ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра інженерії програмного забезпечення

ПОЯСНЮВАЛЬНА ЗАПИСКА
до бакалаврської роботи
на ступінь вищої освіти бакалавр

на тему: «РОЗРОБКА НАВЧАЛЬНОГО МОДУЛЮ З АВТОМАТИЗОВАНОГО ТЕСТУВАННЯ НА ОСНОВІ ТЕХНОЛОГІЙ JAVA, SELENIUM ТА TESTNG»

Виконав: студент 4 курсу, групи ПД–42
спеціальності
121 Інженерія програмного забезпечення
(шифр і назва спеціальності/спеціалізації)

Студент Б.М.
(прізвище та ініціали)

Керівник Негоденко О.В.
(прізвище та ініціали)

Рецензент

(прізвище та ініціали)

Київ – 2023

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра Інженерії програмного забезпечення
Ступінь вищої освіти - «Бакалавр»
Спеціальність підготовки – 121 «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ
Завідувач кафедри
Інженерії програмного забезпечення
Негоденко О.В.
"____" _____ 2023 року

ЗАВДАННЯ
НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТА

Студенту Богдану Михайловичу
(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка навчального модулю з автоматизованого тестування на основі технологій Java, Selenium та TestNG»

Керівник роботи: Негоденко Олена Василівна, к.т.н, доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом вищого навчального закладу від «24» лютого 2023 року №26.

2. Строк подання студентом роботи «01» червня 2023 року.

3. Вхідні дані до роботи:

3.1 Науково-технічна література з питань, пов'язаних з програмним забезпеченням щодо розробки .
3.2 Існуючі інструменти автоматизації тестування ПЗ.
3.3 Науково-технічна література.
3.4 Положення побудови систем автоматизації тестування ПЗ.

4. Зміст розрахунково-пояснювальної записки.

4.1 Аналіз актуальності та проблематики розроблюваного навчального модулю.
4.2 Аналіз та вибір інструментів для реалізації навчального модулю.
4.3 Проектування навчального модулю.
4.4 Висновки.

5. Перелік демонстраційного матеріалу

5.1 Титульний слайд.

5.2 Мета, об’єкт, предмет, наукова новизна дослідження.

5.3 Актуальність.

5.4 Аналіз аналогів.

5.5 Технічні завдання.

5.4 Програмні засоби та інструменти реалізації.

5.6. Розробка архітектури.

5.7 Реалізація програми.

5.8 Висновки.

5.9 Апробація результатів дослідження.

5.10 Кінцевий слайд.

6. Дата видачі завдання «25» лютого 2023р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	25.02.2023	Виконано
2	Дослідження аналогів та актуальності додатку	13.03.2023	Виконано
3	Аналіз та вибір інструментів для розробки додатку	25.03.2023	Виконано
4	Проектування та реалізація	08.04.2023	Виконано
5	Вступ, висновки, реферат	02.05.2023	Виконано
6	Розробка обов‘язкових демонстраційних матеріалів	10.05.2023	Виконано
7	Попередній захист роботи	19.05.2023	Виконано
8	Здача роботи	01.06.2023	Виконано

Студент Студент Б.М.

(підпис) (прізвище та ініціали)

Керівник роботи Негоденко О.В.

(підпис) (прізвище та ініціали)

РЕФЕРАТ

Текстова частина бакалаврської роботи с. 58, табл. 1, рис. 33, джерел 20.

НАВЧАЛЬНИЙ МОДУЛЬ, АВТОМАТИЗОВАНЕ ТЕСТУВАННЯ, JAVA, SELENIUM, TESTNG.

Об'єкт дослідження – процес навчання автоматизованого тестування.

Предмет дослідження – навчальний модуль з автоматизованого тестування програмного забезпечення.

Мета роботи – спрощення процесу навчання технологій автоматизованого тестування за допомогою модуля з технологіями Java, Selenium та TestNG.

Методи дослідження – методи структурного аналізу і проектування, методи розробки програмного забезпечення, методи тестування програмного забезпечення, методи верифікації програмного забезпечення.

Для реалізації поставленої мети потрібно вирішити наступні завдання:

1.Провести аналіз ринку навчальних модулів та систем в цій галузі.

2. Проаналізувати переваги і недоліки програмних інструментів розробки.

3. Розглянути та проаналізовати особливості наявних систем навчання автоматизованого тестування.

4. Створити навчальний модуль за допомогою мови розмітки HTML та стилю сторінок CSS у середовищі розробки IntelliJ IDEA мовою програмування Java з використанням технологій Selenium та TestNG.

5. Використати Figma для створення елементів графічного дизайну та прототипування.

Практичне значення отриманих результатів: Даний навчальний модуль слугує для того, щоб надавати навчальні матеріали для користувачів, які прагнуть вивчити основи автоматизованого тестування та опанувати сучасні інструменти створення авто тестів.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

QA – Quality Assurance

AQA – Automation Quality Assurance

IDE – integrated development environment

ПЗ – програмне забезпечення

ВСТУП

У сучасному світі, де швидкість розвитку програмного забезпечення щоразу зростає, автоматизоване тестування стає необхідною складовою процесу розробки. Навички роботи з автоматизованими тестами допомагають підвищити ефективність тестування, зменшити тривалість процесу, знизити ймовірність помилок та забезпечити більш швидку і якісну доставку програмного забезпечення.

Зростаюча кількість платформ, на яких розробляється програмне забезпечення, робить процес тестування ще більш складним завданням. Тестування на різних платформах вимагає від фахівців високої кваліфікації та знань про інструменти автоматизованого тестування.

Крім того, недостатня кількість фахівців із знанням автоматизованого тестування створює значну кількість ризиків для процесу розробки програмного забезпечення. Такі ризики можуть призвести до затримок у розробці, недоліків у програмному забезпеченні та його неякісної функціональності.

Створення навчального модулю з вивчення автоматизованого тестування має на меті надати можливість вивчити основні принципи та інструменти автоматизованого тестування програмного забезпечення. Такий модуль допоможе надати загальне уявлення про використання технологій, які зможуть забезпечити якість програмного забезпечення в різних сферах діяльності.

Зокрема, вивчення автоматизованого тестування може бути корисним для користувачів, які планують працювати в проектах пов’язаних з галузями медицини, банківської справи, фінансів, телекомунікацій та інших сфер діяльності, де розробка програмного забезпечення є необхідною. Також, навчальний модуль може забезпечити можливість підвищити кваліфікацію фахівців, які вже працюють у сфері розробки програмного забезпечення, та допомогти їм здійснювати свою роботу більш ефективно та професійно.

Отже, створення навчального модулю з вивчення автоматизованого тестування є дуже актуальною темою на теперішній момент, де значна кількість сфер діяльності вимагає якісного програмного забезпечення. Навчальний модуль

зможе надати базові необхідні знання, навички та інструменти, щоб забезпечити високу якість програмного забезпечення.

Об'єкт дослідження – процес навчання автоматизованого тестування.

Предмет дослідження – навчальний модуль з автоматизованого тестування програмного забезпечення.

Мета роботи – спрощення процесу навчання технологій автоматизованого тестування за допомогою модуля з технологіями Java, Selenium та TestNG.

Модуль повинен мати низький поріг входу та простий та мінімалістичний, для легкої адаптації нових користувачів.

Для реалізації поставленої мети потрібно вирішити наступні завдання:

1. Провести аналіз ринку навчальних модулів та систем в цій галузі.

2. Проаналізувати переваги і недоліки програмних інструментів розробки.

3. Розглянути та проаналізувати особливості наявних систем вивчення автоматизованого тестування.

4. Створити навчальний модуль за допомогою мови розмітки HTML та стилю сторінок CSS у середовищі розробки IntelliJ IDEA мовою програмування Java з використанням технологій Selenium та TestNG.

5. Використати Figma для створення елементів графічного дизайну та прототипування.

Практичне значення отриманих результатів: Даний навчальний модуль слугує для того, щоб надавати навчальні матеріали для користувачів, які прагнуть вивчити основи автоматизованого тестування та опанувати сучасні інструменти створення авто тестів.

Галузь використання – розробка програмного забезпечення для підвищення якості та ефективності тестування.

1. ОГЛЯД АРХІТЕКТУРИ ІСНУЮЧИХ НАВЧАЛЬНИХ МОДУЛІВ ТА АНАЛІЗ ПРОЕКТУ

1.1 Аналіз та загальна характеристика навчальних систем та вивчення автоматизованого тестування

Навчальні модулі - це комп'ютерні програми, створені для навчання людей різних вікових та професійних категорій. Основна мета навчальних модулів полягає в тому, щоб допомогти користувачам засвоїти нові знання та навички шляхом пропонування різноманітних завдань та тестів.

Сучасні технології навчання не стоять на місці, і навчальні модулі є однією з найбільш перспективних галузей, в якій в інноваційних компаніях та університетах вкладаються значні суми грошей. Це пояснюється тим, що навчальні модулі можуть бути ефективним засобом навчання для різних категорій людей, зокрема, для студентів, працівників компаній, які хочуть пройти перепідготовку, та для тих, хто просто хоче отримувати нові знання та навички.

Однією з переваг навчальних модулів є те, що вони можуть бути легко оновлюваними та адаптованими до потреб користувачів. Завдяки цьому, можна швидко та ефективно додавати новий матеріал та функції, а також виправляти помилки та оновлювати вже наявний матеріал.

Навчальні модулі мають певні переваги над традиційними методами навчання, зокрема, вони можуть бути доступними для навчання в будь-який час та з будь-якого місця, що робить їх особливо зручними для тих, хто має зайнятий графік. Крім того, вони забезпечують можливість повторення матеріалу, який вже був вивчений, що дозволяє більш якісно засвоювати знання.

Що стосується засобів навчання, то навчальні модулі вирізняються від традиційних методів навчання наявністю інноваційних функцій та можливостей. Так, модулі можуть бути інтерактивними, забезпечувати багато різноманітних рівнів складності завдань, містити візуальні елементи та інші корисні функції, що роблять процес навчання більш цікавим та зрозумілішим для користувачів.

Обов’язковими ознаками навчальних модулів можна вирізнити:

- Інтеграція окремих та комплексних технологій для навчання;
- Повне охоплення навчального матеріалу в межах окремих модулів;
- Відносна самостійність елементів модуля;
- Ефективний та оптимальний спосіб передачі інформації та методичних матеріалів.

Навчальні модулі забезпечують користувачам можливість самостійно вибирати теми, які їх цікавлять, та займатися навчанням в зручний для них час. Крім того, вони дозволяють контролювати свій прогрес та отримувати зворотний зв'язок щодо виконання завдань та тестів.

У світі, де технології постійно розвиваються, навчальні модулі стають все більш популярними та потрібними. Вони забезпечують зручний та ефективний спосіб навчання, що дозволяє здобувати нові знання та навички швидше та з меншими зусиллями. Тому, вивчення нової інформації та опанування навичок стали значно швидшими та доступнішими для багатьох категорій людей.

Багато процесів вивчення нового матеріалу підлягають інтерактивним методам викладання, для спрощення засвоєння та зниження порогу входу. І саме тут навчальні модулі можуть бути надзвичайно корисними, забезпечуючи відповідні інструменти та можливості, що дозволяють навчатися більш ефективно та результативно.

Отже, можна зробити висновок, що навчальні модулі - це вагомий та перспективний інструмент для навчання, який дозволяє здобувати нові знання та навички швидше та з меншими зусиллями. Їхні переваги полягають у зручності та доступності для користувачів, ефективності та можливості контролювати свій прогрес. Таким чином, навчальні модулі заслуговують на увагу та розгляд як один з найбільш перспективних засобів навчання в сучасному світі.

1.2 Аналіз уже наявних аналогів

Для успішного вивчення автоматизованого тестування програмного забезпечення, важливо мати доступ до навчальних модулів, що містять велику

кількість практичних завдань та інтерактивних матеріалів. Такі практики допоможуть закріплювати отримані знання та навички, а інтерактивні матеріали, такі як документація та відеоуроки, допоможуть краще зрозуміти теоретичні концепції. Порівняємо навчальні модулі наступних платформ: W3school, QAlearning, Udemy та Coursera.

1.2.1 W3school

W3school - це один з найбільш відомих та відвідуваних сайтів для вивчення програмування та автоматизації тестування. Цей ресурс є безкоштовним та надає безліч корисної інформації, що може стати у пригоді як початківцям, так і професіоналам.

Однією з найбільш корисних секцій для тестувальників програмного забезпечення є курс по Selenium. Цей курс дозволяє вивчити основи та використання цього інструменту для автоматизованого тестування веб-додатків. У рамках курсу студенти дізнаються, як створювати тестові скріпти, які можна використовувати для перевірки роботи веб-додатків. Крім того, на курсі розглядаються основні функції та можливості інструмента, що допоможе навчитися створювати їх ефективніше.

Ще одним важливим курсом є навчальний курс по TestNG. Цей фреймворк дозволяє вивчити основи та використання фреймворку для тестування Java-додатків. Курс розрахований на тих, хто вже має певний досвід у програмуванні на Java, але не знайомий з TestNG. На курсі дізнаєтесь про основні функції та можливості фреймворку, а також про те, як можна використовувати його для виконання тестових задач.

Крім цих курсів, на W3school є інші матеріали, що стосуються автоматизованого тестування, такі як курси по JUnit, Cucumber та Appium. Курс по JUnit дозволяє вивчити основи тестування Java-програм, використовуючи JUnit. Курс по Cucumber дозволяє вивчити основи тестування з використанням Cucumber, який дозволяє створювати тести на основі покрокових описів. Курс по Appium дозволяє вивчити основи тестування мобільних додатків на різних платформах.

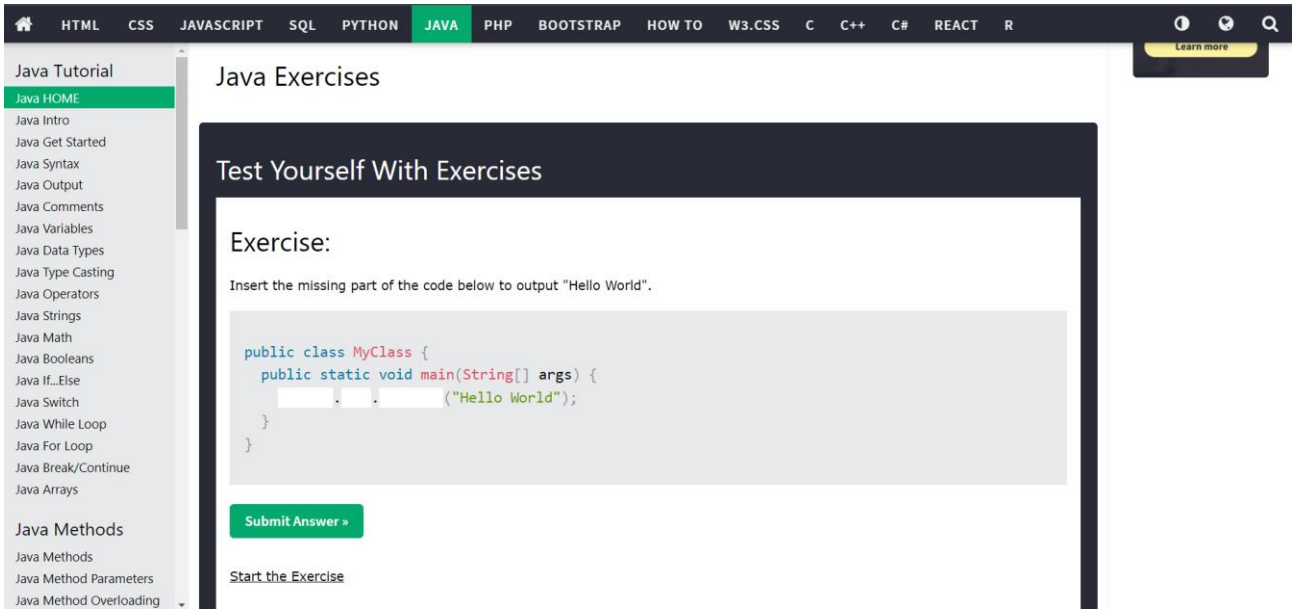


Рисунок 1.1 Сайт W3school. Модуль вивчення Java

Однією з головних переваг W3school є те, що він є безкоштовним. Це дозволяє студентам та початківцям займатися вивченням програмування та автоматизованим тестуванням безкоштовно, що є важливим кроком для отримання нових знань та навичок. Крім того, на W3school є безліч різноманітних навчальних матеріалів, таких як статті, презентації та практичні завдання.

Проте, не дивлячись на багато корисної інформації, що міститься на W3school, цей ресурс має певні недоліки. Наприклад, відсутність великої кількості відео уроків та практичних завдань може стати проблемою для тих, хто більше віддає перевагу візуальному навчанню та практичним заняттям. Також, деякі курси можуть бути застарілими, оскільки технології швидко розвиваються та оновлюються.

Усе ж таки, W3school є чудовим ресурсом для отримання нових знань та навичок у вивченні автоматизованого тестування та програмування загалом. Завдяки великому вибору курсів та матеріалів, студенти можуть вивчати автоматизоване тестування на різних рівнях складності та використовувати отримані знання в своїй роботі.

1.2.2 QAlearning

QAlearning - це платформа, яка допоможе вивченні основ тестування програмного забезпечення. На платформі доступний навчальний модуль для вивчення автоматизованого тестування з використанням Selenium та TestNG. У свою чергу це одні з найпопулярніших інструментів для автоматизованого тестування, і вивчення їх використання може допомогти спеціалісту стати більш конкурентоспроможним на ринку праці.

Навчальний модуль надає багато корисних матеріалів та інструментів для вивчення Selenium. Доступність багатьох відео уроків та практичних завдань допоможе вам засвоїти нові знання та отримати практичний досвід, що може бути особливо корисним для тих, хто тільки починає вивчати автоматизоване тестування.

Окрім вище згаданих технологій, QAlearning також пропонує курси для вивчення мови програмування Java, яка є дуже популярною серед тестувальників програмного забезпечення. Вивчення цієї мови програмування дозволить створювати складніші тестові сценарії та розробляти власні фреймворки для автоматизованого тестування.

