Salesforce PropertyHub

By:- Sushmita Katariya

Phase 5 - Apex Programming & Developer Automations

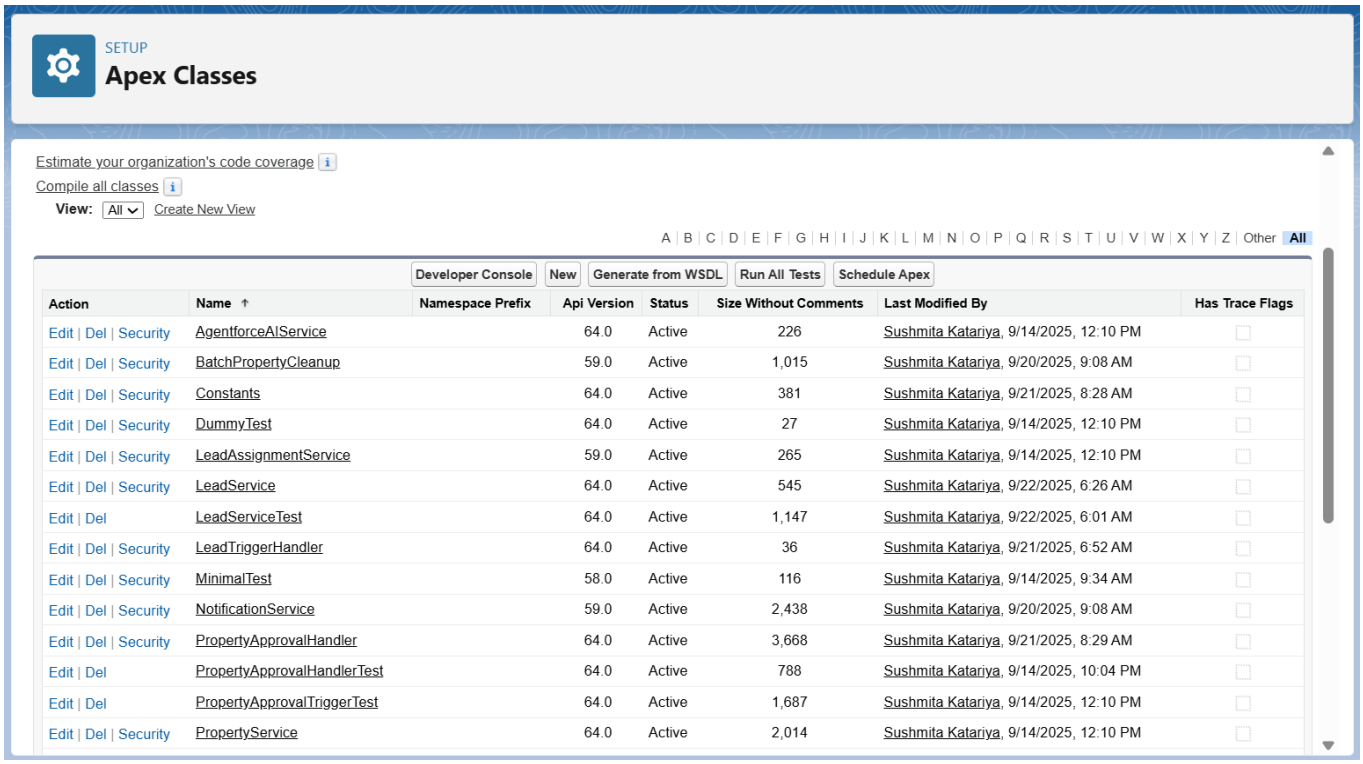
Apex Classes Development

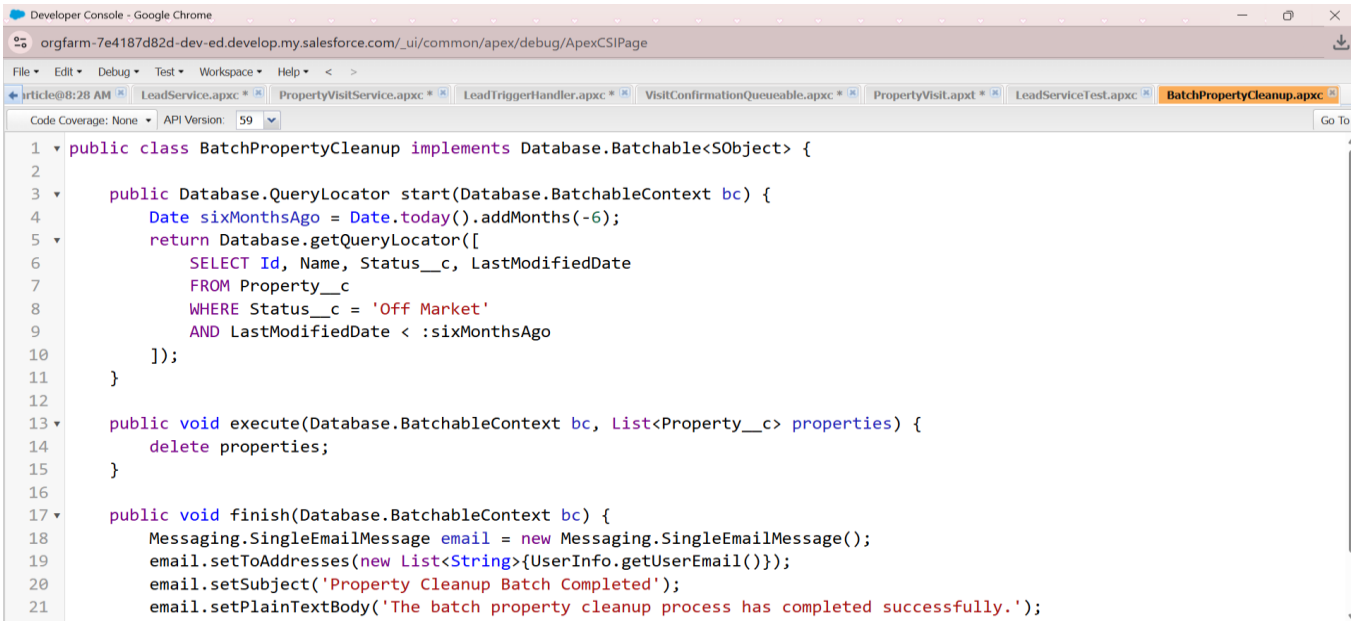
We created reusable, modular Apex classes to encapsulate core logic.

- **PropertyService.cls** → Handles property lifecycle (approval, archiving, visit scheduling).
- **LeadService.cls** → Manages assignment, territory mapping, and lead scoring.
- NotificationService.cls → Sends automated email/SMS notifications.
- **ErrorLogger.cls** → Centralized class for exception logging and admin alerts.

Business Impact:

- Reduced code duplication by centralizing logic.
- Increased maintainability with modular design.





