NM1051 – SERVICENOW ADMINISTRATOR – SMART INTERNZ

STREAMLINING TICKET ASSIGNMENT FOR EFFICIENT SUPPORT OPERATIONS

A PROJECT REPORT

Submitted by

KAVINILA K	(815422104301)
SRIHARINI A	(815422104055)
GRACELIN RUBAVATHY R	(815422104017)
JASMINE RITHIKA P	(815422104022)

BACHELOR OF ENGINEERING

IN

SEVENTH SEMESTER

COMPUTER SCIENCE AND ENGINEERING

SRI RAMAKRISHNA COLLEGE OF ENGINEERING

ANNA UNIVERSITY : CHENNAI 600 025
NOV/DEC 2025

ANNA UNIVERSITY : CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **“STREAMLINING TICKET ASSIGNMENT FOR EFFICIENT SUPPORT OPERATIONS”** is the bonafide work of **“KAVINILA K, SRIHARINI A, GRACELIN RUBAVATHY R, JASMINE RITHIKA P”** who carried out the project work under my supervision. No part of the dissertation has been submitted for any degree or any other academic award anywhere before.

SIGNATURE	**SIGNATURE**
Mrs. C.SURYA M.E.,	**Mr. R.DINESH RAJ M.E.,**
ASSISTANT PROFESSOR	**HEAD OF THE DEPARTMENT**
Department of CSE	Department of CSE
Sri Ramakrishna College of Engineering	Sri Ramakrishna College of Engineering
Perambalur	Perambalur

Submitted for the University Practical Examination held on……………….

INTERNAL EXAMINER **EXTERNAL EXAMINER**

TABLE OF CONTENT

1. ABSTRACT

This project, titled “Calculating Family Expenses Using ServiceNow,” focuses on developing an automated system to simplify household budgeting and expense management. The main objective is to replace traditional, manual tracking methods—such as notebooks or spreadsheets—with a centralized digital platform that ensures accuracy, accessibility, and real-time visibility.

The system leverages ServiceNow’s low-code/no-code capabilities to create a custom application that allows users to record, categorize, and analyse daily expenses efficiently. Each family member can enter their spending details, and the system automatically organizes the data based on category, date, and amount. By using tables, relationships, related lists, and business rules, the platform provides a complete structure for managing and summarizing all financial transactions.

2. PROBLEM DEFINITION

Managing family expenses manually can often lead to confusion, errors, and lack of clarity about where the money goes each month. Traditional methods like using notebooks or spreadsheets require manual calculations and frequent updates, which are time-consuming and prone to mistakes.

The goal of this project is to overcome these issues by developing a Family Expense Management System using ServiceNow. The system automates the process of recording and organizing expenses made by different family members. It ensures accuracy, easy tracking, and instant access to data.

By using ServiceNow’s features like tables, relationships, and business rules, the project provides a digital solution that makes expense tracking simple, efficient, and transparent for every household.

3. INTRODUCTION

Financial management is a key aspect of every household, yet many families struggle to maintain accurate and consistent records of their daily expenses. Traditional methods—like writing in notebooks, using spreadsheets, or relying on memory—often lead to missing data, confusion, and lack of clarity about where money is spent. With technology playing a larger role in everyday life, automating this process has become both practical and necessary.

This project aims to build a Family Expense Management System using ServiceNow, a cloud-based workflow automation platform widely known for IT and business process management. The same platform can be customized to track and analyze personal expenses through forms, tables, workflows, and reports.

In this system, each expense entry includes key details such as date, category, amount, and payment method. The information is stored securely in a database and linked to the respective family member who made the expense. Relationships and business rules help automate calculations and maintain data accuracy, while related lists and configuration options ensure the interface remains user-friendly.

4. MODULES AND TOOLS USED

The project was built entirely using ServiceNow, a cloud-based platform known for its low-code application development. Several key modules and tools were used during the development process:

1. Tables and Columns: Used to create structured data models for storing family and expense details.

2. Form Layouts and Design: Customized to make data entry easy and organized.

3. Reference Fields: Used to link the Family Member table with the Expense table.

4. Business Rules: Created to automate calculations and validations.

5. Client Scripts: Used for small frontend validations and dynamic field behavior.

6. Related Lists: Configured to display linked expenses under each family member.

7. Reports and Dashboards: Used to generate summaries and visual insights.

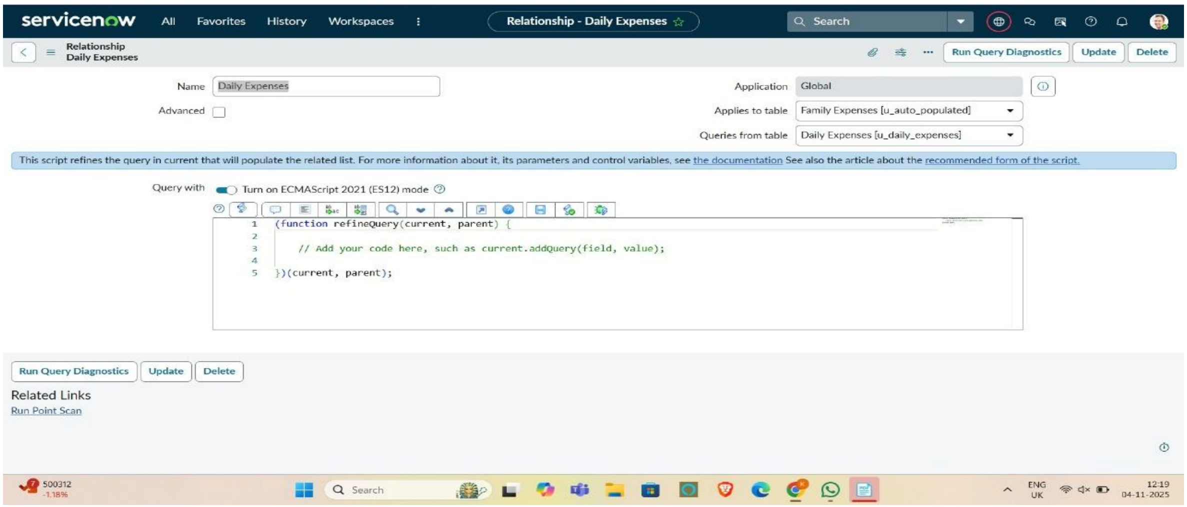
These modules together formed a simple yet complete system to manage and analyze expenses effectively.

5. CREATION OF TABLE

In ServiceNow, a table is the foundation for storing data. For this project, a custom table was created to record and manage family expense details.

The table includes essential fields such as:

- Date – To record the date of each expense.
- Category – To classify the expense (e.g., groceries, rent, bills, entertainment).
- Amount – To store the total spent.
- Payment Method – To indicate if the expense was paid by cash, card, or UPI.
- Description – To provide additional details about the transaction.



Name: Daily Expenses

Advanced ☐

Application: Global

Applies to table: Family Expenses [u_auto_populated]

Queries from table: Daily Expenses [u_daily_expenses]

