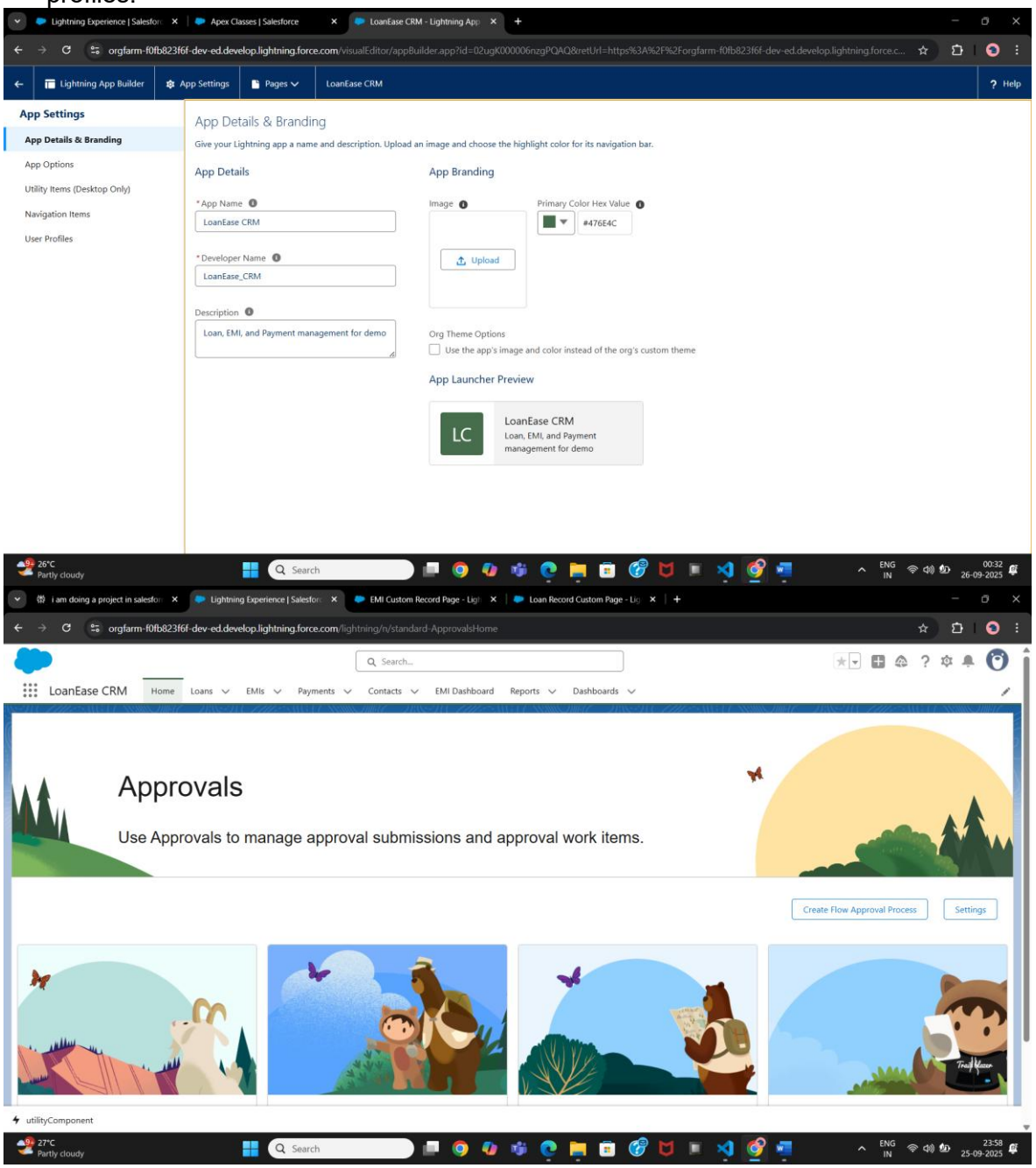
LoanEase: Salesforce Loan Management CRM

