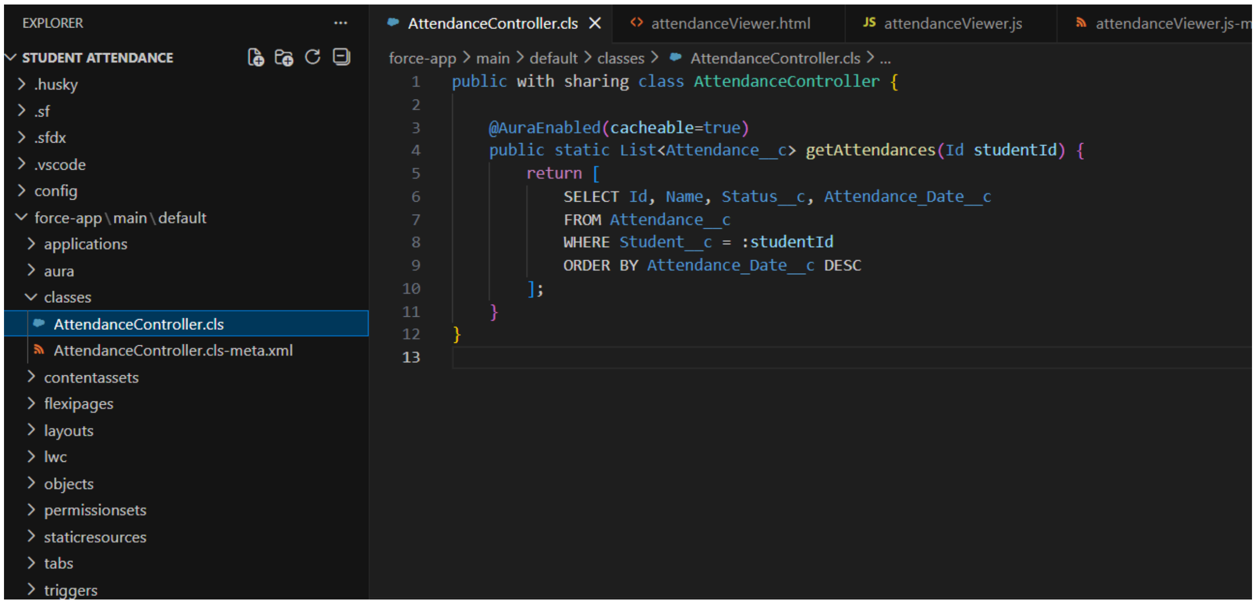
6.7 Apex with LWC

Use Case

Apex is used as a backend controller to fetch and process data for LWCs. This allows secure server-side data access.

Implementation

An Apex class `AttendanceController` was created using `@AuraEnabled` to fetch attendance records.



```
public with sharing class AttendanceController {

    @AuraEnabled(cacheable=true)
    public static List<Attendance__c> getAttendances(Id studentId) {
        return [
            SELECT Id, Name, Status__c, Attendance_Date__c
            FROM Attendance__c
            WHERE Student__c = :studentId
            ORDER BY Attendance_Date__c DESC
        ];
    }
}
```

6.8 Events in LWC

Use Case

Events enable communication between Lightning Web Components. They help update UI dynamically without page reloads.

Implementation

Custom events were used to:

- Refresh attendance data
- Notify parent components of changes

6.9 Wire Adapters

Use Case

Wire adapters allow reactive data fetching. When data changes, the UI updates automatically.

Implementation

The `@wire` decorator was used to fetch attendance data based on Student ID.

```
    @wire(getAttendances, { studentId: '$recordId' })

    wiredAttendances({ error, data }) {

        if (data) {

            this.attendances = data;

            this.error = undefined;

        } else if (error) {

            this.error = error;

            this.attendances = [];

        }

    }
```

6.10 Imperative Apex Calls

Use Case

Imperative Apex calls are used for user-triggered actions like button clicks where immediate backend processing is required.

Implementation

Imperative calls were used to:

