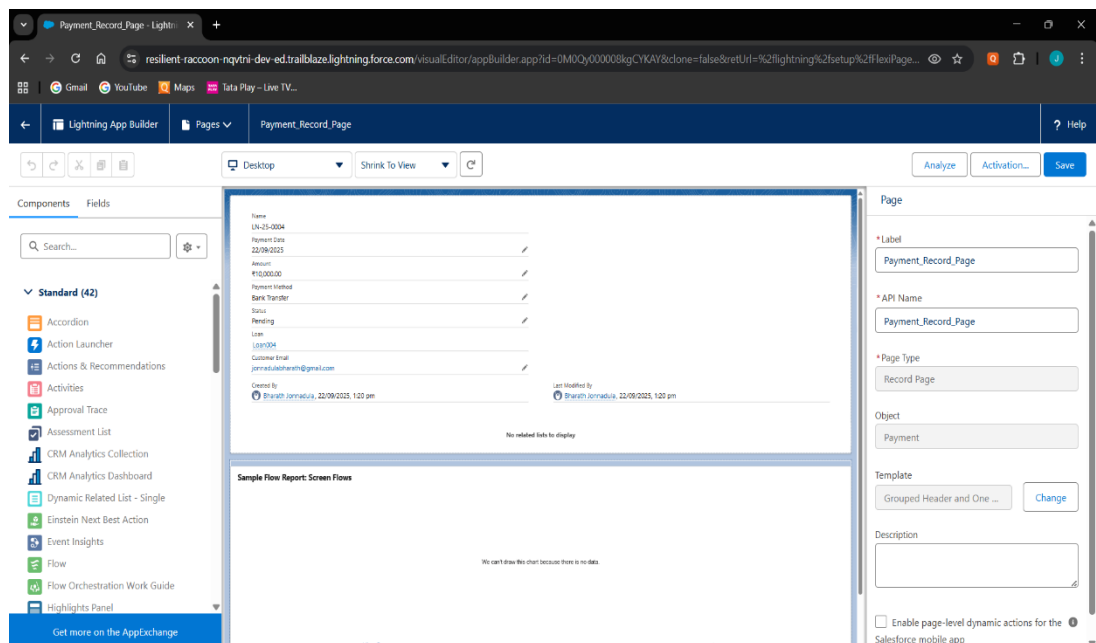
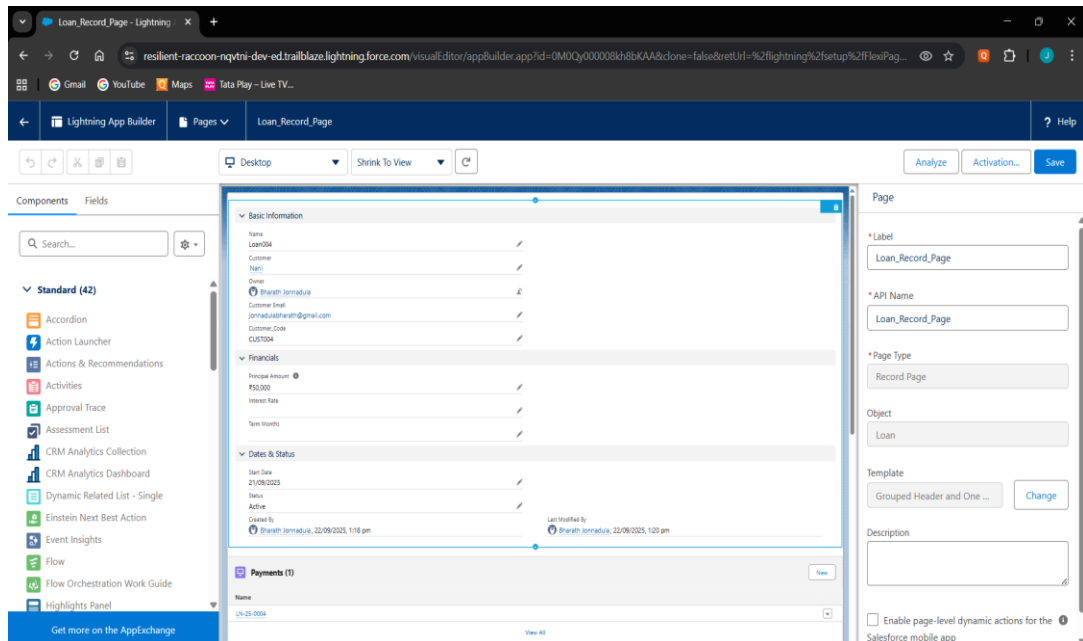


