



**K.S.K COLLEGE OF ENGINEERING AND TECHNOLOGY**

**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING  
AND TECHNOLOGY**

**CALCULATING FAMILY EXPENSES SERVICENOW**

**TEAM ID : NM2025TMID05880  
TEAM SIZE : 4**

**TEAM LEADER : PRAVEEN K - 821022104037**

**TEAM MEMBER : ESWAR G - 821022104013**

**TEAM MEMBER : MUGESHWARAN D - 821022104031**

**TEAM MEMBER : PRADEEP S - 821022104036**

## **Problem Statement:**

There is currently no efficient system in place to track, categorize, and analyze family expenses in a way that is integrated, automated, and accessible in real-time. This leads to poor budget planning, missed payments, overspending, and a lack of financial visibility.

## **Objective:**

To design and implement a solution using **ServiceNow** that allows family members to:

1. **Input and categorize expenses** easily via a user-friendly interface or mobile portal.
2. **Automate recurring expenses** (e.g., rent, subscriptions).
3. **Track total and category-wise spending** on a monthly/weekly basis.
4. **Set spending limits and receive alerts** when thresholds are crossed.
5. **Generate reports and dashboards** to visualize trends and support better financial decisions.
6. **Share visibility and responsibilities** across family members via roles or user groups.

## **Skills:**

- **ServiceNow App Development** (App Engine Studio)
- **Custom Tables & Data Modeling**
- **Form Design & UI Customization**
- **Flow Designer** (for automation)
- **Business Rules & Scripting**
- **Reporting & Dashboards**
- **User Roles & Access Control (ACLs)**

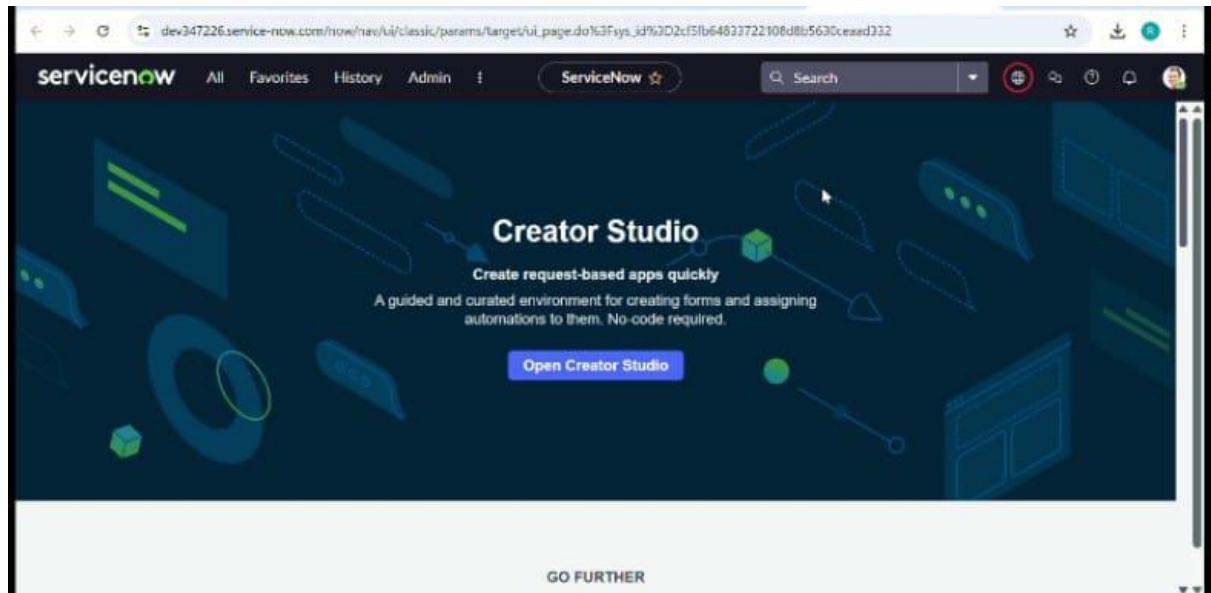
## **TASK INITIATION**

### **Milestone 1 : 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: 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.

A screenshot of the ServiceNow 'Update Set - Create Educational org...' form. The form has fields for Name (Family Expenses), State (In progress), Parent (empty), Release date (empty), and Description (empty). There are buttons for 'Submit' and 'Submit and Make Current'. Above the form, a message says 'Application scope: Global' and 'Update set: Default [Global]'. The top navigation bar shows 'All', 'Favorites', 'History', and 'Workspaces'.

