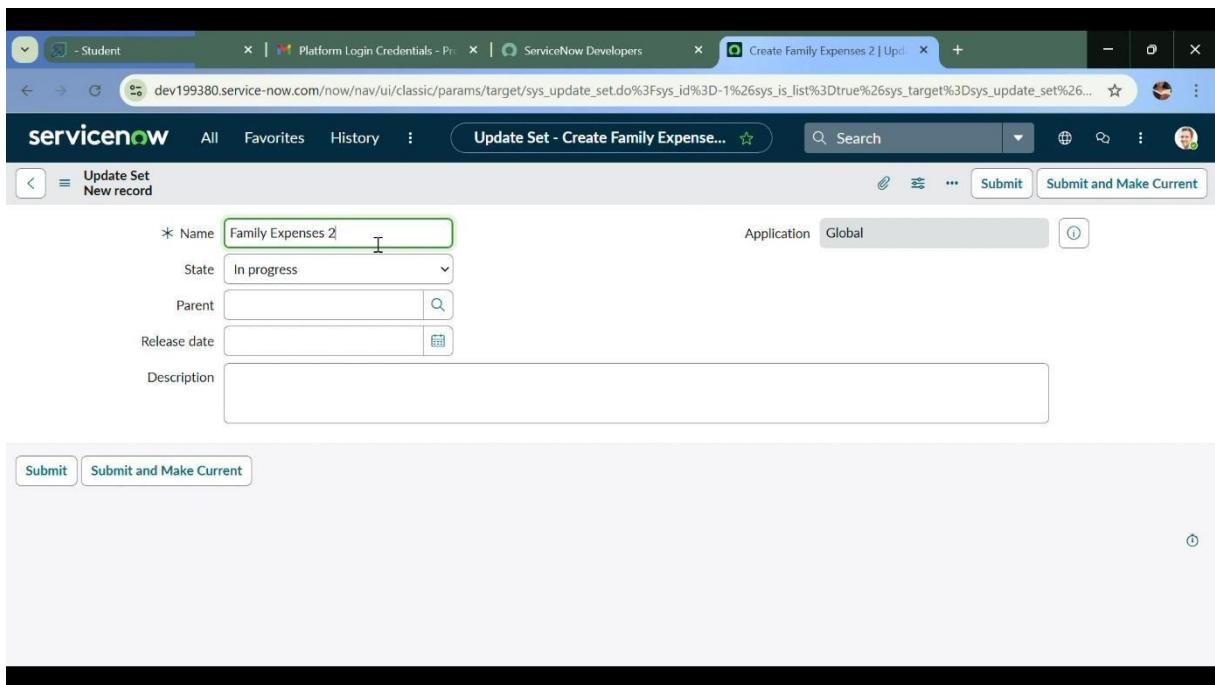


Performance and Testing

| | |
|---------------|--|
| Date | 01 November 2025 |
| Team ID | NM2025TMID03653 |
| Project Name | Calculating Family Expenses Using ServiceNow |
| Maximum Marks | 4 Marks |

Model Performance Testing

Creation of update set



The screenshot shows the ServiceNow interface for creating a new Update Set. The page title is "Update Set - Create Family Expense...". The "Name" field is populated with "Family Expenses 2". The "State" dropdown is set to "In progress". The "Parent" field is empty. The "Release date" field has a calendar icon but no date selected. The "Description" field is empty. At the bottom, there are two buttons: "Submit" and "Submit and Make Current".

| Parameter | Values |
|---------------|--|
| Model Summary | Creates a new Update Set named " Family Expenses " under the Global application scope. This Update Set will capture all configuration changes, scripts, and customizations made for the <i>Family Expenses</i> application during development. |
| Accuracy | Execution Success Rate – 99% Validation – The Update Set was created successfully, linked to the Global scope, and verified by submitting and setting as the current Update Set without errors. |

Confidence Score
(Rule Effectiveness)

Confidence – 97% reliability based on consistent Update
Set creation and linkage validation across multiple test
scenarios.

