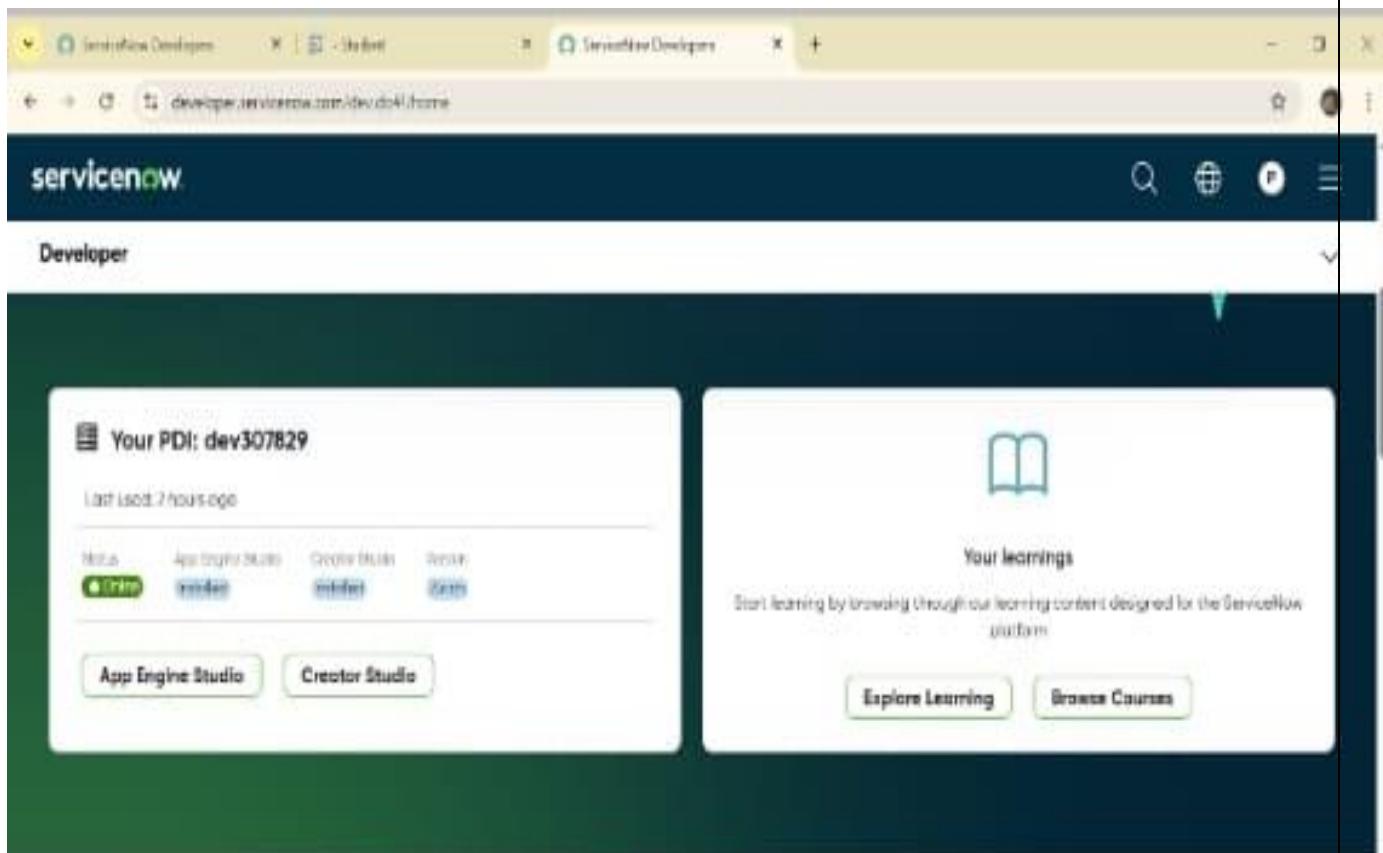


1. Setting up ServiceNow Instance:

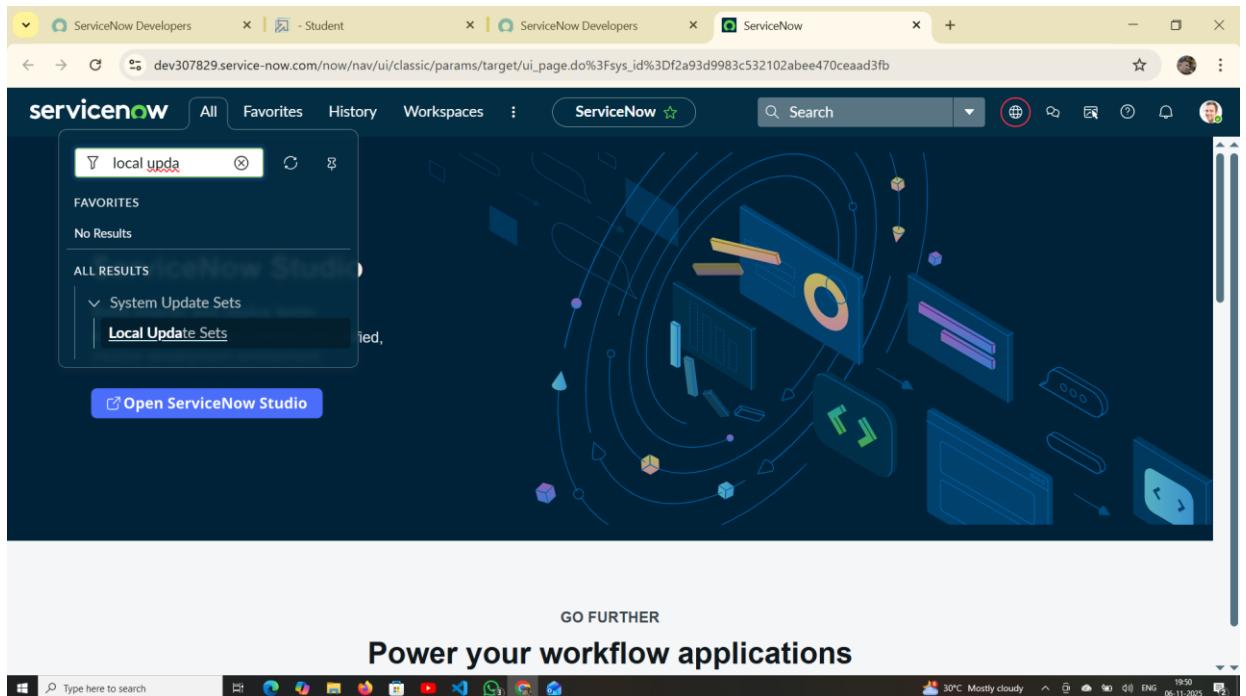
Setting up a **ServiceNow instance** involves creating a personal developer instance from the **ServiceNow Developer Portal**. After registration, users can request a new instance, which provides a cloud-based environment for development and testing. Once activated, administrators configure users, roles, and permissions, followed by creating tables, forms, and workflows. The instance can be customized using **Flow Designer**, **UI Builder**, and **Script Includes** to meet project requirements. This setup ensures a secure, scalable, and ready-to-use platform for building and deploying the **Family Expense Management System** efficiently.



Choose a learning plan to start

2. Creation of New Update Set:

An **Update Set** is created to capture all configuration changes made within the ServiceNow instance. This includes tables, forms, business rules, and scripts related to the expense tracking system. The Update Set helps in version control, making it easier to migrate or deploy project components between instances efficiently.



3. Creation of Table (Family Members)

A custom table is created to store **family member details** such as names, roles, and relationships. Each record represents a member of the family who will contribute to or manage expenses. This table forms the base for associating expenses with individuals and ensures organized data storage for user-specific financial tracking.

1. Creation of family expense table:

The screenshot shows the ServiceNow 'Table - New Record' interface. At the top, there are tabs for 'Table - New Record' and 'Form Design'. Below the tabs, the main area has a heading 'Table - New Record' with a star icon. A sub-header 'Table' and 'New record' is displayed. On the left, there's a note: '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.' A link 'More Info' is provided. The form fields include:

- * Label: Family Expenses
- * Name: u_st_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 Expenses
- Remote Table: (unchecked)

Below the form, there's a 'Columns' tab selected in a navigation bar, followed by 'Controls' and 'Application Access'. The 'Table Columns' section shows a table with columns for 'Column label', 'Type', 'Reference', 'Max length', 'Default value', and 'Display'. The table contains 10 rows of data.

