

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ЦЕНТР ЗАОЧНОЇ, ДИСТАНЦІЙНОЇ ТА ВЕЧІРНЬОЇ ФОРМ НАВЧАННЯ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК**

Секція інформаційно-комунікаційних технологій

ВИПУСКНА РОБОТА

на тему: «Клієнтська частина веб-сайту за допомогою JavaScript»

Завідувач

випускаючої кафедри

Довбиш А. С.

Керівник роботи

Шутильєва О. В.

Студентки гр. Інз-61С

Теницька А.С.

СУМИ 2020

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

Кафедра комп'ютерних наук

Затверджую _____

Зав. кафедрою Довбиш А.С.

« _____ » _____ 2020 р.

ЗАВДАННЯ

до випускної роботи

Студентки четвертого курсу, групи ІНз-61С спеціальності «Комп'ютерні науки» заочної форми навчання Теницька Альона Сергіївна

Тема: «Клієнтська частина веб-сайту за допомогою JavaScript»

Затверджена наказом по СумДУ

№ _____ від _____ 2020 р.

Зміст пояснювальної записки: 1) інформаційний огляд; 2) постановка задачі та опис інструменту вирішення поставлених задач; 3) практична реалізація.

Дата видачі завдання « _____ » _____ 2020 р.

Керівник випускної роботи _____ Шутилева О.В.

Завдання прийняв до виконання _____ Теницька А.С.

РЕФЕРАТ

Записка: 74 стор., 23 рис., 5 додатків, 25 джерел.

Об'єкт дослідження – сфера інтернет-торгівлі в агрономічній промисловості

Мета роботи – розробити клієнтську частину інтернет-магазину з продажу добрив з використанням актуальних інструментів для створення веб-ресурсів

Методи дослідження – експертний аналіз інтернет-магазинів аналогів, експертний аналіз інструментів для розробки інтернет-магазину, SMART

Результати – розроблена клієнтська частини інтернет-магазину бренду «Agrohelp» за допомогою JavaScript (JS), HTML, CSS, PHP, який дозволяє аграріям легко купувати добрива від виробника.

ІНТЕРНЕТ-МАГАЗИН, JAVASCRIPT, HTML,
CSS, PHP, MySQL

ЗМІСТ

ВСТУП	5
1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	6
1.1 Актуальність проблеми.....	6
1.2 Аналіз існуючих аналогів	7
1.3 Постановка задачі	11
2 ІНСТРУМЕНТИ ДЛЯ ВИРІШЕННЯ ПОСТАВЛЕНИХ ЗАДАЧ	13
2.1 Проектування клієнтської частини.....	16
2.2 Use Cases діаграма	21
3 РЕАЛІЗАЦІЯ КЛІЄНТСЬКОЇ ЧАСТИНИ ІНТЕРНЕТ-МАГАЗИНУ	3
ПРОДАЖУ ДОБРІВ «AGRONELP».....	23
3.1 Визначення структури інтернет-магазину	23
3.2 Проектування та реалізація бази даних.....	26
3.3 Розробка інтерфейсу.....	28
3.4 Розробка програмного коду.....	33
ВИСНОВКИ.....	36
СПИСОК ЛІТЕРАТУРИ.....	37
ДОДАТОК А.....	39
ДОДАТОК Б	42
ДОДАТОК В.....	50
ДОДАТОК Г	68
ДОДАТОК Д.....	72

ВСТУП

Сучасні проблеми потребують сучасних рішень. Раніше, щоб зробити важливі покупки, треба було вийти з дому або офісу. Потім дійти до магазину, підійти до продавця, запитати про наявність товару і вже при наявності та приємній ціні придбати його.

Відтепер завдяки інформаційним технологіям ми маємо інтернет-магазини. Ми заходимо на нього через мережу інтернет, відразу бачимо інформацію про ціну та наявність, натискаємо на кнопку «Додати у кошик», заповнюємо форму та очікуємо на доставку товару поштою або кур'єром.

Особливо це актуально для великих підприємств, в цьому випадку – для аграрної сфери. При вирощенні врожаю є постійний попит на допоміжні засоби, зокрема добрива.

Аграрії можуть не мати часу виїхати в магазин, тому їм зручніше приєднатися до мережі інтернет та за декілька хвилин зробити замовлення потрібних добрив.

Окрім цього, більшість виробників не мають власних інтернет-магазинів. Це впливає на ціну для кінцевого споживача та популярність продукту виробника, адже на звичайному аграрному інтернет-магазині його бренд загубиться серед інших.

На основі аналізу ринку аналогів було сформульовано мету дипломної роботи: розробка інтернет-магазину органічних добрив бренду «Сиптомат» з використанням JavaScript.

Для досягнення мети необхідно вирішити наступний перелік задач:

- проаналізувати предметну область;
- розробити структуру інтернет-магазину;
- розробити інтернет-магазин;
- додати товари до інтернет-магазину;
- провести тестування продукту.

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Актуальність проблеми

Якщо виробник хоче заробляти гроші на своєму продукті, він повинен бути зручним для кінцевого користувача. Важливо відслідковувати тренди не лише торгівлі, а також і веб-розробки. Перед тим, як створити інтернет-магазин, потрібно проаналізувати:

- Актуальність існуючих мов програмування, фреймворків, інструментів;
- Оптимальність інструментарію, потрібного для створення власного інтернет-магазину;
- Тенденції торгівлі, оформлення сайту.

Виробник повинен спільно з розробником розглянути всі ці аспекти, спираючись на user experience (UX), адже він має за мету не лише існувати як майданчик своєї продукції, а також заробляти на продажі та бути лідером серед конкурентів.

Роль розробника в цьому процесі – це бути в курсі веб-технологій, які вплинуть на зручність та швидкодію інтернет-магазину. Розробнику важливо використовувати актуальний інструментарій на 100%, але також враховувати ризики, що можуть погіршити роботу сайту та не надати потрібний ефект.

Робота розробника полягає у дослідженні сфери, для якої він створює продукт. Аналіз конкурентів, сфери для якої створюється інтернет-магазин, перемовини із замовником, присутність на фокус-групах щодо створення продукту – це все потрібно для написання якісного інтернет-магазину.

