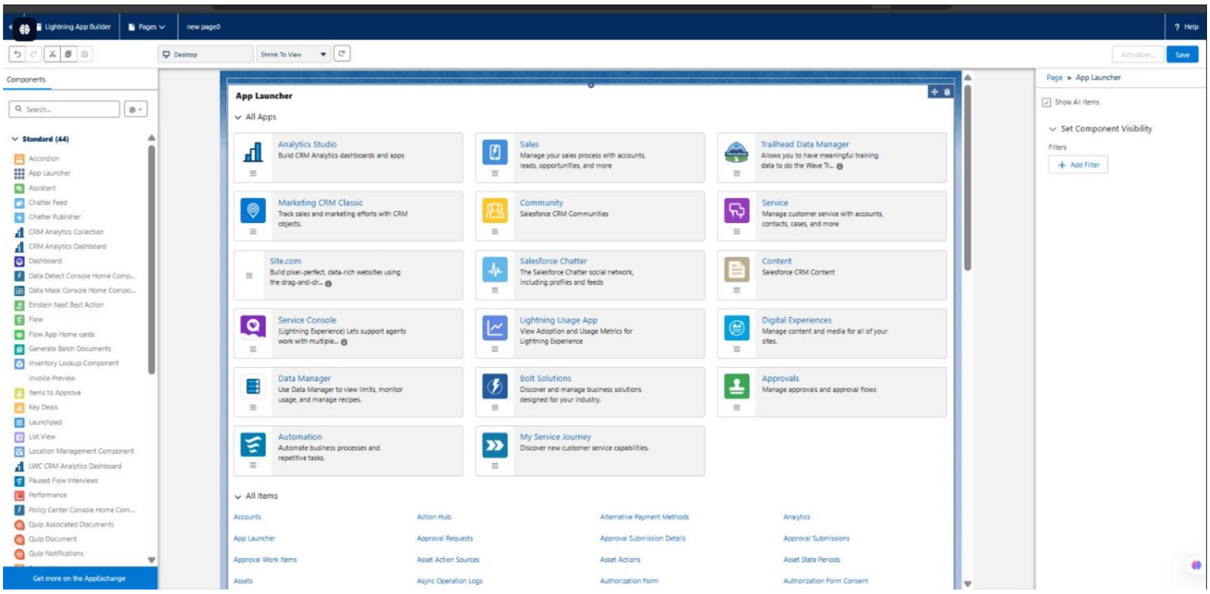
Phase 6: User Interface Development:

Name:Shaik Javeed Ahammad

1. Phase Overview

Phase 6 focuses on the finalisation, testing, and deployment of the Expense Approval System in Salesforce. This phase ensures that the system is fully functional, secure, scalable, and user-friendly. It also includes end-to-end testing, validation of business rules, performance optimization, and user training. The goal of Phase 6 is to transition the system from development into a production-ready solution that eliminates inefficiencies in expense management.



2. Objectives of Phase 6

• Complete System Testing: Verify that all triggers, flows, workflows, and email notifications are functioning as expected.

