Summer Fellowship Report

On

A Drupal Website for Library : Layout and Automation

Submitted by

Mukta Girish Deshpande

Under the guidance of

Prof. Albert Sunny
Department of Computer Science and Engineering
Indian Institute of Technology Palakkad

July 23, 2020

Acknowledgment

This report has been prepared for the fellowship that has been done in Indian Institute of Technology Palakkad (Remote Mode) in order to study the practical applications of Drupal in real fields with the purpose of learning and exploring as well as for professional development.

The aim of this fellowship is to get familiar with the practical implementation of Drupal and understanding the uses of theoretical knowledge. So, I have successfully completed this fellowship and compiled this report as a summary.

I would like to express my sincere gratitude to my project mentor Prof. Albert Sunny who has given me this valuable opportunity to work in such a learning environment and has guided me throughout the duration of the fellowship.

Mukta Girish Deshpande
Place : Nagpur
Date : 23 July 2020

Contents

Chapter 1

Introduction

1.1 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 percent 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 website installation and administration of the framework requires 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. For any special functionality that one needs in his website can be attained by modules provided by Drupal. Drupal makes it easy for the user to build a website. Drupal is one of the CMS because:

1. Drupal comes with the option of choosing from the variety of modules and third-party integrations that can be used for developing a website adhering to your preference.

2. Drupal comes with clean markup code that makes it easy for developers to manage content publishing.

3. The competency of Drupal lies in its versatility. Since it is open-source, any web developer can work on it and provide the user with numerous choices.

Chapter 2

Overview of Central Library

2.1 Insight into Central Library IIT Palakkad

Central Library, the informatics center of Indian Institute of Technology Palakkad provides an enjoyable learning experience with a carefully developed collection of books, journals, standards, magazines and newspapers. The library also stores collection of audio-visual materials such as CD-ROM, scientific kits etc.

The library opened its doors to the students, faculty and staff in August 2015 with a collection of 700 printed books which has grown to nearly 6000 (5942) printed barcoded and RFID tagged books (textbooks, reference, popular sciences and literature) in the past six years. Based on the needs and requirements of researchers, the library has subscribed to a number of electronic journals for its users. The library also has the support of national consortium E-Shodh Sindhu (INFLIBNET) to fulfill maximum journal requirement.

The operations of the library are fully computerized and enabled with the RFID system for fast transactions, for ease of access as well as for the security of the library. The RFID based kiosk provides self-check-in and self-check-out of books. Library is under a 24x7 CCTV surveillance system for security. The library is also enabled with Wi-Fi and LAN facility for unlimited high-speed internet access. Online facilities of the library are available 24x7x365 days for its registered users. Users can renew, reserve books through Online Public Access Catalog (OPAC) at any time. The library also renders services such as Reference and Consultation as well as updates the users with the Current Awareness Services.

The users of Central Library of IIT Palakkad are also registered with the National Digital Library sponsored by MHRD and coordinated by IIT Kharagpur.

Chapter 3

Content Types

3.1 About Content Types

In Drupal, a content type is a pre-defined collection of data types (fields) which relate to each other by an informational context.Content types are how site editors can input original content on a Drupal site.

A developer can add new content types to collect specific information for display. Also, existing content types can be extended by adding additional fields.All the posts that exist on a Drual website are instances of content types.

3.2 Drupal Core Content Types

Drupal 8 provides the following core Content Types :

- **Article:** Used for time-sensitive content like news, press releases or blog posts.
- **Basic Page:** Used for static content, such as an 'About us' page.
- **Book Page:** Books have a built-in hierarchical navigation. Used for handbooks or tutorials.
- **Forum Topic:** A forum topic starts a new discussion thread within a forum.

3.3 Creating Custom Content Types

Following steps should be followed to create a custom content type in Drupal:

1. Navigate to the administration toolbar and click on Structure / Content Types
2. Then click on add content type button

3. Fill in the name,description and title field label for the Content Type
4. Click on save and manage fields
5. Then click on add field button to add fields to the content type if required
6. Save the settings by clicking on save configuration button

Chapter 4

Modules

The basic functionality needed to develop a website is provided by Drupal in its core itself. The user just needs to enable it from the extension.

The core modules make it easy to add content, publish them and create pages. The module gives the user full control over how they want their website's functionality.

A module is a set of PHP, JavaScript, and/or CSS files that extends site features and adds functionality.The user can enable the features of a module by installing it and remove the features by uninstalling it.Removing the data and configuration related to the module before uninstalling it may be required. Each module that is installed adds to the time needed to generate pages on site, so it is a good idea to uninstall modules that are not needed.

Hence to improve the features of the library website the following contributed module has been used.

Chapter 5

Devel Module

5.1 About

Devel is a suite of modules that provides helper functions, admin pages, and additional development Drush commands to use during development.

When Devel module is downloaded in the site folders , four new Devel modules appear in the Extend page :

- **Devel :** Provides various blocks, pages and functions for developers
- **Devel Generate :** Generates dummy users,nodes and taxonomy terms
- **Devel Knit :** Wrapper for the Kint debugging tool
- **Web Profiler :** A port of the Symfony Web Profiler Bundle toolbar for Drupal 8

Each of these Devel tools can help you with development.
Devel,of course, is the main module and must be enabled for the other modules to operate.

Devel module provides the following features :

- adds new PHP functions to help with development and debugging.
- adds blocks: for running custom PHP on a page, for quickly accessing Devel pages, for masquerading as other users and a mail-system class which redirects outbound email to files.
- allows the developer to inspect Services and Parameters registered in the Service Container.

- allows the developer to inspect routes information, gathering all routing data from routing.yml files and from classes which subscribe to the route build/alter events.
- The module allows the developer to inspect listeners registered in the event dispatcher.

5.2 Installation and Configuration

Following steps should be followed to install the Devel module :

1. Visit the url: https://www.drupal.org/project/devel
2. Download the tar or zip file of the module that is compatible with the version of Drupal used
3. Extract the downloaded file in the modules directory of your project
4. On the administration toolbar of your Drupal site, click on Extend
5. Under the Development section , click on the checboxes before the Devel modules to enable them.

