

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ

Кафедра програмної інженерії

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему: «**РОЗРОБКА WEB-ДОДАТКУ
ВИВЧЕННЯ СЛІВ АНГЛІЙСЬКОЇ МОВИ
НА БАЗІ ІГРОВОЇ МОДЕЛІ**»

Виконала: студентка 2 курсу, групи 8.1218

спеціальності 121 інженерія програмного забезпечення
(шифр і назва спеціальності)

освітньої програми інженерія програмного забезпечення
(назва освітньої програми)

Д.О. Авраменко

(ініціали та прізвище)

Керівник доцент кафедри фундаментальної
математики, к.ф.-м.н., Панасенко Є.В.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Рецензент доцент кафедри комп'ютерних наук,
доцент, к.т.н. Решевська К.С.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет математичний

Кафедра програмної інженерії

Рівень вищої освіти магістр

Спеціальність 121 інженерія програмного забезпечення

(шифр і назва)

Освітня програма інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри програмної
інженерії, к.ф.-м.н., доцент

Лісняк А.О.

(підпис)

« 09 » грудня 2020 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТЦІ

Авраменко Дар'ї Олександрівні

(прізвище, ім'я та по-батькові)

1. Тема роботи (проекту) «Розробка Web-додатку вивчення слів англійської мови на базі ігрової моделі»

керівник роботи (проекту) Панасенко Євген Валерійович, к.ф.-м.н., доцент

(прізвище, ім'я та по-батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від « 29 » травня 2019 року № 811-с

2. Строк подання студентом роботи _____

3. Вихідні дані до роботи 1. Постановка задачі.

2. Перелік літератури.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Постановка задачі.

2. Основні теоретичні відомості.

3. Проектування і реалізація

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____

Презентація

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.		
2.	Збір вихідних даних.		
3.	Обробка методичних та теоретичних джерел.		
4.	Розробка першого розділу.		
5.	Розробка другого розділу.		
6.	Оформлення та нормоконтроль кваліфікаційної роботи.	27.12.2019	
7.	Захист кваліфікаційної роботи.	09.12.2020	

Студент _____
(підпис)

Авраменко Д.О.
_____ (ініціали та прізвище)

Керівник роботи _____
(підпис)

Панасенко Є.В.
_____ (ініціали та прізвище)

Нормоконтроль пройдено

Нормоконтролер _____
(підпис)

О.В. Кудін
_____ (ініціали та прізвище)

РЕФЕРАТ

Кваліфікаційна робота магістра «Розробка Web-додатку вивчення слів англійської мови на базі ігрової моделі»: 63 с., 34 рис., 24 джерел, 2 додатки.

CMS, MYSQL, PHP, HTML, IP, SPA, XML, WEB-САЙТ, СИСТЕМА УПРАВЛІННЯ, ОС, БД, СУБД, ТЕСТИ, ЗОШ, ТЕСТ, ЗАГАДКИ, ГРА.

Об'єкт дослідження – технології проектування Web-додатків.

Мета роботи: розробити WEB-додаток з вивчення слів англійської мови на базі ігрової моделі.

Метод дослідження – аналітичний, порівняльний, методи моделювання, проектування, конструювання та тестування програмного забезпечення.

Для реалізації основної мети поставлені наступні задачі:

1. Розробка програмного продукту для CMS Drupal;
2. Створення бази даних на мові запитів MySQL;
3. Написання коду на мові програмування PHP, JS, HTML та CSS;
4. Розробити мінімалістичний дизайн;
5. Розроблений програмний продукт запуснути на сервері та протестувати на клієнті.

У роботі проведено аналіз проектування Web-додатків на базі ігрової моделі для процесу вивчення англійських слів. Представлено реалізацію Web-додатку вивчення слів англійської мови з використанням мов програмування PHP та JS для CMS Drupal.

SUMMARY

Master's Qualification Thesis "Development of a web application for learning English words on the basis of a game model": 63 pages., 34 figures, 24 references, 2 supplements.

CMS, MYSQL, PHP, HTML, IP, SPA, XML, WEB-SITE, CONTROL SYSTEM, OS, DB, DBMS, TESTS, ZOSH, TEST, MYSTERIES, GAME.

The object of the study is a software product that controls the work process by the administrator and the regular user.

The aim of the study is develop a WEB-application for learning English words based on the game model.

The methods of research are analytical, comparative, writing software product code.

To achieve the main goal the following tasks are set:

1. Software product development on CMS Drupal;
2. Creating a database in the language of MySQL queries;
3. Writing code in the programming language PHP, JS, HTML and CSS;
4. Develop a minimalist design;
5. Run the developed software on the server and test it on the client.

In the master's work it is planned to develop a practical part with the presentation of the software product in the form of a WEB-application for the process of learning English words from the English language in the form of a game model for the learning process. Write and draw up a theoretical part with an explanatory note.

ЗМІСТ

Завдання на магістерську роботу	2
Реферат	4
SUMMARY	5
ВСТУП	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ВИВЧЕННЯ ОСНОВНИХ ПРОБЛЕМ ПЕДАГОГІЧНОГО ТЕСТУВАННЯ	10
1.1 Тест	10
1.1.1 Педагогічне тестування	10
1.1.2 Тест IQ	17
1.2 Аналоги досліджуваної предметної області	18
1.3 Практичні властивості розробленої системи тест-загадок «Quiz»	20
1.4 Підхід до вирішення та постановка задач	22
1.5 Висновки до розділу 1	23
2 АНАЛІЗ ІНСТРУМЕНТАРІЮ РОЗРОБКИ ПРОГРАМНОГО ПРОДУКТУ МАГІСТЕРСЬКОЇ ДИПЛОМНОЇ РОБОТИ	24
2.1 Система управління Drupal	24
2.2 Мова програмування PHP	32
2.3 Open Server	33
2.4 PHPMyAdmin	36
2.5 Висновки до розділу 2	37
3 КЕРІВНИЦТВО КОРИСТУВАЧА ТА ОРГАНІЗАЦІЯ БАЗИ ДАННИХ	38
3.1 Керівництво користувача та можливості розробки	38
3.2 Модернізація бази даних розробленої системи	43
3.3 Опис програмного продукту та оцінка ефективності	54
3.4 Висновки до розділу 3	55

ВИСНОВКИ	56
ПЕРЕЛІК ПОСИЛАНЬ	57
Додаток А. Код WEB-додатку	59
Додаток В. Файл API	60

ВСТУП

Загадки – це своєрідний і дуже давній вид народної творчості. В сиву давнину загадкам надавали магічного значення, вони були пов'язані з культовими обрядами й звичаями, в них помітні сліди первісного світогляду. Вміле відгадування загадок вважалося ознакою розуму і щасливої вдачі. Загадки служили засобом випробовування мудрості, зрілості людини. Згодом вони втратили свою колишню функцію і використовувались як одна з форм культурної розваги. В передньому слові до збірки «350 загадок молодим і старим на забаву» (Коломия, 1911) її упорядник А. Онищук писав з цього приводу:

«Збереться де гурток людей — чи то на вечерницях у кого, на хрестинах або весіллю, а вже — біля веселих жартів і пісень — сиплють також загадочки одна за одною, наче з рукава. Один загадує, інші думають над розгадкою, дають відповіді, та не все влучні, а він, певний, що ніхто не відгадає, — жартує з присутніх... Але нараз хтось дотепний відгадає, а сам, відгадавши, новою загадкою, як кажуть, заб'є клина своєму попередникові та тощо, забава весело продовжується...»

Загадка вживається не лише для активізації пізнавальних можливостей дитини (та й дорослих), практикованих етно-педагогікою, чи гуртової розваги, а й почасти містить у собі філософський зміст («Ой, що росте без кореня, / А що біжить без поводу, / А що цвіте да без цвіту?»), пов'язується із міфологічним світобаченням: «Зоря-зірниця, красна дівиця, по небу гуляла, сльози роняла. Місяць бачив — не підняв, сонце встало і забрало» (роса). Загадки характеризуються яскравою пластичною образністю, багатством алітерацій та асонансів, вишуканою поетичністю:

Через Дунай глибокий,

Біг кінь білобокий,

Як упав, заіржав —

Білий світ задрожав (гроза, грім).

У наш час загадки продовжують своє широке побутування, в першу чергу, у спілкуванні українців, але також почасти перейшовши у сферу ЗМІ, в тому числі і електронних.

Іншими словами загадка - це інтелектуальна гра, в яку грає увесь світ, і ось чому б в неї не пограти у вигляді тесту з певною складністю завдань, реалізований в єдиний програмний продукт.

В даний час в багатьох ВНЗ багато використовуються системи оцінки знань студентів, які спрямовані на підвищення ефективності процесу навчання. Для отримання рейтингової оцінки необхідно пройти кілька етапів контролю (поточний, проміжний, рубіжний, підсумковий) [1].

Актуальною буде створити програмний продукт, а саме комп'ютерне тестування, що дозволе дати об'єктивну оцінку знань не тільки студентів та і звичайного школяра на різну тематику, при цьому знизити неоднорідність пропонованих вимог і підвищити продуктивність праці викладача та вчителя в школі.

Комп'ютерна система тестування, звичайно, дуже зручна для застосування, в процесі навчання. Для перевірки практичних умінь і навичок школяра використовується завдання, тобто тестове завдання у вигляді загадок, для відповіді на який необхідно провести певні інтелектуальні дії, пов'язані з вирішенням завдання.

Крім того, при проведенні тестування знань, як дистанційно, так і локально, важливим є забезпечення рівня безпеки (авторизація, захист бази тестів «загадок» і тощо), а також організація роботи зі школярами, так і з вчителями.

Таким чином практична цінність магістерської дипломної роботи, спрямована на розробку нових способів тестування знань у вигляді саме загадок та розробку на їх основі системи тестування знань, є дуже актуальною.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ВИВЧЕННЯ ОСНОВНИХ ПРОБЛЕМ ПЕДАГОГІЧНОГО ТЕСТУВАННЯ

1.1 Тест

Тест або випробування - спосіб вивчення глибинних процесів діяльності системи, за допомогою приміщення системи в різні ситуації і відстеження доступних спостереженню змін в ній.

1.1.1 Педагогічне тестування

Педагогічне тестування - це форма вимірювання знань учнів, заснована на застосуванні педагогічних тестів. Включає в себе підготовку якісних тестів, власне проведення тестування і подальшу обробку результатів, яка дає оцінку навченості тестованих.

Педагогічний тест - це інструмент оцінювання навченості учнів, що складається з системи тестових завдань, стандартизованої процедури проведення, обробки та аналізу результатів.

Тести можна класифікувати за різними ознаками:

1. За програмними цілями - інформаційні, діагностичні, навчальні, мотиваційні, атестаційні;
2. За процедурою створення – стандартизовані та не стандартизовані;
3. За способом формування завдань - детерміновані, стохастичні, динамічні;
4. За технологією проведення - паперові, в тому числі паперові з використанням оптичного розпізнавання, приклад показано на Рис. 1.1; натурні, з використанням спеціальної апаратури, комп'ютерні;

5. За формою завдань - закритого типу, відкритого типу, встановлення відповідності, упорядкування послідовності;

6. По наявності зворотного зв'язку - традиційні і адаптивні.



Рис. 1.1 – Приклад тесту – ЗНО

Тести поділяються на два види: традиційні та адаптивні,

Традиційний тест містить список питань і різні варіанти відповідей. Кожне питання оцінюється в певну кількість балів. Результат традиційного тесту залежить від кількості питань, на які було дано правильну відповідь. На думку Аванесова В. С. традиційний тест - система завдань, що пред'являється в порядку збільшення складності в один і той же час, з однаковою системою оцінювання для всіх тестованих [2].

Адаптивний тест - особливий вид тесту, в якому кожне наступне завдання вибирається залежно від відповідей на попередні завдання. Послідовність завдань і їх кількість в такому вигляді тесту визначається динамічно. Самими значущими перевагами комп'ютерного адаптивного тестування перед традиційним є:

1. Можливість адаптації під рівень знань тестованого (не доведеться відповідати на занадто складні або занадто прості питання);

2. Економія часу і сил за рахунок скорочення кількості завдань (довжина тесту може бути зменшена до 60%) без втрати рівня достовірності.

Завдання з вибором відповідей (закриті завдання, завдання з вибором однієї правильної відповіді), рисунок 1.2:

При наборі тексту слова відокремлюються один від одного ...

- а) двокрапкою;
- б) комою;
- в) прогалиною;
- г) точкою.

Рис. 1.2 – Вид тесту де декілька відповідей правильно

Завдання з вибором однієї неправильної відповіді, показано на рисунку 1.3:

Операція не має ознаки, за якою підібрані інші операції, представлені в списку ...

- а) збереження тексту;
- б) форматування тексту;
- в) видалення фрагмента тексту;
- г) переміщення фрагмента тексту;
- д) копіювання фрагмента тексту.

Рис. 1.3 – Вид тесту на вибір неправильної відповіді

Завдання на встановлення відповідності, методикою підстановки або порівняння, приклад показано на рисунку 1.4.

Установіть відповідність між командами і сполученнями клавіш.

команда

сполучення клавіш

1. Вирізати фрагмент тексту;	а) CTRL + X;
2. Копіювати фрагмент тексту;	б) CTRL + C;
3. Вставити фрагмент тексту.	в) CTRL + V.

Рис. 1.4 – Вид вибіркового завдання

Завдання з вибором кількох правильних відповідей, рисунок 1.5.

Використання сліпого десятипальцевого методу веде до ...

- а) зниження напруги на пальці;
- б) зменшення швидкості друку;
- в) зменшення кількості помилок і помилок;
- г) швидкої стомлюваності пальців.

Рис. 1.5 – Вид тесту з вибором кількох правильних відповідей

Упорядкування послідовності, у хронологічному порядку, зображено на рисунку 1.6.

Розмістіть в хронологічному порядку

- iii. Бородінська битва
- i. Льодове побоїще
- ii. Куликовська битва

Рис. 1.6 – Вид тесту з хронологічним порядком

Завдання з відкритою відповіддю, з виконанням практичного друкування, показано на рисунку 1.7.

