

**Avanti Alumni Network**

**Force for Good: 2020-21**

**Technical Design Document**

**JP Morgan Chase & Co.**

Table of Contents

[Revisions 3](#_Toc79021494)

[Definitions 4](#_Toc79021495)

[Overview 5](#_Toc79021496)

[Technology Stack 5](#_Toc79021497)

[End to End data flow 5](#_Toc79021498)

[Applications 6](#_Toc79021499)

[Connecting AWS 7](#_Toc79021500)

[Connecting to AWS VM GUI: 7](#_Toc79021501)

[Python Scripts 8](#_Toc79021502)

[Installation and Cron Set up 8](#_Toc79021503)

[Python Script Configurations: 11](#_Toc79021504)

[Drupal Application 12](#_Toc79021505)

[Installations 12](#_Toc79021506)

[Authorization and Authentications 13](#_Toc79021507)

[Drupal site contents 14](#_Toc79021508)

[Drupal Themes 14](#_Toc79021509)

[Feeds Module used in Drupal 16](#_Toc79021510)

[Contact Us form in Drupal: 17](#_Toc79021511)

[Contact Details in Drupal 18](#_Toc79021512)

[Alumni Dashboard (Tabular View) 18](#_Toc79021513)

# Revisions

|  |  |  |
| --- | --- | --- |
| Author | Version and Date | Change(s) |
| Pranaya Ranjan Behera | Version- 1.0.0  Date: 2021-07-20 | Initial Documentation |
|  |  |  |
|  |  |  |
|  |  |  |

# Definitions

|  |  |
| --- | --- |
| **Terms** | **Meaning** |
| Master Sheet | Alumni-Data Tab of Google sheet <https://docs.google.com/spreadsheets/u/1/d/17hXCfv6Q8qfrw9aMJ-ixp0OLTBgSH-4IaGCsSmMQQhY/edit#gid=1468372192> |
| LinkedIn-Master sheet/ Combined data sheet | Master-LinkedIn-Data sheet of  <https://docs.google.com/spreadsheets/d/1wyBuwamj7SQZ9kl1yAecWutC-6UI_DNl1dbZzslEEFY/edit#gid=0> |
| Control Sheet | Control-sheet Tab of  <https://docs.google.com/spreadsheets/d/1wyBuwamj7SQZ9kl1yAecWutC-6UI_DNl1dbZzslEEFY/edit#gid=0> |
| Charts sheet | Charts Tab of  <https://docs.google.com/spreadsheets/d/1wyBuwamj7SQZ9kl1yAecWutC-6UI_DNl1dbZzslEEFY/edit#gid=0> |
| LinkedIn data sheet | LinkedIn-Data sheet of  <https://docs.google.com/spreadsheets/d/1wyBuwamj7SQZ9kl1yAecWutC-6UI_DNl1dbZzslEEFY/edit#gid=0> |
| LinkedIn Page/  AAN LinkedIn Page | <https://www.linkedin.com/school/avanti-alumni-network/people/> |
| Portal/  AAN Portal/  Drupal Portal | Site at this address:  <http://13.126.43.196/> |
| Alumni info git-hub/  github | <https://github.com/avantifellows/alumni_info/> |
| Readme.md/  README.md/  Readme page | <https://github.com/avantifellows/alumni_info/#readme> |
| AAN | Avanti Alumni Network |
| AAN Admins/  Avanti Admins | Users having Admin Role to AAN Portal, Read Write access to Google Sheets |
| Python Script Path | /home/ec2-user/web\_scraping/src/main |
| cfg.yaml path | /home/ec2-user/web\_scraping/src/conf |
| Chromedriver path | /home/ec2-user/web\_scraping/ |
| Chrome driver | Down load required one from this link based on OS and Chrome version:  <https://chromedriver.chromium.org/downloads> |

# Overview

Avanti Fellow is a non-profit organization helping students from low-income backgrounds, to have equitable access to high-quality colleges and professional growth that will create a better life and future for them. They are maintaining their student details in Google sheet.

