

Calculating Family Expenses

Team Leader: SWETHA S

Team member: MUTHU SELVI P

Team member: DHANAM P

PROBLEM STATEMENT:

- Managing household finances is essential for every family to ensure financial stability and better planning. This project aims to develop a simple and efficient system using ServiceNow to record, track, and calculate monthly family expenses. ServiceNow, being a powerful cloud-based platform, offers custom application development that can be leveraged for personal finance tracking.

OBJECTIVE:

- To build a Family Expense Management System using ServiceNow that allows users to:

- Enter daily or monthly expenses under different categories (e.g., groceries, rent, transport, education, etc.)
- View summary reports of total and category-wise expenses
- Set monthly budgets and receive alerts if spending exceeds limits
- Generate downloadable reports for any time period

SKILL:

- 1. ServiceNow Application Development**
 - Creating custom tables
 - Designing forms and views
 - Using Flow Designer for automation
- 2. Workflow & Automation**
 - Using ServiceNow Flow Designer to automate calculations and notifications
- 3. Database Management**
 - Creating and managing custom tables for storing expenses and budgets
- 4. Reporting & Dashboard Design**
 - Designing ServiceNow reports
 - Building graphical dashboards for data visualization

TASK INITIATION:

MILESTONE 1: Setting up ServiceNow Instance

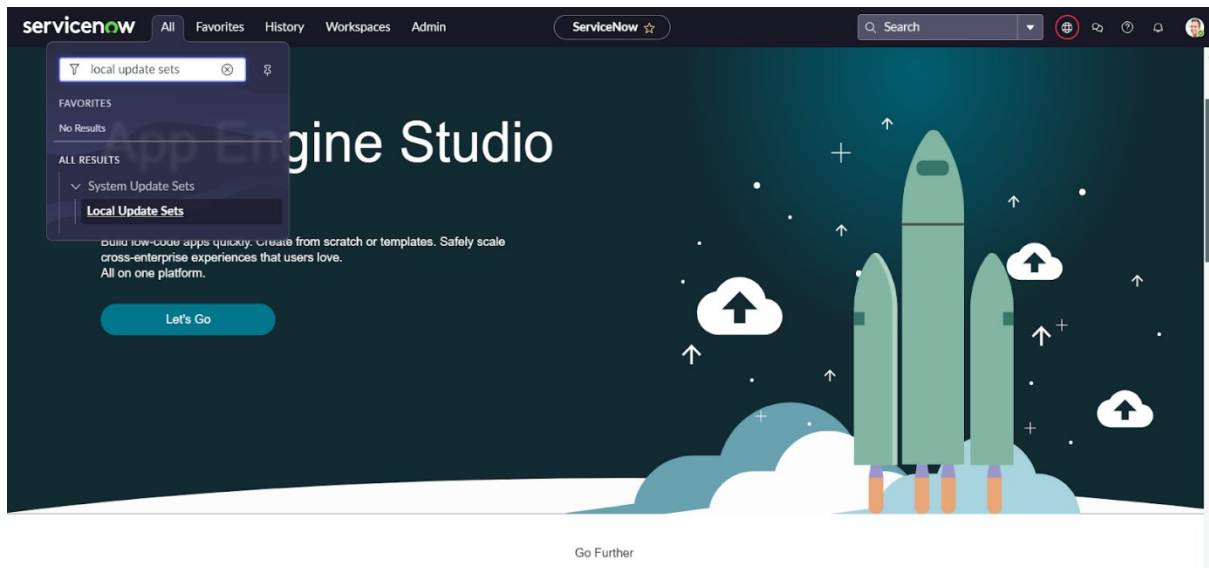
ACTIVITY 1: Setting up ServiceNow Instance

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.

