TITLE: CycleCare-A period tracker

Phase 1: Problem Understanding & Industry Analysis

1. Requirement Gathering

Purpose: Collect detailed requirements from potential users.

Steps:

1. Identify target users:
    - Patients (tracking cycle & appointments)
    - Gynecologists/doctors (appointment management, patient history)
    - Admins (monitoring overall app usage)
2. Gather functional needs:
    - Track cycle dates (start, end, irregularities)
    - Predict ovulation & fertile windows
    - Book doctor appointments & send reminders
    - Locate gynecologists nearby (integration with maps/doctor APIs)
    - Notifications for irregular cycles → suggest medical visits
3. Gather non-functional needs:
    - Mobile-first design
    - Secure handling of sensitive medical data
    - Simple and intuitive UI
    - Scalable for large user base

2. Stakeholder Analysis

Purpose: Understand **who benefits from the app** and what they need.

Stakeholder	Needs	Pain Points
Patients	Cycle tracking, reminders, ovulation prediction	Forgetting cycles, lack of medical awareness
Doctors	Appointment scheduling, patient history access	Time management, lack of patient info
Admins	Manage app usage, reports, compliance	Ensuring data security & accurate insights

3. Business Process Mapping

Purpose: Visualize workflows inside the app.

Example Processes for Cycle Care:

- **Cycle Tracking Flow:**
  User logs cycle → System predicts ovulation → Sends reminders → Flags irregular cycles
- **Appointment Booking Flow:**
  User searches doctor → Finds nearby gynecologist → Books slot → Doctor confirms → Reminder sent
- **Doctor Interaction Flow:**
  Doctor logs in → Views appointments → Accesses patient cycle data → Suggests treatment

(Best represented in a flowchart or BPMN diagram – I can create one if you'd like 👀).

4. Industry -Specific Use Case Analysis

Purpose: See what's already in the **FemTech (female health tech)** industry and tailor accordingly.

Common Features in Similar Apps:

- **Period Tracking Apps (Clue, Flo, Ovia):** AI-based predictions, symptom logging
- **Doctor Appointment Apps (Practo, Zocdoc):** Location-based search, teleconsultations
- **Wellness Apps (Fitbit, Apple Health):** Integration with wearables, lifestyle tracking

Insights for Cycle Care:

- Start simple (cycle tracking + doctor appointments).
- Later enhancements → wearable integration, AI predictions, telemedicine.

5. AppExchange Exploration

Purpose: Research what Salesforce already offers.

Steps:

1. Go to **AppExchange** → Search terms: *"Healthcare", "Patient Management", "Appointments", "FemTech"*
2. Analyze existing apps:
   - Health Cloud (by Salesforce) → for patient records & care plans
   - Doctor Appointment Scheduler apps
   - Maps & Location services
3. Learn from them:
   - What features do they already provide?
   - What gaps can Cycle Care fill? (e.g., personalized female health focus)

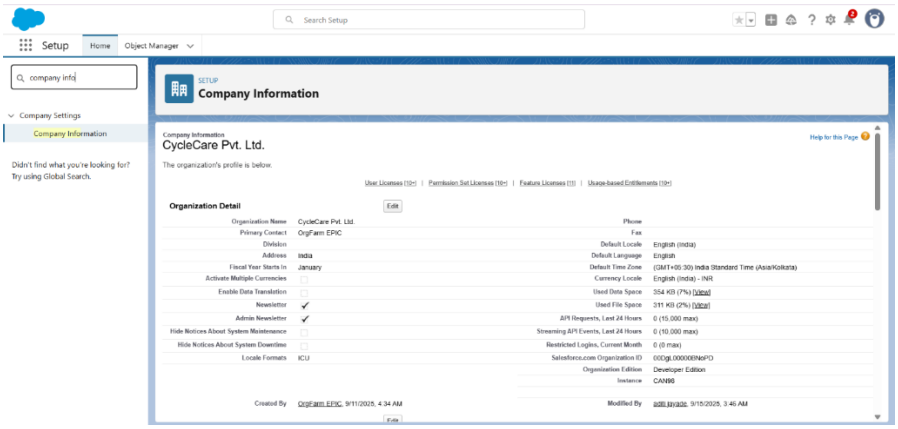
Phase 2: Org Setup & Configuration

1. Salesforce Edition

- Used **Developer Edition**: free, full features, can test apps before bigger editions.

