Extracting Files and Information from A Given Set of URLs at A Particular Depth Using Web Scraping

Mohammad Nabiluzzaman Neloy, 1831251042,
mohammad.nabiluzzaman@northsouth.edu
Department of ECE
North South University, Dhaka, Bangladesh

Rafid Ahmed, 1831395642,
rafid.ahmed@northsouth.edu
Department of ECE
North South University, Dhaka, Bangladesh

***Abstract* — The internet is full of contents nowadays, and sometimes while working on a research paper we might want to search for some important files that could only be found on a given URL. In addition, we might also require to perform some more research on the content that we will extract out of the given URL. In this paper, we have worked on a web application which gives the ability to extract all the file information from a given set of URLs at some given depth. Furthermore, it also provides the facility to perform some search on the content after the information has been extracted from the given URLs.**

***Index Terms* — Docx, Files, Html, Pdf, PPT, Research, Searching, Txt, Uniform Resource Locator, URLs, Web Scraping, Web Crawling**

I. Introduction

Research might require a lot of information that needs to be studied. Sometimes, the research might only have to be done from a given set of URLs. Extracting all the documents available on the given URL along with all the feasible information for the research might be difficult and time consuming if we do it manually. Our project "Spitrack" is a web application that enables researchers to provide a "set or cluster of URLs" with a particular depth and to select the document types (.pdf or .docx or .ppt or .txt or html content) that they want to extract out of them, and perform some search on the extracted information to obtain some practical results using a simple keyword and the selection of the created clusters. This report, explains how we have developed our web application using the web scraping technologies. Furthermore, it also shows how to use it and the results that we have obtained from our tests.

II. Methodology

This section describes the working procedures and implementations of the web scraping technology to design the proposed web-application and android-application for the simplification of research techniques.

A. *Basic Ideas and Flow-Charts*

The main idea of our project is to obtain a cluster of URLs from the user, extract all the available documents and information from the URLs, and finally perform some search on the extracted data to acquire the required results.

Fig. 1 – Procedure Flow-Chart

The major part of our project is the cluster creation. To begin with the cluster creation process we need to enter a minimum of three URLs and provide a cluster name. These URLs need to be the origin from which we want to extract the data. Next, we need to mention the minimum depth up to which we want our cluster to retrieve all documents and information. A more detailed explanation about the "depth" concept is provided in this paper later on. Finally, we need to select the required document types that we want to extract. Once we have configured all the information for the cluster, the cluster details will be passed to a "Cluster Creation API" which will run the web crawler to extract the required documents from the provided URLs and beyond up to the given depth. After the documents have been extracted, the content of the documents is parsed "based on the document type" and stored or indexed in a database, which is further used for searching.

These tasks may take a lot of time, that is why once the process is completed the "Cluster Creation API" sends an automatic email notification to the email registered by the user.

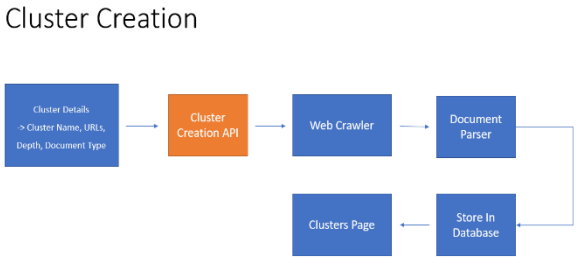

Fig. 2 – Cluster Creation Flow-Chart

B. *Depth Concept*

Considering a tree data structure, we can visualize the initial three URLs as the root nodes of the three different trees. Each root URL definitely has a number of sub-URLs "child nodes" within its page, and so does the URLs within them. Each level of sub-URLs is referred to as a depth in this project. The orange nodes in "Fig. 3 and Fig. 4" are the nodes where an actual document is found and the document has been parsed and stored into the database.

