

CALCULATING FAMILY EXPENSES USING SERVICENOW

TEAM ID : NM2025TMID14334

TEAM MEMBERS: 4

TEAM LEADER : DHEENU OMAL HAREESH KUMAR

TEAM MEMBER 1: A ASARAF

TEAM MEMBER 2: R MOHANA PRIYA

TEAM MEMBER 3: M AKASH

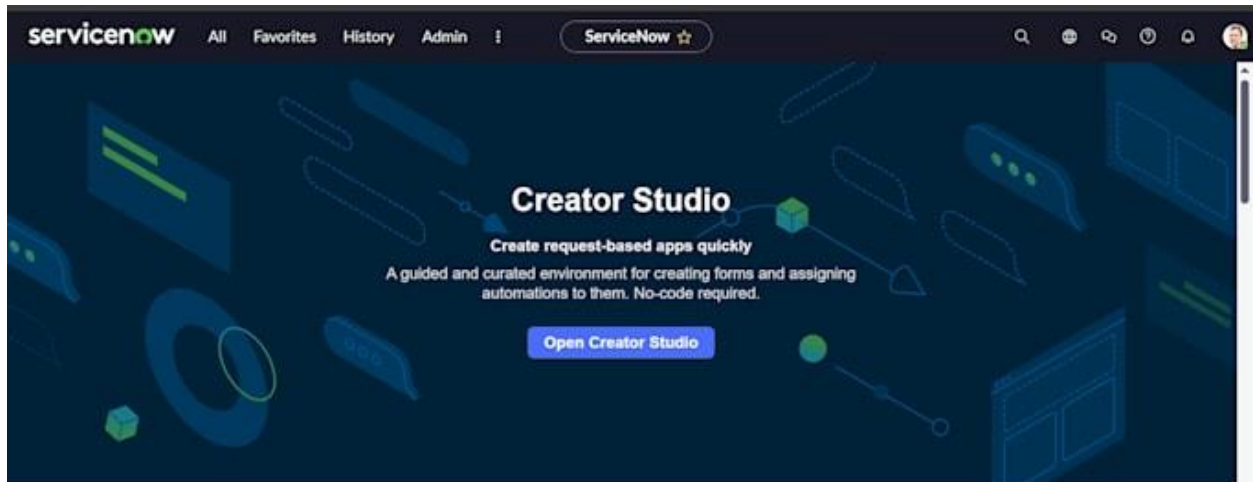
PROBLEM STATEMENT: Individuals and families lack an efficient, centralized system to track, categorize, and analyze their personal expenses, leading to difficulties in budgeting, managing spending, and making informed financial decisions.

OBJECTIVE: To leverage the ServiceNow platform to develop an automated and intuitive family expense management system that provides real-time financial tracking, personalized budget alerts, and insightful reporting to empower users with better control over their finances.

SKILLS: ServiceNow Platform Development , Data Management , Financial Management , User Interface (UI) & User Experience (UX).

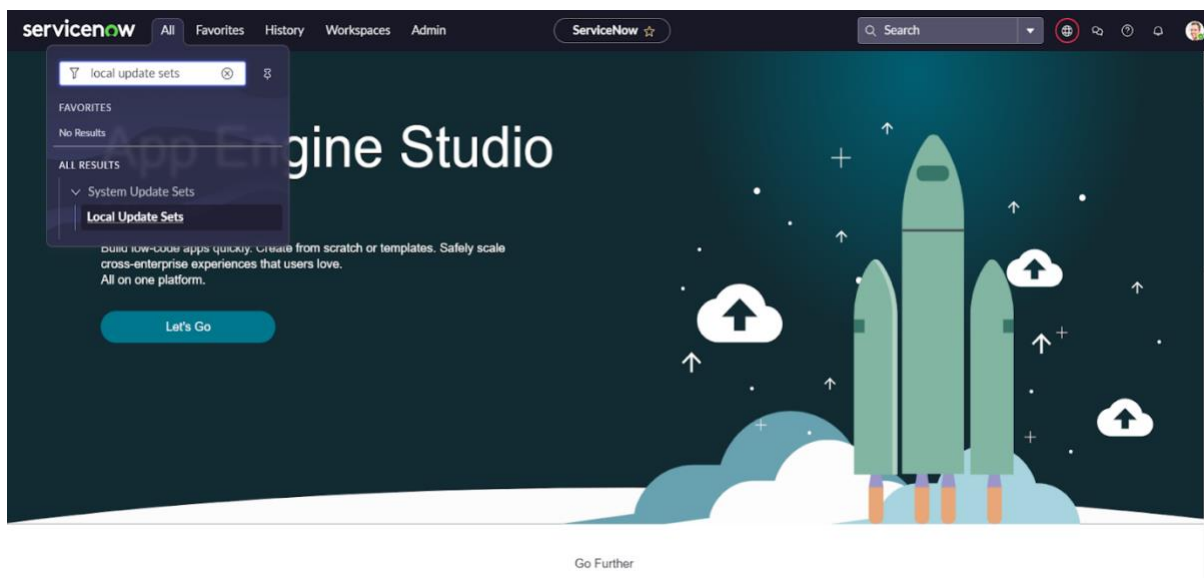
Milestone 1: Setting up ServiceNow Instance

