

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА
на тему:
«Удосконалений програмний модуль для визначення рівня професійних знань»

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

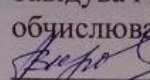
Керівник: д-р техн. наук, доцент, проф.
_____ Азаров О.Д.
«__» _____ 2022 р.

Опонент: к.т.н., доц.,
_____ Яремчук Ю. Є.
«__» _____ 2022 р.

Допущено до захисту
Завідувач кафедри ОТ
д.т.н., проф. Азаров О.Д.
«__» _____ 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.

ЗАВДАННЯ

НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Трояну Івану Юрійовичу

1 Тема роботи «Удосконалений модуль для визначення рівня професійних знань», керівник Азаров Олексій Дмитрович - д.т.н., проф. каф. ОТ затверджені наказом №205-А вищого навчального закладу від 10.09.2022 р.

2 Строк подання студентом роботи 22 листопада 2022 року.

3 Вихідні дані до роботи: Вихідні дані до роботи: кількість записів в базі даних — не менше 20, кількість символів опису заявки не повинна перевищувати 270; час від подачі заявки до збереження її в базі даних не перевищує 45 секунд; кількість тестів у режимі іспиту — 24, кількість тем для тестів — 1.



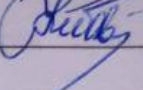
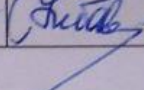
4 Зміст пояснювальної записки (перелік питань, які потрібно розробити): вступ, обґрунтування доцільності розробки програмного модуля для визначення рівня професійних знань, проектування програмного модуля для визначення рівня професійних знань, програмна реалізація програмного модуля для визначення рівня професійних знань, висновки, перелік використаних джерел, додатки.

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): технічне завдання, загальний алгоритм функціонування системи, алгоритм авторизації, алгоритм модуля тестування, лістинг основного коду програми, архітектурний шаблон MVC.

Схема загального алгоритму функціонування системи; загальна структурна схема; вигляд інтерфейсу користувача; приклади роботи програми.

6 Консультанти розділів роботи

Таблиця 1 — Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1,2,3,4	Азаров О.Д., доц. каф. КН		
5	Небава Микола Іванович, к.е.н., професор каф. ЕПВМ		

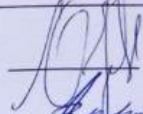
7 Дата видачі завдання 07.09.2022 р.

8 Календарний план наведено в таблиці 2.

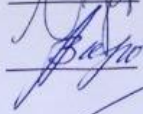
Таблиця 2 — Календарний план

№	Назва етапів виконання магістерської роботи	Строк виконання етапів роботи	Примітка
1	Постановка задачі роботи	01.09.22	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 модуля для визначення рівня професійних знань	02.10—18.10.22	вик.
6	Розробка алгоритмів модуля для визначення рівня професійних знань	19.10—31.10.22	вик.
7	Практична модуля для визначення рівня професійних знань	11.11—16.11.22	вик.
8	Розрахунок економічної частини роботи	17.11—30.11.22	вик.
9	Оформлення пояснювальної записки та ілюстративного матеріалу	01.12—06.12.22	вик.
10	Аналіз виконання роботи, висновки, додатки	07.12—14.12.22	вик.
11	Перевірка якості виконання магістерської роботи та усунення недоліків	15.12.22	вик.

Студент

 Троян І. Ю.

Керівник роботи

 Азаров О. Д.

АНОТАЦІЯ

УДК 004.42

Троян І. Ю. Удосконалений модуль для визначення рівня професійних знань. Магістерська кваліфікаційна робота за спеціальністю 123 — комп'ютерна інженерія, освітня програма-комп'ютерна інженерія. Вінниця: вид-у ВНТУ, 2022. 99с.

На укр.мові. Бібліогр.: 45 назв; рис.: 22; табл. 09.

У дипломній роботі бакалавра було проведено аналіз тематичної галузі програмного модуля з метою визначення рівня професійних знань. Розглянуто специфіку реалізації програми. Розроблено модель програмного модуля і схеми алгоритмів.

Згідно розробленої моделі, схема алгоритму програмно реалізована в додатку. Програмне забезпечення розроблено на мові програмування JavaScript в середовищі WebStorm з використанням технології Express. Розроблене додаток було протестовано.

Ключові слова: тестування, модуль, оцінка знань.

ABSTRACT

Troyan I. Yu. advanced module for determining the level of professional knowledge. Master's qualification work in specialty 123-computer engineering, educational program — computer engineering. Vinnytsia: VNTU publ., 2022. 99c.

In Ukrainian language. Refs.: 45 titles; fig.: 22; table. 09.

In The Bachelor's thesis, an analysis of the thematic area of the program module was carried out in order to determine the level of professional knowledge. The specifics of the program implementation are considered. A model of the software module and algorithm schemes are developed.

According to the developed model, the algorithm scheme is programmatically implemented in the application. The software is developed in the JavaScript programming language in the WebStorm environment using Express technology. The developed app has been tested.

Keywords: testing, module, knowledge assessment.

ВСТУП

Актуальність теми полягає в тому, що з розвитком мережевих технологій, зокрема мережі Internet, постала потреба швидко, конфіденційно та без великих зусиль обмінюватись збереженою інформацією в базах даних. З'явилась потреба в створенні програмних продуктів, які б працювали з базами даних і водночас були віддаленими. Такі програмні продукти стали досить актуальними, оскільки вони не прив'язані ні до робочого місця користувача, ні до конкретної операційної системи, а всі дані зберігаються на віддаленому сервері. Кожен користувач у будь-який момент може отримати доступ до своїх даних.

В мережі Internet є популярні різні тести: тести IQ, які дозволяють дізнатись коефіцієнт інтелекту, психологічні тести, які покликані допомогти людині виміряти свої індивідуально-психологічні особливості. Разом з цим існує мало тестових систем, які допомагають визначити професійну придатність людей, наприклад програмістів. Такі тести проводять деякі компанії при оцінюванні персоналу, при прийомі працівників на роботу з метою визначення професійної придатності.

Саме тому варто розробити програмний модуль для визначення рівня професійних знань, який сприятиме якісному підбору кваліфікованого персоналу на роботу.

Метою роботи є вдосконалення та розширення функціональних можливостей модуля.

Для досягнення мети необхідно вирішити такі завдання:

— проаналізувати технічний рівень програми та визначити рівень експертності;

— обґрунтувати доцільність створення програмного модуля для визначення рівня експертних знань- розробити архітектуру і структуру програмного модуля для визначення рівня експертних знань;

— провести аналіз і обґрунтувати вибір інструментів розробки;

— виконати програмну реалізацію програмного модуля для визначення рівня експертних знань;

— провести тестування програми та проаналізувати отримані результати.

Наукова новизна полягає в удосконаленні алгоритму, завдяки чому збільшується точність результату при проходженні тестування; зменшення часу на оцінювання результатів та швидкодією.

Практична цінність роботи полягає в удосконаленні алгоритму завдяки збільшенню типів питань та удосконаленні інтерфейсу веб-додатку, дозволяє користувачу ознайомитися з інструкцією проходження тесту; збільшеною упередженістю під час оцінювання знань;

Апробація результатів досліджень була здійснена на І науково-технічній конференції професорсько-викладацького складу, співробітників та студентів ВНТУ з участю працівників науково-дослідних організацій.

Матеріали роботи доповідались та опубліковувались [1]:

Троян І. Ю., Азаров О. Д. Удосконалений модуль для визначення рівня професійних знань // Тези доповіді. XLIX регіональної науково-технічної конференції професорсько-викладацького складу, співробітників та студентів ВНТУ з участю працівників науково-дослідних організацій. Вінниця 2022 р.

Режим доступу: <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-21/paper/view/12686>

1 АНАЛІЗ АБО ДОСЛІДЖЕННЯ МОЖЛИВОСТЕЙ ВДОСКОНАЛЕННЯ МОДУЛЯ ДОЛЯ ВИЗНАЧЕННЯ

1.1 Аналіз технічних рішень для удосконалення модуля для визначення рівня професійних знань

Одним із найефективніших способів визначення рівня компетентності є тести. Більшість досліджень у цій галузі відповідають потребам часу, але стосуються окремих аспектів і проблем тестування.

Останнім часом в Інтернеті стали популярні різні тести: IQ—тести, що дозволяють людям дізнатися свій IQ, психологічні тести, які покликані допомогти людині виміряти його індивідуальні психічні особливості. Але крім цього, тести такого роду мають інше застосування: вони є засобом реклами для компаній, а також служать комерційними проектами, які повинні приносити дохід. Однак існує небагато систем тестування, які допомагають визначити професійну придатність таких людей, як програмісти. Такі тести деякі компанії проводять при оцінці персоналу, при прийомі на роботу співробітників для визначення професійної придатності. Наприклад, відома компанія «Майкрософт» під час прийому на роботу проводить тестування на розгадування проблем і головоломок, щоб виявити з числа талановитих кандидатів більш креативних.

Тому метою написання дипломного проекту є створення тестової системи визначення професійної придатності програмістів. Ця тема актуальна на сьогоднішній день і має знайти своє практичне застосування.

Тестова система формалізованих завдань, часових тестів, призначена для визначення рівня освіченості особистості, кількісних і якісних індивідуально-психологічних відмінностей. Тест як засіб вимірювання визначається як система завдань, що дозволяє якісно оцінити структуру й ефективно виміряти рівень знань [2].

Основні характеристики тесту такі:

— мати заплановану систему тестових завдань, яку неможливо змінити або модифікувати;

— попередній контроль тестового набору і тесту в цілому методами математичної статистики, що приводить їх до певних кількісних і якісних характеристик;

— стандартизована процедура тестування;

— стандартизована система оцінювання та порівняння результатів зі стандартними шкалами визначення рівня предметів [3].

Чіткої класифікації тестів немає. Зазвичай тести класифікують за призначенням або спрямованістю діагностики: тести здібностей, інтелекту, властивостей особистості, психічних процесів і т. д. Існує розподіл тестів за формою (ілюстративні, проектні, форми).

Першими перевітками навчальних досягнень є контрольні роботи з розв'язування арифметичних, орфографічних, почеркознавчих завдань та завдань на міркування. З того часу вони зазнали значних змін, але незмінним залишається їхнє призначення — визначення рівня освіти [4].

Останнім часом для перевірки знань все частіше використовують тести. Тести створюються в кілька етапів:

1) визначення мети дослідження, виду дослідження та підходу до його підготовки.

2) аналіз змісту конкретної дисципліни та відбір матеріалу для колоквіуму.

3) розробка структури та специфікації тесту.

4) розробка тестових завдань.

5) експертний аналіз змісту та форми контрольних робіт з подальшою адаптацією змісту та форми завдань до результатів іспиту.

6) проведення пробного ЗНО та обробка його результатів.

7) оцінка якості виконання завдань і тестів у цілому статистичними методами та подальше коригування тесту.

8) розробка остаточного варіанту тесту, завдань та його стандартизація.

9) розробка інструкції для користувачів.

Після проходження всіх етапів тест готовий і може бути використаний для безпосереднього тестування [5].

1.2 Методика аналізу та оцінка результатів тестування

Виникає необхідність привести результати дослідження до певної системи, проаналізувати та обробити їх. Вибір технології обробки результатів тесту залежить від цілей тесту. Дані відгуків дозволяють не тільки оцінити результати навчання кожної людини шляхом проходження тесту, а й зробити обґрунтовані висновки про специфіку навчання, технології навчання, зробити діагностику чи прогноз процесу навчання [6].

Існує дві групи методів аналізу завдань: експертні та математичні. Математичні методи застосовуються після тестування на основі отриманого емпіричного матеріалу, тому ми поговоримо про них нижче. Експертні методи дозволяють не тільки оцінити, якою мірою точно виміряти знання, уміння та навички, для яких він розроблений тестувати, а й оцінювати завдання за формою. В основі експертних методик лежить співвідношення змісту і форми завдань із вимогами, зазначеними в специфікації. Завдання, для вирішення яких не зголосився жоден експерт коментарі в тесті. Такі завдання вважаються об'єктивними і правильно сформульованими. Інші завдання переформульовано або взагалі вилучено з тесту. Після розробки та проведення експертного аналізу завдань виходить перший варіант тесту. Для підвищення його якості використовуються математичні методи. Експериментальні результати випробування необхідні для застосування цих методів. Такі випробування називаються попередніми.

Якщо оцінка, виражена в балах, що оцінюють ефективність педагогічного процесу, піддається математичній обробці, то не можна забувати, що вони є узагальненим виразом знань, умінь і навичок, які виключають можливість виявлення зв'язків між учителем і досвідчені у всій їх різноманітності.

Використовуючи математичний метод обробки даних, перед тестувальниками постає питання про точність і достовірність педагогічних висновків на основі математичних формул. Це відіграє важливу роль у теорії

відносності щодо процесу навчання. Тому неприпустимо бути бездарним, формальне застосування математичних методів

Використання математичної статистики, як свідчать дослідження, - один із ефективних шляхів пізнання об'єктивних закономірностей виховання і розвитку. Він є законним і ефективним лише тоді, коли підтримується для вмілого та всебічного якісного аналізу, коли математичні формули є конкретним вираженням якісних ознак формування знань, умінь. Усе це передбачає встановлення певних співвідношень між показниками та факторними характеристиками, що характеризують різні сторони підготовки досліджуваних. Знання функціонального зв'язку між ними дозволяє прогнозувати рівень підготовки для кожного пункту.

Слід зазначити, що людина як об'єкт дослідження надто складний її проявів, так що її поведінку можна віднести до певних формули, тому основна увага приділяється вивченню статистичних формул знання. Але в кожному випадку основою для висновків є оцінка досягнень окремих предметів. Тож спершу зосередимося на цьому.

Успіх у роботі з тестом означає виконання його завдань. Ось чому тест — це система завдань, які необхідно порівняти з результатами різні завдання між собою.

При виконанні завдання в закритій формі випробуваному пред'являються дві супутні завдання: визначити правильні відповіді (їх може бути кілька) і знаходити помилки в неправильних відповідях. Додаткові труднощі для випробуваного в розпізнаванні правильних відповідей створює той факт, що вони можуть бути певною мірою пов'язані з одним завданням і можуть перебувати в різних відносинах. Варіанти відповідей вибираються як вірогідні, де помилки маскуються. Іноді виявити помилку у відповіді важче, ніж розпізнати правильну (можливо, загальновідому) відповідь. Оцінюючи складність завдання, не можна не звернути увагу на складність розпізнавання неправильних відповідей.

При оцінюванні результатів виконання випробуваними завдань розробник тесту стикається з рядом проблем. Проблеми немає, лише якщо відповідь точно

збігається з правильною еталонною відповіддю. Що робити, якщо визначено не всі правильні відповіді? Рішення може бути іншим: якщо ці правильні відповіді рівні, завдання можна вважати виконаним, наприклад, коли знайдено хоча б одну правильну відповідь; якщо відповіді доповнюють одна одну і тільки разом дають повну, завершену відповідь, укладач повинен визначити її відношення до кожного конкретного випадку.