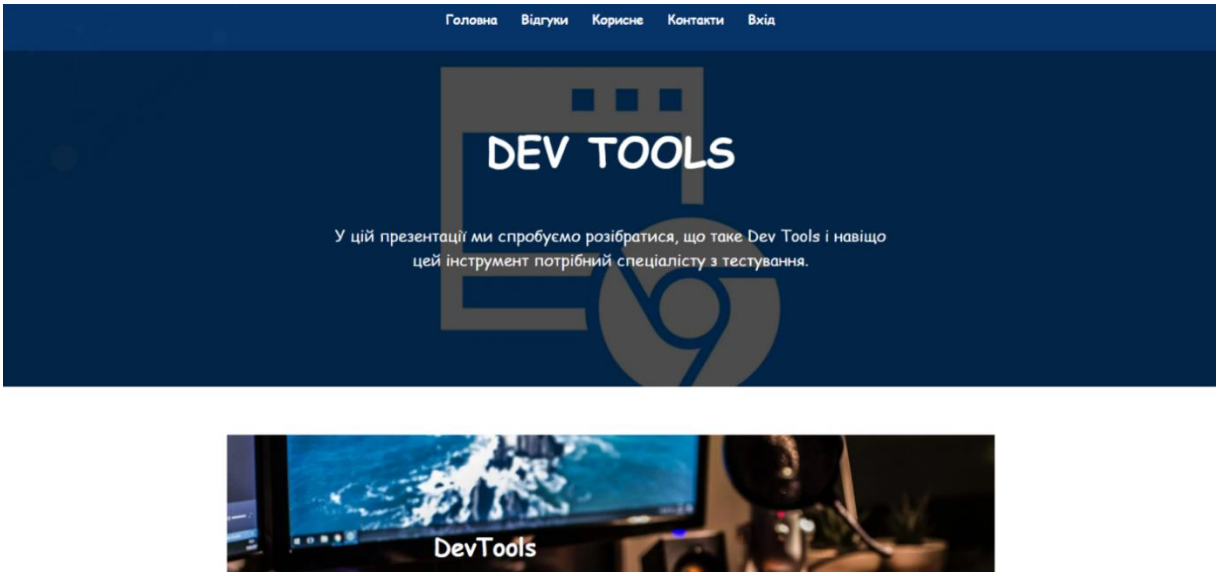


Рисунок 1.2 Сайт QAlearning. Навчальне відео

Якщо студент володіє базовими знаннями Java, на QAlearning є ресурси, де

можна знайти курси, які допоможуть подальшому вивченні використання цих інструментів разом. Крім того, на платформі доступні матеріали для вивчення інших інструментів для авто тестів, таких як Appium, JMeter та TestNG.

Однією з переваг цієї платформи є можливість отримати сертифікати після проходження курсів. Вони можуть бути корисними при пошуку роботи у айті галузі.

Незважаючи на те, що цей модуль є платним, пропонується багато корисних матеріалів та інструментів для вивчення Selenium та TestNG. Вивчення автоматизованого тестування з QAlearning може стати корисним для розвитку кар'єри в тестуванні програмного забезпечення.

1.2.3 Udemy

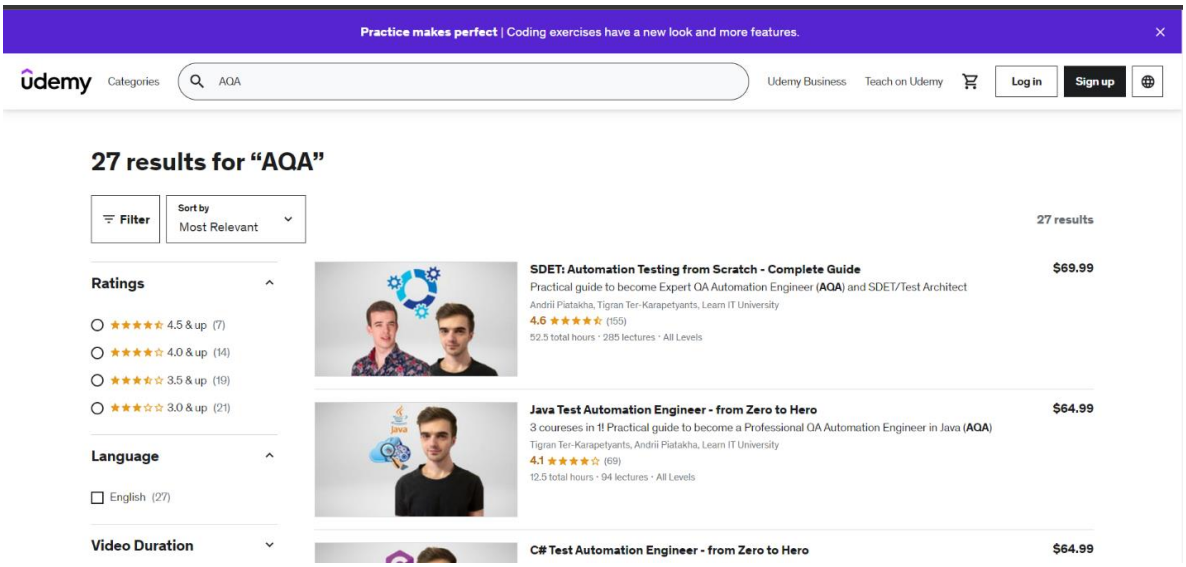


Рисунок 1.3 Сайт Udemy. Меню з вибором відео-курсів.

Udemy є однією з найбільш популярних та якісних платформ для вивчення автоматизованого тестування. Ця платформа пропонує безліч корисних курсів, що допоможуть вам зрозуміти, як правильно проводити автоматизоване тестування програмного забезпечення.

Вивчення автоматизованого тестування на платформі Udemy може стати відмінним вибором для тих, хто прагне стати професійним тестувальником

програмного забезпечення. Платформа пропонує широкий вибір курсів, які допоможуть вам засвоїти необхідні знання та навички, щоб ефективно проводити автоматизоване тестування.

TestNG - один з найбільш популярних інструментів для автоматизованого тестування на Java. Вивчення цього тестового фреймворку є дуже важливим для тих, хто прагне стати професійним тестувальником програмного забезпечення. На платформі Udemy велика кількість курсів з вивчення TestNG, що допоможуть легко стати засвоїти необхідні знання. Навчальні матеріали детально розкривають функціональні можливості тестогово фреймворку та дозволять створювати багаторівневі тести та сценарії.

Ще одним плюсом вивчення автоматизованого тестування на платформі Udemy є те, що на цій платформі є багато відеоуроків та практичних завдань, що допомагає краще засвоїти матеріал. Такий формат навчання є більш ефективним, оскільки дозволяє студенту більше часу приділяти практичному вивченню та вирішенню реальних завдань. Ви зможете практикуватись у створенні автоматизованих тестів та відлагодженні проблем, що допоможе вам зрозуміти, як ефективно використовувати TestNG у своїй роботі.

Крім того, на платформі Udemy є можливість отримати сертифікат за успішне завершення курсу, що може бути корисним при пошуку роботи в галузі тестування. Такий сертифікат свідчить про наявність необхідні знання та навички, що потрібні для роботи в галузі автоматизованого тестування.

Також варто зазначити, що на платформі Udemy є не тільки курси з відеоуроками, але й інші типи матеріалів, такі як статті, електронні книги та завдання для самостійного вивчення. Це дозволяє студентам вивчати матеріал у форматі, який найбільше підходить для них. Ви можете використовувати різноманітні матеріали, щоб засвоїти матеріал більш ефективно та з комфортом.

Крім того, на Udemy є можливість взяти курси від провідних експертів у галузі автоматизованого тестування. Це дає можливість студентам навчитися від найкращих, отримати цінні поради та рекомендації, що допоможуть їм стати експертами у своїй галузі. Ви зможете отримати безцінний досвід та знання від

фахівців, що успішно працюють у сфері тестування програмного забезпечення.

Завдання, які можна виконувати на платформі Udemy, пов'язані з практичною роботою з TestNG та Java. Ви зможете створювати складні тести, практикуватись у відлагодженні проблем, та засвоїти найкращі методики та практики у галузі автоматизованого тестування.

1.2.4 Coursera

Один із найбільш популярних курсів з автоматизованого тестування на платформі Coursera є "Java Programming: An Introduction to Software". Цей курс викладається на англійській мові та складається з 5 тижнів. За допомогою цього курсу, ви будете мати можливість навчитися писати автоматизовані тестові скрипти на мові програмування Java, що є однією з найпопулярніших мов програмування для тестування.

Курс "Java Programming: An Introduction to Software" на платформі Coursera дозволяє вивчати основи автоматизованого тестування на мові програмування Java. Курс складається з 5 тижнів, кожен з яких зосереджений на певних аспектах тестування.

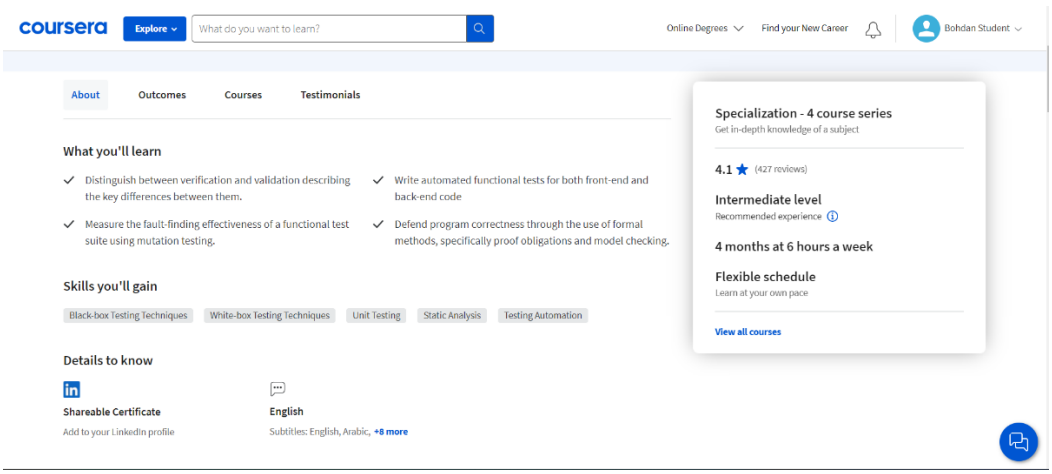


Рисунок 1.4 Сайт Coursera. Сторінка із начальним курсом

Ще, перед початком навчання на платформі Coursera, варто врахувати як переваги, так і недоліки цієї платформи, щоб зробити свій вибір та максимально

використати можливості навчання. Однією з переваг Coursera є те, що ця платформа пропонує багато курсів з різних галузей, включаючи IT. Крім того, курси на платформі Coursera є абсолютно гнучкими, тому можливо вчитися в будь-який зручний для час.

Варто зазначити, що відсутність можливості отримати особисту допомогу від викладачів є однією з найбільших недоліків вивчення на платформі Coursera. Це може бути проблемою для тих, хто швидко не засвоює нові матеріали або має запитання щодо певних тем. Також, важливо зазначити, що курс на платформі Coursera може не підійти тим, хто шукає інтенсивний темп навчання або хоче отримати сертифікат за успішне проходження курсу.

1.2.5 Порівняння

Вивчення технології Java: Якщо відвідувачі платформи бажають вивчати технологію Java, то вони можуть обрати усі вище згадаті модулі. На цих платформах є курси з вивчення Java та автоматизованого тестування програмного забезпечення.

Вивчення технології Selenium: Якщо відвідувачі платформи бажають вивчати технологію Selenium, то вони можуть обрати наступні платформи: W3school, QAlearning та Coursera. Усі ці платформи містять курси з вивчення Selenium та автоматизованого тестування програмного забезпечення.

Вивчення технології TestNG: Якщо відвідувачі платформи бажають вивчати технологію TestNG, то вони можуть обрати наступні платформи: QAlearning та Udemy. Ці дві платформи містять курси з вивчення TestNG та автоматизованого тестування програмного забезпечення.

Нижче наведена таблиця з перевагами та недоліками платформ для вивчення автоматизованого тестування, включаючи курси з вивчення Selenium та TestNG. Таблиця містить додаткову інформацію про наявність курсів з вивчення Java на кожній з платформ.

Таблиця 1.1 – Переваги та недоліки популярних навчальних модулів

Платформа	**Переваги**	**Недоліки**	**Наявність курсів з вивчення Selenium**	**Наявність курсів з вивчення TestNG**	**Наявність курсів з вивчення Java**
W3school	Безкоштовна, містить багато корисної інформації	Не має великої кількості відеоуроків та практичних завдань	Так	Ні	Так
QAlearning	Має багато відеоуроків та практичних завдань	Платна	Так	Так	Так
Udemy	Має багато курсів з відеоуроками та практичними завданнями	Платна	Ні	Так	Так
Coursera	Має багато курсів з відеоуроками та практики	Платна	Так	Ні	Так

Підводячи підсумки, можна зробити висновок, що для вивчення Selenium, TestNG та Java існують різні платформи, кожна з яких має свої переваги та недоліки.

Кожна платформа має свої переваги та недоліки, які можуть бути важливими при виборі платформи для навчання.

Перша платформа, яку розглядаємо, - W3school. Вона є безкоштовною та містить багато корисної інформації, що робить її чудовим варіантом для тих, хто хоче познайомитись з основними поняттями Selenium, TestNG та Java. Однак, W3school не має великої кількості відеоуроків та практичних завдань, що може бути недостатньо для тих, хто шукає більше практики та поглибленого вивчення.

Наступна платформа - QAlearning. Вона має багато відеоуроків та практичних завдань, що робить її однією з найкращих платформ для вивчення Selenium, TestNG та Java. Однак, QAlearning є платною платформою, що може стати проблемою для тих, хто має обмежений бюджет.

Udemy - це ще одна платформа для навчання, яка має багато курсів з відеоуроками та практичними завданнями. Вона є платною, але може бути кращим варіантом для тих, хто шукає багато курсів не лише для вивчення Selenium, TestNG та Java, але й для інших технологій.

Нарешті, Coursera - ще одна платформа з багатою кількістю курсів з відеоуроками та практичними завданнями. Також як і QAlearning, Coursera є платною, але має перевагу в тому, що вона пропонує курси з вивчення Selenium, TestNG та Java. Це може бути важливим фактором при виборі платформи для вивчення.

Загалом, якщо важливо вивчити всі три технології, то QAlearning може бути кращим варіантом, оскільки вона пропонує курси з вивчення Selenium, TestNG та Java. Якщо важливо зосередитись на вивченні Java, то можна вибрати W3school або Coursera. Udemy може бути кращим варіантом для тих, хто шукає багато курсів з відеоуроками та практичними завданнями в цілому, а не лише для вивчення Selenium, TestNG та Java.

1.3 Актуальність проекту

Тенденції в галузі свідчать про значне зростання попиту на молодших спеціалістів з автоматизації QA. Останнім часом багато компаній почали усвідомлювати переваги автоматизованого тестування, що призвело до зростання

кількості вакансій для молодших спеціалістів з автоматизації QA. Хоча в деяких випадках перевага надається попередньому досвіду в автоматизованому тестуванні, більшість вакансій відкриті для свіжих випускників або осіб з невеликим досвідом роботи або взагалі без нього.

Варто зазначити, що найбільш затребуваними технологіями для молодших спеціалістів з автоматизації QA є Java, Selenium та TestNG. Вони широко використовуються в індустрії, і їхнє розуміння може дати перевагу під час пошуку роботи в цій галузі.

Лише в Україні за останні кілька місяців з'явилося понад 100 вакансій для молодших спеціалістів з автоматизації QA. Це яскраве свідчення того, що компанії готові інвестувати в нові таланти і активно шукають людей, які володіють необхідними навичками і знаннями в цій галузі. Зростання попиту на молодших спеціалістів з автоматизації QA є позитивним явищем для галузі, і очікується, що ця тенденція збережеться в найближчі роки.

Автоматизоване тестування стало необхідністю для бізнесу, оскільки воно дозволяє компаніям економити час і зусилля, а також зменшити витрати на наймання спеціалізованих тестувальників. Крім того, автоматизоване тестування допомагає виявити баги і помилки в програмному забезпеченні на ранній стадії, що може значно знизити загальні витрати на розробку і тестування програмного забезпечення. Оскільки все більше компаній визнають переваги автоматизованого тестування, попит на людей з необхідними навичками і знаннями в цій галузі, ймовірно, буде ще більше зростати.

Створення такого навчального модулю має на меті надати можливість студентам та спеціалістам з інших галузей вивчити технології автоматизованого тестування та навчитися їх використовувати в роботі.

Вище згадані технології є дуже популярними в сфері автоматизації. Розробка навчального модулю з використанням цих фреймворків та засобів дозволить користувачам ознайомитися з ними та навчитися їх використовувати в практичній роботі.

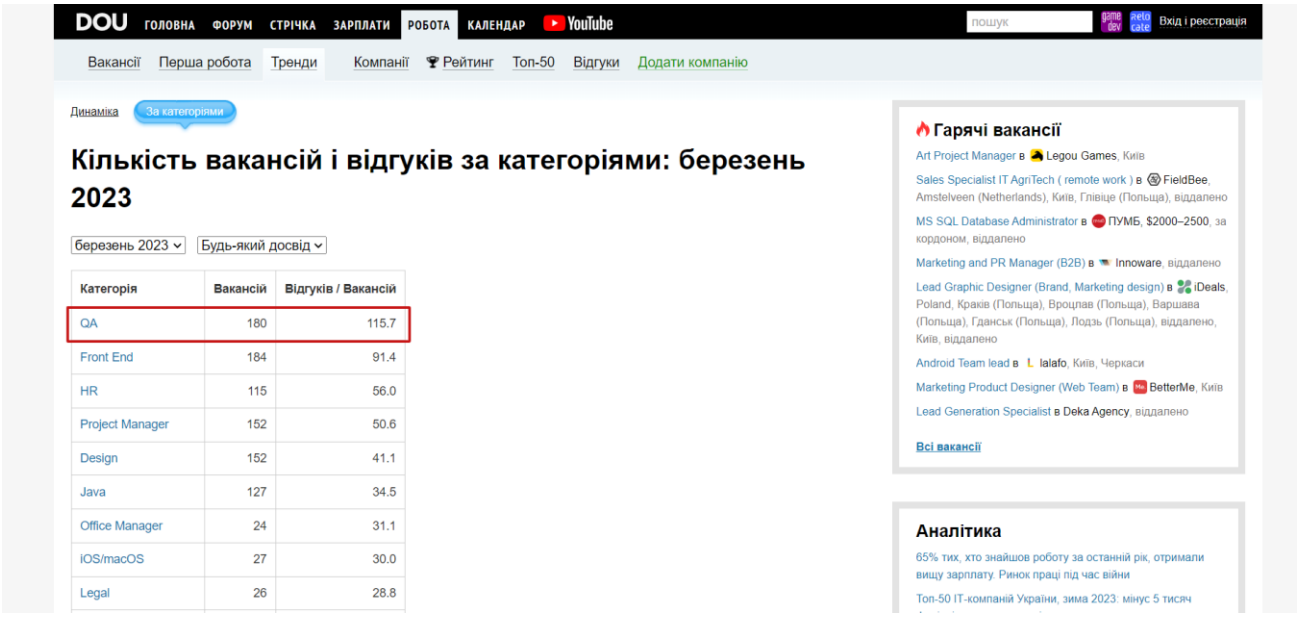


Рисунок 1.5 Сайт DOU. Статистика кількості вакансій

Крім того, створення навчального модулю з автоматизованого тестування є вкрай важливим на сьогоднішній день, оскільки розробники програмного забезпечення все частіше використовують різні технології та фреймворки, які потребують автоматизованого тестування. Таким чином, знання технологій автоматизованого тестування з використанням Java, Selenium та TestNG є надзвичайно важливим для розробників програмного забезпечення, тестувальників, які відповідають за якість та безпеку програмного забезпечення.

