

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ НАЦІОНАЛЬНИЙ АГРАРНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕКОНОМІКИ І МЕНЕДЖМЕНТУ**

Кафедра кібернетики та інформатики

КВАЛІФІКАЦІЙНА РОБОТА

освітній ступінь - «Бакалавр»

**на тему: Розробка вебзастосунку для ведення замовлень
на виготовлення сільсько-господарського обладнання**

Виконала:

студентка спеціальності

126 «Інформаційні системи та технології»

Блаженко Дарія Олександрівна

Керівник:

В'юненко Олександр Борисович

к.е.н., доцент кафедри кібернетики та інформатики

Рецензент:

Ободяк Віктор Корнелійович

к.т.н., доцент комп'ютерних наук СумДУ

СУМСЬКИЙ НАЦІОНАЛЬНИЙ АГРАРНИЙ УНІВЕРСИТЕТ

Факультет	Економіки і менеджменту
Кафедра	Кібернетики та інформатики
Освітній ступінь	«Бакалавр»
Спеціальність	126 «Інформаційні системи та технології» ОП «Інформаційні системи та технології»

Затверджую:
Завідувач кафедри Світлана АГАДЖАНОВА

«_____» «_____» 2024_____ р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ *Блаженко Дарії Олександрівні*

1. Тема роботи: Розробка вебзастосунку для ведення замовлень на виготовлення сільсько-господарського обладнання.

2. База практичного дослідження: ТОВ «ДЗСТ 2».

керівник роботи В'юненко Олександр Борисович к.е.н., доцент

затверджена наказом по університету від «_____» «_____»

3. Строк подання студентом закінченого проекту
(роботи)

«28» червня 2024 року

4. Вихідні дані до проекту (роботи):

Результати аналітичних досліджень зарубіжних та вітчизняних вчених, інтернет-джерела, нормативні документи за чинним законодавством.

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

- Виконати постановку задачі;
- Вибрати середовище для реалізації;
- розробити технічне завдання;
- розробити програмний продукт відповідно до поставлених задач;
- розробити інструкцію для користувача;

6. Дата видачі завдання:

«_____» «_____» 2024р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів проекту	Примітка
1	Визначення об'єкту, предмету дослідження, формулювання мети та задач кваліфікаційної роботи, складання плану	січень 2024 р.	
2	Підбір та вивчення літературних джерел, технічної документації	січень 2024 р.	
3	Розробка технічного завдання	січень 2024 р..	
4	Розробка вебзастосунку	січень-квітень 2024 р.	
5	Створення інструкції користування вебзастосунком для користувачів	до 14 травня 2024 р.	
6	Оформлення пояснювальної записки	до 30 травня 2024 р..	
7	Оформлення презентації	до 1 червня 2024 р.	
8	Опрацювання зауважень керівника, перевірка на автентичність	5 червня 2024 р.	
9	Підготовка доповіді та її попередній захист	10 червня 2024 р.	
10	Перевірка з академічної доброчесності кваліфікаційної роботи	10 червня 2024 р.	
11	Представлення кваліфікаційної роботи на кафедрі, деканат	14 червня 2024 р.	
12	Захист кваліфікаційної роботи	28 червня 2024р.	

Студент _____ **Дарія БЛАЖЕНКО**
(підпис)

Керівник роботи _____ **Олександр В'ЮНЕНКО**
(підпис)

Перевірку на автентичність проведено _____ **Надія БАРАНИК**
(підпис)

Кваліфікаційна робота допущена до захисту _____ **Маргарита ЛИШЕНКО**
(підпис)

АНОТАЦІЯ

Блаженко Д.О. Розробка вебзастосунку для ведення замовлень на виготовлення сільсько-господарського обладнання.

Кваліфікаційна робота зі спеціальності 126 «Інформаційні системи та технології» ОП Інформаційні системи та технології Сумського національного аграрного університету. Суми, 2024 р. – Рукопис.

Дана кваліфікаційна робота присвячена розробці вебзастосунку для ведення замовлень на виготовлення сільсько-господарського обладнання. У роботі аналізуються існуючі інформаційні системи для ведення замовлень. Розглядаються поняття інтернет-торгівлі, структура вебзастосунку, засоби та методи, що використовуються для його розробки.

Об'єктом дослідження є ТОВ «ДЗСТ 2» (Дубовицький завод сільгосптехніки 2), а предметом є вебзастосунок для ведення замовлень на виготовлення сільсько-господарського обладнання.

Метою роботи є розробка вебзастосунку для ведення замовлень, який переглядати товари на сайті, додавати їх у кошик, здійснювати замовлення, створити обліковий запис та мати доступ до інформації та контактних даних.

Результатом виконання кваліфікаційної роботи за допомогою сучасних вебтехнологій є вебзастосунок для ведення замовлень, у якому користувач може вибрати потрібний вид товару і залишити заявку на його купівлю, заповнивши необхідні дані. Основа сайту дозволяє використовувати її для реалізації вебзастосунку на будь-яку тематику, оскільки всі найважливіші компоненти були створені: кошик товарів, панель адміністрування, форми реєстрації та авторизації, можливості сортування товарів за різними критеріями.

Ключові слова: інтернет-торгівля, вебзастосунок, інформаційна система, інтерфейс користувача, обліковий запис, замовлення, база даних, HTML, CSS, JAVASCRIPT, PHP.

SUMMARY

Blazhenko D.O. Development of a web application for managing orders for the production of agricultural equipment.

Qualification work on specialty 126 "Information systems and technologies" OP Information systems and technologies of the Sumy National Agrarian University. Sumy, 2024 - Manuscript.

This qualification work is devoted to the development of a web application for managing orders for the manufacture of agricultural equipment. The work analyzes the existing information systems for keeping orders. The concepts of Internet trade, the structure of a web application, tools and methods used for its development are considered.

The object of the research is LLC "DZST 2" (Dubovytskyi Plant of Agricultural Machinery 2), and the subject is a web application for managing orders for the manufacture of agricultural equipment.

The purpose of the work is to develop a web application for managing orders, which will allow you to view products on the site, add them to the cart, make an order, create an account and have access to information and contact data.

The result of performing qualification work with the help of modern web technologies is a web application for keeping orders, in which the user can choose the desired type of product and leave an application for its purchase, filling in the necessary data. The basis of the site allows you to use it to implement a web application on any topic, since all the most important components have been created: a shopping cart, an administration panel, registration and authorization forms, the ability to sort products by various criteria.

Keywords: e-commerce, web application, information system, user interface, account, order, database, HTML, CSS, JAVASCRIPT, PHP

ЗМІСТ

ВСТУП		7
РОЗДІЛ 1	АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	9
1.1	Аналіз діяльності підприємства	9
1.2	Аналіз реалізації товарів засобами Інтернету	10
1.3	Аналіз програмних продуктів та аналогів	14
РОЗДІЛ 2	ПОСТАНОВКА ЗАДАЧІ ТА ВИБІР ЗАСОБІВ РЕАЛІЗАЦІЇ	20
2.1	Постановка задачі	20
2.2	Вибір та обґрунтування засобів реалізації	23
РОЗДІЛ 3	ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ВЕБЗАСТОСУНКУ ДЛЯ ВЕДЕННЯ ЗАМОВЛЕНЬ НА ВИГОТОВЛЕННЯ СІЛЬСЬКОГОСПОДАРСЬКОГО ОБЛАДНАННЯ	28
3.1	Розробка функціональних діаграм програмного засобу	28
3.2	Проектування бази даних вебзастосунку для реєстрації даних моніторингу навколишнього середовища	31
3.3	Розробка макетів вебзастосунку	36
3.4	Фізична реалізація бази даних та вебзастосунку	41
ВИСНОВКИ		53
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ		55
ДОДАТКИ		57
Додаток А	Технічне завдання	
Додаток Б	Планування робіт	
Додаток В	Копії публікацій	
Додаток Г	Лістинги програм	

ВСТУП

Актуальність теми. Сьогодні глобальна мережа Інтернет виступає як бізнес-інфраструктура, що являє собою багатомільйонний ринок потенційних покупців і продавців. У зв'язку з цим підприємства різної бізнес-спрямованості створюють вебзастосунки для участі в цьому ринку з метою власної комерційної вигоди.

Вебзастосунок визначатимемо як сукупність вебсторінок, доступних за доменним ім'ям, що підтримуються для повноцінного представлення інформації, необхідних електронних документів, ведення інтерактивної взаємодії зі сторонніми користувачами як приватними особами, так і організаціями з метою забезпечення представництва власних інтересів у мережі Інтернет.

Вебзастосунок для ведення замовлень ТОВ «ДЗСТ 2» повинно являти собою інформаційний ресурс із функціоналом Інтернет-магазину, на якому клієнти зможуть швидко ознайомитися з переліком сільськогосподарського обладнання, його характеристиками, вартістю, отримати можливість правильного підбору товару, отримати відомості щодо оформлення замовлення, гарантій, варіантів доставки, повернення й обміну товарів, здійснити оформлення замовлення та його оплату, написати запитання.

На цей час існує конкуренція між численними компаніями в галузі виготовлення обладнання. Найкращим способом заявити про себе потенційним клієнтам - це розповісти про компанію, про продукцію та її вартість. Для цього компанії створюють власні торгові майданчики в інтернеті. ТОВ «ДЗСТ 2» не є виключенням, тому розробка вебзастосунку для ведення замовлень підвищить конкурентоспроможність, збільшить цільову аудиторію та впізнаваність компанії.

Мета і завдання дослідження. Метою роботи є розробка вебзастосунку для ведення замовлень підприємству ТОВ «ДЗСТ 2».

Для досягнення поставленої мети випускної кваліфікаційної роботи необхідно вирішити наступні завдання:

- Проаналізувати діяльність підприємства ТОВ «ДЗСТ 2»;

- Здійснити аналіз реалізації товарів засобами Інтернету;
- Здійснити огляд чинних вебзастосунків для продажу сільськогосподарського обладнання;
- Розробити постановку задачі;
- Обґрунтувати вибір засобів реалізації;
- Виконати проектування інтерфейсу вебзастосунку;
- Виконати проектування структури бази даних вебзастосунку;
- Виконати проектування модульної структури вебзастосунку;
- Виконати програмну реалізацію вебзастосунку.

Об'єктом дослідження є підприємство ТОВ «ДЗСТ 2».

Предметом дослідження є автоматизація ведення замовлень на виготовлення сільськогосподарського обладнання для об'єкта дослідження.

Апробація результатів кваліфікаційної роботи. Результати кваліфікаційної роботи апробовані на конференціях: 1) всеукраїнської наукової конференції студентів і аспірантів, присвяченої міжнародному дню студента (13-17 листопада 2023 р.), Суми; 2) науково-практичної конференції викладачів, аспірантів та студентів Сумського НАУ (14-16 травня 2024 р.).

Публікації. За матеріалами кваліфікаційної роботи опубліковано 2 тези наукових конференцій.

Структура і обсяг роботи. Кваліфікаційна робота складається із вступу, трьох розділів, висновків та пропозицій, списку використаних джерел з 18 найменування, додатків. Основний текст викладений на 50 сторінках комп'ютерного тексту, робота містить 7 таблиць, 40 рисунків, 4 додатків.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз діяльності підприємства

ТОВ "ДЗСТ 2" (Дубовицький завод сільгосптехніки 2) є підприємством у галузі виробництва сільськогосподарської техніки та металевих виробів у Сумській області. Зареєстроване 18 серпня 2022 року, воно розташоване в селі Дубовичі, Конотопського району.

Основна діяльність підприємства включає:

- допоміжна діяльність у рослинництві;
- виробництво інших готових металевих виробів;
- ремонт і технічне обслуговування готових металевих виробів;
- ремонт і технічне обслуговування машин і устаткування промислового призначення;
- технічне обслуговування та ремонт автотранспортних засобів;
- роздрібна торгівля деталями та приладдям для автотранспортних засобів;
- гуртова торгівля сільськогосподарськими машинами й устаткуванням;
- неспеціалізована гуртова торгівля;
- інша допоміжна діяльність у сфері транспорту [4].

Протягом 2023 року підприємство продемонструвало фінансовий результат виторгу в розмірі 12,723,500 гривень і чистого прибутку в 74,000 гривень. "ДЗСТ 2" активно розвивається, співпрацюючи з місцевими організаціями та іншими суб'єктами господарювання [5].

Під керівництвом Онікієнка Віктора Борисовича, який є директором підприємства, "ДЗСТ 2" прагне до подальшого розширення своєї діяльності та вдосконалення виробничих процесів для потреб сільськогосподарського сектору України. ТОВ "ДЗСТ 2" відіграє важливу роль у розвитку місцевої економіки,

надаючи сучасні технологічні рішення для сільського господарства та сприяючи створенню нових робочих місць у регіоні.

1.2 Аналіз реалізації товарів засобами Інтернету

На сьогодні інтернет-торгівля є сферою підприємницької діяльності, що активно розвивається. Цей вид бізнесу формує новий віртуальний канал збуту, який істотно скорочує торговельні витрати. Ще кілька років тому досвід покупок через інтернет мали лише деякі, а кількість компаній, що пропонують свої товари і послуги в інтернеті, була вкрай незначною. Наразі кількість on-line операцій зростає бурхливими темпами, а більшість компаній активно починають освоювати і використовувати у своїх цілях переваги інтернет-торгівлі.

Вебзастосунок для ведення замовлень - це сайт для ведення інтернет-торгівлі, призначений для:

- надання покупцям за допомогою мережі інтернет відомостей, необхідних для здійснення покупок, у тому числі про асортимент товарів, ціни, інформацію про товари, способи та умови доставлення й оплати;
- отримання від покупців за допомогою інтернету повідомлень з наміром купівлі продукції, що продається;
- надання послуг доставлення товарів продавцем за вказаною адресою покупця [11].

Оскільки сайт забезпечує користувача інформацією у вигляді каталогу товарів, що міститься в базі даних, то сайт представляється у вигляді інформаційної системи. Інформаційні системи поділяються на різні види, відповідно до наданої інформації, але кожна інформаційна система має характерні особливості:

- виконання однієї, декількох функцій щодо інформації;
- єдність системи (наявність спільної файлової бази, єдиних стандартів і протоколів, єдиного управління);
- можливість композиції та декомпозиції об'єктів системи при виконанні заданих функцій.

Завданням інформаційної системи є забезпечення користувача інформацією, якої той потребує, у межах певної предметної області, в повному обсязі та в найкоротші терміни [1].

Рішення вибору сайту як інформаційної системи є раціональним, оскільки доступ до інформації надано через комп'ютер, що використовує будь-яку операційну систему, а також за допомогою інших комп'ютерних пристроїв, здатних здійснювати відображення сторінок сайту, які мають підключення до інтернету.

Для побудови сайту необхідно використовувати вебсервер. Це сервер, що приймає HTTP-запити від клієнтів, зазвичай веббраузерів, і видає їм HTTP-відповіді разом із HTML-сторінкою, зображенням, файлом, медіапотоком або іншими даними.

Вебсервером називають програмне забезпечення, що виконує функції сервера, так і комп'ютер, на якому це програмне забезпечення працює.

Клієнт, яким є веббраузер, передає вебсерверу запити на отримання ресурсів, позначених URL-адресами. Ресурси - це HTML-сторінки, зображення, файли, медіапотоки або інші дані, які необхідні клієнту. У відповідь вебсервер передає клієнту запитані дані. Цей обмін відбувається за протоколом HTTP [8].

Вебсервери працюють на різних операційних системах, таких як Windows, Linux, macOS та інші. Вибір операційної системи залежить від вимог проєкту та специфічних умов експлуатації.

Важливою частиною роботи вебсервера є налаштування безпеки. Сюди входить налаштування SSL/TLS для забезпечення безпечного обміну даними між клієнтом і сервером та налаштування правил доступу.

Крім того, вебсервери інтегруються з базами даних (наприклад, MySQL, PostgreSQL) для зберігання та управління динамічним контентом, а також з різними мовами програмування та фреймворками (наприклад, PHP, Python, Ruby on Rails) для створення складних вебдодатків.

Робота вебсервера та клієнта зображена на рисунку 1.1.

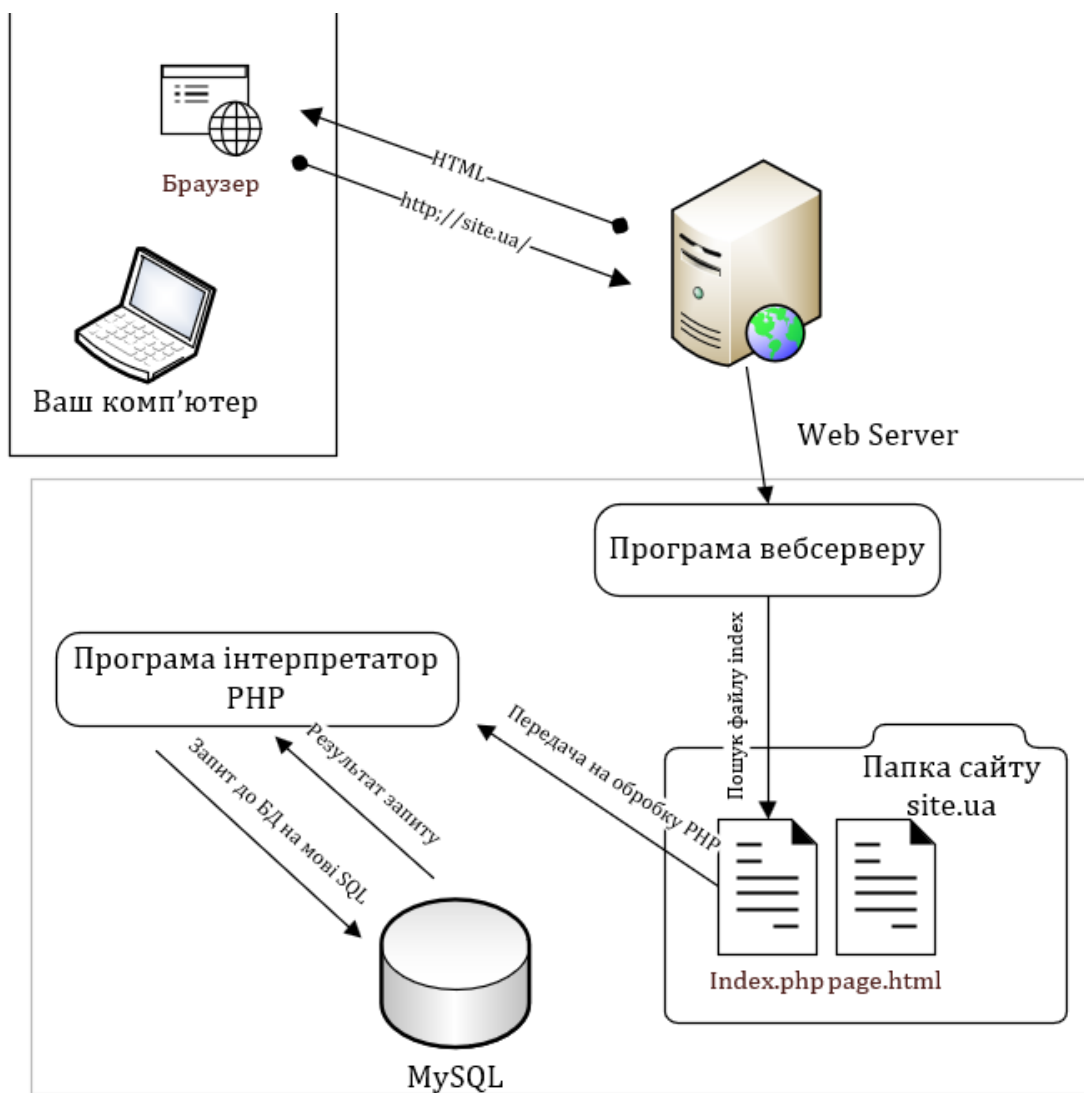


Рисунок 1.1 – Робота вебсервера та клієнта

Джерело: запропоновано автором

Вся інформація, яка надається користувачеві у вигляді відповіді вебсервера, зберігається в базі даних. База даних - це сукупність структурованих даних, які зберігаються в пам'яті обчислювальної системи і відображають стан об'єкта в даній області і його взаємозв'язок.

Керування базою даних здійснюється за допомогою системи керування (управління) базами даних. Система керування базами даних (СКБД) - це набір мовних і програмних засобів, призначених для створення, обслуговування і спільного використання баз даних багатьма користувачами. СКБД відрізняються за моделлю даних, яка використовується. Тому СКБД, заснована на використанні реляційної моделі даних, називається реляційною базою даних.

Характерними ознаками системи управління базами даних є:

- сепаратність доступу;
- цілісність даних;
- секретність;
- синхронізація;
- захист від відмов.

Оскільки інформація, наявна в базі даних, має бути систематизованою і з легкістю піддаватися обробці з боку користувача і системи, для кожного сегмента інформації створюється своя таблиця (сутність). Кожна із сутностей містить у собі поля (атрибути), що дають повною мірою охарактеризувати інформацію, що зберігається в цій таблиці [1].

Для того щоб користувач сайту міг отримати і проаналізувати інформацію з бази даних, потрібно реалізувати виведення інформації з бази даних на сайт. Для виконання цього завдання потрібно застосувати скриптову мову, яка взаємодіє з базою даних. З метою обробки інформації, що надходить, інтерпретатор такої мови встановлюється на вебсервері. Файли(скрипти), складені такою мовою, будуть інтерпретовані і запущені тільки, коли опиняться на сервері. У зв'язку з цим, під час складання таких скриптових файлів, потрібно заздалегідь розрахувати їхній функціонал.

Результати, отримані в ході інтерпретації скрипта, генерують HTML-код, на основі якого браузер генерує результуючу відповідь у вікні браузера користувача.

