Requirement 1:

Create a field on Account named "Number of Contacts". Populate this field with the number of contacts related to an account.

Code:

CountRelatedContactsHandler:

public class CountRelatedContactsHandler {

public static void call(List<contact> newcontact, List<contact> oldcontact){

set<id> accIds= new set<id>();

if(newcontact != null){

for(Contact c : newcontact){

if(c.AccountId!=null){

accids.add(c.accountid);

}

}

}

if(oldcontact != null){

for(Contact c : oldcontact){

accids.add(c.accountid);

}

}

List<Account> accList = [Select Id, Number\_of\_Contacts\_\_c, (Select id from Contacts) from Account where Id IN :accIds];

if(accList!=null){

for(Account acc : accList){

acc.Number\_of\_Contacts\_\_c = acc.Contacts.size();

}

}

if(!accList.isEmpty()){

update accList;

}

}

}

Trigger: CountRelatedContacts

trigger CountRelatedContacts on Contact (after insert, after update, after delete, after undelete) {

//call handler for best practice

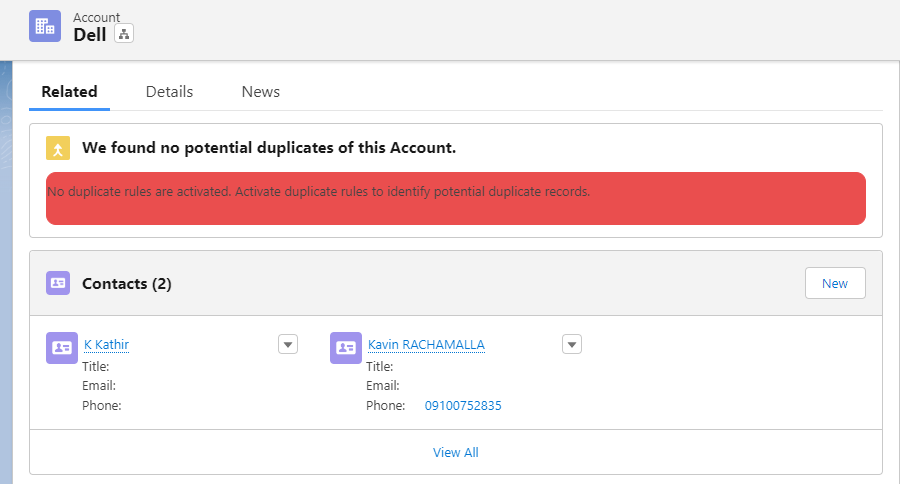
if(Trigger.isinsert || Trigger.isupdate || trigger.isdelete || Trigger.isundelete){

CountRelatedContactsHandler.call(trigger.new, trigger.old);

}

}

Result:



Graphical user interface, text, application, email

Description automatically generated

Requirement 2:

Build a basic lightning component that can query a list of 10 most recently created accounts and display it using a lightning app.

Apex controller:

public class DisplayListOfRecordsUsingIteration {

@AuraEnabled

public static list<Account> fetchAccountRecords(){

return [select Name,Rating,Phone from Account Order By createdDate DESC LIMIT 10];

}

}

Lightning Component:

<aura:component controller="DisplayListOfRecordsUsingIteration">

<aura:attribute name="ShowRecords" type="List"/>

<aura:handler name="init" value="{!this}" action="{!c.Accountcontroller}" />

<div class="size">

<table class="slds-table slds-table--bordered slds-table--striped">

<thead>

<tr>

<th scope="col"><span class="slds-truncate"> Name</span></th>

<th scope="col"><span class="slds-truncate"> Rating</span></th>

<th scope="col"><span class="slds-truncate"> Phone</span></th>

</tr>

</thead>

<tbody>

<aura:iteration items="{!v.ShowRecords}" var="A">

<tr>

<td>{!A.Name}</td>

<td>{!A.Rating}</td>

<td>{!A.Phone}</td>

</tr>

</aura:iteration>

</tbody>

</table>

</div>

</aura:component>

Javascript Controller:

({

Accountcontroller : function(component, event, helper) {

helper.Accounthelpermethod(component);

}

})

Javascript Helper:

({

Accounthelpermethod:function(component,event,helper){

var action = component.get("c.fetchAccountRecords");

action.setCallback(this,function(a){

component.set("v.ShowRecords",a.getReturnValue());

});

$A.enqueueAction(action);

}

})

Css:

.THIS.size {

height:1300px;

width:1500px;

}

Application:

<aura:application Extends="force:slds">

<c:DisplayListOfRecordsUsingIteration/>

</aura:application>

Result:

Graphical user interface, application

Description automatically generated

Requirement 3:

Make a basic http callout and print the result using system.debug

Code:

public class postman\_echo {

public String endPoint;

public String headers;

public String result;

public Integer statuscode;

public void request(){

endPoint='https://postman-echo.com/get?foo1=bar1&foo2=bar2';

headers='{"x-forwarded-proto": "https","x-forwarded-port": "443","host": "postman-echo.com","x-amzn-trace-id": "Root=1-6442203d-541a32c616b31e0c636107b9",';

headers=headers+'"sec-ch-ua": "\"Chromium\";v=\"112\", \"Google Chrome\";v=\"112\", \"Not:A-Brand\";v=\"99\"","sec-ch-ua-mobile": "?0","sec-ch-ua-platform": "\"Windows\"","upgrade-insecure-requests": "1", "user-agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36",';

headers=headers+'"accept": "text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,\*/\*;q=0.8,application/signed-exchange;v=b3;q=0.7","sec-fetch-site": "cross-site","sec-fetch-mode": "navigate","sec-fetch-user": "?1",';

headers=headers+'"sec-fetch-dest": "document","referer": "https://mail.google.com/","accept-encoding": "gzip, deflate, br","accept-language": "en-US,en;q=0.9,pt;q=0.8"}';

Http http = new Http();

HttpRequest request = new HttpRequest();

request.setEndpoint(endPoint);

request.setMethod('GET');

request.setHeader('Cookie', headers);

HttpResponse response=http.send(request);

result=response.getBody();

statuscode=response.getStatusCode();

system.debug('Status Code:'+statuscode);

system.debug('Result:'+result);

}

}

Result:

Graphical user interface, text, application, email

Description automatically generated