



NM1051-SERVICENOW ADMINISTRATOR

CALCULATING FAMILY EXPENSES USING SERVICENOW

A PROJECT REPORT

*Submitted By*

<b>A Mohamed Fatheen</b>	<b>-</b>	<b>961322104028</b>
<b>R Durga</b>	<b>-</b>	<b>961322104014</b>
<b>A Gayathri</b>	<b>-</b>	<b>961322104015</b>
<b>R.H Anto Jefrin</b>	<b>-</b>	<b>961322104008</b>

BACHELOR OF ENGINEERING

IN SEVENTH SEMESTER

M.E.T ENGINEERING COLLEGE , AARALVAAIMOZHI- 629301

ANNA UNIVERSITY: CHENNAI 600025 /DECEMBER -20

## ACKNOWLEDGEMENT

I am deeply grateful to express my sincere thanks to M.E.T Engineering college, for providing me the opportunity to undertake this project **titled** "ServiceNow Project Management and Role-Based Access Control System."

I would like to extend my heartfelt gratitude to our Principal, and Head of the Department, for their continuous encouragement, support, and for providing all the necessary facilities to carry out this project successfully.

My sincere thanks to my project guide, for their valuable guidance, patience, and constant motivation throughout the project. Their expert advice and constructive feedback have been instrumental in shaping this work.

I also extend my appreciation to all the faculty members **of the** Department of Computer Science and Engineering for their support and valuable suggestions during the various stages of this project.

A special thanks to my friends and classmates for their continuous encouragement, valuable insights, and cooperation throughout the completion of this project.

Finally, I express my deepest gratitude to my family members for their unwavering love, moral support, and understanding during the entire duration of this work.

<b>S.no</b>	<b>Content</b>	<b>Page no</b>
<b>1.</b>	Introduction	<b>3</b>
<b>2.</b>	Objectives	<b>3</b>
<b>3.</b>	Methodology	<b>4</b>
<b>4.</b>	System Requirements	<b>4</b>
<b>5.</b>	System Design	<b>5</b>
<b>5.1.</b>	System Architecture	<b>5</b>
<b>5.2.</b>	Database Design	<b>6</b>
<b>6.</b>	Workflow	<b>6</b>
<b>7.</b>	Data Flow Diagram (DFD)	<b>7</b>
<b>8.</b>	Application Implementation	<b>10</b>
<b>9.</b>	Implementation	<b>14</b>
<b>10.</b>	Flow chart	<b>24</b>
<b>11.</b>	Testing	<b>25</b>
<b>12.</b>	Conclusion	<b>26</b>
<b>13</b>	Future Scope	<b>27</b>

# Family Expense Calculation System using ServiceNow

## Abstract

This project aims to develop a \*\*Family Expense Calculation and Management System on the ServiceNow platform. The system provides a centralized solution to record, categorize, and analyze family expenses efficiently. It helps family members manage their monthly budgets, visualize spending patterns, and make data-driven financial decisions. Leveraging ServiceNow's powerful reporting and automation capabilities, this project enables seamless expense tracking, dynamic report generation, and scheduled notifications for financial management. The system is scalable, user-friendly, and accessible through the ServiceNow interface and mobile app.

## 1. Introduction

Managing household expenses effectively is vital for financial stability. Traditional methods such as spreadsheets or manual records are time-consuming and prone to errors. ServiceNow, known for its enterprise workflow and reporting capabilities, provides an ideal platform to automate and digitize this process.

The Family Expense Calculation System is designed to record, categorize, and calculate family expenses while providing analytical reports and insights. Each family member can log expenses under different categories (e.g., food, utilities, education), and the system will automatically generate reports showing total spending, category-wise breakdowns, and monthly trends.

## 2. Objectives

The primary objectives of this project are:

1. To design and develop a digital expense tracking system using ServiceNow.
2. To create automated reports for analyzing family expenditure.
3. To categorize expenses by type, date, and family member.
4. To provide a user-friendly dashboard for budget tracking.
5. To generate alerts or notifications when expenses exceed predefined limits.
6. To enhance financial transparency and planning among family members.

### 3. Methodology

The project follows a System Development Life Cycle (SDLC) approach using the Agile methodology to ensure iterative development and user feedback incorporation.

#### Phases:

1. Requirement Analysis – Gather and analyze user needs (expense tracking, reports, etc.).
2. System Design – Design the database schema and UI forms in ServiceNow.
3. Development – Implement tables, forms, workflows, and reports in ServiceNow.
4. Testing – Conduct unit and system testing to validate data accuracy and performance.
5. Deployment – Deploy the solution in a ServiceNow instance for family use.
6. Maintenance – Regular updates and improvements based on feedback.