We need to create a portal to view and manage the students and with ability to fetch students details update from their LinkedIn as well as the master Google sheet that they were using.

# Technology Stack

* Python for Web-scraping
* Google Sheet as their another student data source, Charts and Graphs
* Drupal as CMS
* Maria DB
* Apache Web Server
* AWS Hosting Environment

# End to End data flow

There are two data source for AAN portal, viz : master sheet and AAN Linked In page. Master sheet contains students information collected offline and filled manually by Avanti Admins. Avanti Admins ask their students to create linked in page and become member of AAN LinkedIn page. We have a python web scraping script which will fetch the AAN LinkedIn page member’s public information and store it in Linked in data sheet and LinkedIn master sheet. LinkedIn master sheet contains combined data of Master sheet and LinkedIn data sheet. In case of a record exist in both the sheet, LinkedIn data sheet takes the precedence and override the master sheet data.

LinkedIn-Master sheet is source for the AAN portals student’s details. LinkedIn-Master sheet student details are read by portal’s feed service periodically and updated the ANN portal’s student details. In case a new student details are added in master sheet or any student becomes member of LinkedIn page, then details will be available in LinkedIn-Master sheet once a python web scraping completed. After that in next feed processing, these student details will be synched to the portal.

Python web scrapping frequency and portal feed frequency are different. We will discuss more about this in later section.

|  |
| --- |
|  |

(Figure 1.1: Data Flow)

# Applications

When we talk about application, Python Scripts and Drupal Portal setup comes in to picture. Both are independent as point of installation and run, LinkedIn-Master Google sheet intermediary which keeps both connected. Python Scripts web scraps the LinkedIn profiles, reads master sheet data, merge both and save it in LinkedIn-Master sheet, whereas Drupal portals only reads updated data from LinkedIn-Master sheet and keep portal updated with students details update. So Key components here are : Python Scripts, Google sheets and Drupal portal.

## Connecting AWS

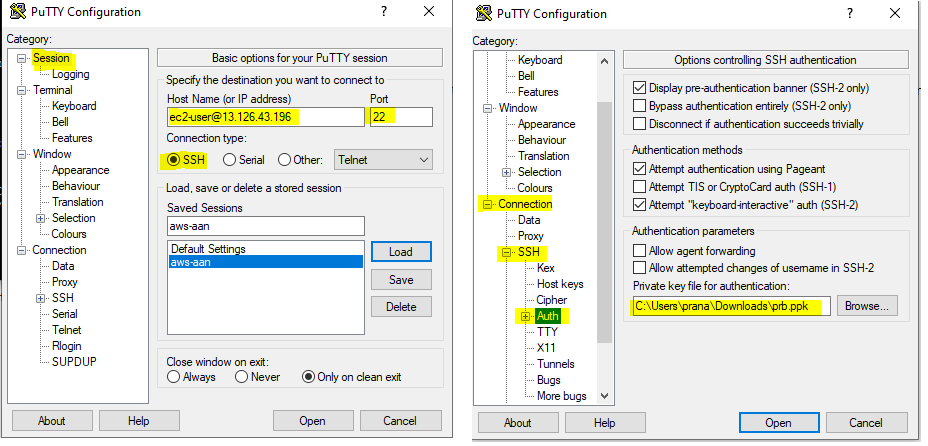
AWS EC2 instance is the hosting environment for AAN portal. Both Drupal portal and python scripts runs here. To connect to EC2 instance, you need PEM file or credential details.

Commands to connect via pem file :

* *sudo chmod 400 AvantiFFG.pem*
* *ssh -i AvantiFFG.pem*[*ec2-user@13.126.43.196*](mailto:ec2-user@13.126.43.196)

Here *AvantiFFG.pem*is the **.pem** file. Avanti Fellows team has this file.

