

Project Documentation: Expense Request Approval Process

Name: Shaik Javeed Ahammad

1. Introduction

Managing expenses in an organization is a crucial process. Traditionally, employees submit requests for purchases, which are manually checked against budgets, approved or rejected by managers, and then communicated back. This manual approach is **time-consuming, errorprone, and inefficient**.

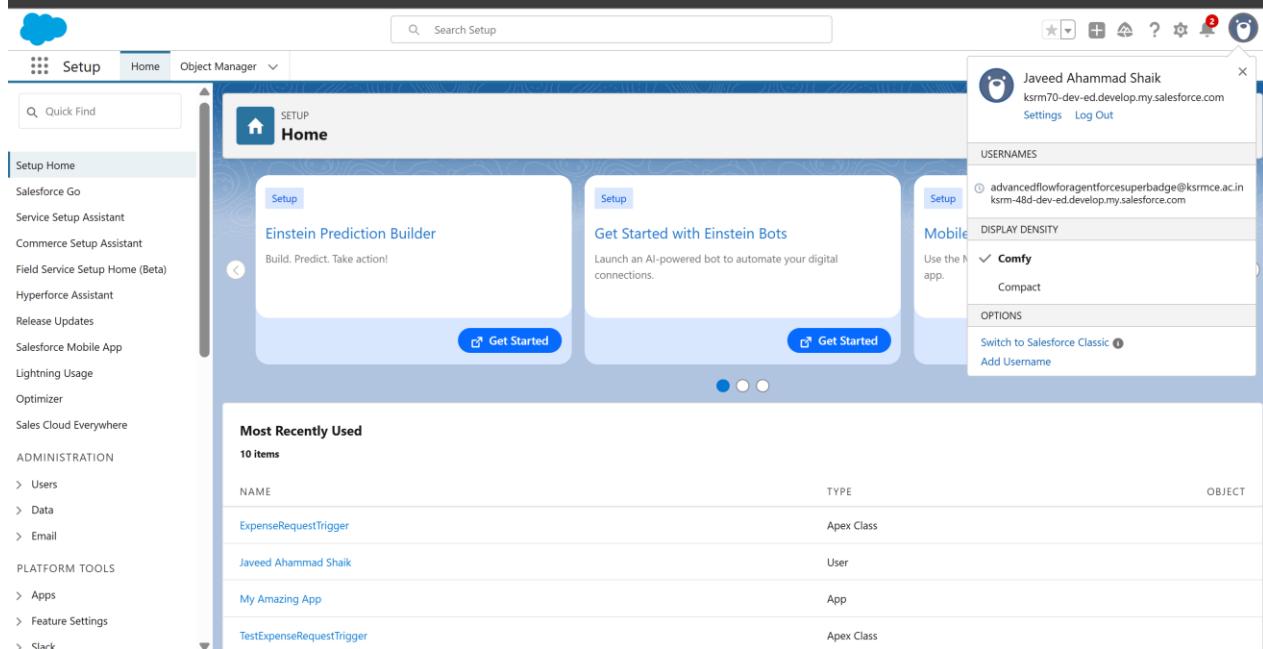
This project implements a **Salesforce Record-Triggered Flow** that automates the **approval/rejection process based on expense category and amount**, and sends an email to the requester using **Send Email action** (without using a template).

Objectives:

- Automate approvals for category-based expenses
 - Notify requesters immediately via email
 - Minimize manual intervention
 - Keep the object and flow simple (Name, Amount, Category, Status)
-
- To begin setting up your Salesforce environment, start by accessing your Salesforce Org. Open your web browser and visit <https://trailhead.salesforce.com>
 - Salesforce's official learning and development platform. If you already have an account, log in using your existing credentials.
 - If not, you can easily create a Developer Edition org, which provides a free Salesforce environment specifically for learning, experimentation, and testing without affecting any production data.
 - Once your account is set up and verified, log in to your Salesforce Developer Org to begin customizing it.
-
- After successfully logging in, you'll need to access the Setup area to manage and configure your Salesforce environment.
 - In the top-right corner of the Salesforce interface, click the Gear Icon (⚙️) and select Setup from the dropdown menu.

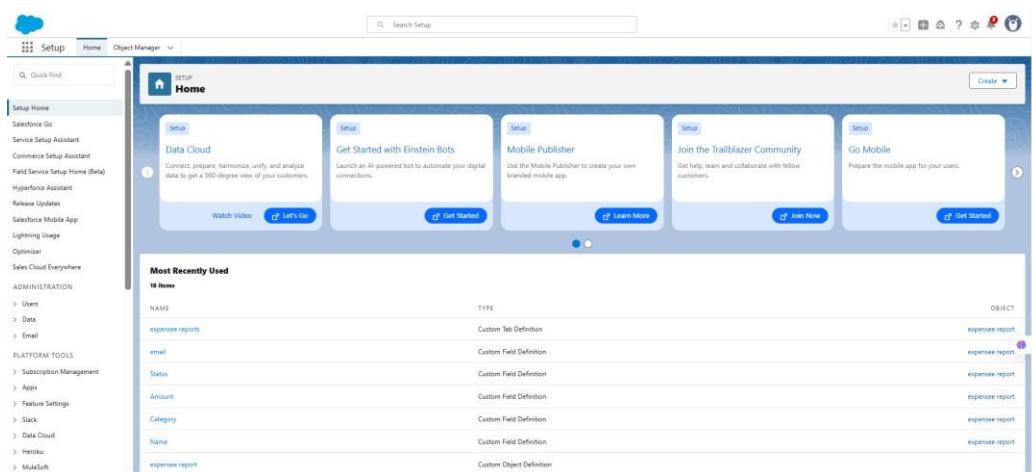
This will open the main administration and configuration page where all customization options are available. Inside the Setup page, use the Quick Find search box on the left panel to locate different configuration tools. Type Object Manager in the search bar and select it from the results. The Object Manager is