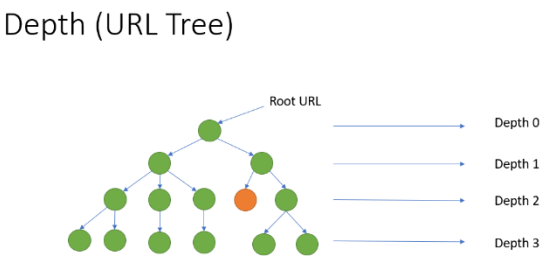

Fig. 3 – URL Depth Concept

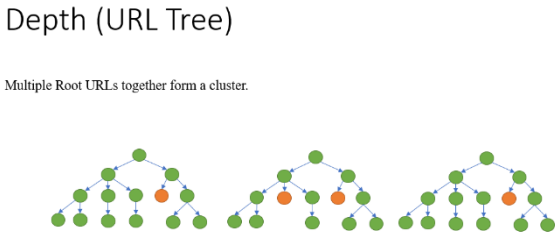

Fig. 3 – URL Depth Concept with three URLs

C. *Search Option*

Searching is an important feature that is highly required while doing a research, as a lot of information will be extracted using the cluster and only some specific information will be required. To perform a search, the user needs to enter a keyword about the topic for which they need the search results. Also, they have to select the clusters from which they want they want their information. The keyword and the select clusters are sent to the "Search API" which retrieves all the information related to the topic available in the database which were extracted during the cluster creation. Once the necessary information is retrieved, it is represented as results in the client side.

Fig. 4 – Searching Procedure Flow-Chart

D. *Software Architecture*

We have already discussed several parts of the main activities of our software in the previous parts of this paper. In this section, we will go through the overall view of the software architecture.

Firstly, the client part shows all the possible actions that a user or admin can perform through the software. New users can "sign up" to create an account, and then login to their account to start using the rest of the features. Users can use the "Create Cluster" option to create new clusters, use the "Clusters" option to see their previously created clusters, and use the "Search" option to search for the content that they require out of the extracted information. In case of the admin panel, we just have the option for adding new admins for now.

Secondly, the backend of the whole software is hosted into the web-server. The web-server takes all the requests from the client, and runs the appropriate APIs that we have created accordingly. It is to be noted, that some of our APIs use the "Djoser APIs" for authentication purposes which are already available on the web. The "Create Cluster API" uses the Scrapper functions based on the type of the document that needs to be scrapped. Furthermore, each scrapper uses a parser to retrieve all the content from the documents extracted using the scrappers so that they can be stored into the database.

Lastly, the database stores all the information about the users, admins, clusters and the scrapped content in detail. It also retrieves the required information for the APIs.

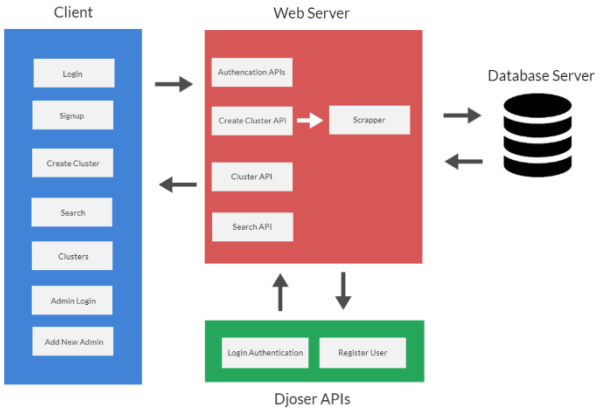

Fig. 4 – Software Architecture

E. *Software Stack*

(a) Software

- ✓ UI-Design – Adobe XD
- ✓ Version Control System – Git
- ✓ Repository Hosting Service – Bitbucket
- ✓ Task Management – Trello
- ✓ Communication Platform – Slack
- ✓ Backend Development Using Python (IDE) – JetBrains PyCharm
- ✓ Frontend Development (IDE) – VS Code
- ✓ Android Development – Android Studio
- ✓ Local Database Server (SQL Database) – Apache XAMPP
- ✓ Backend Server – Heroku
- ✓ Frontend Hosting – Firebase
- ✓ Database Server

(b) Technologies

- ✓ Django
- ✓ React Js
- ✓ Beautiful Soup for Web Scraping
- ✓ HTML
- ✓ CSS
- ✓ Bootstrap
- ✓ JavaScript
- ✓ Python
- ✓ XML
- ✓ Java
- ✓ MySQL
- ✓ Django Rest Framework

F. *Trello Boards*

The following Trello boards describe the work progress in both web-application and android application.

