Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 2, April 2024
DOI : https://doi.org/10.33395/v8i2.13569

e-ISSN : 2541-2019
p-ISSN : 2541-044X

Web Program Testing Using Selenium Python: Best Practices and Effective Approaches

Rusdiansyah[1)*], **Nining Suharyanti**[2)], **Hendra Supendar**[3)], **Tuslaela**[4)]
[1,2,3)]Universitas Bina Sarana Informatika, [4]Universitas Nusa Mandiri, Indonesia
[1)] rusdiansyah.rds@bsi.ac.id, [2)] nining.nni@bsi.ac.id, [3)] hendra.hds@bsi.ac.id, [4)] tuslaela.tll@nusamandiri.ac.id,

gelesen (nicht mitteilen)

Submitted :Mar 10, 2024 | **Accepted** : Mar 29, 2024 | **Published** : Apr 1, 2024

Abstract: The use of Selenium to automate the login process, goods transactions, and filling in customer data has become the main focus in developing efficient and reliable web applications. The research discusses the implementation of Selenium in the context of the website www.saucedemo.com, which offers an interactive and efficient online shopping experience. First, we discuss the login process which involves automatically filling in the username and password using Selenium. Element identification techniques such as ID, class name, and CSS selector are used to ensure the login process runs smoothly and safely. Next, we explore the item transaction process, where Selenium enables item selection from the inventory page and addition to the shopping cart quickly and efficiently. The use of automatic wait and click methods ensures this process runs without a hitch. In addition, this article discusses automatically filling in customer data on the first checkout page (checkout-step-one.html). We highlight the importance of accurate and secure data entry using Selenium, as well as effective input element identification strategies. In conclusion, using Selenium in online shopping process automation brings many benefits, including efficiency, consistency, and improved user experience. Apart from that, Selenium is also used for automated testing to ensure the quality and reliability of important functions on the website. This article provides comprehensive insight into how automation technologies such as Selenium can improve efficiency and quality in managing online shopping transactions.

Keywords: Selenium WebDriver; Filling in the text box; Python; Test automation; Website

INTRODUCTION

Websites are now the primary means by which companies, organizations, and individuals communicate with their online audiences in the ever evolving digital age we live in (Chitra, 2019). Program testing is a process that goes unnoticed but is extremely important, hidden beneath the elegant design and intricate functionality. A crucial phase in the software development cycle is website program testing(Setiawan, 2023). This is a measure to guarantee that the website can offer a positive user experience in addition to good functionality (Mustika, 2024). Program testing benefits businesses by reducing the chance of failure, boosting client confidence, and guaranteeing adherence to relevant laws and industry standards (Silalahi, 2023).

This research endeavors to explore several approaches for program testing websites, concentrating on the application of Selenium with Python. In order to automate website functionality, performance, and security testing, it is important to comprehend how successful these technologies are (James W, 20 C.E.). Using Selenium Python, this study also attempts to determine best methods for online program testing. The following areas will be investigated in this study in an attempt to meet the goal.

1. Selenium Python is used for website functionality testing.
   Testing the functioning of websites with Selenium Python automated testing is done with the popular web browser automation framework Selenium. With Selenium, you can test how a website works by clicking on links, completing forms, confirming content, and more.
2. Automated tools for testing website performance.
   In order to make sure that a website can offer a positive and responsive user experience, particularly when dealing with large traffic loads, testing website performance utilizing automation tools is a crucial step. In this regard, page load time, server response, and website stability can all be measured in a variety of scenarios using automation tools like Selenium.
3. Automated methods of assessing websites for security.

*name of corresponding author

4. List the recommended practices for website testing programs.

In order to improve the quality and security of websites in this rapidly changing digital world, software developers, researchers, and practitioners of informatics engineering could get useful insights from this research by addressing these problems

LITERATURE REVIEW

This research's literature review attempts to look at works about program testing on websites, with a focus on using Selenium with Python (Wahyudin, 2020). To create a solid research foundation, this study critically analyzes a number of hypotheses, studies, and associated discoveries (Wekke, 2019). Program testing on websites has been the subject of several prior studies that have used a range of methodologies and resources. Numerous studies emphasize how crucial it is to evaluate a website's operation to make sure it performs as end users would anticipate. But as websites get more complicated, problems appear, necessitating a more automated and effective testing strategy.