the section where you can create and manage all objects — including standard and custom ones — that define the data structure of your Salesforce org. From here, you can proceed to create custom objects such as the Expense Report object, add fields, define relationships, and configure layouts according to your project needs.



The screenshot shows the Salesforce Setup Home interface. The left sidebar has a 'Setup' icon and lists categories like 'Setup Home', 'Administration', and 'Platform Tools'. The main area displays cards for 'Einstein Prediction Builder' and 'Get Started with Einstein Bots'. Below these is a section titled 'Most Recently Used' with a table:

NAME	TYPE	OBJECT
ExpenseRequestTrigger	Apex Class	
Javeed Ahammad Shaik	User	
My Amazing App	App	
TestExpenseRequestTrigger	Apex Class	



This screenshot shows the same Salesforce Setup Home interface, but with a different set of cards in the main area: 'Data Cloud', 'Get Started with Einstein Bots', 'Mobile Publisher', 'Join the Trailblazer Community', and 'Go Mobile'. The 'Most Recently Used' table is identical to the one in the first screenshot.

To create a custom object named `Expense_Report__c` in Salesforce, begin by navigating to the Setup area of your Salesforce environment. Once there, open the Object Manager tab from the Setup menu. At the top-right corner, click on the Create button and select Custom Object from the available options. This opens a form where you can define the basic attributes and structure of the new object.

In the form, enter `Expense Report` as the *Label*. The label is the name that users will see in the Salesforce interface, such as in the app menu or record headers. Next, set the Plural Label to `Expense Reports`, which will be displayed when multiple records of this object appear in list views or related lists. Salesforce will automatically generate an API Name, which in this case will be `Expense_Report__c`. The “`_c`” suffix indicates that this is a custom object created by the user.

For the Record Name field, retain the default label as *Name* and set its data type to *Text*. This field acts as the primary identifier for each record, meaning every Expense Report record will be uniquely identified by this name.

To keep the initial setup simple, leave all optional features unchecked. These include features such as allowing reports, activities, or field history tracking. While these can be useful for advanced functionality, they can be enabled later once the core object is properly configured. By excluding these options for now, you maintain a clean and straightforward object structure that is easier to manage in the early stages of configuration.

Finally, click the Save button to create the custom object. Salesforce will confirm the creation, and the new object named **Expense_Report__c** will appear under the Object Manager list. You can now proceed to define additional elements for the object, such as custom fields, relationships, page layouts, and permissions, depending on how you want to use the Expense Report in your application.

The **Expense_Report__c** object includes three key custom fields designed to capture essential information about each expense record: **Amount__c**, **Category**, and **Status__c**.

The **Amount__c** field is of the *Currency* data type, which allows users to record the monetary value of an expense with precision. It is labeled as *Amount* and configured with a length of 16 and two decimal places, enabling accurate financial entries such as 12345.67. This field ensures that all expenses are stored in a standardized financial format, supporting consistent reporting and calculations within Salesforce.

The **Category** field is defined as a *Picklist*, allowing users to classify expenses based on predefined categories. The available options — *Veggies*, *Groceries*, *Clothing*, *Savings*, *Travel*, and *Cosmetics* — provide clear classifications for different types of expenditures. This helps in filtering, grouping, and analyzing spending patterns effectively, especially when generating reports or dashboards related to expense tracking.

The **Status__c** field, also a *Picklist*, is used to indicate the current stage of the expense report within an approval process. It includes three values: *Submitted*, *Approved*, and *Not Approved*. These statuses provide a simple yet effective workflow indicator, allowing both employees and managers to easily track where each expense report stands in the review cycle. Together, these fields form the backbone of the Expense Report object, ensuring structured, accurate, and easily manageable data for expense management processes in Salesforce.

Fields & Relationships					
	FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
	Amount	Amount_c	Currency(18, 0)		
	Category	Category_c	Picklist		
	Created By	CreatedById	Lookup(User)		
	Email	Email_c	Email		
	Expense Report Name	Name	Text(80)		
	Last Modified By	LastModifiedById	Lookup(User)		
	Name	Name_c	Text(16)		
	Owner	OwnerId	Lookup(User,Group)		
	Status	Status_c	Picklist		

4. Build Record-Triggered Flow

To automate processes in Salesforce related to the **Expense_Report__c** object, a **Record-Triggered Flow** is used. This type of flow allows specific actions to be executed automatically whenever a new record is created, updated, or deleted. In this case, the flow is configured to trigger when a new expense report record is added, ensuring real-time automation without the need for manual intervention.