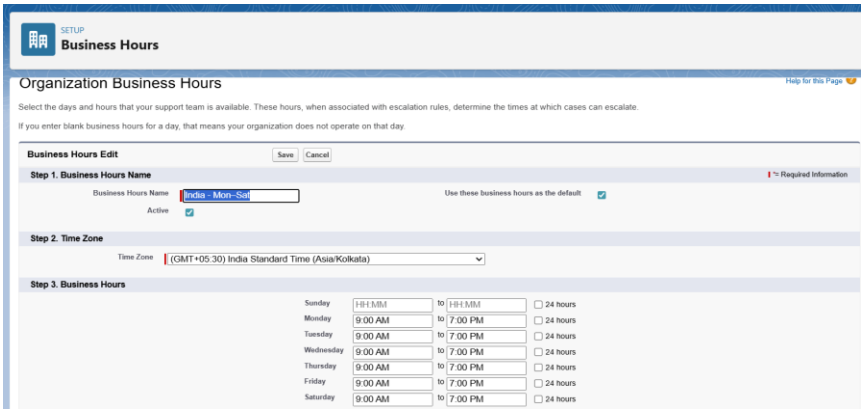
2. Company Profile

- **Settings:** India, currency INR, English.
- **Why:** Correct money calculations, accurate dates, compliance.

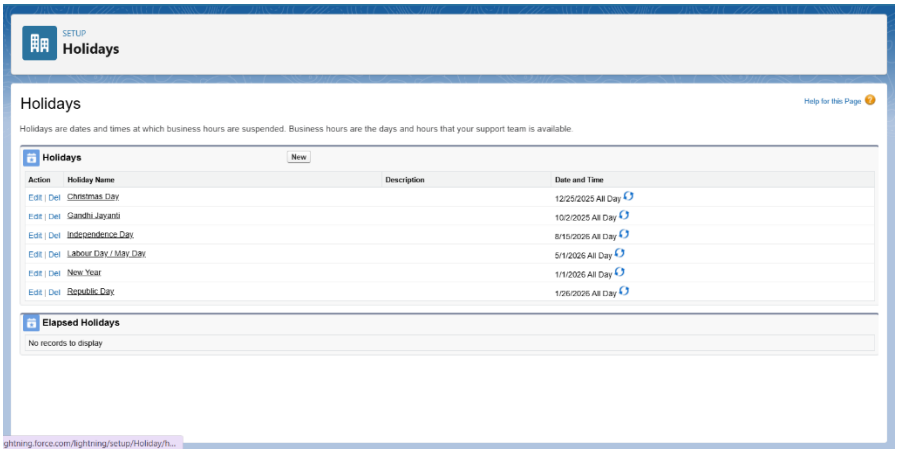


3. Business Hours & Holidays

- **Work hours:** Mon–Sat, 9 AM – 7 PM



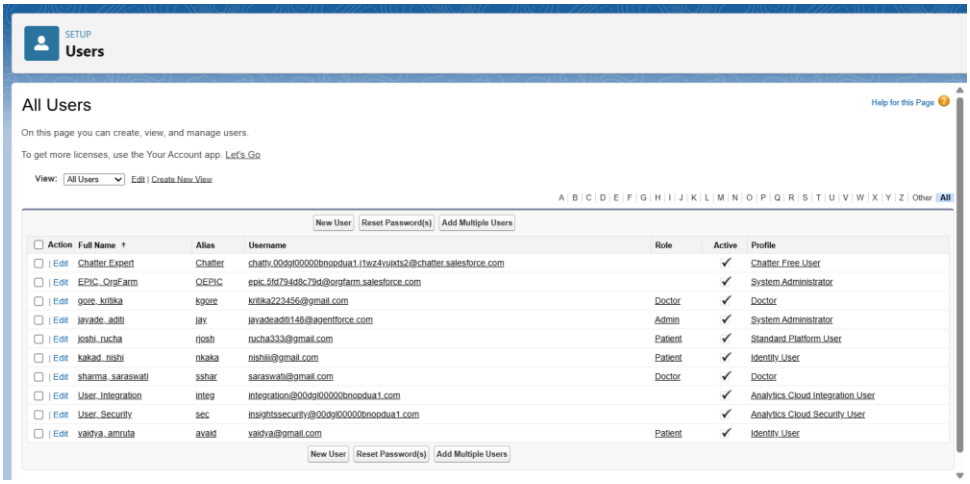
- **Holidays:** National holidays.



