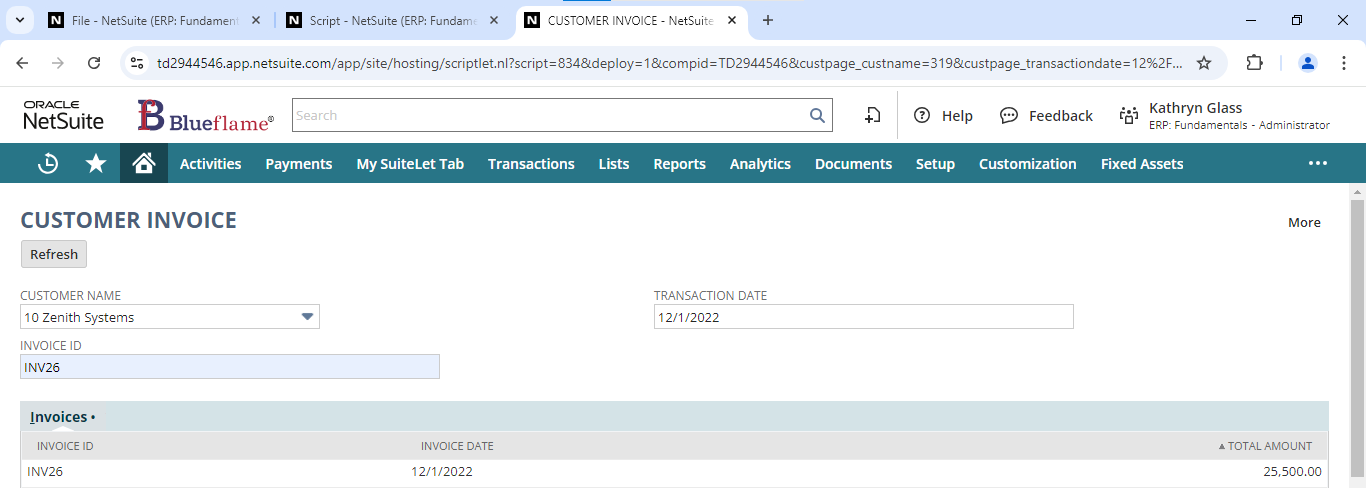
**TECHNICAL ASSIGNMENT**

**Que.1 Create a suitelet which will list out the invoices based on three filters i.e. Customer name, Invoice Id and transaction date.**



**Que.2 Create a custom button on the same suitelet which will trigger the Map Reduce script which will send an email to the customer with the attached invoices in the form of PDF.**

**Que.3 Write a script to add ‘other charge’ item on end of the sales order line level item. And its amount will be the sum of all existing items of the sales order and quantity would be hard coded to 1.**

Here I used User Event Script, and beforeSubmit entry point. So, once I added my required items in the item sublist and save the record, a new sublist will be automatically added with the net quantity and net amount of the item in that sales order.

const beforeSubmit = (scriptContext) => {

log.debug('Before submit triggered')

if(scriptContext.type === scriptContext.UserEventType.CREATE || scriptContext.type === scriptContext.UserEventType.EDIT){

var current\_record = scriptContext.newRecord

var itemCount = current\_record.getLineCount({

sublistId: 'item'

})

var totalPrice = 0

var netQuantity = 0

for(var i=0; i<itemCount; i++){

var currentPrice = current\_record.getSublistValue({

sublistId: 'item',

fieldId: 'rate',

line: i

})

totalPrice += currentPrice

var currentQuantity = current\_record.getSublistValue({

sublistId: 'item',

fieldId: 'quantity',

line: i

})

netQuantity += currentQuantity

}

var otherChargeItemId = 201

current\_record.insertLine({

sublistId: 'item',

line: itemCount

})

current\_record.setSublistValue({

sublistId: 'item',

fieldId: 'item',

line: itemCount,

value: otherChargeItemId

})

current\_record.setSublistValue({

sublistId: 'item',

fieldId: 'description',

line: itemCount,

value: `Net price of above ${itemCount} items`

})

current\_record.setSublistValue({

sublistId: 'item',

fieldId: 'rate',

line: itemCount,

value: totalPrice

})

current\_record.setSublistValue({

sublistId: 'item',

fieldId: 'quantity',

line: itemCount,

value: netQuantity

})

log.debug(`Other charge item added with quantity ${netQuantity} and price ${totalPrice}`)

}

}

