

SRI VIDYA COLLEGE OF ENGINEERING & TECHNOLOGY

VIRUDHUNAGAR 626005

**Approved by AICTE, New Delhi and Affiliated to
Anna University, Chennai**



**Subject Name : ServiceNow Administrator (NM1051)
(Under Naan Mudhalvan Scheme)**

Project Title : Calculating Family Expenses using ServiceNow

Team ID: NM2025TMID04202

Team Members

K.Sriram - 922022104031

R.Sheela Devi -922022104030

G.Vijaya Prabakar-922022104303

L.Yogesh-922022104041

Calculating Family Expenses using ServiceNow

1. Objective

The main objective of this project is to **develop a family expense management system** using the **ServiceNow platform**. The system helps users efficiently track, categorize, and manage family expenses. It aims to promote financial awareness and better budgeting within the household.

2. Introduction

In today's fast-paced world, managing family expenses is a critical part of financial stability. Manual tracking often leads to confusion, errors, and lack of insight into spending patterns.

This project leverages **ServiceNow**, a robust cloud-based platform, to create an automated and user-friendly **Family Expense Tracker**.

The application allows users to record expenses, set budgets, categorize spending, and generate real-time reports.

By building this project, we demonstrate how ServiceNow can be extended beyond IT service management to real-world, practical use cases like family budgeting.

3. Project Scope

- Expense Categorization
 - Budget Setting and Comparison
 - Real-time Expense Tracking
-

5. Tools and Technologies Used

Tool/Technology	Description
ServiceNow	Cloud platform for workflow automation
Update Sets	To capture and move customizations
Tables & Fields	Used to store expense records
Forms & Lists	For user interaction and record display
Reports Module	For data visualization and analytics
Browser	Any modern browser (Chrome/Edge)

6. System Requirements

Hardware Requirements

- Processor: Dual Core or higher
- RAM: Minimum 4 GB
- Internet Connection: Stable broadband

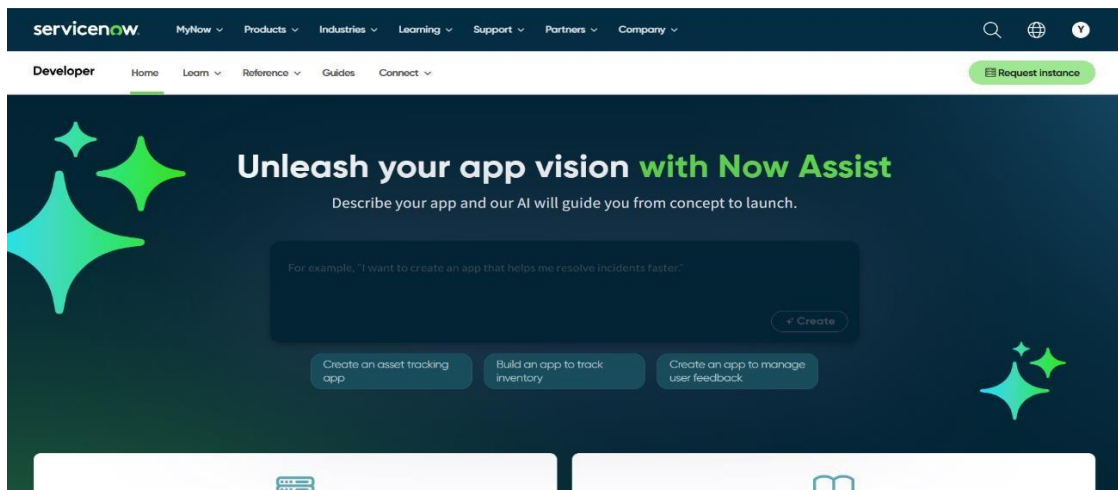
Software Requirements

- Web Browser (Google Chrome preferred)
- ServiceNow Developer Instance
- ServiceNow Account (developer.servicenow.com)

7. Project Implementation Steps

Step 1: Setting up ServiceNow Instance

1. Visit <https://developer.servicenow.com>
2. Sign up for a free developer account.
3. Log in to your instance to start customizing.



Step 2: Creation of a New Update Set

1. Navigate to **All** → **Local Update Sets** → **New**
2. Enter details:
 - **Name:** Family Expenses
3. Click **Submit and Make Current**
This ensures all future configurations are captured under the same update set.

servicenow All Favorites History Workspaces Update Set - Create New Update Set Search

Update Set New record

* Name: Family Expenses Application: Global

State: In progress

Parent:

Release date:

Description:

Submit Submit and Make Current

Step 3: Creation of Family Expenses Table

1. Navigate to **All** → **Tables** → **New**
2. Enter details:
 - **Label:** Family Expenses
 - **Name:** (Auto-populated)
 - **New menu name:** Family Expenditure
3. Right-click the header and select **Save**.