To connect to AWS Unix box from Windows OS, You need putty and need to generate **.ppk** file from .pem file. Look at this link how to convert .ppk file from .pem file. <https://aws.amazon.com/premiumsupport/knowledge-center/convert-pem-file-into-ppk/>



### Connecting to AWS VM GUI:

There are several ways to connect VM GUI. Here we had used tigerVNC for this purpose from MAC or UNIX based OS.

**Steps to connect** (For Mac, We used DMG) **:**

1. Download appropriate file for installing tigervnc from - [**https://sourceforge.net/projects/tigervnc/files/stable/1.11.0/**](https://sourceforge.net/projects/tigervnc/files/stable/1.11.0/)
2. ssh using this command - **ssh -L 5901:localhost:5901 -i AvantiFFG.pem**[**ec2-user@13.126.43.196**](mailto:ec2-user@13.126.43.196)
3. Start tigerVNC in your local computer (That's like your remote desktop client). In VNC server mention **localhost:1**

Password - **ffgavantifellows123**

The desktop environment should show up.

## Python Scripts

Python script code is available in Alumni info github. It has one main script and several supporting scripts. We need to run ***main.py*** once or multiple times in a day. It will do task bases on the control sheets data. This ***main.py*** is responsible for doing all kind of works we need in its multiple runs. In First run it will check when web scrapping was done last time. We have a configuration property, in how many days, web scrapping should happen like monthly once or weekly once. If last web scrapping was had started within the specified day’s gap, then it will not do anything, program will simply exit. In next schedule run, it will check again. If it finds last web scrape started before the specified gap, it will fetch the all linked in members name and linked in profile link in that date and update in control sheet. In next run, it will fetch (web-scrape) all the required public details of 40 or configured number of profiles and update it in control sheet. In next runs, it will fetch all the required public details of next 40 or configured number of profiles and update it in control sheet. This process continues until it web scrape for all profiles and update it in control sheet. Once all profile details are web scraped, it will read student details from Master sheet and web scrape details from control sheet, and merges (if one record present in both, then LinkedIn information will take precedence) both the data and update (remove all entries and then add all) LinkedIn web-scrape data in LinkedIn sheet and combined merged data in LinkedIn-Master sheet. Now LinkedIn-Master sheet is ready with updated information to be consumed by Drupal portal’s feed service and get synced.

Python program has only read access to Master sheet and both read and write access to Alumni sheets.

### Installation and Cron Set up

* Install chrome browser in AWS box first, try to access Google sheet, check connections.
* Move python script src package to AWS box root folder.
* Install Python 3.6 or higher and check version after installation. It should be 3.6 or higher
* sudo yum install python36
* python --version
* Install supporting packages:
* pip install -U selenium
* pip install pandas
* pip install PyYAML
* pip install pygsheets

|  |
| --- |
|  |

(Figure 2.1: Web scrapping logic Flow Chart)

* Download( and unzip ) ChromeDriver in desired location if not there:

wget <https://chromedriver.storage.googleapis.com/2.37/chromedriver_linux64.zip>

unzip chromedriver\_linux64.zip

sudo mv chromedriver /usr/bin/chromedriver

chromedriver --version

* Install Google Chrome for ChromeDriver to work in not installed by this or by other command:

curl https://intoli.com/install-google-chrome.sh | bash

sudo mv /usr/bin/google-chrome-stable /usr/bin/google-chrome

google-chrome --version && which google-chrome

* Incase pip install does not work; install pip or pip3, then install above. Following is the steps:
* curl -O https://bootstrap.pypa.io/get-pip.py
* python3 get-pip.py --user
* pip3 --version
* For first time run, run it as *headless = false* mode, by modifying in cfg.yaml and run from terminal
* Verify first time run both from console logs as well as from LinkedIn screen. Observe Chrome driver logged in screen, if any security questions. If any security question, then it has to be resolved manually.
* Once manual check done set up cron, Following are the steps:
* Check all files by *ls* command.
* Run a test file : python test.py , and verify before pointing *main.py*.
* Setup a cron job to automatically run python scripts at a specified interval or time in your ec2 instance:
* Start the cron service sudo service crond start
* Edit the crontab file using crontab –e
* You can restart after cron edit: sudo service crond restart
* Add the file path of the file you want to run. Hit the “i” key to start editing the file and then add 0 6 \* \* \* python *path/to*/main.py . Make sure you give the full path to the python script that you want to run.