### 4. System Requirements

#### 4.1 Hardware Requirements

Component	Minimum Specification
Processor	Dual-Core 2.0 GHz or higher
RAM	4 GB or higher
Storage	20 GB
Internet	Broadband connection

## 4.2 Software Requirements

Software	Description
ServiceNow Platform	Cloud-based development and reporting platform
Web Browser	Chrome, Edge, Firefox
Operating System	Windows / macOS / Linux
Scripting	JavaScript (for ServiceNow scripting)

## 5. System Design

### 5.1 System Architecture

The system consists of:

User Interface Layer – Family members interact through ServiceNow forms.

Application Layer\*\* – Business rules and workflows handle logic for expense entry, validation, and calculation.

Database Layer\*\* – Stores expense details and report data.

### 5.2 Database Design (Table Structure)

Field Name	Type	Description
Expense ID	Auto Number	Unique identifier for each expense
Date	Date	Date of expense
Category	Choice	Food, Housing, Utilities, etc.
Amount	Currency	Expense amount
Description	String	Details of the expense

Expense Type	Choice	Fixed / Variable
Family Member	Reference (User	Person who made the expense
Payment Method	Choice	Cash / Credit / Bank Transfer

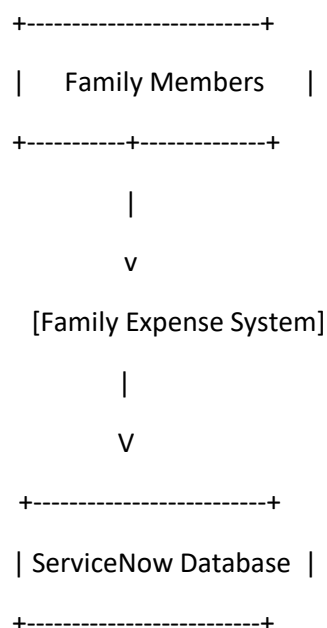
## 6. Workflow

### Workflow Steps:

1. Expense Submission: Family member logs into ServiceNow and submits an expense entry.
2. Data Storage: Expense data is saved in the u\_family\_expenses table.
3. Validation: Business rules ensure valid data entry (e.g., no negative amounts).
4. Calculation: The system aggregates data to calculate monthly totals and category-wise spending.
5. Reporting: Automated reports generate charts, summaries, and budget comparisons.
6. Notifications If spending exceeds the budget, a notification is sent via email or ServiceNow alert.

## 7. Data Flow Diagram (DFD)

### Level 0 DFD (Context Diagram)



## Level 1 DFD

Family Member --> (Enter Expense) --> [Expense Form]

[Expense Form] --> (Store Data) --> [Expense Table]

[Expense Table] --> (Generate) --> [Expense Reports]

[Expense Reports] --> (View / Export) --> Family Member

## 8. Application Implementation

Modules Implemented in ServiceNow:

1. Custom Table: `u\_family\_expenses` for storing expense records.
2. Form Design: A form for users to input expense data.
3. Business Rules: Validate entries and trigger alerts when the budget exceeds.
4. Reports:
  - \* Monthly Expense Summary Report
  - \* Category-wise Expense Chart
  - \* Family Member Spending Comparison
5. Dashboard: Centralized visualization using bar and pie charts.
6. Scheduled Jobs: Automatically generate monthly reports and email summaries.



