

Naan Mudhalvan
Salesforce Developer(Course)
Assignment no 1

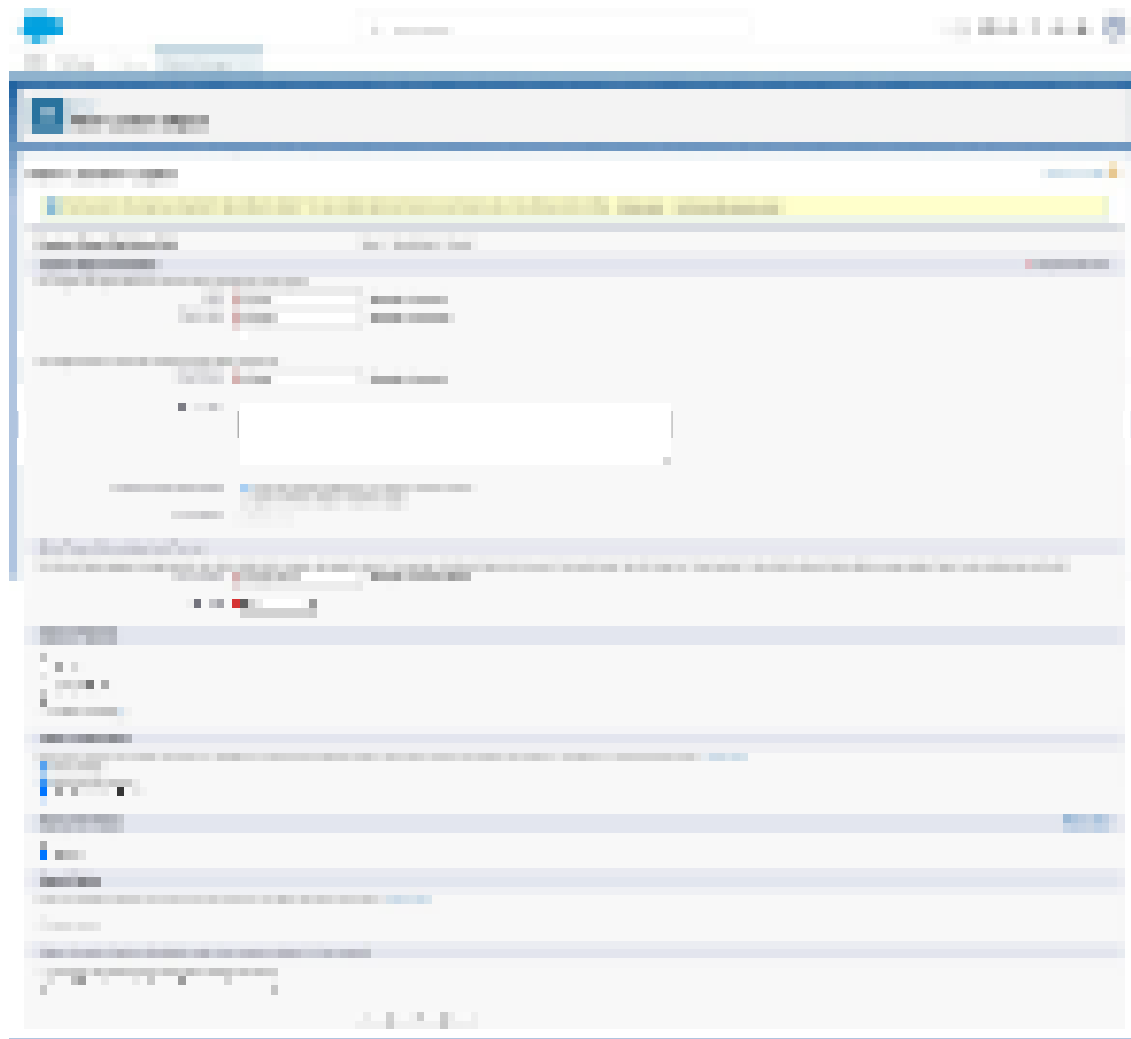
Name : Saranya M
Naan Mudhalvan id : au611420104069
Year & Dep : 4th year & CSE
Batch : 2024
Zone no : Zone 8

1.Create a Master-Detail Relationship between two Custom objects and also create a Roll Up Summary Field to Calculate total number of records.

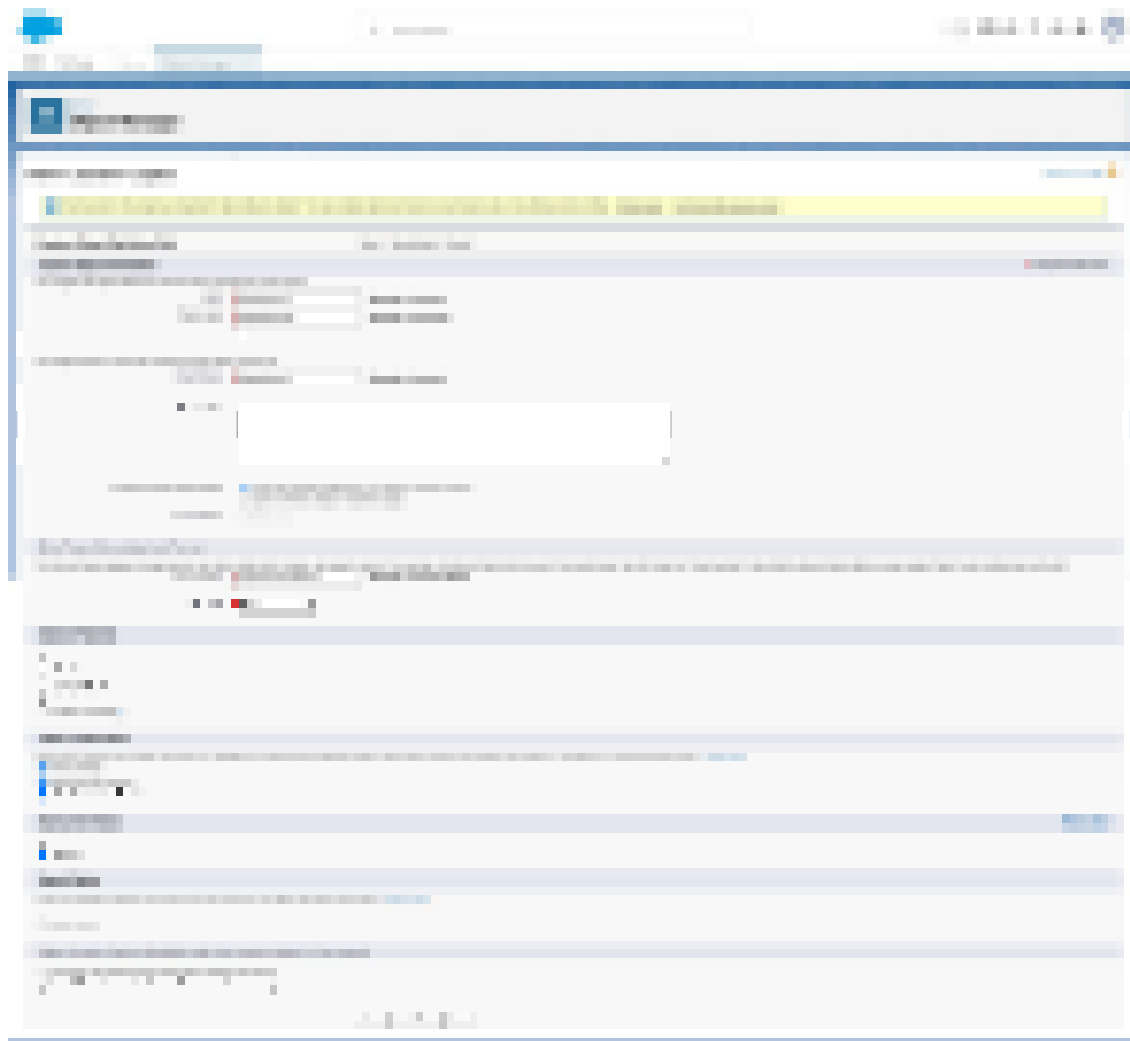
Solution:

Step 1: Create Custom Objects

Assuming you have two custom objects, let's call them "College_C" and "C Department_C". If you haven't already created these objects, you can do so by going to Setup > Object Manager > Create > Custom Object.



Second custom objects, let's call them
"Department_C"

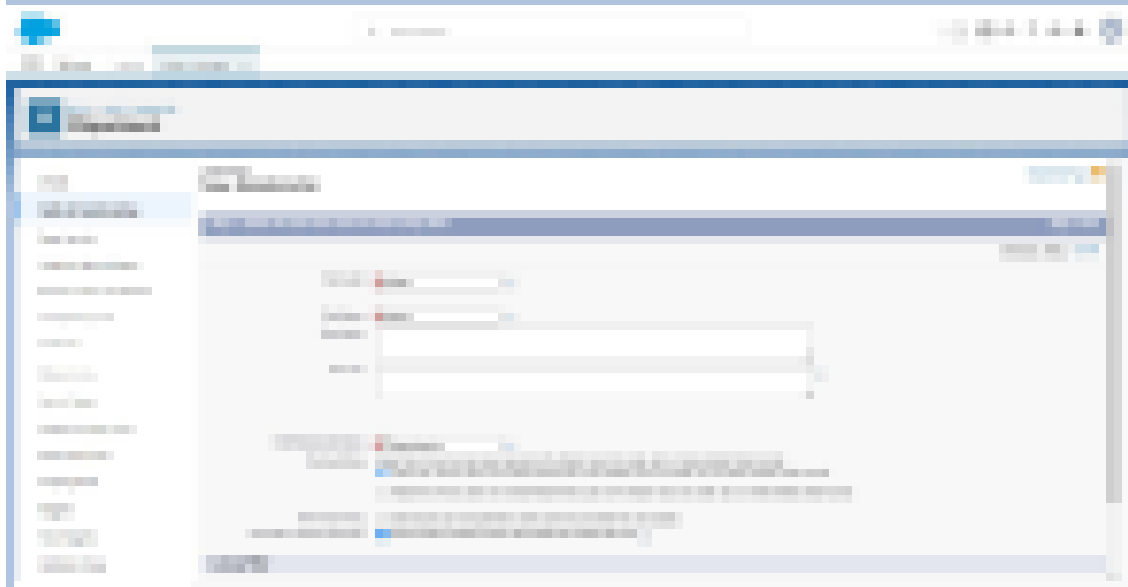
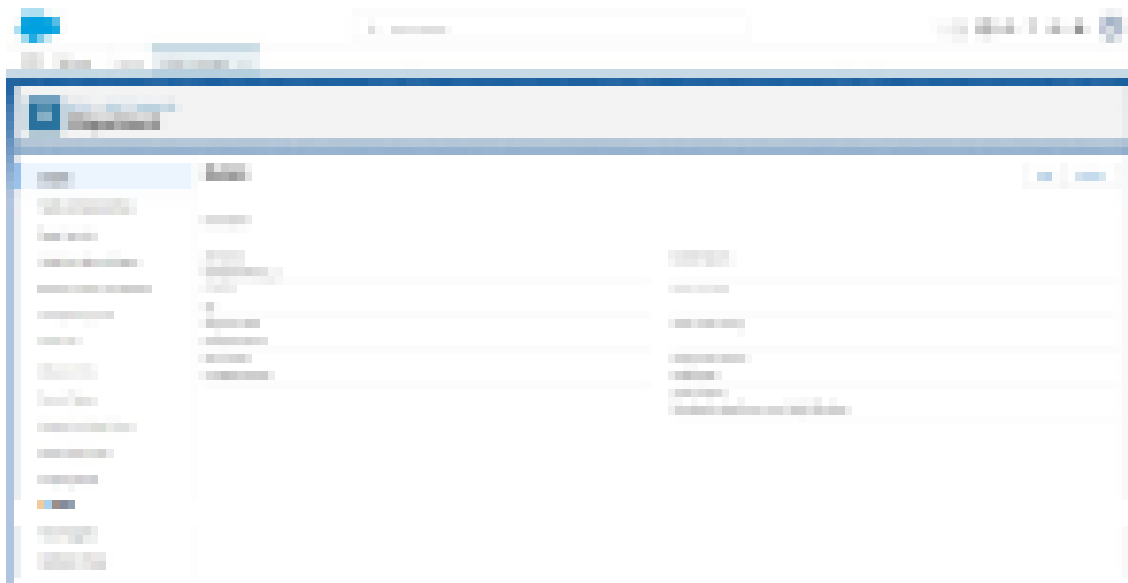


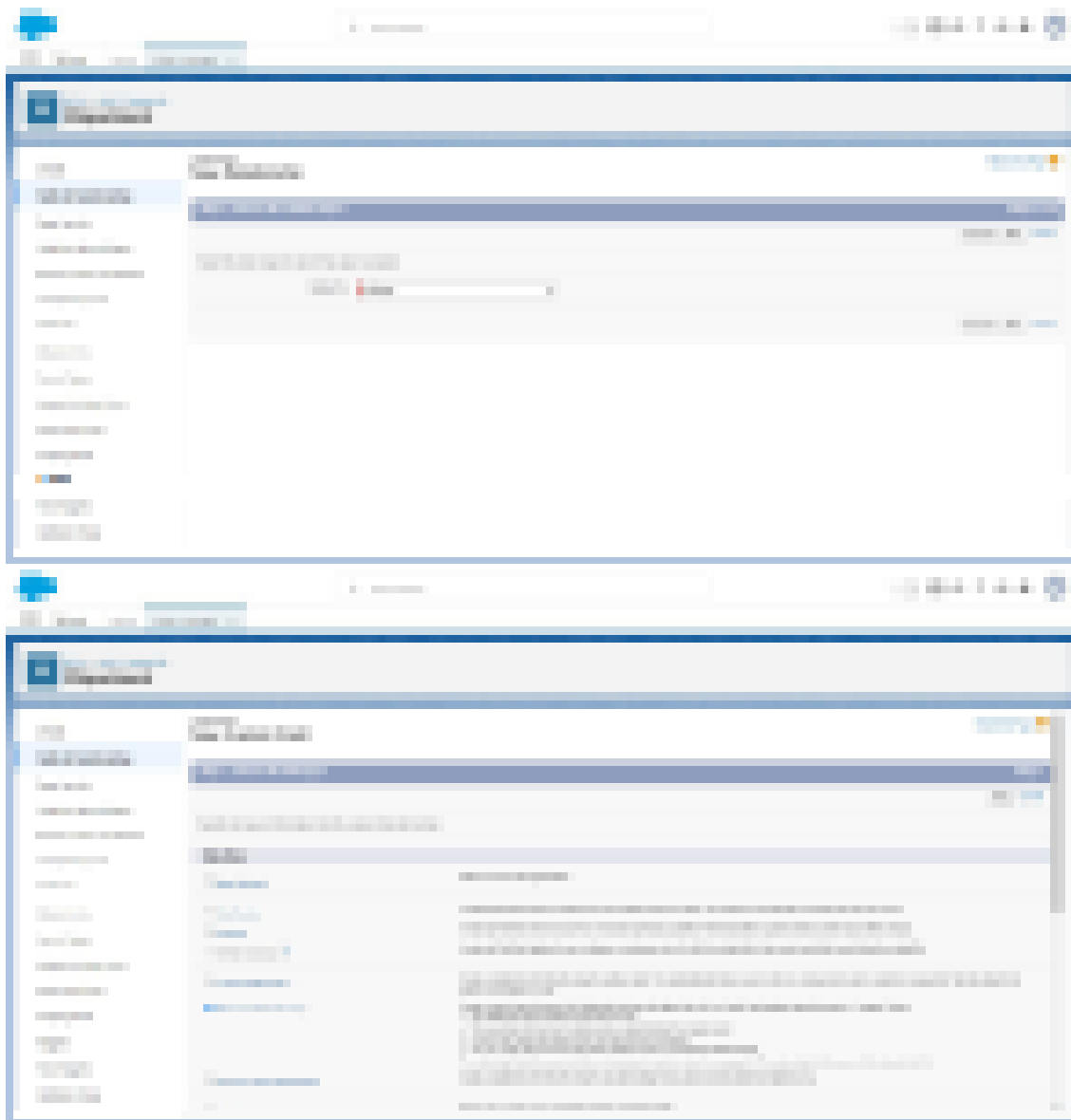
Step 2: Create a Master-Detail Relationship

To create a Master-Detail relationship between these two custom objects, follow these steps:

1. Go to Setup > Object Manager.
2. Click on "College__c" to open its settings.
3. In the left sidebar, click on "Fields & Relationships."

4. Click the "New" button to create a new custom field.
5. Choose "Master-Detail Relationship" as the data type.
6. Enter a label for the relationship, e.g., "Department __c."
7. Choose " Department__c" as the related object.
8. Configure other settings as needed and click "Next."
9. Specify the field-level security and add it to relevant page layouts.
10. Click "Next" and "Save" to create the relationship.



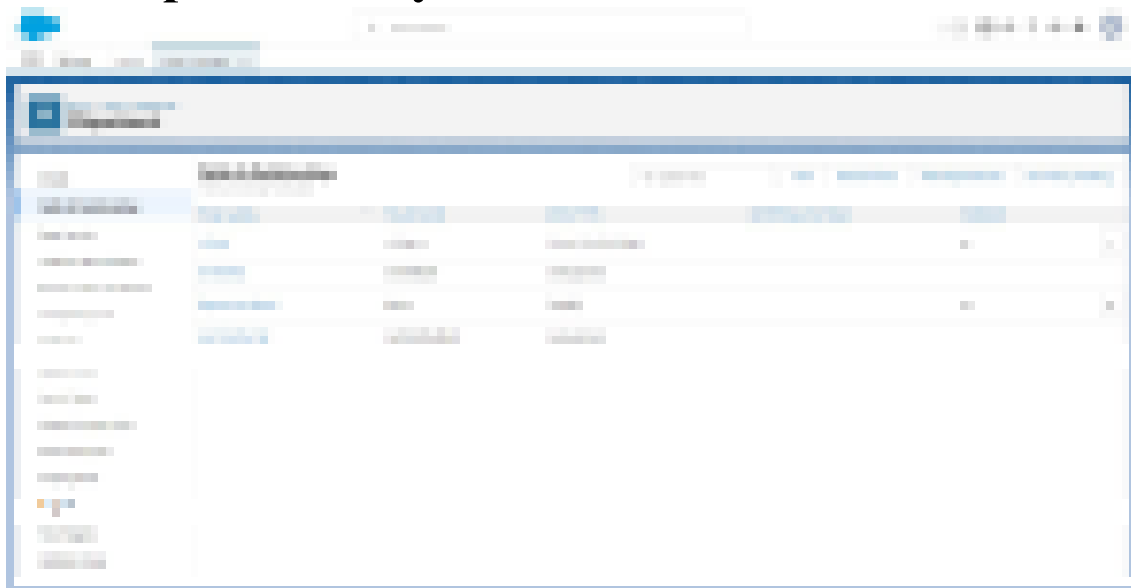


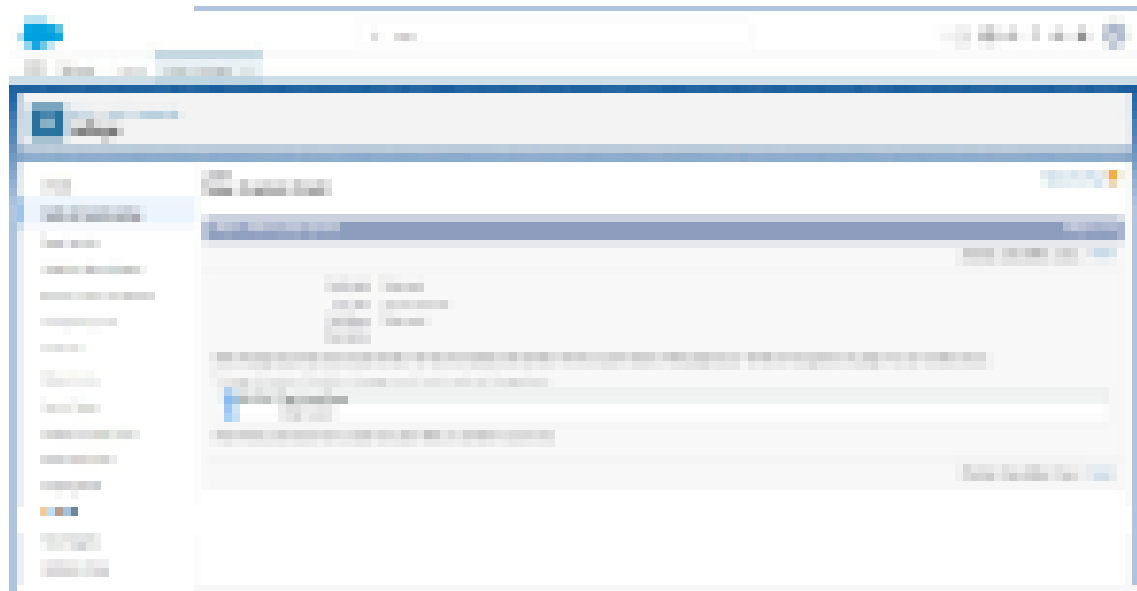
Step 3: Create the Roll-Up Summary Field

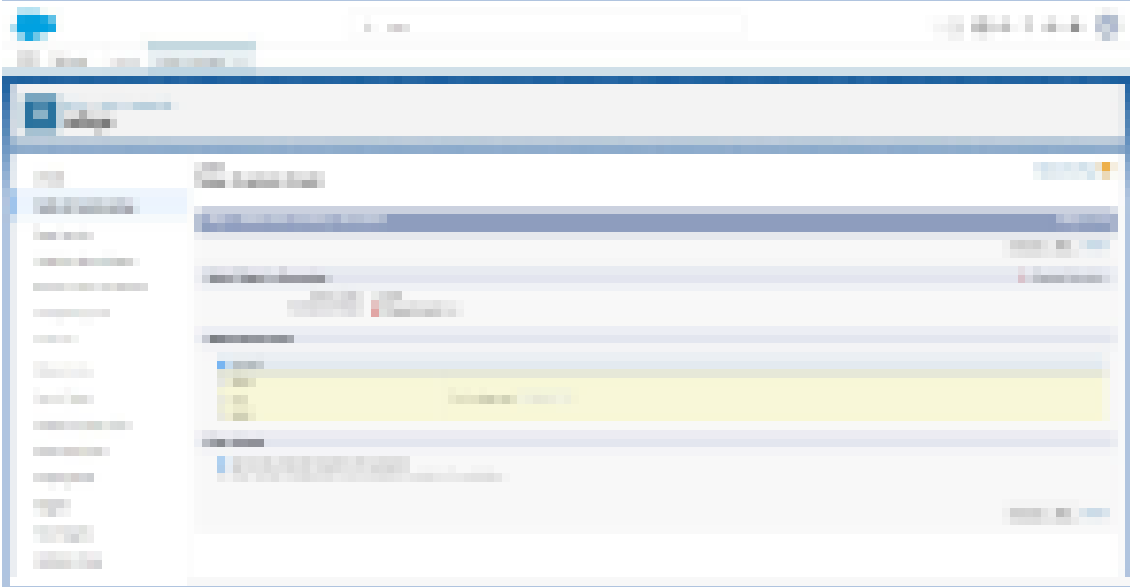
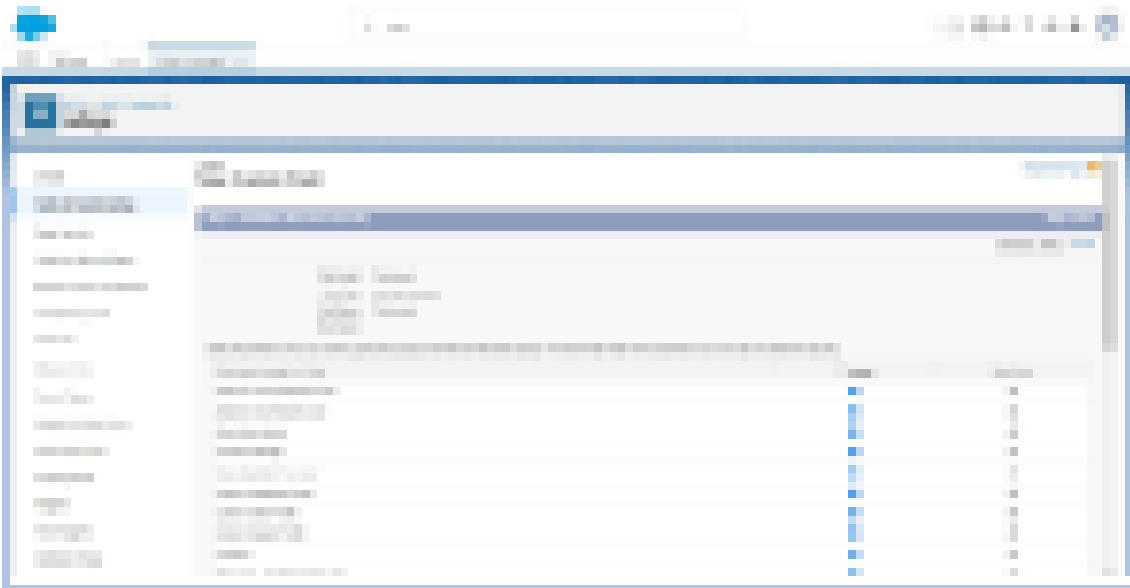
Now, let's create a Roll-Up Summary Field on the "College_C" to calculate the total number of related records in "Department__C":

