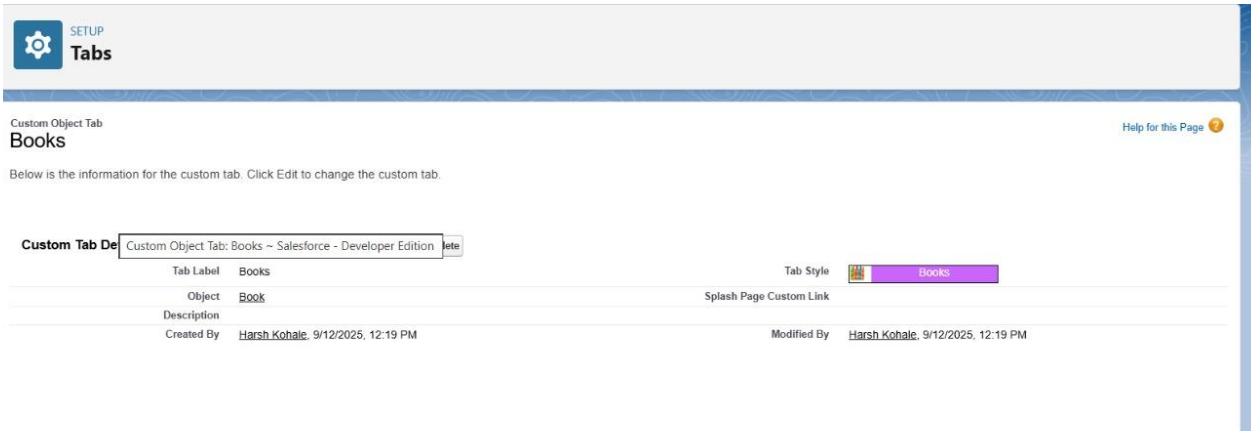
Phase 6: Lightning App Builder & Custom Development

This phase involves the creation and configuration of the user interface using Salesforce Lightning App Builder and custom Lightning Web Components (LWC) to manage rental processes.

Custom Object Tabs

Custom tabs were created for key objects to make them accessible within the Salesforce user interface. Each tab is associated with a specific custom object and has a unique style.

- **Books Tab**: A custom object tab with the label "Books" was created for the "Book" object[1]. The tab style is represented by an icon of a stack of books.



SETUP

Tabs

Custom Object Tab

Books

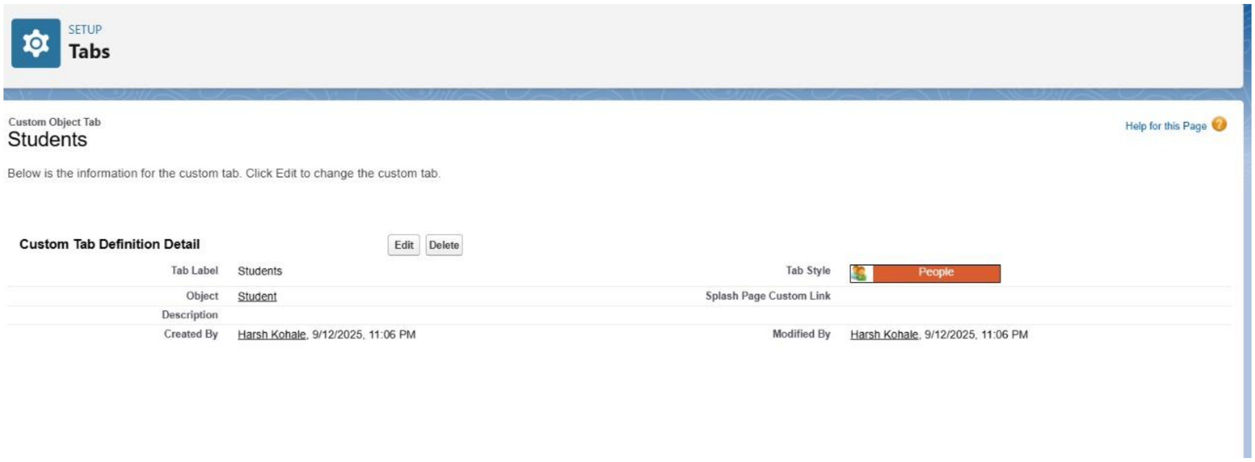
Help for this Page

Below is the information for the custom tab. Click Edit to change the custom tab.

