МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
КАФЕДРА КОМП'ЮТЕРНЫХ НАУК

КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА

на тему:

«Інтерактивний тренажер для дистанційного курсу "Розробка інтернет-магазину мовою PHP"»

Завідувач
випускаючої кафедри **Довбиш А.С.**

Керівник роботи **Шелехов І.В.**

Студент гр. ІК.м-91 **Семеренко Є.С.**

СУМИ 2020

РЕФЕРАТ

Записка: 91 стор., 18 рис., 1 табл., 1 додаток, 11 джерел.

Об’єкт дослідження — процес створення інтерактивного тренажеру для дистанційного курсу.

Мета роботи — розробити систему тестування студентів та дослідити створену систему.

Методи дослідження — в процесі дослідження були використані такі технології як PHP, HTML, CSS, а також фреймворк такий, як Bootstrap.

Результати — розроблено веб-сервіс, за допомогою якого студенти які навчались на умовному курсі “Розробка інтернет-магазину мовою PHP” можуть перевірити та закріпити свої знання і досліджено створену систему. Веб-сервіс розроблений і готовий до використання.

PHP, MVC, AJAX, HTML, CSS, BOOTSTRAP

Факультет *ЕлІТ* Кафедра *Комп'ютерних наук*

Спеціальність *Інформаційно-комунікаційні технології*

Затверджую:

Зав.кафедрою _____

_____

"_____" _____ 20___р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТОВІ

Семеренку Євгенію Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) *Інтерактивний тренажер для дистанційного курсу "Розробка інтернет-магазину мовою PHP"*

затверджую наказом по інституту від "____" _____ 20 ___ р. № _____

2. Термін здачі студентом закінченого проекту (роботи) _____

3. Вхідні данні до проекту (роботи) _____
_____
_____

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити) *1) Огляд існуючих рішень. 2) Постановка задачі. 3) Вибір методу рішення. 4) Проєктування та програмне забеспечення системи. 5) Розробка інформаційної системи*

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

6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання _____

Керівник _____

(підпис)

Завдання прийняв до виконання _____

(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної магістерської роботи	Термін виконання проекту (роботи)	Примітка
1.			
2.			
3.			
4.			

Студент – дипломник _____

(підпис)

Керівник проекту _____

(підпис)

ЗМІСТ

ВСТУП 6
1 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ 8
1.1 Переваги і недоліки систем тестувань 8
1.2 Поняття ефективності тесту 10
1.3 Критерії відбору матеріалу для тестових завдань 12
1.4 Види тестувань 14
1.5 Огляд сайтів для онлайн-навчання та тестування 17
1.6 Постановка задачі 19
1.7 Вимоги до систем дистанційного навчання 19
2 ВИБІР МЕТОДУ РІШЕННЯ 21
2.1 Вибір програмного інструментарію 21
2.1.1 Скриптова мова програмування PHP 21
2.1.2 Паттерн MVC (Model–View-Controller) 24
2.1.3 Каскадні таблиці стилів CSS 29
2.1.4 Технологія AJAX 29
2.1.5 Реляційна система управління базами даних MySQL 31
2.1.6 Серверна платформа Open Server 31
2.2 Опис середовища розробки ПО 32
3 ПРОЄКТУВАННЯ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ 34
3.1 Архітектурна схема програмної системи 34
3.2 Опис функціональності системи 44
3.3 Опис бази даних 45
4 ПРОГРАМНА РЕАЛІЗАЦІЯ 48
4.1 Загальний метод впровадження тестової системи 48
4.2 Призначення системи 48
4.3 Опис користувачів 48
4.4 Тестування користувачів 49
ВИСНОВКИ 53
СПИСОК ЛІТЕРАТУРИ 54
ДОДАТОК 55

ВСТУП

У важкий період нашого соціального розвитку, коли здається є багато невирішувальних проблем, не вистачає кваліфікованих фахівців у всіх сферах нашого життя. Підготовка кваліфікованих фахівців для нашої країни - це проблема, яка повинна бути вирішена системою освіти найближчим часом. У сучасному світі сучасне життя вимагає використання нових прогресивних форм навчання.

Крім того, для всіх форм навчання ми можемо з абсолютною упевненістю сказати, що всі вони використовуються найбільш ефективно і застосовні до кожного потенційного предмета. Кожен повинен мати можливість здобувати освіту та усвідомлювати свою мудрість. Система дистанційного навчання відповідає цим вимогам у системі вищої професійної освіти.

Під дистанційним навчанням слід розуміти навчальну організацію, в якій студент може відвідувати навчальні матеріали та консультації викладача в будь-який час доби, сім днів на тиждень, і де він знаходиться.

В даний час актуальність використання дистанційного навчання більше не піддається сумніву.

Застосування дистанційного навчання має багато переваг, як правило, включаючи такі переваги:

- вміння поєднувати різні форми подання інформації (текст, графіка, анімація, відео, аудіо);
- використання вправ (навчання на практиці);
- можливість адаптації навчальної програми до індивідуальних особливостей учня;
- надати студентам право контролювати розмір та порядок деяких підручників;
- забезпечити технічну базу для гнучкої взаємодії між викладачами та студентами;
- провести ефективну підготовку з "механічної" роботи.

Дистанційне навчання підходить для місць, де важко організувати щось інше. Традиційно він використовується для підготовки майстерності сотень співробітників у різних офісних процедурах. Дистанційне навчання великої компанії дозволяє організувати навчання та перепідготовку одночасно у всіх його філіях, незалежно від віддаленості від штабу.

Дистанційне навчання відрізняється від інших методів: його легко адаптувати до конкретних вимог конкретної організації і навіть до індивідуального рівня підготовки кожного працівника. Тестова система покаже новим студентам, яку частину навчання слід розпочати. Якщо технологію або версію курсу оновлять, система може швидко адаптуватися до цих змін. Отже, дистанційне навчання дозволяє уникнути старіння знань та втрати навичок експертів компанії, що важливо в швидкозмінних технологіях.

Отже, головними перевагами дистанційного навчання є: зменшення витрат на навчання, значне підвищення ефективності навчального процесу, якість, постійна доречність та гнучкі графіки.

1 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

1.1 Переваги і недоліки систем тестувань

Тест є стандартизованим методом діагностичної готовності та структури. У такому тесті всі випробувані оцінювали відповіді з однаковими правилами та за однакових умов і відповідали на однакові завдання одночасно. Основною метою застосування традиційного тестування є встановлення взаємозв'язку між порядками знань між тестувальниками на основі рівня знань, показаного в тесті. І на цій основі визначається позиція (або рівень) на кожному даному тестовому наборі. Іспити визначаються як все більш складні системи завдань, які дозволяють результативно оцінювати рівні та якісно оцінювати структуру підготовки студентів.

Тест має три ознаки: композицію, склад і цілісність. Він складається із норм, правил користування, оцінки кожного завдання та рекомендацій щодо інтерпретації підсумків екзамену. Повнота тесту вказує на взаємозв'язок між завданнями, які належать до загального коефіцієнта вимірювання. Кожне тестове завдання виконує призначену йому роль, тому всі ці завдання не можна видалити з тесту, не втрачаючи якості вимірювань. Структура тесту формується через зв’язок із завданням. Переважно, це так звана факторна побудова, де кожне завдання пов'язане з іншими завданнями шляхом тестування загального змісту та загальних змін результатів тесту.

Традиційне тестування - це єдність щонайменше трьох систем:

- змістовна система знань, описана мовою тестового предмета;
- дедалі складніша система офіційних місій;
- статистичні характеристики завдань та результати дослідження.

Спільнота з єдиною дисципліною сприяє підвищенню якості тестової системи, яка реалізує ідею вимірювання готовності випробуваних до будь-якої конкретної дисципліни. Сукупність таких завдань, обраних відповідно до вимог до тесту, є однорідним тестом, який може вимірювати будь-яку якість (атрибут).

Час часто називають іншим фактором формування системи. Дійсно, одним із міркувань для створення тесту є наявність інструменту, який може швидко та відносно точно оцінювати великі дисципліни.

Потреба в економії часу стала закономірною в процесі масового виробництва, який став освітою. Одним із сучасних напрямків сучасного тестового контролю є персоналізація контролю, що призводить до значної економії часу на тестування. Контроль здійснюється за допомогою попередньо відкаліброваних складних завдань.

Інший аспект проблеми полягає в тому, що вона є суттєвою з самого початку тесту. Це залежить від особливості результатів. Кожен тест має однаковий час тестування, перевищення або зменшення цього часу знизить якість тесту. Відповідно до різниці тестових даних, визначається найкращий час тестування, виходячи з досвіду. Якщо ми відкладемо час тесту на горизонтальну вісь і вертикальну вісь - значення дисперсії результатів тесту, отриманих після з'єднання кожного контролю разом, і пов'яжемо ці точки, ми можемо зрозуміти зміну дисперсії. Це значення вказуватиме на оптимальний час, необхідний для тестуючого контролю.

Деякий зміст означає, що в тесті використовуються лише контрольні матеріали, які відповідають змісту навчального курсу. У будь-якому іншому випадку він не включається в тест. Зміст тесту існує в одній з двох основних діючих форм завдань: зберігання та передача. Позакласна робота не включається в тест.

Зміст іспиту перевіряють досвідчені викладачі, яким пропонується відповісти на основне питання - за допомогою запропонованого завдання, чи можна правильно оцінити рівень знань та структуру цієї предметної групи? При оцінці змісту тесту завжди виникатимуть питання про мету, зміст та якість тесту.

Проаналізувавши зміст завдання та проведений таким чином загальний тест, ви можете визначити знання, вміння, навички та ідеї, необхідні для правильного виконання завдання. При використанні завдань у формі тестів для

атестації випускників навчальних закладів важливо мати завдання, які дозволяють робити висновки про мінімально допустимі здібності випускників.

Традиційно якість тесту знижується, щоб визначити його надійність та достовірність результатів. Якісними та об'єктивними можна назвати лише методи вимірювання, які є науковими та обґрунтованими та можуть дати необхідні результати.

У західній літературі традиційно розглядаються два основні стандарти якості: обґрунтованість та надійність.

Термін дії стосується того, чи підходять результати тесту для цілей тесту. Ефективність залежить від якості задач, кількості завдань, рівня повноти та глибини охоплення змісту предмета (теми) у тестовому завданні. Крім того, ефективність також залежить від складності, збалансованості та розподілу завдання, способу вибору тестового завдання із загальної бібліотеки завдань, освітлення результатів тесту, організації збору даних та вибірки об’єктів.

Надійність тесту - це ступінь узгодженості результатів, коли одна і та ж людина проходить повторне тестування за однакових умов.

1.2 Поняття ефективності тесту

Ефективність можна назвати своєрідним тестом, він кращий за інші тести, він може вимірювати знання предмета таким чином, що менше завдань, краще, швидше та дешевше, і, якщо це можливо, поєднувати все це. Поняття "ефективність" пов'язане та подібне до поняття "оптимальність". Останнє вважається найкращим вибором з точки зору дотримання кількох критеріїв, які приймаються послідовно або спільно.

Ефективний тест не може містити неоптимальних завдань. У цьому випадку закономірно постане питання розмежування дійсних та недійсних завдань. Що стосується змісту, ефективне завдання перевіряє важливі елементи змісту предмета, який часто називають ключем до структури знань предмета. Тому тест включає лише ті завдання, які експерти вважають ключовими елементами предмета, що вивчається.

При визначенні ефективності тесту слід звернути увагу на два ключові елементи - кількість тестових завдань та рівень підготовки випробуваних. Якщо ви вибрали меншу кількість найкращих варіантів з будь-якого тесту з великою кількістю завдань, він може сформувати систему, продуктивність якої не буде значно нижчою, ніж у тесті з великою кількістю завдань.

У цьому випадку тест із меншою кількістю завдань можна сказати відносно ефективним. Крім того, ефективність тесту можна оцінити на основі складності тесту до ступеня готовності до поточного тесту. Враховуючи валідність тесту, можна сказати, що такі оцінки в літературі зазвичай відносять до валідності за рівнем.

Легко зрозуміти справжню безглуздість того грізного завдання, яке дається слабким студентам. На більшість тем може бути дано неправильну відповідь. Те саме стосується і простих іспитів: марно (неефективно) давати його сильнішим студентам, оскільки воно має велику ймовірність, але тепер правильних відповідей, тому оцінки за іспити майже з усіх предметів будуть такі самі. В обох випадках теми не будуть відрізнятися одна від одної. Тому через невідповідність рівня складності тесту до рівня готовності вимірювання проводитись не буде. З цих міркувань легко зробити висновок, що найбільш ефективним тестом є тест, який точно відповідає складності завдання до рівня підготовки випробуваного.

Термін дії тесту може залежати від форми. Порушення тестової форми завжди призведе до поганого висловлення змісту та до поганого розуміння сенсу завдання.

Ефективність тесту також залежить від принципу вибору завдання. Якщо обрано завдання вимірювання, складність якого варіюється у всьому діапазоні, точність вимірювання в певній області зменшиться. І навпаки, якщо ви хочете точно виміряти предметні знання, такі як середній рівень підготовки, вам потрібно буде виконати більше завдань такої складності. Тому тест може виявитися зовсім неефективним у всьому діапазоні підготовки суб'єкта. Це може бути ефективнішим на одному рівні знань, але може бути не таким ефективним

на іншому рівні знань. Це значення було закладено в концепцію ефективності диференціальних випробувань.

Відповідну залежність між складністю тесту та ступенем підготовленості випробуваних можна спробувати оцінити показник обсягу інформації, отриманої в процесі вимірювання.

1.3 Критерії відбору матеріалу для тестових завдань

Розробимо критерії вибору змісту тестових матеріалів:

Значення. Цей принцип показує, що до тесту повинні бути включені лише ті елементи знань, які можна віднести до найважливіших. Без цих знань знання стають неповними, і є багато прогалин. Через свою важливість такі елементи знань можна назвати структурними знаннями. Тому тест повинен включати лише ті матеріали, які служать структурними елементами в особистих знаннях.

Наукова надійність. Тест містить лише зміст предмета, який об'єктивно відповідає дійсності та зазнав деяких раціональних аргументів. Тому не рекомендується включати всі суперечливі погляди (звичайні для наукового співтовариства) до тестового завдання. Суть тестових завдань полягає саме в тому, що вони повинні бути чіткими відповідями, які вчителі знають заздалегідь, та отримувати їх схвалення в процесі об’єктивного формулювання завдань.

Ступінь відповідності змісту іспиту та поточного наукового рівня. Дотримання цього принципу виходить за межі природного попиту на підготовку фахівців та перевірку їх знань із сучасних матеріалів.

Репрезентативність. Тест не тільки містить важливі елементи змісту, але також звертає увагу на повноту та адекватність його контролю. Насправді ви можете взяти від п’яти до шести елементів і перевірити їх розуміння предмету випробуваних. Однак де впевненість у тому, що дисципліна знає інші важливі елементи змісту дисципліни? Шлях до створення цієї впевненості полягає у найбільш повному відображенні необхідних знань у тестовому завданні.

Репрезентативність не обов'язково включає всі важливі елементи змісту в тест. Зрештою, багато з них, очевидно, пов’язані між собою в загальній структурі знань і містяться один в одному повністю або частково. Крім того, багато

елементів у структурі знань підпорядковані ієрархії. Цей принцип відповідає основному завданню, яке найкраще називати структурним завданням.

Складність навчальних матеріалів зростаюча. Цей принцип означає, що кожен навчальний елемент у процесі контролю має певний ступінь середнього показника для предмета, на який спрямована увага вчителя.

Майже всі підручники та посібники базуються на принципі збільшення складності. У логіці, математиці, іноземних мовах, статистиці, філософії та інших дисциплінах розуміння наступних елементів курсу суворо залежить від знання попередніх навчальних елементів. Тому такі дисципліни можна вивчати лише з самого початку, і без прогалин.

Складний зміст зазвичай відповідає важкій задачі. Ті, хто правильно відповів на складні завдання, швидше за все, дадуть відповідь на прості завдання правильно.

Змінність змісту. Зміст іспиту не може залишатися незмінним і не має нічого спільного з розвитком науки, прогресом науки і техніки, новим змістом предмета та новими підручниками. У міру змісту предмета, зміст тесту також повинен змінюватися. При цьому враховується можливість досліджуваних. Якщо тестується слабкий набір тем, то оригінальні складні тестові завдання просто не працюватимуть – жоден суб’єкт не може відповісти на них правильно, тому ці завдання можна вилучити з подальшої обробки та інтерпретації даних тесту. Тестовий вміст слабкого значно відрізняється від тестового вмісту сильного.

Зміст системи. Це означає вибір змісту тестового завдання, який відповідає вимогам системних знань. Окрім відбору завдань із системним змістом, важливо також мати завдання, пов’язані із загальною структурою знань. Це можливо лише в тому випадку, якщо тестове завдання пов’язане із загальною факторною структурою знань. Цей взаємозв'язок визначається за допомогою факторного аналізу.

Складність та збалансованість змісту тесту. Тест, призначений для остаточного контролю знань, не може містити лише матеріал з предмета, навіть

якщо предмет є найбільш критичним предметом у цій дисципліні. Необхідно знайти завдання, які повністю відображають (якщо не всі) основні теми курсу. У той же час люди сподіваються збалансовано відображати основні теоретичні матеріали-концепції законів і моделей, припущення, факти, структурні компоненти теорій і методів наукової та практичної діяльності, а також здатні ефективно вирішувати типові професійні проблеми. Такі завдання можна порівняно легко перетворити у форму тестових завдань для встановлення відповідності або правильного порядку, тим самим ставши навчальною моделлю, яку можна успішно використовувати для контролю та навчання.