Згідно з даними досліджень, математична статистика є одним із найефективніших способів пізнання об'єктивних закономірностей навчання. Його застосування є ефективно виправданим, коли воно базується на вмілому якісному аналізі, а математичні формули є конкретним вираженням якісних особливостей формування знань і умінь.

Пройти тест означає виконати завдання, які в ньому містяться. Оскільки тест є компонуванням завдань, необхідно порівняти їх результати.

При оцінці результатів тестування виникає ряд проблем, яких немає лише тоді, коли відповідь абсолютно збігається з еталонною відповіддю. Однак не всі правильні відповіді можна визначити, не всі неправильні відповіді можна вибрати або всі відповіді однаково дійсні [7].

1.3 Огляд програм-аналогів

Кожен тест складається з безлічі різних завдань, де рівень складності поступово зростає. Серед них тестові завдання на логічне та просторове мислення, а також завдання інших типів. За результатами тесту підраховується IQ. Помічено, що чим більше варіантів тесту проходить випробовуваний, тим кращі результати він демонструє. Найвідомішим тестом є Тест Айзенка. Більш точними є тести Векслера, тест Равена, тест Амтгауера. При використанні теста Равена, критичним вважається рівень в 70 балів. Нижче цього рівня клінічні психологи класифікують розумову відсталість. На цей момент не існує єдиного стандарту на тести IQ.

Тести поділяються за віковими групами та показують розвиток людини, відповідно до її віку. Тобто дитина в 10 років та випускник вишу можуть мати

однаковий IQ, оскільки розвиток кожного з них відповідає своїй віковій групі. Тест Айзенка розроблений для вікової групи від 18 років і передбачає максимальний рівень IQ у 160 балів.

Кожен тест складається із серії завдань, для виконання яких людина повинна зосередитися, подумати та визначити закономірності. Щоб ваш IQ вважався більшим за 100, ви повинні розв'язати більше завдань, ніж кількість розв'язків, за які присуджуються 100 балів. Час виконання тесту повинен бути обмежений!

Кількість тестових завдань, які людина виконує за цей обмежений час, насправді є показником інтелекту.

Майже всі тести IQ в Інтернеті застаріли. Тести IQ, які ви знайдете в журналах, швидше за все, непрофесійні. Якщо ви хочете дізнатися свій точний результат, вам доведеться звернутися в професійний психодіагностичний центр.

Розглянемо деякі з них і проведемо аналіз, щоб вибрати найбільш підходящий для використання в якості прототипу.

IQ Test 2.2 один з найпопулярніших тестів, для визначення знання, де максимальний бал 174, а середній статистичний в районі 90 балів.

Всього в додатку близько 40 завдань при обмеженому ліміті часу в 30 хвилин [8].

До переваг можна віднести:

- не потребує інсталяції.
- є інструкції, які допоможуть орієнтуватися.

До недоліків можна віднести:

- немає вибору мови інтерфейсу.
- застарілий дизайн.

Приклад програмного додатку IQ Test 2.2 показано на рисунку 1.1.

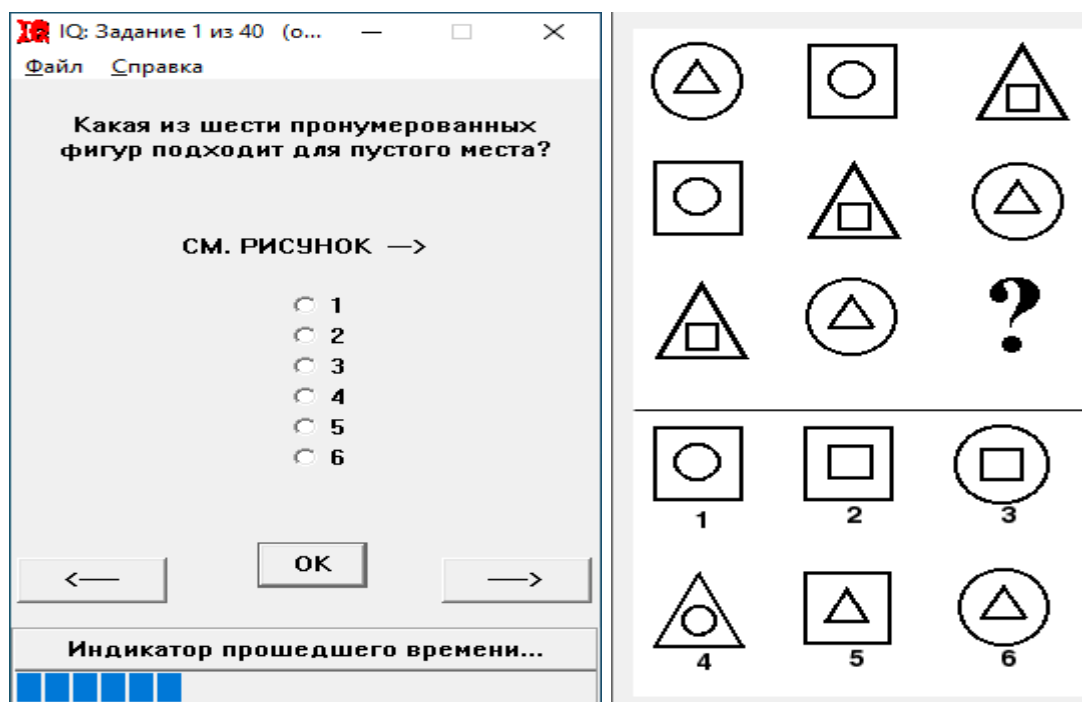


Рисунок. 1.1 — Приклад роботи програми IQ Test 2.2

Тест Айзенка — це тест, що дозволяє з максимальною точністю визначити інтелектуальні здібності людини. Застосування цього тесту можливо як при прийомі на роботу або в будь-який навчальний заклад на конкурсній основі, так і для самостійного вимірювання власного коефіцієнта інтелекту (рис. 1.2).

До переваг можна віднести:

- не потрібно завантажувати.
- є інструкція користувача.
- показано таймер до закінчення часу.

Недоліки:

- залежність правильних відповідей від написання малих або великих літер.
- не показано, які відповіді правильні, які неправильні [9].

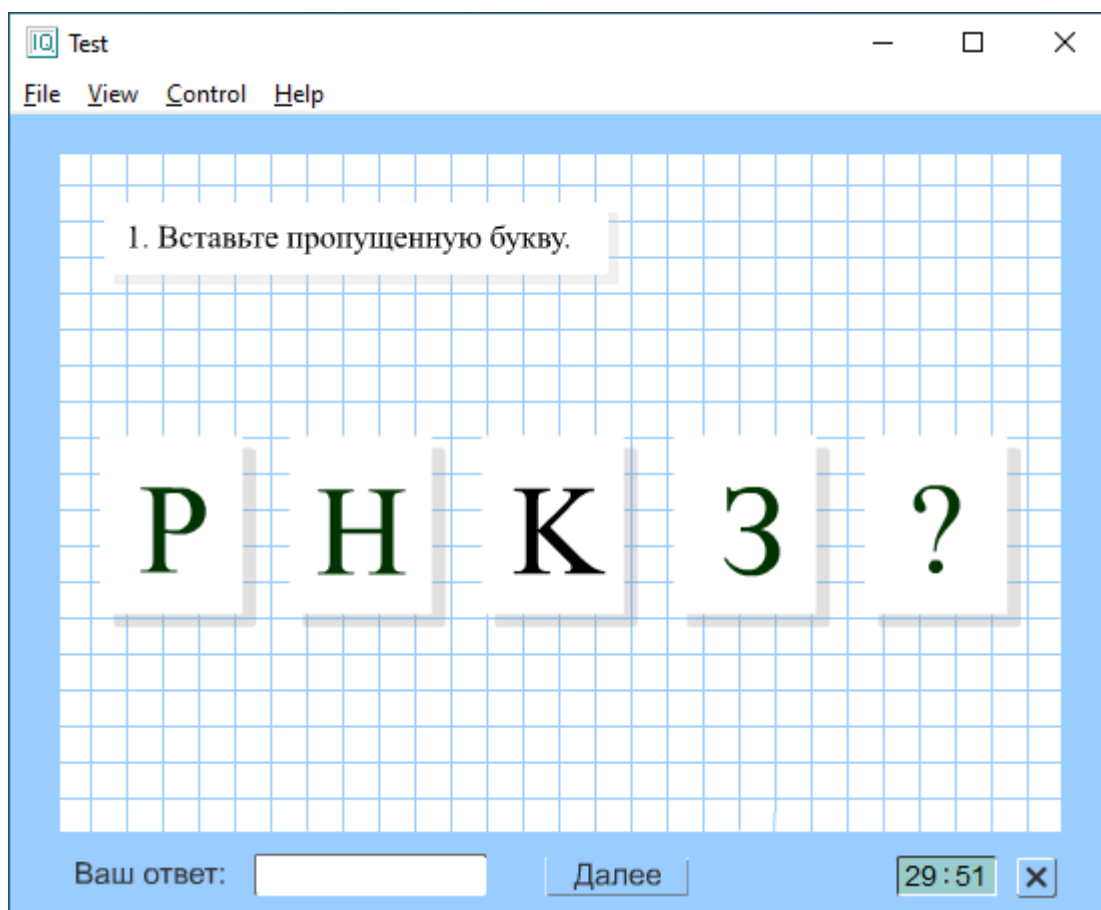


Рисунок. 1.2 — Приклад роботи Тесту Айзенка.

Прогресивні матриці Равена — тест інтелекту. Призначений для вимірювання рівня інтелектуального розвитку, вивчення логічності мислення. Подвоюються малюнки з фігурами, пов'язаними між собою певною залежністю. Однією фігури не дістає, а внизу вона дається серед 6–8 інших фігур. Завдання: встановити закономірність, що зв'язує між собою фігури на малюнку, і в полі відповіді, вказавши шукану фігуру серед пропонованих варіантів. Тест складається з 60 таблиць (рис. 1.3) [10].

До переваг можна віднести:

- не потребує інсталяції.
- є інструкції, які допоможуть орієнтуватися.

До недоліків можна віднести:

- немає вибору мови інтерфейсу.

— потрібна реєстрація.

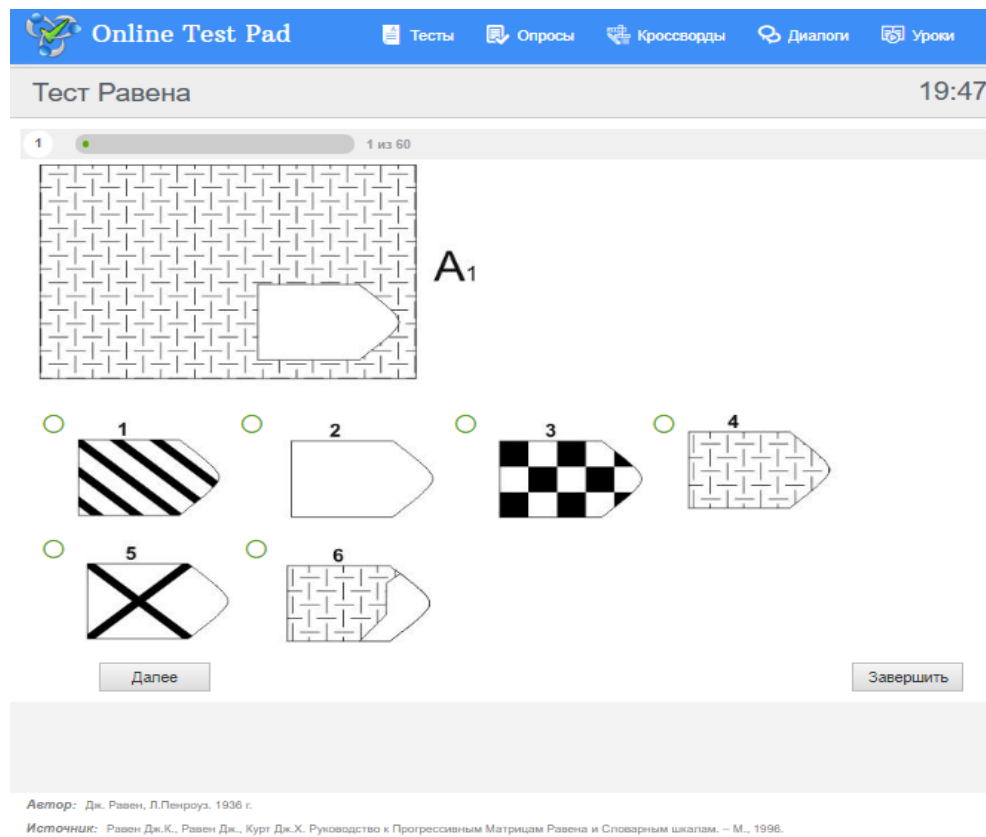


Рисунок 1.3 — Приклад роботи Тесту Равена

Iq-test 2021 це міжнародний тест, розроблений в 2021 році, спеціально для різних країн, де можна подивитися результати останніх проходжень (рис. 1.4) [11].

До переваг можна віднести:

- вибір якої мови інтерфейсу.
- можливість подивитися останні результати
- серед недоліків програми є:
- немає вибору мови інтерфейсу.
- потрібна реєстрація.
- багато непотрібної інформації.