## Milestone 3: Table(Family Expenses)

### 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

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

The screenshot shows the ServiceNow 'Table - New Record' page. At the top, it says 'Table - New Record'. Below that, there's a message: 'ServiceNow recommends creating custom tables in scoped applications. To learn more about creating scoped applications, click [here](#).'. A note below it states: 'A table is a collection of records in the database. Each record corresponds to a row in a table, and each field on a record corresponds to a column on that table. Applications use tables and records to manage data and processes. [More Info](#)'.

The form fields include:

- \* Label: Family Expenses
- \* Name: u\_family\_expenses
- Extends table: (empty)
- Application: Global (with a help icon)
- Create module: checked
- Create mobile module: checked
- Add module to menu: -- Create new --
- New menu name: Family Expenditure (highlighted with a blue border)
- Remote Table: (checkbox)

## Activity 2: Creation of Family Expenses Table

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

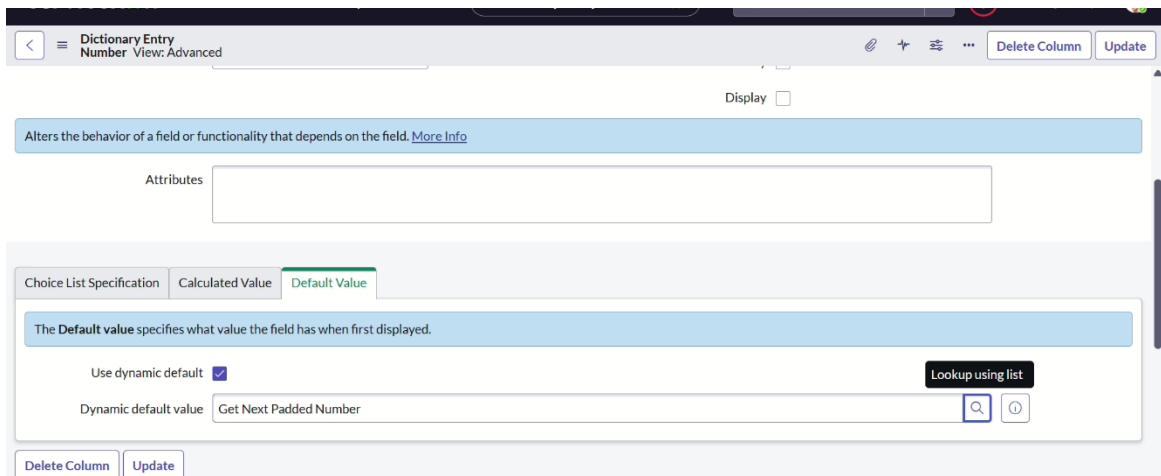
5. Double click on insert a new row again

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.

Table - Family Expenses						
Dictionary Entries		Column label	Type	Reference	Max length	Default value
		Updated	Date/Time	(empty)	40	false
		Created by	String	(empty)	40	false
		Sys ID	Sys ID (GUID)	(empty)	32	false
		Created	Date/Time	(empty)	40	false
		Updated by	String	(empty)	40	false
		Updates	Integer	(empty)	40	false
x	/	Number	String			false
x	/	Date	Date			false
x	/	Amount	Integer			false
x	/	Expense Details	String		800	false

### 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.
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

8. Click on Submit

The screenshot shows a 'Number New record' form for the 'Family Expenses' table. The 'Table' field is set to 'Family Expenses'. The 'Prefix' field contains 'MFE'. The 'Number' field is set to '1,000'. The 'Application' field is set to 'Global'. The 'Number of digits' field is set to '7'. At the bottom left is a 'Submit' button.

## 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.
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 'Form Design' interface for the 'Family Expenses [u\_family\_expenses]' table. The interface is divided into two columns. The first column contains fields for 'Number' and 'Amount'. The second column contains a field for 'Date'. Below these columns is a section labeled 'Expense Details'. On the left side, there is a sidebar with sections for 'Fields' (listing 'Created', 'Created by', 'Updated', 'Updated by', 'Updates'), 'Field Types', 'Filter', and 'Formatters' (listing 'Activities (filtered)' and 'Contextual Search Results'). At the top right, there are 'Undo' and 'Save' buttons.

## Milestone 4: Table(Daily Expenses)

### Activity 1: Creation of Table(Daily Expenses)

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.

The screenshot shows the ServiceNow Table - New Record screen. At the top, there are tabs for All, Favorites, History, Admin, and a search bar. Below the tabs, it says "Table - New Record". On the left, there's a breadcrumb trail "Table > New record". On the right, there are "Submit" and "Cancel" buttons. A yellow banner at the top states: "① ServiceNow recommends creating custom tables in scoped applications. To learn more about creating scoped applications, click [here](#)". Below this, a blue box contains the text: "A table is a collection of records in the database. Each record corresponds to a row in a table, and each field on a record corresponds to a column on that table. Applications use tables and records to manage data and processes." [More Info](#). The main form fields are: "Label" (Daily Expenses), "Name" (u\_daily\_expenses), "Application" (Global), "Create module" (checked), "Create mobile module" (checked), "Add module to menu" (Family Expenditure), and "Remote Table" (unchecked). There is also a "Extends table" field with a search icon.

