

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Програмний модуль інтерактивного тестування знань»

Виконав: студент 2 курсу, групи 2КІ-21м
напряму підготовки (спеціальності)
123 — «Комп'ютерна інженерія»

Герасик Герасик О.С.

Керівник: к.т.н., доц. каф ОТ

Савицька Савицька Л.А.

«19» 12 2022 р.

Опонент: д.т.н., проф. каф ПЗ,

Романюк Романюк О.Н.

«20» 12 2022 р.

Допущено до захисту
Завідувач кафедри ОТ
д.т.н., проф. Азаров О.Д.

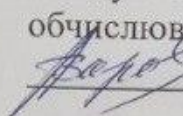
Азаров Азаров О.Д.
«21» 12 2022 р.

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки
Освітньо-кваліфікаційний рівень: магістр
Спеціальність 123 — «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

обчислювальної техніки

 проф., д.т.н. О.Д. Азаров

«15» 09 2022 р.

З А В Д А Н Н Я

НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Герасику Олександрю Сергійовичу

- 1 Тема роботи «Програмний модуль інтерактивного тестування знань», керівник роботи Савицька Людмила Анатоліївна, к.т.н., доцент кафедри ОТ затверджені наказом №205-А вищого навчального закладу від 15.09.2022 р.
- 2 Строк подання студентом роботи 15.12.2022 р.
- 3 Вихідні дані до роботи: проаналізувати методи і засоби для розробки програмного модуля інтерактивного тестування знань. Розробити алгоритм, що дозволяє автоматично генерувати тест із заданими налаштуваннями.
- 4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): розглянути та проаналізувати використання інтерактивного тестування як методу перевірки та діагностики знань. Провести порівняння з аналогами. Розробити схеми алгоритму автоматичної генерації тесту та структури програмного модуля інтерактивного тестування знань. Реалізувати систему у вигляді веб-додатку.
- 5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): технічне завдання, взаємодія між модулями програмного модуля

інтерактивного тестування знань, діаграми активності модулів, лістинг основного коду програми.

6 Консультанти розділів роботи представлені в таблиці 1.

Таблиця 1 — Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1,2,3,4	Савицька Людмила Анатоліївна, к.т.н., доцент каф. ОТ		
5	Небава Микола Іванович, к.е.н., професор каф. ЕПВМ		

7 Дата видачі завдання 07.09.2022 р.

8 Календарний план наведено в таблиці 2.

Таблиця 2 — Календарний план

№	Назва етапів виконання магістерської роботи	Строк виконання етапів роботи	Примітка
1	Постановка задачі роботи	01.09.21	вик.
2	Огляд та аналіз джерел інформації	02.09—09.09.22	вик.
3	Розробка технічного завдання	10.09—18.09.22	вик.
4	Огляд засобів для інтерактивного тестування знань	19.09—01.10.22	вик.
5	Аналіз засобів для інтерактивного тестування знань	02.10—18.10.22	вик.
6	Розробка алгоритму автоматизованої генерації тесту	19.10—31.10.22	вик.
7	Розробка діаграми активності алгоритму автоматичної генерації тесту	01.11—10.11.22	вик.
8	Практична реалізація програмного модуля інтерактивного тестування знань	11.11—16.11.22	вик.
9	Розрахунок економічної частини роботи	17.11—30.11.22	вик.
10	Оформлення пояснювальної записки та ілюстративного матеріалу	01.12—06.12.22	вик.
11	Аналіз виконання роботи, висновки, додатки	07.12—14.12.22	вик.
12	Перевірка якості виконання магістерської роботи та усунення недоліків	15.12.22	вик.

Студент

Герасик О.С.

Керівник роботи

Савицька Л.А.

АНОТАЦІЯ

УДК 004.42

Герасик О.С. Програмний модуль інтерактивного тестування знань. Магістерська кваліфікаційна робота зі спеціальності 123 — комп'ютерна інженерія, освітня програма — комп'ютерна інженерія. Вінниця: ВНТУ, 2022. 99с.

На укр.мові. Бібліогр.: 45 назв; рис.: 27; табл. 09.

У магістерській кваліфікаційній роботі розроблено програмний модуль інтерактивного тестування знань, який дозволяє підвищити об'єктивність та зменшити упередженість при оцінюванні знань. Обґрунтовано вибір технологій, проведено проектування та реалізовано програмний модуль інтерактивного тестування знань за допомогою мови програмування JavaScript та середовища розробки WebStorm.

Проведено тестування програмного модуля інтерактивного тестування знань.

В магістерській роботі були проведені економічні розрахунки для визначення доцільності та конкурентоспроможності створеного веб-додатку.

Ключові слова: тестування, інтерактивне тестування, автоматична генерація, оцінювання знань.

ABSTRACT

Gerasyk O.S. Software module for interactive knowledge testing. Master's qualification thesis on specialty 123 — computer engineering, educational program — computer engineering. Vinnytsia: VNTU, 2022. 99p.

In the Ukrainian language. Bibliography: 45 titles; fig.: 27; table 09.

In the master's qualification work, a program module for interactive knowledge testing was developed, which allows to increase objectivity and reduce bias in the assessment of knowledge. The choice of technologies was justified, the design was carried out and the software module for interactive knowledge testing using the JavaScript programming language and the WebStorm development environment was implemented.

Testing of the interactive knowledge testing software module was conducted.

In the master's thesis, economic calculations were carried out to determine the feasibility and competitiveness of the created software product.

Keywords: testing, interactive testing, automatic generation, knowledge assessment.

ЗМІСТ

ВСТУП	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ДЛЯ СТВОРЕННЯ ПРОГРАМНОГО МОДУЛЯ ІНТЕРАКТИВНОГО ТЕСТУВАННЯ ЗНАНЬ	10
1.1 Тестування як форма контролю та діагностики знань	10
1.2 Сучасні вимоги до тестування знань.....	13
1.3 Інтерактивна модель навчання.....	15
1.4 Інтерактивне тестування знань	19
1.5 Методи генерації тесту	21
1.6 Аналіз сучасних засобів тестування знань	23
1.6.1 LEARNINGAPPS	24
1.6.2 ONLINE TEST PAD.....	26
1.6.3 CLASSMARKER.....	27
2 РОЗРОБКА АЛГОРИТМУ ІНТЕРАКТИВНОГО ТЕСТУВАННЯ ЗНАНЬ	30
2.1 Розробка алгоритму інтерактивного тестування знань	30
2.2 Розробка алгоритму генерації тесту	32
3 РОЗРОБКА СТРУКТУРИ ТА ПРОЕКТУВАННЯ ПРОГРАМНОГО МОДУЛЯ ІНТЕРАКТИВНОГО ТЕСТУВАННЯ ЗНАНЬ	34
3.1 Розробка структури програмного модуля інтерактивного тестування знань.....	34
3.2 Обґрунтування вибору середовища розробки для реалізації програмного модуля інтерактивного тестування знань	36

					<i>08-23.МКР.019.00.000 ПЗ</i>				
					Програмний модуль інтерактивного тестування знань. Пояснювальна записка.	<i>Лім.</i>	<i>Маса</i>	<i>Масштаб</i>	
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>					
<i>Розроб.</i>		<i>Герасик О.С.</i>							
<i>Керівник</i>		<i>Савицька Л.А.</i>							
<i>Реценз.</i>		<i>Романюк О.Н.</i>				<i>Арк.</i>	6	<i>Аркушів</i>	99
<i>Н. Контр.</i>		<i>Швець С.І.</i>			<i>ВНТУ, гр. 2КІ-21м</i>				
<i>Затверд.</i>		<i>Азаров О.Д.</i>							

3.3 Обґрунтування вибору мови програмування для розробки програмного модуля інтерактивного тестування знань	39
3.4 Обґрунтування вибору бази даних для реалізації програмного модуля інтерактивного тестування знань.....	42
3.5 Обґрунтування вибору фреймворку для реалізації програмного модуля інтерактивного тестування знань.....	43
3.6 Розробка модуля генерації	46
3.7 Розробка модуля тестування	48
3.8 Розробка модуля створення та редагування банку завдань	48
3.9 Розробка модуля результатів	49
3.10 Розробка бази даних.....	49
4 ТЕСТУВАННЯ РОЗРОБЛЕНОГО ДОДАТКУ ІНТЕРАКТИВНОГО ТЕСТУВАННЯ ЗНАНЬ ТА АНАЛІЗ РЕЗУЛЬТАТІВ РОБОТИ.....	51
5 РОЗРАХУНОК ЕКОНОМІЧНОЇ ДОЦІЛЬНОСТІ СТВОРЕННЯ ПРОГРАМНОГО МОДУЛЯ ІНТЕРАКТИВНОГО ТЕСТУВАННЯ ЗНАНЬ	61
5.1 Комерційний та технологічний аудит науково-технічної розробки.....	61
5.2 Розрахунок витрат на здійснення науково-дослідної роботи.....	64
5.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором	69
ВИСНОВКИ	74
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	76
ДОДАТОК Ж Протокол перевірки кваліфікаційної роботи	79

ВСТУП

Актуальність теми дослідження полягає в потребі постійного оновлення та перевірки знань. Так як технології досить стрімко розвиваються, то комп'ютерні засоби широко використовуються у вирішенні цієї проблеми. Використання інформаційних технологій дає безліч переваг, серед яких можна відзначити індивідуалізацію навчальних процесів, забезпечення оперативного самоконтролю та контролю за допомогою діагностики помилок. Також є можливість зворотного зв'язку. Найбільшого розповсюдження в процесі навчання набуває тестування, яке використовується для контролю та діагностики знань. Тестування має безліч переваг в порівнянні з іншими формами контролю діагностики та перевірки знань, зокрема підвищення об'єктивності та зменшення упередженості.

Для того, щоб максимально підвищити об'єктивність та зменшити упередженість під час оцінювання потрібно мати можливість автоматизованого формування тесту та його перевірки, що забезпечить найкращі умови при яких кожен, хто буде проходити тест, буде знаходитись в рівних умовах з іншими.

Метою роботи є розширення функціональних можливостей при генерації тестів для тестування знань.

Для досягнення цієї мети необхідно вирішити такі **задачі**:

- проаналізувати сучасний стан розвитку інтерактивного тестування;
- на основі проведених досліджень розробити програмний модуль інтерактивного тестування знань для підвищення об'єктивності та зменшення упередженості під час оцінювання знань.

Об'єктом дослідження є процес тестування знань.

Предметом дослідження є програмні засоби для інтерактивного тестування знань.

Наукова новизна полягає в покращенні алгоритму автоматизованої генерації інтерактивних тестів, який дозволяє уникати повторень запитань в

інтерактивному тесті та встановлювати кількість запитань кожного рівня, а саме: низький, середній та високий. Також є можливість встановлювати співвідношення теоретичних та практичних запитань.

Практична цінність роботи полягає в розробці алгоритму та веб-додатку, який відрізняється покращеним алгоритмом автоматизованої генерації інтерактивного тесту, завдяки чому збільшується об'єктивність та зменшується упередженість під час оцінювання знань.

Результати розробки прийняті до друку у Міжнародному науково-технічному журналі «Інформаційні технології та комп'ютерна інженерія» у вигляді наукової статті.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ДЛЯ СТВОРЕННЯ ПРОГРАМНОГО МОДУЛЯ ІНТЕРАКТИВНОГО ТЕСТУВАННЯ ЗНАНЬ

1.1 Тестування як форма контролю та діагностики знань

Від правильності освоєння методів контролю знань багато в чому залежить ефективність процесу навчання. Контроль знань необхідний у кожній системі освіти та в кожній організації навчального процесу. Це спосіб управління навчальною діяльністю учнів. Але для того, щоб функції навчання реалізовувалися разом із функцією перевірки, необхідно створити певні умови, головною з яких є об'єктивність перевірки знань.

Важливим елементом навчально-виховного процесу є діагностика знань, умінь і навичок учнів. Результат навчання багато в чому залежить від його правильної організації. У процесі діагностики виявляються переваги та недоліки знань і вмінь студентів, що дає змогу керувати навчальним процесом, удосконалювати форми та методи навчання [1].

Тести є одним із засобів діагностики, який дозволяє швидко і ефективно перевірити навчальні та засвоювальні результати учнів.

Тест у педагогіці — це система спеціально підібраних завдань певного змісту і специфічної форми, що дозволяють кількісно оцінити навчальні досягнення. Цей термін введено американським психологом Джеймсом Кеттелом у 1899 році. Тести успішно використовуються в різних галузях, у тому числі і в процесі навчання. Отже, якісно складені тести дозволяють вчителю отримувати достовірну інформацію, щодо підготовленості учнів. За В. С. Аванесовим педагогічний тест це «...система репрезентативних паралельних завдань зростаючої складності, специфічної форми, яка дозволяє якісно та ефективно визначити рівень та структуру підготовленості учнів» [2].

Тестування — це метод визначення рівня знань і вмінь учнів за допомогою спеціальних тестових завдань, які подаються як правило, у формі запитань або задач.

За всю історію використання тестів як інструменту діагностики навчальних досягнень дітей ставлення до них було неоднозначним..

За результатами тестування можна судити про рівень знань, умінь і навичок учнів, ступінь розвитку окремих їхніх особистісних якостей, що означає успішність чи, навпаки, неуспішність на певному етапі навчання в цілому. класу або окремих учнів. Тести передбачають володіння учнем певним обсягом інформації, тому найчастіше використовуються для повторення та закріплення знань. Поєднання предметних тестів із тестами здібностей, зокрема здатності до навчання, покращує прогностичні властивості перевірки навчальних досягнень учнів, з високою ймовірністю прогнозування успішності їх подальшого навчання [3].

Серед головних переваг використання тестового контролю знань студентів можна виділити такі:

- можливість використання як засобу контролю всіх видів, а саме базового та попереднього, поточного та тематичного, етапного та прохідного, підсумкового та екзаменаційного контролю, а також самоконтролю;

- можливість детально перевірити рівень засвоєння кожного змістовного модуля дисципліни;

- наявність чіткої та однозначної відповіді і оцінювання на основі цінника;

- економія навчального часу при постійному контролі знань та об'єктивності оцінки результатів дослідження;

- зведення до мінімуму емоційного впливу вчителя на учня [4].

Крім цього, традиційно до переваг тестування відносяться:

- індивідуальний характер контролю, можливість контролювати роботу кожного студента, його особисту навчальну діяльність, а також можливість отримувати результати успішності групи, що допоможе викладачеві оцінити ефективність методики. і прийоми, що використовуються в класі;

- можливість регулярного та систематичного контролю контрольних робіт на всіх етапах навчального процесу;
- можливість поєднання із іншими звичайними формами педагогічного контролю;
- об'єктивність тестового контролю знань, що має виключати суб'єктивні оцінки та висновки викладача;
- можливість проведення комп'ютерного тестування в локальній мережі та паперовому варіанті;
- врахування індивідуальних особливостей учнів, що вимагає використання різних методик розробки тестів і тестових завдань залежно від рівня їх мовленнєвої компетенції відповідно до цих особливостей;
- висока змістовна точність тестового контролю, що полягає у включенні до тестових завдань дидактичних одиниць навчальної програми [5].

Проте тестування не можна розглядати як єдину форму контролю та діагностики знань студентів на кожному етапі навчального процесу та ще й при вивченнілюбих навчальних дисциплін, так як тести в основному перевіряють знання конкретних фактів, явищ, правил і частково перевіряють знання. сформованість умінь і навичок, але вони не можуть оцінити творчий рівень професійної компетентності студента, оскільки наявність вибору відповідей автоматично обмежує простір для творчих підходів до вирішення задач і дає змогу студентам із розвиненим інтуїтивним і практичним мисленням досягти значно кращі результати, ніж реальні. Тобто перевірка засвоєння конкретної теоретичної та практичної основи за окремо обраною темою, тестування є найбільш ефективним та ще й економною формою контролю та діагностики, але отримання результатів щодо творчого застосування знань, умінь та навичок студентами економічних спеціальностей в умовах, які наближені до реальності, тести не можуть дати об'єктивної оцінки, так як основна мета тестів — отримати інформацію, а не перевірити поведінку в практичних ситуаціях [6].

1.2 Сучасні вимоги, які ставляться до тестування знань

Так як один з найефективніших засобів контролю знань є тестування через те, що воно має безліч переваг над іншими видами контролю знань, а саме основною перевагою є — оцінка результатів при тестуванні більш об'єктивна та зменшення упередженості при оцінюванні, то до тестування ставиться безліч вимог. Основні з них: логічна форма висловлювання, стислість, правильність форми.

Логічна форма висловлювання є універсальною формою для чіткого висловлення ідеї людиною, яка має змогу виявити знання з вивченої дисципліни. У більшості випадках ця форма замінює питання.

Навчальні запитання породжують повні й неповні, правильні й неправильні відповіді, різні за формою, змістом і структурою, внаслідок чого оцінка таких відповідей вимагає участі вчителя й супроводжується певним суб'єктивізмом. Питання і відповіді іноді настільки розпливчасті і нудні, що для того, щоб дізнатися їхню правдивість, потрібен великий розум, а технологічний метод тестування полягає в тому, щоб чітко і швидко диференціювати відповіді. У цьому сенсі традиційні запитання та відповіді не є технологічними; їх не рекомендується включати в тестування.

Логічною перевагою тестового завдання є можливість його природного перетворення після доповіді учня у вигляді правильного чи хибного виразу.

Технологічна перевага тестових завдань виявляється в їх відповідності вимогам автоматизації рутинних елементів навчання та контролю знань. Якщо його вставляти в кожне завдання автоматично залежно від відповіді «1» чи «0», неважко побачити переваги використання такого оцінювання на основі нечіткої логіки. Це дозволить швидко записати відповіді та об'єктивно оцінити їх за попередньо розробленими правилами, які однаково поширюються на всіх без винятку. Завдання в тестовій формі легко вводяться в комп'ютер, компактно відображаються на екрані монітора і добре виділяються за формою і змістом відповіді.

Семантична перевага тестових завдань полягає в найкращому розумінні їх змісту та значення. Найперше це пов'язано зі словесною структурою завдання в тестовій формі: зміст тестового речення завжди краще зрозумілий, ніж зміст запитання. Тестові твердження не містять зайвих слів і навіть символів, тоді як питання вимагають серії інших слів і символів, які виражають потрібне значення та інтонацію.

