

Project Title: Smart Healthcare CRM – Patient Appointment & Care Management System

Industry: Healthcare

Project Type: B2C Salesforce CRM implementation

Target Users: Clinic Administrators, Doctors, and Patients

Problem Statement:

A mid-sized healthcare clinic faces challenges in managing patient appointments, follow-ups, and doctor availability. Manual scheduling leads to double bookings, missed follow-ups, and poor communication with patients. Doctors do not have centralized access to patient records, and the management lacks visibility into clinic performance.

To overcome these issues, the clinic wants to implement a

Salesforce-based Healthcare CRM to:

- Automate patient appointment booking and reminders
- Maintain centralized patient health records
- Improve doctor–patient communication with alerts and notifications
- Provide dashboards for clinic performance monitoring

Use Cases:

1. Appointment Management

- Patients book appointments online through a portal (LWC).
- Automatic **email alerts** confirm appointments.
- Doctors receive **custom notifications** inside Salesforce when new appointments are scheduled or cancelled.
- Tasks are created for clinic staff (e.g., “Prepare Room for Consultation”).

2. Patient Records

- Centralized patient profiles store medical history, prescriptions, and lab reports.
- Secure access based on role (Doctor, Admin, Patient).
- Doctors can view patient history in real-time before appointments.

3. Follow-up Automation

- The system automatically creates follow-up appointments post-consultation.
- **Email alerts** remind patients of upcoming follow-ups.
- **Tasks** are generated for staff to call patients for lab results or medication updates.

4. Notifications & Communication

- **Custom Notifications** notify doctors instantly about rescheduled/cancelled appointments.
- Patients receive appointment updates via SMS/Email.

5. Apex Development & Async Processing

- Modular Apex classes, including a trigger handler to keep logic clean and a Queueable class for async SMS/email reminders with callouts.
- Error logging via a custom object and develop unit tests with HTTP callout mocks to ensure code quality and coverage.
- Deployment will be done using SFDX to the Developer Org, followed by end-to-end testing to verify notifications, task creation, and error handling.

6. Asynchronous Processing

- Bulk SMS/WhatsApp reminders sent using **Future Methods / Queueable Apex** (to avoid governor limits).

- Asynchronous jobs update patient records when integrating with external EHR systems.
- Large-scale data (e.g., 1,000+ patients follow-up schedule) processed in the background.

7. Integration & External Access

- Configured Named Credentials to securely authenticate with external healthcare APIs using bearer tokens.
- Set up the API endpoint within Named Credential to avoid hardcoding URLs in Apex callouts.
- Developed a Queueable Apex class to perform HTTP callouts for syncing appointment data externally.

8. Data Management & Deployment

- Imported Patient and Doctor data using Data Import Wizard for basic records setup.
- Used Data Loader to insert Appointment records with lookup fields.
- Deployed metadata using VS Code and SFDX to enable version control and team collaboration.

9. Reporting, Dashboards & Security Review

- Created various report types (Tabular, Summary, Matrix, Joined) and dashboards to visualize data.
- Configured profiles, roles, permission sets, and users for secure access control.

- Applied OWD, sharing rules, and session settings to manage data visibility and session security.

10. Quality Assurance Testing

- Designed test cases covering features like Flows, Triggers, Validation Rules, and Approval Processes.
- Captured inputs, expected results, and actual outputs with mandatory screenshots for each test.
- Summarized the testing approach, highlighted future enhancements, and concluded project outcomes.