## 9. Implementation

### **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

## MILESTONE 3: Creation of Table

### ACTIVITY 1: Creation of New Update Set

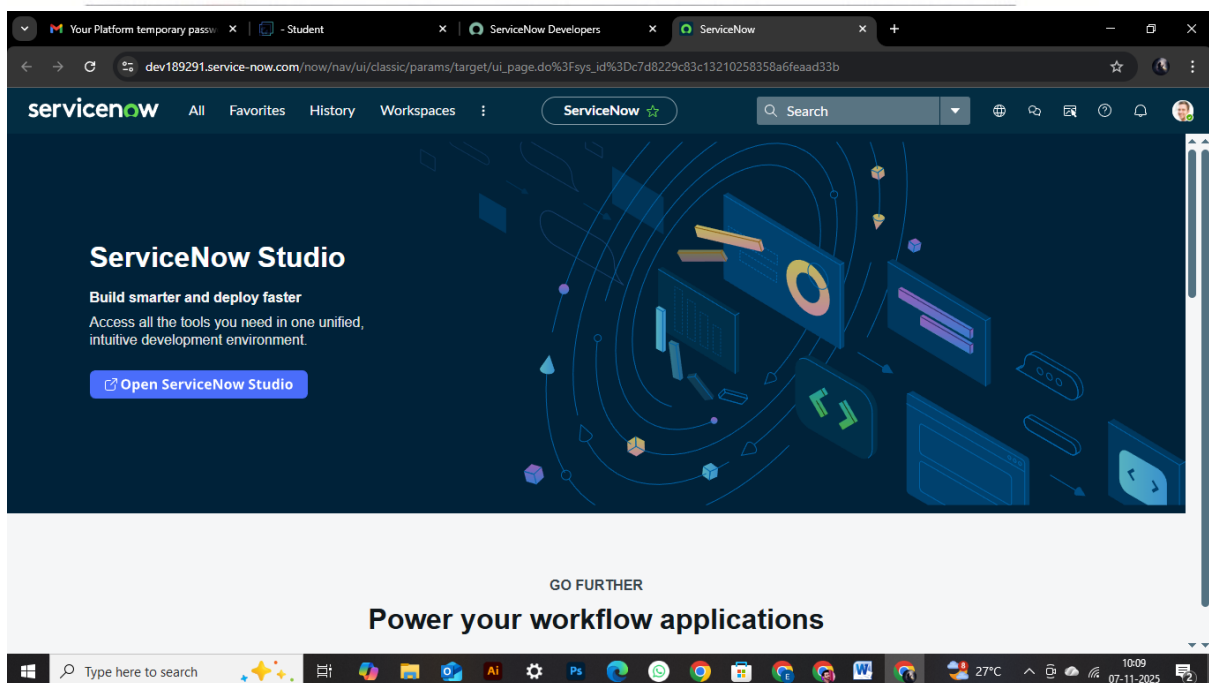
1. Go to All >> In the filter search for Local Update set > click on New.



2. Enter the Details as:

Name: Family Expenses

3. Then click on Submit and Make current.



## 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.

Update Set  
Family Expenses

\* Name: Family Expenses

State: In progress

Parent:

Release date:

Install date:

Installed from:

Description:

Application: Global

Created: 2025-11-06 20:44:00

Created by: admin

Merged to:

Update

Related Links

[Merge With Another Update Set](#)

[Scan Update Set](#)

Customer Updates

Created

Search

Actions on selected rows...

Update set = Family Expenses

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

## 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

Column label	Type	Reference	Max length	Default value	Display
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
Updated	Date/Time	(empty)	40		false
Number	String				false
Date	Date				false
Amount	Integer				false
Expense Details	String		800		false

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

### ACTIVITY 2: Creation of Columns (Fields)

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

The screenshot shows the ServiceNow 'Table - New Record' form for 'Family Expenses'. The form is divided into two main sections: 'Table' and 'Columns'.

**Table Section:**

- \* Label:** Family Expenses
- \* Name:** u\_auto\_populated
- Extends table:** (empty)
- Application:** Global
- Create module:** ☒
- Create mobile module:** ☒
- Add module to menu:** -- Create new --
- New menu name:** (empty)
- Remote Table:** ☐