Правильна форма завдань — це спосіб організації та ефективної організації змісту контрольної роботи. Організація забезпечує аналіз змісту навчальної дисципліни, класифікацію навчального матеріалу, створення тематичних і міжпредметних зв'язків, укрупнення навчальних підрозділів, подання цих підрозділів через елементи композиції завдань. Форма завдань є правильною, якщо вона дозволяє точно викласти зміст, зрозумілий кожному під час контрольної роботи, і виключає можливість неправильних відповідей з формальних причин. Якщо в завданнях є готові відповіді на вибір (зазвичай один правильний, а другий неправильний), то такі завдання варто називати завданнями з вибору однієї правильної відповіді. Логічною основою цієї форми є закон третього виключеного, вперше сформульований Арістотелем. Вибір правильної відповіді дає правдиве судження, а вибір неправильної дає хибне судження. Третього немає.

Звідси випливає методичний принцип: у кожному завданні з вибору однієї вірної відповіді має бути інша, яка дає чітку картину самого завдання і не допускає суперечливих тлумачень тим, хто виконує тест. Ще один варіант тих же самих тестових завдань, де є не одна, а декілька правильних відповідей. Це завдання на вибір правильних правильних відповідей. У свою чергу для порівняльної перевірки знань використовується інший, третій варіант завдань такої ж форми — з вибором однієї, найправильнішої відповіді із запропонованих відповідей, з різним ступенем правильності. Кожен варіант першої форми завдань можна поділити на кілька типів, наприклад завдання з двома, десятьма, чотирма, п'ятьма і більше відповідями. Повага права виключеної третьої сторони логічно забороняє використання відповідей, які

все ще існують на практиці: «немає правильної відповіді», «усі відповіді є правильні» або «всі відповіді є неправильні».

У другій формі завдання сформульоване таким чином, що готової відповіді немає, кожен має перевірити відповідь самостійно під час контрольної роботи, у визначеному місці. Ці завдання можна назвати відкритими, так як після виконання завдання з конкретною відповіддю твердження є істинним або хибним. Задачі, в яких необхідно поєднати елементи однієї множини з елементами іншої, можна назвати задачами на відповідність. Це є третьою формою. Наприклад, у випадках, якщо необхідно визначити правильний порядок обчислень, дій, кроків, операцій, понять у визначеннях, для визначення правильного порядку використовуються завдання. Це четверта форма квестів. Кожна з цих форм дозволяє перевірити певні види знань, а також потребує відповідних контрольних матеріалів. Вибір форм залежить від мети перевірки та змісту тексту, від технічних можливостей і рівня підготовки вчителя з теорії та методики перевірки контролю знань. Стислість завдань у тестовій формі зумовлена перевагами логічної форми висловлювання, завдяки якій завдання завжди є коротшими за завдання та запитання. Точність має забезпечуватись ретельним підбором слів, символів і графіків, що дозволяє мінімальними засобами досягти наочності змісту завдання. Виключаються повтори, незрозумілі, рідко вживані та незнайомі учням, іншомовні слова, що ускладнюють розуміння змісту. Добре, якщо завдання не містять більше одного підрядного речення. Для короткого завдання краще поставити його [7].

1.3 Інтерактивна модель навчання

Розглядають такі моделі навчання: пасивна, активна та інтерактивна [8].

Роль учня в пасивній моделі навчання полягає в тому, щоб бути суб'єктом навчання, тобто він повинен лише слухати і дивитися. Цю модель навчання ще називають «гличик і чашка». На початку 20 століття навчання в більшості навчальних закладів відбувалося за технологією «гличик-чашка».

Учитель виступає як фахівець у певній галузі, він повинен надати своїм учням оброблену інформацію, яку вони сприймають, запам'ятовують і вміють відтворювати. Під час іспиту перевіряється пройдений матеріал. Як зазначає А. Вірасінгхе, така модель навчання неефективна і має багато недоліків. По-перше, студенти, які навчалися за такою технологією і навчені лише запам'ятовувати інформацію, не готові вирішувати складні та незвичайні завдання на роботі. Пасивна роль не дозволяє креативити і проявляти творчі здібності. По-друге, учні бояться помилитися, що знижує рівень самоаналізу та самовиховання, які є важливими аспектами розвитку особистості. Такої ж думки дотримується Дж. Скрівенер, який вважає такі методи неефективними, оскільки студенти повинні мати можливість спілкуватися один з одним, виконувати певні завдання, отримувати оцінки та зворотний зв'язок щодо покращення своєї роботи. Однак Д. Керролл стверджує, що метод «розмова і крейда» або «гличик і чашка» стає досить ефективним способом навчання, якщо в класі використовуються такі технології, як розумна дошка або планшети [9].

У моделі активного навчання учень є суб'єктом навчання. У цій моделі навчання переважають завдання, що стимулюють пізнавальну активність і самостійність учнів. В. Матвійчук зазначає, що основними методами є самостійна робота, запитання учня до вчителя і навпаки, які розвивають творче мислення, проблемні та творчі завдання. Визначимо сильні та слабкі сторони моделі активного навчання. До переваг відносимо таке:

- інформація може бути отримана великою кількістю людей;
- швидкість засвоєння інформації учнями досить висока порівняно з пасивною моделлю навчання;
- високий рівень інформативності (проблемний метод);
- все залежить від кваліфікації вчителя;
- вчитель може контролювати рівень засвоєння знань.

Існують також недоліки такої моделі:

- спілкування відбувається лише між учителем і учнем;

- цей метод використовується тільки для опитувань;
- учень відчуває напругу «питати – не питати»;
- не з усіма учнями можна провести опитування на уроці [10].

Назва інтерактивної моделі походить від слів *inter* (взаємний), *act* (дія). Процес навчання має відбуватись в умовах постійної та активної взаємодії всіх учнів. Відомо, що термін «інтерактивна педагогіка» ввів Г. Фріц. Більшість сучасних методичних інновацій пов'язані з використанням інтерактивних технологій навчання. Семантичне значення слова інтерактив походить від англійського слова «interact», де «inter» означає взаємний, «act» — дія. Інтерактив — це процес навчання, під час якого відбувається безперервна та активна взаємодія всіх учасників процесу. Отже, інтерактивне навчання — це діалогове навчання, де відбувається обмін інформацією між викладачем та студентом. Отже, інтерактив здатний до взаємодії та діалогу. Інтерактивне навчання — є специфічною формою організації діяльності, яка має свою мету — створити комфортні умови навчання під час яких кожен учень відчує свою успішність та інтелектуальну ефективність.

За словами О. Кустовської, «тільки десять відсотків учнів здатні добре сприймати інформацію традиційними методами навчання. Інші дев'яносто відсотків також можуть засвоїти матеріал, але в інший спосіб — через діалог, виконання конкретних дій або творчих завдань. Результати цих досліджень підтверджують думку про необхідність переходу від традиційних форм навчання до інтерактивних. Головною метою традиційної освіти є повне засвоєння учнем інформації. Проте основною метою інтерактивних технологій є взаємонавчання, яке є «колективним вивченням конкретної проблеми та шляхів її ефективного вирішення». Учасники уроку розбирають проблему, узагальнюють відому інформацію, ведуть дискусію та відповідають на запитання.

На думку А. Мудрак призначення інтеракцій — стимуляція природної активності студентів:

— розумової (інтенсивність мислення, генерування ідей, висловлювання припущень, проектування, моделювання, конструювання, дослідження тощо, творча уява, зосередженість, увага, спостережливість, аналітико-синтезуючі операції); емоційної (емоційна напруга, переживання);

— соціальної (імітація виконання соціальних ролей, обмін думками, ставлення, судження тощо);

— фізичної (напруження, практична діяльність, рухливість).

При цьому дослідниця дала визначення, що "активність розуміється як енергійна, підсилена (інтенсивна) діяльність, діяльнісна участь, діяльнісний стан".

Саме завдяки поєднанню всіх видів природної активності веде до покращення результатів при вивченні нового матеріалу на уроці.

Виділимо основні характеристики інтерактивного навчання:

— навчальний процес відбувається за умови, коли постійно активно взаємодіють всі учні;

— це співнавчання, взаємонавчання (колективне, групове та звичайно кооперативне), коли учень з вчителем є рівноправними або рівнозначними суб'єктами такого навчання;

— організація інтерактивного навчання полягає в моделюванні життєвих ситуацій, застосуванні рольових ігор, спільному вирішенні проблем на основі аналізу обставин і провідної ситуації;

— інтерактивне навчання ефективно сприяє формуванню комунікативних умінь і навичок, розвитку цінностей, створенню атмосфери співпраці та взаємодії;

— цей тренінг виявляє лідерські якості, але передбачає як домінування одного з учасників навчального процесу над іншими, так і домінування однієї думки над іншою.

Однак у цієї моделі є і недоліки. По-перше, для вивчення певної інформації потрібно багато часу. Також необхідно створити інший підхід до

оцінювання знань учнів. Найбільшим недоліком є недостатня розробленість організації занять з використанням інтерактивних технологій [11].

Отже, проаналізувавши різні моделі навчання, можемо стверджувати, що інтерактивна модель навчання є найбільш ефективною у сучасному світі.

1.4 Інтерактивне тестування знань

Інтерактивні тести є одним із найоптимальніших методів контролю, який повністю задовольняє всі вимоги викладачів на рахунок об'єктивності отриманої оцінки та якості процесу контролю знань, позитивно впливає на ставлення студента до навчального процесу в цілому.

Переваги інтерактивного тестування:

- миттєве, та досить швидке і досить комфортне отримання результатів тесту;

- об'єктивність та правильність отриманої оцінки;

- інтерактивне тестування, з точки зору сприйняття студентами стає більш цікавішим і більш захоплюючим, якщо порівнювати цей вид тестування з традиційними іспитами за білетами або письмовими контрольними роботами;

- інтерактивні тести є досить справедливим методом оцінювання набутих знань, оскільки одночасно всі учні однієї конкретної групи знаходяться в однакових умовах, як під час самого інтерактивного тесту, так і при отриманні оцінки в кінці;

- тестування — це більш комплексний інструмент, за допомогою тестів викладач може встановити рівень набутих студентом знань як з окремої частини окремого предмета в цілому, так і з дисципліни в цілому;

- такий вид контролю ефективніший з економічної точки зору, через те що учитель створює базу запитань і може користуватися цією базою тривалий час;

- також інтерактивне тестування — більш гуманний інструмент через те що воно повинно знімати нервові напруження, яке майже завжди виникає

перед традиційним іспитом і в деяких випадках може суттєво вплинути на отриману оцінку;

— після контрольних робіт вчитель отримує інформацію про засвоєння навчального матеріалу для кожного учня індивідуально, аналізуючи ці дані, можна наочно побачити інформацію про труднощі засвоєння навчального матеріалу з тієї чи іншої теми.

Недоліки інтерактивного тестування:

— в деяких випадках причини неякісного написання контрольної роботи є об'єктивними і заслуговують на підтримку вчителя, а це неможливо помітити під час контрольної роботи;

— навіть якщо питання та завдання для проведення інтерактивних контрольних робіт складені грамотно та зрозуміло для учня, вони все одно не зможуть продемонструвати творчий рівень знань учня;

— іноді студентам просто не вистачає встановленого на це часу щоб скласти тест;

— щоб завдання, тобто запитання на відповідних рівнях, не повторювалися щоразу, потрібно з самого початку створення тесту вносити в базу відповідну кількість питань, щоб учень не отримував однакові запитання кожного разу в іншому судовому процесі, іншими словами, вчителю потрібен творчий підхід до підготовки тестових завдань, щоб бути впевненим в об'єктивності та достовірності отриманої оцінки;

— в інтерактивному тестуванні, як і в будь-якому процесі, не можна забувати про елемент випадковості, це означає, що інколи старанний учень не відповідав на запитання першого рівня, а студент, який не знав навчального матеріалу, з легкістю відповідав на запитання другого та третього рівня, причиною цього може бути як нездача питання наприклад першого рівня, так і навіть елементарне вгадування правильної відповіді на питання другого рівня, в результаті результати тестів спотворюються.

Аналізуючи переваги та недоліки інтерактивного тестування з дисциплін, очевидно, що, порівняно з традиційними формами іспитів,

запропонований вид контролю є достатньо об'єктивним та якісним, що обмежує вплив суб'єктивних факторів на оцінку студента в процесі перевірки. рівень знань з конкретної дисципліни [12].

1.5 Методи генерації тесту

Є три найвідоміші методи генерації інтерактивного тесту:

- параметризованих тестів;
- семантичних мереж;
- понятійно-тезисної моделі [13].

Методу параметризованих тестів полягає в тому, що студентам подається різні шаблонні завдання, яке відрізняється тільки певними параметрами, які генеруються автоматично [14]. Кожен учень отримує індивідуальне завдання, а система за власною формулою чи алгоритмом, підставляє параметри та отримує правильну відповідь для того щоб далі перевірити відповіді, уведені учнем. Недоліком методу є те, що він вузько предметно спрямований.

Перевагами методу є:

- відсутність будь-яких семантичних та лексичних невідповідностей;
- складність тестових завдань є заздалегідь визначеною та відсутні повторення інтерактивних тестових завдань.

Недоліком параметризованих тестів є те, що вони добре підходять для організації та контролю практичних навичок тільки у точних науках або програмуванні, проте їх не рекомендується використовувати для перевірки будь-яких теоретичних знань у гуманітарних науках.

Метод який базується на основі семантичних мереж ґрунтується на використанні різних тріад [15]. Завдання інтерактивного тесту будують упушенням одного елементу тріади і запитанням про те, чого не вистачає.

Перевага даного методу:

- можливість контролю рівня знань в майже у всіх науках.

Недоліками даного методу є:

— незрозумілість лінгвістично, а дуже часто — недоцільність завдань, які генеруються;

— складність завдань заздалегідь не визначають через що можливе повторення інтерактивних тестових завдань.

Метод понятійно-тезисної моделі. Головною структурною ідеєю для формування інтерактивних тестових завдань є зв'язок “навчальний матеріал”-“тези”-“поняття” [16]. Автоматична генерація тестів відбувається на основі створених різних тестових шаблонів. Понятійно-тезисна модель передбачає можливість розширення всіх множини класів користувачькими класами тез, це дозволяє розширювати модель і налаштовувати її під різні області.

Переваги даного методу:

— краща лексична зрозумілість порівняно з іншими методами;

— можливість контролювати та діагностувати знання в різних науках.

Недоліками методу є те, що заздалегідь не визначають складність завдань, а також є можливість повторення інтерактивних тестових завдань.

Порівняльна характеристика методів наведена в таблиці 1.1.

Таблиця 1.1 — Порівняльна характеристика методів

Назва методу	Відсутність семантичних та лексичних невідповідностей	Можливість контролю знань в різних науках	Неможливість повторення тестових завдань	Заздалегідь визначена складність завдання
Параметризованих тестів	+	—	+	+
Понятійно-тезисної моделі	+	+	—	—
Семантичних мереж	—	+	—	—

Методи, які описані вище не здатні враховувати складність тестових завдань, допускають великі семантичні, а іноді лексичні невідповідності. Тестові завдання повторюються та мають одну тему. Тому є підстава використовувати метод на основі методу параметризованих тестів.

1.6 Аналіз сучасних засобів тестування знань

На сьогодні серед сучасних засобів інтерактивного тестування знань найбільшого поширення набули Windows-додатки та веб-додатки.

До програм Windows належать різні програми, написані мовами програмування, такими як: C ++, Visual Basic, Delphi, C # та багато інших. Завдяки цим технологіям можна не тільки створити проект, але також і створити панель управління, де адміністратор має змогу змінити вміст проекту. Незважаючи на переваги та різні інструменти, які надані цими мовами програмування, користувачеві постійно доведеться встановити додаткове забезпечення. Наприклад, якщо робити з додатком, написаним на C#, то потрібно встановити .NET Framework.

Веб-додатки — додатки, які створені для роботи в серверному середовищі та написані різними мовами програмування, як: PHP та ASP та Perl, Ruby або Java. Ці мови надають відмінну можливість створювати повнофункціональні програми та ще й створювати системи управління для різних додатків.

Веб-додатки нажаль працюють лише в серверному середовищі, це означає вам потрібно встановити свій сервер для роботи. Такі програми також можна розповсюджувати в Інтернеті, там сервер вже налаштований, і для роботи користувач повинен мати лише веб-браузер та доступ до Інтернету. Ще перевагою веб-додатків є — користувач може мати доступ до проекту через Internet з кожного девайсу [17].

Переваги та недоліки windows і веб-додатків наведені в таблиці 1.2

Враховано всі переваги і недоліки та було прийнято рішення розробляти інтерактивний веб-додаток, бо він має низку переваг в порівнянні з windows-додатками.

Таблиця 1.2 — Переваги та недоліки windows та веб-додатків

Тип додатку	Відсутня потреба встановлення	Є можливість працювати без доступу до інтернету	Наявний доступ з різних пристроїв	Доступ з різних операційних систем
Windows-додатки	—	+	—	—
Веб-додатки	+	—	+	+

1.6.1 LEARNINGAPPS

LEARNINGAPPS — сервіс підтримки навчального процесу шляхом створення та підтримки інтерактивних ігрових вправ. Розрахована на дітей шкільного віку та дорослих. Завдання найкраще підходять для узагальнення та систематизації знань з певної дисципліни. Вправи чи блоки завдань ніколи не замінюють повноцінного уроку, тому вони можуть бути лише певною частиною під час уроку.

Сайт має бути особливо цікавий учням молодших та середніх класів, адже вправи побудовані у формах незвичайних інтелектуальних ігор. Можна вибрати рівень складності тестових завдань. Платформа підтримує більшість типів завдань: підбір пар, кросворди, класифікація, шкала часу, проста поза, сортування зображень, введення тексту, вікторина, заповнення пропусків, програми-сітки, аудіо/відеоконтент, «Хто хоче стати мільйонером? », головоломка «Вгадай», шибениця, слова з букв, гра «Пари». Категорія предметів дуже різноманітна: російська, англійська, німецька, італійська, французька, іспанська, українська та інші мови, інформатика, історія, інструменти, економіка, астрономія, біологія, географія, математика, мистецтво, музика, політологія, здоров'я, оточення, релігієзнавство, професійно-технічна освіта, психологія, філософія, робота, технічні науки,

робота, фізкультура, фізика та хімія. Багато мов інтерфейсу є доступні, включно з українською. Учитель повинен створити обліковий запис зі своїм іменем користувача/адресою електронної пошти та паролем. Студенти також повинні зареєструватися, перш ніж вчитель зможе додати їх до свого класу. Посилання на завдання можна надати у вигляді QR-коду, система автоматично генерує його для кожного завдання. Також можна додати практичні посилання на свій веб-сайт. Окрім підготовлених тестів для інших користувачів, платформа дає змогу створювати власні завдання. Ви також можете залишити відгук якщо дали правильну відповідь. Створене інтерактивне завдання може бути як відкритим (буде видимим для інших користувачів) так і закритим (доступним учням, яким ви надіслали посилання). Власні інтерактивні завдання зберігаються і ви можете редагувати їх за потреби або змінювати видимість для окремих користувачів тощо [18]. Приклад роботи програми LEARNINGAPPS показано на рисунку 1.1.

