USE CASE

----

1) **Project (EXCEl Sheet/ User Requirements gathering)**
2) **Design Phase Document (Technical Architecture)**
3) **Agile process (plan execution)**
   Tools:
   1) **JIRA (Project overview, tasks), GITLAB**
   2) **AZURE Devops (Project overview, tasks, Gitlab)**

AZURE DEVOPS DEVELOPMENT

Azure Devops : Cloud based VSTSI
Project Management: JIRA : **Azure Board**

Source Control : Gitlab : **Azure Repos :**
Central repository : Maintains history of changes : so all developers can work together
Push their changes

Build and Deployment : release Cycles : **Pipeline CI/CD**
Bring together all the puzzle by all developer together and deploy to the server
CI/CD : continuous integration and continuous delivery

Setup :-
For remote Devops:
Create Outlook Account
Azure Devops Account (Where code will go)

For Local Development:
Visual Studio Code : IDE
And whatever plugins and software needed for development

For local development to remote repo:
Tortoise

Working:

1. Create project (Private/ Public) / Use Workspace of project:::-
   1) Azure Devops Project Management Course:

Create and Assign:
Project Divided into **release** and release is made up of **user story**/ backlog items and user story is completed by **task**.

EPIC(**Application**/ Multiple release) → Feature(**Model**/ Multiple Sprint) → UserStory(**Class**/ Sprint) → Task

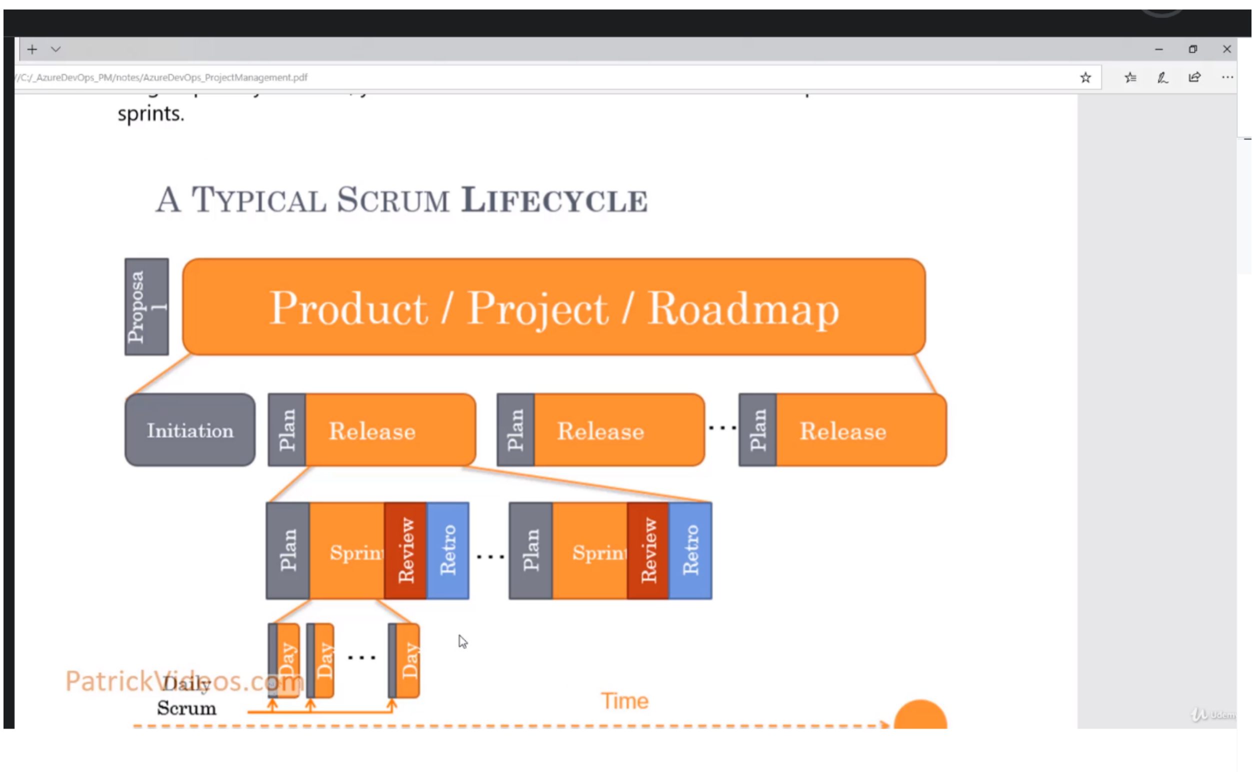
---

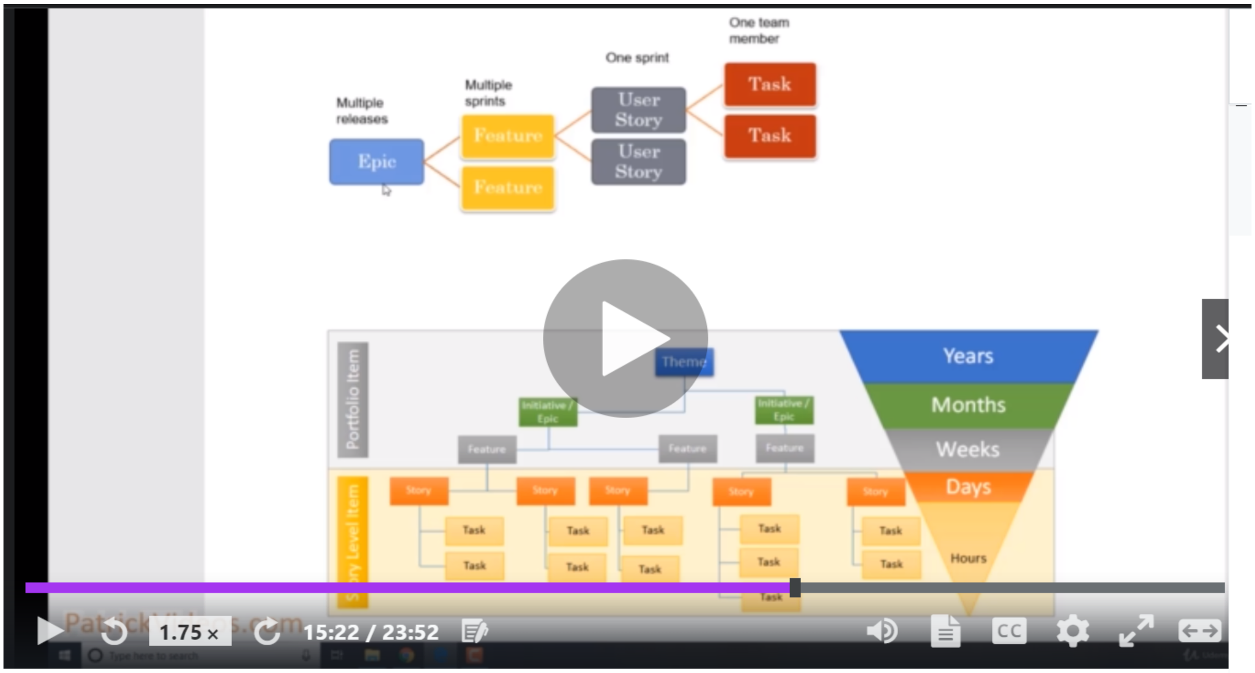
Work:
User story/backlog item → particular Branch → Release branch→ Pull request to the master branch

---

Overview:
Project → Release →Sprint → User story → tasks→ Sprints → Dashboard → Overview process(Completed. pending) → Assignee

---





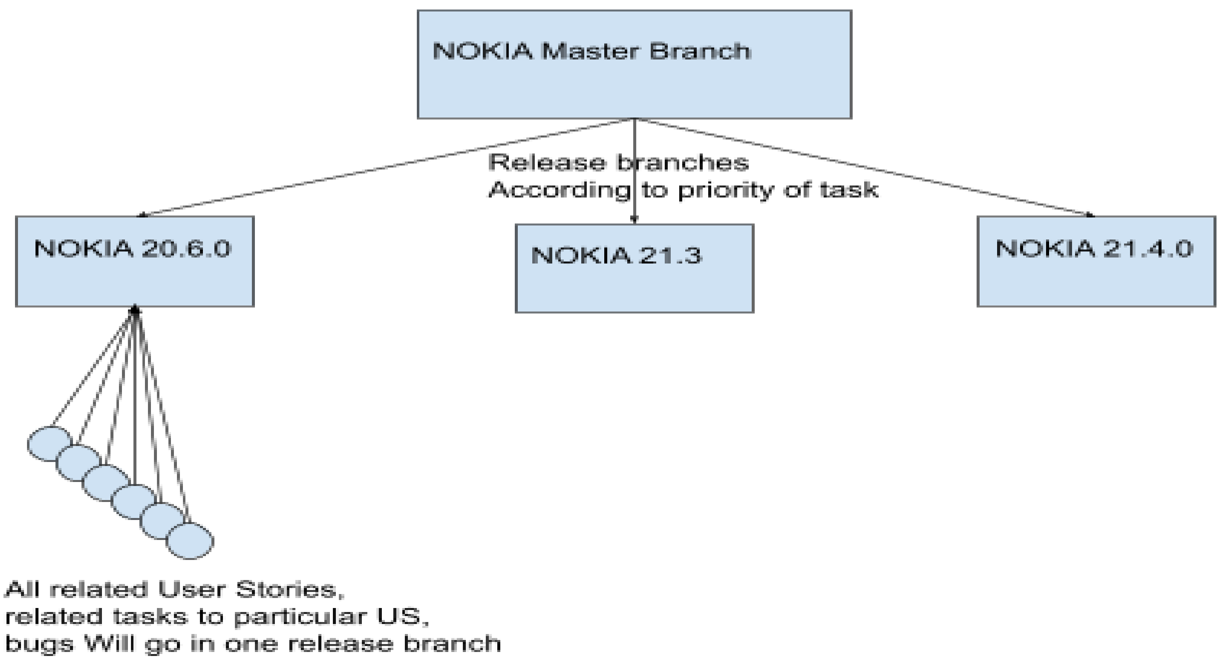
Some Terminology:
Format (Release, Branch, Tag)
Task: (Sprint wise)
Task related informations (What is task), priority (what is priority of that task),
Related attachment, related tasks, related discussion of the task

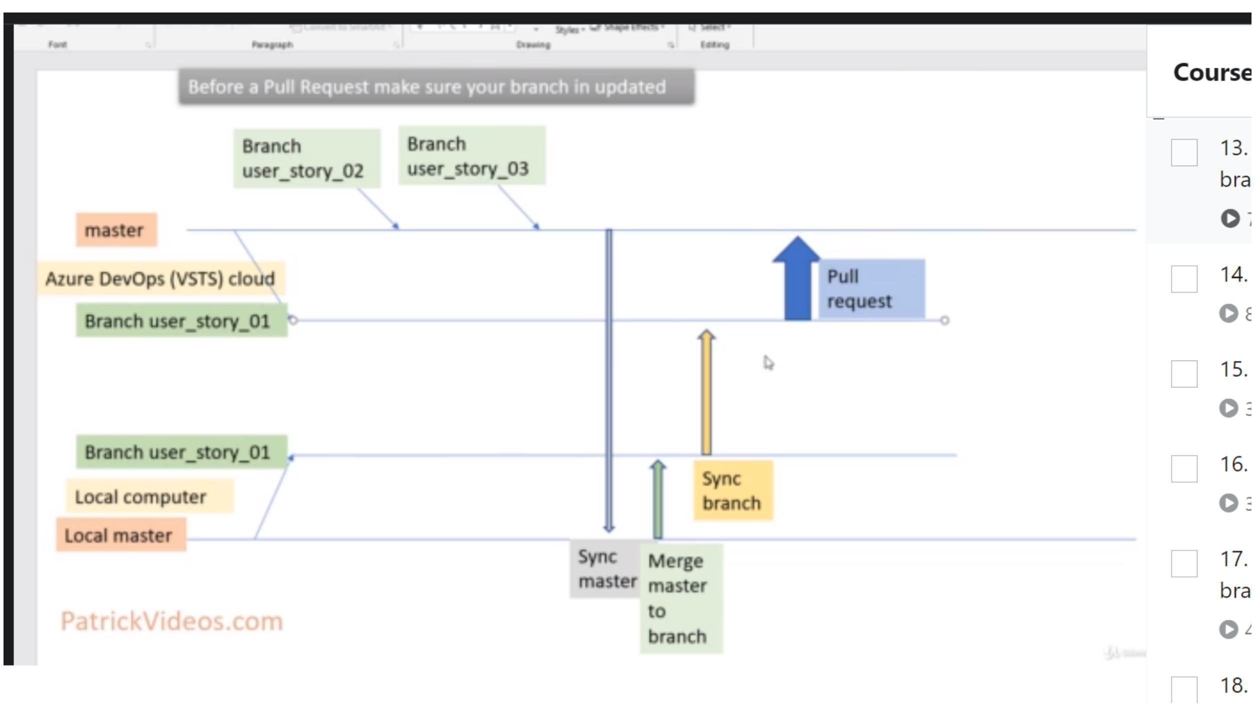
EPIC : a item which takes multiple releases to complete **(Application)**
(Use Acronym so multiple user story/ feature goes there)
Feature: a item which takes multiple sprint to complete **(Model)**
User Story: one sprint **(Class)**

Area: which item is related to particular folder
Iteration : **release**
Product backlog item : **user story**
Effort/ Business value

WorkItem: everything id wise
Board: UserStory Kanban chart
Backlog: UserStory (Epic tree structure)
Sprint: Task Kanban Chart, work capacity
Query, Chart, Dashboard, Email send, Excel, Notifications

2) Azure Devops Repo Course:





Note:

Never worked on master branch directly, Always create branch and then merge (Pull Request) to your branch with master branch.

Through Command Line Interface

----

git clone

----

git init
git add -A
git commit -m 'Added my project'

git remote add origin git@github.com:scotch-io/my-new-project.git
git remote -v => To check which remote url added

if u wanna go with master branch go with it:-
    git push -f origin branch_name
else create a branch then push back the code:-
    git branch branch_name
    git checkout branch_name
    Upstream branch creation github code
    git push -f origin branch_name

To ensure you have all update while you worked on locally.(if you are unable to add it)
git pull --rebase origin master
To pull data: git pull

Note: There is no need to worry because it does not delete anything and show the diff wherever the changes were done.
Some terminology:-
updated
stashed
staged and unstaged

==

Through Tortoise tool :

1) Clone the project / Pull the refresh copy of particular branch
2) Add your changes (Metadata downloaded through Salesforce organiser or workbench)
   Object Name and package.JSON
3) Create Branch and push the changes
4) Log version and monitor who did the changes on a particular file

==

Through Visual Studio Code:
Live server / In scratch org/ dev org/ production org
Page No: 93

on remote server/ repo
Git config and setup

Bottom Line for development use : visual studio, scratch org, dev org, production org
For deployment use Azure Devops/ Gitlab and tortoise
==
BASIC QUESTION ANSWERED
PULL, ADD, COMMIT, PUSH
Creating project on server / cloning the project : :: folder and file

Creating a new project locally/ Pushing the project ::: you need to create project and upstream branch on devops platform else where will u push the code

Pushing the code to main branch ::: we dont push the code directly to master branch

Pushing the code to local branch
instead we create a sub branch and then merge the code to master branch
(depending on the project length)
Branches are folder and files are the content of the project

Merging the branch
You can push to same folder or you can create a new folder and then push and merge

Managing the local version of code :: tortoise tool

Commit: Commit means added changes are finalized changes.
StageFile: staged files are those files which will go in the next commit.
To avoid Merge Conflict use : Use Sync and Push option (Pull latest changes and then go)

Pull (Fetch + Merge)
Revert (Will remove the changes and commit newly with branch)

Stash: Means tried out different version of code (Testing multiple piece of code which one work better)
----

3) Azure Devops Pipeline CI/CD Course:

So overview:
First : organization setting → create a template
Then project by using that template
Then team → Member → Permission
Notification, Subscription
Release → Sprint
After preliminary setup → Create EPIC → Feature → Userstory
Query → Chart → Dashboard

VS Code setup:

Theme
Font
Auto save
Auto format, format to prettier extension
live server

NOKIA CAPS SYSTEM

request (Send/ Call)

System 1 ---> System2

<---

response(Receive/Set)

Architecture: Salesforce → NAIP → CAPS → S3

Salesforce ← NAIP ← External System

Simulation tool: PostMan, SOAPUI

Basic API Functionality:

1) Get User
2) Get Security Group
3) Add/ remove users from the Security group based on NOKIA ID.

