

# **CALCULATING FAMILY EXPENSES USING SERVICE NOW**

**NAAN MUDALVAN PROJECT  
TEAM ID : MN 2025TMID 17840**

**Team Leader : SURENDAR**

**Team Member : GOWTHAM**

**Team Member : JEEVA**

**Team Member : BHARATH**

**ST. JOSEPH 'S COLLEGE (ARTS &SCIENCE )**

## INDEX

1. Introduction      2. Abstract      3. Problem statement      4. Solution      5. Practical use      6. Knowledge gained      7. Milestone 1: Setting up servicenow instance      8. Milestone 2: Creation of new update set      9. Milestone 3: Creation of table family expenses
  
10. Milestone 4: Creation of table daily expenses
  
11. Milestone 5: Creation of relationship
  
12. Milestone 6: Configuring Related list on family expenses
  
13. Milestone 7: Creation of Business rules
  
14. Milestone 8: Configure the relationship
  
15. Conclusion

## INTRODUCTION

Managing family expenses is one of the most important aspects of financial planning, yet it is often overlooked or handled in unstructured ways like handwritten notes or spreadsheets. Families frequently face challenges such as overspending, lack of budget control, and difficulty in analyzing where their money goes. To address these issues, technology can play a vital role by providing structured, automated, and user-friendly solutions.

This project, "Calculating Family Expenses Using ServiceNow," aims to create a digital system that helps track, categorize, and manage family expenses efficiently. ServiceNow, known for its robust workflow automation and application development capabilities, provides an excellent platform for building such a system. By leveraging

ServiceNow, this project transforms expense management into a more streamlined process with features like expense categorization, daily tracking, budget limits, reporting, and automated business rules.

The system will not only simplify financial management for families but also provide real-time insights into spending patterns, allowing better decision-making. With its scalability and flexibility, the solution can be expanded to suit different

family structures and even adapted for small business expense management. Ultimately, the project showcases how a powerful enterprise platform like ServiceNow can be applied beyond IT workflows to solve everyday problems in an innovative and practical way.

## ABSTRACT

Expense management plays a crucial role in maintaining

financial stability within families, yet traditional methods such as manual tracking or spreadsheets often lead to errors, lack of visibility, and poor decision-making. To address this challenge, this project focuses on designing and implementing a Family Expense Management System using ServiceNow. The application leverages ServiceNow's powerful low-code development tables, environment to create structured relationships, and automated workflows for recording and analyzing categorizing expenses. Key features include expenses (such as food, utilities, and transport), maintaining family member details, linking daily transactions, and applying business rules to validate data and automate calculations. By configuring related lists and creating meaningful reports, the system provides users with a clear overview of their financial habits and budget limits. The proposed solution not only simplifies day-to-day expense tracking spending but also provides real-time insights into patterns, enabling families to make smarter financial

## PROBLEM STATEMENT

Managing household expenses is often a difficult and time-consuming task for families. Most families rely on manual methods such as notebooks, receipts, or spreadsheets to track their daily spending. These methods come with several challenges:

Lack of real-time tracking of expenses.

Difficulty in categorizing and consolidating expenses like food, utilities, rent, and transportation.

Limited ability to analyze spending patterns or generate reports.

High chances of errors due to manual data entry.

No automation to alert families about overspending or exceeding budgets.

As a result, families often lose visibility into their financial flow, making it harder to control budgets or make informed financial decisions. This creates the need for a systematic, automated, and user-friendly solution to manage and calculate family expenses efficiently.

## SOLUTION

The proposed solution is to build a Family Expense Management System on the ServiceNow platform. ServiceNow, being a robust low-code/no-code platform, provides all the tools required to create structured applications without needing extensive programming knowledge.

The solution involves:

Creating custom tables to store family member details and daily expenses.

Defining relationships between family members and their respective expenses for easy tracking.

Configuring related lists so that expenses linked to each family member can be viewed in one place.

Implementing business rules to automate calculations, validate entries, and ensure data accuracy.

Generating reports and dashboards to visualize monthly/annual spending and highlight budget deviations.

