JOURNAL OF UNIVERSAL SCIENCE RESEARCH

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-4

A Program for Analyzing the Structure of a Web site Development Using the Parsing Method Based on the Python

Vladyslav Yevsieiev [1], Svitlana Maksymova [1], Ahmad Alkhalaileh [2]

[1] Department of Computer-Integrated Technologies, Automation and Robotics, Kharkiv National University of Radio Electronics, Ukraine

[2] Senior Developer Electronic Health Solution, Amman, Jordan

Abstract:

This article discusses the development of a program in Python to analyze the structure of a website using the parsing method. The work presents the developed algorithm for the program, describes the software and conducts experiments on web site parsing. The program's algorithm includes sending a GET request to a website, receiving and analyzing the HTML code of the page using the BeautifulSoup library. The find_all('a') method was used to analyze the website structure and extract link information. The obtained data was processed and displayed in a convenient format. It allows you to automate the process of analyzing the structure of a website, which can be useful for web developers, SEO specialists and web resource owners.

Key words: Parsing, Website, Structure, Python, Analysis, Manufacturing Innovation, Industrial Innovation.

Introduction

In the modern world, websites are the main communication and information exchange tool for organizations and users [1]-[5]. However, for the effective use and improvement of web resources, it is necessary to analyze their structure. Various methods and approaches can be used here [6]-[15].

The Python programming language is very powerful and is used in a wide variety of areas: from robotics [16]-[31] to website testing [32]-[38].

The development of a program for analyzing the structure of a website using the parsing method based on the Python language is becoming increasingly relevant and necessary.

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-4

The main goal of such a program is to study the internal structure of a website, its components and the connections between them. This allows you to evaluate the ease of use of the site, optimize its navigation and increase the overall efficiency of the web project. Analyzing a website's structure also helps identify errors and vulnerabilities, improve SEO performance, and improve overall accessibility of information.

One of the key tools for analyzing the structure of a website is the parsing method, which allows you to extract and analyze data on web pages. Python is one of the most popular programming languages for creating such programs due to its simplicity, flexibility and rich ecosystem of libraries.

Website structure analysis programs can be useful for both web resource owners and web development and SEO optimization specialists. They allow you to systematize information about the structure of the site, identify problems and propose solutions to eliminate them.

Related works

The problem of website analysis is considered by many scientists. Moreover, many of them use Python language tools. Let's look at some works on this topic.

Alnavar, K., & et al. [32] propose a computer-assisted document parsing tool. The proposed approach employs Python packages such as pytesseract, textblob and beautifulsoup to perform Optical Character Recognition, Translation and Extraction of Hypertext Markup Language data respectively.

The poorly designed data that largely fills up the Internet is difficult to extract and hard to use in an automated process [33]. Web scraping cuts this manual job of extracting information and organizing information and provides an easy-to-use way to collect data from the webpages, convert it into some desired format, and store it in some local repository. The paper also comments on the legality associated with Web scraping at the end.

The author in [34] discusses how Python language with its vast library support implements web scraping to collect data from internet.

Jin, D. in [35] studies the image information collection system based on Python web crawler technology. This paper studies and develops a data acquisition system

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-4

based on Python web crawler technology, which realizes the automatic collection of subject data.

The study [36] proposes an open-source Python-based Machine Learning SEO software tool to the general public, catering to the needs of both website owners and SEO specialists.

Ablahd, A. Z in [37] presents a system that is aimed to detect the web application vulnerabilities before exploited by an attacker. A special scanner was built using python 3.7 built-in tools like AST, CFG, Flask, and Django to detect these vulnerabilities.

The goal of the thesis by Andersson, P. [38] is to build a python-based web scraper that collects data from TimeEdit and saves this in a structured manner.

We see that the use of the Python language is widespread in various areas of website testing, as well as working with information in general. Later in this article we will consider a program for analyzing the structure of a web site development using the parsing method based on the Python.

An Algorithm and Program Development for the Web site structure analyzing

Website parsing is the process of extracting and analyzing data contained in web pages. It is often used to automate information collection, content analysis, or monitoring changes to a site. Parsing may involve extracting text, images, links, metadata, and other page elements.

Various technologies are often used to scrape websites, such as Python libraries (for example, BeautifulSoup, Scrapy), which allow you to programmatically access the HTML code of the page and extract the necessary data. Parsing can be useful for a variety of tasks, from creating archives of news articles to analyzing product prices in online stores.