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, parent);
```

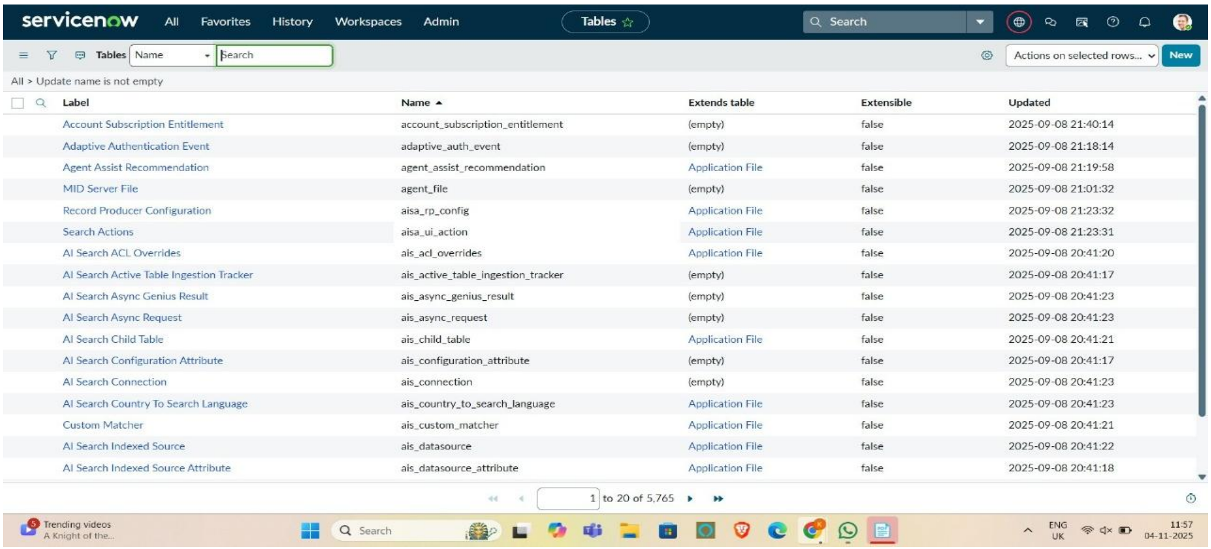
Columns (Fields)

Open the Family Expenses table and navigate to the Columns section.

Add new fields to capture necessary data such as:

- Date – Date/Time type to record when the expense occurred.
- Category – Choice field to select expense type (Food, Rent, Utilities, etc.).
- Amount – Currency or Integer type for expense value.
- Payment Method – Choice field for mode of payment (Cash, UPI, Card).
- Description – String/Text type for additional notes.
- Recorded By – Reference field linking to the Family Member table.
- After adding fields, save and verify that all are visible in the form layout.

Each field is designed with appropriate data type and mandatory settings to maintain data accuracy.



All > Update name is not empty

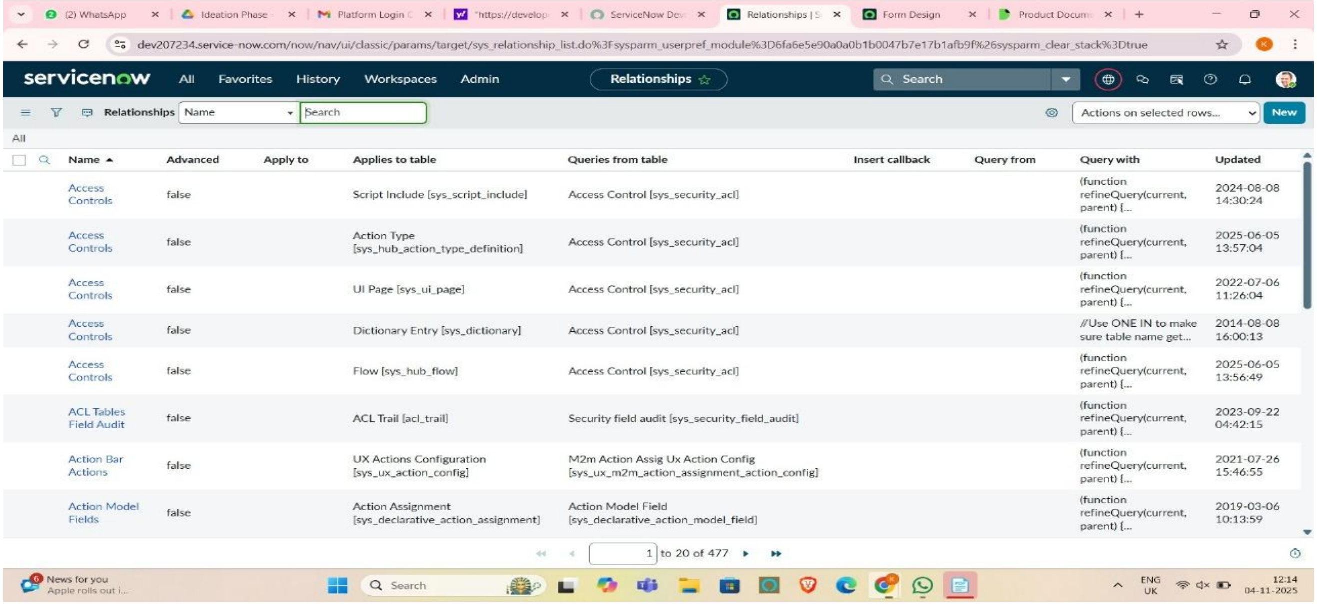
Label	Name ▲	Extends table	Extensible	Updated
Account Subscription Entitlement	account_subscription_entitlement	(empty)	false	2025-09-08 21:40:14
Adaptive Authentication Event	adaptive_auth_event	(empty)	false	2025-09-08 21:18:14
Agent Assist Recommendation	agent_assist_recommendation	Application File	false	2025-09-08 21:19:58
MID Server File	agent_file	(empty)	false	2025-09-08 21:01:32
Record Producer Configuration	aisa_rp_config	Application File	false	2025-09-08 21:23:32
Search Actions	aisa_ui_action	Application File	false	2025-09-08 21:23:31
AI Search ACL Overrides	ais_acl_overrides	Application File	false	2025-09-08 20:41:20
AI Search Active Table Ingestion Tracker	ais_active_table_ingestion_tracker	(empty)	false	2025-09-08 20:41:17
AI Search Async Genius Result	ais_async_genius_result	(empty)	false	2025-09-08 20:41:23
AI Search Async Request	ais_async_request	(empty)	false	2025-09-08 20:41:23
AI Search Child Table	ais_child_table	Application File	false	2025-09-08 20:41:21
AI Search Configuration Attribute	ais_configuration_attribute	(empty)	false	2025-09-08 20:41:17
AI Search Connection	ais_connection	(empty)	false	2025-09-08 20:41:23
AI Search Country To Search Language	ais_country_to_search_language	Application File	false	2025-09-08 20:41:23
Custom Matcher	ais_custom_matcher	Application File	false	2025-09-08 20:41:21
AI Search Indexed Source	ais_datasource	Application File	false	2025-09-08 20:41:22
AI Search Indexed Source Attribute	ais_datasource_attribute	Application File	false	2025-09-08 20:41:18

1 to 20 of 5,765

6. CREATION OF RELATIONSHIP

In ServiceNow, relationships are used to connect two or more tables so that data can be shared or referenced easily. For our project, we created a relationship between the Family Member table and the Expense table.

Each family member is linked to their respective expense records using a reference field. This relationship helps identify which member spent how much and on what category. It improves the clarity of the overall expense tracking process.



dev207234.service-now.com/now/nav/ui/classic/params/target/sys_relationship_list.do%3Fsysparm_userpref_module%3D6fa6e5e90a0a0b1b0047b7e17b1afb9f%26sysparm_clear_stack%3Dtrue

Name	Advanced	Apply to	Applies to table	Queries from table	Insert callback	Query from	Query with	Updated
Access Controls	false		Script Include [sys_script_include]	Access Control [sys_security_acl]			(function refineQuery(current, parent) {...	2024-08-08 14:30:24
Access Controls	false		Action Type [sys_hub_action_type_definition]	Access Control [sys_security_acl]			(function refineQuery(current, parent) {...	2025-06-05 13:57:04
Access Controls	false		UI Page [sys_ui_page]	Access Control [sys_security_acl]			(function refineQuery(current, parent) {...	2022-07-06 11:26:04
Access Controls	false		Dictionary Entry [sys_dictionary]	Access Control [sys_security_acl]			//Use ONE IN to make sure table name get...	2014-08-08 16:00:13
Access Controls	false		Flow [sys_hub_flow]	Access Control [sys_security_acl]			(function refineQuery(current, parent) {...	2025-06-05 13:56:49
ACL Tables Field Audit	false		ACL Trail [acl_trail]	Security field audit [sys_security_field_audit]			(function refineQuery(current, parent) {...	2023-09-22 04:42:15
Action Bar Actions	false		UX Actions Configuration [sys_ux_action_config]	M2m Action Assig Ux Action Config [sys_ux_m2m_action_assignment_action_config]			(function refineQuery(current, parent) {...	2021-07-26 15:46:55
Action Model Fields	false		Action Assignment [sys_declarative_action_assignment]	Action Model Field [sys_declarative_action_model_field]			(function refineQuery(current, parent) {...	2019-03-06 10:13:59

1 to 20 of 477

CONFIGURING RELATED LIST ON FAMILY EXPENSES

Once the relationship between the Family Member and Expense tables was created, the next step was to configure the related list. A related list in ServiceNow displays records from another table that are linked through a reference field or relationship. It provides a quick view of all associated records without having to navigate between multiple tables.

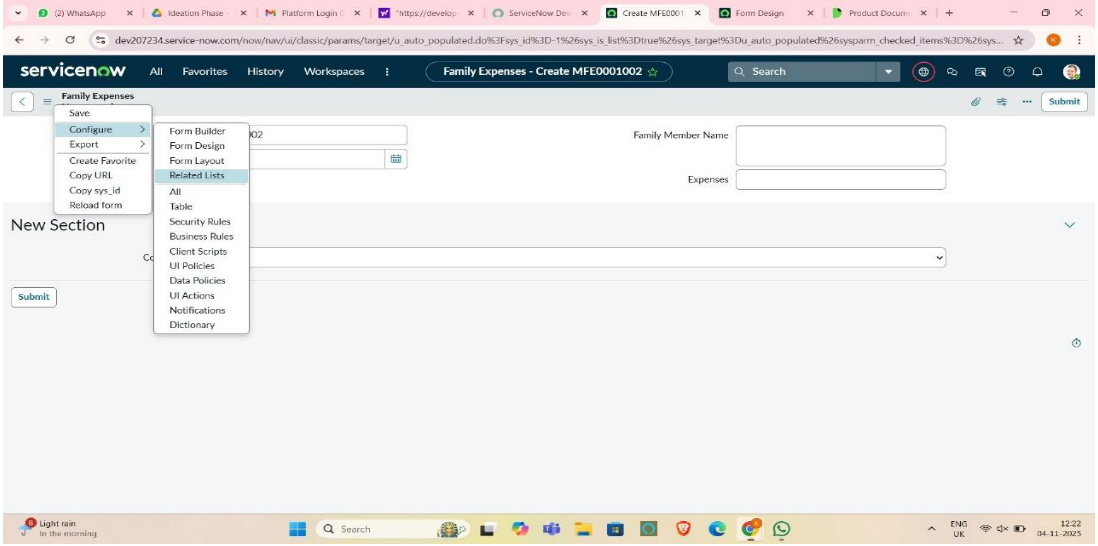
To configure the related list, we performed the following steps:

1. Opened the Family Member form in Form Design or Form Layout.

2. Selected the Related Lists tab.

3. Added the Expenses table as a related list using the reference field created earlier.

4. Saved and reloaded the form to verify that the related list appeared correctly.



7. CREATION OF BUSINESS RULES

Business Rules in ServiceNow are server-side scripts that run automatically when a record is inserted, updated, deleted, or displayed. They are mainly used to enforce consistency, apply conditions, and automate processes without manual input. In our Family Expense Management project, business rules helped maintain data accuracy and reduce repetitive work.

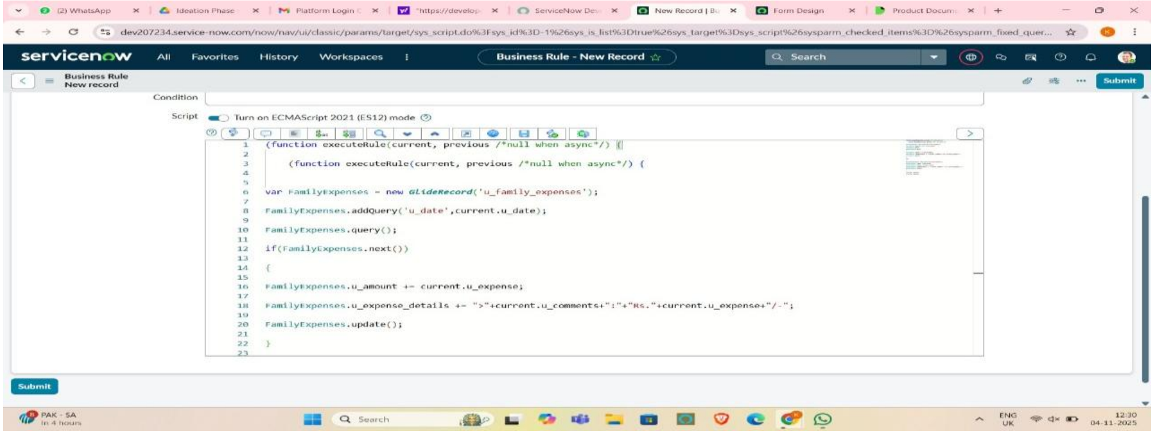
Steps to Create a Business Rule:

1. Navigate to System Definition → Business Rules.

2. Click on New to create a new rule.

3. Enter a name like Validate Expense Entry or Auto Calculate Total.

4. Choose the Table (in our case, “Expense”).


Business Rule - New Record
Condition
Script
Turn on ECMAScript 2021 (ES12) mode
Submit

