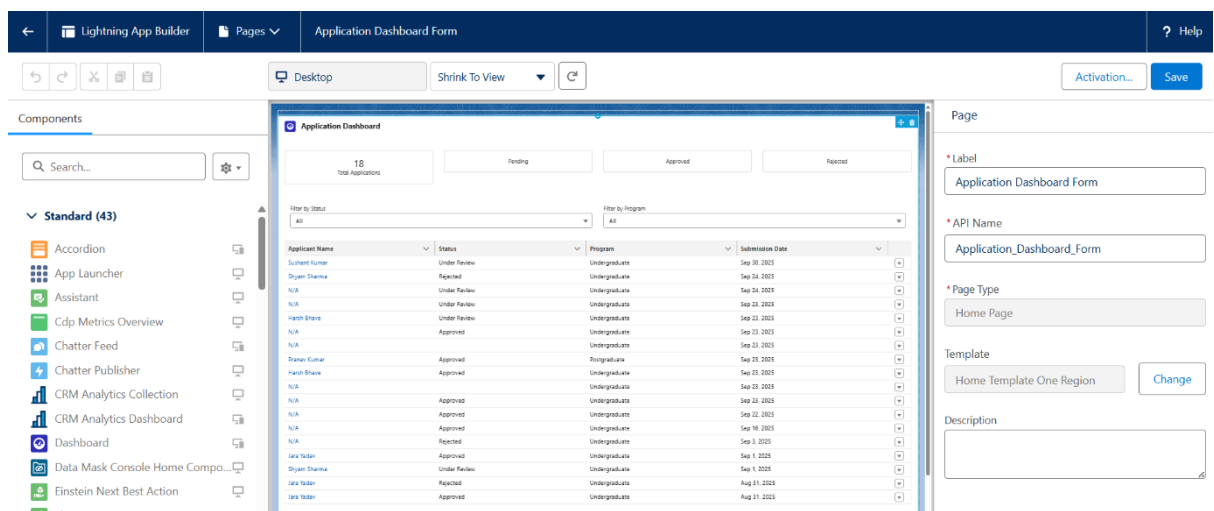


Phase 6: User Interface Development

Goal: Make the Student Success Hub app user-friendly by customising Lightning pages, adding navigation options, and building interactive LWCs integrated with Apex.

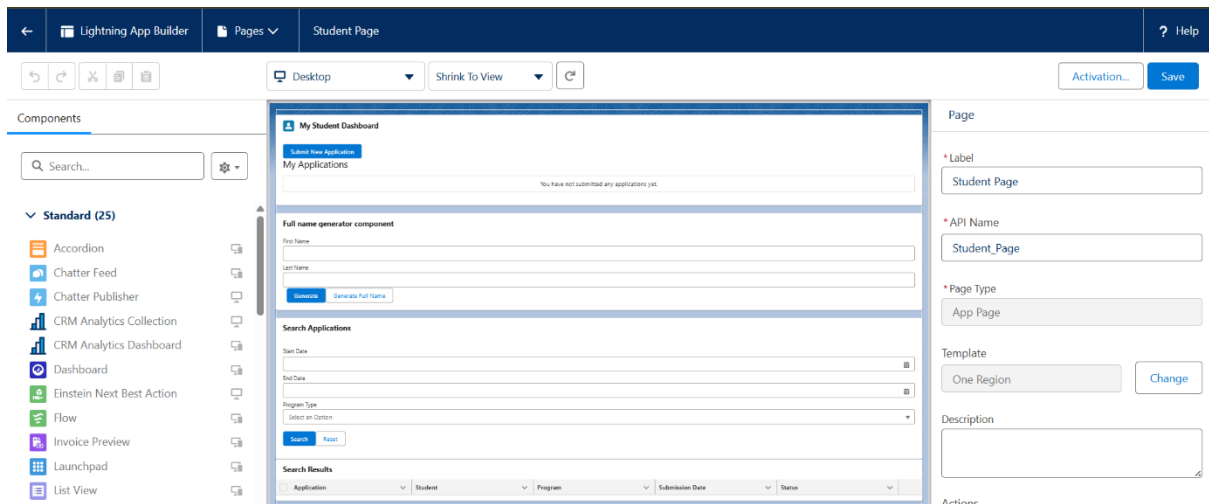
1. Lightning App Builder

- The Student Success Hub App was customised using Lightning App Builder.
- Navigation items were added: Students, Applications, Mentorship Sessions, Reports, and Dashboards.
- The app provides one central workspace for staff to manage students, applications, and mentorship activities.



