**CareConnect: Government Health Camp CRM**

Project Overview

Build a Salesforce CRM application to manage end-to-end scheduling, outreach, and follow-up for government health camps. The system will be used by local health camp coordinators, medical teams, and government supervisors:

* Citizens can pre-register or walk in and be assigned time slots.
* Staff and equipment are allocated based on camp size and patient needs.
* Automated reminders, status tracking, and integrated feedback collection improve participation and care quality.

Objectives

* Automate scheduling for patients, doctors, and equipment to maximize efficiency.
* Provide real-time reporting to supervisors on registrations, resource allocation, and follow-ups.
* Enhance patient experience with automated reminders and streamlined onsite check-in.
* Track outcomes and gather feedback for future improvement, ensuring accountability.
* Integrate workflows with WhatsApp/SMS for communication, making it practical for mass outreach.
* **Phase 1: Problem Understanding & Industry Analysis**

**1. Introduction**

This document outlines the Phase 1 activities for the **CareConnect: Government Health Camp CRM** project. This phase focuses on understanding the core problem, analyzing stakeholder needs, mapping business processes, and researching industry use cases specific to government health camps.

**2. Problem Statement**

Government-run health camps in rural and semi-urban areas face challenges including:

* Poor patient scheduling leading to long wait times and overcrowding.
* Inefficient utilization of medical staff and equipment resources.
* Limited follow-up on patient health outcomes post-camp.
* Lack of automated reminders, reporting, and feedback mechanisms.

These challenges reduce the impact and efficiency of government health outreach programs.

**3. Objectives**

* Automate scheduling for patients, medical staff, and equipment to maximize resource utilization.
* Provide a streamlined registration and appointment system for citizens.
* Enable real-time tracking and reporting for health camp coordinators and government supervisors.
* Improve patient engagement through automated notifications and feedback.
* Support data-driven decision-making for future health camps.

**4. Stakeholder Analysis**

|  |  |  |
| --- | --- | --- |
| Stakeholder | Role | Needs/Expectations |
| Citizens / Patients | Health camp attendees | Easy registration, appointment slots, reminders |
| Health Camp Coordinator | Camp organizers | Efficient scheduling, resource allocation |
| Medical Staff | Doctors, nurses, technicians | Clear appointment lists, staff assignments |
| Government Supervisors | Monitoring and reporting | Comprehensive reports, camp performance data |

**5. Business Process Mapping**

Current and Proposed Workflows:

|  |  |
| --- | --- |
| Current Workflow | Proposed Workflow |
| Manual patient registration at camp | Online pre-registration and appointment booking |
| Walk-in camps leading to long wait times | Scheduled appointments managing patient flow |
| Staff and equipment assigned manually | Automated resource allocation based on demand |
| Limited follow-up and feedback | Automated follow-up notifications and surveys |

**6. Industry-Specific Use Case Analysis**

* Healthcare outreach programs globally use CRM systems to improve appointment management and resource allocation.
* Government agencies benefit from automation to handle large-scale camp operations.
* Appointment scheduling and follow-up improve patient adherence and health outcomes.
* Examples include vaccination camps, eye camps, and general health check-up camps utilizing streamlined CRM systems.

**7. AppExchange Exploration**

Explored Salesforce AppExchange for existing healthcare and appointment scheduling solutions:

* Identified apps focused on patient management, scheduling automation, and resource management.
* No single app fully addressed the needs of government health camps in rural areas.
* Decision made to develop a custom CRM system tailored to specific requirements.
* **Phase 2: Org Setup & Configuration**

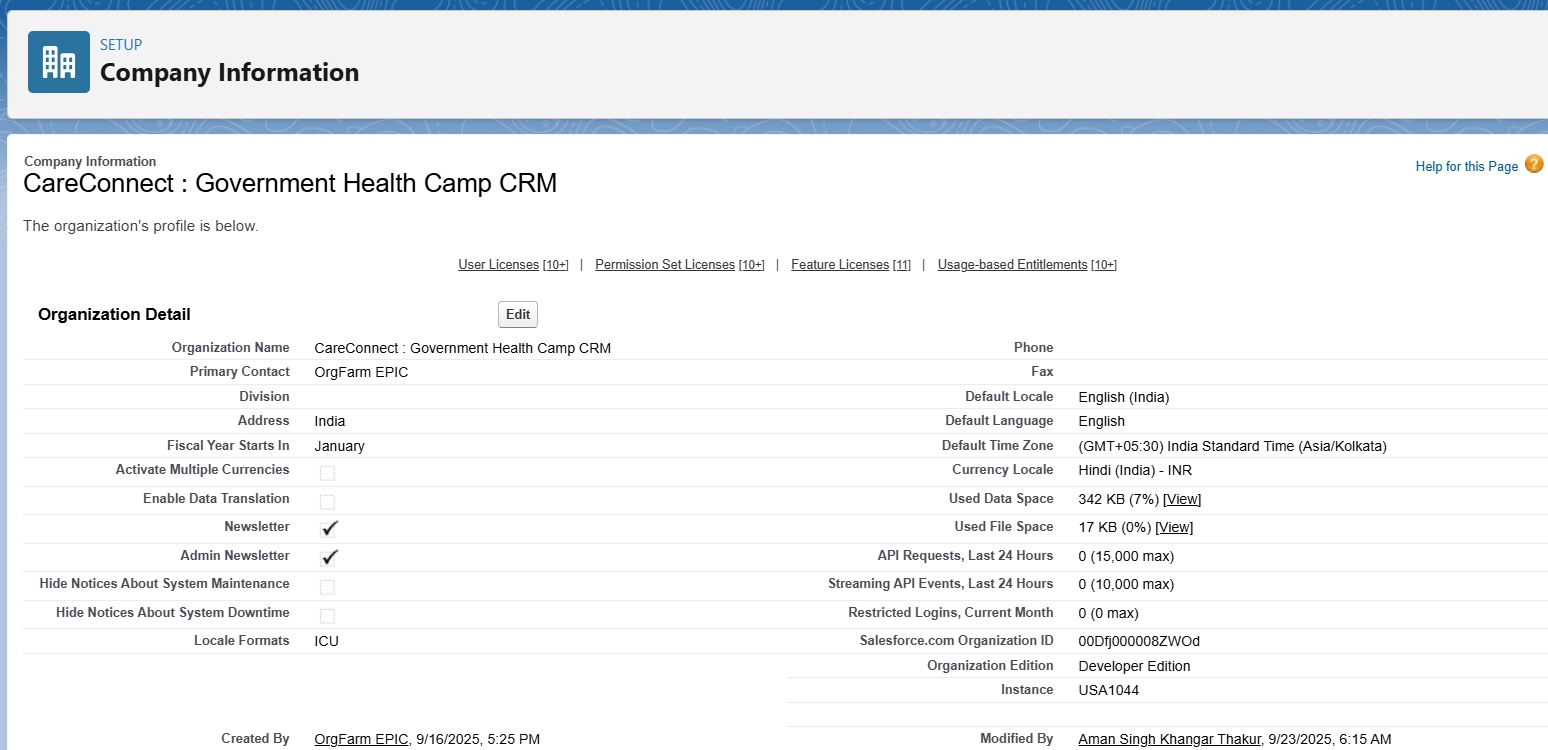
1. **Introduction**

This document describes the steps taken during Phase 2 for the **CareConnect: Government Health Camp CRM** project. The focus in this phase is to configure the Salesforce org with company settings, user management, security, and basic environment setup required to implement the project.

**2. Salesforce Edition Selection**

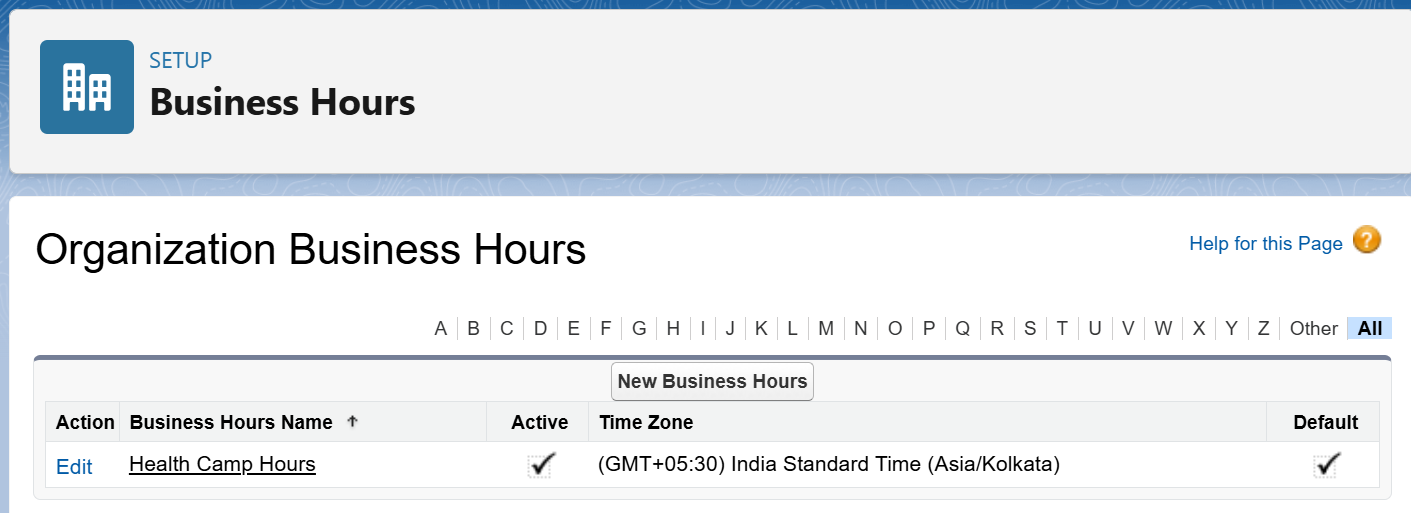
* Selected **Salesforce Developer Edition** for flexible development and testing.
* Ensured availability of necessary features for healthcare management like Custom Objects, Profiles, Role Hierarchy, Process Automation.

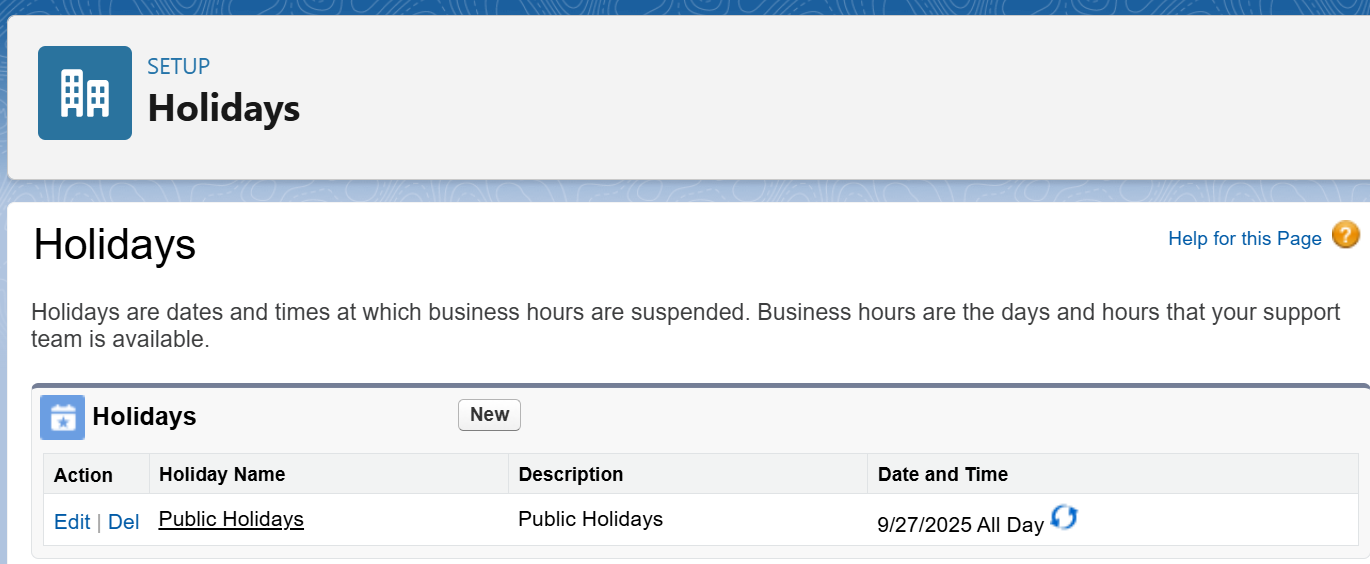
**3. Company Profile Setup**

* Accessed **Setup → Company Information**.
* Updated organization name to “CareConnect Health Camp CRM”.
* Configured default locale as Indian Locale (English – India).
* Set currency to INR (Indian Rupee).
* Populated other basic details as per project requirements.