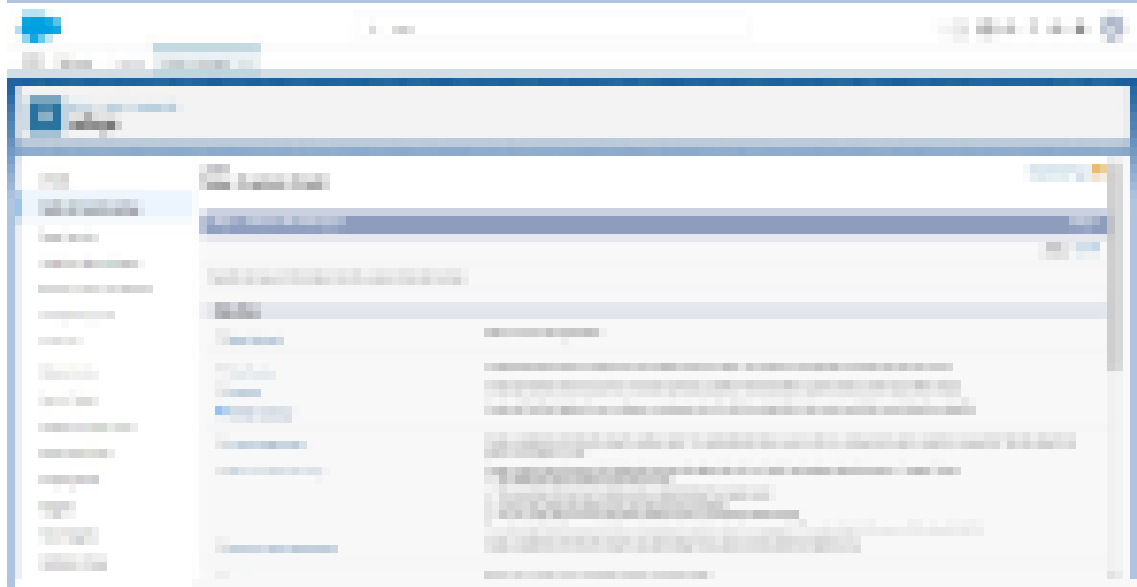
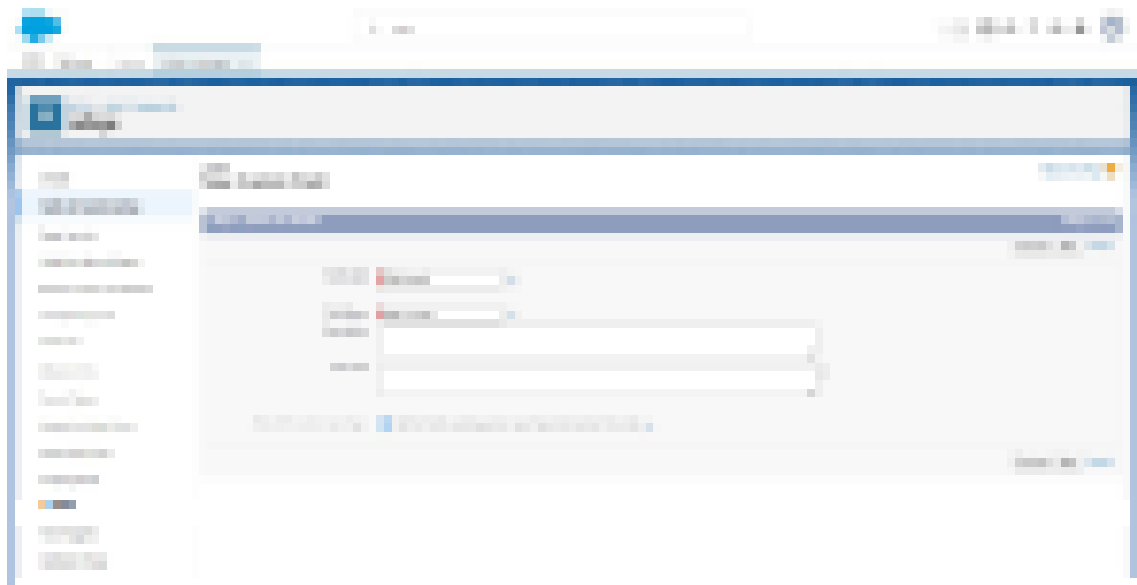
- 1. Still on the "College__c" settings, go to "Fields & Relationships."**

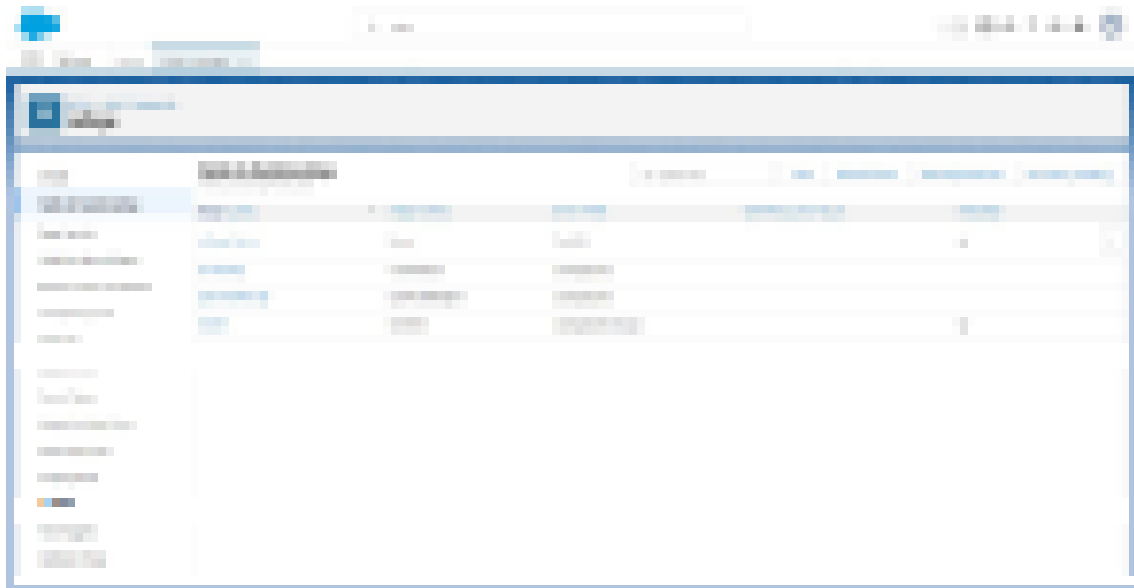
- 2. Click the "New" button to create a new custom field.**
- 3. Choose "Roll-Up Summary" as the data type.**
- 4. Enter a label for the field, e.g.,**
- 5. Choose "Count" as the Roll-Up Type.**
- 6. Select " Department__c" as the object to roll up information from.**
- 7. Specify the filter criteria if you want to filter the related records.**
- 8. Configure other settings as needed and click "Next."**
- 9. Specify the field-level security and add it to relevant page layouts.**
- 10. Click "Next" and "Save" to create the Roll-Up Summary Field.**









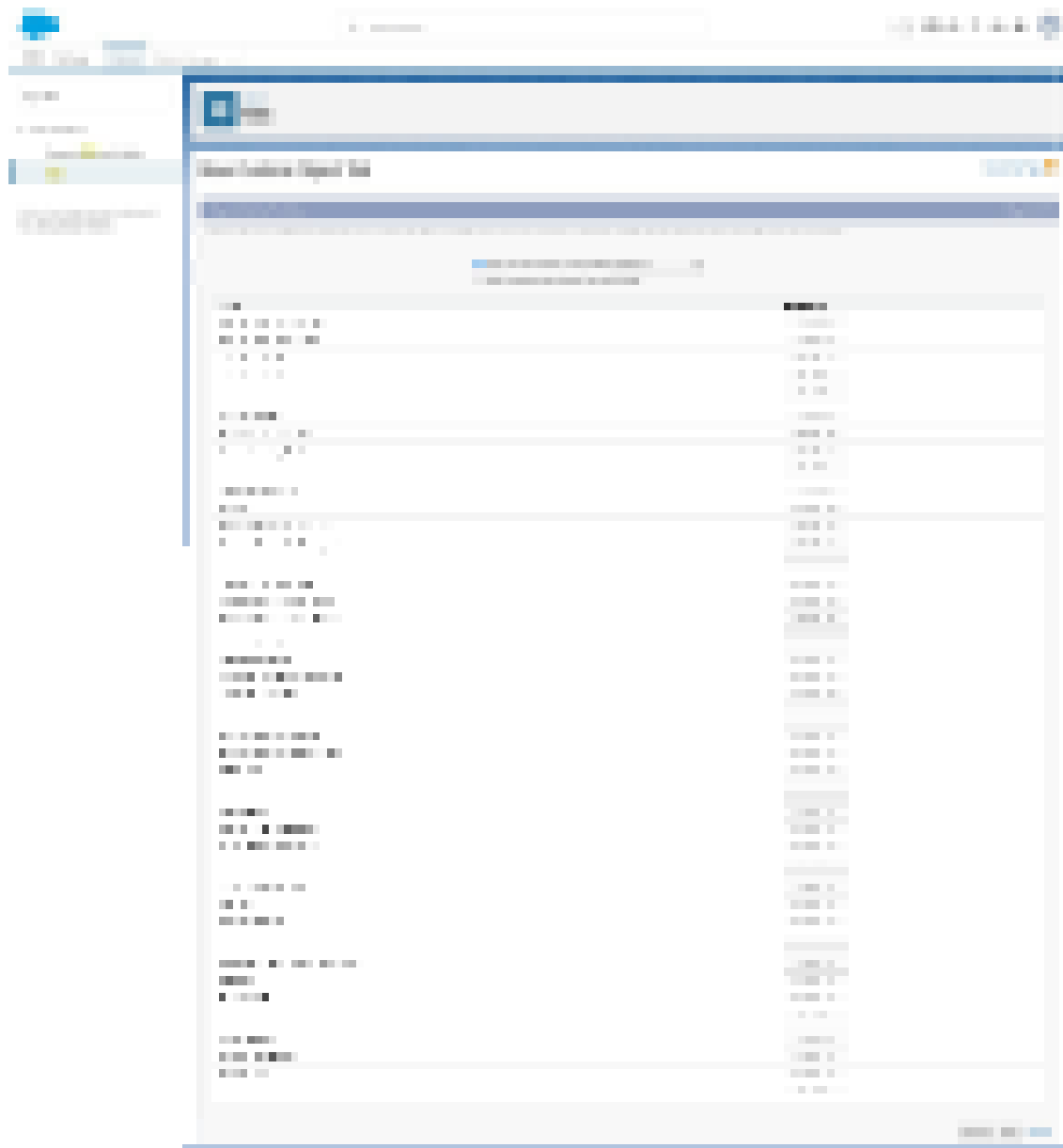


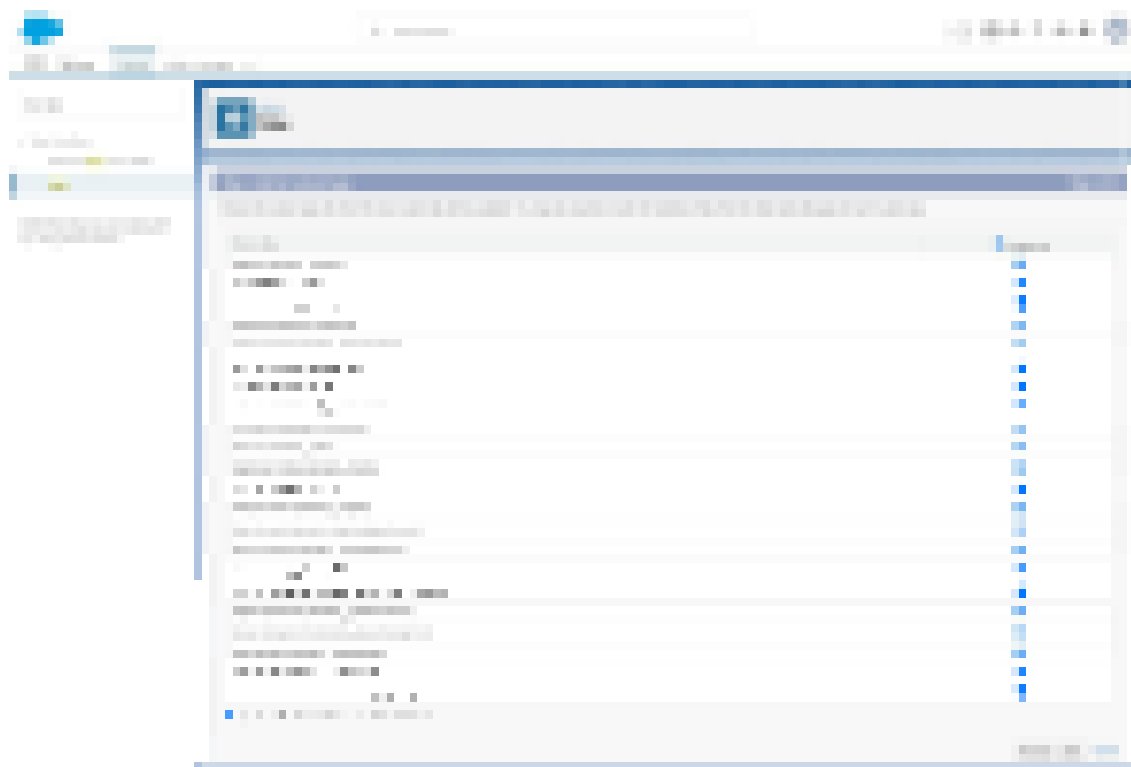
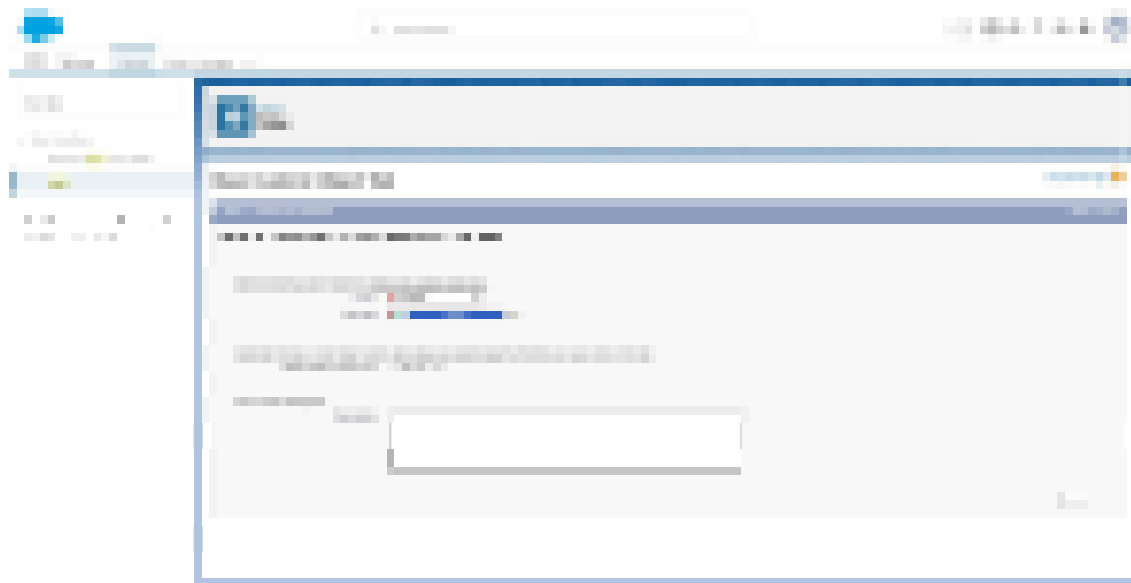
Step 4: Create a Lightning App

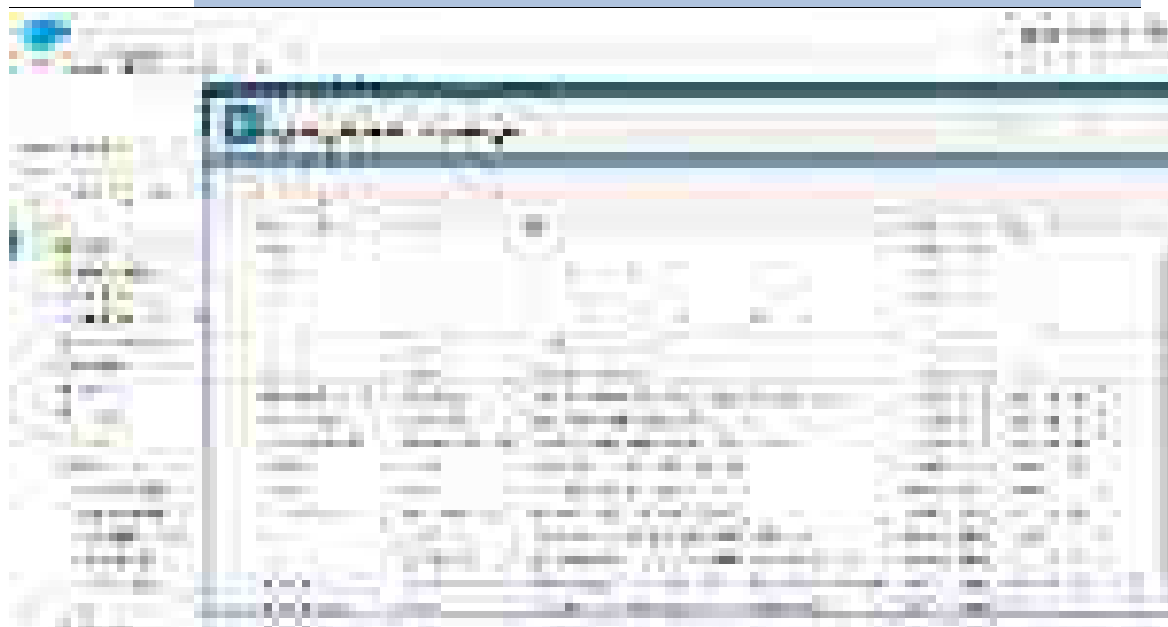
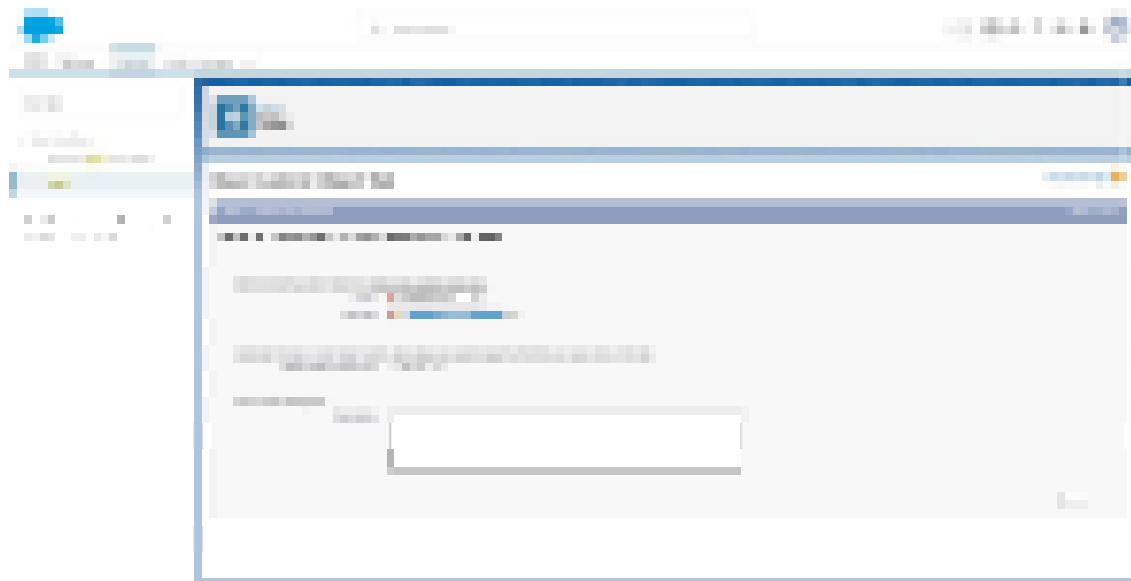
- 1.Type and select "App Manager."**
- 2.Click "New Lightning App."**
- 3.Fill in basic information (Name, Developer Name, Description).**
- 4.Choose the App Type (Standard, Console, Custom).**
- 5.Customize the Logo and Colour Scheme.**
- 6.Configure Navigation Items (objects to appear in the app's menu).**
- 7.Set the App Visibility (default access).**
- 8.Optionally, choose Record Pages (Lightning Record Pages).**
- 9.Review and Save the app.**

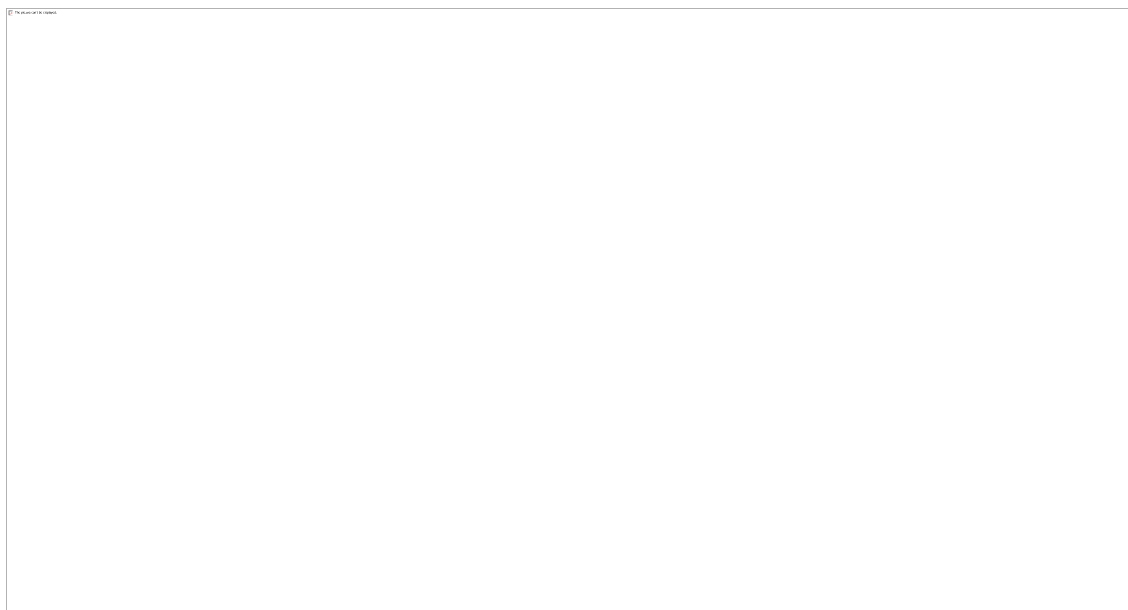
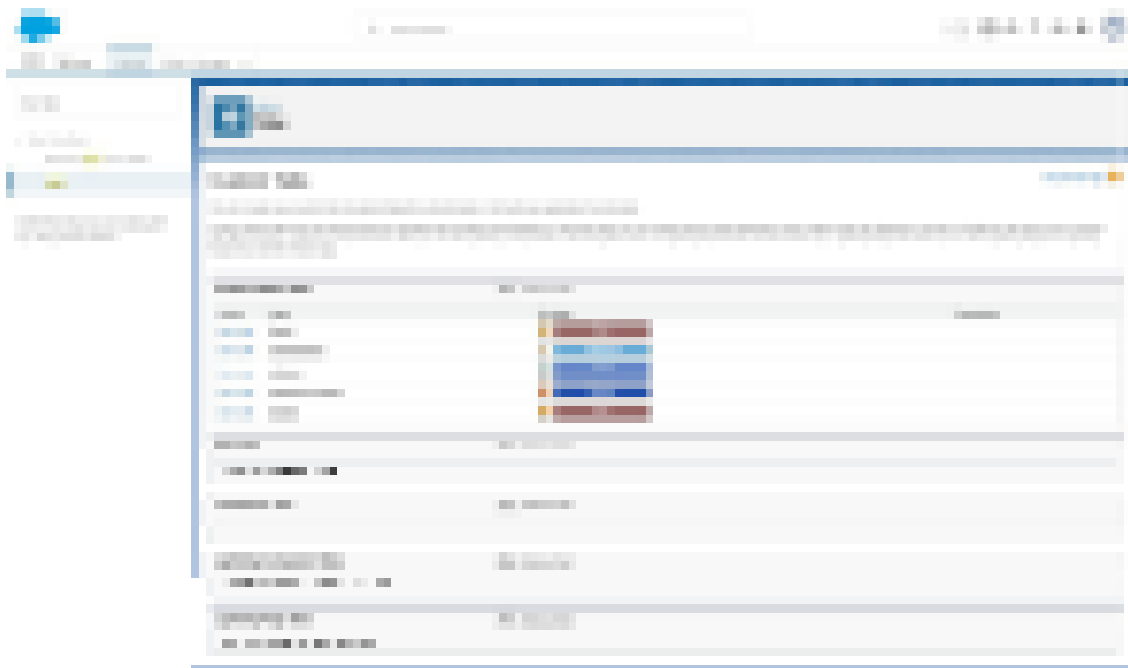
10. Assign the app to users or profiles.