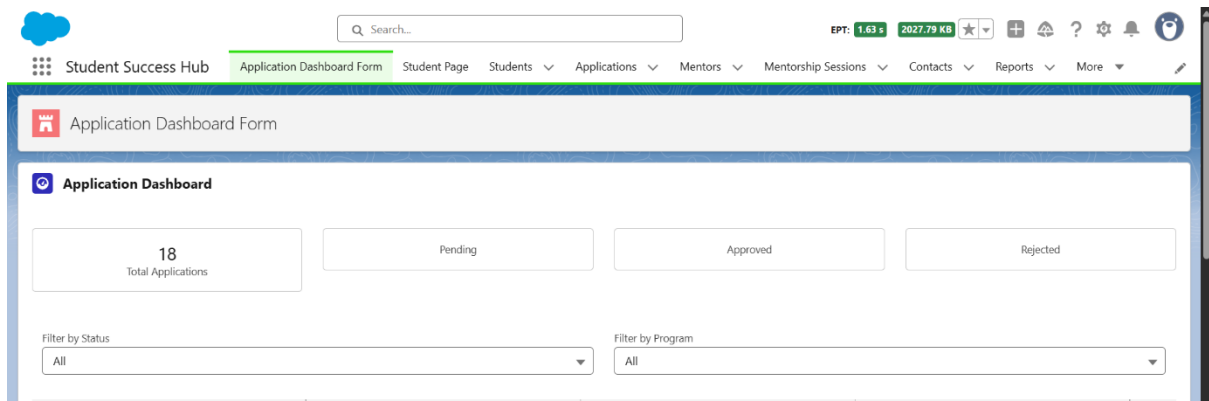
2. Record Pages

- Student Record Page → Displays student details (name, contact info) with related lists of Applications and Mentorship Sessions.
- Application Record Page → Shows program type, submission date, and application status with a Path Component for stage tracking (Submitted → Under Review → Approved/Rejected).
- Mentorship Session Record Page → Displays mentor, student, date, and session notes with related sessions for continuity.



3. Tabs

- Students Tab → Quick access to all students.
- Applications Tab → View and filter applications by status.
- Mentorship Sessions Tab → View upcoming and past mentorship meetings.



4. Home Page Layouts

- A custom Home Page was created for coordinators and mentors.
- Key elements added:
 - Dashboard → Displays application distribution by status.
 - Recent Records → Provides quick access to recently used Students and Applications.
 - Assistant → Reminders for upcoming mentorships and pending reviews.

Student Success Hub Application Dashboard Form Home Page Student Page Students Applications Mentors Mentorship Sessions Contacts More

Home Page

Search Applications

Start Date

End Date

Program Type

Select an Option

Search Reset

Quick New Application

Student Id

Search Applications

Start Date

Full name generator component

First Name

5. Utility Bar

- A Quick Action: A new application has been added to the utility bar.
- Allows staff to create a new application without leaving their current page.

Lightning App Builder App Settings Pages Student Success Hub ? Help

Back Settings

App Details & Branding

App Options

Utility Items (Desktop Only)

Navigation Items

User Profiles

Utility Items (Desktop Only)

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

Add Utility Item

Utility Bar Alignment Default

newApplicationUtility

PROPERTIES newApplicationUtility

Utility Item Properties

* Label newApplicationUtility

Icon fallback

Panel Width 340

6. Lightning Web Components (LWC)

- Search Applications by Student component developed.
- Input: Student name or program type.
- Output: Applications displayed in a datatable.
- The LWC ensures faster searching and better visibility of student applications.