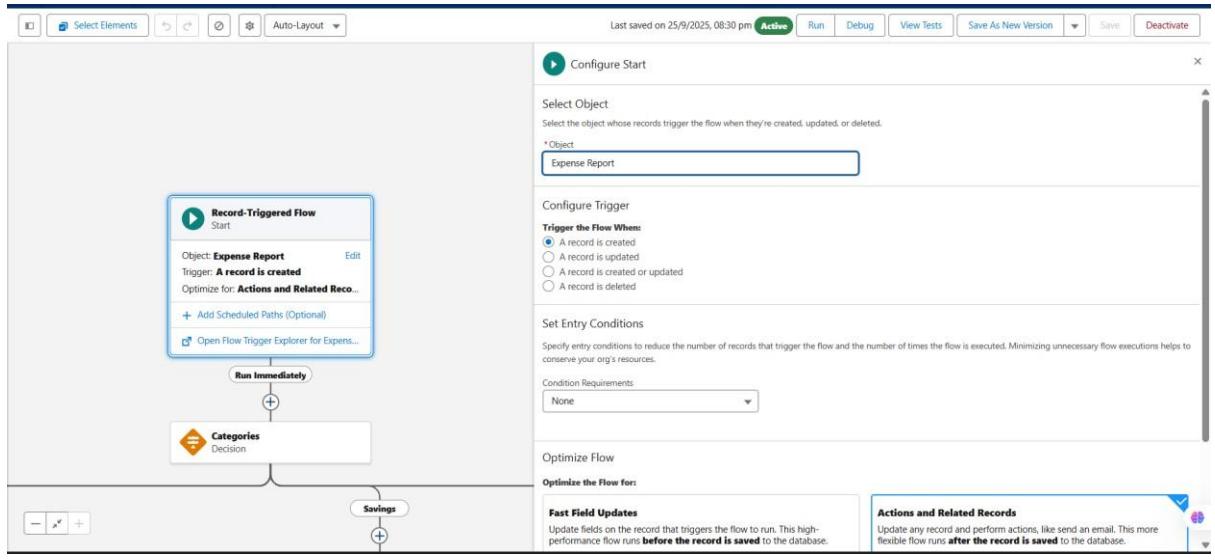
The setup begins in **Flow Builder**, Salesforce's visual automation tool that allows users to design workflows through a drag-and-drop interface. Within the **Setup** area, navigating to **Flow** and selecting **New Flow** opens a configuration window where different flow types are available. Choosing **Record-Triggered Flow** specifies that the automation will begin in response to changes in a Salesforce record.

The **object** associated with this flow is **Expense_Report__c**, which means the automation will act on expense report records. The **trigger** is set to activate **when a record is created**, ensuring that the flow runs every time a new expense report is entered into the system. Since no specific conditions are defined, the **entry condition** is left as "None," allowing the flow to run for all new records regardless of their field values. The flow is also optimized for **Actions and Related Records**, which ensures faster execution when performing operations such as updating related data, sending notifications, or creating dependent records.

This setup provides the foundation for building advanced automation logic — such as sending approval notifications, updating status fields, or logging related transactions — immediately after

a new expense report record is created

- Click **Done**



Step 2: Add Decision Element – Category + Amount

The **Decision element** in this flow, titled **Categories**, serves as the logical control point that determines how each expense report is processed based on its **Category** and **Amount**. Its primary purpose is to evaluate the submitted data and automatically decide whether an expense should be approved or rejected without requiring manual review. This intelligent decision-making mechanism ensures faster and more consistent handling of expense requests.

Within the **Expense Request Approval Flow**, predefined limits are set for each expense category to maintain budgetary control and prevent excessive spending. The maximum allowable limits are clearly defined — *Veggies* up to ₹500, *Groceries* up to ₹2000, *Clothing* up to ₹4000, *Savings* up to ₹8000, *Travel* up to ₹6000, and *Cosmetics* up to ₹5000. When a new expense report is created, the flow compares the **Amount_c** field value against the limit associated with the selected **Category**.

If the entered amount is **less than or equal to** the defined threshold, the flow automatically updates the **Status_c** field to *Approved* and triggers an approval notification email to the requester. Conversely, if the amount **exceeds** the category's limit, the **Status_c** field is updated to *Not Approved*, and a rejection email is sent to inform the requester about the decision.