Recent research in internet testing systems has focused on the usage of Selenium, particularly when combined with the Python programming language (Avrianto, 2022). Automating user interactions on websites is possible with Selenium, a test automation tool (Fauzan, 2023). Website functionality testing can be completed automatically with Selenium, which expedites the testing procedure and improves result accuracy (Wendy Winata, 2022).

However, even though Selenium offers various advantages, there are still several obstacles that need to be overcome. Several studies highlight the challenges of automating website performance and security testing with Selenium. Additionally, it is necessary to identify best practices in using Selenium for program testing on websites to maximize its potential(Rees, 2007).

Though selenium has many benefits, there are still a number of challenges that must be solved. Numerous studies draw attention to the difficulties in using Selenium to automate security and performance testing of websites. To fully utilize Selenium for program testing on websites, it is also essential to determine best practices (Fahrizal, 2023). by analyzing this review of the literature. This study will close this gap by examining how well Selenium Python performs when used to test the functionality, performance, and security of websites (Fahrizal, 2023), and by figuring out the best ways to test programs on websites that use this tool.

The following are examples of earlier studies that the author consulted in order to compare this research with it:

1. Research conducted by Amanny Ulfah, dkk, 2023. The purpose of this study is to create a webpage with information specifically related to academic references that can assist students at Bakti Asih High School and researchers in completing reference checks for academic articles. This system's implementation uses web scraping techniques to gather data for academic articles. It uses Python programming language and libraries such as BeautifulSoup, Openpyxl, Selenium, and Pandas. Results of data scraping in the form of an Excel file created on a web page(Ulfah, 2023).
2. Study carried out in 2022 by Albert Stevan Yondra and colleagues. The goal of this project is to create a web scraping program that gathers data from multiple Indonesian e-commerce sites by utilizing parallel processing and multithreading techniques. The number of threads that can be used ranges from one to twenty. It was discovered that the best scraping procedure used five threads based on test results with varying numbers of threads(Yondra, 2022).
3. Study carried out in 2023 by Apriza Zicka Rizquina and colleagues. This study discovered that Shopee's bot and CAPTCHA detection features prevent the HTML Parsing and CSS Selector techniques from being used for web scraping. Tokopedia has, nevertheless, shown success with this approach. Ten web scraping attempts were made on two product pages, yielding 160–166 data points apiece, with 3–18 data duplications. A program runs for an average of one minute and fifty-five seconds(Rizquina, 2023).

METHOD

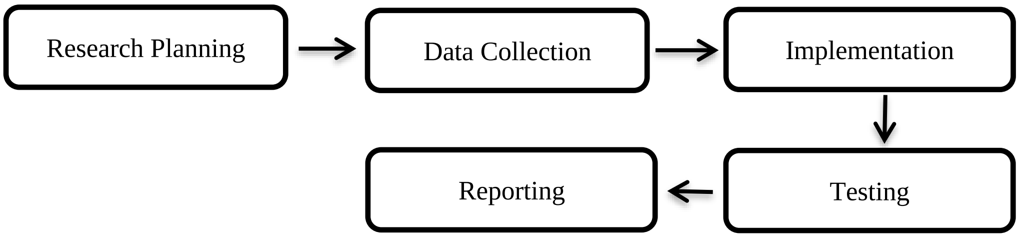


Figure 1. Stages in the pelenium feasibility testing methodology

*name of corresponding author

In Figure 1. This section elucidates the methodology employed to address the issues raised in this study(Assyakurrohim, 2022). outlines the procedures used to look at the viability of testing websites' programs using Selenium and the Python programming language(Muhammad Ramadan Fikri, 2022).

Research Planning:

a. Identify the research objectives and problems to be solved.
b. Creation of a research framework to organize the steps to be taken.Pengumpulan Data:

Data Collection:

a. Gathering data regarding Python and Selenium as research tools and programming languages.
b. Learn how to utilize Selenium for website test automation by reading through the documentation and other online resources.

Implementation:

a. Setting up the Python programming language and installing Selenium WebDriver.
b. Creating test scripts for several elements of website testing, such as functionality, performance, and security, using Python and Selenium WebDriver.

Testing:

a. utilizing scripts designed to assess the dependability and efficiency of utilizing Selenium Python in website testing programs, conduct a number of tests.
b. Examine test findings to determine the methods' advantages and disadvantages.