Above cron expression is for 6am everyday

* Commands to verify cron set up : crontab –l
* Command to check cron status: sudo service crond status
* To find the cron has run execute the below command: nano /var/spool/mail/ec2-user

* Do a test and monitor with cron set up if cron set up is working. Initially you can set up cron for every 30min and later change to desired one.

### Python Script Configurations:

Following configurations has to be understood by Avanti Admin and Tech team. Details are at: <https://github.com/avantifellows/alumni_info/blob/main/src/conf/cfg.yaml>

|  |  |
| --- | --- |
| **Property Name** | **Description and values** |
| login: email | Email id used for Logging in to LinkedIn to web scrape the profiles  Current value : ffgadmin@avantifellows.org |
| login: password | LinkedIn Password for above email id |
| config: headless | Chrome driver’s headless = True/False mode  Current value : True |
| config: ws\_schedule\_days\_diff | Minimum number of days after which next web scrape should happen  Current value : 7 |
| linkedin: num.of.profiles | Maximum number of profile to be web scrapped in a run  Current value : 40 |
|  |  |
|  |  |

## Drupal Application

Drupal is a powerful open source CMS used for dynamic website creation. It requires PHP, Web server (Apache server or Nginx or other), Database (MariaDB or MySql). Several supported plugging can be installed in Drupal Portal to provide additional features. All these have to be installed and hosted somewhere, in our case it is AWS EC2 instance.

### Installations

Follow the steps mentioned in the below documentation to install Apache, MariaDB and PHP on EC2 Instance. (Installed mariadb10.5 and php8.0 which is supported for Drupal 9 instead of lamp-mariadb10.2-php7.2 and php7.2)

Following are the Software to be installed and set up sequentially:

* Install PHP
* Install Apache-Server
* Install MariaDB and create database for Drupal application and grant privileges
* Install Drupal

We have referred following link for above installation: <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-lamp-amazon-linux-2.html>

sudo amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2

sudo yum install -y httpd mariadb-server

sudo systemctl start httpd

Installation of Drupal and configuration setup followed as per the below documentation: <https://aws.amazon.com/getting-started/hands-on/deploy-drupal-with-amazon-rds/4/>

wget https://www.drupal.org/download-latest/tar.gz

tar -xzf tar.gz

mv drupal-\* Drupal

drupal tar.gz

cd drupal

sudo rsync -avz . /var/www/html

sudo chown -R apache:apache /var/www/html

sudo service httpd restart

Enter Public DNS(IPv4) value into your web browser. You should set the beginning of the Drupal setup wizard GUI and install.

Created Database and Drupal DB user in MariaDB before Drupal installation (root and drupaluser password is shared separately):

|  |
| --- |
|  |

#### Installation Details

|  |
| --- |
| Drupal version: 9.1.8  Database type: Maria DB  Database name: drupaldb  Database username: drupaluser  Advanced Options not required as RDS is not used.  Site email address: vishesh@avantifellows.org  Site maintenance account details shared separately.  Drupal Avanti Alumni Portal site configured on http://13.126.43.196/ |

Website Admin password is sent separately. After successful installation, AAN portal site can be accessed by above URL.

### Authorization and Authentications

Admin can add user, edit any user’s role, permission, block, unblock users and even reset their passwords. Apart from that, anyone can sign up at AAN portal but admin has to approve their request. Their by default role will be Logged in User.

Anyone can access AAN portal but they can view limited pages as per page role base access configuration set by Admin.

Drupal provides its role base access for authorizations for its content.

Feed service is set up to import users (add if user does not exist, update its attributes if already exist in Drupal) from Google sheet periodically. New users will be imported as blocked users without password (one time only, in subsequent update password and active/blocked status will be unaffected).

