

ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЕКТ

на тему:

«Розробка універсального засобу адміністрування баз даних через web-інтерфейс»

Студента 2м курсу, 6 групи,
спеціальності 121 «Інженерія
програмного забезпечення»
спеціалізації «Інженерія
програмного забезпечення»

Ткешелашвілі Давіда
Левановича

підпис студента

Науковий керівник
доктор технічних наук,
професор кафедри інженерії
програмного забезпечення та
кібербезпеки

Криворучко Олена
Володимирівна

підпис керівника

Гарант освітньої програми
доктор технічних наук,
професор кафедри інженерії
програмного забезпечення та
кібербезпеки

Криворучко Олена
Володимирівна

підпис гаранта

Київський національний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь магістр

Спеціальність 121 «Інженерія програмного забезпечення»

Затверджую

Зав. кафедри інженерії програмного
забезпечення та кібербезпеки

Криворучко О. В.

8 листопада 2019 р.

Завдання

на випускний кваліфікаційний проект студентіві

Ткешелашвілі Давіду Левановичу

(прізвище, ім'я, по батькові)

1. Тема випускного кваліфікаційного проекту «Розробка універсального
засобу адміністрування баз даних через web-інтерфейс»

Затверджена наказом ректора від "13" грудня 2019 р. № 4304

2. Строк здачі студентом закінченої проекту 01 грудня 2020

3. Цільова установка та вихідні дані до проекту

Мета проекту: створення веб-інтерфейсу для адміністрування бази даних.

Об'єкт дослідження: процес розробки веб-додатку

Предмет дослідження: розробка веб-інтерфейсу для адміністрування баз
даних.

4. Консультанти проекту із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Зміст випускного кваліфікаційного проекту (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ТА АНАЛІЗ ПІДХОДІВ ДО СТВОРЕННЯ ВЕБ-ДОДАТКУ

1.1. Характеристика веб-додатку та веб-інтерфейсу

1.2. Аналіз мов веб-програмування та платформ

1.3. Висновок до розділу 1

РОЗДІЛ 2. ЕТАПИ ТА ЗАСОБИ РОЗРОБКИ ПРОГРАМНОГО ПРОДУКТУ

2.1. Етапи розробки веб-додатку

2.2. Технічні вимоги до веб-додатку

2.3. Загальна концепція веб-додатку

2.4. Створення візуального начерку веб-додатку

2.5. Верстка інтерфейсу та кодування логіки додатку

2.6. Встановлення платформи Open Server та налаштування проекту

2.7. Висновок до розділу 2

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ЗАСОБУ АДМІНІСТРУВАННЯ БАЗИ ДАНИХ ЧЕРЕЗ ВЕБ-ІНТЕРФЕЙС

3.1. Елементи користувацького інтерфейсу

3.2. Сторінка логіну

3.3. Основна сторінка

3.4. Вигляд бази даних

3.5. Вигляд структури таблиці

3.6. Вигляд створення таблиці у форматі SQL Query

3.7. Вигляд таблиці

3.8. Додавання, видалення та редагування записів

3.9. Дерево елементів

3.10. Вигляд змінні бази даних

3.11. Висновок до розділу 3

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

6. Календарний план виконання проекту

№ пор.	Назва етапів випускного кваліфікаційного проекту	Строк виконання етапів проекту	
		за планом	фактично
1	2	3	4
1.	<i>Вибір теми випускного кваліфікаційного проекту</i>	20.09.2019	20.09.2019
2.	<i>Розробка та затвердження завдання на проект магістра</i>	08.11.2019	08.11.2019
3.	<i>Вступ та перелік літературних джерел</i>	24.01.2020	24.01.2020
4.	<i>Наукова стаття</i>	01.09.2020	01.09.2020
5.	<i>Технічне завдання</i>	28.02.2020	28.02.2020
6.	<i>Розділ 1. Дослідження та аналіз підходів до створення веб-додатку</i>	25.06.2020	25.06.2020
7.	<i>Розділ 2. Етапи та засоби розробки програмного продукту</i>	07.09.2020	07.09.2020
8.	<i>Розділ 3. Реалізація засобу адміністрування бази даних через веб-інтерфейс</i>	19.10.2020	19.10.2020
9.	<i>Програма та методика тестування</i>	21.10.2020	21.10.2020
10.	<i>Керівництво користувача</i>	23.10.2020	23.10.2020
11.	<i>Висновки та пропозиції</i>	30.10.2020	30.10.2020
12.	<i>Здача випускного кваліфікаційного проекту на кафедрі (перша перевірка)</i>	05.11.2020	05.11.2020
13.	<i>Підготовка автореферату та презентації доповіді</i>	05.11.2020	05.11.2020
14.	<i>Попередній захист випускного кваліфікаційного проекту</i>	25.11.2020- 27.11.2020	25.11.2020 -27.11.2020
15.	<i>Зовнішнє рецензування випускного кваліфікаційного проекту</i>	01.12.2020	01.12.2020
16.	<i>Підготовка до публічного захисту випускного кваліфікаційного проекту</i>	08.12.2020- 09.12.2020	

7. Дата видачі завдання «8» листопада 2019 р.

8. Науковий керівник випускного кваліфікаційного проекту Криворучко О. В.

(прізвище, ініціали, підпис)

9. Гарант освітньої програми Криворучко О. В.

(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент Ткешелашвілі Д. Л.

(прізвище, ініціали, підпис)

11. Відгук керівника випускного кваліфікаційного проекту

Науковий керівник випускного кваліфікаційного проекту

(підпис, дата)

Відмітка про попередній захист Криворучко О. В.

(ПІБ, підпис, дата)

12. Висновок про випускний кваліфікаційний проект

Випускний кваліфікаційний проект студента Ткешелашвілі Д. Л.

(прізвище, ініціали)

може бути допущена до захисту екзаменаційній комісії.

Гарант освітньої програми _____

Криворучко О. В.

(прізвище, ініціали, підпис)

Завідувач кафедри _____

Криворучко О. В.

(підпис, прізвище, ініціали)

« _____ » _____ 20 _____ р.

АНОТАЦІЯ

Відповідно до мети дослідження робота присвячена розробці універсального засобу адміністрування баз даних через web-інтерфейс.

Розглянуто сучасні тенденції ринку веб-розробок і проведено огляд діючих веб-інтерфейсів, веб-додатків, проаналізовані їх особливості, відзначені недоліки. Виконано аналіз мов розробки веб-програмування, та програмної платформи Open Server. Візуальний нарис веб-додатку був розроблений за допомогою веб-ресурсу WireFrame. Написання веб-додатку було здійснено за допомогою середовища розробки PhpStorm. Була розгорнута та налаштована серверна платформа Open Server.

Готове програмне забезпечення WebAdmin було успішно протестовано відповідно до функціональних вимог.

Ключові слова: *веб-сервер, браузер, адміністрування, веб-додаток, база даних.*

ABSTRACT

According to the purpose the research is devoted to developing of a universal tool for database administration through a web-interface. The modern tendencies of the market of web developments are considered and the review of the operating web interfaces, web applications is carried out, their features are analyzed, shortcomings are noted.

The analysis of web programming development languages and Open Server software platform is performed. The visual outline of the web application was developed using the web resource WireFrame The web application was written using PhpStorm. The Open Server platform was deployed and configured.

The finished WebAdmin software has been successfully tested according to the functional requirements.

Keywords: *web-server, browser, administration, web application, database.*

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ПЗ – програмне забезпечення

IDE (англ. Integrated Development Environment) – інтегроване середовище розробки

API – (англ. Application Programming Interface) – опис способів (набір класів, процедур, функцій, структур або констант), якими одна комп'ютерна програма може взаємодіяти з іншою програмою.

					<i>КНТЕУ 121 06-19.МР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.		Криворучко О.В.		19.10.20	Розробка універсального засобу адміністрування баз даних через web-інтерфейс	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Керівник		Криворучко О.В.		19.10.20		<i>ПС</i>	2	37
Гарант		Криворучко О.В.		19.10.20		Факультет інформаційних технологій 2м курс, 6 група		
Розробив		Ткешелашвілі Д.Л.		19.10.20				
					<i>Перелік умовних скорочень</i>			

ЗМІСТ	
ВСТУП.....	3
РОЗДІЛ 1. ДОСЛІДЖЕННЯ ТА АНАЛІЗ ПІДХОДІВ ДО СТВОРЕННЯ ВЕБ-ДОДАТКУ	4
1.1. <i>Характеристика веб-додатку та веб-інтерфейсу.....</i>	<i>4</i>
1.2. <i>Аналіз мов веб-програмування та платформ</i>	<i>6</i>
1.3. <i>Висновки до розділу 1</i>	<i>13</i>
РОЗДІЛ 2. ЕТАПИ ТА ЗАСОБИ РОЗРОБКИ ПРОГРАМНОГО ПРОДУКТУ	14
2.1. <i>Етапи розробки веб-додатку.....</i>	<i>14</i>
2.2. <i>Технічні вимоги до веб-додатку</i>	<i>14</i>
2.3. <i>Загальна концепція веб-додатку</i>	<i>16</i>
2.4. <i>Створення візуального начерку веб-додатку</i>	<i>17</i>
2.5. <i>Верстка інтерфейсу та кодування логіки додатку.....</i>	<i>19</i>
2.6. <i>Встановлення платформи Open Server та налаштування проекту.....</i>	<i>20</i>
2.7. <i>Висновки до розділу 2</i>	<i>21</i>
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ЗАСОБУ АДМІНІСТРУВАННЯ БАЗИ ДАНИХ ЧЕРЕЗ ВЕБ-ІНТЕРФЕЙС	22
3.1. <i>Елементи користувацького інтерфейсу.....</i>	<i>22</i>
3.2. <i>Сторінка логіну.....</i>	<i>23</i>
3.3. <i>Основна сторінка</i>	<i>24</i>
3.4. <i>Вигляд бази даних</i>	<i>27</i>
3.5. <i>Вигляд структури таблиці.....</i>	<i>28</i>
3.6. <i>Вигляд створення таблиці у форматі SQL Query.....</i>	<i>28</i>
3.7. <i>Вигляд таблиці</i>	<i>29</i>
3.8. <i>Додавання, видалення та редагування записів</i>	<i>30</i>
3.9. <i>Дерево елементів</i>	<i>32</i>
3.10. <i>Вигляд змінні бази даних</i>	<i>33</i>
3.11. <i>Висновок до розділу 3</i>	<i>34</i>
ВИСНОВКИ ТА ПРОПОЗИЦІЇ	35
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	36
ДОДАТКИ.....	38

					<i>КНТЕУ 121 06-19.МР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Розробка універсального засобу адміністрування баз даних через web-інтерфейс	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Зав. каф.		Криворучко О.В.		19.10.20		<i>Зміст</i>	3	37
Керівник		Криворучко О.В.		19.10.20		Факультет інформаційних технологій 2м курс, 6 група		
Гарант		Криворучко О.В.		19.10.20				
Розробив		Ткешелашвілі Д.Л.		19.10.20	<i>Зміст</i>			

ВСТУП

Актуальність. Сучасне життя нерозривно пов'язане з інформаційними технологіями, які докорінно змінюють комунікативні та пізнавальні можливості суспільства. ІТ спеціалісти кожного дня розробляють велику кількість різноманітного ПЗ у різноманітних галузях.

З розвитком ІТ вдосконалюються середовища розробки, інструменти, що допомагають виконувати роботу ІТ спеціалістів швидше, зручніше, ефективніше.

Основи сучасних інформаційних технологій становлять бази даних (БД) та системи керування базами даних, роль яких як єдиного засобу зберігання, обробки й доступу до великих обсягів інформації постійно зростає.

Тому було вирішено створити інструмент адміністрування баз даних через веб-інтерфейс для веб-програмістів та усіх споріднених професій у яких є необхідність роботи з великою кількістю даних на серверах.

Мета проекту: створення веб-інтерфейсу для адміністрування бази даних.

Об'єкт дослідження: – процес розробки веб-додатку

Предмет дослідження: – веб-інтерфейс для адміністрування баз даних.

Зм.	Аркуш	№ докум.	Підпис	Дата	<i>КНТЕУ 121 06-19.МР</i>			
Зав. каф.		Криворучко О.В.		24.01.20	Розробка універсального засобу адміністрування баз даних через web-інтерфейс	Стадія	Аркуш	Аркушів
Керівник		Криворучко О.В.		24.01.20		В	4	37
Гарант		Криворучко О.В.		24.01.20		Факультет інформаційних технологій 2м курс, 6 група		
Розробив		Ткешелашвілі Д.Л.		24.01.20				
					<i>Вступ</i>			

РОЗДІЛ 1

ДОСЛІДЖЕННЯ ТА АНАЛІЗ ПІДХОДІВ ДО СТВОРЕННЯ ВЕБ-ДОДАТКУ

1.1. Характеристика веб-додатку та веб-інтерфейсу

Веб-додаток – клієнт-серверний додаток, в якому клієнт взаємодіє з веб-сервером за допомогою браузера. Логіка веб-додатку розподілена між сервером і клієнтом, зберігання даних здійснюється, переважно, на сервері, обмін інформацією відбувається по мережі.

Одним з переваг такого підходу є той факт, що клієнти не залежать від конкретної операційної системи користувача, тому веб-додатки є межплатформенними службами.