Using update sets to track and migrate customizations, ensuring proper version control.

## PRACTICAL USE

The Family Expense Management System built on ServiceNow helps families track, categorize, and analyze their expenses in a structured way. It simplifies budgeting, reduces errors from manual tracking, and provides real-time insights into spending patterns. Families can use it to set limits, monitor monthly expenses, and generate reports for better financial decisions. Beyond households, the same system can be adapted for small businesses to manage cash flow and daily transactions effectively, proving the versatility of ServiceNow in solving practical, non-IT problems.

## KNOWLEDGE GAINED

- Learned how to setup and configure a ServiceNow developer instance for building applications.
- Understood the importance of update sets for tracking and migrating customizations.
- Gained practical skills in creating custom tables and defining fields to store structured data.
- Learned how to establish relationships between tables for linked data management.
- Practiced configuring related lists for easier navigation and record visibility.
- Understood how to create and apply business rules for automation and validations.
- Gained insights into data modeling and database concepts within ServiceNow.
- Learned how to generate reports and dashboards for real-time analysis.
- Understood how ServiceNow can be applied to non-IT use cases like family expense tracking.
- Improved overall knowledge of workflow automation and low-code development.



# MILESTONE1:SETTING UP THE SERVICE NOW INSTANCE

GototheofficialServiceNowDeveloperportal:  
<https://developer.servicenow.com> and create a developer account.

Aftersigningin,openthePersonalDeveloperInstancesection from the dashboard.

SelectRequestInstancetogenerateafreshService Now environment for development.

Providethenecessarydetails(likeversionselection)and confirm your request.

Waitfortheconfirmationemailcontainingyourinstance URL and login credentials.

UsethecredentialstologintoyournewlycreatedServiceNow instance.

Onceinside,exploretheinterfaceandbeginworkingon the platform.

# MILSTONE2:CREATION OF NEW UPDATE

The screenshot shows the ServiceNow Developer portal interface. At the top, there's a navigation bar with links like 'MyNow', 'Products', 'Industries', 'Learning', 'Support', 'Partners', 'Company', and 'Developer'. Below the navigation is a search bar and a 'Start building' button. The main content area is titled 'Manage my instance (dev325422)'. It contains several sections: 'Status' (green 'Online'), 'Instance URL' (https://dev325422.service-now.com), 'User name' (admin), 'Current password' (redacted), and 'User role' (Admin). To the right is a 'Useful links' sidebar with links to 'Personal Developer Instance [PDI] Guide', 'PDI FAQs', 'Managing your PDI', and 'Developer advocate blog'. Below the status section is a 'Plugins for your instance (46)' section. It lists several plugins: 'Activate all Software Asset Management Professional plugins' (Not activated), 'Cloud Management' (Not activated), 'Customer Service Virtual Agent Conversations' (Not activated), 'Discovery' (Not activated), 'Domain Support - Domain Extensions Installer' (Not activated), and 'ECC Queue Retry Policy' (Not activated). At the bottom of the page, there's a note to 'Activate Windows' with a link to 'Go to Settings to activate Windows.'

## SET

Log into your ServiceNow instance and go to the Application Navigator.

Search for Update Sets and open Local Update Sets under System Update Sets.

Click on New to create a fresh update set.

Enter the following details:

- o Name: Family Expenses
- o Description: Update set to capture all configurations related to the Family Expense Management project.

The screenshot shows the 'Update Set - Family Expenses' page in ServiceNow. The page has a header with tabs like 'All', 'Favorites', 'History', and 'Admin'. Below the header, there's a search bar and an 'Update' button. The main form contains the following fields:

- \* Name: Family Expenses
- State: In progress
- Parent: (empty)
- Release date: (empty)
- Install date: (empty)
- Installed from: (empty)
- Description: (empty)

On the right side of the form, there are details:

- Application: Global
- Created: 2025-09-02 01:50:30
- Created by: admin
- Merged to: (empty)

At the bottom left, there's an 'Update' button. At the bottom right, there's a 'Related Links' section with links like 'Make This My Current Set', 'Merge With Another Update Set', and 'Scan Update Set'. At the very bottom, there are navigation buttons for 'Customer Updates (27)', 'Update Set Logs', 'Child Update Sets', and 'Install History', along with a search bar and an 'Actions on selected rows...' button.