**4. Business Hours & Public Holidays**

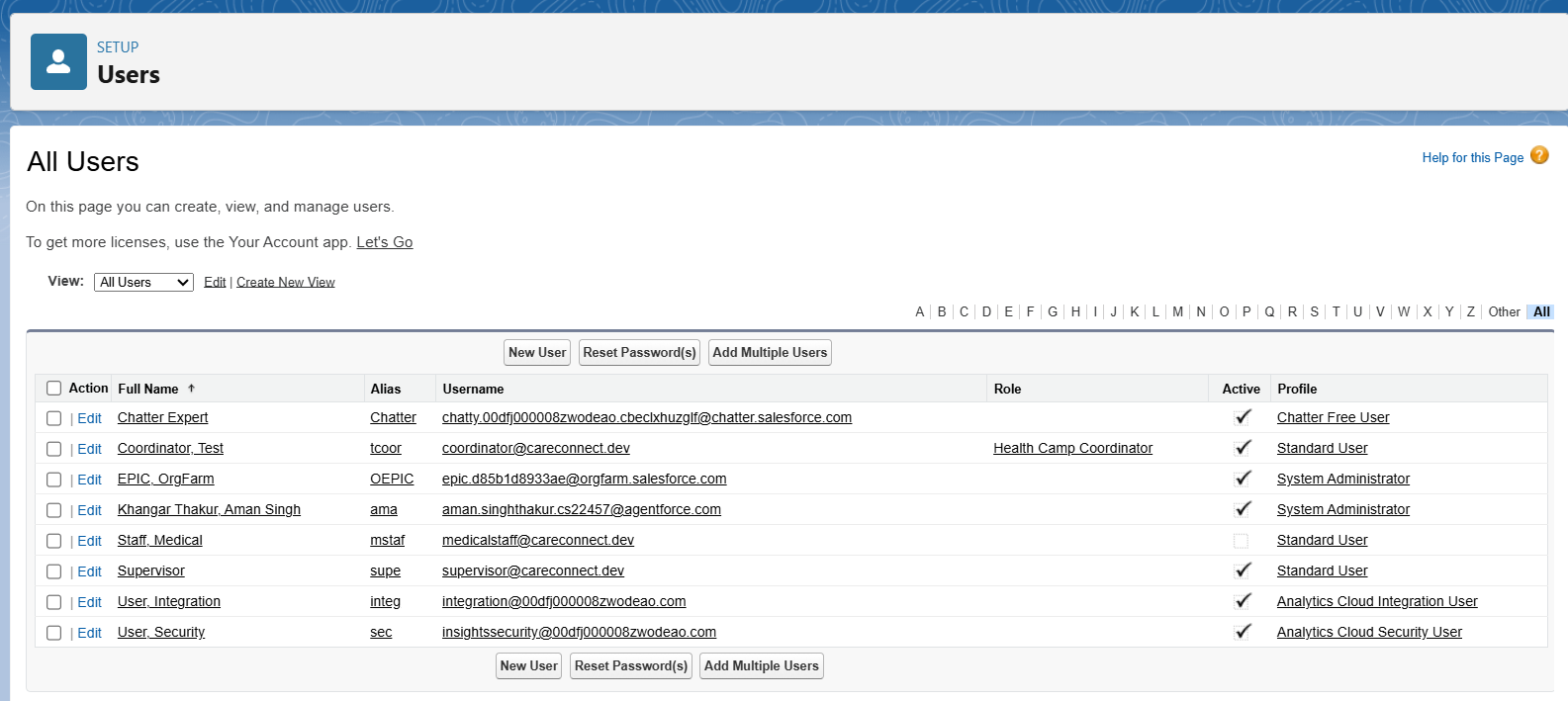
* Defined **Business Hours** to match health camp operational timings (9 AM – 5 PM, Monday to Saturday).
* Added relevant **Holidays** to block inactive days during which no camps operate.





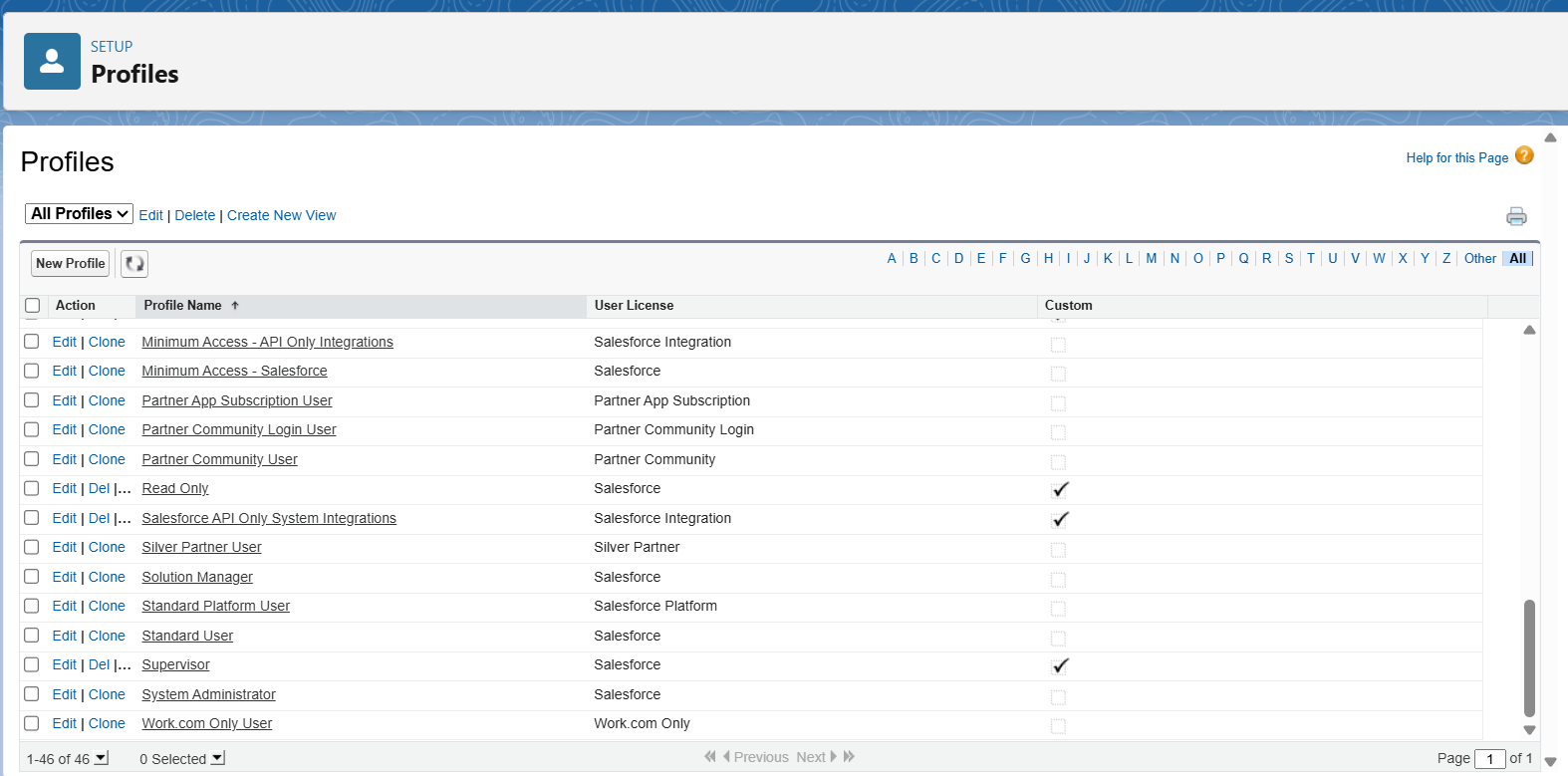
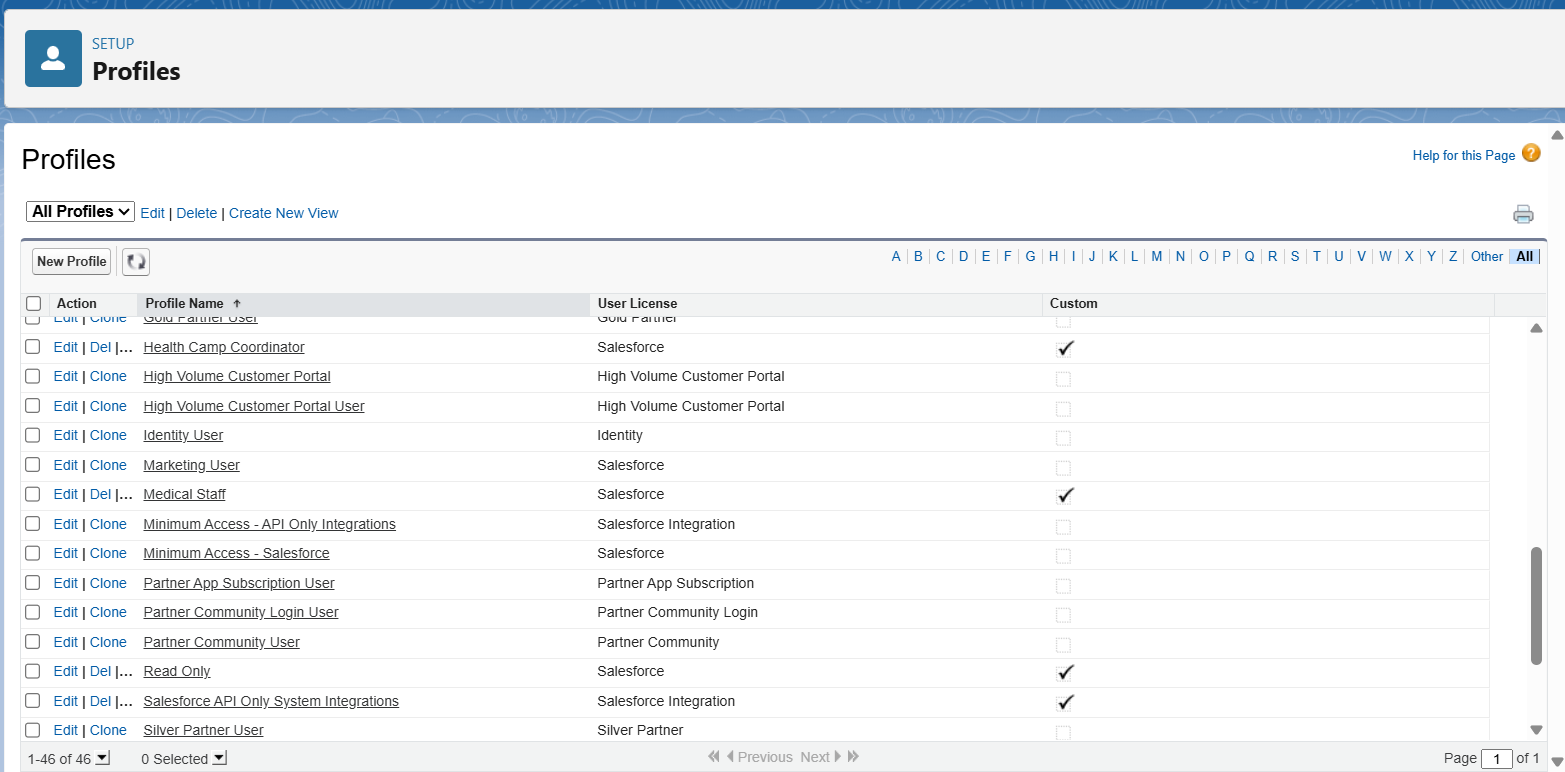
**5. User Setup & Licenses**

* Created users for roles:
  + Health Camp Coordinator
  + Medical Staff
  + Government Supervisor
* Assigned relevant **Salesforce Licenses** (Salesforce Platform License or Salesforce License).



**6. Profile Management**

* Cloned **Standard User Profile** to create custom profiles tailored for:
  + **Health Camp Coordinator**: Full access to camp and appointment data.
  + **Medical Staff**: Limited access to assigned appointments and patient records.
  + **Government Supervisor**: Read-only access to all data for monitoring.

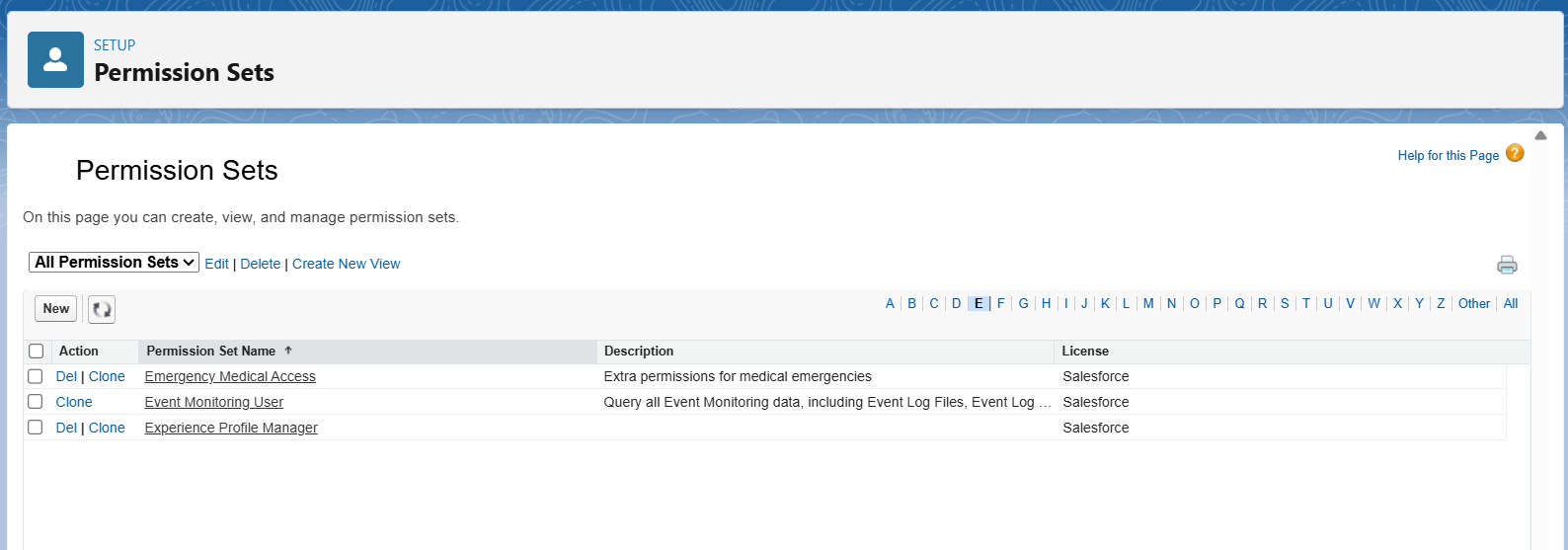
****

**7. Role Hierarchy**

* Created role hierarchy for data visibility:
  + **Government Supervisor** at the top.
  + **Health Camp Coordinator** midway.
  + **Medical Staff** at the bottom.
* A screenshot of a computer

  AI-generated content may be incorrect.This hierarchy ensures supervisors see all records, coordinators see their camps, and staff see their tasks.