Створення навчального модулю з автоматизованого тестування з використанням цих технологій є дуже актуальним та корисним для студентів та нових людей у професії з інших галузей, які бажають вивчити технології автоматизованого тестування та навчитися їх використовувати у своїй роботі. Крім того, він може бути корисним для компаній, які мають власне програмне забезпечення та відповідають за його якість. Більш того, студенти та люди, що планують змінити професію, які вивчають ці технології, можуть знайти роботу в компаніях, яким потрібні нові кандидати на посаду автоматизаторів.

Завдяки такому модулю студенти зможуть поглибити свої знання та вміння з автоматизованого тестування. Вони зможуть ознайомитися з основними поняттями, принципами та інструментами автоматизації тестування, а також

навчитися їх використовувати в практичній роботі.

Крім того, користувачі з інших галузей, які вивчатимуть цей модуль, зможуть здобути нові знання та вміння, які можуть стати додатковими перевагами при пошуку роботи.

Також варто зазначити, що автоматизація тестування дозволяє виявляти помилки в програмному забезпеченні на ранніх етапах розробки, що зменшує витрати на подальшу розробку та тестування.

Отже, створення навчального модулю з автоматизованого тестування з використанням технологій Java, Selenium, TestNG є вкрай важливим та актуальним завданням на сьогоднішній день, яке має не тільки академічне значення, а й практичну користь для бізнесу та розробників програмного забезпечення.

Додатково можна зазначити, що з пандемією коронавірусу більшість навчальних закладів перейшли на дистанційне навчання, що зробило актуальним створення навчальних матеріалів з автоматизованого тестування онлайн. Такий модуль зможе стати додатковим ресурсом для студентів та викладачів, які перейшли на дистанційне навчання.

Також можна розглянути можливість розширення змісту модулю з автоматизованого тестування шляхом додавання нової інформації, наприклад, про підхід Behavior-driven development (BDD) та інші.

2. ПРОЕКТУВАННЯ НАВЧАЛЬНОГО МОДУЛЮ З АВТОМАТИЗОВАНОГО ТЕСТУВАННЯ

2.1 Аналіз та обґрунтування вибору середовища та інструментів розробки проекту

В цьому розділі будуть розглянуті основні інструменти та середовище розробки, обрані для створення навчального модулю з вивчення автоматизованого тестування.

Для створення навчального модулю було обрано мову програмування Java. Ця мова є однією з найпопулярніших мов програмування в світі, що забезпечує наявність великої кількості ресурсів для вивчення та підтримки. Крім того, Java є мовою програмування, яка є платформо-незалежною, тобто код, написаний на мові Java, можна запускати на будь-якій операційній системі, що є дуже зручним.

Для розробки тестів було обрано фреймворк Selenium. Він є одним з найпопулярніших фреймворків для автоматизованого тестування веб-додатків. Він дозволяє розробляти тестові скріпти з використанням різних мов програмування, включаючи Java. Більш того, Selenium має багато корисних функцій, таких як можливість виконувати тести на різних веб-браузерах, що дозволяє перевірити, як добре працює програма на різних платформах.

Для структурування тестів та підвищення їх якості було обрано фреймворк TestNG. TestNG дозволяє писати більш структурований тестовий код та використовувати анотації для різних етапів тестування. Крім того, TestNG підтримує паралельне виконання тестів, що дозволяє зменшити час виконання тестового набору та прискорити процес тестування.

Середовище розробки IntelliJ IDEA було обрано через його можливості та популярність серед розробників. IntelliJ IDEA є інтегрованою середовище розробки (IDE), яка надає можливості для роботи з Java та іншими мовами програмування. Крім того, IntelliJ IDEA має ряд корисних функцій, таких як автодоповнення коду, вбудоване тестування та дебагер, що дозволяє зменшити час розробки та

підвищити якість написаного коду.

Отже, вибір мови програмування Java та фреймворків Selenium та TestNG, а також середовища розробки IntelliJ IDEA є обґрунтованим та дозволить створити якісний навчальний модуль з вивчення автоматизованого тестування.

2.1.1 IntelliJ IDEA

IntelliJ IDEA - це потужне інтегроване середовище розробки програмного забезпечення, яке підтримує різні мови програмування, зокрема Java, Kotlin, Groovy, Scala, JavaScript та багато інших. Розроблене компанією JetBrains, це середовище розробки відоме своїми широкими можливостями та високою продуктивністю.

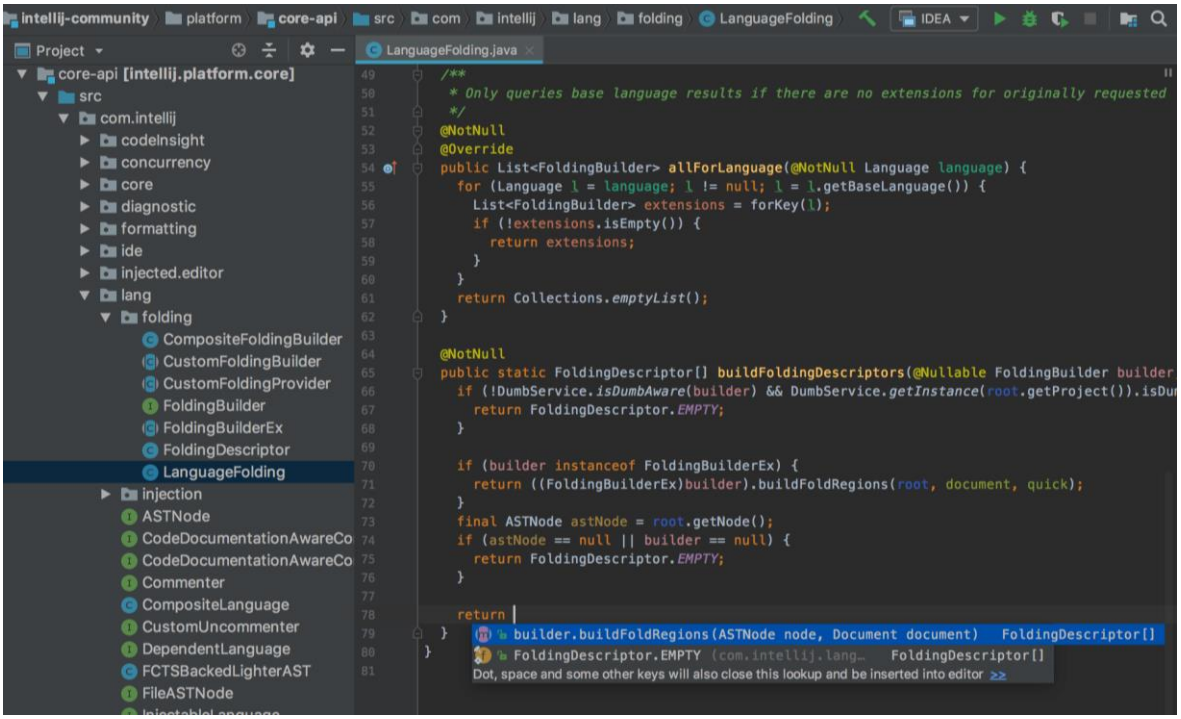


Рисунок 2.1 Середовище розробки IntelliJ IDEA

Нижче наведено кілька ключових функцій середовища розробки:

Розширена підтримка мов програмування

IntelliJ IDEA підтримує багато мов програмування, включаючи Java, Groovy, Scala, JavaScript, TypeScript, HTML, CSS, SQL та багато інших. Іншою корисною перевагою, цього середовища є забезпечення інтеграції зі швидкими інструментами

перетворення коду, що дозволяє легко переключатися між різними мовами програмування та робити перевірку коду для кожної мови розробки.

Підтримка інструментів системи контролю версій

Середовище підтримує інтеграцію з різними системами контролю версій, такими як Git, SVN, Mercurial тощо. Це дозволяє розробникам працювати з кодом в команді та зберігати його історію. Також, IntelliJ IDEA забезпечує можливість роботи з різними гілками проекту, що дозволяє працювати з різними версіями коду та визначати конфлікти.

Вбудований дебагер

IntelliJ IDEA має вбудований дебагер з можливістю відстеження виконання коду та вирішення проблем в процесі відладки. Це дозволяє розробникам ефективно виявляти та виправляти помилки. Варто додати, що ця система забезпечує можливість відстеження статусу змін в коді та швидке виявлення помилок.

Вбудована система збирання та залежностей

Варто згадати систему збирання та залежностей, таку як Maven та Gradle. Це дозволяє розробникам легко створювати та керувати проектами з різними залежностями. Іншою зручністю є забезпечення можливість автоматичного визначення залежностей та їх встановлення.

Підтримка розширень

IntelliJ IDEA надає велику кількість розширень для покращення функціоналу персоналізації та спрощення використання. Ці файли можуть бути встановлені безпосередньо з магазину розширень, та пристутня можливість створення власних розширень та їх використання.

Вбудована підтримка тестування

Однією з ключових особливостей підтримки автоматизованого тестування в цій системі є інтеграція з популярними тестовими фреймворками, такими як JUnit, TestNG, Cucumber та Arquillian. Це дозволяє розробникам створювати та запускати тести за допомогою улюбленого інтсрументу тестування безпосередньо у IDEA. Розширені можливості для тестування, такі як аналіз тестового покриття,

налагодження тестів та створення тестових звітів, дуже зручні у використанні.

Функція аналізу тестового покриття допомагає тестувальникам визначити області коду, які не покриті тестами. Це дозволяє створювати додаткові перевірки, щоб переконатися, що всі частини коду покриті тестами. Інша корисна функція - налагодження тестів , що дозволяє налагоджувати та модифікувати тестові артефакти, що полегшує виявлення та вирішення проблем.

Середовище розробки пропонує підтримку систем безперервної інтеграції (CI) та безперервної доставки (CD), таких як Jenkins, Travis CI та TeamCity. Це дозволяє тестувальникам автоматизувати перевірки та розгортання своїх додатків, гарантуючи, що зміни коду тестуються та розгортаються автоматично.

На додаток до вбудованих функцій тестування, також підтримка сторонніх тест інструментів та плагінів. Наприклад, IDE забезпечує інтеграцію з популярними інструментами для перевірки якості коду, такими як SonarQube, FindBugs і PMD. Ці інструменти дозволяють розробникам аналізувати свій код і виявляти потенційні проблеми, такі як запах коду, уразливості безпеки та проблеми з продуктивністю.

Загалом, підтримка автоматизованого тестування в рамках проекту IntelliJ IDEA є важливим інструментом для розробників, які хочуть бути впевнені, що їхні додатки будуть якісними та функціонуватимуть належним чином. Завдяки підтримці популярних фреймворків тестування, аналізу тестового покриття, налагодження тестів та інтеграції з системами CI/CD, IntelliJ IDEA полегшує розробникам написання та запуск тестів, гарантуючи, що їхній код ретельно перевірений та готовий до розгортання.

Підтримка роботи з базами даних

IntelliJ IDEA підтримує роботу з різними базами даних, такими як MySQL, Oracle, PostgreSQL тощо. Це дозволяє розробникам зручно працювати з базами даних під час розробки своїх додатків. Середовище забезпечує можливість відображення структури баз даних та генерування SQL запитів.

Вбудована підтримка створення веб-додатків

IDE має вбудовану підтримку створення веб-додатків з використанням

фреймворків, таких як Spring MVC, Struts2 тощо. Це дозволяє розробникам створювати веб-додатки більш ефективно та швидко.

Висновок: IntelliJ IDEA - це потужне інтегроване середовище розробки, яке забезпечує широкі можливості для розробки додатків на мові програмування Java та інших мовах. Завдяки вбудованим інструментам та підтримці різних фреймворків та технологій, розробка додатків стає більш ефективною та продуктивною. IDE забезпечує можливість швидкого та ефективного розв'язання різних завдань, що дозволяє розробникам працювати більш продуктивно та ефективно.

2.1.2 Selenium

Для вивчення ключових елементів автоматизації та створення авто тестів та автоматизації використана бібліотека Selenium WebDriver.

Selenium WebDriver є набором API з відкритим вихідним кодом, який дозволяє взаємодіяти з будь-яким з сучасних веб-браузерів і автоматизувати дії користувача в цьому браузері. Це є важливою складовою сімейства інструментів Selenium, які призначені для автоматизації тестування.

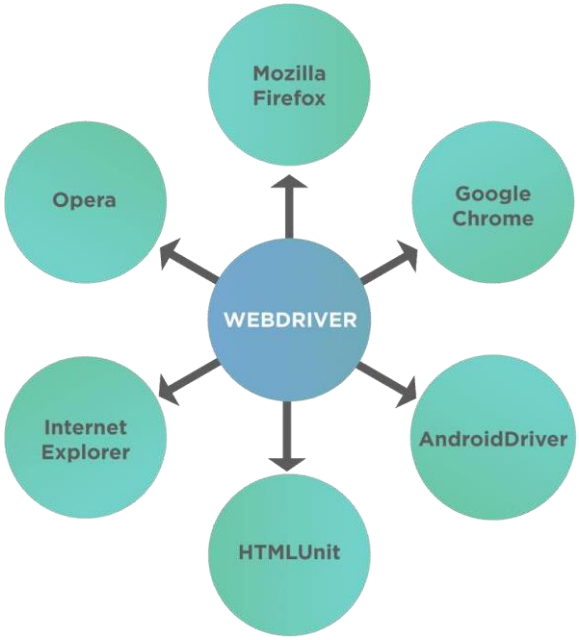


Рисунок 2.2. Підтримувані драйвери браузерів Selenium Web– Driver

WebDriver має кілька унікальних характеристик, які додають йому популярності як інструменту веб-автоматизації. Серед цих характеристик можна виділити:

• Сумісність із декількома браузерами - це одна з основних переваг Selenium та WebDriver. За допомогою цього інструменту можна запускати той самий код скриптів на різних браузерах, що дозволяє імітувати поведінку реального користувача, використовуючи власну підтримку браузера. Це дає можливість виконувати прямі виклики API без необхідності використання будь-якого проміжного програмного забезпечення або пристрою. Список підтримуваних браузерів можна знайти на рисунку 2.2.

• Багатомовна підтримка - оскільки Selenium підтримує багато мов програмування, спеціаліст може використовувати JS, Python, Java, C# та Ruby. Це дає можливість вибрати мову програмування, яка є зручною для користувача та використовується на проекті загалом.

• Швидке виконання - на відміну від Selenium RC, WebDriver не залежить від сервера проміжного програмного забезпечення для взаємодії з браузером. WebDriver спілкується з браузерами за допомогою визначеного протоколу (JSON Wire), що дозволяє йому працювати швидше, ніж більшість інших інструментів Selenium. Крім того, оскільки JSON Wire використовує дуже простий JSON для передачі даних, обсяг передачі даних при виклику є мінімальним.

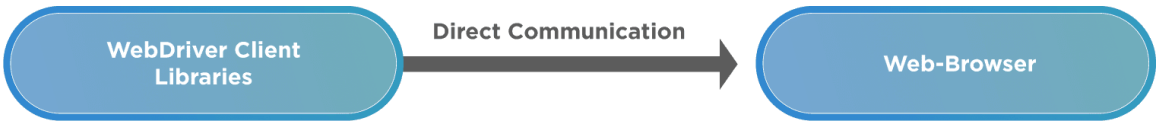


Рисунок 2.3. Принцип взаємодії Selenium з браузером.

• Локація веб–елементів – для виконання таких дій, як клацання, drag & drop, та інші необхідно спочатку визначити, з яким веб-елементом (кнопка, прапорець, випадаюче меню, текстове поле) потрібно взаємодіяти. Щоб спростити цей процес, WebDriver реалізує методи ідентифікації веб-елементів за допомогою різних атрибутів HTML, таких як ID, ім'я, клас, CSS, ім'я тегу, XPath, текст посилання

тощо.

• Обробка динамічних веб–елементів – Іноді на веб-сторінках зустрічаються динамічні елементи, які змінюються при кожному оновленні сторінки і це ускладнює їх ідентифікацію через зміну атрибутів HTML. Selenium має багато методів для вирішення цієї проблеми. Наприклад, можна використовувати абсолютний XPath, який містить повний XML-шлях до елемента, або функції Contains(), які дозволяють шукати елементи за частковим або повним текстом атрибутів і використовувати для роботи з динамічними елементами. Також є функція StartsWith(), яка дозволяє шукати елементи за початковим текстом атрибутів.

