Phase 6: User Interface Development – GreenFuture CRM

1. Lightning App Builder

Purpose:
Create custom applications and user experiences for different stakeholders (Sustainability Managers, Auditors, Partners).

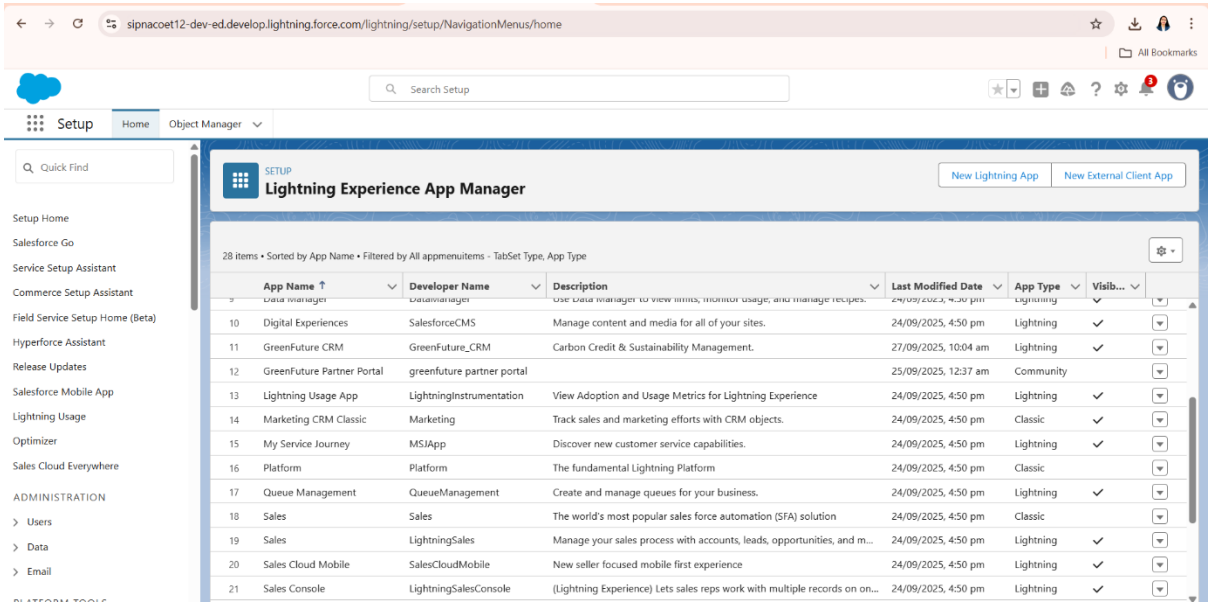

Fig 6.1. – App Manager showing “GreenFuture CRM”