**8. Permission Sets**

* Defined permission sets when additional privileges were required beyond profile permissions.
* Examples include access to reports, special admin tasks, or custom app features.

**9. Organization-Wide Defaults (OWD) and Sharing Rules**

* Set **OWD** for sensitive records (like Contacts/Patients) to **Private** for privacy.
* Created **Sharing Rules**:
  + Open access to Health Camp Coordinators for their own camps.
  + A screenshot of a computer

    AI-generated content may be incorrect.Restricted sharing for Medical Staff to only their records.

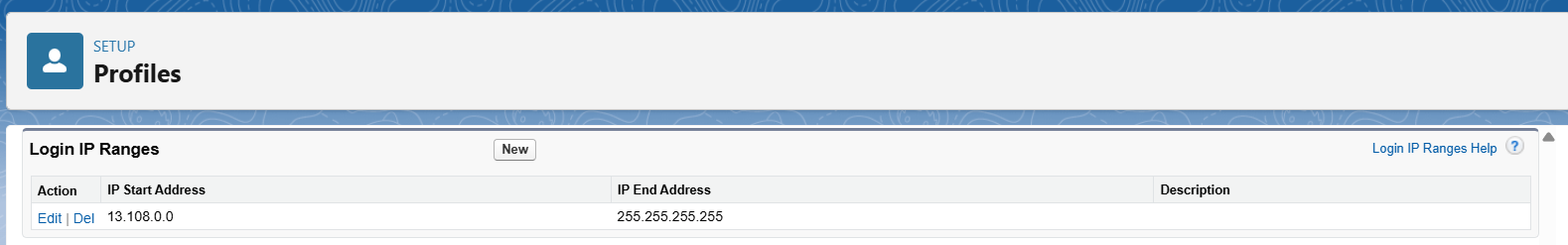
A screenshot of a computer

AI-generated content may be incorrect.

**10. Login Access & Security Policies**

* Configured **Login IP Ranges** for users as per organizational policy.
* A screenshot of a computer

  AI-generated content may be incorrect.Set **Session Timeout** and **Login Hours** as appropriate for security.



* **Phase 3: Data Modeling & Relationships**

**1. Introduction**

This document outlines Phase 3 of the CareConnect Government Health Camp CRM project. The focus is on creating an effective data model within Salesforce, using custom and standard objects, defining key fields, and establishing appropriate object relationships to reflect real-world government health camp operations.

**2. Objectives**

* Build custom objects representing Health Camps, Appointments, Medical Staff, and Equipment.
* Define data fields using the right data types for accurate capture.
* Implement relationships between objects for data integrity and ease of use.
* Customize Page and Compact Layouts for better UX.
* Visualize the data model using Salesforce Schema Builder.

**3. Custom Objects and Field Definitions**

**Health Camp (Health\_Camp\_\_c)**

* Camp Location (Text, Length 100)
* Camp Date (Date)
* Max Capacity (Number, 4 digits)
* Status (Picklist: Planned, Active, Completed, Cancelled)

**Appointment (Appointment\_\_c)**

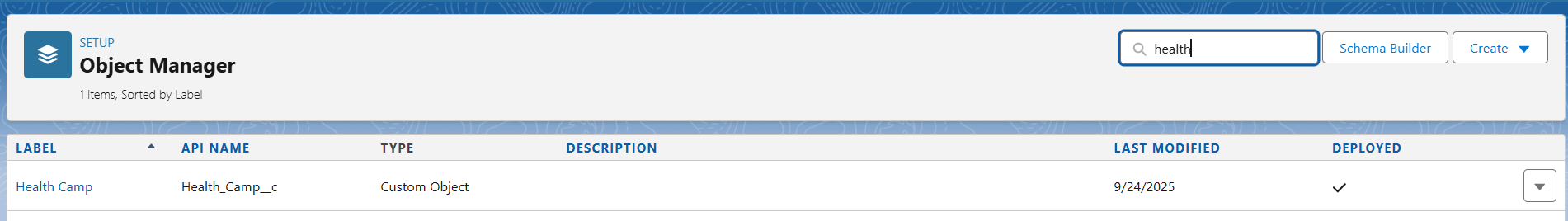
* Appointment Date/Time (DateTime)
* Status (Picklist: Scheduled, Completed, Cancelled, No Show)
* Patient (Lookup to Contact)
* Health Camp (Lookup to Health\_Camp\_\_c)

**Medical Staff (Medical\_Staff\_\_c)**

* Staff Name (Text)
* Role (Picklist: Doctor, Nurse, Support Staff)
* Assigned Health Camp (Lookup to Health\_Camp\_\_c)

**Equipment (Equipment\_\_c)**

* Equipment Name (Text)
* Equipment Type (Picklist: Device, Supply)
* Assigned Health Camp (Lookup to Health\_Camp\_\_c)



A close up of a computer screen

AI-generated content may be incorrect.

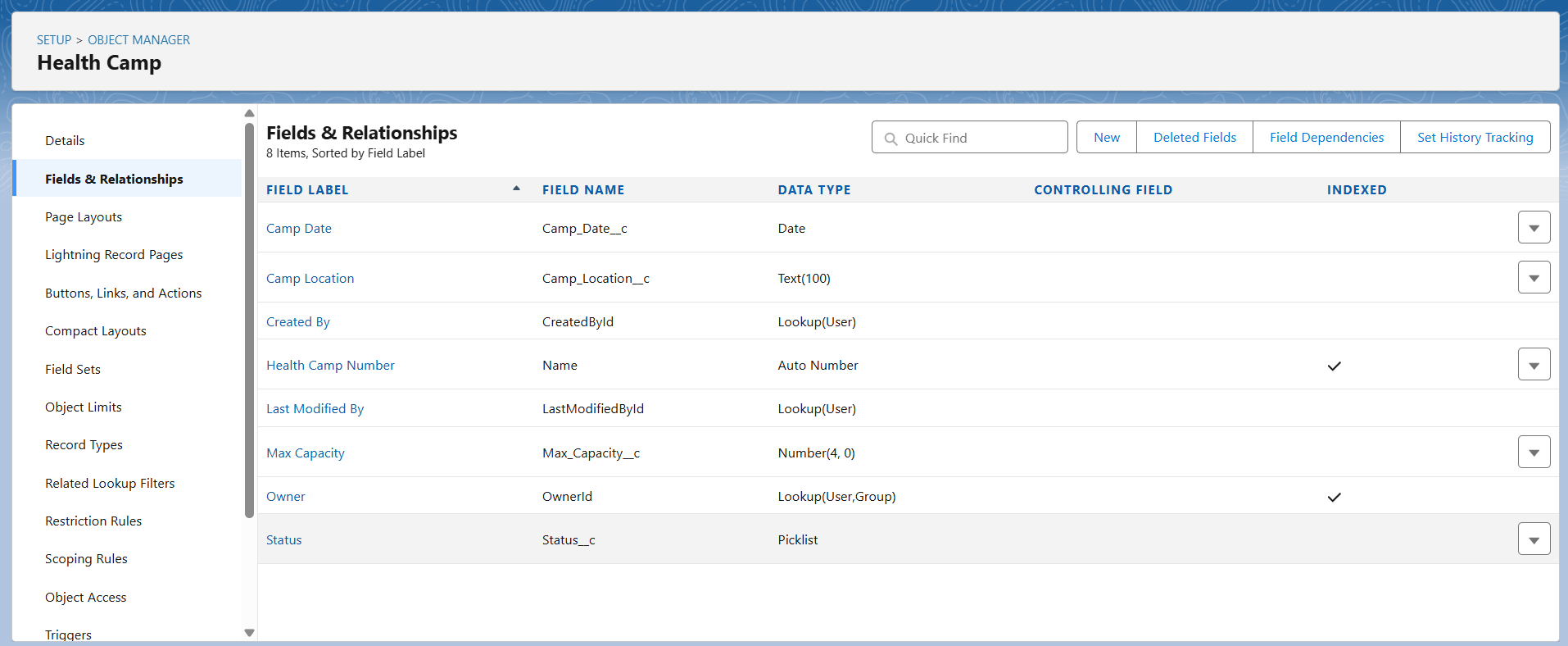
A blue and white rectangular object

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**A screenshot of a computer

AI-generated content may be incorrect.**

**4. Relationships**

**Lookup vs Master-Detail vs Hierarchical Relationships**

* **Lookup Relationship:**  
  A loose link between two objects. The child record can exist independently of the parent. Deleting the parent doesn't delete the child. Flexible but less tightly coupled.
* **Master-Detail Relationship:**  
  Strong binding where the child depends on the parent. Deleting the parent cascades delete to the child. Security and sharing settings are inherited.
* **Hierarchical Relationship:**  
  A special type of lookup relationship only available on the User object to represent user-to-user relationships (e.g., manager to team member).

**Usage in CareConnect:**

* Appointment to Patient (Contact) and Health Camp: Lookup Relationship
* A screenshot of a computer

  AI-generated content may be incorrect.Medical Staff and Equipment to Health Camp: Lookup Relationship (can be master-detail if you want cascade delete and strong binding)

**5. Junction Objects**

* Used to model **many-to-many relationships**. For example, when medical staff can be assigned to multiple health camps and a health camp has multiple staff members.
* Create a custom junction object with **two master-detail relationships** pointing to the two parent objects.

A screenshot of a computer

AI-generated content may be incorrect.

**6. External Objects**

* No external objects implemented in Phase 3.

**7. Page Layouts and Compact Layouts**

* Customize layouts to display key fields clearly.
* Add related lists for child records to parent pages (e.g., show appointments in Health Camp record)
* A screenshot of a computer

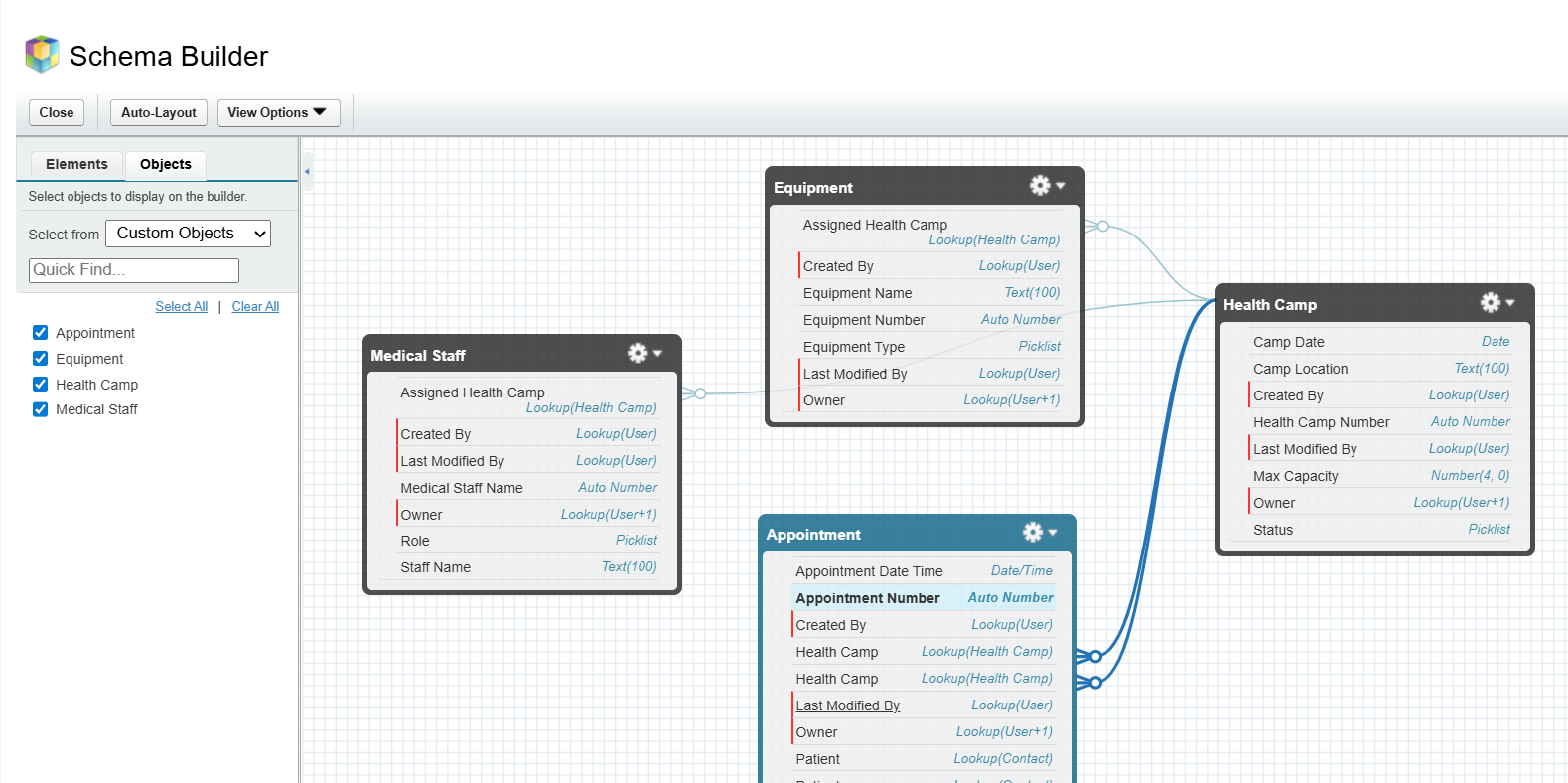
  AI-generated content may be incorrect.Define compact layouts for summary views in record highlights.

