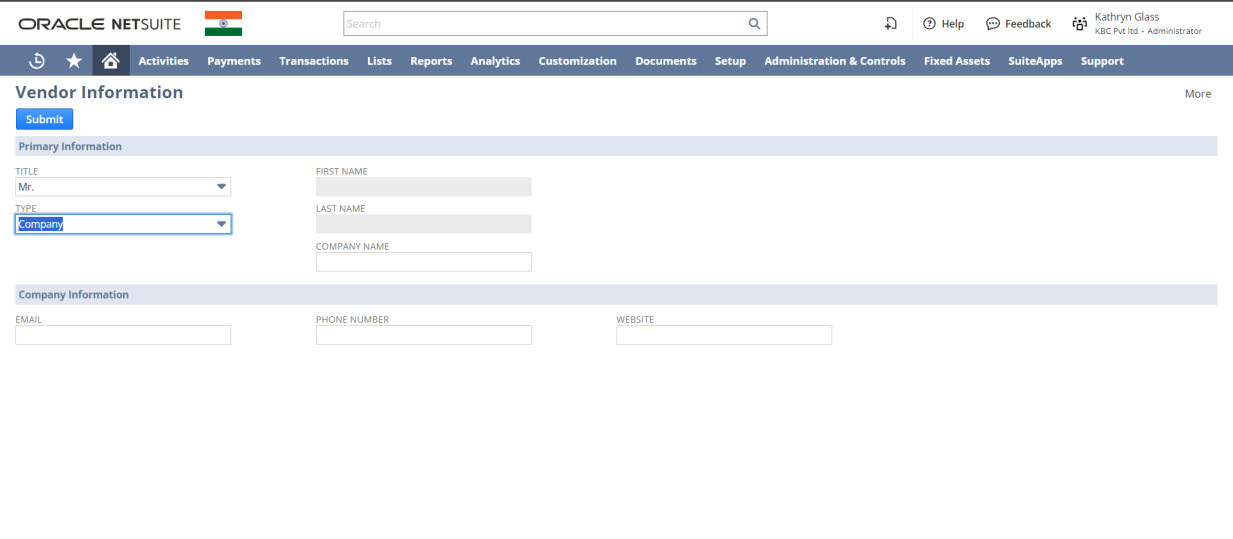
**SUITELET SCRIPT ASSIGNMENT**

**1. Create the following form with** **fields and sublist and link it with the client script. Use the client script to disable or enable the fields according to the vendor type.**



* Here I have used **N/ui/serverWidget** module to work with the user interface within NetSuite.
* I have used **onRequest** entry point which is the only entry point available in SuiteLet and it defines the Suitelet script trigger point.
* It has two methods. i.e. **GET** and **POST**.
* GET method is used to get some data from the server and the POST method is used to submit the forms or create new records.

***Code:***

/\*\*

\* @NApiVersion 2.0

\* @NScriptType Suitelet

\*/

define(['N/ui/serverWidget', 'N/record', 'N/log'],

function(serverWidget, record, log) {

function onRequest(scriptContext) {

if(scriptContext.request.method == 'GET') {

// Question No. 1

{

var form = serverWidget.createForm({

title: 'Vendor Information'

})

form.addButton({

id: 'custpage\_buttonid',

label: 'Submit'

})

// SUBTAB 1

form.addFieldGroup({

id: 'custfield\_subtab1',

label: 'Primary Information'

})

// TITLE

var titleField = form.addField({

id: 'custpage\_title',

label: 'TITLE',

type: serverWidget.FieldType.SELECT,

container: 'custfield\_subtab1'

})

titleField.addSelectOption({

value: 'mr',

text: 'Mr.'

})

titleField.addSelectOption({

value: 'mrs',

text: 'Mrs.'

})

// TYPE

var typeField = form.addField({

id: 'custpage\_type',

label: 'TYPE',

type: serverWidget.FieldType.SELECT,

container: 'custfield\_subtab1'

})

typeField.addSelectOption({

value: 'company',

text: 'Company'

})

// FIRST NAME

form.addField({

id: 'custpage\_firstname',

label: 'FIRST NAME',

type: serverWidget.FieldType.TEXT,

container: 'custfield\_subtab1'

})

// LAST NAME

form.addField({

id: 'custpage\_laststname',

label: 'LAST NAME',

type: serverWidget.FieldType.TEXT,

container: 'custfield\_subtab1'

})

// SUBTAB 2

form.addFieldGroup({

id: 'custpage\_subtab2',

label: 'Company Information'