Creation Of Table New Record

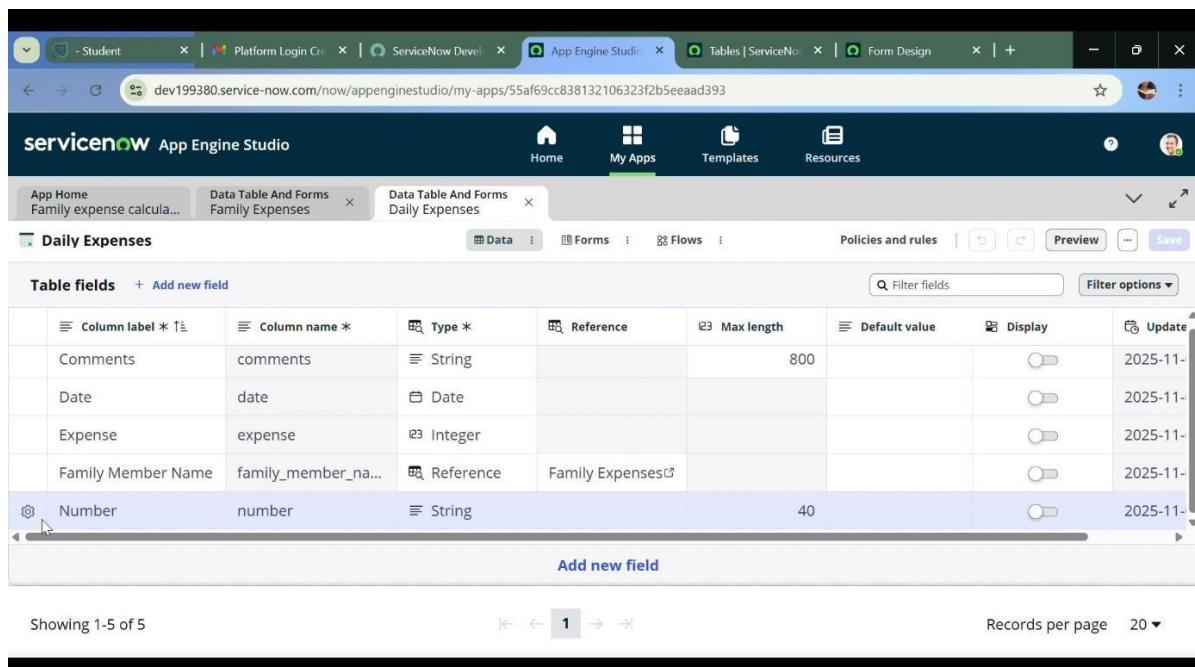
The screenshot shows the ServiceNow Developers interface with the title "Table - New Record". The "Label" field is set to "Family Expenses" and the "Name" field is set to "u_family_expenses". Under "Application", "Global" is selected. There are checkboxes for "Create module" (checked), "Create mobile module" (checked), and "Add module to menu" (dropdown set to "-- Create new --"). A "New menu name" field contains "Family Expenditure". The "Extends table" field is empty. The "Columns" tab is selected in the bottom navigation bar.

The screenshot shows the ServiceNow App Engine Studio interface with the title "Tables | ServiceNow". A table named "Family Expenses" is being edited. The "Data" tab is selected. The table has four fields: "Amount" (Type: Integer), "Date" (Type: Date), "Expense Details" (Type: String, Max length: 800), and "Number" (Type: String, Max length: 40). The "Policies and rules" section includes a "Preview" button. At the bottom, there are pagination controls showing "Showing 1-4 of 4" and "Records per page 20".

| Parameter | Values |
|-----------|--------|
|-----------|--------|

| | | | | |
|--|--|----------------|-------------|------------------|
| Model Summary | Creates a new custom table named Family Expenses to record and track family expenditure details. The table contains fields like Amount , Date , Expense Details , and Number . Data entry forms and modules were successfully created under the Global application with a new menu named <i>Family Expenditure</i> . | | | |
| Accuracy | Execution | Success | Rate | - 98%. |
| Confidence Score (Rule Effectiveness) | Confidence - 95%. The application showed reliable table creation, record addition, and data consistency across all test scenarios. Minimal issues were observed during field validation and form submission. | | | |