Якщо раніше покупці звертали увагу на яскравий дизайн та великі тексти про кожен продукт і бренд в цілому, то зараз все що їм потрібно – це швидка робота інтернет-магазину, мати всі потрібні елементи під рукою для пошуку потрібних товарів та оформлення замовлення.

Сфера інтернет-магазинів переходить до спрощення. Все коротко: категорії товарів, поле пошуку по ключовим словам, короткий опис товару з

наявністю потрібних характеристик, кнопка «Додати у кошик» та мінімальна форма «Оформити замовлення».

Раніше, коли можливість продавати була лише оф-лайн, орендувалося одне або декілька приміщень для магазинів або складів, наймалися продавці, адміністратори, охоронці, бухгалтери, розроблялася логістика перевезень товарів, замовлялися недешеві вантажоперевезення. Також потрібно було купувати рекламу у газетах, на білбордах, на радіо.

Інтернет-магазин дозволяє зекономити кошти. Для його створення потрібно замовити розробника, який виконає роботу. Для підтримки потрібно оплачувати хостинг та послуги адміністратора сайту. В залежності від масштабів продаж, наймаються онлайн-продавці, які контролюють процес замовлень: передача їх до поштових сервісів та доставка до покупця. Оренднуються лише склади, на яких зберігається товар та оплачується робота вантажників, які передають замовлення до поштових сервісів.

Тому інтернет-магазин це не лише зручність цільовій аудиторії виробника, а також економія його коштів завдяки автоматизації багатьох процесів.

1.2 Аналіз існуючих аналогів

Напередодні розробки інтернет-магазину органічних добрив, потрібно проаналізувати аналоги. У наш час, незважаючи на велику кількість інтернет-магазинів з продажу засобів для аграрної промисловості, цільовій аудиторії частіше за все легше буде поїхати до оф-лайн магазину та придбати потрібне. Перш за все, проблема полягає в тому, що більшість інтернет-магазинів незручні у використанні. Недолік інтернет-магазинів з продажу добрив – це недостатня оптимізація коду, що впливає на швидкодію, а в свою чергу швидкодія погіршує зручність користування. Також вони мають багато зайвих елементів, немає оперативної комунікації, через що доводиться писати на пошту або дзвонити по телефону, що займає час.

Першим аналогом інтернет-магазину з продажі добрив є UkrSemena. UkrSemena – це магазин з продажі різних видів добрив від різноманітних виробників [2]. В інформації про себе вказують, що мають найнижчі ціни, зручну підтримку та легку оплату.

Проте, виконавши аналіз відгуків від користувачів магазину, було отримано наступні дані:

- відсутність можливості замовлення за допомогою «Нової пошти»;
- занадто велика кількість добрив;
- низька швидкодія сайту;
- недостатня адаптивність для мобільних пристроїв;
- замовлення лише від 200 грн;

На рисунку 1.1 представлена початкова сторінка UkrSemena.

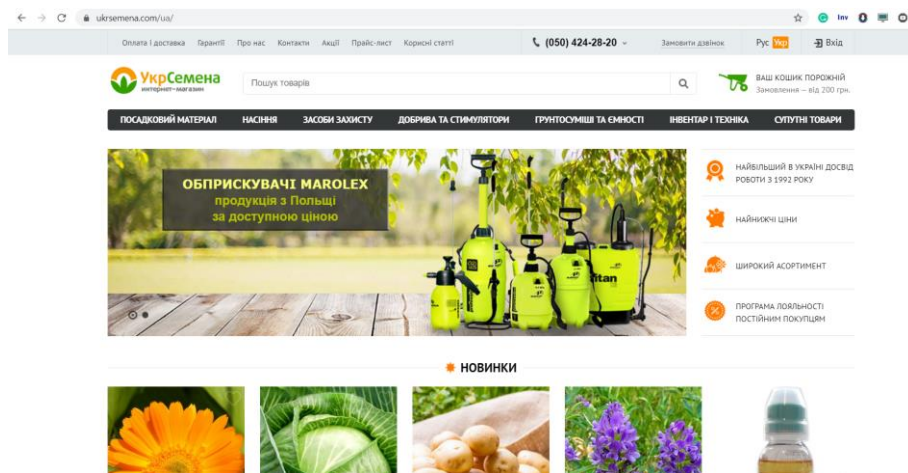


Рисунок 1.1 – Початкова сторінка UkrSemena

Платформа передбачає реєстрацію користувача. Проте, існує можливість авторизації за допомогою соціальної мережі «Вконтакте», яка є забороненою в Україні. Окрім цього, також можливо зареєструватися за допомогою Facebook, але при спробі це зробити виникає помилка. Реєстраційну форму та помилку можна переглянути на рисунку 1.2.

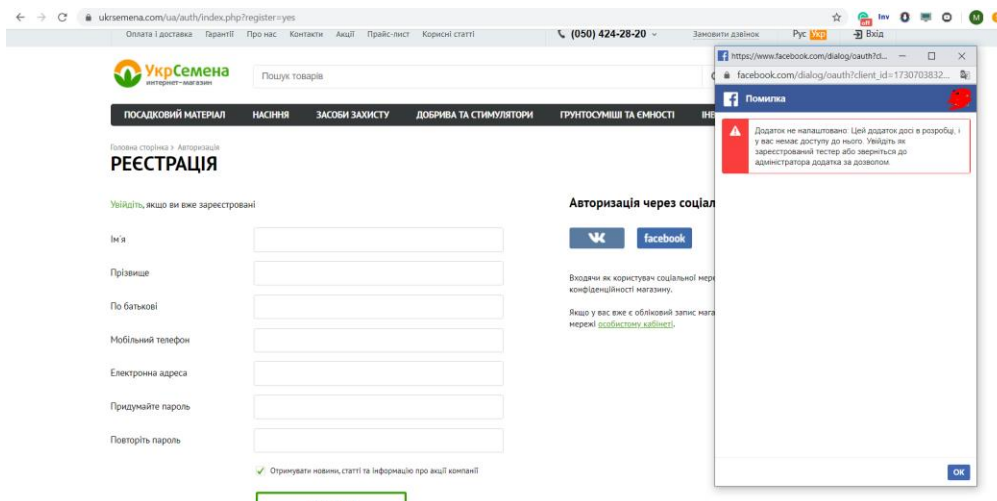


Рисунок 1.2 – Реєстраційна форма та помилка авторизації через Facebook

Це впливає на можливу покупку. При роботі з великою кількістю потрібних товарів, у замовника немає часу на додаткову авторизацію і тому користувач приймає рішення знайти більш зручний сайт.

