Smartpack

**1. Introduction**

This document describes in details the design followed to achieve smart pack flow in AEM.

**2. Requirement**

In Abbott, each product in the product page is going to have the barcode. When user clicks on corresponding bar code then the smartpack page should display. The data for this page should come from excel.

Business has provided excel with the following.

1. Excel has Master sheet, Tab1, Tab2, Tab3
2. Master sheet – will consists of SKUNUMBER, Productname, Tab1, Tab2, Tab3.
   1. SKUNumber refers to bar code or product code.
   2. Productname refers to name of the product.
   3. Tab1, 2, 3 refers to 1,2,3 sheets which are in excel.
   4. We are adding a column called “Action”. When ever some action taken place this column should be updated with Insert\Delete\Update.
3. Sheet 1, 2, 3 in the excel should display as Tab1, 2 ,3 on the page
4. Each Sheet has Sections (columns) such as Section1, Section2 and Section3. Each Section should appear as accordion in the tab.
5. That is Sheet1 should appear as Tab1. Inside Sheet1 we have Section 1, 2, 3, 4 Which should appear as Accordion1,2,3,4 inside Tab1.
6. Sheet 2 should appear as Tab2. Inside Sheet2 we have Section 1, 2, 3, 4 Which should appear as Accordion1,2,3,4 inside Tab2.
7. Sheet 3 should appear as Tab3. Inside Sheet3 we have Section 1, 2, 3, 4 Which should appear as Accordion1,2,3,4 inside Tab3.
8. Each Section i.e; Sheet1(Tab1) –Section1(Column) in the excel has some content. This content should appear as content inside accordion.
9. Accordion content is rich text and image.

Business wants to upload 1500 records. We are proposing to upload all 1500 records in the same excel.

**Functionality**: Initially we will read the data from excel for all the 1500 records and create corresponding pages for all those records.

We are providing authoring capacity for the users. Hence users can do the content change on the page.

But if there is a change in excel then excel should override the content on the page. That is page content changes done by author will stay until excel modification or addition.

Note:

* The script that reads the spreadsheet/JSON will only create new pages. It will skip the product if the page already exists.
* Also script should execute if there is any change in the existing products.

**Master sheet format:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SKUNUMBER | Product name | Action (Insert/Delete/Update) | Tab 1 Title (Default) | Tab 2 Title | Tab 3 Title |
| DF1434 | Dublexa | I | Product Info | Therapy Info | Mandatory info |

Based on the Action column will identify the corresponding action in excel .

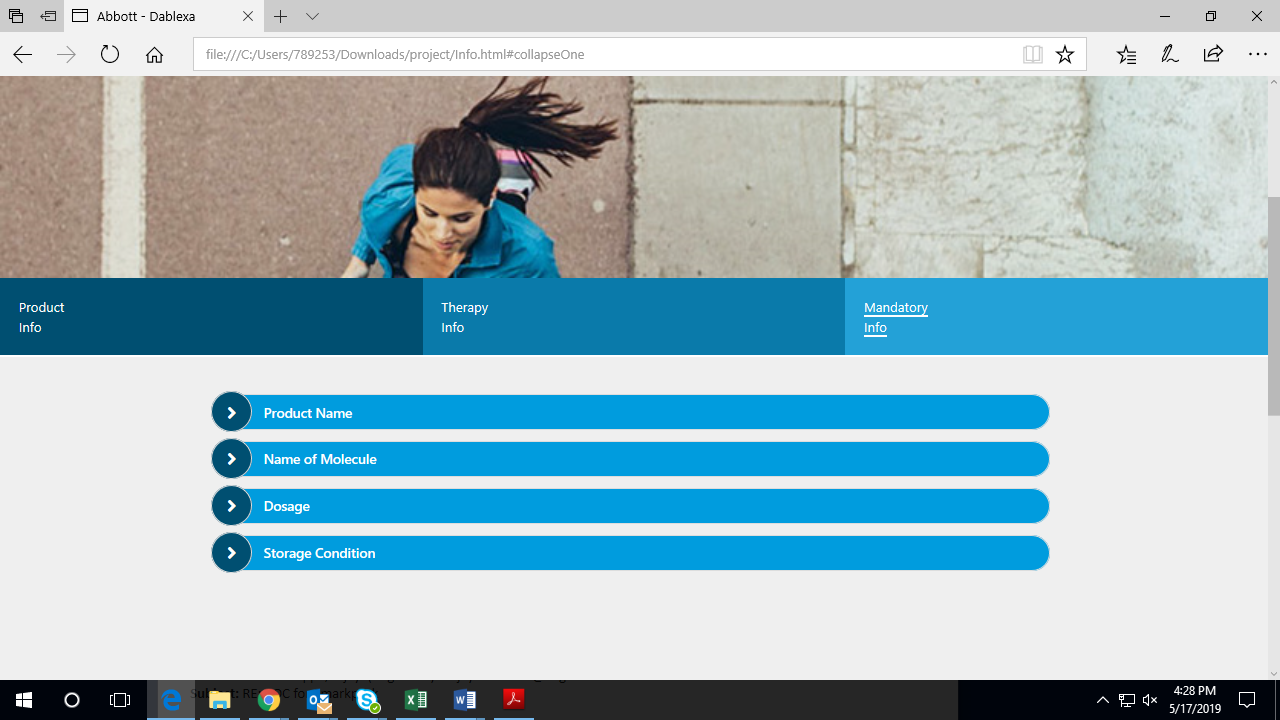
* Image or Video should be uploaded in DAM or Server and corresponding path should be entered in excel carousel column (Example Section 1 Carousel)
* Special characters should not entered in excel and format of excel should not change.