- **Why:** Service booking, case escalation, and SLAs depend on working hours.

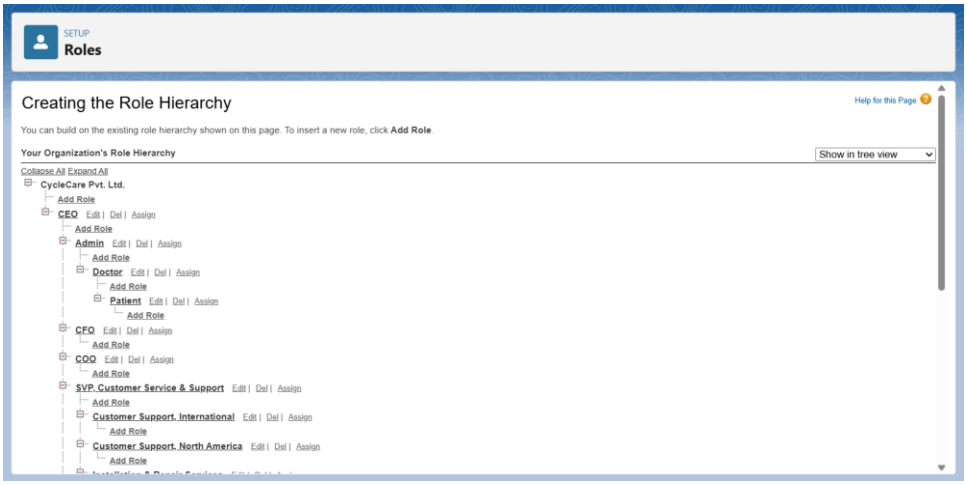
4. User Setup

- Users: Admin,Doctor,Patient.
- Assign proper licenses for access.



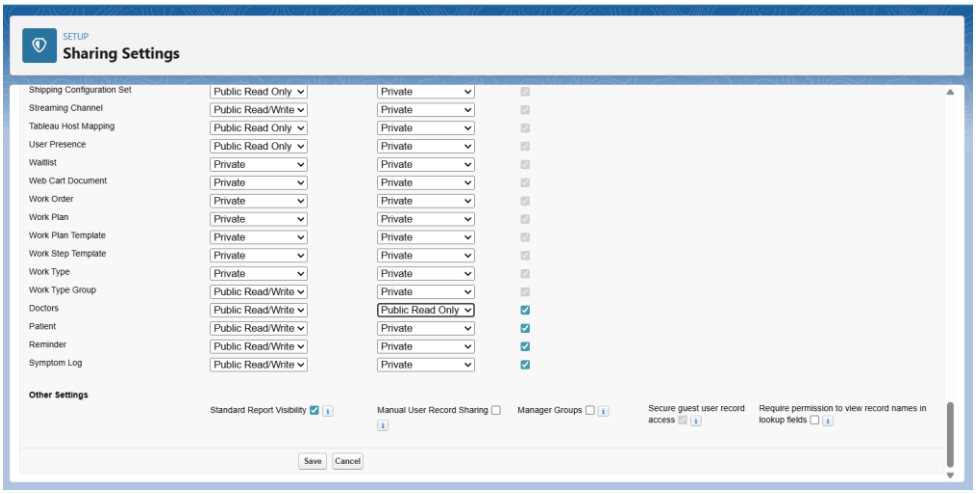
5. Profiles & Roles

- **Profiles:** Control what each user can do.
  - I have basically made three profiles and they are: Admin ,Doctor and Patient
- **Roles:** Show hierarchy (Admin → Doctor →Patient)



6. OWD & Sharing Rules

- **OWD:** Private for service records.
- **Sharing:** Doctor sees all, customer sees own records.