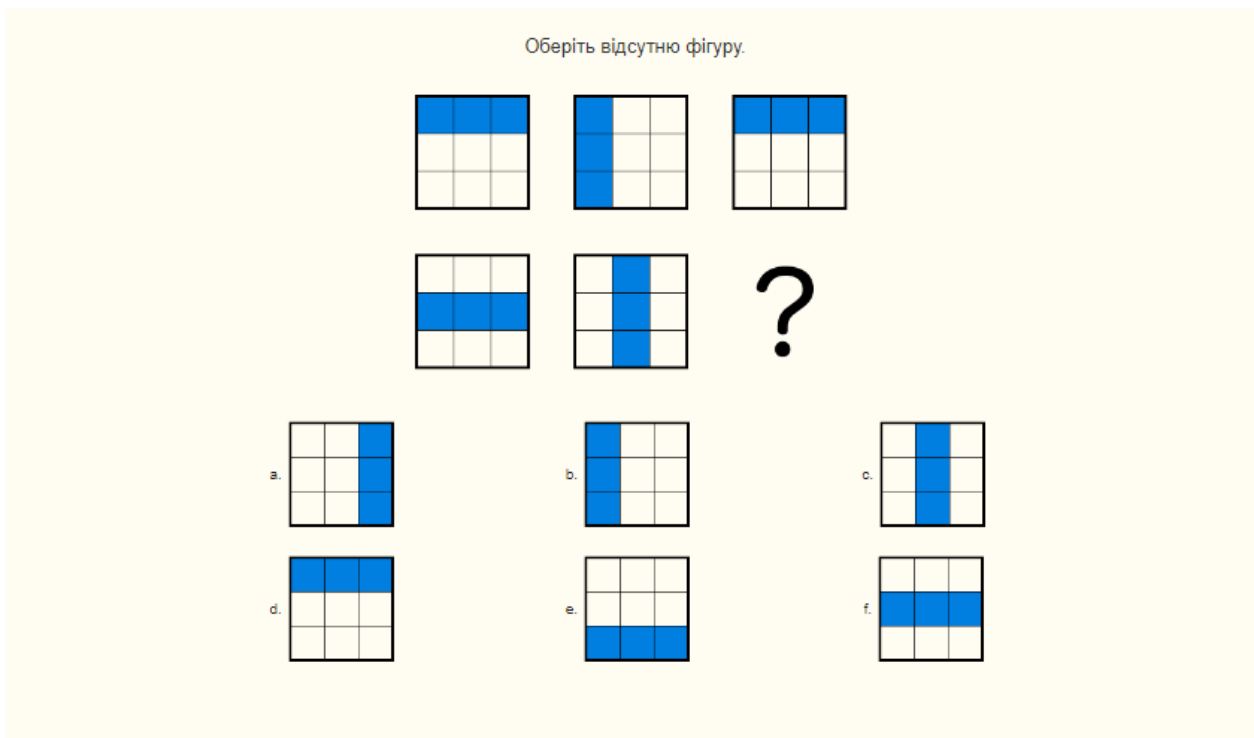


Рисунок 1.4 — Приклад роботи Iq-test 2021

1.4 Формулювання вимог та постановка задачі

Сьогодні тест є досить актуальним, адже він існує доступний кожному, хто має доступ до Інтернету. Користувач може в будь-який час перевірити рівень своїх знань.

У загальному описі функціональних можливостей програмного модуля представлена реалізація входу в систему та можливість процесу реєстрації, а також можливість проходження тесту необмежену кількість разів.

Для визначення рівня професійних знань у бакалаврській роботі необхідно впровадити програмний модуль. Необхідно розробити додаток, в якому користувач зможе поглибити свої знання.

Вхідними даними для тестування та збереження результатів є: «форма реєстрації користувача» та «форма авторизації». Нормативно-довідковою інформацією для створення тестової системи визначення професійної придатності розробників є наступні посібники:

- питання, створені адміністратором тестової системи;
- відповіді, які формулює адміністратор тестової системи;

— рівні придатності, створені системним адміністратором.

Обов'язковим елементом комплексної автоматизації є розробка і використання єдиної номенклатури кодів, простіше кажучи, класифікації об'єктів. Класифікація — науковий метод вивчення природи предметів і явищ за допомогою певних правил їх систематизації.

Для визначення професійної придатності програмістів розраховуються такі показники:

- кількість балів, отриманих за це питання;
- кількість питань;
- рівень складності запитання: ;
- максимально можлива загальна кількість балів за питання.
- коефіцієнт професійної корисності:

Програмне забезпечення для програмування дозволить швидко та без особливих зусиль зберігати інформацію про питання та відповіді на тести, щоб визначити професійну придатність програмістів.

Реєстрація користувача полягатиме в заповненні та збереженні таких даних: пароль, логін для авторизації;

Після проходження тесту користувач отримає результат, який містить наступну інформацію:

- результат - результат випробування;
- час перевірки.

Програмне забезпечення виконуватиме певний набір завдань, які можна розділити на дві групи:

- Завдання, які будуть доступні звичайним користувачам;
- завдання, які будуть доступні адміністраторам.

До завдань першої групи входять:

- проходження тесту;
- реєстрація;
- збереження результату;
- відправка результату другу.

До другої групи входять такі завдання:

- керування зареєстрованими користувачами;
- створення/редагування/видалення запитання;
- створення/редагування/видалення відповідей;
- налаштування тестової системи: кількість запитань, які повинні бути показані за сеанс;
- налаштування рейтингу користувачів;
- керуючі адміністратори.

1.5 Аналіз інформаційного забезпечення предметної області

Нині існують різні реалізації тестів здібностей: від локальних версій, написаних у вигляді програми, через операційну систему, до т.зв. веб-версія. Вони суттєво відрізняються, оскільки для їх створення використовуються різні технології. Основні з них: Flash, програми Windows, програми Internet.

Flash — це продукт Macromedia, який дозволяє створювати інтерактивні мультимедійні програми. Сфера використання Flash буває різною, це можуть бути ігри, сайти, презентації, банери і просто мультфільми. При створенні продукту можна використовувати мультимедійні, звукові та графічні файли. Flash заснований на векторному морфінгу, технології комп'ютерної графіки, яка забезпечує плавний перехід від одного ключового кадру до іншого. Це дозволяє створювати досить складні сцени анімації, встановлюючи лише кілька ключових кадрів для кожного персонажа.

Ще однією перевагою технології Flash є повна програмованість. Flash використовує мову програмування ActionScript, яка синтаксично схожа на JavaScript. Остання версія мови (ActionScript 3.0) є повноцінною об'єктно-орієнтованою мовою.

Серед цих переваг також є ряд особливостей, які можна віднести до недоліків технології Flash. Наприклад, для перегляду проекту, створеного у Flash, користувач повинен додатково встановити Flash Player. Крім того, проект, написаний у Flash, є статичним, що означає, що вміст програмного забезпечення

не буде змінено користувачем без втручання в код [2]. Програми Windows — це програми, написані на таких мовах програмування як: C++, Visual Basic, Delphi, C# та ін.. За допомогою такої технології можна не тільки розробити проект, але й створити для нього панель управління, в якій адміністратор міг змінити зміст проекту. Однак, незважаючи на переваги та зручності, які надають ці мови програмування, користувачеві все одно доведеться встановлювати додаткове програмне забезпечення. Наприклад, щоб працювати з додатком, написаним на C#, необхідно встановити .NET Framework.

Веб-додатки — це програми, створені для роботи в серверному середовищі і написані на таких мовах програмування як: PHP, ASP, Perl, Ruby, Java та інших. Ці мови програмування дозволяють створювати повноцінні програми, створювати системи управління контентом для цих програм. Веб-додатки працюють тільки в серверному середовищі, тобто для роботи з ними необхідно встановити сервер, оскільки скрипти виконуються на сервері. Але в той же час такі додатки можна розміщувати в Інтернеті, де вже налаштований сервер, тому для роботи користувач повинен мати веб-браузер і доступ до Інтернету. Перевага веб-додатків також полягає в тому, що користувач може отримати доступ до проекту через Інтернет у будь-який час.

Серед доступних в Інтернеті систем тестування небагато таких, які дозволяють пройти тест і перевірити свою професійну придатність. Знайти тестову систему з підходящою тематикою дуже складно, до того ж більшість з них платні і для отримання результату користувачеві необхідно відправити платне СМС.

2 УДОСКОНАЛЕННЯ ПРОГРАМНОГО МОДУЛЯ ДЛЯ ВИЗНАЧЕННЯ РІВНЯ ПРОФЕСІЙНИХ ЗНАНЬ

2.1 Розробка архітектури та структури програмного модуля для визначення рівня професійних знань

Таке програмне забезпечення призначене для перевірки та визначення професійної придатності програмістів. Ви можете додавати та редагувати питання та відповіді. У цьому процесі відбувається безперервний обмін даними між модулями. Клієнтська частина надсилає заповнену форму разом із даними на веб-сервер. На веб-сервері дані обробляються за допомогою скриптів php. На основі отриманих даних PHP—скрипти створюють SQL-запити та відправляють їх на сервер бази даних. Сервер бази даних отримує текст SQL-запитів, обробляє їх і надсилає результати цих запитів назад на веб-сервер. На основі отриманого результату веб-сервер створює HTML-сторінку, яка надсилається користувачеві у веб-браузері.

Додаток для оцінювання професійних знань містить такі функції:

- створення тестів;
- редагування тесту;
- генерація тестів;
- вихід тестових завдань;
- розрахунок результату;
- вихід результату.

Модуль генерації працює з базою та отримує завдання з тесту. Потім він генерує тест і взаємодіє з навчальним модулем.

Модуль проходження відповідає за функції організації тестових завдань, підрахунку правильних відповідей і відображення результату проходження обраного тесту.

Модулі створення та редагування відповідають за функції створення та редагування тесту, який працює з базою даних, завантаження, збереження та редагування інформації в базі даних.

Взаємодія між модулями показано на рисунку 2.1

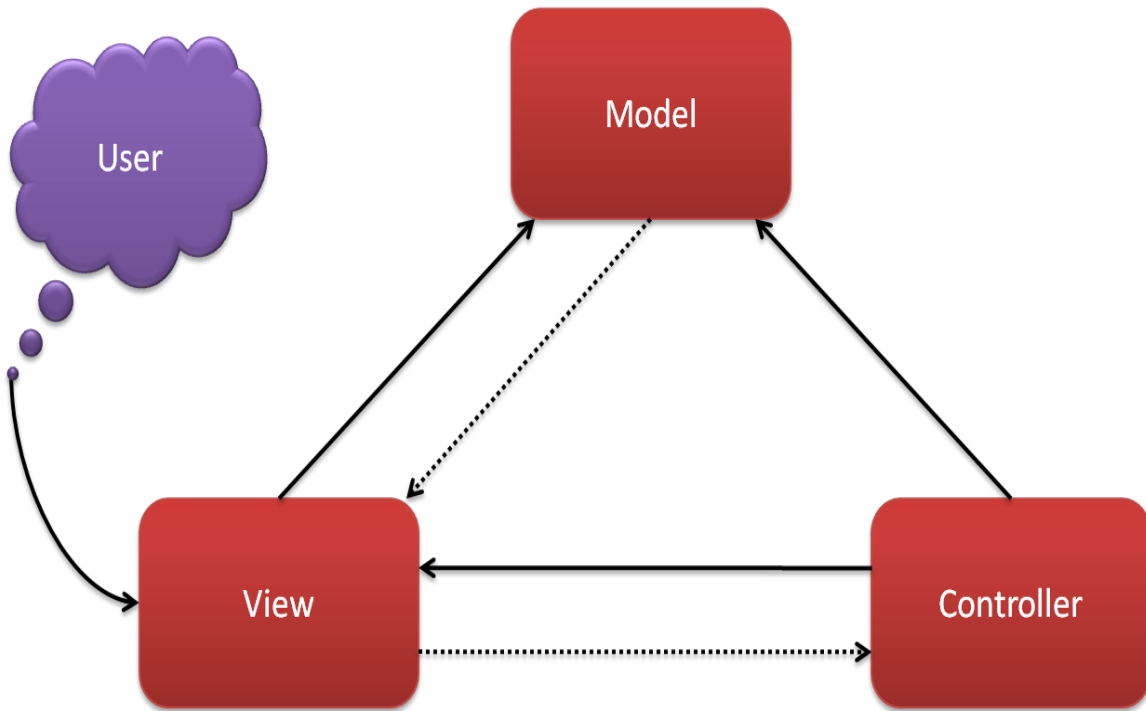


Рисунок 2.1 — Використання архітектурного шаблону MVC

2.2 Архітектурний шаблон MVC

Model-View-Controller — це архітектурний шаблон, який використовується при проектуванні та розробці програмного забезпечення.

Цей шаблон ділить систему на три частини: модель даних, представлення даних і елемент керування. Він використовується для відокремлення даних (моделі) від інтерфейсу користувача (перегляду), щоб зміни в інтерфейсі користувача мали мінімальний вплив на роботу з даними, а зміни в моделі даних можна було вносити без зміни інтерфейсу користувача.

Завдання шаблону — гнучкий дизайн програмного забезпечення, який має сприяти подальшим змінам або розширенням програм, а також забезпечувати можливість повторного використання окремих елементів програми. Крім того, застосування цього шаблону до великих систем призводить до певного впорядкування їх структури і робить їх більш зрозумілими за рахунок зменшення складності.

Важливою особливістю Java 2 Enterprise Edition (J2EE) є те, що шаблон моделі Beans визначено в цій технології. Представленням є сторінка сервера Java

або, альтернативно, код сервлету, який генерує представлення. Сервлет можна вважати контролером.

Java Swing, на відміну від інших платформ, не тільки надає інтерфейс програмування на основі шаблону MVC, але й реалізований на його основі. Представлення — це клас, який успадковує клас JComponent. Завдяки організації моделі подій Java на інтерфейсах, контролер є набором анонімних класів для обробки відповідних подій. Як і інші платформи, Swing надає розробнику модель розробки [3]. Концепція MVC:

Концепція MVC в останні роки багато згадується у світі веб-розробки. Кожен, хто має хоч якесь відношення до розробки веб-додатків, так чи інакше стикався з цією аббревіатурою. Сьогодні ми розберемося, що таке концепція MVC і чому вона стала популярною.

Контролер обробляє запити користувачів (отримані як запити HTTP GET або POST, коли користувач натискає елементи інтерфейсу для виконання різних дій). Його основною функцією є виклик і координація роботи необхідних ресурсів і об'єктів, необхідних для виконання зазначених користувачем дій. Зазвичай контролер викликає відповідну модель завдання та вибирає відповідне представлення.

Модель — це дані та правила роботи з даними, які складають концепцію управління додатком. У будь-якій програмі вся структура моделюється як дані, оброблені певним чином. Хто є користувачем програми — повідомлення чи книга? Лише дані, які мають бути оброблені згідно з правилами (дата не може бути в майбутньому, електронна пошта має бути в певному форматі, ім'я не має містити більше X символів тощо).

Модель надає контролеру перегляд даних, які запитує користувач (повідомлення, сторінка книги, фотоальбом тощо). Модель даних буде однаковою незалежно від того, як ми хочемо представити її користувачеві. Тому ми вибираємо будь-яке доступне представлення для відображення даних. Модель містить найважливішу частину логіки нашої програми, логіку, яка вирішує

завдання, яке ми маємо справу. Контролер в основному містить організаційну логіку для самої програми (дуже схоже на ведення господарства).

Представлення надає різні способи представлення даних, отриманих з моделі. Це може бути шаблон, заповнений даними. Можливі кілька різних типів, і контролер вибирає той, який найкраще відповідає поточній ситуації.

Веб-додаток зазвичай складається з набору контролерів, моделей і представлень. Контролер може бути налаштований як головний, який отримує всі запити та викликає інші контролери для виконання дій залежно від ситуації.

Переваги: Найбільш очевидною перевагою використання концепції MVC є чітке відокремлення логіки представлення (інтерфейс користувача) від логіки програми.

Підтримка різних типів користувачів, які використовують різні типи пристроїв, сьогодні є загальною проблемою. Наданий інтерфейс має відрізнятися, якщо запит надходить із персонального комп'ютера чи мобільного телефону. Модель повертає однакові дані, єдина відмінність полягає в тому, що контролер вибирає різні типи виведення.

