

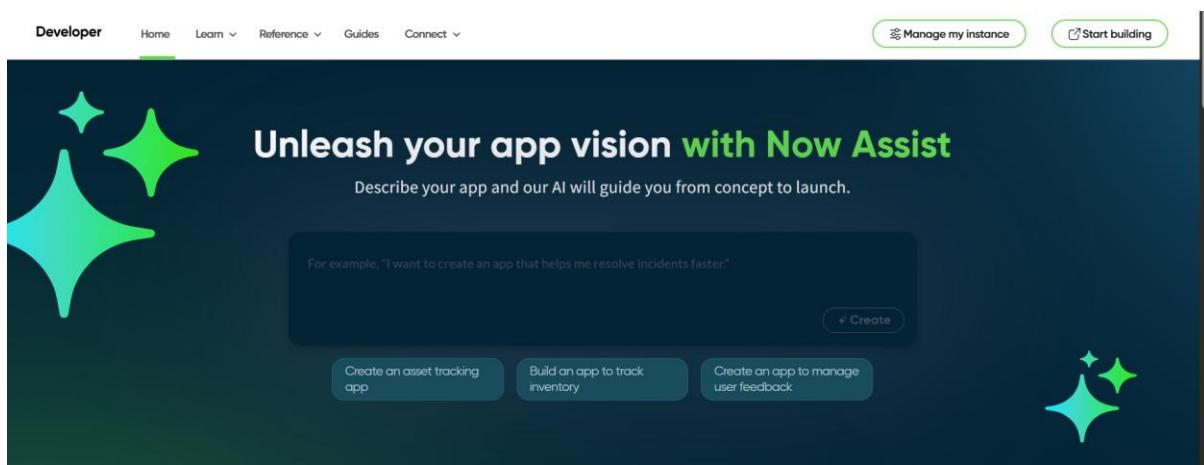
# Calculating Family Expenses using Service Now

## 1. Setting up ServiceNow Instance

Sign up for a developer account at [developer.servicenow.com](https://developer.servicenow.com).

Go to Personal Developer Instance → Request Instance and fill in the required details.

Once your instance is ready, log in using the provided credentials to access ServiceNow.



## 2. Creation of New Update Set

In the filter navigator, search Local Update Set → click New.

A screenshot of the ServiceNow filter navigator. The top navigation bar shows "servicenow" and various menu items like All, Favorites, History, Workspaces, Admin, and a search bar. A sidebar on the left contains a "FAVORITES" section with "No Results" and an "ALL RESULTS" section with a "System Update Sets" category expanded. Under "Local Update Sets", there is a single result highlighted with a black border. Below the results is a "GO FURTHER" section with the text "Power your workflow applications". At the bottom, a URL is visible: "https://dev300245.service-now.com/sys\_update\_set\_list.do?sysparm\_userpref\_module=50047c06c0a8016c0135a14cebcb8191b&amp;sysparm\_clear\_stack=true&amp;sysparm\_clear\_stack=true".

Enter Name: Family Expenses, then click Submit and Make Current.

The screenshot shows the 'Update Set' interface for a record named 'Family Expenses'. The record details are as follows:

Field	Value
Name	Family Expenses
State	In progress
Parent	(Search icon)
Release date	(Search icon)
Install date	
Installed from	
Description	

Metadata on the right side of the screen includes:

- Application: Global
- Created: 2025-10-29 22:10:32
- Created by: admin
- Merged to: (empty)

At the bottom left are 'Update' and 'Delete' buttons.

### 3. Creation of Family Expenses Table

In the filter navigator, search Tables → click New.

Enter Label: Family Expenses, Menu Name: Family Expenditure, then right-click the header and select Save.

The screenshot shows the 'Table' creation interface for a table named 'Family\_Expenses'. The table details are as follows:

Field	Value
Label	Family_Expenses
Name	u_family_expenses

Metadata on the right side of the screen includes:

- Application: Global
- Remote Table: (empty)

### 4. Creation of Columns (Fields)

Add new rows under **Columns** with the following details:

- **Number** – String
- **Date** – Date
- **Amount** – Integer
- **Expense Details** – String (Max length: 800)

Table Family\_Expenses

\* Label: Family\_Expenses  
\* Name: u\_family\_expenses

Application: Global

Remote Table:

Column label	Type	Reference	Max length	Default value	Display
Expense Details	String	(empty)	800		false
Updated by	String	(empty)	40		false
Sys ID	Sys ID (GUID)	(empty)	32		false
Updated	Date/Time	(empty)	40		false
Date	Date	(empty)	40		false
Amount	Integer	(empty)	40		false
Created by	String	(empty)	40		false
Number	String	(empty)	40	javascript:getNextObjNumberPadded();	false
Created	Date/Time	(empty)	40		false
Updates	Integer	(empty)	40		false

Column label	Type	Reference	Max length	Default value	Display
Expense Details	String	(empty)	800		false
Updated by	String	(empty)	40		false
Sys ID	Sys ID (GUID)	(empty)	32		false
Updated	Date/Time	(empty)	40		false
Date	Date	(empty)	40		false
Amount	Integer	(empty)	40		false
Created by	String	(empty)	40		false
Number	String	(empty)	40	javascript:getNextObjNumberPadded();	false
Created	Date/Time	(empty)	40		false
Updates	Integer	(empty)	40		false

+ Insert a new row...

