**Project Title :**

**Optimizing User, Group, and Role Management with Access Control and Workflows**

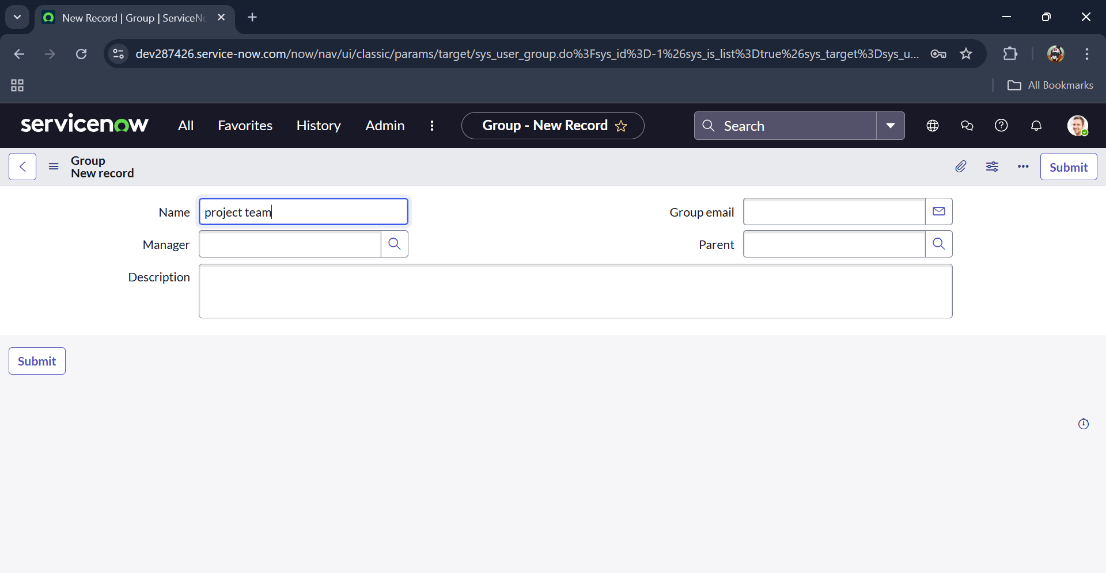
**Task: Creating a Group Record Named "Project Team"**

**Objective:**To establish a dedicated group in ServiceNow that represents the *Project Team*, enabling role-based access, assignment, and streamlined collaboration within the instance.

**Implementation:**

1. Navigated to *User Administration > Groups* in the ServiceNow instance.
2. Clicked on New to create a group record.
3. Entered the group name as Project Team.
4. Configured relevant details such as description, manager (if applicable), and email for the group.
5. Saved the record to finalize the creation.
6. (Optional) Added users who are part of the project team to this group for proper role-based management.

**Outcome:**A new group named Project Team was successfully created. This group can now be used for task assignments, access permissions, and collaborative workflows, ensuring efficient project execution and management within ServiceNow.



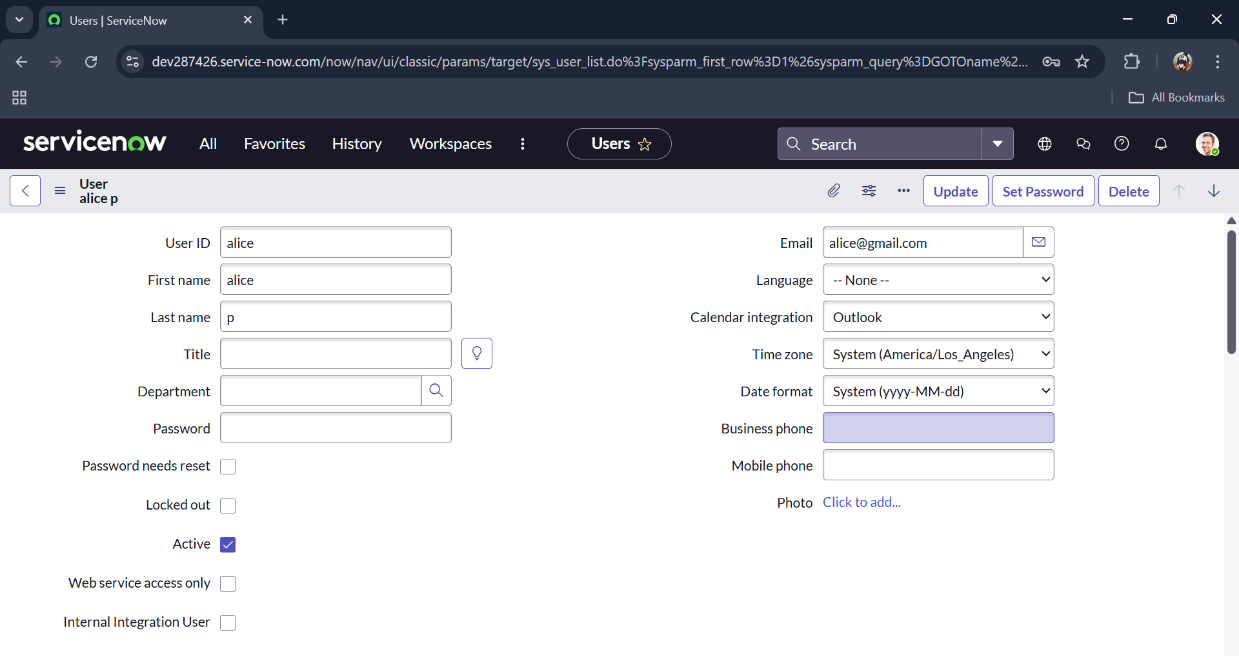
**Task: Creating a User Record for "Alice P"**

**Objective:**  
To add a new user, *Alice P*, into the ServiceNow instance, ensuring she has a unique identity for authentication, role assignment, and participation in project-related activities.

**Implementation:**

1. Navigated to *User Administration > Users* in the ServiceNow instance.
2. Clicked on New to create a user record.
3. Entered user details such as:
   * First Name: Alice
   * Last Name: P
   * User ID / Username: alice.p (or as per naming convention)
   * Email: [alice.p@example.com](mailto:alice.p@example.com) (if applicable)
4. Configured additional fields such as password and department, if required.
5. Saved the record to finalize the creation.
6. (Optional) Assigned *Alice P* to the Project Team group for project-related access.

**Outcome:**  
The user record for Alice P was successfully created. She now has access to the ServiceNow platform with her credentials and can be assigned roles, added to groups, and associated with project activities as needed.



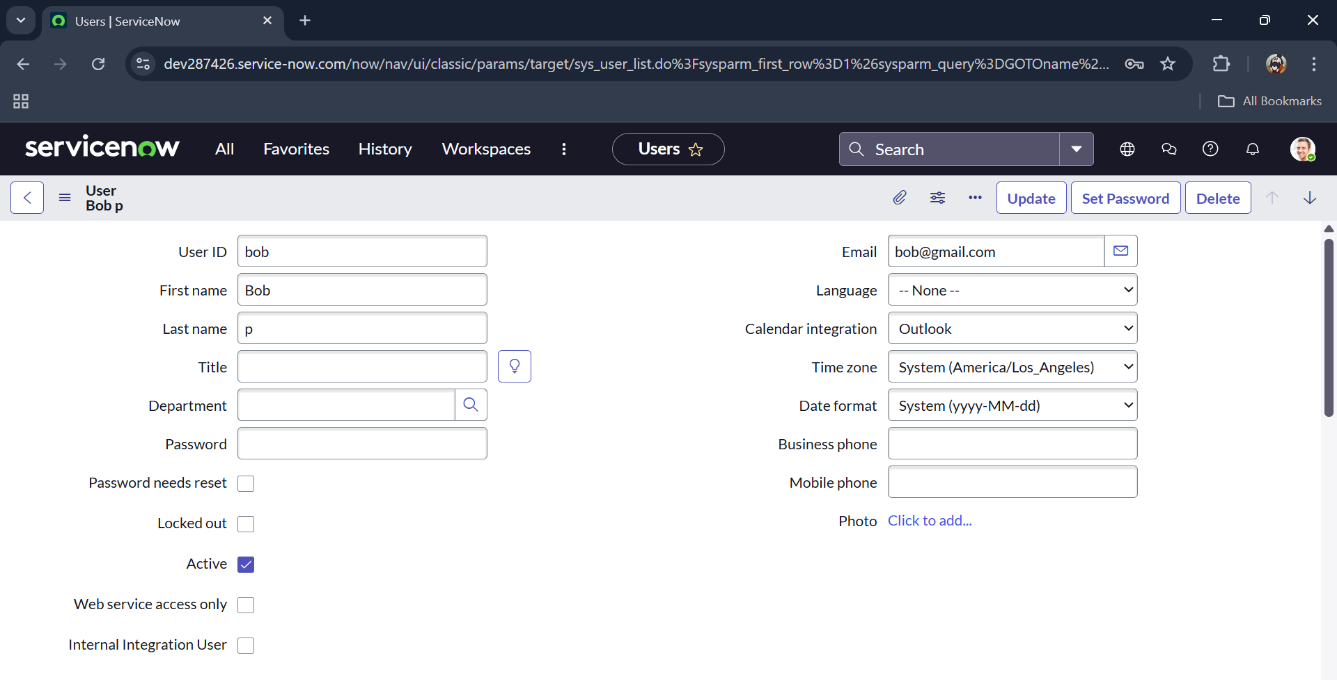
**Task: Creating a User Record for "Bob"**

**Objective:**  
To add a new user, *Bob*, into the ServiceNow instance so he can securely log in, be assigned to groups, and participate in project-related workflows.

**Implementation:**

1. Navigated to *User Administration > Users* in the ServiceNow instance.
2. Clicked on New to create a user record.
3. Entered user details such as:
   * First Name: Bob
   * Last Name: (left blank or filled as per requirement)
   * User ID / Username: bob (or according to naming convention)
   * Email: [bob@example.com](mailto:bob@example.com) (if applicable)
4. Configured additional fields such as password, department, or title if required.
5. Saved the record to complete the creation.
6. (Optional) Assigned *Bob* to the Project Team group to ensure he has the required project access.

**Outcome:**  
The user record for Bob was successfully created. He now has his own login credentials and can be added to groups, assigned roles, and associated with ServiceNow tasks or workflows as needed.



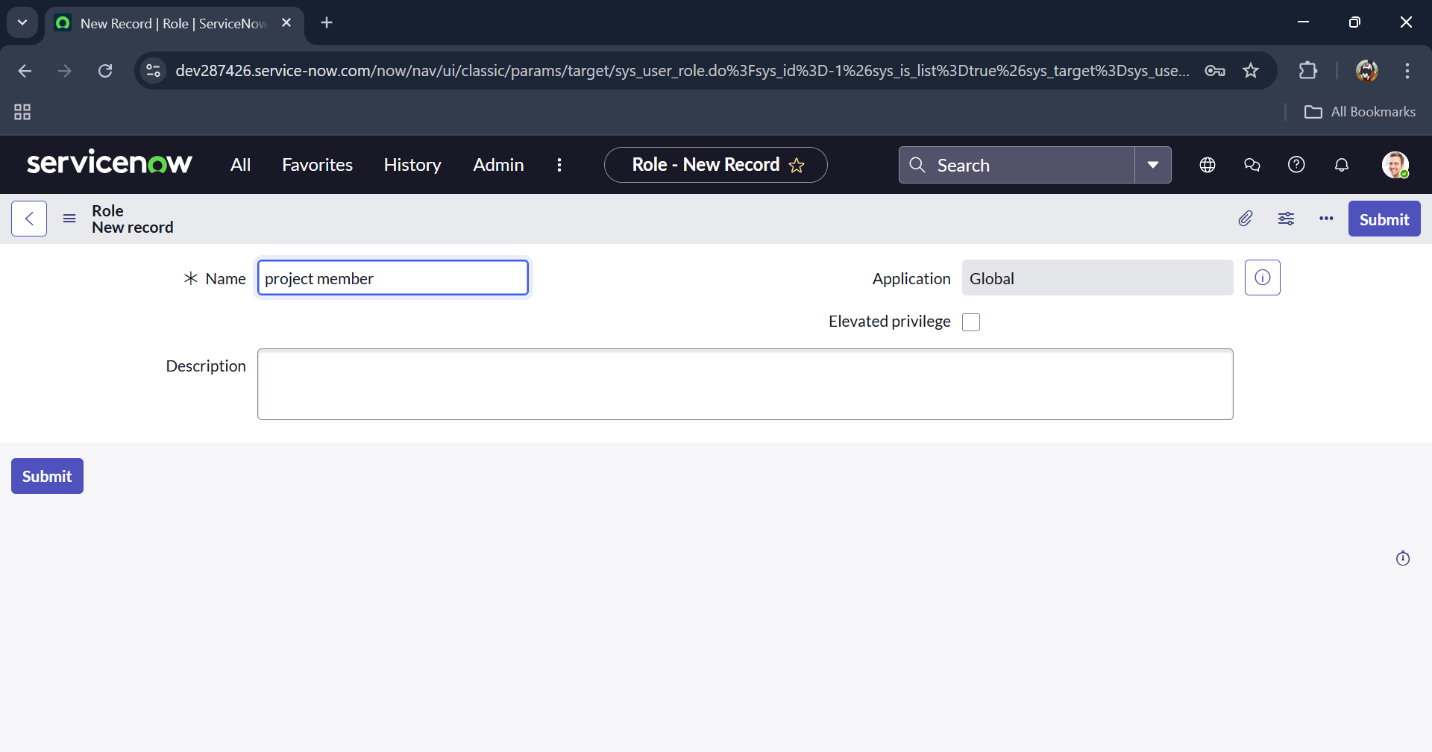
**Task: Creating a Role Record Named "Project Member"**

**Objective:**  
To create a new role, *Project Member*, in the ServiceNow instance that can be assigned to users and groups for controlled access and permissions within project-related activities.