})

// EMAIL

form.addField({

id: 'custfield\_email',

label: 'EMAIL',

type: serverWidget.FieldType.TEXT,

container: 'custpage\_subtab2'

})

// PHONE NUMBER

form.addField({

id: 'custfield\_phone',

label: 'PHONE NUMBER',

type: serverWidget.FieldType.TEXT,

container: 'custpage\_subtab2'

})

// WEBSITE

form.addField({

id: 'custfield\_website',

label: 'WEBSITE',

type: serverWidget.FieldType.TEXT,

container: 'custpage\_subtab2'

})

}} else{

}

}

return {

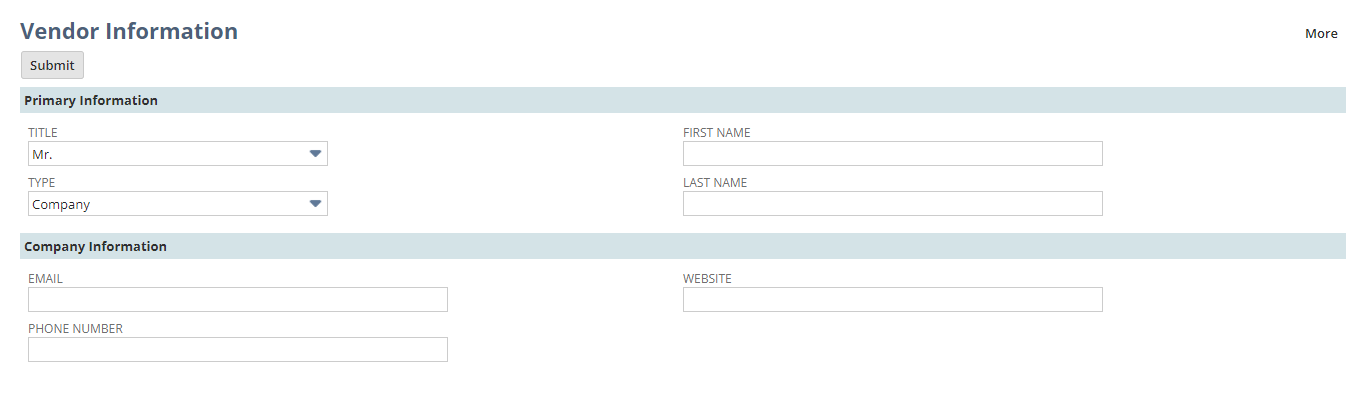
onRequest: onRequest

}

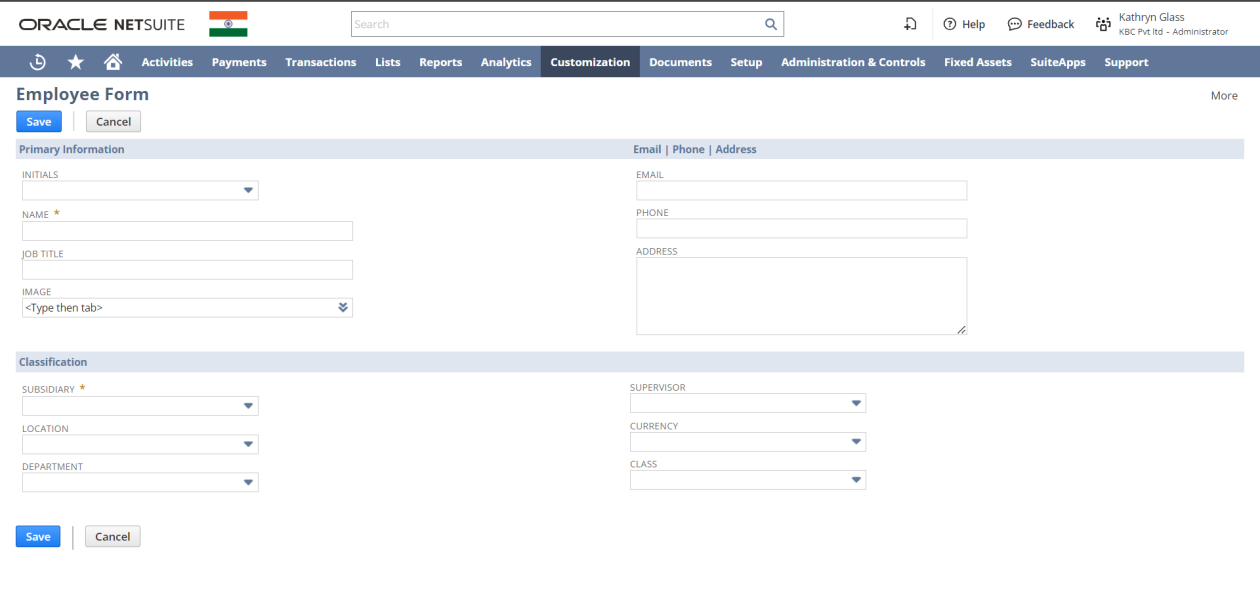
}

)

