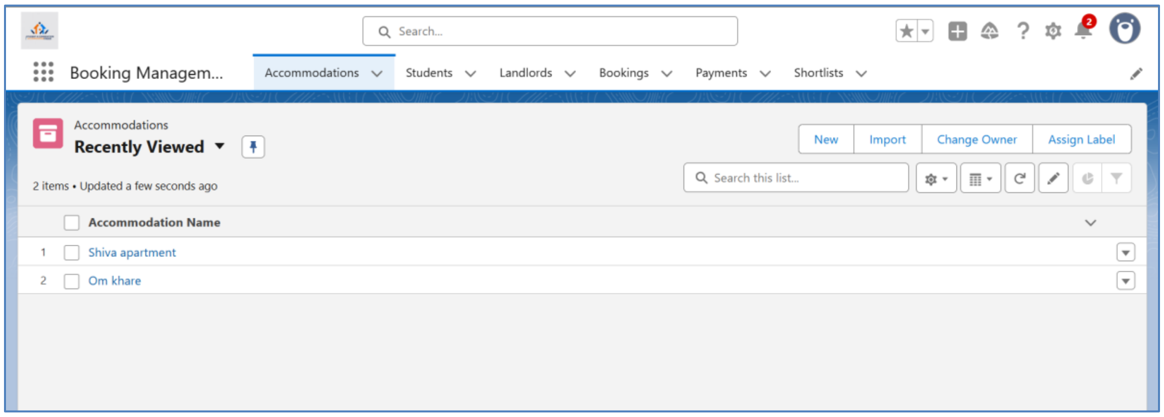
## Phase 6: User Interface Development – Student Accommodation Finder

This document provides a step-by-step implementation guide for UI development in the Student Accommodation Finder project.

1. Lightning App Builder
Goal: Create a custom app for students and landlords.
Steps:
1. Salesforce Setup → Quick Find → App Manager → New Lightning App
2. App Name: Student Accommodation Finder
3. Add Navigation Items: Students, Landlords, Accommodations, Bookings, Payments
4. Finish → App launch



| | Accommodation Name |
|---|---|
| 1 | Shiva apartment |
| 2 | Om khare |

2. Record Pages
Goal: Customize each object page to show relevant details.
Steps:
1. Setup → Object Manager → Select object (e.g., Accommodation)
2. Lightning Record Pages → New
3. Choose Page Template → Standard or Custom
4. Drag components: Record Detail, Related List, LWC components (optional)
5. Activate page

3. Tabs
Goal: Easy navigation.
Steps:
1. App Manager → Edit Student Accommodation Finder
2. Add Tabs: Students | Landlords | Accommodations | Bookings | Payments
3. Save & Assign to users

4. Home Page Layouts
Goal: Student/landlord friendly home page.
Steps:
1. Setup → Lightning App Builder → Home Page → New
2. Drag components:
   - Students → Quick Action: Find Accommodation, My Bookings
   - Landlords → Quick Action: Add New Accommodation, View Payments
3. Activate page

5. Utility Bar
Goal: Quick access tools.
Steps:
1. App Manager → Edit App → Utility Bar
2. Add utilities: Chat Support, Notes, Quick Search Accommodation
3. Save & assign

6. LWC (Lightning Web Components)
Goal: Interactive and responsive components.
Example Components: accommodationSearch, bookingForm, paymentComponent
Steps:
1. VS Code + Salesforce Extensions → Create LWC:

```
   sfdx force:lightning:component:create --type lwc --componentname
accommodationSearch
```

2. Implement HTML, JS, CSS
3. Deploy to Salesforce → Add to Record Page/Home Page

7. Apex with LWC
Goal: Call backend logic from LWC.
Example Apex Controller:

```
public with sharing class AccommodationController {
    @AuraEnabled(cacheable=true)
    public static List<Accommodation__c> getAvailableAccommodations() {
        return [SELECT Id, Name, Location__c, Rent__c FROM Accommodation__c WHERE
Status__c = 'Available'];
    }
}
```

LWC JS:

```
import { LightningElement, wire } from 'lwc';
import getAvailableAccommodations from
'@salesforce/apex/AccommodationController.getAvailableAccommodations';
export default class AccommodationSearch extends LightningElement {
    @wire(getAvailableAccommodations) accommodations;
}
```

## 8. Events in LWC

Goal: Component-to-component communication.
Example:

```
const selectedEvent = new CustomEvent('selected', { detail: this.accommodationId });
this.dispatchEvent(selectedEvent);
```

BookingForm listens to the event to populate data.

## 9. Wire Adapters

Goal: Reactive data fetch.
Example:

```
import { LightningElement, wire } from 'lwc';
import { getRecord } from 'lightning/uiRecordApi';
import NAME_FIELD from '@salesforce/schema/Student__c.Name';
export default class StudentDetail extends LightningElement {
  @wire(getRecord, { recordId: '001XXXXXXXXXXXX', fields: [NAME_FIELD] })
  student;
}
```

## 10. Imperative Apex Calls

Goal: Data fetch/update based on user action.
Example: Book Now button

```
import createBooking from '@salesforce/apex/BookingService.createBooking';
handleBooking() {
  createBooking({ booking: this.bookingData })
    .then(result => { console.log('Booking Successful'); })
    .catch(error => { console.error(error.body.message); });
}
```

## Summary

- Lightning App Builder → App structure & navigation
- Record Pages & Tabs → Object details & easy access
- Home Page & Utility Bar → Quick actions & tools
- LWC + Apex → Dynamic UI & backend logic
- Wire & Imperative Calls → Data fetch on load or user action
- Events → Component communication

These steps create a complete interactive UI for Student Accommodation Finder project where students can search and book accommodations, and landlords can manage listings and payments.