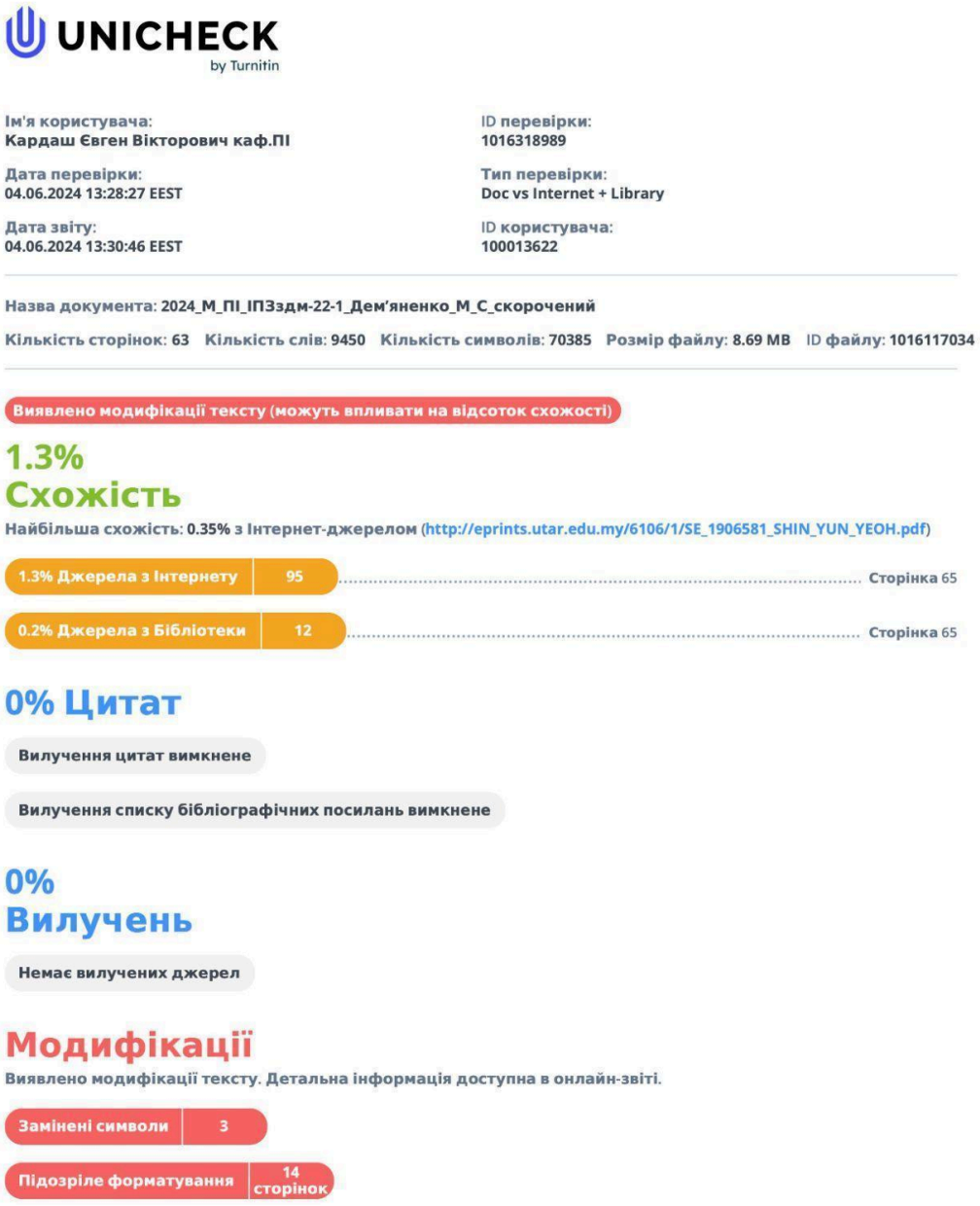
ДОДАТОК А

Перелік джерел посилання за науковими напрямами керівника та науковців кафедри програмної інженерії

19. Груздо І.В., Мічурін І.Є. Особливості використання хмарних технологій для розміщення програмного забезпечення. Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління : тези доп. 12-ї міжнар. наук.-техн. конф., 27-28 квітня 2022 р., Баку–Харків–Жиліна : [у 2 т.]. Т. 2 : секція 5 / Військ. акад. збройних сил Азербайджанської Республіці

ДОДАТОК Б

Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ



UNICHECK
by Turnitin

Ім'я користувача:
Кардаш Євген Вікторович каф.ПІ

ID перевірки:
1016318989

Дата перевірки:
04.06.2024 13:28:27 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
04.06.2024 13:30:46 EEST

ID користувача:
100013622

Назва документа: 2024_М_ПІ_ІПЗздм-22-1_Дем'яненко_М_С_скорочений

Кількість сторінок: 63 Кількість слів: 9450 Кількість символів: 70385 Розмір файлу: 8.69 MB ID файлу: 1016117034

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

1.3%
Схожість

Найбільша схожість: 0.35% з Інтернет-джерелом (http://eprints.utar.edu.my/6106/1/SE_1906581_SHIN_YUN_YEOH.pdf)

