

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерних наук

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему Застосування адаптивного підходу для реалізації системи
опитувань та тестувань

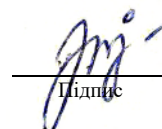
Галузь знань 12 – Інформаційні технології

Шифр і назва галузі знань

Спеціальність 122 – Комп'ютерні науки

Шифр і назва спеціальності

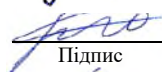
Виконала: студентка 2 курсу, група КНм-20-1


Підпис

P.I. Галкіна

Ініціали, прізвище

Керівник: к.т.н., доцент кафедри КНІТ


Підпис

P.O. Багрій

Ініціали, прізвище

Нормоконтроль: к.т.н., доцент кафедри КНІТ


Підпис

P.O. Багрій

Ініціали, прізвище

До захисту допускаю:

Зав. кафедри КН, д.т.н., професор

7 грудня 2021 р.


Підпис

O.V. Бармак

Ініціали, прізвище

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій

Кафедра комп'ютерних наук

Освітній ступінь магістр

Галузь знань 12 – Інформаційні технології

Спеціальність 122 – Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук


(підпис)

д.т.н., професор О.В. Бармак

« 1 » вересня 2021 року

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

1. Тема кваліфікаційної роботи магістра: «Застосування адаптивного підходу для реалізації системи опитувань та тестувань»
2. Завдання видано студенту Галкіній Руслані Ігорівні
(прізвище, ім'я, по батькові)
3. Керівник роботи к.т.н., доцент Багрій Руслан Олександрович
(прізвище, ім'я, по батькові)
4. Затверджені наказом університету від « 25 » серпня 2021 р. № 102
5. Зміст пояснювальної записки (перелік задач) та вихідні дані:

Мета роботи – створення інформаційної системи для проведення опитувань та тестувань, що надасть можливість зменшити час, витрачений на проведення тестування, отримання більш точних результатів тестування та спрощення процесу перевірки результатів. Для досягнення мети необхідно дослідити існуючі підходи до рішення задачі, що базуються на традиційному тестуванні та передумови використання адаптивного тестування.

Реферат

Кваліфікаційна робота магістра присвячена розробці інформаційної системи для проведення опитувань та тестувань з використанням адаптивного підходу.

Актуальність теми. У результаті розробки величезної кількості нових інформаційних систем сьогодні проводиться глобальна автоматизація та комп'ютеризації майже усіх процесів, що оточують людину задля вдосконалення якості її життя. Такого роду оптимізація процесів також позитивно впливає і на економію часу, що є найважливішим ресурсом людини. Проведення соціальних опитувань чи освітянських тестувань було завжди актуальною темою. Тому автоматизація цього процесу, а також його вдосконалення ніколи не були зайвими.

За класичного тестування проходження тесту, а також перевірка його результатів завжди займали великий обсяг часу, особливо до впровадження автоматизованих систем. Результати опитувань містять інформацію та актуальні думки людей про обрану тему, а результати тестів - оцінку якості досліджуваного об'єкту. Ці елементи є цінними під час впровадження змін для вдосконалення того чи іншого процесу.

Наразі процеси опитування та тестування (як автоматизовані, так і звичайні) використовують традиційне тестування, не враховуючи жодні вхідні дані, такі як рівень володіння інформацією опитуваного та його попередні відповіді, що можуть виключати цінність наступних запитань.

Завдяки сучасним підходам можна вдосконалити цей процес, а саме динамічно враховувати цінність кожного наступного питання, що дозволить ефективно використовувати час як опитуваного, так і перевіряючого.

Мета і задачі роботи. Мета магістерської роботи полягає у реалізації інформаційної системи для проведення опитувань та тестувань з використанням адаптивного підходу.

Для досягнення поставленої мети визначені наступні задачі:

- дослідити існуючі методи та підходи для проведення опитувань та тестувань;
- вдосконалити процес проведення опитувань та тестувань шляхом динамічної зміни наступного питання та їх загальної кількості;
- спроектувати інформаційну систему для проведення опитувань та тестувань з використанням адаптивного підходу;
- забезпечити отримання результатів опитування чи тестування у формі, придатної для подальшого проведення аналізу.
- здійснити порівняння підходів при проведенні класичного та адаптивного тестування.

Об'єкт дослідження – проведення тестування та опитувань у комп'ютерних системах.

Предмет дослідження – методи проведення тестувань у комп'ютерних системах та інформаційна система для проведення опитувань та тестувань.

Наукова новизна одержаних результатів. В результаті проведеної роботи були отримані такі результати:

1. Вдосконалено метод проведення тестувань та опитувань з використанням адаптивного підходу, що дало можливість зменшити кількість запитань в порівнянні з традиційними підходами без втрати точності отриманих результатів.

2. Розроблено інформаційну систему для проведення опитувань та тестувань з застосуванням адаптивного підходу, що дало можливість скоротити час проходження, а також усунути надлишкові варіанти відповідей до запитань.

3. Досліджено практичну ефективність використання адаптивного підходу під час проведення тестувань, що дозволило оцінити затрачений час на проведення тестування та якість отриманих результатів.

Практичне значення одержаних результатів. На основі результатів досліджень, отриманих в результаті виконання дипломної роботи було розроблено відповідне програмне забезпечення з використанням адаптивного підходу. Застосування інформаційної системи надає можливість проведення тестування в

менші проміжки часу, спрощує процес перевірки шляхом виключення надлишкових відповідей та дає інформацію щодо вдосконалення теми проведення тестування.

Апробація результатів дипломної роботи магістра та публікації:

Доповідь на тему «Застосування адаптивного підходу для реалізації системи опитувань та тестувань» на XIII Всеукраїнській науково-практичній конференції «Актуальні проблеми комп'ютерних наук АПКН-2021» (Хмельницький, 15-16 жовтня 2021 року, Хмельницький національний університет).

За темою дипломної роботи магістра автором виконана *наукова публікація*.

Структура та обсяг роботи. Дипломна робота магістра складається з завдання, реферату, змісту, переліку скорочень, вступу, 4 розділів, висновків, переліку посилань із 45 найменувань. Загальний обсяг дипломної роботи магістра становить 102 сторінка, з них 75 сторінок основного тексту та 23 сторінки додатків. У роботі наведено 33 рисунки та 10 таблиць.

Ключові слова: інформаційна система, методи тестувань, адаптивний підхід.

Зміст

Перелік скорочень.....	4
Вступ.....	5
Розділ 1	8
Аналіз сучасного стану використання інформаційних систем тестових завдань .	8
1.1 Аналіз предметної області	8
1.2 Аналіз методів та підходів проведення опитувань та тестувань	15
1.3 Аналіз видів вебдодатків.....	17
1.4 Аналіз існуючого програмного забезпечення предметної області.....	20
1.4.1 Google forms.....	20
1.4.2 Survio	22
1.4.3 SurveyMonkey	25
1.5 Постановка задачі.....	26
Висновки до розділу 1	27
Розділ 2	28
Розробка інформаційної системи для проведення опитувань та тестувань.....	28
2.1 Загальний опис інформаційної системи проведення опитувань та тестувань	28
2.2 Методи проведення тестувань	30
2.2.1 Традиційний та класичний метод проведення опитувань та тестувань.....	31
2.2.2 Розширення традиційного методу проведення опитувань та тестувань – адаптивний метод.....	32
Висновки до розділу 2	34
Розділ 3	35
Проектування інформаційної системи проведення опитувань та тестувань.....	35

3.1 Основні вимоги для проектування інформаційної технології	35
3.2 Проектування загальної схеми інформаційної системи.....	35
3.3 Використання адаптивного підходу для підбору наступного питання та визначення складності тесту	37
3.4 Проектування структури бази даних	39
3.5 Проектування інтерфейсу інформаційної системи.....	43
Висновки до розділу 3	46
Розділ 4	47
Програмна реалізація.....	47
4.1 Обґрунтування вибору мови програмування для розробки інформаційної системи	47
4.2 Розробка інформаційної системи для проведення опитувань та тестувань ..	53
4.3 Апробація інформаційної системи	68
Висновки до розділу 4	72
Висновки	73
Перелік посилань.....	75

ДОДАТКИ

Перелік скорочень

Скорочення, термін, позначення	Пояснення
ІС	Інформаційна система
ОС	Операційна система
ПЗ	Програмне забезпечення
МСП	Малі та середні підприємства
JS	JavaScript
JVM	Java virtual machine
WORA	Write once, run anywhere
IQ	Intelligence quality
SAT	Scholastic Aptitude Test
ITC	International Test Commission

Вступ

Кваліфікаційна робота магістра присвячена розробці інформаційної системи для проведення опитувань та тестувань з використанням адаптивного підходу.

Актуальність теми. У результаті розробки величезної кількості нових інформаційних систем сьогодні проводиться глобальна автоматизація та комп'ютеризації майже усіх процесів, що оточують людину задля вдосконалення якості її життя. Такого роду оптимізація процесів також позитивно впливає і на економію часу, що є найважливішим ресурсом людини. Проведення соціальних опитувань чи освітянських тестувань було завжди актуальною темою. Тому автоматизація цього процесу, а також його вдосконалення ніколи не були зайвими.

За класичного тестування проходження тесту, а також перевірка його результатів завжди займали великий обсяг часу, особливо до впровадження автоматизованих систем. Результати опитувань містять інформацію та актуальні думки людей про обрану тему, а результати тестів - оцінку якості досліджуваного об'єкту. Ці елементи є цінними під час впровадження змін для вдосконалення того чи іншого процесу.

Наразі процеси опитування та тестування (як автоматизовані, так і звичайні) використовують традиційне тестування, не враховуючи жодні вхідні дані, такі як рівень володіння інформацією опитуваного та його попередні відповіді, що можуть виключати цінність наступних запитань.

Завдяки сучасним підходам можна вдосконалити цей процес, а саме динамічно враховувати цінність кожного наступного питання, що дозволить ефективно використовувати час як опитуваного, так і перевіряючого.

Мета і задачі роботи. Мета магістерської роботи полягає у реалізації інформаційної системи для проведення опитувань та тестувань з використанням адаптивного підходу.

Для досягнення поставленої мети визначені наступні задачі:

- дослідити існуючі методи та підходи для проведення опитувань та тестувань;
- вдосконалити процес проведення опитувань та тестувань шляхом динамічної зміни наступного питання та їх загальної кількості;
- спроектувати інформаційну систему для проведення опитувань та тестувань з використанням адаптивного підходу;
- забезпечити отримання результатів опитування чи тестування у формі, придатної для подальшого проведення аналізу.
- здійснити порівняння підходів при проведенні класичного та адаптивного тестування.

Об'єкт дослідження – проведення тестування та опитувань у комп'ютерних системах.

Предмет дослідження – методи проведення тестувань у комп'ютерних системах та інформаційна система для проведення опитувань та тестувань.

Наукова новизна одержаних результатів. В результаті проведеної роботи були отримані такі результати:

1. Вдосконалено метод проведення тестувань та опитувань з використанням адаптивного підходу, що дало можливість зменшити кількість запитань в порівнянні з традиційними підходами без втрати точності отриманих результатів.

2. Розроблено інформаційну систему для проведення опитувань та тестувань з застосуванням адаптивного підходу, що дало можливість скоротити час проходження, а також усунути надлишкові варіанти відповідей до запитань.

3. Досліджено практичну ефективність використання адаптивного підходу під час проведення тестувань, що дозволило оцінити затрачений час на проведення тестування та якість отриманих результатів.

Практичне значення одержаних результатів. На основі результатів досліджень, отриманих в результаті виконання дипломної роботи було

розроблено відповідне програмне забезпечення з використанням адаптивного підходу. Застосування інформаційної системи надає можливість проведення тестування в менші проміжки часу, спрощує процес перевірки шляхом виключення надлишкових відповідей та дає інформацію щодо вдосконалення теми проведення тестування.

Апробація результатів дипломної роботи магістра та публікації:

Доповідь на тему «Застосування адаптивного підходу для реалізації системи опитувань та тестувань» на XIII Всеукраїнській науково-практичній конференції «Актуальні проблеми комп'ютерних наук АПКН-2021» (Хмельницький, 15-16 жовтня 2021 року, Хмельницький національний університет).

За темою дипломної роботи магістра автором виконана *наукова публікація*.

Структура та обсяг роботи. Дипломна робота магістра складається з завдання, реферату, змісту, переліку скорочень, вступу, 4 розділів, висновків, переліку посилань із 45 найменувань. Загальний обсяг дипломної роботи магістра становить 102 сторінка, з них 75 сторінок основного тексту та 23 сторінки додатків. У роботі наведено 33 рисунки та 10 таблиць.

Ключові слова: інформаційна система, методи тестувань, адаптивний підхід

Розділ 1

Аналіз сучасного стану використання інформаційних систем тестових завдань

1.1 Аналіз предметної області

Сьогодні ніхто не здивується використанню технологій у тій чи іншій сфері. Тим паче їх віддаленого використання через мережу Інтернет. Все більше користувачів надають перевагу автоматизованим процесам, на відміну від класичного проведення діяльності.

Застосування технологій для підвищення комфорту стало одною із актуальних тем, проте також і з'явилась необхідність проведення тестування якості нововведень. Одним із найпоширеніших видів отримання користувацьких думок – проведення опитувань.

Опитування – це метод збору соціологічної інформації про досліджуваний об'єкт під час безпосереднього (усне опитування, інтерв'ю) або опосередкованого (письмове опитування, анкетування) спілкування того хто опитує з респондентом. Опитування бувають соціологічні, політологічні, маркетингові, психологічні – залежно від предмету дослідження [1].

Проте, якщо мова йде про оцінку якості отримання інформації, то найпоширенішим методом є тестування.

Тестування застосовується для визначення відповідності предмета випробування заданим специфікаціям. До завдань тестування не належить визначення причин невідповідності заданим вимогам. Тестування – один з розділів діагностики.

Тестування застосовується в техніці, медицині, психіатрії, освіті для визначення придатності об'єкта тестування для виконання тих чи інших функцій. Якість тестування і достовірність його результатів значною мірою залежить від методів тестування та складу тестів [2].

В умовах впровадження карантину більшість процесів перейшли у дистанційний режим. У тому числі й надання психологічної допомоги, як процес, що не потребує очної присутності ні лікаря, ні пацієнта.

Психологічне тестування – термін психології, що позначає процедуру встановлення та вимірювання індивідуально-психологічних відмінностей. У вітчизняній психології також вживається термін «психодіагностичне обстеження». Основним інструментом психологічної діагностики є психологічний тест. Психологічне тестування використовується в різних сферах: профорієнтації, профвідбору, психологічному консультуванні, плануванні корекційної роботи, дослідницької діяльності і т.д. Правила організації і проведення психологічного тестування описуються в рамках відповідної галузі психології – психологічної діагностики [3].

Психологічні тести передбачають спостереження, зроблені на «ретельно відібраному зразку поведінки індивіда» [4]. Психологічний тест часто призначений для вимірювання неспостережуваних конструктів, також відомих як «латентні змінні». Психологічні тести можуть включати серію завдань або проблем, які респондент повинен вирішити. Психологічні тести можуть включати анкети та інтерв'ю, які також призначені для вимірювання неспостережуваних конструкцій. Шкали на основі анкетування та інтерв'ю зазвичай відрізняються від психоосвітніх тестів, які вимагають максимальної продуктивності респондента. Шкали на основі анкетування та інтерв'ю, навпаки, вимагають типову поведінку респондента як вхідні дані [5]. Тести на симптоми частіше називають шкалами. Корисний психологічний тест/шкала має бути як дійсним (тобто є докази, які підтверджують ідею, що тест вимірює те, що вон повинен, і «наскільки добре він це робить» [6]), і надійним (тобто він внутрішньо узгоджений або дає послідовні результати в часі, між оцінювачами тощо).

Існує кілька широких категорій психологічних тестів.

Тести досягнень – це тести, які оцінюють знання особи в предметній області. Тести академічних досягнень призначені для проведення кваліфікованим оцінювачем окремої особи або групи людей. Під час тестів досягнень особі, яку оцінюють, представляється серія тестових завдань. Вважається, що оцінка на тесті відображає досягнення зі шкільного предмета [4].

Багато тестів досягнень посилаються на норми. Відповіді людини оцінюються відповідно до стандартизованих протоколів, і результати можна порівняти з відповідями нормуючої групи після завершення тесту [4].

Деякі тести досягнень є критеріями, метою яких є з'ясувати, чи учасник тесту оволодів заздалегідь визначеним об'ємом знань, а не порівнювати його з усіма іншими, хто проходить тест. Тести досягнень часто протиставляють тестам, які вимірюють здібності, більш загальну і стабільну когнітивну рису [4].

Психологічні тести були розроблені для вимірювання конкретних здібностей, таких як канцелярські, перцептивні, числові або просторові здібності. Іноді ці тести повинні бути спеціально розроблені для певної роботи, але існують також тести, які вимірюють загальні канцелярські та механічні здібності або навіть загальні здібності до навчання. Прикладом тесту на професійні здібності є Міннесотський канцелярський тест, який вимірює швидкість і точність сприйняття, необхідні для виконання різних службових обов'язків. Широко використовуваним тестом на здібності в бізнесі є тест Wonderlic [7]. Існують здібності, які, як вважають, пов'язані з конкретними професіями і використовуються для профорієнтації, а також відбору та найму [8].

Дані свідчать про те, що тести на здібності, такі як тести IQ, чутливі до минулого навчання і не можуть уникнути вимірювання минулих досягнень, хоча колись вважалося, що вони вимірюють здібності без навчання [9]. Тест SAT, який раніше називався Scholastic Aptitude Test, змінив назву, оскільки результативність тесту чутлива до навчання.

Шкала ставлення оцінює настрої людини щодо події (наприклад, рішення Верховного суду), особи (наприклад, губернатора), концепції (наприклад, носіння масок під час пандемії), організації (наприклад, бойскаути) або об'єкта (наприклад, ядерна зброя) на одновимірному континуумі сприятливого та несприятливого ставлення. Шкали ставлення використовуються в маркетингу для визначення індивідуальних переваг щодо брендів. З плином часу соціальні психологи розробили шкали ставлення для оцінки ставлення індивідів до Організації Об'єднаних Націй і расових відносин [10]. Зазвичай шкали Лайкерта використовуються в дослідженні ставлення. Історично до розробки шкали Лайкерта [11] використовувалася шкала Терстона. Шкала Лайкерта значною мірою витіснила шкалу Терстона [12].

Нейропсихологічні тести покликані бути об'єктивним і стандартизованим виміром зразка поведінки. Пункти тестів, що відповідають нормам, були випробувані в групі нормування, і оцінки в тесті можна класифікувати як високі, середні або низькі, а також градації між ними [4]. Ці тести дозволяють вивчати індивідуальні відмінності. Оцінки в тестах досягнень, що відповідають нормам, пов'язані з процентильними рангами по відношенню до інших осіб, які відповідають віку або класу учасника тестування.

Нейропсихологічні тести – це спеціально розроблені завдання, які використовуються для вимірювання психологічної функції, яка, як відомо, пов'язана з певною структурою мозку або шляхом [13]. Тести використовуються для дослідження функцій мозку та в клінічних умовах для діагностики певних дефіцитів. Зазвичай вони передбачають систематичне адміністрування чітко визначених процедур у формальному середовищі. Нейропсихологічні тести зазвичай проводяться одній людині, яка працює з екзаменатором у тихому офісному середовищі, без відволікаючих факторів. Таким чином, можна стверджувати, що нейропсихологічні тести іноді дають оцінку максимального рівня когнітивної діяльності людини. Нейропсихологічні

тести є основним компонентом процесу проведення нейропсихологічної оцінки разом з особистісними, міжособистісними та контекстуальними факторами.

Більшість нейропсихологічних тестів, які зараз використовуються, засновані на традиційній психометричній теорії. У цій моделі необроблений результат тесту порівнюється з великою загальною нормативною вибіркою, яка в ідеалі має бути обрана з групи людей, подібних до людини, яку досліджують. Нормативні дослідження часто надають дані, розрізнені за віком, рівнем освіти та/або етнічної приналежністю – дослідження показало, що такі фактори впливають на результати окремих тестів. Це дозволяє порівняти результативність людини з відповідною контрольною групою і, таким чином, забезпечити справедливу оцінку її поточної когнітивної функції.

Аналіз широкого спектру нейропсихологічних тестів можна розділити на чотири категорії. По-перше, це аналіз загальної продуктивності або того, наскільки добре люди справляються від тесту до тесту, а також наскільки вони успішні у порівнянні із середнім балом. По-друге, це порівняння «лівого і правого»: наскільки добре людина виконує конкретні завдання, пов'язані з лівою та правою сторонами тіла. По-третє, це патогномічні ознаки або конкретні результати тесту, які безпосередньо стосуються певного розладу. Нарешті, остання категорія – це диференційні моделі, які зазвичай використовуються для діагностики конкретних захворювань або типів пошкоджень.

Особистісні тести оцінюють конструкції, які вважаються складовими особистості. Приклади конструкцій особистості включають риси великої п'ятірки, такі як інтроверсія-екстраверсія та сумлінність. Вважається, що особистісні конструкції мають вимір. При дослідженні та підборі працівників використовуються особистісні показники. Вони включають шкали самозвіту та звітів спостерігачів [14].

Існують тисячі розроблених *психологічних тестів*. Деякі були виготовлені комерційними тестовими компаніями, які платять за їх

використання. Інші були розроблені дослідниками, і їх можна знайти в академічній дослідницькій літературі. Тести для оцінки конкретних психологічних конструкцій можна знайти, провівши пошук у базі даних. Деякі бази даних мають відкритий доступ, інші є власністю та доступні через бібліотеку в університетах.

Існує багато різних типів оцінки особистості. Самозвіту включає в себе управління багатьма пунктами, що вимагає від респондентів інтроспективної оцінки власних характеристик особистості. Це дуже суб'єктивно, і через прозорість елементів такі показники Q-даних дуже сприйнятливі до спотворення мотивації та реакції [15]. Респонденти повинні вказати рівень своєї згоди з кожним пунктом, використовуючи шкалу Лайкерта або, точніше, шкалу типу Лайкерта[11]. Наприклад, елемент анкети особистості може попросити респондентів оцінити ступінь їхньої погодження з твердженням «Я розмовляю з багатьма різними людьми на вечірках» за шкалою від 1 («абсолютно не згоден») до 5 («абсолютно згодний»).

Значна кількість досліджень і роздумів було присвячено темі розвитку тесту особистості. Розробка особистісних тестів, як правило, є ітераційним процесом, в якому тест поступово уточнюється. Розробка тесту може здійснюватися на теоретичних або статистичних підставах. Виділяють зазвичай три використовувані загальні стратегії: індуктивна, дедуктивна та емпірична [15]. Створені сьогодні шкали часто включають елементи всіх трьох методів.

Тест особистості можна провести безпосередньо до особи, яку оцінюють, або спостерігача. У самозвіті людина реагує на особистісні пункти, оскільки вони стосуються самої людини. Самозвіти є розповсюдженим методом. У звіті спостерігача людина реагує на особистісні елементи, оскільки вони стосуються когось іншого. Щоб отримати найбільш точні результати, спостерігач повинен знати людину, яку оцінюють. Поєднання балів самозвіту та звіту спостерігача може зменшити кількість помилок, забезпечуючи більш точне зображення

особи, яку оцінюють. Самозвіти та звіти спостерігачів, як правило, дають подібні результати, підтверджуючи їх достовірність.

Багато психологічних та психоосвітніх тестів недоступні для громадськості. Видавці тестів встановлюють обмеження щодо того, хто має доступ до тесту. Рада з ліцензування психології також обмежують доступ до тестів, які використовуються в ліцензуванні психологів [16]. Видавці тестів вважають, що як авторські права, так і професійна етика вимагають від них захисту тестів. Видавці продають тести лише людям, які підтвердили свою освіту та професійну кваліфікацію. «Покупці» юридично зобов'язані не відкривати відповіді на тести або самі тести представникам громадськості, якщо це не дозволено видавцем [16].

Міжнародна тестова комісія (ІТС) – міжнародна асоціація національних психологічних товариств і видавців тестів – публікує Міжнародні рекомендації щодо використання тестів, в яких прописані заходи, які необхідно вжити для «захисту цілісності» тестів, шляхом публічної відсутності описання методів тестування та «тренування індивідів» так, щоб вони «не могли несправедливо впливати на результати тестів».

З початку ХХІ ст. у процесі тестування стали широко застосовуватися комп'ютери. Саме так виник окремий напрямок – комп'ютерне тестування, у якому пред'явлення тестів, оцінювання результатів опитуваного і видача результатів здійснюється з допомогою ПК[17].

Етап генерації тестів технологічно може протікати по-різному, зокрема шляхом введення в комп'ютер бланкових тестів.

Хоча комп'ютерне тестування значно полегшує роботу при пред'явленні та оцінюванні результатів виконання тестів, його поширення багато в чому не більше ніж данина моді, всі негативні наслідки які досі не виявлені повною мірою.Звертатися до комп'ютерного тестування слід у випадках, коли є нагальна потреба відмовитися від традиційних бланкових тестів (вимушене віддалене

знаходження опитуваного чи того, хто опитує, самоізоляція, психологічні розлади, що унеможливають процес очного спілкування тощо).

Комп'ютерне тестування має певні переваги в порівнянні з традиційним бланковим тестуванням, які виявляються особливо помітно при масових перевірках. Пред'явлення варіантів тесту на комп'ютері дозволяє заощадити кошти, що витрачаються зазвичай на друк та транспортування бланкових тестів. Також цей спосіб є більш екологічним рішенням.

Завдяки комп'ютерному тестуванню можна підвищити інформаційну безпеку та запобігти розсекреченню тестуза рахунок високої швидкості передачі інформації та спеціального захисту електронних файлів[17]. Спрощується також процедура підрахунку результуючих балів у тих випадках, коли тест містить лише завдання з вибором відповідей. Інші переваги комп'ютерного тестування виявляються у поточному контролі – завдяки комп'ютеру можна негайно видати тестовий бал і вжити невідкладних заходів, наприклад, щодо корекції засвоєння нового матеріалу на основі аналізу протоколів за результатами виконання коректуючих та діагностичних тестів[17].

Найрозповсюдженішим ресурсом для проведення не очної психологічної оцінки є веб-додаток.

1.2 Аналіз методів та підходів проведення опитувань та тестувань

Серед підходів до процесу проведення опитувань та тестувань (анкетувань) виділяють класичний (так званий традиційний) та адаптивний підходи.

Традиційний тест являє собою стандартизований метод оцінки рівня знань і структури підготовленості людини. При проведенні такого тестування всі відповідають на одні і ті ж завдання протягом однакового часу, в однакових умовах і з однаковими правилами оцінювання відповідей [3].

Готовий тест, стандартизований або призначений для поточного контролю, вводиться в спеціальну оболонку, функції якої можуть розрізнятися за ступенем повноти. Зазвичай при підсумковому тестуванні «оболонка» може подавати завдання, та давати змогу оцінювати результати їх виконання, формувати матрицю результатів тестування, обробляти її та шкалювати первинні бали досліджуваних шляхом перекладу в одну із стандартних шкал для видачі кожному з випробуваних тестових балів і протоколу його оцінок за завданням тесту. Під оболонкою мається на увазі паперовий бланк чи інтерфейс програмного додатку.

Адаптивне тестування – різновид тестування, при якому порядок представлення запитань (або складність) залежить від відповідей того, хто тестується, на попередні питання [18]. Процес можливий лише під час комп'ютерного тестування та може протікати двома шляхами.

Перший передбачає автоматизовану генерацію варіантів тесту, що здійснюється за допомогою інструментальних засобів. Варіанти створюються перед тестуванням або безпосередньо під час проведення з банку каліброваних тестових завдань зі стійкими статистичними характеристиками. Калібрування досягається завдяки тривалій попередній роботі(формування бланка), параметри завдань якого отримують на репрезентативній вибірці опитуваних, як правило, базуючись на результатах бланкових тестів. Змістовна валідність та паралельність варіантів забезпечуються за рахунок строго регламентованого відбору завдань кожного варіанту відповідно до специфікації тесту.

Другий шлях – комп'ютерне адаптивне тестування. У основі ідей адаптивності лежать міркування у тому, що тестованому марно давати завдання тесту, що він виконає напевно правильно без найменших труднощів, чи гарантовано не впорається з високої труднощі. Тому пропонується оптимізувати складність завдань, адаптуючи її до рівня підготовленості кожного випробуваного, та скоротити за рахунок виключення частини завдань довжину тесту[18].

Дані методи також можна комбінувати.

Можна виділити багато переваг адаптивного методу над традиційним:

- скорочення часу проведення опитування чи тестування;
- полегшення процесу проведення опитування чи тестування для опитуваного шляхом корекції наступного питання в залежності від відповіді;
- отримання біль точних результатів;
- полегшення процесу перевірки результатів опитування (як безпосередньо перевіряючим, так і через автоматизовану систему) завдяки запобіганню надлишкових відповідей.

1.3 Аналіз видів вебдодатків

Веб-додаток – це прикладне програмне забезпечення, яке працює на веб-сервері, на відміну від комп'ютерних програм, які запускаються локально в операційній системі (ОС) пристрою. Доступ до веб-додатків здійснюється користувачем через веб-браузер із активним мережевим підключенням [19].

Серед веб-додатків важливо виділити зокрема десктопний веб-застосунок, веб-застосунок з адаптивним дизайном та мобільний застосунок.

Десктопний веб-додаток, як можна зрозуміти з терміну, використовується лише з комп'ютера або ноутбука. Його користувацький дизайн не дозволяє використовувати додаток на інших видах пристроїв, таких як планшети або мобільні пристрої, що вимагає розробки окремих додатків під пристрої інших форматів. Адаптивний дизайн дозволяє уникнути цієї проблеми.

