

## 1. **Introduction**

- Project Title: Prevent User Deletion in ServiceNow

- Team Members:

Team Leader: Nikhil Reddy – Backend Logic & Documentation

Team Member: Ajay Pydi – Testing & Validation

Team Member: Pulikonda Chandu – Research & Analysis

Team Member: Karthikeyan Reddy Musali – Design & Diagrams

Team Member: Venkata Datta Chandan Sure – Implementation Support

## 2. **Project Overview**

- Purpose:

The goal of this project is to prevent accidental or unauthorized deletion of user records in ServiceNow by implementing validation logic using Business Rules and role-based access control to ensure data integrity and system reliability.

- Features:

- Prevention of user deletion if linked to records (e.g., incidents)
- Server-side Business Rule validation
- Role-based access restriction
- Error message alerts on delete attempts
- Secure user management within ServiceNow

## 3. **Architecture**

- Frontend:

ServiceNow Web UI (Admin dashboard and user management interface).

- Backend:

ServiceNow server-side scripting (Business Rules using JavaScript) to intercept and validate delete operations.

- Database:

ServiceNow internal database (sys\_user table, incident table) used for storing user and assignment data.

#### 4. **Setup Instructions**

- Prerequisites:
  - ServiceNow Developer Instance
  - Web Browser (Chrome/Edge)
  - Basic knowledge of ServiceNow platform
  - Installation:
    1. Log in to ServiceNow Developer Instance
    2. Navigate to System Definition → Business Rules
    3. Create a Business Rule on “sys\_user” table
    4. Add script to block deletion if user is assigned to records
- 1. Save and test the rule in user module
- 2. Folder Structure
  - Client:  
Not applicable (uses built-in ServiceNow UI)
  - Server:  
Business Rule Script (Deletion Prevention Logic)  
Documentation Files (Reports, Templates, Diagrams)
- 3. Running the Application
  - Open ServiceNow instance
  - Navigate to User (sys\_user) table
  - Attempt to delete a user record

- System automatically triggers Business Rule validation

(No npm or local server required)

#### 4. API Documentation

This project does not expose external REST APIs.

All operations are handled internally within the ServiceNow platform using server-side scripting and database tables.

#### 5. **Authentication**

Authentication and authorization are handled using ServiceNow Role-Based Access Control (RBAC).

Only authorized roles (Admin) can manage users, and additional Business Rules enforce deletion restrictions at the server level.

#### 6. **User Interface**

The user interface is the default ServiceNow dashboard, including:

- User Management Module
- Incident Assignment Screen
- Error Alert Messages during deletion attempts

#### 10. **Testing**

Testing Strategy:

- Manual Functional Testing
- Negative Testing (delete assigned user)
- Positive Testing (delete unassigned user)

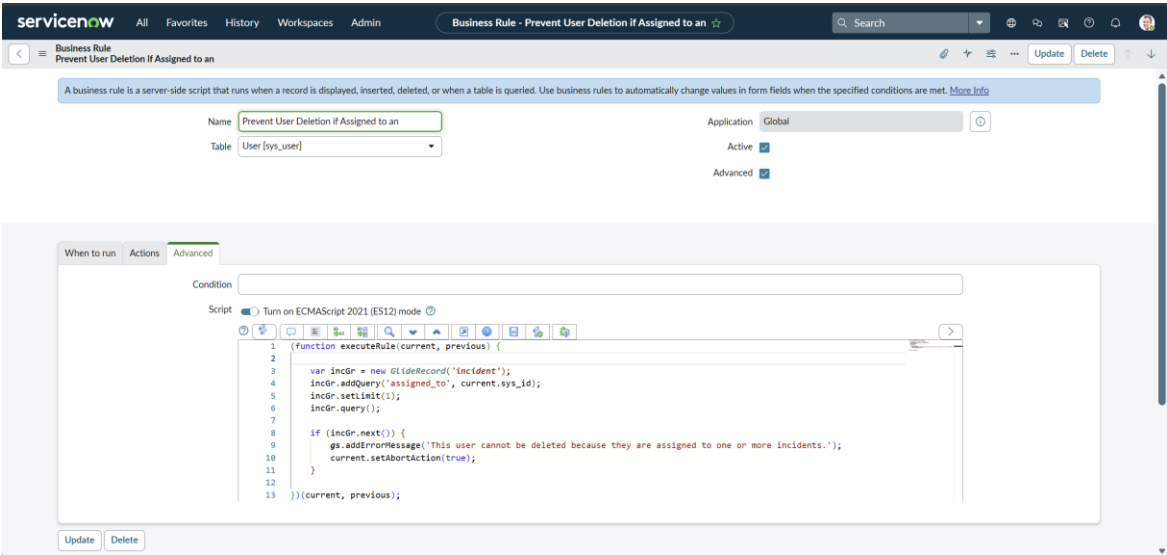
Tools Used:

- ServiceNow Platform
- Built-in testing environment

## 11. Screenshots or Demo

Include screenshots of:

- Business Rule configuration



- User creation

The screenshot shows the ServiceNow Users list page. The table displays a list of users with columns for User ID, Name, Email, Active, Created, and Updated. The table is sorted by Name. The first 20 users are listed, and the total count is 634.