1.3% Джерела з Інтернету 95 ..... Сторінка 65

0.2% Джерела з Бібліотеки 12 ..... Сторінка 65

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

Модифікації

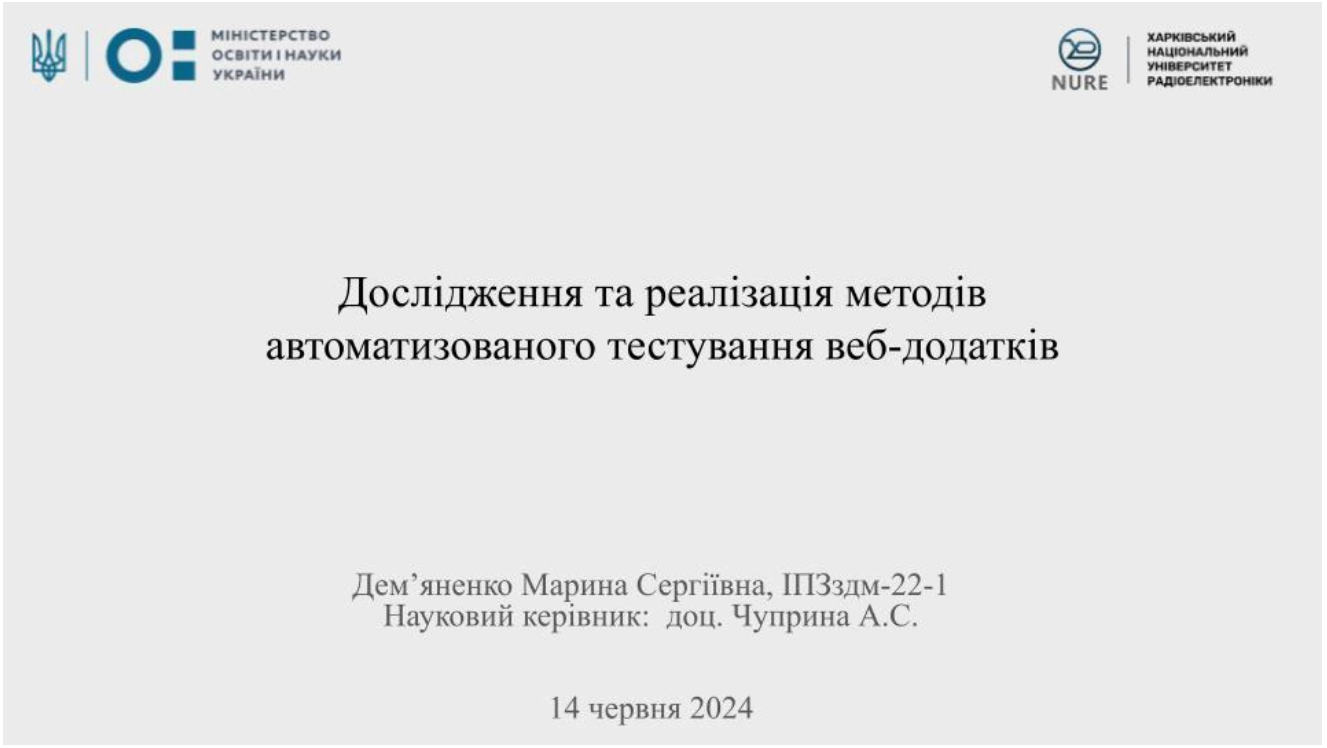
Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

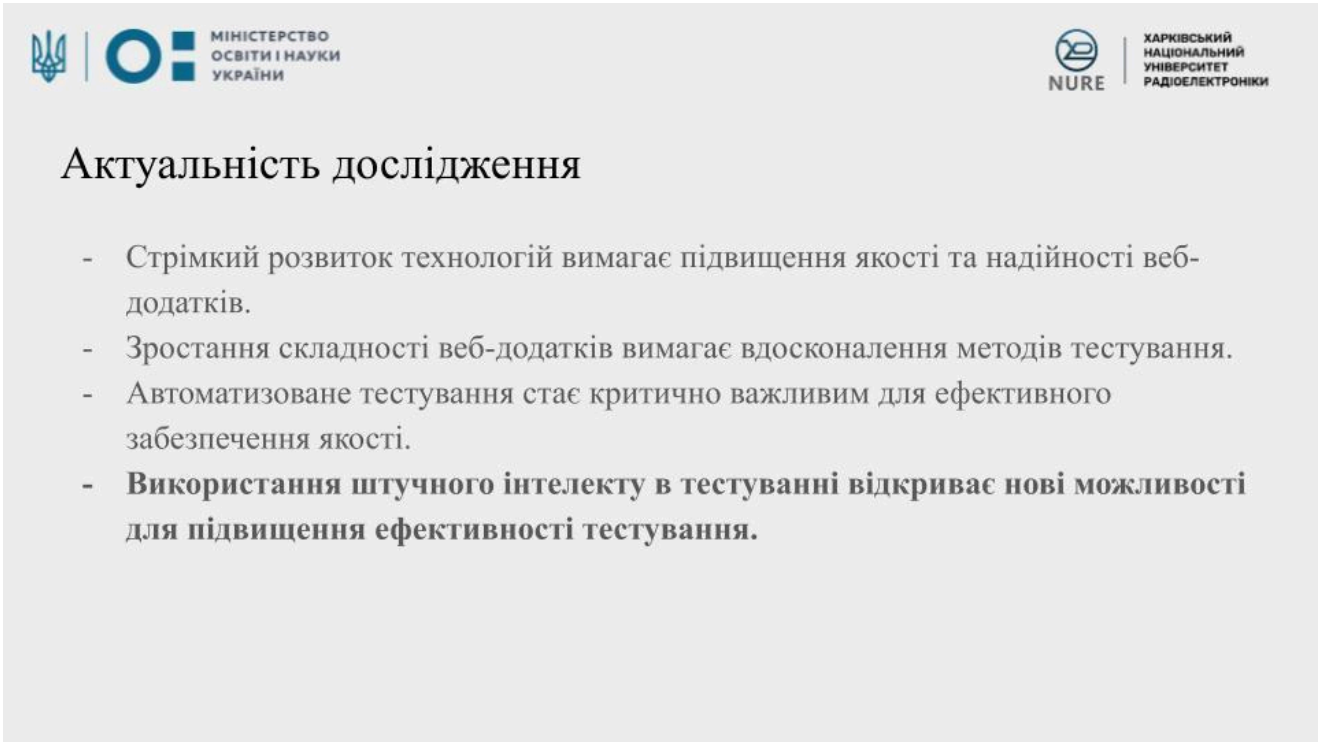
Замінені символи 3

Підозріле форматування 14 сторінок

ДОДАТОК В

Слайди презентації





МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

NURE ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Аналіз предметної області

