

## Зміст

ВСТУП.....	5
1 АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ ВЕБСЕРВІСІВ ПІДТРИМКИ ПРОДАЖІВ ПРОМИСЛОВИХ ТОВАРІВ .....	7
1.1 Опис предметної сфери веб сервісів підтримки продажів промислових товарів.....	7
1.2 Інформаційна система на базі інтернет магазину.....	9
1.3 Електронна комерція.....	11
1.4 Аналіз та порівняльна характеристика сучасних торгових інформаційних систем .....	14
1.5 Специфікація вимог до вебсервісу підтримки продажу промислових товарів.....	17
Висновки до розділу 1 .....	20
2 МОДЕЛЮВАННЯ ПРОЦЕСУ РОБОТИ ВЕБСЕРВІСУ .....	21
2.1 Етапи розробки вебзастосунків .....	21
2.2 Вибір CMS .....	25
2.3 Вибір системи керування бази даних.....	27
2.4 Сценарії використання системи .....	28
2.5 Діаграма прецедентів .....	32
2.6 Контекстна модель .....	34
Висновки до розділу 2 .....	35
3 ПРОЄКТУВАННЯ СИСТЕМИ ВЕБСЕРВІСУ ПІДТРИМКИ ПРОДАЖУ ПРОМИСЛОВИХ ТОВАРІВ .....	36
3.1 Проектування інтерфейсу.....	36
3.2 Діаграма класів.....	41
3.3 Діаграма послідовності .....	42
3.4 Діаграми розгортання бази даних .....	45
Висновки до розділу 3 .....	46
4 РОЗРОБКА ЗАСТОСУНКУ. ПРОГРАМНА РЕАЛІЗАЦІЯ.....	47
4.1 Архітектура застосунку .....	47
4.2 Програмна реалізація .....	48
4.3 Структура проекту .....	51

4.3.2 Додатковий функціонал .....	53
4.4 Тестування .....	56
Висновок до розділу 4.....	59
ВИСНОВКИ .....	61
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ .....	63
ДОДАТОК А .....	65
ДОДАТОК Б .....	73

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ**

**ІС** – Інформаційна система

**НТД** – Неструктурований тип даних

**СТД** – Структурований тип даних

**ЕК** – Електронна комерція

**ТІС** – Торгова інтернет система(Trade Internet System)

**CMS** (Content Management System) – система керування вмістом

**БД** – База даних

**СКБД** – Системи керування баз даних

**UML** (Unified Modeling Language) – уніфікована мова моделювання

## ВСТУП

**Актуальність теми.** В наш час важливим фактором престижності будь-якого підприємства є – власний сайт. Наявність сайту не лише спрощує роботу підприємству, а й приваблює багато клієнтів. В реаліях коронавірусної епідемії, онлайн купівлі стали лідером в сфері торгівлі, на це вказує статистика та відгуки самих клієнтів. Покладаючись на дані факти, можна затвердити, що тема бакалаврської кваліфікаційної роботи є актуальною в повній мірі.

З плином часу інтернет технології розвивались, ставали більш складними та доскональними, це в свою чергу давало більше можливостей для розвитку ринку торгівлі. Основний сплеск розвитку онлайн торгівлі відбувся в десятих роках XXI століття. В певній мірі – це перевернуло сприйняття торгівлі с ніг на голову. Звісно, адже раніше для того, щоб придбати якусь річ потрібно було самому ходити й шукати її будь-де, а в умовах відкриття онлайн магазинів, достатньо було відкрити браузер та замовити потрібну річ.

Онлайн торгівля одразу стала одним із світових лідерів в сфері бізнесу, багато бізнесменів вчасно звернули увагу на цей напрям й почали вкладати кошти в цей напрям.

Створення сайту магазину, це можливість привести до бізнесу нові функції та можливості, так як:

- Залучення нової аудиторії, будь-то клієнт, чи можливо постачальник товарів.
- Формування власного бренду, та його пізнаваність серед аудиторії;
- Більший наплив клієнт означає, більше прибутку та більше можливостей для розширення;
- Збирати базу даних клієнтів для того, щоб постійно підтримувати контакти та пропонувати послуги або товари, здатні їх зацікавити.
- Надати максимальну кількість інформації онлайн, це означає, менший обсяг спілкування вже безпосередньо під час продажу;

Але слід розуміти, для того щоб досягти поставлених цілей, потрібно чітко та коректно сформулювати та виконати всі етапи роботи.

**Метою** є автоматизація електронної комерції за рахунок розробки інтернет-магазину на платформі WordPress.

Для досягнення поставленої мети у роботі необхідно вирішити наступні завдання:

- 1) Зробити аналіз сучасного стану задачі електронної комерції та розробки інтернет магазинів;
- 2) Обрати технологій розробки інтернет-магазинів, визначити структури та етапів реалізації;
- 3) Дослідити особливості трекінгу показників вашого інтернет-магазину;
- 4) Запропонувати заходи щодо удосконалення аналітики;

**Об'єктом** є процес моделювання системи для веб-застосунку підтримки продажів промислових товарів.

**Предмет** є технологія розробки інтернет-магазину на платформі WordPress.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ ВЕБСЕРВІСІВ ПІДТРИМКИ ПРОДАЖІВ ПРОМИСЛОВИХ ТОВАРІВ

## 1.1 Опис предметної сфери веб сервісів підтримки продажів промислових товарів

Технологія розробки інтернет-магазину – це клопіткий процес, що вимагає від розробника чималих знань в даній сфері. Якогось конкретного шаблону для розробки веб сервісів немає, кожен розробник, так чи інакше створює свій сайт по своєму, хтось спочатку пише основний скелет сайту, а потім уже допрацьовує потрібні механіки, хтось навпаки, спочатку обдумує все до найменших дрібниць, створює потрібні блок-схеми, супутню документацію, а тільки після цього, приступає до роботи. Це вибір кожного, проте в професійній спільноті, прийнято вважати, що розробка продукту без попереднього представлення його у вигляді тих самих блок-схем, москитер схем, сценаріїв використання та іншого, значною мірою сповільнює процес розробки, призводить до виникнення проблем під час конструювання, а в подальшому збільшення витрат на створення. Тому, в великих компаніях, які займаються розробкою веб-сайтів, все підготовлюють до початку розробки, адже за допомогою, чітко поставлених вимог, правильно сформованих завдань та правильного розподілу обов'язків, робота проходить чітко, сплановано та коректно.

### 1.1.1 Веб-сервіс (веб-сайт) для інтернет-магазину

Веб-сайт інтернет-магазину – це веб-ресурс, який знаходиться на просторах інтернету, де відбувається реалізація та продаж товару кінцевому споживачу [2].

Шляхів для розробки інтернет-магазину чимало, проте в даному випадку використовується сервіс WordPress. WordPress – це зручна й багато містка система керування вмістом з відкритим кодом, яка через простоту

використання здобула чималу популярність серед користувачів, й широко використовується для створення невеликих блогів або масштабних веб-сайтів. На базі WordPress, вбудована система тем та плагінів, що значною мірою спрощує використання, та створення нових веб-сайтів. За допомогою даних тем та плагінів, та у поєднанні з вдалою архітектурою, дозволяє конструювати на базі WordPress практично будь-які веб-проекти.

Створення веб-сервісів на базі WordPress – не вважають складною задачею. Справді, сервіс пропонує багато готових рішень, за допомогою яких, можна швидко сконструювати придатний до користування сайт. Проте, WordPress, надає можливості і для власного креативу. Наявна можливість вбудовувати власний код, створювати свої шаблони, тощо. Це сприяє швидкості розробки, проте не завдає шкоди якості. Багато популярних сайтів, були створені на базі WordPress, і в цьому не має нічого страшного.

Якщо взяти до уваги не великий магазин, якому потрібно сайт, для показу товарів, можливо простої реєстрації клієнта, для подальшого спілкування, і просто розповсюдження інформації про заклад, то створення сайту на «WordPress», це чудове рішення, адже це буде не дорого, й у разі необхідності все можна швидко змінити.

### **1.1.2 Класифікація інтернет-магазинів**

Розрізняють декілька систем класифікації інтернет-магазинів:

1. За методом роздрібного продажу товарів:

Веб-вітрини; Інтернет-магазини; Торгові системи; Торгові ряди; Контентні проекти (системи інтернет-замовлень товарів, споживацькі каталоги тощо);

2. За бізнес-моделлю:

Повністю онлайн-магазин; Суміщення онлайн-магазину з офф-лайн-магazinом (Коли на основі вже діючої торгової структури створюється інтернет-магазин);

### 3. За ступенем автоматизації:

Серед торгових систем електронних магазинів розрізняють веб-вітрини, інтернет-магазини та торгові інтернет системи (ТІС) [2].

Веб-вітрина – це каталог товарів, з наявною системою навігації та оформлення замовлення. Після того, як товар був замовлений, дані стосовно замовлення передаються через систему, менеджеру, який в подальшому опрацьовує його, передає дані на склад та логістичну компанію. Менеджер здійснює вручну дії, які потрібні для відправлення товару клієнту, він контролює процес оплати замовлення до завершення процесу купівлі. Паралельно ведеться робота стосовно маркетингової, рекламної та аналітичної частини бізнесу.

Якщо брати до уваги систему інтернет-магазину, вона є повністю автоматизованою. В даній системі наявна повна автоматизація процесу замовлення та оплати, що забезпечує можливість працювати з кожним клієнтом індивідуально.

Спільна риса інтернет-магазину та ТІС – це можливість здійснювати повний торговий цикл у режимі онлайн. При цьому ТІС, додатково інтегрована в систему документообігу підприємства. Залишається лише одна неавтоматизована складова, як інтернет-магазину так і ТІС – це доставка товару.

#### **1.2 Інформаційна система на базі інтернет магазину**

Під час торгівлі, важливим фактором є – відслідковування, збір, обробка, зберігання, передавання, використання та захист інформації. Це відноситься стосовно реалізованого товару, нових клієнтів та банківських транзакцій під час торгівлі, збирання інформації стосовно вподобань та актуальності товару серед споживача тощо [5]. Все це – Інформаційні процеси. Під час інформаційного процесу, різного роду дані перетворюються з одного типу в інший. Інформаційні процеси відбувається під час передачі інформації від



джерела до приймача, за допомогою так званих каналів передач, або потоків. Інформаційні потоки є невід'ємною та ключовою складовою будь якої інформаційної системи (ІС).

В загальному інформаційна система – це система в якій здійснюються, або в якій відбуваються інформаційні процеси. В залежності від розміру ІС, в ній можуть відбуватись один, два або декілька інформаційних процеси, а в більш масштабних системах, кількість процесів може буди наближена то десятків сотень тисяч [5].

У більш вузькому розумінні, інформаційна система – це комплекс інформаційних, програмних, технічних та організаційних засобів, які потребуються для автоматизованого опрацювання інформації.

В інформаційних системах відбуваються наступні процеси:

- Введення інформації, отриманої з різних джерел;
- Перетворення (опрацювання) інформації;
- Зберігання опрацьованої потрібної інформації;
- Виведення вже готової інформації;
- Отримання та відправка інформації мережею.

Під час розробки інформаційної системи, вирішуються наступні завдання:

- Заповнення системи даними, певної галузі виробництва, які придатні до використання або для подальшого опрацювання;
- Створення графічного(не обов'язково) інтерфейсу користувача.

Зберігання даних в ІС ділять на декілька типів.

Перший тип називається – неструктурований тип даних (НТД), а другий має назву – структурований тип даних (СТД).

Неструктурований тип даних (НТД) – це текстові документи. Наприклад: реферати, книги, статті, журнали. В системах, в яких зберігаються не структуровані дані, не завжди можна знайти чітку відповідь на запит,

зазвичай видається масив даних, в яких потрібно шукати потрібну інформацію вручну.

Структурований тип даних (СТД) передбачає задання вимог (правил), які визначають їх характеристики, тобто: тип, розмір, форму, значення.

Якщо розглядати ІС на базі інтернет-магазину, потрібно розуміти, що в даному випадку потрібно використовувати саме СТД. Це потрібно для того, щоб відвідувач сайту, чітко розумів, як користуватись сайтом компанії, як знайти потрібний йому товар тощо. Можна навести приклад: якщо буде використовуватись НТД, клієнт, який завітав до сайту компанії, й намагається вибрати потрібний йому товар, через не досконалість побудови системи, буде занадто довго шукати потрібний товар, що йому не сподобається й клієнт піде з сайту, або просто не знайде потрібний товар, серед масиву всіх даних, в результаті, клієнт піде з сайту. Можна зробити висновок стосовно використання типу зберігання даних в інформаційній системі інтернет-магазину: потрібно зробити ІС максимально зрозумілою для користувача, розробити та реалізувати графічний інтерфейс, та структурувати дані на сайті, в такому разі простота використання системи звичайним користувачем зросте та підвищить електронно комерційну діяльність на сайті.

### **1.3 Електронна комерція**

Електронна комерція (англ. e-commerce) - сфера економіки, що включає всі торгові та фінансові транзакції, що здійснюються за допомогою комп'ютерних мереж, а також всі бізнес-процеси, що стосуються проведення цих транзакцій [4].

Іншими словами електронною комерцією можна назвати будь-яку фінансову діяльність в інтернеті, це будь-яка транзакція з використанням девайсу, який підключений до мережі інтернет.

В сфері ЕК виділяють п'ять основних напрямів:

- Грошові перекази, здійснені з електронних гаманців і карт через будь-які платіжні системи;
- Інтернет банкінг;
- Інформаційні сайти;
- Маркетингова діяльність в мережі;
- Онлайн торгівля.

Доцільним й основним прикладом ЕК є онлайн-магазин. В даному проєкті було використано плагін сервісу WordPress, під назвою – WooCommerce.

WooCommerce – це плагін для електронної комерції (ЕК) з відкритим кодом на базі WordPress, та який призначений для інтернет-продавців малого та великого розміру.

WooCommerce, це масштабний плагін, за допомогою якого можна побудувати інтернет-магазин з нуля, проте, який буде виконувати всі основні функції та навіть більше. За допомогою цього плагіну можна реалізувати, клієнтський кошик, систему онлайн продажу та багато інших особливостей притаманних інтернет магазинам.

### **1.3.1 Основні типи електронної комерції**

В світовій спільноті було визначено певні типи ЕК, які є загально прийнятими та використовуються щоденно [4].

Основні типи ЕК:

- C2C (Consumer-to-Consumer);
  - B2B (Business-to-Business);
  - B2C (Business-to-Consumer);
  - E2E (Exchange-to-Exchange);
  - G2C (Government-to-Citizens).
1. **C2C** (Consumer-to-Consumer).

Схема «споживач-до-споживача». Наприклад: торговий майданчик eBay, навіть, якщо людина не підприємець, вона все одно може виставити якусь річ на продаж.

## 2. **B2B** (Business-to-Business).

Схема «бізнес-до-бізнесу». Основною характеристикою даного типу комерції є – продаж оптових партій товару від виробника (дилера). Дилер реалізує товар дрібним оптом інтернет-магазинам, а вже потім товар потрапляє до кінцевого споживача.

## 3. **B2C** (Business-to-Consumer)

Схема «бізнес-споживач». Характеризує даний тип комерції, угоди між юридичними і фізичними особами. Тобто розрахунок між клієнтом та онлайн-магазином. Наприклад оренда програмного забезпечення, або будь-яка роздрібна торгівля.

## 4. **E2E** (Exchange-to-Exchange)

Схема «біржа-до-біржі». В загальному та в інтернеті даний термін (**E2E**) використовується для позначення обміну інформацією або угод між веб-сайтами, які в свою чергу самі служать в якості брокерів або бірж для обміну товарами і послугами між підприємствами. Можна розглянути простий приклад: транзакція з банківської карти на електронний гаманець, або навпаки. E2E можна розглядати як форму B2B.

## 5. **G2C** (Government-to-Citizens)

Схема «уряд-до-громадян». Яскравим прикладом даної схеми є сплата податків, комунальних послуг, ліцензій, і так далі.

Також існують й інші схеми ЕК, проте вони вже є більш вузько направлені, й вважаються простими відгалуженням вже перерахованих типів.

### 1.3.2 Переваги електронної комерції

Основним, але далеко не останнім плюсом ЕК, вважається відсутність географічних обмежень. Багато великих торгових компаній співпрацюють разом з логістичними компаніями, що допомагає охопити аудиторію по всьому світі.

Коли компанія розширює свої межі, її бренд стає відомим та більш впізнаним серед аудиторії, відомості про бренд розлітаються дуже швидко. Цьому сприяє, розвиток соціальних мереж, де люди можуть ділитись своїми вподобаннями між друзями та не знайомим людьми, можуть демонструвати товар компанії, який придбали, цим самим його рекламувати.