Існує два способи освоєння клавіатури при друкуванні сліпим десятипальцевим методом:

1. соло на клавіатурі ergosolo.ua _____

2. stamina _____

Рис. 1.7 – Вид тесту з друкуванням

Тестове завдання - складова частина педагогічного тесту, що відповідає вимогам технологічності, форми, змісту, зі статистичними вимогам:

1. Відома складність;
2. Достатньої варіації тестових балів;
3. Позитивної кореляцією балів завдання з балами по всьому тесту.

Типи завдань у тесті, бувають двох типів: закриті та відкриті.

Закриті:

1. Завдання альтернативних відповідей;
2. Завдання множинного вибору;
3. Завдання на відновлення відповідності;
4. Завдання на встановлення правильної послідовності.

Відкриті:

1. Завдання вільного викладу;
2. Завдання-доповнення.

Тестування в педагогіці виконує три основні взаємопов'язані функції: діагностичну, навчальну і виховну:

1. Діагностична функція полягає у виявленні рівня знань, умінь, навичок учня. Це основна і найочевидніша функція тестування. За об'єктивності, широті і швидкості діагностування, тестування перевершує всі інші форми педагогічного контролю;

2. Навчальна функція тестування полягає в мотивуванні учня до активізації роботи по засвоєнню навчального матеріалу. Для посилення

навчальної функції тестування можуть бути використані додаткові заходи стимулювання студентів, такі як: роздача викладачем примірних переліку питань для самостійної підготовки, наявність в самому тесті навідних запитань і підказок, спільний розбір результатів тесту;

3. Виховна функція проявляється в періодичності та неминучості тестового контролю. Це дисциплінує, організовує і направляє діяльність учнів, допомагає виявити і усунути прогалини в знаннях, формує прагнення розвинути свої здібності.

У порівнянні з іншими формами контролю знань тестування має свої переваги і недоліки. Переваги:

1. Тестування є більш якісним і об'єктивним способом оцінювання, його об'єктивність досягається шляхом стандартизації процедури проведення, перевірки показників якості завдань і тестів цілком.

2. Тестування - більш справедливий метод, воно ставить всіх учнів в рівні умови, як в процесі контролю, так і в процесі оцінки, практично виключаючи суб'єктивізм викладача. За даними Британської асоціації NEAB, що займається підсумковою атестацією учнів Великобританії, тестування дозволяє знизити кількість апеляцій більш ніж в три рази, зробити процедуру оцінювання однаковою для всіх учнів незалежно від місця проживання, типу і виду освітньої установи, в якому займаються учні;

3. Тести - це більш об'ємний інструмент, оскільки тестування може включати в себе завдання з усіх тем курсу, в той час як на усний іспит зазвичай виноситься 2-4 теми, а на письмовий - 3-5. Це дозволяє виявити знання учня по всьому курсу, виключивши елемент випадковості при витягуванні квитка. За допомогою тестування можна встановити рівень знань учня по предмету в цілому і по окремих його розділах;

4. Тест - це більш точний інструмент, так, наприклад, шкала оцінювання тесту з 20 питань, складається з 20 ділень, в той час, як звичайна шкала оцінки знань - тільки з чотирьох;

5. Тестування більш ефективно з економічної точки зору. Основні витрати при тестуванні припадають на розробку якісного інструментарію, тобто мають разовий характер. Витрати ж на проведення тесту значно нижче, ніж при письмовому або усному контролі. Проведення тестування і контроль результатів в групі з 30 чоловік займає півтори-дві години, усний або письмовий іспит - не менше чотирьох годин;

6. Тестування - це більш м'який інструмент, вони ставлять всіх учнів в рівні умови, використовуючи єдину процедуру і єдині критерії оцінки, що призводить до зниження передекзаменаційні нервових напружень.

Недоліки: 1. Розробка якісного тестового інструментарію - тривалий, трудомісткий і дорогий процес. Стандартні набори тестів для більшості дисциплін ще не розроблені, а розроблені зазвичай мають дуже низьку якість;

2. Дані, одержувані викладачем в результаті тестування, хоча і включають в себе інформацію про прогалини в знаннях по конкретних розділах, але не дозволяють судити про причини цих прогалин;

3. Тест не дозволяє перевіряти і оцінювати високі, продуктивні рівні знань, пов'язані з творчістю, тобто імовірнісні, абстрактні і методологічні знання;

4. Широта охоплення тем в тестуванні має і зворотну сторону. Учень при тестуванні, на відміну від усного або письмового іспиту, не має достатньо часу для скільки-небудь глибокого аналізу теми [3];

5. Забезпечення об'єктивності і справедливості тесту вимагає прийняття спеціальних заходів щодо забезпечення конфіденційності тестових завдань. При повторному застосуванні тесту бажано внесення в завдання змін;

6. У тестуванні присутній елемент випадковості. Наприклад, учень, що не відповів на просте питання, може дати правильну відповідь на більш складний. Причиною цього може бути, як випадкова помилка в першому питанні, так і вгадування відповіді у другому. Це спотворює результати тесту і призводить до необхідності врахування ймовірнісної складової при їх аналізі.

1.1.2 Тест IQ

Коефіцієнт інтелекту - кількісна оцінка рівня інтелекту людини (коефіцієнт розумового розвитку): рівень інтелекту щодо рівня інтелекту середньостатистичної людини (такого ж або середнього віку); в більш вузькому сенсі - відношення так званого розумового віку до істинного хронологічним віком даної особи (індивіда). Визначається за допомогою спеціальних тестів (таких як тест Айзенка). Коефіцієнт інтелекту є спробою оцінки чинника загального інтелекту (g).

Тести IQ розробляються так, щоб результати описувалися нормальним розподілом з середнім значенням IQ, рівним 100, і таким розкидом, щоб 50% людей мали IQ між 85 і 115 і по 25% - нижче 90 і вище 110, приклад показано на рисунку 1.8. Значення IQ менше 70 часто кваліфікується, як розумова відсталість.

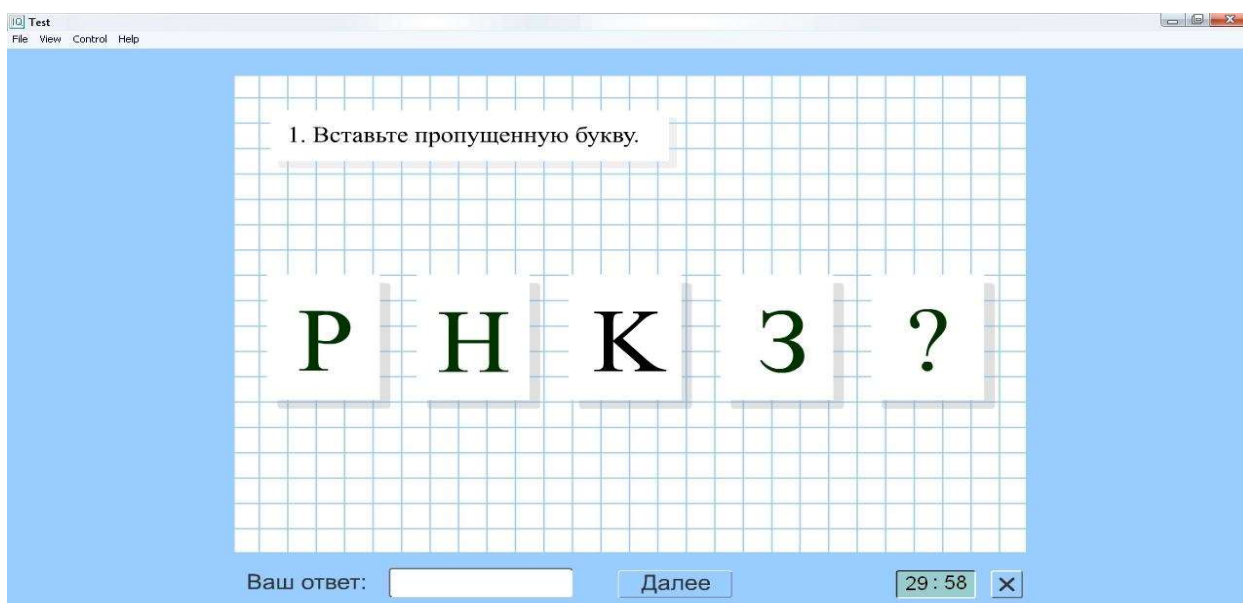


Рис. 1.8 – Завдання тесту IQ

Розроблено також три спеціальних теста: словесний, числовий і зорово-просторовий. Ці тести спрямовані на виявлення специфічних здібностей, які неможливо визначити за допомогою загальних тестів. Також в збірники тестів як правило включається декілька додаткових розважальних завдань під

жартівливою назвою «головоломки для інтелектуалів» (або «розминка для інтелектуалів» в деяких перекладах). З огляду на підвищеної складності цих завдань, їх результати не підлягають кількісній оцінці, а самі завдання додаються виключно «заради задоволення пошукати рішення».

Тести призначені для оцінки інтелектуальних здібностей для людей у віці від 18 до 60 років, які мають освіту не нижче середнього. Коефіцієнт інтелекту - кількісна оцінка рівня інтелекту людини: рівень інтелекту щодо рівня інтелекту середньостатистичної людини такого ж віку. Визначається за допомогою спеціальних тестів. Тести IQ розраховані на оцінку розумових здібностей, а не рівня знань (ерудованості). Коефіцієнт інтелекту є спробою оцінки чинника загального інтелекту (g).

Тести IQ розробляються так, щоб результати описувалися нормальним розподілом з середнім значенням IQ, рівним 100 і таким розкидом, щоб 50% людей мали IQ між 90 і 110 і по 25% - нижче 90 і вище 110. Значення IQ менше 70 зазвичай інтерпретується як розумова відсталість. Тест Айзенка передбачає максимальний рівень IQ 160 балів.

1.2 Аналоги досліджуваної предметної області

Подібних систем на ринку достатньо, тому переглянемо тільки саму популярну – Moodle, рисунок 1.9.

Moodle - система управління курсами (електронне навчання), також відома як система управління навчанням або віртуальне навчальне середовище є аббревіатурою від Modular Object-Oriented Dynamic Learning Environment (модульне об'єктно-орієнтоване динамічне навчальне середовище). Являє собою вільний (розповсюджується за ліцензією GNU GPL) веб-додаток, що надає можливість створювати сайти для онлайн-навчання. Перша версія написана 20 серпня 2002 року.



Рис. 1.9 – Вид логотипу системи зрізу знань з електронного навчання «Moodle»

Ejudge – це система для проведення різних заходів, в яких необхідна автоматична перевірка програм. Система може застосовуватися для проведення олімпіад і підтримки навчальних курсів. Інтерфейс учасника представлений на рисунку 1.10 [3].



Рис. 1.10 – Вид тесту «Ejudge»

Переглядаються деякі недоліки, а саме:

1. Не зручний турнірний та адміністраторський інтерфейс;
2. Не зручна установка програмного продукту та настройка.

Codewars – це інтерактивний збірник задач з інформатики, вирішуючи певні завдання. Стартова сторінка даного збірника задач виглядає досить таки просто і легко використовується, показано на рисунку 1.11.

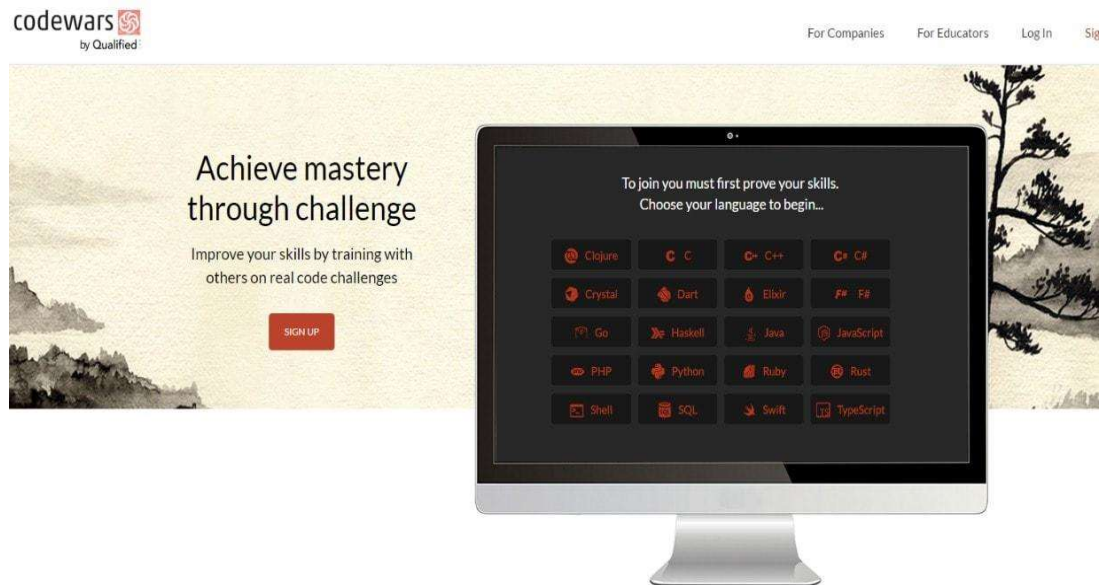


Рис. 1.11 – Вид тест (збірника) «Codewars»

1.3 Практичні властивості розробленої системи тест-загадок «Quiz»

Тест-загадки «Quiz» - багатоцільова система інтерактивного тестування, розроблена для автоматизації контролю знань. Тест-загадки «Quiz» - відрізняється від аналогічних систем перш за все легкістю при налаштуванні і використанні, дозволяє автоматизувати процес перевірки питань-завдань.

Основна область застосування тест-загадки «Quiz», підходить до наступного переліку:

1. Для керівників відділів розробки – тестування рівня знань інженерів в відповідності предметної області, підтримка його відповідно до останніх технологій, що дозволяє розробляти проекти на високому якісному рівні;
2. При прийомі фахівців на роботу – перевірка спеціалізації майбутніх працівників;

3. У вищих і середніх навчальних закладах – автоматизоване проведення контрольних опитувань, заліків та іспитів, а також тестування компетенції учнів з профілюючих предметів, що дає їм можливість більш глибокого розуміння індустріальних стандартів, знання яких будуть необхідні їм в майбутньому.