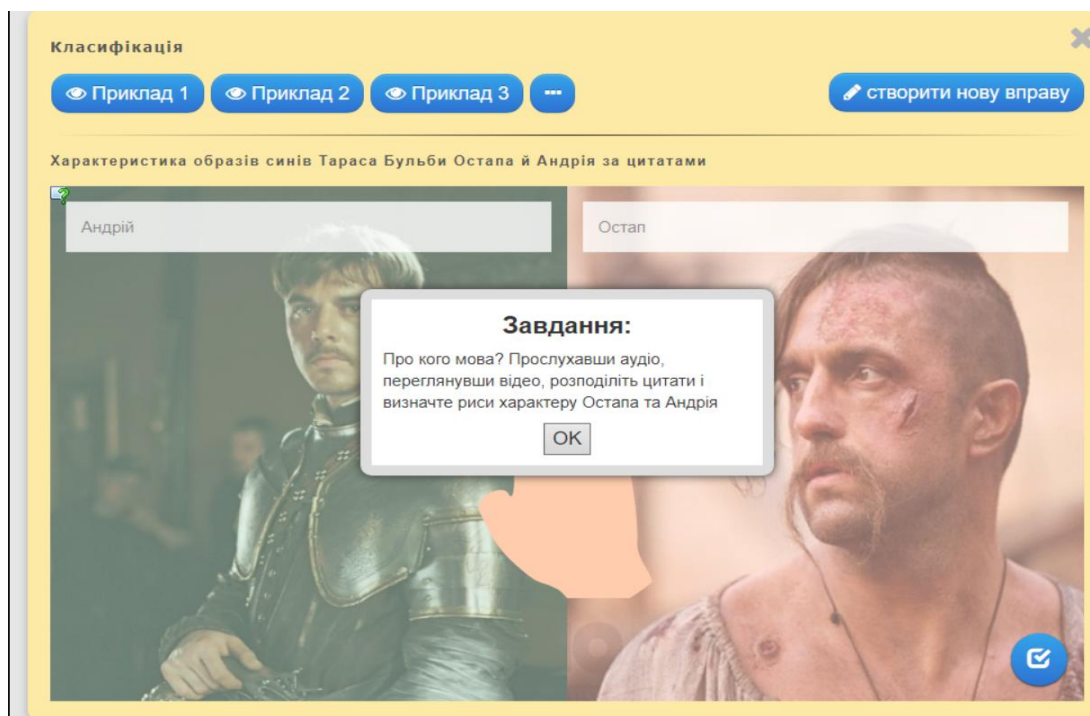


Рисунок 1.1 — Приклад роботи програми LEARNINGAPPS

1.6.2 ONLINE TEST PAD

ONLINE TEST PAD — безкоштовний багатофункціональний сервіс наприклад для організації навчального процесу або тестування через Інтернет. Зручний веб-додаток для створення будь-яких навчальних матеріалів і типів завдань і організації їх у папках. Крім того, на сайті є представлена величезна кількість інтерактивних завдань з предметів. Інтерфейс на сайті представлений кількома мовами, в тому числі українською. Включає інтегрований конструктор тестів із кількома налаштуваннями для типів запитань і оцінок, статистичними звітами та стилями завдань. Формат інтерактивних тестових питань включає безліч варіантів: вільна відповідь, завантаження файлу, послідовність і листування, заповнення пропусків, повзунок, службовий текст, послідовне виключення або інтерактивний диктант.

Конструктор кросвордів дозволяє створювати 5 типів завдань: класичний кросворд, сканування слова, японський кросворд, кольоровий японський кросворд, слово-заповнювач. Платформа містить детальні інструкції щодо заповнення онлайн-опитування за допомогою інструменту Survey Builder. У розділі «Тренажери діалогу» можна створювати розмовні ситуації з віртуальними учасниками різної складності, розгалуження та об'єктів. Від вибору перших фраз залежить подальша побудова мовної ситуації. Гарна можливість вивчити іноземні мови. Розділ «Складні завдання» містить необмежену кількість тестових запитань, кросвордів та ігор-головоломок. Цей формат підходить як для домашніх, так і для індивідуальних і контрольних робіт. Завдання будь-якого типу можна опублікувати у відкритому доступі в Інтернеті. Учитель має отримувати html-код, який дозволить розмістити інтерактивне завдання на власному сайті або блозі. Викладач також може додавати нових користувачів або групувати їх у групи та організації [19]. Приклад роботи програми ONLINE TEST PAD зображено на рисунку 1.2.

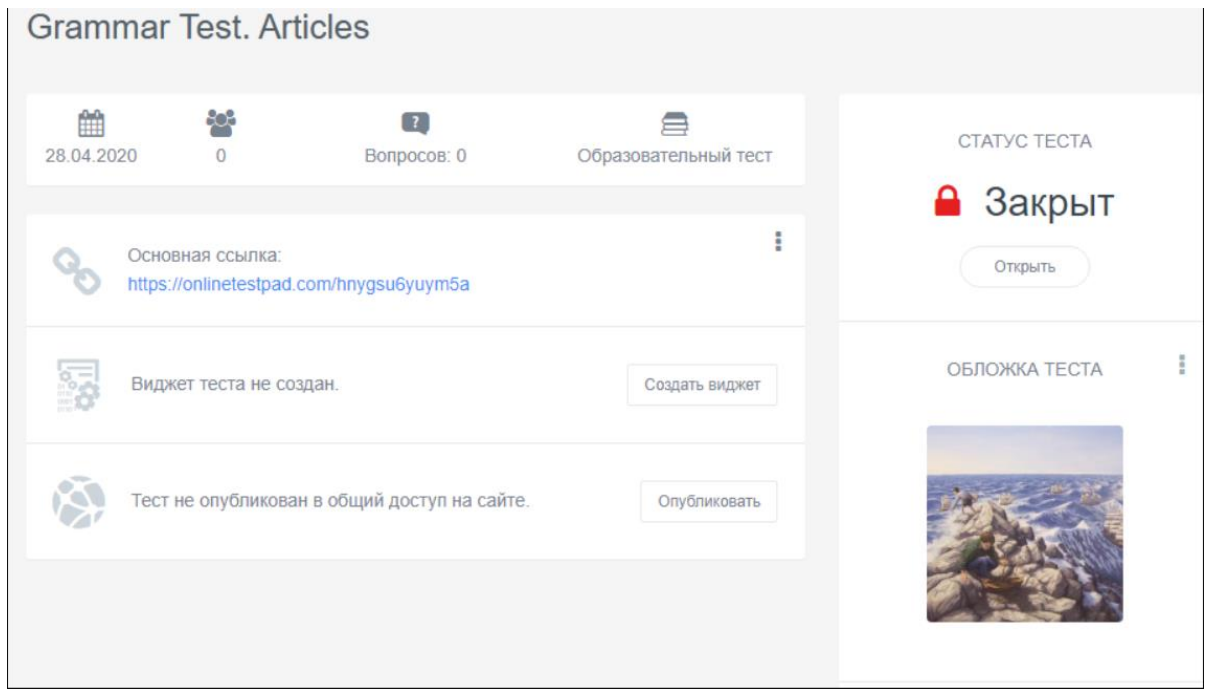


Рисунок 1.2 — Приклад роботи програми ONLINE TEST PAD

1.6.3 CLASSMARKER

CLASSMARKER — англomовний сайт для швидкої побудови тестових завдань і опитувань з найширшим форматом відповідей. Викладач має змогу створювати табо редагувати свої тести, також зберігати їх у банку завдань та у своєму профілі. Учитель створює свій клас, надсилаючи інформацію про реєстрацію студента на електронну адресу кожного студента або надсилаючи реєстраційний код. Безкоштовна версія дозволяє створювати до 100 тестів. Розширені пакети коштують від 16,50 доларів за 400 тестів на місяць, також є пакет 33 долари за 1000 тестів на місяць, але якщо ви використовуєте Google Chrome, то текст платформи може бути перекладений на українську мову. Цей ресурс дасть вам змогу продавати тести в Інтернеті з комісією 10% від зазначеної суми.

Ви можете отримати швидкі посилання на тестові завдання. До свого запитання можна додати зображення, аудіо, формулу чи відео. Потім вказуємо кількість варіантів відповідей. Ви також можете включити запитання щодо відгуків, кількості балів, шкали оцінювання [20]. Приклад роботи програми CLASSMARKER зображено на рисунку 1.3.

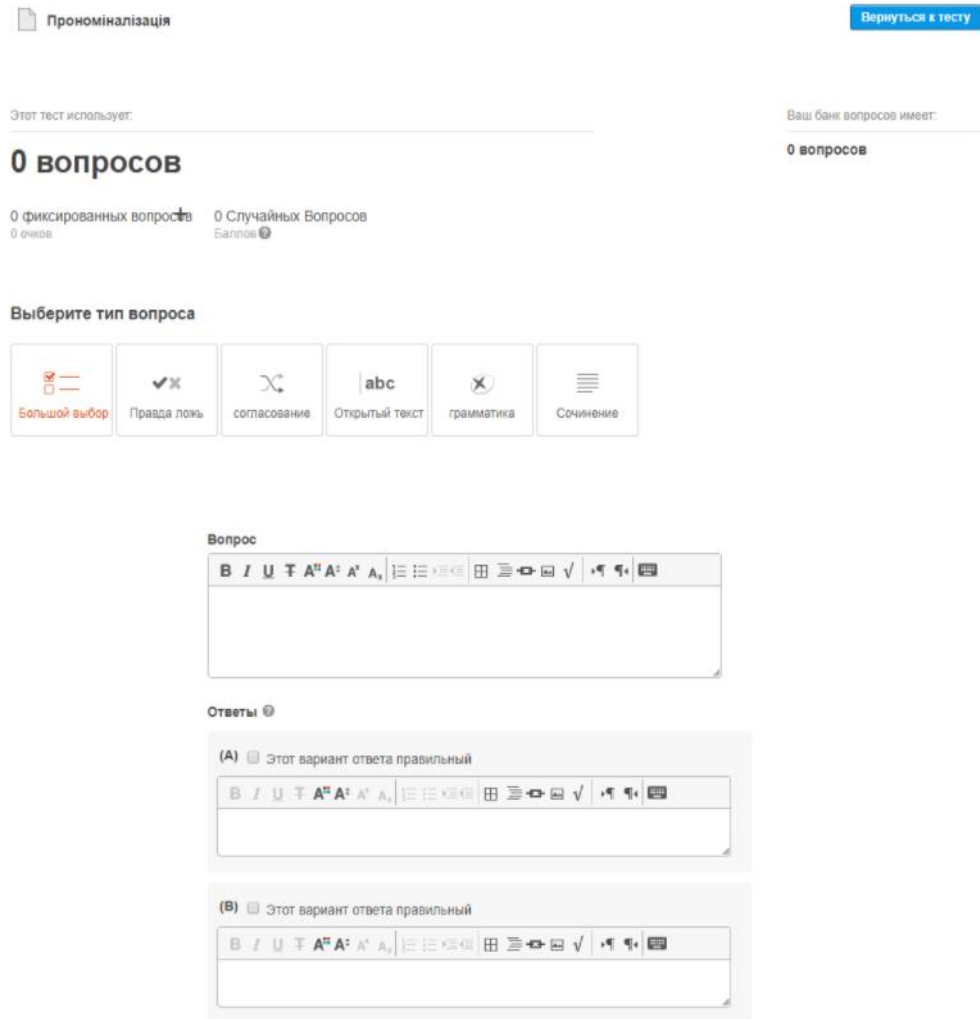


Рисунок 1.3 — Приклад роботи програми CLASSMARKER

Проаналізувавши дані додатки, можна також виділити певні переваги і недоліки, вони описані в таблиці 1.3.

Провівши аналіз існуючих інтерактивних додатків було виявлено те, що вони мають однаковий функціонал, хоча всі додатки, окрім classmarker, не підтримують автоматизовану генерацію тесту.

Хоч додаток classmarker має автоматизовану генерацію тесту, він не передбачає відсутності повторень завдань, а також встановлення складності завдань.

Таблиця 1.3 — Переваги та недоліки додатків learningapps, online test pad та classmarker

Назва додатку	Можливість автоматизованої генерації тесту	Відсутність повторень завдань	Можливість встановлювати складність завдань	Можливість перегляду результатів
learningapps	—	+	+	+
online test pad	—	+	—	+
classmarker	+	—	—	+

Таким чином, потрібно розробити додаток на основі classmarker та алгоритм генерації тесту має мати такі особливості:

- мати можливість автоматизованої генерації тесту;
- відсутні повторення завдань;
- є можливість перегляду результатів;
- можливість встановлення складності завдань.

2 РОЗРОБКА АЛГОРИТМУ ІНТЕРАКТИВНОГО ТЕСТУВАННЯ ЗНАНЬ

2.1 Розробка алгоритму інтерактивного тестування знань

Загальний алгоритм інтерактивного тестування знань має включати такі кроки:

— крок 1 — генерація тесту, під час даного кроку інтерактивний тест автоматизовано генерується з великого банку тематичних завдань, які мають знаходитись в базі даних і відповідно до налаштувань, які виставляє відповідальна за проведення тесту особа;

— крок 2 — вибір тесту, під час даного кроку користувач обирає інтерактивний тест для того щоб пройти зі списку доступних інтерактивних тестів, ці інтерактивні тести завчасно мають автоматизовано генеруватися відповідальною за проведення тесту особою, також інтерактивний тест генерується відповідно до налаштувань тесту;

— крок 3 — якщо тест обрано, то крок 4, якщо ні — крок 2;

— крок 4 — виведення інтерактивного тесту, під час даного кроку виводяться по черзі інтерактивні тестові завдання;

— крок 5 — проходження тесту, під час цього проходження інтерактивного тесту користувачу потрібно буде обирати відповіді з запропонованих;

— крок 6 — якщо тест буде пройдено, то перехід до кроку 7, якщо ні — крок 5;

— крок 7 — виведення результатів тесту користувачеві, після закінчення інтерактивного тесту, користувач має отримати результат проходження, в якому він зможе побачити наприклад скільки він дав правильних або неправильних відповідей на кожне запитання.

Діаграма активності інтерактивного тестування наведена на рисунку 2.1.



Рисунок 2.1 — Діаграма активності інтерактивного тестування знань

2.2 Розробка алгоритму генерації тесту

Після створення відповідальною за тест особою банку завдань та внесенням їх в базу даних, вона матиме змогу автоматизовано згенерувати інтерактивний тест для будь-якого користувача, під час генерації буде працювати алгоритм генерації, він матиме такі кроки:

- крок 1 — відбудеться зчитування налаштувань обраного тесту;
- крок 2 — відбудеться вибір запитання низької складності;
- крок 3 — якщо запитання вже є, то перехід до кроку 2, якщо ні — крок 4;
- крок 4 — якщо необхідну кількість запитань низького рівня складності згенеровано, то відбудеться перехід до кроку 5, якщо ні — крок 2;
- крок 5 — відбудеться вибір запитання середньої складності;
- крок 6 — якщо запитання вже є, то перехід до кроку 5, якщо ні — крок 7;
- крок 7 — якщо необхідну кількість запитань середнього рівня складності згенеровано, то відбудеться перехід до кроку 8, ні — крок 5;
- крок 8 — відбудеться вибір запитання високої складності;
- крок 9 — якщо запитання вже є, то перехід до кроку 8, якщо ні — крок 10;
- крок 10 — якщо необхідну кількість завдань високого рівня складності згенеровано, то відбудеться перехід до кроку 11, якщо ні — крок 8;
- крок 11 — кінець генерації інтерактивного тесту.

Даний алгоритм робить неможливим повторення тестових завдань у згенерованому тесті.

Діаграма активності генерації тесту наведена на рисунку 2.2.