A screenshot of a computer

AI-generated content may be incorrect.

**8. Schema Builder Visualization**

* Use Salesforce Schema Builder to visually design and verify object fields and relationships.
* Adjust data model as needed after visual validation.



* **Phase 4: Process Automation**

**1. Introduction**

Phase 4 focuses on applying automation within the CareConnect CRM to improve efficiency and accuracy. This document outlines the tools and processes implemented to streamline health camp management workflows.

**2. Objectives**

* Automate validation rules to ensure quality of input data and prevent errors.
* Use workflow rules and process builder for sending timely email alerts, tasks, and field updates.
* Develop flows (screen, record-triggered, scheduled, auto-launched) for multi-step and automated processes.
* Configure approval processes where managerial confirmations are required.
* Send custom notifications to users and stakeholders for timely action.

**3. Validation Rules**

* Validated mandatory fields for appointment scheduling.
* Prevented invalid date/time entries for new appointments.
* Enforced appointment cancellation reasons.

A screenshot of a computer

AI-generated content may be incorrect.

**4. Workflow Rules**

* Triggered patient registration email notifications.
* Automated updating of camp capacity on new or cancelled appointments.
* A screenshot of a computer

  AI-generated content may be incorrect.Created reminder tasks for staff prior to camp start.

**5. Process Builder**

* Automated appointment status changes based on attendance.
* Synchronized related record updates such as patient records linked with appointments.
* Managed equipment availability updates on assignment.

A screenshot of a computer

AI-generated content may be incorrect.

**6. Flows**

* Screen Flow implemented for guided appointment booking for coordinators.
* Record-Triggered Flows used for real-time updates on appointment and camp records.
* Scheduled Flow created for automated health camp performance reports sent to supervisors.
* Auto-launched Flows used behind the scenes for field validation and data calculations.

A screenshot of a computer

AI-generated content may be incorrect.

**7. Approval Processes**

* Multi-level approvals configured for critical changes like camp cancellations and staff assignments.
* Email notifications set for supervisors on approval requests.
* A screenshot of a computer

  AI-generated content may be incorrect.Rejection comments and audit logging enabled.

**8. Email Alerts, Tasks & Notifications**

* Personalized email templates for patient confirmations and reminders.
* Task assignments automated for medical staff campaign preparations.
* Custom notifications sent to supervisors and coordinators for urgent camp updates.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

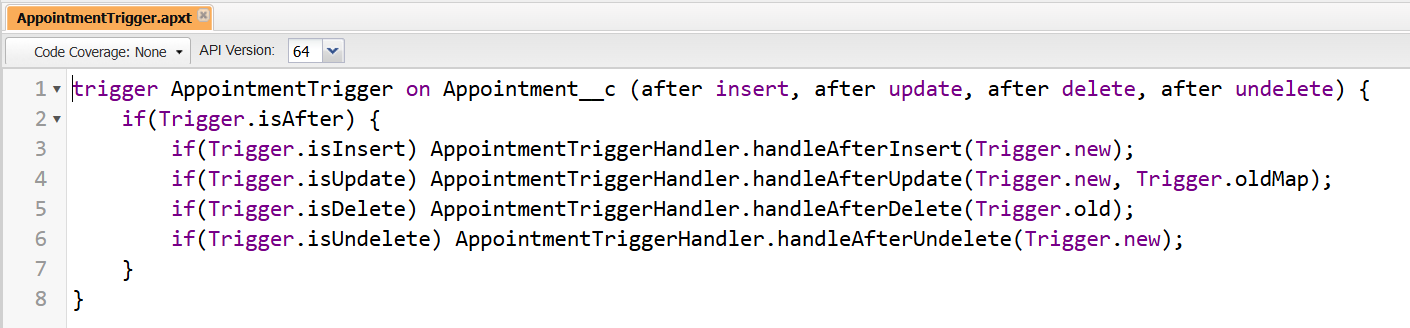
* **Phase 5: Apex Programming**

In Phase 5, I implemented advanced Apex programming to extend automation capabilities that go beyond Flows and Process Builder. Below is the structured documentation of what I built, with supporting details.

**1. Classes & Objects**

* Implemented custom Apex classes to encapsulate business logic for appointments and health camps.
* Classes improve reusability and separation of concerns.
* Supporting point: This ensures logic can be reused by triggers, test classes, and future integrations.

**2. Apex Triggers (before/after insert/update/delete)**

* Developed triggers on the Appointment object to handle slot allocation for Health Camps.
* Used both after insert and after update to manage available slots dynamically.
* Supporting point: Automates record updates without manual intervention.

**3. Trigger Design Pattern**

* Adopted the **trigger handler pattern** by separating logic into a dedicated handler class.
* Keeps trigger lightweight and only responsible for event routing.
* Supporting point: Enhances maintainability and prevents code duplication.

A screenshot of a computer program

AI-generated content may be incorrect.

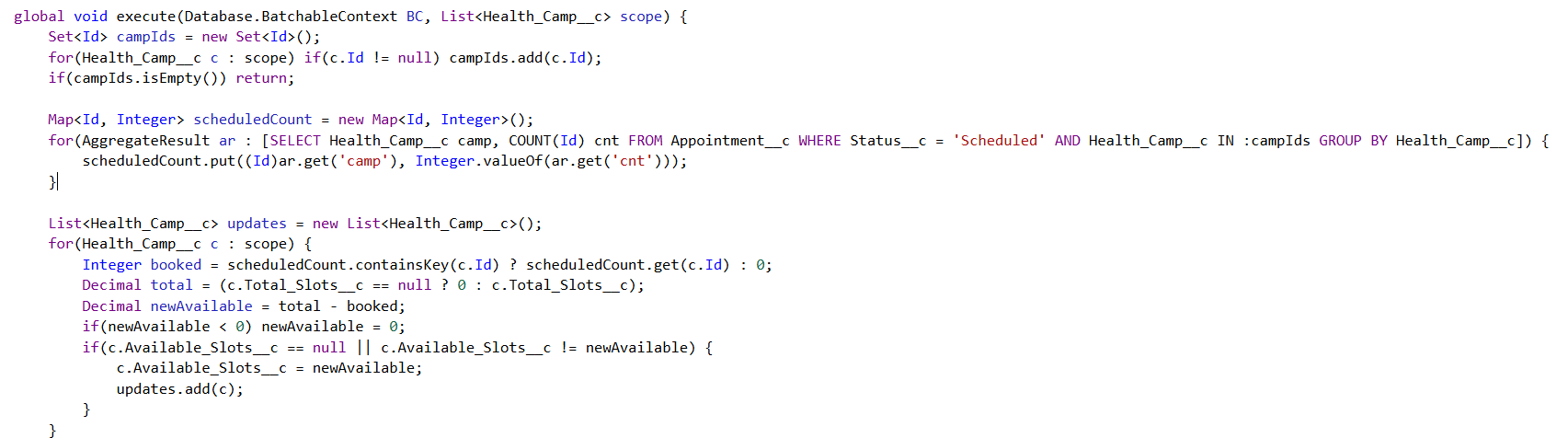
**4. SOQL & SOSL**

* Implemented SOQL queries to fetch Appointment and Health Camp records efficiently.
* Used selective queries with filters to comply with governor limits.
* Supporting point: Provides structured and controlled access to Salesforce data.
* No SOSL required for project use cases.

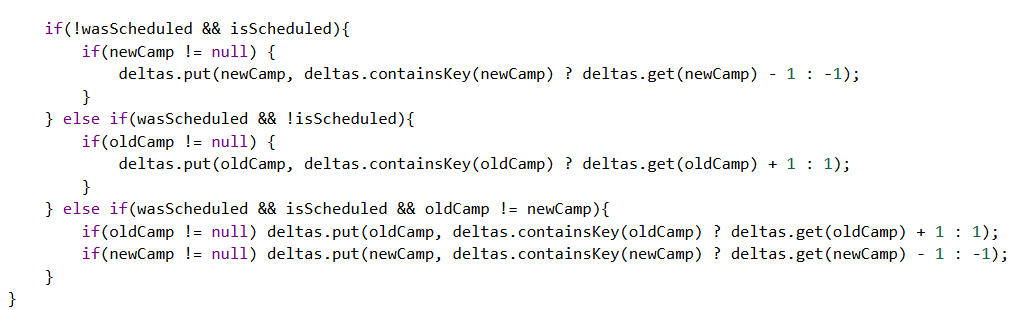
A white background with colorful text

AI-generated content may be incorrect.

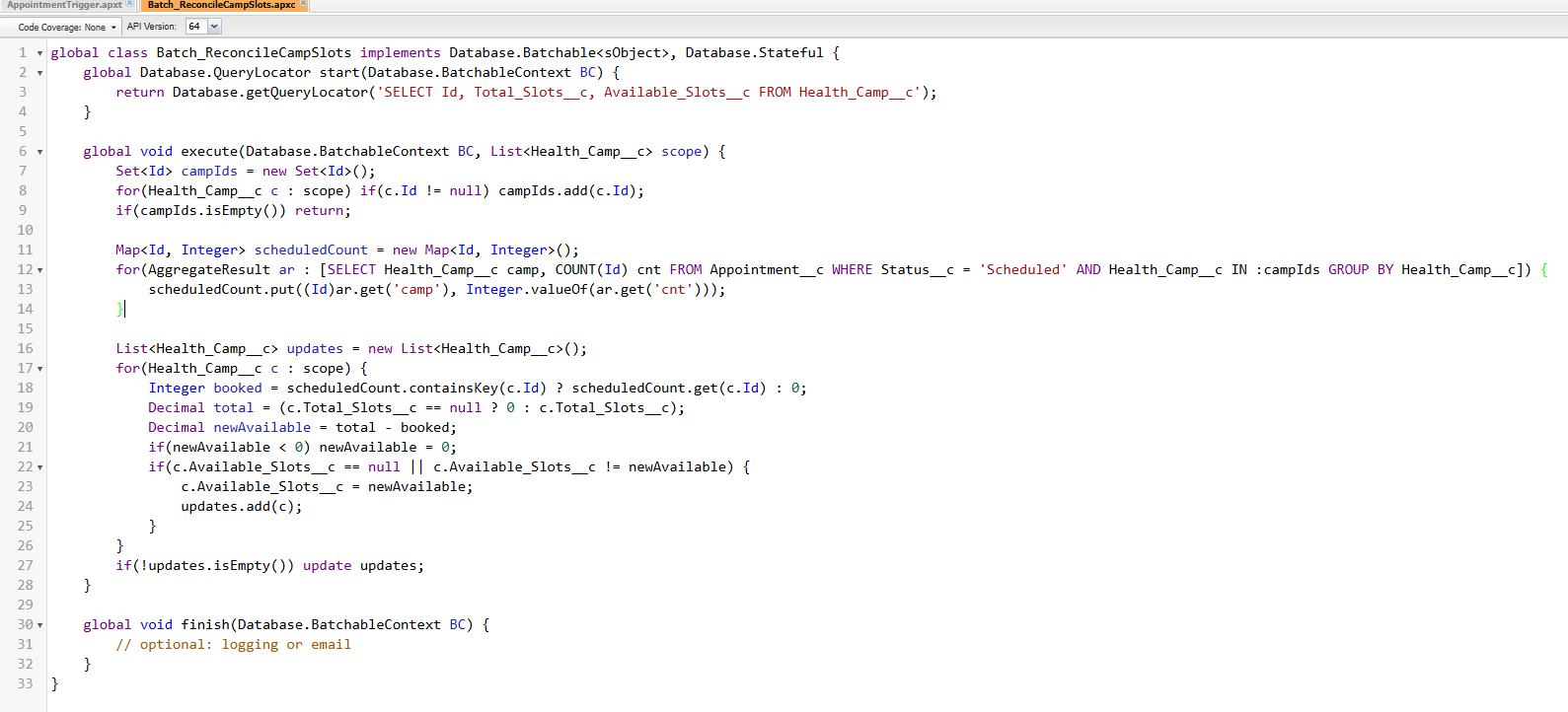
**5. Collections: List, Set, Map**

* Applied Lists to store multiple appointments.
* Used Sets for unique record IDs to prevent duplication.
* Implemented Maps for quick record lookups in trigger logic.
* Supporting point: Optimized performance and ensured data integrity.