Phase 6: User Interface Development

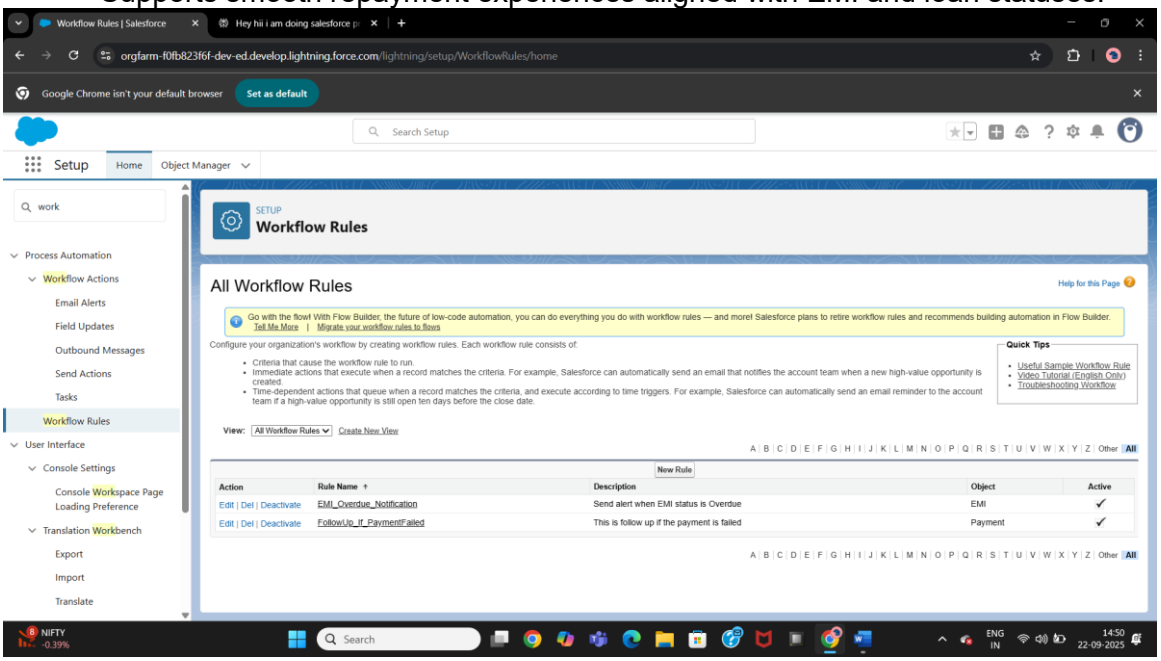
Lightning App Builder

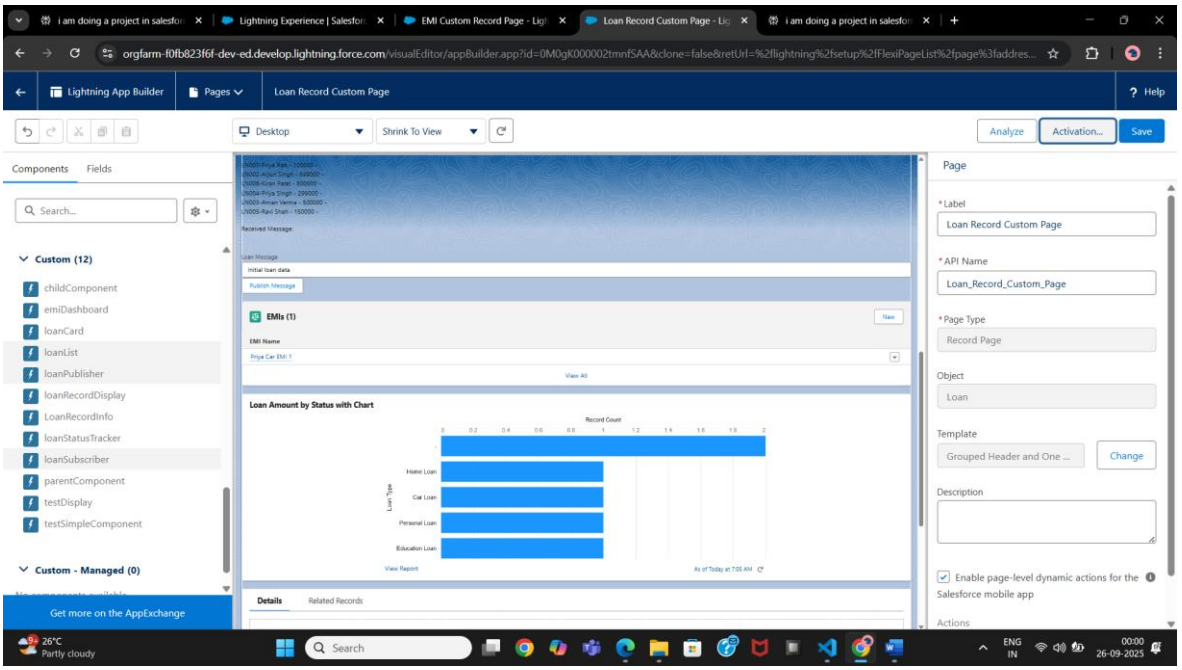
- Added key LWCs like loanRecordDisplay, loanList, emiDashboard, and parentComponent to custom Lightning Record Pages specific to Loan and EMI objects in the org.
- Configured Utility Bar integration with custom LWCs for quick access tools.
- Arranged components on pages to ensure smooth navigation and clear display of loan and EMI information.
- Enabled interaction workflows such as event-driven communication between related components for dynamic UI updates.
- Tested the layout responsiveness and UI behavior consistently in different user profiles.