***Output:***



**2. Create an employee form as shown below, with all the field groups, buttons and the marked fields.**



* Here I have created an employee record dynamically.
* First, I created the form and fetched the fieldId of each custom field from an employee record and set the value of my newly created custom employee form to the particular fieldId under the POST method.

***Code:***

/\*\*

\* @NApiVersion 2.0

\* @NScriptType Suitelet

\*/

define(['N/ui/serverWidget', 'N/record', 'N/log'],

function(serverWidget, record, log) {

function onRequest(scriptContext) {

if(scriptContext.request.method == 'GET') {

// Question No. 2

{

var form = serverWidget.createForm({

title: 'Employee Form'

})

form.addSubmitButton({

label: 'Save'

})

form.addButton({

id: 'custpage\_buttonCancel',

label: 'Cancel'

})

// SUBTAB 1.1

var field\_group1 = form.addFieldGroup({

id: 'custpage\_subtab1\_1',

label: 'Primary Information'

})

field\_group1.isSingleColumn = true

// INITIALS

form.addField({

id: 'custpage\_initials',

label: 'INITIALS',

type: serverWidget.FieldType.SELECT,

container: 'custpage\_subtab1\_1'

})

// JOB TITLE

form.addField({

id: 'custpage\_jobtitle',

label: 'JOB TITLE',

type: serverWidget.FieldType.TEXT,

container: 'custpage\_subtab1\_1'

})

// NAME

var name = form.addField({

id: 'custpage\_name',

label: 'NAME',

type: serverWidget.FieldType.TEXT,

container: 'custpage\_subtab1\_1'

})

name.isMandatory = true

// IMAGE

form.addField({

id: 'custpage\_image',

label: 'IMAGE',

type: serverWidget.FieldType.IMAGE,

container: 'custpage\_subtab1\_1'

})

// SUBTAB 1.2

var field\_group2 = form.addFieldGroup({

id: 'custpage\_subtab1\_2',

label: 'Email | Phone | Address',

})

field\_group2.isSingleColumn = true

// EMAIL

form.addField({

id: 'custpage\_email',

label: 'EMAIL',

type: serverWidget.FieldType.TEXT,

container: 'custpage\_subtab1\_2'

})

// PHONE

form.addField({

id: 'custpage\_phone',

label: 'PHONE',

type: serverWidget.FieldType.TEXT,

container: 'custpage\_subtab1\_2'

})

// ADDRESS

form.addField({

id: 'custpage\_address',

label: 'ADDRESS',

type: serverWidget.FieldType.TEXTAREA,

container: 'custpage\_subtab1\_2'

})

// SUBTAB 2

form.addFieldGroup({

id: 'custpage\_subtab2',

label: 'Classification',

})

var subsidary = form.addField({

id: 'custpage\_subs',

label: 'SUBSIDARY',

type: serverWidget.FieldType.SELECT,

container: 'custpage\_subtab2'

})

subsidary.addSelectOption({

text: 'BFL',

value: 12 // internal id of that subsidary

})

subsidary.isMandatory = true

form.addField({

id: 'custpage\_loca',

label: 'LOCATION',

type: serverWidget.FieldType.SELECT,

container: 'custpage\_subtab2'

})

form.addField({

id: 'custpage\_dept',

label: 'DEPARTMENT',

type: serverWidget.FieldType.SELECT,

container: 'custpage\_subtab2'

})

form.addField({

id: 'custpage\_supervisor',

label: 'SUPERVISOR',

type: serverWidget.FieldType.SELECT,

container: 'custpage\_subtab2'

})

form.addField({

id: 'custpage\_currency',

label: 'CURRENCY',

type: serverWidget.FieldType.SELECT,

container: 'custpage\_subtab2'

})

form.addField({

id: 'custpage\_class',

label: 'CLASS',

type: serverWidget.FieldType.SELECT,

container: 'custpage\_subtab2'

})

}

scriptContext.response.writePage(form);

