**Steps to Onboard/Offboard - ObserveID:-**

CreatedDate : 12/04/2024.

**Created by** : **Aman & Rakesh (Megthink Solutions).**

**TABLE OF CONTENT:-**

| **S.no** | **CONTENT** | **Pg.no.** |
| --- | --- | --- |
| **1.** | **CSV TO HR SOURCE INTEGRATION** | [**3**](#v9l7vp4pm9fs) |
| **2.** | **ONBOARDING** | [**14**](#d8iau41vqw3h) |
| **3.** | **OFFBOARDING** | [**25**](#io8bssx0icwz) |

**3 OBSERVEID CSV TO HR SOURCE INTEGRATION:-**

Prepare a csv file with the following properties given below.

1. - Id (it can be anything, e.g. number starting from 1)

2.  - First Name

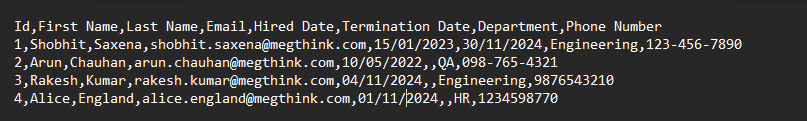
3.  - Last Name

4.  - Email

5.  - Hired Date

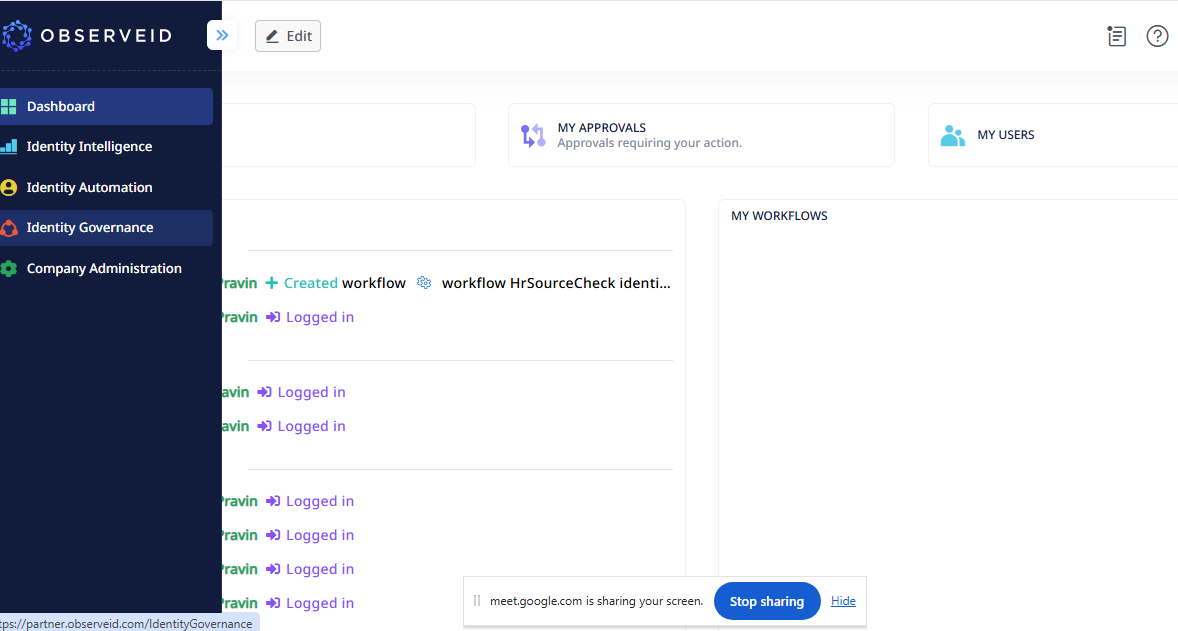
6.  - Termination Date (it may be empty for active employees)

7.  - A couple of columns of your choice - department name, phone number, it is up to you, pick or suggest your own.