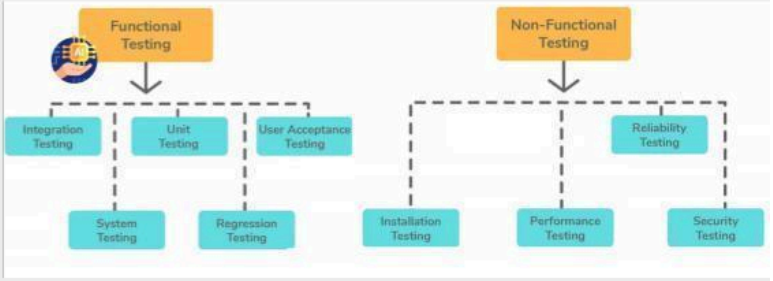
Мета дослідження: Оцінка ефективності застосування великих мовних моделей у процесах QA тестування веб-додатків.

Проблеми сучасних методів:

- Складність створення та підтримки актуальності тестів
- Високі витрати часу та ресурсів на процес тестування
- Обмежена здатність виявлення складних дефектів

Використання штучного інтелекту:

- Підвищення точності у процесі тестування
- Скорочення часу на тестування
- Виявлення прихованих дефектів та неочевидних рішень



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

NURE ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Постановка задачі

- провести аналіз великих мовних моделей;
- розробити план експериментального дослідження (обрати критерії для визначення ефективності);
- спроектувати програму для втілення експериментів та запити для мовної моделі;
- створити допоміжні засоби для втілення експериментів;
- провести експериментальне дослідження;
- оцінити ефективність застосування великих мовних моделей у процесах QA тестування веб-додатків з точки зору:
  - релевантності генерації тестових сценаріїв;
  - ефективності виявлення дефектів;
  - повноти охоплення тестами і загального покращення часу циклу тестування.
- провести аналіз результатів.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

NURE ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Математичний опис задачі дослідження

Модель оцінки ефективності:

- Метод зваженої суми (**Weighted Sum Model - WSM**)
  - wi - вагові коефіцієнти, що відображають відносний внесок окремих критеріїв,
  - Ci - нормалізоване значення і-го критерію.

$$E = \sum_{i=1}^{n} (w_i \times C_i)$$

Критерій	Пояснення	Одиниці виміру	Шкала
Час на вирішення завдання (TRT)	Час, що минув від моменту запуску тестового завдання до його успішного завершення.	Хвилини	0-5: Відмінно 5-10: Добре 10-20: Задовільно >20: Незадовільно
Час обробки результатів (TPR)	Час, необхідний для обробки результатів, що генеруються великою мовною моделлю	Хвилини	0-5: Відмінно 5-10: Добре 10-20: Задовільно >20: Незадовільно
Релевантність отриманих результатів (RGR)	Відображає точність і корисність отриманих даних	Відсотки	90-100%: Відмінно 80-89%: Добре 70-79%: Задовільно <70%: Незадовільно
Кількість уточнень (NC)	Показує, скільки разів необхідно уточнювати результати або вносити корективи	Кількість уточнень	0-1: Відмінно 2-3: Добре 4-5: Задовільно >5: Незадовільно

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

NURE ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Проектування засобів для експерименту

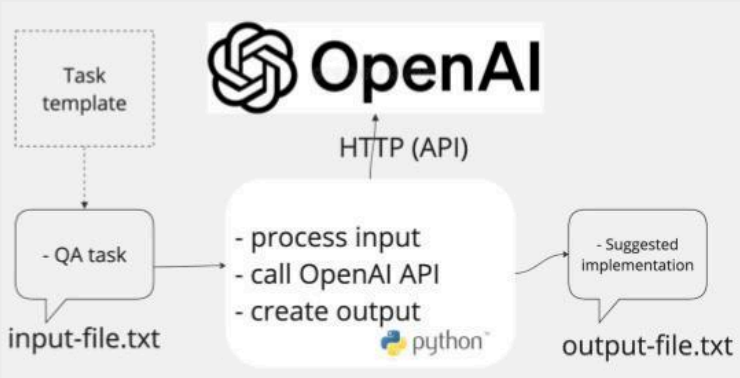
- Python програма-асистент:
  - Бібліотека OpenAI для взаємодії з API мовної моделі GPT-4
  - Шаблон промпту для запиту до OpenAI API
- Код веб-додатку з базовими функціями логіну
  - Фронтенд React (JavaScript)
  - бекенд Node.js та Express
- Проєкт Cypress:
  - Каталог для запуску авто-тестів у форматі cy.js
  - Базові налаштування проекту для роботи з веб-додатком

