SUMMER FELLOWSHIP REPORT 2020

ON

DRUPAL

UNDER FOSS NAME

WEB DEVELOPMENT

Submitted By:

Dolon Mandal

NIT AGARTALA

Under the guidance of

Prof. P. Sunthar

Chemical Engineering Department

IIT BOMBAY

APR 21, 2020 TO – JUNE 20, 2020

Acknowledgement

This report has been prepared for the internship that has been done virtually, in order to study the practical aspects of the Drupal and its implementation in the real field with the purpose of learning and exploring in the mentioned field and for professional development.

The aim of the internship is to be familiar with the implication Drupal and uses of concerned theoretical knowledge and clarifying the career goals so, I have successfully completed the internship and compiled this report as the summary.

I would like to express my sincere gratitude to my project head Prof. P. Sunthar

Who has given me this valuable opportunity to work in such a learning environment to get industrial-based experience I am also grateful to the team members, my mentors Miss Ruchi Kumari and Mr Tejas Vaidya, their constant supervision and support was truly helpful in completing my tasks and my internship.

Lastly thanks to my parents, my sister and friends for all the moral support.

DOLON MANDAL

Place: Kolkata

Date: 20-06-2020

CONTENT

6. Plugins

- **Introduction to Plugins**
- **Source, Process, Destination Plugins**
- **Custom Plugins**

7. Composer

- **Introduction of Composer**
- **How to install composer**
- **Requirement of Composer**

What is DRUPAL??

Drupal is a free and open-source content management framework written in PHP and distributed under the General Public License (GNU). Drupal provides a back-end framework for at least 2.3% of all websites worldwide ranging from personal blogs to corporate, political, and government sites. The standard release of Drupal, known as Drupal core, contains basic features common to content-management systems. These include user account registration and maintenance, menu management, RSS feeds, taxonomy, page layout customization, and system administration. The Drupal core installation can serve as a simple website, an Internet forum, or a community website providing for user-generated content.

Although Drupal offers a sophisticated API for developers, basic Web-site installation and administration of the framework require no programming skills because of which the user with minimal knowledge of coding can build a website with Drupal. Drupal runs on any computing platform that supports both a web server capable of running PHP and a database to store content and configuration. Drupal is a content management software. It's used to make many of the websites and applications you use every day. But what sets it apart is its flexibility; modularity is one of its core principles. Its tools help you build the versatile, structured content that dynamic web experiences need.

MIGRATION

We use the word "migration" as a term for any process that seeks to take data from some source to the current Drupal 8 site and use it to automatically create nodes, users, configuration, and any other component of our site. In short, automating what might otherwise be a tedious job of copying and pasting.
Drupal 8 brings a new migration system into Drupal core, with the goal of making it easier to import data from a variety of sources. The migrate system is both a framework designed to facilitate writing custom migrations, and an implementation of that framework aimed at Drupal-to-Drupal migrations.

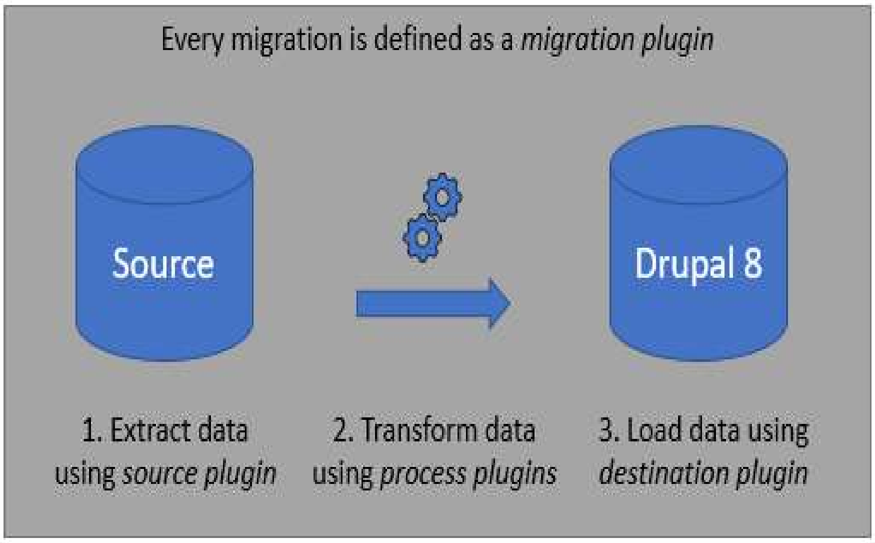
The migration system makes it possible to pull content into Drupal from just about anywhere. The core API supports extraction from any SQL data source, including previous versions of Drupal. Contributed Modules extend this system to support other data types like CSV or JSON, as well as other platforms like WordPress.
Some of the existing data sources include:

- MySQL
- Previous versions of Drupal
- CSV
- JSON,
- XML
- Etc.