Основні можливості:

1. Створення бази питань по будь-якій темі без обмеження на різний обсяг;
2. Використання чотирьох основних типів питань: однозначну відповідь, коротка відповідь, довга відповідь, ЕСС;
3. Додаткові типи питань, з можливістю самому створювати розділи;
4. Досить гнучкі настройки по обмеженню часу, кількості підказок, допущених помилок, а також за оцінкою відповідей користувача;
5. Настроюється система підрахунку балів при проведенні опитування;
6. Захист бази питань від несанкціонованого доступу і зміни;
7. Задання повного звіту по проведеним опитуванням і розміщення його тільки на сервері.

Діаграма прецедентів даного програмного продукту представлена на рисунку 1.12.

Даний програмний продукт призначений для проходження тестування онлайн. Там виділено 2 типу користувачів, які можуть працювати в системі. Рядовий користувач та адміністратор. Рядовий користувач має початковий рівень підготовки, адміністратор знайомий з системою тест-загадок «Quiz» та управляє нею.

Адміністратор даного WEB-сайту має певні права, які представлені на рисунку 1.12 у вигляді прецедентів:

1. Управління питаннями;
2. Управління користувачами;
3. Управління результатами.

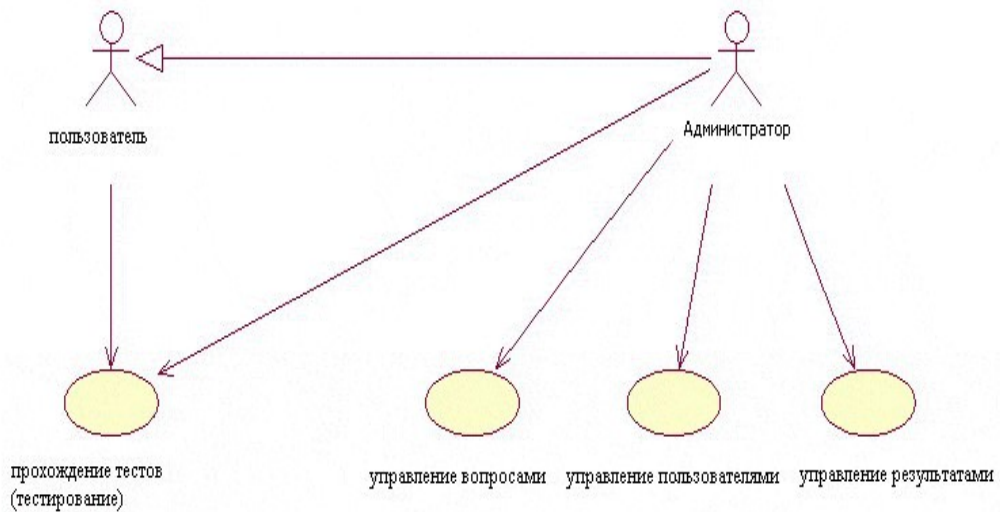


Рис. 1.12 – Вид діаграми прецедентів

1.4 Підхід до вирішення та постановка задач

Необхідно розробити WEB-додаток з вивчення слів англійської мови на базі ігрової моделі. Отже WEB-додаток буде під назвою: «Quiz», з представленням певної ігрової форми: Відгадай-загадку у вигляді тесту. Тестування може проходити учні починаючи з початкової школи до випускних класів. Майбутній функціонал планується досягти наступні можливості:

1. Створення тесту з можливостями сортування питань у випадковому порядку за бажанням викладача, відображення результатів і правильних відповідей, виведення оцінки по завершенню тестування;
2. Розрахунок оцінки відбувається в залежності від правильності і ємкості відповіді, з можливістю оцінювання не тільки ПК, але з обов'язковим оцінюванням вчителя;
3. Створення питання з вибором кількості правильних варіантів відповіді. З можливістю розміщення відповідей у випадковому порядку і з можливістю додавання даного питання в уже існуючий тест;
4. Створення питання з рукописною відповіддю. Вчитель сам перевіряє відповідь школяра;

5. Використання групової політики доступу для забезпечення безпеки: школяр, незареєстрований користувач, адміністратор і вчитель;
6. Призначення максимальної кількості балів за кожну повністю вірну відповідь;
7. Написання коду на мові програмування PHP, JS, HTML та CSS;
8. Розробити мінімалістичний дизайн;
9. Розробити та модернізувати базу даних;
10. Захист бази питань від несанкціонованого доступу і зміни;
11. Розроблений програмний продукт запустити на сервері та протестувати на клієнті.

1.5 Висновки до розділу 1

Тестування - це форма вимірювання знань учнів, заснована на застосуванні педагогічних тестів. Включає в себе підготовку якісних тестів, власне проведення тестування і подальшу обробку результатів, яка дає оцінку навченості тестованих.

Загадка - (від слова «гадати» — думати, вгадувати, загадувати) — жанр усної народної творчості, дотепне запитання, часто у віршованій формі; є в фольклорі всіх народів світу.

2 АНАЛІЗ ІНСТРУМЕНТАРІЮ РОЗРОБКИ ПРОГРАМНОГО ПРОДУКТУ МАГІСТЕРСЬКОЇ РОБОТИ

2.1 Система управління Drupal

Drupal - система управління вмістом (CMS), логотип показано на рисунку 3.1, яка використовується також як каркас для веб-додатків (CMF), написана на мові PHP і використовує як сховище даних реляційну базу даних (підтримуються MySQL , PostgreSQL та інші). Drupal є вільним програмним забезпеченням, захищеним ліцензією GPL, і розвивається зусиллями ентузіастів зі всього світу.

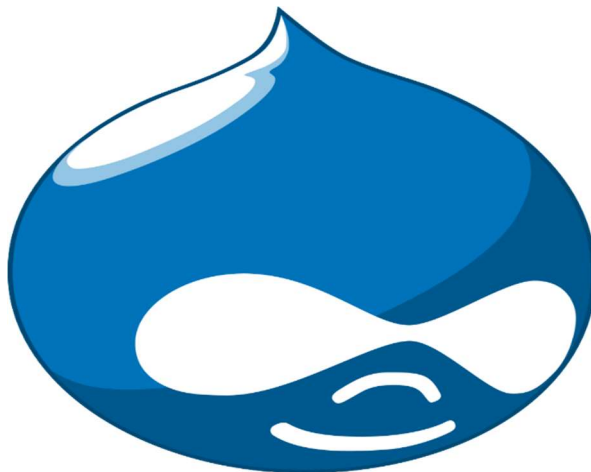


Рис. 2.1 – Логотип CMS Drupal

Почав розробку 2000 року бельгієць Дріс Бейтарт, який і понині є керівником проекту. Назва drupal є спотвореною вимовою голландського слова druppel (крапля) і з'явилася в результаті забавної помилки. У 2000 році кілька студентів Антверпенського університету використовували загальний ADSL-модем для доступу в інтернет, і для спілкування користувачів цієї маленької локальної мережі Дріс створив внутрішній сайт. Програма, на якій працював сайт, не мала імені, поки Дріс не зібрався реєструвати доменне ім'я

для нового сайту, і в результаті помилки замість dorp.org (від голландського dorp - село, цілком доречне ім'я для невеликого співтовариства) був зареєстрований drop.org. Аудиторія нового сайту з часом змінилася на які цікавляться розвитком нових веб-технологій, і новий сайт поступово перейшов з любительського експерименту в проект з обговорень ідей веб-програмування. У 2001 році Дріс опублікував вихідний код drop.org під ім'ям «Drupal». З тих пір розробка перемістилася на зареєстрований пізніше сайт drupal.org, який набув статусу офіційного сайту проекту.

З часу створення в процес розробки були залучені сотні людей. Офіційні розробники ядра зараз - це десятки людей, але багато людей надсилають свої патчі, що включаються в код системи. Також функціональність движка розширюється додатковими модулями, які створюють сторонні розробники. Наприклад, в релізі 4.6 взяли участь 50 осіб, що надіслали 338 поправок до коду, а в релізі 4.7 взяли участь вже 338 розробників, що надіслали приблизно 1500 поправок. За п'ять років існування проекту (до 2006 року) вийшло 13 релізів. На даний момент було розроблено понад 7000 додаткових модулів, що розширюють базові функції CMS. За статистикою звернення до оновлень движка, на березень 2009 року в інтернеті - в межах 120 тис. Активних сайтів на Drupal 6 (без урахування сайтів, що знаходяться за файрволлами, сайтів з відключеним механізмом оновлень). Це також не враховує сайтів, що працюють на ранніх версіях Drupal.

У 2007 році Дріс Бейтарт створив фірму Asquia , спрямовану на комерційну підтримку проектів і сервісів, заснованих на Drupal, натхненний успіхами Canonical і RedHat , що побудували бізнес на підтримку відкритих проектів. У 2008 Бейтарт отримав докторський ступінь в галузі інформатики та приладобудування («Computer Science and Engineering»).

Згодом навколо міжнародного сайту з'явилися національні спільноти, функціонують понад 20 національних сайтів підтримки на різних мовах.

Емблема проекту. Після того, як Drupal був створений, очевидним питанням став вибір і створення емблеми проекту. Учасники спільноти

вважали, що вона повинна бути пов'язана з краплею або водою. Спочатку емблема Drupal зображувалася як крапля в колі, з часом прийшла ідея зробити емблему у вигляді карикатурною краплі з особою. Стівен Віттенс створив тривимірну краплю, але ідея не пішла в маси головним чином тому, що тривимірний об'єкт важко друкувати, редагувати і т. д.

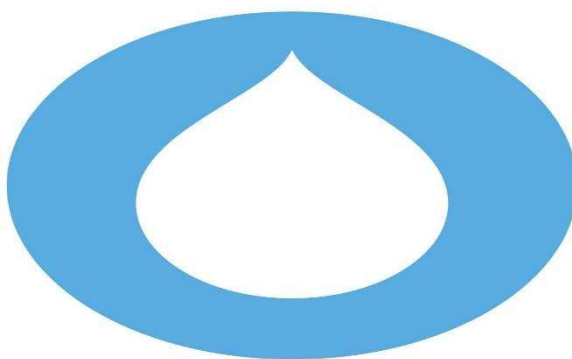


Рис. 2.2 – Перший логотип

Поточна емблема системи придумана Крістьяном Янсенем, вона являє собою краплю, яка зображує обличчя, очі показуються як дві з'єднані краплі у вигляді знаку нескінченності і висловлюють в символічній формі нескінченні можливості.

Нагороди:

1. Тричі - у 2007, 2008 і 2009 роках - система отримала нагороду Webware 100 (присуджується кращим на сьогодні сайтам, сервісам і додаткам Мережі);
2. Друпал - абсолютний переможець Packt Open Source CMS в 2008 р і переможець в категорії PHP в 2008 і 2009 рр. Основним суперником Drupal була «Joomla!». Переможці вибиралися журі з експертів у своїй галузі, а також відвідувачами сайту Packt;
3. Переможець конкурсу «Best open source applications 2010» в категорії «Краща система управління web-контентом»

Підтримка локалізації, представлено на рисунку 2.3. Система підтримує локалізацію інтерфейсу на більш ніж 100 мовами (проте не всі переклади є повними). Є підтримка російської мови. В Drupal використовується власний

механізм для локалізації зі зберіганням перекладів в базі даних, нарівні з рештою вмісту сайту. Імпорт і експорт перекладів сайту здійснюється у вигляді ро-файлів (формат використовується бібліотекою gettext).

Drupal™

Download & Extend

Drupal Core Distributions Modules Themes

Localization update

[View](#) [Version control](#) [Automated testing](#)

By Sutharsan on 12 January 2010, updated 18 September 2017

Automatically downloads and updates your translations by fetching them from localize.drupal.org or any other Localization server.

This module is integrated into Drupal 8 core.

Improve efficiency

To improve the efficiency of translation updates, I advise to exclude custom modules and features from update checks. Configure *Disable update* at `/admin/config/regional/language/update`.

Recommended modules

- cURL HTTP Request:** Use this module if your site is behind a proxy or if you experience 'allow_url_fopen' problems.
- Git deploy:** Use this module if you use git checkouts instead of package tarballs or zip files. The projects and versions will be identified properly.
- Localization client:** Provides on page translations that can be automatically submitted to the translation server.
- Localized Drupal:** An install profile that will download translations automatically.

Project information

Module categories: [Multilingual](#)

180,746 sites report using this module
1,693,184 downloads

Stable releases for this project are covered by the [security advisory policy](#).
Look for the shield icon below.

14

Maintainers for Localization update

Sutharsan - 169 commits
last: 2 months ago, first: 9 years ago

Gábor Hojtsy - 25 commits
last: 8 years ago, first: 9 years ago

Jose Reyero - 23 commits
last: 9 years ago, first: 9 years ago

[View all committers](#)
[View commits](#)

Issues for Localization update

To avoid duplicates, please search before submitting a new issue.

Advanced search
All issues
21 open, 365 total
Bug report
7 open, 207 total
Statistics

New issues 0

Рис. 2.3 – Вид проходження та завантаження локалізацій

Повна локалізація вмісту підтримується сторонніми модулями, які не входять до поставки Drupal, але можуть бути завантажені з сайту розробників.

Методи структуризації вмісту. В Drupal пропонується гнучка схема організації структури сайту на основі таксономії. Таксономія - механізм, що дозволяє створювати будь-яку кількість тематичних категорій для вмісту сайту і асоціювати їх з модулями, що забезпечують введення і виведення інформації. Категорії можуть представляти плоскі або ієрархічні списки, або складні структури, де елемент може мати декілька «батьків» і кілька дочірніх елементів. За допомогою подібної схеми одними і тими ж модулями можлива організація різних варіантів структуризації вмісту. Наприклад, легко

створюється наскрізний список «ключових слів» для всіх документів WEB-сайту і т. д.

Таксономія o

Таксономія дозволяє класифікувати контент. В таксономії використовуються словари і терміни, які входять до слівників. Наприклад, словник, який називається «Фрукти», може містити терміни «Яблуко», «Апельсин» і «Банан».

[+ Додати словник](#)

[Показати вагу рядка](#)

ІМЯ СЛОВАРЯ	ДЕЙСТВИЯ		
+ Разделы	змінити словник	список термінів	додати терміни
+ Теги	змінити словник	список термінів	додати терміни

[Зберегти](#)

Рис. 2.4 – Приклад таксономії

Інша парадигма з'явилася зі створенням в Drupal розширення Content Construction Kit (ССК). ССК дозволяє доповнювати документи новими полями різних типів - від полів введення URL і email, до полів зберігання і відображення мультимедійних файлів. Також за допомогою додаткових модулів до ССК (наприклад Node reference) можна організувати зв'язку між документами, не використовуючи механізм таксономії.