11. Test the app with the assigned users.





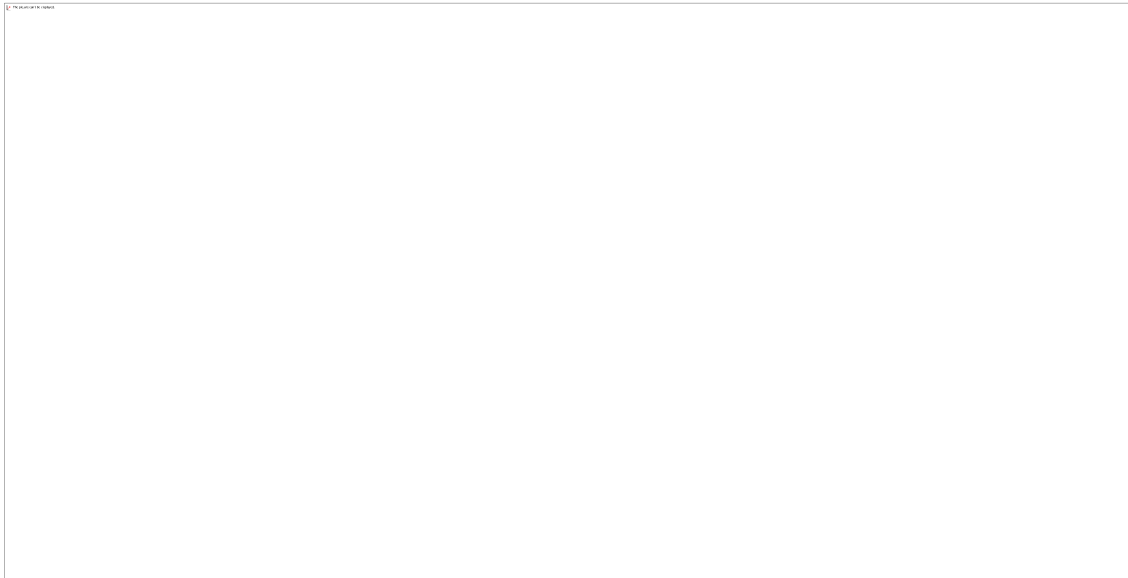


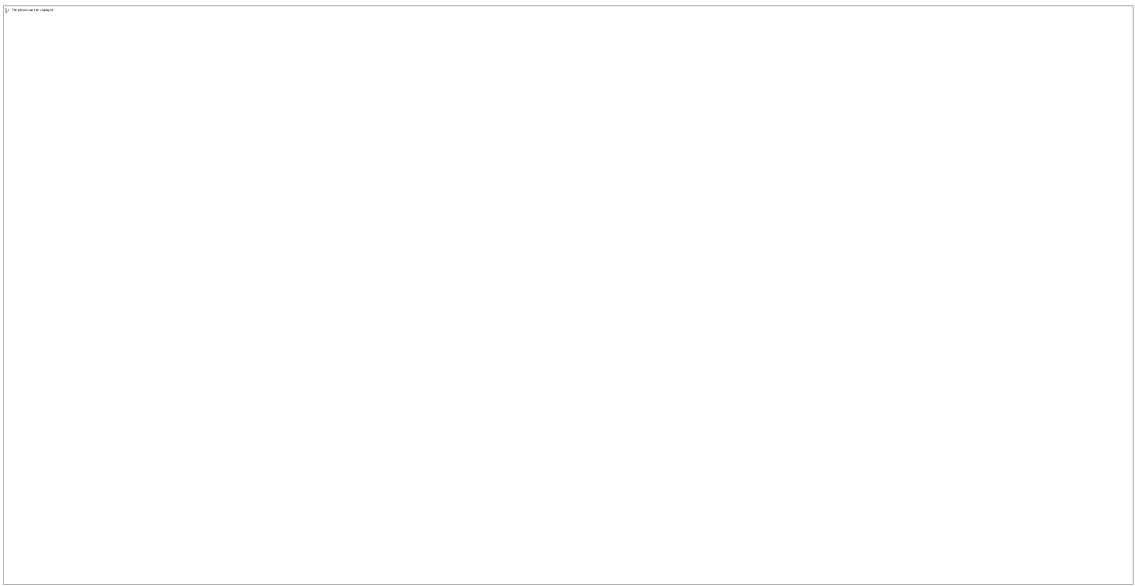
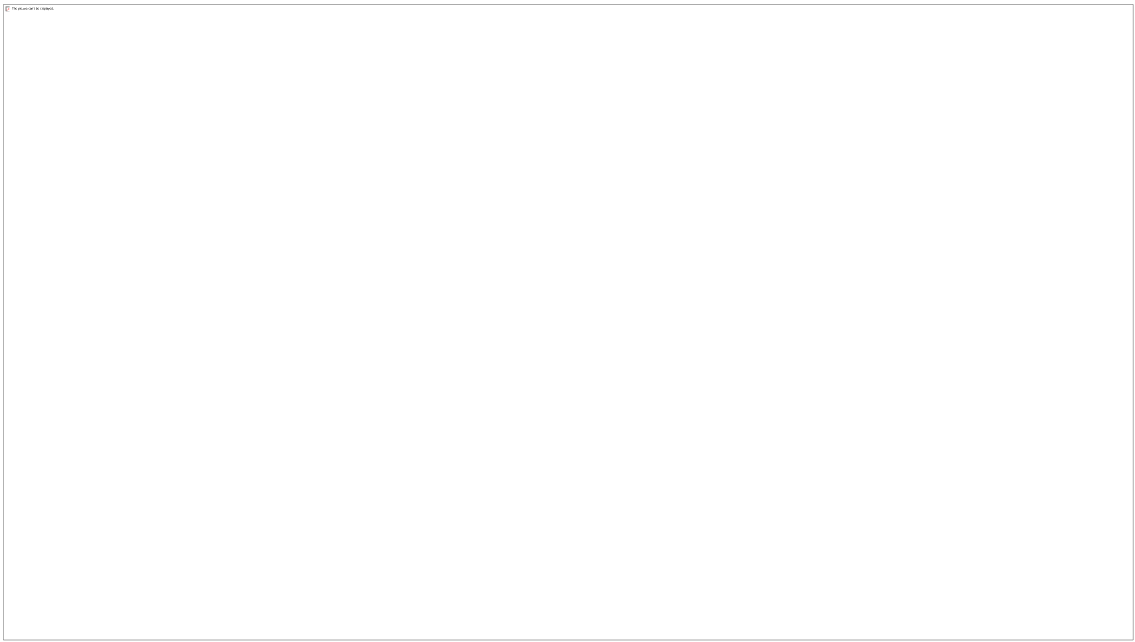


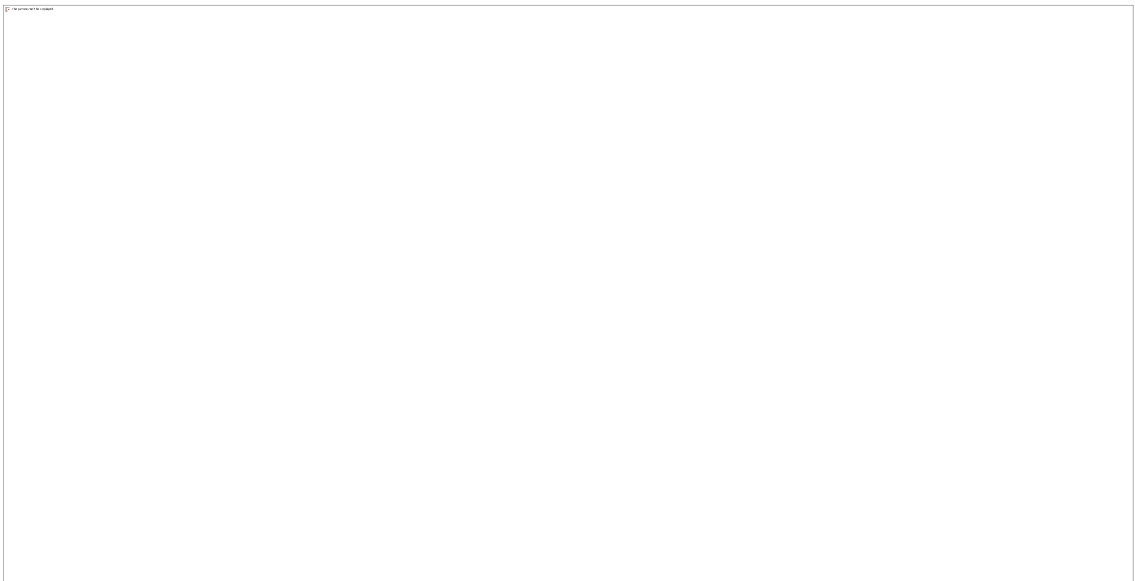
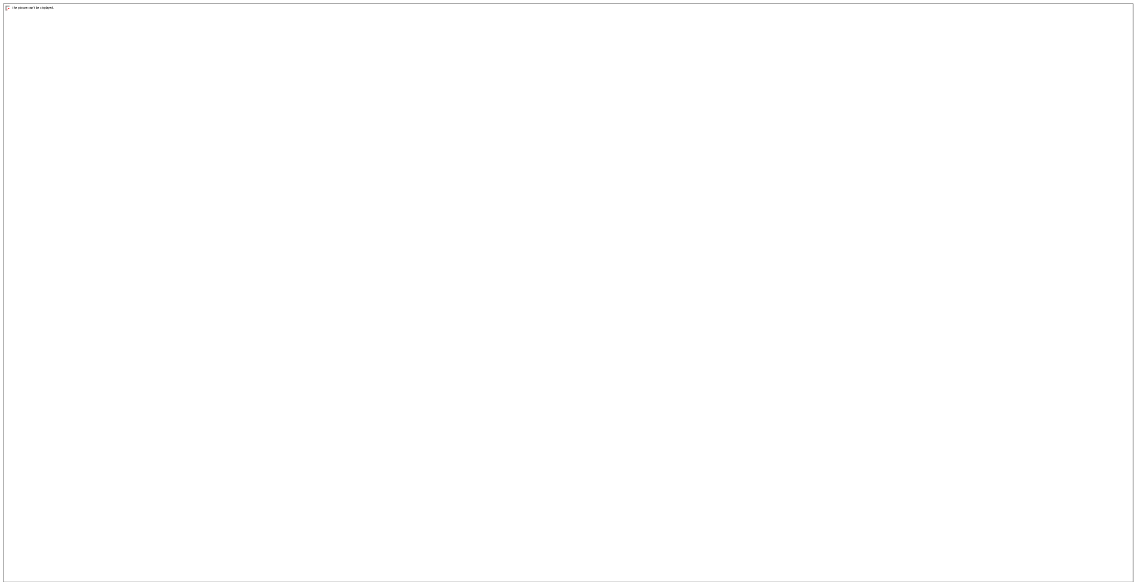
Conclusion:

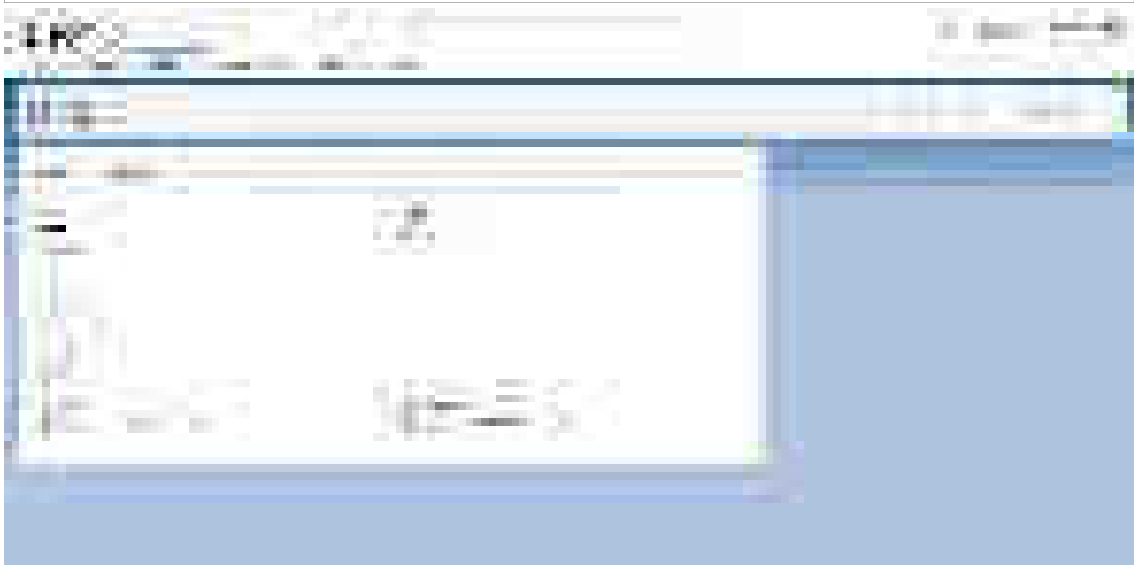
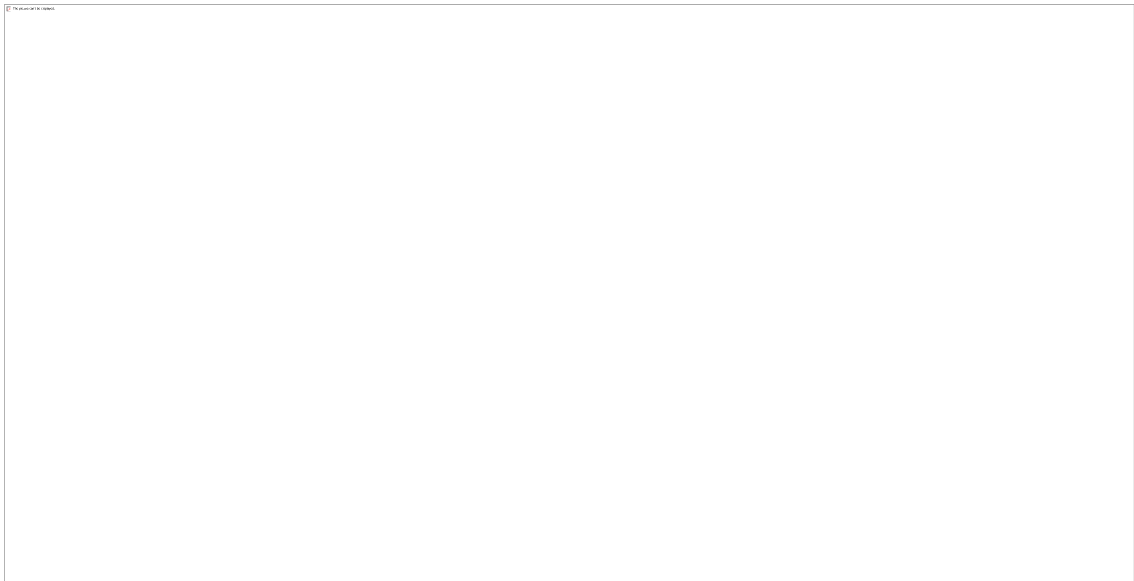
Now, whenever you create or update a record in the "Department__c" related to a "College__c," the "TotalCount__c" field on the "College__c" will automatically update to show the total number of related records.

Remember to adjust field-level security, validation rules, and page layouts as needed to ensure that your custom objects and fields are appropriately configured for your organization's requirements.





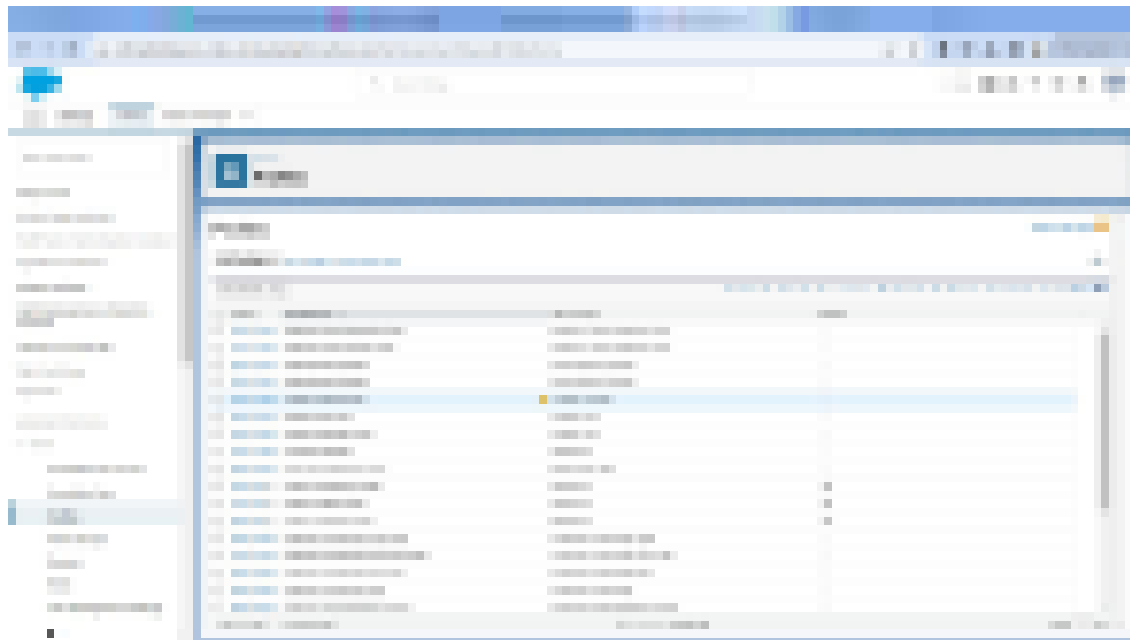




2. If there is 2 user, User A and User B in the organisation and we want in Account object that User A should not see the User B Record and user B should not see User A record then apply the Security for the users.

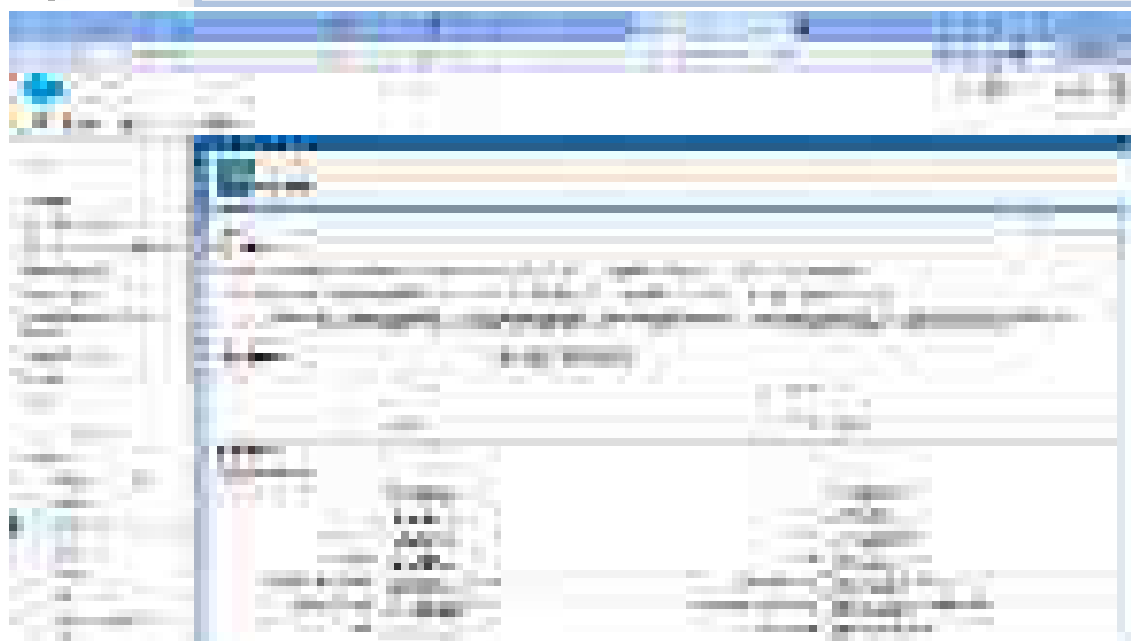
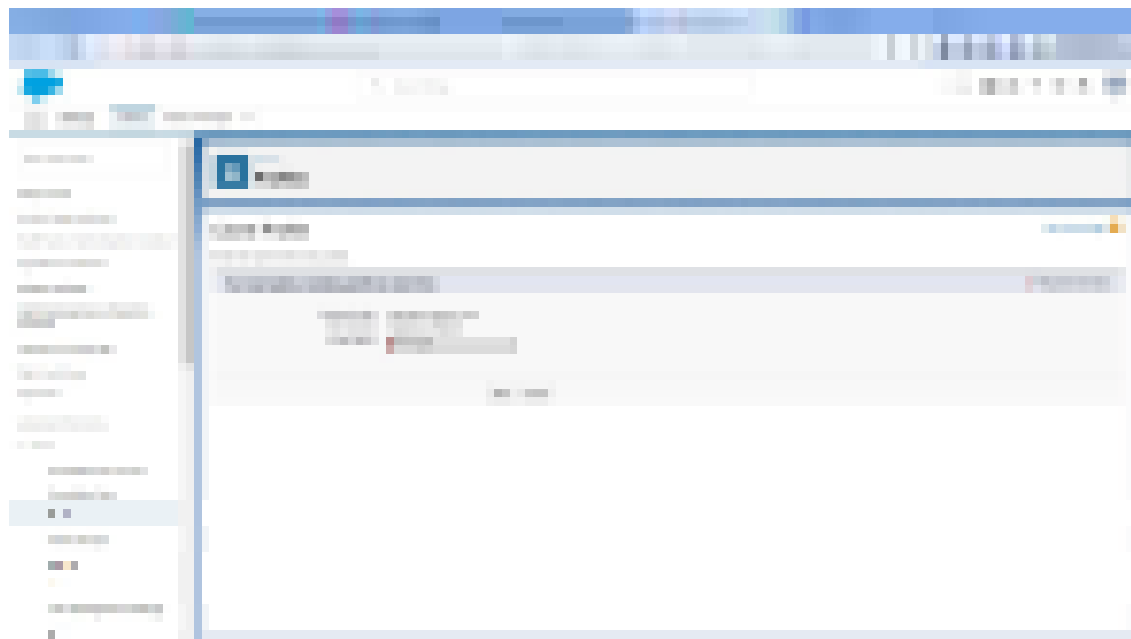
Solution:

Step 1: Create two separate custom profiles, one for User A and one for User B.