Адаптивний веб-дизайн – дизайн веб-сторінок, що забезпечує оптимальне відображення та взаємодію сайту з користувачем незалежно від роздільної здатності та формату пристрою, з якого здійснюється перегляд сторінки.

Метою адаптивного веб-дизайну є практичне відображення інформації на всіх пристроях (від стаціонарних ПК до мобільних телефонів). За технологією адаптивного веб-дизайну не потрібно створювати окремі версії веб-додатку [20]. Один сайт може працювати на всьому спектрі пристроїв (рисунок 1.1).

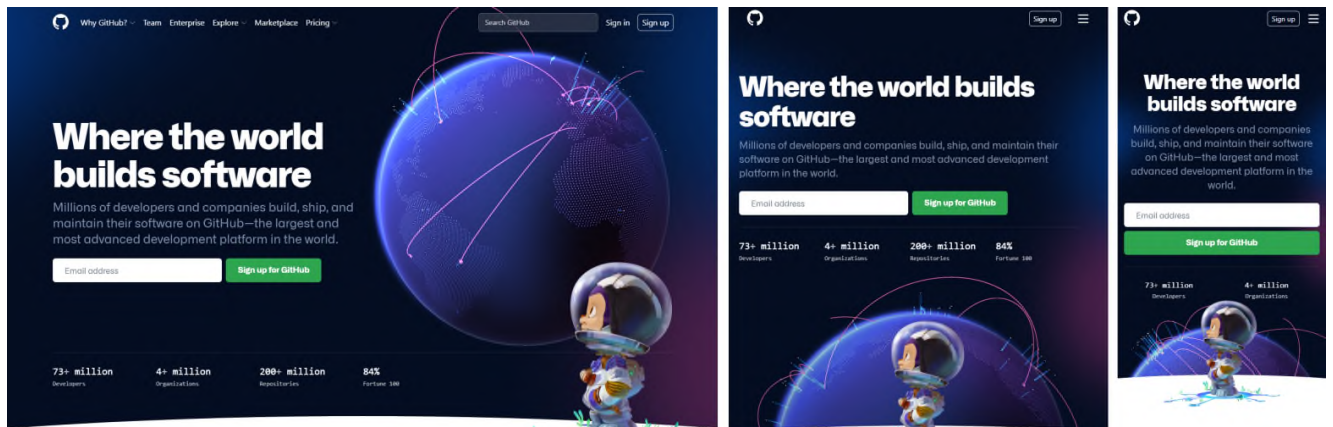


Рисунок 1.1 – Приклад адаптивного дизайну додатка.

Важливо зазначити, що це цілий підхід до розробки, а не лише розробка універсального дизайну, що буде зручним одночасно на всіх розширеннях. Розробнику варто пам'ятати про унікальний функціонал для підтримки того ж мобільного додатку, а також вимоги для ергономічного користувацького інтерфейсу мобільного додатку.

Мобільний додаток, також званий мобільним застосунком –це комп'ютерна програма або програмне забезпечення, призначене для роботи на мобільному пристрої, такому як телефон, планшет або годинник. Мобільні програми часто відрізняються від десктопних програм, які призначені для запуску на настільних комп'ютерах, і веб-програм, які працюють у мобільних веб-браузерах, а не безпосередньо на мобільному пристрої [21].

Спочатку програми були призначені для підвищення продуктивності. Серед них електронна пошта, календар і бази даних контактів, але суспільний попит на програми викликав швидке розширення в інших сферах, таких як

мобільні ігри, автоматизація виробництва, GPS і послуги на основі визначення місцезнаходження, відстеження замовлень і покупки квитків, тож тепер доступні мільйони програм. Для багатьох програм потрібен доступ до Інтернету. Додатки зазвичай завантажуються з магазинів додатків, таких як App Store, Google Play, Windows Phone Store та інших, які є типом платформ цифрового розповсюдження. App Store, Google Play є дистриб'юторами додатків для операційних систем iOS та Android відповідно.

iOS (раніше iPhone OS) – це мобільна операційна система, створена та розроблена Apple Inc. виключно для свого апаратного забезпечення. Саме операційна система працює на багатьох мобільних пристроях компанії, включаючи iPhone та iPod Touch; цей термін також включав версії, що працюють на iPad, доки ім'я iPadOS не було представлено з версією 13 у 2019 році. Це друга за поширеністю операційна система для мобільних пристроїв у світі після Android. Це основа для трьох інших операційних систем Apple: iPadOS, tvOS і watchOS. Це закрите програмне забезпечення, хоча деякі його частини є відкритим вихідним кодом за ліцензією Apple Public Source License та іншими ліцензіями [22].

Android – це мобільна операційна система, заснована на модифікованій версії ядра Linux та іншого програмного забезпечення з відкритим вихідним кодом, розроблена переважно для мобільних пристроїв із сенсорним екраном, таких як смартфони та планшети. Android розробляється консорціумом розробників, відомим як Open Handset Alliance і комерційно спонсорується Google. Він був представлений у листопаді 2007 року, а перший комерційний пристрій Android, HTC Dream, був запущений у вересні 2008 року [23].

Це безкоштовне програмне забезпечення з відкритим кодом; його вихідний код відомий як Android Open Source Project (AOSP), який переважно ліцензується за ліцензією Apache. Однак більшість пристроїв Android постачається з попередньо встановленим додатковим власним програмним забезпеченням, зокрема Google Mobile Services (GMS), який включає основні

програми, такі як Google Chrome, цифрова платформа розповсюдження Google Play і пов'язана платформа розробки служб Google Play [24].

1.4 Аналіз існуючого програмного забезпечення предметної області

За результатами аналізу програмного забезпечення для проведення опитувань та тестувань у відкритому доступі, можна виділити наступні веб-додатки: Google forms, Survio, Survey Monkey та SurveyMonkey.

1.4.1 Google forms

Google Forms – це програмне забезпечення для адміністрування опитування, яке входить до складу безкоштовного веб-редактора документів Google, який пропонує Google (рисунок 1.2). Служба також включає Google Docs, Google Sheets, Google Slides, Google Drawings, Google Sites і Google Keep. Google Forms доступний лише як веб-додаток. Додаток дозволяє користувачам створювати та редагувати опитування онлайн, співпрацюючи з іншими користувачами в режимі реального часу. Зібрана інформація може бути автоматично введена в електронну таблицю [25].

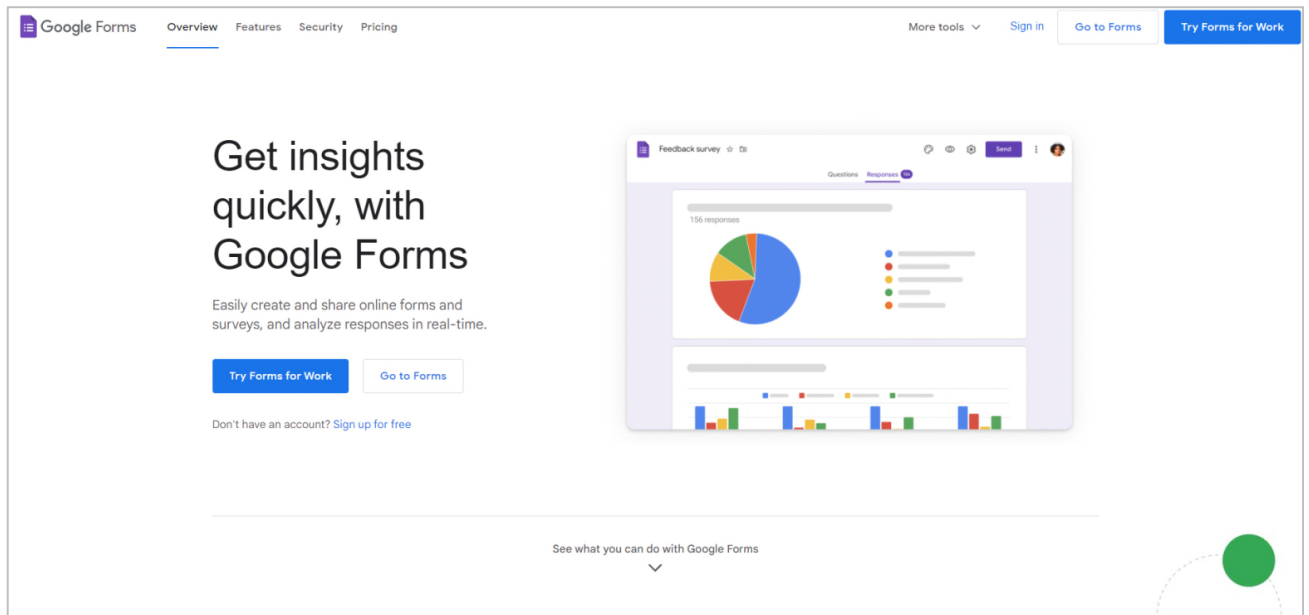


Рисунок 1.2 – Головна сторінка Google forms

Протягом багатьох років служба Google Forms зазнала кількох оновлень. Функції включають, але не обмежуються цим, пошук у меню, перемішування запитань у випадковому порядку, обмеження відповідей на одну особу, коротші URL-адреси, користувацькі теми, автоматичне створення пропозицій відповідей під час створення форм та «Завантажити файл» для користувачів, які відповідають на запитання, які вимагають від них ділитися вмістом або файлами зі свого комп'ютера або Диска Google [26]. «Інтелектуальна перевірка відповіді» здатна виявляти введення тексту в полях форми, щоб визначити, що написано, і попросити користувача виправити інформацію, якщо вона введена неправильно. Залежно від налаштувань спільного доступу до файлів на Google Диску користувачі можуть запитувати завантаження файлів від окремих осіб за межами відповідей із кількома параметрами в таблиці. У налаштуваннях користувачі можуть вносити зміни, які впливають на всі нові форми, наприклад, завжди збирати електронні адреси.

У Google Forms є всі функції спільної роботи та обміну, які є в Документах, Таблицях, Презентаціях, Малюнках і Сайтах [27].

Серед переваг Google forms варто виділити легкість створення та налаштування форми опитування – завдяки зручному інтерфейсу навіть непідготовлений користувач з легкістю зможе створити нову форму. Додаток не потребує додаткових реєстрацій, якщо користувач вже має Google профіль. Також важливо відмітити широкий спектр додаткових функцій налаштувань самого питання (обробка завантаження файлу, попередня перевірка відповіді користувача, використання інших функцій сервісів Google).

Класичний підхід до тестування, що призводить до виникнення плутанини у опитуваного та надлишкових даних для перевіряючого, та невелика кількість типів запитань відносяться до недоліків додатку.

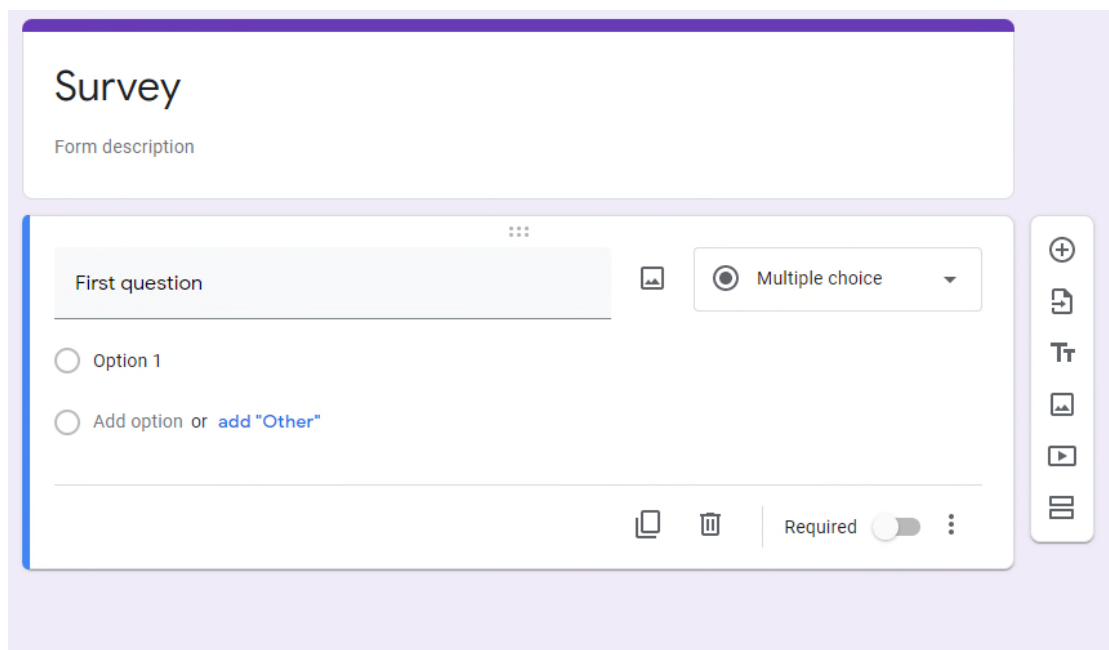


Рисунок 1.3 – Форма створення нового опитування в Google forms

1.4.2 Survio

Survio – це онлайн-система опитувань для підготовки анкет, збору та аналізу даних та обміну результатами [28].

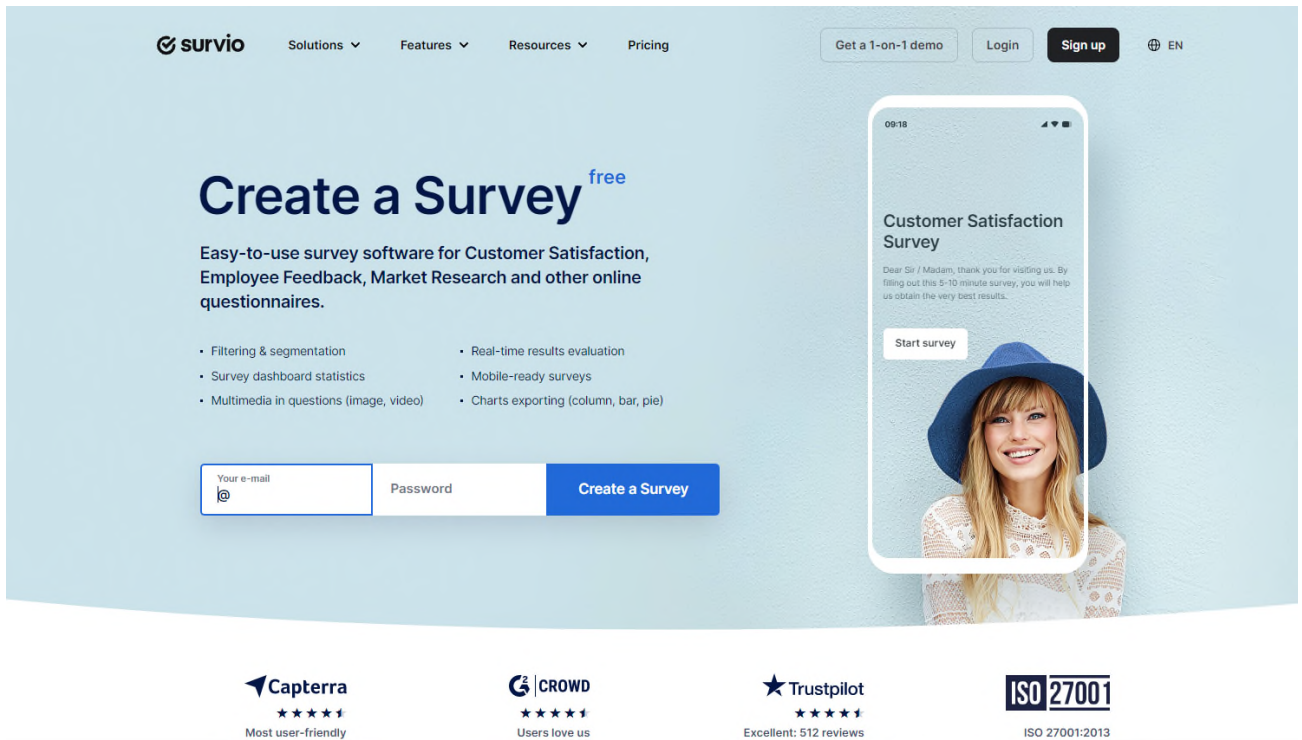


Рисунок 1.4 – Головна сторінка Survio

Survio розроблено спеціально для малих і середніх підприємств (МСП). Послуга також підходить для всіх, кому потрібно самостійно підготувати анкети: офіси, школи, студенти, некомерційні організації тощо. Проте все частіше Survio обирають великі міжнародні корпорації. Дуже часто Survio використовується як рішення для різноманітних опитувань серед кінцевих клієнтів, включаючи вимірювання задоволеності клієнтів. Також часто застосовуються опитування персоналу та робочого середовища [28].

Серед плюсів варто відзначити приємний користувацький дизайн та легкість створення простої форми опитування. Проте невелика кількість типів запитань, надлишковість інтерфейсу користувача та використання класичного підходу для проведення тесту є відчутними недоліками додатку.

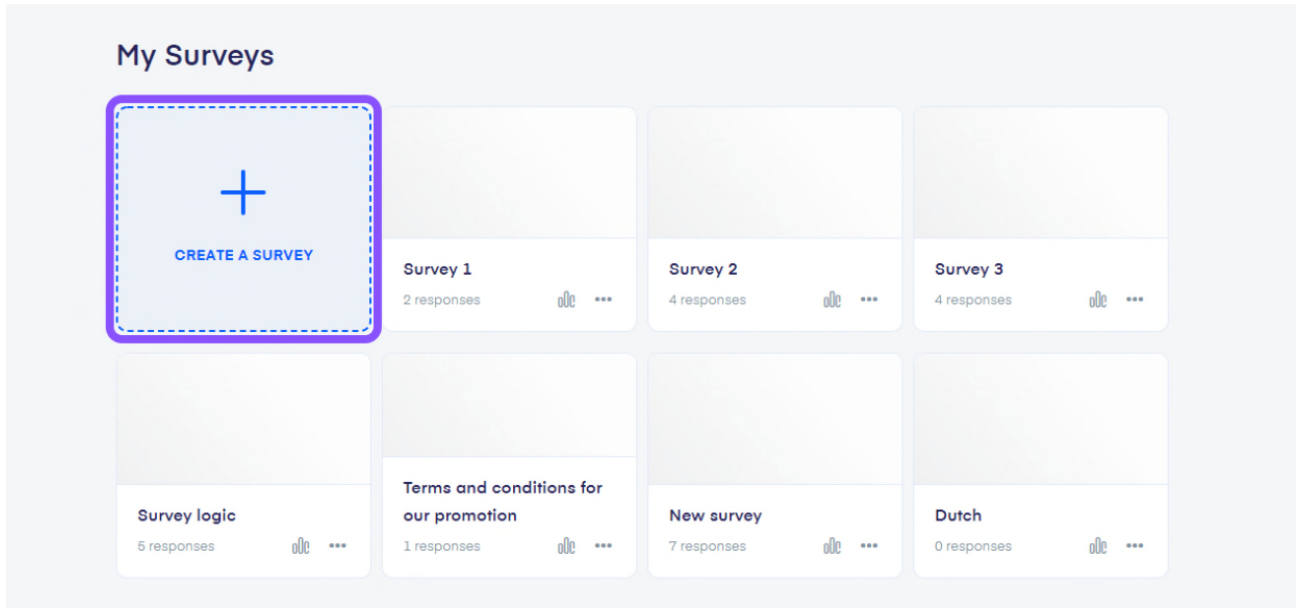


Рисунок 1.5 – Сторінка створення та перегляду існуючих форм опитувань

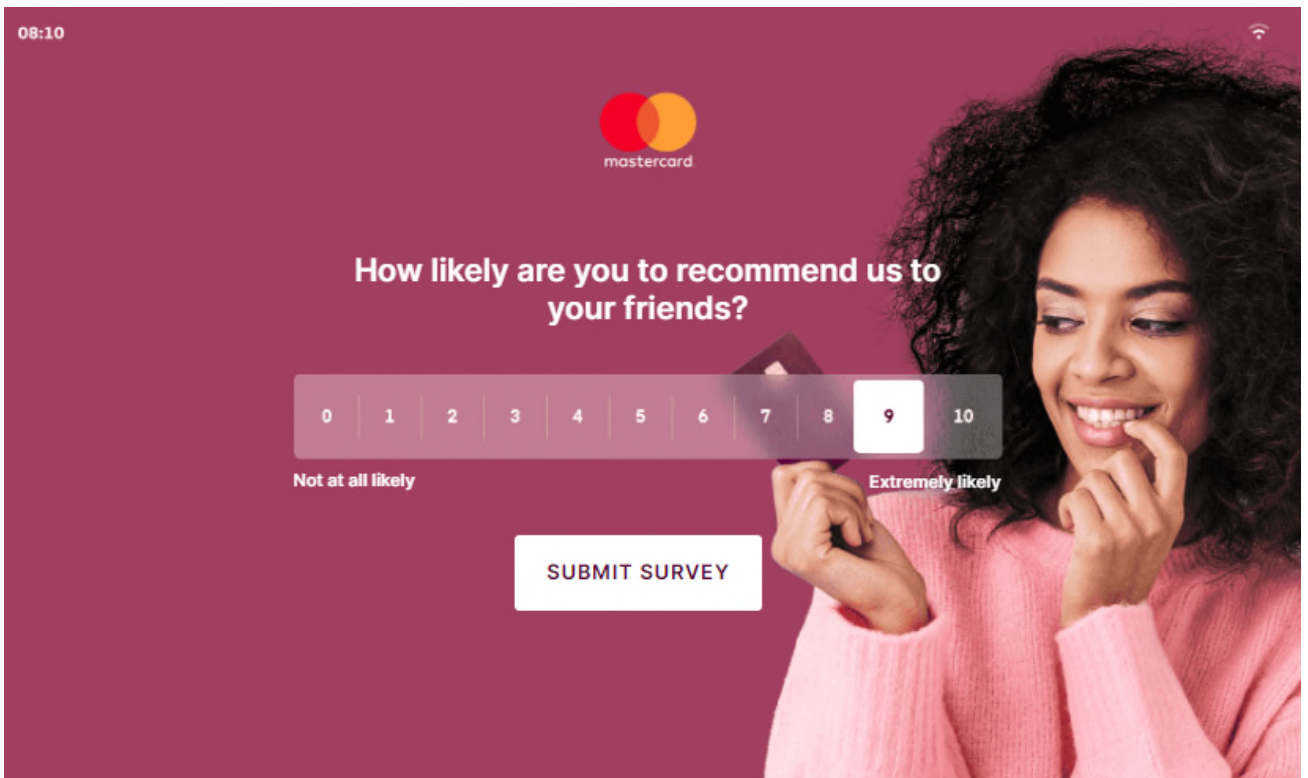


Рисунок 1.5–Один із варіантів питання Survio

1.4.3 SurveyMonkey

SurveyMonkey – онлайн-середовище для створення форм опитування [29]. Широко використовується підприємствами для внутрішніх опитувань та збору зовнішньої інформації.

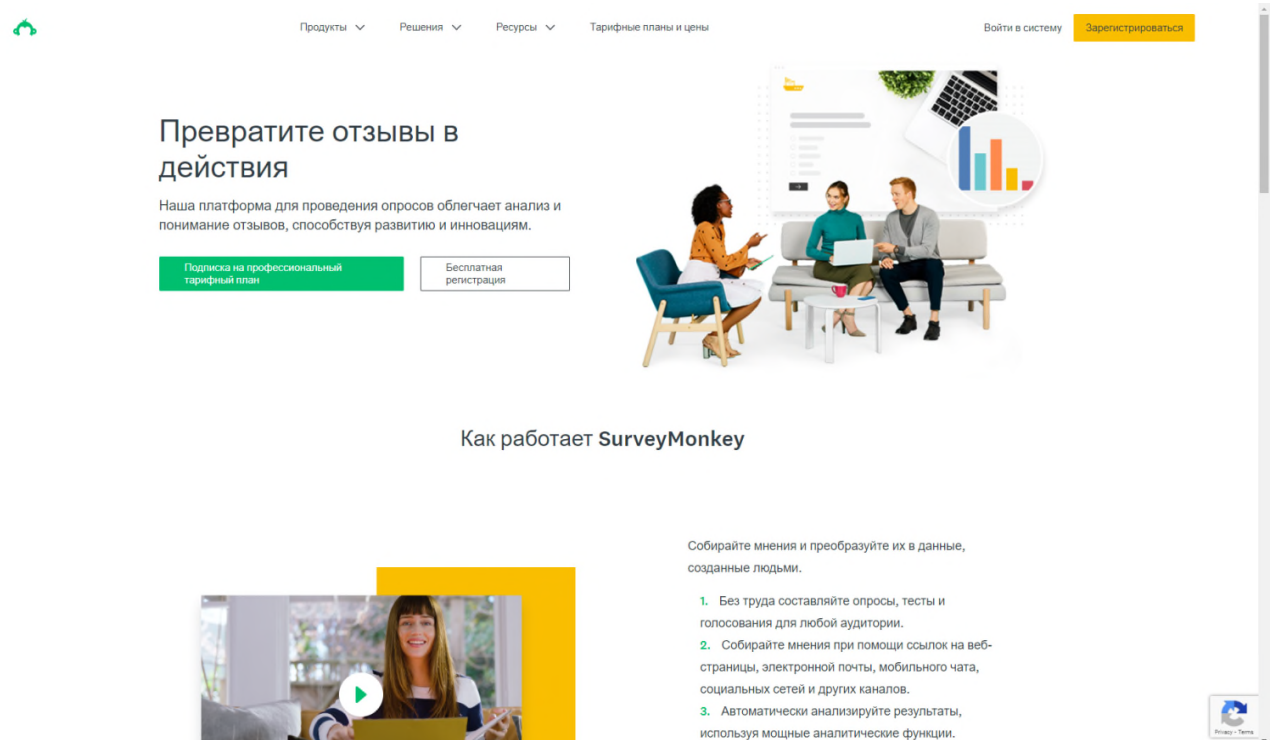


Рисунок 1.6 – Головна сторінка веб-ресурсу SurveyMonkey

Серед переваг ресурсу важливо виділити шаблонний (рис. 1.7) підхід до створення питань, що значно заощаджує час створення форми загалом. Також є підтримка адаптивного тестування за допомогою критерію складності тесту.

Недоліками є недостатня кількість функціоналу для створення та налаштування форми, а також обов'язкове внесення коштів.

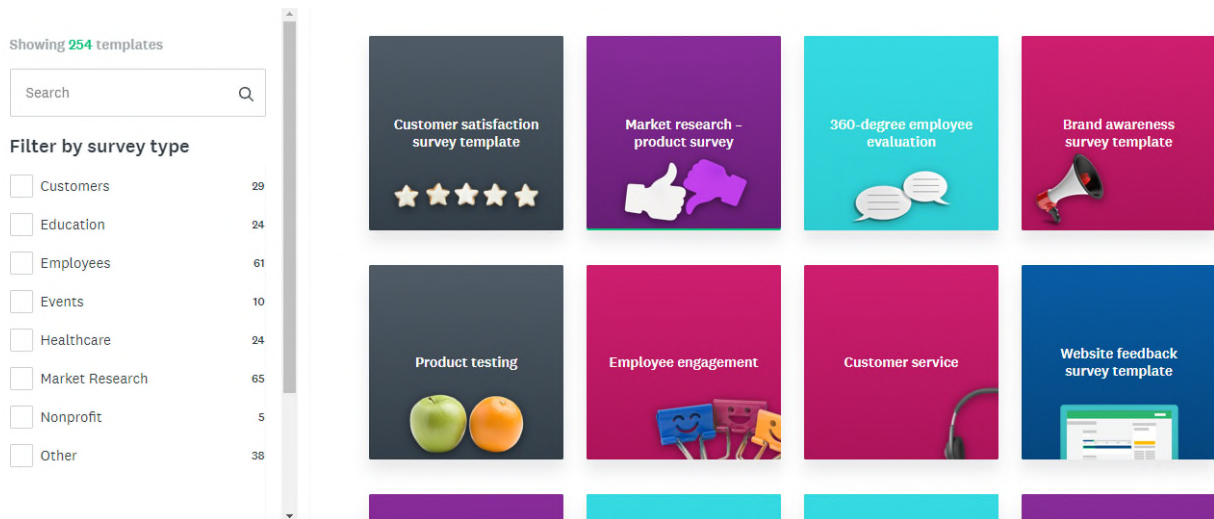


Рисунок 1.7 – Шаблони типів питань в SurveyMonkey

1.5 Постановка задачі

Метою магістерської роботи є розробка вдосконаленої інформаційної системи для проведення опитувань та тестувань. Для досягнення мети було визначено наступні задачі:

- дослідити існуючі методи та підходи для проведення опитувань та тестувань;
- вдосконалити процес проведення опитувань та тестувань шляхом динамічної зміни наступного питання та їх загальної кількості;
- спроектувати інформаційну систему для проведення опитувань та тестувань з використанням адаптивного підходу;
- забезпечити отримання результатів опитування чи тестування у формі, придатної для подальшого проведення аналізу.
- здійснити порівняння підходів при проведенні класичного та адаптивного тестування.

Висновки до розділу 1

В першому розділі розглянуто проблему традиційного проведення опитування, проведено аналіз видів вебдодатків. Як результат, було проведено дослідження існуючих інформаційних систем проведення опитувань та тестувань, та сформовано перелік їх переваг та недоліків. Було розглянуто основні методи проведення тестувань – класичний та адаптивний.

В результаті було сформовано мету та перелік задач для її досягнення.

Отже, вирішено спроектувати та розробити вдосконалену інформаційну систему для проведення опитувань та тестувань.