Наступний аналог розглянутих інтернет-магазинів– «Професійні насіння», інтернет-магазин виробника насіння, добрив, засобів захисту врожаю, біопрепаратів тощо [3]. Авторизуватися на даному сайті можна за допомогою Facebook або Google-акаунту, форма реєстрації містить в собі лише декілька важливих пунктів, що пришвидшує даний процес для кінцевого користувача.

Даний інтернет-магазин пропонує більший вибір способів доставки та різноманітний асортимент, перша сторінка зустрічає зручними іконками для навігації по сайту (рис.1.3).

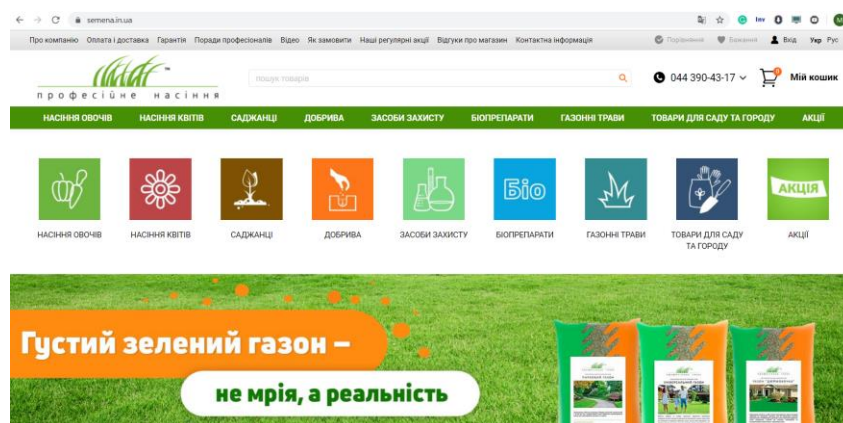


Рисунок 1.3 – Інтернет-магазин «Професійні насіння»

Дане зображення демонструє, який асортимент пропонується користувачу при першому відвідуванні сторінки. Недоліки даного інтернет-магазину наступні:

- відсутність зручності комунікації з магазином;
- довге оброблення замовлення.

Останнім розглянутим аналогом є інтернет-магазин – AgreeMarket [4]. Магазин має приємні кольори для очей користувача, що може приємно вплинути на можливу покупку. На стартовій сторінці бачимо блок навігації по сайту та по категоріям товарів; новини, пов'язані з роботою магазину; актуальні акції; контакти для звернення.

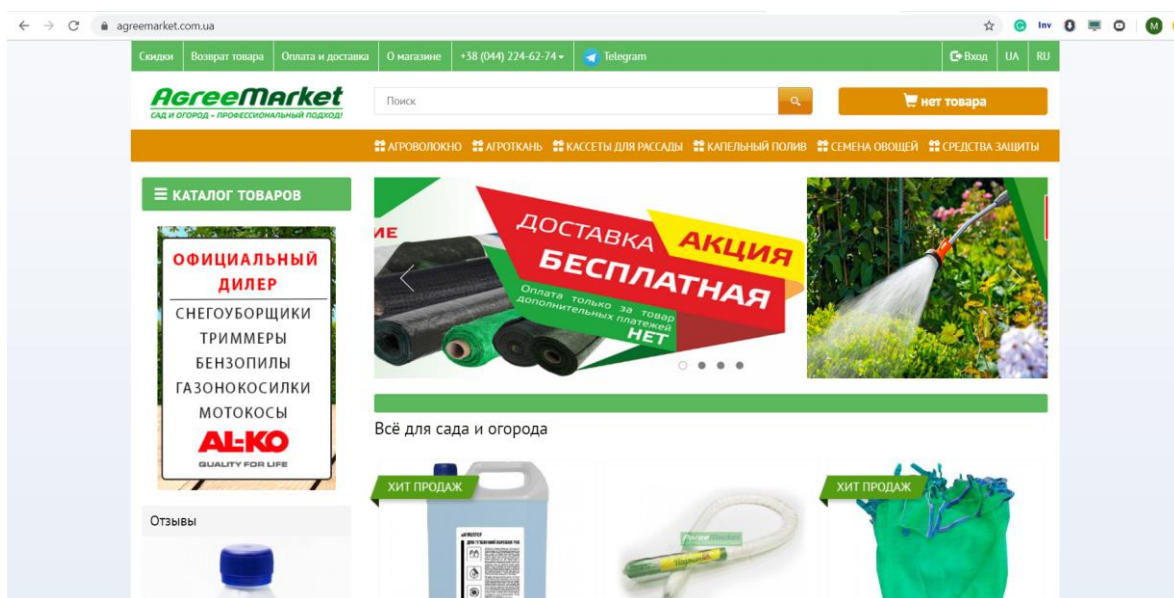


Рисунок 1.4 – Інтернет-магазин AgreeMarket

Всі аналоги об'єднує одне: різноманітність брендів. Під час пошуку аналогів, не було виявлено власних інтернет-магазинів виробників, що впливає на їх прибуток та впізнаваність серед цільової аудиторії.

Згідно аналізу аналогів, була сформована порівняльна таблиця розглянутих прикладів та майбутнього інтернет-магазину. Результати аналізу представлено у таблиці 1.1.