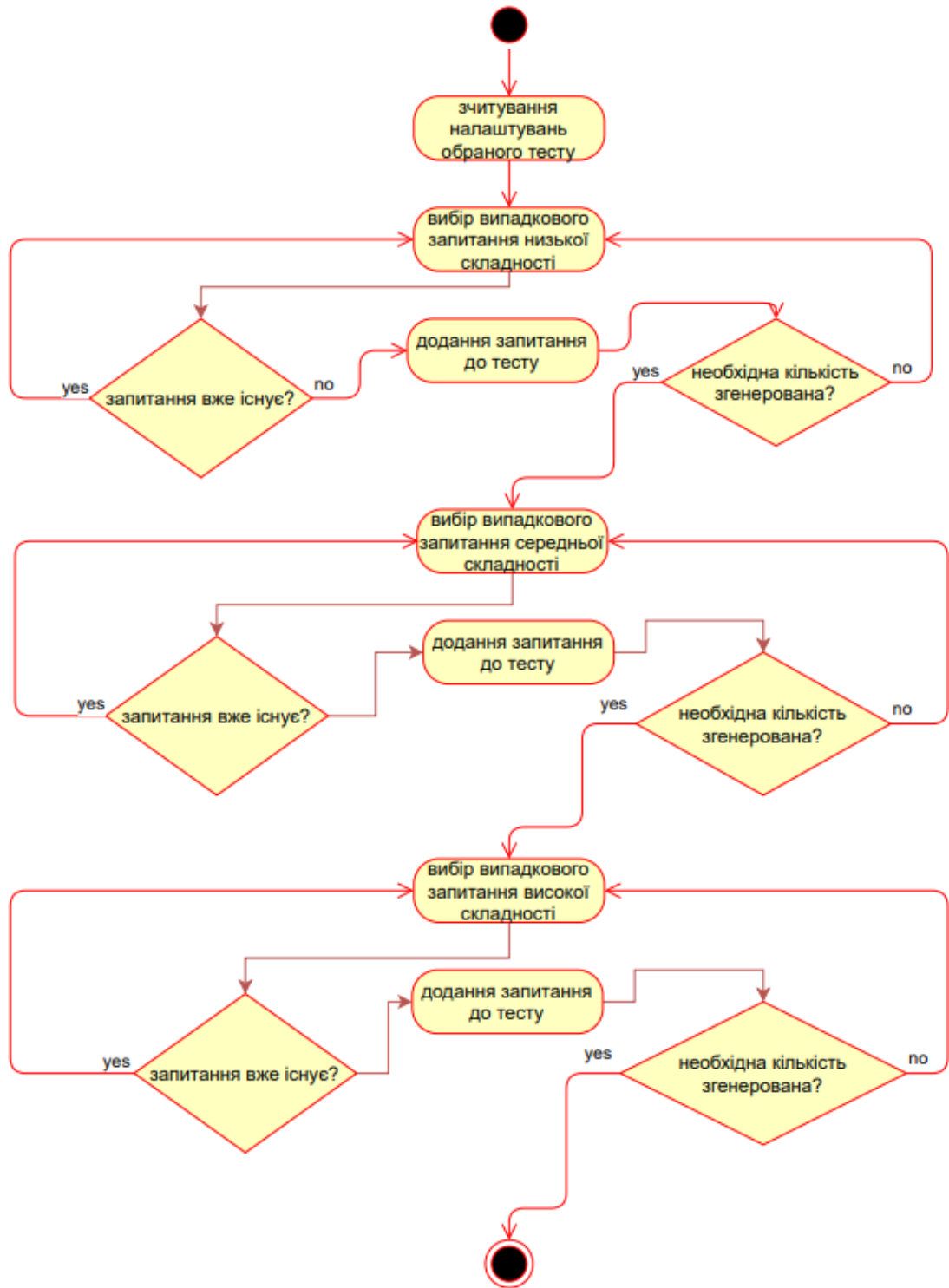


Рисунок 2.2 — Діаграма активності генерації тесту

3 РОЗРОБКА СТРУКТУРИ ТА ПРОЕКТУВАННЯ ПРОГРАМНОГО МОДУЛЯ ІНТЕРАКТИВНОГО ТЕСТУВАННЯ ЗНАНЬ

3.1 Розробка структури програмного модуля інтерактивного тестування знань

Розроблений програмний модуль інтерактивного тестування знань буде виконувати такі функції:

- функція виведення списку тестів для проходження;
- функція генерації тесту;
- функція виведення тестових завдань;
- функція підрахунку результату;
- функція виведення результату;
- функція створення тесту;
- функція редагування тесту.

Для того, щоб реалізувати означені функції модуль інтерактивного тестування знань повинен мати:

- модуль генерації;
- модуль тестування;
- модуль результатів;
- модуль створення та редагування банку завдань.

Генерацію тесту має виконувати модуль генерації. Головна задача модуля — автоматизована генерація тесту. Ще він дає змогу редагувати тест, тобто змінити його налаштування або згенерувати його знову.

Модуль взаємодіє з базою даних:

- отримує налаштування;
- отримує всі завдання;
- зберігає автоматизовано згенерований інтерактивний тест в базу даних.

Окремий модуль виконує такі важливі функції як:

- функція виведення тематичних тестових завдань;

- функція виведення можливих для проходження інтерактивних тестів;
- функція підрахунок правильних відповідей;
- функція виведення результату проходження інтерактивного тесту.

Це є модуль тестування. Він взаємодіє з базою даних, тобто отримує список інтерактивних тестів, а також отримує завдання тесту або відповіді до них. Модуль має функціонувати за таким алгоритмом:

- виводити список інтерактивних тестів які доступні для проходження;
- після початку тестування модуль отримує завдання відповідного тесту а також правильну відповідь;
- виводить запитання інтерактивного тесту а також його відповіді;
- зчитує і запам'ятовує відповіді дані користувачем;
- після закінчення інтерактивного тесту модуль видає результат проходження, в якому він детально має описати правильність даних відповідей. Якщо відповідь правильна, користувач отримає повідомлення про правильність. Якщо ж відповідь невірна, то буде видано повідомлення що відповідь неправильна на дане запитання.

За функції створення та редагування банку завдань має відповідати модуль створення та редагування банку завдань, він взаємодіє з базою даних, тобто отримує інформацію та записує інформацію або редагує інформацію в базі даних. Модуль має такі можливості:

- створювати інтерактивні завдання в базі даних;
- редагувати інтерактивні завдання в базі даних;
- видаляти інтерактивні завдання в базі даних;
- виставляти тип інтерактивного завдання;
- виставляти тему інтерактивного завдання;
- виставляти рівень складності інтерактивного завдання.

За функції виведення результатів проходжень тестів користувачами відповідає модуль результатів, він взаємодіє з базою даних, тобто отримує інформацію про результати проходжень тестів.

Модуль має такі можливості:

- виведення списку пройдених тестів;
- виведення користувачів, які пройшли певний тест;
- виведення часу проходження тесту загалом та затрачений час на кожне запитання окремо;
- виведення відповідей користувачів на кожне запитання.

Взаємодія між складовими програмного модуля інтерактивного тестування знань наведена в Додатку Б.

3.2 Обґрунтування вибору середовища розробки для реалізації програмного модуля інтерактивного тестування знань

Проаналізовано середовища розробки WebStorm і Atom. Програмне забезпечення JetBrains WebStorm — це хороший інструмент для створення веб-сайтів і редагування коду HTML, CSS і JavaScript. Рішення забезпечує швидку навігацію файлами та повідомлення про проблеми з кодом у реальному часі. JetBrains WebStorm має дозволяти додавати теги для документів HTML та елементів SQL безпосередньо до JavaScript. JetBrains WebStorm може розгорнути та синхронізувати проекти за допомогою FTP.

Використовуючи всі функції коду HTML/XHTML і XML, WebStorm забезпечує автоматичне доповнення посилань, стилів, атрибутів та інших елементів коду. Під час роботи з CSS заповнюється код класу та номери HTML і ключові слова. WebStorm містить автоматичні рішення таких проблем наприклад як вибір формату, класи, властивості, посилання на файли та різні атрибути CSS. Таке рішення використовує потужність інструменту кодування Zen для створення HTML-коду, який відображає діяльність бренду на веб-сайті. WebStorm завершує код JavaScript для ключових слів, тегів, змінних, налаштувань і функцій DOM, а також підтримує деякі функції популярних браузерів. Реалізовані в рішенні функції рефакторинга JavaScript дозволяють трансформувати структуру коду та js-файли.

WebStorm також забезпечує налагодження JavaScript і надає різні функції: виявлення точок зупину в HTML і JavaScript, налаштування

параметрів точок зупину, тестування синтаксису коду в реальному часі та багато іншого. Продукт також підтримує платформи JQuery, Qooxdoo та Bindings та YUI та Prototype, DoJo та MooTools. WebStorm забезпечує інтегровану перевірку тексту на теги, кодові послідовності та орфографічні помилки. WebStorm дозволяє редагувати файли та автоматично синхронізувати їх за запитом під час віддаленої роботи або під час зберігання.

Продукт підтримує контроль версій і всі попередні версії коду, а ще реєструє всі дії та зміни. Коли створюєте історію в WebStorm, ви також можете відновлювати вирази коду та блоки і навіть цілі файли.

Особливості WebStorm:

- підтримка різних фреймворків Vue, Meteor, React, Angular;
- інтеграція Apache Cordova, PhoneGap, React Native і IonicFramework для розробки різних мобільних додатків;
- підтримка Polymer;
- інтеграція Gulp;
- інструмент такий як spy-js для трасування та профілювання різних додатків Node.js з автозавершення коду та виявлення проблемних фрагментів;
- підтримка модулів таких як EcmaScript 6;
- автоматичне оновлення всіх додатків Node.js;
- перевірка стилістики всього коду за допомогою ESLint і JSCS;
- вбудована підтримка різних плагінів як EditorConfig, він дозволяє автоматично визначати та вибирати конфігурації стилю коду для різних IDE;
- удосконалення CSS3;
- postfix-шаблони наприклад для JavaScript;
- нова функція для налагодження різних додатків, що відображає значення змінних безпосередньо в самому редакторі;
- функції такі як Scratch Files для створення начерків наприклад безпосередньо в редакторі;
- підтримка останніх версій Stylus та Sass [21].

Atom — безкоштовний редактор тексту і коду, який розроблений GitHub, для використання як окремого рішення та як стек технологій наприклад для створення різноманітних спеціалізованих рішень. Наприклад, на платформі Atom створено середовище розробки «Visual Studio Code» від «Microsoft» і «Nuclide» від «Facebook».

Atom надає міжплатформні інструменти редагування коду, містить інтегрований менеджер пакетів та інтерфейс навігації файлової системи, надає інструменти для спільної роботи над кодом, має інтелектуальну систему автозаповнення, забезпечує режими сумісності Vim і Emacs і підтримує різні API для розробки розширень. Також ви можете відкривати кілька файлів у багатьох вкладках і переглядати їх одночасно та за допомогою вертикальної або горизонтальної обробки. Інтерфейс налаштовується за допомогою тем оформлення, підтримуваних вкладок, вкладок, інтелектуального контекстного пошуку коду, переносу блоків коду, використання кількох курсорів і областей вибору одночасно, зміни візуалізації, а також автозаповнення і перевірки коду для різних мов (Python, Ruby та SQL). Perl, PHP, Objective—C, C/C++, Java, JavaScript, Go тощо). Ви можете використовувати тег Markdown для створення статей і документації.

Функціонал редактора створюється шляхом надання набору додаткових пакетів, для установки яких пропонується вбудований менеджер пакетів `apm`, схожий на `npm` з проекту Node.js. Формат пакета схожий на `npm` і відрізняється наданням додаткових блоків для визначення меню, стилів, комбінацій клавіш, завдань логічної активації. Створення надбудови дещо відрізняється від створення програми Node.js, які включають всі доступні модулі Node.js, а також такі популярні бібліотеки JavaScript, як jQuery, Underscore і SpacePen. Додатки реалізують усі функції, які виходять за рамки базових модифікацій різного коду, а також включаючи панель та підсвічування синтаксису, а також реєстрацію інтерфейсу та форми файлів тощо. Крім базових доповнень, існує каталог сторонніх пакетів, який містить вже більше двох тисяч доповнень і майже сімсот тем оформлення.

Atom базується на компонентах таких як Electron (раніше Atom Shell), який є ядром на основі Chromium і Node.js, на якому реалізовано редактор. Electron — це окрема структура, яку можна використовувати для створення будь-якої спеціальної програми, логіка якої визначається JavaScript, HTML і CSS, а функціональність якої можна розширити за допомогою додаткової системи. Розробники також мають доступ до модулів Node.js, а також розширених API для створення користувацьких діалогів, створення контекстних меню, керування вікнами та взаємодії з підсистемами Chrome.

Необхідність використовувати рідне ядро браузера на основі Chrome замість розробки редактора як веб-програми, що працює у звичайному браузері та через необхідність реалізації додаткових функцій, які недоступні через звичайні веб-API. Наприклад, Atom забезпечує вбудований файловий менеджер і гнучкі функції пошуку файлів, такі які неможливо реалізувати за допомогою наприклад звичайних веб-додатків. Редактор також використовує деякі і внутрішні підсистеми Chromium, наприклад такі як механізми регулярних виразів та власні діалогові елементи. Більше того, оскільки компоненти Atom майже завжди працюють локально, то це спрощує розробку надбудов, доступ до ресурсів і спільний доступ [22].

Для реалізації програмного модуля інтерактивного тестування знань було обрано середовище розробки WebStorm.

3.3 Обґрунтування вибору мови програмування для розробки програмного модуля інтерактивного тестування знань

Проаналізовано мови програмування JavaScript і C#.

JavaScript — це динамічна та об'єктно-орієнтована мова програмування. Є реалізацією стандарту ECMAScript. Найчастіше JS використовується для створення скриптів сайту, які наприклад дозволяють клієнту взаємодіяти з користувачем та керувати браузером, а також асинхронно обмінюватися даними з сервером і змінювати структуру і зовнішній вигляд сайту.

JavaScript класифікується як мова сценаріїв з динамічним типом. Крім того, JavaScript також частково підтримує інші парадигми програмування а також деякі пов'язані елементи архітектури, зокрема: динамічну і слабку типізацію, також автоматичне керування пам'яттю та успадкування прототипів, такі функції, як об'єкти першого класу..

JavaScript використовується для:

- написання сценаріїв для веб-сайтів, що забезпечують їм інтерактивність;
- створення односторінкових та прогресивних веб-додатків (React, AngularJS, Vue.js);
- програмування на стороні сервера (Node.js (Express.js));
- стаціонарні програми (Electron, NW.js);
- мобільні додатки (React Native, Cordova);
- сценарії в додатках (наприклад, у програмах Adobe Creative Suite або Apache JMeter);
- всередині документів PDF.

Переваги JavaScript:

- швидкість — JavaScript може бути дуже швидким, так як він часто запускається безпосередньо в браузері клієнта, поки не потрібні зовнішні ресурси, JavaScript не сповільнюється під час викликів серверного сервера, крім того, усі основні браузери підтримують JIT-компіляцію JavaScript (точно вчасно), тобто немає великої необхідності компілювати код перед його запуском;
- простота — синтаксис JavaScript натхненний Java, і його також відносно легко вивчити порівняно з іншими популярними мовами, такими як C++.
- популярність — JavaScript є майже повсюди в Інтернеті, і також з появою Node.js він використовується все більше і більше на стороні сервера, існує незліченна кількість ресурсів для навчання JavaScript, як StackOverflow,

так і GitHub демонструють дедалі більше проектів JavaScript, і популярність, яку він здобув за останні роки, лише зростатиме;

— сумісність — на відміну наприклад від PHP або інших мов сценаріїв, JavaScript можна вбудувати в майже будь-яку веб-сторінку, JavaScript можна також використовувати в багатьох різних типах програм, наприклад підтримуючи інші мови, такі як Perl і також PHP;

— навантаження на сервер — JavaScript працює на стороні клієнта, тому, як правило, він зменшує потребу в серверах, і простим програмам може і взагалі не знадобитися цей сервер;

— розширені інтерфейси — JavaScript також можна використовувати для створення наприклад таких функцій, як перетягування та скидання, і таких компонентів, як повернення, що постійно покращує інтерфейс користувача та роботу в Інтернеті;

— розширена функціональність — розробники можуть розширити функціональність веб-сторінок, написавши фрагменти JavaScript для різних сторонніх доповнень, як Greasemonkey;

— універсальність — є багато різних способів використовувати JavaScript через сервери Node.js, отже якщо ви хотіли завантажити Node.js за допомогою Express, то використовувати базу даних документів, таку як MongoDB, і використовувати JavaScript на інтерфейсі, можна розробити цілу програму JavaScript спереду назад, коли використовуючи лише JavaScript;

— оновлення — з моменту появи ECMAScript 5 (специфікація сценаріїв, на яку покладається JavaScript), ECMA International займається оновленням JavaScript щорічно, наразі ми отримали підтримку браузера для ES6 у 2017 році і сподіваємось на підтримку ES7 у майбутньому [23].

C# — це об'єктно-орієнтована, безпечна для типів мова програмування для .NET розроблено Андерсом Галесбергом, Скоттом Вільхамутом і Пітером Голдом під Microsoft Research (належить Microsoft).

Синтаксис C# схожий на C++ і Java. Мова суворо статично типізована, також підтримує поліморфізм, перевантаження операторів та покажчики на

функції-члени класу, події, атрибути, властивості, коментарі у форматі XML винятки. Прийняття багатьох з своїх попередників — Object Pascal, C++, Module і Smalltalk — C#, засноване практично на практиці їх використання, виключає деякі моделі, такі які виявилися проблематичними наприклад при розробці програмного забезпечення, такі наприклад як C#, на відміну від C++, не контролює успадкування багатьох класів [24].

Для розробки мого програмного модуля інтерактивного тестування знань було також обрано мову JavaScript.

3.4 Обґрунтування вибору бази даних для реалізації програмного модуля інтерактивного тестування знань

Проаналізовано базу даних SQL і MySQL.

MySQL — це є безкоштовна система керування реляційними базами даних, розроблена ТХХ для підвищення швидкості обробки великих баз даних. Також ця система управління базами даних з відкритим кодом була створена як альтернатива комерційним системам. MySQL була дуже схожа на mSQL з самого початку, але з часом вона розвинулася, і тепер MySQL є однією з найпопулярніших систем керування базами даних. Він в основному використовується і для створення динамічних веб-сайтів, так як має відмінну підтримку різних мов програмування.

MySQL — це компактний багатопоточний сервер баз даних. Відрізняється високою швидкістю, стабільністю і простотою використання.

MySQL вважається хорошим рішенням для програм малого та середнього розміру. Коди серверів скомпільовані на кількох платформах. Повністю можливості сервера представлені в системах UNIX, де є підтримка багатопоточної обробки, що підвищує ефективність всієї системи.

Параметри сервера MySQL:

- простий в установці і використанні;
- підтримується велика кількість користувачів, які одночасно працюють з вашою базою даних;

- кількість рядків у таблицях бази може досягати 50 мільйонів;
- висока швидкість виконання команди;
- наявність простої і ефективної системи безпеки [25].

SQL, також відомий як Microsoft SQL, з'явився набагато раніше, ніж MySQL. Корпорація Майкрософт розробила SQL у 1980-х роках. Вони залишаються основою якісного SQL навіть після багатьох років і є важливим рішенням для програмного забезпечення великих підприємств.

SQL Server більше підходить для розробників, таких як використовують .NET як мову програмування як конкуруючу комбінацію PHP із MySQL.

SQL Server використовує єдину систему, розроблену Microsoft, проти кількох механізмів, запропонованих MySQL. Це дає розробникам MySQL більше гнучкості, оскільки вони також можуть вибирати різні системи для різних таблиць та на основі швидкості, надійності чи інших параметрів. Популярним двигуном MySQL є InnoDB, який втрачає деяку швидкість, але забезпечує підвищену надійність.

Корпорація Майкрософт вимагає від вас придбати ліцензію на запуск кількох баз даних на SQL Server. Є безкоштовна версія, але вона лише для ознайомлення з реляційною СУБД. MySQL, з іншого боку, використовує ліцензію GNU, що робить його абсолютно безкоштовним [26].

Для реалізації мого програмного модуля інтерактивного тестування знань була також обрана база даних MySQL.

3.5 Обґрунтування вибору фреймворку для реалізації програмного модуля інтерактивного тестування знань

Проведено аналіз фреймворків Vue.js і React.js.