#### Administrator’s one time activity for new imported users

|  |
| --- |
| Admin has to access each newly imported profile once, should set user status active from blocked and set temporary password for the user and send the user credentials to user via email or message or by any mean. This is a onetime activity when a new user is imported. |

### Drupal site contents

Any content in Drupal site can be added/modified by users with Admin and Moderator Role. Also users with Editorial board Role have access add/modify some limited content. Public and Logged in user does not have any access to add/modify any content but they can contact Admin by contact us form.

Refer Drupal User Guide and Drupal Documentation for details.

User Guide: <https://www.drupal.org/docs/user_guide/en/index.html>

Documentation: <https://www.drupal.org/documentation>

### Drupal Themes

The look and feel of the entire portal is controlled using a Drupal theme. Changing a theme will affect the appearance of all the screens (including grids, input forms, menu bar etc). Each theme provides its own set of features and privileges. The current theme is set to **Zuvi 8.1.6** (unpaid version)

Documentation - <https://www.drupar.com/zuvi-documentation>

Official website - <https://www.drupar.com/theme/zuvi>

**Steps to install a theme**

* Find a suitable theme on Drupal’s official website (use filters for the best match)
  + <https://www.drupal.org/project/project_theme>
* Drupal gives an option to download the themepack as tar.gz or zipped file. **Copy the link to download the file as tar.gz**
* Login to the portal as an admin
* **Appearance > Install new theme > Install from a URL**
* Paste the copied URL (tar.tz file) and click **Install**. This step adds the theme to Drupal.
* Once the above steps are completed, go back to the Appearance menu, and find the newly added theme in the “**Uninstalled Themes**” section at the bottom
* To install the theme completely, click on the install option provided below the newly added theme.

**Steps to set a theme**

* Login to the portal as an admin
* Go to the “**Appearance**” menu and find a suitable theme under the ‘Installed themes’ section. The theme must be installed and present under the ‘Installed themes’ section before using it.
* Click on “**Set as Default**” to use the theme.

**Theme configurations**

* Login to the portal as an admin
* Go to the “**Appearance**” menu and find a suitable theme under the ‘Installed themes’ section.
* Click on “**Settings**” to configure the theme.
* Make the required changes and click **Save Configuration**.
* To find the block divisions (page skeleton) provided by the theme :
  + Structure menu > Block layout > Choose the current theme > Demonstrate block regions (<theme\_name>)