Розділ 2

Розробка інформаційної системи для проведення опитувань та тестувань

2.1 Загальний опис інформаційної системи проведення опитувань та тестувань

У результаті проведення аналізу існуючих додатків для опитувань та тестувань, розглянутих у розділі 1.4, важливо відмітити, що більшість з них приділяють велику увагу користувацькому інтерфейсу. Проте через використання негнучких технологій з'являється проблема великої кількості витраченого часу як для проведення тестування, так і для його перевірки. На формах з невеликою кількістю запитань це не так помітно. Проте зі збільшенням кількості питань час, затрачений на проходження опитування та його перевірку зростає експоненціально. Також на час перевірки відповідей впливатиме й збільшення кількості опитуваних.

Для оптимального проходження опитування час, затрачений на нього, не повинен перевищувати кількості запитань (якщо відсутні питання з розгорнутою відповіддю) [30]. Навіть у оптимізованому опитуванні іноді виникають питання, які можуть бути виключені відповіддю на будь-яке інше запитання. Як у автоматизованій системі опитувань, так і в звичайній, на такі питання все одно буде витрачений час і зі сторони опитуваного, і з сторони перевіряючого.

Тому необхідно розробити вдосконалену інформаційну систему для вирішення цього недоліку.

На рисунку 2.1 запропонована схема роботи інформаційної системи

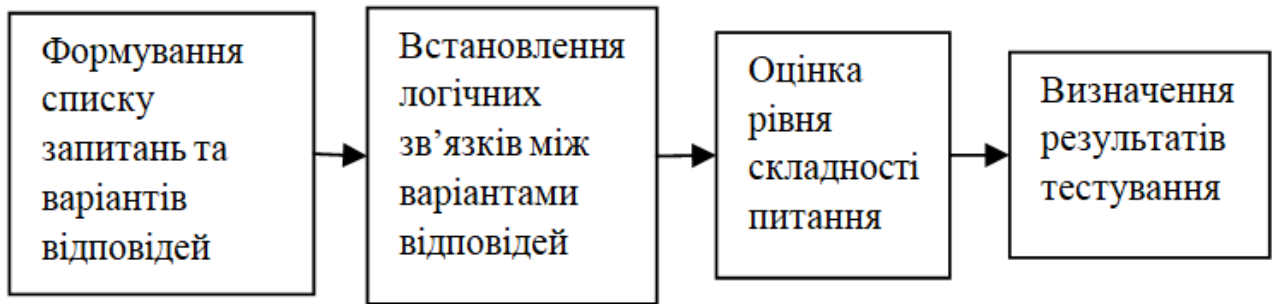


Рисунок 2.1 – Принцип роботи інформаційної системи

Для отримання результату спочатку необхідно сформувати список запитань з масивом відповідей на них. Важливим пунктом буде вибір користувачем типу відповіді запитання (далі – тип запитання). Системи, розглянуті у розділі 1.4 мали або мінімум типів запитань, або взагалі їх не мали. Тому пропонується наступний перелік типів запитань:

- welcome screen – не потребує відповіді;
- long text – потребує розгорнуту відповідь;
- short text – потребує відповідь у декілька слів;
- choice – надає вибір між декількома опціями;
- picture choice – виконує ту ж функцію, що і choice, проте дає змогу відобразити варіанти відповіді через зображення;
- yes-no – альтернативний тип відповіді;
- opinion scale – потребує вибір відповіді відносно певної шкали;
- rating – потребує оцінку користувача;
- statement – не потребує відповіді;
- video – потребує відповідь користувача у форматі відео.

Кожне із питань може бути як обов'язковим, так і добровільним.

Важливою властивістю інформаційної системи є можливість відео опитування. Враховуючи що дана система рекомендується до використання у сфері психології, даним чином перевіряючий зможе отримати більше інформації

з відповіді опитуваного, адже міміка та жести є іноді важливими складовими відповіді.

Далі встановлюються логічні зв'язки між варіантами відповідей та індексами наступних питань. Згідно з даними відповідями на місце наступного питання буде обрано більш релевантне в залежності від заданих умов. Наступний крок – оцінка рівня складності питання, що досягається шляхом надання певної кількості балів за кожен варіант відповіді на питання. Як результат отримується вибірка пар «питання-відповідь» та інформація про набрані бали, що можуть використовуватись для загальної оцінки опитуваних, аналізу релевантності питань до обраної теми, а також формування статистики [31].

Даний різновид тестування дозволяє підлаштовуватись під індивідуальні можливості кожного, хто проходить тестування, завдяки виключенню занадто важких, або занадто легких завдань, що одночасно впливає на час проведення тестування. Таким чином, при кожній вірній відповіді наступне питання буде більш складним, а при неправильних відповідях складність тесту буде зменшуватись. Також на основі відповідей можна визначити наступне питання із тієї теми, що менш відома тому, хто проходить тест. Це дозволить зменшити кількість помилок тестування, що іноді виникають при неправильному трактуванні теми опитуваним [31].

Запропонована система із використанням більш гнучких методів проведення тестування дозволить скоротити час, затрачений на проходження та перевірку опитування чи тестування.

2.2 Методи проведення тестувань

Проблема надлишкового затраченого часу у проведенні опитування чи тестування виникає не вперше, проте у сучасному світі час стає все ціннішим

ресурсом. У багатьох публікаціях та тезах розглядається це питання та пропонуються варіанти рішення цієї проблеми.

Для дослідження було обрано наступні підходи для проведення опитувань та тестувань: традиційний, так званий класичний, та адаптивний.

2.2.1 Традиційний та класичний метод проведення опитувань та тестувань

Традиційний тест являє собою стандартизований метод оцінки рівня знань і структури підготовленості людини. При проведенні такого тестування всі відповідають на одні і ті ж завдання протягом однакового часу, в однакових умовах і з однаковими правилами оцінювання відповідей. Одне з головних питань теорії тестів – питання підбору оптимального за деякими критеріями тесту [32]. Кожен тест може мати відмінності від інших тестів за безліччю параметрів, у тому числі кількістю питань та їх складністю.

Практика показує, що оптимальним рішенням є відносно невеликий тест, проте добре відображає поставлену задачу. Поняття "довжина тесту" введено на початку ХХ століття Ч. Спірманом і означає кількість завдань у тесті. Чим довший тест, тим більше в ньому завдань. Від кількості завдань деяким чином залежить точність виміру знань [32]. На рисунку 2.2 представлена схема роботи традиційного методу проведення опитування чи тестування.



Рисунок 2.2 – Схема роботи традиційного методу проведення тестування

Перевага цього методу очевидна – простота організації як звичайного, так і автоматизованого проходження опитування чи тестування.

2.2.2 Розширення традиційного методу проведення опитувань та тестувань – адаптивний метод

Враховуючи недоліки традиційного методу проведення опитувань та тестувань, протягом часу метод зазнав деяких вдосконалень. Так з'являється адаптивне тестування.

Адаптивне тестування – різновид тестування, при якому порядок представлення запитань (або складність) залежить від відповідей того, хто тестується, на попередні питання [33].

Адаптивне тестування можна організувати по-різному. Один із варіантів - розділити питання тесту за складністю. Якщо педагогічний тест визначити коротко як систему завдань зростаючої складності, то стане зрозуміло, що складність завдань є найважливішим тестоутворюючим показником [34]. Показник складності тесту і тестових завдань є змістовним і формальним одночасно.

Змістовним – тому що на складність впливають як рівень знань тих, що тестують, ефективна організація проведення тесту, так і сам зміст тесту. Формальним – тому що виникає процес протистояння завданням тесту (у цьому випадку існує лише два типи результату – перемога при правильній відповіді, чи поразка при неправильній). Оцінка результату такої боротьби залежить від співвідношення рівня знання до рівня складності завдання [34].

На рисунку 2.3 представлена схема роботи адаптивного методу проведення опитування чи тестування.

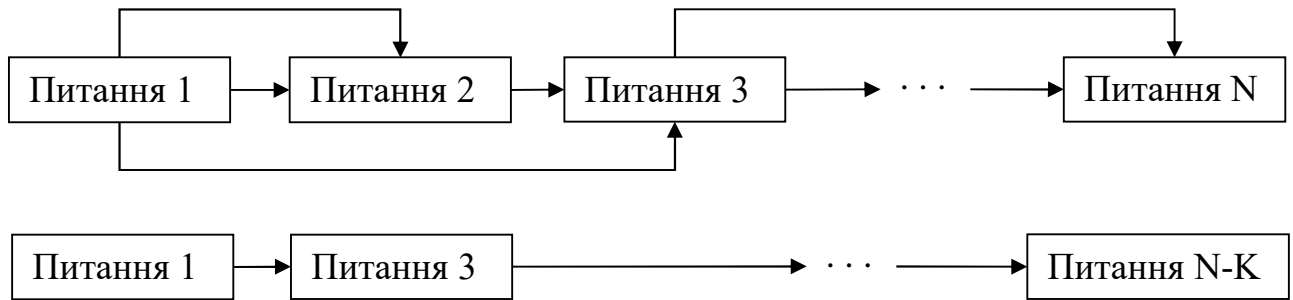


Рисунок 2.3 – Схема роботи адаптивного методу проведення тестування

Де N – загальна кількість питань, K – кількість вилучених питань за відсутності їх релевантності відносно даних відповідей.

Перевагами цього методу є зменшення надлишкових даних, що дає:

- зменшення часу, витраченого на проведення опитування чи тестування;
- зменшення часу, витраченого на перевірку відповідей опитування чи тестування;
- підвищена точність результатів опитування чи тестування;

Висновки до розділу 2

В другому розділі описано інформаційну систему для проведення опитувань та тестувань. Розглянуто основні методи проведення опитувань та запропоновано до аналізу два основних – класичний та адаптивний метод, як вдосконалення класичного методу.

Проаналізовано ефективність двох методів та отримано результати.

Із результатів аналізу слідує, що адаптивний підхід є більш оптимізованим, тому пропонується використовувати саме його.

Розділ 3

Проектування інформаційної системи проведення опитувань та тестувань

3.1 Основні вимоги для проектування інформаційної технології

Проектування інформаційної системи починається із формування її мети та задач задля її досягнення.

Метою роботи є реалізація інформаційної системи для проведення опитувань та тестувань з використанням адаптивного підходу.

Для досягнення цього необхідно:

- відповідну до вимог функціональну модель;
- компоненти створення та керування формою опитування;
- компоненти конфігурування питань опитування;
- компоненти опрацювання відповідей на опитування.

Необхідно сформувати блоки моделі та зв'язки між ними.

Розроблена інформаційна система повинна забезпечити функціональні вимоги для проходження опитування чи тестування, а також опрацювання відповідей на них.

3.2 Проектування загальної схеми інформаційної системи

Для виконання поставлених функціональних вимог було сформовано функціональну модель, що складається із трьох модулів (рисунок 3.1):



Рисунок 3.1 – Схема взаємодії модулів інформаційної системи

За отримання вхідних даних від користувача, їх перегляд, редагування та видалення відповідає *модуль інтерфейсу користувача*. Він надає можливість зручно користуватись інформаційною системою, а також дає змогу об'єднати функціонал інших модулів.

Модуль обробки запитань відповідає за створення форми, формування банку запитань із вхідних даних, заданих користувачем, аналізу відповідей та обрахунку оцінки. Це основний модуль, що містить найважливіші обрахунки інформаційної системи, а саме реалізацію адаптивного підходу у тестуванні та реалізацію алгоритму обрахунку складності запитань.

Модуль обробки відео-запитань – це відгалуження від основного модуля. За використання окремої API відповідає за збір відеоінформації та її відтворення.

Дана схема описує основні модулі та процес їх взаємодії між собою. Описані модулі дозволять реалізувати інформаційну систему для проходження опитувань та тестувань з використанням адаптивного підходу.

3.3 Використання адаптивного підходу для підбору наступного питання та визначення складності тесту

Однією із найважливіших частин інформаційної системи є аналіз відповідей та обрахунок оцінки. Одночасно ці параметри є головними критеріями обрахування складності питання та тесту загалом.

Оцінка кожного наступного питання безпосередньо впливає на складність самого тесту. Адаптивне тестування передбачає відбір завдань безпосередньо з банку за певними алгоритмами, які прогнозують оптимальну складність наступного завдання за результатами виконання опитуваним попереднього завдання адаптивного тесту. Відмінною особливістю адаптивного тестування є покрокова переоцінка рівня підготовленості опитуваного, що робиться після кожного виконання чергового завдання опитування чи тесту (рисунок 3.2).

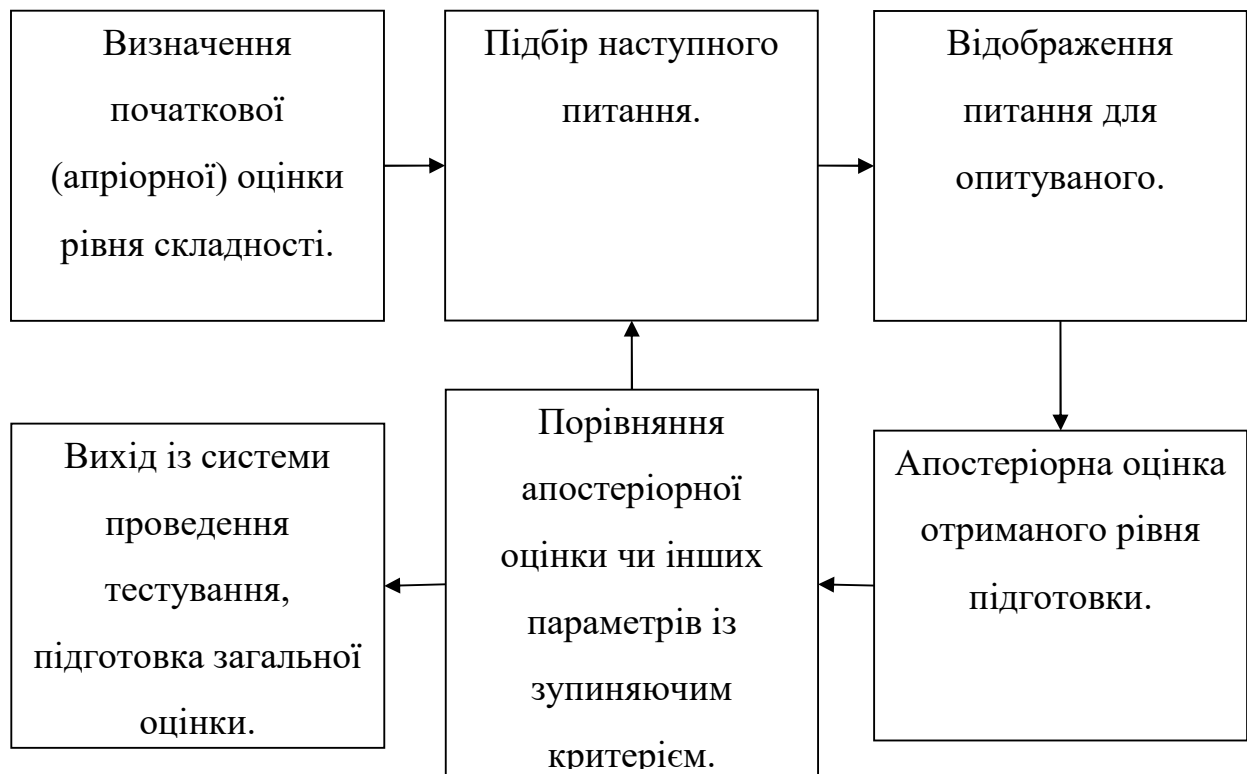


Рисунок 3.2 – Алгоритм багатокрокового адаптивного тестування

Детальне відображення алгоритму представлено на рисунку 3.3:

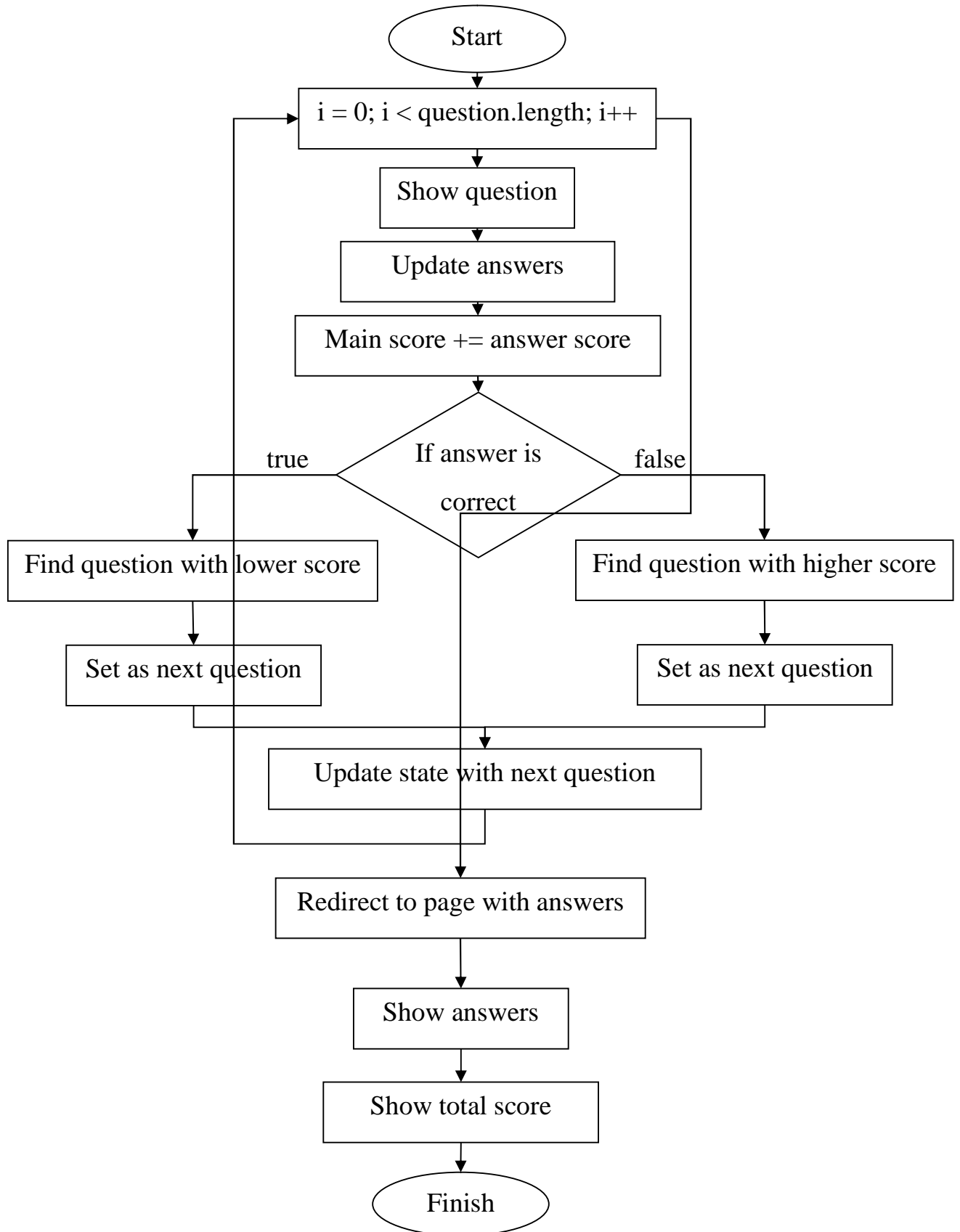


Рисунок 3.3 – Блок-схема алгоритму обрахування наступного питання та загального балу

3.4 Проектування структури бази даних

Для забезпечення функціональних потреб інформаційної системи необхідно сформувати структуру сховища даних, яка ляже в основу БД додатку. На рисунку 3.4 зображена спроектована структура БД, що складається з наступних таблиць: «forms», «questions», «answers», «answerScoreConditions» та «media».

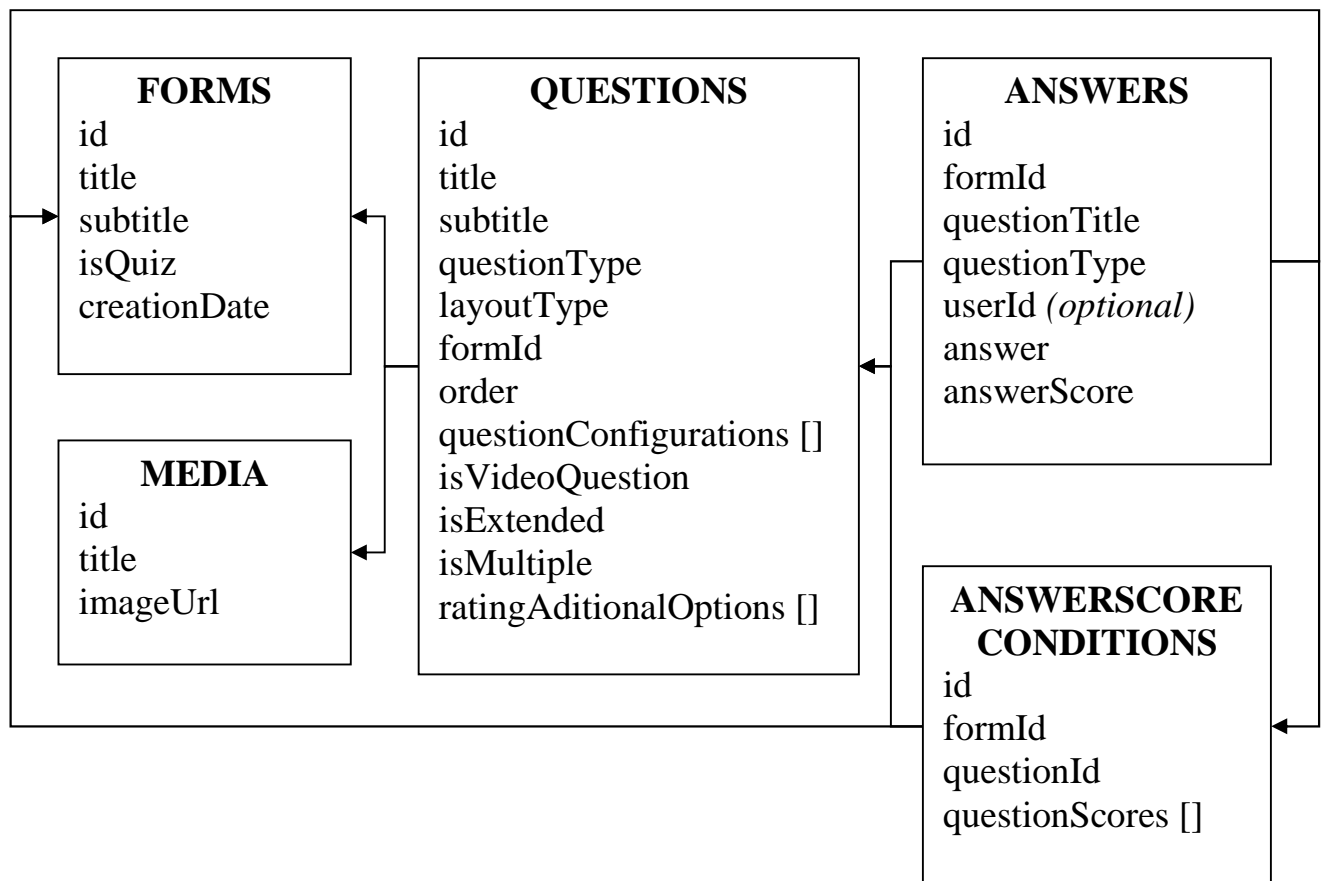


Рисунок 3.4 – Структура бази даних

У таблиці «forms» міститься уся інформація про форму для проходження опитування чи тестування.

Таблиця 3.1 – Атрибути таблиці «forms»

Таблиця «forms»		
id	String	Унікальний ідентифікатор форми для опитування
title	String	Назва форми
subtitle	String	Додатковий опис форми (необов'язкове поле)
isQuiz	Boolean	Поле, що зберігає інформацію, чиєформа тестом, чи опитуванням
creationDate	Timestamp	Дата та час створення форми

Таблиця «questions» зберігатиме масив питань форм, його властивості та систему зв'язків для проведення адаптивного тестування, а також всю інформацію, необхідну для коректної підтримки користувацького інтерфейсу стосовно конфігурації питань.

Таблиця 3.2 – Атрибути таблиці «questions»

Таблиця «questions»		
id	String	Унікальний ідентифікатор питання
title	String	Текстзапитання
subtitle	String	Додатковий опис питання (необов'язкове поле)
order	Number	Порядковий номер питання у формі, додаткове поле для організації обрахунку наступного питання
questionType	String	Тип відповіді на питання

layoutType	String	Тип відображення питання
formId	String	Вторинний ключ для однозначної ідентифікації відповідного запису у таблиці «forms»
questionConfigurations	Array	Дані для конфігурації логіки обрахунку наступного питання, а також збору відповідей та балів
ratingAdditionalOptions	Array	Додаткове поле для конфігурації розширеної форми питання
isVideoQuestion	Boolean	Поле з інформацією, чи є форма запитання – відео
isExtended	Boolean	Поле з інформацією, чи є питання розширеним
isMultiple	Boolean	Конфігураційне поле для розширеного питання

Таблиця 3.3 – Субколекції таблиці «questions».

questionConfigurations	answerOption
	answerOptionId
	redirectConditionRule
	redirectQuestion
ratingAdditionalOptions	answerOption
	answerOptionId

Таблиця «answers» зберігатиме інформацію про відповіді після проходження опитування. Якщо батьківський додаток не надасть інформацію про поточного користувача, буде використовуватись користувач по замовченню.

Таблиця 3.4 – Атрибути таблиці «answers»

Таблиця «answers»		
id	String	Унікальний ідентифікатор відповіді
answer	String	Текст відповіді на запитання
answerScore	Number	Оцінка, нарахована за відповідь
formId	String	Вторинний ключ для однозначної ідентифікації відповідного запису у таблиці «forms»
userId	String	Вторинний ключ для однозначної ідентифікації відповідного запису у таблиці «users» або данні, що прийшли від батьківського додатку
questionTitle	String	Текст запитання
questionType	String	Тип запитання
Date	Timestamp	Дата створення запису

Таблиця «media» містить масив посилань на медіа файли у сховищі. Дані медіа дані використовуються лише для конфігурації питань.

Таблиця 3.5 – Атрибути таблиці «media»

Таблиця «media»		
id	String	Унікальний ідентифікатор медіа-файлу
title	String	Назва медіа-файлу
imageUrl	String	Шлях до медіа-файлу у сховищі

Таблиця «answerScoreConditions» містить у собі конфігураційні дані питання що стосуються оцінки складності питання. Дана таблиця також містить субколекцію:

Таблиця 3.6 – Атрибути таблиці «answerScoreConditions»

Таблиця «answerScoreConditions»		
id	String	Унікальний ідентифікатор оцінки
formId	String	Вторинний ключ для однозначної ідентифікації відповідного запису у таблиці «forms»
questionId	String	Вторинний ключ для однозначної ідентифікації відповідного запису у таблиці «questions»
questionScores	Array	Дані про питання, за яке були начислені бали

Таблиця 3.7 – Субколекції таблиці «answerScoreConditions».

questionScores	answerOption
	answerOptionId

3.5 Проектування інтерфейсу інформаційної системи

Веб-додаток має єдиний інтерфейс завдяки відсутності власної системи авторизації. Загальний користувацький інтерфейс додатку не має мати вираженого шаблону самостійного додатку (як до прикладу, додатки, розглянуті у розділі 1.3), проте має задовольняти вимоги ситуацій, що передбачають відсутній інтерфейс батьківського додатку, та мати інтерфейс зрозумілий для користувача.

Враховуючи вищеперераховані вимоги, було вирішено відмовитись від футера, який зазвичай містить інформацію про систему. Звичайна навігація на стартовій сторінці також не потрібна, тому вона повинна бути інтегрована в блок контенту системи.

На рисунку 3.5 зображено типовий шаблон інформаційної системи, адаптований до потреб додатку, що буде розроблятися.

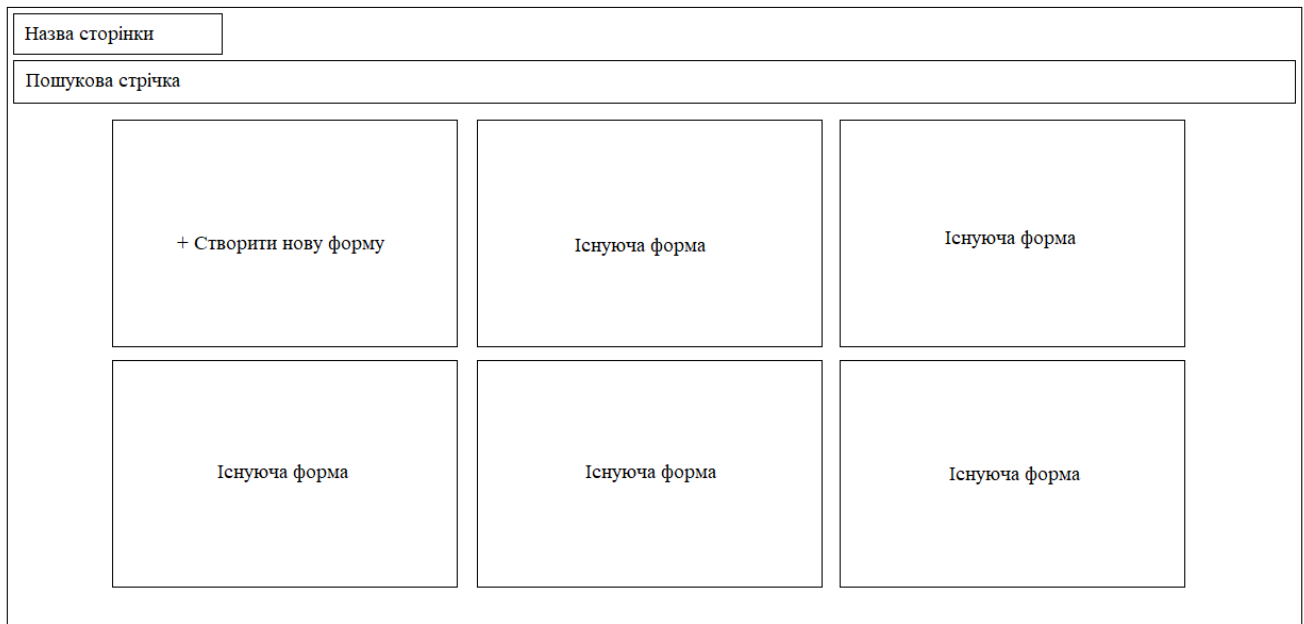


Рисунок 3.5 – Адаптований шаблон інтерфейсу інформаційної системи

Для реалізації зручного користувацького інтерфейсу, що при цьому виконує поставлені вимоги, необхідно дати можливість користувачу, особливо, якщо він користується додатком вперше, створювати нові форми опитування, переглядати, редагувати та видаляти уже існуючі та зручно переходити до наступних сторінок. Розширений елементом додавання нових форм список уже існуючих задовольнить цю вимогу. Також необхідно організувати систему пошуку необхідної форми по назві для підтримки великої кількості елементів. На приладах з іншим розширенням схема майже не зміниться завдяки адаптивному підходу до розробки.

