**Phase 6: User Interface Development**

**Objective:**

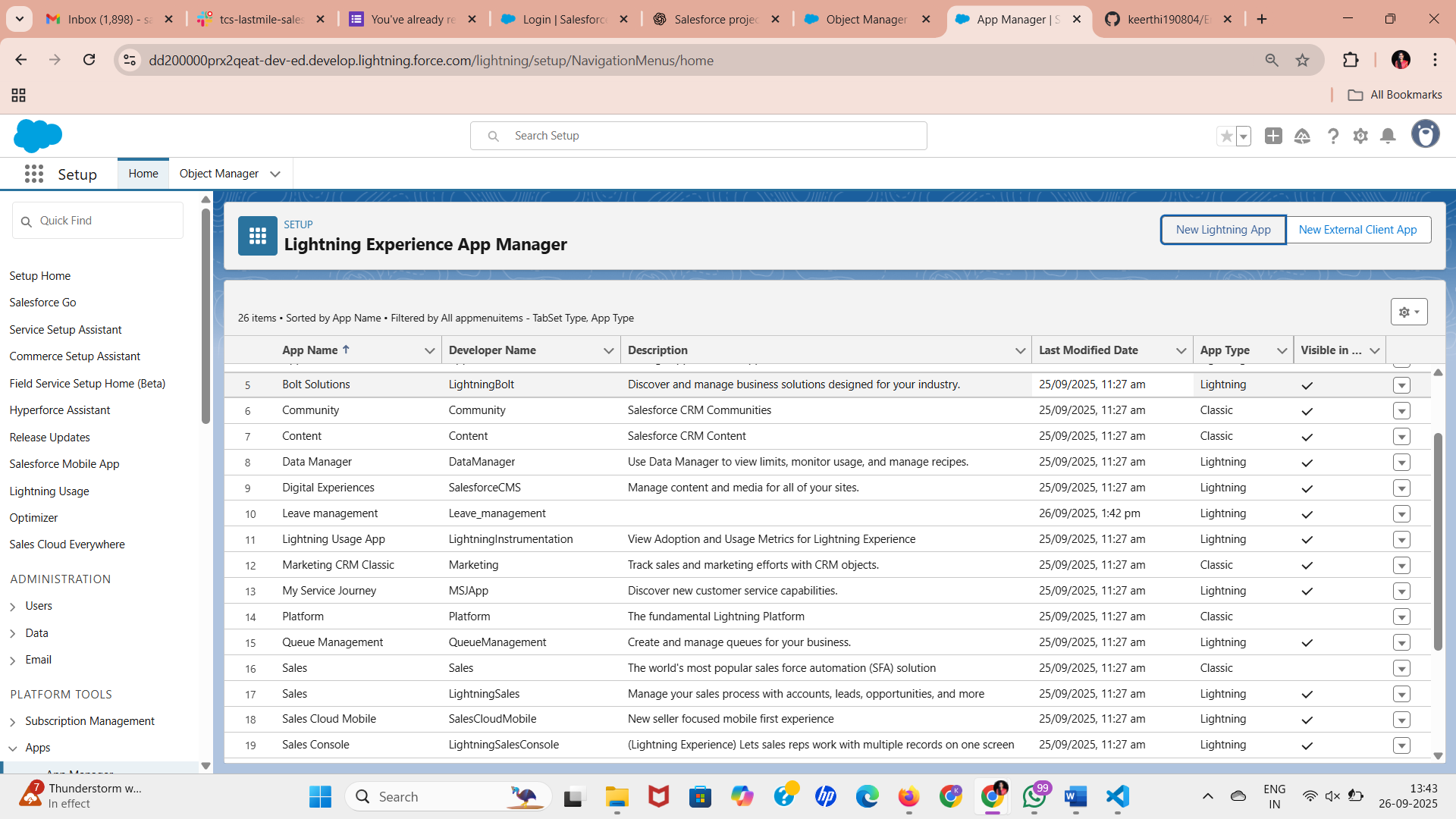
Create a **user-friendly interface** so users can submit leave requests, view balances, and approve/reject requests easily.

**Step 1: Lightning App Builder**

**Purpose:** Create a custom app to access all leave-related objects in one place.

1. Setup → **App Manager** → Click **New Lightning App**
2. Enter Name: Leave Management → Click **Next**
3. Choose Branding & Color (optional) → Next
4. Select Navigation Items → Add:
   * Leave Requests
   * Leave Balance
   * Reports (optional)
5. Assign User Profiles → Finish

Now your users can open the **Leave Management App** from the App Launcher.