Відповідна мета. Зміст тесту залежить від мети тесту. Якщо ви хочете вибрати невелику кількість добре навчених предметів, очевидно, що ці завдання повинні бути складними, адже лише за допомогою таких завдань ви можете вибрати найкращий. Навпаки, якщо вам потрібно видалити найслабшу частину, найкраще це робити за допомогою відносно простих завдань; ті, хто не виконав ці завдання, є найслабшими. Якщо необхідно оцінювати теми у всьому діапазоні знань - від низького до високого, тест повинен включати легкі, помірні та складні завдання.

1.4 Види тестувань

1. Гомогенні тести є більш поширеними, ніж інші тести. У методі навчання метою його створення є контроль знань певного предмета чи предмета (наприклад, тривимірних предметів, наприклад, фізики). У подібних навчальних тестах не дозволяються завдання, що розкривають інші атрибути. Наявність останнього порушує вимоги щодо чистоти викладання навчальних тестів. Зрештою, кожен тест вимірює заздалегідь визначене значення.

Наприклад, тести з фізики можуть вимірювати знання, вміння, навички та сприйняття в цій галузі науки. Однією з труднощів цього вимірювання є те, що знання з фізики дуже пов'язані з математикою. Тому тестування з фізики може вміло встановити рівень математичних знань, що використовуються для вирішення задач з фізики. Перевищення допустимих рівнів призведе до змін результатів; коли вони перевищують межу, останні починають все більше

покладатися не тільки на фізичні знання, але і на іншу науку, математичну. Іншим важливим аспектом є те, що деякі автори сподіваються, що перевірка знань, включена в тест, полягає не лише у здатності вирішувати фізичні проблеми, щоб включити інтелектуальний компонент у міру фізичної підготовленості.

2. Гетерогенне тестування - це система, завдання якої стають дедалі складнішою.Це система спрямована на об’єктивну, якісну та ефективну оцінку структури студентів з різних дисциплін та вимірювання рівня їх підготовки. Ці тести зазвичай включають інтелектуальні завдання для оцінки рівня інтелектуального розвитку.

Як правило, неоднорідні тести використовуються для проведення всебічного оцінювання випускників шкіл, оцінки особистості під час прийняття на роботу та відбору найбільш добре підготовлених претендентів на вищу освіту. Оскільки кожен гетерогенний тест складається з подібних тестів, інтерпретація результатів тесту ґрунтується на відповіді на кожне тестове завдання (тут називається шкалою) .Крім того, використовуються різні методи агрегування балів, щоб спробувати оцінити випробовуваних в цілому.

3. Інтегрований тест

Інтеграційне тестування можна назвати тестом, що складається із системи завдань, що відповідає вимогам інтеграційного змісту та форми тесту. Ця мета - поставити комплексний остаточний діагноз для підготовки випускників навчальних закладів, тим самим збільшуючи складність завдання. Діагностика проводиться шляхом подання таких завдань, а правильна відповідь вимагає всебічних (широких, взаємозалежних) знань з двох або більше дисциплін. Цей тип тесту підходить лише для вчителів, які знають багато предметів, розуміють важливу роль міждисциплінарних зв’язків у навчанні та можуть створювати завдання, знання учнів з різних предметів та правильні відповіді для правильного застосування цих знань.

4. Адаптивне тестування

Адаптивне тестування - це варіант автоматизованих систем тестування, в яких параметри складності відомі заздалегідь і можливість розрізнити кожне завдання. Система створюється у вигляді комп'ютерних завдань і розміщується відповідно до цікавих характеристик завдань.

Найважливішою особливістю адаптивного тестового завдання є його складність, яка досягається експериментами, а це означає, що перед входом до банку кожне завдання вимагає емпіричного тесту на значній кількості зацікавлених типових студентів.

5. Стандартно-орієнтоване тестування

Якщо основним завданням є з’ясувати, які елементи змісту предмета засвоєні конкретним випробуваним, то це стосується предметно-навчального методу інтерпретації результатів тесту. Загальний набір завдань (англ. Domain), який суб'єкт знає і не знає. Інтерпретація результатів здійснюється вчителем мовою предмета. Висновок будується за логічним ланцюгом: зміст предмета - загальне завдання вимірювання знань - тест як зразок цього комплексу завдань, відповідь суб'єкта - ймовірний висновок про його знання предмета. Для зосередження уваги на цьому виді тесту потрібна велика кількість завдань і досить повне визначення змісту предмета дослідження. Інтерпретацію результатів здійснює вчитель-предметник.

Суперечка ведеться навколо двох основних питань:

1. Правильність змісту іспиту, що означає точність формулювання завдання, обґрунтованість науки про предмет і чи можна прийняти іспит для перевірки цікавих знань з цього виду предмета. Виступаючи за складання іспиту, вчителі-предметники повинні покладатися на чіткіші інструменти, мову та принципи та, як правило, залежати від свого розуміння предмета. У такому випадку, говорять про тести з орієнтованою на зміст інтерпретацією результатів.

2. Відповідно до результатів тестування предмета невеликої частини тестового завдання, оцінюється ефективність знань усього предмета; загальна вибірка всіх потенційних або реальних завдань може бути надана випробуваним для надійного та обґрунтованого оцінювання . По суті, це питання перевірки

точності індуктивних міркувань щодо великої кількості питань на основі відповідей на невелику кількість тестових завдань.

Другий тип тестування включає зосередження уваги на таких конкретних цілях і задачах, наприклад, перевірка відносно коротких необхідних знань, умінь та навичок рівня оволодіння, які можуть бути використані як встановлені стандарти або стандарти для засвоєння. Наприклад, для атестації випускників навчальних закладів важливо мати завдання, яке може зробити висновки про мінімально допустимі здібності випускників. При перевірці мінімального рівня знань зміст завдання буде радикально спрощений.

1.5 Огляд сайтів для онлайн-навчання та тестування

Предметна область - це система перевірки знань учнів. Теорія тест-системи та імітаційної системи є однією з найдосконаліших теорій викладання.

Сучасна педагогічна наука вважає тестування основою перевірки контролю знань. Це основа навчальних систем у багатьох західних країнах.

Широко відомі приклади, такі як:

- EdX

EdX - провідний міжнародний онлайн-освітній портал, створений Гарвардським університетом та Массачусетським технологічним інститутом, що дозволяє кожному отримати доступ до курсів та лекцій найкращих університетів світу.

Надані послуги: тисячі онлайн-курсів англійської мови, лекцій та програм від таких гігантів, як Гарвардський університет, Массачусетський технологічний інститут, Каліфорнійський університет, Берклі, Вашингтонський університет та інші.

- Лекториум

Лекториум дозволяє старшокласникам, студентам та професіоналам отримувати знання, необхідні для академічного та професійного розвитку.

Надані послуги: онлайн-курси МООС та відеолекції у понад 20 провідних російських університетах.

- Coursera

Coursera - це унікальна міжнародна платформа дистанційної освіти, яка забезпечує курси кращих університетів світу.

Надані послуги: більше 2000 онлайн-курсів із 146 шкіл, включаючи Манчестерський університет, Принстонський університет, Єльський університет та Стенфордський університет.

- Udacity

Освітній портал Udacity був створений професором Стенфордського університету і є провідним веб-сайтом з питань навчання в галузі технологій, бізнесу та IT.

Надані послуги: спеціалізовані та унікальні платні плани (термін менше 12 місяців) та безкоштовні курси з IT-технологій, розробки програм, інженерії, програмування та інших предметних областей.

- Stepic

Портал Stepic пропонує відкриті інтерактивні лекції та курси з масштабними предметними курсами.

Надані послуги: курси російської та англійської мови з гуманітарних, точних та природничих наук, інформатики та статистики.

- MIX.СумДУ

Середовище змішаного навчання в Сумському Державному Університеті, де проводиться дистанційне навчання та тестування студентів.

Надані послуги: різні курси в залежності від спеціальності студента, лекції та дистанційні тестування по цим курсам, є зв’язок з особистим кабінетом.

- Canvas Network

Спільно з Canvas Network ви можете дізнатись усі найцікавіші та необхідні знання не тільки в університетах, а й у провідних компаній та організацій.

Надані послуги: багато безкоштовних та багато сертифікованих навчальних курсів, призначених для вивчення конкретних предметів (математики, аналізу, авіації, економіки тощо) та розвитку загальних знань.

Основні функції тест-системи включають:

- реєстрація випробуваних (ім'я, прізвище, ідентифікатор, пароль тощо);
- додавання системи правил проведення тестів (короткий опис цілей і завдань, правила щодо матеріалів, умови доступу до системи тестування тощо);
- система подання питань та відповідей на запитання (наприклад: одне питання-три варіанти відповіді-одна правильна відповідь);
- система зберігання, додавання та редагування питань та відповідей на запитання;
- статистична система;
- система видачі сертифікатів;
- система реєстрації іспитів у глобальній мережі викладання.

Система тестування знань абітурієнтів і студентів є частиною тренажерно-тестуючої системи СумДУ.

1.6 Постановка задачі

У цій освітній дослідницькій роботі ставиться завдання використання Інтернет-технологій для створення системи освіти та іспитів.

Система повинна забезпечити навчальні матеріали з курсу "Розробка інтернет-магазину мовою PHP" та пройти тести на об'єднання матеріалів цього курсу. Отже, система навчання та тестування повинна бути розроблена та спроектована, як система, що містить дві основні підсистеми:

- підсистема підготовки, призначена для випуску навчальних матеріалів;
- діагностична підсистема, що використовується для перевірки ефективності роботи учнів.

1.7 Вимоги до систем дистанційного навчання

Рішення для дистанційного навчання повинно відповідати наступним вимогам:

- комплексність, яка охоплює всі етапи навчання та всіх учасників навчального процесу, учнів та викладачів;

- налаштованість, тобто адаптування до потреб замовника, змінюючи налаштування та додаткові програмні компоненти;
- забезпечення простого та зрозумілого користувальницького інтерфейсу для студентів та викладачів, які можуть не бути IT-професіоналами;
- забезпечення можливості моніторингу успішності студентів зацікавлених сторін.

Розвиток Інтернету та зростання його інформаційно-комунікаційних можливостей сприяли швидкому розвитку дистанційного навчання. Однак дистанційна технологія, впроваджена в навчальний процес, вимагає більш ретельної практики методів засвоєння знань та аналізу пріоритетності факторів, що впливають на ефективність роботи вчителів у віддаленому середовищі.

Система дистанційної освіти може і повинна посідати місце в системі освіти, оскільки, маючи свій компетентний орган, вона може забезпечити якісну освіту, яка відповідає вимогам сучасного суспільства.

2 ВИБІР МЕТОДУ РІШЕННЯ

2.1 Вибір програмного інструментарію

2.1.1 Скриптова мова програмування PHP

PHP - це мова сценаріїв загального призначення, яка широко використовується для розробки веб-додатків. В даний час взаємодіє з великою кількістю провайдерів хостинг-послуг і є одним з лідерів мов програмування для розробки веб-сайтів. Ця мова програмування спеціально розроблена для веб-розробки і може бути реалізована безпосередньо в коді веб-програми.

Основна відмінність цієї мови сценаріїв від сценаріїв, написаних, наприклад, на C ++ або Perl, полягає в тому, що програмний код створюється не програмою, яка генерує HTML-код, а кількома вбудованими командами PHP. Для реалізації коду PHP він відокремлюється спеціальними початковими та кінцевими тегами. Процесор PHP використовує ці теги для визначення початку та кінця частини HTML-коду, що містить команди PHP. [1]

Мова програмування PHP включає безліч вбудованих функцій, таких як: функції файлової системи, функції HTTP, функції дати та часу, функції обробки рядків та масивів тощо. Кожен програмний продукт, написаний на PHP, містить набір спеціальних конструкцій. Такою конструкцією може бути будь-який елемент, що використовується в PHP-коді, наприклад, цикли, оператори, функції тощо. Головною особливістю мови є те, що вона може добре взаємодіяти з усіма сучасними веб-технологіями та підтримує велику кількість сучасних веб-протоколів, а саме: протокол прикладного рівня для доступу до електронної пошти IMAP, протокол FTP, протокол POP, протокол SNMP тощо. PHP також дуже підходить для використання баз даних. Більшість сучасних баз даних підтримують мову сценаріїв PHP.

Сценарій, написаний на PHP, може містити 10000 рядків коду і може складатися з одного рядка - все залежить від того, яке завдання вам потрібно вирішити за допомогою цього сценарію. Щоб почати виконувати PHP-код, його потрібно позначити спеціальними символами (позначками), які ініціюватимуть

послідовність операцій (<?), і продовжуватимуть виконувати ці операції, поки позначка (?>) не буде закрита.

Для розробників PHP він надає гнучкі та ефективні засоби захисту, які можна розділити на дві категорії: інструменти на системному рівні та інструменти на рівні програми. Механізм захисту, реалізований в PHP, управляється адміністратором і може забезпечити максимальний захист після належної настройки. Подібним чином, мова сценаріїв PHP реалізує так званий режим безпеки, який обмежує можливості певних категорій користувачів використовувати код. Наприклад, безпечний режим може обмежити максимальний час, необхідний для запуску сценаріїв, та обсяг використаної пам'яті, оскільки неконтрольоване споживання пам'яті може негативно позначитися на роботі веб-сервера. Крім того, адміністратор може встановити обмеження на використання каталогів, за допомогою яких користувачі можуть переглядати інформацію та виконувати PHP-скрипти, а також може використовувати PHP-скрипти для перегляду конфіденційної інформації на веб-сервері.

Мовні функції PHP включають безліч надійних механізмів шифрування. Крім того, сценарії, написані на PHP, сумісні з багатьма програмами незалежних компаній. Це полегшує інтеграцію із захищеною технологією електронної комерції. [2]

Ще однією перевагою цієї мови є те, що вихідний текст сценарію PHP можна переглядати в браузері, оскільки він компілюється безпосередньо перед надсиланням на запит користувача. Як результат, це виконання сценарію не дозволяє користувачеві викрасти текст коду вихідної команди PHP. Щодо потреб розробників, мова PHP має гнучкість та ефективність конфігурації.

Оскільки PHP не містить коду для певного веб-сервера, його можна використовувати з сучасними серверами (тобто сервером Microsoft IIS, сервером Netscape Enterprise, сервером Apache, сервером Stronghold тощо).

Принципи ООП в PHP

Основні поняття та принципи ООП.

Принципом ООП (об’єктно-орієнтоване програмування) є парадигма, заснована на методі, заснованому на об’єктній моделі.

Об’єкт - це сукупність характеристик та поведінки, що відповідають сутності. Для створення об’єкта в програмі слід описати шаблон об’єкта, який називається класом. Об'єкти створюються на основі існуючих класів. Характеристики об’єктів класу називаються змінними, які називаються полями. Поведінка об'єктів у класі описується функціями, які називаються методами.

Об'єктно орієнтоване програмування базується на принципах:

- інкапсуляція
- успадкування
- абстракція
- поліморфізм

Інкапсуляція - метод поєднання полів та методів у класі, щоб вимкнути прямий доступ до полів та увімкнути їх для управління цими полями. Відповідно до пакету, рекомендується закрити доступ до поля за допомогою відповідного специфікатора доступу. Створіть спеціальні методи, що дозволяють вносити зміни або отримувати значення полів. Метод повернення значення поля називається геттером (від англійського get - «отримувати»). Метод встановлення значення поля називається сеттером (від англійського set - «встановлювати»).

Метод SET повинен перевіряти правильність значення, введеного в поле, а якщо значення неправильне, виправляти його або генерувати помилку. Наприклад, одна хвилина не може перевищувати 60 секунд тощо.

Спадкування - дозволяє створювати класи на основі існуючих класів. Це спрощує завдання створення нових класів з використанням існуючого програмного коду. Успадкований клас називається базовим класом або батьківським класом. Класи, похідні від базового класу, називаються нащадками, спадкоємцями або похідними класами. PHP також використовує абстрактні класи.

Абстрактний клас - це клас, який містить принаймні один абстрактний метод. Він описаний у програмі, містить поля та методи і не може використовуватися для безпосереднього створення об'єктів. Іншими словами, його можна наслідувати лише з абстрактних класів. Створюйте об'єкти лише на основі похідних класів, успадкованих від абстракції.

Поліморфізм (грец. "Різноманітність форми") - під програмуванням розуміється можливість використання одного і того ж імені для методів різних класів, що мають однакову ієрархію успадкування (тобто у споріднених класах) для виконання подібних операцій. Метод, який створює те саме ім'я, що і батьківський метод у похідному класі, називається методом перевизначення.

2.1.2 Паттерн MVC (Model–View-Controller)

Безпосередньо при створенні онлайн-магазину по курсу ми будемо використовувати паттерн MVC.

Основа MVC була закладена для Smalltalk в 1978 році. Це дуже просто і враховує реальність часу. Якщо перебільшити, контролер - це пристрій введення - клавіатура та її обробник (наприклад, драйвер). Як тільки що-небудь відбувається в контролері (наприклад, користувач натискає кнопку), подія / дані передаються в модель. У моделі вирішується, що саме потрібно зробити із вхідною інформацією. Наприклад, якщо натиснути клавішу Enter, виконується введена команда. Після виконання дії вам потрібно його якось відобразити. За це вже відповідає подання, у найпростішому випадку це звичайний монітор, принтер або файл. Головне - відокремити модуль вводу-виводу від усього коду і зберегти основну функцію програми як модель.

З часом MVC прийняв більш абстрактну форму, але загальна концепція не змінилася.

- Контролер вказує моделі, що потрібно зробити.
- Модель виконує завдання.
- Вид відображає / віддає результат.

У класичних додатках реалізація Model-View-Controller дуже проста, оскільки вони складаються зі статичних інтерфейсів. На етапі розробки програми

заздалегідь буде створена форма, яка містить кнопки, мітки, меню та інші елементи, що не змінить її форму чи дизайн під час виконання програми. Тому дуже легко взаємодіяти з користувачем через контролер. Контролер розуміє, що при натисканні кнопки потрібно виконати певну операцію. Але з користуванням PHP стає все складніше.