Delete Update Delete All Records

Access Controls (4) Security Data Filters Labels (1) Database Indexes (1) Table Subscription Configuration (1)

Name	Decision Type	Operation	Type	Active	Updated by	Updated
u_family_expenses	Allow If	delete	record	true	admin	2025-10-31 17:39:42
u_family_expenses	Allow If	write	record	true	admin	2025-10-31 17:39:42
u_family_expenses	Allow If	create	record	true	admin	2025-10-31 17:39:41
u_family_expenses	Allow If	read	record	true	admin	2025-10-31 17:39:42

Right-click the header and select **Save**.

## 4. Making Number Field an Auto-Number

Open the Number field → Advanced view.

Check Use dynamic default and select Get Next Padded Number, then click Update.

The screenshot shows the 'Dictionary Entry' screen for the 'Number' field. The top navigation bar includes 'Delete Column', 'Update', and other standard buttons. The left side displays field properties: Table 'Family\_Expenses [u\_family\_expenses]', Type 'String', Column label 'Number', Column name 'u\_number', and Max length '40'. On the right, settings like Application 'Global', Active (checked), Function field, Read only, Mandatory, and Display are shown. A note below states: 'Alters the behavior of a field or functionality that depends on the field. More Info'. The 'Default Value' tab is selected in the bottom navigation, with the 'Use dynamic default' checkbox checked and 'Dynamic default value' set to 'Get Next Padded Number'.

Search Number Maintenance → click New.

Set Table: Family Expenses, Prefix: MFE, and click Submit.

The screenshot shows the 'Number' maintenance screen with a new entry. The table is set to 'Family\_Expenses' and the prefix is 'MFE'. The number is '1,000', application is 'Global', and the number of digits is '7'. Below the form, there are 'Update' and 'Delete' buttons, and a 'Related Links' section with a 'Show Counter' link.

## 5. Configure the Form

Search Family Expenses and open it.

Click New, then right-click the header → Configure → Form Design.

Arrange fields as needed.

Set Number as *Read-only* and make Date and Amount *Mandatory*.

Click Save.

## 6. Creation of Daily Expenses Table

Search Tables → click New.

Enter Label: Daily Expenses, add to Menu: Family Expenditure, then right-click the header and select Save.

Column label	Type	Reference	Max length	Default value	Display
Updated by	String	(empty)	40		false
Sys ID	Sys ID (GUID)	(empty)	32		false
Updates	Integer	(empty)	40		false
Expense	Integer	(empty)	40		false
Family Member Name	Reference	User	32		false
Updated	Date/Time	(empty)	40		false
comments	String	(empty)	800		false
Number	String	(empty)	40	javascript:getNextObjNumberPadded();	false
Created by	String	(empty)	40		false
Date	Date	(empty)	40		false

Dictionary Entries

Table Columns for text Search

1 to 11 of 11 New

Column label Type Reference Max length Default value Display

- Updated by String (empty) 40 false
- Sys ID Sys ID (GUID) (empty) 32 false
- Updates Integer (empty) 40 false
- Expense Integer (empty) 40 false
- Family Member Name Reference User 32 false
- Updated Date/Time (empty) 40 false
- comments String (empty) 800 false
- Number String (empty) 40 javascript:getNextObjNumberPadded() false
- Created by String (empty) 40 false
- Date Date (empty) 40 false
- Created Date/Time (empty) 40 false

Insert a new row...

Delete Update Delete All Records

Access Controls (4) Security Data Filters Labels (1) Database Indexes (2) Table Subscription Configuration (1)

Decision Type Search Actions on selected rows...

Access Controls

Name	Decision Type	Operation	Type	Active	Updated by	Updated
u_daily_expenses	Allow If	create	record	true	admin	2025-10-31 17:56:33
u_daily_expenses	Allow If	write	record	true	admin	2025-10-31 17:56:33
u_daily_expenses	Allow If	read	record	true	admin	2025-10-31 17:56:33
u_daily_expenses	Allow If	delete	record	true	admin	2025-10-31 17:56:33

1 to 4 of 4

## 7. Making Number Field an Auto-Number

Open the Number field → Advanced view.