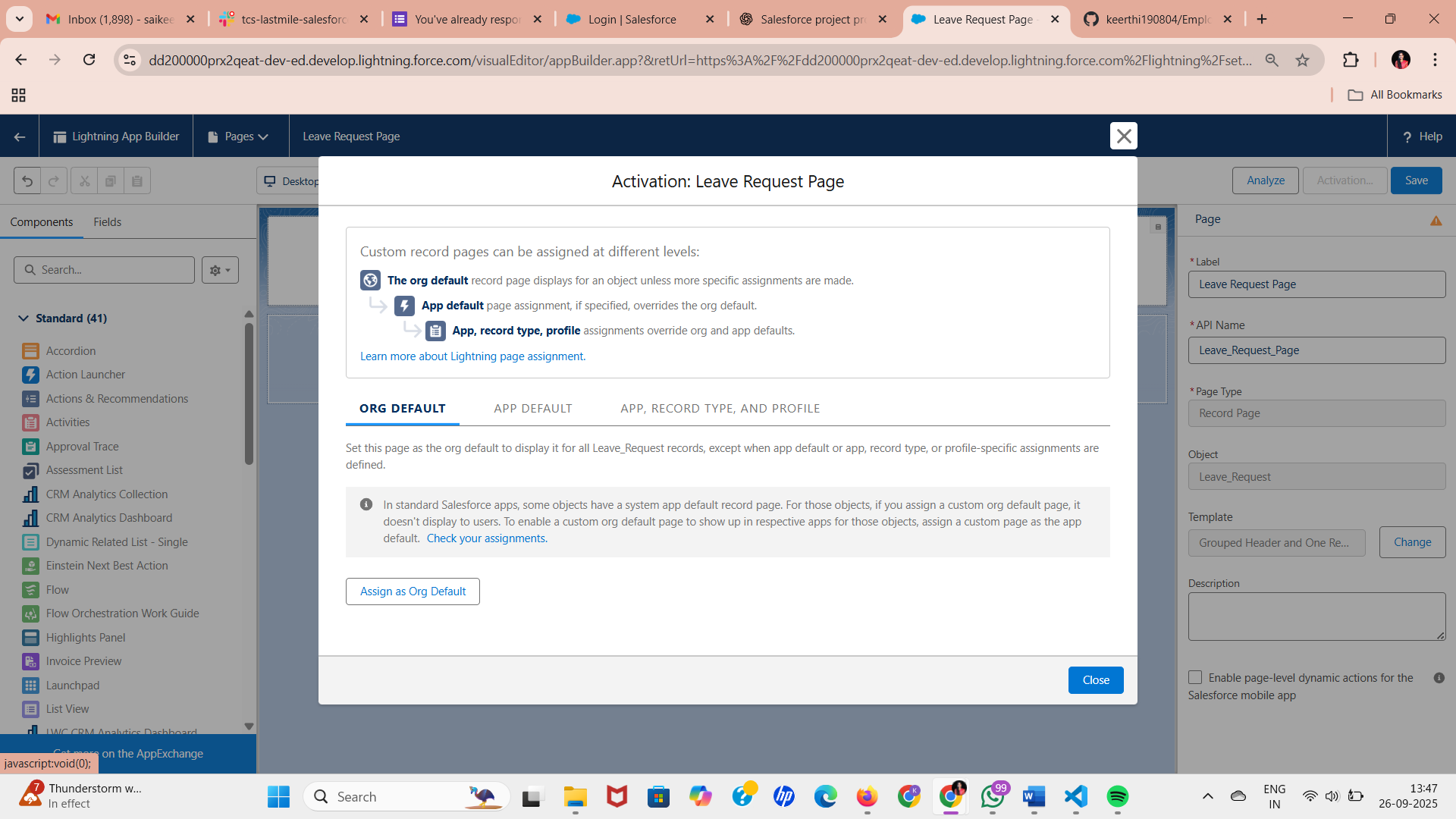
**Step 2: Customize Record Pages**

**Purpose:** Display only relevant fields and sections for each object.

1. Setup → Object Manager → Leave\_Request\_\_c → **Lightning Record Pages** → New
2. Choose Page Template → Give Name: Leave Request Page
3. Drag & Drop Components:
   * **Record Details** → Shows fields like Employee, Start Date, End Date, Status
   * **Related Lists** → Leave Approvals, History (if any)
4. Click **Save & Activate** → Set as default for all users