Vue.js — це фреймворк JavaScript, який використовує шаблон MVVM для створення користувацьких інтерфейсів на основі моделей даних за допомогою реактивного зв'язування даних. MVVM — це модель розробки, таку яка дозволяє розділити додаток на три функціональні частини:

— `model` — основна логіка програми (наприклад робота з даними, розрахунки, запити тощо).

— `view` — перегляд або відображення (наприклад інтерфейс).

— `viewmodel` — модель подання, вона служить шаром між видом та моделлю.

Цей розділ дозволяє наприклад пришвидшити розробку та підтримувані програми — ви можете також змінити один компонент, щоб не впливати на код. Взаємодія між функціональними частинами показана на рисунку 3.1.

Vue використовує синтаксис шаблонів на основі HTML, який дозволяє декларативно прив'язувати відтворення DOM до базових екземплярів даних у Vue. Усі шаблони Vue є також дійсними HTML і можуть аналізуватися браузерами та аналізаторами HTML.

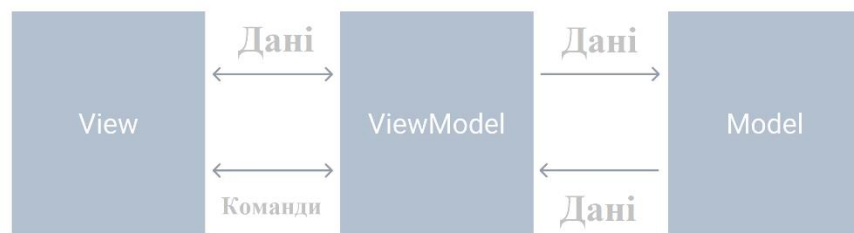


Рисунок 3.1 — Взаємодія між функціональними частинами шаблону MVVM

Внутрішньо Vue компілює шаблони у віртуальні можливості відтворення DOM. У поєднанні з системою інжектора Vue може розумно обчислити кількість компонентів для перемальовування та мінімально маніпулювати DOM під час зміни стану програми.

У Vue ви можете також використовувати синтаксис шаблону або ще писати функції відтворення безпосередньо за допомогою JSX. Для цього треба замінити шаблон функцією візуалізації. Візуалізація дає змогу створювати ефективні шаблони на основі компонентів — наприклад, нова транспортна система тепер повністю заснована на компонентах і використовує внутрішні та можливості візуалізації.

Однією з найвідмітніших особливостей Vue є система дискретного струму. Моделі — це просто плоскі об'єкти JavaScript. Завдяки цьому управління статусом дуже просте та інтуїтивно зрозуміле. Vue забезпечує оптимізоване перемальовування без додаткової роботи. Кожен компонент відстежує свої реактивні залежності під час рендерингу, тому система точно знає, коли відображати та які компоненти рендерити.

Vue надає багато способів застосування ефектів переходу, коли елемент додається, оновлюється або видаляється з DOM. Наприклад:

- автоматизоване застосування класів для CSS переходів і анімацій;
- інтегрування сторонніх бібліотек для CSS анімацій, наприклад таких як `Animate.css`;
- використовувати JavaScript як для прямих маніпуляцій з DOM під час переходів;
- інтегрування різних сторонніх JavaScript бібліотек анімацій, як `Velocity.js` [27].

React — це платформа JavaScript з відкритим кодом, створена Джорданом Валком з Facebook. Компанії доручено вдосконалити процес створення і відтворення інтерфейсів, необхідних Facebook для їх розробки та підтримки. На той час це був величезний проект і існуючі технології були недоступні. React вперше був використаний у стрічці новин Facebook у 2011 році, а також пізніше в Instagram у 2012 році. Тільки в 2013 році весь вихідний код був представлений всій спільноті розробників на американській конференції JSConf. Це було давно із сотнями оновлень, виправлень і доданих функцій, але з тих пір React все ще залишається найпопулярнішим.

У порівнянні з Vue, React має дещо бідну документацію, неможливо перерахувати документацію самої бібліотеки та величезної кількості складних пакетів для неї.

React можна вважати тираном світу веб-фреймворків. Це порушує всі правила. Кращі практики. Якщо завжди вважалося необхідним розділяти теги, стилі та JavaScript в окремі файли, то з появою наприклад JSX, де все може

бути в одному місці, як погляди розробників розділяться. Деякі люди вважають, що це спрощує розробку, дозволяє відразу знайти помилку, тому що React не компілюється, коли чогось не вистачає або є щось зайве. А інші стверджують, що це тільки заплутує і ускладнює весь процес [28].

Для реалізації програмного модуля інтерактивного тестування знань був обраний фреймворк Vue.

3.6 Розробка модуля генерації

Модуль генерації призначений для того, щоб автоматизовано генерувати тест з наявного банку завдань. Він також призначений для створення та редагування тесту, а саме дозволяє встановлювати назву тесту, тему та його налаштування. Він взаємодіє з базою даних.

Модуль виконує такі дії:

- отримання з бази даних та зміна назви тесту в базі даних;
- отримання з бази даних та зміна теми тесту в базі даних;
- налаштування часу на проходження інтерактивного тесту;
- налаштування кількості можливих проходжень інтерактивного тесту;
- налаштування кількості питань, які буде згенеровано автоматизовано за допомогою розробленого алгоритму;
- налаштування кількості питань простого рівня складності;
- налаштування кількості питань середнього рівня складності;
- налаштування кількості питань складного рівня складності;
- налаштування співвідношення практичних і теоретичних завдань;
- налаштування повторення тем завдань інтерактивного тесту;
- автоматизована генерація інтерактивного тесту.

Діаграма активності модуля генерації зображена на рисунку 3.2.



Рисунок 3.2 — Діаграма активності модуля генерації

3.7 Розробка модуля тестування

Модуль тестування використовується для виведення можливих для проходження тестів, а також за процес тестування, а саме за виведення тестових завдань та зчитування відповідей і видачу результатів проходження. Модуль також взаємодіє з базою даних, отримуючи з неї дані.

Модуль має виконувати такі дії:

- отримує список усіх назв тестів з бази даних;
- отримує всі інтерактивні тестові завдання відповідного тесту і також відповіді до них з бази даних;
- виводить інтерактивні тестові завдання;
- запам'ятовує відповіді користувача і зрівнює їх з правильними;
- вкінці виводить результат, де дає детальну інформацію про правильність даних відповідей користувача на всі питання.

Діаграма активності модуля тестування зображена в Додатку В.

3.8 Розробка модуля створення та редагування банку завдань

Даний модуль створення та редагування банку завдань призначений для зміни, створення, видалення інтерактивних тестових завдань. Модуль також взаємодіє з базою даних отримуючи та записуючи дані.

Модуль має виконувати такі дії:

- зміна, створення та видалення тестового запитання;
- встановлення певного рівня складності запитання.
- встановлення певного типу запитання;
- зміна, створення та видалення відповідей на кожне запитання;
- виведення списку інтерактивних тестових завдань.

Діаграма активності модуля створення та редагування банку завдань зображена в Додатку Г.

3.9 Розробка модуля результатів

Модуль результатів призначений для перегляду результатів тестування користувачів. Модуль взаємодіє з базою даних отримуючи дані.

Модуль виконує такі дії:

- виведення списку пройдених тестів;
- виведення користувачів, які пройшли певний тест;
- виведення часу проходження тесту загалом та затрачений час на кожне запитання окремо;
- виведення відповідей користувачів на кожне запитання.

Діаграма активності модуля результатів зображена в Додатку Д.

3.10 Розробка бази даних

Як було визначено, для реалізації мого додатку інтерактивного тестування знань обрано базу даних MySQL.

Розрізняють такі бази даних за моделлю організації:

— ієрархічна — представляється у вигляді дерева, яке складається з об'єктів різних рівнів, між цими об'єктами існують зв'язки «предок-нащадок», у даному випадку можливо, що об'єкт може не мати дочірніх або має декілька, тоді як дочірній постійно повинен мати лише одного предка;

— мережева — є подібною до ієрархічної, але за винятком того, що кожен об'єкт має більше одного предка;

— реляційна — зберігає дані у вигляді таблиць, найбільш поширені СУБД використовують реляційну модель даних;

— об'єктно-орієнтована — у базі даних дані створюються у вигляді моделей об'єктів [29].

Для мого додатку інтерактивного тестування знань була обрана реляційна база даних. Реляційна база даних має значні переваги над іншими:

— простота та доступність для розуміння кожним користувачем, єдиною використовуваною конструкцією є «таблиця»;

— суворі правила проектування, вони базуються на математичному апараті;

— також є повна незалежність даних, зміни в програмі при зміні реляційної БД є мінімальними;

— для організації запитів та написання прикладного ПЗ також немає необхідності знати дану організацію БД у зовнішній пам'яті [30].

База даних мого додатку інтерактивного тестування знань буде містити такі таблиці:

— запитання — буде містити тему запитання, назву запитання, номер запитання, тип запитання, складність запитання;

— відповіді — буде містити номер і відповіді на всі запитання;

— користувачі — буде містити логіни і паролі користувачів;

— адміністратори — буде містити логіни та паролі лише адміністраторів;

— тести — буде містити назву тесту, номер тесту, номери завдань, які належать до цього інтерактивного тесту;

— налаштування тесту — буде містити номер тесту і всі його налаштування;

— час проходження;

— кількість питань кожного рівня складності;

— співвідношення теоретичних і практичних питань;

— налаштування повтору тем тестових завдань.

Даний додаток інтерактивного тестування знань матиме два рівні доступу: адміністратор та користувач.

Адміністратори зможуть створювати, редагувати та видаляти завдання з бази даних, а також змінювати налаштування тесту, назву тесту, створювати, редагувати або видаляти тести і, звичайно, можливість пройти тестування.

Звичайні користувачі матимуть лише можливість пройти тестування і не матимуть можливості змінювати та записувати дані до бази даних.

4 ТЕСТУВАННЯ РОЗРОБЛЕНОГО ДОДАТКУ ІНТЕРАКТИВНОГО ТЕСТУВАННЯ ЗНАНЬ ТА АНАЛІЗ РЕЗУЛЬТАТІВ РОБОТИ

Експериментальні дослідження щодо функціонування програмного модуля інтерактивного тестування знань передбачали дослідження з перевірки таких можливостей:

- автоматизовано генерувати тест із наявного банку завдань;
- керувати складністю завдань тесту;
- встановлення часу на проходження тесту;
- встановлення кількості спроб на проходження тесту;
- створення, зміни та видалення тестових завдань;
- керувати кількістю завдань тесту;
- переглядати результати користувачів;
- виставити співвідношення теоретичних та практичних питань;
- керувати повтореннями тем завдань інтерактивного тесту;
- тест не має містити жодних повторів і відповідати налаштуванням;
- мають бути два рівні доступу: адміністратор та користувач.

Під час запуску додатку відкривається сторінка авторизації з можливістю реєстрації, як можна побачити на рисунку 4.1.

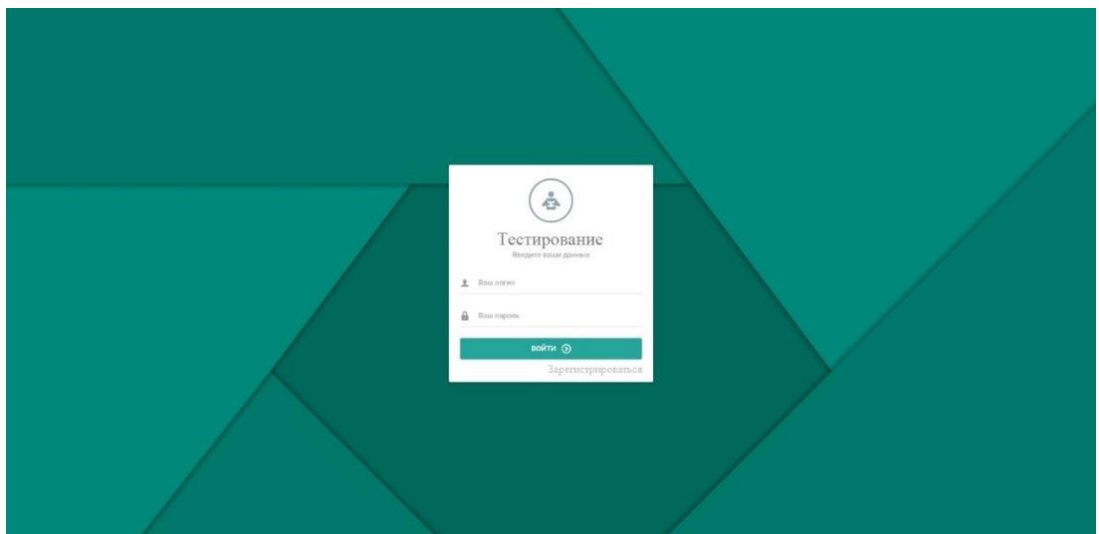


Рисунок 4.1 — Сторінка авторизації

Після авторизації, в залежності від того, який рівень доступу (користувач або адміністратор), відкриється головна сторінка додатку. Вид головної сторінки для адміністратора зображений на рисунку 4.2, для звичайного користувача — на рисунку 4.3.

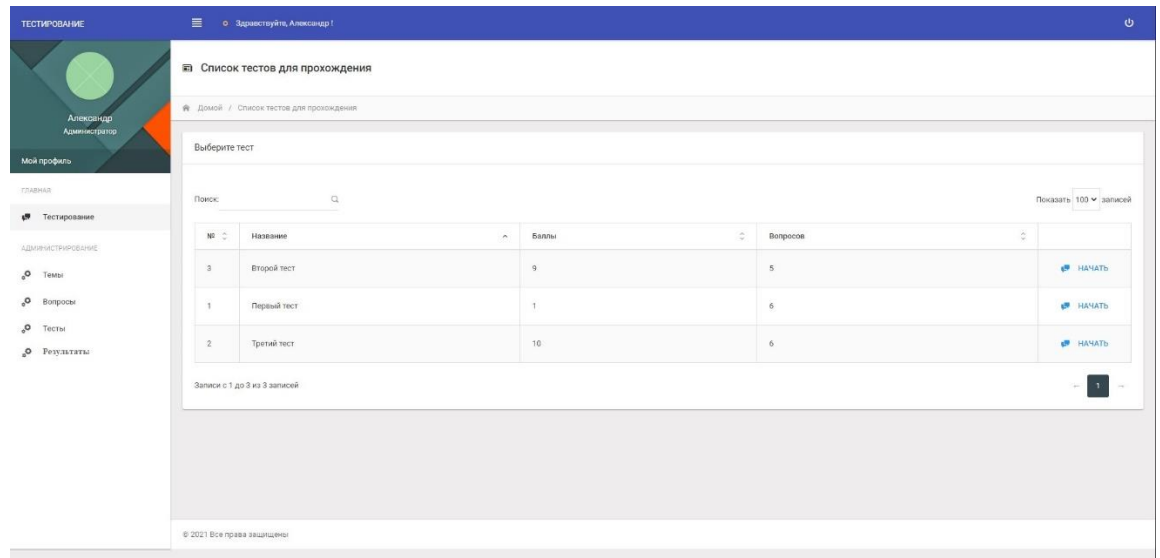


Рисунок 4.2 — Головна сторінка адміністратора

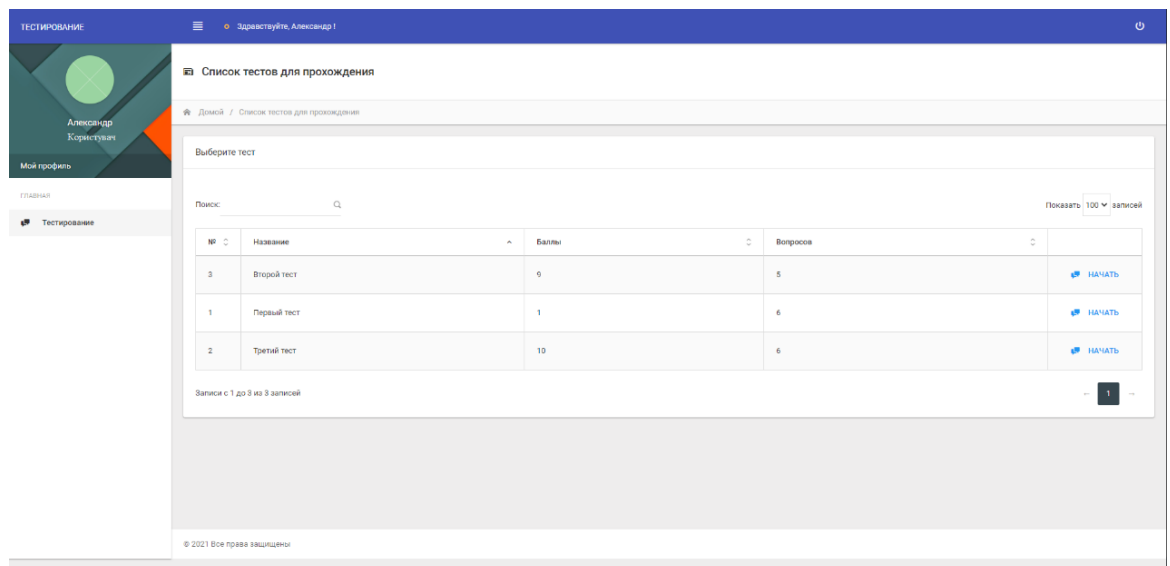


Рисунок 4.3 — Головна сторінка користувача

Можна побачити, що в адміністратора з'являється вкладка «Адміністрування», що дає можливість створювати та редагувати тести, завдання, а також теми та дивитись результати тестування користувачів. Користувачеві доступна лише вкладка «Тестування», він може пройти тест на

вибір. Так реалізовано два рівня доступу до додатку: адміністратор та користувач.