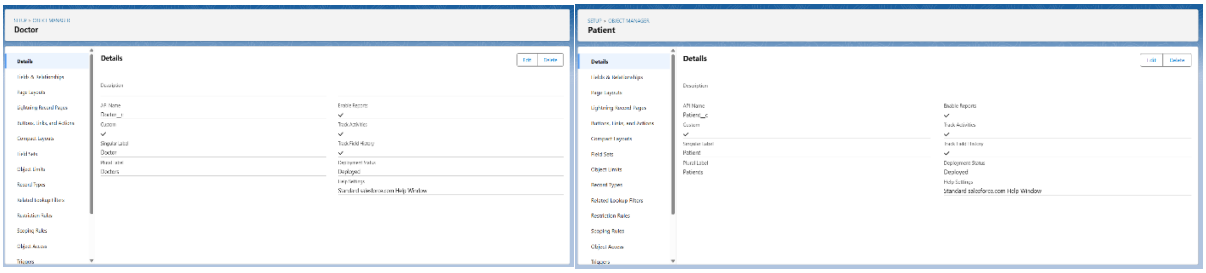


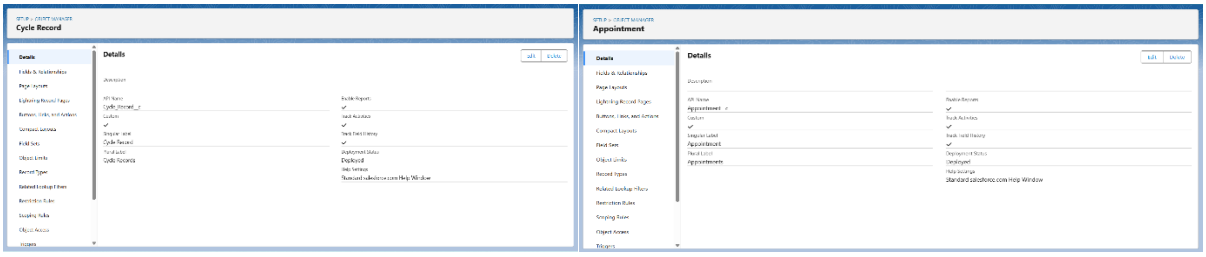
Screenshots: Company Info, User page, Profiles, Roles, OWD settings.

