Health & Wellness Tracking Portal - Phase 1

The Health & Wellness Tracking Portal is an application designed to help patients monitor their health metrics and enable healthcare providers to track wellness trends. The system allows patients to log vital information such as blood pressure, sugar levels, and other health indicators, while doctors and health coaches can analyze these inputs through dashboards and reports.

Beyond simple record keeping, the portal identifies at-risk patients, highlights common health issues, and promotes proactive healthcare management. It integrates health tracking, analytics, and patient-doctor engagement into a single platform—an innovative solution for lifestyle and wellness management.

Requirement Gathering

- ➢ Allow patients to log vitals such as blood pressure, sugar levels, weight, and heart rate.
- ➢ Enable healthcare providers (doctors/coaches) to view patient trends over time.
- ➢ Generate dashboards that highlight at-risk patients based on abnormal readings.
- ➢ Send automated alerts/notifications when a patient logs critical values.
- ➢ Optionally, track lifestyle details like exercise, sleep, and diet for holistic wellness management.

Stakeholder Analysis

- ➢ Primary User: Patients who log their vitals and wellness information.
- ➢ Admin Role: Configures the system, manages patient data structure, and sets automation rules.
- ➢ Healthcare Providers: Doctors and wellness coaches who monitor trends and provide advice.
- ➢ Secondary Users: Family members or caregivers with shared access to patient data.

Business Process Mapping

- ➢ Log Health Data: Patients enter their daily vitals (e.g., BP, sugar levels).
- ➢ Monitor & Track: Data is stored and linked to patient profiles for ongoing monitoring.
- ➢ Alerts & Notifications: Automated alerts are triggered when abnormal values are recorded.
- ➢ Review Trends: Healthcare providers use dashboards and reports to analyze wellness patterns.
- ➢ Intervention: Doctors/coaches take preventive action for at-risk patients based on trends.

Industry-specific Use Case Analysis

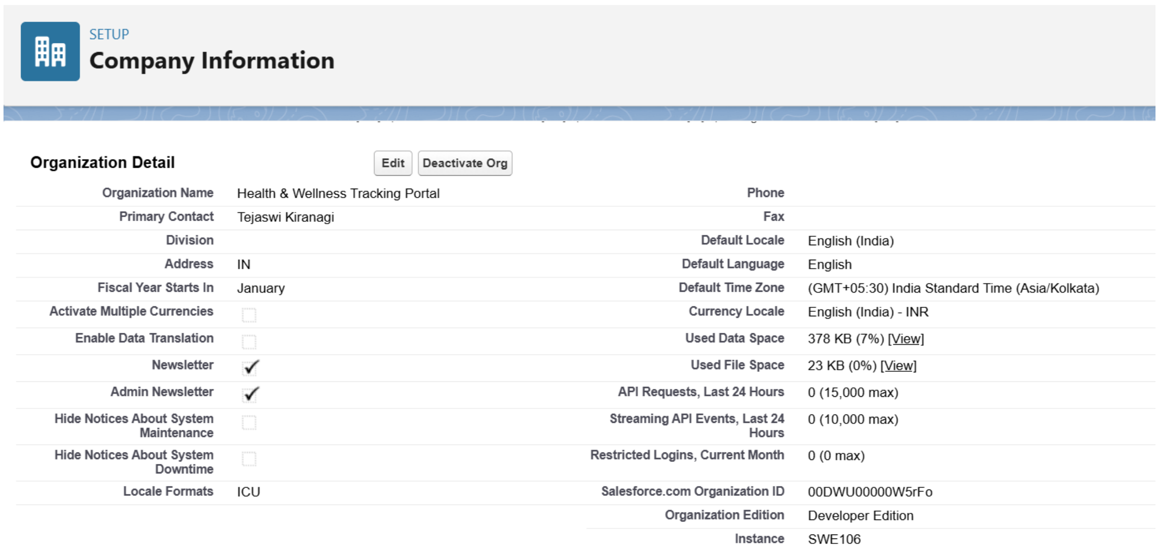
- ➢ Healthcare: Supports preventive care by monitoring health indicators in real time.

- ➢ Patient Engagement: Encourages active participation from patients in tracking their wellness.
- ➢ Analytics & Insights: Dashboards help healthcare providers identify trends and risks early.
- ➢ Lifestyle Management: Goes beyond treatment by including exercise, diet, and overall wellness tracking.

