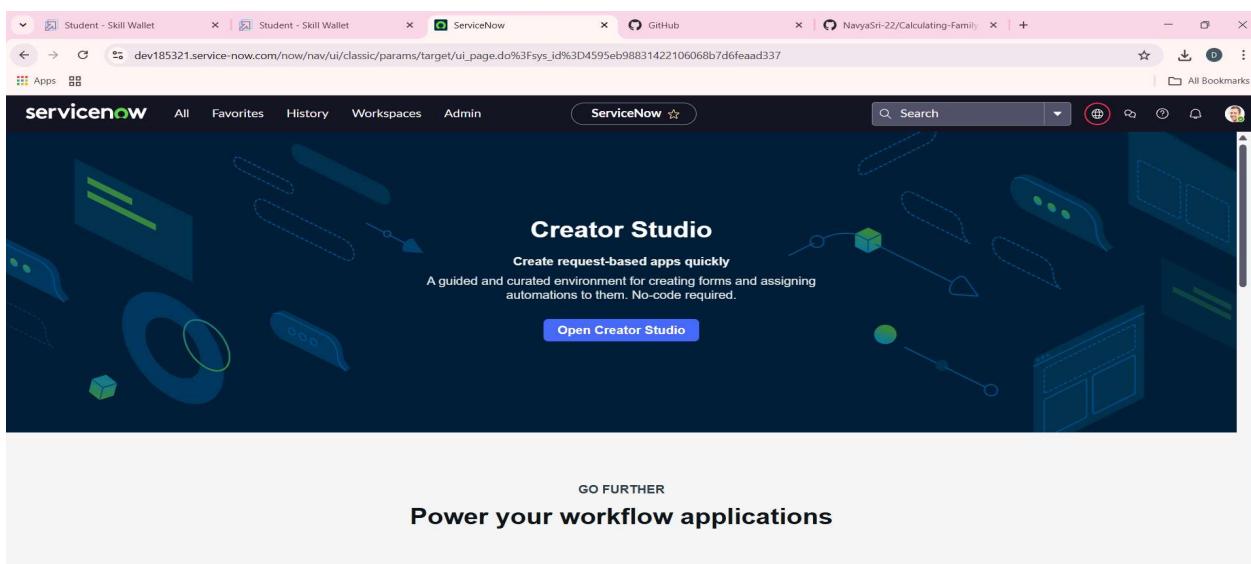


Calculating Family Expenses using Service Now

The project aims to develop a comprehensive expense calculation system using ServiceNow. This system will enable users to track and manage family expenses efficiently. It will include features such as expense categorization, budget setting, real-time tracking, and reporting capabilities. Utilizing ServiceNow's robust platform, the project will ensure seamless integration, user-friendly interface, and scalability to accommodate varying family sizes and financial complexities. The end goal is to empower users with the tools they need to make informed financial decisions and promote financial well-being within the family unit.

Setting up ServiceNow Instance

1. Sign up for a developer account on the ServiceNow Developer site "<https://developer.servicenow.com>".
2. Once logged in, navigate to the "Personal Developer Instance" section.
3. Click on "Request Instance" to create a new ServiceNow instance.
4. Fill out the required information and submit the request.
5. You'll receive an email with the instance details once it's ready.
6. Log in to your ServiceNow instance using the provided credentials.
7. Now you will navigate to the ServiceNow.



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 top navigation bar includes tabs for Student - Skill Wallet, Update Sets | ServiceNow, GitHub, and NavyaSri-22/Calculating-Family. The main menu has links for All, Favorites, History, Workspaces, Admin, and Update Sets. A search bar and various icons are also present.

The central area displays the 'Update Set' form for 'Family Expenses'. The form fields include:

- Name: Family Expenses
- State: In progress
- Parent: (empty)
- Release date: (empty)
- Install date: (empty)
- Installed from: (empty)
- Description: (empty)
- Application: Global
- Created: 2025-07-08 10:25:43
- Created by: admin
- Merged to: (empty)

Below the form, there is a 'Related Links' section with links for 'Merge With Another Update Set' and 'Scan Update Set'. A 'Customer Updates' table is shown, with one row listed:

Created	Type	View	Target name	Updated by	Remote update set	Action
2025-07-14 00:00:25	Password Reset Process	(empty)	Service-Desk Password Reset for Local ServiceNow	admin	(empty)	INSERT_OR_UPDATE

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

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

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.

The screenshot shows the ServiceNow interface with the 'Table - Family Expenses' application open. The 'Columns' tab is selected, displaying a list of columns:

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

Buttons at the bottom include 'Delete', 'Update', and 'Delete All Records'.

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.
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
9. Click on Submit.

The screenshot shows the ServiceNow Dictionary Entry - Number page. The 'Default Value' tab is selected. A note says: 'The Default value specifies what value the field has when first displayed.' Below is a section for 'Dynamic default value' with a dropdown set to 'Get Next Padded Number'. At the bottom, there are 'Delete Column' and 'Update' buttons.

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.
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.