Phase 6: User Interface Development

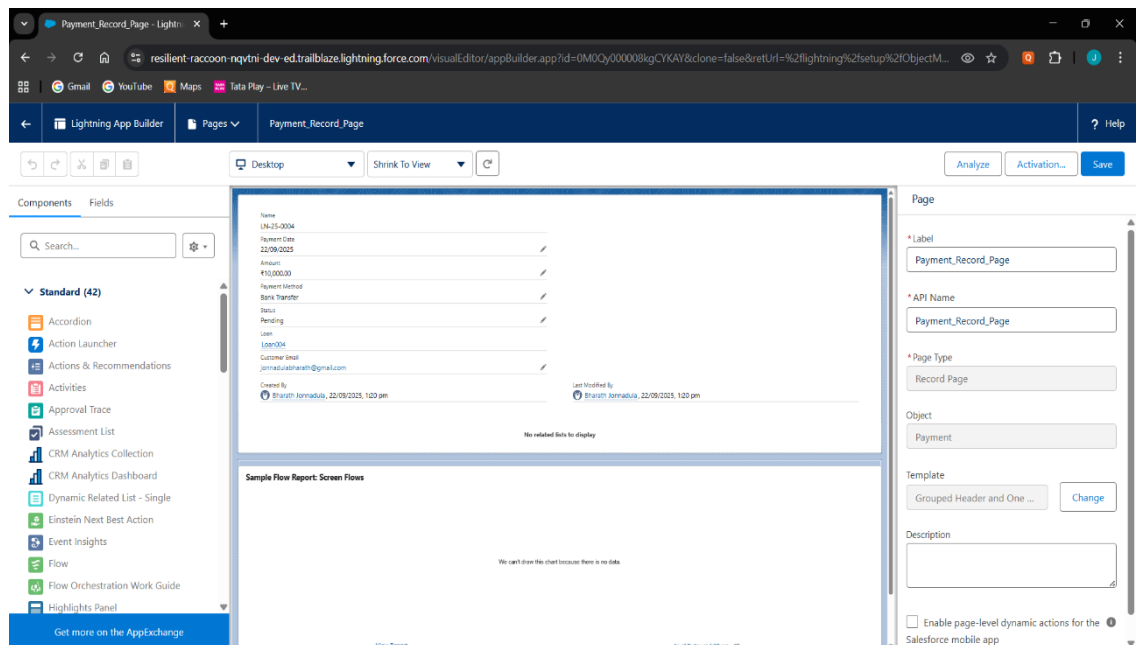
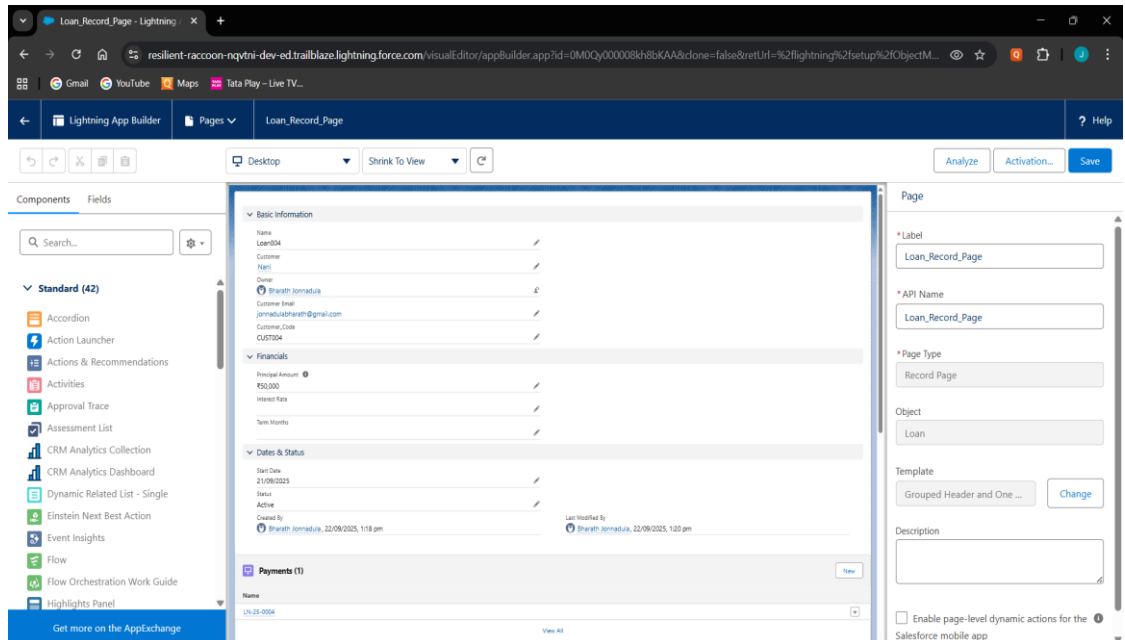
Lightning App Builder