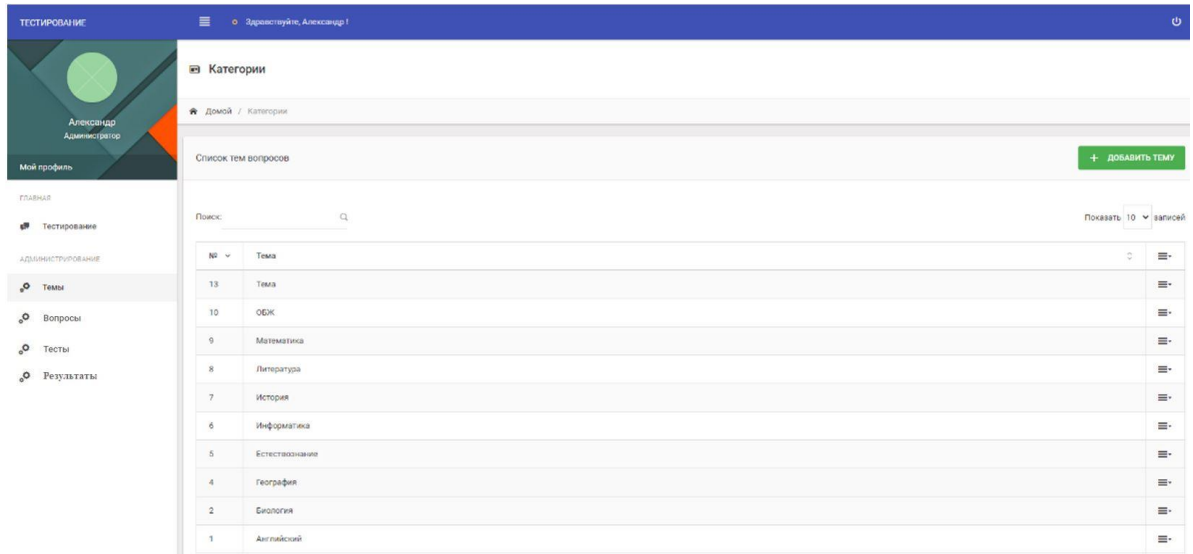


Рисунок 4.4 — Сторінка списку тем

У вкладці «Адміністрування» є чотири вкладки: «теми», «завдання», «результати» та «тести». Якщо натиснути на вкладку «теми» — відкриється список тем завдань з можливістю додавання, редагування і видалення теми, вигляд сторінки можна побачити на рисунку 4.4. Додавання теми можна побачити на рисунку 4.5. Редагування теми можна побачити на рисунку 4.6.

Редагування, створення і видалення теми коректно працює.

При переході на вкладку «Завдання» відкриється сторінка із списком тестових завдань з можливістю редагування, створення і видалення тестових завдань. Сторінка із списком завдань показана на рисунку 4.7, сторінка додавання завдання показана на рисунку 4.8, сторінка редагування завдання — на рисунку 4.9.

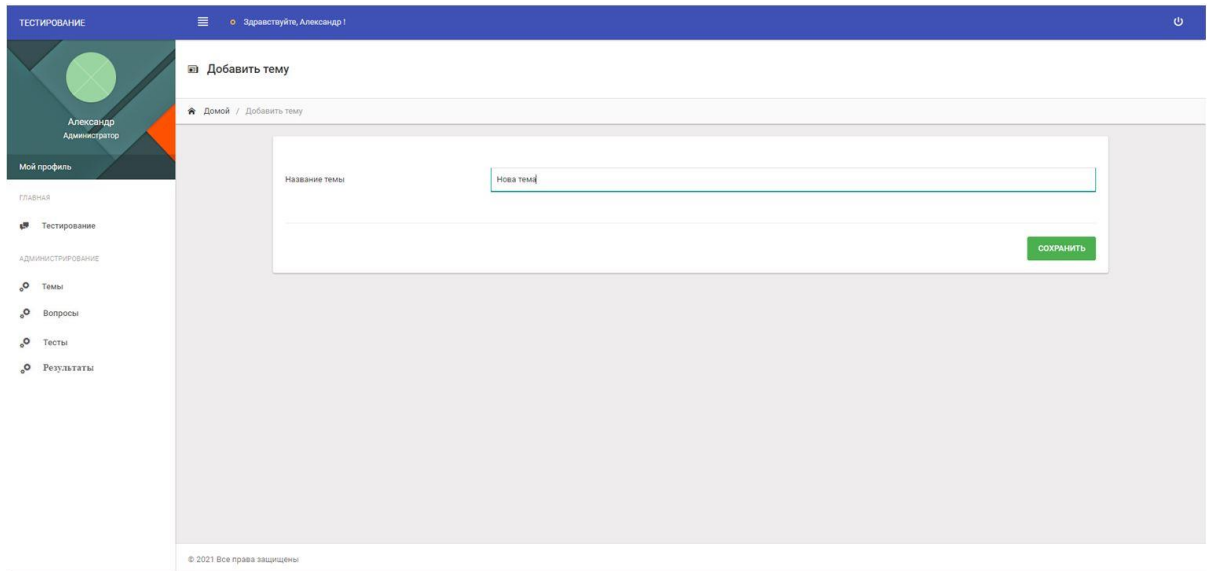


Рисунок 4.5 — Сторінка додавання теми

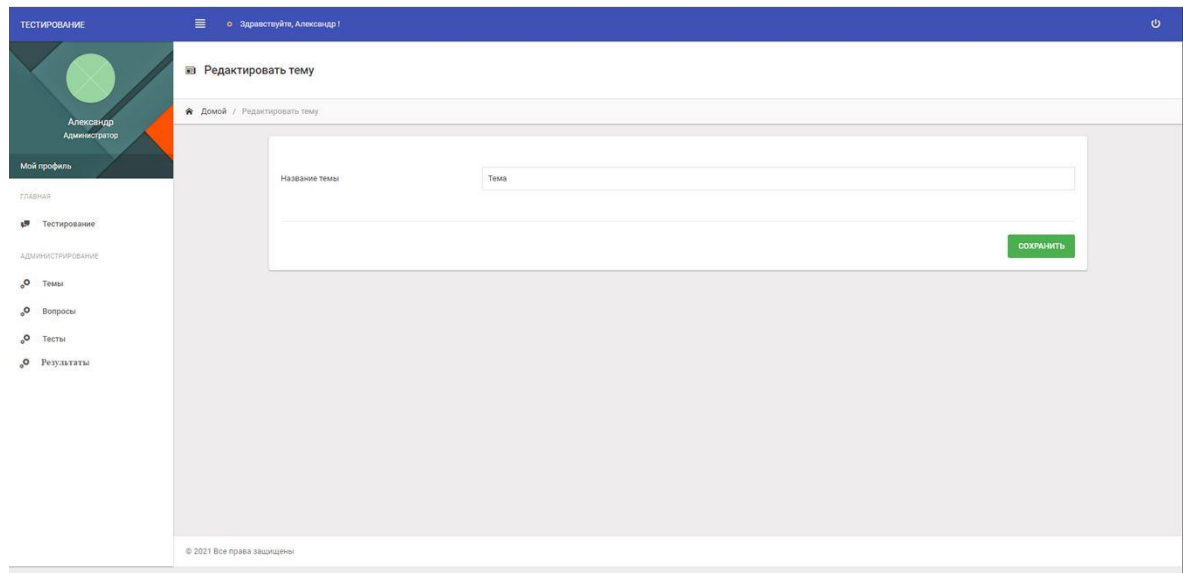


Рисунок 4.6 — Сторінка редагування теми

Можна побачити, що завдання має тему, текст, рівень складності, тип і відповіді на запитання із поміткою правильної відповіді. Редагування, створення і видалення тестового завдання працюють коректно.

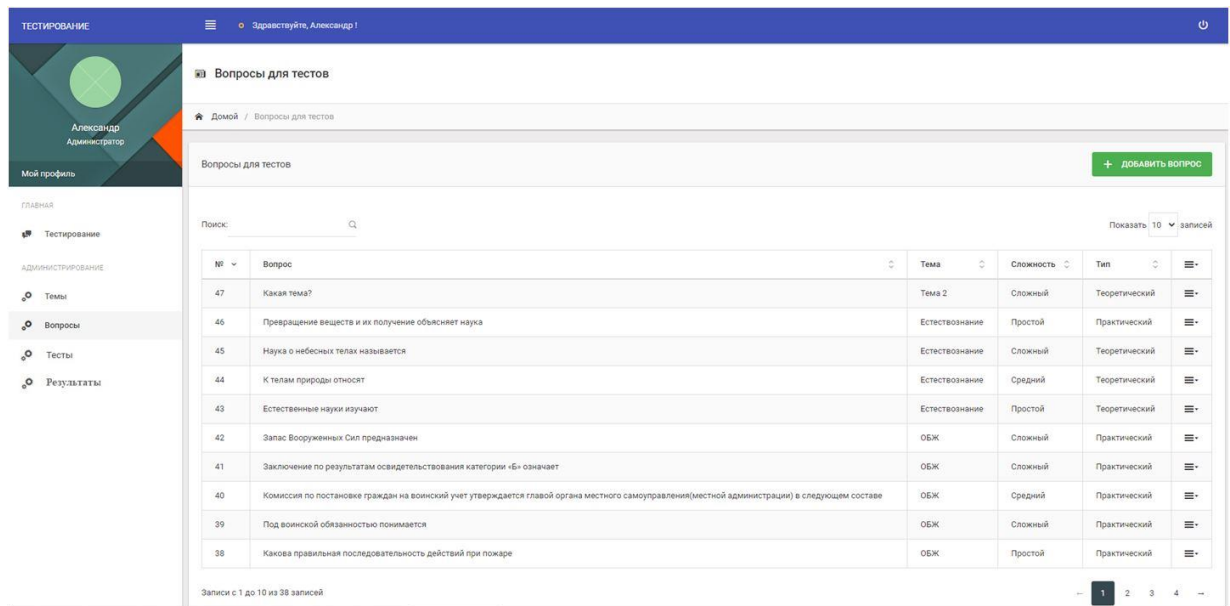


Рисунок 4.7 — Сторінка списку тестових завдань

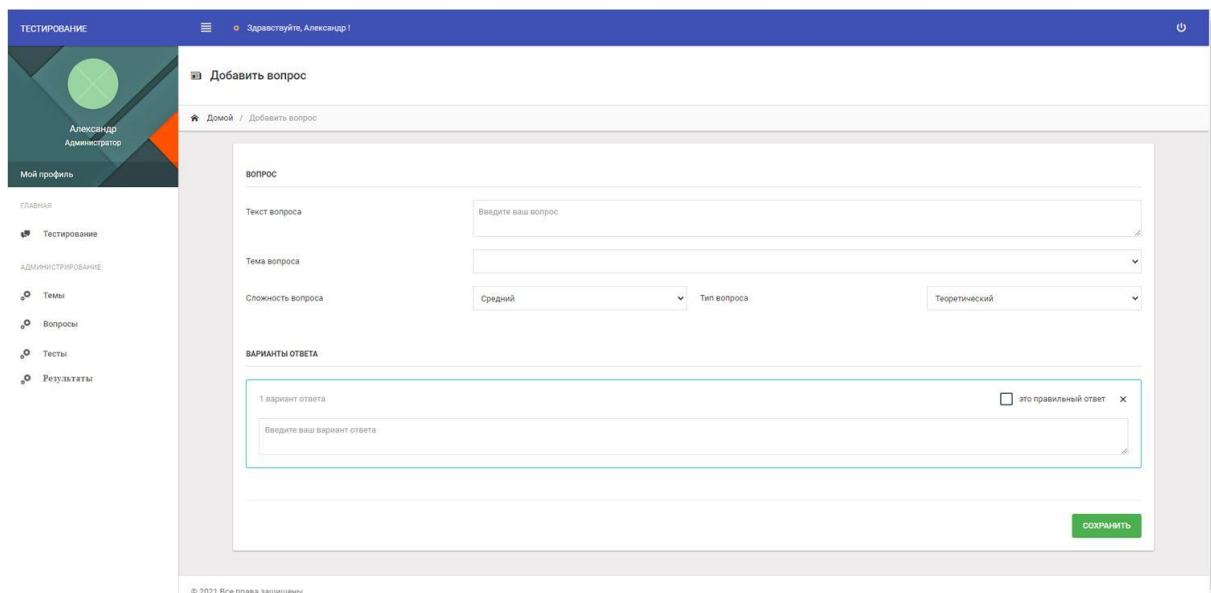


Рисунок 4.8 — Сторінка додавання завдання

Натиснувши на вкладку «тести», відкриється сторінка із списком тестів з можливістю їхнього редагування, створення і видалення. Сторінка списку тем зображена на рисунку 4.10, сторінка створення тесту зображена на рисунку 4.11, сторінка редагування — на рисунку 4.12.

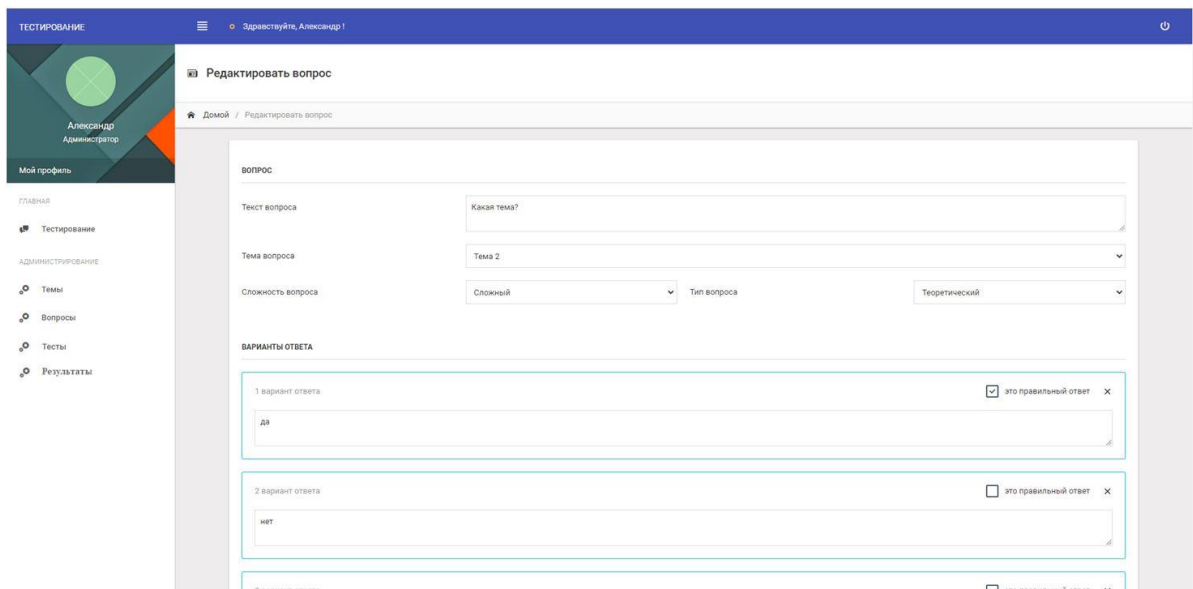


Рисунок 4.9 — Сторінка редагування завдання

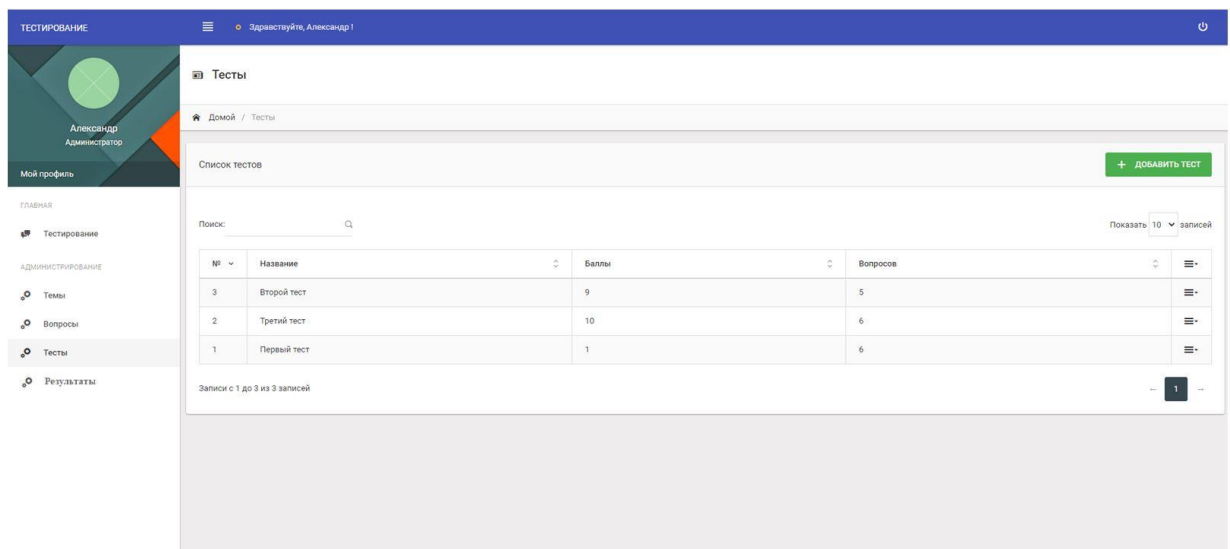


Рисунок 4.10 — Сторінка списку тестів

Якщо натиснути на вкладку «результати», після чого вибрати відповідний тест, то відкриється список користувачів, які пройшли даний тест. Далі потрібно вибрати певного користувача і відкриються результати проходження тесту як показано на рисунку 4.13.