```
(function executeRule(current, previous /*null when async*/) {

    (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();

}

```

8. CONFIGURE THE RELATIONSHIP

After creating the relationships and business rules, the next step was to configure them properly so that data flows smoothly between the connected tables. In ServiceNow, configuring relationships ensures that when a change happens in one table, the related tables update automatically, keeping the system consistent and synchronized.

Steps Followed to Configure the Relationship:

1. Opened the System Definition → Tables module.

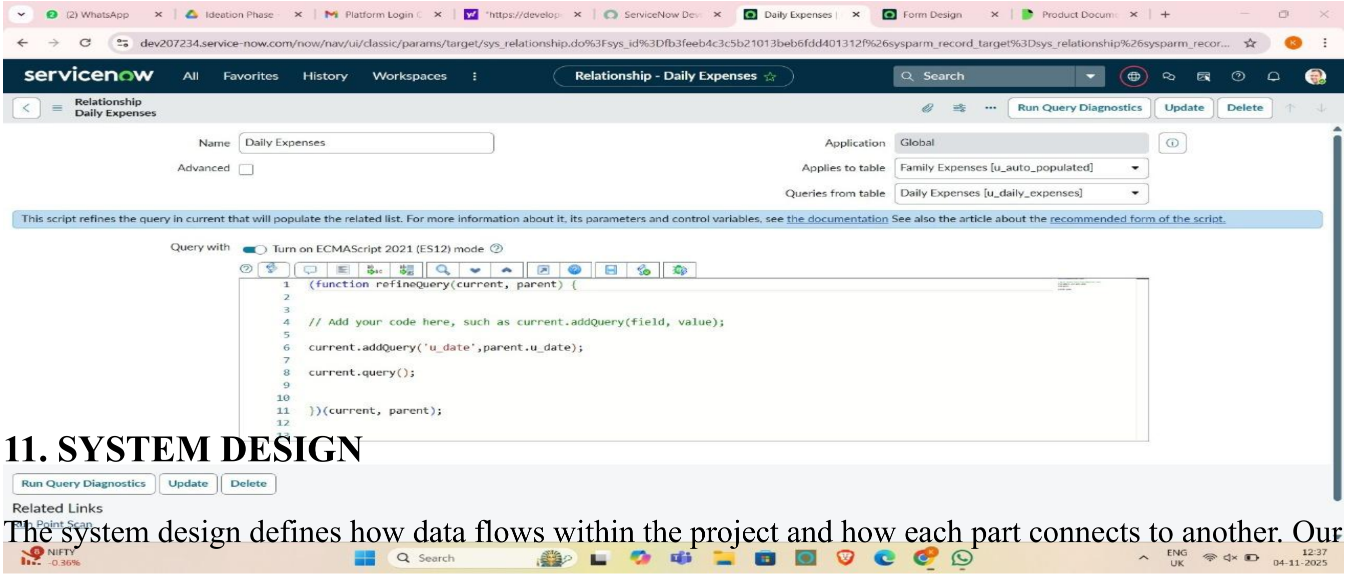
2. Selected the Expense table, where the reference field to the Family Member table was added.

3. Verified the reference field settings, ensuring it pointed to the correct table (Family Member).

4. Configured the Display Value so that the member’s name appears instead of just a system ID.

5. Enabled the Related List visibility under the Family Member form to show all linked expense records.

6. Tested the configuration by creating new family member and expense entries to confirm that relationships were reflected properly.


(2) WhatsApp
Ideation Phase
Platform Login
"https://develop
ServiceNow Dev
Daily Expenses
Form Design
Product Docum
dev207234.service-now.com/now/nav/ui/classic/params/target/sys_relationship.do%3Fsys_id%3Dfb3feeb4c3c5b21013beb6fdd401312f%26sysparm_record_target%3Dsys_relationship%26sysparm_recor...
servicenow
All
Favorites
History
Workspaces
Relationship - Daily Expenses
Search
Relationship
Daily Expenses
Run Query Diagnostics
Update
Delete

Name: Daily Expenses

Advanced: □

Application: Global

Applies to table: Family Expenses [u_auto_populated]

Queries from table: Daily Expenses [u_daily_expenses]

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);

```

11. SYSTEM DESIGN

The system design defines how data flows within the project and how each part connects to another. Our

9. IMPLEMENTATION PROCESS

The implementation involved a step-by-step setup within ServiceNow to bring all components together.