**Implementation:**

1. Navigated to *User Administration > Roles* in the ServiceNow instance.
2. Clicked on **New** to create a role record.
3. Entered the role name as **Project Member**.
4. Added a description to specify the purpose of the role, e.g., *“Provides access and permissions required for project participation and task handling.”*
5. Saved the role record to finalize creation.
6. (Optional) Associated this role with the **Project Team** group so all members inherit the role permissions.

**Outcome:**  
A new role named **Project Member** was successfully created. This role can now be assigned to users or groups, ensuring that project participants have the appropriate access rights and privileges within the ServiceNow environment.



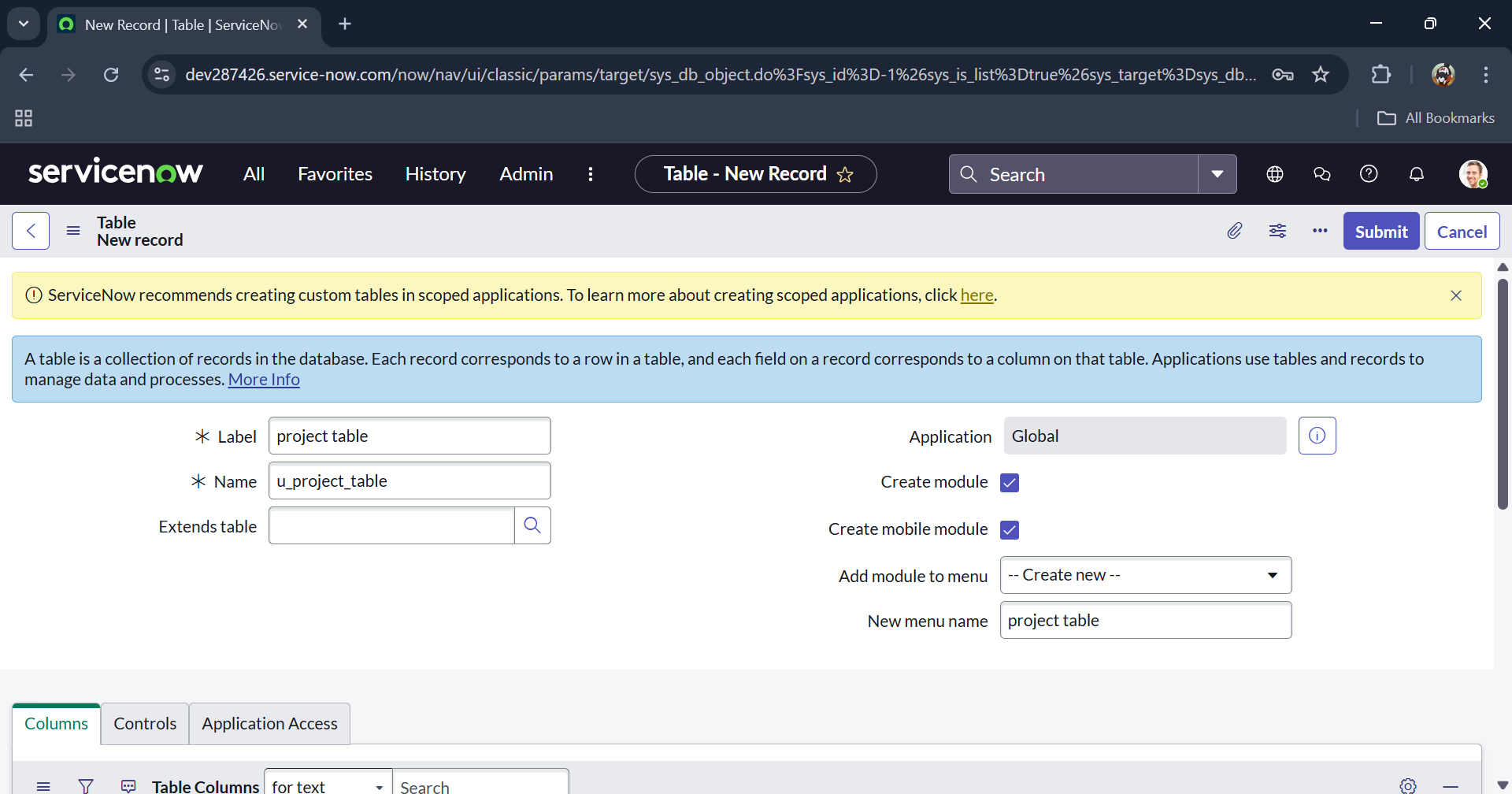
**Task: Creating a Table Named "Project Table"**

**Objective:**  
To create a custom table, *Project Table*, in ServiceNow for storing and managing project-related data in a structured manner, ensuring scalability and ease of use.

**Implementation:**

1. Navigated to *System Definition > Tables* in the ServiceNow instance.
2. Clicked on **New** to create a table record.
3. Entered the table label as **Project Table**.
   * The system automatically generated a database name (e.g., u\_project\_table)..
4. Saved the record to finalize the table creation.
5. (Optional) Added columns/fields like *Project Name, Start Date, End Date, Status, Owner,* and *Description* to capture project-specific details.

**Outcome:**  
The **Project Table** was successfully created and is now available in the ServiceNow instance. It serves as the central data structure for storing and managing project-related information, supporting future workflows, forms, and reporting.



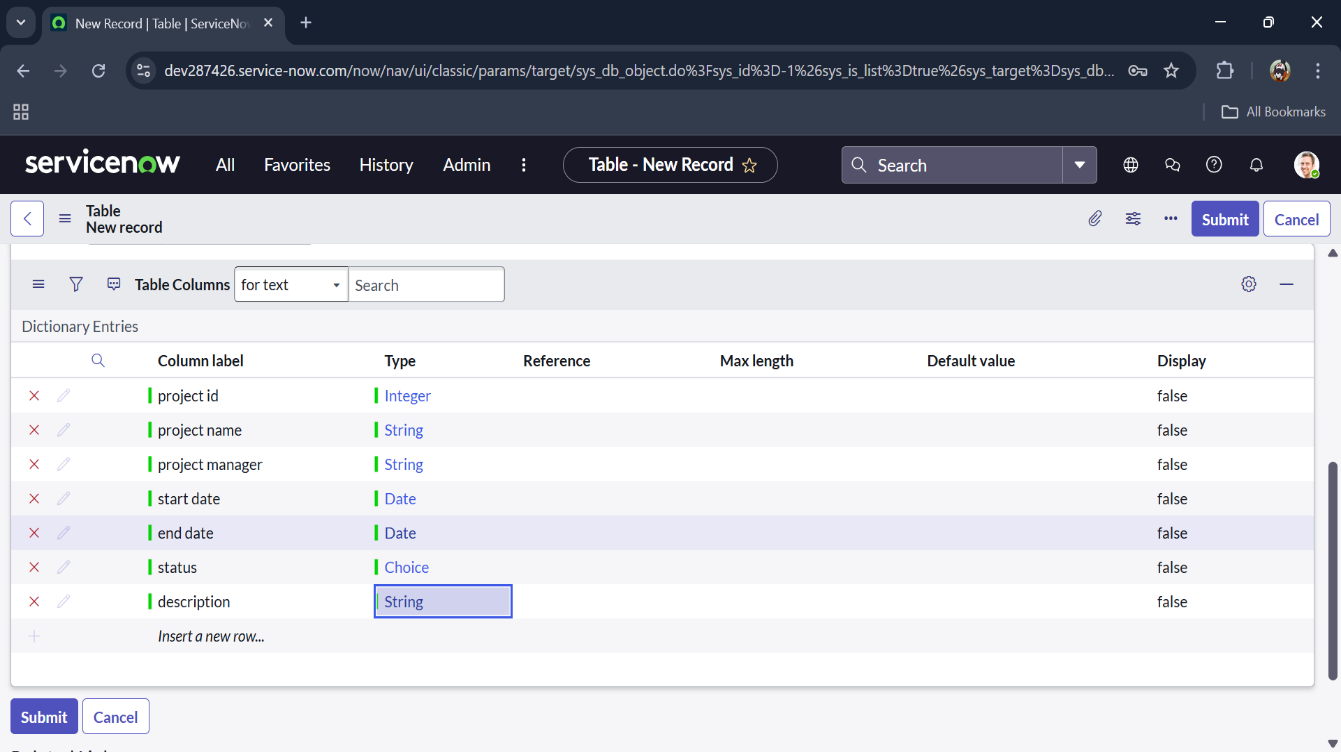
**Task: Creating Columns in the Project Table**

**Objective:**  
To design and configure columns (fields) in the *Project Table* to capture all necessary project-related details, ensuring data consistency and usability across workflows.

**Implementation:**  
The following columns were created in the **Project Table**:

1. **Project ID** *(Integer)*
   * A unique numeric identifier for each project.
   * Useful for referencing and indexing projects.
2. **Project Name** *(String)*
   * Stores the official name/title of the project.
   * Provides easy identification in records and reports.
3. **Project Manager** *(String)*
   * Captures the name or identifier of the project manager.
   * Helps in tracking project ownership and accountability.
4. **Start Date** *(Date)*
   * Represents the planned or actual start date of the project.
   * Assists in scheduling and timeline management.
5. **End Date** *(Date)*
   * Indicates the planned or actual completion date of the project.
   * Supports project tracking and deadline monitoring.
6. **Status** *(Choice)*
   * A dropdown/choice field for selecting the current status of the project (e.g., *Not Started, In Progress, Completed*).
   * Helps in progress tracking and reporting.
7. **Description** *(String)*
   * A text field for entering additional details, notes, or context about the project.
   * Provides flexibility for documentation and clarifications.

**Outcome:**  
The **Project Table** was successfully designed with all essential columns required for project management. This structure ensures proper tracking of project details, timelines, ownership, and progress within the ServiceNow environment.