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Проектування засобів для експерименту

```
Input data:

Test case:
#### Test Case 1: Access Login Page from Home Page
**Objective**: Verify that the login page is accessible
from the home page.
**Steps**:
1. Open the home page of the website.
2. Locate and click on the login link or button.
**Expected Result**:
- The login page should be displayed.

Task:
Create an auto test for the functionality described in
the test case.
Generate some cy.js file that will be used in
cypress/e2e/ to run the tests

Additional info:
Our website accessible at:
http://ec2-54-77-212-54.eu-west-1.compute.amazonaws.com
Home page accessible at
http://ec2-54-77-212-54.eu-west-1.compute.amazonaws.com/h
ome
We already have the cypress project setup.
```

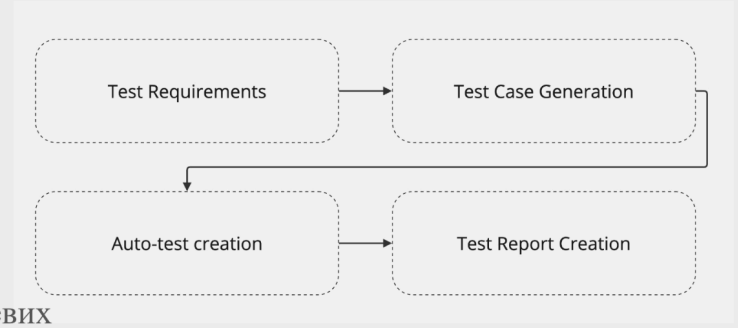


МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Планування експериментального дослідження

- Формування Промптів за Темплейтом
- Запуск програми-помічника на Python
- Аналіз Отриманих Результатів
- Уточнення промптів
- Перегляд та підтвердження кінцевих результатів
- Оцінка за критеріями



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Експериментальні результати - тестування вимог

- user story
- чіткі та стислі очікування щодо функціональних можливостей, що розробляються
- застосування шаблону запиту
- перевірка та оцінка отриманого результату

```
) py test.py --input input-data/requirements-testing.txt
Request completed in 32.37 seconds.
Output written to output-data/output_2024-04-11_14-26-51.txt
### Original User Story
**As a user, I want to be able to log in to our website using an email and password.**

### Improved Requirements for QA Testing

#### Functional Requirements
1. **User Authentication Interface**
   - The website should have a dedicated login page accessible from the home page.
   - The login page must include fields for the user to enter their email address and password.
```

Input data:

User story: As a user i want to be login to be able to login at our website using email and password.

Task:
- Perform Requirements Testing based on the user story for the QA testing.

Additional info:
- the result should contain improved requirements
- the requirements should be only related to the functionality described in the story
- no testing scenarios should be provided as the result of requirements testing improvement

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Експериментальні результати - тестування вимог

****Reviewed Requirements:****

1. **Clarity and Completeness**
   - As a registered user (including administrators and regular users), I want to be able to log in to our website using my email and password so that I can access user-specific functionalities.

2. **Security Requirements**
   - Ensure that the website uses HTTPS to securely transmit the email and password from the client to the server.
   - **Passwords must be stored securely using modern hashing algorithms (e.g., bcrypt) in the database.**

3. **User Feedback**
   - Upon successful login, redirect the user to their dashboard and display a welcome message.
   - On unsuccessful login, display an error message that does not specify whether the email or password was incorrect for security reasons.

4. **Accessibility**
   - The user story should consider accessibility features for users with disabilities.

5. **Usability**
   - Provide an option for users to remain logged in on their device to avoid re-entering credentials on each visit.

6. **Validation and Error Handling**
   - The email field must accept only valid email formats and should be validated both on the client-side and server-side.
   - Password field should enforce a minimum complexity (e.g., minimum 8 characters, including numbers and symbols).
   - If the login attempt fails, the system should log the attempt and provide a generic error message to the user.

7. **Performance**
   - The login process should complete within 2 seconds under normal conditions for 95% of requests.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Експериментальні результати - тестування вимог

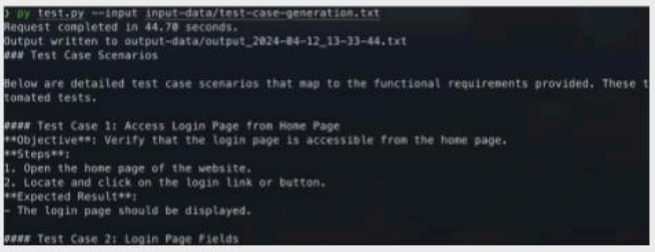
Оцінка	Значення під час експерименту	Критерій
4	32.37 секунди	Час вирішення завдання (TRT)
3	5 хвилин	Час обробки результатів (TPR)
0,66	18/27	Релевантність отриманих результатів (RGR)
3	0	Кількість уточнень (NC)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

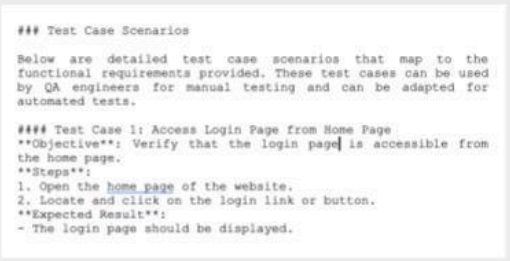
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Експериментальні результати - генерація тестових сценаріїв

- **специфікації бізнес-вимог**
- **специфікації вимог до програмного забезпечення**
- **специфікації функціональних вимог**



```
> py test.py --input input-data/test-case-generation.txt
Request completed in 44.70 seconds.
Output written to output-data/output_2024-04-12_13-33-44.txt
### Test Case Scenarios

Below are detailed test case scenarios that map to the functional requirements provided. These t
tomated tests.

#### Test Case 1: Access Login Page from Home Page
**Objective**: Verify that the login page is accessible from the home page.
**Steps**:
1. Open the home page of the website.
2. Locate and click on the login link or button.
**Expected Result**:
- The login page should be displayed.

#### Test Case 2: Login Page Fields
```