One of the key aspects of website parsing is accessing pages in accordance with the site's rules of use (robots.txt) and avoiding overloading the server with requests. Parsing can also involve processing dynamic page elements such as JavaScript to fully extract data.

Through website parsing and subsequent data analysis, you can gain valuable information for a variety of purposes, from market research to improving SEO

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-4

strategies. However, you must be aware of laws and data usage regulations to avoid violating copyrights or site rules.

As a base site for researching parsing methods to analyze the structure of a Web site and all links and URLs, we will use the site of the Department of Computer-Integrated Technologies, Automation and Robotics (CITAR) Kharkiv National University of Radioelectronics located at https://tapr. nure.ua/en/.

Before developing a program for parsing the Web site https://tapr.nure.ua/en/ based on the Python language, the following algorithm was developed, which is presented in Figure 1.

As you can see from Figure 1, the following libraries will be used in the developed program:

– requests is used to make HTTP requests in the Python programming language. Using requests allows you to retrieve web page content, make POST requests, pass parameters, manage cookies, and other aspects of HTTP requests. It is convenient for working with website APIs, obtaining data for parsing or interacting with web servers.

– BeautifulSoup is a tool for parsing HTML and XML documents in Python. It allows you to parse incoming data from web pages and conveniently extract the desired elements using CSS selectors, XPath or other methods. BeautifulSoup makes HTML parsing simple and efficient, and is used in many projects to retrieve data from the Internet.

```
pip install requests
pip install beautifulsoup4
```

The first command will install the requests library, and the second will install beautifulsoup4. Once these libraries are installed, you can import them into your Python programs and use their functionality.

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-4

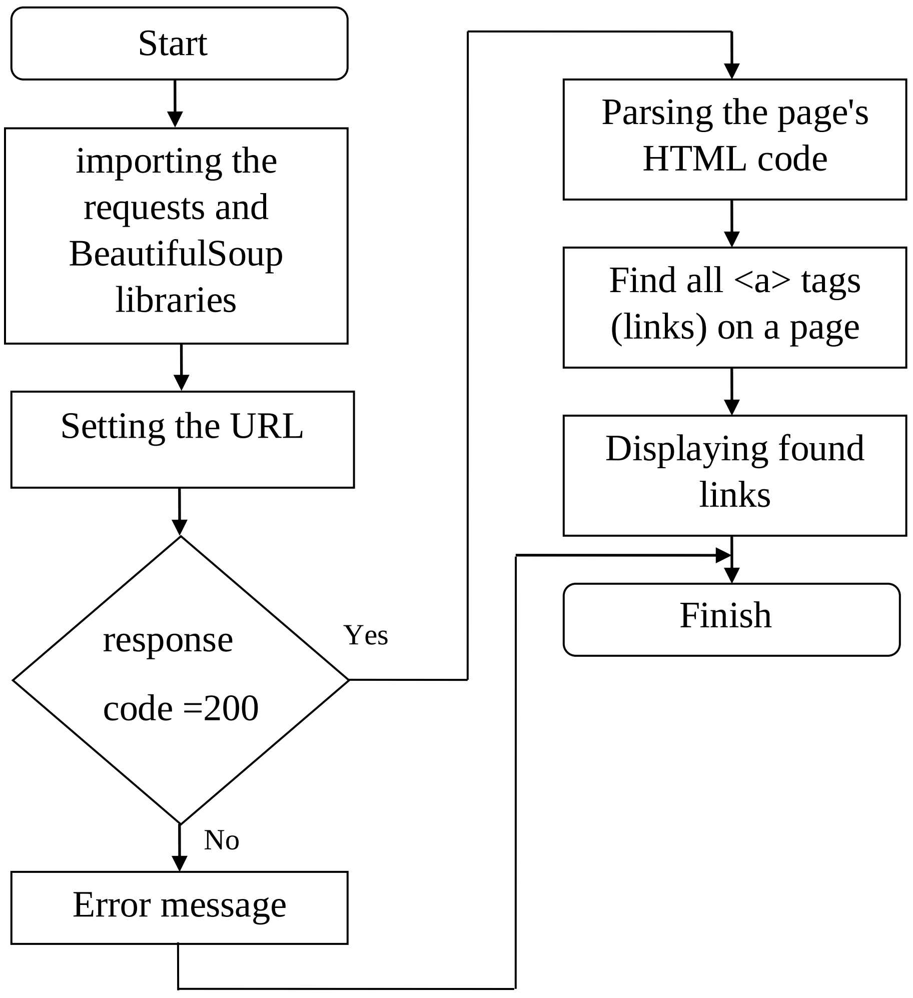

Start
importing the requests and BeautifulSoup libraries
Setting the URL
response code =200
Yes
No
Error message
Parsing the page's HTML code
Find all <a> tags (links) on a page
Displaying found links
Finish