MILESTONE 2: Creation of New Update Set

ACTIVITY 1: Creation of New Update Set

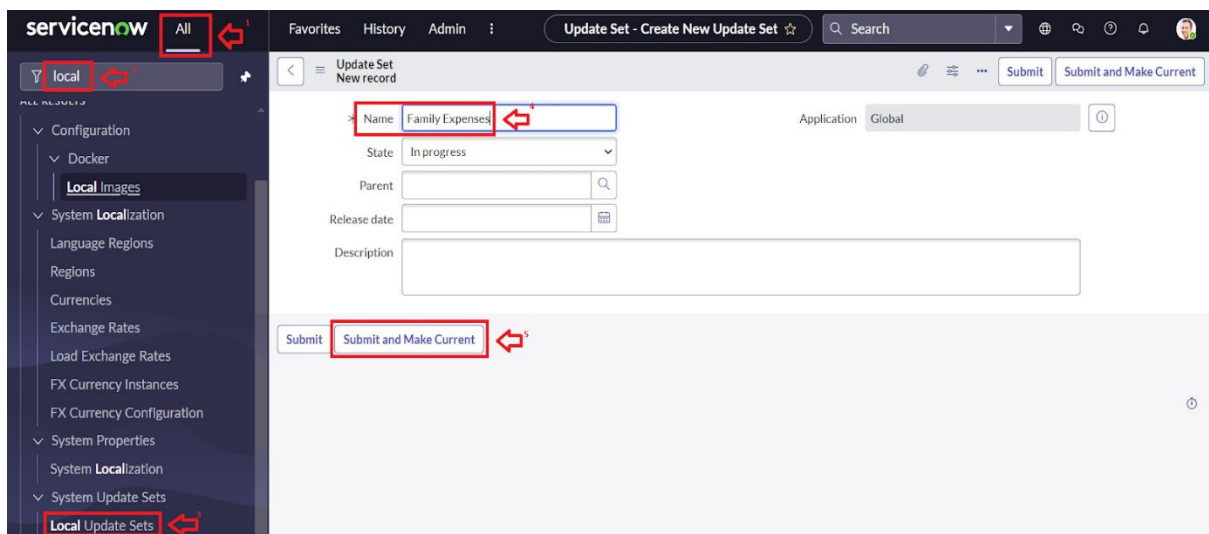
1. Go to All >> In the filter search for Local Update set > click on New.



2. Enter the Details as:

Name: Family Expenses

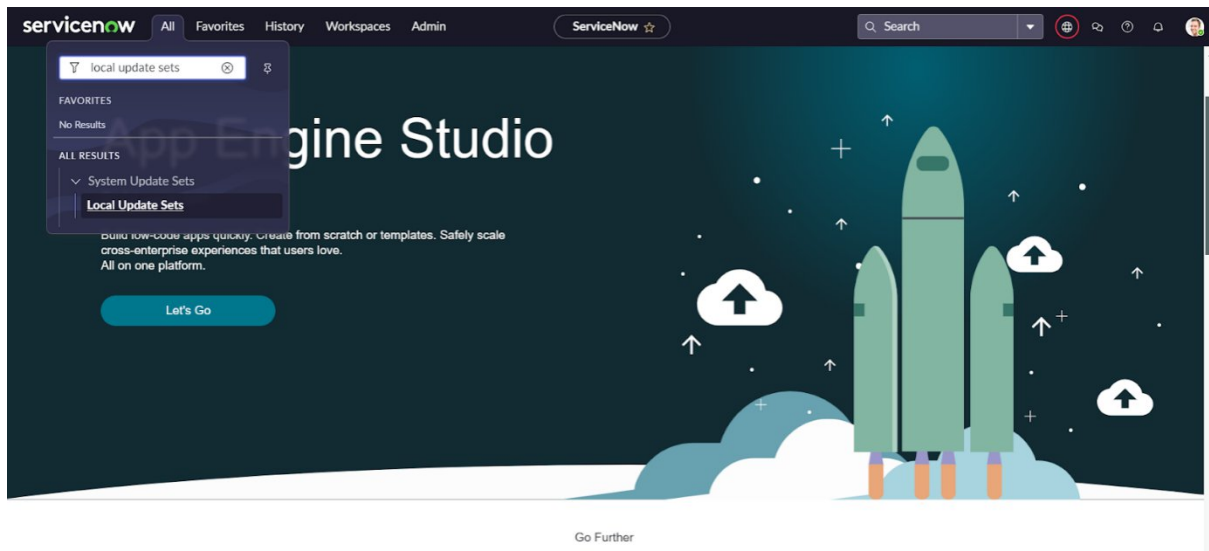
3. Then click on Submit and Make current.