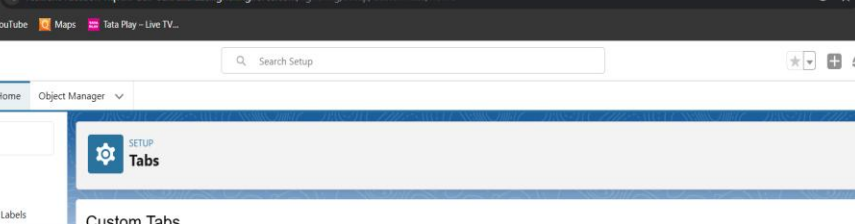
- Created a custom Loan_Record_Page, Payment_Home_Page, Payment_Record_Page, Recovery_record_Page



Record Pages

- Built custom Record Pages for Loan and Payment objects using Lightning App Builder.



- 

resilient-raccoon-nqytn-dev-ed.trailblaze.lightning.force.com/lightning/setup/CustomTabs/home

Setup Home Object Manager

Search Setup

Setup Tabs

User Interface

Rename Tabs and Labels

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	Loans	Credit card	
Edit Del	Payments	Computer	
Edit Del	Recoveries	Cell phone	

Web Tabs [New](#) [What is This?](#)

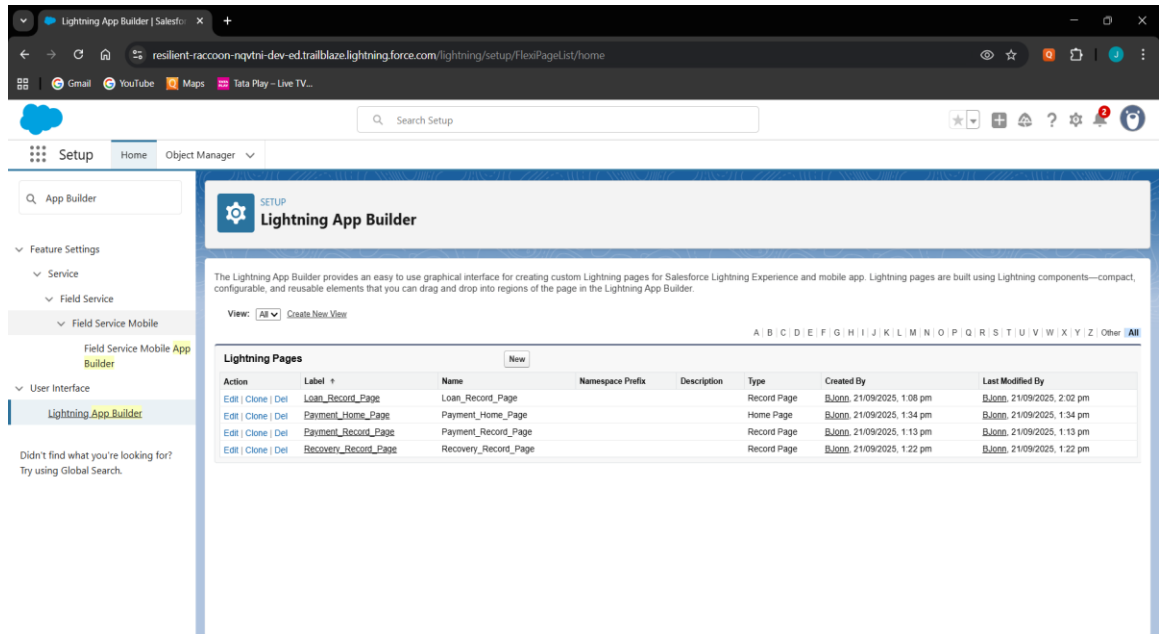
No Web Tabs have been defined

Visualforce Tabs [New](#) [What is This?](#)