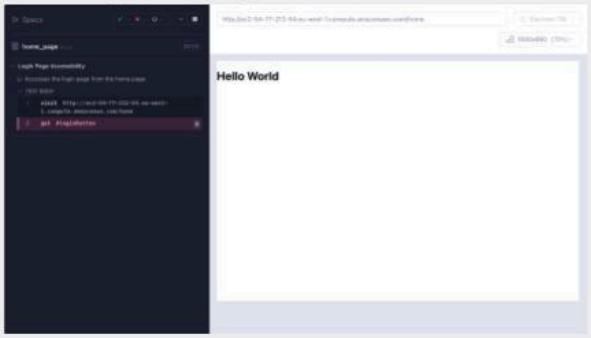
```
### Test Case Scenarios

Below are detailed test case scenarios that map to the
functional requirements provided. These test cases can be used
by QA engineers for manual testing and can be adapted for
automated tests.

#### Test Case 1: Access Login Page from Home Page
**Objective**: Verify that the login page is accessible from
the home page.
**Steps**:
1. Open the home page of the website.
2. Locate and click on the login link or button.
**Expected Result**:
- The login page should be displayed.
```

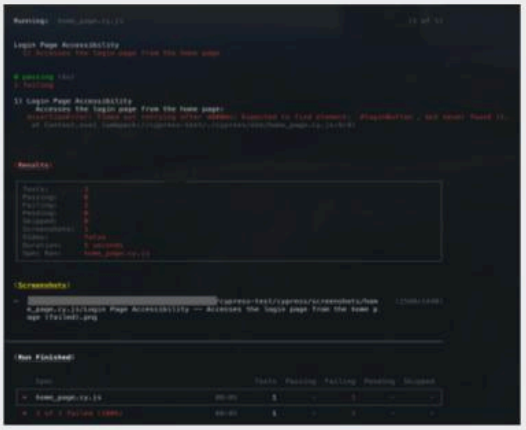
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Експериментальні результати - генерація авто тестів

- **сформовані тестові сценарії**
- **Cypress фреймворк проєкт**
- **запуск згенерованих авто-тестів**



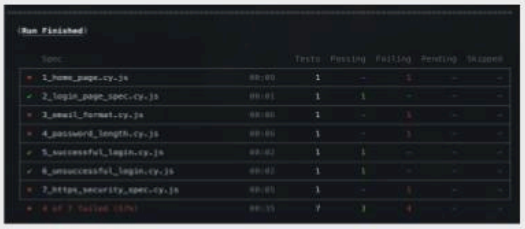


МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Експериментальні результати - створення звіту з тестування

- **дані вхідні - попередні результати**
- **результат запуску усіх авто-тестів**
- **ясність, точність та вичерпність звіту**



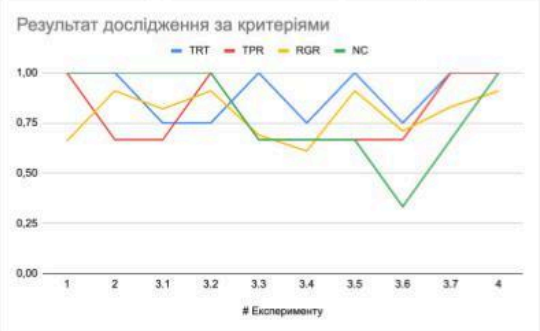

```
**QA Test Report: Website Login Functionality**

**Date**: [Insert Date]

**Test Environment**: [Specify Environment, e.g., Staging]

**Overview**: A series of seven automated test cases were
executed to validate the login functionality of the website.
The tests ranged from page accessibility to input validation
and security protocols.

**Test Results Summary**:
- Total Tests: 7
- Passed: 3
- Failed: 4

**Detailed Test Outcomes**:

1. **Access Login Page from Home Page**
   - **Result**: Failed
    - **Issue**: The login page is not accessible from the home
page. The link or button intended to navigate to the login page
may be missing or not functioning as expected.

2. **Login Page Fields**
   - **Result**: Passed
```

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Аналіз результатів дослідження

- **нормовані результати**
- **високі оцінки за кожним критерієм**
  - **(здебільшого вище %66)**

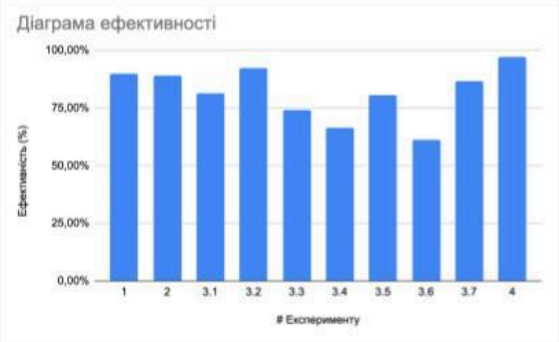


Критерій/Оцінка				
# Експерименту	TRT	TPR	RGR	NC
1	1,00	1,00	0,66	1,00
2	1,00	0,67	0,91	1,00
3.1	0,75	0,67	0,82	1,00
3.2	0,75	1,00	0,91	1,00
3.3	1,00	0,67	0,69	0,67
3.4	0,75	0,67	0,61	0,67
3.5	1,00	0,67	0,91	0,67
3.6	0,75	0,67	0,71	0,33
3.7	1,00	1,00	0,83	0,67
4	1,00	1,00	0,91	1,00

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Аналіз результатів дослідження