MILESTONE 3: Creation of Table

ACTIVITY 1: Creation of New Update Set

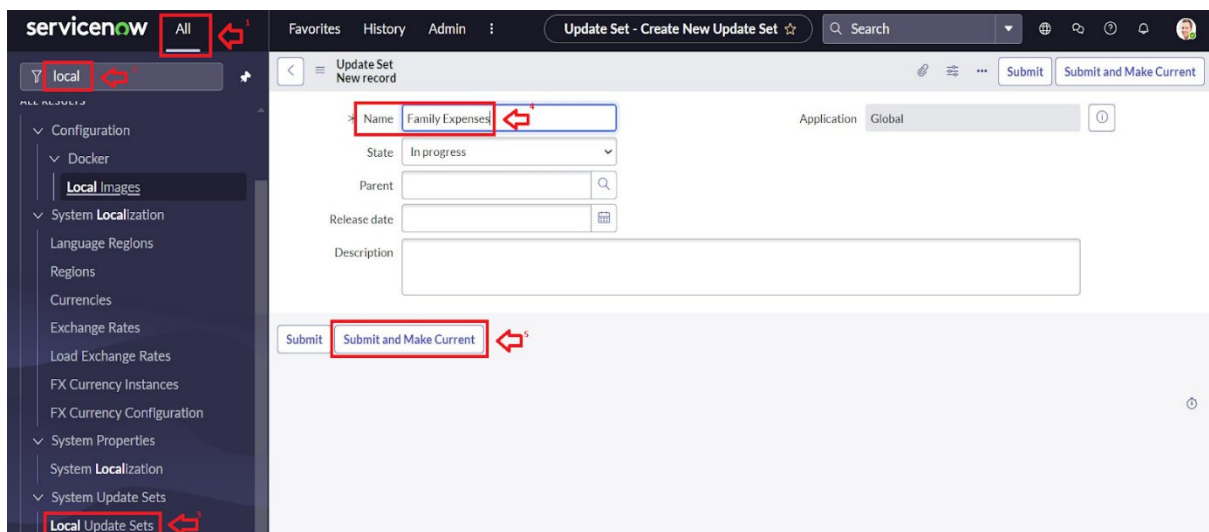
1. Go to All >> In the filter search for Local Update set > click on New.



2. Enter the Details as:

Name: Family Expenses

3. Then click on Submit and Make current.



MILESTONE 4: 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

The screenshot shows a configuration window for a table. At the top, there are two input fields: 'Label' with the value 'Family Expenses' and 'Name' with the value 'u_st_family_expenses'. To the right, there are several settings: 'Application' is set to 'Global', 'Remote Table' is checked, 'Create module' is checked, 'Create mobile module' is checked, 'Add module to menu' is set to '-- Create new --', and 'New menu name' is set to 'Family Expenditure'. Below these settings, there are three tabs: 'Columns', 'Controls', and 'Application Access'. The 'Columns' tab is active, showing a table with three columns: 'Column label', 'Type', and 'Display'. The table has three rows: 'Number' (String), 'Date' (Date), and 'Amount' (Integer). Each row has a red 'X' icon in the first column and a green checkmark in the 'Display' column.

Column label	Type	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 clicks on insert a new row again

4. Give the details as:

Column label: Date

Type: Date

5. Double clicks on insert a new row again

6. Give the details as:

Column label: Amount

Type: Integer

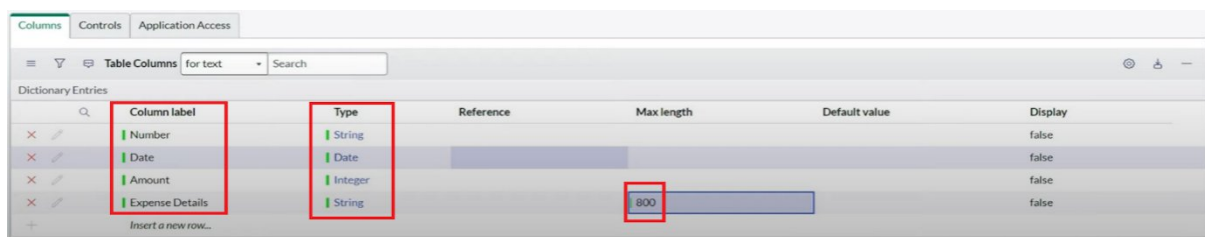
7. Double clicks on insert a new row again

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 | | Update | Delete

* Table: Family Expenses

Prefix: MFE

* Number: 1,000

Application: Global

Number of digits: 7

Update | Delete

9. 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 screenshot shows a form design interface for a table named 'Family Expenses [u_family_expenses]'. The form is configured with a 2-column layout. The first column contains a 'Number' field. The second column contains 'Date' and 'Amount' fields. Each field has a gear icon for configuration. Below this, there is a section header 'Expense Details' with a 1-column layout.