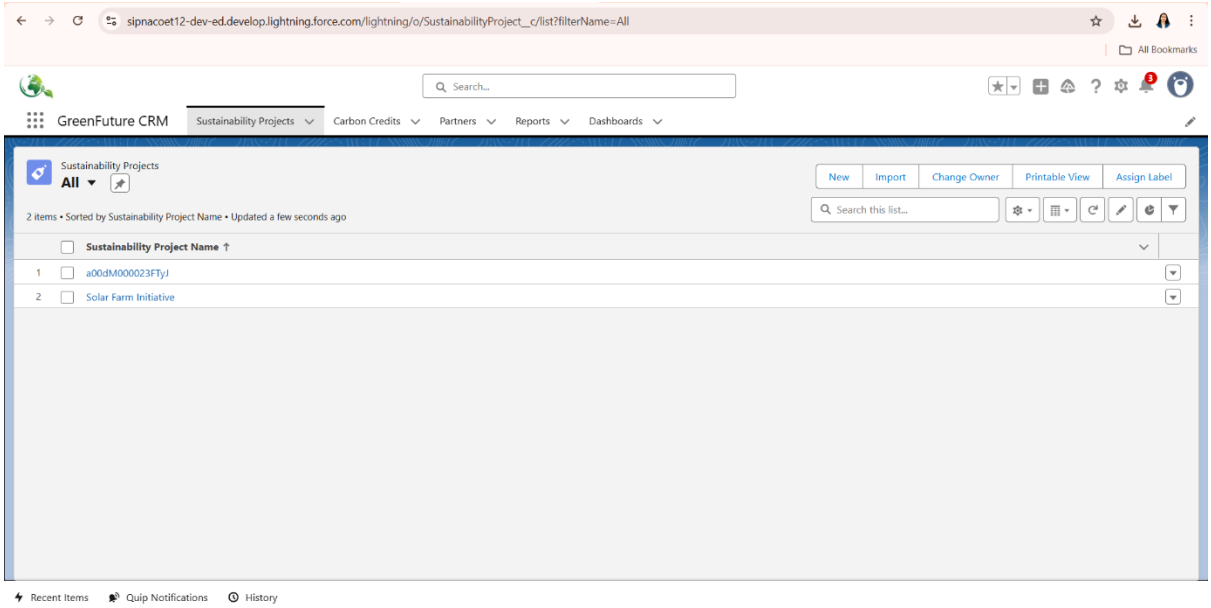

Fig 6.2. – Lightning App Setup Wizard

2. Record Pages

Purpose:
Customize record views for **Projects, Credits, Partners** with related lists and key highlights. Each stakeholder gets a focused view with relevant fields, related lists, and components.

