

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ВІННИЦЬКИЙ НАЦІОНАЛЬНИЙ АГРАРНИЙ УНІВЕРСИТЕТ

Факультет економіки та підприємництва

ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ

В.о. завідувача кафедри комп'ютерних наук та економічної  
кібернетики д.е.н., професор \_\_\_\_\_ С.В. Коляденко

«\_\_» \_\_\_\_\_ 2023 р.

Лукомський Антон Ігорович

РОЗРОБКА WEB-САЙТУ ТОРГОВОГО ПІДПРИЄМСТВА З  
ПРОДАЖУ БІОГАЗОВИХ УСТАНОВОК

Робота на здобуття першого (бакалаврського)  
освітнього ступеню

Науковий керівник  
к.т.н., доцент кафедри  
комп'ютерних наук та  
економічної кібернетики  
Красиленко В.Г.

ВІННИЦЯ – 2023

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ВІННИЦЬКИЙ НАЦІОНАЛЬНИЙ АГРАРНИЙ УНІВЕРСИТЕТ  
ФАКУЛЬТЕТ ЕКОНОМІКИ ТА ПІДПРИЄМНИЦТВА  
Кафедра комп'ютерних наук та економічної кібернетики

ЗАТВЕРДЖУЮ  
В.о.завідувача кафедри

« \_\_\_\_\_ » « \_\_\_\_\_ » 202\_ р.

### ЗАВДАННЯ НА ДИПЛОМНУ РОБОТУ

Лукомському Антону Ігоровичу

(прізвище, ім'я, по батькові)

1. Тема роботи Розробка WEB-сайту торгового підприємства з продажу біогазових установок

затверджена наказом по університету від «28» грудня 2022 року № 207м

Керівник Красиленко Володимир Григорович, к.т.н., доцент

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

2. Строк подання студентом роботи 08.05.23 р.

3. Вихідні дані до роботи визначення структурної схеми сайту; керівництво програміста (включаючи налаштування сервера; CMS WordPress); керівництво користувача; визначення оптимізації та налаштування сайту...

4. Календарний план виконання кваліфікаційної роботи

Структура роботи	Термін підготовки
<i>ВСТУП</i>	<i>27.05.23</i>
<i>РОЗДІЛ 1 ТЕХНОЛОГІЇ РОЗРОБКИ САЙТІВ</i>	<i>27.05.23</i>
<i>РОЗДІЛ 2 РОЗРОБКА БАЗИ ДАНИХ</i>	<i>28.02.23</i>
<i>РОЗДІЛ 3 РОЗРОБКА WEB САЙТУ ТОРГОВОГО ПІДПРИЄМСТВА З ПРОДАЖУ БІОГАЗОВИХ УСТАНОВОК</i>	<i>31.03.23</i>
<i>ВИСНОВКИ ТА ПРОПОЗИЦІЇ</i>	<i>14.04.23</i>
<i>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ</i>	<i>14.04.23</i>

5. Термін подання роботи на кафедру для попереднього захисту 03.05.23 р.

Завдання видав «28» грудня 2022 р. Керівник Красиленко В.Г.

(підпис)

Дипломник Лукомський А.І.

(підпис)

## ЗМІСТ

ВСТУП.....	5
РОЗДІЛ 1. ТЕХНОЛОГІЇ РОЗРОБКИ САЙТІВ .....	8
1.1. Аналіз вихідних даних і вимог до WEB системи.....	8
1.2 Аналіз сучасних WEB систем.....	12
1.4 Короткий опис WEB-редакторів.....	17
1.4.1. Adobe GoLive.....	17
1.4.2. Macromedia Dreamweaver/Dreamweaver UltraDev Dreamweaver .....	19
1.5. Розробка специфікації вимог SRS.....	21
1.5.2 Побудова діаграм використання сайту.....	25
1.5.3.Побудова діаграм послідовності сайту .....	27
РОЗДІЛ 2. РОЗРОБКА БАЗИ ДАНИХ .....	30
2.1 Опис предметної області.....	30
2.2 Реляційна модель бази даних .....	32
2.3 Розробка бази даних .....	34
РОЗДІЛ 3 РОЗРОБКА WEB САЙТУ ТОРГОВОГО ПІДПРИЄМСТВА З	3
ПРОДАЖУ БІОГАЗОВИХ УСТАНОВОК .....	42
3.1 Структурна схема сайту.....	42
3.2 Керівництво програміста .....	43
3.2.1. Налаштування сервера .....	43
3.3 Керівництво користувача .....	53
3.4. Оптимізація та налаштування сайту.....	55
ВИСНОВКИ .....	57
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	58
ДОДАТКИ .....	60



## ВСТУП

Актуальність теми. Розробка WEB-сайту для торгового підприємства з продажу біогазових установок є актуальною темою в сучасному бізнесі, оскільки віртуальні технології та інтернет-маркетинг є важливими інструментами для просування товарів та послуг.

Розробка WEB-сайту може забезпечити підприємству доступ до більш широкої аудиторії клієнтів та покупців, підвищити рівень продажів та конкурентоспроможності на ринку. Крім того, WEB-сайт дозволить забезпечити швидкий та зручний доступ до інформації про продукцію, її характеристики, вартість та умови доставки, що є важливим фактором для клієнтів.

Розробка сайту дозволяє компанії збільшити свій охоплення аудиторії, відкриваючи доступ до продуктів та послуг не тільки на місцевому рівні, але і на національному та міжнародному. Крім того, сайт дозволяє підприємству зберігати додаткові витрати на рекламу та залучення нових клієнтів, оскільки його вірогідніше знайдуть потенційні клієнти самостійно, використовуючи пошукові системи.

Отже, можна стверджувати, що дана тема є дуже актуальною та має великий потенціал для розвитку та успіху бізнесу в наш час.

Метою дипломної роботи є створення функціонального та зручного для користувачів сайту, який дозволить компанії збільшити своє охоплення аудиторії та підвищити ефективність своєї роботи. Для досягнення поставленої мети необхідно виконати такі завдання:

1. Дослідити технології розробки сайтів
2. Визначити основні терміни та поняття.
3. Дослідити сучасне становище у сфері WEB-дизайну
4. Вивчити ключові особливості сучасних WEB-редакторів та описати коротко WEB-редактори.
5. Розробити базу даних.

6. Описати предметну область.
7. Вивчити реляційну модель бази даних.
8. Розробити WEB-сайт торгового підприємства:
9. Створити структурну схему сайту.

Об'єктом дослідження виступає WEB – сайт торгового підприємства з продажу біогазових установок.

Предметом дослідження дипломної роботи є розробка WEB-сайту для торгового підприємства, що спеціалізується на продажі біогазових установок.

У рамках дослідження буде проведено аналіз потреб та очікувань потенційних клієнтів, а також розроблено функціонал сайту, що відповідає цим потребам. Крім того, будуть розглянуті питання щодо дизайну та інтерфейсу сайту, його оптимізації для пошукових систем та безпеки інформації. Результатом дослідження буде розроблена та впроваджена WEB-платформа, що дозволить підприємству залучати нових клієнтів, збільшувати продажі та підвищувати ефективність взаємодії з поточними клієнтами.

Дослідження включає в себе вивчення технологій розробки сайтів, розробку бази даних, створення структурної схеми сайту, налаштування сервера та використання CMS WordPress.

У дипломній роботі «Розробка WEB-сайту торгового підприємства» можуть бути використані наступні методи дослідження:

1. Аналіз наукової літератури - для ознайомлення з основними поняттями, термінами, принципами розробки WEB-сайтів, з вивченням сучасного стану ринку WEB-дизайну та розробки.
2. Експертний аналіз - для отримання висновків від фахівців з різних галузей, що стосуються проекту, з метою вибору оптимальних технологій, програмного забезпечення, методів розробки та інших аспектів проекту.
3. Квалітативні та кількісні дослідження - для вивчення попиту на продукцію та послуги, які пропонує торгове підприємство, та визначення особливостей цільової аудиторії.
4. Експериментальні дослідження - для перевірки ефективності

розроблених рішень, щодо розробки сайту, його функціональності та зручності використання.

5. Статистичний аналіз - для обробки отриманих даних, їх систематизації та визначення ступеня відповідності результатів до мети дослідження.

Науковою новизною являються такі аспекти:

1. Розробка бази даних для торгового підприємства, яка буде включати в себе необхідну інформацію про товари та послуги, які надає підприємство, та забезпечувати ефективний доступ до цієї інформації для користувачів сайту.

2. Використання сучасних технологій розробки WEB-сайтів, зокрема, використання CMS WordPress, що дозволяє швидко створювати та змінювати контент на сайті.

3. Розробка структури сайту з урахуванням потреб та вимог користувачів, забезпечення зручного та швидкого пошуку інформації.

4. Вивчення попиту на продукцію та послуги, що пропонує торговельне підприємство, та визначення особливостей цільової аудиторії, що дозволяє зробити сайт більш привабливим для потенційних клієнтів.

5. Розробка детального керівництва користувача, яке містить інформацію про можливості та функції сайту, а також підказки та поради щодо його ефективного використання.

Усі ці аспекти дозволяють створити ефективний та зручний WEB-сайт для торгового підприємства, що відповідає вимогам сучасного ринку та забезпечує зручне та швидке обслуговування користувачів.

Практична цінність полягає в тому що, торговельному підприємству дозволить підвищити ефективність роботи, полегшити процеси продажу та обслуговування клієнтів.

## РОЗДІЛ 1

### ТЕХНОЛОГІЇ РОЗРОБКИ САЙТІВ

#### 1.1. Аналіз вихідних даних і вимог до WEB системи

Щоб розмістити інформацію в Інтернеті, її необхідно подати у вигляді файлів, які програмами користувачів будуть визначені як сторінки документа (цей документ може бути призначений для читання, перегляду графіки або відео, заповнення інтерактивних форм та інших цілей). Ці файли створюються за допомогою спеціальної мови опису та розмітки гіпертексту HTML (Hyper Text Markup Language) і називаються HTML-файлами.

*WEB-сторінка* – це документ, який може отримати користувач у результаті прочитання HTML-файлу, використовуючи WEB-браузер. WEB-сторінка може містити текст, графіку, посилання на інші документи та різні файли: текстові, графічні, аудіо або відео [15].

*WEB-браузер* – спеціальна програма, призначена для читання HTML-файлів та їх відображення. В даний час існує безліч таких програм, але найбільшого поширення набули MS Internet Explorer, Opera, Netscape Navigator. WEB-браузер можна використовувати для переходів з однієї WEB-сторінки на іншу за гіперпосиланнями, для завантаження файлів з Internet на комп'ютер, для відтворення впроваджених на WEB-сторінку файлів мультимедіа, а також для запуску програм (наприклад, поштових)

*WEB-сайт/WEB-вузол* – декілька WEB-сторінок, об'єднаних у загальну структуру.

*WEB-сервер* - підключений до мережі Internet комп'ютер, що підтримує протокол HTTP (Hyper Text Transfer Protocol) Він використовується для доставки даних в Internet і дозволяє користувачам клієнтських програм завантажувати та відтворювати текст, малюнки, звук, відео та інші дані. Протокол FTP (File Transfer Protocol) також є найпоширенішим способом завантаження файлів на віддалений сервер [16].

*Гіперпосилання* – це текстовий або графічний елемент WEB-сторінки,



який є вказівником переходу на іншу WEB-сторінку. Розрізняють внутрішні та зовнішні гіперпосилання. Внутрішні – забезпечують перехід на іншу частину цієї сторінки, файл або іншу мандрівнику цього WEB-сайту. Зовнішні – вказують на файл або сторінку іншого WEB-сайту.

Аналіз вихідних даних та вимог до WEB-системи є першим кроком у розробці WEB-сайту торгового підприємства. Етап дозволяє використовувати потреби і очікування користувачів, а також цей необхідний функціонал та можливості WEB-сайту.

Однією з цих головних задач етапу є збір вимог до WEB-системи від клієнта та аналізу користувачів. Для цього необхідно провести детальний аналіз бізнес-процесів торгового підприємства та взаємодії з його клієнтами. Також варто дослідити WEB-сайти конкурентів та розробити їхні особливості та переваги.

Кожному, важливо відповідати наступним вимогам та функціональним можливостям WEB-системи [16]:

- Наявність інформації про товари та послуги, їх опис, ціни та умови придбання.
- Наявність функціонального кошика для збору товарів та оформлення замовлення.
- Можливість реєстрації користувачів та авторизації на сайті.
- Підтримка онлайн-оплати замовлень.
- Можливість розміщення та оновлення новин, акцій та інших матеріалів.
- Функціональний для зворотного зв'язку користувачів з технічною підтримкою або відділом продажу.

В результаті аналізу вихідних даних та вимог до WEB-системи, буде сформована концепція WEB-сайту, яка відповідає потребам та очікуванням користувачів та бізнес-процесів торгового підприємства.

Метою даної системи є створення WEB-сайту для торгового підприємства, який забезпечуватиме швидкий пошук, обробку та зберігання інформації про події, що відбуваються на підприємстві. Для реалізації цієї

мети потрібно розробити відповідну WEB-систему. Роль інтерфейсу між програмною програмою та базою даних виконуватиме програмний комплекс - система керування базами даних (СКБД).

База даних буде зберігатися на сервері, а управління нею здійснюватиметься за допомогою системи управління базами даних MySQL.

У табл.1.1 наведені дані, які мають міститись у базі даних та інформацію яка має бути на WEB – сайті.

*Таблиця 1.1*

Інформація в базі даних та на WEB – сайті

Інформація в базі даних	Інформація на WEB-сайті
Розклад подій	Календар подій
Інформація про підприємство	Вид діяльності
Рекламні дії	Актуальні пропозиції
Контактна інформація	Контактна інформація
Карта підприємства	План торгового підприємства
Відгуки про підприємство	Секція відгуків

*Джерело: сформовано автором на основі даних [17]*

З таблиці видно, що інформація буде відображена на WEB-сайті торгового підприємства у вигляді календаря подій, виду діяльності, актуальних пропозицій, контактної інформації, плану торгового підприємства та секції відгуків. Відповідно, база даних та WEB-сайт торгового підприємства повинні бути розроблені з метою забезпечення швидкого пошуку, обробки та зберігання інформації про заходи, що проходять на території підприємства та надання користувачам зручного та доступного інтерфейсу для отримання необхідної інформації.

Для реалізації доступу користувачів до інформації бази даних вирішено використовувати HTML-технології та WEB-браузер Google Chrome як інтерфейс для клієнта, що має переваги в простоті доступу до інформації та відсутності необхідності установки додаткового ПЗ. Розроблюваний WEB сайт має надавати користувачам можливість фільтрування інформації за різними критеріями з можливістю задавати тип фільтрації для підвищення швидкості роботи. При цьому програмне забезпечення, що використовується,

повинно бути ліцензійно чистим, містити вбудовані можливості забезпечення безпеки та надійності зберігання даних. Для входу в систему користувача передбачено проходження реєстрації та введення пароля.

