

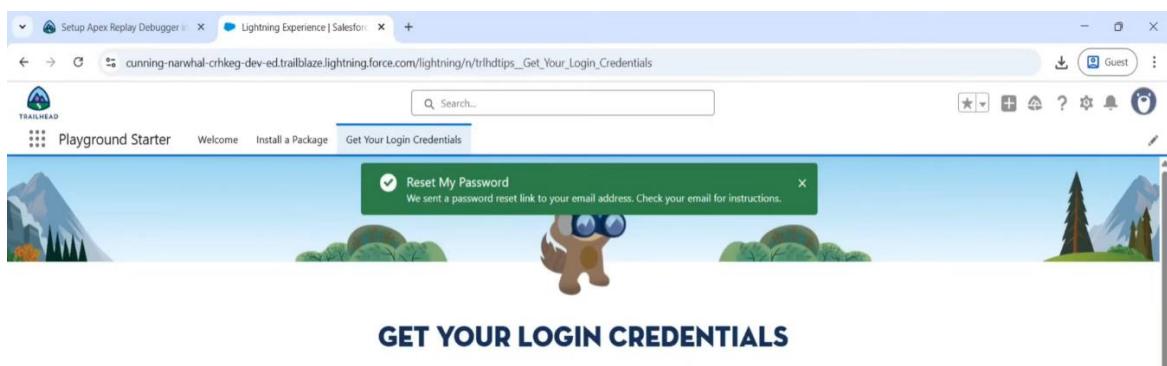
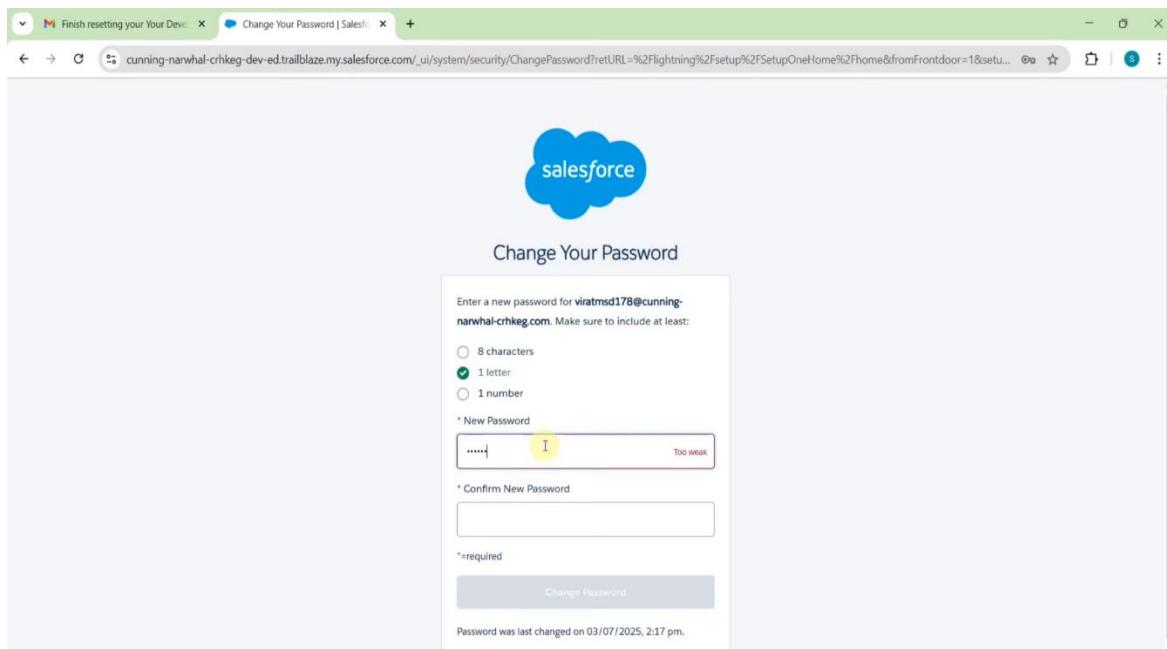
PHASE 4: Process Automation (Admin)

Debug Log Analysis and Bug Fixing in Salesforce Apex

1. Objective of Phase 4

Although this project mainly focuses on **Apex programming and debugging**, basic **admin-level automation** is required to support test executions, logging, and code behavior verification.

In this phase, we configure Salesforce automation tools—**Flows, Validation Rules, Assignment Rules, and Debug Log settings**—to ensure that Apex code executes correctly and produces replay-enabled debug logs for analysis.



2. Debug Log Automation

Replay Debugger requires **continuous and consistent debug logs**. Therefore, admin automation includes:

a. Creating a Trace Flag

- Navigate to: **Setup → Debug Logs**
- Add your user in "Monitored Users"
- Set log levels:
 - **Apex Code = FINEST**
 - **Visualforce = FINER**
 - System = FINEST (optional)

This ensures that every Apex test produces a replay-enabled log.

b. Setting Auto-Expiration

- Trace flags are valid for **30 minutes**
- Admin settings help auto-refresh logs when needed

Purpose:

To make sure every time you run the Apex test, the system automatically generates detailed logs needed by the Apex Replay Debugger.

3. Optional Flow for Pre-Populating Account Fields

Although this project uses Apex to create accounts, a simple Flow can be configured to:

- Pre-fill Account fields
- Validate data before Apex executes
- Ensure consistent input values for debugging

4. Validation Rules (Optional)

Validation rules are created to ensure that:

- Required fields are not empty
- Test data is consistent
- Apex failures are not caused by bad input

Example:

TickerSymbol must be exactly 3–5 characters.

This helps Apex tests run smoothly without unexpected validation errors during debugging.