Record Pages

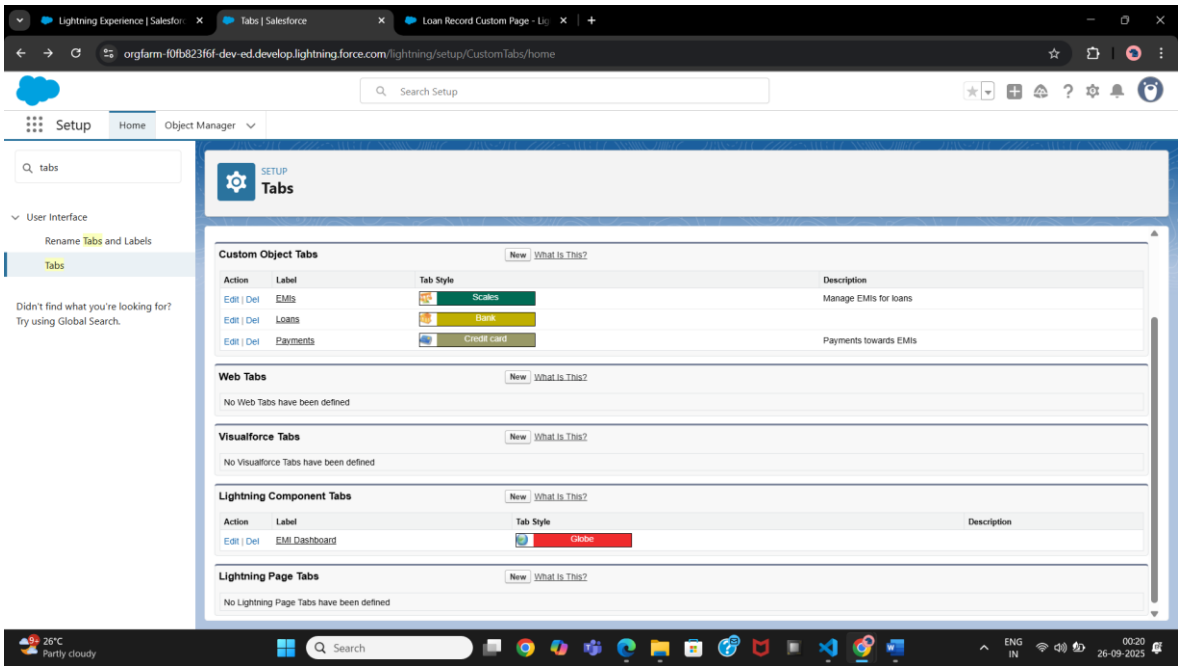
- **EMI Custom Record Page**
  - Focus: Equated Monthly Installments (EMIs) management.
  - Contains components such as emiDashboard for visual summary and tracking of EMIs.
  - Streamlines user workflows specific to EMI tracking and overdue management.
- **Loan Custom Record Page**
  - Focus: Comprehensive loan information management.
  - Includes components like loanRecordDisplay, loanList, and loanCard to provide detailed loan data views and interactions.
  - Enables efficient handling of loan lifecycle data and statuses.
- **Payment Custom Record Page**
  - Focus: Loan payment processes and history.
  - Designed to host payment-related LWCs to facilitate payment monitoring and processing.
  - Supports smooth repayment experiences aligned with EMI and loan statuses.