1. Creating a Custom Application:

2. Designing Tables:

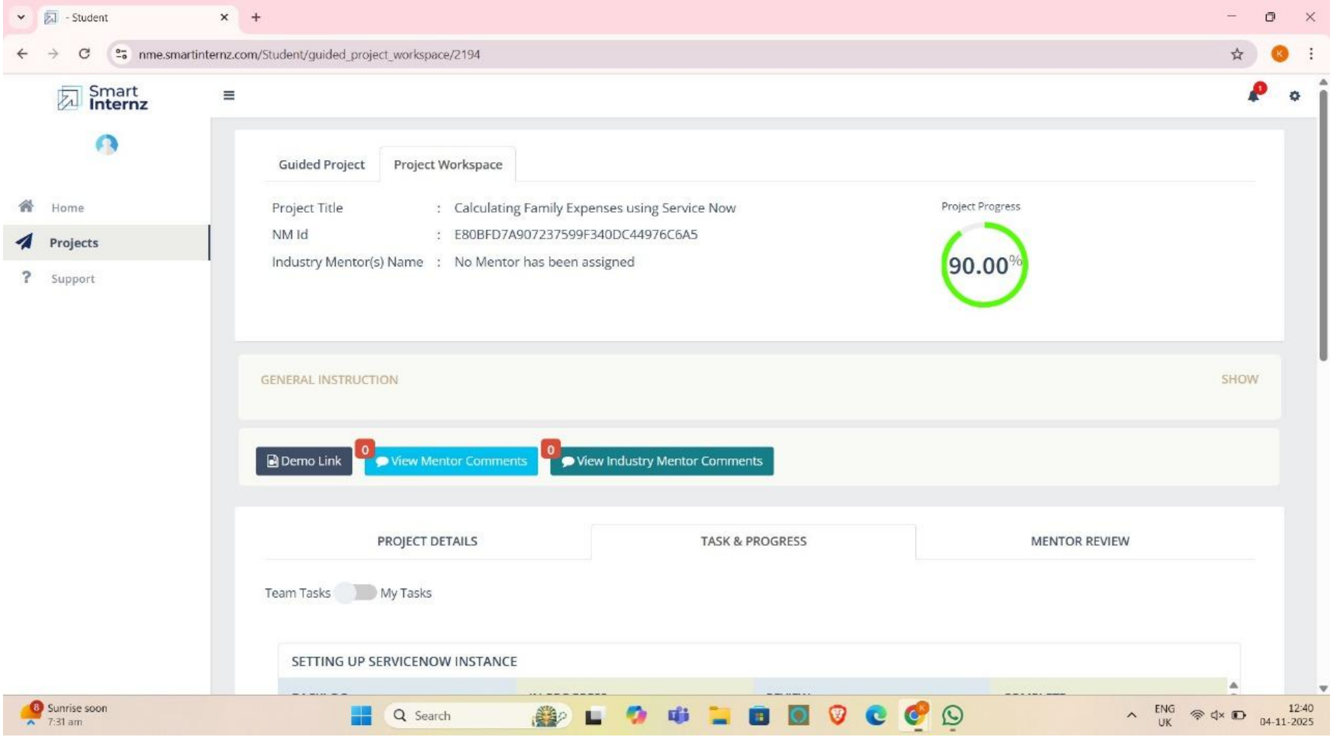
3. Establishing Relationships:

4. Configuring Related Lists:

5. Adding Business Rules:

10. CONCLUSION

The project “Calculating Family Expenses using ServiceNow” successfully demonstrated how the ServiceNow platform can be used to design and implement a practical solution for managing and analyzing financial data. Through this project, we learned how to create custom tables, establish relationships, configure related lists, and automate tasks using business rules. Each step helped in building a structured system capable of tracking expenses efficiently.



Project Title : Calculating Family Expenses using Service Now
NM Id : E80BFD7A907237599F340DC44976C6A5
Industry Mentor(s) Name : No Mentor has been assigned

Project Progress 90.00%

11. PRACTICE SCENARIOS FOR SERVICENOW ADMIN

1. Create a new user for a contractor, assign them to an "IT Support" group, and ensure they can only access the Incident application.

- **Create the Contractor User**
- Navigate to **Users** → *User Administration* > *Users*.
- Click **New**.
- Fill in details:
  - **User ID**: contractor1
  - **First name / Last name**: Contractor User
  - **Email**: contractor1@gmail.com
  - **Active**: Checked.
- Save.
- **Assign the User to the "IT Support" Group**
- On the user record, scroll to **Groups** (related list).
- Click **Edit**.
- Add to the **IT Support** group.
- Save.
- **Restrict Access to Only the Incident Application**

  Now we need to make sure this contractor can only work with **Incident**.

Option A: Role-Based Control (Mostly Preferred)

- By default, Incident application requires **itil** role.
- Instead of giving full **itil** access (which gives too much), do the following:
  - Create a **new custom role**, ex: **incident_contractor**.
  - Assign this role only to permissions needed for Incident (using ACLs).

    - Assign the new role to your contractor user.
    - Do **not** give **itil** or other broad roles.

Option B: Application Menu Restriction

- Go to **System Definition > Application Menus**.
- Open the **Incident** application menu.
- In the **Roles** field, add your custom role (**incident_contractor).**
    - This ensures only users with this role can see the Incident.
- **Verify Access**
- **Impersonate** the contractor user.
- Check:
    - They should only see the **Incident application** in the left nav.
    - They can open/create/edit incidents (based on the ACLs you configured).
    - They cannot access other apps (like Change, Problem, etc.).

2. Assign a role to a new group so members can read *Knowledge Articles* but cannot create or edit them.

    - **Create a New Group**
    - Navigate to User Administration > Groups.
    - Click New.
    - Enter a Name for the group (e.g., Knowledge Readers).
    - Optionally, add a Description.
    - Click Submit.
    - **Assign the Appropriate Role**
    - To allow read-only access to Knowledge Base articles, assign the **knowledge** role:

- Open the newly created group.
- Scroll to the Roles related list.
- Click Edit.
- Add the role: knowledge
- This role allows users to view published articles.
- Click Save.
- **Do NOT assign roles like **knowledge_admin or knowledge_manager,** which grant create/edit permissions.
- **Add Users to the Group**
- In the group record, scroll to the Group Members related list.
- Click Edit.
- Select users you want to add.
- Click Save.
- **Verify Access**
- Log in as one of the group members.
- Navigate to Knowledge > Articles.
- Confirm they can view articles.
- Try creating or editing an article — they should not have access.

3. Configure a UI Policy that hides the "Work Notes" field unless the state is "In Progress".
Solution:

- **Navigate to UI Policies**
- Go to Application Navigator → type UI Policies → click System UI > UI Policies.

- Create a New UI Policy
- Click New.
- Select the Table → e.g., *Incident* (or whichever table you're working on).
- Provide a Name (e.g., *Hide Work Notes unless In Progress*).
- In the Conditions section, set:
- Field = *State*
- Operator = *is*
- Value = *In Progress*.
- Check the box Active.
- Save the record.
- **Add a UI Policy Action**
- In the same UI Policy record, scroll to UI Policy Actions (Related List).
- Click New.
- Configure the action:
- Field name = *Work notes*
- Visible = *True* (since you want it visible only when the condition is met).
- Submit the action

4. Configure a UI Policy to hide Notes section in incident, when state is In Progress.

- **Solution**:
- **Navigate to UI Policies**
- Go to Application Navigator → type UI Policies → click System UI > UI Policies.
- Create a New UI Policy

- Click New.
- Select the Table → e.g., *Incident* (or whichever table you're working on).
- Provide a Name (e.g., *Hide Work Notes unless In Progress*).
- In the Conditions section, set:
- Field = *State*
- Operator = *is*
- Value = *In Progress*.
- Check the box Active.
- Save the record.
- **Make Run Script box True**
- Just write one line of code:
- g_form.setSectionDisplay('notes',false);
- Submit the action

5. **Configure a response SLA, the SLA should pause, when the incident state is in On Hold vice versa**.

- **Create or Modify an SLA Definition**
- Navigate to **Service Level Management > SLA Definitions**.
- Click **New** or open an existing SLA (e.g., "Response SLA").
- Fill in the basic details:
- **Name**: Response SLA
- **Table**: **Incident**
- **Type**: Response
- **Duration**: Set your desired time (e.g., 1 hour)
- **Set SLA Conditions**

- Under the **Start Condition**:
- Example: **State is New**
- Under the **Stop Condition**:
- Example: State is Resolved or Closed
- Under the **Pause Condition**:
- Add: **State is On Hold**
- This ensures the SLA timer **pauses** when the incident is moved to **On Hold**, and **resumes** when it returns to another **New** state
- **Test the SLA Behavior**
- Create a test incident.
- Confirm SLA starts when an incident is created.
- Change state to **On Hold** — SLA should pause.
- Change back to **Active** — SLA should resume.
- Resolve the incident — SLA should stop.

6.Configure an email notification that alerts the assigned group whenever a new *Change Request* is created.

- **Solution:**
- **Navigate to Notifications**
- In the **Application Navigator**, type **Notifications**.
- Go to **System Notification > Email > Notifications**.
- **Create a New Notification**
- Click **New**.
- Fill in the basic details:
- **Name**: *New Change Request Assigned Group Alert*

- **Table**: *Change Request [change_request]*
- **Active**: Checked
- **Define When to Send**
- Under **When to send**, configure:
- **When to send**: *Insert* (since you want this when a new record is created).
- **Define Who Will Receive**
- In the **Recipients** tab:
- Under **Users/Groups in fields**, choose **Assigned to group** (or the field name for assigned group).
- This ensures the entire assigned group gets the email.
- **Define What Will Contain**
- In the **What it will contain** tab:
- **Subject**: New Change Request Created - ${number}
- **Message HTML** (sample):

  A new Change Request has been created.
- Number: ${number}
- Short Description: ${short_description}
- Requested By: ${requested_by}
- Assignment Group: ${assignment_group}
- State: ${state}
- Please review and take necessary action.
- **Save & Test**
- Save the Notification.

- Create a new **Change Request** record, assign it to a group.
- Verify that the email goes out to all members of the Assigned Group.

7. Create a report showing the number of incidents opened by each department in the last 30 days**.**

- **Navigate to Reports**
- Go to Reports > Open Reports Modules.
- Click Create a Report.
- **Define Report Source**
- Name: **Incidents by Department - Last 30 Days**
- Source Table: **Incident**
- **Set Conditions**
- **Under Filter, add:**
- Opened At → on or after → Today - 30 days
- Department → is not empty ***(optional, to exclude unassigned)***
- **Choose Report Type**
- Select Type: **Bar Chart** or **Pie Chart** (or **List** if you prefer tabular view)
- **Configure Grouping**
- Under Group By, select: **Department**
- Under Aggregation, choose: **Count**
- **Save and Run**
- Click Save.
- Click Run to view the report.