Figure 1: The Web Site https://tapr.nure.ua/en/ parsing program algorithm

The developed software code for parsing the Web site https://tapr.nure.ua/en/ is presented below with an explanation.

```
import requests
from bs4 import BeautifulSoup
```

This code snippet imports two libraries for working with web pages in Python: requests and BeautifulSoup.

requests is a library that allows you to interact with websites, send requests to the server and receive responses. Thanks to this library, you can get the HTML code of the page you want to analyze.

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-4

BeautifulSoup is a library for working with HTML code. It allows you to parse (analyze) HTML code and extract the necessary data from it, such as texts, links, tables, etc. Thanks to this library, you can create a structured object from the HTML code of the page that is easy to parse and process.

```
# URL of the page to be parsed
url = ' https://tapr.nure.ua/en/'
```

his piece of code defines the page URL that we want to parse. It points to the specific web resource from which we want to retrieve data. This URL will later be used in function calls to send requests and receive a response from the server.

```
# Send a GET request to the page and receive a response
response = requests.get(url)
```

This piece of code sends a GET request to the specified URL using the requests library and receives a response from the server. A GET request is used to retrieve data from the server and in this case we are using it to retrieve the content of a page at a given URL. The result of this request is a response object containing various information about the server response, such as status code, headers, and page content.

```
# Check if we received a successful response (code 200)
if response.status_code == 200:
    # Parsing a page using BeautifulSoup
    soup = BeautifulSoup(response.text, 'html.parser')
```

This piece of code checks if we have received a successful response from the server. Code 200 means that the request was successfully processed by the server and returned a response. If the response is successful, then further parsing of the page is performed using the BeautifulSoup library. In this case, response.text contains the HTML code of the page, which is passed to the BeautifulSoup constructor for further parsing.

```
    # Find all <a> tags (links) on the page
     links = soup.find_all('a')
```

This code snippet uses the find_all() method of the BeautifulSoup object to find all <a> tags (links) on the page. The find_all() method returns a list of all items found that match the specified search criteria. In this case, we are looking for a link on the page for further analysis or processing.

```
# Display found links
```

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-4

```
for link in links:
    print(link.get('href'))
```

This code snippet displays the found links on the page. It iterates through each element of the links list (containing the <a> tags it finds), and for each element calls the get('href') method to get the value of the href attribute (the URL of the link) and display it on the screen. This will give you a list of URLs pointing to the various pages that are linked to on that page.

```
else:
    print('Error retrieving page:', response.status_code)
```

This code snippet displays an error message if the response from the server has a status code other than 200. Otherwise, if the status code is not 200, an error message is printed and the status code itself is provided as additional information to the user. This approach allows you to track errors while retrieving a page and provides a convenient way to debug if you encounter problems retrieving data from a website. A general view of the Web site https://tapr.nure.ua/en/ taken to study parsing methods to analyze the structure of the Web site and all links and URLs is presented in Figure 2.