При розробці WEB-системи (BC) необхідно передбачити виконання наступних вимог (табл.1.2).

Таблиця 1.2

Вимоги до розробки WEB-сайту торгового підприємства

Назва вимоги	Опис вимоги	Характеристика вимоги
Користувацька аутентифікація	Вхід в систему повинен проходити через реєстрацію і введення пароля	Обов'язкова
Доступ до інформації	WEB-сайт повинен містити інформацію про заходи, що проходять на території центру	Обов'язкова
Фільтрація інформації	WEB-сайт повинен дозволити фільтрувати інформацію за різними критеріями (час, відвідувач і т.п.) з можливістю вибору типу фільтра (містить, починає, так само, закінчується)	Обов'язкова
WEB-інтерфейс	Перегляд інформації повинен бути реалізований з використанням HTML-технологій, а як інтерфейс для клієнта - WEB-браузер Google Chrome	Обов'язкова
Безпека і надійність зберігання даних	Використовування програмного забезпечення повинно бути ліцензійно чистим, Включіть вбудовані можливості забезпечення безпеки та надійності зберігання даних	Рекомендована
Мобільна версія сайту	WEB-сайт повинен бути оптимізований для відображення на мобільних пристроях	Опціональна

*Джерело: сформовано автором на основі даних [17]*

З таблиці видно, що при розробці WEB-сайту торгового підприємства необхідно враховувати різноманітні вимоги, що стосуються як функціональності, так і технічних характеристик. Необхідно забезпечити швидкий та зручний доступ до інформації про заходи та події, що проходять в торговому центрі. Крім того, потрібно забезпечити можливість фільтрування цієї інформації за різними критеріями, що значно спростить процес пошуку потрібної інформації.

Також вимоги стосуються безпеки даних та захисту від несанкціонованого доступу. Важливо забезпечити ліцензійне програмне забезпечення, яке містить вбудовані засоби забезпечення безпеки та надійності зберігання даних.

Загалом, для успішної реалізації проєкту необхідно уважно дотримуватися всіх вимог, що вказані в таблиці, щоб забезпечити зручний та безпечний доступ до інформації для користувачів.

## 1.2 Аналіз сучасних WEB-систем

Процес розробки розвиненого мережного проєкту, як правило, не закінчується створенням усіх складових гіпертекстових документів та структури посилань. Щоб «вживити» новий сайт у «тканину» всесвітньої Мережі, потрібно виконати великий обсяг робіт із супроводу проєкту. Реєстрація на пошукових машинах та в каталогах, організація мережевої реклами, обмін посиланнями та банерами з спорідненими мережевими ресурсами – ось далеко не повний перелік необхідних заходів щодо підтримки WEB-вузла. Всі ці операції супроводжуються змінами версій документів на віддаленому сервері. В оболонку сучасних WEB-редакторів інтегрований спеціальний засіб, призначений для обслуговування опублікованих у мережі документів та сайтів. Воно має усі необхідні інструменти для дистанційного оновлення версій документів. Цей засіб підтримує розподілену роботу кількох виконавців над одним мережовим проєктом, має механізм синхронізації версій документів та захисту від несанкціонованого доступу.

Відомо, що багато візуальних редакторів HTML дуже нераціонально поводяться з кодами гіпертекстової розмітки. Це означає, що вони породжують надлишковий, громіздкий вихідний текст, перевантажений складними мовними конструкціями. Сучасні WEB-редактори породжують чистий лаконічний вихідний текст, у якому майже немає допоміжних дескрипторів та громіздких службових коментарів.

Сучасні WEB-редактори ґрунтуються на принципі відкритої

архітектури. Це означає, що повністю відкритий інтерфейс прикладного програміста (Application Programming Interface, API), за допомогою якого сторонні програмісти та софтверні фірми можуть виконати радикальні зміни функціональних можливостей програми та її інтерфейсу: додати новий інструмент, створити палітру або розділ меню, запрограмувати новий об'єкт або мультимедійний ролик та інше [16].

З метою розробки WEB-сайту торгового підприємства було проведено аналіз сучасних WEB-систем, які використовують для розробки та підтримки подібних проєктів.

Однією з найпоширеніших технологій є CMS (Content Management System), яка дозволяє створювати WEB-сайти з використанням готових шаблонів та функціональних компонентів. Найвідомішими CMS є WordPress, Joomla та Drupal. Вони мають велику кількість плагінів і тем, що дозволяють створювати WEB-сайти різних типів і складності. Однак вони можуть бути менш гнучкими, а також вимагати більш високого рівня знань із програмування для налаштувань та модифікації деяких функціональних компонентів.

Іншою популярною технологією є фреймворки, такі як Laravel, Django та Ruby on Rails. Вони можуть створити більш складні та гнучкі WEB-додатки, які мають високу продуктивність та зручне управління базами даних. Однак вони можуть вимагати велику кількість часу на розробку та потребують високого рівня знань з програмування.

Також була важливою можливістю використання фронтенд-фреймворків, таких як Angular, React та Vue.js, для розробки WEB-інтерфейсу. Вони можуть створювати більш інтерактивні та динамічні WEB-сторінки, які працюють з даними в режимі реального часу та мають високу продуктивність. Однак вони можуть вимагати велику кількість часу на розробку та потребують високого рівня знань з програмування [14].

З кількістю вищевказаних факторів, з якими необхідно виконати при розробці WEB-сайту торгового підприємства, існує багато інструментів та

технологій, які можуть бути використані для розробки WEB-сайту. Наприклад, з розвитком технологій на сьогоднішній день, WEB-розробники мають доступ до широкого спектру інструментів, які не дозволяють створювати WEB-сайти з безкоштовними функціональними можливостями та естетичним дизайном.

Одним із найпоширеніших інструментів для розробки WEB-сайтів є Content Management System (CMS) - система управління вмістом, яка дозволяє створювати та підтримувати WEB-сайти без великих зусиль та знань WEB-розробки. Найпопулярнішими CMS на сьогоднішній день є WordPress, Drupal та Joomla. Вони можуть створювати WEB-сайти з широким спектром функціональних можливостей та дизайну, а також мати широкий спектр плагінів та розширень для покращення їх функціональності.

Іншими інструментами для розробки WEB-сайтів є фреймворки та бібліотеки, такі як Angular, React, Vue.js, які можуть швидко та ефективно розробити WEB-додатки з високою продуктивністю та масштабістю.

Крім того, для забезпечення безпеки WEB-сайту та його користувачів можуть використовуватися спеціальні сервіси, такі як Cloudflare, Sucuri, Incapsula, які надають послуги захисту від DDoS-атак, вірусів та інших загроз.

Отже, при розробці WEB-сайту торгового підприємства необхідно отримати багато факторів, таких як функціональність, продуктів та послуги, візуальний дизайн, швидкість завантаження, доступність для користувачів з обмеженими можливостями та інше. Для досягнення успіху необхідно дотримуватись критерій хорошого WEB-сайту, які складаються в таблиці 1.3.

Дотримання цих критеріїв є успіхом для успіху WEB-сайту торгового підприємства та допомагає забезпечити задоволення клієнтів і підвищення конкурентоспроможності компанії. Крім того, важливо забезпечити постійне оновлення та підтримку WEB-сайту, забезпечення його безпеки та забезпечення швидкості роботи. Важливо також забезпечити зручний та легкий для навігації інтерфейс користувача, що забезпечить зручний пошук та доступ до необхідної інформації.

## Критерії хорошого WEB-сайту торгового підприємства

Критерії хорошого WEB-сайту	Опис
Функціональність	WEB-сайт повинен зберігати всі функції безпеки, які без користувачів легко розміщувати та купувати товари та послуги.
Візуальний дизайн	Дизайн WEB-сайту повинен бути привабливим і професійним, щоб залучити користувачів і збільшити їх задоволення від відвідування сайту.
Швидкість завантаження	WEB-сайт повинен завантажуватися швидко, щоб не втратити користувачів.
Доступність для користувачів з обмеженими можливостями	WEB-сайт повинен бути доступним для всіх користувачів, у тому числі тих, у кого є обмежені можливості, такі як вікна зору або слуху.
Безпека та приватність	WEB-сайт має бути захищеним від хакерських атак та захистом персональної інформації користувачів у безпеці.
адаптивність	WEB-сайт повинен бути адаптивним до різних розмірів екрану, щоб бути зручним для використання на будь-яких пристроях.
SEO-оптимізація	WEB-сайт повинен мати ефективну SEO-стратегію, щоб забезпечити його високу позицію в пошукових системах.

*Джерело: сформовано автором*

Успішна реалізація цих критеріїв дозволить підприємству збільшити продажі та збільшити свою аудиторію.

### 1.3 Ключові особливості сучасних WEB-редакторів

WEB-редактор - це програмне забезпечення, яке використовується для створення та редагування WEB-сторінок. Сучасні WEB-редактори мають низку ключових особливостей, які створюють їх корисними для розробки WEB-сайтів.

Основні риси сучасних WEB-редакторів включають в себе [11]:

*Редактор джерел коду.* Редактор джерел коду надає необмежені можливості контролю над HTML-тегами та їх форматування. Щоб переключитися з режиму розмітки сторінки в режим редагування коду, достатньо просто клацнути по вкладці у вікні документа. Внівши зміни до коду, ви можете відразу переглянути їх, клацнувши по іншій вкладці.

*Контроль синтаксису.* Досвідчені WEB-програмісти знають, що підтримка HTML і функції JavaScript трохи відрізняється на різних платформах. Саме тому сучасні WEB-дизайнери має вбудовану функцію перевірки синтаксису. Вона дозволяє дізнатися, які браузері і на яких платформах підтримують всі ті елементи, які ви передбачили на своїй сторінці, будь то DHTML або таблиці стилів. Ви можете створити сторінку в HTML 4.0, перетворити її на стандартний формат HTML 3.2 або зробити в обох варіантах, надавши право вибору користувачеві.

*Динамічні компоненти та шаблони.* Якщо на WEB-сайті є такі спільні елементи дизайну, як навігаційні панелі та заголовки, які присутні на кожній сторінці, то вам особливо сподобаються динамічні компоненти, що дозволяють використовувати деякі елементи сторінок, зроблені раніше та вже оснащені текстами, картинками чи іншим вмістом. Простіше кажучи, можна створювати шаблонні сторінки і використовувати їх знову і знову для створення складних сайтів.

*Вбудований ftp-клієнт.* Сучасні WEB-сайти оснащені повним інтегрованим FTP-клієнтом, який дозволяє завантажувати та оновлювати WEB-сайт. Коли створюється необхідність оновити сайт, запускається процес - буде оновлено лише змінені файли. Можна також переносити окремі сторінки, просто «перетягуючи» їх з панелі FTP або, навпаки, на неї. Для зручності роботи сучасні WEB-дизайнери розраховують час, який знадобиться відвідувачам сайту для завантаження ваших сторінок.

*Робота із спецефектами.* Стандарт для підтримки мультимедійних засобів, заснованих на WEB. Користувач може імпортувати звук, відео, спецефекти та визначати зв'язки між ними. Новий редактор дозволяє комбінувати відео та аудіо, створювати переходи та робити різні інші перетворення для підвищення якості відео – і все це завдяки інтуїтивному інтерфейсу сучасних WEB-сторінок.

*Готові до використання Java-сценарії.* Сучасні WEB-редактори містять цілий набір готових до використання Java-скриптів, які дозволяють внести



елементи інтерактивності до WEB-сторінок.

*Таблиці стилів (Cascading Style Sheets).* Сучасні WEB-редактори підтримують технологію так званих таблиць стилів (cascading style sheets (CSS)) - cutting-edge. Технологія надає безпрецедентні можливості контролю за зовнішнім виглядом сторінок, у тому числі й платформи-незалежними варіантами. Вам більше не доведеться створювати спеціальні однопикселові зображення та вкладені таблиці, щоб згорнути сторінку так, як ви хочете.

WEB-редактор дозволяє візуально створювати та редагувати таблиці стилів, позиціонувати зображення та текстові блоки на сторінці за допомогою плаваючих областей, визначати шрифти, кольори та розміри для досягнення найкращого варіанту дизайну, а також створювати та редагувати як зовнішні стилі, визначені для всього сайту, так та стилі окремих сторінок.

#### 1.4 Короткий опис WEB-редакторів

##### 1.4.1. Adobe GoLive

Цей редактор є досить професійним інструментом для WEB-розробки, він може працювати з графічними програмами компанії Adobe (Photoshop, Illustrator, LiveMotion тощо). Програма має потужні можливості для дизайну. Також програма має чудову підтримку можливостей мультимедіа - в неї вбудований редактор QuickTime, а також передбачена підтримка форматів SVG і Flash. Програма генерує чіткий вихідний текст та забезпечує широкі можливості керування сайтами. Але за всі ці достоїнства доведеться платити - освоїти програму нелегко, але в результаті ви зможете створювати рішення для WEB дуже високого класу. Також цей пакет підтримує технологію drag-and-drop майже всюди, де це можливо. Професіонали в області WEB-дизайну та фанати інших продуктів Adobe безсумнівно вважають GoLive дуже вдалим доповненням до звичного набору інструментів. Allaire ColdFusion ColdFusion орієнтований на серйозних розробників, які створюють програми WEB, що підтримують електронну комерцію та бази даних [10].

У цій програмі реалізовано візуальне програмування професійного

рівня, а також інструменти розробки баз даних та налагодження; вона містить набір майстрів WEB Application Wizards. ColdFusion дозволяє безпосередньо під час редагування згорнути вихідний текст і швидко перейти до іншої секції. Також є можливість створювати складні SQL-оператори за допомогою інструментального набору Visual Database, використовувати переваги вбудованих редакторів каскадних таблиць стилів та Image Map, а також налагоджувати програми в інтерактивному режимі, рядок за рядком.

Одним із величезних плюсів програми є її здатність маніпулювати великими проектами, в яких задіяно багато розробників та серверів. У програму також вбудовано потужну серверну мову сценаріїв (Cold Fusion Markup Language, або CFML) із синтаксисом на основі тегів, який працює з HTML та XML. А підтримка WML (Wireless Markup Language – мова розмітки для бездротових пристроїв) – це явно націлено на майбутнє.

У ColdFusion інтегрований пакет HomeSite, тому якщо вам здебільшого доводиться генерувати звичайні WEB-сторінки, то є сенс придбати інструмент HomeSite замість всього комплекту. Allaire HomeSite Паке́т HomeSite включає більшість інструментів, що використовуються професіоналами, дотримуючись балансу між новітніми можливостями і зручністю використання. Орієнтований цей паке́т скоріше розробників, ніж дизайнерів. Має достатній набір функцій, щоб привернути увагу любителів ручного кодування, підтримує такі сучасні технології, як DHTML, SMIL, CSS, ASP, JSP, Perl та XML.

