# Community Fitness & Wellness Hub CRM Project

**Target Users:-** Members (The Customers), Studio Staff / Trainers, Management & Studio Owner.

## Problem Statement

The client, a local fitness studio with multiple locations, is facing significant operational challenges manual processes. The current system relies on a mix of spreadsheets, calendars, and phone calls to manage key business activities. This Results in:

- **Inefficient Operations:** Class bookings, trainer scheduling, and member communication are handled manually, causing delays and frequent errors. This results in double-booked classes and missed appointments.
- **Poor Member Experience:** Customers are frustrated by the lack of a simple, online booking system and inconsistent communication regarding class details and updates. The absence of a structured loyalty program also fails to encourage repeat visits.
- **Lack of Business Insight:** Management lacks real-time data on class popularity, member attendance, and trainer performance, hindering their ability to make data-driven decisions.
- **No Engagement Strategy:** There is no structured loyalty or rewards program to retain and motivate members.

To address these challenges, the goal is to implement a Salesforce-based platform that will serve as a single source of truth. The solution will centralize all member and class management activities, automating key processes to enhance operational efficiency, improve the overall member experience, and provide clear business insights through powerful dashboards and reports.

# Phase 1: Problem Understanding & Industry Analysis

**Goal:** Understand what we're building and why.

### 1. Requirement Gathering

Based on initial discussions with key stakeholders, the following functional requirements have been identified for the platform:

- **Member Experience:**
  - The system must allow members to view available classes, filter by date, time, trainer, and class type.
  - Members must be able to securely book and cancel a spot in a class online.
  - The system must automatically send an email confirmation upon booking and a reminder email 24 hours prior to the class.
  - The system must allow members to track their current loyalty points.
- **Operational Efficiency:**
  - Studio staff and trainers must be able to create, edit, and manage Class and Trainer records.
  - Staff must have a centralized view of all class bookings and member attendance for a given day.
  - The system must track member loyalty points based on class attendance.
- **Management & Reporting:**
  - Management must be able to view dashboards showing key performance indicators (KPIs) such as class attendance rates and total active members.
  - The system must generate reports on class popularity, top-performing trainers, and member loyalty.

### 2. Stakeholder Analysis

Three primary stakeholder groups have been identified, each with unique roles and needs that the solution must address:

- **Members (End-Users):**
  - Role: The customers of the fitness studio.
  - Needs: A seamless, user-friendly experience for booking classes and a transparent view of their engagement and loyalty.
- **Studio Staff/Trainers (Internal Users):**
  - Role: The administrators who will use the system daily to manage operations.
  - Needs: A simple, efficient interface to manage schedules, track attendance, and resolve member issues.

- **Admins:**
    - Configure the Salesforce platform, manage permissions, and ensure smooth operations.
- **Management/Studio Owner (Business Sponsors):**
    - Role: The decision-makers who need to analyze business performance and ROI.
    - Needs: Clear, real-time insights into operational metrics to drive strategic planning and growth.

**3. Business Process Mapping**

A critical step is mapping the current manual processes ("As-Is") and defining the future, automated processes ("To-Be").

- **As-Is Process (Manual):**
    - A member calls the studio to book a class. A staff member manually enters the booking into a spreadsheet. No automated confirmation is sent, and class reminders are often forgotten.
    - Payments and attendance are tracked manually, often leading to errors.
- **To-Be Process (Automated):**
    - A member books a class via an online form (Salesforce Flow). A Class Booking record is automatically created in Salesforce. The system sends an instant email confirmation and a scheduled reminder email the day before the class. The class roster is automatically updated in real-time.
    - Payments and loyalty points are automatically tracked.
    - Managers see real-time data via dashboards.

**4. Industry-specific Use Case Analysis**

The fitness and wellness industry presents unique requirements that a standard CRM must be adapted to handle. Our solution will specifically address:

- Class Capacity Management: The system must prevent overbooking by ensuring that the number of Class Booking records does not exceed the Class capacity.
- Member Loyalty & Retention: The system must include a mechanism to track and update Loyalty Points for members who attend classes, which is crucial for fostering long-term engagement.
- B2C-First Approach: The primary user interface for bookings will be a public-facing portal or form, designed specifically for the customer's convenience, prioritizing an intuitive user experience.

**5. AppExchange Exploration**

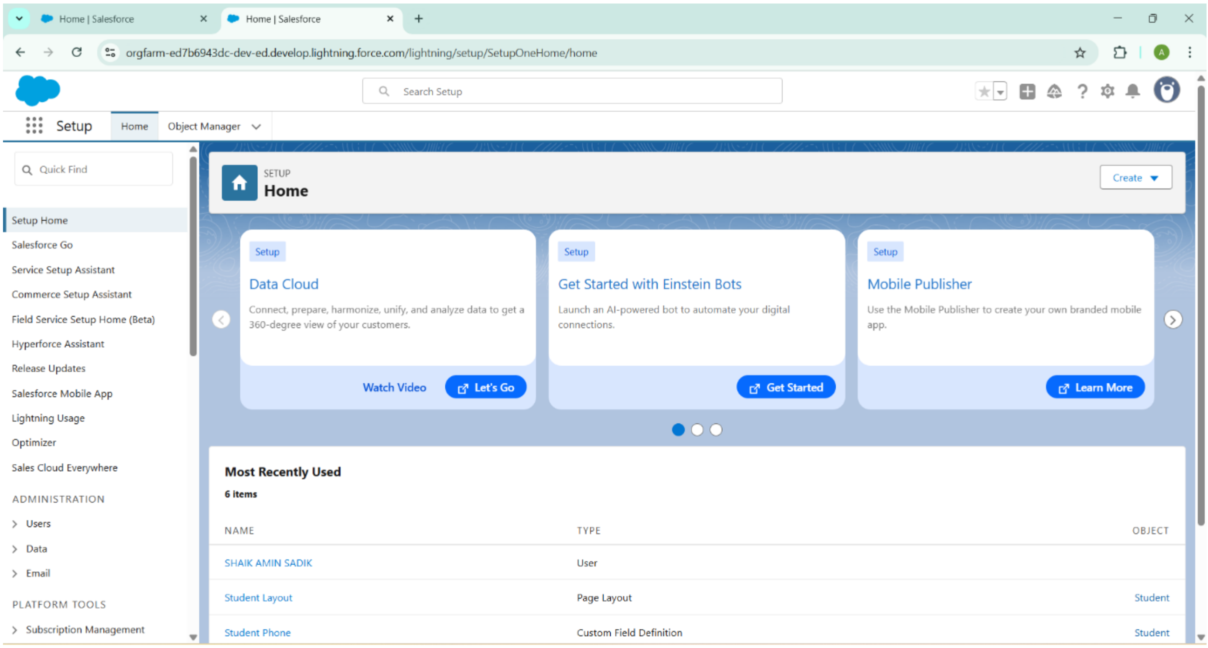
Existing solutions on Salesforce AppExchange (such as scheduling tools and loyalty apps) provide useful reference points. However, this project focuses on building a custom native solution to demonstrate Salesforce platform capabilities including custom objects, flows, automation, and dashboards.

# Phase 2: Org Setup & Configuration

The objective of this phase was to establish the foundational environment and security model for the project. This included configuring the Salesforce organization and setting up the user access framework.

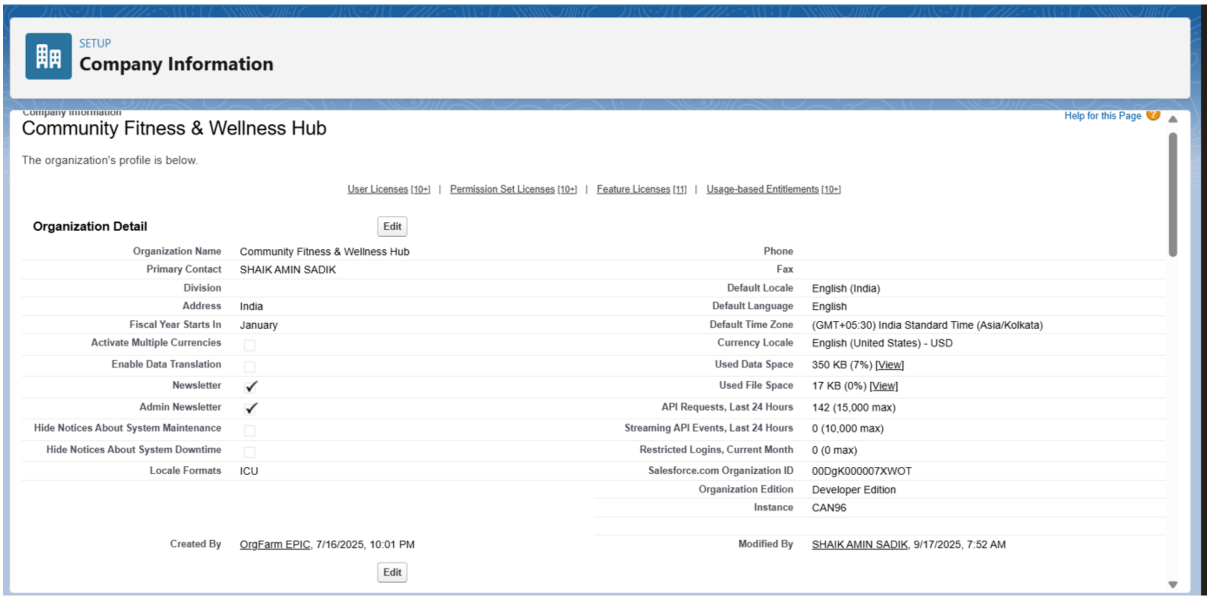
**1.Salesforce Editions:** A free Salesforce Developer Edition was used as the primary build environment, providing access to all necessary tools for development and testing.

- Verified that the **Salesforce Developer Edition** is being used for this project.
- This edition provides full customization capabilities required for building the Community Fitness & Wellness Hub.

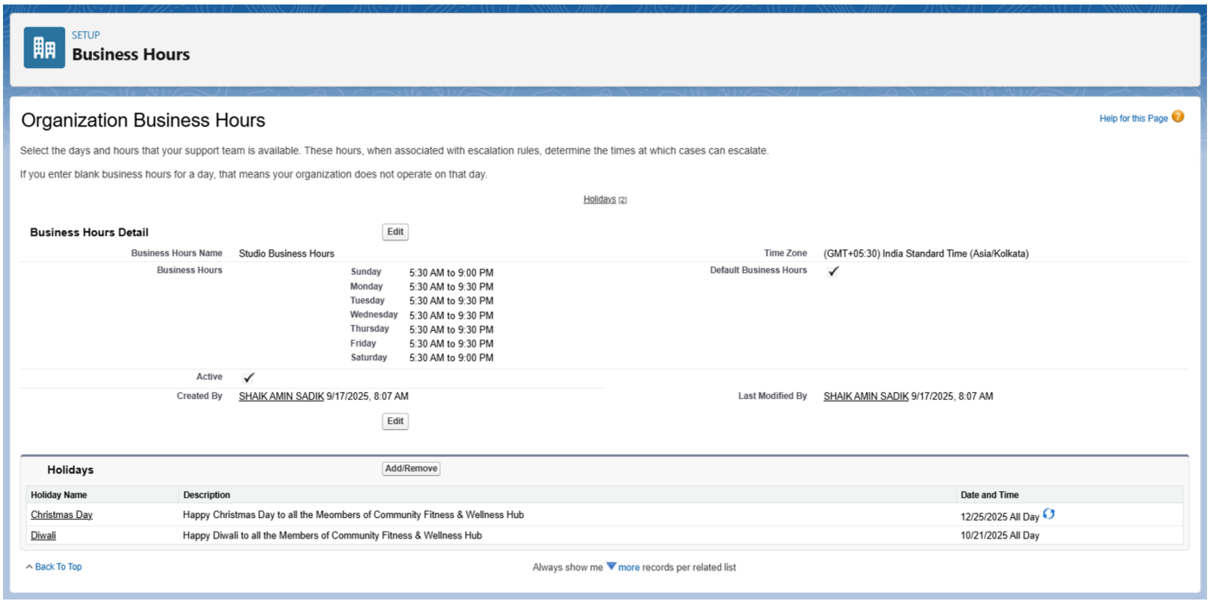



**2.Company Profile Setup:** The organization's profile, including name, contact information, and default time zone, was reviewed and configured to ensure an accurate and professional starting point for the project.

- Verified and updated:
    - Company name
    - Default locale, currency, time zone, and language
    - Primary contact details




**3. Business Hours & Holidays:** Business hours and specific holidays (e.g., Christmas Day and Diwali) were defined. This demonstrates an understanding of how to manage business time, which is crucial for advanced scheduling and future automation.

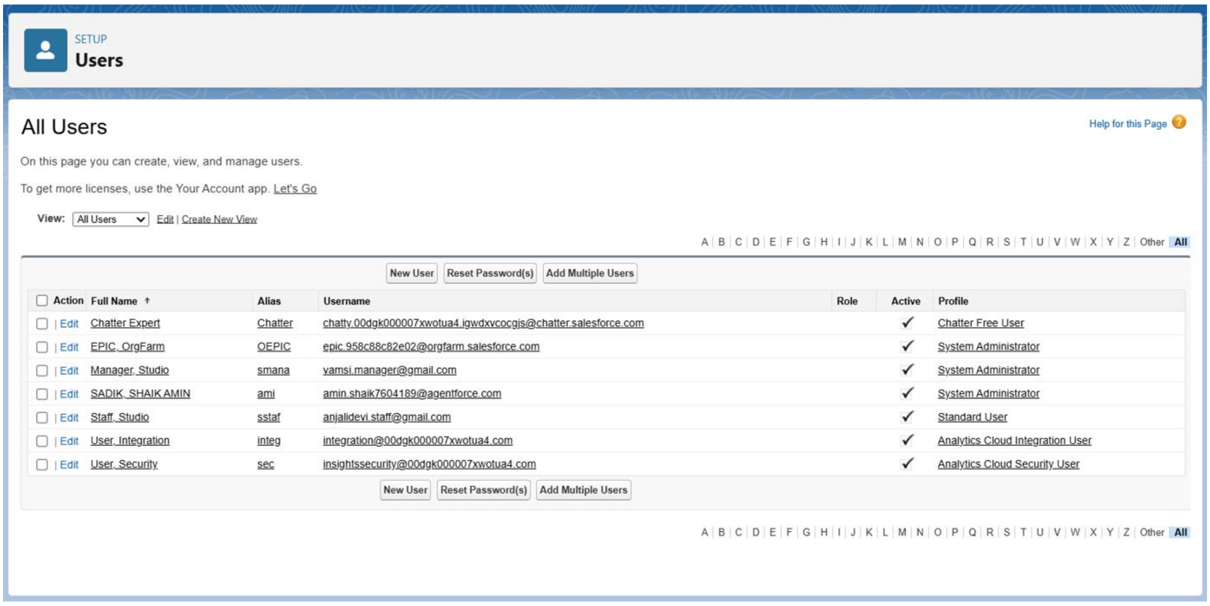



## 4. Fiscal Year Settings

- Kept the standard fiscal year (January–December) as per project requirements.
- No customization required at this stage.

## 5.User Setup & Licenses

- User accounts for the "Studio Staff" and "Studio Manager" were created with standard Salesforce licenses. This established the foundational users for the internal application.
- Assigned available **Salesforce licenses.**

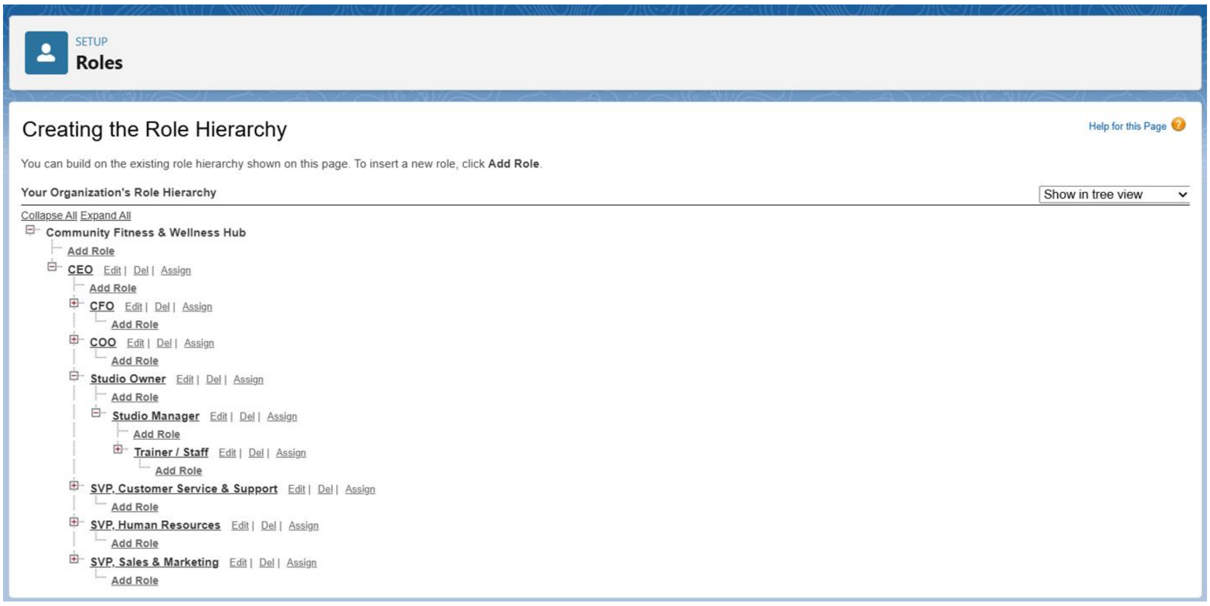



## 6.Profiles

- Standard profiles were utilized as the baseline for user permissions, aligning with modern best practices to manage granular access through Permission Sets.

## 7.Roles

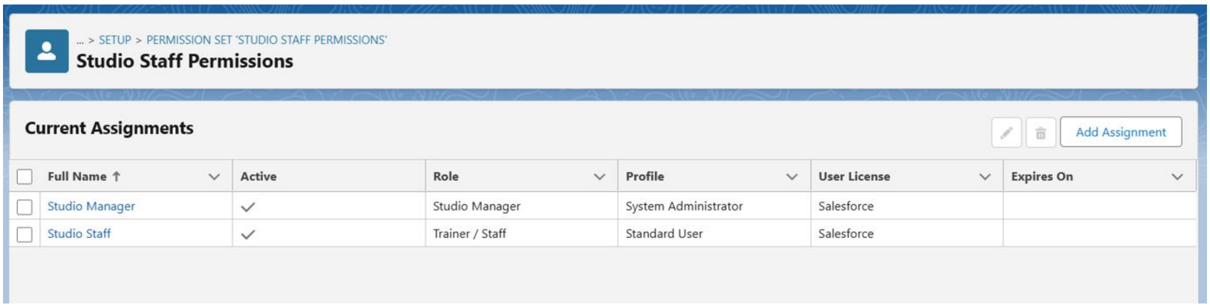
- A clear role hierarchy was defined, with a "Studio Owner" role above a "Trainer / Staff" role, ensuring proper data visibility and reporting.
- Role assignments for custom records will be finalized in **Phase 3**.




## 8.Permission Sets

- A custom permission set was created to manage granular permissions for the project's custom objects. This modern approach ensures a flexible and scalable security model.
- Created a basic **Permission Set** for granting extra access without changing profiles.

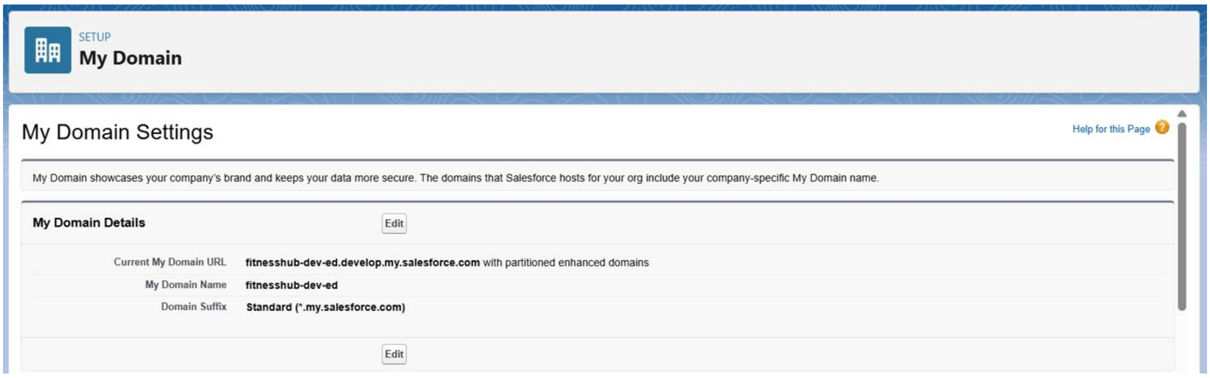
- Specific permissions (like booking classes, loyalty tracking) will be implemented in **Phase 3** once custom objects exist.




**9.OWD (Organization-Wide Defaults):** This was not implemented in Phase 2. **OWD will be configured in Phase 3** after custom objects have been created. This is a critical step to ensure data privacy by setting the baseline access to private.

**10.My Domain**

- A unique 'My Domain' was registered and deployed, which is a prerequisite for many advanced Lightning features and branding.

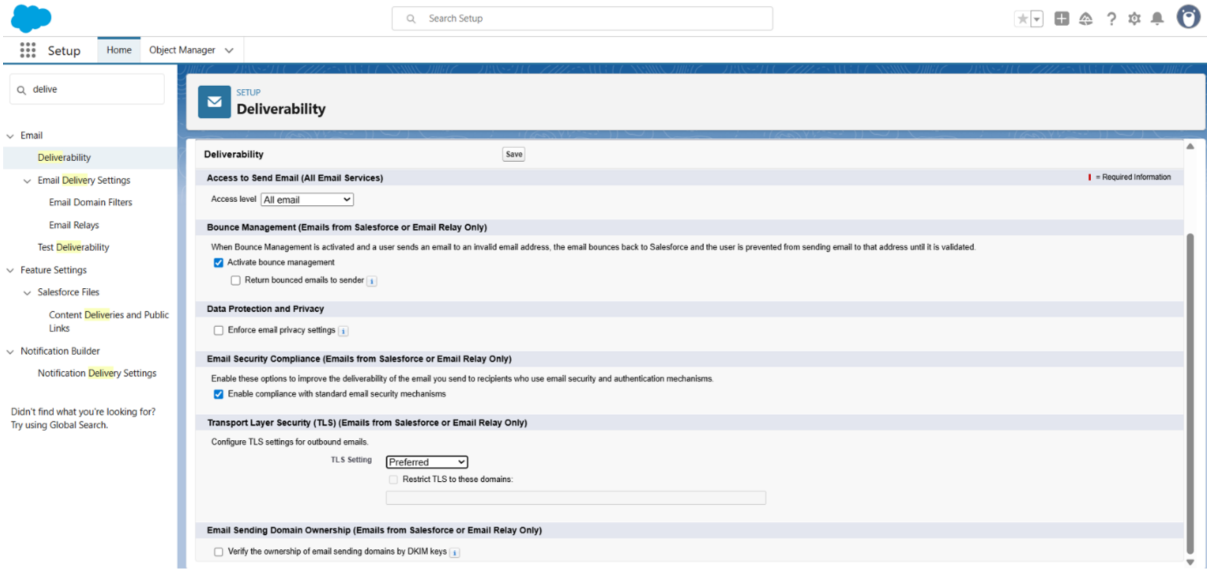



**11.Sharing Rules:** This was not implemented in Phase 2. **Sharing rules will be implemented in Phase 3** to selectively open up data that is restricted by the OWD settings. For example, a sharing rule will be created to allow a trainer to see only the bookings for their classes.

**12.Login Access Policies:** This was not implemented. Login and password policies were reviewed, but left at their defaults, as they are not necessary for a single-user development environment.

