

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

**Mrs. C.SURYA M.E.,
ASSISTANT PROFESSOR**

Department of CSE
Sri Ramakrishna College of Engineering
Perambalur

SIGNATURE

**Mr. R.DINESH RAJ M.E.,
HEAD OF THE DEPARTMENT**

Department of CSE
Sri Ramakrishna College of Engineering
Perambalur

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

INTERNAL EXAMINER

EXTERNAL EXAMINER

TABLE OF CONTENT

S.NO	CONTENTS	PAGE NO
1.	Abstract	4
2.	Problem Definition	4
3.	Introduction	4
4.	Modules and tools used	5
5.	Creation of tables	6
6.	Creation of relationship	7
7.	Creation of business rules	9
8.	Configure the relationship	10
9.	Implementation process	11
10.	Conclusion	12
11.	Practice scenarios for servicenow admin	15

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.

This screenshot shows the ServiceNow interface for managing relationships. The top navigation bar includes 'All', 'Favorites', 'History', 'Workspaces', and a search bar. The main title is 'Relationship - Daily Expenses'. Below the title, there are fields for 'Name' (set to 'Daily Expenses'), 'Advanced' (unchecked), 'Application' (set to 'Global'), 'Applies to table' (set to 'Family Expenses [u_auto_populated]'), and 'Queries from table' (set to 'Daily Expenses [u_daily_expenses]'). A note at the bottom 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.' A code editor window contains the following ECMAScript code:

```

Query with Turn on ECMAScript 2021 (ES12) mode
1 (function refinequery(current, parent) {
2   // Add your code here, such as current.addquery(field, value);
3 }
4 )(current, parent);

```

Below the code editor are buttons for 'Run Query Diagnostics', 'Update', and 'Delete'. A 'Related Links' section includes a 'Run Point Scan' link. The bottom status bar shows system information: ENG UK, 12:19, 04-11-2025.

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.

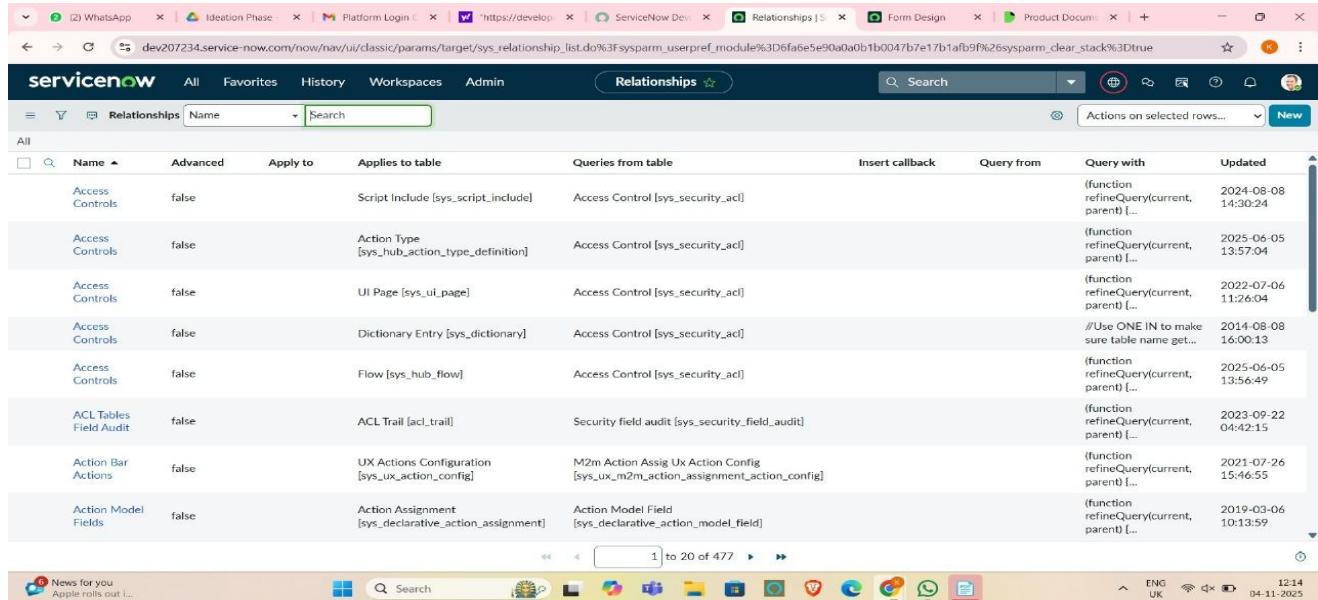
This screenshot shows the ServiceNow 'Tables' page. The top navigation bar includes 'All', 'Favorites', 'History', 'Workspaces', 'Admin', and a search bar. The main title is 'Tables'. Below the title, there is a search bar and a 'Actions on selected rows...' dropdown. A note at the top says: 'All > Update name is not empty'. The main content area displays a table with columns: 'Label', 'Name', 'Extends table', 'Extensible', and 'Updated'. The table lists various application files, many of which have '(empty)' in the 'Extends table' and 'Extensible' columns. The 'Updated' column shows dates ranging from 2025-09-08 to 2025-09-08. At the bottom, there is a pagination bar showing '1 to 20 of 5,765' and a status bar with system information: Trending videos, ENG UK, 11:57, 04-11-2025.