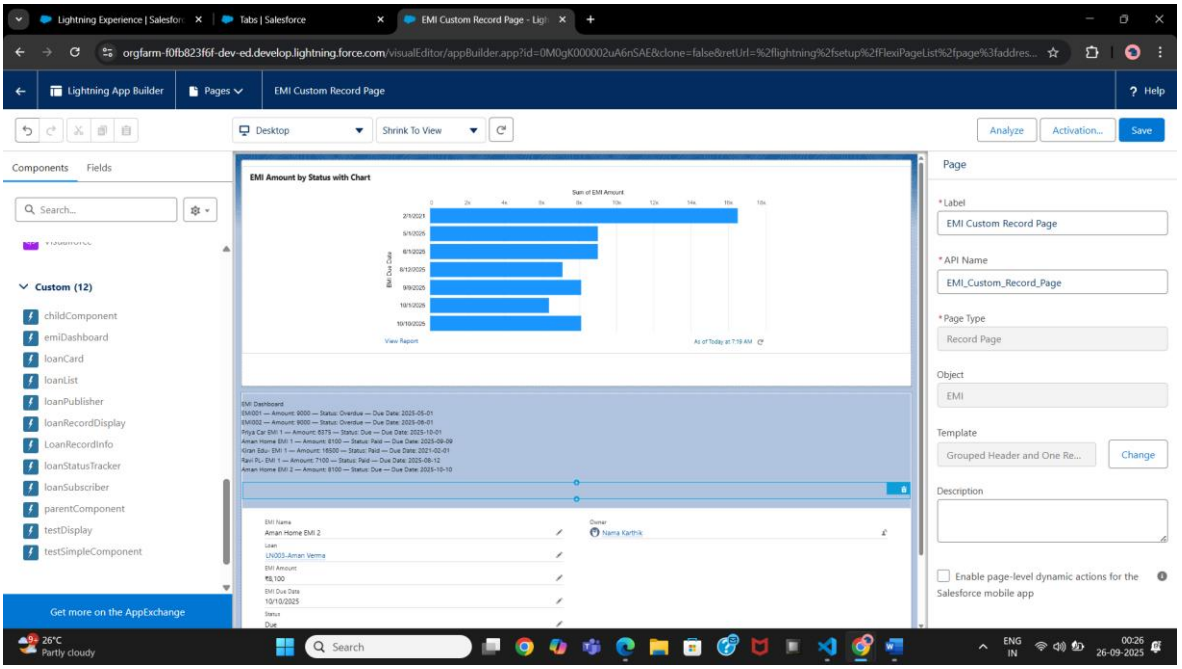
Tabs

- Created distinct Custom Object Tabs for Loans, EMIs, and Payments to categorize and provide easy navigation for major business entities.
- Each tab is styled for clear visual identification and linked to one of the core objects: Loans, EMIs, or Payments.
- Loans Tab: Access and manage all loan records and related processes.
- EMIs Tab: Focuses on managing EMIs for different loans.
- Payments Tab: Displays and processes payments towards EMIs and loan accounts.
- Added a Lightning Component Tab for the EMI Dashboard to provide a quick, visual summary of EMI-related statistics and health.