(a) Web-Application (Neloy)

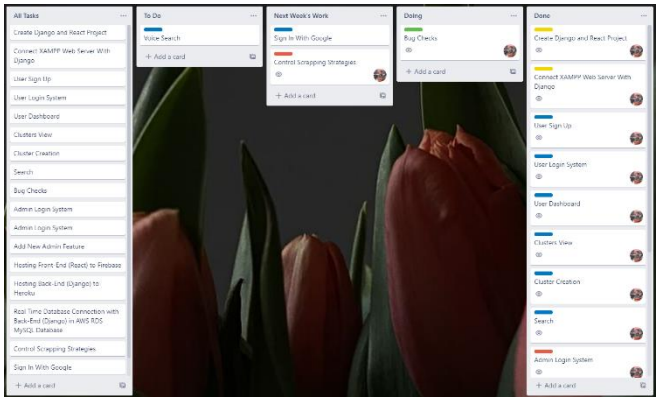

Fig. 5 – Trello Board Web-Application 1

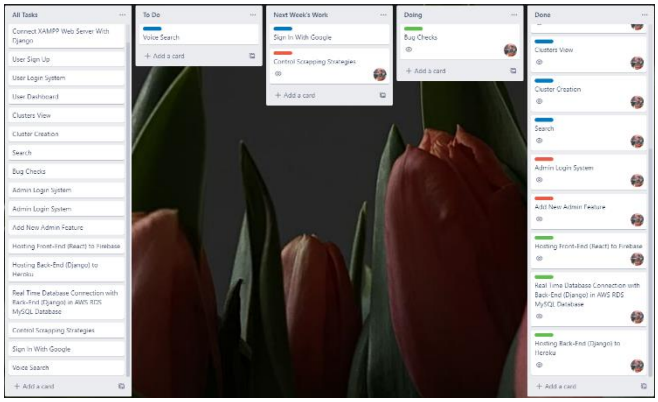

Fig. 6 – Trello Board Web-Application 2

(b) Android-Application (Rafid)