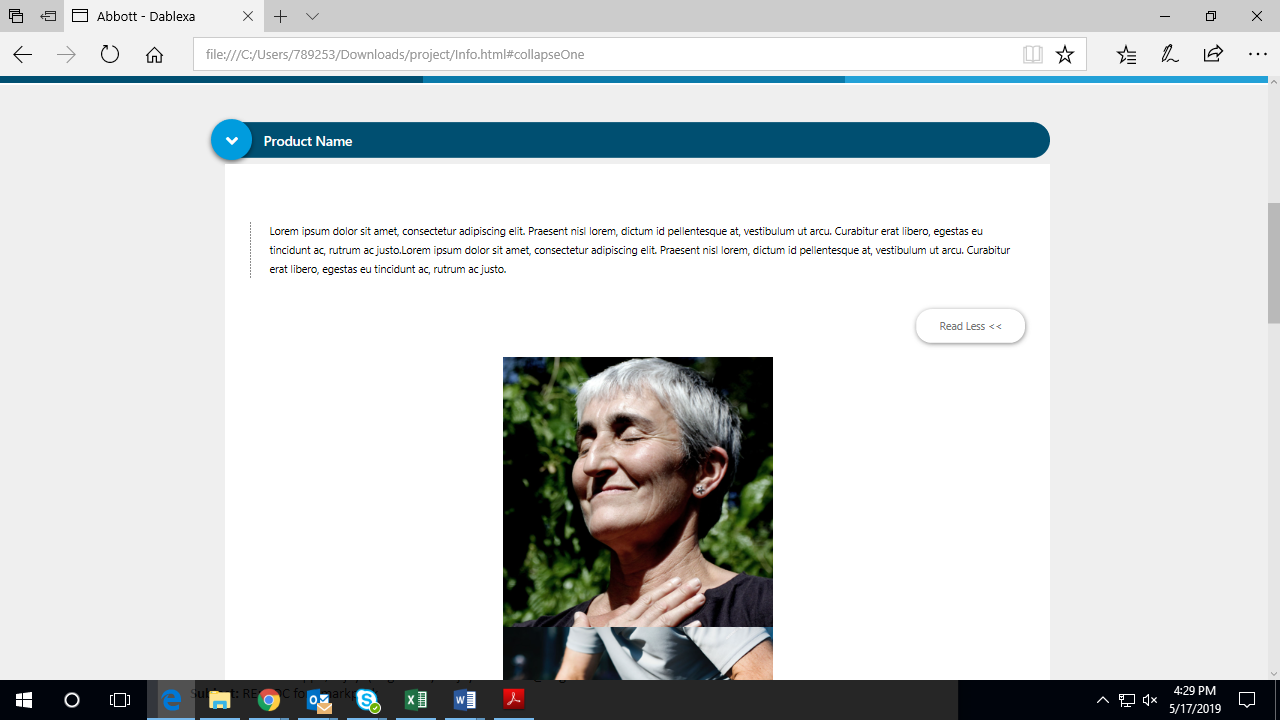
**Existing approach**

It’s a new implementation

**Technical Specification**

New Pages Added





**AEM Changes**

Create template – Lets consider as smartpack template

Create below components

* Header – Static
* Footer - Static
* Tabs – create custom component than using existing out of box tabs because it needs to incorporate with accordion.
* Accordion – Can be extend from out of box and customize
* Rich text - Out of box can be reuse for time being
* Image - Out of box can be reuse for time being

**Services**

Service 1 – To create nodes in jcr by reading excel

Service 2- To read the stored data from JCR .Lets consider the tentative path as etc\products\product codes.

***Implementation:***

1. Upload excel to the dam. Make sure the path which we stored excel is static. Also format of excel should not change and excel does not have special characters.
2. Write a service (for example service1) that reads the data from excel.
3. Service should go the master sheet of excel and read the product codes. By iterating the product codes, it should create a node for each product code in etc\products\productcode. Then it should go back to the excel Tab1 . Service should identify the productcode (SKUNumbe) in the tab1 and it should iterate the section1,2,3 and so on for the particular product. Similarly it should read and iterate for all the product codes. Also repeat the same logic for other sheets(tabs).Each tab and corresponding data should store as etc\products\productcode\product1\..\..
4. We can use apache poi or any other plug in which supports both 6.1 & 6.3 for reading data from excel.
5. The data in the nodes should store in JSON or any other structural format.
6. Once the data is stored in JSON format then it can able to read and print.
7. Going back to component part, from the tabs component we have to do the call for model class.
8. This model class will call the service 2.
9. Need to implement the service 2 should read the stored data in the JCR and provide the printing logic to the component.

With the above implementation we can able to read the data from excel, store in jcr and print on the page.

Once it is done then we need to implement the workflow to make sure data is updating whenever data excel change.

***Workflow***

Create a launcher which keeps listening to excel. Whenever some activity happened in the excel then the launcher will trigger the work flow. Create a custom workflow which should trigger automatically with the excel changes. Business can add, modify or delete data in excel. Workflow should trigger when ever some action happens on excel. That is workflow should trigger automatically if some modification happens in excel. This workflow should call Service1 which reads the data and corresponding above mentioned actions happened. Content on the page will update with workflow execution.

**Project Structure:**

We are doing it as POC and not merging with existing code base for time being. Hence lets keep simple structure as follows

Create smartpack project inside apps. That is apps\smartpack

And create corresponding components, templates , install, src folders inside the smartpack folder.