На додаток до ізоляції представлень від логіки програми, концепція MVC значно зменшує складність великих програм. Код набагато більш структурований, що полегшує підтримку, тестування та повторне використання рішень.

2.3 Розробка схеми алгоритму роботи програмного модуля

Необхідність використання адаптивних алгоритмів перевірки знань викликана поширенням комп'ютерних систем навчання. Для досягнення реального ефекту при перевірці знань ці системи повинні автоматично адаптуватися до процесу тестування за низкою параметрів.

Оптимальним є модульний принцип побудови структури змістовної частини навчальних матеріалів, за якого вони поділяються на систему взаємозалежних модулів – частин курсу, що охоплюють окремі закінчені теми. Кожен модуль містить усі необхідні теоретичні відомості, а також приклади розв'язування всіх необхідних типів практичних завдань. Важливим елементом

інформаційних матеріалів є питання самоперевірки. З одного боку, їх формування дозволяє викладачеві оцінити правильність поданої в модулі інформації. З іншого боку, їх зміст має допомогти студенту підготуватися до контрольних матеріалів.

Перевірка знань є окремим багатогранним напрямом педагогіки, призначеним для вирішення ряду завдань перевірки знань. Існує багато способів вибору типу, структури та змісту тестів. Оптимальні рішення багато в чому залежать від мети іспиту, виду матеріалу, що викладається, рівня підготовки студентів, вимог до рівня дисципліни та інших факторів.

Для збереження модульної структури викладу знань у системі навчання як мінімально перевіреної одиниці навчальна дисципліна також повинна мати модуль. В електронних освітніх системах тестування за призначенням поділяється на два основних види:

1) дидактичний, який здійснюється учнями самостійно з метою самоперевірки своїх знань,

2) перевірка, яка здійснюється авторизаційною перевіркою під наглядом викладача для об'єктивної оцінки фактичного рівня знань.

На практиці ці можливості забезпечуються використанням двох різних алгоритмів тестування. Навчальні тести завжди складаються лише за модулями. Результати аналізу відповідей на кожне теоретичне питання або рішення кожного завдання одразу оголошуються студенту. Також у результаті аналізу попередніх відповідей формується контрольна траєкторія — обирається кількість і зміст наступних питань і практичних завдань. Наприкінці навчального заліку виставляється загальний бал і, за необхідності, загальні рекомендації щодо повторного вивчення матеріалу.

Контрольні роботи можуть проводитися по блоках, розділах і по всьому курсу — залежно від побажань виконавця перевірки знань. Для скорочення часу проведення опитування результати поточного контролю не повідомляються студенту. Загальний бал наводиться в кінці опитування.

Відповідно до державних освітніх стандартів частка самостійної роботи в загальному навчальному навантаженні студентів ВНЗ постійно зростає. У цих

умовах використання автоматизованих систем навчання є одним із найефективніших механізмів активізації та реальної реалізації цієї частини самостійної роботи студентів, яка спрямована на підготовку до конкретних видів занять у рамках дисципліни, що вивчається. Знання студентів на курсах електронного навчання, його можна розділити на наступний спосіб:

- 1) для тестування — для навчальних і контрольних цілей,
- 2) за охопленням матеріалу, що вивчається, для екзаменаційних модулів, розділів і всієї дисципліни,
- 3) залежно від складу тестових матеріалів — одноетапне, двоетапне та триетапне тестування.

Адаптивна екзаменаційна система, окрім адаптації до пунктів 1) - 3), також має адаптуватися до рівня знань студента, який складає тести, щоб розпізнавати його більш повно. Розглянемо запропонований підхід до його побудови.

Кожен модульний тестовий блок містить базовий набір із 20 — 30 теоретичних запитань. Якщо тестується оглядовий матеріал, який містить поглиблені теоретичні положення або методи розв'язування задач, цього матеріалу достатньо для перевірки знань. Ми назвемо це однорівневим тестуванням.

Якщо підрозділ містить теоретичні матеріали та/або методiku розв'язування задач, то основну контрольну частину, залежно від змісту навчального матеріалу, необхідно доповнити такими матеріалами: 1) теоретичні питання підвищеної складності, 2) завдання, 3) завдання підвищеної складності. Загалом кількість k запропонованих типів тестового матеріалу на підгрупу на практиці не повинна перевищувати 3. Залежно від кількості k запропонованих рівнів тестування (1, 2 або 3), відповідна схема тестування називається 1-, 2- і 3-ярусний. Для ефективнішої перевірки знань тестування на різних рівнях слід проводити послідовно в порядку зростання складності матеріалу, що перевіряється на кожному рівні.

Щоб адаптувати процес тестування до рівня підготовки студента, пропонується виділити 2 фази тестування на кожному рівні:

1) основний (обов'язковий), на якому виставляються попередні тестові завдання і

2) уточнюючий (на вибір), де подаються додаткові тестові завдання, що дозволяють уточнити знання студента на поточному рівні.

Як загальне правило адаптації процесу тестування на поточному рівні до рівня підготовки студента пропонується прийняти таке правило: якщо кількість правильних відповідей в основному етапі анкети не перевищує половини (недостатній рівень знання), то уточнюються знання на поточному рівні, якщо кількість правильних відповідей перевищує половину (високого) рівня знань), то основне опитування переноситься на наступний рівень.

Для даного програмного веб-додатку використовується доволі простий алгоритм роботи.

Розглянемо алгоритм роботи програмного модуля для визначення рівня професійних знань, який зображений на рисунку 2.2.

Алгоритм складається з таких кроків:

Крок 1. Авторизація користувача.

Крок 2. Відображення вікна авторизації нового користувача.

Крок 3. Перевірка коректності даних для логіна.

Крок 4. Виведення головної сторінки.

Крок 5. Пошук у БД запитань по зростанню рівня складності.

Крок 6. Проведення тестування для даного користувача.

Крок 7. Вихід з програмного додатку.

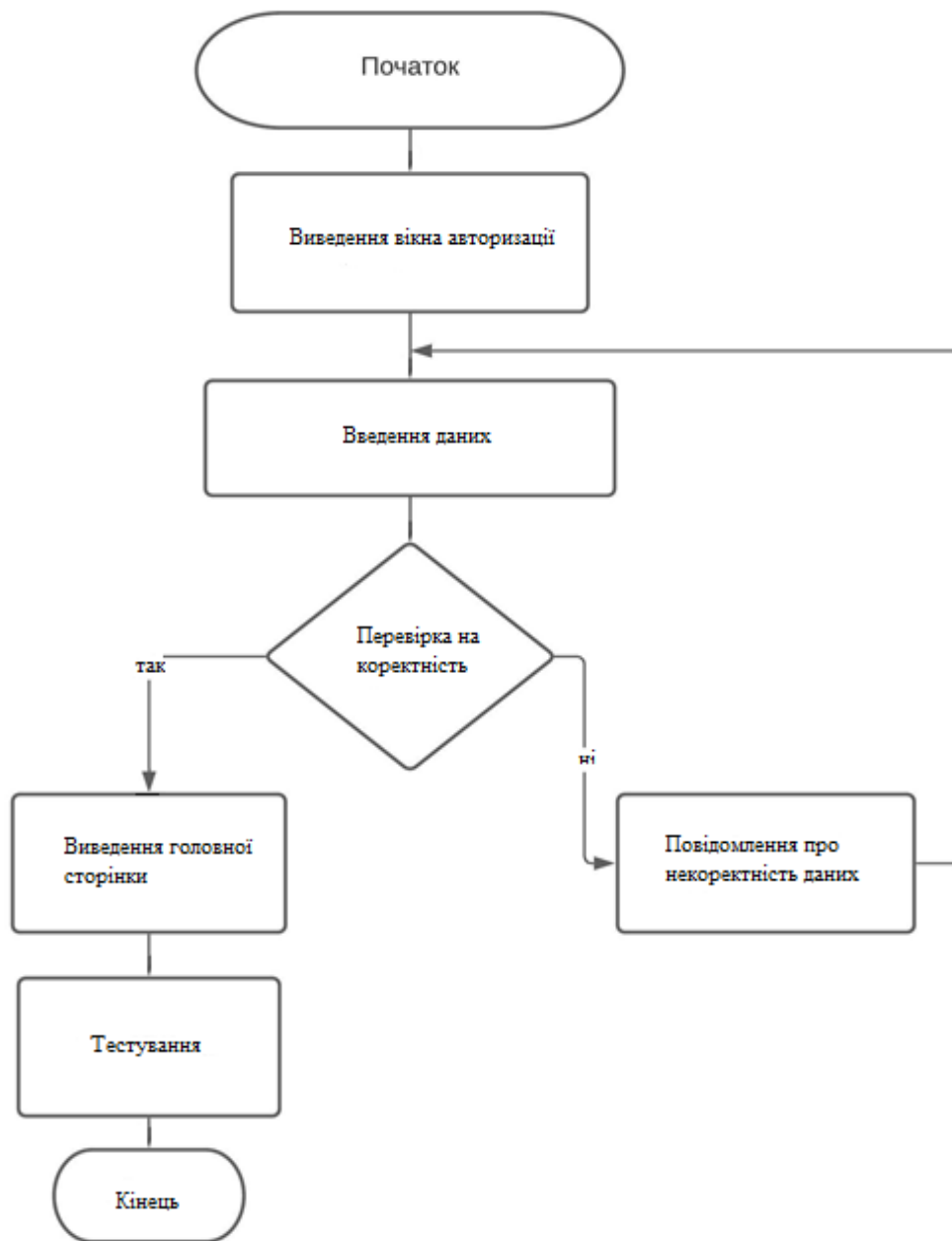


Рисунок 2.2 — Схема загального алгоритму функціонування системи

3 ПРОГРАМНА РЕАЛІЗАЦІЯ УДОСКОНАЛЕНОГО МОДУЛЯ ДЛЯ ВИЗНАЧЕННЯ РІВНЯ ПРОФЕСІЙНИХ ЗНАНЬ

3.1 Обґрунтування вибору програмних засобів для реалізації програмного модуля для визначення рівня професійних знань

IntelliJ IDEA — це інтегроване середовище розробки для багатьох мов програмування. Наприклад, Java, JavaScript, Python, розроблені JetBrains.

IntelliJ IDEA — це технологічно просунутий набір тісно інтегрованих інструментів розробки, який включає інтелектуальний редактор вихідного тексту з розширеними інструментами автоматизації, потужні інструменти рефакторингу коду, вбудовану підтримку технологій J2EE, механізми інтеграції з тестовим середовищем і версію керування Ant/JUnit системи, унікальна оптимізація та перевірка коду, а також інноваційний візуальний дизайнер графічних інтерфейсів.

IntelliJ IDEA в основному розглядається як середовище розробки для Java. Середовище найкраще працює з такою мовою [12]. Не менш важливою є підтримка таких передових технологій, як Groovy, Scala та ін.

Вони поєднують різні можливості, як-от Java з функціями Ruby, Smalltalk тощо.

Такі середовища, як Eclipse і NetBeans, також використовуються для розробки Java. IntelliJ IDEA був обраний тому, що він найбільш гнучкий і зручний для розробки веб-додатків.

Технологія Spring Framework

Spring зазвичай описують як полегшену структуру для створення Java—проектів. Spring можна використовувати для створення будь-яких Java—додатків (автономних, веб-додатків або додатків Java Enterprise Edition (JEE)), на відміну від багатьох інших фреймворків (таких як Apache Struts, який обмежується створенням лише веб-додатків). Фреймворк Spring є легким, оскільки для повного використання переваг ядра Spring потрібні мінімальні зміни в коді програми.

Spring Framework — це контейнер реалізації залежностей із кількома зручними рівнями (доступ до бази даних, аспектно—орієнтоване програмування,

RPC, веб-інфраструктура MVC). Це дозволяє швидше та зручніше розробляти програми Java.

Spring має модульну структуру, яка дозволяє комбінувати ті модулі, які потрібні для конкретного застосування. Саме такий підхід допоміг Spring випередити свого конкурента EJB і захопити лідерство. Spring Framework складається з таких модулів, як доступ до даних; мережа; ядро та інші.

Ядро Spring Framework базується на принципі інверсії управління (IoC). Це означає, що створення та керування залежностями між компонентами стає зовнішнім.

Реалізація DI Spring заснована на ключових концепціях Java, таких як компоненти та інтерфейси JavaBean. Використовуючи Spring як постачальника DI, ви можете гнучко визначати конфігурації залежностей у своїх програмах різними способами (файли XML, класи конфігурації Java, анотації в коді або новий метод визначення компонента Groovy). JavaBeans (POJO) надають стандартний механізм для створення настроюваних ресурсів Java різними способами, такими як конструктори та сетери. Spring використовує специфікацію JavaBean для створення ядра своєї конфігураційної моделі DI [13-18].

Інтерфейси та DI є взаємовигідними технологіями. Звичайно, проектування та написання коду інтерфейсу призначене для гнучких програм, але досить складно зв'язати програми, які використовують інтерфейси, і це вимагає від програмістів більшої роботи з програмування. Використання DI дозволяє скоротити кількість написаного коду до мінімуму при розробці додатків на основі інтерфейсів. Використання інтерфейсів дозволяє Spring використовувати динамічні проксі JDK (шаблон проксі) для надання потужних концепцій, таких як AOP для наскрізної функціональності [13-18].

WebStorm — це інтегроване середовище розробки HTML, CSS, TypeScript і JavaScript, створене JetBrains, успадковане від платформи IntelliJ IDEA. WebStorm постачається з попередньо встановленими плагінами JavaScript (наприклад, Node.js), які безкоштовно доступні для PhpStorm. WebStorm — це спеціалізована версія PhpStorm, яка надає деякі його функції. WebStorm підтримує CoffeeScript,

TypeScript, JavaScript і навіть Dart. Ключовою перевагою WebStorm IDE є робота з проектами (включаючи рефакторинг коду JavaScript в різні файли та каталоги проекту, а також вбудовування в HTML). WebStorm забезпечує автозаповнення, аналіз коду поля, навігацію по коду, рефакторинг, налагодження та інтеграцію з системами управління активами. Підтримуються кілька вкладень (якщо HTML-документ містить вбудований Javascript, вбудований в інший HTML-код із вбудованим Javascript) — такі конструкції підтримують правильний рефакторинг. Основні характеристики Індивідуальна інтеграція з Subversion, Git, GitHub, Perforce, Mercurial і CVS із можливістю перегляду змін і змін, що очікують на розгляд Інтеграція з системами налагодження Редагування файлів .css, html, .js під час перегляду результатів (редагування в реальному часі, деякі джерела виклику ця функція «пряме редагування файлів», «у реальному часі» або «без перезавантаження сторінки») Віддалене розгортання через FTP, SFTP, підключені мережеві диски тощо з автоматичною синхронізацією.

Веб-підтримка: Angular, React, Vue.js. Для сервера: Node.js, Meteor.