Щоб реалізувати взаємодію користувача з сайтом, потрібен інтерфейс, який забезпечуватиме всі необхідні елементи управління. Для створення такого інтерфейсу, потрібно використовувати за основу, мову гіпертекстової розмітки - HTML. За допомогою цієї мови створюються сторінки сайту і контент [9].

Оскільки основним завданням вебзастосунку для ведення замовлень на виготовлення сільськогосподарського обладнання є продаж асортименту продукції, то користувач повинен мати доступ замовляти ці товари.

Після оформлення замовлення та підготовки до доставлення покупець отримає сповіщення про те, що його замовлення готове.

1.3 Аналіз програмних продуктів та аналогів

Перед створенням або відкриттям будь-якої діяльності в інтернеті, яка пов'язана з комерцією, першим кроком є огляд та аналіз аналогів для виявлення переваг та недоліків в таких компонентах:

- структура сайту;
- дизайн сторінок;
- користувацький функціонал.

Для цього обрано сайти компаній, які виготовляють та продають сільськогосподарське обладнання:

- «ОЛЕАНДР АГРО» [3];
- «agroZIP» [2].

Першим прикладом є вебзастосунок «ОЛЕАНДР АГРО». На головній сторінці сайту розташовано навігаційне меню з пошуком за каталогом, панель із категоріями товарів, інформаційна панель, список усіх брендів, особистий кабінет, закладки, кошик, оформлення замовлення, контактна інформація.

Головна сторінка вебзастосунку «OLEANDR AGRO» надає користувачам зручний доступ до функцій сайту та інформації.

Сторінка сайту «ОЛЕАНДР АГРО» зображена на рисунку 1.2.

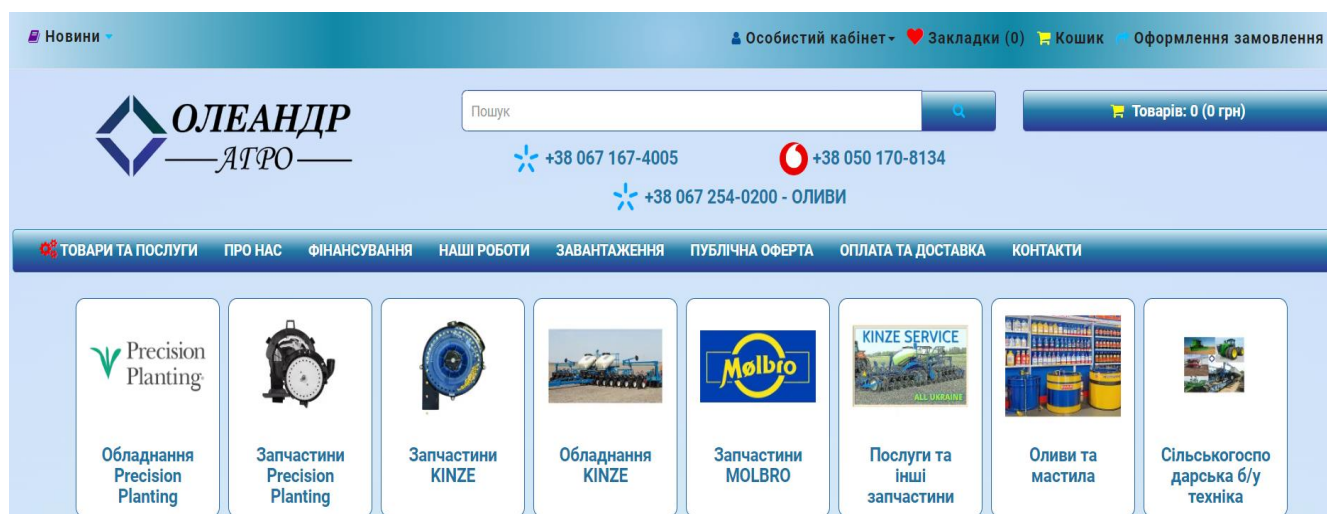


Рисунок 1.2 – Головна сторінка сайту «ОЛЕАНДР АГРО».

Джерело: [3]

Сторінка товару містить назву товару, код, невеликий опис, характеристики, ціну, різні зображення товару, посилання на додавання в кошик, відомості про наявність товару на складі, розділи для питань та відгуків, посилання на інші сторінки сайту (рисунок 1.3).



Рисунок 1.3 - Сторінка товару

Джерело: [3]

Далі проведено огляд особистого кабінету та оформлення замовлення сайту. Після натискання на «Особистий кабінет» користувач проходить реєстрацію або авторизується. Форма для реєстрації містить такі графи: «ПІБ», «Телефон», «Email», «Пароль», «Повторити пароль». Сторінка реєстрації зображена на рисунку 1.4.

ОСНОВНА ІНФОРМАЦІЯ

Підписатись на новини ☒ Так ☐ Ні

* ПІБ

* Телефон

* Email

* Пароль

* Підтвердіть пароль

☐ Показати пароль

Юридичний статус ☐ Юридична особа ☐ Приватна особа

Рисунок 1.4 - Сторінка реєстрації користувача.

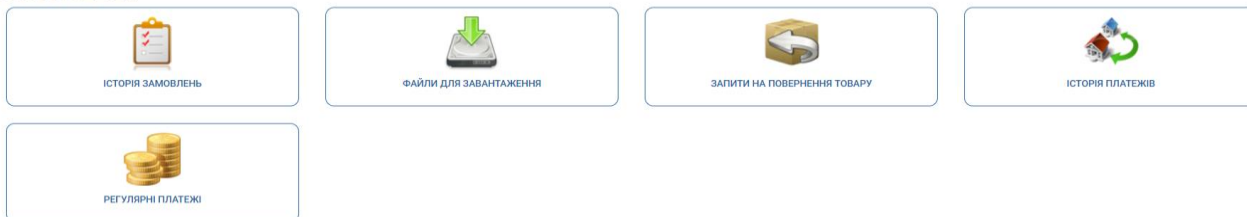
Джерело: [3]

Обліковий запис після авторизації містить контактні дані, пароль, адреси та реквізити, закладки, історія замовлень, запити на повернення, історії платежів, регулярні платежі. Вигляд сторінки «Обліковий запис» зображено на рисунку 1.5.

ОБЛІКОВИЙ ЗАПИС



ЗАМОВЛЕННЯ



ПІДПИСКА



Рисунок 1.5 - Обліковий запис користувача

Джерело: [3]

Сторінка оформлення замовлення містить список обраних товарів, форму для вибору способу доставлення, форму для введення адреси доставлення та інформацію про клієнта, яка вноситься автоматично при наявності облікового запису (рисунок 1.6).

ФОТО	НАЙМЕНУВАННЯ ТОВАРУ	МОДЕЛЬ	КІЛЬК.	ЦІНА З ПДВ	РАЗОМ З ПДВ
	Адаптер долота M7 - MOL4150	MOL4150	1	1 155 грн	1 155 грн
Разом:					963 грн
ПДВ 20%:					193 грн
Всього до сплати з ПДВ:					1 155 грн

Покупець	
* Телефон	
Телефон має бути від 3 до 32 символів!	
* ПІБ	
Прізвище, Ім'я, по Батькові	
Ім'я має бути від 3 до 200 символів!	
Юридичний статус	☐ Юридична особа ☒ Приватна особа

Спосіб доставки	
☒	Нова пошта
☒	Доставка у відділення Нової Пошти - 127 грн Термін доставки - 2 дні
☐	Доставка кур'єром Нової Пошти - 169 грн Термін доставки - 2 дні
☐	Доставка в поштомат Нової Пошти - 127 грн Термін доставки - 2 дні
Самовивіз	
☐	Самовивіз зі складу компанії - 0 грн

Адреса доставки	
* Область	
* Місто	
* Відділення/Поштомат Нової Пошти	

Рисунок 1.6 - Сторінка оформлення замовлення

Джерело: [3]

Наступним аналогом розглянуто сайт «agroZIP». Головна сторінка містить каталог товарів, рядок пошуку, контактний номер та кнопку «Зателефонувати Нам», посилання на інші сторінки, кошик (рис. 1.7).

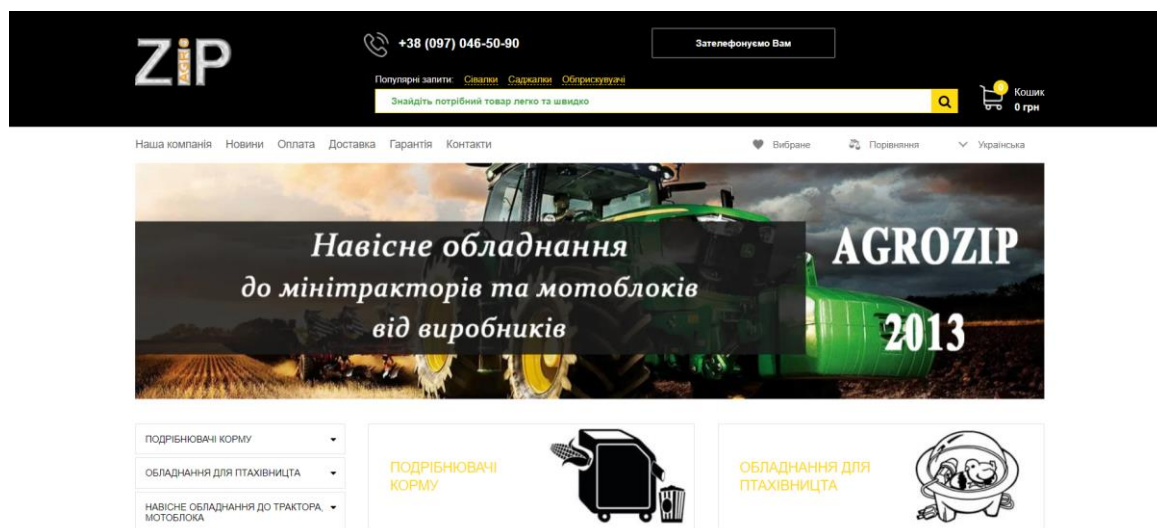


Рисунок 1.7 - Головна сторінка сайту «agroZIP»

Джерело: [2]

Сторінка товару має схожу побудову з попереднім представником, однак є відмінності, такі як: наявність детальнішої характеристики, сторінку для порівняння товарів, функцією поділитися посиланням та кнопки «Купити в 1 Клік».

Сторінка товару наведена на рисунку 1.8.

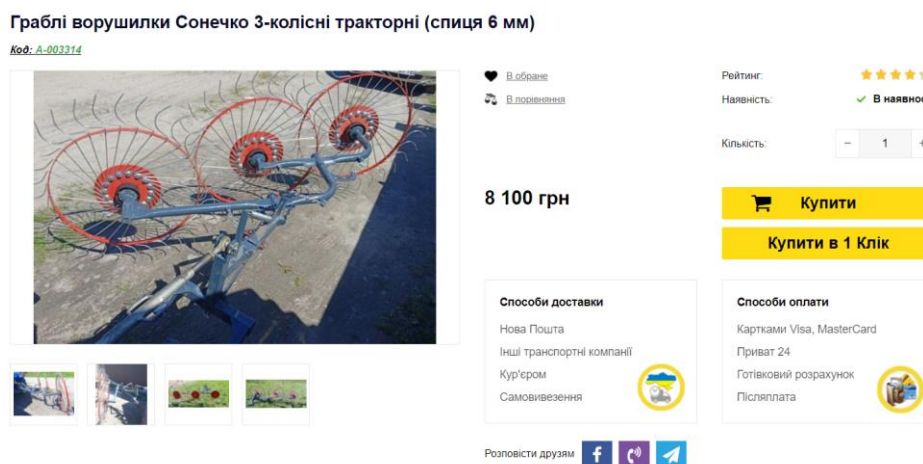


Рисунок 1.8 - Сторінка товару сайту «agroZIP»

Джерело: [2]

Кошик для оформлення замовлень зображений на рисунку 1.9.

Кошик

Фото	Найменування	Ціна	Кількість	Сума
	Комплект для посадки картоплі КС7 до важкого мотоблока	2 995 грн	- 1 +	2 995 грн ×
				Разом: 2 995 грн

Способи доставки

- ☒  Нова Пошта (доставка у відділення) (Сплачується окремо згідно з тарифами логістичної компанії)
- ☐  Нова Пошта (адресна доставка "До дверей") (Сплачується окремо згідно з тарифами логістичної компанії)
- ☐  Meest Express (доставка у відділення) (Сплачується окремо згідно з тарифами логістичної компанії)
- ☐ 

Рисунок 1.9 - Кошик замовлень сайту «agroZIP»

Джерело: [2]

Кошик містить список обраних товарів, загальну суму замовлення, способи доставлення, способи оплати та вікно «Контактні дані», куди замовник вносить свої дані.

Кнопка «Купити в 1 Клік» вимагає для оформлення замовлення ім'я та номер телефону замовника. Вікно для оформлення швидкого замовлення зображено на рисунку 1.10.

ШВИДКЕ ЗАМОВЛЕННЯ

Граблі ворушилки Сонечко 3-колісні тракторні (спиця 6 мм)

Номер телефону*

Замовити

Рисунок 1.10 – Швидке замовлення на сайті «agroZIP»

Джерело: [2]

На сайті «agroZIP» відсутня авторизація користувачів, що впливає на інтерфейс та функціональність. Без облікового запису неможливо зберегти дані та історію замовлень, що обмежує персоналізацію вмісту та зменшує зручність використання сайту. Також це ускладнює процес відстежування індивідуальних запитів та проблем, які виникають у користувачів.

Отже, виділимо основні складові вебзастосунку:

- каталог товарів - являє собою складну і багаторівневу структуру даних, яка повинна виводити впорядковані товари. Каталог представлений у вигляді дерева об'єктів, верхній рівень якого складається зі списку розділів;

- пошукова система є обов'язковим елементом інтернет-магазину. Попри те, що каталог забезпечує впорядкування та групування даних, пошукова система дає користувачеві швидкий доступ до інформації, коли каталог являє собою розгалужену структуру даних з великою кількістю розділів, підрозділів і товарів;

- користувацький кошик – це місце для зберігання обраного користувачем товару. У кошику доступні такі варіанти дій: кількість обраного товару, видалити з кошика, а також оформити замовлення;

- форма оформлення замовлення - призначена для введення контактної інформації замовника і відправки її та замовлення на електронну скриньку організації;

- форма входу - необхідна для входу в особистий кабінет, де користувач дізнається про статус свого замовлення, змінює свої особисті дані тощо;

- форма реєстрації потрібна для аутентифікації користувача.

РОЗДІЛ 2

ПОСТАНОВКА ЗАДАЧІ ТА ВИБІР ЗАСОБІВ РЕАЛІЗАЦІЇ

2.1 Постановка задачі

Однією з цілей вебзастосунку для ведення замовлень є збільшення продажів та конкурентоспроможності, що досягається завдяки поліпшенню якості обслуговування клієнтів, зручності користування платформою.

Управління вебзастосунком передбачає вирішення двох основних завдань:

- створення і розміщення інформації про продукцію або послуги;
- організація процесів отримання та обробки даних за замовленнями клієнтів.

Такі завдання вирішуються за допомогою систем, що керують вебзастосунком. Для їх вирішення необхідно організувати ефективний обмін інформацією про товари і послуги, про замовлення і покупців.

Одним із головних завдань є задоволення потреб клієнтів. Для цього необхідно надавати клієнтам достатню кількість інформації про виріб, а також забезпечувати швидку і зручну доставку.

Розробка та впровадження вебзастосунку сприяє розвитку бренду підприємства і збільшенню впізнаваності серед потенційних клієнтів. Це досягається шляхом розробки унікального дизайну інтерфейсу, оригінального контенту або ексклюзивної продукції.

Вебзастосунок для продажу сільськогосподарського обладнання розроблюється для покращення якості обслуговування клієнтів, що містить швидке опрацювання замовлень, а також для розв'язання проблем, які можуть виникнути.

Щоб вебзастосунок був доступним і зрозумілим для покупця він повинен мати чітку структуру (рисунку 2.1).

Кожен вебзастосунок для ведення замовлень повинен містити всі перераховані елементи на рисунку. За потреби додаються інші додаткові модулі.



Рисунок 2.1 - Структура вебзастосунку для ведення замовлень

Джерело: запропоновано автором

При розробці необхідно враховувати різні вимоги, що надходять від двох ключових груп користувачів: покупців і адміністраторів вебзастосунку. Це забезпечить успіх проєкту, зробивши його зручним і ефективним для обох категорій.

Вимогами покупця, який відвідав сайт вебзастосунку, є отримати необхідну інформацію або знайти товар, який відповідає його потребам і очікуванням.

Список вимог користувача наведено в таблиці 2.1.

Таблиця 2.1 – Вимоги покупця до вебзастосунку.

Назва вимоги	Опис вимоги
Реєстрація в системі	Реєстрація користувача. Для реєстрації на сайті вебзастосунку необхідно ввести логін, адресу електронної поштової скриньки, пароль.
Вхід у систему	Вхід в систему за допомогою введення адреси електронної поштової скриньки та пароля.
Особистий кабінет	В особистому кабінеті є перегляд історії покупок, зміна особистих налаштувань.
Кошик товарів	Видимість усіх товарів, які хоче замовити покупець, з відображенням способу доставлення та підсумковою вартістю.
Перегляд каталогу товарів	Швидке і зручне ознайомлення з товарами, представленими на сайті.
Пошук	Здійснювати пошук за категоріями.

Джерело: запропоновано автором

Адміністратор - користувач, у якого є максимальні права в системі. Ця людина слідкує за наповненням сайту вебзастосунку товарами, додаванням атрибутів до товару, сортуванням товарів за категоріями/каталогами.

Вимоги до адміністратора наведено в таблиці 2.2.

Таблиця 2.2 – Вимоги адміністратора до вебзастосунку

Назва вимоги	Опис вимоги
Управління каталогом товарів	Каталог товарів - розділ сайту для перегляду всіх товарів, доступних для покупки.
Додавання / зміна видалення товарів	Під час додавання/зміни товарів повинна вказуватись така інформація: назва товару, опис товару, вартість товару, статус товару, матеріал.
Додавання зображень	Завантаження зображень товарів. На додавання зображення існують обмеження: максимальний розмір файлу 5 MB, формат зображень «JPG», «PNG», «GIF».
Управління замовленнями	Зміна статусу замовлення (в наявності, немає в наявності або відобразити дату появи в наявності).
Керування зареєстрованими користувачами	Редагувати / видаляти користувачів із системи. Редагувати статус користувача, роль у системі (адміністратор/покупець), пароль користувача.

Джерело: запропоновано автором

При розробці вебзастосунку, дотримання всіх вказаних вище вимог покупців та адміністратора забезпечить високу якість обслуговування користувачів, ефективність продажів і конкурентоспроможність.

2.2 Вибір та обґрунтування засобів реалізації

Brackets [6] - це сучасний безкоштовний текстовий редактор з відкритим вихідним кодом, розроблений для веброзробників. Він розроблений і підтримується компанією Adobe Systems, відомою своїм програмним забезпеченням для дизайну та веброзробки. Вперше Brackets був представлений у 2014 році як інструмент з відкритим вихідним кодом, який активно розвивається спільнотою GitHub.

Brackets пропонує широкий спектр функцій, які роблять його потужним інструментом для веброзробки:

- функція «Live Preview»(попереднього перегляду в реальному часі) дозволяє бачити зміни в коді безпосередньо в браузері, що прискорює розробку та налагодження вебсторінок;
- Brackets підтримує HTML, CSS, JavaScript та інші мови, що використовуються у веброзробці;
- функція автоматичного завершення коду дозволяє розробникам писати код швидше, пропонуючи варіанти завершення тегів, атрибутів та інших елементів;
- Brackets підтримує роботу з препроцесорами, такими як LESS і SASS;
- за допомогою інтегрованого менеджера розширень доступно встановлювати додаткові плагіни та теми, які розширюють функціональність редактора;
- функція розділення робочого простору на кілька панелей для роботи з різними файлами одночасно, що підвищує продуктивність.

Основні переваги:

- у Brackets зручний для налаштування інтерфейс;
- велика кількість доступних розширень дозволяє адаптувати Brackets до конкретних потреб розробника, додавати нові функції та інтегрувати з іншими інструментами;
- розробники Brackets забезпечують регулярні оновлення, виправлення помилок і появу нових функцій;

- інтеграція з іншими продуктами Adobe, такими як Photoshop для імпорту графічних елементів і спрощення процесу створення вебдизайну;
- Brackets доступний для Windows, macOS і Linux.

Brackets - це потужний і гнучкий інструмент для веброзробників, який поєднує в собі зручність, функціональність і можливості налаштування. Відкритий вихідний код та підтримка компанії-розробника роблять його інструментом, що постійно розвивається і може задовольнити потреби користувача.

«PHP» [17] (від англ. PHP: Hypertext Preprocessor - «PHP: препроцесор гіпертексту» - це скриптова мова програмування з відкритим вихідним кодом, створена для розробки вебдодатків, яка широко використовується для генерації сторінок вебсервером.

У наш час мова «PHP» підтримується практично всіма хостинг-провайдерами і є основним інструментом розробки вебсайтів. Популярність її зумовлена простотою, швидкістю виконання і наявністю величезної кількості «модулів-розширень». Головним фактором мови PHP є практичність. PHP повинен надати програмісту засоби для швидкого та ефективного вирішення поставлених завдань.

«JavaScript» [15] (JS) - прототипно-орієнтована сценарна мова програмування. Є реалізацією мови «ECMAScript». JS зазвичай використовується як вбудована мова для програмного доступу до об'єктів додатків. Найширше застосування знаходить у браузерях як мова сценаріїв для надання інтерактивності вебсторінкам.