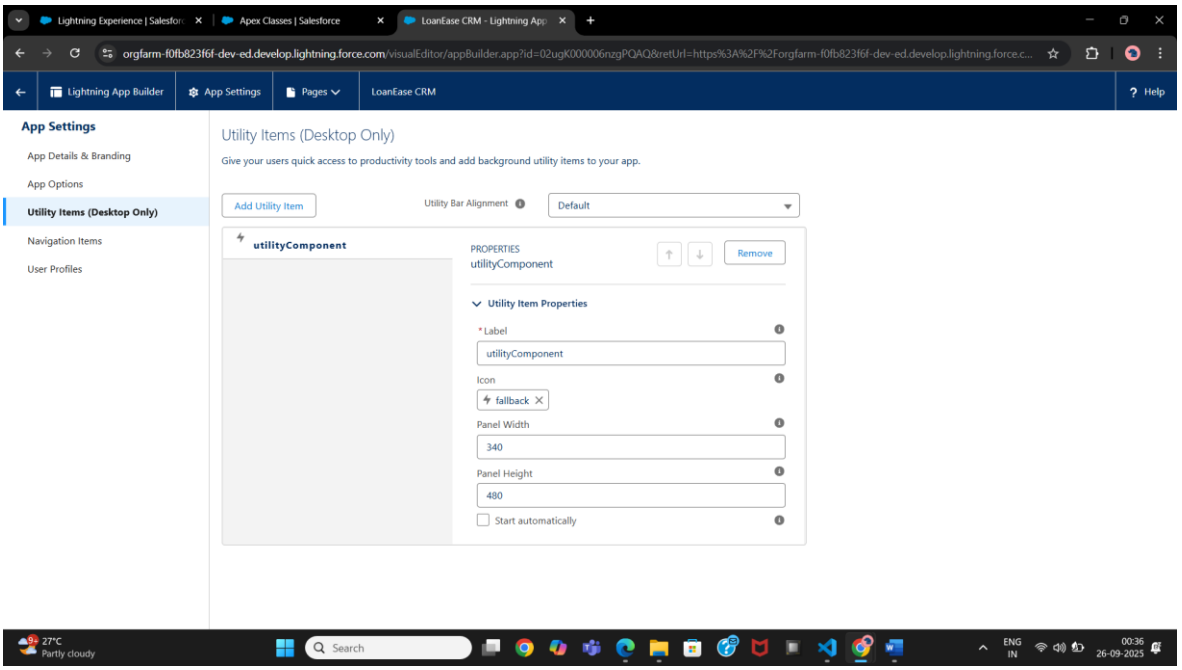
Home Page Layouts

- Added relevant Lightning Web Components to provide loan and EMI summaries, enhancing the dashboard experience for users upon login.
- Incorporated visual components such as charts, lists, and quick action buttons to enable efficient loan management awareness.
- Designed Home Pages to offer quick snapshots of loan portfolios, pending EMIs, and payment statuses.
- Ensured responsive design compatibility across desktop and mobile Lightning Experience.
- Home Page layouts serve as the central hub for users, streamlining workflows in the Loan Management System.