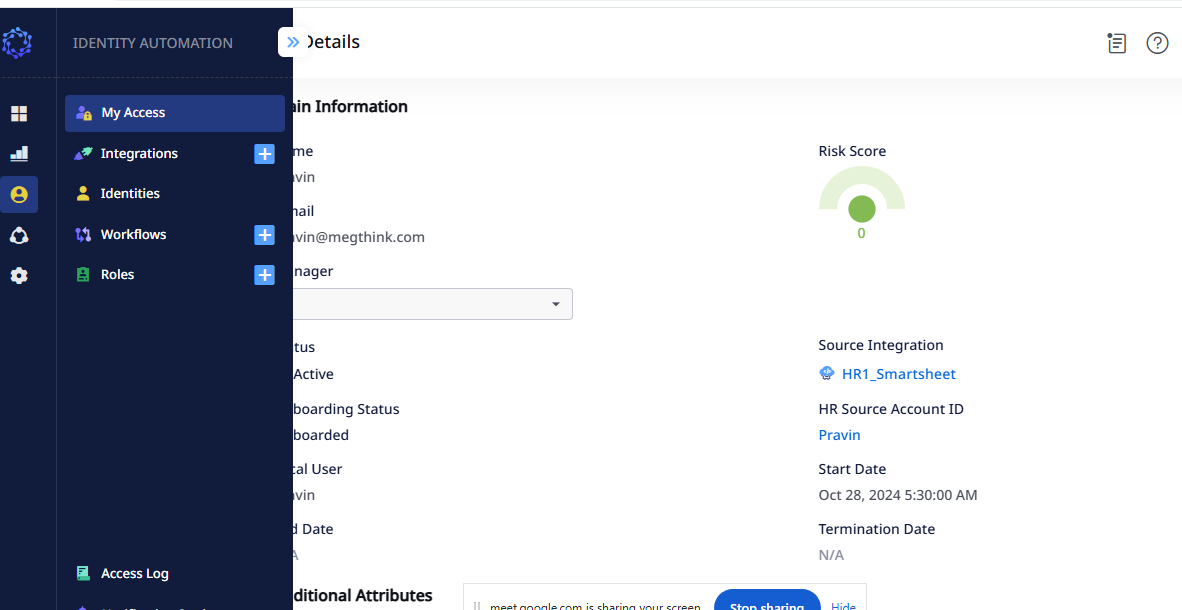
**Example:-**

**HERE ARE THE STEPS FOR CSV INTEGRATION.**

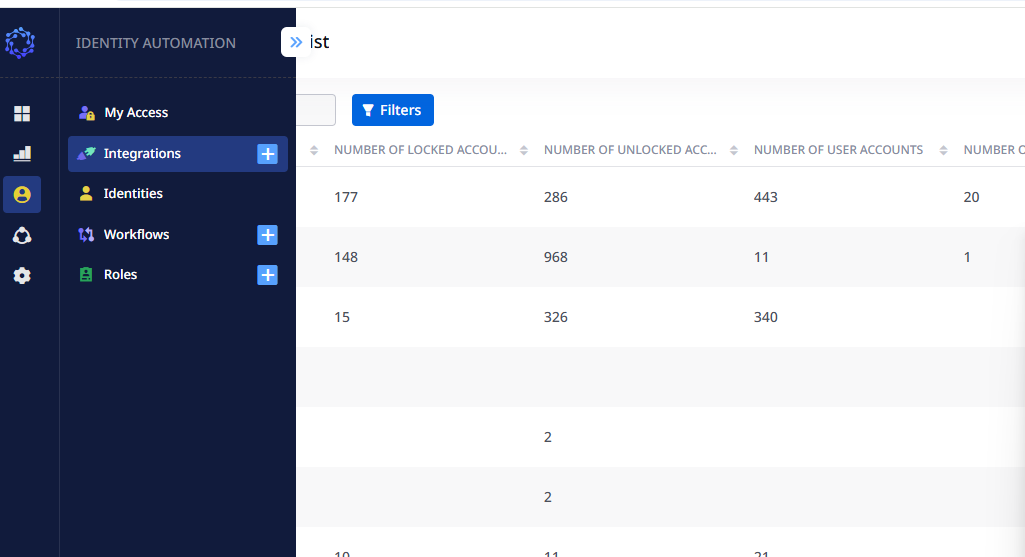
1. Login to partner.observeid.com
2. On the HomePage of ObserveId Dashboard you will see an Identity Automation Tab.



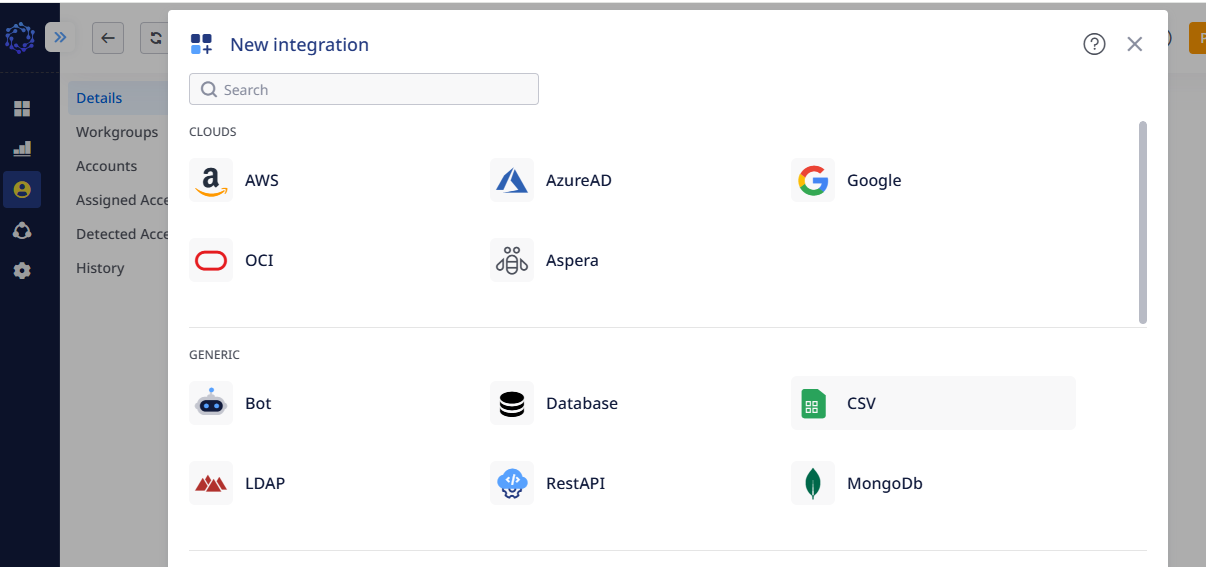
1. Click on the Identity Automation Tab now you will be redirected to Identity Automation Window .



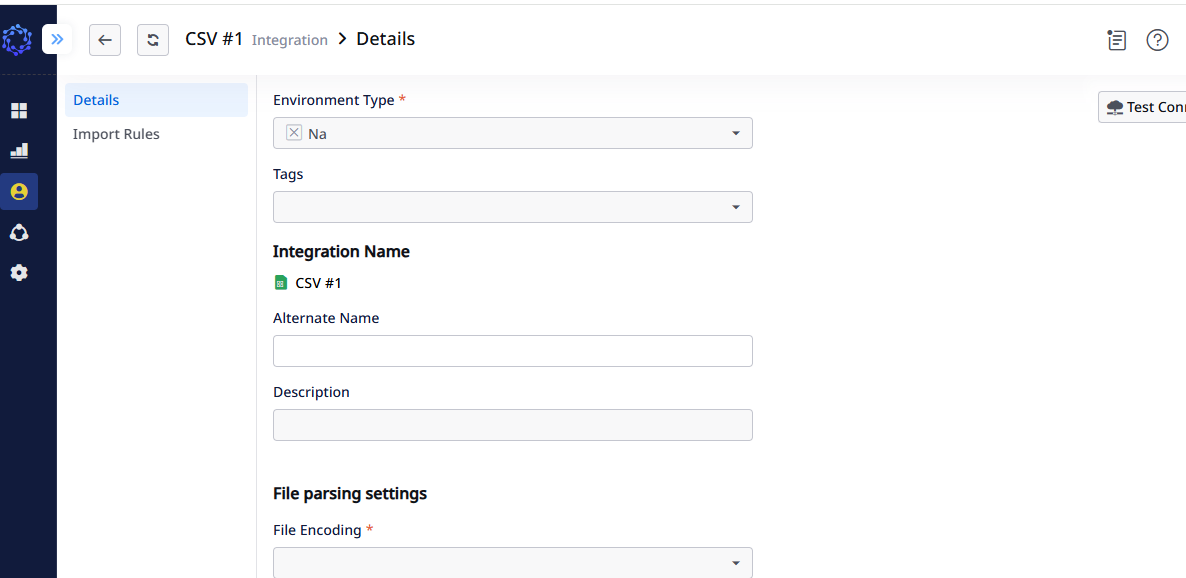
1. Click on the Integration Tab on the leftSide .