«HTML» [9] (від англ. HyperText Markup Language - «мова гіпертекстової розмітки») - стандартна мова розмітки документів у Глобальній мережі. Більшість вебсторінок містять опис розмітки мовою «HTML» (або «XHTML»). Мова «HTML» інтерпретується браузерами, отриманий у результаті інтерпретації форматований текст відображається на екрані монітора комп'ютера або мобільного пристрою.

«HTML» - це найпоширеніший інструмент створення сайтів, якщо звернути увагу на вихідні коди будь-якої сторінки в Інтернеті, то можна побачити, що серед усіх онлайн-проектів більшість пишеться саме на базі «HTML».

Головною перевагою «HTML» є швидкість завантаження сторінок. Це відбувається внаслідок маленького розміру сторінок на сервері. Швидкість завантаження сторінок своєю чергою впливає на пошукову видачу інформації.

«CSS» [14] (від англ. Cascading Style Sheets - «каскадні таблиці стилів») - це мова, призначена для оформлення вебсторінок і інших видів документів. Розробник стандартів «HTML» і «CSS», Консорціум Всесвітньої Павутини, рекомендує розділяти структуру і подання - тобто кодувати в HTML тільки логічну структуру документа, а все, що пов'язане з відображенням документа на екрані або поданням на інших пристроях, виконувати засобами «CSS». До основних переваг «CSS» належить:

- «чистіший» код легше підтримувати, швидше завантажується, краще оптимізований для пошукових систем;
- правила стилю можуть застосовуватися до безлічі сторінок, однаковий дизайн;
- точність контролю (позиціонування, розмір, поля).

Таким чином, CSS-код - це список інструкцій для браузера, - як і де відображати елементи вебсторінки. Під «елементами» зазвичай маються на увазі теги «XHTML/HTML» та їхній вміст.

Системою управління базами даних (СУБД) прийнято рішення використовувати «MySQL», тому що ця СУБД є найпопулярнішою і найпоширенішою в усьому світі.

«MySQL» [16] - це серверна реляційна система управління базами даних з відкритим кодом, яка поширюється за безкоштовною ліцензією.

СУБД «MySQL» розроблялася для вебсайтів і забезпечувала швидку індексацію даних у сховищах та оптимізацію послідовного доступу до даних.

З часом додано такий функціонал:

- кеш запитів - збільшує продуктивність;
- «MySQL Cluster» створено для масштабування БД більш ніж на один фізичний сервер.

Переваги «MySQL»:

- простота в роботі - встановити «MySQL» досить просто. Додаткові додатки, наприклад GUI, дають змогу доволі легко працювати з БД;
- багатий функціонал «MySQL» підтримує більшу частину функціонала SQL;
- функції, що забезпечують безпеку, підтримується за замовчуванням;
- «MySQL» розроблено для роботи з великими обсягами даних і легко масштабується;
- спрощення стандартів «MySQL» збільшує продуктивність.

У реляційній базі даних дані зберігаються в окремих таблицях, завдяки чому досягається більша швидкість та гнучкість.

Таблиці зв'язуються між собою за допомогою зв'язків, завдяки чому забезпечується можливість об'єднувати при виконанні запиту дані з декількох таблиць. SQL як частину системи MySQL можна схарактеризувати як мову структурованих запитів та найпоширенішу стандартну мову, що використовується для доступу до баз даних.

«phpMyAdmin» [18] - це програма з графічним інтерфейсом, яка використовується для керування базою даних MySQL. База даних та її таблиці створюються вручну. Він надає вебінтерфейс і працює на будь-якому сервері. Оскільки він веборієнтований, доступ до нього є з будь-якого комп'ютера.

«phpMyAdmin» підтримує функції, які наведені нижче:

- створювати, змінювати, переглядати та видаляти бази даних, подання, таблиці, стовпці та індекси;
- відображати декілька наборів результатів за допомогою запитів та збережених процедур;
- використовувати збережені процедури та запити для відображення декількох наборів результатів;
- відстежувати зміни, зроблені в базах даних, поданнях і таблицях;
- створювати PDF-графіку макета нашої бази даних;
- підтримка mysqli, що є покращеним розширенням MySQL;
- взаємодіяти з 80 різними мовами;

- додавати, виконувати й додавати в закладки будь-які SQL-оператори і навіть пакетні запити;
- створювати резервні копії бази даних у різних форматах.

Переваги «phpMyAdmin»:

- «phpMyAdmin» використовують на будь-якому сервері або в будь-якій операційній системі, оскільки має веббраузер;
- з допомогою «phpMyAdmin» адміністратор контролює дозволи користувачів і керує кількома серверами одночасно.

Для реалізації вебдодатку обрано текстовий редактор для веброзробки Brackets. Обрано мови для реалізації вебзастосунку - PHP, JavaScript, HTML і CSS.

Для адміністрування баз даних обрано phpMyAdmin, який розроблено з графічним інтерфейсом для керування базами даних MySQL.

РОЗДІЛ 3

ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ВЕБЗАСТОСУНКУ ДЛЯ ВЕДЕННЯ ЗАМОВЛЕНЬ НА ВИГОТОВЛЕННЯ СІЛЬСЬКОГОСПОДАРСЬКОЇ ПРОДУКЦІЇ

3.1 Розробка функціональних діаграм програмного засобу

UML діаграми варіантів використання моделюють поведінку системи і допомагають зафіксувати вимоги до системи [10].

Діаграми варіантів використання описують функції та сферу застосування системи. Ці діаграми також визначають взаємодію між системою та її акторами. Варіанти використання та актори на діаграмах варіантів використання описують те, що робить система і як актори її використовують, але не те, як система працює всередині.

Діаграми варіантів використання ілюструють і визначають контекст і вимоги або всієї системи, або її важливих частин. Ви можете моделювати складну систему за допомогою однієї діаграми варіантів використання або створити багато діаграм варіантів використання для моделювання компонентів системи. Діаграми варіантів використання розробляють на ранніх етапах проєкту і звертаються до них протягом усього процесу розробки.

Діаграми варіантів використання корисні в наступних ситуаціях:

- перед початком проєкту. Створена діаграма варіантів використання для того, щоб всі учасники проєкту мали спільне розуміння працівників, клієнтів та діяльності бізнесу;
- під час збору вимог, щоб зафіксувати вимоги до системи та представити іншим, що повинна робити система;
- використовувати на етапах аналізу та проєктування варіанти використання та акторів з діаграм варіантів використання, щоб визначити класи, які потрібні системі.

Елементи моделі на діаграмах варіантів використання:

- варіанти використання. Варіант використання описує функцію, яку виконує система для досягнення мети користувача. Варіант використання повинен давати видимий результат, який має цінність для користувача системи;

- актори. Актор представляє роль користувача, який взаємодіє з системою, яка проєктується. Користувачем може бути людина, організація, машина або інша зовнішня система;

- підсистеми. У моделях UML підсистеми - це тип компонентів, які представляють незалежні, поведінкові одиниці в системі. Підсистеми використовуються на діаграмах класів, компонентів і варіантів використання для представлення великомасштабних компонентів у системі, яку ви моделюєте;

- зв'язки на діаграмах варіантів використання. В UML відношення - це зв'язок між елементами моделі. Зв'язок в UML - це тип елемента моделі, який додає моделі семантики, визначаючи структуру і поведінку між елементами моделі.

Діаграма варіантів використання зображена на рисунку 3.1.

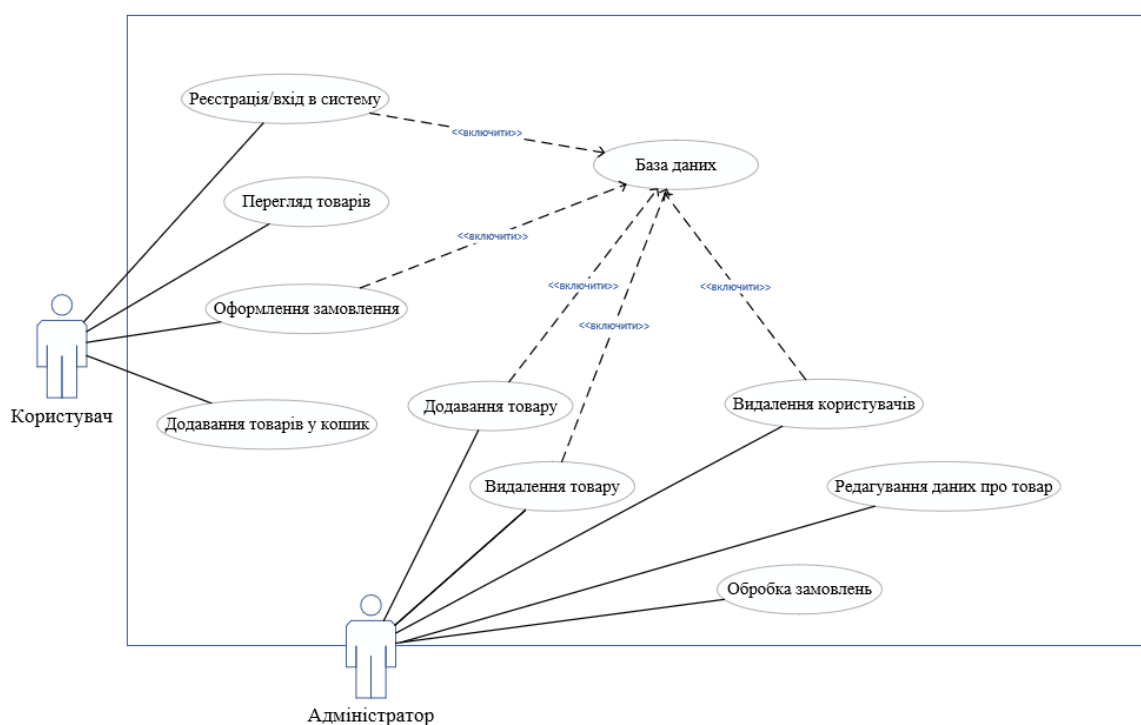


Рисунок 3.1 - Діаграма варіантів використання

Джерело: запропоновано автором

Діаграма потоків даних (DFD) або контекстна діаграма рівня 0 [12] - це найвищий рівень діаграми потоків даних (DFD), створена для уявлення про всю систему. Створена для показу основних процесів, потоків даних і сховища даних в системі, не надаючи жодних деталей про внутрішню роботу цих процесів.

Це представлення, що показує систему як єдиний процес з його взаємозв'язками із зовнішніми сутностями. Вона представляє всю систему у вигляді одного прямокутника з вхідними та вихідними даними, позначеними стрілками вхідних/вихідних даних. Нижче наведено компоненти діаграми потоку даних.

Зовнішня сутність, відома як термінатор, актор, є компонентом діаграми потоків даних (DFD), який знаходиться за межами системи й взаємодіє з системою. Це можуть бути організації, такі як банки, групи людей, такі як клієнти, або різні відділи однієї організації, які не є частиною модельованої системи і є зовнішніми сутностями. Модельовані системи також взаємодіють з термінатором.

Перетворення входу на вихід у системі відбувається завдяки функції процесу. Символами процесу є прямокутник із заокругленими кутами, овал, прямокутник або коло. Процес підписано коротким реченням, одним словом або словосполученням, щоб висловити його суть.

Потік даних описує передачу інформації між різними частинами системи. Символом потоку даних є стрілка. Потоку слід дати відповідну назву, щоб визначити інформацію, яка переміщується. Потік даних також представляє матеріал разом з інформацією, яка переміщується. Матеріальні переміщення моделюються в системах, які не є просто інформативними. Даний потік повинен передавати тільки один тип інформації. Напрямок потоку зображується стрілкою, яка також може бути двонапрямною.

Дані зберігаються у сховищі для подальшого використання. Дві горизонтальні лінії представляють символ сховища. Сховище не обмежується лише файлом даних, це може бути що завгодно, наприклад, тека з документами, оптичний диск, картотека. Сховище даних можна розглядати незалежно від його реалізації. Коли дані надходять зі сховища, це називається зчитуванням даних, а коли дані надходять до сховища, це називається введенням або оновленням даних.

Контекстна діаграма обробки замовлень на виготовлення обладнання зображена на рисунку 3.2.

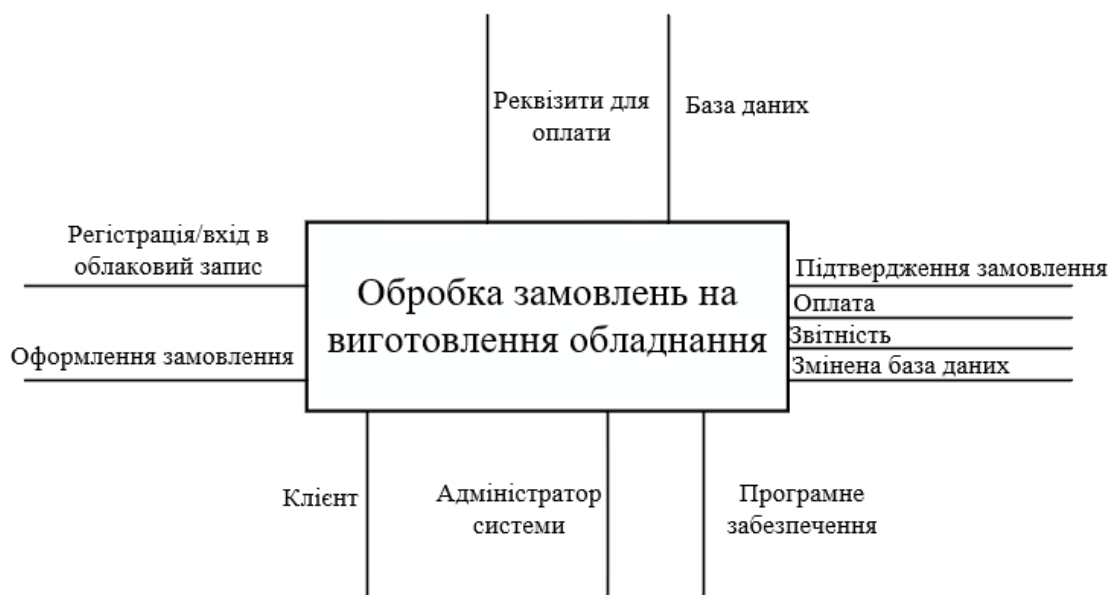


Рисунок 3.2 - Контекстна діаграма обробки замовлень

Джерело: запропоновано автором

DFD ілюструє, як дані проходять через систему, а UML - це мова моделювання, яка використовується в об'єктно-орієнтованому проєктуванні програмного забезпечення для надання більш детального уявлення.

3.2 Проєктування бази даних вебзастосунку для ведення замовлень на виготовлення сільськогосподарського обладнання

База даних вебзастосунку для ведення замовлень, формується на основі документів та довідників. Перелік документів та довідників наведений в таблиці 3.1.

Таблиця 3.1 – Інформаційні сутності

№	Назва інформаційного об'єкту	Тип документа
1	2	3
1	Замовлення продукції	Реєстр замовлення

Продовження таблиці 3.1

1	2	3
2	Адреса	Реєстр адреси
3	Обрана продукція	Реєстр обраної продукції
4	Продукція	Довідник
5	Користувачі	Реєстр користувачів

Джерело: запропоновано автором

Опис реквізитів інформаційних об'єктів. У таблиці 3.2 наведений перелік усіх реквізитів вхідних і вихідних документів.

Таблиця 3.2 - Перелік усіх вхідних і вихідних документів.

№ п/п	Реквізит	Позначення	Тип даних, розмір
1	ІД Користувача	Id_user	Числовий
2	Ім'я	Name	Текстовий
3	Прізвище	Surname	Текстовий
4	Телефон користувача	Phone	Числовий
5	Електронна пошта	Mail	Текстовий
6	Пароль	Password	Текстовий
7	Назва продукту	Name	Текстовий
8	Класифікація	Classification	Текстовий
9	Тип	Type	Текстовий
10	Опис повний	Description	Текстовий
11	Короткий опис	S_description	Текстовий
12	Картинка товару	Image	Текстовий
13	Ціна товару	Price	Числовий
14	ІД Доставка	Id_delivery	Числовий
15	Область	Region	Текстовий
16	Місто	City	Текстовий
17	Тип доставки	Type	Текстовий
18	Адреса доставки	Address	Текстовий
19	Тип доставки	Pay_method	Числовий
20	Сума за всі товари	Total_price	Числовий
21	ІД обраної продукції	Id_faviforite	Числовий
22	ІД товару	Id_product	Числовий
23	Кількість товару	Count	Числовий
24	Статус	Status	Текстовий
25	Код замовлення	Code	Числовий
26	ІД Адреса	Id	Числовий

Джерело: запропоновано автором

Щоб створити структуру для кожного інформаційного об'єкта, використаємо дані з таблиці 3.2. У таблиці 3.3 ми визначимо поля кожної таблиці бази даних,

вказавши основне поле (ключ), яке ідентифікує кожний запис у таблиці. Ці ключі будуть використовуватися для з'єднання даних між різними таблицями.

Таблиця 3.3 - Структурний склад.

Інформаційний об'єкт		Інформаційна одиниця	Ключ
1	Замовлення	Код замовлення ІД товару Кількість товару ІД Користувача ІД Доставка Статус	*
2	Адреса	ІД Адреса ІД Користувача Область Місто Тип Адреса доставки Тип доставки Сума за всі товари	* *
3	Обрана продукція	ІД обраної продукції ІД товару ІД Користувача Кількість товару Статус	* * *
4	Продукція	ІД товару Назва продукту Класифікація Тип Опис повний Короткий опис Картинка товару Ціна товару	*
5	Користувачі	ІД Користувача Ім'я Прізвище Телефон користувача Електронна пошта Пароль	*

Джерело: запропоновано автором

Структурний взаємозв'язок між інформаційними об'єктами в базі даних створюється шляхом визначення ключових полів, спільних для декількох таблиць. Ці зв'язки дозволяють взаємодіяти між різними наборами даних, що зберігаються в різних таблицях, і одночасний доступ до цих даних через взаємодію між

відповідними ключами. Опис структурних зв'язків представлено у таблиці 3.4, яка чітко вказує спільні ключові поля та зв'язки між таблицями в базі даних.

Таблиця 3.4 - Опис структурних зв'язків інформаційних об'єктів.

№ п/п	Інформаційні об'єкти		Тип відношення	Ключ зв'язку
	Головний	Підпорядкований		
1	Продукція	Обрана продукція	1:Б	ІД товару
2	Обрана продукція	Замовлення	1:Б	ІД товару
3	Користувачі	Обрана продукція	1:Б	ІД Користувача
4	Користувачі	Замовлення	1:Б	ІД Користувача
5	Користувачі	Адреса	1:Б	ІД Користувача
6	Адреса	Замовлення	1:Б	ІД Доставка

Джерело: запропоновано автором

Діаграма взаємозв'язків сутностей (ER діаграма) [7] — це візуальне представлення різних сутностей у системі та їх зв'язку між собою. Це інструмент, який використовують для проєктування та моделювання реляційних баз даних і показує логічну структуру бази даних. У діаграмі ER використано символи для представлення сутностей, атрибутів і відношень, які допомагають пояснити зв'язок між сутностями в базі даних.

Сутності зображено у вигляді прямокутників, а зв'язки — у вигляді ромбів. Лінії, що з'єднують сутності та зв'язки, представляють відношення один до одного, вказуючи на його тип (наприклад, «один до одного», «один до багатьох» або «багато до багатьох»). В таблиці 3.5 описано зв'язки між інформаційними об'єктами.

Таблиця 3.5 - Назви зв'язків між інформаційними об'єктами

№п/п	Перший інформаційний об'єкт	Другий інформаційний об'єкт	Назва зв'язку
1	Продукція	Обрана продукція	Вибір
2	Користувачі	Обрана продукція	Вибір
3	Обрана продукція	Замовлення продукції	Замовлення
4	Обрана продукція	Адреса	Внесення

Джерело: запропоновано автором

На рисунку 3.3 зображена схема діаграми взаємозв'язків сутностей. На діаграмі показано як користувачі обирають продукцію, роблять замовлення і як замовлення пов'язані з адресами доставки.