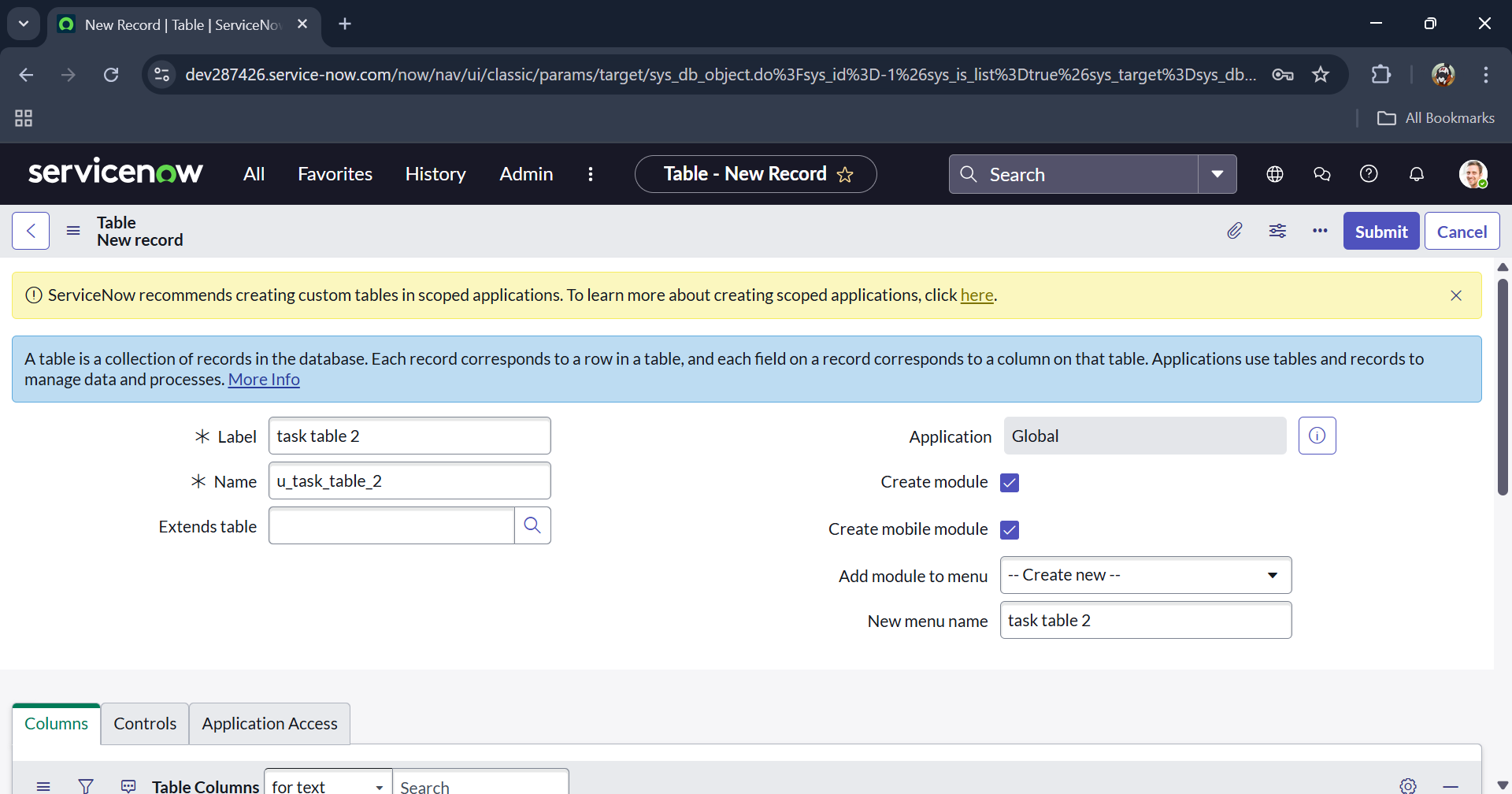
**Task: Creating a Table Named "Task Table 2"**

**Objective:**  
To create a custom table, *Task Table 2*, in the ServiceNow instance for tracking project-related tasks, ensuring detailed task management and association with project records.

**Implementation:**

1. Navigated to *System Definition > Tables* in the ServiceNow instance.
2. Clicked on **New** to create a new table record.
3. Entered the table label as **Task Table 2**.
   * The system auto-generated a database name (e.g., u\_task\_table\_2).
4. Saved the table record to finalize creation.
5. (Optional) Added specific fields like *Task Name, Assigned To, Due Date, Status, Priority,* and *Description* to support detailed task tracking.

**Outcome:**  
The **Task Table 2** was successfully created in the ServiceNow instance. It now serves as a structure for managing and monitoring individual project tasks, enabling better organization, reporting, and linkage to the main *Project Table*.



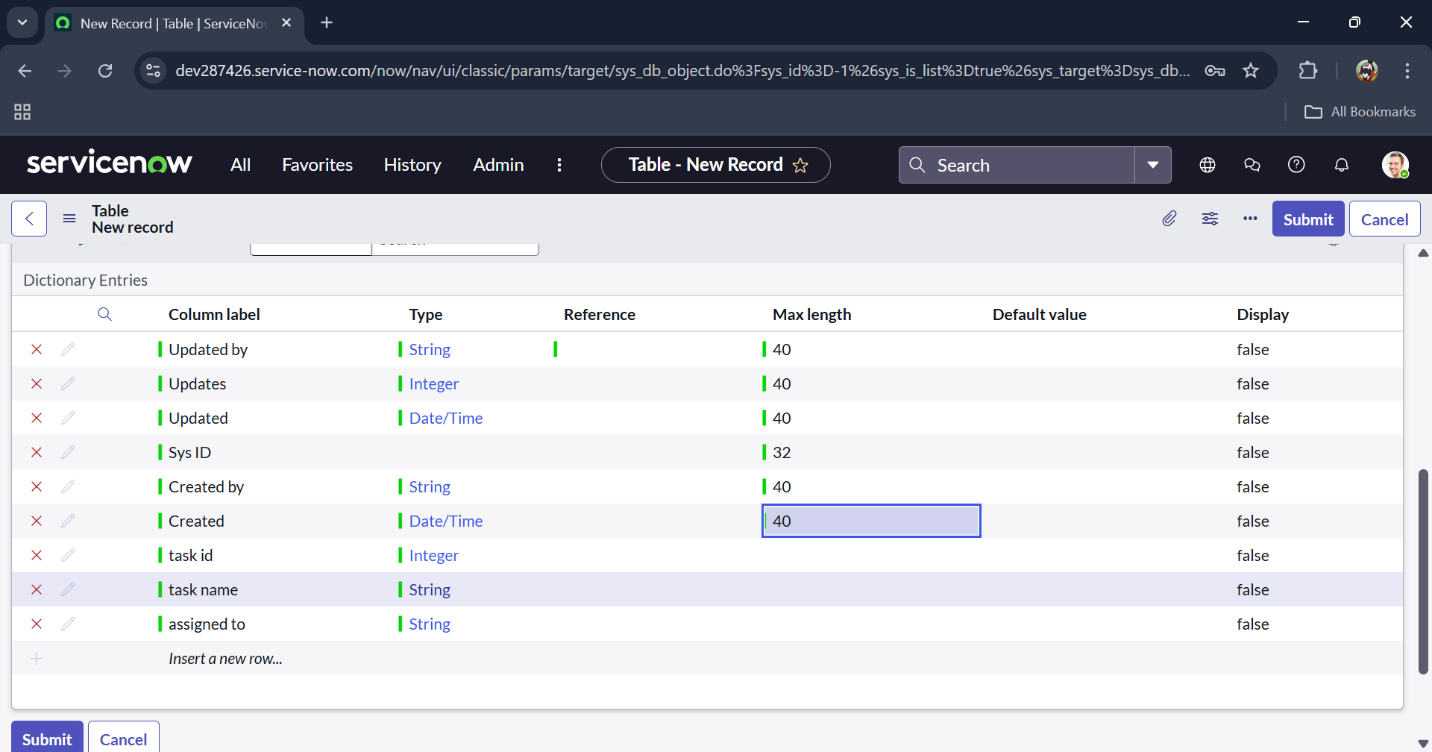
**Task: Creating Columns in Task Table 2**

**Objective:**  
To define and configure columns in the *Task Table 2* to store and manage task-related details, ensuring proper tracking, ownership, and accountability.

**Implementation:**  
The following columns were created in the **Task Table 2**:

1. **Updated by** *(String)*
   * Stores the name/identifier of the user who last updated the task record.
2. **Updates** *(Integer)*
   * Tracks the number of times a record has been updated.
3. **Updated** *(Date/Time)*
   * Captures the exact date and time when the record was last modified.
4. **Sys ID** *(Integer/String – System Generated)*
   * A unique system identifier for each task record, ensuring internal tracking.
5. **Created by** *(String)*
   * Stores the name/identifier of the user who created the task record.
6. **Created** *(Date/Time)*
   * Captures the exact date and time when the task record was initially created.
7. **Task ID** *(Integer)*
   * A unique numeric identifier assigned to each task for easy reference.
8. **Task Name** *(String)*
   * Stores the title or name of the task, providing quick identification.
9. **Assigned To** *(String)*
   * Specifies the user to whom the task has been assigned.

**Outcome:**  
The **Task Table 2** was successfully designed with all necessary columns for task management. These fields ensure accurate tracking of task ownership, creation, updates, and responsibility assignments within the ServiceNow platform.



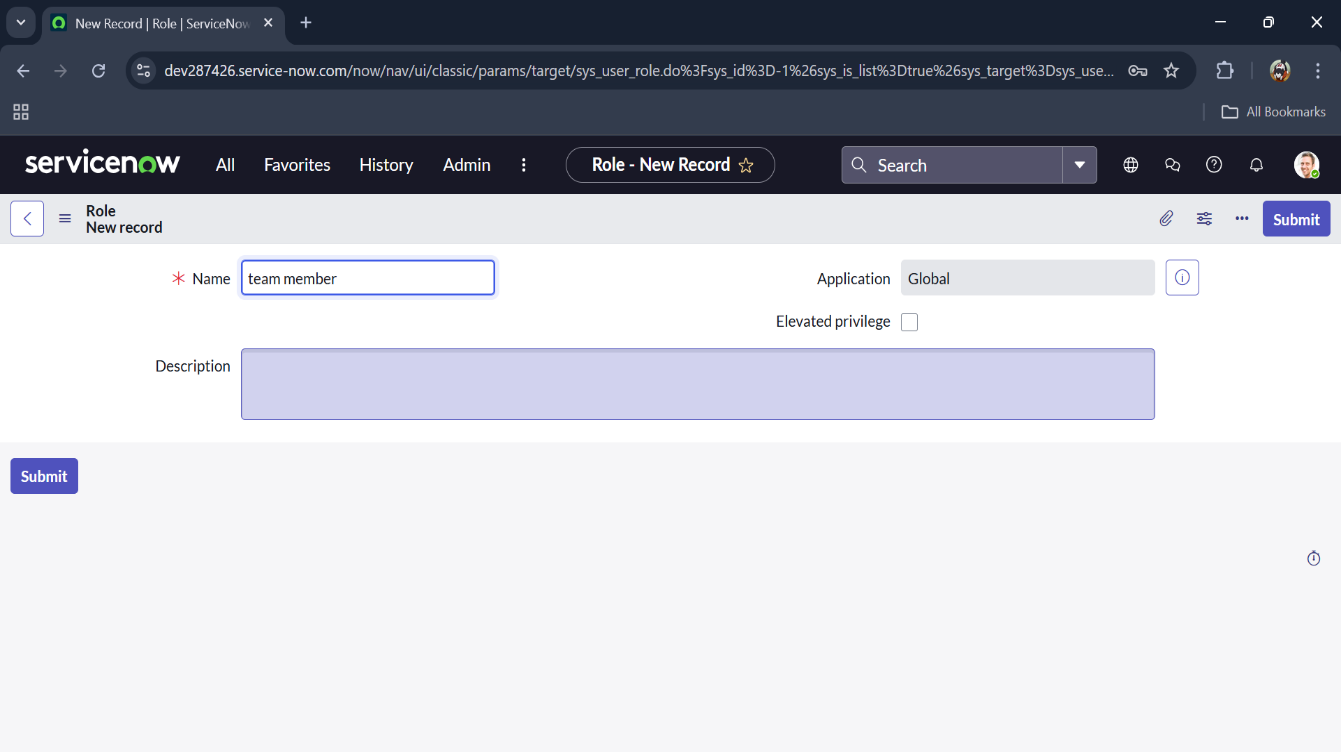
**Task:** Creating a Role Named "Team Member"

**Objective:**  
To create a custom role, **Team Member**, in the ServiceNow instance to manage access permissions for project participants, ensuring controlled and secure access to project-related modules and records.

**Implementation:**

1. Navigated to **User Administration > Roles** in the ServiceNow instance.
2. Clicked on **New** to create a new role record.
3. Entered the **Role Name** as **Team Member**.
   * (Optional) Added a description, e.g., “Role for project participants to access and manage tasks and project data.”
4. Saved the role record to finalize creation.
5. (Optional) Assigned this role to users (like Alice and Bob) to grant them the appropriate permissions.

**Outcome:**  
The **Team Member** role was successfully created in the ServiceNow instance. It now provides role-based access control for project participants, allowing them to perform necessary project activities while maintaining system security.