На рисунку 3.6 зображена спроектована схема інтерфейсу користувача детального перегляду форми. Через велику кількість елементів дана сторінка на

мобільних пристроях буде недоступна – замість неї відобразиться напис з проханням змінити тип пристрою.



Рисунок 3.6 – Інтерфейс користувача: сторінка конфігурації питань обраної форми

Інтерфейс перегляду форми не відрізняється від блоку контенту сторінки конфігурації питань обраної форми. Для перегляду відповідей буде використовуватись таблиця без додаткових налаштувань.

Розроблені схеми будуть використовуватись при розробці інформаційної системи як ескіз інтерфейсу користувача.

Висновки до розділу 3

В третьому розділі розглянуто основні засади проектування інформаційної системи для проведення опитувань та тестувань, а також спроектовано загальну схему інформаційної системи.

Запропоновано функціональну модель інформаційної системи, що буде забезпечувати виконання вимог, які необхідно враховувати при проектуванні системи.

Розглянуто типові шаблони інтерфейсу користувача, та сформовані власні, що будуть задовольняти вимоги інформаційної системи. Сформовано структуру бази даних для інформаційної системи.

Розділ 4

Програмна реалізація

4.1 Обґрунтування вибору мови програмування для розробки інформаційної системи

На основі результатів, отриманих у розділі 1.3, було отримано висновок, що для проведення опитувань та тестувань потрібно охопити якомога більше платформ. Тому розгортання додатку на десктопі, IOS та Android пристроях є доцільним.

Для досягнення цієї цілі було обрано адаптивний веб-додаток як реалізацію інформаційної системи. Завдяки даному підходу в розробці дизайн буде підлаштовуватись під будь які пристрої із виходом у мережу Інтернет. Це має наступні переваги:

- значно зменшиться час на реалізацію додатку;
- буде спрощено процес підтримки додатку у майбутньому.

Для проведення дослідження мов програмування було використано інформацію із сформованого рейтингу на «dou.ua» (рисунок 4.1) [35]:

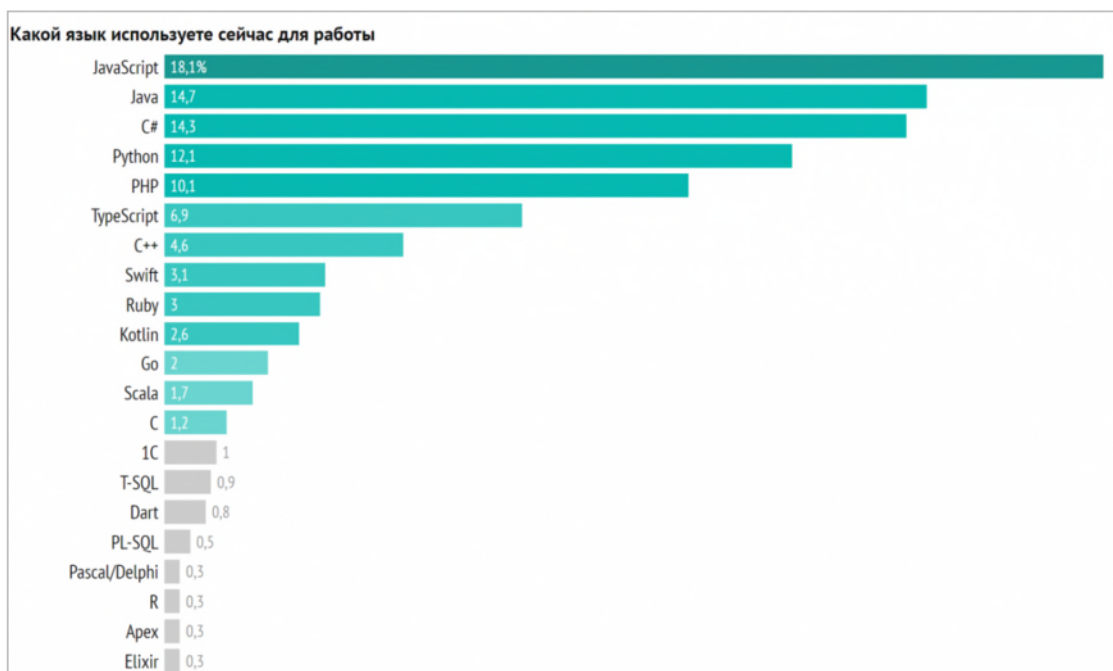


Рисунок 4.1 – Діаграма дослідження рейтингу мов програмування

Слідуючи даним таблиці, розглянемо лідерів рейтингу: JavaScript, Java та C#.

JavaScript (JS) – це легка, інтерпретована або «вчасно скомпільована» мова програмування з першокласними функціями. Хоча він найбільш відомий як мова сценаріїв для веб-сторінок, багато середовищ, які не є браузерами, також використовують його, наприклад Node.js, Apache CouchDB та Adobe Acrobat. JavaScript – це заснована на прототипах, багатопарадигмальна, однопотокова динамічна мова, яка підтримує об'єктно-орієнтований, імперативний та декларативний (наприклад, функціональне програмування) стилі [36].

Стандартом мови JavaScript є ECMAScript. Станом на 2012 рік, всі сучасні браузери повністю підтримують ECMAScript 5.1. Старі версії браузерів підтримують принаймні ECMAScript 3. 17 червня 2015 року відбувся випуск шостої версії ECMAScript. Ця версія офіційно називається ECMAScript 2015, яку найчастіше називають ECMAScript 2015 чи просто ES2015. З недавнього часу стандарти ECMAScript випускаються щороку. Ця документація відноситься до останньої версії чернетки, якою є ECMAScript 2018 [37].

JavaScript є простою для освоєння мовою, проте, завдяки її гнучкості, задовольняє безліч вимог та потреб розробника. Завдяки відкритому коду мова розвивається з величезною швидкістю, адже будь який розробник може внести вклад в її розвиток.

Java – це високорівнева, заснована на класах, об'єктно-орієнтована мова програмування, яка розроблена так, щоб мати якомога менше залежностей реалізації. Це мова програмування загального призначення, призначена для того, щоб програмісти могли писати один раз, запускати в будь-якому місці (WORA), що означає, що скомпільований код Java може працювати на всіх платформах, які підтримують Java, без необхідності перекомпіляції [38]. Програми Java зазвичай компілюються в байт-код, який може працювати на

будь-якій віртуальній машині Java (JVM) незалежно від базової архітектури комп'ютера. Синтаксис Java подібний до C і C++, але має менше засобів низького рівня, ніж будь-який з них. Середовище виконання Java надає динамічні можливості (такі як відображення та модифікація коду під час виконання), які зазвичай недоступні в традиційних скомпільованих мовах. Станом на 2019 рік Java була однією з найпопулярніших мов програмування, що використовуються відповідно до GitHub, особливо для веб-додатків клієнт-сервер, із 9 мільйонами розробників [38].

На сьогодні Java все ще є одним із лідерів на ринку, та використовується для розробки як десктопних, так і мобільних додатків.

C# – це універсальна, багатопарадигмальна мова програмування. C# охоплює дисципліни програмування зі статичною типізацією, жорсткою типізацією, лексичною, імперативною, декларативною, функціональною, загальною, об'єктно-орієнтованою (на основі класів) і компонентно-орієнтованою [39].

C# був розроблений Андерсом Хейлсбергом з Microsoft у 2000 році і пізніше був затверджений як міжнародний стандарт Ecma (ECMA-334) у 2002 році та ISO (ISO/IEC 23270) у 2003 році. Microsoft представила C# разом з .NET Framework та Visual Studio, обидва були з закритим вихідним кодом. На той час у Microsoft не було продуктів з відкритим кодом. Через чотири роки, у 2004 році, розпочався безкоштовний проект з відкритим вихідним кодом під назвою Mono, який забезпечував міжплатформний компілятор і середовище виконання для мови програмування C#. Через десять років Microsoft випустила Visual Studio Code (редактор коду), Roslyn (компілятор) і уніфіковану платформу .NET (фреймворк програмного забезпечення), усі з яких підтримують C# і є безкоштовними, з відкритим вихідним кодом і міжплатформними. Mono також приєднався до Microsoft, але не був об'єднаний з .NET [39].

Враховуючи дані статистики, а також легкість використання та легкість підтримки додатку у майбутньому, JavaScript є оптимальною мовою розробки для інформаційної системи.

У результаті висновків з розділу 2.1, вибору JavaScript мови для розробки інформаційної системи, для зручності написання фронтенд-частини було обрано бібліотеку React.

React – це безкоштовна бібліотека JavaScript із відкритим вихідним кодом для створення інтерфейсів користувача на основі компонентів інтерфейсу користувача [40].

React спрощує створення інтерактивних інтерфейсів. Розробнику необхідно лише описати, як різні частини інтерфейсу виглядають у кожному стані вашого додатку і React ефективно оновить та відрендерить лише потрібні компоненти, коли ваші дані зміняться. Декларативні інтерфейси роблять код більш передбачуваним і його набагато легше налагоджувати [40].

Особливістю розробки на React є побудова додатку на компонентах та їх перевикористанні. Компоненти зазвичай пишуться на JSX – розширенні для мови JavaScript. JSX може нагадувати мову шаблонів, але з усіма перевагами JavaScript.

Ще однією перевагою цієї бібліотеки є використання властивостей та станів.

Властивості (скорочено від «властивості») і стани є простими об'єктами JavaScript. Хоча обидва містять інформацію, яка впливає на вихідний результат візуалізації, вони відрізняються одним важливим чином: props передається компоненту (подібно до параметрів функції), тоді як стан керує всередині компонента (подібно до змінних, оголошених у функції) [41].

Однією із важливих особливостей розробки за допомогою React є перевикористання коду. Не тільки окремих компонент, але і цілих модулів. Керувати ними дозволяє `prop`.

npm – це менеджер пакетів для мови програмування JavaScript [42]. npm є менеджером пакетів за замовчуванням для середовища виконання JavaScript Node.js. Він складається з клієнта командного рядка, який також називається npm, і віддаленої бази даних загальнодоступних і платних приватних пакетів, що називається реєстром npm. Доступ до реєстру здійснюється через клієнт, а доступні пакети можна переглядати та шукати на веб-сайті npm [43].

Node.js – це кросплатформне середовище виконання JavaScript із відкритим вихідним кодом, яке працює на движку V8 і виконує код JavaScript за межами веб-браузера [33]. Node.js дозволяє розробникам використовувати JavaScript для написання інструментів командного рядка та для запуску сценаріїв на стороні сервера для створення динамічного вмісту веб-сторінки до того, як сторінка буде відправлена у веб-браузер користувача. Отже, Node.js являє собою парадигму «JavaScript всюди», що об'єднує розробку веб-додатків навколо однієї мови програмування, а не різних мов для сценаріїв на стороні сервера та клієнта [44].

Node.js дозволяє створювати веб-сервери та мережеві інструменти за допомогою JavaScript і набору «модулів», які обробляють різні основні функції. Модулі призначені для файлової системи введення / виводу, мережевих імен (DNS, HTTP, TCP, TLS / SSL або UDP), двійкових даних (буферів), криптографічних функцій, потоків даних, а також інших основних функцій. Модулі Node.js використовують API, призначений для зменшення складності написання серверних додатків.

JavaScript є єдиною мовою, яку Node.js підтримує за замовчуванням, але є багато мов компіляції в JS. В результаті програми Node.js можна писати на CoffeeScript, Dart, TypeScript, ClojureScript та інших.

Node.js в основному використовується для створення мережевих програм, таких як веб-сервери. Найбільш істотна відмінність між Node.js і PHP полягає в тому, що більшість функцій у PHP блокують до завершення (команди виконуються лише після завершення попередніх команд), тоді як функції

Node.js не блокуються (команди виконуються одночасно або навіть паралельно, і використовують зворотні виклики для сигналу про успіх або збій) [44].

У ролі СКБД було обрано платформу firebase, що прекрасно взаємодіє з додатками, розробленими на Node.js.

Firebase – це платформа, розроблена Google для створення мобільних і веб-додатків. Вона є їх флагманською пропозицією для розробки додатків.

Сервіси firebase допомагають керувати даними користувача з моменту реєстрації (це можливо зробити не тільки через розроблену авторизаційну систему, але й через Meta (Facebook), Twitter, Google чи GitHub).

Проект Firebase – це сутність верхнього рівня для Firebase. У проекті створюються програми Firebase та реєструється Apple, Android або веб-програми. Після реєстрації програм у Firebase можна додати пакети SDK Firebase для будь-якої кількості продуктів Firebase, як приклад Analytics, Cloud Firestore, моніторинг продуктивності або віддалена конфігурація [45].

Firebase Analytics – безкоштовне рішення для оцінки застосунків, яке дає змогу ознайомитись із використанням застосунків та залученням користувачів [45].

Firebase Cloud Messaging (FCM) – це крос-платформове рішення для повідомлень і нотифікацій для Android, iOS та веб-застосунків, які наразі можна використовувати безкоштовно [45].

Firebase Auth – це служба, яка може аутентифікувати користувачів, використовуючи лише код на стороні клієнта. Він підтримує соціальні логін-провайдери Facebook, GitHub, Twitter і Google (і Google Play Games). Крім того, вона включає в себе систему управління користувачами, за допомогою якої розробники можуть увімкнути автентифікацію користувача за допомогою входу з електронної пошти та пароля, що зберігаються в Firebase [45].

Firebase надає в режимі реального часу базу даних та бекенд як службу. Ця служба надає розробникам застосунків API, який дозволяє синхронізувати дані застосунків між клієнтами та зберігати їх у хмарі Firebase [45]. Компанія

також надає клієнтські бібліотеки, які дозволяють інтеграцію із застосунками Android, iOS, JavaScript / Node.js, Java, Objective-C, Swift. База даних також доступна через REST API та прив'язки до декількох сценаріїв JavaScript, таких як AngularJS, React, Ember.js та Backbone.js. REST API використовує протокол подій із сервером, який є інтерфейсом для створення HTTP-з'єднань для отримання push-повідомлень від сервера. Розробники, які використовують Realtime Database, можуть захищати свої дані за допомогою правил безпеки, що застосовуються на сервері [45].

Firestore забезпечує надійне звантаження та вивантаження файлів для застосунків Firebase незалежно від якості мережі. Розробник може використовувати його для зберігання зображень, аудіо-, відео- чи іншого вмісту, створеного користувачами. Зберігання Firestore підтримується Google Cloud Storage [45].

Firestore – це статичний та динамічний веб-хостинг, який було запущено 13 травня 2014 року. Він підтримує хостинг статичних файлів, таких як CSS, HTML, JavaScript та інші файли, а також динамічну підтримку Node.js через Cloud Functions. Служба передає файли через мережу доставки контенту (CDN) за допомогою протоколу HTTPS та шифрування SSL. Firestore підтримує Fastly, CDN, щоб забезпечити підтримку CDN Firestore Hosting. Компанія стверджує, що хостинг Firestore виріс із запитів клієнтів; розробники використовували Firestore для своєї бази даних в режимі реального часу, але вони потребували місця для розміщення їхнього вмісту [45].

4.2 Розробка інформаційної системи для проведення опитувань та тестувань

Розроблена інформаційна система реалізує наступні функціональні можливості:

- конфігурація форм та запитань;

- проведення опитування чи тестування;
- збір результатів для подальшої їх обробки.

Дана система працює як модуль, тому не передбачає сторінки реєстрації користувача, проте здатна використовувати інформацію батьківського додатку щодо персональних даних користувача (для персоналізації відповідей на питання).

Перш за все, користувачу необхідно створити форму для опитування для подальшого її опрацювання (редагування, перегляд чи видалення). Для цього користувачу потрібно заповнити поля форми.

Програмний код форми виглядає наступним чином:

```
<Form {...rest} initialValues={initialValues}>
  <Form.Item
    name="title"
    rules={[{ required: true, message:
TEXTINGS.formValidationRule }]}>
    <Input
      mb={4}
      allowClear
      placeholder={formTitlePlaceholder ||
TEXTINGS.formTitlePlaceholder}
    />
  </Form.Item>
  <Form.Item name="subtitle">
    <TextArea
      mb={4}
      showCount
      autoSize={{ minRows: 3, maxRows: 5 }}
      maxLength={1000}
      allowClear
      placeholder={
        formSubtitlePlaceholder ||
TEXTINGS.formSubtitlePlaceholder
      }
    />
  </Form.Item>
  <Form.Item
    name="isQuiz"
    label={quizSwitcherText ||
TEXTINGS.quizSwitcherText}>
    <Switch defaultChecked={!formData?.isQuiz} />
  </Form.Item>
  {children}
</Form>
```

```
</Form>
```

Після підтвердження ним коректності введених даних, відбудеться запит до колекції firebase типу SET у асинхронній функції:

```
await setData(COLLECTIONS.FORMS, formId, formData).catch((e)
=>
    message.error(e.message)
)
```

Рисунок 4.2 – Форма створення форми для проведення опитування

Одночасно в цій функції генерується повідомлення користувачу у разі помилки роботи форми чи зв'язку з базою даних.

Редагування існуючої форми відбувається таким самим чином.

Результат виконання операції зображено на рисунку 4.3:

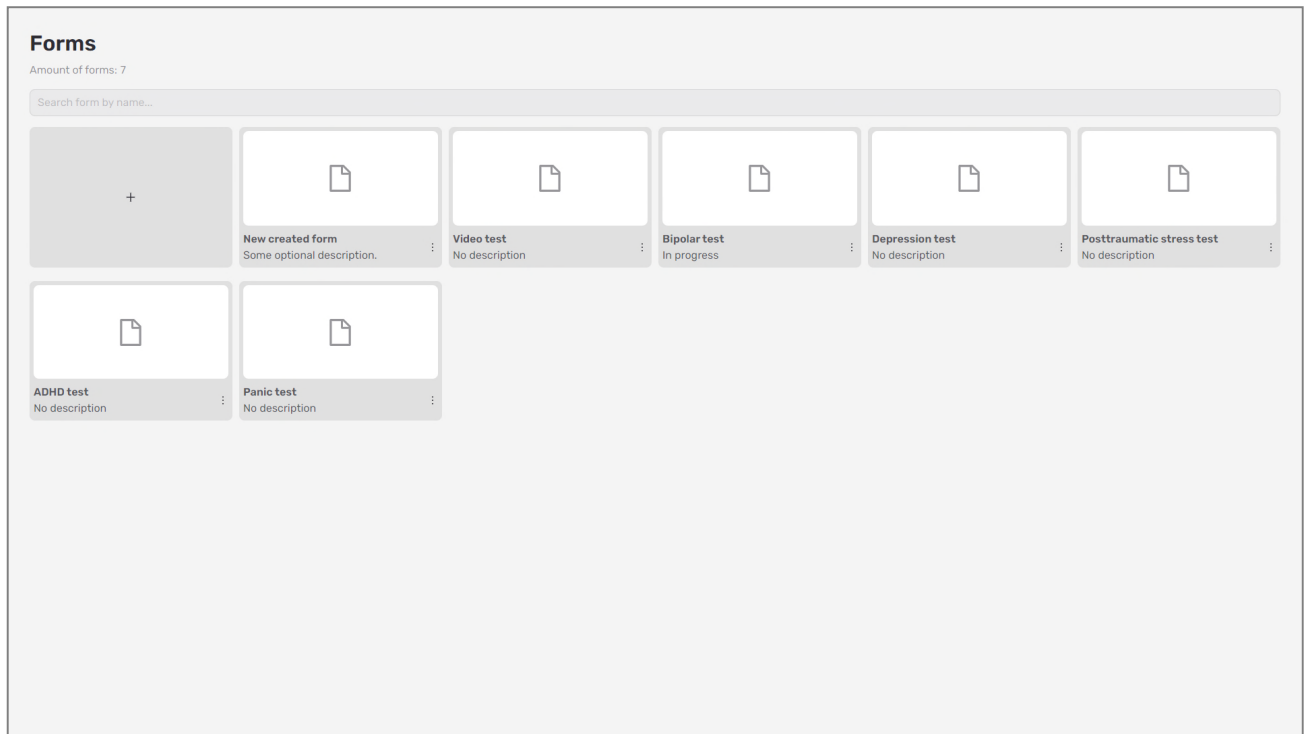


Рисунок 4.3 – Новостворена форма поряд із вже існуючими

Кожен елемент списку форм є роутинговим, та веде на сторінку перегляду редагування питань обраної форми.

Для перегляду питань форми, відбувається запит до колекції запитань на firebase з додатковою умовою – отримати лише питання, що містять id форми.

```
const [questions] = useCollectionData(
  getCollectionRef(COLLECTIONS.QUESTIONS).where('formId',
'==' formId)
)
```

Розглянемо сторінку налаштування питань форми (рис.4.4). Вона складається з двох основних блоків – переліку питань форми з drag and drop функцією для коригування порядку питань та областю перегляду самого питання.

```
<Container display="flex" height="inherit"
overflowX="hidden">
  <Box
    flex={1}
```

```

        display="flex"
        p={containerPadding}
        flexDirection="column">
        {showHeader && (
<PageHeader
        id={id}
        handlesPreview
        title={form?.title}
        onBack={history.goBack}
        showAnswers={showAnswers}
        smallScreen={smallScreen}
        />
    )}
<QuestionForm
        defaultTab={defaultTab}
        questionsList={questionsList}
        questionData={currentQuestion}
        brightnessValue={brightnessValue}
        setBrightnessValue={setBrightnessValue}
        customQuestionTypes={customQuestionTypes}
        onQuestionTypeChange={onQuestionTypeChange}
        welcomeScreenShowRule={welcomeScreenShowRule}
        onQuestionLayoutChange={onQuestionLayoutChange}
        />
</Box>
<EditorSidebar
        id={id}
        formData={form}
        endings={endings}
        questions={questions}
        answerScoresData={answerScoresList}
        customQuestionTypes={customQuestionTypes}
        welcomeScreenShowRule={welcomeScreenShowRule}
        />
</Container>

```

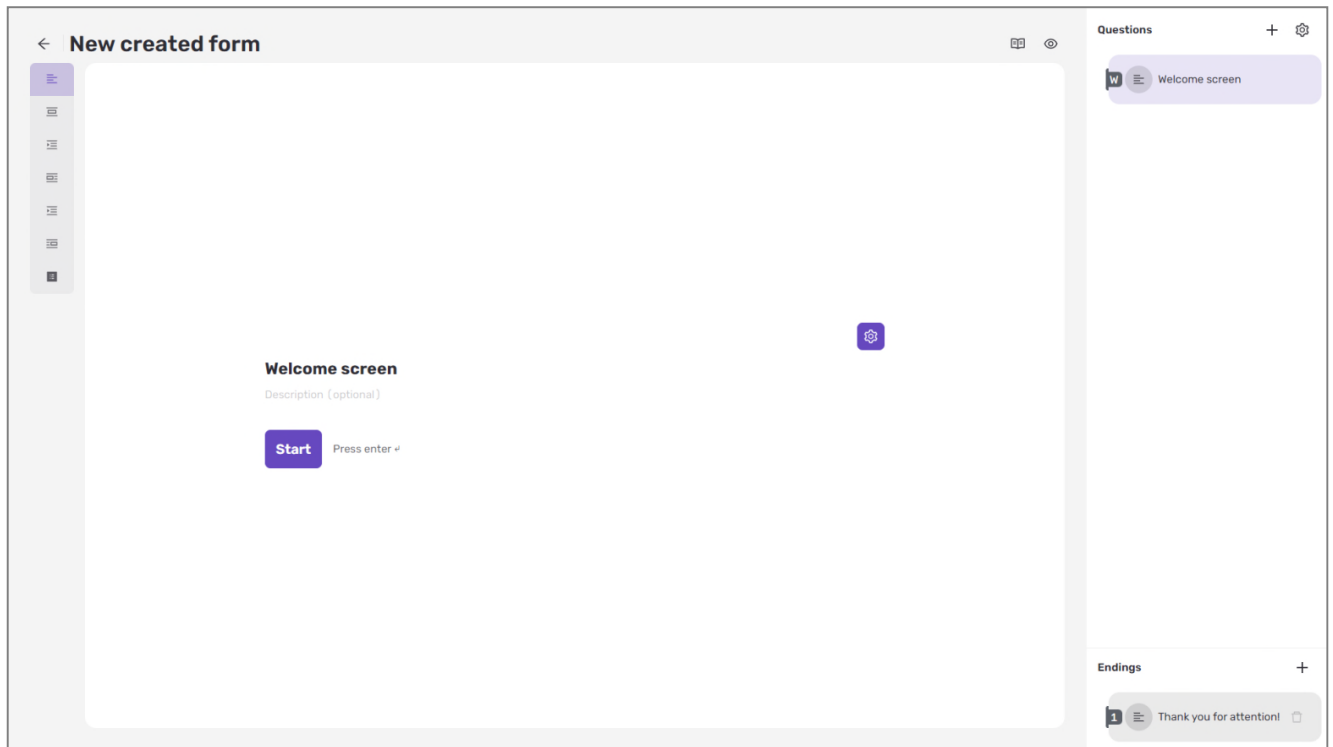


Рисунок 4.4 – Результат виконання коду: сторінка налаштування питань форми

На сторінці головними компонентами є `FormAdvancedForm` та `EditorSidebar`. Зверху розташована навігація – кнопки «назад», перегляду відповідей на питання форми та перегляд готової форми. Навігація по додатку організована за допомогою пакету `react-router-dom`.

При перегляді обраного питання його можна налаштувати – редагувати текст запитання, змінити його тип із запропонованого переліку, додати зображення та додаткові параметри (наприклад, `required`).

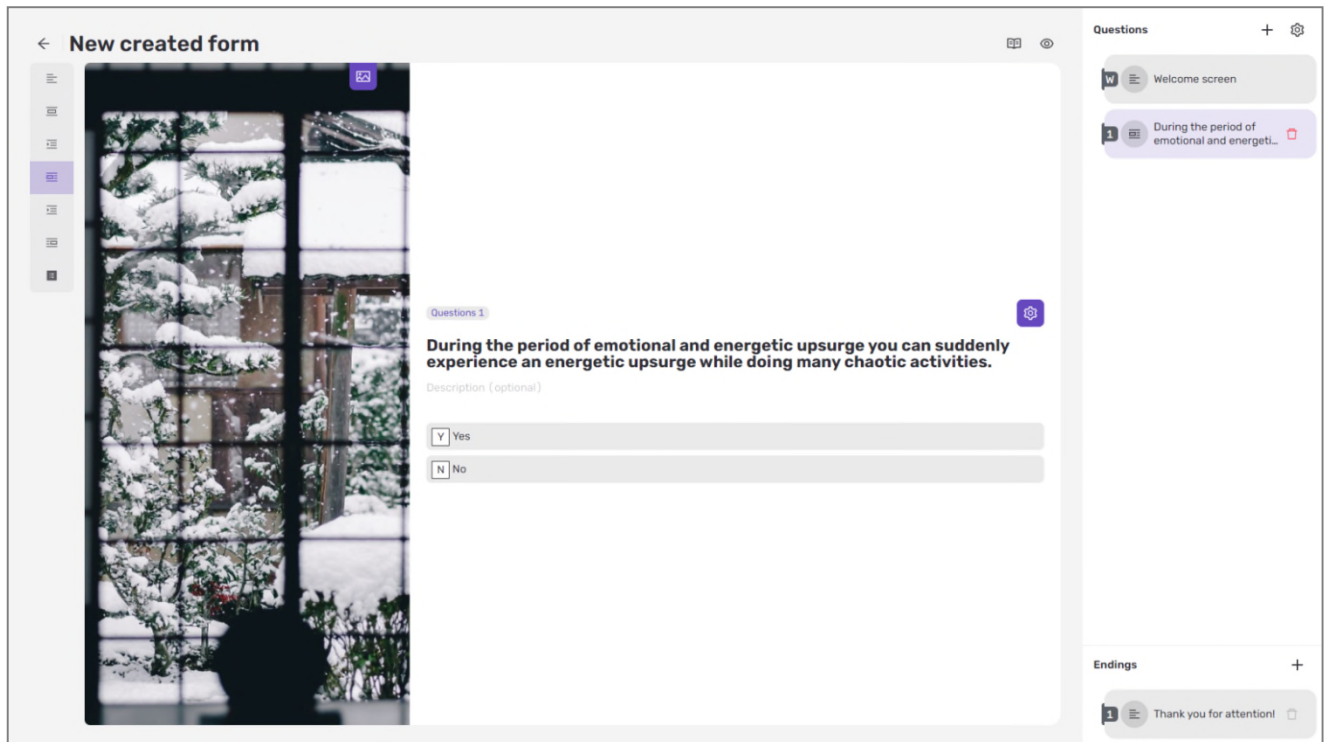


Рисунок 4.5 – Створене та налаштоване нове питання типу так-ні із вкладеним зображенням.