Критерій/Оцінка					
# Експерименту	TRT	TPR	RGR	NC	Ефективність (%)
1	1,00	1,00	0,66	1,00	89,80%
2	1,00	0,67	0,91	1,00	88,97%
3.1	0,75	0,67	0,82	1,00	81,27%
3.2	0,75	1,00	0,91	1,00	92,30%
3.3	1,00	0,67	0,69	0,67	74,03%
3.4	0,75	0,67	0,61	0,67	66,63%
3.5	1,00	0,67	0,91	0,67	80,63%
3.6	0,75	0,67	0,71	0,33	61,30%
3.7	1,00	1,00	0,83	0,67	86,57%
4	1,00	1,00	0,91	1,00	97,30%
Вагові коефіцієнти	0,2	0,25	0,3	0,25	

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Висновки

- **Підвищення ефективності та якості:**
  - **Автоматизація складних і неявних аспектів генерації та аналізу тестів**
  - **Зменшення часових та трудових витрат**
- **Висока релевантність:**
  - **Генерація тестових сценаріїв**
  - **Виявлення дефектів і повнота охоплення тестами**
- **Скорочення часу на тестування:**
  - **Швидша генерація тестів**
  - **Аналіз результатів**
- **Обмеження:**
  - **Необхідність оптимізації вхідних даних**
  - **Потреба у додаткових ресурсах та тренінгу команд**
- **Рекомендації:**
  - **Розширити використання мовних моделей у тестуванні**
  - **Провести додаткові дослідження для оптимізації**
  - **Розробити тренінгові програми для QA тестувальників**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Апробація та впровадження

Тези на тему “Сучасні підходи до тестування та якості програмного забезпечення” пройшли апробацію на 28-му Міжнародному молодіжному форумі «РАДІОЕЛЕКТРОНІКА І МОЛОДЬ В XXI СТОЛІТТІ»

На основі дослідження та експериментів проведених в рамках дипломної роботи, розпочато впровадження запропонованої технології на поточний проект.

ДОДАТОК Г

Апробація результатів роботи

УДК 004.415.3 DOI: https://doi.org/10.30837/IYF.IIS.2024.487

СУЧАСНІ ПІДХОДИ ДО ТЕСТУВАННЯ ТА ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Дем'яненко М. С.
Науковий керівник – к.т.н., доц. Чуприна А. С.
Харківський національний університет радіоелектроніки, каф. МЕЕПП
м. Харків, Україна
e-mail: maryna.demianenko.cpe@nure.ua

In the contemporary tech environment, the critical role of software testing cannot be overstated. Its core purpose is to evaluate and enhance software quality, ensuring functionality, security, and ultimately user satisfaction, which in turn builds customer trust. The emergence of Artificial Intelligence (AI) in software development heralds a new era, significantly boosting the efficiency and effectiveness of these processes. The integration of AI innovates the various phases of software testing, markedly improving the testing methodology and leading to the creation of superior software products.

У сучасному технологічному середовищі важливість тестування програмного забезпечення та забезпечення якості неможливо переоцінити. Тестування програмного забезпечення відіграє ключову роль у розробці додатків. Цей процес гарантує, що програмні системи функціонують належним чином, є безпечними, задовольняють потреби зацікавлених сторін і, зрештою, приносять користь кінцевим користувачам. Виявляючи дефекти на ранніх стадіях процесу розробки, тестування програмного забезпечення не тільки покращує якість продукту, але й сприяє довірі клієнтів і задоволенню, надаючи продукт, який був ретельно оцінений. Крім того, тестування програмного забезпечення відіграє важливу роль у виявленні вразливостей системи безпеки, що має першочергове значення, враховуючи зростаючу поширеність кіберзагроз [1].

Проблеми, з якими стикається сучасне тестування програмного забезпечення, і відповідна потреба в інноваціях є багатогранними. Технологічний ландшафт, що постійно розвивається, вимагає постійної адаптації та вдосконалення методологій тестування, щоб йти в ногу зі складнощами сучасних програмних додатків. Середовища гнучкого тестування та DevOps запровадили більш спільну та швидшу стратегію тестування, що підкреслює потребу в інноваціях у методах тестування для досягнення ефективності та результативності.

Роль штучного інтелекту (Artificial intelligence) у розробці програмного забезпечення стає дедалі помітнішою, пропонуючи нову парадигму для підвищення ефективності та результативності процесу розробки. Можливості штучного інтелекту поширюються на створення коду, підтримку розробників і аналіз складних баз коду для виявлення критичних аспектів, які потребують уваги.

У свою чергу, це явище не лише спрощує процес розробки, але й відкриває нові можливості для автоматизації та оптимізації різних етапів тестування програмного забезпечення. Використовуючи штучний інтелект, команди тестувальників можуть автоматизувати повторювані завдання, покращити відстеження помилок і дефектів та запровадити безперервне тестування протягом життєвого циклу розробки. Тому інтеграція штучного інтелекту не тільки розширює можливості команд розробників, але також може значно сприяти покращенню процесу тестування задля створення програмних продуктів вищої якості [2].

Прикладом впровадження інновації може стати автоматична модульна система у вигляді веб-додатку, що використовується командами тестувальників на початкових або ранніх стадіях існування проекту, коли процес тестування все ще має ознаки невизначеності.

