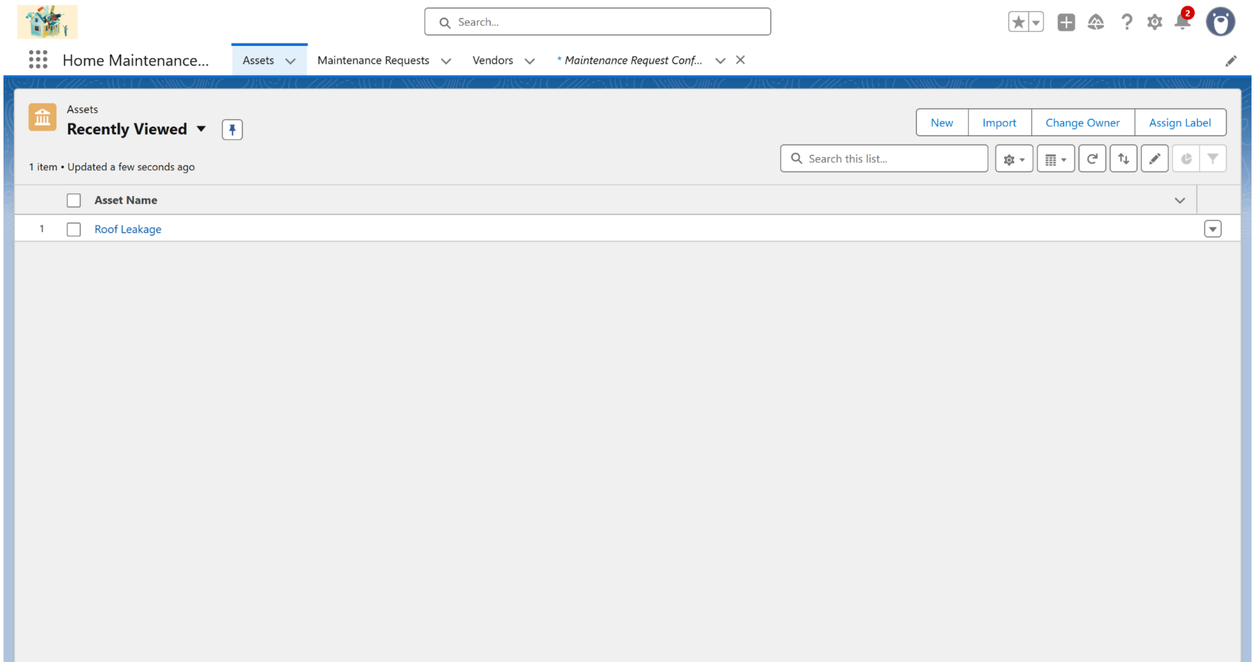
PHASE 6: USER INTERFACE DEVELOPMENT

Step 1: Lightning App Builder – Creating the Application

- Created a dedicated Home Maintenance and Repair App using Lightning App Builder.
- Consolidated key tabs such as Assets, Vendors, Maintenance Requests into one unified interface.
- Enabled users to quickly navigate across objects to perform maintenance operations efficiently.



Assets

Recently Viewed

1 item • Updated a few seconds ago

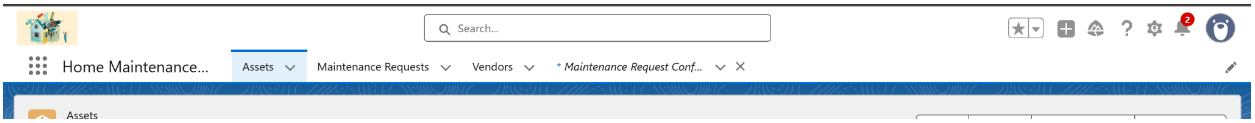
	Asset Name
1	Roof Leakage

Step 2: Record Pages

- Built custom Record Pages for Asset, Vendor, and Maintenance Request objects.
- Asset record page includes: asset details, related maintenance requests, and vendor information.
- Vendor record page includes: vendor profile, contact details, and related maintenance requests.
- Maintenance Request record page consolidates: request details, asset and vendor associations, and status updates.
- These pages provide a comprehensive 360° data view for each record.

Step 3: Tabs

- Created new custom tabs for Asset, Vendor, and Maintenance Request objects.
- These tabs make the custom objects accessible directly from the app's navigation menu for ease of access.



Step 4: Home Page Layouts

- Developed customized Home Page layouts targeted at Maintenance Managers.
- Home Page features:
    - List view of recent or open maintenance requests.
    - Quick links/buttons to create new assets, vendors, or maintenance requests.
    - Dashboards summarizing metrics like pending requests and vendor performance.
- Provides actionable insights immediately upon login.

Step 5: Utility Bar

- Added a Utility Bar within the app for quick access to common tasks.
- Includes:
    - Direct access to maintenance reports.
    - Recent Items for faster navigation to frequently used records.
- Utility Bar is persistently visible at the bottom of the app interface.

Step 6: Lightning Web Components (LWCs)

- Developed custom LWCs to extend functionality and enhance user experience.
- Examples include:
    - Search Maintenance Requests: component to filter requests by asset, vendor, or status.
    - Asset Detail Viewer: modular display of asset information and maintenance history.

Step 7: Apex Integration with LWCs

- LWCs are connected to Apex controllers for server-side data handling.
- Apex handles complex queries for assets and maintenance requests in real-time.

- Ensures dynamic, responsive UI without needing full page reloads.

Step 8: Parent-Child Component Communication

- Implemented event-driven communication between LWCs.
- Example: Selecting an asset in the search component sends an event ID to the maintenance request form component.
- Facilitates modular, reusable components improving system maintainability.

ACHIEVEMENTS IN PHASE 6

- Delivered a unified Lightning App focused on efficient home maintenance management.
- Created tailored record pages and home pages supporting role-specific workflows.
- Integrated custom LWCs enhancing the user interface with search and detail display.
- Established Apex-LWC communication for real-time data retrieval and updates.
- Enhanced user productivity with Utility Bar and leveraged modern navigation and layout best practices.
- Improved overall application usability combining Salesforce standard capabilities with custom extensions.