Поряд із зображеним типом питання, представлені наступні:

- Long text;
- Short text;
- Date;
- Yes-No;
- Choice;
- Picture choice;
- Opinion scale;
- Rating;
- File upload;
- Statement;
- Video.

Типи питання є важливою перевагою додатку, адже вони допомагають скоротити час відповіді для опитуваного – широкий спектр типів відповідей

можуть також дати більш зрозуміле середовище для проведення опитування. Тип питання обирається під час його створення, проте підтримується зміна типу питання через дропдаун-меню (рисунок 4.6):

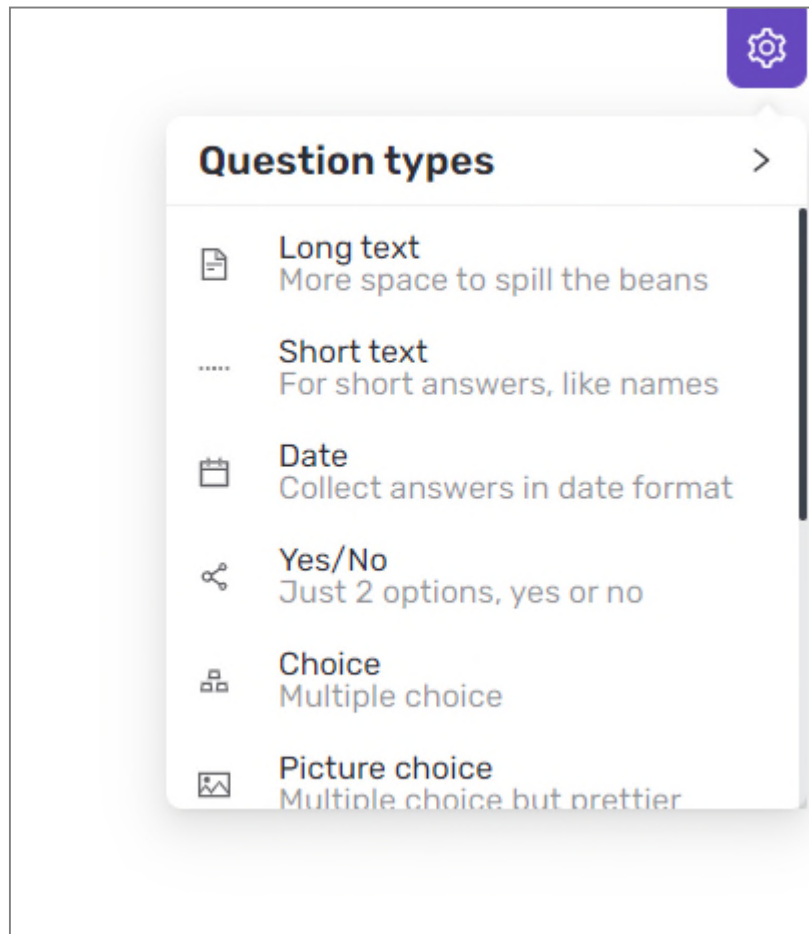


Рисунок 4.6 – Меню зміни типу питань

The screenshot shows a 'New created form' interface. The main area displays a question titled 'Describe your current state, where 1 is worst, and 5 - best.' Below the question is a horizontal scale with five buttons labeled 1, 2, 3, 4, and 5. On the right side, there is a 'Questions' panel with a list of questions: 'Welcome screen', 'During the period of emotional and energet...', and 'Describe your current state, where 1 is worst...'. At the bottom right, there is an 'Endings' panel with a 'Thank you for attention!' screen.

Рисунок 4.7 – Питання типу opinion scale.

Для проведення адаптивного опитування чи тестування є модульне вікно з налаштуваннями бальної системи та черги питань в залежності від відповіді (рис. 4.8-4.9):

The screenshot shows a 'Logic' configuration screen. It has a 'Reset logic' button in the top right corner. Below the title, there are three tabs: 'Logic jumps', 'Endings', and 'Answer score configurations'. The 'Logic jumps' tab is active. It displays three questions with their respective answer options and logic rules:

- 1** Have you ever experienced emotional and energetic upsurge?

A Yes → Go to the next question

B No → Go to the next question
- 2** During the period of emotional and energetic upsurge you can suddenly experience an energetic upsurge while doing many chaotic activities.

A Yes → Go to the next question

B No → Go to the next question
- 3** During the period of emotional and energetic upsurge it is typical for you to unexpectedly experience incredible excitement, an overly elevated, euphoric mood.

A Yes → Go to the next question

B No → Go to the next question

A 'Close' button is located at the bottom right of the screen.

Рисунок 4.8 – Модульне вікно для налаштування зв'язків питань форми

The screenshot shows a 'Logic' configuration window with a 'Reset logic' button in the top right corner. Below the title bar, there are three tabs: 'Logic jumps', 'Endings' (which is selected and underlined), and 'Answer score configurations'. The main content area contains two numbered ending configurations:

- 1** Thank you for attention! Be healthy!
Below this text is a horizontal bar containing two items: '1A Yes x' and '2A Yes x'.
- 2** There's nothing to worry about, but ask your doctor for next steps!
Below this text is an empty horizontal bar.

A 'Close' button is located in the bottom right corner of the window.

Рисунок 4.9 – Модульне вікно для налаштування закінчень проходження форми

Logic Reset logic

Logic jumps Endings Answer score configurations

1 Have you ever experienced emotional and energetic upsurge?

A Yes 10

B No 0

2 During the period of emotional and energetic upsurge you can suddenly experience an energetic upsurge while doing many chaotic activities.

A Yes 10

B No 0

3 During the period of emotional and energetic upsurge it is typical for you to unexpectedly experience incredible excitement, an overly elevated, euphoric mood.

A Yes 10

B No 0

Close

Рисунок 4.10 – Модульне вікно для конфігурації оцінок

Кількість вкладок для налаштувань на модальному вікні керується динамічно – на це впливає конфігурація самої форми. На рисунку зображено дві вкладки – із налаштуванням переходів між питаннями та коригуванням закінчення проходження форми в залежності від відповідей опитуваного.

Наступний код генерує список правил, за якими буде слідувати форма при її проходженні:

```
const addRedirectQuestion = (nextQuestion, answerIndex,
index) => {
  //create new array questionConfigurations of certain
  question

  const updatedQuestionConfigurations =
data[index]?.questionConfigurations
  // update redirect question of certain question
  const isDataChanged =

updatedQuestionConfigurations[answerIndex].redirectQuestion !==
  nextQuestion
```

```

    // if data changed - write to db, else - nothing
    if (isDataChanged) {

updatedQuestionConfigurations[answerIndex].redirectQuestion =
nextQuestion
        //write new data to db
        setData(COLLECTIONS.QUESTIONS, data[index]?.id, {
            questionConfigurations: updatedQuestionConfigurations
        })
    }
}
}

```

Оцінку чи специфічне закінчення формі задати можуть лише питання із вибором, альтернативне питання, рейтинг тощо.

```

const filteredAnswerForEndings = useMemo(
  () =>
    data
      ? data?.filter((item) =>
        [
          QUESTION_TYPES.CHOICE,
          QUESTION_TYPES.PICTURE_CHOICE,
          QUESTION_TYPES.OPINION_SCALE,
          QUESTION_TYPES.RATING,
          QUESTION_TYPES.YES_NO
        ].includes(item.questionType)
      )
      : [],
  [data]
)

```

Відображення форми опитування для проходження розроблено за допомогою компоненти «карусель» antd. До неї надходить масив налаштованих запитань, проте динамічне прорахування наступного питання відбувається лише в момент відповіді на поточне запитання.

```

const next = async (skipButtonEvent) => {
  if (currentSlideData?.isRequired && skipButtonEvent) {
    message.error(requiredAnswerMessage ||
TEXTINGS.requiredAnswerMessage)
  } else {
    //check if carousel navigation button was pressed, to
avoid repetition in answers context
    if (skipButtonEvent) {

```

```

        await setIsAnswered(true)
        //form the answer according to the answers context
structure
        const answerData = {
            question: currentSlideData,
            answer: { value: '' }
        }
        //set empty answer to answers
        answersContextDispatch({
            type: ANSWERS_DISPATCH_EVENTS.ADD_ANSWER,
            payload: answerData
        })
    }
    carouselRef.current?.next()

    setIsAnswered && setIsAnswered(false)
}
}

```

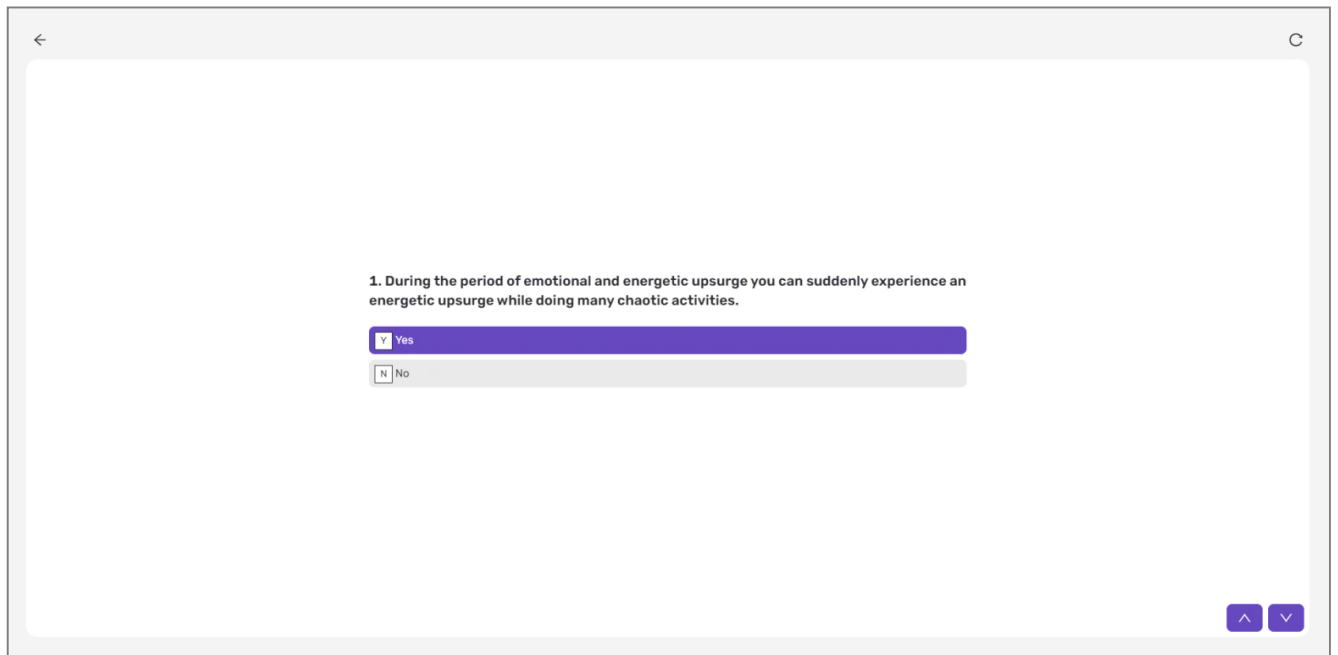


Рисунок 4.11 – Відображення форми для проходження

Опитування рахується завершеним тільки в момент натиснення кнопки Finish на фінальному слайді. В цей момент усі відповіді зберігаються до бази даних для подальшого їх опрацювання.

```

const onClick = (answerData) => {
    !!answerData &&
    answersDispatch({

```

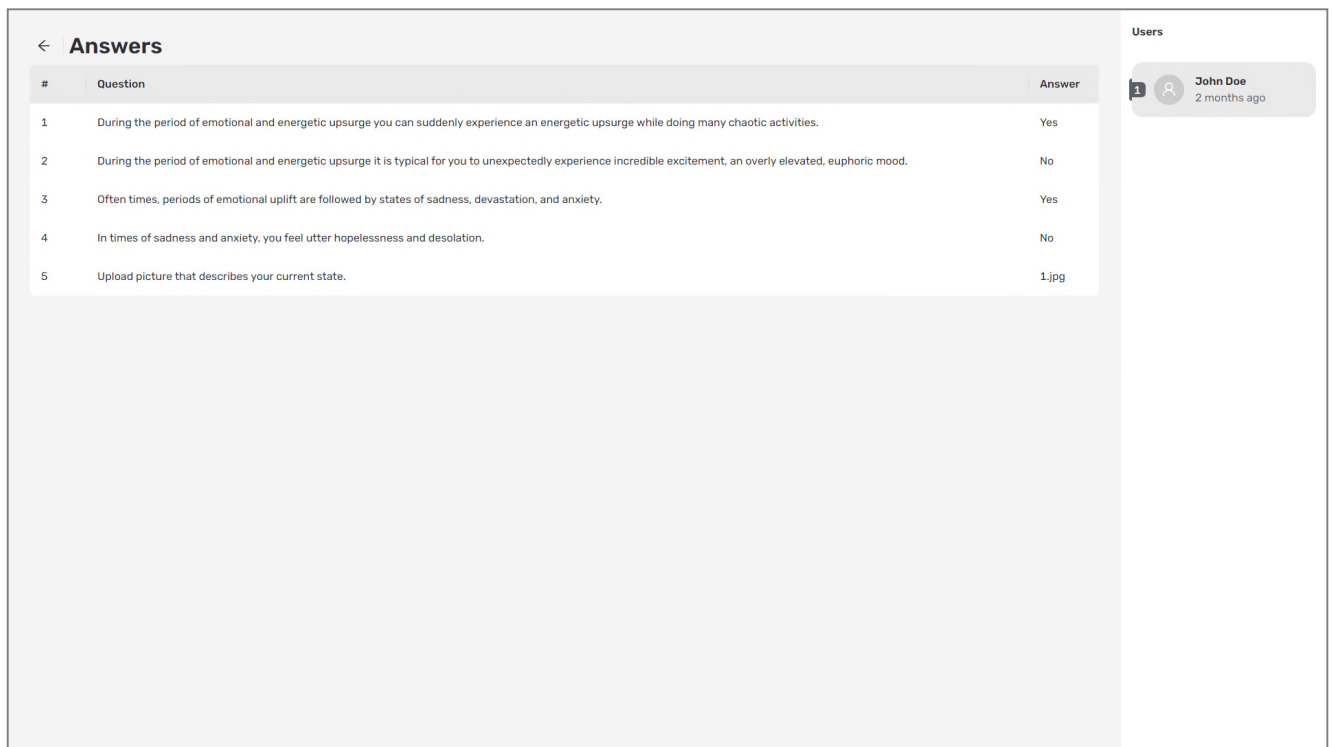
```

        type: ANSWERS_DISPATCH_EVENTS.ADD_ANSWER,
        payload: answerData
    })

    setIsAnswered(true)
    setPreviousQuestionOrder((prevState) =>
        // index: containWelcomeScreen ? questionsData?.length
- 1 : questionsData?.length
        prevState?.[prevState?.length - 1] !== currentSlide
        ? [...(prevState || []), currentSlide]
        : prevState || []
    )
}

```

Отримання даних до таблиці відповідей відбувається таким же чином, що і у попередніх сторінках – запит до firebase з вказаною назвою колекції та id форми.



The screenshot shows a mobile application interface. On the left, there is a table titled "Answers" with a back arrow. The table has three columns: "#", "Question", and "Answer". It contains five rows of data. On the right, there is a "Users" section with a profile card for "John Doe" with a timestamp "2 months ago".

#	Question	Answer
1	During the period of emotional and energetic upsurge you can suddenly experience an energetic upsurge while doing many chaotic activities.	Yes
2	During the period of emotional and energetic upsurge it is typical for you to unexpectedly experience incredible excitement, an overly elevated, euphoric mood.	No
3	Often times, periods of emotional uplift are followed by states of sadness, devastation, and anxiety.	Yes
4	In times of sadness and anxiety, you feel utter hopelessness and desolation.	No
5	Upload picture that describes your current state.	1.jpg

Рисунок 4.12 – Таблиця відповідей на форму

При включеному режимі тесту, до відповідей додається також кількість набраних балів (рис.4.13):

#	Question	Answer	Score
1	During the period of emotional and energetic upsurge you can suddenly experience an energetic upsurge while doing many chaotic activities.	Yes	10
2	During the period of emotional and energetic upsurge it is typical for you to unexpectedly experience incredible excitement, an overly elevated, euphoric mood.	No	0
3	Often times, periods of emotional uplift are followed by states of sadness, devastation, and anxiety.	Yes	10
4	In times of sadness and anxiety, you feel utter hopelessness and desolation.	Yes	10
5	Upload picture that describes your current state.	1.png	

Рисунок 4.13 – Таблиця відповідей на форму у режимі тесту

Для розробки даного модуля було використано пакет `qonsoll/qvideo`, що базується в свою чергу на стандартній `react`-бабліотеці `video-js`. За допомогою цього пакету було створено дві компоненти – рекордер та програвач.

```
import React from 'react'
import { Recorder } from '@qonsoll/qvideo'

function VideoRecording(props) {

  return (
    <Recorder
      apiKey={API_KEY}
      onUpload={onUpload}
    />
  )
}

export default VideoRecording

import React from 'react'
import { Player } from '@qonsoll/qvideo'

function VideoRecording(props) {

  return (
    <Player
      apiKey={API_KEY}
      onUpload={onUpload}
    />
  )
}
```

```
export default VideoRecording
```

Результат виконання даного коду зображений на рисунку 4.13:

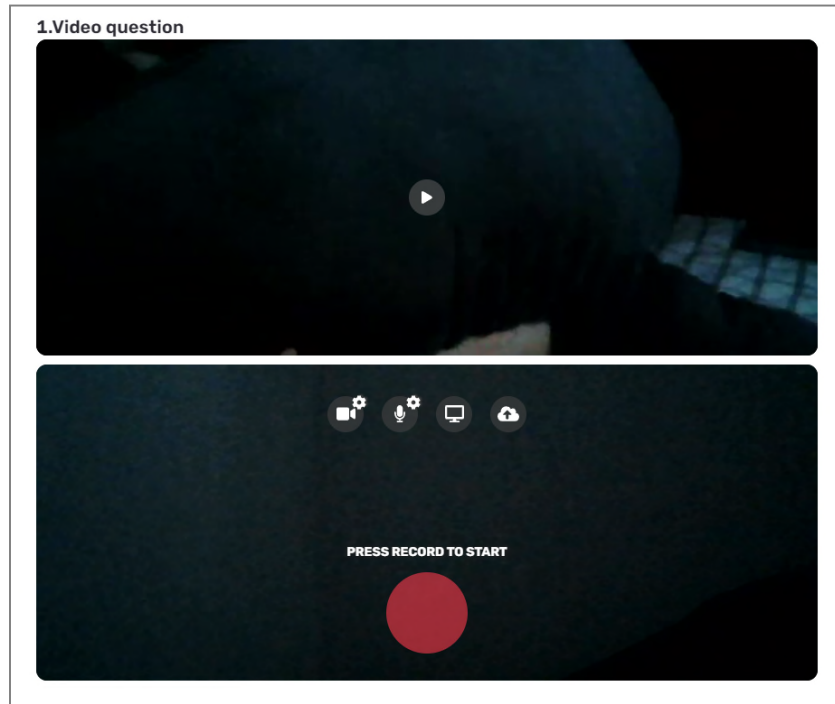


Рисунок 4.14 – Результат виконання коду: відеопрогравач та рекордер

4.3 Апробація інформаційної системи

Після розробки вдосконаленої інформаційної системи необхідно дослідити її практичну ефективність. Для цього було досліджено предметну область психології та створено тестову форму з наступними вхідними даними (таблиця 4.1):

Таблиця 4.1 – Вхідні дані.

Питання	Варіанти відповіді
Often you can feel sadness and/or anxiety.	Так, Ні
In times of sadness and anxiety, you feel utter hopelessness and desolation.	Так, Ні
During the period of emotional and energetic upsurge you can suddenly experience an energetic upsurge while	Так, Ні

doing many chaotic activities.	
Describe your current state, where 1 is worst, and 5 - best.	1-5

Для отримання доцільного результату необхідно пройти цей тест у двох режимах – за класичного підходу, та при налаштуванні адаптивного.

У першому випадку тест не буде піддаватись ніяким змінам та кожне наступне питання буде визначено до початку тесту. Результати проходження опитування за цим методом представлені у таблиці 4.2:

Таблиця 4.2 – Результат проходження опитування класичним та адаптивним методами.

Номер питання	Класичний підхід		Адаптивний підхід	
	Відповідь	Затрачений час	Відповідь	Затрачений час
1	Ні	< 1хв.	Ні	< 1 хв.
2	Ні	< 1 хв.	-	0
3	Так	1.05 хв.	Так	1 хв.
4	4	< 1 хв.	4	< 1 хв.
Загалом: прибл. 4хв.			Загалом: прибл. 3хв.	

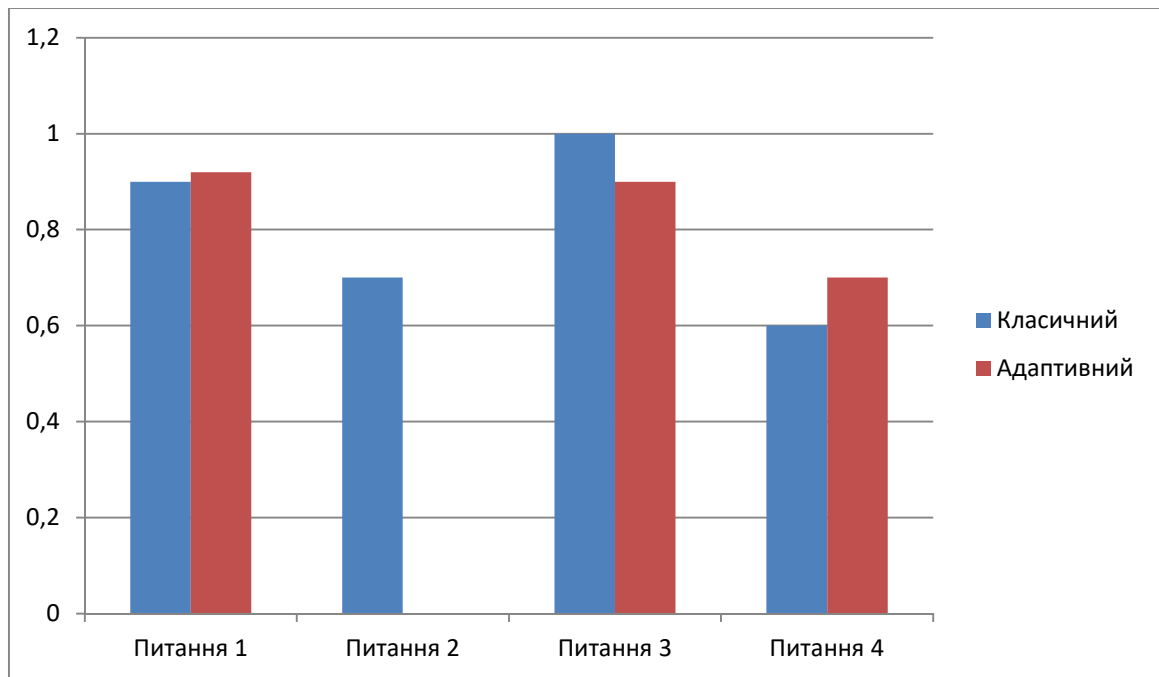


Рисунок 4.15 – Графік відображення результатів проходження опитування класичним та адаптивним підходами

Як можна помітити, при першому проходженні варіант відповіді на перше питання ніяк не вплинув на подальше тестування – усі наступні відповіді відобразились у порядку за замовчуванням.

За адаптивного тестування було встановлено зв'язок між відповідями на перше питання та індексом другого питання за позитивної відповіді, та третього за негативної. Через невелику вибірку питань, використання простих типів питань та лише одного логічного зв'язку результат покращився на 25%.

При проведенні додаткових випробувань з більшою кількістю питань та логічних зв'язків, було сформовано наступну таблицю результатів:

Таблиця 4.3 – Результати подальших випробувань.

Підхід	Загальна кількість питань	Кількість питань, на які відповіли	Затрачений час
Класичний підхід, спроба 1	24	24	28 хв.

Адаптивний підхід, спроба 1	24	21	22 хв.
Адаптивний підхід, спроба 2	24	19	19 хв.
Адаптивний підхід, спроба 3	24	21	21 хв.
Адаптивний підхід, спроба 4	24	16	18 хв.

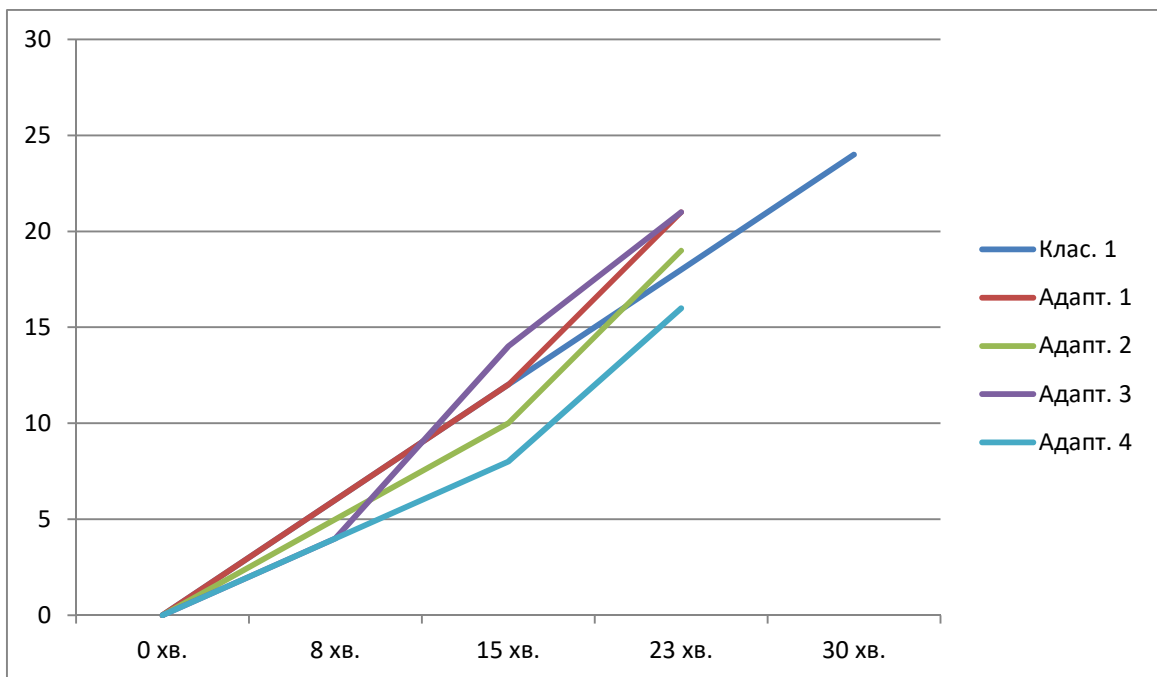


Рисунок 4.16 – Графік відображення результатів випробувань

Як добре видно, за адаптивного тестування значно скорочується як час проведення опитування, так і кількість надлишкових відповідей. Часові показники також відрізняються.

Таблиця 4.4 – Середній витрачений час на проходження тесту

Підхід	Середній час проходження
Класичний підхід	прибл. 28 хвилини
Адаптивний підхід	прибл. 20 хвилин

Висновки до розділу 4

Проведено аналіз та обґрунтовано обрані технології для розробки інформаційної системи, а саме мову програмування, сервер для розгортання додатку та сховище даних.

Розроблено глобальні та доменні компоненти користувацького інтерфейсу додатку, розроблено систему відеоопитування.

У результаті створено програмну реалізацію інформаційної системи для проведення опитувань та тестувань.

Висновки

Кваліфікаційна робота магістра вирішує задачу вдосконалення процесу проходження тестувань та опитувань. Для досягнення мети було визначено та вирішено наступні задачі:

- дослідити існуючі методи та підходи для проведення опитувань та тестувань;
- вдосконалити процес проведення опитувань та тестувань шляхом динамічної зміни наступного питання та їх загальної кількості;
- спроектувати інформаційну систему для проведення опитувань та тестувань з використанням адаптивного підходу;
- забезпечити отримання результатів опитування чи тестування у формі, придатної для подальшого проведення аналізу.

В результаті проведеної роботи були отримані такі положення наукової новизни:

- вдосконалено метод проведення тестувань та опитувань з використанням адаптивного підходу, що дало можливість зменшити кількість запитань в порівнянні з традиційними підходами без втрати точності отриманих результатів;
- розроблено інформаційну систему для проведення опитувань та тестувань з застосуванням адаптивного підходу, що дало можливість скоротити час проходження, а також усунути надлишкові варіанти відповідей до запитань;
- досліджено практичну ефективність використання адаптивного підходу під час проведення тестувань, що дозволило оцінити затрачений час на проведення тестування та якість отриманих результатів;

При виконанні поставлених задач було досліджено наразі існуючі сервіси для проведення тестувань та опитувань а також виділено їх переваги та недоліки. Найбільшим недоліком серед них був, незважаючи на невелику кількість доступних типів запитань, час затрачений на проходження тестування.

Проаналізовано підходи до тестування та взято до розгляду два основних – класичний (традиційний) та адаптивний. За результатами аналізу було обрано адаптивний підхід.

Описано мову та технології розробки, що були використані під час створення інформаційної системи. Спроектовано структуру інформаційної системи, її користувацького інтерфейсу та сховища даних.