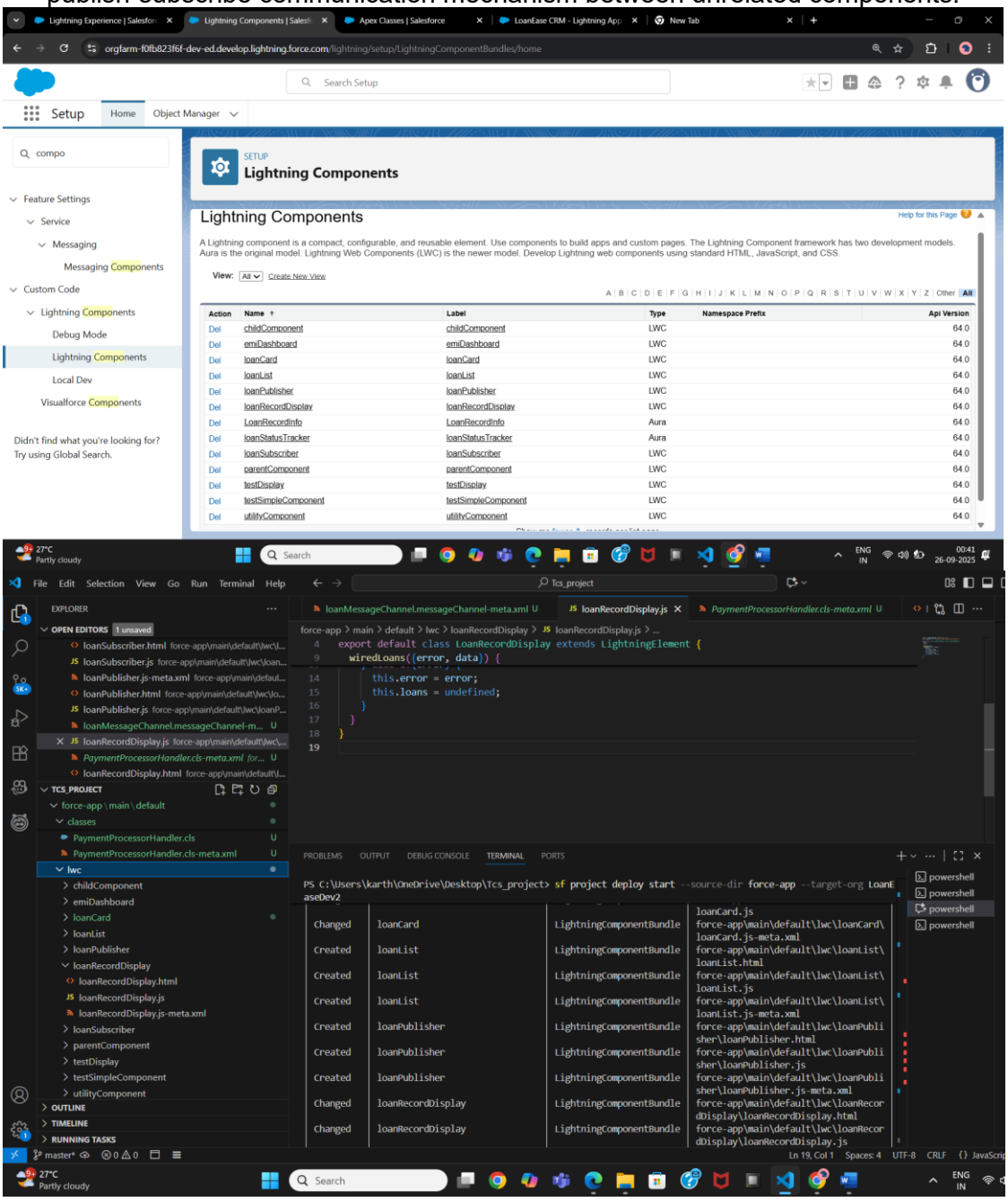
Utility Bar

- Configured the Salesforce Utility Bar to provide quick access tools and LWCs within the Lightning app.
- Added custom LWCs such as loan related quick actions, EMI dashboards, and payment trackers as utility items.
- Utility Bar enables users to interact with important features without leaving their current context or page.
- Helps improve workflow efficiency by making critical components easily accessible from any page.
- The bar can contain Lightning components, Visualforce pages, and standard utilities like notes and history.
- Customized appearance and behavior of the Utility Bar to fit user roles and business needs.



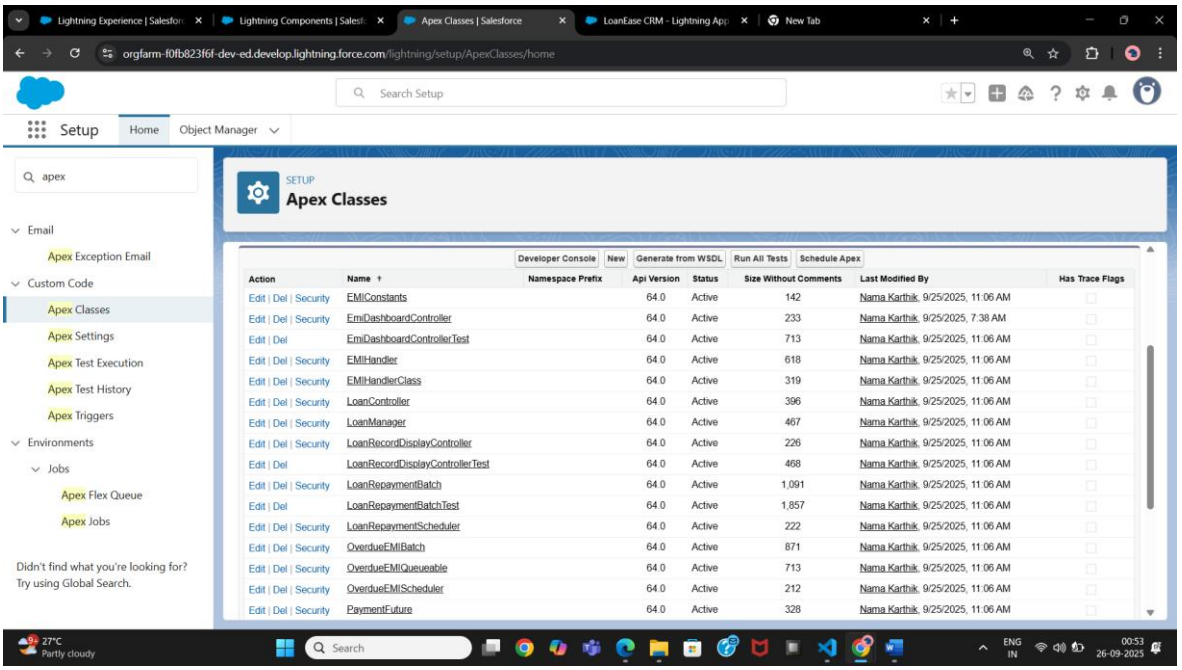
Lightning Web Components (LWC)