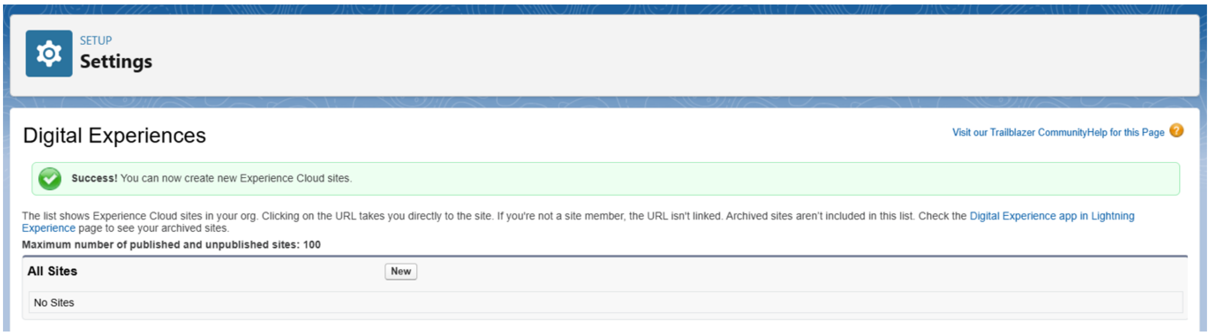
**13.Email Configuration**

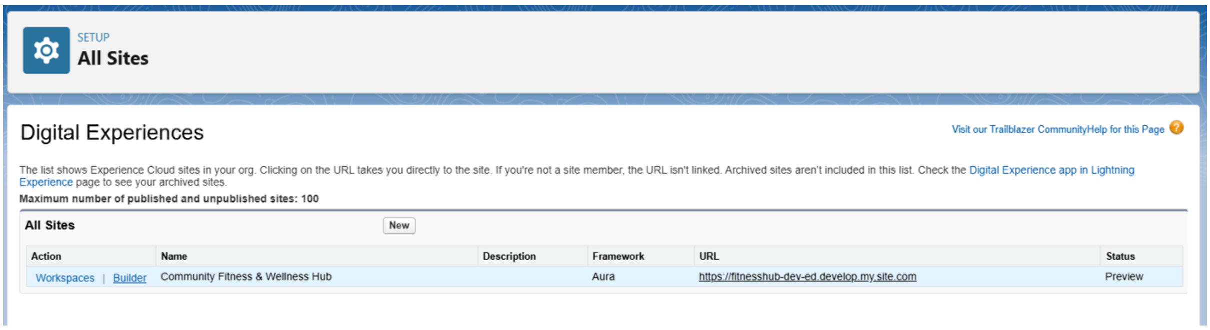
- The email deliverability settings were set to "All email," and an organization-wide email address was configured and verified. This ensures all automated system communications are professional and reliable.




## 14. Experience Cloud

- Experience Cloud was enabled to explore the possibility of building a public-facing member portal. A basic site was created to demonstrate an understanding of this technology.




## 15. Sandbox Usage

- This was not implemented as a separate sandbox is not available in a Developer Edition. The documentation will explain that in a real-world scenario, a developer sandbox would be used for building and testing before deploying to production.

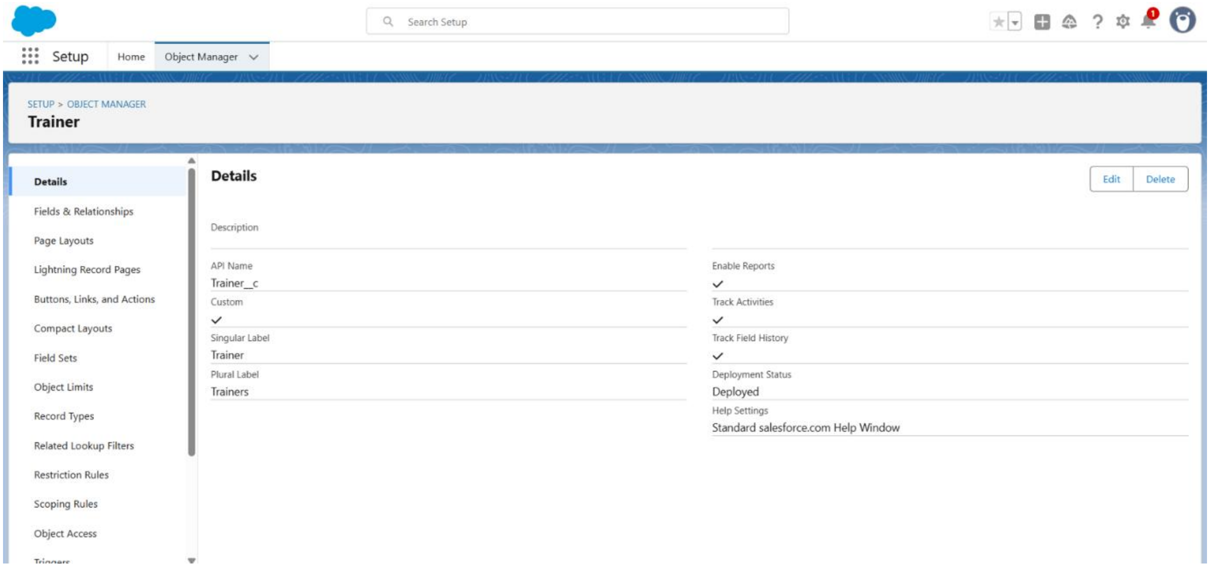
### 16.Dev Org Setup & Deployment Basics

- Confirmed that all development will be done in a **Developer Org** environment.
- This was not implemented in this phase. The **deployment strategy will be discussed and implemented in Phase 8**, where the project will be prepared for a real-world rollout.

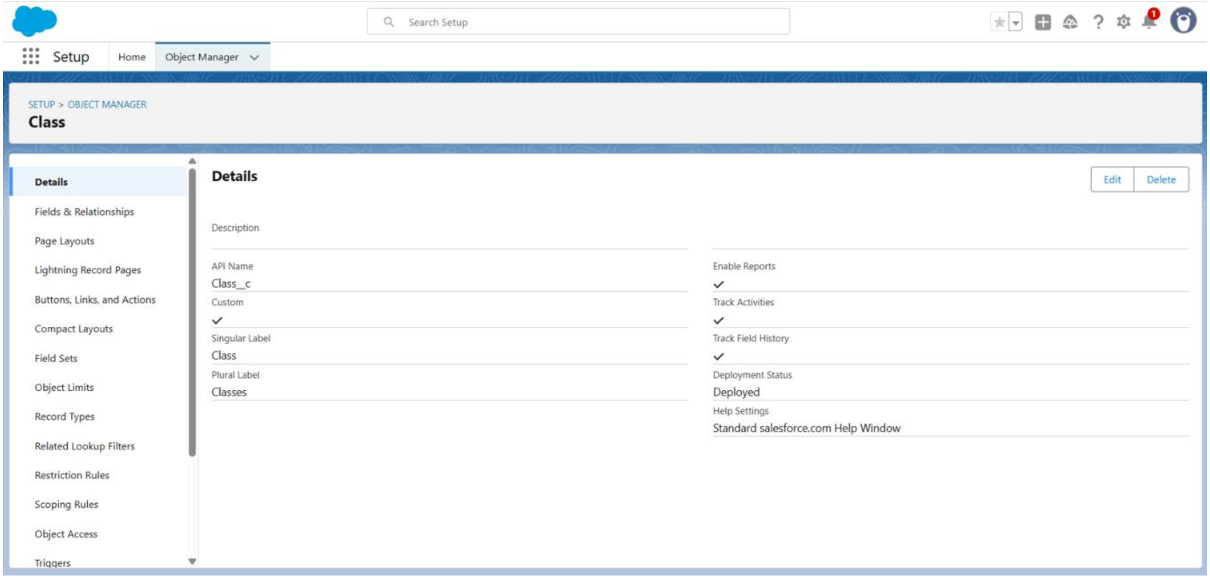
# Phase 3: Data Modeling & Relationships

### 1.Standard & Custom Objects

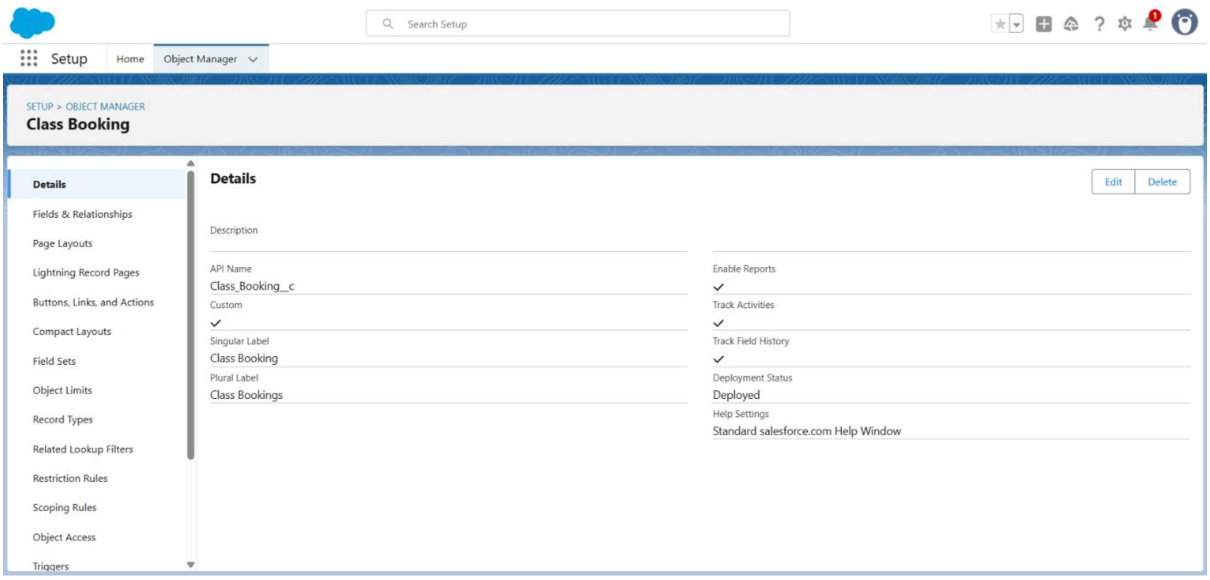
The project utilized the standard Contact object to represent members. Five custom objects were created to support the core functionality: Trainer, Class, Class Booking, Payments, and Loyalty Points.



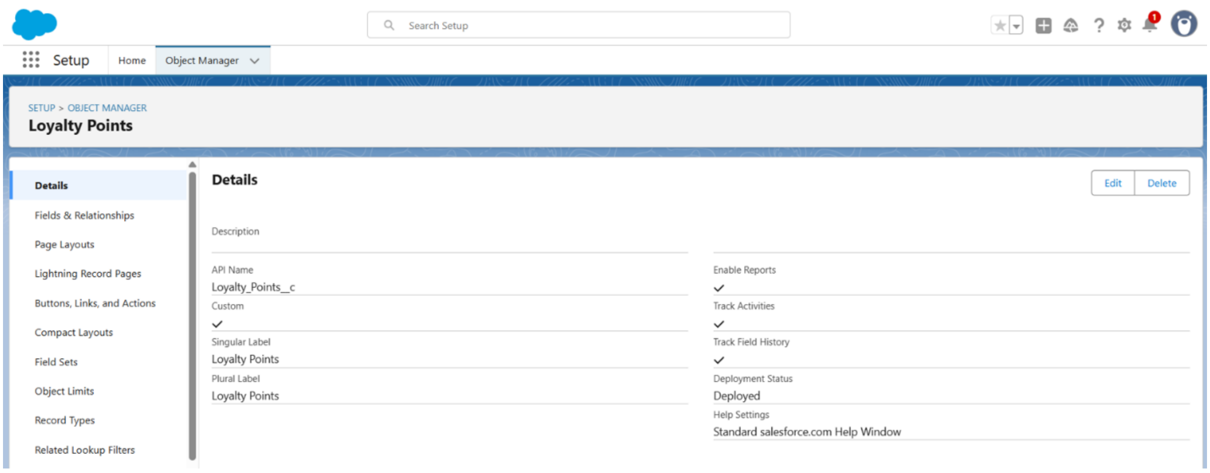

The **Trainer** object was created to store all information about the studio's trainers.



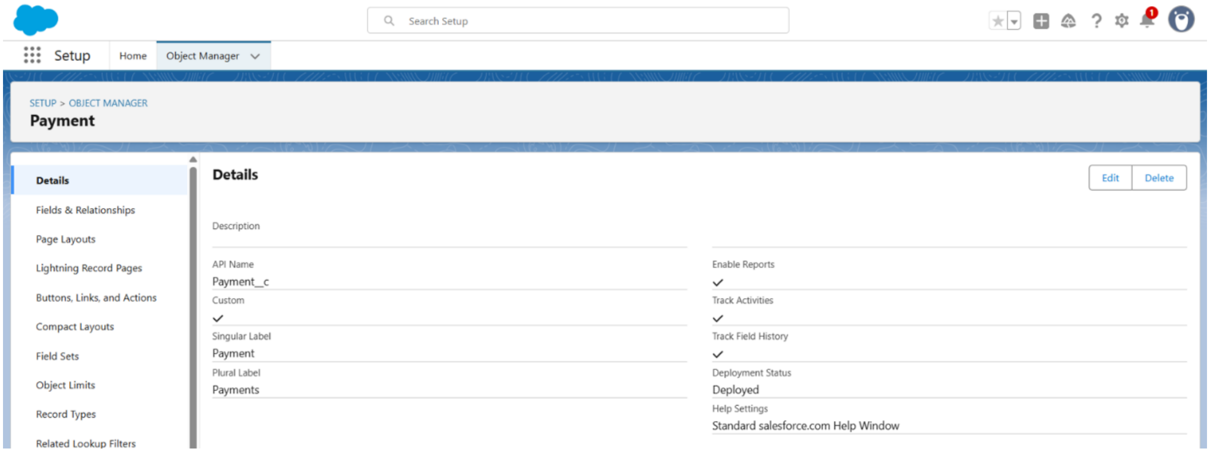

The **Class** custom object was created to manage all classes offered by the studio.




The **Class Booking** custom object was created to track each member's booking.




The **Loyalty Points** custom object was created to manage the member engagement program.



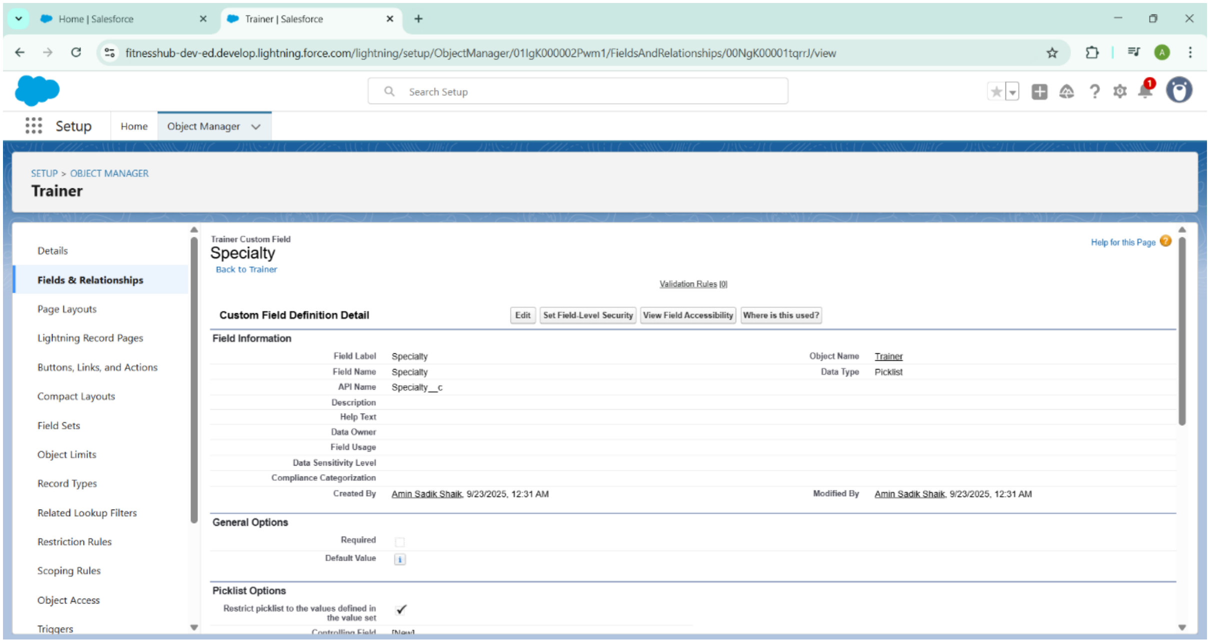

The **Payments** custom object was created to manage all payment records.

## 2.Record Types

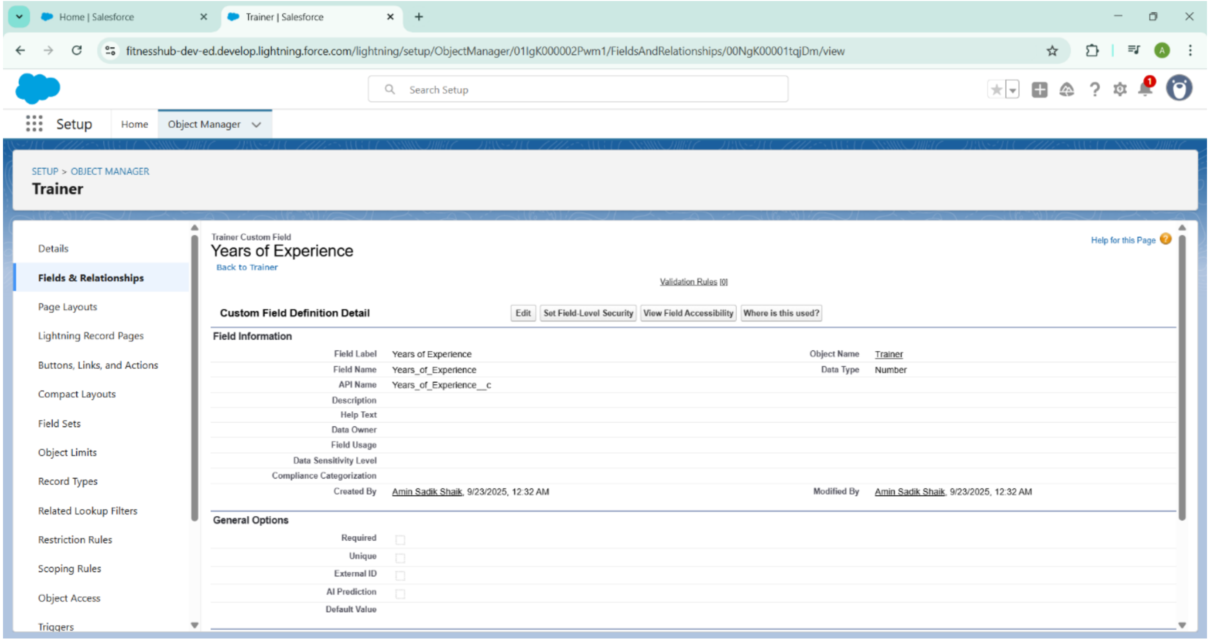
Record types were not implemented in this phase. The project utilized the default record type to maintain a simple, focused, and efficient data model.

## 3.Fields

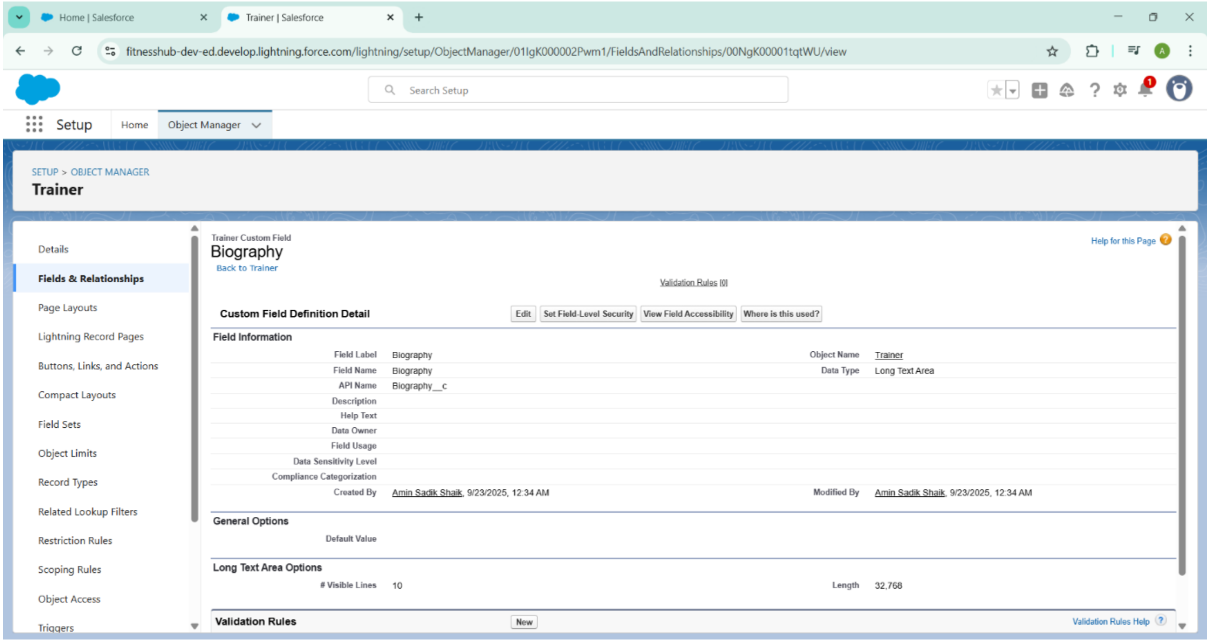
Custom fields were created on each object to capture project-specific data. Examples include a Start Date/Time field on the Class object, a Status picklist on the Class Booking object, and a Points Earned number field on the Loyalty Points object.



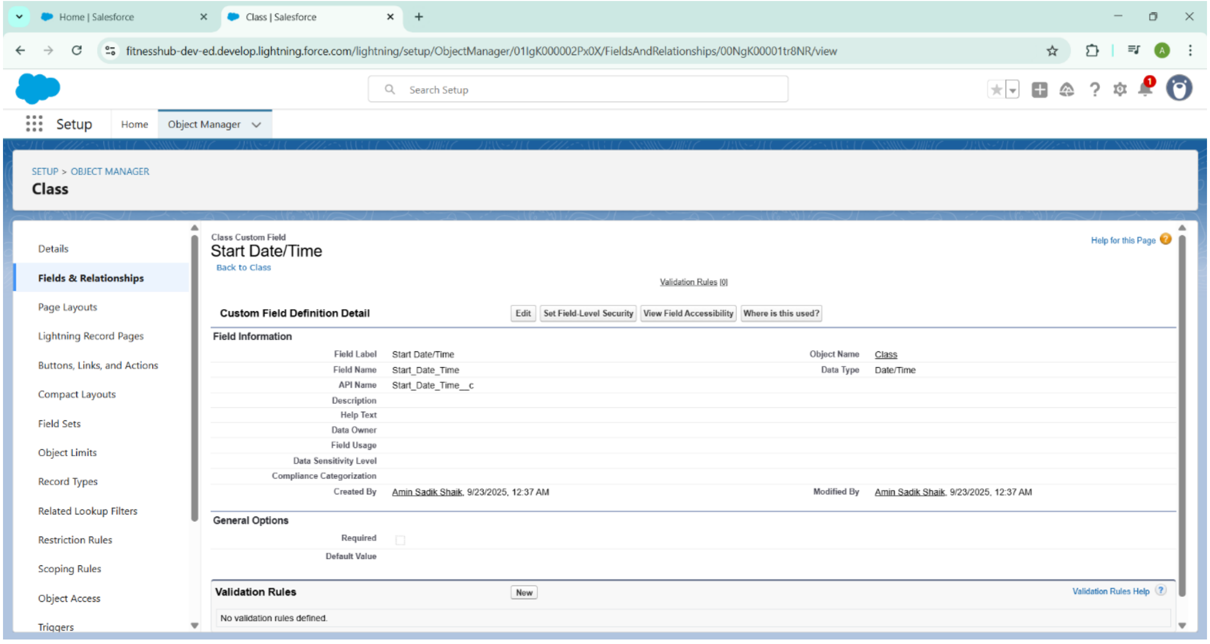

A **Picklist field** named 'Specialty' was created on the Trainer object.