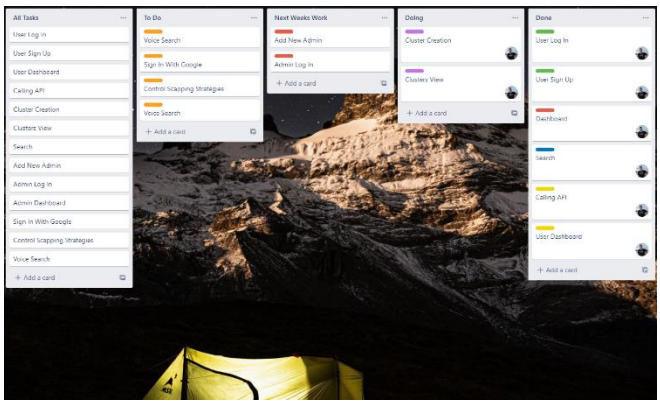

Fig. 7 – Trello Board Android-Application

III. Results and Analysis

A. *Web-Application Results*

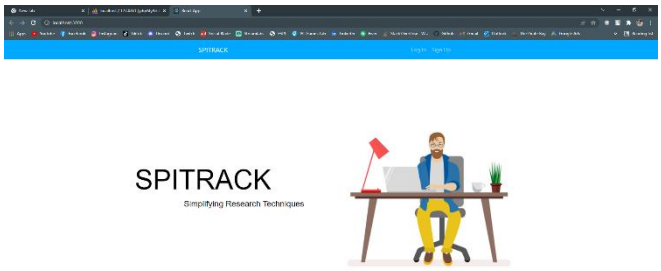

Fig. 8.1 – Landing Page

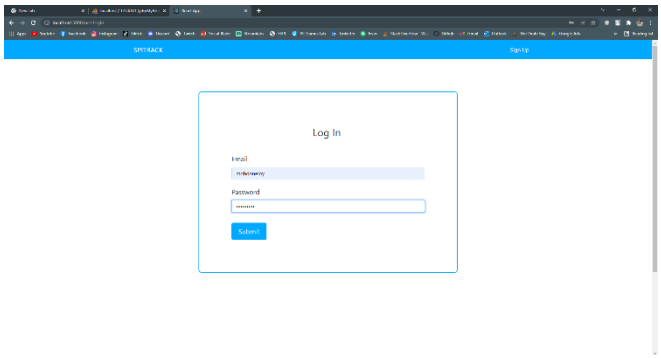

Fig. 8.2 – Login Page

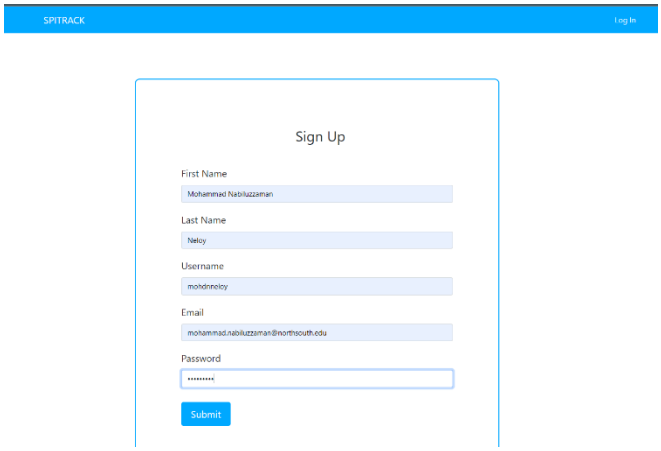

Fig. 8.3 – Sign Up Page

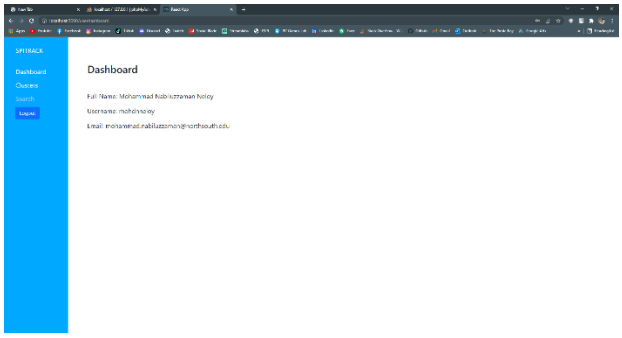

Fig. 8.4 – User Dashboard

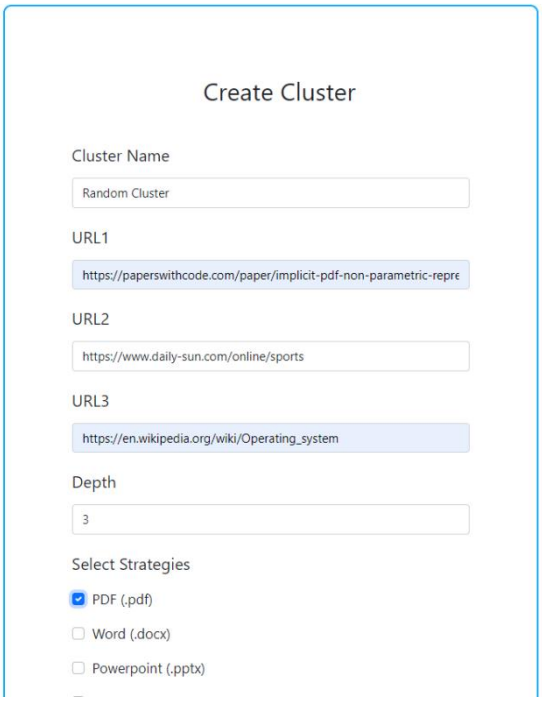

Fig. 8.5 – Create Cluster 1

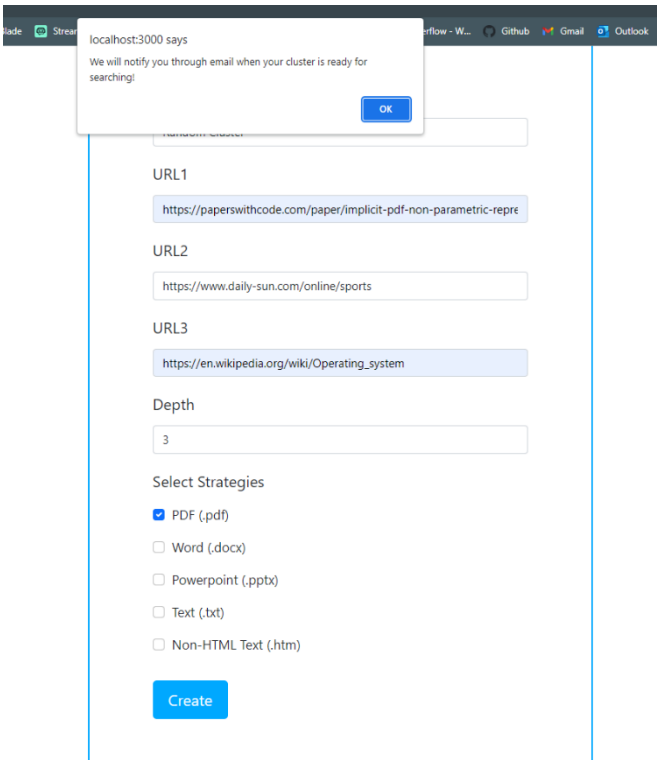

Fig. 8.6 – Create Cluster 2 (Alert)