* Label: Family Expenses

* Name: ust_family_expenses

Application: Global

Remote Table: ☒

Create module: ☒

Create mobile module: ☒

Step 4: Creation of Columns (Fields)

In the **Columns** section, add the following fields:

Column	Label	Type	Max Length
Number		String	-
Date		Date	-
Amount		Integer	-
Expense Details		String	800

After adding all fields, **Save** the table.

servicenow

All Favorites History Workspaces Admin

Table - Family Expenses

Search

Table Family Expenses

Columns Controls Application Access

Table Columns for text Search

Dictionary Entries

Column label	Type	Reference	Max length	Default value	Display
Updated by	String	(empty)		40	false
Created by	String	(empty)		40	false
Updated	Date/Time	(empty)		40	false
Sys ID	Sys ID (GUID)	(empty)		32	false
Created	Date/Time	(empty)		40	false
Updates	Integer	(empty)		40	false
Number	String				false
Date	Date				false
Amount	Integer				false
Expense Details	String		800		false
Insert a new row...					

Delete Update Delete All Records

Related Links

Step 5: Auto-Number for Family Expenses

Open the **Number** field → **Advanced View** → enable **Use dynamic default** with **Get Next Padded Number** → **Update**.

Then go to **Number Maintenance** → **New**, set **Table: Family Expenses**, **Prefix: MFE**, and click **Submit**.

Step 6: Configure Family Expenses Form

Go to **Family Expenses** → **New** → **Configure** → **Form Design**.

Make **Number** read-only, and set **Date** & **Amount** as mandatory → **Save**.

Family Expenses [u_family_expenses] 2 Columns

Number Date Amount

Expense Details 1 Column

Step 7: Create Daily Expenses Table

Navigate to **Tables** → **New**, name it **Daily Expenses**, and link to **Family Expenditure**. Add fields: **Number**, **Date**, **Expense**, **Family Member Name**, **Comments**, then **Save**.

* Label Daily Expenses

* Name u_daily_expenses

Extends table

Application Global

Create module

Create mobile module

Add module to menu Family Expenditure

Step 8: Auto-Number for Daily Expenses

Open **Number** field → enable **Get Next Padded Number** → Update.

In **Number Maintenance**, create new with **Table: Daily Expenses**, **Prefix: MDE**, then Submit.

The first screenshot shows the 'Default Value' configuration for a field. It includes a tab labeled 'Default Value' and a text box explaining that the default value specifies what value the field has when first displayed. Below this, there is a checkbox for 'Use dynamic default' which is checked, and a dropdown menu for 'Dynamic default value' set to 'Get Next Padded Number'. The second screenshot shows the 'Number Maintenance' form. It has a header 'Number New record'. The form contains several fields: 'Table' set to 'Daily Expenses', 'Prefix' set to 'MDE', 'Number' set to '1,000', 'Application' set to 'Global', and 'Number of digits' set to '7'. A 'Submit' button is located at the bottom left of the form.

Step 9: Configure Daily Expenses Form

Go to **Daily Expenses** → **New** → **Configure** → **Form Design**.

Set **Number** read-only, and **Date & Family Member Name** mandatory → Save.

The screenshot shows the 'Form Design' interface for the 'Daily Expenses' form. The form is titled 'Daily Expenses [u_daily_expenses]' and has a '2 Column' dropdown. The form layout consists of a grid with four fields: 'Number', 'Date', 'Family Member Name', and 'Expense'. The 'Number' field is marked as read-only. Below the grid is a 'Comments' field. The interface includes a header bar with the title 'Daily Expenses [u_daily_expenses]' and a '2 Column' dropdown.

Step 10: Create Relationship

Open **Relationships** → **New**, set **Name: Daily Expenses**, **Applies to Table: Family Expenses**, and link to **Daily Expenses** → Save.

Add **Daily Expenses** as a related list in **Family Expenses** → Save.

servicenow All Favorites History Workspaces Relationship - New Record

Name: Application:

Advanced: ☐ Applies to table:

Queries from table:

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
3     // Add your code here, such as current.addQuery(field, value);
4
5 })(current, parent);

```

Submit

Step 11: Create Business Rule

Go to **Business Rules** → **New**, name it **Family Expenses BR**, and select **Table: Daily Expenses**. Enable **Advanced**, **Insert**, **Update** and add script to auto-update Family Expenses totals → **Save**.

Relationship: **Daily Expenses** Application:

Advanced: ☐ Applies to table:

Queries from table:

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

