Project Report: Family Expenses Tracker using ServiceNow

1. Project Title

Family Expenses Tracker Using ServiceNow

2. Objective

The aim of this project is to design and implement a basic expense tracking application on the ServiceNow platform. It includes creating custom tables, configuring forms, automating updates using business rules, and defining relationships between tables to associate daily expenses with overall family expenses.

3. Creation of Update Set

Step:

- Go to *All > Local Update Sets > New*
- **Name:** Family Expenses
- Click **Submit and Make Current**

Purpose:
The update set allows us to track all configuration changes made in this project.

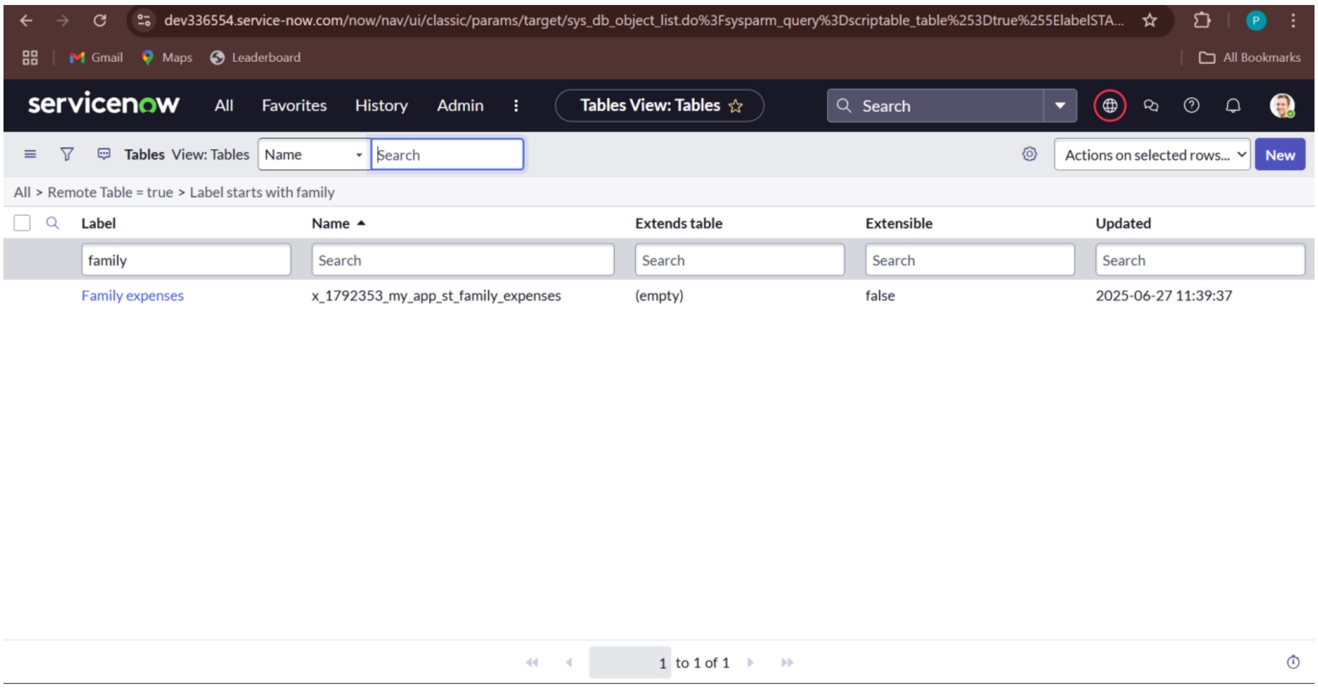
4. Creation of Family Expenses Table

Steps:

- Navigate to *System Definition > Tables > New*
- **Label:** Family Expenses
- **Name:** Auto-generated
- **Menu:** Family Expenditure

Columns Created:

Column Label	**Type**	**Max Length**	**Default Value**
Number	String	40	javascript:getNextObjNumberPadded()
Date	Date	-	-
Amount	Integer	40	-
Expense Details	String	800	-


dev336554.service-now.com/now/nav/ui/classic/params/target/sys_db_object_list.do%3Fsysparm_query%3Dscriptable_table%253Dtrue%255ElabelSTA...