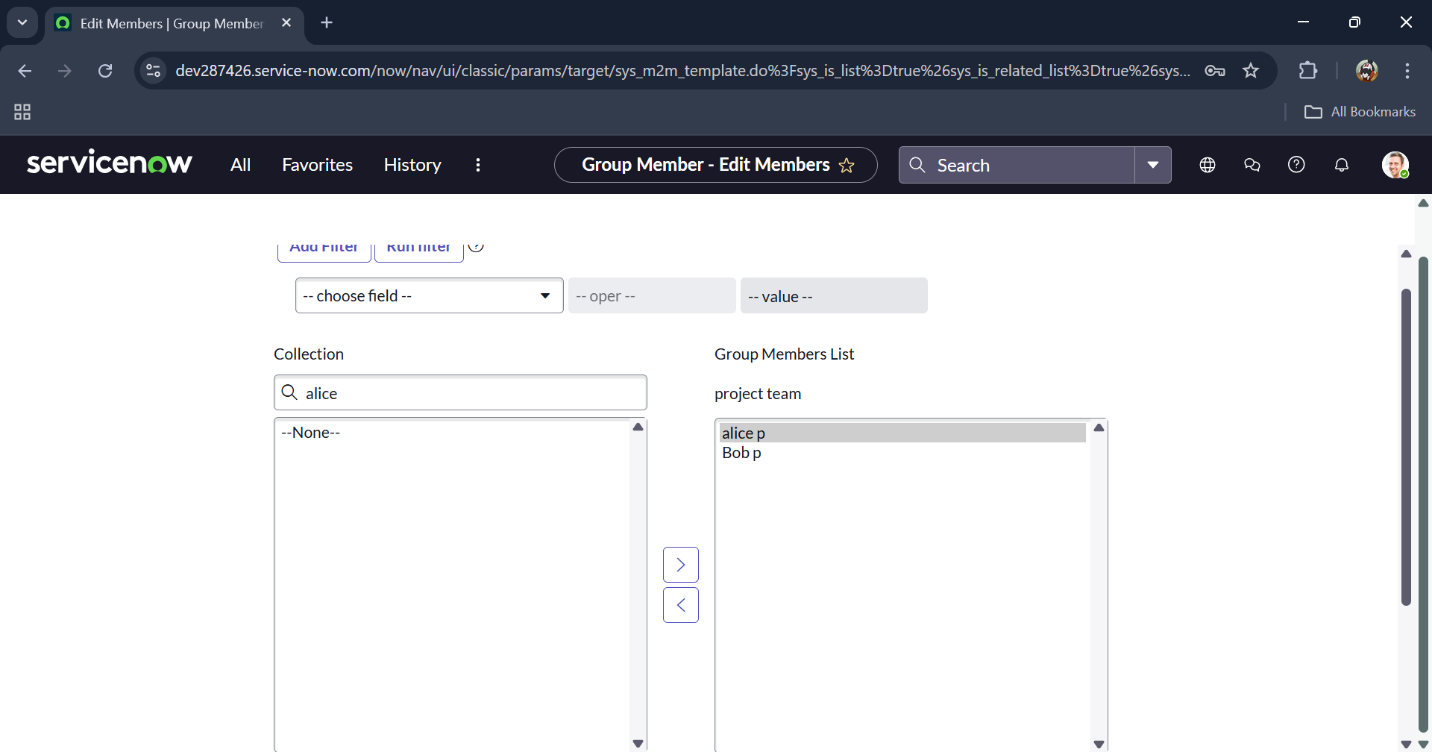
**Task:** Assign Users (Alice, Bob) to Group "Project Team"

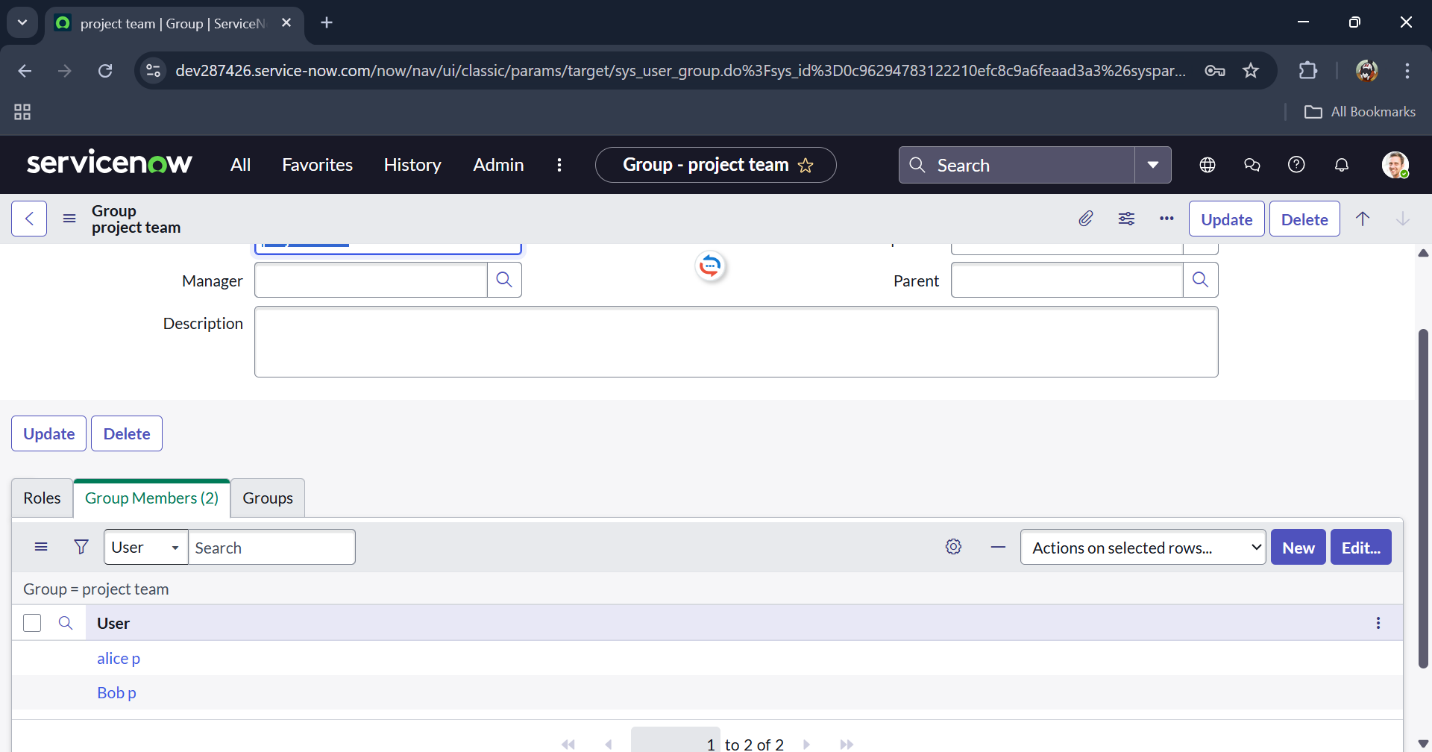
**Objective:**  
To add users **Alice** and **Bob** to the **Project Team** group in the ServiceNow instance, enabling them to inherit the permissions and access associated with the group for effective project collaboration.

**Implementation:**

1. Opened the **ServiceNow** instance.
2. Navigated to **All > Groups** using the search bar.
3. Selected **Tables** under **System Definition**.
4. Chose the **Project Team** group from the list.
5. Under **Group Members**, clicked on **Edit**.
6. Selected **Alice P** and **Bob P** from the list of available users.
7. Clicked **Save** to finalize the assignment.

**Outcome:**  
Users **Alice** and **Bob** were successfully added to the **Project Team** group. They now have access to the permissions and roles associated with the group, enabling them to participate in project activities and collaborate effectively.





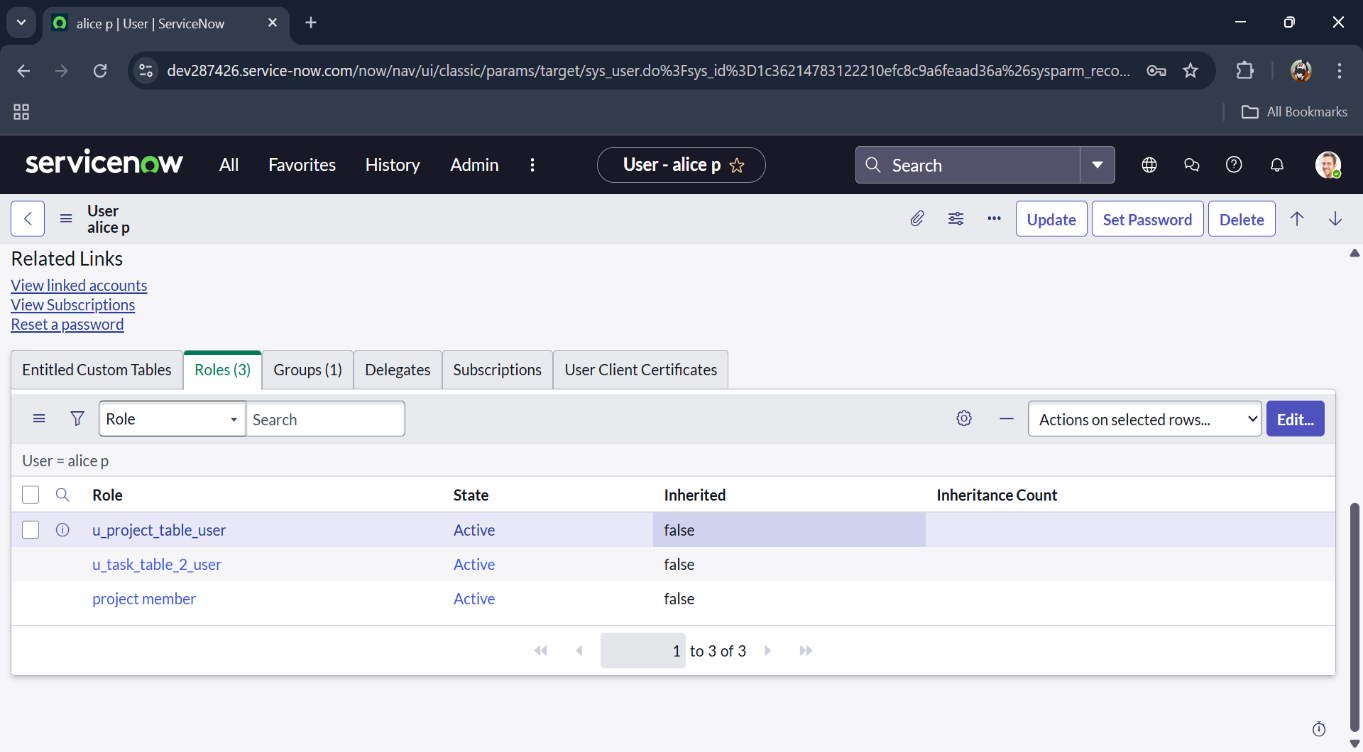
**Task:** Assign Roles to User "Alice"

**Objective:**  
To assign appropriate roles to the user **Alice** in the ServiceNow instance, granting her the necessary permissions to access and manage project and task-related records.

**Implementation:**

1. Opened the **ServiceNow** instance.
2. Navigated to **All > Users** using the search bar.
3. Selected **Tables** under **System Definition**.
4. Chose the user **Alice** from the list (under Project Manager).
5. Clicked on **Edit** under the user record.
6. Assigned the **Project Member** role and clicked **Save**.
7. Clicked **Edit** again to add the **u\_project\_table** and **u\_task\_table** roles.
8. Clicked **Save** and updated the form to finalize the changes.

**Outcome:**  
The user **Alice** was successfully assigned the **Project Member**, **u\_project\_table**, and **u\_task\_table** roles. She now has access to project and task management functionalities, enabling her to perform her project-related responsibilities effectively.