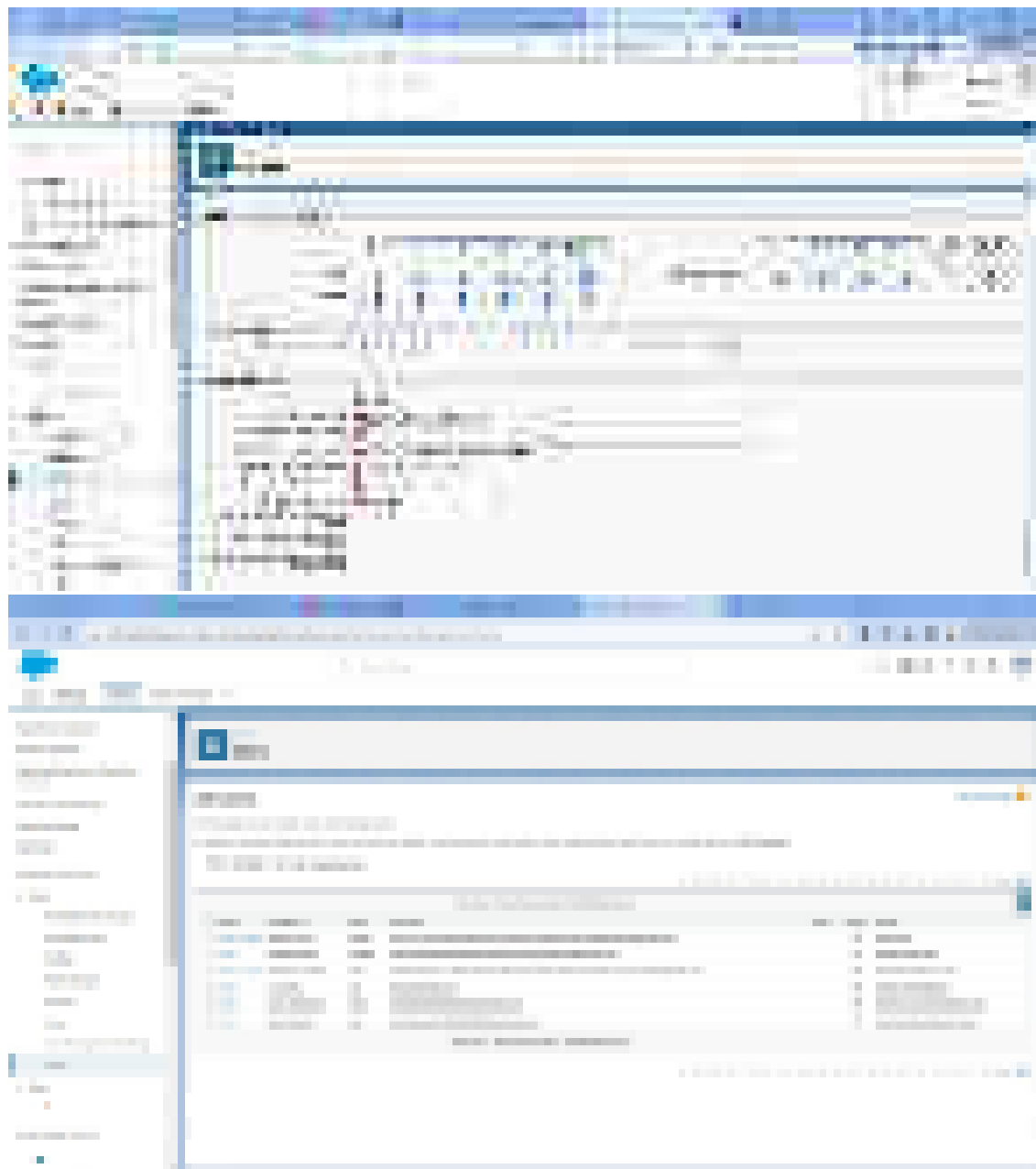


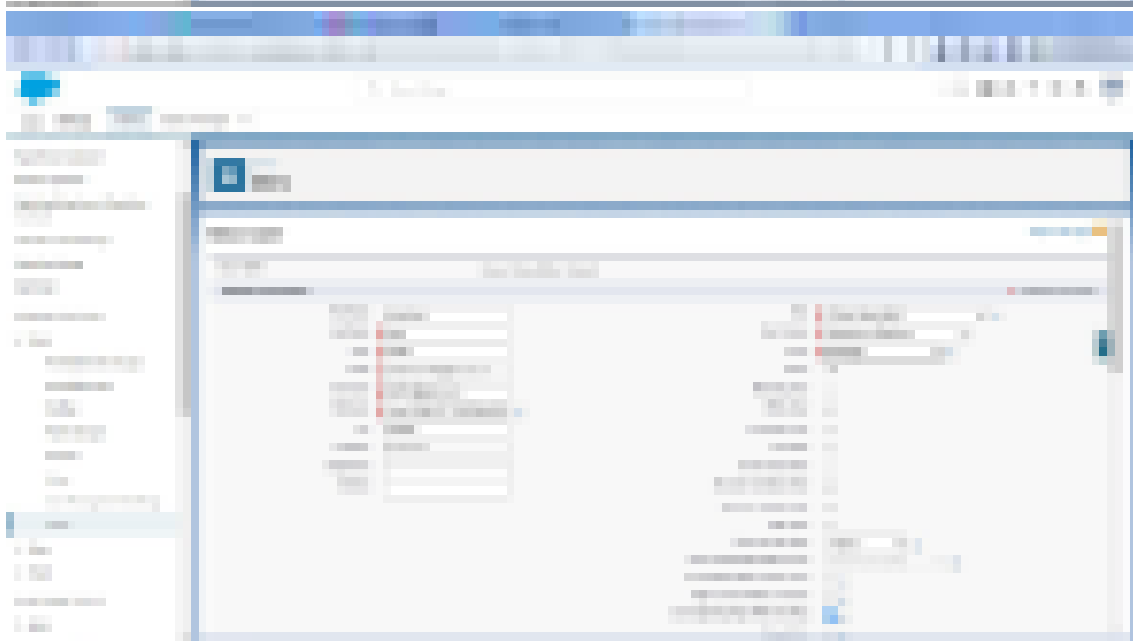
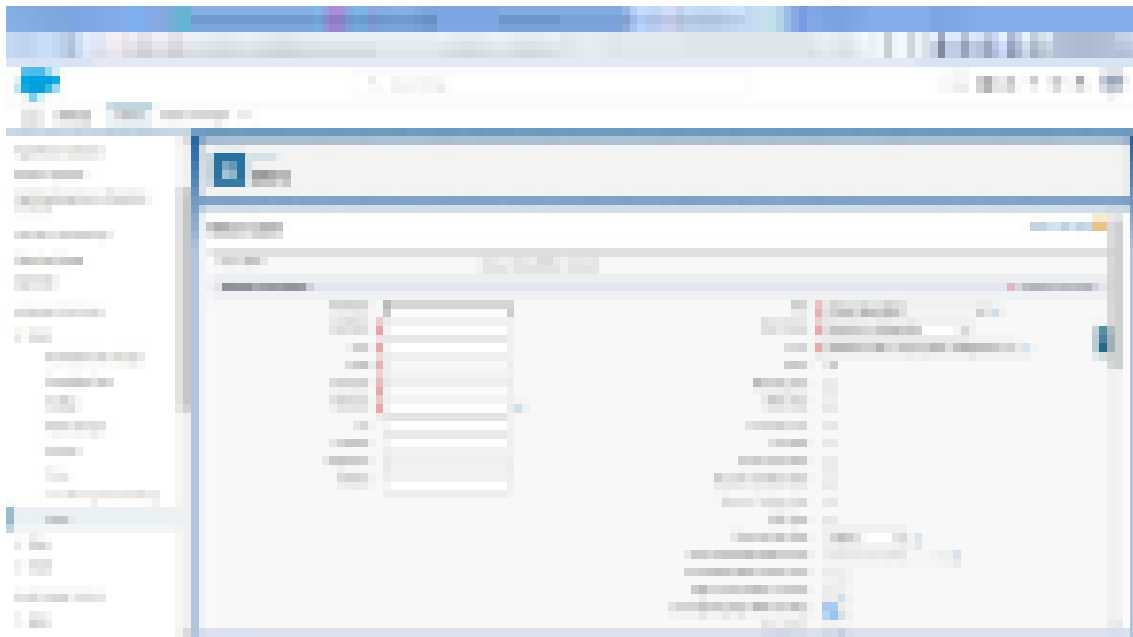


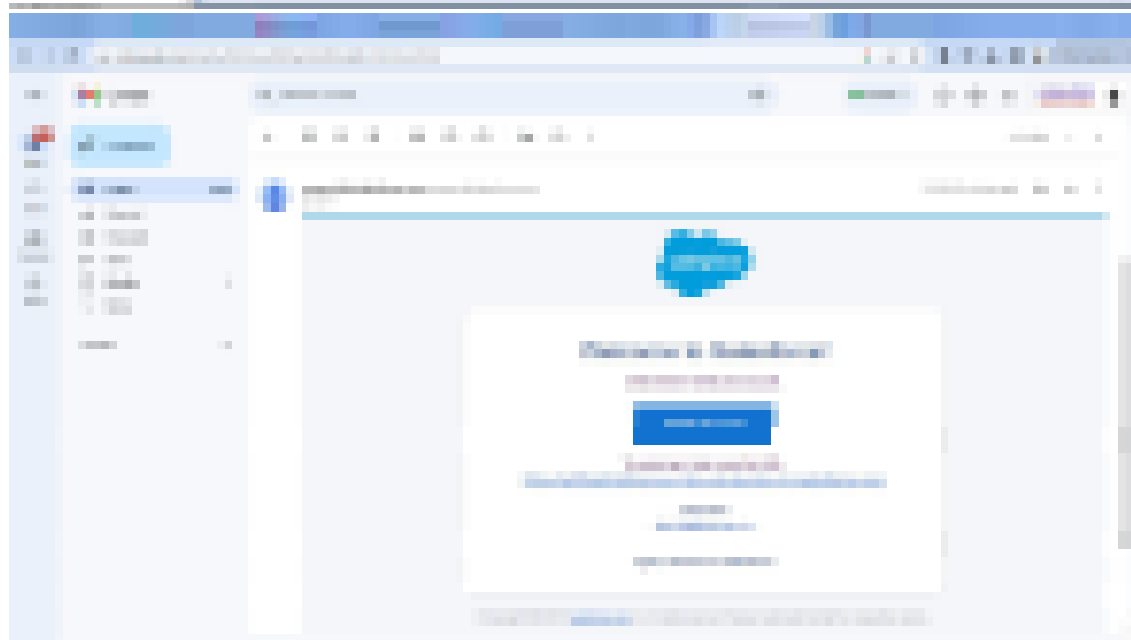
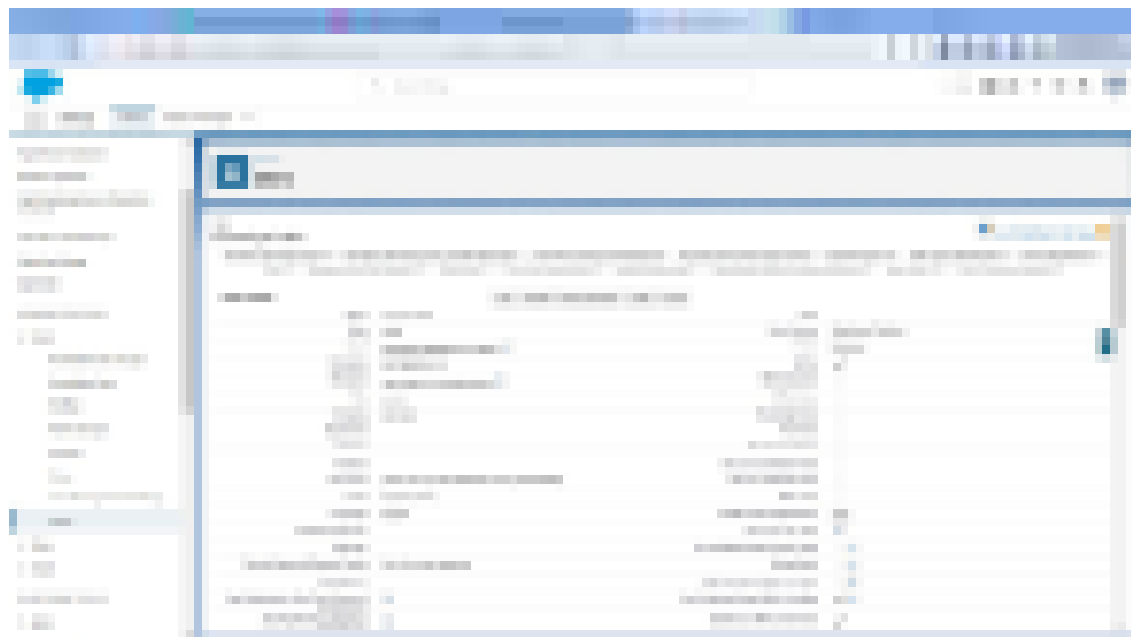
The screenshot shows a web application interface. On the left, there is a sidebar with a list of navigation links, including "Home", "About", "Contact", "Services", "Products", "Blog", "FAQ", and "Privacy Policy". The main content area displays a list of items, each with a thumbnail image and a title. The items are arranged in a grid. The top of the page has a header with a logo and a navigation menu. The bottom of the page has a footer with additional information.

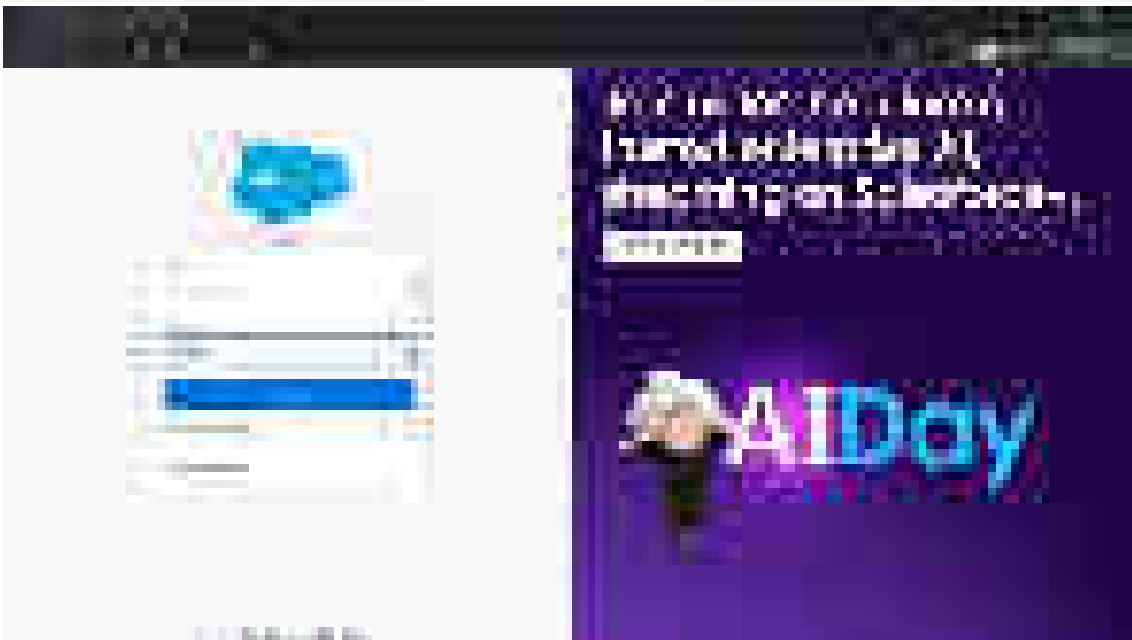
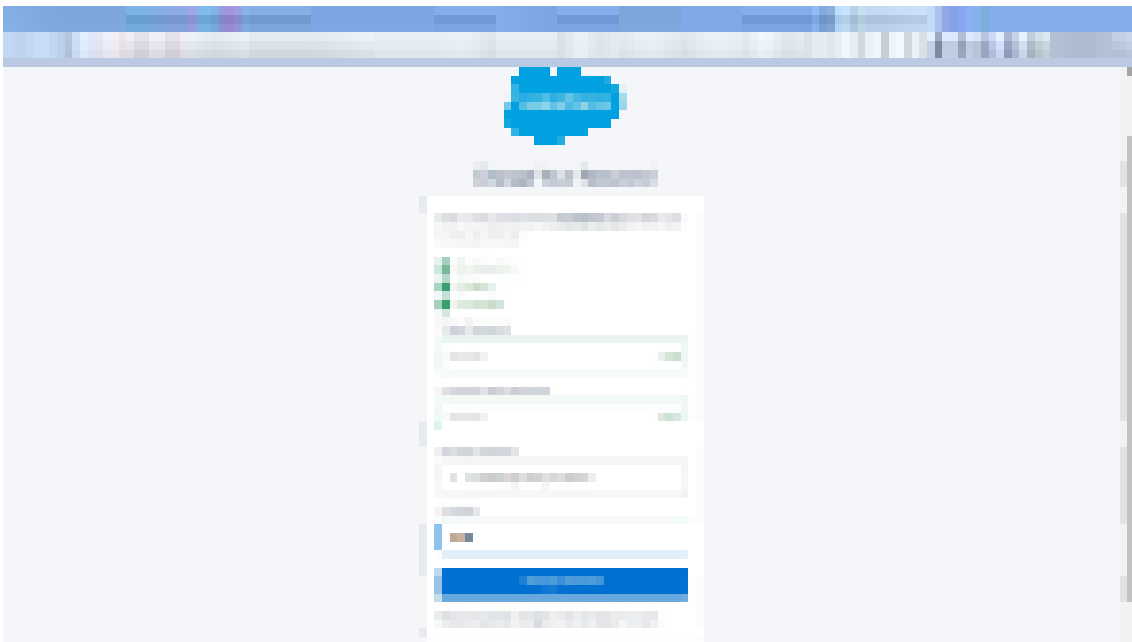


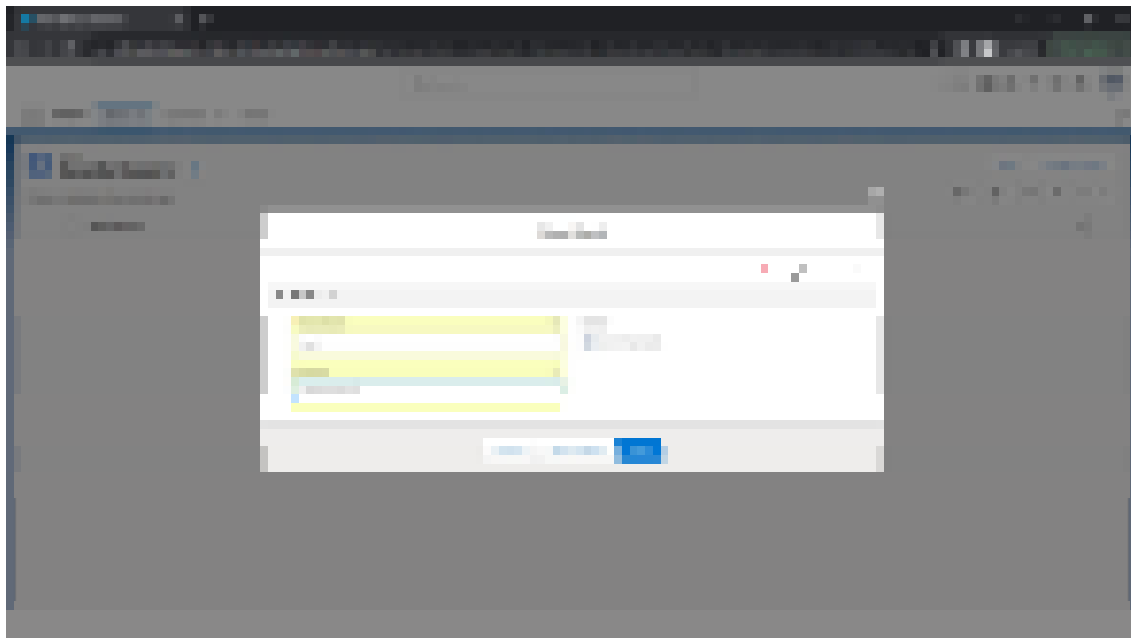
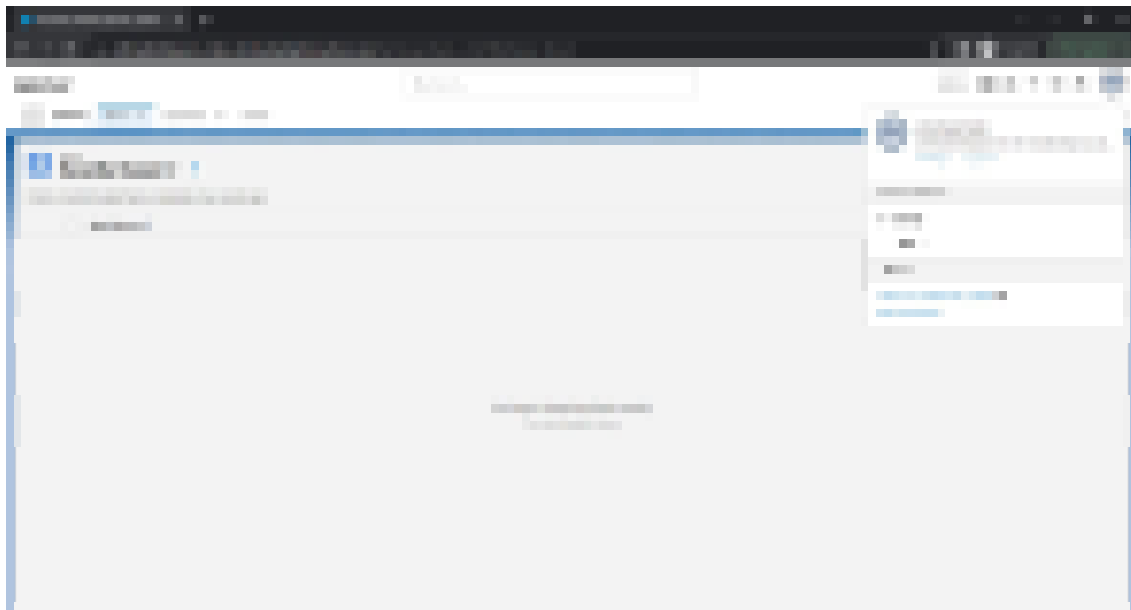


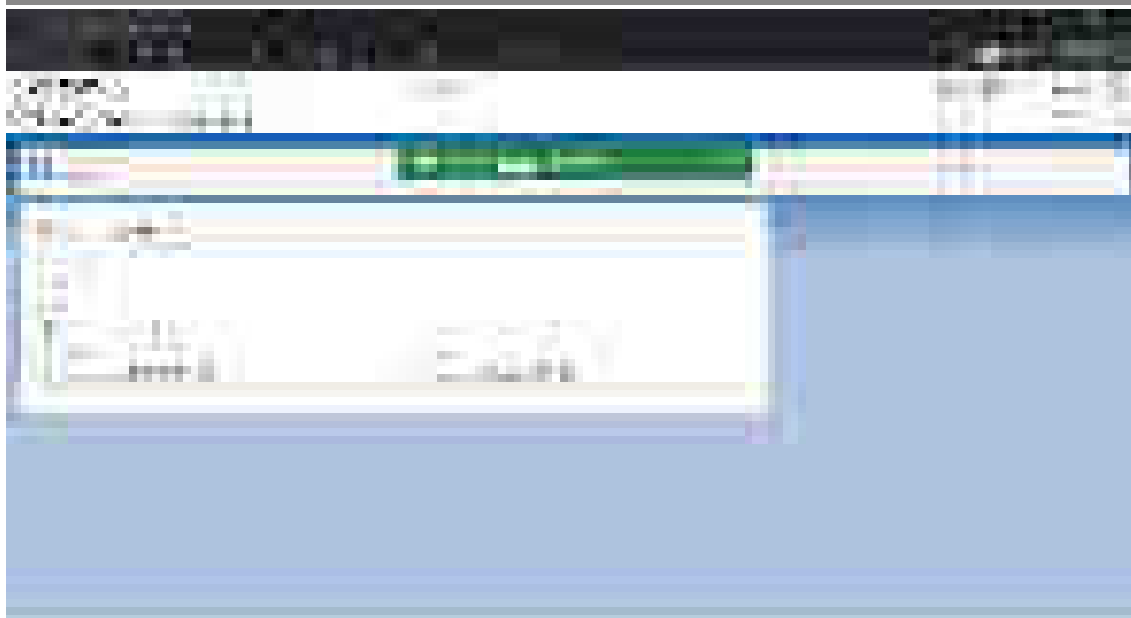
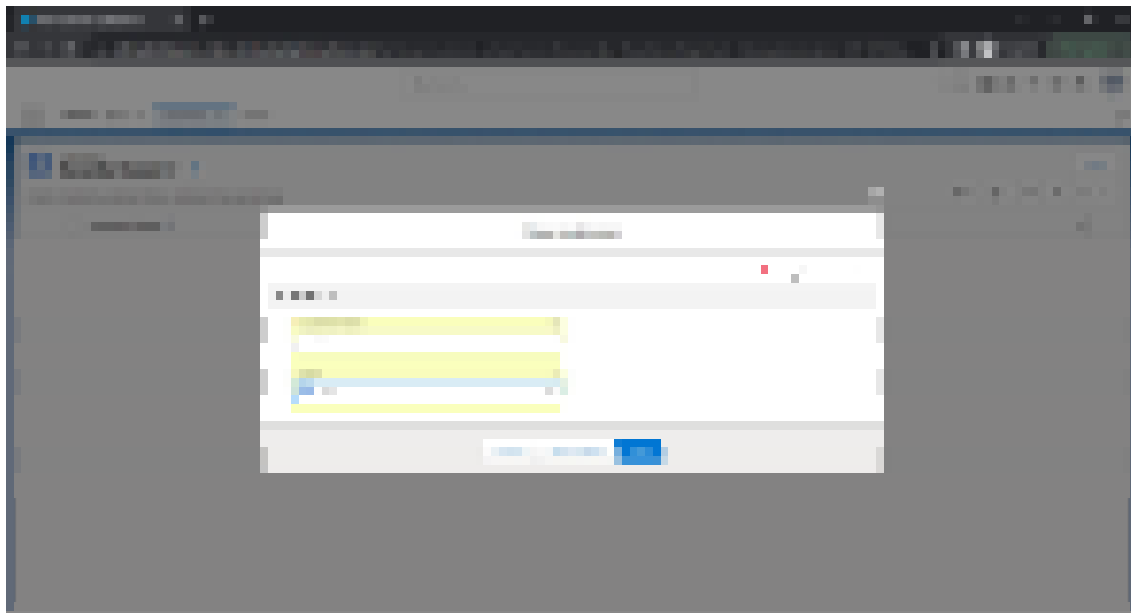