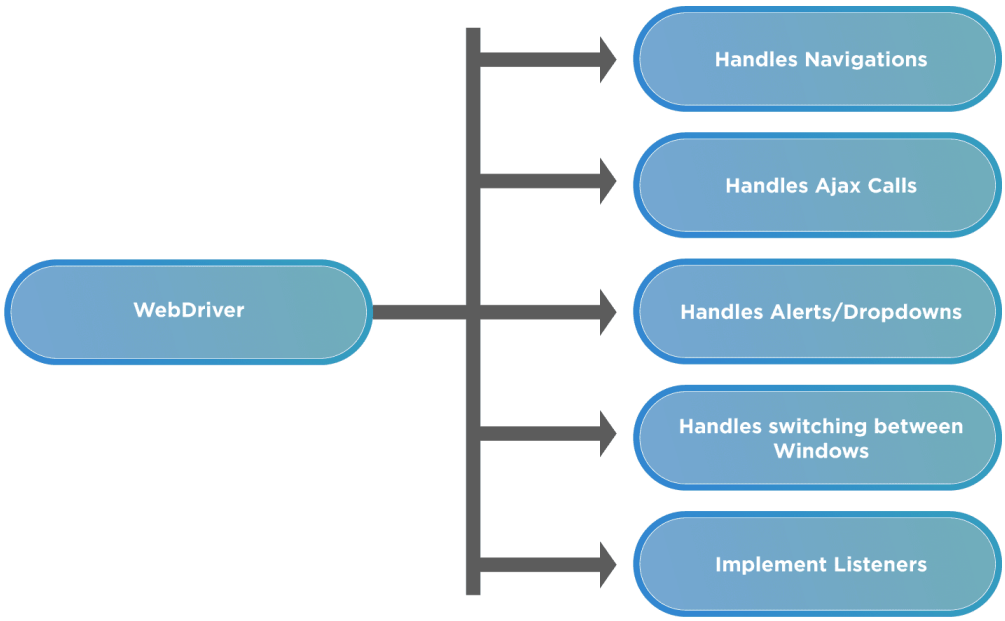


Рисунок 2.4. Основні функції WebDriver

• Обробка очікування елементів – WebDriver надає різноманітні механізми очікування, які дозволяють призупиняти виконання скрипту на необхідний час, залежно від умов, що можуть виникнути при обробці різних веб-сторінок. Не всі сторінки мають однакову структуру, і деякі з них можуть містити значну кількість даних, викликів AJAX або високонавантажених елементів, які потребують більше часу для завантаження.

Після аналізування загальної системи компонентів ми прийшли до висновку,

що Selenium WebDriver не може бути розглянутий як самостійний інструмент тестування. Він складається з різних компонентів, які є важливою складовою архітектури Selenium, необхідною для запуску тестів.

JSON WIRE PROTOCOL – Згідно з архітектурою Selenium JSON Wire Protocol відповідає за комунікацію між браузером та кодом в рамках Selenium. Це ядро функціоналу Selenium. Для передачі даних між браузером та кодом використовується провідний протокол JSON, який забезпечує RESTful API (передача стану у вигляді представлення ресурсів) через HTTP за допомогою JSON.

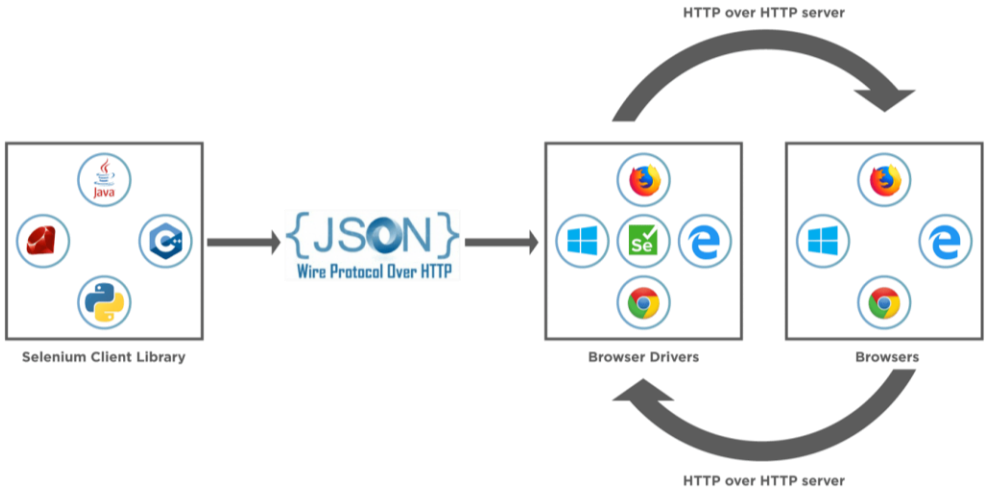


Рисунок 2.5. Модель архітектури Selenium

Оскільки Selenium підтримує різні браузери, для кожного з них створена власна реалізація стандарту W3C. Протокол JSONWire встановлює зв'язок між бінарними файлами браузера та клієнтськими бібліотеками.

Для тестування в браузерах за допомогою Selenium необхідно встановити браузер на локальній машині або на серверних машинах.

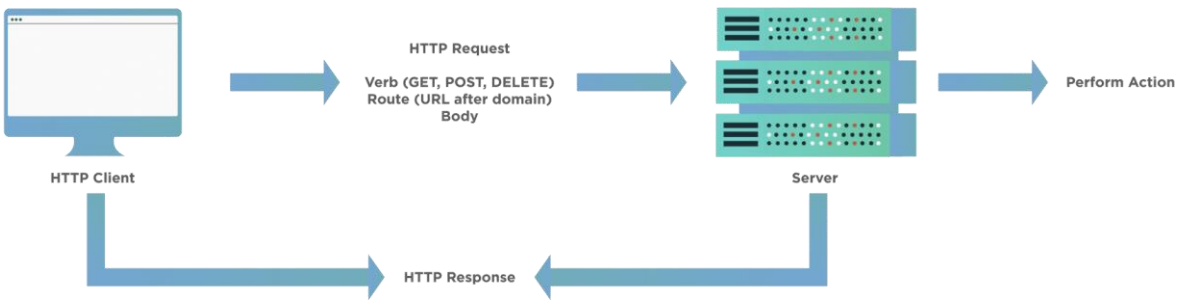


Рисунок 2.6. Архітектура внутрішнього сервера Selenium.

При виконанні коду WebDriver в Selenium автоматично відбувається наступний процес :

• Створюється запит HTTP, який надсилається до відповідного драйвера браузера (Chrome, IE, Firefox). Кожна команда Selenium має свій власний запит.

• Драйвер браузера отримує запит через HTTP-сервер.

• HTTP-сервер визначає необхідні дії/інструкції для виконання в браузері.

• Браузер виконує дії/кроки.

• HTTP-сервер отримує статус виконання, а потім повертає статус до сценарію автоматизації, який відображає результат (позитивний, виняток або помилку виконання).

2.1.3 TestNG

TestNG є потужним фреймворком для тестування, що дозволяє розробляти та виконувати автоматизовані тести на мові програмування Java. Він надає розширені можливості для тестування, такі як підтримка анотацій, групування тестів, залежності між тестами та багато іншого.

TestNG є відкритим програмним забезпеченням, що було розроблено з метою полегшення процесу тестування програмного забезпечення на мові програмування Java.

Встановлення та конфігурація

Перед початком роботи потрібно встановити його у своє середовище розробки. Для цього можна скористатися менеджером залежностей Maven або Gradle, або ж завантажити з офіційного сайту та додати його до проекту.

Цей фреймворк має дуже просту та зрозумілу структуру, що дозволяє швидко і легко виконувати тести. Після встановлення TestNG потрібно налаштувати конфігурацію проекту для його використання. Для цього можна використовувати файл конфігурації testng.xml, де вказувати каталоги з тестами, налаштування запуску тестів та інші параметри.

Створення тестів

TestNG дозволяє створювати тести за допомогою анотацій. Наприклад, анотація @Test вказує, що метод є тестом. Також є підтримка анотації для налаштування додаткових параметрів тестування, таких як час очікування, залежності між тестами та інші. Крім того, фреймворк дозволяє групувати тести та виконувати їх у певному порядку.

Наприклад, для групування використовується анотації `@Test(groups = {"group1"})`, можна групувати тести в групу з назвою `group1`. Для їх виконання у певному порядку, можна використовувати анотацію `@Test(dependsOnMethods = {"testMethod1"})`, де `testMethod1` вказує на метод, виконання якого потрібно завершити перед виконанням поточного тесту.

```java
import org.testng.Assert;
import org.testng.annotations.Test;

public class MyTestNGTest {
  @Test
  public void test() {
    String message = "Hello World!";
    Assert.assertEquals(message, "Hello World!");
  }
}
```

Рисунок 2.7. Виконання простого тесту з TestNG.

У цьому прикладі створено тестовий метод test, який перевіряє, чи дорівнює рядок message рядку "Hello World!". Метод Assert.assertEquals() перевіряє, чи два значення однакові. Якщо значення не співпадають, то тест не пройде.

Запуск тестів

Фреймворк надає багато можливостей для запуску тестів, таких як запуск тестів з командного рядка, запуск тестів через IDE, запуск тестів на віддаленому

сервері та багато іншого. Запуск тестів з TestNG дозволяє отримати детальну інформацію про їх виконання, таку як час виконання, кількість успішних тестів, кількість помилок та інші. Іншою корисною функцією є те що, TestNG дозволяє визначати приорітети тестів та налаштовувати їх запуск відповідно.

```
import org.testng.Assert;
import org.testng.annotations.DataProvider;
import org.testng.annotations.Test;

public class MyTestNGTest {
  @DataProvider(name = "testData")
  public Object[][] createData() {
    return new Object[][] {
      { 1, 1, 2 },
      { 2, 2, 4 },
      { 3, 3, 6 },
      { 4, 4, 8 },
      { 5, 5, 10 }
    };
  }

  @Test(dataProvider = "testData")
  public void test(int a, int b, int expected) {
    int result = a * b;
    Assert.assertEquals(result, expected);
  }
}
```

Рисунок 2.8. Виконання тестів з параметрами

У цьому прикладі використано анотацію `@DataProvider`, щоб створити дані для тестів. Метод `createData()` повертає двовимірний масив, де кожен рядок містить три значення: `a`, `b` та `expected`. Також використовуємо анотацію `@Test` з параметром `dataProvider`, щоб вказати TestNG, що цей тест повинен бути запущений з різними параметрами, які визначили в `createData()`.

Також є можливіть використовувати різні методи запуску тестів, такі як

паралельний запуск тестів, запуск тестів у певній послідовності та інші. При такому запуску TestNG використовує механізм потоків для запуску тестів у різних потоках, що дозволяє зменшити час виконання тестів та збільшити продуктивність

```
import org.testng.annotations.Test;

public class MyTestNGTest {
  @Test(priority = 1)
  public void test1() {
    System.out.println("Test 1");
  }

  @Test(priority = 2)
  public void test2() {
    System.out.println("Test 2");
  }

  @Test(priority = 3)
  public void test3() {
    System.out.println("Test 3");
  }
}
```

Рисунок 2.9. Виконання тестів з різними пріорітетами.

У цьому прикладі ми використовуємо параметр priority у кожному тесті, щоб вказати, який тест має бути виконаний першим, другим та т.д. У цьому випадку виконається тест1, тест2 та тест3 в такому порядку.

Переваги використання

TestNG є одним з найпопулярніших фреймворків для тестування програмного забезпечення на мові програмування Java. Можна визначити багато переваг, серед яких можна виділити:

- Підтримка анотацій та групування тестів.
- Можливість виконувати тести у певному порядку.
- Підтримка залежностей між тестами.
- Велика кількість можливостей для запуску тестів.

- Детальна інформація про виконання тестів.

TestNG є потужним фреймворком для тестування програмного забезпечення на мові програмування Java. Його використання допомагає забезпечити успішне випуск програмного забезпечення та задовольнення потреб користувачів. TestNG має багато переваг, включаючи підтримку анотацій, групування тестів, залежності між тестами та багато іншого. Цей інструмент є дуже популярним серед тестувальників та використовується для тестування різних типів програмного забезпечення.

Загалом, TestNG є потужним фреймворком для тестування програмного забезпечення на мові програмування Java. Він дозволяє розробляти та виконувати автоматизовані тести, що підвищує якість програмного забезпечення та знижує кількість помилок в продукції. Він має багато переваг, серед яких можна виділити підтримку анотацій та групування тестів, можливість виконувати тести у певному порядку, підтримку залежностей між тестами, велику кількість можливостей для запуску тестів та детальну інформацію про виконання тестів.

2.1.4 Протоколи взаємодії HTTP\HTTPS

HTTP - це протокол рівня додатку, який забезпечує комунікацію між клієнтом та веб-сервером в Інтернеті. Він є основним протоколом для передачі даних у Всесвітній павутині та дозволяє клієнтам, зазвичай, веб-браузерам, запитувати веб-сторінки з веб-серверів. HTTP забезпечує простий спосіб передачі повідомлень та даних між клієнтом та сервером, що робить його широко використовуваним протоколом в Інтернеті.

Коли користувач хоче отримати доступ до веб-сторінки, браузер надсилає повідомлення з запитом HTTP на веб-сервер. Потім сервер відповідає запитаною веб-сторінкою відповідно до запиту. Це може бути статична веб-сторінка або динамічний контент, залежно від запиту клієнта.

Веб-сервери є важливими компонентами в Інтернеті, які надають доступ до веб-сторінок та інших ресурсів. Вони обслуговують запити від клієнтів, які можуть бути розташовані в будь-якій точці світу. При цьому, веб-сервери за замовчуванням

використовують порт TCP 80 для обробки HTTP-запитів, що дозволяє їм працювати з браузерами та іншими клієнтами.

Проте, залежно від потреб користувача, веб-сервер може використовувати інші порти, що може зробити його недоступним для певних користувачів. Тому, щоб забезпечити доступність веб-сайту для всіх користувачів та зменшити навантаження на веб-сервер, важливо ретельно вибирати порти.

Ще одним важливим аспектом взаємодії між клієнтами та веб-серверами є безпека передачі даних. Для цього може бути використаний протокол HTTPS, який використовує шифрування для захисту даних під час передачі. Це особливо важливо, коли клієнти відправляють конфіденційну інформацію, таку як паролі чи реквізити банківських карток.

Клієнти та веб-сервери взаємодіють між собою за допомогою методів відповіді на запит. Клієнти надсилають HTTP-запити, зазвичай методами GET або POST, наприклад GET /homepage.html. Веб-сервери відповідають HTTP-відповідями, які включають повідомлення про стан, наприклад 200, якщо запит був успішним, та запитаний ресурс.

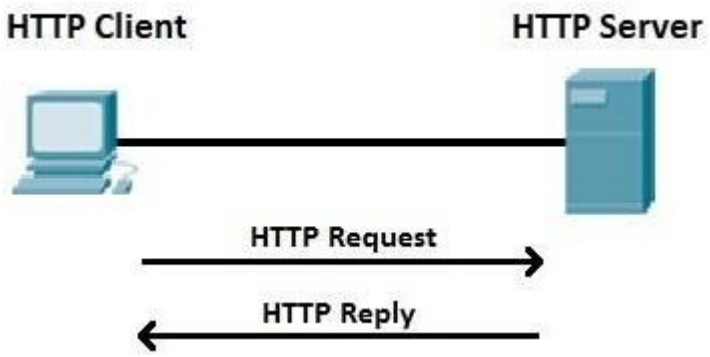


Рисунок 2.10. Схема роботи протоколу HTTP.

Клієнт хоче отримати доступ до http://google.com і вводить URL-адресу в свій браузер для надсилання запиту HTTP на веб-сервер, на якому розміщений цей веб-сайт. Веб-сервер оброблює запит і повертає вміст веб-сторінки відповіддю HTTP. (Це приклад запиту HTTP).

Зазвичай веб-сервери використовують порт TCP 80. Якщо порт не вказаний

у URL-адресі, браузер автоматично використовує порт 80 для надсилання HTTP-запиту. Наприклад, результат запиту http://google.com та [http://google.com:80](http://google.com/) буде однаковий.

На сьогодні найбільш поширеною версією HTTP є HTTP / 1.1, але більшість сучасних браузерів підтримує новішу версію, HTTP / 2.

HTTPS (Hyper Text Transfer Protocol Secure) - це захищена версія HTTP, яка забезпечує захищене з'єднання між клієнтом, таким як веб-браузер, та сервером, таким як веб-сервер, за допомогою шифрування. Для шифрування HTTPS використовує протокол TLS (Transport Layer Security) або попередній рівень Secure Sockets Layer (SSL). Цей протокол дозволяє забезпечити конфіденційність та надійність передачі даних між клієнтом та сервером, що робить з'єднання безпечним.

Основною метою HTTPS є захист конфіденційної інформації, передаваної через Інтернет, від несанкціонованого доступу та зловживань. Це особливо важливо для фінансових операцій, онлайн-покупок, передачі конфіденційної інформації про користувачів та іншої конфіденційної інформації. HTTPS забезпечує захист від різноманітних видів атак, таких як перехоплення інформації, підробка даних та внесення змін в передані дані.

URL-адреси HTTPS починаються з https замість http. Це дає можливість відрізнити захищений веб-сайт від незахищеного. У адресному рядку браузера можна побачити блокування, що свідчить про те, що веб-сайт використовує HTTPS. Крім того, HTTPS використовує добре відомий порт TCP 443, що робить його відмінним від HTTP, який використовує порт TCP 80.

Усі ці функції роблять HTTPS надійним та безпечним для передачі конфіденційної інформації через Інтернет.

HTTPS зазвичай використовується для створення захищеного каналу через якусь небезпечну мережу, наприклад Інтернет. Багато трафіку в Інтернеті є незашифрованим і легкодоступним для хакерських атак. HTTPS шифрує конфіденційну інформацію, що робить з'єднання безпечним.

Адреси URL HTTPS починаються з https замість http. Ви можете відразу

визнати, що веб-сайт використовує HTTPS, оскільки праворуч від адресного рядка з'являється блокування, що вказує на захищений з'єднання. HTTPS забезпечує захист конфіденційної інформації, передаваної через Інтернет, від несанкціонованого доступу та зловживань. Для шифрування HTTPS використовує протокол TLS або SSL, що дозволяє забезпечити надійність передачі даних між клієнтом та сервером. Усі ці функції роблять HTTPS надійним та безпечним для передачі конфіденційної інформації через Інтернет.

2.1.5 Формат даних JSON

JSON або JavaScript Object Notation - це формат для структурування даних, який є альтернативою XML. В основному він використовується для передачі даних між веб-додатком та сервером. Squarespace використовує JSON для зберігання та організації вмісту сайту, що створено за допомогою CMS. Цей формат дозволяє зберігати дані в парах ключ / значення, де ключ завжди є рядком, а значення може бути рядком, числом, логічним виразом, масивом або об'єктом.

JSON складається з пар ключ / значення. Ключ - це рядок, що завжди укладається у лапки, а значення може бути рядком, числом, логічним виразом, масивом або об'єктом. Пара ключ / значення розділяється двокрапкою, а пари розділяються комами. Наприклад, "foo": "bar" - це пара ключ / значення, де ключ - "foo", а значення - "bar". JSON підтримує такі типи даних, як масиви, логічні значення, числа, об'єкти та рядки.

