PHASE - 5

Introduction

Phase 5 of a typical Salesforce project usually focuses on Apex Programming, including writing Apex classes, triggers, asynchronous jobs, and test classes. However, for this project, Apex programming was determined to be unnecessary. All requirements were successfully implemented using Salesforce's declarative (point-and-click) features such as Validation Rules, Workflow Rules, Process Builder, and Flow Builder.

Reason for Skipping Apex

Apex is generally required when business logic is too complex for declarative automation or when handling very large data volumes. In this project, however, the automation requirements — such as Task reminders, escalations, approval flows, and notifications — were fully achievable with declarative tools.

To keep the project lightweight, maintainable, and aligned with Salesforce best practices, Apex programming (classes, triggers, batch jobs, etc.) was intentionally skipped.

Declarative Features Used Instead of Apex

The following declarative features replaced the need for Apex:

- Validation Rules → enforce data quality (e.g., mandatory fields before completing a Task).
- Workflow Rules → send basic reminder alerts (legacy, included for demonstration).
- Process Builder → simple record-based automation (with migration planned to Flows).
- Flow Builder → core automation engine for reminders, escalations, and approvals.
- Approval Process → manage sign-offs on Tasks when required.

- Email Alerts & Templates → send automated task reminder emails.
- Custom Notifications → deliver real-time alerts to users in Salesforce and mobile.

Project Alignment

By skipping Apex programming, the project stays aligned with its academic/demo objectives. The focus remains on showcasing Salesforce Admin and Declarative Development skills, which were sufficient to meet all requirements around Task reminders, overdue escalations, and user approvals**.**