**Columns Section:**

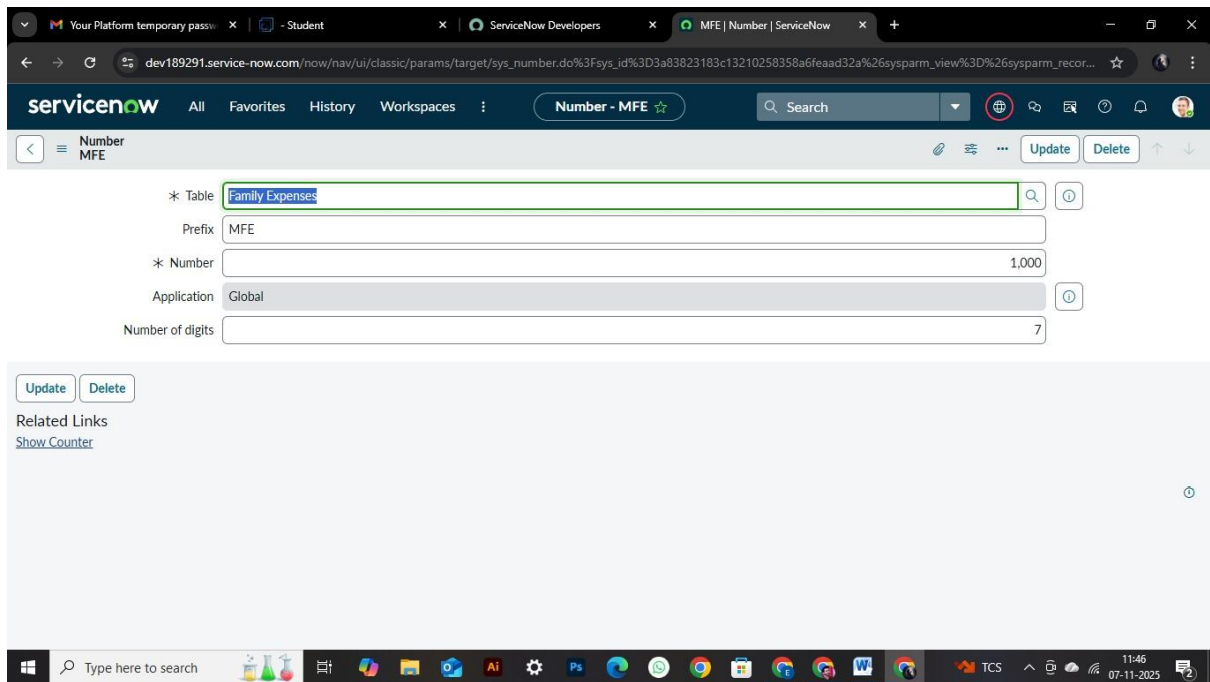
The 'Columns' section is currently active, showing a table of 'Dictionary Entries'.

Column label	Type	Reference	Max length	Default value	Display
Created by	String	(empty)	40		false
Sys ID	Sys ID (GUID)	(empty)	32		false
Created	Date/Time	(empty)	40		false

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 has three fields: 'Number', 'Date', and 'Amount'. Each field has a gear icon for configuration. Below the form is a section titled 'Expense Details' with a gear icon.

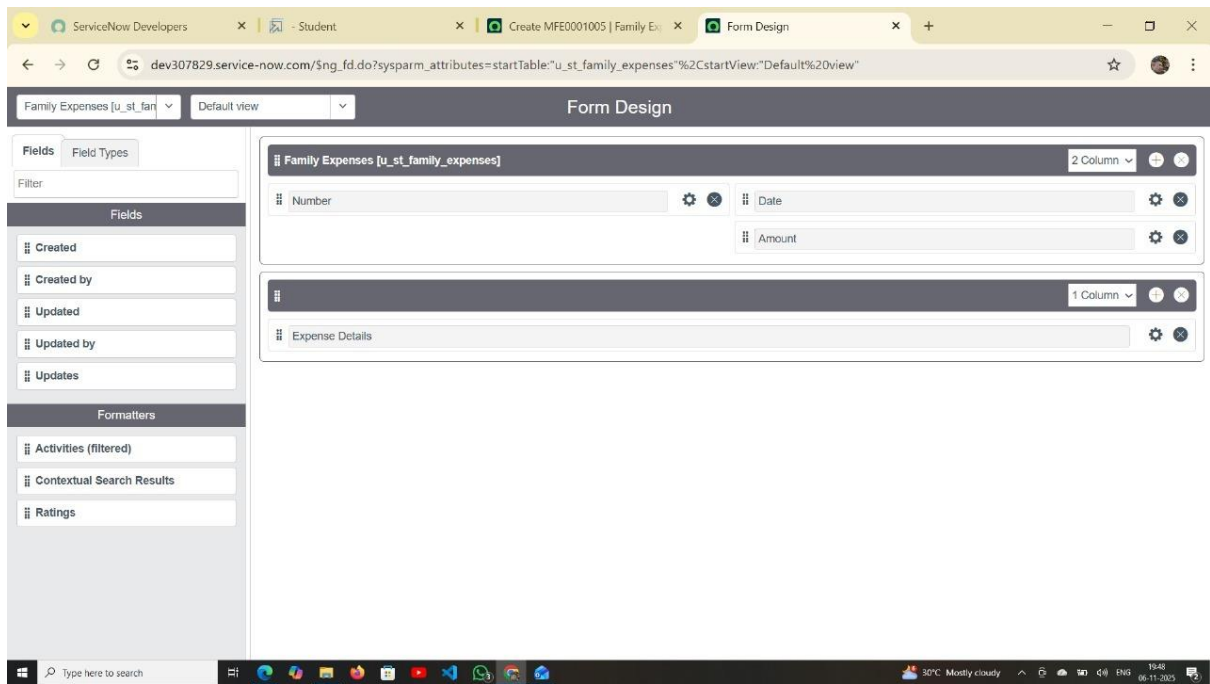
5. Make Number Read-Only Field by clicking on the gear icon and checking Read-Only
6. Make Date, Amount Mandatory Field by clicking on the gear icon and checking Mandatory
7. Click on Save.