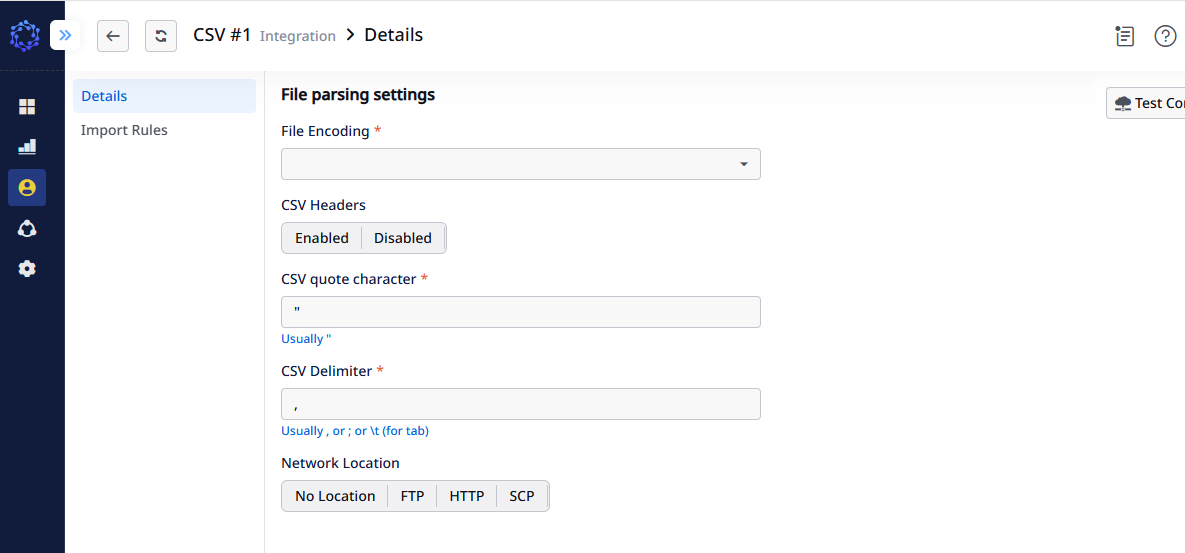
3.Clickon the Integration (+)Icon. The NewIntegration DialogBox will open .Then select CSV.



4.Enter Alternate Name and Description .



5.Select File Encoding from DropDownList .



6. Select CSV Headers to Enabled.

7.Select Network Location .If Uploading from local machineThen Select No location.

8. GoTo importRule Add .

**GetMetaDataRule**:-

**Example MetadataRule:-**

integrationMetadata

.WithResourceType("Root", true, false)

.WithAccountProperty("Id", AdditionalPropertyTypeForRules.String, false, false, false, true)

.WithAccountProperty("First Name", AdditionalPropertyTypeForRules.String, false, false, false, true)

.WithAccountProperty("Last Name", AdditionalPropertyTypeForRules.String, false, false, false, true)

.WithAccountProperty("Email", AdditionalPropertyTypeForRules.String, false, false, false, true)

.WithAccountProperty("Hired Date", AdditionalPropertyTypeForRules.DateTime, false, false, false, true)

.WithAccountProperty("Termination Date", AdditionalPropertyTypeForRules.DateTime, false, false, false, true)

.WithAccountProperty("Department", AdditionalPropertyTypeForRules.String, false, false, false, true)

.WithAccountProperty("Phone Number", AdditionalPropertyTypeForRules.String, false, false, false, true)

.WithOperationPermissions(true, true, false, false, false, false, false, false, false, false, false, false, false, false, false, false, false, true, true, false, false);

return new();

9. Click on Compile.

**10.GetIntegration Data Rule:-**

**Example code:-**

const string rootResourceId = "Root";

const string rootResourceName = "Add your csv file name";

const string rootResourceType = "Root";

var resourceBuilder = integrationData.WithRootResource(rootResourceType, rootResourceId, rootResourceName);

foreach (var account in files.Tables.Cast<DataTable>()

.SelectMany(x => x.Rows.Cast<DataRow>()))

{

var accountBuilder = resourceBuilder.WithAccount(

account["Id"].ToString()!,

null,

account["Email"].ToString()!,

false, // Assuming a boolean value for active status (adjust as needed)

false,

null

);

DateTime hireDate;

DateTime terminationDate;

accountBuilder.WithAccountAdditionalProperty("First Name", new MixedValueForRules(account["First Name"].ToString()));

accountBuilder.WithAccountAdditionalProperty("Last Name", new MixedValueForRules(account["Last Name"].ToString()));

accountBuilder.WithAccountAdditionalProperty("Email", new MixedValueForRules(account["Email"].ToString()));

bool successHire = DateTime.TryParse(account["Hired Date"].ToString(), out hireDate);

if (successHire)

{

accountBuilder.WithAccountAdditionalProperty("Hired Date", new MixedValueForRules(hireDate));

}

else{

accountBuilder.WithAccountAdditionalProperty("Hired Date", new MixedValueForRules(account["Hired Date"].ToString()));

}

bool successTerm = DateTime.TryParse(account["Termination Date"].ToString(), out terminationDate);

if (successTerm)

{

accountBuilder.WithAccountAdditionalProperty("Termination Date", new MixedValueForRules(terminationDate));

}

else

{

accountBuilder.WithAccountAdditionalProperty("Termination Date", new MixedValueForRules(account["Termination Date"].ToString()));

}

accountBuilder.WithAccountAdditionalProperty("Department", new MixedValueForRules(account["Department"].ToString()));

accountBuilder.WithAccountAdditionalProperty("Phone Number", new MixedValueForRules(account["Phone Number"].ToString()));

}

return new();

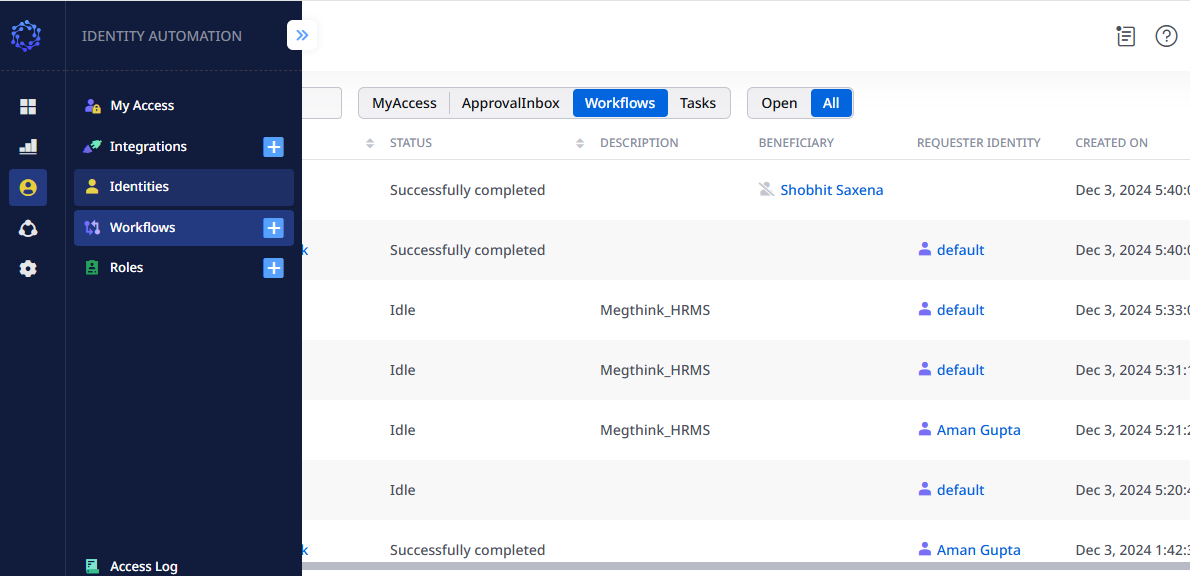
**NOTE:-** Change the Fields or Attributes According to your CSV file in Above Rules.