Custom Tab De Custom Object Tab: Books ~ Salesforce - Developer Edition lete

Tab Label	Books	Tab Style	Books
Object	Book	Splash Page Custom Link	
Description			
Created By	Harsh Kohale, 9/12/2025, 12:19 PM	Modified By	Harsh Kohale, 9/12/2025, 12:19 PM

- **Students Tab**: A custom object tab with the label "Students" was created for the "Student" object[3]. The tab style uses an icon representing "People.



SETUP

Tabs

Custom Object Tab

Students

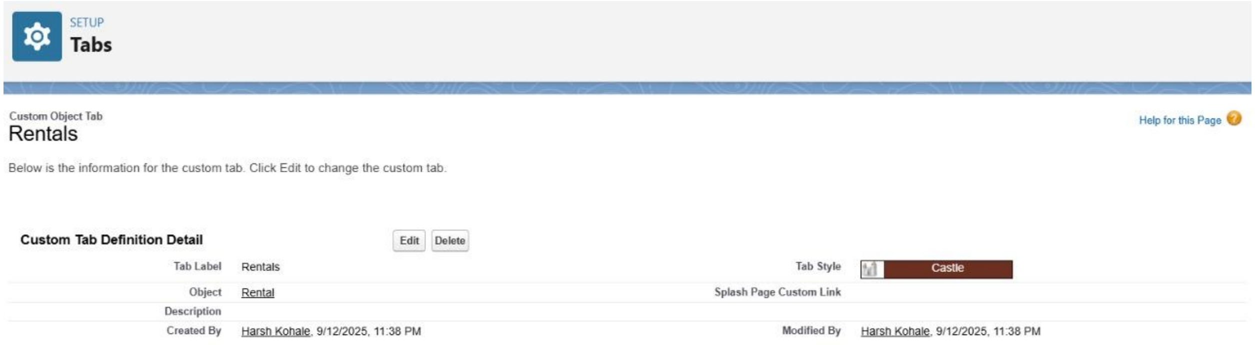
Help for this Page

Below is the information for the custom tab. Click Edit to change the custom tab.

Custom Tab Definition Detail Edit Delete

Tab Label	Students	Tab Style	People
Object	Student	Splash Page Custom Link	
Description			
Created By	Harsh Kohale, 9/12/2025, 11:06 PM	Modified By	Harsh Kohale, 9/12/2025, 11:06 PM

- **Rentals Tab**: A custom object tab with the label "Rentals" was created for the "Rental" object[5]. The tab style is represented by a "Castle" icon.



SETUP

Tabs

Custom Object Tab

Rentals

Help for this Page

Below is the information for the custom tab. Click Edit to change the custom tab.

Custom Tab Definition Detail Edit Delete

Tab Label	Rentals	Tab Style	Castle
Object	Rental	Splash Page Custom Link	
Description			
Created By	Harsh Kohale, 9/12/2025, 11:38 PM	Modified By	Harsh Kohale, 9/12/2025, 11:38 PM

Lightning App Builder

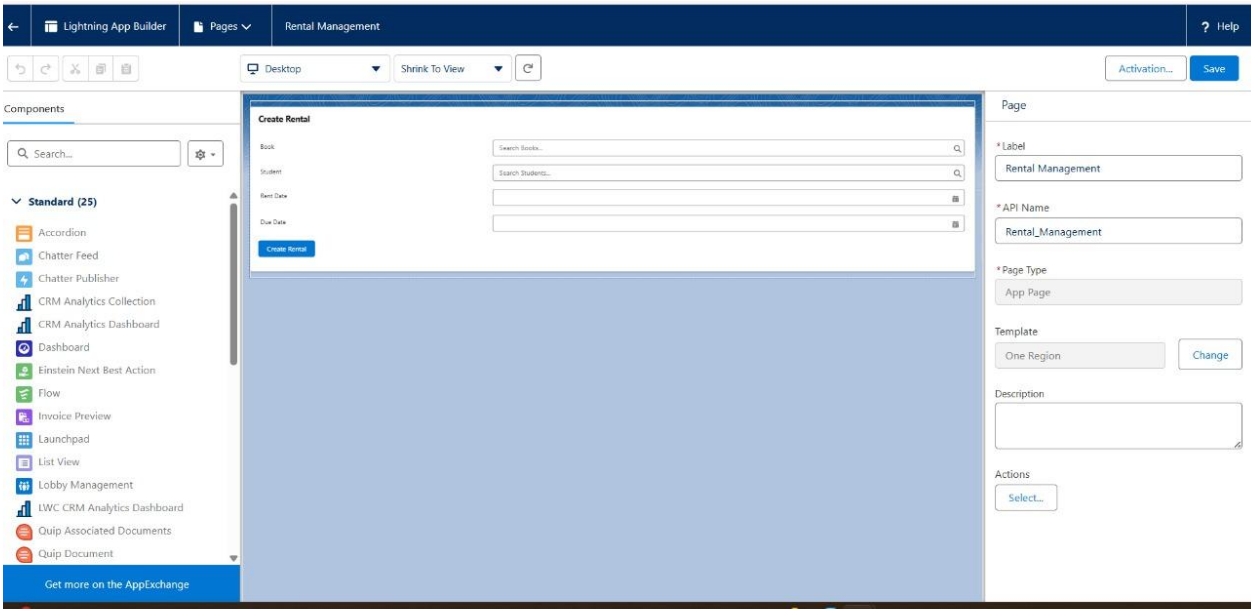
A new app page named **"Rental Management"** was created to serve as the primary interface for managing book rentals[7].