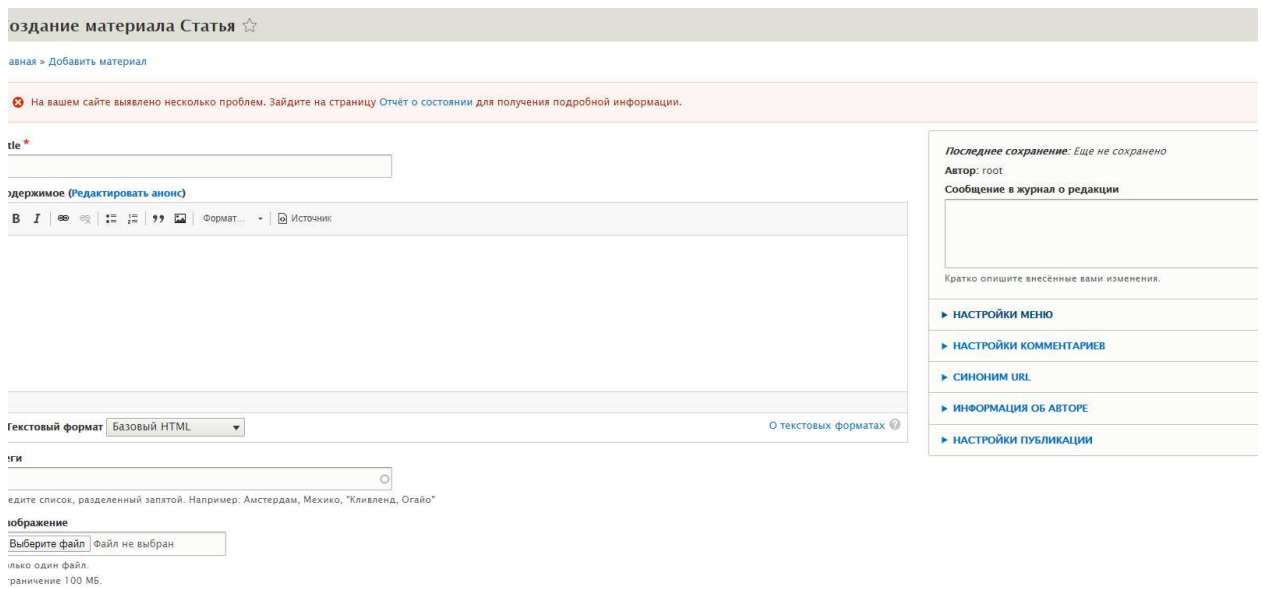


Рис. 2.5 – Створення контенту

Механізми розширення функціональності. Drupal має модульну архітектуру з компактним ядром, які надають API, до якого можуть звертатися модулі. Стандартний набір модулів включає такі функції, як новинна стрічка, блог, форум, завантаження файлів, збирач новин, голосування, пошук і інші. Дизайн сайту змінюється також за допомогою спеціальних модулів - «тем оформлення».

Теми оформлення. Дизайн сайту на Drupal можна міняти за допомогою тем оформлень. Як шаблонізатора («движок теми» в термінології Drupal) використовується PHPTemplate, проте є можливість використовувати Xtemplate, Smarty, Twig, рисунок 2.6 і інші.



Рис. 2.6 – Логотип шаблонізатора Twig

Починаючи з версії 4.7 Drupal підтримує технологію Ајах, показано на рисунку 2.7 для динамічної підвантаження вмісту без повного оновлення сторінок. У версії 5 для роботи з JavaScript додана бібліотека JQuery, рисунок 2.8. С версії 6.0 з'явилася можливість динамічно змінювати форми за допомогою АНАН. У версію 7.0 увійшла бібліотека jQuery UI.



Рис. 2.7 – Логотип Ајах



Рис. 2.8 – Логотип JQuery

Системні вимоги. Drupal в першу чергу орієнтований на платформу LAMP, схема показана на рисунк 2.9, але можливі варіанти використання іншого системного програмного забезпечення.



Рис. 2.9 – Вид схеми LAMP

1. Операційна система: сімейство Unix (в тому числі FreeBSD, Mac OS X), Windows (будь-які ОС , на яких може працювати інтерпретатор PHP);
2. Веб-сервер: Apache, Nginx, Lighttpd, IIS Versions 6, 7 (через Fast CGI) (теоретично будь-який веб-сервер, що допускає виконання PHP-коду);
3. СУБД : MySQL, PostgreSQL (в принципі нескладна адаптація до інших реляційних СУБД, наприклад IBM DB2);
4. PHP версії 4.3.3 і вище (для сумісності з усіма сторонніми модулями рекомендується застосовувати PHP 5, замість PHP 4). Drupal 7 вимагає PHP 5.2.4 і вище.

Недоліки. Критики Drupal ставлять в докір розробникам слабе використання об'єктних можливостей PHP. Дійсно, API Drupal практично не використовує наявні в PHP можливості ООП. Розробники аргументують це слабкою реалізацією ООП в мові (особливо до версії PHP 5). Об'єктна модель в Drupal присутній, але в дещо нетрадиційному для PHP вигляді. В Drupal 7 проте присутній ряд нововведень, що задіюють об'єктні можливості PHP 5. А у версії Drupal 8 офіційно прийнято використання ООП. Однак, в співтоваристві присутні вельми суперечливі думки з даного приводу.

До недоліків (але одночасно і до достоїнств) Drupal можна віднести відсутність зворотної сумісності API. У кожному новому великому релізі відбуваються великі зміни API, коли поряд з додаванням нових функцій прибираються деякі старі або змінюються параметри виклику існуючих. Це призводить до необхідності розробникам сторонніх модулів адаптувати їх для роботи з новими версіями Drupal. Однак зміни API і процедура адаптації модулів до нових версій описуються в документації для кожного релізу, також завжди пропонується механізм автоматизованого апгрейда ядра системи на нову версію. Плюс даної схеми розробки - немає необхідності тягти з версії у версію програмний шар сумісності зі старими API, що полегшує поточний код системи.

Форк Drupal. Проект Backdrop був створений як форк Drupal, його мету ставиться зберегти принципи архітектури Drupal, застосовувані до Drupal 8.



Рис. 2.10 – Логотип Backdrop

Спільнота. На drupal.org зареєстровано понад 1 млн облікових записів, з них - 30 тис. Облікових записів розробників. Постійно діють кілька списків розсилки, відкрито кілька IRC -каналів.

Асоціація Drupal - некомерційна організація, що займається організацією заходів DrupalCon, збором пожертвувань, навчанням і просуванням.

2.2 Мова програмування PHP

PHP - скриптова мова загального призначення, інтенсивно застосовується для розробки веб-додатків. В даний час підтримується переважною більшістю хостинг-провайдерів і є одним з лідерів серед мов, що застосовуються для створення динамічних веб-сайтів.

Мова і її інтерпретатор (Zend Engine) розробляються групою ентузіастів в рамках проекту з відкритим кодом. Проект поширюється під власною ліцензією, несумісною з GNU GPL.

В області веб-програмування, зокрема серверної частини, PHP - один з популярних сценарних мов (поряд з JSP, Perl і мовами, використовуваними в ASP.NET).

Популярність в області побудови веб-сайтів визначається наявністю великого набору вбудованих засобів і додаткових модулів для розробки веб-додатків.

Основні з них:

1. Автоматичне вилучення POST і GET-параметрів, а також змінних оточення веб-сервера в зумовлені масиви;
2. Взаємодія з великою кількістю різних систем управління базами даних через додаткові модулі (MySQL, MySQLi, SQLite, PostgreSQL, Oracle (OCI8), Oracle, Microsoft SQL Server, Sybase, ODBC, mSQL, IBM DB2, Cloudscape і Apache Derby, Informix, Ovrimos SQL, Lotus Notes, DB ++, DBM, dBase, DBX, FrontBase, FilePro, Ingres II, SESAM, Firebird/InterBase, Paradox File Access, MaxDB , Інтерфейс PDO), Redis;
3. Автоматизована відправка HTTP-заголовків;
4. Робота з HTTP-авторизацією;
5. Робота з cookies і сесіями;
6. Робота з локальними і віддаленими файлами, сокетами;
7. Обробка файлів, що завантажуються на сервер;
8. Робота з XForms.

В даний час PHP використовується сотнями тисяч розробників. Згідно з рейтингом корпорації ТЮВЕ, що базується на даних пошукових систем, в травні 2016 року PHP знаходився на 6 місці серед мов програмування. До найбільших сайтів, які використовують PHP, відносяться Facebook , Wikipedia та ін.

2.3 Open Server

Open Server Panel - це портативна серверна платформа і програмне середовище, створена спеціально для веб-розробників з урахуванням їх рекомендацій і побажань.

Програмний комплекс має багатий набір серверного програмного забезпечення, зручний, багатофункціональний продуманий інтерфейс, має потужні можливості з адміністрування та налаштування компонентів. Платформа широко використовується з метою розробки, налагодження і

тестування веб-проектів, а так само для надання веб-сервісів в локальних мережах.

Хоча спочатку програмні продукти, що входять до складу комплексу, не розроблялись спеціально для роботи один з одним, така зв'язка стала дуже популярною серед користувачів Windows, в першу чергу через те, що вони отримували безкоштовний комплекс програм з надійністю на рівні Linux серверів.

Основні компоненти:

1. OSPanel 5.3.5;
2. Apache 2.2.31 / 2.4.38 / 2.4.41;
3. Bind 9.14.5;
4. ConEmu 19.07.144;
5. FTP FileZilla 0.9.60;
6. Ghostscript 9.27;
7. Git 2.23.0;
8. HeidiSQL 10.2.0.5599;
9. Nginx 1.17.3;
10. NNCron Lite 1.17;
11. Sendmail 32;
12. Wget 1.20.3.

Системи управління базами даних:

1. MariaDB 5.5.63 / 10.0.38 / 10.1.38 / 10.2.22 / 10.3.13;
2. Memcached 1.2.6 / 1.4.5;
3. MongoDB 2.4.14 / 2.6.12 / 3.0.15 / 3.2.22 / 3.4.19 / 3.6.11 / 4.0.6 / 4.2.0;
4. MySQL 5.1.73 / 5.5.62 / 5.6.43 / 5.7.25 / 8.0.15;
5. PostgreSQL 9.2.24 / 9.3.25 / 9.4.21 / 9.5.16 / 9.6.12 / 10.7 / 11.2;
6. Redis 2.8.2402 / 3.0.504 / 3.2.100.

PHP програми:

1. Adminer 4.7.3;
2. PHPMemcachedAdmin 1.3;

3. PHPMyAdmin 4.9.0.1;
4. PHPPgAdmin 7;
5. PHPRedisAdmin 1.11.4.

Системні вимоги:

1. Підтримувані версії Windows (32-біт і 64-біт): Windows 7 SP1 і все більш нові версії;
2. Мінімальні апаратні вимоги: 500 МБ вільної RAM і 3 ГБ вільного місця на HDD;
3. Потрібна наявність Microsoft Visual C ++ 2005-2008-2010-2012-2013-2017 Redistributable Package;

Можливості керуючої програми:

1. Швидкі старт і зупинка;
2. Автостарт сервера при запуску програми;
3. Кілька режимів управління доменами;
4. Монтування віртуального диска;
5. Підтримка управління через командний рядок;
6. Підтримка профілів налаштувань;
7. Зручний перегляд логів всіх компонентів;
8. Перемикання HTTP, MySQL і PHP модулів;
9. Вичерпна і зрозуміла документація;
10. Доступ до доменів в один клік;
11. Швидкий доступ до шаблонів конфігурації;
12. Багатомовний інтерфейс;

Особливості комплексу:

1. Не вимагає установки (портативність);
2. Можливість роботи з USB накопичувача;
3. Одночасна робота з Denwer, Xampp і т.д .;
4. Робота на локальному / мережевому / зовнішньому IP адресу;
5. Підтримка SSL без всякої дополн. настройки;
6. Створення домену шляхом створення звичайної папки;

7. Підтримка кириличних доменів;
8. Підтримка аліасів (доменних покажчиків);
9. Захист сервера від зовнішнього доступу;
10. Punycode конвертер доменних імен;
- 11.Пакет з понад 40 портативних програм;
12. Планувальник завдань (cron);

2.4 PHPMyAdmin

phpMyAdmin - веб-додаток з відкритим кодом, написаний на мові PHP і представляє собою веб-інтерфейс для адміністрування СУБД MySQL. PHPMyAdmin дозволяє через браузер і не тільки здійснювати адміністрування сервера MySQL, запускати команди SQL і переглядати вміст таблиць і баз даних. Додаток користується великою популярністю у веб-розробників, так як дозволяє управляти СУБД MySQL без безпосереднього введення SQL команд.

Додаток поширюється під ліцензією GNU General Public License і тому багато інших розробники інтегрують його в свої розробки, наприклад XAMPP, Denwer , AppServ, Open Server.

Проект на даний момент часу локалізований на більш ніж 62 мовах.



Рис. 2.11 - phpMyAdmin

Ініціатором є німецький розробник Тобіас Ратшіллер і ґрунтується так само, як і phpPgAdmin, на аналогічному додатку MySQL-Webadmin, який з'явився в 1997 році .

З 2001 року проект phpMyAdmin у засновника Тобіаса Ратшіллера перейняв Марк ДеЛісл, почавши разом з Олівером Мюллером і Лоик Шапо нову еру phpMyAdmin.

На сьогоднішній день Марк ДеЛісл є адміністратором проекту і має найбільший внесок в phpMyAdmin у вигляді написаного коду. Основними помічниками проекту є Міхал Чіхарж, Себастьян Мендель і Герман ван Рінк.

Версії. Починаючи з версії 3.0.0, phpMyAdmin приєднався до ініціативи GoPHP5 і відмовився від підтримки сумісності з застарілими версіями PHP і MySQL. Для роботи phpMyAdmin 3.0.0 і вище потрібна наявність PHP 5.2 і MySQL5. Для використання старих версій PHP і MySQL продовжує розвиватися друга гілка скрипта (2.x), однак її підтримка обмежується закриттям знайдених вразливостей, нових функцій в неї не додається.