2. Creation of Columns(Fields)

The screenshot shows the ServiceNow 'Table - Family Expenses' interface. The main area displays a table of columns for the 'Family Expenses' table. The table has columns for 'Column label', 'Type', 'Reference', 'Max length', 'Default value', and 'Display'. The data in the table is as follows:

	Column label	Type	Reference	Max length	Default value	Display
1	Sys ID	Sys ID (GUID)	(empty)	32		false
2	Date	Date	(empty)	40		false
3	Number	String	(empty)	40	javascript:getNextObjNumberPadded();	false
4	Amount	Integer	(empty)	40		false
5	Updated by	String	(empty)	40		false
6	Updates	Integer	(empty)	40		false
7	Updated	Date/Time	(empty)	40		false
8	Created by	String	(empty)	40		false
9	Created	Date/Time	(empty)	40		false
10	Expense Details	String	(empty)	800		false
	Insert a new row...					

3. Making Number Field an Auto-Number

The screenshot shows the ServiceNow Dictionary Entry - Number configuration page. The top navigation bar includes tabs for All, Favorites, History, Admin, and a search bar. The main area displays a form for a field named 'u_number'.

Field Details:

- Table: Family Expenses [u_st_family_ex...]
- Type: String
- Column label: Number
- Column name: u_number
- Max length: 40

Properties:

- Application: Global
- Active: checked
- Function field: unchecked
- Read only: checked
- Mandatory: unchecked
- Display: unchecked

A note at the bottom states: "Alters the behavior of a field or functionality that depends on the field. [More Info](#)".

Attributes: A large empty text input field.

Default Value: A tabbed section showing "Default Value" (selected) and "Calculated Value". A note says: "The Default value specifies what value the field has when first displayed."

4. Configure the Form

The screenshot shows the ServiceNow Form Design screen for a 'Family Expenses' form. The top navigation bar includes tabs for All, Favorites, History, Admin, and a search bar. The main area displays a form with two sections: 'Family Expenses [u_st_family_expenses]' and 'Expense Details'.

Fields Section:

- Number
- Date
- Amount

Expense Details Section:

- Expense Details

4. Creation of Table (Daily Expenses)

The **Daily Expenses** table records all individual transactions and expenses incurred by family members. Each entry includes details like expense date, category, amount, and description. This table serves as the central repository for financial data and connects with other tables through relationships for report generation and budgeting.

1. Creation of Daily Expenses Table

The screenshot shows the ServiceNow Table - New Record interface. The top navigation bar includes tabs for All, Favorites, History, Admin, and a search bar. Below the navigation is a toolbar with icons for back, forward, search, and other functions. The main area is titled "Table - New Record". A message bar at the top says "ServiceNow recommends creating custom tables in scoped applications. To learn more about creating scoped applications, click [here](#)". Below this, a note explains what a table is. The configuration form has the following fields:

* Label	Daily Expenses	Application	Global
* Name	u_daily_expenses	Create module	<input checked="" type="checkbox"/>
Extends table		Create mobile module	<input checked="" type="checkbox"/>
		Add module to menu	Family Expenditure
Remote Table <input type="checkbox"/>			

Below the configuration form is a "Table Columns" section with the following columns:

Column label	Type	Reference	Max length	Default value	Display

The status bar at the bottom shows the Windows taskbar with various pinned icons and the date/time as 06-11-2023.