8. Build a dashboard for Service Desk Managers showing KPIs like incidents by priority, created within a week, state wise also

- **Step 1: Create Individual Reports**
- **You’ll need to create three separate reports first:**
- **Incidents by Priority**
- Go to: Reports > Create New
- Name: Incidents by Priority
- Source Table: Incident
- Type: Bar Chart or Pie Chart
- Group By: Priority
- Aggregation: Count
- Filter: Opened At → on or after → Today - 30 days
- **Incidents Created Within a Week**
- Name: Incidents Created - Last 7 Days
- Source Table: Incident
- Type: Time Series or Bar Chart
- Filter: Opened At → on or after → Today - 7 days
- Group By: Opened At (Daily)
- Aggregation: Count
- **Incidents by State**
- Name: Incidents by State
- Source Table: Incident
- Type: Bar Chart or Pie Chart
- Group By: State

- Aggregation: Count
- Filter: Opened At → on or after → Today - 30 days
- **Step 2: Create a Dashboard**
- Go to Self-Service > Dashboards.
- Click Create New Dashboard.
- Name: **Service Desk Manager KPIs**
- Add a Proper Description
- Click Submit.
- **Step 3: Add Reports to the Dashboard**
- Open the newly created dashboard.
- Click Edit Content.
- Use Add Reports to include:
- **Incidents by Priority**
- **Incidents Created - Last 7 Days**
- **Incidents by State**
- Arrange the widgets as needed for clarity.

9. Restrict the ability to delete records in the *Change Request* table so only users with the "admin" role can do so.

- **Navigate to Access Control (ACLs)**
- In the **Application Navigator**, type **Access Control**.
- Go to **System Security > Access Control (ACL)**.
- **Create a New ACL Rule**
- Click **New**.
- Fill in details:

- **Type**: *record*
- **Operation**: *delete*
- **Table**: *Change Request [change_request]*
- **Name**: *(auto-populates when you pick table + operation)*
- **Define the Condition / Role**
- In the **Requires role** field, add: **admin**
- This ensures only users with the **admin** role can delete records.
- **Save & Test**
- Save the ACL.
- Test with a non-admin user → they should **not** see the delete option (or get a permission error if they try via URL).
- Test with an admin user → delete should work normally.

10. Create a custom table and create two reference fields (ex: assignment group and assigned to). Display the users based on selection of assignment group.

- **Create a Custom Table**
- In the Application Navigator, type **Tables**.
- Go to **System Definition > Tables**.
- Click **New**.
- Name: *u_custom_task*
- Label: *Custom Task*.
- Save.
- **Add Fields**
- Open your table and go to the **Columns** tab.

- Add two reference fields:
- **Assignment Group** → Type = *Reference*, Table = *sys_user_group*.
- **Assigned To** → Type = *Reference*, Table = *sys_user*.
- **Configure Reference Qualifier on "Assigned To"**
- We need to filter "Assigned To" users based on the selected Assignment Group.
- **Using Reference Qualifier**
- Right click on the **Assigned To** field, click on **Configure Dictionary.**
- Go to **Dependent** Section, give the name of the Assignment Group(ex: u_ass_group)
- Update and Test the functionality.

11.How to auto assign incidents when user selects a category as network, the same incident be assigned to Network group.

- **Solution:**
- Go to Flow Designer → Designer.
- Click New Flow.
- Name: Assign Incident by Category
- Trigger: Created or Updated → Table = Incident
- Add a If action (Condition) with expression:
- Select Trigger Record Category is Network
- Under the If branch, add Action → Update Record:
- Record: Trigger → Incident(Trigger Record)
- Set field Assignment group → Network
- Save and Activate the flow.

- Test the Flow.

12. HR Groups members are only able to see HR Related Records in servicenow?

- **Solution:**
- **Step 1: Create a Role for HR Access**
- Navigate to:
  User Administration → Roles → New
- Enter:
- Name: hr_access
- Description: Role to allow access to HR Cases
- Click Submit.
- **Step 2: Assign the Role to HR Group**
- Navigate to:
  User Administration → Groups
- Open your HR group record.
- In the Roles tab → click Edit.
- Move hr_access from Available → Selected.
- Click Save.
- Now all members of the HR group have the hr_access role.
- **Step 3: Create Access Control (ACL) for Viewing HR Cases**
- Navigate to:
  System Security → Access Control (ACL)
- Click New.
- **Step 4: Define Access Condition (No Script)**
- Scroll down to the Requires role section:

- Add the Role hr_access.
- This means only users with the hr_access role can read/view HR Case records.
- **Step 5: Save and Test**
- Click Submit or Update to save the ACL.
- Impersonate a non-HR user:
- Go to your profile → click Impersonate User → choose a user *not in the HR group*.
- Try opening an HR Case record → You should see a "Security constraints prevent access to requested page" message.
- Now impersonate an HR group member:
- They should be able to open HR Cases normally

13. When the Incident state changes to In Progress, Child incident related list should be hidden.

- **Solution:**
- Navigate to System UI → UI Policies → New.
- Fill the header:
- Name: Hide related lists when State is In Progress
- Table: Incident
- Active: checked
- Global: checked
- Condition: **State is In Progress**
  (Use the exact label used in your instance for the In Progress state.)
- Submit the UI Policy record.
- In the UI Policy record click **New** under **UI Policy Actions**.
- Set:

- **Field name:** select the related list–Child incident
- **Visible:** false
- **Read only:** optional
- Save and Test the UI Policy Action.

14.How to Display Incident number while loading the incident form

- **Solution:**
- Navigate to System UI → Client Scripts → New.
- Fill the header:
- Name: Show Incident Number on Load
- Table: Incident
- Type: onLoad
- Active: True
- Add this script:
- function onLoad() {
-   // Get the Incident number field value
-   var incNum = g_form.getValue('number'); // 'number' is the field name
-   alert('Incident Number: ' + incNum);
- }

15. When the Incident state changes to In Progress, description should be hidden and short description should be mandatory.

- **Solution:**
- **Step 1 — Navigate to Client Scripts**

- Go to:
  System UI → Client Scripts → New
- Fill the header:
- Name: Hide Description and Make Short Description Mandatory
- Table: Incident
- Type: onChange
- Field name: state
- Active: checked
- **Step 2 — Add the Client Script Code**
- function onChange(control, oldValue, newValue, isLoading) {
- if (isLoading) return;
- if (newValue === '2') {
- g_form.setDisplay('description', false);
- g_form.setMandatory('short_description', true);
- } else {
- g_form.setDisplay('description', true);
- g_form.setMandatory('short_description', false);
- }
- }
- Click **Submit** or **Update** to save.

15. If the description field is empty in the incident table, prevent the form submission.

- **Solution:**
- **Step 1 — Navigate to Client Scripts**

- Go to:
  System UI → Client Scripts → New
- Fill the header:
- Name: Prevent Submit if Description Empty
- Table: Incident
- Type: onSubmit
- Active: checked
- **Step 2 — Add the Client Script Code**
- function onSubmit() {
- var description = g_form.getValue('description');
- if (description == ") {
- g_form.addErrorMessage('Description cannot be empty');
- return false;
- } else {
- return true;
- }
- }

16. Users can not change the state field values in the incident list.

- **Solution:**
- **Step 1 — Navigate to Client Scripts**
- Go to:
  System UI → Client Scripts → New
- Fill the header:
- Name: Prevent State Inline Edit

- Table: Incident
- Type: onCellEdit
- Field name: state
- Active: checked
- **Step 2 — Add the Client Script Code**
- if(newValue==2){
- alert('You can not edit this value');
- saveAndClose==false;
- }else{
- saveAndClose==true;
- }

17. How to set the Caller to Logged in user automatically in the incident table.

- **Solution:**
- Navigate: System Definition → Business Rules → New
- Fill the details:
- Name: Set Caller on Incident Create
- Table: Incident
- When: before
- Insert/update: checked
- Advanced: true
- **Script**:
- current.caller_id = gs.getUserID();

18. When a user updates an incident record, priority should change to Critical automatically.

- **Solution:**

- Navigate: System Definition → Business Rules → New

- Settings:
- Name: Set Priority field
- Table: Incident
- When: before
- Update:checked
- Script:
- current.impact = 1;
- current.urgency = 1;

19.Create a button on the Incident form that allows users to mark an Incident as Resolved with a single click.

- **Solution:**
- Navigate: System UI → UI Actions → New
- Settings:
- Name: Resolve Incident
- Table: Incident
- Action type: Form button
- Active: checked
- Script:
- current.state = 6;

20.Create a button on the incident table that copies the Short Description value into the Description field.