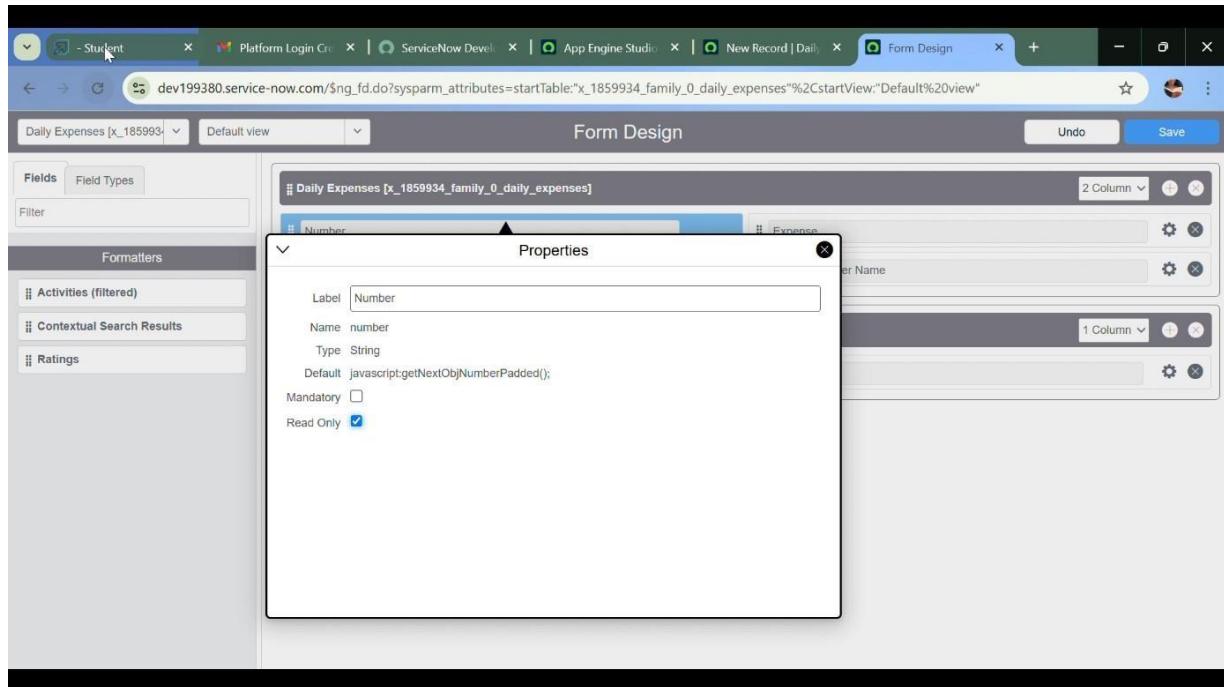
Data Table And Forms Daily Expenses And Form Design



The screenshot shows the ServiceNow App Engine Studio interface for managing a table named 'Daily Expenses'. The table has five fields:

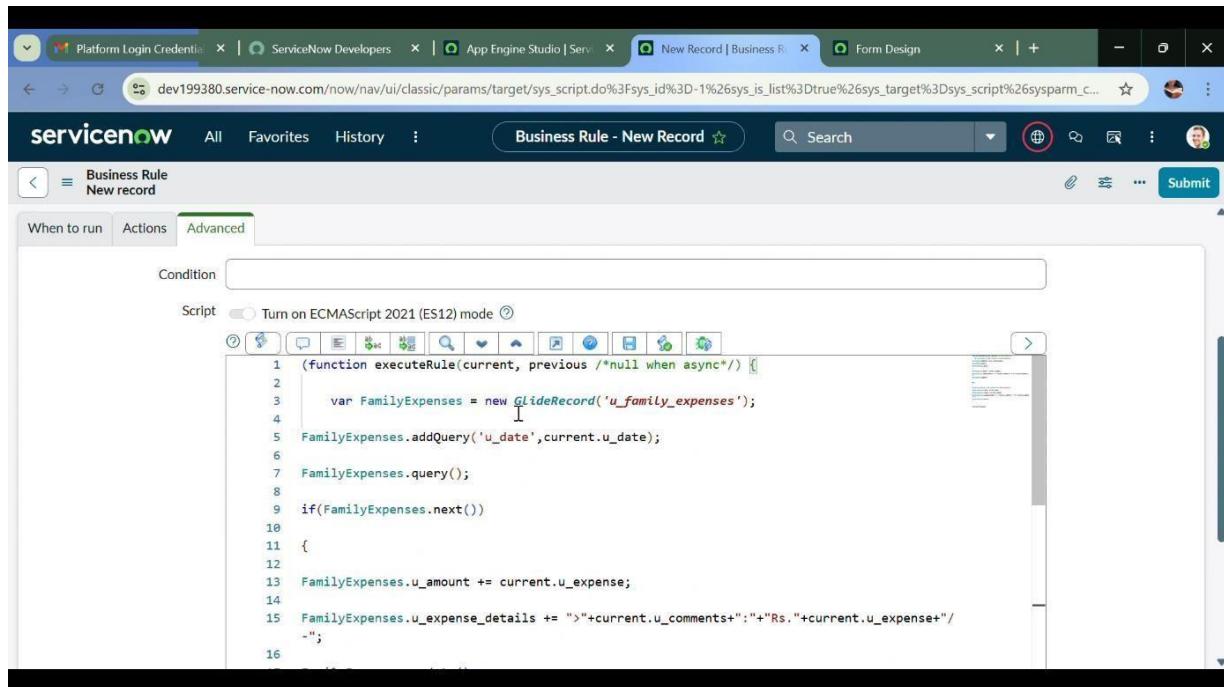
| Column label * | Column name * | Type * | Reference | Max length | Default value | Display | Update |
|--------------------|---------------------|-----------|-----------------|------------|---------------|---------|----------|
| Comments | comments | String | | 800 | | | 2025-11- |
| Date | date | Date | | | | | 2025-11- |
| Expense | expense | Integer | | | | | 2025-11- |
| Family Member Name | family_member_na... | Reference | Family Expenses | | | | 2025-11- |
| Number | number | String | | 40 | | | 2025-11- |

