МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
УМАНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ САДІВНИЦТВА
Факультет економіки і підприємництва
Кафедра інформаційних технологій

Допущено до захисту
"_____" _____ 2024 р.
Зав. кафедри, к.т.н., доцент
_____ **Ліщук Р.І.**

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

РОЗРОБКА ДОДАТКУ ДЛЯ ІНТЕРАКТИВНОГО ТЕСТУВАННЯ ЗНАНЬ

Виконала – студентка 2 курсу, групи 21м-з-кн
освітньої програми "Комп'ютерні науки"
спеціальності 122 Комп'ютерні науки
Маньковська Вікторія Сергіївна
Науковий керівник – д.т.н., професор
Кучерук Володимир Юрійович

Умань – 2024

РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Тестування як форма контролю та діагностики знань

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

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

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

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

Тестування – це метод визначення рівня знань і вмінь учнів за допомогою спеціальних тестових завдань, які подаються як правило, у формі запитань або задач. За всю історію використання тестів як інструменту діагностики навчальних досягнень дітей ставлення до них було неоднозначним.

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

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

- можливість використання як засобу контролю всіх видів, а саме базового та попереднього, поточного та тематичного, етапного та прохідного, підсумкового та екзаменаційного контролю, а також самоконтролю;
- можливість детально перевірити рівень засвоєння кожного змістовного модуля дисципліни;
- наявність чіткої та однозначної відповіді і оцінювання на основі цінника;
- економія навчального часу при постійному контролі знань та об’єктивності оцінки результатів дослідження;
- зведення до мінімуму емоційного впливу вчителя на учня [4].

Крім цього, традиційно до переваг тестування відносяться:

- індивідуальний характер контролю, можливість контролювати роботу кожного студента, його особисту навчальну діяльність, а також можливість отримувати результати успішності групи, що допоможе викладачеві оцінити ефективність методики. і прийоми, що використовуються в класі;
- можливість регулярного та систематичного контролю контрольних робіт на всіх етапах навчального процесу;
- можливість поєднання із іншими звичайними формами педагогічного контролю;
- об’єктивність тестового контролю знань, що має виключати суб’єктивні

оцінки та висновки викладача;

- можливість проведення комп'ютерного тестування в локальній мережі та паперовому варіанті;
- врахування індивідуальних особливостей учнів, що вимагає використання різних методик розробки тестів і тестових завдань залежно від рівня їх мовленнєвої компетенції відповідно до цих особливостей;
- висока змістовна точність тестового контролю, що полягає у включенні до тестових завдань дидактичних одиниць навчальної програми [5].

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

1.2 Сучасні вимоги, які ставляться до тестування знань

Так як один з найефективніших засобів контролю знань є тестування через те, що воно має безліч переваг над іншими видами контролю знань, а саме основною перевагою є – оцінка результатів при тестуванні більш об’єктивна та зменшення упередженості при оцінюванні, то до тестування ставиться безліч

вимог. Основні з них: логічна форма висловлювання, стислість, правильність форми.

Логічна форма висловлювання є універсальною формою для чіткого висловлення ідеї людиною, яка має змогу виявити знання з вивченої дисципліни. У більшості випадках ця форма замінює питання. Навчальні запитання породжують повні й неповні, правильні й неправильні відповіді, різні за формою, змістом і структурою, внаслідок чого оцінка таких відповідей вимагає участі вчителя й супроводжується певним суб'єктивізмом. Питання і відповіді іноді настільки розпливчасті і нудні, що для того, щоб дізнатися їхню правдивість, потрібен великий розум, а технологічний метод тестування полягає в тому, щоб чітко і швидко диференціювати відповіді. У цьому сенсі традиційні запитання та відповіді не є технологічними; їх не рекомендується включати в тестування.

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

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

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

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

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

У другій формі завдання сформульоване таким чином, що готової відповіді немає, кожен має перевірити відповідь самостійно під час контрольної роботи, у визначеному місці. Ці завдання можна назвати відкритими, так як після виконання завдання з конкретною відповіддю твердження є істинним або хибним. Задачі, в яких необхідно поєднати елементи однієї множини з елементами іншої, можна

назвати задачами на відповідність. Це є третьою формою. Наприклад, у випадках, якщо необхідно визначити правильний порядок обчислень, дій, кроків, операцій, понять у визначеннях, для визначення правильного порядку використовуються завдання. Це четверта форма квестів. Кожна з цих форм дозволяє перевірити певні види знань, а також потребує відповідних контрольних матеріалів. Вибір форм залежить від мети перевірки та змісту тексту, від технічних можливостей і рівня підготовки вчителя з теорії та методики перевірки контролю знань. Стислість завдань у тестовій формі зумовлена перевагами логічної форми висловлювання, завдяки якій завдання завжди є коротшими за завдання та запитання. Точність має забезпечуватись ретельним підбором слів, символів і графіків, що дозволяє мінімальними засобами досягти наочності змісту завдання. Виключаються повтори, незрозумілі, рідко вживані та незнайомі учням, іншомовні слова, що ускладнюють розуміння змісту. Добре, якщо завдання не містять більше одного підрядного речення. Для короткого завдання краще поставити його [7].

1.3 Інтерактивні моделі навчання та оцінювання знань

Розглядають такі моделі навчання: пасивна, активна та інтерактивна [8]. Роль учня в пасивній моделі навчання полягає в тому, щоб бути суб’єктом навчання, тобто він повинен лише слухати і дивитися. Цю модель навчання ще називають «глечик і чашка». На початку ХХ століття навчання в більшості навчальних закладів відбувалося за технологією «глечик-чашка».

Викладач виступає як фахівець у певній галузі, він повинен надати своїм учням оброблену інформацію, яку вони сприймають, запам’ятовують і вміють відтворювати. Під час іспиту перевіряється пройдений матеріал. Як зазначає А. Вірасінгхе, така модель навчання неефективна і має багато недоліків. По- перше, студенти, які навчалися за такою технологією і навчені лише запам’ятовувати інформацію, не готові вирішувати складні та незвичайні завдання на роботі. Пасивна роль не дозволяє креативити і проявляти творчі здібності. По-друге, учні бояться помилитися, що знижує рівень самоаналізу та самовиховання, які є

важливими аспектами розвитку особистості. Такої ж думки дотримується Дж. Скрівенер, який вважає такі методи неефективними, оскільки студенти повинні мати можливість спілкуватися один з одним, виконувати певні завдання, отримувати оцінки та зворотний зв'язок щодо покращення своєї роботи. Однак Д. Керролл стверджує, що метод «розмова і крейда» або «глечик і чашка» стає досить ефективним способом навчання, якщо в класі використовуються такі технології, як розумна дошка або планшети [9].

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

До переваг відносимо таке:

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

Існують також недоліки такої моделі:

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

Назва інтерактивної моделі походить від слів inter (взаємний), act (дія). Процес навчання має відбуватись в умовах постійної та активної взаємодії всіх учнів. Відомо, що термін «інтерактивна педагогіка» ввів Г. Фріц. Більшість сучасних методичних інновацій пов'язані з використанням інтерактивних

технологій навчання. Семантичне значення слова інтерактив походить від англійського слова «interact», де «inter» означає взаємний, «act» – дія. Інтерактив – це процес навчання, під час якого відбувається безперервна та активна взаємодія всіх учасників процесу. Отже, інтерактивне навчання – це діалогове навчання, де відбувається обмін інформацією між викладачем та студентом. Отже, інтерактив здатний до взаємодії та діалогу. Інтерактивне навчання – є специфічною формою організації діяльності, яка має свою мету – створити комфортні умови навчання під час яких кожен учень відчує свою успішність та інтелектуальну ефективність.

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

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

- розумової (інтенсивність мислення, генерування ідей, висловлювання припущень, проектування, моделювання, конструювання, дослідження тощо, творча уява, зосередженість, увага, спостережливість, аналітико-синтезуючі операції); емоційної (емоційна напруга, переживання);
- соціальної (імітація виконання соціальних ролей, обмін думками, ставлення, судження тощо);
- фізичної (напруження, практична діяльність, рухливість).

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

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

Виділимо основні характеристики інтерактивного навчання:

- навчальний процес відбувається за умови, коли постійно активно взаємодіють всі учні;
- це співнавчання, взаємонавчання (колективне, групове та звичайно кооперативне), коли учень з вчителем є рівноправними або рівнозначними суб'єктами такого навчання;
- організація інтерактивного навчання полягає в моделюванні життєвих ситуацій, застосуванні рольових ігор, спільному вирішенні проблем на основі аналізу обставин і провідної ситуації;
- інтерактивне навчання ефективно сприяє формуванню комунікативних умінь і навичок, розвитку цінностей, створенню атмосфери співпраці та взаємодії;
- цей тренінг виявляє лідерські якості, але передбачає як домінування одного з учасників навчального процесу над іншими, так і домінування однієї думки над іншою.

Однак у цієї моделі є і недоліки. По-перше, для вивчення певної інформації потрібно багато часу. Також необхідно створити інший підхід до оцінювання знань учнів. Найбільшим недоліком є недостатня розробленість організації занять з використанням інтерактивних технологій [11]. Отже, проаналізувавши різні моделі навчання, можемо стверджувати, що інтерактивна модель навчання є найбільш ефективною у сучасному світі.