All > Remote Table = true > Label starts with family

Label	Name	Extends table	Extensible	Updated
Family expenses	x_1792353_my_app_st_family_expenses	(empty)	false	2025-06-27 11:39:37

1 to 1 of 1

This table holds summarized daily expenses by date. The Number field is auto-generated.

5. Configuring Family Expenses Form

Steps:

- Navigate to *Family Expenses > Configure > Form Design*
- Arrange fields using drag & drop
- Set **Number** field as Read-Only
- Set **Date** and **Amount** fields as Mandatory

6. Creation of Daily Expenses Table

Steps:

- Navigate to *System Definition > Tables > New*
- **Label:** Daily Expenses
- **Menu:** Family Expenditure

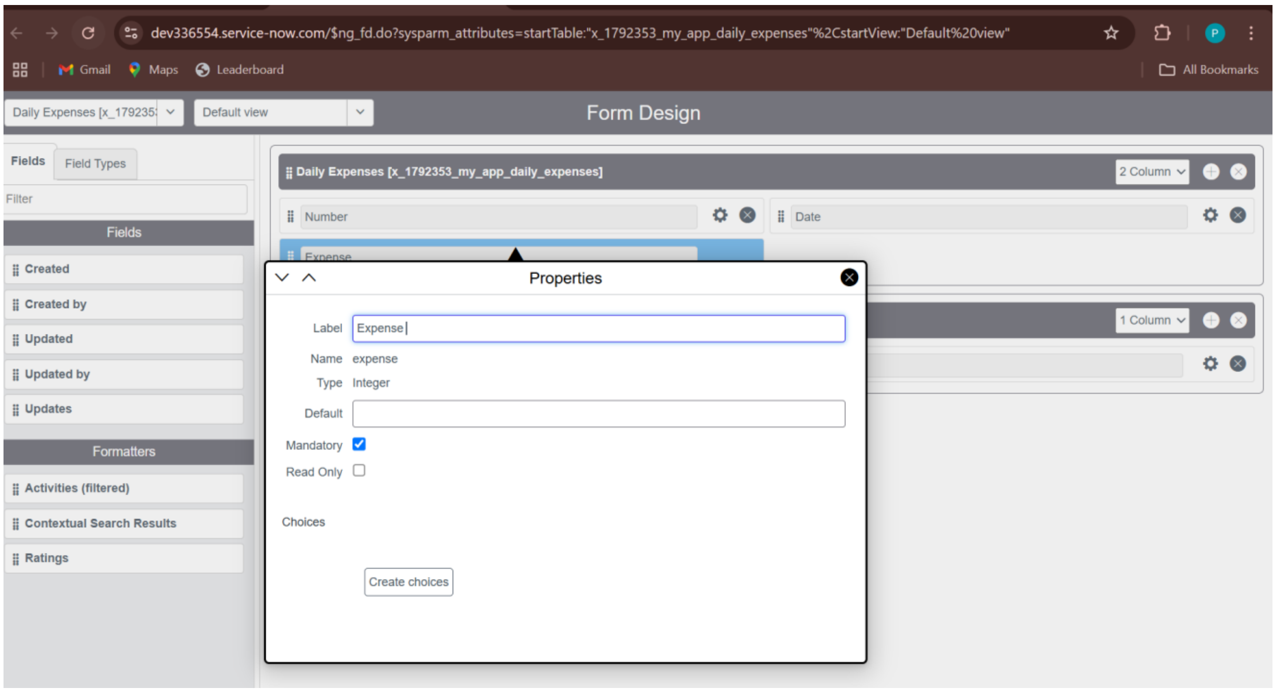
Columns Created:

Column Label	Type	Max Length	Default Value
Number	String	40	javascript:getNextObjNumberPadded()
Date	Date	-	-
Expense	Integer	-	-
Family Member Name	Reference	800	-
Comments	String	800	-

7. Configuring Daily Expenses Form

Steps:

- Navigate to *Daily Expenses > Configure > Form Design*
- Arrange fields properly
- Set **Number** as Read-Only
- Set **Date** and **Family Member Name** as Mandatory


dev336554.service-now.com/$ng_fd.do?sysparm_attributes=startTable:"x_1792353_my_app_daily_expenses"%2CstartView:"Default%20view"
Gmail
Maps
Leaderboard
All Bookmarks
Daily Expenses [x_179235:
Default view
Form Design
Fields
Field Types
Filter
Fields
Created
Created by
Updated
Updated by
Updates
Formatters
Activities (filtered)
Contextual Search Results
Ratings
Daily Expenses [x_1792353_my_app_daily_expenses]
2 Column
Number
Date
Expense
1 Column
Properties
Create choices

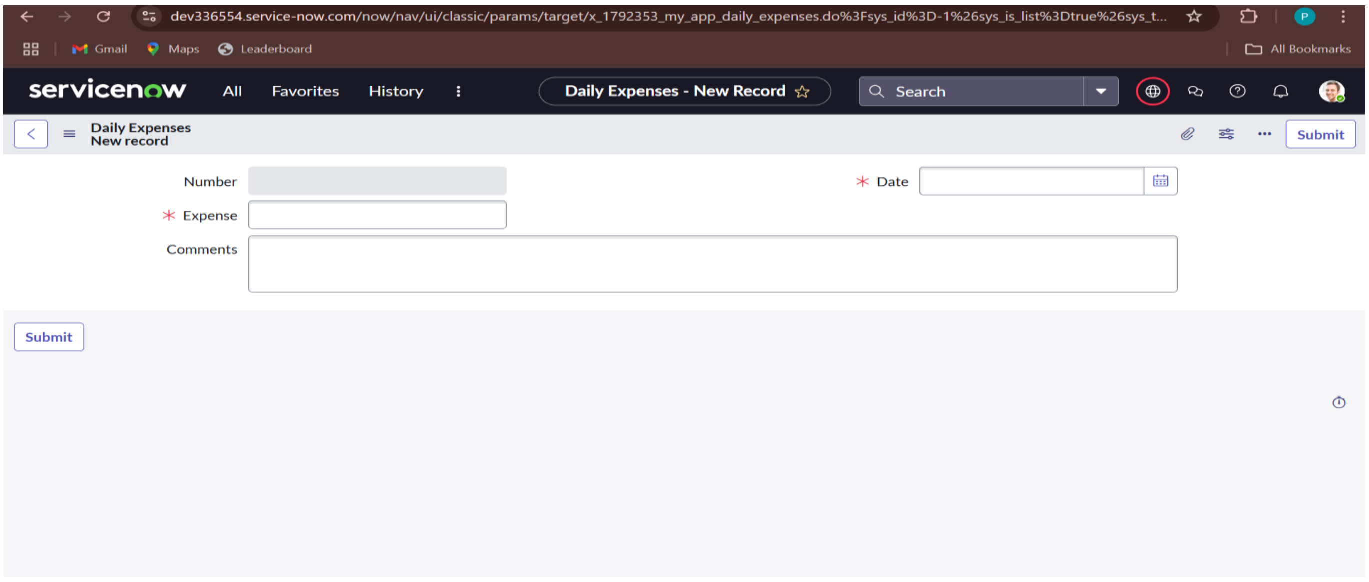
Label: Expense
Name: expense
Type: Integer
Default:
Mandatory: ☑
Read Only: ☐
Choices

Simplifies user input and maintains form integrity.

8. Creating a Daily Entry

Step:

- Navigate to *Daily Expenses > New*
- Enter sample values


dev336554.service-now.com/now/nav/ui/classic/params/target/x_1792353_my_app_daily_expenses.do%3Fsys_id%3D-1%26sys_is_list%3Dtrue%26sys_t...
Gmail
Maps
Leaderboard
All Bookmarks
servicenow
All
Favorites
History
Daily Expenses - New Record
Search
Daily Expenses
New record
Submit
Number
Expense
Comments
Date
Submit

Demonstrates how new daily expense entries are submitted.