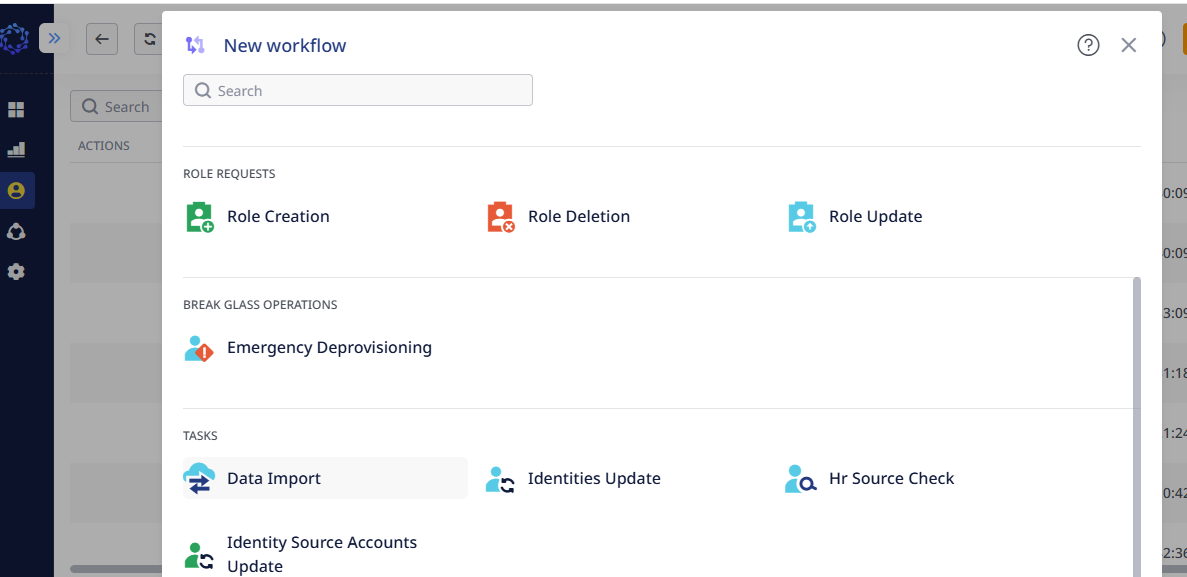
**E.G.:-**  First Name , Last Name….etc.

11. Click on Compile Button.

12. Then Click on Save Button.Then Click on TestConnection Button.

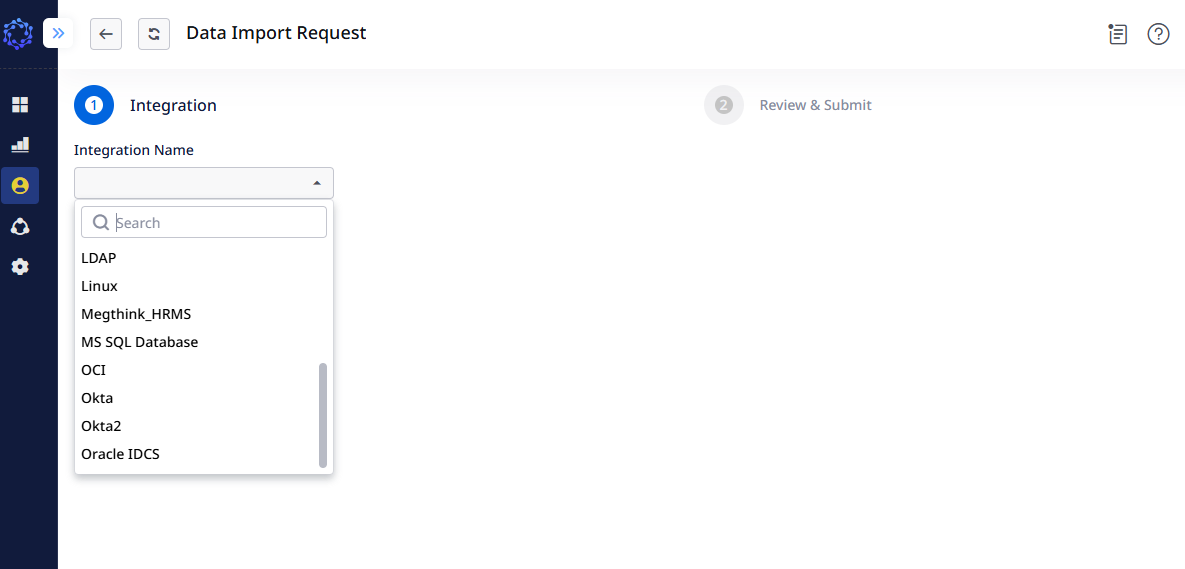
13. Click on WorkFlow (+)Icon on the Leftside of Identity Automation Page.





14. Click on DataImport .

15. Select the Integration Name From The DropDown List of CSV file.



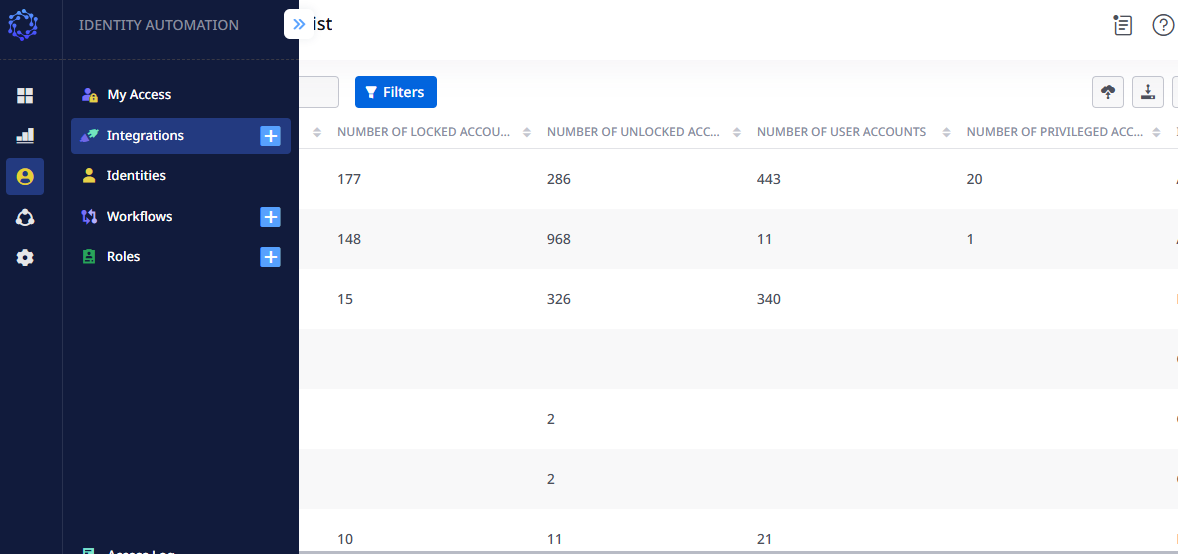
16.Select the Update type to full and Click on the next Button**.**

17.Then Click on The Create Task .