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_ad_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

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.



The screenshot shows the ServiceNow Relationships list page. The table has columns: Name, Advanced, Apply to, Applies to table, Queries from table, Insert callback, Query from, Query with, and Updated. There are 477 rows listed. A vertical scroll bar is on the right. The status bar at the bottom shows '1 to 20 of 477' and the date '04-11-2023'. A small number '7.' is in the top right corner.

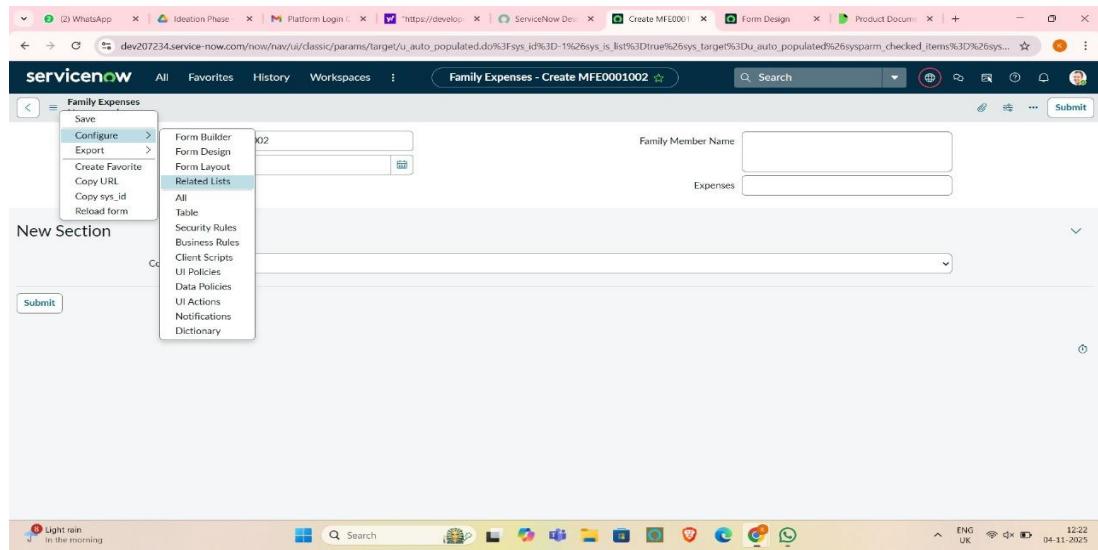
Name	Advanced	Apply to	Applies to table	Queries from table	Insert callback	Query from	Query with	Updated
Access Controls	false		Script Include [sys_script_include]	Access Control [sys_security_acl]	(function refineQuery(current, parent) { ... })		2024-08-08 14:30:24	
Access Controls	false		Action Type [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	

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.

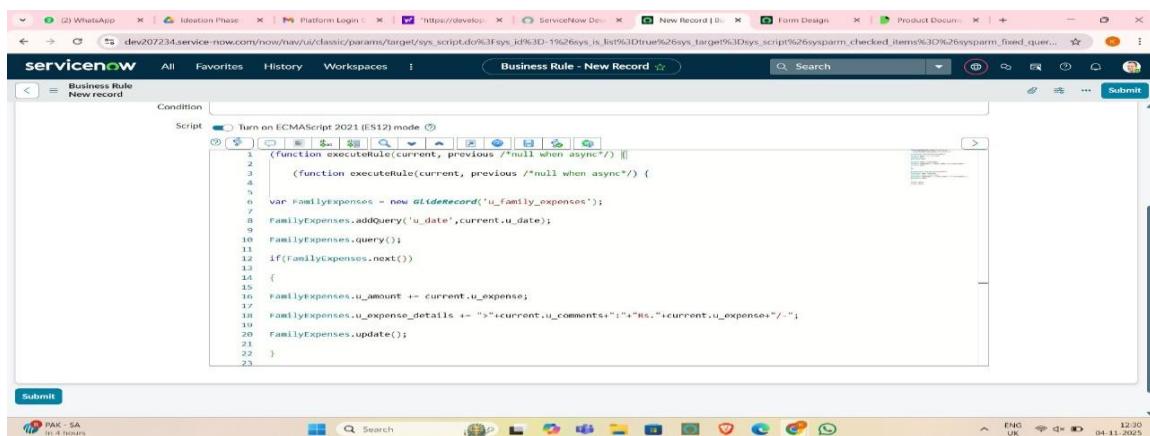


7. CREATION OF BUSINESS RULES

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”).



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.