}else{

// Question No. 2

{

// getting the name

var fieldValueName = scriptContext.request.parameters.custpage\_name

log.debug('Field value is : ' + fieldValueName)

var newRecord = record.create({

type: record.Type.EMPLOYEE,

isDynamic: true

})

newRecord.setValue({ // setting the name

fieldId: 'firstname',

value: fieldValueName

})

// getting subsidary

var fv\_subsidary = scriptContext.request.parameters.custpage\_subs

log.debug('Subsidary is ' + fv\_subsidary)

newRecord.setValue({

fieldId: 'subsidiary',

value: fv\_subsidary

})

// getting job title

var fv\_jobTitle = scriptContext.request.parameters.custpage\_jobtitle

newRecord.setValue({

fieldId: 'title',

value: fv\_jobTitle

})

// getting mail

var fv\_mail = scriptContext.request.parameters.custpage\_email

newRecord.setValue({

fieldId: 'email',

value: fv\_mail

})

// getting phone

var fv\_phone = scriptContext.request.parameters.custpage\_phone

newRecord.setValue({

fieldId: 'phone',

value: fv\_phone

})

// getting address

var fv\_address = scriptContext.request.parameters.custpage\_address

newRecord.setValue({

fieldId: 'defaultaddress',

value: fv\_address

})

//getting image

fv\_image = scriptContext.request.parameters.custpage\_image

newRecord.setValue({

fieldId: 'image',

value: fv\_image

})

var recordId = newRecord.save()

scriptContext.response.write('Record created with record id: ' + recordId)

}

}

}

return {

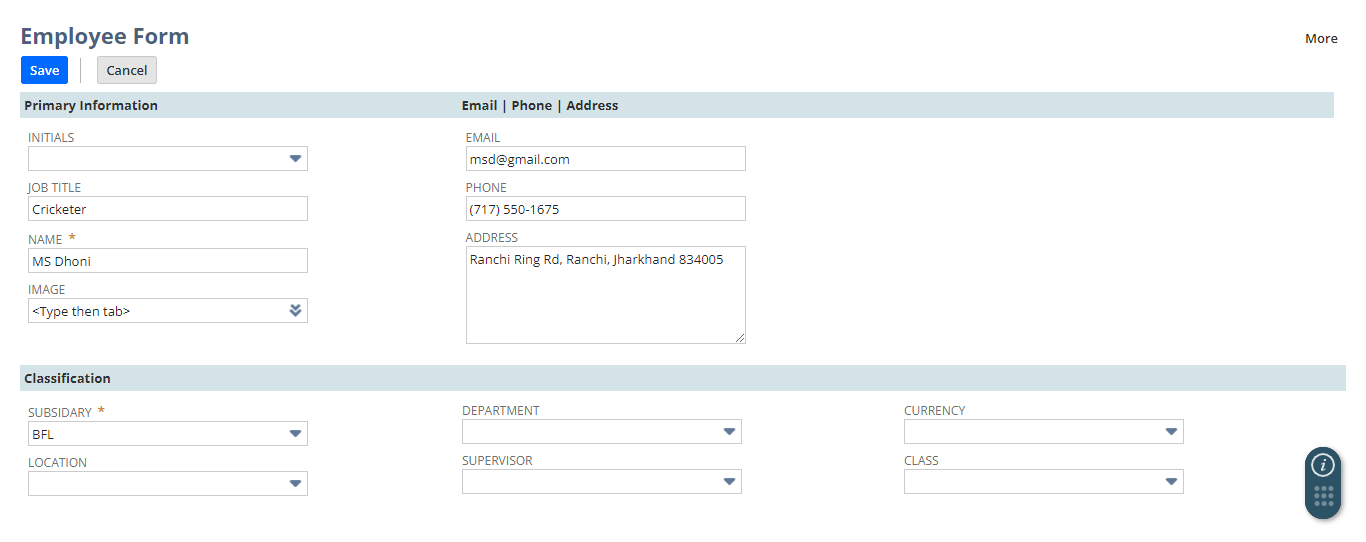
onRequest: onRequest

}

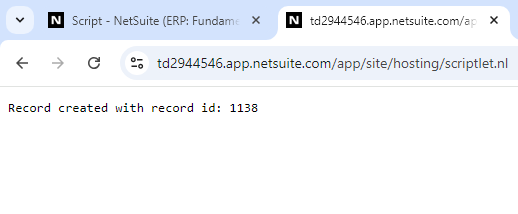
}

)