Ще одним не від'ємним плюсом електронної комерції є – мінімізація витрат на утримання оффлайн точок реалізації продукції, покупку обладнання та виплату заробітних коштів персоналу.

Саме ці фактори виводять ЕК в топ лідерів серед торгівлі.

#### **1.4 Аналіз та порівняльна характеристика сучасних торгових інформаційних систем**

В час стрімкого розширення в сфері ІТ технологій, з кожним днем з'являються все нові сайти, нові інтернет-магазини тощо. Тому доцільно буде зауважити, що для того, щоб правильно сконструювати, а потім розробити інтернет-магазин, потрібно знати сучасні тенденції в професійній спільноті. Для того, щоб інтернет-магазин приносив прибуток, потрібно чітко сформулювати вимоги під-час його створення. Проте, спочатку потрібно звернути увагу, на інші вже готові рішення, інших компаній. Після збору та опрацювання інформації, буде отриманий чіткий план дій, спрямований на максимальне покращення інтернет-магазину.

В сфері онлайн-торгівлі, кожна компанія, чи можливо маленький магазин, намагається створити свій сайт, для просування його в мережі, за для більшого розповсюдження власного бренду серед мас. Під-час створення

інтернет-магазину, важливо розуміти сучасні тенденції розробки. Доцільно буде поставити наступні питання:

- Що подобається користувачу?
- Чи зручно буде користуватись сайтом?
- Чи можна буде здійснювати купівлю не лише з ПК?
- Що наявне у інших компаній, чого не має у нас?

Відповівши на ці запитання, можна вже скласти не малий список особливостей, які притаманні інтернет-магазинам та які значною мірою допоможуть зрозуміти, що потрібно робити.

Спираючись на вище наведену інформацію було прийнято рішення здійснити аналіз інформаційних систем інтернет-магазинів України.

Аналіз ТМ «Світ-Буд»

Таблиця 1.1 – Аналіз ТМ «Світ-Буд»

ТМ «Світ-Буд»	
Виробник	ТМ «Світ-Буд»
Мова реалізації	Html, css, javascript
Основні функції	<ul style="list-style-type: none"> <li>- Продаж онлайн</li> <li>- Чат з оператором в будь-якому месенджері або в браузері</li> <li>- Наочність представлення товару</li> </ul>
Переваги	<ul style="list-style-type: none"> <li>- Швидко завантажується</li> <li>- Відсутність рекламних банерів</li> </ul>
Недоліки	<ul style="list-style-type: none"> <li>- Простий дизайн</li> <li>- Проблема з кнопками(часто перенаправляють не туди куди потрібно)</li> </ul>
Джерело інформації	<a href="https://svit-bud.com/">https://svit-bud.com/</a>

ТМ «Світ-Буд» - це компанія на території України, яка займається торгівлею, у сфері оптових продажів. Якщо розглядати тип електронно комерційної діяльності даної компанії, то доцільно буде зауважити, що – це B2B (Business-to-Business). Дана компанія, не має офф-лайнових точок

реалізації, а має лише власний сайт, за допомогою якого й веде власну комерційну діяльність.

Сайт компанії виглядає сучасно проте дуже просто, має можливість перегляду каталогу товарів, замовлення, консультування онлайн. Проте, були виявлені й проблемні місця, по-перше це – адаптивність сайту, при переході на мобільний девайс, сайт веде себе не коректно, форми залізають один на одного, що здійснює використання сайту не зручним.

Аналіз ТМ «Eldorado»

Таблиця 1.2 – Аналіз ТМ «Eldorado»

ТМ «Eldorado»	
Виробник	ТМ «Eldorado»
Мова реалізації	Html, css, javascript
Основні функції	<ul style="list-style-type: none"> <li>- Зручна розмітка сторінки</li> <li>- Зрозумілий інтерфейс</li> <li>- Адаптивність під різні пристрої</li> <li>- Реєстрація та авторизація</li> </ul>
Переваги	<ul style="list-style-type: none"> <li>- Стильний інтерфейс</li> <li>- Широкий спектр сортування товарів</li> <li>- Чат-бот, який відповідає на питання</li> </ul>
Недоліки	<ul style="list-style-type: none"> <li>- Не завжди коректне представлення товарів при сортуванні</li> <li>- Великі рекламні банери</li> <li>- Повільне завантаження сторінок</li> </ul>
Джерело інформації	<a href="https://eldorado.ua/">https://eldorado.ua/</a>

ТМ «Eldorado» - це велика компанія, яка займається електронною комерційною діяльністю на території України. Є одним із найбільших представників торгівельних компаній на ринку. Має, як оффлайн точки реалізації, так і потужний сайт. На сайті наявна можливість реєстрації особистого кабінету, перегляду каталогу товарів, сортування товарів за групами тощо. Проте навіть у такого гіганта торгівлі, також є деякі нюанси й проблеми з сайтом.

## **1.5 Специфікація вимог до вебсервісу підтримки продажу**

### **промислових товарів**

1. Призначення застосунку, для кого розробляється ПЗ  
Призначенням застосунку, що проектується, є автоматизація електронної комерції.
2. Погодження, що були ухвалені в програмній документації  
Було узгоджено, що для створення застосунку та подальшої його реалізації буде використовуватись вебсервіс «WordPress».
3. Межі проєкту ПЗ  
Крайня дата завершення робіт над проєктом – 12.06.2022р.

### **Загальний опис**

1. Сфера застосування  
Дане ПЗ буде застосовано для комерційної діяльності в місті Миколаїв, на приватному підприємстві.
2. Характеристики користувачів  
Основні характеристики користувачів: наявність будь-якого девайсу з можливістю виходу до мережі Інтернет (телефон, комп'ютер, ноутбук).
3. Загальна структура системи  
Для повноцінної роботи системи потрібно використовувати: вебзастосунок, БД, сервер.
4. Обмеження  
До обмежень зв'язаних з роботою ПЗ належить лише – доступ до мережі Інтернет.

### **Функції вебсервісу підтримки продажів промислових товарів**

#### Функція перегляду товарів

##### Опис функції

Функція дає можливість для користувача, відкрити повний каталог товарів на сторінці магазину, щоб детально ознайомитись з асортиментом, характеристиками, цінами товарів, що пропонуються.

##### Вхідна і вихідна інформація



Вхідна інформація – назва товару, ціна, тип товару, артикул, виробник.

Вихідна інформація – товар або низка товарів, які відповідають вимогам заданими користувачем [1].

Функціональні вимоги

Бази даних з товарами та доступ до мережі Інтернет.

Функція онлайн оплати товару

Опис функції

Функція дає можливість для користувача, в форматі онлайн, здійснити оплату вибраного раніше товару.

Вхідна і вихідна інформація

Вхідна інформація – назва товару (товарів).

Вихідна інформація – чек та накладна, яка засвідчує завершення етапу оплати.

Функціональні вимоги

База даних з інформацією про товар, доступ до мережі Інтернет, наявність інтернет банкінгу у користувача.

### **Вимоги до інформаційного забезпечення**

#### 1. Джерела і зміст вхідної інформації (даних)

У ПЗ, що проєктується, основним джерелом вхідної інформації є як система так і користувач. Користувач надає потрібно інформацію про товар, який бажає переглянути чи придбати. Система надає інформацію стосовно товарів, які можуть підійти користувачеві за його вподобаннями.

#### 2. Нормативно-довідкова інформація (класифікатори, довідники тощо)

Вимоги до даного пункту відсутні.

### **Вимоги до технічного забезпечення**

Особливих вимог до технічного забезпечення не має. Девайс, будь-то ПК чи ноутбук, повині мати операційну систему Windows не нище 7-ї версії, а також можливість працювати з сучасними браузерями.

## **Вимоги до програмного забезпечення**

### 1. Архітектура програмної системи

Архітектура ПЗ: клієнтської частини, серверної частини та БД.

### 2. Системне програмне забезпечення

Застосунок має бути побудованим з використанням фреймворку Wordpress. В якості БД для застосунку обрано MySQL.

### 3. Мережне програмне забезпечення

Для створення ПЗ використано ОС Windows 10, у якості IDE обрано PhpStorm, та Google Chrome для перегляду вебсторінок.

### 4. Програмне забезпечення ведення інформаційної бази

Через MySQL має відбуватися керування, ведення інформаційною базою застосунку. Для цього використовується програмний застосунок phpMyAdmin.

### 5. Мова і технологія розробки ПЗ

Мови розробки – PHP та JavaScript.

## **Вимоги до зовнішніх інтерфейсів**

### 1. Інтерфейс користувача

Інтерфейс користувача повинен буди побудований за усі вимогами UX та UI.

### 2. Апаратний інтерфейс

Апаратним інтерфейсом буде виступати обраний девайс користувача, тобто: ПК, ноутбук, смартфон, планшет.

### 3. Програмний інтерфейс

Програмним інтерфейсом виступає платформа «WordPress».

## **Властивості програмного забезпечення**

### 1. Доступність

Застосунок повинене бути доступним для будь-якого користувача.

### 2. Техніка

ПЗ може використовуватись на різних девайсах. ПК – ОС Windows, не нище 7-ї версії, смартфон на базі ОС IOS та Android, версій не нище 12.0.0 та 7.1 відповідно.

### 3. Продуктивність

Продуктивність роботи застосунку безпосередньо пов'язане з швидкістю Інтернету. При стандартному режимі роботи, час виконання запитів не повинен перевищувати 2-4 секунди.

### 4. Надійність

Дані користувача (у разі замовлення або реєстрації) є абсолютно приватними, ні ФІО ні будь-яка інша інформація не буде передана третім лицям.

## Висновки до розділу 1

Було проаналізовано предметну сферу у якій будується вебсервіс. Знання щодо сфери розробки, надають повне розуміння того, що потрібно робити. Чітке представлення системи, дає можливість розробки власної інформаційної системи.

Було досліджено різноманітні типи та класифікації інформаційних системи інтернет-магазинів. Ці знання дають змогу та можливість орієнтуватись у існуючих структурах.

Були розглянуті аналоги інформаційних систем інтернет-магазинів. Було виявлено низку існуючих проблем. Було визначено те, що не існує якоїсь ідеальної системи, де не буде прогалин або проблем і яка би змогла виконувати усі необхідні функції.

Було досліджено специфікацію вимог щодо розроблюваної інформаційної системи. Було описані функціональні вимоги застосунку.

Згідно вище написаного, можна зазначити, що дослідження галузі ІС цілком закінчено, що в свою чергу означає, можна розпочинати моделювання та проектування застосунку.

## **2 МОДЕЛЮВАННЯ ПРОЦЕСУ РОБОТИ ВЕБСЕРВІСУ**

### **2.1 Етапи розробки вебзастосунків**

Розробка вебзастосунків відбувається у певному порядку. Для правильного та швидкого виконання завдання потрібно притримуватись певних правил [6]. Вебсайти створюються у декілька етапів:

1. Планування — визначення суті сайту, його призначення та застосування, розробка ТЗ.
2. Розробка — розробка структури сайту, підбір потрібних технологій та рішень, які потрібні для виконання поставленої задачі.
3. Створення спеціальних окремих сторінок відповідно до структури, включення до них гіперпосилань.
4. Тестування — перевірка сайту на вміст помилок, багів.
5. Розміщення — розміщення сайту в Інтернеті.
6. Підтримка — оновлення вмісту сайту.

#### **Етап планування**

На етапі планування слід визначити найголовніші аспекти сайту, тобто, яким сайт повинен бути, яка мета створення сайту, за, що він повинен відповідати. Потрібно визначити також, чи буде сайт тематичним (на якусь конкретну тему), чи можливо різні його сторінки будуть присвячені різним темам. Планування дуже важливий етап, адже від правильно сформованого завдання, залежить успіх усього проекту. Потрібно чітко сформулювати завдання, розподілити обов'язки тощо.

#### **Етап розробки**

Етап розробки, також дуже важливий етап, без якого не можливо, навіть уявити, роботу над проектом. Цей етап характеризується розробкою структури сайту. Під час розробки структури сайту, потрібно визначитись з кількістю сторінок та встановити зв'язки між ними. Розрізняють декілька видів структур сайту: лінійну, ієрархічну та довільну структури[5].

Лінійна структура сайту – це найпростіша структура сайту. Сторінки в даній структурі йдуть одна за одною, без чітко розмежованих переходів, користувач переглядає їх, як слайд-шоу. У структурі даного типу не існує поділу контенту на рівні, адже всі сторінки рівноправні, і їх повинен побачити кожен користувач. Лінійна структура вважається найпростішою, проте це не означає, що в ній не має мінусів, навпаки, недоліків в даній структурі явно більше ніж плюсів. Тому логічно припустити, що область використання даної структури доволі обмежена. Зазвичай вона використовується на іміджевих сайтах, сайтах-презентаціях і в онлайн-навчальних посібниках[6].

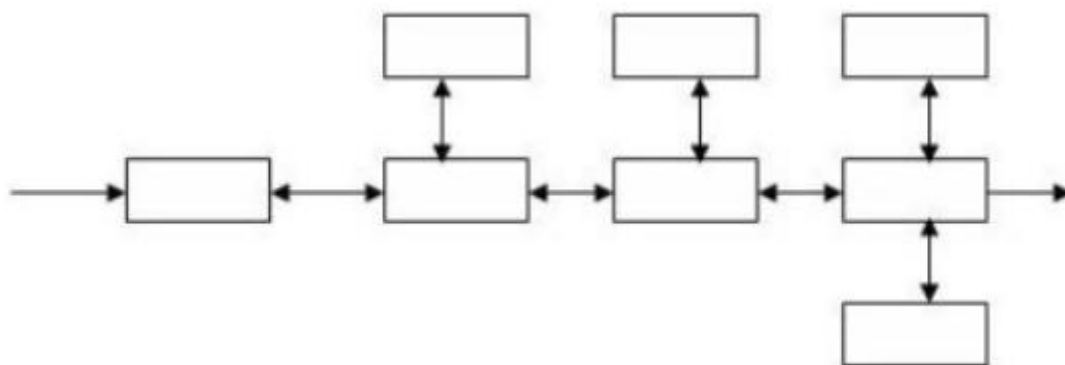


Рисунок 2.1 – Лінійна структура сайту

Ієрархічна структура сайту – при даній структурі, спочатку створюється головна сторінка, яка не має попередніх, решта сторінок мають лише одну попередню. При ієрархічній структурі кожна сторінка може містити посилання на довільну кількість сторінок сайту. Така структура найкраще підходить для сайтів, що містять різну за тематикою інформацію: каталогів, збірань статей з різних тем або добірок послань [5].

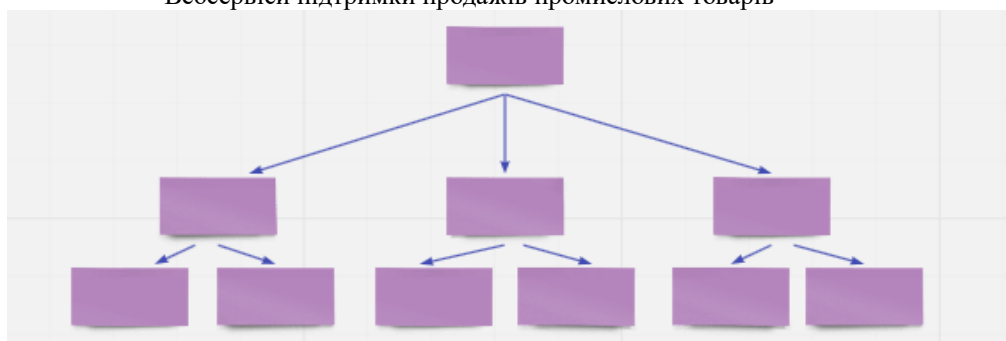


Рисунок 2.2 – Ієрархічна структура сайту

Довільна структура сайту – найбільш широко використовувана структура. Містить в собі деякі характерні особливості від двох попередніх варіантів. Така структура характерна тим, що сторінки пов'язані між собою довільним чином. На перший погляд можна подумати, що довільна структура зовсім неорганізована, й це правда, проте саме в цьому й полягає принцип використання даної структури. Перехід між сторінками відбувається в різні способи.