Chapter 6

Views

6.1 Overview

A view is a listing of content on a website. The core Views module handles the display of views, and the core Views UI module allows you to create and edit them in the administrative interface. When you define views, you are interested in taking data from your website and displaying it to the user.

Views can be created to practically output any conent entity that is stored in the system.
A listing created by a view can be in any of the following forms :

- Table with sortable fields
- Grid Layout
- Teasers or pictures that link to article
- Blocks
- JSON output
- RSS Feeds
- Calenders
- On-screen slideshows

The following views have been used to display content on the library website

Chapter 7

Grid View

7.1 Creating a Grid View

Following are the steps to implement grid view :

1. **Create an image style**

   a. Click Configuration / Image styles
   b. Click Add image style
   c. Give the new image style a proper name
   d. Click Create new style
   e. Select Crop from the dropdown
   f. Click Add
   g. Set height and width for the crop effect (make sure both dimensions are equal)
   h. Leave the default crop anchor at the center of the image
   i. Click Add effect
   j. Make sure the image effect was recorded properly and click Save

2. **Create the View**

   a. Create content nodes in your Drupal installation with their images
   b. Click Structure / Views / Add view
   c. Configure initial settings

   - Select content type
   - Create a block
   - Select unformatted list of fields
   - Specify number of items per block

d. Click Save and edit
e. In the FIELDS area click Add
f. Type image in the search box
g. Check it
h. Click Add and configure fields
i. Click the checkbox before Exclude this from display
j. Change the Image style
k. Link the image to the content
l. Click Apply
m. Rearrange the fields, so that the Image field is above
n. Click Apply

3. **Create a CSS Class**

a. Click on the Title field to edit it
b. Display the Style settings
c. Check Customize HTML element
d. Select SECTION
e. Check create a CSS class
f. Type gallery-item
g. Uncheck Add default classes
h. Open the Rewrite results section
i. Check Override the output of this field with custom text
j. Enter this code: field image title
k. Click Apply
l. Save the view

4. **Placing the Block**

a. Click Structure / Block layout
b. Scroll down to the Content section
c. Click Place block
d. Search for your block
e. Click Place block
f. Uncheck Display title
g. Click Save block
h. Drag the cross handle and place the block above the Main content
i. Scroll down and click Save blocks

Chapter 8

List View

8.1 Creating a List View

Following are the steps to create a list view :

1. In the Manage administrative menu, navigate to Structure / Views / Add view .The Add view wizard appears

2. Create a view

   a. View basic information/View name
   b. View settings / Show
   c. View settings / of type
   d. View settings / sorted by
   e. Page settings / Create a page
   f. Page settings / Page title
   g. Page settings / Path
   h. Page settings / Page display settings/Display format
   i. Page settings / Items to display
   j. Page settings / Use a pager
   k. Page settings / Create a menu link
   l. Page settings / Menu
   m. Page settings / Link text

3. Click Save and edit. The view configuration page appears.

4. Under Fields, click Add from the dropdown button. The Add fields pop-up appears.

5. Enter the word ”image” in the search field.

6. Check Main image in the table.

7. Click Apply. The Configure field: Content: Main Image pop-up appears.

8. Fill in the following fields for image settings

    a. Create a label
    b. Image style
    c. Link image to

9. Click Apply. The view configuration page appears.

10. Under Fields, click Add from the dropdown button. The Add fields pop-up appears.

11. Enter the word ”body” in the search field.

12. Select Body in the table.

13. Click Apply.

14. Fill in the following fields to configure field settings :

    a. Create a label
    b. Create a label
    c. Trimmed limit:

15. Click Apply. The view configuration page appears.

16. Under Fields, click Content: Title (Title). The Configure field: Content: Title pop-up appears.

17. Uncheck Create a label. This will remove the label that was created by the wizard.

18. Click Apply. The view configuration page appears.

19. Under Fields, click Rearrange in the dropdown button. The Rearrange fields pop-up appears.

20. Drag the cross bar handles of fields to put them into the right order: Image, Title, Body. As an alternative to dragging, you can click the Show row weights link at the top of the table and enter numerical weights (fields with lower or more negative weights will be shown first).

21. Click Apply. The view configuration page appears.

22. Optionally, click Update preview for a preview.

23. Click Save.

24. Navigate to the homepage and click Vendors from the main navigation to see the result.

Chapter 9

Taxonomy

9.1 About

Taxonomy is the practice and science of classifying things. In Drupal, the Taxonomy module allows you to classify your website content, and it can be an important part of your information architecture.

The first step in establishing a taxonomy is creating a new vocabulary. Next you provide the terms that fall within that vocabulary. The arrangement can be ”flat,” as in a tagging system, or hierarchical.

Sometimes the developer might want to create a ”controlled vocabulary,” where content authors can assign terms that have been predetermined. If so, the developer will need to add those terms to vocabulary in advance.
An alternative model for organizing information is the use of ”tags.” User-defined tags can be added to Drupal content on the fly.
By using multiple vocabularies it is possible to classify an individual node in many ways. For example, a node representing a musical work might have a genre vocabulary and a time-period vocabulary (including terms such as: seventeenth century, eighteenth century). The node might also be identified using a vocabulary term, such as ”sonata”. Adding a vocabulary for ”composers”, might lead to the following combination of terms: as an ”early-eighteenth-century” ”sonata” by ”Bach”, and it could be located by any of these three terms.

9.2 Adding a vocubulary

- Navigate to the administration toolbar and click Structure then on Taxonomy.
- Click on Add vocabulary button.
- Choose a name for your vocabulary.
- Give your vocabulary a description.

- You can also give your vocabulary ”help text” to help your users choose. Note that when creating a new content type, existing vocabularies do not have the new content type selected; you need to edit them to add the type.

- Vocabularies can have hierarchies of terms.

- Finally, you can delete the vocabulary altogether, thereby also deleting all its terms, but not the nodes to which they were assigned.

9.3 Adding terms to vocabulary

Add terms to a vocabulary by navigating to admin/structure/taxonomy/[your-vocabulary-name]/add. From there, Drupal will prompt for:

- Term name (Required) – The name for this term. Example: Technology.