1. Open service now Developer site
“<https://developer.servicenow.com>”.
2. Sign up for your developer account
3. Once logged in, click on “start building”.
4. Then click on "Request Instance" to create a new ServiceNow instance.
5. It take few minutes to build your instance
6. Once instance created, Log in to your ServiceNow instance using the provided credentials.
7. Now you will navigate to the ServiceNow.



Milestone 2: Creation of New Update Set

1. Go to All >> In the filter search for Local Update set > click on New.



2. Enter the Details as:

Name : Family Expenses

3. Then click on Submit and Make current.

The screenshot shows the ServiceNow interface for creating a new update set. The left sidebar contains a navigation menu with items like 'Configuration', 'System Localization', and 'System Update Sets'. The 'Local Update Sets' item is highlighted with a red box and a red arrow. The main form area is titled 'Update Set - Create New Update Set' and contains fields for 'Name', 'State', 'Parent', 'Release date', and 'Description'. The 'Name' field is set to 'Family Expenses' and is highlighted with a red box and a red arrow. The 'State' is set to 'In progress'. The 'Application' is set to 'Global'. The 'Submit and Make Current' button is highlighted with a red box and a red arrow.

Milestone 3: Creation of Table

Activity 1: Creation of Family Expenses Table

1. Go to All > In the filter search for Tables > click on New.

2. Enter the Details:

Label : Family Expenses

Name : Auto-Populated

New menu name : Family Expenditure

* Label
 * Name

Application ⓘ
 Remote Table ☒
 Create module ☒
 Create mobile module ☒
 Add module to menu
 New menu name

Columns Controls Application Access

Table Columns ⓘ

Dictionary Entries

	Column label ▼	Type	Reference	Max length	Default value	Display
✕ ✎	Number	String				false
✕ ✎	Date	Date				false
✕ ✎	Amount	Integer				false

3. Go to the Header and right click there>> click on Save.

Activity 2: Creation of Columns(Fields)

1. Near Columns Double click near insert a new row.

2. Give the details as:

Column label : Number

Type : String

3. Double click on insert a new row again

4. Give the details as:

Column label : Date

Type : Date

5. Double click on insert a new row again

6. Give the details as:

Column label : Amount

Type : Integer

7. Double click on insert a new row again

8. Give the details as:

Column label : Expense Details

Type : String

Max length : 800



9. Go to the Header and right click there>> click on Save.

Activity 3: Making Number Field an Auto-Number

1. Double click on the Number Field/Column.

2. Go down and double click on Advanced view

3. In Default Value:

Use dynamic default : check the box

Dynamic default value : Get Next Padded Number

4. Click on Update.

Choice List Specification | Calculated Value | **Default Value** 

The **Default value** specifies what value the field has when first displayed.

Use dynamic default ☒ 

Dynamic default value | Get Next Padded Number 

Delete Column | **Update** 

5. Go to All >> In the filter search for Number Maintenance
>> select Number Maintenance

6. Click on New.

7. Enter the below Details:

Table : Family Expenses

Prefix : MFE

Number MFE

* Table Family Expenses

Prefix MFE

* Number 1,000

Application Global

Number of digits 7

Update Delete

8. Click on Submit.

Activity 4: Configure the Form

1. Go to All >> In the filter search for Family Expenses >> Open Family Expenses
2. Click on New
3. Go to the Header and right click there>> click on Configure >> Select Form Design
4. Customize or Drag Drop the form as per your requirement.