2. Creation of Columns(Fields)

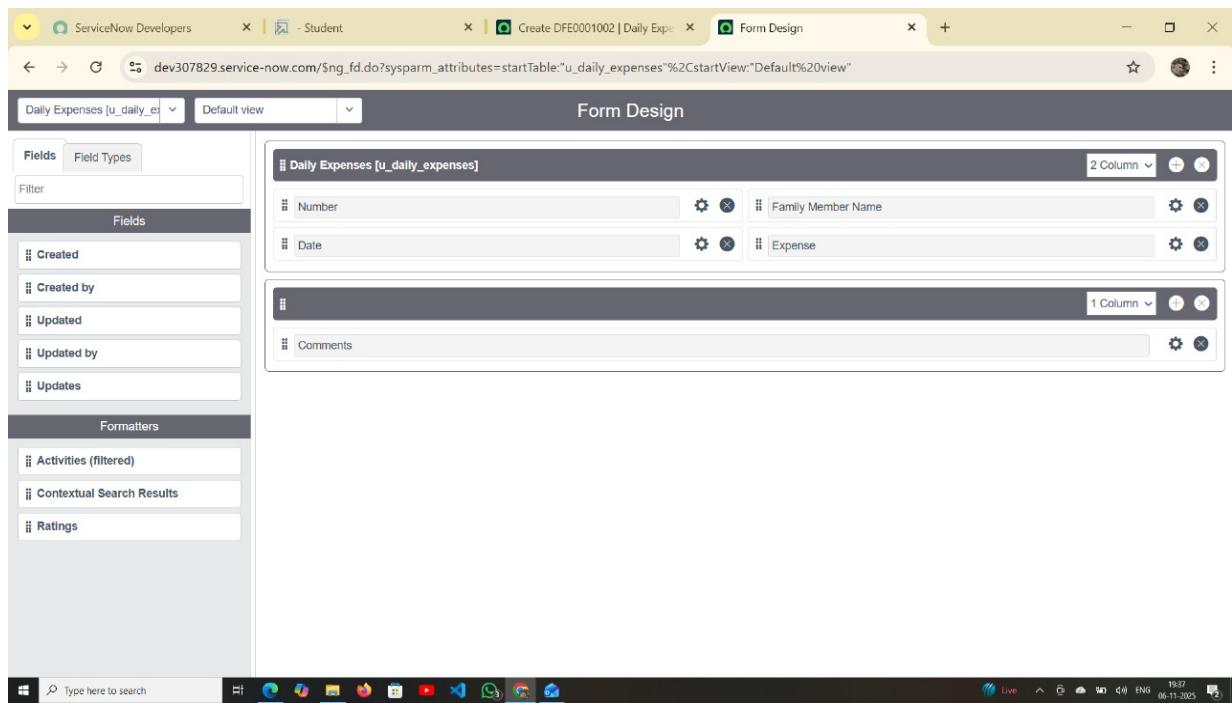
The screenshot shows the ServiceNow Table - Daily Expenses page. The top navigation bar includes tabs for All, Favorites, History, Admin, and a search bar. Below the navigation is a toolbar with Delete, Update, and Delete All Records buttons. The main content area is titled "Table Columns" and lists 11 columns with their details:

Column label	Type	Reference	Max length	Default value	Display
Family Member Name	Reference	Reference Style	32		false
Sys ID	Sys ID (GUID)	(empty)	32		false
Number	String	(empty)	40	javascript:getNextObjNumberPadded();	false
Updates	Integer	(empty)	40		false
Expense	Integer	(empty)	40		false
Date	Date	(empty)	40		false
Updated	Date/Time	(empty)	40		false
Updated by	String	(empty)	40		false
Created by	String	(empty)	40		false
Created	Date/Time	(empty)	40		false
Comments	String	(empty)	800		false

3. Making Number Field an Auto-Number

The screenshot shows the ServiceNow Dictionary Entry - Number configuration page. The top navigation bar includes tabs for All, Favorites, History, Admin, and a search bar. Below the navigation is a toolbar with Delete Column and Update buttons. A note at the top states: "A dictionary entry manages how ServiceNow stores data in tables and fields (columns). For new dictionary entries, select a Table and the field Type of the new column. Also enter a column label, which becomes the field label, and the column name. If necessary, set a Max length for text String type fields, make the field Mandatory to save a record, and make the field a Display Value for reference fields so it appears on records that reference this table. [More Info](#)". The configuration form includes fields for Table (Daily Expenses [u_daily_expenses]), Type (String), Column label (Number), Column name (u_number), Max length (40), Application (Global), Active (checked), Function field (unchecked), Read only (checked), Mandatory (unchecked), and Display (unchecked). At the bottom, there is a note about attributes and a choice list specification tab.

4. Configure the Form



5. Creation of Relationship

Relationships are established between tables (e.g., Family Members → Daily Expenses → Categories) to ensure data consistency. These links allow each expense to be tied to a specific member and category, enabling ServiceNow to produce structured reports and visual summaries for better financial insights.