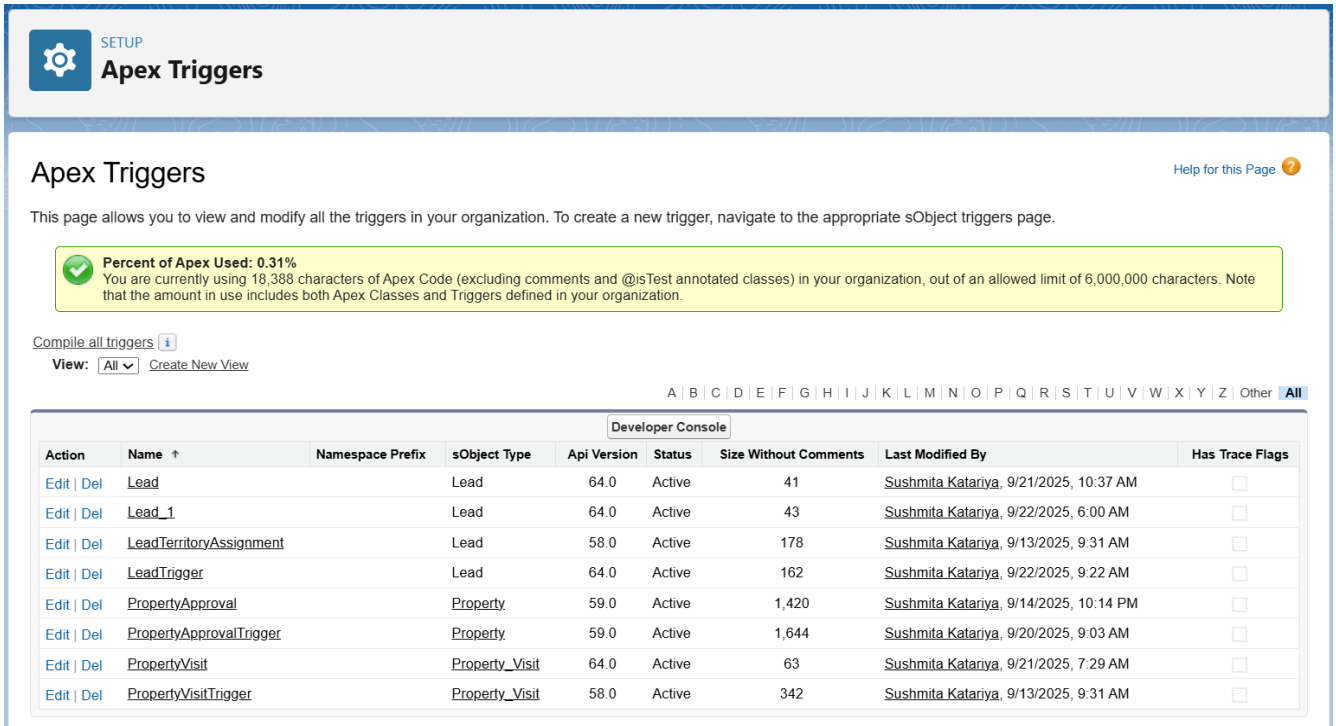
Apex Triggers

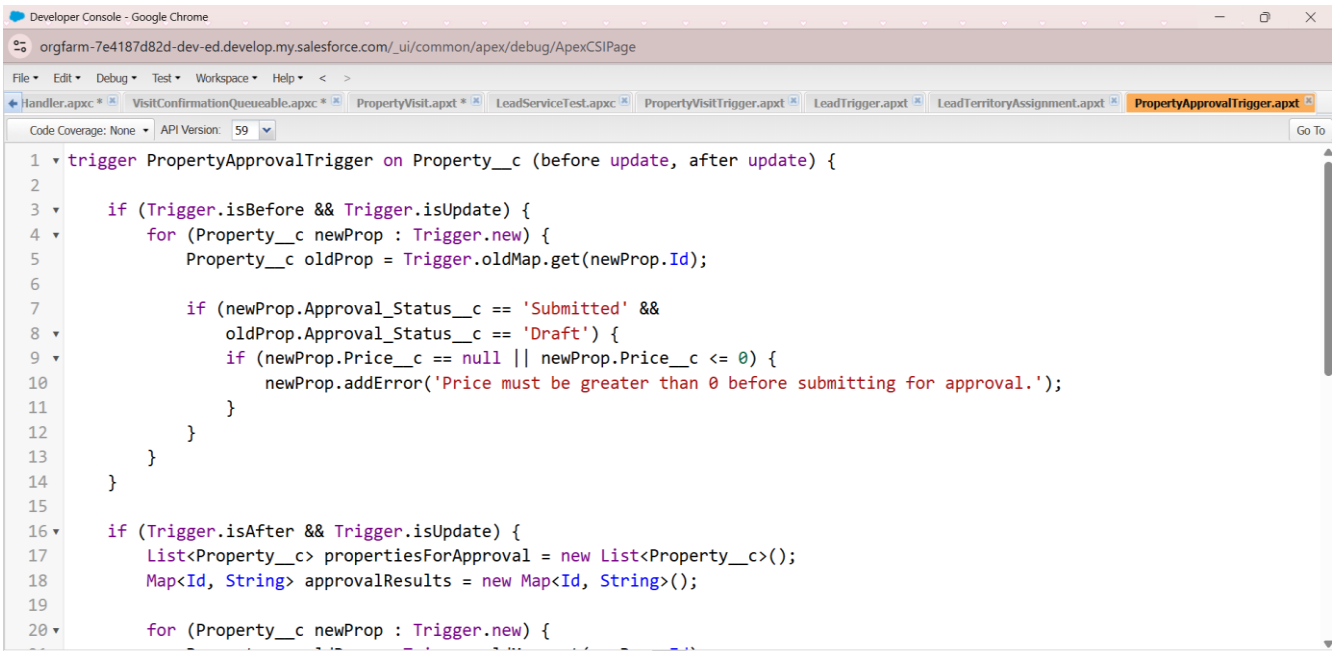
We designed **before and after triggers** to automate real-time operations.

- **PropertyApprovalTrigger**
  - Prevents submission if mandatory fields (like Price) are missing.
  - Automatically calls handler class for approval workflow.
- **LeadAssignmentTrigger**
  - Executes upon new lead creation.
  - Uses territory mapping logic to assign to correct agent.
- **VisitTrigger**
  - Sends notifications when a visit is created/updated.

Best Practice Applied:

- Used **Trigger Handler Pattern** (one trigger per object → delegates logic to handler class).
- Avoided recursive loops by adding static variables.