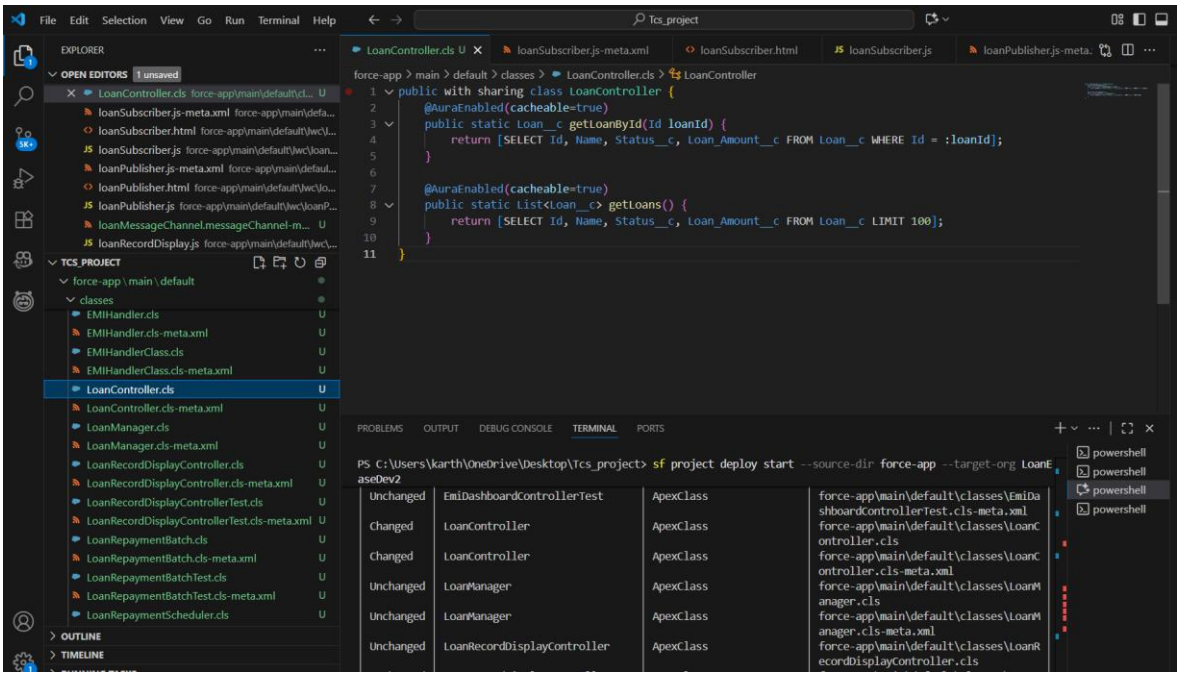
- LoanRecordDisplay: Displays comprehensive list and details of Loan records using wire service for reactive data fetching from Apex.
- loanCard: Shows individual loan details in a card format, reusable for loan listings and summaries.
- emiDashboard: Presents an overview dashboard of EMIs, including total and pending installments, assisting in EMI management.
- testSimpleComponents: A simple example component displaying static messages used for testing and demonstration of basic LWC features.
- testDisplay: Receives and shows text data passed from a parent component, demonstrating parent-to-child data binding.
- parent and child component: Implements child-to-parent communication via event dispatching and handling, illustrating event-driven component interaction.
- loanList: Fetches and displays a reactive list of loans using a wire adapter invoking Apex methods.
- loanPublisher and loanSubscriber: Uses Lightning Message Service to demonstrate publish-subscribe communication mechanism between unrelated components.



Apex with LWC

- Developed Apex classes exposing business logic and data access via @AuraEnabled methods.
- LWCs consume these Apex methods using wire service for reactive data or imperative Apex calls for explicit interaction.
- This integration supports dynamic UI updates based on backend data changes and user inputs.