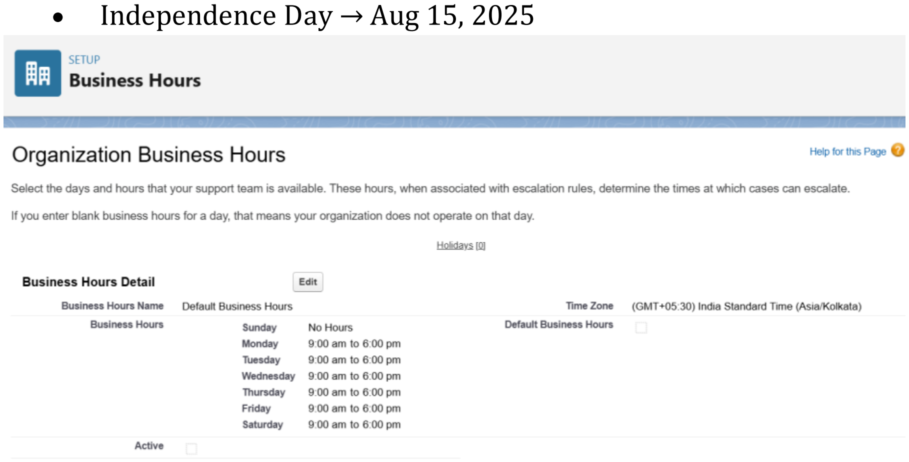
AppExchange Exploration

- ➢ Explored Salesforce Health Cloud and wellness-related apps for patient data tracking and engagement.
- ➢ Studied apps that automate alerts and integrate wearable data for best practices.
- ➢ The solution will be implemented using Salesforce tools like custom objects, flows, and dashboards, making it simple, scalable, and effective.

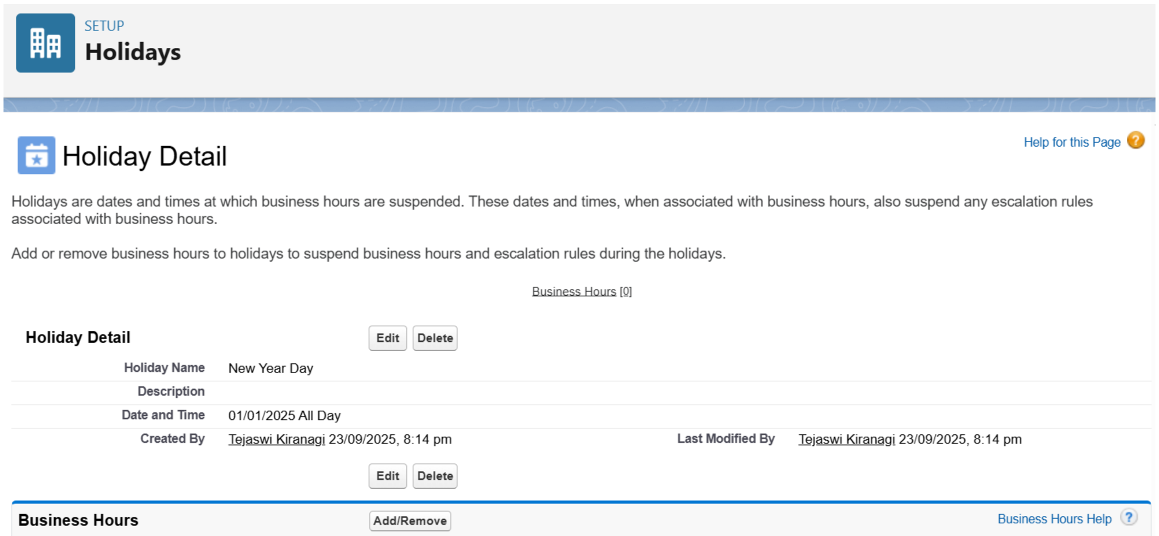
- **Phase-2: Org Setup & Configuration**
- **Step 1 — Company Profile**
- Configured the Salesforce org company profile to set basic organizational information.
  - Organization Name: Health & Wellness Tracking Portal
  - Default Time Zone: *(our timezone)*
  - Default Locale: English (India)
  - Default Currency: INR
  - Primary Contact: tejas/ health.@admin.com



- **Step 2 — Business Hours & Holidays**
- Defined business hours and public holidays for proper case escalation.
- **Business Hours:**
  - Name: Default Business Hours
  - Days Open: Monday–Saturday
  - Hours: 09:00 AM – 06:00 PM
- **Holidays:**
  - New Year's Day → Jan 1, 2025
  - Independence Day → Aug 15, 2025