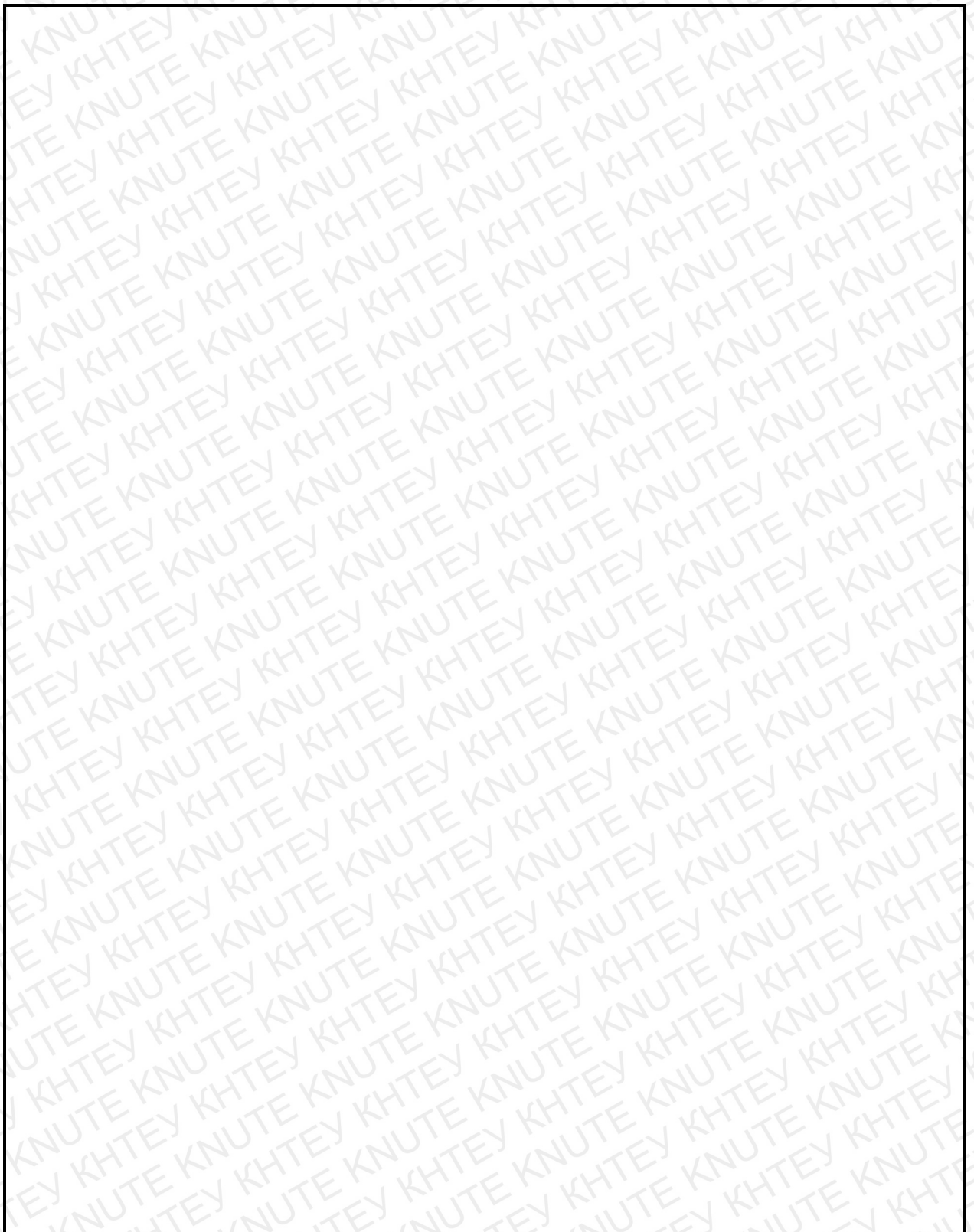
Веб-додаток складається з клієнтської і серверної частин, тим самим реалізуючи технологію «клієнт-сервер».

Клієнтська частина реалізує користувальницький інтерфейс, формує запити до сервера і обробляє відповіді від нього.

Серверна частина отримує запит від клієнта, виконує обчислення, після цього формує веб-сторінку і відправляє її клієнту через мережу з використанням протоколу HTTP.

Саме веб-додаток може виступати в якості клієнта інших служб, наприклад, бази даних або іншого веб-додатку, розташованого на іншому сервері. Яскравим прикладом веб-додатки є система управління вмістом статей Вікіпедії: безліч її учасників можуть брати участь у створенні мережевої енциклопедії, використовуючи для цього браузери своїх операційних систем (будь то Microsoft Windows, GNU / Linux або будь-яка

					<i>КНТЕУ 121 06-19.МР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Розробка універсального засобу адміністрування баз даних через web-інтерфейс <i>Дослідження та аналіз підходів до створення веб-додатку</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Зав. каф.	Криворучко О.В.			25.06.20		<i>РІ</i>	<i>5</i>	<i>37</i>
Керівник	Криворучко О.В.			25.06.20		Факультет інформаційних технологій		
Гарант	Криворучко О.В.			25.06.20		2м курс, 6 група		
Розробив	Ткешелашвілі Д.Л.			25.06.20				



Зм.	Аркуш	№ докум.	Підпис	Дата				
Зав. каф.		Криворучко О.В.		25.06.20	Розробка універсального засобу адміністрування баз даних через web-інтерфейс	Стадія	Аркуш	Аркушів
Керівник		Криворучко О.В.		25.06.20		РІ	5	37
Гарант		Криворучко О.В.		25.06.20		Факультет інформаційних технологій 2м курс, 6 група		
Розробив		Ткешелашвілі Д.Л.		25.06.20				
					Дослідження та аналіз підходів до створення веб-додатку			

інша операційна система) і не завантажуючи додаткових виконуваних модулів для роботи з базою даних статей.

Веб-інтерфейс – веб-сторінка або сукупність веб-сторінок, що надає користувальницький інтерфейс для взаємодії з сервісом або пристроєм за допомогою протоколу HTTP і веб-браузера. Веб-інтерфейси набули широкого поширення в зв'язку з ростом популярності всесвітньої павутини і відповідно - бути широко розповсюдженим веб-браузерів. Одним з основних вимог до веб-інтерфейсів є їх однаковий зовнішній вигляд і однакова функціональність при роботі в різних браузерах.

Класичним і найбільш популярним методом створення веб-інтерфейсів є використання HTML із застосуванням CSS і JavaScript'a. Однак різна реалізація HTML, CSS, DOM і інших специфікацій в браузерах викликає проблеми при розробці веб-додатків і їх подальшої підтримки. Крім того, можливість користувача налаштувати багато параметрів браузера (наприклад, розмір шрифту, кольору, відключення підтримки сценаріїв) може перешкоджати коректної роботи інтерфейсу.

Інший (менш універсальний) підхід полягає у використанні Adobe Flash, Silverlight або Java-апплетів для повної або часткової реалізації призначеного для користувача інтерфейсу. Оскільки більшість браузерів підтримує ці технології (як правило, за допомогою плагінів), Flash- або Java-додатки можуть виконуватися з легкістю. Так як вони надають програмісту більший контроль над інтерфейсом, вони здатні обходити багато несумісності в конфігураціях браузерів, хоча несумісність між Java або Flash реалізаціями на стороні клієнта може призводити до різних ускладнень.

1.2. Аналіз мов веб-програмування та платформ

Створити якісну web-сторінку непросто, для цього будуть потрібні не тільки навички дизайнера для красивого і стильного оформлення, а й досвід програмування. Складність і обсяг програм, необхідних для створення того чи

						Аркуш
					<i>КНТЕУ 121 06-19.МР</i>	6
Зм.	Аркуш	№ докум	Підпис	Дата		

іншого сайту, залежить від безлічі факторів, таких, як структура, мета, необхідна функціональність, забезпечення надійності і т.п.

Для створення сучасних сайтів зазвичай використовуються: HTML, CSS, JavaScript, PHP.

HTML – це не мова програмування і не мова оформлення документів. Це, в першу чергу, засіб розмітки тексту. На сьогоднішній день HTML залишається самим універсальним, навіть незамінним засобом розмітки гіпертексту, а, отже, і публікації в Інтернет.

Написання Web-сторінок на HTML не вимагає інтерпретації вихідного коду в двійковий код. Мова розмітки гіпертексту за визначенням повинен інтерпретуватися браузером. Це, безумовно, накладає деякі обмеження на можливості мови і на сумісність нових конструкцій зі старими версіями браузерів. Сучасні Web-сторінки вже не обходяться одним тільки HTML. Його гармонійно доповнюють кошти динамічного HTML: скрипт мови JavaScript і / або VBScript, каскадні таблиці стилів (CSS), іноді присутні Java-аплети.

CSS (Cascading Style Sheets) – мова таблиць каскадних стилів. Він розроблений для того, щоб розширити можливості по оформленню Web-сторінок.

CSS використовується веб-розробниками для завдання зовнішнього вигляду (шрифтів, квітів, відступів, розташування та ін.) Веб-сторінок. CSS розроблений для відділення основного вмісту документа (написаного на мові розмітки, наприклад HTML) від оформлення цього вмісту (написаного на CSS). Таке відділення надає веб-розробникам велику гнучкість, спрощує завдання зовнішнього вигляду документів та оформлення повторюваних елементів розмітки.

При використанні HTML і CSS важливо розуміти, наступне:

						Аркуш
					<i>КНТЕУ 121 06-19.МР</i>	7
Зм.	Аркуш	№ докум	Підпис	Дата		

HTML-код формує текст логічно, тобто задає структури Web-сторінки: розташування і порядок проходження абзаців, графічних зображень, рядків і осередків в таблиці і особливе значення окремих фрагментів тексту.

Таблиці стилів CSS формують тексти фізично, тобто задають уявлення Web-сторінки: яким шрифтом будуть набрані звичайний текст абзаців, яким кольором виділити заголовки, чи будуть у таблиці рамка тощо.

Правила хорошого тону Web-дизайну вимагають, щоб уявлення Web-сторінки було відокремлено від її структури. Тому професійні Web-дизайнери по можливості виносять ухвалу стилів CSS в окремі файли.

Каскадні таблиці стилів по суті своїй динамічні. Вони дозволяють визначати, як буде виглядати документ при завантаженні і не більше того. Але властивості Web-сторінок, створених за допомогою CSS, можна динамічно змінювати за допомогою мови JavaScript.

JavaScript – мультіпарадігменний мову програмування. JavaScript зазвичай використовується як вбудований мова для програмного доступу до об'єктів додатків. Найбільш широке застосування знаходить в браузерах як мова сценаріїв для додання інтерактивності веб-сторінок.

Популярність мови JavaScript пов'язана з його широкими можливостями по взаємодії з елементами веб-сторінки без її перезавантаження.

JavaScript використовується в клієнтській частини веб-додатків: клієнт-серверних програм, в якому клієнтом є браузер, а сервером – веб-сервер, що мають розподілену між сервером і клієнтом логіку. Обмін інформацією в веб-додатках відбувається по мережі. Одним з переваг такого підходу є той факт, що клієнти не залежать від конкретної операційної системи користувача, тому веб-додатки є кроссплатформенними сервісами. Але це не все. Сфер застосування дуже багато, ось деякі з них:

- JavaScript використовується в AJAX, популярному підході до побудови призначених для користувача інтерфейсів веб-додатків

						Аркуш
					<i>КНТЕУ 121 06-19.МР</i>	8
Зм.	Аркуш	№ докум	Підпис	Дата		

- Користувацькі скрипти в браузері
- Серверні додатки, мобільні додатки

В даний час підтримується переважною більшістю хостинг-провайдерів і є одним з лідерів серед мов, що застосовуються для створення динамічних веб-сайтів. Це дозволяє ховати і показувати фрагменти дизайну, переміщати їх і міняти оформлення. Шляхом таких дій можна створювати презентаційні ефекти, меню, невеликі гри, обробляти дані форм і керувати вмістом.

Підтримка шарів. Шари надають собою фрагменти HTML, які можна розміщувати на веб-сторінці шляхом накладення їх один на одного з точністю до пікселя. Шари все більше набувають визнання як засіб верстки веб-сторінок і створення різних ефектів. Частково це пов'язано з тим, що їх параметри легко змінюються динамічно через скрипти, що дає можливість створювати меню і інші виразні засоби дизайну.

Робота з формами. Через скрипти зручно отримувати і обробляти будь-які дані форм, це дозволяє перевірити інформацію на правильність введення перед її відправкою на сервер. Можна створити «захист від дурня» для контролю того, щоб в елементи форм вводилися коректні повідомлення. Так, якщо в текстовому полі просять ввести число, слід відразу припинити можливість введення нечислових символів.

Робота з зображеннями. Через скрипти можна робити попередню завантаження зображень. Ще до прямого звернення до зображень, браузер поміщає їх в свою пам'ять, щоб у міру необхідності швидше відобразити в документі. Основні параметри зображень, такі як: ширина, висота картинок і адреса графічного файлу, теж можна міняти динамічно. Це дозволяє створювати ефект перекочування, коли малюнок змінюється на інший при наведенні на нього курсора миші і назад, коли курсор виводиться з зображення. Подібним методом робляться і галереї фотографій – зміна зображень відбувається в межах однієї сторінки.

						Аркуш
						9
Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121 06-19.МР	

Визначення дати і часу. Дуже улюблена чомусь багатьма дизайнерами можливість вставляти поточну дату, і навіть час до себе на сторінку, реалізується за допомогою об'єкту Date, спеціально створеного для роботи з датою.

Все що можна отримати від часу, тут уже є. Робота з cookies. JavaScript підтримує повноцінну роботу з cookies - невеликі текстові файли на локальному комп'ютері, в яких зберігається технічна інформація. Cookies можна використовувати для збереження дати останнього відвідування читача, паролів, а також будь-якої інформації про дії користувача на сайті. Подібне застосування дозволяє персоналізувати сайт і зробити його більш зручним для відвідувачів.

Відстеження подій. Подією називається певним чином впливати користувача або зміна стану документа. JavaScript відстежує більшість подій і дозволяє визначати реакцію на них. Наприклад, при завантаженні веб-сторінки відбувається подія onLoad. Якщо необхідно запустити скрипт відразу після завантаження документа, слід цієї події призначити функцію, яка буде виконуватися при його настанні.

Створення нових вікон. Скрипти дозволяють створювати нові вікна, задавати у цих вікон вид відображення і формувати їх вміст. Причому, у створених таким способом вікон можна відключати адресний рядок, смуги прокрутки, меню і управляти їх розміром і положенням на екрані.

Перевірка браузера. Підхід різних браузерів до об'єктної моделі, хоч і стандартизований, але ще не всіма до кінця прийнятий. Тому для створення універсального документа, однаково працює всюди, доводиться робити перевірку, який в даний момент використовується браузер, і давати йому відповідний код.

						Аркуш
					<i>КНТЕУ 121 06-19.МР</i>	10
Зм.	Аркуш	№ докум	Підпис	Дата		

Математичні функції. JavaScript містить всі необхідні арифметичні операції, підтримує всі стандартні математичні функції, як з цілими числами, так і з плаваючою крапкою.

Управління вмістом документа. Через метод write () можна вставляти будь-яку інформацію в уже створений документ, наприклад, сьогоднішню дату. А також формувати повністю новий документ динамічно. Це дозволяє враховувати особливості поведінки різних браузерів і операційних систем, виводячи для них свій власний текст.