- масиви: Асоціативний масив значень;
- логічні значення: Істинно чи хибно;
- числа: цілі та числа з плаваючим знаком розміром до 64біт;
- об’єкти: Асоціативний масив пар ключ / значення;
- рядки: Кілька символів простого тексту, які зазвичай утворюють слово;

Один з прикладів використання JSON у створенні авто-тестів може бути пов'язаний із використанням Selenium WebDriver для автоматизованого тестування веб-додатків. У цьому випадку, тестувальник може використовувати JSON-файл для збереження тестових даних та параметрів, які використовуються в тестах.

Наприклад, розглянемо тестування веб-сторінки, яка містить форму для відправлення повідомлення. Тестувальник може використовувати JSON-файл для збереження даних, необхідних для відправлення повідомлення, таких як адреса електронної пошти, тема повідомлення та текст повідомлення.

У тесті, тестувальник може використовувати наступний код на мові Python для зчитування даних з JSON-файлу та вводу їх на форму веб-сторінки за допомогою Selenium WebDriver:

```
import json
from selenium import webdriver
from selenium.webdriver.common.keys import Keys

# Зчитування даних з JSON-файлу
with open('message_data.json') as json_file:
    data = json.load(json_file)

# Відкриття веб-сторінки
driver = webdriver.Chrome()
driver.get("<https://example.com/contact>")

# Введення даних на форму
driver.find_element_by_name("email").send_keys(data["email"])
driver.find_element_by_name("subject").send_keys(data["subject"])
driver.find_element_by_name("message").send_keys(data["message"])
driver.find_element_by_name("submit").click()

# Перевірка успішного відправлення повідомлення
assert "Your message has been sent" in driver.page_source

# Закриття веб-сторінки
driver.close()
```

Рисунок 2.11. Зчитування даних з повідомлення

У цьому прикладі, тестувальник зчитує дані про повідомлення з JSON-файлу та використовує їх для вводу даних на форму веб-сторінки за допомогою методів Selenium WebDriver. Після відправлення повідомлення, тестувальник перевіряє, чи відображається підтвердження відправлення повідомлення на сторінці.

Таким чином, використання JSON з Selenium WebDriver у створенні авто-тестів дозволяє тестувальникам зберігати тестові дані та параметри в

структурованому форматі, що спрощує процес написання авто-тестів і дозволяє використовувати їх для різних наборів даних та параметрів.

Ще один приклад використання JSON у створенні авто-тестів полягає у зберіганні тестових даних. Наприклад, при написанні авто-тестів для веб-додатку, можна зберігати дані про користувачів, які використовують додаток, у форматі JSON. Це дозволяє створювати тести з різними наборами даних, щоб перевірити різні варіанти використання додатку.

Ось приклад коду на мові Java, який демонструє використання JSON для зберігання тестових даних:

```
import org.json.simple.JSONObject;
import org.json.simple.parser.JSONParser;
import org.json.simple.parser.ParseException;

public class User {
    private String name;
    private String email;
    private String password;

    // Конструктор для створення об'єкту користувача з JSON
    public User(String json) throws ParseException {
        JSONParser parser = new JSONParser();
        JSONObject obj = (JSONObject) parser.parse(json);
        this.name = (String) obj.get("name");
        this.email = (String) obj.get("email");
        this.password = (String) obj.get("password");
    }

    // Геттери та сеттери для полів
    public String getName() { return this.name; }
    public String getEmail() { return this.email; }
    public String getPassword() { return this.password; }
    public void setName(String name) { this.name = name; }
    public void setEmail(String email) { this.email = email; }
    public void setPassword(String password) { this.password = password; }
}

public class Test {
    public static void main(String[] args) throws ParseException {
        String json = "{\\"name\\":\\"John Smith\\",\\"email\\":\\"john.smith@example.co
m\\",\\"password\\":\\"password123\\"}";
        User user = new User(json);
```

Рисунок 2.12 Зберігання тестових даних

У цьому прикладі JSON використовується для зберігання тестових даних користувача, якими потім можна скористатися при написанні авто-тестів. Клас `User` слугує для створення об'єкту користувача з JSON, який можна використовувати у тестах. Клас `Test` демонструє використання об'єкту користувача у тесті.

Таким чином, JSON дозволяє зберігати тестові дані в структурованому форматі, що спрощує процес написання авто-тестів та дозволяє використовувати їх для різних наборів даних.

2.2 Загальний алгоритм реалізації проекту

Оскільки проект передбачає створення навчального модулю з автоматизованого тестування, то будь-яка схожа система є програмним продуктом, розробка якого включає наступні етапи:

- Формулювання завдань проекту,
- Визначення вимог до їх реалізації,
- Аналіз та розробка архітектури проекту,
- Написання коду,
- Тестування та відлагодження,
- Документування,
- Впровадження та підтримка проекту у майбутньому.

У першому розділі проекту були визначені завдання та цілі, які включали в себе створення навчального модулю з автоматизованим тестуванням програмного забезпечення. Основними вимогами було створення навчального модулю простим та зрозумілим інтерфейсом, що відповідає вимогам обраної галузі. Вони були узагальнені в технічному завданні та описі основних функцій додатка. Для аналізу та розробки архітектури були використані існуючі аналоги продукту, встановлені вимоги та планування реалізації програмного продукту. Підготовка матеріалів була здійснена для створення продукту, обрано середовище створення та інструменти, мову програмування, планування етапів розробки та тестування.

Організація процесу включає в себе встановлення задач та перехід до написання коду, що є найдовшою та головною частиною проекту. Тестування проводиться на етапі кодування проекту та після. Відбувається аналіз відповідності реалізованої задачі встановленим вимогам. Документація програмного забезпечення включає в себе супровідні документи з описом деталей, необхідних для використання продукту, такі як відомість власників оригіналів, текст програми, опис програми, програма і методика випробувань, технічне завдання, пояснювальна записка.

Після проходження всіх етапів тестування та відповідності встановленим вимогам, відбувається реліз комп'ютерної гри, а також подальший супровід та підтримка програми. Також важливим етапом алгоритму створення навчальних модулів є підготовка правильно підібраних завдань зі збільшенням рівня складності по мірі проходження модулю.

2.3 Функціонал продукту

Продукт розробки являє собою навчальний модуль з автоматизовано тестування. Основна його мета – надати можливість для вивчення теорії та практики базових елементів та технологій автоматизованого тестування шляхом проходження тестів та інтерактивних завдань.

Отже основні вимоги до продукту є такими:

- Основна мета модулю – вивчення на практиці основ автоматизованого тестування через виконання завдань із написанням програмного коду та проходженням тестів для перевірки та закріплення теоретичних знань;
- Практика у вирішенні задач із використанням сучасних технологій автоматизації;
- Посібні матеріали у налаштуванні та встановленні технологій для автоматизації.

В першу чергу реалізуются саме ці функції продукту, так як є основними реалізаціями критеріїв створення навчального модулю.

3. ПРОГРАМНА РЕАЛІЗАЦІЯ НАВЧАЛЬНОГО МОДУЛЮ

3.1 Оцінка та планування розробки навчального модулю з автоматизованого тестування

У процесі розробки продукту, після визначення основних вимог та архітектури системи, необхідно було скласти список задач для реалізації Мінімальної життєздатної версії (MVP). Важливим фактором при розробці MVP є обмежений час, який потрібно використовувати максимально ефективно та задовольнити найбільшу частину вимог продукту.

Тому, для забезпечення ефективної та успішної розробки MVP, було вирішено провести оцінку розробки з акцентом на принцип "головні функції спочатку" (KFF - Key Features First), який є одним з ключових положень підходу до розробки MVP. Використання техніки Experience Based для оцінки часу на розробку, що базується на минулому досвіді розробника та його рівні знань, також допомагає забезпечити більш точну та реалістичну оцінку розробки.

	Функціональні вимоги
1	Виконання завдань в редакторі коду. Модуль має дозволяти користувачам виконувати завдання в редакторі коду.
2	Проходження тестових завдань. Модуль має дозволяти користувачу проходити тестові завдання із вибором правильних відповідей.
3	Перегляд результатів тестування. Модуль має дозволяти користувачам переглядати результати своїх тестів, включаючи оцінки та відповіді на питання.
4	Доступ до додаткових ресурсів. Модуль має надавати користувачам доступ до додаткових ресурсів, таких як відео-уроки, статті, блоги та інше.

Рисунок 3.1. Формування функціональних вимог.

Кожна вимога була описана в стилі use-case, що дозволяє застосовувати підхід BDD (Behavioral Driven Development), орієнтований на розробку функціоналу з фокусом на сценаріях використання та поведінці системи. Такий опис дозволяє збільшити ефективність розробки та уникнути неправильного розуміння вимог.

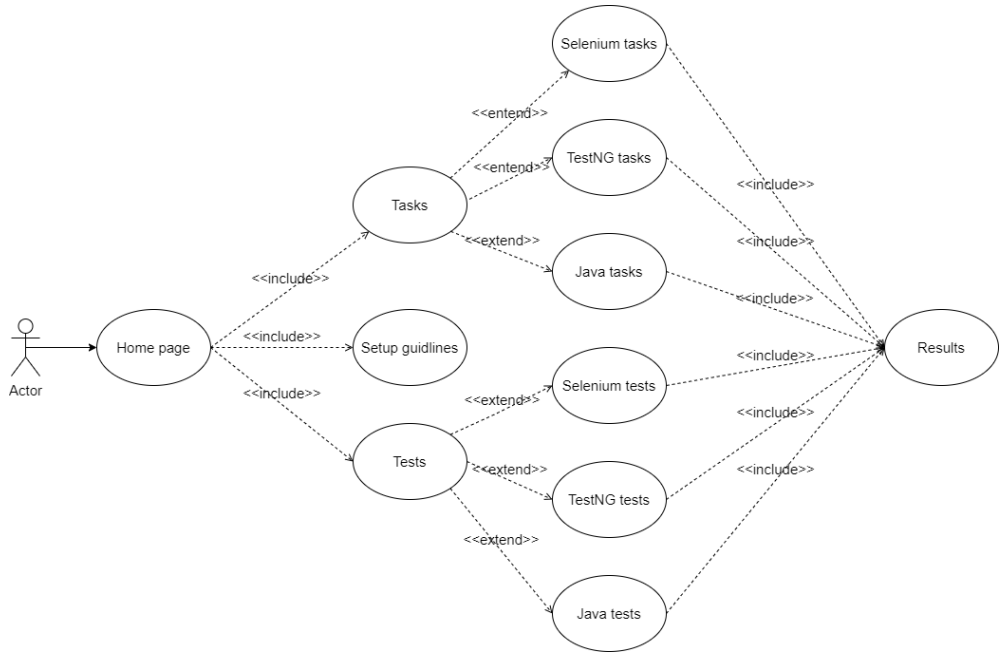


Рисунок 3.2. Use case діаграма.

Усі задачі були поміщені у систему відстежування задач Trello для кращої простежуваності їх виконання та труднощів, які можуть виникати з ними. Це дозволяє керувати процесом розробки, визначати ступінь виконання робіт та виключити прострочення термінів.

У цілому, використання технік KFF, Experience Based, ABC-аналізу, а також стилів use-case та BDD допомогли забезпечити ефективну розробку MVP, відповідно до вимог продукту та умов, які були на той момент. Такий підхід дозволяє забезпечити максимальну ефективність розробки, уникнути помилок та забезпечити високу якість продукту.

При безпосередньому переміщенні задач до спринту вони розбивалися на окремі технічні підзадачі, які оцінювалися та реалізовувалися поступово.

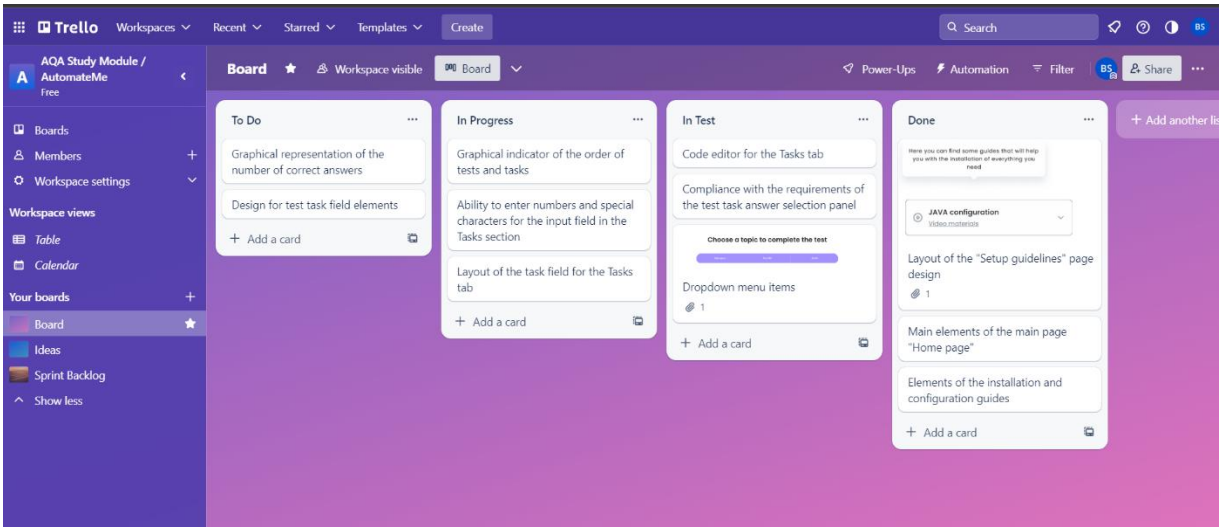


Рисунок 3.3. Використання Trello для відстеження стану розробки.

Як результат, збільшується ефективність розробки та уникнення неправильного розуміння вимог, коректного визначення ступеня виконання робіт.

У цілому, використання технік KFF, Experience Based, ABC-аналізу, а також стилів use-case та BDD допомогли забезпечити ефективну розробку MVP, відповідно до вимог продукту та умов, які були на той момент. Такий підхід дозволяє забезпечити максимальну ефективність розробки, уникнути помилок та забезпечити високу якість продукту.

Щоб забезпечити ще більшу ефективність розробки, було вирішено організувати розробку з використанням методології Agile. Вона дозволяє забезпечити більш гнучкий та адаптивний підхід до розробки, що дозволяє більш швидко відповідати на зміни в вимогах продукту та забезпечувати більш ефективне виконання задач.

3.2 Набір інструментів використаних для розробки навчального модулю

Створення навчального модулю для вивчення технологій автоматизованого тестування - це комплексна задача, яка потребує великої кількості роботи та підходів. Навчальний модуль має допомогти студентам зрозуміти базові принципи та засоби автоматизованого тестування веб-додатків та забезпечити їхню

підготовку до роботи в цій галузі. У цьому пункті описано кроки, які були виконані при створенні навчального модулю, включаючи розробку макету у Figma та розробку в середовищі IntelliJ IDEA.

Визначення мети та цілей навчання

Перший крок у створенні навчального модулю - визначення мети та цілей навчання. Метою навчального модулю було передати студентам теоретичні знання та практичні навички, необхідні для автоматизації тестування веб-додатків з використанням Selenium та TestNG. Цілі навчання встановлювалися з урахуванням вимог та потреб ринку праці. Визначення мети та цілей навчання дозволило зосередитися на головних завданнях та визначити стратегію навчання.

Розробка плану навчання

Другий крок - розробка плану навчання. План навчання складався тестів та завдань. При розробці матеріалів навчання було використано найновіші методики та підходи, що дозволило забезпечити максимальну якість та ефективність навчання. Розробка плану навчання дозволила структурувати матеріали та покращити організацію навчання.

Розробка змісту навчання

Третій крок - розробка змісту навчання. Зміст навчання було створено з урахуванням різноманітних методів та підходів навчання, що дозволить знизити поріг мінімальних знань для початку та сприяє ефективного вивченню матеріалу та збагачення знання у сфері автоматизованого тестування. Для демонстрації правильного використання Selenium та TestNG були розроблені приклади коду, які були включені до матеріалів навчання.

Розробка макету у Figma

Четвертий крок - розробка макету у Figma. Це дозволило створити візуальну концепцію модулю, побачити його у вигляді макету та внести необхідні зміни перед тим, як розробити фінальний продукт. Застосування Figma дозволило більш ефективно організовувати роботу над навчальним модулем та зробити його більш доступним та зрозумілим для студентів. Розробка макету у Figma дозволила покращити візуальну привабливість та зрозумілість навчального модулю.

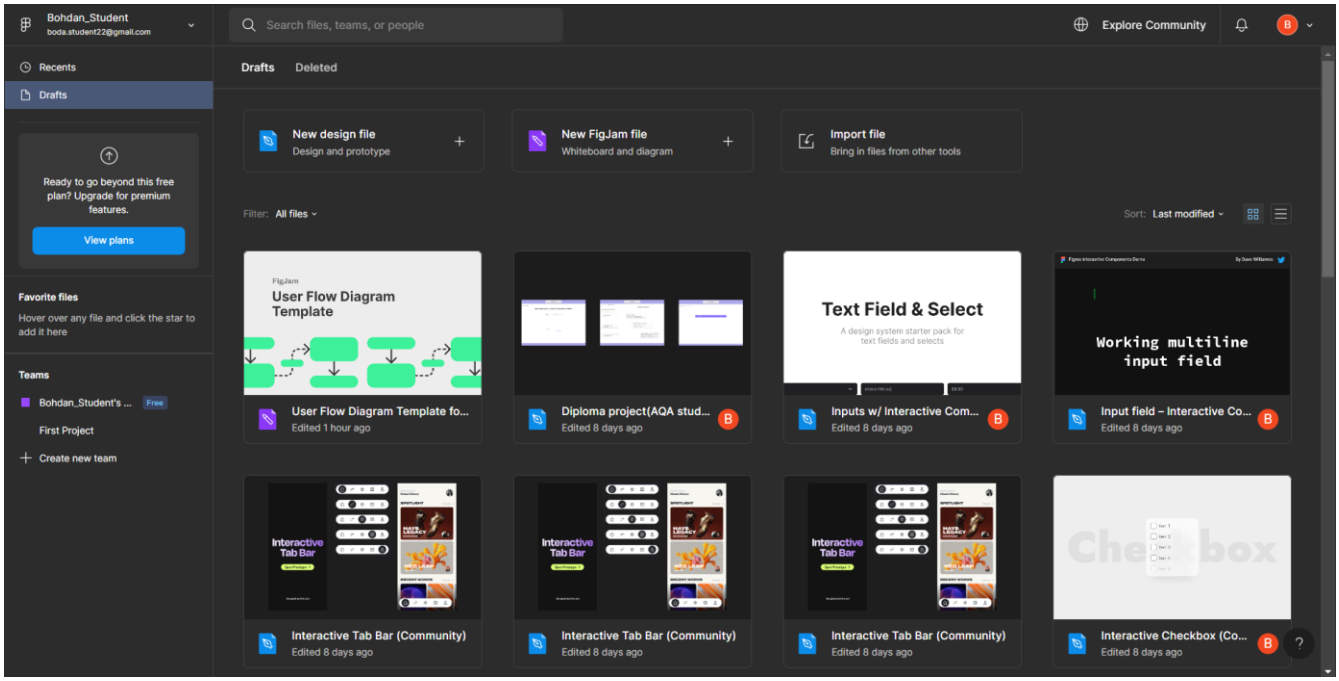


Рисунок 3.4. Макетування у Figma.