Few Basic coverage:

Profile: CH_Functional Account
User: For Running Dashboard and Report
Connected App: Provide the CAPS Required ID and Establish the Connection
Custom Setting: Security Group Profile Names for CAPS

Rest API:

1) **Get New Users: (Request is made from CAPS to Salesforce System)**
   Get the user Which has a valid emp id

1) End Point:
   /services/apexrest/GetNewUsers/

2) Request:
   /services/apexrest/GetNewUsers/**startdateTime**=2012-04-12T13:50:58.00Z/
   enddateTime=2019-04-16T10:50:58.000z

3) Response:
   1) Nokia Employee ID
   2) Profile Name
   3) Group Name
      [“NokiaEmployeeid”: “1212112”,
      “profileName”: “CH_Agent”,
      “groupNames”: [
      {“name”: “sec_alt”},
      {“name”: “sec_alt2”},
      {“name”: “sec_alt3”}]]

Retrieve Data Will be stored in Custom Setting : Security Group Profile Names for CAPS

2) Get Users From Groups:

Get all the Users from a Particular/ specific security Group.

1) End Point:
   /services/apexrest/GetUsersFromGroup

2) Request:
   /services/apexrest/GetUsersFromGroup?sfGroupId = 00G3C00001WesJ

3) Response:
   1) Nokia Employee Id
   2) Profile Name
   3) Salesforce User Id

   [{“NokiaEmployeeid”: “1212112”,
   “profileName”: “CH_Agent”,
   “sfUserId”: “0053C000001nSaNQAU”}]

3) Get Security Groups:

Get All the security groups

Note: HTTP Get Call to SF With necessary Authentication To get info

1) End Point:
   /services/apexrest/GetSFCHSecurityGroups

2) Request:

3) Response
   1) Group Id
   2) Group Name

   [{“groupSFName”:”SEC_VWireless”,
   “groupSFId”:”00G3C000001Wes9UAC”}]

4) Get User and Group Based on Nokia ID:

User’s Group Details when Quired by Nokia Emp id:

Note: Multiple Nokia Emp Id can be provided in a Single Call but overall request and Response size should be less than 6 MB.

1) End Points:
   /services/apexrest/GetUserAndGroupBasedOnNokiaId
2) Request:
   1) Nokia ID

   [{“nokiaEmployeeId”: “69144707”,
   “nokiaEmployeeId”: “00000012”}]

3) Response:
   a) Nokia Employee ID
   b) Profile Name
   c) Group Name
   d) Salesforce User ID

   [{“NokiaEmployeeid”: “1212112”,
   “profileName”: “CH_Agent”,
   “sfUserId”: “0053C000001nSaNQAU”,
   “groupNames”: [
   {“name”: “sec_alt”},
   {“name”: “sec_alt2”},
   {“name”: “sec_alt3”}]

5) Add or Remove Group Membership:
Adding or removing of a user from a group

1) End Point:
   /services/apexrest/AddRemoveGroupMembership
2) Request:
   1) Nokia Emp ID
   2) Group ID
   a) Operation

   [{“nokiaEmployeeId”: “69144707”,
   “SFGroupID”:”00G4100002222”,
   “Operation”: “add”}]

3) Response:

6) Get SFCH Security Group:

1) New Security Group Added
2) All Latest Configuration of Salesforce Security Group

UseCase:

1) **CAPS run a Batch in every 4hrs of interval to retrieve the new User Info (GetNewUsers API)**
2) **CAPS Will Analysis the Correct security group and assign the membership (AddRemoveGroupMembership)**
3) **After Operation CAPS Will send a Mail to DVT Team with Batch Summary (Number of Success/ Number of Errors)**
   Note: if Connectivity Issue CAPS Will send Email after 2 unsuccessful batch run
   Silent Retry Option is also there if retry fails.

Status Code:
URLS AND Component:
Upload:
1) CAPS_UplaodMultipleAttachment: Lightning Component
2) CAPS_ProgressBar: Lightning Component
3) CAPS_MultipleFileUploadController: Apex Class
4) CAPS_MultipleFileUploadWrapper: ApexClass

Download:
1) CH_DownloadAttachment: Lightning Component
2) CAPS_AttachmentController: ApexClass
3) CAPS_Attachments_Download: Apex Class
4) CAPS_Attachment_Util: Apex Class
5) CAPS_ConstantHandler: Apex Class (Constant Class)

All start with CAPS or S3 (Secure Storage System)
1) Remote Site Setting
2) CSP Site Setting
3) Custom Setting
4) Custom Label
5) Custom URL
6) Named Convention

CAPS URLS:
1) https://capsv.nokia.com/93f6cc8e/downloadattachment?id=attachment_id
2) https://capsv.nokia.com/93f6cc8e/uploadattachment
3) https://capsv.nokia.com/93f6cc8e/api/log/attachments
4) https://capsv.nokia.com/93f6cc8e/api/log/attachments/attachment_id
5) https://capsv.nokia.com/93f6cc8e/api/log/customers
6) https://capsv.nokia.com/93f6cc8e/api/log/products
7) https://capsv.nokia.com/93f6cc8e/api/log/nctclassifications
8) https://capsv.nokia.com/93f6cc8e/api/log/classifications

1) CH_ExternalInterfaceWebServices
2) CH_ExternalInterfaceUtil (Header Information)
3) CH_CaseTriggerHandler
4) CH_CaseTriggerHelper
5) CH_AttachmentsTriggerHandler
6) CAPS_AttachmentController
7) CAPS_AttachmentDelOnCloseCancelQueue
8) CAPS_AttachmentManualDeleteQueueable

EmailUpload:
Inbound Email Handler Email services to process email attachment
Classes:

1) CH_EmailAttachmentHandler
2) CH_EmailAttachmentHelper

Update Metadata to CAPS for NAIP Attachments:

1) CH_CaseTriggerHandler
2) CH_CaseTriggerHelper
3) CAPS_UpdateToCAPS4NAIPATTACHMENT

Share Attachment to R&D (Interface Type, pronto, JIRA):

1) CH_DownloadAttachment (lightning component)
2) CAPS_AttachmentController(apex class)

Send attachment to NAIP for S2S Case:

1) CAPS_UploadMultipleAttachment(lightning component)
2) CH_AttachmentsTriggerHandler (Apex Class)

Custom Setting:

SecurityGroupsProfileNameForCAPS
CHSecurityGroup
LogInterfaceLogOrNotForSecurityGroupAPIS

Custom MetaData:

CHPermissionSet2ProfileMapSecurityGroup

API CLASSES:

CAPS_GetNewUsersInDefinedSpanREST
CAPS_GetSFCHSecurityGroupsREST
CAPS_GetUsersInGroupREST
CAPS_SecurityGroupsRESTUtil
CAPS_GetUsersAndGroup4NokiaIdREST
CAPS_AddRemoveUsersToFromGroupsREST

Operation:

1) **Upload:** Only Case Owner and Team Member can Upload The file
   System admin till Only created.

2) **Download:**
   There are restriction also who can download the file based on account, region

3) **Delete:**
   Only Owner of file can Delete the file
   And to Delete a File Parameter Need to set That is **Retention Period**

NOTE:
Case Context: (S2S)
For Cancelled and Completed Case:
No Updation
Can Use Ref No and Case No to create new case

For Closed Case:
Updation is Allowed
Cant use ref. No to create new Case

USE CASE: Technically (What we need to keep in mind)
1) **End-Point URL (URL + Session ID + Authentication Id)**
2) **Request Header (Type: JSON / Excel) , Request Body (Wrapper, String)**
3) **Response (Parsing)**

UPLOAD: (POST, PUT HTTP METHOD USE)
Salesforce Configuration → CAPS Configuration(INFORMATION PASSING TO CAPS SYSTEM)
1) Customer and Product information
2) User Authorization information
3) Attachment Information
1) **Customer ID (Customer information, CDB ID)**
2) **File information (Name, Size (Byte))**
3) **Product (Product , Product Release, Product information, CDB ID)**
4) **NCT Classification (2,4)**
5) **Service_Class (CARE)**
6) **App Information (E.g: Service cloud)**
7) **Purpose**
8) **Description**
9) **Ticket(App, Ticket ID)**
10) **Security Group**
11) **Case Status**
12) **Customer Visible**
13) **Retention Period**

8 CALLOUTS from Salesforce to caps System:--

CAPS_UploadMultipleAttahcmentHelper.js : **Lightning Component** :**Set some input parameter**
Main JS File to Upload: CAPS_UploadMultipleAttachmentHelper.js
CAPS_MultiFileUplaodController **: Main Apex Class : initiate Upload Method**

Query case object to get necessary information
Note: CDB(Customer DataBase) Customer Id defined or get from CASE Object (person need to be authentic) (Customer Code with Product and Product Release should be mapped in CDB)
(CDB Information in Customer Setting)

Build initiate upload JSON with Customer, Product, product release & all other information)
CAPS_ConstantHanddler to set some standard constant
Note: Product Release is also an Object(Case. CH_Product_Release__r.CH_NSNLegaceyCode__c)

Attachment -> String -> Blob
Build Signature (Build Blob) (Custom Setting : CH_Setting: CAPS_HardCoded value)
(Sign and Secret Key for encoding, decoding)

Wrapper class to wrap up JSON (CAPS_MultipleFileUploadWrapper)
Send back Wrapper input to Lightning component
(CAPS_UploadMultipleAttahcmentHelper.js)

Http Callout Header Set (CAPS_UploadMultipleAttahcmentHelper.js)

Onload function to get Response of first Call out

Based on the response of first call out we perform second call out

1) **Parse Response and use that to perform subsequent call out**

First Call out : Initiation of Upload / Web CallOut

1) Salesforce Sends MetaData to CAPS System
2) CAPS Create Container of that Attachment
3) In response caps provide authorization to get token

Second Call out: Authentication : S3 Sign API: TimeStamp(Provide security token)

Third Call Out : Upload Id (with Token) from S3 System

Fourth Call Out: Content Upload Authentication
Fifth call out: Content upload to S3
Upload File in Chunk (Part Number and Etag Used to monitor)
Repeat the steps if file size is bigger

Method Name: uploadMultiPart

Send Part Number, Start Position, End position

E.g:

```
Var partNumber = 1;
Var startPositon = 0;
Var endPosition = Math.min(fileSelectedForUpload.File.Size, startPosition + self.chunkSize)
Var blob = fileSelectedForUpload.File.slice(startPosition, endPosition);
Var fileReader = new FileReader();
fileReader.onload = function(){
        self.uploadMultiPart(component, counter, s3uploadId, partNumber, startPostion,endPostion)
        }
```

6) CALLS to CAPS Sign (ETAG)
7) Part Number to Assemble the files
8) File assembling and file ready 2 use

Information is stored in CH_Attachment Object

Note that:

Why Status is Uploading even it is uploaded to CAPS:

Because CAPS is performing Encryption and Scrambling and till then it is Hardcoded to Uploading
Once it is done. Caps send status updation to ready and status will change.
Delete Original File only Scrambling and Encryption file will remain

Encryption and Scrambling is done on CAPS based on customer and product information

---

For Apex: Remote Site Setting

For Lightning Component : (Java Script)
Content Security Policy Trusted Sites
CORS Setting
Tool to debug: Google Chrome

Download: (CH_DownloadAttachment.cmp) (CAPS_AttachmentController)
CAPS System configure or taken care download stuff.
First call out to send info about user to determine user has access or not. Alias name of user pass
Second Call out

CH_DownloadAttachment.cmp (Lightning Component)
Note: Encryption key is the password

1) **END URL 2) Request Header(Method) 3) Request Body (String Data) 4) Response**

E.g:
```
Http hr = new Http();
httpRequest hr = new httpRequest();
hr.sendHeader();
Return hr;
hr.send();
httpResponse ht = new httpResponse();

//Calculate Signature
// Set Header
```

Or
```
    Http h = new Http();

    HttpRequest hr = new HttpRequest();
    hr.setEndpoint('https://postman-echo.com/get?foo1=bar1&foo2=bar2');
    hr.setMethod('GET');

    HttpResponse hr1 = new HttpResponse();
    hr1 =  h.send(hr);
    System.debug('body' + hr1.getBody());
```

Delete: (Just Update the retention days, rest all work done by caps) (Do Callout)
1) DeleteSelectedAttachments: Client Controller
2) c.updateRetenionDays: server Controller in CAPS_AttachmentController Class
3) CAPS_AttachmentDelete
4) updateRetenionDaysForLightning
5) CAPS_MaualDeleteQueuable
6) CAPS_Attachment_Util

Note: Suspected Data Breach (CheckBox)
User neither can close the case nor download the attachment
CAPS_Constant_Hnadler: Retention Days Setup
For Signature: UseUtilMethod

S2S Email Case: (Case Creation) (In email single shot we are uploading because we kept overall size is 11 MB) (And Multiple File Multiple time Queueable Class called)

Email Body Attachment: Should store in Salesforce
Email Attachment: Should Upload in CAPS

Setup-> Email Services (Forwarding Email Address / Routing)
Name: CH_EmailAttachmentHandler
Apex Class : CH_EmailAttachmentHandler
Using Thread Id, get Case Id and setup the process
(Feed Updating, Caps Uploading, File System UPloading) → CAPS_Mulltiple FileUpload Controller

NAIP (Through Interface to Upload) : (because NAIP handles Customer System)
Customer can also have flexibility to upload directly

After uploading, NAIP Sends Metadata to Salesforce in pure JSON Format
We need to update metadata for caps and create case creation in salesforce after that

Select id, caseNumber, ch_caps_AttachmentMetaDataFromNaip__c from Case where caseNumber ='';
Select id , attachmentId__c from ch_attachment__c where attachment_id=''

Select id, attachmentId__c, case__c, from ch_attachment__c where case__r.recordTypeId =''

CH_CaseTriggerHelper
(before update, after insert)

Bulkification : FOR Loop and If condition
chatterPost: Connect API

Security Group : Public Group : Sharing Setting Rules are Defined there (DVT team take care
Note: Whenever Security group gets created, in custom setting we need to add in manually
For all new user we have one enhancement custom meta data, if they have permission set we are sending a universal information. (CH_Basic_Case_Handling)
Because caps only do segregation based on profiles
CH_Setting__c.get();: get information from ch_setting

DPC(Data privacy controller)
CAPS ====> S3
DPC validate if users have access to the files based on some c\parameter

HWS CASE FLOW USE CASE ( ):

HWS_CaseFlow Component: (Screen / If Condition)

Contact ->> Account

1) **Product**
2) **Product Part Item/ Code**
3) **Product Items Code Version**

Address ->> Review, Submit

Submit ---> SOO (DownStream System)(Pega) (Will Process and deliver part)
(ordering / logistic/ monitor)
Searlizing the Information and in response we get unique SOO reference no
Pega is **Business Process Management**
Only case created/ case process on salesforce
Logistic/ product monitoring and tracking all other care by other system

---

CASE ID/ REST API.