Робота Web-додатку.

Сайт може по-різному взаємодіяти з користувачами. Наприклад, натискання викликає кнопку. Іншими словами, хоча програмне забезпечення є кодом HTML, контролер повинен відстежувати кнопку onClick.

Однак усі сайти мають одне спільне - усі вони взаємодіють за допомогою URL-адрес. Будь-яке посилання означає операцію. Наприклад, для виведення домашньої сторінки чи запису заголовка або надсилання форми на основі запиту ajax (також URL-адреси). Тобто у всіх випадках користувачі надсилають свої дії у формі посилань (http-запит).

Інший варіант - приховати передачу даних через POST (Ajax). Адреса сторінки не змінилася, хоча обробник повинен якимось чином зрозуміти, якщо це запит на публікацію, потрібно зробити одне, якщо його отримати, потрібно лише роздрукувати сторінку за замовчуванням. Отже, у веб-застосунку контролер повинен мати функцію маршрутизації. Маршрутизація реалізується по-різному, але у всіх випадках URL-адресу потрібно спочатку проаналізувати, наприклад, за допомогою parse_url() та визначення методу передачі даних ($ _POST тощо).

Маршрутизація в контролері визначає подальші дії, які тісно пов'язані зі структурою програми. Наприклад, ви можете викликати функції, вкладати файли або виконувати php-подібні методи.

У будь-якому випадку контролеру потрібно точно знати, що робити далі і який функціональний модуль завершить роботу. Контролер може також передавати вхідні дані в модель, і модель вже визначала, який модуль буде виконувати завдання. Насправді це одне і те ж, різниця лише в розміщенні цього «списку дій».

Після того, як управлінські права передаються моделі, починається основна робота, тобто ціль відвідувачів цього сайту. Припустимо, це оригінальний висновок про дату та час. У моделі ви можете негайно згенерувати необхідний html-код для виводу, але це порушить концепцію MVC, оскільки модель повинна надавати лише дані, а сама візуалізація повинна виконуватися у Представленні.

Вид / Представлення - це, по суті, остаточний html-код, який розроблений для необхідного дизайну. Наприклад, якщо ви хочете відобразити дату іншими кольорами, вам потрібно лише змінити стиль CSS у шаблоні. У цьому випадку ні контролер, ні модель не зміниться.

Проблеми View.

Тут ми стикаємося з проблемою: як точно перенести дані в HTML-код? Зрештою, HTML є статичним - це не мова програмування, а просто розмітка документа.

Навіть у найпростішій реалізації, шаблон html потрібно зробити у вигляді php-файлу, де ви повинні змішувати html і php-код між собою. У простих випадках це не проблема, але коли для виведення є велика кількість даних, це змішування вже є серйозною проблемою (погана читабельність та складна підтримка). На жаль, хоча існують способи спростити написання коду цього типу (див. нижче), вони все одно не можуть вирішити проблему.

Неправильне розуміння MVC.

Інтерпретація спотворень модель-вигляд-контролер є дуже поширеною. Проблема полягає у фактичній реалізації: PHP - це дуже гнучка мова, яка дозволяє використовувати будь-який метод. Не думаю, що хтось поспішатиме переписувати свої проекти під суворим MVC, але розуміння цієї концепції стане в нагоді при створенні нового MVC.

Як правило, під моделлю розуміють лише використання бази даних. Іншими словами, контролер викликає певні функції / методи Моделі, які повертають вихідні дані. Проблема тут полягає в тому, що хтось повинен мати справу з цими даними. Тобто в цьому випадку модель не виконала всю роботу, і

дані повинні оброблятися у поданні або тому ж контролері. В обох випадках це вже проблема.

Схема роботи MVC

Використання бази даних - це лише частина роботи моделі. Ні Controller, ні View не повинні знати жодної інформації про базу даних та способи отримання даних. Для них модель - це чорний ящик, куди ви можете передати введені дані та отримати готові.

Ще одним спотворенням MVC будемо вважати, що контролер відповідає за всю логіку та операції програми. З одного боку, він справді відповідає за маршрутизацію, але логіки виконання не повинно бути. По правильному, контролеру навіть не потрібно розуміти вигляд. Коли контролеру присвоюються різні ролі, це інша концепція: Model-View-ViewModel, Model-View-Presenter тощо.

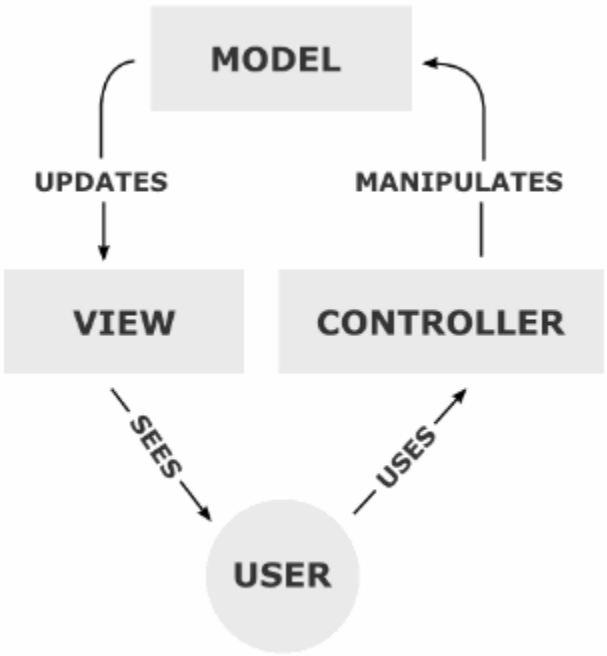

Рис. 2.1.1 – MVC модель

Модульність PHP.

Насправді прийнято розділяти php-код на окремі файли. Цей метод дозволяє уникнути брудного коду та уникнути відповідальності за кожну дію. Не

лише відокремлення файлів, але й групування файлів в окремі каталоги також є загальноприйнятою практикою.

Якщо проект базується на ООП, зазвичай це завдання організації декількох класів. Наприклад, у CodeIgniter 2/3 кожен клас є окремим файлом, але всі файли всіх класів знаходяться у спільному каталозі. Це накладає певні обмеження, оскільки вимагає розміщення всього коду, що належить класу, в одному файлі.

CodeIgniter 4 вже використовує модульний підхід, коли класи розміщуються в окремих каталогах і можуть містити кілька файлів.

MVC - це лише концепція, тому фактична реалізація може бути іншою. Зазвичай використовується ООП, оскільки це вирішує проблему просторів імен у PHP, а модуляризація спрощує підтримку класу. Хоча використання класів та ООП не означає дотримання MVC.

Архітектура програми.

Чи можна застосувати строгий MVC в PHP? Якщо проблема основна, то в PHP можна легко реалізувати один із шаблонів дизайну, реалізований MVC: Спостерігач, Стратегія та Компоновщик. Все це взаємодія між класами в ООП, з яким PHP взагалі не має проблем.

Але з моєї точки зору, загальна архітектура програми та фактична реалізація окремих модулів будуть важливішими. Наприклад, у тій же конфігурації CodeIgniter 2/3, представленій звичайним масивом, тоді як у версії 4 - це клас PHP, який можна використовувати з тим самим масивом, але вже є внутрішнім. Іншими словами, це суворіше дотримання теорії, але з практичної точки зору - код безглуздо розширюється. Іншими словами, у багатьох випадках зручніше і простіше використовувати ідіоматичний require_once() замість створення складного завантажувача класу php.

Тому перед створенням веб-додатку важливо врахувати ці нюанси та підтримувати баланс між зручністю та можливістю подальшого розвитку проекту.

2.1.3 Каскадні таблиці стилів CSS

CSS або каскадні таблиці стилів - це офіційна мова, що використовується для опису зовнішнього вигляду документів, написаних мовою розмітки. Розробники веб-сайтів використовують CSS для встановлення шрифтів, положення шрифтів, що використовуються в процесі розробки на сторінці, та інших принципів появи блоків веб-сторінок. Основним завданням, яке потрібно вирішити при створенні CSS, є розділення вмісту, що, в свою чергу, може збільшити доступність документа, забезпечити більшу гнучкість та контроль над поданням веб-сторінки та зменшити складність його структури вмісту. Крім того, каскадні таблиці стилів дозволяють представити одну і ту ж веб-сторінку в різних стилях відображення (наприклад, візуалізація екрана, читання голосу тощо). CSS складається з набору правил. У свою чергу, кожне правило містить один або кілька комутаторів, розділених комами, та блок визначення, що складається з фігурних дужок ({}) та набору атрибутів та їх значень. [3]

При безпосередньому відображенні веб-сторінок таблиці стилів можуть надходити з кількох джерел. Ієрархія стилів включає:

- стиль автора (інформація про стиль, надана розробником сторінки);
- на замовлення;
- стиль браузера.

Крім того, каскадні таблиці стилів надають можливість обробляти дизайн шрифту документа на вищому рівні, ніж стандартний HTML-код, уникаючи використання графіки для обтяження сторінок.

2.1.4 Технологія AJAX

AJAX розшифровується як асинхронний Javascript та XML. Насправді AJAX - це не нова технологія, оскільки Javascript та XML існують давно, а AJAX - це синтез технології розмітки. AJAX найчастіше асоціюється з терміном Web 2.0 і вітається як остання веб-програма.

При використанні AJAX немає необхідності щоразу оновлювати всю сторінку, оскільки оновлюється лише певна частина сторінки. Оскільки довго чекати не доводиться, це зручніше, а оскільки не у всіх необмежений Інтернет,

він економічніший. Однак у цьому випадку розробник повинен переконатися, що користувач знає, що відбувається на сторінці. Ви можете використовувати індикатор завантаження для обміну текстовими повідомленнями із сервером для цього. Ви також повинні розуміти, що не всі браузери підтримують AJAX (старіші браузери та текстові браузери). Крім того, користувачі можуть вимкнути Javascript. Тому не слід зловживати технологіями та використовувати інші методи для представлення інформації на сайті.

Давайте підсумуємо переваги AJAX:

- Можливість створення зручного веб-інтерфейсу;
- Активна взаємодія з користувачами;
- Часткове перезавантаження сторінки, а не всієї;
- Простий у використанні.

AJAX використовує два методи обробки веб-сторінок: модифікація веб-сторінки без її перезавантаження та динамічний доступ до сервера.

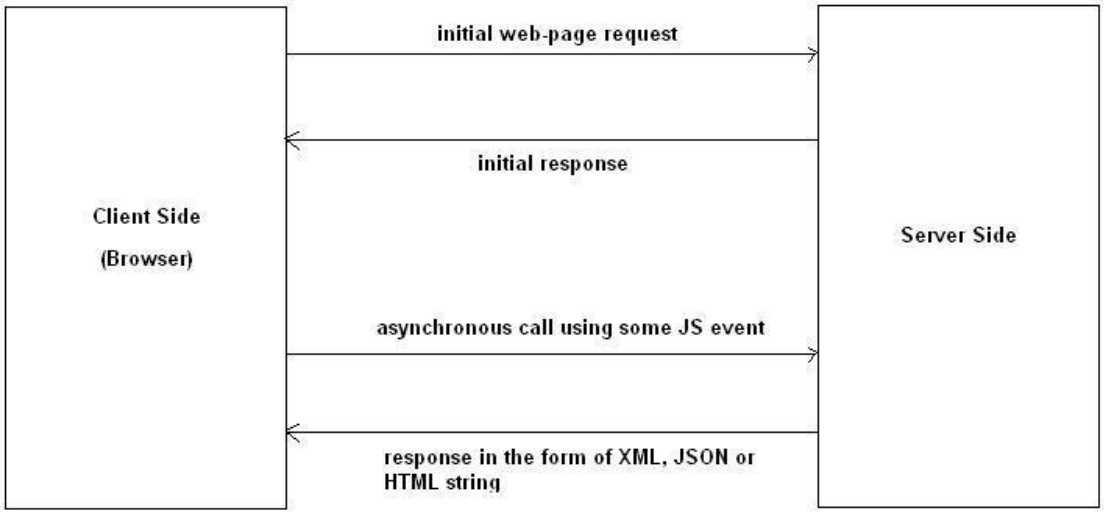

Рис. 2.1.2 – Загальний виклик AJAX

2.1.5 Реляційна система управління базами даних MySQL

В даний час MySQL є однією з найнадійніших, швидких, якісних та найвідоміших баз даних серед усіх існуючих сучасних систем управління базами даних. Основною причиною цього є те, що він розповсюджується безкоштовно разом із вихідним кодом, а інша причина полягає в тому, що MySQL є досить швидкою базою даних.

Основною функцією бази даних MySQL є використання мови структурованих запитів-SQL як основної мови в управлінні базами даних, а саме: створення або видалення таблиць у базі даних, вибірка з бази даних та безпосереднє заповнення таблиць даними.

MySQL має характеристики високої стабільності, високої швидкості, простоти встановлення та використання, а вихідний код сервера компілюється на багатьох платформах, тому база даних MySQL є хорошим рішенням для невеликих програм. [4]

Основні функції MySQL:

1. Дозволяє необмеженій кількості користувачів одночасно використовувати базу даних;
2. Кількість рядків у таблиці може досягати 50 мільйонів;
3. Високошвидкісне виконання команд користувача;
4. Проста та ефективна система безпеки.

Ще однією перевагою перед іншими базами даних є те, що MySQL може використовуватися разом із SQL у стандарті ANSI 92 і має безліч встановлених розширень стандарту, які не надаються жодною іншою системою управління базами даних.

Недоліком цієї бази даних є відсутність підтримки вкладених запитів, таких як SELECT * FROM table_1 WHERE id IN (SELECT id FROM table_2). Немає підтримки транзакцій, а також тригерів та збережених процедур.

2.1.6 Серверна платформа Open Server

На перший погляд здається, що деякі починаючі веб-розробники вважають, що Open Server - це лише заміна Денверу, але це не так. Окрім

розширених функцій, однією з головних переваг Open Server є його повна автономність, тобто незалежність від конкретного комп'ютера. Ми можемо використовувати програму де завгодно. Крім того, крім основної серверної платформи, ми також отримуємо повний набір найнеобхідніших портативних програм, які можна викликати з будь-якого приводу.

Open Server - це портативний локальний портативний сервер WMA / WNMP з багатофункціональними драйверами та великою кількістю плагінів. Запропонований пакет - це не чергова аматорська асамблея, зібрана «на колінах», це перший професійний інструмент, створений спеціально для веб-розробників з урахуванням їхніх пропозицій та побажань.

Головною перевагою Open Server є зручність та простота використання. Підтримка швидкого запуску та зупинки, доступ до шаблону конфігурації модуля, встановлення віртуального диска, використання командного рядка, журналу, налаштування конфігураційного файлу.

Домени можна створювати, створюючи каталоги, локальні субдомени, підтримуються SSL, псевдоніми, кириличні домени, мережеві IP-адреси, захист серверів від зовнішнього доступу та багато інших корисних та цікавих функцій. Open Server не вимагає інсталяції і може запускатися з будь-якого портативного носія.

2.2 Опис середовища розробки ПО

Програмне забезпечення JetBrains PhpStorm - це спеціалізований інструмент веб-розробки, який орієнтований на веб-програми та інші типи програм, які можна створювати за допомогою PHP, а також HTML, JavaScript та CSS. Рішення PhpStorm розгортає та синхронізує проекти за допомогою FTP. Функції, що надаються середовищем PhpStorm, можуть автоматично доповнювати структуру мови PHP у коді, перевіряти код, мають в наявності різні алгоритми реконструкції та швидку навігацію кодом.

Графічний налагоджувач PHP, реалізований у PhpStorm, підтримує умовні точки зупинки, відстеження значень та автоматичне ведення журналу окремих процесів. Тестова програма підтримує структуру модуля тестування PhpUnit та

графічний інтерфейс для запуску тестів. Під час редагування коду буде виділено структуру синтаксису та виконано розширене форматування конфігурації, виявлення помилок у реальному часі та завершення коду. Редактор PhpStorm розгляне коментарі до коду, коли код буде заповнений, щоб автоматично вибрати найкраще рішення проблеми. Рефакторинг PHP та редагування шаблонів можуть забезпечити зміну проекту за найкоротший час. PhpStorm дозволяє візуалізувати код в ієрархічній формі та забезпечує швидку навігацію по всіх елементах.

За допомогою тестування PHPUnit ви можете швидко переглянути результати генерації коду для одного блоку або всієї програми. Якщо тест не вдається, продукт дозволяє переглядати окремі рядки коду, де була виявлена помилка. PhpStorm забезпечує налагодження JavaScript та надає безліч функцій: пошук точок зупинки в HTML та JavaScript, налаштування параметрів точки зупинки, тестування синтаксису коду в реальному часі тощо

За перевірку коду відповідають сотні інспекцій, і проект повністю аналізується під час розробки. Підтримує PHPDoc, впорядковувач коду, переформатування коду, форматування коду з конфігурацією стилю коду та інші функції, які можуть допомогти розробникам писати стислий та простий у обслуговуванні код.

PhpStorm включає всі функції WebStorm (редактор HTML / CSS, редактор JavaScript), а також додану повнофункціональну підтримку PHP та бази даних / SQL.

3 ПРОЄКТУВАННЯ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

3.1 Архітектурна схема програмної системи

Для досягнення цього завдання було вирішено використовувати трирівневу архітектуру, яка складається з таких компонентів: клієнт, сервер та база даних. Схема цієї архітектури наведена на рисунку 3.1.1.

Основним центром програмного забезпечення є сервер. Він централізує базову бізнес-логіку та логіку доступу до бази даних. Завдяки використанню серверів можна ідентифікувати користувачів для надання індивідуального доступу до програмних додатків. Сервер - це єдиний зв’язок між користувачем та базою даних, який може запобігти пошкодженню та зловживанню даними. Для того, щоб користуватися програмою, потрібно увійти в систему. Ця логіка реалізована на рівні сервера, оскільки на рівні користувача ви можете змінювати дозволи доступу або інші неконтрольовані методи доступу до даних.