18. Click on ImportCsv Button and Upload the CSV file . And Click on the Submit Button. Click on Save.

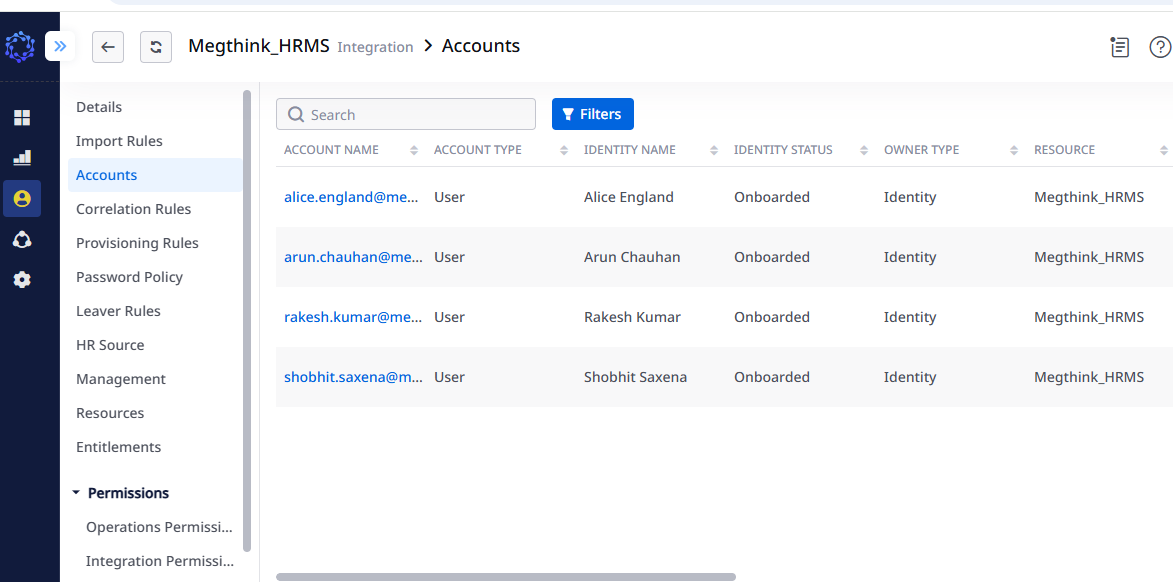
19. Then Click on the Trigger Button.

20. Now Goto Integrations on the left side of the Identity Automation tab.



21.Then Click on the CSV file which you have imported**.**

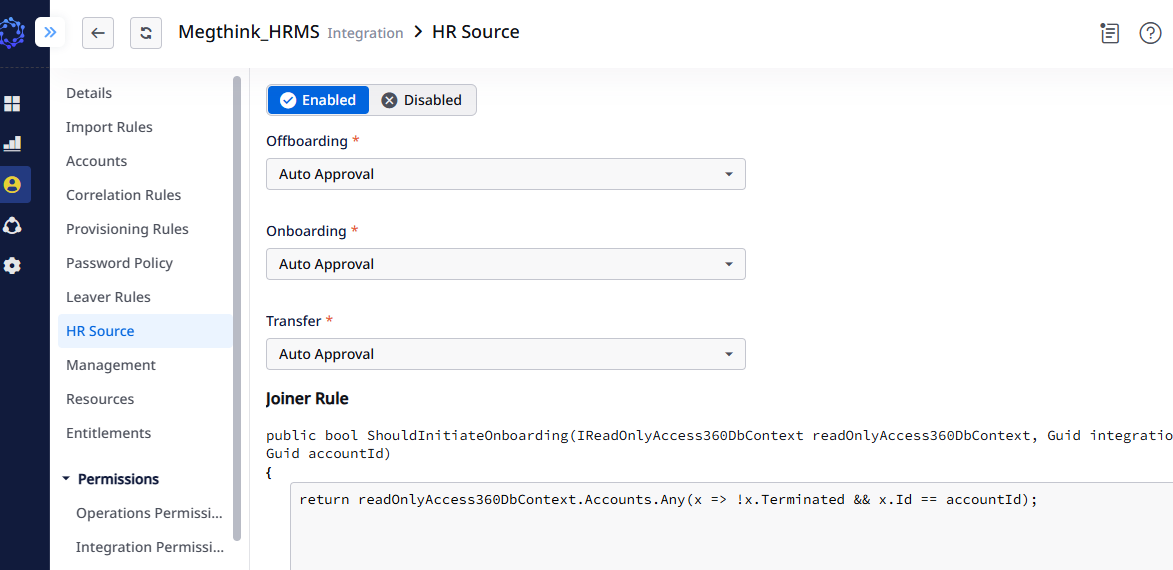
22. Click on the Accounts. See whether the accounts get imported or not.



23. Account Import is Successful.

**14 ONBOARDING:-**

24.Then to enable CSV to HRSouce .Click on HRSource tab.

****

Click on Enabled then Select AUTO APPROVAL from all the DropDown List (i.e);Offbording , OnBoarding and Transfer.

**25.Now GoTo JOINER RULE:-**

**Example Code:-**

return readOnlyAccess360DbContext.Accounts.Any(x => !x.Terminated && x.Id == accountId);

26. Click on Compile Button.

**27. MOVERRULE:-**

**Example Code:-**

return false;

**28.LeaverRule:-**

**Example Code:-**

var terminationDateString = readOnlyAccess360DbContext.AccountStringAdditionalProperties.FirstOrDefault(x => x.AccountId == accountId

&& !x.Terminated

&& !x.Account.Terminated

&& x.Name == "Termination Date"

)?.Value;

var estTimeZone = TimeZoneInfo.FindSystemTimeZoneById("Eastern Standard Time");

if(terminationDateString != default && terminationDateString is not null)

{

DateTime? terminationDate = DateTime.ParseExact(terminationDateString, "dd/MM/yyyy",null);

if(terminationDate is not null && terminationDate <= DateTimeOffset.Now )

{

return true;

}

}

return false;

29**.Reinstatement Rule:-**

**Example Code:-**

return readOnlyAccess360DbContext.Accounts.Any(x => x.IsLocked && x.Id == accountId)

? false

: true;

30. **Identity Mapping Rule:-**

**ExampleCode:-**

var accountData = await readOnlyAccess360DbContext.Accounts

.Where(x => x.IntegrationId == integrationId

&& x.Id == accountId)

.Select(x => new

{

x.Id,

x.Name

})

.FirstOrDefaultAsync();

var accountEmail = await readOnlyAccess360DbContext.AccountStringAdditionalProperties

.Where(x => x.AccountId == accountData.Id && x.Name == "Email")

.OrderByDescending(x => x.LastUpdateTimestamp)