Альтернативи. Adminer (колишній phpMinAdmin) поставляється як єдиний php-файл. Відомий і ряд інших веб-інтерфейсів до MySQL, напр. Chive, розроблений з використанням Yii.

2.5 Висновки до розділу 2

Отже проаналізувавши ряд програмних модулів та продуктів обрано самі популярні та описані в другому розділі. Програмний продукт а саме WEB-додаток буде розроблений за допомогою вище описаних програмних продуктів, також буде протестований та розвернутий на сервері домашнього персонального комп'ютеру.

LAMP - поширений набір програмного забезпечення для створення і хостингу веб-сайтів (Linux, Apache, MySQL, PHP).

3 КЕРІВНИЦТВО КОРИСТУВАЧА ТА ОРГАНІЗАЦІЯ РОБОТИ БАЗИ ДАНИХ

3.1 Керівництво користувача та можливості розробки

Система складається з так званої - вікторини, в якій адміністратори створюють повний комплект запитань, які завчасно формує шкільний вчитель, викладач, куратор гуртка, керівник курсів - на різну форму та тематику.

Спочатку адміністратор при заходженні в систему потрапляє на панель управління модуля, показано на рисунку 3.1.

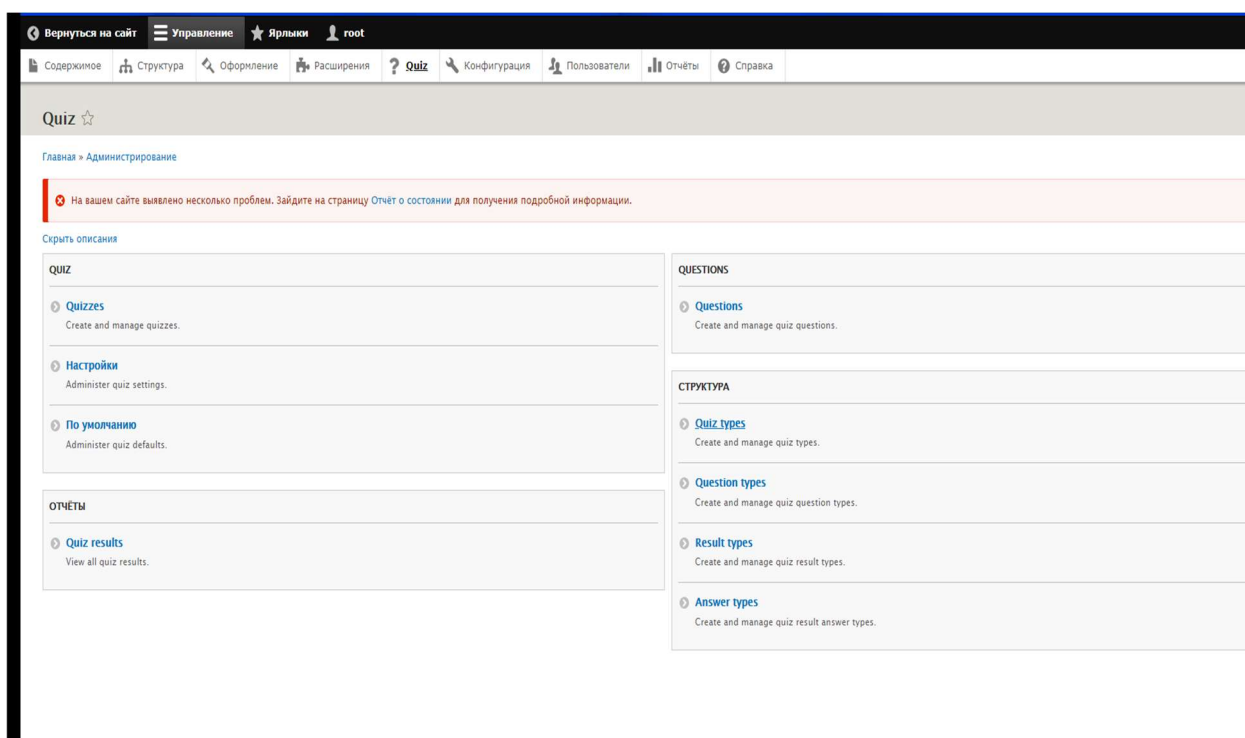


Рис. 3.1 – Вид панелі управління модулем вікторин

На панелі управління вікторинами можна додавати або видаляти дані тестів тобто тематику і теми. Панель створення/редагування вікторини в тест-загадок «Quiz», показано на рисунку 3.2, виглядає як стандартна панель для редагування сторінок або записів в Drupal, але має свої параметри.

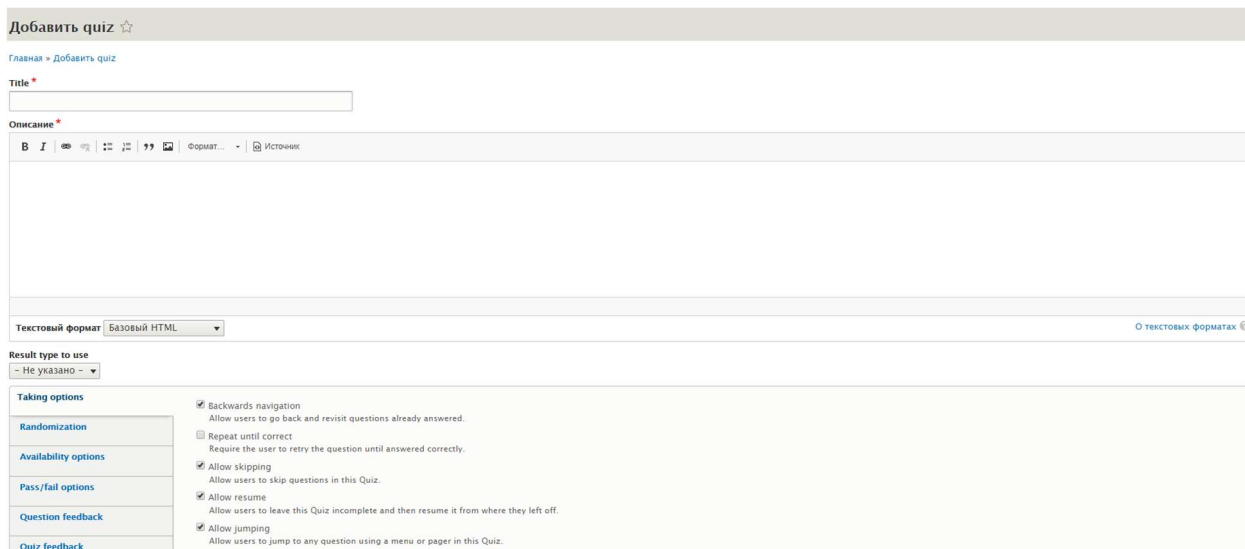


Рис. 3.2 – Панель редагування вікторини тест-загадок «Quiz»

Параметрів багато, так як потрібно створювати вікторини на будь яку ситуацію.

Після встановлення всіх параметрів вікторини тест-загадок «Quiz», можна додати запитання на рисунку 3.3 показний цей принцип.

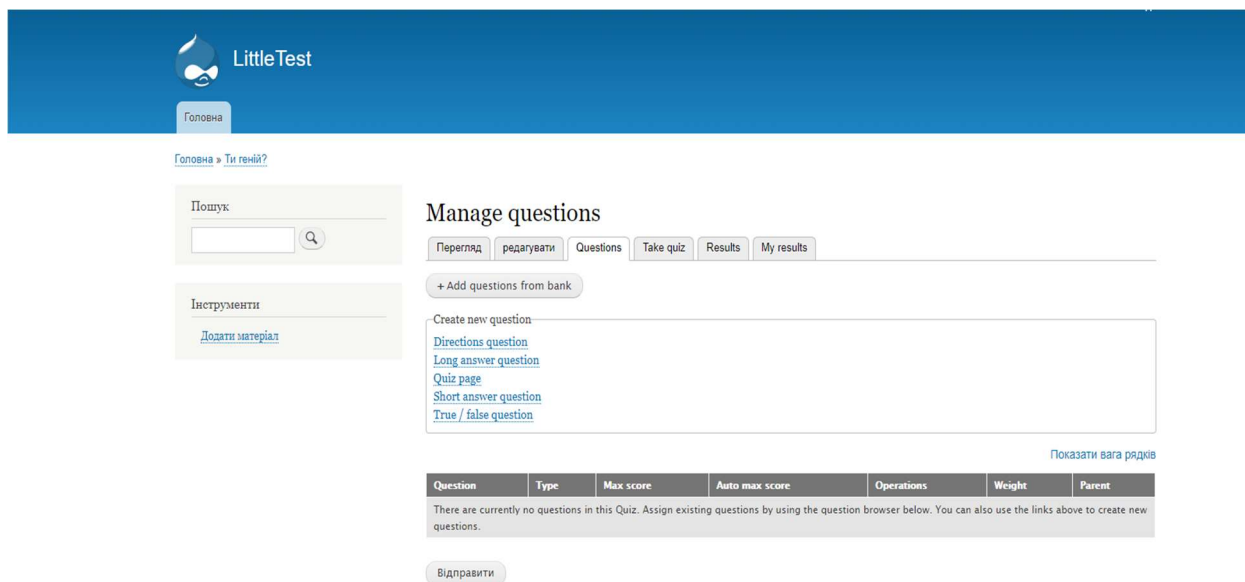


Рис. 3.3 – Вид менеджера редагування запитань

Система тест-загадок «Quiz» всього складеться з п'яти типів питань:

1. Directions question (короткі питання);

2. Long answer question (відкриті питання/есе);
3. Quiz page (заглушка/сторінка без питань);
4. Short answer question (питання з текстовою відповіддю);
5. True/false question (питання «Так або Ні»), приклад одного із таких типів показано на рисунку 3.4. Тут представлена функція розробки Directions question яка дозволяє користувачу при відповіді на запитання дати коротку відповідь. Це означає що користувач може вести деякі слова а бо символи з обмеженою кількістю.

Добавить Directions question

B I | | | | Формат... | Источник

Текстовый формат [О текстовых форматах ?](#)

The generic feedback text to show for this Question.

B I | | | | Формат... | Источник

Текстовый формат [О текстовых форматах ?](#)

Информация о редакции

Нет редакции

Сохранить

Рис. 3.4 – Вид створення короткого питання «Directions question»

На панелі вікторини відображаються всі створені питання, як показано на рисунку 3.5.

Також можна переглянути всі складові закладки:

1. Перегляд;
2. Редагування;
3. Questions;
4. Take quiz;
5. Results;
6. My Results.

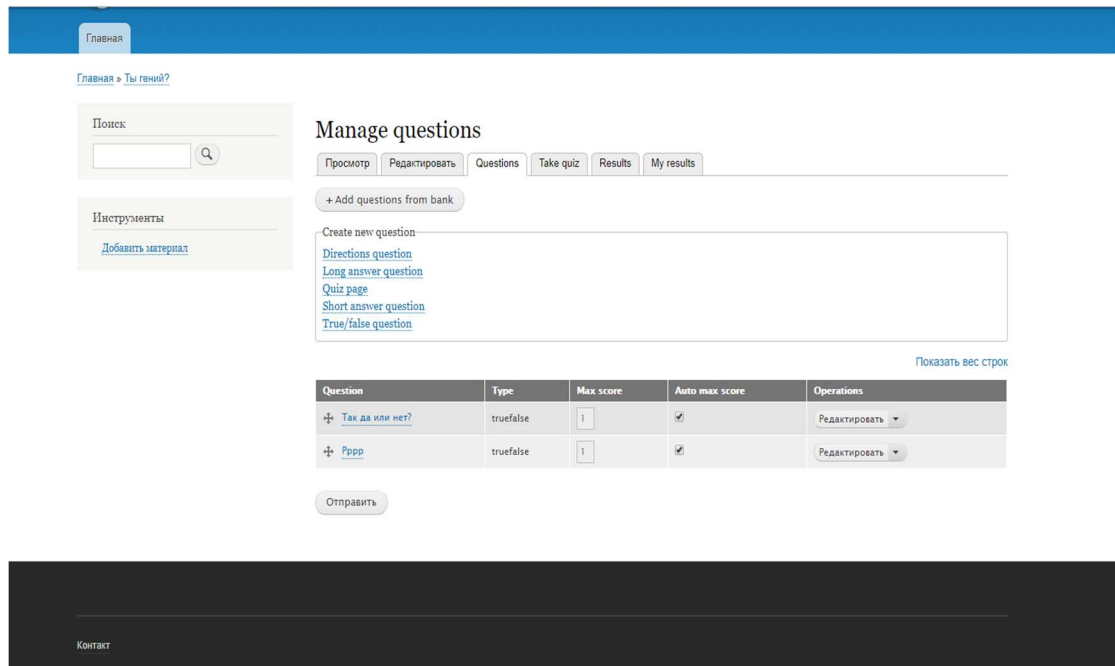


Рис. 3.5 – Менеджер питань з таблицею

Самі питання можна корегувати або видаляти. Додавати різну тематику і вести певну статистику. Також можна заливати певні картинки, схеми, графіки головне розуміти на яку аудиторію розрахований даний тест, а саме якщо це школярі то він не повинен виходити ра рамки навчально процесу.

Як саме дітлахи на чолі з вчителем проходять тестування можна переглянути на представлений схематичній моделі на рисунку 3.6.