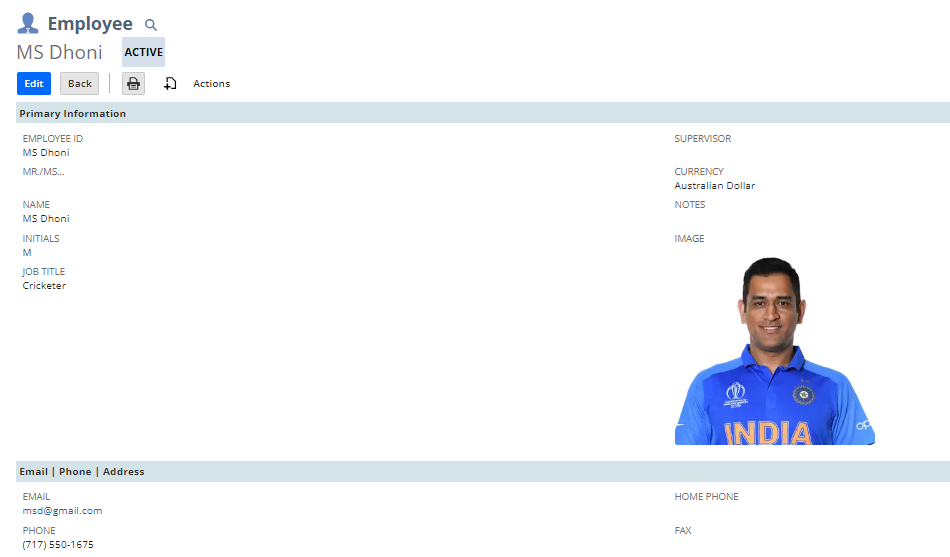
***Output:***



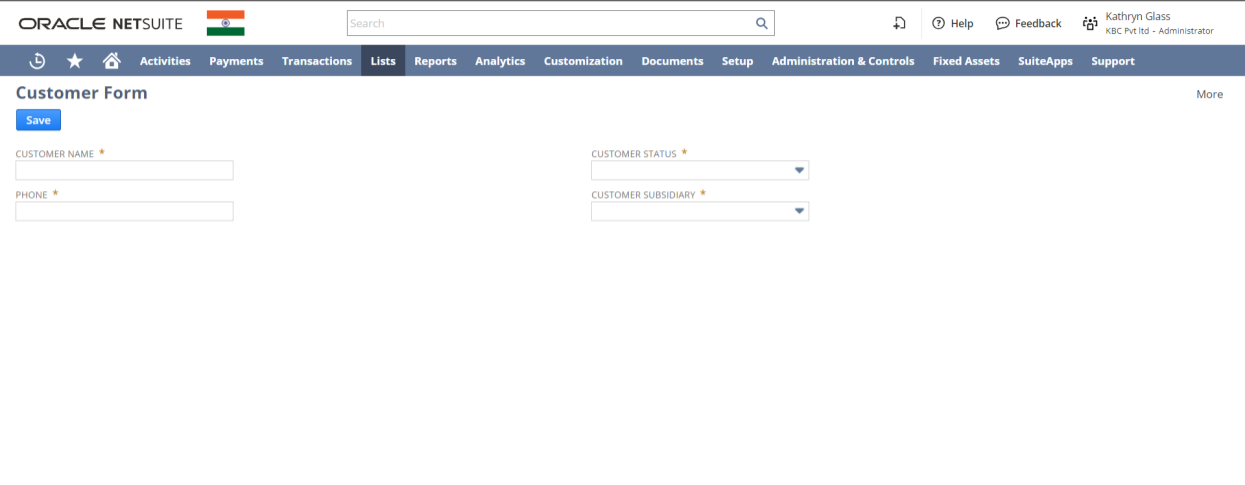
After adding the detail in the field and after clicking the Save button which is the **addSubmitButton** a record id will be generated for this particular record.



And now by navigating to List -> Employees -> Employees, you will be able to get the particular record.



**3. Create a customer record in NetSuite using the same suitelet.**



* Here I have created a customer dynamically.
* First, I created the form and fetched the fieldId of each custom field from a customer record and set the value of my newly created custom customer form to the particular fieldId under the POST method.