## **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:







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:



ServiceNow interface showing the 'Number - New Record' form and the 'Form Design' view.

**Number - New Record Form:**

- Table: Family Expenses
- Prefix: MFE
- Number: 1,000
- Application: Global
- Number of digits: 7

**Form Design View:**

Form Design for 'Daily Expenses [u\_auto\_populated2]'.

Fields:

- Created
- Created by
- Updated
- Updated by
- Updates

Formatters:

- Activities (filtered)
- Contextual Search Results
- Ratings

Form Structure:

- 2 Column layout:
  - Number
  - Date
- 1 Column layout:
  - comments
  - Expense

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

ServiceNow interface showing the configuration of a relationship between 'Daily Expenses' and 'Family Expenses'.

**Top Screenshot (Daily Expenses Relationship Configuration):**

- Name:** Daily Expenses
- Application:** Global
- Applies to table:** Family Expenses [u\_auto\_populate...]
- Queries from table:** Daily Expenses [u\_auto\_populated...]
- Query with:** Turn on ECMAScript 2021 (ES12) mode
- Script:**

```
1 (function refineQuery(current, parent) {  
5 })(current, parent);
```
- Buttons:** Run Query Diagnostics, Update, Delete

**Bottom Screenshot (Configuring related lists on Family Expenses form):**

- Available:** Attachments
- Selected:** Daily Expenses
- Buttons:** Cancel, Save
- View name:** Default view
- Related Links:** [Show versions](#), [Related list performance diagnostics](#)



ServiceNow Business Rule - Family Expenses BR

Condition

Script ☒ Turn on ECMAScript 2021 (ES12) mode

```
1 (function executeRule(current, previous /*null when async*/) {
10   if(FamilyExpenses.next())
11   {
12     {
13       FamilyExpenses.u_amount += current.u_expense;
14       FamilyExpenses.u_expense_details += ">" + current.u_comments + ":" + "Rs." + current.u_expense + "/-";
15       FamilyExpenses.update();
16     }
17   }
18   else
19   {
20     {
21       var NewFamilyExpenses = new GlideRecord('u_family_expenses');
22       NewFamilyExpenses.u_date = current.u_date;
23     }
24   }
25 }
```

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

Table: Daily Expenses [u\_auto\_populated...]