The screenshot shows the ServiceNow interface for configuring a relationship. The top navigation bar includes links for 'WhatsApp', 'Ideation Phase', 'Platform Login', 'https://develop...', 'ServiceNow Dev...', 'Daily Expenses', 'Form Design', 'Product Document', and 'Relationship - Daily Expenses'. The main content area is titled 'Relationship - Daily Expenses' and shows the following configuration details:

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

A script editor window contains the following ECMAScript 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 } ) (current, parent);  
13  
14 } ) (current, parent);  
15  
16 } ) (current, parent);  
17  
18 } ) (current, parent);  
19  
20 } ) (current, parent);  
21  
22 } ) (current, parent);  
23  
24 } ) (current, parent);  
25  
26 } ) (current, parent);  
27  
28 } ) (current, parent);  
29  
30 } ) (current, parent);  
31  
32 } ) (current, parent);  
33  
34 } ) (current, parent);  
35  
36 } ) (current, parent);  
37  
38 } ) (current, parent);  
39  
40 } ) (current, parent);  
41  
42 } ) (current, parent);  
43  
44 } ) (current, parent);  
45  
46 } ) (current, parent);  
47  
48 } ) (current, parent);  
49  
50 } ) (current, parent);  
51  
52 } ) (current, parent);  
53  
54 } ) (current, parent);  
55  
56 } ) (current, parent);  
57  
58 } ) (current, parent);  
59  
60 } ) (current, parent);  
61  
62 } ) (current, parent);  
63  
64 } ) (current, parent);  
65  
66 } ) (current, parent);  
67  
68 } ) (current, parent);  
69  
70 } ) (current, parent);  
71  
72 } ) (current, parent);  
73  
74 } ) (current, parent);  
75  
76 } ) (current, parent);  
77  
78 } ) (current, parent);  
79  
80 } ) (current, parent);  
81  
82 } ) (current, parent);  
83  
84 } ) (current, parent);  
85  
86 } ) (current, parent);  
87  
88 } ) (current, parent);  
89  
90 } ) (current, parent);  
91  
92 } ) (current, parent);  
93  
94 } ) (current, parent);  
95  
96 } ) (current, parent);  
97  
98 } ) (current, parent);  
99  
100 } ) (current, parent);  
101  
102 } ) (current, parent);  
103  
104 } ) (current, parent);  
105  
106 } ) (current, parent);  
107  
108 } ) (current, parent);  
109  
110 } ) (current, parent);  
111  
112 } ) (current, parent);  
113  
114 } ) (current, parent);  
115  
116 } ) (current, parent);  
117  
118 } ) (current, parent);  
119  
120 } ) (current, parent);  
121  
122 } ) (current, parent);  
123  
124 } ) (current, parent);  
125  
126 } ) (current, parent);  
127  
128 } ) (current, parent);  
129  
130 } ) (current, parent);  
131  
132 } ) (current, parent);  
133  
134 } ) (current, parent);  
135  
136 } ) (current, parent);  
137  
138 } ) (current, parent);  
139  
140 } ) (current, parent);  
141  
142 } ) (current, parent);  
143  
144 } ) (current, parent);  
145  
146 } ) (current, parent);  
147  
148 } ) (current, parent);  
149  
150 } ) (current, parent);  
151  
152 } ) (current, parent);  
153  
154 } ) (current, parent);  
155  
156 } ) (current, parent);  
157  
158 } ) (current, parent);  
159  
160 } ) (current, parent);  
161  
162 } ) (current, parent);  
163  
164 } ) (current, parent);  
165  
166 } ) (current, parent);  
167  
168 } ) (current, parent);  
169  
170 } ) (current, parent);  
171  
172 } ) (current, parent);  
173  
174 } ) (current, parent);  
175  
176 } ) (current, parent);  
177  
178 } ) (current, parent);  
179  
180 } ) (current, parent);  
181  
182 } ) (current, parent);  
183  
184 } ) (current, parent);  
185  
186 } ) (current, parent);  
187  
188 } ) (current, parent);  
189  
190 } ) (current, parent);  
191  
192 } ) (current, parent);  
193  
194 } ) (current, parent);  
195  
196 } ) (current, parent);  