The screenshot shows the ServiceNow Form Design page for the 'Family Expenses [u_family]' table. The main area is titled 'Form Design' and contains two sections: one with a 2-column layout and another with a 1-column layout. The 2-column layout includes fields for Number, Date, and Amount. The 1-column layout is for 'Expense Details'. On the left, there's a sidebar with tabs for Fields, Field Types, and Formatters, and a list of fields including Created, Updated, and Expense Details.

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

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.

Making Number Field an Auto-Number

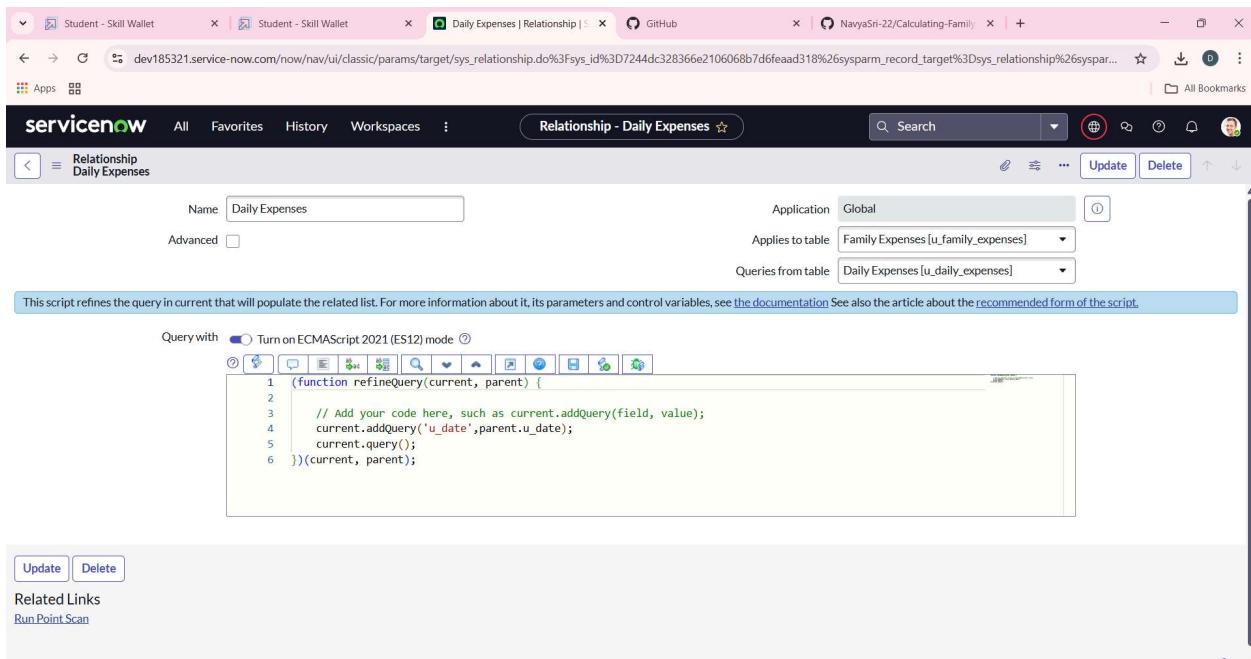
1. Double click on the Number Field/Column.
2. Go down and double click on Advanced view
3. In Default Value:
Use dynamic default : check the box
Dynamic default value : Get Next Padded Number
4. Click on Update.
5. Go to All >> In the filter search for Number Maintenance >> select Number Maintenance
6. Click on New.
7. Enter the below Details:
Table : Family Expenses
Prefix : MFE
9. Click on Submit.

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.
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.

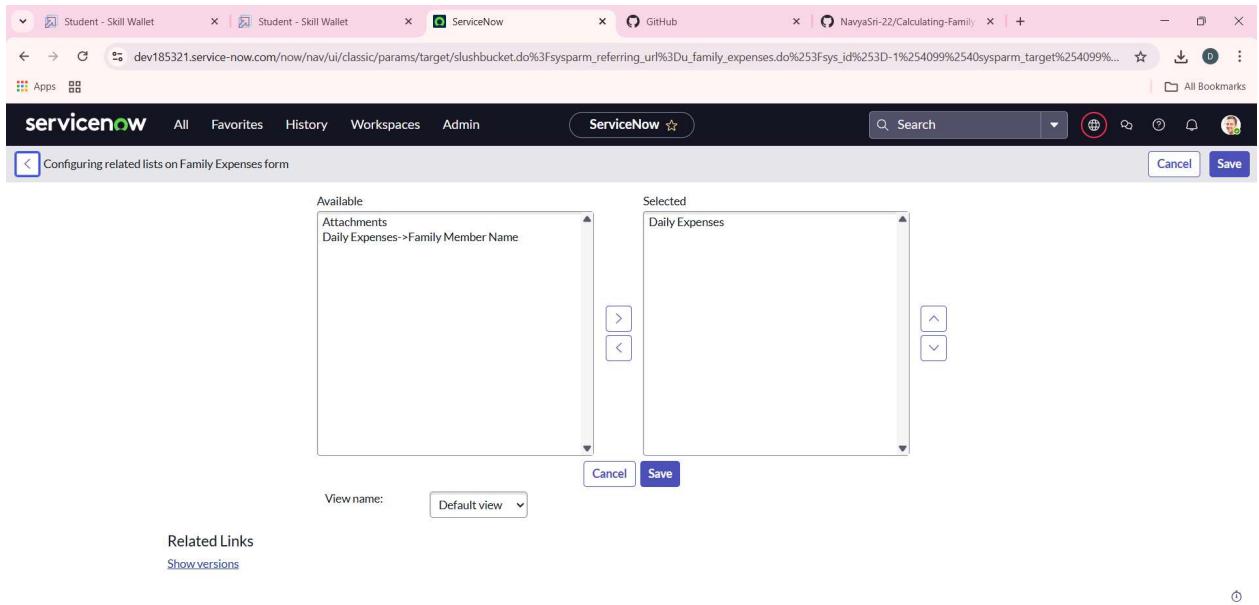
Creation of Relationship between Family Expenses and Daily Expenses tables