**Repeat for:**

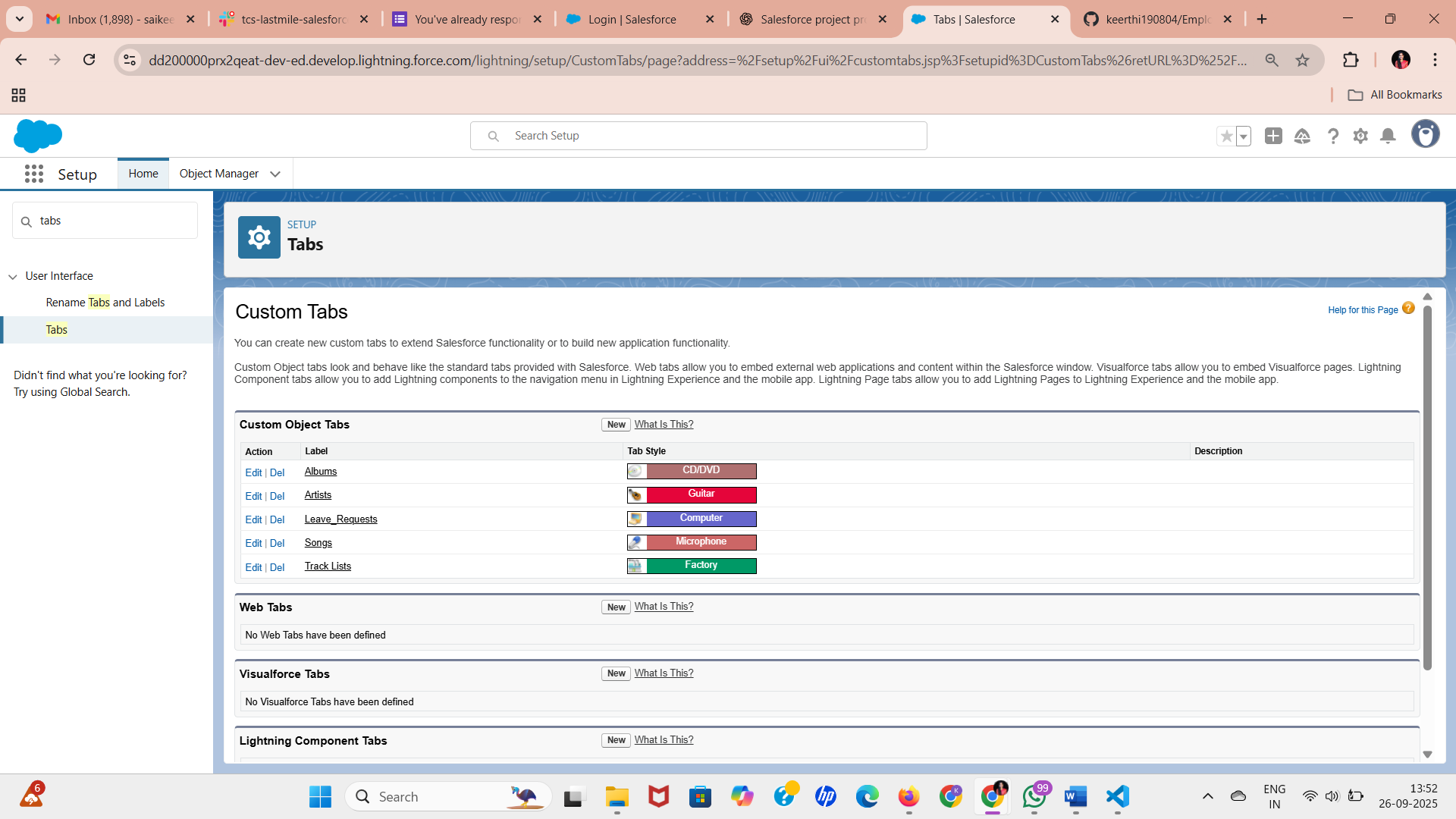
* Leave\_Balance\_\_c → Show Employee leave info



**Step 3: Tabs**

**Purpose:** Organize objects in navigation bar for easy access.

1. Setup → Tabs → Click **New Tab**
2. Choose Object → Leave\_Request\_\_c → Name it Leave Requests
3. Repeat for Leave\_Balance\_\_c → Leave Balance
4. Add tabs to your **Leave Management App**