197  
198 } ) (current, parent);  
199  
200 } ) (current, parent);  
201  
202 } ) (current, parent);  
203  
204 } ) (current, parent);  
205  
206 } ) (current, parent);  
207  
208 } ) (current, parent);  
209  
210 } ) (current, parent);  
211  
212 } ) (current, parent);  
213  
214 } ) (current, parent);  
215  
216 } ) (current, parent);  
217  
218 } ) (current, parent);  
219  
220 } ) (current, parent);  
221  
222 } ) (current, parent);  
223  
224 } ) (current, parent);  
225  
226 } ) (current, parent);  
227  
228 } ) (current, parent);  
229  
230 } ) (current, parent);  
231  
232 } ) (current, parent);  
233  
234 } ) (current, parent);  
235  
236 } ) (current, parent);  
237  
238 } ) (current, parent);  
239  
240 } ) (current, parent);  
241  
242 } ) (current, parent);  
243  
244 } ) (current, parent);  
245  
246 } ) (current, parent);  
247  
248 } ) (current, parent);  
249  
250 } ) (current, parent);  
251  
252 } ) (current, parent);  
253  
254 } ) (current, parent);  
255  
256 } ) (current, parent);  
257  
258 } ) (current, parent);  
259  
260 } ) (current, parent);  
261  
262 } ) (current, parent);  
263  
264 } ) (current, parent);  
265  
266 } ) (current, parent);  
267  
268 } ) (current, parent);  
269  
270 } ) (current, parent);  
271  
272 } ) (current, parent);  
273  
274 } ) (current, parent);  
275  
276 } ) (current, parent);  
277  
278 } ) (current, parent);  
279  
280 } ) (current, parent);  
281  
282 } ) (current, parent);  
283  
284 } ) (current, parent);  
285  
286 } ) (current, parent);  
287  
288 } ) (current, parent);  
289  
290 } ) (current, parent);  
291  
292 } ) (current, parent);  
293  
294 } ) (current, parent);  
295  
296 } ) (current, parent);  
297  
298 } ) (current, parent);  
299  
300 } ) (current, parent);  
301  
302 } ) (current, parent);  
303  
304 } ) (current, parent);  
305  
306 } ) (current, parent);  
307  
308 } ) (current, parent);  
309  
310 } ) (current, parent);  
311  
312 } ) (current, parent);  
313  
314 } ) (current, parent);  
315  
316 } ) (current, parent);  
317  
318 } ) (current, parent);  
319  
320 } ) (current, parent);  
321  
322 } ) (current, parent);  
323  
324 } ) (current, parent);  
325  
326 } ) (current, parent);  
327  
328 } ) (current, parent);  
329  
330 } ) (current, parent);  
331  
332 } ) (current, parent);  
333  
334 } ) (current, parent);  
335  
336 } ) (current, parent);  
337  
338 } ) (current, parent);  
339  
340 } ) (current, parent);  
341  
342 } ) (current, parent);  
343  
344 } ) (current, parent);  
345  
346 } ) (current, parent);  
347  
348 } ) (current, parent);  
349  
350 } ) (current, parent);  
351  
352 } ) (current, parent);  
353  
354 } ) (current, parent);  
355  
356 } ) (current, parent);  
357  
358 } ) (current, parent);  
359  
360 } ) (current, parent);  
361  
362 } ) (current, parent);  
363  
364 } ) (current, parent);  
365  
366 } ) (current, parent);  
367  
368 } ) (current, parent);  
369  
370 } ) (current, parent);  
371  
372 } ) (current, parent);  
373  
374 } ) (current, parent);  
375  
376 } ) (current, parent);  
377  
378 } ) (current, parent);  
379  
380 } ) (current, parent);  
381  
382 } ) (current, parent);  
383  
384 } ) (current, parent);  
385  
386 } ) (current, parent);  
387  
388 } ) (current, parent);  
389  
390 } ) (current, parent);  
391  
392 } ) (current, parent);  
393  
394 } ) (current, parent);  
395  
396 } ) (current, parent);  
397  
398 } ) (current, parent);  
399  
400 } ) (current, parent);  
401  
402 } ) (current, parent);  
403  
404 } ) (current, parent);  
405  
406 } ) (current, parent);  
407  
408 } ) (current, parent);  