The screenshot shows the ServiceNow interface for creating a new relationship. The top navigation bar includes tabs for 'ServiceNow Developers', '- Student', 'New Record | Relationship | Ser...', 'Form Design', and other system icons. The main title is 'Relationship - New Record'. The form fields include:

- Name: Daily Expenses
- Application: Global
- Advanced:
- Applies to table: Family Expenses [u_st_family_ex...]
- Queries from table: Daily Expenses [u_daily_expenses]

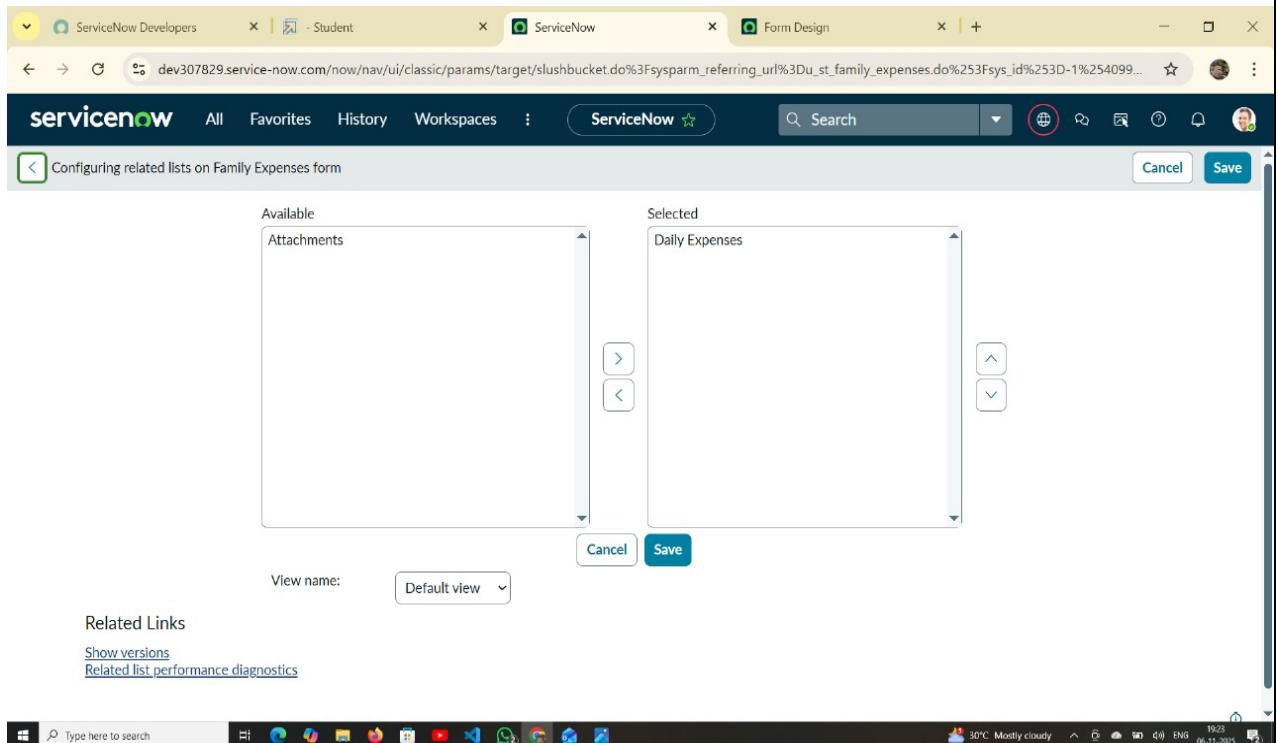
A note below the form states: "This script refines the query in current that will populate the related list. For more information about it, its parameters and control variables, see [the documentation](#). See also the article about the recommended form of the script."

The 'Query with' section contains a code editor with ECMAScript 2021 (ES12) mode enabled. The code is:function refineQuery(current, parent) {
 // Add your code here, such as current.addQuery(field, value);
}(current, parent);

At the bottom, there is a 'Submit' button and a taskbar with various application icons.

6. Configuring Related List on Family Expenses

In this step, **related lists** are configured to display connected records in one view. For example, the Family Member record shows all related daily expenses. This enhances usability, allowing users to access and manage associated data easily without navigating multiple tables or modules.



7. Conclusion

The project successfully demonstrates how **ServiceNow** can be used beyond traditional IT workflows to build an efficient **Family Expense Management System**. By combining automation, data structuring, and real-time reporting, the system helps users monitor spending, set budgets, and make informed financial decisions—promoting financial awareness and control within the family.