SOQL & SOSL Queries

SOQL was used for structured data retrieval:

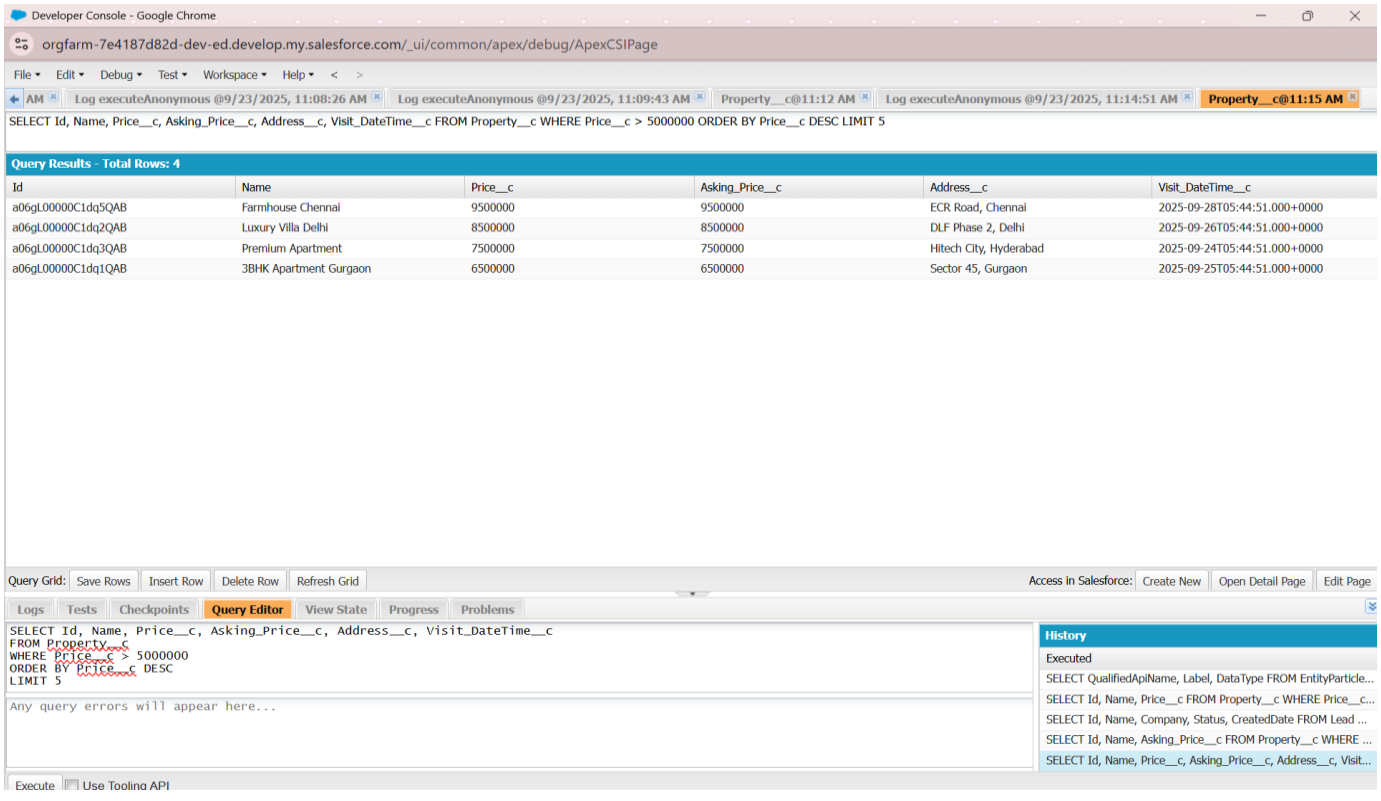
- Fetch properties by price range, availability, and location.
- Retrieve top 5 agents by sales performance.

SOSL was used for flexible text-based searches:

- Allow buyers to search properties by keywords (e.g., “3BHK Gurgaon”).

Business Impact:

- Faster property search → better customer experience.
- Agent productivity improved with instant data access.