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

Переваги інтерактивного тестування:

- миттєве, та досить швидке і досить комфортне отримання результатів тесту;
- об'єктивність та правильність отриманої оцінки;

- інтерактивне тестування, з точки зору сприйняття студентами стає більш цікавішим і більш захоплюючим, якщо порівнювати цей вид тестування з традиційними іспитами за білетами або письмовими контрольними роботами;
- інтерактивні тести є досить справедливим методом оцінювання набутих знань, оскільки одночасно всі учні однієї конкретної групи знаходяться в однакових умовах, як під час самого інтерактивного тесту, так і при отриманні оцінки в кінці;
- тестування – це більш комплексний інструмент, за допомогою тестів викладач може встановити рівень набутих студентом знань як з окремої частини окремого предмета в цілому, так і з дисципліни в цілому;
- такий вид контролю ефективніший з економічної точки зору, через те що учитель створює базу запитань і може користуватися цією базою тривалий час;
- також інтерактивне тестування – більш гуманний інструмент через те що воно повинно знімати нервове напруження, яке майже завжди виникає перед традиційним іспитом і в деяких випадках може суттєво вплинути на отриману оцінку;
- після контрольних робіт вчитель отримує інформацію про засвоєння навчального матеріалу для кожного учня індивідуально, аналізуючи ці дані, можна наочно побачити інформацію про труднощі засвоєння навчального матеріалу з тієї чи іншої теми.

Недоліки інтерактивного тестування:

- в деяких випадках причини неякісного написання контрольної роботи є об’єктивними і заслуговують на підтримку вчителя, а це неможливо помітити під час контрольної роботи;
- навіть якщо питання та завдання для проведення інтерактивних контрольних робіт складені грамотно та зрозуміло для учня, вони все одно не зможуть продемонструвати творчий рівень знань учня;

- іноді студентам просто не вистачає встановленого на це часу щоб скласти тест;
- щоб завдання, тобто запитання на відповідних рівнях, не повторювалися щоразу, потрібно з самого початку створення тесту вносити в базу відповідну кількість питань, щоб учень не отримував однакові запитань кожного разу в іншому судовому процесі, іншими словами, вчителю потрібен творчий підхід до підготовки тестових завдань, щоб бути впевненим в об'єктивності та достовірності отриманої оцінки;
- в інтерактивному тестуванні, як і в будь-якому процесі, не можна забувати про елемент випадковості, це означає, що інколи старанний учень не відповідав на запитання першого рівня, а студент, який не знав навчального матеріалу, з легкістю відповідав на запитання другого та третього рівня, причиною цього може бути як нездача питання наприклад першого рівня, так і навіть елементарне вгадування правильної відповіді на питання другого рівня, в результаті результати тестів спотворюються.

Аналізуючи переваги та недоліки інтерактивного тестування з дисциплін, очевидно, що, порівняно з традиційними формами іспитів, запропонований вид контролю є достатньо об’єктивним та якісним, що обмежує вплив суб’єктивних факторів на оцінку студента в процесі перевірки. рівень знань з конкретної дисципліни [12].

1.4 Методи генерації тесту

Є три найвідоміші методи генерації інтерактивного тесту: параметризованих тестів; семантичних мереж; понятійно-тезисної моделі [13].

Методу параметризованих тестів полягає в тому, що студентам подається різние шаблонне завдання, яке відрізняється тільки певними параметрами, які генеруються автоматично [14]. Кожен учень отримує індивідуальне завдання, а система за власною формулою чи алгоритмом, підставляє параметри та отримує

правильну відповідь для того щоб далі перевірити відповіді, уведені учнем. Недоліком методу є те, що він вузько предметно спрямований.

Перевагами методу є:

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

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

Метод який базується на основі семантичних мереж ґрунтується на використанні різних тріад [15]. Завдання інтерактивного тесту будують упущенням одного елементу тріади і запитанням про те, чого не вистачає. Перевага даного методу: можливість контролю рівня знань в майже у всіх науках. Недоліками даного методу є:

- незрозумілість лінгвістично, а дуже часто – недоцільність завдань, які генеруються;
- складність завдань заздалегідь не визначають через що можливе повторення інтерактивних тестових завдань.

Метод понятійно-тезисної моделі. Головною структурною ідеєю для формування інтерактивних тестових завдань є зв’язок «навчальний матеріал» – «тези» – «поняття» [16]. Автоматична генерація тестів відбувається на основі створених різних тестових шаблонів. Понятійно-тезисна модель передбачає можливість розширення всіх множини класів користувацькими класами тез, це дозволяє розширювати модель і налаштовувати її під різні області. Переваги даного методу: краща лексична зрозумілість порівняно з іншими методами; можливість контролювати та діагностувати знання в різних науках. Недоліками методу є те, що заздалегідь не визначають складність завдань, а також є можливість повторення інтерактивних тестових завдань.

Порівняльна характеристика методів наведена в таблиці 1.1. Методи, які описані вище не здатні враховувати складність тестових завдань, допускають великі семантичні, а іноді лексичні невідповідності. Тестові завдання повторюються та матють одну тему. Тому є підстава використовувати метод на основі методу параметризованих тестів.

Таблиця 1.1 – Порівняльна характеристика методів

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

1.5 Аналіз сучасних засобів тестування знань

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

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

Веб-додатки – додатки, які створені для роботи в серверному середовищі та написані різними мовами програмування, як: PHP та ASP та Perl, Ruby або Java. Ці мови надають відмінну можливість створювати повнофункціональні програми та ще й створювати системи управління для різних додатків. Веб-додатки нажаль працюють лише в серверному середовищі, це означає вам потрібно встановити свій сервер для роботи. Такі програми також можна розповсюджувати в Інтернеті, там

сервер вже налаштований, і для роботи користувач повинен мати лише веб-браузер та доступ до Інтернету. Ще перевагою веб-додатків є – користувач може мати доступ до проекту через Internet з кожного девайсу [17]. Переваги та недоліки windows і веб-додатків наведені в таблиці 1.2. Враховано всі переваги і недоліки та було прийнято рішення розробляти інтерактивний веб-додаток, бо він має низку переваг в порівнянні з windows-додатками.

Таблиця 1.2 – Переваги та недоліки windows та веб-додатків

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

1.5.1 LearningApps

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

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

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



Рисунок 1.1 – Приклад роботи програми LearningApps

1.5.2 Online Test Pad

Online Test Pad – безкоштовний багатофункціональний сервіс наприклад для організації навчального процесу або тестування через Інтернет. Зручний веб-