- **Solution:**
- Navigate: System UI → UI Actions → New
- Settings:
- Name: Copy Short Description
- Table: Incident
- Action type: Form button
- Active: checked

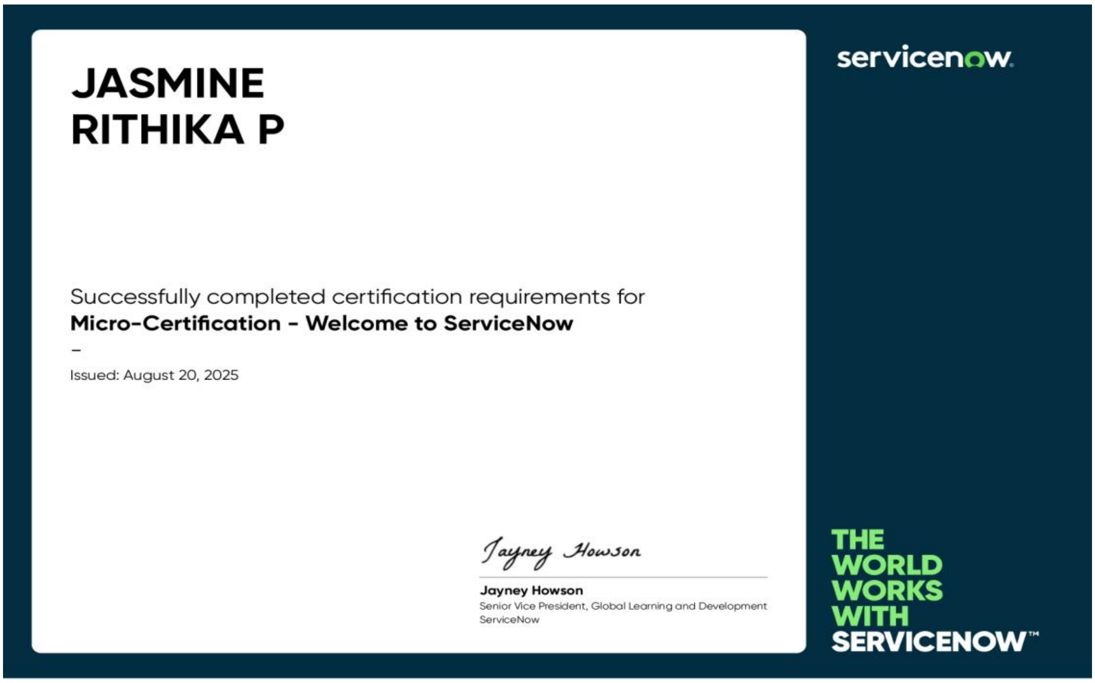
- Script:
- current.description = current.short_description;
- current.update();
- action.setRedirectURL(current);

Source Code

https://github.com/Kavimagizh19/family__expenses

1. CERTICICATIONS

1. Micro-Certification - Welcome to ServiceNow



servicenow.

JASMINE RITHIKA P

Successfully completed certification requirements for

Micro-Certification - Welcome to ServiceNow

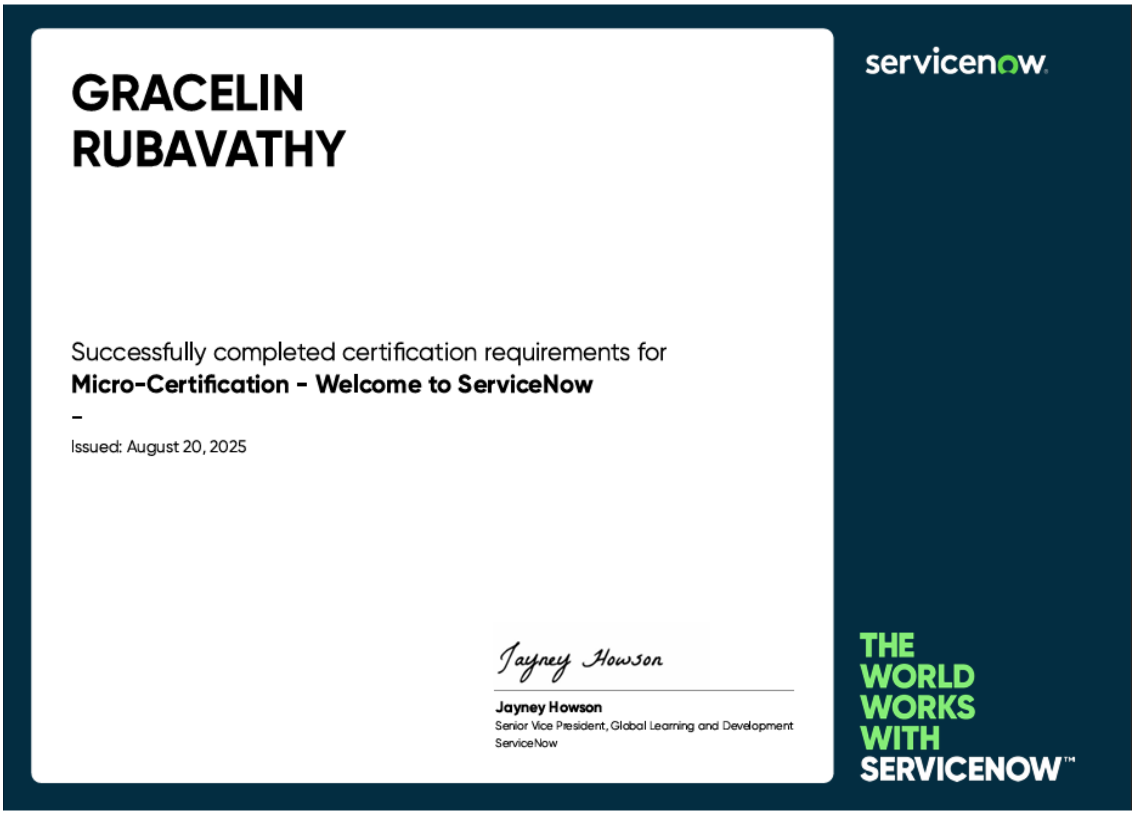
–

Issued: August 20, 2025

Jayney Howson

Jayney Howson
Senior Vice President, Global Learning and Development
ServiceNow

THE WORLD WORKS WITH SERVICENOW™



servicenow.

