

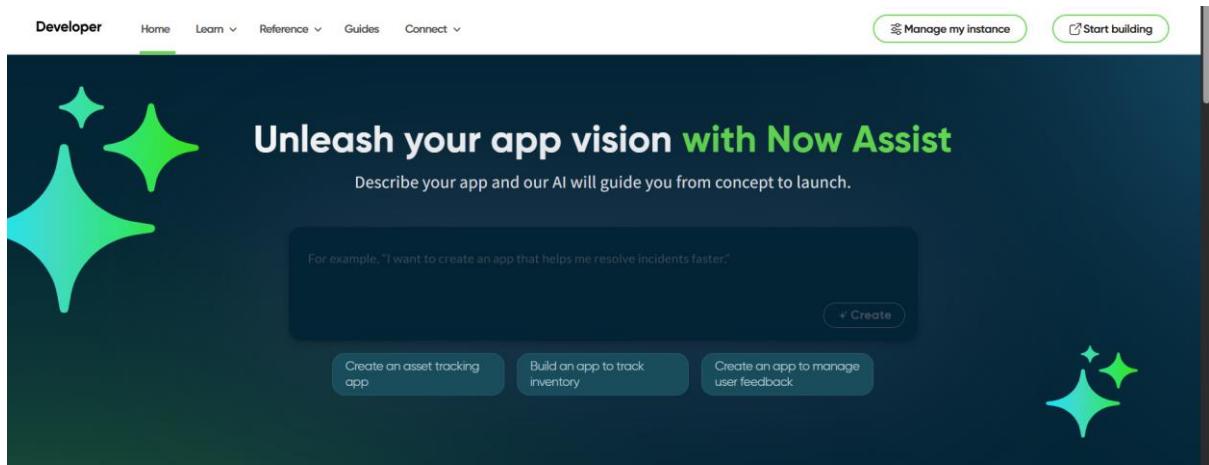
Calculating Family Expenses using Service Now

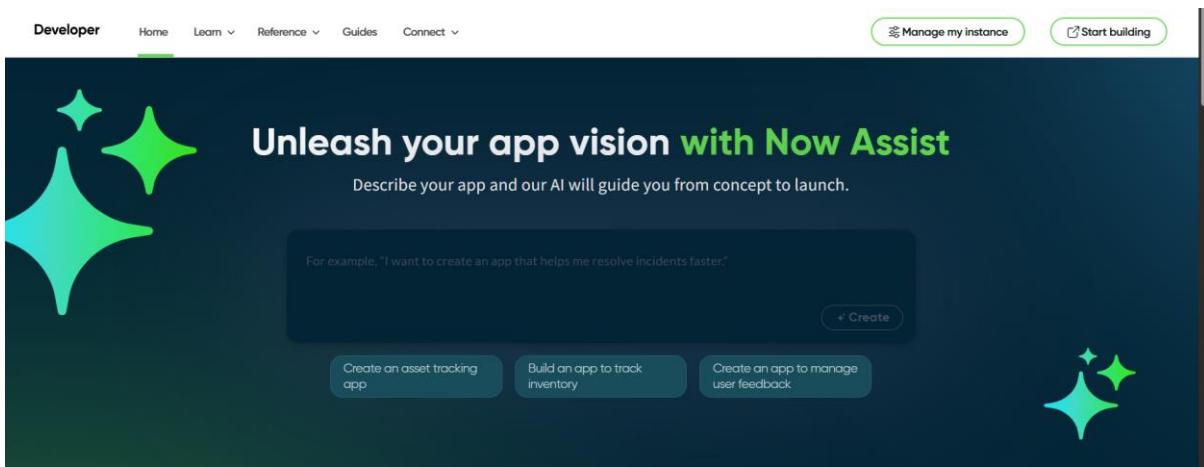
1. Setting up ServiceNow Instance

Sign up for a developer account at 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 search bar at the top contains the query '.local Update Sets'. The results pane shows 'FAVORITES' with 'No Results' and 'ALL RESULTS' with 'Local Update Sets' selected under 'System Update Sets'. A tooltip indicates 'personalization, enhancements'. Below the results is a 'GO FURTHER' section with the heading 'Power your workflow applications'. The URL in the browser address bar is https://dev300245.service-now.com/sys_update_set_list.do?sysparm_userpref_module=50047c06c0a8016c0135a14cebc8191b&sysparm_clear_stack=true&sysparm_clear_stack=true. The overall theme is dark with blue and green highlights.