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

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

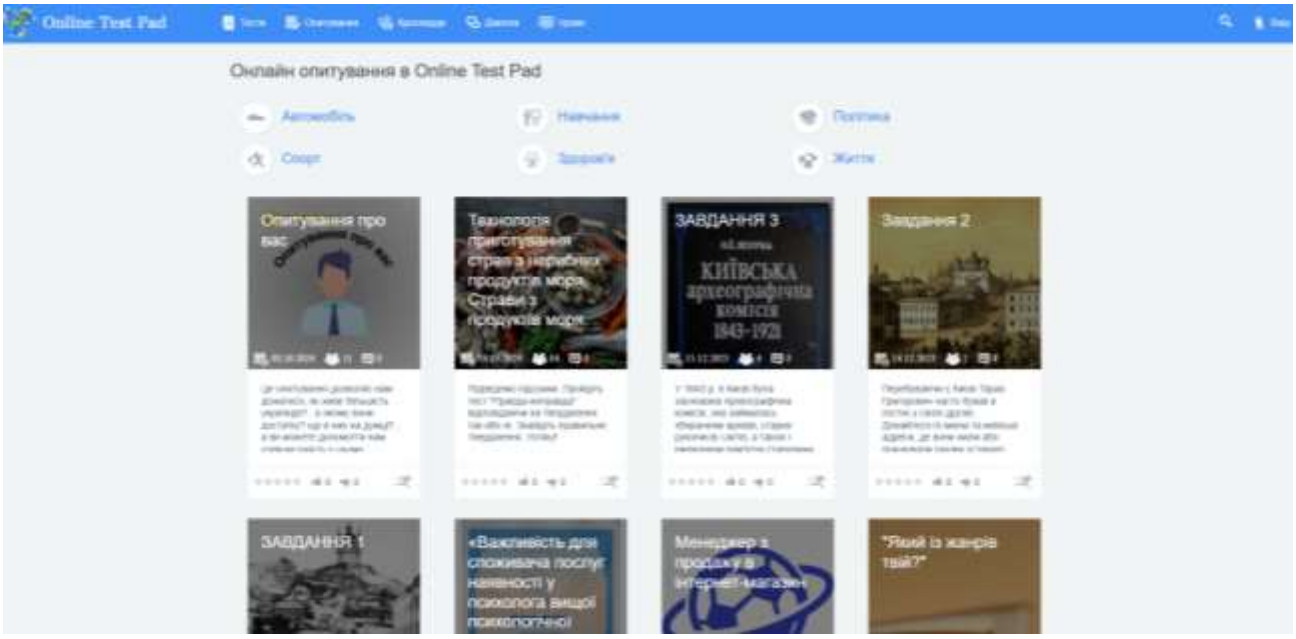


Рисунок 1.2 – Приклад роботи програми Online Test Pad

1.5.3 ClassMarker

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

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

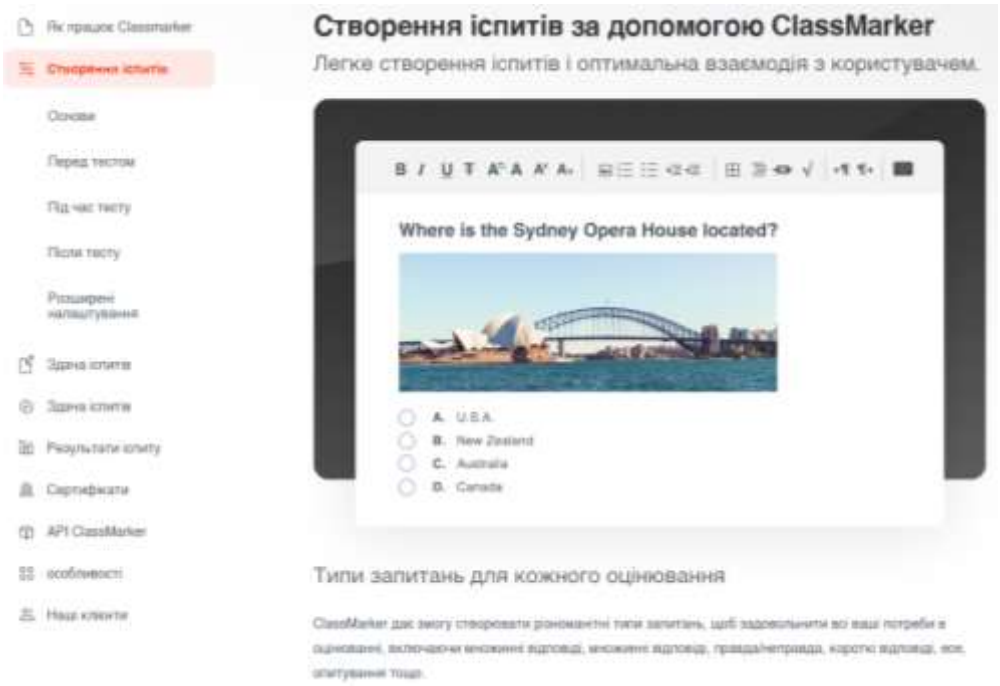


Рисунок 1.3 – Приклад роботи програми ClassMarker

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

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

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

Таблиця 1.3 – Переваги та недоліки додатків LearningApps, Online Test Pad та ClassMarker

Назва додатку	Можливість автоматизованої генерації тесту	Відсутність повторень завдань	Можливість встановлювати складність завдань	Можливість перегляду результатів
LearningApps	–	+	+	+
Online Test Pad	–	+	–	+
ClassMarker	+	–	–	+

1.6 Постановка задачі

Ціллю магістерської кваліфікаційної роботи є розробка додатку для інтерактивного тестування знань, що забезпечує автоматизоване формування, проведення тестів та аналіз результатів із високою точністю, об’єктивністю та зручністю використання. Для її реалізації необхідно здійснити аналітичний огляд предметної області, включаючи аналіз існуючих підходів до тестування знань, розглянути та проаналізувати моделі навчання й оцінювання знань, дослідити методи автоматизованої генерації тестів.

Завдання розробки полягає у створенні додатку, який дозволить: автоматизувати процес генерації тестів із врахуванням складності завдань і відсутністю повторів, забезпечити об’єктивність оцінювання завдяки чітким алгоритмам перевірки.

Для реалізації завдань розробки необхідно: розробити алгоритми генерації тестів і оцінювання результатів, створити базу даних для збереження тестових завдань і результатів користувачів, виконати тестування додатку на відповідність функціональних і якісних вимог.

Результатами роботи стане готовий до впровадження додаток для інтерактивного тестування знань, що відповідає сучасним вимогам та стандартам освітнього процесу.

РОЗДІЛ 2. РОЗРОБКА АЛГОРИТМУ ІНТЕРАКТИВНОГО ТЕСТУВАННЯ ЗНАНЬ

2.1 Розробка алгоритму інтерактивного тестування знань

Загальний алгоритм інтерактивного тестування знань має включати такі кроки:

- step 1 – генерація тесту, під час даного кроку інтерактивний тест

автоматизовано генерується з великого банку тематичних завдань, які мають знаходитись в базі даних і відповідно до налаштувань, які виставляє відповідальна за проведення тесту особа;

- step 2 – вибір тесту, під час даного кроку користувач обирає інтерактивний тест для того щоб пройти зі списку доступних інтерактивних тестів, ці інтерактивні тести завчасно мають автоматизовано генеруватися відповідальною за проведення тесту особою, також інтерактивний тест генерується відповідно до налаштувань тесту;
- step 3 – якщо тест обрано, то крок 4, якщо ні – крок 2;
- step 4 – виведення інтерактивного тесту, під час даного кроку виводяться по черзі інтерактивні тестові завдання;
- step 5 – проходження тесту, під час цього проходження інтерактивного тесту користувачу потрібно буде обирати відповіді з запропонованих;
- step 6 – якщо тест буде пройдено, то перехід до кроку 7, якщо ні — крок 5;
- step 7 – виведення результатів тесту користувачеві, після закінчення інтерактивного тесту, користувач має отримати результат проходження, в якому він зможе побачити наприклад скільки він дав правильних або неправильних відповідей на кожне запитання.

Діаграма активності інтерактивного тестування наведена на рисунку 2.1.



Рисунок 2.1 – Діаграма активності інтерактивного тестування знань

2.2 Розробка алгоритму генерації тесту

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

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

Даний алгоритм робить неможливим повторення тестових завдань у згенерованому тесті.

Діаграма активності генерації тесту наведена на рисунку 2.2.

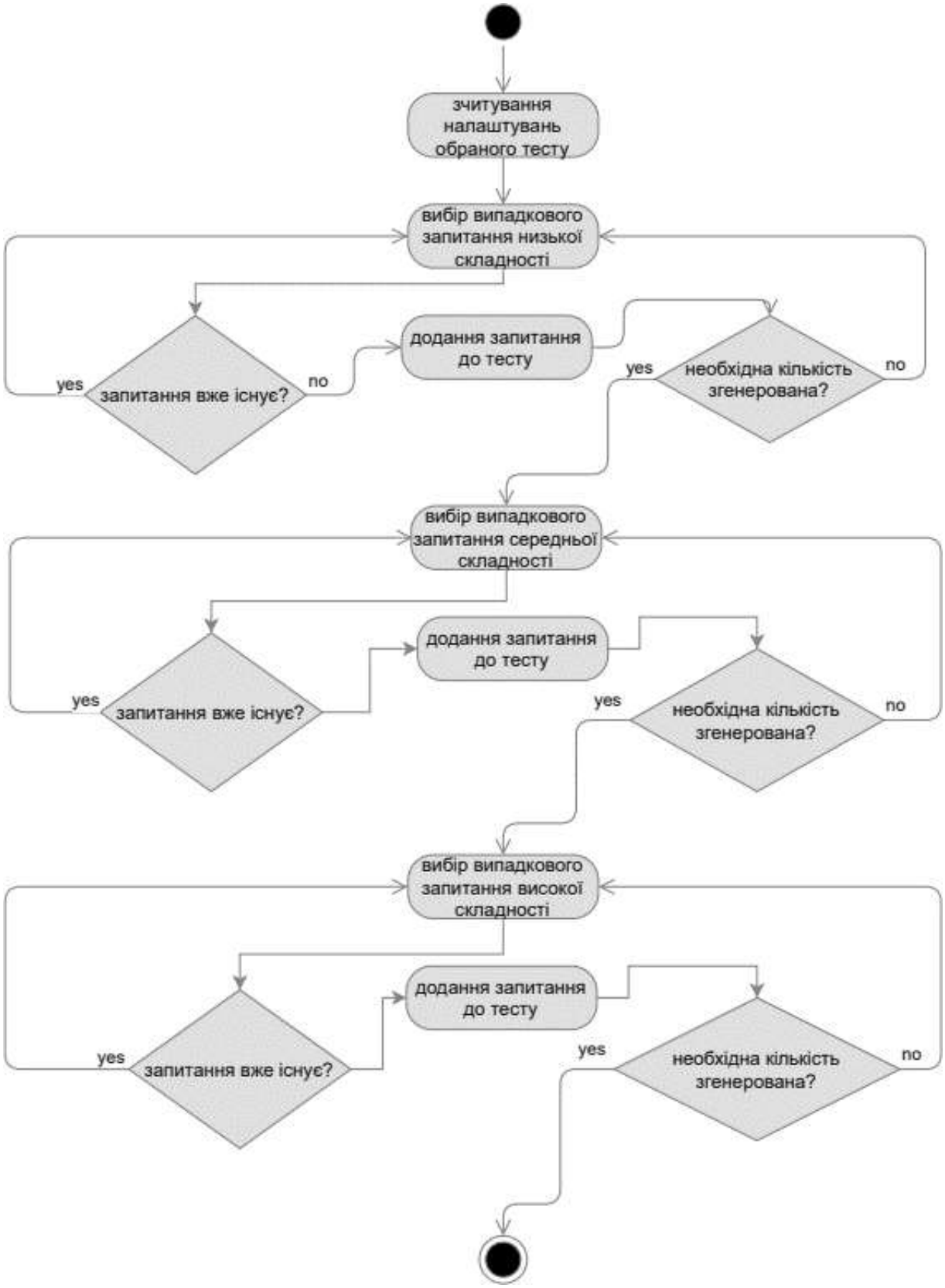


Рисунок 2.2 – Діаграма активності генерації тесту

РОЗДІЛ 3. ПРОЕКТУВАННЯ ДОДАТКУ ДЛЯ ІНТЕРАКТИВНОГО ТЕСТУВАННЯ ЗНАНЬ

3.1. Розробка структури додатку для інтерактивного тестування знань

Розроблений додаток для інтерактивного тестування знань буде виконувати такі функції:

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

Для того, щоб реалізувати дані функції додаток для інтерактивного тестування знань повинен мати: модуль генерації, модуль тестування, модуль результатів, модуль створення та редагування банку завдань. Генерацію тесту має виконувати модуль генерації. Головна задача модуля – автоматизована генерація тесту. Ще він дає змогу редагувати тест, тобто змінити його налаштування або згенерувати його знову.