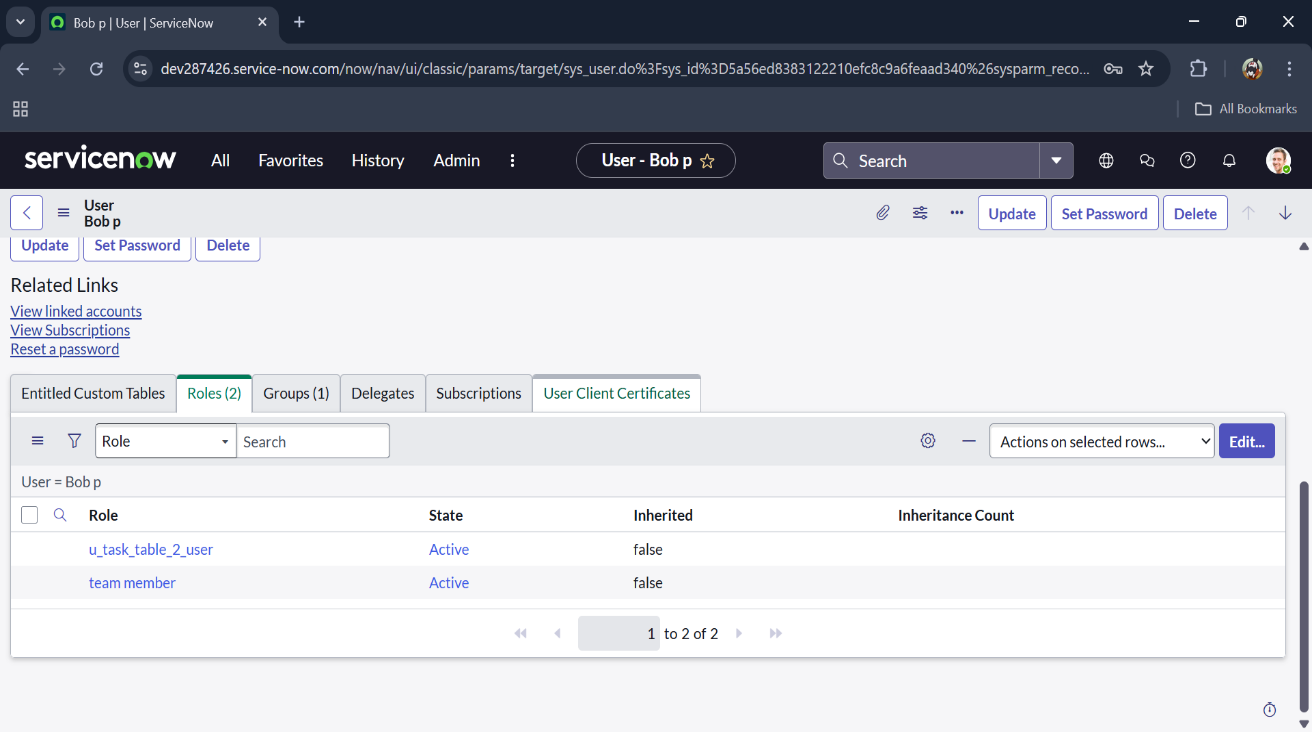
**Task:** Assign Roles to User "Bob"

**Objective:**  
To assign the **Team Member** role and table-level permissions to the user **Bob** in the ServiceNow instance, enabling him to access and interact with project and task records as per his role.

**Implementation:**

1. Opened the **ServiceNow** instance.
2. Navigated to **All > Users** using the search bar.
3. Selected **Tables** under **System Definition**.
4. Selected the user **Bob P** from the list.
5. Clicked on **Edit** under the user record.
6. Assigned the **Team Member** role and added the necessary table-level roles.
7. Clicked **Save** to finalize the role assignment.
8. Clicked on the **Profile Icon** and selected **Impersonate User** to log in as **Bob**.
9. Verified that **Task Table 2** is accessible for the user.

**Outcome:**  
The user **Bob** was successfully assigned the **Team Member** role along with the required table-level permissions. Upon impersonation, he can access **Task Table 2**, confirming that the role and permissions were correctly applied.



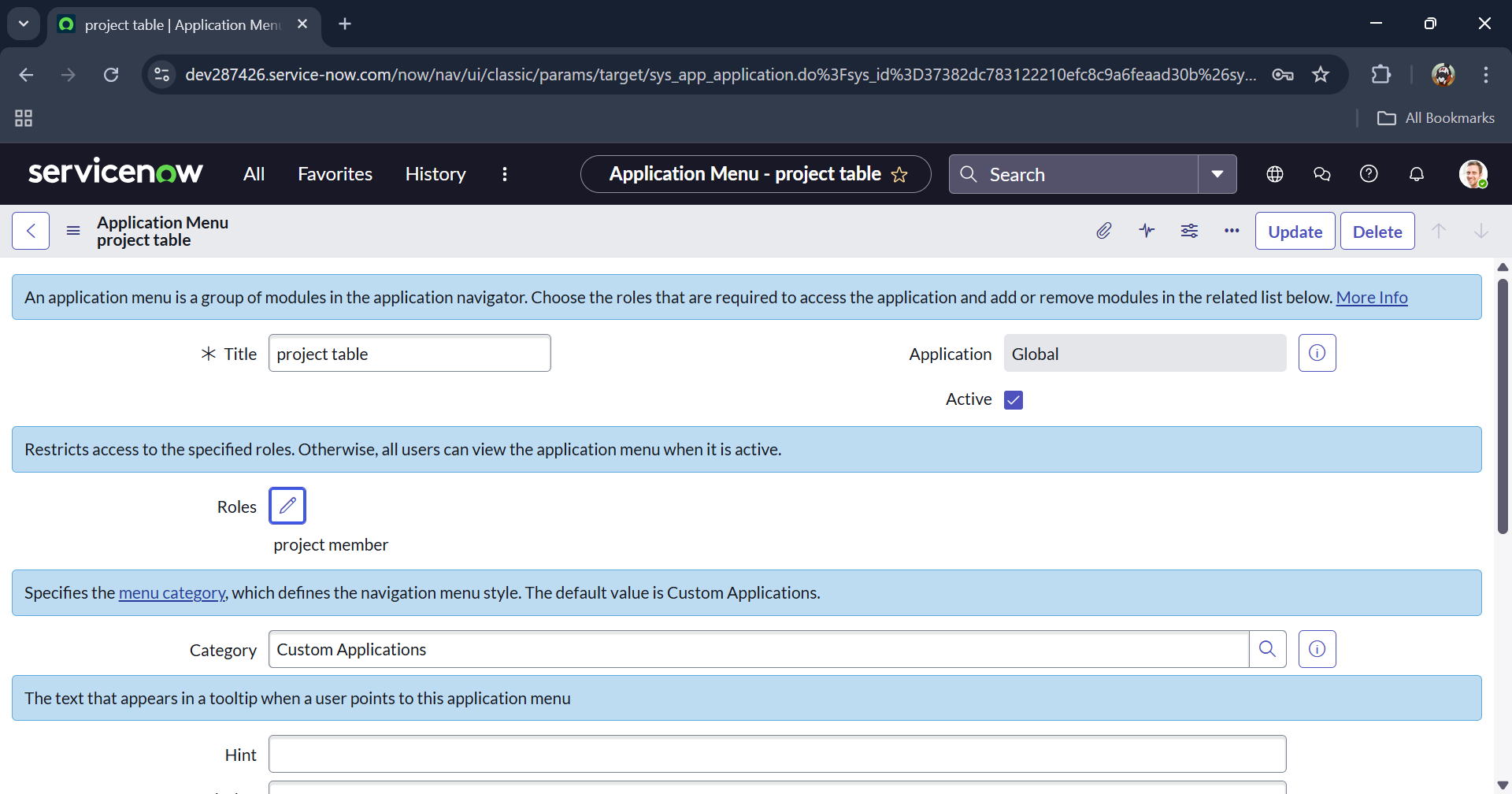
**Task:** Assign Table Access to Application

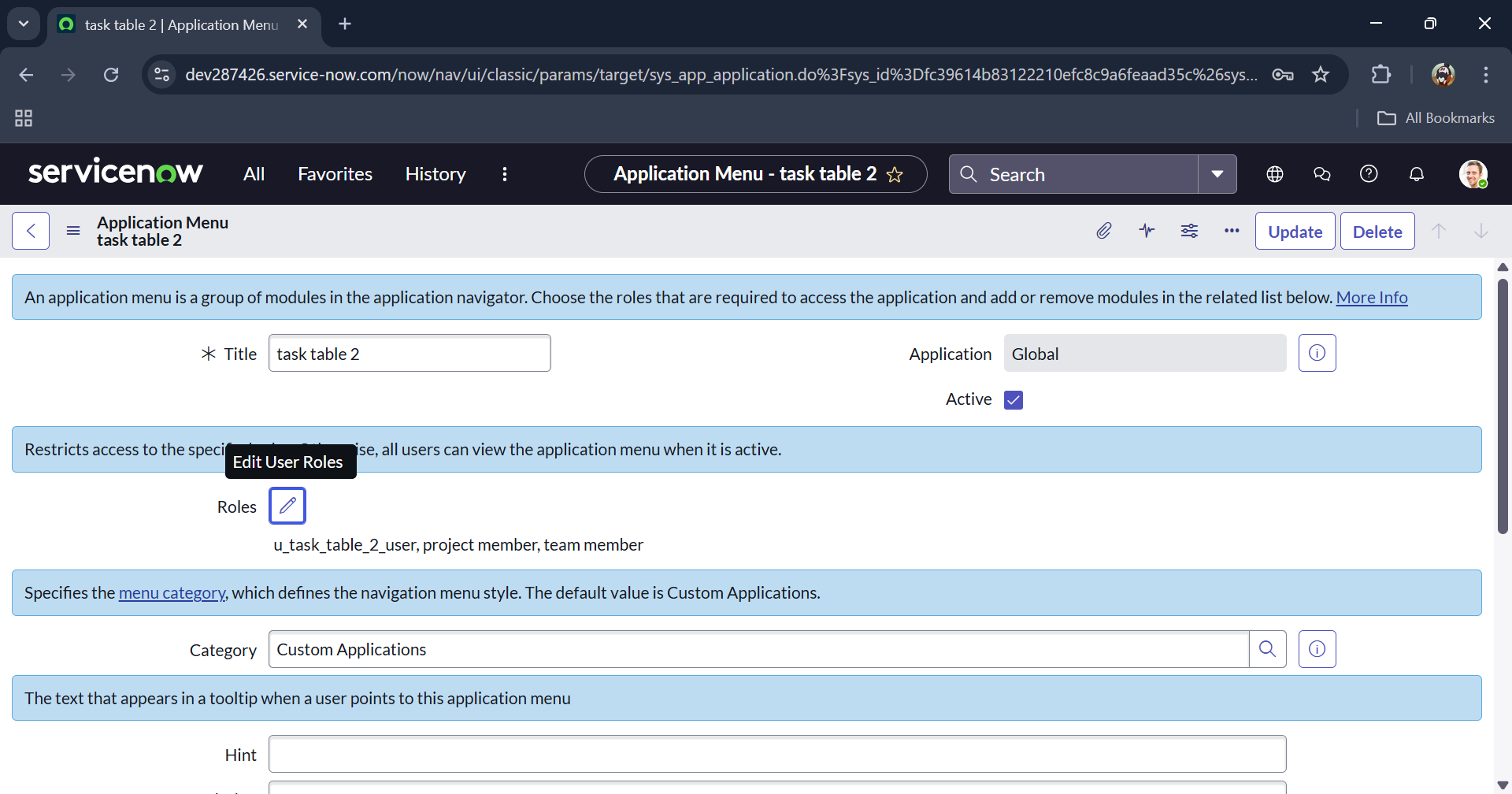
**Objective:**  
To assign appropriate roles to applications associated with tables in the ServiceNow instance, ensuring that only authorized users can access and interact with the corresponding modules.

**Implementation:**

1. While creating a table, ServiceNow automatically generates an **application** and **module** for that table.
2. Navigated to the **Application Navigator** and searched for the **Project Table** application.
3. Clicked on **Edit Module** for the Project Table application.
4. Assigned the **Project Member** role to the application.
5. Searched for the **Task Table 2** application and clicked on **Edit Application**.
6. Assigned the **Project Member** and **Team Member** roles to the Task Table 2 application.

**Outcome:**  
The applications associated with **Project Table** and **Task Table 2** now have role-based access control applied. Users with the assigned roles can access the modules, while unauthorized users are restricted, ensuring secure and controlled access to application data.