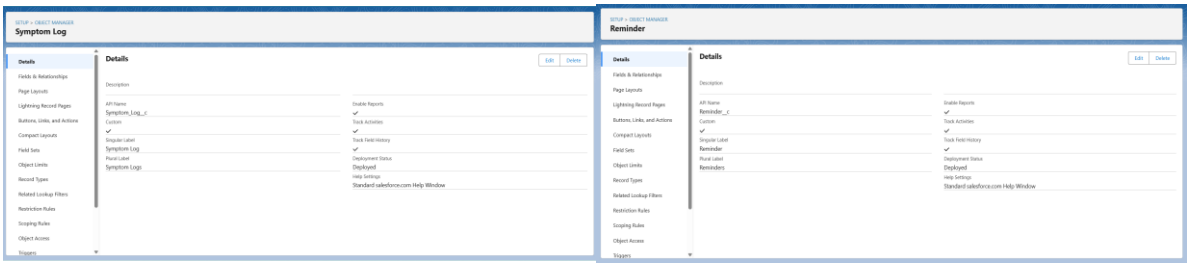
Phase 3: Data Modeling & Relationships

1. Custom Objects

- Doctor, Patient, Appointment, Cycle Record, Symptom logs, Reminders

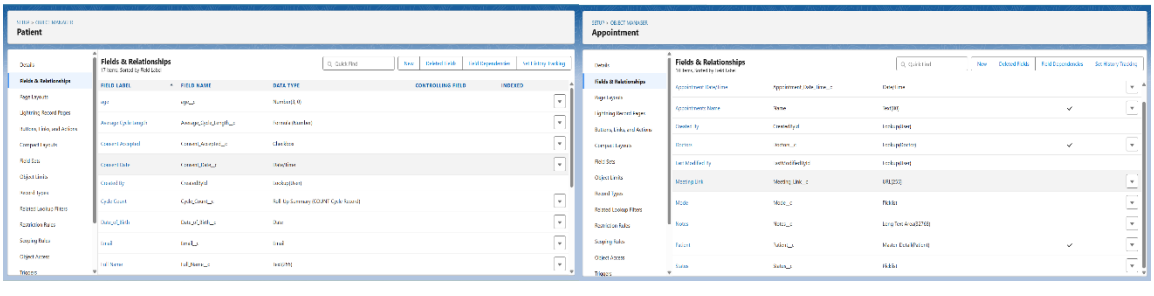


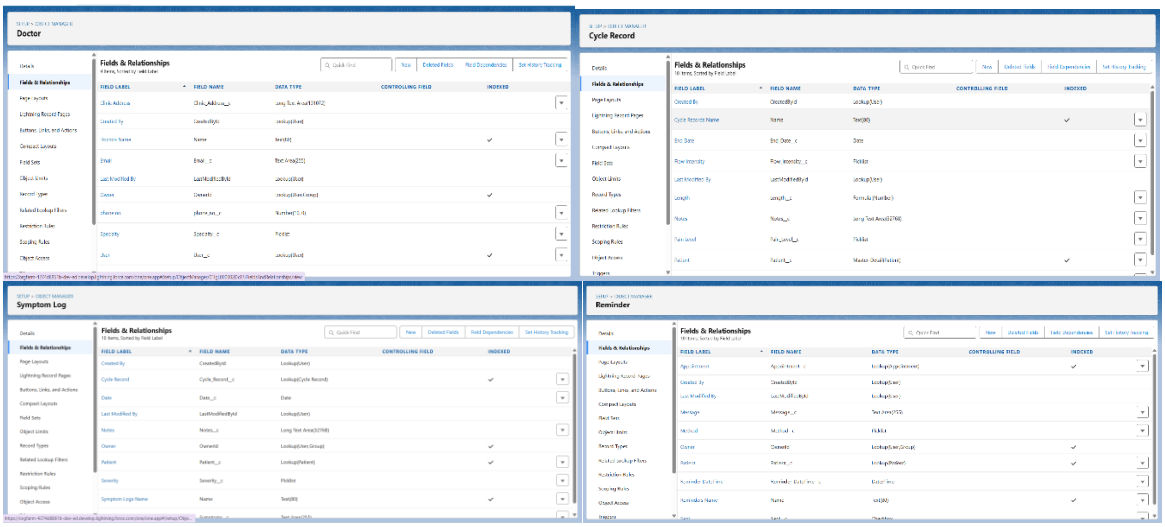




2. Fields

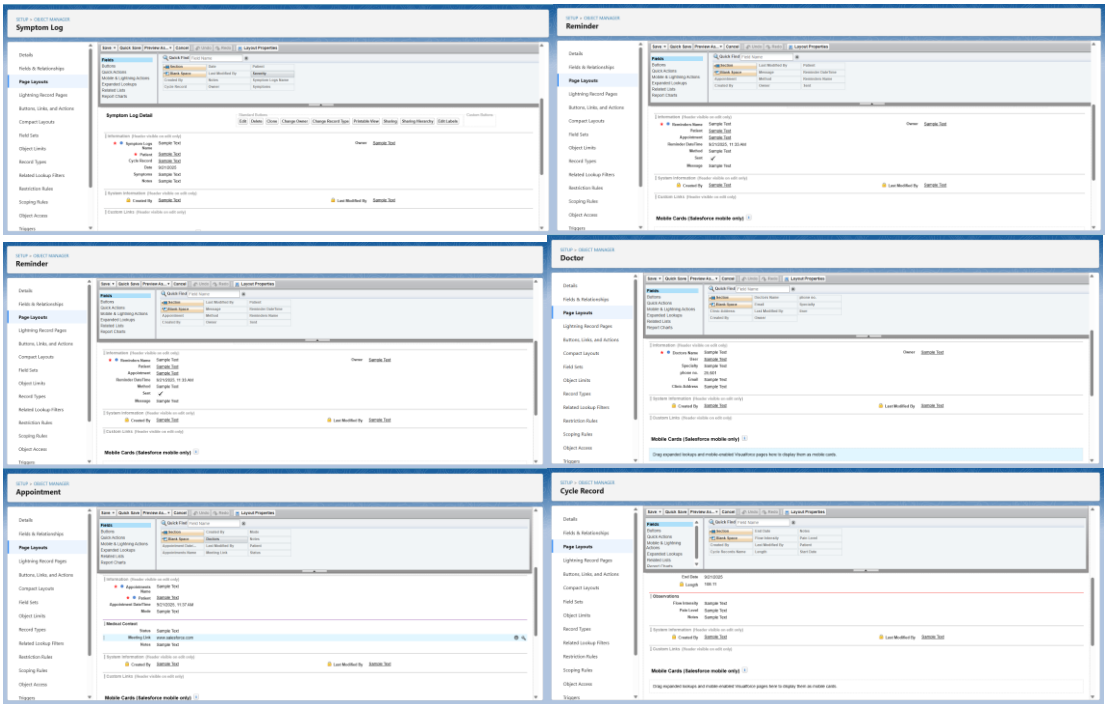
- **Doctor:** Name, Phone, Email, Address
- **Patient:** Model, Frame No., Purchase Date, Warranty
- **Appointment:** Date, Status, Mechanic assigned
- **Cycle_record:** Amount, Mode, Status, linked booking
- **Symptom logs:** Amount, Mode, Status, linked booking
- **Reminders**: Amount, Mode, Status, linked booking