Під час використання цієї програми користувач взаємодіє з клієнтською програмою (в даному випадку веб-сайтом). На рівні користувача реалізований інтерфейс, за допомогою якого ви можете налаштувати програму та переглянути результати роботи. Перед відправкою на сервер попередня обробка даних також проводиться на рівні користувача, і результати сервера також обробляються. На цьому рівні це також перший етап аутентифікації користувачів, який обмежує неконтрольований доступ до програми.

Важливим завданням рівня бази даних є забезпечення того, щоб сервер зберігання зберігав дані для подальшого використання. Ви також можете забезпечити цілісність даних за допомогою зовнішніх посилань та ключів. На рівні бази даних ви також можете реалізувати бізнес-логіку, яка не вимагає використання зовнішніх джерел даних (крім самої бази даних та її таблиць).

Традиційна архітектура клієнт-сервер складається з двох рівнів або посилань: клієнта та сервера. В останні роки трирівнева архітектура клієнт-сервер стає все більш поширеною (рис. 3.1.1). У цій архітектурі прикладне

програмне забезпечення поділяється на три типи комп’ютерів: комп’ютери користувачів, проміжні сервери та сервери баз даних. Машина користувача є замовником, а в трирівневій моделі вона, як правило, є «тонким» замовником. Проміжна машина по суті є шлюзом між "тонким" клієнтом та різними серверами баз даних. Проміжний комп’ютер повинен вміти перетворювати протокол і відображати певні типи запитів до бази даних до інших типів. Крім того, проміжний комп'ютер повинен поєднувати результати запитів з різних джерел даних. Нарешті, ці комп’ютери повинні виступати шлюзом між настільними програмами та системою управління базами даних, яка існує в організації з давніх часів, тим самим виступаючи посередником між двома «світами». По суті, ця архітектура означає інтеграцію корпоративних програм (Enterprise Application Integration, EAI). Взаємодія між проміжним сервером та сервером баз даних також відповідає моделі клієнт-сервер. Отже, проміжна система виступає як клієнтом, так і сервером.

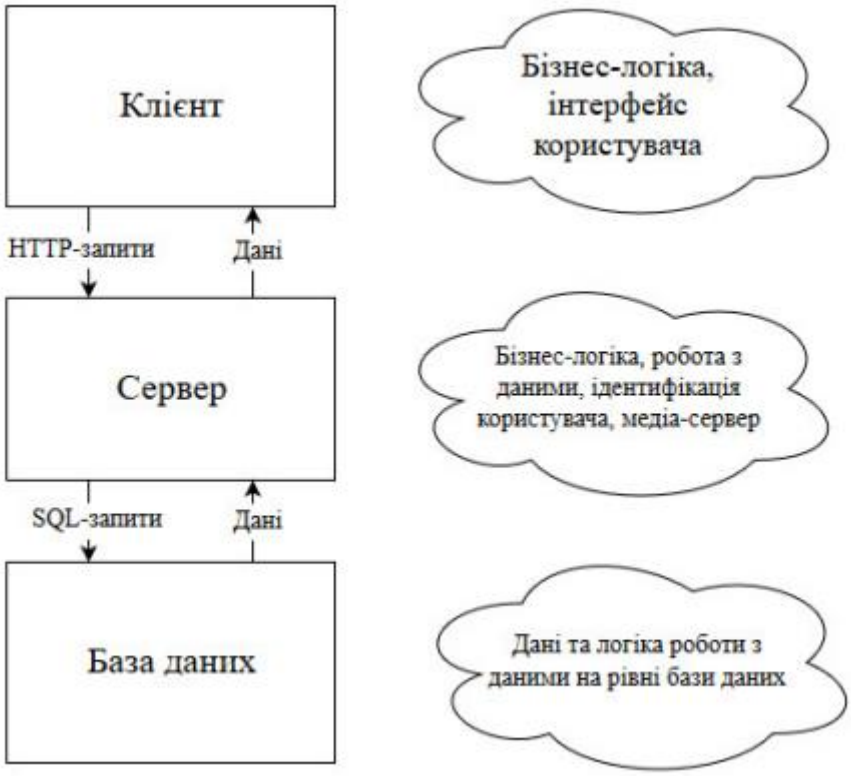

Рис. 3.1.1 – Триланкова архітектура програмного комплексу

Проміжне програмне забезпечення

Процес розробки та розгортання програмних продуктів, пов'язаних з архітектурою клієнт-сервер, значно перевершив процес стандартизації всіх аспектів розподілених обчислень від фізичного рівня до рівня додатків. Відсутність стандартів ускладнює впровадження інтегрованих конфігурацій, що складаються з пристроїв різних виробників. Оскільки основні переваги архітектури клієнт-сервер пов’язані з модульністю та можливістю комбінувати різні платформи та додатки, необхідно вирішити проблему співпраці між цими платформами та програмами.

Для того, щоб максимізувати ефективність архітектури клієнт-сервер, розробники повинні мати набір інструментів, які повинні забезпечувати однакові інструменти та забезпечувати єдиний доступ до системних ресурсів на всіх платформах. Це дозволить програмістам створювати програми, які не тільки однаково виглядають на різних ПК та робочих станціях, але й можуть отримати однаковий доступ до даних, незалежно від їх розташування.

Найпоширенішим способом задоволення цих вимог є використання стандартних інтерфейсів та протоколів між програмами, з одного боку, та програмним забезпеченням зв’язку та операційними системами, з іншого. Ці стандартизовані інтерфейси та протоколи називаються проміжним програмним забезпеченням. Існування стандартних програмних інтерфейсів допомагає розгортати один і той же додаток на різних типах серверів і робочих станцій. Очевидно, що це перевага не лише для споживачів, а й для виробників. Це тому, що споживачі купують додатки, а не сервери. Споживачі вибирають лише серверні продукти, на яких запущені необхідні програми. Взаємодія між різними серверними інтерфейсами та клієнтами вимагає стандартизованих протоколів.

Існують різні проміжні пакети - від дуже простих до дуже складних. Все це об’єднує здатність приховувати від користувачів складність та непослідовність різних мережевих протоколів та операційних систем. Виробники клієнтів і серверів зазвичай пропонують багато варіантів популярних програмних пакетів проміжного програмного забезпечення. Таким чином, користувачі можуть вибрати конкретну стратегію для розгортання проміжного програмного забезпечення, а потім придбати обладнання у різних виробників, які підтримують цю стратегію.

Архітектура проміжного програмного забезпечення

На малюнку 3.1.2 демонструється можливе використання проміжного програмного забезпечення в архітектурі клієнт-сервер. Точна роль проміжного програмного забезпечення залежить від обчислювальної моделі клієнт-сервер. Існує кілька типів програм клієнт-сервер, які залежать від розподілу функцій програми. У всякому разі, рис. 3.1.2 дає добре розуміння цієї архітектури.

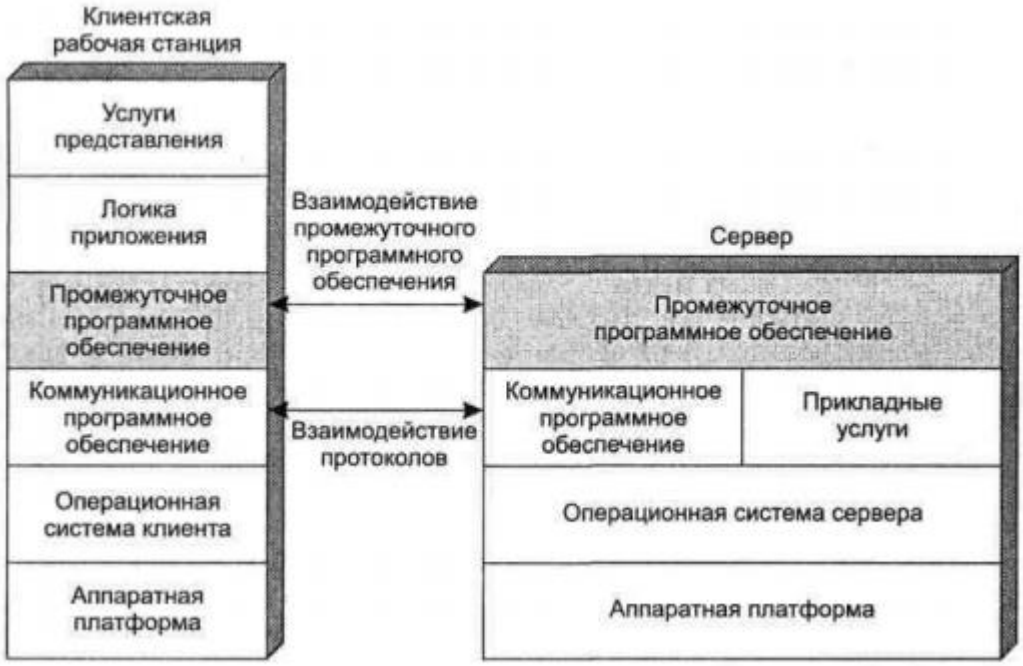

Рис. 3.1.2 – Роль проміжного програмного забезпечення в архітектурі клієнт-сервер

Зверніть увагу, що проміжне програмне забезпечення складається з двох компонентів - клієнта та сервера. Основною метою проміжного програмного забезпечення є надання доступу програмам або користувачам для доступу до різних служб, що надаються на сервері, так що не потрібно турбуватися про різницю між серверами. Стандартним методом доступу до реляційних баз даних для локальних та віддалених користувачів або додатків є мова структурованих запитів (SQL). Однак багато постачальників реляційних баз даних додали власні розширення, підтримуючи мову SQL. З одного боку, це дозволяє виробникам відрізняти свою продукцію, але з іншого боку, це призводить до несумісності.

Наприклад, розглянемо розподілену систему, яка обслуговує відділ кадрів. Основна інформація про працівників (наприклад, ім'я, адреса тощо) може зберігатися в базі даних Gupta, а інформація про заробітну плату - у базі даних Oracle. Коли користувач відділу кадрів вимагає доступу до певних записів, він не повинен розглядати, в якій базі даних зберігаються потрібні

йому записи та хто є виробником бази даних. Проміжне програмне забезпечення має забезпечувати рівний доступ до цих систем.

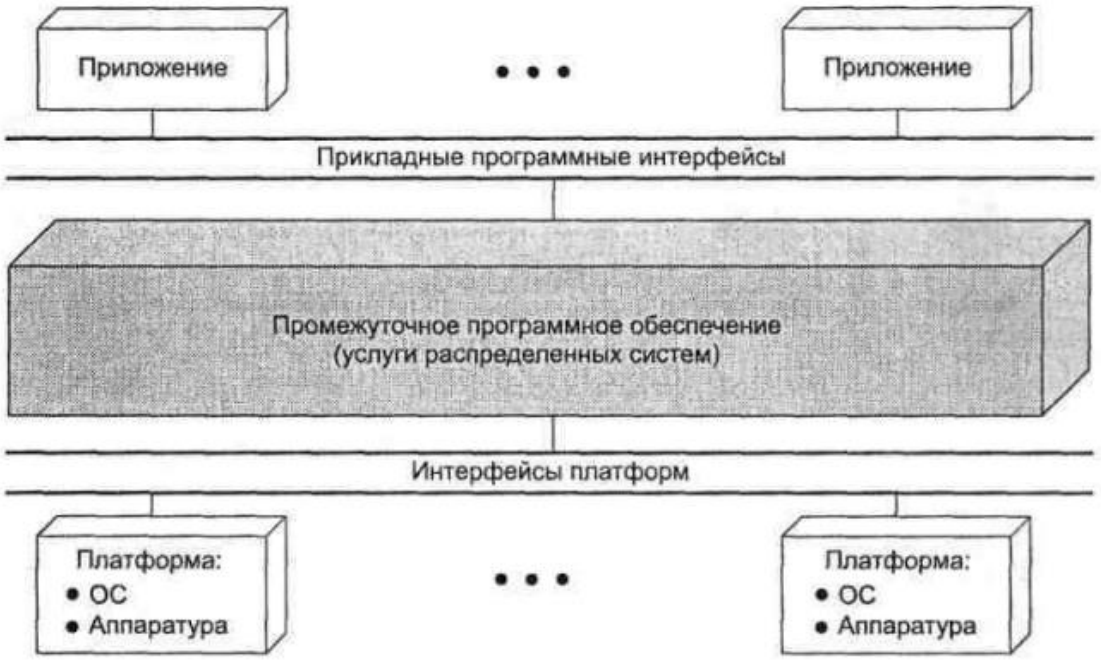

Рис. 3.1.3 - Проміжне програмне забезпечення з точки зору логіки

Корисно розглядати роль проміжного програмного забезпечення з точки зору логіки, а не реалізації (рис. 3.1.3). Вся розподілена система може розглядатися, як сукупність програм та ресурсів, доступних для користувачів. Користувачам не потрібно турбуватися про розташування даних або додатків. Усі програми працюють на уніфікованому інтерфейсі програмування програм (API). Проміжне програмне забезпечення, яке існує на всіх клієнтських та серверних платформах, відповідає за маршрутизацію запитів клієнтів на відповідний сервер.

Використовуючи проміжне програмне забезпечення для інтеграції несумісних продуктів - у цьому випадку проміжне програмне забезпечення використовується для усунення несумісності мережі та операційної системи. Мережі DECnet, Novell та TCP / IP підключені через магістральну мережу. Проміжне програмне забезпечення, що працює в кожній мережі, гарантує, що всі користувачі мережі мають прозорий доступ до програм та ресурсів у всіх трьох мережах.

Хоча на ринку є різні проміжні програми, усі ці продукти, як правило, базуються на одному з наступних трьох механізмів: обмін повідомленнями, виклики віддалених процедур та об'єктно-орієнтовані механізми.

Передача повідомлень

Розподілена передача повідомлень, яка реалізує функцію архітектури клієнт-сервер, показана на рис. 3.1.4. Клієнтський процес запитує послуги (наприклад, читання або друк файлу на принтері). Для цього він надсилає повідомлення із запитом на цю послугу процесу сервера. Процес сервера отримує повідомлення, обробляє запит, а потім відповідає на повідомлення. У найпростішому випадку потрібні лише дві функції: надсилання та отримання. Функція надсилання на вході повинна вказувати одержувача та вміст повідомлення. Функція прийому повинна вказати, від кого отримувати повідомлення (включаючи опцію "всі"), а також адресу буфера, в який потрібно помістити отримане повідомлення.

Рис. 3.1.4 – Механізм реалізації розподіленої передачі повідомлень

На малюнку. 3.1.5 показана можлива реалізація механізму передачі повідомлення. Процес використовує послуги модуля обміну повідомленнями. Запити на ці послуги можуть мати форму примітивів та параметрів. Примітиви відповідають виконуваним функціям, а параметри використовуються для передачі даних та управління інформацією. Конкретна форма примітиву залежить від програм обміну повідомленнями. Це може бути виклик процедури або повідомлення про процедуру операційної системи.

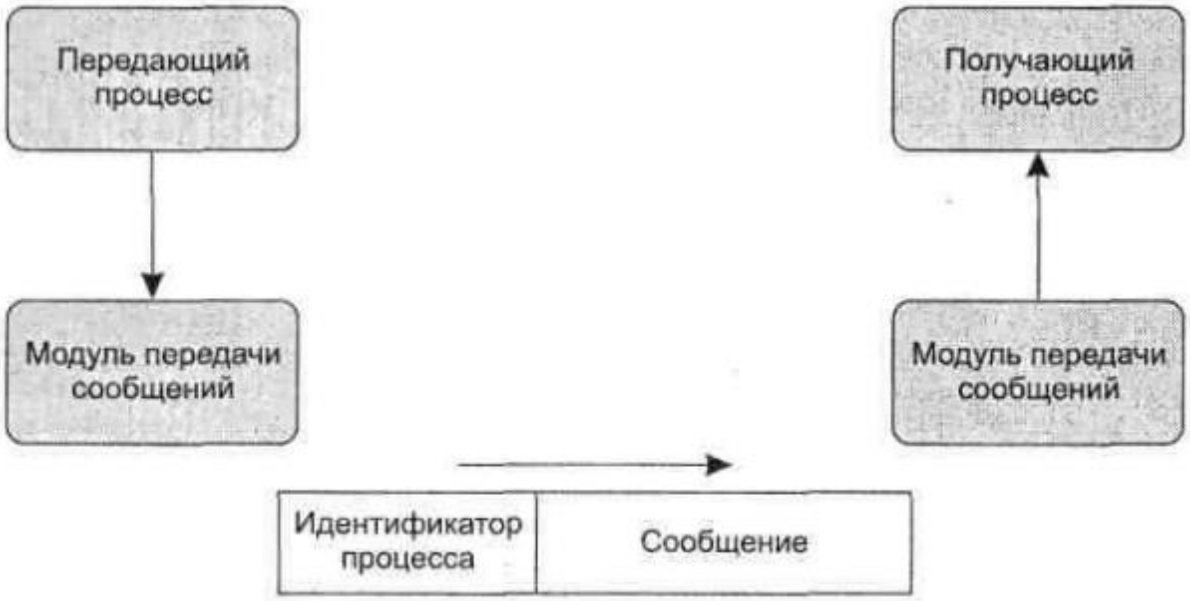

Рис. 3.1.5 – Базові примітиви передачі повідомлень

Процес надсилання повідомлення використовує примітив "Send". Вхідними параметрами цього примітиву є ідентифікатор процесу отримання та вміст повідомлення. Модуль обміну повідомленнями формує блок даних, що містить ці два елементи. Блок даних надсилається на комп'ютер, на якому запущений процес отримання, за допомогою будь-якого мережевого протоколу (наприклад, TCP / IP). Коли одержувач отримує блок даних, блок даних буде надісланий модулю обміну повідомленнями. Цей модуль перевіряє поле ідентифікатора процесу і зберігає повідомлення в буфері цього процесу.