HomeSite виділяється тим, що дає користувачам можливість налаштувати інтерфейс: є можливість створювати власні інструментальні панелі, додавати кнопки на замовлення, коригувати клавіатурні прискорювачі програми і навіть використовувати Visual Tool Markup Language для зміни інтерфейсу програми.

Можна також пристиковувати, ховати чи відокремлювати частини інтерфейсу. У програмі є чудова функція CodeSweeper, яка дає змогу вносити глобальні зміни на всьому WEB-сайті. Ще один плюс - режим автоматичного

завершення, коли під час введення тексту задані користувачем рядки завершуються самі.

Напрошується висновок: великий плюс цієї програми, особливо для тих розробників, які люблять створювати зручні для себе інтерфейси, полягає в тому, що дана програма має значно розширені можливості налаштування в порівнянні з іншими подібними додатками. *Vare Bones VBEedit* Цей пакет спрямований на HTML-програмістів, які звикли робити самі, тобто. тут немає такої величезної кількості функцій *drag-and-drop*, як в інших програмах, хоча програма має безліч кнопок для додавання тегів, елементів заголовків і посилань. Пакет має безліч корисних функцій, таких як вбудована перевірка орфографії, багаторазова скасування, підтримка 13 різних мов програмування, попередній перегляд по одному клацанню миші, інструменти завершення тегів та вбудований генератор таблиць. Але він призначений головним чином для користувачів, які знають, як писати вихідні тексти, люблять створювати сторінки в текстовому редакторі і бажають уникнути візуальних WEB-програм, які можуть змінити формат.

#### 1.4.2. Macromedia Dreamweaver/Dreamweaver UltraDev Dreamweaver

Даний WEB-редактор, став улюбленцем програмістів для WEB-дизайну майже з самого моменту свого створення. У програмі дотримано балансу між створенням чіткого вихідного тексту HTML з повним контролем над ним та реалізацією необхідних засобів *drag-and-drop*, що полегшують розробку. Ця програма не вносить небажаних змін у вихідні тексти та створює компактні WEB-сайти. Нещодавно Macromedia вдосконалила програму, випустивши *Dreamweaver UltraDev*, який побудований на основі ядра *Dreamweaver* і має розширені функціональні можливості. *UltraDev* дозволяє з'єднувати WEB-сторінки з базами даних, переглядати реальні дані в середовищі розробки та має інтерактивність на рівні сервера. Ці продукти не мають готових засобів електронної комерції, але все одно безліч фанатів програмування довіряють саме цим пакетам! *Microsoft FrontPage* Професійні розробники часто не

сприймають всерйоз сайти, створені за допомогою FrontPage, оскільки більшість їх вихідного тексту генерується автоматично. Але вони забувають про вичерпний набір функціональних можливостей і про те, як багато людей долучилися до WEB-дизайну завдяки цьому продукту [19].

В інтерфейсі програми багато запозичено зі знайомих програм Microsoft Office, тому освоєння програми не складає труднощів. У програмі поєднуються прості та зрозумілі майстри з потужною функціональністю. FrontPage має інтерфейс, що працює в різних уявленнях: є подання сторінки для роботи над сторінками WEB, подання папок для вибору змісту та навігаційне подання для перевірки посилань.

У комплекті з програмою пропонуються 60 шаблонів або Themes (тем), з широким діапазоном видів та стилів. Корисні майстри дозволяють почати роботу з шаблонами, але у вас завжди є доступ до вихідного тексту. Вміння працювати з FrontPage не замінить знання HTML, але допоможе вам швидко створювати чудові сайти. Загалом FrontPage є чудовою програмою для початківця.

NetObjects Fusion На перший погляд зручний у навігації інтерфейс та надійні шаблони говорять про Fusion, як про просту програму, але за зовнішньою простотою ховаються безліч потужних функцій. Fusion не призначений для програмістів на HTML – насправді програма не забезпечує легкої зміни вихідного тексту HTML вручну, проте вона має великий набір таких опцій, як «складання» мережевого магазину та підключення зовнішніх баз даних за допомогою досить зручних майстрів. Наприклад, можна відразу змінити логотип відразу на 200 сторінках або додати ще кілька навігаційних посилань, але вихідний текст може виявитися досить «важким», що збільшить час завантаження сторінки. Але цілком очевидно, що Fusion допоможе з мінімальними зусиллями та за найкоротший термін створити сайт від початку до кінця [19].

### 1.5. Розробка специфікації вимог SRS

Вимоги до WEB-додатку визначають набір характеристик, якості та функцій програмного забезпечення, яке вже розробляється або знаходиться у процесі розробки. Ці вимоги формуються під час аналізу вимог та фіксуються в специфікації вимог, діаграмах прецедентів та інших артефактах, які стосуються процесу аналізу та розробки вимог [4]. Дана WEB-сторінка призначена для користувачів різного віку та статі, але якщо вона має дорослий контент, то перед тим, як користувачі розпочнуть роботу з нею, виводиться попередження.

У табл. 1.4 представлені вимоги WEB-додатку для користувачів, адміністраторів та функціональні вимоги WEB-системи.

*Таблиця 1.4*

#### Вимоги WEB-додатку для користувачів, адміністраторів та функціональні вимоги WEB-системи

Вимоги до WEB-додатку для звичайних користувачів	Вимоги до WEB-системи для адміністраторів	Функціональні вимоги WEB-системи
Простий і зрозумілий інтерфейс користувача	Можливість редагування бази даних	Сортування новин
Можливість пошуку інформації за ключовими словами	Наявність зручного інтерфейсу для взаємодії з базою даних і сервісом	Сортування афіші
Можливість коментування		Доступ до бази даних
Можливість переглядати новини		Збереження коментарів
Можливість перегляду афіші		Можливість робити онлайн покупки
		Збереження фотогалереї
		Можливість оцінювати роботу

*Джерело: сформовано автором на основі даних [10]*

WEB-додаток для звичайних користувачів повинен мати простий та зрозумілий інтерфейс користувача, можливість пошуку інформації за

ключовими словами, коментування, перегляд новин та афіші. WEB-система для адміністраторів повинна мати можливість редагування бази даних та наявність зручного інтерфейсу для взаємодії з базою даних і сервісом. Функціональні вимоги WEB-системи включають сортування новин та афіші, доступ до баз даних, збереження коментарів, можливість робити онлайн-покупки, збереження фотогалереї та можливість оцінювати роботу. Дотримання цих вимог дозволить створити зручний та функціональний WEB-додаток для користувачів і WEB-системи для адміністраторів, що задовольнятимете їх потреби та вимоги.

Специфікація вимог для WEB-системи є описом поведінки програми, яку треба розробити, і включає в себе різні користувацькі сценарії, які описують взаємодію між користувачами та програмним забезпеченням. Ці сценарії представляють функціональні вимоги, але специфікація також містить нефункціональні вимоги, які обмежують дизайн або реалізацію, такі як продуктивність, стандарти якості і проектні обмеження.

WEB-система, що розглядається, є незалежною від інших програмних виробів і може виконувати свої функції автономно. В цьому документі WEB-система описується як взаємопов'язана структура, що складається з WEB-сайту, бази даних та системи управління базами даних (СУБД).

У табл.1.5. наведені характеристики функцій ВС.

Користувачі інформаційної WEB-системи мають доступ до різноманітних функцій, таких як пошук інформації, коментування, перегляд новин та афіш, редагування бази даних, збереження даних, онлайн-покупки, фото та можливість оцінювати роботу систем. Кожен користувач може мати свій рівень доступу, який збільшується системою. Обліковий запис дозволяє користувачам зберігати персональні налаштування та історію взаємодії з системою.

У таблиці 1.6 нами наведено характеристику користувачів WEB-системи.

## Функції інформаційної WEB-системи

Характеристика	Опис
Рівень доступу	Рівень прав користувача в системі (адміністратор, користувач)
Обліковий запис	Можливість створення електронного запису для користувача
Авторизація	Механізм перевірки ідентифікації користувача
Пошук інформації	Можливість знаходження інформації за ключовими словами
Коментування	Можливість залишити коментарі під статтями і новинами
Перегляд новин	Можливість перегляду останніх новин у системі
Перегляд афіші	Можливість переглядати інформацію про майбутні події
Робота з базою даних	Можливість редагування бази даних
Онлайн покупки	Можливість здійснення покупок через систему
Збереження фото	Можливість збереження фотографій та інших медіа-файлів
Оцінювання роботи	Можливість відвідувачів оцінювати якість роботи системи

*Джерело: сформовано автором на основі даних [16]*

Дана таблиця містить вимоги до функціональності сайту, які мають задовольняти потреби різних класів користувачів, таких як відвідувачі, читачі, магазини, адміністратори та технічна підтримка. Кожен клас користувачів має свої вимоги до сайту, включаючи функції, які мають бути доступними на сайті, такі як перегляд інформації, можливість реєстрації та авторизації, додавання та керування товарами, коментування новин тощо.

Загальні вимоги до того, що стосуються всіх класів користувачів, включають швидкість переходу між сторінками, повну та безпомилкову реалізацію виявлених функцій сайту та знищення видаленої інформації з бази даних.

Важливо забезпечити зручний та швидкий спосіб пошуку та сортування інформації для всіх класів користувачів, а також можливість коментування новин.

## Характеристика користувачів WEB-системи

Клас користувачів	Функції сайту
Відвідувачі	- Перегляд інформації - Реєстрація - Пошук інформації - Сортування новин
Користувачі(читач)	- Перегляд інформації - Авторизація Пошук інформації - Сортування - Коментування новин - Сервіс онлайн покупок
Користувачі(магазин)	- Перегляд інформації - Авторизація - Додавання товарів - Керування товарами (редагування, видалення, доповнення) - Пошук товарів - Сортування товарів - Коментування новин
Адміністратор	- Надання інформації - Контроль за магазинами - Редагування контенту ВС
Технічна підтримка	- Розробка нового функціоналу - Внесення змін у зовнішній вигляд сайту

*Джерело: сформовано автором*

Для адміністраторів важливо мати можливість контролювати магазини та редагувати контент на сайті.

Для технічної підтримки необхідно мати можливість розробки нового функціоналу та внесення змін у зовнішній вигляд сайту.

Усі ці вимоги мають бути виконані для забезпечення задоволення потреб користувачів та ефективності роботи сайту.

Зазвичай, WEB-сайти є складними та динамічними структурами, на які виділяється короткий час для розробки з метою якнайшвидшого запуску та отримання результату. Часто розробники не докладають достатньо зусиль у планування розробки та архітектури системи, і відразу ж переходять до написання коду. Як наслідок, код серверної частини може бути написаний з



нуля, таблиці в базах даних створюються за потребою, і у підсумку архітектура системи може розвиватися у непередбачуваний спосіб. Однак, за допомогою простого моделювання та суворого підходу до програмування, можна зробити процес розробки більш гладким та гарантувати те, що створена WEB-система буде легко підтримуватися у майбутньому.

За допомогою моделювання можна вирішити чотири завдання: візуалізувати поточний або бажаний стан системи для замовника, описати структуру або поведінку системи, отримати шаблон для конструювання системи та документувати прийняті рішення за допомогою отриманих моделей. Моделювання набуває особливого значення при розробці великих і складних систем, оскільки людське сприйняття таких сутностей є обмеженим. Моделювання дозволяє зосередитись на певних аспектах системи в кожен момент, звузити проблему та розділити її на менші, що полегшує вирішення складних завдань. Крім того, правильно обрана модель дозволяє створювати проекти на вищих рівнях абстракції.

Щоб ефективно і швидко розробити програмний продукт з мінімальними витратами, необхідно правильно підібрати робочу силу, визначити відповідні інструменти та встановити правильний напрямок роботи. Для досягнення поставленої мети з урахуванням витрат на життєвий цикл системи, необхідно детально продумати процес розробки програмного продукту та пристосувати його до потреб вашого бізнесу та технології.

### 1.5.2 Побудова діаграм використання сайту

Були описані наступні блоки для роботи WEB-сайту на діаграмі варіантів використання:

Відвідувач - особа, яка перейшла на сайт для ознайомлення з контентом (рис. 1.1);

Читач - особа, яка зареєструвалася на сайті з метою отримання новин або здійснення онлайн-покупок (рис. 1.2);

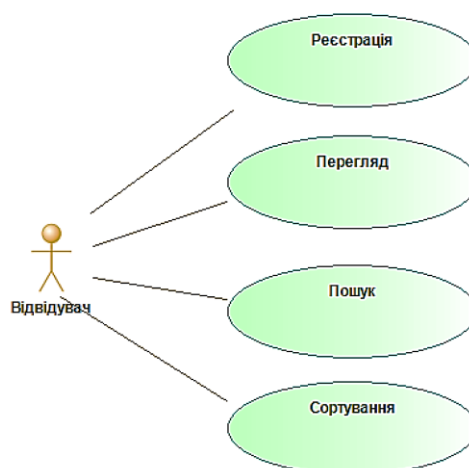


Рис.1.1. Діаграма варіантів використання «Відвідувач»

Джерело: сформовано автором

Магазин - особа, яка зареєструвалася на сайті для продажу товарів (рис. 1.3);

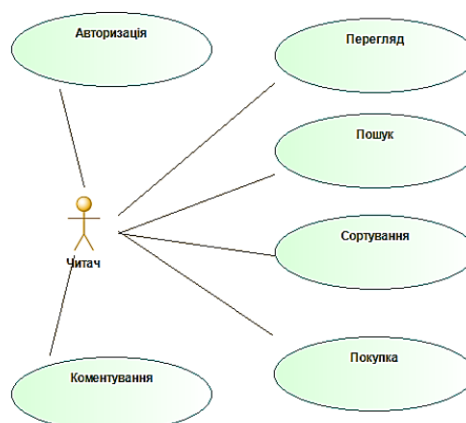


Рис.1.2. Діаграма варіантів використання «Читач»

Джерело: сформовано автором

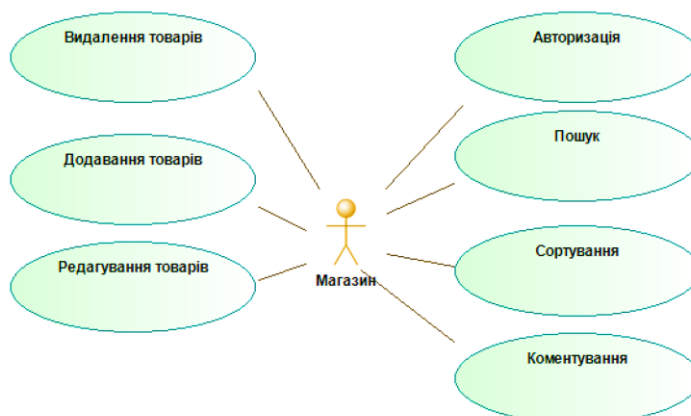


Рис. 1.3. Діаграма варіантів використання «Магазин»

Джерело: сформовано автором

Адміністратор - людина, яка може редагувати різноманітну інформацію на сайті (рис. 1.4);

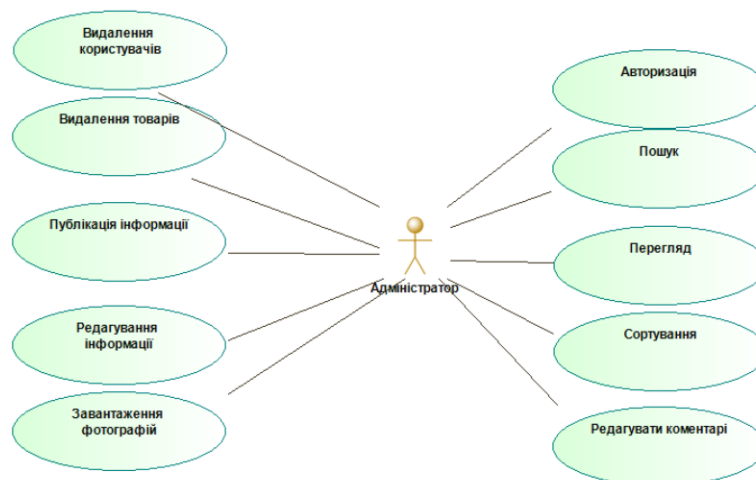


Рис.1.4. Діаграма варіантів використання «Адміністратор»

*Джерело: сформовано автором*

Технічна підтримка - людина, яка вибирає різноманітні функції, що забезпечують коректну роботу та регулярне оновлення інтернет-ресурсу, починаючи з технічної підтримки сайту та завершуючи його візуальним та текстовим доповненням (рис. 1.5).

