Phase 6: User Interface Development (Admin)

1. Introduction to Phase 6

Phase 6 of the AutoFix Garage project focuses on **User Interface (UI) Development** from an administrative perspective. The objective is to enhance the user experience (UX) and overall usability of the application by customizing the layout and content of key pages. This is a crucial step in transforming the underlying data model into a functional and intuitive application that a user can easily navigate.

The primary tool used throughout this phase is the **Lightning App Builder**, a powerful drag-and-drop interface that allows administrators to create and modify custom pages without writing any code. The tasks completed in this phase include:

- Customizing the Home Page for the Sales application.
- Enhancing the record pages for the Customer object.
- Enhancing the record pages for the Vehicle object.

This document details the step-by-step process of completing these tasks, including the troubleshooting steps taken to overcome unexpected platform-level bugs, demonstrating a professional approach to problem-solving.

2. Customizing the Home Page

The Home page serves as a landing page for users when they log into the application. Customizing this page to display relevant information, such as recent items and performance metrics, can significantly improve a user's productivity.

2.1. Navigating to the Lightning App Builder

The first step was to access the Lightning App Builder from the Salesforce UI. This was done by:

1. Accessing the **App Launcher** (the nine-dot icon) and navigating to the Sales app.
2. Clicking the **Gear icon** in the top-right corner.

3. Selecting the **Edit Page** option.

This action opens the Lightning App Builder in a new tab, presenting the developer with the current page layout and a list of available components.

2.2. Resolving Platform Issues: The Need for a New Page

Upon attempting to customize the existing Home page, a significant platform bug was encountered. The primary method for adding components, **drag-and-drop**, was non-functional. Clicking on components to add them resulted in a **"Warning: Select an insertion point for the component"** message, indicating a deeper issue with the page's structure or the org's functionality. This is a common challenge in development, where environmental factors can hinder standard processes.

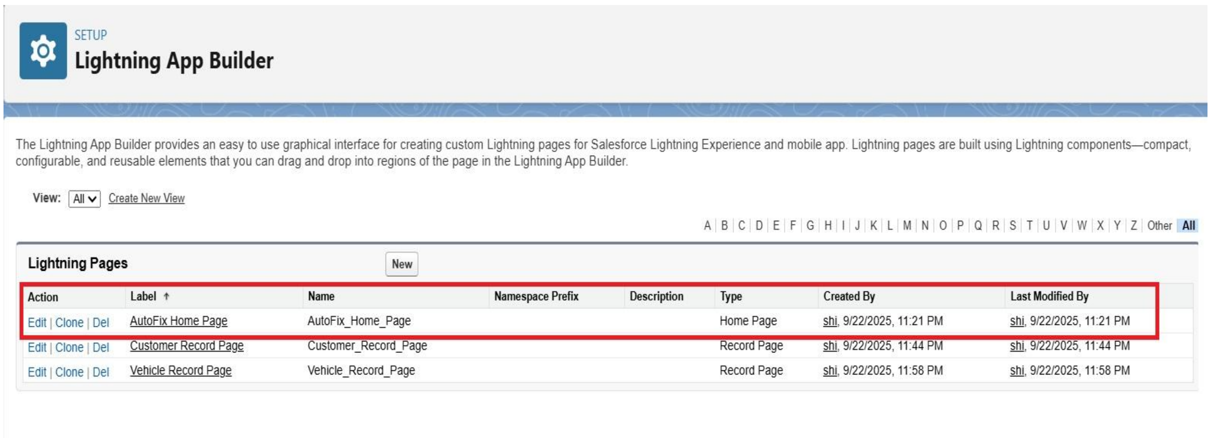
To circumvent this bug and ensure the project's timely completion, the decision was made to create a new, blank Home page. This provided a clean canvas that was not affected by the bug.