Security Group: Apex/ Static rule
RMA Number: Custom Field/ Auto Number: ParentCase- AutoNumber

---

Portal:
All are diff-2 Component, which are communicating through events

---

Bulk Upload:
Multiple PartCode for a Product And for each part code for each shipping address information

---

Single Update for each child case (note in bulk because each status would be different)

In bulk Code when we do a great outrageous:
Same Country
Same Contract
Same Service Type
Child cases/ parent cases created by contract number

To Load the data there are Batch classes that run at 10AM (External System: falu, fnokia)
Product ---> Product Part Code(Alternate Product code) ---> Part Version
Version Items
External ⇒ staging
HWS_ProductSserviceSalesItemBatch → SI - subassembly - component - nsn1, nsn2, nsn3
HWS_ProductServiceableVersionItemBatch
HWS_Product_Relation_Batch
HWS_ProductVersionItem_Batch
HWS_ProductServicableAltCode_Batch
HWS_PMDSORComponentToProductBatch
NSN1, NSN2, NSN3- query - url - request - pmdqa - json data desirilizable Product 2
HWS_SubAssemblyTOProductBatch
Staging to Product :
HWS_PMDSOR_Import SubAssembly Product
HWS_EDP_SalesItemToProduct_Batch
HWS_EDP_SVIttOVersionItem_Batch
HWS_EDP_PVITtoVersionItem_Batch
HWS_EDP_Buildrelation_Product_Part

Account (Legal Entity) (One company need to register in diff, company, diff subsidiary)(each subsidiary have different logistic account (shipping address))
Contact (Authorized Contact (List of Authorized Contacts for that service contract)
(one contact can be part of multiple accounts but if its entitled to anything specific then in can not)
GWC: global welcoming center (Same as CH_Agent) (Hardware_Internal_Agent_Permission_Set)

Why Part code and Contract Number two option are there
Because some time they know part code and sometime they don't so by using contract number agent can select or create a case by using contract number

And While creating a case we are capturing child(All part is child) and parent case information
And for one case, U can add Multiple part but there would be one recipient address only
If u want to use different shipping address, u need to create case, or need to create it again
Address Calculation/ Delivery Address calculation
Business Hours/ Distance

Version code :(Part Revision) have version number if no version number then ANY also there.

Start Date // end date /// terminate date
CARE/ CBD / FALU/ FNokia/ SOO / CAPS / R&D
Ordering System / Case Service / Logistic System

External System to staging to salesforce
Stage Table: Different Different object are there to load in salesforce to mapping purpose
HWS_CaseFlow
HWS_Support_Ticket_MainFlow
--

String Matching Data Fetcher Project :

Application:

```
 <aura:application >
   <c:StringFinder />
</aura:application>
```

Component:

```
<aura:component controller= "StringMatcher" >
   <h1> Data Finder </h1>
   <lightning:formattedText value="Enter your Desire input that will search on Account Object" />
   <aura:attribute type="String" name="query" />
   <lightning:input type = "text" value="{!v.query}" name="query" onkeyup="{!c.stringFinder}" />
   <br/>
   <aura:attribute type="Account" name="list"  />
   <aura:iteration items="{!v.list}" var="listData" >
      <p>{!listData.Name}</p> <br/>
   </aura:iteration>
   <br/>
</aura:component>
```

Note:

Onkeyup is used to fire the action method as soon as the key type.
Attribute type account because using Account List to fetch
U can also used Object as standard Non Primitive list

E.g: `<aura:attribute type="Object" name="list"  />`

Client side controller:

```
   stringFinder: function(component, event, helper) {
      //alert("hi");
      var a = component.get("v.query");
      //console.log(a);
                  var action = component.get("c.accountStringFinder");
      action.setParams({
         searchKey : a

      });
      action.setCallback(this, function(response){
         console.log(JSON.stringify(response.getReturnValue() , null, 4));
         component.set("v.list", response.getReturnValue());
      });
       $A.enqueueAction(action);
   }})
```

Note:
To get server side controller

```
var action = component.get("c.accountStringFinder");
```

To set parameter for passing in server controller

```
    action.setParams({
        searchKey : a
    });
```

To get response the value

```
    action.setCallback(this, function(response){
        console.log(JSON.stringify(response.getReturnValue() , null, 4));
        component.set("v.list", response.getReturnValue());
    });
```

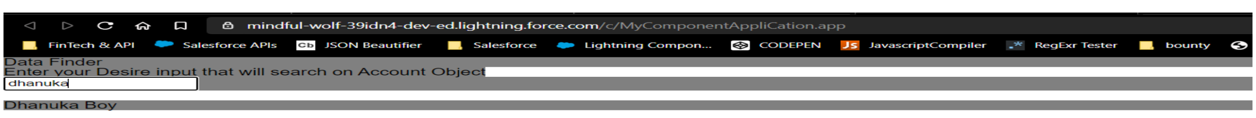
To enqueue all the operation ::: it work on FIFO (First operation first)
If u try to call back first without setting param it will throw an error.

```
     $A.enqueueAction(action);
```

Server Side controller:

```
public class StringMatcher {
  @auraEnabled
  public static List<Account> accountStringFinder(String searchKey){
    String name = '%'+searchKey +'%';
    return [select id, Name from Account where Name Like :name];
  }

}
```

Passing the method parameter through server side controller and getting the response (return)



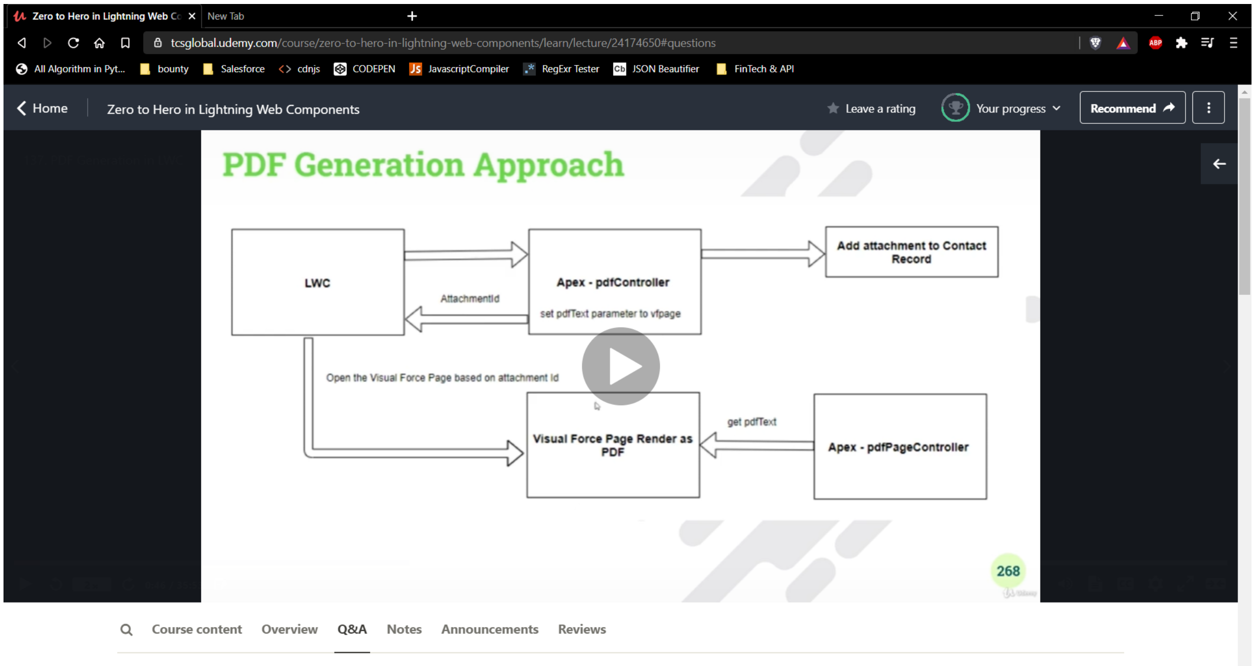
Project:

All Source code are uploaded in github

PDF generation:

PDF generation in LWC: (Design format and data pass in string encapsualted format and use standard functionality)

Architecture:




```
generatePDF({ recordId: this.recordId, htmlData: content.outerHTML
})
```

```
 Pagereference page = Page.renderAsPDF;
       page.getParameters().put('pdfText', htmlData);
```

```
    public String pdfText{get;set;}
    public pdfPageController() {
        pdfText = String.escapeSingleQuotes(
            ApexPages.currentPage().getParameters().get('pdfText')
        );
    }
```

```
<apex:page controller="pdfPageController" renderAs="pdf"
applyHtmlTag="false" showHeader="false" cache="true"
readOnly="true">
    <html>
        <head>
            <meta http-equiv="Content-Type"
content="text/html;charset=UTF-8"/>
        </head>
        <apex:outputText value="{!pdfText}" escape="false"/>
    </html>

</apex:page>
```

pdfPageController

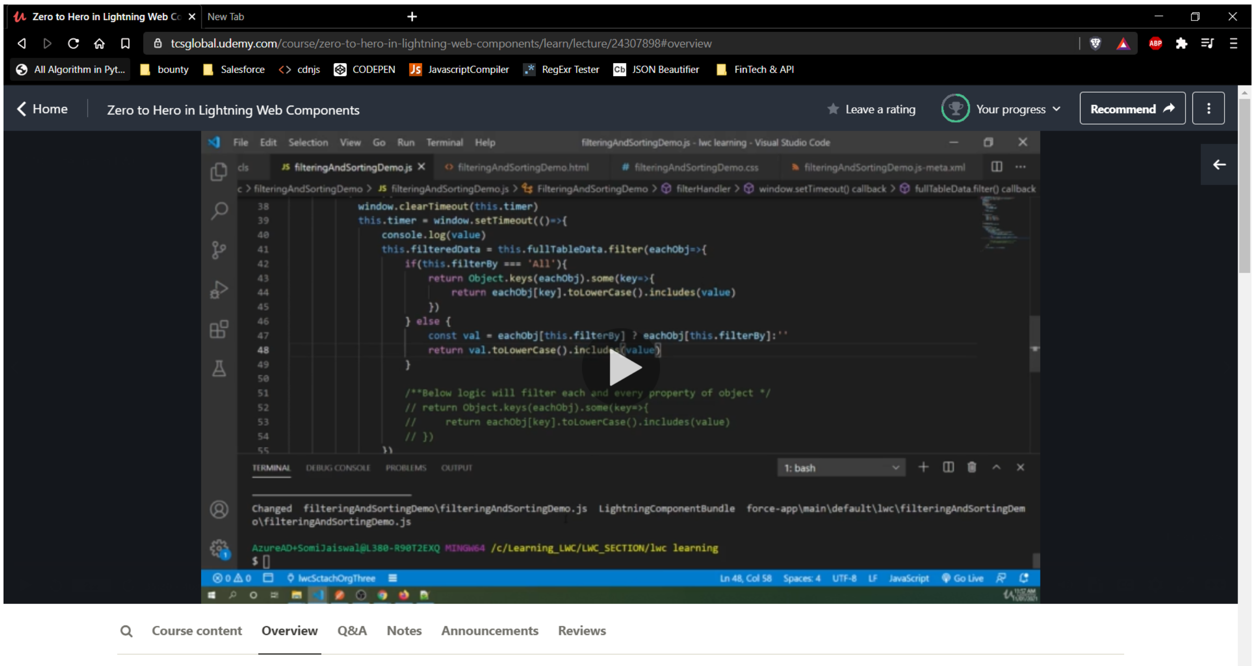
```
public with sharing class pdfPageController {
    public String pdfText{get;set;}
    public pdfPageController() {
        pdfText = String.escapeSingleQuotes(
            ApexPages.currentPage().getParameters().get('pdfText')
        );
    }
}
```

Putting pdf in attachment section :-

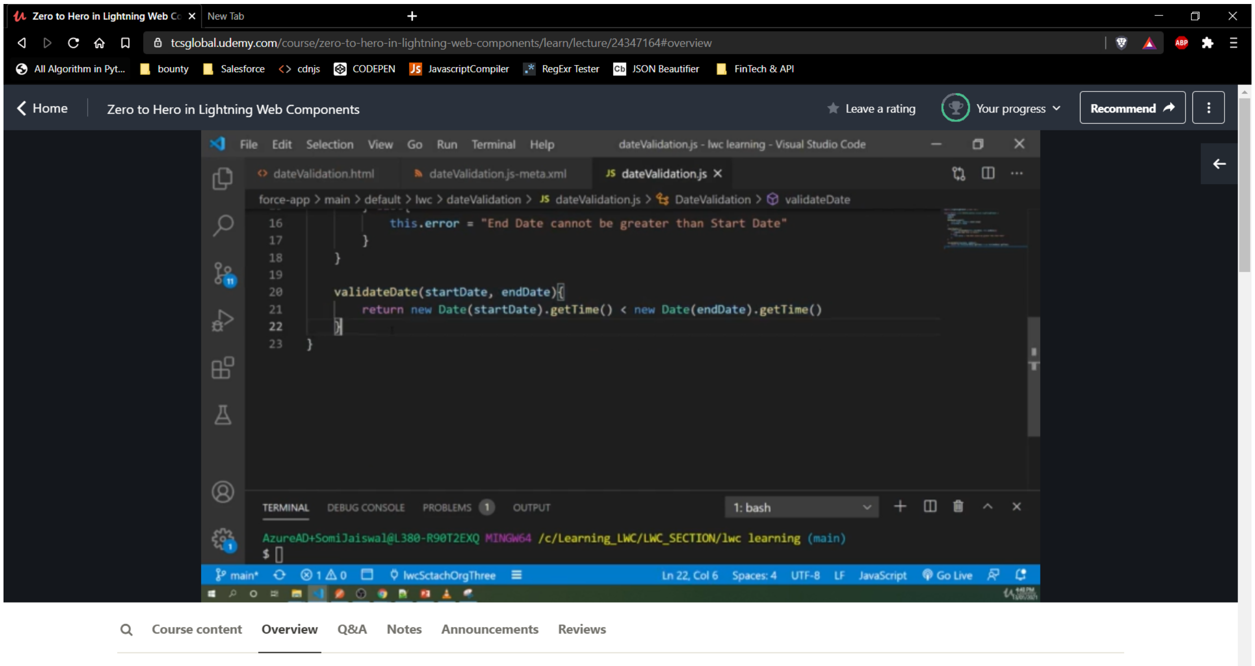
pdfController

```
public with sharing class pdfController {
    @AuraEnabled
    public static Attachment generatePDF(Id recordId, String htmlData){
        Pagereference page = Page.renderAsPDF;
        page.getParameters().put('pdfText', htmlData);

        Contact con = new Contact(Id=recordId);
        Attachment objAttachment = new Attachment();
        objAttachment.Name='invoice.pdf';
        objAttachment.ParentId = con.Id;
        objAttachment.Body = page.getContentaspdf();
        objAttachment.IsPrivate = false;
        insert objAttachment;
        return objAttachment;
    }
}
```

Pdf: standard web
Chart in LWC: library, live data
Map : standard api function
Searching (click, onekeyup, onkeyup with time interval)
Filtering: (with one value, with group of value(All) in column)
Sorting (in Asc, Desc order(number, string))