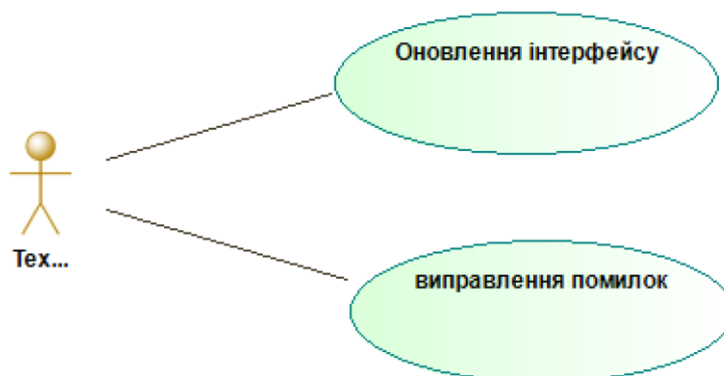


Рис.1.5. Діаграма варіантів використання «Технічна підтримка»

*Джерело: сформовано автором*

Для швидкої та ефективної розробки програмного продукту з мінімальними витратами необхідно застосувати правильний напрямок роботи, використовуючи правильні інструменти та залучаючи робочу силу, а також продумати процес розробки програми, щоб впоратися з поставленим завданням із витратами на системи життєвого циклу.

### 1.5.3. Побудова діаграм послідовності сайту

Діаграма взаємодії використовується для відображення життєвого циклу об'єкта на єдиній тимчасовій осі, а також взаємодії дійових осіб у межах певного прецеденту в інформаційній системі. Основні елементи діаграми включають прямокутники з назвами об'єктів, вертикальні лінії життя, що відображають плинний час, прямокутники, що відображають діяльність об'єкта або виконання ним певної функції, і стрілки, що відображають обмін сигналами або повідомленнями між об'єктами. Діаграма послідовності є одним з інструментів мови UML для моделювання системи.

На діаграмі WEB-сайту можна описати послідовність взаємодії різних блоків сайту, що реалізує різні функції. Для побудови діаграми взаємодії сайту необхідні програми, які блоки взаємодіють між собою, і яку послідовність цієї взаємодії можна виділити.

Опис прецеденту «читання новин». Основний виконавець: користувач. Передумови немає. *Основний потік*: 1) користувач обирає тему; 2) користувач шукає потрібним йому за темою новину; 3) користувач залишає коментар; 4) додаток записує результат у базу.

На даній діаграмі об'єкти розташовуються зліва направо (додаток А).

Опис прецеденту редагування бази даних (додаток Б). Основний виконавець прецеденту адміністратор. Передумови: адміністратор авторизований. *Основний потік*: 1) адміністратор редагує базу даних; 2) додаток надсилає запит до бази даних; 3) додаток обробляє запит, якщо запит невдалий, додаток повертає повідомлення про помилку; 4) додаток повертає результат.

Опис прецеденту реєстрації користувачів (додаток В), який має відвідувача як головного виконавця. Прецедент може бути виконаний лише якщо відвідувач не авторизований. *Основний потік* прецеденту складається з наступних кроків: 1) відвідувач відкриває форму реєстрації; 2) відвідувач вводить дані в відповідні поля; 3) додаток перевіряє введені дані на коректність, і якщо вони не є правильними, то додаток виведе повідомлення

про помилку; 4) додаток виконує запит до бази даних для перевірки існуючих даних; 5) база даних перевіряє введені дані і повертає результат. Якщо введені дані вже існують в базі даних, то додаток виведе повідомлення про помилку; б) відвідувач підтверджує свою реєстрацію; 7) відвідувач закриває форму реєстрації.

На основі наданої інформації можна зробити висновок, що діаграма послідовності є важливим інструментом для моделювання життєвого циклу об'єктів та взаємодії між ними в рамках певного прецеденту. Діаграма послідовності складається з позначень об'єктів, вертикальних «ліній життя», прямокутників, які відображають діяльність об'єктів та стрілок, які відображають обмін сигналами або повідомленнями між об'єктами.

Крім того, було наведено опис прецеденту реєстрації користувачів, який включає основний потік взаємодії між відвідувачем та додатком, що включає в себе проведення валідації введених даних та перевірку їх наявності в базі даних.

## РОЗДІЛ 2

### РОЗРОБКА БАЗИ ДАНИХ

#### 2.1 Опис предметної області

Інформаційна база даних торгового підприємства повинна містити дані про замовників заходів, проведені заходи, їх реєстрацію, місце проведення, співробітників служби підтримки та організації заходів, клієнтів, які реєструються, товари і магазини.

Для проведення дослідження було обрано розробку WEB-сайту торгового підприємства з продажу біогазових установок

На даному етапі розробки реалізована підтримка торгового підприємства з продажу біогазових установок. Для побудови інформаційно-логічної (концептуальної) моделі була використана методика, описана детально в роботі [8]. В результаті були визначені інформаційні об'єкти бази даних та зв'язки між ними. Опис і ключові реквізити відображені у таблиці 2.1.

*Таблиця 2.1*

#### Угруповання об'єктів з інформаційних елементів

Об'єкт	Інформаційні елементи
Біогазова установка малої потужності	Технічні характеристики, Принцип роботи, Вартість, Детальний опис
Біогазова установка середньої потужності	Технічні характеристики, Принцип роботи, Вартість, Детальний опис, Рекомендації щодо встановлення та експлуатації
Біогазова установка великої потужності	Технічні характеристики, Принцип роботи, Вартість, Детальний опис, Рекомендації щодо установки та експлуатації, Фото та відео матеріали
Додаткові послуги	Послуги з монтажу та налагодження, Технічна підтримка, Консультації з експлуатації, Гарантійне обслуговування

*Джерело: сформовано автором*

З таблиці групування об'єктів з інформаційних елементів з продажу біогазових установок можна зробити наступні висновки:

1. Біогазові установки можна встановити на дві основні категорії: розділи

землі та підземні.

2. Об'єкти можна групувати за їхньою потужністю: до 50 кВт, від 50 до 250 кВт, від 250 до 500 кВт та понад 500 кВт.

3. Інформаційні елементи можна розділити на три групи: технічні характеристики біогазових установок, інформація про ціни та інформація про процес продажу.

4. Найбільші інформаційні елементи для клієнтів є технічні характеристики біогазових установок та інформація про ціни.

5. Для забезпечення ефективної роботи сайту необхідно розмістити достатню кількість інформаційних елементів з усіх трьох груп.

На рис. 2.1 представлена інформаційно-логічна модель розглянутої БД, побудована відповідно до виявлених інформаційними об'єктами і зв'язками між ними.

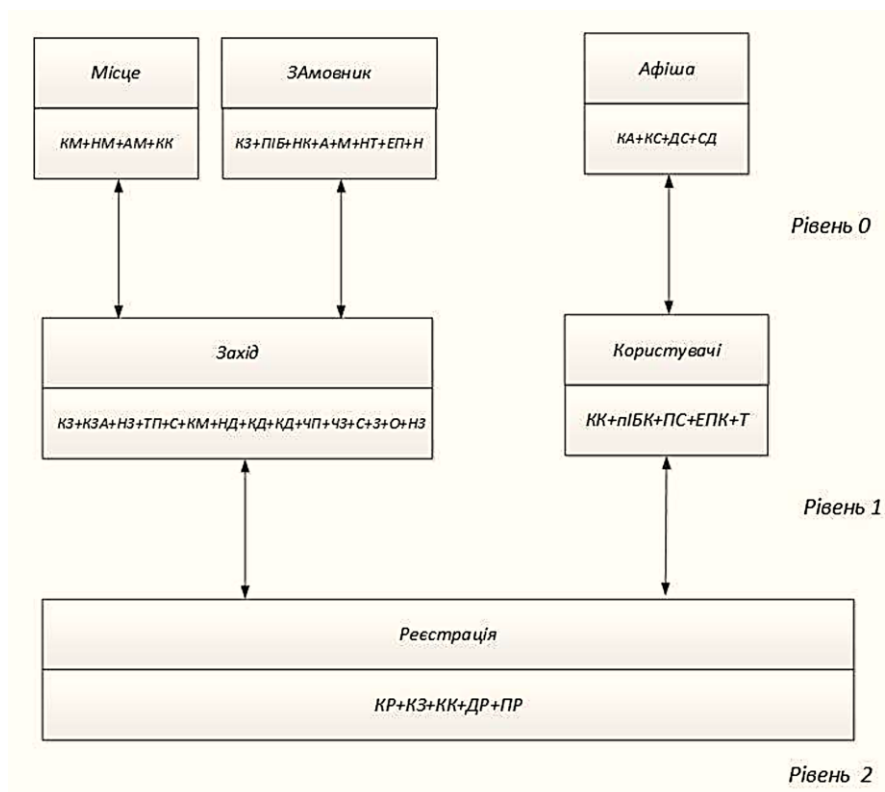


Рис. 2.1. Концептуальна модель предметної області

*Джерело: сформовано автором*

Отже, концептуальна модель предметної області відображає візуальне бачення сутностей, атрибутів та зв'язків між ними в предметній області, яке

дозволяє розуміти основні принципи та структуру системи, яка буде розроблятися.

Це базова модель, яка передбачає встановлення сутностей, атрибутів та зв'язків між ними, а також способи взаємодії між сутностями та їх атрибутами. Концептуальна модель є важливим інструментом для проектування систем баз даних, оскільки вона дозволяє спроектувати структуру бази даних з урахуванням потреб користувачів та вимог системи.

## 2.2 Реляційна модель бази даних

Реляційна модель бази даних є одним із видів логічної моделі даних, які використовуються для зберігання даних у базах даних. У реляційній моделі база даних складається з таблиці, яка складається з рядків (записів) та стовпців (поля). Кожен запис у таблиці містить дані про певний об'єкт, а кожен стовпець являє собою окремий атрибут цього об'єкта.

Ключовим елементом реляційної моделі є відношення між таблицями, які застосовуються за допомогою зв'язків між таблицями. Стандартні зв'язки запускаються за допомогою зовнішніх ключів, які пов'язують записи в одній таблиці із записами в іншій таблиці.

Однією з переваг реляційної моделі бази даних є її простота та легкість використання, а також можливість використання стандартної мови запитів SQL для взаємодії з базою даних. Однак реляційна модель не завжди є найкращим варіантом для всіх типів даних, тому інші види логічних моделей даних, які використовуються в певних випадках.

На рисунку 2.2 наведена логічна (реляційна) модель, а опис таблиць можна знайти в табл.2.3.

Фізичні моделі баз даних визначають, як дані будуть зберігатись і доступні на фізичному рівні. Раніше використовувалися файлові структури та системи управління файлами (СУФ), які були частинами операційних систем. СКБД надавала надбудову над цими файловими моделями, щоб організувати всі файли в єдину сукупність і керувати ними централізовано.



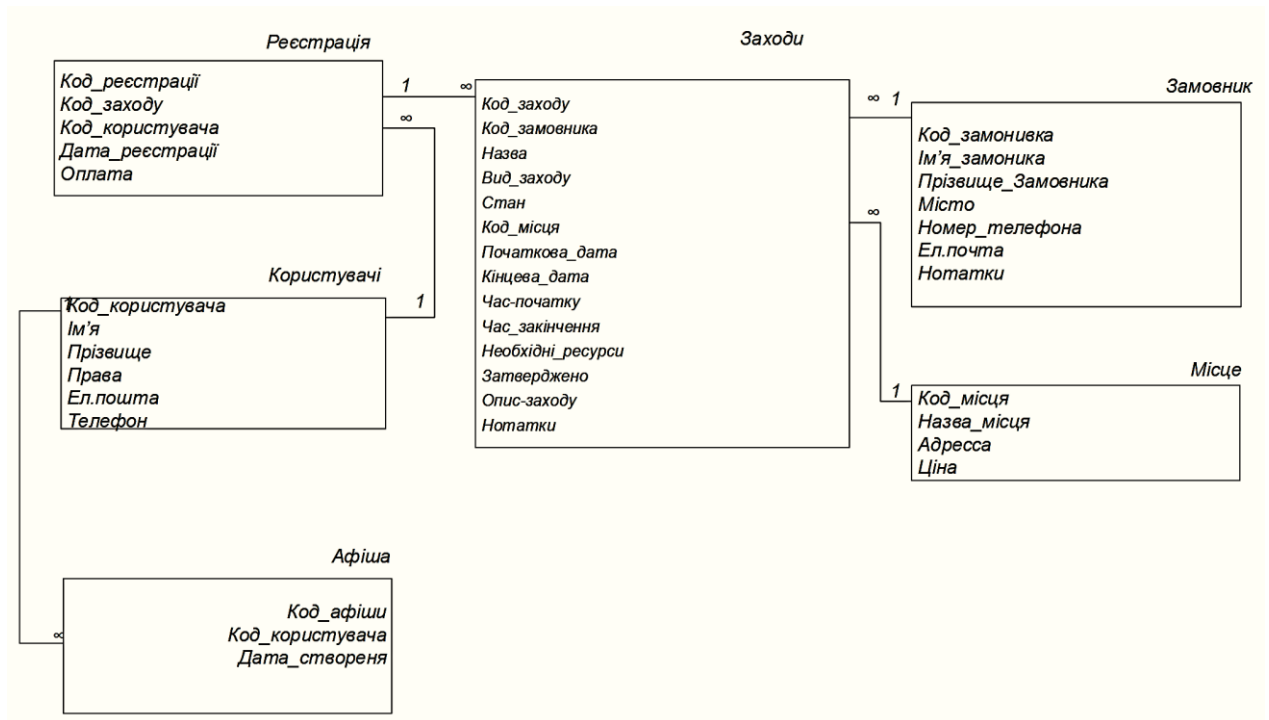


Рис. 2.2. Логічна (реляційна) модель бази даних

Джерело: сформовано автором

Проте безпосередній доступ до файлів відбувався на рівні файлових команд, які використовувалися СКБД для маніпулювання даними в файлах, які були частинами однієї або декількох баз даних.