The image shows a configuration window for a table named 'Family Expenses [u_family_expenses]'. It is divided into two sections. The top section is set to '2 Column' and contains three fields: 'Number', 'Date', and 'Amount'. Each field has a gear icon to its right, indicating it can be configured. The bottom section is set to '1 Column' and contains one field: 'Expense Details', also with a gear icon.

5. Make Number Read-Only Field by clicking on the gear icon and checking Read-Only
6. Make Date, Amount Mandatory Field by clicking on the gear icon and checking Mandatory
7. Click on Save.

Milestone 4: Creation of Table(Daily Expenses)

Activity 1: Creation of Daily Expenses Table

1. Go to All > In the filter search for Tables > click on New.
2. Enter the Details:

Label : Daily Expenses

Name : Auto-Populated

Add Module to menu : Family Expenditure

✖ Label Daily Expenses ↩ 1

✖ Name u_daily_expenses ↩ 2

Extends table

Application Global ⓘ

Create module ☒

Create mobile module ☒

Application Menu

Add module to menu Family Expenditure ↩ 3

3. Go to the Header and right click there>> click on Save.

Activity 2: Creation of Columns(Fields)

1. Near Columns Double click near insert a new row.

2. Give the details as:

Column label : Number

Type : String

3. Double click on insert a new row again

4. Give the details as:

Column label : Date

Type : Date

5. Double click on insert a new row again

6. Give the details as:

Column label : Expense

Type : Integer

7. Double click on insert a new row again

8. Give the details as:

Column label : Family Member Name

Type : Reference

Max length : 800

9. Double click on insert a new row again

10. Give the details as:

Column label : Comments

Type : String

Max length : 800

11. Go to the Header and right click there>> click on save

Activity 3: Making Number Field an Auto-Number

1. Double click on the Number Field/Column.

2. Go down and double click on Advanced view

3. In Default Value:

Use dynamic default : check the box

Dynamic default value : Get Next Padded Number

4. Click on Update.

This screenshot shows the 'Default Value' configuration interface. At the top, there are three tabs: 'Choice List Specification', 'Calculated Value', and 'Default Value', with the third tab selected and marked with a red box and arrow 1. Below the tabs, a blue banner states: 'The Default value specifies what value the field has when first displayed.' Underneath, there is a checkbox labeled 'Use dynamic default' which is checked, marked with a red box and arrow 2. Below this is a text input field for 'Dynamic default value' containing 'Get Next Padded Number', marked with a red box and arrow 3. At the bottom left, there are two buttons: 'Delete Column' and 'Update', with the 'Update' button marked by a red box and arrow 4.

5. Go to All >> In the filter search for Number Maintenance
>> select Number Maintenance

6. Click on New.

7. Enter the below Details:

Table : Family Expenses

Prefix : MFE

This screenshot shows the 'Number New record' form. At the top left, there is a back arrow and a menu icon. The title 'Number New record' is displayed. On the top right, there are icons for edit, help, settings, and a 'Submit' button. The form contains several fields: 'Table' with the value 'Daily Expenses' (marked with a red box and arrow 1), 'Prefix' with the value 'DFE' (marked with a red box and arrow 2), 'Number' with a value of '1,000', 'Application' with the value 'Global', and 'Number of digits' with a value of '7'. At the bottom left, there is a 'Submit' button marked with a red box and arrow 3.

8. Click on Submit.

Activity 4: Configure the Form

1. Go to All >> In the filter search for Daily Expenses >> Open Daily Expenses
2. Click on New
3. Go to the Header and right click there>> click on Configure >> Select Form Design
4. Customize or Drag Drop the form as per your requirement.

The screenshot shows a form configuration window for 'Daily Expenses [u_daily_expenses]'. The form is configured in a '2 Column' layout. It contains four fields: 'Number', 'Date', 'Family Member Name', and 'Expense'. Each field has a gear icon for configuration. Below this, there is a section for 'Comments' with a '1 Column' layout.

5. Make Number Read-Only Field by clicking on the gear icon and checking Read-Only
6. Make Date, Family Member Name Mandatory Field by clicking on the gear icon and checking Mandatory
7. Click on Save.