**Step 4: Home Page Layouts**

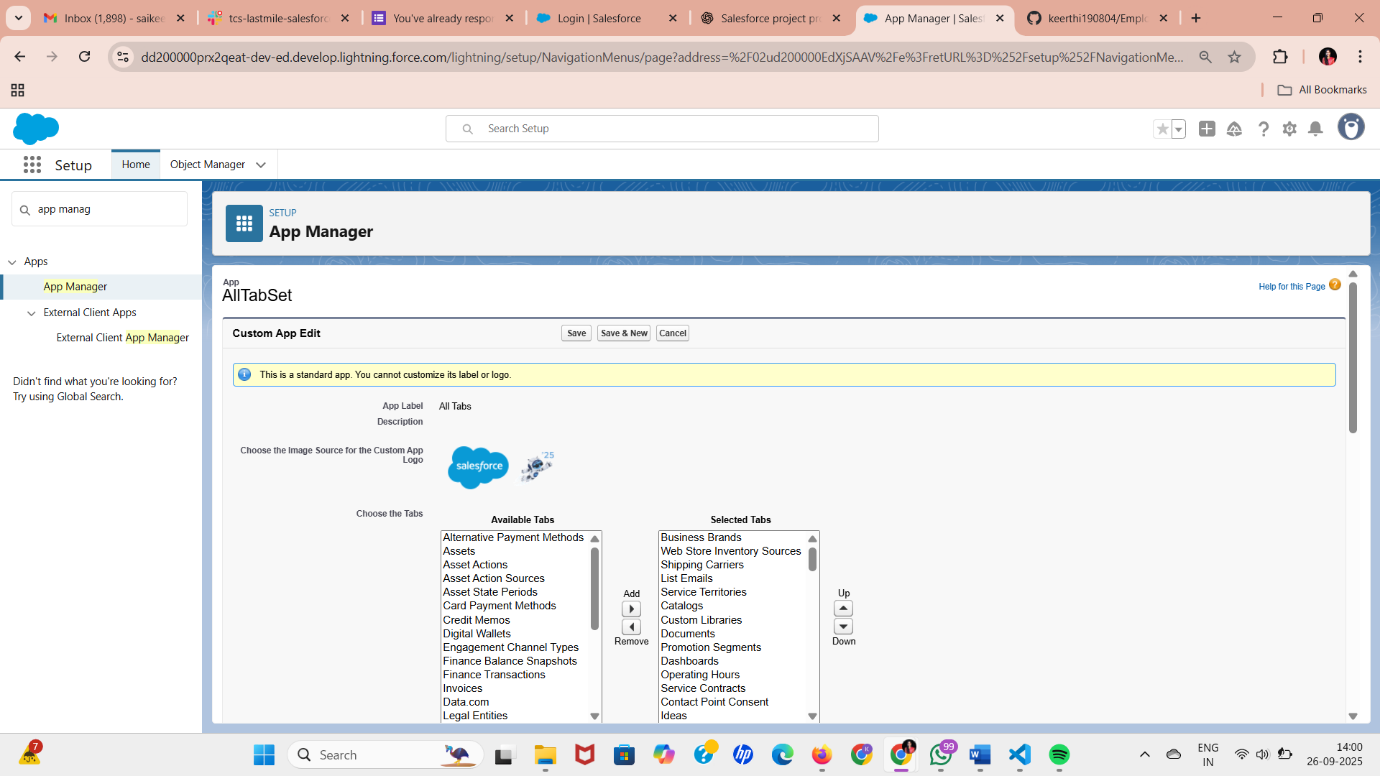
**Purpose:** Give employees a quick overview of leave status.

1. Setup → Lightning App Builder → **Home Page** → Clone Standard Home
2. Add Components:
   * **Report Chart** → Pending Leaves, Remaining Leaves
   * **Recent Leave Requests** → List component
3. Save & Assign → Set as default for Employee profile

**Step 5: Utility Bar**

**Purpose:** Quick access to important functions from any page.

1. Setup → App Manager → Edit Leave Management App → Utility Bar
2. Add Components:
   * **Create New Leave Request** → Quick Action
   * **View Leave Balance** → Object Page
3. Save



**Step 6: Lightning Web Components (Optional Advanced)**

**Purpose:** Custom interactive UI features for enhanced experience.

1. Developer Console → File → New → Lightning Web Component
2. Example: Custom calendar to select leave dates
3. Use Apex to fetch Leave Balance → Display dynamically