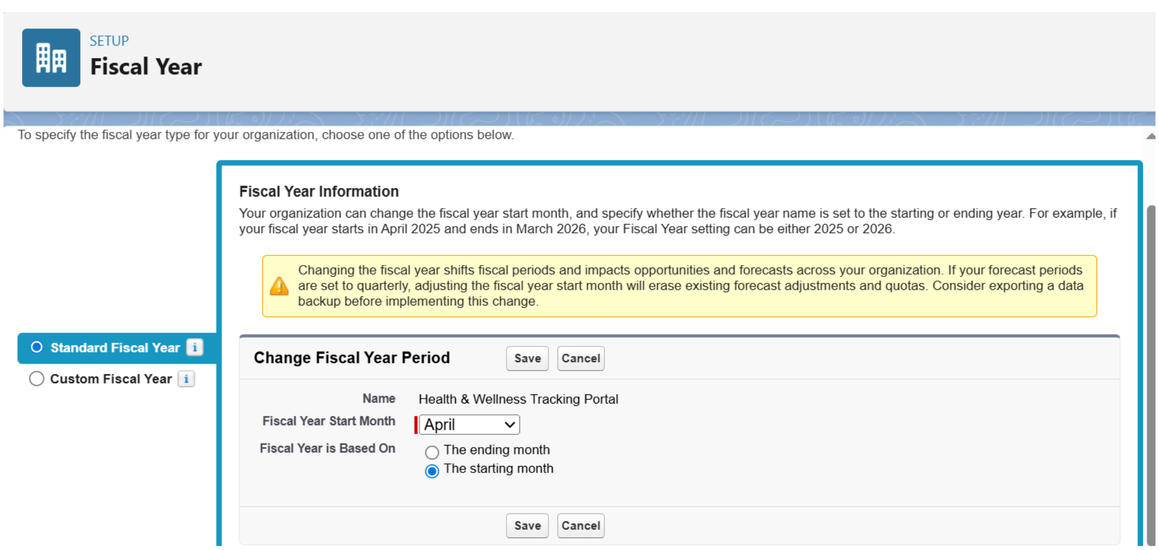


- **Step 3 — Fiscal Year**
- Configured standard fiscal year for reporting alignment.
  - Fiscal Year: Jan–Dec

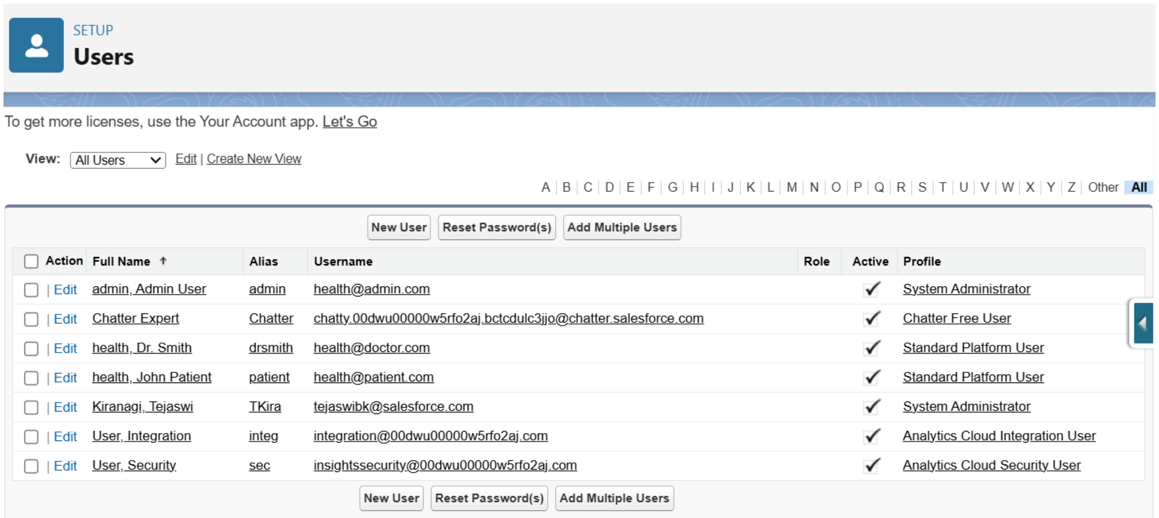


- **Step 4 — Users & Licenses**
- Created test users to simulate real-world roles:

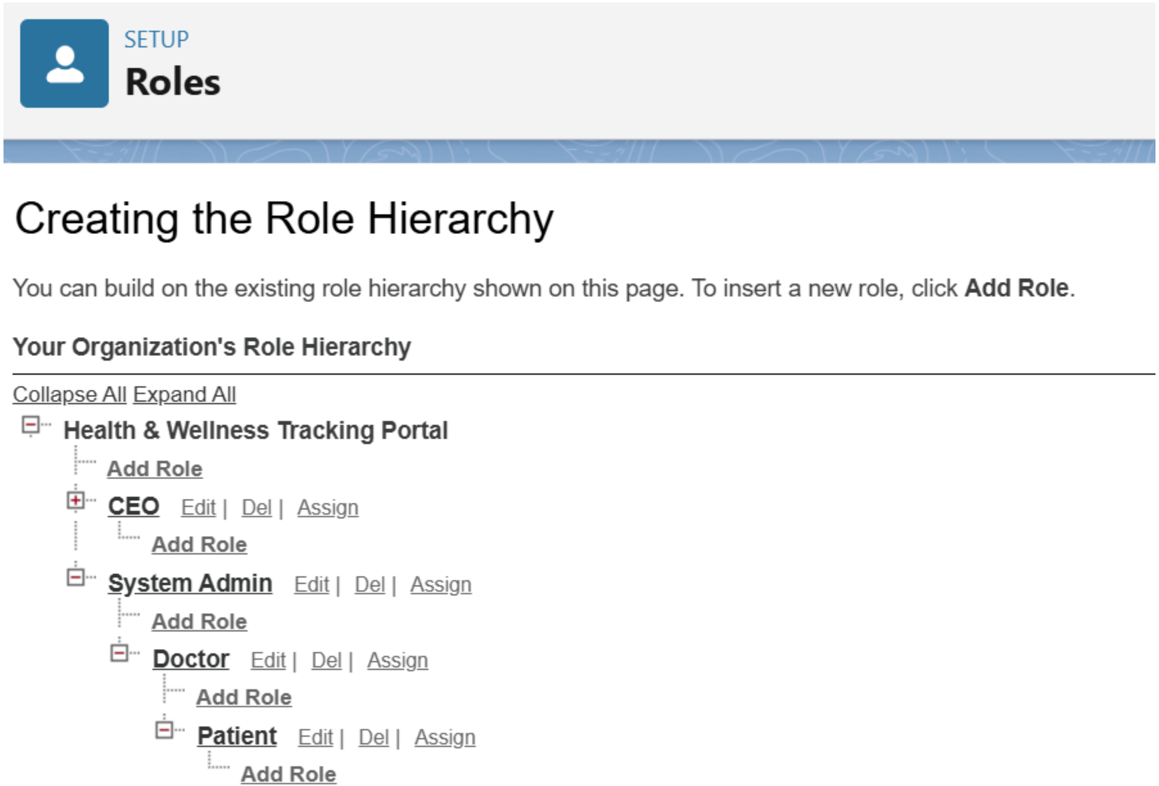
User	**Profile**	**Role**	**License**
Admin User	System Administrator	System Admin	Salesforce
Dr. Smith	Standard User	Doctor	Salesforce
John Patient	Standard User	Patient	Salesforce



