ROADMAP OF BOOK AND MANAGE PATIENT MANAGEMENT SYSTEM PROJECT

Phase 1: Problem Understanding & Industry Analysis

Goal: Define what and why you're building.

Steps

1. Document clinic workflow (appointments, reminders, cancellations).
2. List all users: Admin, Receptionist, Doctors, Patients.
3. Create a business process diagram showing:
    - Patient registers → Receptionist books → Doctor attends → Appointment completed.
4. Create a feature list:
    - Appointment booking
    - No overlapping
    - Confirmation + reminder
    - Reports & dashboards
    - Automatic status updates
5. Collect 5–10 real sample records (doctors/patients) to use for testing later.

Phase 2: Org Setup & Configuration

Goal: Prepare Salesforce environment.

Steps

1. Create a **Developer Edition** Salesforce org.
2. In **Company Settings** → set Company Info (Clinic), Fiscal Year, Locale.
3. **Business Hours**: Setup → Company Settings → Business Hours → "ClinicHours" (Mon–Sat 9–6).
4. **Holidays**: Setup → Company Settings → Holidays → add few sample holidays.

5. **Users**:
    - Admin (you)
    - Receptionist
    - 2–3 Doctor users
6. **Profiles**:
    - Receptionist: CRED on Patient/Appointment
    - Doctor: Read-only on others, full on their own appointments
    - Admin: Full
7. **Roles**:
    - Admin → Receptionist → Doctors
8. **Permission Sets**:
    - "Flow Access" (Flow + View All Data) → assign to Receptionist
9. **OWD (Sharing Settings)**:
    - Appointment: Private
    - Doctor/Patient: Public Read Only
10. **Sharing Rules**:

- Appointment → grant read to Receptionist group.

11. **Login Hours/IPs** (optional security):

- Restrict Receptionist login to working hours.

Phase 3: Data Modeling & Relationships

Goal: Create data structure.

Steps

1. Create Custom Objects:
    - Doctor__c
    - Patient__c
    - Appointment__c

2. Add Fields:
    - **Doctor**: Specialization (Text), Phone, Email
    - **Patient**: Age (Number), Gender (Picklist), Phone, Email
    - **Appointment**: Date/Time (DateTime), Status (Picklist), Notes (Long Text), Doctor (Lookup), Patient (Lookup)
3. Set Relationship:
    - Appointment → Doctor (Lookup)
    - Appointment → Patient (Lookup)
4. Record Types:
    - Appointment: Regular / Follow-up
5. Page Layouts:
    - Show related appointments on Doctor and Patient layout
6. Compact Layouts:
    - Appointment: Date, Doctor, Patient, Status
7. Use **Schema Builder** to verify your ER diagram visually.

Phase 4: Process Automation (Admin)

Goal: Automate business logic.

Steps

1. **Validation Rules:**
    - Appointment Date ≥ TODAY()
2. **Before-Save Flow (Record-Triggered):**
    - Object: Appointment
    - Check if same Doctor has appointment at same date/time → if found → throw error
3. **After-Save Flow:**
    - Send **Email Alert** to Patient using their Email field
4. **Scheduled Path (Flow):**

- When Appointment is created → schedule 1 day before Date → send Reminder Email

5. **Field Update:**
    - When current date > appointment date → set Status = "Completed" automatically using scheduled Flow.
6. **Custom Notification:**
    - Send to Receptionist on new appointment (Optional but include now)
7. **Create Task:**
    - Flow: Create a task for Doctor “Appointment with [Patient] on [Date]”.

Phase 5: Apex Programming (Developer)

Goal: Add advanced logic.

Steps

1. Create **Apex Trigger** on Appointment:
    - If appointment Date < Today and Status != Completed → auto set Status = “Missed”
2. Create **Apex Class** for bulk rescheduling:
    - Method to reschedule all appointments of a doctor to next day
3. Create **Batch Apex**:
    - Delete all appointments older than 1 month
4. Create **Queueable Apex**:
    - To send background email reminders (if not using Flow)
5. **Scheduled Apex**:
    - Run daily at 8AM → email Admin count of appointments
6. **Future Methods**:
    - Call external SMS API on appointment creation
7. **Test Classes**:

- Write test coverage (75%+) for all classes/triggers

8. Use **Exception Handling** to handle overlap/conflicts.

Phase 6: User Interface Development

Goal: Create user-friendly UI.

Steps

1. Lightning App Builder → New App → "Clinic Management"
2. Add Tabs: Doctors, Patients, Appointments
3. Create Custom Record Pages for Appointment and Doctor
4. Add Related Lists and Quick Actions: "Book Appointment" on Doctor page
5. Utility Bar → Add Quick New Appointment
6. Create **Home Page** → Add Dashboard components
7. Build **LWC Component**:
    - Calendar view of appointments (Doctor-wise filter)
8. Build **LWC Form** to book appointment (calls Apex)
9. Use **Navigation Service** to redirect after booking
10. Use **Wire/Imperative Apex** to fetch and display appointments dynamically.

Phase 7: Integration & External Access

Goal: Connect external systems.

Steps

1. Create **Experience Cloud site**:
    - Patients can book appointments online
2. Create **Public Web-to-Lead**-style form:
    - Map to Patient + Appointment object
3. Configure **Named Credentials** for SMS API
4. Create **Apex Callout Class** to send SMS

5. Create **REST Web Service** (Apex @RestResource):
    - To expose appointment booking endpoint
6. Enable **Remote Site Settings** for SMS API URL
7. Use **Platform Events** to notify reception when appointment is cancelled
8. Use **Change Data Capture** on Appointment for external sync
9. Apply **API Limits** monitoring in Setup
10. Configure **OAuth Connected App** (if you want secure patient login).

Phase 8: Data Management & Deployment

Goal: Manage data and metadata.

Steps

1. Use **Data Import Wizard** to insert Doctors and Patients.
2. Use **Data Loader** to bulk insert Appointments.
3. Create **Duplicate Rules** for Patient Email field.
4. Export data regularly for backup.
5. Use **Change Sets** to deploy metadata to sandbox/production.
6. Use **VS Code + SFDX CLI** for version control:
    - sf project generate
    - sf retrieve metadata
    - sf deploy metadata
7. Use **ANT Migration Tool** if you want CLI-based deploy.

Phase 9: Reporting, Dashboards & Security Review

Goal: Monitor and secure data.

Steps

1. Create Custom Report Type: Appointment with Doctor and Patient
2. Create Reports:

- Today's Appointments
- Appointments by Doctor
- Cancelled Appointments

3. Create Dashboards:
    - Appointments by Doctor (Pie)
    - Appointments per Day (Bar)
4. Enable **Dynamic Dashboards** (User-Specific)
5. Field Level Security:
    - Hide phone/email for doctors from non-admins
6. Login IP Ranges and Session Timeouts
7. Enable **Audit Trail** to track setup changes.

Phase 10: Final Presentation & Demo Day

Goal: Deliver and present the project.

Steps

1. Create PPT:
    - Problem → Features → Architecture → Demo → Benefits
2. Prepare Demo Script:
    - Create patient → book appointment → test overlap block → show emails/SMS → show dashboard
3. Add Documentation:
    - ERD, Field list, Flow screenshots, Trigger code
4. Post on LinkedIn/GitHub as a portfolio project.