**Basic Example of LWC using Apex:**

// JS Controller

import { LightningElement, wire } from 'lwc';

import getLeaveBalance from '@salesforce/apex/LeaveCalculatorController.getLeaveBalance';

export default class LeaveBalanceWidget extends LightningElement {

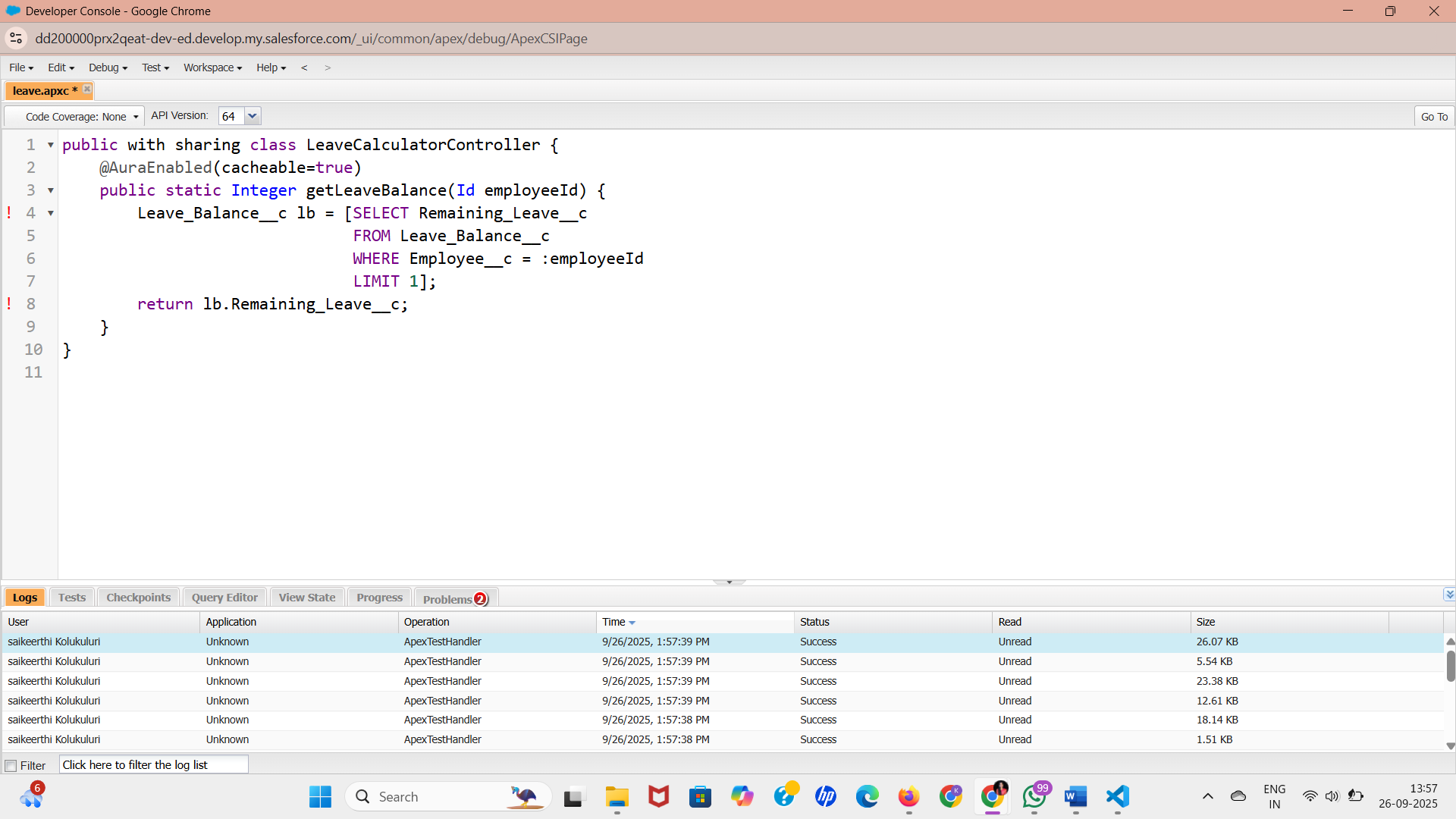
leaveBalance;

@wire(getLeaveBalance) wiredBalance({data, error}) {

if(data) this.leaveBalance = data;

}

}



**Step 7: Combine Everything in App**

* **Employees**: Submit leave, view leave balance
* **Managers**: Approve/reject leave requests
* **HR**: View all leave requests, generate reports