- Description (Optional) – Description of the term (this item may be used by some modules and feeds).

Chapter 10

Automation

10.1 About

Automation can be defined as a technology concerned with performing a process by means of programmed commands combined with automatic feedback control to ensure proper execution of the instructions. The resulting system is capable of operating without human intervention.

10.2 Automation using Selenium

Introduction

Selenium Python bindings provides a simple API to write functional/acceptance tests using Selenium WebDriver. Through Selenium Python API you can access all functionalities of Selenium WebDriver in an intuitive way.

The API grants access to Selenium WebDrivers like Firefox, Ie, Chrome, Remote etc. The current supported Python versions are 2.7, 3.5 and above.

10.3 Downloading Python bindings for Selenium

Python bindings for Selenium can be downloaded by running the following command :

pip install selenium

10.4 Drivers

Selenium requires a driver to interface with the chosen browser.These drivers need to be installed before running automation tests.

Following are the web drivers used to automate the library website and links to install them :

Chrome: https://sites.google.com/a/chromium.org/chromedriver/downloads

Firefox: https://github.com/mozilla/geckodriver/releases

10.5 Python Libraries used for Automation

The following Python libraries have been used to write the automation script for data entry :

- **NumPy:** NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

  NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

  Command to install NumPy : pip install numpy

- **Pandas:** Pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with structured (tabular, multidimensional, potentially heterogeneous) and time series data both easy and intuitive.Pandas is well suited for many different kinds of data.

  The two primary data structures of pandas, Series (1-dimensional) and DataFrame (2-dimensional), handle the vast majority of typical use cases in finance, statistics, social science, and many areas of engineering. For R users, DataFrame provides everything that R's data.frame provides and much more. pandas is built on top of NumPy and is intended to integrate well within a scientific computing environment with many other 3rd party libraries.

  Command to install pandas : pip install pandas

- **Textdistance:** Python library for comparing distance between two or more sequences by many algorithms.

  Command to install textdistance : pip install textdistance

- **getpass:** The getpass module provides two functions:

  - **getpass.getpass(prompt='Password: ', stream=None)** Prompt the user for a password without echoing. The user is prompted using the string prompt, which defaults to 'Password: '. On Unix, the prompt is written to the file-like object stream using the replace error handler if needed. stream defaults to the controlling terminal (/dev/tty) or if that is unavailable to sys.stderr (this argument is ignored on Windows).

    If echo free input is unavailable getpass() falls back to printing a warning message to stream and reading from sys.stdin and issuing a GetPassWarning.

    exception getpass.GetPassWarning A UserWarning subclass issued when password input may be echoed.

  - **getpass.getuser()** Return the "login name" of the user.

    This function checks the environment variables LOGNAME, USER, LNAME and USERNAME, in order, and returns the value of the first one which is set to a non-empty string. If none are set, the login name from the password database is returned on systems which support the pwd module, otherwise, an exception is raised.

    In general, this function should be preferred over os.getlogin().

- **time:**This module provides various time-related functions.

  time.sleep(secs) Suspend execution of the calling thread for the given number of seconds. The argument may be a floating point number to indicate a more precise sleep time. The actual suspension time may be less than that requested because any caught signal will terminate the sleep() following execution of that signal's catching routine. Also, the suspension time may be longer than requested by an arbitrary amount because of the scheduling of other activity in the system.

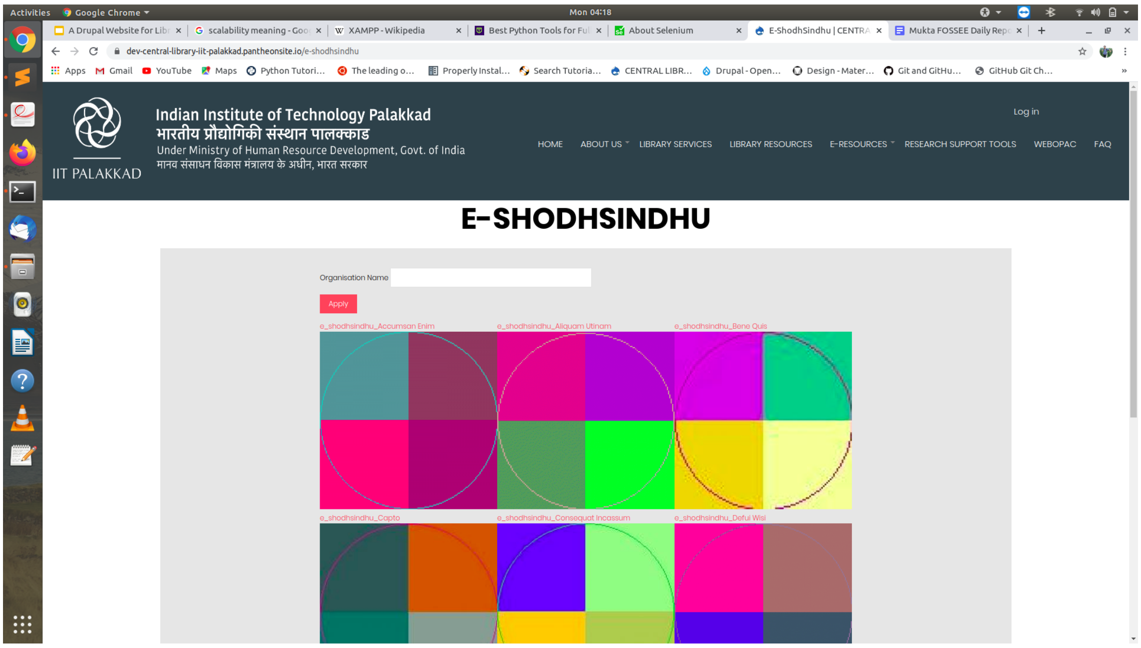
  Changed in version 3.5: The function now sleeps at least secs even if the sleep is interrupted by a signal, except if the signal handler raises an exception