PHP - це мова програмування, призначений для створення сайтів. PHP дозволяє автоматизувати роботу з сайтом. Це скрипт-мову, вбудований в HTML, який інтерпретується і виконується на сервері. Відмінність PHP від JavaScript, полягає в тому, що PHP-скрипт виконується на сервері, а клієнту передається результат роботи, тоді як в JavaScript-код повністю передається на клієнтську машину і тільки там виконується.

У минулому столітті, щоб створити сайт «на світовому рівні», досить було просто вміти працювати з HTML і володіти художнім смаком. Через деякий час вимоги ускладнилися: використання JavaScript і Dynamic HTML стало рутинним, а дизайн сторінок, які не мають подібних «прикрас», вважався застарілим. Незабаром правилом «хорошого тону» для корпоративного сайту стало застосування складних програм на Perl або C ++. Однак Web-дизайнери не бажали миритися з такою ситуацією. І на світ з'явився PHP - мова програмування, що володіє можливостями складних скриптових мов, але в той же час дивно простий і легкий у вивченні і застосуванні.

PHP є впроваджуються мовою сценаріїв. Мета мови полягає в тому, щоб дозволити веб-розробникам швидко створювати динамічно генеруються сторінки. Крім того, розробникам Web-додатків немає необхідності говорити, що web-сторінки - це не тільки текст і картинки. Гідний уваги сайт повинен підтримувати деякий рівень інтерактивності з користувачем: пошук

					Аркуш
					11
Зм.	Аркуш	№ докум	Підпис	Дата	

КНТЕУ 121 06-19.МР

інформації, продаж продуктів, конференції і т.п. PHP працює як частина Web-сервера. У цій мові немає строгої типізації даних і немає необхідності в діях по виділенню / звільненню пам'яті. Програми, написані на PHP, досить легко читані. Написаний PHP - код легко візуально прочитати і зрозуміти.

Можливості PHP:

На PHP можна зробити все: обробляти дані з форм, генерувати динамічні сторінки, одержувати і посилати сеанси - куки (cookies). Крім цього в PHP включена підтримка багатьох баз даних (databases), що робить написання Web-додатків з використанням БД до неможливості простим. Додатково до всього PHP розуміє протоколи IMAP, SNMP, NNTP, POP3 і навіть HTTP, а також має можливість працювати з сокетом (sockets) і спілкуватися по інших протоколах. PHP може серйозно полегшити роботу творця сайту. За допомогою команди include () можна автоматично вставляти один і той же фрагмент HTML-коду в безліч сторінок, просто помістивши в них цю команду з ім'ям файлу, що містить код загальної частини. В результаті для поновлення такої загальної частини буде достатньо відредагувати один файл - той, який її, власне, і містить, і відповідно зміняться всі інші сторінки. Також, при оновленні новин редагування доведеться піддавати лише сам їх текст, а не HTML-код головної сторінки, що і самому робити легше, і недосвідченому помічникові довірити можна.

Open Server -це одна із найпопулярніших програмних платформ для створення локальних сайтів на базі Windows. Робота з Open Server на своєму комп'ютері дозволяє створювати сайти не виходячи в інтернет та не використовуючи сторонніх серверів.

Багато веб-програмістів саме так і створюють та тестують веб-ресурси. Це зручно, не залежите від роботи сторонніх серверів, можна вести скільки завгодно проектів.

						Аркуш
					<i>КНТЕУ 121 06-19.МР</i>	
Зм.	Аркуш	№ докум	Підпис	Дата		12

Open Server являє собою так званий WAMP комплекс, суть якого можна зрозуміти з розшифровки цієї аббревіатури:

Windows – операційна система, для роботи в якій призначений даний локальний сервер

Apache – web-сервер, який «піднімається» при запуску програми Open Server.

MySQL – дуже популярна система управління базами даних, яка є обов'язковою умовою для роботи багатьох двигунів сайтів, в тому числі таких популярний, як Joomla та WordPress

PHP - інтерпретатор серверного мови програмування, на якому написано більшість CMS і створено безліч інших веб-додатків

1.3. Висновки до розділу 1

У першому розділі ми розглянули характеристики веб-додатку та інтерфейсів. Був проведений аналіз предметної області, відзначені недоліки та переваги мов програмування, в результаті чого були виділені основні вимоги до розроблюваного веб-додатку для адміністрування баз даних.

Для створення ПЗ було вибрано JavaScript для опису дій які будуть виконуватися на клієнтській частині веб-додатку та використання мови програмування PHP для опису серверної частини веб-додатку, а розміщуватися буде на серверній платформі Open Server.

						Аркуш
					<i>КНТЕУ 121 06-19.МР</i>	13
Зм.	Аркуш	№ докум	Підпис	Дата		

РОЗДІЛ 2

ЕТАПИ ТА ЗАСОБИ РОЗРОБКИ ПРОГРАМНОГО ПРОДУКТУ

2.1. Етапи розробки веб-додатку

Цикл розробки ПЗ складається з:

- Аналіз вимог, опрацювання технічного завдання (ТЗ);
- Проектування. Створення макета веб-інтерфейсу, концепції веб-додатку
- Розробка та кодування;
- Функціональне та інтеграційне тестування
- Супроводження та підтримка

2.2. Технічні вимоги до веб-додатку

1. Тематика і призначення веб-ресурсу:

Мета веб-додатку – адміністрування СУБД MySQL у вікні браузера

Веб-додаток виконує такі функції:

- Підключитися до сервера баз даних за допомогою імені користувача та пароля
- Можливість вибору існуючої бази даних або створення нової
- Список variables бази даних
- Відображення користувачів, їх прав, можливість їх змінити
- Створення, відображення, оновлення, видалення баз даних
- Створення, відображення, оновлення, видалення таблиць
- Додавання та видалення стовпців
- Виконання SQL скриптів

Зм.	Аркуш	№ докум.	Підпис	Дата	<i>КНТЕУ 121 06-19.МР</i>			
Зав. каф.		Криворучко О.В.		07.09.20	Розробка універсального засобу адміністрування баз даних через web-інтерфейс	Стадія	Аркуш	Аркушів
Керівник		Криворучко О.В.		07.09.20		P2	14	37
Гарант		Криворучко О.В.		07.09.20		Факультет інформаційних технологій 2м курс, 6 група		
Розробив		Ткешелашвілі Д.Л.		07.09.20				
					<i>Етапи та засоби розробки програмного продукту</i>			

- Пошук даних в базах даних, таблицях та відображення результатів
- Підтримка основних типів даних
- Адміністрування декількох серверів

Цільова аудиторія веб-додатку включає наступні групи людей:

- Веб-програмісти
- Користувачі хостингів

Мови веб-ресурсу:

Одна мовна версія веб-сайту: Англійська. У подальшому можливе створення і україномовної версії веб-додатку. 2. Структура та функціональність веб-ресурсу

Вимоги до програмної платформи:

- Веб-сервер: в нашому випадку це Apache, який входить до складу Open Server.
- HTTP модуль – Apache_2.4-PHP_7.2 та обов'язкове ведення логуювання викликів
- Підтримка PHP - PHP_7.2
- Підтримка СУБД– MySQL-8.0 та обов'язкове ведення логуювання викликів
- Кодування MySQL серверу– UTF-8_GENERAL_CI
- Щоб отримати доступ до ПЗ потрібен веб-браузер з ввімкненими cookies та JavaScript

Вимоги до дизайну користувацького інтерфейсу:

Загальний дизайн сторінок веб-додатку має базуватися на стабільних стандартах та не повинен використовувати можливостей, доступних тільки в окремих браузерах.

Дизайн веб-сайту повинен відповідати усім сучасним нормам та напрямкам у веб-дизайні, бути простим й одночасно сучасним та інформативним. Дизайн необхідно виконати в спокійній кольоровій гамі.

					<i>КНТЕУ 121 06-19.МР</i>	Аркуш
						15
Зм.	Аркуш	№ докум	Підпис	Дата		

Дизайн повинен бути виконаний в стилі мінімалізму (нічого зайвого, відволікаючого увагу).

Інтерфейс користувача повинен забезпечувати інтуїтивно зрозуміле представлення структури розміщеної на ньому інформації та швидкий і логічний перехід між розділами.

Дизайн веб-інтерфейсу має коректно відображатись в наступних основних браузерях: ІЕ 8.0 і вище, останні версії Mozilla Firefox, Opera, Chrome

2.3. Загальна концепція веб-додатку

На сьогоднішній день існує два принципових підходи до створення веб-додатків: традиційні веб-додатки, велика частина логіки яких виконується на сервері, а також односторінкові додатки, логіка призначеного для користувача інтерфейсу яких виконується переважно в веб-браузері, а взаємодія з веб-сервером здійснюється головним чином через веб-API.

Також можливий гібридний підхід, при якому в простому випадку в рамках великого традиційного веб-додатку розміщуються одне або кілька повнофункціональних підлеглих додатків, побудованих на основі односторінкової моделі.

Більшість користувачів багатьох веб-додатків можуть працювати тільки з функціями читання. Додатки, призначені виключно або переважно для читання, зазвичай набагато простіше тих, в яких реалізується управління власністю.

Наприклад, в пошуковій системі цілком достатньо реалізувати одну точку входу з текстовим полем і другу сторінку для відображення результатів пошуку. Запити можуть виконувати анонімні користувачі, в зв'язку з чим на стороні клієнта практично не потрібно реалізовувати логіку. Аналогічним чином, публічні додатки блогів або систем управління вмістом працюють переважно з вмістом і практично не мають функцій, що реалізуються на стороні клієнта.

						Аркуш
					<i>КНТЕУ 121 06-19.МР</i>	16
Зм.	Аркуш	№ докум	Підпис	Дата		

Такі додатки легко створювати в форматі традиційних серверних веб-додатків, які виконують логіку на веб-сервері і перетворюють HTML-код для відображення в браузері. Той факт, що кожна унікальна сторінка сайту має власний URL-адресу, який може додаватися в закладки або індексуватися пошуковими системами (така поведінка реалізується за замовчуванням і не вимагає додавання в додаток окремої функції), також є очевидною перевагою такого сценарію.

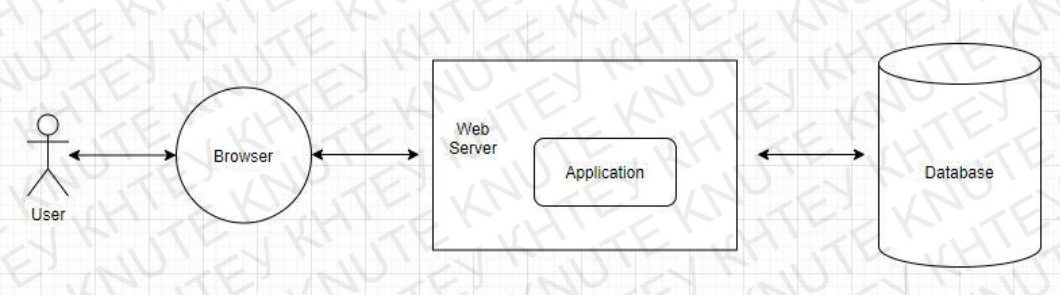


Рис.2.1. Загальна концепція веб-додатку

2.4. Створення візуального начерку веб-додатку

Всі етапи створення сайту важливі для якісного функціонування ресурсу, але розробка дизайну - один з головних. Адже дизайн - це те, що бачить відвідувач в першу чергу, оцінює його і приймає рішення залишитися на сторінці або закрити вкладку браузера. Використовуючи сайт WireFrame був створений візуальний начерк користувацького інтерфейсу (рис.2.1, рис. 2.2).

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		17

Db connection

Username

Password

Host

Remember in cookies

Рис. 2.1. Візуальний начерк сторінки авторизації

Databases Sign out

Time : Affected:

Database:	Time	Affected
██████████	██████████	██████████
██████████	██████████	██████████
██████████	██████████	██████████
██████████	██████████	██████████
██████████	██████████	██████████
██████████	██████████	██████████

Author: David Tkeshelashvili Email: david.tkesh@gmail.com

Рис. 2.2. Візуальний начерк головної сторінки

					Аркуш
					<i>КНТЕУ 121 06-19.МР</i>
Зм.	Аркуш	№ докум	Підпис	Дата	18

Грунтуючись на технічному завданні, створюю кнопки відповідно до зазначених функцій, списки для відображення даних, та інші графічні елементи. Іншими словами, той прототип, який був створений на першому етапі розробки сайту, отримує зовнішній вигляд користувацького інтерфейсу.

2.5. Верстка інтерфейсу та кодування логіки додатку

За допомогою відкритого фреймворку Bootstrap (фреймворк, котрий використовують веб-програмісти для швидкої верстки адаптивних дизайнів сайтів та веб-додатків. Спрощує написання HTML, CSS, Js) переноситься готовий макет в робочий продукт. Ресурс відображається у всіх існуючих браузерах. Все це доповниться PHP кодом.

За допомогою середовища розробки PhpStorm було написано HTML, CSS та PHP частина веб-додатку (рис. 2.3).

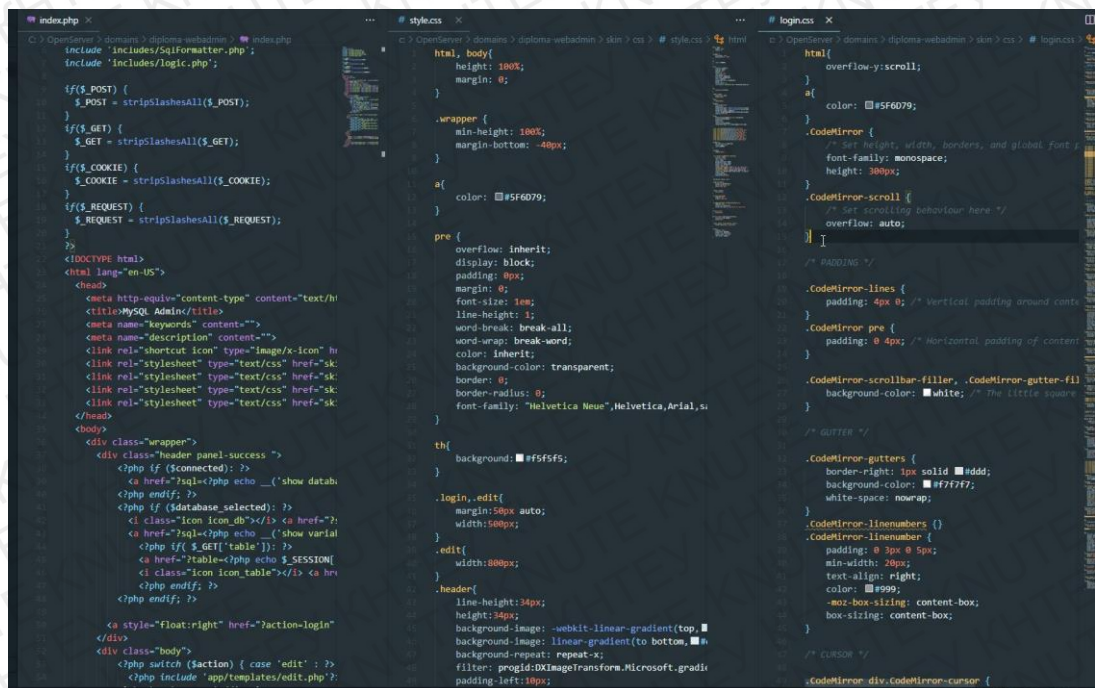


Рис. 2.3 PhpStorm та код HTML, CSS, PHP

Для динамічного відображення таблиць баз даних та іншого – було використано PHP.