У цьому випадку процес прийому повинен оголосити, що він хоче отримувати повідомлення, виділити буфер повідомлень та повідомити модуль повідомлень за допомогою примітива "Receive". Альтернативний метод не вимагає такого оголошення. Коли модуль повідомлення тільки отримує повідомлення, він сигналізує про процес, до якого слід звернутися, а потім поміщає повідомлення в спільний буфер.

Надійна передача повідомлень

Надійна система доставки повідомлень - це система, яка гарантує доставку повідомлень (якщо це можливо). Такі системи використовують надійні протоколи передачі або подібну логіку і повинні підтримувати перевірку помилок, цитування, повторну передачу та отримання повідомлень. Оскільки доставка гарантована, немає необхідності повідомляти процес передачі доставки повідомлення. Однак може бути корисним повернути підтвердження в процес передачі, щоб повідомити, що повідомлення доставлено. У будь-якому випадку, якщо система не може доставити повідомлення (наприклад, через збій мережі або збій одержувача), вона повідомить, що процес передачі не вдався. Інша крайність - це система обміну повідомленнями, яка надсилає повідомлення лише до комп'ютерної мережі, не повідомляючи про успіх чи невдачу. Ця альтернатива значно спрощує поштову систему та накладні витрати на обробку та взаємодію. Якщо програма вимагає підтвердження доставки повідомлень, її можна реалізувати на рівні програми.

Синхронні та асинхронні системи передачі повідомлень

При доступі до асинхронних (також відомих як неблокувальні) примітивів процес не зупиняється. Отже, після того, як процес викликає примітив Send, операційна система повертається до процесу управління відразу після того, як повідомлення поставлено в чергу для передачі або після створення копії повідомлення. Коли повідомлення буде відправлено або скопійовано в безпечне місце для подальшої передачі, процес передачі буде перерваний, і буфер повідомлення можна буде використовувати знову. Якщо копія повідомлення не створюється, будь-які зміни, внесені в повідомлення в процесі передачі після виклику примітиву Send, але перед відправленням повідомлення, є ризикованими. Подібним чином, після доступу до асинхронного примітиву Receive, процес продовжується. Коли надійде повідомлення, процес буде повідомлений про подію шляхом переривання або періодичного опитування.

Асинхронні примітиви забезпечують ефективний та гнучкий доступ процесу до системи обміну повідомленнями. Недоліком цього методу є те, що програми, що використовують такі примітиви, важко перевірити та налагодити.

Події, пов'язані з часом, які неможливо відтворити, можуть бути джерелом складних проблем.

Альтернативою є використання синхронізації або так званих примітивів блокування. Коли ви викликаєте синхронний примітив надсилання, керування не повертається до процесу передачі, поки повідомлення не доставлено (ненадійна послуга) або поки не отримає підтвердження, що повідомлення доставлено (надійна служба). Примітив блокування прийому не повертає керування, поки повідомлення не потрапить у призначений йому буфер.

Об'єктно-орієнтовані механізми

Після того, як в системному програмуванні домінували об’єктно-орієнтовані технології, розробники програм клієнт-сервер також почали застосовувати цей метод. У цьому випадку клієнт та об'єкти на сервері обмінюються повідомленнями. Взаємодія між об’єктами може базуватися на передачі повідомлень, віддалених викликах процедур або безпосередньо на основі об’єктно-орієнтованих функцій операційної системи.

Клієнти, яким потрібна ця послуга, надсилають запити до проксі-сервера об’єктного запиту, який діє, як каталог усіх віддалених служб, доступних у мережі (див. рис. 3.1.6). Брокер викликає відповідний об'єкт і передає необхідні дані. Потім віддалений об’єкт обробляє запит і відповідає проксі, а проксі повертає відповідь клієнту.

Успіх об’єктно-орієнтованого підходу залежить від стандартизації об’єктного механізму. На жаль, існує кілька конкуруючих схем, що співіснують у цій галузі. Однією з них є компонентна об'єктна модель Microsoft (COM), яка є основою OLE (Object Linking and Embedding). Digital Equipment Corporation підтримує цей метод, який розробив механізм COM для операційних систем UNIX. Технологія CORBA (Common Object Request Architecture), розроблена Object Management Group, конкурує з цим загальногалузевим підходом. IBM, Apple, Sun та багато інших компаній підтримують архітектуру CORBA.

Рис. 3.1.6 - Механізм реалізації брокера об’єктних запитів

3.2 Опис функціональності системи

Програмне забезпечення, що використовується для організації та проведення тестів та опитувань у процесі навчання, містить трьох основних учасників-користувачів системи;

На рисунку 3.2.1 показана попередня схема, що описує функції та дії учасників системи.

Рис. 3.2.1 – Діаграма прецедентів системи

Відповідно до плану користувачі поділяються на три типи: студенти, викладачі та адміністратори. Вчителі створюють тести та опитування, студенти також повинні складати іспити, адміністратори реєструють викладачів та учнів у системі та можуть видалити застарілу інформацію. Усі три типи користувачів можуть переглядати результати тестування та опитування.

3.3 Опис бази даних

Весь зміст системи та її основні дані (тобто дані про тести та опитування) зберігаються в базі даних.

База даних складається з наступних таблиць: “simple_test”, “average_test”, “difficult_test”. На рисунку 3.3.1 представлені основні таблиці, тип та кодування цих таблиць.

	Таблица	Действие	Строки	Тип	Сравнение	Размер	Фрагментировано
☐	average_test		10	InnoDB	utf8_general_ci	16 КиБ	-
☐	difficult_test		10	InnoDB	utf8_general_ci	16 КиБ	-
☐	simple_test		10	InnoDB	utf8_general_ci	16 КиБ	-
	3 таблицы	Всего	30	InnoDB	utf8_general_ci	48 КиБ	0 Байт

Рис.3.3.1 – Основні таблиці бази даних ресурсу

Структура цих трьох таблиць така (таблиця 3.3.1):

Таблиця 3.3.1 – Структура таблиць

Ім’я поля	**Тип і розмір поля**	**Опис поля**
id	int(11)	Первинний ключ
question_blocks	text	Текст питання
filled_blocks	text	Текст з правильною відповіддю
date	timestamp	Дата створення
answer	text	Правильна відповідь
input	text	Поле для вводу відповіді

Таблиця складається з таких полів як Первинний ключ, Текст питання, Текст з правильною відповіддю, Дата створення питання, Правильна відповідь та Поле для вводу відповіді. Всі три таблиці за структурою однакові змінюється

лише зміст в таблицях, зважаючи на складність тесту. Структура таблиці в phpMyAdmin показана на рисунку 3.3.2:

	#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
☐	1	id	int(11)			Нет	*Нет*		AUTO_INCREMENT	✎ ✕ ▾
☐	2	question_blocks	text	utf8_general_ci		Нет	*Нет*			✎ ✕ ▾
☐	3	filled_blocks	text	utf8_general_ci		Нет	*Нет*			✎ ✕ ▾
☐	4	date	timestamp			Нет	CURRENT_TIMESTAMP			✎ ✕ ▾
☐	5	answer	text	utf8_general_ci		Нет	*Нет*			✎ ✕ ▾
☐	6	input	text	utf8_general_ci		Нет	*Нет*			✎ ✕ ▾

Рис. 3.3.2 – Структура таблиці в phpMyAdmin

Зміст в таблицях змінюється залежно від вибраної складності тесту. Тобто якщо студент відкриває базовий тест, то і питання там будуть відповідні. Зміст таблиці “simple_test”(Базовий) представлено на рисунку 3.3.3:

id	question_blocks	filled_blocks	date	answer	input

1	<pre>
	public static _____ getProducts()
	{...
		<pre>
		public static function getProducts()...
			2020-10-08 21:32:28	function	<input type="text" id="answer" name="answer1" val...

2	<pre>
	public function actionIndex()
	{
	...
		<pre>
		public function actionIndex()
		{
		...
			2020-11-05 22:10:06	self	<input type="text" id="answer" name="answer2" val...

3	<pre>
	public function actionIndex()
	{
	...
		<pre>
		public function actionIndex()
		{
		...
			2020-11-05 22:33:27	Product	<input type="text" id="answer" name="answer3" val...

4	<pre>
	public function actionAbout()
	{
	...
		<pre>
		public function actionAbout()
		{
		...
			2020-11-05 23:10:20	require_once	<input type="text" id="answer" name="answer4" val...

5	<pre>
	public function actionAdd($id)
	{
	...
		<pre>
		public function actionAdd($id)
		{
		...
			2020-11-05 23:29:55	$_SERVER	<input type="text" id="answer" name="answer5" val...

6	<pre>
	public function actionAddAjax($id)
	{<b...
		<pre>
		public function actionAddAjax($id)
		{<b...
			2020-11-05 23:29:55	return	<input type="text" id="answer" name="answer6" val...

7	<pre>
	public function actionDelete($id)
	{
	...
		<pre>
		public function actionDelete($id)
		{
		...
			2020-11-05 23:39:50	header	<input type="text" id="answer" name="answer7" val...

8	<pre>
	public function actionIndex()
	{
	...
		<pre>
		public function actionIndex()
		{
		...
			2020-11-05 23:39:50	if	<input type="text" id="answer" name="answer8" val...

9	<pre>
	public function actionIndex()
	{
	...
		<pre>
		public function actionIndex()
		{
		...
			2020-11-05 23:48:20	ROOT	<input type="text" id="answer" name="answer9" val...

10	<pre>
	public function actionCategory($categoryId,...
		<pre>
		public function actionCategory($categoryId,...
			2020-11-05 23:48:20	$page	<input type="text" id="answer" name="answer10" va...

Рис. 3.3.3 – Зміст таблиці “simple_test”

Зміст таблиці “average_test”(Оптимальний) показано на рисунку 3.3.4:

id	question_blocks	filled_blocks	date	answer	input

1	<pre>
	public static function addProduct($id)
	
	...
		<pre>
		public static function addProduct($id)
		
		...
			2020-11-17 20:56:02	$_SESSION['products']	<input type="text" id="answer" name="answer1" val...

2	<pre>
	public static function countItems()
	
	{<...
		<pre>
		public static function countItems()
		
		{<...
			2020-11-17 21:22:46	return 0;	<input type="text" id="answer" name="answer2" val...

3	<pre>
	public static function getTotalPrice($produ...
		<pre>
		public static function getTotalPrice($produ...
			2020-11-17 21:31:49	if ($productsInCart)	<input type="text" id="answer" name="answer3" val...

4	<pre>
	public static function clear()
	
	{
	
	...
		<pre>
		public static function clear()
		
		{
		
		...
			2020-11-17 21:40:00	unset($_SESSION['products']);	<input type="text" id="answer" name="answer4" val...

5	<pre>
	public static function deleteProduct($id)<b...
		<pre>
		public static function deleteProduct($id)<b...
			2020-11-17 21:45:19	$productsInCart[$id]	<input type="text" id="answer" name="answer5" val...

6	<pre>
	public static function getStatusText($statu...
		<pre>
		public static function getStatusText($statu...
			2020-11-17 21:50:42	switch ($status)	<input type="text" id="answer" name="answer6" val...

7	<pre>
	public static function getImage($id)
	
	{...
		<pre>
		public static function getImage($id)
		
		{...
			2020-11-17 21:56:57	$path . $id . '.jpg'	<input type="text" id="answer" name="answer7" val...

8	<pre>
	public static function checkLogged()
	
	{...
		<pre>
		public static function checkLogged()
		
		{...
			2020-11-17 22:03:05	isset($_SESSION['user'])	<input type="text" id="answer" name="answer8" val...

9	<pre>
	public function actionCreate()
	
	{
	
	...
		<pre>
		public function actionCreate()
		
		{
		
		...
			2020-11-17 22:11:31	$name, $sortOrder, $status	<input type="text" id="answer" name="answer9" val...

10	<pre>
	class AdminController _____ AdminBase
	...
		<pre>
		class AdminController extends AdminB...
			2020-11-17 22:18:52	extends	<input type="text" id="answer" name="answer10" va...

Рис. 3.3.4 – Зміст таблиці “average_test”

Зміст таблиці “difficult_test”(Складний) показано на рисунку 3.3.5:

id	question_blocks	filled_blocks	date	answer	input

1	<pre>
	public static function createCategory($name...
		<pre>
		public static function createCategory($name...
			2020-11-18 22:55:52	$result->bindParam	<input type="text" id="answer" name="answer1" val...

2	<pre>
	public static function getCategoryById($id)...
		<pre>
		public static function getCategoryById($id)...
			2020-11-18 23:08:35	Db::getConnection()	<input type="text" id="answer" name="answer2" val...

3	<pre>
	public static function updateCategoryById($...
		<pre>
		public static function updateCategoryById($...
			2020-11-18 23:12:31	PDO::PARAM_INT	<input type="text" id="answer" name="answer3" val...

4	<pre>
	public static function deleteCategoryById($...
		<pre>
		public static function deleteCategoryById($...
			2020-11-18 23:18:30	$result->execute()	<input type="text" id="answer" name="answer4" val...

5	<pre>
	public static function getCategoriesListAdm...
		<pre>
		public static function getCategoriesListAdm...
			2020-11-18 23:23:45	$db->query	<input type="text" id="answer" name="answer5" val...

6	<pre>
	public static function getOrderById($id)<br...
		<pre>
		public static function getOrderById($id)<br...
			2020-11-18 23:29:58	PDO::FETCH_ASSOC	<input type="text" id="answer" name="answer6" val...

7	<pre>
	public static function deleteOrderById($id)...
		<pre>
		public static function deleteOrderById($id)...
			2020-11-18 23:33:13	DELETE FROM	<input type="text" id="answer" name="answer7" val...

8	<pre>
	public static function deleteProductById($i...
		<pre>
		public static function deleteProductById($i...
			2020-11-18 23:38:17	$sql	<input type="text" id="answer" name="answer8" val...

9	<pre>
	public static function getProductsList()<br...
		<pre>
		public static function getProductsList()<br...
			2020-11-18 23:43:04	$row['code']	<input type="text" id="answer" name="answer9" val...

10	<pre>
	public static function register($name, $ema...
		<pre>
		public static function register($name, $ema...
			2020-11-18 23:47:44	INSERT INTO	<input type="text" id="answer" name="answer10" va...

Рис. 3.3.5 – Зміст таблиці “difficult_test”

Маємо три різні складності для проходження тестів, “simple”, тобто базовий рівень складності, “avarage”, тобто оптимальний та “difficult”, тобто складний. Кожна із складностей включає в себе знання синтаксису мови php, операторів, управляючих структур, функцій, ООП, класів, методів, запитів до бази даних та структурованої мови SQL.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ

4.1 Загальний метод впровадження тестової системи

Вирішення завдання визначається в рамках дипломної роботи, а метою є створення системи тестування, призначеної для перевірки студентів та оцінки рівня їх знань після курсу.

Які вимоги до тестової системи можна призначити:

- Інтуїтивний графічний інтерфейс, але не дуже важкий;
- Простий у використанні;
- Можливість віддаленого використання.

4.2 Призначення системи

Проблема нестачі коштів для автоматичного виставлення оцінок та отримання оцінок зачіпає переважно студентів та вчителів. Результатом цієї проблеми є складність процесу, але створення інтерактивного тренажеру може повністю його видалити і виконати процес з більш високою якістю та меншим часом.

Система іспитів повинна забезпечити користувачам ефективну взаємодію для спрощення викладання предметів та зменшення навантаження на викладачів при здачі іспитів студентами. Тестова система повинна мати можливість не тільки зберігати та отримувати поточну інформацію, але й зберігати попередню інформацію.

Метою тестування є автоматизація процесу тестування, узагальнення та надання викладачам необхідної інформації про результати тестування учнів.

4.3 Опис користувачів

Передбачається, що система тестування призначена для використання вчителями, які займаються інформаційними технологіями. Тестова система, впроваджена в цій роботі, була використана викладачами для тестування в курсі "Розробка інтернет-магазину мовою PHP".

Тестова система підтримує таких користувачів:

- "Студент" - перевіряючий системою;

- "Вчитель" – системний керуючий;
- "Адміністратор" – редагуючий системою.

4.4 Тестування користувачів

Після того, як користувач прибуде на сторінку, для нього буде сформовано окремий набір тестових запитань у псевдовипадковому порядку. Представлено три тести на вибір, різної складності, кожен тест відповідає своїй складності. Представлено на рисунку 4.4.1. Під час процесу тестування кожна операція користувача буде відстежуватися і зберігатися у внутрішніх змінних.

