**Architectural overview:**

Org Details: There are Two Salesforce orgs involved:

1. The New Org with domain name as neworg1 is the go-forward implementation (ORG-1 mentioned in the assignment).

Credentials are(user name :ruchika@neworg1.org, password: test12345 It will ask for verification code at time of logging in I can send that)

1. The old org with domain name as oldorg is the go-forward implementation (Legacy ORG-2 mentioned in the assignment).

Credentials are(user name :ruchika@org1.org, password: test12345 )

Test Data:

The oldorg has test accounts created.

Metadata on orgs:

oldorg : Created connected app named 'New Org Connect' to use OAuth in REST callout

neworg1:

* + - Created button 'Fetch Old Accounts' am account List view. Available in https://github.com/ruchikasharma029/IntegrationExample/blob/master/src/Objects/Account.object
    - Created remote site settings for salesforce login and connecting to oldorg.
    - Created custom external id field ‘Ext Id’ to for reference to matching record in old org. Available in https://github.com/ruchikasharma029/IntegrationExample/blob/master/src/Objects/Account.object
    - Created Static resource for JQuery reference. Available in https://github.com/ruchikasharma029/IntegrationExample/tree/master/src/staticresources
    - Created custom Setting REST Configuration for saving the org wise login credentials. Available in https://github.com/ruchikasharma029/IntegrationExample/blob/master/src/Objects/REST\_Configuration\_\_c.object

Development

neworg1:

1. The new Visual force page Named 'AccountListView' is created in neworg1 which will fetch the Accounts from oldorg.available in https://github.com/ruchikasharma029/IntegrationExample/blob/master/src/pages/AccountListView.page

* Page Description:

This page can be accessed from Account list view page. I have created new custom button 'Fetch Old Accounts' on Account list view page. When the button is clicked the AccountListView is displayed and it will ask for login credentials for the org you want to fetch records in this case we will use oldorg. If you want to use some other org the REST webservice to return similar to class 'AccountListREST'

should be implemented. Once correct login credentials are provided( you have to append security token with password if your IP is not

included in security controls). Id the org we used credentials had account records they will be shown on the page.

* This page has button to create new Account record in neworg1.
* This page has buttons to go back to list view page or the is you want to change the credentials
* This page is read only (for fetch maximum records as per view state)
* This page has pagination
* This page has restriction of fetching only 4000 records max to overcome view state issue.

1. The new visaulforce controller(apex class) named ‘AccountsListviewExt’ is created for page 'AccountListView' . available in https://github.com/ruchikasharma029/IntegrationExample/blob/master/src/classes/AccountsListViewExt.cls

Controller Description:

* This controller logs into the org using credentials provided on page and fetches the accounts
* I am using partner login API to log in and get the session id for REST request (old school but still works)
* If there is some error in login or REST call error is shown on page
* If the login is done but the other org does not have any Account records, error is shown on page.
* This controller takes care of pagination and go back logic.

1. The new Trigger Handler class for Account after insert trigger (apex class) named ‘AccountTriggerHandler’ is created for page 'AccountTrigger'. available in https://github.com/ruchikasharma029/IntegrationExample/blob/master/src/classes/AccountTriggerHandler.cls

Class Description:

* This class has inner class ‘LoginSetup’ which takes care of login into oldorg using OAuth thru connected app setup (I used different method here just to show that different methods are available to log into salesforce orgs). The custom setting REST Configuration will be used for the credentials.
* This class has future method, which fires every time Account record(s) inserted ,to log into the old org and fetch matching record (if any) using REST callout and gets the id of the record from old org in response and populate that in Ext\_Id field.
* When one Account record is inserted then I am using REST Get call and appending account name to the endpoint url
* When more than one Account records are inserted (bulk data) I am using REST POST call (not very appropriate) to handle bulk insert, as there will be list of account names to be sent to REST webservice to find match.

1. The new Trigger Account after insert named 'AccountTrigger'

oldorg:

1. REST webservice Class(Restresourse) ‘AccountListREST’. Available in https://github.com/ruchikasharma029/IntegrationExample/blob/master/srcRESTwebservice/classes/AccountListREST.cls

Class Description:

* This class is the REST API which handled REST post and get calls and exposes data appropriately.
* I am making use of same GET API for visualforce page and trigger future methods when there is only one record inserted.