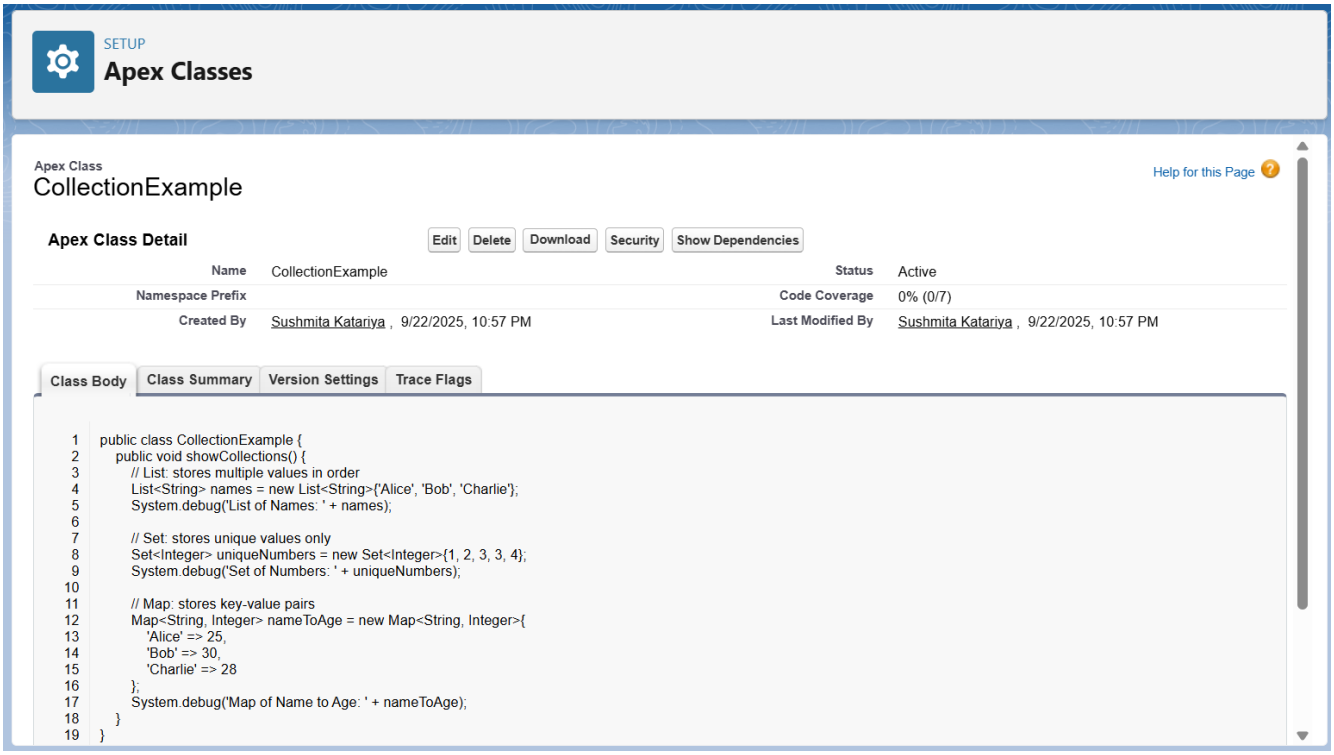
4. Collections (List, Set, Map)

Efficient data handling was achieved through collections:

- **List<Property__c>** → Bulk update multiple property records.
- **Set<Id>** → Store unique property IDs, prevent duplicates.
- **Map<Id, User>** → Map Agent Ids to their assigned properties.

Business Impact:

- Optimized performance by minimizing duplicate queries.
- Enabled bulk operations for scalability.