Для етапу аналізу вимог та планування тестування буде використовуватись модуль збору інформації про проект та середовище. Дані про проект створюються за допомогою збору інформації від користувача, у вигляді "налаштування проекту", фактично – опитування, яке перетворюється у набір критеріїв, що будуть описувати майбутній інструмент або фреймворк для автоматизованого тестування, таких як: необхідні браузери, мови програмування для написання автоматичних тестів, необхідні емулятори та/або можливість використання вбудованих пристроїв для тестування, наявність вичерпної документації, відкритість коду, популярність. Зібрані дані оброблятимуться методом, що являтиме собою вирішення задачі оптимізації за допомогою моделі лінійної адитивної згортки з ваговими коефіцієнтами. Для цього будуть використані заздалегідь сформовані шкали для оцінки кожного критерію:

$$Z^* = max \sum_{j=1}^{n} a_j \beta_j a_{ij}, \quad (1)$$

де β_j – вагові коефіцієнти, що відображають відносний внесок окремих критеріїв до загального критерію, α_j – нормуючі множники, α_{ij} – значення критерію

Для покращення точності оцінки проекту у наступних ітераціях доного застосунку, дані про вибір кожного користувача зберігаються до внутрішньої бази даних та будуть використані для навчання регресійної моделі.

Для виконання всебічного аналізу файлів проекту наданих у формі архіву за допомогою OpenAI API створюється zip-файл для завантаження користувачем, який містить запропоновану структуру для автоматизованого тестування проекту.

Ідентифікація мови та виявлення фреймворків: використовуючи методи обробки природної мови (NLP), програма визначає основну мову

програмування та фреймворки, які використовуються в проекті, а також результат враховує критерії оптимізаційної задачі [4]. Цей крок гарантує, що подальші рекомендації адаптовані до конкретного технологічного стеку, який використовується в проекті.

Ідентифікація критичних функціональних можливостей: використовуючи механізми синтаксичного аналізу, що надає OpenAI API, програма визначає критичні функціональні можливості в рамках проекту. Для знайдених функціональних можливостей, за допомогою функції генерації коду, створюються тести, з якими команда може почати роботу [5]. Таким чином запроваджується забезпечення охоплення тестами.

Структура проекту автоматизації тестування: програма пропонує оптимальну структуру проекту для автоматизованого тестування, що сформована та упакована до zip-архіву. Також, для полегшення інтеграції генерується код для запуску створених тестів у CI/CD середовищі.

Запропонована модель з інтеграцією ШІ значно покращує ефективність впровадження автоматизації в проекти, дозволяє прийняти обгрунтоване рішення щодо вибору методів автоматизації, оптимальних для конкретного проекту, прискорюючи процес розробки та підвищуючи якість кінцевих програмних рішень через автоматизоване тестування.

Список використаних джерел:

1. Why Do We Need Software Quality Assurance and Testing?. Softvelopers. URL: https://www.softvelopers.com/blog/importance-software-testing-quality-assurance (дата звернення: 21.03.2024).

2. Testim. What Is the Software Testing Life Cycle? A Complete Guide. AI-driven E2E automation with code-like flexibility for your most resilient tests. URL: https://www.testim.io/blog/software-testing-life-cycle/ (дата звернення: 21.03.2024).

3. Радіоелектроніка та молодь у XXI столітті. Т. 6 : Конференція "Інформаційні інтелектуальні системи" : матеріали 23-го Міжнар. молодіж. форуму, 16–18 квіт. 2019 р. / М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. Харків, 2019. 306 с.

4. AI-Assisted Test Automation: Revolutionizing Software Testing Like Never Before. Opkey: #1 Test Automation Platform for Enterprise Continuous Testing. URL: https://www.opkey.com/blog/ai-assisted-test-automation-revolutionizing-software-testing-like-never-before (дата звернення: 21.03.2024).

5. Methods of Semantic Structured Search / Pohuliaiev, Y., Smelyakov, K., Chupryna, A., Ruban, I. 2022 IEEE 9th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), м. Kharkiv, Ukraine, 10–12 жовт. 2022 р. 2022. URL: https://doi.org/10.1109/picst57299.2022.10238538 (дата звернення: 22.03.2024).

ДОДАТОК Д

Код веб-додатку для експерименту

Реалізація бекенду викладена у наступному прикладі коду:

```
const express = require('express');
const bodyParser = require('body-parser');
const cors = require('cors');

const app = express();
const port = process.env.PORT || 5050;

app.use(cors());
app.use(bodyParser.json());

app.post('/login', (req, res) => {
  const { email, password } = req.body;

  // Mock user validation
  if (email === 'user@example.com' && password === 'example') {
    res.json({ message: 'Login successful' });
  } else {
    res.status(401).json({ message: 'Invalid credentials' });
  }
});

app.listen(port, () => {
  console.log(`Server is running on port ${port}`);
});
```

Реалізація фронтенду викладена у наступному прикладі коду:

```
import React, { useState } from 'react';
import axios from 'axios';
import { BrowserRouter as Router, Route, Routes, Navigate } from
'react-router-dom';

function Login({ onLoginSuccess }) {
  const [email, setEmail] = useState('');
  const [password, setPassword] = useState('');

  const handleSubmit = async (event) => {
    event.preventDefault();
    try {
          await axios.post('http://localhost:5050/login', { email,
password });
      onLoginSuccess();
    } catch (error) {
      alert('Login failed');
```

```
    }
  };

  return (
        <div style={{ display: 'flex', justifyContent: 'center',
alignItems: 'center', height: '100vh' }}>
          <form onSubmit={handleSubmit} style={{ display: 'flex',
flexDirection: 'column' }}>
                <input type="email" value={email} onChange={e =>
setEmail(e.target.value)} placeholder="Email" required />
            <input type="password" value={password} onChange={e =>
setPassword(e.target.value)}     placeholder="Password"     required
style={{ marginTop: '10px' }} />
                <button type="submit" style={{ marginTop: '10px'
}}>Login</button>
      </form>
    </div>
  );
}

function Home() {
  return <h1>Hello World</h1>;
}

function App() {
  const [isLoggedIn, setIsLoggedIn] = useState(false);

  const handleLoginSuccess = () => {
    setIsLoggedIn(true);
  };

  return (
    <Router>
      <Routes>
          <Route path="/" element={isLoggedIn ? <Navigate to="/home"
/> : <Login onLoginSuccess={handleLoginSuccess} />} />
        <Route path="/home" element={<Home />} />
      </Routes>
    </Router>
  );
}

export default App;
```