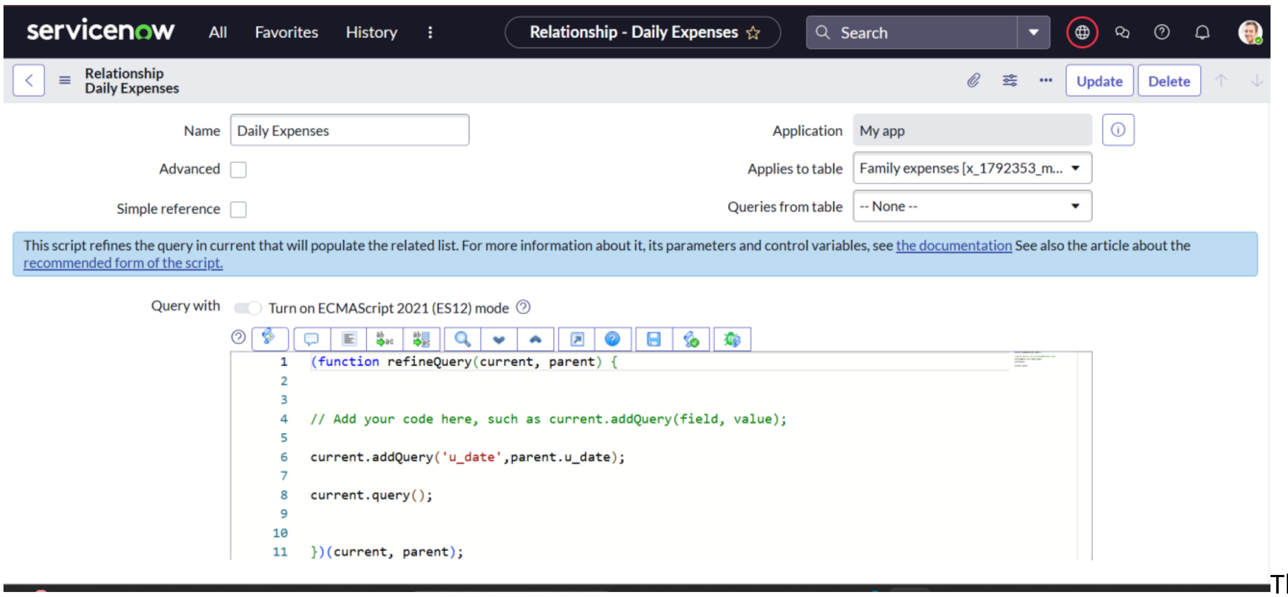
9. Creating Relationship Between Tables

Steps:

- Go to *System Definition > Relationships > New*
- **Name:** Daily Expenses
- **Applies to Table:** Family Expenses
- **Queries from Table:** Daily Expenses

Script Used:

```
(function refineQuery(current, parent) {
   current.addQuery('u_date', parent.u_date);
   current.query();
})(current, parent);
```



Name: Daily Expenses
Advanced: ☐
Simple reference: ☐
Application: My app
Applies to table: Family expenses [x_1792353_m...
Queries from table: -- None --

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

```
(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);
```

This links Daily Expenses with Family Expenses for the same date.

10. Configuring Related List

Steps:

- Go to *Family Expenses > Configure > Related Lists*
- Add **Daily Expenses** as related list

Purpose:
To view all related daily entries from within a family expense record.

11. Business Rule Creation

Steps:

- Go to *System Definition > Business Rules > New*
- **Name:** Family Expenses BR
- **Table:** Daily Expenses
- Check **Advanced**, and choose **Insert + Update**

Script Used:

```
(function executeRule(current, previous /*null when async*/) {
  var FamilyExpenses = new GlideRecord('x_1792353_my_app_st_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('x_1792353_my_app_st_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);
```

Explanation:
Automatically updates or inserts a family record when a daily entry is made.

12. Conclusion

This ServiceNow guided project allowed for hands-on experience in:

- Creating and customizing tables
- Managing data relationships
- Automating updates with business rules
- Designing forms for efficient data entry

The application successfully tracks family expenses by consolidating daily entries and generating summaries.