409  
410 } ) (current, parent);  
411  
412 } ) (current, parent);  
413  
414 } ) (current, parent);  
415  
416 } ) (current, parent);  
417  
418 } ) (current, parent);  
419  
420 } ) (current, parent);  
421  
422 } ) (current, parent);  
423  
424 } ) (current, parent);  
425  
426 } ) (current, parent);  
427  
428 } ) (current, parent);  
429  
430 } ) (current, parent);  
431  
432 } ) (current, parent);  
433  
434 } ) (current, parent);  
435  
436 } ) (current, parent);  
437  
438 } ) (current, parent);  
439  
440 } ) (current, parent);  
441  
442 } ) (current, parent);  
443  
444 } ) (current, parent);  
445  
446 } ) (current, parent);  
447  
448 } ) (current, parent);  
449  
450 } ) (current, parent);  
451  
452 } ) (current, parent);  
453  
454 } ) (current, parent);  
455  
456 } ) (current, parent);  
457  
458 } ) (current, parent);  
459  
460 } ) (current, parent);  
461  
462 } ) (current, parent);  
463  
464 } ) (current, parent);  
465  
466 } ) (current, parent);  
467  
468 } ) (current, parent);  
469  
470 } ) (current, parent);  
471  
472 } ) (current, parent);  
473  
474 } ) (current, parent);  
475  
476 } ) (current, parent);  
477  
478 } ) (current, parent);  
479  
480 } ) (current, parent);  
481  
482 } ) (current, parent);  
483  
484 } ) (current, parent);  
485  
486 } ) (current, parent);  
487  
488 } ) (current, parent);  
489  
490 } ) (current, parent);  
491  
492 } ) (current, parent);  
493  
494 } ) (current, parent);  
495  
496 } ) (current, parent);  
497  
498 } ) (current, parent);  
499  
500 } ) (current, parent);  
501  
502 } ) (current, parent);  
503  
504 } ) (current, parent);  
505  
506 } ) (current, parent);  
507  
508 } ) (current, parent);  
509  
510 } ) (current, parent);  
511  
512 } ) (current, parent);  
513  
514 } ) (current, parent);  
515  
516 } ) (current, parent);  
517  
518 } ) (current, parent);  
519  
520 } ) (current, parent);  
521  
522 } ) (current, parent);  
523  
524 } ) (current, parent);  
525  
526 } ) (current, parent);  
527  
528 } ) (current, parent);  
529  
530 } ) (current, parent);  
531  
532 } ) (current, parent);  
533  
534 } ) (current, parent);  
535  
536 } ) (current, parent);  
537  
538 } ) (current, parent);  
539  
540 } ) (current, parent);  
541  
542 } ) (current, parent);  
543  
544 } ) (current, parent);  
545  
546 } ) (current, parent);  
547  
548 } ) (current, parent);  
549  
550 } ) (current, parent);  
551  
552 } ) (current, parent);  
553  
554 } ) (current, parent);  
555  
556 } ) (current, parent);  
557  
558 } ) (current, parent);  
559  
560 } ) (current, parent);  
561  
562 } ) (current, parent);  
563  
564 } ) (current, parent);  
565  
566 } ) (current, parent);  
567  
568 } ) (current, parent);  
569  
570 } ) (current, parent);  
571  
572 } ) (current, parent);  
573  
574 } ) (current, parent);  
575  
576 } ) (current, parent);  
577  
578 } ) (current, parent);  
579  
580 } ) (current, parent);  
581  
582 } ) (current, parent);  
583  
584 } ) (current, parent);  
585  
586 } ) (current, parent);  
587  
588 } ) (current, parent);  
589  
590 } ) (current, parent);  
591  
592 } ) (current, parent);  
593  
594 } ) (current, parent);  
595  
596 } ) (current, parent);  
597  
598 } ) (current, parent);  
599  
600 } ) (current, parent);  
601  
602 } ) (current, parent);  
603  
604 } ) (current, parent);  
605  
606 } ) (current, parent);  
607  
608 } ) (current, parent);  
609  
610 } ) (current, parent);  
611  
612 } ) (current, parent);  
613  
614 } ) (current, parent);  
615  
616 } ) (current, parent);  
617  
618 } ) (current, parent);  
619  
620 } ) (current, parent);  
621  