No Visualforce Tabs have been defined

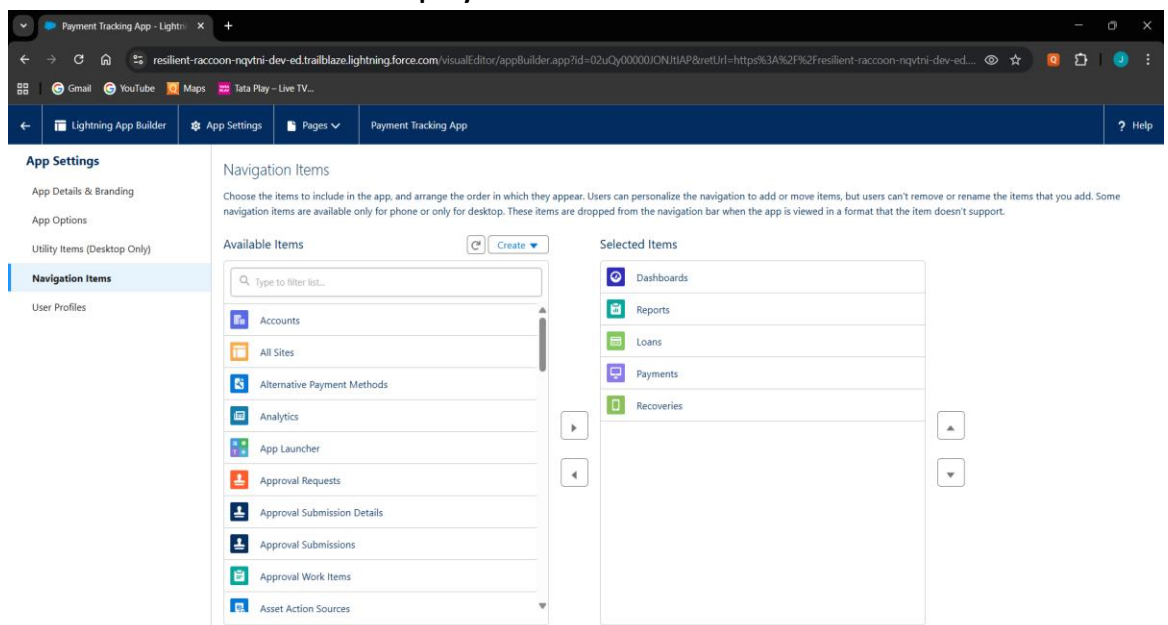
Home Page Layouts

- Customized the Home Page with components like Recent Loans, Pending Payments



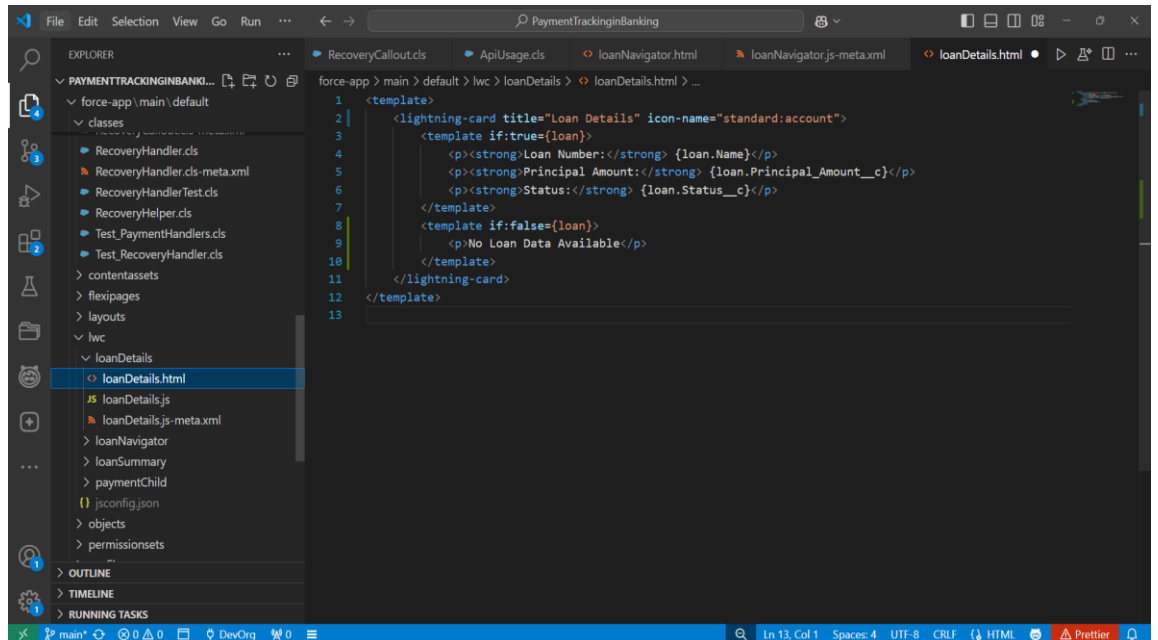