SavetherecordandmarkitastheCurrentUpdateSet,soevery  
change you make is tracked under this set.

Verifythattheupdatesetisactivebycheckingtheheaderat the top  
of the screen.

Fromthispointforward,allcustomizations(tables,  
relationships, and business rules) will be recorded inside the  
Family Expenses update set.

# MILSTONEr : CREATIONOFTABLEFAMILY EXPENSES

## Activity -CreatingtheFamilyExpensesTable

- In your ServiceNow instance, navigate to All Tables using the filter navigator.
- Click on New to create a new table.
- Fill in the required details:
  - Label : Family Expenses
  - Name : (This will be auto-generated based on the label)
  - New Menu Name : Family Expenditure
- Save the record to create the new table.

The screenshot shows the ServiceNow Table creation interface. At the top, there are fields for 'Label' (Family Expenses) and 'Name' (u\_family\_expenses). Below this, under 'Application', it says 'Global'. There is also a 'Remote Table' button. The main area shows a table titled 'Dictionary Entries' with columns: Column label, Type, Reference, Max length, Default value, and Display. The table contains five rows with the following data:

Column label	Type	Reference	Max length	Default value	Display
Created	Date/Time	(empty)	40		false
Sys ID	Sys ID (GUID)	(empty)	32		false
Date	Date	(empty)	40		false
Expense Details	String (Full UTF-8)	(empty)	255		false
Number	String	(empty)	40	javascript:getNextObjNumberPadded();	false

# MILSTONE3:CREATIONOFTABLEFAMILY EXPENSES

## Activity2-AddingColumnstotheFamilyExpensesTable

Duration:1Hour

SkillTags:TableConfiguration,DataModeling,ServiceNowBasics

OpenthenewlycreatedFamilyExpensestable.

To add columns, double-click near the existing columns to insert a new row.

Enter the following details one by one:

1. ColumnLabel: Number o

Type:String

2. ColumnLabel: Date o

Type:Date

3. ColumnLabel: Amount o

Type:Integer

4. ColumnLabel: ExpenseDeta

ils o Type:String o

MaxLength:800

The screenshot shows the ServiceNow Table - Family Expenses page. The table has the following columns and data:

Column Label	Type	Default Value	Length	Notes
Sys ID	Sys ID (GUID)	(empty)	32	false
Date	Date	(empty)	40	false
Expense Details	String (Full UTF-8)	(empty)	255	false
Number	String	(empty)	40	javascript:getNextObjNumberPadded();
Updated by	String	(empty)	40	false
Updates	Integer	(empty)	40	false
Updated	Date/Time	(empty)	40	false
Created by	String	(empty)	40	false
Amount	Integer	(empty)	40	false

At the bottom of the table, there is a link "Insert a new row..." and buttons for Delete, Update, and Delete All Records. Below the table, there is a section titled "Related Links" with links to Form Builder, Design Form, Layout Form, Layout List, Show Form, Show List, and Show Schema Map.

# MILSTONE3:CREATIONOFTABLEFAMILY EXPENSES

Activity—Making the Number Field an Auto-Number  
Open the Family Expenses table.