Модуль взаємодіє з базою даних:

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

Окремий модуль виконує такі важливі функції як:

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

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

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

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

Модуль має такі можливості:

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

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

Модуль має такі можливості:

- виведення списку пройдених тестів;
- виведення користувачів, які пройшли певний тест;

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

Взаємодія між складовими додатку для інтерактивного тестування знань наведена в Додатку А.

3.2 Обґрунтування вибору середовища розробки для реалізації додатку для інтерактивного тестування знань

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

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

WebStorm також забезпечує налагодження JavaScript і надає різні функції: виявлення точок зупину в HTML і JavaScript, налаштування параметрів точок зупину, тестування синтаксису коду в реальному часі та багато іншого. Продукт також підтримує платформи JQuery, Qooxdoo та Bindows та YUI та Prototype, DoJo

та MooTools. WebStorm забезпечує інтегровану перевірку тексту на теги, кодові послідовності та орфографічні помилки. WebStorm дозволяє редагувати файли та автоматично синхронізувати їх за запитом під час віддаленої роботи або під час зберігання. Продукт підтримує контроль версій і всі попередні версії коду, а ще реєструє всі дії та зміни. Коли створюєте історію в WebStorm, ви також можете відновлювати вирази коду та блоки і навіть цілі файли.

Особливості WebStorm:

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

Atom – безкоштовний редактор тексту і коду, який розроблений GitHub, для використання як окремого рішення та як стек технологій наприклад для створення різноманітних спеціалізованих рішень. Наприклад, на платформі Atom створено

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

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

Electron – це окрема структура, яку можна використовувати для створення будь-якої спеціальної програми, логіка якої визначається JavaScript, HTML і CSS, а

функціональність якої можна розширити за допомогою додаткової системи. Розробники також мають доступ до модулів Node.js, а також розширених API для створення користувацьких діалогів, створення контекстних меню, керування вікнами та взаємодії з підсистемами Chrome. Необхідність використовувати рідне ядро браузера на основі Chrome замість розробки редактора як веб-програми, що працює у звичайному браузері та через необхідність реалізації додаткових функцій, які недоступні через звичайні веб-API. Наприклад, Atom забезпечує вбудований файловий менеджер і гнучкі функції пошуку файлів, такі які неможливо реалізувати за допомогою наприклад звичайних веб-додатків. Редактор також використовує деякі і внутрішні підсистеми Chromium, наприклад такі як механізми регулярних виразів та власні діалогові елементи. Більше того, оскільки компоненти Atom майже завжди працюють локально, то це спрощує розробку надбудов, доступ до ресурсів і спільний доступ [22]. Для реалізації додатку для інтерактивного тестування знань було обрано середовище розробки WebStorm.

3.3 Обґрунтування вибору мови програмування для розробки додатку для інтерактивного тестування знань

Проаналізовано мови програмування JavaScript і C#. JavaScript – це динамічна та об’єктно-орієнтована мова програмування. Є реалізацією стандарту ECMAScript. Найчастіше JS використовується для створення скриптів сайту, які наприклад дозволяють клієнту взаємодіяти з користувачем та керувати браузером, а також асинхронно обмінюватися даними з сервером і змінювати структуру і зовнішній вигляд сайту. JavaScript класифікується як мова сценаріїв з динамічним типом. Крім того, JavaScript також частково підтримує інші парадигми програмування а також деякі пов’язані елементи архітектури, зокрема: динамічну і слабку типізацію, також автоматичне керування пам’яттю та успадкування прототипів, такі функції, як об’єкти першого класу.

JavaScript використовується для:

- написання сценаріїв для веб-сайтів, що забезпечують їм інтерактивність;

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

Переваги JavaScript:

- швидкість – JavaScript може бути дуже швидким, так як він часто запускається безпосередньо в браузері клієнта, поки не потрібні зовнішні ресурси, JavaScript не сповільнюється під час викликів серверного сервера, крім того, усі основні браузери підтримують JIT-компіляцію JavaScript (точно вчасно), тобто немає великої необхідності компілювати код перед його запуском;
- простота – синтаксис JavaScript натхненний Java, і його також відносно легко вивчити порівняно з іншими популярними мовами, такими як C++.
- популярність – JavaScript є майже повсюди в Інтернеті, і також з появою Node.js він використовується все більше і більше на стороні сервера, існує незліченна кількість ресурсів для навчання JavaScript, як StackOverflow, так і GitHub демонструють дедалі більше проектів JavaScript, і популярність, яку він здобув за останні роки, лише зростатиме;
- сумісність – на відміну наприклад від PHP або інших мов сценаріїв, JavaScript можна вбудувати в майже будь-яку веб-сторінку, JavaScript можна також використовувати в багатьох різних типах програм, наприклад підтримуючи інші мови, такі як Pearl і також PHP;
- навантаження на сервер – JavaScript працює на стороні клієнта, тому, як правило, він зменшує потребу в серверах, і простим програмам може і взагалі не знадобитися цей сервер;

- розширені інтерфейси – JavaScript також можна використовувати для створення наприклад таких функцій, як перетягування та скидання, і таких компонентів, як повернення, що постійно покращує інтерфейс користувача та роботу в Інтернеті;
- розширена функціональність – розробники можуть розширити функціональність веб-сторінок, написавши фрагменти JavaScript для різних сторонніх доповнень, як Greasemonkey;
- універсальність – є багато разних способів використовувати JavaScript через сервери Node.js, отже якщо ви хотіли завантажити Node.js за допомогою Express, то використовувати базу даних документів, таку як MongoDB, і використовувати JavaScript на інтерфейсі, можна розробити цілу програму JavaScript спереду назад, коли використовуючи лише JavaScript;
- оновлення – з моменту появи ECMAScript 5 (специфікація сценаріїв, на яку покладається JavaScript), ECMA International займається оновленням JavaScript щорічно, наразі ми отримали підтримку браузера для ES6 у 2017 році і сподіваємось на підтримку ES7 у майбутньому [23].

C# – це об’єктно-орієнтована, безпечна для типів мова програмування для .NET розроблено Андерсом Ґалесбергом, Скоттом Вільхамутом і Пітером Голдом під Microsoft Research (належить Microsoft). Синтаксис C# схожий на C++ і Java. Мова суворо статично типізована, також підтримує поліморфізм, перевантаження операторів та покажчики на функції-члени класу, події, атрибути, властивості, коментарі у форматі XML винятки. Прийняття багатьох з своїх попередників – Object Pascal, C++, Module і Smalltalk – C#, засноване практично на практиці їх використання, виключає деякі моделі, такі які виявилися проблематичними наприклад при розробці програмного забезпечення, такі наприклад як C#, на відміну від C++, не контролює успадкування багатьох класів [24]. Для розробки додатку для інтерактивного тестування знань було також обрано мову JavaScript.

3.4 Обґрунтування вибору бази даних для реалізації додатку для інтерактивного тестування знань

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

MySQL – це компактний багатопоточний сервер баз даних. Відрізняється високою швидкістю, стабільністю і простотою використання. MySQL вважається хорошим рішенням для програм малого та середнього розміру. Коди серверів скомпільовані на кількох платформах. Повністю можливості сервера представлені в системах UNIX, де є підтримка багатопоточної обробки, що підвищує ефективність всієї системи.

Параметри сервера MySQL:

- простий в установці і використанні;
- підтримується велика кількість користувачів, які одночасно працюють з вашою базою даних;
- кількість рядків у таблицях бази може досягати 50 мільйонів;
- висока швидкість виконання команди;
- наявність простої і ефективної системи безпеки [25].

SQL, також відомий як Microsoft SQL, з’явився набагато раніше, ніж MySQL. Корпорація Microsoft розробила SQL у 1980-х роках. Вони залишаються основою якісного SQL навіть після багатьох років і є важливим рішенням для програмного забезпечення великих підприємств. SQL Server більше підходить для розробників, таких які використовують .NET як мову програмування як конкуруючу комбінацію PHP із MySQL. SQL Server використовує єдину систему, розроблену Microsoft,

проти кількох механізмів, запропонованих MySQL. Це дає розробникам MySQL більше гнучкості, оскільки вони також можуть вибирати різні системи для різних таблиць та на основі швидкості, надійності чи інших параметрів. Популярним двигуном MySQL є InnoDB, який втрачає деяку швидкість, але забезпечує підвищену надійність. Корпорація Microsoft вимагає від вас придбати ліцензію на запуск кількох баз даних на SQL Server. Є безкоштовна версія, але вона лише для ознайомлення з реляційною СУБД. MySQL, з іншого боку, використовує ліцензію GNU, що робить його абсолютно безкоштовним [26].

Для реалізації додатку для інтерактивного тестування знань була також обрана база даних MySQL.

