# Enhancing Sportirena testing through custom Selenium Automation tools for PT Technova

Julian Naufal Azzam[1], Pradana Atamadiputra[2], Normalisa[3]

[1,2,3]Computer Science, International University Liaison Indonesia, Associate Tower IntermarkIndonesia, Jl. Lingkar Luar Timur, Indonesia, 15310

email: julian.azzam@stud.iuli.ac.id[1], pradana.atmadiputra@iuli.ac.id[2], normalisa@iuli.ac.id[3]

**ABSTRACT. This research paper, entitled "Improving the Testing of Sportirena through Personalized Selenium Automation Tools," highlights the urgent requirement for automated testing within the Sportirena platform. In a similar manner to enhancing fire resistance in composite sandwich panels, which aims to strengthen against potential dangers, our investigation concentrates on enhancing the reliability of Sportirena's application. We have developed customized automation tools, comparable to the addition of fire retardants, utilizing Selenium, with a specific focus on user authentication. The results demonstrate significant improvements in testing efficiency, coverage, and defect detection when compared to manual techniques. The insights obtained from existing literature support our findings and validate the chosen methodology. By drawing parallels between fire resistance in composite panels and automated testing in Sportirena, this study contributes to the overall objective of ensuring safety and resilience in intricate systems.**

**Keywords: testing, efficiency, coverage, and defect detection.**

# 1. Introduction

PT Technova Optima Prima is an IT solutions company focused on areas like smart city solutions, analytics, app and web development and cloud consulting for clients across various industries. However, the company still use manual testing method for their project. These challenges span diverse areas, including:

- Develop a Saas Enterprise website
- Application development
- Web development

# 2. Purpose of this System

This project is to develop an automated testing bot using selenium and python to optimize the login, account creation. procedures. By using automating these tasks, the system aims to enhance operational efficiency, minimizing human error and expedite processes, offering significant time savings compared to manual testing .

# 3. System Research Path

This project focuses on evaluating automated testing versus manual testing methods in the context of developing an automated bot using Selenium with Python for login and create account process.

- Test Scenario Design: Develop a test scenario covering the process of login, create account, and booking, ensuring a comprehensive range of functionalities.
- Visualization of findings: Look through the source code and find the id class
- Execution and Analysis: Implement automated testing and manual testing to evaluate the performance metrics such as speed, accuracy and effectiveness.

# 3. Development Approach

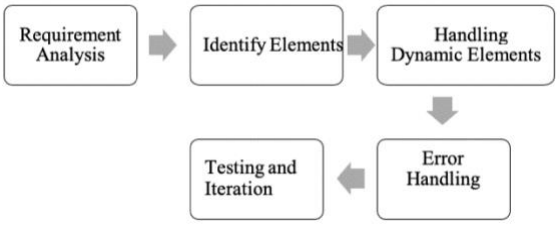


Figure 1. system research path

In this project, it will go through a few important steps. First, analyze the requirements to figure out what functions are needed. Then, use Selenium to identify elements for tasks like logging in, creating accounts, and making bookings. Additionally dealing with handling dynamic elements to ensure everything is adaptable. Moreover, implement error handling to make the system robust. Finally, everything thoroughly and make any necessary adjustments to improve functionality and reliability. The goal of this development approach is to create an efficient and trustworthy automated bot that can streamline processes.

# 4. Automation bot Development Login process

## 4.1 Login process


Figure 2. Homepage

This image shows the homepage of the Sportirena website. The bot search for the login button id to open the login page.

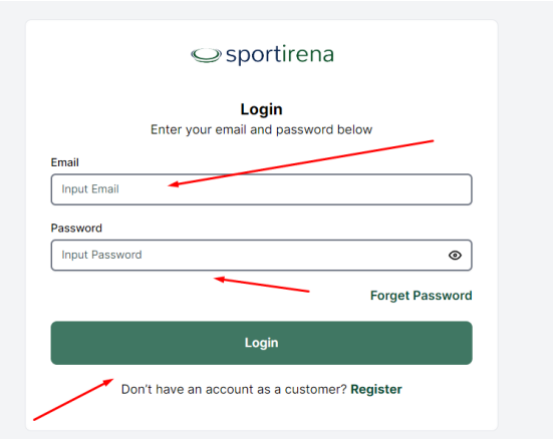


Figure 3. Login Page

The images display the login page of the Sportirena website. The bot is prompted to enter their email and password to login, after the criteria is field the bot finds the login button and click it.

```
driver.get("https://sportirena.com/auth/login")

# Find the email and password input fields and the submit button
email_input = driver.find_element(By.ID, 'email')
password_input = driver.find_element(By.ID, 'password')
submit_button = driver.find_element(By.XPATH, '//button[@type="submit"]')

# Input the login credentials
email_input.send_keys("test@gmail.com")
password_input.send_keys("Test123")

# Submit the form
submit_button = driver.find_element(By.XPATH, '//button[@type="submit"]')
submit_button.click()
```

Figure 4. Code Process login

This images show a snippet of a code process used for automating the login process. The script navigates to the login page, find the email and password to input the fields and submit the form.