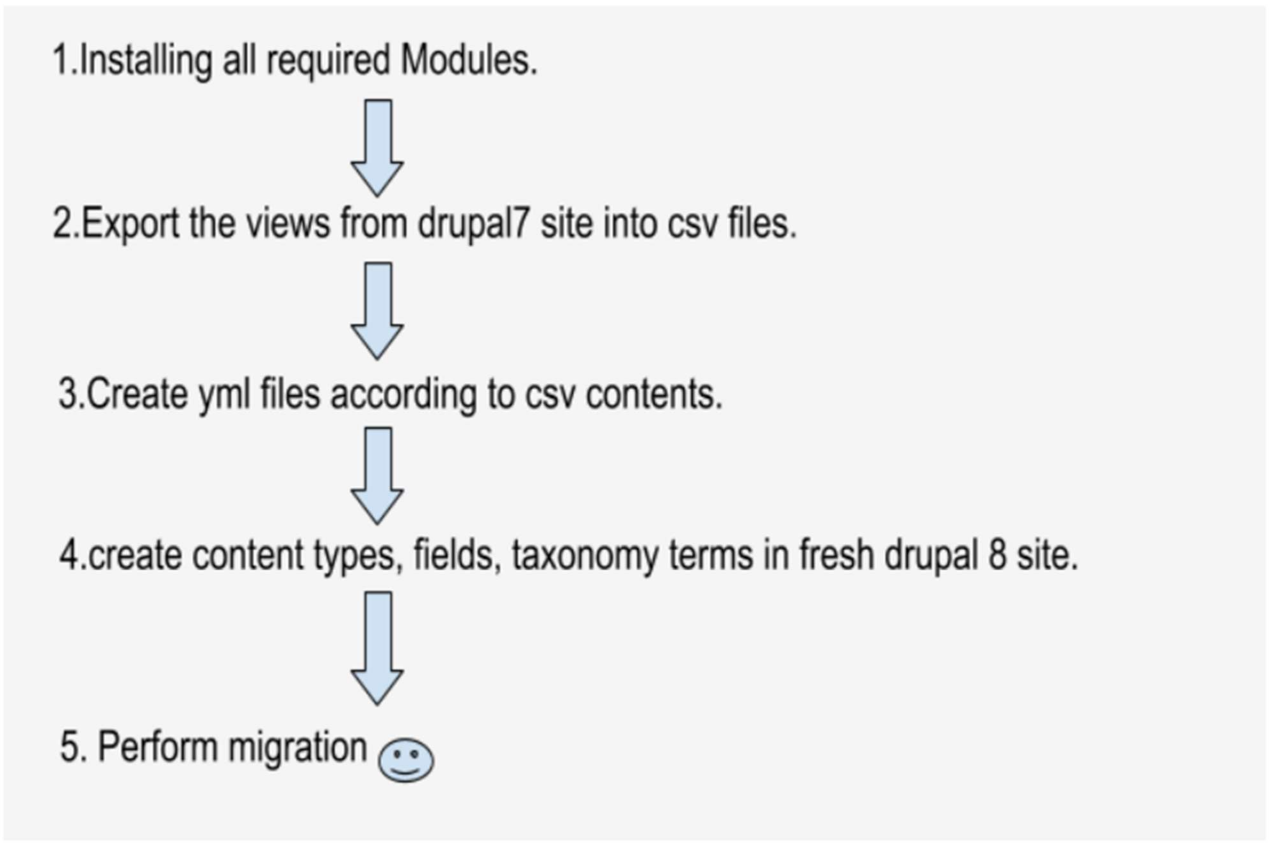
Some of the things we can create with a migration include:

Content (nodes, taxonomy, any generic entity) including any attached files and images
Content types
User accounts
Roles and permissions
Simple configuration like the site name
Complex configuration like image styles

The E-T-L process (Extract-Transform-Load)


Every migration is defined as a *migration plugin*
Source
Drupal 8
1. Extract data using *source plugin*
2. Transform data using *process plugins*
3. Load data using *destination plugin*

STEPS TO PERFORM THE MIGRATION.



1.Installing all required Modules.

2.Export the views from drupal7 site into csv files.

3.Create yml files according to csv contents.

4.create content types, fields, taxonomy terms in fresh drupal 8 site.

5. Perform migration

MODULES

A module is a set of PHP, JavaScript, and/or CSS files that extends site features and adds functionality. We can turn the features and functionality on by *installing* the module, and we can turn it off by *uninstalling* the module; before uninstalling, you may need to remove data and configuration related to the feature or functionality. Each module that is installed adds to the time needed to generate pages on your site, so it is a good idea to uninstall modules that are not needed.

Modules needed for Migration

Drupal 8 core *Migrate* module implements the general-purpose framework.
Drupal 8 core *Migrate Drupal* module builds on that foundation to provide an upgrade path from Drupal 6 and Drupal 7 to Drupal 8.

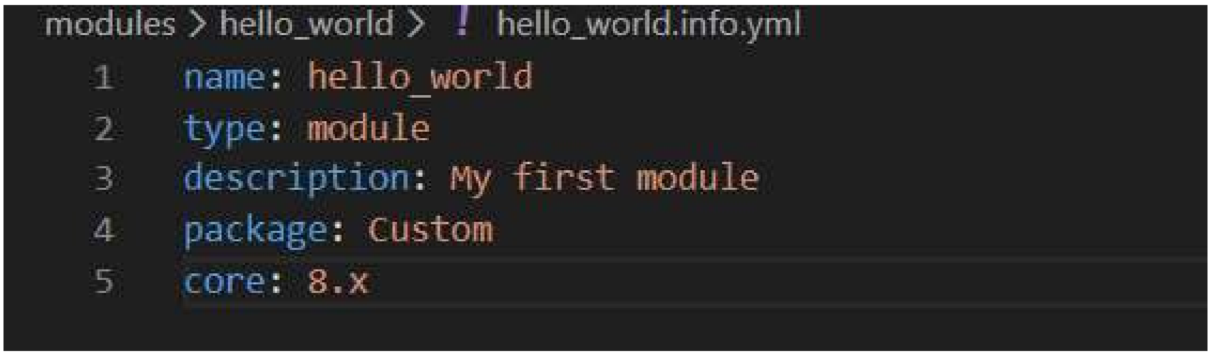
The contributed *Migrate Source CSV* module provides capabilities for migrating data to Drupal 8 from a simple CSV file.