GRACELIN RUBAVATHY

Successfully completed certification requirements for

Micro-Certification - Welcome to ServiceNow

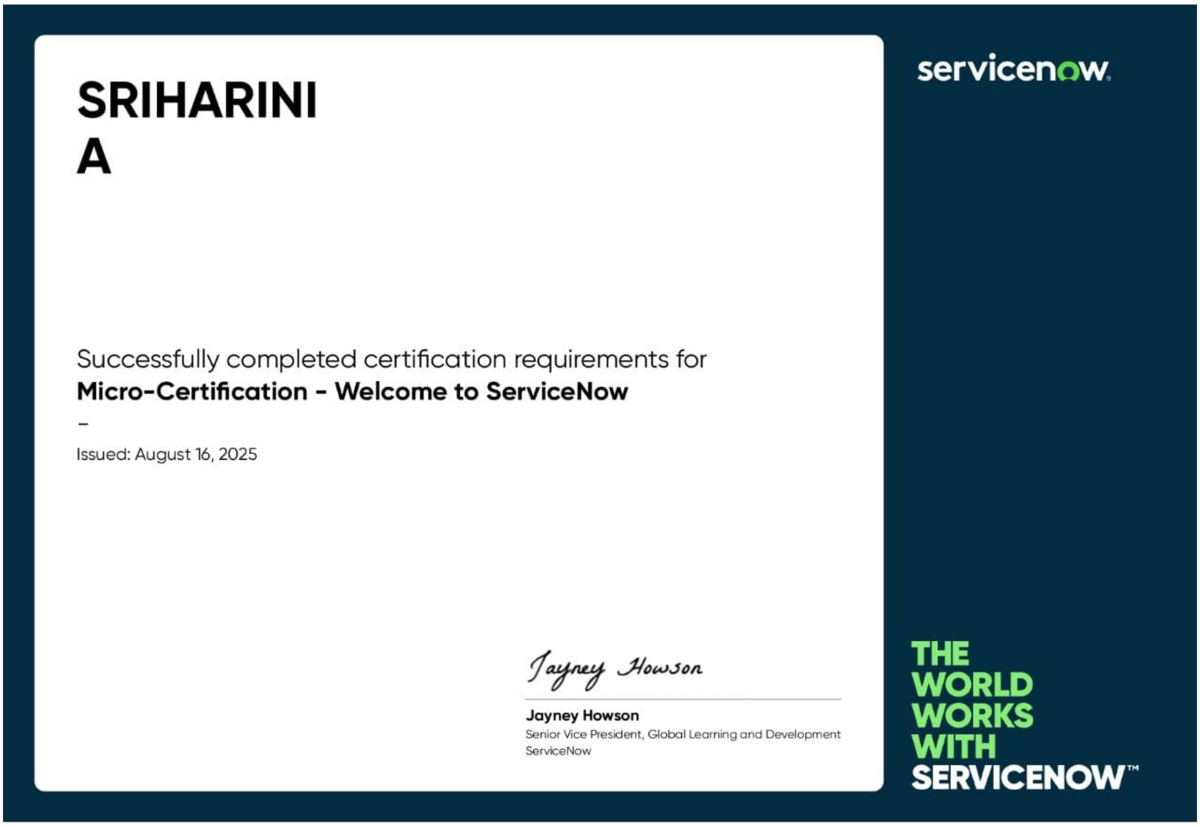
–

Issued: August 20, 2025

Jayney Howson

Jayney Howson
Senior Vice President, Global Learning and Development
ServiceNow

THE WORLD WORKS WITH SERVICENOW™



servicenow

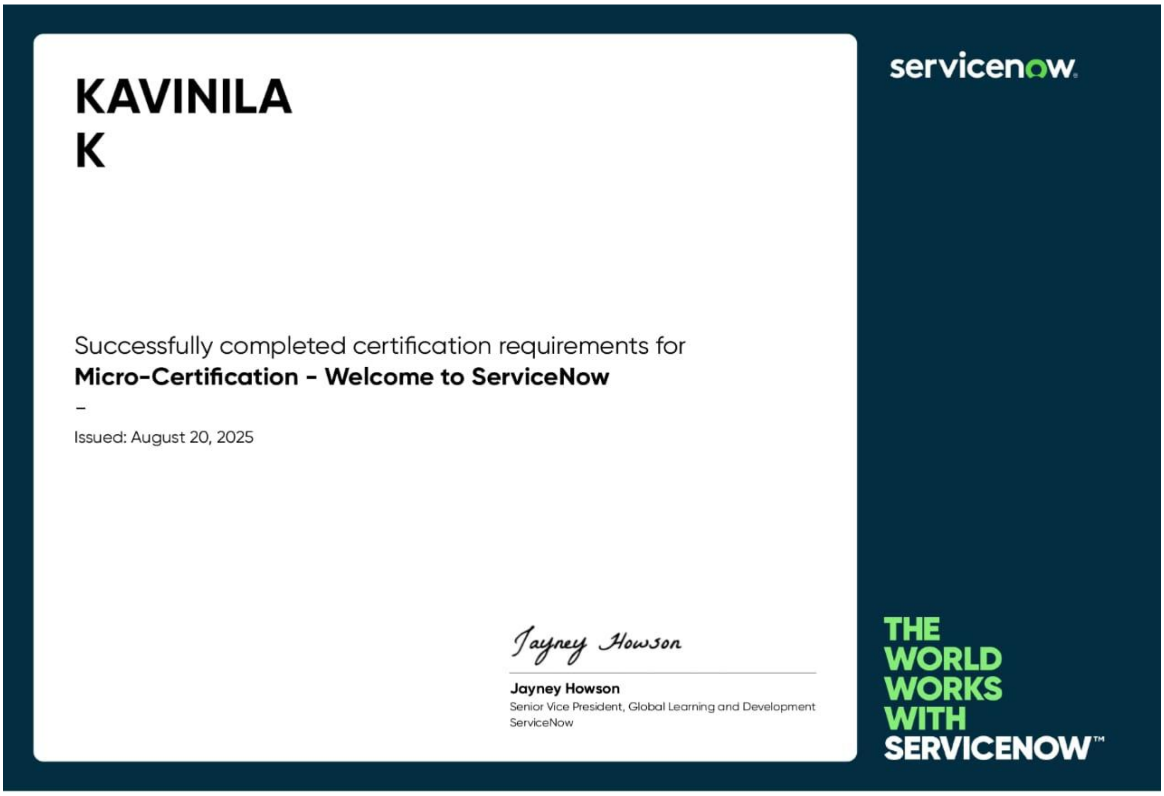
SRIHARINI
A

Successfully completed certification requirements for
Micro-Certification - Welcome to ServiceNow
–
Issued: August 16, 2025

Jayney Howson

Jayney Howson
Senior Vice President, Global Learning and Development
ServiceNow

THE
WORLD
WORKS
WITH
SERVICENOW™



servicenow

KAVINILA
K

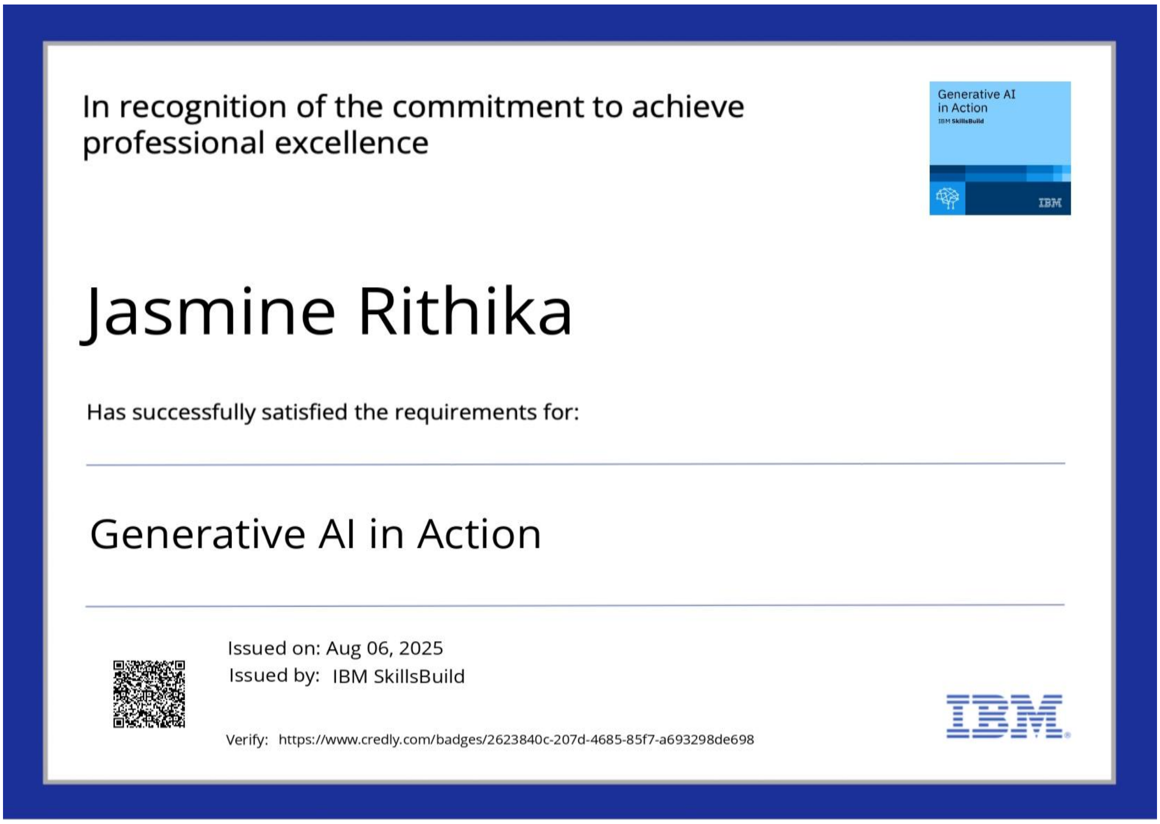
Successfully completed certification requirements for
Micro-Certification - Welcome to ServiceNow
–
Issued: August 20, 2025