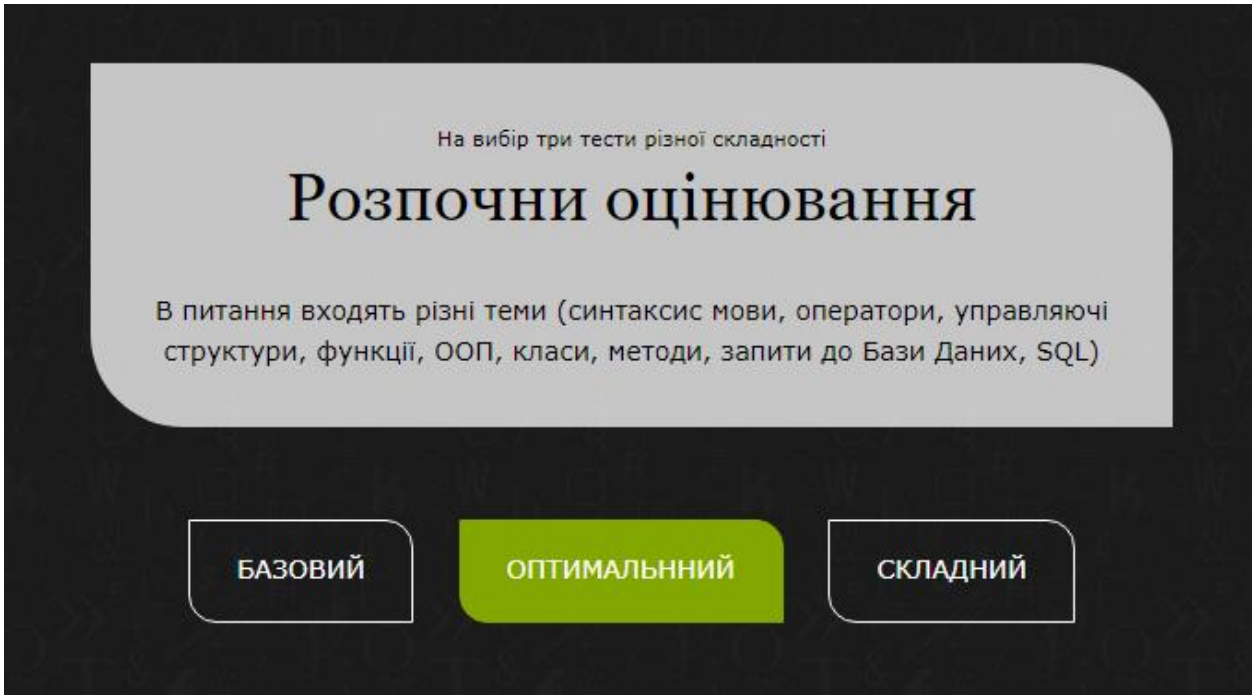

Рис. 4.4.1 – Основний блок вибору тестів

Тестова система складається з єдиного вікна, що відображає запитання та елементи користувача, такі як інтерфейс та інформаційна панель. Всі питання забезпечено наведеним прикладом коду з пропущеним відрізком, який і потрібно вписати, це дозволяє більш наочно передати сенс питання. Представлено на рисунку 4.4.2. Коли користувач входить і починає складати іспит, він має можливість переглядати панель навігації, що дозволяє:

- перейти на наступну сторінку;
- перейти на попередню сторінку;
- повернення на головну сторінку.

Протягом усього курсу була створена послідовна, логічна та функціональна навігаційна система.

Рис. 4.4.2 – Приклад блоків питань тесту

Блоки питань беруться із бази даних, за допомогою циклу записуються в масив, і в цьому ж циклі відразу віддаються на сторінку.

```
if ($count) {
    echo "<form action='result_simple_test.php' method='post'>";
    echo "    <ol class=''>";
    while ($row = mysqli_fetch_array($result)) {
        echo "    <div class='test'>";
        echo "        <li>{$row['question_blocks']}</li>";
        echo "        Відповідь:<br><br>";
        echo         $row['input'];
        echo "    </div>";
    }
    echo "    </ol>";
    echo "    <div class='center'><br>";
    echo "        <input class='btn' type='submit' value='Результат'><br><br>";
    echo "    </div>";
    echo "</form>";
```

}

Головним елементом інтерфейсу користувача є тестування, після проходження якого, тестуючому виводиться його результат в графічному вигляді. Наведено на рисунку 4.4.3:

Рис. 4.4.3 – Виведення результату

Після закінчення тестування студент може повернутися на головну сторінку за допомогою навігації, або переглянути свій індивідуальний зворотній зв’язок, як показано на рисунку 4.4.4:

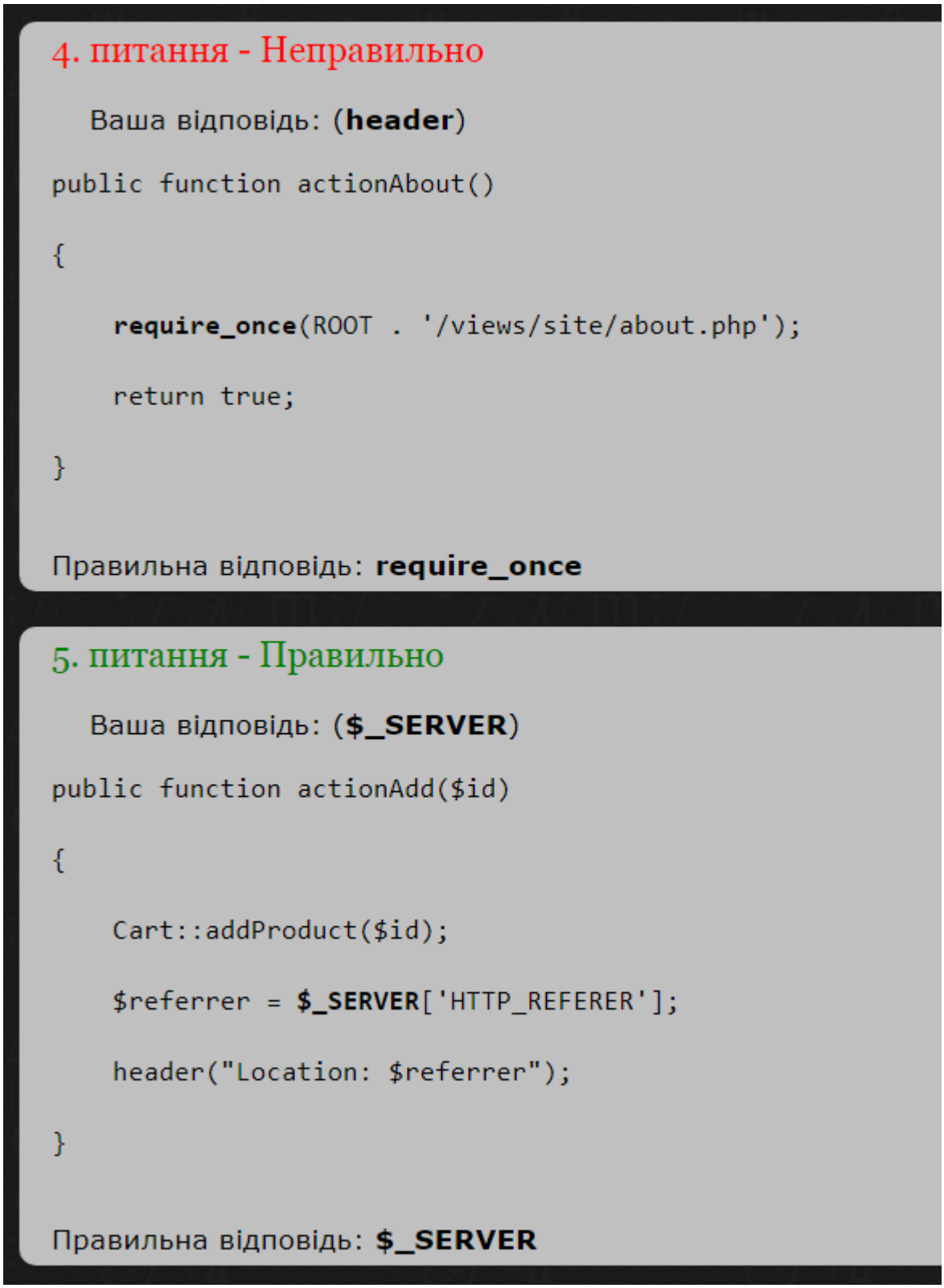

Рис. 4.4.4 – Наведення зворотнього зв’язку

Переглядаючи зворотній зв’язок студент має можливість зробити для себе відповідні висновки, де він зробив помилки і що саме йому треба повторити.

ВИСНОВКИ

Було проведене наукове дослідження та виявлено, що коли людина вивчає яку-небудь тему, то при використанні теоретичних навичок на практиці їй дуже складно зорієнтуватися. Тому було прийнято рішення створити спеціальний тренажер на якому можна було б тренувати свої практичні навички та закріплювати їх на створенні справжнього інтернет-магазину.

Загалом, був розроблений веб-сервіс, за допомогою якого студенти які навчались на умовному курсі “Розробка інтернет-магазину мовою PHP” можуть перевірити та закріпити свої знання. Тест охоплює знання та навички скриптової мови php. В запитаннях фігурують різні теми, синтаксис мови, оператори, управляючі структури, функції, ООП, класи, методи та запити до Бази Даних SQL.

Були використані такі технології, як html, css, php, а також фреймворк такий, як bootstrap.

За допомогою цього тренажеру користувачі зможуть навчатися розробляти свої інтернет магазини, що не тільки покращить їх конкурентоспроможність на ринку праці, але і допоможе окремим людям, які мають свій невеликий бізнес зробити свій онлайн магазин не звертаючись до спеціаліста.

СПИСОК ЛІТЕРАТУРИ

1. Дронов В.А. Разработка современных Web-сайтов. - СПб.: БХВПетербург, 2013. – 414 с.

2. Зольников Д.С. PHP 5. Как самостоятельно создать сайт любой сложности. 2 изд. - М.: НТ Пресс, 2014. - 272 с.

3. Кожемякин А. А. HTML и CSS в примерах. Создание Web-страниц - М.: Альтекс-А, 2014. - 416 с.

4. Федорчук Д. А. Разработка WEB приложений на PHP и MySQL - СПб. : Корона-принт, 2013. – 340 с.

5. Томсон Л., Веллинг Л. Разработка Web-приложений на PHP и MySql. ДиаСофтЮП, 2013. - 672с.

6. Веллинг Л., Томсон Л. Разработка веб-приложений с помощью PHP и MySQL. - М.: Вильямс - 2014. - 848 с.

7. Колисниченко Д.Н. Joomla! Руководство пользователя. - М.: Диалектика, 2013. - 256 с.

8. Яргер Р.Дж., Риз, Дж. Кинг. MySQL и mSQL: Базы данных для небольших предприятий и Интернета. - СПб: Символ-Плюс, 2013. -340 с.

9. Прохоренок Н. М. HTML, JavaScript, PHP и MySQL. Джентельменский набор Web-мастера, 2013. – 912 с.

10. Морозов Б. С. MySQL в связке с PHP. - СПб.: Корона-принт, 2014. – 310 с.

11. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL и JavaScript. - СПб.: Питер - 2011. - 496 с.