					Аркуш
					19
Зм.	Аркуш	№ докум	Підпис	Дата	

2.6. Встановлення платформи Open Server та налаштування проекту

В першу чергу нам необхідно підібрати необхідну версію Open Server. Яка відповідає нашим потребам. Всі необхідні модулі є в складі базової версії Open Server. Тому завантажуюмо та розгортаємо її локально на комп'ютері. Тепер маємо портативну серверну платформу.

Щоб успішно запустити наш проект необхідно налагодити Open Server відповідно до ТЗ.

На вкладці налаштувань «Модулі» виставляємо HTTP, PHP, MySQL модулі відповідно : Apache_2.4-PHP_7.2, PHP_7.2, MySQL-8.0

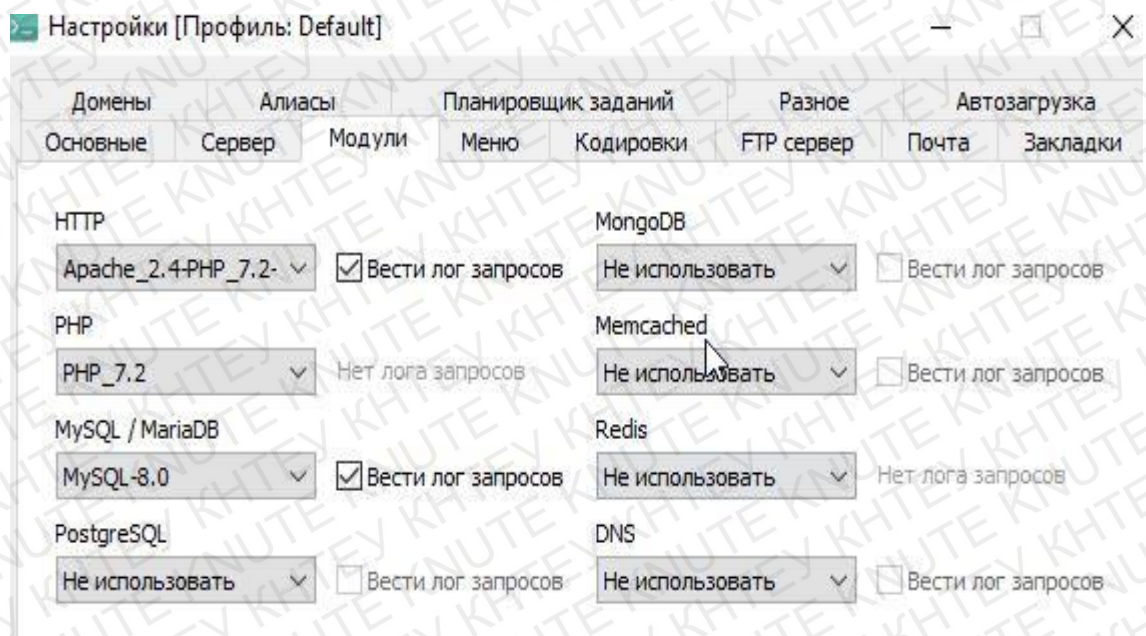


Рис. 2.4. Вкладка «Модулі» налаштувань Open Server

На вкладці «Кодування» встановлюємо стандартне кодування «UTF-8_GENERAL_CI».

На вкладці «Домени» додаємо новий локальний домен який будемо використовувати для запуску проекту.

Було додано домен «http://diploma-webadmin.local»

					Аркуш
					20
Зм.	Аркуш	№ докум	Підпис	Дата	

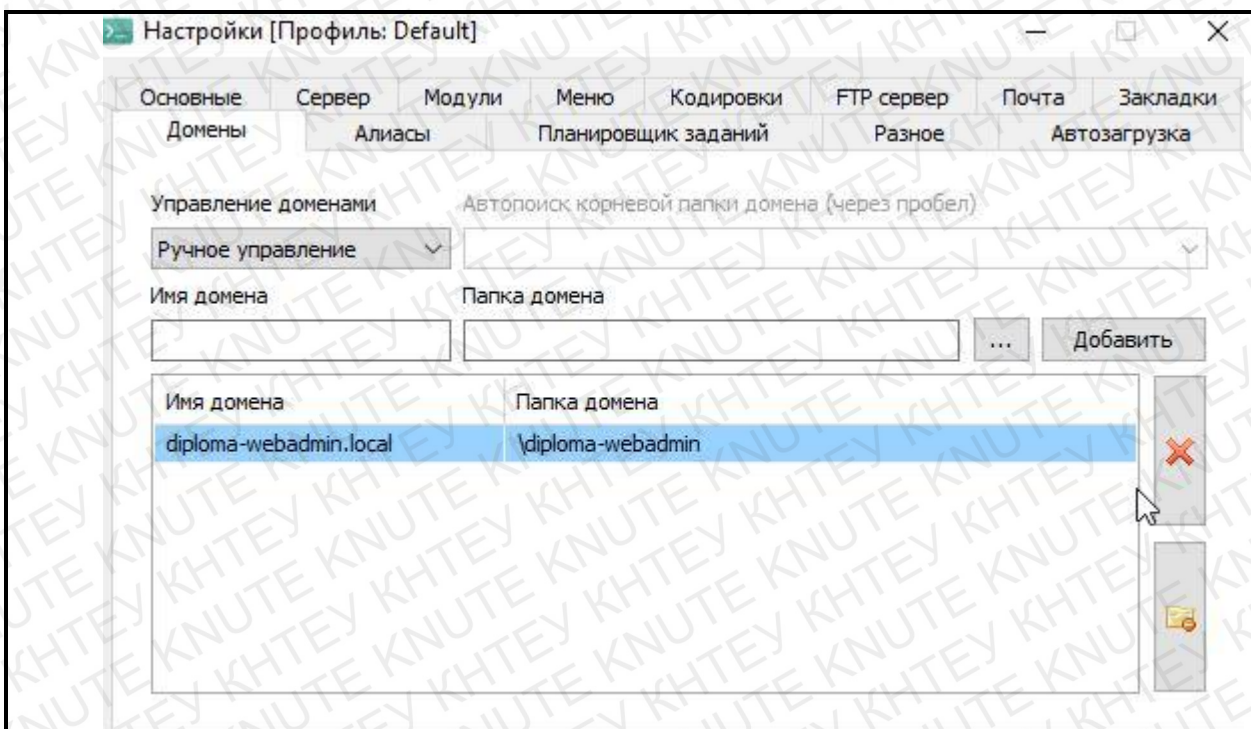


Рис. 2.5. Вкладка «Домени» налаштувань Open Server

На цьому наши налаштування завершуються. Але щоб запустити проект, необхідно в папку домену завантажити складові веб-додатку: HTML, CSS, JS, PHP файли. Після цього, коли всі необхідні файли будуть завантажені - можна починати роботу.

2.7. Висновки до розділу 2

Отже, в цьому розділі мною досліджено особливості веб-додатку, його специфікація та було сформована загальна концепція веб-додатку.

Були представлені етапи розробки веб-додатку. Згідно яких буде проводитися розробка ПЗ. Візуальний нарис веб-додатку був розроблений за допомогою веб-ресурсу Wireframe.

Як середовище розробки була обрана PhpStorm.

Для реалізації поставленої мети була розгорнута та налаштована серверна платформа Open Server, для подальшої відладки проекту.

						Аркуш
						21
Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121 06-19.МР	

РОЗДІЛ 3

РЕАЛІЗАЦІЯ ЗАСОБУ АДМІНІСТРУВАННЯ БАЗИ ДАНИХ ЧЕРЕЗ ВЕБ-ІНТЕРФЕЙС

3.1. Елементи користувацького інтерфейсу

Веб-додаток складається з 4х темплейтів та однієї головної сторінки яка переключає ці темплейти:

- Сторінка логіну
- Основна сторінка
- Сторінка редагування запису
- Сторінка додавання запису

Коли користувач взаємодіє з веб-додатком через веб-інтерфейс він виконує деякі дії і в залежності від дії, яку виконує користувач, перемикаються темплейти інтерфейсу. (рис. 3.1.)

```
</div>
<div class="body">
  <?php switch ($action) { case 'edit' : ?>
    <?php include 'app/templates/edit.php'?>
  <?php break; case 'add' : ?>
    <?php include 'app/templates/add.php'?>
  <?php break; case 'login': ?>
    <?php include 'app/templates/login.php'?>
  <?php break; case 'disconnect': ?>
    <?php include 'app/templates/login.php'?>
  <?php break; case 'index' : ?>
    <?php include 'app/templates/index.php'?>
  <?php break; } ?>
</div>
</div>
```

Рис. 3.1. Фрагмент програмного коду «Переключення темплейтів інтерфейсу»

					<i>КНТЕУ 121 06-19.МР</i>			
Зм.	Аркуш	№ докум.	Підпис	Дата	Розробка універсального засобу адміністрування баз даних через web-інтерфейс <i>Реалізація засобу адміністрування бази даних через веб-інтерфейс</i>	Стадія	Аркуш	Аркушів
Зав. каф.	Криворучко О.В.			19.10.20		РЗ	22	37
Керівник	Криворучко О.В.			19.10.20		Факультет інформаційних технологій		
Гарант	Криворучко О.В.			19.10.20		2м курс, 6 група		
Розробив	Ткешелашвілі Д.Л.			19.10.20				

3.2. Сторінка логіну

При відкритті веб-додатку відображається темплейт інтерфейсу логіну відповідно до дії користувача «login». Ініціалізується сесія та створюється cookie запис «PHPSESSID», що містить ідентифікатор сеансу.

На сторінці логіну знаходяться такі елементи:

- Поле вводу для користувача бази даних
- Поле вводу для паролю копістувача
- Поле вводу для адреси сервера MySQL
- Чекбокс «Remember in cookies»
- Кнопка «Try to connect»

Після того як користувач заповнив всі необхідні поля, то виконується валідація введених даних. Також є можливість зберегти в куки, тобто, користувач який поставить прапорець у чекбоксі «Remember in cookies», матиме змогу автоматичного заповнення полів. Якщо валідація успішна, то виконується метод підключення до бази даних. (рис. 3.2.)

						Аркуш
					<i>КНТЕУ 121 06-19.МР</i>	23
Зм.	Аркуш	№ докум	Підпис	Дата		


```

function connect($host, $user, $password, $charset = false) {
    if ($charset)
        $this->charset = $charset;

    try {
        if (!empty($host) && !empty($user)) {
            mysqli_report(MYSQLI_REPORT_STRICT);

            // Host validation
            if (!$this->isValidHost($host)) {
                $this->error = 'Host name is not valid.';
                return false;
            }

            $mysqli = new mysqli($host, $user, $password);

            /* check connection */
            if ($mysqli->connect_errno) {
                printf("Connect failed: %s\n", $mysqli->connect_error);
                $mysqli->close();
            } else {
                $this->connid = $mysqli;
                return true;
            }
        }
    }
}

```

Рис. 3.2. Метод «connect»

3.3. Основна сторінка

Після авторизації виконується дія index та відповідно відкривається темплейт основної сторінки. Так як ми у стані «connected» до сервера бази даних, то виконується get виклик “show databases;”. Після цього з сервера баз даних повертаються список баз даних. Створюються візуальні елементи: (рис 3.3.)

- Header таблиці
- Таблиця повернутими в результаті get виклику

						Аркуш
						24
Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121 06-19.МР	


```

<table id="result_table" class="table table-hover table-condensed table-bordered table-striped">
  <tr>
    <?php
      $primary_key = '';
      for ($i = 0; $i < $numFields; $i++) {
        $flags = mysqli_fetch_field_direct($inq, $i)->flags;
        $primary_key = mysqli_fetch_field_direct($inq, $i)->name;
        echo '<th>'.analyze_header($sql, $primary_key, is_primary($inq, $i), $tableName).'</th>';
      }
    <?php
  </tr>
  <?php
  while ($row = $db->__($inq)) {
    <?php
    $i = 0;
    foreach ($row as $key => $value) {
      echo '<td><pre>'.analyze($sql, $value, $key, is_primary($inq, $i), $tableName).'</pre></td>';
      $i++;
    }
    <?php
  </tr>

```

Рис. 3.3. Створення таблиці на основі повернутих даних

Ця таблиця наповнюється даними які було отримано в результаті get виклику. Записи у таблиці створюються у вигляді посилань, з відповідною для бази даних іконкою та іншими характеристиками.

Створення цих елементів виконуються за допомогою викликаних методів (рис. 3.4.)

```

function analyze($sql, $value, $key, $primary, $table) {
  global $config;
  if (preg_match('#^show#iu', trim($sql))) {
    if (preg_match('#databases#iu', trim($sql))) {
      return '<a href="dbname='.$value.'"><i class="icon icon_db"></i> '.$value.'</a>';
    } elseif (preg_match('#tables#iu', trim($sql))) {
      return '<a class="col-sm-4" href="?table='.$value.'&sql=.__('select * from `'.$value.'`')>';
    } elseif (preg_match('#CREATE[\s\t\n]+TABLE#iu', trim($sql))) {
      return htmlspecialchars(trim($value));
    }
    return $value;
  } elseif (preg_match('#^select#iu', trim($sql))) {
    return ($table && $primary ? '<input type="checkbox" name="value" value=".'.$value.'"> <a style='
  }
  return $value;
}

// Getting header column data for tables
function analyze_header($sql, $value, $primary, $table) {
  if (is_select($sql)) {
    return ($primary && $table ? '<input onclick="toggleAll.call();" type="checkbox" name="header_v
  }
  return $value;
}

```

Рис. 3.4. Методи які надають повернутим даним відповідні візуальні характеристики.