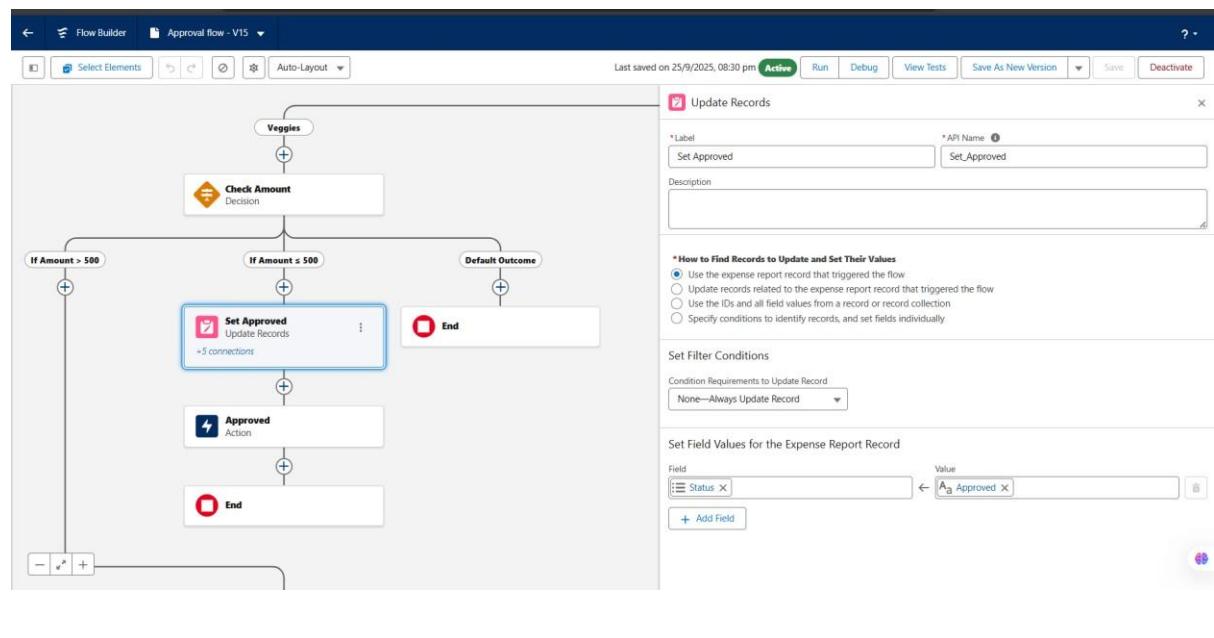
This structured and rule-based approach eliminates the need for manual expense verification for smaller claims, reduces processing delays, and ensures that every expense submission is evaluated consistently according to predefined financial policies. It enhances operational efficiency and keeps users promptly informed about the approval status of their expense reports.

Category	Maximum Amount	Status if ≤ Max	Status if > Max	Email Message if Approved	Email Message if Rejected
Veggies	500	Approved	Not Approved	“Your expense request of amount {!Amount_c} has been approved under Veggies.”	“Your expense request of amount {!Amount_c} exceeds the allowed budget for Veggies.”
Groceries	2000	Approved	Not Approved	“Your expense request of amount {!Amount_c} has been approved under Groceries.”	“Your expense request of amount {!Amount_c} exceeds the allowed budget for Groceries.”
Clothing	4000	Approved	Not Approved	“Your expense request of amount {!Amount_c} has been approved under Clothing.”	“Your expense request of amount {!Amount_c} exceeds the allowed budget for Clothing.”
Savings	8000	Approved	Not Approved	“Your expense request of amount {!Amount_c} has been approved under Savings.”	“Your expense request of amount {!Amount_c} exceeds the allowed budget for Savings.”
Travel	6000	Approved	Not Approved	“Your expense request of amount {!Amount_c} has been approved under Travel.”	“Your expense request of amount {!Amount_c} exceeds the allowed budget for Travel.”
Cosmetics	5000	Approved	Not Approved	“Your expense request of amount {!Amount_c} has been approved under Cosmetics.”	“Your expense request of amount {!Amount_c} exceeds the allowed budget for Cosmetics.”

In this stage of the **Expense Request Approval Flow**, additional refinements and record update actions are added to make the automation more reliable and precise. The **Decision element** should always include a **default outcome**, which acts as a safety net for any unexpected or undefined categories that may appear in future data entries. This ensures that the flow continues to run smoothly even if a category does not match any predefined condition. It is also good practice to assign **descriptive names** to each outcome—such as “*Veggies – Approved*” or “*Clothing – Rejected*”—so that the flow remains easy to understand and maintain, especially when multiple branches are involved.

After defining the decision branches, an **Update Records** element is added to each branch to modify the **Status_c** field based on the flow’s evaluation. On the branch where the expense falls within the permissible limit, an Update Records element is used to set **Status_c = Approved**. On the branch where the amount exceeds the allowed threshold, another Update Records element sets **Status_c = Not Approved**. These updates ensure that each expense report’s approval status is accurately reflected in real time within Salesforce.

It is important that this record update occurs **before any email notification** is sent to the user. Updating the record first guarantees that the most recent and correct status value—whether “Approved” or “Not Approved”—is included in the notification message. This sequencing avoids confusion, ensures data integrity, and provides users with consistent and accurate communication about their expense requests.



In the final step of the Expense Request Approval Flow, a **Send Email** action is added to automatically notify users about the outcome of their expense submissions. This action is configured to run immediately after the **Update Records** element in each branch—one for approved requests and another for rejected ones. The purpose of this step is to ensure that users receive instant confirmation about the status of their expense reports once the system has processed them.