/\*\*

\* @NApiVersion 2.0

\* @NScriptType Suitelet

\*/

define(['N/ui/serverWidget', 'N/record', 'N/log'],

function(serverWidget, record, log) {

function onRequest(scriptContext) {

if(scriptContext.request.method == 'GET') {

// Question No. 3

{

var form = serverWidget.createForm({

title: 'Customer Form'

})

form.addSubmitButton({

label: 'Save'

})

// CUST NAME

var name = form.addField({

id: 'custpage\_name',

label: 'CUSTOMER NAME',

type: serverWidget.FieldType.TEXT

})

name.isMandatory = true

// PHONE

var phn = form.addField({

id: 'custpage\_phone',

label: 'PHONE',

type: serverWidget.FieldType.TEXT

})

phn.isMandatory = true

// EMAIL

var email = form.addField({

id: 'custpage\_email',

label: 'EMAIL',

type: serverWidget.FieldType.TEXT

})

email.isMandatory = true

// STATUS

var status = form.addField({

id: 'custpage\_status',

label: 'CUSTOMER STATUS',

type: serverWidget.FieldType.SELECT

})

status.isMandatory = true

status.addSelectOption({

text: 'Lost Customer', // CUSTOMER-Lost Customer

value: 16

})

status.addSelectOption({

text: 'Closed Won', //CUSTOMER-Closed Own

value: 13

})

status.addSelectOption({

text: 'Renewal', //CUSTOMER-Renewal',

value: 15

})

// SUBSIDIARY

var subs = form.addField({

id: 'custpage\_subs',

label: 'CUSTOMER SUBSIDIARY',

type: serverWidget.FieldType.SELECT

})

subs.isMandatory = true

subs.addSelectOption({

text: 'Canada',

value: 7

})

subs.addSelectOption({

text: 'Australia',

value: 9

})

subs.addSelectOption({

text: 'United Kingdom',

value: 6

})

}

scriptContext.response.writePage(form);

}else{

// Question No. 3

{

var newRecord = record.create({

type: record.Type.CUSTOMER,

isDynamic: true

})

var fieldValue\_name = scriptContext.request.parameters.custpage\_name

newRecord.setValue({

fieldId: 'companyname',

value: fieldValue\_name

})

var fv\_phone = scriptContext.request.parameters.custpage\_phone

newRecord.setValue({

fieldId: 'phone',

value: fv\_phone

})

var fv\_email = scriptContext.request.parameters.custpage\_email

newRecord.setValue({

fieldId: 'email',

value: fv\_email

})

var fv\_status = scriptContext.request.parameters.custpage\_status

newRecord.setValue({

fieldId: 'entitystatus',

value: fv\_status

})

var fv\_subs = scriptContext.request.parameters.custpage\_subs

newRecord.setValue({

fieldId: 'subsidiary',

value: fv\_subs

})

var recordId = newRecord.save()

scriptContext.response.write('Record created with record id: ' + recordId)

}

}

}

return {

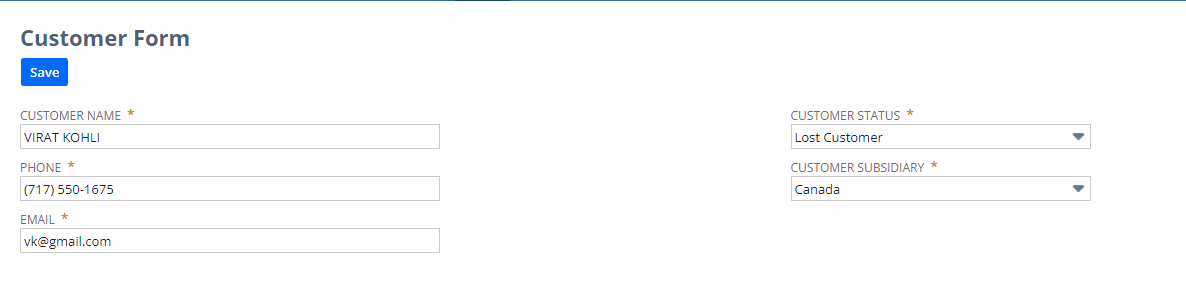
onRequest: onRequest

}

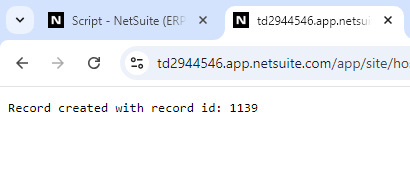
}

)