Date: standard tag
Validation in between start date and end date



Generate CSV in LWC::
Data Checking, creating to string and exporting

Excel Generation:- **Excel sheet (Importing and exporting)**

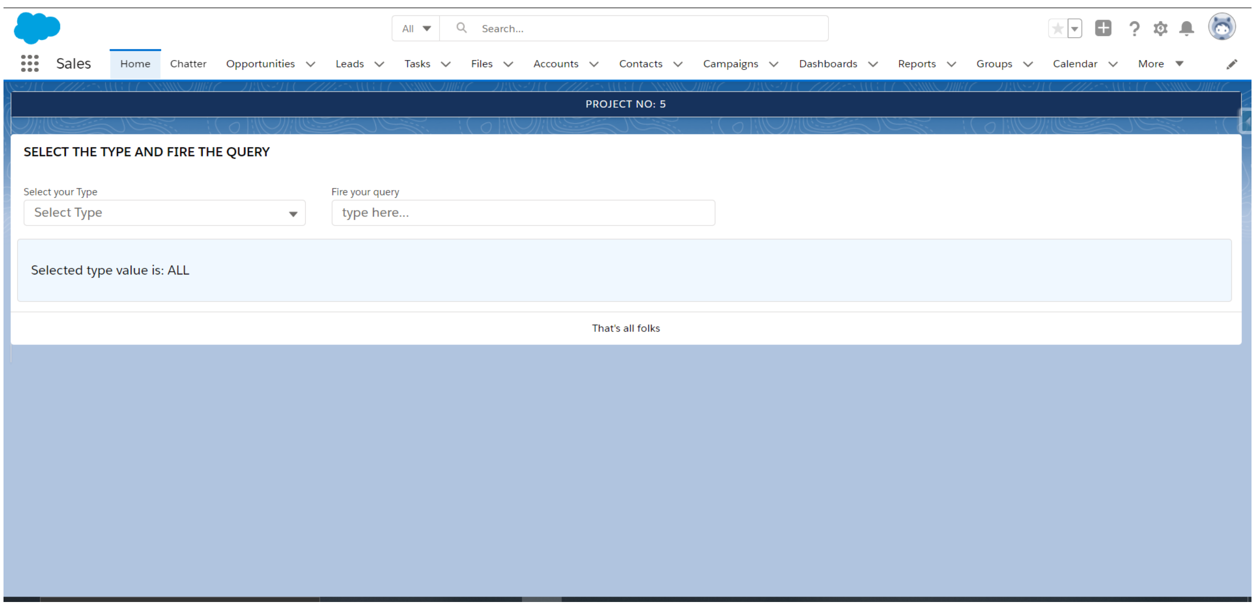
Import and Exporting of excel and parsing of data
"Excel Data to chain of Block (Block Formation) "
"Excel Live Data to chain of Block
Consensys (Synchronisation help here)"
Excel formation/ Data Exporting/ Data Importing

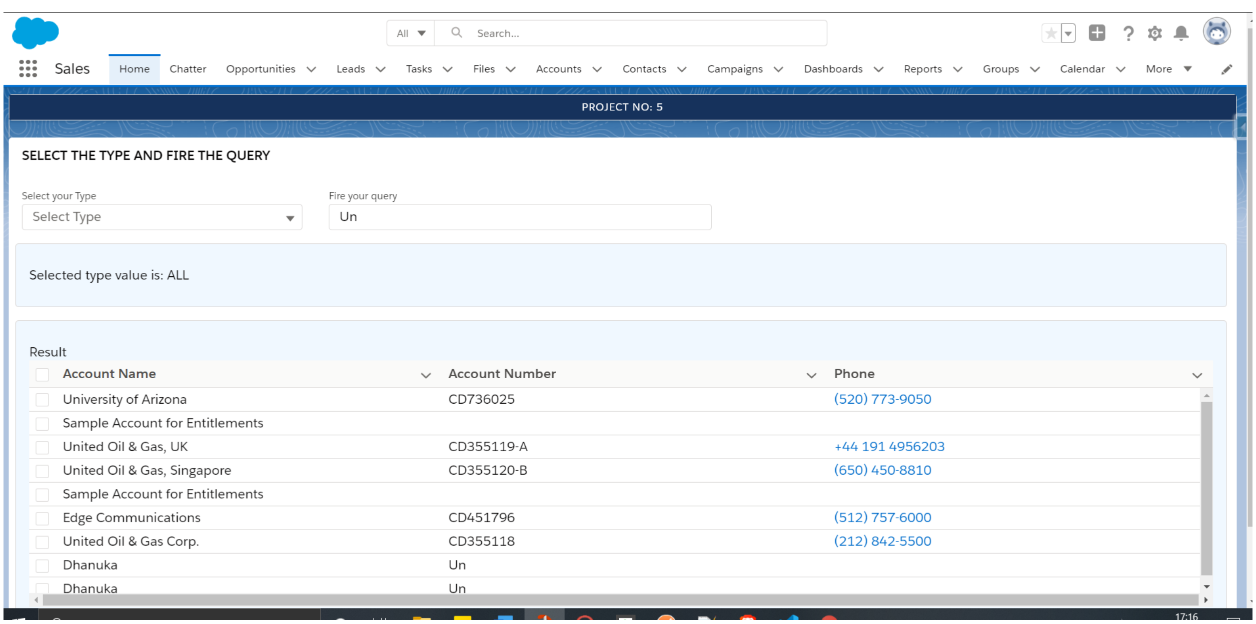
Cuorsel:
Custom Carousel : slider of images

Project 5:

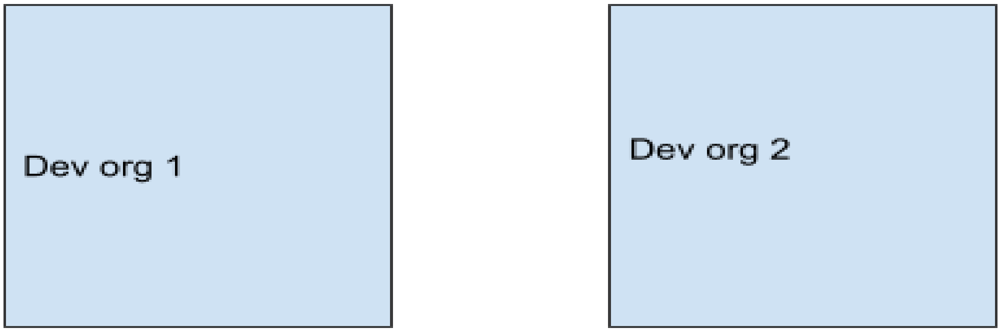
Pagination (token)

String Searching Functionality
After searching result
Show in a page
Do the pagination



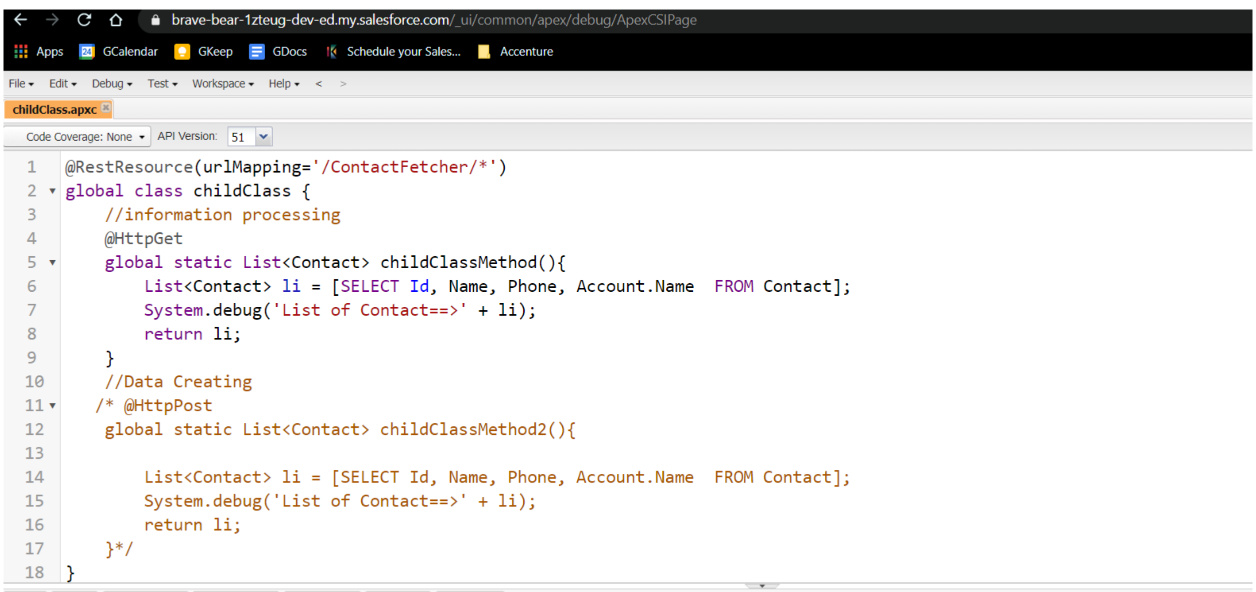


Project 4:

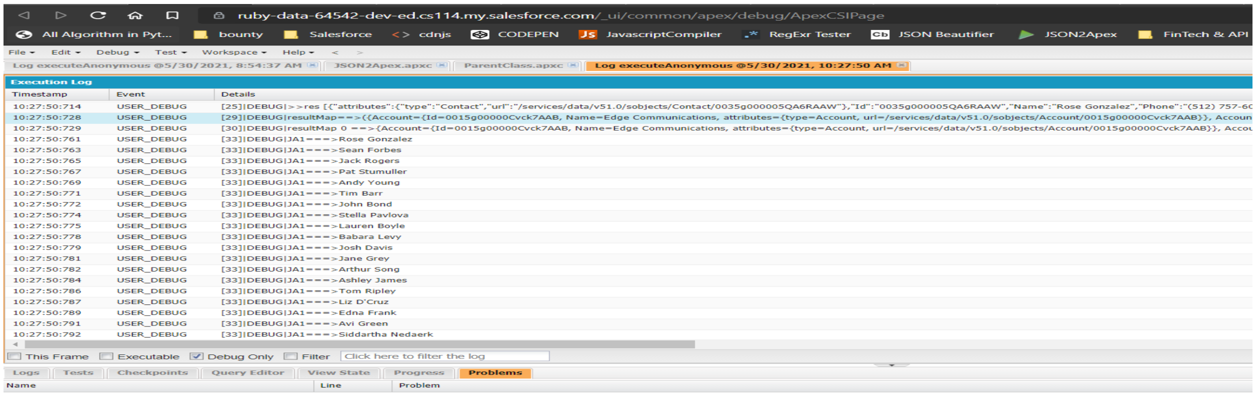


Call one Class method to another org

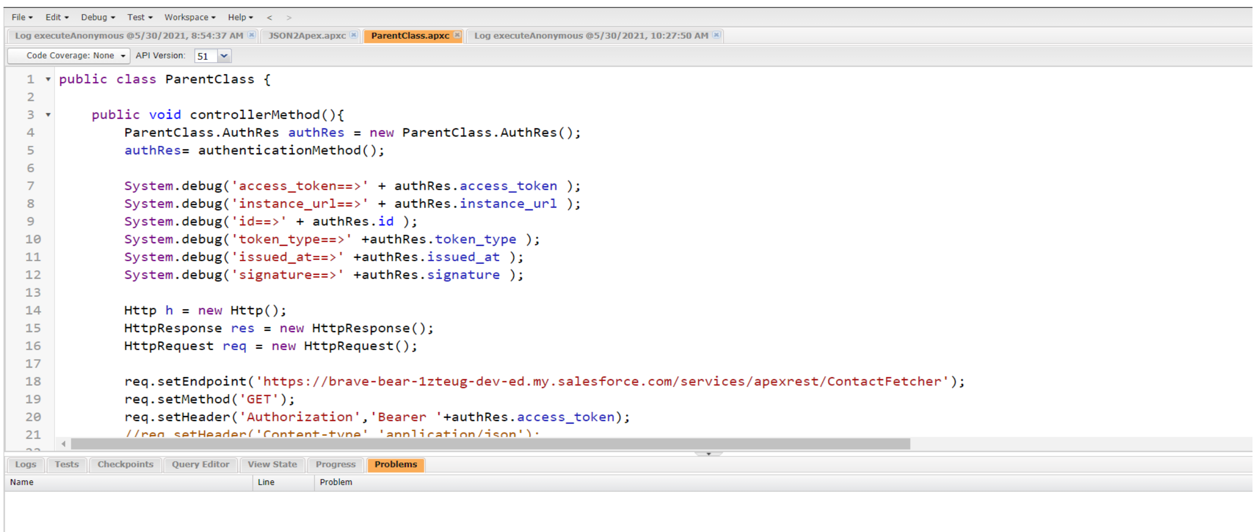
Fetch fucntion usage
To API call



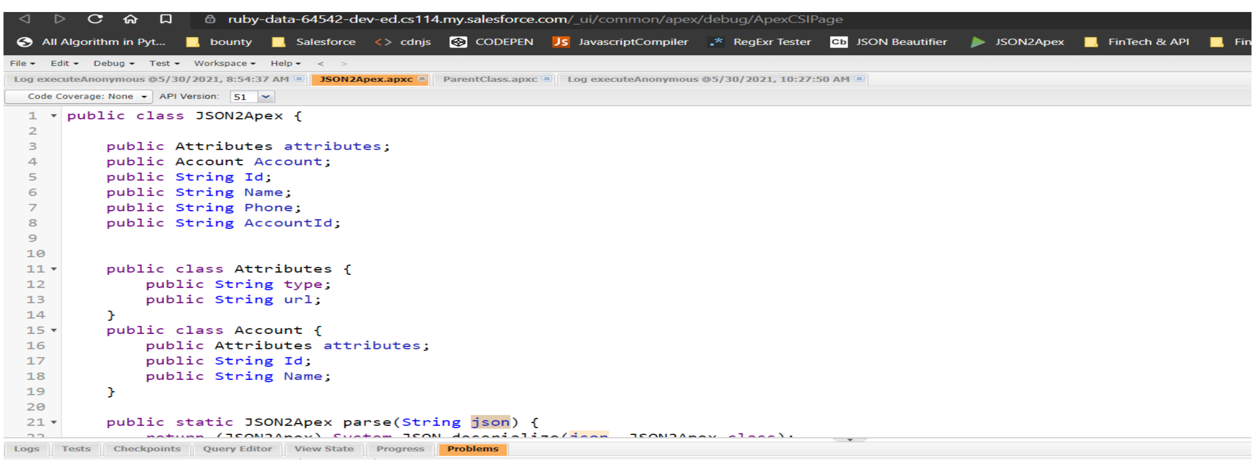

```
@RestResource(urlMapping='/ContactFetcher/*')
global class childClass {
    //information processing
    @HttpGet
    global static List<Contact> childClassMethod(){
        List<Contact> li = [SELECT Id, Name, Phone, Account.Name  FROM Contact];
        System.debug('List of Contact==>' + li);
        return li;
    }
    //Data Creating
   /* @HttpPost
    global static List<Contact> childClassMethod2(){

        List<Contact> li = [SELECT Id, Name, Phone, Account.Name  FROM Contact];
        System.debug('List of Contact==>' + li);
        return li;
    }*/
}
```



ruby-data-64542-dev-ed.cs114.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
All Algorithm in Pyt...
bounty
Salesforce
cdnjs
CODEPEN
JavascriptCompiler
RegExr Tester
JSON Beautifier
JSON2Apex
FinTech & API
File
Edit
Debug
Test
Workspace
Help
Log executeAnonymous @5/30/2021, 8:54:37 AM
JSON2Apex.apxc
ParentClass.apxc
Log executeAnonymous @5/30/2021, 10:27:50 AM
Execution Log
Timestamp | Event | Details
10:27:50:714 | USER_DEBUG | [25]|DEBUG|>>res [{"attributes":{"type":"Contact","url":"/services/data/v51.0/sobjects/Contact/0035g000005QA6RAAW"},"Id":"0035g000005QA6RAAW","Name":"Rose Gonzalez","Phone":"(512) 757-6
10:27:50:728 | USER_DEBUG | [29]|DEBUG|resultMap==>({Account={Id=0015g00000Cvck7AAB, Name=Edge Communications, attributes={type=Account, url=/services/data/v51.0/sobjects/Account/0015g00000Cvck7AAB}}, Accoun
10:27:50:729 | USER_DEBUG | [30]|DEBUG|resultMap 0 ==>{Account={Id=0015g00000Cvck7AAB, Name=Edge Communications, attributes={type=Account, url=/services/data/v51.0/sobjects/Account/0015g00000Cvck7AAB}}, Accou
10:27:50:761 | USER_DEBUG | [33]|DEBUG|JA1===>Rose Gonzalez
10:27:50:763 | USER_DEBUG | [33]|DEBUG|JA1===>Sean Forbes
10:27:50:765 | USER_DEBUG | [33]|DEBUG|JA1===>Jack Rogers
10:27:50:767 | USER_DEBUG | [33]|DEBUG|JA1===>Pat Stumuller
10:27:50:769 | USER_DEBUG | [33]|DEBUG|JA1===>Andy Young
10:27:50:771 | USER_DEBUG | [33]|DEBUG|JA1===>Tim Barr
10:27:50:772 | USER_DEBUG | [33]|DEBUG|JA1===>John Bond
10:27:50:774 | USER_DEBUG | [33]|DEBUG|JA1===>Stella Pavlova
10:27:50:775 | USER_DEBUG | [33]|DEBUG|JA1===>Lauren Boyle
10:27:50:778 | USER_DEBUG | [33]|DEBUG|JA1===>Babara Levy
10:27:50:779 | USER_DEBUG | [33]|DEBUG|JA1===>Josh Davis
10:27:50:781 | USER_DEBUG | [33]|DEBUG|JA1===>Jane Grey
10:27:50:782 | USER_DEBUG | [33]|DEBUG|JA1===>Arthur Song
10:27:50:784 | USER_DEBUG | [33]|DEBUG|JA1===>Ashley James
10:27:50:786 | USER_DEBUG | [33]|DEBUG|JA1===>Tom Ripley
10:27:50:787 | USER_DEBUG | [33]|DEBUG|JA1===>Liz D'Cruz
10:27:50:789 | USER_DEBUG | [33]|DEBUG|JA1===>Edna Frank
10:27:50:791 | USER_DEBUG | [33]|DEBUG|JA1===>Avi Green
10:27:50:792 | USER_DEBUG | [33]|DEBUG|JA1===>Siddartha Nedaerk
This Frame
Executable
Debug Only
Filter
Click here to filter the log
Logs
Tests
Checkpoints
Query Editor
View State
Progress
Problems
Name
Line
Problem