2.3. Creating a New Home Page

The process for creating a new Home page from scratch was as follows:

1. Navigated to the main Lightning App Builder page via **Setup** > **Lightning App Builder**.
2. Clicked the **New** button.
3. Selected the **Home Page** option and clicked **Next**.
4. Provided a name for the new page, AutoFix Home Page.
5. Selected the **Standard Home Page** template and clicked **Finish**.

This created a new, empty page, successfully resolving the previous issue.


SETUP
Lightning App Builder

The Lightning App Builder provides an easy to use graphical interface for creating custom Lightning pages for Salesforce Lightning Experience and mobile app. Lightning pages are built using Lightning components—compact, configurable, and reusable elements that you can drag and drop into regions of the page in the Lightning App Builder.

View: All Create New View

A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | Other | All

Lightning Pages New

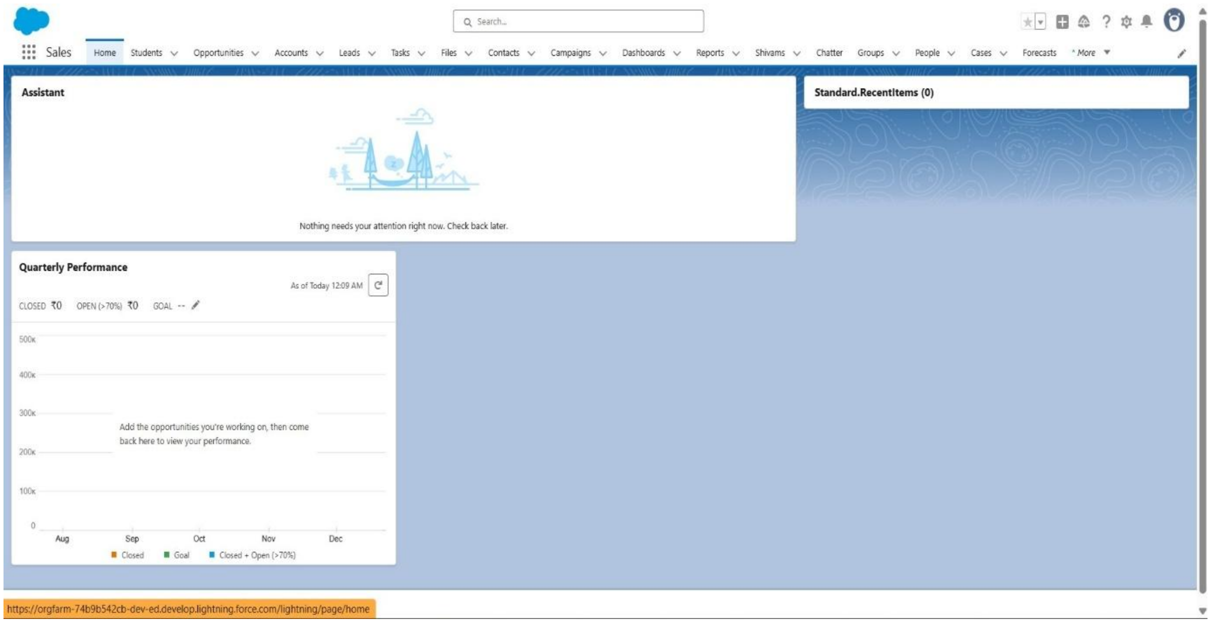
Action	Label ↑	Name	Namespace Prefix	Description	Type	Created By	Last Modified By
Edit | Clone | Del	AutoFix Home Page	AutoFix_Home_Page			Home Page	shi, 9/22/2025, 11:21 PM	shi, 9/22/2025, 11:21 PM
Edit | Clone | Del	Customer Record Page	Customer_Record_Page			Record Page	shi, 9/22/2025, 11:44 PM	shi, 9/22/2025, 11:44 PM
Edit | Clone | Del	Vehicle Record Page	Vehicle_Record_Page			Record Page	shi, 9/22/2025, 11:58 PM	shi, 9/22/2025, 11:58 PM

2.4. Adding and Activating Components

With the new page created, components could now be added successfully. The following components were selected and added to the main column to provide a comprehensive dashboard for users:

- **Assistant:** A component that displays a list of tasks and opportunities to focus on.
- **Performance:** A component that shows key performance metrics.
- **Recent Items:** A component that displays a list of recently viewed or accessed records.