Chapter 11

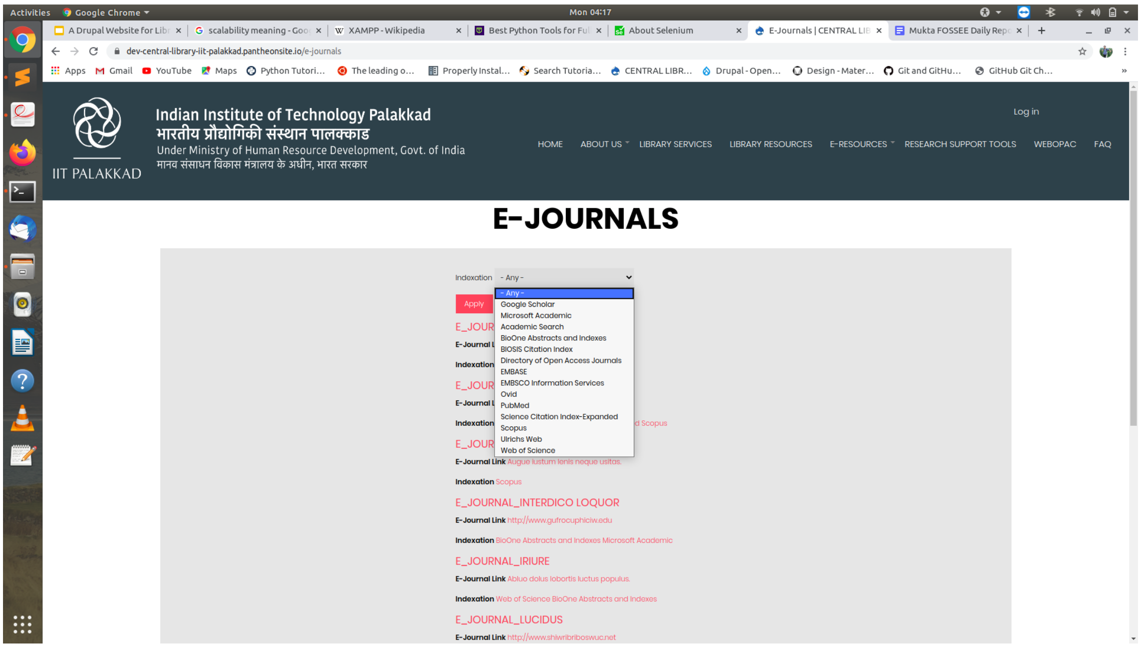
Overview of Work Done

11.1 Layout

Following are the images of the layouts designed :



1. Grid Layout for resources obtained from E-ShodhSindhu with an organisation name filter.



dev-central-library-iit-palakkad.pantheonsite.io/e-journals

Indian Institute of Technology Palakkad
भारतीय प्रौद्योगिकी संस्थान पालक्काड
Under Ministry of Human Resource Development, Govt. of India
मानव संसाधन विकास मंत्रालय के अधीन, भारत सरकार

IIT PALAKKAD

Log in

HOME ABOUT US LIBRARY SERVICES LIBRARY RESOURCES E-RESOURCES RESEARCH SUPPORT TOOLS WEBOPAC FAQ

E-JOURNALS

Indexation - Any -

- Any -
Google Scholar
Microsoft Academic
Academic Search
BioOne Abstracts and Indexes
BIOSIS Citation Index
Directory of Open Access Journals
EMBASE
EMBSCO Information Services
Ovid
PubMed
Science Citation Index-Expanded
Scopus
Ulrichs Web
Web of Science

Apply

E-Journal Link Augue iustum lenis neque usitas.

Indexation Scopus

E_JOURNAL_INTERDICO LOQUOR

E-Journal Link http://www.gufrocuphiciw.edu

Indexation BioOne Abstracts and Indexes Microsoft Academic

E_JOURNAL_IRIURE

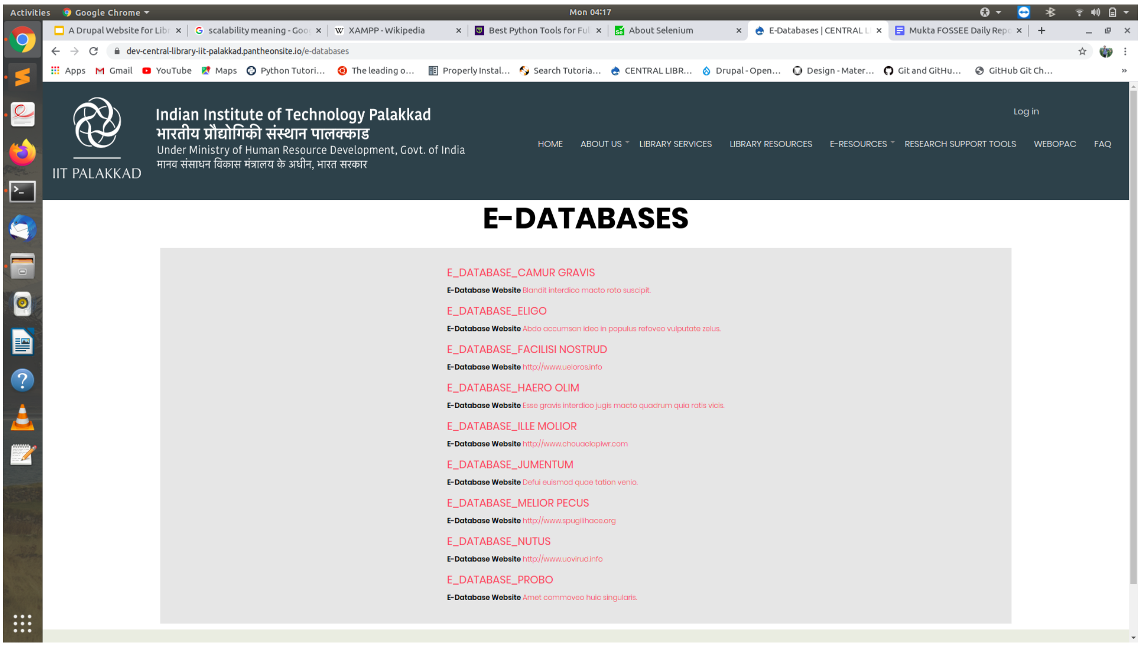
E-Journal Link Abluo dolus lobortis luctus populus.