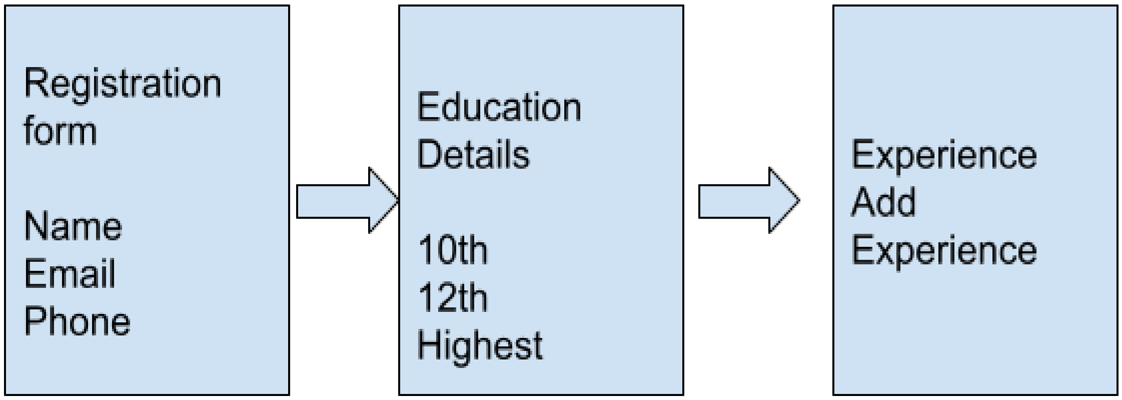


File
Edit
Debug
Test
Workspace
Help
Log executeAnonymous @5/30/2021, 8:54:37 AM
JSON2Apex.apxc
ParentClass.apxc
Log executeAnonymous @5/30/2021, 10:27:50 AM
Code Coverage: None
API Version: 51
public class ParentClass {
public void controllerMethod(){
ParentClass.AuthRes authRes = new ParentClass.AuthRes();
authRes= authenticationMethod();
System.debug('access_token==>' + authRes.access_token);
System.debug('instance_url==>' + authRes.instance_url);
System.debug('id==>' + authRes.id);
System.debug('token_type==>' +authRes.token_type);
System.debug('issued_at==>' +authRes.issued_at);
System.debug('signature==>' +authRes.signature);
Http h = new Http();
HttpResponse res = new HttpResponse();
HttpRequest req = new HttpRequest();
req.setEndpoint('https://brave-bear-1zteug-dev-ed.my.salesforce.com/services/apexrest/ContactFetcher');
req.setMethod('GET');
req.setHeader('Authorization','Bearer '+authRes.access_token);
Logs
Tests
Checkpoints
Query Editor
View State
Progress
Problems
Name
Line
Problem



ruby-data-64542-dev-ed.cs114.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
All Algorithm in Pyt...
bounty
Salesforce
cdnjs
CODEPEN
JavascriptCompiler
RegExr Tester
JSON Beautifier
JSON2Apex
FinTech & API
File
Edit
Debug
Test
Workspace
Help
Log executeAnonymous @5/30/2021, 8:54:37 AM
JSON2Apex.apxc
ParentClass.apxc
Log executeAnonymous @5/30/2021, 10:27:50 AM
Code Coverage: None
API Version: 51
public class JSON2Apex {
public Attributes attributes;
public Account Account;
public String Id;
public String Name;
public String Phone;
public String AccountId;
public class Attributes {
public String type;
public String url;
}
public class Account {
public Attributes attributes;
public String Id;
public String Name;
}
public static JSON2Apex parse(String json) {
Logs
Tests
Checkpoints
Query Editor
View State
Progress
Problems
Name
Line
Problem

Project 3:



Save All details from previous screen

And after final saving trigger the confirmation email

Event
Dynamic binding
Server Side Submitting
Email Alert

First screen submit take response id and pass it

Logic: After submitting first screen data, get the id and pass that id in subsequent screen

----

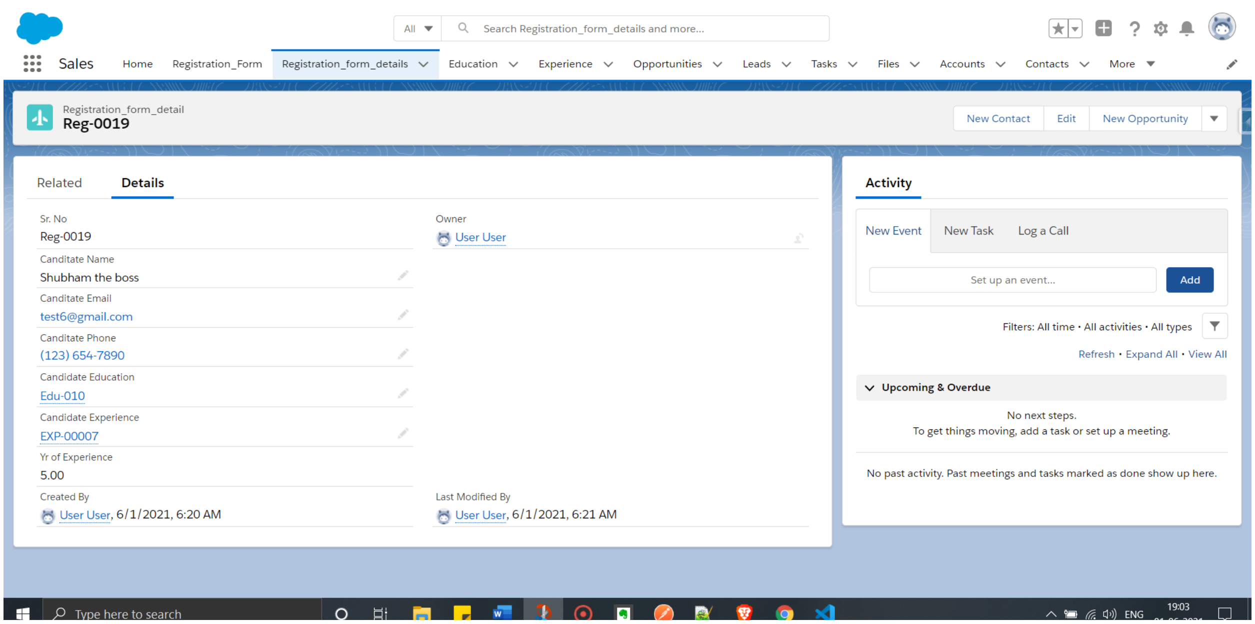
Page Layout, Record Type, Profile entitlement,
Workflow (record_type)field update,
trigger(lookup field value update) , flow(auto create record)
Communication is completed (through record based trigger flow)

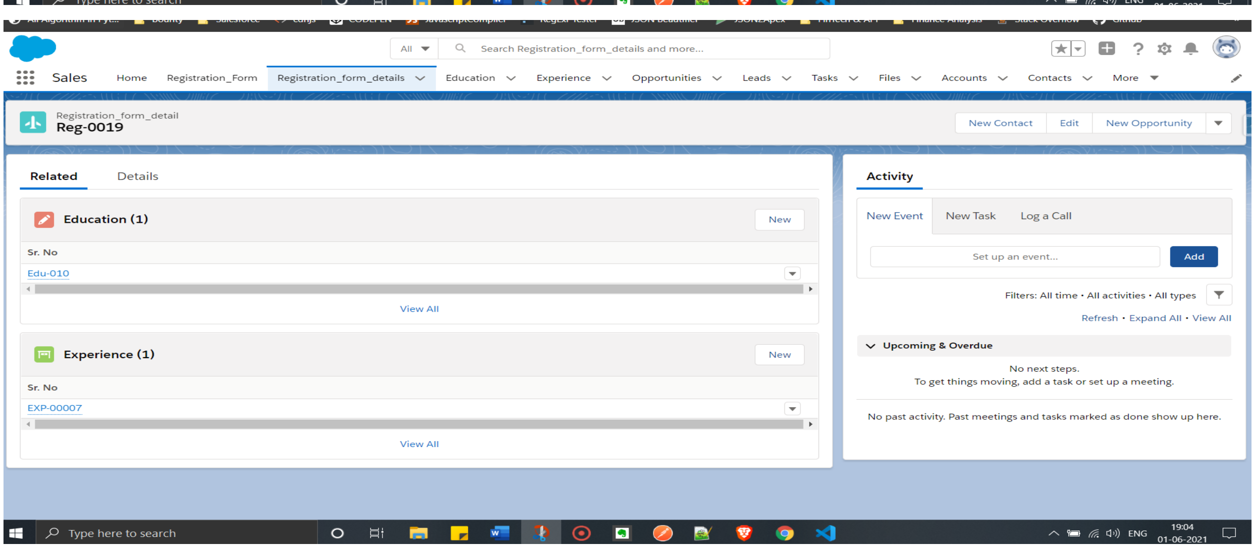
--->

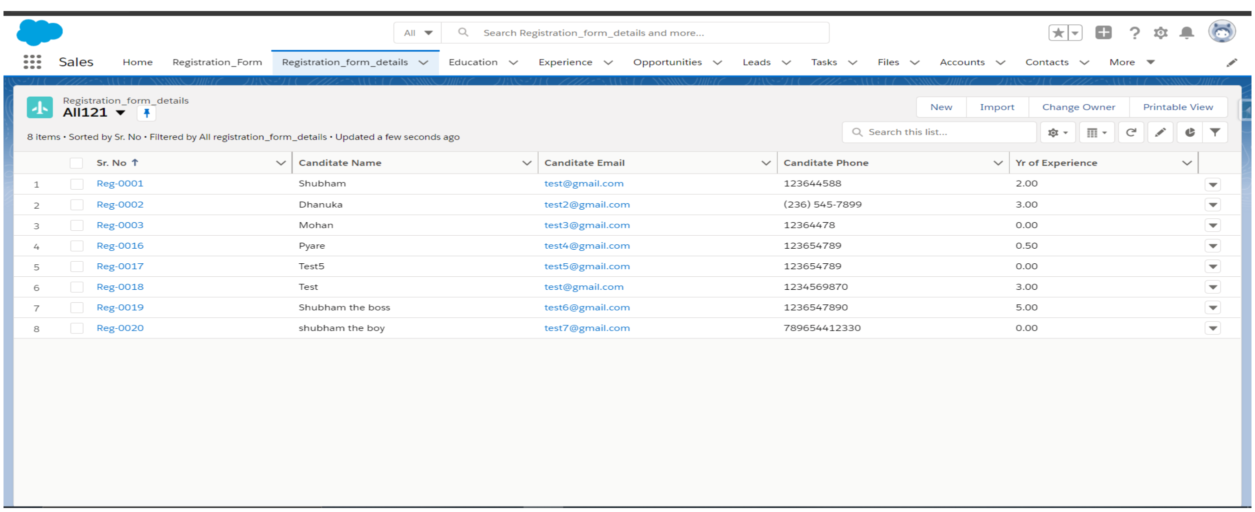
Communication is completed(field update) (through trigger)

←--

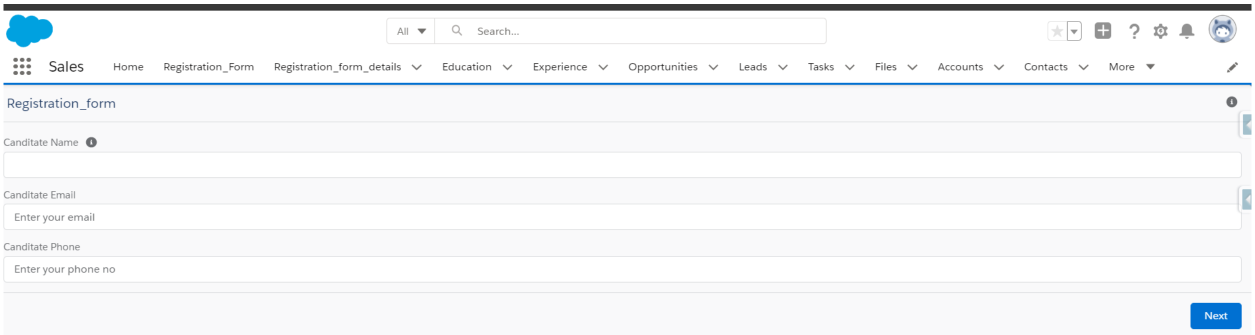
Database architecture set:

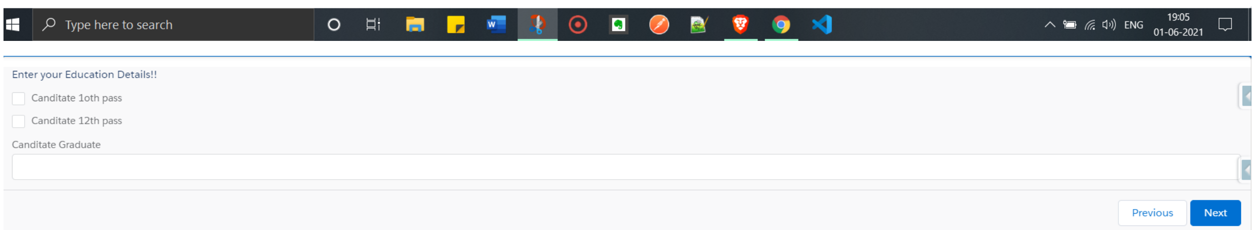

Search Registration_form_details and more...
Sales
Home
Registration_Form
Registration_form_details
Education
Experience
Opportunities
Leads
Tasks
Files
Accounts
Contacts
More
Registration_form_detail
Reg-0019
New Contact
Edit
New Opportunity
Related
Details
Sr. No
Reg-0019
Owner
User User
Canditate Name
Shubham the boss
Canditate Email
test6@gmail.com
Canditate Phone
(123) 654-7890
Candidate Education
Edu-010
Candidate Experience
EXP-00007
Yr of Experience
5.00
Created By
User User, 6/1/2021, 6:20 AM
Last Modified By
User User, 6/1/2021, 6:21 AM
Activity
New Event
New Task
Log a Call
Set up an event...
Add
Filters: All time • All activities • All types
Refresh • Expand All • View All
Upcoming & Overdue
No next steps.
To get things moving, add a task or set up a meeting.
No past activity. Past meetings and tasks marked as done show up here.
Type here to search
ENG
19:03