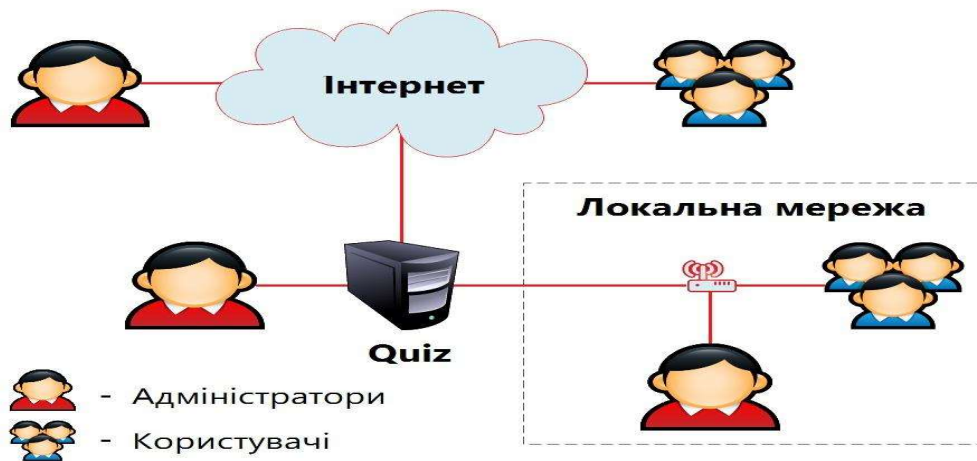


Рис. 3.6 – Схематична модель ролей та процесу проходження тесту-загадок «Quiz»

Адміністратори – це може бути як і відповідальна особа в школі за комп’ютерну лабораторію, також рядовий вчитель який створив свій тест увімкну його підчас уроку з певних причин з контролю зрізу знань. (лабораторна, контрольна, самостійна).

Користувачі – це самі учні школи різного стуленню навчання, тест з відгадування загадок уможлиблює проходження кожного з мінімальним розумінням ПК.

Адміністратор тесту - може подивитись результати та підпити певну статистику, показано на рисунку 3.7.

Quiz results ☆

[Просмотр](#)
[Редактировать](#)
[Questions](#)
[Take quiz](#)
[Results](#)
[My results](#)

Головна » Ты гений?

Delete selected attempts

NAME	ATTEMPT START TIME	ATTEMPT END TIME	SCORE	EVALUATED	OPERATIONS
root	вс, 12/15/2019 - 19:31				view score

Delete selected attempts

Рис. 3.7 – Вид переглядів результатів користувачів

А для початку тест-вікторини знову з тим же користувачем, треба видалити його попередній результат або створити нову вікторину.

У запитаннях тесту є огляд, тобто деяка картка, яка відображається перед початком тестування, рисунок 3.8.

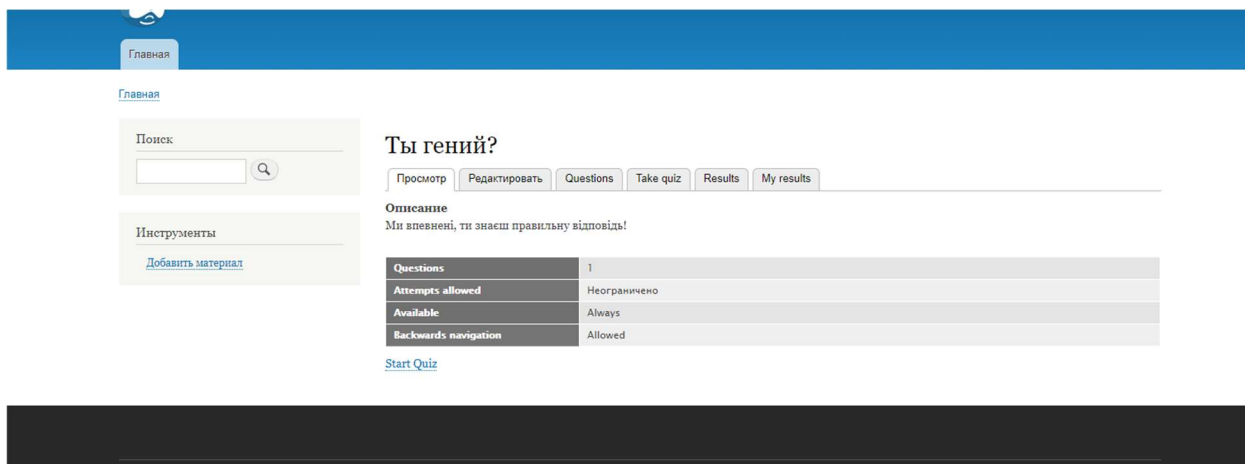


Рис. 3.19 – Вид огляду вікторини

3.2 Модернізація бази даних розробленої системи

Побудована база даних системи тест-загадок «Quiz», має 135 таблиць, кожна з них не об'єднана зв'язками так, як дані передаються на РНР-запити:

1. Batch;
2. block_content;
3. block_content_field_data;
4. block_content_field_revisio;
5. block_content_revision;
6. block_content_revision__body;
7. block_content__body;
8. cachetags;
9. cache_bootstrap;
10. cache_config;
11. cache_container;

12. cache_data;
13. cache_default;
14. cache_discovery;
15. cache_dynamic_page_cache;
16. cache_entity;
17. cache_menu;
18. cache_page;
19. cache_render;
20. cache_toolbar;
21. comment;
22. comment_0entity_statistics;
23. comment_field_data;
24. comment__comment_body;
25. config;
26. file_managed;
27. file_usage;
28. history;
29. key_value;
30. key_value_expire;
31. locales_location;
32. locales_source;
33. locales_target;
34. locale_file;
35. menu_link_content;
36. menu_link_content_data;
37. menu_link_content_field_revision;
38. menu_link_content_revision;
39. menu_tree;
40. node;
41. node_access;

42.node_field_data;
43.node_field_revision;
44.node_revision;
45.node_revision__body;
46.node_revision__comment;
47.node_revision__field_image;
48.node_revision__field_tags;
49.node__body;
50.node__comment;
51.node__field_image;
52.node__field_tags;
53.paragraphs_item;
54.paragraphs_item_field_data;
55.paragraphs_item_revision;
56.paragraphs_item_revision_field_data;
57.paragraph_revision__multichoice_answer;
58.paragraph_revision__multichoice_correct;
59.paragraph_revision__multichoice_feedback_chosen;
60.paragraph_revision__multichoice_score_chosen;
61.paragraph_revision__multichoice_score_not_chosen;
62.paragraph_revision__quiz_feedback;
63.paragraph_revision__quiz_feedback_range;
64.paragraph_revision__quiz_question_max_score;
65.paragraph_revision__quiz_question_number;
66.paragraph_revision__quiz_question_tid;
67.paragraph_r__5792255841;
68.paragraph__multichoice_answer;
69.paragraph__multichoice_correct;
70.paragraph__multichoice_feedback_chosen;
71.paragraph__multichoice_feedback_not_chosen;

72. paragraph__multichoice_score_chosen;
73. paragraph__multichoice_score_not_chosen;
74. paragraph__quiz_feedback;
75. paragraph__quiz_feedback_range;
76. paragraph__quiz_question_max_score;
77. paragraph__quiz_question_number;
78. paragraph__quiz_question_tid;
79. path_alias;
80. path_alias_revision;
81. Queue;
82. Quiz;
83. quiz_question;
84. quiz_question_relationship;
85. quiz_question_revision;
86. quiz_question_revision__alternatives;
87. quiz_question_revision__choice_boolean;
88. quiz_question_revision__choice_multi;
89. quiz_question_revision__choice_random;
90. quiz_question_revision__long_answer_rubric;
91. quiz_question_revision__short_answer_correct;
92. quiz_question_revision__short_answer_evaluation;
93. quiz_question_revision__truefalse_correct;
94. quiz_question__alternatives;
95. quiz_question__choice_boolean;
96. quiz_question__choice_multi;
97. quiz_question__choice_random;
98. quiz_question__long_answer_rubric;
99. quiz_question__short_answer_correct;
100. quiz_question__short_answer_evaluation;
101. quiz_question__truefalse_correct;

102. quiz_result;
103. quiz_result_answer;
104. quiz_result_answer__long_answer;
105. quiz_result_answer__multichoice_answer;
106. quiz_result_answer__short_answer;
107. quiz_result_answer__truefalse_answer;
108. quiz_revision;
109. quiz_revision__quiz_terms;
110. quiz_revision__result_options;
111. quiz__quiz_terms;
112. quiz__result_options;
113. router;
114. search_dataset;
115. search_index;
116. search_total;
117. semaphore;
118. sequences;
119. sessions;
120. shortcut;
121. shortcut_field_data;
122. shortcut_set_users;
123. taxonomy_index;
124. taxonomy_term_data;
125. taxonomy_term_field_data;
126. taxonomy_term_field_revision;
127. taxonomy_term_revision;
128. taxonomy_term_revision__parent;
129. taxonomy_term__parent;
130. users;
131. users_data;

- 132. users_field_data;
- 133. user_roles;
- 134. user_user_picture;
- 135. watchdog, показано на рисунку 3.9.

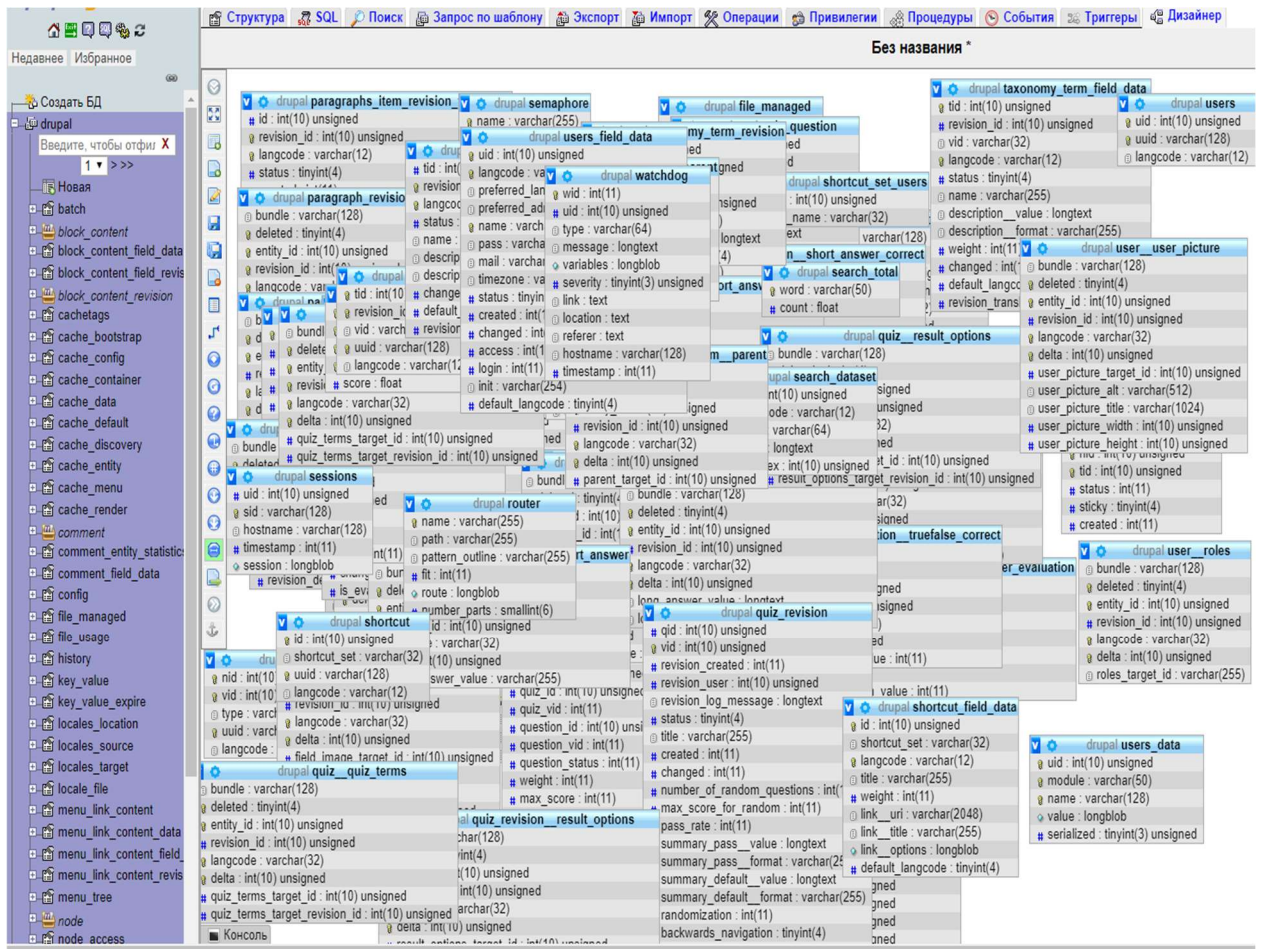


Рис. 3.9 – Схема бази даних

Вище вказані таблиці зберігають всі дані, різноманітну інформацію та матеріали. Також серед них можна виділити основні таблиці, це п'ять головних:

1. Таблиця «Вчителі»;
2. Таблиця «Учні»;
3. Таблиця «Теми тестів»;
4. Таблиця «Тести»;
5. Таблиця «Статистика», головні таблиці представлені на рисунку 3.10.

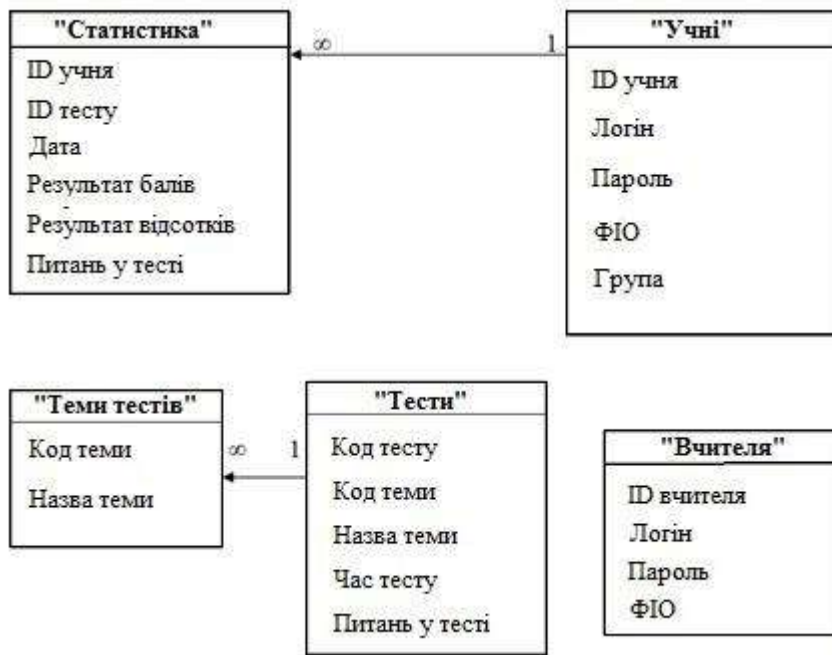


Рис. 3.10 – Модель схеми бази даних головних таблиць

Кожна таблиця представленої моделі містить свої основні дані .