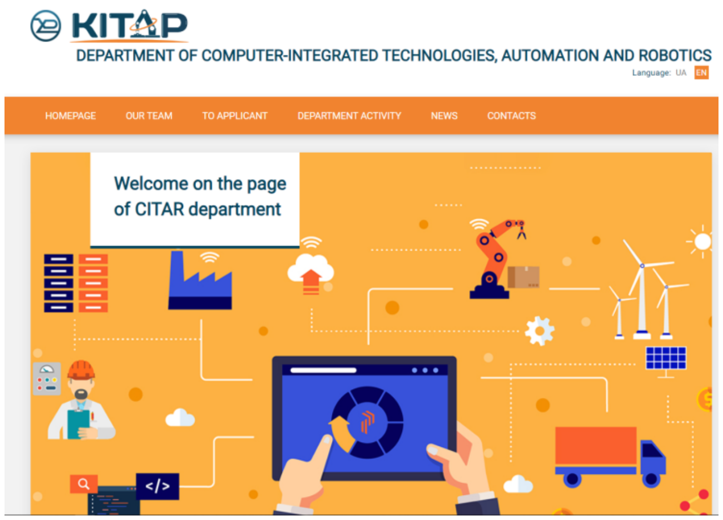
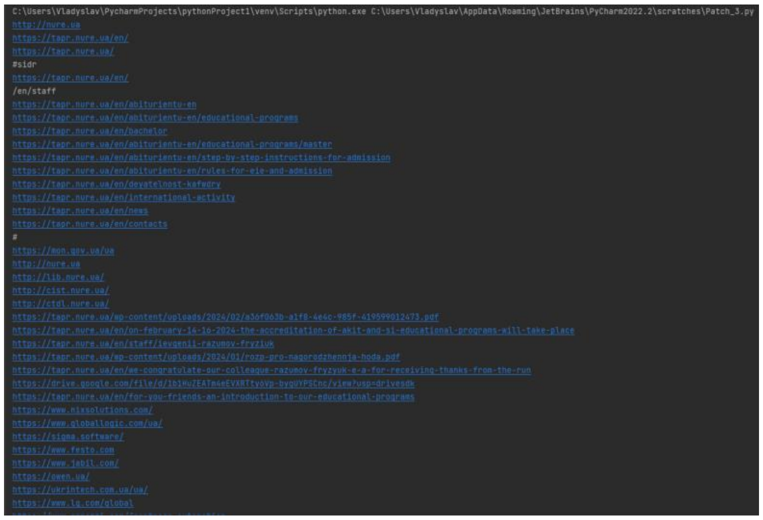

KITAP
DEPARTMENT OF COMPUTER-INTEGRATED TECHNOLOGIES, AUTOMATION AND ROBOTICS
Language: UA EN
HOMEPAGE
OUR TEAM
TO APPLICANT
DEPARTMENT ACTIVITY
NEWS
CONTACTS
Welcome on the page of CITAR department

Figure 2: General view of the Web site https://tapr.nure.ua/en/

The results of parsing the website https://tapr.nure.ua/en/ based on the Python language are presented in Figure 3

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-4



```
C:\Users\Vladyslav\PycharmProjects\pythonProject1\venv\Scripts\python.exe C:\Users\Vladyslav\AppData\Roaming\JetBrains\PyCharm2022.2\scratches\Patch_3.py
http://nure.ua
https://tapr.nure.ua/en/
https://tapr.nure.ua/
#sidr
https://tapr.nure.ua/en/
/en/staff
https://tapr.nure.ua/en/abiturientu-en
https://tapr.nure.ua/en/abiturientu-en/educational-programs
https://tapr.nure.ua/en/bachelor
https://tapr.nure.ua/en/abiturientu-en/educational-programs/master
https://tapr.nure.ua/en/abiturientu-en/step-by-step-instructions-for-admission
https://tapr.nure.ua/en/abiturientu-en/rules-for-eie-and-admission
https://tapr.nure.ua/en/deyatelnost-kafedry
https://tapr.nure.ua/en/international-activity
https://tapr.nure.ua/en/news
https://tapr.nure.ua/en/contacts
#
https://mon.gov.ua/ua
http://nure.ua
http://lib.nure.ua/
http://cist.nure.ua/
http://ctdl.nure.ua/
https://tapr.nure.ua/wp-content/uploads/2024/02/a36f063b-a1f8-4e4c-985f-419599012473.pdf
https://tapr.nure.ua/en/on-february-14-16-2024-the-accreditation-of-akit-and-si-educational-programs-will-take-place
https://tapr.nure.ua/en/staff/ievgenii-razumov-fryziuk
https://tapr.nure.ua/wp-content/uploads/2024/01/rozp-pro-nagorodzhennja-hoda.pdf
https://tapr.nure.ua/en/we-congratulate-our-colleague-razumov-fryzyuk-e-a-for-receiving-thanks-from-the-run
https://drive.google.com/file/d/1b1HuZEATm4eEVXRTtyoVp-bygUYPSCnc/view?usp=drivesdk
https://tapr.nure.ua/en/for-you-friends-an-introduction-to-our-educational-programs
https://www.nixsolutions.com/
https://www.globallogic.com/ua/
https://sigma.software/
https://www.festo.com
https://www.jabil.com/
https://owen.ua/
https://ukrintech.com.ua/ua/
https://www.lg.com/global
```

Figure 3: Results of parsing the Web site https://tapr.nure.ua/en/

Conclusion

During the development of a program for analyzing the structure of a website using the parsing method in Python, a detailed analysis of the website of the Department of Computer-Integrated Technologies, Automation and Robotics (CITAR) of the Kharkov National University of Radioelectronics (KNURE) was carried out at https://tapr. nure.ua/en/.