Utility Bar

- Switch between loan & payment records.



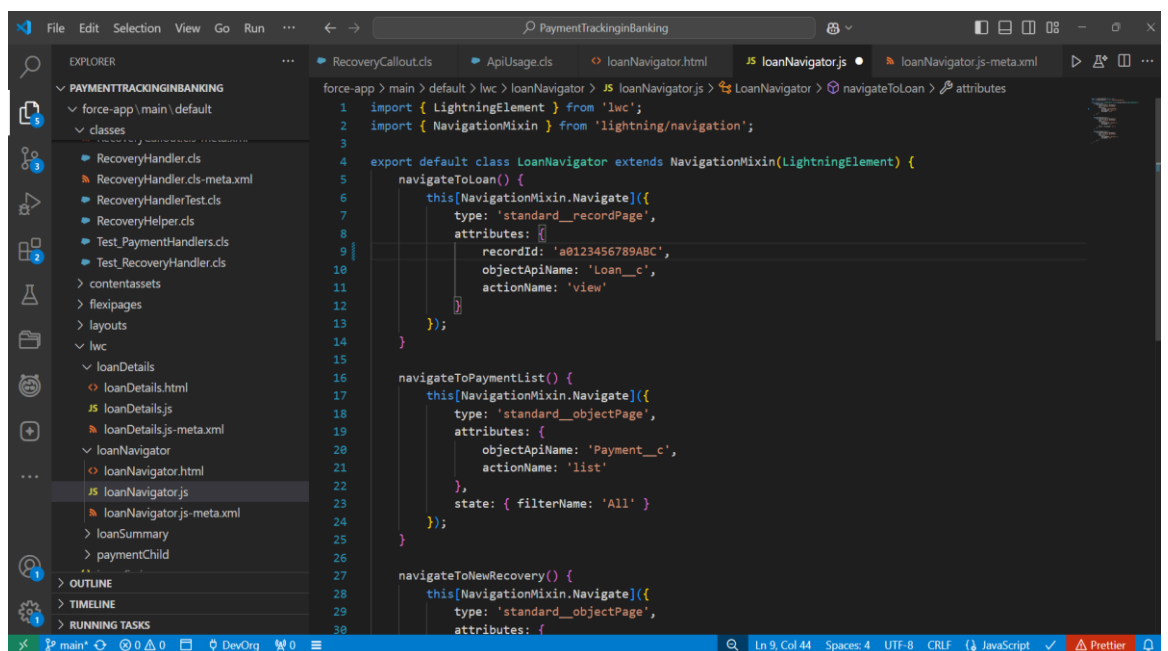
LWC (Lightning Web Components)

- Built LWC components (loanSummary, loanDetails, paymentChild).