The *Migrate_plus* project provides extensions to core migration framework functionality, as well as examples.

The *Migrate Tools* module provides tools for running and managing Drupal 8 migrations.

Custom Module

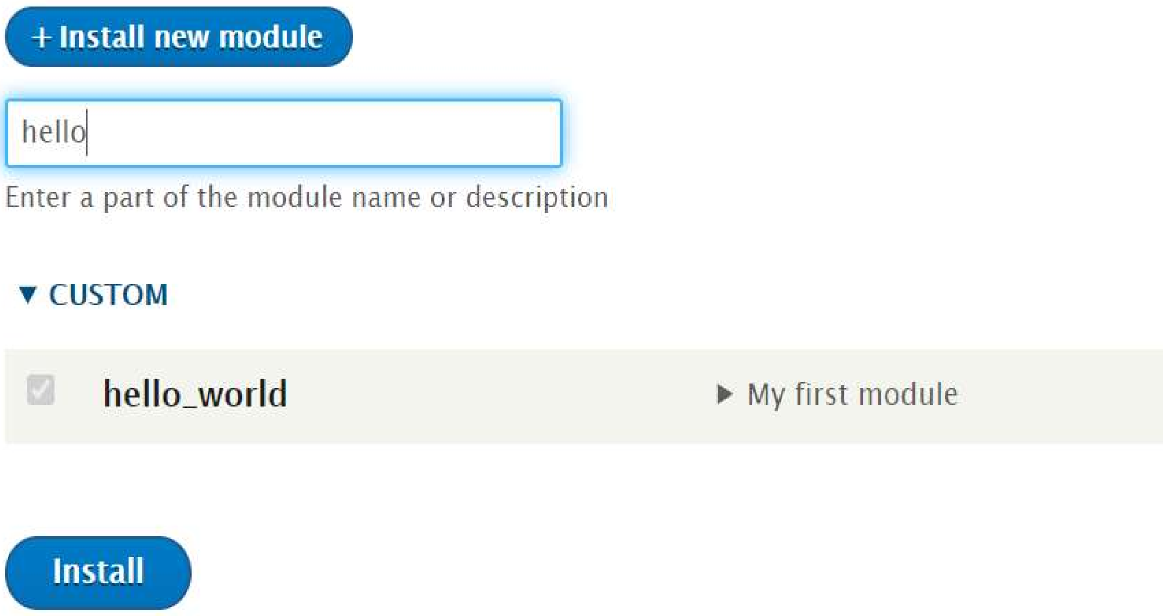
->create a new folder [module name] here, hello_world in Drupal/modules and inside that create src/plugin/migrate/process/hello_world.info.yml



modules > hello_world > ! hello_world.info.yml

```yaml
name: hello_world
type: module
description: My first module
package: Custom
core: 8.x
```

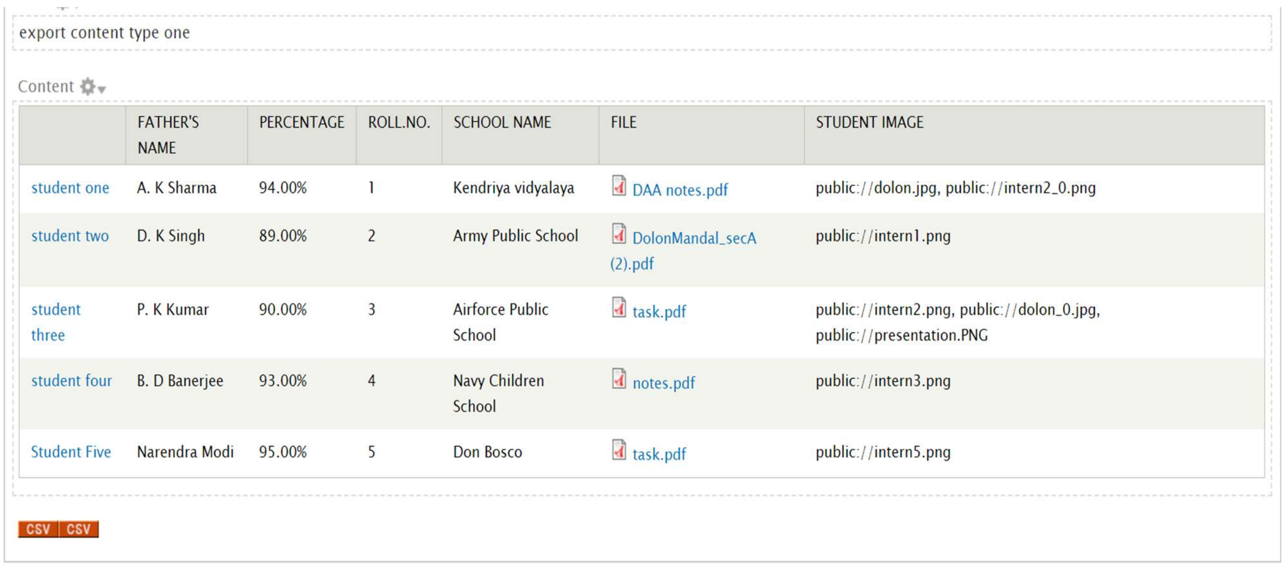
after creating it go to Drupal UI->extend->install new module->hello_world. Check the box shown.