Indexation Web of Science BioOne Abstracts and Indexes

E_JOURNAL_LUCIDUS

E-Journal Link http://www.shiwribriboswuc.net

2. List Layout of Teasers for E-Journals with an indexation filter.



dev-central-library-iit-palakkad.pantheonsite.io/e-databases

Indian Institute of Technology Palakkad
भारतीय प्रौद्योगिकी संस्थान पालक्काड
Under Ministry of Human Resource Development, Govt. of India
मानव संसाधन विकास मंत्रालय के अधीन, भारत सरकार

IIT PALAKKAD

Log in

HOME ABOUT US LIBRARY SERVICES LIBRARY RESOURCES E-RESOURCES RESEARCH SUPPORT TOOLS WEBOPAC FAQ

E-DATABASES

E_DATABASE_CAMUR GRAVIS

E-Database Website Blandit interdico macto roto suscipit.

E_DATABASE_ELIGO

E-Database Website Abdo accumsan ideo in populus refoveo vulputate zelus.

E_DATABASE_FACILISI NOSTRUD

E-Database Website http://www.ueloros.info

E_DATABASE_HAERO OLIM

E-Database Website Esse gravis interdico jugis macto quadrum quia ratis vicis.

E_DATABASE_ILLE MOLIOR

E-Database Website http://www.chouaclapiwr.com

E_DATABASE_JUMENTUM

E-Database Website Defui euismod quae tation venio.

E_DATABASE_MELIOR PECUS

E-Database Website http://www.spugilihace.org

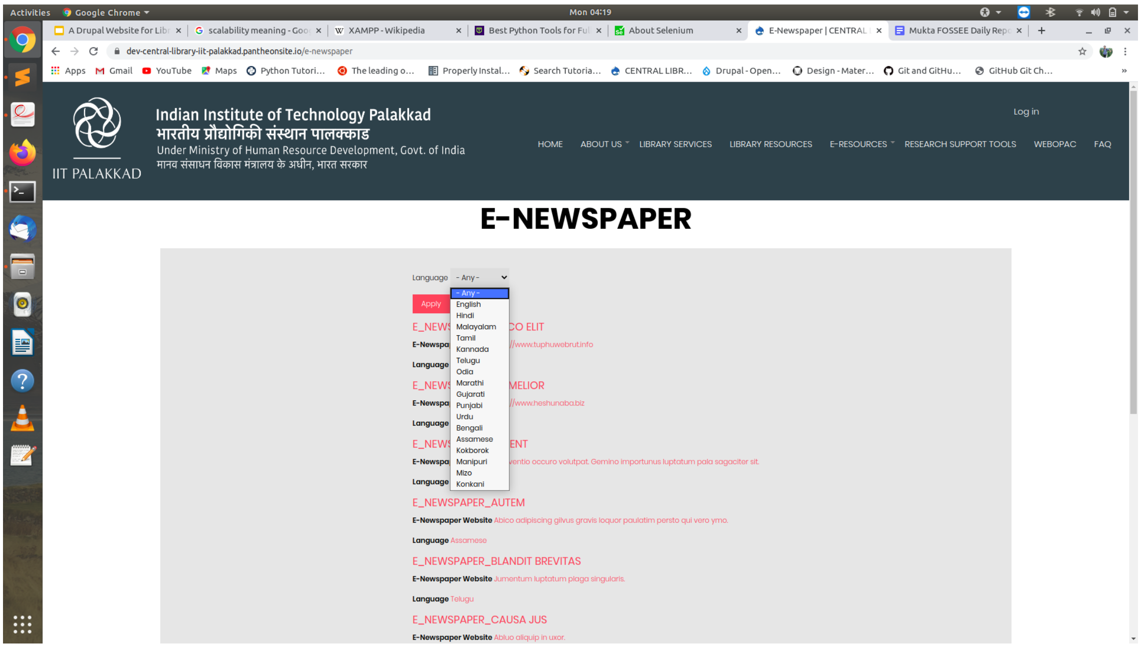
E_DATABASE_NUTUS

E-Database Website http://www.uovirud.info

E_DATABASE_PROBO

E-Database Website Amet commoveo huic singularis.

3. List Layout of Teasers for E-Databases.



Indian Institute of Technology Palakkad
भारतीय प्रौद्योगिकी संस्थान पालक्काड
Under Ministry of Human Resource Development, Govt. of India
मानव संसाधन विकास मंत्रालय के अधीन, भारत सरकार
IIT PALAKKAD

E-NEWSPAPER

Language - Any -

- Any -
English
Hindi
Malayalam
Tamil
Kannada
Telugu
Odia
Marathi
Gujarati
Punjabi
Urdu
Bengali
Assamese
Kokborok
Manipuri
Mizo
Konkani