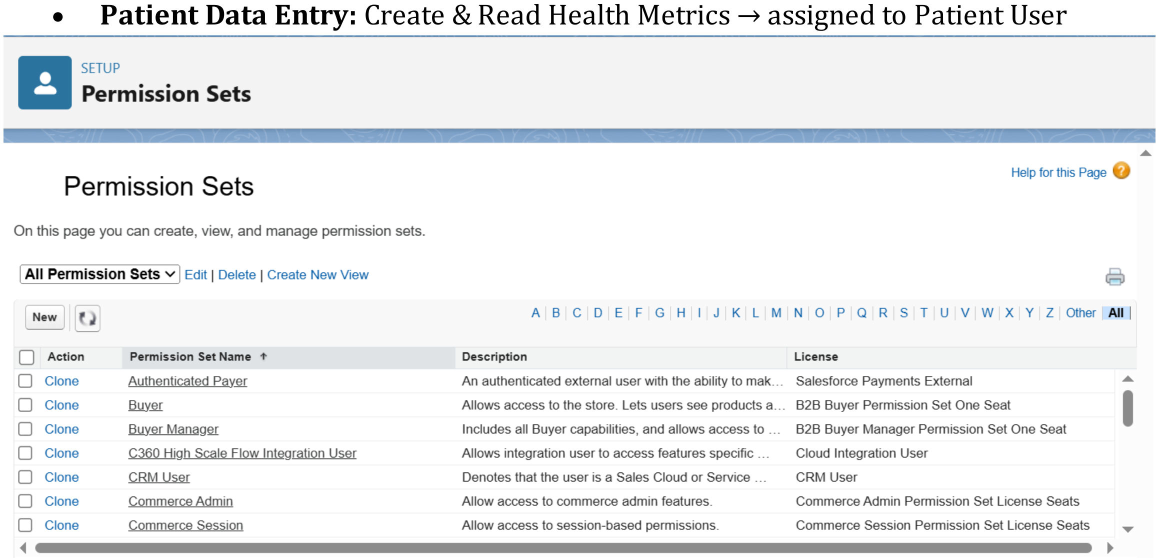
- **Step 5 — Profiles**
- Created custom profiles by cloning Standard User to define role-specific permissions.
  - **Doctor Profile:** Access to Patients & Health Metrics, Reports & Dashboards.
  - **Patient Profile:** Limited access to Health Metrics (create/view own records).



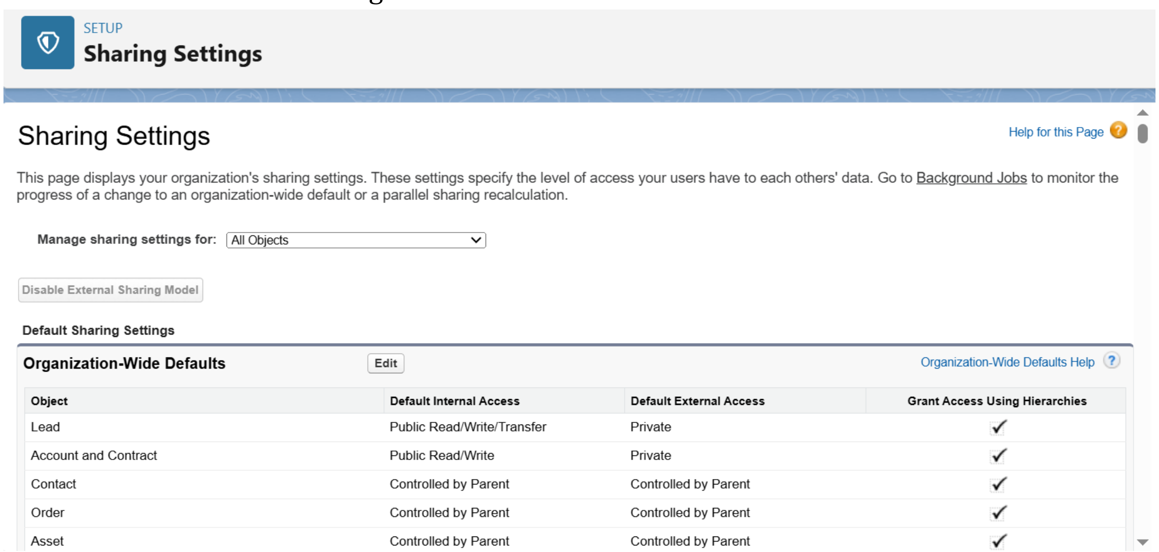
- **Step 6 — Roles & Role Hierarchy**
- Defined role hierarchy for record-level access control.
- **Hierarchy:**
- System Admin
- └── Doctor
- └── Patient