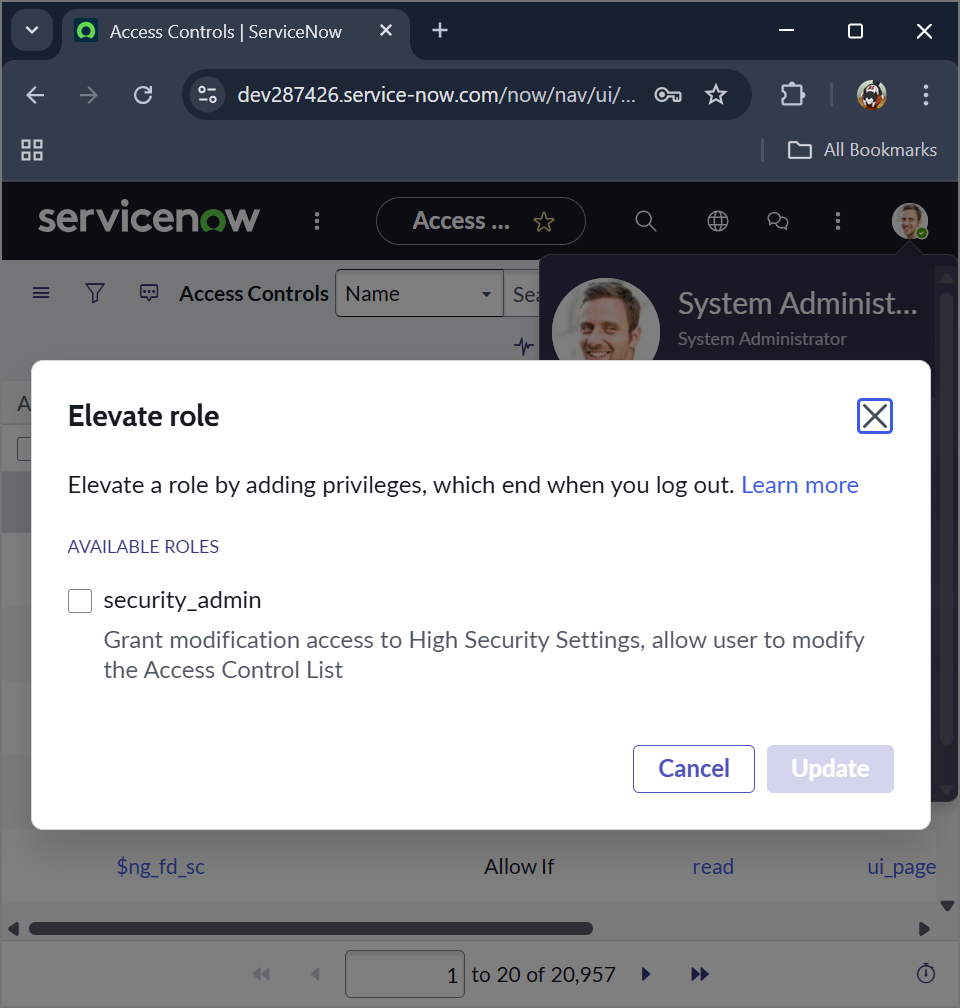
**Task:** Create ACL (Access Control List)

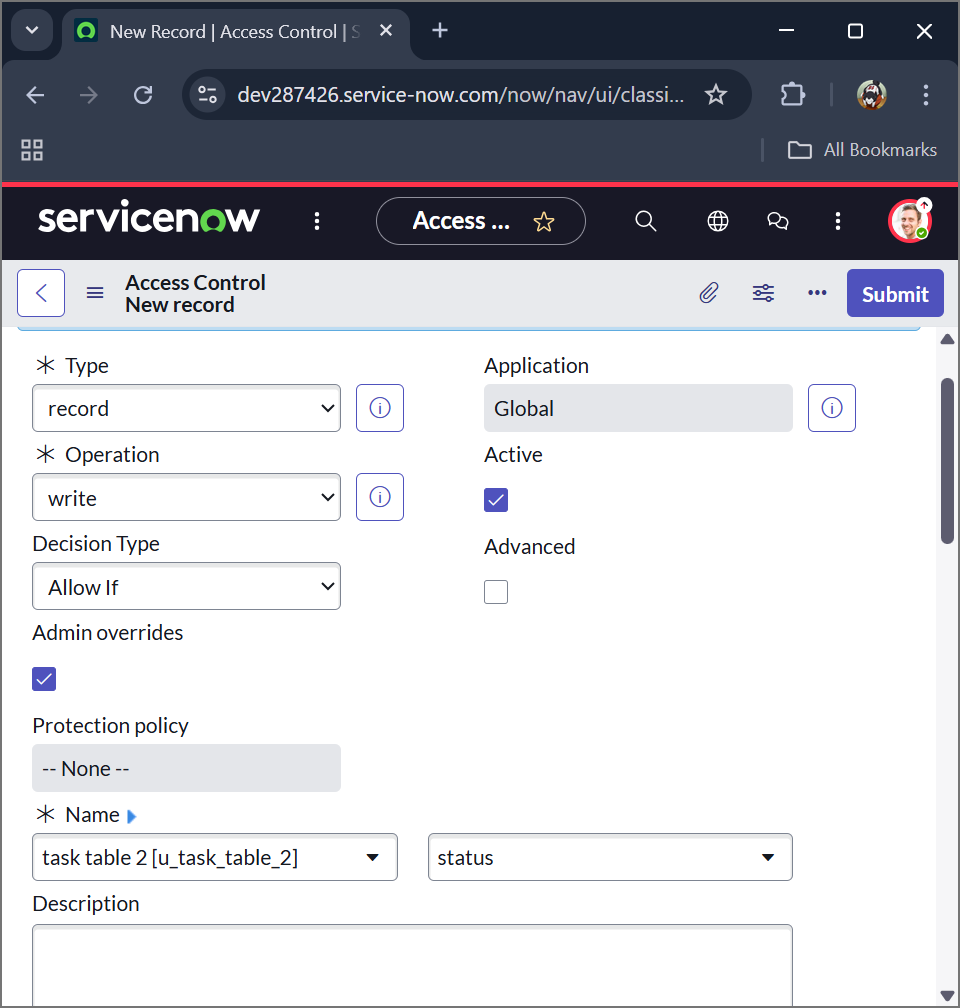
**Objective:**  
To create a new **Access Control List (ACL)** in the ServiceNow instance to define permissions for specific users or roles, controlling access to tables, records, or fields securely.

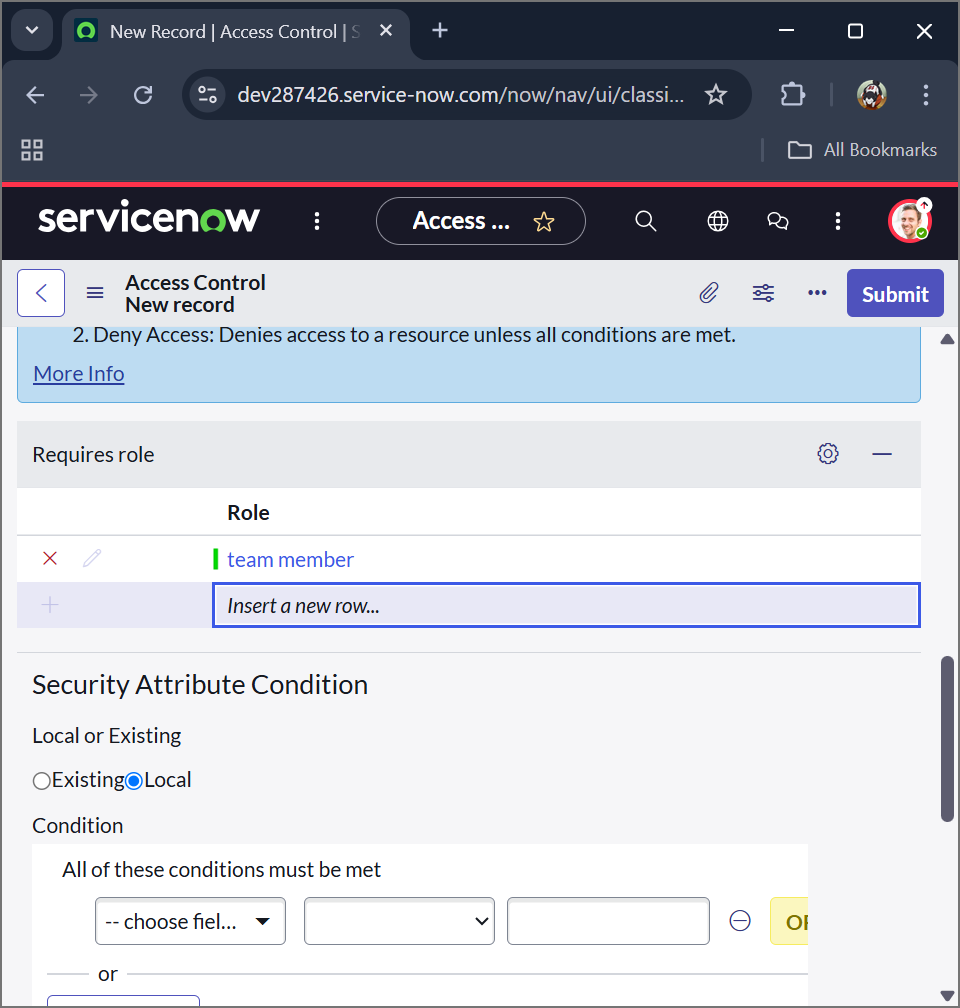
**Implementation:**

1. Opened the **ServiceNow** instance.
2. Navigated to **All > ACL** using the search bar.
3. Selected **Access Control (ACL)** under **System Security**.
4. Clicked on **Elevate Role** to gain administrative privileges required for creating ACLs.
5. Clicked **New** to create a new ACL record.
6. Filled in the necessary details to define the ACL, including:
   * **Type:** (e.g., record, table, or field)
   * **Operation:** (e.g., read, write, create, delete)
   * **Name:** (table or field name to secure)
   * **Roles:** (roles authorized to perform the operation)
7. Saved the ACL to apply the access control rule.

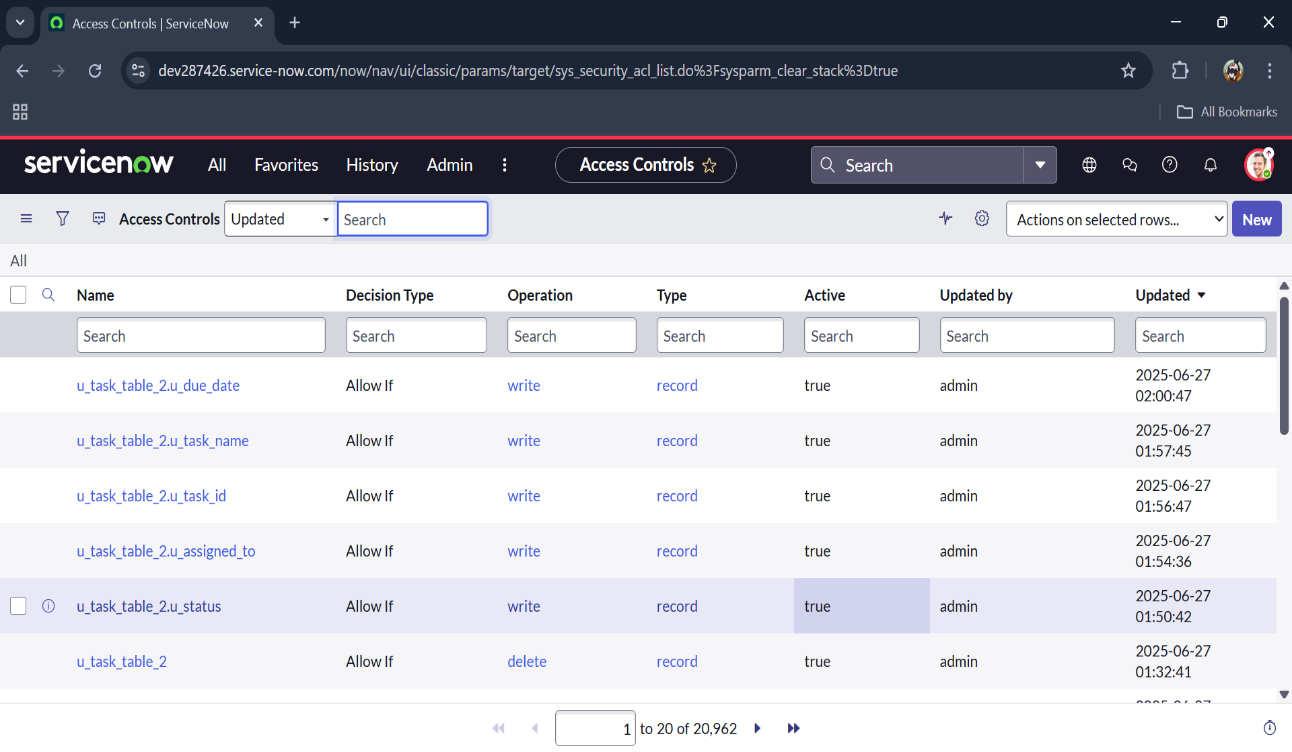
**Outcome:**  
A new ACL was successfully created in the ServiceNow instance, ensuring that access to the specified table, record, or field is restricted to authorized roles only, thereby enhancing security and enforcing role-based access control.







Repeated the process to create **four additional ACLs** for the required fields, assigning the appropriate roles to each.