- **Page Label**: Rental Management [8]
- **API Name**: Rental_Management [9]
- **Page Type**: App Page [10]

- **Template**: One Region [11]

The page layout is designed to allow users to create new rental records directly[12]. The "Create Rental" component on this page includes input fields for:

- Book
- Student
- Rent Date
- Due Date



Create Rental

Book: Search Books...

Student: Search Students...

Rent Date:

Due Date:

Page

*Label: Rental Management

*API Name: Rental_Management

*Page Type: App Page

Template: One Region

Description:

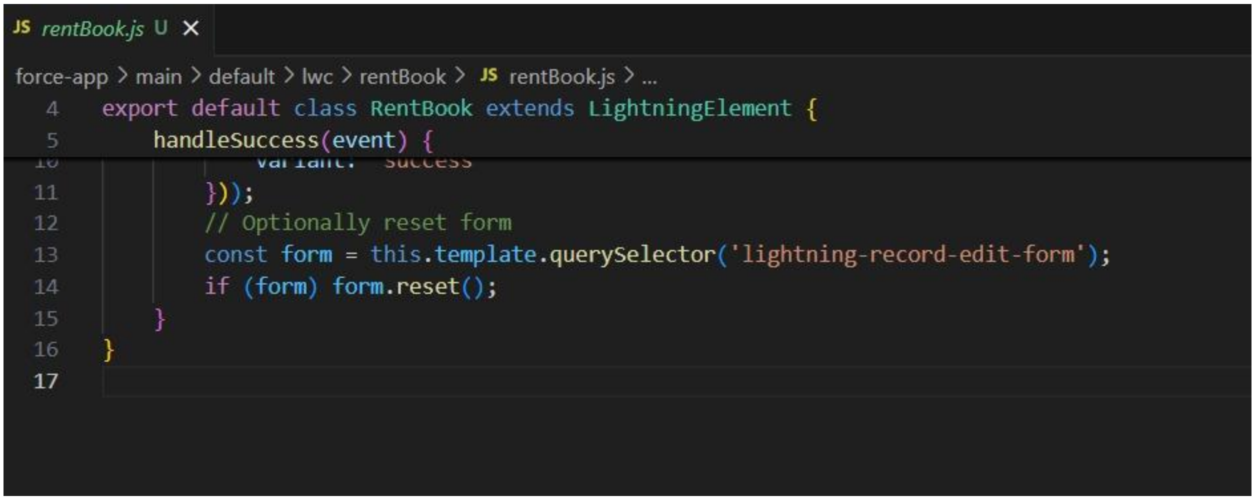
Actions: Select...

Lightning Web Components (LWC)

A custom Lightning Web Component named **rentBook** was developed to handle the rental creation process.