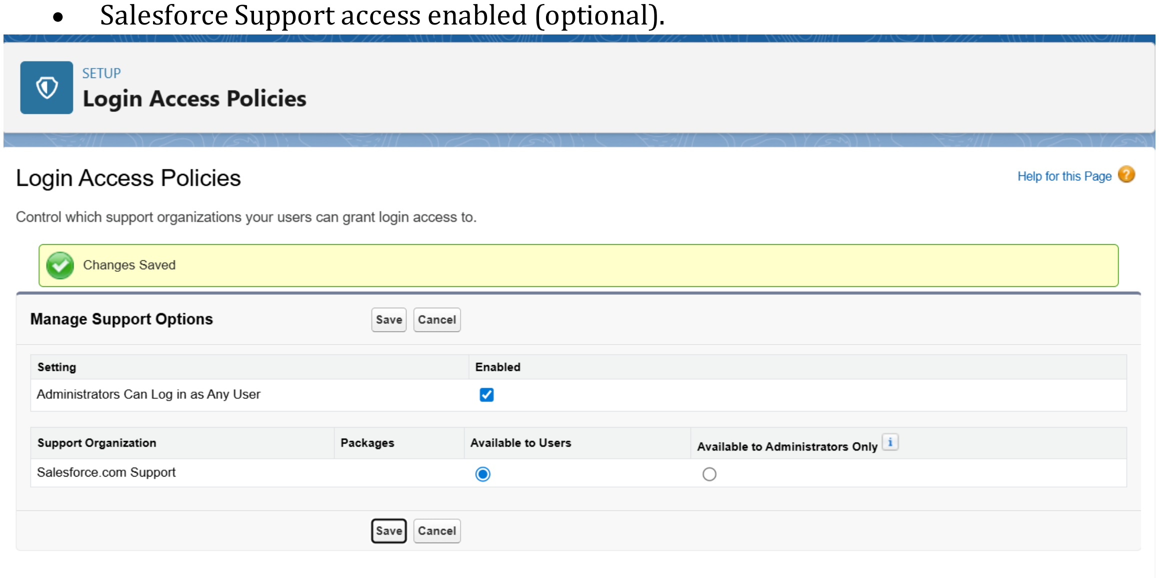
- **Step 7 — Permission Sets**
- Created additional access beyond profiles.
    - **Doctor Report Access:** Run/Create Reports & Dashboards → assigned to Doctor User
    - **Patient Data Entry:** Create & Read Health Metrics → assigned to Patient User



- **Step 8 — OWD & Sharing Rules**
- Configured baseline record access and exceptions.
- **Organization-Wide Defaults:**
    - Patient → Private
    - Health Metric → Private
    - Doctor → Public Read Only
- **Sharing Rules:**
    - Doctors can view & edit assigned Patient records.
    - Doctors can view assigned Health Metrics.



- **Step 9 — Login Access Policies**
- Enabled administrator login access for testing & support.
    - Admins can log in as any user.
    - Salesforce Support access enabled (optional).



- **Step 10 — Developer Org Setup**
- Set up a Salesforce **Developer Org** to serve as the main environment for the project.
  - Developer Org provides a permanent org for testing, building, and showcasing the project.
  - Recreated Phase-2 configurations (company profile, users, profiles, roles, OWD, permission sets) in this org.
-

- **Step 11 — Sandbox Usage**
- Explored Salesforce **Sandbox usage**:
  - Sandboxes allow testing changes without affecting production data.
  - Developer Orgs do not have Sandboxes by default.
  - Used Playground environments as testing grounds before replicating configurations in the main Dev Org.

- **Step 12 — Deployment Basics**
- Learned deployment methods in Salesforce:
  - **Change Sets**: Add components in source org → upload → deploy in target org.
  - Alternative options: Salesforce CLI, third-party CI/CD tools.

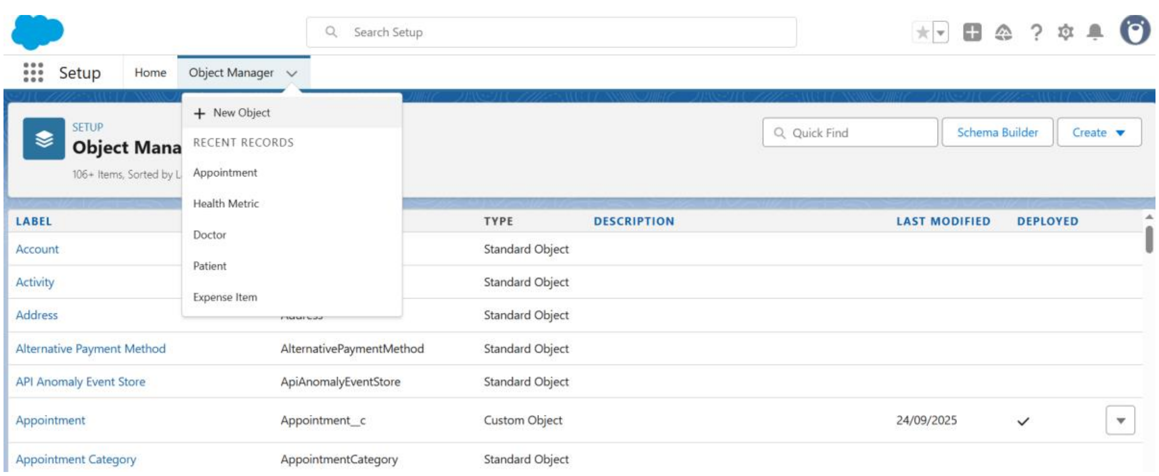
For this project, Change Sets are documented conceptually for future deployment.