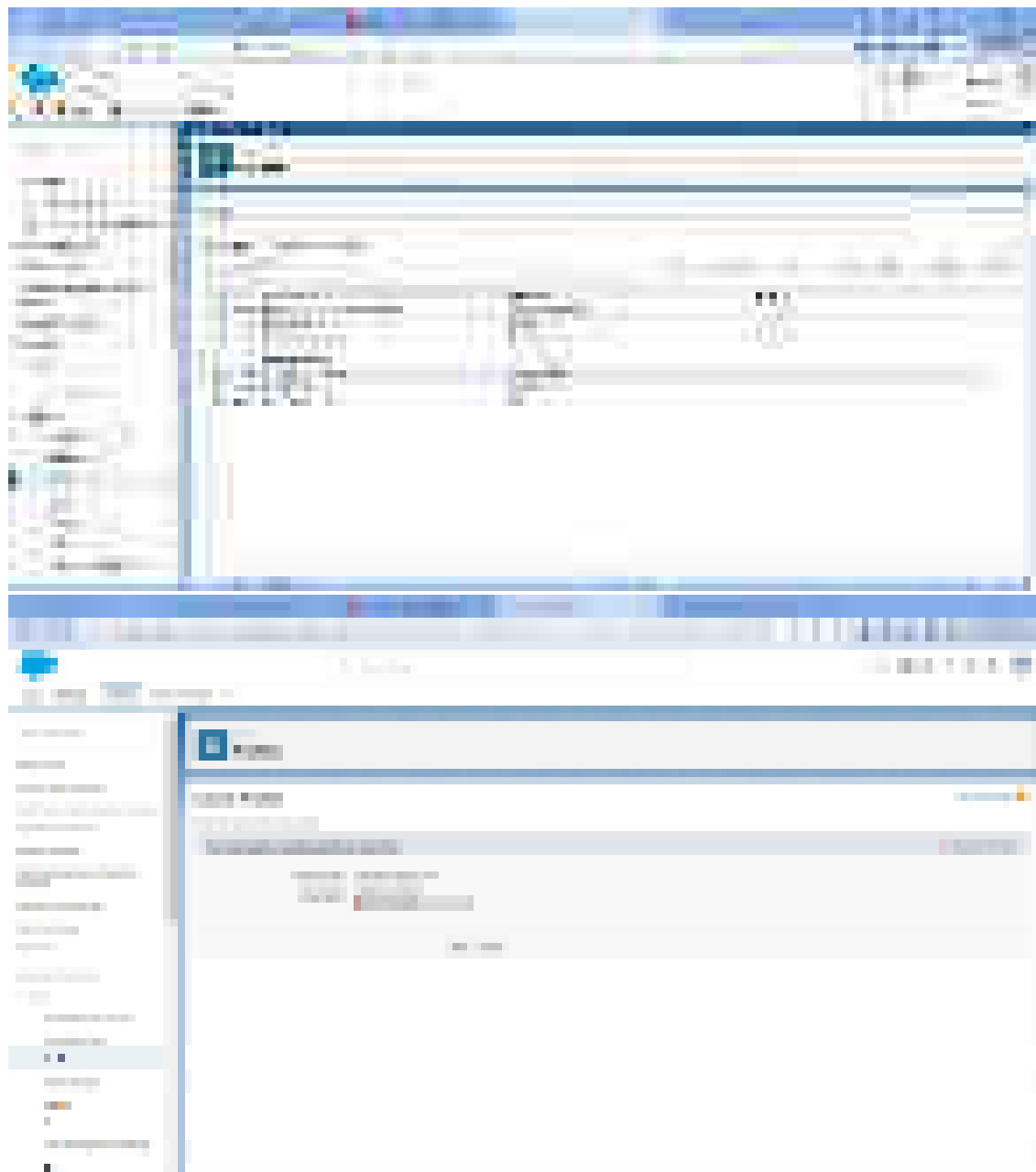


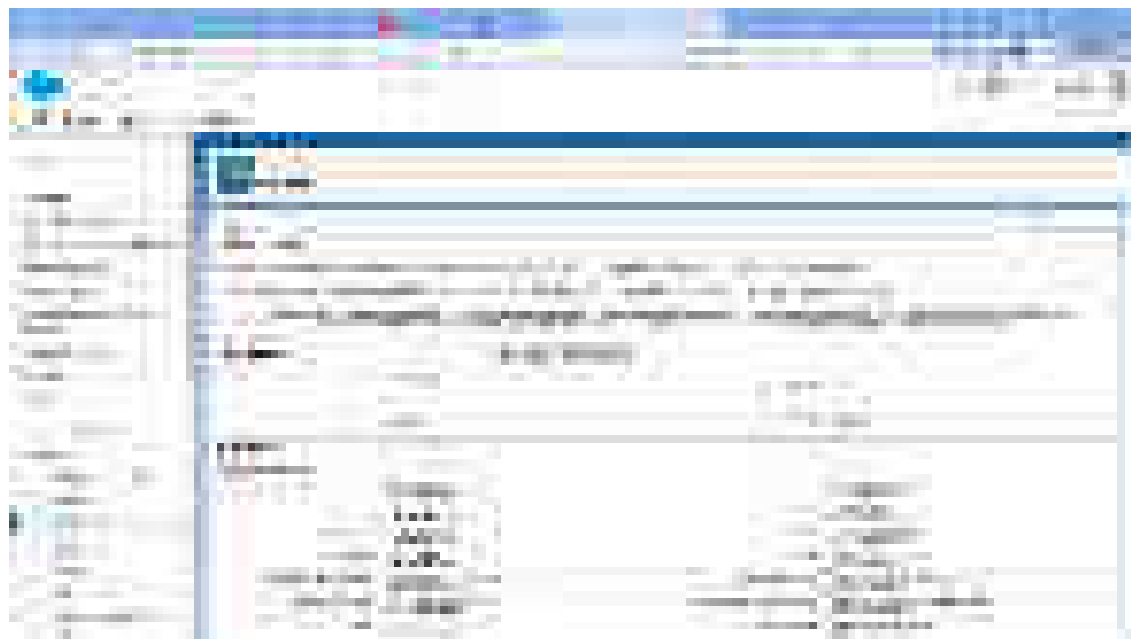




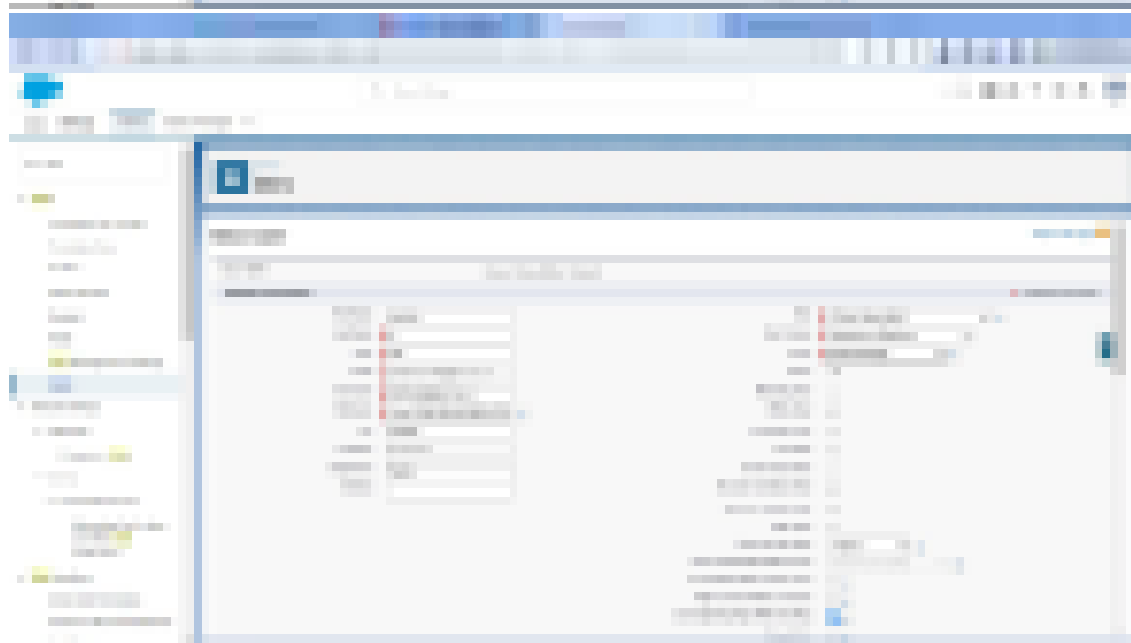
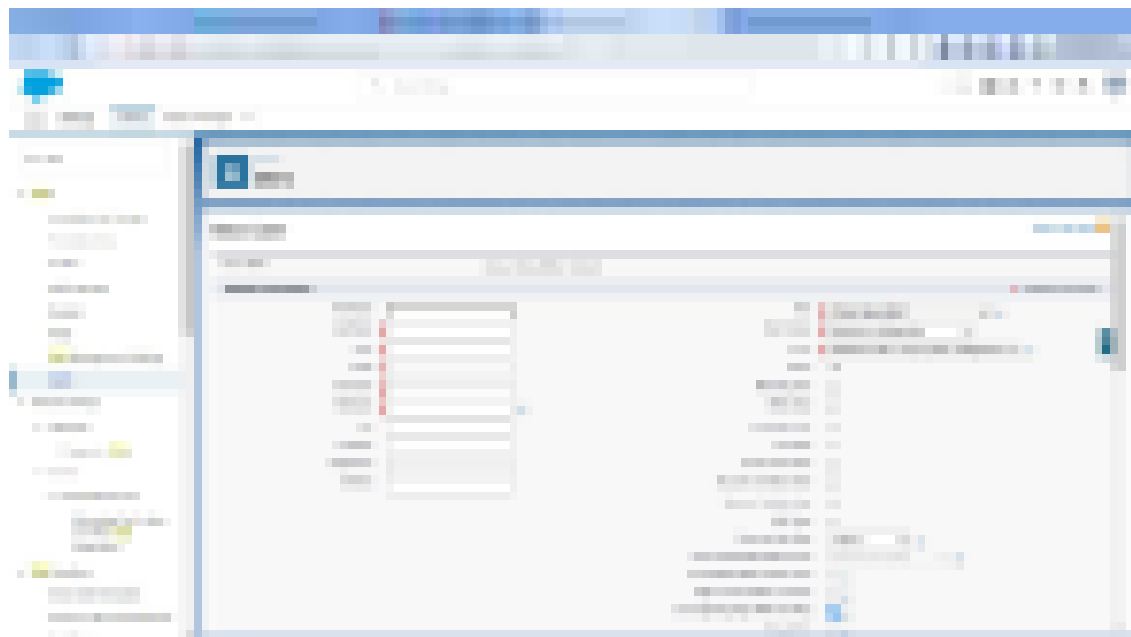


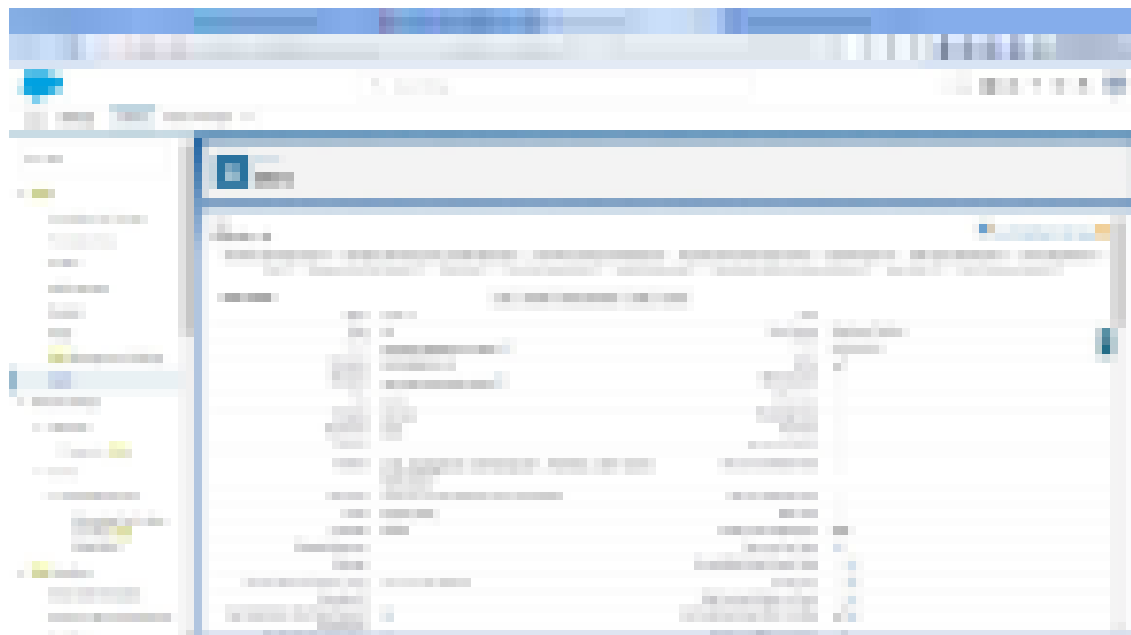


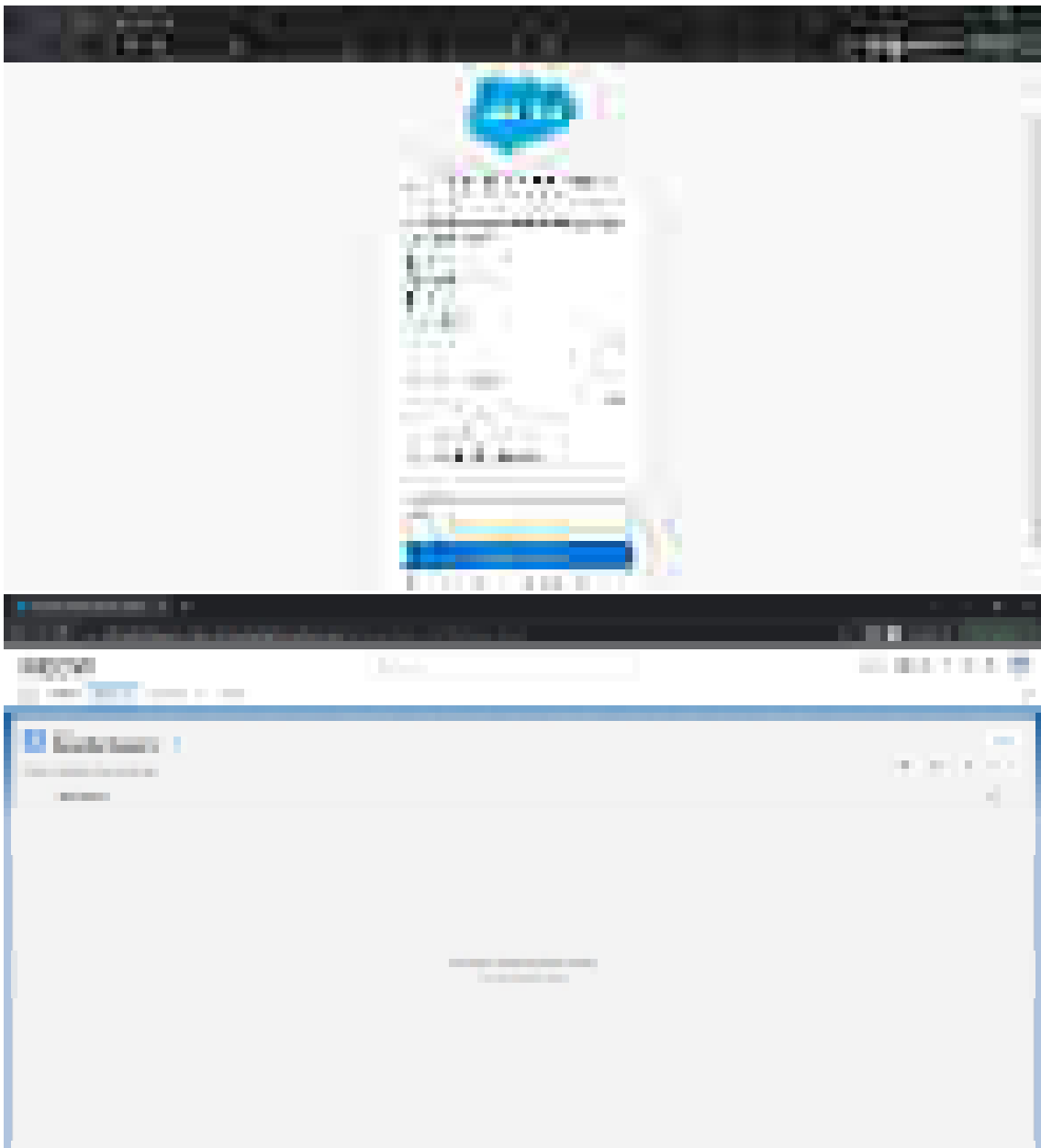


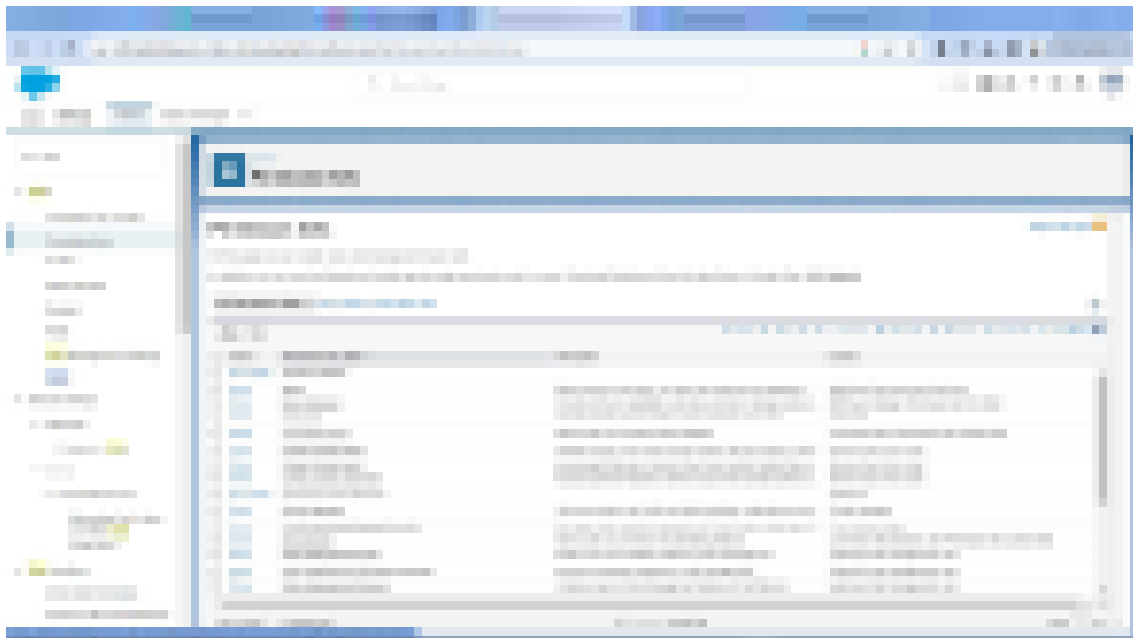
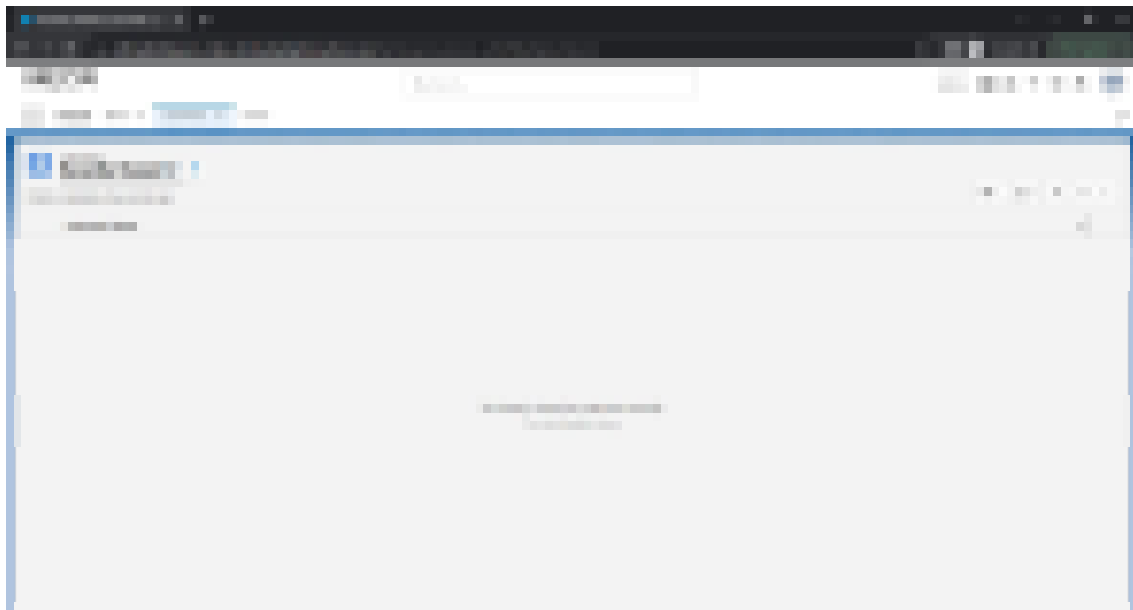












Step 2:

Permission Sets:

- Create two permission sets, one for User A and one for User B.

Object-Level Security:

- In each profile and permission set, set the object-level security for the Account object to "Read" to ensure that both I-Jser A and I-Jser B can view Account records.

Record-Level Security:

- Implement record-level security using Criteria-Based Sharing Rules.
- Create a sharing rule that shares Account records owned by User A with User A and records owned by IJser B with user B.
- For the sharing rule criteria, specify that records owned by User A are shared with user A, and records owned by User B are shared with User B.

Ownership:

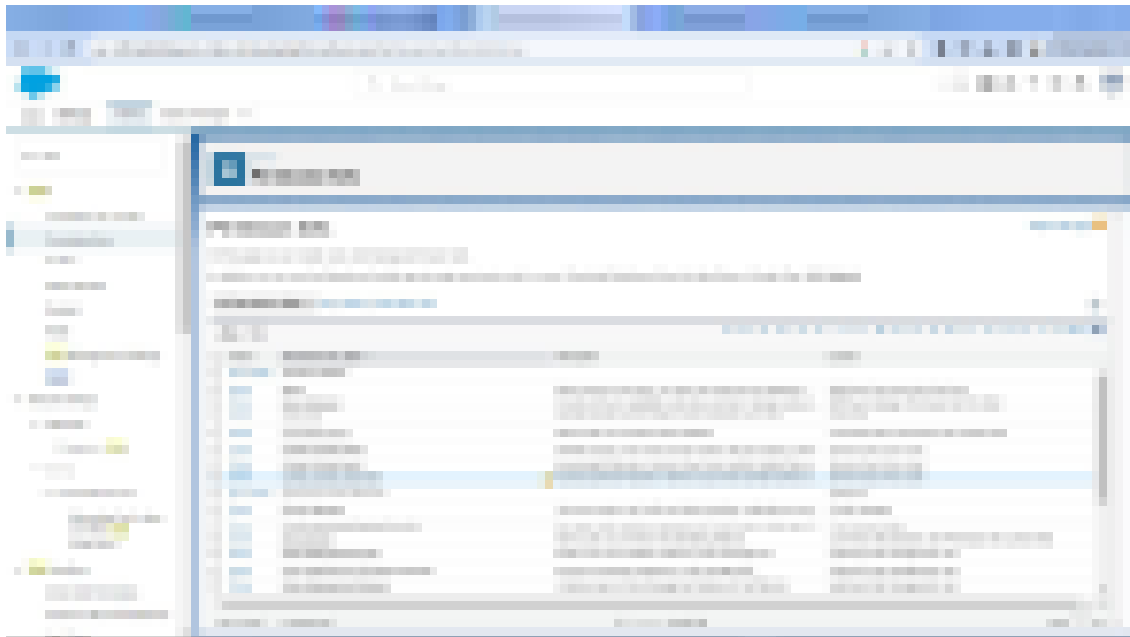
- Ensure that the Account records are owned by the respective users, with User A owning their records and User B owning their records.