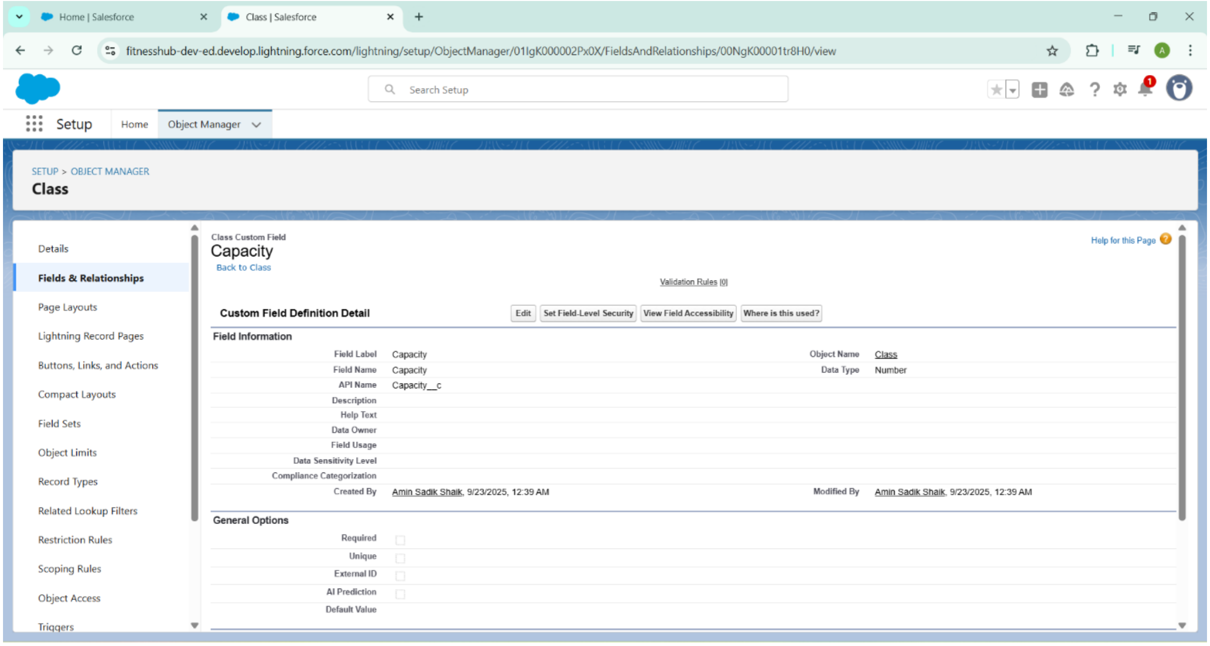
A **Number field** for 'Years of Experience' was created on the Trainer object.



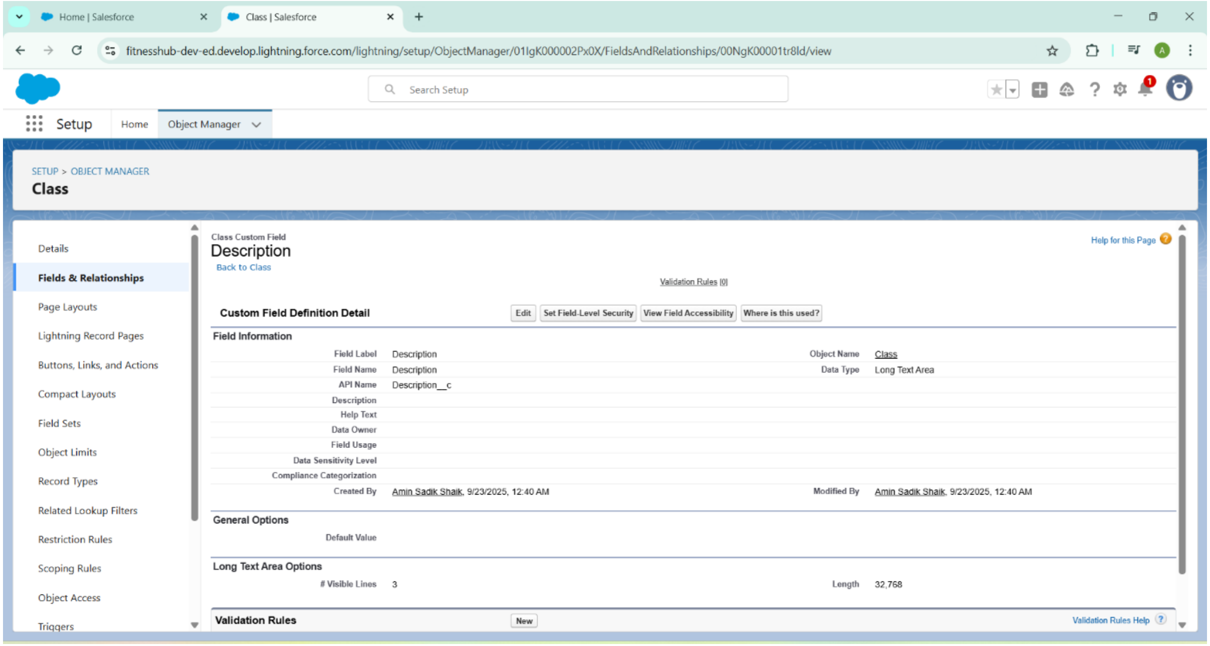
A **Long Text Area field** for 'Biography' was created on the Trainer object.



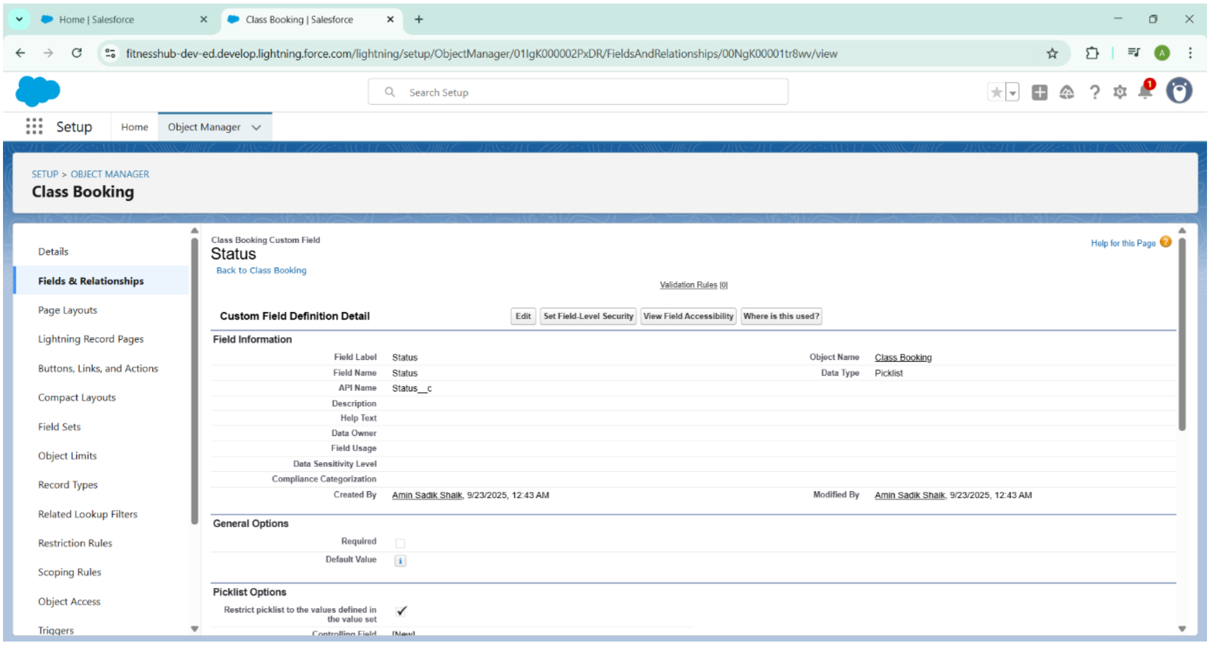
A **Date/Time field** named 'Start Date/Time' was created on the Class object.



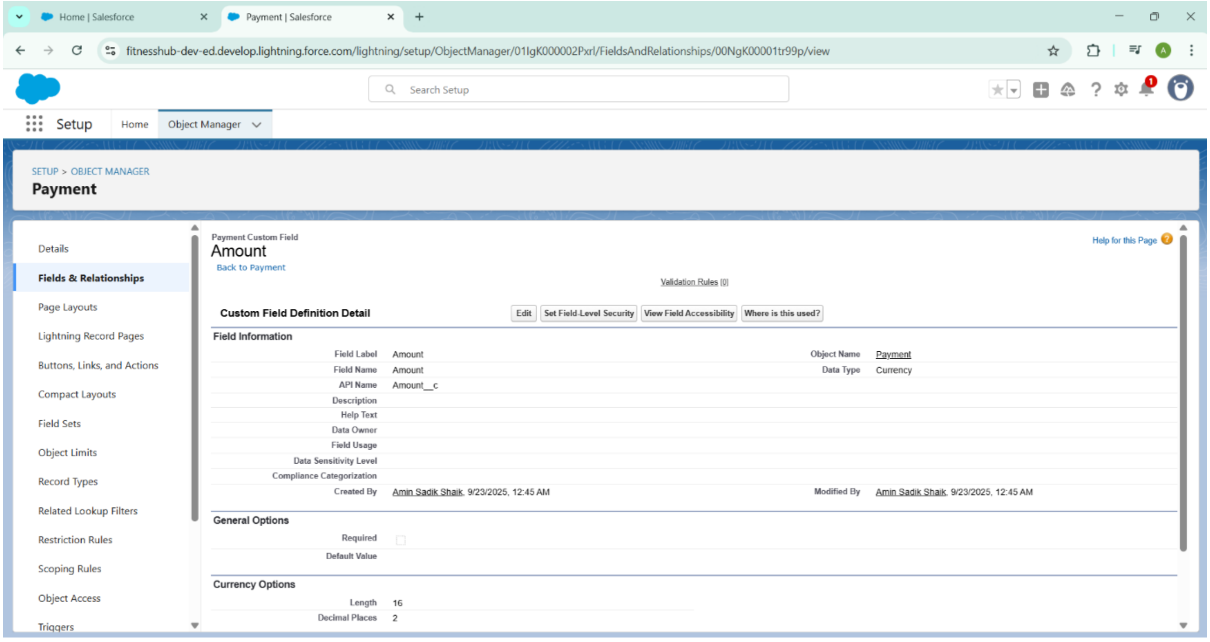
A **Number field** named 'Capacity' was created on the Class object.



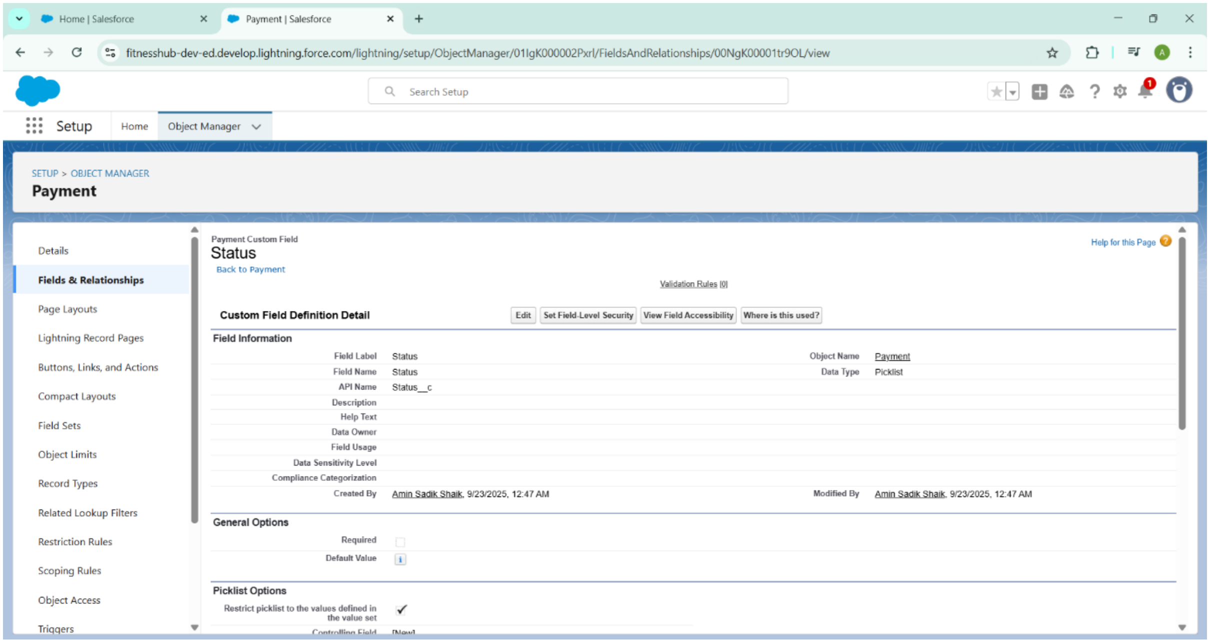
A **Long Text Area field** for 'Description' was created on the Class object.



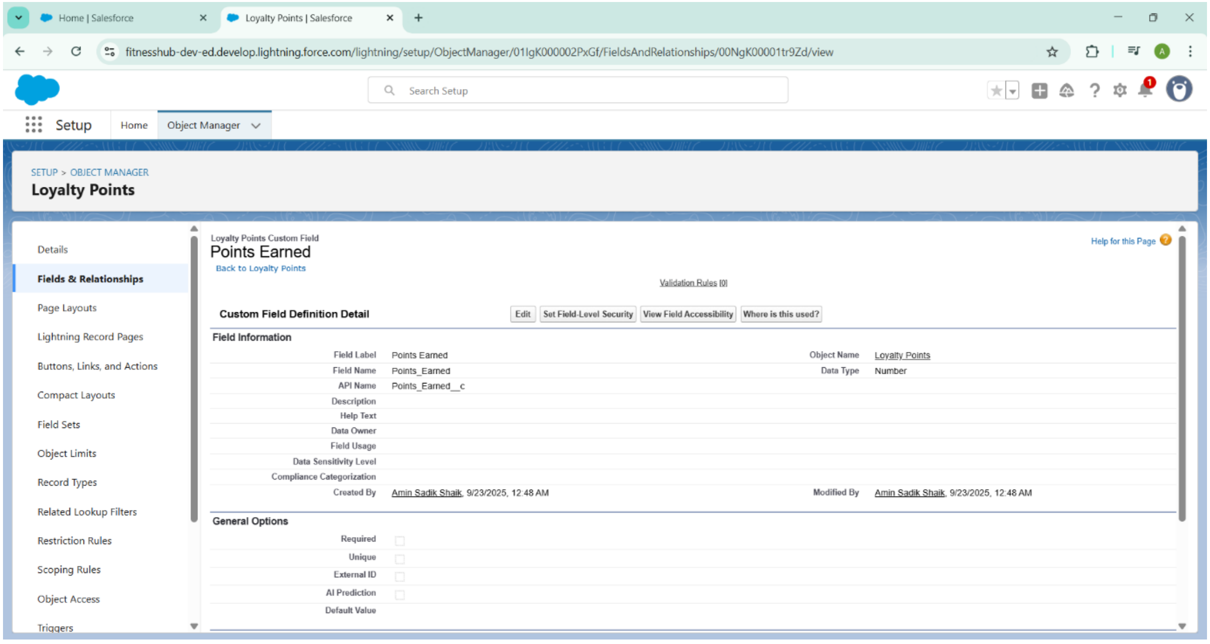
A **Picklist field** named 'Status' was created on the Class Booking object.



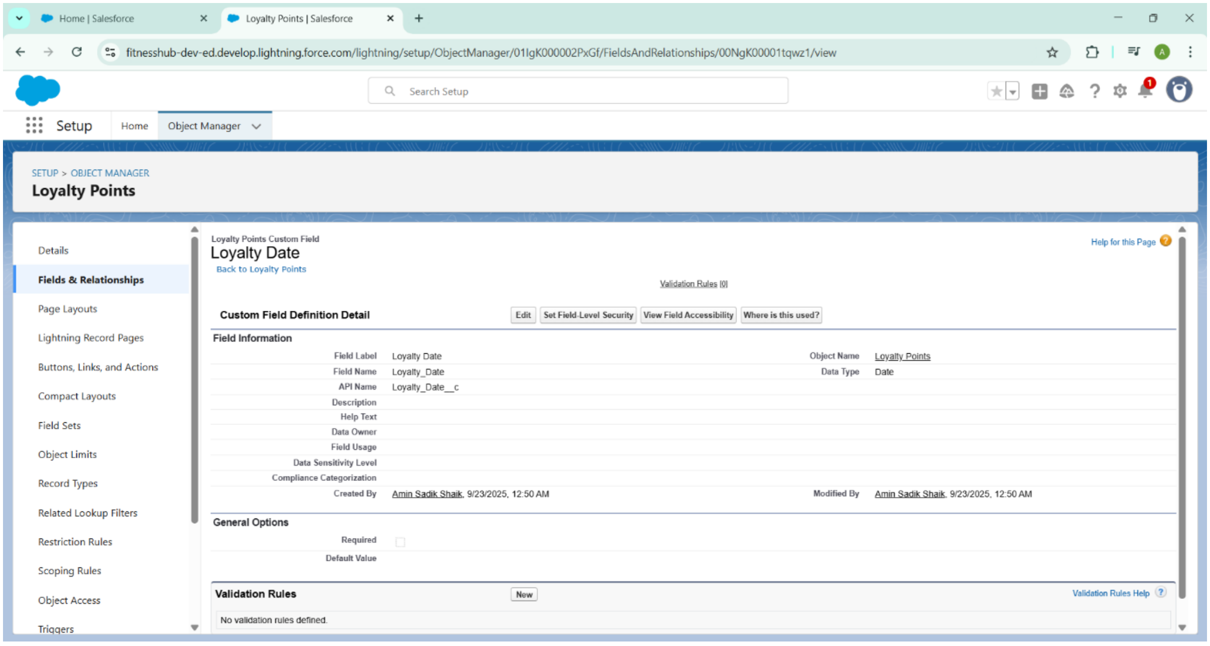
A **Currency field** named 'Amount' was created on the Payments object.



A **Picklist field** named 'Status' was created on the Payments object.




A **Number field** named 'Points Earned' was created on the Loyalty Points object.




A **Date field** named 'Loyalty Date' was created on the Loyalty Points object.

### 4.Compact Layouts

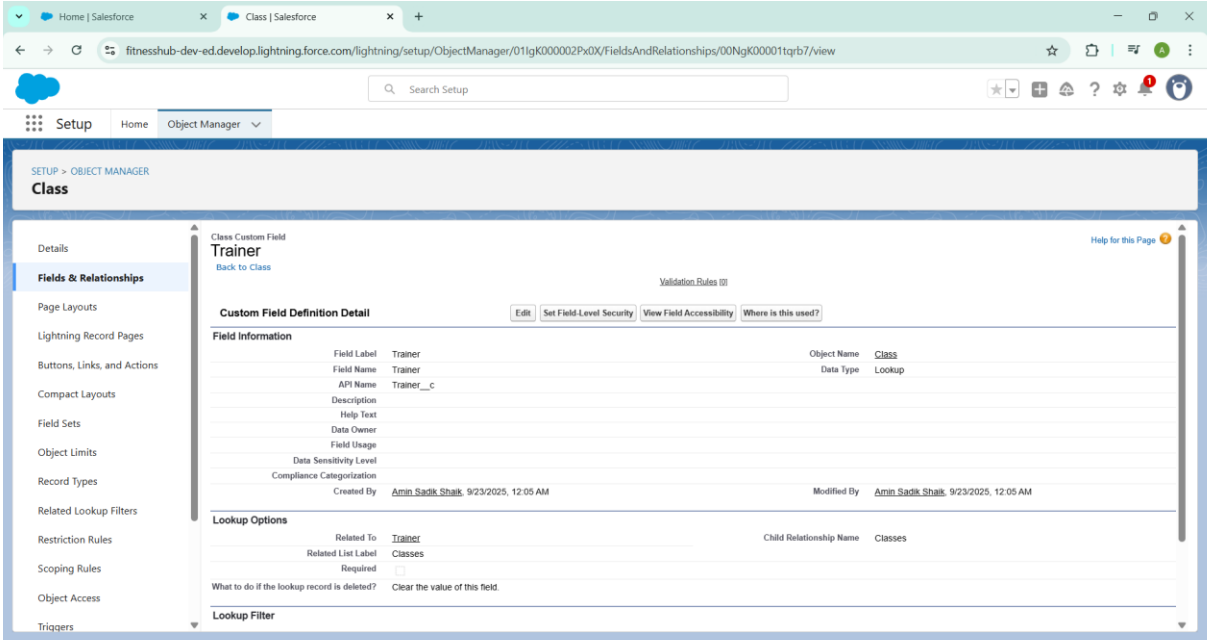
Custom compact layouts were configured for each object. These layouts display the most critical information in the highlights panel at the top of a record, allowing for quick and efficient data consumption.

### 5.Junction Objects

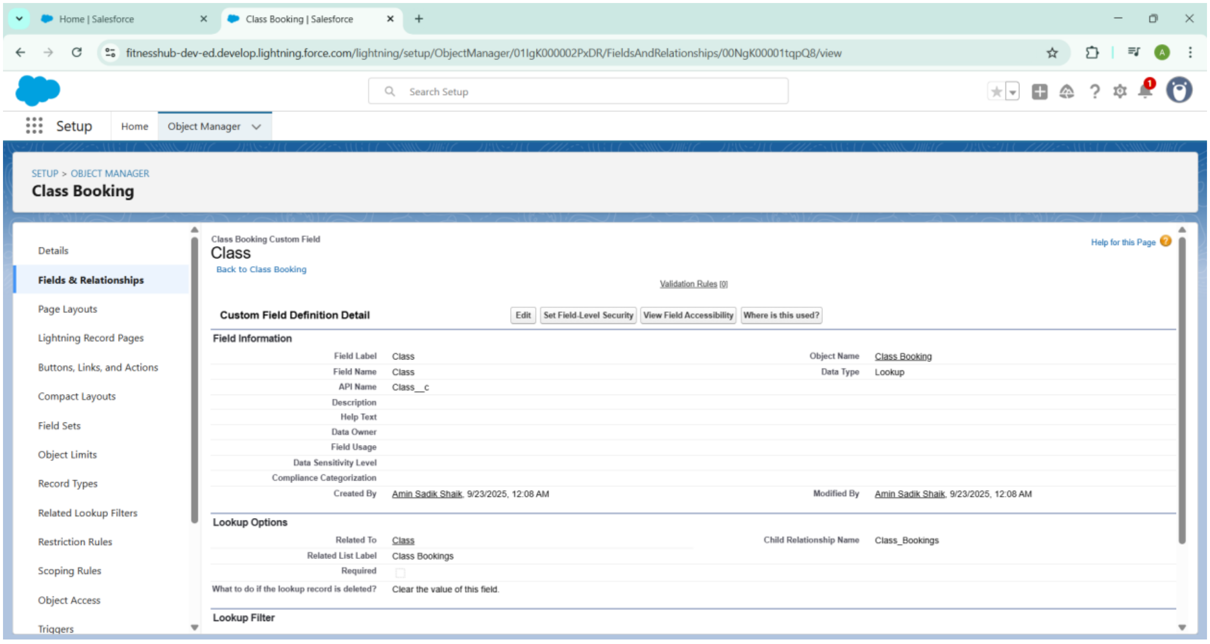
The Class Booking custom object was designed to function as a **junction object**. It connects the Class object and the Contact object, allowing a member to be associated with multiple classes, and a class to have multiple members.