Jayney Howson

Jayney Howson
Senior Vice President, Global Learning and Development
ServiceNow

THE
WORLD
WORKS
WITH
SERVICENOW™

2. IBM SkillsBuild – Generative AI in action


Generative AI
in Action
IBM SkillsBuild
IBM
IBM

In recognition of the commitment to achieve professional excellence

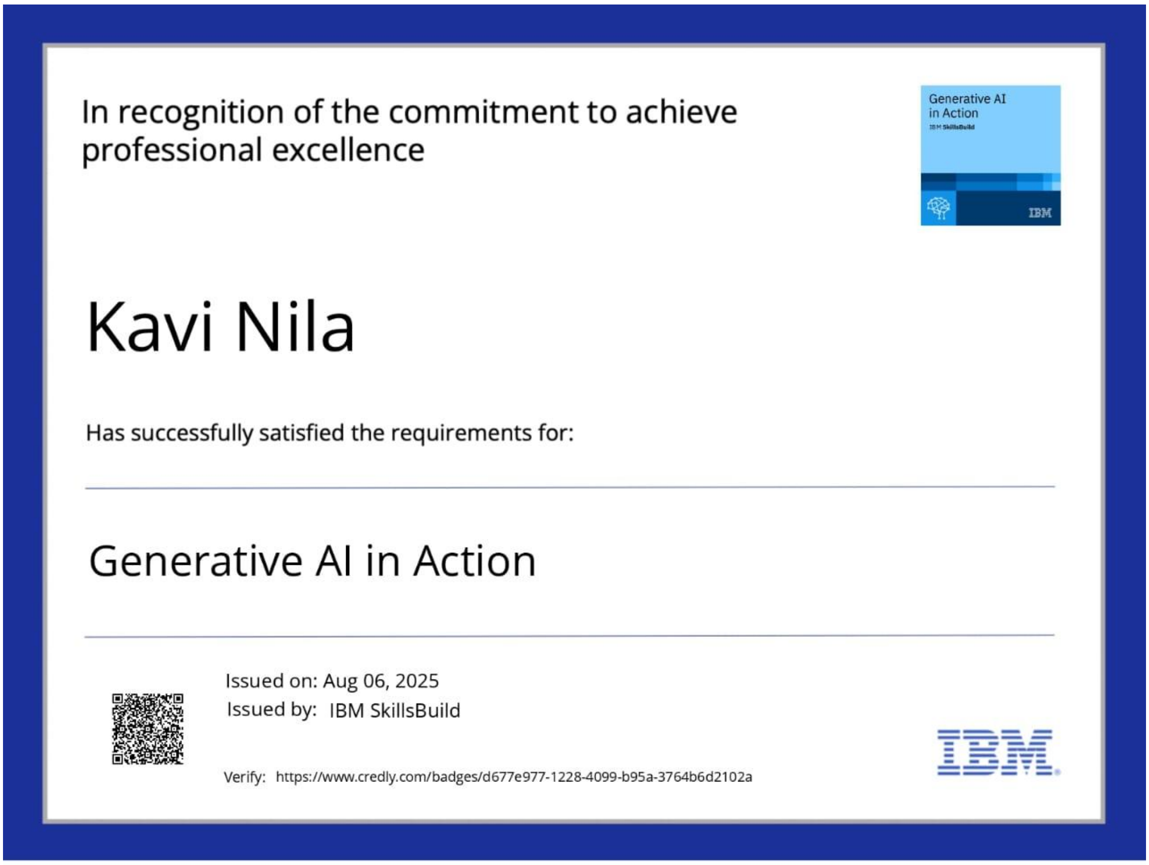
Jasmine Rithika

Has successfully satisfied the requirements for:

Generative AI in Action

Issued on: Aug 06, 2025
Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/2623840c-207d-4685-85f7-a693298de698


Generative AI
in Action
IBM SkillsBuild
IBM
IBM

In recognition of the commitment to achieve professional excellence

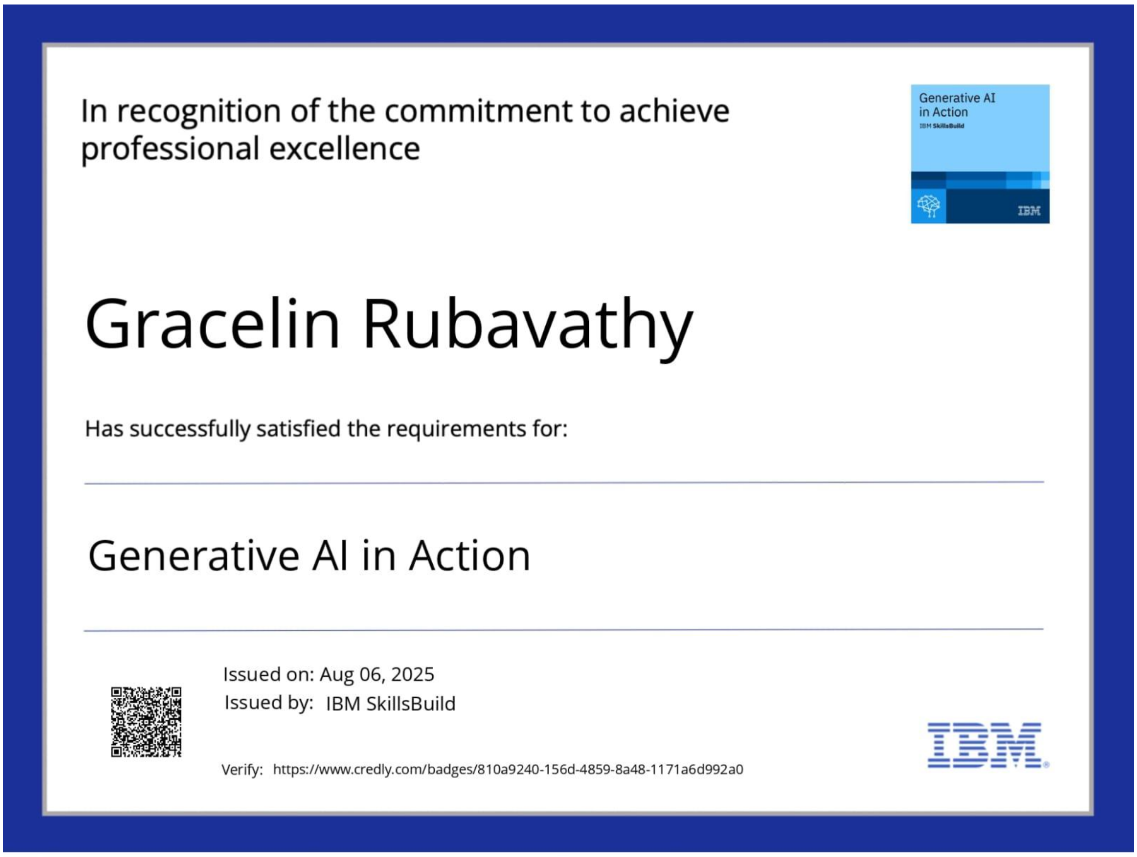
Kavi Nila

Has successfully satisfied the requirements for:

Generative AI in Action

Issued on: Aug 06, 2025
Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/d677e977-1228-4099-b95a-3764b6d2102a



Generative AI
in Action
IBM SkillsBuild
IBM

In recognition of the commitment to achieve professional excellence

Gracelin Rubavathy

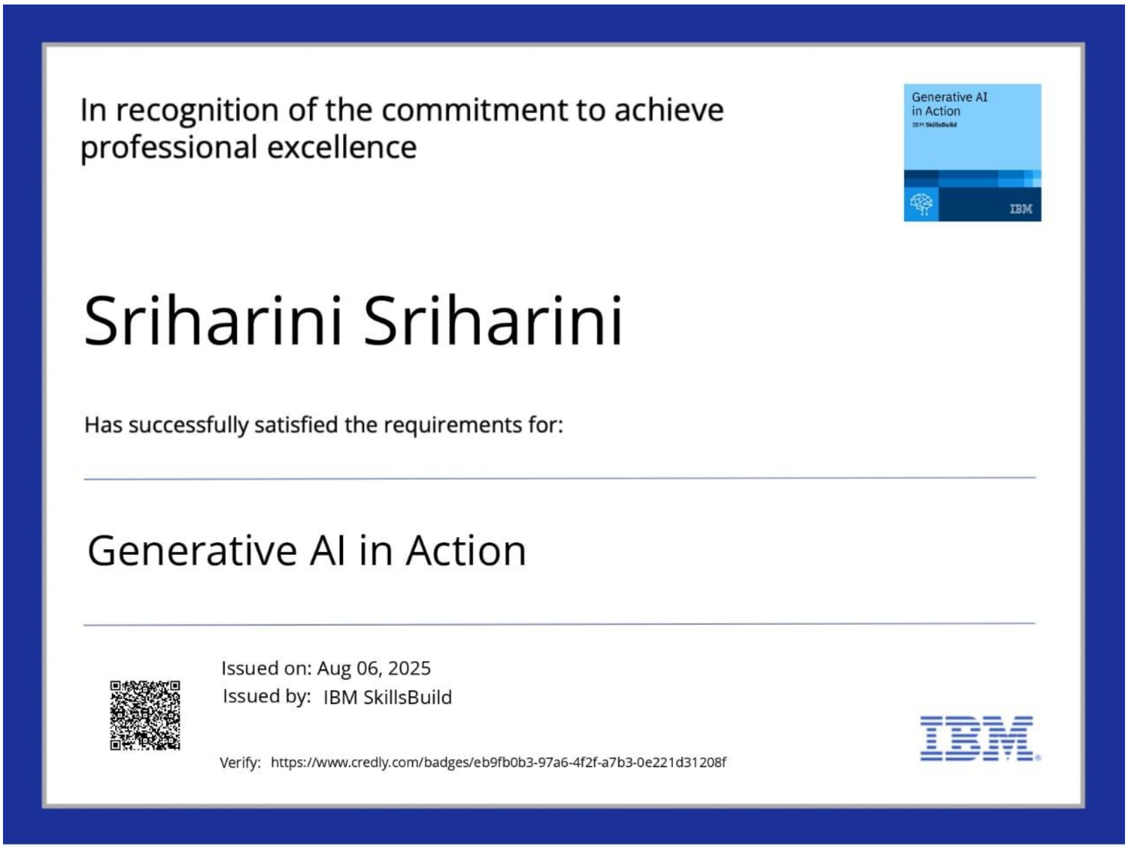
Has successfully satisfied the requirements for:

Generative AI in Action

Issued on: Aug 06, 2025
Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/810a9240-156d-4859-8a48-1171a6d992a0

IBM



Generative AI
in Action
IBM SkillsBuild
IBM

In recognition of the commitment to achieve professional excellence

Sriharini Sriharini

Has successfully satisfied the requirements for:

Generative AI in Action

Issued on: Aug 06, 2025
Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/eb9fb0b3-97a6-4f2f-a7b3-0e221d31208f

IBM