Enable Use dynamic default and choose Get Next Padded Number, then click Update

Dictionary Entry  
Number View: Advanced\*

\* Table Daily\_Expenses [u\_daily\_expenses] Application Global

\* Type String Active

\* Column label Number Function field

\* Column name u\_number Read only

\* Max length 40 Mandatory

Display

Attributes

Reference Specification Choice List Specification Function Definition Dependent Field Calculated Value Default Value

Use dynamic default

Dynamic default value Get Next Padded Number

Delete Column Update

Search Number Maintenance → New.

Set Table: Daily Expenses, Prefix: DFE, and click Submit.

Number DFE

\* Table: Daily\_Expenses

Prefix: DFE

\* Number: 1,000

Application: Global

Number of digits: 7

Update Delete

Related Links  
Show Counter

## 8. Configure the Form

Search Daily Expenses and open it.

Click New, then right-click the header → Configure → Form Design.

Arrange fields as needed.

Set Number as *Read-only* and make Date and Family Member Name *Mandatory*.

Click Save.

Daily\_Expenses [u\_daily\_e] Default view Form Design

Fields Field Types Filter

**Fields**

- # Created
- # Created by
- # Updated
- # Updated by
- # Updates

**Formatters**

- # Activities (filtered)
- # Contextual Search Results
- # Ratings

**Form Design**

2 Column

Number	Date
Expense	

1 Column

Comments
Member name

## 9. Creating Relationship Between Tables

Search Relationships → click New.

Set Name: Daily Expenses, Applies to Table: Family Expenses, Related Table: Daily Expenses, then click Save.

Relationship Daily\_Expenses

Name: Daily\_Expenses

Advanced:

Application: Global

Applies to table: Family\_Expenses [u\_family\_expenses]

Queries from table: Daily\_Expenses [u\_daily\_expenses]

```

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.

Query with: Turn on ECMAScript 2021 (ES12) mode ⓘ
1 (function refineQuery(current, parent) {
2   // Add your code here, such as current.addQuery(field, value);
3   current.addQuery('u_date',parent.u_date);
4   current.query();
5 }
6 })(current, parent);
7 
```

Run Query Diagnostics | Update | Delete

## 10. Configuring Related List

Open Family Expenses → New → Configure > Related Lists → add Daily Expenses → Save.

Configuring related lists on Family\_Expenses form

Available: Attachments

Selected: Daily\_Expenses

View name: Default view

Related Links: Show versions, Related list performance diagnostics

Cancel | Save

## 11. Business Rule Creation

Navigate to All → Business Rules → New.

Name: Family Expenses BR

Table: Daily Expenses

Add query: required

```

(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
}

```

## 12. Configure Relationship

Go to All → Relationships and open Daily Expenses Relationship.  
Set Applies to table: Family Expenses.

Add Query:

```
(function refine Query (current, parent) {
    current.add Query('update', parent.u_date);
    current.query ();
}) (current, parent);
```

Query with  Turn on ECMAScript 2021 (ES12) mode [?](#)



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

Run Query Diagnostics [Update](#) [Delete](#)

Update the Relationships by clicking on update.

- Built in **ServiceNow** to manage family spending.
  - **Daily Expenses** table records:
    - Date, Amount, Comments, Family Member
  - **Family Expenses** table shows:
    - Total spent per date, with summary
  - Tables are **linked**:
    - Daily entries update family totals automatically
  - Helps track spending and keep financial records organized.

Daily_Expenses					Number	Search	Actions on selected rows...	New
All								
	Number	comments	Date	Expense	Family Member Name			
	DFE0001003	money	2025-10-31	500	Abraham Lincoln			
	DFE0001002	Mobile	2025-11-01	500	Abel Tuter			

	Family_Expenses	Number	Search	Actions on selected rows...	New
All					
	Number	Amount	Date	Expense Details	
	MFE0001011	1,000	2025-10-31	new	
	MFE0001009	500	2025-11-01	Mobile	

The screenshot shows the ServiceNow interface for managing family expenses. At the top, a modal window titled 'Family\_Expenses' displays a record with number 'MFE0001009'. The form fields include 'Number' (MFE0001009), 'Expense Details' (Mobile), 'Date' (2025-11-01), and 'Amount' (500). Below the modal is a list view for 'Daily\_Expenses' with one item: DFE0001002, Mobile, 2025-11-01, 500, Abel Tuter.

Number	comments	Date	Expense	Family Member Name
DFE0001002	Mobile	2025-11-01	500	Abel Tuter

## Conclusion:

In conclusion, the *Family Expenses Calculation System* built on ServiceNow provides an efficient and organized way to manage household finances. By leveraging ServiceNow's automation and data management capabilities, the system simplifies expense tracking, ensures accuracy, and offers real-time insights into family spending patterns. This project not only enhances financial transparency but also promotes better budgeting and informed decision-making—ultimately contributing to improved financial stability and well-being for families.

Done By,

**Team ID:**

Team Member: H.Naveen Sam Raj

Team Member: P.Selva Bharath

Team Member: E.Vasanthan

Team Member: T.Sirajudeen

**Thank You!**