ДОДАТОК

Код Web-сайту

```
<!DOCTYPE html>
<html>
<head>
<title>Дипломна робота</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0,
maximum-scale=1.0, user-scalable=no">
<link href="arialogic/layout/styles/layout.css" rel="stylesheet"
type="text/css" media="all">
</head>
<body id="top">

<div class="bgded" style="background-
image:url('arialogic/images/demo/backgrounds/type.png');">
  <div class="wrapper overlay">
    <header id="header" class="hoc clear">
      <nav id="mainav" class="clear">
        <ul class="clear">
          <li class="active"><a href="index.html">Головна</a></li>
        </ul>
      </nav>
      <div id="logo">
        <h1><a href="index.html">Інтерактивний тренажер</a></h1>
        <p>для дистанційного курсу "Розробка інтернет-магазину мовою
PHP"</p>
      </div>
    </header>
  </div>

  <div id="pageintro" class="hoc clear">
    <article>
      <div class="introtxt">
        <p class="font-xs nospace">На вибір три тести різної
складності</p>
        <h2 class="heading">Розпочни оцінювання</h2>
```

```
        <p>В питання входять різні теми (синтаксис мови, оператори,
управляючі структури, функції, ООП, класи, методи,
          запити до Бази Даних, SQL)</p>
      </div>
      <footer>
        <ul class="nospace inline pushright">
          <li><a class="btn inverse"
href="/simple_test/index.php">Базовий</a></li>
          <li><a class="btn"
href="/average_test/index.php">Оптимальнний</a></li>
          <li><a class="btn inverse"
href="/difficult_test/index.php">Складний</a></li>
        </ul>
      </footer>
    </article>
  </div>
</div>

<div class="wrapper coloured">
  <div class="hoc clear">
    <figure id="logos">
      <ul class="nospace group">
        <li><a href="#"><img src="images/demo/logo.png" alt=""></a></li>
        <li><a href="#"><img src="images/demo/logo.png" alt=""></a></li>
        <li><a href="#"><img src="images/demo/logo.png" alt=""></a></li>
        <li><a href="#"><img src="images/demo/logo.png" alt=""></a></li>
        <li><a href="#"><img src="images/demo/logo.png" alt=""></a></li>
      </ul>
      <figcaption class="hidden"><a class="btn small" href="#">More
&raquo;</a></figcaption>
    </figure>
  </div>
</div>
<div class="wrapper row5" style="height: 50px">
  <div id="copyright" class="hoc clear">
    <p class="fl_left">Copyright &copy; 2016 - All Rights Reserved - <a
href="#">Domain Name</a></p>
```

```
    <p class="fl_right">Template by <a target="_blank"
href="http://www.os-templates.com/" title="Free Website Templates">OS
Templates</a></p>
  </div>
</div>

<a id="backtotop" href="#top"><i class="fa fa-chevron-up"></i></a>
<!-- JAVASCRIPTS -->
<script src="arialogic/layout/scripts/jquery.min.js"></script>
<script src="arialogic/layout/scripts/jquery.backtotop.js"></script>
<script src="arialogic/layout/scripts/jquery.mobilemenu.js"></script>
<!-- IE9 Placeholder Support -->
<script
src="arialogic/layout/scripts/jquery.placeholder.min.js"></script>
<!-- / IE9 Placeholder Support -->
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
    <title>Дипломна робота</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0, maximum-scale=1.0, user-scalable=no">
    <link href="/arialogic/layout/styles/layout.css" rel="stylesheet"
type="text/css" media="all">
    <link href="/arialogic/layout/styles/test.css" rel="stylesheet"
type="text/css"/>
</head>
<body id="top">
<!-- Top Background Image Wrapper -->
<div class="bgded" style="background-
image:url('/arialogic/images/demo/backgrounds/type.png');">

    <div class="wrapper overlay">
        <header id="header" class="hoc clear">
            <nav id="mainav" class="clear">
                <ul class="clear">
                    <li class="active"><a href="/index.html">На
головну</a></li>
                </ul>
            </nav>
            <div id="logo">
                <h1><a href="/index.html">Інтерактивний
тренажер</a></h1>
                <p>для дистанційного курсу "Розробка інтернет-магазину
мовою PHP"</p>
            </div>
        </header>
    </div>

    <?
    $connection = mysqli_connect("localhost", "root", "", "diploma");

    mysqli_set_charset($connection, "utf8");
```

```
    if (mysqli_connect_errno()) {
        echo "Failed to connect to MySQL: " . mysqli_connect_error();
    }

    $query = "SELECT * From `simple_test`";

    $result = mysqli_query($connection, $query);

    $count = mysqli_num_rows($result);

    if ($count) {
        echo "<form action='result_simple_test.php' method='post'>";
        echo "    <ol class=''>";
        while ($row = mysqli_fetch_array($result)) {
            echo "    <div class='test'>";
            echo "        <li>{$row['question_blocks']}</li>";
            echo "        Відповідь:<br><br>";
            echo          $row['input'];
            echo "    </div>";
        }
        echo "    </ol>";
        echo "    <div class='center'><br>";
        echo "        <input class='btn' type='submit'
value='Результат'><br><br>";
        echo "    </div>";
        echo "</form>";
    }
    ?>

    <div class="wrapper coloured">
        <div class="hoc clear">
            <figure id="logos">
                <ul class="nospace group">
                    <li><a href="#"><img src="images/demo/logo.png"
alt=""></a></li>
                    <li><a href="#"><img src="images/demo/logo.png"
alt=""></a></li>
                    <li><a href="#"><img src="images/demo/logo.png"
alt=""></a></li>
```

```
                    <li><a href="#"><img src="images/demo/logo.png"
alt=""></a></li>
                    <li><a href="#"><img src="images/demo/logo.png"
alt=""></a></li>
                </ul>
                <figcaption class="hidden"><a class="btn small"
href="#">More &raquo;</a></figcaption>
            </figure>
        </div>
    </div>

    <div class="wrapper row5">
        <div id="copyright" class="hoc clear">
            <p class="fl_left">Copyright &copy; 2016 - All Rights
Reserved - <a href="#">Domain Name</a></p>
            <p class="fl_right">Template by <a target="_blank"
href="http://www.os-templates.com/" title="Free Website Templates">OS
Templates</a></p>
        </div>
    </div>
    <a id="backtotop" href="#top"><i class="fa fa-chevron-up"></i></a>
    <!-- JAVASCRIPTS -->
    <script src="/arialogic/layout/scripts/jquery.min.js"></script>
    <script
src="/arialogic/layout/scripts/jquery.backtotop.js"></script>
    <script
src="/arialogic/layout/scripts/jquery.mobilemenu.js"></script>
    <!-- IE9 Placeholder Support -->
    <script
src="/arialogic/layout/scripts/jquery.placeholder.min.js"></script>
    <!-- / IE9 Placeholder Support -->
</body>
</html>
```

```
<?php
session_start();

if ($_POST) {
    for ($i = 1; $i <= 10; $i++) {
        $currentAnswers["answer$i"] = $_POST["answer$i"];
    }
}

$_SESSION['currentAnswers'] = $currentAnswers;
?>

<!DOCTYPE html>
<html>
<head>
    <title>Дипломная работа</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0, maximum-scale=1.0, user-scalable=no">
    <link href="/arialogic/layout/styles/layout.css" rel="stylesheet"
type="text/css" media="all">
    <link href="/arialogic/layout/styles/test.css" rel="stylesheet"
type="text/css"/>
</head>
<body id="top">

<div class="bgded" style="background-
image:url('/arialogic/images/demo/backgrounds/type.png');">
    <div class="wrapper overlay">
        <header id="header" class="hoc clear">
            <nav id="mainav" class="clear">
                <ul class="clear">
                    <li class="active"><a href="/index.html">На
головну</a></li>
                </ul>
            </nav>
            <div id="logo">
                <h1><a href="/index.html">Інтерактивний
тренажер</a></h1>
```

```
                <p>для дистанційного курсу "Розробка інтернет-магазину
мовою PHP"</p>
            </div>
        </header>
    </div>

    <?php
    $connection = mysqli_connect("localhost", "root", "", "diploma");

    mysqli_set_charset($connection, "utf8");

    if (mysqli_connect_errno()) {
        echo "Failed to connect to MySQL: " . mysqli_connect_error();
    }

    $query = "SELECT `question_blocks`, `answer` From `simple_test`";

    $result = mysqli_query($connection, $query);

    $count = mysqli_num_rows($result);

    if ($count) {
        $i = 0;
        while ($row = mysqli_fetch_array($result)) {
            $i++;
            $questions["question$i"] = $row['question_blocks'];
            $rightAnswers["answer$i"] = $row['answer'];
        }
    }
    ?>

    <?php
    if ($_POST) {
        $right = 0;
        $wrong = 0;

        for ($i = 1; $i <= count($currentAnswers); $i++) {
            if ($rightAnswers["answer$i"] ==
$currentAnswers["answer$i"]) {
```

```
                $right += 1;
            } else {
                $wrong -= 1;
            }
        }

        $rating = $right;

        $_SESSION['rating'] = $right;
    }
    ?>

    <div id="pageintro" class="hoc clear">
        <article>
            <div class="introtxt">
                <p class="nospace heading">Ваш результат:</p>
                <h2 class="heading"><?php echo $rating . 0; ?>
Балів</h2>
                <p>Ви можете переглянути свої відповіді та звірити їх з
правильними, перейшовши на наступну строрінку</p>
            </div>
            <footer>
                <ul class="nospace inline pushright">
<!--                    <li><a class="btn inverse"
href="/test/simple_test.php">Легко</a></li>-->
                    <li><a class="btn"
href="/simple_test/feedback_simple_test.php">Зворотній зв'язок</a></li>
<!--                    <li><a class="btn inverse" href="#">Правая
кнопка</a></li>-->
                </ul>
            </footer>
        </article>
    </div>
</div>

<div class="wrapper coloured">
        <div class="hoc clear">
            <figure id="logos">
                <ul class="nospace group">
```

```
                    <li><a href="#"><img src="images/demo/logo.png"
alt=""></a></li>
                    <li><a href="#"><img src="images/demo/logo.png"
alt=""></a></li>
                    <li><a href="#"><img src="images/demo/logo.png"
alt=""></a></li>
                    <li><a href="#"><img src="images/demo/logo.png"
alt=""></a></li>
                    <li><a href="#"><img src="images/demo/logo.png"
alt=""></a></li>
                </ul>
                <figcaption class="hidden"><a class="btn small"
href="#">More &raquo;</a></figcaption>
            </figure>
        </div>
    </div>
    <div class="wrapper row5">
        <div id="copyright" class="hoc clear">
            <p class="fl_left">Copyright &copy; 2016 - All Rights
Reserved - <a href="#">Domain Name</a></p>
            <p class="fl_right">Template by <a target="_blank"
href="http://www.os-templates.com/" title="Free Website Templates">OS
Templates</a></p>
        </div>
    </div>
    <a id="backtotop" href="#top"><i class="fa fa-chevron-up"></i></a>
    <!-- JAVASCRIPTS -->
    <script src="/arialogic/layout/scripts/jquery.min.js"></script>
    <script
src="/arialogic/layout/scripts/jquery.backtotop.js"></script>
    <script
src="/arialogic/layout/scripts/jquery.mobilemenu.js"></script>
    <!-- IE9 Placeholder Support -->
    <script
src="/arialogic/layout/scripts/jquery.placeholder.min.js"></script>
    <!-- / IE9 Placeholder Support -->
</body>
</html>
```

```
<?php
session_start();

if ($_SESSION) {
    for ($i = 1; $i <= 10; $i++) {
        $currentAnswers["answer$i"] =
$_SESSION["currentAnswers"]["answer$i"];
    }
}

$rating = $_SESSION['rating'];
?>

<!DOCTYPE html>
<html>
<head>
    <title>Дипломная работа</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0, maximum-scale=1.0, user-scalable=no">
    <link href="/arialogic/layout/styles/layout.css" rel="stylesheet"
type="text/css" media="all">
    <link href="/arialogic/layout/styles/test.css" rel="stylesheet"
type="text/css"/>
</head>
<body id="top">
<div class="bgded" style="background-
image:url('/arialogic/images/demo/backgrounds/type.png');">
    <div class="wrapper overlay">
        <header id="header" class="hoc clear">
            <nav id="mainav" class="clear">
                <ul class="clear">
                    <li class="active"><a href="/index.html">На
головну</a></li>
                </ul>
            </nav>
            <div id="logo">
                <h1><a href="/index.html">Інтерактивний
тренажер</a></h1>
```

```
                <p>для дистанційного курсу "Розробка інтернет-магазину
мовою PHP"</p>
            </div>
        </header>
    </div>

    <?php
    $connection = mysqli_connect("localhost", "root", "", "diploma");

    mysqli_set_charset($connection, "utf8");

    if (mysqli_connect_errno()) {
        echo "Failed to connect to MySQL: " . mysqli_connect_error();
    }

    $query = "SELECT `filled_blocks`, `answer` From `simple_test`";

    $result = mysqli_query($connection, $query);

    $count = mysqli_num_rows($result);

    if ($count) {
        $i = 0;
        while ($row = mysqli_fetch_array($result)) {
            $i++;
            $rightAnswers["answer$i"] = $row['answer'];
            $array[$i] = $row['filled_blocks'];
        }
    }
    ?>
    <div id="pageintro" class="hoc clear">
        <article>
            <div class="introtxt">
                <p class="nospace heading">Ваш результат:</p>
                <h2 class="heading"><?php echo $rating . 0; ?> / 100
Балів</h2>
            </div>
        </article>
    </div>
```

```
    <?php
    if ($_SESSION) {
        for ($i = 1; $i <= count($currentAnswers); $i++) {
            if ($rightAnswers["answer$i"] ==
$currentAnswers["answer$i"]) {
                echo "<div class='notation'>";
                echo "    <p class='green heading'>$i. питання -
Правильно</p> ";
                echo "    Ваша відповідь: (<b>" .
$currentAnswers["answer$i"] . "</b>)";
                echo $array[$i];

                echo "    Правильна відповідь:
<b>{$rightAnswers["answer$i"]}</b>";
                echo "</div><br>";
            } else {
                echo "<div class='notation'>";
                echo "    <p class='red heading'>$i. питання -
Неправильно</p> ";
                echo "    Ваша відповідь: (<b>" .
$currentAnswers["answer$i"] . "</b>)";
                echo $array[$i];

                echo "    Правильна відповідь:
<b>{$rightAnswers["answer$i"]}</b>";
                echo "</div><br>";
            }
        }
    }
    ?>
    <div class='center'><br>
        <a class='btn' href="/index.html">На головну</a><br><br>
    </div>
    <div class="wrapper coloured">
        <div class="hoc clear">
            <figure id="logos">
                <ul class="nospace group">
                    <li><a href="#"><img src="images/demo/logo.png"
alt=""></a></li>
```

```
                    <li><a href="#"><img src="images/demo/logo.png"
alt=""></a></li>
                    <li><a href="#"><img src="images/demo/logo.png"
alt=""></a></li>
                    <li><a href="#"><img src="images/demo/logo.png"
alt=""></a></li>
                    <li><a href="#"><img src="images/demo/logo.png"
alt=""></a></li>
                </ul>
                <figcaption class="hidden"><a class="btn small"
href="#">More &raquo;</a></figcaption>
            </figure>
        </div>
    </div>
    <div class="wrapper row5">
        <div id="copyright" class="hoc clear">
            <p class="fl_left">Copyright &copy; 2016 - All Rights
Reserved - <a href="#">Domain Name</a></p>
            <p class="fl_right">Template by <a target="_blank"
href="http://www.os-templates.com/" title="Free Website Templates">OS
Templates</a></p>
        </div>
    </div>
    <a id="backtotop" href="#top"><i class="fa fa-chevron-up"></i></a>
    <!-- JAVASCRIPTS -->
    <script src="/arialogic/layout/scripts/jquery.min.js"></script>
    <script
src="/arialogic/layout/scripts/jquery.backtotop.js"></script>
    <script
src="/arialogic/layout/scripts/jquery.mobilemenu.js"></script>
    <!-- IE9 Placeholder Support -->
    <script
src="/arialogic/layout/scripts/jquery.placeholder.min.js"></script>
    <!-- / IE9 Placeholder Support -->
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
    <title>Дипломна робота</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0, maximum-scale=1.0, user-scalable=no">
    <link href="/arialogic/layout/styles/layout.css" rel="stylesheet"
type="text/css" media="all">
    <link href="/arialogic/layout/styles/test.css" rel="stylesheet"
type="text/css"/>
</head>
<body id="top">
<!-- Top Background Image Wrapper -->
<div class="bgded" style="background-
image:url('/arialogic/images/demo/backgrounds/type.png');">
    <div class="wrapper overlay">
        <header id="header" class="hoc clear">
            <nav id="mainav" class="clear">
                <ul class="clear">
                    <li class="active"><a href="/index.html">На
головну</a></li>
                </ul>
            </nav>
            <div id="logo">
                <h1><a href="/index.html">Інтерактивний
тренажер</a></h1>
                <p>для дистанційного курсу "Розробка інтернет-магазину
мовою PHP"</p>
            </div>
        </header>
    </div>

    <?
    $connection = mysqli_connect("localhost", "root", "", "diploma");

    mysqli_set_charset($connection, "utf8");

    if (mysqli_connect_errno()) {
```

```
        echo "Failed to connect to MySQL: " . mysqli_connect_error();
    }

    $query = "SELECT * From `average_test`";

    $result = mysqli_query($connection, $query);

    $count = mysqli_num_rows($result);

    if ($count) {
        echo "<form action='result_average_test.php' method='post'>";
        echo "    <ol class=''>";
        while ($row = mysqli_fetch_array($result)) {
            echo "    <div class='test'>";
            echo "        <li>{$row['question_blocks']}</li>";
            echo "        Відповідь:<br><br>";
            echo          $row['input'];
            echo "    </div>";
        }
        echo "    </ol>";
        echo "    <div class='center'><br>";
        echo "        <input class='btn' type='submit'
value='Результат'><br><br>";
        echo "    </div>";
        echo "</form>";
    }
    ?>

    <div class="wrapper coloured">
        <div class="hoc clear">
            <figure id="logos">
                <ul class="nospace group">
                    <li><a href="#"><img src="images/demo/logo.png"
alt=""></a></li>
                    <li><a href="#"><img src="images/demo/logo.png"
alt=""></a></li>
                    <li><a href="#"><img src="images/demo/logo.png"
alt=""></a></li>
```

```
                    <li><a href="#"><img src="images/demo/logo.png"
alt=""></a></li>
                    <li><a href="#"><img src="images/demo/logo.png"
alt=""></a></li>
                </ul>
                <figcaption class="hidden"><a class="btn small"
href="#">More &raquo;</a></figcaption>
            </figure>
        </div>
    </div>

    <div class="wrapper row5">
        <div id="copyright" class="hoc clear">
            <p class="fl_left">Copyright &copy; 2016 - All Rights
Reserved - <a href="#">Domain Name</a></p>
            <p class="fl_right">Template by <a target="_blank"
href="http://www.os-templates.com/" title="Free Website Templates">OS
Templates</a></p>
        </div>
    </div>
    <a id="backtotop" href="#top"><i class="fa fa-chevron-up"></i></a>
    <!-- JAVASCRIPTS -->
    <script src="/arialogic/layout/scripts/jquery.min.js"></script>
    <script
src="/arialogic/layout/scripts/jquery.backtotop.js"></script>
    <script
src="/arialogic/layout/scripts/jquery.mobilemenu.js"></script>
    <!-- IE9 Placeholder Support -->
    <script
src="/arialogic/layout/scripts/jquery.placeholder.min.js"></script>
    <!-- / IE9 Placeholder Support -->
</body>
</html>
```

```
<?php
session_start();

if ($_POST) {
    for ($i = 1; $i <= 10; $i++) {
        $currentAnswers["answer$i"] = $_POST["answer$i"];
    }
}

$_SESSION['currentAnswers'] = $currentAnswers;
?>

<!DOCTYPE html>
<html>
<head>
    <title>Дипломная работа</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0, maximum-scale=1.0, user-scalable=no">
    <link href="/arialogic/layout/styles/layout.css" rel="stylesheet"
type="text/css" media="all">
    <link href="/arialogic/layout/styles/test.css" rel="stylesheet"
type="text/css"/>
</head>
<body id="top">

<div class="bgded" style="background-
image:url('/arialogic/images/demo/backgrounds/type.png');">
    <div class="wrapper overlay">
        <header id="header" class="hoc clear">
            <nav id="mainav" class="clear">
                <ul class="clear">
                    <li class="active"><a href="/index.html">На
головну</a></li>
                </ul>
            </nav>
            <div id="logo">
                <h1><a href="/index.html">Інтерактивний
тренажер</a></h1>
```

```
                <p>для дистанційного курсу "Розробка інтернет-магазину
мовою PHP"</p>
            </div>
        </header>
    </div>

    <?php
    $connection = mysqli_connect("localhost", "root", "", "diploma");

    mysqli_set_charset($connection, "utf8");

    if (mysqli_connect_errno()) {
        echo "Failed to connect to MySQL: " . mysqli_connect_error();
    }

    $query = "SELECT `question_blocks`, `answer` From `average_test`";

    $result = mysqli_query($connection, $query);

    $count = mysqli_num_rows($result);

    if ($count) {
        $i = 0;
        while ($row = mysqli_fetch_array($result)) {
            $i++;
            $questions["question$i"] = $row['question_blocks'];
            $rightAnswers["answer$i"] = $row['answer'];
        }