- **Phase 3: Data Modeling & Relationships**
- 📌 **Objective**
- **The goal of this phase is to design the data model for the Health & Wellness Tracking Portal. It defines how different entities (objects) relate to each other, the fields they contain, and how information flows across the system.**

1. Standard & Custom Objects

- **Standard Objects Used**:
  - **User** → represents doctors, patients, and admins.
  - **Contact** → stores basic contact details.
- **Custom Objects Created**:

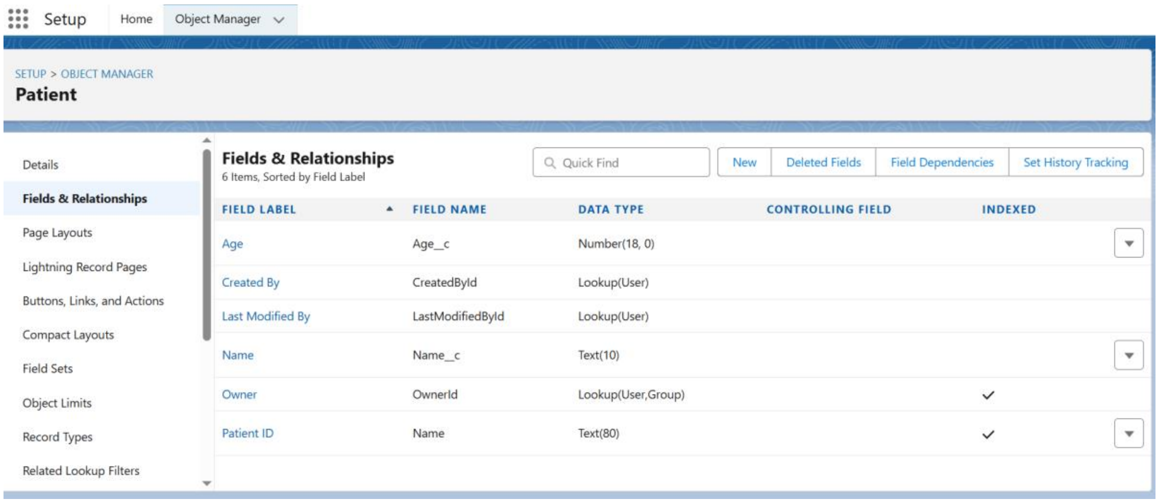
1. **Patient** → Stores patient-specific details.
2. **Doctor** → Maintains information about doctors.
3. **Appointment** → Tracks consultations between patients and doctors.
4. **Health Metric** → Logs details like blood pressure, sugar level, etc.

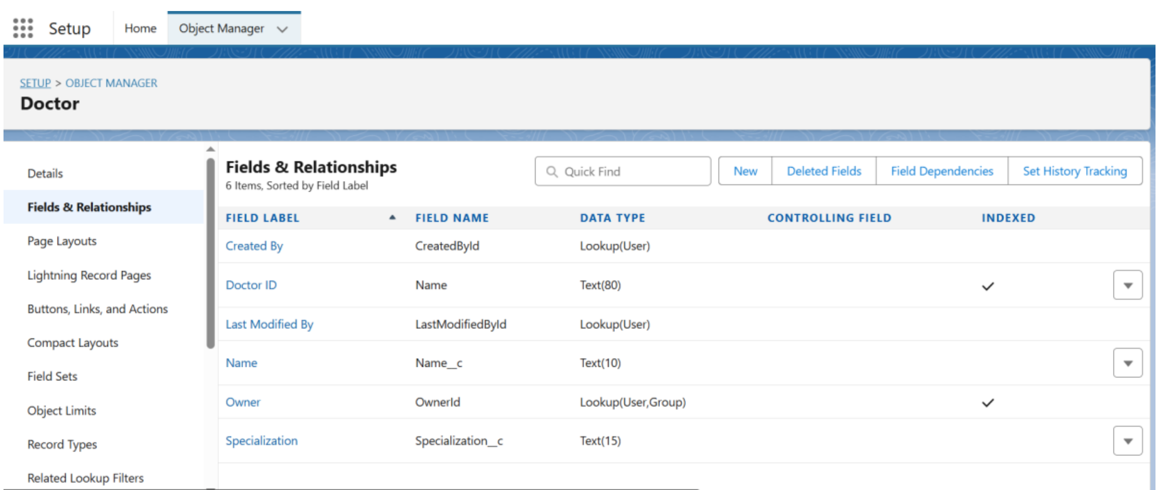


- **2. Fields**
- Each object is enhanced with custom fields to capture essential information:
  - **Patient Object**
    - Age (Number)
    - Gender (Picklist: Male, Female, Other)
    - Contact Number (Phone)
    - Address (Text Area)
  - **Doctor Object**
    - Specialization (Picklist: General, Cardiologist, etc.)
    - Experience (Number)
    - Contact Email (Email)
  - **Appointment Object**