Custom modules are required when we make custom source/process plugins (in our project we had made Explode Array plugin for use).

VIEWS

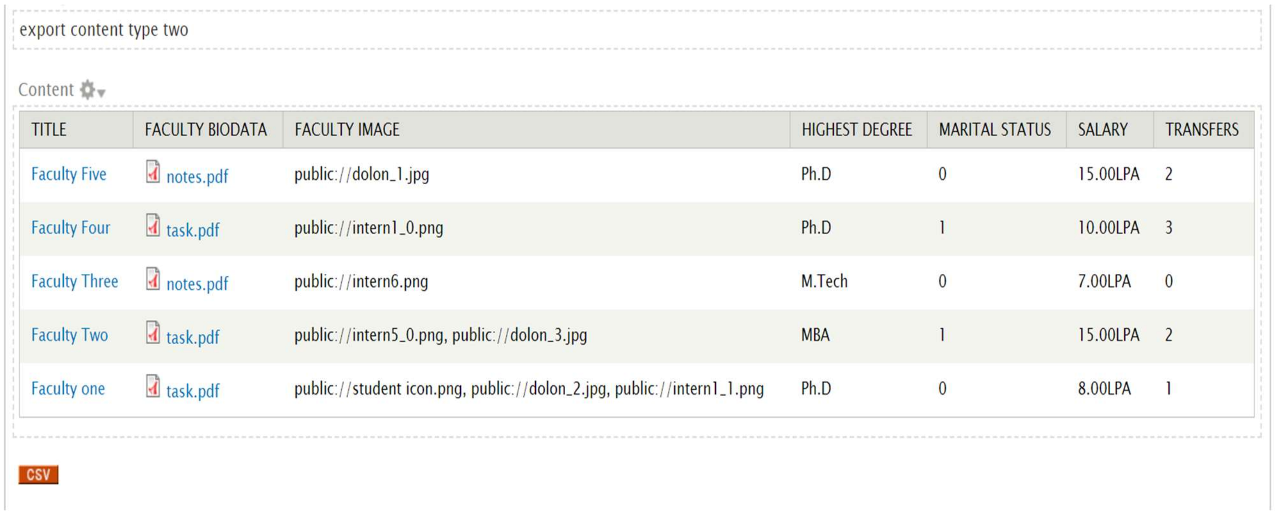
The views module allows administrators and site designers to create, manage, and display lists of content. Each list managed by the views module is known as a "view", and the output of a view is known as a "display". Displays are provided in either block or page form, and a single view may have multiple displays. Optional navigation aids, including a system path and menu item, can be set for each page-based display of a view. By default, views may be created that list content (a *Node* view type), content revisions (a *Node revisions* view type) or users (a *User* view type). A view may be restricted to members of specific user roles, and may be added, edited or deleted at the views administration page. Examples of views are: -


export content type one

Content

	FATHER'S NAME	PERCENTAGE	ROLL.NO.	SCHOOL NAME	FILE	STUDENT IMAGE
student one	A. K Sharma	94.00%	1	Kendriya vidyalaya	DAA notes.pdf	public://dolon.jpg, public://intern2_0.png
student two	D. K Singh	89.00%	2	Army Public School	DolonMandal_secA (2).pdf	public://intern1.png
student three	P. K Kumar	90.00%	3	Airforce Public School	task.pdf	public://intern2.png, public://dolon_0.jpg, public://presentation.PNG
student four	B. D Banerjee	93.00%	4	Navy Children School	notes.pdf	public://intern3.png
Student Five	Narendra Modi	95.00%	5	Don Bosco	task.pdf	public://intern5.png

CSV CSV


export content type two

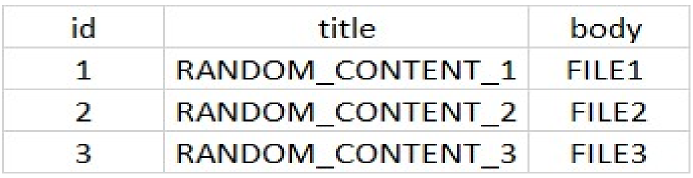
Content

TITLE	FACULTY BIODATA	FACULTY IMAGE	HIGHEST DEGREE	MARITAL STATUS	SALARY	TRANSFERS
Faculty Five	notes.pdf	public://dolon_1.jpg	Ph.D	0	15.00LPA	2
Faculty Four	task.pdf	public://intern1_0.png	Ph.D	1	10.00LPA	3
Faculty Three	notes.pdf	public://intern6.png	M.Tech	0	7.00LPA	0
Faculty Two	task.pdf	public://intern5_0.png, public://dolon_3.jpg	MBA	1	15.00LPA	2
Faculty one	task.pdf	public://student icon.png, public://dolon_2.jpg, public://intern1_1.png	Ph.D	0	8.00LPA	1