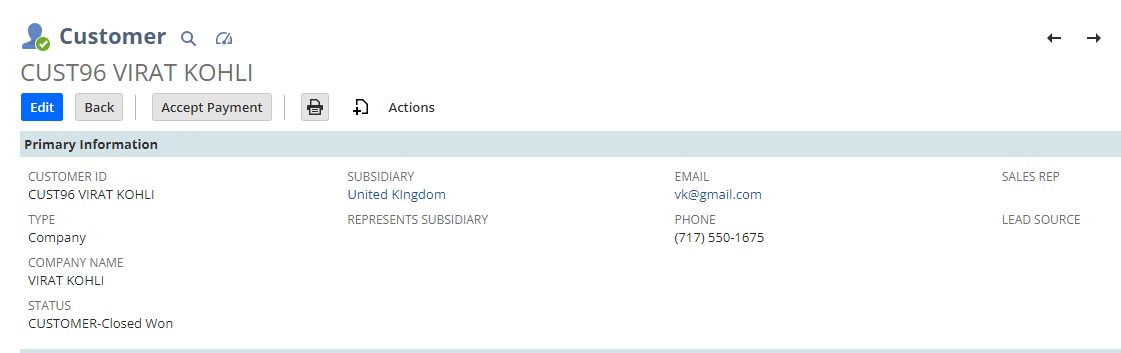
***Output:***



Now by putting all the values in the respective field, when we click Save a new record will be generated with a record id.



Now we will be able to see the new customer by navigating to Lists -> Relationship -> Customer.



**4.How to link the suitelet to the Center Tab.**

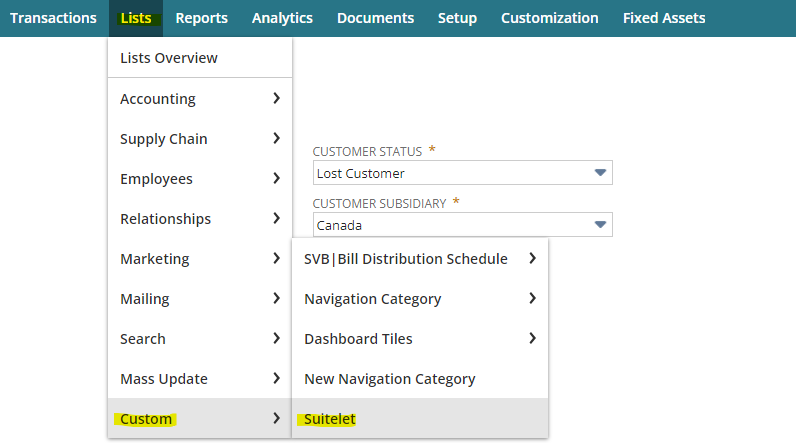
Step 1 – Go to the script deployment and find that particular suiteLet script. Click on Edit.

Step 2 – Under Links subtab add all the value you want as per your requirement as where to show this ..

Then Click on Save



Now we can find it under Lists -> Custom -> SuiteLet and then we will be able to access that script file.



**5.What are the ways to link the client script with suitelet ? Explain in detail Id and path**

There are primarily two ways to link a client script with suitelet. i.e. Link by script ID and Link by path

1. ***Link by script ID:*** This method links the Client Script to the SuiteLet using the SuiteLet's Script ID which is unique.

Step 1 – Create a Client Script and write the alert function inside the pageInit trigger point.

Step 2 – Upload this script in the file cabinet under SuiteScript file and get the internal id of it.

Step 3 – Go to the SuiteLet deployed script and use the **form.clientScriptFieldId** API and it is used while attaching a client script to a server script.

***ClientScript Code:***

define([],

function() {

function pageInit(scriptContext) { // Internal Id - 8916

alert('pageInit triggered')