Таблиця 1.1 – Порівняльна характеристика аналогів

Назва критерію	Назва ресурсу			
	UkrSemena	Професійне насіння	Agreemarket	AgroHelp
1. Використання ресурсу без реєстрації	+	+	+	+
2. Зручність навігації	-	-	+	+
3. Належність до одного бренду	-	-	-	+
4. Адаптивність сайту під мобільну версію	-	-	+	+

1.3 Постановка задачі

Метою випускної роботи є розробка інтернет-магазину з продажу добрив компанії «Agrohelf» з використанням JavaScript (JS). Продукт дозволить популяризувати виробника та продавати продукцію за нижчими цінами, ніж на інших майданчиках. Окрім цього, він кардинально відрізнятиметься від інших магазинів завдяки простому дизайну та швидкості завантаження сторінок. Інтернет-магазин буде мати попит серед аграріїв, які шукатимуть добрива для свого врожаю.

При подальшій підтримці та модернізації інтернет-магазину існує можливість автоматизувати більшість процесів: чат-бот, який відповідатиме на найпоширеніші питання; чат для спілкування з оператором магазину; автоматична обробка та передача замовлення до офісу виробника.

Розроблений інтернет-магазин повинен підтримувати наступні функції використання:

- можливість перегляду наявного асортименту добрив;
- можливість замовлення без реєстрації;
- можливість порівняння товарів;

Функціональні вимоги до веб-додатку:

- адаптивність сайту;
- наявність форми реєстрації/входу;
- наявність фільтрів для сортування товарів.

Нефункціональні вимоги:

- простота дизайну
- зрозумілі навігація та процес оформлення замовлення.

Інтернет-магазин «Agrohelp» повинен бути розроблений за останніми трендами сайтів, адже магазин повинен мати попит серед аграріїв та займати конкуруючі позиції серед інших виробників. повинен мати попит серед сучасних користувачів-початківців. Окрім цього, розробник повинен пам'ятати, що більшість замовників використовує мобільні пристрої і тому інтернет-магазин повинен бути адаптивним під будь-які розміри.

Для реалізації поставленої мети в дипломному проекті необхідно реалізувати наступні задачі:

- провести аналіз предметної області;
- провести аналіз технологій розробки інтернет-магазину;
- розробити інтерфейс інтернет-магазину;
- провести проектування бази даних;
- виконати підключення бази даних до інтернет-магазину;
- провести тестування коректної роботи інтернет-магазину.

Для покрокового та коректного процесу реалізації інтернет-магазину з урахуванням всіх вимог до функціоналу було розроблено технічне завдання.

2 ІНСТРУМЕНТИ ДЛЯ ВИРІШЕННЯ ПОСТАВЛЕНИХ ЗАДАЧ

Процес розробки інтернет-магазинів включає наступні етапи:

- розробка макету сайту;
- верстка шаблону;
- програмування інтерактивних елементів;
- наповнення сайту товаром;

У зв'язку з вище зазначеними етапами розробки та необхідність наявності вище згаданих елементів, було прийнята рішення про використання наступних технологій:

- HTML та CSS
- JavaScript
- PHP
- MySQL
- jQuery
- Bootstrap
- AJAX

HTML – це комп'ютерна мова, розроблена для створення веб-сайтів. Сайти, спроектовані мовою HTML, можуть бути переглянуті на будь-якому електронному носії, що мають підключення до Інтернет-мережі. Також технологія HTML є досить легкою у вивченні для будь-якого початкового користувача. Розвиток HTML не стоїть на місці, тому існує велика кількість ресурсів для вивчення тих чи інших аспектів використання можливостей HTML [5].

HTML впроваджує засоби для:

- створення документу завдяки структуризації складу тексту: заголовки, абзаци, цитати та інше;
- отримання інформації із мережі Інтернет з використанням гіперпосилання;
- включення медіа-контенту та інших об'єктів на сторінку;
- розробка інтерактивних форм.

CSS розшифровується як каскадні таблиці стилів з акцентом на «Стиль». Хоча HTML використовується для структури веб-документа (визначаючи такі речі, як заголовки та абзаци, і дозволяючи вставляти зображення, відео та інші медіа), CSS визначає стиль спроектованого документа – макети сторінки, кольори та шрифти визначаються. за допомогою CSS [6].

Окрім цього, за допомогою CSS можна розробити різне відображення на різних пристроях, тобто адаптувати під мобільні пристрої, телевізори тощо. Також стандарт CSS визначає, в якому порядку та яким чином застосовувати стилі. Тобто, застосовується принцип каскадності, коли для елементів вказується лише змінена інформація про стилі.

Переваги використання CSS:

- розмежування коду і оформлення;
- різне оформлення для різних пристроїв;
- розширені в порівнянні з html способи оформлення елементів;
- прискорення завантаження сайту;
- єдине стильове оформлення безлічі документів;
- централізоване зберігання [7].

JavaScript – це сценарій або мова програмування, що дозволяє реалізовувати складні функції на веб-сторінках:

- відображення своєчасного оновлення вмісту;
- підключення інтерактивних карт;
- підключення анімації 2D/3D-графіки;
- прокрутка відео роботів [8].

Також JavaScript у браузері:

- реагує на дії користувача: виконання натискання миші, руху вказівника, натискання клавіш;
- надсилає запити по мережі на віддалені сервери, завантажує та вивантажує файли;
- робота з cookie-файлами;
- функція локального зберігання [9].

jQuery – це популярна open-source JS-бібліотека, що була представлена Джоном Ресігом в січні 2006 року у BarCamp NYC Джоном. Згідно з інформацією досліджень W3Techs, JQuery використовується більшою половиною мільйона популярних веб-ресурсів. jQuery є бібліотекою JavaScript, яка активно використовується в сьогоденні. JQuery є freeware-забезпеченням під ліцензією MIT.

Bootstrap – це збірка open-source інструментів, що використовується для створення веб-ресурсів та містить шаблони CSS та HTML для різноманітних компонентів інтерфейсу. Збірка націлена на спрощення розробки динамічних веб-ресурсів.

Bootstrap – це клієнтський фреймворк, що націлений на роботу з інтерфейсом для користувача. Репозиторій із цим фреймворком є популярним на GitHub. До того ж, його використовують MSNBC та NASA. [11]

PHP – це open-source сценарійна мова загального призначення, що підходить для веб-розробки та має можливість вбудови в HTML [12].