Проектування системи

П'ятий крок - проектування системи. Для створення навчального модулю ми використовували підхід user-centered design, що дозволив нам зосередитися на потребах та очікуваннях студентів та забезпечити їх задоволеність від користування продуктом.

Розролена діаграма User flow дозволила проаналізувати та покращити процес взаємодії користувачів з навчальним модулем, забезпечивши максимальну зручність та ефективність використання. В результаті, користувачі мають можливість легко переходити між компонентами навчального модулю та здійснювати повторні перегляди та вправи. Діаграма є важливою складовою в розробці будь-якого продукту, оскільки вона дозволяє оцінити ефективність взаємодії з користувачами та зробити відповідні покращення, що забезпечують кращу якість взаємодії з продуктом. Крім то, це дозволило ідентифікувати можливі проблеми з користуванням та вирішити їх ще на етапі проектування модулю.

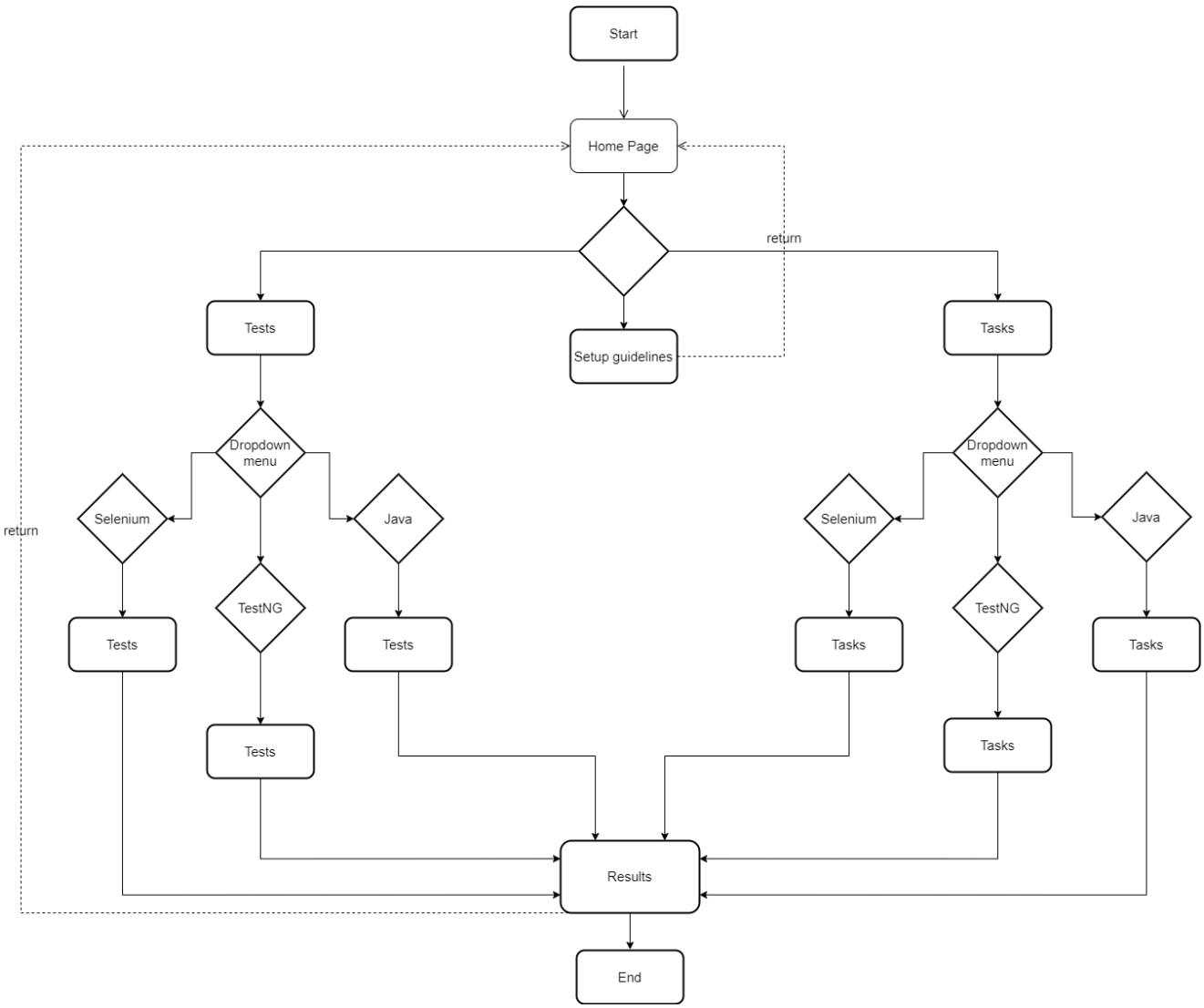


Рисунок 3.5. Діаграма «User flow».

Розробка в середовищі IntelliJ IDEA

Шостий крок - розробка в середовищі IntelliJ IDEA. Для цього були використані Java, бібліотека Selenium та фреймворк TestNG. Ця мова програмування є однією з найпоширеніших у світі, яка використовується для розробки програмного забезпечення, включаючи інструменти автоматизованого тестування. Використання Java для створення навчального модулю з вивчення автоматизованого тестування має кілька переваг.

По-перше, Java є мовою програмування, яка має високу стабільність та надійність, що важливо для розробки програмного забезпечення для автоматизованого тестування. Вона також дозволяє розробникам швидко створювати програмне забезпечення, що сприяє швидкому створенню навчального

модулю

По-друге, Java має широку спільноту розробників та підтримується багатьма відкритими фреймворками для тестування, такими як JUnit, TestNG та Selenium, що дозволяє розробникам створювати ефективні тести для програмного забезпечення.

По-третє, Java є платформонезалежною мовою програмування, що означає, що навчальний модуль може працювати на будь-якій операційній системі, що підтримує цю мову.

Було розроблено практичні завдання та тести, які допоможуть студентам закріпити отримані знання та навички. Крім того, було проведено тестування навчального модулю з метою виявлення можливих недоліків та помилок, які було виправлено перед публікацією. Розробка в середовищі IntelliJ IDEA дозволила побачити, як на практиці використовуються Selenium та TestNG та допомогла підготуватися до роботи з цими технологіями.

Загалом, виконання цих дій допомогло створити якісний навчальний модуль, який дозволить студентам вивчити технології автоматизованого тестування Selenium та TestNG та підвищити їхні навички в автоматизованому тестуванні. Отримані знання можуть допомогти студентам знайти роботу у сфері тестування програмного забезпечення та покращити свої кар'єрні можливості. Крім того, навчальний модуль буде корисним не тільки для студентів, але й для фахівців, які хочуть покращити свої знання у цій галузі та залишатися в тренді.

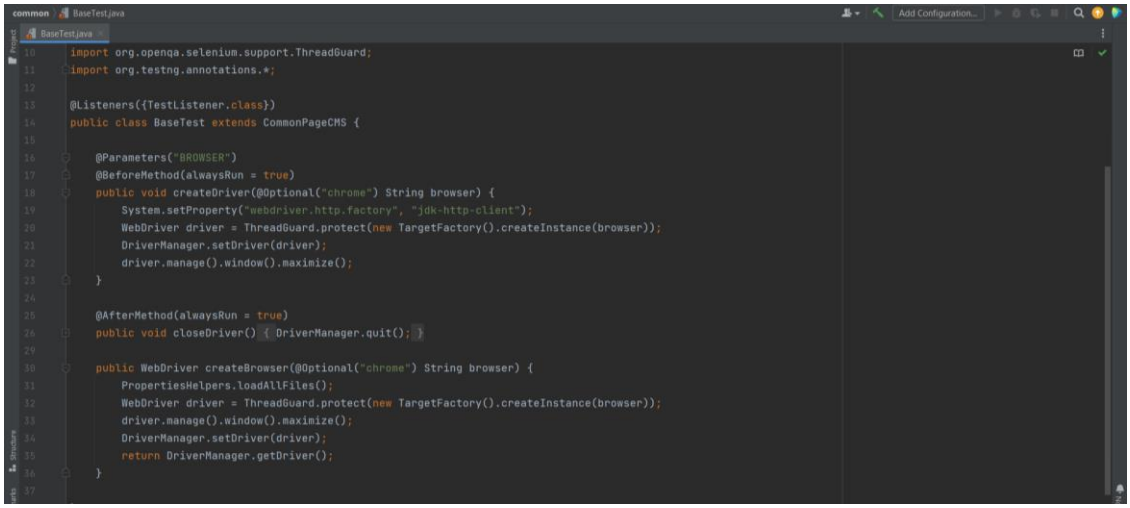


Рисунок 3.6. Процес розробки в середовищі IntelliJ IDEA

3.3 Функціонал навчального модулю.

Основним функціоналом модулю є наступні елементи:

- Тестові завдання;
- Інтерактивний редактор коду;
- Модулі із перевірки та підрахунків правильних та неправильних відповідей;
- Сторінка із практичними інструкціями та матеріалами із встановленням та налаштуванням технологій.

Для того, щоб детальніше ознайомитися із системою, потрібно пройти повний шлях користування нею. При відкритті основної сторінки, відображаються основні елементи модулю. Завдяки ним здійстьється уся навігація та основні інтеракції. Цей клас має у собі дочірні системи, завдяки яким можна обрати технологію, яку користувач хоче вивчати, та саме виконання тестових, чи практичних завдань.

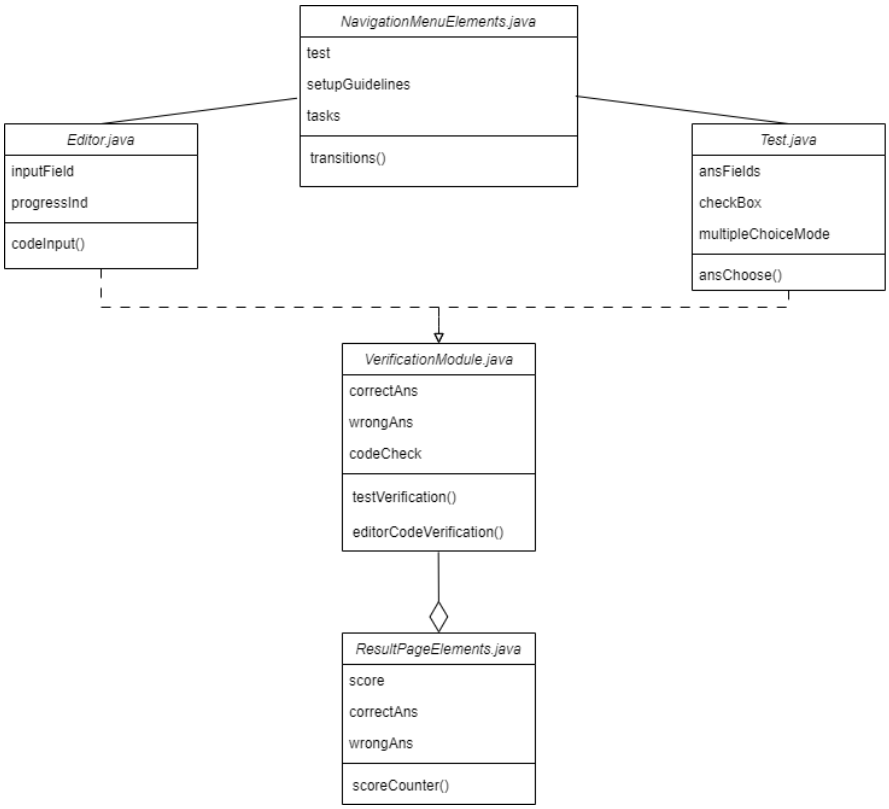


Рисунок 3.7. Діаграма основних класів.

Далі, після визначення із родом технологій та типом завдань, почнеться уже сам процес вивчення. При виконанні практичних завдань, користувач буде взаємодіяти із редактором коду, самим завданням та іншими графічними елементами модулю.

Суть вправи полягає в тому, що у пропуски у програмному коді потрібно додати відсутні частини за допомогою текстового поля нижче, що виступає редактором-компілятором. Основний елемент - велике текстове поле для введення коду. Там можна ввести будь-який текст, який буде відображатися у вікні. Іншою зручною можливістю є автоматичного перенесення рядків, а також можливість встановлювати табуляцію. Поле має можливість редагування тексту з клавіатури та миші, наприклад, вставляти текст з буферу обміну, видаляти, вирізати, копіювати, виділяти текст, переміщувати курсор тощо.

```
// Set up the code input area
codeInput = new JTextArea();
codeInput.setFont(new Font("Monospaced", Font.PLAIN, 12));
JScrollPane scrollPane = new JScrollPane(codeInput);
mainPanel.add(scrollPane, BorderLayout.CENTER);
// Set up the "submit" button to send the code to VerificationModule
JButton submitButton = new JButton("Submit");
submitButton.addActionListener(new ActionListener() {
   @Override
   public void actionPerformed(ActionEvent e) {
      // Get the code from the input area
      String code = codeInput.getText();
```

Цей код створює вікно з редактором коду, який містить поле вводу, поле виводу, кнопку "Submit" та показник прогресу. При натисканні на кнопку "Submit" код з поле вводу відправляється до класу VerificationModule за допомогою методу verifyCode(). Після відправки коду поле вводу очищається, а показник прогресу оновлюється.

Він являє собою універсальний програмний елемент, який при взаємодії з

модулем перевірки, передає введений код. Для збереження та передачі інформації від редактора до **ResultPageElements** через **VerircationModule**, використовується JSON.

```
"input_data": {
    "technology": "Java",
    "task_type": "practical"
},
"result": {
    "is_correct": true,
    "time": 1579600000,
    "message": "The task was successfully completed."
```

Рисунок 3.8. Передача інформації за допомогою JSON.

Цей код містить дві основні частини - input_data та result.

input_data містить інформацію про технологію та тип завдання, яке розробник виконує. У даному випадку, технологія - Java, а тип завдання - практичне. result містить результат виконання завдання, який буде відображений на **ResultPageElements**. В даному випадку, is_correct дорівнює true, що означає, що завдання було успішно виконане, а time вказує на час виконання в Unix часі. Крім того, у полі message можна додати додаткову інформацію про виконання завдання.

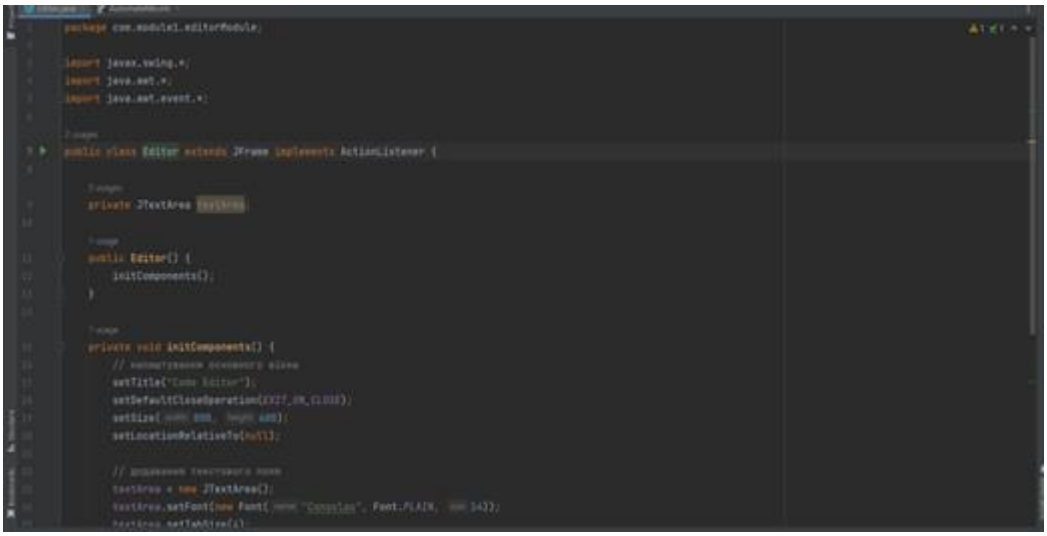


Рисунок 3.9. Програмні елементи редактора коду.

Цей JSON код допомагає зберегти та передати інформацію від редактора до **ResultPageElements** через **VerificationModule**. В процесі передачі, дані проходять

перевірку за допомогою технологій JSON, що дозволяє зберігати усю необхідну інформацію та передавати її безпечно та ефективно. Після того, як дані досягнуть **ResultPageElements**, вони будуть відображені на сторінці з результатами.

Наступним ключовим елементом є тестове поле, створене за допомогою HTML та Java, є інтерактивним елементом веб-сторінки, який дозволяє користувачам відповідати на запитання, обирати правильну відповідь.

Для створення тестового поля на чотири відповіді використано HTML-форму, яка складається з блоків коду, що містять текст запитання та варіанти відповідей. Крім того, використовується Java-скрипти для додавання функцій перевірки відповідей та відображення результатів.