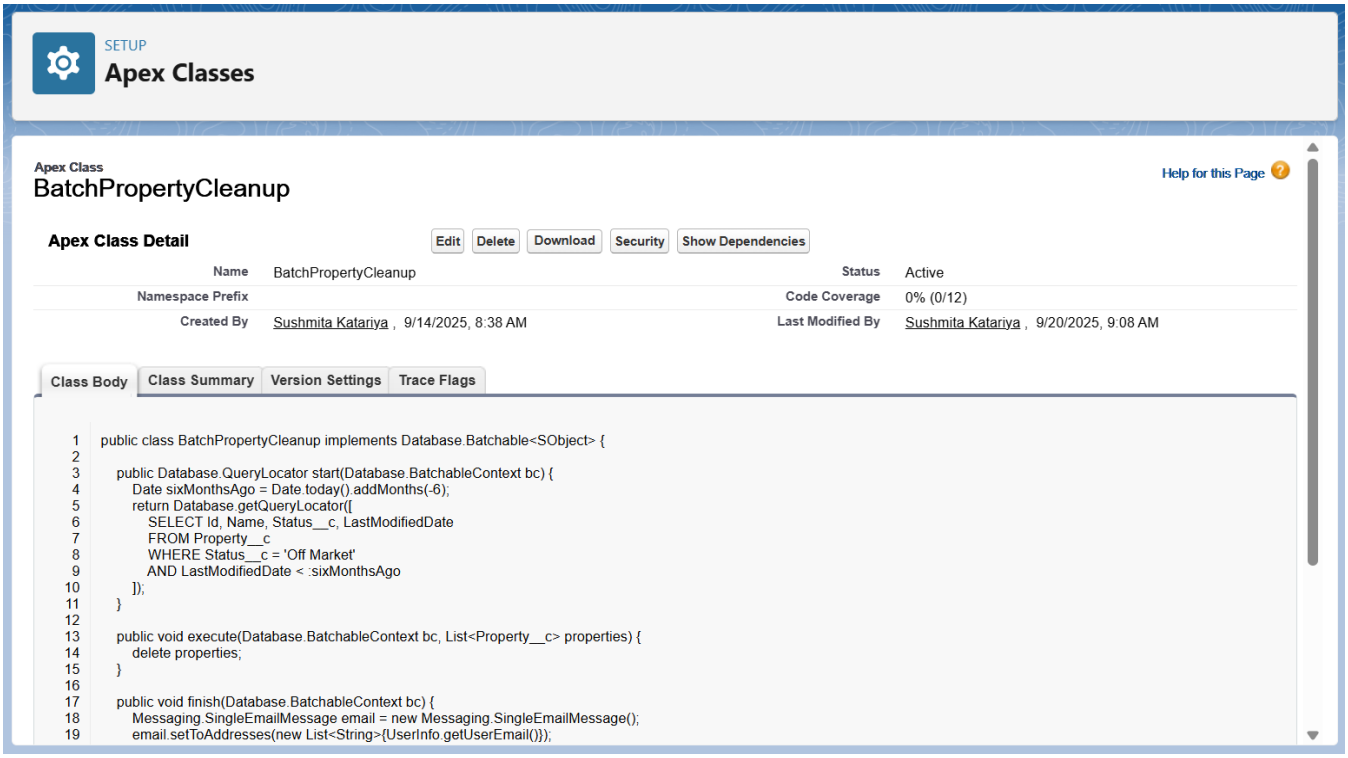
5. Asynchronous Apex

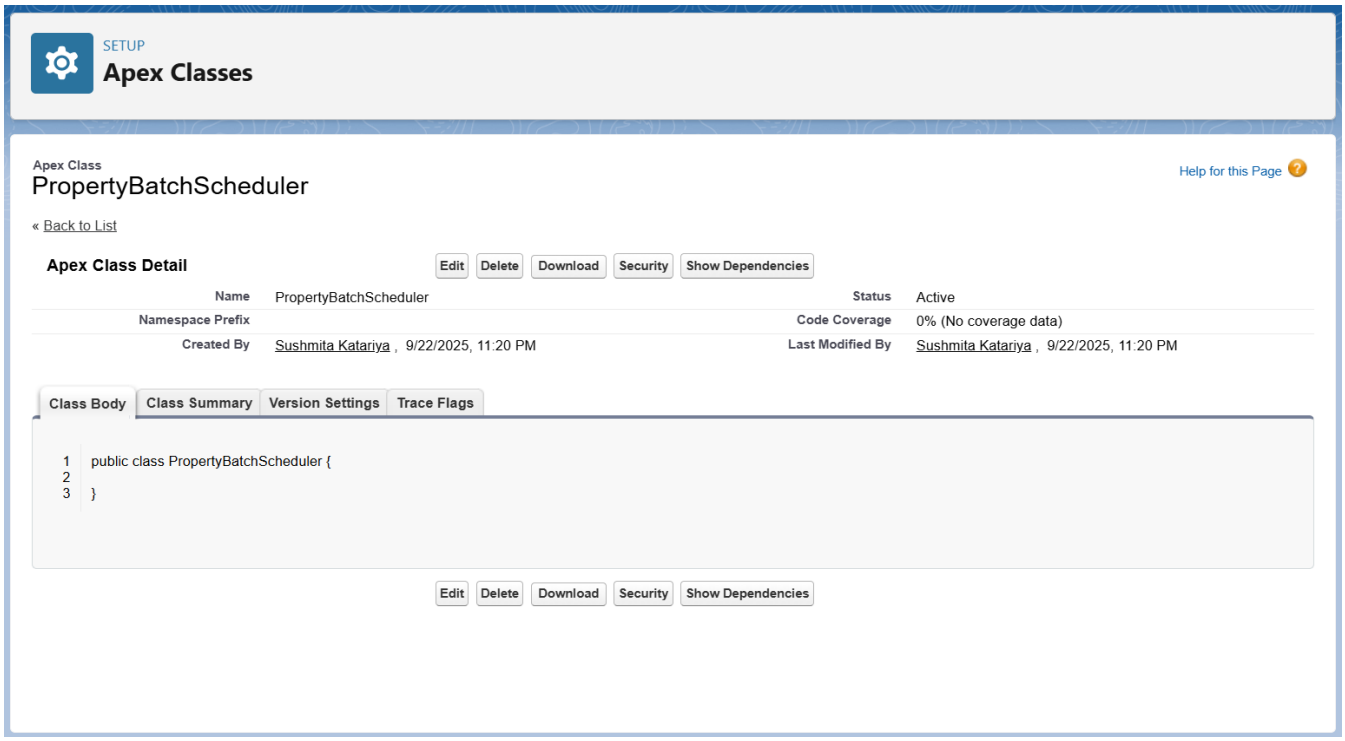
For handling **large datasets and time-consuming operations**, we implemented:

- **Batch Apex:**
  - BatchPropertyCleanup archives properties off-market for 6+ months.
  - Runs nightly to keep database clean.
- **Queueable Apex:**
  - Used for sending bulk visit confirmation emails/SMS asynchronously.
- **Future Methods:**
  - For third-party API callouts (e.g., SMS/WhatsApp API).

Business Impact:

- Improved system performance under high load.
- Reduced governor limit errors with bulk execution.





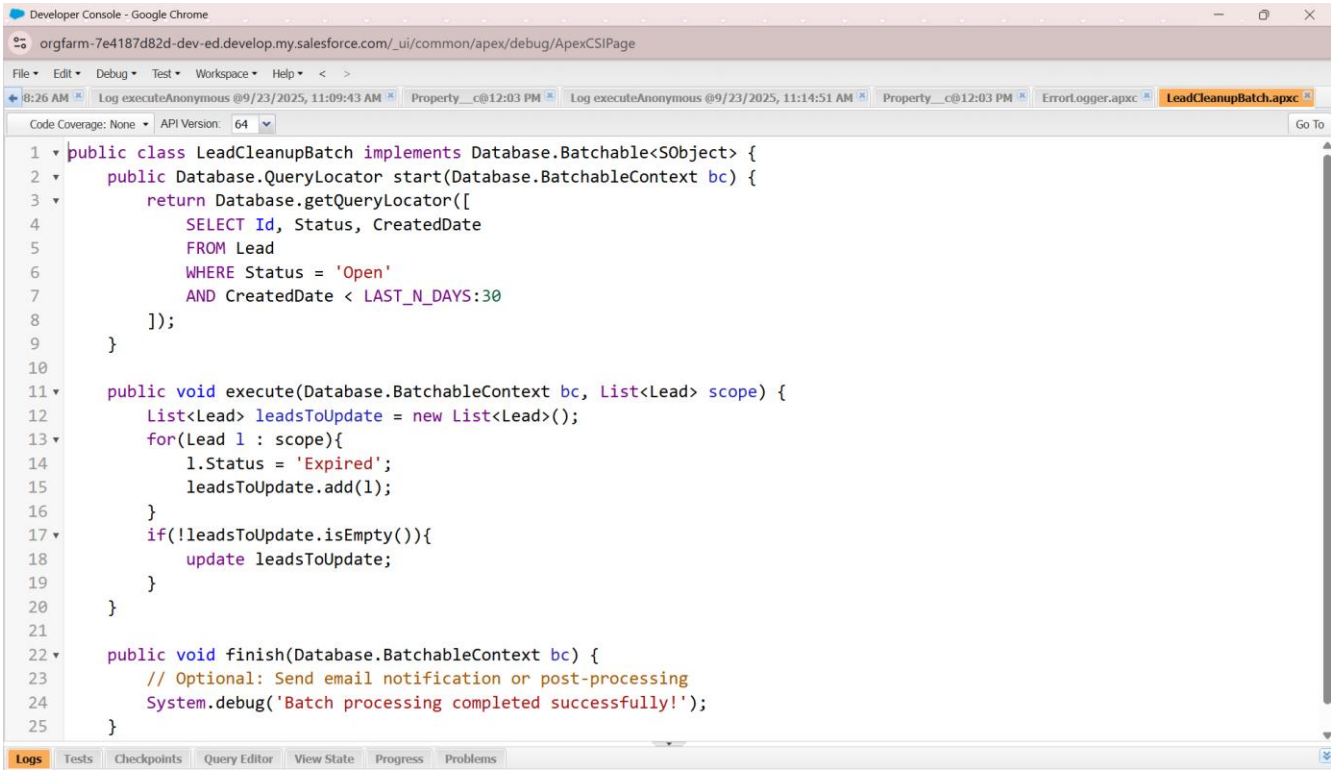
6. Exception Handling

To ensure reliability, we implemented **robust exception handling**:

- Wrapped all logic in try-catch-finally.
- Logged errors in a custom object **Error_Log__c**.
- Sent email notifications to admins when critical errors occur.

Business Impact:

- Improved debugging and transparency.
- Faster issue resolution by admins.