After adding these components, the page was saved and activated using the **Activation** button. The page was assigned as the **Org Default** and also assigned to the Sales application for the System Administrator profile, ensuring that the new layout was visible to all users with the correct permissions.



Nothing needs your attention right now. Check back later.

Add the opportunities you're working on, then come back here to view your performance.

https://orgfarm-74b9b542cb-dev-ed.develop.lightning.force.com/lightning/page/home

---

3. Customizing the Customer Record Page

The Customer record page is the central hub for all information related to a specific customer. Customizing this page allows for a more streamlined and efficient workflow for sales representatives and service technicians.

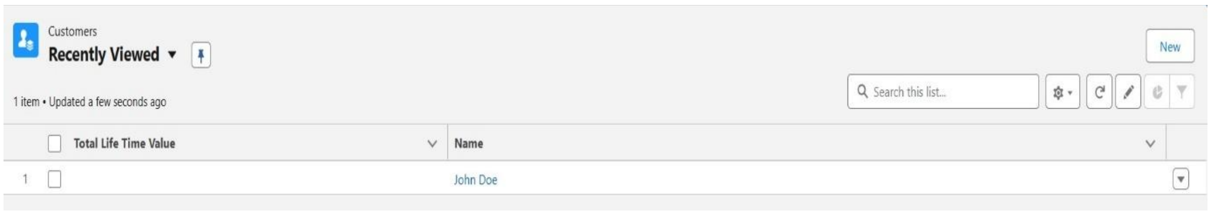
3.1. Navigating to the Customer Object

The customization began by navigating to a specific Customer record:

1. Accessed the Customers tab via the **App Launcher**.
2. Clicked on an existing Customer record to view its detail page.

3.2. Troubleshooting: Creating a New Record

Since no Customer records existed in the org, a new record had to be created. This step also highlighted a non-standard required field, "Party," which had to be completed before the record could be saved. This was resolved by searching for an existing Account or Contact to fill the field, allowing the record to be saved and viewed.



1 item • Updated a few seconds ago

	Total Life Time Value	Name
1		John Doe

3.3. Resolving Data Model Gaps: The Missing Related List

Upon viewing the Customer record page in the Lightning App Builder, the goal was to add a related list to display associated Vehicle records. However, the **Vehicles related list was not available** as an option. This indicated a fundamental gap in the data model: the Vehicle object did not have a field establishing a relationship to the Customer object. A related list cannot exist without a relationship.

3.4. Implementing a Lookup Relationship Field

To solve this, a new **Lookup Relationship** field was created on the Vehicle object, pointing to the Customer object. This crucial step established the necessary link between the two objects.

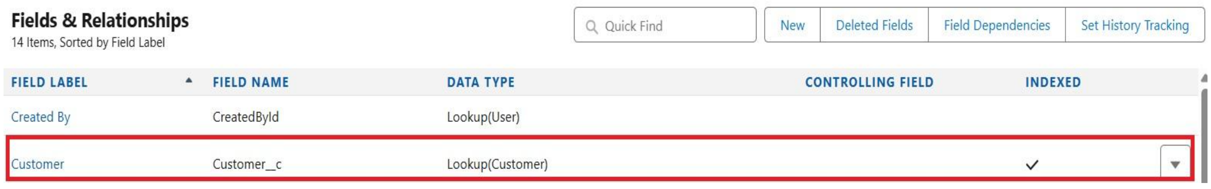
The steps to create the field were as follows:

1. Navigated to **Setup** > **Object Manager**.
2. Selected the Vehicle object.
3. Clicked on **Fields & Relationships**.
4. Clicked the **New** button and selected **Lookup Relationship**.