LiveEdit — це функція WebStorm, випущена з версії 5, яка дозволяє одночасно редагувати код HTML, CSS або JavaScript і бачити, як результат відображається в браузері. Ця функція вимагає підтримки браузера, тому WebStorm встановить плагін Google Chrome під час встановлення [19].

3.2 Обґрунтування вибору мови програмування для реалізації програмного модуля для визначення рівня професійних знань

Було проаналізовано мови програмування JavaScript та Python.

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

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

JavaScript використовується для:

- написання скриптів для сайтів, забезпечення їх інтерактивності;
- створення односторінкових і прогресивних веб-додатків (React, AngularJS, Vue.js);
- серверне програмування (Node.js (Express.js));
- стаціонарні програми (Electron, NW.js);
- мобільні додатки (React Native, Cordova);
- скрипти в програмах (наприклад, Adobe Creative Suite або Apache JMeter);
- всередині документів PDF.

Переваги JavaScript:

- швидкість.

JavaScript може бути дуже швидким, оскільки він часто запускається безпосередньо в браузері клієнта. Поки не потрібні зовнішні ресурси, JavaScript не сповільнюється викликами серверного сервера. Крім того, усі основні браузери підтримують JIT-компіляцію JavaScript (точно вчасно), що означає, що немає необхідності компілювати код перед його запуском.

- простота

Синтаксис JavaScript натхненний Java, і його відносно легко вивчити порівняно з іншими популярними мовами, такими як C++.

- популярність

JavaScript є всюди в Інтернеті, і 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 проєктів JavaScript, і популярність, яку він здобув за останні роки, лише зростатиме.

— сумісність

На відміну від PHP або інших мов сценаріїв, JavaScript можна вставити в будь-яку веб-сторінку. JavaScript можна використовувати в багатьох різних типах програм, підтримуючи інші мови, такі як Perl і PHP.

— завантаження сервера. JavaScript працює на стороні клієнта, тому, як правило, зменшується потреба в серверах, і простим програмам може взагалі не знадобитися сервер.

— розширені інтерфейси

JavaScript можна використовувати для створення 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-сайтів, написавши фрагменти JavaScript для додатків сторонніх розробників, таких як Greasemonkey.

— універсальність

Існує багато способів використання JavaScript на серверах Node.js. Якщо ви хочете завантажити Node.js з Express, використовувати базу даних документів, як-от MongoDB, і використовувати JavaScript на інтерфейсі, ви можете розробити цілу програму JavaScript від початку до кінця, використовуючи лише JavaScript. Оновлення 3 ECMAScript 5 (специфікація сценаріїв, на якій покладається JavaScript) ECMA International щорічно оновлює JavaScript. Наразі ми підтримуємо браузер для ES6 у 2017 році та сподіваємося підтримувати ES7 у майбутньому.

Python — це інтерпретована об'єктно-орієнтована мова програмування високого рівня зі строгою динамічною типізацією. Високорівневі структури даних разом із динамічною семантикою та динамічним зв'язуванням роблять його привабливим для швидкої розробки додатків і як спосіб підключення існуючих компонентів. Python підтримує модулі та пакети модулів, що сприяє модульності та повторному використанню коду. Інтерпретатор Python і стандартні бібліотеки доступні як у скомпільованій формі, так і у вихідному вигляді на всіх основних

платформах. Мова програмування Python підтримує кілька парадигм програмування, включаючи об'єктно-орієнтоване, процедурне, функціональне та аспектно-орієнтоване.

Його основні переваги:

- чистий синтаксис

Для виділення блоків потрібно використовувати відступи.

- портативність програми (характерна для більшості мов перекладу)

Стандартний дистрибутив містить велику кількість корисних модулів (включаючи модуль для створення графічного інтерфейсу);

- можливість використовувати python в режимі діалогу

Дуже корисно для експериментів і вирішення простих задач.

- стандартний дистрибутив має просте, але в той же час досить потужне середовище розробки, яка називається IDLE і яка написана на Python;

- зручний для розв'язування математичних задач

Має інструменти для роботи з комплексними числами, може працювати з цілими числами будь-якого розміру, може використовуватися як потужний калькулятор у діалоговому режимі.

- з відкритим кодом (може редагуватися іншими користувачами). Python має потужні високорівневі структури даних і простий, але ефективний підхід до об'єктно-орієнтованого програмування.

Елегантний синтаксис Python, динамічна обробка типів і той факт, що це інтерпретована мова, роблять його ідеальним для створення сценаріїв і швидкої розробки додатків у багатьох галузях промисловості на більшості платформ.

Інтерпретатор Python і обширну стандартну бібліотеку (як вихідні, так і двійкові дистрибутиви для всіх основних операційних систем) можна завантажити з веб-сайту Python www.python.org [архівовано 17 квітня 2018 року на Wayback Machine.] і вільно поширювати. Цей же сайт містить дистрибутиви та посилання на багато модулів, програм, інструментів і додаткової документації.

Інтерпретатор Python можна розширити функціями та типами даних, розробленими на C або C++ (або будь-якій іншій мові, яку можна викликати з

C). Python також є зручною мовою розширення для програм, які потребують подальшого налагодження.

Отже, в результаті обґрунтувань було вибрано мову JavaScript.

3.3 Обґрунтування вибору фреймворку для реалізації програмного модуля інтерактивного тестування знань

Framework — програмна платформа, яка визначає структуру програмної системи; програмне забезпечення для полегшення розробки та інтеграції різних компонентів великого програмного проекту.

Фреймворк відрізняється від поняття бібліотеки тим, що бібліотека може використовуватися в програмному забезпеченні просто як набір підпрограм із близькою функціональністю, не впливаючи на архітектуру програмного забезпечення та не накладаючи на неї жодних обмежень. У той час як «фреймворк» диктує правила побудови архітектури програми, встановлюючи поведінку за замовчуванням на початковому етапі розробки - «фреймворк», який доведеться розширювати і змінювати відповідно до конкретних вимог. Приклад програмної основи — C.M.F. (Content Management Framework), а прикладом бібліотеки є модуль електронної пошти.

Фреймворк додатки. Одне з головних переваг при використанні «каркасних» додатків — «стандартність» структури додатки. "Каркаси" стали популярними з появою графічних інтерфейсів користувача, які мали тенденцію до впровадження стандартної структури для додатків. З їх використанням стало набагато простіше створювати засоби для автоматичного створення графічних інтерфейсів, так як структура внутрішньої реалізації коду програми стала відома заздалегідь. Для забезпечення каркаса зазвичай використовуються техніки об'єктно-орієнтованого програмування (наприклад, частини програми можуть успадковуватися від базових класів фреймворку).

Крім того, на відміну від бібліотеки, яка об'єднує набір близьких функцій, «скелет» може містити велику кількість бібліотек з різною тематикою.

Ключова відмінність між «фреймворком» і бібліотекою полягає в інверсії управління: код користувача викликає функції бібліотеки (або класів) і отримує керування при виклику. У «фреймворку» код користувача може реалізовувати певну поведінку, яка вбудована в більш загальний — «абстрактний» код фреймворку. При цьому «фреймворк» викликає функції (класи) коду користувача.

Створення веб-додатків без використання фреймворків частково означає згоду дотримуватися цих платформ і не використовувати роботу команд розробників фреймворків, які часто тісно співпрацюють з розробниками браузерів.

Angular 2+:

Сильні сторони:

Головною перевагою Angular 2+ є його популярність. Можна сказати, що ім'я Google асоціюється з ним, і це впливає на його сприйняття. Angular 1 швидко став популярним, оскільки люди з інших середовищ розробки виявили, що це знайомий шаблон MVC для розробки односторінкових програм. Після модернізації Angular 1 і редизайну деяких частин фреймворку Angular 2+ буквально злетів. Кількість офіційних і неофіційних тренувань на ньому вражає. На ринку є високий попит на розробників Angular. Крім того, це одна з небагатьох фреймворків, розглянутих у цій статті, яка має офіційний набір багатофункціональних компонентів для створення інтерфейсів користувача.

Слабкі сторони та можливі труднощі впровадження:

Ми вважаємо, що Angular зосереджується на створенні інтерфейсів користувача для односторінкових програм і не відповідає потребам розробників великих проєктів. Це може ускладнити підтримку проєктів, якщо базові принципи, на яких вони базуються, не будуть чітко сформульовані на початку. На практиці розробникам доводиться вдаватися до чудес винахідливості, щоб змусити додаток Angular виконувати речі, які не є частиною фреймворку. Це ще більше зменшує інтерес розробників до TypeScript, на якому написаний фреймворк.

Якщо архітектура вашого веб-додатку відповідає шаблону MVC, ви також можете розглянути Angular 2+ React + Redux.

Сильні сторони:

Головною перевагою React і Redux є їх відносна простота і те, що вони спрямовані на вирішення однієї задачі — розробки інтерфейсів. Якщо ви націлитесь на те, щоб знайти щось, що робить щось одне, але робить це добре, ви можете сказати, що обидві бібліотеки чудові у тому, що від них очікується. Хоча підхід до контейнера з підтримкою стану може здатися комусь незнайомим, більшість розробників можуть легко зрозуміти цю концепцію та зрозуміти переваги архітектури одностороннього потоку даних і те, як такий підхід може спростити програми зі складним інтерфейсом користувача.

Слабкі сторони та можливі труднощі впровадження:

Найбільшими недоліками React і Redux є не особливості реалізації того, що вони можуть робити, а те, що вони не можуть. Щоб створити складний веб-додаток, вам знадобиться багато інших технологій. Після того, як ви відійдете від основних функцій React, Redux та кількох інших бібліотек, вам доведеться багато думати про «правильні інструменти» з незліченною кількістю рішень і шаблонів, які іноді легко інтегрувати у вашу програму, а іноді ні. .

Отже, оскільки React і Redux — це бібліотеки, які зосереджені на розв'язанні вузького кола спеціалізованих завдань, недосвідченим командам дуже легко побудувати на їх основі щось непідтримуване, не знаючи, що рішення, які вони приймають, призводять до низької продуктивності веб-додатків. або до помилок. Навіть досвідчені розробники можуть зіткнутися з тим, що відсутність чіткого архітектурного планування рішення або жорстких правил на початку розробки може дуже неприємно позначитися на проекті в майбутньому. Легко обдурити, подумавши, що використання React і Redux у вашій організації заощадить ваш час і ресурси, що зменшить проблеми з продуктивністю розробки. Без ретельно розроблених угод і стандартизації інших бібліотек і шаблонів перехід на React і Redux — це все одно, що сказати: «Ми переходимо на JavaScript, щоб писати програми та покращувати продуктивність».

Якщо ви перебуваєте в ситуації, коли шукаєте хороші бібліотеки, а не якийсь комплексний фреймворк, можливо, вам підійде комбінація React + Redux. Також будьте реалістичними щодо можливостей вашої команди та організації не лише на початковому етапі розробки, а й з огляду на довгострокову підтримку програми.

Vue.js:

Сильні сторони:

Мабуть, головною перевагою цього фреймворку є можливість його поступового впровадження. Vue характеризується чіткою та раціональною архітектурою, яку легко освоїти та застосувати на практиці.

Існує згуртована спільнота ентузіастів і сторонніх проєктів, які роблять Vue.js ще цікавішим. Крім того, різні рішення, орієнтовані на Vue, можна досить легко комбінувати в більш складні рішення під час створення нових проєктів.

Слабкі сторони та можливі труднощі впровадження:

Спроба розмежувати ідеї додатків на основі MVC і додатків на основі контейнерів із збереженням стану може заплутати. Здається, розробники фреймворків хочуть робити все правильно, не віддаючи перевагу одному шаблону розробки додатків над іншим. Ми вважаємо це, м'яко кажучи, заплутаним для тих, хто шукає Vue.js як платформу для повномасштабного веб-рішення, і може призвести до використання різних шаблонів, ускладнюючи підтримку програми.

Однією з головних проблем Vue.js є те, що проєкт залежить від однієї людини. Звичайно, інші фреймворки теж від когось залежать, але зазвичай це організації. Навколо Vue.js сформувалося широке співтовариство, є багато інноваційних проєктів надбудов, але основна технологія повністю лежить на плечах одного розробника.

Ми вважаємо, що було б круто побачити, як Vue.js інтегрує нові стандарти, але зараз ми бачимо щось на кшталт шаблону веб-компонента, але не самі веб-компоненти, тому мова йде більше про плани щодо впровадження нових технологій, ніж про те, що вже реалізовано. Чому вибрати Vue.js?

Якщо у вас є старіша веб-програма, якій потрібен більш надійний і сучасний механізм, Vue.js може стати для вас рішенням. Він містить чіткі шаблони, і навіть якщо команда розробників не дуже досвідчена, вони зможуть легко відокремити правильний підхід від неправильного. Хоча Vue не має готової до використання стандартної інтерфейсної бібліотеки, існує багато сторонніх фреймворків на основі Vue, які можуть бути вам корисними.

Dojo 2:

Сильні сторони:

Dojo заповнює багато прогалів, які існують у чомусь на зразок React + Redux. Це пояснюється тим, що його творці намагаються створити повне середовище розробки, яке враховує шаблон реактивних компонентів, побудованих на архітектурі контейнерів із збереженням стану.

Дизайн Dojo 2 враховує той факт, що він не існує у вакуумі. Він включає в себе можливість імпорту та експорту веб-компонентів і створений з урахуванням існування різних сценаріїв використання, які необхідно обробляти, але в той же час він забезпечує структуровану структуру з деякими функціями. Крім того, в основному функціоналі Dojo 2 велике значення надається модульності компонентів платформи. Dojo 2 пропонує рішення для багатьох поширених завдань і функцій, важливі для повномасштабних веб-програм, яким не приділяється багато уваги в більшості інших фреймворків. Зокрема, існує система інтернаціоналізації та шаблони для забезпечення доступності додатків. Крім того, існує підтримка тем і шаблонів, які зосереджені на аспектах розробки поза TypeScript/JavaScript, таких як робота з такими ресурсами, як CSS.

Dojo 2 має на меті забезпечити структуроване середовище розробки, зручне для розробників. Використовуючи TypeScript і різноманітні шаблони, він намагається дати програмістам своєрідний орієнтир, який направлятиме їх на шлях продуктивної роботи, не обмежуючи при цьому тих, хто точно знає, що робить. Фреймворк розроблений, щоб зробити програмування продуктивнішим і безпечнішим. Його мета — дати можливість командам розробників швидко створювати кращі веб-додатки.

Слабкі сторони та можливі труднощі впровадження:

Проект знаходиться в стадії розробки, тягнеться надовго, що гальмує розвиток Dojo 2. Є багато інших, більш вузьких фреймворків, творці яких завдяки спеціалізації можуть швидше публікувати нові релізи.