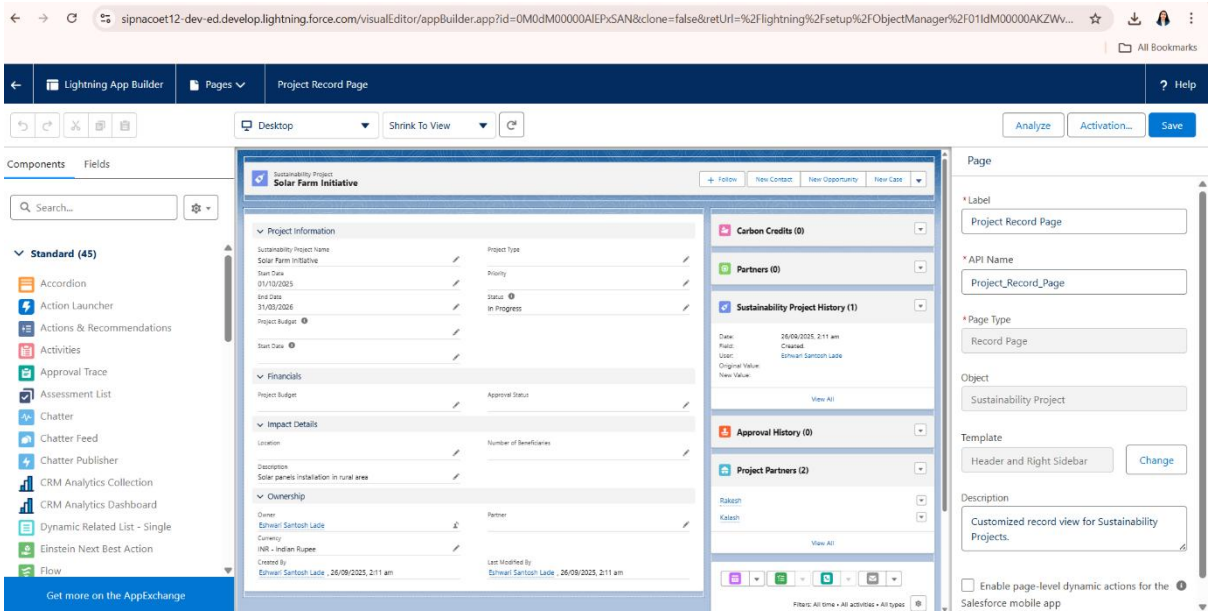

Fig.6.3. Project Record Page in Lightning App Builder

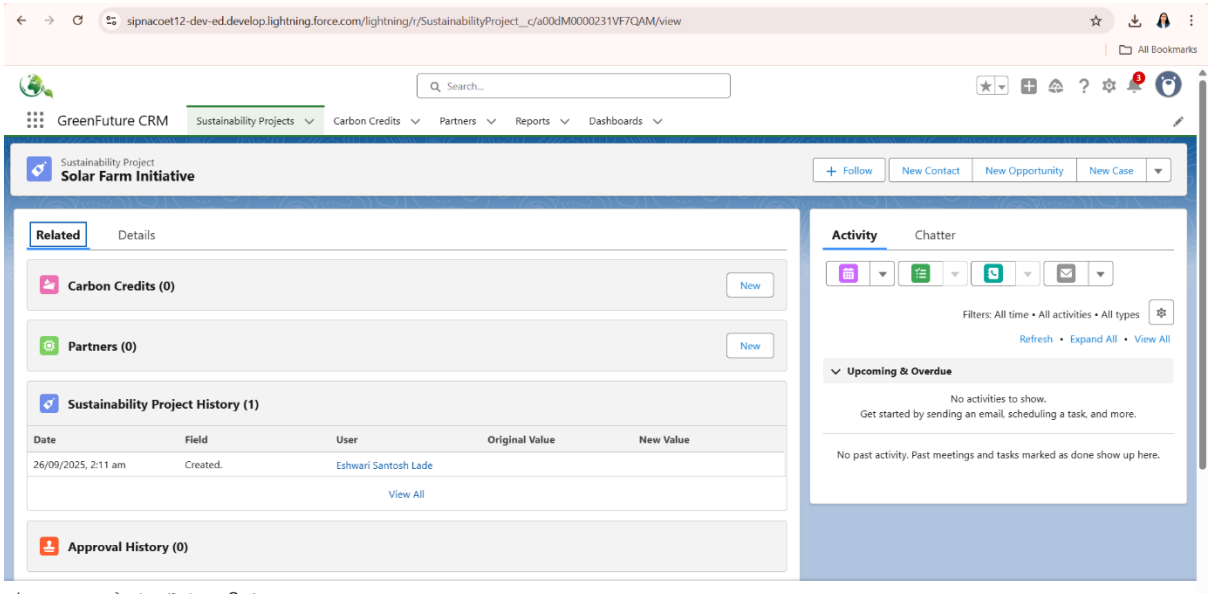

Fig.6.4. Project Record Page Review

3. Tabs

Purpose:
Provide quick navigation for custom objects.