Загалом на цій сторінці знаходяться такі елементи (рис.3.5.):

- Поле, що показує виконану команду
- Поле для вводу SQL Query
- Кнопка Exec. Після натискання виконується SQL Query
- Час, за який команда була виконана
- Кількість елементів, до яких було застосована виконана команда
- Список всіх доступних баз даних для даного користувача.
- Кнопка «Sign out». Після натискання завершується робоча сесія та виконується дія disconnect.

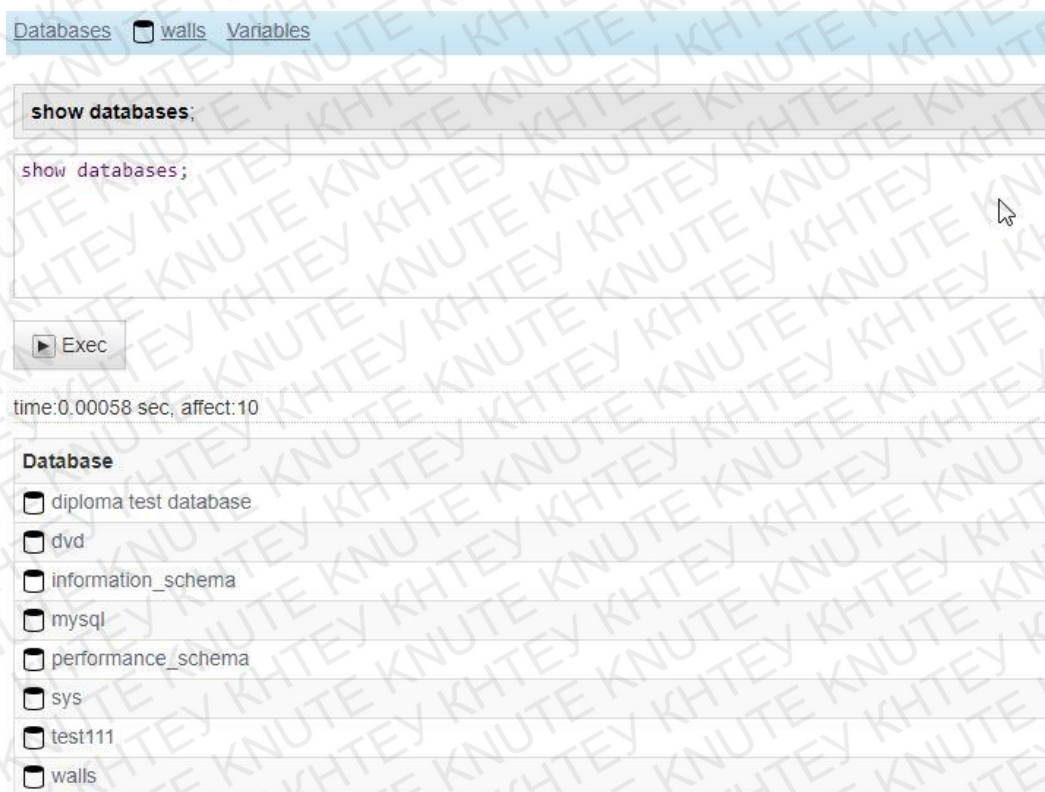


Рис. 3.5. Основна сторінка

						Аркуш
					<i>КНТЕУ 121 06-19.МР</i>	26
Зм.	Аркуш	№ докум	Підпис	Дата		

3.4. Вигляд бази даних

Після вибору необхідної бази даних. Виконується get виклик вибраної бази даних, дія index, створення візуальних елементів згідно з повернутими результатами та відображаються такі елементи (рис. 3.6.):

The screenshot shows a database management interface with the following elements:

- Navigation tabs: Databases, **walls**, Variables
- Command input: SHOW TABLES FROM `walls`
- Execution button: Exec
- Execution statistics: time:0.00177 sec, affect:12
- Results table: Tables_in_walls

Table Name	Structure	SQL
wp_links	structure	SQL
wp_options	structure	SQL
wp_postmeta	structure	SQL
wp_posts	structure	SQL
wp_term_relationships	structure	SQL
wp_term_taxonomy	structure	SQL
wp_termmeta	structure	SQL
wp_terms	structure	SQL
wp_usermeta	structure	SQL
wp_users	structure	SQL

Рис. 3.6. Вигляд бази даних

- Поле виконаної команди
- Час за який ця команда була виконана
- Кількість елементів, до яких було застосована виконана команда
- Список таблиць у цій базі даних
- Є можливість відкрити структуру таблиці та вигляд створення таблиці у форматі SQL Query

Після побудування елементів, відображається остання виконана команда, час за який вона була виконана та кількість елементів які до яких вона була застосована.

3.5. Вигляд структури таблиці

Після натискання кнопки «structure» необхідної таблиці відобразяться такі елементи: (рис. 3.7.)



The screenshot shows the MySQL command prompt with the command 'SHOW COLUMNS FROM wp_links;' and its output. The output is a table with columns: Field, Type, Null, Key, Default, and Extra. The 'link_id' field is a primary key with an auto-increment attribute.

Field	Type	Null	Key	Default	Extra
link_id	bigint unsigned	NO	PRI		auto_increment
link_url	varchar(255)	NO			
link_name	varchar(255)	NO			
link_image	varchar(255)	NO			
link_target	varchar(255)	NO			
link_description	varchar(255)	NO			
link_visible	varchar(20)	NO			
link_owner	bigint unsigned	NO	MUL	1	
link_rating	int	NO		0	
link_updated	datetime	NO		0000-00-00 00:00:00	
link_rel	varchar(255)	NO			
link_notes	mediumtext	NO			
link_rs	varchar(255)	NO			

Рис. 3.7. Вигляд структури таблиці

- Поле відповідної виконаної команди
- Час за який ця команда була виконана
- Кількість елементів, до яких було застосована виконана команда
- Список Атрибутів цієї таблиці, а конкретно: Назва поля, Тип, Чи може бути Null це поле, Які поля є ключовими, Дефолтні значення полей та додаткове поле.

3.6. Вигляд створення таблиці у форматі SQL Query

Після натискання кнопки «SQL» необхідної таблиці відобразяться такі елементи: (рис. 3.8.)

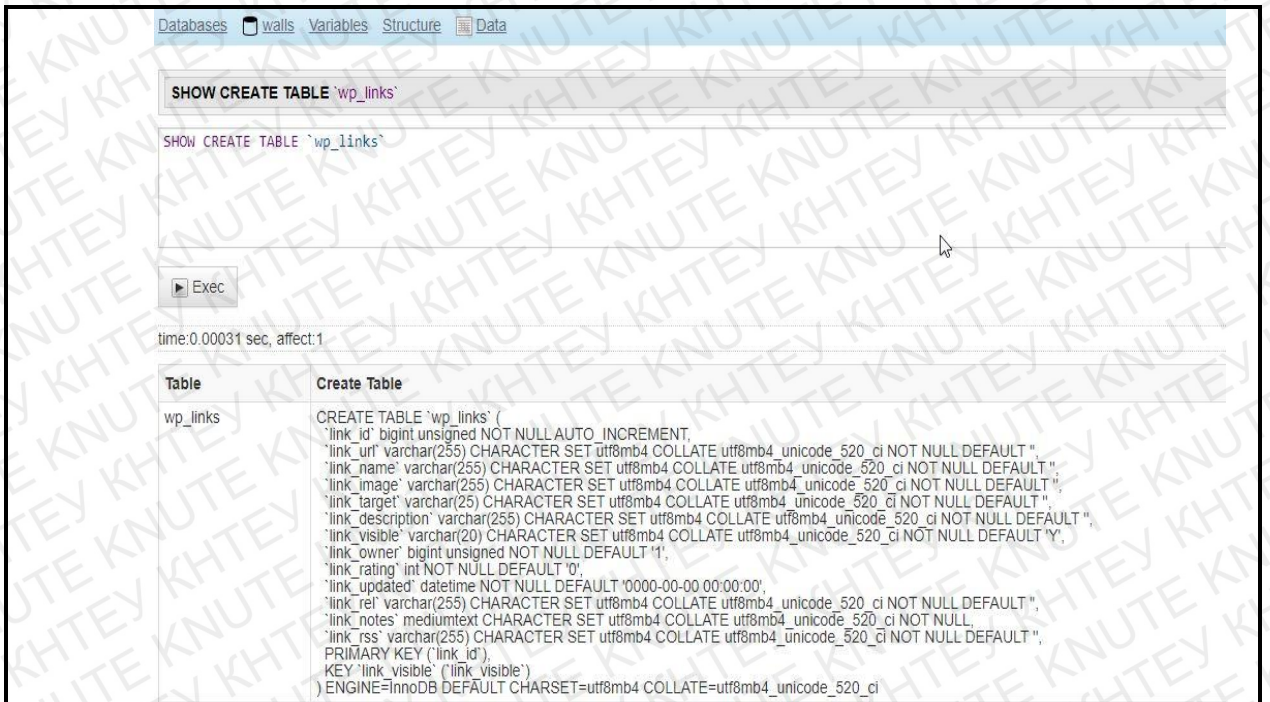


Рис. 3.8. Вигляд створення таблиці у форматі SQL Query

- Поле відповідної виконаної команди
- Час за який ця команда була виконана
- Кількість елементів, до яких було застосована виконана команда
- Назва таблиці
- Таблиця у форматі SQL Query

3.7. Вигляд таблиці

Після натискання на назву таблиці відобразяться такі елементи:
(рис. 3.9.)

						Аркуш
						29
Зм.	Аркуш	№ докум	Підпис	Дата		


```

select
from
  'wp_options'
limit
  20

```

select * from 'wp_options' limit 20

Exec

time:0.00037 sec. affect:20

<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	option_id	option_name	option_value	autoload
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1	siteurl	http://wails.local	yes
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2	home	http://wails.local	yes
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	3	blogname	Сейчасное Hello	yes
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	4	blogdescription	Матюшное потопом - Харьков	yes
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	5	users_can_register	0	yes
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	6	admin_email	nitoxes@gmail.com	yes
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	7	start_of_week	1	yes
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	8	use_balanceTags	0	yes
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	9	use_smilies	1	yes
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	10	require_name_email	1	yes
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	11	comments_notify	1	yes
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	12	posts_per_rss	10	yes
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	13	rss_use_excerpt	0	yes
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	14	mailserver_url	mail.example.com	yes
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	15	mailserver_login	login@example.com	yes
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	16	mailserver_pass	password	yes
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	17	mailserver_port	110	yes
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	18	default_category	1	yes
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	19	default_comment_status	open	yes
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	20	default_ping_status	open	yes

+

Рис. 3.9. Вигляд таблиці

- Поле відповідної виконаної команди
- Час за який ця команда була виконана
- Кількість елементів, до яких було застосована виконана команда
- Дані таблиці відповідно до колонок
- Можливість вибирати(чекбокси навпроти запису), редагувати або видаляти записи
- Кнопки додавання та видалення запису

3.8. Додавання, видалення та редагування записів

Після натискання на назву таблиці є можливість видалення та редагування записів. Навпроти кожного з записів є дві іконки «Редагувати» з іконкою олівця та «Видалити» з іконкою хрестик.

Клікнувши на «редагувати» виконається дія edit і відобразиться темплейт інтерфейсу форми редагування та такі елементи на ній: (рис. 3.10.)

					Аркуш
					30
Зм.	Аркуш	№ докум	Піопис	Дата	

The image shows a web form titled "Edit Row". It contains four input fields: "meta_id" with the value "1", "post_id" with the value "2", "meta_key" with the value "_wp_page_template", and "meta_value" with the value "default". At the bottom of the form are three buttons: "Save" (blue), "Save & Close" (green), and "Cancel" (red).

Рис. 3.10. Форма редагування

- Групбокс з елементами
- Назва колонок та значення відповідно за запису
- Кнопки «Save», «Save&Close», «Cancel»

Також є можливість додавання запису. Після натискання «Add» з іконкою плюс внизу таблиці з'явиться аналогічна форма, але вже з пустими полями.

Щоб видалити декілька записів одночасно потрібно відмітити чекбокси навпроти необхідних записів та натиснути «Remove» з іконкою зрестика внизу таблиці. Вионається метод видалення (рис.3.11.)

```
function deleteAllSelected() {
    const table = document.getElementById('result_table');
    let inputs = table.getElementsByTagName('input');
    selected = [];
    for (let i = 0; i < inputs.length; i++) {
        if (inputs[i].getAttribute('type') && inputs[i].getAttribute('type').toLowerCase() === 'checkbox' && inputs[i].getAttribute('name') !== 'header_value' && inputs[i].checked) {
            selected.push('values[' + inputs[i].value);
        }
    }
    if (selected.length) {
        if (confirm('Are you sure you want to delete ' + selected.length + ' records?')) {
            document.location = this.getAttribute('href') + '&' + selected.join('&');
        }
    } else {
        alert('No row in the table is selected');
    }
    return false;
}
```

Рис. 3.11. Метод видалення вибраних записів

Після цього відобразиться повідомлення у вікні браузера відповідно до цього ж методу (рис. 3.12.)

						Аркуш
						31
Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121 06-19.МР	

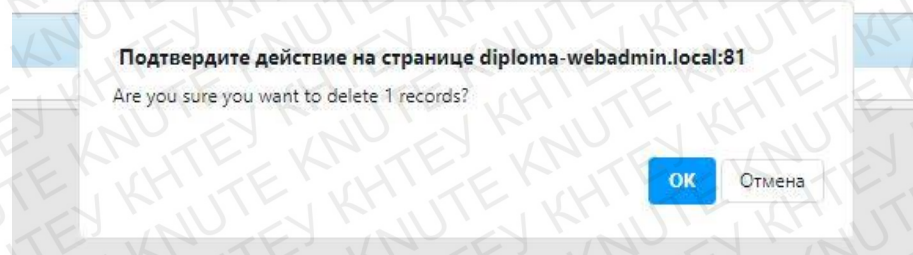


Рис. 3.12. Повідомлення

З текстом «Ви впевнені, що ви хочете видалити ці записи?» та можливістю підтвердити або відмінити свою дію.