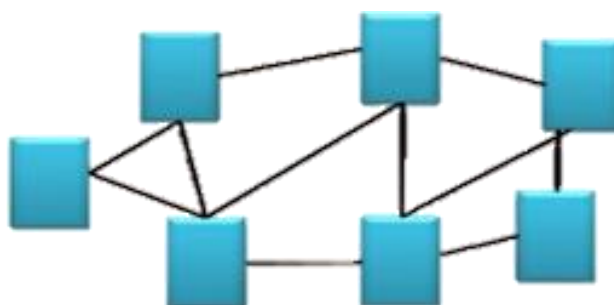


Рисунок 2.3 – Довільна структура сайту

Також на етапі розробки створюється супутня документація, наприклад блок-схеми, тосир-схеми, різного роду діаграми. Це допомагає в подальшому більш чітко розуміти сам процес роботи, й, що вимагається від розробника. Багато хто нехтує розробкою даної документації, проте це не правильно. В великих компаніях, які працюють над складними проектами, створення такої документації дуже важливо, це дає змогу, більш чітко зрозуміти, що саме бажає отримати клієнт, після завершення робіт, допомагає чітко розподілити обов'язки, що в свою чергу сприяє більш якісному виконанню завдання [6].

### **Етап створення**

Етап створення, або початок роботи над проектом. Цей етап вважається другим по складності, адже це по своїй суті і є сама робота, яку потрібно виконати. На відміну від попереднього етапу розробки, даний етап це практична реалізація поставлених питань та задач. На даному етапі, розробники створюють продукт, від перших його сторінок, й до фінального завершення основної частини роботи. На даному етапі від команди розробників потребується повністю задовольнити всі вимоги, які були представлені під час розробки проекту. Під час етапу створення відбувається наповнення сторінок конкретними матеріалами (контент), а також створення та встановлення гіперпосилань, для того, щоб можна було зручно подорожувати сайтом. При цьому потрібно пам'ятати, що під час реалізації дизайну, та розробки сайту в цілому, потрібно притримуватись чітко тих вимог, які були представлені в ТЗ.

### **Етап тестування**

Після завершення етапу створення, наступним йде тестування. Дуже важливий етап, який допомагає виявити несправності, баги та недоліки системи, що розроблялась. На цьому етапі потрібно провести ряд дій, спрямованих на перевірку коректності ПЗ, потрібно:

1. перевірити коректність роботи гіперпосилань;
2. перевірити зручність використання навігації на сайті;
3. перевірити, чи коректно працюють всі графічні елементи на сторінці;
4. перевірити сайт на адаптивність та правильну роботу в різних браузерах.

У разі виникнення будь-яких помилок чи багів, потрібно внести виправлення та ремарки.

### **Етап розміщення**

На даному етапі потрібно визначитись де буде розміщений даний сайт.

Існує декілька варіантів розміщення:

1. Власні сервера (наприклад сервер університету);
2. Сервер вашого провайдера;
3. На сервері спеціальної організації, яка спеціалізується на наданню даних послуг;
4. На сервері, який надає послуги вільного й безкоштовного розміщення сайтів.

Переваги кожного варіанту особливі, кожен має право сам вибирати, де розміщувати власний сайт.

### **Етап підтримки**

Даний етап характерний тим, що він майже ніколи не закінчується. Інколи клієнт і компанія розробник, домовляються про підтримку ПЗ на довгий час, тобто розмова йде про оновлення, додавання нового матеріалу, виправлення тощо [7].

## **2.2 Вибір CMS**

CMS (Content Management System), або система керування вмістом – програмне забезпечення для організації вебсайтів чи інших інформаційних ресурсів [7].

По суті CMS це веб-застосунок в якому розробники створюють та в подальшому керують застосунком (сайт). Особливістю більшості CMS є наявність особистого кабінету в якому і відбувається вся основна робота по створенню сайту, також такі особисті кабінети ще називають – консоллю. Існують багато рішень, які можна використовувати, проте явним лідером серед запропонованих варіантів являється WordPress [17]. На долю WordPress приходить приблизно 30-40% усіх існуючих сайтів. Це зумовлено простотою в розробці та великою кількістю додаткових інструментів, що сприяють швидкому, а головне якісному виконанню роботи [9].

Переваги використання CMS:



1. простота створення сайтів;
2. не потрібно мати особливих навичок в дизайні чи програмуванні;
3. потрібно значно менше часу та коштів;
4. зручна система керування сайтом.

Недоліки використання CMS:

1. Постійне оновлення доповнень, а в наслідку проблеми з сумісністю;
2. Не весь функціонал можливо реалізувати;
3. Зниження продуктивності за рахунок збільшення кількості доповнень.

CMS також мають декілька видів, їх можна умовно розділити на 4 типи: CMS з відкритим кодом, коробочні CMS, самописні та framework.

CMS з відкритим кодом означає, що даний двигун має можливість модифікувати будь-хто. Завдяки цьому в подібних CMS регулярно з'являються безліч різних доповнень, тем та плагінів, що сприяють збільшенню різноманітності контенту. В таких системах швидше знаходять уразливі місця та ліквідують їх.

Коробочні CMS в більшій частині схожі з попереднім типом, проте основною відмінністю являється закритий код, тобто це означає, що ввести зміни до двигуна можуть лише офіційні розробники.

Самописні CMS, подібні двигуни розробляються на замовлення, їх функціонал значно менший ніж у вище представлених, проте чітко відповідає замовленим характеристикам, та не навантажений зайвими інструментами, які ніколи не будуть використані. Основним мінусом даної системи являється підтримка та оновлення. Якщо у разі необхідності потрібно буде, щось змінити, або виправити уразливі місця, то потрібно буде звернутись безпосередньо до офіційного розробника, або знайти стороннього спеціаліста.

Framework – надбудова над мовою програмування, тобто це набір бібліотек, за допомогою яких можливо створити будь-який сайт для будь-яких

завдань. Особливістю даного типу, являється чітко поставлений функціонал, який відповідає усім можливим вимогам проекту. Проте потрібно розуміти, що створення сайту за допомогою framework-ів займає більше часу, та вимагає більших капіталовкладень [8].

В даному проекті було прийнято рішення також використовувати WordPress.

WordPress – система керування вмістом сайту з відкритим кодом. Мова розробки – PHP. Сервер БД – MySQL.

### **2.3 Вибір системи керування бази даних**

БД (база даних) – сукупність, організованих, наборів даних, які зберігаються в пам'яті комп'ютера.

Системи керування баз даних (СКБД) – програмне забезпечення, яке відповідає за створення, збереження, оновлення та пошук інформації в базі даних, та надає контроль доступу до даних [11].

Потрібно розуміти, навіщо використовують СКБД. В основному використовувати СКБД доводиться, коли обсяг інформації для обробки занадто великий, тому використання СКБД значною мірою спрощує та автоматизує обробку потрібно інформації.

Існує багато популярних варіантів, для використання, наприклад: MySQL, Microsoft SQL Server, PostgreSQL, Sybase, Oracle, Firebird, Interbase та IBM DB2.

Для даного проекту було обрано одну з найпопулярніших СКБД – MySQL.

MySQL – реляційна система управління базами даних із відкритим вихідним кодом. Вона використовується у більшості відомих CMS, а в деяких є лише єдиним варіантом використання. Дана система має чималу кількість позитивних якостей, за які й була обрана для використання.

Переваги та характеристики MySQL:

1. база даних з відкритим вихідним кодом;

2. підтримує багато поширених операційних систем;
3. має великий обсяг супутньої документації;
4. підтримує багато мов програмування;
5. декілька механізмів для зберігання даних;
6. безкоштовна.

Саме через наявність даного спектру переваг, використання MySQL є доцільним. Також у порівнянні з іншими системи використання MySQL є набагато простіше, має широкий функціонал та користується великою популярністю серед багатьох користувачів, саме тому в мережі найбільше інформації саме про дану систему.

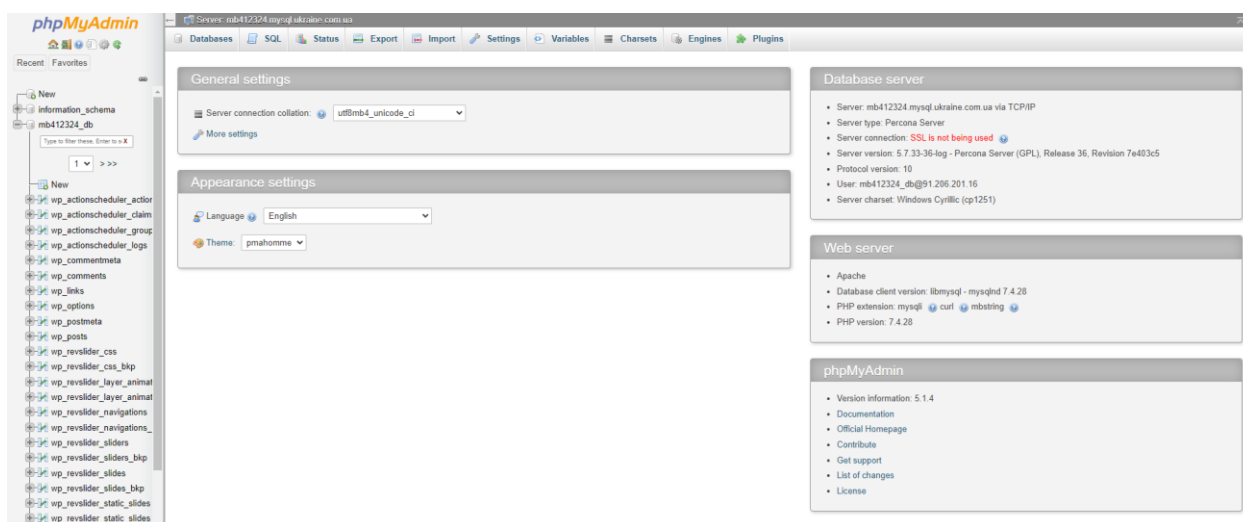


Рисунок 2.4 – Головний інтерфейс phpMyAdmin

PhpMyAdmin – зручний вебдодаток з відкритим кодом, який має графічний інтерфейс для адміністрування бази даних. Дана система зручна у використанні, адже не потребує встановлення, якогось додатково ПЗ, працювати з PhpMyAdmin можна прямо в браузері, система дозволяє керувати базою даних MySQL без вводу команд SQL, за допомогою інтерфейсу, який вказаний на рисунку 2.1 [10].

## 2.4 Сценарії використання системи

Сценарії використання – це опис поведінки системи, її реакція та відповіді на зовнішні запити. За допомогою сценаріїв використання може бути

описано «хто?» і(або) «що?» може зробити з представленою системою. За допомогою побудови сценаріїв використання, розробник більш чітко розуміє вимоги до системи [21].

#### Користувачі системи

Під час розробки системи важливо розуміти, які користувачі повинні бути наявні в системі. Це дасть більш точне уявлення щодо системи, яка проектується, та в подальшому дозволить сформулювати більш точні вимоги до системи.

#### Користувачі системи:

- Гість;
- Зареєстрований користувач;
- Продавець;
- Адміністратор.

Було розроблено декілька сценаріїв використання системи.

Таблиця 2.1 – Сценарій №1: Реєстрація

Діючі особи	Користувач, система.
Мета	Створення особистого кабінету.
Передумова	Користувач не авторизований в системі.

## Продовження таблиці 2.1

Успішний сценарій:	
<ol style="list-style-type: none"> <li>1) Користувач переходить до форми реєстрації;</li> <li>2) Проводить заповнення контактних даних;</li> <li>3) Система проводить перевірку введених даних;</li> <li>4) Система зберігає обліковий запис в базі даних;</li> <li>5) Користувач отримує повідомлення про успішну реєстрацію та перенаправляється на головну сторінку.</li> </ol>	
Результат	Новий обліковий запис збережено в базі даних.

Таблиця 2.2 – Сценарій №2: Авторизація

Діючі особи	Користувач, система
Мета	Авторизуватися.
Передумова	Покупець не авторизований.
Успішний сценарій:	
<ol style="list-style-type: none"> <li>1. Користувач вводить свій email та пароль.</li> <li>2. Система здійснює перевірку введених даних.</li> <li>3. Користувач авторизувався в системі.</li> </ol>	
Результат	Користувач в системі.
Розширення	
*а	Система не може підключитися до бд. Система виводить відповідне повідомлення.
Результат	Користувач не авторизований.

## Продовження таблиці 2.2

1a	Користувач зробив помилку при введенні email або паролю.
Результат	Користувач не авторизований.

Таблиця 2.3 – Сценарій №3: Сценарій пошуку товарів в каталозі

Діючі особи	Гість або Клієнт, система
Мета	Пошук товарів в каталозі.
Успішний сценарій:	
<ol style="list-style-type: none"> <li>1. Користувач натиснув на кнопку «пошук».</li> <li>2. Система вивела поле для введення пошукового запиту.</li> <li>3. Користувач написав в пошукове поле запит та натиснув «Enter».</li> <li>4. Система виводить результати запиту.</li> </ol>	
Результат	Користувач отримав товари, які відповідають запиту.

Таблиця 2.4 – Сценарій №4: Додавання товару до списку «Сподобалось»

Діючі особи	Авторизований користувач, система.
Мета	Додати товар до списку «Сподобалось».
Передумова	Користувач авторизований в системі.
Успішний сценарій:	
<ol style="list-style-type: none"> <li>1) Користувач обирає потрібний товар.</li> <li>2) Користувач додає товар в список «Сподобалось».</li> <li>3) Система зберігає дані про список вподобаних товарів.</li> </ol>	
Результат	Користувач додав товар до списку.

Таблиця 2.5 – Сценарій №5: Оформлення замовлення

Діючі особи	Авторизований користувач або Гість, система.
Мета	Оформлення замовлення.
Передумова	Обрано відповідний товар.
Успішний сценарій:	
<ol style="list-style-type: none"> <li>1) Користувач переходить до кошику.</li> <li>2) Користувач заповнює відповідні поля, даними.</li> <li>3) Натискає кнопку «Оплатити».</li> <li>4) Переходить на сторінку перевірки оплати.</li> <li>5) Автоматично повертається до кошику.</li> </ol>	
Результат	Користувач оформив замовлення
Розширення	
*а	Система не знаходить інформацію про оплату.
Результат	Замовлення анульовано.
la	Користувач не має достатньої суми на балансі.
Результат	Замовлення анульовано.

## 2.5 Діаграма прецедентів

Діаграма прецедентів (діаграма варіантів використання) – діаграма, на якій зображується залежність та відношення між прецедентами та акторами, в конкретній системі. Дана діаграма являється – графом, що складається з певної кількості акторів та прецедентів, які в свою чергу поєднуються зв'язками, показують відношення одного елементу системи до іншого. За правилами побудови даних діаграм завжди існують межі системи, представлені прямокутником, це означає, якщо дія виконується в межах даного

прямокутника, то вона виконується в системі, якщо дія знаходиться за прямокутником, то вона не є частиною даної системи [15].

Суть даного типу діаграм це подання системи, у вигляді множини, сутностей та акторів, що взаємодіють з системою, за допомогою так званих варіантів використання.

Існує декілька типів відношень між акторами та системою:

1. асоціації (англ. association relationship);
2. включення (англ. include relationship);
3. розширення (англ. extend relationship);
4. узагальнення (англ. generalization relationship).

Асоціації – відношення, одне з основних понять у мові UML. Вони використовуються під час побудови діаграми, майже в кожному випадку.

Включення – це різновид відношення залежності між базовим варіантом використання і його окремим випадком.

Розширення – визначає взаємозв'язок базового варіанту використання з іншим варіантом використання, функціональна поведінка якого залучається базовим не завжди, а тільки за виконання додаткових умов.