The **Send Email** action is created without using a predefined email template. Instead, the email content is written directly within the flow using **HTML** formatting for better presentation. The

recipient can be a user's email address or an email field associated with the expense report record. The email subject can simply be "Expense Request Status" or include dynamic details using merge fields, such as "Expense Request Update – {!Category},” to make the message more personalized.

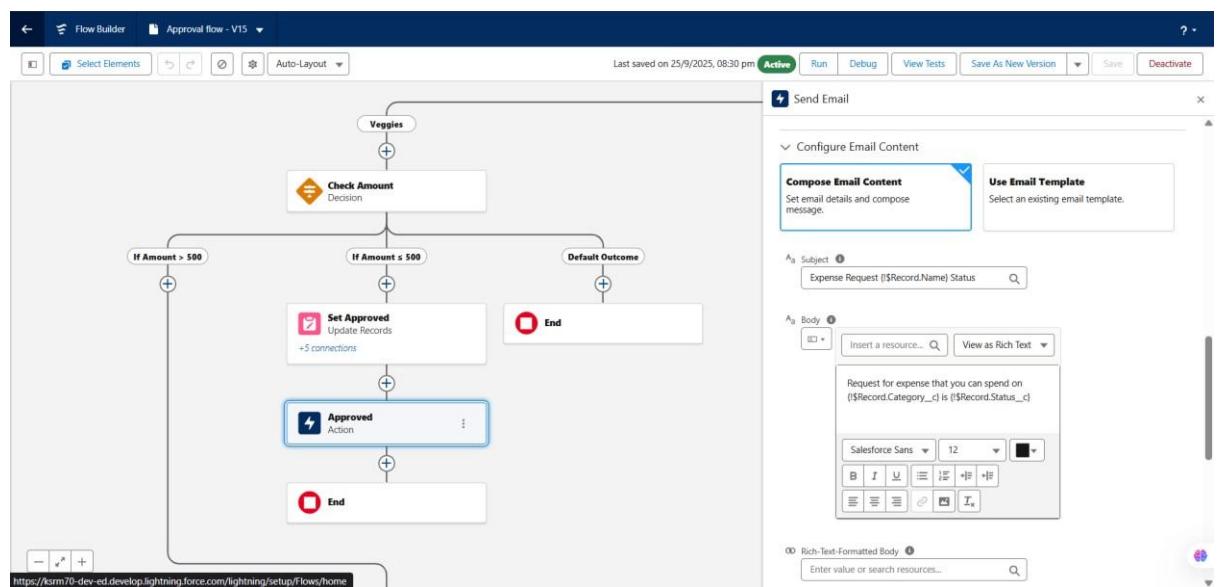
The body of the email contains dynamic content using merge fields to include specific details from the record. For approved requests, the message might read:

<p>Hello, your expense request of amount {!Amount_c} has been approved under the {!Category} category.</p> <p>You may proceed with your purchase.</p>

For rejected requests, the message could be:

<p>Hello, your expense request of amount {!Amount_c} exceeds the allowed budget for the {!Category} category.</p> <p>Please review and adjust your request.</p>

Using merge fields like {!Amount_c} and {!Category} allows Salesforce to automatically insert record-specific data into each email, ensuring that users receive accurate and personalized information. Placing this email action after the record update guarantees that the most recent Status_c value is already stored in the system before the notification is sent. This ensures both data accuracy and effective communication, providing users with a smooth and transparent expense approval experience.



5. Testing

Step 1: Create Test Records

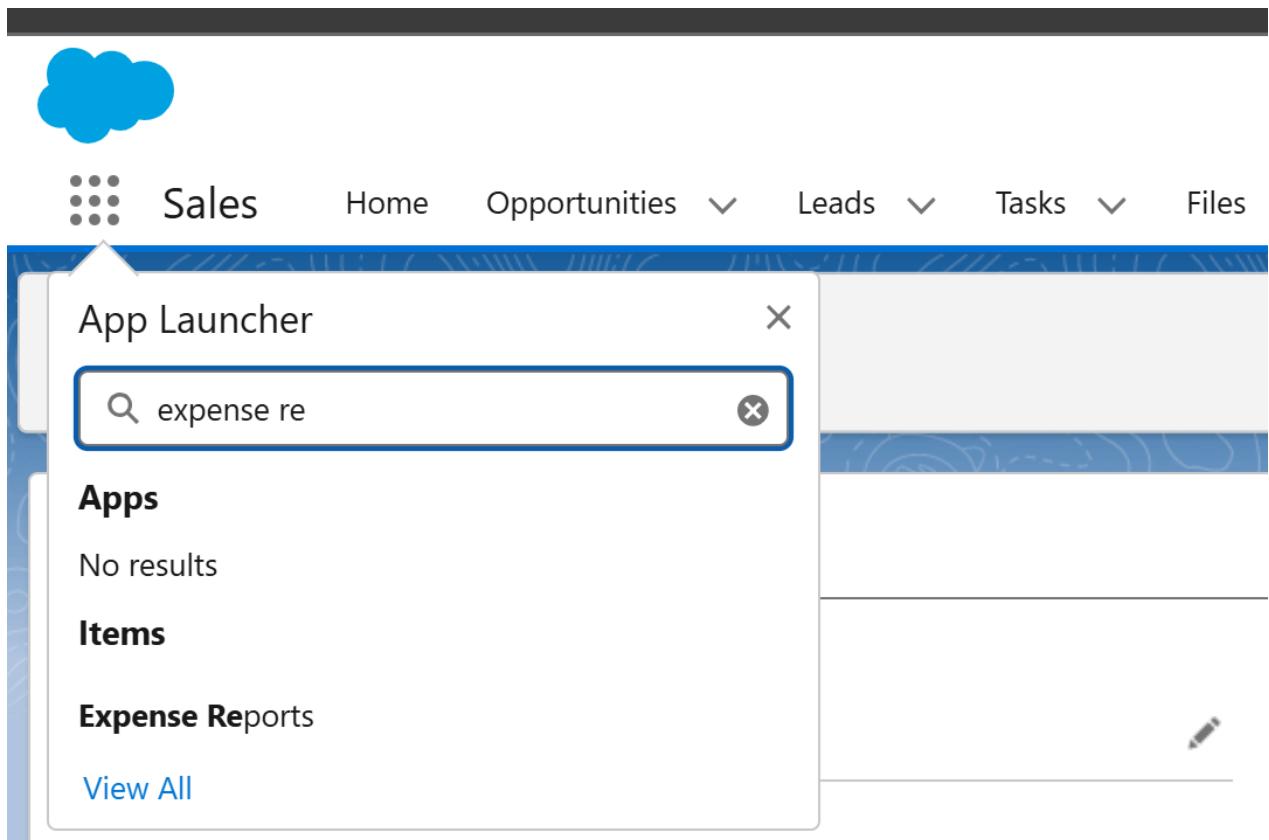
- Go to **Expense Reports** → **New**
- Enter different combinations of Category and Amount