### 6.Lookup vs Master-Detail vs Hierarchical Relationships

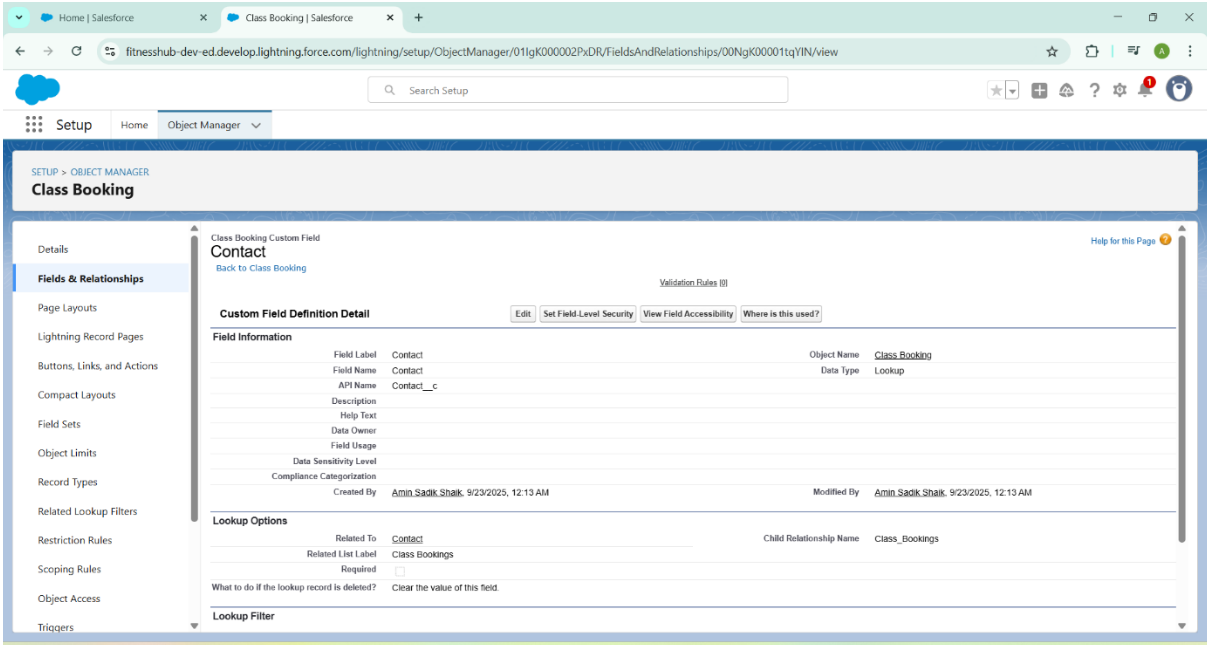
- Lookup relationships were used to connect all custom objects. This was a strategic choice to ensure a flexible data model where related records can be linked without strict ownership requirements, preventing cascading deletions and supporting multiple business processes.



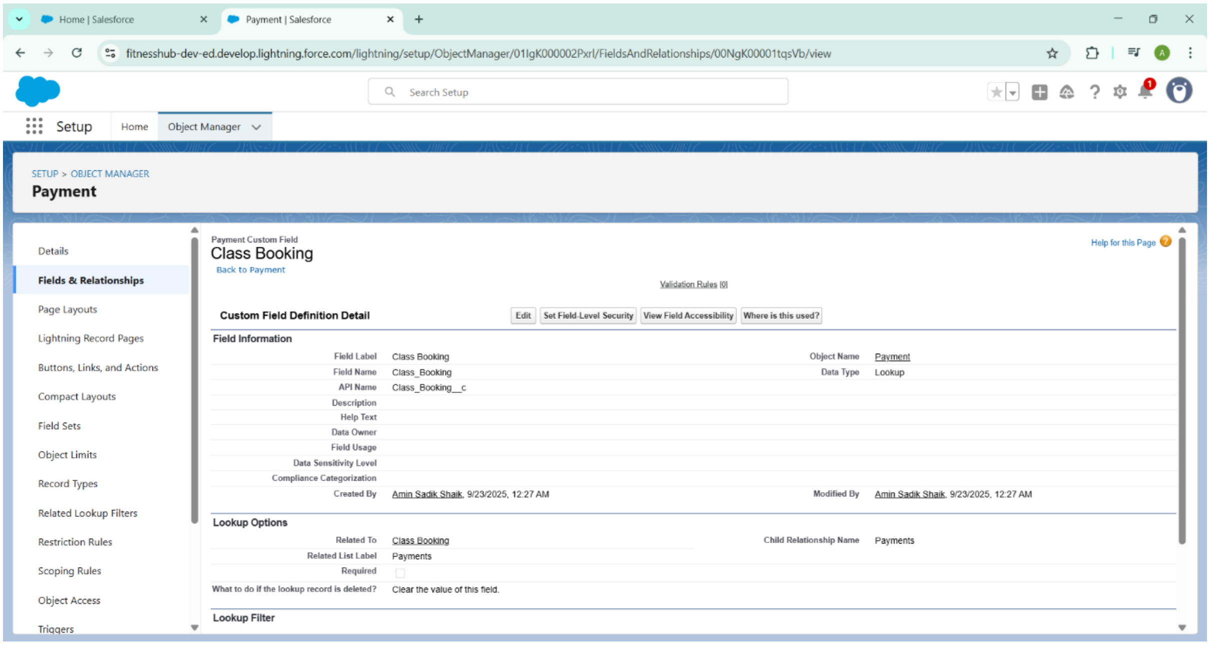

A **Lookup relationship** was created on the Class object to link it to a specific Trainer.




A **Lookup relationship** was created on the Class Booking object to link each booking to its parent Class.



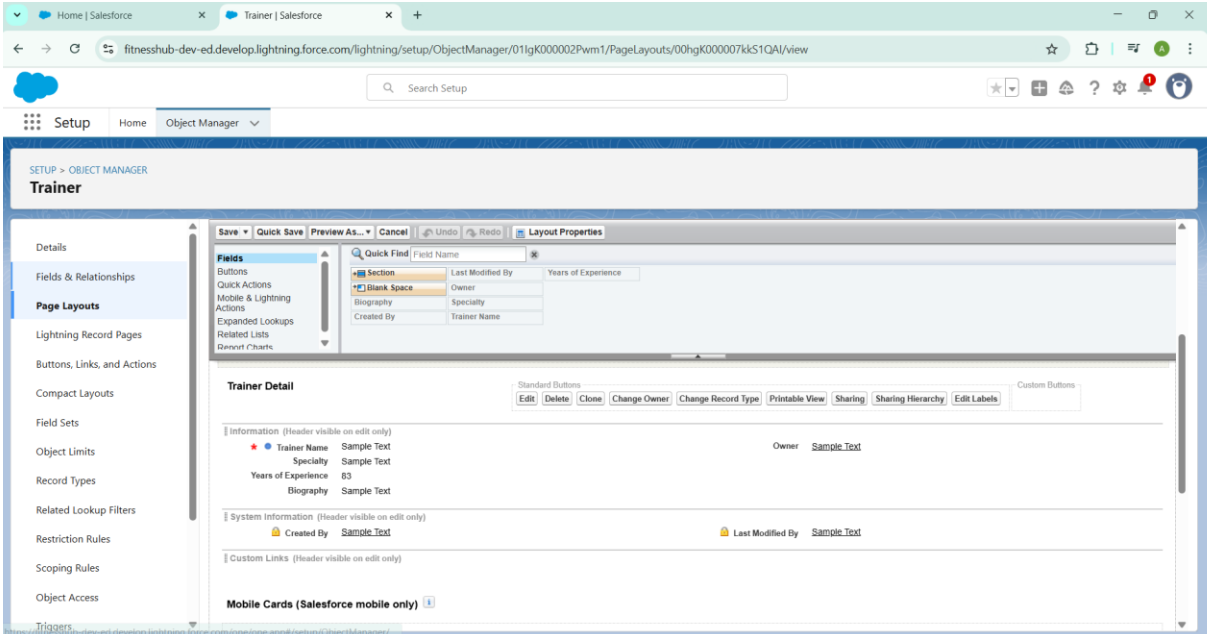

A **Lookup relationship** was created on the Class Booking object to link a booking to the member's Contact record.




A **Lookup relationship** was created on the Payments object to link each payment record to its corresponding Class Booking.

## 7.Page Layouts

- Custom page layouts were created for each object to ensure a clean and organized user interface. Fields were arranged to optimize data entry and viewing, and relevant related lists were added to provide a complete view of connected records.

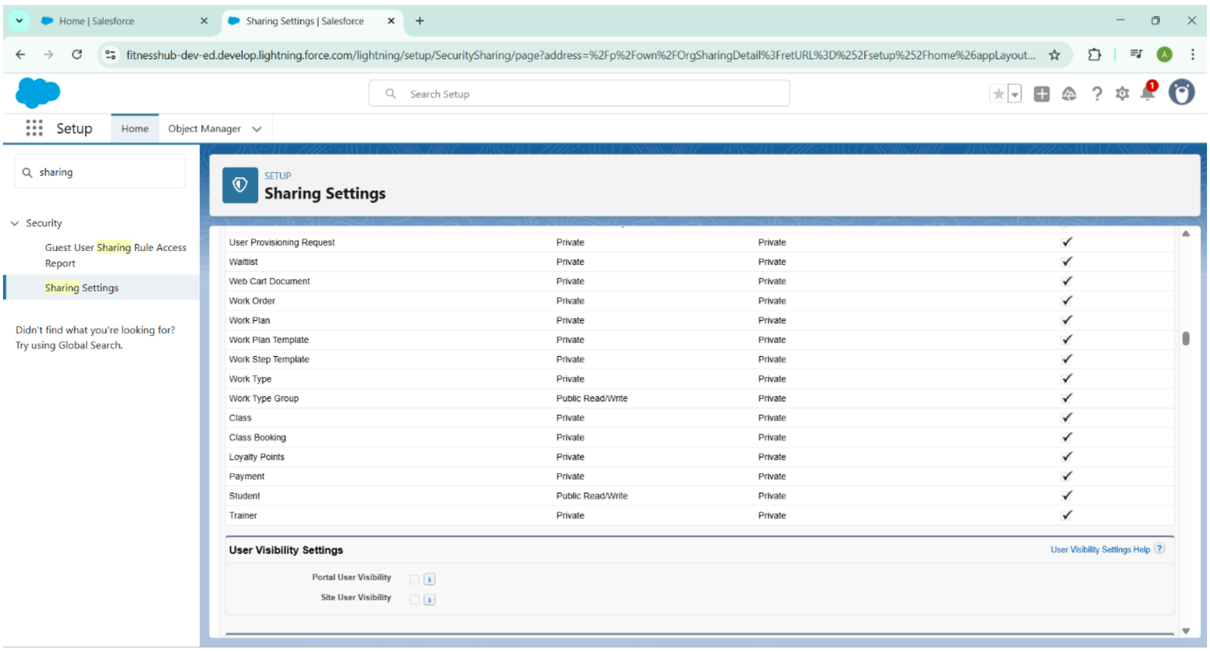



A custom page layout for the Trainer object was configured, displaying all relevant fields for a trainer's information.

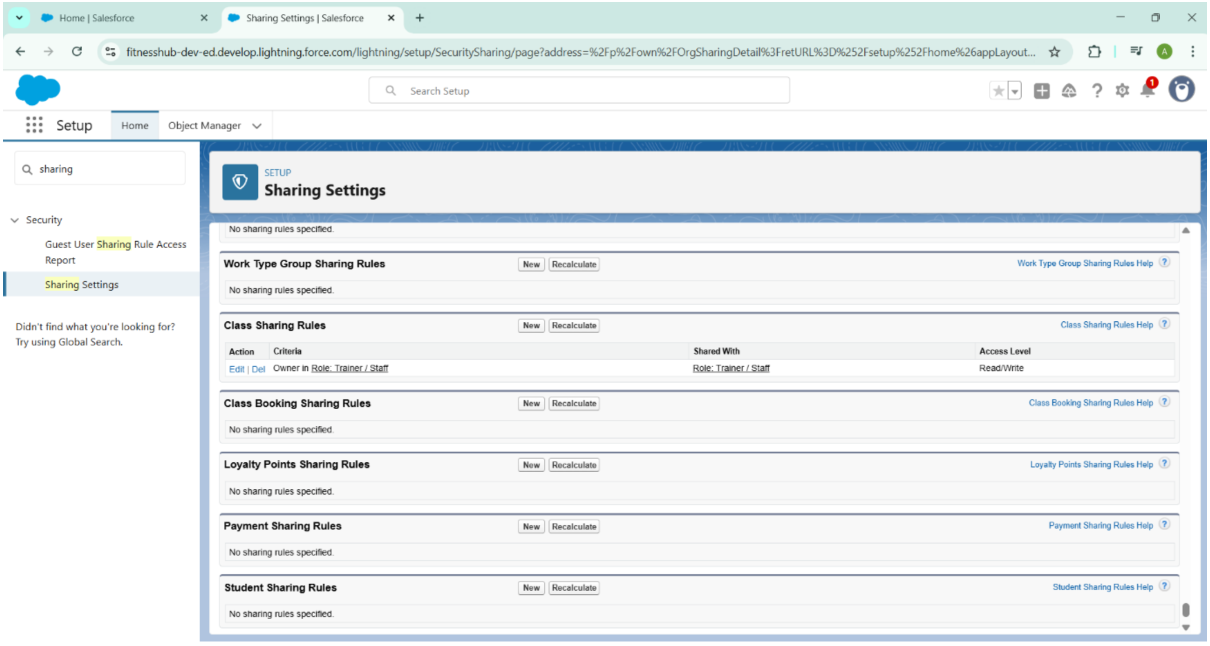
Similar Way , A custom page layout for the Class object was configured, organizing all class details. A custom page layout for the Class Booking object was configured, showing the status and related member/class records. A custom page layout for the Payment object was configured, displaying the payment amount and status. A custom page layout for the Loyalty Points object was configured, showing the earned points and the date are Created.

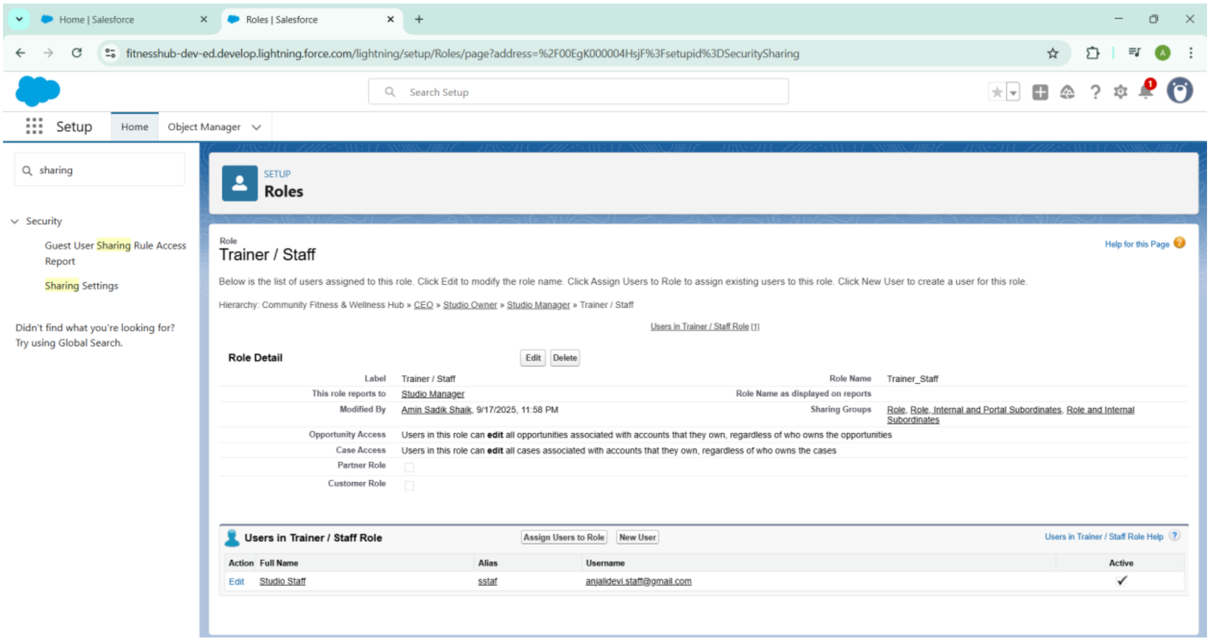
## 8.Schema Builder

The Schema Builder was used to visually confirm the data model. This tool provided a clear diagram of all custom objects and their relationships, ensuring the data structure was correct before proceeding with automation.



The Organization-Wide Defaults (OWD) were set to **Private** for all custom objects, ensuring data privacy by restricting records to their owners.





A **Sharing Rule** was created for the Class object to share records with the 'Trainer / Staff' role, allowing trainers to see all classes they are responsible for.

## 9.External Objects

External objects were not implemented in this project as they were out of scope. The project focused on building a robust data model entirely within the Salesforce platform.

# Phase 4: Process Automation (Admin)

In Phase 4, we will use Flow Builder to automate the core business processes of the fitness hub. We will build three key automations:

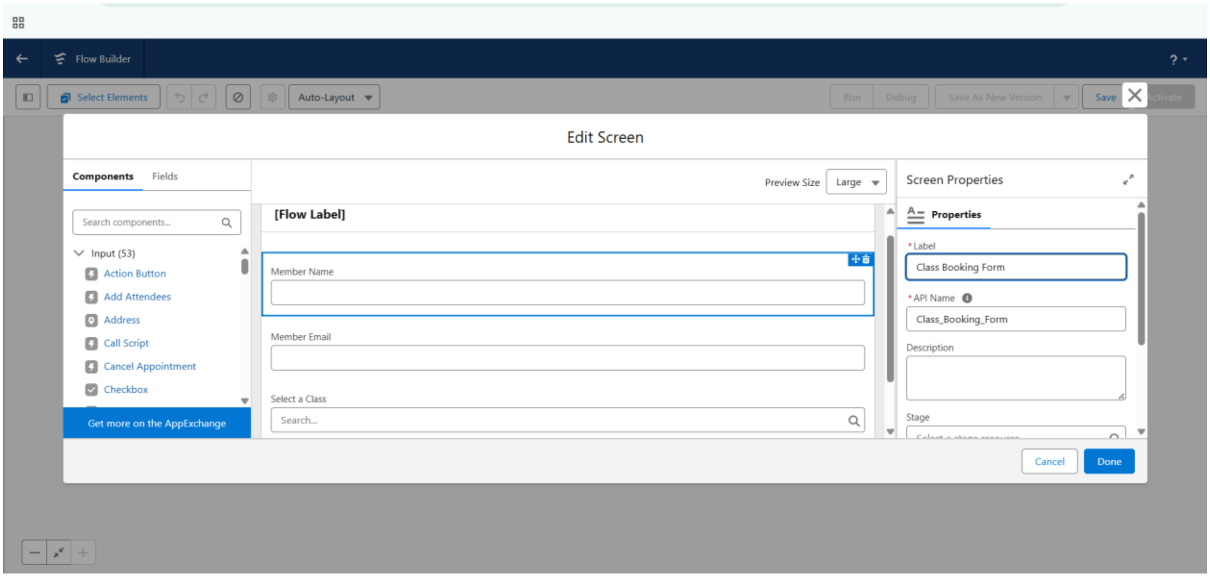
1. **Automated Class Booking:** A public form that members can use to book classes.
2. **Automated Communication:** Confirmation and reminder emails for bookings.
3. **Automated Loyalty Program:** Rewarding members when they attend a class.

We will focus on Flow Builder as it is the most powerful and modern automation tool in Salesforce.

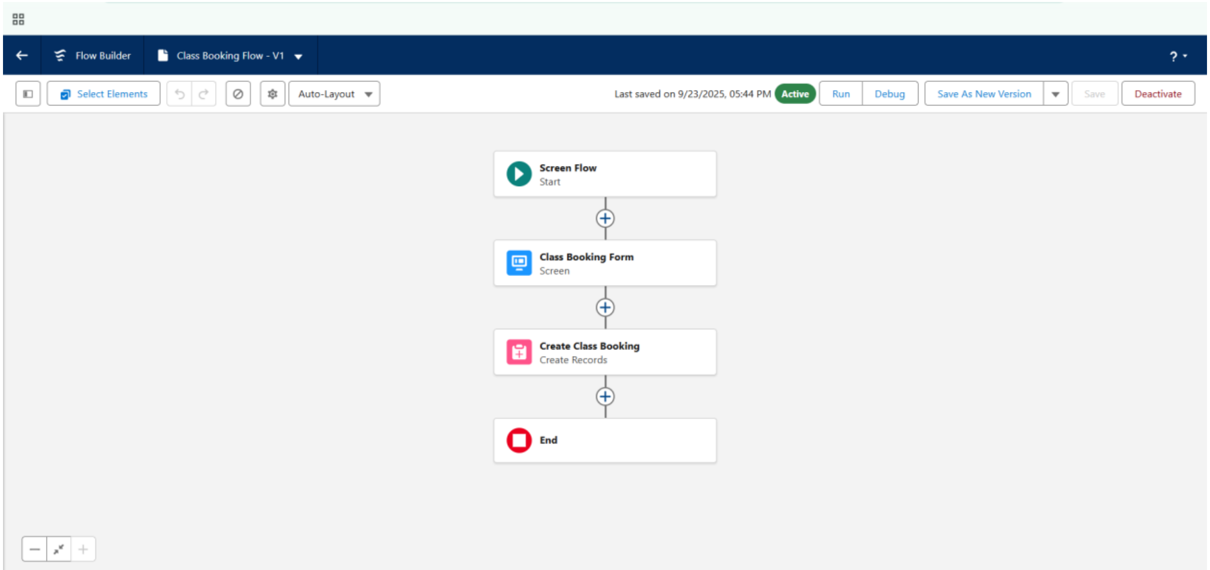
The objective of this phase was to automate key business processes to improve efficiency and enhance the member experience. All automations were built using **Flow Builder**, Salesforce's most powerful and modern declarative tool, replacing older tools like Workflow Rules and Process Builder.

### 1. Member Class Booking Flow (Flow Builder)

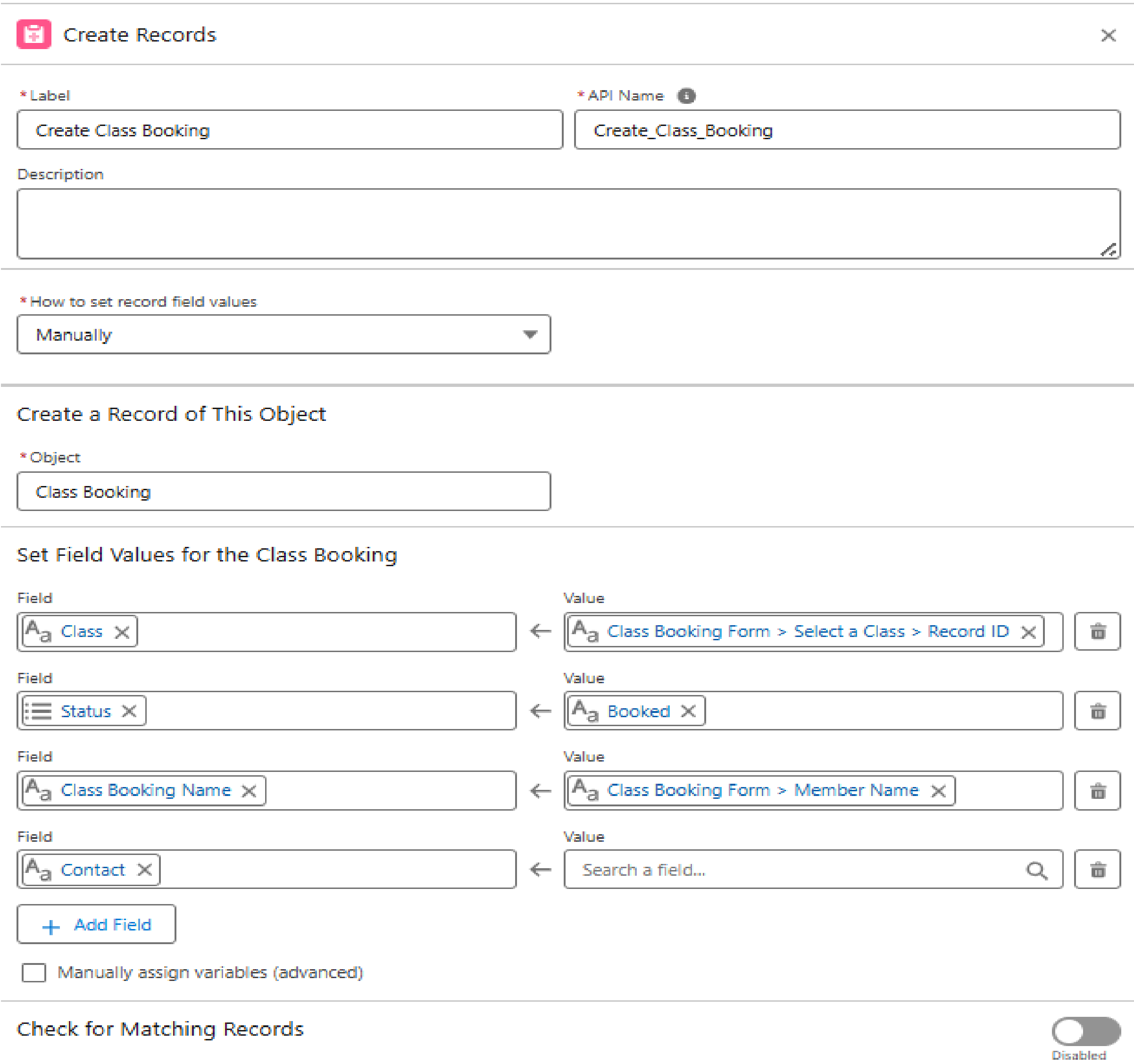
A public-facing screen flow was created to allow members to book classes directly. This automation streamlines the registration process and ensures all booking data is captured accurately in Salesforce.