CSV

CSV

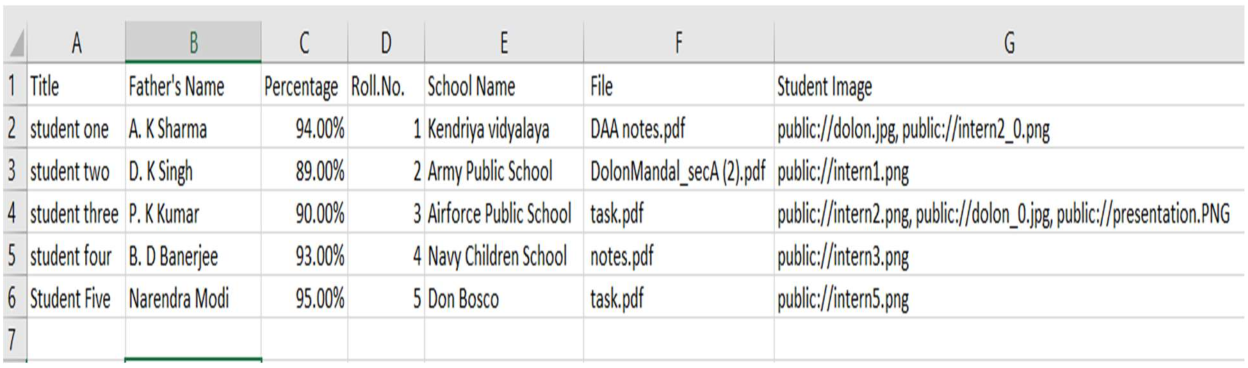
A CSV File is a Comma Separated File used to store data in a tabular format. It is an extension of a file, which may initially be in txt format or even xlsx format or any other format. The first line of a CSV File represents the header fields of the data we want to migrate. Below mentioned is a sample of a CSV File:



id	title	body
1	RANDOM_CONTENT_1	FILE1
2	RANDOM_CONTENT_2	FILE2
3	RANDOM_CONTENT_3	FILE3

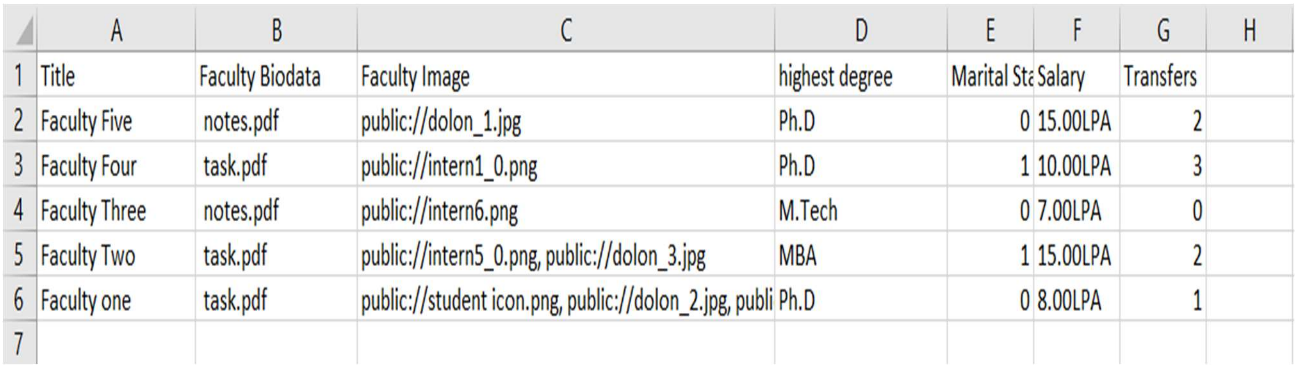
Here, the header fields are ‘id’, ‘title’ and ‘body’. This CSV File contains 3 sets of data.

The csv for above mentioned content type one is given below. We can get this by going to views ->data->export as csv file.



	A	B	C	D	E	F	G
1	Title	Father's Name	Percentage	Roll.No.	School Name	File	Student Image
2	student one	A. K Sharma	94.00%	1	Kendriya vidyalaya	DAA notes.pdf	public://dolon.jpg, public://intern2_0.png
3	student two	D. K Singh	89.00%	2	Army Public School	DolonMandal_secA (2).pdf	public://intern1.png
4	student three	P. K Kumar	90.00%	3	Airforce Public School	task.pdf	public://intern2.png, public://dolon_0.jpg, public://presentation.PNG
5	student four	B. D Banerjee	93.00%	4	Navy Children School	notes.pdf	public://intern3.png
6	Student Five	Narendra Modi	95.00%	5	Don Bosco	task.pdf	public://intern5.png
7							

The csv for above mentioned content type two is given below.



	A	B	C	D	E	F	G	H
1	Title	Faculty Biodata	Faculty Image	highest degree	Marital Sta	Salary	Transfers	
2	Faculty Five	notes.pdf	public://dolon_1.jpg	Ph.D	0	15.00LPA	2	
3	Faculty Four	task.pdf	public://intern1_0.png	Ph.D	1	10.00LPA	3	
4	Faculty Three	notes.pdf	public://intern6.png	M.Tech	0	7.00LPA	0	
5	Faculty Two	task.pdf	public://intern5_0.png, public://dolon_3.jpg	MBA	1	15.00LPA	2	
6	Faculty one	task.pdf	public://student icon.png, public://dolon_2.jpg, publi	Ph.D	0	8.00LPA	1	
7								

YML

A YML file is a specific type of file, which is an extension to the YAML file, but the Drupal adds some extra information to it (by using some more key-value pairs) to make it more elaborative.