.ThenBy(x => x.Terminated)

.Select(x => x.Value)

.FirstOrDefaultAsync();

var accountName = accountData.Name;

var identityEmail = !string.IsNullOrWhiteSpace(accountEmail) ? accountEmail : accountName;

if (!identityEmail.Contains("@"))

{

identityEmail += "@CSVtest.com.invalid";

}

var accountGivenName = await readOnlyAccess360DbContext.AccountStringAdditionalProperties

.Where(x => x.AccountId == accountData.Id && x.Name == "First Name")

.OrderByDescending(x => x.LastUpdateTimestamp)

.ThenBy(x => x.Terminated)

.Select(x => x.Value)

.FirstOrDefaultAsync();

var accountSurname = await readOnlyAccess360DbContext.AccountStringAdditionalProperties

.Where(x => x.AccountId == accountData.Id && x.Name == "Last Name")

.OrderByDescending(x => x.LastUpdateTimestamp)

.ThenBy(x => x.Terminated)

.Select(x => x.Value)

.FirstOrDefaultAsync();

var identityName = $"{accountGivenName} {accountSurname}";

if (string.IsNullOrWhiteSpace(identityName))

{

identityName = accountName;

}

identityName = identityName.Trim();

/\*\*\*\*\*\*\*\* Account Swap by Email \*\*\*\*\*\*\*\*\*\*\*\*\*\*/

var existingIdentityId = await readOnlyAccess360DbContext.Identities

.Where(x => (x.HrSourceAccountId == accountData.Id

|| x.Email == identityEmail))

.Select(x => x.Id)

.FirstOrDefaultAsync();

var existingIdentityIdValue = existingIdentityId != Guid.Empty ? existingIdentityId : (Guid?)null;

/\*\*\*\*\*\*\*\*\*\*\*\*\* Manager \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

var accountManager = await readOnlyAccess360DbContext.AccountStringAdditionalProperties

.Where(x => x.AccountId == accountData.Id && x.Name == "managerId")

.OrderByDescending(x => x.LastUpdateTimestamp)

.ThenBy(x => x.Terminated)

.Select(x => x.Value)

.FirstOrDefaultAsync();

var accountManagerAccountIdValue =(Guid?)null;

if (accountManager is not null)

{

if (Guid.TryParse(accountManager, out var accountManagerId))

{

var accountManagerAccountId = await readOnlyAccess360DbContext.Accounts

.Where(x => x.Id == accountManagerId)

.OrderByDescending(x => x.LastUpdateTimestamp)

.ThenBy(x => x.Terminated)

.Select(x => x.Id)

.SingleOrDefaultAsync();

accountManagerAccountIdValue = accountManagerAccountIdValue != Guid.Empty ? accountManagerAccountId : (Guid?)null;

}

}

var startDateAccount = await readOnlyAccess360DbContext.AccountStringAdditionalProperties

.Where(x => x.AccountId == accountData.Id && x.Name == "Hired Date" && !x.Terminated)

.OrderByDescending(x => x.LastUpdateTimestamp)

.Select(x => x.Value)

.FirstOrDefaultAsync();

DateTime? startDateAccountValue = null;

if (DateTime.TryParse(startDateAccount , out var startDateAccountValue1))

{

startDateAccountValue = startDateAccountValue1;

}

return new

(

existingIdentityIdValue,

identityName,

identityEmail,

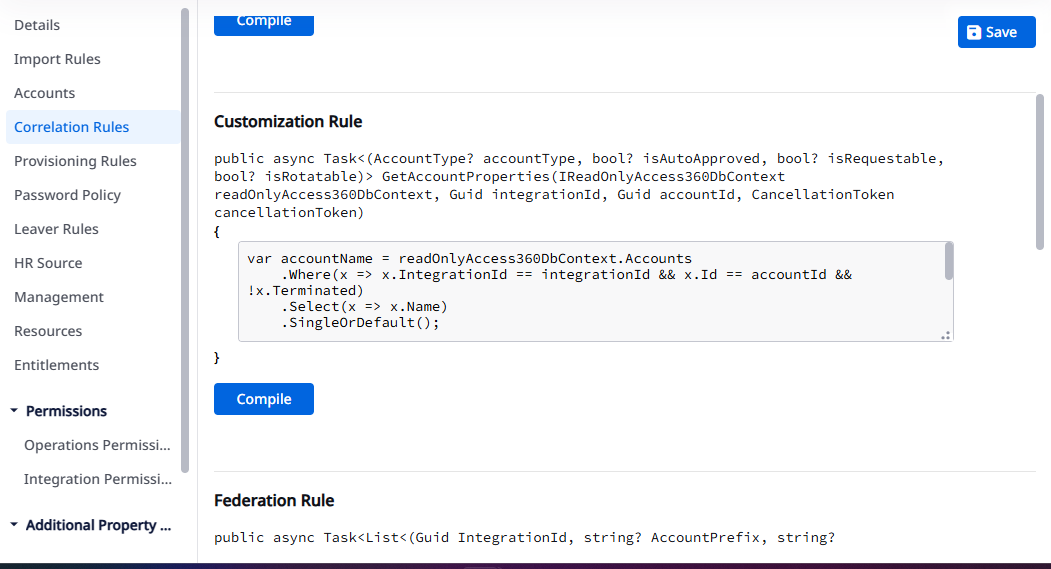
"All Permissions",

accountManagerAccountIdValue,

startDateAccountValue,

Null

);



**Customization Rule:-**

Go to the Correlation rule and add the customization Rule.

Click on compile and save.  
  
**Example code:-**  
  
var accountName = readOnlyAccess360DbContext.Accounts

.Where(x => x.IntegrationId == integrationId && x.Id == accountId && !x.Terminated)

.Select(x => x.Name)

.SingleOrDefault();

if (accountName is not null)

{

if (accountName.StartsWith("adm\_"))