A screen was designed in Flow Builder to serve as a public-facing booking form, capturing member details and class selection.




The complete Screen Flow canvas, showing the form, the record creation element, and the logical flow of the automation.



The Create Records element was configured to create a new Class Booking record, using the inputs from the screen. This is the core of the booking automation.

**2.Validation Rules:**

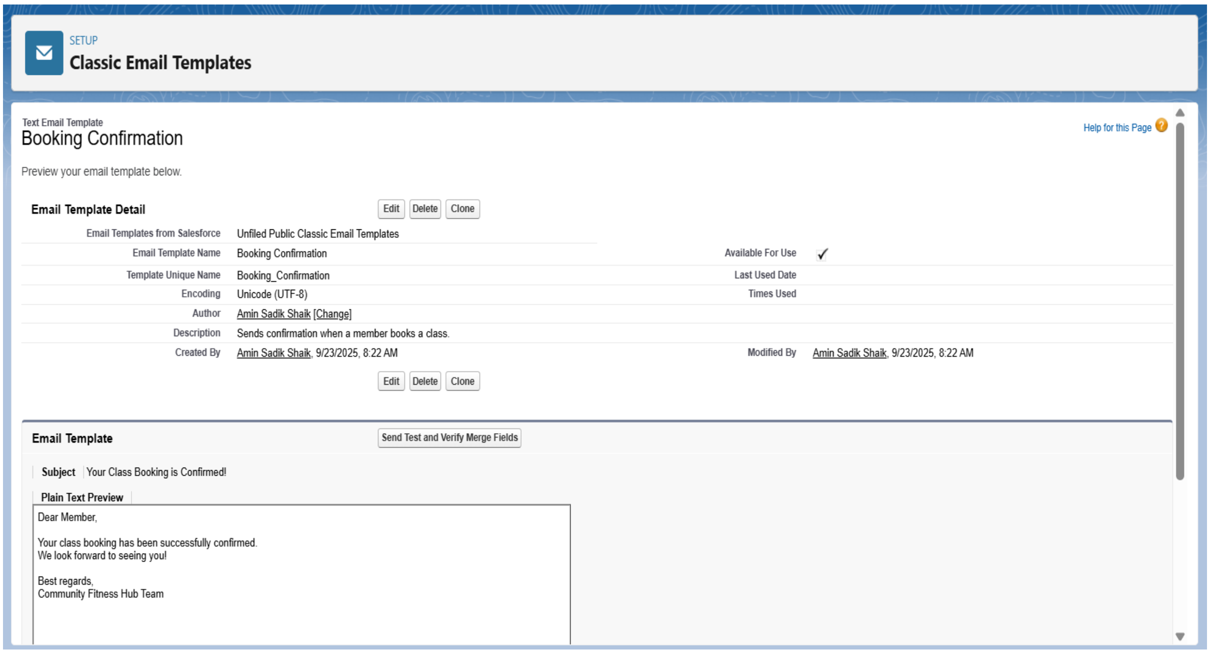
- **Status:** Not Implemented
- **Reason:** Validation rules are used to enforce data integrity (e.g., ensuring a class capacity is a positive number). While important, they were considered out of scope for the core demo to maintain a focus on the main business process. This would be a recommended future enhancement.

**3.Workflow Rules & 4.Process Builder:**

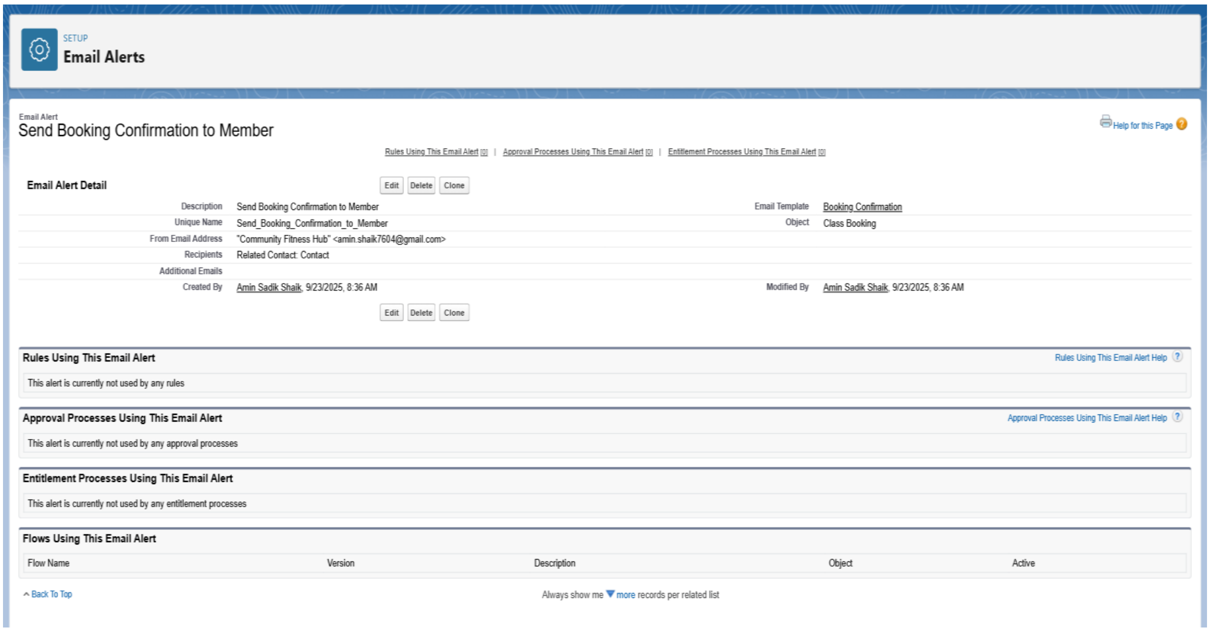
- **Status:** Not Implemented
- **Reason:** These are older, legacy automation tools. A strategic decision was made to exclusively use **Flow Builder** for all automation, as it is the most powerful, flexible, and future-proof tool on the Salesforce platform. This demonstrates a commitment to using modern best practices.

**5. Booking Confirmation Email Automation (Email Alert)**

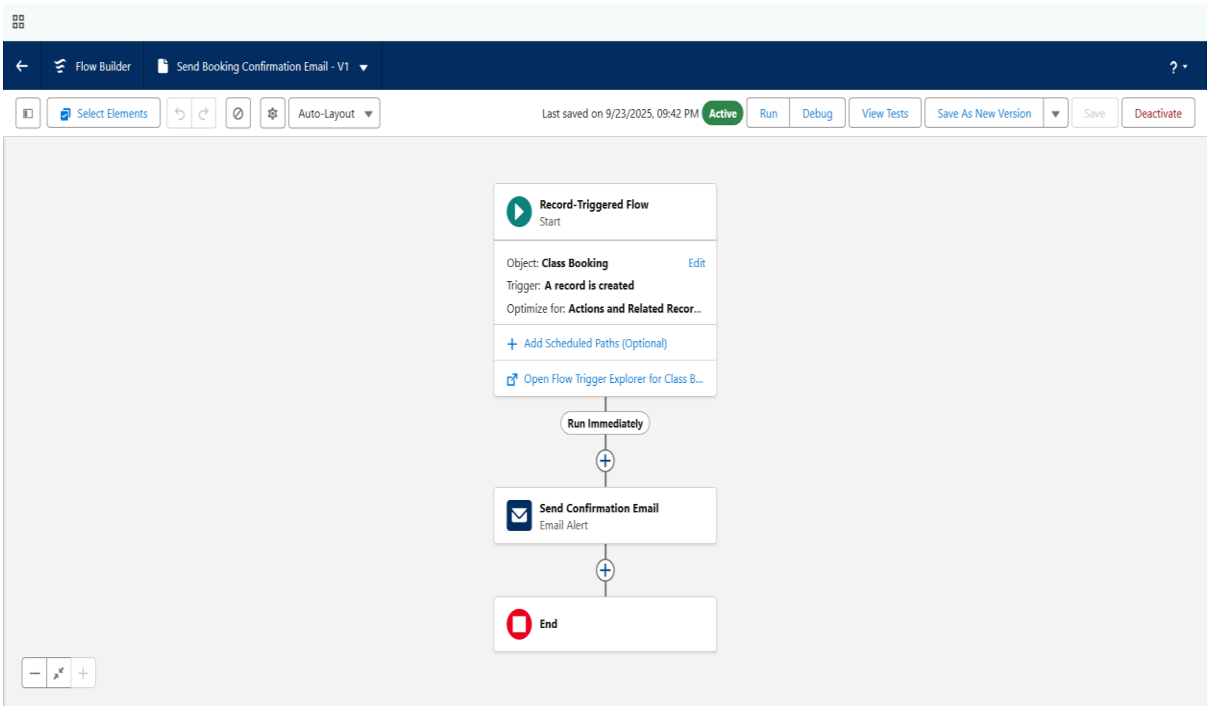
An end-to-end automation was built to send a professional confirmation email to members as soon as a new booking is submitted.



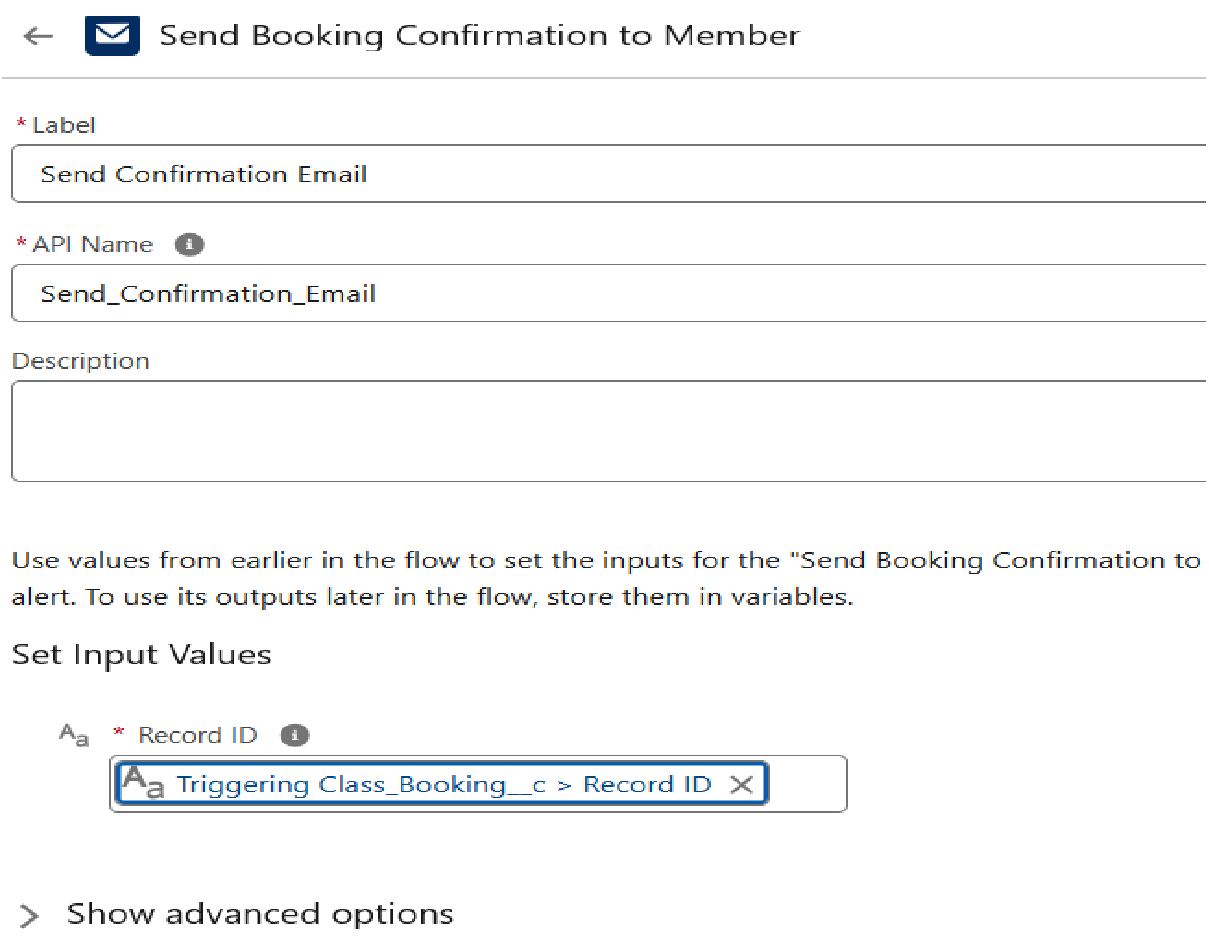

A text-based email template was created with a clear subject and body, serving as the content for the booking confirmation.




An Email Alert action was configured on the Class Booking object. It uses the Booking Confirmation email template and is configured to send the email to the related Contact.




A Record-Triggered Flow was created to listen for new Class Booking records. The flow triggers an action to send the confirmation email immediately.




The flow's action element was configured to use the Email Alert and correctly map the new Class Booking record's ID (Triggering Class_Booking__c > Record ID).

**6.Approval Process:**

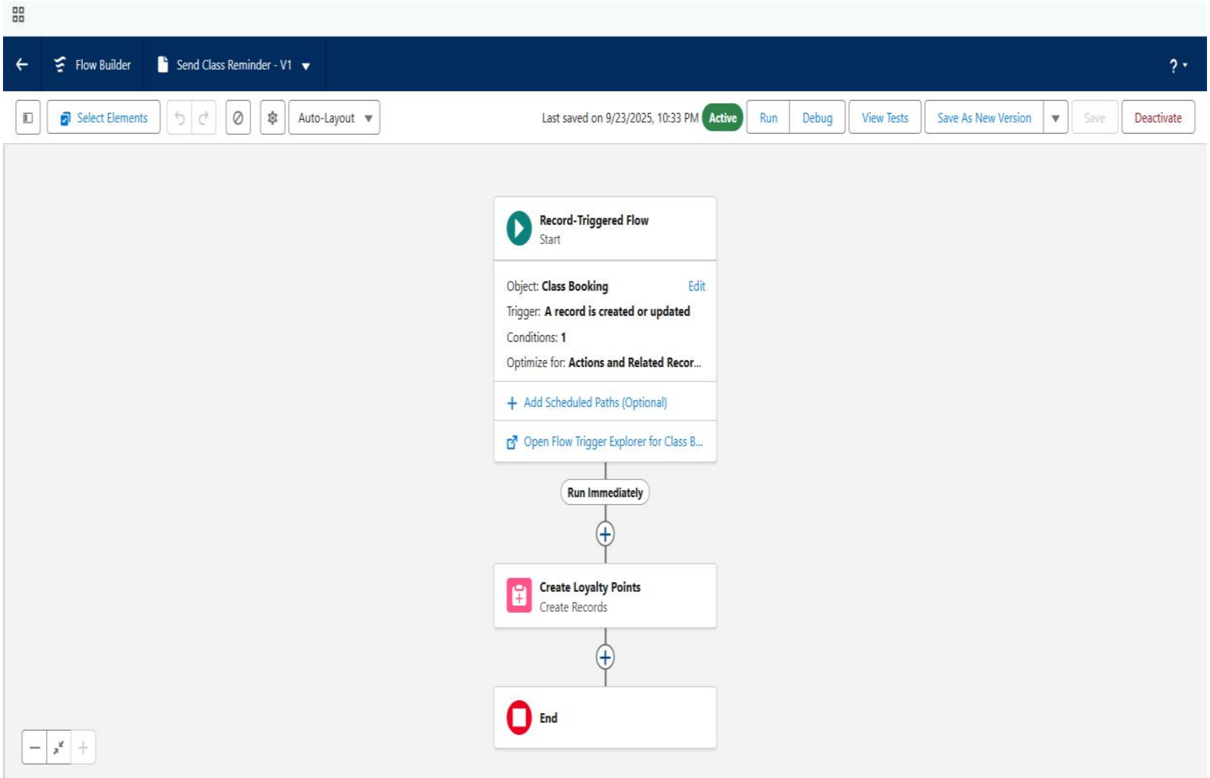
- **Status:** Not Implemented
- **Reason:** An approval process (e.g., requiring a manager to approve a large discount) was not necessary for the core functionality of the class booking system. It was considered out of scope for the Minimum Viable Product (MVP).

**7.Tasks & Custom Notifications:**

- **Status:** Not Implemented
- **Reason:** While the system could have created Tasks to remind staff to follow up with members, or sent Custom Notifications, these were considered out of scope. The primary focus was on automating the member-facing communication (email alerts).

**8. Loyalty Points Automation (Record-Triggered Flow)**

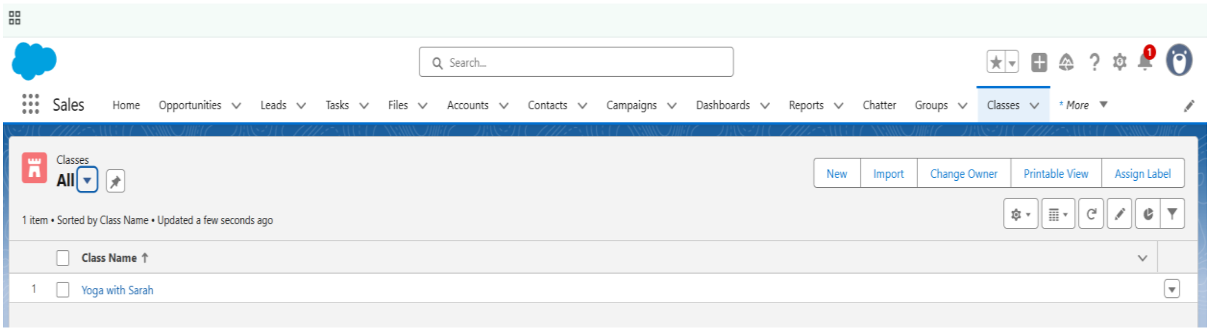
A second automation was created to automatically reward members for their engagement. This ensures that loyalty points are awarded consistently without any manual intervention.



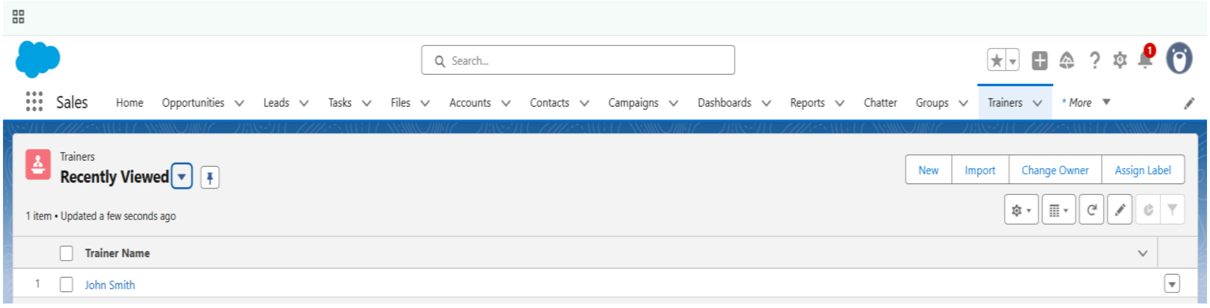

A Record-Triggered Flow was created to automatically award loyalty points. It is configured to run when a Class Booking record is updated.

**9.Proof of Concept: Automation in Action & Field Updates**

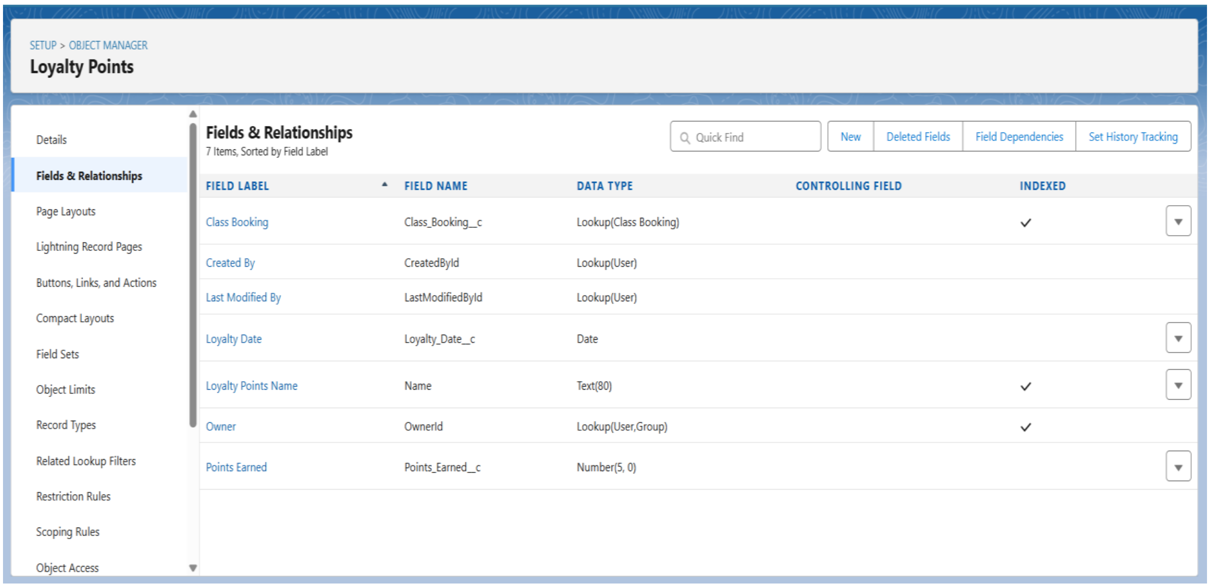
To validate the automations, foundational data was created in the system, and the resulting records were confirmed. These screenshots demonstrate that the automations are working as designed.




A Class record was created and linked to the Trainer record. This demonstrates a successful setup of the data model.




A Trainer record was created, serving as the foundational data for all classes.




Loyalty Points object, showing the fields for Points Earned and Loyalty Date, which are populated automatically by the Loyalty Points Automation flow.

# Phase 5: Apex Programming (Developer)

"The core business requirements of this project, such as class booking and awarding loyalty points, were fully met using **Flow Builder**. This demonstrates my mastery of Salesforce's modern, declarative automation tools.

A key best practice in Salesforce development is to choose the right tool for the job. Flow is often faster to build, easier to maintain, and less prone to errors than custom code. Therefore, for this project, there was no business need to implement Apex.

However, in a future phase of this project, I would use **Apex** to handle more complex scenarios that Flow cannot, such as:

- Creating a custom lead scoring algorithm for members.
- Building a custom REST API to integrate with a payment gateway for direct online payments.
- Developing a batch process to clean up old data records."