Таблиця 2.3

## Опис таблиць бази даних

Назва таблиці	Опис полів/стовпців
Клієнти	ID, Прізвище, Ім'я, Адреса, Телефон, Електронна пошта
Замовлення	ID, ID клієнта (зв'язок з таблицею Клієнти), Дата замовлення, Статус замовлення
Установка	ID, Назва, Опис, Ціна, Кількість на складі
Замовлені товари	ID замовлення (зв'язок з таблицею Замовлення), ID продукту (зв'язок з таблицею Продукти), Кількість замовлень

Джерело: сформовано автором

Ці таблиці складають базу даних для підприємства з продажу біогазових установок. Таблиця «Клієнти» містить основну інформацію про клієнтів, що розміщуються на замовлення. Таблиця «Замовлення» містить інформацію про замовлення клієнтів, включаючи дату замовлення та статус замовлення.

Таблиця «Продукти» містить інформацію про товари, що продаються на підприємстві, включаючи назву, опис, ціну та кількість на складі. Таблиця «Замовлені продукти» містить інформацію про товари, які були замовлені клієнтами, включаючи кількість товару в замовленні.

### 2.3 Розробка бази даних

База даних, що відповідає логічній моделі предметної області, була розроблена з використанням СУБД MySQL та інструментів MySQL Workbench. Для створення таблиць спочатку необхідно натиснути кнопку «Додати таблицю», що призведе до відображення форми, яка зображена на рис. 2.3.

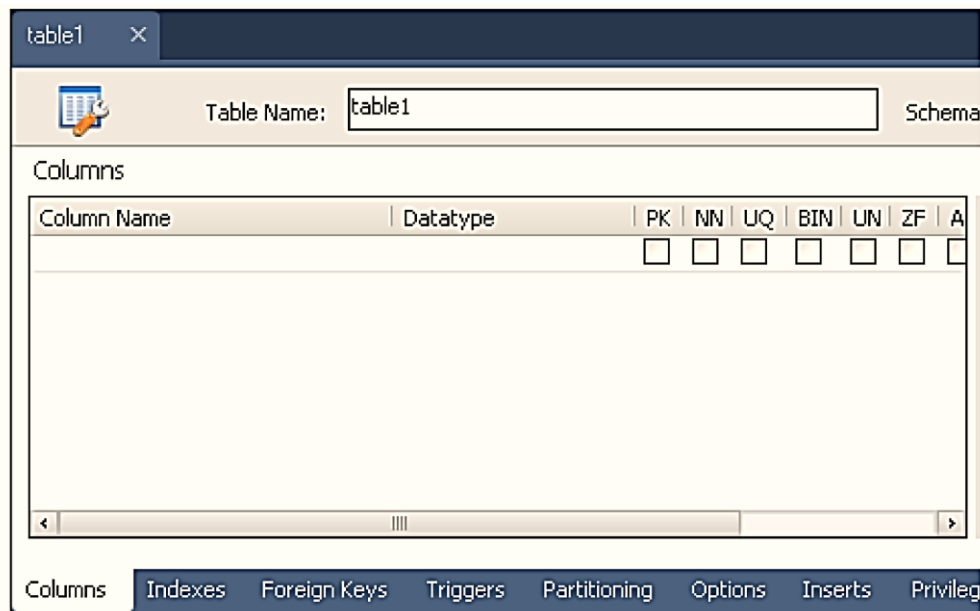


Рис. 2.3. Створення нової таблиці

*Джерело: сформовано автором*

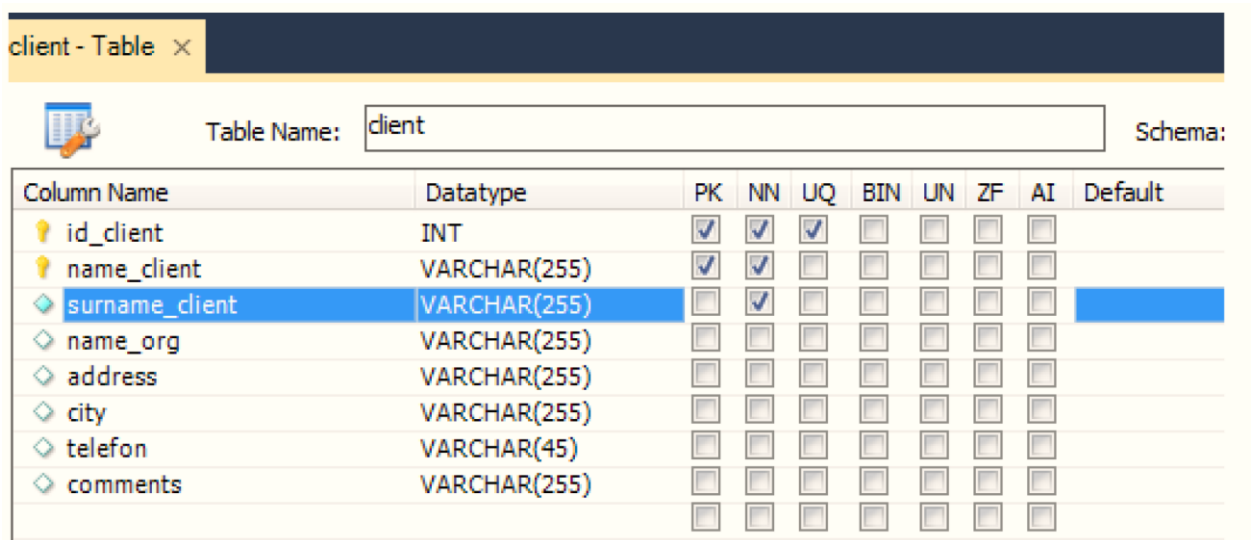
Спочатку необхідно створити таблицю клієнта для зберігання даних про клієнтів організації. Для цього поля «Назва таблиці» вказуємо назву таблиці, а в розділі «Стовпці» створюємо поля таблиці з такими параметрами:

- Поле «id\_client» містить унікальний номер клієнта і має властивості: «Auto Increment», «Not Null», «Primary key» і «Unique». Тип даних - ціле число (ціле).
- Друге і третє поля містять інформацію про П.І.Б. клієнта, і має

властивості: «Not Null», «Primary key», а тип даних - строковий тип «VARCHAR» з довжиною в 255 символів.

- Четверте поле «name\_org» містить назву організації також і має довжину в 255 символів.
- Поля «address», «city», «e\_mail», «comments» і «telefon» складаються з адреси, міста, електронної адреси, коментаря та телефону відповідно, і мають строковий тип «VARCHAR» довжиною в 255 символів, за допомогою останнього поля «type», для якого достатньо 45 символів.

Після виконання цих дій форма з іменем таблиці «users» виглядає так, як показано на рис. 2.4.



Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
id_client	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
name_client	VARCHAR(255)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
surname_client	VARCHAR(255)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
name_org	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
address	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
city	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
telefon	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
comments	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Рис. 2.4. Форма таблиці «client»

*Джерело: сформовано автором*

На діаграмі з'явиться таблиця client с полями і індексами (рис.2.5).

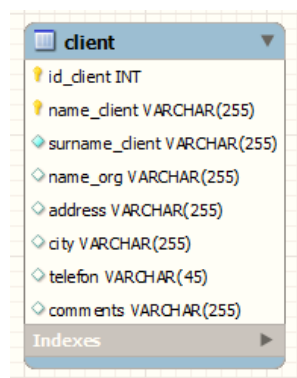


Рис. 2.5. Таблиця «client»

*Джерело: сформовано автором*

Ми створюємо таблицю `place`, яка містить інформацію про місце проведення заходу, аналогічним методом, як у випадку таблиці `client`. Поля таблиці містять наступну інформацію: `id` - унікальний номер місця, `name_place` - назва місця (закладу), `address` - адреса місця, `ticket` - кількість місць в даному закладі (рис. 2.6).

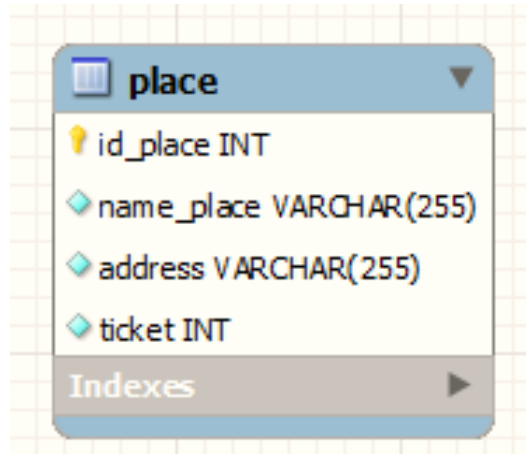


Рис. 2.6. Таблиця «place»

*Джерело: сформовано автором*

Далі за допомогою відомого методу ми створюємо таблицю `register`, що містить інформацію про реєстрацію на заходи. Поле `id_register` містить код реєстрації, `id_party` містить код заходу, `id_worker` містить код співробітника, який відповідальний за захід, `date` містить дату реєстрації, а `pay` містить інформацію про оплату (рис. 2.7).

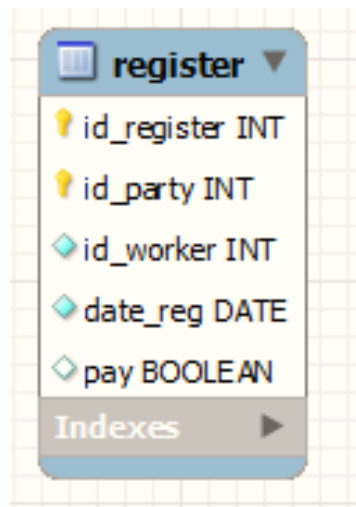


Рис.2.7. Таблиця «register»

*Джерело: сформовано автором*

Наступним кроком є створення таблиці `worker` для зберігання даних про працівників підприємства. Поле `id_worker` містить унікальний номер співробітника, поля `name` та `surname` містять інформацію про його ім'я та прізвище, відповідно. Поле `post` вказує на посаду співробітника, а поля `e-mail` та `telefon` містять контактну інформацію про нього (рис. 2.8).

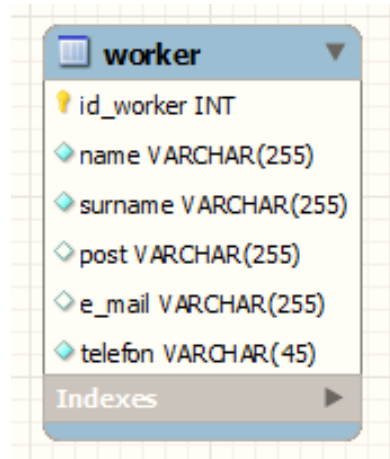


Рис.2.8. Таблиця «worker»

*Джерело: сформовано автором*

Далі ми створюємо таблицю, яка містить інформацію про розроблені афіші з назвою «афіша». Вона має унікальне, ненульове та ключове поле `id_afisha`, яке автоматично збільшується для цілочисельного типу. Також таблиця містить поле `id_worker` з кодом співробітника, який відповідає за цю афішу, та інформацію про час її створення та актуальності - дату та час (рис. 2.9).

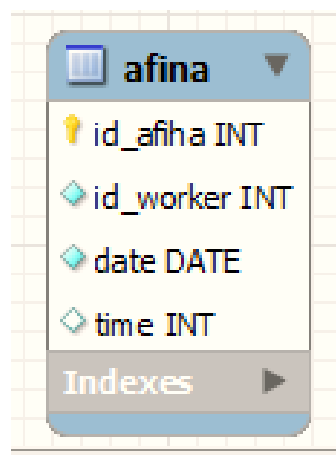


Рис.2.9. Таблиця афіша

*Джерело: сформовано автором*

Таблиця events містить інформацію про різноманітні заходи, які проводить підприємство. Ключовим полем є id, яке є унікальним цілим числом з автоінкрементом і може бути нульовим. Поле id\_client містить інформацію про замовника заходу, поле name - назву заходу, а тип поля - тип заходу (наприклад, концерт, прем'єра, вечірка тощо). Поле state вказує на стан даного заходу, а поле id\_place містить код місця проведення. Дата початку і закінчення заходів зберігається в полях start\_date і end\_date відповідно. Час початку і завершення заходу зберігається в полях time\_start і time\_end. Поле ресурси містить додаткові ресурси для проведення заходу, а містить затверджену інформацію про затвердження та погодження з адміністрацією. Опис заходу зберігається в полі опису, а замітки - в полі коментарів. (рис. 2.10)

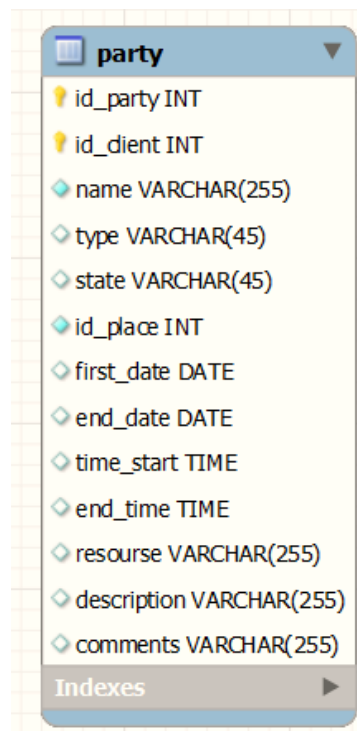


Рис.2.10. Таблиця party

*Джерело: сформовано автором*

Тепер необхідно зв'язати таблиці бази даних, яку ми розробили, використовуючи вже практично розроблені основні поля цілочисельного типу. Для зв'язку двох таблиць потрібно двічі клацнути лівою кнопкою миші на діаграму таблиці і вибрати вкладку «Foreign keys» (зовнішні ключі). Потім в поле «Foreign key name» вводимо унікальне ім'я зовнішнього ключа, двічі

клацаємо по вкладці «Referenced table» і вибираємо другу таблицю. Потім в формі, яка розташована праворуч, вибираємо поле «type\_id» в розкритому списку «id».