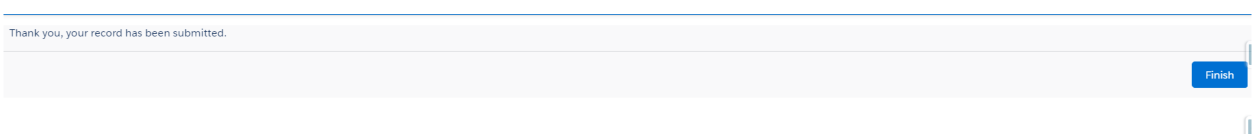

Search Registration_form_details and more...
Sales
Home
Registration_Form
Registration_form_details
Education
Experience
Opportunities
Leads
Tasks
Files
Accounts
Contacts
More
Registration_form_detail
Reg-0019
New Contact
Edit
New Opportunity
Related
Details
Education (1)
New
Sr. No
Edu-010
View All
Experience (1)
New
Sr. No
EXP-00007
View All
Activity
New Event
New Task
Log a Call
Set up an event...
Add
Filters: All time • All activities • All types
Refresh • Expand All • View All
Upcoming & Overdue
No next steps.
To get things moving, add a task or set up a meeting.
No past activity. Past meetings and tasks marked as done show up here.
Type here to search
ENG
19:04
01-06-2021


Search Registration_form_details and more...
Sales
Home
Registration_Form
Registration_form_details
Education
Experience
Opportunities
Leads
Tasks
Files
Accounts
Contacts
More
Registration_form_details
All121
New
Import
Change Owner
Printable View
8 items • Sorted by Sr. No • Filtered by All registration_form_details • Updated a few seconds ago
Search this list...
Sr. No
Canditate Name
Canditate Email
Canditate Phone
Yr of Experience
1 Reg-0001 Shubham test@gmail.com 123644588 2.00
2 Reg-0002 Dhanuka test2@gmail.com (236) 545-7899 3.00
3 Reg-0003 Mohan test3@gmail.com 12364478 0.00
4 Reg-0016 Pyare test4@gmail.com 123654789 0.50
5 Reg-0017 Test5 test5@gmail.com 123654789 0.00
6 Reg-0018 Test test@gmail.com 1234569870 3.00
7 Reg-0019 Shubham the boss test6@gmail.com 1236547890 5.00
8 Reg-0020 shubham the boy test7@gmail.com 789654412330 0.00

Done with Flow







With LWC:

Registration data will go to server

Registration Form ----------------------------------> Database server

←----------------------------------

In response, will get record id

Id will create a event

Clicking on Next button on first screen:

1) Submitting the data
2) And in response creating id and move to next screen

Will pass that registration form id and education data

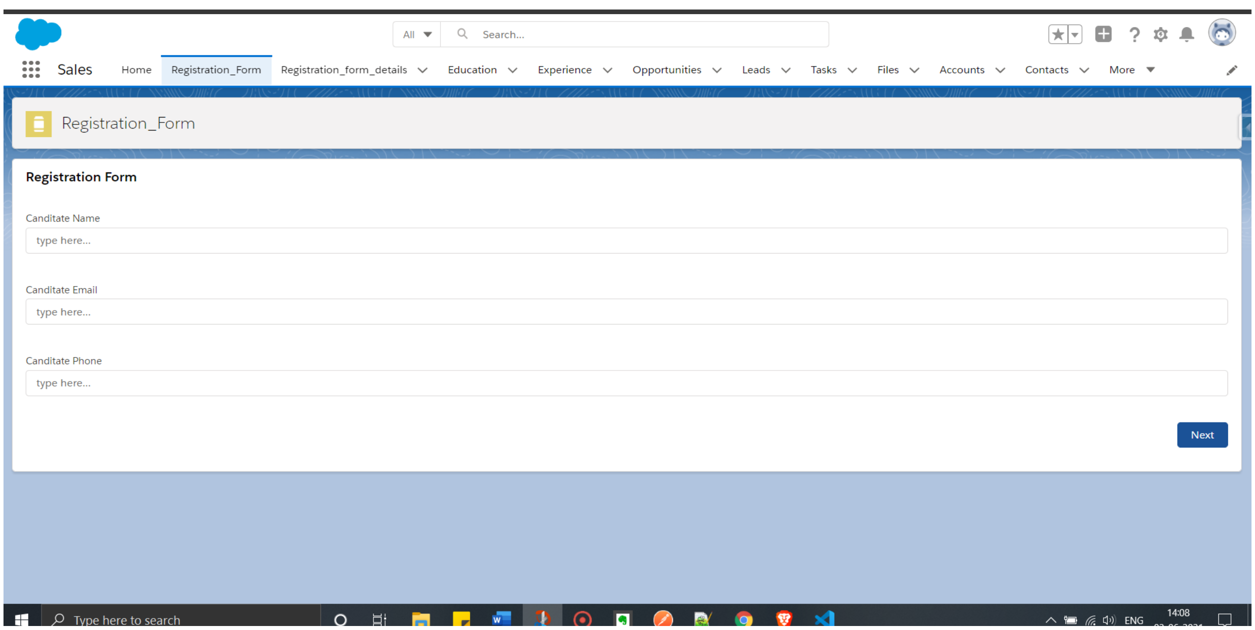
--->

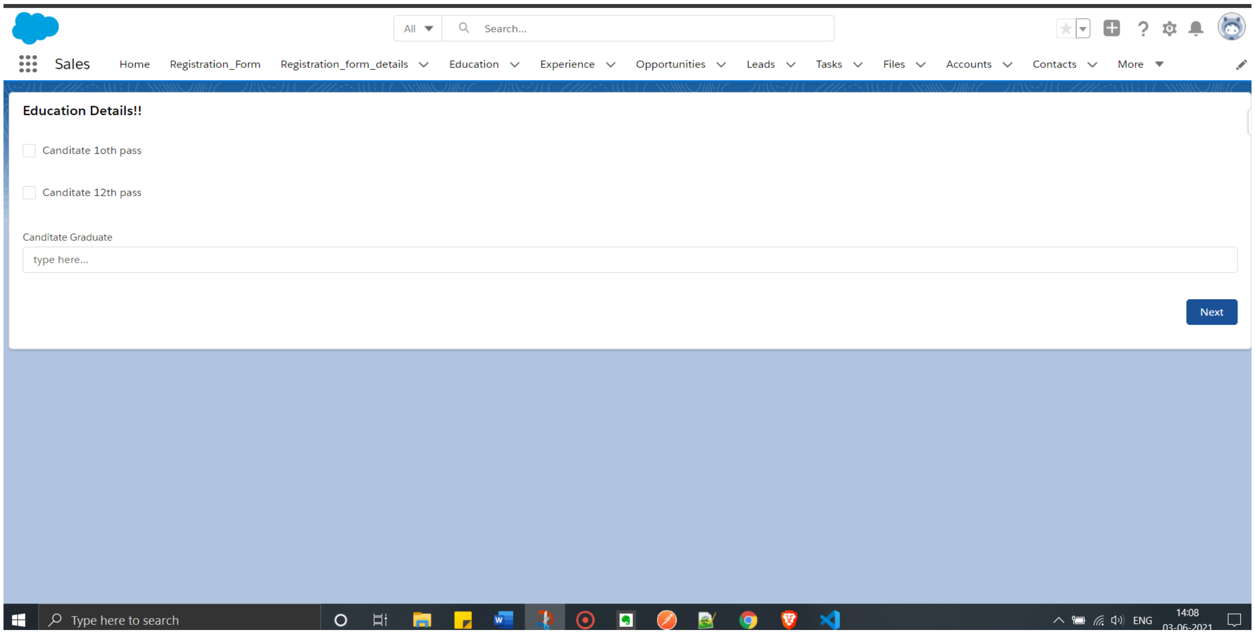
Education form   Database server

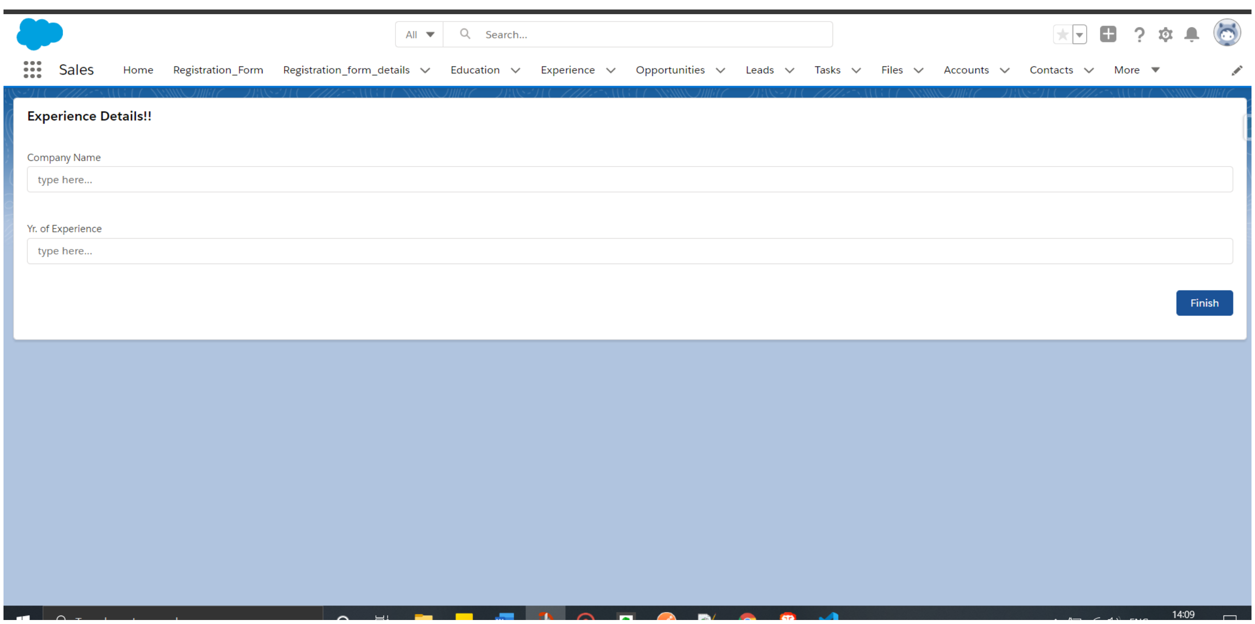
Will pass that registration form id and experience data

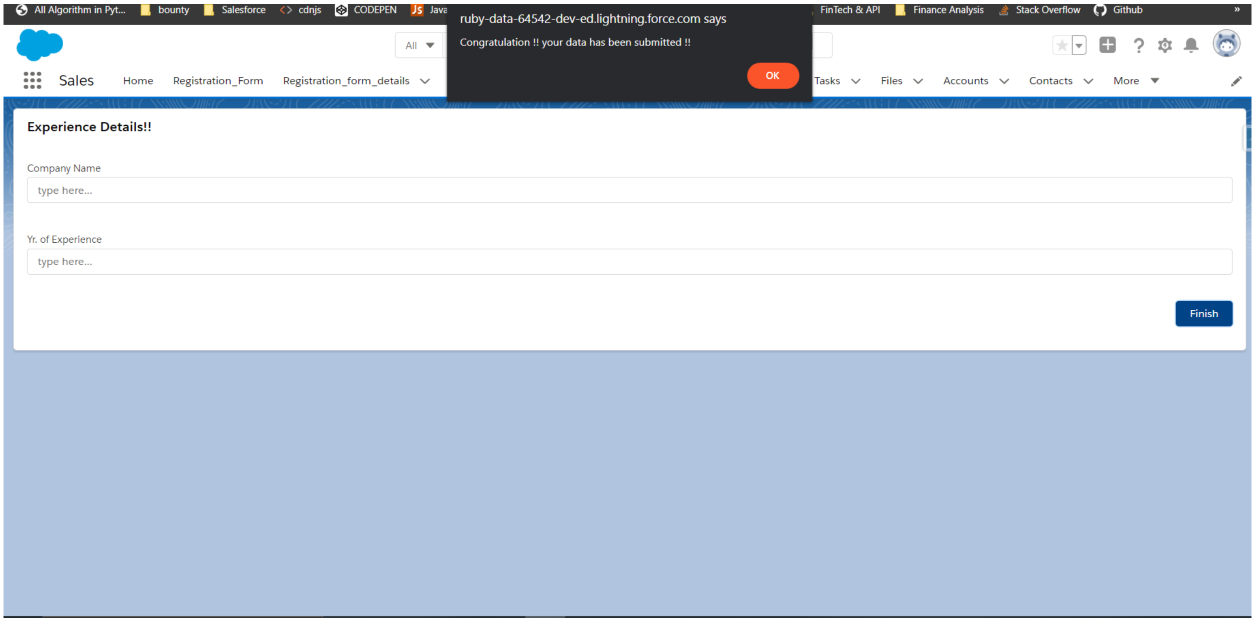
--->

Experience form   Database server




All
Search...
Sales
Home
Registration_Form
Registration_form_details
Education
Experience
Opportunities
Leads
Tasks
Files
Accounts
Contacts
More
Education Details!!
Canditate 1oth pass
Canditate 12th pass
Canditate Graduate
type here...
Next
Type here to search
ENG
14:08


All
Search...
Sales
Home
Registration_Form
Registration_form_details
Education
Experience
Opportunities
Leads
Tasks
Files
Accounts
Contacts
More
Experience Details!!
Company Name
type here...
Yr. of Experience
type here...
Finish
14:09


All Algorithm in Pyt...
bounty
Salesforce
cdnjs
CODEPEN
FinTech & API
Finance Analysis
Stack Overflow
Github
ruby-data-64542-dev-ed.lightning.force.com says
Congratulation !! your data has been submitted !!
OK
Sales
Home
Registration_Form
Registration_form_details
Tasks
Files
Accounts
Contacts
More
Experience Details!!
Company Name
type here...
Yr. of Experience
type here...
Finish

FIRST USE CASE :
===> **WEALTH MANAGEMENT :::**

Second UseCase ⇒ Loan processing
KIVA : Salesforce with loan lending, repayment

Application
Underwriting
Decision
Offer
Document Processing
Closing

Identified the customer (social advertisement)
Know the authenticity (OCR, facial, document)
Get the process done (get processed all the data)

Common to all: (Integration with vendor)
Salesforce with finance service

How IPO WORK:: Category:

1) Institute (QIP)
2) HNI (NRI)
3) Retail
4) Company (Promoter Holding)

Filtration :

1) Wrong Application
   (Double with Same pan card, UPi Error, Bank broker acc name difference)
2) Not Cut off Price

Normal Subscription: 1-> n
High Subscription : n->1

How Share Market Work: **Buying and Selling at a Certain Price**
Equity: Loan + interest on profit
Debt : Loan + Interest

Promoter ⇒ Raise Money ⇒ Diluted Equity ⇒ Less Ownership
Profit Increase // Market Cap increase (PE Ratio)