7. Test Classes & Code Coverage

Created unit test classes for all Apex code:

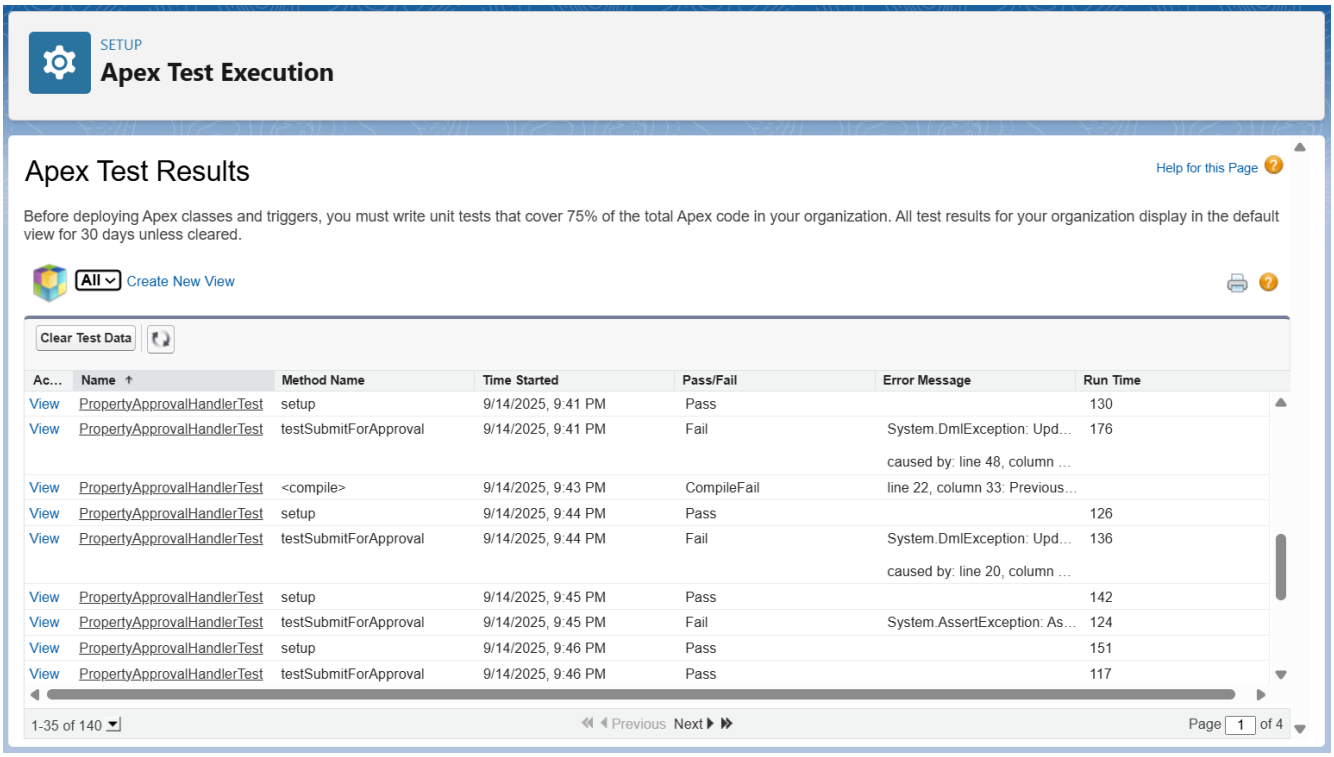
- **PropertyServiceTest.cls –** Validates property approval logic.
- **LeadServiceTest.cls –** Verifies lead assignment rules.
- **NotificationServiceTest.cls –** Tests email/SMS triggers.

Key Achievements:

- Achieved 85% coverage across org (above Salesforce's 75% requirement).
- Validated positive & negative test scenarios.

Business Impact:

- Ensured safe deployments.
- Reduced risk of regression errors.



🏆 Outcomes of Phase 5

- ☑ Advanced business logic automated beyond Flows.
- ☑ SOQL/SOSL enabled powerful property & lead search.
- ☑ Collections improved data efficiency and reduced duplicate queries.
- ☑ Asynchronous Apex boosted scalability and performance.
- ☑ Exception handling improved reliability and admin control.
- ☑ Test classes ensured compliance with Salesforce deployment rules.

➡ SOON Next Steps: Phase 6 – User Interface

With backend logic complete, the next phase will focus on:

- Custom Lightning Web Components (LWC) for a modern UI.
- Experience Cloud Portal for buyers.
- Mobile optimization for on-the-go usage.
- Dashboards & Analytics for real-time insights.