E_NEWS CO ELIT
E-Newspa //www.tuphuwebrut.info
Language

E_NEWS MELIOR
E-Newspa //www.heshunaba.biz
Language

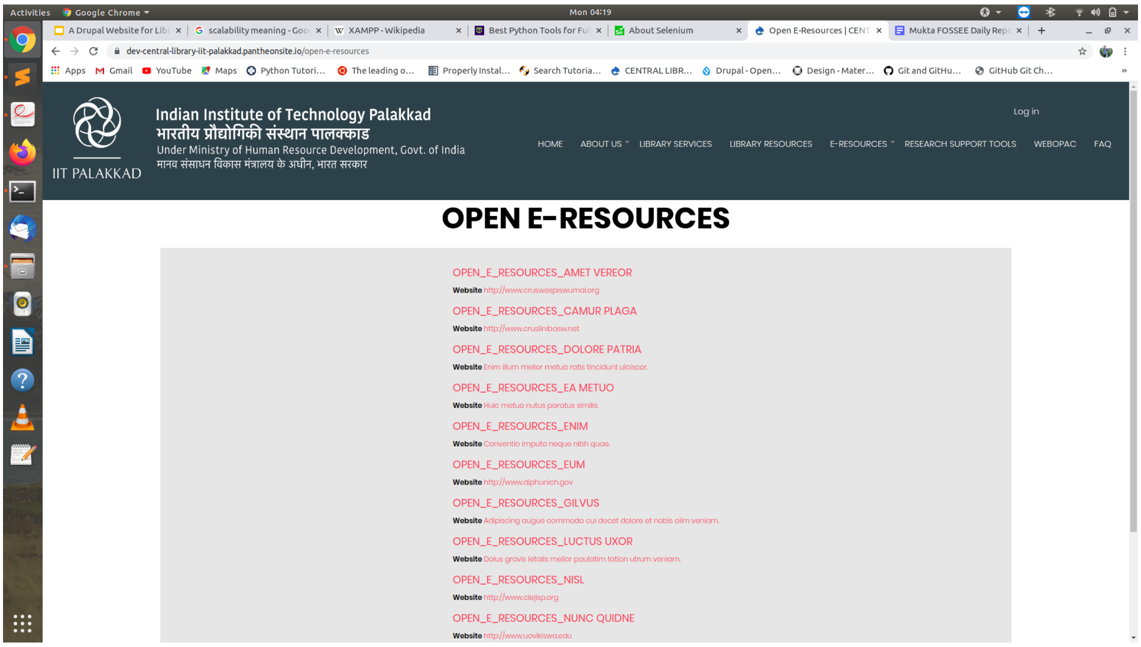
E_NEWS ENT
E-Newspa ventio occuro volutpat. Gemino importunus luptatum pala sagaciter sit.
Language

E_NEWSPAPER_AUTEM
E-Newspaper Website Abico adipiscing gilvus gravis loquor paulatim persto qui vero ymo.
Language Assamese

E_NEWSPAPER_BLANDIT BREVITAS
E-Newspaper Website Jumentum luptatum plaga singularis.
Language Telugu

E_NEWSPAPER_CAUSA JUS
E-Newspaper Website Abluo aliquip in uxor.

4. List Layout of Teasers for E-Newspapers with a language filter.



Indian Institute of Technology Palakkad
भारतीय प्रौद्योगिकी संस्थान पालक्काड
Under Ministry of Human Resource Development, Govt. of India
मानव संसाधन विकास मंत्रालय के अधीन, भारत सरकार
IIT PALAKKAD

OPEN E-RESOURCES

OPEN_E_RESOURCES_AMET VEREOR
Website http://www.cruswespiswumal.org

OPEN_E_RESOURCES_CAMUR PLAGA
Website http://www.cruslinibasw.net

OPEN_E_RESOURCES_DOLORE PATRIA
Website Enim illum melior metuo ratis tincidunt ulciscor.

OPEN_E_RESOURCES_EA METUO
Website Huic metuo nutus paratus similis.

OPEN_E_RESOURCES_ENIM
Website Conventio imputo neque nibh quae.

OPEN_E_RESOURCES_EUM
Website http://www.diphunich.gov

OPEN_E_RESOURCES_GILVUS
Website Adipiscing augue commodo cui decet dolore et nobis olim veniam.

OPEN_E_RESOURCES_LUCTUS UXOR
Website Dolus gravis letalis melior paulatim tation utrum veniam.

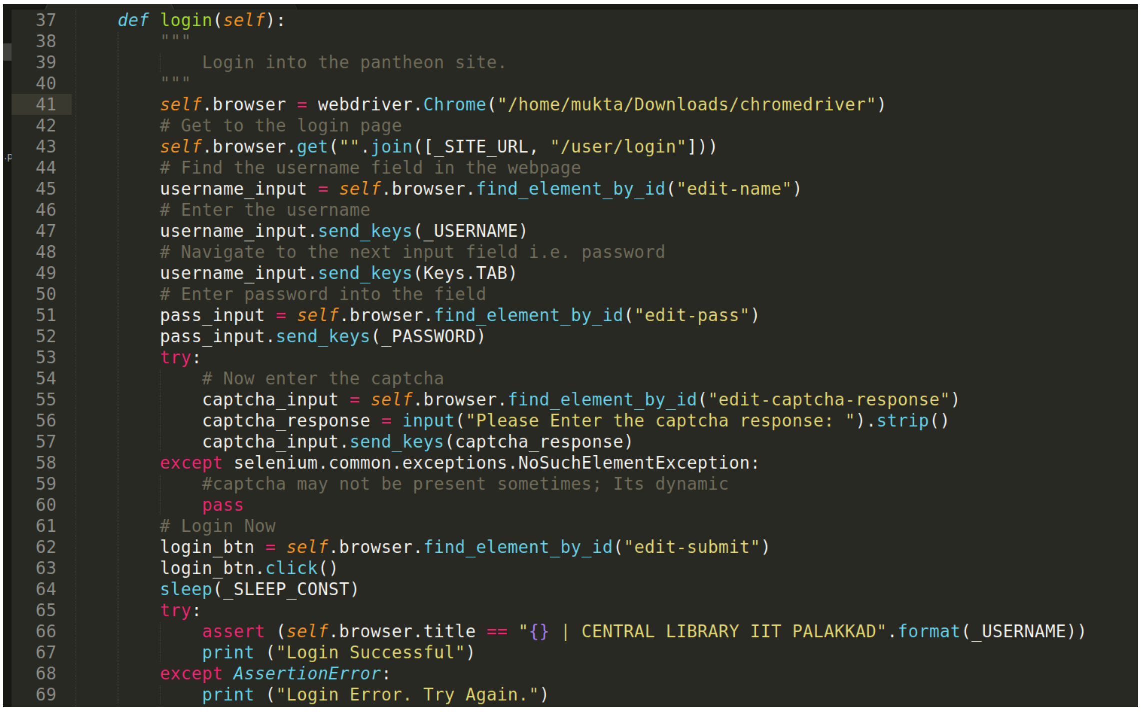
OPEN_E_RESOURCES_NISL
Website http://www.clejisp.org