Evaluation and Interpretation:

a. Interpret test results and see whether research objectives have been achieved.
b. Analyze the suitability of the proposed solution to the problems discussed.

Reporting:

a. Preparation of research reports that include findings, analysis and conclusions from tests that have been carried out.
b. Use pictures, diagrams and flow charts if necessary to explain solutions to the problems discussed.

By following the above methods, you can contribute the most recent ideas for solving issues with Selenium

RESULT

Test scenarios are intended to make testers more focused. Test scenarios include those described in the test plan.

Table 1. Idea Selenium Testing Scenarios

Test Plan Name	Number of tests	Testing
Login User_id	2	Selenium ideas
Click Add to Cart	2	Selenium ideas
Click CheckOut	1	Selenium ideas
Fill in customer data	1	Selenium ideas

In Table 1. In the login test, the pseudocode for the Selenium program that logs in to www.saucedemo.com using the username "standard_user" and the password "secret_sauce"

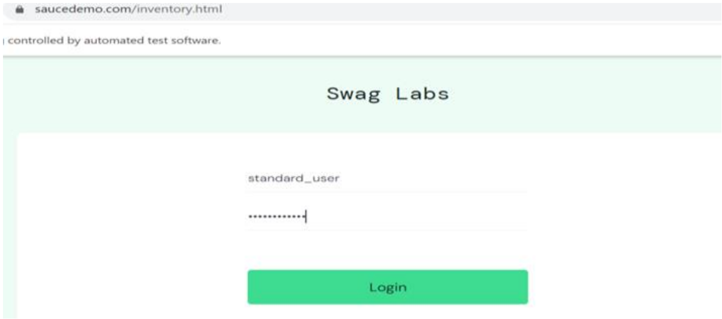
In Pseudocede Selenium python program to login as follows:

```
Start
    Initialize WebDriver with Chrome
    Try
        Go to the page www.saucedemo.com
        Wait until the username input is visible (maximum 10 seconds)
        Enter "standard_user" into the username input
        Find the password input using ID
        Enter "secret_sauce" into the password input
        Find the Login using ID button
        Click the Login button
  Wait until the URL changes to "https://www.saucedemo.com/inventory.html" (maximum 10 second)
        Print "Login successful!"
    Catch Exception
        Print "Login failed:", error message
    Finally
```

*name of corresponding author

```
        Close the browser when finished
        Close WebDriver
Finished
```

Login Output Display



saucedemo.com/inventory.html

controlled by automated test software.

Swag Labs

standard_user

Gambar 2. Login User-Id

In Fig. 2. The outcomes of each stage in the login procedure will determine how this application outputs. An explanation of each potential output is provided below:

a) The software will produce the following message if the login is successful: "Login successful!" This shows that the login procedure was successful and that the user was able to access the inventory page following their login.
b) If Login Error: The software will produce the following message if there is a problem with the login process: less
Login failed: [error message]
Details regarding the precise cause of the unsuccessful login attempt will be included in the error message. An error message might, for instance, notify about a network issue or an element that was not found.
c) Print Login Status: The application prints the message "Login successful!" if the login process is successful. The program prints the message "Login failed:" followed by the specific error message if there is a problem with the login process.

Table 2. Login Test Results

Test design	Number of Tests	Succeed	Fail
Login Admin dan User	2	2	

In Table 2. In the end, the program's output tells us whether the login procedure succeeded or failed and offers some insight into any potential reasons why it might have failed. This enables users to monitor the progress of the website's login capability testing and take appropriate action if needed. We utilize Selenium in the provided code to automate form input on the https://demoqa.com/text-box page. The objective of this test is to provide some user information in a form, click the submit button, and then review the test results that appear after completing the form.