{

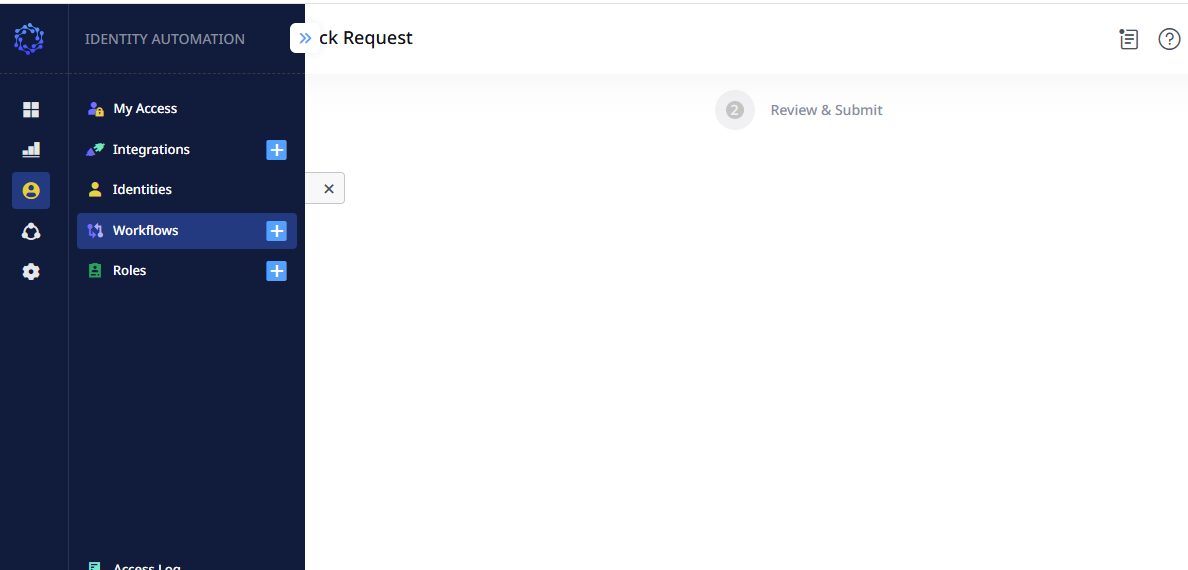
return (AccountType.Privileged, null, null, null);

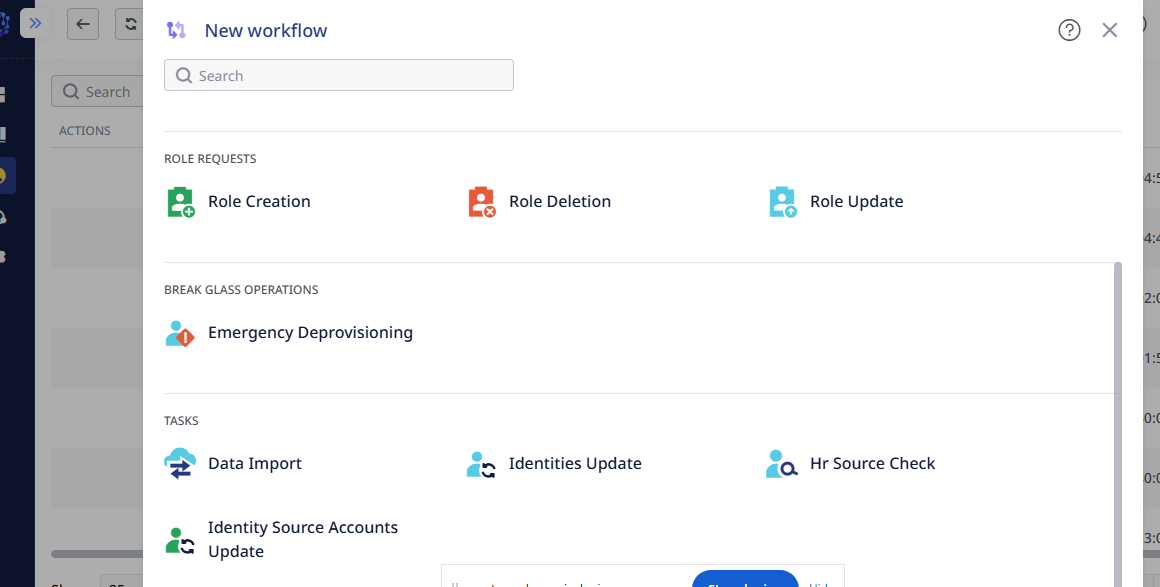
}

}

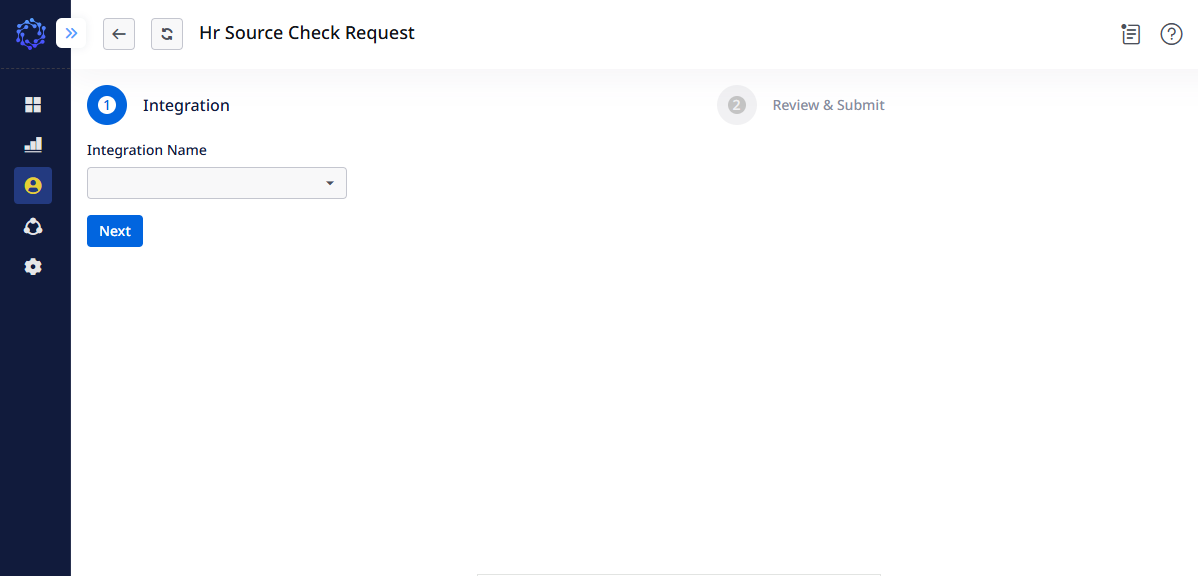
return (AccountType.User, null, null, null);

31.Click On WorkFlow (+)Icon And Select HRSOURCE.