Проблема може полягати в тому, що прагнення до гнучкості та взаємозамінності компонентів може поставити під сумнів сам сенс існування Dojo 2, не пояснюючи чітких причин, чому його варто використовувати.

Отже, проаналізувавши дані фреймворки, для розробки модуля було обрано фреймворк Vue.js.

3.4 Розробка модуля тестування

Модуль результатів призначений для перегляду результатів тестування користувачів.

Модуль виконує такі дії:

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

Алгоритм роботи модуля результатів зображений на рисунку 3.1.



Рисунок 3.1 — Алгоритм роботи модуля результатів

3.5 Опис програмних рішень

Ви повинні мати дозвіл на роботу на сайті. Для цього заповніть відповідні поля на сторінці авторизації. Потім дані повинні бути перевірені, і користувач отримує відповідні дозволи.

Спочатку користувач вводить логін і пароль у форму авторизації, якщо дані введені неправильно, то система за допомогою технології надсилає користувачу повідомлення про помилку без оновлення сторінки. Якщо дані введені правильно, вони надсилаються на сервер для перевірки правильності даних з його боку. У разі неправильних даних система повідомляє користувача про помилку авторизації. Якщо дані вірні, запитується збіг логіна та пароля, а потім перевіряється наявність користувача в системі. Якщо користувача немає в системі, він отримує помилку

авторизації. Якщо він існує, він отримує статус «авторизований». Алгоритм авторизації показаний на рисунку 3.2.



Рисунок 3.2 — Діаграма діяльності сервера для авторизації

Зображений вище алгоритм можна записати у вигляді коду:

```

app.post("/login", (req, res) => {
  if (req.session.userID) {
    res.redirect("/");
  }
  if (!req.body.email && !req.body.password) {
    res.json({
      error: "All fields are required"
    });
  }
});
  
```

```
    }  
else  
{  
    db.getUsers().then(loggingin => {  
        for (var i = 0; i < loggingin.length; i++) {  
            if (loggingin[i].email == req.body.email) {  
                var compare = loggingin[i].hashed_password;  
                var userID = loggingin[i].id;  
            }  
        }  
        if (!compare)  
{  
            res.json({  
                error: "Invalid username or password"  
            });  
        }  
    }  
else  
{  
    db.checkPassword(req.body.password, compare)  
        .then(passwordMatch => {  
            console.log("is the password correct?: ", passwordMatch);  
            if (passwordMatch)  
{  
                req.session.userID = userID;  
                console.log("userIDinlogin: ", userID);  
                // console.log("loggingin: ", loggingin);  
                res.json(loggingin);  
            }  
        }  
else  
{
```

```
res.json({
  error: "Invalid username or password" //both have the same error message.
```

This prevents attackers from enumerating valid usernames without knowing their passwords.

```
    });
  }
})
.catch(err => {
  console.log(err);
});
}
}
);
}
}
);
```

Аутентифікація паролем.

Цей метод покладається на те, що користувач повинен надати ім'я користувача та пароль для успішної ідентифікації та автентифікації в системі. Пара ім'я користувача/пароль встановлюється користувачем під час реєстрації в системі, і ім'ям користувача може бути адреса електронної пошти користувача.

Для веб-додатків існує кілька стандартних протоколів автентифікації пароля, які ми обговоримо нижче.

Аутентифікація NTTP.

Цей протокол, описаний у стандартах NTTP 1.0/1.1, існує вже дуже давно і досі активно використовується в корпоративному середовищі. Для веб-сайтів це працює так:

Коли неавторизований клієнт отримує доступ до захищеного ресурсу, сервер надсилає NTTP-статус «401 Unauthorized» і додає заголовок «WWW—Authenticate», що вказує на схему автентифікації та параметри.

Отримавши таку відповідь, браузер автоматично відображає діалогове вікно для введення імені користувача та пароля. Користувач вводить дані свого облікового запису.

У всіх наступних запитах до веб-сайту браузер автоматично додає HTTP-заголовок «Авторизація», в якому дані користувача передаються на аутентифікацію серверу.

Сервер автентифікує користувача за допомогою цього заголовка. Рішення про надання доступу (авторизації) приймається окремо на основі ролі користувача, списку ACL або інших даних облікового запису.

Весь процес стандартизований і добре підтримується всіма веб-переглядачами та серверами. Існує кілька схем автентифікації, які відрізняються рівнем безпеки:

Базовий — це найпростіша схема, у якій ім'я користувача та пароль користувача надсилаються в заголовку авторизації в незашифрованому вигляді (закодовано base64). Однак використання HTTPS (HTTP через SSL) є відносно безпечним.

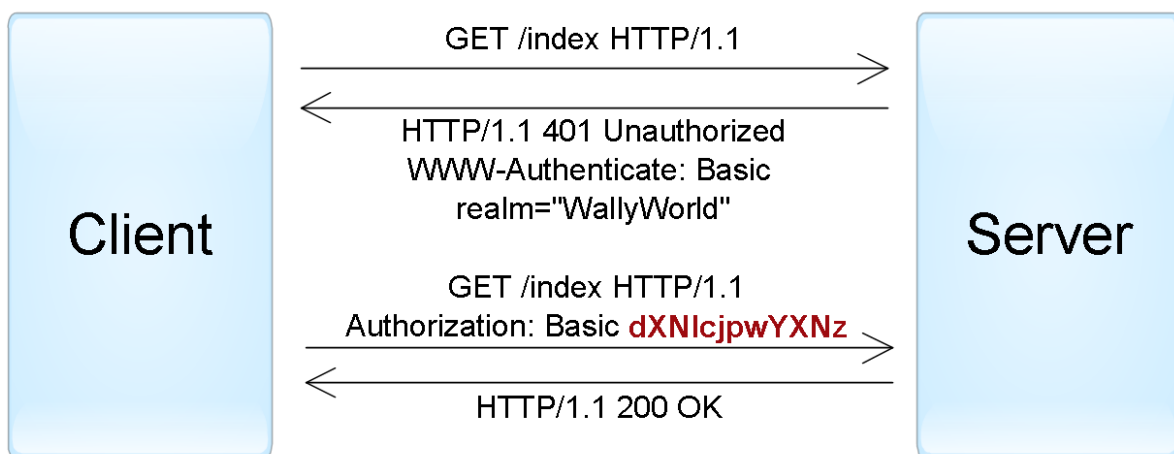


Рисунок 3.3 — Приклад аутентифікації HTTP за базовою схемою.

Дайджест — це схема виклик-відповідь, у якій сервер надсилає унікальний одноразовий номер, а браузер надсилає хеш MD5 пароля користувача, обчислений за допомогою певного одноразового номера. Більш безпечна альтернатива базовій

схемі для незахищених з'єднань, але вразлива до атак типу «людина посередині» (схема замінена на базову). Крім того, використання цієї схеми не дозволяє використовувати сучасні хеш-функції для зберігання паролів користувачів на сервері.

NTLM (також відомий як автентифікація Windows) також покладається на підхід виклик-відповідь, коли пароль не надсилається явно. Ця схема не є стандартом HTTP, але підтримується більшістю браузерів і веб-серверів. В основному використовується для автентифікації користувачів Windows Active Directory у веб-додатках. Вразливий до атак з передачі хешу.

Negotiate — ще одна схема в сімействі автентифікації Windows, яка дозволяє клієнту вибирати між автентифікацією NTLM і Kerberos. Kerberos — це більш безпечний протокол, заснований на принципі єдиного входу. Однак це може працювати, лише якщо клієнт і сервер знаходяться в області інтрамережі та є частиною домену Windows. Варто зауважити, що під час використання автентифікації HTTP користувач не має стандартної можливості вийти з веб-програми, окрім закриття всіх вікна браузера.

4. ТЕСТУВАННЯ РОЗРОБЛЕНОГО УДОСКОНАЛЕНОГО ПРОГРАМНОГО МОДУЛЯ ДЛЯ ВИЗНАЧЕННЯ РІВНЯ ПРОФЕСІЙНИХ ЗНАНЬ ТА АНАЛІЗ РЕЗУЛЬТАТІВ ЙОГО РОБОТИ

Для розробленого програмного модуля були проведені тести на визначення рівня професійних знань, які підтвердили правильність його функціонування та відповідність вимогам, зазначеним у завданні.

Додаток протестовано 120 разів. Завдяки цьому можна повною мірою оцінити роботу, виявити недоліки, які необхідно доопрацювати в майбутньому.

Випробування проводилися на ноутбучі, на якому було створено додаток.

Щоб розпочати роботу з додатком, необхідно авторизуватися (рис. 3.2).

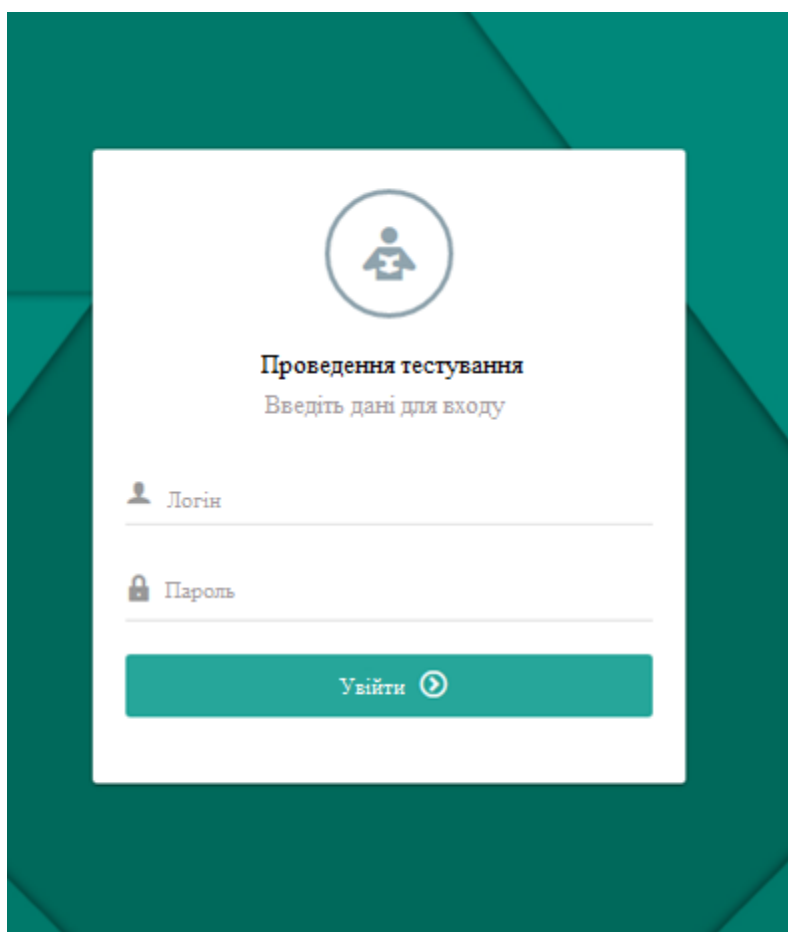


Рисунок 3.2 — Вікно авторизації користувача

Після авторизації відкриється домашня сторінка користувача, на якій автоматично розпочнеться тестування (рис. 3.3). Тести мають чотири варіанти

відповіді, лише одна з яких правильна. Після вибору правильної відповіді тест автоматично задає наступне запитання. Крім того, після натискання кнопки відповіді система автоматично вкаже, чи була відповідь правильною. Якщо відповідь неправильна, система вкаже правильну відповідь.

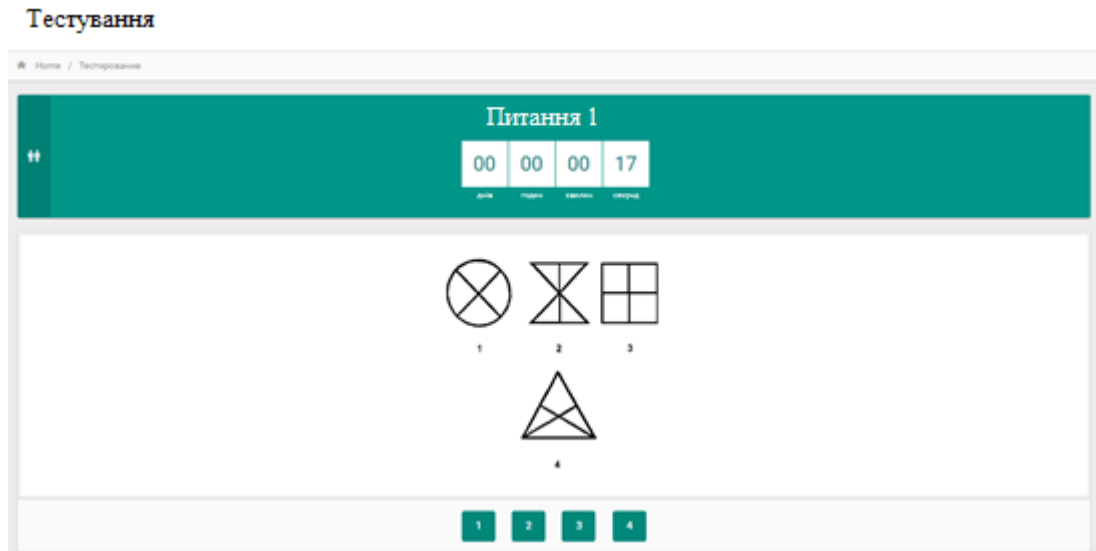


Рисунок 3.3 — Проходження тесту

Після завершення тесту відобразиться вікно з інформацією про успішність: описано кількість правильних/неправильних відповідей на запитання та приблизний рівень IQ (рис. 3.4).

Результати тестування

✓ Вітаємо! Ваш рівень IQ становить 76!

№	Питання	Відповідь
1	1	Стат: 3, Неправильно Правильно: 4
2	2	Стат: 3, Неправильно Правильно: 3
3	3	Стат: 3, Правильно
4	4	Стат: 3, Неправильно Правильно: 4
5	5	Стат: 3, Неправильно Правильно: 4
6	6	Стат: 3, Неправильно Правильно: 4
7	7	Стат: 3, Неправильно Правильно: 1

Рисунок 3.4 — Результати тестування

Випробування розробленого додатку пройшли успішно. У процесі тестування в програмному забезпеченні не виявлено помилок. Перевірені користувачі відзначили, що додаток простий у використанні, інтуїтивно зрозумілий, а дизайн стриманий і лаконічний.

У таблиці 3.1 порівнюються результати тестування з аналогічними програмами.

Таблиця 3.1 — Порівняння результатів тестування

	Інсталяція	Наявність інструкції	Вибір мови інтерфейсу	Реєстрація	Швидкість завантаження додатку
Розроблений додаток	ні	ні	ні	ні	0,70 сек
IQ Test 2.2	ні	ні	ні	ні	1,0 сек
Тест Айзенка	так	так	ні	так	1,25 сек
Тест Равена	ні	ні	ні	ні	1,15 сек
Iq-test 2021	ні	так	так	так	1,2 сек

Отже, після тестування розробленого додатку можна сказати, що він працює швидко, жодних помилок не виявлено; порівняно з аналогічними програмами показала хороші результати за характеристиками. Користувачі помітили, що програма проста та інтуїтивно зрозуміла у використанні. За обсягом ОП розроблений модуль використовує близько 30 Мб, що є одним із найменших значень, а за швидкістю завантаження випереджає інших на 0,3 с і більше.