Використання адаптивного підходу дало можливість оптимізувати цей процес скоротивши час проведення опитування в середньому на 25-30% спрощення перевірки опитування. Також була покращена якість вихідних даних – відповіді на запитання звільнені від надлишкових.

Основні наукові й практичні результати кваліфікаційної роботи магістра доповідались на XIII Всеукраїнській науково-практичній конференції «Актуальні проблеми комп'ютерних наук АПКН-2021» (15-16 жовтня 2021 року) у доповіді на тему «Застосування адаптивного підходу для реалізації системи опитувань та тестувань»; за темою роботи автором виконано наукову публікацію.

Перелік посилань

1. Социология. Словарь / Сост. Т. Е. Зерчанинова. – Екатеринбург: УрАГС, 2006. – 64 с.
2. Тест, Жан-Батист // Энциклопедический словарь Брокгауза и Ефрона : в 86 т. (82 т. и 4 доп.). – СПб., 1890 – 1907.
3. Азимов Э.Г. Новый словарь методических терминов и понятий (теория и практика обучения языкам): словарь / [под ред. Э.Г.Азимова, А.Н. Щукина]. – М.: Издательство ИКАР, 2009. – 448 с.
4. Urbina, Susana; Anastasi, Anne (1997). Psychological testing (7th ed.). Upper Saddle River, NJ: Prentice Hall. p. 4.
5. Mellenbergh, G.J. (2008). Chapter 10: Surveys. In H.J. Adèr & G.J. Mellenbergh (Eds.) (with contributions by D.J. Hand), Advising on Research Methods: A consultant's companion (pp. 183-209). Huizen, The Netherlands: Johannes van Kessel Publishing.
6. Mellenbergh, Gideon J. (1989). "Item bias and item response theory". International Journal of Educational Research. 13 (2): 127–143.
7. Wonderlic [Электронный ресурс]. – Режим доступа: <https://wonderlic.com/>
8. Aiken, Lewis R. (1998). Tests and examinations: Measuring abilities and performance. Wiley.
9. Ceci, S. J. (1991). How much does schooling influence general intelligence and its cognitive components? A reassessment of the evidence. Developmental Psychology, 27, 703–722.
10. Brown, R. (1965). Social psychology. New York: The Free Press.
11. Wuensch, Karl L. (October 4, 2005). "What is a Likert Scale? and How Do You Pronounce 'Likert?'". East Carolina University. Retrieved April 30, 2009
12. Andrich, D. (1978b) Relationships between the Thurstone and Rasch approaches to item scaling. Applied Psychological Measurement, 2, 449-460

13. Boyle, G.J., Saklofske, D.H., & Matthews, G. (2012). (Eds.), SAGE Benchmarks in Psychology: Psychological Assessment, Vol. 3: Clinical Neuropsychological Assessment. London: SAGE.
14. Ashton, M. C., (2017). Individual Differences and Personality (3rd ed.). Amsterdam: Elsevier.
15. Boyle, G.J. (1985). Self-report measures of depression: Some psychometric considerations. *British Journal of Clinical Psychology*, 24, 45-59.
16. Kenneth R. Morel (2009-09-24). "Test Security in Medicolegal Cases: Proposed Guidelines for Attorneys Utilizing Neuropsychology Practice". *Archives of Clinical Neuropsychology*.
17. Переваги та недоліки комп'ютерного тестування [Електронний ресурс]. – Режим доступу: https://studbooks.net/2178458/informatika/preimuschestva_nedostatki_kompyuternogo_testirovaniya
18. Федорук П.І. Адаптивні тести: статистичні методи аналізу результатів тестового контролю знань // Математичні машини і системи. – 2007. – № 3,4. – С. 122-138.
19. Бурлачук Л. Ф. Психодіагностика: учебник для ВУЗов. – СПб.: Питер, 2006. – 351 с.
20. Alex Chaffee. "What is a web application (or "webapp")?" – 2000-08-17
21. Gillenwater, Zoe Mickley (December 15, 2010). Examples of flexible layouts with CSS3 media queries. *Stunning CSS3*. p. 320
22. Amazon Web Services, Inc. "Mobile Application Development".
23. iOS [Електронний ресурс]. – Режим доступу: <https://www.apple.com/ios/ios-15/>
24. Android (Operating system) [Електронний ресурс]. – Режим доступу: <https://www.android.com/>
25. Google forms [Електронний ресурс]. – Режим доступу: <https://www.google.com/forms/about/>

26. Google drive [Електронний ресурс]. – Режим доступу: <https://drive.google.com/drive/>
27. Google docs [Електронний ресурс]. – Режим доступу: <https://docs.google.com/document/>
28. Survio [Електронний ресурс]. – Режим доступу: <https://www.survio.com/en/>
29. Surveymonkey [Електронний ресурс]. – Режим доступу: <https://www.surveymonkey.ru/>
30. Microsoft опитування [Електронний ресурс]. – Режим доступу: <https://support.microsoft.com/en-us/office/frequently-asked-questions-about-microsoft-forms-495c4242-6102-40a0-add8-df05ed6af61c>
31. Галкіна Р.І, Багрій Р.О, Скрипник Т.К. Застосування адаптивного підходу для реалізації системи опитувань та тестувань. – Актуальні проблеми комп'ютерних наук АПКН-2021 – Хмельницький, 15-16.10 2021р.
32. Аванесов В.С. Теория и методика педагогических измерений (материалы публикаций). – М.: ЦТ и МКО УГТУ-УПИ, 2005. – 98 с.
33. Федорук П.І. Адаптивні тести: статистичні методи аналізу результатів тестового контролю знань // Математичні машини і системи. – 2007. – № 3,4. – С. 122-138.
34. Lord F.M. Application of Item Response Theory to Practical Testing Problems. Hillsdale N-J. Lawrence Erlbaum Ass., Publ., 1980. – 266 p.
35. DOU.UA [Електронний ресурс]. – Режим доступу: <https://dou.ua>
36. JavaScript[Електронний ресурс]. – Режим доступу: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
37. ECMA-Script[Електронний ресурс]. – Режим доступу: <https://www.ecma-international.org/publications-and-standards/standards/ecma-262/>
38. Java [Електронний ресурс]. – Режим доступу: <https://www.java.com/en/>
39. C# [Електронний ресурс]. – Режим доступу: <https://docs.microsoft.com/en-us/dotnet/csharp/>
40. React[Електронний ресурс]. – Режим доступу: <https://reactjs.org/>

41. StatesinReact[Электронный ресурс]. – Режим доступа:
<https://reactjs.org/docs/faq-state.html>
42. AboutNPM[Электронный ресурс]. – Режим доступа:
<https://www.geekwire.com/2020/microsoft-owned-github-acquire-javascript-package-manager-npm/>
43. NPMofficialsite[Электронный ресурс]. – Режим доступа:
<https://www.npmjs.com/>
44. Node.js[Электронный ресурс]. – Режим доступа: <https://nodejs.org/en/>
45. Firebase[Электронный ресурс]. – Режим доступа:
<https://firebase.google.com/>

ДОДАТКИ

Додаток А

Програмні коди

```

import React from 'react'
import PropTypes from 'prop-types'
import { Carousel } from
'../../../../../../components'

const FormAdvancedView = (props)
=><Carousel {...props} />

FormAdvancedView.propTypes = {
  children: PropTypes.node,
  isAnswered: PropTypes.bool,
  setIsAnswered: PropTypes.func,
  setCurrentSlide: PropTypes.func,
  submitLoading: PropTypes.bool
}

export default FormAdvancedView

import { Tabs } from 'antd'
import PropTypes from 'prop-types'
import React, { useMemo } from
'react'
import {
  ConditionForm,
  ScoreConditionsAdvancedView,
  EndingsSimpleView
} from
'../../../../../../domains/Condition/components'
import useFunctions from
'../../../../../../hooks/useFunctions'
import { Container, Box, NoData }
from '@qonsoll/react-design'
import { useTranslation } from
'../../../../../../context/Translation'
import { EmptyState, CustomTabs }
from './FormConditionsForm.styles'
import { QUESTION_TYPES,
COLLECTIONS, TEXTINGS } from
'../../../../../../constants'

const { TabPane } = Tabs

function FormConditionsForm(props) {
  const { data, endings, formData,
onTabChange, answerScores } = props

  // [ADDITIONAL HOOKS]
  const { setData } = useFunctions()
  const {
    conditionsQuizTab,
    conditionsEndingsTab,
    conditionsLogicJumpsTab,
    conditionsNoData,
    conditionAddQuestionType
  } = useTranslation()

  // [CLEAN FUNCTIONS]
  const getQuestionListRedirect =
(itemIndex) => {
    return data?.filter((_, index)
=> itemIndex !== index)
  }

  const addCondition = (answer,
index) => {
    setData(COLLECTIONS.QUESTIONS,
data?.[index]?.id, {
      questionConfigurations:
[...data?.[index]?.questionConfigurations,
answer]
    })
  }

  const addRedirectQuestion =
(nextQuestion, answerIndex, index) => {
    //create new array
questionConfigurations of certain question

    const
updatedQuestionConfigurations =
data[index]?.questionConfigurations
    // update redirect question of
certain question
    const isDataChanged =

updatedQuestionConfigurations[answerIndex].
redirectQuestion !==
nextQuestion
    // if data changed - write to
db, else - nothing
    if (isDataChanged) {
      updatedQuestionConfigurations[answerIndex].
redirectQuestion = nextQuestion
      //write new data to db
      setData(COLLECTIONS.QUESTIONS,
data[index]?.id, {
        questionConfigurations:
updatedQuestionConfigurations
      })
    }
  }

  const findAnswerScoreByQuestionId
= (questionId) =>
  answerScores?.find((item) =>
item?.questionId === questionId)

  // [COMPUTED PROPERTIES]
  const filteredAnswerForEndings =
useMemo(
    () =>
    data
    ? data?.filter((item) =>
[
      QUESTION_TYPES.CHOICE,
QUESTION_TYPES.PICTURE_CHOICE,
QUESTION_TYPES.OPINION_SCALE,
QUESTION_TYPES.RATING,
QUESTION_TYPES.YES_NO
].includes(item.questionType)
)
    : [],
    [data]
  )

  return (

```

```

<Container>
  <CustomTabs onChange={onTabChange}>
    <TabPage>
      tab={conditionsLogicJumpsTab ||
TEXTINGS.conditionsLogicJumpsTab}
        key="1"
        style={{
          overflowY: 'scroll',
          overflow: 'hidden'
        }}>
        {!!data?.length ? (
          data?.map((item, index)
=> (
            <Box mb={3} ml={2} key={index}>
              <ConditionForm
                key={index}
                item={item}
                index={index}

addCondition={addCondition}

addRedirectQuestion={addRedirectQuestion}

getQuestionListRedirect={getQuestionListRed
irect}
              />
            </Box>
          ))
        ) : (
          <EmptyState

image={NoData.PRESENTED_IMAGE_SIMPLE}

description={conditionsNoData ||
TEXTINGS.conditionsNoData}
              />
            </TabPage>
            <TabPage
              tab={conditionsEndingsTab
|| TEXTINGS.conditionsEndingsTab}
              key="2"
              style={{ overflowY:
'scroll', overflow: 'hidden' }}>
            {!!filteredAnswerForEndings?.length &&
!!endings?.length ? (
              endings?.map((item,
index) => (
                <Box mb={3} ml={2} key={index}>
                  <EndingsSimpleView
                    key={index}
                    item={item}
                    index={index}

addCondition={addCondition}

questionsData={filteredAnswerForEndings}

addRedirectQuestion={addRedirectQuestion}

getQuestionListRedirect={getQuestionListRed
irect}
              />
            </Box>
          ))
        ) : (
          <EmptyState
image={NoData.PRESENTED_IMAGE_SIMPLE}

description={conditionsNoData ||
TEXTINGS.conditionsNoData}>
          {`${
conditionAddQuestionType ||
TEXTINGS.conditionAddQuestionType
}:`}
          <br />
          {`${QUESTION_TYPES.CHOICE},
${QUESTION_TYPES.PICTURE_CHOICE},
${QUESTION_TYPES.OPINION_SCALE},
${QUESTION_TYPES.RATING},
${QUESTION_TYPES.YES_NO}`}
          </EmptyState>
        )}
      </TabPage>
    )}
  </CustomTabs>
</Container>
)
}

FormConditionsForm.propTypes = {
  onTabChange: PropTypes.func,
  data: PropTypes.array.isRequired,
  endings: PropTypes.array
}

```

```

export default FormConditionsForm

import { Form } from 'antd'
import React, { useMemo } from
'react'
import { TEXTINGS } from
'../../../../../../constants'
import { TextArea, Input, Switch }
from '@qonsoll/react-design'
import { useTranslation } from
'../../../../../../context/Translation'

// const { TextArea } = Input
function FormSimpleForm(props) {
  const { formData, children,
...rest } = props

  // [ADDITIONAL_HOOKS]
  const { formTitlePlaceholder,
formSubtitlePlaceholder, quizSwitcherText }
=
  useTranslation()

  // [COMPUTED PROPERTIES]
  const initialValues = useMemo(()
=> {
    const { title, subtitle = '',
isQuiz = false } = formData || {}
    return {
      title,
      subtitle,
      isQuiz
    }
  }, [formData])

  return (
    <Form {...rest}
initialValues={initialValues}>
      <Form.Item
        name="title"
        rules={[{ required: true,
message: TEXTINGS.formValidationRule }]}>
        <Input
          mb={4}
          allowClear

placeholder={formTitlePlaceholder ||
TEXTINGS.formTitlePlaceholder}
        />
      </Form.Item>
      <Form.Item name="subtitle">
        <TextArea
          mb={4}
          showCount
          autoSize={{ minRows: 3,
maxRows: 5 }}
          maxLength={1000}
          allowClear
          placeholder={
            formSubtitlePlaceholder
            || TEXTINGS.formSubtitlePlaceholder
          }
        />
      </Form.Item>
      <Form.Item
        name="isQuiz"
        label={quizSwitcherText ||
TEXTINGS.quizSwitcherText}>
        <Switch
          defaultChecked={!!formData?.isQuiz} />

```

```

</Form.Item>
      {children}
    </Form>
  )
}

FormSimpleForm.propTypes = {}

export default FormSimpleForm

import { Form, Modal } from 'antd'
import React, { useState } from
'react'
import { TEXTINGS } from
'../../../../../../constants'
import { Title, Button } from
'@qonsoll/react-design'
import { useTranslation } from
'../../../../../../context/Translation'
import { FormSimpleForm } from
'../../../../../../domains/Form/components'

const FormSimpleFormWithModal =
(props) => {
  const {
    formData,
    setEdit,
    isModalVisible,
    setIsModalVisible,
    onModalSubmit,
    isEdit,
    children
  } = props

  // [STATE]
  const [loading, setLoading] =
useState(false)

  // [ADDITIONAL_HOOKS]
  const [form] = Form.useForm()
  const {
    formModalCreateTitle,
    formModalEditTitle,
    formModalEditButton,
    formModalCreateButton
  } = useTranslation()

  // [HELPER_FUNCTIONS]
  const handleCancel = async () => {
    await setLoading(false)
    await setIsModalVisible(false)
    form.resetFields()
  }

  const onFormEdit = (data) => {
    setLoading(true)
    onModalSubmit(data)
    !isEdit && form.resetFields()
    setLoading(false)
    setIsModalVisible(false)
    setEdit?.(isEdit)
  }

  // [COMPUTED PROPERTIES]

  return (
    <Modal
      title={
        <Title color="var(--qf-typography-
title-color)" level={4}>
          {isEdit}

```

```

        ? formModalEditTitle ||
TEXTINGS.formModalEditTitle
        : formModalCreateTitle
|| TEXTINGS.formModalCreateTitle
    </Title>
    }
    visible={isModalVisible}
    onCancel={handleCancel}
    footer={
    <Button
        onClick={() => {
            handleCancel()
        }}>
        Cancel
    </Button>,
    <Button onClick={() =>
form.submit()} type="primary"
loading={loading}>
        {isEdit
            ? formModalEditButton ||
TEXTINGS.formModalEditButton
            : formModalCreateButton
|| TEXTINGS.formModalCreateButton}
        </Button>
    ]}>
    <FormSimpleForm
        form={form}
        formData={formData}
        children={children}
        onFinish={onFormEdit}
    />
    </Modal>
    )
}

export default
FormSimpleFormWithModal

import Fuse from 'fuse.js'
import PropTypes from 'prop-types'
import { Modal, Upload, message }
from 'antd'
import { StaticList } from
'../../../../components'
import useFunctions from
'../../../../hooks/useFunctions'
import React, { useEffect, useRef,
useState } from 'react'
import { COLLECTIONS, TEXTINGS }
from '../../../../constants'
import { SearchOutlined,
EditOutlined } from '@ant-design/icons'
import { useTranslation } from
'../../../../context/Translation'
import { useCollectionData } from
'react-firebase-hooks/firestore'
import { Row, Col, Box, Input,
Title, Button } from '@qonsoll/react-
design'
import {
    MediaListContainer,
    CustomButton,
    CustomText
} from './MediaLibraryModal.styles'

function MediaLibraryModal(props) {
    const { btnProps, onClick,
onContinue, isHovering } = props

    // [CUSTOM_HOOKS]

```

```

        const { getCollectionRef, setData,
storage } = useFunctions()

        // [ADDITIONAL HOOKS]
        const [media = []] =
useCollectionData(getCollectionRef(COLLECTI
ONS.MEDIA))
        const {
            mediaLibraryCounter,
            mediaLibraryButton,
            mediaLibrarySearchPlaceholder,
            mediaLibraryTitle
        } = useTranslation()

        const searchRef = useRef()

        // [COMPONENT STATE HOOKS]
        const [selectedBackgroundImg,
setSelectedBackgroundImg] = useState(false)
        const [isModalVisible,
setIsModalVisible] = useState(false)
        const [imagesList, setImagesList]
= useState(media)

        // [COMPUTED PROPERTIES]
        const amountFiles =
imagesList.length
        const fuse = new Fuse(media, {
keys: ['title'] })

        // [CLEAN FUNCTIONS]
        const onMediaUploaded = (data) =>
{
            const mediaId =
getCollectionRef(COLLECTIONS.MEDIA).doc().i
d
            setData(COLLECTIONS?.MEDIA,
mediaId, { id: mediaId, ...data
}).catch((e) =>
                message.error(e.message)
            )
        }
        const onModalContinue = async ()
=> {
            setIsModalVisible(!isModalVisible)
            onContinue &&
onContinue(selectedBackgroundImg)
        }
        const onModalCancel = () => {
            setIsModalVisible(!isModalVisible)
        }
        const modalStateChange = () => {
            setIsModalVisible(!isModalVisible)
            onClick?.()
        }
        const onChange = (input) => {
            searchData(input.target.value)
        }
        const customRequest = (data) => {
            const { onSuccess } = data
            const ref =
storage.ref('images').child(data.file.uid)
            const image = ref.put(data.file)
            image.on(
                'state_changed',
                (snapshot) => {},
                (error) => {

```

```

upload
    // Handle error during the
    message.error(error.message)
  },
  () => {
    image.snapshot.ref
      .getDownloadURL()
      .then((downloadURL) => {
        setImagesList([
          ...imagesList,
          {
            title:
              data.file.name,
            imageUrl:
              downloadURL
          }
        ])
        onMediaUploaded({
          title: data.file.name,
          imageUrl: downloadURL
        })
      })
      .then(() => onSuccess())
  }
)
const searchData = () => {
  if
    (searchRef.current.input.value) {
      const searchRes =
        fuse.search(searchRef.current.input.value)
      setImagesList(searchRes?.map((item) =>
        item.item))
    } else setImagesList(media)
  }

  const beforeUpload = (file) => {
    if
      (!file?.type?.includes('image')) {
        message.error(`${file.name}
        ${TEXTINGS.mediaLibraryTypeError}`)
      }
    return
      file?.type?.includes('image') ? true :
      Upload.LIST_IGNORE
  }

  // [USE_EFFECTS]
  useEffect(() => {
    let isComponentMounted = true
    isComponentMounted && imagesList
    && setImagesList(media)

    // [CLEAN UP FUNCTION]

    return () => {
      // [FINAL CLEAN UP]
      isComponentMounted = false
    }
    // eslint-disable-next-line
  }, [media])

  return (
    <>
      {isHovering && (
        <CustomButton {...btnProps}
        onClick={modalStateChange}>
          <Box display="flex">
            <Box mr={2}>
              <EditOutlined />
            </Box>
            {mediaLibraryButton ||
              TEXTINGS.mediaLibraryButton}
          </Box>
        </CustomButton>
      )}
      <Modal
        visible={isModalVisible}
        footer={null}
        closable={false}
        width="1024px"
        centered
        bodyStyle={{
          // gonna use this height
          when filter is on
            // height: '768px',
            padding: 0,
            zIndex: 10000
          }}>
        <Row v="center" py={3} px={3}>
          <Col>
            <Title color="var(--qf-typography-
            title-color)" level={3}>
              {mediaLibraryTitle ||
                TEXTINGS.mediaLibraryTitle}
            </Title>
            <CustomText>
              {`${mediaLibraryCounter ||
                TEXTINGS.mediaLibraryCounter}: `}
              {amountFiles}
            </CustomText>
          </Col>
        </Row>
        <Row px={3} pb={3}>
          <Col>
            <Input
              allowClear
              ref={searchRef}
              prefix={<SearchOutlined />}
              placeholder={`${
                mediaLibrarySearchPlaceholder ||
                TEXTINGS.mediaLibrarySearchPlaceholder
              }...`}
              onSearch={searchData}
              onChange={onChange}
            />
          </Col>
        </Row>
        <MediaListContainer px={4} pt={2}>
          {/* RENDER MEDIA */}
          <Box width="100%" mr="-10px">
            <StaticList
              hasMedia
              data={imagesList}
            >
              beforeUpload={beforeUpload}
              customRequest={customRequest}
              selectedBackgroundImg={selectedBackgroundIm
              g}
              setSelectedBackgroundImg={setSelectedBackgr
              oundImg}
            </StaticList>
          </Box>
        </MediaListContainer>
      </>
    </>
  )
}

```

```

        />
    </Box>
  </MediaListContainer>
  <Row
    p={3}
    h="right"
    noGutters

borderBottomLeftRadius="var(--qf-border-
radius-md)"

borderBottomRightRadius="var(--qf-border-
radius-md)">
  <Col cw="auto" mr={2}>
    <Button type="text"
onClick={onModalCancel}>

{TEXTINGS.mediaLibraryCancelButton}
  </Button>
</Col>
  <Col cw="auto">
    <Button type="primary"
onClick={onModalContinue}>

{TEXTINGS.mediaLibrarySubmitButton}
  </Button>
</Col>
</Row>
</Modal>
</>
)
}
MediaLibraryModal.defaultProps = {
  btnProps: { children: 'Open Modal'
}
}
MediaLibraryModal.propTypes = {
  btnProps:
PropTypes.object.isRequired,
  data: PropTypes.array
}

export default MediaLibraryModal

import React from 'react'
import PropTypes from 'prop-types'
import { Box } from '@qonsoll/react-
design'
import { TEXTINGS } from
'../../../../../constants'
import RangeSlider from
'../../../../../components/RangeSlider'
import { useTranslation } from
'../../../../../context/Translation'
import { CustomBox, CustomText }
from './MediaLibrarySimpleView.styles'
import { MediaLibraryModal } from
'../../../../../domains/MediaLibrary/component
s'

import {
  useCurrentQuestionContextDispatch,
  DISPATCH_EVENTS
} from
'../../../../../context/CurrentQuestion'
import { useHover } from
'@umijs/hooks'

function
MediaLibrarySimpleView(props) {
  const {
    bgImage,
    brightnessValue,
    setBrightnessValue,
    setIsImageEditVisible
  } = props

  // [ADDITIONAL HOOKS]
  const currentQuestionDispatch =
useCurrentQuestionContextDispatch()
  const [isHovering, hoverRef] =
useHover()
  const { mediaLibraryBrightness } =
useTranslation()

  // [CLEAN FUNCTIONS]
  const onMediaModalContinue =
(selectedImage) => {
    currentQuestionDispatch({
      type:
DISPATCH_EVENTS.UPDATE_CURRENT_QUESTION,
      payload: { image:
selectedImage }
    })
  }
  const onBlur = () => {
    currentQuestionDispatch({
      type:
DISPATCH_EVENTS.UPDATE_CURRENT_QUESTION,
      payload: { imageBrightness:
brightnessValue }
    })
  }

  return (
    <Box>
      <CustomBox backgroundImage={bgImage}
mb={2} ref={hoverRef}>
        <MediaLibraryModal
          onClick={() => {
            setIsImageEditVisible(false)
          }}

          onContinue={onMediaModalContinue}
          isHovering={isHovering}
          btnProps={{
            type: 'primary'
          }}
        />
      </CustomBox>
      <CustomText>
        {mediaLibraryBrightness ||
TEXTINGS.mediaLibraryBrightness}
      </CustomText>
      <Box>
        <RangeSlider
          onBlur={onBlur}

          inputValue={brightnessValue}

          setInputValue={setBrightnessValue}
        />
      </Box>
    </Box>
  )
}

MediaLibrarySimpleView.propTypes = {
  setIsImageEditVisible:
PropTypes.func,

```

```

    bgImage: PropTypes.string
  }

  export default
  MediaLibrarySimpleView

  import useMedia from 'use-media'
  import PropTypes from 'prop-types'
  import React, { cloneElement } from
  'react'
  import SideLayoutImage from
  '../QuestionForm/SideLayoutImage'
  import MiddleLayoutImage from
  '../QuestionForm/MiddleLayoutImage'
  import { useTranslation } from
  '../../../../../../context/Translation'
  import { Col, Row, Box, Title, Text
  } from '@qonsoll/react-design'
  import { QUESTION_TYPES,
  LAYOUT_TYPES, TEXTINGS } from
  '../../../../../../constants'
  import {
    styles,
    StyledBox,
    BackgroundImage
  } from
  '../QuestionAdvancedView.styles'
  import {
    Rate,
    ShortText,
    YesnoButton,
    RangeButton,
    ChoiceButton,
    FileUploader,
    LongText,
    DateTimeInput,
    SubmitButton,
    VideoPlayer,
    VideoAnswer
  } from '../../../../../../components'

  function QuestionAdvancedView(props)
  {
    const {
      data,
      onClick,
      isFormQuiz,
      currentSlide,
      submitLoading,
      questionNumber,
      answersScoreData,
      preventFirebaseUsage
    } = props
    // [ADDITIONAL_HOOKS]
    const { questionFinishButton,
    questionStartButton, questionVideo } =
    useTranslation()

    // [COMPUTED PROPERTIES]
    const questionTypesMap = {
      [QUESTION_TYPES.YES_NO]: {
        component: <YesnoButton
onClick={onClick}
currentSlide={currentSlide} />
      },
      [QUESTION_TYPES.PICTURE_CHOICE]:
    {
      component: (
        <ChoiceButton

```

```

      choices={data?.questionConfigurations}
        onClick={onClick}
      )
    },
    [QUESTION_TYPES.CHOICE]: {
      component: (
        <ChoiceButton
choices={data?.questionConfigurations}
        onClick={onClick}
      )
    },
    [QUESTION_TYPES.OPINION_SCALE]:
    {
      component: (
        <RangeButton onKeyDown
onClick={onClick}
currentSlide={currentSlide} />
      )
    },
    [QUESTION_TYPES.RATING]: {
      component: <Rate
onClick={onClick}
currentSlide={currentSlide} />
    },
    [QUESTION_TYPES.SHORT_TEXT]: {
      component: <ShortText
onClick={onClick}
currentSlide={currentSlide} />
    },
    [QUESTION_TYPES.LONG_TEXT]: {
      component: <LongText
onClick={onClick}
currentSlide={currentSlide} />
    },
    [QUESTION_TYPES.DATE]: {
      component: (
        <DateTimeInput
onDateChange={onClick}
currentSlide={currentSlide} />
      )
    },
    [QUESTION_TYPES.FILE_UPLOAD]: {
      component: (
        <FileUploader onContinue={onClick}
currentSlide={currentSlide} />
      )
    },
    [QUESTION_TYPES.STATEMENT]: {
      component: <SubmitButton
onClick={onClick}
currentSlide={currentSlide} />
    },
    [QUESTION_TYPES.VIDEO_ANSWER]: {
      component: (
        <Box>
        <VideoAnswer onClick={onClick} />
        </Box>
      )
    },
    [QUESTION_TYPES.WELCOME_SCREEN]:
    {