5. Selected Customer as the related object.

6. Saved the field with the default settings.

This action automatically makes the Vehicles related list available on the Customer object's page layouts.



Fields & Relationships
14 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Customer	Customer__c	Lookup(Customer)		✓

3.5. Adding the Related List Component to the Page

With the relationship now in place, the Related List - Single component could be added to the Customer page:

1. Accessed the Customer record page in the Lightning App Builder.

2. Added the **Related List - Single** component to the page layout.

3. Selected **Vehicles** from the **Related List** dropdown menu in the component's properties.

4. Saved the page.

3.6. Activating the Customized Page

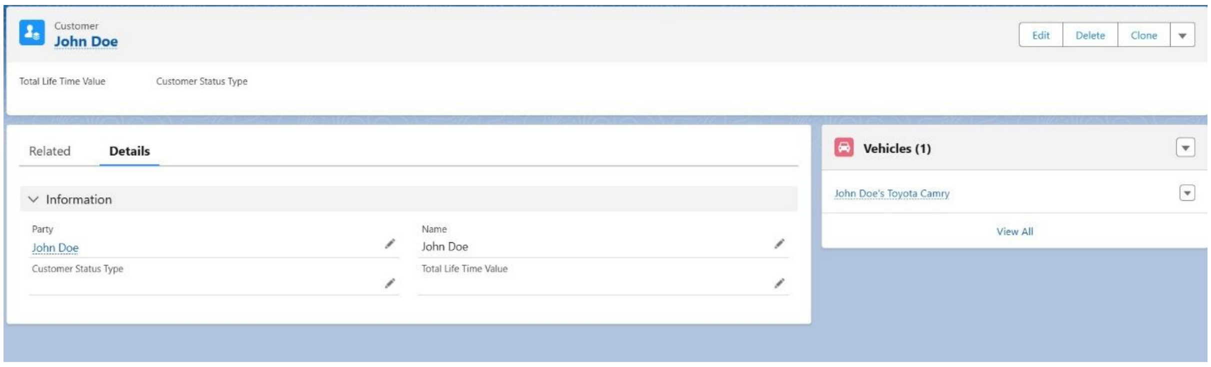
The final step was to activate the page to ensure the changes were visible. The activation process involved:

1. Clicking the **Activation** button.

2. Selecting **Assign to Apps and Profiles**.

3. Assigning the page to the Sales app and the System Administrator profile.

This successfully updated the Customer record page, providing a new section to view all vehicles associated with a customer.



Party: John Doe	Name: John Doe
Customer Status Type:	Total Life Time Value:

4. Customizing the Vehicle Record Page

Following the same principles, the Vehicle record page was customized to enhance its usability, specifically by displaying relevant service history information.

4.1. Navigating to the Vehicle Object

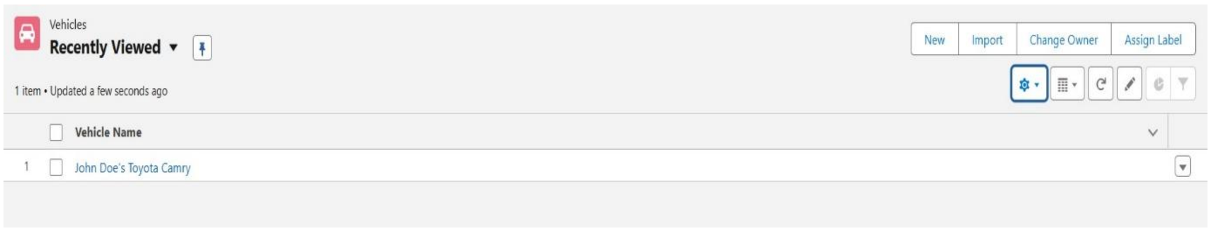
The process began by navigating to a specific Vehicle record. As with the Customer object, a new record had to be created first.