3.9. Дерево елементів

З самого початку роботи з веб-додатком, коли статус «connected» до серверу баз даних та коли виконується умова, що користувач вибрав деяку базу даних, то зверху відображається назва бази даних (рис. 3.13.)

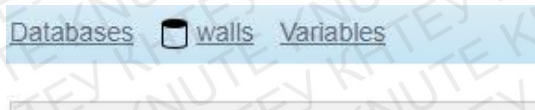


Рис. 3.13. Дерево елементів для вибраної бази даних

Далі, коли користувач вибирає деяку таблицю, то з'явиться можливість перейти на структуру або на дані цієї таблиці (рис. 3.15.) (рис. 3.14.)

```

<div class="header panel-success ">
  <?php if ($connected): ?>
    <a href="?sql=?php echo __('show databases;'); ?>">Databases</a>
  <?php endif; ?>
  <?php if ($database_selected): ?>
    <i class="icon icon_db"></i> <a href="?sql=?php echo __('show tables;'); ?>">?php echo $_SESSION['table']</a>
    <a href="?sql=?php echo __('show variables;'); ?>">Variables</a>
    <?php if( $_GET['table']): ?>
      <a href="?table=?php echo $_SESSION['table'];&sql=?php echo __('SHOW COLUMNS FROM `'.$db->
      <i class="icon icon_table"></i> <a href="?table=?php echo $_SESSION['table']; ?>">Data</a>
    <?php endif; ?>
  <?php endif; ?>
  <?php if ($connected): ?>
    <a style="float:right" href="?action=disconnect" title="Sign out"><i class="fa fa-sign-out-alt" a
  <?php endif; ?>
</div>
<div class="body">

```

Рис.3. 14. Умови відображення дерева елементів



Рис. 3.15. Дерево елементів для вибраної таблиці

					Аркуш
					32
Зм.	Аркуш	№ докум	Підпис	Дата	

3.10. Вигляд змінні бази даних

Після натискання кнопки «Variables» відображається список всіх змінних цієї бази даних (рис. 3.17.)

Databases walls Variables

show variables;

show variables;

time:0.00434 sec, affect:556

Variable_name	Value
activate_all_roles_on_login	OFF
admin_address	
admin_port	33062
auto_generate_certs	ON
auto_increment_increment	1
auto_increment_offset	1
autocommit	ON
automatic_sp_privileges	ON
avoid_temporal_upgrade	OFF
back_log	128
basedir	w:\modules\database\MySQL-8.0\
big_tables	OFF
bind_address	127.0.0.1
binlog_cache_size	32768
binlog_checksum	CRC32
binlog_direct_non_transactional_updates	OFF
binlog_encryption	OFF
binlog_error_action	ABORT_SERVER
binlog_expire_logs_seconds	2592000
binlog_format	ROW
binlog_group_commit_sync_delay	0
binlog_group_commit_sync_no_delay_count	0
binlog_gtid_simple_recovery	ON
binlog_max_flush_queue_time	0

Рис. 3.17. Вигляд змінних бази даних

3.11. Висновок до розділу 3

Отже, було розроблено та продемонстровано роботу засобу для адміністрування бази даних через веб-інтерфейс згідно з технічним завданням та на основі спроектованої архітектури програмного продукту було написано програмний код логіки серверної частини веб-додатку з використанням мови програмування PHP та клієнтської частини з використанням JavaScript.

За допомогою мови розмітки гіпертексту HTML, мови описання зовнішнього виду документа CSS була реалізована візуальна частина веб-додатку.

Поточний веб-додаток містить в собі всі необхідні компоненти, для того щоб виконувати різноманітні дії над базами даних. Було проведено функціональне та інтеграційне тестування. Виходячи з результатів тестування можна зробити висновок, що веб-додаток відповідає ТЗ, виконуючи наведені функції.

						Аркуш
					<i>КНТЕУ 121 06-19.МР</i>	34
Зм.	Аркуш	№ докум	Підпис	Дата		

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

Отже, було розглянуто характеристики веб-додатку та інтерфейсів. Був проведений аналіз предметної області, відзначені недоліки та переваги мов програмування, в результаті чого були виділені основні вимоги до розроблюваного веб-додатку для адміністрування баз даних.

Для створення ПЗ було вибрано JavaScript для опису дій які будуть виконуватися на клієнтській частині веб-додатку та використання мови програмування PHP для опису серверної частини веб-додатку.

Візуальний нарис веб-додатку був розроблений за допомогою веб-ресурсу WireFrame.

Написання веб-додатку було здійснено за допомогою середовища розробки PhpStorm. Для реалізації поставленої мети була розгорнута та налаштована серверна платформа Open Server.

Виходячи з результатів програми методики випробувань, можна зробити висновок, що веб-додаток відповідає ТЗ, виконуючи наведені функції. Даний веб-додаток не є кінцевим продуктом. Використання повного функціоналу стане доступним у наступній версії веб-додатку.

Зм.	Аркуш	№ докум.	Підпис	Дата				
Зав. каф.		Криворучко О.В.		30.10.20	Розробка універсального засобу адміністрування баз даних через web-інтерфейс	Стадія	Аркуш	Аркушів
Керівник		Криворучко О.В.		30.10.20		ВП	35	39
Гарант		Криворучко О.В.		30.10.20		Факультет інформаційних технологій 2м курс, 6 група		
Розробив		Ткешелашвілі Д.Л.		30.10.20				
					Висновки та пропозиції			

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Е. Фріман, Head First HTML with CSS&XHTML, 2010 — 656с.
2. Е. Тігтел, HTML, XHTML & CSS For Dummies, 7th Edition, 2011. — 400 с.
3. П. Лаббертс, Pro HTML5 Programming: Powerful APIs for Richer Internet Application Development, 2011. — 272 с.
4. Д. Ленгсторф. Pro PHP and jQuery, 2010. — 352 с.
5. М. Зандстра PHP Objects, Patterns and Practice, Third Edition, 2010. — 560 с.
6. Д. Роббінс – HTML5, CSS3 і JavaScript, 2014. – 528 с.

Інтернеті ресурси

7. PHP Tutorial [Електронний ресурс]. - Режим доступу:
<https://www.w3schools.com/php/default.asp>
8. Документація PhpMyAdmin [Електронний ресурс]. - Режим доступу:
<https://www.phpmyadmin.net>
9. Документація Open Server [Електронний ресурс]. - Режим доступу:
<https://ospanel.io/docs/>
10. Проектування веб-додатків [Електронний ресурс]. - Режим доступу:
https://stud.com.ua/97678/informatika/proektuvannya_dodatki
11. Веб-додаток [Електронний ресурс]. - Режим доступу:
<https://uk.wikipedia.org/wiki/Вебзастосунок>
12. Документація MySQL [Електронний ресурс]. - Режим доступу:
<https://dev.mysql.com/doc/>

					<i>КНТЕУ 121 06-19.МР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.		Криворучко О.В.		24.01.20	Розробка універсального засобу адміністрування баз даних через web-інтерфейс <i>Список використаних джерел</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Керівник		Криворучко О.В.		24.01.20		<i>СВД</i>	36	37
Гарант		Криворучко О.В.		24.01.20		Факультет інформаційних технологій 2м курс, 6 група		
Розробив		Ткешелашвілі Д.Л.		24.01.20				

13. Моделі життєвого циклу, принципи і методології розробки програмного забезпечення [Електронний ресурс]. - Режим доступу: <https://evergreens.com.ua/ua/articles/software-development-methodologies.html>
14. Прикладний програмний інтерфейс (API) [Електронний ресурс]. - Режим доступу: https://uk.wikipedia.org/wiki/Прикладний_програмний_інтерфейс
15. Веб-інтерфейс [Електронний ресурс]. - Режим доступу: <https://uk.wikipedia.org/wiki/Вебінтерфейс>

					<i>КНТЕУ 121 06-19.МР</i>	Аркуш
						37
Зм.	Аркуш	№ докум	Підпис	Дата		

ДОДАТОКИ

Лістинг коду index.php

Оновної сторінки на якій перемикаються темплейти інтерфейсу:

```
<?php
    session_start();
    include 'includes/config.php';
    include 'includes/connector.php';
    include 'includes/functions.php';
    include 'includes/SqlFormatter.php';
    include 'includes/logic.php';

    if($_POST) {
        $_POST = stripSlashesAll($_POST);
    }
    if($_GET) {
        $_GET = stripSlashesAll($_GET);
    }
    if($_COOKIE) {
        $_COOKIE = stripSlashesAll($_COOKIE);
        list($data['host'], $data['username'], $data['password'], $data['charset']) =
        _decode($_COOKIE[$config['cookie']]);
    }
    if($_REQUEST) {
        $_REQUEST = stripSlashesAll($_REQUEST);
    }
?>
<!DOCTYPE html>
<html lang="en-US">
<head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8"/>
    <title>MySQL Admin</title>
    <meta name="keywords" content="">
    <meta name="description" content="">
    <link rel="shortcut icon" type="image/x-icon"
href="skin/images/favicon.ico">
    <link rel="stylesheet" type="text/css"
href="skin/css/bootstrap/bootstrap.min.css">
    <link rel="stylesheet" type="text/css" href="skin/css/bootstrap/bootstrap-
theme.min.css">
    <link rel="stylesheet" type="text/css" href="skin/css/codemirror.css">
    <link rel="stylesheet" type="text/css" href="skin/css/style.css">
```



```

<link rel="stylesheet" type="text/css"
href="skin/css/fontawesome/all.css">
</head>
<body>
<div class="wrapper">
<div class="header panel-success ">
<?php if($connected): ?>
<a href="?sql=<?php echo __('show databases;'); ?>">Databases</a>
<?php endif; ?>
<?php if($database_selected): ?>
<i class="icon icon_db"></i><a href="?sql=<?php echo __('show
tables;'); ?>"><?php echo $_SESSION['dbname']?></a>
<a href="?sql=<?php echo __('show variables;'); ?>">Variables</a>
<?php if( $_GET['table']): ?>
<a href="?table=<?php echo $_SESSION['table']?>&sql=<?php
echo __('SHOW COLUMNS FROM `'.$db-
>_($_SESSION['table']).`'); ?>">Structure</a>
<i class="icon icon_table"></i><a href="?table=<?php echo
$_SESSION['table']?>">Data</a>
<?php endif; ?>
<?php endif; ?>
<?php if($connected):?>
<a style="float:right" href="?action=disconnect" title="Sign out"><i
class="fa fa-sign-out-alt" aria-hidden="true"></i></a>
<?php endif; ?>
</div>
<div class="body">
<?php switch ($action) { case 'edit' : ?>
<?php include 'app/templates/edit.php'?>
<?php break; case 'add' : ?>
<?php include 'app/templates/add.php'?>
<?php break; case 'login': ?>
<?php include 'app/templates/login.php'?>
<?php break; case 'disconnect': ?>
<?php include 'app/templates/login.php'?>
<?php break; case 'index' : ?>
<?php include 'app/templates/index.php'?>
<?php break; } ?>
</div>
</div>
<footer>
<div>
Author: <strong>David Tkeshelashvili</strong>

```

```

    Email:<a
href="mailto:"><strong>david.tkesh@gmail.com</strong></a>
    </div>
</footer>
<script type="text/javascript" src="skin/js/codemirror.js"></script>
<script type="text/javascript" src="scripts/file.js"></script>
</body>
</html>

```

Лістинг коду темплейту інтерфейсу add.php

```

<div class="edit panel panel-info">
  <div class="panel-heading">Add Record</div>
  <div class="panel-body">
    <?php if ($data['error']) { ?>
      <div class="alert alert-danger"><?php echo $data['error']; ?></div>
    <?php } ?>
    <form class="form-horizontal" role="form" method="post"
onsubmit="return confirm('Are you shure?')">
      <input type="hidden" name="action" value="save">
      <input type="hidden" name="table" value="<?php echo $inq-
>fetch_field_direct(1)->table; ?>">
      <?php
      $row = $db->__($inq);
      $i = 0;
      $primary_found = false;

      foreach ($row as $key => $value) { ?>
        <div class="form-group">
          <label for="key_<?php echo $key?>" class="col-sm-3 control-
label"><?php echo $key; ?></label>
          <div class="col-sm-9">
            <?php
            echo get_input($inq, $inq->fetch_field_direct(1)->table,
$inq->fetch_field_direct($i)->type, $key, ", $i);
            if (is_primary($inq, $i) && !$primary_found) {
              $primary_found = true;
            } ?>
            <input type="hidden" name="primary_key" value="<?php
echo htmlspecialchars($key); ?>">
            <?php } ?>
          </div>
        </div>
      } ?>
    </form>
  </div>
</div>

```

```

    </div>
    <?php
    $i++;
    }?>
    <div class="form-group">
        <div class="col-sm-offset-3 col-sm-10">
            <button type="submit" name="save" class="btn btn-
primary">Save</button>
            <button type="submit" name="close" class="btn btn-
success">Save & Close</button>
            <button type="button"
onclick="document.location='?table=<?php echo __($_GET['table']); ?>';"
class="btn btn-danger">Cancel</button>
        </div>
    </div>
</form>
</div>
</div>

```

2. Edit.php

```

<div class="edit panel panel-info">
    <div class="panel-heading">Edit Row</div>
    <div class="panel-body">
        <?php if ($data['error']) { ?>
            <div class="alert alert-danger"><?php echo $data['error']; ?></div>
        <?php } ?>
        <form class="form-horizontal" role="form" method="post" onsubmit="return
confirm('Are you shure?')">
            <input type="hidden" name="action" value="save">
            <input type="hidden" name="table" value="<?php echo $tableName =
$inq->fetch_field_direct(1)->table; ?>">
            <?php
            $row = $db->__($inq);
            $i = 0;
            $primary_found = false;

            foreach ($row as $key => $value) { ?>
                <div class="form-group">
                    <label for="key_<?php echo $key ?>" class="col-sm-3 control-
label"><?php echo $key; ?></label>
                    <div class="col-sm-9">