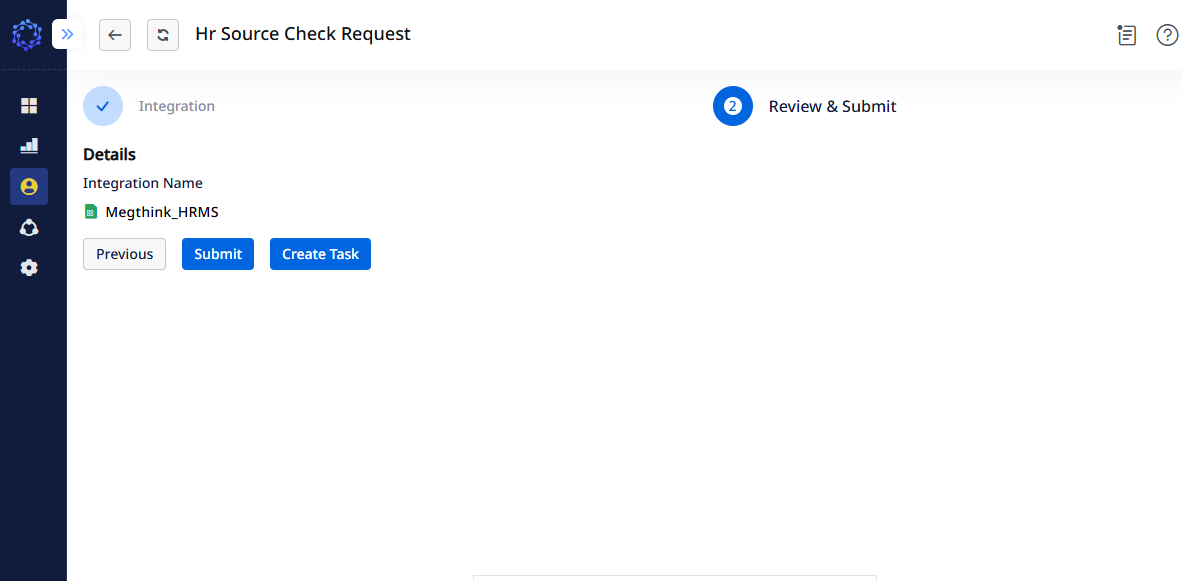


****

**32. Select Integration Name from Dropdown list.**

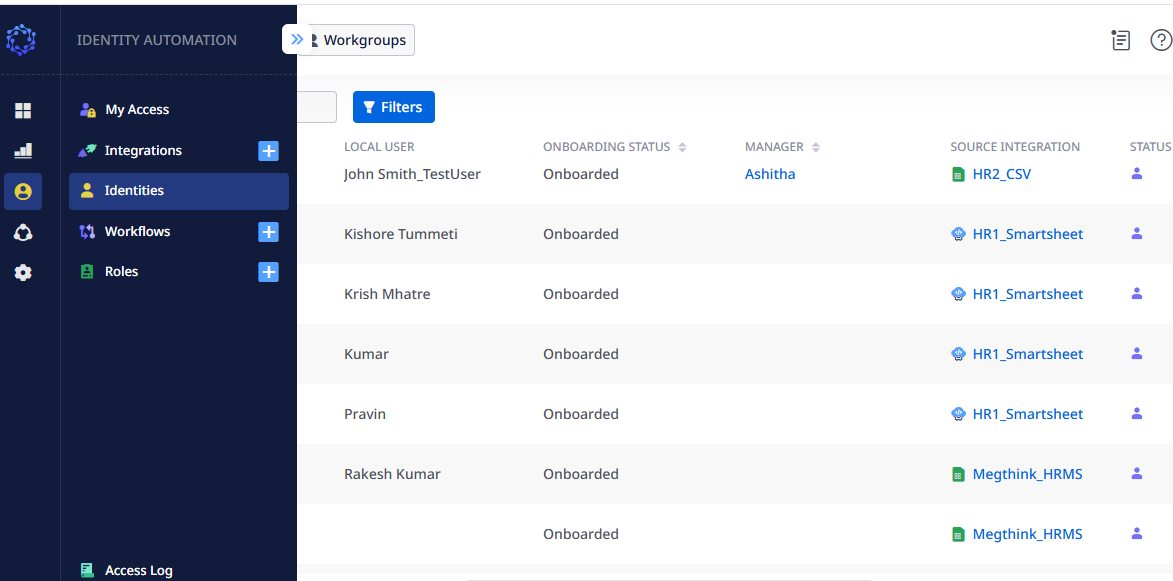
****

**33. Click on Next Button.**

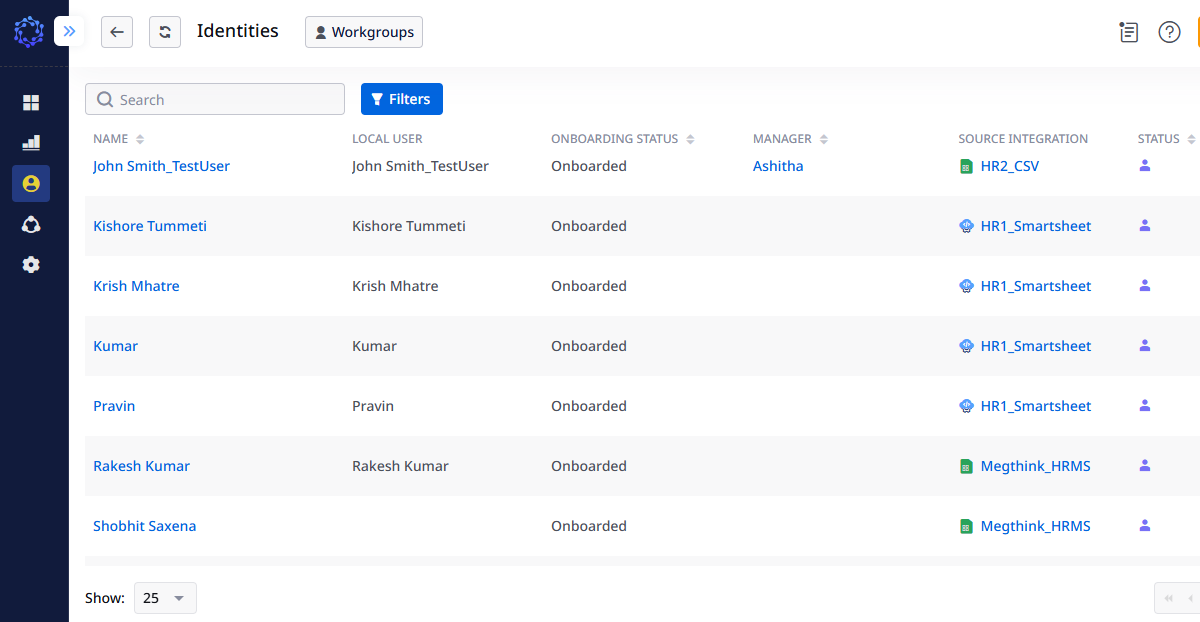
****

**34. Click on the Submit button.**

**35. Goto Identites on the Left Side of Dashboard.**

****

**36.Now you will be able see all the Identities Successfully Onboarded.**

****

**25 37.OFFBOARDING STEPS:-**

**LeaverRule:-**

var terminationDateString = readOnlyAccess360DbContext.AccountStringAdditionalProperties.FirstOrDefault(x => x.AccountId == accountId

&& !x.Terminated

&& !x.Account.Terminated

&& x.Name == "Termination Date"

)?.Value;

var estTimeZone = TimeZoneInfo.FindSystemTimeZoneById("Eastern Standard Time");

if(terminationDateString != default && terminationDateString is not null)

{

DateTime? terminationDate = DateTime.ParseExact(terminationDateString, "dd/MM/yyyy",null);

if(terminationDate is not null && terminationDate <= DateTimeOffset.Now )

{

return true;

}

}

return false;

38. Modify the CSV file ,Add Termination Date to one of the user.

39.Do the DataImport for the modified csv file.

40.Run the HRSourceCheck Task.

41. In the WorkFlows You can see the user with the termination date is offboarded.