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 5: 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 1

* Name 2

Extends table

Application

Create module ☒

Create mobile module ☒

Application Menu

Add module to menu 3

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

ACTIVIT 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 clicks on insert a new row again

4.Give the details as:

Column label: Date

Type: Date

5.Double clicks 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 clicks 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 clicks 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.

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 design interface for a table named 'Daily Expenses [u_daily_expenses]'. The form is organized into two columns. The first column contains two fields: 'Number' and 'Date'. The second column contains two fields: 'Family Member Name' and 'Expense'. Each field has a gear icon next to it, indicating that it can be configured. Below the two-column section, there is a section for 'Comments' which is currently set to 1 column. The interface also shows a '2 Column' dropdown and some control icons in the top right corner.

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 6: Creation of Relationship

ACTIVITY 1: Creation of Relationship between Family Expenses and Daily Expenses table

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 7: 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 8: 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	Family Expenses BR	← ¹	
Table	Daily Expenses [u.daily_expenses]	← ²	

Application: Global

Active: ☒

Advanced: ☒ ←³

ⓘ

4. In when to run Check Insert and Update

When to run **Errors** Advanced

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

When: before
Order: 100

Filter Conditions: [Add Filter Condition](#) [Add "OR" Clause](#)

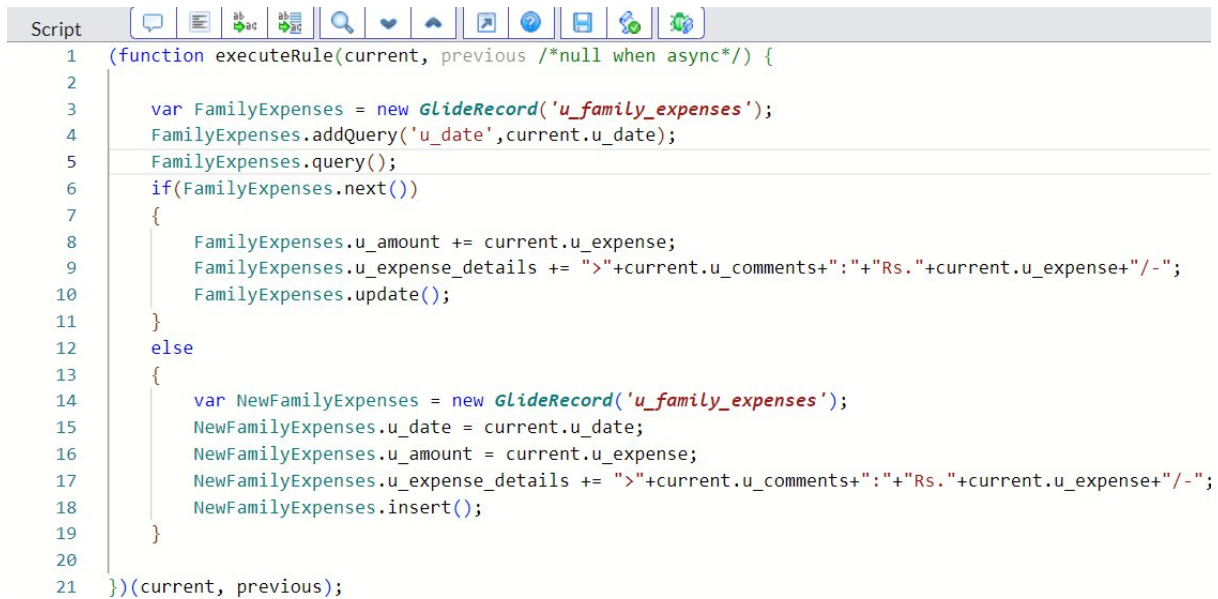
-- choose field -- -- oper -- -- value --

Role conditions [Edit](#)

Insert ☒
Update ☒
Delete ☐
Query ☐

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);

```

MILESTONE 9: 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.

Relationship
Daily Expenses

Name:

Application:

Advanced: ☐

Applies to table: ← 1

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

```
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);
```

 ← 2

Update Delete

 ← 3

Conclusion:

The Family Expense Calculation System on ServiceNow offers an easy and efficient way to manage household finances. With features like expense tracking, budget setting, and reporting, it helps users monitor spending and make informed financial decisions. The system's scalability and user-friendly interface ensure it supports families of all sizes, promoting better financial planning and control.