The screenshot shows the Visual Studio Code editor with the 'PaymentTrackinginBanking' project open. The Explorer pane on the left shows the file structure, with 'loanDetails.html' selected under the 'lwc' directory. The main editor displays the HTML content of 'loanDetails.html'.

```
1 <template>
2 |
3 |   <lightning-card title="Loan Details" icon-name="standard:account">
4 |     <template if:true={loan}>
5 |       <p><strong>Loan Number:</strong> {loan.Name}</p>
6 |       <p><strong>Principal Amount:</strong> {loan.Principal_Amount__c}</p>
7 |       <p><strong>Status:</strong> {loan.Status__c}</p>
8 |     </template>
9 |     <template if:false={loan}>
10 |      <p>No Loan Data Available</p>
11 |    </template>
12 |  </lightning-card>
13 </template>
```

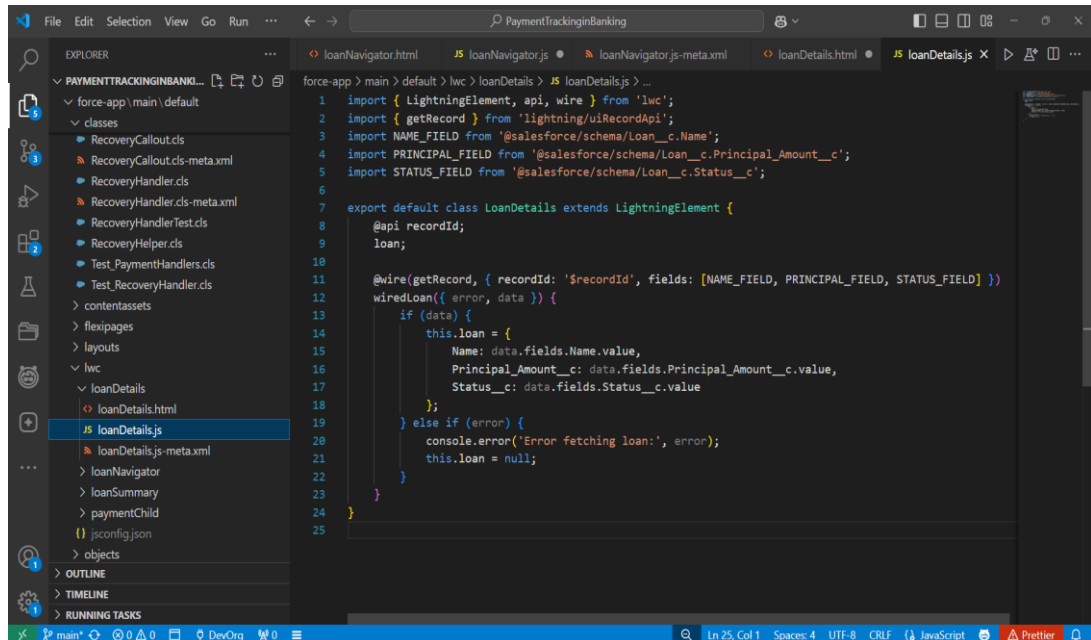


The screenshot shows the Visual Studio Code editor with the 'PaymentTrackinginBanking' project open. The Explorer pane on the left shows the file structure, with 'loanNavigator.js' selected under the 'lwc' directory. The main editor displays the JavaScript code for 'loanNavigator.js'.

```
1 import { LightningElement } from 'lwc';
2 import { NavigationMixin } from 'lightning/navigation';
3
4 export default class LoanNavigator extends NavigationMixin(LightningElement) {
5   navigateToLoan() {
6     this[NavigationMixin.Navigate]({
7       type: 'standard__recordPage',
8       attributes: {
9         recordId: 'a0123456789ABC',
10        objectApiName: 'Loan__c',
11        actionName: 'view'
12      }
13    });
14  }
15
16  navigateToPaymentList() {
17    this[NavigationMixin.Navigate]({
18      type: 'standard__objectPage',
19      attributes: {
20        objectApiName: 'Payment__c',
21        actionName: 'list'
22      },
23      state: { filterName: 'All' }
24    });
25  }
26
27  navigateToNewRecovery() {
28    this[NavigationMixin.Navigate]({
29      type: 'standard__objectPage',
30      attributes: {
```