|  |
| --- |
| Feeds Module used in Drupal We are using the following feeds module for importing the data from the Google spreadsheet in the csv format into Drupal: <https://www.drupal.org/project/feeds/releases/8.x-3.0-alpha10>  **Location:**  Manage > Extend > Feeds > Feeds  **Steps to install any module:**   * Go to Extend * Install new module * Install from URL: Give the link to the module. Eg: https://ftp.drupal.org/files/projects/feeds-8.x-3.0-alpha10.tar.gz   **Wiki link for the complete feeds setup:**  <https://www.drupal.org/docs/contributed-modules/feeds/creating-and-editing-import-feeds>  **For importing the csv data as *users* into drupal**   * **Create a feed type:** We’ve created a feed type as “Avanti\_alumni\_final” with required configurations as can be found on the site   **Location:** Manage > Structure > Feed types > Avanti\_alumni\_final  **To edit the configurations:** click on Edit   * **Edit:** For the basic settings including the fetcher and parser settings. * **Mapping:** For the headers on the spreadsheet being mapped onto user profile attributes. * **Manage Fields**: No changes being made. * **Manage Form Display:** No changes being made. * **Manage Display:** No changes being made. * **To setup the cron:**   Manage > Configuration > System > Cron > Cron Settings > “Run cron every 1 week” > Save configuration  To manually run the cron job click “Run Cron”   * **To add the content for users**: Manage > Content > Feeds > “Alumni Details of Avanti”   Click on “Edit” to edit the configurations:  Enter the Feed URL: <https://docs.google.com/spreadsheets/d/e/2PACX-1vQp4O1OYsgvAUnmgQIEEFYDPxCRZnJkBjg_SXnZXeydsxmjraZVdJSQFauYIuJf2H5gutY32ESLW40L/pub?gid=1752137495&single=true&output=csv>  “Save and import”  To view all the imported users: Click on the “Items” tab   * **After the users have been imported:**   The users are imported as blocked, with no passwords. To activate the users:  Go to: Manage > People > List  The admin can “unblock” the selected users from the list from the “Action” dropdown bar.  For providing passwords, click on individual users’ “Edit” tab, and add the password.  **For importing the csv data as *nodes* into drupal for the Alumni Dashboard**  The steps are quite similar with the csv data being imported as “nodes” instead of “users”.   * **Create a Feed Type**: “Avanti\_alumni\_node” is used as Feed type. * **To add Content Type:** Manage > Structure > Content types > “Avanti\_alumni”   Click on “Manage fields” to change the config.   * **To add content for nodes**: “Avanti Alumni Grid” is used as Feed content with content type being “Avanti\_alumni” * **After the nodes have been imported**:   The content can be viewed from Manage > Content > list of nodes of students imported.  The only reason the data is again imported as “nodes'' is that the grid/tabular view of data in the Alumni Dashboard is not feasible with the “users” format. This import will no way affect the users on the site.  **To search the users:**   * Click on the search button in the navigation bar. * Enter the name of the user to be searched in the search box. * Go to the “Users” tab and click on the required user from the list.   **To Edit the content of the users:**  The admin and the students both can change their information on the site. For the student to login and edit their info, the admin first needs to unblock the user and provide them with a password. They can go to the “Edit” tab and change almost all the attributes and click on “Save”.  **To add additional fields/attributes to the user’s profile page:**  Manage > Configuration > People > Account Settings > Manage Fields > Add field  *Reports tab under Manage can be used for checking the updates and log messages on Drupal.* |

### Contact Us form in Drupal:

Current setup uses the “**Contact Us**” module for generating “Queries and Feedback” form.

<https://www.drupal.org/project/contact_us>

Location in Drupal : **Extend > Contact > Contact Us**

**Manage fields in the contact us form**

* Login to Drupal as an admin
* **Structure > Content Types > Contact Us Form > Manage Fields**
* Make the required modifications like adding extra fields, changing data types, removing fields etc. and save the configuration

**View responses captured by the form**

* Login to the portal as an admin
* Go to the **Content** menu
* In the filters provided, choose “**Content Type”** as “**Contact Us Form**” and filter the results
* Click on the “Contact Us Form” link under the **Title** column to view the responses.

### Contact Details in Drupal

A custom block called “**contact us**” has been added to display the contact details. Currently this block is placed in the “Footer first” section of all the pages.

**Structure > Block Layout > Footer first > Contact Us**

**Edit the contents of this contact block**

* Login to the portal as an admin
* **Structure > Block Layout > Custom block library (tab) > Contact Us > Edit**
* Make the required changes and click **Save**

### Alumni Dashboard (Tabular View)

We have created a separate page called ***Avanti Alumni*** which contains basic information about all the users (alumni) of the portal in a single table. This dashboard is currently accessible to all the authenticated users, but if required, the access level can be changed by the admin. The table currently contains data like the name, email address, job details etc.

The key features of this dashboard are as follows:

* A user can click on the alumnus’s name in the table to view his complete profile i.e. his skills, job location, LinkedIn profile URL, interests etc.
* Currently, the entire grid can be sorted based on company or job designation. Furthermore, admin has rights to add other columns in the grid and control their sort-ability.

To create this table, we have used the ***DataTables*** module available for Drupal 9.

<https://www.drupal.org/project/datatables>

**Edit/Configure the dashboard**

* Log in to the portal as an admin
* **Structure > Views > Avanti Alumni > Edit**
  + To add fields, click on ***Add*** button under the *FIELDS* section
  + To change the settings, click on ***Settings*** link under the *FORMAT* section