# Register process

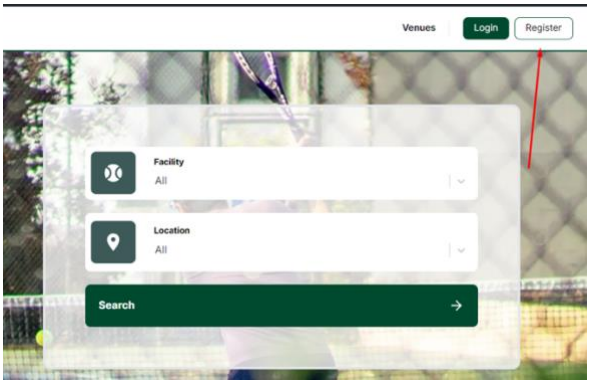


Figure 5. Bot Search

This images indicated that the bot search for the register button id to open the register page

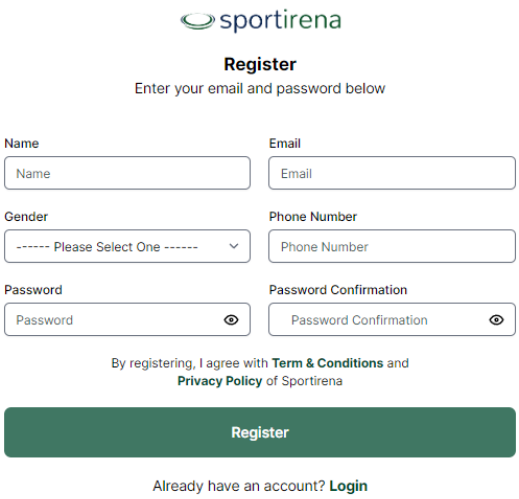


Figure 6. Register Page

The images present the registration form on the Sportirena website. The bot needs to fill in the the name, email, gender, phone number, password, and password confirmation fields, after it fills the criteria, it click the register button

```
# Find the fields
name = driver.find_element(By.ID, "name")
email = driver.find_element(By.ID, "email")
gender = driver.find_element(By.ID, "gender")
phone = driver.find_element(By.ID, "phoneNumber")
password = driver.find_element(By.ID, "password")
confirm_password = driver.find_element(By.ID, "confirmPassword")

# Enter fake data
name.send_keys("kalya")
email.send_keys("kalya123@yahoo.co.id")
gender.send_keys("Male")
phone.send_keys("08123456789")
password.send_keys("Password123")
confirm_password.send_keys("Password123")

# Submit form
submit = WebDriverWait(driver, 10).until(EC.element_to_be_clickable((By.ID, "button[type='submit']")))
submit.click()

submit = WebDriverWait(driver, 10).until(EC.element_to_be_clickable((By.XPATH, "//button[@type="submit"]")))
submit.click()

# Wait for success message
try:
    success_msg = WebDriverWait(driver, 10).until(EC.visibility_of_element_located((By.XPATH, "//div[contains(text(), 'Registration Success')]
    print(success_msg.text)
    print("Registration Passed √")
except:
    print("Registration Failed x")


driver.quit()
```

Figure 7. Script Automating Registration Process

This image shows a snippet of the Selenium script used for automating the registration process. The script in the registration form with predefined fake data, submit the form and waits for a success message to confirm the registration.

# 5. Conclusion

From the implementation and testing of the registration and login automation scripts for the Sportirena website, several conclusions can be drawn :

1. The automation of registration and login process using selenium has effectively streamlined the user onboarding experience for Sportirena. By automating the completion of registration forms and login procedures, these scripts have significantly reduced the time and effort required for users to access the platform.

2. The registration and login automation scripts have contributed to improving operational efficiency by eliminating manual intervention in user registration and login process. By designing user actions and interactions with the web, these scripts have minimized the time and resources required for user registration and login, ultimately enhancing the overall efficiency of the platform‘s onboarding process.

3. Additionally, the automation of the registration and login process facilitated better user satisfaction website without encountering manual error or delays. This streamlined process promotes a positive user experience. Overall, the implementation of these automation scripts has contributed to enhancing user satisfaction and usability of the Sportirena website platform.

**REFERENCE**

Andreza MFV de Castro et al. "Extension of Selenium RC tool to perform automated testing with databases in web applications," in Proc. International Workshop on Automation of Software Test, 2013.

Satish Gojare, Rahul Joshi, and Dhanashree Gaigaware, "Analysis and design of selenium webdriver automation testing framework," Procedia Computer Science 50, pp. 341-346, 2005.

Nicey Paul and Robin Tommy, "An Approach of Automated Testing on Web Based Platform Using Machine Learning and Selenium," in Proc. International Conference on Inventive Research in Computing Applications, 2018.