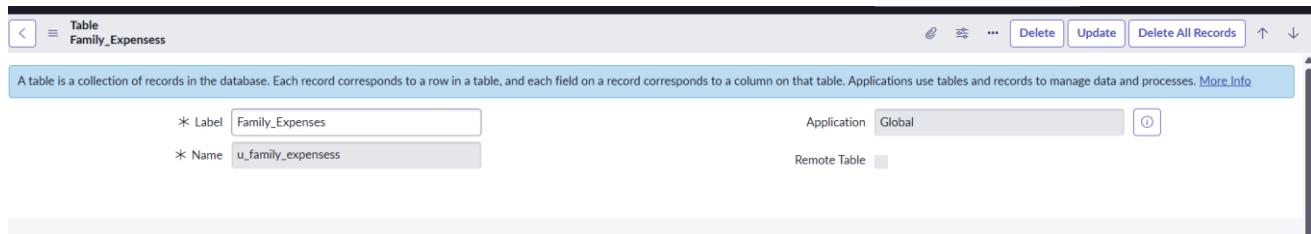
Enter Name: Family Expenses, then click Submit and Make Current.

A screenshot of the ServiceNow Update Set creation form. The title bar says 'Update Set Family Expenses'. The form fields include: Name (Family Expenses), State (In progress), Parent (dropdown with a search icon), Release date (date picker), Install date (date picker), Installed from (text input), and Description (text area). On the right, there are application details: Application (Global), Created (2025-10-29 22:10:32), Created by (admin), and Merged to (text input). At the bottom 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.



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)

A screenshot of the 'Columns' tab in the table configuration interface. The top shows the table name 'Family_Expenses' and its application 'Global'. Below is a table of columns with the following data:

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

The screenshot shows the 'Table Columns' configuration screen. At the top, there are tabs for 'Columns', 'Controls', and 'Application Access'. Below the tabs is a search bar with the placeholder 'Search'. A message 'Dictionary Entries' is displayed above a table. The table has columns: 'Column label', 'Type', 'Reference', 'Max length', 'Default value', and 'Display'. The rows show columns for 'Expense Details' and 'Number'. For 'Expense Details', the 'Default value' is 'javascript:getNextObjNumberPadded();'. At the bottom of the table, there is a note 'Insert a new row...'. Below the table are buttons for 'Delete', 'Update', and 'Delete All Records'.

The screenshot shows the 'Access Controls' configuration screen. At the top, there are tabs for 'Access Controls (4)', 'Security Data Filters', 'Labels (1)', 'Database Indexes (1)', and 'Table Subscription Configuration (1)'. Below the tabs is a search bar with the placeholder 'Search'. A message 'Access Controls' is displayed above a table. The table has columns: 'Name', 'Decision Type', 'Operation', 'Type', 'Active', 'Updated by', and 'Updated'. The rows show four entries for 'u_family_expenses' with different decision types (Allow If) and operations (delete, write, create, read). The last column 'Updated' shows dates from 2025-10-31 17:39:42 to 2025-10-31 17:39:41. At the bottom of the table, there is a note '1 to 4 of 4'.

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' advanced view for the 'Number' field. At the top, there is a back button, a search bar, and a toolbar with icons for 'Delete Column', 'Update', and others. The main area has sections for 'Table' (Family_Expenses [u_family_expenses]), 'Type' (String), 'Column label' (Number), 'Column name' (u_number), and 'Max length' (40). To the right, there are sections for 'Application' (Global), 'Active' (checked), 'Function field' (unchecked), 'Read only' (unchecked), 'Mandatory' (unchecked), and 'Display' (unchecked). Below this is a note about altering field behavior. At the bottom, there are tabs for 'Choice List Specification', 'Calculated Value', and 'Default Value' (which is selected). A note says 'The Default value specifies what value the field has when first displayed.' Below this are fields for 'Use dynamic default' (checked) and 'Dynamic default value' (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 the title 'Number MFE'. It has fields for 'Table' (Family_Expenses), 'Prefix' (MFE), 'Number' (1,000), 'Application' (Global), and 'Number of digits' (7). Buttons for 'Update' and 'Delete' are visible at the bottom left, along with 'Related Links' and 'Show Counter'.

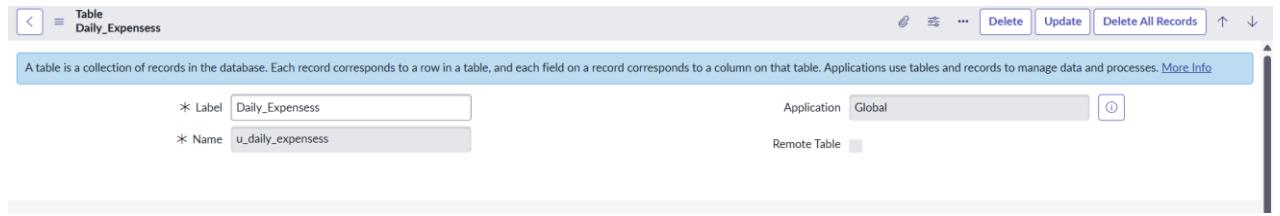
5. Configure the Form

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.