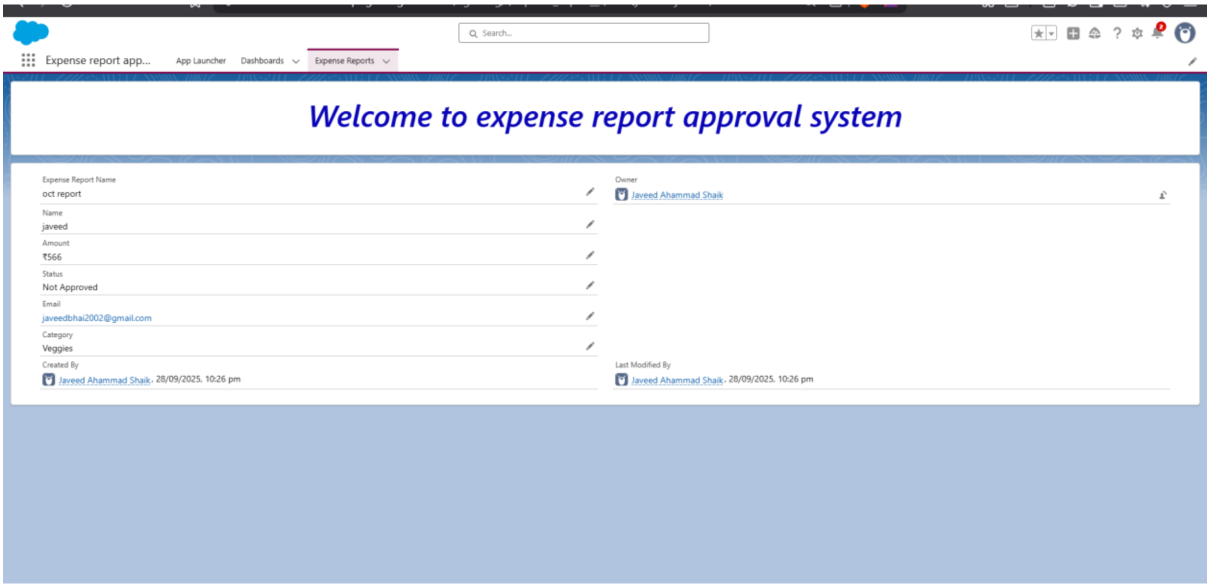
• Performance Optimization: Ensure the system handles large volumes of expense records without delays.

• Security Validation: Confirm that access controls, data visibility, and sharing rules are correctly applied.

• User Acceptance Testing (UAT): Allow end-users to test real-world scenarios and provide feedback.

• Deployment Preparation: Prepare metadata, triggers, and flows for deployment to production.

• Documentation and Training: Provide detailed manuals, guidelines, and training sessions for end-users and administrators.



3. System Components Finalized in Phase 6

3.1. Salesforce Objects

• Expense Request Object: o Fields: Name, Amount, Expense Type, Status, Submission Date, Comments.

o Relationships: Linked to Employee object.

o Validation Rules: Ensure Amount is positive and Expense Type is selected.

• Employee Object:

o Fields: Name, Employee ID, Department, Email, Manager.

o Relationships: Lookup with Expense Request object for reporting.

3.2. Apex Triggers

• ExpenseRequestTrigger: Handles automation when a new expense is submitted or updated.

Before Insert: Set default Status to Pending. o Before Update: Prevent status change without proper approvals

After Insert/Update: Send email notifications to managers or employees based on status.

• Error Handling: Proper try-catch blocks to handle exceptions.

• Bulkification: Optimized to handle multiple records in a single transaction to comply with Salesforce governor limits.

3.3. Flows

• Approval Flow: Automates expense approval without using Salesforce standard Approval Processes.

o Trigger: Record creation (Expense Request)

o Decision Element: Checks Amount and Expense Type.

o Update Records: Sets Status to Approved or Rejected based on manager decision.

o Email Alerts: Sends notifications upon approval or rejection.

3.4. Email Notifications

• Triggered via Action – Send Email in Flows.

• Emails include:

o Expense amount and type

o Status updates o Manager comments

• Configured without using templates to allow dynamic content insertion.

4. Testing Strategy

4.1. Unit Testing

• Each Apex class and trigger tested independently.

• Test cases include:

o Submitting new expense requests

o Updating existing requests

o Invalid data handling (e.g., negative amounts)

4.2. Integration Testing

• Validates the interaction between Apex triggers, Flows, and email notifications.

• Ensures that the sequence of actions executes correctly across multiple objects.

4.3. User Acceptance Testing (UAT)

• Conducted with selected employees and managers.