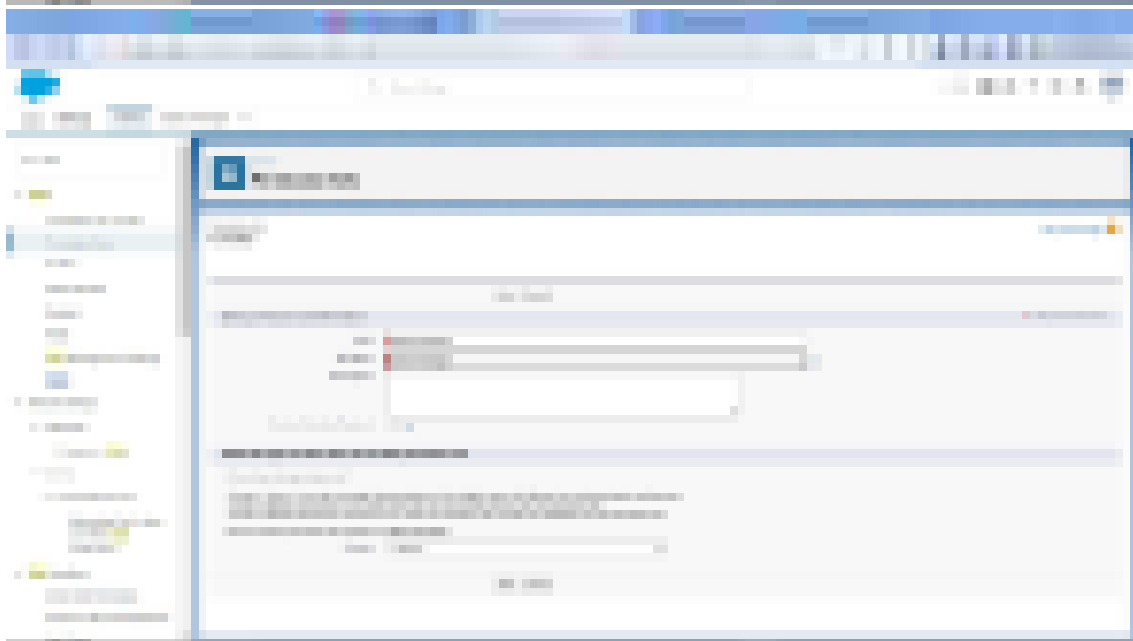
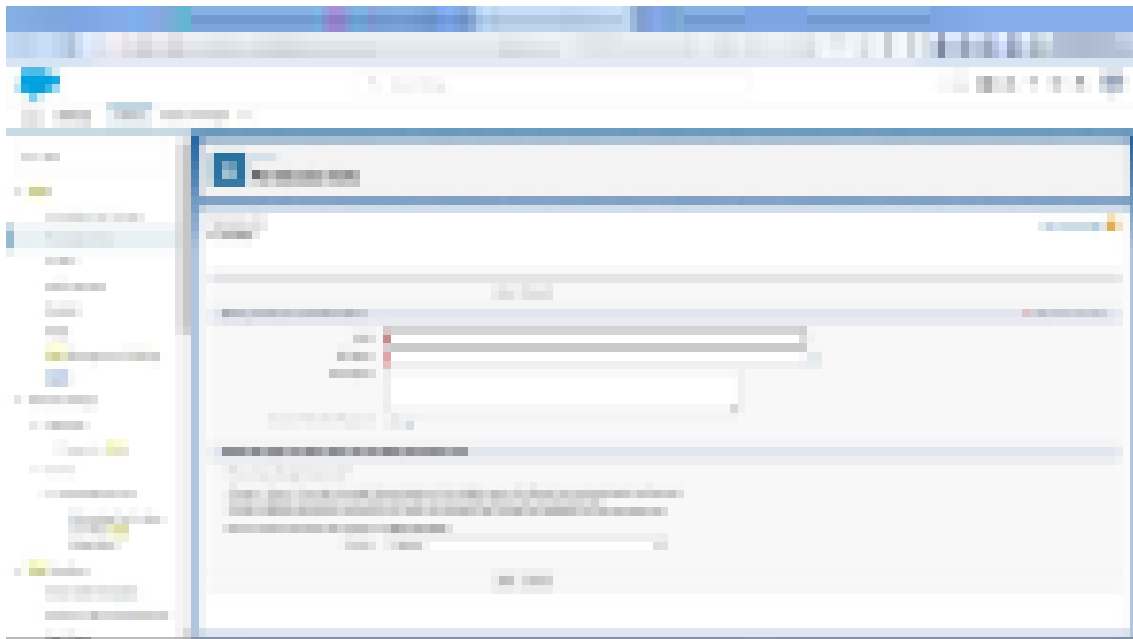
Organization-Wide Defaults:

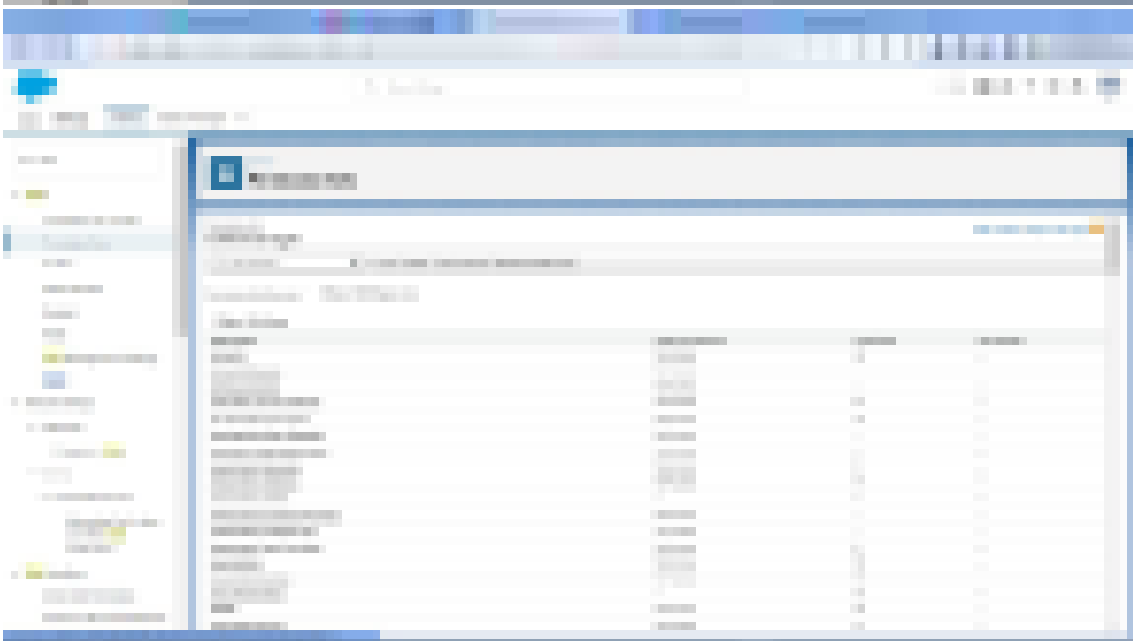
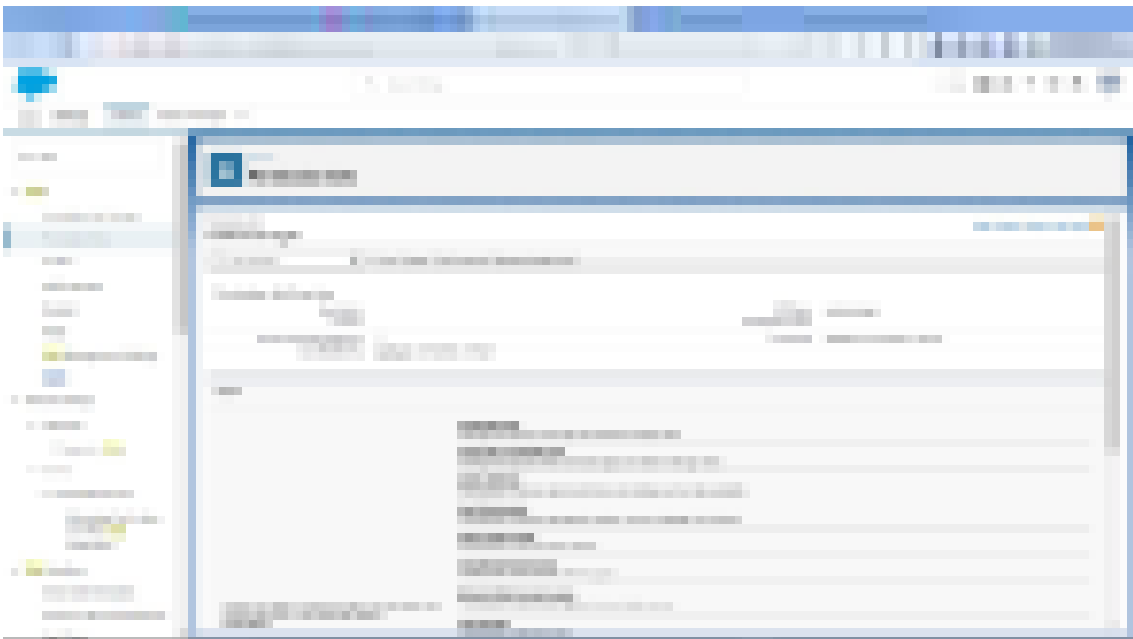
- Set the organization-wide defaults for the Account object to "Private" to ensure that records are private by default.

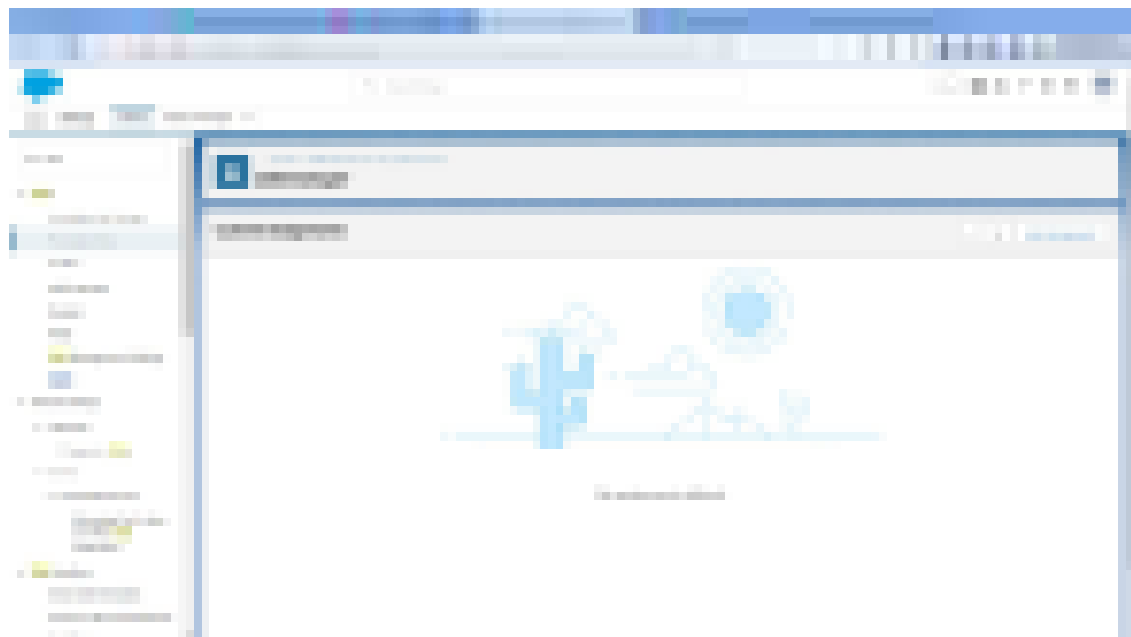
Testing:

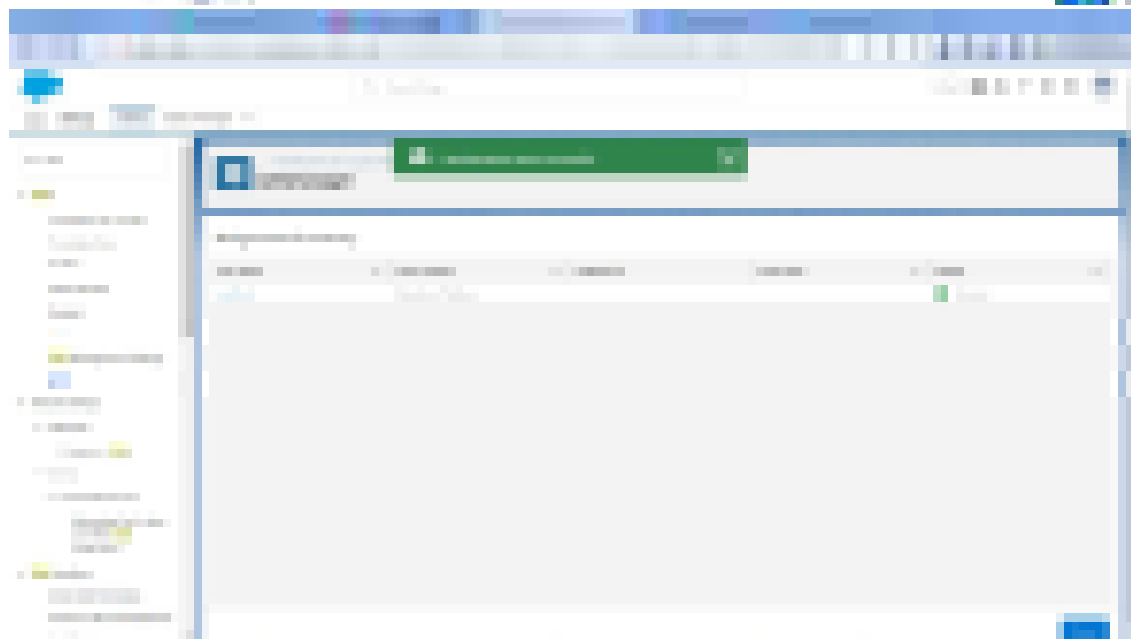
- Test the setup by logging in as User A and User B separately to verify that they cannot access each other's records.

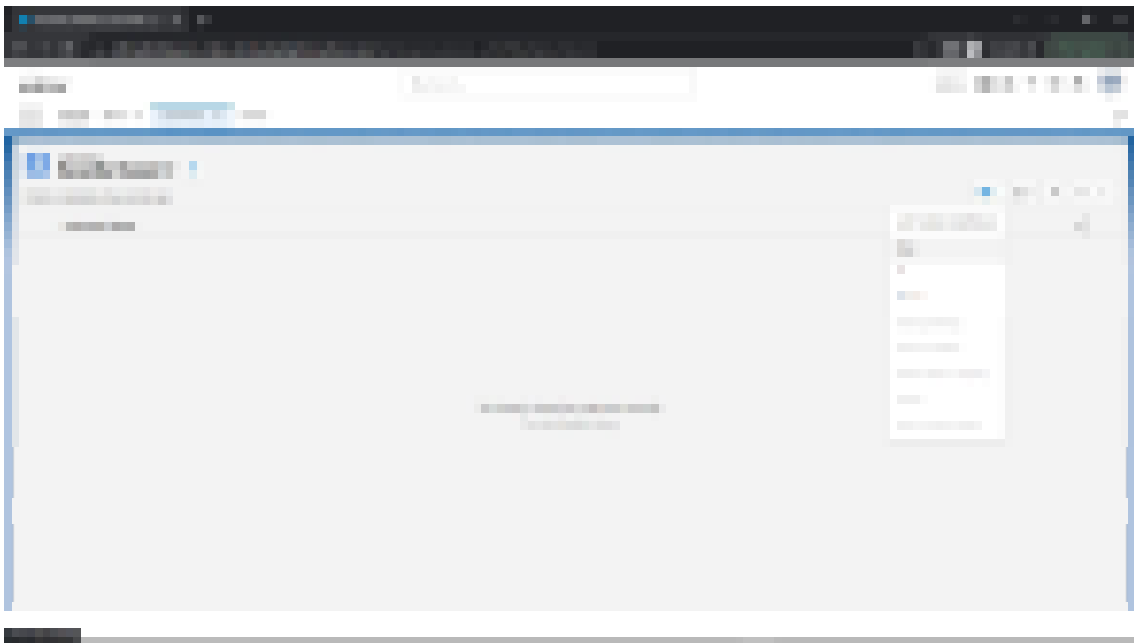












3. . Suppose there are 2 Users and they are having Create, Read, Edit access on Account Object with the same profile but we want to open up the access for one user to delete how will you implement the Security setting.

Solution:

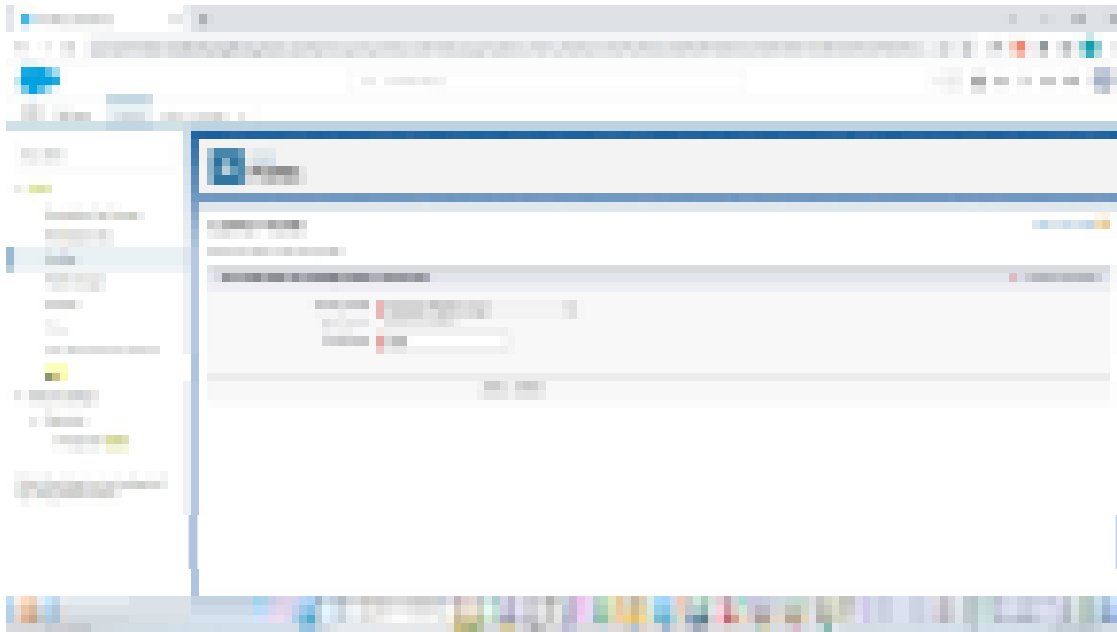
Step 1: we need create a profile for the two user which has the access to Create, Read, Edit for follow as per.

Setup-quick search[profile]



Step 2:

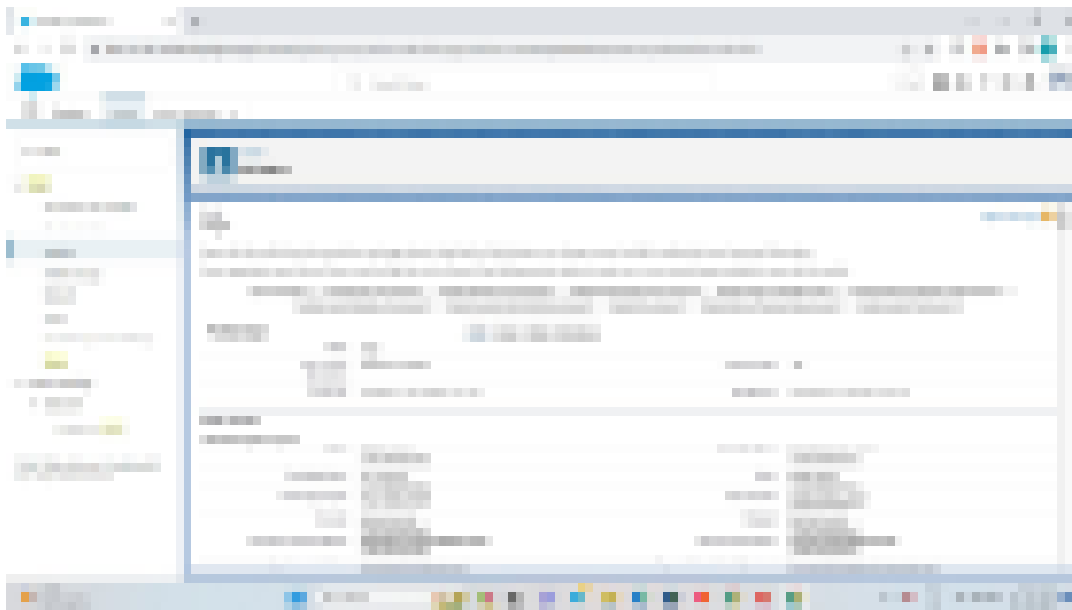
Click on the new to create a new profile along with the label and Api

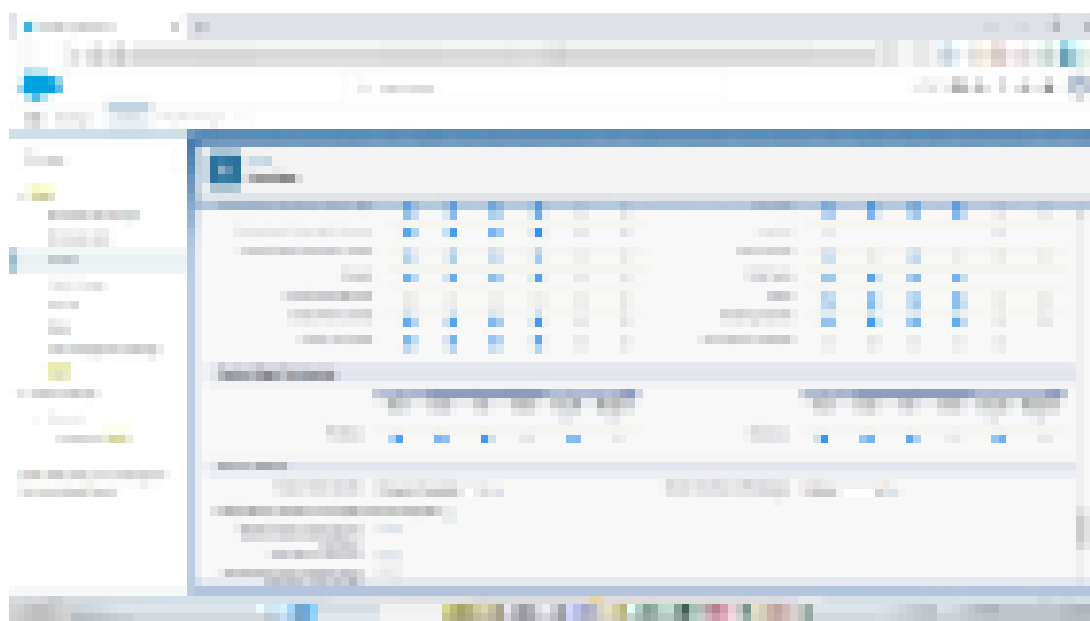
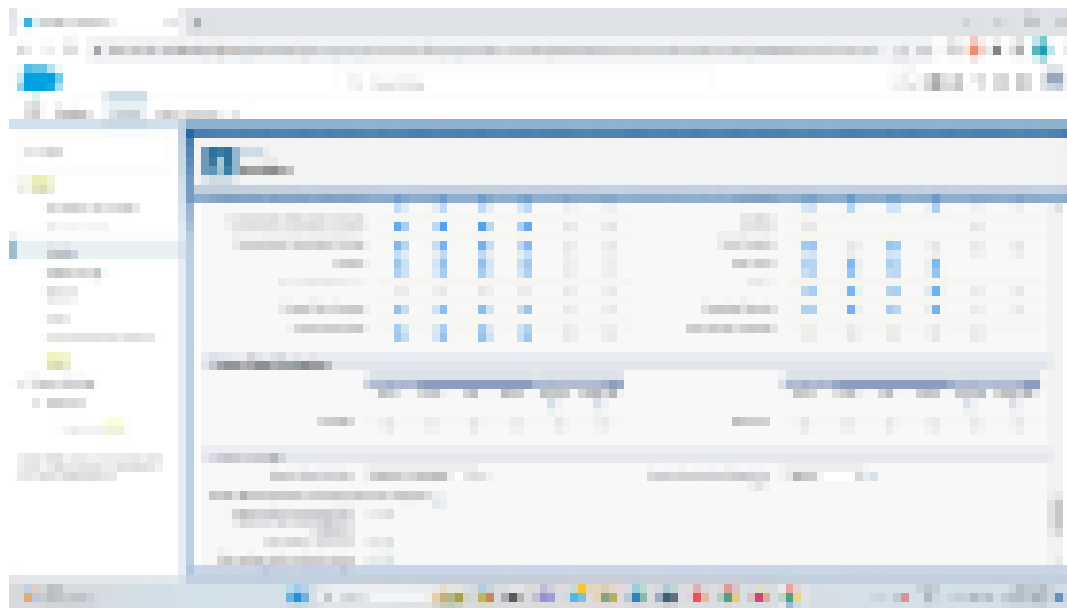


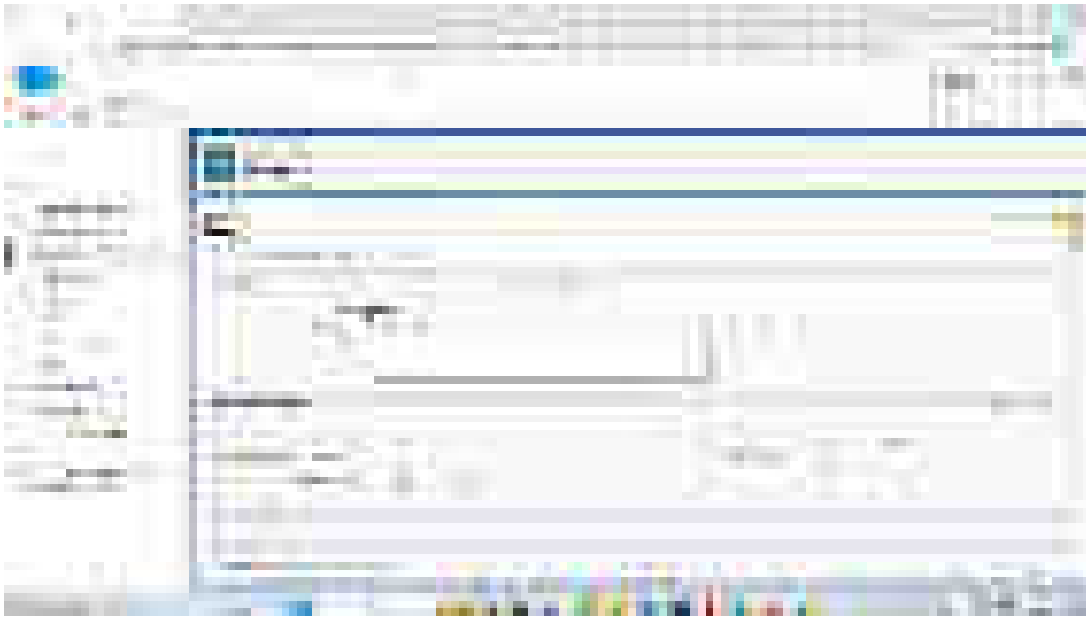
Here I had made it my profile name as Jaga and the existing profile as Standard Platform User.