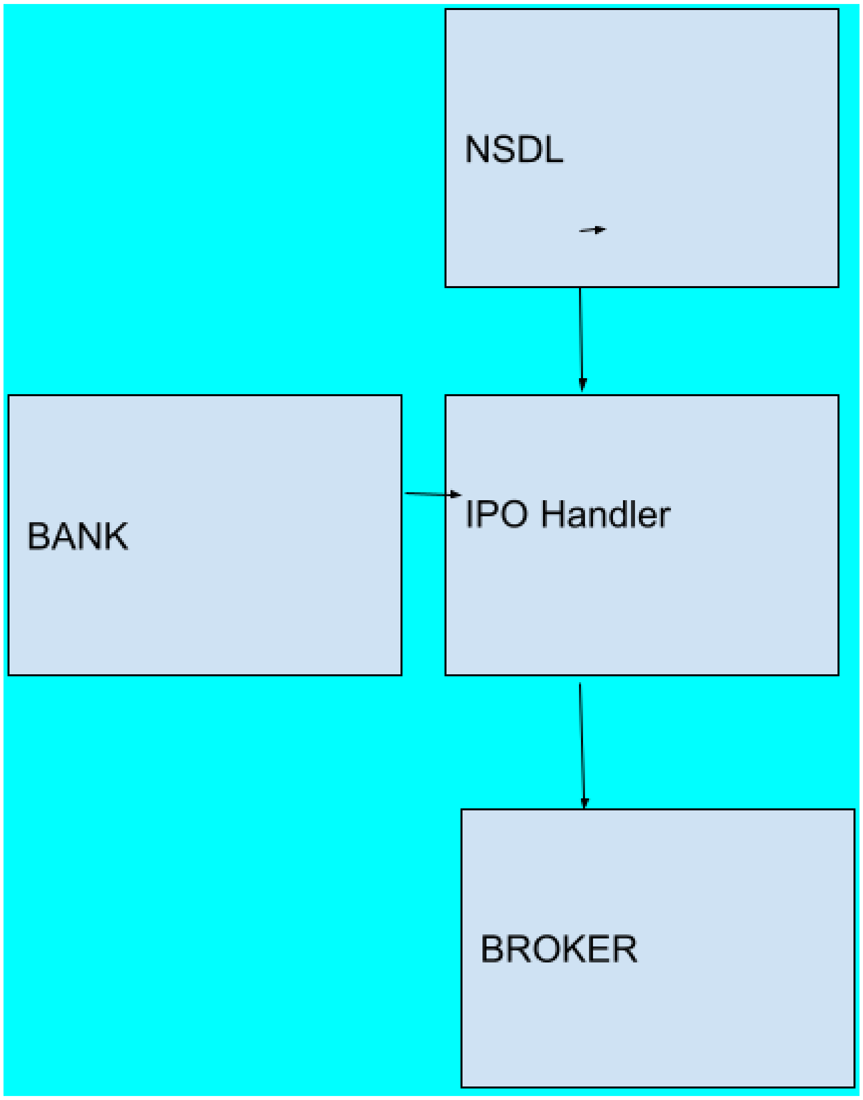
Offer for sale**(Selling Again)**, Share BuyBack **(Buying again)**

Dividended **(Profit and bonuses) (EBIDATA/ CAGR)**

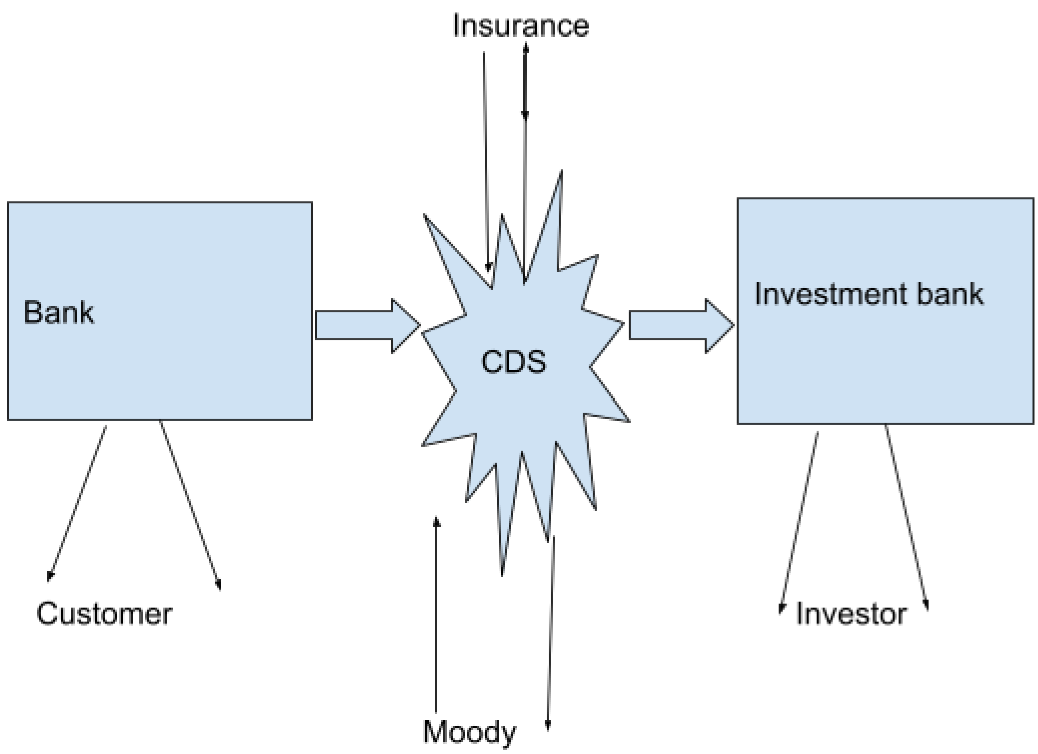
Number of share= Total Market cap/ face value
Value per share = Total Market Cap/ No of share
Holding = Total amount invested / Per share value

Transaction Keeps happen and value increase
If transaction gets hold
Share split up

Key:-
Get early in boat
Publicly goes to avoid Conflict

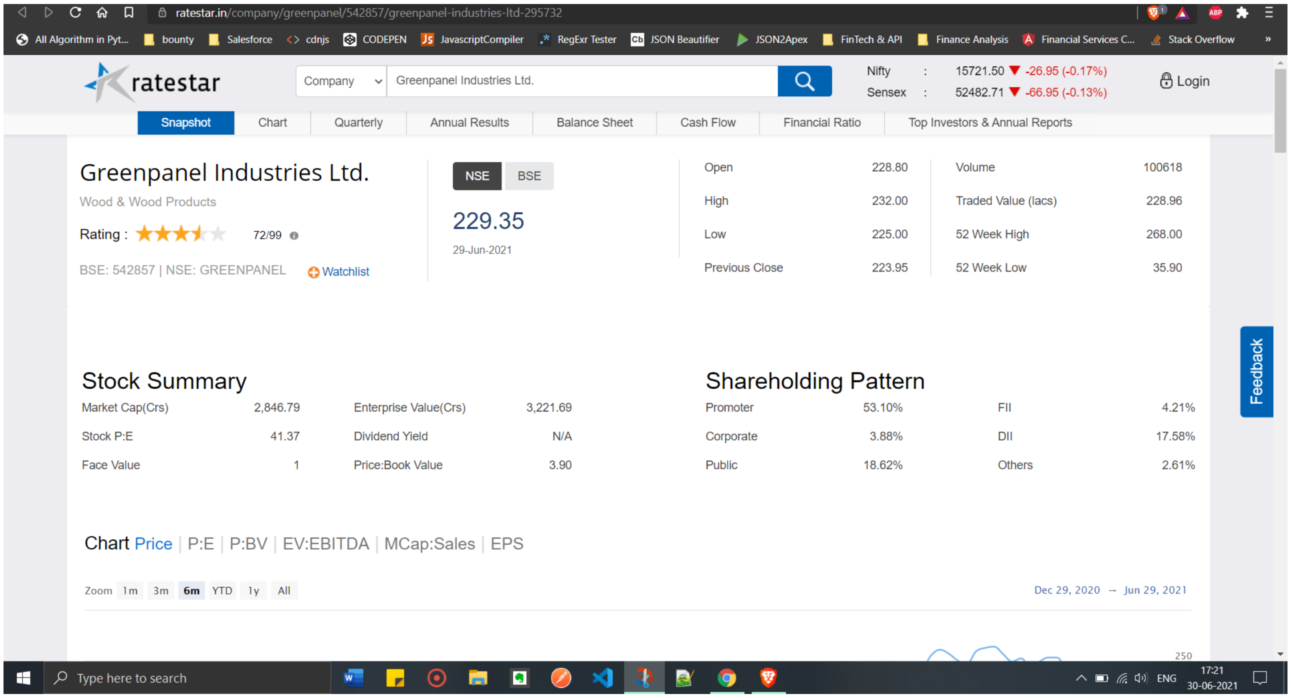

NSDL
BANK
IPO Handler
BROKER

How mortgage work:-



Fourth Use Case: (Live data updating) : Salesforce with Stock Wealth Management
Penny Stock Query ::::

```
Market Capitalization > 200 AND Return on equity > 20 AND
Free cash flow 5years > 0 AND Free cash flow 10years > 0 AND
Debt to equity < 1 AND Interest Coverage Ratio > 3.5 AND
Return over 1year > 0 AND Return over 5years > 10 AND
Promoter holding > 35 AND Sales growth 10Years > 7.5 AND
Sales growth 5Years > 7.5 AND Profit growth 10Years > 15 AND
Profit growth 5Years > 17.5 AND Profit growth 3Years > 10 AND
Profit growth > 0
```


ratestar.in/company/greenpanel/542857/greenpanel-industries-ltd-295732
All Algorithm in Pyt... bounty Salesforce cdnjs CODEPEN JavascriptCompiler RegExr Tester JSON Beautifier JSON2Apex FinTech & API Finance Analysis Financial Services C... Stack Overflow
ratestar
Company
Greenpanel Industries Ltd.
Nifty : 15721.50 -26.95 (-0.17%)
Sensex : 52482.71 -66.95 (-0.13%)
Login
Snapshot Chart Quarterly Annual Results Balance Sheet Cash Flow Financial Ratio Top Investors & Annual Reports
Greenpanel Industries Ltd.
Wood & Wood Products
Rating : 72/99
BSE: 542857 | NSE: GREENPANEL
Watchlist
NSE BSE
229.35
29-Jun-2021
Open 228.80
High 232.00
Low 225.00
Previous Close 223.95
Volume 100618
Traded Value (lacs) 228.96
52 Week High 268.00
52 Week Low 35.90
Feedback
Stock Summary
Market Cap(Crs) 2,846.79
Stock P:E 41.37
Face Value 1
Enterprise Value(Crs) 3,221.69
Dividend Yield N/A
Price:Book Value 3.90
Shareholding Pattern
Promoter 53.10%
Corporate 3.88%
Public 18.62%
FII 4.21%
DII 17.58%
Others 2.61%
Chart Price | P:E | P:BV | EV:EBITDA | MCap:Sales | EPS
Zoom 1m 3m 6m YTD 1y All
Dec 29, 2020 → Jun 29, 2021
250
Type here to search
ENG
17:21
30-06-2021

Accenture Task1:

Email Message is an Object where all email related informations have been stored.

1) It is not accessible from app launcher and it has no UI Record capabilities so lightning record page also can not be created.
2) It always comes under the relation of the object from where it's being shared.

Launching a flow :

Form a record details page or detail page button
Recordid can be passed in that flow.

Record details Page Layout configuration is needed when to pass record id
For details page button it is not needed as it automatically passes it.

Note: record id while declaring the variable

Duplicate: get, create with update and delete previous one
One field : workflow
Entire object material : trigger, screenflow

Refresh :
Redirection : **Navigation to record**

Accenture Task2:

Multilingual language org, business hour, holiday, email template (seperate folder, one email translation)

Task 3_Accenture_Company_project :

To transform the TransferEmailMessage flow into omni script
What i did in flow:

1) Pass the record id in flow by adding flow in a button and through button pass record id
2) Get the email message record of that case id
3) In screen show to enter a new case
4) Create a new email message record of that new case with exact same data
5) Delete old email message
6) redirect / refresh to new record

Same procedure in omniscript:

Example: edit any account through omni script

1) Load the particular data
2) Show the field that you wanna edit
3) Save the data
4) Navigate

Personal Project:

1) SFDC Blog
2) Youtube video creator

SFDC FACTORY

BLAZE YOUR TRIAL

1st video :

What is salesforce? Cloud framework ? Application ?
sales, marketing, service, health could, fsc
Vlocity

2nd video :

How to pass record id to flow ?
2 ways (through button and through page record id)

Accenture Email-to-Case Task: (ID 40)

1) Email-to-Case
2) Queue (Administrative)
3) Organization wide address
4) Gmail Verification and forwarding email message setting

Question:

Dev Org Access?
Email Address Access?
Who will be the members in the Administrative queue?

Auto Labelling :

1) case setting added on routing
2) case record type page

3) trigger to update : post deployment step of custom label value update

4) Metadata Wrapper

Auto response (Case Auto Response)
Auto response after customer send first email (Support Setting)

If still email does not gets created, check required fields for case creation

----

Development: (Configuration)

1) **Trigger**
2) **Test Class**
3) **Post deployment step to update custom label**

Accenture: (ID 60)

1) Case page Layout
   Fields
   Email related tab List
   Feed

Accenture: US 79: claim assessment task creation through action plan

1) Task Record creation

2) Activity object field
How are Activity object fields shown on task objects?
because custom fields for Tasks and Events are created at the Activity level.
(record type we need to create)

3) Type can not be filled to internal task : use by default value

How to include more fields on task object Area on Action Plan template? : NO
How to use record type in assignment of task creation through action plan? :NO
From where these fields are populating on this record ?
--

Accenture Full Work Outline :

1) Standard Email-to-Case
2) Custom Email-to-case
3) Preliminary Classification
4) Case Categorization
    => Status changes, get assigned to right queue
    => Auto linking of entitlement and service agreement happens
5) Queue Member can see and assign themselves
    => Milestone start

6) Claim Creation happens
7) Auto claim participant creation happens
8) New claim participant/ Broker functionality

9) Internal Request for processing functionality done
10) Manual Submission, Auto Submission, Auto Approved functionality happens
11) Queue Assignment, field validation, Workaround functionality
12) Notification, task, note, attachment, field value updation happens

13) Checklist
14) object, field, validation rule, page layout, record page, list view

<u>Accenture Account Model:-</u>
Account (Individual, Household, partner-broker)
Person Account (Customers, Partners)
Customers: insured parties(family, employees), policy owners
Partners: witness, lawyer

Flow: After partner creation, just check whether they are owning the policy or not if yes create a customer account for that.

Note: one person can be a customer, and can be a partner as well.

Flow2: when a claim gets registered, claim participant gets created with policy and if the claim participant is not registered in account as a person account then create a person account as well.

Case with insurance policy <=> Claim with Insurance Policy
Account => Contact
Product => Insurance Policy => Coverage type => Insurance policy Coverage

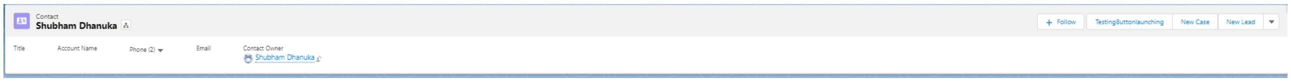
Basic steps to understand in details:-
1) Case Creation (Email service handling)
2) Case Categorization *(based on department, language)*
case is in Handling/ Assessment stage

3) Claim Creation
covered, Not covered
Allow in both cases and if not covered then create a internal request to check and then cancel or allowed for covered
4) Automatic Claim Participant creation, Manual participation creation

Claim in handling/Assessment stage
5) checklist of claim assessment, internal request to shorted out all the issues

6) Creation of claim coverage
7) Payment processing

button to launch the component:-
https://salesforcediaries.com/2020/10/06/open-lightning-component-from-url-button-in-lightning/



Direct landed on creation page with what record type and request type needs to be processed :-

preselected checkbox value on creation page
preselected picklist value on creation page

This is Navigation:-
Open for creating new record
Open for creating new record with default value

HTML:

```
<template>
    <h1>Here we want to learn!!</h1>
    <lightning-button label="Navigation" title="Non-primary
action" onclick={Navigate}>
    </lightning-button>
    <lightning-button label="navigatetonewrecord"
title="Non-primary action" onclick={navigatetonewrecord}>
    </lightning-button>
    <lightning-button label="navigatetonewrecordwithvalue"
title="Non-primary action"
        onclick={navigatetonewrecordwithvalue}>
    </lightning-button>
</template>
```

JS:-

```
import { LightningElement } from 'lwc';
import { NavigationMixin } from 'lightning/navigation';
import { encodeDefaultFieldValues } from 'lightning/pageReferenceUtils';
export default class LightningWebComponentLearning extends
NavigationMixin(LightningElement) {
    Navigate() {
        this[NavigationMixin.Navigate]({
            type: 'standard__namedPage',
            attributes: {
                pageName: 'home'
            },
        });
    }

    navigatetonewrecord() {
        this[NavigationMixin.Navigate]({
            type: 'standard__objectPage',
            attributes: {
                objectApiName: 'Contact',
                actionName: 'new'
            },
        });
    }

    navigatetonewrecordwithvalue() {
        const defaultValues = encodeDefaultFieldValues({
            FirstName: 'Shubham',
            LastName: 'Dhanuka',
            LeadSource: 'Other'
        });
        this[NavigationMixin.Navigate]({
            type: 'standard__objectPage',
            attributes: {
                objectApiName: 'Contact',
                actionName: 'new'
            },
            state: {
                defaultFieldValues: defaultValues
            }

        });
    }
}
```

Duplication Rule:-

Standard Duplication and matching rule is there.