The screenshot shows the 'Form Design' interface for the 'Family_Expenses [u_family_expenses]' table. The left sidebar lists fields like Created, Updated, and Expense Details. The main area shows a 1-column layout for the first section and a 2-column layout for the second section, which contains 'Date' and 'Amount' fields.

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.

The screenshot shows the 'Table Columns' configuration screen for the 'Daily_Expenses' table. It has tabs for 'Columns', 'Controls', and 'Application Access'. The 'Columns' tab is selected. It displays a list of columns with their properties:

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

At the bottom, there are buttons for 'Delete', 'Update', and 'Delete All Records'. The 'Controls' tab is also visible above the table.

Access Controls (4) Security Data Filters Labels (1) Database Indexes (2) Table Subscription Configuration (1)							
Decision Type ▾		Actions on selected rows...					
Access Controls							
□	Q	Name	Decision Type ▾	Operation	Type	Active	Updated by
		u_daily_expenses	Allow If	create	record	true	admin
		u_daily_expenses	Allow If	write	record	true	admin
		u_daily_expenses	Allow If	read	record	true	admin
		u_daily_expenses	Allow If	delete	record	true	admin

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] * Type: String * Column label: Number * Column name: u_number * Max length: 40

Application: Global Active: Function field: Read only: Mandatory: Display:

Attributes:

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.

Form Design

Daily_Expenses [u_daily_expenses]

Fields

Number (Read-only)

Date

Expense

Comments

Member name

Formatters

Activities (filtered)

Contextual Search Results

Ratings

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.

Name: Daily_Expense
Advanced:

Application: Global
Applies to table: Family_Expenses [u_family_expenses]
Queries from table: Daily_Expense [u_daily_expense]

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);
```

Run Query Diagnostics | Update | Delete

10.Configuring Related List

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

Available: Attachments
Selected: Daily_Expense

View name: Default view
Cancel | Save

Related Links:
Show versions | Related list performance diagnostics

11.Business Rule Creation

Navigate to All → Business Rules → New.

Name: Family Expenses BR

Table: Daily Expenses

Add query: required

The screenshot shows the 'Business Rule' configuration page for 'Family Expenses BR'. At the top, there are fields for 'Name' (Family Expenses BR), 'Table' (Daily_Expenses [u_daily_expenses]), 'Application' (Global), and checkboxes for 'Active' and 'Advanced'. Below this, the 'Actions' tab is selected under 'When to run'. It shows 'When' set to 'before' and 'Order' set to 100. Under 'Actions', 'Insert' and 'Update' are checked, while 'Delete' and 'Query' are unchecked. Below these are 'Filter Conditions' buttons for 'Add Filter Condition' and 'Add OR Clause', along with dropdowns for 'choose field', 'oper', and 'value'. The 'Advanced' tab is also visible. In the bottom half of the screen, the 'Advanced' tab is selected, showing a script editor with the following code:

```
1 (function executeRule(current, previous /*null when async*/) {
2
3
4 var FamilyExpenses = new GlideRecord('u_family_expenses');
5
6 FamilyExpenses.addQuery('u_date',current.u_date);
7
8 FamilyExpenses.query();
9
10 if(FamilyExpenses.next())
11 {
12
13   FamilyExpenses.u_amount += current.u_expense;
14
15   FamilyExpenses.u_expense_details += ">" + current.u_comments + ":" + "Rs." + current.u_expense + "/";
16
17   FamilyExpenses.update();
18
19 }
20
21
22 else
23 }
```

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);

```



The screenshot shows a ServiceNow interface for editing a query. At the top, there's a toolbar with various icons. Below the toolbar is a code editor window containing the following JavaScript code:

```

Query with Turn on ECMAScript 2021 (ES12) mode ⓘ
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);

```

Below the code editor, there are three buttons: "Run Query Diagnostics", "Update", and "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.



The screenshot shows a ServiceNow grid view for the "Daily_Expenses" table. The grid has the following columns: Number, comments, Date, Expense, and Family Member Name. There are two rows of data:

All	Number	comments	Date	Expense	Family Member Name
	DFE0001003	money	2025-10-31	500	Abraham Lincoln
	DFE0001002	Mobile	2025-11-01	500	Abel Tuter

The screenshot displays two ServiceNow modules: Family_Expenses and Daily_Expenses.

Family_Expenses:

Number	Amount	Date	Expense Details
MFE0001011	1,000	2025-10-31	new
MFE0001009	500	2025-11-01	Mobile

Detailed View of Family_Expenses (MFE0001009):

Number: MFE0001009	Expense Details: Mobile
Date: 2025-11-01	Amount: 500

Daily_Expenses:

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: NM2025TMID06301

Team Leader: Ranish T

Team Member: Simiyon J

Team Member: Uthaya Kumar M

Thank You!