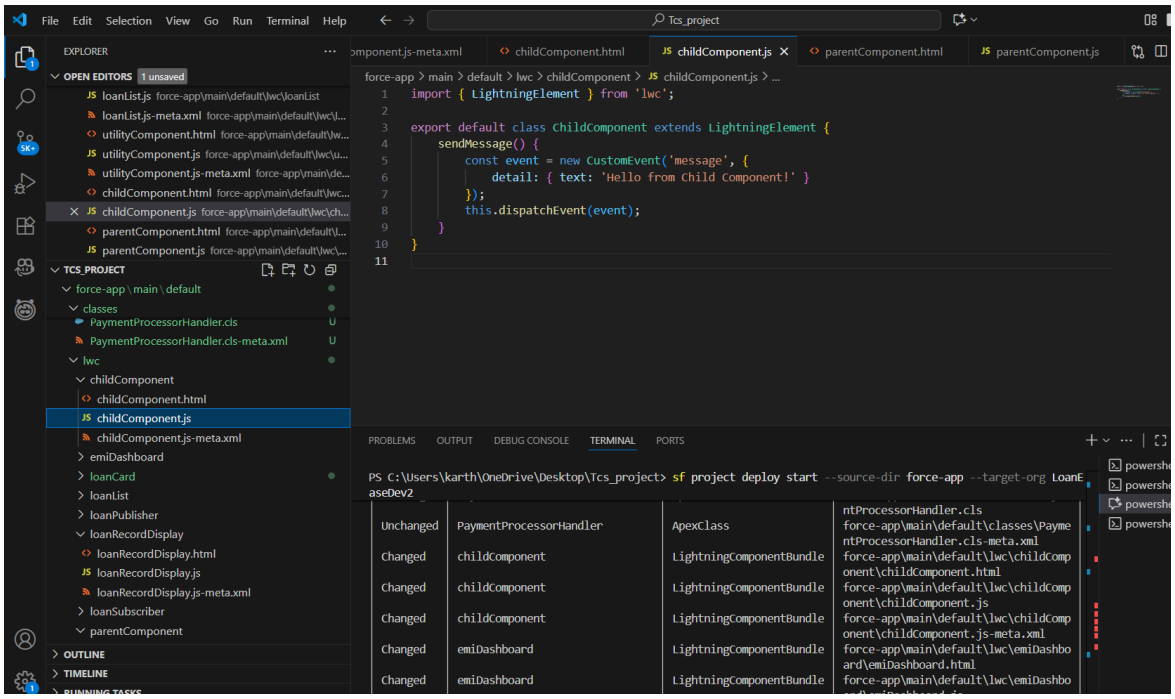




Events in LWC

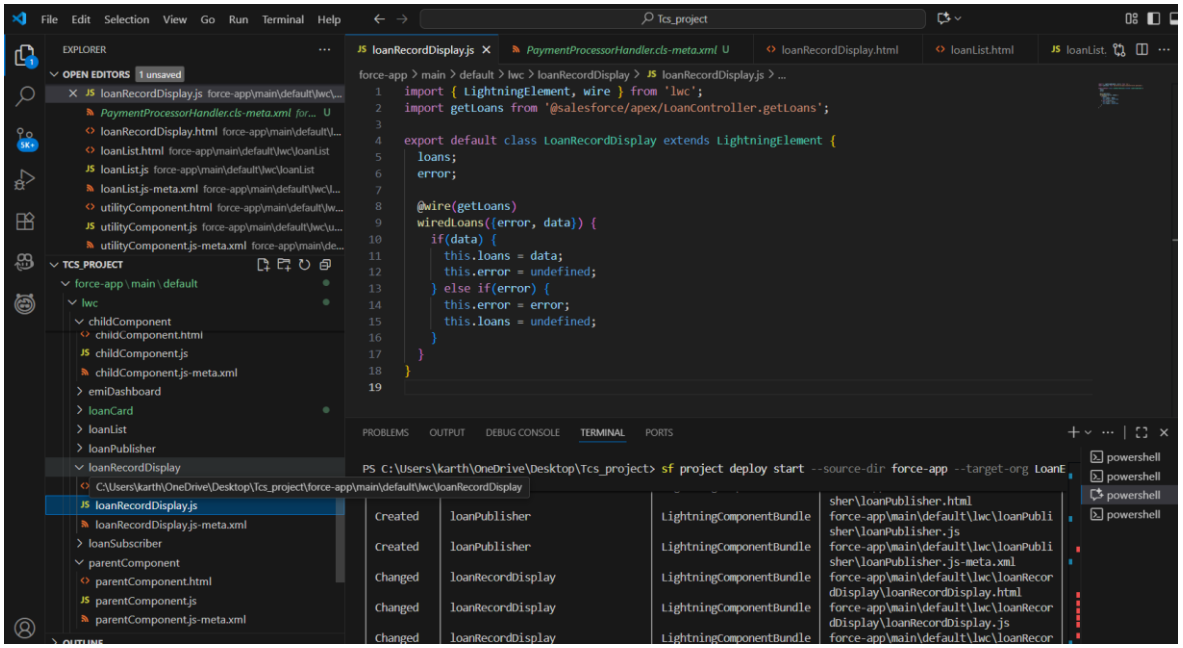
- Implemented component communication via custom events (child-to-parent).
- Used dispatchEvent in child components and handled events in parents to update UI or data states.
- Enables modular and loosely coupled UI design with interactive flows.

- Used in components like parentComponent and childComponent to demonstrate event-driven data flow.
- Allows parent components to respond to user actions or internal state changes within child components.
- Applied in UI flows where child components need to notify or trigger logic in parents without tight integration.



Wire Adapter

- Used wire service decorators (@wire) to reactively bind Apex data to component state.
- Handles automatic data refreshes and creates clean separation of concerns.
- Ideal for loading Salesforce records or metadata efficiently.



Imperative Apex Calls

- Used imperative method calls in LWCs to invoke Apex under user-triggered events.
- Allows manual control over server interaction and response handling.
- Facilitates functions like saving records or performing server-side calculations.

Navigation Services

- Utilized Lightning Navigation Service to programmatically redirect users and navigate between pages.
- Enables dynamic and context-sensitive navigation flows from LWCs.
- Used standard page references or custom URL routes.