Тестове поле має наступну структуру:

- Блок з текстом запитання
- Чотири блоки з варіантами відповідей
- Кнопка "Previous", яка дозволяє переглянути попереднє запитання
- Кнопка "Submit", яка запускає функцію перевірки вибраних відповідей
- Блок з результатом тестування

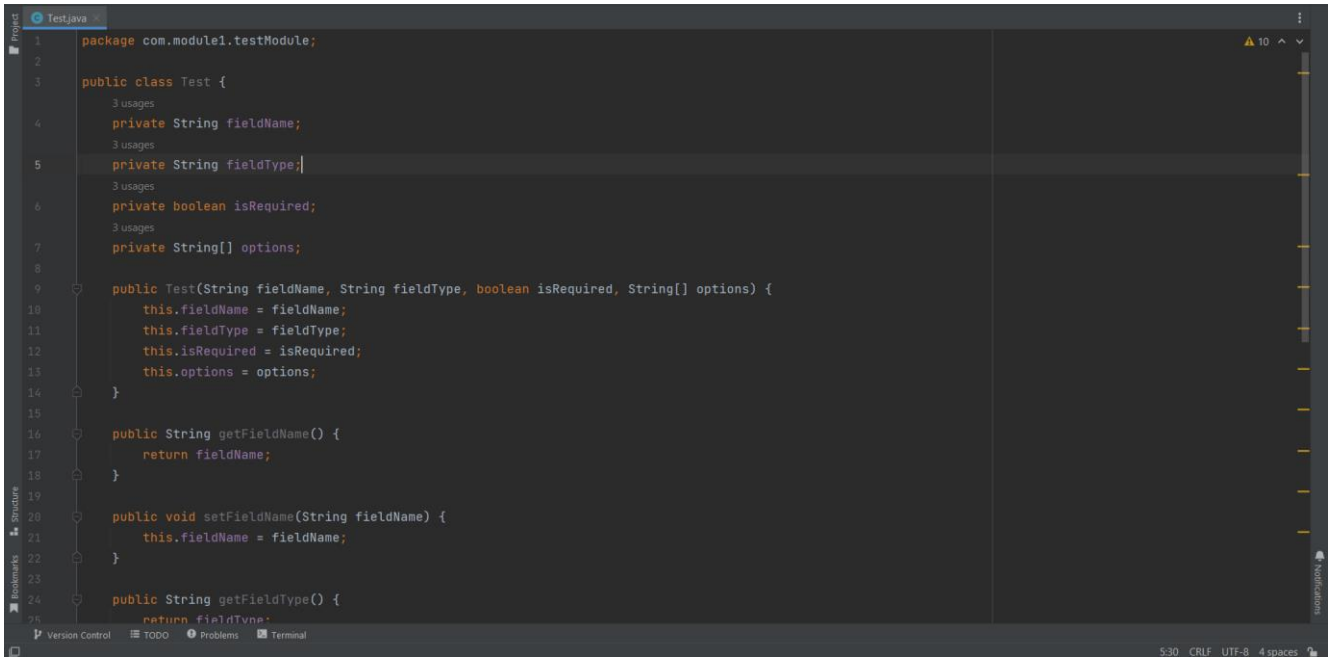


Рисунок 3.10. Програмні елементи тестового поля.

Після того, як користувач вибрав відповіді та натиснув на кнопку "Submit",

скрипт Java перевіряє правильність вибраних відповідей та відображає результат тестування у відповідному блоку.

Для збереження відповідей користувача та їх подальшої перевірки можна використовувати формат JSON, методи та класи згадані вище. Після того, як користувач вибрав відповіді та натиснув на кнопку "Submit", Java-скрипт генерує об'єкт JSON, який містить інформацію про вибрані відповіді.

Модуль з перевіркою використовує інформацію з JSON-об'єкту, щоб перевірити правильність вибраних відповідей та згенерувати результат тестування.

Таким чином, використання цього текстового формату дозволяє зберігати та переносити інформацію про відповіді користувачів між різними модулями. Крім того, цей формат дозволяє легко зберігати й обробляти великі об'єми даних, що може бути корисним для створення складних тестових завдань.

```
public class VerificationModule {
    public static boolean verifyAnswer(String question, String selectedAnswer) {
        JSONParser parser = new JSONParser();
        try {
            // Розбір JSON-об'єкту питання
            JSONObject questionObj = (JSONObject) parser.parse(question);
            // Отримання правильної відповіді з JSON-об'єкту
            String correctAns = (String) questionObj.get("answer");
```

Перевірка, чи відповідь на питання вірна.

```
            if (selectedAnswer.equals(correctAnswer)) {
                // Збереження результату у ResultPageElements класі
                ResultPageElements.setResult((long) questionObj.get("id"), true);
                return true;
            }
```

Код також зберігає результат у класі ResultPageElements.

```
public class ResultPageElements {
    private List<String> questions;
    private List<String> userAnswers;
```

```
    private List<String> correctAnswers;
    private int correctAnsCount;
    private int wrongAnsCount;
    private int score;
    public ResultPageElements(List<String> questions, List<String> userAnswers,
List<String> correctAnswers) {
        this.questions = questions;
        this.userAnswers = userAnswers;
        this.correctAnswers = correctAnswers;
        this.correctAnsCount = 0;
        this.wrongAnsCount = 0;
        this.score = 0;
```

Тепер, після того, як клас VerificationModule перевірить відповіді користувача на правильність, можна викликати метод scoreCounter() класу ResultPageElements для підрахунку загального балу і збереження його у полі score.

```
    public void scoreCounter() {
        for(int i = 0; i < correctAnswers.size(); i++) {
          if(userAnswers.get(i).equals(correctAnswers.get(i))) {
            correctAnsCount++;
            score++;
          } else {
            wrongAnsCount++;
```

Іншим важливим аспектом виступає сторінка, яка призначена для надання корисних посилань та практичних порад із посібними матеріалами для встановленням та налаштуванням Selenium та TestNG. Основна функціональна вимога сторінки полягає у забезпеченні користувачів необхідною інформацією для використання зазначених інструментів. Її важливість варто відмітити, тому що не у всіх розглянутих аналогах, були покрокові інструкції для встановлення цих технологій, навіть не говорячи про практичні елементи та деякі документація.

4. ПРИКЛАДИ ВИКОРИСТАННЯ ТА ТЕСТУВАННЯ МОДУЛЮ

4.1 Опис роботи модулю.

Розділ з прикладами використання навчального модулю з вивчення автоматизованого тестування містить в собі детальний опис та приклади, що демонструють використання Selenium та TestNG для автоматизації тестування. У цьому розділі будуть розглянуті тестові та інтерктивні завдання із написання авто тестів та зразки використання навчального модулю.

Робота з системою починається із основної сторніки. Після ознайомленням із інтерфейсом та підказками, користувач обирає відповідне меню.

Доступні наступні пункти:

- Tests;
- Setup guidelines;
- Tasks.

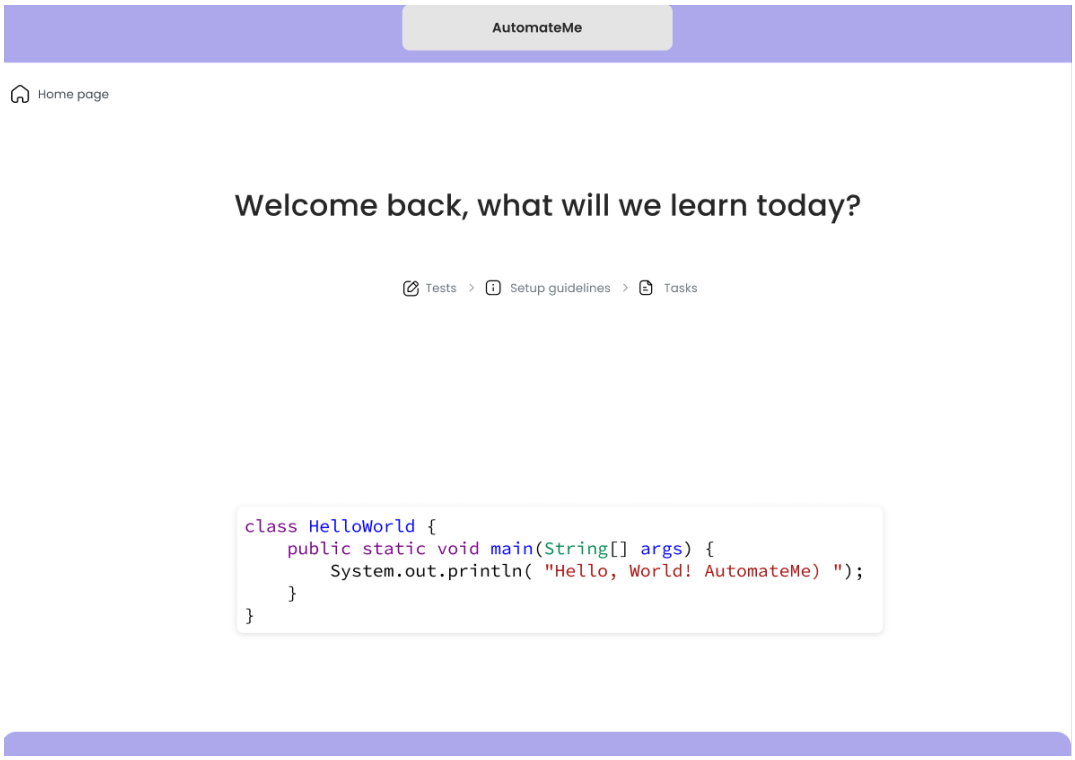


Рисунок 4.1. Головне меню.

При розробці інтерфейсу було обрано мінімалістичний стиль дизайну. Таке рішення дозволяє швидко вивчити інтерфейс та функціонал інструменту, зменшуючи поріг входу для нових користувачів. Мінімалістичний дизайн зробив інтерфейс зручним та простим у використанні.

Однією з переваг такого рішення покращення користувацького досвіду та полегшення навігації по модулю. Користувачі можуть зосередитися на основних функціях та завданнях, що дозволяє їм виконувати роботу швидше та ефективніше. Мінімалістичний дизайн також дозволяє перевести фокус на важливі елементи та зробити їх більш помітними, що допомагає забезпечити більшу концентрацію користувачів на головному завданні та зменшити кількість відволікаючих елементів.

Переходячи до меню тестів, надається можливість обрати з поміж доступних тестів та обрати технологію для перевірки.

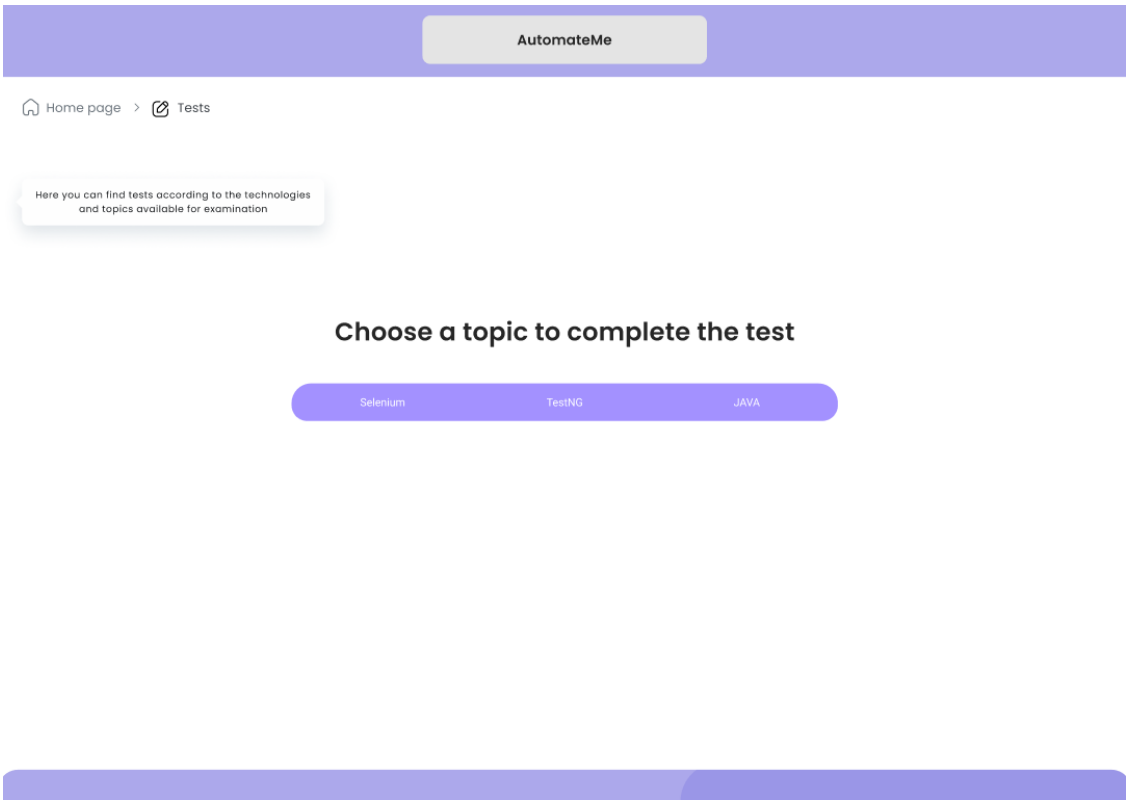


Рисунок 4.2. Сторінка вибору теми тесту.

Після вибору тесту користувач автоматично переходить до сторінки з восьми

питаннями, на які потрібно відповісти. На цій сторінці зображено поле з завданнями та можливими відповідями, графічний індикатор, який відображає прогрес проходження тесту, та кнопки для навігації. Наприклад, користувач може перейти до попереднього питання, якщо хоче перевірити свою відповідь, або перейти до наступного питання, щоб продовжити виконання.

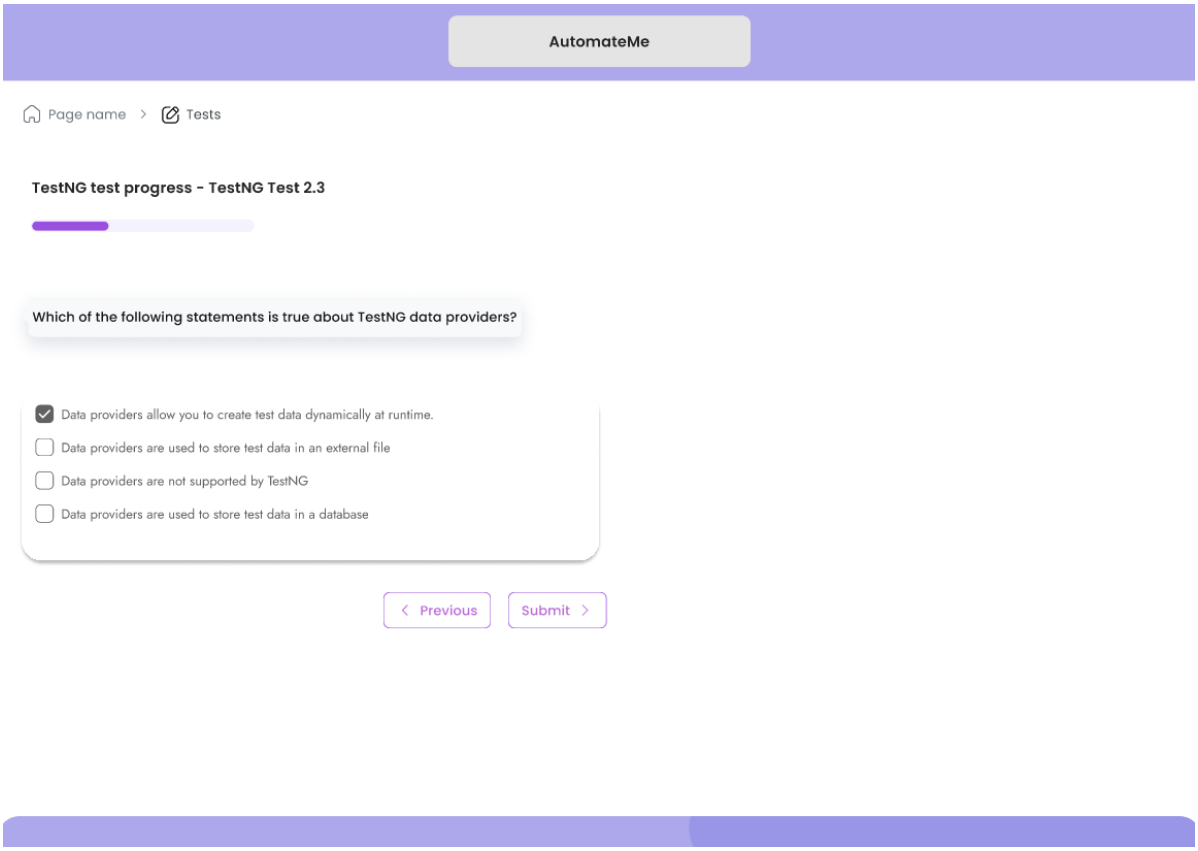


Рисунок 4.3. Процес проходження тесту.

Після завершення проходження тестування користувач переходить в меню, де йому надається можливість ознайомитися з результатами. Там можна побачити графічний показник, який демонструє кількість правильних відповідей, а також список обраних варіантів з розподілом на вірні та невірні. Окрім цього, на сторінці присутні візуальні елементи, які сприяють поліпшенню користувацького досвіду, зокрема, меню навігації, які надають можливість повернутися на основну сторінку, або перейти до вибору проходження інших тестів.

Наступним елементом є сторінка, що надає практичні та теоретичні поради та документацію для інсталяції та конфігурації усіх необхідних елементів та

технологій, та містить наступні блоки:

Встановлення Selenium

У даному блоку розглядається процес встановлення Selenium, який є одним з найпопулярніших інструментів для автоматизованого тестування. Надається інформація про те, як встановити Selenium на різних операційних системах, а також про необхідні налаштування.

Окрім інформації про встановлення та налаштування, у блоку знаходяться поради щодо використання Selenium, що дозволить користувачам швидше оволодіти цим інструментом. Детально розглядаються основні можливості фреймворку та його використання, що дозволить ефективніше застосовувати цей інструмент.

Встановлення TestNG

Цей блок розглядається для процесу встановлення TestNG, який є розширенням для Selenium і дозволяє зручно та ефективно організувати автоматизоване тестування. Надається інформація про те, як встановити програмне забезпечення, а також про те, як його налаштувати для виконання тестів.

У блоку щодо TestNG надано поради, які допоможуть користувачам швидше оволодіти цим інструментом, окрім того, детально розглянуто основні можливості TestNG та їх використання, що дозволить користувачам більш ефективно та гнучко використовувати та налаштувати його.

Посібники

У даному блоку надаються посібники з практичними порадами щодо використання Selenium та TestNG. Посібники включають в себе опис основних функціональних можливостей, які надають ці інструменти, а також приклади використання.

Посібники допоможуть користувачам краще ознайомитися з можливостями інструментів та зрозуміти, як їх використовувати. Також у них знаходяться приклади коду, які демонструють роботу Selenium та TestNG та поради коректної конфігурації їх роботи з Java.