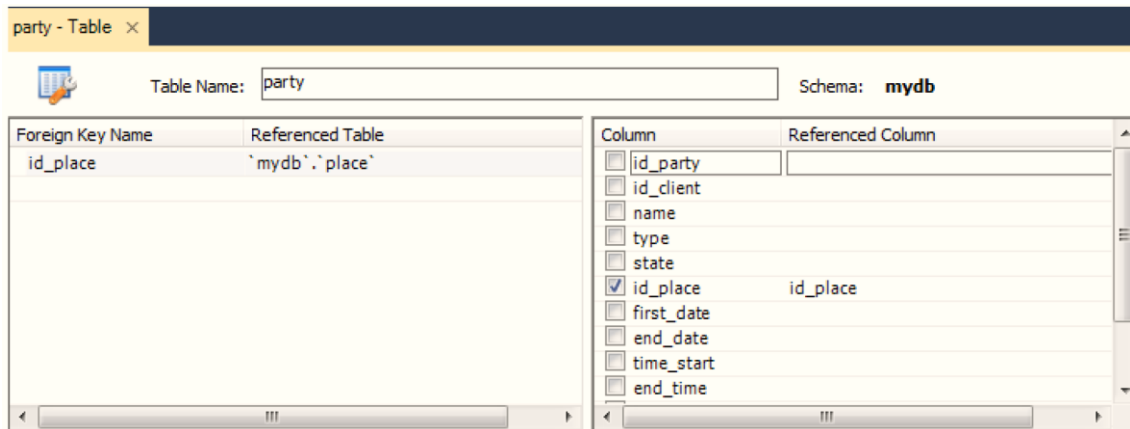


Рис.2.11. Властивості зв'язку таблиць

*Джерело: сформовано автором*

Отже, в результаті обидва поля таблиці будуть зв'язані. Далі потрібно визначити тип відносин між таблицями. Це можна зробити, відкривши вікно зв'язку між таблицями та вибравши вкладку «Foreign Key». В розділі «Cardinality» потрібно вибрати тип зв'язку «один до багатьох» та закрити вікно. Як результат, на діаграмі буде відображено зв'язок між таблицями (рис.2.12).

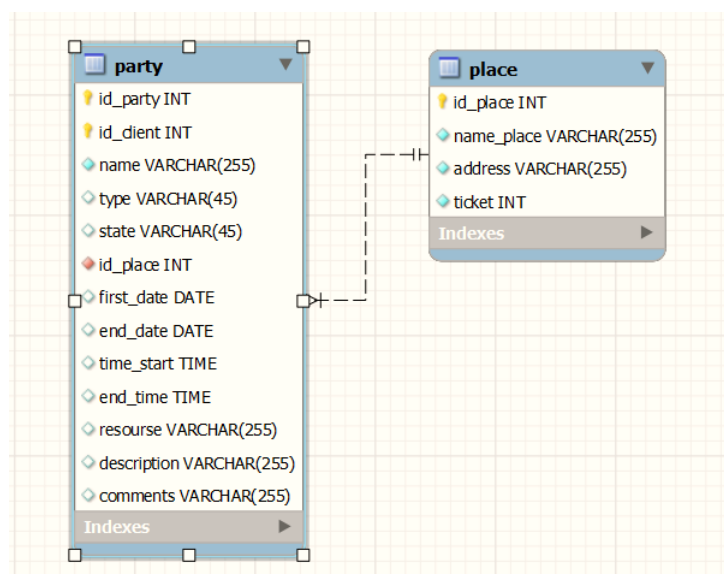


Рис.2.12. Зв'язок таблиць

*Джерело: сформовано автором*

Далі ми зв'яжемо всі ключові поля в таблицях аналогічним чином, щоб вони мали логічний зв'язок між собою. Після цього ми переконуємося, що наша база даних відповідає третій нормальній формі, яка описує якість зв'язку в реляційній моделі даних з точки зору надмірності, що може призвести до помилкових результатів вибірки або зміни даних.

У реляційній моделі відношення завжди знаходиться в першій нормальній формі, але різні таблиці можуть не бути правильними уявленнями відносин і не перебувати в першій нормальній формі. Відношення знаходиться в другій нормальній формі, якщо воно знаходиться в першій нормальній формі і кожен не ключовий атрибут неприводимого залежить від потенційного ключа. База даних буде знаходитися в третій нормальній формі, якщо вона приведена до другої нормальної форми і кожен не ключовий стовпець незалежний один від одного.

Таким чином, наша база даних знаходиться в третій нормальній формі, оскільки кожен не ключовий стовпець незалежний один від одного, це можна побачити на діаграмі нашої бази даних на рисунку 2.13.

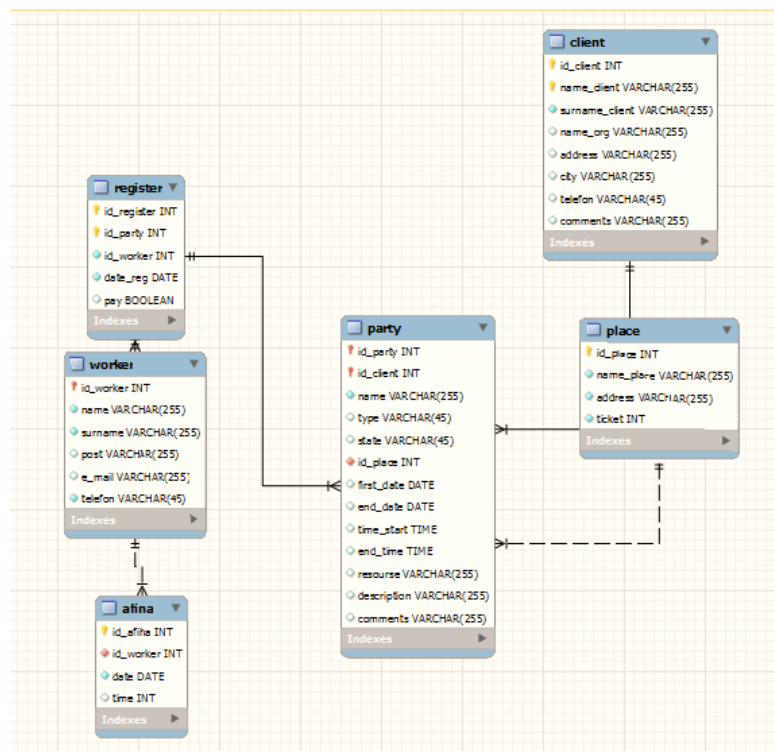


Рис.2.13. Структура бази даних

Джерело: сформовано автором



Під час розробки бази даних були виконані наступні етапи: аналіз вимог користувачів, проектування структури бази даних та розробка схем, впровадження бази даних на СУБД та перевірка її функціональності. Було створено реляційну базу даних, яка включає 4 таблиці: книги, автори, видавництва та замовлення. При проектуванні були використані відповідні ключі для зв'язку між таблицями та була перевірена відповідність бази даних третьої нормальної форми. У результаті була розроблена функціональна база даних, яка може бути використана для зберігання та організації інформації про книги та замовлення користувачів. Вона може бути доповнена новими функціями, такими як зберігання даних про клієнтів або статистику продажів.

Розробка бази даних захисту, організувати та використовувати інформацію в більш ефективний спосіб. Вона забезпечує збереження даних у структурованому форматі, що дозволяє легко здійснювати пошук, фільтрацію та сортування. База даних є важливою складовою будь-якої системи управління даними та може бути використана в різних галузях, включаючи бізнес, освіту та науку.

## РОЗДІЛ 3

### РОЗРОБКА WEB САЙТУ ТОРГОВОГО ПІДПРИЄМСТВА З ПРОДАЖУ БІОГАЗОВИХ УСТАНОВОК

#### 3.1 Структурна схема сайту

Перед початком написання коду та створення WEB-сторінок для сайту необхідно розробити його структуру, визначити кількість сторінок та систему посилань між ними. Для цього потрібно скласти схему сайту та його структурну діаграму, де будуть показані сторінки та їхні зв'язки. Також важливо передбачити можливість подальшого розширення сайту та додавання нових розділів та посилань.

Структурна діаграма сайту також визначає методи навігації, які будуть використовуватися на сайті. Існує три типи навігації: переміщення за посиланнями, використання форми пошуку та доступ через портал до вмісту. Не обов'язково використовувати всі три методи, зазвичай найпоширенішим є перший.

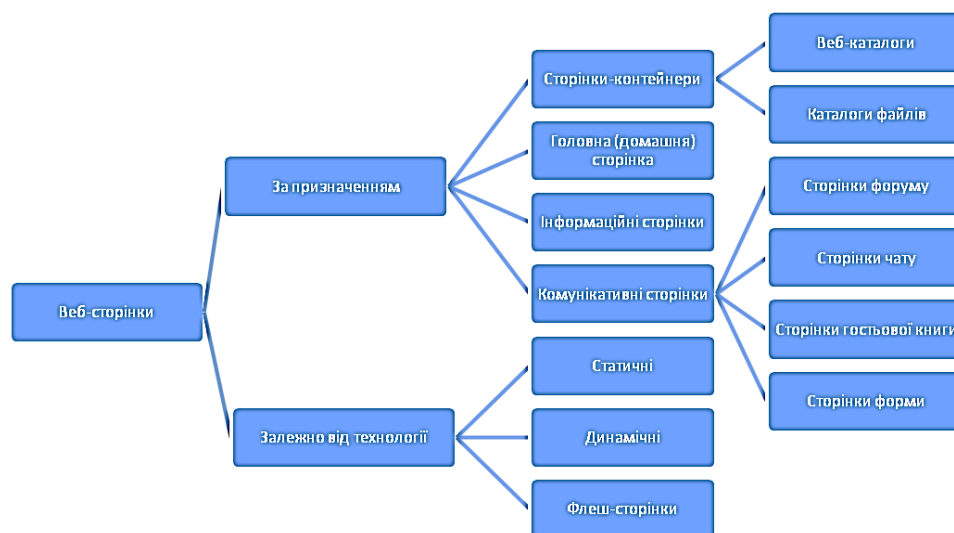


Рис.3.1. Структурна схема WEB сайту підприємства із продажу біогазових установок

*Джерело: сформовано автором*

Далі розглянемо детальніше елементи першого рівня сайту.

- Головна сторінка є першою сторінкою, яку відвідувач бачить при

заходженні на сайт і є необхідною для всіх видів WEB-ресурсів - від онлайн-магазинів до форумів. Ваша основна функція працює у зверненні уваги цільової аудиторії та перетворенні користувачів на реальних клієнтів.

- Меню є найбільш розширеною навігаційною системою і відображає вміст до книги. Воно складається з коротких текстових посилань на певні сторінки сайту і найкраще розташовувати окремі з них на розділах, які найбільш активні на сайті. Для великих сайтів, які не входять в Опублікувати всі розділи назви в меню, краще додати пошукову форму, яка дозволить користувачу знайти кілька текстів, у яких зустрічалося шукане слово, і більше ознайомитися з цікавою інформацією. Однак недоліком пошуку є неналежне сортування та збільшення документів, що створює шукане слово, що ускладнює фільтр та вивчення інформації.

- Рубрики є числом елементів навігації на будь-якому сайті. Дуже часто меню рубрик стає головним меню, тоді як основне меню відходить в тінь. Хорошим прикладом є сайт, де головне меню складається зі сторінок і виконує другорядну роль, тоді як рубрик меню забезпечує основну навігацію за вмістом сайту.

- Мітки класифікують вміст сайту за частотою використання певних тем. Вага мітки збільшується зі збільшенням багатьох записів, яким ця мітка була присвоєна. Використання міток дозволяє швидко знайти статті на певну тему, а також побачити, які теми є найбільш популярними серед користувачів. Однак, якщо мітки неправильно використані або створені неоднаково, це може призвести до зниження якості класифікації і, відповідно, до складнощів з пошуком необхідного матеріалу. Також важливо мати можливість фільтрування матеріалу за мітками для зручності користувачів.

## 3.2 Керівництво програміста

### 3.2.1. Налаштування сервера

Ми вибрали XAMPP як дистрибутив для встановлення, який включає WEB-сервер, а також MySQL, Apache, FileZilla.

Після завантаження дистрибутива XAMPP запускаємо процес встановлення (див. рис. 3.2).

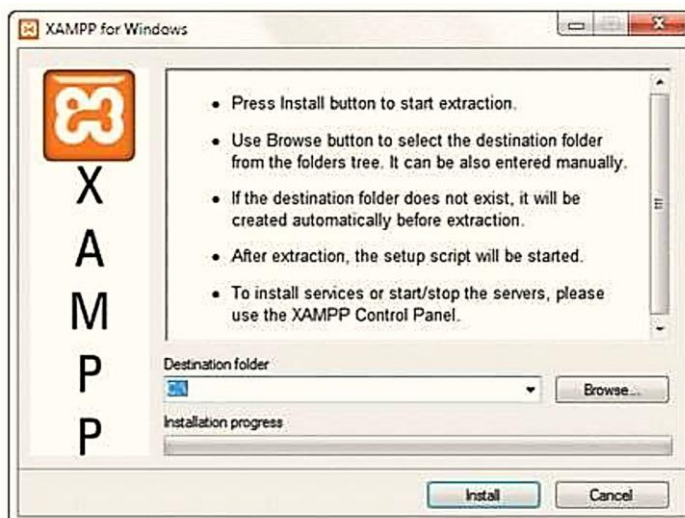


Рис. 3.2. Установка WEB-сервера

*Джерело: сформовано автором*

За замовчуванням шлях для установки WEB сервера вказано диск C: \. Як видно на рис. 3.3 відбувається вилучення файлів.



Рис. 3.3. Вилучення файлів

*Джерело: сформовано автором*

Після завершення установки, дивимося в трейм, там знаходиться значок XAMPP натискаємо на нього і перед нами з'являється панель управління XAMPP (рис. 3.4). Встановлюємо галочки на модулях Apache, MySQL і звертаємо вікно управління XAMPP (рис.3.5).