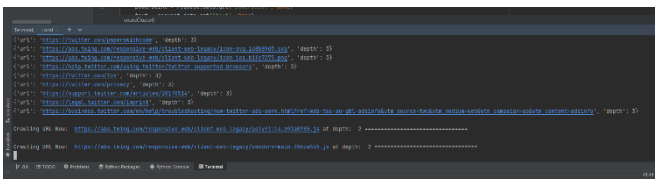

Fig. 8.7 – Crawling through URLs

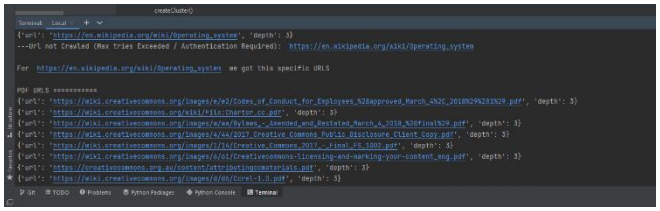

Fig. 8.8 – Extracted PDF URLs

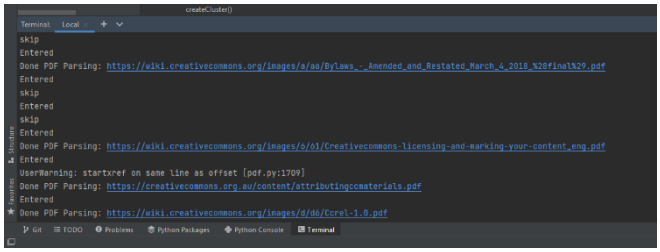

Fig. 8.9 – Parsing extracted PDF documents

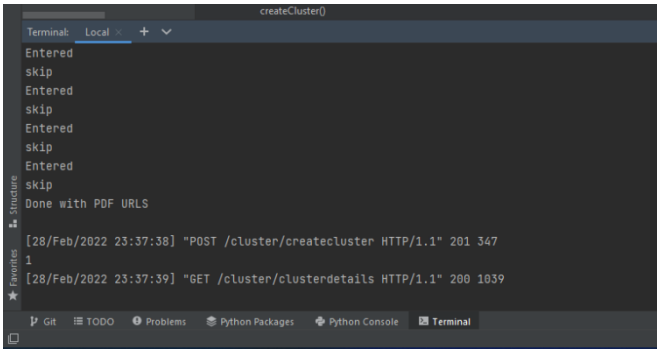

Fig. 8.10 – Cluster Created

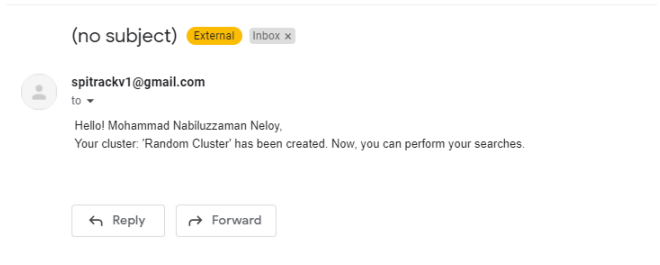

Fig. 8.11 – Email Notification

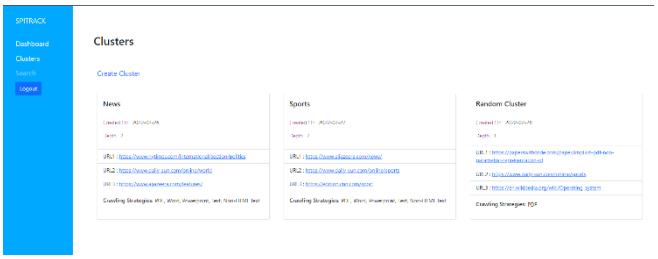

Fig. 8.12 – Clusters Page

Fig. 8.13 – All Clusters in Database

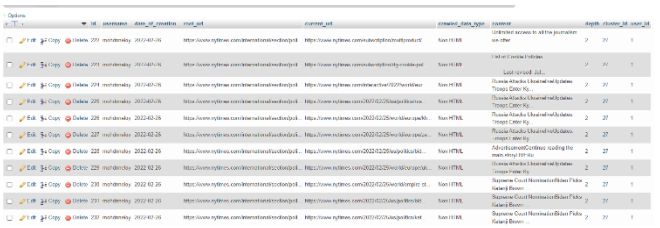

Fig. 8.14 – All extracted information using the Cluster URLs

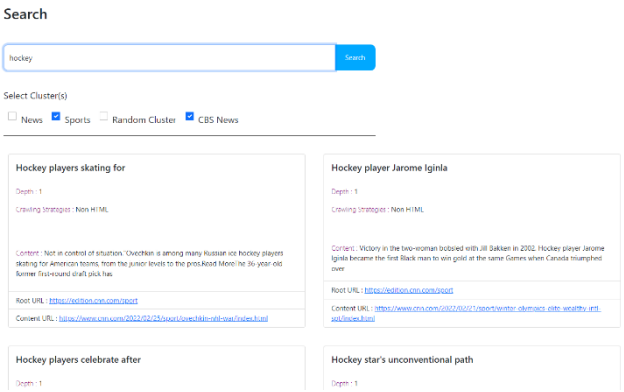

Fig. 8.15 – Searching

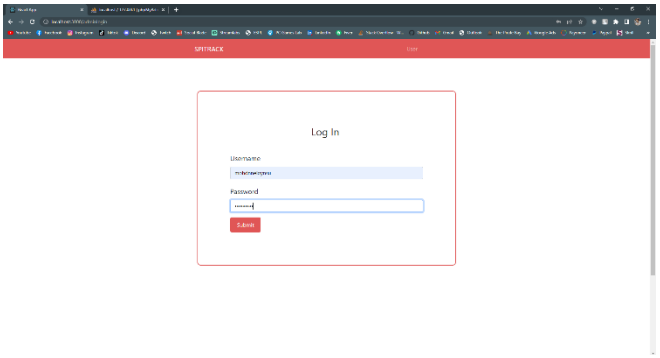

Fig. 8.16 – Admin Login Page

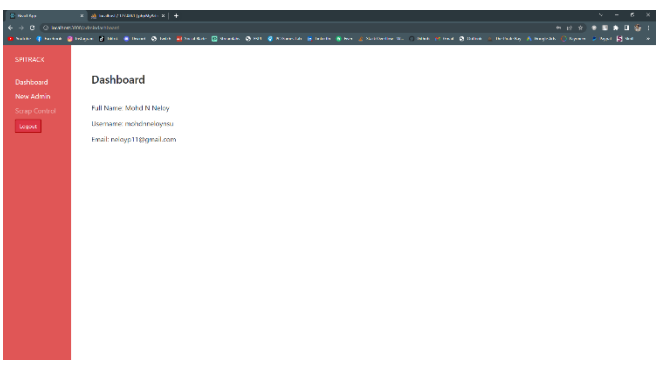

Fig. 8.17 – Admin Dashboard

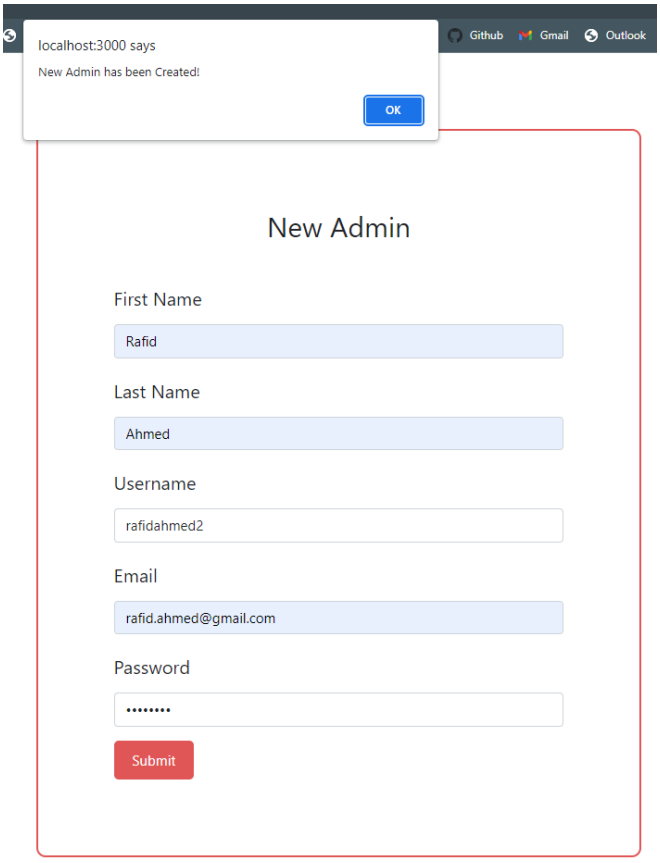

Fig. 8.18 – Add New Admin

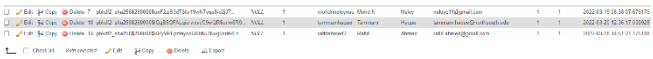

Fig. 8.19 – New Admin Added

B. *Android-Application Results*

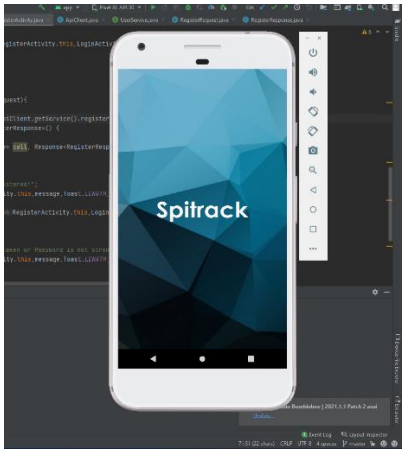

Fig. 9.1 – Splash Screen

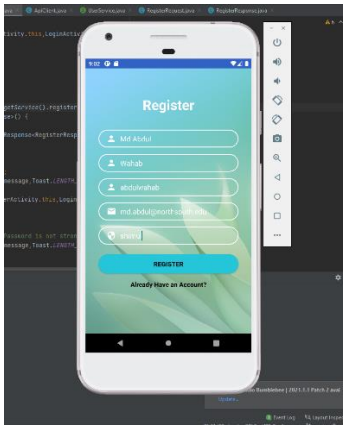

Fig. 9.2 – Registration Page

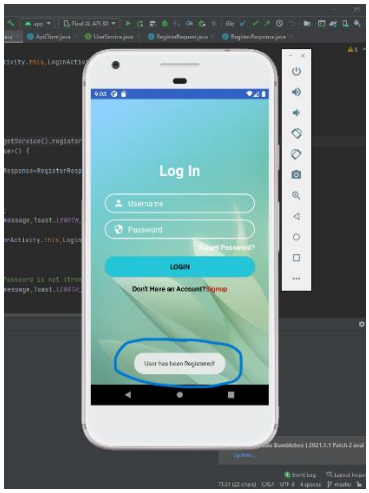

Fig. 9.3 – Successful Registration

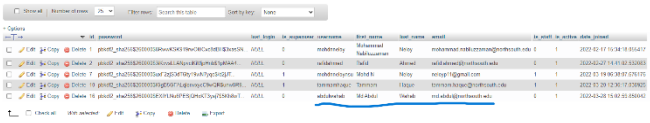