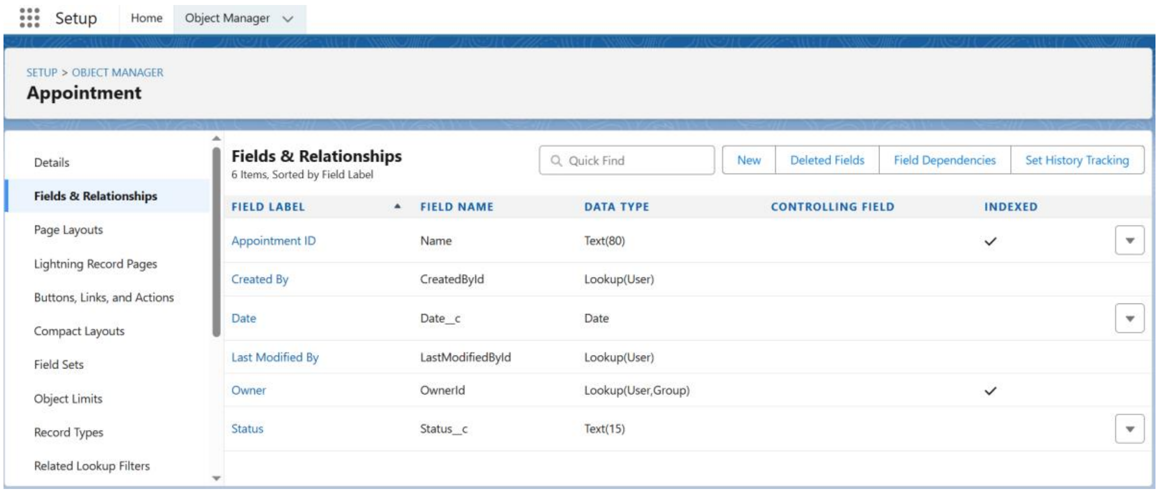
- Appointment Date (Date/Time)
- Status (Picklist: Scheduled, Completed, Cancelled)
- Notes (Long Text Area)

- **Health Metric Object**
  - Metric Type (Picklist: BP, Sugar, Weight, etc.)
  - Value (Number)
  - Recorded Date (Date)

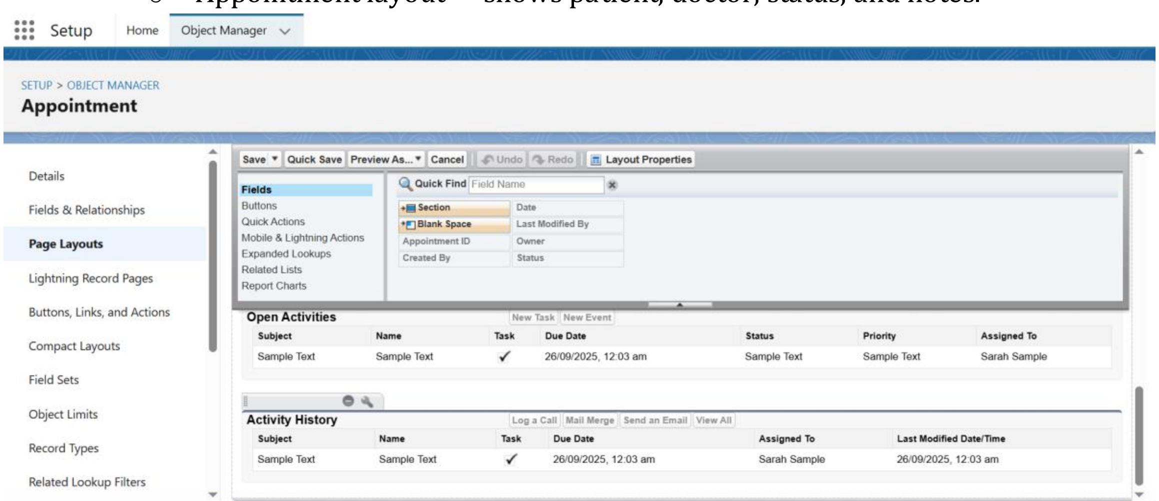




- **3. Record Types**
  - **Appointment Object** → Different record types created:
    - *General Checkup*
    - *Specialist Visit*
    - *Follow-up*