3. Page Layouts & Compact Layouts

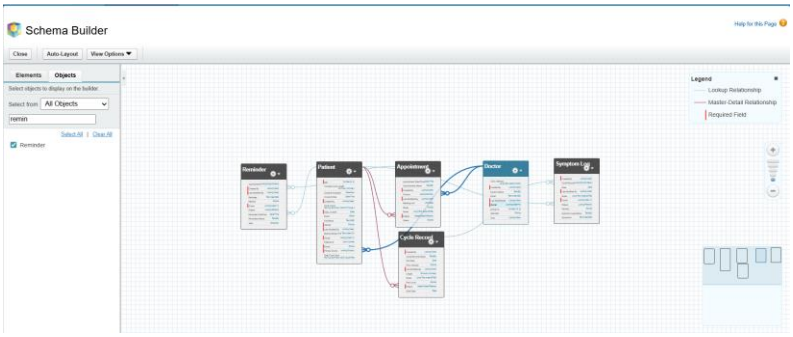
- Arrangement of fields, sections, related lists.



4. Schema Builder (ERD)

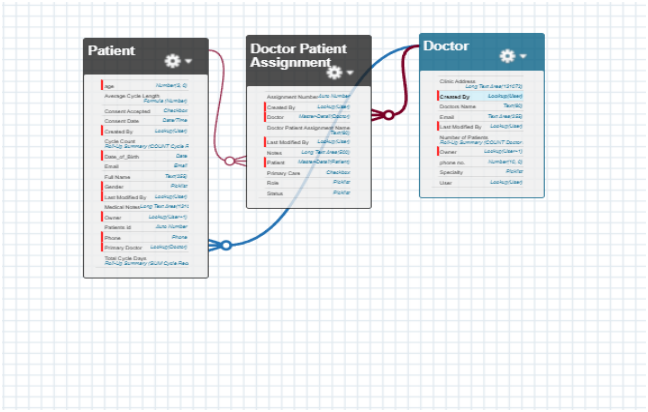
- Visual representation of objects and their relationships.
  - Appointment → Lookup → Doctor
  - Appointment → MDR→ Patient
  - Patient → Lookup → Doctor
  - Reminder→ Lookup → Patient

    - Reminder→ Lookup → Appointment
    - Cycle_Record → MDR→ Patient
    - Symptom Log→ Lookup → Patient
    - Symptom Log→ Lookup → Cycle_Record
- **Why:** Proper relationships help reporting and automation.