- Fetch filtered attendance data
- Perform real-time updates

6.11 Navigation Service

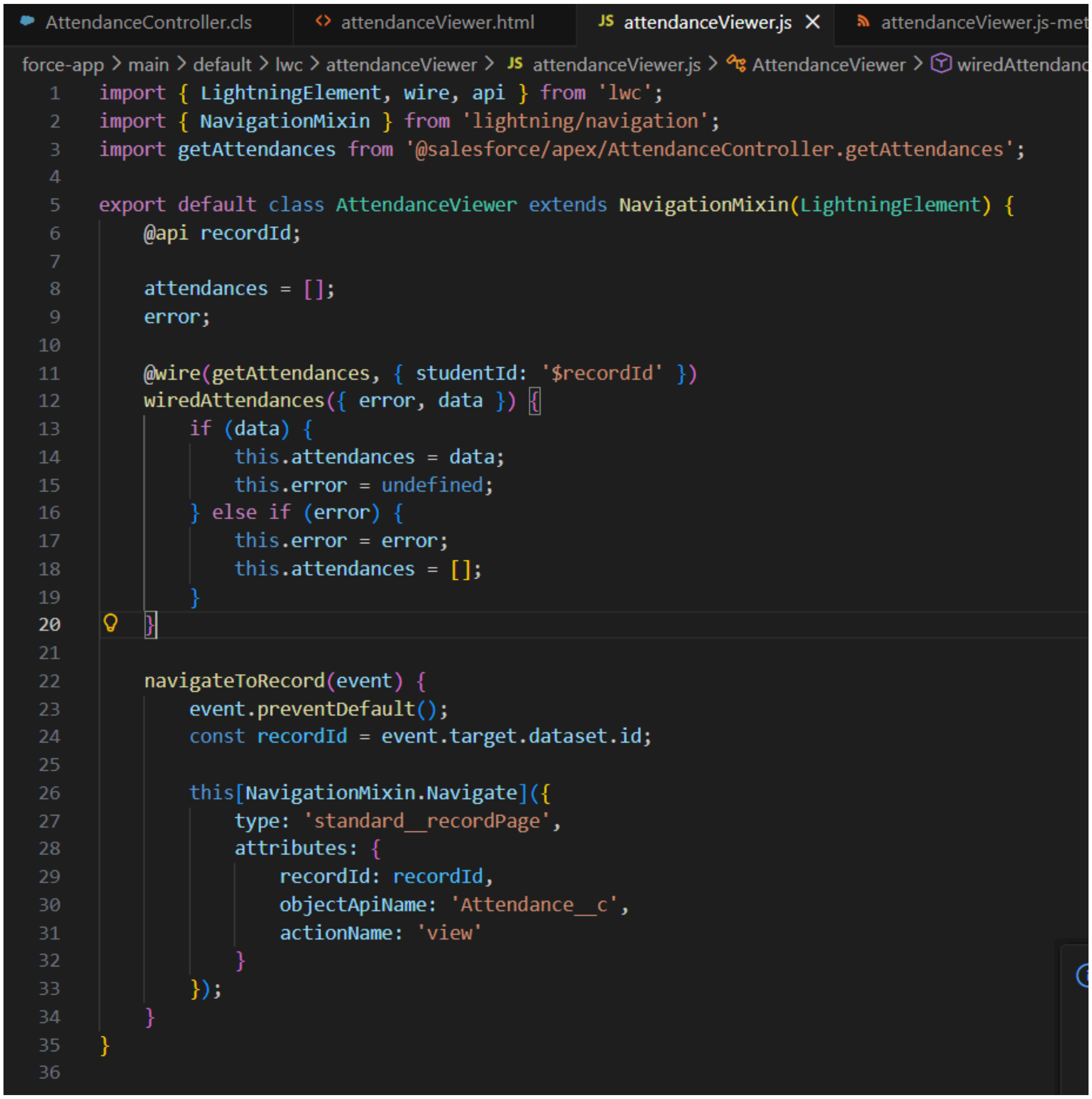
Use Case

Navigation Service improves user experience by allowing programmatic navigation between records and pages.

Implementation

`NavigationMixin` was used to:

- Navigate to Student records
- Open Attendance records dynamically



AttendanceController.cls | attendanceViewer.html | attendanceViewer.js | attendanceViewer.js-met

force-app > main > default > lwc > attendanceViewer > attendanceViewer.js > AttendanceViewer > wiredAttendanc

```javascript
import { LightningElement, wire, api } from 'lwc';
import { NavigationMixin } from 'lightning/navigation';
import getAttendances from '@salesforce/apex/AttendanceController.getAttendances';

export default class AttendanceViewer extends NavigationMixin(LightningElement) {
    @api recordId;

    attendances = [];
    error;

    @wire(getAttendances, { studentId: '$recordId' })
    wiredAttendances({ error, data }) {
        if (data) {
            this.attendances = data;
            this.error = undefined;
        } else if (error) {
            this.error = error;
            this.attendances = [];
        }
    }

    navigateToRecord(event) {
        event.preventDefault();
        const recordId = event.target.dataset.id;

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

✅ Phase 6 Outcome

- Fully customized Lightning UI
- Improved usability and navigation
- Dynamic components using LWC + Apex
- Clean, organized, and responsive user interface
- Meets Salesforce project documentation standards