Below mentioned is a brief description of all the key-value pairs used in this yml:

- **'id'**: This key maps to the migration id. Please note that migration id for all migrations have to be all distinct. As otherwise, the system will throw an error. For each unique migration id, Drupal assigns a unique uuid to each migration.
- **'label'**: A label is simply assigned to a migration, so that the user can identify this migration from UI. This may be the same for different migration ids.
- **'migration group'**: By default, there is a migration group named 'default', which will contain all of your migrations. But we can make your custom migration group and use it for specific migration. In such cases, we have to write the exact machine name for this key-value pair.

The later section of the yml is further divided into 3 categories:

1. Source
2. Process
3. Destination

Here is a brief about all the 3 subsections and the associated terms:

- **Source**

This subsection contains information about the source of the data from where the data has to be migrated.

1. **'**plugin'**:** In this case this key is mapped to 'csv', as we will be extracting the data from a CSV file. It could also be set to 'embedded data', in which we simply mention all of the data to be migrated in the yml itself.
2. **'**path': This contains the absolute path of the CSV file.

3. ‘delimiter’: This contains a specific character (generally ‘,’), which is used to help distinguish Drupal between 2 adjacent fields.
4. ‘enclosure’: An enclosure is set to “” (double quotes), so that it by default encloses all the values mentioned in the CSV with double quotes. It can’t be set to single quotes (Drupal throws an error).
5. **‘**header_offset’: This key is mostly set to 0, as this describes the gap between the 1st line and the line containing the header fields in the CSV file.
6. ‘ids’: The above mentioned syntax for this key has to be followed consistently, as it makes an array of all the data sets mentioned in the CSV and processes them one-by-one.