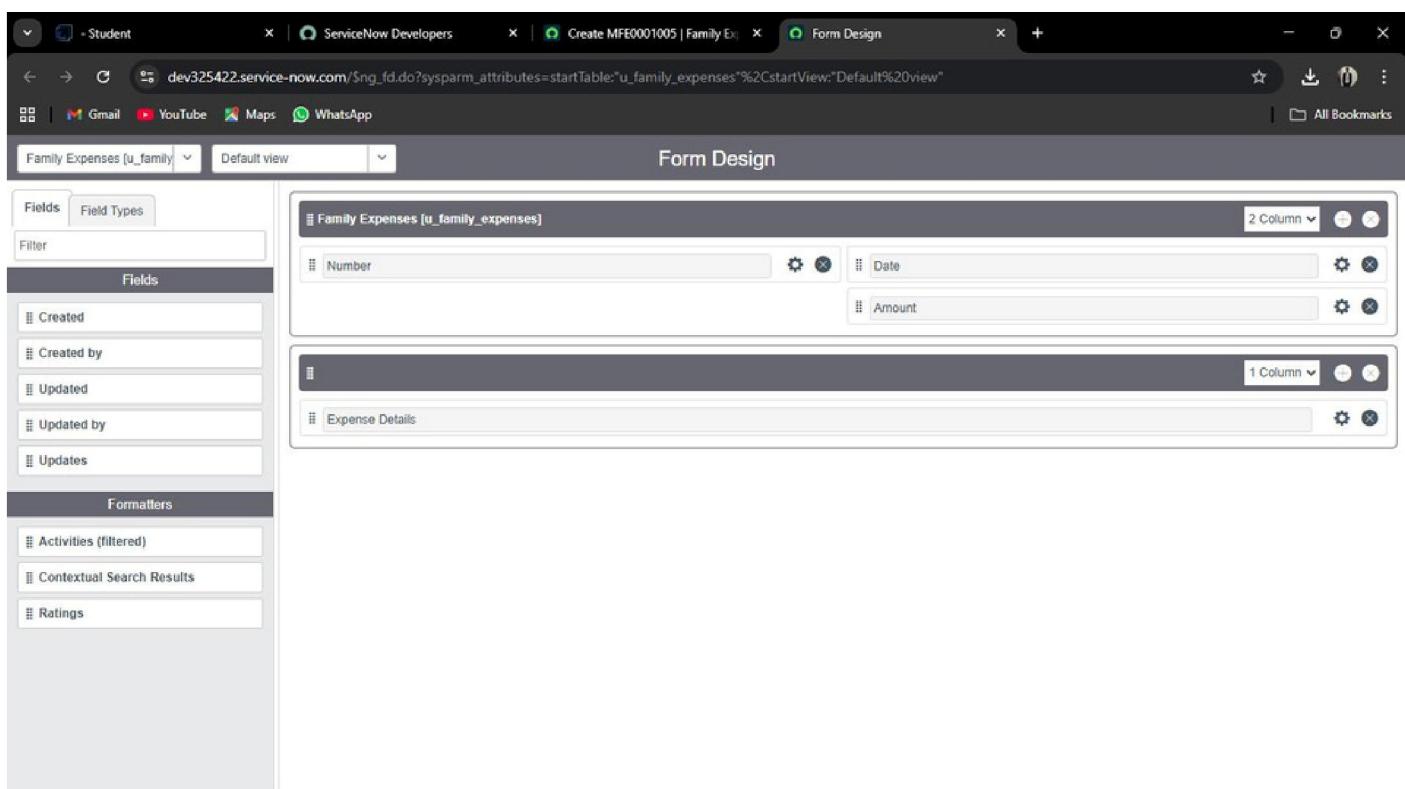
- Locate the Number field / column and double-click to open its properties.
- Scroll down and switch to the Advanced view.
- In the Default Values section:
  - Check the box for Use Dynamic Default.
  - Set the Dynamic Default Value to Get Next Padded Number.
- Click Update to save the changes.

The screenshot shows the ServiceNow Dictionary Entry - Number page. The browser title is "Number | Dictionary Entry | Sen". The URL is "dev325422.service-now.com/now/nav/ui/classic/params/target/sys\_dictionary.do%3Fsys\_id%3DS51063afc93fb62109d16393efaba102a%26sysparm\_record\_list%3Dname...". The page header includes "servicenow" and navigation links for All, Favorites, History, Admin, and a search bar. The main content area is titled "Dictionary Entry - Number". A sub-header says "Dictionary Entry Number View: Advanced". Below this, a message states "Alters the behavior of a field or functionality that depends on the field. [More Info](#)". A "Display" checkbox is present. A "Attributes" section has an empty input field. A tabs section at the bottom includes "Choice List Specification", "Calculated Value", and "Default Value", with "Default Value" being the active tab. A note below says "The Default value specifies what value the field has when first displayed." It shows a checked "Use dynamic default" checkbox and a "Dynamic default value" input field containing "Get Next Padded Number". Action buttons "Delete Column" and "Update" are at the bottom. Related links include "Related Links", "Show Table", and "Run Point Scan". At the very bottom, there are tabs for "Access Control", "Choices", "Attribution", and "Labels (4)".