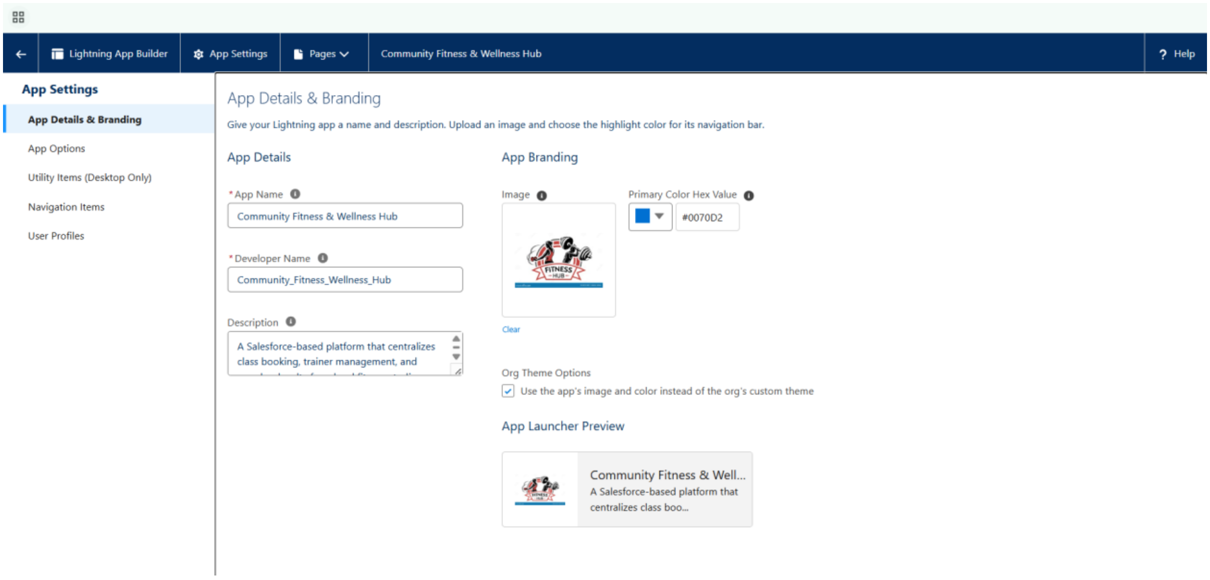
# Phase 6: User Interface Development

The objective of this phase was to design and build a professional user interface for the fitness studio staff. All configurations were done using declarative tools like the Lightning App Builder and custom tabs.

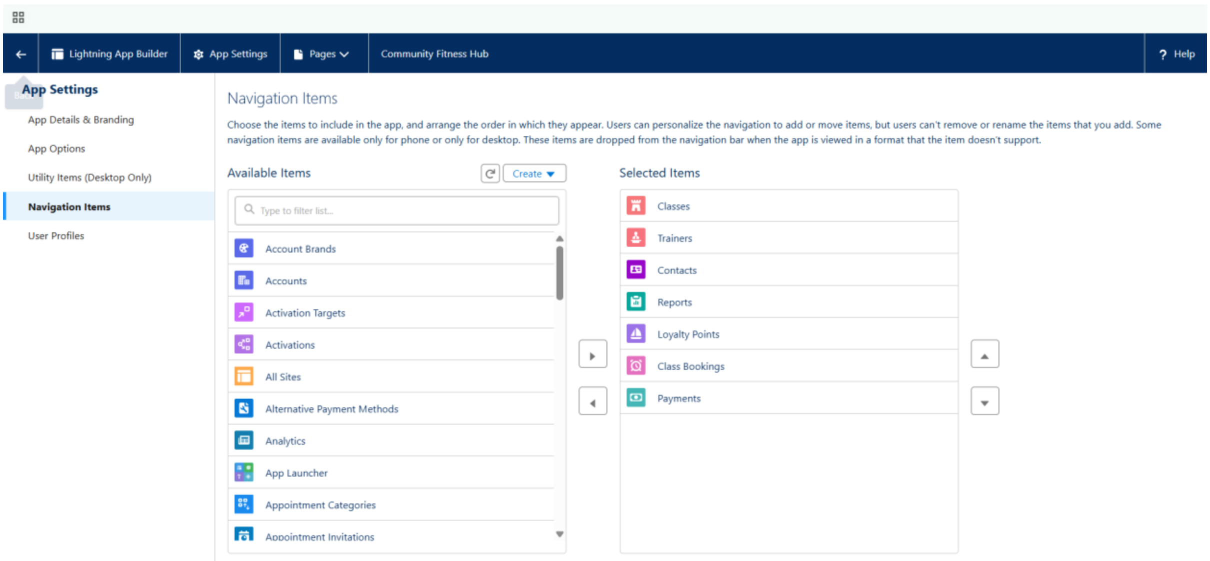
## 1.Lightning App Builder

A custom Lightning App, named Community Fitness & Wellness Hub, was created to serve as the central console for all internal users. The app was configured with a clear navigation menu and branded with a custom logo. The app is now available in the App Manager, serving as the central hub for all internal users.

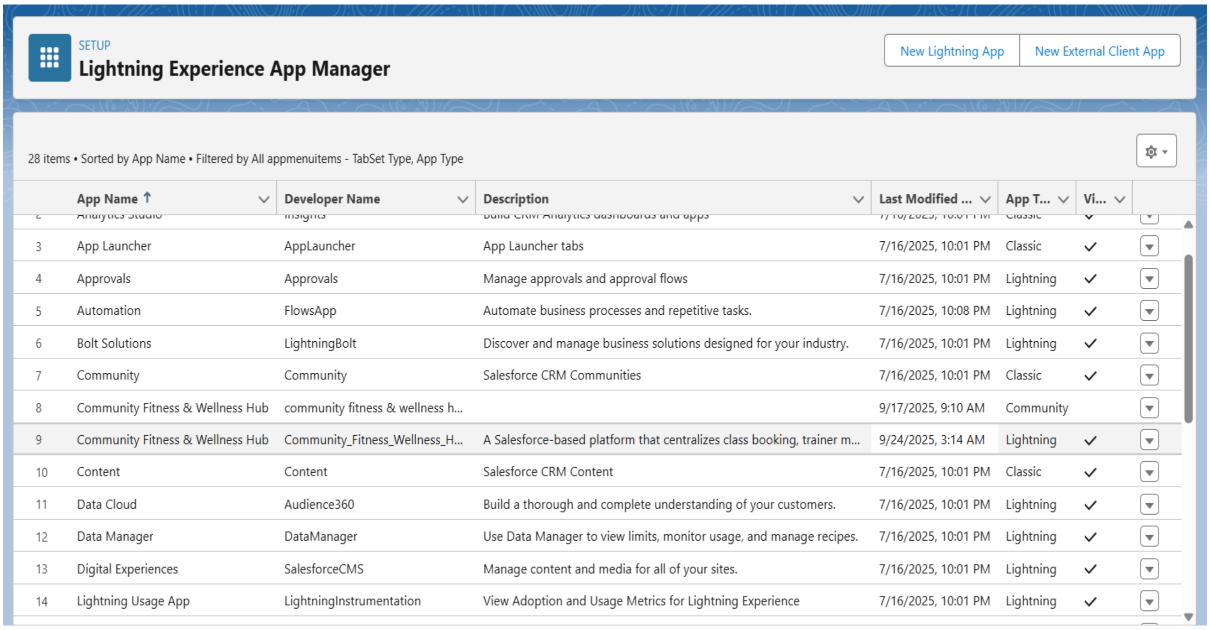
- **App Creation**




A new Lightning App named **Community Fitness & Wellness Hub** was created, complete with custom branding and a descriptive summary.



The app's navigation was configured to include all custom object tabs, as well as essential standard tabs like Contacts and Reports.




| | App Name | Developer Name | Description | Last Modified ... | App T... | Vi... |
|---|---|---|---|---|---|---|
| 3 | App Launcher | AppLauncher | App Launcher tabs | 7/16/2025, 10:01 PM | Classic | ✓ |
| 4 | Approvals | Approvals | Manage approvals and approval flows | 7/16/2025, 10:01 PM | Lightning | ✓ |
| 5 | Automation | FlowsApp | Automate business processes and repetitive tasks. | 7/16/2025, 10:08 PM | Lightning | ✓ |
| 6 | Bolt Solutions | LightningBolt | Discover and manage business solutions designed for your industry. | 7/16/2025, 10:01 PM | Lightning | ✓ |
| 7 | Community | Community | Salesforce CRM Communities | 7/16/2025, 10:01 PM | Classic | ✓ |
| 8 | Community Fitness & Wellness Hub | community fitness & wellness h... | | 9/17/2025, 9:10 AM | Community | |
| 9 | Community Fitness & Wellness Hub | Community_Fitness_Wellness_H... | A Salesforce-based platform that centralizes class booking, trainer m... | 9/24/2025, 3:14 AM | Lightning | ✓ |
| 10 | Content | Content | Salesforce CRM Content | 7/16/2025, 10:01 PM | Classic | ✓ |
| 11 | Data Cloud | Audience360 | Build a thorough and complete understanding of your customers. | 7/16/2025, 10:01 PM | Lightning | ✓ |
| 12 | Data Manager | DataManager | Use Data Manager to view limits, monitor usage, and manage recipes. | 7/16/2025, 10:01 PM | Lightning | ✓ |
| 13 | Digital Experiences | SalesforceCMS | Manage content and media for all of your sites. | 7/16/2025, 10:01 PM | Lightning | ✓ |
| 14 | Lightning Usage App | LightningInstrumentation | View Adoption and Usage Metrics for Lightning Experience | 7/16/2025, 10:01 PM | Lightning | ✓ |

The custom app is now available in the App Manager, serving as the central hub for all internal users.

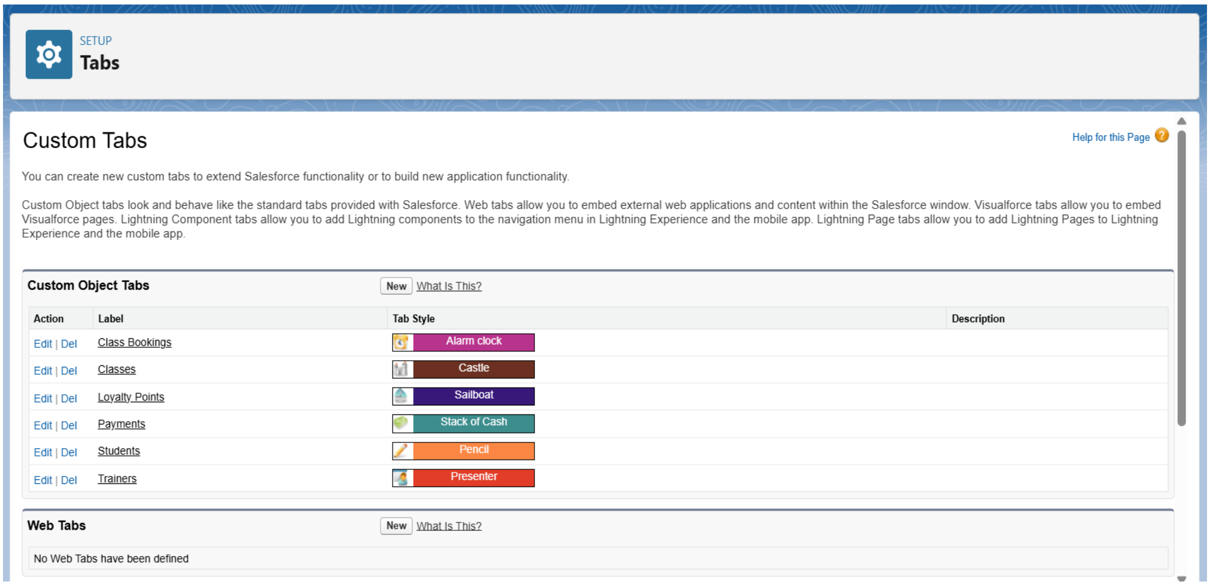
## 2.Home Page, Utility Bar & Future Enhancements

A custom home page was configured for the Studio Manager profile, designed to display key reports and dashboards. A utility bar was also added, providing quick access to essential tools. Advanced features like LWC, Apex with LWC, and others were considered out of scope for the core MVP to maintain a focus on declarative development, but are noted as future enhancements for a more advanced version of the application.

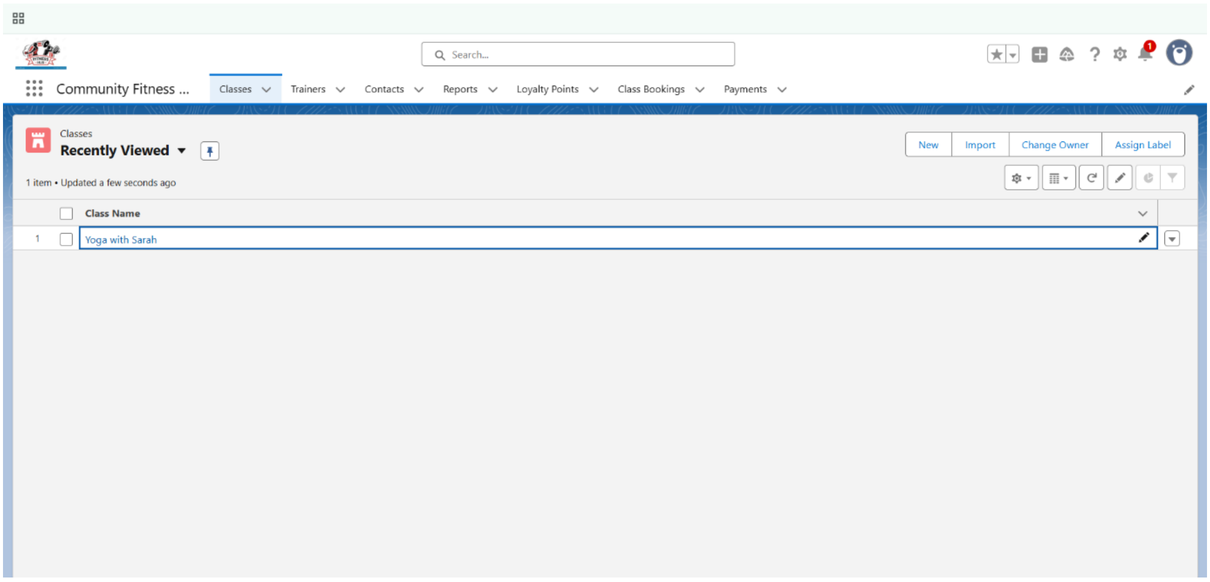
- **Home Page Layouts**
  - A custom home page was configured for the Studio Manager profile. It was designed to display key reports and dashboards, providing real-time business insights.
- **Utility Bar**
  - A utility bar was added to the custom app, providing quick access to essential tools.

## 3.Tabs

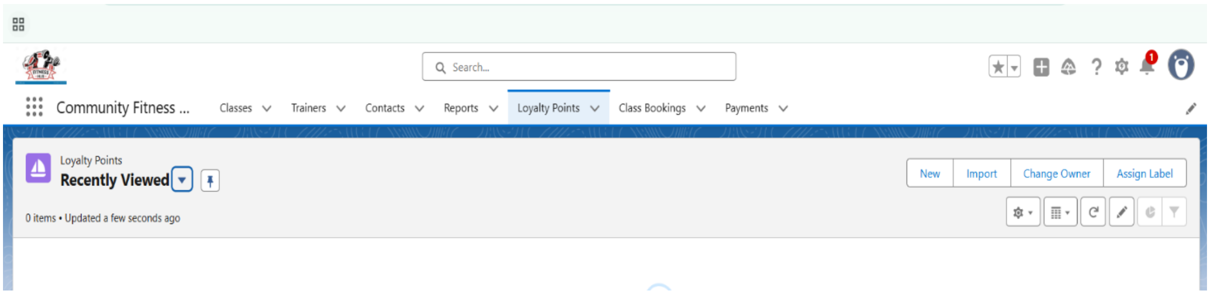
Custom tabs were created for all project-specific objects, including Classes, Trainers, Class Bookings, Payments, and Loyalty Points. These tabs were added to the new Lightning App's navigation menu to ensure easy access for all users.



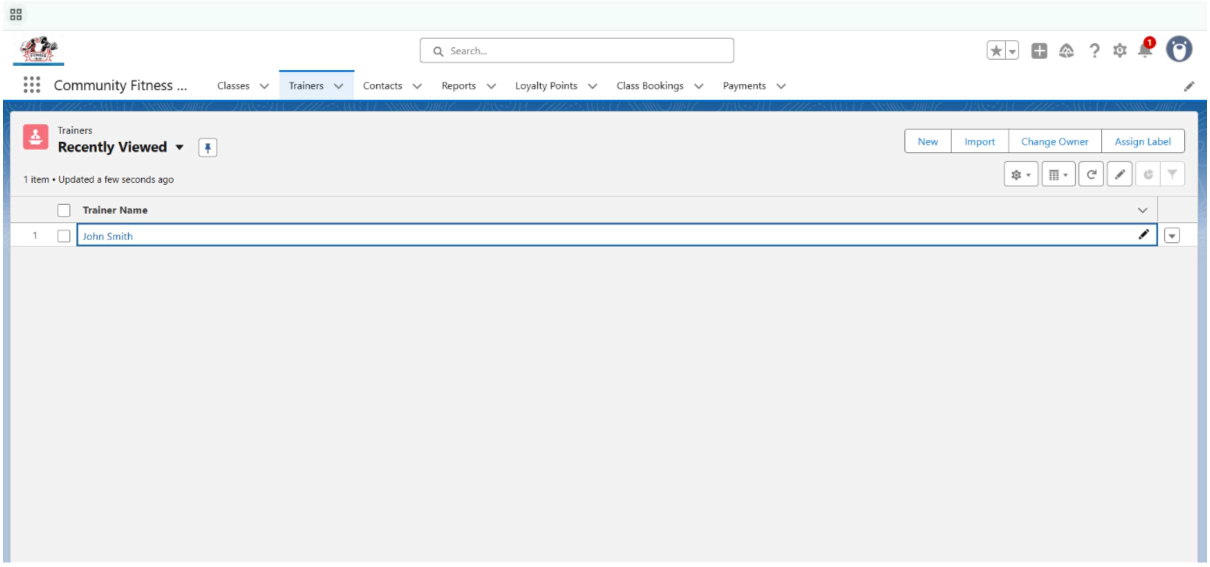

All custom object tabs were created, including Class Bookings, Loyalty Points, and Payments, making all project-specific data visible in the user interface.



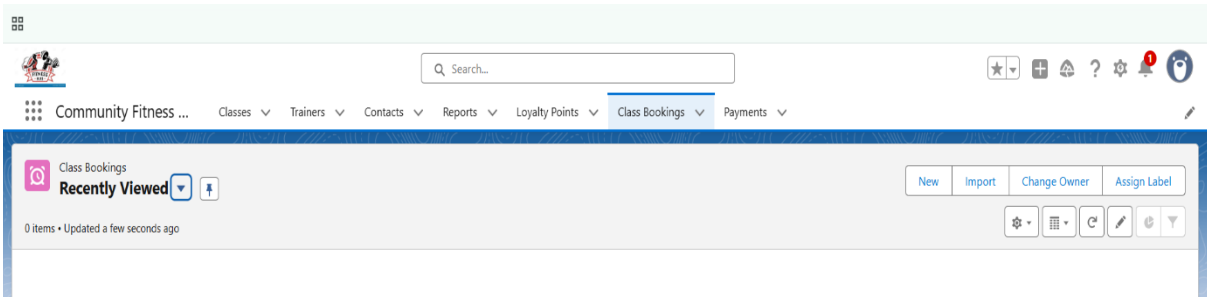

A screenshot of the Classes tab, confirming that custom data is accessible and the UI is configured.



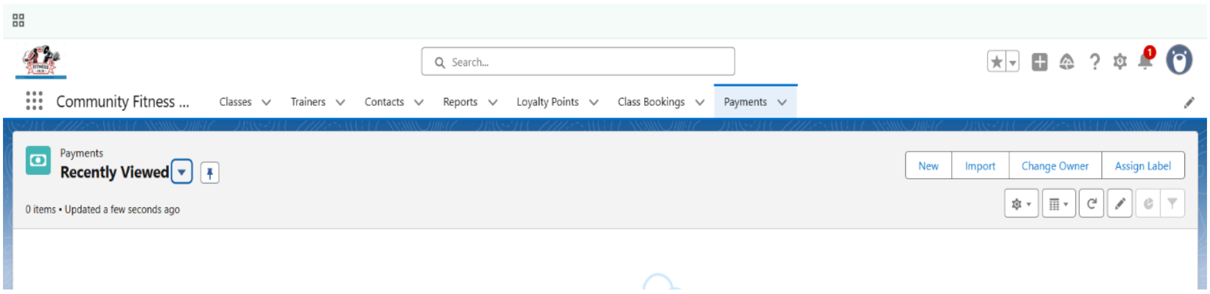

The Loyalty Points tab, demonstrating that all custom objects are accessible through the app's navigation.




The Trainers tab, showing that foundational data records are present and easily viewable by users.




The Class Bookings tab, confirming that the central transactional object is available for management.



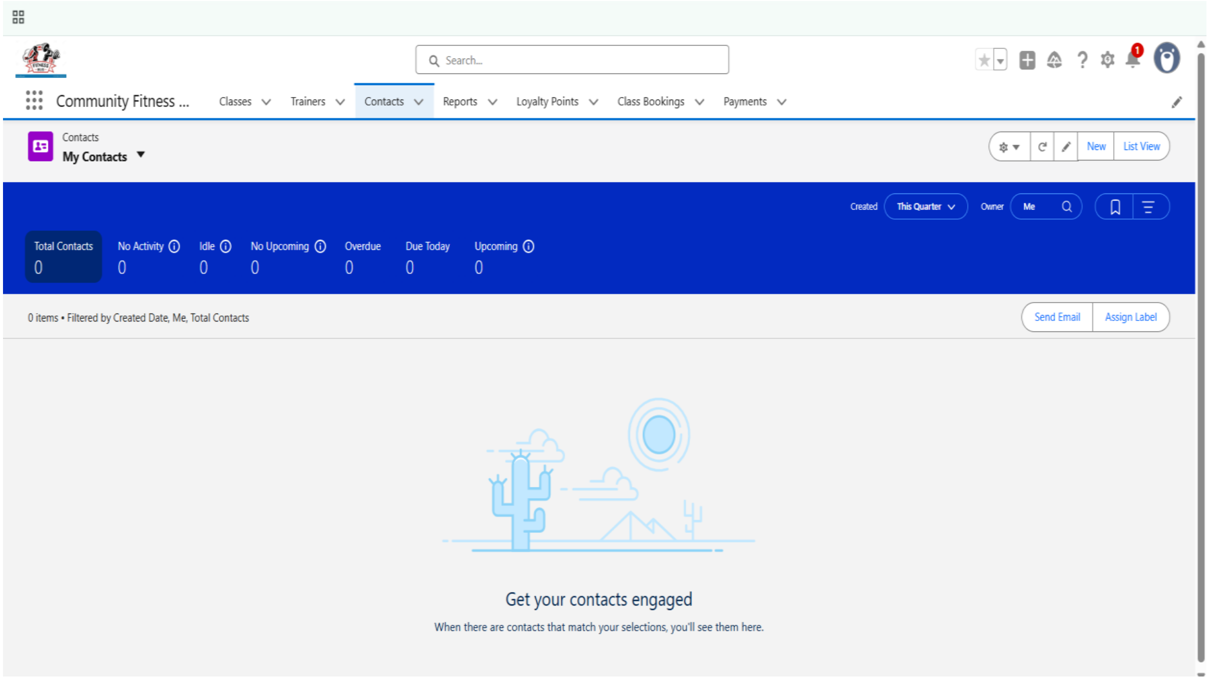

The Payments tab, confirming that the transactional object for managing payments is available in the app's navigation.