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.



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



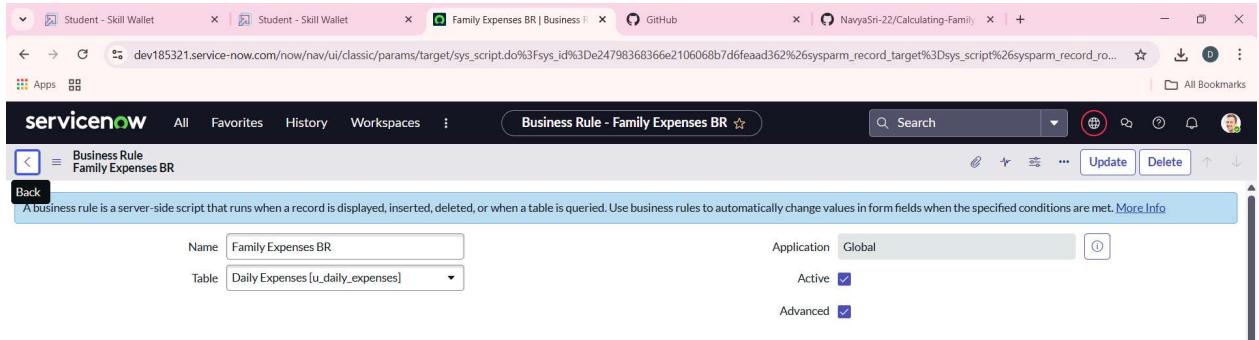
Creation of Business Rules

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:

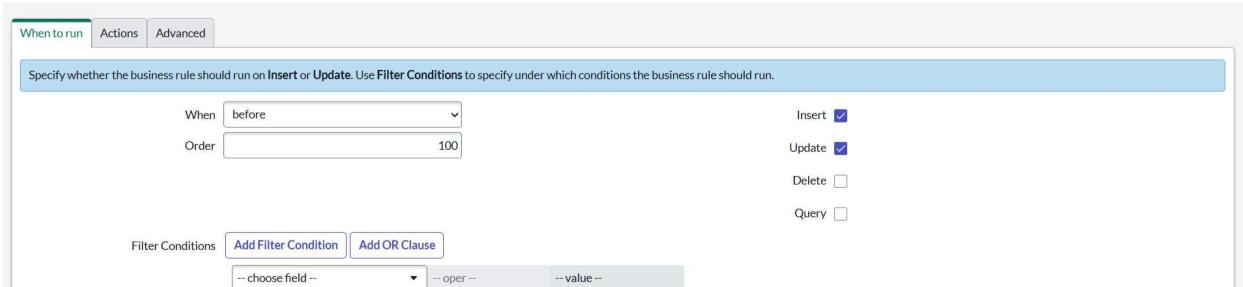
Name : Family Expenses BR

Table : Select Daily Expenses

Check Advanced



4. In when to run Check Insert and Update



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);
```

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

```

(function executeRule(current, previous /*null when async*/) {
    // Add your code here
    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);

```

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);

```

5. Click on Update.

The screenshot shows a ServiceNow browser window with multiple tabs open. The active tab is titled 'Relationship - Daily Expenses'. The interface includes a navigation bar with 'Relationships' and 'Daily Expenses' selected. Below the navigation is a search bar and a toolbar with 'Update' and 'Delete' buttons. The main area contains configuration fields: 'Name' set to 'Daily Expenses', 'Application' set to 'Global', 'Applies to table' set to 'Family Expenses [u.family_expenses]', and 'Queries from table' set to 'Daily Expenses [u.daily_expenses]'. 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 article about the recommended form of the script.' A code editor window displays the following ECMAScript 2021 (ES12) code:

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

At the bottom of the page are 'Update' and 'Delete' buttons, and a 'Related Links' section with a 'Run Point Scan' link.

Conclusion

In conclusion, implementing the "Calculation of Family Expenses Using ServiceNow" project offers numerous benefits to the household. By leveraging the robust capabilities of ServiceNow, family members can efficiently submit, track, and manage their expenses in a centralized and streamlined manner.