3.5 Обґрунтування вибору фреймворку для реалізації додатку для інтерактивного тестування знань

Проведено аналіз фреймворків Vue js і React js. Vue.js – це фреймворк JavaScript, який використовує шаблон MVVM для створення користувацьких інтерфейсів на основі моделей даних за допомогою реактивного зв’язування даних. MVVM – це модель розробки, таку яка дозволяє розділити додаток на три функціональні частини:

- model – основна логіка програми (наприклад робота з даними, розрахунки, запити тощо);
- view – перегляд або відображення (наприклад інтерфейс);
- viewmodel – модель подання, вона служить шаром між видом та моделлю.

Цей розділ дозволяє наприклад пришвидшити розробку та підтримувані програми – ви можете також змінити один компонент, щоб не впливати на код. Взаємодія між функціональними частинами показана на рисунку 3.1. Vue використовує синтаксис шаблонів на основі HTML, який дозволяє декларативно прив’язувати відтворення DOM до базових екземплярів даних у Vue. Усі шаблони Vue є також дійсними HTML і можуть аналізуватися браузерами та аналізаторами HTML.

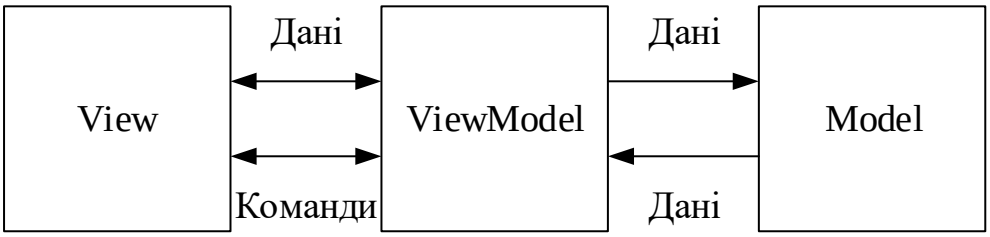


Рисунок 3.1 – Взаємодія між функціональними частинами шаблону MVVM

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

У Vue ви можете також використовувати синтаксис шаблону або ще писати функції відтворення безпосередньо за допомогою JSX. Для цього треба замінити шаблон функцією візуалізації. Візуалізація дає змогу створювати ефективні шаблони на основі компонентів – наприклад, нова транспортна система тепер повністю заснована на компонентах і використовує внутрішні та можливості візуалізації. Однією з найвідмітніших особливостей Vue є система дискретного струму. Моделі – це просто плоскі об’єкти JavaScript. Завдяки цьому управління статусом дуже просте та інтуїтивно зрозуміле. Vue забезпечує оптимізоване перемальовування без додаткової роботи. Кожен компонент відстежує свої реактивні залежності під час рендерингу, тому система точно знає, коли відображати та які компоненти рендерити.

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

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

переходів;

- інтегрування різних сторонніх JavaScript бібліотек анімацій, як Velocity.js [27].

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

React можна вважати тираном світу веб-фреймворків. Це порушує всі правила. Кращі практики. Якщо завжди вважалося необхідним розділяти теги, стилі та JavaScript в окремі файли, то з появою наприклад JSX, де все може бути в одному місці, як погляди розробників розділяться. Деякі люди вважають, що це спрощує розробку, дозволяє відразу знайти помилку, тому що React не компілюється, коли чогось не вистачає або є щось зайве. А інші стверджують, що це тільки заплутує і ускладнює весь процес [28]. Для реалізації додатку для інтерактивного тестування знань був обраний фреймворк Vue.

3.6 Розробка модулів

3.6.1 Розробка модуля генерації

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

Модуль виконує такі дії:

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

Діаграма активності модуля генерації зображена на рисунку 3.2.

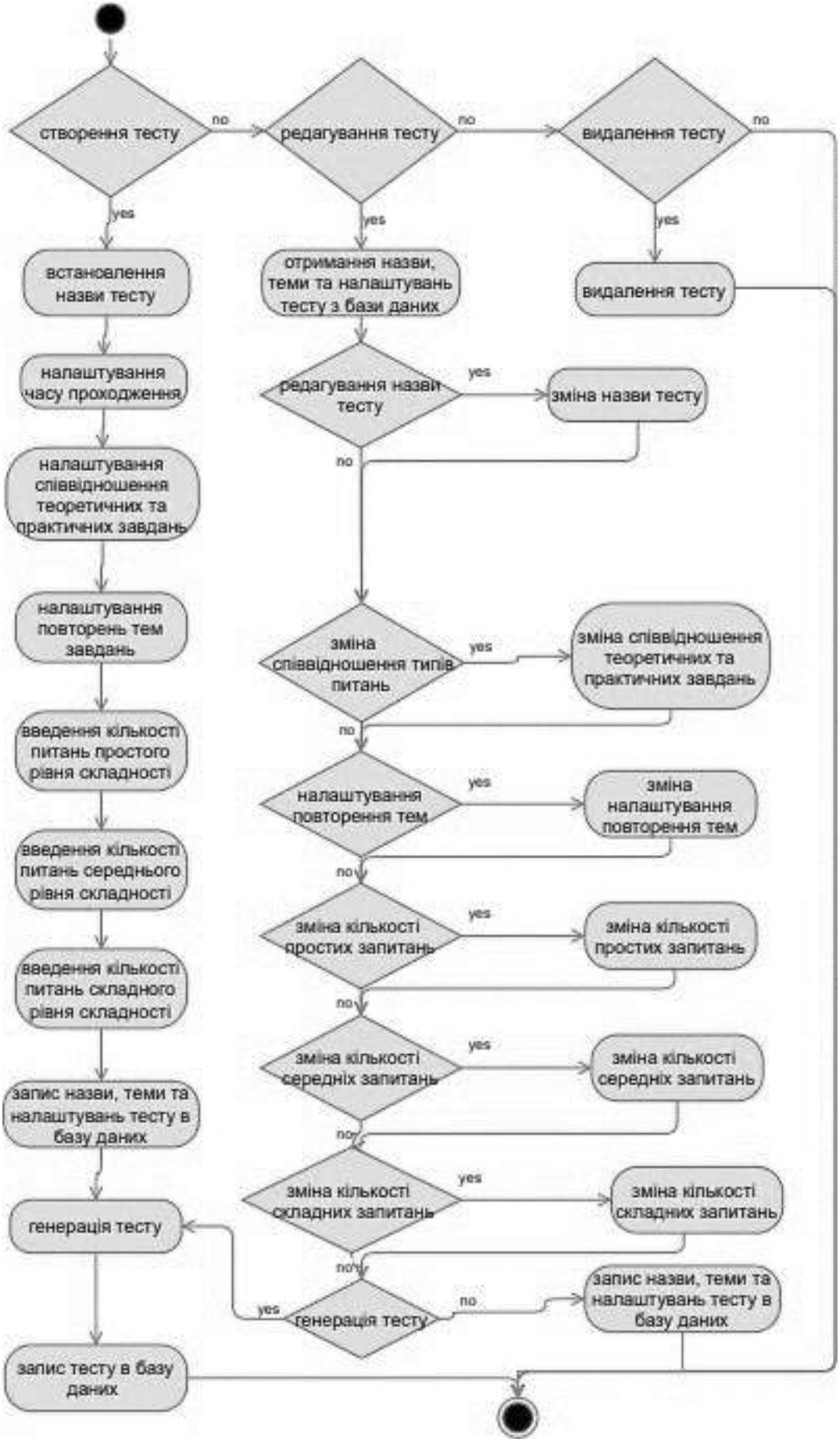

створення тесту
no
редагування тесту
no
видалення тесту
no
yes
yes
yes
встановлення назви тесту
отримання назви, теми та налаштувань тесту з бази даних
видалення тесту
налаштування часу проходження
редагування назви тесту
yes
зміна назви тесту
no
налаштування співвідношення теоретичних та практичних завдань
налаштування повторень тем завдань
зміна співвідношення типів питань
yes
зміна співвідношення теоретичних та практичних завдань
no
введення кількості питань простого рівня складності
налаштування повторення тем
yes
зміна налаштування повторення тем
no
введення кількості питань середнього рівня складності
зміна кількості простих запитань
yes
зміна кількості простих запитань
no
введення кількості питань складного рівня складності
зміна кількості середніх запитань
yes
зміна кількості середніх запитань
no
запис назви, теми та налаштувань тесту в базу даних
зміна кількості складних запитань
yes
зміна кількості складних запитань
no
генерація тесту
yes
генерація тесту
no
запис назви, теми та налаштувань тесту в базу даних
запис тесту в базу даних

Рисунок 3.2 – Діаграма активності модуля генерації

3.6.2 Розробка модуля тестування

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

Модуль має виконувати такі дії:

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

Діаграма активності модуля тестування зображена в Додатку Б.

3.6.3 Розробка модуля створення та редагування банку завдань

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

Модуль має виконувати такі дії:

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

3.6.4 Розробка модуля результатів

Модуль результатів призначений для перегляду результатів тестування

користувачів. Модуль взаємодіє з базою даних отримуючи дані.

Модуль виконує такі дії:

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

3.7 Розробка бази даних

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

Розрізняють такі бази даних за моделлю організації:

- ієрархічна – представляється у вигляді дерева, яке складається з об’єктів різних рівнів, між цими об’єктами існують зв’язки «предок-нащадок», у даному випадку можливо, що об’єкт може не мати дочірніх або має декілька, тоді як дочірній постійно повинен мати лише одного предка;
- мережева – є подібною до ієрархічної, але за винятком того, що кожен об'єкт має більше одного предка;
- реляційна – зберігає дані у вигляді таблиць, найбільш поширені СУБД використовують реляційну модель даних;
- об’єктно-орієнтована – у базі даних дані створюються у вигляді моделей об’єктів [29].