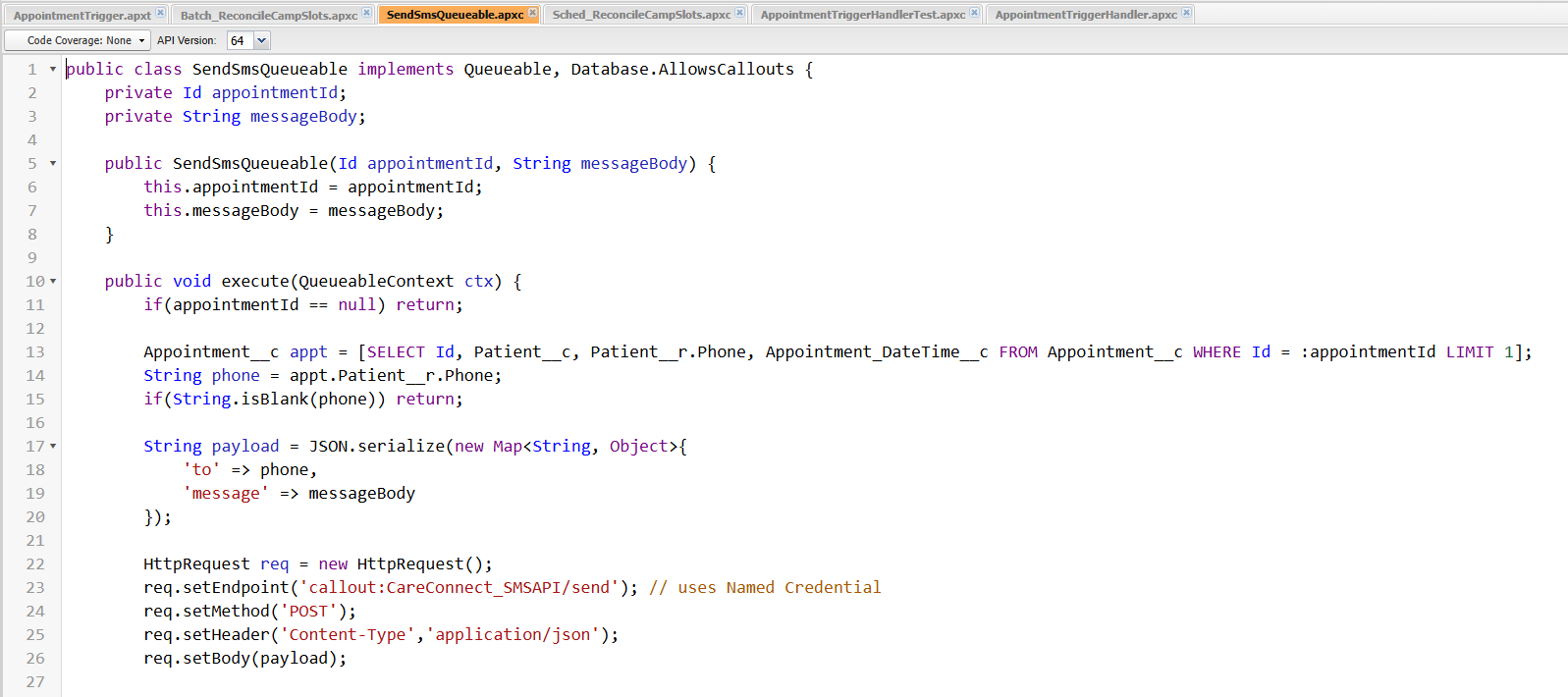
**6. Control Statements**

* Incorporated if-else conditions for slot management (Scheduled, Cancelled, Deleted).
* Used for loops to iterate over record collections in bulk operations.
* Supporting point: Enabled decision-making and control flow in automation.

**7. Batch Apex**

* Created a Batch Apex class for reconciling Health Camp slots in bulk.
* Processes large volumes of data asynchronously without hitting governor limits.
* Supporting point: Ensures data consistency during large-scale updates.

**8. Queueable Apex**

* Implemented Queueable Apex to send notifications to patients asynchronously.
* Allows job chaining for complex workflows.
* Supporting point: Offers flexibility and reliability for background processes.

**9. Scheduled Apex**

* Built Scheduled Apex to run the Batch reconciliation process at defined intervals (e.g., nightly).
* Supporting point: Provides automated scheduling without manual runs.

A screenshot of a computer

AI-generated content may be incorrect.

**10. Future Methods**

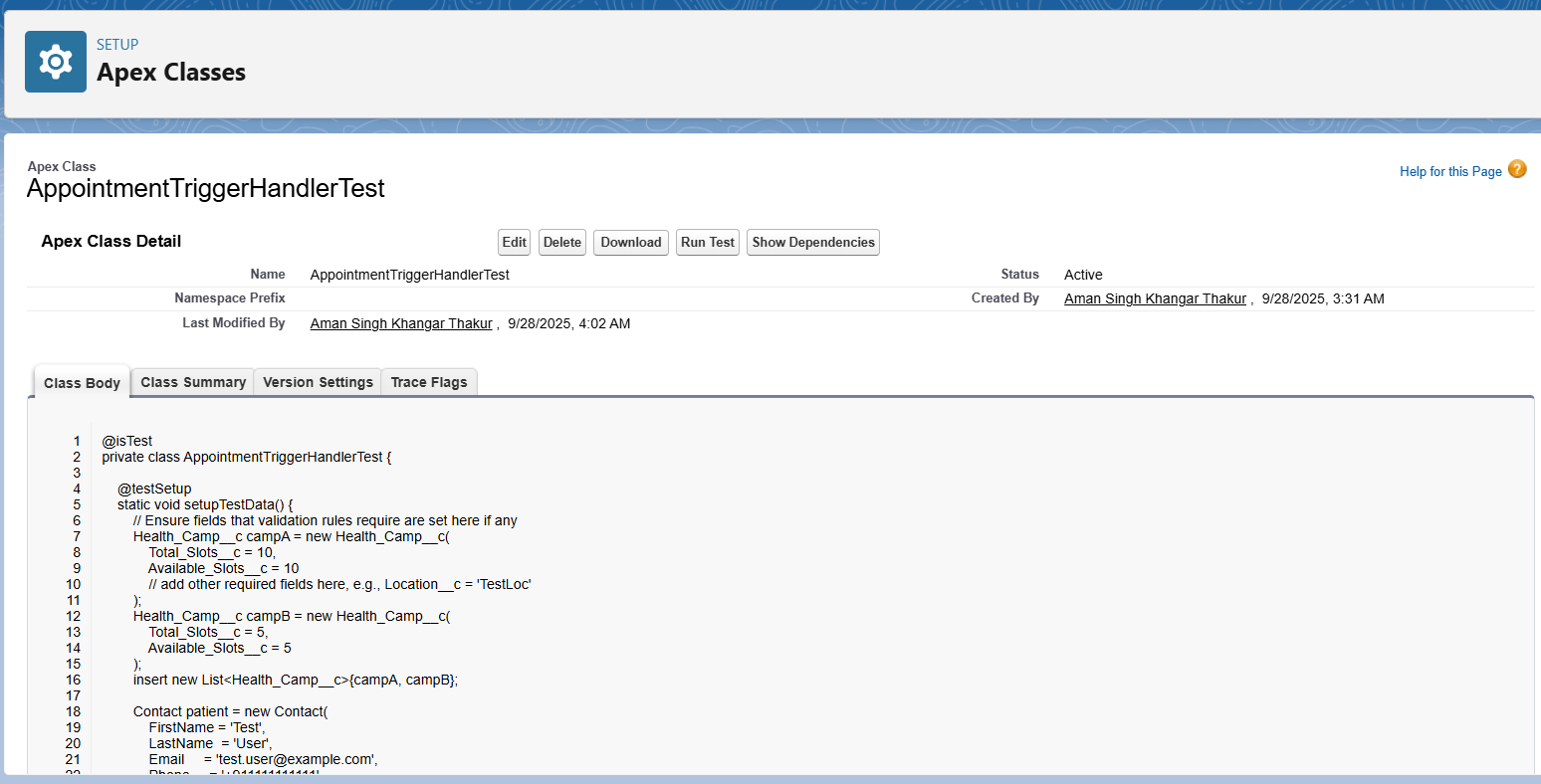
* Used @future methods to handle callouts and lightweight async tasks.
* A computer screen shot of a computer screen

  AI-generated content may be incorrect.Supporting point: Decouples non-critical operations from the main transaction, improving performance.

**11. Exception Handling**

* Added try-catch blocks in Apex classes to handle unexpected errors.
* Logged failures for debugging and monitoring.
* Supporting point: Ensures smooth execution and prevents transaction rollbacks.

**12. Test Classes**

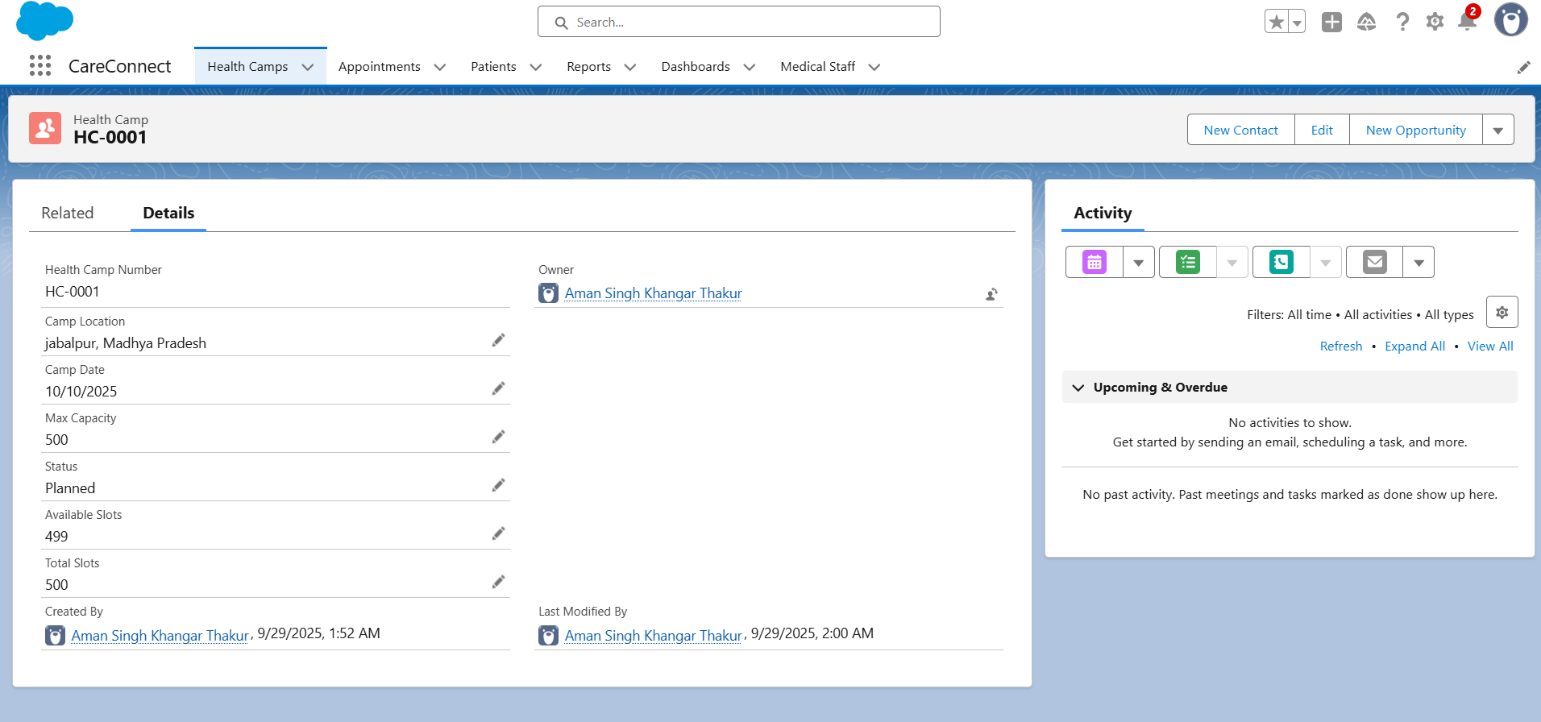
* Developed comprehensive test classes for all Apex triggers and classes.
* Achieved required code coverage (>75%) for deployment readiness.
* Supporting point: Guarantees reliability of business logic and validates all possible scenarios.

**13. Asynchronous Processing**

* Implemented multiple async approaches: Batch Apex, Queueable Apex, Scheduled Apex, and Future methods.
* Supporting point: Handles high-volume operations efficiently and ensures Salesforce governor limits are respected.
* **Phase 6: UI Development**

