Phase 5: Apex Programming (Developer)

This phase covers advanced Salesforce developer tools, including coding with Apex for automation, triggers, asynchronous processing, and custom logic.

For the **Customer Feedback CRM – Complaint & Resolution Tracker**, **Apex is not required**, as all necessary functionality has been implemented using **admin tools** like Flows, Approval Processes, Email Alerts, Field Updates, Tasks, and Custom Notifications.

---

1. Classes & Objects

- **Purpose:** Encapsulate logic in reusable code.
- **Why not used:** All business rules and automation (assignment, notifications, approvals) are handled via **Record-Triggered Flows**. No custom classes are required.

---

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

- **Purpose:** Run custom logic automatically when records change.
- **Why not used:** Record-Triggered Flows replace the need for triggers in this project. All automation (assigning complaints, sending emails, creating tasks) is implemented with **Flows**.

---

3. Trigger Design Pattern

- **Purpose:** Best practice for organizing multiple triggers on one object.
- **Why not used:** No Apex triggers are created, so trigger patterns are not required.

---

4. SOQL & SOSL

- **Purpose:** Query Salesforce records (SOQL) or search text across objects (SOSL).
- **Why not used:** Flows natively access **record fields** and related records without the need for custom queries.

---

5. Collections: List, Set, Map

- **Purpose:** Handle multiple records efficiently in code.

- **Why not used:** All record collections are managed automatically by **Flow elements** like loops and assignments.

---

6. Control Statements

- **Purpose:** Conditional logic (if-else, loops) in Apex.
- **Why not used: Flow decisions** replace the need for Apex conditional logic for this project.

---

7. Batch Apex

- **Purpose:** Process large volumes of data asynchronously.
- **Why not used:** The project does not require mass processing; Flows handle individual record updates and actions efficiently.

---

8. Queueable Apex

- **Purpose:** Run asynchronous jobs for complex processing.
- **Why not used:** No asynchronous or background processing beyond standard Flows and Approval actions is needed.

---

9. Scheduled Apex

- **Purpose:** Execute code at a scheduled time.
- **Why not used:** Complaint notifications, approvals, and tasks run in real time via **Flows**; scheduling is not required.

---

10. Future Methods

- **Purpose:** Asynchronous execution for long-running operations.
- **Why not used:** No heavy processing or integration tasks require asynchronous handling in this project.

---

11. Exception Handling

- **Purpose:** Catch errors in Apex code.
- **Why not used:** No custom code is written; Flows and Approval processes handle errors automatically.

---

12. Test Classes

- **Purpose:** Required for deploying Apex code to production.
- **Why not used:** No Apex classes or triggers exist, so test classes are unnecessary.

---

13. Asynchronous Processing

- **Purpose:** Handle operations that take time or run in background.
- **Why not used:** All actions (emails, tasks, notifications) are managed via **admin automation tools**, which execute efficiently in real time.

Conclusion

For the **Customer Feedback CRM – Complaint & Resolution Tracker**, **Phase 5 (Apex Programming) is not required**. All business logic, automation, and notifications are implemented entirely using **admin tools**.