Кафедра інженерії програмного забезпечення  
Вебсервіси підтримки продажів промислових товарів

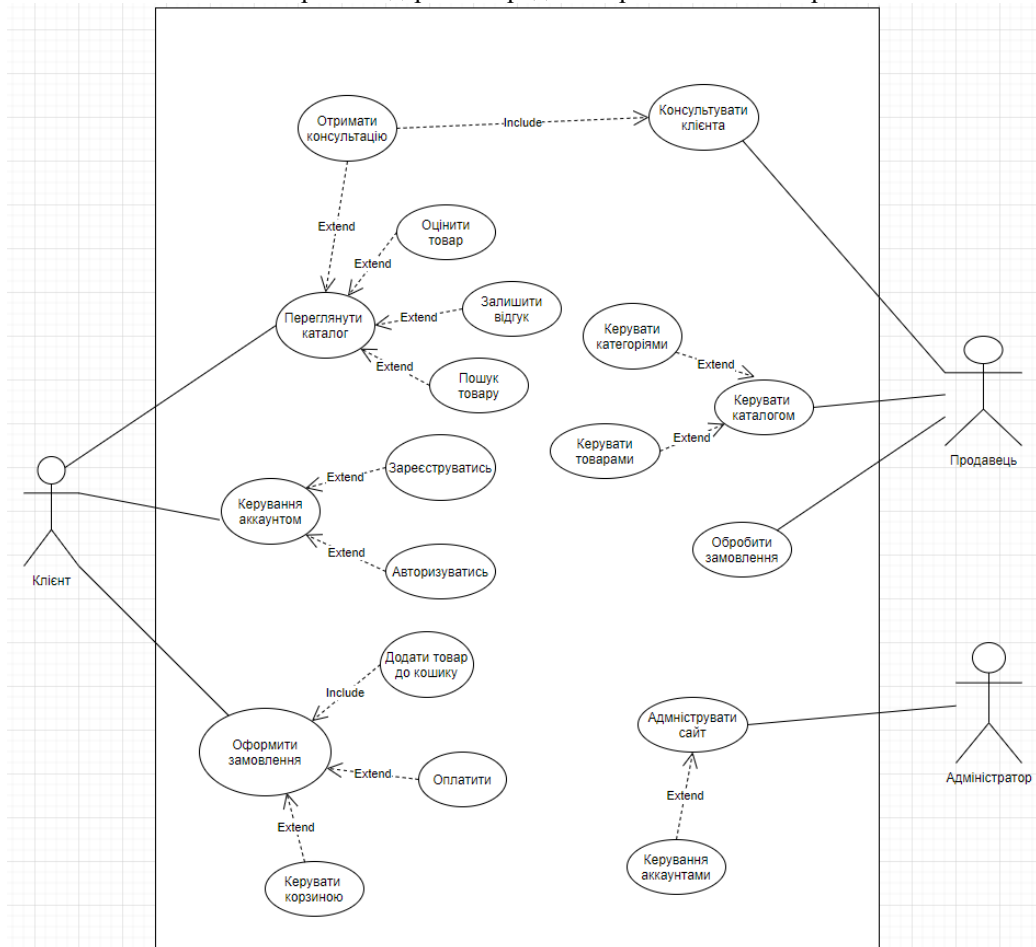


Рисунок 2.5 – Діаграма прецедентів

Дана діаграма показує роботу системи, що розробляється.

## 2.6 Контекстна модель

Контекстна діаграма – діаграма верхнього рівня, показує призначення системи (її функції) та її взаємодії з зовнішніми чинниками.

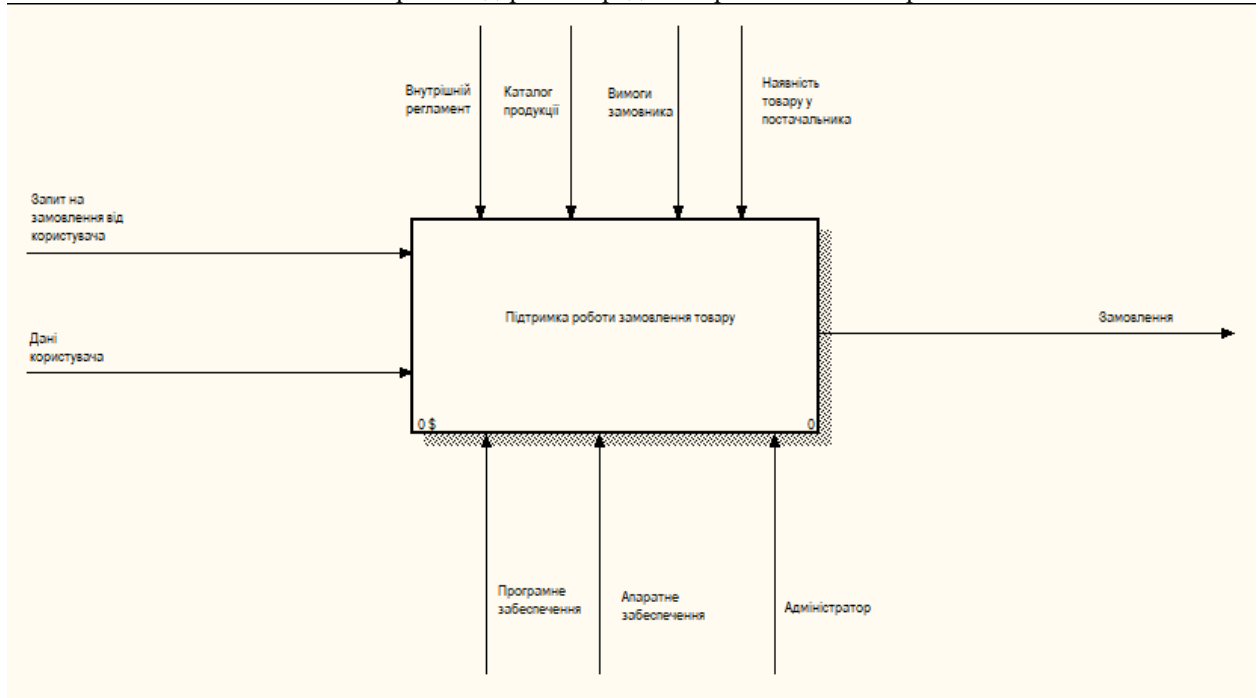


Рисунок 2.6 – Контекстна діаграма

По правилам, в кожній моделі може бути лише одна контекстна діаграма. Після опису основної функції, виконується функціональна декомпозиція, тобто визначаються функції, з яких складається основна функція.

## Висновки до розділу 2

В даному розділі було описано принцип проектування системи, що розробляється. Було описано етапи розробки та наведено графічну та текстову документацію, яка буде використовуватись у подальшій розробці проєкту. Було розроблено та наведено приклади сценаріїв використання окремих елементів системи, дані представлені у вигляді таблиць. Представлено обрану CMS систему та наведено короткий опис. На рисунку 2.5 зображено діаграму прецедентів, яка буде використовуватись у подальшій розробці застосунку. Також було побудовано контекстну діаграму.

## 3 ПРОЄКТУВАННЯ СИСТЕМИ ВЕБСЕРВІСУ ПІДТРИМКИ ПРОДАЖУ ПРОМИСЛОВИХ ТОВАРІВ

Проєктування, або іншими словами етап розробки. Як було описано вище це ключовий етап при створенні будь-якого програмного забезпечення. На даному етапі потрібно розробити багато супутньої документації, різного роду діаграми та блок-схем.

### 3.1 Проєктування інтерфейсу

Інтерфейс застосунку – наочне зображення системи, з якою працюють та яку використовують користувачі. Зображує візуальну реалізацію системи.

Для проєктування та відображення майбутнього інтерфейсу використовують різні способи.

Каркасний спосіб – спосіб представлення інтерфейсу у вигляді каркасу, без заповнення сторінки контентом. Допомогає чітко розуміти, які елементи сторінки, кнопки та банери повинні бути розташовані у тих чи інших місцях. Також даний спосіб ще називають від англ. Wireframe.

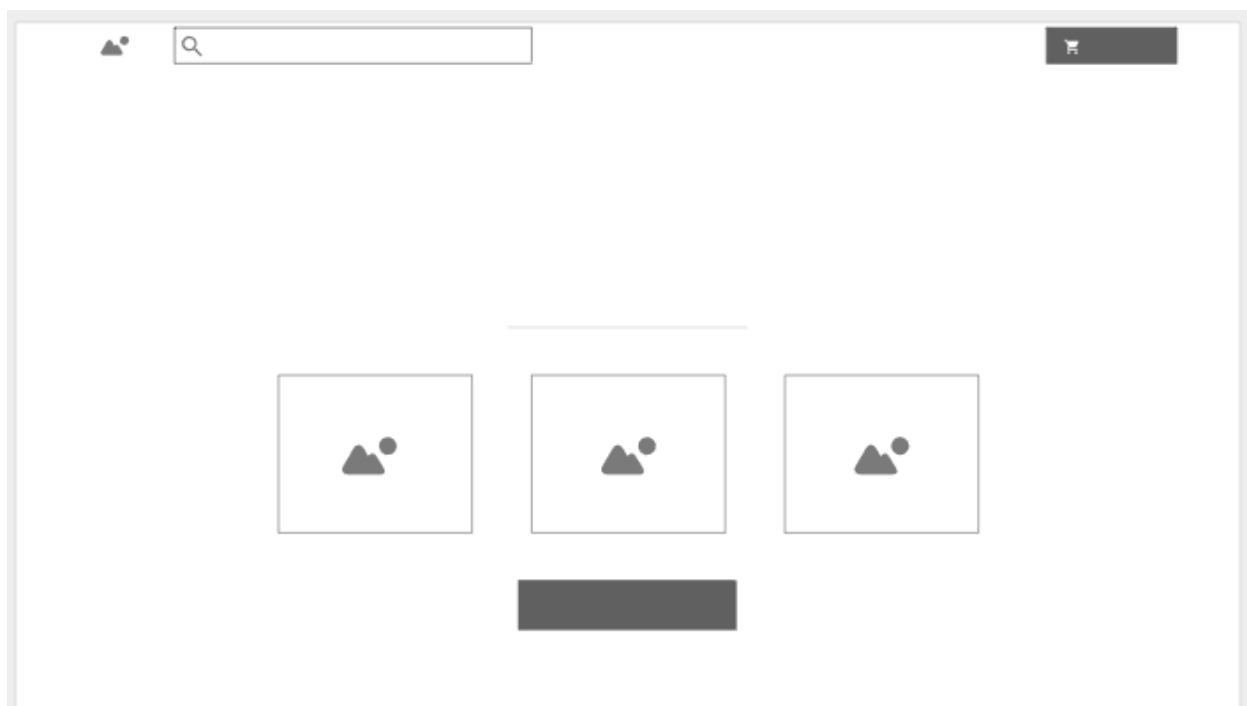


Рисунок 3.1 – Каркасний спосіб представлення

Прототип – певне зображення кінцевого результату дизайну, він не є ідентичним фінальній версії, проте є наближеним до неї. Прототип показує відношення елементів сторінки, їх місце розташування. Під час роботи, кінцевий результат може відрізнятись від прототипу, проте не значною мірою. Потрібно також розуміти різницю між прототипом і каркасом, який був описаний вище. Прототип це ніяк не сіра, абстрактна модель, це вже більш детальний вигляд майбутнього проекту.

Mock-up – найбільш деталізований спосіб подання дизайну. Може бути представлений, як уже фінальний дизайн, без подальших змін, або представлений у вигляді ще не завершеного дизайну, з можливістю вносити зміни. Mock-up найкраще представляє майбутній дизайн, він є набагато зрозумілішим, як для замовника, так і для розробників, які чітко розуміють де повинні знаходитись ті чи інші елементи сторінки.

В даному проєкті були прийнято рішення використовувати mock-up, з фінальним виглядом, тобто mock-up не буде відрізнятись від кінцевого вигляду сторінки.

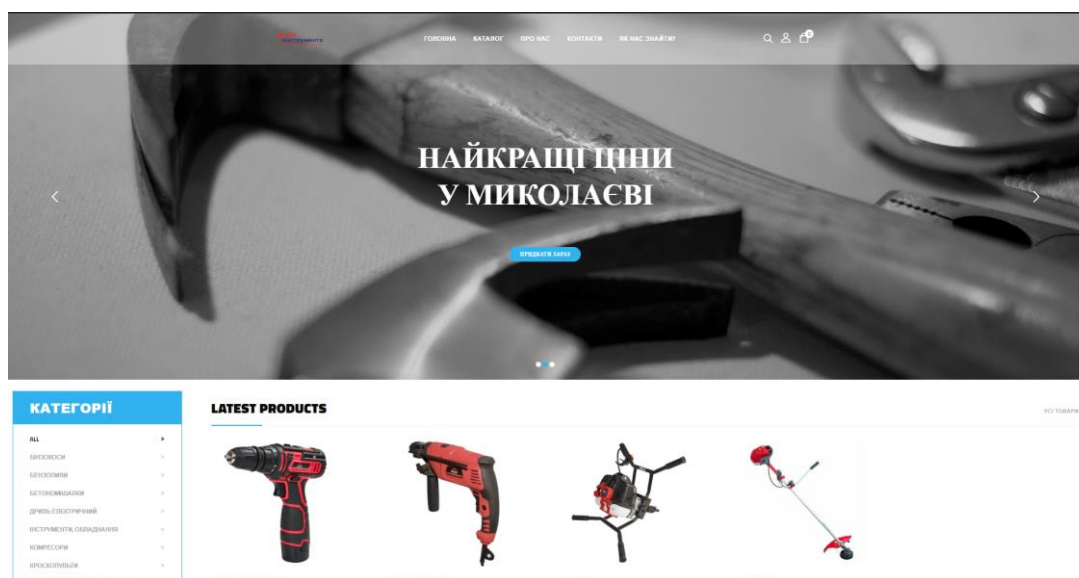


Рисунок 3.2 – Вигляд головної сторінки застосунку

На рисунку 3.2 зображено mock-up головної сторінки застосунку. Дана сторінка доступна користувачеві, який щойно зайшов на сайт, перейшовши за посиланням у своєму браузері. На головній сторінці знаходиться список

Кафедра інженерії програмного забезпечення  
 Вебсервіси підтримки продажів промислових товарів  
 категорій, які користувач може обирати для пошуку потрібного товару. Також  
 на сторінці наявний список останніх придбаних товарів.

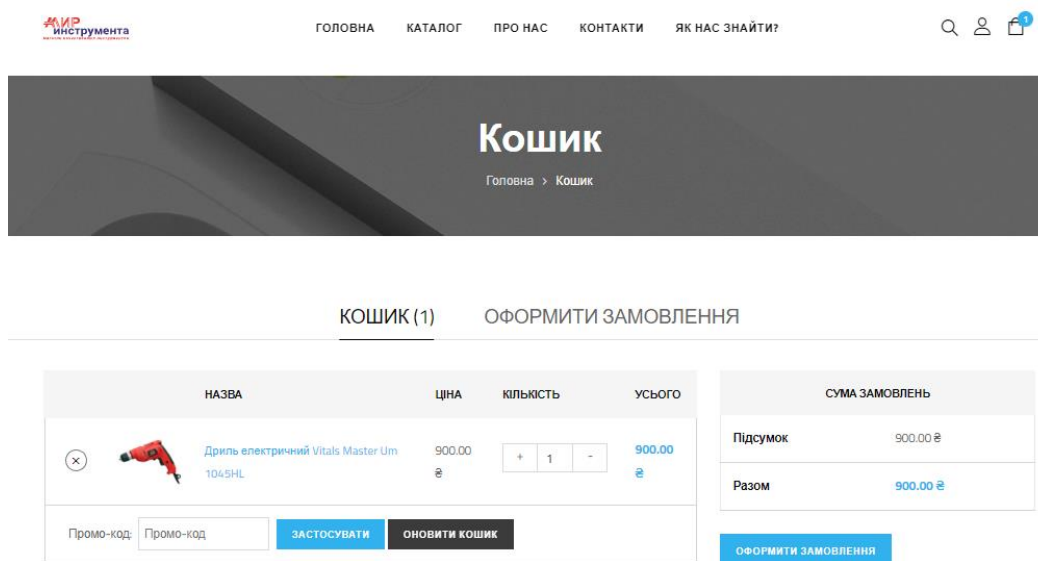
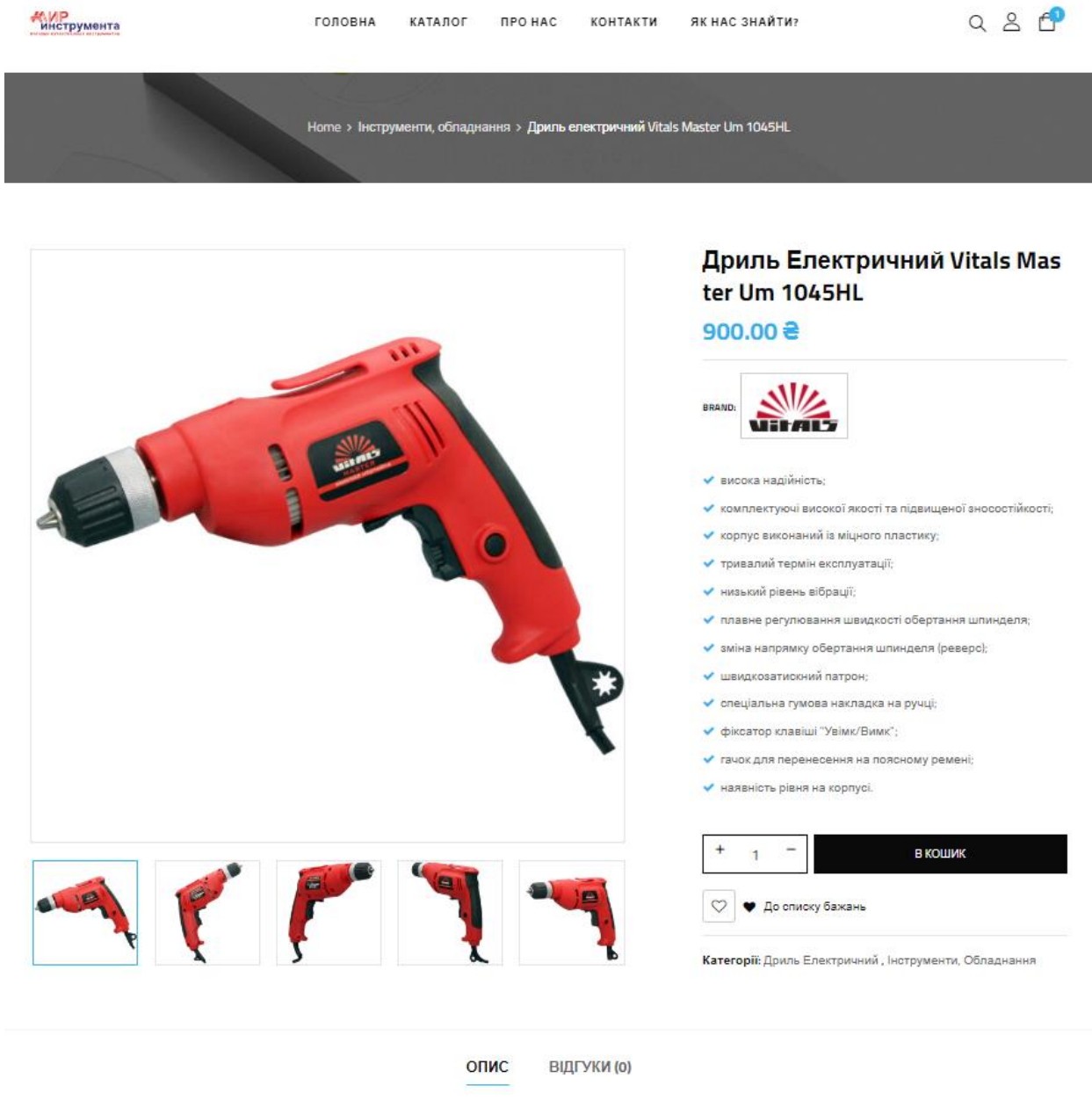