The program was written using the requests libraries to send HTTP requests and BeautifulSoup to parse the HTML code of the page. After sending a GET request and receiving a response from the website, the program analyzed the page structure, extracting all links (<a> tags) using the find_all('a') method and outputting their URLs.

As a result, it was revealed that the CITAR KNURE website contains a variety of information about the department, current news and events, as well as useful resources and contact information. Analysis of the website structure allowed us to evaluate its ease of use and navigation for users.

The program also detected and analyzed all internal and external links on site pages, which can be useful for SEO optimization and web traffic analysis. Finding broken links or outdated information can help you update content and improve the user experience.

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-4

References:

1. Omarov, M., Tikhaya, T., & Lyashenko, V. (2019). Internet marketing metrics visualization methodology for related search queries. International Journal of Advanced Trends in Computer Science and Engineering, 8(5), 2277-2281.
2. Z. Deineko, "Features of Database Types," International Journal of Engineering and Information Systems (IJEAIS), 2021, vol. 5, issue 10, pp. 73-80.
3. S. Sotnik, T. Shakurova, and V. Lyashenko, “Development Features Web-Applications,” vol. 7, no. 1, pp. 79–85, 2023.
4. Sotnik, S., Manakov, V. and Lyashenko, V. (2023), “Overview: PHP and MySQL features for creating modern web projects”, International Journal of Academic Information Systems Research, Vol. 7, No. 1, pp. 11-17.
5. Omarov, M., TYKHA, T., & Lyashenko, V. (2019). Use of Wavelet Techniques in the Study of Internet Marketing Metrics. Eskişehir Technical University Journal of Science and Technology A-Applied Sciences and Engineering, 20, 157-163.
6. Abu-Jassar, A. T., Attar, H., Yevsieiev, V., Amer, A., Demska, N., Luhach, A. K., & Lyashenko, V. (2022). Electronic User Authentication Key for Access to HMI/SCADA via Unsecured Internet Networks. Computational intelligence and neuroscience, 2022, 5866922.
7. Al-Sherrawi, M. H., Lyashenko, V., Edaan, E. M., & Sotnik, S. (2018). Corrosion as a source of destruction in construction. International Journal of Civil Engineering and Technology, 9(5), 306-314.
8. Vasiurenko, O., Lyashenko, V., Baranova, V., & Deineko, Z. (2020). Spatial-Temporal Analysis the Dynamics of Changes on the Foreign Exchange Market: an Empirical Estimates from Ukraine. Journal of Asian Multicultural Research for Economy and Management Study, 1(2), 1-6.
9. Lyashenko, V., Matarneh, R., & Kobylin, O. (2016). Contrast modification as a tool to study the structure of blood components. Journal of Environmental Science, Computer Science and Engineering & Technology, 5(3), 150-160.
10. Nevliudov, I., Yevsieiev, V., Lyashenko, V., & Ahmad, M. A. (2021). GUI Elements and Windows Form Formalization Parameters and Events Method to

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-4

Automate the Process of Additive Cyber-Design CPPS Development. Advances in Dynamical Systems and Applications, 16(2), 441-455.

11. Mustafa, S. K., Yevsieiev, V., Nevliudov, I., & Lyashenko, V. (2022). HMI Development Automation with GUI Elements for Object-Oriented Programming Languages Implementation. SSRG International Journal of Engineering Trends and Technology, 70(1), 139-145.

12. Babker, A., & Lyashenko, V. (2018). Identification of megaloblastic anemia cells through the use of image processing techniques. Int J Clin Biomed Res, 4, 1-5.

13. Mustafa, S. K., Lyashenko, V., Ameer Ahamad, N., Rehan, M., & Ajmal, A. A. (2021). Some aspects of modeling in the study of COVID-19 data. International Journal of Pharmaceutical Research, 4124-4129.

14. Kuzomin, O., Lyashenko, V., Dudka, O., Radchenko, V., & Vasylenko, O. (2020). Using of ontologies for building databases and knowledge bases for consequences management of emergency. International Journal of Advanced Trends in Computer Science and Engineering, 9(4), 5040-5045.

15. Khan, A., Ahmad, M., Joshi, S., & Lyashenko, V. (2016). Synthesis of Alumina Fibre by Annealing Method using Coir Fibre. American Chemical Science Journal, 15(2), 1-7.