OPEN_E_RESOURCES_NUNC QUIDNE
Website http://www.uovikiswa.edu

5. List Layout of Teasers for Open E-Resources.

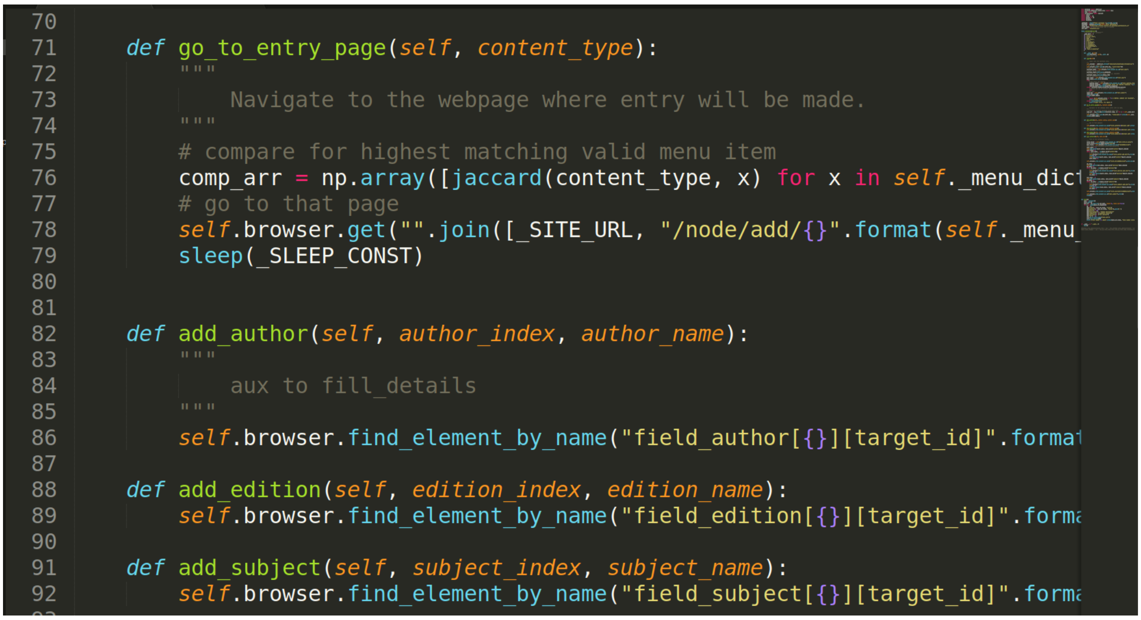
11.2 Automation Code

Following are the images of the automation code :



```
    def login(self):
        """
            Login into the pantheon site.
        """
        self.browser = webdriver.Chrome("/home/mukta/Downloads/chromedriver")
        # Get to the login page
        self.browser.get("".join([_SITE_URL, "/user/login"]))
        # Find the username field in the webpage
        username_input = self.browser.find_element_by_id("edit-name")
        # Enter the username
        username_input.send_keys(_USERNAME)
        # Navigate to the next input field i.e. password
        username_input.send_keys(Keys.TAB)
        # Enter password into the field
        pass_input = self.browser.find_element_by_id("edit-pass")
        pass_input.send_keys(_PASSWORD)
        try:
            # Now enter the captcha
            captcha_input = self.browser.find_element_by_id("edit-captcha-response")
            captcha_response = input("Please Enter the captcha response: ").strip()
            captcha_input.send_keys(captcha_response)
        except selenium.common.exceptions.NoSuchElementException:
            #captcha may not be present sometimes; Its dynamic
            pass
        # Login Now
        login_btn = self.browser.find_element_by_id("edit-submit")
        login_btn.click()
        sleep(_SLEEP_CONST)
        try:
            assert (self.browser.title == "{} | CENTRAL LIBRARY IIT PALAKKAD".format(_USERNAME))
            print ("Login Successful")
        except AssertionError:
            print ("Login Error. Try Again.")
```

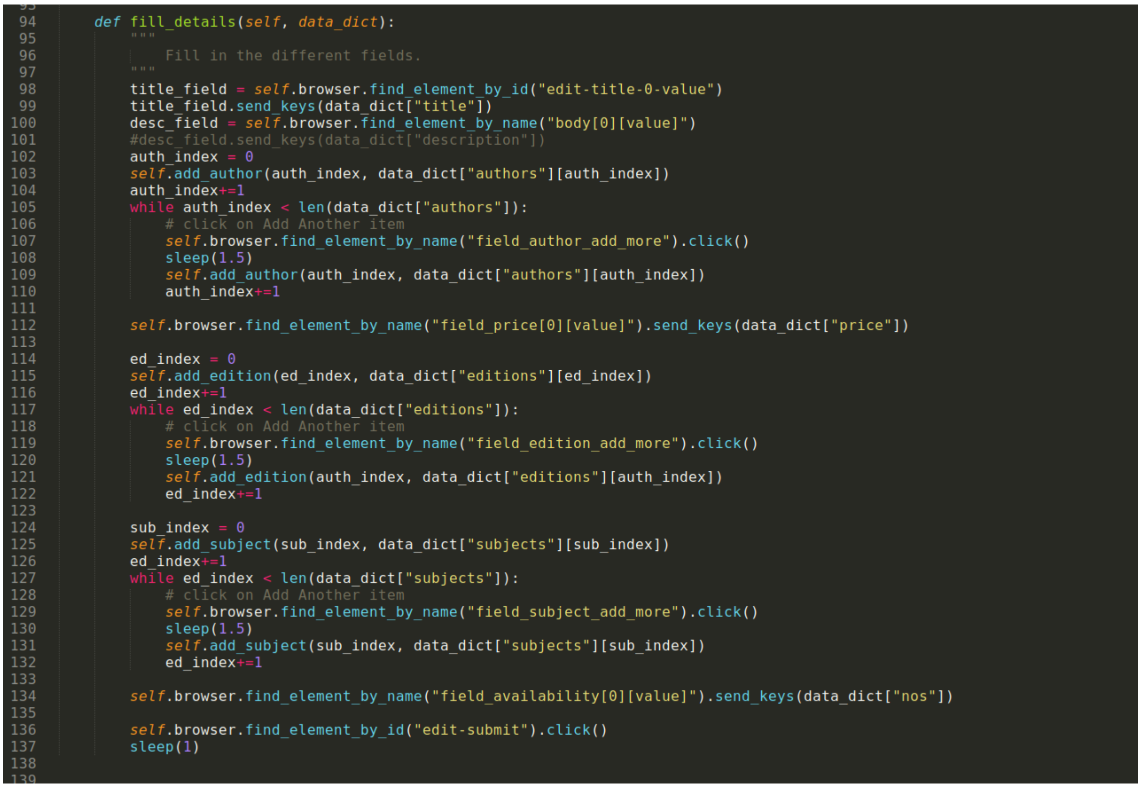
1. Function performing the task of of logging in to the Library Website.



```

    def go_to_entry_page(self, content_type):
        """
            Navigate to the webpage where entry will be made.
        """
        # compare for highest matching valid menu item
        comp_arr = np.array([jaccard(content_type, x) for x in self._menu_dict
        # go to that page
        self.browser.get("".join([_SITE_URL, "/node/add/{}".format(self._menu_
        sleep(_SLEEP_CONST)


    def add_author(self, author_index, author_name):
        """
            aux to fill_details
        """
        self.browser.find_element_by_name("field_author[{}][target_id]".format

    def add_edition(self, edition_index, edition_name):
        self.browser.find_element_by_name("field_edition[{}][target_id]".forma

    def add_subject(self, subject_index, subject_name):
        self.browser.find_element_by_name("field_subject[{}][target_id]".forma
```