Причини використання PHP:

- Open-source;
- швидкість вивчення мови;
- більшість веб-хостингів за замовчуванням підтримують PHP;
- регулярно оновлюється згідно технологічних трендів [13].

MySQL – це open-source підтримувана Oracle СУБД, заснована на мові SQL. MySQL є крос-платформеною, тобто працює на Windows, Linux та Unix.

MySQL дозволяє працювати з даними та зберігати їх завдяки різним механізмам зберігання, таким як: InnoDB, CSV і NDB. Користувачі MySQL можуть отримати доступ до своїх даних за допомогою стандартних команд SQL [14].

MySQL має наступні переваги у використанні:

- підвищення продуктивності програм;
- висока швидкість процедури зберігання даних;
- портативність MySQL;

– безпека процедури зберігання даних [15].

AJAX (асинхронний JavaScript та XML) – це опис набору методів розробки, що використовуються для створення веб-ресурсів. За словами веб-розробника і інструктора WordPress по Skillcrush Енн Каскарано, кращий спосіб зрозуміти AJAX – почати з визначення його конкретної мети в процесі веб-розробки. Основна функція AJAX - асинхронне оновлення веб-вмісту, тобто веб-браузеру користувача не потрібно перезавантажувати всю веб-сторінку для повного змінення даних є основною функцією AJAX[16].

AJAX використовує XHTML для вмісту, CSS для презентації, а також Модель об'єкта документа та JavaScript для динамічного відображення вмісту. Звичайні веб-додатки передають інформацію до та від серйозних, використовуючи синхронні запити. Це означає, що користувач заповнює форму, натискає надіслати та направляє на нову сторінку нову інформацію з сервера. Із AJAX, коли користувач натисне надіслати, JavaScript подасть запит на сервер, інтерпретує результати та оновить поточний екран. У чистому сенсі користувач ніколи не дізнається, що щось навіть передається на сервер [16].

2.1 Проектування клієнтської частини

Під час проектування необхідно розробити діаграми нотації IDEF0 для представлення процесу розробки проекту та розробити діаграму використання для демонстрації сценаріїв дій акторів відносно інтернет-магазину. У Додатку Б детально описаний процес планування робіт для реалізації проекту. Діаграми IDEF0 – методологія функціонального моделювання та графічна нотація для представлення бізнес-процесів. Особливістю IDEF0 є в тому, що в даній нотації розглядається логічне відношення, послідовність робіт, без часової залежності.

Модель IDEF0 представляє собою набір блоків, кожний із яких представляє собою «чорний ящик», що має входи, виходи, механізми та управління. Кожний блок деталізується до рівня, який необхідний розробнику [22].

Перший блок нотації IDEF0 це діаграма А–0, що дає загальне представлення процесу, який буде у майбутньому детально описаний під рівнями. Для проекту «Клієнтська частина інтернет-магазину з продажу добрив за допомогою JavaScript» контекстна діаграма А–0 складається з даних, що описані нижче:

- Вхідні дані: технічне завдання, найменування та зображення продукції виробника.
- Вихідні дані: інтернет-магазин.
- Управління: методологія створення інтернет-магазину, загальноприйняті принципи макетування.
- Механізми: HTML, CSS, Javascript, AJAX, PHP, MySQL, види схем розробки, концептуальна модель БД.

На рисунку 2.1 представлена контекстна діаграма А–0.

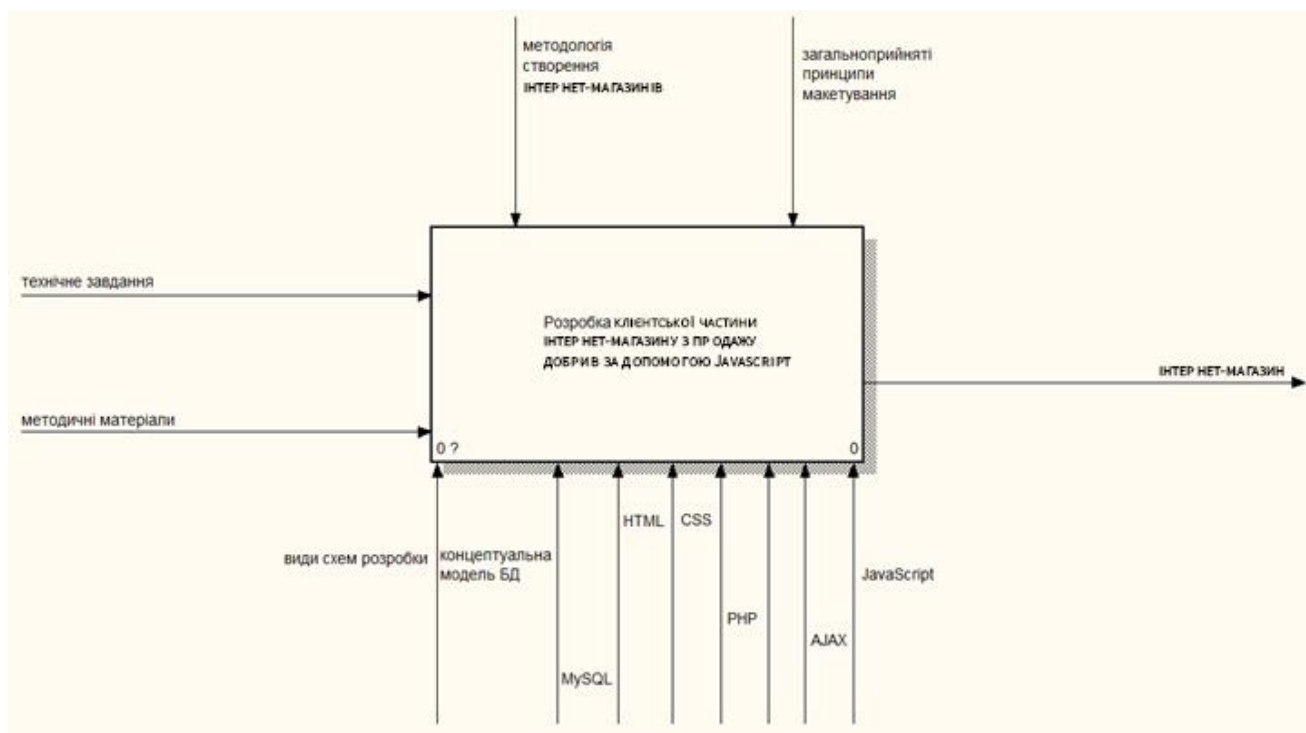


Рисунок 2.1 – Діаграма першого рівня

Так як діаграма присвячена процесу розробки клієнтської частини інтернет-магазину, далі необхідно визначити ключові етапи даного процесу.