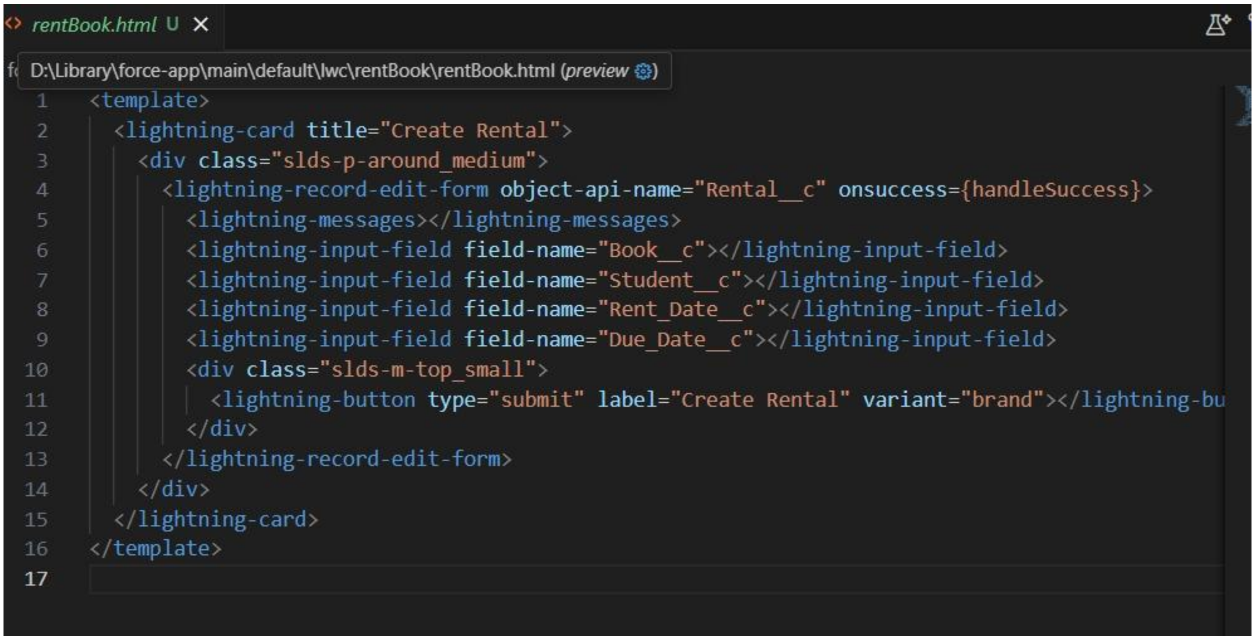
- **Component Structure**: The component consists of a template file (rentBook.html) and a JavaScript file (rentBook.js)[181818].
- **HTML (rentBook.html)**: The template uses lightning-card to display a "Create Renta

form[19]. It includes a lightning-record-edit-form with the object-api-name set to Rental__c, which is the custom object for rentals[20]. The form contains lightning-input-field elements for Book__c, Student__c, Rent_Date__c, and Due_Date__c[21]. A submit button labeled "Create Rental" is also included[22].


rentBook.js U ×
force-app > main > default > lwc > rentBook > JS rentBook.js > ...

```
export default class RentBook extends LightningElement {
    handleSuccess(event) {
                variant: 'success'
            }));
            // Optionally reset form
            const form = this.template.querySelector('lightning-record-edit-form');
            if (form) form.reset();
        }
}
```

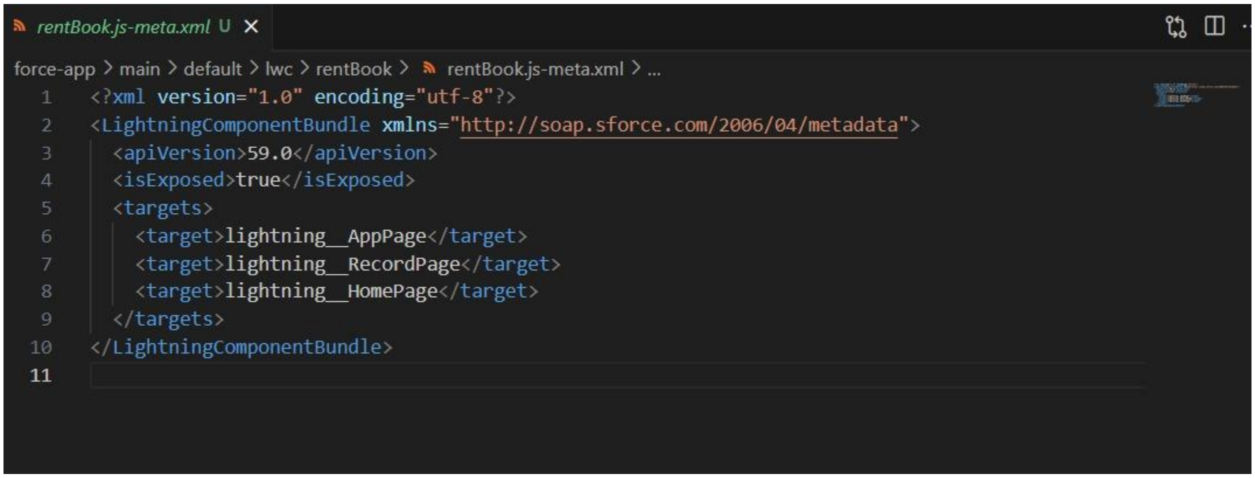
- **JavaScript (rentBook.js)**: The JavaScript controller contains a handleSuccess function[23]. This function is triggered upon successful submission of the form[24]. The function optionally resets the form to allow for a new entry[25].