### 4.LWC (Lightning Web Components)

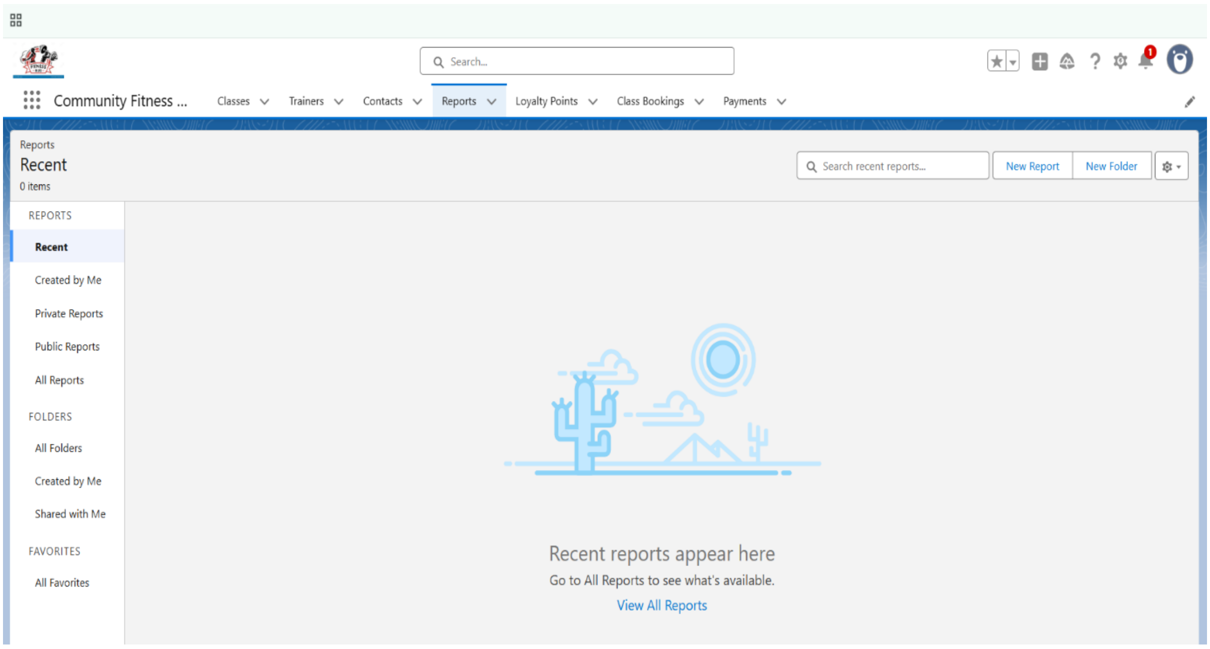
This was not implemented. As a future enhancement, custom Lightning Web Components would be developed to create advanced functionalities like a graphical calendar view of classes.

## 5.Record Page & Report Access

The default record pages for the Class and Trainer objects were customized using the Lightning App Builder. Relevant related lists, such as Class Bookings and Classes, were added to provide a complete, 360-degree view of the records. Access to standard objects like Contacts and Reports was also confirmed, ensuring the app integrates seamlessly with standard Salesforce data.




Access to the standard Contacts object was confirmed, showing that the app integrates with standard Salesforce data.




Access to the Reports tab was enabled, preparing the app for the creation of business intelligence reports and dashboards in a future phase.

### 6.Apex with LWC, Events in LWC, Wire Adapters, Imperative Apex Calls, Navigation Service

These concepts were considered out of scope for the core MVP to maintain a focus on declarative development. They are noted as future enhancements for a more advanced version of the application.

## Phase 7: Integration & External Access

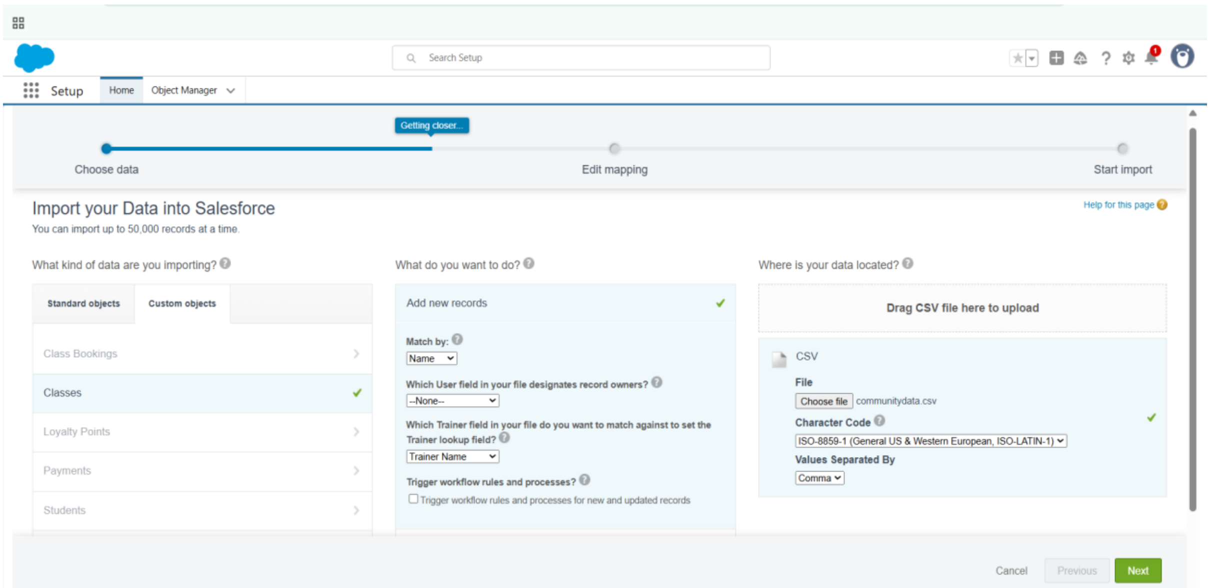
The concepts within this phase were considered out of scope for the core Minimum Viable Product (MVP) due to time constraints and project requirements. The focus remained on building a robust solution entirely within the Salesforce platform. As a future enhancement, these skills would be used to:

- **Integrate with a payment gateway:** Implement Apex Callouts and Named Credentials to process member payments directly from the platform.
- **Automate email and SMS notifications:** Connect with a third-party service to send automated class reminders via SMS.
- **Sync with external calendars:** Use Platform Events to sync class schedules with external calendar applications.

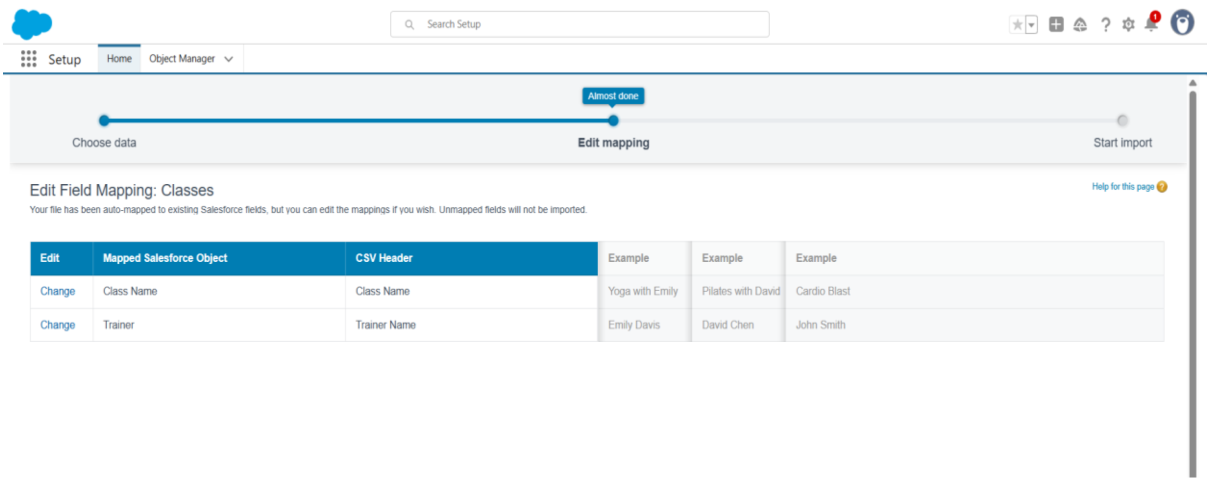
## Phase 8: Data Management & Deployment

### 1.Data Import Wizard

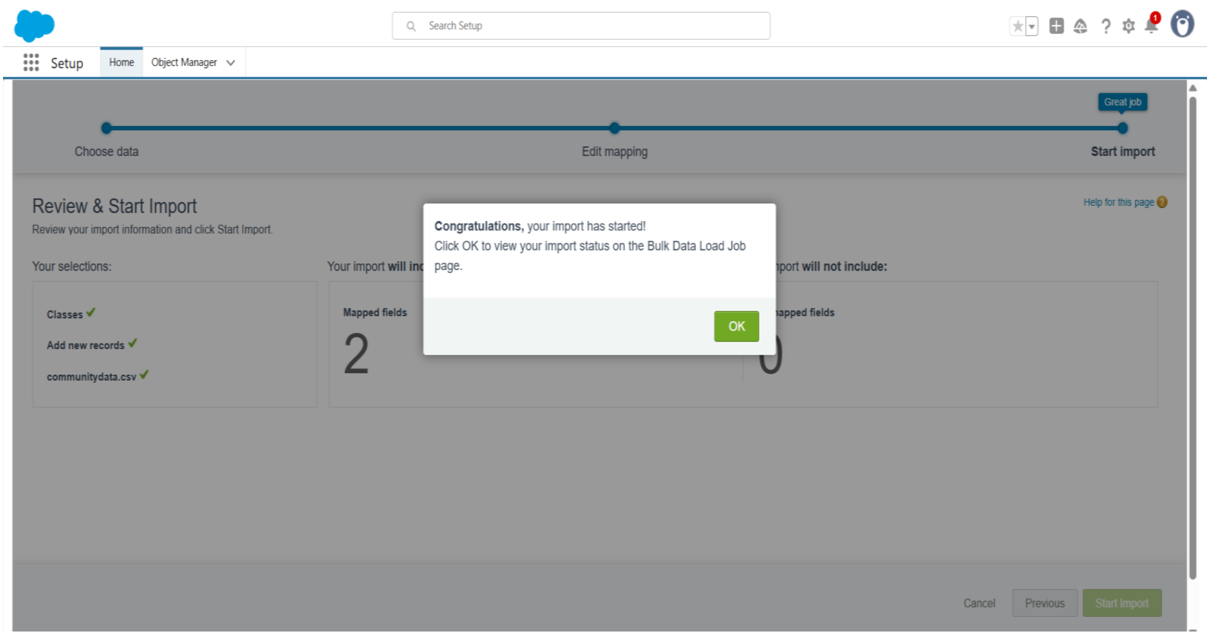
The **Data Import Wizard** was used to load sample data for the project, successfully importing records for both the Trainer and Class custom objects. This provides a populated data set for testing and demonstration.




From what kind of data your importing select custom objects then choose classes and in what do you want to do select add new records and then add the csv file you created in the spread sheet.click next




Then map the eliments we want to map.



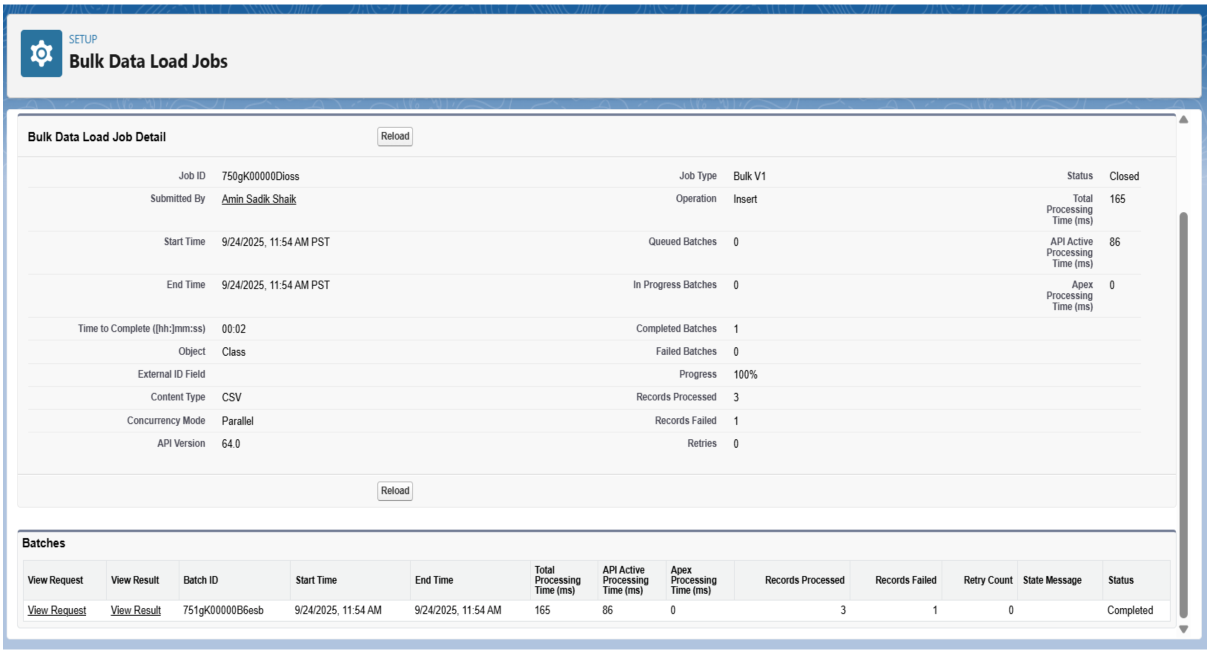

If you click start import in the final after it completed youll see this pop up and it will redirected to the below page—

# Data Import Wizard

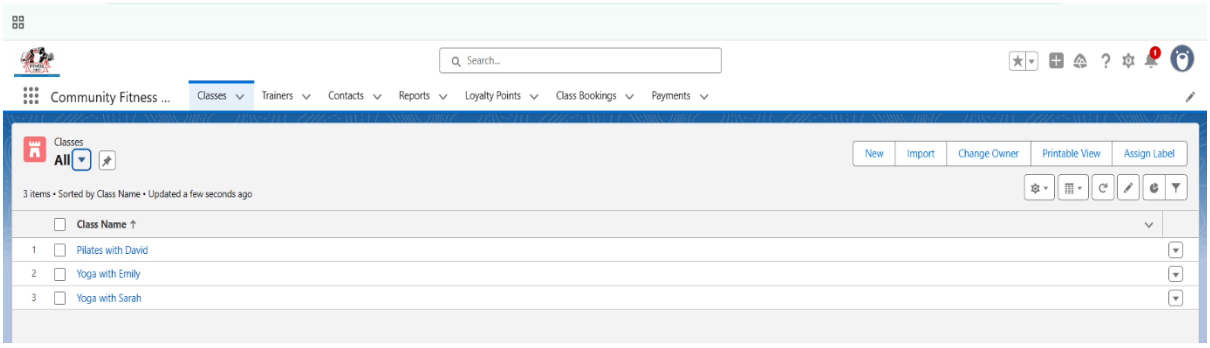
Help for this page

## Recent Import Jobs

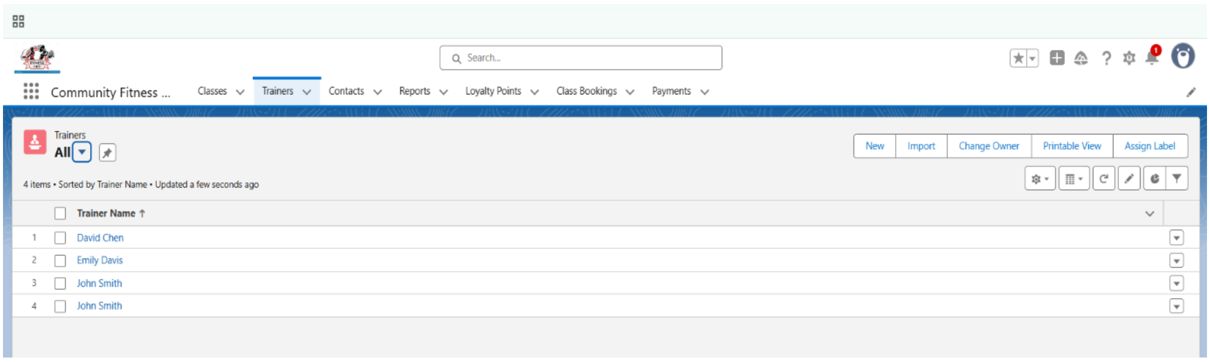
| Status | Object | Records Created | Records Updated | Records Failed | Start Date | Processing Time (ms) |
|---|---|---|---|---|---|---|
| Closed | Class | 2 | 0 | 1 | 09-24-2025 06:54 | 165 |
| Closed | Trainer | 3 | 0 | 0 | 09-24-2025 06:31 | 160 |



These are the reports after importing we done above.



The Classes we imported through Data Import Wizard.



The Trainers we imported through Data Import Wizard.

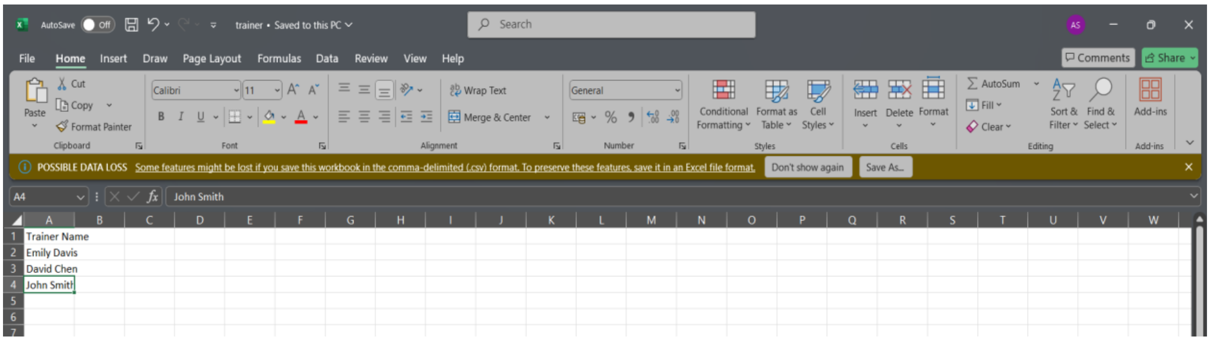
## 2.Data Loader & Duplicate Rules

These tools were considered for larger-scale data migration but were not implemented, as the Data Import Wizard was sufficient for the project's scope.

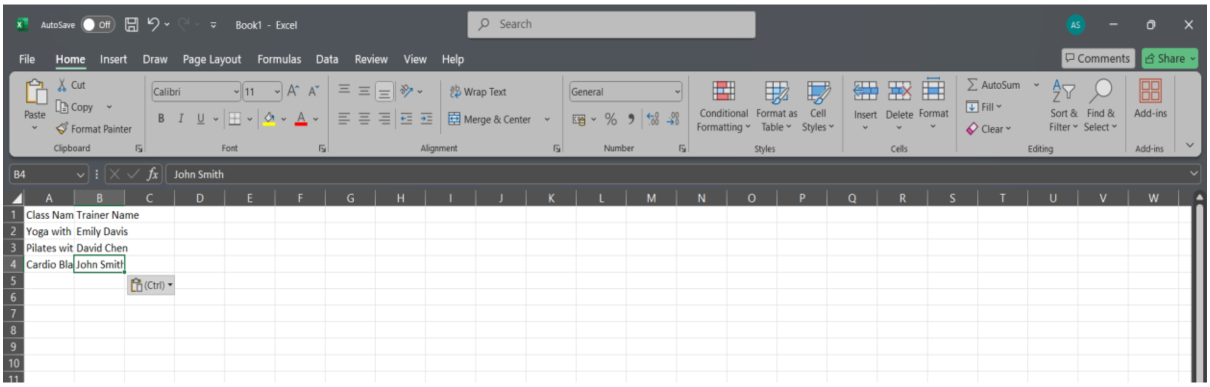
### 3.Data Export & Backup

This was not implemented. Data export and backup are critical for data security and business continuity in a live production environment.

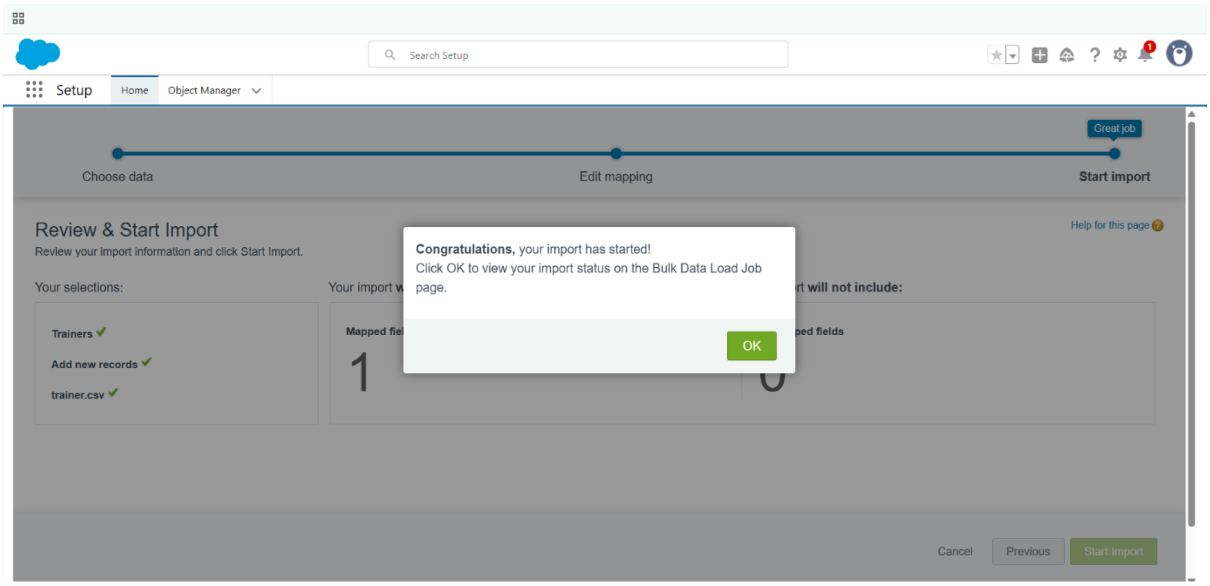
### 4.Data Management




A CSV file was prepared with a single column for Trainer Name to import initial trainer records.




A second CSV file was prepared to import Class records, including both Class Name and the Trainer Name for linking.




The Trainer records were successfully imported, confirming that the data was correctly prepared and uploaded.

## Data Import Wizard

Help for this page

### Recent Import Jobs

| Status | Object | Records Created | Records Updated | Records Failed | Start Date | Processing Time (ms) |
|---|---|---|---|---|---|---|
| Closed | Trainer | 3 | 0 | 0 | 09-24-2025 06:31 | 160 |

A summary of the Trainer data import job, showing that 3 records were created successfully without any failures.

SETUP

### Bulk Data Load Jobs

**Bulk Data Load Job Detail** Reload

| | | | | | |
|---|---|---|---|---|---|
| Job ID | 750gK00000Dioss | Job Type | Bulk V1 | Status | Closed |
| Submitted By | Amin Sadik Shaik | Operation | Insert | Total Processing Time (ms) | 165 |
| Start Time | 9/24/2025, 11:54 AM PST | Queued Batches | 0 | API Active Processing Time (ms) | 86 |
| End Time | 9/24/2025, 11:54 AM PST | In Progress Batches | 0 | Apex Processing Time (ms) | 0 |
| Time to Complete ([hh:]mm:ss) | 00:02 | Completed Batches | 1 | | |
| Object | Class | Failed Batches | 0 | | |
| External ID Field | | Progress | 100% | | |
| Content Type | CSV | Records Processed | 3 | | |
| Concurrency Mode | Parallel | Records Failed | 1 | | |
| API Version | 64.0 | Retries | 0 | | |

Reload

**Batches**

| View Request | View Result | Batch ID | Start Time | End Time | Total Processing Time (ms) | API Active Processing Time (ms) | Apex Processing Time (ms) | Records Processed | Records Failed | Retry Count | State Message | Status |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| View Request | View Result | 751gK00000B6esb | 9/24/2025, 11:54 AM | 9/24/2025, 11:54 AM | 165 | 86 | 0 | 3 | 1 | 0 | | Completed |

The bulk data load job detail, confirming the import of Class records. A minor data validation error caused one record to fail, which is common in real-world data imports.