5.Junction Objects

- Handle **many-to-many relationships**.



Screenshots: Object Manager, Page Layouts, Schema Builder diagram.

Phase 4: Process Automation (Admin)

1. Validation Rules

- Ensure correct data entry (e.g., cycle length must be >10 days).

2. Workflow Rules

- Automate simple tasks like sending reminders.

3. Process Builder / Flow

- Handles conditional automation like irregular cycle detection and Provides powerful automation with Screen Flows, Record-Triggered Flows, and Scheduled Flows (e.g., monthly health reminders).

4. Approval Process

- Used if appointment requests need doctor approval.

5. Email Alerts & Notifications

- Deliver timely alerts for users about cycles or appointments.

{tasks,custom notification}

Screenshots: Validation rule editor, workflow setup, flow canvas, approval process page, email templates.

Phase 5: Apex Programming (Developer)

1. Trigger

- Automate actions before/after records are inserted or updated.

2. SOQL & SOSL Queries

- Query Salesforce data to fetch cycle records or doctor details.

3. Batch / Queueable / Scheduled Apex

- Handle large data jobs like analyzing cycles for all users.

4. Exception Handling & Test Classes

- Ensure reliability and quality of code.

5. Collections (List, Set, Map)

- Manage multiple records efficiently.

6. Future Methods

- Make asynchronous external API calls.

7. Asynchronous Processing

- Improves performance when handling bulk data or external calls.

Screenshots: VS Code/Developer Console, trigger code, execution logs, test results.

Phase 6: User Interface Development

1. Lightning App Builder

- Assembles app pages with standard and custom components.

2. Record Pages , Tabs, Home Page, Utility Bar

- Provide structured navigation.

3. Apex with LWC

- Fetch calculated results such as predicted ovulation date.

4. Events in LWC

- Enable communication between UI components.

5. Wire Adapters & Imperative Apex Calls

- Fetch Salesforce data seamlessly.

6. Lightning Web Components (LWC)

- Create custom interactive features like an ovulation calendar.

7. Navigation Service

- Provides smooth movement between records and pages.

Screenshots: App Builder UI, record page preview, tabs, LWC component preview.

Phase 7: Integration & External Access

1. Named Credentials & Remote Site Settings

- Simplify secure API access.

2. Web Services (REST/SOAP)

- Salesforce can expose or consume APIs for doctor listings or telemedicine services.

3. Platform Events & Change Data Capture (CDC)

- Enable real-time communication with external systems.

4. External Services

- Allow integration through OpenAPI definitions.

5. Callouts

- Apex makes outbound API requests.

6. Salesforce Connect

- Access external databases without storing data.

7. API Limits

- Define boundaries for integrations.

8. OAuth & Authentication

- Ensure secure login and API transactions.{remote site settings}

Screenshots: Named Credential, Remote Site Settings, Postman API request/response, Event Monitoring.

Phase 8: Data Management & Deployment

1. Data Import Wizard

- Import/export user, cycle, and doctor data.

2. Change Sets

- Migrate metadata between Sandbox and Production.

3. Duplicate Rules

- Prevents duplicate patient or doctor records.

4. Data Export & Backup

- Safeguards against data loss.

5. Packages

- **Unmanaged** for prototypes.
- **Managed** for commercial release.

6. ANT Migration Tool & SFDX (VS Code)

- Provide developer-driven deployment and version control.

Screenshots: Import Wizard, Data Loader UI, duplicate rule setup, export log, deployment page.

Phase 9: Reporting & Dashboards

1. Reports & Report Types

- Generate tabular, summary, matrix, and joined reports for tracking cycles, appointments, and irregularities.
- Define custom reporting across objects.

2. Dashboards & Dynamic Dashboards

- Visualize key health metrics personalized for users and admins.

3. Sharing Settings

- . Define record-level access.

4. Field Level Security

- Protect sensitive medical details.

5. Session Settings & Login IP Ranges

- Strengthen login security.

6. Audit Trail

- Track changes in configuration for compliance and monitoring.

Screenshots: Report charts, dashboard snapshot, dynamic dashboard example, security settings.