Milestone 5: Creation of Relationship

Activity 1: Creation of Relationship between Family Expenses and Daily Expenses tables

1. Go to All >> In the filter search for Relationships >>
Open Relationships
2. Click on New.
3. Enter the details:

Name : Daily Expenses

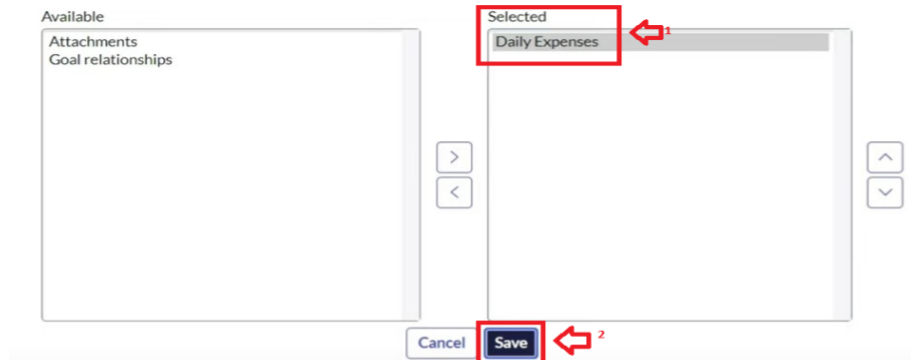
Applies to table : Select Family Expenses

Daily Expenses : Select Daily Expenses
4. Click Save.

Milestone 6: Configuring Related List on Family Expenses

1. Go to All >> In the filter search for Family Expenses >>
Open Family Expenses
2. Click on New
3. Go to the Header and right click there>> click on
Configure >> Select Related Lists
4. Add Daily Expenses to the Selected Area.

5. Click on Save



Milestone 7: Creation of Business Rules

1. Go to All >> In the filter search for Business Rules.
2. Under System Definition Select Business Rules then click on New.
3. Enter the Details:

Name : Family Expenses BR

Table : Select Daily Expenses

Check Advanced

Business Rule
New record

ss rule is a server-side script that runs when a record is displayed, inserted, deleted, or when a table is queried. Use business rules to automatically change values in form fields when the specified conditions are met





Name 


Table 

Application 

Active ☒

Advanced ☒ 

4. In when to run Check Insert and Update


When to run  Advanced

Specify whether the business rule should run on **Insert** or **Update**. Use **Filter Conditions** to specify under which conditions