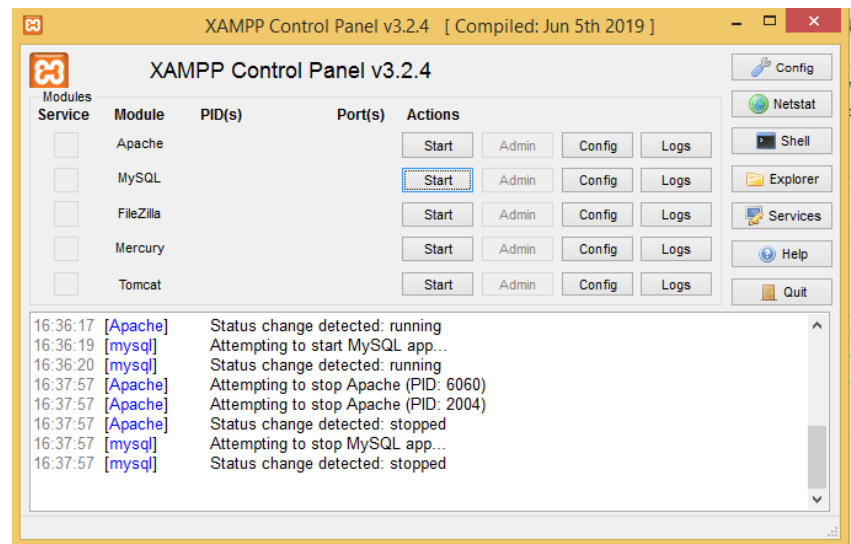


Рис. 3.4. Панель управління WEB-сервера

*Джерело: сформовано автором*

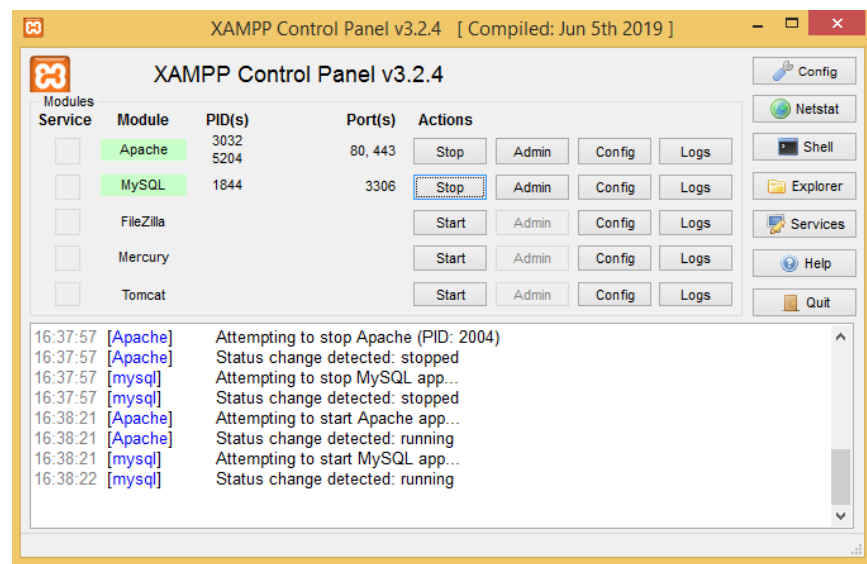


Рис. 3.5. Запуск WEB-сервера

*Джерело: сформовано автором*

Після запуску установки дистрибутиву XAMPP, користувач повинен відзначити необхідні компоненти, які будуть встановлені на його комп'ютері. Зазвичай, встановлюються всі доступні компоненти, так як це забезпечує повну функціональність WEB-сервера. Після вибору компонентів, користувач повинен вказати шлях до папки, де буде встановлено XAMPP. Зазвичай, це папка C:\xampp на системах Windows та /opt/lampp на системах Linux.

Після встановлення XAMPP, користувач повинен запуснути Apache та

MySQL, щоб запустити WEB-сервер та базу даних. Крім того, користувач може запустити FileZilla для налаштування FTP-сервера, який дозволяє здійснювати передачу файлів між сервером та клієнтом. Після запуску всіх необхідних компонентів, користувач може перевірити роботу WEB-сервера за допомогою WEB-браузера, відкривши адресу localhost в адресному рядку браузера.

Перевірка установки XAMPP. Запускаємо браузер і вводимо наступний URL- адресу: <http://localhost>. Відкриється сторінка вітання XAMPP (рис.3.6).



Рис. 3.6. Сторінка вітання XAMPP

*Джерело: сформовано автором*

Після успішної установки XAMPP необхідно переконатися, що всі компоненти дистрибутиву працюють коректно. Для цього можна виконати перевірку шляхом запуску WEB-сервера та MySQL-сервера.

Для запуску WEB-сервера необхідно відкрити браузер та ввести адресу «<http://localhost/>». Якщо все працює коректно, то на екрані з'явиться стартова сторінка XAMPP.

Для запуску MySQL-сервера необхідно відкрити XAMPP Control Panel

та натиснути на кнопку «Start» біля «MySQL». Якщо все працює коректно, то на панелі з'явиться повідомлення про успішний запуск сервера.

Також, можна перевірити роботу FTP-сервера, запустивши FileZilla та спробувати підключитися до нього.

Якщо після запуску будь-якого з компонентів виникають помилки, необхідно перевірити налаштування та логи помилок.

### 3.2.2 CMS WordPress

WordPress - це безкоштовна система управління контентом (CMS), яка дозволяє створювати та публікувати WEB-сайти. Вперше вона була випущена у 2003 році і з тих пір стала однією з найпопулярніших CMS в світі. WordPress базується на мові програмування PHP і використовує базу даних MySQL для зберігання даних.

За допомогою WordPress можна створити різні види WEB-сайтів, включаючи блоги, інтернет-магазини, WEB-журнали, корпоративні сайти, фотогалереї, портфоліо та інші. WordPress має велику кількість безкоштовних та платних тем (шаблонів) і плагінів (додатків), що дозволяє легко налаштовувати зовнішній вигляд та функціональність WEB-сайту.

WordPress має простий та зрозумілий інтерфейс, що дозволяє легко додавати та редагувати контент. Вона також підтримує багатомовність, що дозволяє створювати сайти на різних мовах. Завдяки великій активній спільноті користувачів WordPress, знайти відповіді на свої питання та рішення на проблеми можна досить швидко і легко.

У табл.3.1 наведені можливості WordPress.

Після установки Wordpress копіюємо його в `C:\WEBServers\home\localhost\www`. У файлі `wp-config.php` пишемо назву бази даних, користувача і пароль, в результаті чого отримаємо:

Ім'я бази даних для WordPress:

```
define('DB_NAME', 'database_wordpress');
```

MySQL - Ім'я користувача:

```
define('DB_USER', 'root');
```

MySQL - Пароль бази даних

```
define('DB_PASSWORD', '');
```

MySQL сервер - Іноді необхідно змінити це значення.

```
define('DB_HOST', 'localhost').
```

Далі необхідно створити з такою ж назвою базу даних в phpmyadmin.

Таблиця 3.1

### Можливості WordPress

Можливості WordPress	Опис
Налаштування дизайну	WordPress надає можливість встановлювати теми та розширення, які можуть надати зовнішній вигляд WEB-сайту за допомогою графічного інтерфейсу.
Створення та редагування контенту	WordPress надає зручний інтерфейс для створення, редагування та публікації різноманітного контенту, включаючи статті, фотографії, відео, аудіо, сторінки та інші типи контенту.
Управління користувачами	WordPress надає можливість налаштування прав користувачів і ролей, що дозволяє обмежувати доступ до різних функцій і контенту WEB-сайту.
SEO-оптимізація	WordPress має вбудовану підтримку SEO-оптимізації, а також зовнішнього розширення, що не дозволяє досягти різних аспектів оптимізації для покращення показників WEB-сайту в пошукових системах.
Розширення функціональності	WordPress має велику кількість розширень, які не дозволяють додавати нові функції та можливості WEB-сайту без необхідності програмування.
Безпека та стійкість	WordPress постійно оновлюється та виправляється вразливості, що дозволяє забезпечити високий рівень безпеки та стійкості WEB-сайту.
Спільнота користувачів та розробників	WordPress має спільну велику кількість користувачів та розробників, що дозволяє швидко отримувати відповіді на запитання та підтримку, а також дозволяє розвивати власний WEB-сайт за допомогою розсилки.

*Джерело: сформовано автором*

У директорії з темами wordpress:

```
C:\WEBServers\home\localhost\www\wordpress\wp-content\themes
```

Копіюємо папку з вихідним шаблоном. Для того, щоб wordpress міг використовувати дану тему обов'язково необхідний файл index.php, який є шаблоном майже для всіх сторінок, тобто всі його частини будуть виводиться практично на всіх сторінках за винятком текстів.



Створюємо шаблон в папці:

C:\WEBServers\home\localhost\www\wordpress\wp-content\themes\template, в якому використовуються функції для виведення title десятків заголовків футера.

Наведемо основні функції:

Рядок: `<? Php get_template_part ( 'content', 'page'); ?>` Є php- скриптом, в якому є функція `get_template_part`, що виводить вміст сторінок з адмінки wordpress по її назві. Тобто, переходячи, наприклад, на сторінку `http://localhost/wordpress/main-4/` береться вміст сторінки `main-4`.

Для відображення зображення і флеш необхідно прописати url-адреса теми, наприклад, для css стилів:

```
<Link href = «http://localhost/wordpress/wp-content/themes/template4/style.css» rel = «stylesheet» type = «text / css»>.
```

Файл `content-page.php` здійснює вибірку текстів по сторінці.

Віджети в WordPress грають роль модулів. За допомогою віджетів можна вивести інформацію на сайт. Наприклад: сторінки, категорії (рубрики), хмара тегів, архів записів і т.д. Встановлюються вони разом з плагінами.

Віджети - це блоки, які можна розміщувати в колонках сайту. Вбудовані в WordPress віджети дозволяють додати наступну інформацію в колонках блогу:

- список категорій;
- список сторінок;
- архіви;
- посилання;
- останні повідомлення;
- останні коментарі;
- календар;
- RSS потоки.

Одним з найбільш корисних елементів є текстові блоки. Завдяки текстовому віджету можна вставляти html-код, JavaScript або простий текст.

Це дозволяє використовувати цей віджет для розміщення різноманітних лічильників, рекламних блоків та інших подібних елементів.

Наступним етапом ми внесли власні корективи в стилі WEB-сайту.

### Лістинг 3.1 – Стиль стартової панелі

```
.welcome-panel .welcome-icon:before,
#dashboard_right_now li a:before,
#dashboard_right_now li span:before,
#dashboard_right_now .search-engines-info:before {
    color: #646970;
    font: normal 20px/1 dashicons;
    speak: never;
    display: inline-block;
    padding: 0 10px 0 0;
    position: relative;
    -webkit-font-smoothing: antialiased;
    -moz-osx-font-smoothing: grayscale;
    text-decoration: none !important;
    vertical-align: top;
}
```

### Функції для роботи з категоріями товарів

#### Лістинг 3.2 – Функції для роботи з категоріями товарів

```
function get_categories( $args = '' ) {
    $defaults = array( 'taxonomy' => 'category' );
    $args = wp_parse_args( $args, $defaults );
    $args['taxonomy'] = apply_filters(
        'get_categories_taxonomy',
        $args['taxonomy'], $args );
    if ( isset( $args['type'] ) && 'link' === $args['type'] ) {
        _deprecated_argument(
            __FUNCTION__,
            '3.0.0',
            sprintf( /* translators: 1: «type => link», 2:
                «taxonomy => link_category» */
                __( '%1$s is deprecated. Use %2$s instead.' ),
                '<code>type => link</code>',
            )
        );
    }
}
```

```

        '<code>taxonomy => link_category</code>' )
    );
    $args['taxonomy'] = 'link_category'; }
$categories = get_terms( $args );
if ( is_wp_error( $categories ) ) {
    $categories = array();
} else {
    $categories = (array) $categories;
    foreach ( array_keys( $categories ) as $k ) {
        _make_cat_compat( $categories[ $k ] );
    }
}

```

Клас `php` для збереження коментарів, сутності класу приведені на лістингу 3.3. Для даного класу важливо берегти інформацію про автора коментаря, його персональні данні, час, коли залиши коментар, ідентифікатор посту.

### Лістинг 3.3 – Робота з коментарями

```

public $comment_ID;
public $comment_post_ID = 0;
public $comment_author = '';
public $comment_author_email = '';
public $comment_author_url = '';
public $comment_author_IP = '';
public $comment_date = '0000-00-00 00:00:00';
public $comment_date_gmt = '0000-00-00 00:00:00';
public $comment_content;
public $comment_karma = 0;
public $comment_approved = '1';
public $comment_agent = '';
public $comment_type = 'comment';
public $comment_parent = 0;
public $user_id = 0;
protected $children;
protected $populated_children = false;
protected $post_fields = array(
    'post_author', 'post_date', 'post_date_gmt', 'post_content',
    'post_title', 'post_excerpt',
    'post_status', 'comment_status', 'ping_status', 'post_name',
    'to_ping', 'pinged', 'post_modified', 'post_modified_gmt',
    'post_content_filtered', 'post_parent', 'guid', 'menu_order',
    'post_type', 'post_mime_type', 'comment_count' );

```

Функції класу коментарів наведено на лістингу 3.4.

#### Лістинг 3.4 – Функції роботи з коментарями

```
public static function get_instance( $id ) {
    global $wpdb;

    $comment_id = (int) $id;
    if ( ! $comment_id )
        { return false; }
    $_comment = wp_cache_get(

    $comment_id, 'comment' );

    if ( ! $_comment ) {
        $_comment = $wpdb->get_row( $wpdb->prepare(
        «SELECT * FROM $wpdb->comments WHERE comment_ID = %d LIMIT
1», $comment_id ) );
        if ( ! $_comment ) {
            return false; }

        wp_cache_add( $_comment->comment_ID, $_comment,
        'comment' ); }
    return new WP_Comment( $_comment )
}
```

Функції класу створення нового допису наведено на лістингу 3.5.