Рисунок 3.3 – Сторінка "Кошик"


На даній сторінці, представлено макет «Кошик». Ту буде зосереджено всі товари, які клієнт буде додавати до свого замовлення. На цій сторінці знаходиться список товарів, які були додані до нього, під час перегляду загального списку представлених продуктів. Тут можливо відкоригувати кількість одиниць товару, які клієнт бажає замовити. Також представлено можливість вводу промокоду, що буде сприяти зменшенню ціни, на сторінці показано таблицю з характеристиками замовлення тобто, ціна за одиницю товару, знижка, та загальна вартість усього замовлення. Клієнт може видаляти товари, які були додані раніше.



Home > Інструменти, обладнання > Дриль електричний Vitalis Master Um 1045HL

### Дриль Електричний Vitalis Master Um 1045HL

900.00 €

BRAND: 

- ✓ висока надійність;
- ✓ комплектуючі високої якості та підвищеної зносостійкості;
- ✓ корпус виконаний із міцного пластику;
- ✓ тривалий термін експлуатації;
- ✓ низький рівень вібрації;
- ✓ плавне регулювання швидкості обертання шпинделя;
- ✓ зміна напрямку обертання шпинделя (реверс);
- ✓ швидкозатискний патрон;
- ✓ спеціальна гумова накладка на ручці;
- ✓ фіксатор клавіші "Увімк/Вимк";
- ✓ гачок для перенесення на поясному ремені;
- ✓ наявність рівня на корпусі.

+ 1 - В КОШИК

До списку бажань

Категорії: Дриль Електричний, Інструменти, Обладнання

[ОПИС](#) [ВІДГУКИ \(0\)](#)

Рисунок 3.4 – Сторінка продукту

На сторінці продукт, представлено інформацію, яка стосується одного конкретного товару, який був обраний клієнтом. Тут можливо здійснити наступні дії:

1. переглянути фотоматеріал товару;
2. вибрати кількість одиниць товару;
3. переглянути опис товару (наявний короткий та розгорнутий опис);
4. додати даний товар до кошику;
5. додати даний товар до списку вподобань.

Кафедра інженерії програмного забезпечення  
Вебсервіси підтримки продажів промислових товарів

На сторінці також наявний слайдер з товарами, які часто купуються іншими клієнтами разом з даним товаром.

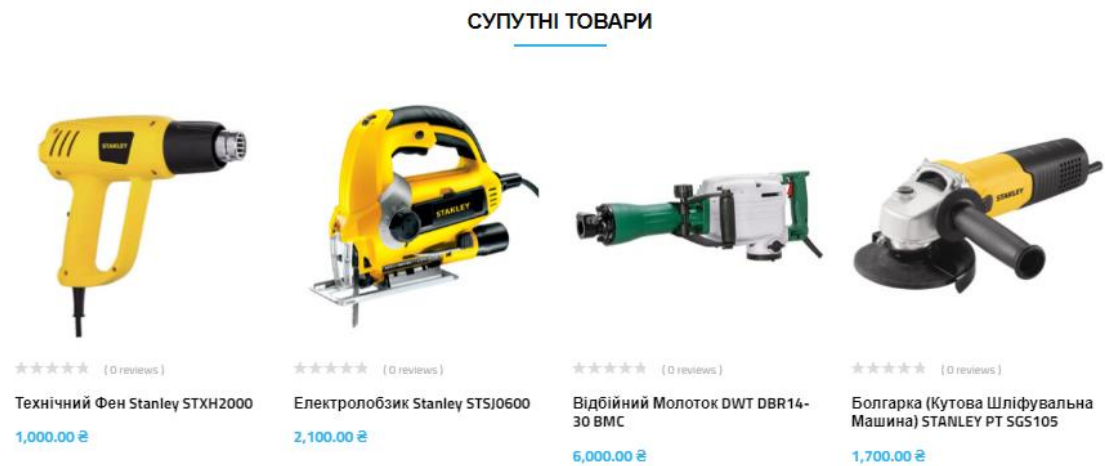
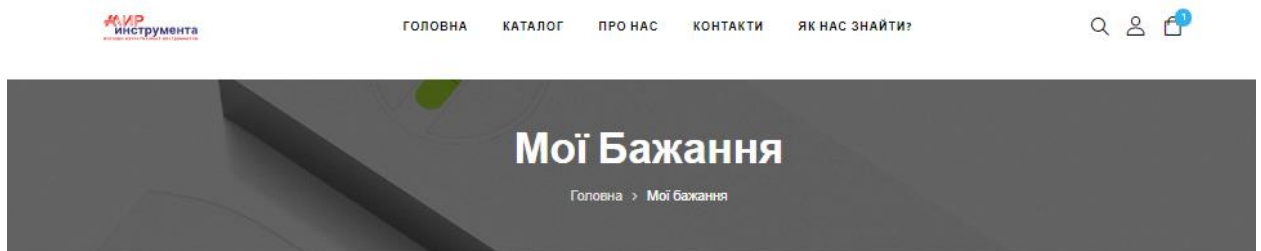


Рисунок 3.5 – Продовження сторінки, слайдер



Мій список бажань



НАЗВА	
	Інвертор ВДІ-200Е DC MMA
	Дриль електричний Vitals Master Um 1045HL
	Дриль електричний Vitals Professional Et 2105YEadk
	DWT Акумуляторний реноватор AMS-10,8 Li BMS

Рисунок 3.6 – Сторінка "Список бажань"

На сторінці «Список бажань», знаходяться товари, які переглядав клієнт та власноруч додав їх до даного списку за допомогою спеціальної кнопки. Сторінка створена для того, щоб клієнт мав змогу відокремити декілька товарів для подальшого ознайомлення з ними. Тут можна лише видаляти товари або переходити на їх сторінки з описом.

### 3.2 Діаграма класів

Діаграми класів (англ. Class diagram) – являє собою набір статичних, декларативних елементів моделі. Представляє такі елементи, як: класи, типи даних, зміст та їх відношення. Вона найкраще та найбільш розгорнуто представляє зв'язки в програмному коді. Це один із видів діаграм в UML.

UML (Unified Modeling Language) – уніфікована мова моделювання. Дана мова моделювання є загальною у всьому світі, й використовується багатьма розробниками. Перевага даної мови в тому, що вона буде зрозуміла кожному розробнику, який знайомий з даною графічною мовою, не залежно від країни та рідної мови розробника. Використання даної мови моделювання, означає створення моделі, яка описує об'єкт [10]. У своєму описі вона містить наступні елементи:

- класи;
- атрибути;
- методи;
- взаємозв'язки між об'єктами.

Для даного проекту також була розроблена діаграма класів, яка в подальшому буде використовуватись для створення готового вебсервісу.



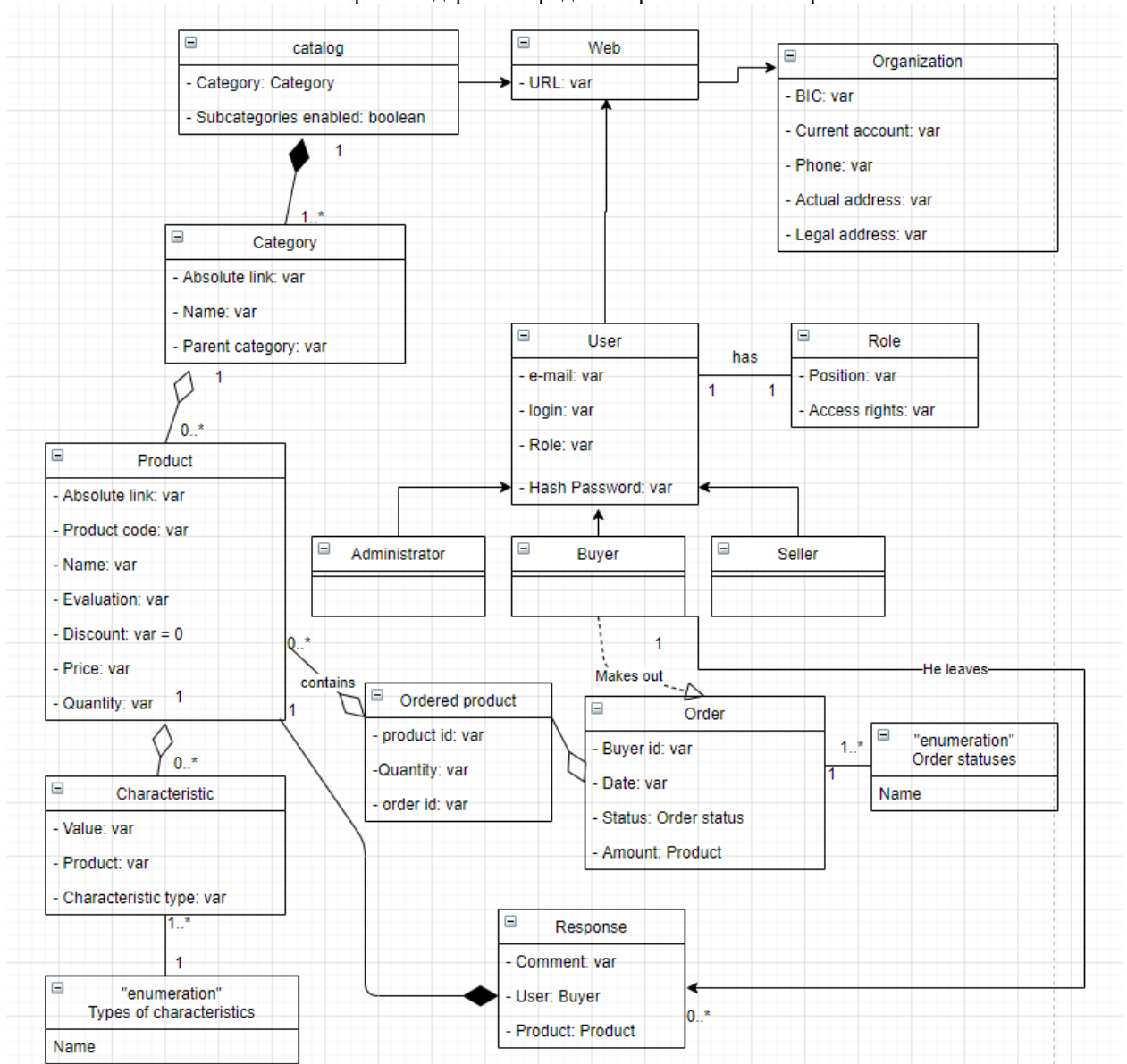


Рисунок 3.7 – Діаграма класів

### 3.3 Діаграма послідовності

Діаграма послідовності – наочне зображення порядку взаємодії певних об'єктів програми у часі. По суті, в даній діаграмі повинно бути продемонстровано, як користувачі системи (актори з діаграм використання) взаємодіють з різними компонентами системи під час роботи з ними, а також повинна показувати, як при цьому різні компоненти системи взаємодіють між собою. Зазвичай, кожна діаграма послідовності описує один з варіантів використання, які зазначені у use-case діаграмі. Основною перевагою даного

виду діаграм, являється можливість показати взаємодію компонентів та зробити опис потоків повідомлень від одних компонентів до інших.

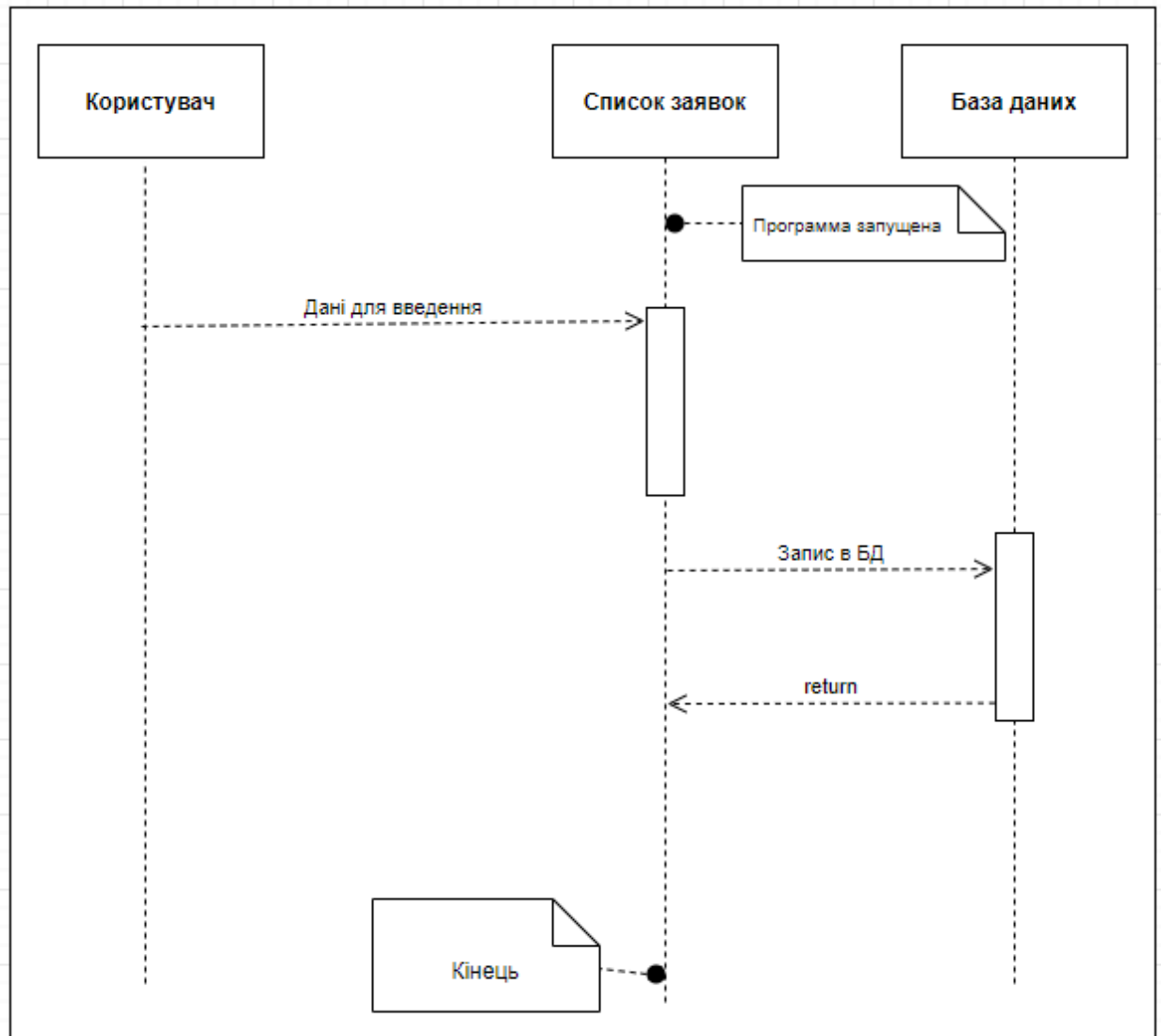


Рисунок 3.8 – Діаграма послідовності для процесу коментування

На рисунку 3.8 зображено діаграму послідовності для процесу коментування. Вона показує, як система реагує на залишений коментар користувачем.