Fig. 9.4 – Registration in Database

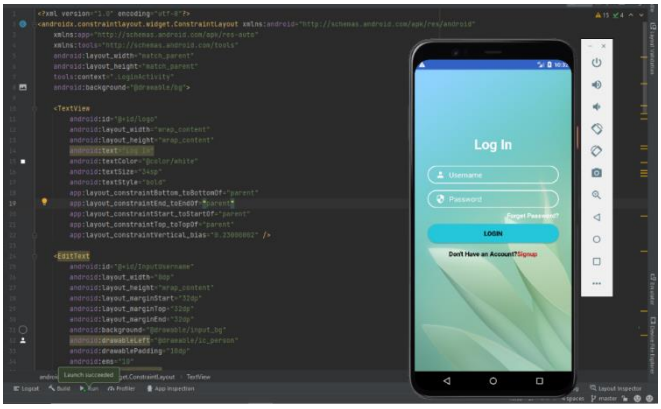

Fig. 9.5 – Login Page

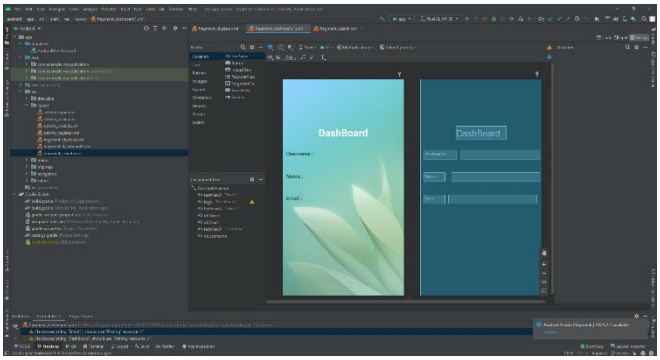

Fig. 9.6 – Dashboard

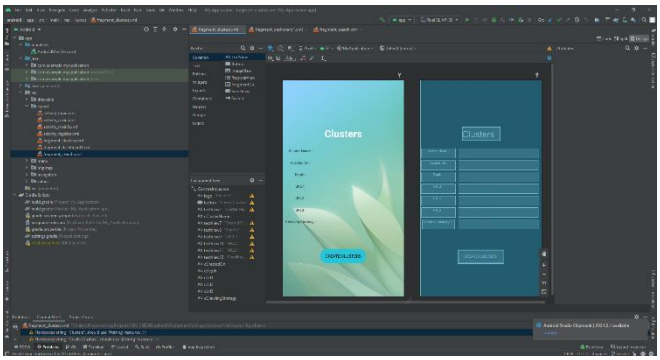

Fig. 9.7 – Clusters Page

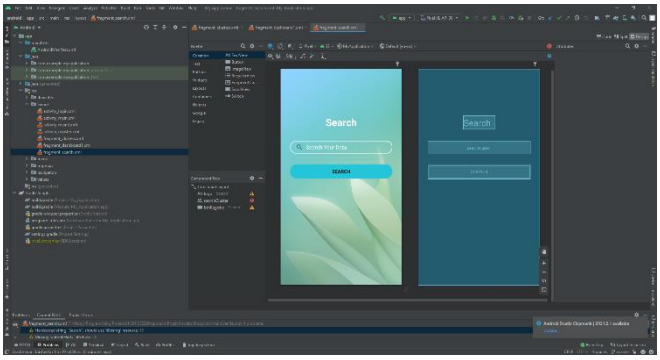

Fig. 9.8 – Search Page

IV. Conclusion

The difficulty of searching for specific document types like (.pdf or .docx or .ppt or .txt or html content) within a given URL might be tough if it is done manually. This research paper tried to find the best possible solution to extract all the documents of the given type out of a set of URLs and store all its content into a database. Not only that, but it also succeeded in retrieving the necessary information out of the database when the searching option was applied.

In future works, more features for the admin panel can be included. Some of the features include: "Cluster Creation Verification" where a cluster that is been placed for creation needs some verification by the admin so that it can actually start the scrapping process. "Third Party Scrapping Apps" where third party scrapping apps developed by external developers can be made public to the users by the admin. For the user panel, the "Search API" can be replaced with other advanced search engines like "Elastic Search". Voice search is also an additional feature