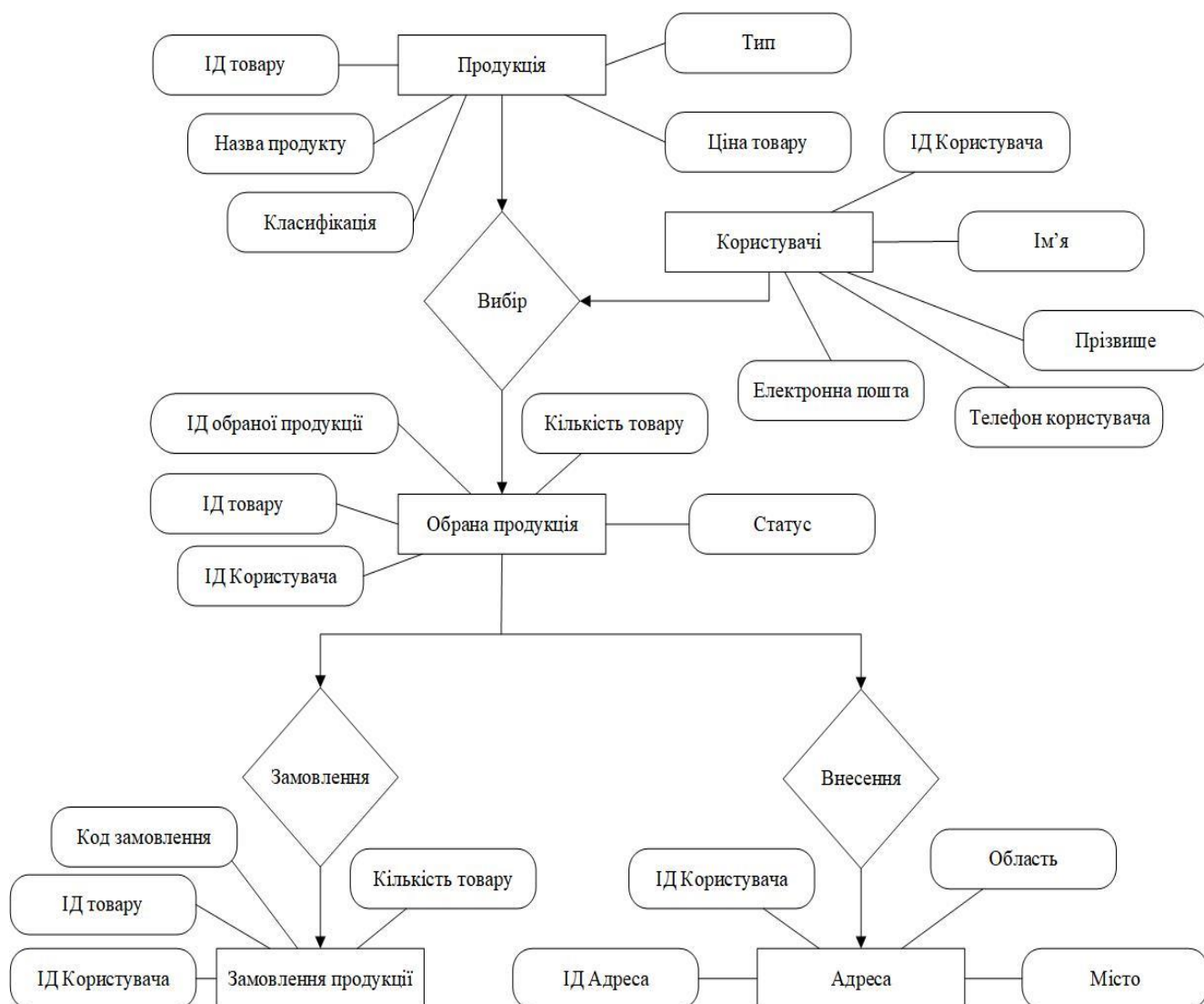


Рисунок 3.3 - структура ER діаграми

Джерело: запропоновано автором

Логічна схема визначає структуру даних у базі даних, не вказуючи, як ці дані фізично зберігаються. Схема бази даних містить об'єкти схеми, які містять:

- таблиці;
- поля;
- пакети;
- подання;
- зв'язки;
- первинний ключ;
- зовнішній ключ [13].

Логічна схема зображена на рисунку 3.4.

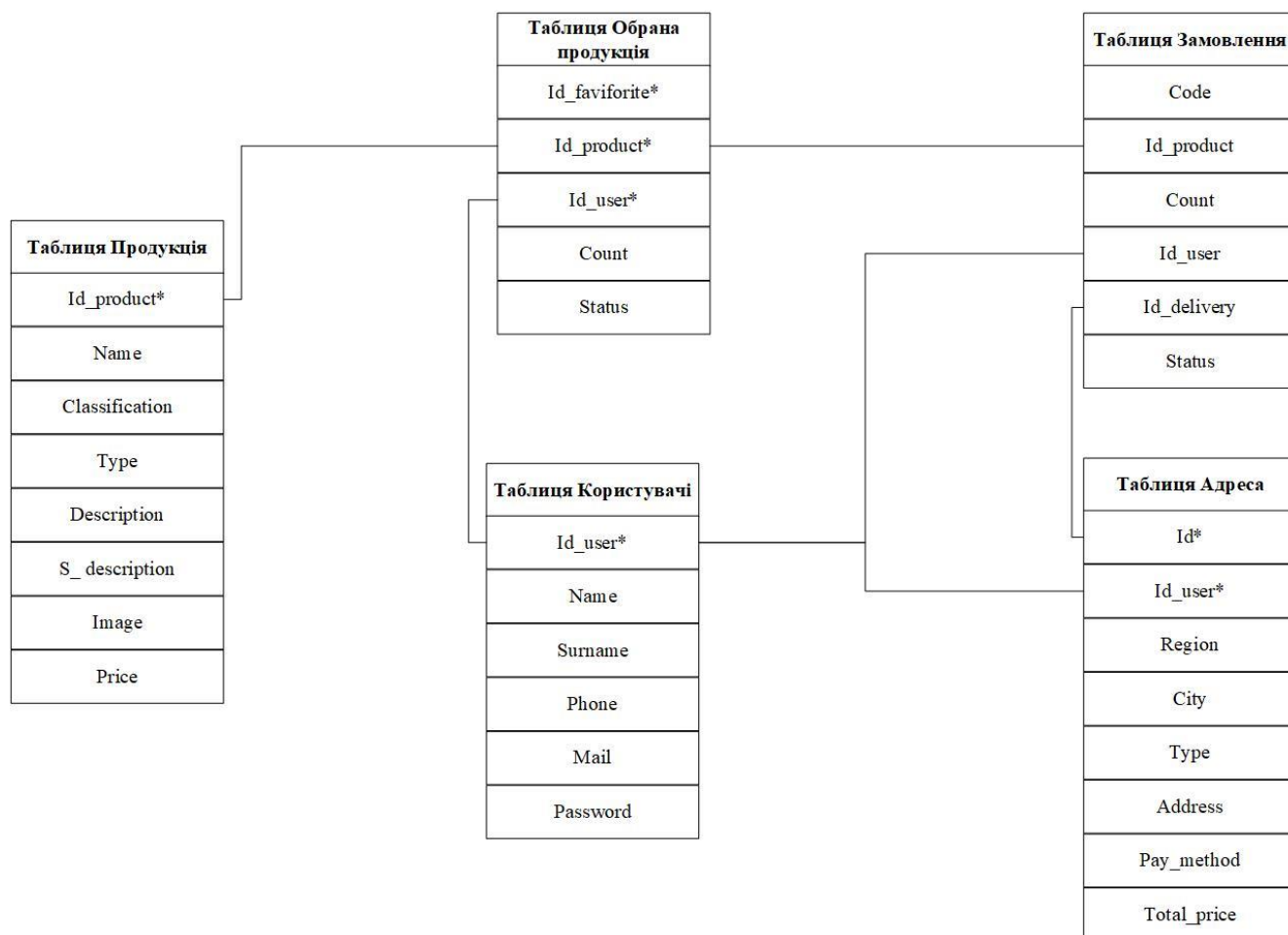


Рисунок 3.4 - Логічна схема бази даних

Джерело: запропоновано автором

3.3 Створення макетів сторінок вебзастосунку

Загальна будова вебзастосунку має складатися з таких розділів:

- розділ адміністрування;
- головна сторінка;
- каталог товарів;
- кошик покупця;
- інформація про компанію;
- оформлення замовлення;
- доставка та оплата.

Інтерфейс вебзастосунку повинен надавати користувачеві інтуїтивно зрозумілу навігацію на всьому сайті, можливість просто і швидко здійснити перехід

від розділу до розділу. Елементи навігації мають бути зрозумілими для користувача: кожне посилання повинне мати заголовок, що відображає їхнє призначення, графічні навігаційні елементи мають позначатися текстом.

Сторінки інтернет-магазину повинні бути динамічно заповнюваними, відповідно до обраного користувачем розділу сайту, програмним методом ґрунтуючись на інформацію, що знаходиться в базі даних.

Додавання, зміна і видалення інформації в розділах сайту має виконуватися адміністратором сайту. Для цих завдань адміністратор повинен мати доступ до спеціальних модулів, які б давали йому здатність контролювати і керувати вмістом інтернет-магазину без знань мов програмування.

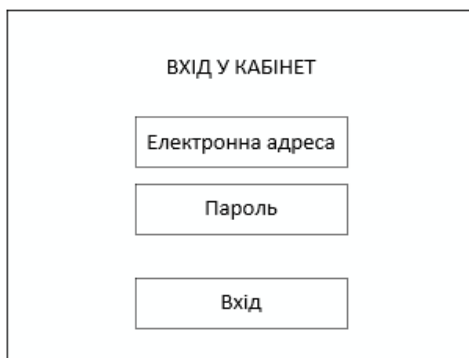
Кожен розділ має містити в собі два обов'язкових модулі:

- навігаційне меню;
- контактні дані.

Навігаційне меню містить такі елементи:

- посилання, що спрямовує користувача на головну сторінку сайту;
- посилання, що спрямовує користувача на сторінку з інформацією про компанію;
- посилання, що вказують шлях до поточної сторінки;
- посилання на особистий кабінет.

Для сторінки авторизації користувача розроблено макет, який зображено на рисунку 3.5.



Макет сторінки авторизації, що складається з прямокутної рамки. В середині рамки розташовані три елементи: нагору — текст «ВХІД У КАБІНЕТ», за ним — поле для введення «Електронна адреса», далі — поле для введення «Пароль», і в самому низу — кнопка «Вхід».

Рисунок 3.5 - Макет сторінки авторизації

Джерело: запропоновано автором

У розділі категорій товарів та товари, здійснено виведення всіх подібних товарів, з коротким описом кожного: назва товару, мале зображення і ціна товару, а також блок із налаштуваннями фільтра для зручного пошуку потрібних товарів (рисунок 3.6).



Рисунок 3.6 - Макет головної сторінки

Джерело: запропоновано автором

Сторінка товару повинна містити у вигляді списку всі характеристики, що стосуються цього товару, а також велике зображення цього товару. Також обов'язкова наявність кнопки додавання товару в кошик (рисунок 3.7).



Рисунок 3.7 - Макет сторінки товару

Джерело: запропоновано автором

Макет реєстрації користувача зображено на рисунку 3.8.

The wireframe shows a registration form layout. At the top left is a box for the logo. Below it is a horizontal navigation menu. The main content area contains a vertical stack of input fields for 'Ім'я', 'Прізвище', 'Телефон', 'Електронна адреса', and 'Пароль', followed by a 'Кнопка «Реєстрація»'. At the bottom, there are three boxes: 'Соціальні мережі' on the left, 'Контактні номери' in the center, and 'Електронна пошта' on the right.

Рисунок 3.8 - Макет реєстрації

Джерело: запропоновано автором

Особистий кабінет користувача має містити обрані користувачем товари та таку інформацію про них:

- назва товару;
- зображення товару;
- ціна товару;
- кількість товару;
- загальна сума товару;
- посилання на видалення товару з кошика.

Особистий кабінет розроблено, щоб зробити процес вибору товару та оформлення замовлення зручним та зрозумілим для користувача і надати усі інструменти, необхідні для ефективного управління покупками. Особистий кабінет – це розділ, щоб зберігати обрані товари та відстежувати замовлення.

Особистий кабінет користувача зображено на рисунку 3.9.



Рисунок 3.9 - Макет сторінки «Особистий кабінет»

Джерело: запропоновано автором

Сторінка оформлення замовлення має містити список товарів, графи для введення інформації про доставку та оплату, загальну суму замовлення(рис. 3.10).

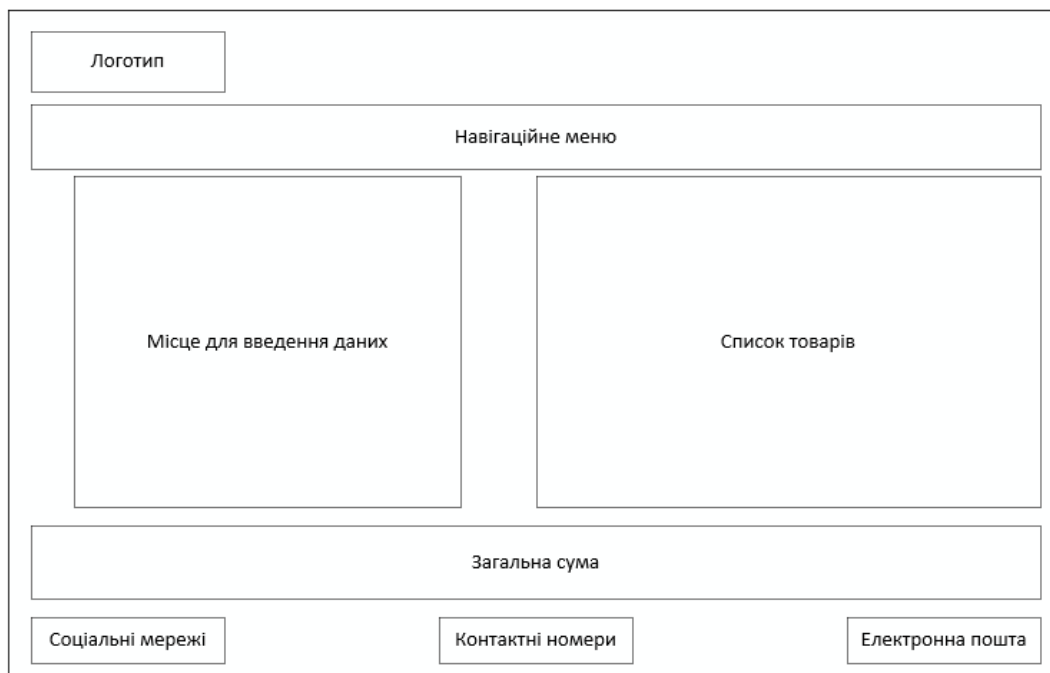


Рисунок 3.10 - Макет сторінки оформлення замовлення

Джерело: запропоновано автором

3.4 Реалізація програмного продукту

Фізична реалізація бази даних вебзастосунку для ведення замовлень на виготовлення сільськогосподарського обладнання зображена на рисунку 3.11.

Таблиця	Дія	Рядки	Тип	Зіставлення	Розмір	Фрагментовані
☐ buy_table	☆ [іконки]	3	InnoDB	utf8mb4_0900_ai_ci	64.0 КБ	-
☐ delivery_table	☆ [іконки]	2	InnoDB	utf8mb4_0900_ai_ci	32.0 КБ	-
☐ favorites_table	☆ [іконки]	5	InnoDB	utf8mb4_0900_ai_ci	48.0 КБ	-
☐ product_table	☆ [іконки]	19	InnoDB	utf8mb4_0900_ai_ci	64.0 КБ	-
☐ user_table	☆ [іконки]	3	InnoDB	utf8mb4_0900_ai_ci	16.0 КБ	-
5 таблиць	Всього	32	InnoDB	utf8mb4_0900_ai_ci	224.0 КБ	0 Б

Рисунок 3.11 - Перелік таблиць бази даних

Джерело: запропоноване автором

Таблиця бази даних створюється відповідно до структури інформаційних об'єктів, які описано в таблиці 3.3. Схема бази даних зображена на малюнку 3.12 ілюструє структуру бази даних для управління даними у вебзастосунку. Вона містить наступні елементи:

1. Таблиці:

- кожна таблиця представлена прямокутником і містить поля, які описують атрибути таблиці;
- ключові поля (первинні та зовнішні ключі) позначені спеціальними позначками для ідентифікації.

2. Зв'язки між таблицями:

- лінії між таблицями показують зв'язки (відносини) між різними таблицями бази даних;
- зв'язки можуть бути одного з трьох типів: один-до-одного, один-до-багатьох та багато-до-багатьох;
- вказано, які поля використовуються для створення зв'язків між таблицями.

3. Атрибути полів:

- для кожного поля таблиці вказані назва, тип даних та обмеження.

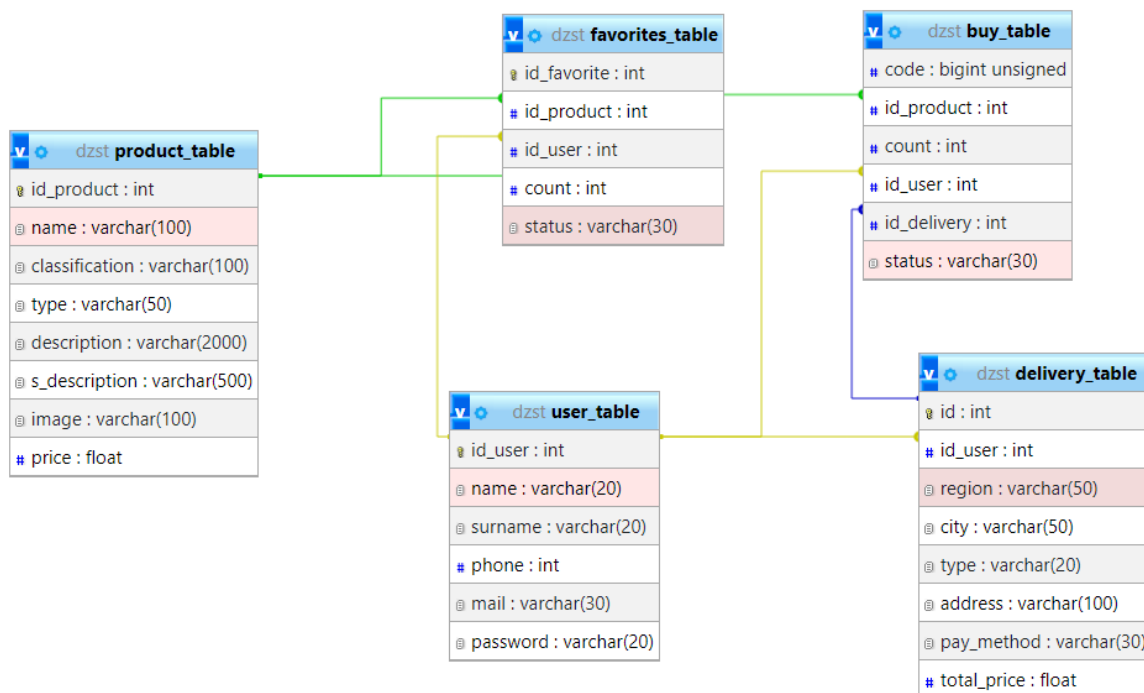


Рисунок 3.12 - Схема бази даних

Джерело: запропоноване автором

Схема бази даних допомагає зрозуміти структуру та взаємозв'язок даних, а також способи зберігання та управління у системі. Це важливо для проєктування, розробки та підтримки вебзастосунку, забезпечуючи зручність роботи з даними та підвищуючи ефективність системи.

Створення вебзастосунку - процес, який охоплює різні аспекти. Для опису результатів розробки прийнято рішення розділити опис на три основні рівні взаємодії: взаємодія користувача з вебзастосунком, взаємодія адміністратора з вебзастосунком і взаємодія адміністратора з користувачем.

Перед аутентифікацією користувача необхідно, щоб цей користувач зареєструвався в системі. Для цього, відповідно до дизайн-макета, на клієнтській частині вебдодатка реалізовано сторінку реєстрації і форму реєстрації, поля якої проводять валідацію введених даних.

«Сторінка реєстрації» зображена на рисунку 3.13.

СТОРІНКА РЕЄСТРАЦІЇ

Ім'я:

Прізвище:

Телефон:

Електронна адреса:

Пароль:

Введи надійний пароль

Повтори свій пароль

Реєстрація

Рисунок 3.13 - Сторінка та форма реєстрації

Джерело: запропоновано автором

Повідомлення про незаповнене поле зображено на рисунку 3.14.

Телефон 

Рисунок 3.14 - Повідомлення про незаповнене поле

Джерело: запропоновано автором

Після реєстрації, аутентифікація користувача на сайті відбувається під час авторизації в системі. Сторінка авторизації на клієнтській частині вебзастосунку зображено на рисунку 3.15.

Під час зазначення даних авторизації надсилається запит на серверну частину вебзастосунку, де перевіряються зазначені дані. Якщо вказану адресу електронної

пошти було не знайдено або вказаний пароль неправильний, то сторінка оновлюється. Якщо адресу електронної пошти було знайдено, то відбувається перехід на головну сторінку вебзастосунку.

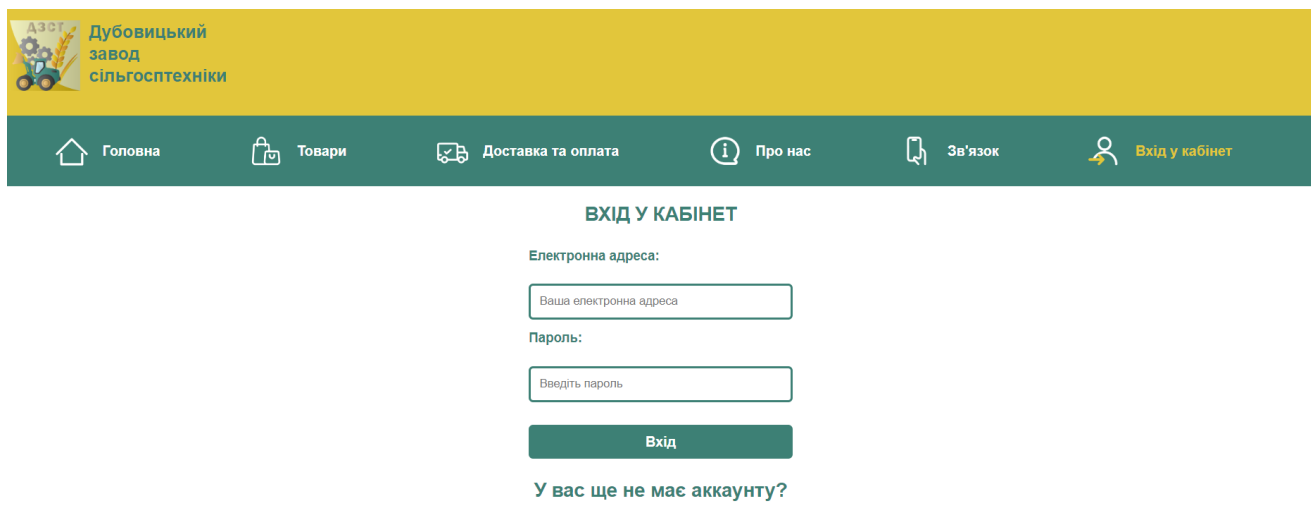


Рисунок 3.15 - Сторінка авторизації користувача

Джерело: запропоновано автором

Після авторизації відкривається головна сторінка, створена відповідно до дизайн-макета (рисунок 3.16).