• Focus on:

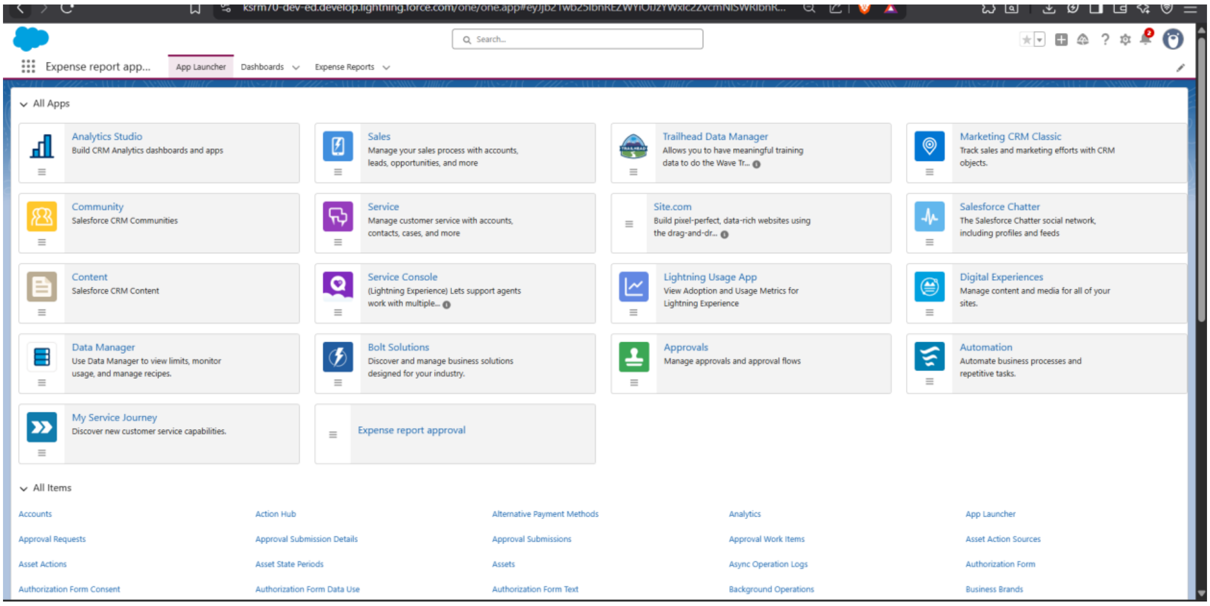
o User interface usability o Accuracy of status updates

o Timeliness of email notifications

4.4. Performance Testing

• Checks system behavior under bulk record creation.

• Ensures triggers and flows do not exceed governor limits.



5. Security and Compliance

• Profiles and Permission Sets: Ensure only authorized users can submit, approve, or view expense requests.

• Field-Level Security: Sensitive fields like Amount and Employee ID are restricted.

• Data Sharing Rules: Managers can only see expense requests from their teams.

• Audit Trail: Track all changes to expense requests for compliance and reporting.

6. Deployment Preparation

• Sandbox Validation: Fully tested in a sandbox environment before deployment.

• Change Sets: Apex classes, triggers, flows, and object modifications included in change sets.

• Rollback Plan: Prepared to revert changes in case of deployment failure.

• Post-Deployment Checks: Validate email notifications, record creation, and status updates in production.

7. End-User Documentation and Training

7.1. User Manual

• Step-by-step guide to create, view, and manage expense requests.

• Screenshots included for clarity.

• Detailed explanation of status values: Pending, Approved, Rejected.

7.2. Training Sessions

• Hands-on sessions for employees and managers.

• Emphasis on:

o Submitting requests correctly

o Reviewing requests
o Understanding system notifications

7.3. FAQs
• How to handle rejected requests
• How to update comments
• How to track expense status

8. Lessons Learned
• Automation vs Manual: Automating approvals significantly reduces human error.
• Flow Limitations: Complex approval logic may require Apex triggers for flexibility.
• Bulk Data Handling: Always design triggers with bulkification in mind.
• End-User Feedback: Early user feedback ensures higher adoption rates.

9. Future Enhancements
• Integration with ERP for automatic reimbursement.
• Mobile-friendly interface for expense submission.
• Advanced reporting dashboard with analytics on expenses.
• Multi-level approval flows based on amount thresholds.

10. Conclusion Phase 6
ensures the Expense Approval System is production-ready, secure, and efficient.
It delivers:
• Faster approvals
• Reduced errors
• Enhanced transparency
• Better compliance The system is now capable of handling all organizational expense approval scenarios and can be scaled as the organization grows.