# MILSTONE3:CREATIONOFTABLEFAMILY EXPENSES

## Activity:-ConfiguringtheForm

- Navigate to All < in the filter, search for Family Expenses.
- Open the Family Expenses table.
- Click on New to create a new form entry.
- On the form header, right-click and select:
  - o Configure Form Design.
- In the Form Designer, use drag-and-drop to:
  - o Rearrange fields.
  - o Group related fields together.
  - o Add sections if required for better clarity.
- Save the customized form layout.



---

## MILSTONE4:CREATIONOFTABLEDAILY EXPENSES

# MILSTONE4:CREATIONOFTABLEDAILY EXPENSES

## Activity\-\CreatingTheDailyExpensesTable

- Navigate to All Tables using the filter navigator.
- Click on New to create a new table.
- Fill in the required details:
  - Label: Daily Expenses
  - Name: (Auto-populated by the system)
  - Add Module to Menu: Family Expenditure
  - Go to the form header, right-click, and select Save.

The screenshot shows the ServiceNow Table - Daily Expenses page. At the top, there's a navigation bar with tabs like Student, ServiceNow Developers, Daily Expenses | Table | Service, and Form Design. Below the navigation is a toolbar with icons for search, refresh, and various actions. The main area has a title 'Table - Daily Expenses' and a sub-header 'Table'. A message box says: '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.' Below this, there are fields for 'Label' (set to 'Daily Expenses') and 'Name' (set to 'u\_daily\_expenses'). There are also dropdowns for 'Application' (set to 'Global') and 'Remote Table'. At the bottom, there's a table titled 'Dictionary Entries' with columns for 'Column label', 'Type', 'Reference', 'Max length', 'Default value', and 'Display'. The table contains five rows with columns labeled: Family Members Name, String, (empty), 800, false; Sys ID, Sys ID (GUID), (empty), 32, false; Created, Date/Time, (empty), 40, false; Updated by, String, (empty), 40, false; and Updates, Integer, (empty), 40, false.

# MILSTONE4:CREATIONOFTABLEDAILY EXPENSES

## Activity2-CreatingColumns(Fields)

Open the Daily Expenses table.

Near Columns, double-click to insert a new row and add the following fields:

1. ColumnLabel: Number o

Type: String

2. ColumnLabel: Date o

Type: Date

3. ColumnLabel: Expense o

Type: Integer

4. ColumnLabel: Family Member Name o Type: Reference

o MaxLength: 800

The screenshot shows the ServiceNow Form Design interface for the 'Table - Daily Expenses' table. The dictionary entries are listed in a grid:

Column label	Type	Reference	Max length	Default value	Display
Family Members Name	String	(empty)	800		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
Number	String	(empty)	40	javascript:getNextObjNumberPadded();	false
Updated	Date/Time	(empty)	40		false
Date	Date	(empty)	40		false
Comments	String	(empty)	40		false
Expense	Integer	(empty)	40		false

At the bottom of the table grid, there is a link 'Insert a new row...'. Below the table, there are buttons for 'Delete', 'Update', and 'Delete All Records'. At the very bottom, there is a section titled 'Related Links' with links to 'Form Builder', 'Design Form', and 'Layout Form'.

# MILSTONE4:CREATIONOFTABLEDAILY EXPENSES

## 5. ColumnLabel:Comments o

Type:String o MaxLength:

800

## Activity3-MakingNumberFieldanAuto-Number

Open the Daily Expenses table.

Locate the Number field/column and double-click to open its properties.

Scroll down and switch to the Advanced View.

In the Default Values section:

- o Check the box for Use Dynamic Default.
- o Set the Dynamic Default Value to Get Next Padded Number.