622 } ) (current, parent);  
623  
624 } ) (current, parent);  
625  
626 } ) (current, parent);  
627  
628 } ) (current, parent);  
629  
630 } ) (current, parent);  
631  
632 } ) (current, parent);  
633  
634 } ) (current, parent);  
635  
636 } ) (current, parent);  
637  
638 } ) (current, parent);  
639  
640 } ) (current, parent);  
641  
642 } ) (current, parent);  
643  
644 } ) (current, parent);  
645  
646 } ) (current, parent);  
647  
648 } ) (current, parent);  
649  
650 } ) (current, parent);  
651  
652 } ) (current, parent);  
653  
654 } ) (current, parent);  
655  
656 } ) (current, parent);  
657  
658 } ) (current, parent);  
659  
660 } ) (current, parent);  
661  
662 } ) (current, parent);  
663  
664 } ) (current, parent);  
665  
666 } ) (current, parent);  
667  
668 } ) (current, parent);  
669  
670 } ) (current, parent);  
671  
672 } ) (current, parent);  
673  
674 } ) (current, parent);  
675  
676 } ) (current, parent);  
677  
678 } ) (current, parent);  
679  
680 } ) (current, parent);  
681  
682 } ) (current, parent);  
683  
684 } ) (current, parent);  
685  
686 } ) (current, parent);  
687  
688 } ) (current, parent);  
689  
690 } ) (current, parent);  
691  
692 } ) (current, parent);  
693  
694 } ) (current, parent);  
695  
696 } ) (current, parent);  
697  
698 } ) (current, parent);  
699  
700 } ) (current, parent);  
701  
702 } ) (current, parent);  
703  
704 } ) (current, parent);  
705  
706 } ) (current, parent);  
707  
708 } ) (current, parent);  
709  
710 } ) (current, parent);  
711  
712 } ) (current, parent);  
713  
714 } ) (current, parent);  
715  
716 } ) (current, parent);  
717  
718 } ) (current, parent);  
719  
720 } ) (current, parent);  
721  
722 } ) (current, parent);  
723  
724 } ) (current, parent);  
725  
726 } ) (current, parent);  
727  
728 } ) (current, parent);  
729  
730 } ) (current, parent);  
731  
732 } ) (current, parent);  
733  
734 } ) (current, parent);  
735  
736 } ) (current, parent);  
737  
738 } ) (current, parent);  
739  
740 } ) (current, parent);  
741  
742 } ) (current, parent);  
743  
744 } ) (current, parent);  
745  
746 } ) (current, parent);  
747  
748 } ) (current, parent);  
749  
750 } ) (current, parent);  
751  
752 } ) (current, parent);  
753  
754 } ) (current, parent);  
755  
756 } ) (current, parent);  
757  
758 } ) (current, parent);  
759  
760 } ) (current, parent);  
761  
762 } ) (current, parent);  
763  
764 } ) (current, parent);  
765  
766 } ) (current, parent);  
767  
768 } ) (current, parent);  
769  
770 } ) (current, parent);  
771  
772 } ) (current, parent);  
773  
774 } ) (current, parent);  
775  
776 } ) (current, parent);  
777  
778 } ) (current, parent);  
779  
780 } ) (current, parent);  
781  
782 } ) (current, parent);  
783  
784 } ) (current, parent);  
785  
786 } ) (current, parent);  
787  
788 } ) (current, parent);  
789  
790 } ) (current, parent);  
791  
792 } ) (current, parent);  
793  
794 } ) (current, parent);  
795  
796 } ) (current, parent);  
797  
798 } ) (current, parent);  
799  
800 } ) (current, parent);  
801  
802 } ) (current, parent);  
803  
804 } ) (current, parent);  
805  
806 } ) (current, parent);  
807  
808 } ) (current, parent);  
809  
810 } ) (current, parent);  
811  
812 } ) (current, parent);  
813  
814 } ) (current, parent);  
815  
816 } ) (current, parent);  
817  
818 } ) (current, parent);  
819  
820 } ) (current, parent);  
821  
822 } ) (current, parent);  
823  
824 } ) (current, parent);  
825  
826 } ) (current, parent);  
827  
828 } ) (current, parent);  
829  
830 } ) (current, parent);  
831  
832 } ) (current, parent);  
833  
834 } ) (current, parent);  
