SECTION 6.
INFORMATION TECHNOLOGIES AND SYSTEMS

Vladyslav Yevsieiev
Doctor of Engineering Science,
Professor of Department of Computer-Integrated Technologies,
Automation and Robotics
Kharkiv National University of Radio Electronics, Ukraine

Stanislav Uskov
Software Engineer, Regulatory Solutions
AxiomSL
Ukraine

THE SOLUTION TO THE TASK OF ANALYZING THE RELIABILITY OF WEB SITES USING PYTHON PARSING

There are several important reasons for conducting a web page reliability analysis. First, the analysis allows you to make sure that the page is accessible to users, ensuring its incompleteness and efficiency. Second, it helps detect errors, such as code errors, structural issues, or server issues, that may occur and cause disruptions in page accessibility or functionality. In addition, performance analysis, including page load time, allows you to evaluate the efficiency and speed of the page, which is important for the user experience. Security analysis allows you to identify vulnerabilities, such as cross-site scripting or SQL injection vulnerabilities, that can cause security breaches and leak confidential information. Finally, monitoring the stability of the connection with the server helps to monitor the stability of the page and respond in time to possible problems with the infrastructure or network. All these aspects of website reliability analysis work together to ensure a high level of functionality, availability, performance and security for users.

Analyzing the reliability of a web page using a Python-based parser has its own characteristics that are important to consider. First, Python provides a wide selection of libraries for working with web pages, such as requests for sending HTTP requests, BeautifulSoup for parsing HTML code, and others. This makes it easy to access and analyze page content. Second, Python has a convenient and understandable syntax that simplifies the development of scripts for analyzing web pages. A third feature is the extensive support of the Python community, which provides many useful packages, documentation, and code examples for a variety of web parsing tasks. In addition, Python can be used to automate routine tasks, such as regularly performing page reliability analysis or automating the process of monitoring connection stability. Another feature is the possibility of integration with other tools and services to obtain a wider range of functionality, for example, saving analysis results in a database or integration with web page monitoring services. Overall, Python is a powerful and convenient tool for web site reliability analysis that allows the process to be done quickly, efficiently, and conveniently.

Requirements for the program: the program must display the results of the analysis: the status code of the server's response; page loading time; the number of requests and responses on the page; page name; page meta description. Resources: information on the store page

https://www.mi.com/ua/; requests library for sending HTTP requests.; the BeautifulSoup library for parsing the HTML code of the page; a Python file system for writing data to a CSV file. A fragment of the software code for the implementation of a passer for checking the reliability of a Web page is presented in Figure 1.

```python
import requests
from bs4 import BeautifulSoup


def analyze_webpage(url):
    try:
        # Відправляємо запит на сервер і отримуємо відповідь
        response = requests.get(url)

        # Перевіряємо статус-код відповіді сервера
        if response.status_code == 200:
            print("Server response status code: 200 (Successful)")
        else:
            print(f"Server response status code: {response.status_code}")

        # Отримуємо час завантаження сторінки
        load_time = response.elapsed.total_seconds()
        print(f"Page load time: {load_time} сек")

        # Отримуємо HTML-код сторінки
        html_content = response.content
        soup = BeautifulSoup(html_content, 'html.parser')

        # Отримуємо кількість запитів і відповідей на сторінці
        requests_count = len(soup.find_all())
        print(f"The number of requests and responses per page: {requests_count}")

        # Вивчаємо метадані сторінки
        title = soup.title.text if soup.title else "The page title is missing"
        meta_description = soup.find('meta', {'name': 'description'})
        description = meta_description['content'] if meta_description else "There is no meta description"
        print(f"Name of the page: {title}")
        print(f"Meta description of the page: {description}")
```

Fig. 1. **A fragment of the software code for the implementation of a passer for checking the reliability of a Web page**

As an example of the obtained result of the analysis of the reliability of the web page https://www.mi.com/ua using the developed parsing program, shown in Figure 2.

```
Server response status code: 200 (Successful)
Page load time: 0.128143 сек
The number of requests and responses per page: 1069
Name of the page: Xiaomi Україна | Офіційний сайт Xiaomi
Meta description of the page: Вітаємо на офіційному сайті Xiaomi Україна! Тут є все про нові телефони Xiaomi і Redmi, пристрої для розумної оселі та інші популярні товари.
```

Fig. 2. **The obtained results of the reliability analysis of the web page https://www.mi.com/ua using the developed parsing program**

Conclusions. Solutions to the problem of website reliability analysis using Python language parsing have shown that website parsing is an effective tool for analyzing their reliability. Using Python allows you to automate the data collection process and reduce manual analysis time. Testing on the website https://www.mi.com/ua demonstrated high accuracy and speed of obtaining the necessary information. The results showed that parsing can be used to monitor changes on the site and identify possible problems with its functioning. Thus, Python parsing is an effective method for ensuring the reliability of web resources.