16. Al-Sharo Y., & et al. (2023). A Robo-hand prototype design gripping device within the framework of sustainable development. Indian Journal of Engineering, 2023, 20, e37ije1673.

17. Yevsieiev, V., & et al. (2024). The Canny Algorithm Implementation for Obtaining the Object Contour in a Mobile Robot's Workspace in Real Time. Journal of Universal Science Research, 2(3), 7-19.

18. Shcherbyna, V., & et al. (2023). Mobile Robot for Fires Detection Development. Journal of Universal Science Research, 1(11), 17-27.

19. Yevsieiev, V., & et al. (2024). Object Recognition and Tracking Method in the Mobile Robot's Workspace in Real Time. Technical Science Research In Uzbekistan, 2(2), 115-124.

20. Basiuk, V., & et al. (2023). Mobile Robot Position Determining Using Odometry Method. Multidisciplinary Journal of Science and Technology, 3(3), 227-234.

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-4

21. Yevsieiev, V., & et al. (2024). Active Contours Method Implementation for Objects Selection in the Mobile Robot's Workspace. Journal of Universal Science Research, 2(2), 135-145.

22. Nikitin, V., & et al (2023). Traffic Signs Recognition System Development. Multidisciplinary Journal of Science and Technology, 3(3), 235-242.

23. Yevsieiev, V., & et al. (2024). Using Contouring Algorithms to Select Objects in the Robots' Workspace. Technical Science Research In Uzbekistan, 2(2), 32-42.

24. Nevliudov, I., & et al. (2023). Mobile Robot Navigation System Based on Ultrasonic Sensors. In 2023 IEEE XXVIII International Seminar/Workshop on Direct and Inverse Problems of Electromagnetic and Acoustic Wave Theory (DIPED), IEEE, 1, 247-251.

25. Yevsieiev, V. V., & et al. (2023). Conveyor Belt Object Identification: Mathematical, Algorithmic, and Software Support. Appl. Math. Inf. Sci, 17, 1073-1088.

26. Al-Sharo, Y. M., Abu-Jassar, A. T., Sotnik, S., & Lyashenko, V. (2023). Generalized Procedure for Determining the Collision-Free Trajectory for a Robotic Arm. Tikrit Journal of Engineering Sciences, 30(2), 142-151.

27. Lyashenko, V., Laariedh, F., Ayaz, A. M., & Sotnik, S. (2021). Recognition of Voice Commands Based on Neural Network. TEM Journal: Technology, Education, Management, Informatics, 10(2), 583-591.

28. Lyashenko, V., & Sotnik, S. (2022). Overview of Innovative Walking Robots. International Journal of Academic Engineering Research (IJAER), 6(4), 3-7.

29. Attar, H., Abu-Jassar, A. T., Lyashenko, V., Al-qerem, A., Sotnik, S., Alharbi, N., & Solyman, A. A. (2023). Proposed synchronous electric motor simulation with built-in permanent magnets for robotic systems. SN Applied Sciences, 5(6), 160.

30. Lyashenko, V., & et al.. (2021). Semantic Model Workspace Industrial Robot. International Journal of Academic Engineering Research, 5(9), 40-48.

31. Sotnik, S., Mustafa, S. K., Ahmad, M. A., Lyashenko, V., & Zeleniy, O. (2020). Some features of route planning as the basis in a mobile robot. International Journal of Emerging Trends in Engineering Research, 8(5), 2074-2079.

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-4

32. Alnavar, K., & et al. (2021). Document Parsing Tool for Language Translation and Web Crawling using Django REST Framework. In Journal of Physics: Conference Series, IOP Publishing, 1962(1), 012018.

33. Nigam, H., & Biswas, P. (2021). Web scraping: from tools to related legislation and implementation using python. In Innovative Data Communication Technologies and Application: Proceedings of ICIDCA 2020, Springer Singapore, 149-164.

34. Kumar, S., & Roy, U. B. (2023). A technique of data collection: web scraping with python. In Statistical Modeling in Machine Learning, Academic Press, 23-36.

35. Jin, D. (2021). Image information collection system based on Python Web crawler technology. converter, 606-612.

36. Roumeliotis, K. I., & Tselikas, N. D. (2023). A Machine Learning Python-Based Search Engine Optimization Audit Software. In Informatics, MDPI, 10(3), 68.

37. Ablahd, A. Z. (2023). Using python to detect web application vulnerability. Res Militaris, 13(2), 1045-1058.

38. Andersson, P. (2021). Developing a Python based web scraper: A study on the development of a web scraper for TimeEdit.