Оперуючи методологією розробки інтернет-магазину, розробником проекту було вирішено, що складові етапи процесу створення інтернет-магазину є:

- проектування клієнтської частини інтернет-магазину;
- реалізація клієнтської частини інтернет-магазину.

Для етапу «Проектування клієнтської частини інтернет-магазину» були визначені дані, що описані нижче:

- Вхідні дані: технічне завдання.
- Вихідні дані: каркас інтернет-магазину.
- Управління: загальноприйняті принципи макетування.
- Механізми: види схем розробки інтернет-магазину.

Аналогічним чином були визначені дані для блоку «Реалізація інтернет-магазину»:

- Вхідні дані: технічне завдання, методичні дані.
- Вихідні дані: інтернет-магазин.
- Управління: методологія створення інтернет-магазинів, база даних інтернет-магазину.
- Механізми: HTML, CSS, PHP, JavaScript, AJAX.

На рисунку 2.2 представлена діаграма другого рівня деталізації.

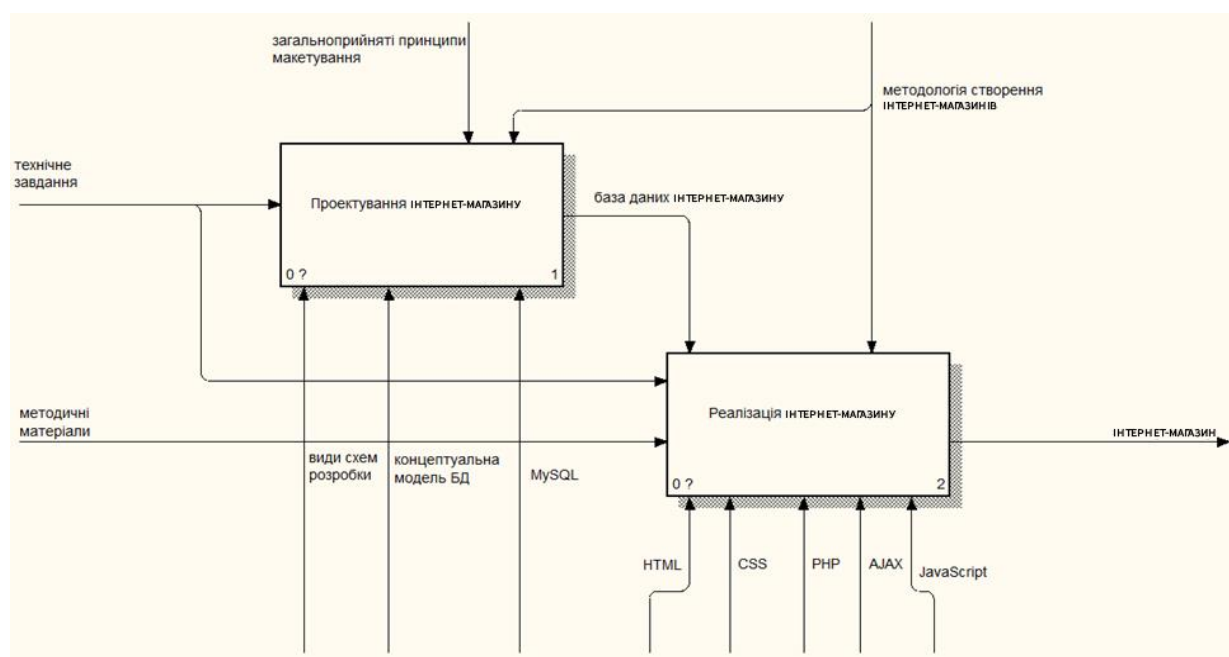


Рисунок 2.2 – Діаграма другого рівня

Коли було виконано деталізацію другого рівня, необхідно виконати розбиття кожного процесу на підрівні для повноцінного представлення процесу розробки клієнтської частини інтернет-магазину.

Під час процесу проектування клієнтської частини інтернет-магазину необхідно було провести експертний аналіз схем розробки онлайн-ресурсів та обрати оптимальну схему, яка б відповідала всім вимогам. Також на даному етапі необхідно було провести проектування бази даних та виконати її реалізацію для подальшого використання під час розробки інтернет-магазину.

З огляду на етапи роботи під час проектування інтернет-магазину були визначені процеси:

- визначення структури інтернет-магазину;
- проектування бази даних.

Для процесу «Визначення структури інтернет-магазину» розробником проекту були зазначені дані:

- Вхідні дані: технічне завдання.
- Вихідні дані: визначена схема клієнтської частини інтернет-магазину.
- Управління: загальноприйняті принципи макетування, методологія створення інтернет-магазинів.
- Механізми: види схем розробки інтернет-магазинів.

Для процесу «Проектування бази даних» аналогічно були визначені дані, що зазначені нижче:

- Вхідні дані: технічне завдання.
- Вихідні дані: база даних інтернет-магазину.
- Управління: визначена схема інтернет-магазину, загальноприйняті принципи макетування, методологія створення інтернет-магазинів.
- Механізми: концептуальна модель бази даних, MySQL.

На рисунку 2.3 представлена діаграма деталізації процесу «Проектування інтернет-магазину».

Аналогічним чином було проведено деталізацію процесу «Реалізація інтернет-магазину». Даний процес складається з двох ключових етапів:

- розробка каркасу інтернет-магазину.
- підключення функціональної частини.