}

return{

pageInit: pageInit

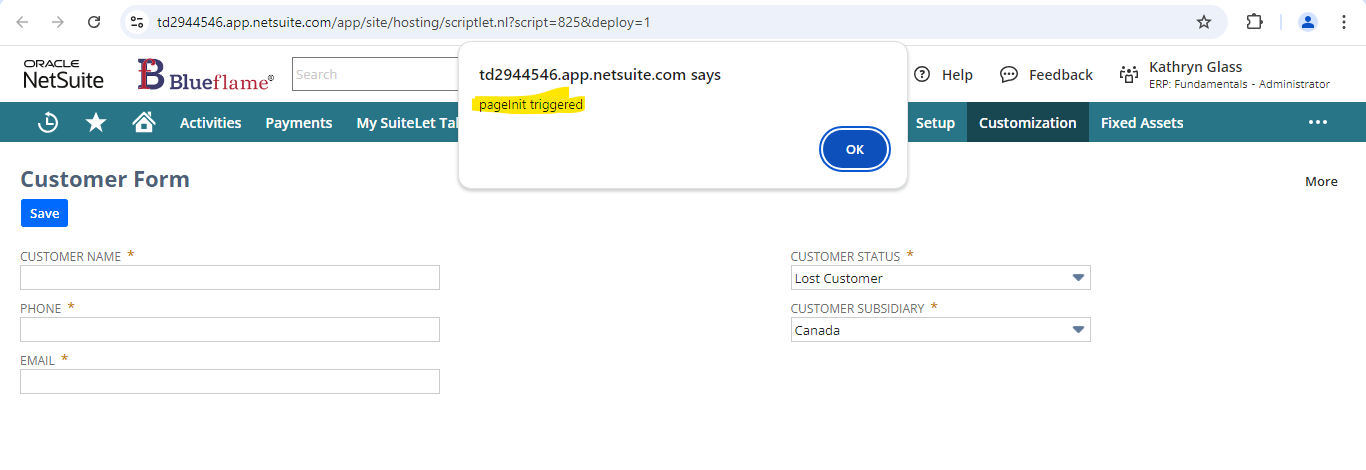
}

}

)

***SuiteLet Code:***

form.clientScriptFileId = 8916



1. ***Link by path:***

Step 1 – Create a Client Script and write the alert function inside the pageInit trigger point.

Step 2 – Upload this script in the file cabinet under SuiteScript file.

Step 3 – Go to the SuiteLet deployed script and use the **form.clientScriptModulePath** API and give the path of that client script from file cabinet by 'SuiteScripts/fileName.js'

***ClientScript Code:***

define([],

function() {

function pageInit(scriptContext) { // Internal Id - 8916

alert('pageInit triggered')

}

return{

pageInit: pageInit

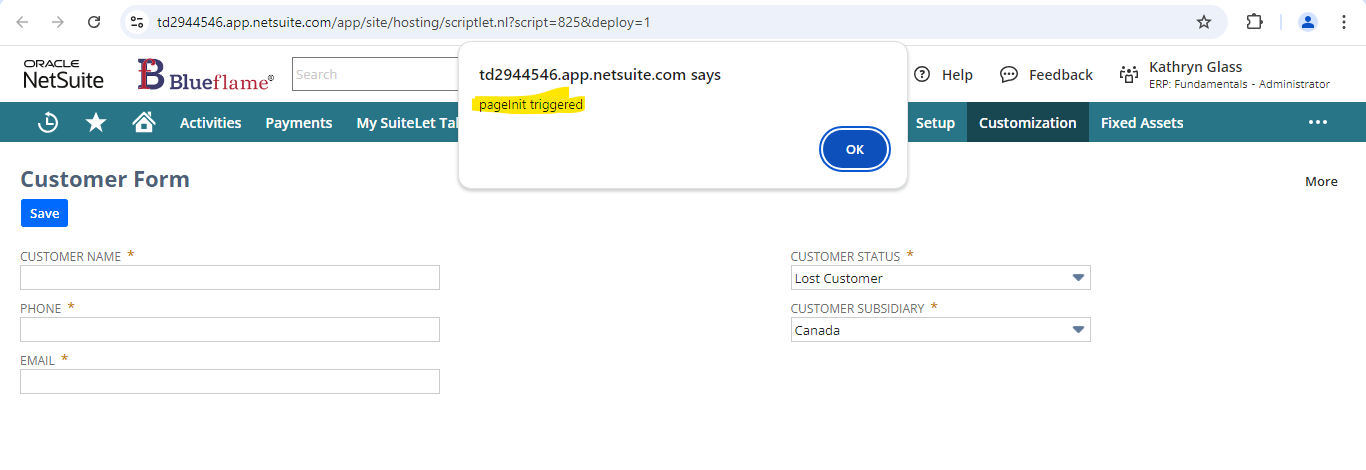
}

}

)

***SuiteLet Code:***

form.clientScriptModulePath = 'SuiteScripts/CLI\_linkToSL.js'



**6. How can I return to the same suitelet (GET) after submitting the form?**

Step 1 – In the define section add a module **‘N/redirect’** to customize navigation within NetSuite

Step 2 – Now there is an API called **redirect.toSuitelet,** which helps to redirect the user to a Suitelet.

Step 3 – Add the below code snippet under the POST function (basically in the else part).

Step 4 – Provide the scriptId and deploymentId of that script.

redirect.toSuitelet({

scriptId: '825',

deploymentId: '1'

});