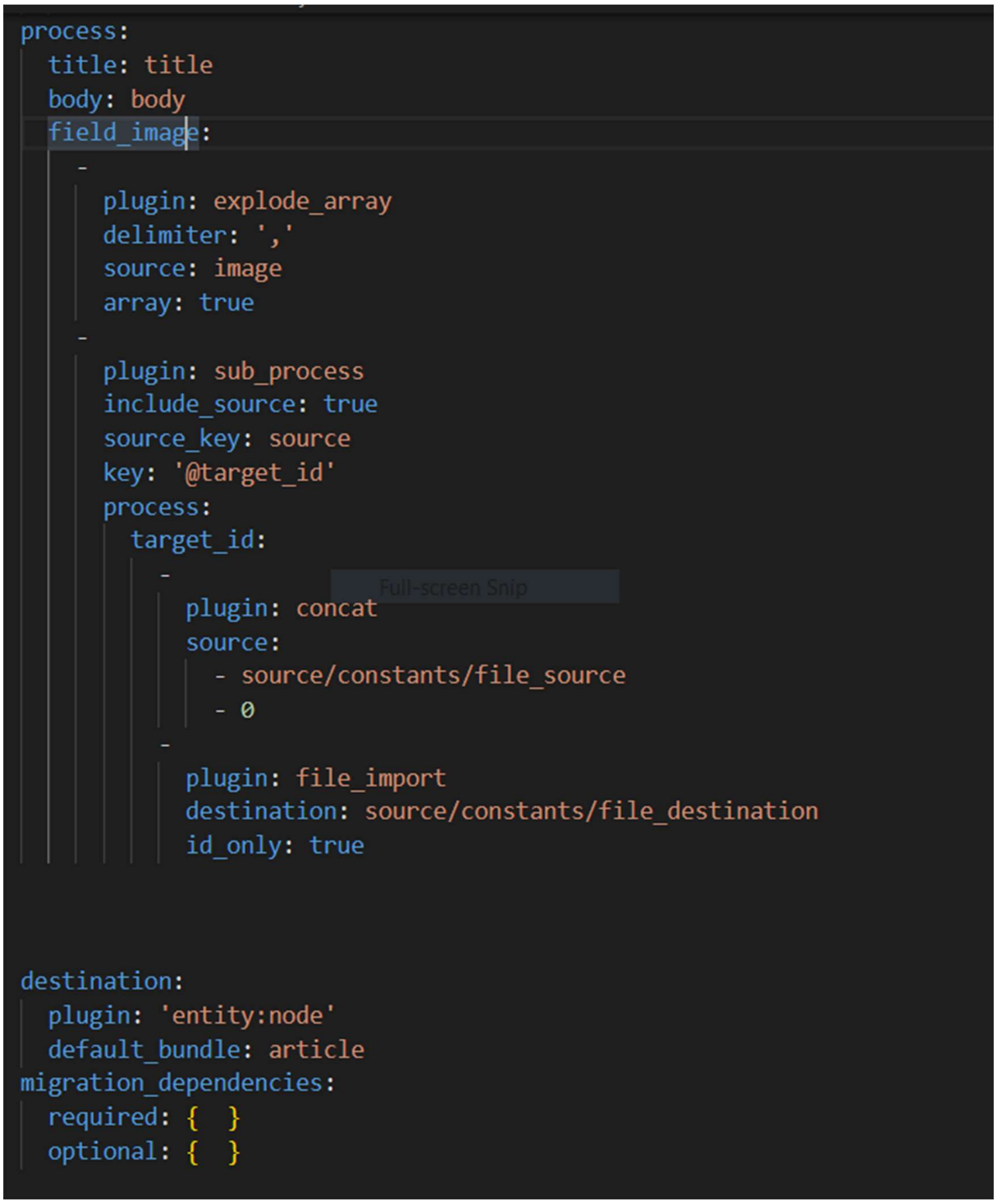
- **Process**

This subsection maps the data of the header fields to the corresponding destinations(fields) in your project. Thus, the name of each header field of the CSV file is mapped to the corresponding machine name of the field where the data is supposed to be injected.

- **Destination**

This part of the YAML does not consist much but generally a ‘plugin: entity’ which is set to ‘node’ for a content type. This is so, because all the migrations aimed at importing data for a content type are inserted in new ‘nodes. This key can also be set to ‘user’ for user migration.

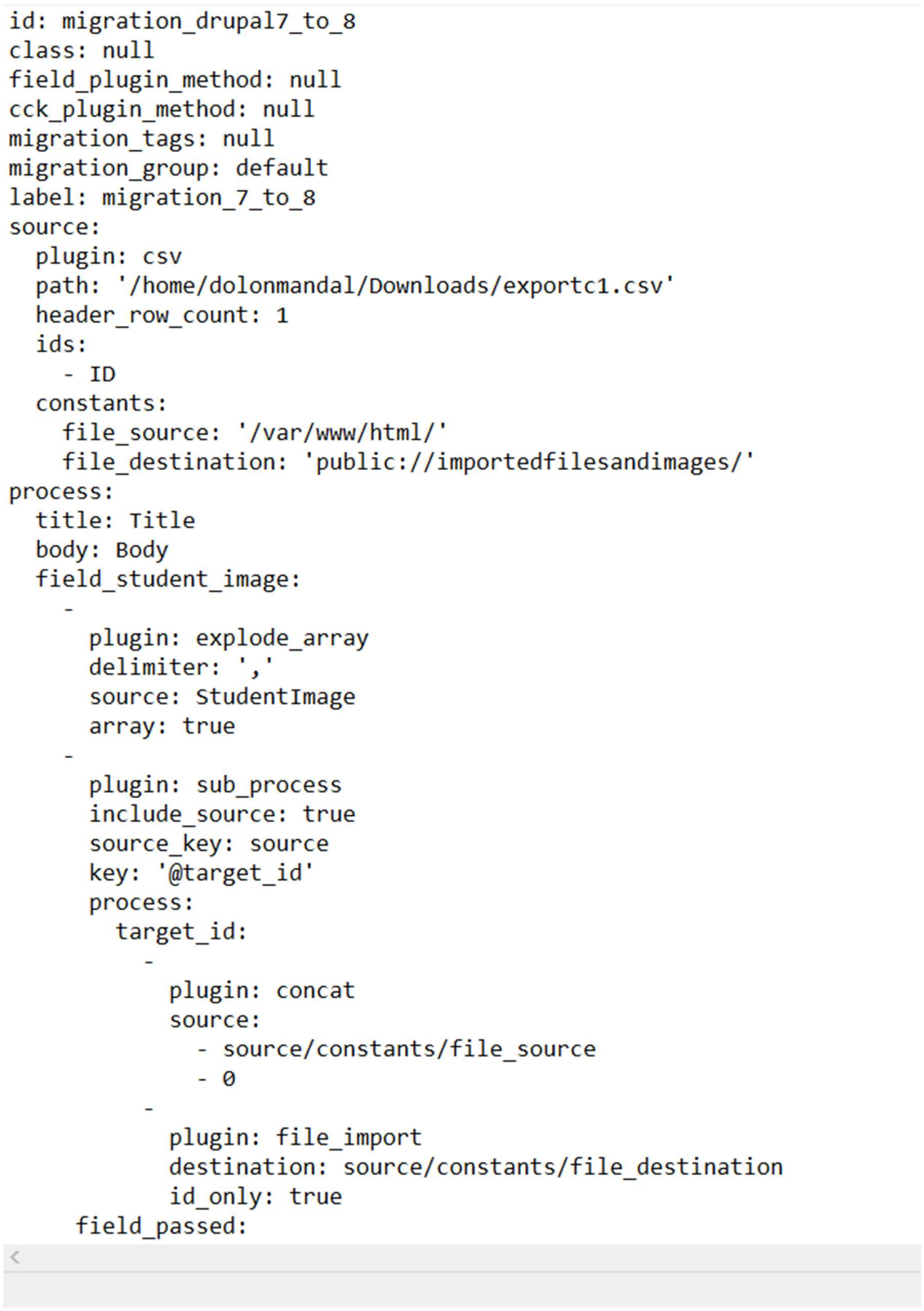
For image and file importing, we have to install a separate module ‘**Migrate Files (extended)’**. This https://www.drupal.org/project/migrate_file contains essential information about this module. The below mentioned YAML is a sample of the one used for image and file import.