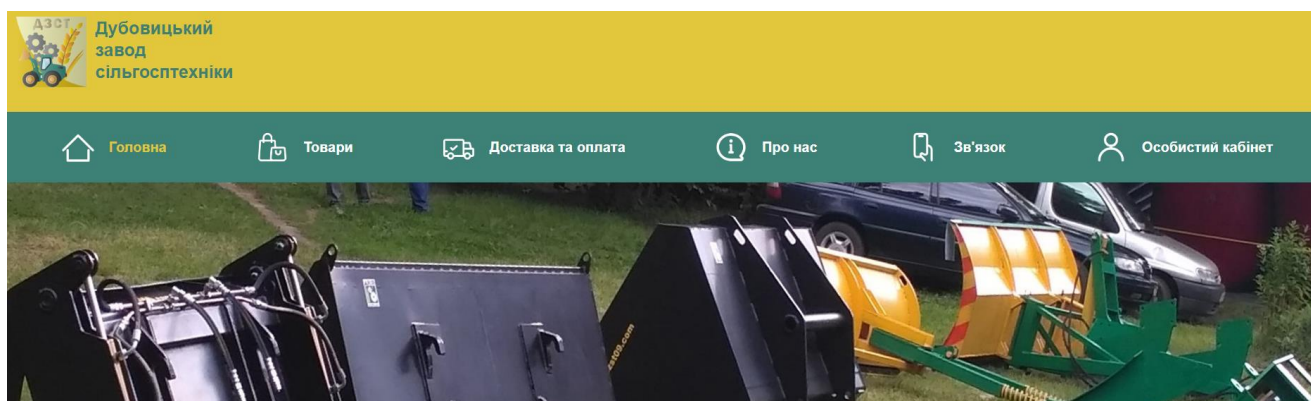


Рисунок 3.16 - «Головна сторінка» вебзастосунку

Джерело: запропоновано автором

Слайдер є елементом управління візуальною інформацією на головній сторінці та використовується для привернення уваги клієнтів до особливих

пропозицій, продуктів чи подій. Такий підхід допомагає підвищити ефективність веб-застосунку як засобу комунікації та маркетингу.

На головній сторінці, відповідно до дизайн-макета, було реалізовано слайдер (рисунок 3.17).

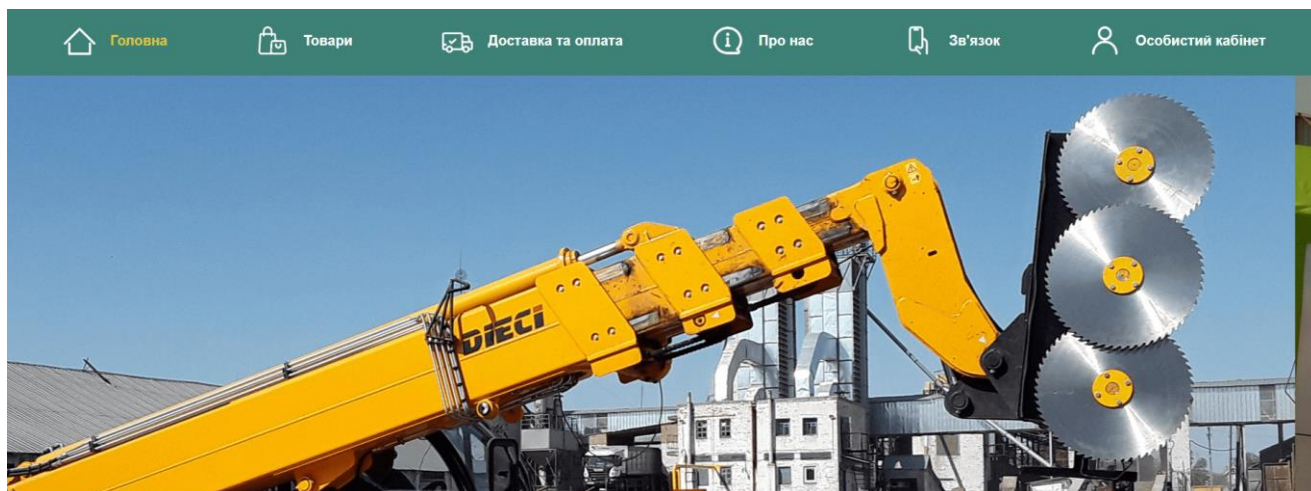


Рисунок 3.17 - Слайдер головної сторінки

Джерело: запропоновано автором

На сторінці «Товари» є фільтри за категоріями, а також можливість сортування товарів за найменуванням. Для зручного пошуку товару за назвою було створено функцію швидкого пошуку (рисунок 3.18).



Рисунок 3.18 - Сторінка каталогу товарів

Джерело: запропоновано автором

Товар представлений на сторінці каталогу товарів у вигляді карток, де вказана назва, опис виробу, при наведенні курсора на зображення, воно збільшується (рисунок 3.19).



Рисунок 3.19 - Картка товару

Джерело: запропоновано автором

Користувач отримає детальну інформацію про товар, перейшовши на сторінку обраного товару (рис. 3.20).



Опис:

КІВШ ДЛЯ ЗАВАНТАЖЕННЯ МІШКІВ БІГ-БЕГ

Ківш для завантаження мішків біг-бег повністю замінює навантажувальний ківш та стаціонарний дозатор. Крім економії коштів на два агрегати, за допомогою цього ковша всі роботи з фасування (наповнення) біг-бегів можуть проводитися однією людиною. Ківш для завантаження мішків біг-бег застосовується у сільськогосподарській та переробній галузі. За допомогою даного обладнання ви зможете легко зробити фасування (наповнення) сипучих матеріалів у біг-бег (мішки). Найважливішим елементом ковша є люк, який щільно закривається, що дозволяє за допомогою одного циліндра чітко контролювати і дозувати норму заповнення. Ківш для завантаження мішків оснащений кріпленням під фронтальний навантажувач.

Ціна: 63000 грн

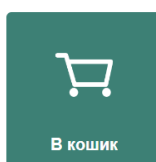


Рисунок 3.20 - Індивідуальна сторінка товару

Джерело: запропоновано автором

Додавання товару в кошик покупок відбувається через натискання кнопки «В кошик» на сторінці товару, якщо користувач авторизований у системі.

Перейшовши на сторінку кошика покупок, користувачеві буде надано можливість відредагувати кількість обраних товарів, видалити товари.

При перегляді товару на його сторінці, користувачеві доступна функція додавання товару в «Обране». За необхідності користувач може видалити товар із розділу «Обране» (рисунок 3.21).

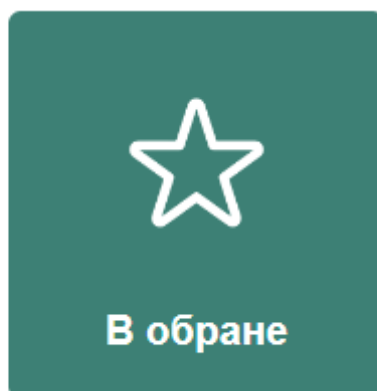


Рисунок 3.21 - Кнопка «В обране»

Джерело: запропоновано автором

Коли користувач готовий до оформлення замовлення, він може перейти до кошика, де йому буде надана можливість відредагувати кількість обраних товарів, видалити товари з кошика (рисунок 3.22).

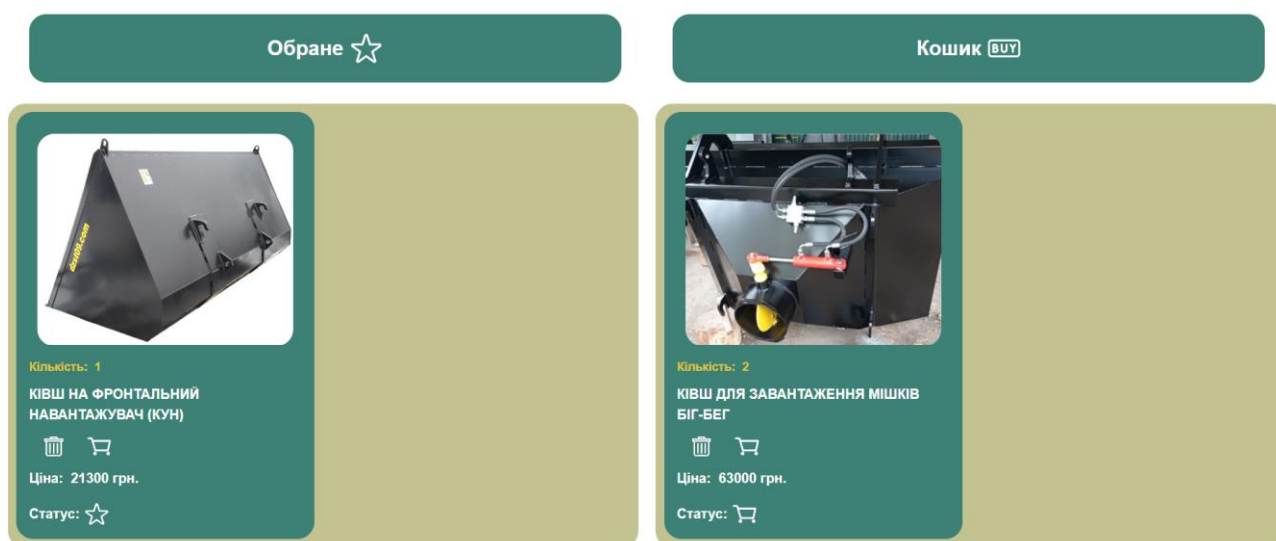


Рисунок 3.22 - Сторінка «Обране» та «Кошик»

Джерело: запропоновано автором

При натисканні на кнопку «Замовлення» користувачу стає доступна форма заповнення даних доставки (рисунок 3.23).

ОФОРМЛЕННЯ ЗАМОВЛЕННЯ:

Вкажіть область:

Вкажіть місто або село:

Оберіть спосіб доставки:

- ☐ Кур'єрська доставка
- ☐ Поштова доставка
- ☐ Самовивіз

Вкажіть адресу:

Оберіть спосіб оплати:



Кількість: 3

КІВШ ДЛЯ ЗАВАНТАЖЕННЯ МІШКІВ БІГ-БЕГ

Ціна: 63000 грн.

Статус: BUY



Кількість: 1

КІВШ НА ФРОНТАЛЬНИЙ НАВАНТАЖУВАЧ (КУН)

Ціна: 21300 грн.

Статус: BUY

Рисунок 3.23 - Сторінка оформлення замовлення

Джерело: запропоновано автором

Після введення даних для замовлення та його підтвердження, користувач перенаправляється на сторінку з замовленнями. На ній показані всі замовлення, їх номер та статус (рисунок 3.24).

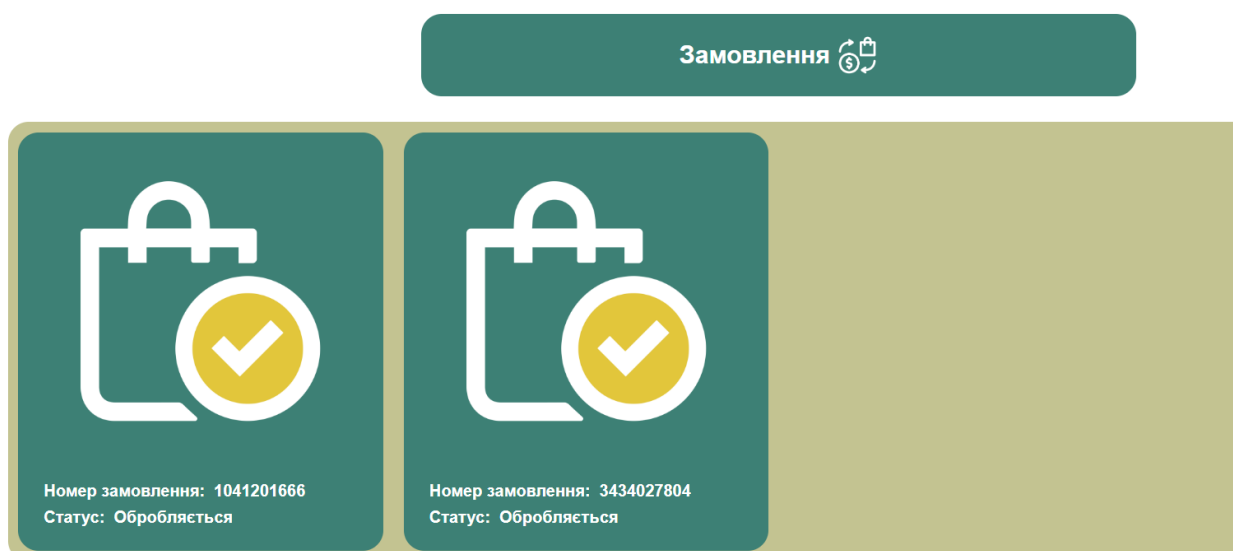


Рисунок 3.24 - Сторінка замовлень

Джерело: запропоновано автором

Сторінка «Доставка та оплата» містить інформацію про способи доставки, умови оплати, географічне розташування підприємства та реквізити для оплати (рисунок 3.25).

ДОСТАВКА ТА ОПЛАТА

Найпоширеніший варіант – доставка товарів в будь-яку точку України за допомогою таких перевізників, як Нова пошта, Делівері, Інтайм, САТ, Автолюкс. Ці перевізники транспортують всі наші товари крім навантажувача ДубОк.

Також можливий варіант перевезення товарів транспортом нашої компанії прямо Вам на підприємство. Якщо буде потрібно, наші фахівці приїдуть до Вас і здійнять монтаж, наладку та навчання.

Не виключений варіант доставки товару Вашим транспортом з нашим завантаженням.

Варіанти оплати – готівковий і безготівковий розрахунок. Тут два варіанти. Якщо замовлений Вами товар є на складі, то Ви оплачуєте товар за фактом продажу (в момент доставки) – 100%. А якщо ми працюємо за Вашим індивідуальним замовленням, то передплата становить 50%.

МИ НА КАРТІ



НАШІ РЕКВІЗИТИ

Назва: ТОВ «ДЗСТ 2»
 Код: 44765929
 ІНН: 447659218079
 Р/р: UA903375680000026003300459145 АТ «Ощадбанк» м.Суми
 МФО: 337568
 Адреса: 41321, Сумська область, Конотопський район, с. Дубовичі, вул. Шевченка 6
 Тел. моб.: (098) 661-7616
 Ел. адреса: dzst@ukr.net
 Директор : Онікієнко Віктор Борисович
 Платник ПДВ. Загальна форма оподаткування.

Рисунок 3.25 - Сторінка «Доставка та оплата»

Джерело: запропоновано автором

Для надання користувачеві інформації про локацію компанії створено інформаційну сторінку з використання API GoogleMap (рис. 3.26).



Рисунок 3.26 - Географічне розташування підприємства

Джерело: запропоновано автором

На сторінці товару був створений розділ питань, де користувачі залишити своє питання щодо товару (рисунок 3.27).

ЗАЛИШТЕ СВОЇ ПИТАННЯ ТУТ:

ПІБ:

Електронна адреса:

Повідомлення:

НАДІСЛАТИ

ОЧИСТИТИ

Рисунок 3.27 - Сторінка «Зв'язок»

Джерело: запропоновано автором

Для управління контентом вебзастосунку створено розділи адмінпанелі «Додати товар», «Замовлення», та «Користувачі».

Розділ роботи з товаром надає адміністратору список усіх товарів з основної інформації про них, а також кнопку видалення (рисунок 3.28).

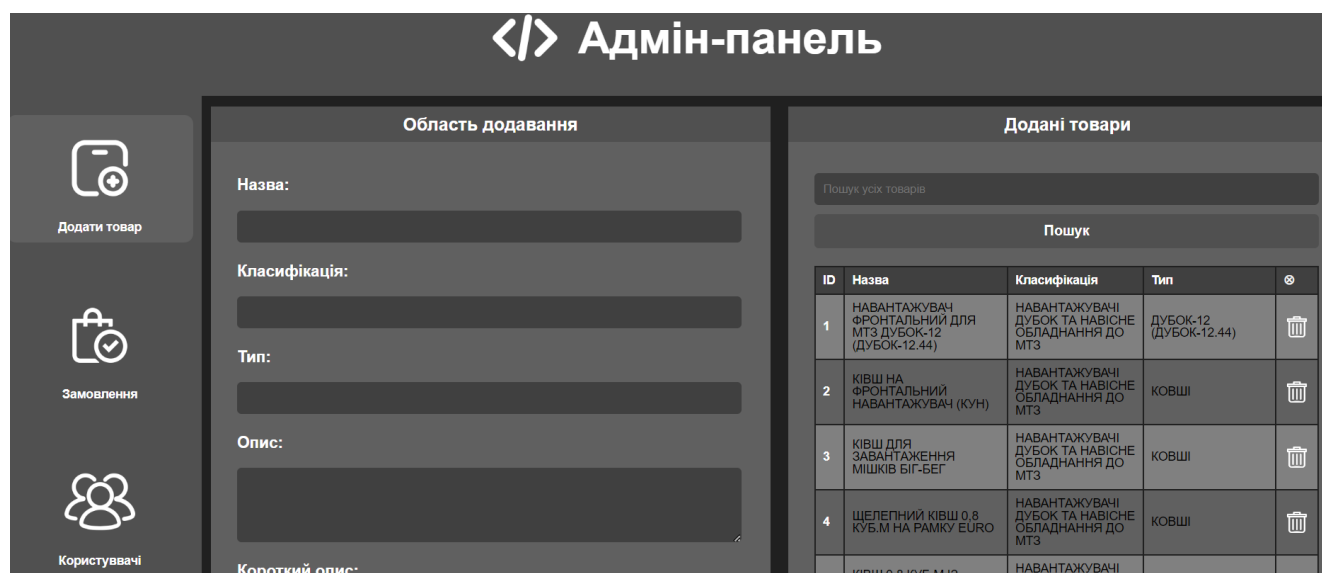


Рисунок 3.28 - Розділ «Додати товар»

Джерело: запропоновано автором

Створено розділ, який дозволяє переглядати інформацію про замовлення та змінювати його статус (рисунок 3.29).

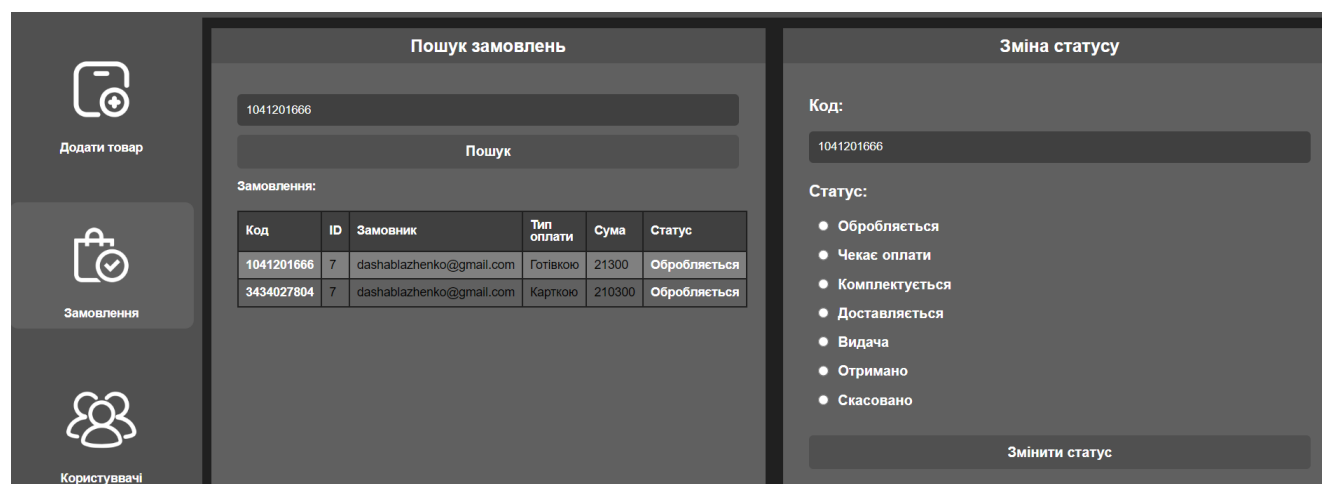


Рисунок 3.29 - Розділ адмінпанелі для роботи з замовленнями

Джерело: запропоновано автором

В розділі «Користувачі» відображаються усі зареєстровані користувачі у вебзастосунку (рисунок 3.30).