V. References

[1]. Django Rest Framework, django-rest-framework.org

Available: https://www.django-rest-framework.org/#installation

[2]. "How to extract text from a PDF file", stackoverflow.com [Online].

Available: https://stackoverflow.com/questions/34837707/how-to-extract-text-from-a-pdf-file

[3]. "Extracting text from multiple power point files using python", stackoverflow.com [Online].

Available: https://stackoverflow.com/questions/39418620/extracting-text-from-multiple-powerpoint-files-using-python

[4]. "How to extract text from an existing docx file using python-docx", stackoverflow.com [Online].

Available: https://stackoverflow.com/questions/25228106/how-to-extract-text-from-an-existing-docx-file-using-python-docx

[5]. "Python Read Text File", pythontutorial.net [Online].

Available: https://www.pythontutorial.net/python-basics/python-read-text-file/

[6]. "Android From Scratch: Using REST APIs", code.tutsplus.com [Online].

Available: https://code.tutsplus.com/tutorials/android-from-scratch-using-rest-apis--cms-27117

[7]. "Beautiful Soup Documentation", crummy.com [Online].

Available: https://www.crummy.com/software/BeautifulSoup/bs4/doc/

[8]. "How to scrape PDF files using Python + Requests and Beautiful Soup", youtube.com [Online].

Available: https://youtu.be/VDd6dVrYzao

[9]. "Building a REST API in Python | Home Automation #02", youtube.com [Online].

Available: https://www.youtube.com/watch?v=4T5Gnrmzjak

[10]. "Simple Login App in Android Studio | 2022", youtube.com [Online].

Available: https://www.youtube.com/watch?v=sOJRJtM_iu0&t=680s

[11]. "How to Send HTTP Request and Parse JSON Data Using Java", youtube.com [Online].

Available: https://www.youtube.com/watch?v=qzRKa8I36Ww

[12]. "Deploying React Web App on Firebase – Completely Free | Easy Hosting", youtube.com [Online].

Available: https://www.youtube.com/watch?v=GXr2mn87lA8

[13]. "Python Tutorial: virtualenv and why you should use virtual environments", youtube.com [Online].

Available: https://www.youtube.com/watch?v=N5vscPTWKOk

[14]. "How To Deploy A Django Project To Heroku", youtube.com [Online].

Available: https://www.youtube.com/watch?v=XZoTukqekzY

[15]. "AWS IAM Tutorial | Identity And Access Management (IAM) | AWS Training Videos | Edureka", youtube.com [Online].

Available: https://www.youtube.com/watch?v=UqKWHZ36yEM&t=2221s