835  
836 } ) (current, parent);  
837  
838 } ) (current, parent);  
839  
840 } ) (current, parent);  
841  
842 } ) (current, parent);  
843  
844 } ) (current, parent);  
845  
846 } ) (current, parent);  
847  
848 } ) (current, parent);  
849  
850 } ) (current, parent);  
851  
852 } ) (current, parent);  
853  
854 } ) (current, parent);  
855  
856 } ) (current, parent);  
857  
858 } ) (current, parent);  
859  
860 } ) (current, parent);  
861  
862 } ) (current, parent);  
863  
864 } ) (current, parent);  
865  
866 } ) (current, parent);  
867  
868 } ) (current, parent);  
869  
870 } ) (current, parent);  
871  
872 } ) (current, parent);  
873  
874 } ) (current, parent);  
875  
876 } ) (current, parent);  
877  
878 } ) (current, parent);  
879  
880 } ) (current, parent);  
881  
882 } ) (current, parent);  
883  
884 } ) (current, parent);  
885  
886 } ) (current, parent);  
887  
888 } ) (current, parent);  
889  
890 } ) (current, parent);  
891  
892 } ) (current, parent);  
893  
894 } ) (current, parent);  
895  
896 } ) (current, parent);  
897  
898 } ) (current, parent);  
899  
900 } ) (current, parent);  
901  
902 } ) (current, parent);  
903  
904 } ) (current, parent);  
905  
906 } ) (current, parent);  
907  
908 } ) (current, parent);  
909  
910 } ) (current, parent);  
911  
912 } ) (current, parent);  
913  
914 } ) (current, parent);  
915  
916 } ) (current, parent);  
917  
918 } ) (current, parent);  
919  
920 } ) (current, parent);  
921  
922 } ) (current, parent);  
923  
924 } ) (current, parent);  
925  
926 } ) (current, parent);  
927  
928 } ) (current, parent);  
929  
930 } ) (current, parent);  
931  
932 } ) (current, parent);  
933  
934 } ) (current, parent);  
935  
936 } ) (current, parent);  
937  
938 } ) (current, parent);  
939  
940 } ) (current, parent);  
941  
942 } ) (current, parent);  
943  
944 } ) (current, parent);  
945  
946 } ) (current, parent);  
947  
948 } ) (current, parent);  
949  
950 } ) (current, parent);  
951  
952 } ) (current, parent);  
953  
954 } ) (current, parent);  
955  
956 } ) (current, parent);  
957  
958 } ) (current, parent);  
959  
960 } ) (current, parent);  
961  
962 } ) (current, parent);  
963  
964 } ) (current, parent);  
965  
966 } ) (current, parent);  
967  
968 } ) (current, parent);  
969  
970 } ) (current, parent);  
971  
972 } ) (current, parent);  
973  
974 } ) (current, parent);  
975  
976 } ) (current, parent);  
977  
978 } ) (current, parent);  
979  
980 } ) (current, parent);  
981  
982 } ) (current, parent);  
983  
984 } ) (current, parent);  
985  
986 } ) (current, parent);  
987  
988 } ) (current, parent);  
989  
990 } ) (current, parent);  
991  
992 } ) (current, parent);  
993  
994 } ) (current, parent);  
995  
996 } ) (current, parent);  
997  
998 } ) (current, parent);  
999  
1000 } ) (current, parent);  
1001  
1002 } ) (current, parent);  
1003  
1004 } ) (current, parent);  
1005  
1006 } ) (current, parent);  
1007  
1008 } ) (current, parent);  
1009  
1010 } ) (current, parent);  
1011  
1012 } ) (current, parent);  
1013  
1014 } ) (current, parent);  
1015  
1016 } ) (current, parent);  
1017  
1018 } ) (current, parent);  
1019  
1020 } ) (current, parent);  
1021  
1022 } ) (current, parent);  
1023  
1024 } ) (current, parent);  
1025  
1026 } ) (current, parent);  
1027  
1028 } ) (current, parent);  
1029  
1030 } ) (current, parent);  
1031  
1032 } ) (current, parent);  
1033  
1034 } ) (current, parent);  
1035  
1036 } ) (current, parent);  
1037  
1038 } ) (current, parent);  
1039  
1040 } ) (current, parent);  
1041  
1042 } ) (current, parent);  
1043  
1044 } ) (current, parent);  