5 РОЗРАХУНОК ЕКОНОМІЧНОЇ ДОЦІЛЬНОСТІ СТВОРЕННЯ УДОСКОНАЛЕНОГО ПРОГРАМНОГО МОДУЛЯ ДЛЯ ВІЗНАЧЕННЯ РІВНЯ ПРОФЕСІЙНИХ ЗНАНЬ

5.1 Комерційний та технологічний аудит науково-технічної розробки.

Метою даного розділу є проведення технологічного аудиту, в даному випадку нового програмного продукту удосконаленого модулю для визначення рівня професійних знань. Особливістю програми є те, що дана технологія розширює функціональні можливості, збільшує швидкодію, надає можливість швидкої перевірки на професійні здібності, можливість автоматизованої перевірки результатів тестування, підвищення об'єктивності оцінювання та зменшення упередженості при оцінюванні знань та виводу правильних відповідей на екран.

Аналогом може бути Classmarker, ціна — 1440 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 табл. 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 динамікою	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 і з 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

Продовження табл. 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	ПІБ експертів		
	Експерт 1	Експерт 2	Експерт 3
	Бали		
Технічна здійсненність концепції	3	3	4
Наявність аналогів на ринку	3	4	4
Цінова політика	4	4	4
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	42	41	43
Середньоарифметична сума балів	$(42+41+43) / 3 = 42$		

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 дана технологія розширює функціональних можливості, збільшує швидкодію, надає можливість швидкої перевірки на професійні здібності, можливість автоматизованої перевірки результатів тестування, підвищення об'єктивності оцінювання та зменшення упередженості при оцінюванні знань та виводу правильних відповідей на екран.

5.2 Прогнозування витрат на виконання науково-дослідної (дослідно-конструкторської) 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, 20 днів;

t — число днів роботи розробника (дослідника).

Результати розрахунків зведемо до таблиці 5.1.

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	30000	1500,00	35	52500,000
Програміст	28000	1400,00	35	49000,000
Всього				101500,00

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

Додаткова заробітна плата прийнято розраховувати як 14 % від основної заробітної плати розробників та робітників:

$$Z_d = Z_o \cdot 14 \% / 100 \% \quad (5.2)$$

$$Z_d = (101500,00 \cdot 14 \% / 100 \%) = 14210,00 \text{ (грн.)}$$

Нарахування на заробітну плату розробників.

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

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

$$H_z = (101500,00 + 14210,00) \cdot 22 \% / 100 \% = 25456,20 \text{ (грн.)}$$

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 23000 грн., термін його корисного використання згідно податкового законодавства – 2 роки, а термін його фактичного використання – 1,75 міс.

$$A_{\text{обл}} = \frac{23000}{2} \times \frac{1,75}{12} = 1677,08 \text{ грн.}$$

Аналогічно визначаємо амортизаційні витрати на інше обладнання та приміщення. Розрахунки заносимо до таблиці 5.2.

Але, так як вартість ліцензійної ОС та спеціалізованих ліцензійних нематеріальних ресурсів є безкоштовно (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 (Asus K53)	23000	2	1,75	1677,083
Офісне обладнання (меблі)	20000	4	1,75	729,167
Приміщення	900000	20	1,75	6562,500
Всього				8968,75

Тарифи на електроенергію для побутових 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. $\Pi = 0,3$ кВт;

Φ — фактична кількість годин роботи обладнання, годин.

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

$$V_e = 0,9 \cdot 0,3 \cdot 8 \cdot 35 \cdot 6,2 = 468,72 \text{ (грн.)}$$

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:

$$I_e = (Z_o + Z_p) \cdot \frac{N_{iv}}{100\%}, \quad (5.6)$$

де N_{iv} — норма нарахування за статтею «Інші витрати».

$$I_e = 101500,00 \cdot 77\% / 100\% = 78155 \text{ (грн.)}$$

До статті «Накладні (загальновиробничі) 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:

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

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

$$H_{нзв} = 101500,00 * 117 \% / 100 \% = 118755 \text{ (грн.)}$$

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

$$B_{заг} = 101500,00 + 14210,00 + 25456,20 + 8968,75 + 468,72 + 78155 + \\ + 118755 = 347513,67 \text{ грн.}$$

Розрахунок загальних витрат на науково-дослідну (науково-технічну) роботу та оформлення її результатів.

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

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

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:

$$ЗВ = 347513,67 / 0,5 = 695027 \text{ грн.}$$

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:

а) вказати, з якого часу можуть бути впроваджені результати науково-технічної розробки;

б) 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);

в) 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;

г) 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.

Розробка чи суттєве вдосконалення програмного засобу (програмного забезпечення, програмного продукту) для використання масовим споживачем.

В цьому випадку 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, $\Pi_0 = \Pi_0 \pm \Delta\Pi_0$;

Π_0 — 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 550 грн. за одиницю виробу, термін збільшення прибутку складе 3 роки. Після завершення розробки і її

вдосконалення, можна буде підняти її ціну на 50 грн. Кількість одиниць реалізованої продукції також збільшиться: протягом першого року — на 25000 шт., протягом другого року — на 20000 шт., протягом третього року на 15000 шт. До моменту впровадження результатів наукової розробки реалізації продукту не було:

$$\Delta\Pi_1 = (0*50 + (550 + 50)*25000)*0,8333*0,32*(1 - 0,18) = 3006666,546 \text{ грн.}$$

$$\Delta\Pi_2 = (0*50 + (550 + 50)*(25000+20000))*0,8333*0,32*(1 - 0,18) = 5903999,764 \text{ грн.}$$

$$\Delta\Pi_3 = (0*50 + (550 + 50)*(25000+20000+15000))*0,8333*0,32*(1 - 0,18) = 7871999,685 \text{ грн.}$$

Отже, комерційний ефект від реалізації результатів розробки за три роки складе 16782666,00 грн.

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:

$$ПП = (3006666,546/(1+0,1)^1) + (5903999,764/(1+0,1)^2) + (7871999,685/(1+0,1)^3) = 2733333,22 + 4879338,648 + 5914349,876 = 13527021,75 \text{ грн.}$$

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

$$PV = k_{инв} * ZB, \quad (5.12)$$

де $k_{инв}$ — 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 $k_{инв}=2...5$, але може бути і більшим;

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

$$PV = 2 * 695027 = 1390054,68 \text{ грн.}$$

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

$$E_{абс} = ПП - PV, \quad (5.13)$$

$$E_{абс} = 13527021,75 - 1390054,68 = 12136967,07 \text{ грн.}$$

Оскільки $E_{абс} > 0$ 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_g . Для цього використаємо формулу:

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

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

$$E_g = \sqrt[3]{(1 + 12136967,07/1390054,68) - 1} = 1,135$$

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

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

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

$$T_{ок} = 1 / 1,135 = 0,88 \text{ р.}$$

Оскільки $T_{ок} < 3$ -х років, а саме термін окупності рівний 0,88 роки, то фінансування даної наукової розробки є доцільним.

Висновки до розділу: економічна частина даної роботи містить розрахунок витрат на розробку нового програмного продукту, сума яких складає 695027 гривень. Було спрогнозовано орієнтовану величину витрат по кожній з статей витрат. Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення, розраховано період окупності витрат для інвестора та економічний ефект при використанні даної розробки. В результаті аналізу розрахунків можна зробити висновок, що розроблений програмний продукт за ціною дешевший за аналог і є висококонкурентоспроможним. Період окупності складе близько 0,88 роки.

ВИСНОВКИ

Під час виконання магістерської кваліфікаційної роботи реалізовано удосконалений модуль визначення рівня професійних знань. Для цього було проведено аналіз подібних програм, визначено їх переваги та недоліки. На цій основі визначаються мета, предмет, предмет і завдання дослідження.

Для розробки програмного модуля визначення рівня професійних знань було вирішено використати архітектурний шаблон MVC, що дозволило його адаптацію в залежності від отриманих даних.

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 модуля для визначення рівня професійних .

У третьому розділі магістерської роботи визначені функції програмного модуля для визначення рівня професійних знань. Визначено модулі, які виконують дані функції. Створено схему взаємодії основних складових програмного модуля алгоритми для визначення рівня професійних знань. Обґрунтовано вибір мови програмування, середовища розробки, бази даних, фреймворку.

У четвертому розділі магістерської роботи було протестовано можливості програмного модуля для визначення рівня професійних 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 для визначення рівня професійних 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 для визначення рівня професійних 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. Троян І. Ю., Азаров О. Д. Розробка програмного модуля для визначення рівня професійних знань [Електронний ресурс]. – Режим доступу до ресурса: <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-21/paper/view/12686/>. – Назва з екрану.
2. Тест [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/Тест> . – Назва з екрану.
3. Daniel K. B. Judgment under Uncertainty: Heuristics and Biases – Cambridge: University Press, 2005. – 555 с. ISBN 978-0-521-28414-1
4. Bayes L., Thomas G., Price R., Philosophical Transactions of the Royal Society. – London: Kraken Opus, 2017. – 418 с.
5. Гайна Г.А. – Основи проектування баз даних: Навчальний посібник. – Київ: КНУБА, 2005. – 204 с.
6. Джудит С. К., Боуманб С. Л. Практическое руководство по SQL. – Москва: Вильямс, 2001. – 352 с.
7. Майоров А. Н. Теория и практика создания тестов для системы образования. – Москва. 2000.
8. IQ Test 2.2 [Електронний ресурс]. – Режим доступу: <https://ru.iq-test.cc/>. – Назва з екрану.
9. Тест Айзенка [Електронний ресурс]. – Режим доступу: <https://brainapps.ru/iqtest-eysenck>. – Назва з екрану.
10. Тест на определение уровня IQ Равена [Електронний ресурс]. – Режим доступу: <https://testometrika.com/blog/a-test-to-determine-the-iq-level-of-ravena/>. – Назва з екрану.
11. Тест на візуально просторові здібності [Електронний ресурс]. – Режим доступу: <https://onlinetestpad.com/ru/test/25025-test-na-vizualno-prostranstvennyye-sposobnosti>. – Назва з екрану.
12. Бахвалов Н. С. Численные методы. – Москва: Наука, 2001. – 631 с.

13. UML diagrams [Електронний ресурс]. – Режим доступу: https://www.tutorialspoint.com/uml/uml_class_diagram.htm. / – Назва з екрану.
14. ReactJs [Електронний ресурс]. – Режим доступу: <https://ru.reactjs.org/>. – Назва з екрану.
15. MongoDB [Електронний ресурс]. – Режим доступу: <https://docs.mongodb.com/>. – Назва з екрану.
16. Переваги MongoDB [Електронний ресурс]. – Режим доступу: <https://echo.lviv.ua/dev/9693/>. – Назва з екрану.
17. nodeJs [Електронний ресурс]. – Режим доступу до ресурсу: <https://nodejs.org/ru/docs/>. – Назва з екрану.
18. Переваги та недоліки GraphQL [Електронний ресурс]. – Режим доступу: <https://codeguida.com/post/1549>. – Назва з екрану.
19. Порівняльна характеристика фреймворків для створення користувацького інтерфейсу [Електронний ресурс]. – Режим доступу до ресурсу: <https://mkdev.me/posts/sravnenie-javascript-freymvorkov-vue-js-react-i-angular-2019/>. – Назва з екрану.

ДОДАТОК А

Міністерство освіти та науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ

_____ проф., д.т.н. О. Д. Азаров

«__» _____ 2022 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання магістерської кваліфікаційної роботи
«Програмний модуль інтерактивного тестування знань»
08-23.МКР.030.00.000 ТЗ

Науковий керівник: проф. каф ОТ

_____ Азаров О.Д.

Магістрант групи 2КІ-21м

_____ Троян І.Ю.

Опонент: д.т.н., проф. каф ПЗ,

_____ Яремчук Ю . Є

Вінниця 2022

1 Підстава для виконання магістерської кваліфікаційної роботи (МКР)

1.1 Актуальність дослідження пов'язана з необхідністю підвищення об'єктивності та зменшення упередженості під час оцінювання знань. Об'єктивність найкраще досягається під час інтерактивного тестування з можливістю автоматизованої генерації тесту без повторень запитань, можливістю встановлювати складність завдання та можливістю контролювати співвідношення теоретичних та практичних запитань.

1.2 Наказ про затвердження теми магістерської кваліфікаційної роботи.

2 Мета і призначення МКР

2.1 Мета магістерської роботи полягає у покращенні алгоритму автоматизованої генерації тесту.

2.2 Призначення розробки — виконання магістерської кваліфікаційної роботи.

3 Вихідні дані для виконання МКР

Виконати розробку програмного модуля інтерактивного тестування знань. Схему взаємодії модулів, діаграми активності інтерактивного тестування, діаграми активності модулів та лістинги програми представити в додатках до роботи.

4 Вимоги до виконання МКР

МКР повинна задовольняти такі вимоги:

- запропонувати програмний модуль інтерактивного тестування знань;
- розробити алгоритм та структуру для програмного модуля інтерактивного тестування знань;
- розробити функціонал для створення тестів, тестових завдань та перегляду результатів тестування користувачів;

- забезпечення зручного та легкого для користувача графічного інтерфейсу;
- результат роботи, а саме веб-додаток інтерактивного тестування.

5 Етапи МКР та очікувані результати в таблиці А.1

Таблиця А.1 — Етапи виконання роботи

№	Назва етапу	Термін виконання		Очікувані результати
		початок	кінець	
1	Аналіз завдання. Вступ	01.09.21	02.09.21	Вступ
2	Аналіз літературних джерел	03.09.21	09.09.21	Розділ 1
3	Розробка технічного завдання	10.09.2021	24.09.21	Технічне завдання
3	Розробка структури удосконаленого модуля для визначення рівня професійних знань	25.09.21	08.10.21	Розділ 2, розробка структури
4	Розробка веб-додатку, проектування веб-додатку	11.10.21	29.10.21	Розділ 3, розробка програми
5	Практична реалізація, результати.	01.11.21	14.11.21	Розділ 3
6	Розробка економічної частини	15.11.21	30.11.21	Розділ 4
7	Оформлення пояснювальної записки	02.12.21	15.12.21	ПЗ, презентація

6 Матеріали, що подаються до захисту МКР

До захисту МКР подаються: пояснювальна записка МКР, ілюстративні та графічні матеріали, протокол попереднього захисту МКР на кафедрі, відзив наукового керівника, відзив рецензента, протоколи складання державних екзаменів, анотації до МКР українською та іноземною мовами, довідка про відповідність оформлення МКР діючим вимогам.