    }
    ?>

    <?php
    if ($_POST) {
        $right = 0;
        $wrong = 0;

        for ($i = 1; $i <= count($currentAnswers); $i++) {
            if ($rightAnswers["answer$i"] ==
$currentAnswers["answer$i"]) {
```

```
                $right += 1;
            } else {
                $wrong -= 1;
            }
        }

        $rating = $right;

        $_SESSION['rating'] = $right;
    }
    ?>

    <div id="pageintro" class="hoc clear">
        <article>
            <div class="introtxt">
                <p class="nospace heading">Ваш результат:</p>
                <h2 class="heading"><?php echo $rating . 0; ?>
Балів</h2>
                <p>Ви можете переглянути свої відповіді та звірити їх з
правильними, перейшовши на наступну строрінку</p>
            </div>
            <footer>
                <ul class="nospace inline pushright">
                    <!--                    <li><a class="btn inverse"
href="/test/simple_test.php">Легко</a></li>-->
                    <li><a class="btn"
href="/average_test/feedback_average_test.php">Зворотній
Зв'язок</a></li>
                    <!--                    <li><a class="btn inverse"
href="#">Правая кнопка</a></li>-->
                </ul>
            </footer>
        </article>
    </div>
</div>

<div class="wrapper coloured">
    <div class="hoc clear">
        <figure id="logos">
```

```
            <ul class="nospace group">
                <li><a href="#"><img src="images/demo/logo.png"
alt=""></a></li>
                <li><a href="#"><img src="images/demo/logo.png"
alt=""></a></li>
                <li><a href="#"><img src="images/demo/logo.png"
alt=""></a></li>
                <li><a href="#"><img src="images/demo/logo.png"
alt=""></a></li>
                <li><a href="#"><img src="images/demo/logo.png"
alt=""></a></li>
            </ul>
            <figcaption class="hidden"><a class="btn small"
href="#">More &raquo;</a></figcaption>
        </figure>
    </div>
</div>

<div class="wrapper row5">
    <div id="copyright" class="hoc clear">
        <p class="fl_left">Copyright &copy; 2016 - All Rights Reserved -
<a href="#">Domain Name</a></p>
        <p class="fl_right">Template by <a target="_blank"
href="http://www.os-templates.com/" title="Free Website Templates">OS
Templates</a></p>
    </div>
</div>
<a id="backtotop" href="#top"><i class="fa fa-chevron-up"></i></a>
<!-- JAVASCRIPTS -->
<script src="/arialogic/layout/scripts/jquery.min.js"></script>
<script src="/arialogic/layout/scripts/jquery.backtotop.js"></script>
<script src="/arialogic/layout/scripts/jquery.mobilemenu.js"></script>
<!-- IE9 Placeholder Support -->
<script
src="/arialogic/layout/scripts/jquery.placeholder.min.js"></script>
<!-- / IE9 Placeholder Support -->
</body>
</html>
```

```
<?php
session_start();

if ($_SESSION) {
    for ($i = 1; $i <= 10; $i++) {
        $currentAnswers["answer$i"] =
$_SESSION["currentAnswers"]["answer$i"];
    }
}

$rating = $_SESSION['rating'];
?>

<!DOCTYPE html>
<html>
<head>
    <title>Дипломная работа</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0, maximum-scale=1.0, user-scalable=no">
    <link href="/arialogic/layout/styles/layout.css" rel="stylesheet"
type="text/css" media="all">
    <link href="/arialogic/layout/styles/test.css" rel="stylesheet"
type="text/css"/>
</head>
<body id="top">


<div class="bgded" style="background-
image:url('/arialogic/images/demo/backgrounds/type.png');">
    <div class="wrapper overlay">
        <header id="header" class="hoc clear">
            <nav id="mainav" class="clear">
                <ul class="clear">
                    <li class="active"><a href="/index.html">На
головну</a></li>
                </ul>
            </nav>
            <div id="logo">
```

```
                <h1><a href="/index.html">Інтерактивний
тренажер</a></h1>
                <p>для дистанційного курсу "Розробка інтернет-магазину
мовою PHP"</p>
            </div>
        </header>
    </div>

    <?php
    $connection = mysqli_connect("localhost", "root", "", "diploma");

    mysqli_set_charset($connection, "utf8");

    if (mysqli_connect_errno()) {
        echo "Failed to connect to MySQL: " . mysqli_connect_error();
    }

    $query = "SELECT `filled_blocks`, `answer` From `average_test`";

    $result = mysqli_query($connection, $query);

    $count = mysqli_num_rows($result);

    if ($count) {
        $i = 0;
        while ($row = mysqli_fetch_array($result)) {
            $i++;
            $rightAnswers["answer$i"] = $row['answer'];
            $array[$i] = $row['filled_blocks'];
        }
    }
    ?>

    <div id="pageintro" class="hoc clear">
        <article>
            <div class="introtxt">
                <p class="nospace heading">Ваш результат:</p>
                <h2 class="heading"><?php echo $rating . 0; ?> / 100
Балів</h2>
```

```
            </div>
        </article>
    </div>

    <?php
    if ($_SESSION) {
        for ($i = 1; $i <= count($currentAnswers); $i++) {
            if ($rightAnswers["answer$i"] ==
$currentAnswers["answer$i"]) {
                echo "<div class='notation'>";
                echo "    <p class='green heading'>$i. питання -
Правильно</p> ";
                echo "    Ваша відповідь: (<b>" .
$currentAnswers["answer$i"] . "</b>)";
                echo $array[$i];
                echo "    Правильна відповідь:
<b>{$rightAnswers["answer$i"]}</b>";
                echo "</div><br>";
            } else {
                echo "<div class='notation'>";
                echo "    <p class='red heading'>$i. питання -
Неправильно</p> ";
                echo "    Ваша відповідь: (<b>" .
$currentAnswers["answer$i"] . "</b>)";
                echo $array[$i];
                echo "    Правильна відповідь:
<b>{$rightAnswers["answer$i"]}</b>";
                echo "</div><br>";
            }
        }
    }
    ?>
    <div class='center'><br>
        <a class='btn' href="/index.html">На головну</a><br><br>
    </div>
    <div class="wrapper coloured">
        <div class="hoc clear">
            <figure id="logos">
                <ul class="nospace group">
```

```
                    <li><a href="#"><img src="images/demo/logo.png"
alt=""></a></li>
                    <li><a href="#"><img src="images/demo/logo.png"
alt=""></a></li>
                    <li><a href="#"><img src="images/demo/logo.png"
alt=""></a></li>
                    <li><a href="#"><img src="images/demo/logo.png"
alt=""></a></li>
                    <li><a href="#"><img src="images/demo/logo.png"
alt=""></a></li>
                </ul>
                <figcaption class="hidden"><a class="btn small"
href="#">More &raquo;</a></figcaption>
            </figure>
        </div>
    </div>
    <div class="wrapper row5">
        <div id="copyright" class="hoc clear">
            <p class="fl_left">Copyright &copy; 2016 - All Rights
Reserved - <a href="#">Domain Name</a></p>
            <p class="fl_right">Template by <a target="_blank"
href="http://www.os-templates.com/" title="Free Website Templates">OS
Templates</a></p>
        </div>
    </div>
    <a id="backtotop" href="#top"><i class="fa fa-chevron-up"></i></a>
    <!-- JAVASCRIPTS -->
    <script src="/arialogic/layout/scripts/jquery.min.js"></script>
    <script
src="/arialogic/layout/scripts/jquery.backtotop.js"></script>
    <script
src="/arialogic/layout/scripts/jquery.mobilemenu.js"></script>
    <!-- IE9 Placeholder Support -->
    <script
src="/arialogic/layout/scripts/jquery.placeholder.min.js"></script>
    <!-- / IE9 Placeholder Support -->
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
    <title>Дипломна робота</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0, maximum-scale=1.0, user-scalable=no">
    <link href="/arialogic/layout/styles/layout.css" rel="stylesheet"
type="text/css" media="all">
    <link href="/arialogic/layout/styles/test.css" rel="stylesheet"
type="text/css"/>
</head>
<body id="top">
<!-- Top Background Image Wrapper -->
<div class="bgded" style="background-
image:url('/arialogic/images/demo/backgrounds/type.png');">
    <div class="wrapper overlay">
        <header id="header" class="hoc clear">
            <nav id="mainav" class="clear">
                <ul class="clear">
                    <li class="active"><a href="/index.html">На
головну</a></li>
                </ul>
            </nav>
            <div id="logo">
                <h1><a href="/index.html">Інтерактивний
тренажер</a></h1>
                <p>для дистанційного курсу "Розробка інтернет-магазину
мовою PHP"</p>
            </div>
        </header>
    </div>

    <?
    $connection = mysqli_connect("localhost", "root", "", "diploma");

    mysqli_set_charset($connection, "utf8");

    if (mysqli_connect_errno()) {
```

```
        echo "Failed to connect to MySQL: " . mysqli_connect_error();
    }

    $query = "SELECT * From `difficult_test`";

    $result = mysqli_query($connection, $query);

    $count = mysqli_num_rows($result);

    if ($count) {
        echo "<form action='result_difficult_test.php' method='post'>";
        echo "    <ol class=''>";
        while ($row = mysqli_fetch_array($result)) {
            echo "    <div class='test'>";
            echo "        <li>{$row['question_blocks']}</li>";
            echo "        Відповідь:<br><br>";
            echo          $row['input'];
            echo "    </div>";
        }
        echo "    </ol>";
        echo "    <div class='center'><br>";
        echo "        <input class='btn' type='submit'
value='Результат'><br><br>";
        echo "    </div>";
        echo "</form>";
    }
    ?>

    <div class="wrapper coloured">
        <div class="hoc clear">
            <figure id="logos">
                <ul class="nospace group">
                    <li><a href="#"><img src="images/demo/logo.png"
alt=""></a></li>
                    <li><a href="#"><img src="images/demo/logo.png"
alt=""></a></li>
                    <li><a href="#"><img src="images/demo/logo.png"
alt=""></a></li>
```

```
                    <li><a href="#"><img src="images/demo/logo.png"
alt=""></a></li>
                    <li><a href="#"><img src="images/demo/logo.png"
alt=""></a></li>
                </ul>
                <figcaption class="hidden"><a class="btn small"
href="#">More &raquo;</a></figcaption>
            </figure>
        </div>
    </div>

    <div class="wrapper row5">
        <div id="copyright" class="hoc clear">
            <p class="fl_left">Copyright &copy; 2016 - All Rights
Reserved - <a href="#">Domain Name</a></p>
            <p class="fl_right">Template by <a target="_blank"
href="http://www.os-templates.com/" title="Free Website Templates">OS
Templates</a></p>
        </div>
    </div>
    <a id="backtotop" href="#top"><i class="fa fa-chevron-up"></i></a>
    <!-- JAVASCRIPTS -->
    <script src="/arialogic/layout/scripts/jquery.min.js"></script>
    <script
src="/arialogic/layout/scripts/jquery.backtotop.js"></script>
    <script
src="/arialogic/layout/scripts/jquery.mobilemenu.js"></script>
    <!-- IE9 Placeholder Support -->
    <script
src="/arialogic/layout/scripts/jquery.placeholder.min.js"></script>
    <!-- / IE9 Placeholder Support -->
</body>
</html>
```

```
<?php
session_start();

if ($_POST) {
    for ($i = 1; $i <= 10; $i++) {
        $currentAnswers["answer$i"] = $_POST["answer$i"];
    }
}

$_SESSION['currentAnswers'] = $currentAnswers;
?>

<!DOCTYPE html>
<html>
<head>
    <title>Дипломная работа</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0, maximum-scale=1.0, user-scalable=no">
    <link href="/arialogic/layout/styles/layout.css" rel="stylesheet"
type="text/css" media="all">
    <link href="/arialogic/layout/styles/test.css" rel="stylesheet"
type="text/css"/>
</head>
<body id="top">

<div class="bgded" style="background-
image:url('/arialogic/images/demo/backgrounds/type.png');">
    <div class="wrapper overlay">
        <header id="header" class="hoc clear">
            <nav id="mainav" class="clear">
                <ul class="clear">
                    <li class="active"><a href="/index.html">На
головну</a></li>
                </ul>
            </nav>
            <div id="logo">
                <h1><a href="/index.html">Інтерактивний
тренажер</a></h1>
```

```
                <p>для дистанційного курсу "Розробка інтернет-магазину
мовою PHP"</p>
            </div>
        </header>
    </div>

    <?php
    $connection = mysqli_connect("localhost", "root", "", "diploma");

    mysqli_set_charset($connection, "utf8");

    if (mysqli_connect_errno()) {
        echo "Failed to connect to MySQL: " . mysqli_connect_error();
    }

    $query = "SELECT `question_blocks`, `answer` From `difficult_test`";

    $result = mysqli_query($connection, $query);

    $count = mysqli_num_rows($result);

    if ($count) {
        $i = 0;
        while ($row = mysqli_fetch_array($result)) {
            $i++;
            $questions["question$i"] = $row['question_blocks'];
            $rightAnswers["answer$i"] = $row['answer'];
        }
    }
    ?>

    <?php
    if ($_POST) {
        $right = 0;
        $wrong = 0;

        for ($i = 1; $i <= count($currentAnswers); $i++) {
            if ($rightAnswers["answer$i"] ==
$currentAnswers["answer$i"]) {
```

```
                $right += 1;
            } else {
                $wrong -= 1;
            }
        }

        $rating = $right;

        $_SESSION['rating'] = $right;
    }
    ?>

    <div id="pageintro" class="hoc clear">
        <article>
            <div class="introtxt">
                <p class="nospace heading">Ваш результат:</p>
                <h2 class="heading"><?php echo $rating . 0; ?>
Балів</h2>
                <p>Ви можете переглянути свої відповіді та звірити їх з
правильними, перейшовши на наступну строрінку</p>
            </div>
            <footer>
                <ul class="nospace inline pushright">
                    <!--                    <li><a class="btn inverse"
href="/test/simple_test.php">Легко</a></li>-->
                    <li><a class="btn"
href="/difficult_test/feedback_difficult_test.php">Зворотній
зв'язок</a></li>
                    <!--                    <li><a class="btn inverse"
href="#">Правая кнопка</a></li>-->
                </ul>
            </footer>
        </article>
    </div>
</div>

<div class="wrapper coloured">
    <div class="hoc clear">
        <figure id="logos">
```

```
            <ul class="nospace group">
                <li><a href="#"><img src="images/demo/logo.png"
alt=""></a></li>
                <li><a href="#"><img src="images/demo/logo.png"
alt=""></a></li>
                <li><a href="#"><img src="images/demo/logo.png"
alt=""></a></li>
                <li><a href="#"><img src="images/demo/logo.png"
alt=""></a></li>
                <li><a href="#"><img src="images/demo/logo.png"
alt=""></a></li>
            </ul>
            <figcaption class="hidden"><a class="btn small"
href="#">More &raquo;</a></figcaption>
        </figure>
    </div>
</div>
<div class="wrapper row5">
    <div id="copyright" class="hoc clear">
        <p class="fl_left">Copyright &copy; 2016 - All Rights Reserved -
<a href="#">Domain Name</a></p>
        <p class="fl_right">Template by <a target="_blank"
href="http://www.os-templates.com/" title="Free Website Templates">OS
Templates</a></p>
    </div>
</div>
<a id="backtotop" href="#top"><i class="fa fa-chevron-up"></i></a>
<!-- JAVASCRIPTS -->
<script src="/arialogic/layout/scripts/jquery.min.js"></script>
<script src="/arialogic/layout/scripts/jquery.backtotop.js"></script>
<script src="/arialogic/layout/scripts/jquery.mobilemenu.js"></script>
<!-- IE9 Placeholder Support -->
<script
src="/arialogic/layout/scripts/jquery.placeholder.min.js"></script>
<!-- / IE9 Placeholder Support -->
</body>
</html>
```

```
<?php
session_start();

if ($_SESSION) {
    for ($i = 1; $i <= 10; $i++) {
        $currentAnswers["answer$i"] =
$_SESSION["currentAnswers"]["answer$i"];
    }
}

$rating = $_SESSION['rating'];
?>

<!DOCTYPE html>
<html>
<head>
    <title>Дипломная работа</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0, maximum-scale=1.0, user-scalable=no">
    <link href="/arialogic/layout/styles/layout.css" rel="stylesheet"
type="text/css" media="all">
    <link href="/arialogic/layout/styles/test.css" rel="stylesheet"
type="text/css"/>
</head>
<body id="top">


<div class="bgded" style="background-
image:url('/arialogic/images/demo/backgrounds/type.png');">
    <div class="wrapper overlay">
        <header id="header" class="hoc clear">
            <nav id="mainav" class="clear">
                <ul class="clear">
                    <li class="active"><a href="/index.html">На
головну</a></li>
                </ul>
            </nav>
            <div id="logo">
```

```
                <h1><a href="/index.html">Інтерактивний
тренажер</a></h1>
                <p>для дистанційного курсу "Розробка інтернет-магазину
мовою PHP"</p>
            </div>
        </header>
    </div>

    <?php
    $connection = mysqli_connect("localhost", "root", "", "diploma");

    mysqli_set_charset($connection, "utf8");

    if (mysqli_connect_errno()) {
        echo "Failed to connect to MySQL: " . mysqli_connect_error();
    }

    $query = "SELECT `filled_blocks`, `answer` From `difficult_test`";

    $result = mysqli_query($connection, $query);

    $count = mysqli_num_rows($result);

    if ($count) {
        $i = 0;
        while ($row = mysqli_fetch_array($result)) {
            $i++;
            $rightAnswers["answer$i"] = $row['answer'];
            $array[$i] = $row['filled_blocks'];
        }
    }
    ?>

    <div id="pageintro" class="hoc clear">
        <article>
            <div class="introtxt">
                <p class="nospace heading">Ваш результат:</p>
                <h2 class="heading"><?php echo $rating . 0; ?> / 100
Балів</h2>
```

```
            </div>
        </article>
    </div>

    <?php
    if ($_SESSION) {
        for ($i = 1; $i <= count($currentAnswers); $i++) {
            if ($rightAnswers["answer$i"] ==
$currentAnswers["answer$i"]) {
                echo "<div class='notation'>";
                echo "    <p class='green heading'>$i. питання -
Правильно</p> ";
                echo "    Ваша відповідь: (<b>" .
$currentAnswers["answer$i"] . "</b>)";
                echo $array[$i];

                echo "    Правильна відповідь:
<b>{$rightAnswers["answer$i"]}</b>";
                echo "</div><br>";
            } else {
                echo "<div class='notation'>";
                echo "    <p class='red heading'>$i. питання -
Неправильно</p> ";
                echo "    Ваша відповідь: (<b>" .
$currentAnswers["answer$i"] . "</b>)";
                echo $array[$i];
                echo "    Правильна відповідь:
<b>{$rightAnswers["answer$i"]}</b>";
                echo "</div><br>";
            }
        }
    }
    ?>
    <div class='center'><br>
        <a class='btn' href="/index.html">На головну</a><br><br>
    </div>
    <div class="wrapper coloured">
        <div class="hoc clear">
            <figure id="logos">
```

```
                <ul class="nospace group">
                    <li><a href="#"><img src="images/demo/logo.png"
alt=""></a></li>
                    <li><a href="#"><img src="images/demo/logo.png"
alt=""></a></li>
                    <li><a href="#"><img src="images/demo/logo.png"
alt=""></a></li>
                    <li><a href="#"><img src="images/demo/logo.png"
alt=""></a></li>
                    <li><a href="#"><img src="images/demo/logo.png"
alt=""></a></li>
                </ul>
                <figcaption class="hidden"><a class="btn small"
href="#">More &raquo;</a></figcaption>
            </figure>
        </div>
    </div>
    <div class="wrapper row5">
        <div id="copyright" class="hoc clear">
            <p class="fl_left">Copyright &copy; 2016 - All Rights
Reserved - <a href="#">Domain Name</a></p>
            <p class="fl_right">Template by <a target="_blank"
href="http://www.os-templates.com/" title="Free Website Templates">OS
Templates</a></p>
        </div>
    </div>
    <a id="backtotop" href="#top"><i class="fa fa-chevron-up"></i></a>
    <!-- JAVASCRIPTS -->
    <script src="/arialogic/layout/scripts/jquery.min.js"></script>
    <script
src="/arialogic/layout/scripts/jquery.backtotop.js"></script>
    <script
src="/arialogic/layout/scripts/jquery.mobilemenu.js"></script>
    <!-- IE9 Placeholder Support -->
    <script
src="/arialogic/layout/scripts/jquery.placeholder.min.js"></script>
    <!-- / IE9 Placeholder Support -->
</body>
</html>
```

CSS media queries

```
.test {
    padding-left: 40px;
    padding-top: 5px;
    background-color: silver;
    color: black;
    line-height: 1;
    font-size: 13pt;
    margin: 20px auto 0 auto;
    width: 1015px;
    border-radius: 10px;
}

input {
    border-radius: 10px;
    padding-left: 5px;
    width: 300px;
}

.btn {
    width: 400px;
    margin: 0 auto 10px auto;
}

.alfa {
    padding-left: 10px;
    background-color: silver;
    color: black;
}

.result {
    background-color: silver;
    color: black;
    width: 500px;
    border-radius: 10px;
    height: 50px;
    font-size: 14pt;
    margin: 30px auto 20px auto;
```

```
    padding-top: 13px;
}

.notation {
    background-color: silver;
    color: black;
    font-size: 13pt;
    margin-left: 10px;
    margin-right: 50px;
    padding-left: 20px;
    padding-bottom: 5px;
    padding-top: 5px;
    width: 95%;
    border-radius: 10px;
}

.red {
    color: red;
}

.green {
    color: green;
}
```