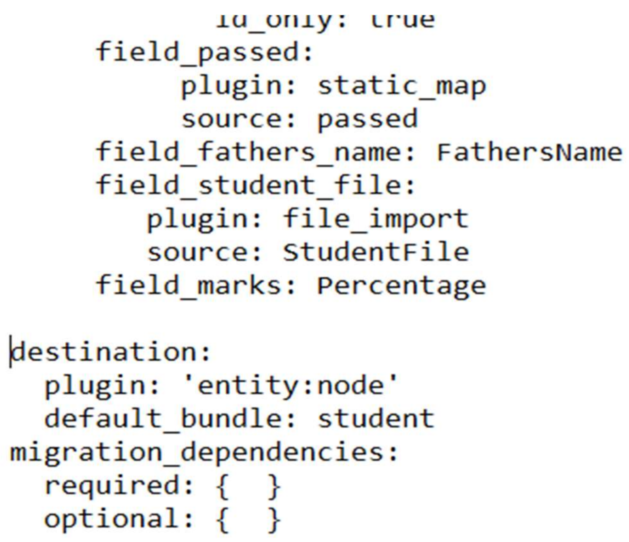
```yaml
process:
  title: title
  body: body
  field_image:
    -
      plugin: explode_array
      delimiter: ','
      source: image
      array: true
    -
      plugin: sub_process
      include_source: true
      source_key: source
      key: '@target_id'
      process:
        target_id:
          -
            plugin: concat
            source:
              - source/constants/file_source
              - 0
          -
            plugin: file_import
            destination: source/constants/file_destination
            id_only: true



destination:
  plugin: 'entity:node'
  default_bundle: article
migration_dependencies:
  required: {  }
  optional: {  }
```

For multiple image migration plugins required are explode_array(custom plugin), sub-process plugin, Concat plugin , file-import plugin.

Below is the complete yml for the very first example of content type one migration whose view and csv is mentioned above:



```yaml
id: migration_drupal7_to_8
class: null
field_plugin_method: null
cck_plugin_method: null
migration_tags: null
migration_group: default
label: migration_7_to_8
source:
  plugin: csv
  path: '/home/dolonmandal/Downloads/exportc1.csv'
  header_row_count: 1
  ids:
    - ID
  constants:
    file_source: '/var/www/html/'
    file_destination: 'public://importedfilesandimages/'
process:
  title: Title
  body: Body
  field_student_image:
    -
      plugin: explode_array
      delimiter: ','
      source: StudentImage
      array: true
    -
      plugin: sub_process
      include_source: true
      source_key: source
      key: '@target_id'
      process:
        target_id:
          -
            plugin: concat
            source:
              - source/constants/file_source
              - 0
          -
            plugin: file_import
            destination: source/constants/file_destination
            id_only: true
     field_passed:
```



```
            id_only: true
      field_passed:
           plugin: static_map
           source: passed
      field_fathers_name: FathersName
      field_student_file:
         plugin: file_import
         source: StudentFile
      field_marks: Percentage

destination:
  plugin: 'entity:node'
  default_bundle: student
migration_dependencies:
  required: {  }
  optional: {  }
```

Perform Migration.

- Install fresh Drupal 8 site
- Create content type with fields given in csv
- go to ->structure -> migration -> create migration group -> name it default.
- go to -> configuration ->configuration synchronisation -> import -> single import-> paste the migration -> submit.
- go to structures -> migration -> default -> select migration id -> import.
- The content type will be filled with contents present in the csv file.
- If due to some fault migration does not take place rollback, delete, reset and then import again.

PLUGINS

Plugins are small pieces of functionality that are swappable. Plugins that perform similar functionality are of the same plugin type.
Drupal contains many different plugins, of different types. For example, 'Field widget' is a plugin type, and each different field widget type is a plugin. The admin user may select from the list of field widget plugins to set the widget that a field uses.

The D8 plugin system provides a set of guidelines and reusable code components to allow developers to expose pluggable components within their code and (as needed) support managing these components through the user interface.

Plugins are defined by modules: a module may provide plugins of different types, and different modules may provide their own plugins of a particular type.

Some modules which we have used in our yml are:

Sub-process Plugin:

This plugin is very essential when you want to import multiple data in a specific multi-value field in a content type. Enabling this plugin iterates over all the data of a specific field and migrates them one-by-one.

Explode Plugin:

This plugin is again important for importing data in a multi-value field. This plugin creates an array of strings by splitting the source parameter on boundaries formed by the delimiter.

Concat Plugin:

The function of this plugin is to concatenate strings which previously have been separated (for example, used to obtain a path of source of file by forming a connected string of directories and subdirectories).

COMPOSER

Composer is a dependency manager for PHP. Drupal core uses Composer to manage core dependencies like Symfony components and Guzzle.
There are many benefits to using Composer. In short, it allows us to systematically manage a sprawling list of dependencies (and their subsidiary dependencies). It assists with locating, downloading, validating, and loading said packages, all while ensuring that exactly the right versions for each package are used.

Some modules such as the migrate_source_csv module has some dependencies which needs to be installed composer. Run command

Composer require “Drupal/migrate_source_csv” from the root directory of Drupal site from terminal.

Install composer from https://getcomposer.org/Composer-Setup.exe and run the installer.

References

1. Drupal official website https://www.drupal.org/
2. OS Training Videos
3. Spoken Tutorials