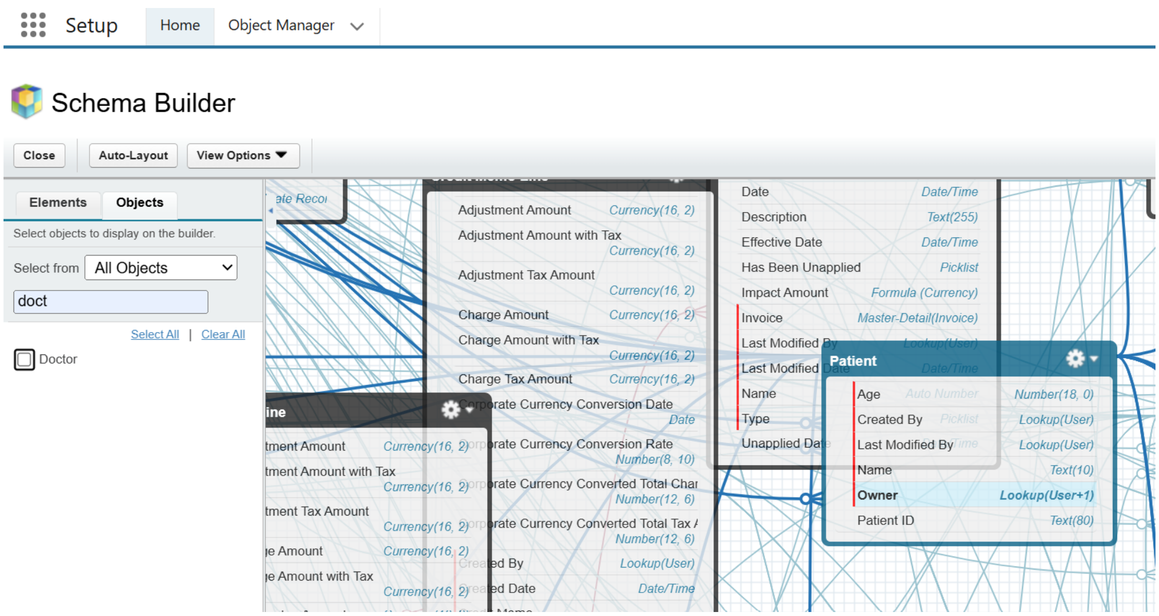


- **4. Page Layouts**
  - Customized page layouts to display relevant fields.
    - Patient layout → shows demographics + health records.
    - Doctor layout → shows specialization + experience.
    - Appointment layout → shows patient, doctor, status, and notes.



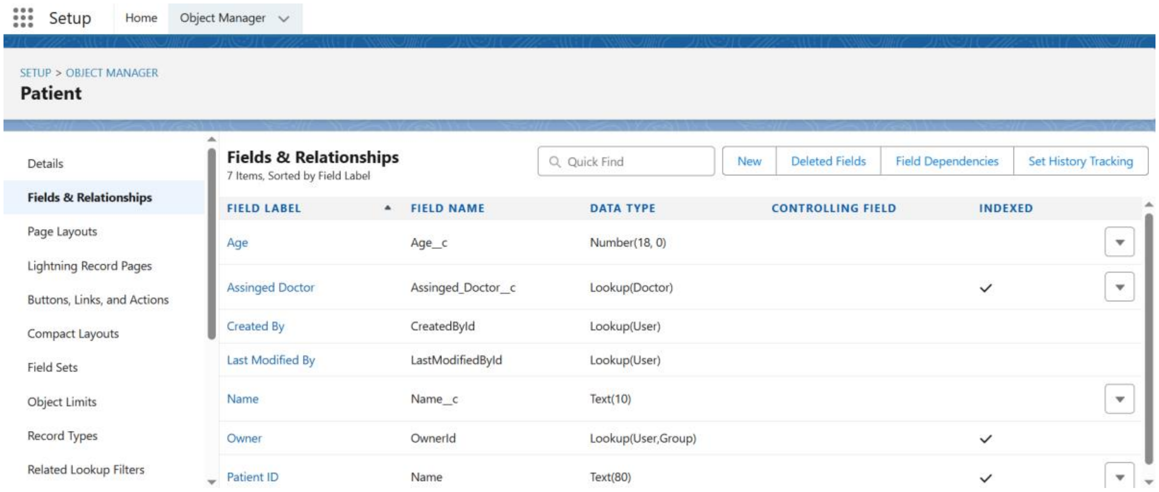
- **5. Compact Layouts**
  - Designed compact layouts to display key fields in the record header.
    - Patient: Name, Age, Gender
    - Doctor: Name, Specialization, Experience
    - Appointment: Date, Status, Patient

- **6. Schema Builder**
  - Used **Schema Builder** to visualize all relationships.
  - Objects (Patient, Doctor, Appointment, Health Metric) are linked for a clear ERD-style view.



- **7. Relationships**
  - **Lookup Relationship**
    - Appointment → Doctor
    - Appointment → Patient
  - **Master-Detail Relationship**
    - Health Metric → Patient (metrics are deleted if patient is deleted).
  - **Hierarchical Relationship**
    - Used in User object (e.g., Doctor reporting to Senior Doctor).

- **8. Junction Objects**
  - Created **Patient_Doctor_Assignment** junction object to manage **many-to-many relationship** between Patients and Doctors.
    - Example: One patient can consult multiple doctors, and one doctor can treat multiple patients.



- **9. External Objects**
  - Configured an **External Object** (Health Reports) to integrate with external health device data (for simulation).