1045  
1046 } ) (current, parent);  
1047  
1048 } ) (current, parent);  
1049  
1050 } ) (current, parent);  
1051  
1052 } ) (current, parent);  
1053  
1054 } ) (current, parent);  
1055  
1056 } ) (current, parent);  
1057  
1058 } ) (current, parent);  
1059  
1060 } ) (current, parent);  
1061  
1062 } ) (current, parent);  
1063  
1064 } ) (current, parent);  
1065  
1066 } ) (current, parent);  
1067  
1068 } ) (current, parent);  
1069  
1070 } ) (current, parent);  
1071  
1072 } ) (current, parent);  
1073  
1074 } ) (current, parent);  
1075  
1076 } ) (current, parent);  
1077  
1078 } ) (current, parent);  
1079  
1080 } ) (current, parent);  
1081  
1082 } ) (current, parent);  
1083  
1084 } ) (current, parent);  
1085  
1086 } ) (current, parent);  
1087  
1088 } ) (current, parent);  
1089  
1090 } ) (current, parent);  
1091  
1092 } ) (current, parent);  
1093  
1094 } ) (current, parent);  
1095  
1096 } ) (current, parent);  
1097  
1098 } ) (current, parent);  
1099  
1100 } ) (current, parent);  
1101  
1102 } ) (current, parent);  
1103  
1104 } ) (current, parent);  
1105  
1106 } ) (current, parent);  
1107  
1108 } ) (current, parent);  
1109  
1110 } ) (current, parent);  
1111  
1112 } ) (current, parent);  
1113  
1114 } ) (current, parent);  
1115  
1116 } ) (current, parent);  
1117  
1118 } ) (current, parent);  
1119  
1120 } ) (current, parent);  
1121  
1122 } ) (current, parent);  
1123  
1124 } ) (current, parent);  
1125  
1126 } ) (current, parent);  
1127  
1128 } ) (current, parent);  
1129  
1130 } ) (current, parent);  
1131  
1132 } ) (current, parent);  
1133  
1134 } ) (current, parent);  
1135  
1136 } ) (current, parent);  
1137  
1138 } ) (current, parent);  
1139  
1140 } ) (current, parent);  
1141  
1142 } ) (current, parent);  
1143  
1144 } ) (current, parent);  
1145  
1146 } ) (current, parent);  
1147  
1148 } ) (current, parent);  
1149  
1150 } ) (current, parent);  
1151  
1152 } ) (current, parent);  
1153  
1154 } ) (current, parent);  
1155  
1156 } ) (current, parent);  
1157  
1158 } ) (current, parent);  
1159  
1160 } ) (current, parent);  
1161  
1162 } ) (current, parent);  
1163  
1164 } ) (current, parent);  
1165  
1166 } ) (current, parent);  
1167  
1168 } ) (current, parent);  
1169  
1170 } ) (current, parent);  
1171  
1172
```

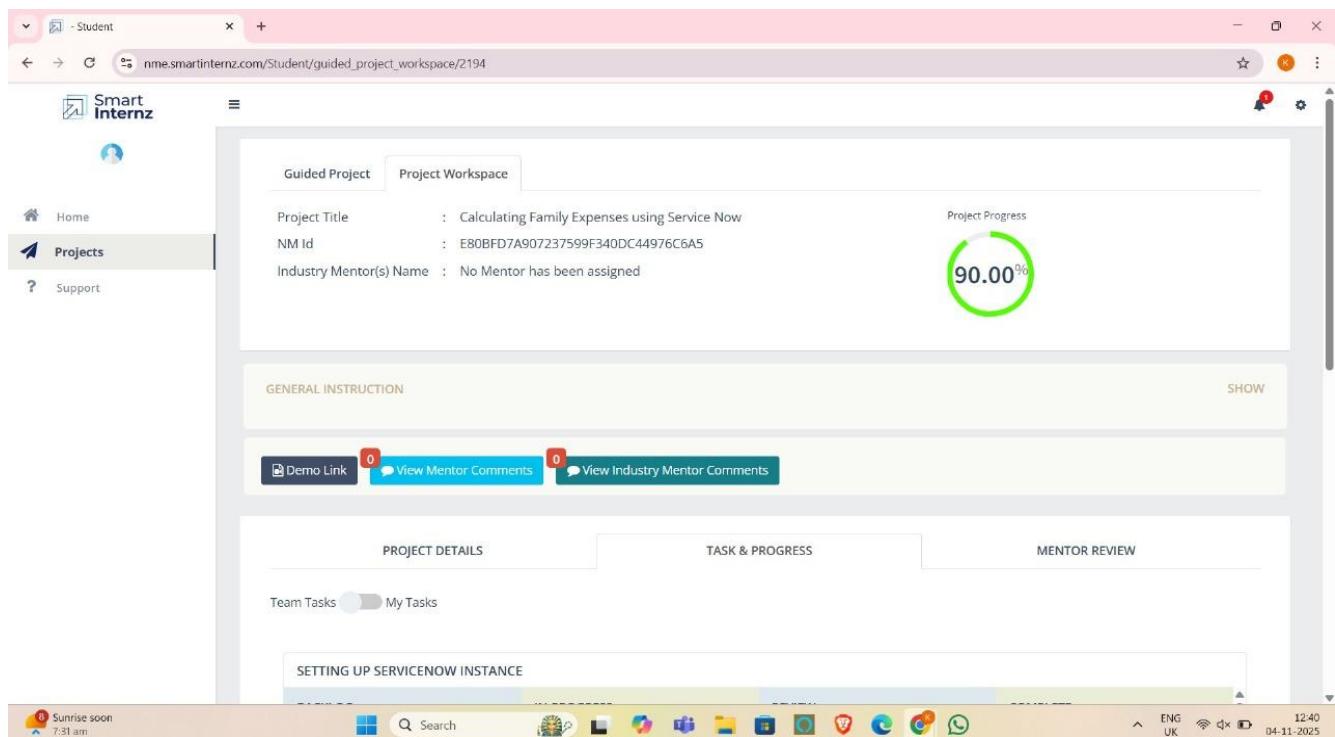
9. IMPLEMENTATION PROCESS

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.



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.

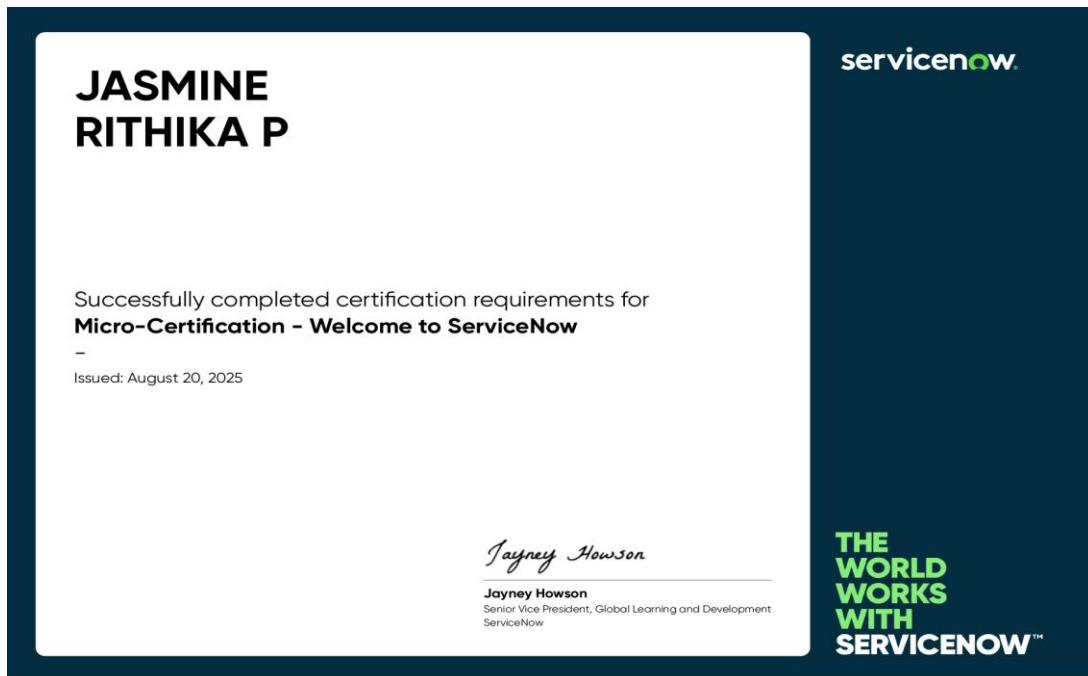
- **Solution:**
 - Navigate: System UI → UI Actions → New
 - Settings:
 - Name: Copy Short Description
 - Table: Incident
 - Action type: Form button
 - Active: checked
 - Script:
 - `current.description = current.short_description;`
 - `current.update();`
 - `action.setRedirectURL(current);`

Source Code

https://github.com/Kavimagizh19/family_expenses

1. CERTIFICATIONS

1. Micro-Certification - Welcome to ServiceNow



SRIHARINI A

servicenow.

Successfully completed certification requirements for
Micro-Certification - Welcome to ServiceNow

—
Issued: August 16, 2025

Jayne Howson

Jayne Howson
Senior Vice President, Global Learning and Development
ServiceNow

THE
WORLD
WORKS
WITH
SERVICENOW™

KAVINILA K

servicenow.

Successfully completed certification requirements for
Micro-Certification - Welcome to ServiceNow

—
Issued: August 20, 2025

Jayne Howson

Jayne Howson
Senior Vice President, Global Learning and Development
ServiceNow

THE
WORLD
WORKS
WITH
SERVICENOW™

2. IBM SkillsBuild – Generative AI in action



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>



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>