Table 3. Goods Transaction Testing Scenarios

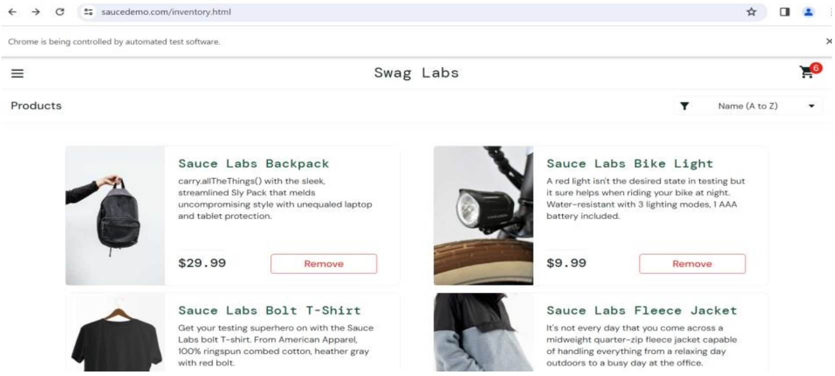
Test Plan Name	Number of tests	Testing
Click Add to Cart	2	Selenium ideas
Click CheckOut	2	Selenium ideas
Fill in customer data	1	Selenium ideas

In Table 3. In the Goods Transaction test scenario, pseudocode for the Selenium program that logs in to www.saucedemo.com/inventory.html by clicking on a selected item.

In the Pseudocede Selenium Python program for goods transactions as follows:

```
    Initialize WebDriver with Chrome
        Find one of the items to purchase using the CSS class "inventory_item"
        Retrieve item name using CSS class "inventory_item_name"
        Find the "Add to Cart" button using the CSS class "btn_inventory"
        Click the "Add to Cart" button
        Print "Added [item_name] to cart."
```

Output Display


saucedemo.com/inventory.html

Chrome is being controlled by automated test software.

Swag Labs

Products

Name (A to Z)

Sauce Labs Backpack
carry.allTheThings() with the sleek, streamlined Sly Pack that melds uncompromising style with unequaled laptop and tablet protection.
$29.99
Remove

Sauce Labs Bike Light
A red light isn't the desired state in testing but it sure helps when riding your bike at night. Water-resistant with 3 lighting modes, 1 AAA battery included.
$9.99
Remove

Sauce Labs Bolt T-Shirt
Get your testing superhero on with the Sauce Labs bolt T-shirt. From American Apparel, 100% ringspun combed cotton, heather gray with red bolt.

Sauce Labs Fleece Jacket
It's not every day that you come across a midweight quarter-zip fleece jacket capable of handling everything from a relaxing day outdoors to a busy day at the office.

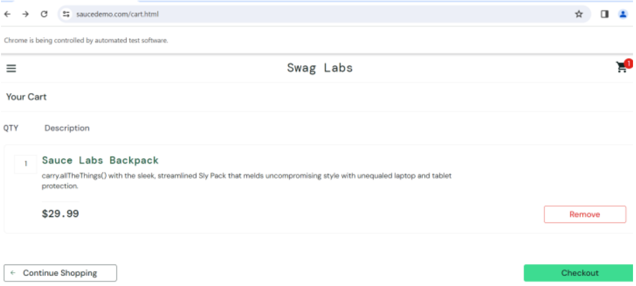
Figure 3. Goods Transaction Options

In figure 3. Display: Select one of the items from the inventory page and add it to the basket. Continue to the cart page by clicking the "Shopping Cart" button.

In Pseudocede Selenium python program to Checkout as follows:

```
Find the "Shopping Cart" button using the CSS class "shopping_cart_link"
    Click the "Shopping Cart" button
    Find the "Checkout" button using the ID
    Click the "Checkout" button
    Print the page URL after checkout
    Print "URL after checkout: [URL_checkout]"
Catch Exception
    Print "Process failed:", error message
Finally
    Close the browser when finished
    Close WebDriver
```

Output Display


saucedemo.com/cart.html

Chrome is being controlled by automated test software.

Swag Labs

Your Cart

QTY Description

1 Sauce Labs Backpack
carry.allTheThings() with the sleek, streamlined Sly Pack that melds uncompromising style with unequaled laptop and tablet protection.
$29.99
Remove

Continue Shopping

Checkout

Figure 4. Goods Checkout Process

In Figure 4, carry out the checkout process by clicking the "Checkout" button. Prints the page URL after checkout and ends the processIn the Pseudocede Selenium Python program to checkout the customer identity as follows:

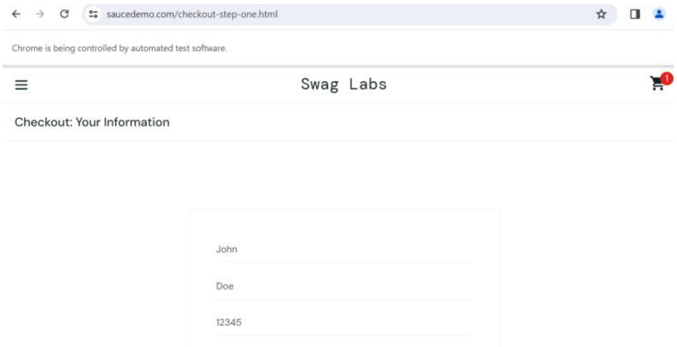
```
Find the "Checkout" button using the ID
        Click the "Checkout" button
        Wait until the URL changes to "https://www.saucedemo.com/checkout-step-one.html" (maximum
         10 seconds)
        Print "Checkout Step One."
        Find the "First Name" input using the ID
        Enter "John" into the "First Name" input
        Find the "Last Name" input using the ID
        Enter "Doe" into the "Last Name" input
        Find the "Postal Code" input using the ID
        Enter "12345" into the "Postal Code" input
        Find the "Continue" button using the ID
        Click the "Continue" button
```

*name of corresponding author

```
        Print the page URL after checkout step two
        Print "URL after checkout step two: [URL_checkout_step_two]"

    Catch Exception
        Print "Process failed:", error message
    Finally
        Close the browser when finished
        Close WebDriver
```

Finished



saucedemo.com/checkout-step-one.html

Chrome is being controlled by automated test software.

Swag Labs

Checkout: Your Information

John

Doe

12345

Figure 5. Filling in customer data

In Figure 5. Checkout display – your information, by filling in customer data after making a goods transaction. This program covers the steps from logging in to automatically filling in checkout information on the checkout-step-one.html page and continuing to the checkout step two page. If there is an error or the process is unsuccessful, the program will catch and print an error message.

Table 4. Goods Transaction Test Results

Test design	Number of Tests	Succeed	Fail
Click Add to Cart	2	Succeed	
Click CheckOut	1	Succeed	
Fill in customer data	1	Succeed	

In Table 4. The pseudocode above describes the program flow using Selenium to carry out the shopping and checkout process on the website www.saucedemo.com. This program covers the steps from logging in to automatically filling in checkout information on the checkout-step-one.html page and proceeding to the step two checkout page. If there is an error or the process is unsuccessful, the program will catch and print an error message.

DISCUSSIONS

In the context of the discussion, topics that can be discussed regarding the login process, goods transactions, and filling in customer data using Selenium for automation are as follows: Login Process: The importance of security in the login process and the use of automation technology such as Selenium to manage the login process safely. Effective login element identification strategies using Selenium, such as using IDs, class names, or CSS selectors. The benefits of using the wait method are to handle loading times and the presence of important elements on the login page. Item Transaction: Use Selenium to automatically select items from the inventory page and add them to the shopping cart. Discussion regarding speed and efficiency in the goods transaction process using automation with Selenium. Checkout Process: Challenges and solutions in performing the checkout process automatically using Selenium. Discussion regarding data validation on the checkout page and using the wait method to handle complex processes. Filling in Customer Data: The importance of filling in customer data accurately and automatically in the online shopping process. Strategy for identifying input elements and filling in customer data using Selenium. Additionally, discussions may also include the benefits of automating these processes in increasing efficiency, reducing errors, and providing a better shopping experience for users. A comprehensive discussion can provide a better understanding of Selenium implementation in the context of online shopping processes and customer data management.

CONCLUSION

In conclusion, the discussion regarding the login process, goods transactions, and filling in customer data using Selenium for automation on the website www.saucedemo.com presents several important points: Efficiency and

Consistency: Using Selenium allows the login process, goods transactions, and filling in customer data to be carried out automatically. automatically with a high level of efficiency and consistency. This helps increase productivity in managing online shopping transactions. Data Security: This automation process must be implemented with attention to data security, especially in the login process and filling in customer data. The use of appropriate element identification methods and secure management of sensitive data is a primary concern. Improved User Experience: By automating this process, users can experience a better and smoother shopping experience. Automatic data entry and efficient transaction processes provide a positive experience for users. Automatic Testing: Apart from that, Selenium can also be used for automatic testing to test important functions on the website, such as the login process, goods transactions, and filling in customer data. This helps ensure the quality and consistency of important functions on the website. Thus, the use of Selenium in automating the login process, goods transactions, and filling in customer data has significant benefits in improving efficiency, data security, user experience, as well as in testing and maintaining overall website quality.