Application: Global

Active: ☒

Advanced: ☒

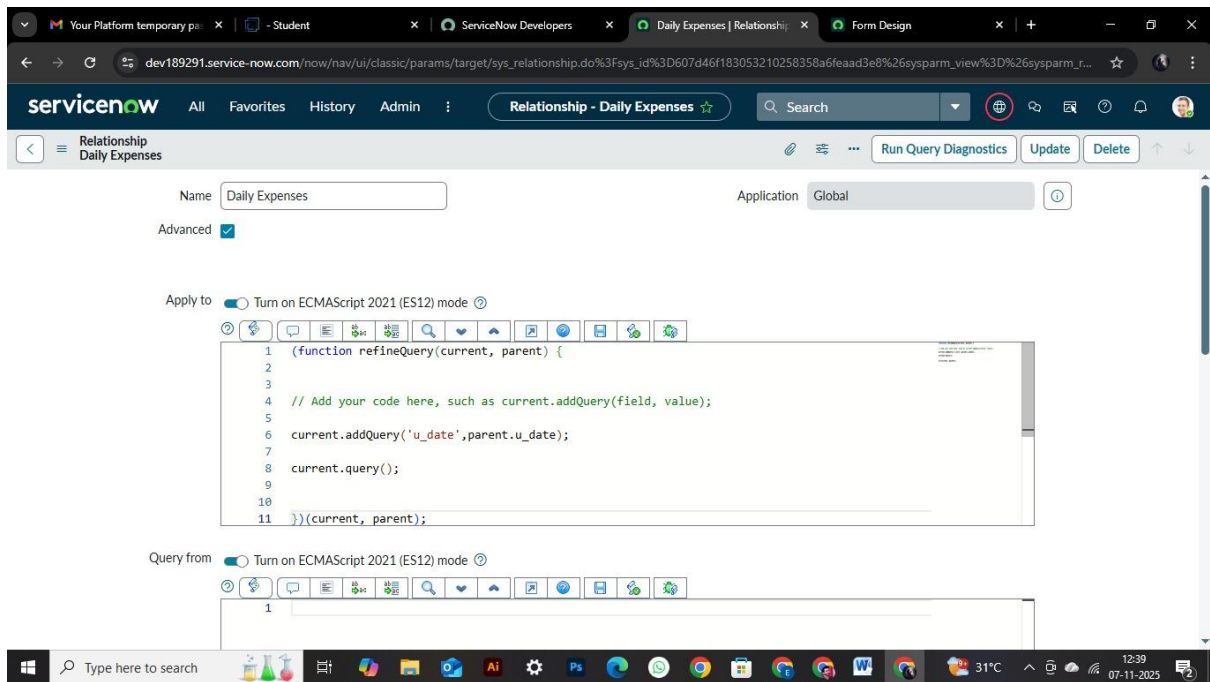
When to run: Actions Advanced

Condition

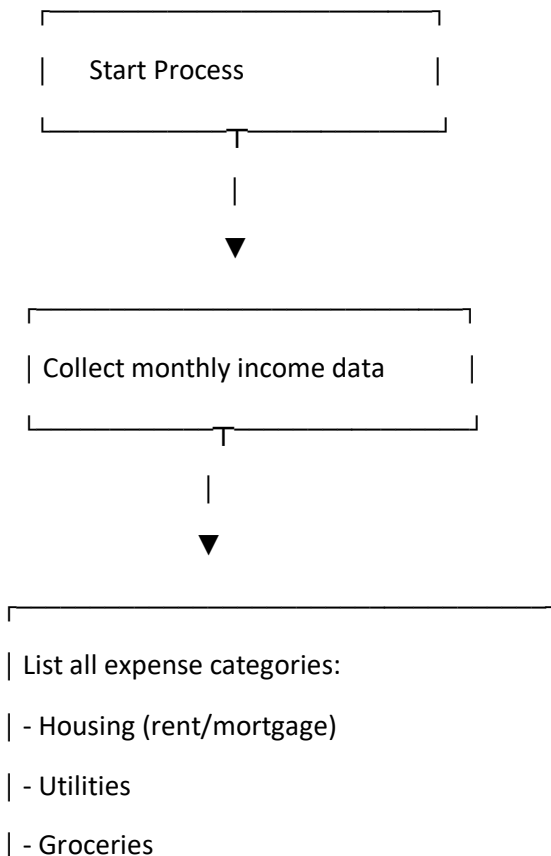
Script ☒ Turn on ECMAScript 2021 (ES12) mode