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

- простота та доступність для розуміння кожним користувачем, єдиною використовуваною конструкцією є «таблиця»;
- суворі правила проектування, вони базуються на математичному апараті;
- також є повна незалежність даних, зміни в програмі при зміні реляційної БД є мінімальними;

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

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

- запитання – буде містити тему запитання, назву запитання, номер запитання, тип запитання, складність запитання;
- відповіді – буде містити номер і відповіді на всі запитання;
- користувачі – буде містити логіни і паролі користувачів;
- адміністратори – буде містити логіни та паролі лише адміністраторів;
- тести – буде містити назву тесту, номер тесту, номери завдань, які належать до цього інтерактивного тесту;
- налаштування тесту – буде містити номер тесту і всі його налаштування;
- час проходження;
- кількість питань кожного рівня складності;
- співвідношення теоретичних і практичних питань;
- налаштування повтору тем тестових завдань.

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

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

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

4. ТЕСТУВАННЯ РОЗРОБЛЕНОГО ДОДАТКУ ІНТЕРАКТИВНОГО ТЕСТУВАННЯ ЗНАНЬ ТА АНАЛІЗ РЕЗУЛЬТАТІВ РОБОТИ

Експериментальні дослідження щодо функціонування додатку для інтерактивного тестування знань передбачали дослідження з перевірки таких можливостей:

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

Під час запуску додатку відкривається сторінка авторизації з можливістю реєстрації.

Після авторизації, в залежності від того, який рівень доступу (користувач або адміністратор), відкриється головна сторінка додатку. Вид головної сторінки для адміністратора зображений на рисунку 4.1, для звичайного користувача – на рисунку 4.2.

Можна побачити, що в адміністратора з’являється вкладка «Адміністрування», що дає можливість створювати та редагувати тести, завдання, а також теми та дивитись результати тестування користувачів. Користувачеві

доступна лише вкладка «Тестування», він може пройти тест на вибір. Так реалізовано два рівня доступу до додатку: адміністратор та користувач.

У вкладці «Адміністрування» є чотири вкладки: «теми», «завдання», «результати» та «тести».

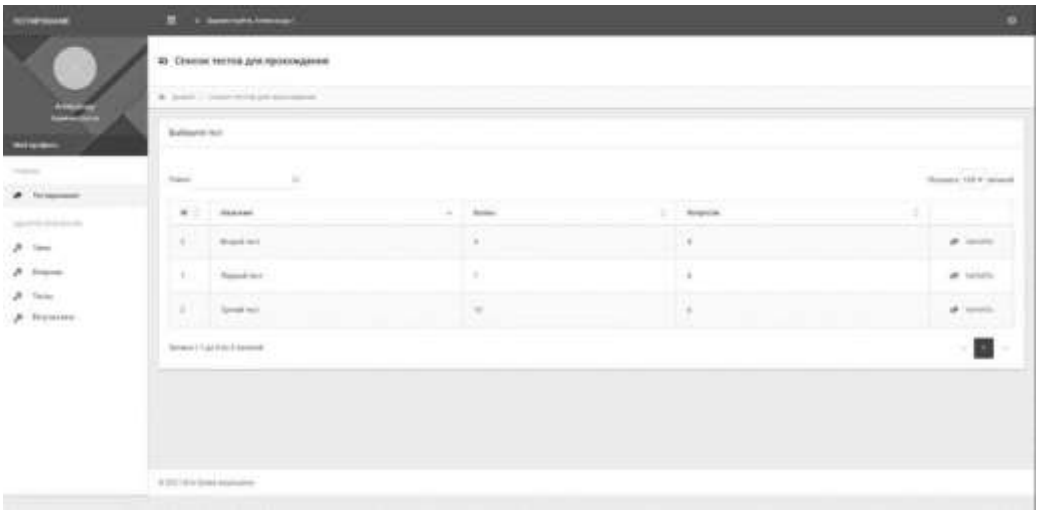


Рисунок 4.1 – Г оловна сторінка адміністратора

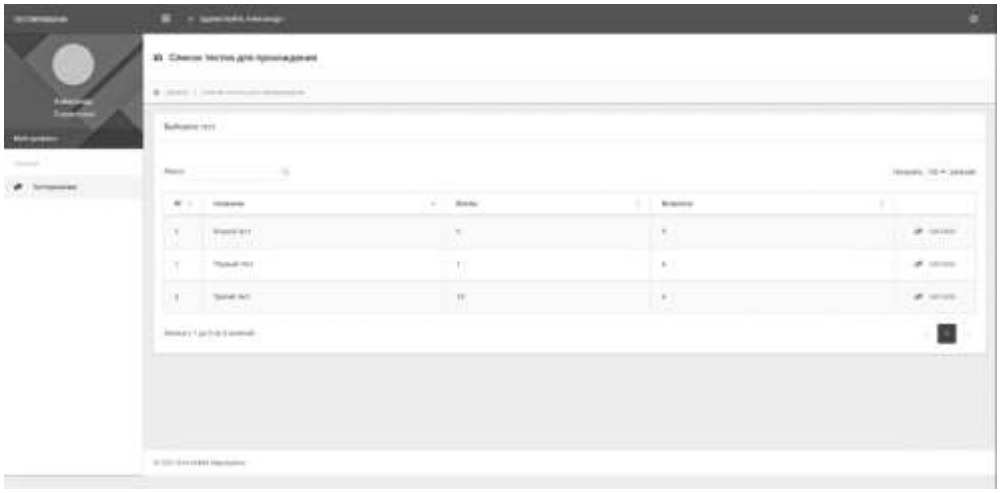


Рисунок 4.2 – Головна сторінка користувача



Рисунок 4.3 – Сторінка списку тем

Якщо нажати на вкладку «теми» – відкриється список тем завдань з можливістю додавання, редагування і видалення теми, вигляд сторінки можна побачити на рисунку 4.3. Додавання теми можна побачити на рисунку 4.4. Редагування теми можна побачити на рисунку 4.5. Редагування, створення і видалення теми коректно працює.

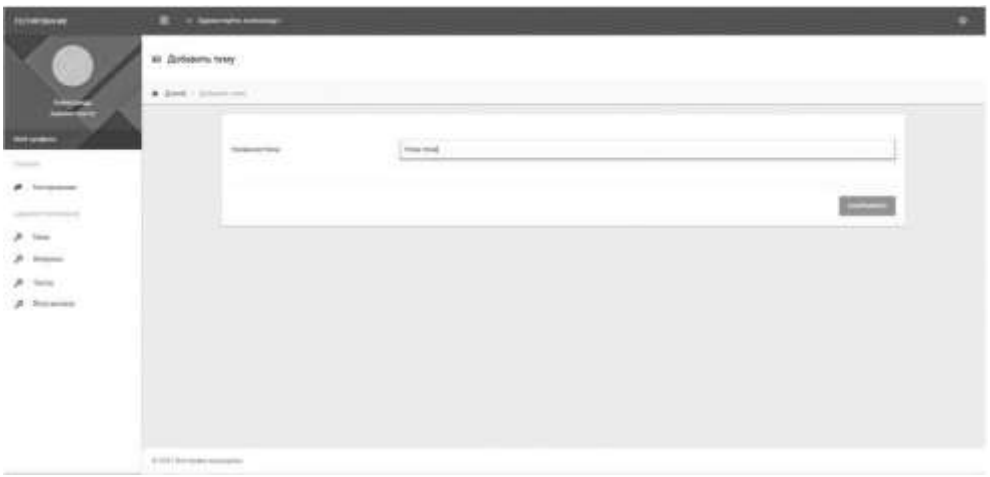


Рисунок 4.4 – Сторінка додавання теми

При переході на вкладку «Завдання» відкриється сторінка із списком тестових завдань з можливістю редагування, створення і видалення тестових завдань. Сторінка із списком завдань показана на рисунку 4.6, сторінка додавання завдання показана на рисунку 4.7, сторінка редагування завдання – на рисунку 4.8.

