TITLE: **CycleCare-**A period tracker

Phase 1: Problem Understanding & Industry Analysis

1. Problem Statement

Cycle Care faces several challenges because it doesn't use a digital system like Salesforce:

- **Missed Reminders:** Customers forget service dates → delays and unhappy customers.
- **Lost Revenue:** No follow-ups → missed sales on parts or extra services.
- **Workload Issues:** Managers can't see which mechanic is busy → some are overloaded, some idle.
- **Payment Tracking:** Manual records → mistakes and delays in accounting.
- **Customer Experience:** Only phone/walk-in booking → no easy updates or transparency.

Example: Like small bike repair shops where customers keep calling to check their service.

Screenshots: Show manual logs or notes of bookings, sample calls, or missed payment records.

2. Industry Analysis

- **Demand for Timely Service:** Customers want reminders for brakes, gears, tires.
- **Digital Preference:** People like online booking, app notifications, e-invoices.
- **Customer Retention:** Shops using CRM keep customers coming back; manual shops lose them.

Comparison:

- *Old-style Shops:* Manual, cash-only, no reminders.
- *Modern Shops:* Automated bookings, dashboards, reminders, digital payments.
- *Cycle Care:* In-between — needs Salesforce to stay competitive.

Screenshots: Market examples, competitor website snapshots, notes on trends.

3. Requirement Gathering

Functional: Booking system, service tracking, payment tracking, automated reminders, reports.
Non-functional: Easy to use, secure, scalable, integrates with email/SMS/payment.

Screenshots: Excel/Word requirement notes, flowchart of old vs new process.

4. Stakeholder Analysis

- **Admin:** Full system control.
- **Manager:** Assign jobs, track workload, approve tasks.
- **Mechanics:** Update task status, log parts.
- **Customers:** Book services, get reminders, track payments.

Screenshots: Role chart, sample user accounts, process flow.

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, Manager, Mechanic (+ optional customer portal).
- Assign proper licenses for access.

5. Profiles & Roles

- **Profiles:** Control what each user can do.
  - Mechanic: update service only
  - Manager: approve jobs
- **Roles:** Show hierarchy (Admin → Manager → Mechanic → Customer)

6. OWD & Sharing Rules

- **OWD:** Private for service records.
- **Sharing:** Manager sees all, mechanic sees assigned only, customer sees own records.

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

Phase 3: Data Modeling & Relationships

1. Custom Objects

- Customer, Bike, Service Booking, Payment.
- Separates data, avoids duplication, easy reporting.

2. Fields

- **Customer:** Name, Phone, Email, Address
- **Bike:** Model, Frame No., Purchase Date, Warranty
- **Service Booking:** Date, Status, Mechanic assigned
- **Payment:** Amount, Mode, Status, linked booking

3. Record Types

- Regular Service vs Emergency Repair → helps route tasks and track SLA.

4. Page Layouts

- **Mechanic:** Show only assigned tasks, parts.
- **Manager:** Show all booking info, payments, approvals.

5. Schema Builder (ERD)

- **Relationships:**
  - Customer → Bike: Lookup (one customer may have many bikes)
  - Bike → Service Booking: Master-Detail (booking can't exist without bike)
  - Service Booking → Payment: Master-Detail
- **Why:** Proper relationships help reporting and automation.

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

Phase 4: Process Automation (Admin)

1. Validation Rules

- Prevent saving a service booking without a bike.
- Formula: `ISBLANK(Bike__c)`
- **Why:** Ensures no orphan bookings.

2. Workflow Rules

- Send automatic email after booking is created.

3. Process Builder / Flow

- Automated service reminders:
  - Regular service → 3-day reminder
  - Emergency → no reminder

4. Approval Process

- Repair estimate > ₹5,000 → manager approval required before work starts.

5. Email Alerts & Notifications

- Templates for booking confirmation, reminders, approval updates.

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

Phase 5: Apex Programming (Developer)

1. Trigger

- Auto-create Payment record when a booking is made.

2. SOQL Queries

- Fetch last 5 services for a bike.
- Aggregate revenue by service type.

3. Batch / Queueable Apex

- Send monthly maintenance reminders asynchronously.

4. Exception Handling

- Use `try/catch` to avoid errors in triggers or batch jobs.

5. Test Classes

- Cover code > 75%
- Test bulk records, normal and edge cases.
- Assert outcomes like payment creation or reminders sent.

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

Phase 6: User Interface Development

1. Lightning App Builder

- App for Customers, Bikes, Bookings, Payments.
- Easy navigation for each role.

2. Record Pages

- Booking page shows customer, bike, status, parts, payment, history.
- Path component shows booking progress.

3. Tabs & Utility Bar

- Tabs: Customers, Bikes, Bookings, Payments, Reports
- Utility Bar: Notes, Timer, Chat, Quick Booking

4. LWC — "Book a Service"

- Customers/staff can book service through a small component.
- Shows form, validates input, shows success message.

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

Phase 7: Integration & External Access

1. Named Credentials

- Connect securely to Razorpay/Stripe for payments.

2. REST APIs

- Allow external apps to create bookings using JSON:

```
{ "customerId": "CUST-1001", "bikeId": "BIKE-203", "serviceType": "Regular
Service", "serviceDate": "2025-09-22" }
```

3. Platform Events

- Notify manager instantly if a VIP customer books service.

4. API Limits

- Daily and per-transaction limits apply.
- Use batching/events to stay within limits.

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

Phase 8: Data Management & Deployment

1. Data Import Wizard

- Import customers and bikes quickly using CSV.

2. Data Loader

- Bulk load many service bookings.

3. Duplicate Rules

- Prevent duplicate bikes (e.g., same frame number).

4. Data Export

- Backup all data (Customers, Bikes, Bookings, Payments).

5. Deployment

- Use Change Set or SFDX to move metadata from Sandbox → Production.

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

Phase 9: Reporting & Dashboards

1. Reports

- Bookings per month, revenue by service type, customer retention.

2. Dashboards

- Charts for bookings, revenue, retention in one view.

3. Dynamic Dashboards

- Mechanic sees only their tasks.
- Manager sees all bookings, revenue, team performance.

4. Security Review

- Field-level security: sensitive fields hidden for some roles.
- Audit trail tracks changes.

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

Phase 10: Final Presentation & Demo

1. Pitch Presentation

- 6–7 slides: intro, problem, solution, features, stakeholders, integrations, expected results.

2. Demo Walkthrough

- Step 1: Booking → Step 2: Approval (if needed) → Step 3: Service → Step 4: Payment → Step 5: Notifications/Dashboard

3. Feedback Collection

- Mentor form to rate usability, workflow, UI, integrations.

4. Handoff Document

- Small user guide: login, create booking, update status, approve, generate reports.

Screenshots: Sample PowerPoint slide, demo video stills, feedback form, user guide page.