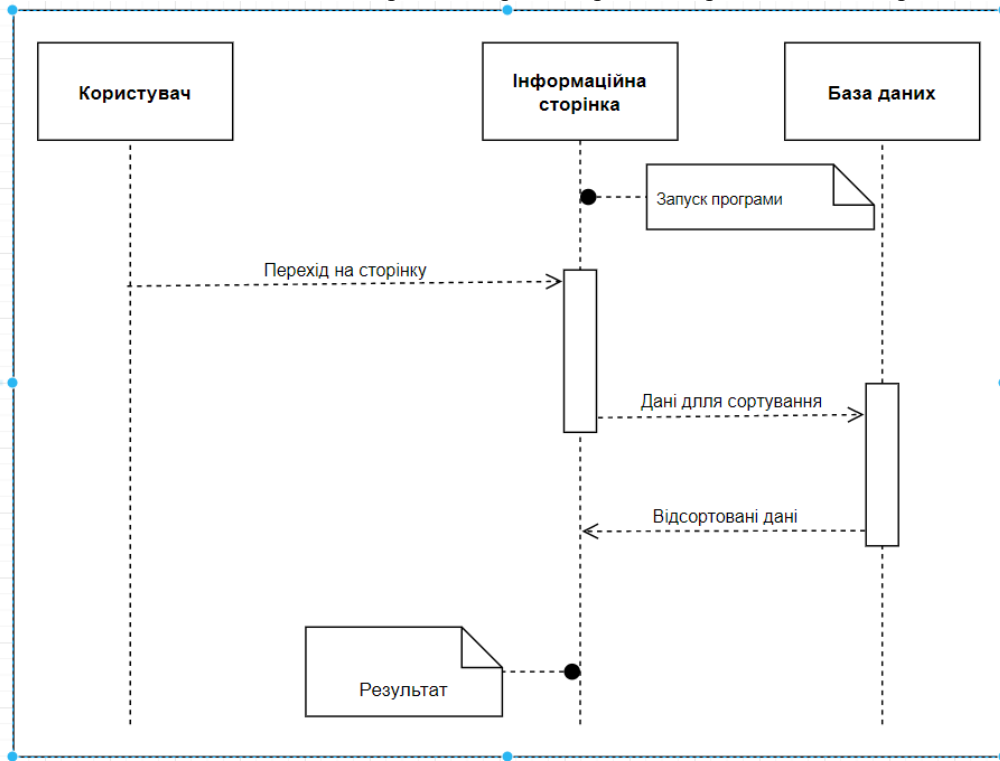


Рисунок 3.9 – Діаграма послідовності перегляду інформації

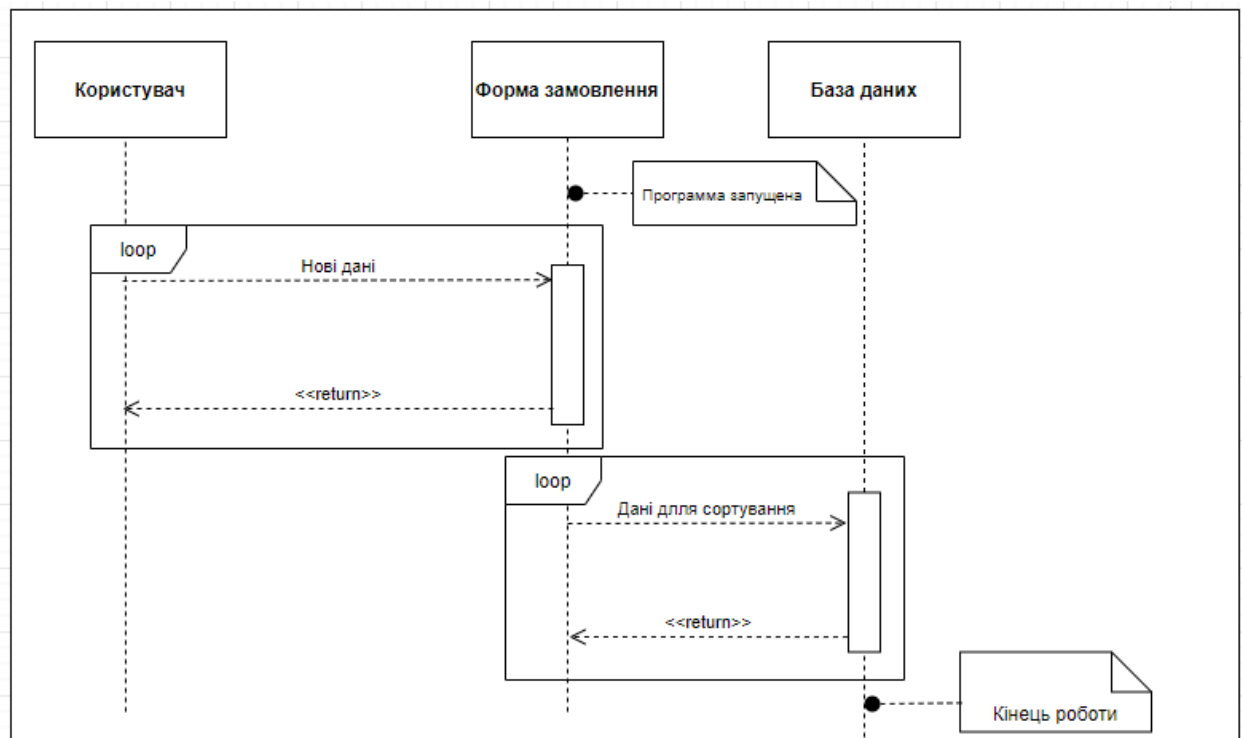


Рисунок 3.10 – Діаграма послідовності оформлення замовлення

На рисунку 3.10 зображено принцип оформлення замовлення.

### 3.4 Діаграми розгортання бази даних

Діаграма розгортання БД – це діаграма в мові UML, яка зображує обчислювальні вузли під час роботи програми, об'єкти та компоненти, які виконуються на даних вузлах.

Кожен вузол повинен мати своє унікальне ім'я, яке буде відрізнити його від інших. Ім'я вузла це текстовий рядок. Таке ім'я називають – простим. Проте існує складене ім'я, або як ще називають – кваліфіковане. Це таке ім'я, до якого спереду додається ім'я пакета в якому воно знаходиться.

Існує декілька видів вузлів:

- вузол пристроїв;
- вузол середовища виконання.

Вузол середовища виконання – обчислювальний ресурс, який працює в середні зовнішнього вузла та який являє собою сервіс, що виконує інші виконувані програмні елементи [12].

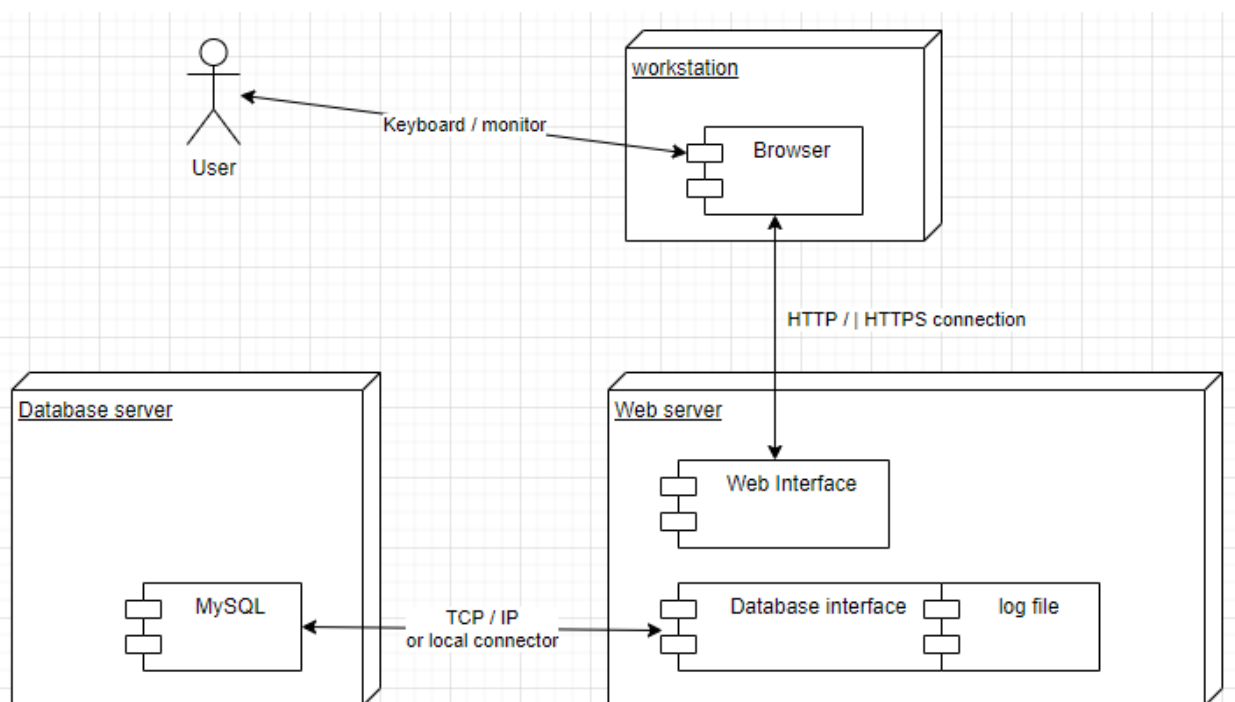


Рисунок 3.11 – Діаграма розгортання БД

На рисунку 3.11 зображено діаграму розгортання БД, яка використовується в даному проекті.

### **Висновки до розділу 3**

У ході написання 3 розділу, було продовжено проєктування системи, що розробляється. Представлено mock-up зображення, які показують фінальний вигляд застосунку. Було представлено діаграму класів, з повним описом її компонентів, що зображує внутрішній принцип роботи застосунку. Було змодельовано діаграми послідовності, які зображують принцип послідовності роботи окремо взятого процесу у системі, описано кожний із представлених, процес. Було також представлено, описано та побудовано діаграму розгортання, яка показує принцип роботи застосунку та взаємодію його з базою даних.

Виходячи з вище представлених даних, та опираючись на пророблену роботу можна зробити висновок, що проєктування системи вважається завершеним, що свідчить про початок безпосередньої розробки застосунку. Дані, які представлені у розділах 1-3, будуть використовуватись під час розробки системи й чітко притримуватись поставленим вимогам у даних розділах.

## 4 РОЗРОБКА ЗАСТОСУНКУ. ПРОГРАМНА РЕАЛІЗАЦІЯ.

### 4.1 Архітектура застосунку

Сайт буде складатись з декількох частин, а саме адміністративної та користувацької.

Адміністративна частина буде складатись з елементів, які будуть надавати можливість редагування, додавання та видалення різного роду інформації. За допомогою адміністративної частини сайту можна буде здійснювати зміни над: статичними та динамічними сторінками, вносити редакції до розділів та блоків сайту.

Користувацький інтерфейс буде складатись з елементів, які будуть сприяти зручному користуванню застосунком. Він повинен бути інтуїтивно зрозумілий. Він повинен бути побудований згідно вимог представлених у попередніх розділах.

Веб-застосунок повинен складатись з наступних розділів:

- сторінка «Каталог» – містить категорії:
  - All;
  - Бензо-коси;
  - Бензопили;
  - Бетономішалки;
  - Дриль електричний;
  - Інструменти та обладнання;
  - Компресори;
  - Краско-пульти;
  - Лазерна вимірювальна техніка;
  - Зварювальні апарати;
  - Електро-коси.
- Сторінка «Про нас» – текстова сторінка;

Кафедра інженерії програмного забезпечення  
 Вебсервіси підтримки продажів промислових товарів

- Сторінка «Доставка та оплата» – текстова сторінка;
- Сторінка «Зворотній Зв'язок» – текстова сторінка, форма для заповнення.
- Сторінка «Кошик» – сторінка з відображенням кількості товарів для замовлення.

## 4.2 Програмна реалізація

Для роботи застосунку потрібно місце його розміщення, тобто сервер та хостинг, на якому і буде розміщений сам веб-сайт.

Для даного проекту було обрано один з самих відомих сайтів, який надає послуги хостингу в Україні – <https://www.ukraine.com.ua/>

The screenshot shows the 'Хостинг Україна' website with a navigation menu and a main banner. Below the banner is a search bar for domains and a table comparing three hosting plans: 'Быстрый', 'Лучший', and 'Качественный'.

	Быстрый	Лучший	Качественный
Место на NVMe-диске	10 Gb	20 Gb	30 Gb
Бесплатный SSL-сертификат	✓	✓	✓
Конструктор сайтов	✓	✓	✓
Apache, OpenLiteSpeed	✓	✓	✓
Сайтов	1	5	неограниченно
Memory limit (RAM)	512 Mb	1024 Mb	1536 Mb
Расположение серверов	🇺🇦 🇵🇱 🇩🇪 🇬🇧 🇮🇹 🇪🇺	🇺🇦 🇵🇱 🇩🇪 🇬🇧 🇮🇹 🇪🇺	🇺🇦 🇵🇱 🇩🇪 🇬🇧 🇮🇹 🇪🇺
Цена	от 109 грн/мес	от 192 грн/мес	от 341 грн/мес

Рисунок 4.1 – Головна сторінка "Хостинг Україна"

Для початку потрібно зареєструвати аккаунт. Потім перейти до вкладки «Панель управління» та здійснити ряд налаштувань, без яких робота сайту не можлива. Спочатку потрібно зареєструвати домен, і тільки після цього створити сайт. Для того, щоб сайт став доступний усім користувачам мережі потрібно прив'язати домен до хостингу. Для цього потрібно:

- Вказати хостингову DNS адресу;
- Додати хостинг до головної панелі управління хостингу.

## 4.2.1 Встановлення CMS WordPress

Після налаштування хостингу та домену потрібно завантажити CMS систему. Як було описано вище в даному проекті використовується CMS WordPress. За допомогою різних допоміжних сервісів, можна встановити потрібну CMS, проте це можливо здійснити і вручну. Також для встановлення доступні чимала кількість інших систем керування вмістом даних: Joomla, MODx, OpenCart, Drupal, OkayCMS, Moodle, October CMS, Grav та інші.

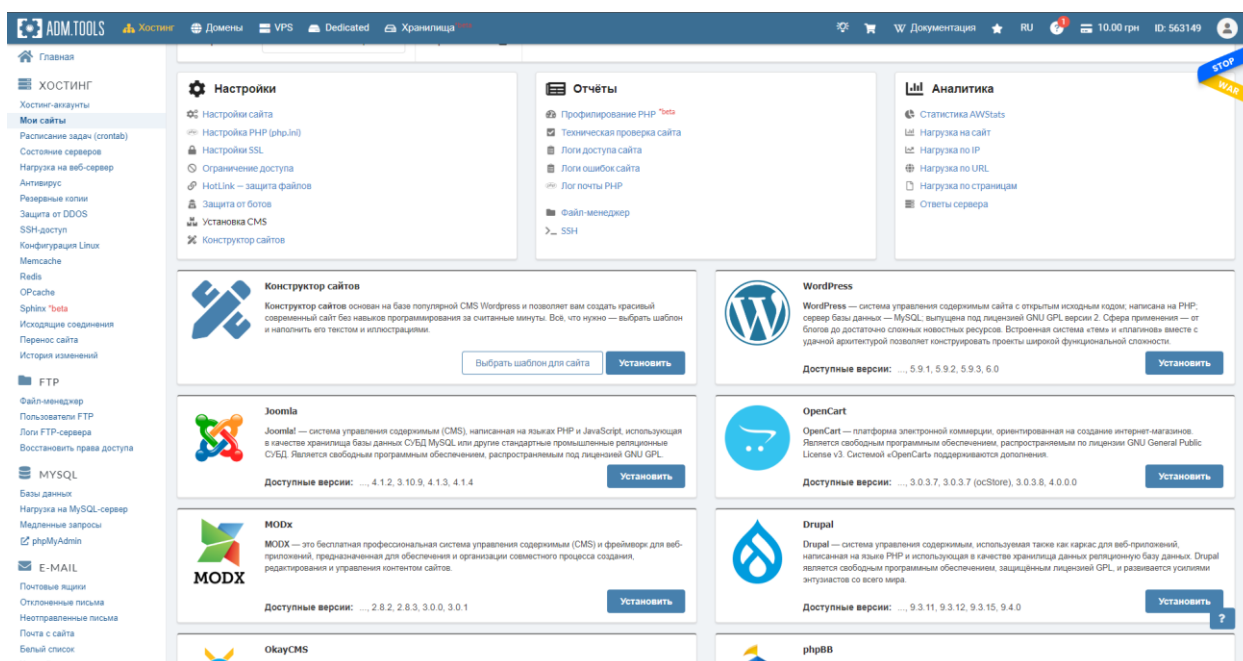


Рисунок 4.2 – Вікно вибору CMS

Після встановлення WordPress, потрібно провести ще ряд важливих налаштувань: вибрати адресу сайту та БД, створити логін, пароль, ввести електронну пошту адміністратора, назву сайту та бажану мову.