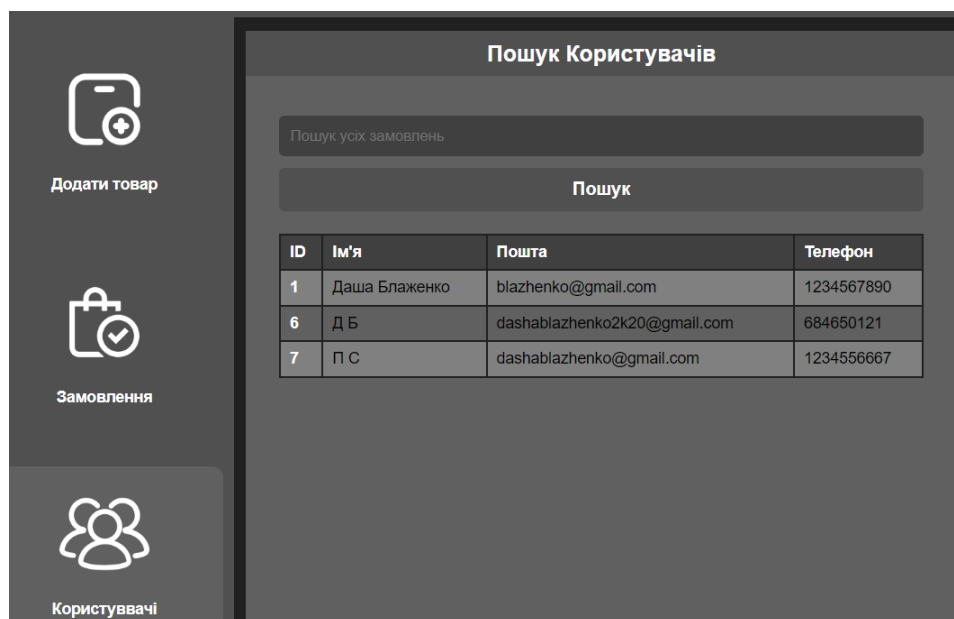


Рисунок 3.30 - Розділ «Користувачі»

Джерело: запропоновано автором

В результаті цього етапу здійснено розробку основних функціональних елементів вебзастосунку для ведення замовлень на виготовлення сільськогосподарського обладнання. Розроблено функціональні елементи взаємодії користувача з вебзастосунком. Для керування контентом вебзастосунку розроблено панель адміністратора, щоб виконувати операції зі створення, видалення та редагування даних. Також, розроблено інформаційні сторінки, на яких вказано користувачам інформацію про географічне розташування підприємства, способи доставки та типи оплати замовлень.

ВИСНОВКИ

Мета цієї кваліфікаційної роботи полягала в проєктуванні та розробці вебзастосунку для ведення замовлень на виготовлення сільськогосподарського обладнання для підприємства ТОВ «ДЗСТ 2».

Проведено аналіз діяльності ТОВ «ДЗСТ 2», вивчено організацію торговельної діяльності в інтернеті, проаналізовано чинні реалізації вебзастосунків для ведення замовлень на виготовлення сільськогосподарського обладнання з метою зрозуміти сутність вебзастосунку. На основі досліджень виявлено основні функціональні елементи вебзастосунку для ведення замовлень та потреби користувачів.

Виконано постановку задачі для розробки вебзастосунку для ведення замовлень з усіма вимогами до функціонала вебзастосунку та обґрунтовано вибір засобів реалізації. Редактором коду обрано програму «Brackets» компанії Adobe Systems. Реалізовано вебзастосунок технологіями HTML, PHP, CSS, JavaScript.

Для реалізації бази даних обрано систему управління базами даних MySQL, а для керування цією базою вирішено використовувати програму з графічним інтерфейсом phpMyAdmin.

Розроблено UML діаграму варіантів використання та діаграму потоків даних (DFD). Створено макети основних елементів вебзастосунку. Спроектовано базу даних та складено її логічну та функціональну схеми.

Реалізовано інструменти взаємодії користувача з вебзастосунком, які надають йому можливість обирати товари та оформляти замовлення. Для адміністрування вебзастосунку розроблено панель адміністратора, яка дає змогу керувати замовленнями сайту.

Для підвищення функціональності та ефективності вебзастосунку для ведення замовлень на виготовлення сільськогосподарського обладнання запропоновано такі можливості розширення:

- чат-підтримка в реальному часі дозволить користувачу напряму звернутися до менеджера з питанням. Ця функція підвищить рівень обслуговування клієнтів;

- система відгуків та рейтингів дозволить користувачам залишати відгук про товар та оцінювати його. Користувачі, які зайшли на сайт вперше зможуть оцінити якість обладнання та обслуговування;

- інтегрування з системами оплати через електронний гаманець зробить оплату зручною та безпечною. Існує велика кількість платіжних систем, наприклад, Apple Pay, Google Pay, PayPal, та інші;

- панель аналітики для адміністратора показуватиме статистику продажів, популярність товарів.

Ці розширення допоможуть підвищити якість обслуговування, забезпечать зручність і безпеку платежів, зроблять бізнес поінформованим і прозорим за допомогою аналітики.

Результатом кваліфікаційної роботи є розроблений вебзастосунок для ведення замовлень на виготовлення сільськогосподарського обладнання для підприємства ТОВ «ДЗСТ 2».

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бутенко Т. А., Сирий В. М. Інформаційні системи та технології : навч. посіб. Харків : ХНАУ ім. В.В. Докучає., 2020. 207 с. URL: https://repo.btu.kharkov.ua/bitstream/123456789/4849/1/INFO_SYSTEMS_20.pdf (дата звернення: 04.02.2024).
2. Інтернет-магазин AGROZIP. *Інтернет-магазин AGROZIP*. URL: <https://agrozip.com/> (дата звернення: 09.02.2024).
3. Олександр агро. *ОЛЕАНДР АГРО*. URL: <https://oagro.com.ua/> (дата звернення: 08.02.2024).
4. Опендатабот. *Опендатабот*. URL: <https://opendatabot.ua/c/44765929> (дата звернення: 02.02.2024).
5. ТОВАРИСТВО З ОБМЕЖЕНОЮ ВІДПОВІДАЛЬНІСТЮ ДЗСТ 2: вся інформація про компанію - 44765929, дата реєстрації, власники. VKURSI.PRO. *Vkursi*. URL: <https://vkursi.pro/card/tov-dzst-2-44765929> (дата звернення: 03.02.2024).
6. Brackets - contribute. *Brackets - A modern, open source code editor that understands web design*. URL: <https://brackets.io/contribute.html> (дата звернення: 23.02.2024).
7. Creately. What is an entity relationship diagram (ERD)? | an introduction to ER diagram. *Visual Collaboration & Diagramming Platform*. URL: <https://creately.com/guides/er-diagrams-tutorial/> (дата звернення: 20.02.2024).
8. Ghate S. HTTP made easy: understanding the web client-server communication. *HackerNoon - read, write and learn about any technology*. URL: <https://hackernoon.com/http-made-easy-understanding-the-web-client-server-communication-yz783vg3> (дата звернення: 03.02.2024).
9. HTML документ. Структура html документа. Що тестувати?. *Онлайн-курси від компанії QATestLab. Головна сторінка*. URL: <https://training.qatestlab.com/blog/technical-articles/html-document-structure-what-to-test/> (дата звернення: 20.02.2024).

10. Rational software architect 9.6.1. *IBM - United States*. URL: <https://www.ibm.com/docs/en/rational-soft-arch/9.6.1?topic=diagrams-use-case> (дата звернення: 15.02.2024).

11. Tereshchenko E. The development of e-commerce in the modern business environment. *Efektivna ekonomika*. 2019. № 9. URL: <https://doi.org/10.32702/2307-2105-2019.9.53> (дата звернення: 03.02.2024).

12. Types and components of data flow diagram (DFD) - geeksforgeeks. *GeeksforGeeks*. URL: https://www.geeksforgeeks.org/types-and-components-of-data-flow-diagram-dfd/?ref=ml_lbp (дата звернення: 16.02.2024).

13. What is a logical data model? Definition and examples. *Datamation*. URL: <https://www.datamation.com/big-data/logical-data-model/> (дата звернення: 21.02.2024).

14. What is CSS and how does it work?. *Hostinger Tutorials*. URL: <https://www.hostinger.com/tutorials/what-is-css> (дата звернення: 18.02.2024).

15. What is JavaScript? - JavaScript (JS) Explained - AWS. *Amazon Web Services, Inc.* URL: https://aws.amazon.com/what-is/javascript/?nc1=h_ls (дата звернення: 18.02.2024).

16. What is MySQL?. *Oracle. Cloud Applications and Cloud Platform*. URL: <https://www.oracle.com/sa/mysql/what-is-mysql/> (дата звернення: 19.02.2024).

17. What is PHP hypertext preprocessor. *Web Design Latest News*. URL: <https://web-design-eastbourne.co.uk/News/what-is-php-the-basics/> (дата звернення: 20.02.2024).

18. What is phpmyadmin: tutorial, features, installation - javatpoint. *www.javatpoint.com*. URL: <https://www.javatpoint.com/phpmyadmin> (дата звернення: 29.02.2024).

ДОДАТКИ

ДОДАТОК А

ТЕХНІЧНЕ ЗАВДАННЯ

**На розробку вебзастосунку для ведення замовлень на виготовлення
сільсько-господарського обладнання**

Погоджено:

Кандидат технічних наук, доцент кафедри
кібернетики та інформатики

Олександр В'ЮНЕНКО

Суми 2024

1. Призначення й мета створення вебзастосунку

1.1 Призначення вебзастосунку

Вебзастосунок призначений для ведення замовлень на виготовлення сільськогосподарського обладнання. Основною функцією вебзастосунку є надання користувачам можливості створювати, відстежувати та керувати замовленнями на виготовлення різного типу сільськогосподарського обладнання.

1.2 Мета створення вебзастосунку

Мета створення вебзастосунку полягає в автоматизації процесів обробки замовлень, покращенні управління виробничими процесами, забезпеченні прозорості та підвищенні ефективності взаємодії між замовниками та виробниками.

1.3 Цільова аудиторія

У цільовій аудиторії вебзастосунку можна виділити таку групу:

- Фермери та аграрні підприємства, які потребують сільськогосподарське обладнання;
- Виробники сільськогосподарського обладнання;
- Постачальники комплектуючих для сільськогосподарського обладнання;

2. Вимоги до вебзастосунку

2.1 Вимоги до вебзастосунку в цілому

2.1.1 Вимоги до структури й функціонування вебзастосунку

- Авторизація користувачів;
- Створення замовлень;
- Відстеження замовлень;
- Управління замовленнями.

2.1.2 Вимоги до персоналу

Адміністратори повинні мати доступ до функцій додавати та видаляти товари та контролювати замовлення.

2.1.3 Вимоги до збереження вебзастосунку

Використання MySQL для зберігання та управління інформацією, забезпечення ефективного збору, зберігання та обробки даних.

2.1.4 Вимоги до розмежування доступу

Доступ до вебзастосунку обмежений для користувачів, які не є в базі даних; оформити замовлення можуть лише ті, хто зареєстрований.

2.2 Структура вебзастосунку

2.2.1 Навігація

Головна сторінка вебзастосунку містить меню навігації, яке розташоване у середній частині сторінки і містить посилання на основні розділи і функції програми.

2.2.3 Наповнення вебзастосунку (контент)

Включає сторінки каталогу товарів, доставки та оплати, інформації про підприємство, зв'язку та

2.3 Вимоги до функціонування системи

2.3.1 Потреби користувача

Забезпечення зручного інтерфейсу для перегляду товару та оформлення замовлення, відстеження його статусу їх виконання.

2.3.2 Функціональні вимоги

Забезпечення реєстрації та авторизації користувачів, створення, редагування і видалення замовлень, перегляд каталогу продукції з описом товарів, автоматичне розрахування вартості замовлення з урахуванням параметрів продукції.

2.3.3 Системні вимоги

Системні вимоги до потреб користувачів-покупців:

- наявність авторизації;
- додавання та видалення товарів з кошика;
- оформлення замовлення.

Системні вимоги до потреб адміністратора:

- авторизація;
- додавання та видалення товарів зі списку продукції в адмінпанелі;
- зміна статусу замовлення.

2.4 Вимоги до видів забезпечення

2.4.1 Вимоги до інформаційного забезпечення

Реалізація застосунку відбувається з використанням:

- PHP;
- MySQL;
- CSS;
- HTML;
- JS.

2.4.2 Вимоги до лінгвістичного забезпечення

Вебзастосунок має бути виконаний українською мовою.

2.4.3 Вимоги до програмного забезпечення

Для розробки вебзастосунку використовуються такі програмні компоненти та технології:

HTML, CSS, JavaScript використовуються для створення інтерфейсу користувача та забезпечення інтерактивності.

PHP - мова програмування на стороні сервера, яка використовується для обробки даних і взаємодії з базою даних.

MySQL - система керування базами даних, яка забезпечує зберігання та управління інформацією про користувачів та замовлення.

3 Склад і зміст робіт зі створення застосунку

Докладний опис етапів роботи зі створення вебзастосунку наведено в таблиці А.1.

Таблиця А.1 - Етапи створення вебзастосунку

№	Етапи розробки	Терміни розробки (дні)
1	2	3
1	Зустрічі з клієнтом для визначення функціональних і технічних вимог.	1
2	Постановка задачі. Формулювання основних задач, які мають бути вирішені.	4
3	Розробка договору на створення вебзастосунку для ведення замовлень.	1
4	Розробка технічного завдання на створення вебзастосунку для ведення замовлень.	1
5	Узгодження договору на створення вебзастосунку для ведення замовлень.	1
6	Узгодження технічного завдання на створення вебзастосунку для ведення замовлень.	1
7	Аналіз конкурентів для визначення переваг та недоліків їхніх рішень.	4
8	Вивчення потреб цільової аудиторії та їхні вимоги до продукту.	4
9	Розробка структури бази даних для зберігання інформації про замовлення.	6
10	Підготовка контенту, який буде розміщений в підсистемі ведення замовлень	4
11	Створення макетів інтерфейсу користувача для основних функцій застосунку.	6
12	Розробка архітектури застосунку, визначивши компоненти та їх взаємодію.	6
13	Вибір основної палітри кольорів підсистеми.	1
14	Вибір типу і розміру основних шрифтів підсистеми.	1
15	Створення візуальних компонентів(кнопки, поля введення, меню і т.д.).	6
16	Розробка дизайну сторінок з урахуванням вимог.	6
17	Додавання динамічних елементів та анімації на вебзастосунок.	15
18	Використання CSS та JavaScript для створення рухомих елементів та переходів між сторінками.	15
19	Перевірка функціональності різних елементів та сторінок вебзастосунку.	2
20	Створення моделей даних для представлення об'єктів системи.	3
21	Написання коду для обробки бізнес-логіки застосунку, таких як обробка замовлень, розрахунок вартості і т.д.	20
22	Написання API для взаємодії фронтенду та бази даних.	5
23	Підключення бази даних і налаштування її взаємодії з серверною частиною.	1
24	Наповнення сайту контентом.	2

Продовження таблиці А.1

1	2	3
25	Проведення тестування компонентів застосунку.	3
26	Виправлення виявлених під час тестування помилок.	4
27	Вдосконалення роботи застосунку на основі отриманих під час тестування результатів.	4
28	Здача вебзастосунку для ведення замовлень замовнику для перевірки.	1
29	Усунення зауважень замовника.	3
30	Інструктаж замовника з роботи із сайтом.	1
	Загальна тривалість робіт	132

Джерело: запропоновано автором

4. Вимоги до складу й змісту робіт із введення вебзастосунку в експлуатацію

Для того, щоб вебзастосунком почали користуватися користувачі необхідно розмістити його у мережі Інтернет, тому необхідно придбати доменне ім'я та місце на хостингу. На хостинг переноситься вебзастосунок і наповнення бази даних з подальшою їх доробкою.

ДОДАТОК Б

ПЛАНУВАННЯ РОБІТ

Впровадження діаграми Ганта в процес розробки веб-додатку для управління замовленнями на виготовлення сільгосптехніки дозволяє забезпечити чітку візуалізацію проєкту, ефективно координувати дії, визначати критичні завдання, відстежувати прогрес, оптимально розподіляти ресурси та своєчасно аналізувати ризики.

Щоб використовувати діаграму Ганта найефективніше, необхідно дотримуватися певних правил. По-перше, необхідно визначити всі завдання, які мають бути виконані, та їхню тривалість. Потім необхідно виявити залежності між завданнями і визначити порядок їх виконання. Після цього можна створити діаграму Ганта для візуалізації процесу проєкту та відстеження прогресу.

Діаграма Ганта зображена на рисунку Б.1.

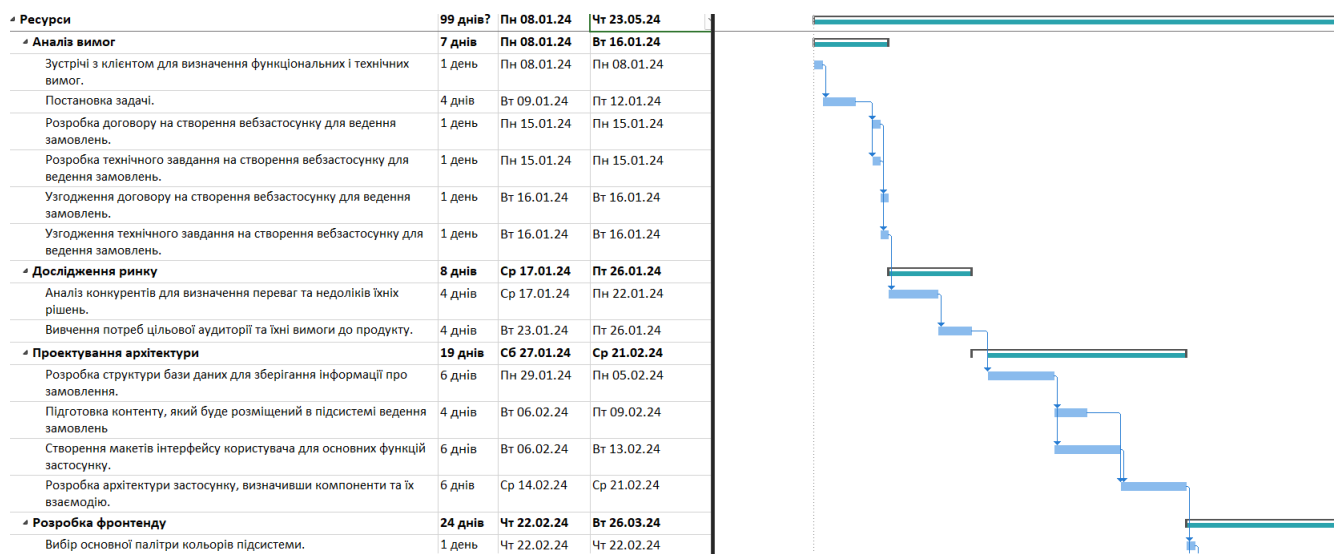


Рисунок Б.1 – Діаграма Ганта

Джерело: запропоновано автором

Діаграма Ганта є важливим інструментом управління проєктами, який надає велику користь в розумінні послідовності робіт, визначенні тривалості завдань та контролі прогресу проєкту.