```



```

<?php
    echo get_input($inq, $inq->fetch_field_direct(1)->table, $inq-
->fetch_field_direct($i)->type, $key, $value, $i);

    if (is_primary($inq, $i) && !$primary_found) {
        $primary_found = true;
    }?>
    <input type="hidden" name="primary_key" value="<?php echo
htmlspecialchars($key); ?>">
    <input type="hidden" name="primary_value" value="<?php echo
htmlspecialchars($value); ?>"><?php }
    ?>
</div>
</div>
<?php
    $i++;
} ?>
<div class="form-group">
    <div class="col-sm-offset-3 col-sm-10">
        <button type="submit" name="save" class="btn btn-
primary">Save</button>
        <button type="submit" name="close" class="btn btn-success">Save &
Close</button>
        <button type="button" onclick="document.location='?table=<?php
echo __($ _GET['table']);?>';" class="btn btn-danger">Cancel</button>
    </div>
</div>
</form>
</div>
</div>

```

3. Index.php

```

<div class="workbox">
    <form role="form" method="post">
        <input type="hidden" name="action" value="index">
        <pre class="already_executed"><?php echo SqlFormatter::format(_trim($db-
>last())); ?></pre>
        <div class="form-group">
            <textarea id="sql" name="sql" placeholder="SQL..." class="form-control"
rows="5"><?php echo htmlspecialchars(_trim($sql)); ?></textarea>
        </div>
        <div class="form-group">
            <div>

```

```

        <button type="submit" class="btn btn-default"><i class="icon
        icon_play"></i> Exec</button>
    </div>
</div>
</form>
<?php if($data['error']): ?>
    <div class="alert alert-danger"><?php echo $data['error']; ?></div>
<?php else: ?>
    <div class="time_for_query">time:<?php echo round($endtime,5); ?> sec,
    affect:<?php echo $db->affect(); ?></div>
    <?php
        if ($inq and $inq !== true):
            $numFields = mysqli_num_fields($inq);
            $tableName = $_GET['table'] ? $_GET['table'] : '';
        ?>
        <table id="result_table" class="table table-hover table-condensed table-
        bordered table-striped">
            <tr>
                <?php
                    $primary_key = '';
                    for ($i = 0; $i < $numFields; $i++) {
                        $flags = mysqli_fetch_field_direct($inq, $i)->flags;
                        $primary_key = mysqli_fetch_field_direct($inq, $i)->name;
                        echo '<th>'.analyze_header($sql, $primary_key, is_primary($inq,
                    $i), $tableName).'</th>';
                    }
                ?>
            </tr>
            <?php
                while ($row = $db->__($inq)) {
                    ?>
                <tr>
                    <?php
                        $i = 0;
                        foreach ($row as $key => $value) {
                            echo '<td><pre>'.analyze($sql, $value, $key, is_primary($inq, $i),
                        $tableName).'</pre></td>';
                            $i++;
                        }
                    ?>
                </tr>
            <?php } ?>
        </table>

```

```

<?php if ($_GET['table']): ?>
    <div style="margin:5px 0px;">
        <?php if ($primary_key): ?>
            <a class="btn btn-default" onclick="return
deleteAllSelected.call(this)" href="?table=<?php echo $tableName;
?>&action=delete&key=<?php echo __($primary_key)?>"><i class="icon
icon_delete"></i></a>
        <?php endif; ?>
        <a class="btn btn-default" href="?table=<?php echo $tableName;
?>&action=add"><i class="icon icon_add"></i></a>
    </div>
<?php endif; ?>
<?php else: ?>
<?php if ($inq !== true): ?>
    <div class="alert alert-danger"><?php echo $db->error(); ?></div>
<?php else: ?>
    <div class="alert alert-success">Success</div>
<?php endif; ?>
<?php endif; ?>
<?php endif; ?>
</div>

```

4.login.php

```

<div class="login panel panel-info">
    <div class="panel-heading">DB Connection Settings</div>
    <div class="panel-body">
        <?php if ($data['error']): ?>
            <div class="alert alert-danger"><?php echo $data['error']; ?></div>
        <?php endif; ?>
        <form class="form-horizontal" role="form" method="post">
            <input type="hidden" name="action" value="connect">
            <div class="form-group">
                <label for="username" class="col-sm-4 control-
label">Username</label>
                <div class="col-sm-7">
                    <input type="text" class="form-control" name="username"
id="username" placeholder="DB user name" value="<?php echo
getLoginFormData($data)['username']; ?>" required>
                </div>
            </div>
            <div class="form-group">
                <label for="password" class="col-sm-4 control-label">Password</label>

```



```

        <div class="col-sm-7">
            <input type="password" class="form-control" name="password"
            id="password" placeholder="Password" value="<?php echo
            getLoginFormData($data)['password']; ?>">
        </div>
    </div>
    <div class="form-group">
        <label for="host" class="col-sm-4 control-label">Host</label>
        <div class="col-sm-7">
            <input type="text" class="form-control" id="host" name="host"
            placeholder="MySQL host" value="<?php echo getLoginFormData($data)['host'];
            ?>" required>
        </div>
    </div>
    <div class="form-group">
        <div class="col-sm-offset-4 col-sm-8">
            <div class="checkbox">
                <label>
                    <input name="remember" value="yes" type="checkbox">
Remember in cookies
                </label>
            </div>
        </div>
    </div>
    <div class="form-group">
        <div class="col-sm-offset-4 col-sm-8">
            <button type="submit" class="btn btn-primary">Try to
connect</button>
        </div>
    </div>
</form>
</div>
</div>

```

Лістинг коду config.php Загальні змінні додатку

```

<?php
$config = array(
    'cookie' => 'key_man',
    'max_str_len' => '100',
    'count_on_page' => '20',
    'ssid_deob' => 'hjhkjsre4',
    'charset' => 'utf8'
);

```

Лістинг коду connector.php Для роботи базою даних

```
<?php
class DB {
    private $connid = null;
    private $error = "";
    private $last_query = "";
    public $separator = ';';
    public $charset = false;

    // Gets error from sql query
    function error() {
        return $this->error;
    }

    // Gets the number of affected rows in a previous MySQL operation
    function affect() {
        return mysqli_affected_rows($this->connid);
    }

    // Gets the last sql query
    function last() {
        return $this->last_query;
    }

    function selectDB($dbName, $charset = 'utf8') {
        if ($this->charset)
            $charset = $this->charset;
        if (!$this->connid) {
            $this->error = 'Check data fore connection';
            return false;
        }
        if (mysqli_select_db($this->connid, $dbName)) {
            return $this->q("SET NAMES '" . $charset . "'") && $this->q("SET CHARACTER SET '" . $charset . "'") && $this->q("SET CHARACTER SET '" . $charset . "'") && $this->q("SET SESSION collation_connection = '{ $charset }_general_ci");
        } else {
            $this->error = mysqli_error($this->connid);
            return false;
        }
    }
}
```

```

function disconnect() {
    if ($this->connid)
        mysqli_close($this->connid);
}

function connect($host, $user, $password, $charset = false) {
    if ($charset)
        $this->charset = $charset;

    try {
        if (!empty($host) && !empty($user)) {
            mysqli_report(MYSQLI_REPORT_STRICT);

            // Host validation
            if (!$this->isValidHost($host)) {
                $this->error = 'Host name is not valid.';
                return false;
            }

            $mysqli = new mysqli($host, $user, $password);

            /* check connection */
            if ($mysqli->connect_errno) {
                printf("Connect failed: %s\n", $mysqli->connect_error);
                $mysqli->close();
            } else {
                $this->connid = $mysqli;
                return true;
            }
        }
    } catch (Exception $e) {
    }

    $this->error = 'No connect. Check your form data.';
    return false;
}

function q($sql) {
    $sqlQueries = explode($this->separator, $sql);
    $this->error = "";
    $inq = false;
    foreach($sqlQueries as $query){

```



```

        if (trim($query)) {
            $inq = mysqli_query($this->connid, $query);
            if (!$inq) {
                $this->error .= mysqli_error($this->connid)."\n";
            }
        }
    }
    $this->last_query = $sql;
    return $inq;
}

// Fetch a result row as an associative array
function __($inq) {
    return mysqli_fetch_assoc($inq);
}

function row($sql, $only = false) {
    $inq = $this->q($sql);
    if ($inq) {
        $row = $this->__($inq);
        return ($only === false) ? $row : isset($row[$only]) ?
$row[$only] : false;
    }else{
        return false;
    }
}

function rows($sql, $field = false, $only = false) {
    $inq = $this->q($sql);
    if($inq) {
        $items = array();
        while($row = $this->__($inq)){
            $item = ($only === false or lisset($row[$only])) ?$row :
$row[$only];
            if($field === false or !isset($row[$field]))
                $items[] = $item;
            else
                $items[$field] = $item;
        }
        return $items;
    }else{
        return false;
    }
}

```

// Escapes special characters in a string for use in an SQL statement, taking into account the current charset of the connection

```
function _($val) {  
    return mysqli_real_escape_string($this->connid, $val);  
}
```

```
private function arrayToSet($values) {  
    $ret = array();  
    if (is_array($values)) {  
        foreach ($values as $key => $value) {  
            $ret[] = is_int($key) ? $value : "`$key`='".  
>_($value)."'";  
        }  
        return implode(',', $ret);  
    } else {  
        return $values;  
    }  
}
```

```
private function condition($values) {  
    $ret = array();  
    if (is_array($values)) {  
        foreach ($values as $key => $value) {  
            $ret[] = is_int($key) ? $value : "`$key`='".  
>_($value)."'";  
        }  
        return implode(' and ', $ret);  
    } else {  
        return $values;  
    }  
}
```

```
function insert($stable, $array) {  
    $sql = 'insert into '.$stable.' set '.$this->arrayToSet($array);  
    $inq = $this->q($sql);  
    if ($inq)  
        return mysqli_insert_id($this->connid);  
    else  
        return false;  
}
```

```
function delete($stable, $condition) {  
    $sql = 'delete from '.$stable.' where '.$this->condition($condition);
```

```

        return $this->q($sql);
    }

    function update($table, $array, $condition) {
        $sql = 'update '.$table.' set '.$this->arrayToSet($array).' where '.$this->condition($condition);
        return $this->q($sql);
    }

    function exists($table, $conditions, $id='id', $order = false, $offset = 0) {
        $sql = 'select * from '.$table.' where '.$this->condition($conditions).'
        ' . ($order ? ' order by '.$order : '') . ($offset ? ' offset '.$offset : '');
        return $this->row($sql, $id);
    }

    // Validation for host in login form
    function isValidHost($host) {
        return $host === 'localhost' ||
        preg_match('/^\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}(:\d+)?$/ ', $host);
    }
}

```

Лістинг коду functions.php Основні виконувані методи

```

<?php
function ssd_deob($String, $Password='dgdfg234') {
    $String = $String.'';
    $Salt = 'GKJ76DL2dsgDP';
    $StrLen = strlen($String);
    $Seq = $Password;
    $Gamma = "";
    while (strlen($Gamma)<$StrLen) {
        $Seq = pack("H*", sha1($Gamma.$Seq.$Salt));
        $Gamma.=substr($Seq,0,8);
    }
    return $String^$Gamma;
}

function stripSlashesAll($array) {
    foreach ($array as $key => $val) {
        if (is_string($val)) {
            $array[$key] = stripslashes($val);
        }
    }
}

```



```

    } elseif (is_array($val)) {
        $array[$key] = stripSlashesAll($val);
    }
}
return $array;
}

function _trim($value) {
    return preg_replace(array('#^\[\n\r\t\s]+\#u', '#[\n\r\t\s]+$#u'), "", $value);
}

function getPHPCharset($value, $charset = "utf-8") {
    $sch = mb_detect_encoding($value);
    return $sch ? $sch : $charset;
}

function _encode($host, $user, $password, $charcode) {
    global $config;
    return
    base64_encode(ssd_deob($host.'&|&'. $user.'&|&'. $password.'&|&'. $charcode, $con
    fig['ssd_deob']));
}

function _decode($val) {
    global $config;
    return explode('&|&', ssd_deob(base64_decode($val), $config['ssd_deob']));
}

function __($sql) {
    return addslashes($sql);
}

function analyze($sql, $value, $key, $primary, $stable) {
    global $config;
    if (preg_match('#^show#iu', _trim($sql))) {
        if (preg_match('#databases#iu', _trim($sql))) {
            return '<a href="?dbname='.$value.'"><i class="icon
            icon_db"></i> '.$value.'</a>';
        } elseif (preg_match('#tables#iu', _trim($sql))) {
            return '<a class="col-sm-4"
            href="?table='.$value.'&sql=.__('select * from `'.$value.`')." "><i class="icon
            icon_table"></i> '.$value.'</a> <a class="col-sm-1"
            href="?table='.$value.'&sql=.__('SHOW COLUMNS FROM

```

```

`'.$value.'"')"'>structure</a> <a href="?table='.$value.'&sql='.__('SHOW
CREATE TABLE`'.$value.'"')"'>SQL</a>;
    } elseif (preg_match('#CREATE[\s\t\n]+TABLE#iu',_trim($sql))) {
        return htmlspecialchars(_trim($value));
    }
    return $value;
} elseif (preg_match('#^select#iu',_trim($sql))) {
    return ($table && $primary ? '<input type="checkbox" name="value"
value="`'.$value.'"> <a style="" onclick="return confirm('\Are you sure?\')"
href="?table='.$table.'&action=delete&key='.__($key).'&value='.__($value).'"><i
class="icon icon_delete"></i></a> <a
href="?table='.$table.'&action=edit&key='.$key.'&value='.$value.'"><i
class="icon icon_edit"></i></a>
':').htmlspecialchars(mb_substr($value,0,$config['max_str_len'],getPHPCharset($
alue)));
}
return $value;
}

// Getting header column data for tables
function analyze_header($sql, $value, $primary, $table) {
    if (is_select($sql)) {
        return ($primary && $table ? '<input onclick="toggleAll.call();"
type="checkbox" name="header_value" value="`'.__($value).'">': '').<a
href="?sql=.rawurlencode(add_to_sql($sql,'order',$value)).'">.htmlspecialchars($
value).</a>;
    }
    return $value;
}

function is_select($sql) {
    return preg_match('#^[ \t]?select#iu',_trim($sql));
}

function is_limited($sql) {
    return preg_match('#[\s\n\r\t]*limit[\s\n\r\t]+#ui',$sql);
}

function is_ordered($sql) {
    return
preg_match('#[\s\n\r\t]*order[\s\n\r\t]+by[\s\n\r\t]+([\s\n\r\t]+)([\s\n\r\t]+|$)#ui',$s
ql,$list)?preg_replace('#[\'"\s\n\t`#',"',$list[1]):false;
}