Step 2: Verify Flow Execution

Test Case	Category	Amount	Expected Status	Email Output
1	Veggies	450	Approved	“Amount 450 approved under Veggies.”
2	Veggies	650	Not Approved	“Amount 650 exceeds allowed budget for Veggies.”
3	Travel	6000	Approved	“Amount 6000 approved under Travel.”
4	Travel	6100	Not Approved	“Amount 6100 exceeds allowed budget for Travel.”
5	Clothing	4000	Approved	“Amount 4000 approved under Clothing.”
6	Clothing	4500	Not Approved	“Amount 4500 exceeds allowed budget for Clothing.”

Edge Cases:

- Category blank → Default Outcome → Not Approved
- Amount = exact limit → Approved



The screenshot shows the Salesforce App Launcher open with the search term "expense re" entered. The results section is titled "Apps" and displays "No results". Below this, under "Items", there is a single result titled "Expense Reports" with a "View All" link.

App Launcher

expense re

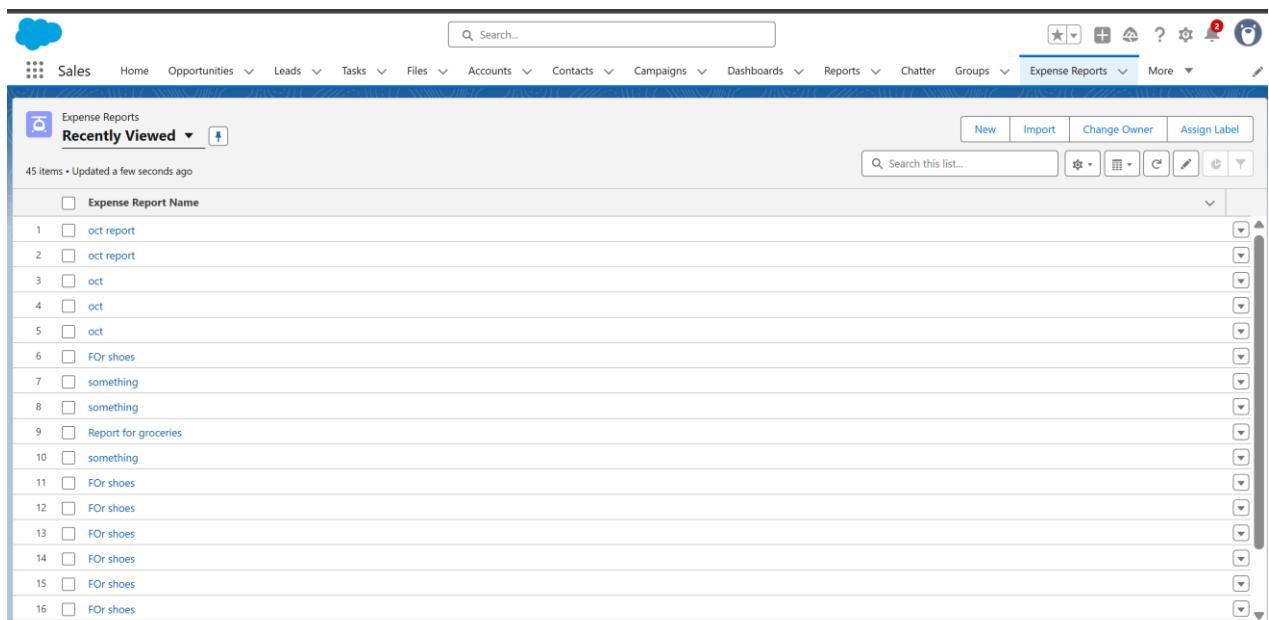
Apps

No results

Items

Expense Reports

[View All](#)