Search Applications

Start Date

End Date

Program Type

Select an Option

Search

Reset

7. Apex with LWC

- Apex methods were written in ApplicationController.cls for data access.
- Features:
- Fetch applications for a given student/program.
- Create new applications.
- Create mentorship sessions linked to applications.


```

1 public with sharing class ApplicationController {
2
3     @AuraEnabled(cacheable=true)
4     public static List<Application__c> getApplications(String startDateStr, String endDateStr, String programType) {
5         Date startDate = String.isBlank(startDateStr) ? null : Date.valueOf(startDateStr);
6         Date endDate = String.isBlank(endDateStr) ? null : Date.valueOf(endDateStr);
7
8         String query = 'SELECT Id, Name, Status__c, RecordTypeId, Student__r.Name, Submission_Date__c FROM Application__c WHERE IsDeleted = false';
9         if (startDate != null) {
10             query += ' AND Start_Date__c >= :startDate';
11         }
12         if (endDate != null) {
13             query += ' AND End_Date__c <= :endDate';
14         }
15         if (!String.isBlank(programType)) {
16             query += ' AND RecordTypeId = :programType';
17         }
18         query += ' ORDER BY Submission_Date__c DESC LIMIT 500';
19     }
20 }

```

8. Events in LWC

- Implemented event handling between components:
 - Child Component (Search Form) → fires an event with input criteria.
 - Parent Component (Results Table) → listens to the event and updates the data shown.


Quick New Application

*Student Name

Program Type

Submission Date

9. Wire Adapters

- Used `@wire` to automatically fetch data from Apex for the applications list.
- Ensures that data updates when underlying records change.


```

JS mentorDashboardForm.js X
Student-Success-Hub > force-app > main > default > lwc > mentorDashboardForm > JS mentorDashboardForm.js > ...
31 export default class MentorDashboard extends LightningElement {
34   @track sessions;
    0 references
35   menteeColumns : ({ label: string; fieldName: string; type... = MENTEE_COLUMNS;
    0 references
36   sessionColumns : ({ label: string; fieldName: string; type... = SESSION_COLUMNS;
37
38   @wire(getRecord, { recordId: '$recordId', fields: MENTOR_FIELDS })
    0 references
39   mentor;
40
41   @wire(getMenteesForMentor, { mentorId: '$recordId' })
    0 references
42   wiredMentees({ error, data } : { error: any; data: any }) : void {
43     if (data) {
44       // Flatten the data for the datatable
45       this.mentees = data.map(item : any => ({
46         ...item,
47         studentName: item.Student__r.Name,
48         studentEmail: item.Student__r.Email, // ✓ Fixed
49         studentPhone: item.Student__r.Phone, // ✓ Fixed
50         studentUrl: `/${item.Student__c}`
51       }));
52     }
53   }
54 }

```

10. Imperative Apex Calls

- Used for actions that require user input:
- On clicking “Submit Application” → Apex method called to insert new Application__c record.
- On clicking “Create Mentorship Session” → Apex method called to insert new Mentorship_Session__c record.

 **Quick New Application**

* Student Name

Program Type

Submission Date

11. Navigation Service

- After creating an application or mentorship session, the user is redirected to the corresponding record page.
- This ensures seamless workflow without manual navigation.

JS newApplicationUtility.js X



Student-Success-Hub > force-app > main > default > lwc > newApplicationUtility > JS newApplicationUtility.js > ...

```
1 import { LightningElement, track } from 'lwc';
2 import createApplication from '@salesforce/apex/ApplicationController.createApplication';
3 import { NavigationMixin } from 'lightning/navigation';
4
5 export default class NewApplicationUtility extends NavigationMixin(LightningElement) {
6     studentId : string = '';
7     programType : string = 'Undergraduate';
8     submissionDate : string = new Date().toISOString().split('T')[0];
9     @track message;
10
11     get programOptions() : { label: string; value: string }[] {
12         return [
13             { label: 'Undergraduate', value: 'Undergraduate' },
14             { label: 'Postgraduate', value: 'Postgraduate' }
15         ];
16     }
17 }
```