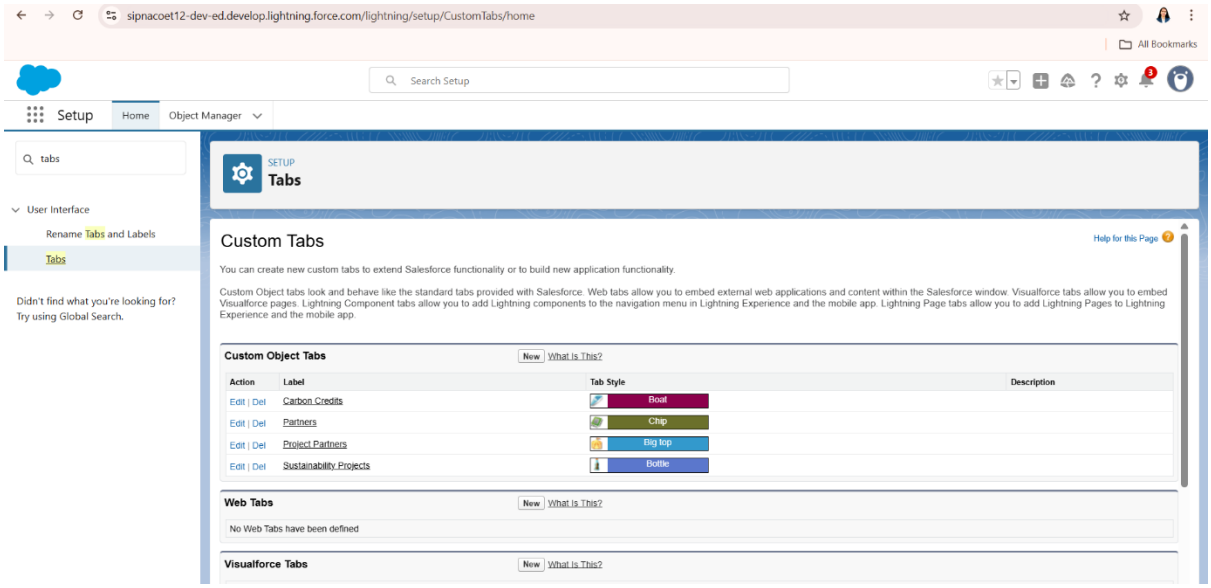

Fig.6.5. Tabs configuration page

4. Home Page Layouts

Purpose:
Customize the home page with dashboards.
Show key metrics on login.

Scenario:

- **Sustainability Manager** should see: Total Projects, Active Credits, Expiring Credits.
- **Auditor** should see: Compliance Reports and Pending Approvals.

Implementation Steps:

1. Go to **Setup → Lightning App Builder → New Home Page**.
2. Add:
    - **Report Chart:** Credits Issued vs Sold.
    - **Recent Projects list.**
3. • Save → Assign to **Sustainability Manager** profile.
4. • Clone and adjust for **Auditor profile**.

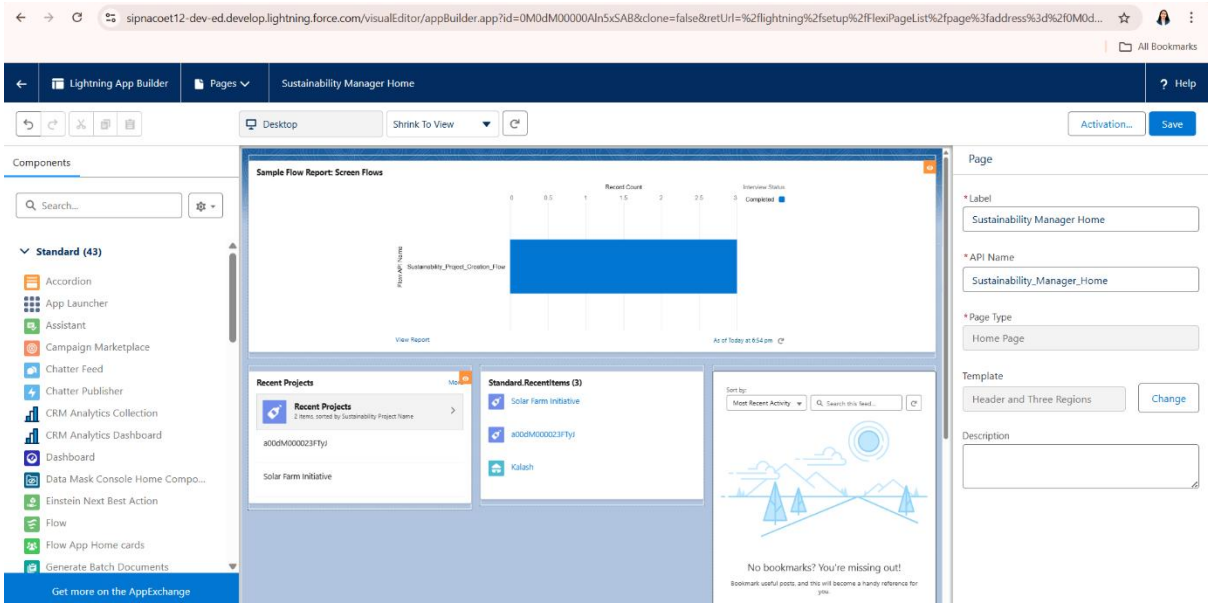

Fig 6.6 – Home Page

5. Utility Bar

Purpose:
Provide quick access tools.

Scenario:

- **Manager** wants to quickly take notes during project review.
- **Auditor** wants to check recent history while cross-verifying records.