User ID	Name	Email	Active	Created	Updated
abel.tuter	Abel Tuter	abel.tuter@example.com	true	2012-02-17 19:04:52	2026-01-28 00:27:19
abraham.lincoln	Abraham Lincoln	abraham.lincoln@example.com	true	2013-07-23 17:15:54	2026-01-28 00:27:20
adela.cervantsz	Adela Cervantsz	adela.cervantsz@example.com	true	2012-02-17 19:04:50	2026-01-28 00:27:16
alileen.mottern	Alleen Mottern	alileen.mottern@example.com	true	2012-02-17 19:04:49	2026-01-28 00:27:19
ajay123	Ajay Pydi	ajay.pydi@example.com	true	2026-01-28 21:05:14	2026-01-28 21:05:14
akhil123	Akhil Reddy	akhil.reddy@example.com	true	2026-01-28 22:01:33	2026-01-28 22:01:33
alejandra.prenatt	Alejandra Prenatt	alejandra.prenatt@example.com	true	2012-02-17 19:04:52	2026-01-28 00:27:16
alejandro.mascall	Alejandro Mascall	alejandro.mascall@example.com	true	2012-02-17 19:04:52	2026-01-28 00:27:20
alene.rabeck	Alene Rabeck	alene.rabeck@example.com	true	2012-02-17 19:04:53	2026-01-28 00:27:21
alfonso.griglen	Alfonso Griglen	alfonso.griglen@example.com	true	2012-02-17 19:04:51	2026-01-28 00:27:16
alissa.mountjoy	Alissa Mountjoy	alissa.mountjoy@example.com	true	2012-02-17 19:04:52	2026-01-28 00:27:19
allan.schwandt	Allan Schwandt	allan.schwandt@example.com	true	2012-02-17 19:04:53	2026-01-28 00:27:20
allie.pumphrey	Allie Pumphrey	allie.pumphrey@example.com	true	2012-02-17 19:04:52	2026-01-28 00:27:20
allyson.gillisple	Allyson Gillisple	allyson.gillisple@example.com	true	2012-02-17 19:04:50	2026-01-28 00:27:16
alva.pennigton	Alva Pennigton	alva.pennigton@example.com	true	2012-02-17 19:04:50	2026-01-28 00:27:21
alyssa.biasotti	Alyssa Biasotti	alyssa.biasotti@example.com	true	2012-02-17 19:04:52	2026-01-28 00:27:16
amelia.caputo	Amelia Caputo	amelia.caputo@example.com	true	2012-02-17 19:04:52	2026-01-28 00:27:20
amos.linnan	Amos Linnan	amos.linnan@example.com	true	2012-02-17 19:04:51	2026-01-28 00:27:18

- Incident assignment

- Delete blocked message

User ID	Name	Email	Active	Created	Updated
abel.tuter	Abel Tuter	abel.tuter@example.com	true	2012-02-17 19:04:52	2026-01-28 00:27:19
abraham.lincoln	Abraham Lincoln	abraham.lincoln@example.com	true	2013-07-23 17:15:54	2026-01-28 00:27:20
adela.cervantsz	Adela Cervantsz	adela.cervantsz@example.com	true	2012-02-17 19:04:50	2026-01-28 00:27:16
alileen.mottern	Aileen Mottern	alileen.mottern@example.com	true	2012-02-17 19:04:49	2026-01-28 00:27:19
ajay123	Ajay Pydi		true	2026-01-28 21:05:14	2026-01-28 21:05:14
akhil123	Akhil Reddy		true	2026-01-28 22:01:33	2026-01-28 22:01:33
alejandra.prenatt	Alejandra Prenatt	alejandra.prenatt@example.com	true	2012-02-17 19:04:52	2026-01-28 00:27:16
alejandro.mascall	Alejandro Mascall	alejandro.mascall@example.com	true	2012-02-17 19:04:52	2026-01-28 00:27:20
alene.rabeck	Alene Rabeck	alene.rabeck@example.com	true	2012-02-17 19:04:53	2026-01-28 00:27:21
alfonso.griglen	Alfonso Griglen	alfonso.griglen@example.com	true	2012-02-17 19:04:51	2026-01-28 00:27:16
alissa.mountjoy	Alissa Mountjoy	alissa.mountjoy@example.com	true	2012-02-17 19:04:52	2026-01-28 00:27:19
allan.schwandt	Allan Schwandt	allan.schwandt@example.com	true	2012-02-17 19:04:53	2026-01-28 00:27:20
allie.pumphrey	Allie Pumphrey	allie.pumphrey@example.com	true	2012-02-17 19:04:52	2026-01-28 00:27:20
allyson.gillisple	Allyson Gillisple	allyson.gillisple@example.com	true	2012-02-17 19:04:50	2026-01-28 00:27:16
alva.pennigton	Alva Pennigton	alva.pennigton@example.com	true	2012-02-17 19:04:50	2026-01-28 00:27:21
alyssa.biasotti	Alyssa Biasotti	alyssa.biasotti@example.com	true	2012-02-17 19:04:52	2026-01-28 00:27:16
amelia.caputo	Amelia Caputo	amelia.caputo@example.com	true	2012-02-17 19:04:52	2026-01-28 00:27:20
amos.linnan	Amos Linnan	amos.linnan@example.com	true	2012-02-17 19:04:51	2026-01-28 00:27:18
andrew.jackson	Andrew Jackson	andrew.jackson@example.com	true	2013-07-23 17:34:44	2026-01-28 00:27:19
andrew.rch	Andrew Rch	andrew.rch@example.com	true	2025-12-03 08:45:02	2026-01-28 00:27:19

## 12. Successful validation output

Demo / Repository:

<https://github.com/Nikhilreddy810/Prevent-User-Deletion ServiceNow>

## 13. Known Issues

- Works only inside ServiceNow environment
- Depends on correct table relationships

- Requires admin permissions for full testing

### **13. Future Enhancements**

- Approval workflow before user deletion
- Audit logs for deletion attempts
- Email notifications to administrators
- Extension to prevent deletion in other critical tables (Incidents, Tasks, Assets)