Таблиця «Статистика» складається із: ID учня, ID тесту, Дата, Результати балів, результати відсотків, питання-загадки тесту.

Таблиця «Учні» - ID учня, логін, пароль, ПІБ, група (клас де навчається).

Таблиця «Вчителя» - ID вчителя, логін, пароль, ПІБ.

Таблиця «Тести» - код тесту, код теми, назва теми, час тесту, питання у тесті.

Таблиця «Теми тестів» - код теми та назва теми.

Таблиця «Статистика» та таблиця «Учні» - мають тип зв'язку один – до – багатьох, таблиця «Тести» та таблиця «Теми тестів» мають той самий зв'язок. А якщо розглядати повну СУБД існує кілька видів запитів:

1. Запити на вибірку, що здійснюють вибірку даних з таблиць, наприклад:

```
SELECT *
```

```
FROM *;
```

вибірка всіх даних з усіх існуючих таблиць;

2. Запити на зміну, що дозволяють модифікувати дані в таблицях (у тому числі видаляти, оновлювати і додавати записи), наприклад:

```
UPDATE Тест
```

```
SET Прилад. [Готовий до перевірки] = True
```

```
WHERE Прибор.Id = 585
```

даний запит дозволяє змінити дані в таблиці Прилад поля Прилад. [Готовий до повірці];

3. Запити на додавання, що дозволяють додавати дані в таблицю, наприклад:

```
INSERT INTO [Номери списків] ([В повірці], Цех) VALUES (True, 5);
```

даний запит додає в таблицю Номери списків в поле клас – номер класу де навчається учень, а в поле В повірці – стан списку;

4. Запити на видалення, що дозволяють видаляти дані з таблиці або самі таблиці, наприклад:

```
DELETE Прибор.Id
```

```
FROM Прилад
```

```
WHERE Прибор.Id = 585
```

даний запит видаляє з таблиці Прилад з номером 585.

5. Запити на автопідстановку, що дозволяють автоматичний заповнювати поля для нового запису (Microsoft Access) [3];

6. Запити на видалення таблиці, що дозволяють видаляти таблицю з розташованими в ній даними, наприклад:

```
DELETE TABLE user;
```

даний запит видаляє таблицю user.

Запит на вибірку містить умови відбору даних і повертає вибірку, відповідну зазначеним умовам, без зміни повертаються даних. В СУБД існує також поняття фільтра, який в свою чергу є набором умов, що дозволяють відбирати підмножину записів або сортувати їх. Основне схожість між запитом на вибірку і фільтрами полягає в тому, що в них проводиться витяг підмножини записів з базової таблиці або запити. Для здійснення вибору

фільтру або запиту необхідно розуміти, як передбачається використовувати відібрані записи. [6] Фільтр можна використовувати при роботі в режимі таблиці для перегляду або зміни підмножини записів.

Запит можна використовувати:

1. Для перегляду підмножини записів таблиці без попереднього відкриття цієї таблиці;
2. Для того щоб об'єднати у вигляді однієї таблиці на екрані дані з декількох таблиць;
3. Для перегляду окремих полів таблиці;
4. Для виконання окремих обчислень над значеннями полів.
5. Основні відмінності запитів і фільтрів полягає в наступному:
6. Фільтри не дозволяють додати ще кілька таблиць, записи яких включаються до повертається набір записів;
7. Фільтри не дозволяють вказати поля, які повинні відображатися в результуючому наборі записів;
8. Фільтри не можуть бути виведені як окремий об'єкт у вікні бази даних;
9. Фільтри не дозволяють обчислювати суми, середні значення, підраховувати кількість записів і знаходити інші підсумкові значення.

Залежно від розв'язуваних завдань у запитах застосовують кілька видів з'єднань це внутрішнє, зовнішнє і рекурсивне.

Найбільш поширеними є внутрішнє з'єднання. Якщо таблиці пов'язані співвідношенням «один – до – багатьох», сполуки ґрунтуються на унікальному значенні поля первинного ключа в одній таблиці і значеннях поля зовнішнього ключа в іншій таблиці. У результуюче безліч запити потрапляють всі записи з головної таблиці (таблиці «один»), для яких є відповідні записи в підлеглий таблиці (таблиця «багато»). Якщо у підпорядкованій таблиці записи із заданою величиною відсутні, то відповідні записи в головній таблиці в результуюче безліч не включається. Результатом такого запити є всі записи, значення пов'язаних полів яких в обох таблицях збігаються. Іншими словами, еквісоедінення пов'язують дані полів записів таблиць ставленням рівності.

Для створення запиту, що об'єднує всі записи з однієї таблиці і лише ті записи з другої, в яких пов'язані поля збігаються, використовують зовнішнє з'єднання. Незалежно від того, чи є відповідні записи в другій таблиці, всі записи першої потрапляють в результуюче безліч запиту.

Якщо необхідно пов'язати дані будь – яким ставленням, крім відносини рівності, використовують з'єднання по відношенню (тета – з'єднання).

Для позначення внутрішніх з'єднань в СУБД використовується зарезервовані слова SQL – це INNER JOIN, а для вказівки зовнішніх з'єднань – слова LEFT JOIN або RIGHT JOIN. Вираз WHERE <поле1> = <поле2> відповідає внутрішньому з'єднанню, а WHERE <поле1> <поле2> відповідає з'єднанню по відношенню (одному з таких з'єднань), наприклад:

Для скріплення даних в одній таблиці застосовують рекурсивне з'єднання, яке створюється шляхом додавання до запиту копії таблиці і зв'язування полів ідентичних таблиць. Рекурсивні з'єднання вкрай рідко використовуються в базах даних не дивлячись на те, що вони і забезпечують цілісність даних і можливість задати обмеження значень.

Також можна переглянути графічне представлення, тих самих таблиць в РНР, показано на рисунку 3.11

<input type="checkbox"/>	node_field_data	★									0	InnoDB	utf8mb4_general_ci	168 КіБ	-
<input type="checkbox"/>	node_field_revision	★									0	InnoDB	utf8mb4_general_ci	48 КіБ	-
<input type="checkbox"/>	node_revision	★									0	InnoDB	utf8mb4_general_ci	64 КіБ	-
<input type="checkbox"/>	node_revision_body	★									0	InnoDB	utf8mb4_general_ci	64 КіБ	-
<input type="checkbox"/>	node_revision_comment	★									0	InnoDB	utf8mb4_general_ci	48 КіБ	-
<input type="checkbox"/>	node_revision_field_image	★									0	InnoDB	utf8mb4_general_ci	64 КіБ	-
<input type="checkbox"/>	node_revision_field_tags	★									0	InnoDB	utf8mb4_general_ci	64 КіБ	-
<input type="checkbox"/>	node_body	★									0	InnoDB	utf8mb4_general_ci	64 КіБ	-
<input type="checkbox"/>	node_comment	★									0	InnoDB	utf8mb4_general_ci	48 КіБ	-
<input type="checkbox"/>	node_field_image	★									0	InnoDB	utf8mb4_general_ci	64 КіБ	-
<input type="checkbox"/>	node_field_tags	★									0	InnoDB	utf8mb4_general_ci	64 КіБ	-
<input type="checkbox"/>	paragraphs_item	★									0	InnoDB	utf8mb4_general_ci	64 КіБ	-
<input type="checkbox"/>	paragraphs_item_field_data	★									0	InnoDB	utf8mb4_general_ci	96 КіБ	-
<input type="checkbox"/>	paragraphs_item_revision	★									0	InnoDB	utf8mb4_general_ci	32 КіБ	-
<input type="checkbox"/>	paragraphs_item_revision_field_data	★									0	InnoDB	utf8mb4_general_ci	32 КіБ	-
<input type="checkbox"/>	paragraph_revision_quiz_feedback	★									0	InnoDB	utf8mb4_general_ci	64 КіБ	-
<input type="checkbox"/>	paragraph_revision_quiz_feedback_range	★									0	InnoDB	utf8mb4_general_ci	48 КіБ	-
<input type="checkbox"/>	paragraph_revision_quiz_question_max_score	★									0	InnoDB	utf8mb4_general_ci	48 КіБ	-
<input type="checkbox"/>	paragraph_revision_quiz_question_number	★									0	InnoDB	utf8mb4_general_ci	48 КіБ	-
<input type="checkbox"/>	paragraph_revision_quiz_question_tid	★									0	InnoDB	utf8mb4_general_ci	64 КіБ	-
<input type="checkbox"/>	paragraph_quiz_feedback	★									0	InnoDB	utf8mb4_general_ci	64 КіБ	-
<input type="checkbox"/>	paragraph_quiz_feedback_range	★									0	InnoDB	utf8mb4_general_ci	48 КіБ	-
<input type="checkbox"/>	paragraph_quiz_question_max_score	★									0	InnoDB	utf8mb4_general_ci	48 КіБ	-
<input type="checkbox"/>	paragraph_quiz_question_number	★									0	InnoDB	utf8mb4_general_ci	48 КіБ	-
<input type="checkbox"/>	paragraph_quiz_question_tid	★									0	InnoDB	utf8mb4_general_ci	64 КіБ	-
<input type="checkbox"/>	path_alias	★									0	InnoDB	utf8mb4_general_ci	96 КіБ	-
<input type="checkbox"/>	path_alias_revision	★									0	InnoDB	utf8mb4_general_ci	32 КіБ	-
<input type="checkbox"/>	queue	★									0	InnoDB	utf8mb4_general_ci	48 КіБ	-
<input type="checkbox"/>	quiz	★									0	InnoDB	utf8mb4_general_ci	224 КіБ	-
<input type="checkbox"/>	quiz_question	★									0	InnoDB	utf8mb4_general_ci	96 КіБ	-
<input type="checkbox"/>	quiz_question_relationship	★									0	InnoDB	utf8mb4_general_ci	64 КіБ	-
<input type="checkbox"/>	quiz_question_revision	★									0	InnoDB	utf8mb4_general_ci	48 КіБ	-
<input type="checkbox"/>	quiz_question_revision_long_answer_rubric	★									0	InnoDB	utf8mb4_general_ci	64 КіБ	-
<input type="checkbox"/>	quiz_question_revision_short_answer_correct	★									0	InnoDB	utf8mb4_general_ci	48 КіБ	-
<input type="checkbox"/>	quiz_question_revision_short_answer_evaluation	★									0	InnoDB	utf8mb4_general_ci	64 КіБ	-
<input type="checkbox"/>	quiz_question_revision_truefalse_correct	★									0	InnoDB	utf8mb4_general_ci	64 КіБ	-
<input type="checkbox"/>	quiz_question_long_answer_rubric	★									0	InnoDB	utf8mb4_general_ci	64 КіБ	-
<input type="checkbox"/>	quiz_question_short_answer_correct	★									0	InnoDB	utf8mb4_general_ci	48 КіБ	-
<input type="checkbox"/>	quiz_question_short_answer_evaluation	★									0	InnoDB	utf8mb4_general_ci	64 КіБ	-
<input type="checkbox"/>	quiz_question_truefalse_correct	★									0	InnoDB	utf8mb4_general_ci	48 КіБ	-
<input type="checkbox"/>	quiz_result	★									0	InnoDB	utf8mb4_general_ci	48 КіБ	-
<input type="checkbox"/>	quiz_result_answer	★									0	InnoDB	utf8mb4_general_ci	64 КіБ	-
<input type="checkbox"/>	quiz_result_answer_long_answer	★									0	InnoDB	utf8mb4_general_ci	64 КіБ	-
<input type="checkbox"/>	quiz_result_answer_short_answer	★									0	InnoDB	utf8mb4_general_ci	48 КіБ	-
<input type="checkbox"/>	quiz_result_answer_truefalse_answer	★									0	InnoDB	utf8mb4_general_ci	64 КіБ	-
<input type="checkbox"/>	quiz_revision	★									0	InnoDB	utf8mb4_general_ci	176 КіБ	-
<input type="checkbox"/>	quiz_revision_quiz_terms	★									0	InnoDB	utf8mb4_general_ci	80 КіБ	-
<input type="checkbox"/>	quiz_revision_result_options	★									0	InnoDB	utf8mb4_general_ci	80 КіБ	-
<input type="checkbox"/>	quiz_quiz_terms	★									0	InnoDB	utf8mb4_general_ci	80 КіБ	-
<input type="checkbox"/>	quiz_result_options	★									0	InnoDB	utf8mb4_general_ci	80 КіБ	-
<input type="checkbox"/>	router	★								489	InnoDB	utf8mb4_general_ci	1,6 МБ	-	
<input type="checkbox"/>	search_dataset	★									0	InnoDB	utf8mb4_general_ci	16 КіБ	-
<input type="checkbox"/>	search_index	★									0	InnoDB	utf8mb4_general_ci	32 КіБ	-
<input type="checkbox"/>	search_total	★									0	InnoDB	utf8mb4_general_ci	16 КіБ	-
<input type="checkbox"/>	semaphore	★									0	InnoDB	utf8mb4_general_ci	48 КіБ	-
<input type="checkbox"/>	sequences	★								1	InnoDB	utf8mb4_general_ci	16 КіБ	-	
<input type="checkbox"/>	sessions	★								1	InnoDB	utf8mb4_general_ci	48 КіБ	-	
<input type="checkbox"/>	shortcut	★								2	InnoDB	utf8mb4_general_ci	48 КіБ	-	
<input type="checkbox"/>	shortcut_field_data	★								2	InnoDB	utf8mb4_general_ci	64 КіБ	-	
<input type="checkbox"/>	shortcut_set_users	★								0	InnoDB	utf8mb4_general_ci	32 КіБ	-	
<input type="checkbox"/>	taxonomy_index	★								0	InnoDB	utf8mb4_general_ci	32 КіБ	-	
<input type="checkbox"/>	taxonomy_term_data	★								0	InnoDB	utf8mb4_general_ci	64 КіБ	-	
<input type="checkbox"/>	taxonomy_term_field_data	★								0	InnoDB	utf8mb4_general_ci	112 КіБ	-	
<input type="checkbox"/>	taxonomy_term_field_revision	★								0	InnoDB	utf8mb4_general_ci	48 КіБ	-	
<input type="checkbox"/>	taxonomy_term_revision	★								0	InnoDB	utf8mb4_general_ci	48 КіБ	-	
<input type="checkbox"/>	taxonomy_term_revision_parent	★								0	InnoDB	utf8mb4_general_ci	64 КіБ	-	
<input type="checkbox"/>	taxonomy_term_parent	★								0	InnoDB	utf8mb4_general_ci	64 КіБ	-	
<input type="checkbox"/>	users	★								2	InnoDB	utf8mb4_general_ci	32 КіБ	-	
<input type="checkbox"/>	users_data	★								0	InnoDB	utf8mb4_general_ci	48 КіБ	-	
<input type="checkbox"/>	users_field_data	★								2	InnoDB	utf8mb4_general_ci	96 КіБ	-	
<input type="checkbox"/>	user_roles	★								1	InnoDB	utf8mb4_general_ci	64 КіБ	-	
<input type="checkbox"/>	user_user_picture	★								0	InnoDB	utf8mb4_general_ci	64 КіБ	-	
<input type="checkbox"/>	watchdog	★								92	InnoDB	utf8mb4_general_ci	112 КіБ	-	
<input type="checkbox"/>	111 таблиц										23,790	InnoDB	utf8_general_ci	24,4 МБ	0 байт

Рис. 3.11 – Вид списку всіх полів таблиць

3.3 Опис програмного продукту та оцінка ефективності

WEB-додаток з вивчення слів з англійської мови спроектовано та розроблено в листопаді 2020 року, версія 0.0.1. Програмний продукт під назвою тест-загадок «Quiz», призначений для навчального процесу з вивчення англійської мови:

1. В загальноосвітній школі;
2. Гуртках позашкільної освіти;
3. Здобувачів освіти з особливими потребами;
4. Дистанційній формі здобування освіти;
5. Також всіх бажаючих які виявили бажання в здобутті знань з іноземної мови.