Implementation Steps:

1. Setup → App Manager → GreenFuture CRM → Edit → Utility Items.
2. Add:
    - Notes.
    - History.
    - Notifications.
3. Save and refresh app.

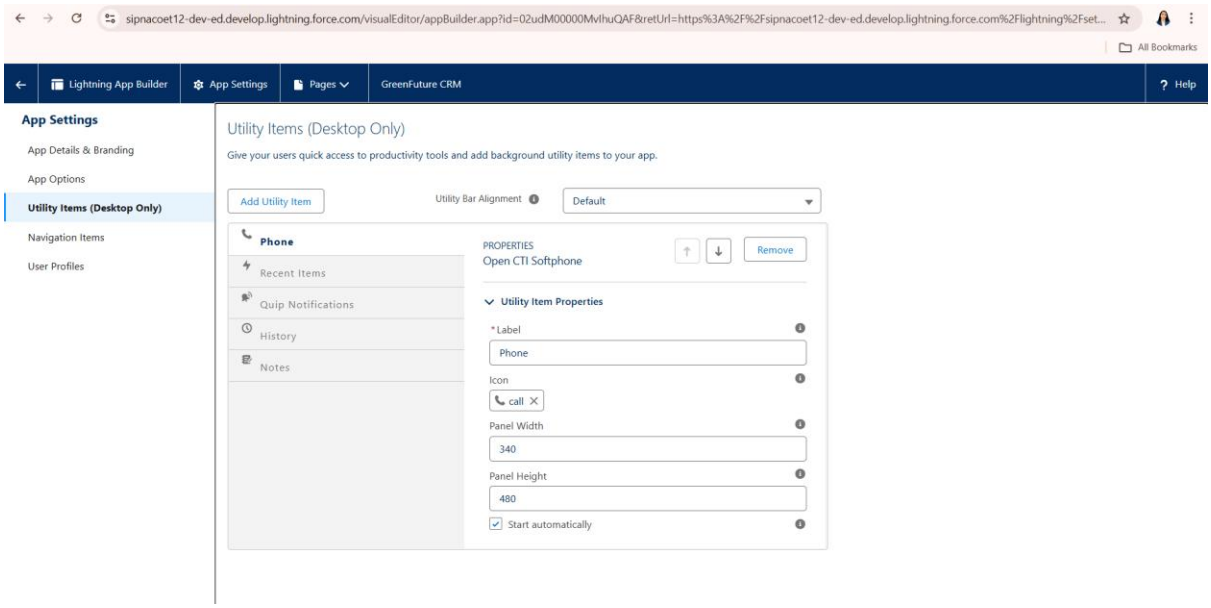

Fig 6.7 – Utility bar setup page.

6. Lightning Web Components (LWC)

Purpose:
Custom dashboards & interactivity.

Scenario:

- **Manager** should see a **Dashboard LWC** with project and credit stats.
- **Auditor** should see compliance-focused dashboards.

Implementation Steps:

1. Open VS Code → Terminal.
2. Run:
3. sfdx force:lightning:component:create --type lwc --componentname projectDashboard --outputdir force-app/main/default/lwc
4. Create **DashboardController Apex** and connect via @AuraEnabled.
5. Add LWC to **Home Page** in App Builder.

Context:
During the GreenFuture CRM project, a custom Lightning Web Component (LWC) projectDashboard was developed to display project and carbon credit statistics on the Home Page.

Observation:
Attempts to add the projectDashboard LWC to the Home Page in the **DevHub org** were unsuccessful. The component did not appear in the **Lightning App Builder**, and deployment attempts resulted in connectivity or component errors.

Reason:

- The **DevHub org** is designed exclusively for **creating and managing Scratch Orgs** and enabling source tracking.
- DevHub **does not support deploying custom Apex classes or Lightning Web Components to its Home Page**.
- Any attempt to use the DevHub as a production-like environment for app testing will fail because it lacks standard Lightning Page hosting capabilities.

Resolution / Best Practice:

- A **Scratch Org** (or Sandbox/Production org) must be used to deploy and test LWCs and Home Page customizations.