In the Standard way when you create a record, it will give you the error.
But suppose u are not creating a record in standard way (new button)

In that case:-
Two way:-
First:

Use Database.insert and map out all failure record and show duplication record

Second:-

```
public class AccountRecords {
    public string accname = 'sForce';
    private boolean allClear { get; set;}

    public AccountRecords(){
        allClear = true;
    }

    public void saveData(){
        allClear = true;
        Account acc = new Account(name=accname);
        list<Account> accounts = new list<Account> ();
        accounts.add(acc);
        //accounts = [select Name from Account];
        list<Id> ids = new list<Id>();
        Datacloud.FindDuplicatesResult[] results = Datacloud.FindDuplicates.findDuplicates(accounts);
        for (Datacloud.FindDuplicatesResult dupeResult : results) {
            for (Datacloud.DuplicateResult dupeRes : dupeResult.getDuplicateResults()) {
                for (Datacloud.MatchResult matchRes : dupeRes.getMatchResults()) {
                    for (Datacloud.MatchRecord matchRec : matchRes.getMatchRecords()) {
                        allClear = false;
                        ids.add((string)matchRec.getRecord().get('id'));
                    }
                }
            }
        }
        if(!allClear){
            System.debug('size of duplicate' + ids.size());
            System.debug('duplicate id' + ids);
        }
    }
}
```

ARAG Object Model

Account : (Customer, Competitor, Partners) (Person involved with your business)
Contact : **(Account Name)**
Case : customer request **(Account)**

Insurance policy : Type of policy (life insurance, auto, home) **(then Product segregation)** **(Product, producer, Account(M-D), claim id)**
Insurance policy product : Multiple product to one insurance policy (**CP ID, insurance policy, product**)

Product : represents product (in auto (car, bike, truck)) (u have one land of 1 cr)

Product coverage: junction object between product and coverage
Policy: package → policy (**Coverage type, product, CP ID**)**(we can cover upto 10 lac)**

Coverage Type: Type of coverage (liability, earthquake, collision damage) **(CP _ID)** **(we will provide coverage in any of following coverage type)**

Insurance policy coverage: junction object in between insurance policy and product coverage (**CP ID, insurance policy(M-D), coverage type**)

Customer Property: represent insurable/insured property of a customer (**CP Id, Primary owner(M-D)**)

Insurance Policy Asset: represent the customer property insured through insurance policy (**CP ID, insurance policy(M-D), customer property**)

Premium pay out:

Claim : Type of claim (life insurance, auto, home) **(Product)**
(cp id, case, producer, , mass claim, Account)

Claim Coverage : insurance policy coverage for a claim
(What you covered under that package) (CP ID, claim(M-D), insurance policy coverage)
This object includes financial activities related to Claim including payout and reserves

Claim participant: A junction object associate the claim object with account object
This object includes insured, claimants, witness
(**CP ID, participant account, claim(M-D), Participant contact**)

Claim payment summary : represent the amount of money that is paid to claimer
(**CP iD, claims(M-D)**)

Financial Account : investment account/ bank account / insurance policy
(CP ID, primary owner)

Financial Account Transaction :
(**CP ID, Claim payment summary, claim, financial account**)

Internal request : custom object that will use to store every request from claim advisor to a colleague manager and other department
(CP ID, claim, claim payment summary)

Mass Claim : Custom object used to link claims about same topics **(CP ID)**
Grouping of people under a company

Producer : represent a agent or agency selling product (Account → Contact)
(**CP ID, Account, Contact**)

Smart conn Integration : Document processing

Product ⇒ Policy ---> coverage (policy can have multiple coverage (1:N))(Emp, spouse)
Claim ---------------------> Coverage (1:N relationship)
One event can have multiple incidents (window. table, bicycle)

===

Policy and claim management :-

1) Product management
2) Quote management
3) Policy management
4) Claim management
5) Offer/Contract management
6) Attribute

Sales and broker management :-

1) Account and contact management
2) Opportunity management
3) Campaign management

Customer Service:-

1) Case management
2) Omnichannel

Broker Portal:-

1) Opportunity management
2) Case & complaint management
3) Claim & Document management

ARAG Different System:- (Mulesoft MiddleWare)

ISABEL
ZOOMIT
IBM CMS
Cloud storage (M-Files)
SAP
BanqueCarrefour
Portima
FSMA Broker scan
MS DWH
DIV (Vehicle registration)
KYC
ILEO
DIO WEB

Lead Account Creation flow (Lead to prospect broker(Account) creation)

Lead : signing up on the site, filling out a form, downloading a buyer's guide, giving their information at a trade show booth, etc

NEWS: twitter authentication and feed news tracking

Opportunity : where we can target

Account and Contact :

Identified the customer (social advertisement) :- LEAD
Know the authenticity (OCR, facial, document):- Account, Contact

Account is having parent account (Master-Details Relationship)

E.g:- Accenture global company
Accenture India, Accenture Uk all are account to Accenture global and all are account
One particular entity (parent, child) have their spoken person (call contact)

After the lead process, Once Lead is converted to Account, Account record type is set up as a partner broker.

Get the basic information, fill in the details, get an approval, sign the agreement and onboard it.

#Supplier Onboarding

Prospect (Broker) Account to customer creation

- **Experience cloud (Community setup) (revolving around with community)**
    - Signed Agreement
    - Customer

Customer to client creation :-
Product/services => price/ Quote(offer) => Opportunity (Account)
Account, policy Creation flow :-

Define product
Define price
Define Quote
Define Contract

Define Product : (Multi product policy set up) *(this is product & services i am having)*

Product page information:
Coverage
Insured item
Rating
Attribute
Rule
Simulation

Define Price :-

Define Quote (Offer) (different type) => **Quote_Add_product**

Can be general
Can be user specific ⇒ **require Approval**

Quote (lookup to insurance policy) **(Quote)**
Quote Line item (master details to Quote) (lookup to product) **(QuoteLineItem)**
Quote Pricing Adjustment
Quote Line Item pricing Adjustment

Quote: Proposed price of company product and service
It can be created from Opportunity, Product and Account itself
Quote line item : different combination with product and services

For service:
Silver, gold, platinum package (Quote for the services)
Each opportunity can have multiple Quote

Define Contract (Policy)

Once a contract is signed, we have step of actions/stages that has to be taken for the contract :-

Final draft after negotiation

Approval

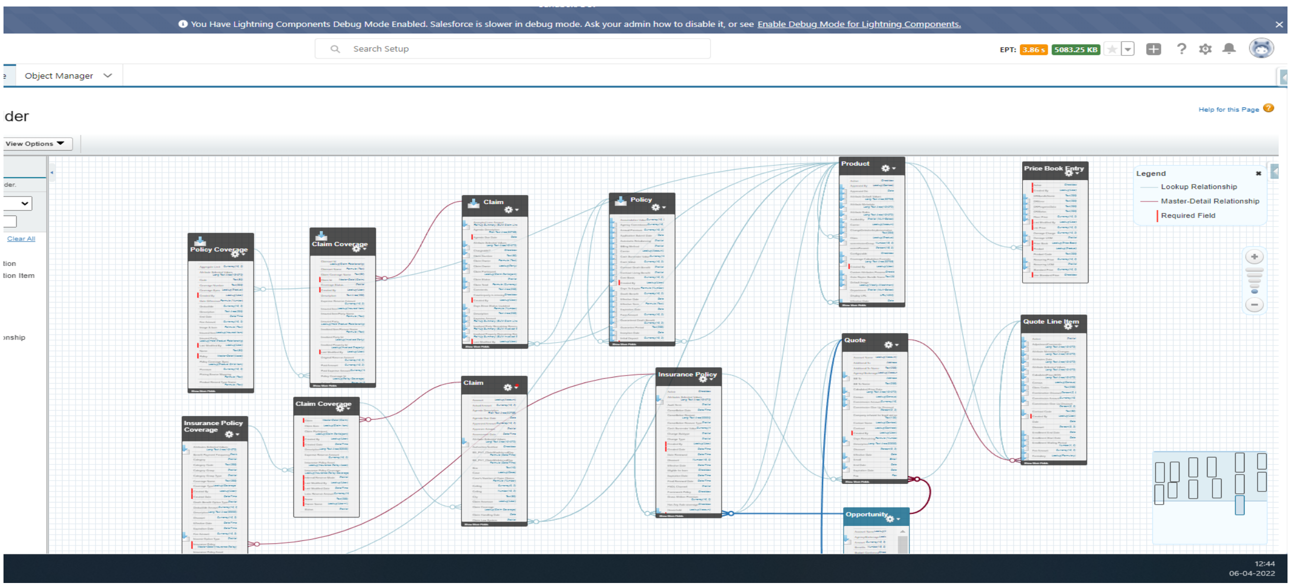
Signed

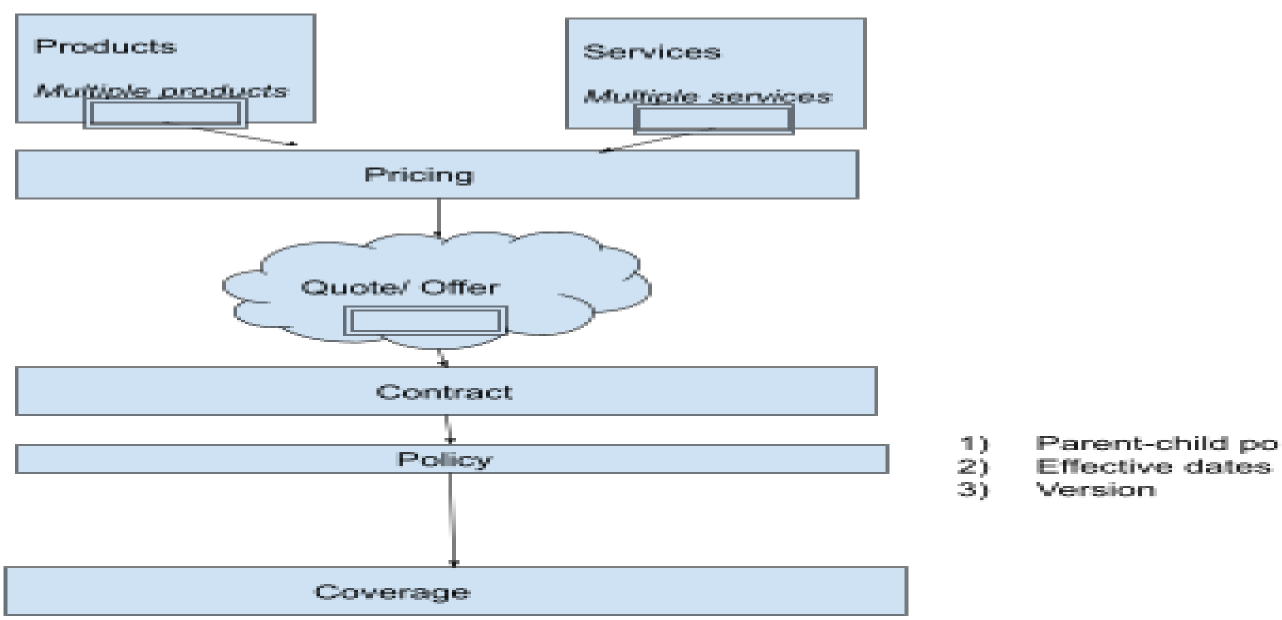
Expired, Renew , Terminated

Document Processing (Get the process done (get processed all the data))

1) Insured items 2) Effective dates 3) Policy version 4) Payment permium

Closing





Services Management :-
Case And Claim Creation flow:-
Case comes to Salesforce (Email Messaging service)
Preliminary classification (record triggering flow on email message)
Case categorization (OS and record triggering flow on case object)
Case stage of processing

Different record type creation:
1) LHL => (Combination of language and department and type) =>Queue assignment
2) Claim => (Combination of language and department and type) => Queue assignment
3) Whistle blower = > for internal user => any Misbehave => Legal team assignment
4) Complaint => services related => complaint management
5) Product => product feedback => Product team

Claim creation (OS) :-

Claim gets created
Claim participation records gets created (record triggering flow)

Claim Assessment :-

Same with Policy (Parent -> Child)

Policy (Self lookup to policy) **(InsurancePolicy)**
Claim (lookup to Policy) **(Claim)**
Claim Coverage (master details to claim) **(ClaimCoverage)**
Policy Coverage (Lookup to claim coverage, master details to policy)**(InsurancePolicyCoverage)**

INTERNAL REQUEST MODULE:- (can be created from case/ claim page layout)

Type of internal request :
Record creation, Auto record details populated

Record Assignment:
Queue, Queue Notification (BE_PH1_ownerfieldUpdate)

Submission:-
(Auto Submission)
(Manual Submission)

AutoApproval	Mandatory: Flow intervention (BE_PH1_InternalRequestTrigger)
Approval	Trigger, AutoLaunchedFlow
Reject	

Notification:-

Task Creation
Notes Creation
Notification

Outsourcing to Lawyer :-
At stages validation rules are there to prevent any misbehave or any confidentiality mismatches

Salesforce outlook integration:-

So Agent/ or any person is working on emails, they wanna have same view on outlook as well without login to salesforce.
We can do that.

1) its like salesforce web
2) salesforce mobile application
3) Salesforce connection so they can view in outlook as well.

Same as mobile app configuration :-

1) View Task
2) View Email
3) Update record
4) Email Template / Quick short
5) Meeting setup

How to turn it on in salesforce ?

How to deploy it for the user?

How to do a log event or create a custom email application using Lightning App builder?

Advance feature:-

Einstein Activity :

1) Automate Email
2) Event logging
3) Sync Contact
4) Calendar events

Security : Salesforce email integration
Einstein Activity capture

More information :-

https://getoncrm.com/outlook-integration-with-salesforce-a-step-by-step-guide/

Product specification :-

1) Root product e.g (**Human Body**)
2) Insured Item and party specs e.g (**eye**)
3) Coverage specs e.g (eye burn, eye disorder) (**eye check up is allowed not specs**)

Define the attribute: -E.g : (Maximum price, Minimum price) (specification) (fact)
E.g: color, skin
Define the rule :- (specification rule) e.g (Hospital network rule)
—---

1) **Workgroup**
2) **Workgroup member**
3) **Workgroup rule**

—---
Company is : Home Heaven
Product is : HomeOwners Insurance product

Root product
Product code

Insured item and party specs
Insured item
Product code
Parent code

Rating and rule
—---
Offer
Mutliproduct select => Multi item select => calculate premium

Follow the process => once finalize

Issue policy

Modiciation:
Endorsment => Correction => Modification => Creating version of policy
And it also creating version of policy whose products are not in effective state

Brokerage Commision:
Brokerage commission