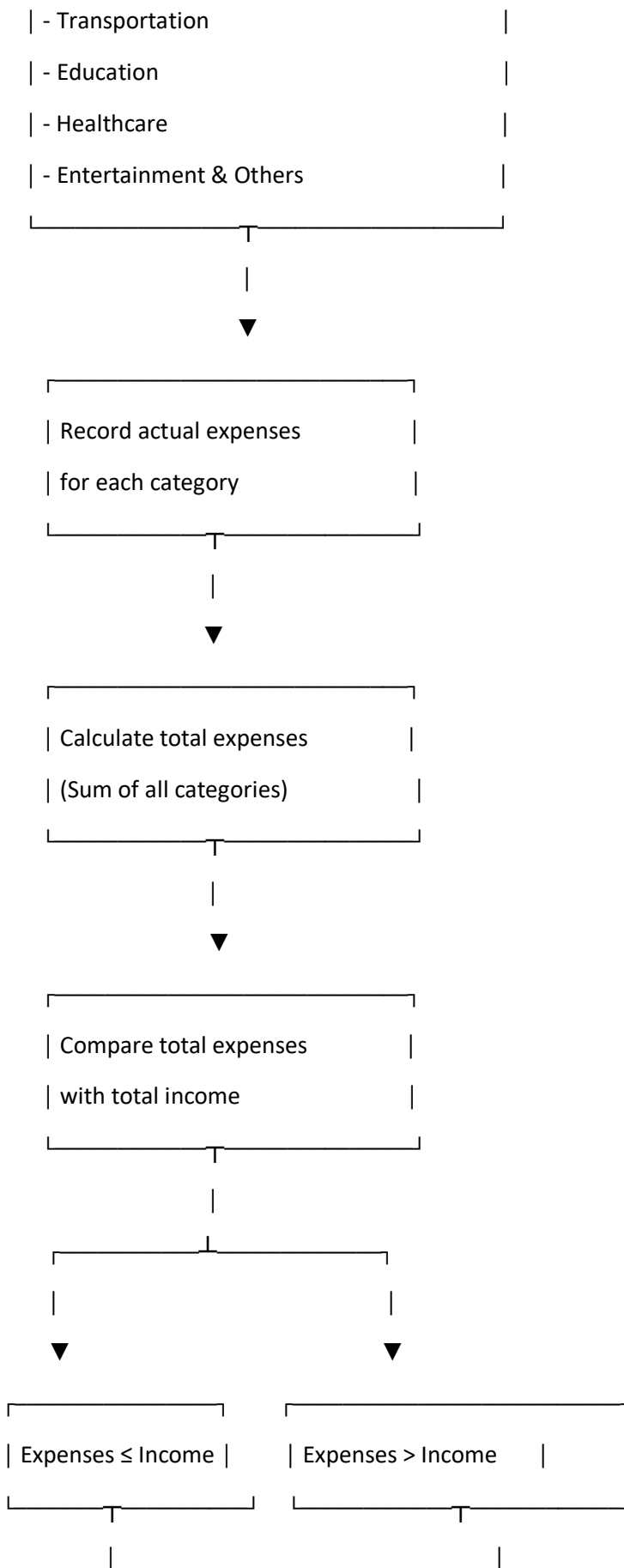
```
1 (function executeRule(current, previous /*null when async*/) {
2
3   var FamilyExpenses = new GlideRecord('u_family_expenses');
4   FamilyExpenses.addQuery('u_date', current.u_date);
5   FamilyExpenses.addQuery('u_date', current.u_date);
6 }
```