Для блоку «Розробка каркасу інтернет-магазину» були визначені дані, що перелічені нижче:

- Вхідні дані: технічне завдання.
- Вихідні дані: каркас інтернет-магазину.
- Управління: методологія створення інтернет-магазинів.
- Механізми: HTML, CSS, PHP, AJAX, JS, Bootstrap.

За аналогією були визначені дані для блоку «Підключення функціональної частини»:

- Вхідні дані: технічне завдання, методичні матеріали.
- Вихідні дані: інтернет-магазин.
- Управління: методологія створення інтернет-магазинів, каркас інтернет-магазину, база даних інтернет-магазину.
- Механізми: AJAX, JavaScript, MySQL, PHP.

На рисунку 2.4 представлена діаграма деталізації процесу «Реалізація інтернет-магазину».

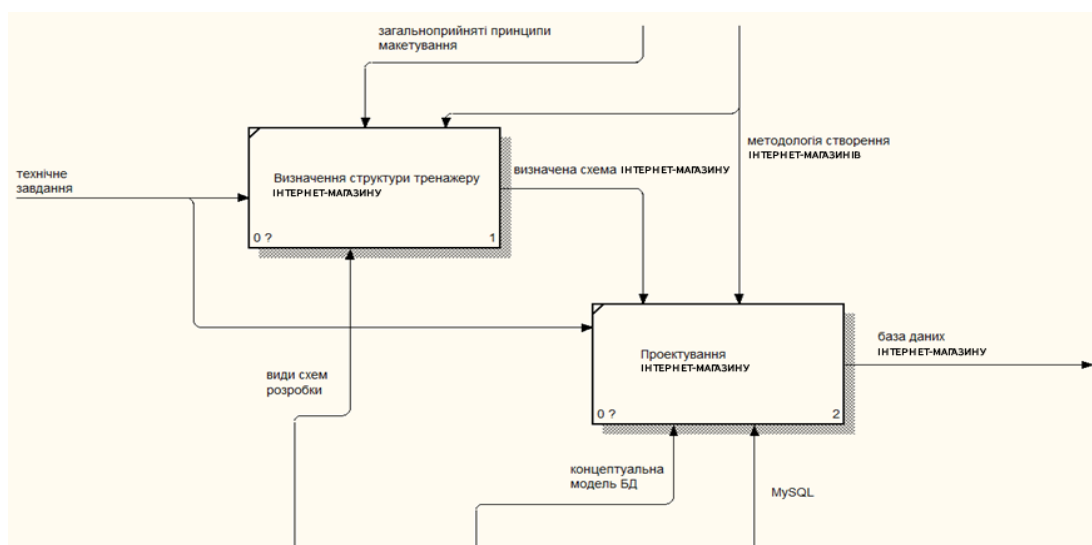


Рисунок 2.3 – Деталізація процесу «Проектування інтернет-магазину»

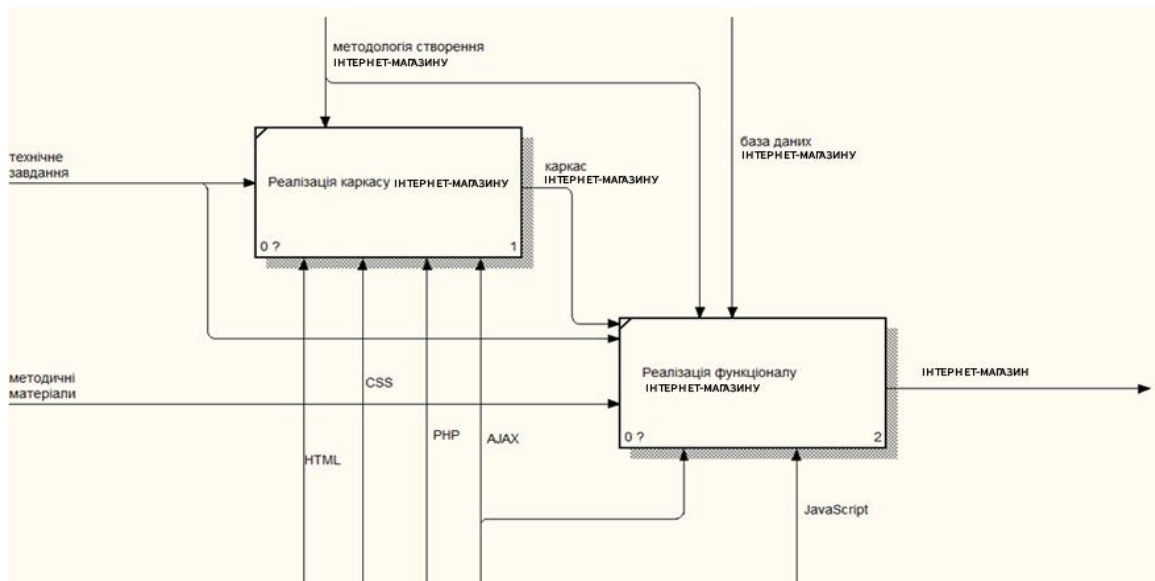


Рисунок 2.4 – Деталізація процесу «Реалізація інтернет-магазину»

2.2 Use Cases діаграма

Діаграма варіантів використання (use case diagram) описує взаємовідносини та залежності між варіантами використання та акторами, що приймають участь у процесі.

Актором виступає дієве обличчя, що є зовнішнім явищем та взаємодіє із системою, використовуючи певний сценарій. Акторами можуть виступати як реальні фізичні особи (клієнт, адміністратор), так і не фізичні суб'єкти (допоміжні програмні засоби) [23].

Для проекту «Клієнтська частина інтернет-магазину з продажу добрив за допомогою JavaScript» були визначені наступні актори:

- Користувач – суб'єкт, що замовляє продукцію.
- Адміністратор – суб'єкт, що виконує обробку замовлень.
- База даних – суб'єкт, що виконує запис про замовлення.
- Local Storage – локальне сховище користувача, на якому зберігаються дані про обрані товари.

На рисунку 2.5 представлена діаграма використання проекту «Клієнтська частина інтернет-магазину з продажу добрив за допомогою JavaScript».