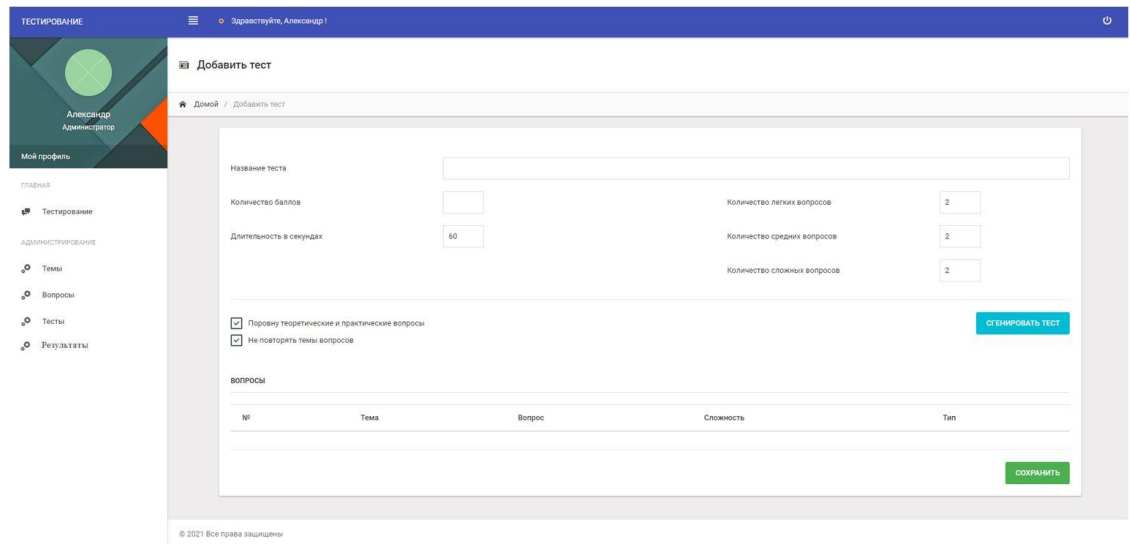


Рисунок 4.11 — Сторінка додавання тесту

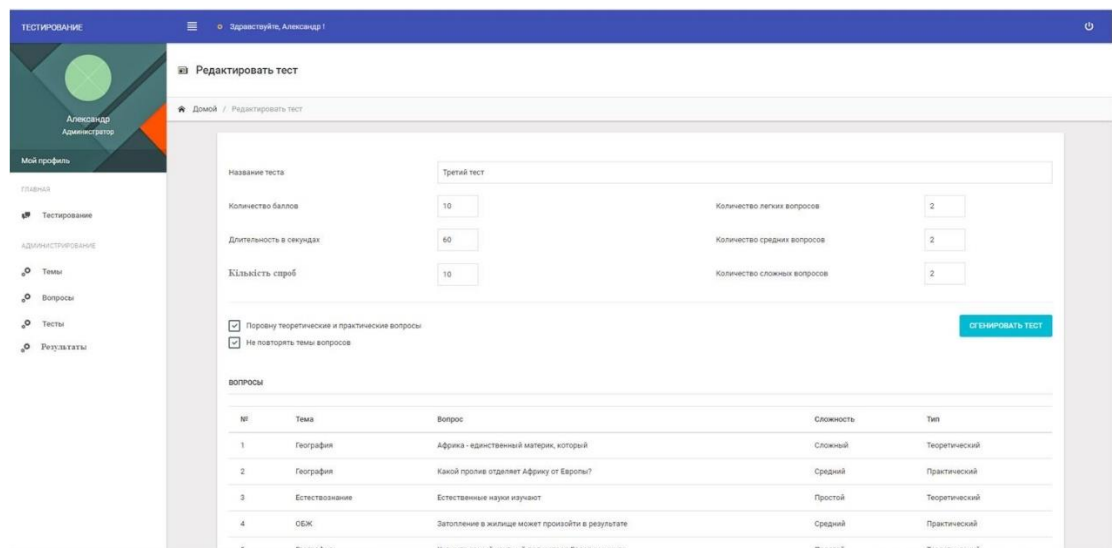


Рисунок 4.12 — Сторінка редагування тесту

Можна побачити, що кожен тест має назву та свої налаштування, а саме: кількість питань кожного рівня складності, час проходження, а також співвідношення теоретичних та практичних питань і можливість керувати повтореннями тем завдань. Редагування, створення і видалення тесту працюють коректно. Також автоматизовано згенеровано тест. Генерація працює коректно, відповідаючи до налаштуванням.

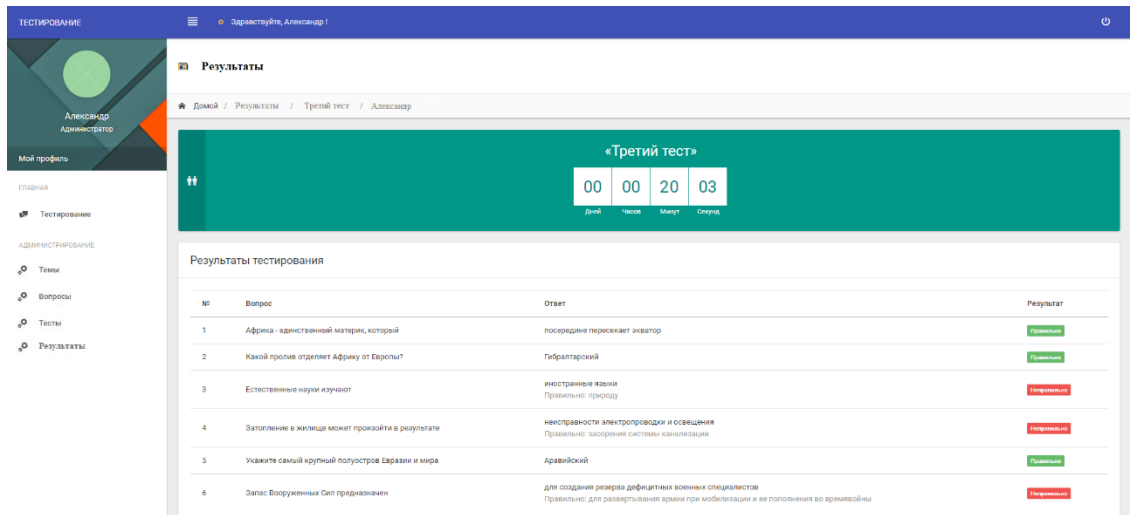


Рисунок 4.13 — Сторінка результату проходження тесту певним користувачем

При переході на вкладку «тестування» відкриється сторінка із списком тестів, їх можна пройти. Сторінка списку тестів показана на рисунку 4.14. Після натискання кнопки «почати» почнеться виведення завдань та відповідей, тому користувачеві потрібно обирати відповідь, після чого відбудеться перехід до другого завдання, поки всі вони не закінчаться або не завершиться час даний на проходження.

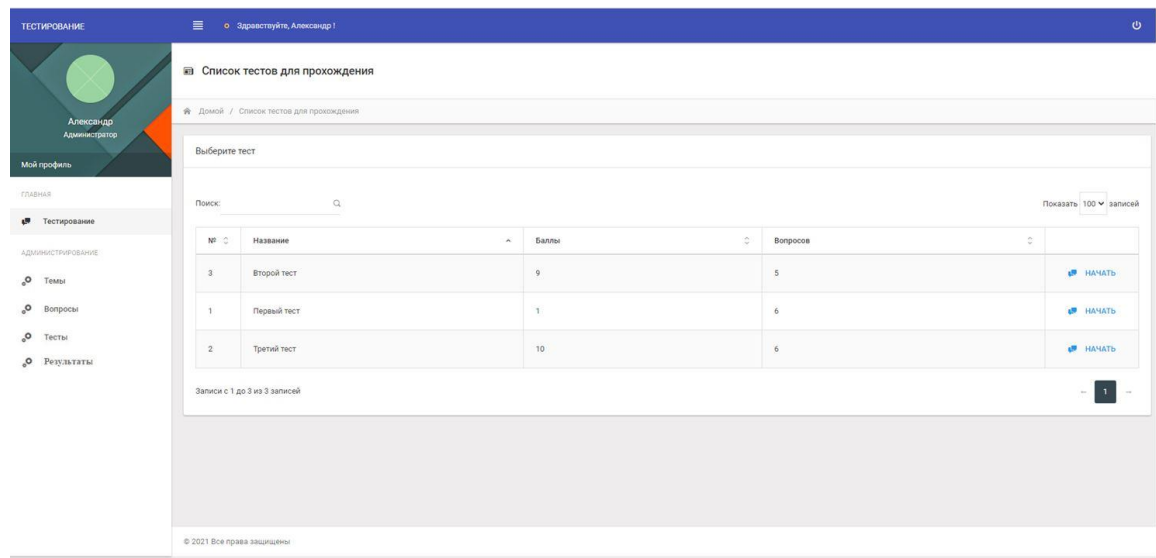


Рисунок 4.14 — Сторінка списку тестів для проходження

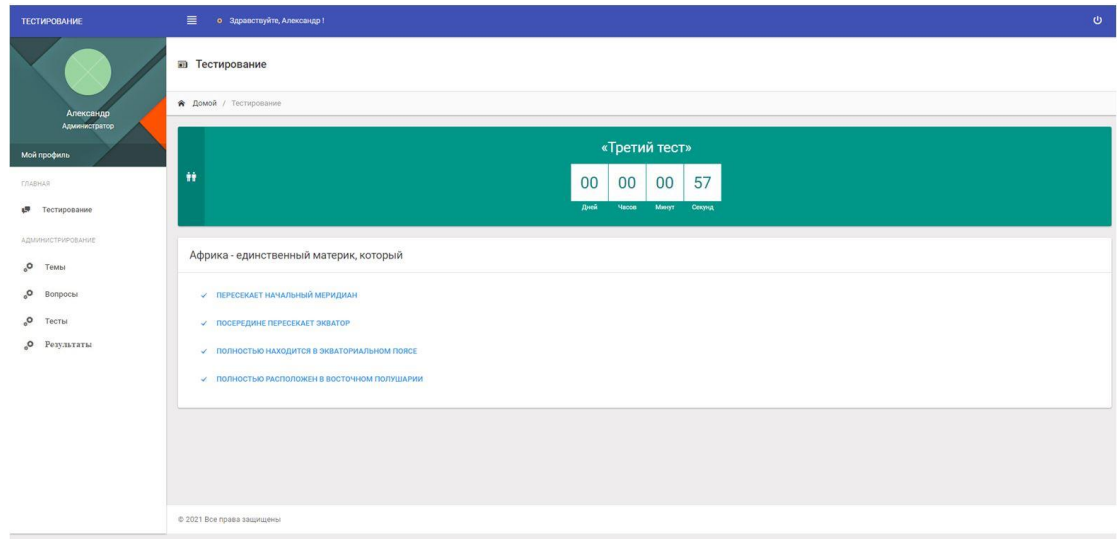
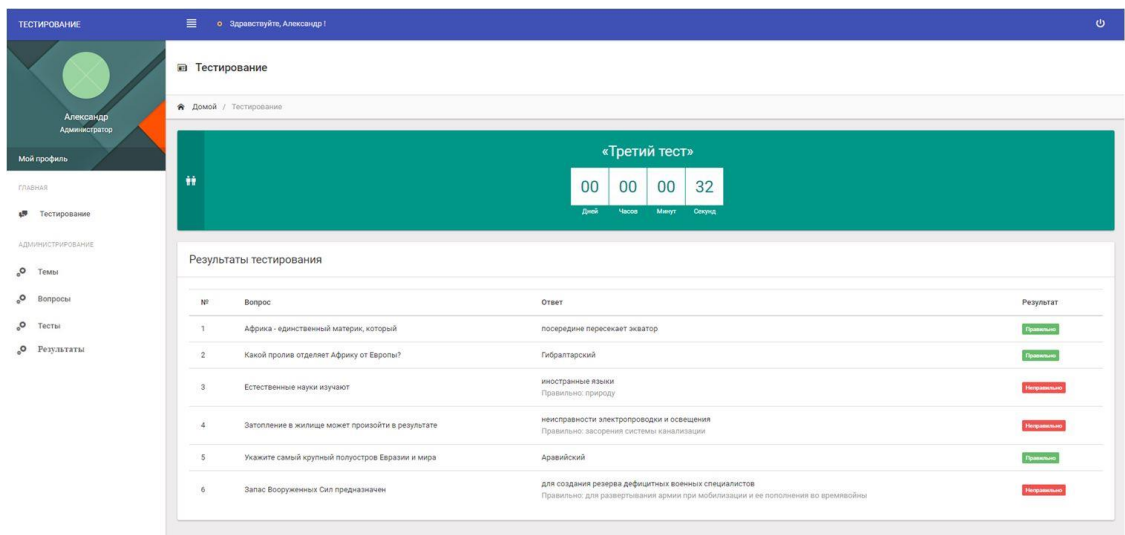


Рисунок 4.15 — Процесс прохождения тесту



Сторінка проходження тесту зображена на рисунку 4.15. Після завершення інтерактивного тесту буде відкрита сторінка результатів, яка зображена на рисунку 4.16, на якій користувач зможе побачити правильність своїх відповідей.

За результатами проведених досліджень в середньому отримано такі значення характеристик програмного модуля інтерактивного тестування знань порівняно з додатком classmarker, як показано в таблиці 4.1.

Таблиця 4.1 — Порівняння характеристик програмного модуля інтерактивного тестування знань та додатку classmarker.

	Можливість автоматизованої генерації тесту	Можливість керувати складністю завдань	Зміна співвідношення теоретичних та практичних завдань	Можливість керувати повторенням тем завдань в тесті
Програмний модуль інтерактивного тестування знань	Є	Є	Є	Є
classmarker	Є	Немає	Немає	Немає

5 РОЗРАХУНОК ЕКОНОМІЧНОЇ ДОЦІЛЬНОСТІ СТВОРЕННЯ ПРОГРАМНОГО МОДУЛЯ ІНТЕРАКТИВНОГО ТЕСТУВАННЯ ЗНАНЬ

5.1 Комерційний та технологічний аудит науково-технічної розробки

Метою даного розділу є проведення технологічного аудиту, в даному випадку нового програмного модулю інтерактивного тестування знань. У сучасних умовах професійно-технічної освіти в Україні набуває значущості можливість автоматизованого формування тесту, можливість автоматизованої перевірки результатів тестування, підвищення об'єктивності оцінювання та зменшення упередженості при оцінюванні знань.

Особливістю програми є те, що дана технологія дає можливість автоматизованого формування тесту, можливість автоматизованої перевірки результатів тестування, підвищення об'єктивності оцінювання та зменшення упередженості при оцінюванні знань

Аналогом може бути Classmarker — ціна від 9600 грн/рік.

Для проведення комерційного та технологічного аудиту залучають не менше 3-х незалежних експертів. Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням п'ятибальної системи оцінювання за 12-ма критеріями, у відповідності із таблиці 5.1.

Таблиця 5.1 — Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Бали (за 5—ти бальною шкалою)					
Кри-терій	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах

Продовження табл. 5.1

Ринкові переваги					
2	Багато аналогів на малому ринку	Ринкові п Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практик на здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так із комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово—промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві

Закінчення таблиці 5.1

11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10—ти років	Термін реалізації ідеї від 3—х до 5—ти років. Термін окупності інвестицій більше 5—ти років	Термін реалізації ідеї менше 3—х років. Термін окупності інвестицій від 3—х до 5—ти років	Термін реалізації ідеї менше 3—х років. Термін окупності інвестицій менше 3—х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь—які регламентні обмеження на виробництво та реалізацію продукту

Усі дані по кожному параметру занесено в таблиці 5.2

Таблиця 5.2 — Результати оцінювання комерційного потенціалу розробки

Критерії оцінювання	ШБ експертів		
	Експерт 1	Експерт 2	Експерт 3
	Бали		
Технічна здійсненність концепції	3	3	4
Наявність аналогів на ринку	3	3	4
Цінова політика	3	4	3
Технічні та споживчі властивості виробу	4	3	4
Експлуатаційні витрати	3	4	3
Ринок збуту	4	3	4
Конкурентоспроможність	3	4	3
Фахівці з технічної і комерційної реалізації	4	3	4
Фінансування	4	4	3
Матеріально-технічна база	3	3	3
Термін реалізації ідеї	4	3	3
Супровідна документація	3	3	4
Сума	41	40	42
Середньоарифметична сума балів	$(41+40+42) / 3 = 41$		

За даними таблиці 5.2 можна зробити висновок щодо рівня комерційного потенціалу даної розробки. Для цього доцільно скористатись рекомендаціями, наведеними в таблиці 5.3.

Таблиця 5.3 — Рівні комерційного потенціалу розробки

Середньоарифметична сума балів СБ, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 — 10	Низький
11 — 20	Нижче середнього
21 — 30	Середній
31 — 40	Вище середнього
41 — 48	Високий

Як видно з таблиці, рівень комерційного потенціалу розроблюваного нового програмного продукту є високим, що досягається за рахунок того, що програмний продукт надає розширення функціональних можливостей при генерації тестів для інтерактивного тестування знань, а саме можливість автоматизованого формування тесту, можливість автоматизованої перевірки результатів тестування, підвищення об'єктивності оцінювання та зменшення упередженості при оцінюванні знань.

5.2 Розрахунок витрат на здійснення науково-дослідної роботи

Основна заробітна плата розробників, яка розраховується за формулою:

$$Z_o = \frac{M}{T_p} \cdot t, \quad (5.1)$$

де M — місячний посадовий оклад конкретного розробника (дослідника), грн.;

T_p — число робочих днів в місяці, 21 днів;

t — число днів роботи розробника (дослідника).

Результати розрахунків зведемо до таблиці 5.4.

Так як в даному випадку розробляється програмний продукт, то розробник виступає одночасно і основним робітником, і тестувальником розроблюваного програмного продукту.

Таблиця 5.4 — Основна заробітна плата розробників

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник проекту	42000	2000,00	28	56000,000
Програміст	35000	1666,67	28	46666,667
Всього				102666,67

Додаткову заробітну плату прийнято розраховувати як 13,6 % від основної заробітної плати розробників та робітників:

$$Z_d = Z_o \cdot 13,6 \% / 100 \% \quad (5.2)$$

$$Z_d = (102666,67 \cdot 13,6 \% / 100 \%) = 13962,67 \text{ (грн.)}$$

Згідно діючого законодавства нарахування на заробітну плату складають 22 % від суми основної та додаткової заробітної плати.

$$H_z = (Z_o + Z_d) \cdot 22 \% / 100\% \quad (5.3)$$

$$H_z = (102666,67 + 13962,67) \cdot 22 \% / 100 \% = 25658,45 \text{ (грн.)}$$

Оскільки для розроблювального пристрою не потрібно витратити матеріали та комплектуючі, то витрати на матеріали і комплектуючі дорівнюють нулю.

Амортизація обладнання, що використовувалось для розробки в спрощеному вигляді амортизація обладнання, що використовувалась для розробки розраховується за формулою:

$$A = \frac{Ц}{T} \cdot \frac{t_{\text{вик}}}{12} \text{ [грн.]} \quad (5.4)$$

де Ц — балансова вартість обладнання, грн.;

T — термін корисного використання обладнання згідно податкового законодавства, років

$t_{\text{вик}}$ — термін використання під час розробки, місяців

Розрахуємо, для прикладу, амортизаційні витрати на комп'ютер балансова вартість якого становить 27000 грн., термін його корисного використання згідно податкового законодавства — 2 роки, а термін його фактичного використання — 1,33 міс.

$$A_{\text{обл}} = \frac{27000}{2} \times \frac{1,33}{12} = 1500 \text{ грн.}$$

Аналогічно визначаємо амортизаційні витрати на інше обладнання та приміщення. Розрахунки заносимо до таблиці 4.2.

Так як вартість ліцензійної ОС та спеціалізованих ліцензійних нематеріальних ресурсів безкоштовна (Webstorm), то $B_{\text{нем.ак.}} = 0$ грн.