7 Порядок контролю виконання та захисту МКР

Виконання етапів розрахункової та графічної документації МКР контролюється науковим керівником згідно зі встановленими термінами. Захист МКР відбувається на засіданні Державної екзаменаційної комісії, затвердженою наказом ректора.

8.1 При оформлювання МКР використовуються:

— ДСТУ 3008: 2015 «Звіти в сфері науки і техніки. Структура та правила оформлювання»;

— ДСТУ 8302: 2015 «Бібліографічні посилання. Загальні положення та правила складання»;

— міждержавний ГОСТ 2.104-2006 «Єдина система конструкторської документації. Основні написи»;

— Методичні вказівки до виконання магістерських кваліфікаційних робіт зі спеціальності 123 — «Комп'ютерна інженерія». Кафедра обчислювальної техніки ВНТУ 2022;

— документами на які посилаються у вище вказаних.

8.2 Порядок виконання МКР викладено в «Положення про кваліфікаційні роботи на другому (магістерському) рівні вищої освіти СУЯ ВНТУ-03.02.02-П.001.01:21».

9 Вимоги щодо технічного захисту інформації в МКР з обмеженим доступом відсутні.

Технічне завдання до виконання отримав _____ Троян І.Ю.

ДОДАТОК В

Використання архітектурного шаблону MVC

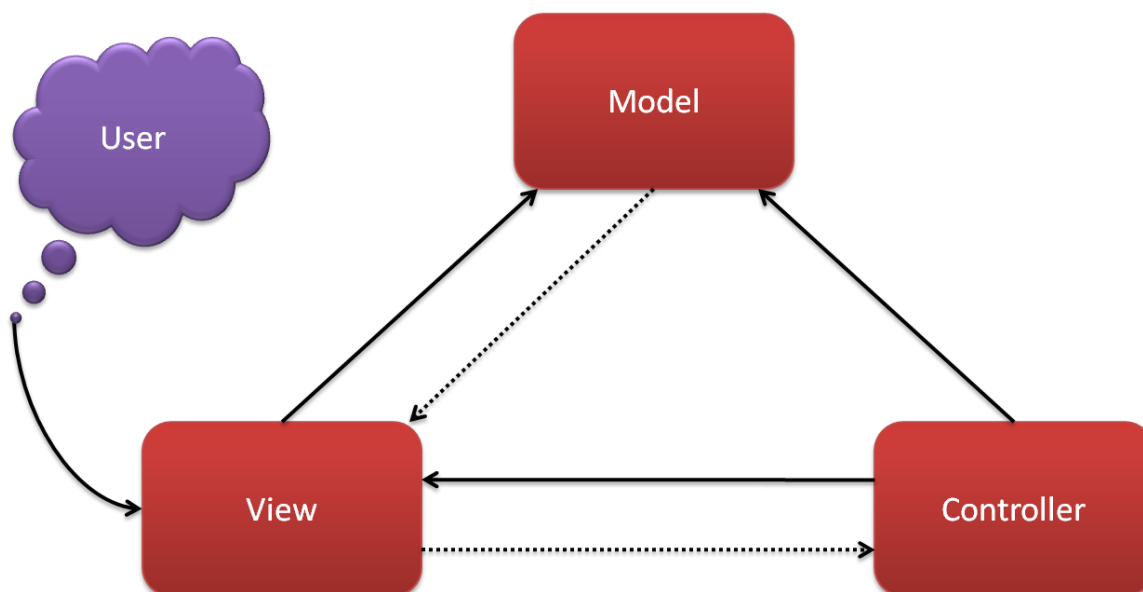


Рисунок Б.1 — Використання архітектурного шаблону MVC

ДОДАТОК В

Схема загального алгоритму функціонування системи

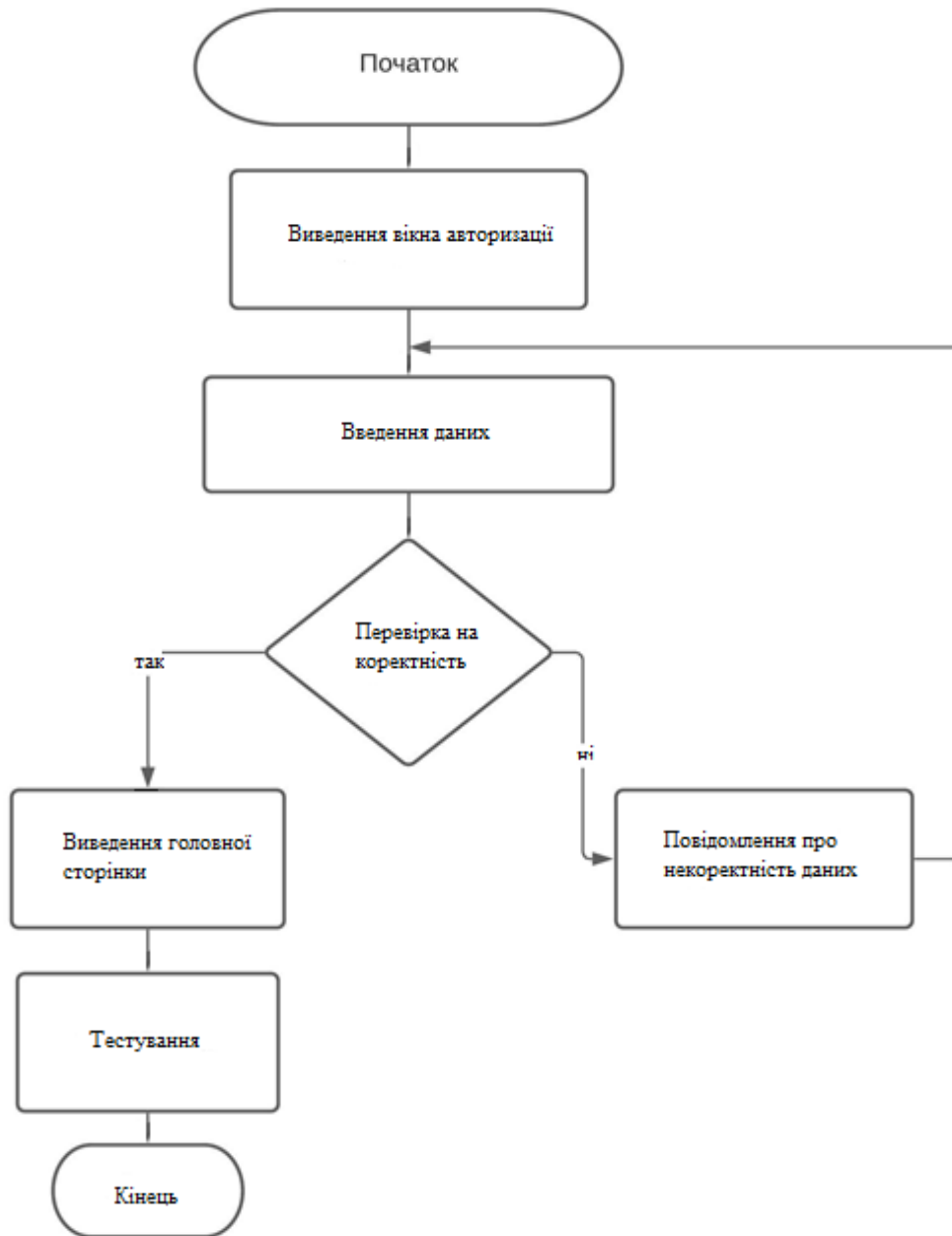


Рисунок В.1 — Схема загального алгоритму функціонування системи

ДОДАТОК Г

Схема роботи сервера авторизації



Рисунок Г.1 — Схема роботи сервера авторизації

ДОДАТОК Д**Алгоритм модуля тестування**

Рисунок Д.1 — Алгоритм модуля тестування

ДОДАТОК Е

Приклад аутентифікації HTTP за базовою схемою

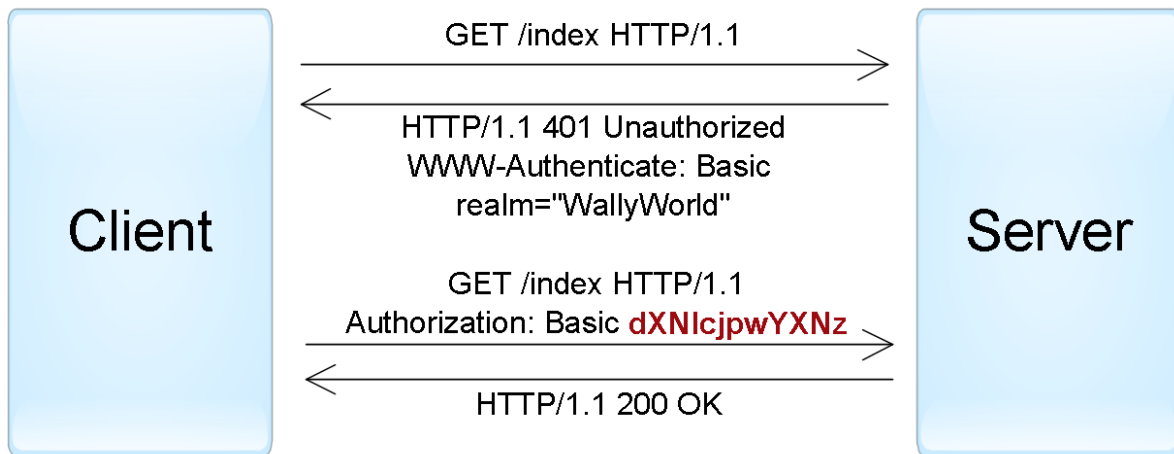


Рисунок Е.1 — Приклад аутентифікації HTTP за базовою схемою

ДОДАТОК Ж

Лістинг програми

Авторизація

```
const express = require("express");
const app = express();
const compression = require("compression");
//
const bodyParser = require("body-parser");
const ca = require("chalk-animation");
const db = require("./db/db");
const cookieSession = require("cookie-session");
const cookieParser = require("cookie-parser");
const csrf = require("csrf");
const path = require("path");

//

app.use(bodyParser.urlencoded({ extended: false }));
app.use(compression());
app.use(cookieParser());
app.use(
  // goes before the csrf
  cookieSession({
    secret: `I'm always angry.`,
    maxAge: 1000 * 60 * 60 * 24 * 14
  })
);
app.use(bodyParser.json());
app.use(express.static("public"));
```

```
app.use(csrf());
app.use(function(req, res, next) {
  res.cookie("mytoken", req.csrfToken());
  next();
});

//

if (process.env.NODE_ENV !== "production") {
  app.use(
    "/bundle.js",
    require("http-proxy-middleware")({
      target: "http://localhost:8081/"
    })
  );
} else {
  app.use("/bundle.js", (req, res) => res.sendFile(`${__dirname}/bundle.js`));
}

// app.get("/", (req, res) => {

// }); //for heroku, but how?

app.post("/login", (req, res) => {
  if (req.session.userID) {
    res.redirect("/");
  }
  if (!req.body.email && !req.body.password) {
    res.json({
      error: "All fields are required"
    });
  }
});
```

```

    });
  }
Else
{
  db.getUsers().then(loggingin => {
    for (var i = 0; i < loggingin.length; i++) {
      if (loggingin[i].email == req.body.email) {
        var compare = loggingin[i].hashed_password;
        var userID = loggingin[i].id;
      }
    }
    if (!compare) {
      res.json({
        error: "Invalid username or password"
      });
    } else {
      db.checkPassword(req.body.password, compare)
        .then(passwordMatch => {
          console.log("is the password correct?: ", passwordMatch);
          if (passwordMatch) {
            req.session.userID = userID;
            console.log("userIDinlogin: ", userID);
            // console.log("loggingin: ", loggingin);
            res.json(loggingin);
          }
        }
    }
  }
Else
{
  res.json({

```

error: "Invalid username or password" //both have the same error message.

This prevents attackers from enumerating valid usernames without knowing their passwords.

```

    });
  }
})
.catch(err => {
  console.log(err);
});
}
});
}
});

app.post("/registration", (req, res) => {
  console.log("inside POST /registration", req.body);
  if (req.session.userID) {
    res.redirect("/");
  }

  if (
    !req.body.firstname ||
    !req.body.lastname ||
    !req.body.email ||
    !req.body.password
  ) {
    res.json({
      error: "All fields are required"
    });
  }
}

```



```
else
{
  db.getUsers().then(loggingin => {
    for (var i = 0; i < loggingin.length; i++) {
      if (loggingin[i].email == req.body.email) {
        var compare = true;
      }
    }
    if (compare) {
      res.json({
        error: "That email is already taken"
      });
    } else {
      db.hashPassword(req.body.password).then(hashedException => {
        db.register(
          req.body.firstname,
          req.body.lastname,
          req.body.email,
          hashedPassword
        )
        .then(signedup => {
          req.session.userID = signedup.id;
          console.log("userID-registration: ", req.session.userID);
          res.json({ success: true });
        })
        .catch(err => {
          console.log(err);
        });
      });
    }
  });
}
```

```
});
}
});
```

```
app.get("/welcome", (req, res, next) => {
  console.log("req.session: ", req.session);
  if (req.session.userID) {
    // if (!req.session.userID) { if I try this it doesn't work
    res.redirect("/"); //I have two, registration and homepage... here it would make sense
    to be the homepage(because req.session.userID = true), but at welcome#/ it shows the
    registration page...
  } else {
    res.sendFile(__dirname + "/index.html");
  }
});
```

```
app.get("/user", (req, res) => {
  console.log("user1");
  if (!req.session.userID) {
    res.redirect("/");
  }
  db.getUserById(req.session.userID)
    .then(data => {
      console.log("user2: ", data);
      // res.json({
      //   ...data
      // });
      res.json(data);
    })
    .catch(err => {
```

```
    console.log("logging error", err);
    res.sendStatus(500);
  });
});

app.get("/logout", (req, res) => {
  req.session = null;
  res.redirect("/");
});

app.get("/questions", function(req, res) {
  console.log("questionsIQ: ", req.session.userID);
  db.getListOfQuestions().then(results => {
    // console.log("getListOfQuestions: ", results);
    res.json({ results });
  });
});

app.post("/finalscore", (req, res) => {
  console.log("final score");
  console.log("req.session.userID", req.session.userID);
  console.log("req.body: ", req.body.score); //req.body because it comes from an axios
  db.insertFinalScore(req.session.userID, req.body.score)
    .then(results => {
      res.json({
        success: true
      });
    })
    .catch(err => {
      res.sendStatus(500);
    });
});
```

```
    console.log(err);
  });
});
app.get("*", function(req, res) {
  if (!req.session.userID) {
    res.redirect("/welcome");
  }
  else
  {
    res.sendFile(__dirname + "/index.html");
  }
});
app.listen(process.env.PORT || 8080, () =>
  ca.rainbow("listening on port 8080")
);
```