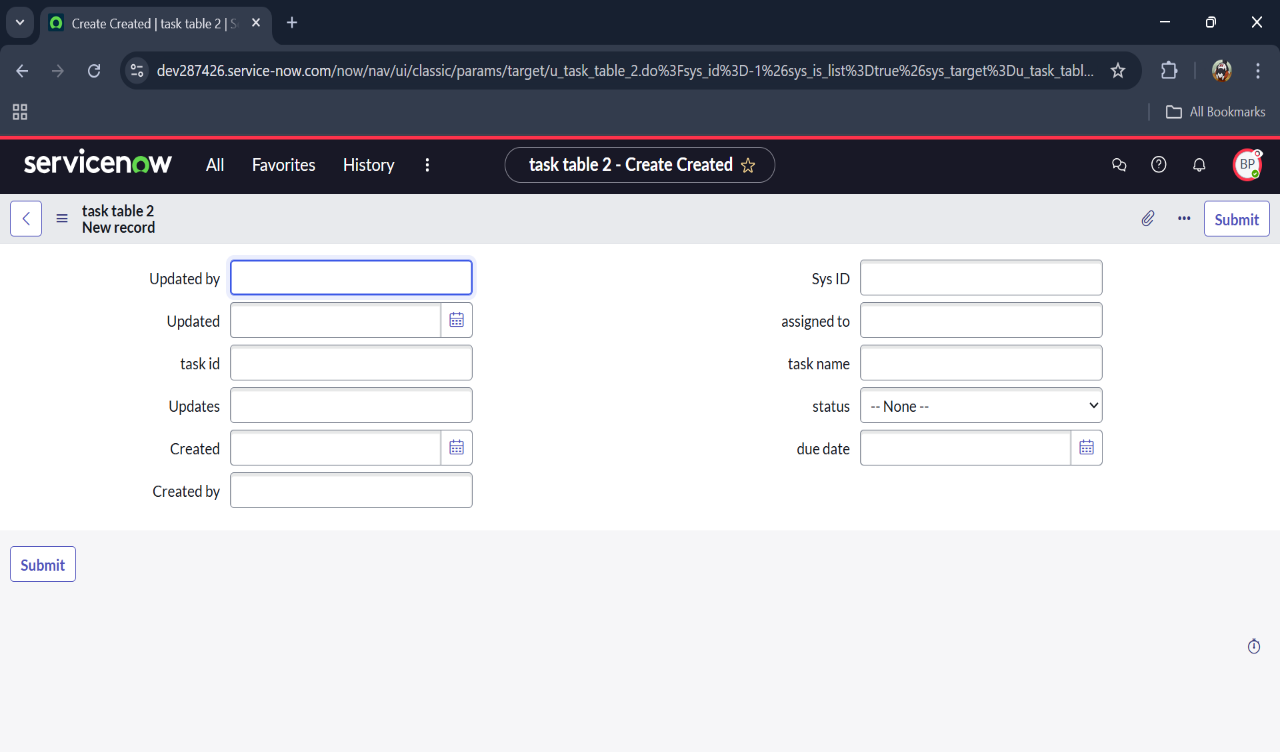
**Task:** Create a Flow to Assign Operations Ticket to Group

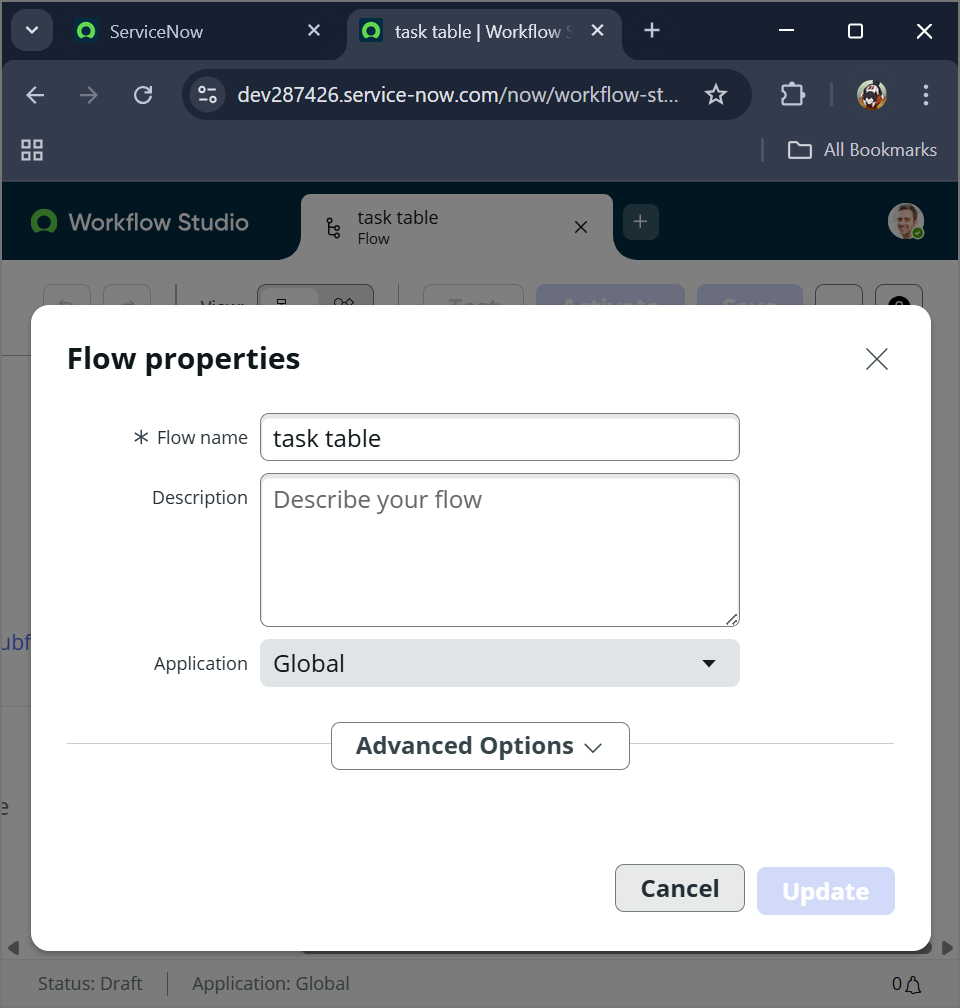
**Objective:**  
To create an automated flow in ServiceNow that assigns operations tickets from **Task Table** to a designated group, ensuring efficient task management and reducing manual assignment efforts.

**Implementation:**

1. Opened the **ServiceNow** instance.
2. Navigated to **All > Flow Designer** using the search bar.
3. Selected **Flow Designer** under **Process Automation**.
4. Clicked **New** and selected **Flow**.
5. Under **Flow Properties**, entered the following:
   * **Flow Name:** Task Table
   * **Application:** Global
6. Clicked **Build Flow** to start designing the flow logic.

**Outcome:**  
A new flow was successfully created in the Flow Designer. This flow will be used to automate the assignment of operations tickets from **Task Table** to the appropriate group, streamlining workflow management and improving operational efficiency.



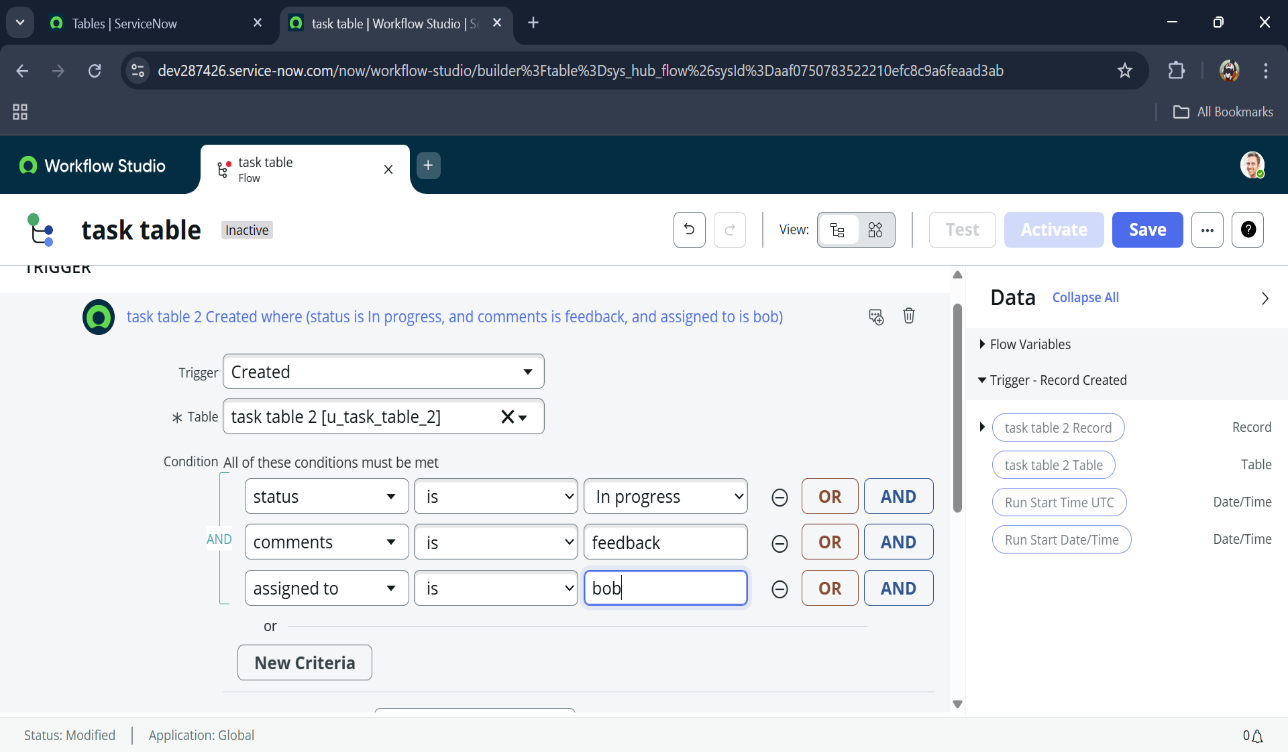


**Task:** Configure Trigger for Flow to Assign Operations Ticket

**Objective:**  
To define a trigger in the flow that initiates the automated assignment when specific conditions are met on **Task Table** records.

**Implementation:**

1. In the **Flow Designer**, clicked **Add a Trigger**.
2. Searched for and selected the **Create Record** trigger.
3. Configured the trigger with the following details:
   * **Table Name:** Task Table
   * **Conditions:**
     + **Field:** Status | **Operator:** is | **Value:** In Progress
     + **Field:** Comments | **Operator:** is | **Value:** Feedback
     + **Field:** Assigned To | **Operator:** is | **Value:** Bob
4. Clicked **Done** to finalize the trigger configuration.



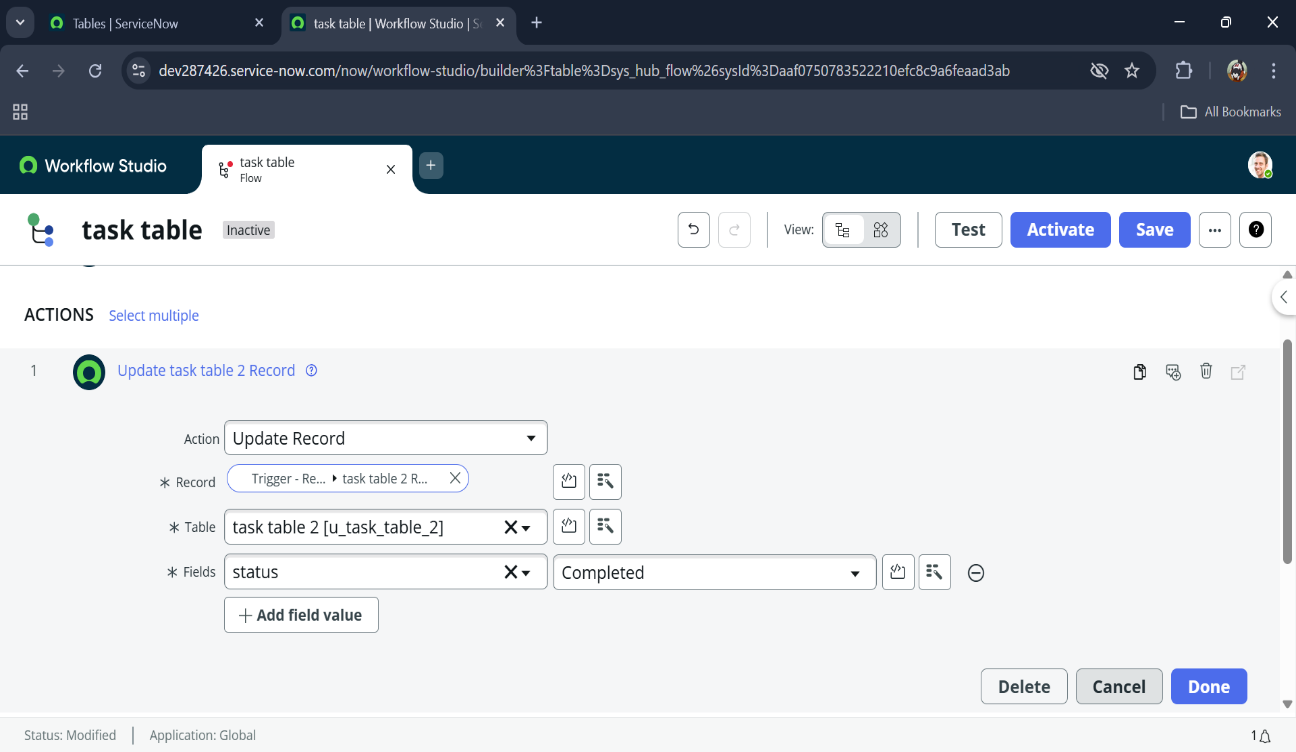
**Task:** Add Action to Update Records in Flow

**Objective:**  
To configure an action in the flow that automatically updates specific fields in the **Task Table** records when the flow is triggered, ensuring task status is correctly tracked.