Таблиця 5.5 — Амортизаційні відрахування матеріальних і нематеріальних ресурсів для розробників

Найменування обладнання	Балансова вартість, грн.	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн.
Комп'ютер та комп'ютерна периферія (HP ProBook)	27000	2	1,33	1500,000
Офісне обладнання (меблі)	20000	4	1,33	555,556
Приміщення	980000	20	1,33	5444,444
Всього				7500,00

Тарифи на електроенергію для непобутових споживачів (промислових підприємств) відрізняються від тарифів на електроенергію для населення. При цьому тарифи на розподіл електроенергії у різних постачальників (енергорозподільних компаній), будуть різними. Крім того, розмір тарифу

залежить від класу напруги (1-й або 2-й клас). Тарифи на розподіл електроенергії для всіх енергорозподільних компаній встановлює Національна комісія з регулювання енергетики і комунальних послуг (НКРЕКП). Витрати на силову електроенергію розраховуються за формулою:

$$V_e = V \cdot \Pi \cdot \Phi \cdot K_{\Pi}, \quad (5.5)$$

де V — вартість 1 кВт-години електроенергії для 1 класу підприємства, $V = 6,2$ грн./кВт;

Π — встановлена потужність обладнання, кВт. $\Pi = 0,35$ кВт;

Φ — фактична кількість годин роботи обладнання, годин.

K_{Π} — коефіцієнт використання потужності, $K_{\Pi} = 0,9$.

$$V_e = 0,9 \cdot 0,35 \cdot 8 \cdot 28 \cdot 6,2 = 437,472 \text{ (грн.)}$$

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками. Витрати за статтею «Інші витрати» розраховуються як 50...100% від суми основної заробітної плати дослідників:

$$I_e = (3_o + 3_p) \cdot \frac{H_{ib}}{100\%}, \quad (5.6)$$

де H_{ib} — норма нарахування за статтею «Інші витрати».

$$I_b = 102666,67 * 60\% / 100\% = 61600 \text{ (грн.)}$$

До статті «Накладні (загальнопромислові) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін. Витрати за статтею «Накладні

(загальновиробничі) витрати» розраховуються як 100...150% від суми основної заробітної плати дослідників:

$$H_{нзв} = (З_o + З_p) \cdot \frac{H_{нзв}}{100\%}, \quad (5.7)$$

де $H_{нзв}$ — норма нарахування за статтею «Накладні (загальновиробничі) витрати».

$$H_{нзв} = 102666,67 * 120 \% / 100 \% = 123200 \text{ (грн.)}$$

Сума всіх попередніх статей витрат дає загальні витрати на проведення науково-дослідної роботи:

$$B_{заг} = 102666,67 + 13962,67 + 25658,45 + 7500,00 + 437,47 + 61600 + \\ + 123200 = 335025,26 \text{ грн.}$$

Загальні витрати на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються ЗВ, визначається за формулою:

$$ЗВ = \frac{B_{заг}}{\eta} \text{ (грн)}, \quad (5.8)$$

де η — коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи.

Так, якщо науково—технічна розробка знаходиться на стадії: науково-дослідних робіт, то $\eta=0,1$; технічного проектування, то $\eta=0,2$; розробки конструкторської документації, то $\eta=0,3$; розробки технологій, то $\eta=0,4$; розробки дослідного зразка, то $\eta=0,5$; розробки промислового зразка, то $\eta=0,7$; впровадження, то $\eta=0,9$. Оберемо $\eta = 0,5$, так як розробка, на даний момент, знаходиться на стадії дослідного зразка:

$$ЗВ = 335025,26 / 0,5 = 670051 \text{ грн.}$$

5.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

В ринкових умовах узагальнювальним позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів цієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку. Саме зростання чистого прибутку забезпечить потенційному інвестору надходження додаткових коштів, дозволить покращити фінансові результати його діяльності, підвищить конкурентоспроможність та може позитивно вплинути на ухвалення рішення щодо комерціалізації цієї розробки.

Для того, щоб розрахувати можливе зростання чистого прибутку у потенційного інвестора від можливого впровадження науково-технічної розробки необхідно:

- вказати, з якого часу можуть бути впроваджені результати науково-технічної розробки;
- зазначити, протягом скількох років після впровадження цієї науково-технічної розробки очікуються основні позитивні результати для потенційного інвестора (наприклад, протягом 3-х років після її впровадження);
- кількісно оцінити величину існуючого та майбутнього попиту на цю або аналогічні чи подібні науково-технічні розробки та назвати основних суб'єктів (зацікавлених осіб) цього попиту;
- визначити ціну реалізації на ринку науково-технічних розробок з аналогічними чи подібними функціями.

При розрахунку економічної ефективності потрібно обов'язково враховувати зміну вартості грошей у часі, оскільки від вкладення інвестицій до отримання прибутку минає чимало часу. При оцінюванні ефективності інноваційних проектів передбачається розрахунок таких важливих показників:

- абсолютного економічного ефекту (чистого дисконтованого доходу);
- внутрішньої економічної дохідності (внутрішньої норми дохідності);
- терміну окупності (дисконтованого терміну окупності).

Аналізуючи напрямки проведення науково-технічних розробок, розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором можна об'єднати, враховуючи визначені ситуації з відповідними умовами.

При розробці чи суттєвому вдосконаленні програмного продукту масовим споживачем майбутній економічний ефект буде формуватися на основі таких даних:

$$\Delta\Pi_i = (\pm\Delta\Pi_0 \cdot N + \Pi_0 \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\vartheta}{100}\right), \quad (5.10)$$

де $\pm\Delta\Pi_0$ — зміна вартості програмного продукту (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу;

N — кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки;

Π_0 — основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки, $\Pi_0 = \Pi_6 \pm \Delta\Pi_0$;

Π_6 — вартість програмного продукту у році до впровадження результатів розробки;

ΔN — збільшення кількості споживачів продукту, в аналізовані періоди часу, від покращення його певних характеристик;

λ — коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%, а коефіцієнт $\lambda = 0,8333$.

ρ — коефіцієнт, який враховує рентабельність продукту;

ϑ — ставка податку на прибуток, у 2022 році $\vartheta = 18\%$.

Припустимо, що при прогнозованій ціні 5500 грн. за одиницю виробу, термін збільшення прибутку складе 3 роки. Після завершення розробки і її вдосконалення, можна буде підняти її ціну на 500 грн. Кількість одиниць

реалізованої продукції також збільшиться: протягом першого року — на 750 шт., протягом другого року — на 600 шт., протягом третього року на 500 шт. До моменту впровадження результатів наукової розробки реалізації продукту не було:

$$\Delta\Pi_1 = (0*500 + (5500 + 500)*750)*0,8333*0,45) * (1 - 0,18) = 1268437,449 \text{ грн.}$$

$$\Delta\Pi_2 = (0*500 + (5500 + 500)*(750+600)*0,8333*0,45) * (1 - 0,18) = 2490749,900 \text{ грн.}$$

$$\Delta\Pi_3 = (0*500 + (5500 + 500)*(750+600+500)*0,8333*0,45) * (1 - 0,18) = 3413249,863 \text{ грн.}$$

Отже, комерційний ефект від реалізації результатів розробки за три роки складе 7172437,21 грн.

Далі потрібно розрахувати ефективність вкладених інвестицій та періоду їх окупності. Розраховуємо приведену вартість збільшення всіх чистих прибутків ПП, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково—технічної розробки:

$$ПП = \sum_1^T \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (5.11)$$

де $\Delta\Pi_i$ — збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої науково—дослідної (науково—технічної) роботи, грн;

T — період часу, протягом якою виявляються результати впровадженої науково—дослідної (науково-технічної) роботи, роки;

τ — ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,05 \dots 0,15$;

t — період часу (в роках).

Збільшення прибутку ми отримаємо починаючи з першого року:

$$\begin{aligned} ПП = & \\ & (1268437,449/(1+0,1)^1) + (2490749,900/(1+0,1)^2) + (3413249,863/(1+0,1)^3) = \\ & 1153124,95 + 2058470,992 + 2564425,142 = 5776021,088 \text{ грн.} \end{aligned}$$

Далі розраховують величину початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{\text{інв}} * ЗВ, \quad (5.12)$$

де $k_{\text{інв}}$ — коефіцієнт, що враховує витрати інвестора на впровадження науково—технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай $k_{\text{інв}}=2\dots5$, але може бути і більшим;

$ЗВ$ — загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

$$PV = 2 * 670051 = 1340101,03 \text{ грн.}$$

Тоді абсолютний економічний ефект $E_{\text{абс}}$ або чистий приведений дохід (NPV, Net Present Value) для потенційного інвестора від можливого впровадження та комерціалізації науково—технічної розробки становитиме:

$$E_{\text{абс}} = \text{ПП} - PV, \quad (5.13)$$

$$E_{\text{абс}} = 5776021,088 - 1340101,03 = 4435920,05 \text{ грн.}$$

Оскільки $E_{\text{абс}} > 0$ то вкладання коштів на виконання та впровадження результатів даної науково-дослідної (науково-технічної) роботи може бути доцільним.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність або показник внутрішньої норми дохідності (IRR, Internal Rate of Return) вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій E_6 . Для цього використаємо формулу:

$$E_6 = T_{жс} \sqrt[3]{1 + \frac{E_{абс}}{PV}} - 1, \quad (5.14)$$

$T_{жс}$ – життєвий цикл наукової розробки, роки.

$$E_6 = \sqrt[3]{(1 + 4435920,05/1340101,03)} - 1 = 0,627$$

Визначимо мінімальну ставку дисконтування, яка у загальному вигляді визначається за формулою:

$$\tau = d + f, \quad (5.15)$$

де d — середньозважена ставка за депозитними операціями в комерційних банках; в 2022 році в Україні $d = (0,09...0,14)$;

f – показник, що характеризує ризикованість вкладень; зазвичай, величина $f = (0,05...0,5)$.

$$\tau_{\min} = 0,14 + 0,05 = 0,19.$$

Так як $E_6 > \tau_{\min}$, то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Розрахуємо термін окупності вкладених у реалізацію наукового проекту інвестицій за формулою:

$$T_{ок} = \frac{1}{E_6}, \quad (5.16)$$

$$T_{ок} = 1 / 0,627 = 1,59 \text{ р.}$$

Оскільки $T_{ок} < 3$ -х років, а саме термін окупності рівний 1,59 роки, то фінансування даної наукової розробки є доцільним.

ВИСНОВКИ

У магістерській кваліфікаційній роботі розроблено програмний модуль інтерактивного тестування знань, який надає можливість автоматизованого формування тесту, можливість автоматизованої перевірки результатів тестування, підвищення об'єктивності оцінювання та зменшення упередженості при оцінюванні знань.

У першому розділі магістерської роботи було проведено аналіз предметної області для створення системи. Розглянуто переваги та недоліки використання тестування як способу контролю та діагностики знань, сучасні вимоги до тестування знань. Проведено порівняння сучасних засобів тестування знань та виявлено їх переваги та недоліки. Розглянуто моделі навчання та проведено їх аналіз та визначено їх переваги та недоліки. Проаналізовано переваги та недоліки інтерактивного тестування. Проведено аналіз методів генерації тесту, визначено їх переваги та недоліки.

У другому розділі магістерської роботи розроблено алгоритми інтерактивного тестування та генерації тесту та наведено їх кроки та діаграми активності.

У третьому розділі магістерської роботи визначені функції програмного модуля інтерактивного тестування знань. Визначено модулі, які виконують дані функції. Створено схему взаємодії основних складових програмного модуля інтерактивного тестування знань. Обґрунтовано вибір мови програмування, середовища розробки, бази даних, фреймворку. Розроблено базу даних та головні модулі. Наведено діаграми активності модулів.

У четвертому розділі магістерської роботи було протестовано можливості програмного модуля інтерактивного тестування знань, а саме: створення, зміна та видалення тем, завдань, тестів та перегляд результатів користувачів. Всі функції працюють коректно. Протестовано автоматизовану генерацію тесту відносно налаштувань. Тест генерується коректно,

відповідаючи своїм налаштуванням. Також пройдено тести зі списку. Всі тести працюють коректно та видають результат.

У п'ятому розділі магістерської роботи було виконано розрахунок економічної доцільності створення програмного модуля інтерактивного тестування знань, проведено технологічний аудит для науково—технічної розробки, здійснено розрахунок потрібних економічних затрат (на оплату праці, відрахування на заробітню плату, сировину та матеріали амортизацію обладнання програмних засобів, на електроенергію), що необхідні для реалізації додатку. Також проведено порівняння з аналогами та визначення економічної ефективності розробки за її впровадження безпосередньо розробником (замовником).

Розроблений додаток є безкоштовним та надійним, що робить його конкурентоспроможним за рахунок:

- ціни та якості;
- надійності;
- швидкодії;
- простоті інтерфейсу.

Програмний модуль інтерактивного тестування знань у повній мірі придатний до використання і може бути використаний в освіті, з метою діагностики, перевірки знань та з метою досягнення об'єктивності і зменшення упередженості під час оцінювання. Додаток, також може бути вдосконалений при потребі. Результати розробки опубліковані у вигляді наукової статті.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. О. С. Герасик, Л.А. Савицька, М. Г. Тарновський ПРОГРАМНИЙ МОДУЛЬ ІНТЕРАКТИВНОГО ТЕСТУВАННЯ ЗНАНЬ. Інформаційні технології та комп'ютерна інженерія. м. Вінниця - 2023. №1, том 56.
2. Форми контролю знань. [Електронний ресурс]. Режим доступу: https://pidru4niki.com/1880041355088/pedagogika/formi_kontrolyu
3. Тест. [Електронний ресурс]. Режим доступу: <https://ua.kursoviks.com.ua/naukovi—roboty/testi>
4. Використання тестування. [Електронний ресурс]. Режим доступу: http://www.rusnauka.com/11_EISN_2008/Philologia/30455.doc.htm
5. Переваги тестування. [Електронний ресурс]. Режим доступу: https://vuzlit.com/1037136/perevagi_nedoliki_testuvannya
6. Недоліки тестування. [Електронний ресурс]. Режим доступу: <http://www.edudirect.net/sopids—1556—1.html>
7. Тестування як спосіб контролю знань. [Електронний ресурс]. Режим доступу: <http://oldconf.neasmo.org.ua/node/767>
8. Сучасні вимоги до тестування. [Електронний ресурс]. Режим доступу: https://knowledge.allbest.ru/pedagogics/2c0a65635a3ac68a5c53a98521216c27_0.html
9. Види моделей навчання. [Електронний ресурс]. Режим доступу: https://pidru4niki.com/10140508/psihologiya/modeli_navchannya
10. Пасивна модель навчання, переваги та недоліки. [Електронний ресурс]. Режим доступу: http://slovogray.blogspot.com/2016/02/blog—post_11.html
11. Активна модель навчання, переваги та недоліки. [Електронний ресурс]. Режим доступу: <https://clincasequest.academy/learning—models/>
12. Інтерактивна модель навчання, переваги та недоліки. [Електронний ресурс]. Режим доступу: <https://psychologist.net.ua/interaktyvne—navchannya/>

13. Інтерактивне тестування, переваги та недоліки. [Електронний ресурс]. Режим доступу: <http://www.smartboard.com.ua/articles/130/>
14. Методи та засоби автоматичної генерації тестових завдань. [Електронний ресурс]. Режим доступу: <https://svitppt.com.ua/rizne/metodi—ta—zasobi—avtomatichnoi—generacii—testovih—zavdan.html>
15. Метод параметричних тестів. [Електронний ресурс]. Режим доступу: <http://dspace.wunu.edu.ua/bitstream/316497>
16. Метод семантичних мереж. [Електронний ресурс]. Режим доступу: <http://ki.lutsk—ntu.com.ua/node/118>
17. Метод генерації тесту на основі понятійно—тезисної моделі. [Електронний ресурс]. Режим доступу: https://www.researchgate.net/publication/332470453_Avtomatizacii_postroenia_testovyh_zadanih_v_sistemah_distancionnogo_obucenia_na_osnove_ponatijno—tezisnoj_modeli
18. Переваги та недоліки веб—додачків. [Електронний ресурс]. Режим доступу: <http://sites.znu.edu.ua/webprog/lect/1191.ukr.html>
19. Classmarker. [Електронний ресурс]. Режим доступу: <https://www.classmarker.com/online—testing/faq/>
20. Online test pad. [Електронний ресурс]. Режим доступу: <https://onlinetestpad.com>
21. Learningapps. [Електронний ресурс]. Режим доступу: <https://learningapps.org>
22. Webstorm. [Електронний ресурс]. Режим доступу: <https://www.jetbrains.com/ru—ru/webstorm/>
23. Atom. [Електронний ресурс]. Режим доступу: <https://atom.io/>
24. JavaScript, переваги та недоліки. [Електронний ресурс]. Режим доступу: <https://developer.mozilla.org/ru/docs/Web/JavaScript>
25. C#, переваги та недоліки. [Електронний ресурс]. Режим доступу: https://flexberry.github.io/ru/gbt_csharp.html

26. MySQL, переваги та популярність. [Електронний ресурс]. Режим доступу: <https://www.mysql.com/>
27. Що таке Sql, недоліки Sql. [Електронний ресурс]. Режим доступу: <https://www.sql.ru/>
28. Фреймворк Vue js, переваги та недоліки. [Електронний ресурс]. Режим доступу: <https://vuejs.org/>
29. Фреймворк React js, причини популярності. [Електронний ресурс]. Режим доступу: <https://ru.reactjs.org/>
30. Реляційна база даних. [Електронний ресурс]. Режим доступу: <https://sites.google.com/site/bazidanihsubdaccess/bazi—danih—subd/ierarhicna—model—danih/relacijna—model—baz—danih>
31. Переваги реляційної бази даних. [Електронний ресурс]. Режим доступу: <http://um.co.ua/8/8—16/8—164394.html>

ДОДАТОК Ж
ПРОТОКОЛ
ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ
РОБОТИ НА НАЯВНІСТЬ ТЕКСТОВИХ
ЗАПОЗИЧЕНЬ

Назва роботи: Програмний модуль інтерактивного тестування знань

Тип роботи: магістерська кваліфікаційна робота
 (БДР, МКР)

Підрозділ кафедра обчислювальної техніки
 (кафедра, факультет)

Показники звіту подібності
Unicheck

Оригінальність 91.1% Схожість 8.9%

Аналіз звіту подібності (відмітити потрібне):

- Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
- Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку _____ Захарченко С.М.
 (підпис) (прізвище, ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи _____ Герасик О.С.
 (підпис) (прізвище, ініціали)

Керівник роботи _____ Савицька Л.А.
 (підпис) (прізвище, ініціали)