**Автоматическая установка WordPress на [www.planeta-instrumenta.com](http://www.planeta-instrumenta.com)**

**Внимание!**  
Корневой каталог сайта будет очищен перед установкой CMS.  
Данная CMS требует версию PHP не ниже 7.4. Будет установлена минимальная доступная версия: PHP 7.4

Название сайта:

E-mail администратора:   
[mirinstrumenta.nikolaev@gmail.com](mailto:mirinstrumenta.nikolaev@gmail.com) [admin@planeta-instrumenta.com](mailto:admin@planeta-instrumenta.com)

Логин администратора:

Пароль администратора:

Версия CMS:

Язык:

База данных:

Очистить базу данных перед установкой CMS

www.planeta-instrumenta.com"/>

Рисунок 4.3 – Завантаження та налаштування CMS

Після встановлення та первинного налаштування особового запису, потрібно встановити WP Emerge. Для встановлення WP Emerge потрібен Composer.

Composer – менеджер пакетів для мови програмування PHP, який забезпечує стандартний формат для управління залежностями у ПЗ та необхідних бібліотек [12].

Для встановлення WP Emerge потрібно встановити у папку з темами WordPress, стартову тему, та створити файл конфігурації config.json, за основу якого взято файл config.json.dist. Потім потрібно виконати команду:

- composer install.

Встановлення основних компонентів мікро-фреймворку може зайняти деякий час. Після встановлення, потрібно перейти на адміністративну панель, та активувати тему.

Для повноцінної роботи сайту в сфері електронної комерції потрібно встановити відповідні плагіни. У розділі 1, було описано, який саме плагін був обраний, та зазначено основні характеристики та переваги плагіну WooCommerce.

### 4.3 Структура проекту

Після завершення встановлення та всіх налаштувань, можна розглянути загальну структуру проекту.

Каталоги наявні в проекті:

- app/helpers/

Каталог для додавання допоміжних файлів PHP, які містять лише визначення функцій.

- app/setup/

Каталог для зміни файлів. Файли, які містять декларації та реєстрації об'єктів WordPress (типи публікацій, таксономії).

- app/src/

Служить для додавання файлів PHP. Усі класи в області App\namespace будуть автоматично завантажуватись згідно стандарту PSR-4.

- resources/images/

Каталог, який служить для додавання зображень.

- resources/styles/theme/

Каталог який служить для додавання файлів .css та .scss. В подальшому додаються до інтерфейсного пакету.

- resources/styles/[admin,editor,login]/

Дані директорії призначені для адміністраторів, редакторів, груп входу.

- resources/scripts/theme/

Каталог, який потрібен для додавання JavaScript-файлів. Потім додаються до зовнішнього блоку. Точка вводу – index.js.

- resources/scripts/[admin,editor,login]/
- Дані директорії призначені для адміністраторів, редакторів, груп входу. По принципу роботи схожі з resources/styles/[admin,editor,login]/
- resources/scripts/theme/directory theme/.

Каталог, який використовується для додавання view теми у heme/partials/ чи theme/templates/. Слід уникати додавання логіки PHP, якщо це не стосується макетування (логіку PHP потрібно вписувати в допоміжні

файли або контролери WP Emerge).

### 4.3.1 Розробка основного функціоналу

Представлений нижче код відображує маршрути з можливістю інтеграції правила перезапису.

```
Router::get( '/', 'HomeController@index' );
Router::get( '/custom', 'CustomController@custom' )->query( function ( $query_vars ) {
return [ // WP_Query query vars ... ];
} );
Router::get(
['post_id', get_option('page_on_front')], 'HomeController@index' );
Router::get( function() {
return is_front_page();
}, 'HomeController@index' );
```

Рисунок 4.4 – Приклад функціонального коду №1

Далі наведено код, який служить для забезпечення відображення сторінок сайту.

```
class HomeController {
public function index( $request ) {
$name = $request->get( 'name' );
return app_view( 'templates/home.php' ) ->with( [
'welcome' => 'Welcome, ' . $name . '!',
] );
} }
}
```

Рисунок 4.5 – Код створеного контролеру WP Emerge

Наступний код служить для того, щоб задати маршрут доступу до головної сторінки.

```
Router::get( '/', 'HomeController@index' ) ->add( function( $request, $next ){
    // perform action before
    $response = $next( $request );
    // perform action after
    return $response;
} );
```

Рисунок 4.6 – Посередник WP Emerge

Для того, щоб можна було переглядати весь вміст сайту, тобто контент, яким він наповнюється було створено макети перегляду.

```
index.php
<?php
get_header(); ?>
<div class="store-container">
    <div id="primary" class="content-area">
        <main id="main" class="site-main" role="main">
            <?php
if ( have_posts() ) :
    while ( have_posts() ) : the_post();
        get_template_part( 'template-parts/content', get_post_format() );
        endwhile;
        the_posts_pagination();
    else :
        get_template_part( 'template-parts/content', 'none' );
    endif; ?>
        </main><!-- #main -->
    </div><!-- #primary -->
    <?php get_sidebar('right'); ?>
</div>
<?php get_footer()
```

Рисунок 4.7 – Макет перегляду контенту

### 4.3.2 Додатковий функціонал

Для зручності користування сайту, було розроблено додатковий функціонал, такий, як відображення популярних товарів, іншими словами – продукція, яка має найбільшу кількість замовлень за останній період. Для створення такого віджету, було використано відкритий шорткод, в плагіні WooCommerce.

[best\_selling\_products] – основна команда віджету.

Цей шорткод потрібно додати до загального макету застосунку, що автоматично створе віджет на сторінці, в яку було додано його, та буде

показано товари за період обраний власником, це може бути один день, тиждень, місяць або квартал.

Також було прийнято міри, щодо удосконалення захисту сайту від хакерських атак. Існує думка, що безпека WordPress не досконала, через відкритий код та чималу кількість відомої інформації про цю платформу в інтернеті, тому було прийнято рішення розробити трьох рівневу автентифікацію до доступу в адміністративну панель управління.

Через значну популярність та кількість інформації в мережі, багато кому відомий стандартна адреса сторінки входу до адмін-панелі WordPress. З адмін-панелі, можна отримати доступ до всіх налаштувань, приватної інформації та до коду. Це і є причина, чому ж його намагаються зламати. Було прийнято рішення реалізувати зміну адреси входу до адмін-панелі.

Це буде працювати наступним чином, у разі входу зловмисником за стандартною адресою, його буде перенаправляти на сторінку з помилкою, тобто буде сказано, що даної сторінки не існує.

Для коректної роботи, потрібно, по-перше – додати на початок файлу wp-config.php такий фрагмент коду:

```
define('WP_ADMIN_DIR', 'mirshop');
define('ADMIN_COOKIE_PATH', 'SITECOOKIEPATH' . 'WP_ADMIN_DIR');
```

Рисунок 4.8 – Код зміни адреси входу

По-друге – потрібно в кінець файлу wp-includes/functions.php додати спеціальний фільтр:

```
add_filter('site_url', 'wadmin_filter', 10, 3);
function wadmin_filter( $url, $path, $orig_scheme ) {
    $old = array( "/(wp-admin)/" );
    $admin_dir = WP_ADMIN_DIR;
    $new = array($admin_dir);
    return preg_replace( $old, $new, $url, 1);
}
```

Рисунок 4.9 – Код фільтру

Та прописати перетворення в файлі .htaccess:

RewriteEngine On

І потім, потрібно вставити даний код:

```
RewriteRule ^mirshop/(.*) wp-admin/$1?%{QUERY_STRING} [L].
```

Все, адресу для входу було створено, проте залишилось видалити попередню адресу. Для цього також потрібно використати простий фільтр, та додати його в кінець файлу wp-includes/functions.php.

```
add_action( 'init', 'block_wp_admin' );  
function block_wp_admin() {  
    if(strpos($_SERVER['REQUEST_URI'],'wp-admin') != false){  
        wp_redirect( home_url().'/404' );  
        exit;  
    }  
}
```

Рисунок 4.10 – Код для блокування стандартної точки входу

Після виконання даних маніпуляцій, стандартна адреса входу, була змінена з wp-admin на mirshop.

Проте це ще не всі корективи до забезпечення безпеки та конфіденційності особистих даних. Наступним кроком буде створення захисту від спеціальних програм по автоматичному підбору паролю. Було розроблено спеціальний плагін, який буде відслідковувати всі невдалі спроби авторизації, та зберігати дані про діапазон IP-адрес, а у разі перевищення максимальної кількості спроб авторизації, даний діапазон IP-адрес буде блокуватись. Це дасть змогу не тільки запобігати небажаного доступу, але й буде повідомляти додаткову інформацію для подальшого реагування.

Даний плагін буде запобігати злому пароля за допомогою програм підбору пароля. В даний час налаштування плагіну наступні: у разі 3 невдалих спроб, блокування відбувається на 1 годину, якщо сценарій повторюється декілька раз, відбувається перманентне блокування. Для подальшого розблокування, потрібно ручне втручання.

#### 4.4 Тестування

Для забезпечення правильної роботи застосунку, всіх його компонентів та сторінок потрібно провести етап тестування. Даний етап характерний перевіркою кожного функціонального компоненту системи на працездатність та коректність виконання його поставленої задачі.

Таблиця 4.1 – Реєстрація

Діючі особи	Користувач, система
Мета	Реєстрація облікового запису
Передумова	Користувач не зареєстрований
Успішний сценарій:	
<ul style="list-style-type: none"> <li>- Користувач переходить до вікна реєстрації;</li> <li>- Користувач вводить дані в спеціальні діалогові вікна;</li> <li>- Система зберігає дані;</li> <li>- Система перенаправляє користувача на головну сторінку;</li> </ul>	
Сценарій успішний. Обліковий запис користувача зареєстровано.	
Розширення	
1a	Користувач не заповнив всі обов'язкові поля для реєстрації. Система виводить повідомлення про помилку. Результат: обліковий запис не зареєстровано.
Усі сценарії розширення успішно виконані.	

Таблиця 4.2 – Авторизація

Діючі особи	Користувач, система
Мета	Авторизація
Передумова	Користувач не авторизований, користувач зареєстрований

## Продовження таблиці 4.2

Успішний сценарій:	
<ul style="list-style-type: none"> <li>- Користувач переходить до вікна авторизації;</li> <li>- Користувач вводить логін та пароль;</li> <li>- Система перевіряє введені дані;</li> <li>- Система перенаправляє користувача на головну сторінку;</li> </ul>	
Сценарій успішний. Користувач авторизований.	
Розширення	
1a	Користувач увів не коректні дані. Система виводить повідомлення про помилку. Результат: користувач не авторизований
Усі сценарії розширення успішно виконані.	

Таблиця 4.3 – Сортування товару

Діючі особи	Користувач, система
Мета	Знайти товар за характеристикою
Успішний сценарій:	
<ul style="list-style-type: none"> <li>- Користувач переходить на сторінку вибору товарів;</li> <li>- В спеціальному вікні вводить характеристику товару(назва, тип, бренд, ціна, виробник);</li> <li>- Система фільтрує введені дані;</li> <li>- Система виводить товари, які мають схожу характеристику.</li> </ul>	
Сценарій успішний. Виведено список конкретних товарів.	
Розширення	
1a	Товару не існує, або закінчився. Система повідомляє про помилку сортування. Результат: Потрібного товару не знайдено.
Усі сценарії розширення успішно виконані.	



Таблиця 4.4 – Оформлення замовлення

Діючі особи	Користувач, система
Мета	Оформлення замовлення
Успішний сценарій:	
<ul style="list-style-type: none"> <li>- Користувач обрав потрібний товар;</li> <li>- Перейшов до вікна оформлення замовлення;</li> <li>- Заповнив форму замовлення;</li> <li>- Система виводить інформацію стосовно конкретного замовлення та просить підтвердити;</li> <li>- Користувач підтверджує замовлення.</li> </ul>	
Сценарій успішний. Замовлення зареєстровано в системі.	
Розширення	
1a	Користувач не підтвердив замовлення. Система перенаправляє користувача до загального каталогу. Результат: замовлення не зареєстровано.
Усі сценарії розширення успішно виконані.	

Таблиця 4.5 – Редагування каталогу товарів

Діючі особи	Користувач, система
Мета	Редагувати каталог товарів
Передумова	Користувач авторизований зі статусом Адміністратор
Успішний сценарій:	
<ul style="list-style-type: none"> <li>- Користувач переходить на сторінку редагування каталогу товарів в адмін-панелі;</li> <li>- Обирає потрібний елемент зі списку;</li> <li>- Вносить корективи;</li> <li>- Зберігає дані;</li> <li>- Система зберігає дані до БД;</li> </ul>	

## Продовження таблиці 4.5

Сценарій успішний. Були внесені корективи до каталогу.	
Розширення	
1a	Користувач не зберіг корективи. Система не вносить зміни до БД. Результат: Каталог товарів залишився без змін
2a	Користувач не заповнив всі обов'язкові поля. Система виводить повідомлення про помилку. Результат: Каталог товарів залишився без змін
Усі сценарії розширення успішно виконані.	

Усі етапи тестування були успішно проведені, результати тестування наведено в таблицях 4.1-4.5.

#### Висновок до розділу 4

В процесі виконання 4 розділу було розроблено сам застосунок, який проєктувався у розділах 2 та 3.

Було описано та показано кожен етап проєктування, наведені короткі фрагменти коду до спеціальних частин, які цього потребували. Було представлено інформацію, як саме потрібно налаштовувати веб-сайт, для коректної роботи та представлено реалізацію даного етапу розробки. У підрозділі 4.1 було представлено архітектуру та основні складові проєкту у вигляді списку.

У підрозділі 4.2 було наведено програмну реалізацію даного проєкту. Представлено дані щодо використання хостингу, системи управління вмістом. Показано та описано, як саме потрібно встановлювати CMS сервіси та правильно вбудовувати їх у проєкт.

У підрозділі 4.3 було наведено інформацію стосовно структури проєкту, його основні каталоги та їх короткий опис, принципи їх роботи. Також було показано приклади, як основного функціоналу так і допоміжного, кожен представлений окремо елемент функціоналу систему супроводжувався коротким уривком коду.

У підрозділі 4.4 було представлено етап тестування та наведено результат проробленої роботи у вигляді таблиць, які чітко показують правильність роботи системи при тих чи інших умовах.

**ВИСНОВКИ**

У ході виконання кваліфікаційної роботи бакалавра, було створено вебсервіс підтримки продажу промислових товарів, що сприяє підвищенню показників у сфері електронної комерції.

Для досягнення поставленої мети було:

- проаналізовано предметну область сфери електронної комерції та розробки веб-застосунків;
- проведено роботи для створення спеціальних вимог щодо розробки даної інформаційної системи на основі отриманих даних, під час вивчення предметної сфери;
- спроектовано інформаційну систему, згідно вимог, які були представлені;
- реалізовано вебзастосунок, проведено тестування та виконано первинне налаштування системи для коректної роботи сервісу.

При аналізі актуальності даної теми кваліфікаційної роботи бакалавра було визначено доцільність розробки даної системи, у світлі останніх подій у світі, а саме коронавірусної інфекції та війни в Україні. Було проаналізовано існуючі аналоги системи та сформовано ряд функціональних та не функціональних вимог даної інформаційної системи. В результаті проробленої роботи було представлено технічне завдання та специфікацію вимог до розроблюваного застосунку.