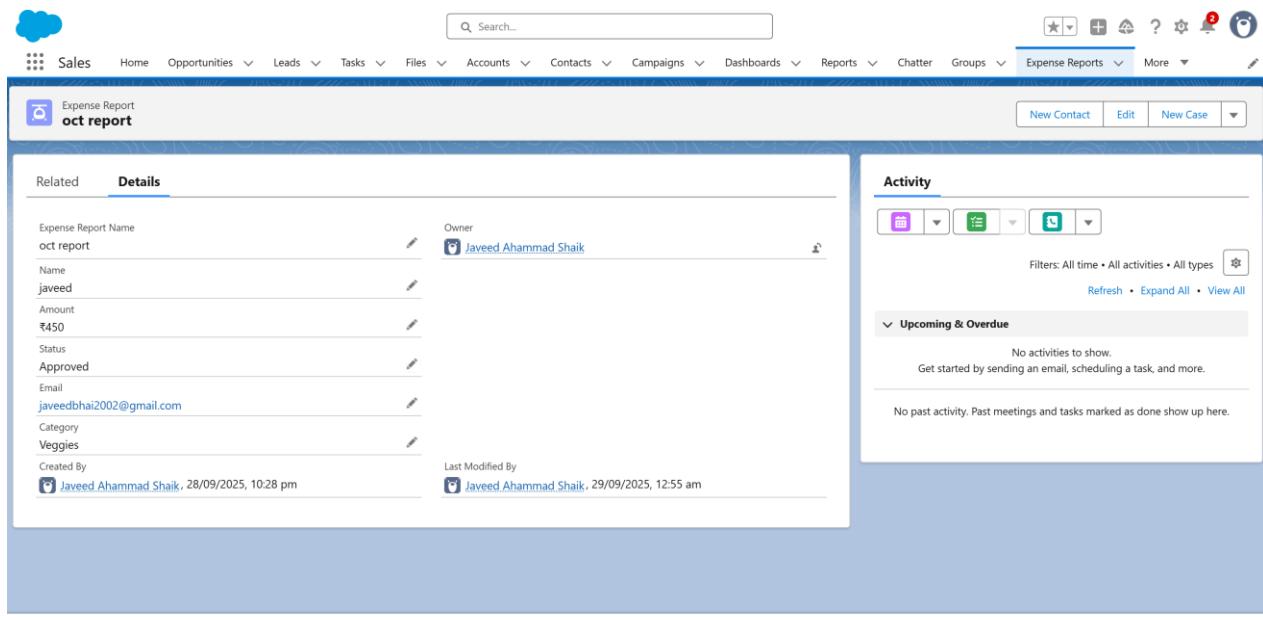
The screenshot shows the Expense Reports list view. The top navigation bar includes links for Sales, Home, Opportunities, Leads, Tasks, Files, Accounts, Contacts, Campaigns, Dashboards, Reports, Chatter, Groups, Expense Reports (which is the active tab), and More. The main content area displays a "Recently Viewed" section with 45 items, all updated a few seconds ago. The list shows various expense report names, many of which appear to be placeholder entries like "oct report" or "FOr shoes".

Expense Reports

Recently Viewed

45 items • Updated a few seconds ago

#	Expense Report Name
1	oct report
2	oct report
3	oct
4	oct
5	oct
6	FOr shoes
7	something
8	something
9	Report for groceries
10	something
11	FOr shoes
12	FOr shoes
13	FOr shoes
14	FOr shoes
15	FOr shoes
16	FOr shoes



6. Error Handling

The Expense Request Approval Flow also provides **auditability and traceability**. Since every action—whether approval, rejection, or email notification—is logged in Salesforce, managers can easily review past expense requests and their outcomes. This ensures compliance with organizational policies and simplifies reporting for audits or financial reviews.

The system promotes **consistency in decision-making** by applying the same rules to every record, removing subjective judgments or inconsistencies that can occur with manual approvals. It also allows for **role-based flexibility**; for example, future enhancements could add multi-level approvals for higher amounts or specific categories, integrating seamlessly into the existing flow structure.

By leveraging Salesforce's **native automation tools**, such as Flow Builder, the solution avoids the need for custom coding, making it easier to maintain and update. Non-technical administrators can adjust limits, categories, or notification content directly through the flow interface, reducing reliance on developers and enabling faster adaptability to organizational changes.

Furthermore, the automated flow enhances **employee productivity**. Employees no longer need to follow up manually on approvals, reducing email traffic and administrative delays. Managers can focus on exceptions or strategic decisions rather than routine approval tasks. In essence, the flow transforms expense management from a repetitive, error-prone process into a **highly efficient, transparent, and scalable workflow** that benefits both employees and management.