REFERENCES

Assyakurrohim, D. (2022). Metode Penelitian Kualitatif: Studi Kasus. Jurnal Pendidikan Sains dan Komputer, *3*(01), 1–9. https://doi.org/10.47709/jpsk.v3i01.1951

Avrianto, R. P. (2022). Robotic Process Automation for Quality Control Assessment Using Selenium. *Jurnal Teknik Informatika (Jutif)*, *3*(5), 1301–1312. https://doi.org/10.20884/1.jutif.2022.3.5.341

Chitra, B. (2019). Strategi Digital Public Relations Pegiat Podcast dalam Membangun Brand Engagement (Studi Kasus dalam Podcast Duo Budjang). *Prologia*, *3*(2), 532. https://doi.org/10.24912/pr.v3i2.6413

Fahrizal, A. A. (2023). *Pengujian Aplikasi Web Absensi Siswa Menggunakan Selenium IDE*. *1*(4), 623–630.

Fauzan, R. (2023). Perbandingan Pengujian Manual dan Terotomasi pada Software Enterprise Resource Planning. *JAIIT (Journal of Advanced in Information and Industrial Technology)*, *5*(1), 23–30.

James W. (20 C.E.). Enterprise Information System. In *Andrew's Disease of the Skin Clinical Dermatology*.

Muhammad Ramadan Fikri. (2022). Web Scraping Situs Berita Menggunakan Bahasa Pemograman Python. *Journal of Students' Research in Computer Science*, *3*(1), 123–136. https://doi.org/10.31599/jsrcs.v3i1.1514

Mustika, A. (2024). Permodelan Sistem Informasi Penjualan Barang Menggunakan Metode Scrum. *Journal of Data Science and Information ...*, *2*(1), 14–25. https://ejournal.techcart-press.com/index.php/dimis/article/view/97%0Ahttps://ejournal.techcart-press.com/index.php/dimis/article/download/97/92

Rees, R. T. (2007). The role of HR in organizational development and innovation. In *Employment Relations Today* (Vol. 33, Issue 4). https://doi.org/10.1002/ert.20127

Rizquina, A. Z. (2023). Implementasi Web Scraping untuk Pengambilan Data Pada Website E-Commerce. *Jurnal Teknologi Dan Sistem Informasi Bisnis*, *5*(4), 377–383. https://doi.org/10.47233/jteksis.v5i4.913

Setiawan, Z. (2023). *Kewirausahaan Digital* (M. . Diana Purnama Sari (ed.)). : PT GLOBAL EKSEKUTIF TEKNOLOGI.

Silalahi, E. (2023). Analisis Penerapan ISO 9001 : 2015 Industri Makanan : Sebuah Narrative Literature Review. *Jisma*, *02*(03), 25–33.

Ulfah, A. (2023). Implementasi Web Scraping Pada Situs Jurnal Sinta Menggunakan Framework Selenium Webdriver Python. *JIKA (Jurnal Informatika)*, *7*(1), 29. https://doi.org/10.31000/jika.v7i1.7037

Wahyudin, Y. (2020). Analisis Metode Pengembangan Sistem Informasi Berbasis Website: A Literatur Review. *Jurnal Interkom: Jurnal Publikasi Ilmiah Bidang Teknologi Informasi Dan Komunikasi*, *15*(3), 26–40. https://doi.org/10.35969/interkom.v15i3.74

Wekke, I. S. (2019). *Ismail Suardi Wekke, dkk*. CV. Adi Karya Mandiri).

Wendy Winata. (2022). Pengujian Website EPOS PT XYZ Menggunakan Metode Black Box Testing. *Jurnal Informatika Atma Jogja*, *3*(2), 99–106. https://doi.org/10.24002/jiaj.v3i2.6780

Yondra, A. S. (2022). Implementasi Web Scraping untuk Mengumpulkan Informasi Produk dari Situs E-commerce dan Marketplace dengan Teknik Pemrosesan Paralel. *Coding : Jurnal Komputer Dan Aplikasi*, *10*(01), 93–102.

*name of corresponding author