**Implementation:**

1. In the **Flow Designer**, clicked **Add an Action**.
2. Searched for and selected the **Update Records** action.
3. In the **Record** field, dragged the appropriate fields from the **Data Pill** navigation on the right side.
4. The **Table** field was automatically assigned based on the trigger.
5. Added the field **Status** and set its value to **Completed**.
6. Clicked **Done** to finalize the action configuration.

**Outcome:**  
The flow is now configured to automatically update the **Status** of matching **Task Table** records to **Completed** when triggered, ensuring accurate task tracking and reducing manual updates.



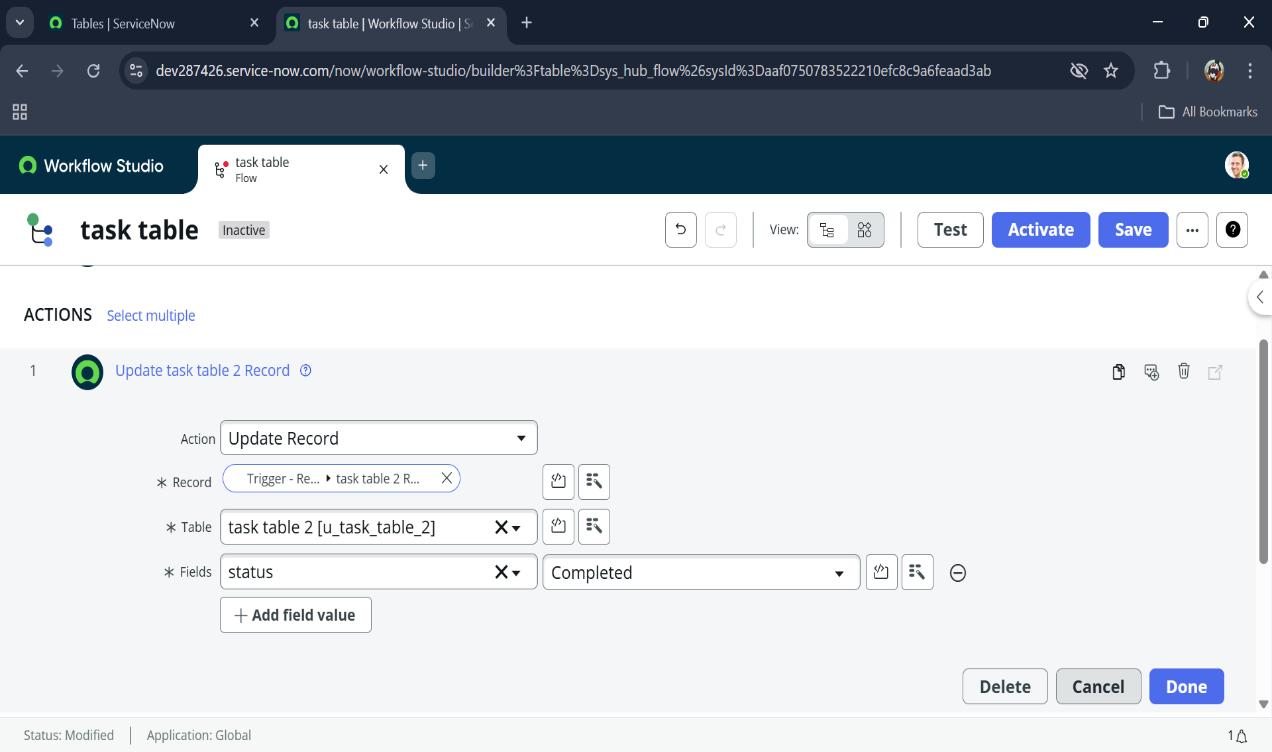
**Task:** Add "Ask for Approval" Action in Flow

**Objective:**  
To configure an approval step in the flow, requiring a designated user to approve a task before it is fully completed, ensuring accountability and proper workflow management.

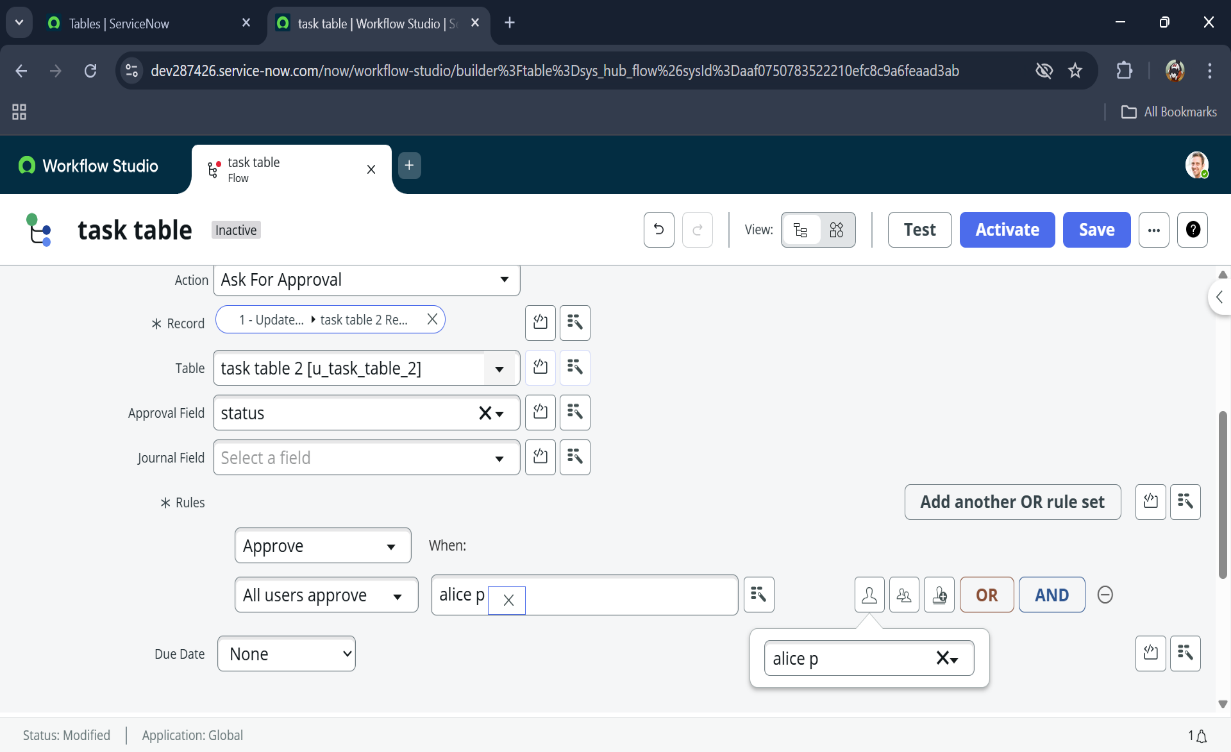
**Implementation:**

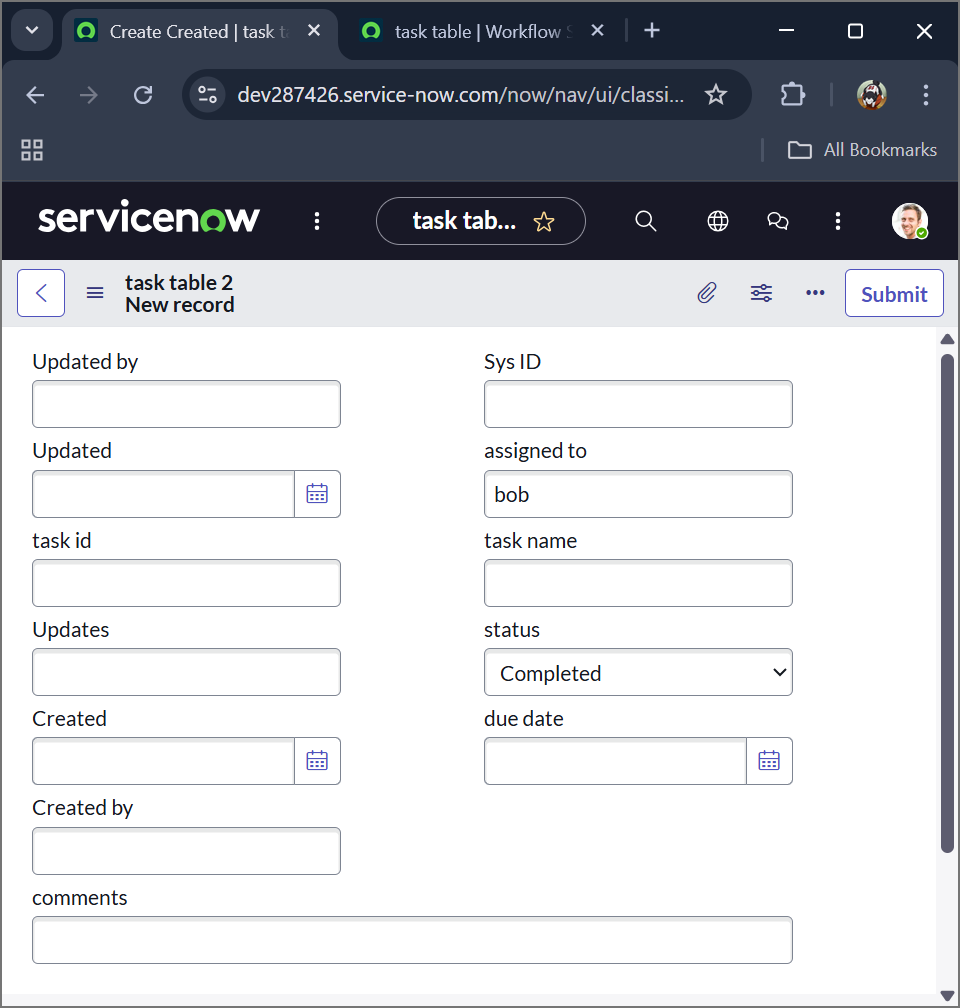
1. In the **Flow Designer**, under **Actions**, clicked **Add an Action**.
2. Searched for and selected the **Ask for Approval** action.
3. In the **Record** field, dragged the appropriate fields from the **Data Pill** navigation on the right side.
4. The **Table** field was automatically assigned based on the trigger.
5. Configured the approval details:
   * **Approve Field:** Status
   * **Approver:** Alice P
6. Clicked **Done** to finalize the action configuration.

**Outcome:**  
The flow now includes an approval step, requiring **Alice P** to approve tasks before final completion. This ensures proper verification and accountability within the task assignment process.



1. Go to application navigator search for task table.
2. It status field is updated to completed





**Task:** Approve Task via "My Approvals"

**Objective:**  
To verify and complete the approval process in the flow by approving the assigned task, ensuring that the automated workflow progresses correctly.

**Implementation:**

1. Opened the **ServiceNow** instance and navigated to the **Application Navigator**.
2. Searched for **My Approvals** and selected it under **Service Desk**.
3. Logged in or impersonated **Alice P** to view pending approvals.
4. Identified the approval request for the task triggered by the flow.
5. Right-clicked on the requested task and selected **Approve**.

**Outcome:**  
The approval request for the task was successfully completed by **Alice P**. This confirmed that the approval action in the flow is functioning as expected, allowing the automated workflow to proceed and update the task status.