Wire Adapters

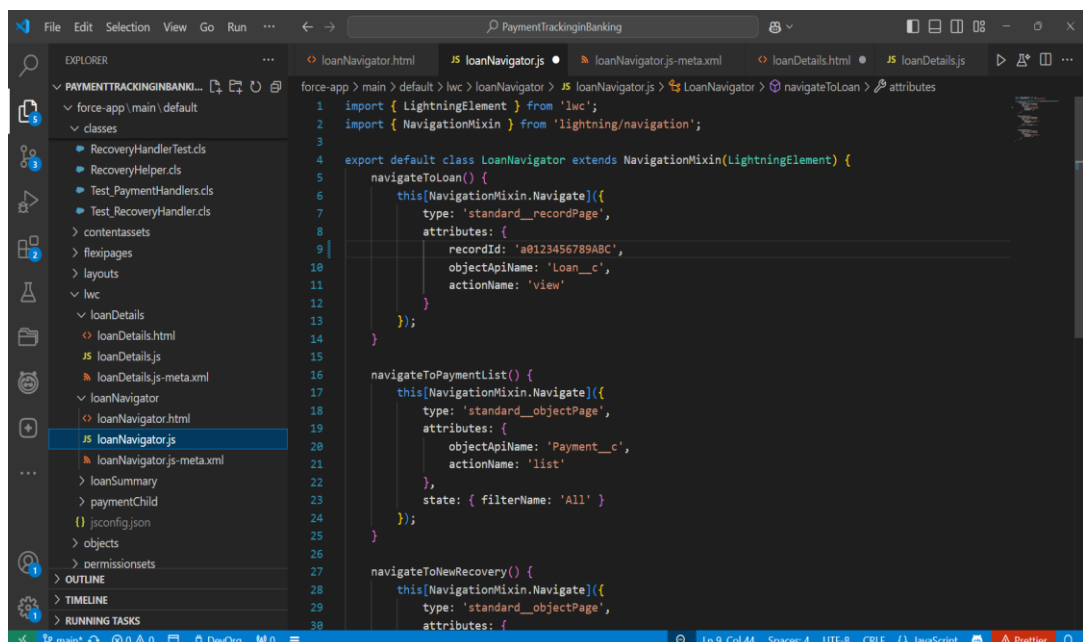
- Used @wire(getRecord) to fetch Loan fields directly in LWC.



```
force-app > main > default > lwc > loanDetails > JS loanDetails.js > ...
1 import { LightningElement, api, wire } from 'lwc';
2 import { getRecord } from 'lightning/uiRecordApi';
3 import NAME_FIELD from '@salesforce/schema/Loan__c.Name';
4 import PRINCIPAL_FIELD from '@salesforce/schema/Loan__c.Principal_Amount__c';
5 import STATUS_FIELD from '@salesforce/schema/Loan__c.Status__c';
6
7 export default class LoanDetails extends LightningElement {
8   @api recordId;
9   loan;
10
11   @wire(getRecord, { recordId: '$recordId', fields: [NAME_FIELD, PRINCIPAL_FIELD, STATUS_FIELD] })
12   wiredLoan({ error, data }) {
13     if (data) {
14       this.loan = {
15         Name: data.fields.Name.value,
16         Principal_Amount__c: data.fields.Principal_Amount__c.value,
17         Status__c: data.fields.Status__c.value
18       };
19     } else if (error) {
20       console.error('Error fetching loan:', error);
21       this.loan = null;
22     }
23   }
24 }
25
```

Navigation Service

- Implemented NavigationMixin to let users navigate from Payment → Loan detail page.
- Improves user experience with one-click record redirection.



```
force-app > main > default > lwc > loanNavigator > JS loanNavigator.js > LoanNavigator > navigateToLoan > attributes
1 import { LightningElement } from 'lwc';
2 import { NavigationMixin } from 'lightning/navigation';
3
4 export default class LoanNavigator extends NavigationMixin(LightningElement) {
5   navigateToLoan() {
6     this[NavigationMixin.Navigate]({
7       type: 'standard__recordPage',
8       attributes: {
9         recordId: 'a0123456789ABC',
10         objectApiName: 'Loan__c',
11         actionName: 'view'
12       }
13     });
14   }
15
16   navigateToPaymentList() {
17     this[NavigationMixin.Navigate]({
18       type: 'standard__objectPage',
19       attributes: {
20         objectApiName: 'Payment__c',
21         actionName: 'list'
22       },
23       state: { filterName: 'All' }
24     });
25   }
26
27   navigateToNewRecovery() {
28     this[NavigationMixin.Navigate]({
29       type: 'standard__objectPage',
30       attributes: {
31
32       }
33     });
34   }
35 }
36
```