References:

1. Zubiena, Luke, et al. "Development and testing of the health information website evaluation tool on neck pain websites–An analysis of reliability, validity, and utility." Patient Education and Counseling 113 (2023): 107762.
2. Aminuddin, M. A. I. M., Zaaba, Z. F., Samsudin, A., Zaki, F., & Anuar, N. B. (2023). The rise of website fingerprinting on Tor: Analysis on techniques and assumptions. Journal of Network and Computer Applications, 212, 103582.
3. Kumar, B., Roy, S., Singh, K. U., Pandey, S. K., Kumar, A., Sinha, A., ... & Rasool, A. (2023). A static machine learning based evaluation method for usability and security analysis in e-commerce website. IEEE Access, 11, 40488-40510.
4. William, P., Singh, S., Kumar, Y., Rao, A. K., Khan, A. K., & Singh, D. (2023, July). Software Reliability Analysis with Various Metrics using Ensembling Machine Learning Approach. In 2023 World Conference on Communication & Computing (WCONF) (pp. 1-6). IEEE.
5. Modarres, M., & Groth, K. (2023). Reliability and risk analysis. CRC Press.
6. Venkatakrishnan, J., Alagiriswamy, R., & Parayitam, S. (2023). Web design and trust as moderators in the relationship between e-service quality, customer satisfaction and customer loyalty. The TQM Journal, 35(8), 2455-2484.
7. Bertolino, A., Angelis, G. D., Guerriero, A., Miranda, B., Pietrantuono, R., & Russo, S. (2023). DevOpRET: Continuous reliability testing in DevOps. Journal of Software: Evolution and Process, 35(3), e2298.
8. Web site reliability analysis using the python parsing method / D. Gurin, S. Maksymova, V. Yevsieiev, Ahmad Alkhalaileh // Journal of Universal Science Research, 2024, 2(5). – P. 113-126.
9. Yevsieiev V. A Program for Analyzing the Structure of a Web site Development Using the Parsing Method Based on the Python / V. Yevsieiev, S. Maksymova, Ahmad Alkhalaileh // Journal of Universal Science Research, 2024, 2(4). – P. 172-183.
10. Abu-Jassar AT, Attar H, Amer A, et al. Remote Monitoring System of Patient Status in Social IoT Environments Using Amazon Web Services (AWS) Technologies and Smart Health Care. International Journal of Crowd Science, 2024, https://doi.org/10.26599/IJCS.2023.9100019.
11. Abu-Jassar AT, Attar H, Amer A, et al. Development and Investigation of Vision System for a Small-Sized Mobile Humanoid Robot in a Smart Environment. International Journal of Crowd Science, 2024, https://doi.org/10.26599/IJCS.2023.9100018.
12. Lyashenko, V., Abu-Jassar, A.T., Yevsieiev, V., Maksymova, S. Automated Monitoring and Visualization System in Production, Int. Res. J. Multidiscip. Technovation, 5(6) 2023 09-18. https://doi.org/10.54392/irjmt2362.
13. Al-Sharo, Y., Abu-Jassar, A., Lyashenko, V., Yevsieiev, V., Maksymova, S. A Robo-hand prototype design gripping device within the framework of sustainable development, Indian Journal of Engineering, 20 2023 e37ije1673. https://doi.org/10.54905/disssi.v20i54.e37ije1673.
14. Моделі та методи кіберфізичних виробничих систем в концепції Industry 4.0 : монографія / І. Ш. Невлюдов, В. В. Євсєєв, А. О. Андрусевич, С. С. Максимова ; – Oktan Print – Prague. 2023. – 321 с.
15. Svitlana Maksymova, Vladyslav Yevsieiev, & Amer Abu-Jassar. (2024). The Bipedal Robot a Kinematic Diagram Development. Journal of Universal Science Research, 2(1), 6–17.