Click Update to save changes.

## Configuring Number Maintenance:

Navigate to All > Number Maintenance.

Click on New.

The screenshot shows the ServiceNow Dictionary Entry - Number form. 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'. Other tabs like 'Calculated Value' and 'Attributes' are visible at the top. The form also includes sections for 'Choice List Specification', 'Related Links', and navigation buttons like 'Delete Column' and 'Update'.

---

## MILSTONE4:CREATIONOFTABLEDAILY EXPENSES

Enter the details as follows:

- o Table: FamilyExpenses o Prefix: MFE Click on Submit.

Student | ServiceNow Developers | New Record | Number | ServiceNow | Form Design

dev325422.service-now.com/nov/nav/ui/classic/params/target/sys\_number.do?sys\_id=1%26sys\_is\_list=1%26sys\_target=true%26sys\_number=26sys\_parm\_checked\_it...

Gmail YouTube Maps WhatsApp All Bookmarks

servicenow All Favorites History Workspaces : Number - New Record ☆ Search

Number New record

Table Daily Expenses

Prefix DFE

Number 1,000

Application Global

Number of digits 7

Submit

Related Links

Show Counter

①

The screenshot shows a ServiceNow 'Number - New Record' form. The 'Table' field is set to 'Daily Expenses'. The 'Prefix' is 'DFE'. The 'Number' field contains '1,000'. The 'Application' is 'Global'. The 'Number of digits' is set to '7'. A 'Submit' button is located at the bottom left. Below the form, there's a 'Related Links' section with a 'Show Counter' link.

# MILSTONE4:CREATIONOFTABLEDAILY EXPENSES

## Activity4-ConfigureTheForm

Navigate to All > Daily Expenses using the filter.

Open the Daily Expenses table.

Click on New to create a new form entry.

On the form header, right-click, then select:

Configure > Form Design.

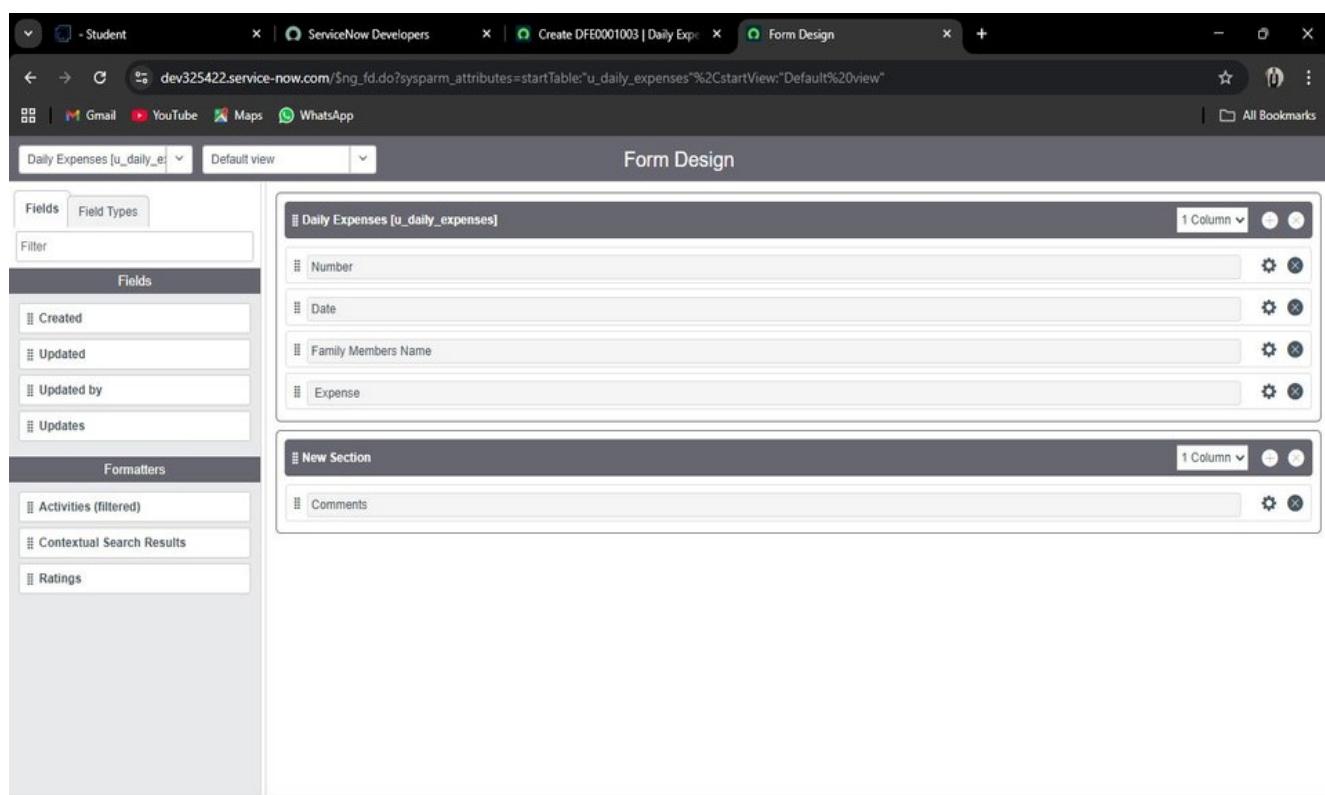
In the Form Designer, drag and drop fields to customize the form layout as per requirement. Apply the following configurations:

Number Field → Set as Read-Only by clicking the gear icon and checking Read-Only.

Date Field → Set as Mandatory by clicking the gear icon and checking Mandatory.

Family Member Name Field → Set as Mandatory using the same method.

Click Save to apply the changes.



# MILSTONE5:CREATIONOFRELATIONSHIP BETWEEN FAMILYEXPENSESAND DAILY EXPENSES TABLES

NavigatetoAll>Relationshipsusingthefilternavigator.  
ClickonNewtocreate anew relationship. Fillinthedetailsasfollows:  
Name:DailyExpenses  
AppliestoTable:FamilyExpenses  
RelatedListTable:DailyExpenses ClickSave.

The screenshot shows the ServiceNow Developers interface with the following details:

- Page Title:** Daily Expenses | Relationship
- URL:** dev325422.service-now.com/nav/uiclassic/params/target/sys\_relationship.do?sys\_id=3D1d15a00d9337a2109d16393efaba109%26sysparm\_record\_target=3Dsys\_re...
- Relationship Name:** Daily Expenses
- Application:** Global
- Table:** Family Expenses [u\_family\_expenses]
- Query:** Daily Expenses [u\_daily\_expenses]
- Script:** 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](#).
- ECMAScript Mode:** Turn on ECMAScript 2021 (ES12) mode (checkbox)
- Code Editor:** A code editor window containing the following JavaScript code:

```
1 (function refineQuery(current, parent) {  
2  
3  
4 // Add your code here, such as current.addQuery(field, value);  
5  
6 current.addQuery('u_date',parent.u_date);  
7  
8 current.query();  
9  
10 }) (current, parent);  
11  
12  
13
```

## MILSTONE6:CONFIGURINGRELATEDLIST ON FAMILY EXPENSES

Navigate to All > Family Expenses using the filter.

Open the Family Expenses table.