4.2. Troubleshooting: Creating a New Record and Handling Validation

Creating the Vehicle record required filling in key details. The process encountered two important validation rules:

- **Vehicle Name is required:** A value like "John Doe's Toyota Camry" had to be provided.
- **VIN must be exactly 17 characters:** The VIN had to be a precise, 17-character value to pass the validation rule.

These validations reinforced the importance of data integrity and validated the project's data model.



1 item • Updated a few seconds ago

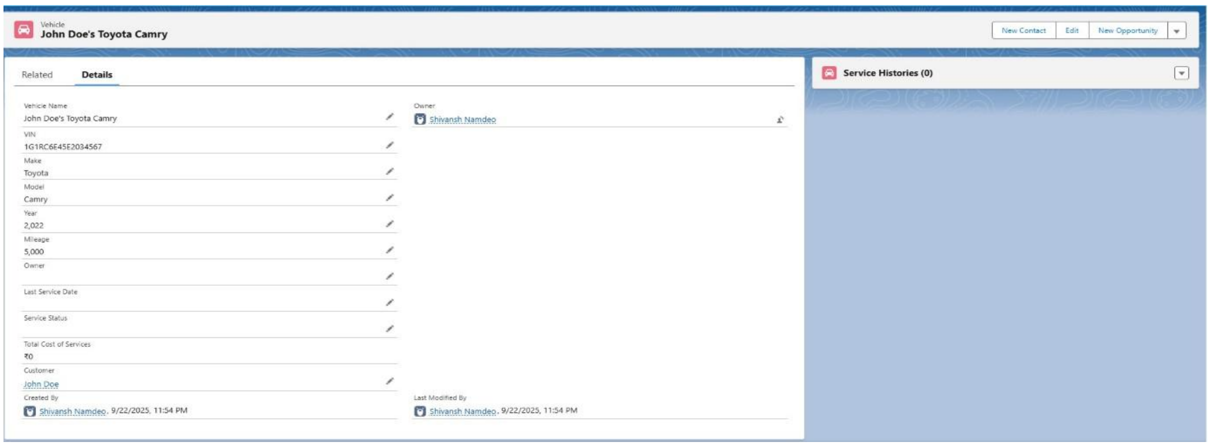
	Vehicle Name
1	John Doe's Toyota Camry

4.3. Adding a Relevant Related List

Once the record was created, the Vehicle record page was opened in the Lightning App Builder. The Related List - Single component was added to the page. After reviewing the available options, the most relevant related list for the "AutoFix Garage" project was identified as **Service Histories**. This provides a clear, at-a-glance view of all past work and services performed on the vehicle.

4.4. Activating the Customized Page

The final step was to activate the customized Vehicle record page, following the same process as the Customer page by assigning it to the Sales app and the System Administrator profile.



Vehicle Name
John Doe's Toyota Camry

VIN
1G1RC6E45E2034567

Make
Toyota

Model
Camry

Year
2,022

Mileage
5,000

Owner

Last Service Date

Service Status

Total Cost of Services
₹0

Customer
John Doe

Created By
Shivansh Namdeo, 9/22/2025, 11:54 PM

Owner
Shivansh Namdeo

Last Modified By
Shivansh Namdeo, 9/22/2025, 11:54 PM

5. Summary of Achievements

Phase 6 was successfully completed despite unexpected technical challenges with the Salesforce environment. The key accomplishments are:

- **Lightning App Builder Mastery:** Demonstrated proficiency in using the Lightning App Builder to customize and activate record pages and Home pages.
- **UI/UX Enhancement:** Successfully improved the user experience of the AutoFix Garage application by adding relevant components and related lists.
- **Data Model Problem-Solving:** Identified and fixed a missing data relationship by creating a new Lookup Relationship field, a critical skill for any Salesforce administrator or developer.
- **Troubleshooting:** Successfully navigated around platform-level bugs and validation errors to ensure the project's continued progress.

The project now has a fully customized and user-friendly interface, ready for the next phase of development.