Step 3:

Now click on the edit and scroll down to custom object settings and enable the read,create,edit and view options. After that click on save.

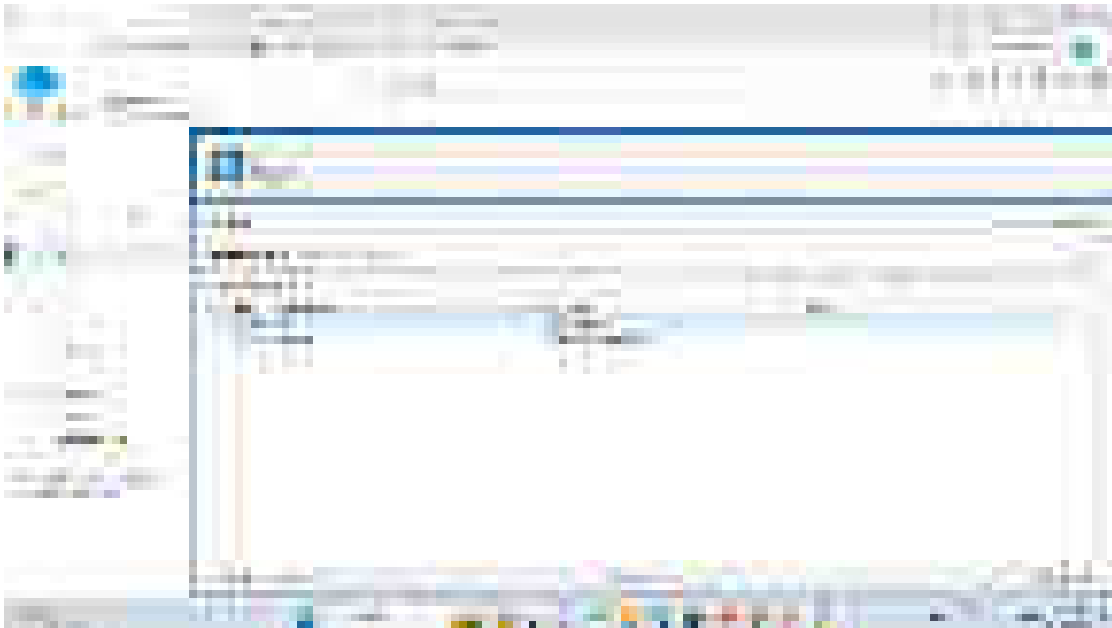






Step 4

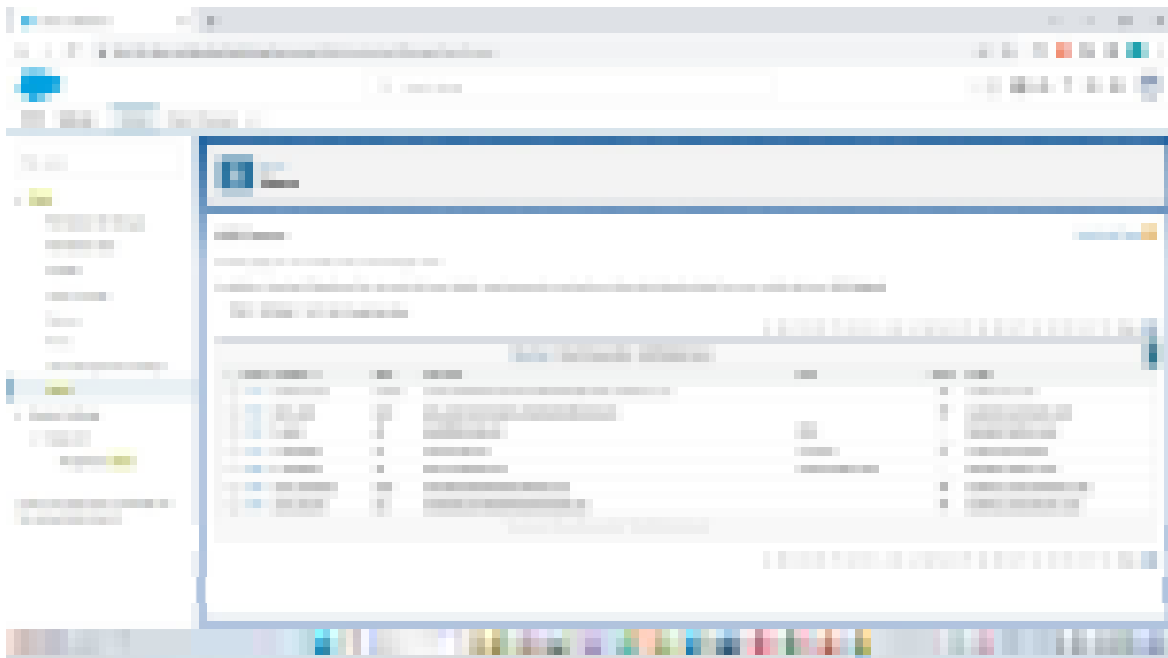
Now you can preview your created profile on the profile option here my profile name jaga has been created with the access of read,create,edit along with view on it



Step 5:

Now create two users by enter into the Setup-quick search[user] and then click on new user after clicking that you need to create two user along

with the profile as Jaga which we have created on the step 2. once the one user has been created click on the save&new so that you can create the second user and there the user name can be created with alternate name but with the same user profile and once the two user are create click on save.

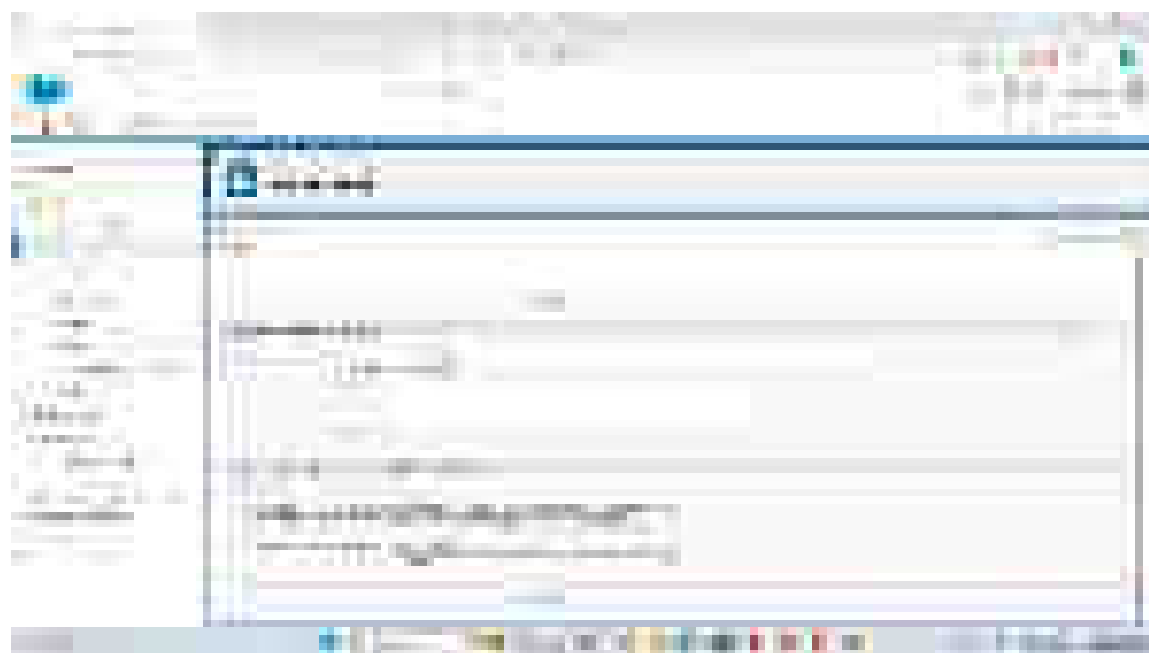
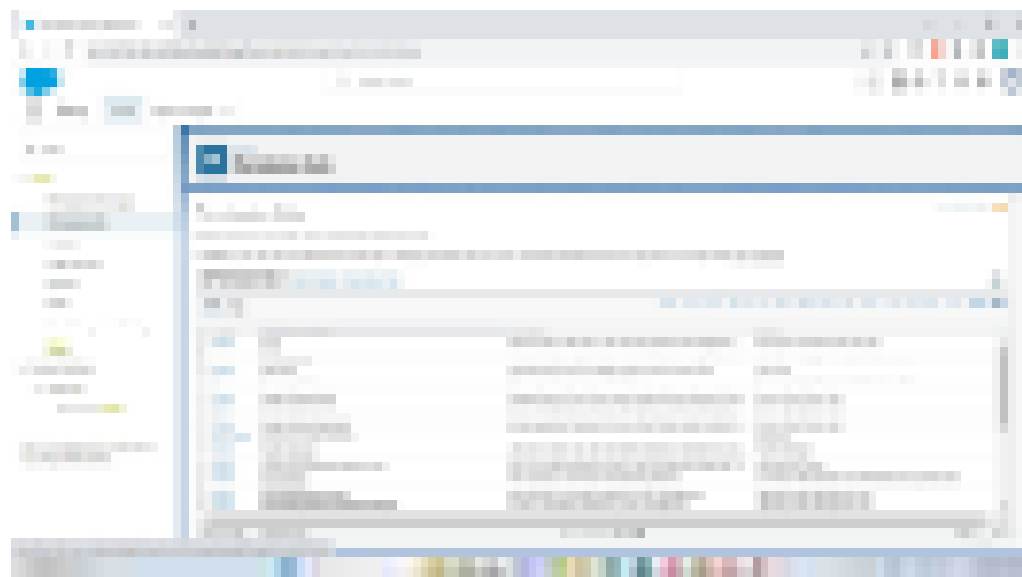


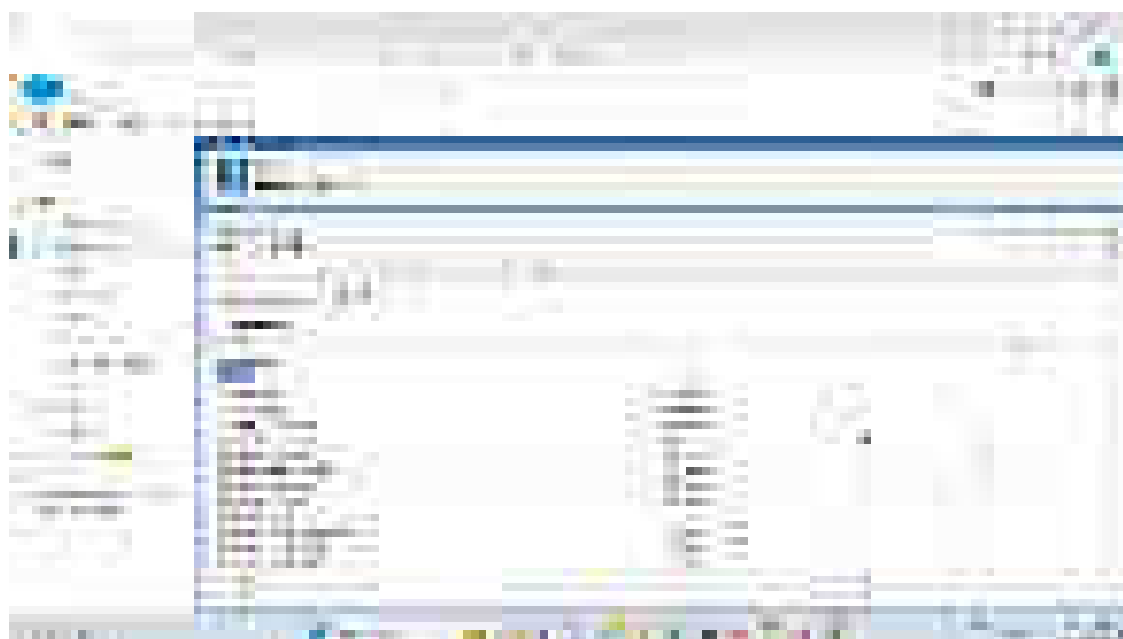
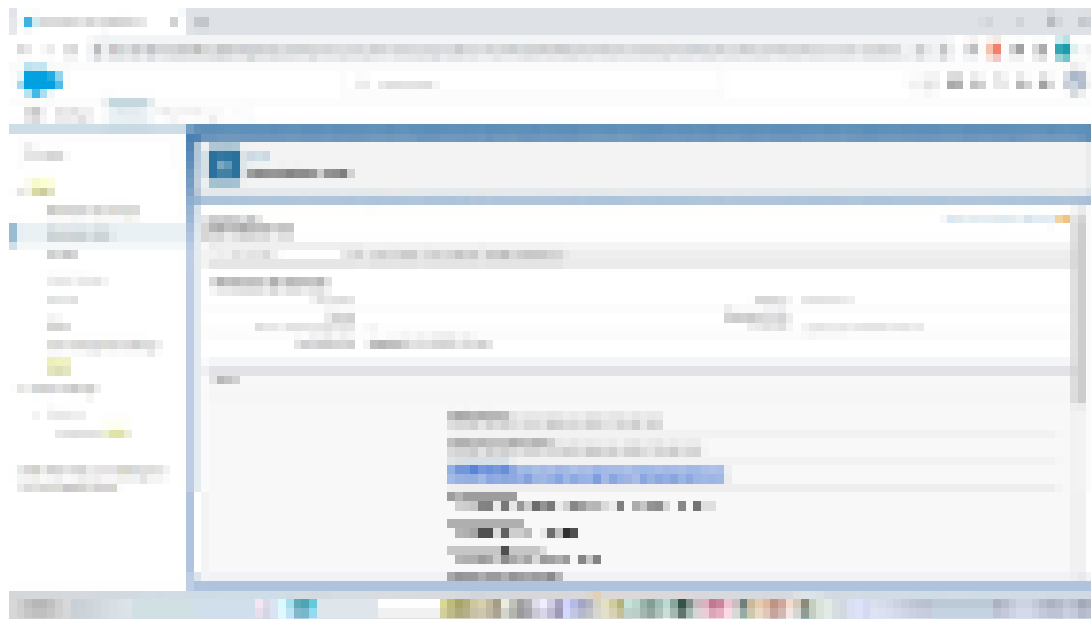
Now you can preview your two user that you have created in my side I had create the two users a Jagadesh11 and Jagadesh22 as a director channel sales with the marketing team.

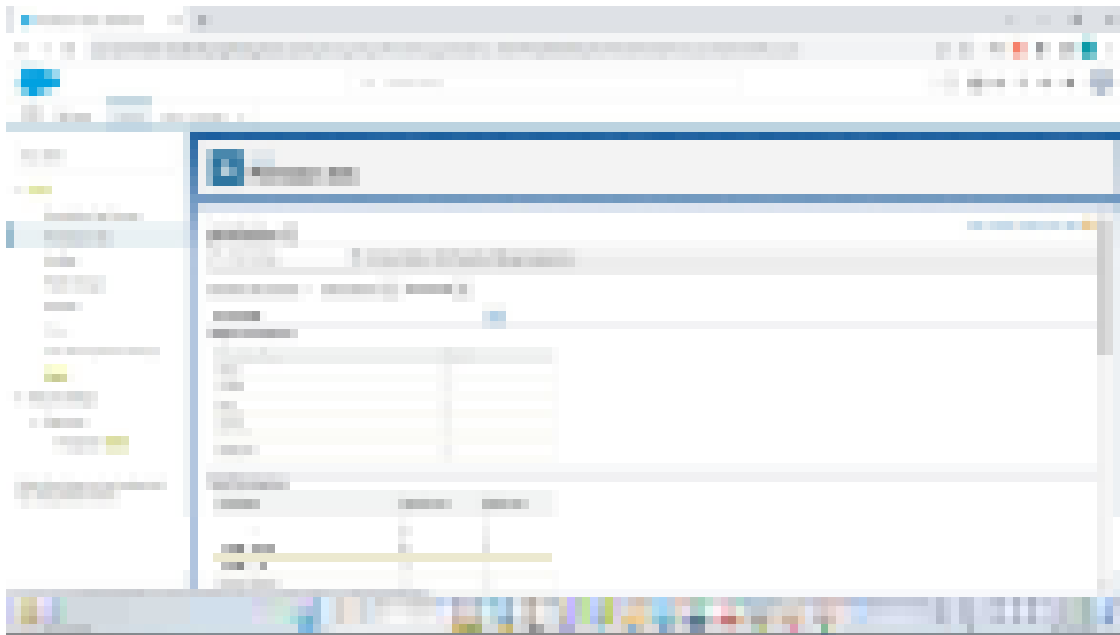
Step 6:

Now the two user as been created with the profile so that two user can perform the Create, Read, Edit and view on both the user. So as per the given task we need to allocate a specific access as delete on one user for that we need create a permission set for one user so it can created as

setup-quick search[permission set]-new-fill label name [auto select the API name]-click on save-object settings-accounts.

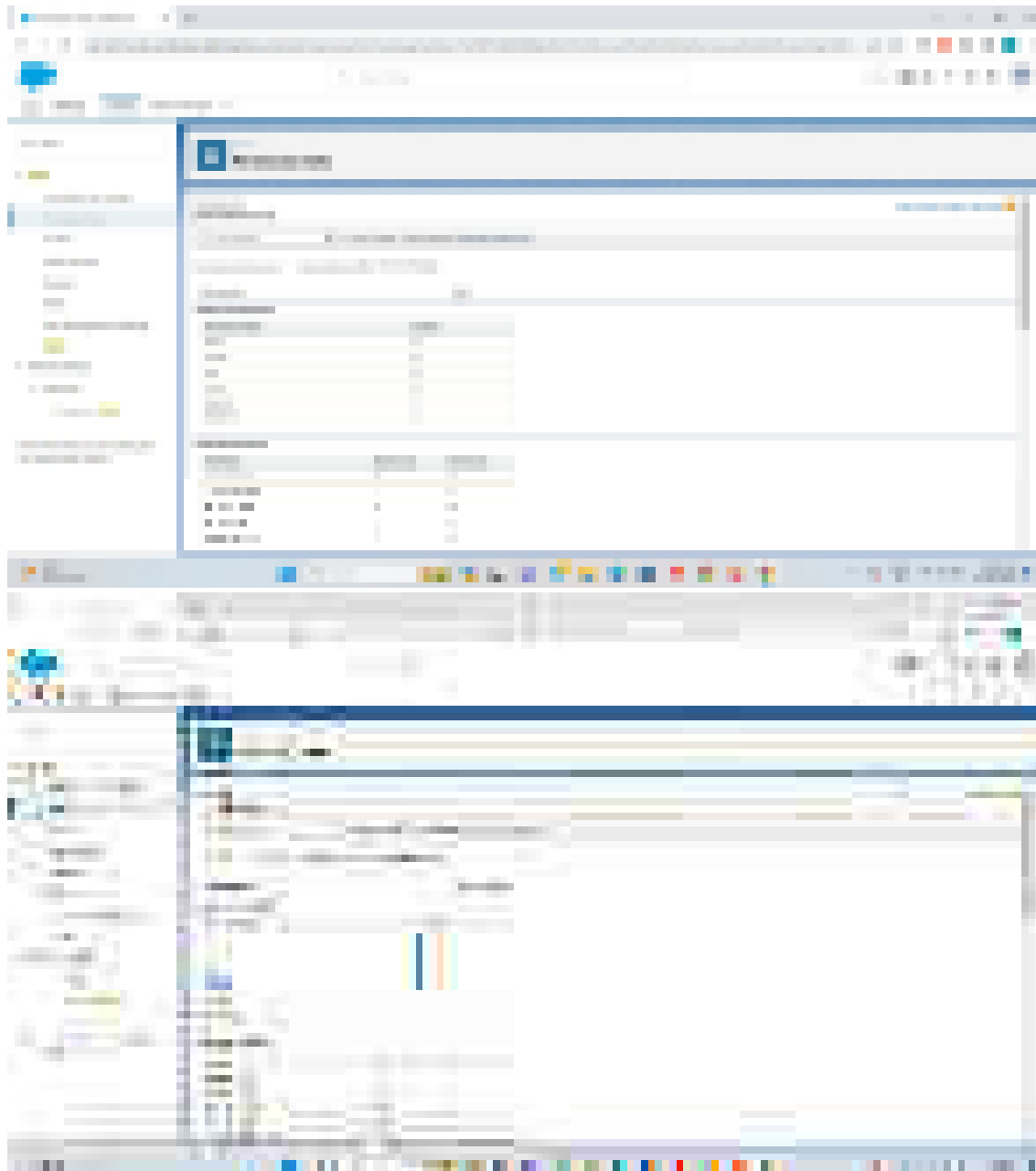






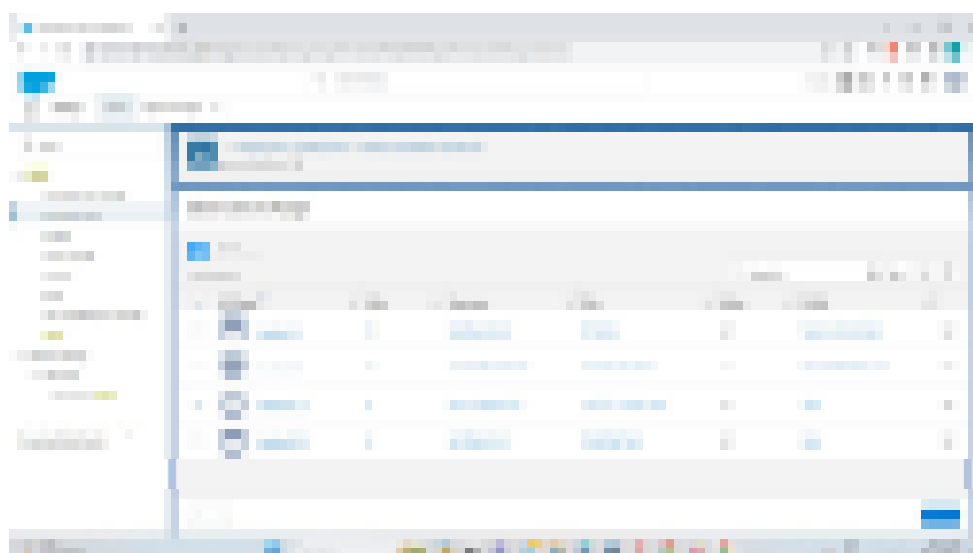
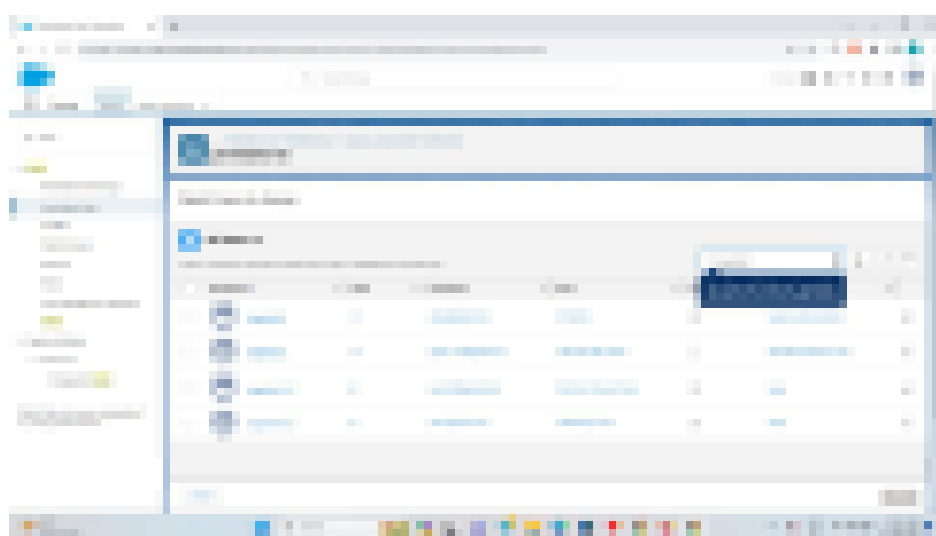
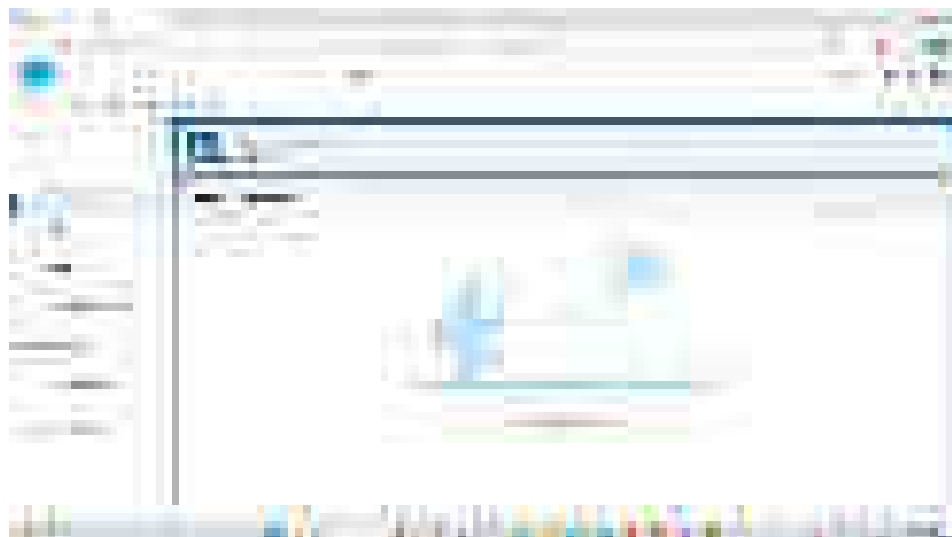
Step 7:

Now to give the specific delete access to the user click on edit on the Account and then enable the read,create,edit and the delete on it so that the permission set will have a specific special access on it. once it has been done click on save and then click on manage assignment.

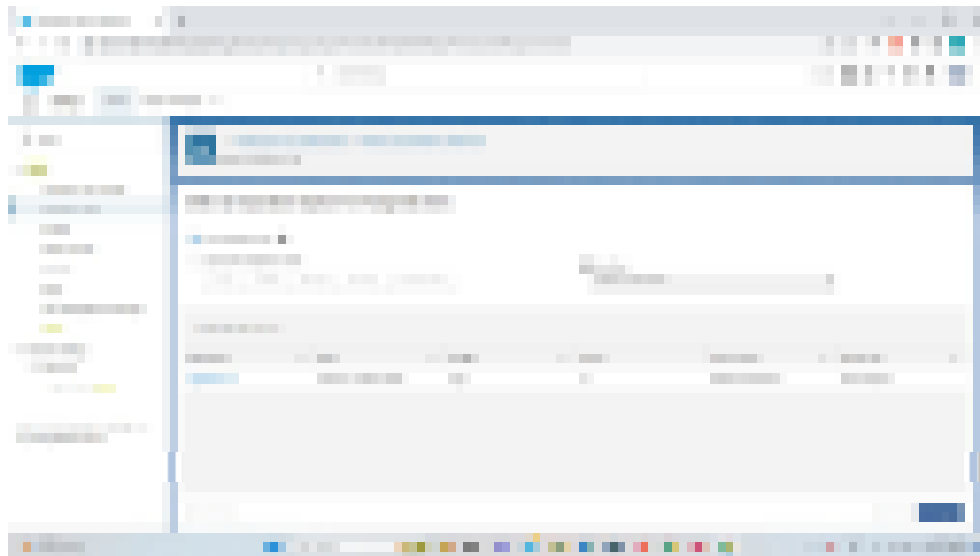


Step 8

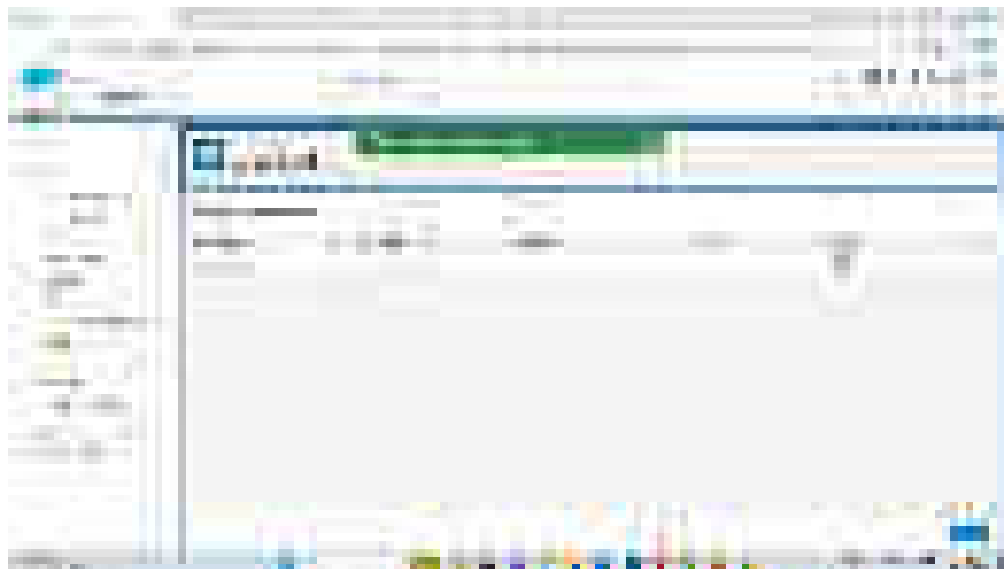
Now click on add assignment there you will find your two created users click on any one user to give a special access as delete on it and then click on assign so that the specific selected user can have a special access as delete on it.



Click on next.



Now click on Assign.



Now the specific access for the Jagadesh11 user has been assigned successfully.

4. Create a screen flow for a basic survey to fill in the details for any form.

Solution:

Step 1: Create a Custom Object

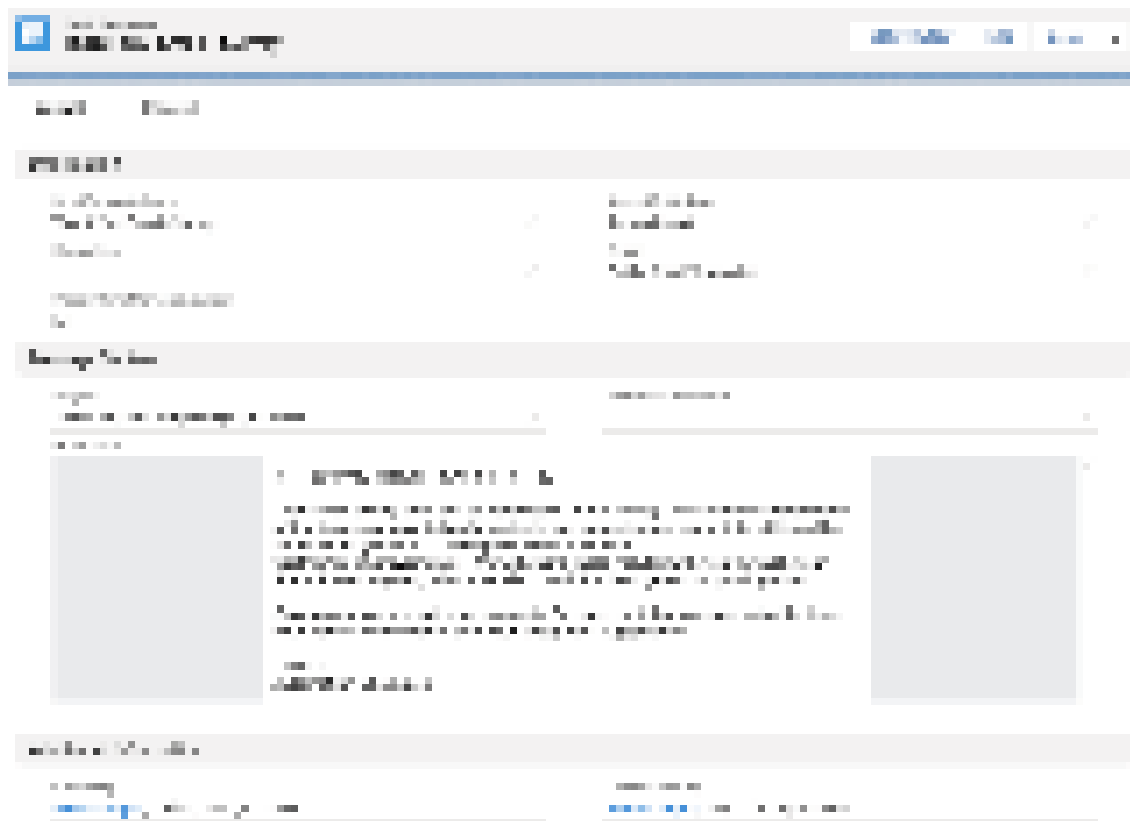
The next step is to create a custom object **Survey Result** and a few custom fields to store survey responses.

1. Click **Setup**.
2. In the Object Manager, click **Create | Custom Object**.
3. Now create a custom object **Survey Result** and fields as shown in the screenshot below:
4. Click **Save**.

Product	Product ID	Product Name	Product Type	Product Status	Product Category	Product Subcategory	Product Description	Product Price	Product Weight	Product Dimensions	Product Image
Product 1	1001	Product 1 Name	Product 1 Type	Product 1 Status	Product 1 Category	Product 1 Subcategory	Product 1 Description	Product 1 Price	Product 1 Weight	Product 1 Dimensions	Product 1 Image
Product 2	1002	Product 2 Name	Product 2 Type	Product 2 Status	Product 2 Category	Product 2 Subcategory	Product 2 Description	Product 2 Price	Product 2 Weight	Product 2 Dimensions	Product 2 Image
Product 3	1003	Product 3 Name	Product 3 Type	Product 3 Status	Product 3 Category	Product 3 Subcategory	Product 3 Description	Product 3 Price	Product 3 Weight	Product 3 Dimensions	Product 3 Image
Product 4	1004	Product 4 Name	Product 4 Type	Product 4 Status	Product 4 Category	Product 4 Subcategory	Product 4 Description	Product 4 Price	Product 4 Weight	Product 4 Dimensions	Product 4 Image
Product 5	1005	Product 5 Name	Product 5 Type	Product 5 Status	Product 5 Category	Product 5 Subcategory	Product 5 Description	Product 5 Price	Product 5 Weight	Product 5 Dimensions	Product 5 Image
Product 6	1006	Product 6 Name	Product 6 Type	Product 6 Status	Product 6 Category	Product 6 Subcategory	Product 6 Description	Product 6 Price	Product 6 Weight	Product 6 Dimensions	Product 6 Image
Product 7	1007	Product 7 Name	Product 7 Type	Product 7 Status	Product 7 Category	Product 7 Subcategory	Product 7 Description	Product 7 Price	Product 7 Weight	Product 7 Dimensions	Product 7 Image
Product 8	1008	Product 8 Name	Product 8 Type	Product 8 Status	Product 8 Category	Product 8 Subcategory	Product 8 Description	Product 8 Price	Product 8 Weight	Product 8 Dimensions	Product 8 Image
Product 9	1009	Product 9 Name	Product 9 Type	Product 9 Status	Product 9 Category	Product 9 Subcategory	Product 9 Description	Product 9 Price	Product 9 Weight	Product 9 Dimensions	Product 9 Image
Product 10	1010	Product 10 Name	Product 10 Type	Product 10 Status	Product 10 Category	Product 10 Subcategory	Product 10 Description	Product 10 Price	Product 10 Weight	Product 10 Dimensions	Product 10 Image

Step 2: Create a Thank You For Survey Lightning Email Template

1. Click **App Launcher**.
2. In the Quick Find box, type **Email Templates**.
3. Clicks on the **New Email template** button.
4. Name the **Lightning Email Template** and make sure to store it in the **Public Email Templates** folder.
5. Create a template like the following screenshot.



Step 3: Create an Email Alert

1. Click **Setup**.
2. In the Quick Find box, type **Email Alerts**.
3. Select **Email Alerts**, click on the **New Email Alert** button.
4. **Name** the **Email Alert** and click the **Tab** button. The **Unique Name** will populate.

2. In the Quick Find box, type **Flows**.
3. Select **Flows** then click on the **New Flow**.
4. Select the **Screen Flow** option and click on **Next** and configure the flow as follows:
 1. **How do you want to start building: Freeform**
5. We will use the **Screen** element to capture a **Survey response** form. Drag and drop a **Screen** element onto the canvas.

Step 4.2: Salesforce Flow — Add a Record Creates Element to Save Survey Response

1. Drag-and-drop the **Create Records** element onto the Flow designer.
2. Enter a name in the **Label (Save Response)** field; the **API Name** will auto-populate.
3. For **How Many Records to Create** – select **One**.
4. For **How to Set the Record Fields** – select **Use separate resources, and literal values**.
5. Select the **Survey_Result__c** object from the dropdown list.
6. **Set Field Values for the Survey Result**
 1. Row 1:
 1. **Field: Comment__c**

2. **Value: {!Comment}**

2. **Click Add Row**

3. **Row 2:**

1. **Field: Email__c**

2. **Value: {!Email.value}**

4. **Click Add Row**

5. **Row 3:**

1. **Field: Name__c**

2. **Value: {!Name.firstName}**
{!Name.lastName}

6. **Click Add Row**

7. **Row 3:**

1. **Field: Rating__c**

2. **Value: {!Rating}**

7. **Click Done.**



Step 4.3: Salesforce Flow — Call an Action — Email Alert to Send Out Thank You Email

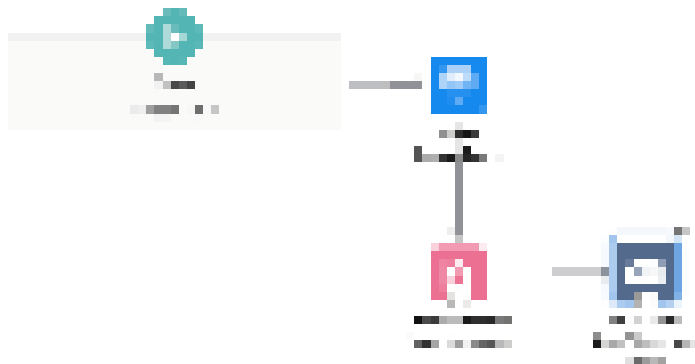
The next step is to call the **Survey – Thank You Email** email alert from flow so that when flow fires it triggers the thank you email to survey participants.

1. Under **Toolbox**, select **Element**.
2. Drag-and-drop **Action** element onto the Flow designer.
3. In the **Action** box, type **Survey – Thank You Email**.

4. Clicks on the **Survey – Thank You Email** email alert.
5. Click **Done**.



In the end, Sergio's **Flow** will look like the following screenshot:



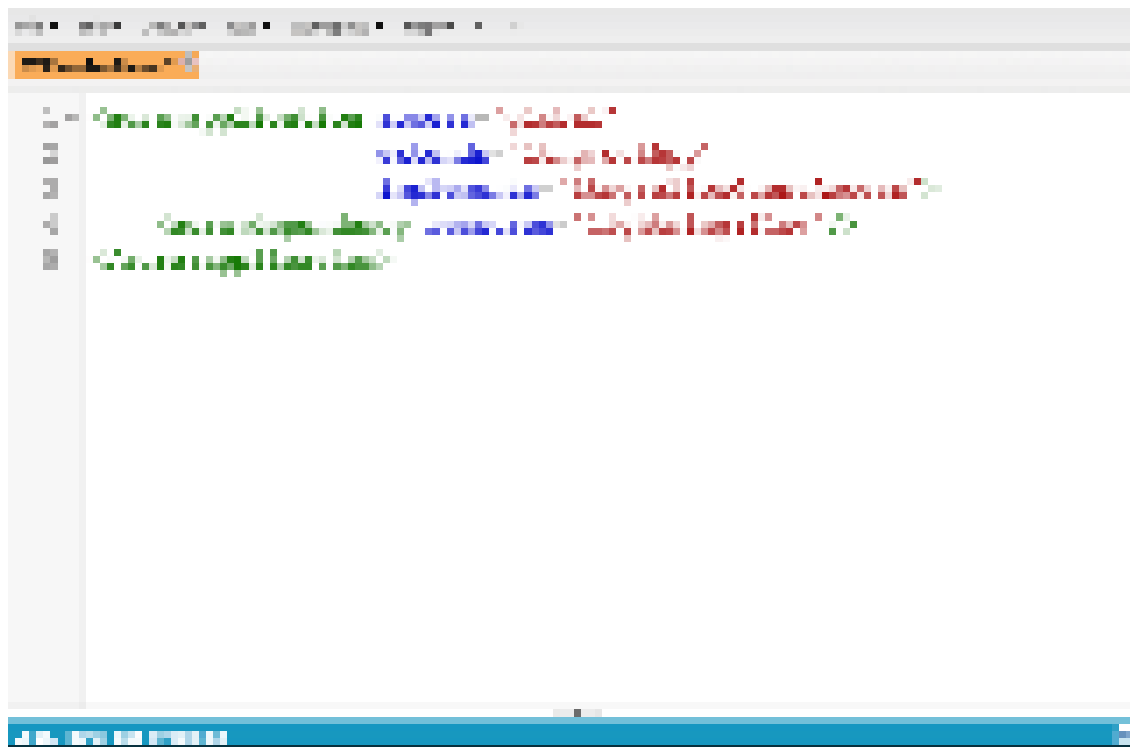
1. Click **Save**.
2. Enter **Flow Label** the **API Name** will auto-populate.
3. Click **Show Advanced**.

4. **How to Run the Flow: User or System Context—Depends on How Flow is Launched**
5. **Type: Screen Flow**
6. **API Version for Running the Flow: 51**
7. **Interview Label: Survey**
{!\$Flow.CurrentDateTime}
8. **Click Save.**

Step 5: Create a Lightning Application to Render Lightning Runtime for Flow in a Visualforce Page

Now we will create a Lightning Application that declares a dependency on the **lightning:flow** component.

1. Click **Setup | Developer Console**
2. Navigate to **File | New | Lightning Application**
3. Enter a **Name (VFPageToLC)** field, make sure to select the **Lightning Out Dependency App** checkbox.
4. Click **Submit**.
5. Copy code from **GitHub** and paste it into your Lightning Application.
6. **Save** your code.



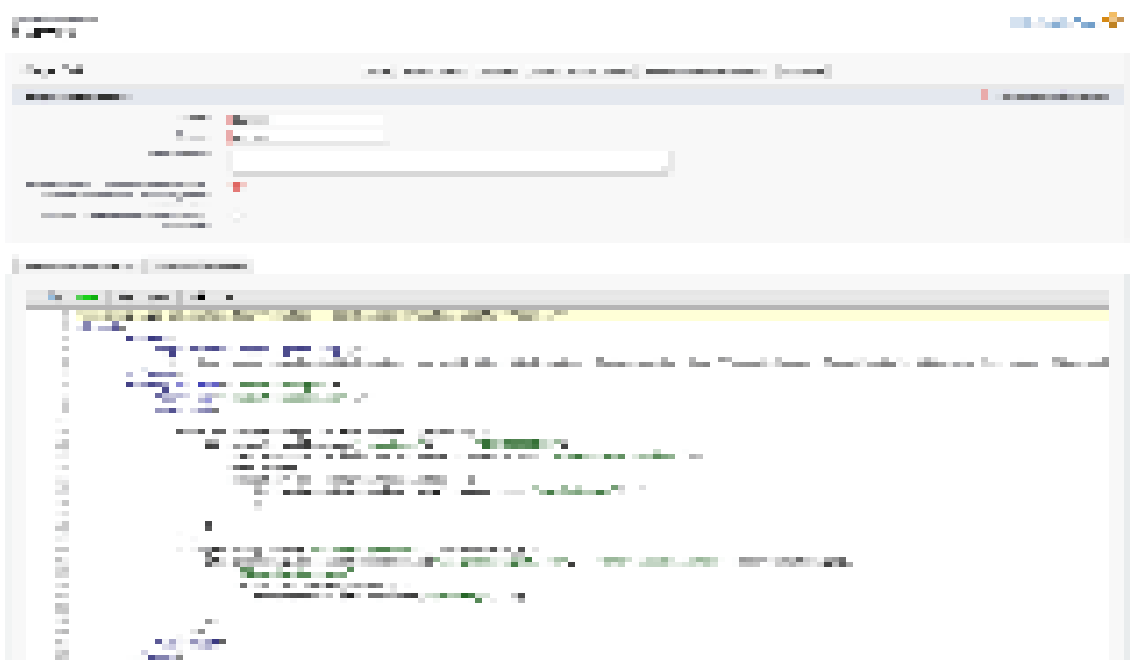
Step 6: Create a Visualforce Page and Embed Your Flow Into It

Now we will create a Lightning Application that declares a dependency on the **lightning:flow** component.

Add the Lightning Components for Visualforce JavaScript library to your Visualforce page using the **<apex:includeLightning/>** component. In the Visualforce page, reference the dependency app. Then write a JavaScript function that creates the

component on the page
using **\$Lightning.createComponent()**

1. Click **Setup**.
2. In the Quick Find box, type **Visualforce Pages**.
3. Clicks on the **New** button.
4. Copy code from **GitHub** and paste it into your visualforce page
5. Click **Save**.



Step 7: Create a Force.com Site to Open Your Flow for Unauthenticated Access

Now we will create a site to open the flow for unauthenticated access.

1. Click **Setup**.

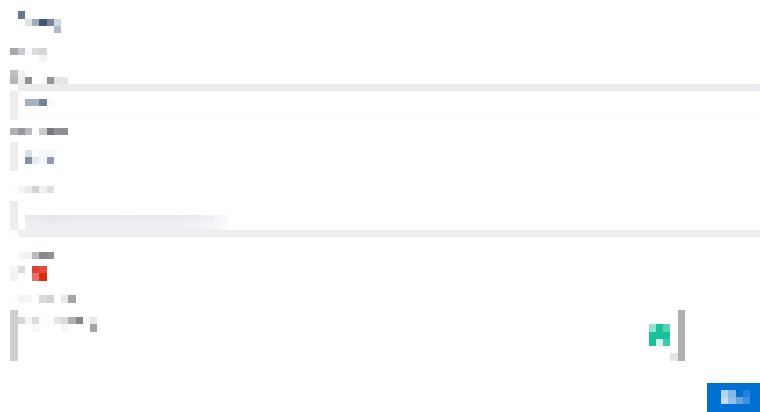
2. In the Quick Find box, type **Sites**.
3. Clicks on the **New** button.
4. Fill the details as per the screenshot below:
5. Click **Save**.



Under site, **Public Access Settings** make sure that guest users have **Create** access on **Survey Result** object and **Edit** on the **fields**.

Proof of Concept

Now onward, if someone opens the site url and fills the form:



After successful submission, he/she will receive an email.