```

```

function is_enum($inq, $i) {
    return strpos(mysql_ii_field_flags($inq, $i), 'enum') !== false;
}

function get_enum_values($table, $field) {
    global $db;
    $type = $db->row("SHOW COLUMNS FROM `{$table}` WHERE Field =
'{$field}'", 'Type');
    preg_match('/^enum\(((.*)\)$/', $type, $matches);
    foreach (explode(',', $matches[1]) as $value) {
        $enum[] = trim($value, "'");
    }
    return $enum;
}

// Checking if is a primary key
function is_primary($inq, $i) {
    $flags = mysql_ii_fetch_field_direct($inq, $i)->flags;
    return $flags & MYSQLI_PRI_KEY_FLAG;
}

function add_to_sql($sql, $name, $value) {
    switch ($name) {
        case 'limit':
            if (is_limited($sql)) {
                $sql = preg_replace('#[\s\t\n\r]*limit[\s\t\n\r]+[0-
9\s]+#ui', 'limit `'.$value.'`', $sql);
            } else {
                $sql .= ' limit `'.$value.'`;';
            }
            break;
        case 'order':
            if ($orderBy = is_ordered($sql)) {
                $sql =
preg_replace('#[\s\n\r\t]*order[\s\n\r\t]+by[\s\n\r\t]+[^\s\n\r\t;]+[\s\n\r\t]*#ui', 'order
by `'.$value.'`', $sql);
                if ($value == $orderBy) {
                    if
(preg_match('#[\s\n\r\t`]+order[\s\n\r\t`]+by[\s\n\r\t`]+`'.$value.'`[\s\n\r\t`]+(desc|asc)
c)#ui', $sql, $list)) {
                        $sql = preg_replace('#(order by
`'.$value.'`[\s\n\r\t`]+)(desc|asc)#ui', '$1'.(mb_strtolower($list[1])=='desc'? 'asc': 'desc
'), $sql);
                    } else {

```



```

        $sql = preg_replace('#( order by `'.$value.'`[\s\n\r\t`]+)#ui','$1desc
        ', $sql);
    }
    } else {
        if (is_limited($sql)) {
            $sql = preg_replace('#[\s\n\r\t]*limit[\s\n\r\t]+#ui', '
            order by `'.$value.'` asc limit ', $sql);
        } else {
            $sql .= 'order by `'.$value.'`;';
        }
    }
    break;
}
return $sql;
}

```

```

function base64_encode_image($fileName, $fileType) {
    if ($fileName) {
        $imgBinary = fread(fopen($fileName, "r"), filesize($fileName));
        return 'data:image/'. $fileType . ';base64,' . base64_encode($imgBinary);
    }
}

```

```

function get_input($inq, $stable, $stype, $key, $value, $i) {
    switch ($stype) {
        case 'real':
        case 'int':
            return '<input type="number" class="form-control"
            name="key[.'.__($key).']" id="key_.'.__($key).'." placeholder=".'.__($key).'."
            value=".'.__($value).'.">';
            break;
        case 'string':
            if (is_enum($inq, $i)) {
                $enum_variants = get_enum_values($stable, $key);
                $sout = '<select class="form-control"
                name="key[.'.__($key).']" id="key_.'.__($key).'.">';
                foreach ($enum_variants as $name) {
                    $sout .= '<option .'__($value==$name?'selected':').'.'
                    value=".'__($name).'.">.'__($name).'</option>';
                }
                $sout .= '</select>';
            } else {

```

```

        $out=<input type="text" class="form-control"
name="key['.__( $key).']" id="key_'.__( $key).'" placeholder="'.__( $key).'"
value="'.__( $value).'">;
    }
    return $out;
    break;
    case 'timestamp':
    case 'year':
    case 'date':
    case 'time':
    case 'datetime':
        return '<input type="text" class="form-control"
name="key['.$key.']" id="key_'. $key.'" placeholder="'. $key.'"
value="'.addslashes($value).'">;
        break;
    default:
        return '<textarea class="form-control" name="key['.$key.']"
id="key_'. $key.'" placeholder="'. $key.'">'.htmlspecialchars($value).</textarea>';
        break;
    }
}

// Getting Primary key field
function get_primary_field($table) {
    global $db;
    return $db->row("SHOW KEYS FROM `". $db->_( $table). "` WHERE
Key_name = 'PRIMARY'", 'Column_name');
}

function ekran($value) {
    global $db;
    return "\".$db->_( $value).\"";
}

function getLoginFormData($data) {
    $loginData = [];
    $loginData['host'] = htmlspecialchars($_REQUEST['host'] ?
$_REQUEST['host'] : $data['host'] ? $data['host'] : "");
    $loginData['username'] = htmlspecialchars($_REQUEST['username'] ?
$_REQUEST['username'] : $data['username'] ? $data['username'] : "");
    $loginData['password'] = htmlspecialchars($_REQUEST['password'] ?
$_REQUEST['password'] : $data['password'] ? $data['password'] : "");

    return $loginData;
}

```

Лістинг коду `logic.php` Основної логіки веб-додатку

```
<?php
// Create child of DB class with getting properties and methods
$db = new DB();

// Set action from request. By default - Index action.
$action = (isset($_REQUEST['action']) and
in_array($_REQUEST['action'],array('index', 'connect', 'disconnect', 'login', 'add',
'edit', 'save', 'delete'))) ? $_REQUEST['action'] : 'index';
$data = array('error'=>'', 'connect'=>);

// Merge data from session, request and config in one array $_REQUEST
$_REQUEST = array_merge($_SESSION, $_REQUEST, $config);

$sql = "";
$database_selected = false;
$connected = false;

// Set data to login form from cookie
if (isset($_COOKIE[$config['cookie']])) {
    list($_REQUEST['host'], $_REQUEST['username'], $_REQUEST['password'],
$_REQUEST['charset']) = _decode($_COOKIE[$config['cookie']]);
}

// Connect to DB if not empty request data
if (!empty($_REQUEST['host']) && !empty($_REQUEST['username']) &&
isset($_REQUEST['password'])) {
    try {
        try {
            if ($action !== 'disconnect') {
                if ($db->connect($_REQUEST['host'], $_REQUEST['username'],
$_REQUEST['password'], $_REQUEST['charset']))
                    $connected = true;
            }
        } catch (mysqli_sql_exception $e) {
            die("Unfortunately, the details you entered for connection are incorrect!");
        }
    } catch (Exception $e) {
        $data['error'] = $db->error();
    }
}
```



```

switch ($action) {
    case 'save':
        if ($connected) {
            if ($db->selectDB($_REQUEST['dbname'])) {
                if (isset($_POST['key']) and is_array($_POST['key']) and
count($_POST['key'])) {
                    if (isset($_REQUEST['table']) and
isset($_REQUEST['primary_key']) and isset($_REQUEST['primary_value'])) {

                        $inq = $db-
>update($_GET['table'],$_POST['key'],'.'.$db-
>_($_REQUEST['primary_key'])."='".$db->_($_REQUEST['primary_value'])."');
                        if(!$inq){
                            $data['error'] = $db->error();
                            $action = 'edit';
                            $inq = $db->q($q = 'select * from
`.'.$db->_($_GET['table']).` where `.'.$db-
>_($_REQUEST['primary_key'])."='".$db->_($_REQUEST['primary_value'])."');
                        }
                    } elseif (isset($_REQUEST['table'])) {
                        $inq = $db-
>insert($_GET['table'],$_POST['key']);
                        if ($inq){
                            $_REQUEST['primary_key'] =
get_primary_field($_GET['table']);
                            if ($_REQUEST['primary_key'])
                                $_REQUEST['primary_value'] =
$inq;
                        } else {
                            $data['error'] = $db->error();
                            $action = 'add';
                            $inq = $db->q($q = 'select * from
`.'.$db->_($_GET['table']).` limit 1');
                        }
                    }
                }
            }
            if ($inq && !$data['error']) {
                if (isset($_REQUEST['close']) or
!isset($_REQUEST['save'])) {
                    $sql = 'select * FROM `.'.$db-
>_($_REQUEST['table'])."');

                    $action = 'index';
                    if (is_select($sql) and !is_limited($sql))

```

```

        $sql =
add_to_sql($sql,'limit',$config['count_on_page']);

        $time = microtime(true);
        $inq = $db->q($sql);
        $endtime = microtime(true) - $time;

        if ($inq and is_select($sql) and $tableName
= $inq->fetch_field_direct(1)->table)
            $_GET['table'] = $_SESSION['table']
= $_REQUEST['table'] = $tableName;
        } else {
            $action = 'edit';
            $inq = $db->q($q = 'select * from `'.$db-
>_($_GET['table']).` where `'.$db->_($_REQUEST['primary_key']).`=\'".$db-
>_($_REQUEST['primary_value']).\'");
        }
    }
    $database_selected = true;
}
} else {
    $data['error'] = $db->error();
    $action = 'login';
}
break;
case 'delete':
    if ($connected) {
        if ($db->selectDB($_REQUEST['dbname'])) {
            if (isset($_GET['table']) and isset($_GET['key']) and
(!empty($_GET['value']) or (!empty($_GET['values']) and
is_array($_GET['values']) and count($_GET['values']))) {
                if (!empty($_GET['value'])) {
                    $inq = $db->delete($_GET['table'],''.$db-
>_($_GET['key']).`=\'".$db->_($_GET['value']).\'");
                } else {
                    $inq = $db->delete($_GET['table'],''.$db-
>_($_GET['key']).` in ('.implode(',',array_map('ekran',$_GET['values'])).')');
                }
                if ($inq) {
                    $action = 'index';
                    $sql = 'select * FROM `'.$db-
>_($_REQUEST['table']).`';
                    $database_selected = true;
                    $action = 'index';

```

```

        if( is_select($sql) and !is_limited($sql) )
            $sql=
add_to_sql($sql,'limit',$config['count_on_page']);
        } else {
            $data['error'] = $db->error();
        }
    }
    $database_selected = true;
}
break;
case 'add':
    if ($connected) {
        if ($db->selectDB($_REQUEST['dbname'])) {
            if (isset($_GET['table'])) {
                $sql = $db->q($q = 'select * from `'.$db->
                >_($_GET['table']).` limit 1');
            }
            $database_selected = true;
        }
    } else {
        $data['error'] = $db->error();
        $action = 'login';
    }
break;
case 'edit':
    if ($connected) {
        if ($db->selectDB($_REQUEST['dbname'])) {
            if (isset($_GET['table']) and isset($_GET['key']) and
isset($_GET['value'])) {
                $sql = $db->q($q = 'select * from `'.$db->
                >_($_GET['table']).` where `'.$db->_($_GET['key']).`= \"'.$db->
                >_($_GET['value']).\"');
            }
            $database_selected = true;
        }
    } else {
        $data['error'] = $db->error();
        $action = 'login';
    }
break;
case 'index':

```



```

if ($connected) {
    if (isset($_REQUEST['dbname'])) {
        if ($db->selectDB($_REQUEST['dbname'])) {
            if (isset($_GET['table'])) {
                $_SESSION['table'] = $_REQUEST['table'];
                $sql = $_REQUEST['sql'];
                $_REQUEST['sql']:'select * FROM `'.$db->_($_REQUEST['table'])."`';
            } else {
                $sql = $_REQUEST['sql']?$_REQUEST['sql']:'SHOW TABLES
FROM `'.$db->_($_REQUEST['dbname'])."`';
            }

            $_SESSION['dbname'] = $_REQUEST['dbname'];

            if (is_select($sql) and !is_limited($sql))
                $sql =
add_to_sql($sql,'limit',$config['count_on_page']);
            $database_selected = true;
        } else {
            $data['error'] = $db->error();
            $sql = $_REQUEST['sql'];
            $_REQUEST['sql']:'SHOW DATABASES;';
        }
    } else {
        $sql = $_REQUEST['sql']?$_REQUEST['sql']:'SHOW
DATABASES;';
    }

    $action = 'index';

    $time = microtime(true);
    $inq = $db->q($sql);
    $sendtime = microtime(true) - $time;

    if ($inq and is_select($sql) and $tableName = $inq->
fetch_field_direct(1)->table) {
        $_GET['table'] = $_SESSION['table'] = $_REQUEST['table'] =
$tableName;
    }
    } else {
        $data['error'] = $db->error();
        $action = 'login';
    }
}
break;

```

```

    case 'connect':
        $db->disconnect();
        mysqli_report(MYSQLI_REPORT_ERROR|MYSQLI_REPORT_STRICT);
        // Set MySQLi to throw exceptions
        try {
            if (!empty($_REQUEST['host']) &&
                !empty($_REQUEST['username'])) {
                if ($db->connect($_REQUEST['host'], $_REQUEST['username'],
                    $_REQUEST['password'], $_REQUEST['charset'])) {
                    if( !empty($_REQUEST['remember'])) {
                        setcookie($config['cookie'],_encode($q =
                            $_REQUEST['host'],$_REQUEST['username'],$_REQUEST['password'],$_REQU
                            EST['charset']));
                    }else{
                        $_SESSION['host']=$_REQUEST['host'];
                        $_SESSION['username']=$_REQUEST['username'];
                        $_SESSION['password']=$_REQUEST['password'];
                        $_SESSION['charset']=$_REQUEST['charset'];
                    }
                    $action = 'index';
                    header('location:?action=index');
                    exit();
                } else {
                    $_SESSION['host']=$_SESSION['username']=
                    $_SESSION['password']="";
                    setcookie($config['cookie'],");
                    $data['error']=$db->error();
                    $action = 'login';
                }
            } else {
                if (empty($_REQUEST['host'])) {
                    $data['error'] = 'Host';
                }
                $_SESSION['host']=$_SESSION['username']=
                $_SESSION['password']="";
                setcookie($config['cookie'],");
                $data['error']=$db->error();
                $action = 'login';
            }
        } catch (mysqli_sql_exception $e) {
            $_SESSION['host']=$_SESSION['username']=$_SESSION['password']
            = "";
            setcookie($config['cookie'],");
            $data['error']=$db->error();

```

```
    $action = 'login';
  }
  case 'disconnect':
    $db->disconnect();
    $_REQUEST['host']=$_REQUEST['username']=$_REQUEST['password']
    =$_REQUEST['table']="";
    $_SESSION['host']=$_SESSION['username']=$_SESSION['password']=
    $_SESSION['table']="";
    $action = 'login';
    break;
  }
}
```