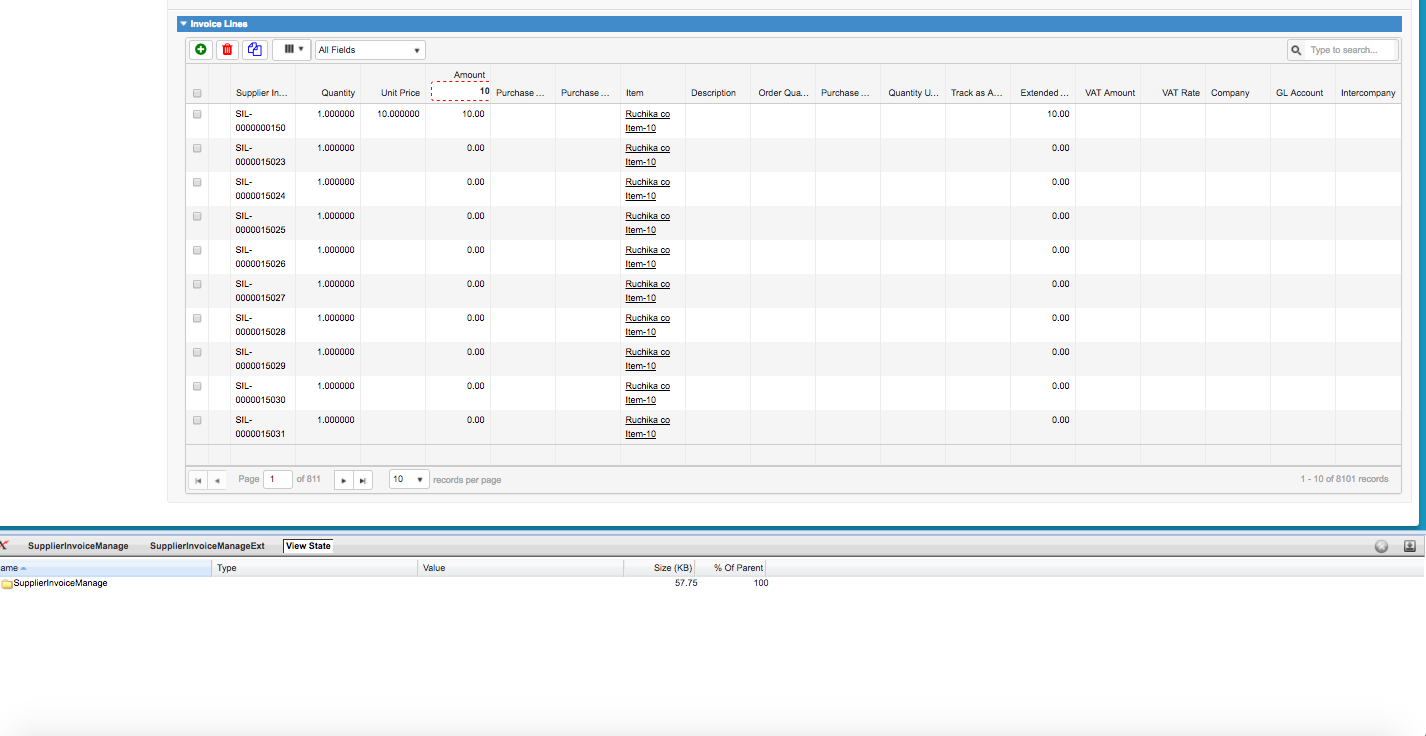
**Alternative solutions:**

Though clear design instructions were give, following are the alternative options I could think of

1. For UI instead of using Visualforce Page block tables we can use third party components(for example Kendo), which can be embedded in visualforce this will help us fetching and showing much more records to users than we can with the use of standard visualforce.

For example in the above mentioned page ‘AccountListView’ I could only fetch 4000 records due to view state and even if I made page read only and if I show less columns the max could be 10,000 as we will be using standard set controller for pagination and we have salesforce restriction for 10,000.

Below figure shows the use of kendo:

****

As you can see there are 811 pages of 10 records so total records fetched are 8000+ and view state is still 57kb so for these many columns we can display at least 25000 records without hitting view state. An moreover we do not have to worry about pagination(it comes out of the box) and there are my added features and events we can make use of.

We can make use of Sales force lightening components(similar to kendo they use aura) and build good UI with less view state.

1. For Fetching records from other salesforce org we can skip creating REST API on oldorg as we can connect through sfroce and male use of Query and query more APIs (either in controller or from Javascript using visualforce remotting )

For example we can make use of following code:

|  |
| --- |
| PageReference theUrl = new PageReference(SERVER\_URL+ '/services/data/v22.0/query/'); |

theUrl.getParameters().put('q','Select a.Phone, a.Name, a.CreatedBy.FirstName, a.CreatedById From Account a limit 10');

and we can set this url as endpoint request.setEndpoint(theUrl.getUrl());

In this way we will make use of services and query to get records using REST. But if wa want to restrict the data access then REST api is more appropriate.

1. Salesforce to Salesforce setup can be used to create this data sharing relationship.
2. We can make use of middleware for integration like informatica ,ETL, Boomi. We are using Boomi.
3. Instead of future method we can make use of Apex batch also