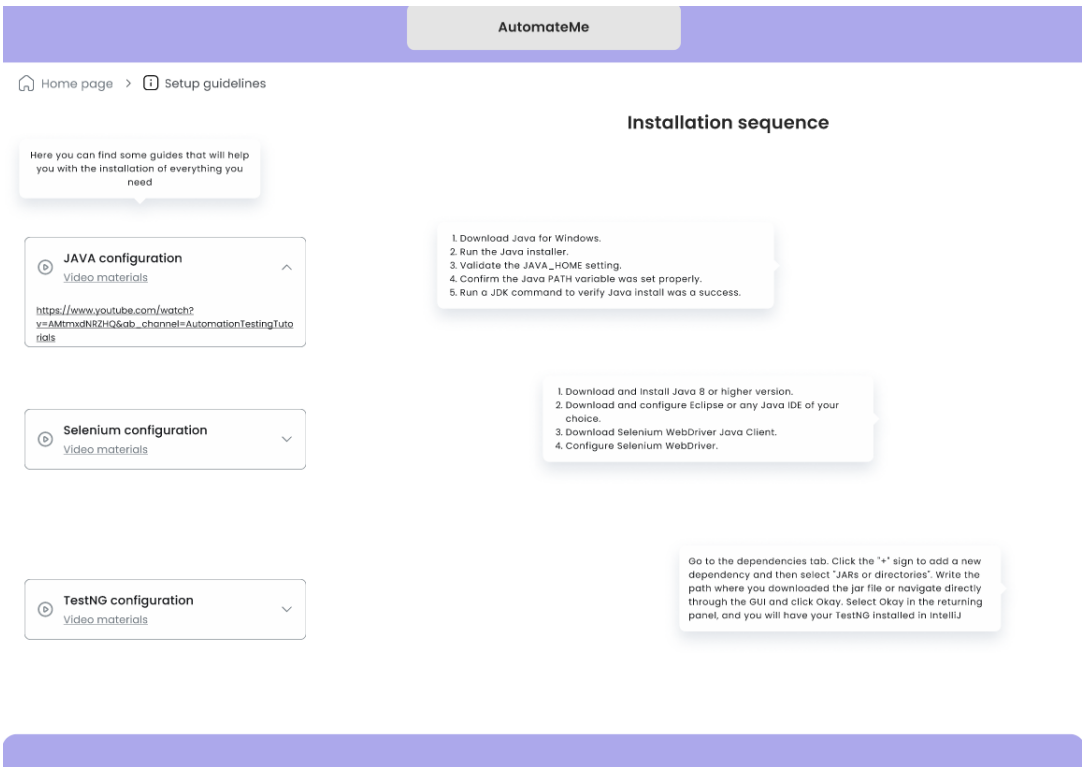


Рисунок 4.4. Сторінка із порадами для інсталяції та документацією.

Далі, розглянемо сторінку проходження практичних завдань. Так як, меню для вибору необхідних вправ ідентичне до сторінки вибору тестів, одразу перейдемо до ключових елементів функціоналу та дизайну.

Рисунок 4.5. Виконання практичного завдання.

Це - формулювання завдання, текстового редактора для введення коду, графічного показника проходження завдання та кнопок навігації. Тут виконуються практичні завдання із доповнення завдання у правій частині інтерфейсу.

4.2 Тестування роботи навчального модулю.

У процесі розробки програмного продукту для забезпечення його якості, було розроблено набір тестів для проведення функціонального тестування з фокусом на негативні та конфліктні сценарії та їх обробку. Цей набір перевірок дозволив виявити та виправити помилки, що можуть виникнути під час тестування програмного продукту та забезпечив високу якість роботи системи.

Для розробки набору тестів було використано різні техніки тест-дизайну, такі як передбачення помилок на основі попереднього досвіду, діаграми станів та переходів, тестування засноване на тест-сценаріях юз-кейсів. Ці техніки тест-дизайну дозволили створити різноманітний набір тестів та забезпечити максимальне покриття функціональності системи.

Окрім тестів для функціонального тестування, було проведено багато статичного тестування (аналізу коду програми) у вигляді тестування білого ящика, де основним процесом є статичний перегляд коду та тестування рішень і тверджень в коді. Цей підхід дозволив виявити та виправити помилки на ранніх етапах розробки та забезпечив високу якість програмного продукту.

Враховуючи важливість якості програмного продукту, набір тестів було створено з метою забезпечення високої якості роботи системи та зменшення кількості помилок, що можуть виникнути в процесі роботи з системою. Під час розробки програмного продукту було приділено особливу увагу тестуванню негативних та конфліктних сценаріїв, що дозволило забезпечити більш точне тестування та виявлення помилок на ранніх етапах розробки.

У підсумку, розробка набору тестів для проведення функціонального тестування з фокусом на негативні та конфліктні сценарії та їх обробку дозволила забезпечити високу якість роботи програмного продукту. Багатофакторний підхід

до розробки перевірок та забезпечення максимального покриття функціональності системи дозволили виявити та виправити помилки на ранніх етапах розробки та забезпечити більш точне тестування системи.

Номер кейсу	Заголовок	Пріоритет	Кроки виконання	Очікуваний результат
1	Вибір відповіді в тестовому завданні	Високий	1. Зайти в навчальний модуль. 2. Відкрити тестове завдання з вибором правильної відповіді. 3. Обрати правильну відповідь. 4. Натиснути кнопку "Перевірити".	Появляється повідомлення про те, що відповідь правильна.
2	Вибір неправильної відповіді в тестовому завданні	Високий	1. Зайти в навчальний модуль. 2. Відкрити тестове завдання з вибором правильної відповіді. 3. Обрати неправильну відповідь. 4. Натиснути кнопку "Перевірити".	Появляється повідомлення про те, що відповідь неправильна.
3	Введення коректного коду в редакторі	Високий	1. Зайти в навчальний модуль. 2. Відкрити редактор коду. 3. Ввести коректний код. 4. Натиснути кнопку "Перевірити".	Появляється повідомлення про те, що код введено правильно.
4	Введення некоректного коду в редакторі	Високий	1. Зайти в навчальний модуль. 2. Відкрити редактор коду. 3. Ввести некоректний код. 4. Натиснути кнопку "Перевірити".	Появляється повідомлення про те, що код введено неправильно.

Рисунок 4.6. Приклад виконуваних перевірок.

ВИСНОВКИ

У результаті виконання дипломної роботи створено навчальний модуль з автоматизованого тестування для вивчення Java, Selenium та TestNG.

1. Проведено аналіз ринку навчальних модулів та систем в галузі автоматизованого тестування. Виявлено, що попит на такі системи постійно зростає, оскільки вони дозволяють ефективно виконувати тестування програмного забезпечення та знижувати кількість помилок. Визначено, що автоматизоване тестування використовується в різних галузях, таких як банківська справа, медицина, телекомунікаційні технології та інші.

2. Проаналізовані переваги і недоліки програмних інструментів розробки, зокрема мови програмування Java, технологій Selenium та TestNG, розглянуто їхні основні функції. Виявлено, що вибрана мова є потужним інструментом для розробки програмного забезпечення, з великою кількістю бібліотек та фреймворків. Визначено, що Selenium дозволяє автоматизувати взаємодію з веб-додатками, а TestNG забезпечує потужні можливості для організації тестових сценаріїв та звітності.

3. Створено навчальний модуль за допомогою мови розмітки HTML та стилю сторінок CSS у середовищі розробки IntelliJ IDEA, який дозволяє студентам ознайомитись з основними принципами автоматизованого тестування та навчитись розробляти тестові сценарії, додано детальні пояснення та приклади коду для покращення розуміння матеріалу.

4. Використано програму Figma для створення елементів графічного дизайну та прототипування, що дозволило забезпечити зручний та привабливий інтерфейс навчального модулю. Проведено тестування навчального модуля з метою виявлення та виправлення помилок, що забезпечує його якість та ефективність.

Таким чином, розробка навчального модулю з автоматизованого тестування є важливим та перспективним напрямком у галузі навчання програмування та тестування.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. IntelliJ IDEA features [Електронний ресурс] // JetBrains. – 2023. – Режим доступу до ресурсу: https://www.jetbrains.com/idea/features/.
2. Java Documentation [Електронний ресурс] // Oracle. – 2023. – Режим доступу до ресурсу: https://docs.oracle.com/en/java/.
3. Офіційний веб-сайт Selenim. [Електронний ресурс]: [Веб–сайт]. – електронні дані. – Режим доступу: https://www.selenium.dev/documentation/en/.
4. W Hetzel. Complete Guide to Software Testing (2e)./- “QED Information Sciences: Wellesley MA”, 1993 – c. 110–131.
5. W3Schools [Електронний ресурс] – Режим доступу до ресурсу: https://www.w3schools.com/.
6. Udemy [Електронний ресурс] – Режим доступу до ресурсу: https://www.udemy.com/.
7. Coursera [Електронний ресурс] – Режим доступу до ресурсу: https://www.coursera.org/.
8. QA Learning [Електронний ресурс] – Режим доступу до ресурсу: https://qalearning.net/.
9. A. Spillner, T. Linz, H. Schaefer. Software Testing Foundations (4e)./- “Rocky Nook: San Rafael CA”, 2014 – c. 223–229.
10. Laiza Krispin, Janette Gregory. Agile Testing: A Practical Guide for Testers and Agile Teams. – c. 210–250.
11. TestNG [Електронний ресурс] – Режим доступу до ресурсу: https://testng.org/doc/.
12. Unmesh Gundecha, Satya Avasarala. Selenium WebDriver 3 Practical Guide: End-to-end Automation Testing for Web and Mobile Browsers with Selenium WebDriver, 2nd Edition. – c. 423–455.
13. Douglas Crockford. JSON: The Fat-Free Alternative to XML, 2009 – 55 p.

14. Elfriede Dustin, Jeff Rashka, John Paul. Automated Software Testing: Introduction, Management, and Performance: Introduction, Management, and Performance./- 2007 – с. 277–295.

15. Rex Black. Managing the Testing Process: Practical Tools and Techniques for Managing Hardware and Software Testing (3e)./- 2012 – с. 101–121.

16. Lisa Crispin, Janet Gregory. More Agile Testing: Learning Journeys for the Whole Team. – с. 152–182.

17. Alan Richardson. Selenium WebDriver with Java: Basics to Advanced, 2nd Edition. – с. 352–382.

18. Software Testing Help [Електронний ресурс] – Режим доступу до ресурсу: https://www.softwaretestinghelp.com/.

19. ISTQB [Електронний ресурс] - Режим доступу до ресурсу: https://www.istqb.org/.

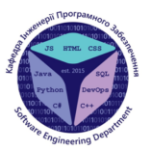
20. Ministry of Testing [Електронний ресурс] – Режим доступу до ресурсу: https://www.ministryoftesting.com/.

ДОДАТОК А
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



Розробка начального модулю з автоматизованого тестування на основі технологій Java, Selenium, TestNG

Виконав студент 4 курсу
групи ПД - 42
Студент Богдан Михайлович
Керівник роботи
Негоденко Олена Василівна

Київ – 2023

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

- **Мета роботи** - спрощення процесу навчання технологій автоматизованого тестування за допомогою модуля з технологіями Java, Selenium та TestNG.
- **Об'єкт дослідження** - процес навчання автоматизованого тестування.
- **Предмет дослідження** - навчальний модуль з автоматизованого тестування програмного забезпечення.

2

ЗАДАЧІ ДИПЛОМНОЇ РОБОТИ

1. *Провести аналіз ринку навчальних модулів та систем в цій галузі.*
2. *Проаналізувати переваги і недоліки програмних інструментів розробки.*
3. *Розглянути та проаналізувати особливості наявних систем вивчення автоматизованого тестування.*
4. *Створити навчальний модуль за допомогою мови розмітки HTML та стилю сторінок CSS у середовищі розробки IntelliJ IDEA мовою програмування Java з використанням технологій Selenium та TestNG.*
5. *Використати Figma для створення елементів графічного дизайну та прототипування.*

3

АНАЛІЗ АНАЛОГІВ

Платформа	Переваги	Недоліки	Наявність курсів з вивчення Selenium	Наявність курсів з вивчення TestNG	Наявність курсів з вивчення Java
W3school	Безкоштовна, містить багато корисної інформації	Не має великої кількості відеоуроків та практичних завдань	Так	Ні	Так
QAlearning	Має багато відеоуроків та практичних завдань	Платна	Так	Так	Так
Udemy	Має багато курсів з відеоуроками та практичними завданнями	Платна	Ні	Так	Так
Coursera	Має багато курсів з відеоуроками та практики	Платна	Так	Ні	Так

4

ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

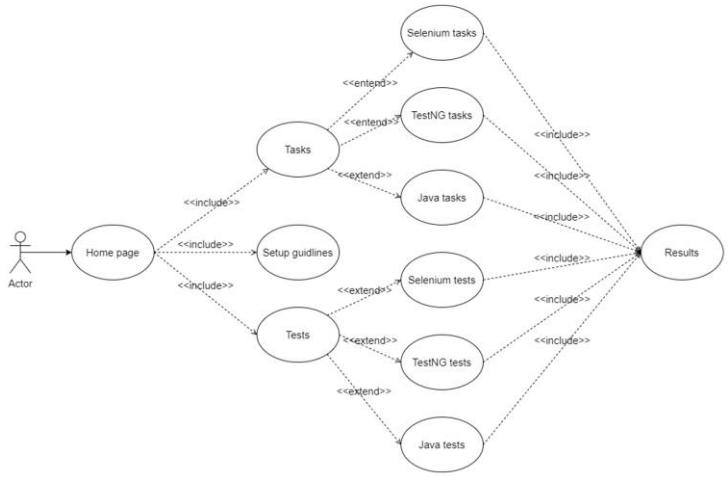
1. Розробити інтерактивні завдання з практикою написання авто тестів.
2. Розробити тестові завдання для перевірки знань користувачів.
3. Надати можливість перевірки правильності виконання завдань через перевірку відповідей користувачів.
4. Надати практичні поради та посібники для встановлення та налаштування використовуваного у навчальному модулі.

5

ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ

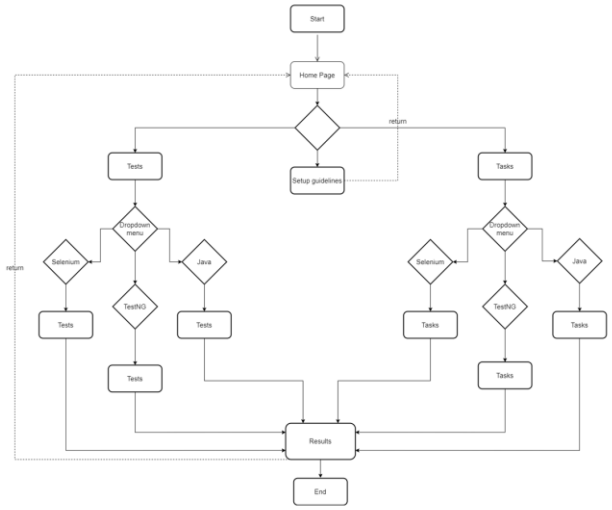
6

Діаграма використання



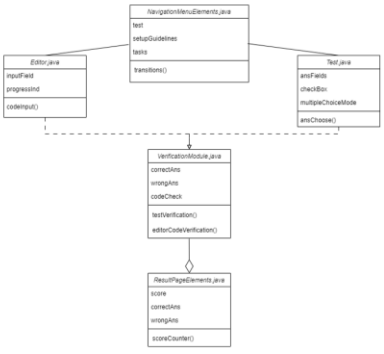
7

Діаграма діяльності

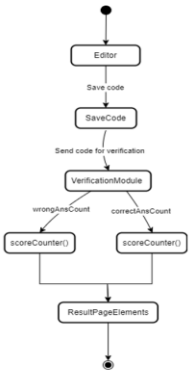


8

Діаграми класів та станів



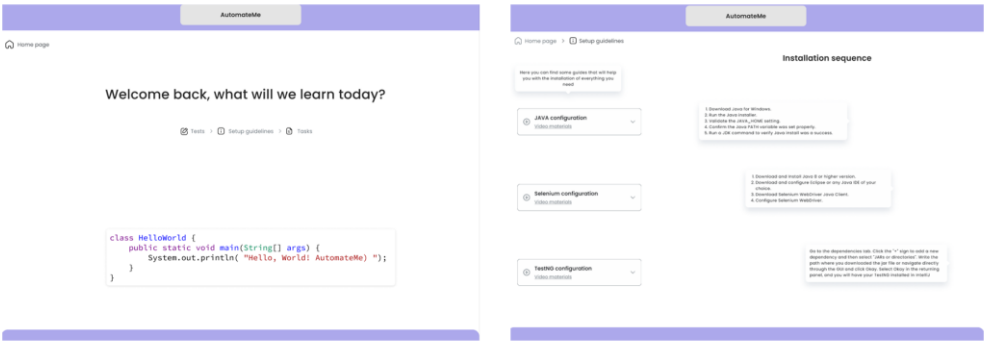
Взаємодія основних класів модулю



Робота валідаційного модулю

9

ЕКРАННІ ФОРМИ



10

ЕКРАННІ ФОРМИ

Проходження практичного завдання

11

ЕКРАННІ ФОРМИ

Проходження тестового завдання

12

ЕКРАННІ ФОРМИ

Сторінка із результатами

13

АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

Студент Б.М. Застосування автоматизованого тестування програмного забезпечення у галузі ІКТ/Негоденко О.В, Студент Б.М// Всеукраїнська науково-технічна конференція «Застосування програмного забезпечення в ІКТ». Збірник тез. 20.04.2023, ДУТ, м. Київ — К.: ДУТ, 2023. — С. 40.

Студент Б.М. Автоматизоване тестування в індустрії IoT/ Негоденко О.В., Студент Б.М. // IV Науково-технічна конференція «Сучасний стан та перспективи розвитку IoT». Збірник тез. 07.04.2023, ДУТ, м. Київ — К.: ДУТ, 2023. — С. 80.

14

ВИСНОВКИ

1. Проведено аналіз предметної галузі, за результатами якої було виявлено що більшість засобів не надають практичних вправ та обмежену кількість матеріалів для вивчення усього спектру технології авто тестів, тому було прийняте рішення по створенню навчального модулю з вивчення автоматизованого тестування програмного забезпечення.
2. Проведено дослідження засобів створення програмного забезпечення.
3. Зроблено концепцію ПЗ та вимоги, UML-діаграми класів, станів, діяльності.
4. Розроблено веб-додаток, який працює на JRE.
5. Реалізовані можливості проходження практичних й тестових завдань та оцінку користувача у навчальному модулі.

15

ДЯКУЮ ЗА УВАГУ!