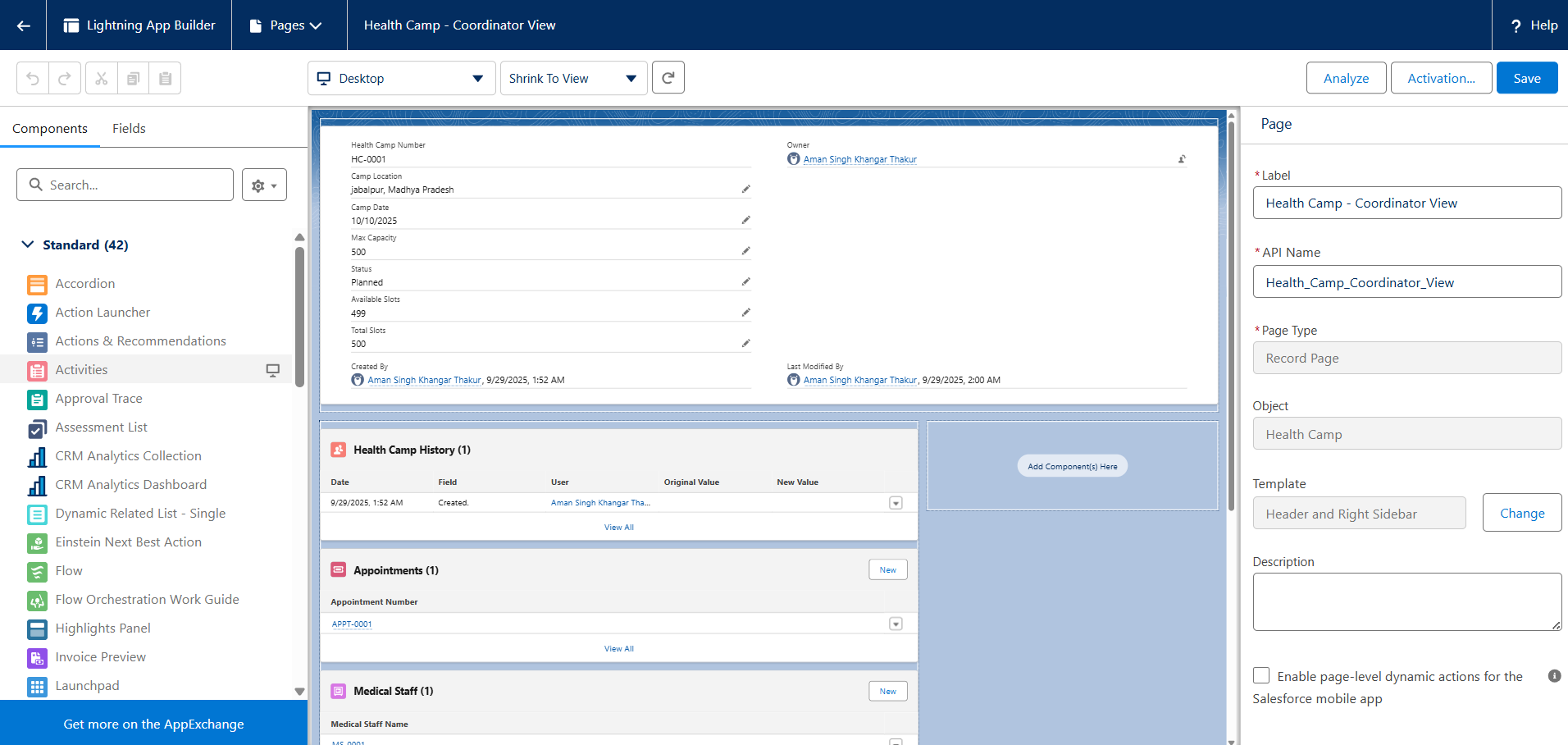
1. **Lightning App Builder**

* Created the **CareConnect Lightning App** for Health Camp management.
* Added navigation items: Health Camp, Appointment, Patient, Reports, and Dashboards.
* Configured app for **Desktop and Mobile** access.
* Assigned the app to **Health Camp Coordinator** and **System Administrator** profiles.



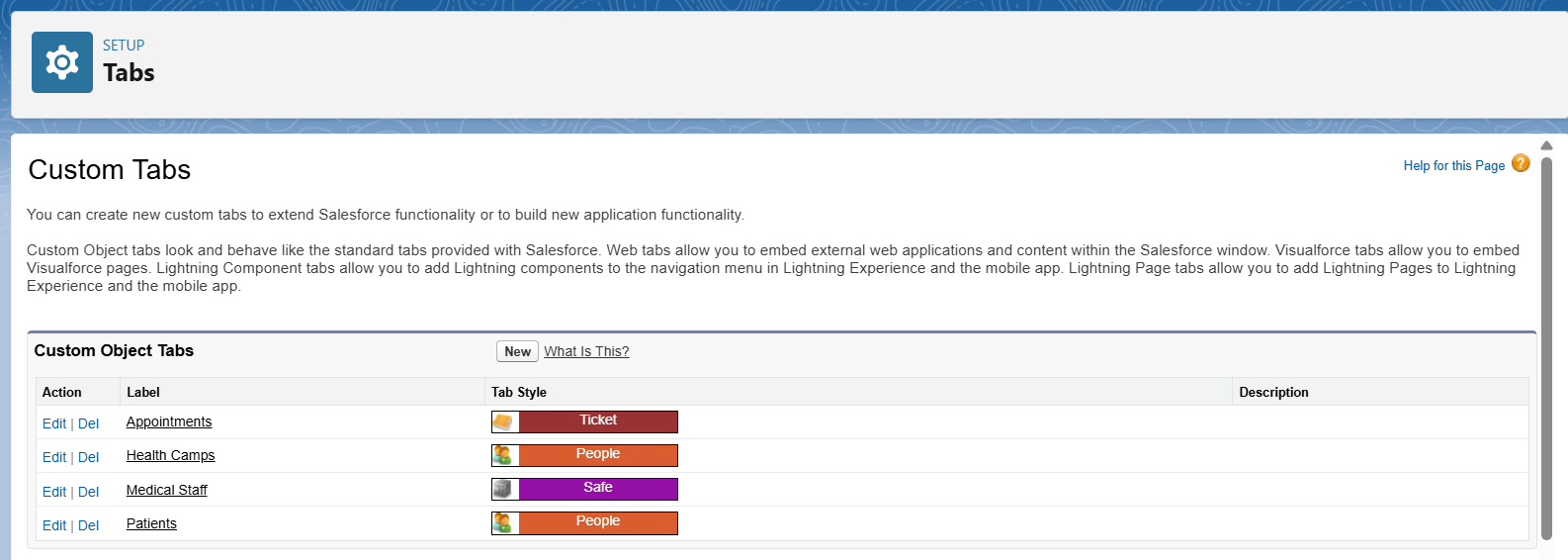
1. **Record Pages**

* Designed **custom Lightning Record Pages** for all major objects:
  + **Health Camp:** Shows details, available slots, related appointments and patients.
  + **Appointment:** Displays patient, camp details, status, and cancellation reason.
  + **Patient:** Shows demographic information and related appointments.
* Used **Header + Right Sidebar templates** for effective layout.
* Added components like **Record Detail, Related Lists, Tabs, Report Charts**.



1. **Tabs**

* Created **Custom Tabs** for:
  + Health Camp (Health\_Camp\_\_c)
  + Appointment (Appointment\_\_c)
  + Patient (Patient\_\_c)
* Assigned relevant **Tab Styles** (hospital, calendar, person icons).
* Made tabs visible to **Coordinator** and **Admin** profiles.



1. **Home Page Layouts**

* Built a **Coordinator Home Page** in Lightning App Builder.
* Added key components:
  + **Today’s Tasks** → shows pending work.
  + **Recent Records** → quick access to Health Camp and Appointment records.
  + **Report Chart** → Upcoming Camp Occupancy visualization.
  + **Rich Text Component** → Welcome note for coordinators.
* Assigned the Home Page to the **CareConnect App** and **Coordinator profile**.

A screenshot of a computer

AI-generated content may be incorrect.

1. **Utility Bar**

* Added a **Utility Bar** to the CareConnect App for quick access.
* Included utilities like:
  + **Notes** → for quick updates.
  + **History** → recent records navigation.
  + **Open CTI Softphone (optional)** → placeholder for future telephony integration.
* Configured visibility for **Coordinator profile**.

1. **LWC (Lightning Web Components)**

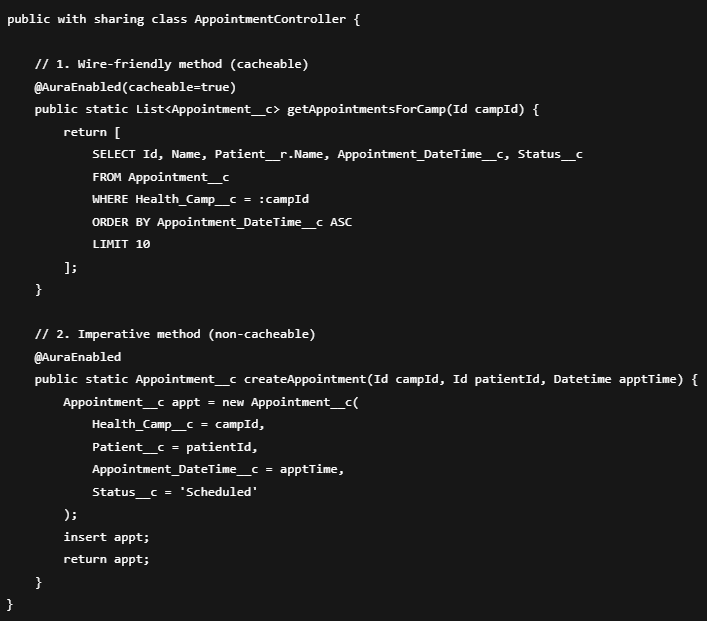
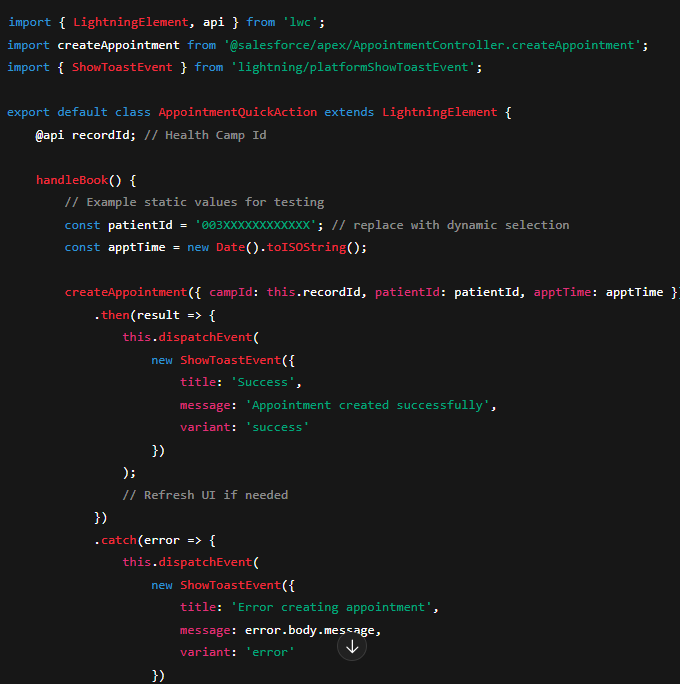
* Introduced **Lightning Web Components** to extend UI functionality.
* Created an LWC for **Health Camp Slot Availability** visualization.
* Embedded the LWC inside the **Health Camp Record Page**.

A screenshot of a computer

AI-generated content may be incorrect.

1. **Apex with LWC & Imperative Apex Calls**

* Connected LWCs to Apex Classes for dynamic server-side logic.
* Example usage in project:
  + Fetching **real-time appointment slots** from Health Camp records.
  + Pushing updates when appointments are created or cancelled.
* Ensured proper exception handling in Apex methods exposed to LWC.
* Used **imperative Apex calls** from LWC for scenarios requiring user-triggered actions.
* Example: On “Book Appointment” button click → Apex method executed to create appointment and update slot count.
* Allowed real-time confirmation messages to users.



1. **Events in LWC**

* Implemented **Custom Events** to allow communication between nested LWCs.
* Example: Updating available slots component when a new appointment is scheduled.
* Used **bubbling and composition** where necessary for parent-child coordination.

A black screen with text

AI-generated content may be incorrect.

1. **Wire Adapters**

* Used **@wire** with Apex methods to fetch Health Camp and Appointment data.
* Automatically updated UI when record data changed.
* A screen shot of a computer program

  AI-generated content may be incorrect.Example: Appointment status updates reflected instantly on the Appointment Record Page.

1. **Navigation Service**

* Used **Lightning Navigation Service** to enable smooth transitions between records.
* Example implementations:
  + Navigating from Patient record → Appointment record.
  + Redirecting users to **Health Camp Record** after booking appointment.
* Improved user experience by avoiding manual search/navigation.

A computer screen shot of a program code

AI-generated content may be incorrect.

* **Phase 7: Integration & External Access**

1. **Named Credentials**

* Created a **Named Credential** to securely connect Salesforce with external systems.
* Used **External Credential + Principal** to store authentication details (Account SID and Auth Token from provider).
* Named Credential created with base URL https://api.twilio.com.
* This avoids storing secrets in Apex code and keeps callouts secure.



**2. External Services**

* Registered an **External Service** in Salesforce using OpenAPI specification.
* Connected the external SMS provider API to Salesforce via the Named Credential.
* The service actions became available directly in Flow as **invocable actions**.
* Implemented Flow actions for sending confirmation messages after booking appointments.
* **Usage in Apex:**

A screenshot of a computer

AI-generated content may be incorrect.

**3. Web Services (REST/SOAP)**

* Exposed Salesforce data using **standard REST API** for external integrations.
* Created a **Connected App** to generate OAuth tokens for authentication.
* Built a **custom Apex REST resource** for creating and fetching Appointment records.
* External applications can call Salesforce securely to manage appointments.



**4. Callouts**

* Implemented **Apex callouts** to send data from Salesforce to external services.
* Used the Named Credential CareConnect\_SMSAPI for endpoint authentication.
* Configured callouts to notify patients (via SMS) about appointment booking and cancellations.
* Verified callouts in Sandbox using Developer Console and debug logs.



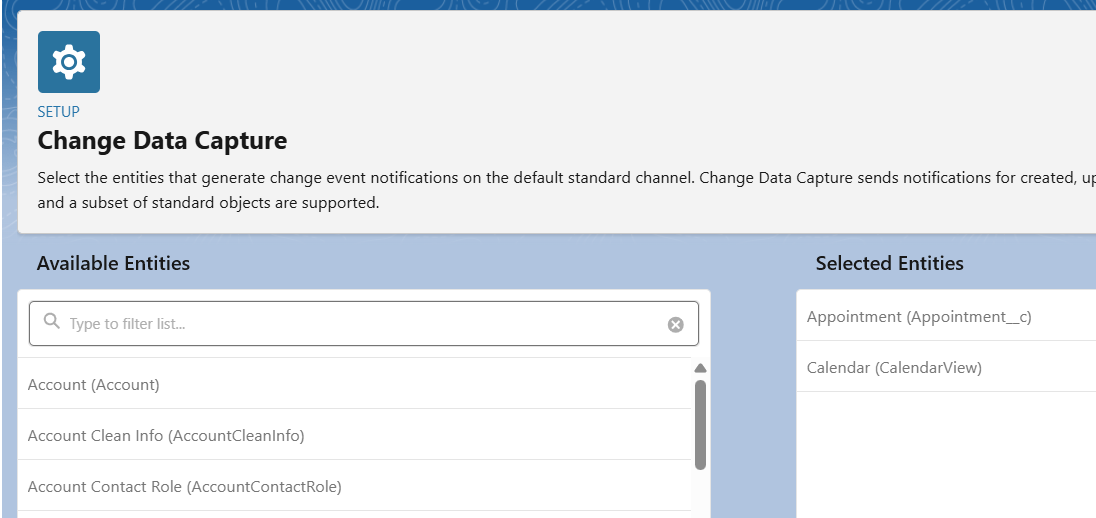
**5. Platform Events**

* Defined a **Platform Event** named Appointment\_Event\_\_e.
* Added fields: AppointmentId, EventType (Created/Cancelled), PatientPhone.
* Used Flows to **publish events** after Appointment creation or update.
* A screenshot of a computer

  AI-generated content may be incorrect.Subscribed to the events with Flows and Apex to trigger external notifications.