#### Лістинг 3.5 – Файл створення ного допису

```

$content = apply_filters( 'wp_mail_original_content',
$content ); if ( false !== stripslashes(
$content_transfer_encoding,
'quoted-printable' ) ) {
    $content = quoted_printable_decode( $content ); }
    if ( function_exists( 'iconv' ) && ! empty( $charset )
) { $content = iconv( $charset, get_option(
'blog_charset'
), $content );}
    $content = explode( $phone_delim, $content );
    $content = empty( $content[1] ) ? $content[0] :
$content[1];
    $content = trim( $content );
    $post_content = apply_filters( 'phone_content',
$content );
    $post_title = xmlrpc_getposttitle( $content );
}
```

```

        if ( `` === trim( $post_title
    ) ) { $post_title = $subject;
    }
    $post_category = array( get_option(
'default_email_category' ) );
    $post_data = compact( 'post_content', 'post_title',
'post_date', 'post_date_gmt', 'post_author', 'post_category',
'post_status' );
    $post_data = wp_slash( $post_data );
    $post_ID = wp_insert_post( $post_data );
    if ( is_wp_error( $post_ID ) ) {
        echo «\n» . $post_ID->get_error_message(); }
    if ( empty( $post_ID ) ) {
        continue; }
    do_action( 'publish_phone', $post_ID ); echo
«\n<p><strong>« . __( 'Author:' ) . '</strong> ' . esc_html(
$post_author ) . '</p>'; echo «\n<p><strong>« . __( 'Posted
title:' ) . '</strong> '
. esc_html( $post_title ) .
'</p>'; if ( ! $pop3-
>delete( $i ) ) { echo
'<p>' . sprintf(
/* translators: %s: POP3 error. */
__( 'Oops: %s' ),
esc_html( $pop3->ERROR )
) . '</p>';
$pop3->reset();
exit;
} else {
    echo '<p>' . sprintf(
/* translators: %s: The message ID. */
__( 'Mission complete. Message %s deleted.' ),
'<strong>' . $i . '</strong>'
) . '</p>'; }}

```

### 3.3 Керівництво користувача

Графічний інтерфейс користувача WEB-сайту побудований на принципі активного діалогу з користувачем та має зручний та дружній інтерфейс. Цей інтерфейс складається з графічних елементів, що розташовані на ррр-сторінках. Щоб здійснити вибірку даних про заходи, користувач заповнює форми, що складаються з текстових полів та кнопок «вибрати». Параметри запиту вводяться в текстове поле, після чого результати відображаються на екрані. Для взаємодії з графічним інтерфейсом, користувач може

використовувати маніпулятор «миша» та клавіатуру. Щоб увійти на WEB-сайт, користувач вводить адресу в командний рядок браузера, після чого з'являється вікно з головною сторінкою сайту.

Після набору в командному рядку браузера Internet - адреси, `http://localhost/party/index.php` перед користувачем з'являється вікно, зображене на рисунку 3.9 (тут ім'я\_сервера - назва сервера в файлі конфігурації операційної системи, `party` - назва директорії з файлами комп'ютерної системи).



Рис. 3.9. Головна сторінка

*Джерело: сформовано автором*

Також, на сайті доступні сторінки для редагування новин та створення нових користувачів, які відображені на рисунках 3.10 та 3.11 відповідно.

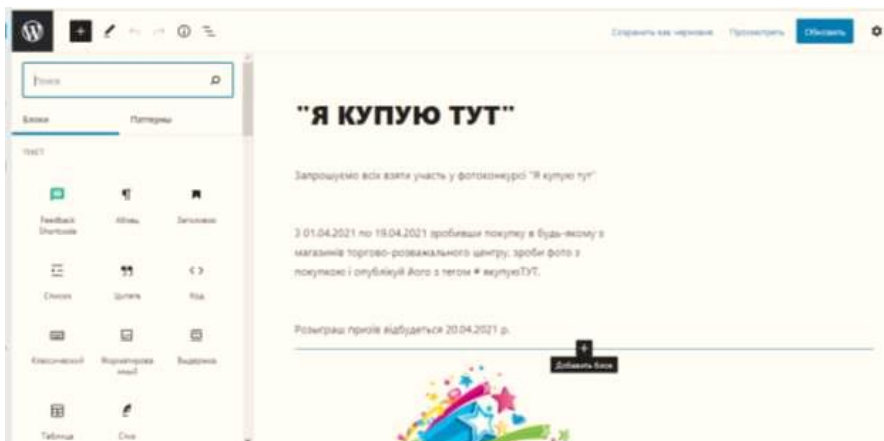


Рис. 3.11. Редагування новини

*Джерело: сформовано автором*

### 3.4. Оптимізація та налаштування сайту

Для визначення сайту відвідуваними клієнтами в Інтернеті було створено семантичне ядро, яке складається з відсортованого за частотою від найбільшого до найменшого списку запитів користувачів на певну тему (табл.3.2).

Таблиця 3.2

#### Ключові слова за запитом «торговий центр»

Ключові Слова
біогаз
біогазова установка
переробка органічних відходів
біомаса
виробництво енергії
біопаливо
метан
анаеробний процес
газопостачання
екологія

Джерело: сформовано автором

Ця таблиця містить ключові слова, пов'язані з біогазовими установками та їхнім використанням для виробництва енергії та забезпечення газопостачання. Серед цих ключових слів - біогаз, біогазові установки, переробка органічних відходів, біомаса, біопаливо, метан, анаеробний процес, газопостачання та екологія. Використання біогазових установок може зменшити залежність від складних геополітичних умов та забезпечити стабільний та екологічно чистий джерело енергії для споживачів.

Після аналізу переліку найпопулярніших ключових фраз можна зробити висновок, що для WEB-сайту торгового підприємства з продажу біогазових установок необхідно вказати місто та адресу розташування на кожній сторінці. Додатково на кожній сторінці необхідно вказати заголовок сторінки, опис та ключові слова, що відповідають тематиці сторінки. Внутрішня структура сайту має бути оптимізована для пошукових систем за допомогою тегів різних рівнів, таких як h1 до h4, де h1 може бути прихованим за допомогою стилю

display: none, щоб пошукові системи могли проаналізувати цей тег, але користувач не зміг його побачити.

В тегу h2 містяться всі назви категорій та розділів, а в тегу h3 - назви типів за категоріями. Теги інших рівнів використовуються за потребою.

Для покращення оптимізації сайту для пошукових систем, у кожному тегу зображення було прописано атрибут alt, який відображається, коли користувач не може завантажити зображення.

Було долучено файл robots.txt, який містить технічну інформацію для пошукових машин, який наведений у лістингу.

#### Лістинг – Файл roboх.txt

```
# Group 1
User-agent: Googlebot
Disallow: /photo/

# Group 2
User-agent: *
Allow: /

Sitemap: http://www.party.su/sitemap.xml
```

Файл robots.txt або індексний файл є звичайним текстовим документом з кодуванням UTF-8, який використовується для протоколів http, https і FTP. Його основна мета - надати рекомендації пошуковим роботам щодо того, які сторінки та файли варто сканувати. Якщо файл містить символи в іншому кодуванні, крім UTF-8, пошукові роботи можна не знати, як їх правильно обробити. Правила, наведені у файлі robots.txt, дійсні тільки для того хоста, протоколу та номера порту, є файлом.



## ВИСНОВКИ

У дипломній роботі було досліджено основні питання, пов'язані з оптимізацією WEB-сайту для пошукових систем. Це різні аспекти оптимізації, включаючи ключові слова, метатеги, внутрішні та зовнішні посилання, вміст, швидкість завантаження, мобільну оптимізацію та інші.

У розділі 1 було проведено аналіз вихідних даних та вимог до WEB-систем, що дозволило забезпечити забезпечення функціональних можливостей для розробки сайту. Також був проведений аналіз сучасних WEB-систем та наведено короткий опис різних WEB-редакторів, які можуть бути використані для розробки сайту. Була розроблена специфікація вимог SRS, яка містить детальний опис функціональних вимог до сайту та побудовано діаграми використання та WEB-сайту сайту.

У розділі 2 було проведено опис предметної області та розроблено реляційну модель бази даних. На основі цього була розроблена база даних, що відповідає вимогам, визначеним у розділі 1.

У розділі 3 було створено структурну схему сайту та надано керівництво програмістом з налаштування сервера. Також було надано керівництво користувача, проведено оптимізацію та налаштування сайту для забезпечення його ефективної роботи.

У цій дипломній роботі було виконано детальний аналіз та дослідження основних питань, пов'язаних з оптимізацією WEB-сайту для пошукових систем. Дані, отримані в результаті дослідження, можуть бути корисними для WEB-розробників, маркетологів та всіх тих, хто прагне покращити показники свого WEB-сайту.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бардус І.О., Лазарєв М.І., Ніценко А.О. Бази даних у схемах (на основі фундаменталізованого підходу): навч. посіб. Харків: Вид-во «Дісаплюс», 2017. 133 с.
2. Вступ до комп'ютерного дизайну / Шикула О.М., Випинівський В.В., Іщеряк С.М., Каргаполов Ю.В., Прокопов С.В., Щербина І.С.. Київ: НШІТ ДУТ, ФОП Гуляева В.М. 2021. 240 с.
3. Герасименко Г.М., Герасименко І.В. Система управління вмістом WEB-сайту навчальний посібник. Київ : НУХТ, 2017. 114 с.
4. Глушаков С.В., Ломотько Д.В. Бази даних. М.: Харків: Фоліо, 2017. 504 с.
5. Гото К., Котлер Е. WEB-редизайн: книга Келли Гото и Эмили Котлер. 2-е изд. СПб: Символ-Плюс, 2006. 416 с.
6. Палермо Д., Богард Д., Хекстер Е., Хінзе М., Скіннер Д. ASP.NET MVC 4 в дії. Створено. 2012. 408 с.
7. Дакетт Д. HTML и CSS. Розробка та дизайн сайтів. М.: Ексмо, 2018. 208 с.
8. Документація MySQL. URL: <https://dev.mysql.com/doc/> (дата звернення: 05.04.2023).
9. Документація PHP. URL: <https://www.php.net/manual/en/> (дата звернення: 05.04.2023).
10. Дронов В.О. Laravel. Швидка розробка сучасних динамічних WEB-сайтів на PHP, MySQL, HTML та CSS. М: БХВ-Петербург, 2018. 349 с
11. Конспект лекцій з дисципліни «WEB-технологіїтаWEB-дизайн» для студентів напряму підготовки 6.050101 «Комп'ютерні науки» Виковий П.Є., Палій І.О., Комар М.П. Тернопіль: ТНЕУ, 2012. 92 с.
12. Косолап А.І. Internet-технології. *HTML та JavaScript*. Дніпропетровськ: Вид-во «Наука та освіта», 2014. 220 с.
13. Орлік С. Введення в програмну інженерію та управління життєвим

циклом програмного забезпечення. Програмна інженерія. Супровід програмного забезпечення. 2021. URL: <http://sorlik.blogspot.com/> (дата звернення: 01.04.2023)

14. Онищенко С.В. WEB-технології: навч.-метод, комплекс. Бердянськ: «БДПУ», 2016. 500 с.

15. Пасічник О.Г., Пасічник О.В., Стеценко І.В. Основи WEB-дизайну. [Навч. посіб.]. К.: Вид. група ВНУ. 2009. 336 с.

16. Побудова WEB-сайтів. В.А. Фрідман, А.В. Александров, Г.Г.Сергєєв, С.П. Костін. М.: Тріумф, 2011. 288 с.

17. Цеслів О.В. WEB-програмування : навч. посібник; М-во освіти і науки, молоді та спорту України, Нац. техн. ун-т України “Київ. політехн. ін-т”. Київ : НТУУ “КПІ”, 2011. 296, с.

18. Шмідт Я. Нова мережа: ознаки, практики і наслідки WEB 2.0 = Das Neue Nets Merkmale, Praktiken und Folgen des WEB 2.0 : посібник для вузів / Ян Шмідт ; [пер. з нім. В. Климченко ; за заг. ред. В. Іванова]. Київ : Академія Української Преси, Центр Вільної Преси, 2013. 283с

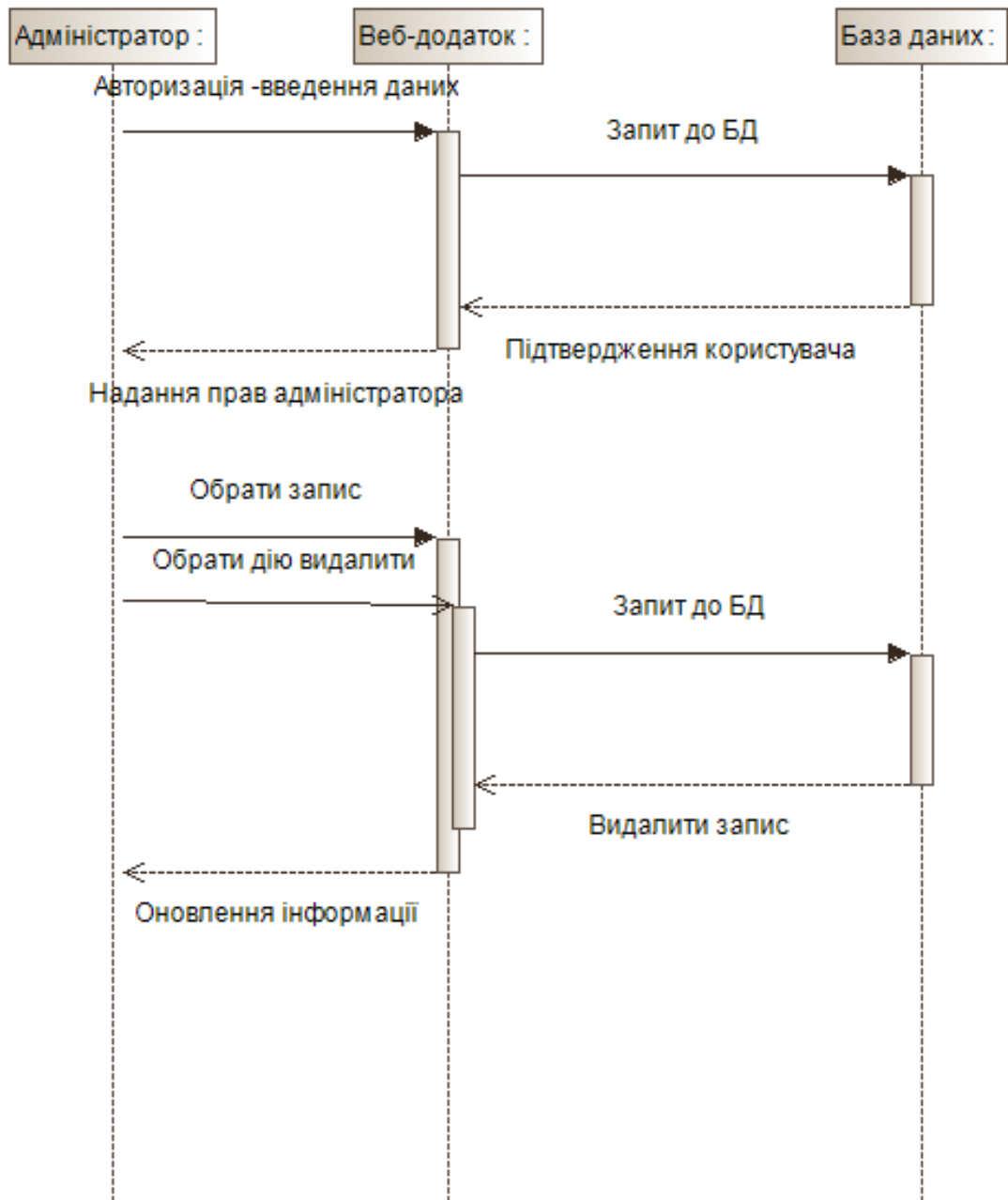
19. Stocks E.J. Sexy WEB design. *Sitpoint*, 2009. 177 р.

## ДОДАТКИ

## Діаграма послідовності читання новини



## Діаграма послідовності видалення записів адміністратором



## Діаграма послідовності реєстрації