ДОДАТОК В
КОПІЇ ПУБЛІКАЦІЙ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ НАЦІОНАЛЬНИЙ АГРАРНИЙ УНІВЕРСИТЕТ

МАТЕРІАЛИ
ВСЕУКРАЇНСЬКОЇ НАУКОВОЇ
КОНФЕРЕНЦІЇ СТУДЕНТІВ
ТА АСПІРАНТІВ, ПРИСВЯЧЕНОЇ
МІЖНАРОДНОМУ ДНЮ СТУДЕНТА

(13 - 17 листопада 2023 р., м. Суми)

Неледва А.О. МОТИВАЦІЯ ПЕРСОНАЛУ В УМОВАХ ЗРОСТАННЯ ВПЛИВУ ШТУЧНОГО ІНТЕЛЕКТУ	313
Новіков Д.С. ДОСЛІДЖЕННЯ ЕКОНОМІЧНИХ ПЕРЕДУМОВ ТА ДОЦІЛЬНОСТІ ВИКОРИСТАННЯ АГРОДРОНІВ.....	314
Пігуль О.В. МЕТОДОЛОГІЧНІ ЗАСАДИ ОЦІНКИ СТАЛОСТІ ЖИТЛОВИХ ПРОЄКТІВ	315
Поздирко В.С. НАПРЯМИ ПІДВИЩЕННЯ ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ГАЛУЗІ ЗЕРНОВИРОБНИЦТВА	316
Поздирко В.С. НАПРЯМИ РОЗВИТКУ ГАЛУЗІ РОСЛИННИЦТВА	317
Сьомушкін В.О. УПРАВЛІННЯ ЖИТТЄВИМ ЦИКЛОМ БУДІВЕЛЬНОЇ ОРГАНІЗАЦІЇ ЯК ІНСТРУМЕНТ СТАЛОГО РОЗВИТКУ	318
Щенякін Д.О. ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ ВИКОРИСТАННЯ СОНЯЧНИХ ПАНЕЛЕЙ НА НАУКОВО-ДОСЛІДНОМУ ПІДПРИЄМСТВІ	319
Белоконь П.А. СУЧАСНІ АВТОМАТИЗОВАНІ ІНФОРМАЦІЙНІ СИСТЕМИ РЕКРУТИНГУ	320
Стебляно В.В. АНАЛІТИЧНИЙ ОГЛЯД ІТ-ТЕХНОЛОГІЙ ЩОДО ОБЛІКУ ВИТРАТ НА ВИРОБНИЦТВО СІЛЬСЬКО-ГОСПОДАРСЬКОЇ ПРОДУКЦІЇ. ПОСТАНОВКА ЗАДАЧІ	321
Агеев О.А. ОГЛЯД ПЛАТФОРМ ЕЛЕКТРОННОГО НАВЧАННЯ.....	322
Блаженко Д.О. МЕТОДИ ТА ПРОБЛЕМИ ПРОЄКТУВАННЯ ІНФОРМАЦІЙНИХ ПІДСИСТЕМ ВЕДЕННЯ ЗАМОВЛЕНЬ ІЗ ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ TRIZ.....	323
Гончар Д.О. АВТОМАТИЗАЦІЯ РОЗРАХУНКУ БАЛАНСУ ГУМУСУ ТА ПОЖИВНИХ РЕЧОВИН В ҐРУНТІ.....	324
Доля А.Р. СУЧАСНІ ТЕХНОЛОГІЇ РОЗРОБКИ АДМІНІСТРАТИВНИХ МОДУЛІВ.....	325
Дубінчин В.С. РОЗРОБКА ІНФОРМАЦІЙНОЇ ПІДСИСТЕМИ ОЦІНЮВАННЯ ПРАЦЕЗДАТНОСТІ КОМП'ЮТЕРИЗОВАНИХ РОБОЧИХ МІСЦЬ	326
Крамський А.С., Бусенко І.В. ПРОГНОЗУВАННЯ КІЛЬКОСТІ СТУДЕНТІВ В УКРАЇНІ З ВИКОРИСТАННЯМ ЕКОНОМЕТРИЧНИХ МЕТОДІВ	327
Олійник А.В. ПОСТАНОВКА ЗАДАЧІ ТА ПРОЄКТУВАННЯ ПІДСИСТЕМИ ОБЛІКУ ПАЛИВНО-МАСТИЛЬНИХ МАТЕРІАЛІВ В ТЕРИТОРІАЛЬНІЙ ГРОМАДІ.....	328
Руденок О.А. ПОШУК ІНФОРМАЦІЇ В ТЕКСТОВИХ ДОКУМЕНТАХ З УРАХУВАННЯМ КОНТЕКСТУ.....	329
Санжаков А. А. АНАЛІТИЧНИЙ ОГЛЯД ВИКОРИСТАННЯ ІТ- ТЕХНОЛОГІЙ У МОБІЛЬНІЙ РОЗРОБЦІ ДЛЯ ОС АНДРОЇД	330
Сіденко А.С. ПРОБЛЕМИ РЕГІСТРАЦІЇ ТА ВІЗУАЛІЗАЦІЇ ДАНИХ МОНІТОРИНГУ НАВКОЛИШНЬОГО СЕРЕДОВИЩА.....	331
Терещенко С.С. ПОСТАНОВКА ЗАДАЧІ ТА ПРОЄКТУВАННЯ ПІДСИСТЕМИ АВТОМАТИЗАЦІЇ ОБЛІКОВИХ ПРОЦЕСІВ В АТЕЛЬє З ВИГОТОВЛЕННЯ ОДЯГУ	332
Чубун М.П. СУЧАСНІ ТЕНДЕНЦІЇ У СТВОРЕННІ ВЕБ-САЙТІВ ДЛЯ ІНТЕРНЕТ-МАГАЗИНІВ КОМП'ЮТЕРИЗОВАНИХ РОБОЧИХ МІСЦЬ	333
Шелковський С.В. СУЧАСНІ ТЕНДЕНЦІЇ ПРОЄКТУВАННЯ ПІДСИСТЕМИ ОБЛІКУ ПІДПРИЄМСТВ ПЕРЕРОБКИ СІЛЬСЬКОГОСПОДАРСЬКОЇ ПРОДУКЦІЇ	334
Горбань С.В. ТЕОРЕТИЧНІ АСПЕКТИ ФОРМУВАННЯ ПОЗИТИВНОГО ІМІДЖУ ПІДПРИЄМСТВА.....	335
Miao Yin Hai INFLUENCE OF MANAGEMENT DECISIONS ON THE ENTERPRISE'S INNOVATION STRATEGY.....	336
Zhao Min THE EFFICIENT MANAGEMENT OF TALENTS FOR THE DEVELOPMENT OF ENTERPRISE.....	337
Заболотна А.І. УПРАВЛІННЯ ПРОЦЕСАМИ ПІДВИЩЕННЯ ЯКОСТІ ЖИТТЯ РЕГІОНУ ЗА МАТЕРІАЛАМИ ДРУЖБІВСЬКОЇ ТГ	338
Глек О.А. ОРГАНІЗАЦІЯ КЕЙТЕРИНГОВОГО ОБСЛУГОВУВАННЯ	339
Малюк Н.М. СТАЛИЙ РОЗВИТОК ЛІСОВОГО ГОСПОДАРСТВА.....	340
Дутов М.М. СТІЙКИЙ РОЗВИТОК ПІДПРИЄМСТВА.....	341
Бондар В.М. ТЕОРЕТИЧНІ ОСНОВИ КОНКУРЕНТОСПРОМОЖНОСТІ ДІЯЛЬНОСТІ ПІДПРИЄМСТВ.....	342
Макарчук П.В. ОРГАНІЗАЦІЙНА КУЛЬТУРА ЯК ОБ'ЄКТ УПРАВЛІНСЬКОЇ ДІЯЛЬНОСТІ.....	343
Могиленець В.М. ТЕОРЕТИЧНІ ОСНОВИ УПРАВЛІННЯ ПЕРСОНАЛОМ В СИСТЕМІ МЕНЕДЖМЕНТУ ПІДПРИЄМСТВА.....	344
Борцов А.О. ЗАСТОСУВАННЯ ДІАГНОСТИЧНИХ ЗАСОБІВ У ЛОГІСТИЦІ ТА ЇХ ПОШИРЕННЯ НА СИСТЕМІ ТРАНСПОРТУ	345
Гризодуб Д.Г. АСПЕКТИ УПРАВЛІННЯ ЯКІСТЮ ПОСЛУГ В СУЧАСНИХ УМОВАХ.....	346

МЕТОДИ ТА ПРОБЛЕМИ ПРОЕКТУВАННЯ ІНФОРМАЦІЙНИХ ПІДСИСТЕМ ВЕДЕННЯ ЗАМОВЛЕНЬ ІЗ ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ TRIZ

Блаженко Д.О., студ. 2 с.т. курсу ФЕiМ, спец. «Інформаційні системи та технології»

Науковий керівник: доцент О.Б.В'юнченко

Сумський НАУ

В сучасному бізнесі ефективне управління замовленнями є критично важливим аспектом. Інформаційні підсистеми для управління замовленнями допомагають підприємствам автоматизувати процеси, зменшити помилки та покращити якість обслуговування клієнтів. Проте, їх проектування і реалізація стикаються з численними проблемами та протиріччями. Серед основних проблем проектування інформаційних підсистем для управління замовленнями виділяють наступні:

1. Протиріччя між ефективністю та витратами ресурсів, тобто підвищення ефективності обробки замовлень може вимагати великих витрат на розробку та впровадження інформаційної підсистеми.

2. Протиріччя між мінімізацією помилок та швидкістю обробки замовлень, тобто зменшення помилок у управлінні замовленнями може призвести до збільшення часу обробки.

Використання TRIZ-підходів може допомогти знайти баланс між ефективністю та витратами, а також між точністю та швидкістю. Для успішного проектування і реалізації інформаційних підсистем важливо постійно вдосконалювати їх та забезпечувати співпрацю між різними відділами підприємства. TRIZ - це методологія, яка допомагає інженерам та дизайнерам знаходити інноваційні рішення на основі визначених принципів та інструментів. Також вона може бути корисною при проектуванні інформаційних підсистем для управління замовленнями. Так принцип "Інше робоче середовище" передбачає, що замість ручної обробки замовлень, можливо використовувати автоматизовані системи для прискорення процесу та зменшення витрат, а принцип "Перетворення інших ресурсів" передбачає застосування технологій машинного навчання та штучного інтелекту, що може допомогти виявляти та виправляти помилки, що також дозволяє покращити якість обробки замовлень.

Для розв'язання проблем, пов'язаних з проектуванням інформаційних підсистем для управління замовленнями, пропонуються наступні рекомендації:

1. Аналіз і оптимізація процесів. Перед тим, як розпочати проектування інформаційної підсистеми, важливо провести аналіз і оптимізацію існуючих процесів управління замовленнями. Це дозволить виявити потреби та проблеми, які повинні бути враховані в проектуванні.

2. Впровадження TRIZ. Використання TRIZ-підходів може допомогти знайти нові та інноваційні рішення для проектування інформаційних підсистем. Ці принципи можуть бути застосовані до різних аспектів системи, включаючи оптимізацію процесів та зменшення витрат.

3. Співпраця між відділами. Важливо забезпечити ефективну співпрацю між відділами, які взаємодіють з інформаційною підсистемою. Це допоможе уникнути конфліктів та забезпечити зручний обмін інформацією.

4. Постійне вдосконалення. Проектування інформаційних підсистем - це постійний процес, який вимагає постійного вдосконалення та оновлення. Важливо слідувати за новими технологіями та підходами і постійно навчатися на досвіді інших підприємств.

Проектування інформаційних підсистем для управління замовленнями - це складний процес, який вимагає аналізу, інновацій та широкої співпраці. Використання методології TRIZ дозволяє вирішити протиріччя, які виникають під час проектування, і знайти оптимальні управлінські рішення. Постійне вдосконалення та співпраця між окремими відділами підприємства допоможуть забезпечити успішну реалізацію інформаційних підсистем та покращити управління замовленнями.

Незважаючи на важливість та потенціал інформаційних підсистем для управління замовленнями, також існують інші напрямки для подальших досліджень, які можуть покращити розуміння та вдосконалення цього аспекту бізнесу:

1. Використання розширених технологій. Дослідження може бути спрямоване на застосування розширених технологій, таких як розпізнавання образів та обробка природної мови, для покращення точності та швидкості обробки замовлень.

2. Вплив на користувача. Дослідження може досліджувати вплив інформаційних підсистем на користувачів, включаючи способи навчання та підтримки користувачів, які допомагають підвищити ефективність обслуговування та задоволеність клієнтів.

3. Безпека та конфіденційність. Врахування аспектів безпеки та конфіденційності даних є важливим для контролю інформаційних підсистем, і подальше дослідження може спрямовуватися на розробку методів та технологій, що забезпечують надійний захист інформації.

Проектування інформаційних підсистем для управління замовленнями - це важливий аспект сучасного бізнесу, який може покращити ефективність та якість обслуговування клієнтів. Знання проблем та методів їх розв'язання є важливими для досягнення успіху в цій галузі. Досягнення балансу між ефективністю, точністю та витратами вимагає уважності та інноваційних підходів, а дослідження та співпраця між спеціалістами можуть сприяти подальшому вдосконаленню цих інформаційних підсистем та управлінню замовленнями підприємства в цілому.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**СУМСЬКИЙ НАЦІОНАЛЬНИЙ
АГРАРНИЙ УНІВЕРСИТЕТ**

МАТЕРІАЛИ

науково-практичної конференції
викладачів, аспірантів та студентів
Сумського НАУ

(14-16 травня 2024 р.)

Luo Hao. A STUDY ON CONSUMER BEHAVIOUR MANAGEMENT THROUGH ADVANCED MARKETING TOOLS	414
Wang Baichao. MANAGEMENT CRISIS IN AGRICULTURAL COMPANIES IN CURRENT SITUATIONS	415
Wang Ruijie. HUMAN RESOURCES MANAGEMENT OPTIMIZATION AND INNOVATION BASED ON ARTIFICIAL INTELLIGENCE	416
Yu Hang. RESEARCH ON THE INTRODUCTION OF MODERN MANAGEMENT TECHNOLOGY INTO ENTERPRISES	417
Zhou Kunming. PECULIARITIES OF MANAGING HIGH-TECH ENTERPRISES IN A TRANSPARENT ECONOMY	418
Бусенко І.В. АНАЛІЗ ОСНОВНИХ РІЗНОВИДІВ SQL-ІН'ЄКЦІЙ, АНАЛІТИКА ЗАСТОСУВАННЯ ТА СПОСОБИ ЗАХИСТУ ВІД НИХ	419
Белоконь П.А. РОЗРОБКА ІНФОРМАЦІЙНОЇ ПІДСИСТЕМИ ПІДТРИМКИ РОБОТИ HR-МЕНЕДЖЕРА	420
Стеблянюк В. В. РОЗРОБКА ВЕБ-ЗАСТОСУНКУ ОБЛІКУ ВИТРАТ НА ВИРОБНИЦТВО ПРОДУКЦІЇ. РЕАЛІЗАЦІЯ ПРОЄКТУ	421
Фідірко Т.Р. РОЗРОБКА ВЕБЗАСТОСУНКУ ОБЛІКУ ТЕХНІЧНОГО СТАНУ ЛЕГКОВИХ АВТОМОБІЛІВ. ПОСТАНОВКА ЗАДАЧІ	422
Фідірко Т.Р. РОЗРОБКА ВЕБЗАСТОСУНКУ ОБЛІКУ ТЕХНІЧНОГО СТАНУ ЛЕГКОВИХ АВТОМОБІЛІВ. РЕАЛІЗАЦІЯ ПРОЄКТУ	423
Биченко Є.В. РОЗРОБКА ПРОТОТИПУ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ ІНТЕРФЕЙСУ КОРИСТУВАЧА ВЕБОРІЄНТОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПІДТРИМКИ УПРАВЛІННЯ ДІЯЛЬНІСТЮ ПІДПРИЄМСТВА	424
Корецький В.А. ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ВЕБІНФОРМАЦІЙНОЇ ПІДСИСТЕМИ МОНІТОРИНГУ ІНТЕРНЕТ-ТРАФІКУ	425
Лалченко І. В. ТА РЕАЛІЗАЦІЯ ВЕБЗАСТОСУНКУ ОРГАНІЗАЦІЇ ДИСТАНЦІЙНИХ КУРСІВ НЕФОРМАЛЬНОЇ ОСВІТИ	426
Хоменко Р.М. ПРОЄКТУВАННЯ ІНТЕРФЕЙСУ КОРИСТУВАЧА ЧАТ-БОТА ДЛЯ ЗРУЧНИХ ТА ЕФЕКТИВНИХ ОНЛАЙН-ЗАМОВЛЕНЬ ЕЛЕКТРОНІКИ	427
Блаженко Д. О. ПРОБЛЕМИ ПРОЄКТУВАННЯ БАЗИ ДАНИХ ПІДСИСТЕМИ ВЕДЕННЯ ЗАМОВЛЕНЬ ДЛЯ ВИГОТОВЛЕННЯ СІЛЬСЬКОГОСПОДАРСЬКОЇ ПРОДУКЦІЇ	428
Ковальов Р. С. ПРАКТИЧНЕ ПРОГРАМУВАННЯ ГЕНЕРАТОРІВ 3D КОНТЕНТУ НА ПРИКЛАДІ ГЕНЕРАТОРУ МЕША РЕЛЬЄФУ МІСЦЕВОСТІ	429
Ковальов Р. С. ПРАКТИЧНЕ ПРОГРАМУВАННЯ ГЕНЕРАТОРІВ 3D КОНТЕНТУ НА ПРИКЛАДІ ГЕНЕРАТОРУ МЕША РЕЛЬЄФУ МІСЦЕВОСТІ	430
Руденко О. А. МІНІМІЗАЦІЯ ЗАТРИМОК У ВЕБДОДАТКАХ ВИКОРИСТОВУЮЧИ ГРАНИЧНІ ТЕХНОЛОГІЇ	431
Сіденко А. С. ПРОБЛЕМИ РЕАЛІЗАЦІЇ ПІДСИСТЕМИ ОБЛІКУ ПРИРОДООХОРОНИ ЗАХОДІВ	432
Терещенко С.С. АВТОМАТИЗАЦІЯ ОБЛІКОВИХ ПРОЦЕСІВ: КЛЮЧ ДО ЕФЕКТИВНОГО УПРАВЛІННЯ В АТЕЛЬЄ З ВИГОТОВЛЕННЯ ОДЯГУ	433
Михайлюк О.С. ОГРУНТУВАННЯ НЕОБХІДНОСТІ РОЗРОБКИ WEB-ОРІЄНТОВАНОЇ ПІДСИСТЕМИ ПІДТРИМКИ ОБЛІКОВИХ ПРОЦЕСІВ РЕГІОНАЛЬНОГО ІНТЕРНЕТ-ПРОВАЙДЕРА	434
Олійник А.В. ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПІДСИСТЕМИ ОБЛІКУ ПАЛИВНО-МАСТИЛЬНИХ МАТЕРІАЛІВ В ТЕРИТОРІАЛЬНІЙ ГРОМАДІ	435
Чубун М.П. АНАЛІЗ ТА ВИЗНАЧЕННЯ ВИМОГ ДО СУЧАСНОГО ВЕБ-САЙТУ	436
Дубінін В. С. ВИЗНАЧЕННЯ ФУНКЦІОНАЛЬНИХ ВИМОГ ДО ІНФОРМАЦІЙНОЇ ПІДСИСТЕМИ ОЦІНЮВАННЯ ПРАЦЕЗДАТНОСТІ КОМП'ЮТЕРИЗОВАНИХ РОБОЧИХ МІСЦЬ	437
Санжаков А. А. НОВИЙ ПІДХІД РОЗРОБКИ ДИЗАЙНУ ВЕБ-ЗАСТОСУНКІВ – MATERIAL 3	438
Федірко Д.Д. ПРОБЛЕМИ ПРОЄКТУВАННЯ ВЕБ-ЗАСТОСУНКІВ ОЦІНКИ ТАРИФНИХ ПРОПОЗИЦІЙ МОБІЛЬНИХ ОПЕРАТОРІВ УКРАЇНИ	439
Федірко Д.Д. РОЗРОБКА ВЕБ-ЗАСТОСУНКУ ДЛЯ АНАЛІЗУ МОБІЛЬНИХ ОПЕРАТОРІВ УКРАЇНИ	440
Маслак О. М. ДЕРЖАВНЕ РЕГУЛЮВАННЯ ПРОЦЕСІВ АГРОПРОМИСЛОВОГО ВИРОБНИЦТВА НА РЕГІОНАЛЬНОМУ РІВНІ (НА ПРИКЛАДІ СУМСЬКОЇ ОБЛАСТІ)	441
Горбань С.В. ІМІДЖ ПІДПРИЄМСТВА: ВПЛИВ, ФОРМУВАННЯ ТА ОЦІНКА	442
Заболотна А.І. НОРМАТИВНО-ПРАВОВЕ ЗАБЕЗПЕЧЕННЯ ЕКОНОМІЧНОГО РОЗВИТКУ ДРУЖІВСЬКОЇ ТЕРИТОРІАЛЬНОЇ ГРОМАДИ	443
Гулініна О.Ю. ШЛЯХИ ОПТИМІЗАЦІЇ ОРГАНІЗАЦІЙНОЇ СТРУКТУРИ УПРАВЛІННЯ ЗАКЛАДУ ОХОРОНИ ЗДОРОВ'Я	444
Сущенко В. О. ПРОЦЕС РОЗВИТКУ ПЕРСОНАЛУ, ЯК ЧИННИК КОНКУРЕНТОЗДАТНОСТІ ПІДПРИЄМСТВА В РИНКОВИХ УМОВАХ	445
Каплун В.М. УПРАВЛІННЯ СІЛЬСЬКОГОСПОДАРСЬКИХ ПІДПРИЄМСТВ ІЗ ЗАСТОСУВАННЯ ПРОГРАМНОГО КОНТРОЛЮ І МОНІТОРИНГУ	446
Бондар А.В. ТЕОРЕТИКО-МЕТОДИЧНІ ПІДХОДИ ДО ВПРОВАДЖЕННЯ ТА ОЦІНКИ СТРАТЕГІЧНОГО УПРАВЛІННЯ РОЗВИТКОМ СІЛЬСЬКОГОСПОДАРСЬКИХ ПІДПРИЄМСТВ	447
Романенко Т.А. СУЧАСНІ НАПРЯМИ ІНТЕНСИФІКАЦІЇ ВИРОБНИЦТВА ПРОДУКЦІЇ РОСЛИННИЦТВА В КОНТЕКСТІ ПРОДОВОЛЬНОЇ БЕЗПЕКИ	448
Мішенін Є. В., Ярова І. Є. ОРГАНІЗАЦІЙНО ЕКОНОМІЧНИЙ МЕХАНІЗМ ЗАБЕЗПЕЧЕННЯ СТАЛОГО ПРОСТОРОВОГО РОЗВИТКУ ЛІСОВОГО ГОСПОДАРСТВА	449
Hu Guanlan CRISIS MANAGEMENT THEORY	450
Zhang Liming HOW DOES BYD COMPANY IMPROVE ITS COMPETITIVENESS?	451
Zhao Jing THE INFLUENCE OF CORPORATE CULTURE ON EMPLOYEE SATISFACTION	452
Cui Xiaoping DEVELOPMENT PATH OF ENTERPRISE MANAGEMENT IN THE ERA OF ARTIFICIAL INTELLIGENCE	453

ПРОБЛЕМИ ПРОЕКТУВАННЯ БАЗИ ДАНИХ ПІДСИСТЕМИ ВЕДЕННЯ ЗАМОВЛЕНЬ ДЛЯ ВИГОТОВЛЕННЯ СІЛЬСЬКОГОСПОДАРСЬКОЇ ПРОДУКЦІЇ

Блаженко Д. О., студ. 2 с.т. курсу ФЕІМ, спец. «Інформаційні системи та технології»
 Науковий керівник: к.е.н., доц. О.Б. В'юненко
 Сумський НАУ

Підсистема управління замовленнями на підприємстві в сучасному бізнесі відіграє важливу роль у забезпеченні ефективного управління виробництвом та задоволенні потреб клієнтів. Проектування бази даних для цієї підсистеми є важливим етапом у визначенні структури, функціональності та ефективності системи управління підприємством.

Розробка бази даних підсистеми управління замовленнями є одним з основним компонентів інформаційної системи підприємства, при виконанні проектних робіт необхідно враховувати наступні проблеми формування інформаційної бази:

1. Надмірність даних – це дублювання інформації, яка міститься в базі даних;
2. Аномалії – порушення цілісності бази даних.