```

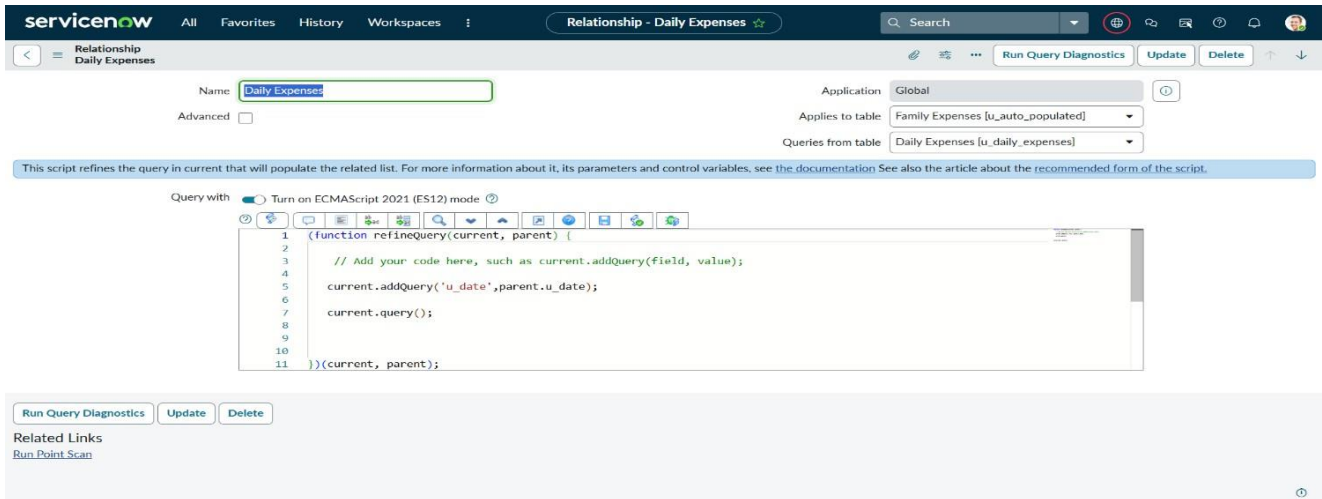
1 (function refineQuery(current, parent) {
2
3     // Add your code here, such as current.addQuery(field, value);
4     current.addQuery('u_date', parent.u_date);
5     current.query();
6
7 })(current, parent);

```

Update Delete

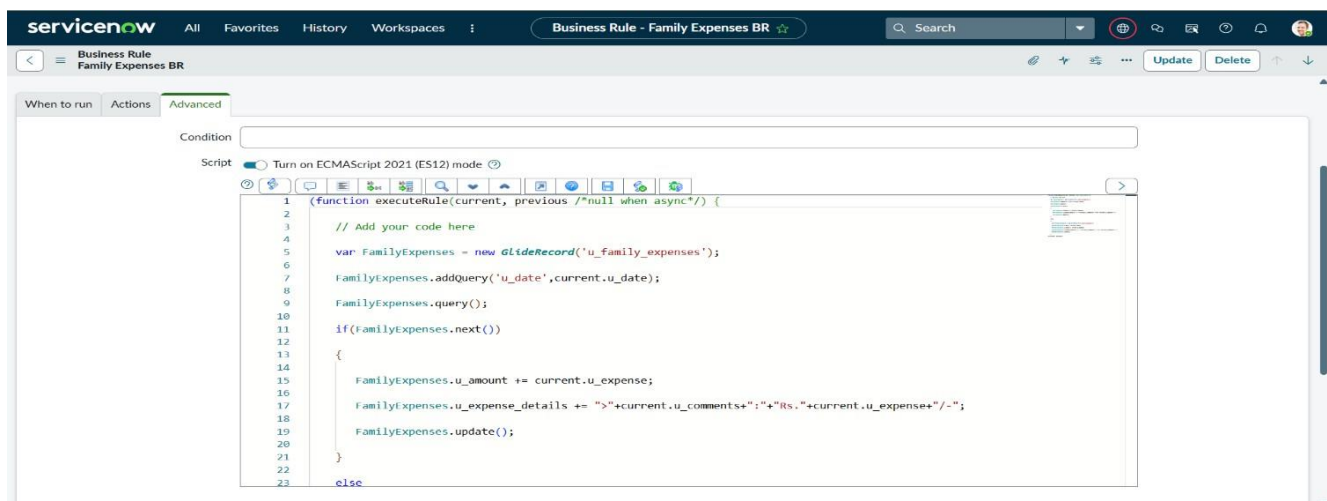
Step 12: Refine Relationship Query

Open **Daily Expenses Relationship**, set **Applies to: Family Expenses**. Add query script to match **u_date** fields between tables → **Update**.



Step 13: Testing & Output

Add test entries in **Daily Expenses** and verify automatic updates in **Family Expenses**. Check related lists, totals, and generate date-wise or member-wise reports.



Conclusion

A complete ServiceNow-based expense tracker with automated calculations, dynamic relationships, and real-time reporting — demonstrating workflow automation for practical financial management.