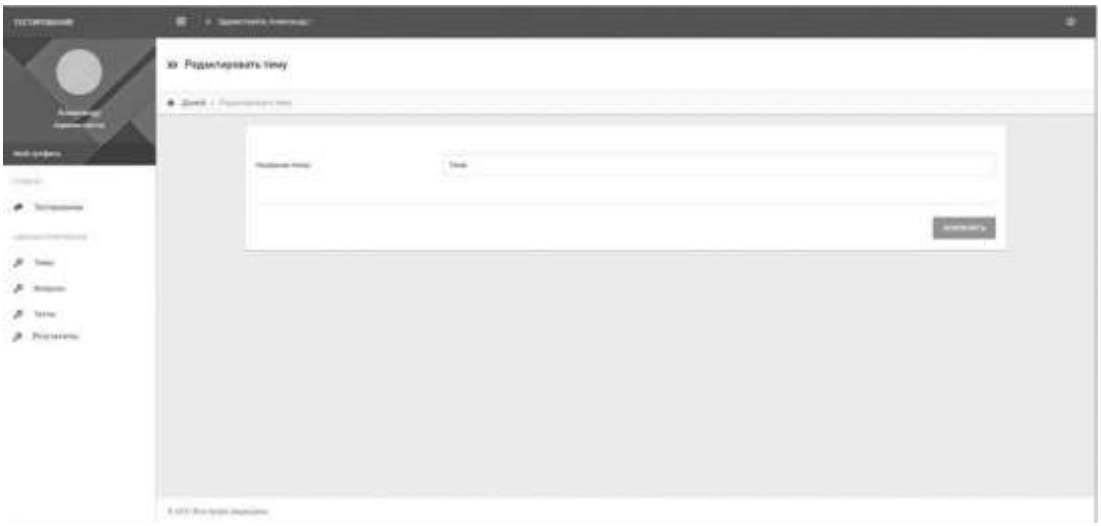


Рисунок 4.5 – Сторінка редагування теми

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

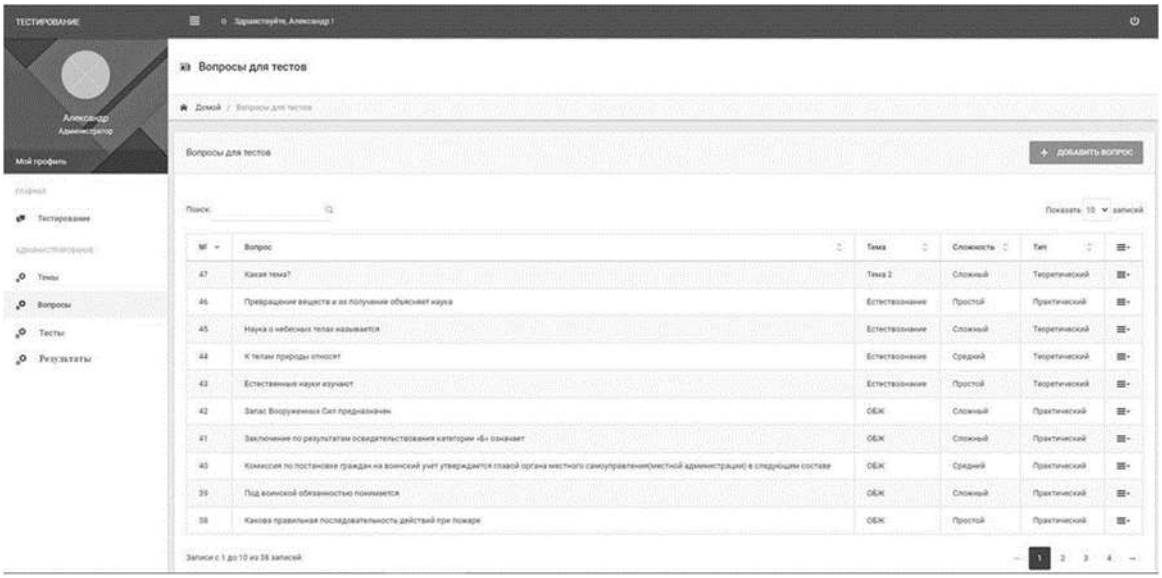


Рисунок 4.6 – Сторінка списку тестових завдань

Натиснувши на вкладку «тести», відкриється сторінка із списком тестів з можливостю їхнього редагування, створення і видалення. Сторінка списку тем зображена на рисунку 4.9, сторінка створення тесту зображена на рисунку 4.10, сторінка редагування – на рисунку 4.11.

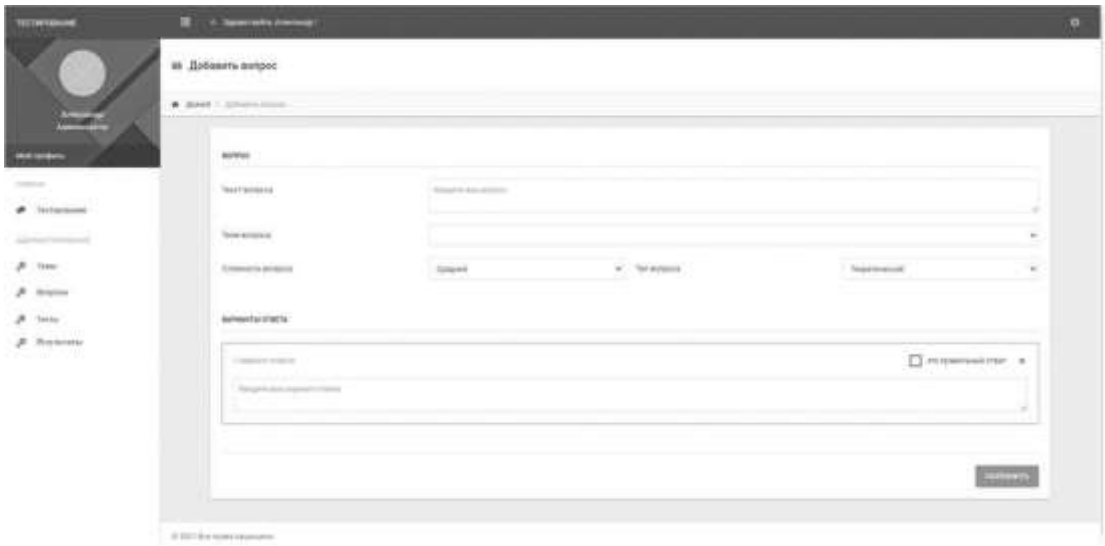


Рисунок 4.7 – Сторінка додавання завдання

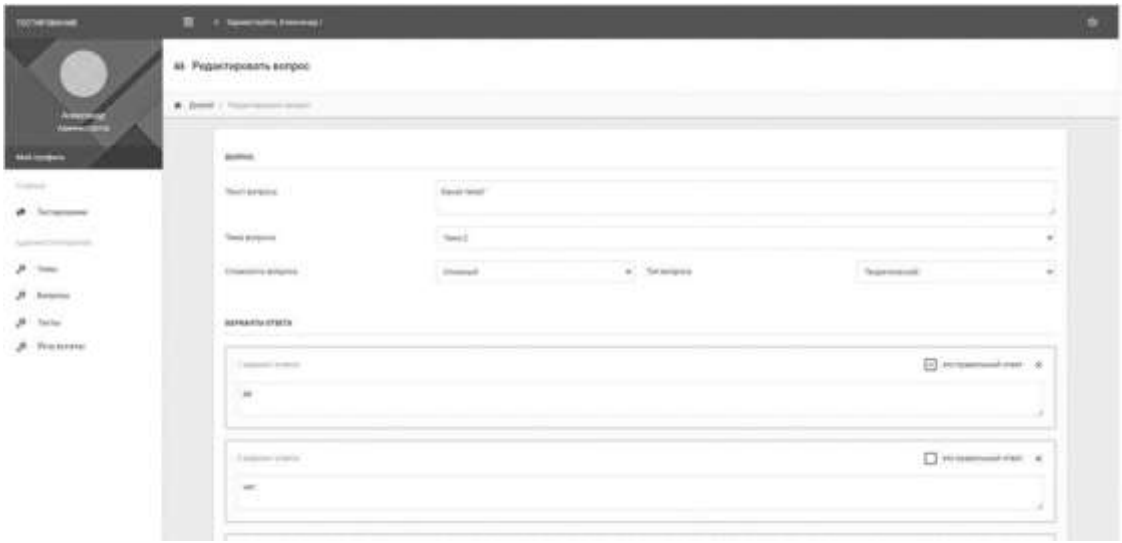


Рисунок 4.8 – Сторінка редагування завдання

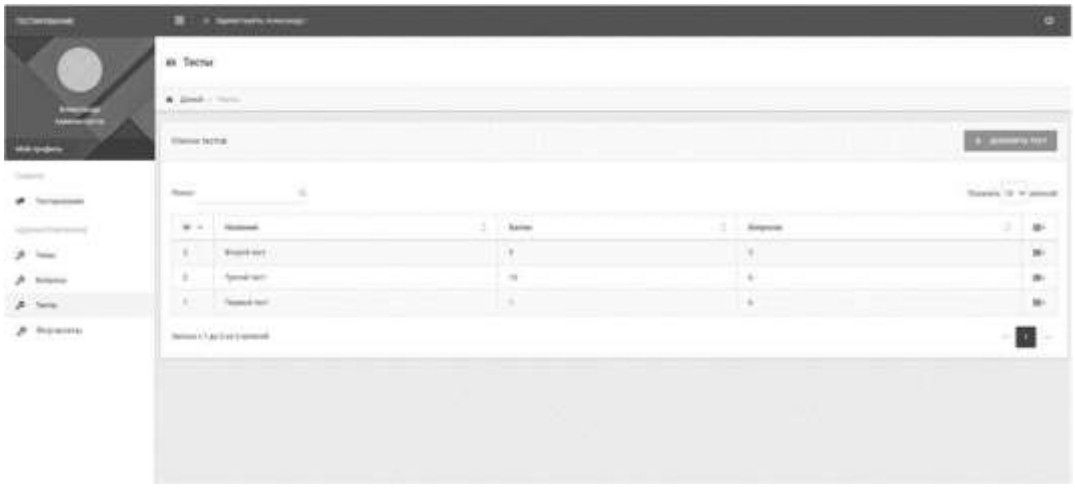


Рисунок 4.9 – Сторінка списку тестів

Якщо натиснути на вкладку «результати», після чого вибрати відповідний тест, то відкриється список користувачів, які пройшли даний тест. Далі потрібно вибрати певного користувача і відкриються результати проходження тесту як показано на рисунку 4.12. Можна побачити, що кожен тест має назву та свої налаштування, а саме: кількість питань кожного рівня складності, час проходження, а також співвідношення теоретичних та практичних питань і можливість керувати повтореннями тем завдань. Редагування, створення і видалення тесту працюють коректно. Також автоматизовано згенеровано тест. Генерація працює коректно, відповідаючи до налаштуванням.