ДОДАТОК Е

Код Python програми для експерименту

```
import argparse
from datetime import datetime
from openai import OpenAI
import os
import time

def parse_arguments():
    """Parse command-line arguments."""
    parser = argparse.ArgumentParser(
          description="Run OpenAI requests with input from a file,
with extended parameters."
    )
    parser.add_argument(
         "--input", type=str, required=True, help="Path to the input
file."
    )
    parser.add_argument(
        "--model",
        type=str,
        default="gpt-4-turbo",
        help="Model to use for the request.",
    )
    parser.add_argument(
           "--temperature", type=float, default=0.7, help="Sampling
temperature."
    )
    parser.add_argument(
        "--max_tokens",
        type=int,
        default=4096,
        help="Maximum number of tokens to generate.",
    )
    parser.add_argument(
```

```
          "--top_p", type=float, default=1.0, help="Nucleus sampling
parameter."
    )
    parser.add_argument(
        "--frequency_penalty",
        type=float,
        default=0.0,
        help="Frequency penalty for repeated tokens.",
    )
    parser.add_argument(
        "--presence_penalty",
        type=float,
        default=0.0,
        help="Presence penalty for repeated tokens.",
    )
    parser.add_argument(
         "--stop", type=str, nargs="+", help="Stop sequence for the
completion."
    )
    return parser.parse_args()

def read_file(filepath):
    """Read and return the content of a file."""
    with open(filepath, "r") as file:
        return file.read()

def write_file(directory, content):
     """Write content to a file with a timestamp in its name in the
specified directory."""
                                                filename            =
f"output_{datetime.now().strftime('%Y-%m-%d_%H-%M-%S')}.txt"
    path = os.path.join(directory, filename)
    with open(path, "w") as file:
        file.write(content)
    print(f"Output written to {path}")



def run_request(input_message, args):
    """Run an OpenAI request and return the result."""
```

```
     # Initialize the OpenAI client (replace '<redacted>' with your
actual API key)
    client = OpenAI(api_key=os.environ['OPENAI_API_KEY'])

    # Create a completion request with additional parameters
    completion = client.chat.completions.create(
        model=args.model,
        messages=[{"role": "user", "content": input_message}],
        temperature=args.temperature,
        max_tokens=args.max_tokens,
        top_p=args.top_p,
        frequency_penalty=args.frequency_penalty,
        presence_penalty=args.presence_penalty,
        stop=args.stop
    )
    # Ensure the response is a string
    return completion.choices[0].message.content

if __name__ == "__main__":
    args = parse_arguments()
    # Read the input message
    input_message = read_file(args.input)
    # Capture the start time of the request
    start_time = time.time()
    # Run the OpenAI request
    response = run_request(input_message, args)
  # Capture the end time of the request and compute the duration
    end_time = time.time()
    duration = end_time - start_time
    # Output the duration of the request
    print(f"Request completed in {duration:.2f} seconds.")
    # Write the response to an output file
    write_file("output-data", response)
    print(response)
```

ДОДАТОК Ж

Експертний висновок нормоконтроль

1

Експертний висновок результатів перевірки кваліфікаційної роботи

студент (посада) | програмної інженерії (кафедра) | ІПЗздм-22-1 (група)

Дем'яненко М.С.
(прізвище, ім'я, по батькові)

Зауваження

Пункт ДСТУ 3008-2015	Зміст пункту	Сторінка кваліфікаційної роботи
1	2	3
	7.1 Загальні положення	
7.1.20	Заголовки структурних елементів звіту та заголовки розділів треба друкувати з абзацного відступу великими літерами напівжирним шрифтом без крапки в кінці. Дозволено їх розміщувати посередині рядка.	7, далі за текстом
	7.3 Нумерація сторінок звіту	
	7.4 Нумерація розділів, підрозділів, пунктів, підпунктів	
	7.5 Рисунки	
	7.6 Таблиці	
	7.7 Переліки	
7.7.2	Якщо подають переліки одного рівня підпорядкованості, на які у звіті немає посилань, то перед кожним із переліків ставлять знак «тире». Якщо у звіті є посилання на переліки, підпорядкованість позначають малими літерами української абетки, далі — арабськими цифрами, далі — через знаки «тире». Після цифри або літери певної позиції переліку ставлять круглу дужку.	10
	7.8 Примітки	
	7.9 Виноски	
	7.10 Формули та рівняння	
7.10.1	Формули та рівняння подають посередині сторінки симетрично тексту окремим рядком безпосередньо після тексту, у якому їх згадано. Найвище та найнижче розташування запису формул(и) та/чи рівняння(-нь) має бути на відстані не менше ніж один рядок від попереднього й наступного тексту.	22
7.10.4	Номер формули чи рівняння друкують на їх рівні праворуч у крайньому положенні в круглих дужках. У багаторядкових формулах або рівняннях їхній номер проставляють на рівні останнього рядка.	22
	7.11 Посилання	
	7.13 Список авторів	
	7.14 Скорочення та умовні познаки	
	7.15 Додатки	