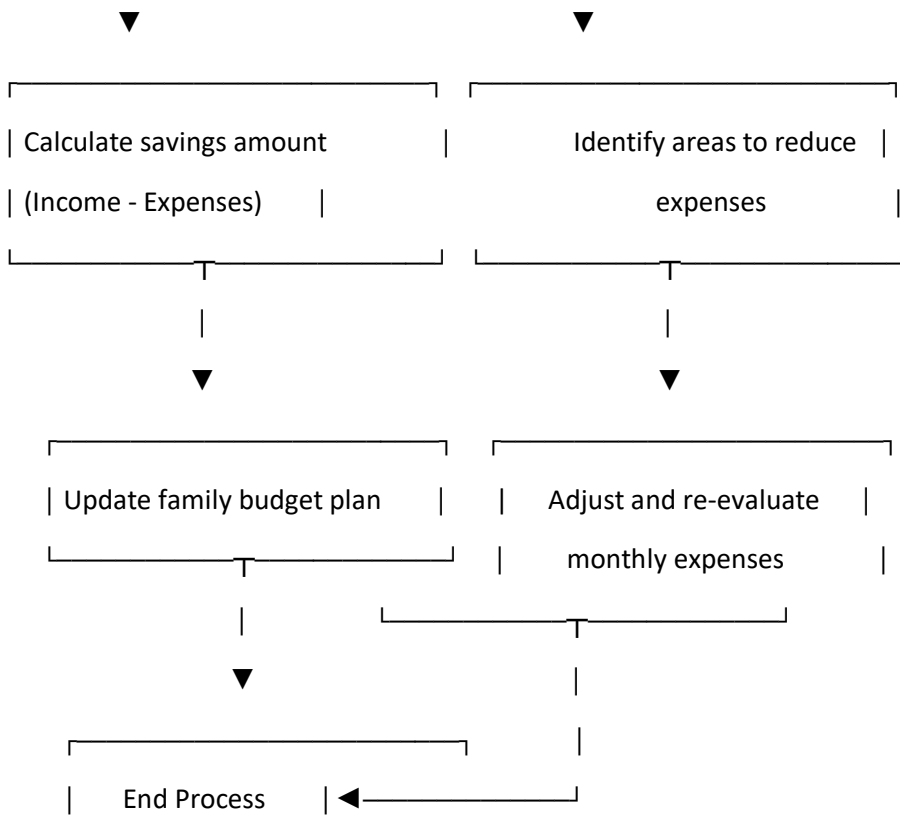




## 10. Flow chart

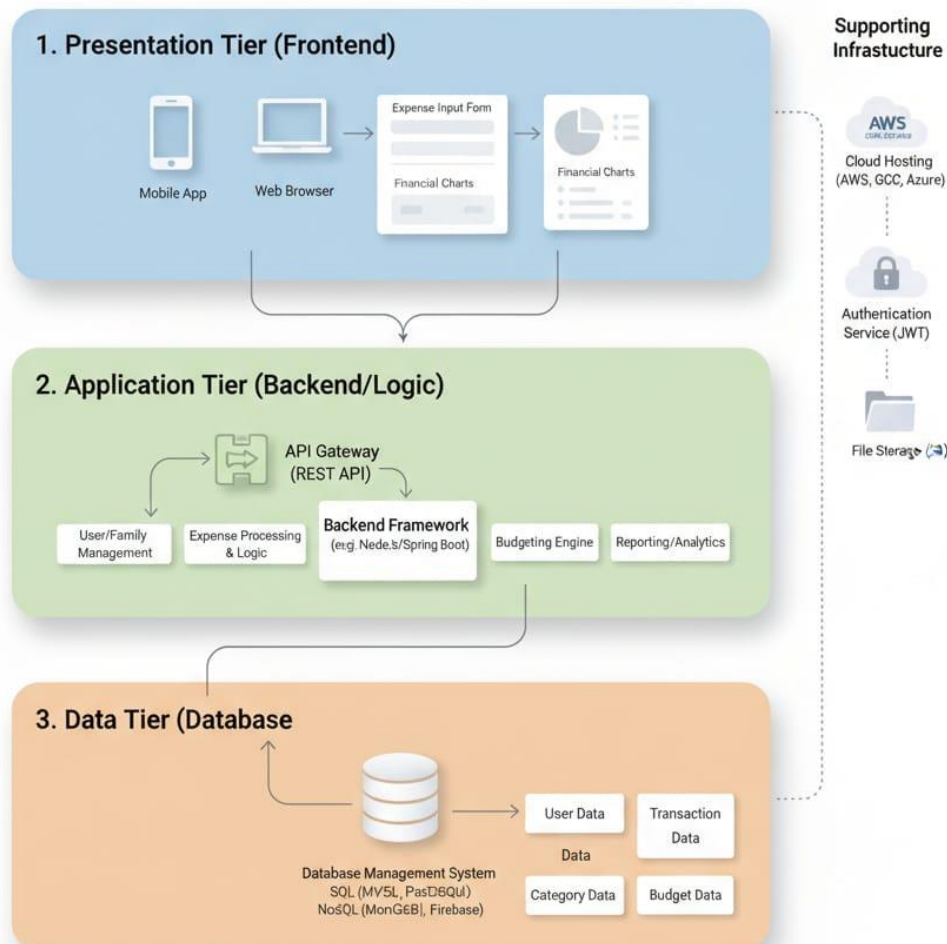






## 11. System architecture

### System Architecture: Family Expense Tracker (Three-Tier)



## 12. Testing

### Testing Types:

Test Type	Description	Result
Unit Testing	Tested each form and script individually	Passed
Integration Testing	Verified interaction between table, reports, and notifications	Passed

System Testing	Ensured full workflow operates correctly	Passed
User Acceptance Testing	Validated with sample family users	Passed

### Sample Test Case:

Test Case	Input	Expected Output	Result
Add Expense	\$100, Category=Food	Record added successfully	Pass
Invalid Amount	-\$50	Error: Invalid Input	Pass
Report Generation	Monthly Filter	Report displays total expenses	Pass

## 13. Conclusion

The Family Expense Calculation System developed on the ServiceNow platform successfully automates the process of tracking and analyzing family expenses. It provides an efficient, transparent, and user-friendly approach to financial management. The solution offers real-time expense monitoring, visual insights, and reporting, making it a practical tool for household budgeting and decision-making. Future enhancements could include integration with bank APIs, mobile notifications, and predictive analytics for smarter budgeting.

## 14. Future Scope

- \* Integration with Google Sheets or Excel for export.
- \* Incorporation of AI-based budget recommendations
- \* Development of a mobile app for on-the-go expense entry.
- \* Voice-based expense logging using virtual assistants.

### Final Deliverables

- \* ServiceNow application: \*Family Expense Management\*
- \* Reports: Monthly and Category-wise Expense Summary

- \* Dashboard: Visual analytics and charts

- \* Documentation: Project Report (this document)