Всі вище перераховані якщо мають постійний доступ в інтернет ресурс можуть скористуватися розробленим WEB-додатком та присвяти себе вивченню мови, при тестуванні та контролю певного рівня знань. Система відповідає ДСТУ програмних проєктів ІСО/МЭК 12119-2000 "Інформаційна технологія. Пакети програм. Вимога до якості та тестування». Програмно-технічні засоби.

Програмний продукт тест-загадок «Quiz», середовищем розробки є:

1. Мова розмітки гіпертекстових документів html;
2. Мова опису зовнішнього виду документа css;
3. Мова програмування скрипкових додатків js;
4. СУБД – MySQL phpMyAdmin 3.5.1, під управлінням операційної системи MS Windows 10 та вище.
5. CMS Drupal.

Технічне забезпечення:

CPU Intel Pentium IV 2,6 GHz/1024 Mb;

1. HDD 160Gb SATA;
2. Мережа 1000/100 Mb/s;
3. ОС MS Windows XP,7,10;

4. СУБД – phpMyAdmin 3.5.1.

Клієнт:

1. CPU Intel Celeron 1,3 GHz/128 Mb;
2. HDD 120 Gb;
3. ОС MS Windows 10 і вище;
4. Актуальний різноплановий браузер;
5. Інтернет.

Програмний продукт — це властивість системи виконувати поставлену мету в заданих умовах з обумовленою якістю. Показники ефективності характеризують ступінь пристосованості системи до виконання поставлених перед нею завдань та є узагальнюючими показниками оптимальності функціонування ІС [20].

3.4 Висновки до розділу 3

Розроблений WEB-додаток тест-загадок «Quiz», має можливість добре організувати робочий інтерактивний процес під час навчання будь якого освітнього закладу. Розробка включає спрощений функціонал на відміну від проаналізованих аналогів, створено функцію розділення запитань завдяки сторінці «Заглушка».

Програмний продукт - є модулем, тому може підключитися до будь-якого сайту на Drupal без додаткових налаштувань.

ВИСНОВКИ

У магістерській роботі було проаналізовано безліч WEB-систем інтернет тестування, виявлено безліч плюсів та мінусів подібних систем. Також після збору матеріалів та аналізу, розроблено програмний продукт який відповідає всім висунутим вимогам і включає в себе наступний функціонал:

1. Самостійне створення тесту з гнучкими налаштуваннями;
2. Створення різних типів запитання за різноманітною тематикою англійських слів;
3. Використання групової політики доступу;
4. Призначення максимальної кількості балів за кожну повністю вірну відповідь слухачеві який займається вивченням слів англійської мови.

Реалізовано основну мету практичної частини магістерської роботи: розроблено WEB-додаток з вивчення слів англійської мови під назвою тест-загадок «Quiz».

Були поставлені задачі, які успішно виконані:

1. Написано код на мові програмування PHP, JS, HTML та CSS;
2. Розроблено мінімалістичний дизайн;
3. Розроблено та модернізовано базу даних;
4. Захист бази питань від несанкціонованого доступу і зміни;
5. Розроблений програмний продукт запущено на сервері та протестована на клієнті.

ПЕРЕЛІК ПОСИЛАНЬ

1. Аванесов В. С. Композиція тестових завдань. - М., Центр тестування, 2002.
2. Зорін С. Ф. Розробка автоматизованої системи контролю знань студентів з дисципліни «Економіка підприємства». МГВМІ, 2007.
3. Майоров А. Н. Теорія і практика створення тестів для системи освіти: Як вибирати, створювати і використовувати тести для цілей освіти. М: Інтелект-Центр, 2002.
4. Морев І. А. Освітні інформаційні технології. Частина 2. Педагогічні вимірювання: Навчальний посібник. - Владивосток: Вид-во Дальневост. ун-ту, 2004.
5. Нейман Ю. М., Хлебніков В. А. Педагогічне тестування як вимір. Ч.1. - М.: Центр тестування МО України, 2002.
6. Челишкова М. Б. Теорія і практика конструювання педагогічних тестів. Уч. Допомога. - М.: Логос, 2002.
7. Кабанова Т. А., Новиков В. А. Тестування в сучасній освіті. Уч. Допомога. - М.: Вища школа, 2010 року.
8. Казиев В. М. Введення в практичне тестування. - М.: Інтуїт.ру, Біном. Лабораторія Знання, 2008.
9. Батешов Е. А. «Основи технологізації комп'ютерного тестування»: Навчальний посібник. - Астана: ТОВ «Поліграф-світ», 2011. - 241 с.
10. Официальный сайт проекта Dgural
11. Сайт русскоязычного Dgural-сообщества
12. Дмитрий Котеров, Алексей Костарев. РНР. — СПб.: «БХВ-Петербург», 2005. — С. 1120.
13. Костарев А. Ф. РНР 5. — СПб.: «БХВ-Петербург», 2008. — С. 1104.
14. Мэтт Зандстра. РНР: объекты, шаблоны и методики программирования = РНР Objects, Patterns and Practice, Third Edition. — 3-е издание. — М.: «Вильямс», 2010. — С. 560.

- 15.Кристиан Дари, Эмилиан Баланеску. PHP и MySQL: создание интернет-магазина = Beginning PHP and MySQL E-Commerce: From Novice to Professional. — М.: «Вильямс», 2010.
- 16.Джейсон Ленгсторф. PHP и jQuery для профессионалов = Pro PHP and jQuery. — М.: «Вильямс», 2010. — С. 352.
- 17.Стив Суэринг, Тим Конверс, Джойс Парк. PHP и MySQL = PHP 6 and MySQL 6 Bible. — 2-е издание. — М.: «Диалектика», 2010. — С. 912. — (Библия программиста).
- 18.Квентин Зервас. Web 2.0: создание приложений на PHP = Practical Web 2.0 Applications with PHP. — М.: «Вильямс», 2009. — С. 544.
- 19.Кузнецов Максим, Симдянов Игорь. PHP 5/6. — СПб.: «БХВ-Петербург», 2009. — С. 1024.
- 20.Кузнецов Максим, Симдянов Игорь. Объектно-ориентированное программирование на PHP. — СПб.: «БХВ-Петербург», 2007. — С. 608.
- 21.Эд Леки-Томпсон, Алек Коув, Стивен Новицки, Хьяо Айде-Гудман. PHP 5 для профессионалов = Professional PHP 5. — М.: «Диалектика», 2006. — С. 608.
- 22.Кузнецов Максим, Симдянов Игорь. Самоучитель PHP 5/6. — 3-е изд., перераб. и доп. — СПб.: «БХВ-Петербург», 2009. — С. 672.
- 23.Кузнецов Максим, Симдянов Игорь. PHP. Практика создания Web-сайтов. — 2-е изд. перераб. и доп. — СПб.: «БХВ-Петербург», 2008. — С.
- 24.Кузнецов Максим, Симдянов Игорь. Головоломки на PHP для хакера. — 2-е изд. перераб. и доп. — СПб.: «БХВ-Петербург», 2008. — С. 544.

Додаток А

Код WEB-додатку

```
(function ($, Drupal) {  
  Drupal.behaviors.quizAnswerConfirm = {
```

```
attach: function (context, settings) {
  $('#quiz-question-answering-form #edit-navigation-submit, #quiz-question-
    answering-form #edit-navigation-skip').once('quizConfirm').click(function
    (event) {
      // Return false to avoid submitting if user aborts
      if (!confirm($('#quiz-question-answering-form').data('confirm-message'))) {
        event.preventDefault();
      }
    });
}
})(jQuery, Drupal);
```

Додаток В

Файл API

```

<?php

/**
 * @file quiz.api.php
 * Hooks provided by Quiz module.
 *
 * These entity types provided by Quiz also have entity API hooks. There are a
 * few additional Quiz specific hooks defined in this file.
 *
 * quiz (settings for quiz nodes)
 * quiz_result (quiz attempt/result)
 * quiz_result_answer (answer to a specific question in a quiz result)
 * quiz_question (generic settings for question nodes)
 * quiz_question_relationship (relationship from quiz to question)
 *
 * So for example
 *
 * hook_quiz_result_presave($quiz_result)
 * - Runs before a result is saved to the DB.
 * hook_quiz_question_relationship_insert($quiz_question_relationship)
 * - Runs when a new question is added to a quiz.
 *
 * You can also use Rules to build conditional actions based off of these
 * events.
 *
 * Enjoy :)
 */
/**
 * Implements hook_quiz_begin().
 *
 * Fired when a new quiz result is created.
 *
 * @deprecated
 *
 * Use hook_quiz_result_insert().
 */
function hook_quiz_begin($quiz, $result_id) {
}
/**
 * Implements hook_quiz_finished().
 *
 * Fired after the last question is submitted.
 *
 * @deprecated
 *
 * Use hook_quiz_result_update().
 */
function hook_quiz_finished($quiz, $score, $data) {
}
/**

```

```

* Implements hook_quiz_scored().
*
* Fired when a quiz is evaluated.
*
* @deprecated
*
* Use hook_quiz_result_update().
*/
function hook_quiz_scored($quiz, $score, $result_id) {
}
/**
* Implements hook_quiz_question_info().
*
* Define a new question type. The question provider must extend QuizQuestion,
* and the response provider must extend QuizQuestionResponse. See those classes
* for additional implementation details.
*/
function hook_quiz_question_info() {
  return array(
    'long_answer' => array(
      'name' => t('Example question type'),
      'description' => t('An example question type that does something.'),
      'question provider' => 'ExampleAnswerQuestion',
      'response provider' => 'ExampleAnswerResponse',
      'module' => 'quiz_question',
    ),
  );
}
/**
* Expose a feedback option to Quiz so that Quiz administrators can choose when
* to show it to Quiz takers.
*
* @return array
* An array of feedback options keyed by machine name.
*/
function hook_quiz_feedback_options() {
  return array(
    'percentile' => t('Percentile'),
  );
}
/**
* Allow modules to alter the quiz feedback options.
*
* @param array $review_options
* An array of review options keyed by a machine name.
*/
function hook_quiz_feedback_options_alter(&$review_options) {
  // Change label.
  $review_options['quiz_feedback'] = t('General feedback from the Quiz.');
```

```

unset($review_options['solution']);
}
/**
 * Allow modules to define feedback times.
 *
 * Feedback times are configurable by Rules.
 *
 * @return array
 * An array of feedback times keyed by machine name.
 */
function hook_quiz_feedback_times() {
  return array(
    '2_weeks_later' => t('Two weeks after finishing'),
  );
}

/**
 * Allow modules to alter the feedback times.
 *
 * @param array $feedback_times
 */
function hook_quiz_feedback_times_alter(&$feedback_times) {
  // Change label.
  $feedback_times['end'] = t('At the end of a quiz');

  // Do not allow question feedback.
  unset($feedback_times['question']);
}

/**
 * Allow modules to alter the feedback labels.
 *
 * These are the labels that are displayed to the user, so instead of
 * "Answer feedback" you may want to display something more learner-friendly.
 *
 * @param $feedback_labels
 * An array keyed by the feedback option. Default keys are the keys from
 * quiz_get_feedback_options().
 */
function hook_quiz_feedback_labels_alter(&$feedback_labels) {
  $feedback_labels['solution'] = t('The answer you should have chosen.');
```

```

}
/**
 * Implements hook_quiz_access().
 *
 * Control access to Quizzes.
 *
 * @see quiz_quiz_access() for default access implementations.
 *
 * Modules may implement Quiz access control hooks to block access to a Quiz or
 * display a non-blocking message. Blockers are keyed by a blocker name, and
```

```

* must be an array keyed by 'success' and 'message'.
*/
function hook_quiz_access($op, $quiz, $account) {
  if ($op == 'take') {
    $today = date('l');
    if ($today == 'Monday') {
      return array(
        'monday' => array(
          'success' => FALSE,
          'message' => t('You cannot take quizzes on Monday.'),
        ),
      );
    }
  }
  else {
    return array(
      'not_monday' => array(
        'success' => TRUE,
        'message' => t('It is not Monday so you may take quizzes.'),
      ),
    );
  }
}
}
/**
 * Implements hook_quiz_access_alter().
 *
 * Alter the access blockers for a Quiz.
 */
function hook_quiz_access_alter(&$hooks, $op, $quiz, $account) {
  if ($op == 'take') {
    unset($hooks['monday']);
  }
}

```