Розрізняють два види надмірності: просте(ненадлишкове) та надлишкове. При першій однакові дані дублюються в різних частинах бази даних. При надлишковій - інформація повторюється в різних форматах запису.

Аномалії можуть виникати як наслідок надмірності даних і існують такі види:

1. Аномалія видалення. Виникає при знищенні певних даних, яке призводить до втрати корисної та потрібної інформації.
2. Аномалія поновлення. Виникає при оновленні даних, яке призводить до некоректності та невідповідності між різними частинами бази даних.
3. Аномалія введення. Виникає, коли додавання нових даних стає неможливим через вимогу наявності додаткової інформації, яка відсутня або недоступна [1].

Для вирішення і передбачення зазначених проблем необхідно виконати нормалізацію створюваної бази даних. Це процес організації інформації у базі даних, який необхідний для зменшення дублювання та забезпечення цілісності та ефективності структури даних. Нормалізація дозволяє розбити вихідну таблицю на кілька таблиць, кожна з яких містить лише один тип даних і є атомарною. Це досягається шляхом поділу таблиці на менші підмножини та зв'язування їх за допомогою зовнішніх ключів. Результатом виконання таких операцій в SQL є усунення надлишкових (повторюваних) даних і забезпечення логічного зберігання даних. Даний підхід дозволяє покращити ефективність запитів до бази даних, спрощує вставку, оновлення та видалення даних, а також забезпечує оперативну підтримку та розширення бази даних з плином часу.

Використання процесу нормалізації має достатньо переваг, але також потрібно розуміти з якими недоліками доведеться стикнутись під час проектування бази даних, наприклад:

1. Підвищена складність. Великою кількістю таблиць із зовнішніми ключами важко керувати, що призводить до непорозумінь;
2. Зменшення гнучкості. Через правила нормалізації гнучкість зберігання даних, які не відповідають цим правилам, буде меншою;
3. Підвищені вимоги до сховища. Потрібно більше місця для зберігання додаткових таблиць та індексів.

Під час проектування бази даних підсистеми управління ведення замовлень потрібно звернути увагу і на інші рекомендації, аби забезпечити її цілісність, ефективність та надійність:

1. Аналіз вимог користувачів. Чітке розуміння потреб користувачів і вимог до бази даних дозволяє створювати структури даних, які відповідають потребам вашої системи.
2. Забезпечення безпеки даних. Використання механізмів аутентифікації, авторизації та шифрування для захисту конфіденційності та цілісності даних.
3. Резервне копіювання та відновлення. Регулярне створення резервних копій бази даних та розроблення процедури відновлення, щоб запобігти втраті даних у разі непередбачуваних ситуацій.

Проектування бази даних підсистеми управління замовленнями є важливим етапом при розробці інформаційної системи підприємства. Розуміти з чим доведеться стикнутись під час роботи і знати як вирішити ці проблеми є гарантією успішного проектування. Зосередження уваги на цих аспектах не тільки сприяє створенню добре структурованої та ефективної бази даних, але й закладає основу для успішної роботи та подальшого розвитку системи управління замовленнями.

Подальшим розвитком системи управління замовленнями є створення засобів аналізу даних та звітності, які допоможуть управлінню приймати обґрунтовані рішення на основі інформації про замовлення, попит та інші ключові метрики.

Література

1. DBMS normalization: an example of a 1NF, 2NF, 3NF database. URL: <https://www.guru99.com/uk/database-normalization.html> (дата звернення: 27.03.2023).

ДОДАТОК Г

ЛІСТИНГИ ПРОГРАМ

Лістинг основних файлів керування роботою додатку:

reg.php

```
<?php
session_start();

$email = ""; // Устанавливаем значение переменной $email до использования
if(isset($_SESSION['user']) && !empty($_SESSION['user'])) {
    header('Location: user.php');
    exit; // не забывайте выходить из скрипта после перенаправления
}

require_once 'php/connect.php';

if (!empty($_POST)) {
    $name = $_POST['name'];
    $surname = $_POST['surname'];
    $phone = $_POST['phone'];
    $email = $_POST['email'];
    $password = $_POST['password'];

    $check_user = mysqli_query($conn, "SELECT * FROM `user_table` WHERE
`mail` = '$email'");

    if (mysqli_num_rows($check_user) < 1) {
        $result = mysqli_query($conn, "INSERT INTO `user_table` (`id_user`,
`name`, `surname`, `phone`, `mail`, `password`) VALUES (NULL, '$name', '$surname',
'$phone', '$email', '$password')");

        header('Location: pass.php');
    } else {
        $_SESSION['message'] = $email . '<br> Така пошта вже
використовується!';
    }
}
```

```

}
?>
<!DOCTYPE html>
<html lang="ua">
<head>
    <meta charset="UTF-8">
    <title>Регістрація</title>
    <link rel="shortcut icon" href="img/icon/icon.png">
    <link rel="stylesheet" href="css/style.css">
    <link rel="stylesheet" href="css/header.css">
    <link rel="stylesheet" href="css/menu.css">
    <link rel="stylesheet" href="css/info.css">
    <link rel="stylesheet" href="css/product.css">
    <link rel="stylesheet" href="css/form.css">
    <link rel="stylesheet" href="css/footer.css">
</head>
<body>
    <div class="wrapper">
        <!--////////////////////////////////////header-->
        <header>
            <a href="index.php"></a>
            <div class="header_p">
                <p>Дубовицький</p>
                <p>завод</p>
                <p>сільгосптехніки</p>
            </div>
        </header>
        <!--////////////////////////////////////nav-->
        <div class="menu">

```



```

name="email" required>
<input type="email" maxlength="50"

name="password" id="password" required placeholder="Введи надійний пароль">
<input type="password" maxlength="20"

name="confirm_pass" id="confirm_pass" required placeholder="Повтори свій
пароль">
<input type="password" maxlength="20"

<p class="mes" style="color:red;">

<?php echo
isset($_SESSION['message']) ? $_SESSION['message'] : "";
unset($_SESSION['message']); ?>

</p><br>

<button type="submit" name="upload"
value="UPLOAD">Реєстрація</button>

</form>

<a href="pass.php">У вас уже є акаунт?</a>

<script>

const passwordInput =
document.getElementById('password');

const confirmPasswordInput =
document.getElementById('confirm_pass');

const submitButton =
document.querySelector('button[type="submit"]');

function checkPasswords() {
    if (passwordInput.value ===
confirmPasswordInput.value) {

        confirmPasswordInput.setCustomValidity("");

    } else {

        confirmPasswordInput.setCustomValidity('Не вірний пароль!');

```

```

        }
    }

    confirmPasswordInput.addEventListener('input', checkPasswords);
    passwordInput.addEventListener('input',
checkPasswords);

</script>
</div>
</div>
</div>
</div>
<!--//////////////////////////////////////footer-->
<footer id="footer">
    <div class="block_f">
        <p>    <a    href="">Facebook</a></p>
        <p><a    href="">Instagram</a></p>
        <p><a    href="">YouTube</a></p>
        <p><a    href="">Viber</a></p>
    </div>
    <div class="block_f">
        <p>(068) 094-2926 –
Бухгалтерія</p>
        <p>(068) 316-4932 –
Відділ збуту</p>
    </div>
    <div class="block_f">
        <p>(068) 31-64-932 -
Відділ по роботі з клієнтами</p>

```

<p>(098) 00-81-778 -
Відділ по роботі з клієнтами</p>

</div>

<div class="block_f">

<p>dzst@ukr.net</p>

</div>

</footer>

</div>

</body>

</html>

product.php

<?php

session_start();

require_once 'php/connect.php';

if (!\$_SESSION['user']) {

 header('Location: pass.php');

}

\$user = \$_SESSION['user']['id_user'];

// Перевірка, чи параметр 'data' присутній у GET-запиті

if (isset(\$_GET['data'])) {

 \$new_data = \$_GET['data'];

 // Запит до бази даних для отримання продукту

 \$result = mysqli_query(\$conn, "SELECT * FROM product_table WHERE
`id\_product` = '\$new_data'");

 \$row = mysqli_fetch_assoc(\$result);

 // Перевірка на кнопку 'buy'

 if (isset(\$_POST['add'])) {

 // Перевірка, чи сесійний користувач існує

 if (isset(\$_SESSION['user'])) {

```

$user = $_SESSION['user']['id_user'];

$result = mysqli_query($conn, "SELECT id_favorite, count FROM
favorites_table WHERE id_user = '$user' AND id_product = '$new_data'");

if ($row = mysqli_fetch_assoc($result)) {
    // запис із заданим id_product вже існує, оновлюємо кількість
    $new_count = $row['count'] + 1;
    $fav_id = $row['id_favorite'];
    $update = mysqli_query($conn, "UPDATE favorites_table SET count =
'$new_count' WHERE id_favorite = '$fav_id' AND id_user = '$user'");
    header('Location: user.php');
    exit;
} else {
    // Продукт не знайдено, додавання нового запису з кількістю 1
    $insert = mysqli_query($conn, "INSERT INTO `favorites_table`
(`id_favorite`, `id_product`, `id_user`, `count`, `status`) VALUES (NULL, '$new_data',
'$user', 1, 'selected')");
    header('Location: user.php');
    exit;
}
} else {
    header('Location: pass.php');
    exit;
}
}
} else {
    // Якщо параметр 'data' відсутній, можна перенаправити користувача або
    вивести повідомлення
    echo "Required data is missing!";
}
?>

```

```

<!DOCTYPE html>
<html lang="ua">
<head>
  <meta charset="UTF-8">
  <title>Продукт</title>
  <link rel="shortcut icon" href="img/icon/icon.png">
  <link rel="stylesheet" href="css/style.css">
  <link rel="stylesheet" href="css/header.css">
  <link rel="stylesheet" href="css/menu.css">
  <link rel="stylesheet" href="css/info.css">
  <link rel="stylesheet" href="css/product.css">
  <link rel="stylesheet" href="css/footer.css">
  <link rel="stylesheet" href="css/slider.css">
</head>
<body>
  <div class="wrapper">
    <!--//////////////////////////////////////header-->
    <header>
      <a href="index.php"></a>
      <div class="header_p">
        <p>Дубовицький</p>
        <p>завод</p>
        <p>сільгосптехніки</p>
      </div>
    </header>
    <!--//////////////////////////////////////nav-->
    <div class="menu">
      <div class="block_m"><a href="index.php">Головна</a></div>

```

```


<a href="serch.php">Товари</a></div>


```

```


<a href="delivery.php">Доставка та оплата</a></div>


```

```


<a href="description.php">Про нас</a></div>


```

```


<a href="communication.php">Зв'язок</a></div>


```

```


<a href="pass.php">Особистий кабінет</a></div>


```

```

</div>

```

```

<!--////////////////////////////////////content-->

```

```

<div class="content">

```

```

<div class="block">

```

```

<div class="info">

```

```

<br>

```

```

<div class="info_img">

```

```

<span> </span>

```

```

</div>

```

```

</div>

```

```

</div>

```

```

<div class="block">

```

```

<div class="info">

```

```

<br><a href="">Опис:</a><br>

```

```

<div class="info_p">

```

```

<p style="line-height: 1.6;"><?=$row['name'];
?></p>

```

```

<p style="line-height: 1.6;"><?=$row['description']; ?></p>

```

```

1.6;">Ціна: &nbsp;<?= $row['price']; ?>&nbsp;<?php
    <p class="p_price" style="line-height:
    </div>
    </div>
    </div>
    <div class="block">
        <div class="info">
            <form action="" method="POST">
                <div class="block_button">
                    <?php
                        $info_f = [];
                        $status = "; // Ініціалізуємо змінну $status перед
ЦИКЛОМ
                        // Перевірка, чи змінна $user визначена перед
ВИКЛИКОМ запиту SQL
                        if (isset($user)) {
                            $result_f = mysqli_query($conn, "SELECT
* FROM favorites_table WHERE `id_product` = $new_data AND `id_user` = $user");
                            // Перевірка наявності результату запиту
                            if ($result_f) {
                                while ($row_f =
mysqli_fetch_assoc($result_f)) {
                                    $info_f[] = $row_f;
                                    $status = $row_f['status']; //
Оновлюємо значення $status всередині циклу
                                }
                            } else {
                                // Обробка помилки запиту
                                echo "Помилка запиту до бази
даних: " . mysqli_error($conn);
                            }

```

// Перевіряємо значення \$status після
завершення циклу і виводимо кнопку тільки один раз

```

        if ($status == 'cart') {
            echo "<button type=\"submit\"
name=\"add\"><img src=\"img/icon/kor.png\"><p>В кошик</p></button>";
        } else {
            echo "<button type=\"submit\"
name=\"add\"><img src=\"img/icon/sel.png\"><p>В обране</p></button>";
        }
    } else {

```

// Обробка випадку, коли змінна \$user не
визначена

```

        echo "Змінна \$user не визначена.";
    }
?>
</div>
</form>
</div>
</div>
</div>
<!--//////////////////////////////////////footer-->
<footer id="footer">
    <div class="block_f">
        <p>    <a href="">Facebook</a></p>
        <p><a href="">Instagram</a></p>
        <p><a href="">YouTube</a></p>
        <p><a href="">Viber</a></p>
    </div>
</div>

```

<!--//////////////////////////////////////footer-->

<footer id="footer">

<div class="block_f">

<p> Facebook</p>

<p>Instagram</p>

<p>YouTube</p>

<p>Viber</p>

```

        </div>

        <div class="block_f">
            <p>(068) 094-2926 –
Бухгалтерія</p>
            <p>(068) 316-4932 –
Відділ збуту</p>
        </div>

        <div class="block_f">
            <p>(068) 31-64-932 -
Відділ по роботі з клієнтами</p>
            <p>(098) 00-81-778 -
Відділ по роботі з клієнтами</p>
        </div>

        <div class="block_f">
            <p>dzst@ukr.net</p>
        </div>

    </footer>

</div>

</body>

</html>

```

user_buy.php

```

<?php
session_start();
require_once 'php/connect.php';
if (!$_SESSION['user']) {
    header('Location: pass.php');}
$user = $_SESSION['user']['id_user'];
$info_user = [];

$result_user = mysqli_query($conn, "SELECT * FROM user_table WHERE
`id_user` = '$user'");

```

```

while ($row_user = mysqli_fetch_assoc($result_user)) {
    $info_user[] = $row_user;
}

// Завантаження інформації для кошика

$cart_products = [];

$result_cart = mysqli_query($conn, "SELECT id_product FROM favorites_table
WHERE `id_user` = '$user' AND `status` = 'cart'");

while ($row = mysqli_fetch_assoc($result_cart)) {
    $cart_products[] = $row['id_product'];
}

$cart_products_info = [];

foreach ($cart_products as $id_product) {
    $result_product = mysqli_query($conn, "SELECT * FROM product_table
WHERE `id_product` = '$id_product'");
    if ($product_data = mysqli_fetch_assoc($result_product)) {
        $cart_products_info[] = $product_data;
    }
}

$total_price = 0; // Ініціалізуємо змінну для зберігання загальної суми цін
foreach ($cart_products_info as $data) {
    $result_favorites = mysqli_query($conn, "SELECT * FROM favorites_table
WHERE `id_product` = '" . $data['id_product'] . "'");
    $count = 0; // присвоюємо значення за замовчуванням
    while ($row_favorites = mysqli_fetch_assoc($result_favorites)) {
        $info_favorites[] = $row_favorites;
        $count = $row_favorites['count']; // Отримання значення count
    }
    $total_price += $data['price'] * $count;
}

?>

```

```

<!DOCTYPE html>
<html lang="ua">
<head>
  <meta charset="UTF-8">
  <title>Оформлення замовлення</title>
  <link rel="shortcut icon" href="img/icon/icon.png">
  <link rel="stylesheet" href="css/style.css">
  <link rel="stylesheet" href="css/header.css">
  <link rel="stylesheet" href="css/menu.css">
  <link rel="stylesheet" href="css/user.css">
  <link rel="stylesheet" href="css/info.css">
  <link rel="stylesheet" href="css/form.css">
  <link rel="stylesheet" href="css/footer.css">
  <link rel="stylesheet" href="css/slider.css">
</head>
<body>
  <div class="wrapper">
    <!--////////////////////////////////////header-->
    <header>
      <a href="index.php"></a>
      <div class="header_p">
        <p>Дубовицький</p>
        <p>завод</p>
        <p>сільгосптехніки</p>
      </div>
    </header>
    <!--////////////////////////////////////nav-->
    <div class="menu">

```

```

        <div                class="block_m"><a                href="index.php">Головна</a></div>

```

```

        <div class="block_m"><a href="serch.php">Товари</a></div>

```

```

        <div                class="block_m"><a                href="delivery.php">Доставка та оплата</a></div>

```

```

        <div                class="block_m"><a                href="description.php">Про нас</a></div>

```

```

        <div                class="block_m"><a                href="communication.php">Зв'язок</a></div>

```

```

        <div class="block_m"><a href="pass.php">Особистий кабінет</a></div>

```

```

</div>

```

```

<!--////////////////////////////////////content-->

```

```

<div class="content">

```

```

    <div class="block_panel">

```

```

        <!--

```

```

ОБРАНЕОБРАНЕОБРАНЕОБРАНЕОБРАНЕОБРАНЕОБРАНЕОБРАНЕОБРАНЕ
ОБРАНЕОБРАНЕОБРАНЕОБРАНЕОБРАНЕОБРАНЕОБРАНЕОБРАНЕОБРАНЕ
ОБРАНЕОБРАНЕОБРАНЕОБРАНЕОБРАНЕОБРАНЕОБРАНЕОБРАНЕ-->

```

```

        <div class="panel" style="background-color: #fff;">

```

```

            <div class="info">

```

```

                <br>                <a                href="">ОФОРМЛЕННЯ
ЗАМОВЛЕННЯ:</a>

```

```

            <div class="info_p">

```

```

                <form                action="php/add_to_buy.php"
method="POST">

```

```

                    <input    type="hidden"    name="user_id"
value="<?php echo $user; ?>">

```

```

                    <input type="hidden" name="total_price"
value="<?php echo $total_price; ?>">

```

```

                    <label                for="region">Вкажіть
область:</label>

```

```

id="region" required>
<input type="text" name="region"
<label for="city">Вкажіть місто або
село:</label>
<input type="text" name="city" id="city"
required>
<label>Оберіть спосіб
доставки:</label><br>
<label for="courier"><input type="radio"
name="delivery_method" id="courier" value="На адресу" required> Кур'єрська
доставка</label>
<label for="post"><input type="radio"
name="delivery_method" id="post" value="Пошта" required> Поштова
доставка</label>
<label for="pickup"><input type="radio"
name="delivery_method" id="pickup" value="Самовивіз" required>
Самовивіз</label><br>
<label for="address">Вкажіть
адресу:</label>
<input type="text" name="address"
id="address" required>
<label>Оберіть спосіб
оплати:</label><br>
<label for="cash"><input type="radio"
name="pay_method" id="cash" value="Готівкою" required> Готівкою</label>
<label for="card"><input type="radio"
name="pay_method" id="card" value="Карткою" required> Карткою</label>
<label for="online"><input type="radio"
name="pay_method" id="online" value="Онлайн" required> Онлайн</label><br>
<p>
<button type="submit"
name="buy">ОФОРМИТИ</button>
<button
type="reset">ОЧИСТИТИ</button>

```

```

        </p>
    </form>
</div>
</div>
</div>
<!--
КОШИККОШИККОШИККОШИККОШИККОШИККОШИККОШИККОШИККО
ОШИККОШИККОШИККОШИККОШИККОШИККОШИККОШИККОШИККО
ШИККОШИККОШИККОШИККОШИККОШИККОШИККОШИККОШИККОШ
ИККОШИК-->
<div class="panel">
    <?php foreach ($cart_products_info as $data): ?>
        <div class="card">
            <div class="card_left">
                <p><a href="product.php?data=<?php echo
                $data['id_product']; ?>"></a></p>
            </div>
            <div class="card_right">
                <?php
                $info_favorites = [];
                $result_favorites = mysqli_query($conn, "SELECT * FROM
                favorites_table WHERE `id_product` = " . $data['id_product'] . "");

                while ($row_favorites = mysqli_fetch_assoc($result_favorites)) {
                    $info_favorites[] = $row_favorites;
                    $count = $row_favorites['count']; // Отримання значення count
                    $id = $row_favorites['id_favorite']; // Отримання значення id
                }
                ?>
                <p>Кількість: &nbsp;<?php echo $count; ?></p>
            </div>
        </div>
    </foreach>
</div>

```

```

        <p><?php echo $data['name']; ?></p>
        <p>Ціна:  &nbsp;<?php  echo  $data['price'];
?>&nbsp;<?php  грн.</p>

        <p      style="display:      flex;      align-items:
center;">Статус:  <span></span></p>

        </div>

    </div>

    <?php endforeach; ?>

</div>

</div>

    <div class="block">
        <div class="info">

            <a href="">Сума до сплати: &nbsp;<?php echo $total_price;
?>&nbsp;<?php  грн.</a>

        </div>

    </div>

</div>

<!--//////////////////////////////////////footer-->
<footer id="footer">
    <div class="block_f">
        <p>      <a      href="">Facebook</a></p>
        <p><a      href="">Instagram</a></p>
        <p><a      href="">YouTube</a></p>
        <p><a      href="">Viber</a></p>

```

</div>

<div class="block_f">

<p>(068) 094-2926 –
Бухгалтерія</p>

<p>(068) 316-4932 –
Відділ збуту</p>

</div>

<div class="block_f">

<p>(068) 31-64-932 -
Відділ по роботі з клієнтами</p>

<p>(098) 00-81-778 -
Відділ по роботі з клієнтами</p>

</div>

<div class="block_f">

<p>dzst@ukr.net</p>

</div>

</footer>

</div>

</body>

</html>