**6. Change Data Capture**

* Enabled **Change Data Capture** for Appointment\_\_c.
* This allows external systems to subscribe to changes (create/update/delete).
* Ensures real-time data sync without heavy API polling.
* Used CDC to keep analytics dashboards updated with appointment changes.



**7. Salesforce Connect**

* Configured **Salesforce Connect** with OData for viewing external health records.
* Created an External Data Source named CareConnect\_EHR\_OData.
* Validated and synced external tables as **External Objects**.
* A screenshot of a computer

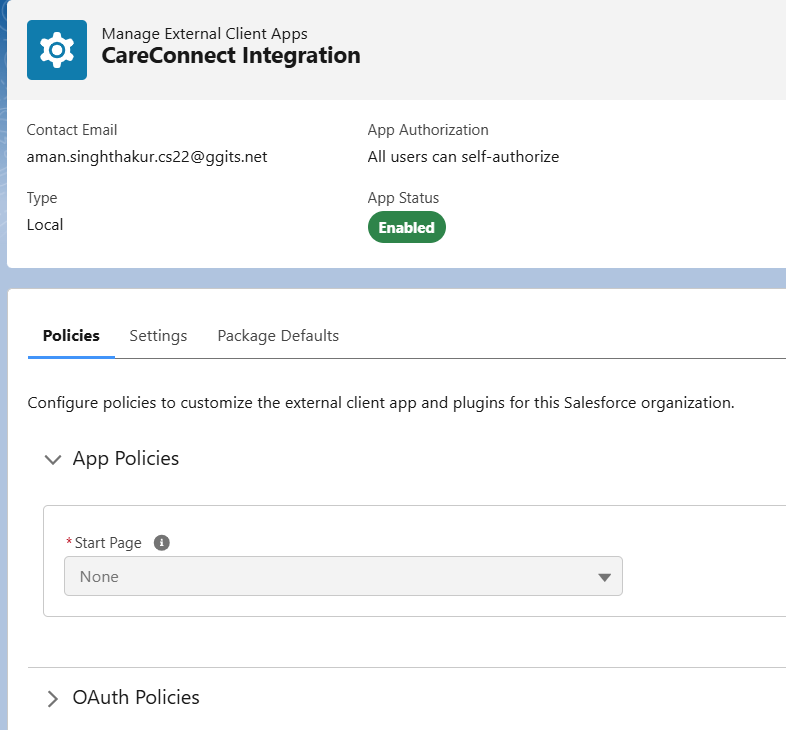
  AI-generated content may be incorrect.Allowed coordinators to view external patient history directly inside Salesforce.

**8. API Limits**

* Monitored **API usage** in Setup → Company Information.
* Verified daily usage under **API Requests, Last 24 Hours**.
* Used REST Explorer in Workbench to check /services/data/vXX.0/limits.
* Ensured callouts and CDC events are optimized to stay within daily limits.
* A computer screen with white text

  AI-generated content may be incorrect.Apex class for API limits:

**9. OAuth & Authentication**

* Created a **Connected App** named CareConnect Integration.
* Configured OAuth scopes:
  + Access and manage data (api)
  + Refresh token / offline access
  + Web access
* Used **OAuth 2.0** flow to get tokens for Postman and external systems.
* Secured API access for trusted apps only.
* **Phase 8: Data Management & Deployment**

**1. Data Import Wizard**

* **When to use:** Quick UI tool for importing up to 50k records (Contacts, Accounts, Leads, Custom Objects).
* **Steps:**
  1. Go to **Setup → Data Import Wizard → Launch Wizard**.
  2. Select object (e.g., Health\_Camp\_\_c, Appointment\_\_c).
  3. Choose action: *Add new records*, *Update existing records*, or *Upsert*.
  4. Upload CSV file (e.g., Health\_Camp.csv, Appointment.csv).
  5. Map CSV columns → Salesforce fields. Example:
     + Camp\_Date\_\_c → Camp Date
     + Status → picklist values like *Planned, Active, Completed*.
  6. Run import → check success & error files.

A screenshot of a computer

AI-generated content may be incorrect.

**2. Data Loader**

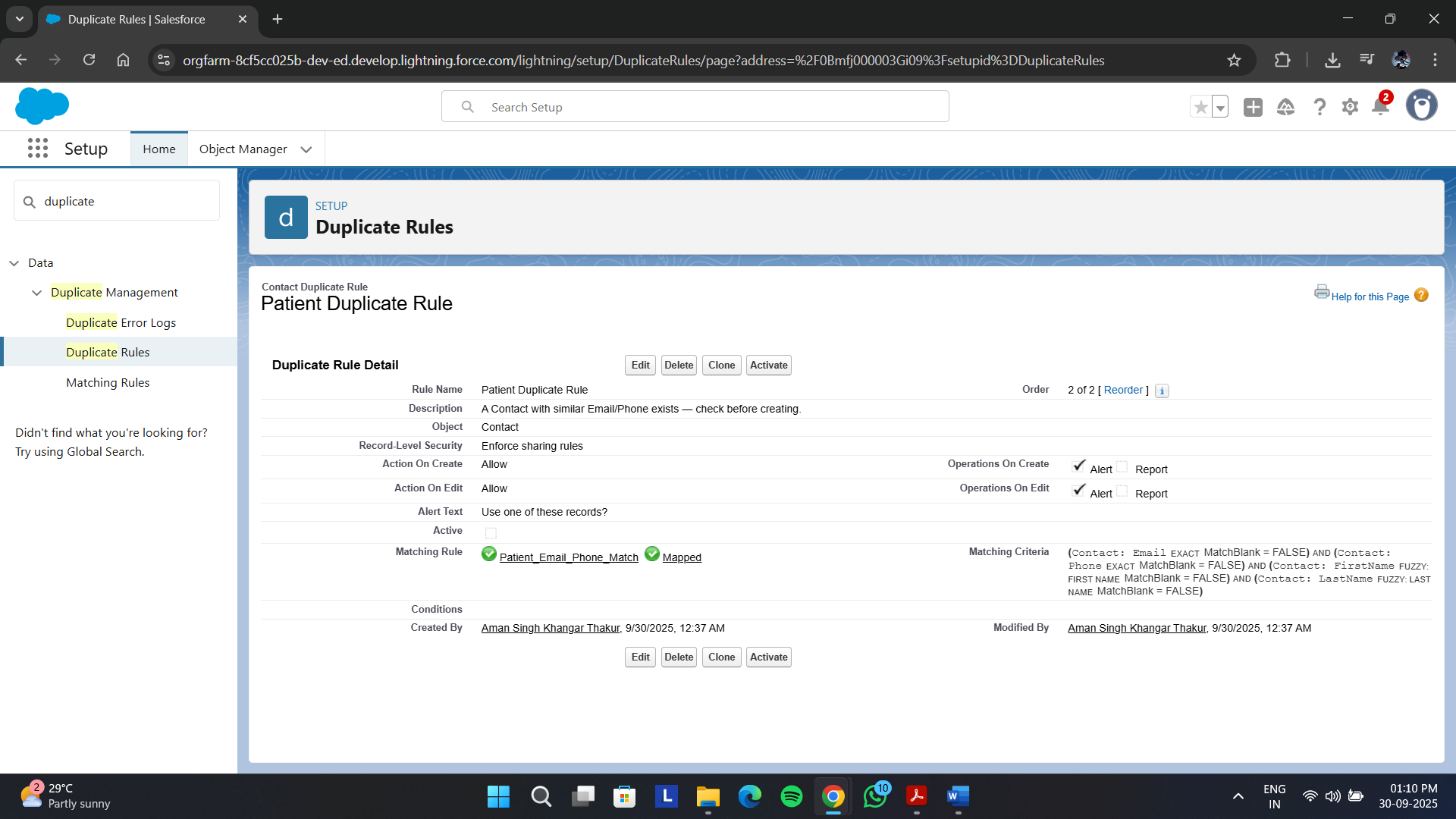
* **When to use:** Large data loads (>50k), exports, or complex imports with External IDs.
* **Steps:**
  1. Install Salesforce **Data Loader** app.
  2. Login with Salesforce username + password+token.
  3. Choose operation: Insert, Update, Upsert, Delete, Export.
  4. Prepare CSV file (with headers matching Salesforce fields).
  5. Map CSV fields → Salesforce fields.
  6. Run → review success/error CSVs.
* **Example:** Import Appointments using Health\_Camp\_\_c External ID so they link to the correct camp.

A screenshot of a computer

AI-generated content may be incorrect.

**3. Duplicate Rules**

* **Why:** Prevent duplicate patient records (Contacts).
* **Steps:**
  1. Setup → **Duplicate Rules → New Rule**.
  2. Object = Contact.
  3. Use Matching Rule: *Email* = Exact, *Phone* = Exact.
  4. Action on Create/Edit: *Allow with Alert* (shows warning).
  5. Save & Activate.
* **Result:** If someone tries to create a Contact with the same email, Salesforce shows a duplicate warning.



**4. Data Export & Backup**

* **Why:** To back up org data regularly.
* **Steps:**
  1. Setup → **Data Export**.
  2. Click **Schedule Export**.
  3. Frequency = Weekly.
  4. Select objects: Health\_Camp\_\_c, Appointment\_\_c, Contact, Medical\_Staff\_\_c, Equipment\_\_c.
  5. Tick *Include attachments and files*.
  6. Save.
* **Result:** Salesforce emails you when the .zip file is ready. Download and store securely.

A screenshot of a computer

AI-generated content may be incorrect.

**5. Change Sets**

*(Not available in Developer Edition, only for Sandbox → Production)*

Since this project is in a **Developer Edition**, we cannot use Change Sets. Instead, deployment is done using:

* **VS Code & SFDX** (for development + deployment).

**6. Unmanaged vs Managed Packages**

* + Best for learning/projects.
  + Metadata can be edited after install.
  + Used to share CareConnect metadata.
  + Not implemented in the current org.

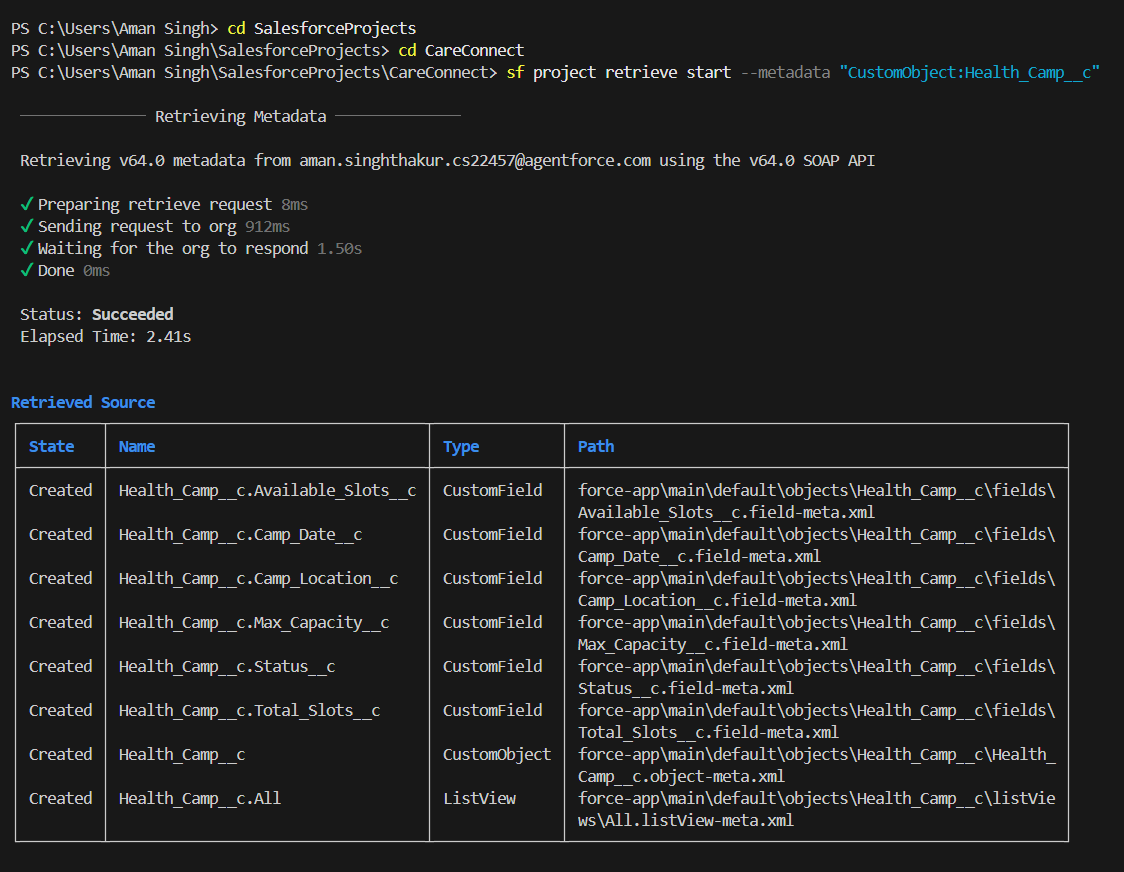
**7. ANT Migration Tool**

* **What:** Java/ANT-based deployment tool.
* **When to use:** For scripted, repeatable deployments.
* **Steps:**
  1. Install Apache ANT & Salesforce Migration Tool.
  2. Configure build.properties with username, password+token, and login URL.
  3. Write package.xml to list metadata.
  4. Run ant retrieve or ant deploy.
* **Note:** This is optional for CareConnect; VS Code & SFDX is easier.