Рисунок 4.10 – Сторінка додавання тесту



Рисунок 4.11 – Сторінка редагування тесту



Рисунок 4.12 – Сторінка результату проходження тесту певним користувачем

При переході на вкладку «тестування» відкриється сторінка із списком тестів, їх можна пройти. Сторінка списку тестів показана на рисунку 4.13. Після натискання кнопки «почати» почнеться виведення завдань та відповідей, тому користувачеві потрібно обирати відповідь, після чого відбудеться перехід до другого завдання, поки всі вони не закінчаться або не завершиться час даний на проходження.

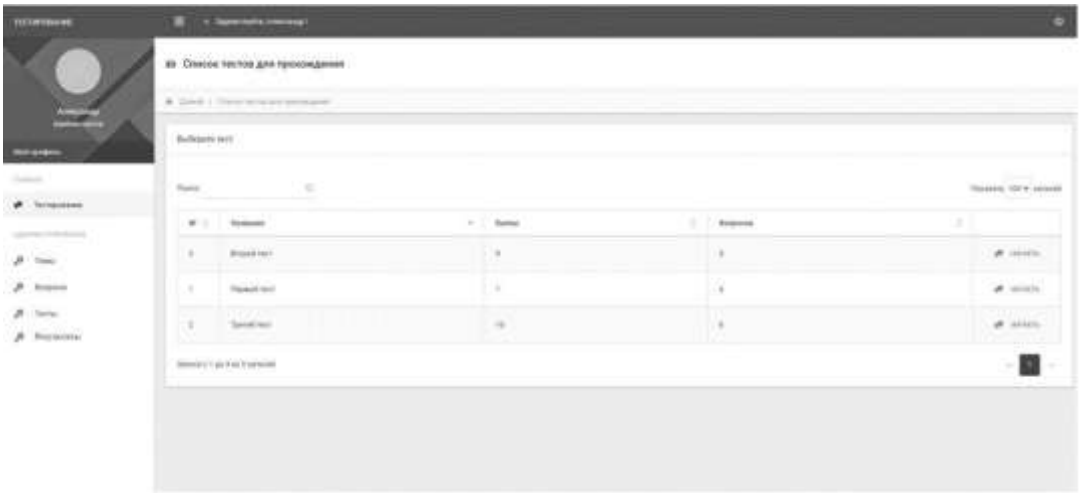


Рисунок 4.13 – Сторінка списку тестів для проходження



Рисунок 4.14 – Процес проходження тесту

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

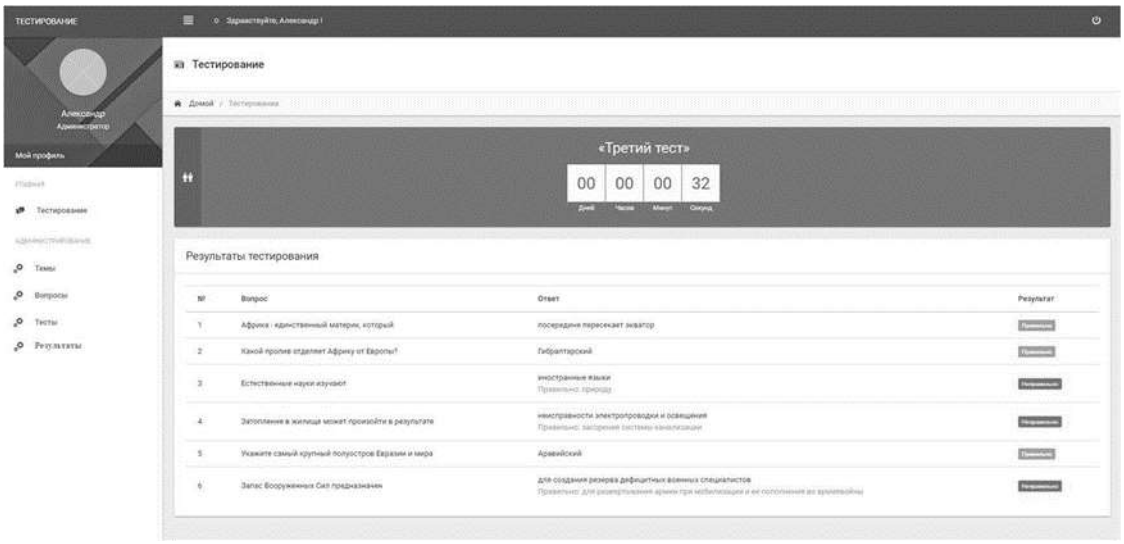


Рисунок 4.15 – Сторінка результатів проходження тесту

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

Таблиця 4.1 – Порівняння характеристик додатку для інтерактивного

тестування знань та додатку classmarker

	Можливість автоматизованої генерації тесту	Можливість керувати складністю завдань	Зміна співвідношення теоретичних та практичних завдань	Можливість керувати повторенням тем завдань в тесті
Додаток для інтерактивного тестування знань	+	+	+	+
classmarker	+	–	–	–

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. В.В. Божкова, Л.Ю. Сагер «Тестування як форма контролю знань: переваги та недоліки», Сучасні проблеми вищої освіти України в контексті інтеграції до європейського освітнього простору, с. 113-115.

2. С. І. Мединська «Тестування як засіб організації та реалізації диференціації навчання при викла данні іноземних мов у немовному ВНЗ», Вісник Дніпропетровського університету ім. Альфреда Нобеля, № 2 (2), с. 71-82.

3. Форми контролю знань. [Електронний ресурс]. Режим доступу: https://pidru4niki.com/1880041355088/pedagogika/formi_kontrolyu

4. Тест. [Електронний ресурс]. Режим доступу: https://ua.kursoviks.com.ua/naukovi-roboty/testi

5. Використання тестування. [Електронний ресурс]. Режим доступу: http://www.rusnauka.com/11_EISN_2008/Philologia/30455.doc.htm

6. Переваги тестування. [Електронний ресурс]. Режим доступу: https://vuzlit.com/1037136/perevagi_nedoliki_testuvannya

7. Недоліки тестування. [Електронний ресурс]. Режим доступу: http://www.edudirect.net/sopids-1556-1.html

8. Тестування як спосіб контролю знань. [Електронний ресурс]. Режим доступу: http://oldconf.neasmo.org.ua/node/767

9. Сучасні вимоги до тестування. [Електронний ресурс]. Режим доступу:https://knowledge.allbest.ru/pedagogics/2c0a65635a3ac68a5c53a98521216c27_0.html

10. Види моделей навчання. [Електронний ресурс]. Режим доступу: https://pidru4niki.com/10140508/psihologiya/modeli_navchannya

11. Пасивна модель навчання, переваги та недоліки. [Електронний ресурс]. Режим доступу: http://slovogray.blogspot.com/2016/02/blog-post-11.html

12. Активна модель навчання, переваги та недоліки. [Електронний ресурс]. Режим доступу: https://clincasequest.academy/learning-models/

13. Інтерактивна модель навчання, переваги та недоліки. [Електронний ресурс]. Режим доступу: https://psychologist.net.ua/interaktyvne-navchannya/

14. Інтерактивне тестування, переваги та недоліки. [Електронний ресурс]. Режим доступу: http://www.smartboard.com.ua/articles/130/

15. Методи та засоби автоматичної генерації тестових завдань. [Електронний ресурс]. Режим доступу: https://svitppt.com.ua/rizne/metodi-ta-zasobi-avtomatichnoi-generacii-testovih-zavdan.html

16. Метод параметричних тестів. [Електронний ресурс]. Режим доступу: http://dspace.wunu.edu.ua/bitstream /316497

17. Метод семантичних мереж. [Електронний ресурс]. Режим доступу: http://ki.lutsk-ntu.com.ua/node/118

18. Метод генерації тесту на основі понятійно-тезисної моделі. [Електронний ресурс]. Режим доступу: https://www.researchgate.net/publication/332470453_Avtomatizacii_postroenia_testovyh_zadanij_v_sistemah_distancionnogo_obucenia_na_osnove_ponatijno—tezisnoj_modeli

19. Переваги та недоліки веб-додатків. [Електронний ресурс]. Режим доступу: http://sites.znu.edu.ua/webprog/lect/1191.ukr.html

20. Classmarker. [Електронний ресурс]. Режим доступу: https://www.classmarker.com/online-testing/faq/

21. Online test pad. [Електронний ресурс]. Режим доступу: https://onlinetestpad.com

22. Learningapps. [Електронний ресурс]. Режим доступу: https://learningapps.org:

23. Atom. [Електронний ресурс]. Режим доступу: https://atom.io/

24. JavaScript, переваги та недоліки. [Електронний ресурс]. Режим доступу: https://developer.mozilla.org/ru/docs/Web/JavaScript

25. C#, переваги та недоліки. [Електронний ресурс]. Режим доступу: https://flexberry.github.io/ru/gbt_csharp.html

26. MySQL, переваги та популярність. [Електронний ресурс]. Режим доступу: https://www.mysql.com/

27. Фреймворк Vue js, переваги та недоліки. [Електронний ресурс]. Режим доступу: https://vuejs.org/

28. Реляційна база даних. [Електронний ресурс]. Режим доступу: https://sites.google.com/site/bazidanihsubdaccess/bazi-danih-subd/ierarhicna-model-danih/relacijna-model-baz-danih

29. Переваги реляційної бази даних. [Електронний ресурс]. Режим доступу: http://um.co.ua/8/8-16/8-164394.html