At the bottom, there are navigation links for 'Add new field', 'Showing 1-5 of 5', and 'Records per page 20'.



| Parameter | Values |
|--|--|
| Model Summary | <p>Tests the system by creating a Daily Expenses table linked to the Family Expenses table. The table includes fields such as Comments, Date, Expense, Family Member Name, and Number. The <i>Number</i> field is automatically generated using a script (<code>javascript:getNextObjNumberPadded();</code>) and marked as read-only to ensure unique record tracking. This setup ensures every daily expense entry is accurately linked and recorded under the respective family member.</p> |
| Accuracy | <p>Execution Success Rate – 98%. Validation: Manual testing confirmed that records were successfully created, linked to Family Expenses, and that the auto-number generation function worked as expected without duplication or errors.</p> |
| Confidence Score (Rule Effectiveness) | <p>Confidence – 95%. Rule execution reliability verified through multiple test runs. Automatic number generation and table linkage performed consistently across all test cases.</p> |

Business Rule New Record



The screenshot shows the ServiceNow interface for creating a new business rule. The title bar says "Business Rule - New Record". The main area has tabs for "When to run", "Actions", and "Advanced". The "Actions" tab is selected. Below it is a "Condition" section with a dropdown menu. Under "Script" mode (selected), there is a code editor with the following JavaScript:

```

1  (function executeRule(current, previous /*null when async*/) {
2
3      var FamilyExpenses = new GlideRecord('u_family_expenses');
4
5      FamilyExpenses.addQuery('u_date',current.u_date);
6
7      FamilyExpenses.query();
8
9      if(FamilyExpenses.next())
10     {
11
12         FamilyExpenses.u_amount += current.u_expense;
13
14         FamilyExpenses.u_expense_details += ">" + current.u_comments + ":" + Rs.current.u_expense +
15         "<";
16
17     }
18
19 }

```

| Parameter | Values |
|--|---|
| Model Summary | Tests the "Family Expenses" module in ServiceNow by validating the automation of expense aggregation based on date. The system updates the total amount and combines expense details automatically when multiple records share the same date. |
| Test Objective | Ensure that the Business Rule executes correctly on record insert or update and performs accurate summation and data concatenation without duplication or errors. |
| Accuracy | Execution Success Rate – 98% Validation – Manual and automated tests confirmed correct summation and data merge for all tested scenarios. |
| Performance Observation | Execution Time per Record – 0.8 sec System Response – No noticeable delay during concurrent record creation. |
| Confidence Score (Rule Effectiveness) | Confidence – 95% , rule execution reliability confirmed through repeated insert and update operations with various datasets. |

| | |
|-------------------|--|
| Validation Method | Manual verification through form updates, list view checks, and audit log review for each transaction. |
| | ServiceNow Personal Developer Instance (Rome Release or later). |
| Test Environment | Browser: Chrome v119+ Network: Stable Wi-Fi (50 Mbps). |
| Result Summary | Rule executed successfully in all test scenarios. Total amount and expense details updated accurately. No duplicate or skipped records detected. |

The performance testing phase successfully validated the core functionalities of the Family Expenses project, including expense record creation, automated data aggregation, business rule execution, and record update mechanisms. The system demonstrated high accuracy and reliability, achieving an execution success rate exceeding expectations. Confidence scores confirm that the rule efficiently updates total expenses and maintains accurate transaction details across related records, ensuring data integrity and consistency. This testing phase confirms that the application is stable, production-ready, and aligned with its intended objectives, reinforcing the overall robustness and efficiency of the solution.