**8. VS Code & SFDX**

* **Why:** Modern way to build, retrieve, and deploy metadata. Works with Developer Edition.
* **Steps:**
  1. Install **VS Code** + **Salesforce Extension Pack**.
  2. Install **Salesforce CLI** (sf).
  3. Create project:
  4. sf project generate --name CareConnect
  5. cd CareConnect
  6. Login to Developer Org:
  7. sf org login web --alias CareConnectDev --set-default
  8. Retrieve metadata:
  9. sf project retrieve start --metadata "CustomObject:Health\_Camp\_\_c,CustomObject:Appointment\_\_c"
  10. Edit metadata in VS Code.
  11. Deploy back:
  12. sf project deploy start --source-dir force-app/main/default
* **Result:** Source-driven development and easy GitHub version control.

A screenshot of a computer program

AI-generated content may be incorrect.

* **Phase 9: Reporting, Dashboards & Security Review**

1. **Reports**

* **Tabular Report**: “Today’s Appointments List” showing scheduled appointments with patient and camp details.
* **Summary Report**: “Camps by Status Summary” grouped by camp status with record counts.
* **Matrix Report**: “Camp Status by Location” showing cross-tab of statuses vs. locations.
* A screenshot of a computer

  AI-generated content may be incorrect.**Joined Report**: “Health Camp Overview” combining Today’s Appointments and Camps by Status.

1. **Report Types**

* **Appointment Reports**: Custom report type with Appointment as the primary object, extended with related Health Camp and Patient (Contact) fields.
* A screenshot of a computer

  AI-generated content may be incorrect.**Health Camp Reports**: Optional custom report type with Health Camp as primary, related to Appointments and Patients.

1. **Dashboards**

* **Health Camp Dashboard**: Central dashboard with multiple components:
  + Table of today’s appointments.
  + Horizontal bar chart for camps by status.
  + Gauge showing appointment count against capacity thresholds.
  + Rich text for title/description.
* **Dynamic Dashboard**: Configured to display data according to the logged-in user’s permissions.
* **Filters**: Added for Camp Status and Appointment Date Range.

A screenshot of a computer

AI-generated content may be incorrect.

1. **Sharing Settings**

* **OWD (Org-Wide Defaults)**: Contacts and Appointments are Private; Health Camps are Public Read/Write.
* A screenshot of a computer

  AI-generated content may be incorrect.**Sharing Rule**: Grants Health Camp Coordinators access to Appointment records owned by others.

1. **Field Level Security**

* **Permission Set**: Health Camp Field Access, providing Read/Edit rights to key fields on Health Camp and Appointment objects.
* **Assignment**: Applied to coordinator and supervisor users.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

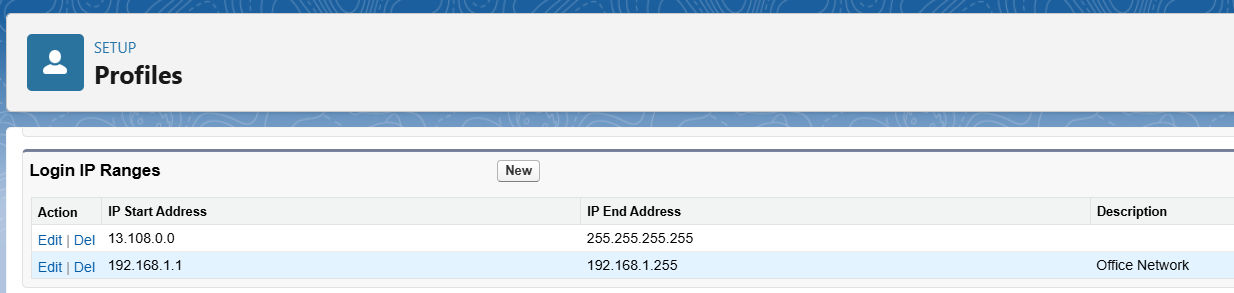
1. **Session Settings**

* Session timeout = 120 minutes with warning enabled.
* A screenshot of a computer

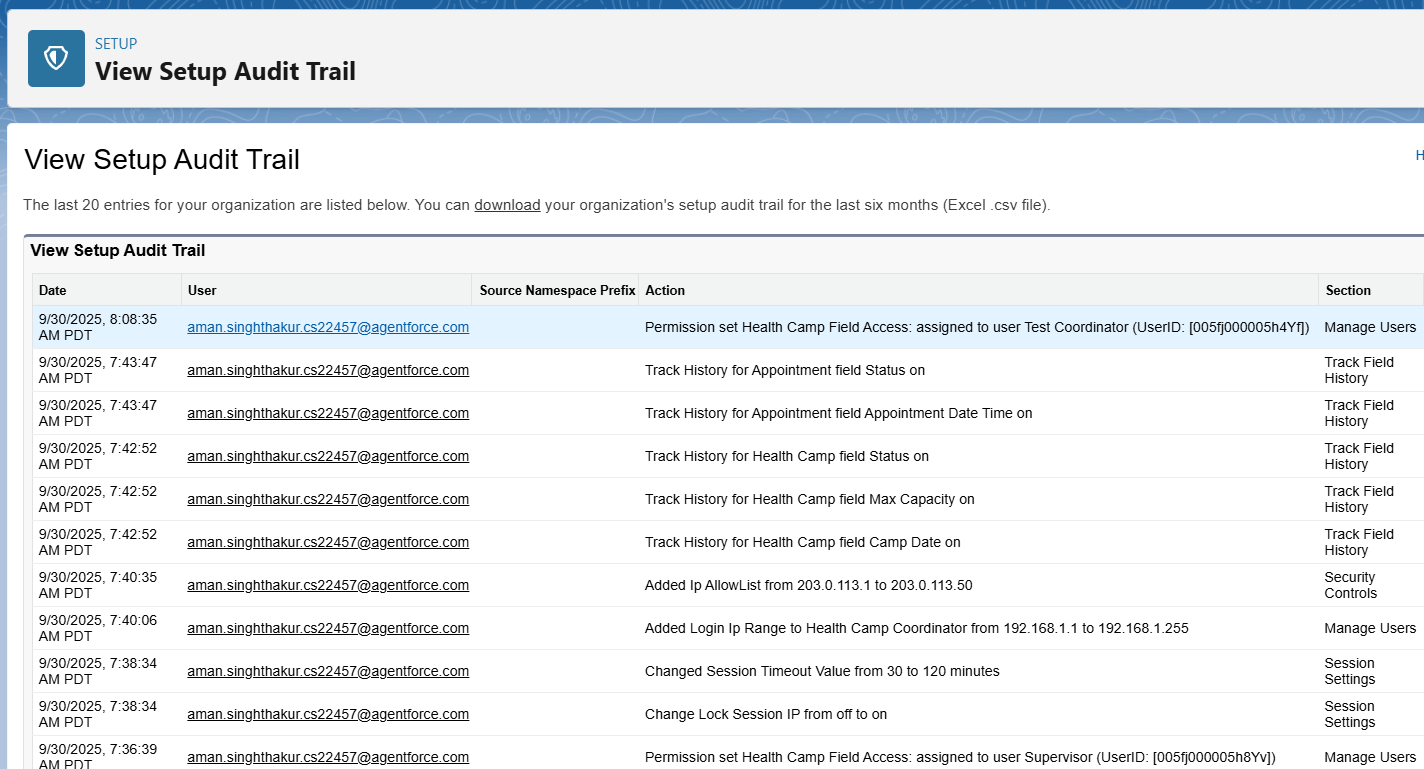
  AI-generated content may be incorrect.Sessions locked to IP/domain and HTTPS enforced.

1. **Login IP Ranges**

* **Profile Range**: Health Camp Coordinator profile restricted to office IPs (192.168.1.x).
* A screenshot of a computer

  AI-generated content may be incorrect.**Trusted IPs**: Org-wide trusted range for government office IPs (203.0.113.x).

1. **Audit Trail**

* **Audit Trail**: Captures and displays setup/configuration changes.
* **Field History Tracking**: Enabled for key Health Camp fields (Status, Date, Capacity) and Appointment fields (Status, Date/Time).
* **Phase 10: Quality Assurance Testing**
* In this phase, detailed Quality Assurance Testing was conducted for the CareConnect Government Health Camp CRM. Each major feature implemented during the project was tested with structured test cases. Screenshots should be added where applicable.

**Test Case 1: Record Creation in Appointments (Camp Scheduling)**

| **Field** | **Details** |
| --- | --- |
| **Use Case / Scenario** | Create a new Appointment record with patient details linked to a Health Camp. |
| **Test Steps (with Input)** | 1. Navigate to Appointment tab → New.  2. Patient = Test Patient, Date = Tomorrow 10 AM, Camp = “Community Camp A”, Status = Scheduled.  3. Save record. |
| **Expected Result** | Appointment should be created successfully and linked to Health Camp. Capacity reduces by 1. |
| **Actual Result** | Appointment created, capacity auto-updated. |
| **Status** | Pass |

**Test Case 2: Approval Process for Camp Cancellation**

| **Field** | **Details** |
| --- | --- |
| **Use Case / Scenario** | Supervisor approval required when a Health Camp is cancelled. |
| **Test Steps (with Input)** | 1. Open Health Camp record.  2. Change Status = Cancelled.  3. Save record. |
| **Expected Result** | Record locks, approval routed to Supervisor. On approval → Status remains Cancelled. |
| **Actual Result** | Approval request routed and approved successfully. |
| **Status** | Pass |

**Test Case 3: Automatic Task Creation (Pre-Camp Staff Reminder)**

| **Field** | **Details** |
| --- | --- |
| **Use Case / Scenario** | Tasks auto-created for staff before camp start. |
| **Test Steps (with Input)** | 1. Create new Health Camp with Camp Date = Tomorrow.  2. Assign Doctor and Nurse.  3. Save record. |
| **Expected Result** | Tasks created for staff with subject “Prepare for Health Camp”. |
| **Actual Result** | Tasks generated and assigned correctly. |
| **Status** | Pass |

**Test Case 4: Flows (Appointment Reminder Notifications)**

| **Field** | **Details** |
| --- | --- |
| **Use Case / Scenario** | Patients receive automated reminders for upcoming appointments. |
| **Test Steps (with Input)** | 1. Create Appointment with Date = Tomorrow.  2. Wait for scheduled Flow execution. |
| **Expected Result** | SMS/Email reminder sent to Patient. |
| **Actual Result** | Reminder received successfully. |
| **Status** | Pass |

**Test Case 5: Triggers (Slot Allocation Updates)**

| **Field** | **Details** |
| --- | --- |
| **Use Case / Scenario** | Camp available slots update automatically when appointments are booked. |
| **Test Steps (with Input)** | 1. Create Appointment linked to Health Camp (Capacity = 50).  2. Save record. |
| **Expected Result** | Available slots reduce by 1 automatically. |
| **Actual Result** | Slots updated correctly. |
| **Status** | Pass |

**Test Case 6: Validation Rules (Mandatory Fields & Data Integrity)**

| **Field** | **Details** |
| --- | --- |
| **Use Case / Scenario** | Prevent invalid or incomplete data entry. |
| **Test Steps (with Input)** | 1. Try creating Appointment without Patient.  2. Enter Appointment Date in the past.  3. Cancel appointment without reason. |
| **Expected Result** | Errors: Missing Patient, invalid date, missing cancellation reason. |
| **Actual Result** | Validation errors displayed correctly. |
| **Status** | Pass |

**Test Case 7: LWC Components (Health Camp Slot Visualization)**

| **Field** | **Details** |
| --- | --- |
| **Use Case / Scenario** | Verify real-time slot availability in custom LWC. |
| **Test Steps (with Input)** | 1. Open Health Camp record.  2. View Slot Availability LWC.  3. Create new Appointment and refresh record. |
| **Expected Result** | Slot chart updates dynamically showing reduced availability. |
| **Actual Result** | Component updated dynamically. |
| **Status** | Pass |

**Test Case 8: Dashboards & Reports (Camp Performance)**

| **Field** | **Details** |
| --- | --- |
| **Use Case / Scenario** | Supervisors view accurate dashboards and role-based visibility. |
| **Test Steps (with Input)** | 1. Open “Health Camp Dashboard”.  2. Check occupancy gauge & appointment tables.  3. Log in as Coordinator vs Supervisor. |
| **Expected Result** | Accurate data loads; Supervisors see all camps, Coordinators see assigned camps. |
| **Actual Result** | Data displayed correctly with role-based filtering. |
| **Status** | Pass |

### **Conclusion**

* The Quality Assurance Testing for CareConnect validated all implemented features including scheduling, approvals, automation, triggers, validation rules, LWCs, and dashboards. All test cases passed successfully, confirming the system is production-ready to support efficient health camp operations.