Future Enhancements

Key Enhancements

1. Manager Approval for High-Value Requests

We will automatically require a **Manager's approval** whenever a request goes over a certain dollar amount. This acts as a necessary **financial check** to prevent accidental overspending and ensures that large expenses are always reviewed and signed off by the appropriate level of management. This step creates a crucial **audit trail** for high-cost items and helps us stick to our budget and spending policies.

2. Conditional Routing for Different Departments

The system will use **smart routing** to send a request to the correct department instantly based on what the request is for. For instance, an expense report will go straight to Finance, while a software request will go directly to IT. This prevents misfiling and delays, meaning requests get to the right specialist **faster** and are handled more **efficiently** without manual forwarding.

3. Include Attachments (e.g., Receipts)

Users will be required to **upload supporting documents**, like receipts or invoices, directly into the request form. This makes the review process much easier and quicker for approvers because they have all the necessary **proof** right away. Requiring attachments eliminates back-and-forth emails and ensures we have complete, secure **records** for all financial transactions.

4. Create Reporting Dashboards

We will build **reporting dashboards** that show real-time performance data for the entire workflow. These dashboards will track key metrics like how long approvals are taking and which departments are submitting the most requests. This gives managers a clear **visual overview** to quickly spot bottlenecks, measure efficiency, and make data-driven decisions to **improve the process**.

5. Make Flow Mobile-Friendly for Submissions

The entire request and submission process will be optimized to work perfectly on **smartphones and tablets**. This is essential for employees who are traveling or working outside the office. A **mobile-friendly interface** makes it easy for everyone to submit requests, take a photo of a

receipt, and check the status of their submission from anywhere, significantly **boosting convenience and adoption**.

These features work together to make the entire process faster, more accurate, and easier to use while maintaining strong financial control.

7. Conclusion

- This description outlines an excellent, robust, and highly automated expense management workflow. Here is an elaboration on these key highlights, suitable for a project document, focusing on the functionality, benefits, and technical strength of the design.
- Detailed Elaboration on the Category and Amount-Based Expense Flow
- This automated expense flow is designed to provide rapid, accurate, and scalable processing of employee expenses, ensuring compliance and immediate communication through sophisticated workflow logic and integrated email functionality.
- Simple Object Structure for Core Data Management
- The workflow utilizes a simple, focused data object comprising four essential fields: Name, Amount, Category, and Status. This design choice prioritizes efficiency and clarity. By limiting the object to only necessary attributes, the system minimizes data entry effort, speeds up processing, and simplifies maintenance.
 - Name identifies the submitter or the expense title.
 - Amount provides the critical financial data for approval logic.
 - Category (e.g., Travel, Meals, Software) drives the conditional routing.
 - Status (e.g., Pending, Approved, Rejected) provides real-time visibility into the expense lifecycle.
- This clean, simple structure ensures scalability and allows for quick reporting and auditing without being bogged down by unnecessary data complexity.
- Flexible Flow Branching for Multiple Categories
- The core strength of this automation lies in its flexible flow branching, which allows the system to handle diverse expense types through a single process. The flow uses the Category field to intelligently route the request down different paths. For example, a "Travel" expense may require approval from a Project Manager, while a "Software Subscription" expense might route directly to the IT department for license review. This conditional logic ensures:
 - Compliance: Every expense type is reviewed by the appropriate subject matter expert.

- Efficiency: It bypasses unnecessary approval steps, directing the request straight to the required decision-maker.
- This architecture is easy to maintain and modify. New categories and their corresponding routing rules can be added quickly without having to redesign the entire workflow, making it inherently ready for production in a changing business environment.
- Dynamically Generated Email Notifications
- The flow includes a powerful Send Email action that is integrated directly with the approval/rejection outcome. The content of the email notifications is dynamically generated, meaning the system automatically pulls relevant data from the expense object (like the submitter's name, the expense amount, and the rejection reason) and inserts it into the email template. This ensures that:
 - Communication is immediate: Notifications are sent the moment an expense status changes.
 - Communication is contextual: The recipient always receives accurate and personalized information about their submission or decision.
 - This automation eliminates manual communication, drastically reducing the time spent on administrative tasks and providing a high-quality, professional user experience.
- Robust Error Handling for Reliable Execution
- A key feature for ensuring production readiness is the inclusion of comprehensive error handling. This means the workflow is designed to anticipate and gracefully manage potential failures (e.g., a connection error with the email server, or an unexpected data format). Instead of simply crashing or stopping, the system will execute specific error management steps, such as:
 - Logging the failure details for developer review.
 - Sending a specific alert to the system administrator.
 - Attempting to re-execute a step where appropriate.
- This robust design ensures high execution reliability, minimizing downtime and preventing the loss of critical expense data, which is essential for a system handling financial transactions.
- Scalable and Easy to Maintain Architecture
- The overall design is focused on scalability and long-term maintainability. By adhering to the simple object structure, separating logic through conditional branching, and implementing clear error paths, the flow is structurally sound. The modular nature of the design means:

- Scalability: The system can easily handle a massive increase in the volume of expense requests without significant performance degradation.
- Maintenance: Updates, regulatory changes, or new business rules (like adjusting approval limits) can be implemented in specific branches of the flow without affecting the core logic or other categories, resulting in low technical debt and a system that is ready for production from day one.