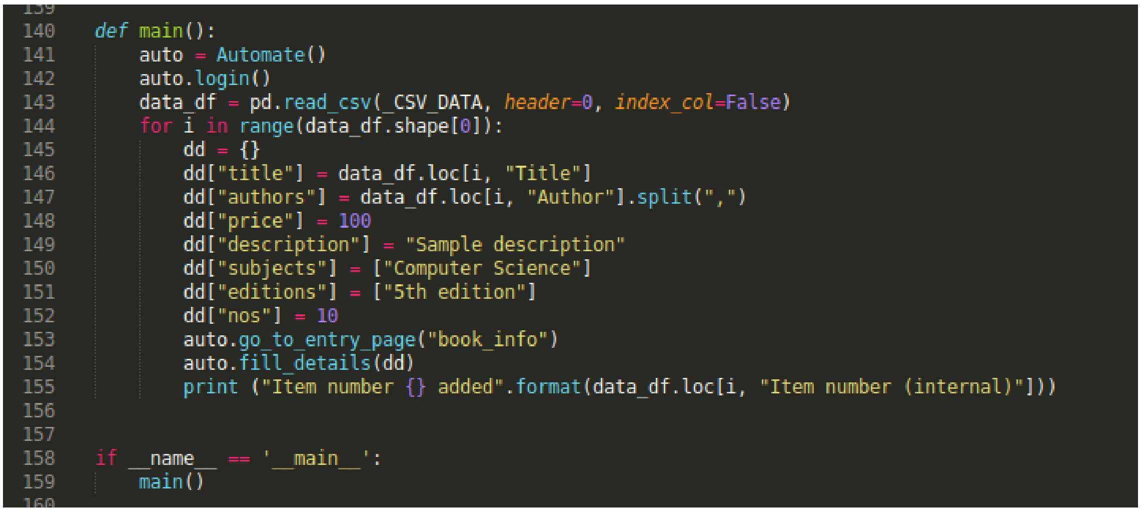
2. Functions performing the task of navigating to the webpage to enter data and

to enter the details of the books respectively.



```python
    def fill_details(self, data_dict):
        """
            Fill in the different fields.
        """
        title_field = self.browser.find_element_by_id("edit-title-0-value")
        title_field.send_keys(data_dict["title"])
        desc_field = self.browser.find_element_by_name("body[0][value]")
        #desc_field.send_keys(data_dict["description"])
        auth_index = 0
        self.add_author(auth_index, data_dict["authors"][auth_index])
        auth_index+=1
        while auth_index < len(data_dict["authors"]):
            # click on Add Another item
            self.browser.find_element_by_name("field_author_add_more").click()
            sleep(1.5)
            self.add_author(auth_index, data_dict["authors"][auth_index])
            auth_index+=1

        self.browser.find_element_by_name("field_price[0][value]").send_keys(data_dict["price"])

        ed_index = 0
        self.add_edition(ed_index, data_dict["editions"][ed_index])
        ed_index+=1
        while ed_index < len(data_dict["editions"]):
            # click on Add Another item
            self.browser.find_element_by_name("field_edition_add_more").click()
            sleep(1.5)
            self.add_edition(auth_index, data_dict["editions"][auth_index])
            ed_index+=1

        sub_index = 0
        self.add_subject(sub_index, data_dict["subjects"][sub_index])
        ed_index+=1
        while ed_index < len(data_dict["subjects"]):
            # click on Add Another item
            self.browser.find_element_by_name("field_subject_add_more").click()
            sleep(1.5)
            self.add_subject(sub_index, data_dict["subjects"][sub_index])
            ed_index+=1

        self.browser.find_element_by_name("field_availability[0][value]").send_keys(data_dict["nos"])

        self.browser.find_element_by_id("edit-submit").click()
        sleep(1)
```

3. Function to fill details in fields.



```python
def main():
    auto = Automate()
    auto.login()
    data_df = pd.read_csv(_CSV_DATA, header=0, index_col=False)
    for i in range(data_df.shape[0]):
        dd = {}
        dd["title"] = data_df.loc[i, "Title"]
        dd["authors"] = data_df.loc[i, "Author"].split(",")
        dd["price"] = 100
        dd["description"] = "Sample description"
        dd["subjects"] = ["Computer Science"]
        dd["editions"] = ["5th edition"]
        dd["nos"] = 10
        auto.go_to_entry_page("book_info")
        auto.fill_details(dd)
        print ("Item number {} added".format(data_df.loc[i, "Item number (internal)"]))


if __name__ == '__main__':
    main()
```

4. Main function where all the functions are called.

Reference

For summing up the report,the following resources were of great help :

- Drupal.org
- Spoken Tutorials
- OS Training
- Drupal Stackexchange
- PyPI.org
- docs.python.org
- NumPy.org