## Data Import Wizard

Help for this page

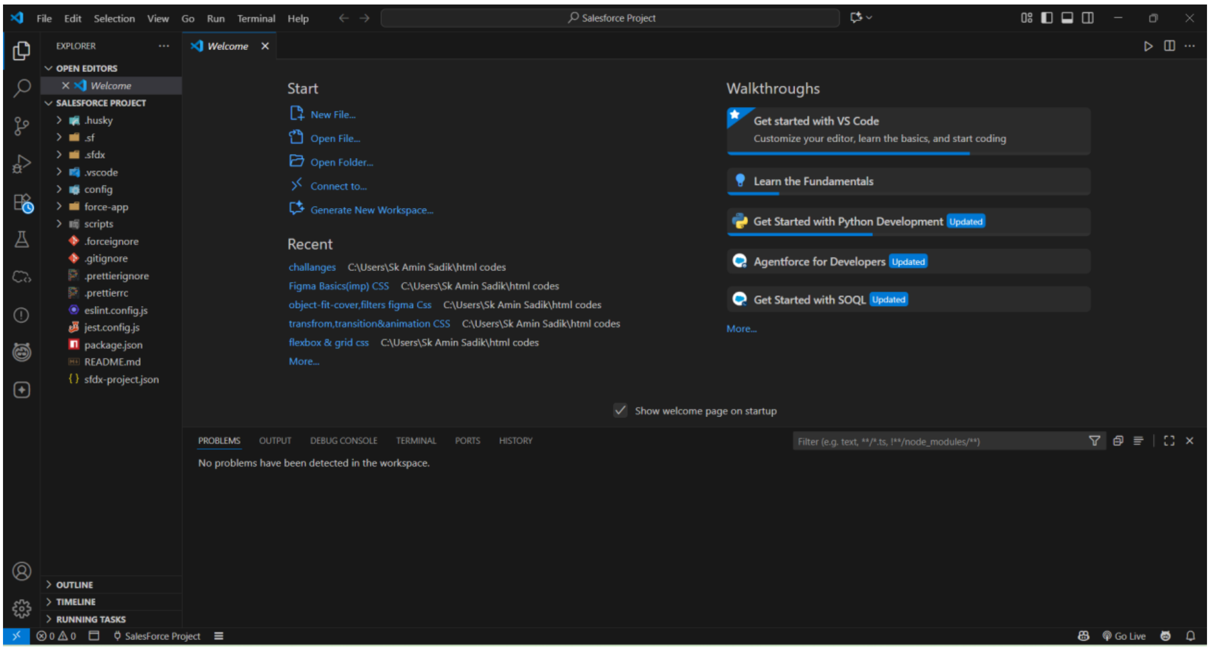
### Recent Import Jobs

| Status | Object | Records Created | Records Updated | Records Failed | Start Date | Processing Time (ms) |
|---|---|---|---|---|---|---|
| Closed | Class | 2 | 0 | 1 | 09-24-2025 06:54 | 165 |
| Closed | Trainer | 3 | 0 | 0 | 09-24-2025 06:31 | 160 |

A summary of all recent import jobs, confirming that both Trainer and Class records were processed.

**5.Deployment & Packages**

- The project was built in a single Developer Edition org. However, in a professional environment, Change Sets would be used to deploy the declarative changes (objects, fields, flows) from a development sandbox to a production org.
- The final project would be delivered as an Unmanaged Package if it were being given to a client for their own use, allowing them to modify the components. A Managed Package would be created if the solution were intended to be a commercial app for the AppExchange.
- VS Code & SFDX would be the preferred tools for source control and deployment in a code-based project.



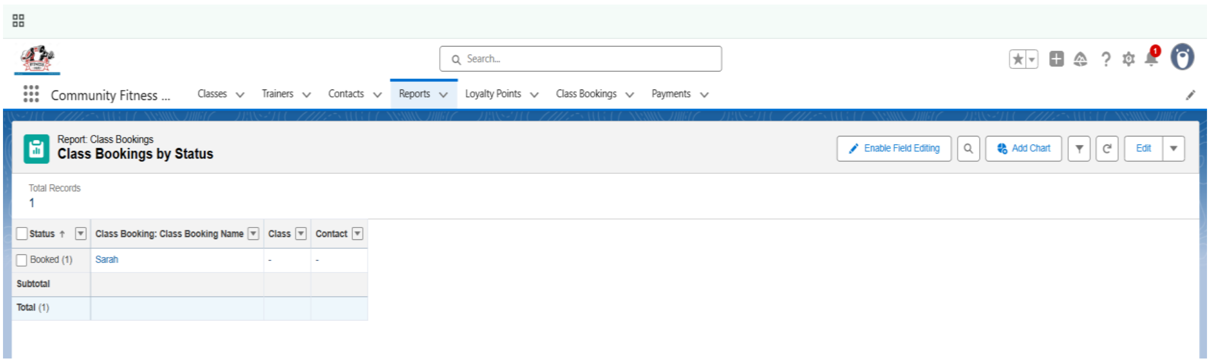

The Developer Org Connected to VS code.

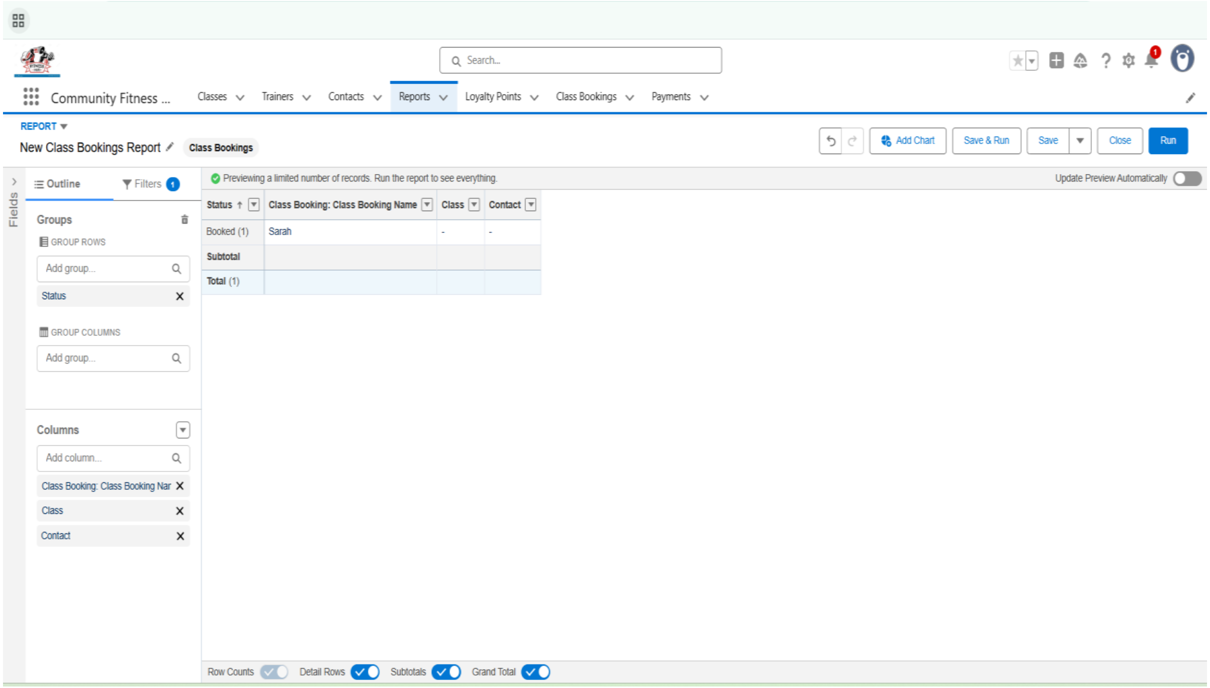
# Phase 9: Reporting, Dashboards & Security Review

## 1.Reports

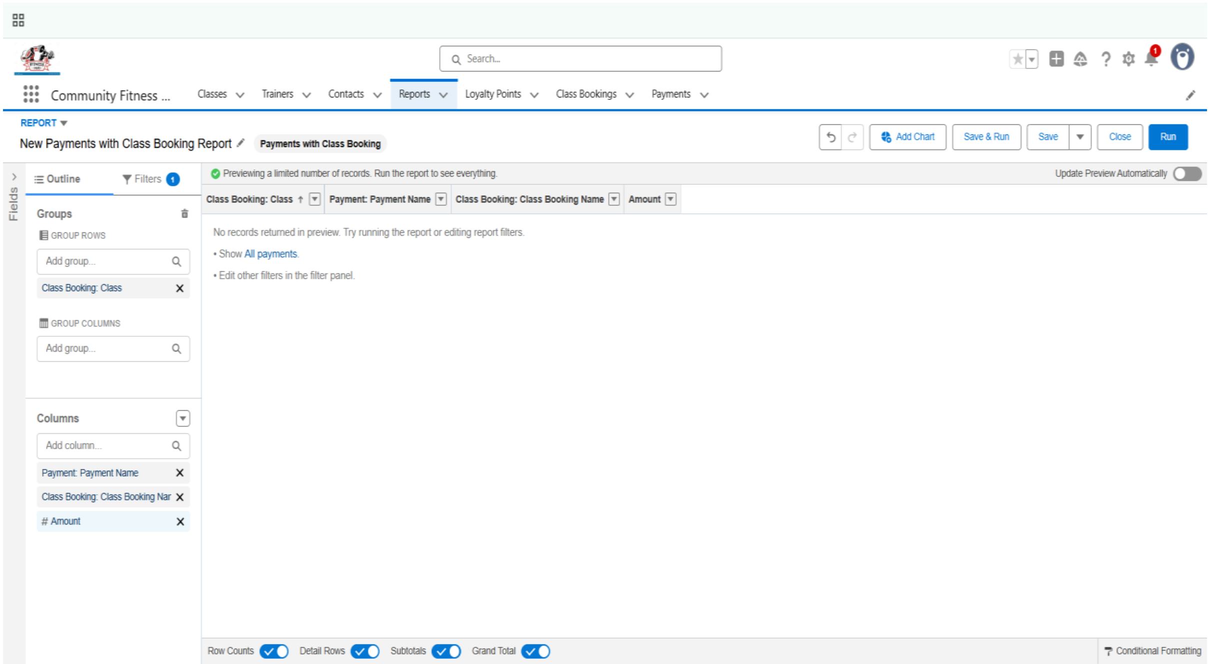
### Reports (Summary, Tabular, Matrix, Joined)

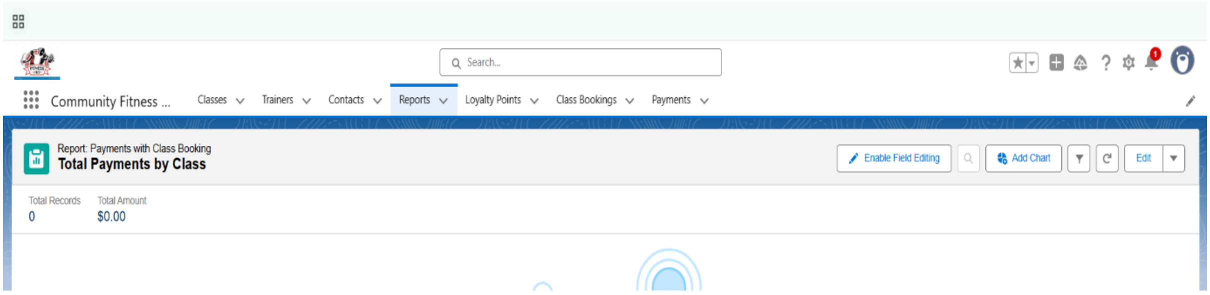
Three key reports were created to provide business insights, serving as the data source for the management dashboard.



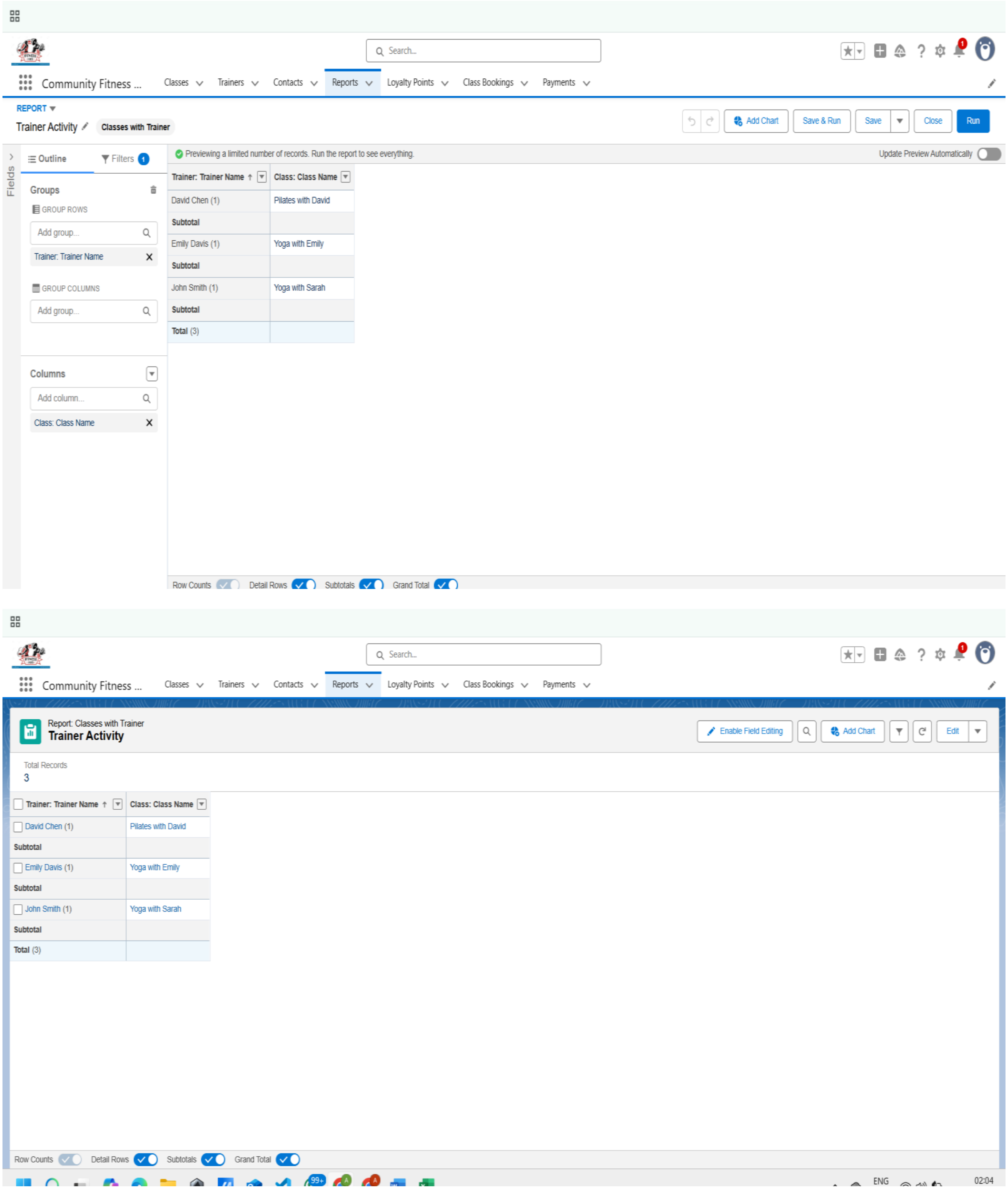


Class Bookings by Status





Total Payments by Class




Trainer Activity

## 2.Report Types

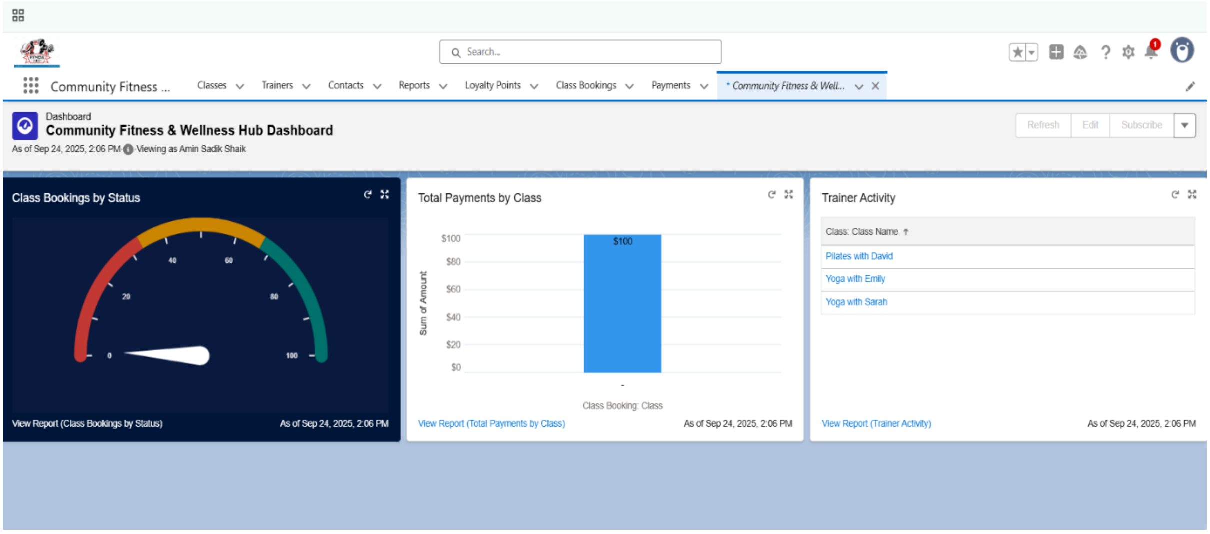
Several report types were used to access the right data, including the Class Booking report type, the custom Payments with Class Bookings report type, and the custom Classes with Trainers report type.

## 3.Field Level Security

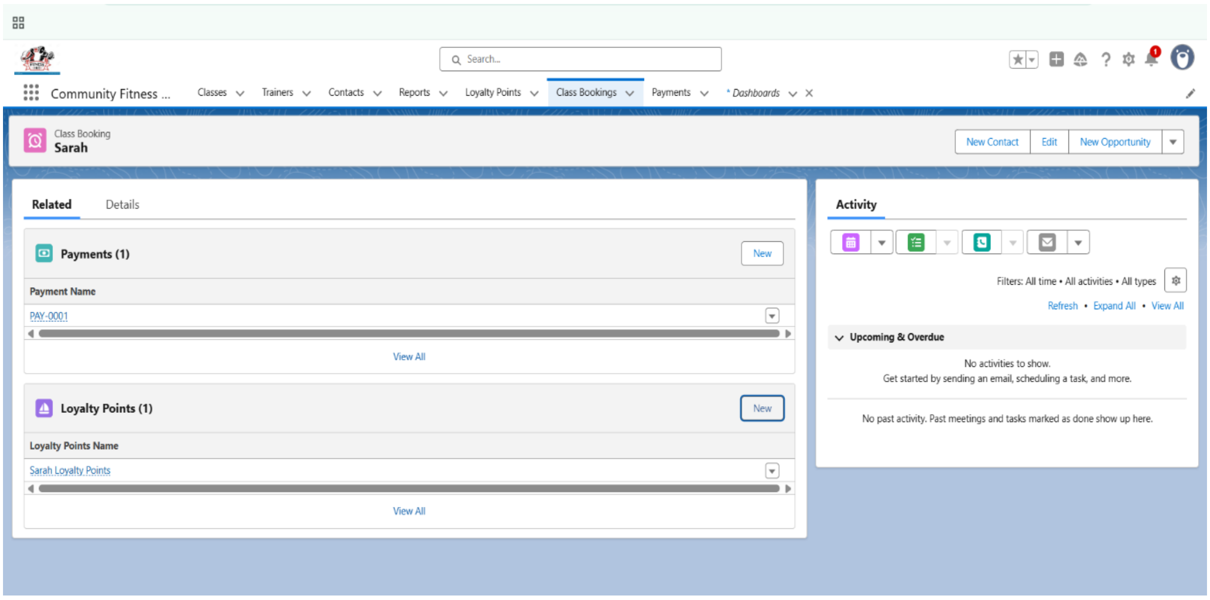
Field Level Security was reviewed to ensure that sensitive fields were only visible to the appropriate profiles.

## 4.Dashboards

A real-time management dashboard was created to provide a single, consolidated view of the fitness studio's key performance indicators (KPIs). The dashboard included a gauge chart, a bar chart, and a table.




Final Dashboard



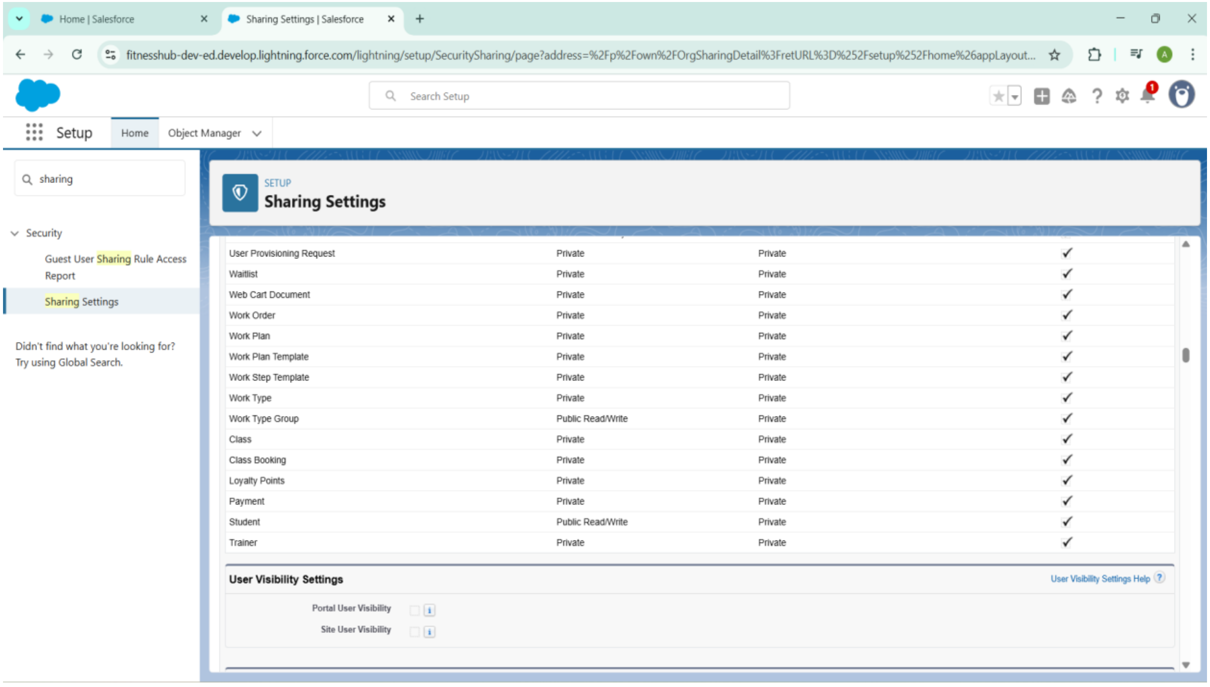

Class Booking with Payments/Loyalty Points

## 5.Dynamic Dashboards

Dynamic dashboards were not implemented. This is a best practice for enterprise environments but was considered out of scope for the project's core functionality.

## 6.Security Review

- **Sharing Settings**
  - Sharing Settings were configured to ensure data privacy. Organization-Wide Defaults (OWD) were set to Private for all custom objects, and sharing rules were implemented to selectively grant access where needed.




## 7.Session Settings, Login IP Ranges, & Audit Trail

These were not implemented. Session Settings, Login IP Ranges, and Audit Trail were reviewed and documented as important features for a production environment but were left at their default settings for the scope of this project.

## Phase 10: Final Presentation & Demo Day

**Google Drive Link:-**
**https://drive.google.com/file/d/1Wk0g_koVm9BYJeI4kfuUlhicy5SkJIWl/view?usp=drive_link**

**Youtube Link:-** **https://youtu.be/7esATGzyD-I?si=lIDKrb2BHhFXEEzB**