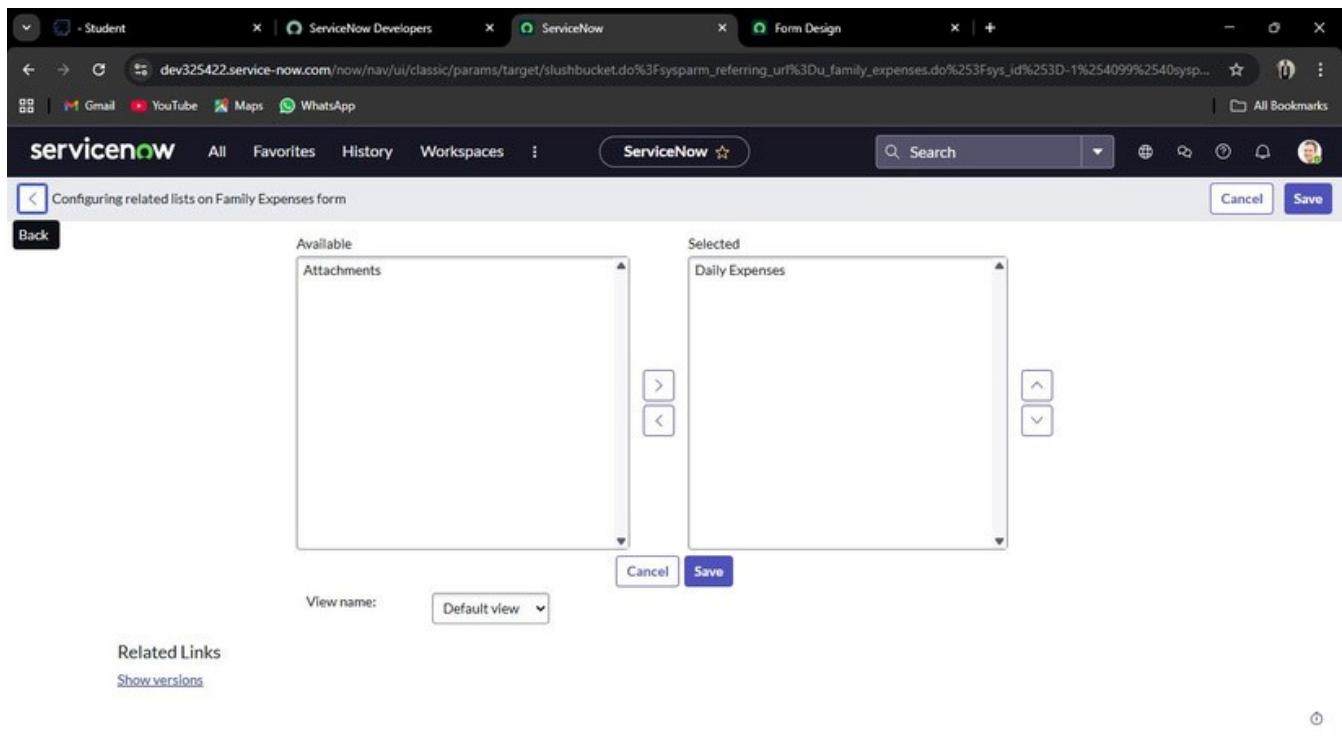
Click on New to open the form view.

On the form header, right-click, then select:

Configure > Related Lists.

From the available options, add Daily Expenses to the Selected Area.

Click Save to apply the changes.





```

        NewFamilyExpenses.insert();

    }

})(current,previous);

```

Goto the form header, right-click, then select Save.

The screenshot shows the ServiceNow Business Rule configuration interface for a rule named "Family Expenses BR". The rule is set to run on the "Daily Expenses [u\_daily\_expenses]" table. It is configured to be global and active. The "Advanced" tab is selected, showing a script editor with the following code:

```

(function executeRule(current, previous /*null when async*/) {
    var FamilyExpenses = new GlideRecord('u_family_expenses');
    FamilyExpenses.addQuery('u_date', current.u_date);
}

```



## CONCLUSION

The Family Expenses Management System built on ServiceNow demonstrates how the platform can be leveraged beyond IT workflows to solve real-world problems. By systematically creating tables, relationships, forms, and business rules, the project enables seamless tracking of both daily expenses and family-level expenses in an automated manner.

The use of auto-numbering, mandatory fields, related lists, and business rules ensures data integrity, consistency, and accuracy. The relationship configuration further enhances visibility by linking daily records to family-level summaries, providing a clear financial overview.

Through this project, we learned how to apply ServiceNow features such as table creation, form design, field configuration, scripting, and automation to build a complete application. More importantly, it highlights how low-code/no-code platforms like ServiceNow can be extended into personal finance, household management, and non-IT use cases.

Overall, the project provides a practical, scalable, and user-friendly solution for managing family expenses efficiently while also strengthening skills in ServiceNow application development.