```

```

        component: (
          <SubmitButton onClick={onClick}
currentSlide={currentSlide}>
            {questionStartButton ||
TEXTINGS.questionStartButton}
          </SubmitButton>
        ),
        [QUESTION_TYPES.ENDING]: {
          component: (
            <SubmitButton
              finish
              onClick={onClick}
              loading={submitLoading}
            >
currentSlide={currentSlide}

preventFirebaseUsage={preventFirebaseUsage}
>
            {questionFinishButton ||
TEXTINGS.questionFinishButton}
          </SubmitButton>
        )
      }
    }
  //component for recieved question
  according to question type
  const component =
questionTypesMap[data?.questionType].compon
ent
  const layoutType =
LAYOUT_TYPES[data?.layoutType]

  // defines special question
  layouts
  const specialLayoutRule =
    data?.questionType ===
QUESTION_TYPES.WELCOME_SCREEN ||
    data?.questionType ===
QUESTION_TYPES.ENDING

  //rule for template to render
  column with image, when layout type ===
  left/right(small/big)
  const imageShowRule =
    layoutType.type !==
LAYOUT_TYPES.BETWEEN.type &&
    layoutType.type !==
LAYOUT_TYPES.FULL_SCREEN.type &&
    layoutType.type !==
LAYOUT_TYPES.DEFAULT.type
  const backgroundRule =
    layoutType.type ===
LAYOUT_TYPES.FULL_SCREEN.type
  const bgImage = {
    ...(layoutType.type ===
LAYOUT_TYPES.FULL_SCREEN.type
  ? {
      backgroundRepeat: 'no-
repeat',
      backgroundImage:
`url(${data?.image})`
    }
    : {}
  )
  const widthTablet = useMedia({
maxWidth: '832px' })
  const deviceImageHeight =
(widthTablet && '40%') || '100%'

    const devicePadding = (widthTablet
&& 2) || 4

    const tabletImageCheck =
widthTablet
      ? imageShowRule
      ? 'start'
      : 'center'
      : 'center'

    const questionNumberRule =
    data?.questionType !==
QUESTION_TYPES.WELCOME_SCREEN

    return (
      <Row {...styles.mainRowStyle}
noGutters>
        {backgroundRule && (
          <BackgroundImage
            cw={12}
            image={bgImage}

imageBrightness={data?.imageBrightness ||
0}
          >
        )}
        <Col mx={2} order={2} cw=[[12, 12,
6, 6]] alignSelf={tabletImageCheck}>
          <StyledBox
            pl={devicePadding}
            pr={devicePadding}
            bordered={false}

specialLayoutRule={specialLayoutRule}>
            <Row noGutters py={2} width="100%">
              <Col cw={12}>
                {data?.isVideoQuestion
? (
                  <Box
                    height={350}

position="relative"
                    display="flex"

flexDirection="column"
                    justifyContent="center">
                    {questionNumberRule && (
                      <Title
                        color="var(--
qf-typography-title-color)"
                        style={{
wordBreak: 'break-word' }}
                        level={4}>
                    {questionNumber}.{questionVideo ||
TEXTINGS.questionVideo}
                      </Title>
                    )}
                  <VideoPlayer
                    autoLoad

videoKey={data?.videoApiKey}

autoStart="false"
                    customOptions={{
                      autoplay:
false
                    }}
                )}
              )}
            )}
          )}
        )}
      )}
    )}
  )}

```



```

    } = props

    console.log(questionData)

    // [ADDITIONAL HOOKS]
    const currentQuestion =
useCurrentQuestionContext()
    const {
      questionStartButton,
      questionFinishButton,
      endingListTitle,
      questionListTitle,
      questionEditableTitleHint,
      questionEditableSubtitleHint
    } = useTranslation()
    // [COMPUTED PROPERTIES]
    const questionTypesMap = {
      [QUESTION_TYPES.WELCOME_SCREEN]:
{
      component: (
        <SubmitButton>
          {questionStartButton ||
TEXTINGS.questionStartButton}
        </SubmitButton>
      )
    },
    [QUESTION_TYPES.YES_NO]: {
      component: <YesnoButton />
    },
    [QUESTION_TYPES.PICTURE_CHOICE]:
{
      component:
<ChoiceEditableGroup withImage />
    },
    [QUESTION_TYPES.CHOICE]: {
      component:
<ChoiceEditableGroup />
    },
    [QUESTION_TYPES.OPINION_SCALE]:
{
      component: <RangeButton />
    },
    [QUESTION_TYPES.RATING]: {
      component: <Rate />
    },
    [QUESTION_TYPES.SHORT_TEXT]: {
      component: <ShortText />
    },
    [QUESTION_TYPES.LONG_TEXT]: {
      component: <LongText />
    },
    [QUESTION_TYPES.DATE]: {
      component: <DateTimeInput />
    },
    [QUESTION_TYPES.FILE_UPLOAD]: {
      component: <FileUploader />
    },
    [QUESTION_TYPES.STATEMENT]: {
      component: <SubmitButton />
    },
    [QUESTION_TYPES.ENDING]: {
      component: (
        <SubmitButton>
          {questionFinishButton ||
TEXTINGS.questionFinishButton}
        </SubmitButton>
      )
    },
    [QUESTION_TYPES.VIDEO_ANSWER]: {
      component: (
        <Box display="flex" fontWeight="500"
m={2}>
          { /* MOVE TO CONSTANTS */
Here will be answer as
          video. ;)
        </Box>
      )
    }
  }

  const computedMediaUrl =
currentQuestion?.image || DEFAULT_IMAGE
  const tagRule =
questionData?.questionType !==
QUESTION_TYPES.WELCOME_SCREEN
  const url =
`url(${computedMediaUrl})`
  const questionTag =
currentQuestion.questionType ===
QUESTION_TYPES.ENDING
    ? endingListTitle ||
TEXTINGS.endingListTitle
    : `${questionListTitle ||
TEXTINGS.questionListTitle} ${
questionData?.order
}`
  const layoutType =
LAYOUT_TYPES[questionData?.layoutType]
  //rule for template to render
column with image, when layout type ===
left/right(small/big)
  const imageShowRule =
layoutType?.type !==
LAYOUT_TYPES.BETWEEN.type &&
layoutType?.type !==
LAYOUT_TYPES.FULL_SCREEN.type &&
layoutType?.type !==
LAYOUT_TYPES.DEFAULT.type
  const bgImage =
layoutType?.type ===
LAYOUT_TYPES.FULL_SCREEN.type &&
computedMediaUrl
  const
isConfigurationPopoverVisible = !(
currentQuestion.questionType ===
QUESTION_TYPES.ENDING
)
  const video =
questionData?.isVideoQuestion
  const configButtonPlacementRule =
[
    LAYOUT_TYPES.FULL_SCREEN.type,
    LAYOUT_TYPES.RIGHT_SIDE_BIG.type,
    LAYOUT_TYPES.RIGHT_SIDE_SMALL.type
  ].includes(layoutType?.type)

  useEffect(() => {
    setBrightnessValue(currentQuestion.imageBri
ghtness || 0)
  }, [currentQuestion,
setBrightnessValue])

  return (
    <ContentCard
      image={bgImage}

```



```

</ContentCard>
  )
}

QuestionForm.propTypes = {
  questionData: PropTypes.object,
  onQuestionTypeChange:
PropTypes.func,
  setShowPopover: PropTypes.func,
  showPopover: PropTypes.bool,
  setIsImageEditVisible:
PropTypes.func,
  isImageEditVisible: PropTypes.bool
}

export default QuestionForm

/* {layoutType?.type ===
LAYOUT_TYPES.FULL_SCREEN.type && (
  <Box position="absolute"
right="48px">
  <QuestionMediaPopover

brightnessValue={questionData?.brightnessVa
lue || brightnessValue}

setBrightnessValue={setBrightnessValue}

MediaModalButtonBackground={popoverImage}
  />
  </Box>
)}

{!!Object.keys(currentQuestion).length && (
  <CustomRow noGutters>
    <Col
  {...styles.questionCardColumnStyle}
  cw=[[12, 12, 10, 8]] px="0">
    <CustomCard bordered={false}>
      <Row noGutters v="center"
h="between">
        <Col cw="auto">
          {tagRule
&&<StyledTag>{questionTag}</StyledTag>
        </Col>

{isConfigurationPopoverVisible && (
  <Col cw="auto">
    <QuestionConfigurationPopover

questionData={questionData}

welcomeScreenShowRule={welcomeScreenShowRul
e}

questionsList={questionsList}

customQuestionTypes={customQuestionTypes}

onQuestionTypeChange={onQuestionTypeChange}
  />
  </Col>
  )}
  </Row>
  <Box>
    <Row noGutters h="between" mb={4}>
      <Col cw="12" mt={2}>
        <QuestionHeader

titlePlaceholder={

questionEditableTitleHint ||
TEXTINGS.questionEditableTitleHint
      }
      subtitlePlaceholder={
questionEditableSubtitleHint ||
TEXTINGS.questionEditableSubtitleHint
      }
    </Col>
  </Row>
  {layoutType?.type
=== LAYOUT_TYPES.BETWEEN.type &&
!questionData?.isVideoQuestion && (
  <Row noGutters>
    <Col cw="auto" flexDirection="end">
      <QuestionImageContainer
{...layoutType.imgSize}
      mb={4}
      image={computedMediaUrl}
      style={{
        filter:
`brightness(${
questionData?.brightnessValue + 100 ||
brightnessValue + 100
      }%)`
      }}>
        <Box display="flex"
justifyContent="flex-end" mr={4}>
          <QuestionMediaPopover

brightnessValue={brightnessValue}

setBrightnessValue={setBrightnessValue}

MediaModalButtonBackground={popoverImage}
  />
        </Box>
      </QuestionImageContainer>
    </Col>
  </Row>
  <Box>
    {cloneElement(
questionTypesMap[questionData?.questionType
].component,
    {
      question:
questionData
    }
  )}
  </Box>
  </Box>
  </CustomCard>
  </Col>
  {imageShowRule && (
  <StyledCol

order={layoutType?.imageOrder}

```

```

{...styles.sideImageColumnStyle}>
  <QuestionImageContainer
layoutType={layoutType?.type}
{...layoutType?.imgSize}
image={computedMediaUrl}
  style={{ filter:
`brightness(${brightnessValue + 100}%)` }}>
  <Row h="right">
    <Col cw="auto" mr={4}>
      <QuestionMediaPopover
brightnessValue={brightnessValue}
setBrightnessValue={setBrightnessValue}
MediaModalButtonBackground={popoverImage}
      />
    </Col>
  </Row>
</QuestionImageContainer>
</StyledCol>
  )}
</CustomRow>
  )} */

import { message } from 'antd'
import PropTypes from 'prop-types'
import React, { useMemo } from
'react'
import { Box } from '@qonsoll/react-
design'
import { DragableList } from
'../../../../components'
import useFunctions from
'../../../../hooks/useFunctions'
import { COLLECTIONS, QUESTION_TYPES
} from '../../../../constants'
import { QuestionSimpleView } from
'../../../../domains/Question/components'
import {
  DISPATCH_EVENTS,
  useCurrentQuestionContext,
  useCurrentQuestionContextDispatch
} from
'../../../../context/CurrentQuestion'

function QuestionsList(props) {
  const {
    data,
    setNewOrder,
    onItemClick,
    disableDelete,
    questionsData,
    endings
  } = props
  // [CUSTOM_HOOKS]
  const { setData, deleteData } =
useFunctions()
  const currentQuestion =
useCurrentQuestionContext()
  const currentQuestionDispatch =
useCurrentQuestionContextDispatch()

  // [CLEAN FUNCTIONS]
  const onUpdate =
(updatedQuestions) => {
    //check if questions contain
Welcome screen
    const containWelcomeScreen =
questionsData?.some(
      (q) => q?.questionType ===
QUESTION_TYPES.WELCOME_SCREEN
    )

    //update current questions order
after drag and drop
    updatedQuestions?.forEach((item,
index) =>
      setData(COLLECTIONS.QUESTIONS,
item?.id, {
        order: !!questionsData
          ? containWelcomeScreen
            ? questionsData?.length
              : questionsData?.length
                + item?.order
          : item?.order
        })
      .then(() => {
        item.id ===
currentQuestion.id &&
currentQuestionDispatch({
          type:
DISPATCH_EVENTS.UPDATE_CURRENT_QUESTION,
          payload: {
            ...item,
            order:
!!questionsData
              ?
containWelcomeScreen
                ?
questionsData?.length + index
              :
questionsData?.length + item?.order
                : item?.order
          })
        })
      .catch((e) =>
message.error(e.message))
    )
  }

  const handleDelete = async (e,
questionId) => {
    e.stopPropagation()

    //filter all questions without
deleted question
    let updatedData =
data?.filter((item) => item.id !==
questionId)

    //search deleted item index
    const deletedItemIndex =
data?.findIndex(
      (item) => item.id ===
questionId && item
    )

    //check if questions contain
Welcome screen
    const containWelcomeScreen =
(questionsData || updatedData).some(

```

```

        (q) => q.questionType ===
QUESTION_TYPES.WELCOME_SCREEN
    )

    //define questions length if
this list used for endings to define
endings correct order
    const questionsLength =
questionsData?.length

    //update order for all
questions/endings
    updatedData =
updatedData?.map((item, index) => {
    //define updated order for
questions/endings
    let updatedOrder =
!!questionsLength ? questionsLength + index
: index
    if (!containWelcomeScreen) {
        updatedOrder++
    }

    //update var that contain
questions/endings excluding
    item.order = updatedOrder
    setData(COLLECTIONS.QUESTIONS,
item?.id, item).catch((e) =>
        message.error(e.message)
    )
    //return item with updated
order
    return item
})

    //delete question/ending
    await
deleteData(COLLECTIONS.QUESTIONS,
questionId).catch((e) =>
    message.error(e.message)
)

    // remove logic jump conditions
for all connected questions
    await
updatedData?.forEach((item) => {
    let writeToDB = false
    const editedQuestionConfig =
item?.questionConfigurations?.map(
        (config) => {
            if
            (config?.redirectQuestion === questionId) {
                writeToDB = true
                return { ...config,
redirectQuestion: '' }
            } else return config
        }
    )
    writeToDB &&

    setData(COLLECTIONS.QUESTIONS, item.id, {
        questionConfigurations:
editedQuestionConfig
    }).catch((e) =>
        message.error(e.message)
    )

    //change current question
    currentQuestion.id ===
questionId &&

```

```

        currentQuestionDispatch({
            type:
DISPATCH_EVENTS.SET_CURRENT_QUESTION_TO_STA
TE,
            payload:
updatedData?.[deletedItemIndex] ||
updatedData?.[deletedItemIndex - 1] ||
        })
    }
}

    // [COMPUTED PROPERTIES]
    const dataSource = useMemo(
        () => (data ? data.sort((a, b)
=> a.order - b.order) : []),
        [data]
    )
    const filteredDataSource =
dataSource?.filter(
        (item) => item.questionType !==
QUESTION_TYPES.WELCOME_SCREEN
    )
    const sortable =
filteredDataSource?.length > 1

    return (
        <Box p={2}>
            {dataSource?.map(
                (item) =>
                    item.questionType ===
QUESTION_TYPES.WELCOME_SCREEN && (
                        <Box pl="24px" mb={2} key={item}>
                            <QuestionSimpleView
                                {...item}
                                endings={endings}
                                number="W"
                                hiddenDelete
                                action={handleDelete}
                                onClick={() =>
onItemClick(item)}
                            />
                        </Box>
                    )
                )}
            <DragableList
                itemLayout="horizontal"
                sortable={sortable}
                onUpdate={onUpdate}
                dataSource={filteredDataSource}
                setNewOrder={setNewOrder}
                renderItem={(item, index) =>
                    (
                        <QuestionSimpleView
                            {...item}
                            data={data}
                            endings={endings}
                            action={handleDelete}
                            number={index + 1}
                            onClick={() =>
onItemClick(item, index)}
                            disableDelete={disableDelete}
                        />
                    )
                }
            />
        </Box>
    )

```

```

    )
  }

  QuestionsList.propTypes = {
    data: PropTypes.array,
    action: PropTypes.func,
    setNewOrder: PropTypes.func,
    onItemClick: PropTypes.func
  }

  export default QuestionsList

  import PropTypes from 'prop-types'
  import React, { useEffect } from
'react'
  import { TEXTINGS } from
'../../../../../../constants'
  import { Box, NoData } from
'@qonsoll/react-design'
  import { Spinner } from
'../../../../../../src/components'
  import { useTranslation } from
'../../../../../../context/Translation'
  import { StyledTable } from
'../../../../../../domains/Response/components/ResponseTable/ResponseTable.style'

  const columns = [
    {
      render: (text, record) => (
<div
      style={{
        wordWrap: 'break-word',
        wordBreak: 'break-word',
        width: '50px'
      }}>
        {text}
</div>
      ),
      title: '#',
      dataIndex: 'order',
      key: 'order',
      sortOrder: 'descend',
      width: '50px'
    },
    {
      render: (text, record) => (
<div
      style={{
        wordWrap: 'break-word',
        wordBreak: 'break-word',
        width: '100%'
      }}>
        {text}
</div>
      ),
      title:
TEXTINGS.answerTableQuestionTitle,
      dataIndex: 'questionTitle',
      key: 'questionTitle'
    },
    {
      render: (text, record) => (
<div
      style={{
        wordWrap: 'break-word',
        wordBreak: 'break-word',
        width: '100%'
      }}>
        {text}
</div>
      ),
      title:
TEXTINGS.answerTableAnswerTitle,
      dataIndex: 'answer',
      key: 'answer'
    }
  ]

  function ResponseTable(props) {
    const { data, loading, isFormQuiz } = props
    const { answersNoSelectedUser } =
useTranslation()

    useEffect(() => {
      if (isFormQuiz)
        columns?.push({
          render: (text, record) => (
<div
          style={{
            wordWrap: 'break-
word',
            wordBreak: 'break-
word',
            width: '100%'
          }}>
            {text}
          </div>
          ),
          title:
TEXTINGS.answerTableScoreTitle,
          dataIndex: 'answerScore',
          key: 'answerScore'
        })
    }, [isFormQuiz])

    return (
<Box overflowX="hidden">
      {loading ? (
<Spinner width="100%" size="large"
      ) : (
<Box>
          {!!data?.length ? (
<StyledTable
            sortOrder
            columns={columns}
            dataSource={data}
            pagination={false}
          ) : (
<NoData
            description={
answersNoSelectedUser ||
TEXTINGS.answersNoSelectedUser
          }
        )
      )}
</Box>
    )}
</Box>
  )
}

ResponseTable.propTypes = {
  data: PropTypes.array,
  loading: PropTypes.bool,
  isFormQuiz: PropTypes.bool

```

}

export default ResponseTab

Додаток Б

**Ксерокопії наукових публікацій, виконаних при роботі над
дипломною роботою магістра**

Хмельницький національний університет

**ХІІІ ВСЕУКРАЇНСЬКА НАУКОВО-ПРАКТИЧНА
КОНФЕРЕНЦІЯ**



**АКТУАЛЬНІ ПРОБЛЕМИ
КОМП'ЮТЕРНИХ НАУК
*АПКН-2021***

15-16 жовтня 2021 р.

м. Хмельницький

УДК 004:37:001:62

Збірник наукових праць за матеріалами XIII Всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2021». Хмельницький – 2021. – 413с.

У збірнику наукових праць подані перспективні практичні розробки аспірантів, студентів та здобувачів в області сучасних інформаційних технологій. Розглянуто актуальні проблеми комп'ютерних наук, комп'ютерної інженерії, прикладної математики й інженерії програмного забезпечення, приведено ряд робіт по впровадженню інформаційних технологій у виробництво та управління. Висвітлено перспективні розробки сучасних систем пошуку, обробки й захисту інформації, медійних та комунікаційних системи.

УДК 004:37:001:62

Матеріали конференції відтворені з авторських оригіналів. При макетуванні можливі незначні зміни компоновки контенту авторських оригіналів.

Участь у конференції та складові всіх її етапів (розгляд праць, макетування, публікація збірника наукових праць та видача сертифікатів) є безкоштовними для всіх учасників. Оргкомітет конференції висловлює подяку учасникам конференції та сподівається на подальшу співпрацю.

З питань проведення конференції та подальшого обміну інформацією звертатись на e-mail конференції: apkt.khnu@gmail.com

Галкіна Р. І., Багрій Р. О., Скрипник Т. К. Застосування адаптивного підходу для реалізації системи опитувань та тестувань.....	306
Гринь С. С., Пивовар О. С., Таранчук А. А. Забезпечення прихованості дії та криптографічного захисту аналогових сигналів в хаотичній системі зв'язку.....	309
Данчук С. В., Багрій Р. О. Технологія автоматизованого отримання даних з веб-ресурсів для бізнес-аналітики.....	312
Длугунович Н. А. Інформаційна технологія фінансового моделювання для розвитку малого підприємництва.....	316
Дрозд А. І., Форкун Ю. В. Метод розподілу обчислювальних ресурсів для обробки розподілених потоків даних.....	319
Дудар О. В., Міхалевський В. Ц., Скрипник Т. К. Інформаційна система для забезпечення підтримки екологічної рівноваги.....	321
Єфімчук А. С., Скрипник Т. К., Мазурець О. В., Молчанова М. О. Автоматизований розподіл процесів при управлінні ІТ-проектами в складних критично-безпекових умовах.....	324
Житкевич В. В., Медведчук В. Ю. Метод відновлення пошкоджених растрових зображень.....	332
Заровний В. І., Скрипник Т. К. Методи шифрованої передачі даних між хмарними підпросторами.....	335
Кудрявцев В. В., Форкун Ю. В. Аналіз та застосування методів оптимізації швидкодії та відмовостійкості програмних продуктів.....	338
Курдибаха А. В., Мазурець О. В., Собко О. В., Молчанова М. О. Інформаційна технологія оцінювання діяльності сімейного лікаря за даними прийомів.....	340
Лаврентій А. А., Петровський С. С. Метод оцінювання наповненості дистанційних курсів предметів у школі.....	349
Левченко Т. В., Блажук В. Д., Молчанова М. О., Собко О. В. Метод оптимізації транспортних перевезень засобами біологічної метаевристики.....	352

АКТУАЛЬНІ ПРОБЛЕМИ КОМП'ЮТЕРНИХ НАУК - 2021

XIII Всеукраїнська науково-практична конференція

Метою конференції є висвітлення актуальних проблем комп'ютерних наук, інформатики та інформаційних технологій.

СЕКЦІЇ КОНФЕРЕНЦІЇ:

1. Комп'ютерні науки та прикладні інформаційні технології.
2. Комп'ютерна інженерія та системи захисту інформації.
3. Математичне моделювання та інженерія програмного забезпечення
4. Телерадіокомунікації, медійні та комунікаційні системи.
5. Проблеми впровадження інформаційних технологій у виробництво та управління.

Робочі мови конференції: українська, англійська

ОРГКОМІТЕТ:

СИНЮК О. М. голова оргкомітету, проректор Хмельницького національного університету з наукової роботи, доктор технічних наук, професор

САВЕНКО О. С. заступник голови оргкомітету, декан факультету Інформаційних технологій ХНУ, доктор технічних наук, професор

БАРМАК О. В. заступник голови оргкомітету, завідувач кафедри Комп'ютерних наук ХНУ, доктор технічних наук, професор

ГОВОРУЩЕНКО Т. О. завідувач кафедри Комп'ютерної інженерії та інформаційних систем ХНУ, доктор технічних наук, професор

ВИСОЦЬКА О. В. доктор технічних наук, завідувач кафедри Радіоелектронних та біомедичних комп'ютеризованих засобів і технологій Національного аерокосмічного університету ім. М. Є. Жуковського «Харківський авіаційний інститут», професор

ЛАВРОВ Є. А. доктор технічних наук, професор (Сумський державний університет)

ТІМОФЄЄВА Л. В. відповідальна за студентську науково-дослідну роботу ХНУ

МАЗУРЕЦЬ О. В. секретар конференції, к.т.н., доцент кафедри Комп'ютерних наук ХНУ

МОЛЧАНОВА М. О. секретар конференції, викладач кафедри Комп'ютерних наук ХНУ

КОНТАКТНА ІНФОРМАЦІЯ:

е-mail для листування: apkt.khnu@gmail.com

УДК 004

Галкіна Р. І., Багрій Р. О., Скрипник Т. К.

*Хмельницький національний університет***ЗАСТОСУВАННЯ АДАПТИВНОГО ПІДХОДУ ДЛЯ РЕАЛІЗАЦІЇ СИСТЕМИ ОПИТУВАНЬ ТА ТЕСТУВАНЬ**

У статті розглянуто основні положення традиційного тестування та передумови використання адаптивного тестування. Класична тестування не завжди може вирішити поставлені вимоги сучасним рівнем розвитку систем контролю знань. Тому у подібних випадках використовується адаптивний підхід тестування. Запропонована інформаційна система для проведення опитувань та тестувань, що дала можливість зменшити час, витрачений на проведення тестування, отримання більш точних результатів тестування та спрощення процесу перевірки результатів.

The article considers the main provisions of traditional testing and the prerequisites for the use of adaptive testing. Classical testing cannot always solve the requirements of the current level of development of knowledge control systems. Therefore, in such cases, an adaptive testing approach is used. An information system for conducting surveys and tests is proposed, which has made it possible to reduce the time spent on testing, obtain more accurate test results and simplify the process of verifying results.

На сьогодні автоматизація та комп'ютеризація торкаються майже всіх процесів, що оточують людину. В тому числі й процес збору інформації, а також оцінки якості її отримання. Ці зміни спричинені постійним вдосконаленням систем, що пов'язані з контролем процесу поширення та засвоєння знань.

Опитування – це метод збору соціологічної інформації про досліджуваний об'єкт під час безпосереднього (усне опитування, інтерв'ю) або опосередкованого (письмове опитування, анкетування) спілкування того хто опитує з респондентом [1].

Тестування – система формалізованих завдань, призначених для встановлення освітнього (кваліфікаційного) рівня особи. Педагогічне тестування – форма оцінювання знань учнів, студентів (абітурієнтів), основана на застосуванні педагогічних тестів.

Традиційний тест являє собою стандартизований метод оцінки рівня знань і структури підготовленості людини. При проведенні такого тестування всі відповідають на одні і ті ж завдання протягом однакового часу, в однакових умовах і з однаковими правилами оцінювання відповідей. Одне з головних питань теорії тестів – питання підбору оптимального за деякими критеріями тесту [2]. Кожен тест

може мати відмінності від інших тестів за безліччю параметрів, у тому числі кількістю питань та їх складністю.

Практика показує, що оптимальним рішенням є відносно невеликий тест, проте добре відображає поставлену задачу. Поняття "довжина тесту" введено на початку XX століття Ч. Спірманом і означає кількість завдань у тесті. Чим довший тест, тим більше в ньому завдань. Від кількості завдань деяким чином залежить точність виміру знань [2].

Динамічна зміна кількості питань або їх складності є доцільним рішенням проблеми довгих тестів.

Адаптивне тестування – різновид тестування, при якому порядок представлення запитань (або складність) залежить від відповідей того, хто тестується, на попередні питання [3].

Адаптивне тестування можна організувати по-різному. Один із варіантів – розділити питання тесту за складністю. Якщо педагогічний тест визначити коротко як систему завдань зростаючої складності, то стане зрозуміло, що складність завдань є найважливішим тестоутворюючим показником [4]. Показник складності тесту і тестових завдань є змістовним і формальним одночасно.

Змістовним – тому що на складність впливають як рівень знань тих, що тестують, ефективна організація проведення тесту, так і сам зміст тесту. Формальним – тому що виникає процес протистояння завданням тесту (у цьому випадку існує лише два типи результату – перемога при правильній відповіді, чи поразка при неправильній). Оцінка результату такої боротьби залежить від співвідношення рівня знання до рівня складності завдання [4].

Метою дослідження є розробка інформаційної системи для проходження опитувань та тестів, використовуючи адаптивний підхід. Важливо зауважити, що на даний момент у відкритому доступі майже відсутні системи, що використовують адаптивний відхід – абсолютна більшість базуються на класичній теорії тестів, тому розробка є актуальною.

На рисунку 1 зображено принцип роботи інформаційної системи:



Рисунок 1 – Принцип роботи інформаційної системи

Для отримання результату спочатку необхідно сформулювати список запитань з масивом відповідей на них. Далі встановлюються логічні зв'язки між варіантами

відповідей та індексами наступних питань. Згідно з даними відповідями на місце наступного питання буде обрано більш релевантне в залежності від заданих умов. Наступний крок – оцінка рівня складності питання, що досягається шляхом надання певної кількості балів за кожен варіант відповіді на питання. Як результат отримується вибірка пар «питання-відповідь» та інформація про набрані бали, що можуть використовуватись для загальної оцінки опитуваних, аналізу релевантності питань до обраної теми, а також формування статистики.

Даний різновид тестування дозволяє підлаштовуватись під індивідуальні можливості кожного, хто проходить тестування, завдяки виключенню занадто важких, або занадто легких завдань, що одночасно впливає на час проведення тестування. Таким чином, при кожній вірній відповіді наступне питання буде більш складним, а при неправильних відповідях складність тесту буде зменшуватись. Також на основі відповідей можна визначити наступне питання із тієї теми, що менш відома тому, хто проходить тест. Це дозволить зменшити кількість помилок тестування, що іноді виникають при неправильному трактуванні теми опитуваним.

Отже запропонована інформаційна система для проведення опитувань та тестувань дозволить проводити оптимізовані тести для оцінки знань. Основними перевагами системи є:

- зменшення часу, витраченого на проведення тестування, завдяки адаптивному підбору наступного питання;
- отримання більш точних результатів проведення опитування;
- спрощення процесу перевірки результатів завдяки відсутності надлишкових відповідей.

Перелік посилань

1. Словник української мови Б. Грінченка [Електронний ресурс]. – Режим доступу: <http://hrinchenko.com/slovar/znachenie-slova/35331-opyt.html>
2. Аванесов В.С. Теория и методика педагогических измерений (материалы публикаций). – М.: ЦТ и МКО УГТУ-УПИ, 2005. – 98 с.
3. Федорук П.І. Адаптивні тести: статистичні методи аналізу результатів тестового контролю знань // Математичні машини і системи. – 2007. – № 3,4. – С. 122-138.
4. Lord F.M. Application of Item Response Theory to Practical Testing Problems. Hillsdale N-J. Lawrence Erlbaum Ass., Publ., 1980. – 266 p.

Кваліфікаційна робота магістра



Застосування адаптивного підходу для реалізації системи опитувань та тестувань

Виконала:
Галкіна Руслана Ігорівна

Керівник:
Багрій Руслан Олександрович



Актуальність теми

Кваліфікаційна робота присвячена реалізації адаптивного підходу для реалізації системи опитувань та тестувань, що, завдяки використанню сучасного підходу, дозволить вираховувати цінність кожного наступного питання, яка в свою чергу дає можливість ефективно використовувати час як опитуваного, так і перевіряючого.



