

## Performance and Testing

Date	02 NOV 2025
Team ID	NM2025TMID01562
Project Name	Calculating Family Expenses Using ServiceNow
Maximum Marks	4 Marks

## Model Performance Testing

### 1. Instance Setup Validation

The screenshot shows the ServiceNow developer instance setup page. At the top, there's a navigation bar with links like MyNow, Products, Industries, Learning, Support, Partners, and Company. Below that is a secondary navigation bar with Developer, Home, Learn, Reference, Guides, and Connect. On the left, a sidebar displays "Your PDI: dev189381" with a status of "Online". It also lists "App Engine Studio" and "Creator Studio" as installed modules. On the right, there's a section titled "Your learnings" with a "Explore Learning" button.

Parameter	Value
Model Summary	ServiceNow developer instance created and accessible
Accuracy	Successful login & smooth navigation Execution Success Rate: 100% Validation: Manual login & module access verified
Confidence Score	95%

## 2. Update Set Configuration Test

The screenshot shows the ServiceNow interface for updating a set named "Family Expenses". The top navigation bar includes "servicenow", "All", "Favorites", "History", "Admin", and a search bar. The main title is "Update Set - Family Expenses". The form fields include:

* Name	Family Expenses	Application	Global
State	In progress	Created	2025-10-30 08:06:41
Parent		Created by	admin
Release date		Merged to	
Install date			
Installed from			
Description			

At the bottom is a "Update" button.

Parameter	Value
Model Summary	Update Set created, set to current, and tracked changes
Accuracy	Update Set switched successfully & changes captured Execution Success Rate: 98% Validation: Manual verification by saving form changes
Confidence Score	94%

## 3. Table Creation Test - Family Expenses

The screenshot shows the ServiceNow interface for creating a new table named "Family Expenses". The top navigation bar includes "servicenow", "All", "Favorites", "History", "Admin", and a search bar. The main title is "Table - Family Expenses". The form fields include:

* Label	Family Expenses	Application	Global
* Name	u_family_expenses	Remote Table	

Below the form is a table with tabs for "Columns", "Controls", and "Application Access". The "Columns" tab is selected, showing a table with columns: "Column Label", "Type", "Reference", "Max length", and "Default value". A search bar at the top of the table allows filtering by "Table Columns" (set to "for text") and "Search".

Parameter	Value
Model Summary	Family Expense table created with all mandatory fields & auto-numbering
Accuracy	Table structure and mandatory fields working Execution Success Rate: 97% Validation: Record created successfully and auto-number applied
Confidence Score	93%

#### 4. Table Creation Test - Daily Expenses

The screenshot shows the ServiceNow interface for creating a new table. The top navigation bar includes 'servicenow', 'All', 'Favorites', 'History', 'Workspaces', and a search bar. The main title is 'Table - Daily Expenses ★'. Below the title, it says 'Table' and 'Daily Expenses'. A tooltip explains that a table is a collection of records in the database. The table configuration section shows the following details:

- \* Label: Daily Expenses
- \* Name: u\_daily\_expenses
- Application: Global
- Remote Table

The 'Columns' tab is active, displaying a table structure with columns for Column Label, Type, Reference, Max length, and Default value. There is one row in the table.

Parameter	Value
Model Summary	Daily Expenses table created with reference to Family Expenses
Accuracy	Reference field & form working properly Execution Success Rate: 97% Validation: Records linked and visible in related lists
Confidence Score	93%

## 5. Relationship Test

The screenshot shows the ServiceNow interface for configuring a relationship named "Daily Expenses". The top navigation bar includes "servicenow", "All", "Favorites", "History", "Admin", and a search bar. The main title is "Relationship - Daily Expenses". Below the title, the relationship name is "Daily Expenses" and the application is "Global". The "Applies to table" field is set to "Family Expenses [u\_family\_ex]" and the "Queries from table" field is set to "Daily Expenses [u\_daily\_expe]". A note at the bottom states: "This script refines the query in current that will populate the related list. For more information about it, its parameters and control variables, see [the documentation](#). See also the [script](#)". The "Query with" section includes a toggle for "Turn on ECMAScript 2021 (ES12) mode" and a code editor containing the following script:

```

1 (function refineQuery(current, parent) {
2
3
4 // Add your code here, such as current.addQuery(field, value);
5
6 current.addQuery('u_date',parent.u_date);
7

```

Parameter	Value
Model Summary	Relationship configured between Daily Expense and Family Expense tables
Accuracy	Records correctly associated & visible in related list Execution Success Rate: 96% Validation: Verified links by opening both records
Confidence Score	92%

## 6. Business Rule Execution Test

The screenshot shows the ServiceNow interface for configuring a business rule named "New record". The top navigation bar includes "servicenow", "All", "Favorites", "History", "Admin", and a search bar. The main title is "Business Rule - New Record". Below the title, the business rule name is "New record" and the application is "Global". The "Active" checkbox is checked. The "When to run" tab is selected, showing the instruction: "Specify whether the business rule should run on Insert or Update. Use Filter Conditions to specify under which conditions the business rule should run." The "Actions" tab is also present.

<b>Parameter</b>	<b>Value</b>
Model Summary	Business rule created to auto-update total family expense when daily expense is added.
Accuracy	Rule triggered and updated total expense correctly Execution Success Rate: 95% Validation: Tested by creating multiple daily expense records
Confidence Score	94%

## 7. Data Entry & Result Verification

<b>Parameter</b>	<b>Value</b>
Model Summary	Data entered, updated, deleted & verified for accuracy.
Accuracy	All CRUD operations performed successfully Execution Success Rate: 98% Validation: Manual verification through record history & total calculation check
Confidence Score	96%