When

Order

Insert ☒


Update ☒ 

Delete ☐

Query ☐

Filter Conditions

-- choose field --

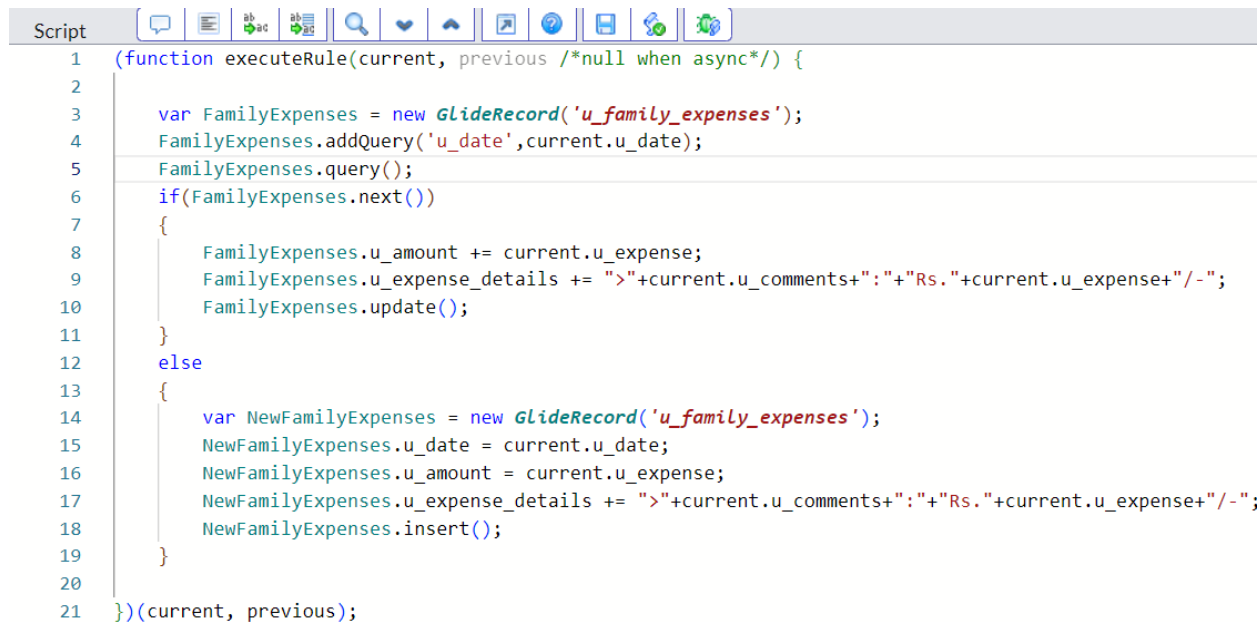
Role conditions 

5. In Advance(we write the code): Write the below code >>

```
(function executeRule(current, previous /*null when
async*/) {
```



```
var FamilyExpenses = new
GlideRecord('u_family_expenses');
FamilyExpenses.addQuery('u_date',current.u_date);
FamilyExpenses.query();
if(FamilyExpenses.next())
{
FamilyExpenses.u_amount += current.u_expense;
FamilyExpenses.u_expense_details +=
">" + current.u_comments + ":" + "Rs." + current.u_expense
+ "/-";
FamilyExpenses.update();
}
Else
{
var NewFamilyExpenses = new
GlideRecord('u_family_expenses');
NewFamilyExpenses.u_date = current.u_date;
NewFamilyExpenses.u_amount = current.u_expense;
NewFamilyExpenses.u_expense_details +=
">" + current.u_comments + ":" + "Rs." + current.u_expense
+ "/-";
NewFamilyExpenses.insert();
}
})(current, previous);
```



```

1 (function executeRule(current, previous /*null when async*/) {
2
3     var FamilyExpenses = new GlideRecord('u_family_expenses');
4     FamilyExpenses.addQuery('u_date', current.u_date);
5     FamilyExpenses.query();
6     if(FamilyExpenses.next())
7     {
8         FamilyExpenses.u_amount += current.u_expense;
9         FamilyExpenses.u_expense_details += ">" + current.u_comments + ":" + "Rs." + current.u_expense + "/-";
10        FamilyExpenses.update();
11    }
12    else
13    {
14        var NewFamilyExpenses = new GlideRecord('u_family_expenses');
15        NewFamilyExpenses.u_date = current.u_date;
16        NewFamilyExpenses.u_amount = current.u_expense;
17        NewFamilyExpenses.u_expense_details += ">" + current.u_comments + ":" + "Rs." + current.u_expense + "/-";
18        NewFamilyExpenses.insert();
19    }
20
21 }(current, previous);

```

6. Go to the Header and right click there>> click on Save.

Milestone 8: Configure the Relationship

1. Go to All >> In the filter search for Relationships >> Open Relationships.
2. In that, open Daily Expenses Relationship.
3. For Applies to table : Select Family Expenses.
4. In Query with : write the below Query.

```
(function refineQuery(current, parent) {  
  
  // Add your code here, such as current.addQuery(field,  
  value);  
  
  current.addQuery('u_date',parent.u_date);  
  
  current.query();  
  
})(current, parent);
```

The screenshot shows the Salesforce Relationship Builder interface. At the top, the relationship is named 'Daily Expenses' and is set to 'Global' application. The 'Applies to table' dropdown is set to 'Family Expenses [u_family_expenses]', highlighted with a red box and arrow 1. The 'Queries from table' dropdown is set to 'Daily Expenses [u_daily_expenses]'. Below this, a blue informational banner 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 article about the recommended form of the script.' The 'Query with' section contains a code editor with the following JavaScript code, highlighted with a red box and arrow 2:

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

At the bottom, the 'Update' button is highlighted with a red box and arrow 3.

5. Click on Update.

Conclusion:

By developing a family expense tracking application on the ServiceNow platform, users can move from manual, fragmented financial management to an integrated, automated, and insightful system. This solution provides an intuitive user experience and empowers individuals to gain better control over their finances, track their spending effectively, and achieve their budgeting goals, ultimately leading to improved financial literacy and stability.