## 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 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.

The screenshot shows the ServiceNow Developers interface with the 'Table - Daily Expenses' tab selected. In the main pane, the 'Daily Expenses' table is listed under 'Dictionary Entries'. A specific row for the 'Comments' column is selected, showing its details in the bottom pane. The 'Type' is listed as 'String', and the 'Max length' is explicitly set to '800'.

### 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.

The screenshot shows the 'Number' field configuration in Advanced view. The 'Default Value' tab is selected. The 'Use dynamic default' checkbox is checked, and the 'Dynamic default value' dropdown is set to 'Get Next Padded Number'. The 'Update' button is visible at the bottom of the form.

5. Go to All >> In the filter search for Number Maintenance >> select Number Maintenance
6. Click on New.
7. Enter the below Details:

## Table : Daily Expenses

Prefix : DFE

8. Click on Submit.

The screenshot shows the ServiceNow interface for creating a new record. The title bar says "Number - New Record". The form has the following fields:

- \* Table: Daily Expenses
- Prefix: DFE
- \* Number: 1,000
- Application: Global
- Number of digits: 7

At the bottom left is a "Submit" button, and at the bottom right are "Related Links" (Show Counter).

## 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.
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.

The screenshot shows the ServiceNow Form Design tool. The title bar says "Form Design". The main area displays the "Daily Expenses [u\_daily\_expenses]" form with the following fields arranged in two columns:

Number	Family Member Name
Date	Expense

Below this is a single-column section with a "Comments" field. On the left sidebar, under "Fields", are "Created", "Created by", "Updated", "Updated by", and "Updates". Under "Formatters", are "Activities (filtered)", "Contextual Search Results", and "Ratings".

## Milestone 5: Creation of Relationship

### Activity 1: 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.

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);

```

## Milestone 6: Configuring Related List on Family Expenses

### Activity 1: 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 7: Business Rules

## Activity 1: 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

A business 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. [More Info](#)

Name: Family Expenses BR      Application: Global

Table: Daily Expenses [u\_daily\_expenses...]

Active:       Advanced:

When to run:  Actions  Advanced

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

### 4. In when to run Check Insert and Update

Business Rule  
New record

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

When: before      Order: 100

Insert:   
Update:   
Delete:   
Query:

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

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

Role conditions: [Edit](#)

Submit

### 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');
```

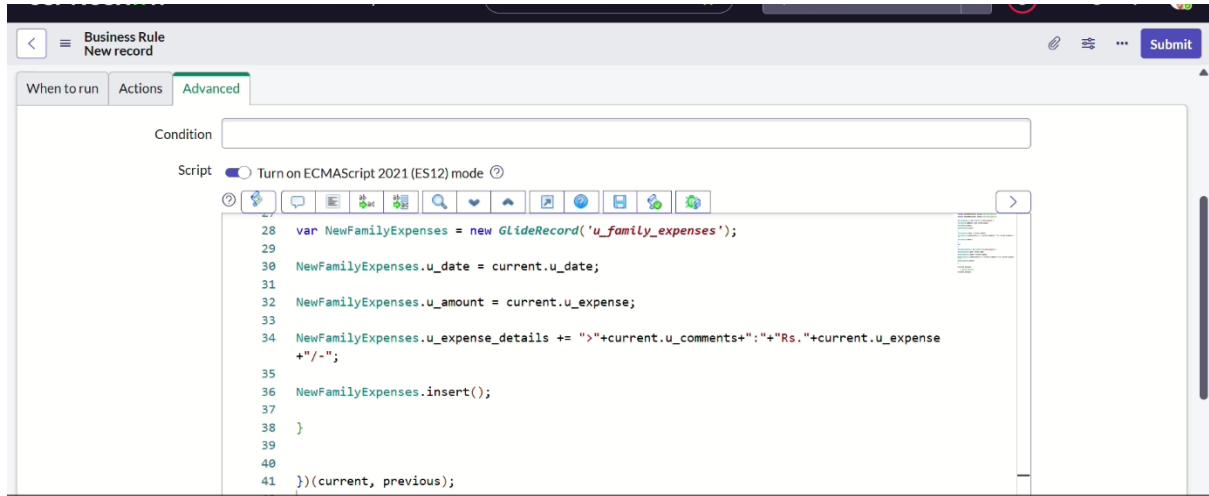
```

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
{
    varNewFamilyExpenses = 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.



## Milestone 8: Relationship

### Activity 1: 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.

```
(functionrefineQuery(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.