rentBook.html U ×
D:\Library\force-app\main\default\lwc\rentBook\rentBook.html (preview ⚙)

```
<template>
  <lightning-card title="Create Rental">
    <div class="slds-p-around_medium">
      <lightning-record-edit-form object-api-name="Rental__c" onsuccess={handleSuccess}>
        <lightning-messages></lightning-messages>
        <lightning-input-field field-name="Book__c"></lightning-input-field>
        <lightning-input-field field-name="Student__c"></lightning-input-field>
        <lightning-input-field field-name="Rent_Date__c"></lightning-input-field>
        <lightning-input-field field-name="Due_Date__c"></lightning-input-field>
        <div class="slds-m-top_small">
          <lightning-button type="submit" label="Create Rental" variant="brand"></lightning-bu
        </div>
      </lightning-record-edit-form>
    </div>
  </lightning-card>
</template>
```

- **Component Metadata (rentBook.js-meta.xml)**: The metadata file exposes the component and specifies its targets, allowing it to be used on different types of pages[26].

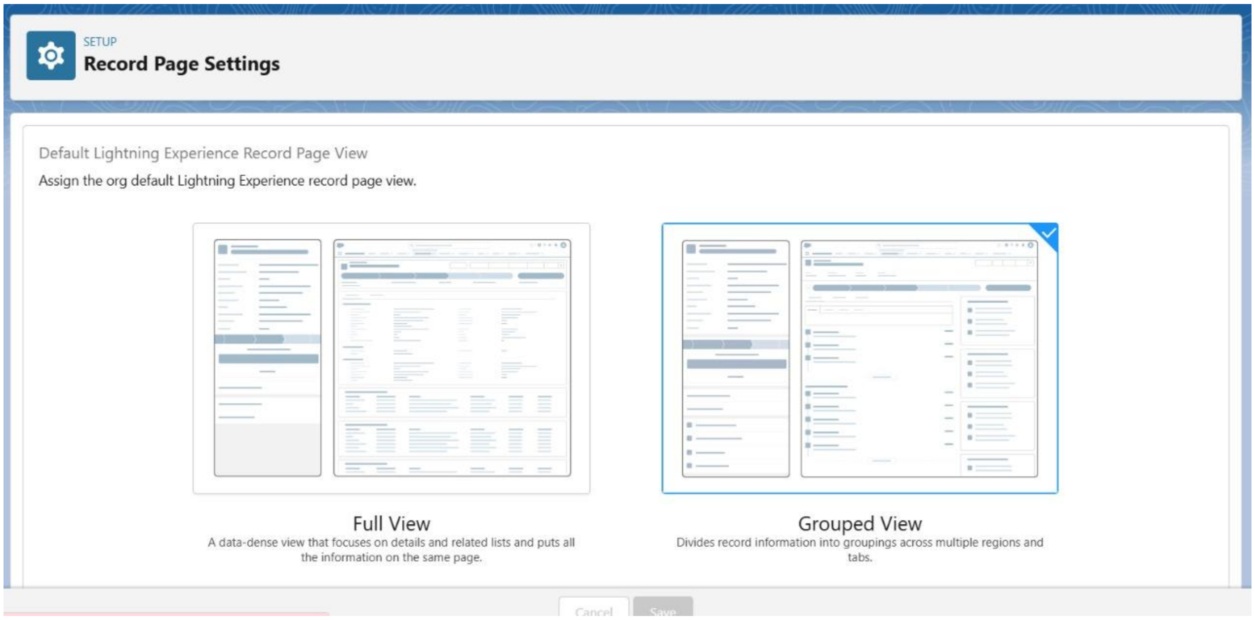
The component is configured to be available on lightning__AppPage, lightning__RecordPage, and lightning__HomePage[27].


rentBook.js-meta.xml U
force-app > main > default > lwc > rentBook > rentBook.js-meta.xml > ...

```xml
<?xml version="1.0" encoding="utf-8"?>
<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
  <apiVersion>59.0</apiVersion>
  <isExposed>true</isExposed>
  <targets>
    <target>lightning__AppPage</target>
    <target>lightning__RecordPage</target>
    <target>lightning__HomePage</target>
  </targets>
</LightningComponentBundle>
```

Record Page Settings

The default Lightning Experience Record Page view was configured[28]. The **"Grouped View"** was selected, which "Divides record information into groupings across multiple regions and tabs"[29]. This view is different from the "Full View," which places all information on a single page[30].



Default Lightning Experience Record Page View

Assign the org default Lightning Experience record page view.

Full View

A data-dense view that focuses on details and related lists and puts all the information on the same page.

Grouped View

Divides record information into groupings across multiple regions and tabs.