Мета роботи

Мета магістерської роботи полягає у реалізації інформаційної системи для проведення опитувань та тестувань з використанням адаптивного підходу.



Задачі дослідження

- дослідити існуючі методи та підходи для проведення опитувань та тестувань;
- вдосконалити процес проведення опитувань та тестувань шляхом динамічної зміни наступного питання та їх загальної кількості;
- спроектувати інформаційну систему для проведення опитувань та тестувань з використанням адаптивного підходу;
- забезпечити отримання результатів опитування чи тестування у формі, придатної для подальшого проведення аналізу;
- здійснити порівняння підходів при проведенні класичного та адаптивного тестування.

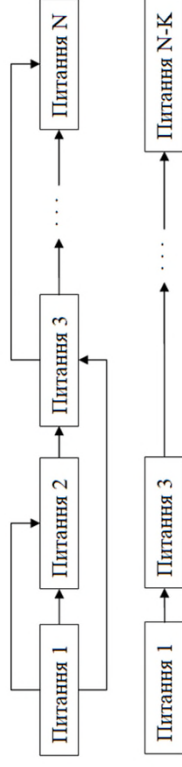


Порівняння підходів проведення опитувань та тестувань

Класичний (традиційний) підхід



Адаптивний підхід





Принцип роботи інформаційної системи

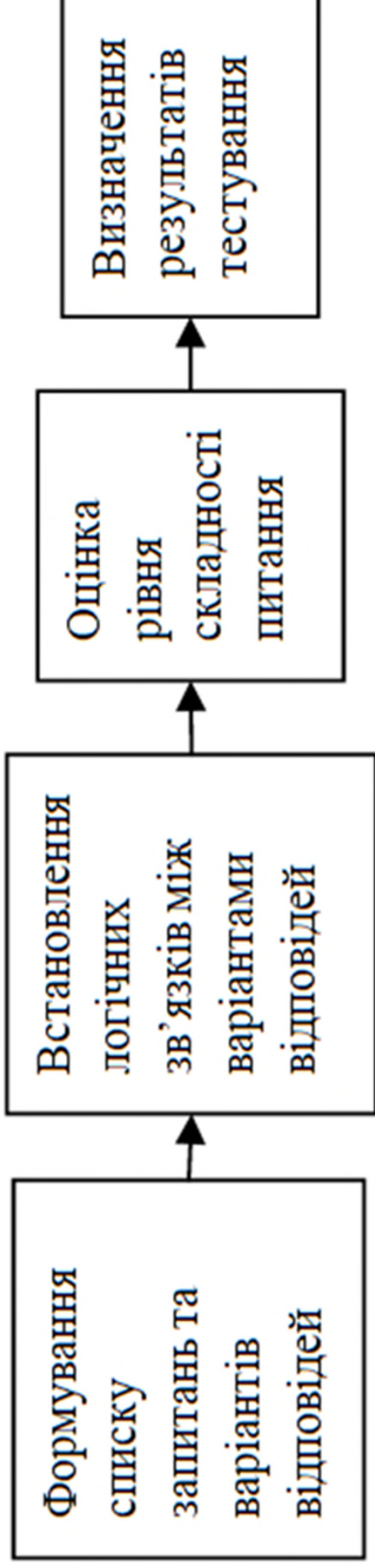
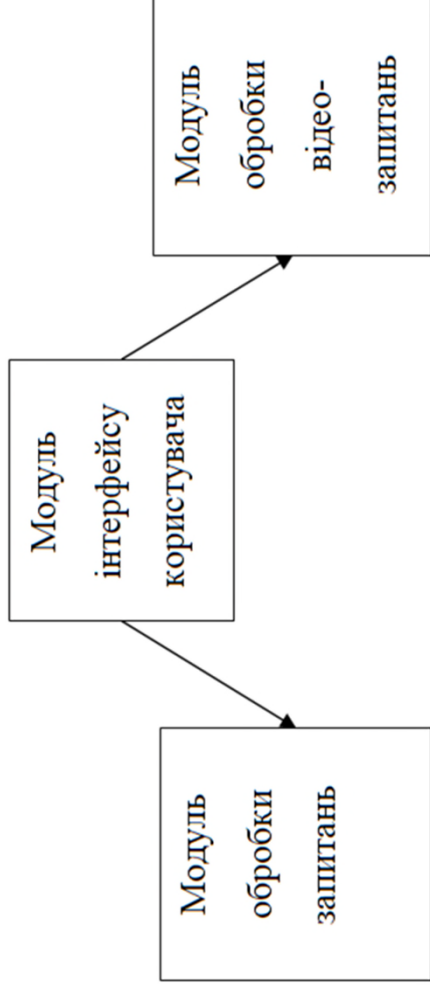


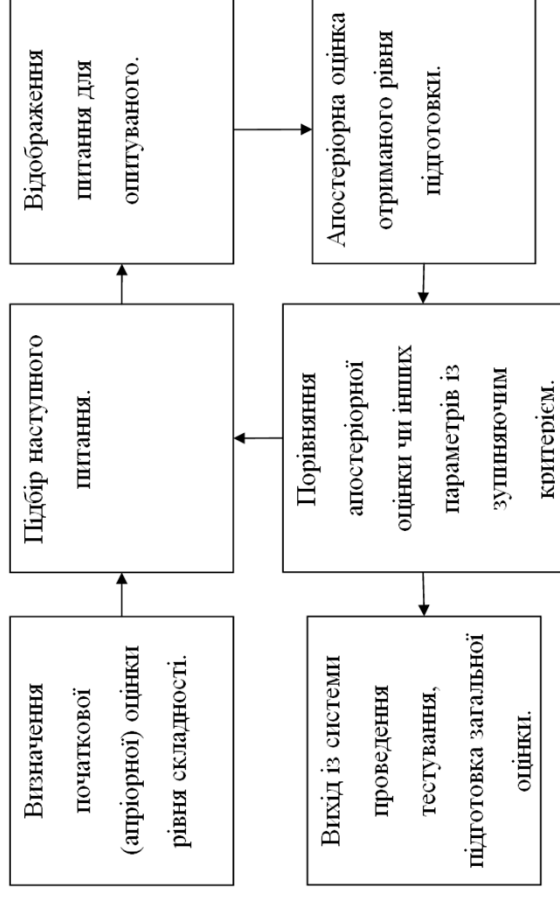


Схема взаємодії модулів інтерфейсу

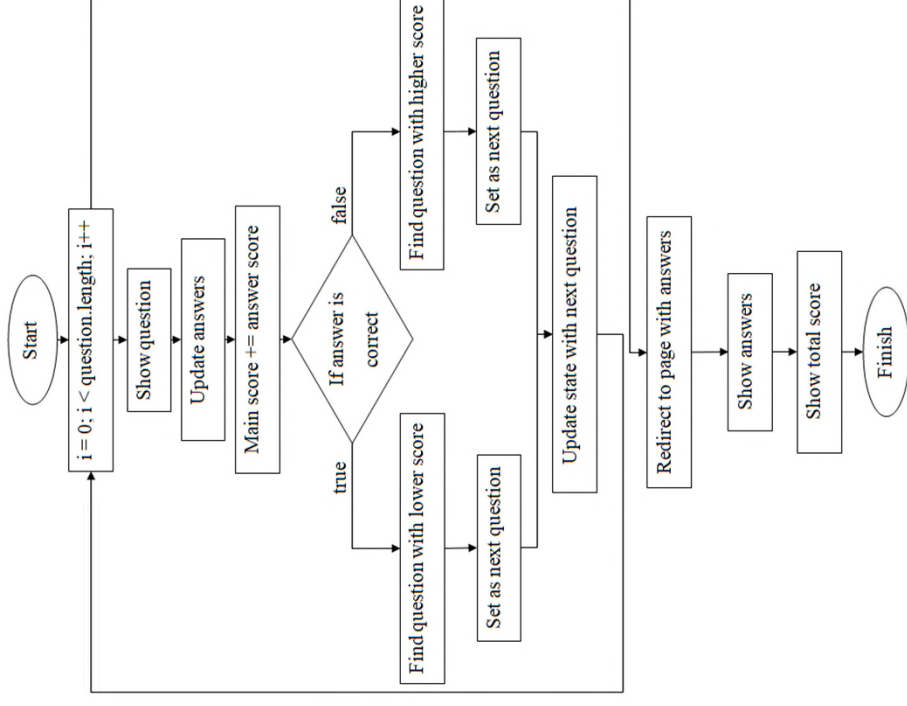




Алгоритм багатокрокового адаптивного тестування

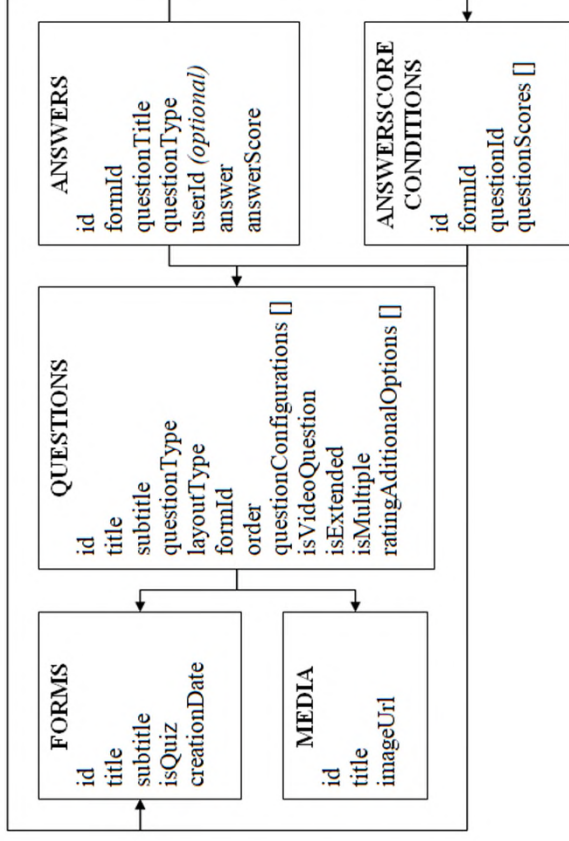


Блок-схема алгоритму оброблення наступного питання та загального балу



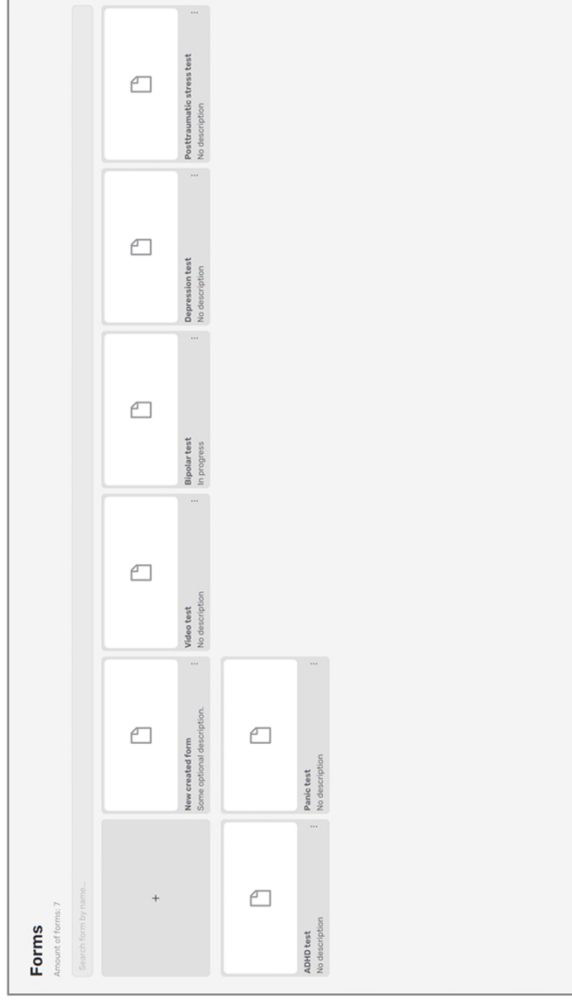


Структура бази даних



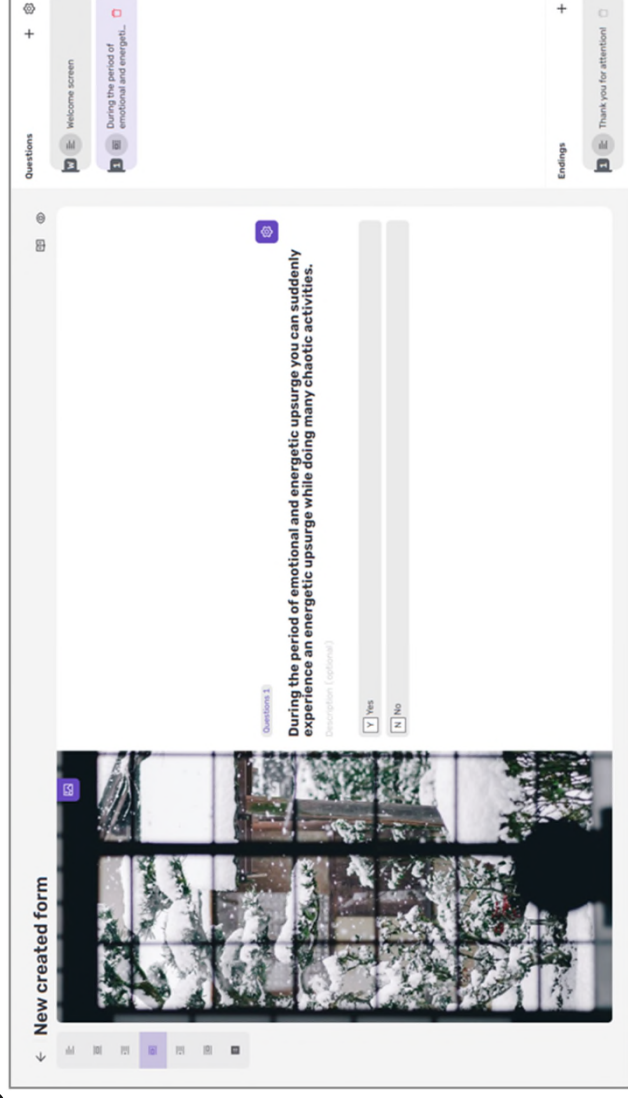


Інформаційна система для проведення опитувань та тестувань





Інформаційна система для проведення опитувань та тестувань





Інформаційна система для проведення опитувань та тестувань

Logic Reset logic

[Logic jumps](#) [Endings](#) [Answer score configurations](#)

1 Have you ever experienced emotional and energetic upsurge?
A Yes Go to the next question
B No Go to the next question

2 During the period of emotional and energetic upsurge you can suddenly experience an energetic upsurge while doing many chaotic activities.
A Yes Go to the next question
B No Go to the next question

3 During the period of emotional and energetic upsurge it is typical for you to unexpectedly experience incredible excitement, an overly elevated, euphoric mood.
A Yes Go to the next question
B No Go to the next question

[Close](#)



Інформаційна система для проведення опитувань та тестувань



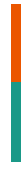
Answers

#	Question	Answer	Score
1	During the period of emotional and energetic upsurge you can suddenly experience an energetic upsurge while doing many chaotic activities.	Yes	10
2	During the period of emotional and energetic upsurge it is typical for you to unexpectedly experience incredible excitement, an overly elevated, euphoric mood.	No	0
3	Often times, periods of emotional uplift are followed by states of sadness, devastation, and anxiety.	Yes	10
4	In times of sadness and anxiety, you feel utter hopelessness and desolation.	Yes	10
5	Upload picture that describes your current state.	1.png	

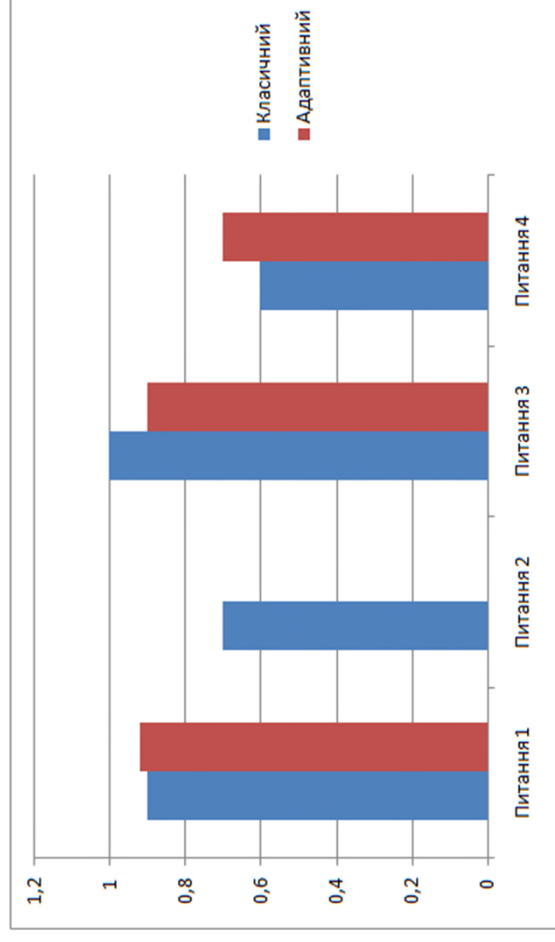


Результати апробації

Номер питання	Класичний підхід		Адаптивний підхід	
	Відповідь	Затрачений час	Відповідь	Затрачений час
1	Ні	< 1хв.	Ні	< 1 хв.
2	Ні	< 1 хв.	-	0
3	Так	1.05 хв.	Так	1 хв.
4	4	< 1 хв.	4	< 1 хв.
		Загалом: прибіл. 4хв.	Загалом: прибіл. 3хв.	



Результати апробації





Результати апробації

Підхід	Загальна кількість питань	Кількість питань, на які відповіли	Затрачений час
Класичний підхід, спроба 1	24	24	28 хв.
Адаптивний підхід, спроба 1	24	21	22 хв.
Адаптивний підхід, спроба 2	24	19	19 хв.
Адаптивний підхід, спроба 3	24	21	21 хв.
Адаптивний підхід, спроба 4	24	16	18 хв.



Висновки

В результаті проведеної роботи були отримані такі положення наукової новизни:

- удосконалено метод проведення тестувань та опитувань з використанням адаптивного підходу, що дало можливість зменшити кількість запитань в порівнянні з традиційними підходами без втрати точності отриманих результатів;
- розроблено інформаційну систему для проведення опитувань та тестувань з застосуванням адаптивного підходу, що дало можливість скоротити час проходження, а також усунути надлишкові варіанти відповідей до запитань;
- досліджено практичну ефективність використання адаптивного підходу під час проведення тестувань, що дозволило оцінити витрачений час на проведення тестування та якість отриманих результатів;

Дякую за увагу.



Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 2.0%

Словари проверки: en_US, ru_RU, ua_UA. **Ошибок в документах: 9%**

ID: 97516 Название: Застосування адаптивного підходу для реалізації системи опитувань та тестувань Добавлено в БД: 2021-11-29 Авторы: Р.І. Галкіна Руководители: Р.О. Багрій Консультанты: Оponentы:	Документ		Суммарное совпадение по Базе Данных	
	Символы	Лексемы	Символы	Лексемы
	52218	463	3844 (7%)	39 (8%)

Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы

Ім'я користувача:
Кафедра КН

ID перевірки:
1009603155

Дата перевірки:
08.12.2021 17:41:46 EET

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
08.12.2021 17:46:23 EET

ID користувача:
100005671

Назва документа: Галкіна_Записка_до_магістерської_v7 Lite

Кількість сторінок: 82 Кількість слів: 11524 Кількість символів: 96241 Розмір файлу: 5.00 MB ID файлу: 1009606928

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

14.5% Схожість

Найбільша схожість: 3.14% з Інтернет-джерелом (<https://uk.wikipedia.org/wiki/Firebase>)

10.4% Джерела з Інтернету 60 Сторінка 84

4.82% Джерела з Бібліотеки 94 Сторінка 85

0.18% Цитат

Цитати 5 Сторінка 86

Посилання 1 Сторінка 86

58.6% Вилучень

Деякі джерела вилучено автоматично (фільтри вилучення: кількість знайдених слів є меншою за 8 слів та 0%)

Немає вилучених Інтернет-джерел

58.6% Вилученого тексту з Бібліотеки 1 Сторінка 86

Модифікації

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

Замінені символи 4

Підозріле форматування 17 сторінок

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ

КАФЕДРИ КОМП'ЮТЕРНИХ НАУК

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ МАГІСТРА ДО ЗАХИСТУ ЗА
РЕЗУЛЬТАТАМИ АНАЛІЗУ ЗВІТУ ПОДІБНОСТІ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Застосування адаптивного підходу для реалізації системи опитувань та тестувань

Автор: Галкіна Руслана Ігорівна

Спеціальність: 122 – Комп'ютерні науки

Освітня програма: освітньо-професійна

Науковий керівник: к.т.н., доц. Багрій Р.О.

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:


- 1) за програмою Anti-Plagiarism виявлені 2% є фрагментарними – містять поширені конструкції, загальновідомі терміни, скорочення та визначення..
- 2) За програмою UNICHECK виявлені 14.5% що є запозиченнями, які розміщені в розділах аналізу існуючих технологій та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи.


Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 2% і 14,5% відповідно, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.


Керівник роботи

Гарант ОП

Завідувач кафедри КН







Руслан Багрій

Руслан Багрій

Олександр Бармак



ВІДГУК ОПОНЕНТА

на кваліфікаційну роботу магістра

гр. КНМ-20-1 Галкіної Руслани Ігорівни за темою: Застосування адаптивного підходу для реалізації системи опитувань та тестувань

1. Актуальність обраної теми

Кваліфікаційна робота присвячена застосуванню адаптивного підходу для реалізації системи опитувань та тестувань, що дозволяє зменшити час проходження тестів за рахунок виключення надлишкових питань. Таким чином, тема кваліфікаційної роботи магістра є достатньо актуальною.

2. Відповідність роботи предметній області спеціальності 122 Комп'ютерні науки та загальним вимогам до наукових робіт

Об'єктом дослідження кваліфікаційної роботи магістра є проведення тестування та опитувань у комп'ютерних системах, а предметом дослідження – методи проведення тестувань у комп'ютерних системах та інформаційна система для проведення опитувань та тестувань.

Кваліфікаційна робота виконана з використанням сучасних методів, алгоритмів та технологій, що відображається у використанні адаптивного підходу для проведення опитувань та тестувань.

Таким чином, тема кваліфікаційної роботи відповідає предметній області спеціальності 122 Комп'ютерні науки та вимогам до кваліфікаційної роботи магістра.

3. Повнота розкриття мети та завдань дослідження

Мета магістерської роботи полягає у реалізації інформаційної системи для проведення опитувань та тестувань з використанням адаптивного підходу.

Для досягнення поставленої мети визначені наступні задачі:

- дослідити існуючі методи та підходи для проведення опитувань та тестувань;
- вдосконалити процес проведення опитувань та тестувань шляхом динамічної зміни наступного питання та їх загальної кількості;
- спроектувати інформаційну систему для проведення опитувань та тестувань з використанням адаптивного підходу;
- забезпечити отримання результатів опитування чи тестування у формі, придатної для подальшого проведення аналізу;

– здійснити порівняння підходів при проведенні класичного та адаптивного тестування.

Наведені задачі дослідження повністю розкривають мету роботи.

4. Наявність наукової новизни

В результаті проведеної роботи були отримані такі результати:

1. Вдосконалено метод проведення тестувань та опитувань з використанням адаптивного підходу, що дало можливість зменшити кількість запитань в порівнянні з традиційними підходами без втрати точності отриманих результатів.

2. Розроблено інформаційну систему для проведення опитувань та тестувань з застосуванням адаптивного підходу, що дало можливість скоротити час проходження, а також усунути надлишкові варіанти відповідей до запитань.

3. Досліджено практичну ефективність використання адаптивного підходу під час проведення тестувань, що дозволило оцінити затрачений час на проведення тестування та якість отриманих результатів.

Основні наукові й практичні результати кваліфікаційної роботи магістра доповідались на XIII Всеукраїнській науково-практичній конференції «Актуальні проблеми комп'ютерних наук АПКН-2021» (15-16 жовтня 2021 року) у доповіді на тему «Застосування адаптивного підходу для реалізації системи опитувань та тестувань» та за темою роботи автором виконано наукову публікацію.

Таким чином, у кваліфікаційній роботі магістра представлені результати містять наукову новизну, мають практичну цінність та достатній рівень апробації.

5. Зміст кожного розділу роботи

У першому розділі наведено аналіз сучасного стану проблеми автоматизованого проведення опитувань та тестувань, який включає аналіз предметної області, аналіз методів та підходів до проведення опитувань та тестувань, аналіз існуючих наразі розроблених додатків. Завершується розділ постановкою задачі та висновками.

У другому розділі наведено інформаційну систему для проведення опитувань та тестувань. Розглянуто основні методи проведення опитувань та запропоновано до аналізу два основних – класичний та адаптивний метод, як вдосконалення класичного методу. Проаналізовано ефективність двох методів та отримано результати.

Третій розділ присвячений проектуванню інформаційної системи для проведення опитувань та тестувань. Було наведено структуру модулів системи, їх взаємозв'язок, сформульовано алгоритм розрахунку наступного питання та загальної оцінки та спроектовано структуру бази даних.

В четвертому розділі виконано реалізацію інформаційної системи для проведення опитувань та тестувань. Було обґрунтовано вибір технологій, що використовувались під час

розробки, проведено апробацію інформаційної системи та представлені результати застосування адаптивного підходу під час проходження форми.

6. Ступінь розкриття теми роботи

В кваліфікаційній роботі магістра основну увагу приділено проведенню тестувань з використанням адаптивного підходу та відсутній опис особливостей застосування адаптивного підходу, що виникають під час опитувань. Також не досліджена практична ефективність використання адаптивного підходу під час проведення опитувань.

7. Якість оформлення кваліфікаційної роботи

Кваліфікаційна робота магістра виконана із дотриманням усіх вимог щодо оформлення. Викладення матеріалу є логічним, послідовним та стилістично грамотним.

8. Недоліки кваліфікаційної роботи

В роботі не досліджена практична ефективність інформаційної системи з використанням адаптивного підходу при проведенні опитувань, також є несуттєві недоліки в оформленні.

9. Загальний висновок (допускається чи не допускається до захисту), якої оцінки заслуговує кваліфікаційна робота.

Роботі виконана студентом в достатньому обсязі, наявна наукова новизна та виконані усі поставлені завдання. Кваліфікаційна робота магістра Галкіної Руслани Ігорівни рекомендується до захисту, але враховуючи допущені помилки, заслуговує на оцінку “добре”.

Опонент _____ к.т.н., доц. Андрій Нічепорук





ВІДГУК НАУКОВОГО КЕРІВНИКА

на кваліфікаційну роботу магістра

гр. КНМ-20-1 Галкіної Руслани Ігорівни за темою: Застосування адаптивного підходу для реалізації системи опитувань та тестувань

1. Актуальність теми

Актуальність теми достатньо обґрунтована, оскільки застосування адаптивного підходу для реалізації системи опитувань та тестувань дозволяє проводити тестування в менші проміжки часу, спрощує процес перевірки шляхом виключення надлишкових відповідей та дає інформацію щодо вдосконалення теми проведення тестування.

2. Відповідність роботи предметній області спеціальності 122 Комп'ютерні науки та загальним вимогам до наукових робіт

Теми кваліфікаційної роботи "Застосування адаптивного підходу для реалізації системи опитувань та тестувань" відповідає предметній області спеціальності 122 Комп'ютерні науки та вимогам до кваліфікаційної роботи магістра, оскільки об'єктом дослідження є процес проведення тестування та опитувань у комп'ютерних системах, предметом дослідження – методи проведення тестувань у комп'ютерних системах та інформаційна система для проведення опитувань та тестувань.

3. Професійні та особистісні якості магістранта

Галкіна Р. І. під час роботи над кваліфікаційною роботою магістра продемонструвала достатній рівень знань та умінь за спеціальністю "Комп'ютерні науки".

4. Ступінь самостійності під час виконання кваліфікаційної роботи

Робота виконана самостійно, академічного плагіату не виявлено, стосовно всіх запозичень наведено відповідні посилання на джерела.

5. Наукова новизна та оригінальність запропонованих підходів

Отримані такі результати: досліджено практичну ефективність використання адаптивного підходу під час проведення тестувань, що дало змогу скоротити час проведення тестування; розроблено інформаційну систему для проведення опитувань та тестувань з застосуванням адаптивного підходу. Отримані результати оприлюднені на XIII всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2021» 15 листопада 2021 р., м. Хмельницький, Україна, доповідь на тему «Застосування адаптивного підходу для реалізації системи опитувань та тестувань».

6. Ступінь оволодіння методами дослідження

Студент Галкіна Р. І. має достатньо високий ступінь володіння методами дослідження, що були використанні у роботі.

7. Повнота та якість розкриття теми роботи

Мета роботи повністю розкрита, отримані результати підтверджують достовірність наукових положень.

8. Логічність, послідовність, аргументованість, літературна грамотність викладу матеріалу

Викладення матеріалу логічне, послідовне та аргументоване. Мова і стиль викладення кваліфікаційної роботи магістра відповідають стандартам, що забезпечує доступність сприймання матеріалу і відповідає вимогам до сучасних наукових робіт.

9. Можливість практичного застосування кваліфікаційної роботи, окремих її частин

Може мати практичне значення при проведенні опитувань та тестувань у різних сферах суспільної та освітньої діяльності.

10. Висновок про можливість допуску кваліфікаційної роботи до захисту, на яку оцінку заслуговує робота

Вважаю, що кваліфікаційна робота студентки Галкіної Руслани Ігорівни може бути рекомендована до захисту та заслуговує на оцінку "добре".

Науковий керівник _____



к.т.н., доц. Руслан Багрій