Наступним етапом було розглянуто етапи створення застосунку. Було наведено етапи проектування системи та представлено обґрунтування вибору тих чи інших рішень у ході подальшої розробки системи. Показано, яку саме систему керування вмістом було обрано, та наведено причини вибору саме даної системи. Описано принцип вибору СКБД, та наведено основні можливості та характеристики даної системи, що була обрана. Було розроблено сценарії використання даної інформаційної системи та представлено їх у вигляді таблиць. Наступним кроком було розроблено

різного роду діаграми, для більш конкретного зображення роботи майбутньої системи. Наведено діаграму прецедентів та контекстну діаграму. Наведено опис кожної з них.

У 3 розділі було представлено процес моделювання даної системи, тобто наведено приклад майбутнього інтерфейсу системи, та представлений опис кожного зі скріншотів. Наведено діаграму класів, яка використовувалась у подальшій розробці. Описано та створено діаграми послідовності, для більш чіткого розуміння роботи тих чи інших окремо взятих елементів системи. Було представлено наочне зображення діаграми розгортання, що описує принцип сумісної роботи вебсервісу та БД.

У четвертому розділі було описано безпосередньо етап розробки застосунку. В даному розділі використовувались матеріали, які були зазначені у попередніх трьох розділах. Тут було представлено архітектуру застосунку. Описано програмну реалізацію проєкту, тобто представлено інформацію, щодо вибору хостингу, та короткий опис встановлення CMS системи. Представлено Структуру проєкту, та короткий опис основних каталогів системи, які були задіяні найбільше. Наведено приклади, як основного функціоналу системи так і додаткового, кожен елемент функціоналу, який був представлений у розділі 4 супроводжувався коротким фрагментом коду. В кінці роботи було проведено успішне тестування застосунку, що показує коректність роботи системи.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Особливості функціональних вимог та не функціональних вимог [Електронний ресурс] / 1 – Режим доступу до ресурсу: <https://uk.myservername.com/sap-crm-interview-questions>. (дата звернення: 22.05.2022)
2. Структура інтернет-магазину: ключові сторінки і розділи Джерело інформації: <https://xn--90aamhd6acpq0s.xn--j1amh/teoriya/struktura-internet-magazinu> [Електронний ресурс]. – 1209. – Режим доступу до ресурсу: <https://xn--90aamhd6acpq0s.xn--j1amh/teoriya/struktura-internet-magazinu/>. (дата звернення: 22.05.2022)
3. Інтернет-магазин: етапи створення та основні переваги [Електронний ресурс]. – 2003. – Режим доступу до ресурсу: <https://armedsoft.com/ua/blog/internet-magazyn-etapy-stvorennya-ta-osnovni-perevagy>. (дата звернення: 23.05.2022)
4. Електронна комерція – Вікіпедія. Вікіпедія. URL: [https://uk.wikipedia.org/wiki/Електронна\\_комерція](https://uk.wikipedia.org/wiki/Електронна_комерція) (дата звернення: 25.05.2022).
5. Інформаційна система – Вікіпедія. Вікіпедія. URL: [https://uk.wikipedia.org/wiki/Інформаційна\\_система](https://uk.wikipedia.org/wiki/Інформаційна_система) (дата звернення: 23.05.2022).
6. Етапи створення веб-сайтів - Все про веб-сайти. Google Sites: Sign-in. URL: <https://sites.google.com/site/siteingschool/etapi-stvorennja-veb-sajtiv> (дата звернення: 23.05.2022).
7. Структура сайтів опис різних структур. URL: <https://jak.koshachek.com/articles/struktura-sajtiv-opis-riznih-struktur.html> (дата звернення: 23.05.2022).
8. Система керування вмістом – Вікіпедія. Вікіпедія. URL: [https://uk.wikipedia.org/wiki/Система\\_керування\\_вмістом](https://uk.wikipedia.org/wiki/Система_керування_вмістом) (дата звернення: 25.05.2022).
9. Що таке CMS сайту. URL: <https://hostiq.ua/wiki/cms/> (дата звернення: 25.05.2022).
10. PhpMyAdmin – вікіпедія. Вікіпедія. URL: <https://uk.wikipedia.org/wiki/PhpMyAdmin> (дата звернення: 27.05.2022).
11. Делиль М. Освоєння phpMyAdmin 3.4 для ефективного керування MySQL : підручник. Packt Publishing, 2012. 394 с.
12. Діаграма розгортання. URL: <https://jak.koshachek.com/articles/diagrama-rozgotannja-studopedija.html> (дата звернення: 26.05.2022).

13. Composer – вікіпедія. Вікіпедія. URL:  
<https://uk.wikipedia.org/wiki/Composer> (дата звернення: 27.05.2022).
14. Порівняння безкоштовних CMS: Wordpress, Joomla, Drupal та ін. Який двигун вибрати для створення своєї гри Найкращі cms. take-box.  
URL: <https://take-box.ru/uk/firmware/sravnenie-besplatnyh-cms-wordpress-joomla-drupal-i-dr-kakoi-dvizhok-vybrat-dlya-sozdaniya.html> (дата звернення: 29.05.2022).
15. Діаграма прецедентів. www.wiki.uk-ua.nina.az.  
URL: [https://www.wiki.uk-ua.nina.az/Діаграма\\_прецедентів.html](https://www.wiki.uk-ua.nina.az/Діаграма_прецедентів.html) (дата звернення: 26.05.2022).
16. Програми як системи. studopedia.com.ua.  
URL: [https://studopedia.com.ua/1\\_123204\\_programi-yak-sistemi.html](https://studopedia.com.ua/1_123204_programi-yak-sistemi.html) (дата звернення: 01.06.2022).
17. Limit attempts by bestwebsoft – wordpress anti-bot and security plugin for login and forms. WordPress.org Україна.  
URL: <https://uk.wordpress.org/plugins/limit-attempts/> (дата звернення: 04.06.2022).
18. Бондари Б., Гриффитс Э. Основи розробки плагінів WordPress 3 :  
Посібник. 3-тє вид. Packt Publishing, 2011. 300 с.
19. Гайджарі В., Ізварин І. Бази даних в інформаційних системах :  
підручник. Ун-т "Україна", 2018. 418 с.
20. Реляційні СУБД: огляд бази даних, приклади. What.  
URL: <https://what.com.ua/reliaciini-sybd-ogliad-bazi-d/> (дата звернення: 14.06.2022).
21. Сценарій використання – Вікіпедія. Вікіпедія. URL:  
[https://uk.wikipedia.org/wiki/Сценарій\\_використання](https://uk.wikipedia.org/wiki/Сценарій_використання) (дата звернення: 10.06.2022).

**ДОДАТОК А****Index.php**

```
<?php
/**
 * Dashboard Administration Screen
 *
 * @package WordPress
 * @subpackage Administration
 */

/** Load WordPress Bootstrap */
require_once __DIR__ . '/admin.php';

/** Load WordPress dashboard API */
require_once ABSPATH . 'wp-admin/includes/dashboard.php';

wp_dashboard_setup();

wp_enqueue_script( 'dashboard' );

if ( current_user_can( 'install_plugins' ) ) {
    wp_enqueue_script( 'plugin-install' );
    wp_enqueue_script( 'updates' );
}

if ( current_user_can( 'upload_files' ) ) {
    wp_enqueue_script( 'media-upload' );
}

add_thickbox();
```



```

if ( wp_is_mobile() ) {
    wp_enqueue_script( 'jquery-touch-punch' );
}

$title = __( 'Dashboard' );
$parent_file = 'index.php';

$help = '<p>'. __( 'Welcome to your WordPress Dashboard! This is the screen
you will see when you log in to your site, and gives you access to all the site
management features of WordPress. You can get help for any screen by clicking
the Help tab above the screen title.' ) . '</p>';

$screen = get_current_screen();

$screen->add_help_tab(
    array(
        'id' => 'overview',
        'title' => __( 'Overview' ),
        'content' => $help,
    )
);

// Help tabs.

$help = '<p>'. __( 'The left-hand navigation menu provides links to all of the
WordPress administration screens, with submenu items displayed on hover. You
can minimize this menu to a narrow icon strip by clicking on the Collapse Menu
arrow at the bottom.' ) . '</p>';

```

*\$help .= '<p>'. \_\_( 'Links in the Toolbar at the top of the screen connect your dashboard and the front end of your site, and provide access to your profile and helpful WordPress information.' ) . '</p>';*

```
$screen->add_help_tab(
    array(
        'id'    => 'help-navigation',
        'title' => __( 'Navigation' ),
        'content' => $help,
    )
);
```

*\$help = '<p>'. \_\_( 'You can use the following controls to arrange your Dashboard screen to suit your workflow. This is true on most other administration screens as well.' ) . '</p>';*

*\$help .= '<p>'. \_\_( '<strong>Screen Options</strong> &mdash; Use the Screen Options tab to choose which Dashboard boxes to show.' ) . '</p>';*

*\$help .= '<p>'. \_\_( '<strong>Drag and Drop</strong> &mdash; To rearrange the boxes, drag and drop by clicking on the title bar of the selected box and releasing when you see a gray dotted-line rectangle appear in the location you want to place the box.' ) . '</p>';*

*\$help .= '<p>'. \_\_( '<strong>Box Controls</strong> &mdash; Click the title bar of the box to expand or collapse it. Some boxes added by plugins may have configurable content, and will show a [Configure](#) link in the title bar if you hover over it.' ) . '</p>';*

```
$screen->add_help_tab(
    array(
        'id'    => 'help-layout',
        'title' => __( 'Layout' ),
        'content' => $help,
    )
);
```

);

```
$help = '<p>' . __( 'The boxes on your Dashboard screen are:' ) . '</p>';
```

```
if ( current_user_can( 'edit_theme_options' ) ) {
```

```
    $help .= '<p>' . __( '<strong>Welcome</strong> &mdash; Shows links for  
some of the most common tasks when setting up a new site.' ) . '</p>';
```

```
}
```

```
if ( current_user_can( 'view_site_health_checks' ) ) {
```

```
    $help .= '<p>' . __( '<strong>Site Health Status</strong> &mdash; Informs  
you of any potential issues that should be addressed to improve the performance or  
security of your website.' ) . '</p>';
```

```
}
```

```
if ( current_user_can( 'edit_posts' ) ) {
```

```
    $help .= '<p>' . __( '<strong>At a Glance</strong> &mdash; Displays a  
summary of the content on your site and identifies which theme and version of  
WordPress you are using.' ) . '</p>';
```

```
}
```

```
$help .= '<p>' . __( '<strong>Activity</strong> &mdash; Shows the upcoming  
scheduled posts, recently published posts, and the most recent comments on your  
posts and allows you to moderate them.' ) . '</p>';
```

```
if ( is_blog_admin() && current_user_can( 'edit_posts' ) ) {
```

```
    $help .= '<p>' . __( "<strong>Quick Draft</strong> &mdash; Allows you to  
create a new post and save it as a draft. Also displays links to the 3 most recent  
draft posts you've started." ) . '</p>';
```

```
}
```

```

$help .= '<p>'. sprintf(
    /* translators: %s: WordPress Planet URL. */
    __( '<strong>WordPress Events and News</strong> &mdash; Upcoming events
near you as well as the latest news from the official WordPress project and the <a
href="<%=s">WordPress Planet</a>.' ),
    __( 'https://planet.wordpress.org/' )
). '</p>';

$screen->add_help_tab(
    array(
        'id'    => 'help-content',
        'title' => __( 'Content' ),
        'content' => $help,
    )
);

unset( $help );

$screen->set_help_sidebar(
    '<p><strong>'. __( 'For more information:' ) . '</strong></p>'.
    '<p>'. __( '<a href="https://wordpress.org/support/article/dashboard-
screen/">Documentation on Dashboard</a>' ) . '</p>'.
    '<p>'. __( '<a href="https://wordpress.org/support/">Support</a>' ) . '</p>'
);

require_once ABSPATH . 'wp-admin/admin-header.php';
?>

<div class="wrap">

```

```

<h1><?php echo esc_html( $title ); ?></h1>

<?php
if ( ! empty( $_GET['admin_email_remind_later'] ) ) :
    /** This filter is documented in wp-login.php */

    $remind_interval = (int) apply_filters( 'admin_email_remind_interval', 3 *
DAY_IN_SECONDS );

    $postponed_time = get_option( 'admin_email_lifespan' );

    /*
    * Calculate how many seconds it's been since the reminder was postponed.
    * This allows us to not show it if the query arg is set, but visited due to caches,
bookmarks or similar.
    */

    $time_passed = time() - ( $postponed_time - $remind_interval );

    // Only show the dashboard notice if it's been less than a minute since the
message was postponed.

    if ( $time_passed < MINUTE_IN_SECONDS ) :
        ?>
        <div class="notice notice-success is-dismissible">
            <p>
                <?php
                printf(
                    /* translators: %s: Human-readable time interval. */
                    __( 'The admin email verification page will reappear after %s.' ),
                    human_time_diff( time() + $remind_interval )
                );
            ?>

```

```

    </p>
</div>
<?php endif; ?>
<?php endif; ?>

<?php
if ( has_action( 'welcome_panel' ) && current_user_can( 'edit_theme_options' ) ) :
    $classes = 'welcome-panel';

    $option = (int) get_user_meta( get_current_user_id(), 'show_welcome_panel',
true );

    // 0 = hide, 1 = toggled to show or single site creator, 2 = multisite site owner.
    $hide = ( 0 === $option || ( 2 === $option && wp_get_current_user()-
>user_email !== get_option( 'admin_email' ) ) );

    if ( $hide ) {
        $classes .= ' hidden';
    }
?>

<div id="welcome-panel" class="<?php echo esc_attr( $classes ); ?>">
    <?php wp_nonce_field( 'welcome-panel-nonce', 'welcomepanelnonce', false );
?>

    <a class="welcome-panel-close" href="<?php echo esc_url( admin_url(
'welcome=0' ) ); ?>" aria-label="<?php esc_attr_e( 'Dismiss the welcome panel'
); ?>"><?php _e( 'Dismiss' ); ?></a>

    <?php
/**
 * Add content to the welcome panel on the admin dashboard.
 *
 * To remove the default welcome panel, use remove_action():

```



**ДОДАТОК Б**  
**wp-comments-post.php**

```
<?php
/**
 * Handles Comment Post to WordPress and prevents duplicate comment posting.
 *
 * @package WordPress
 */

if ( 'POST' !== $_SERVER['REQUEST_METHOD'] ) {
    $protocol = $_SERVER['SERVER_PROTOCOL'];
    if ( ! in_array( $protocol, array( 'HTTP/1.1', 'HTTP/2', 'HTTP/2.0' ), true ) )
    {
        $protocol = 'HTTP/1.0';
    }

    header( 'Allow: POST' );
    header( "$protocol 405 Method Not Allowed" );
    header( 'Content-Type: text/plain' );
    exit;
}

/** Sets up the WordPress Environment. */
require __DIR__ . '/wp-load.php';

nocache_headers();

$comment = wp_handle_comment_submission( wp_unslash( $_POST ) );
if ( is_wp_error( $comment ) ) {
```



```

$data = (int) $comment->get_error_data();

if ( ! empty( $data ) ) {
    wp_die(
        '<p>' . $comment->get_error_message() . '</p>',
        __( 'Comment Submission Failure' ),
        array(
            'response' => $data,
            'back_link' => true,
        )
    );
} else {
    exit;
}
}

$user = wp_get_current_user();

$cookies_consent = ( isset( $_POST['wp-comment-cookies-consent'] ) );

/**
 * Perform other actions when comment cookies are set.
 *
 * @since 3.4.0
 * @since 4.9.6 The `$cookies_consent` parameter was added.
 *
 * @param WP_Comment $comment Comment object.
 * @param WP_User $user Comment author's user object. The user may
not exist.
 * @param bool $cookies_consent Comment author's consent to store cookies.
 */
do_action( 'set_comment_cookies', $comment, $user, $cookies_consent );

```

```

$location = empty( $_POST['redirect_to'] ) ? get_comment_link( $comment ) :
$_POST['redirect_to'] . '#comment-' . $comment->comment_ID;

// If user didn't consent to cookies, add specific query arguments to display the
awaiting moderation message.

if ( ! $cookies_consent && 'unapproved' === wp_get_comment_status( $comment
) && ! empty( $comment->comment_author_email ) ) {

    $location = add_query_arg(

        array(

            'unapproved' => $comment->comment_ID,

            'moderation-hash' => wp_hash( $comment-
>comment_date_gmt ),

        ),

        $location

    );

}

/**
 * Filters the location URI to send the commenter after posting.
 *
 * @since 2.0.5
 *
 * @param string $location The 'redirect_to' URI sent via $_POST.
 * @param WP_Comment $comment Comment object.
 */

$location = apply_filters( 'comment_post_redirect', $location, $comment );

wp_safe_redirect( $location );

exit;

```