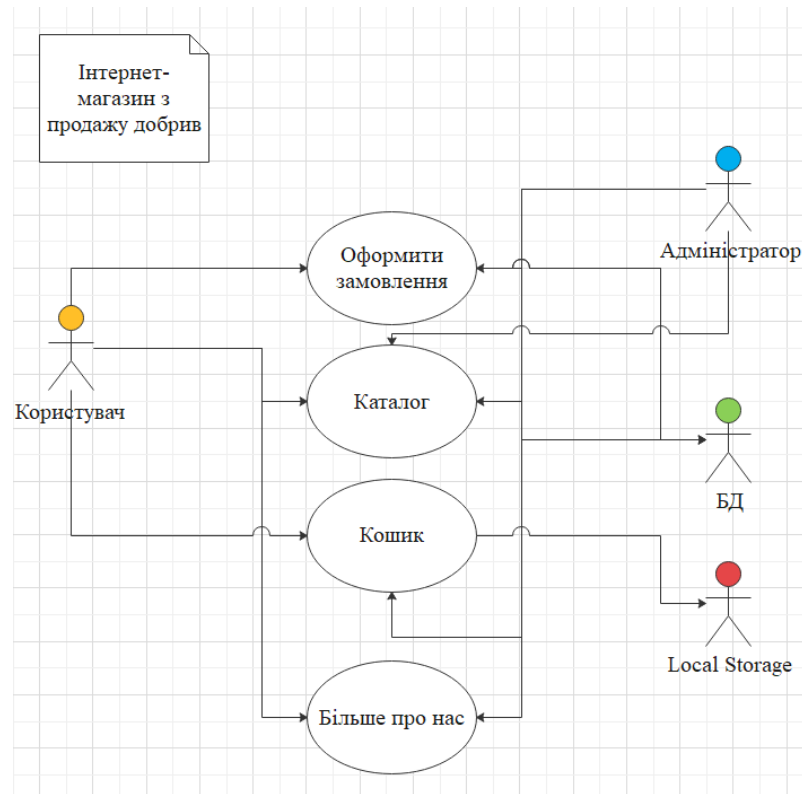


Рисунок 2.5 – Use case diagram

3 РЕАЛІЗАЦІЯ КЛІЄНТСЬКОЇ ЧАСТИНИ ІНТЕРНЕТ-МАГАЗИНУ З ПРОДАЖУ ДОБРИВ «AGROHELP»

3.1 Визначення структури інтернет-магазину

Інтернет-магазин має свою структуру розміщення товарів та елементів для зручної навігації. Інтернет-магазин «Agrohelp» розроблений за трендами мінімалізму та UX. Магазин має простий дизайн та зрозумілу навігацію, що впливає на швидкодію під час роботи з ним.

При процесі реалізації ресурсу розробником було розглянуто структуру сторінки на двох рівнях: зовнішній(логічний) та внутрішній(фізичний).

Логічна структура побудована з урахуванням компетентності різних вікових категорій цільової аудиторії інтернет-магазину. При невдалій побудові логічної структури сайту, користувачі можуть зіштовхнутися з такими проблемами:

- відсутність комфортного перегляду інформації на сайті;
- втрата бажання використовувати ресурс;
- неможливість комфортної роботи з сайтом.

Все це впливає на кількість покупок, відповідно це вплив на прибуток.

Для інтернет-магазину «Agrohelp» був розроблений інтуїтивно зрозумілий та привабливий дизайн. Зовнішня структура сайту зображена на рисунку 3.1.

Структура магазину розроблена таким чином, щоб користувач з перших хвилин його відвідування зрозумів, з якою продукцією має справу, як йому зробити замовлення, де перевірити свій кошик, оформити замовлення або дізнатися більше про виробника. Окрім цього, на кожній сторінці присутній footer, в якому дублюються кнопки верхньої панелі та вказана контактна інформація, за якою користувач може звернутися для консультації щодо придбання.

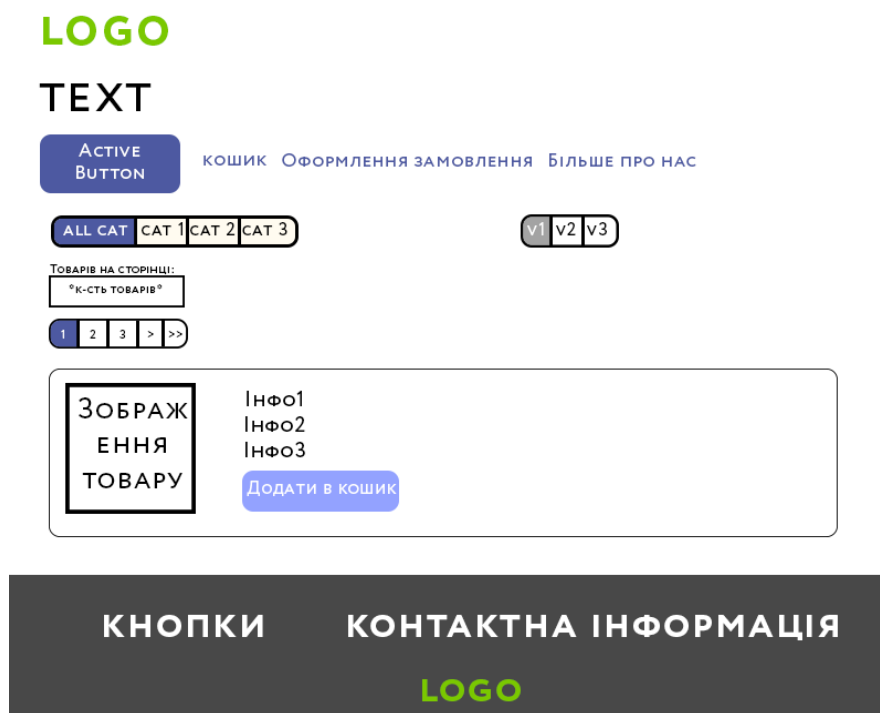


Рисунок 3.1 – Зовнішня структура інтернет-магазину

При розробці внутрішній структурі інтернет-магазину було розглянуто різні схеми:

- лінійна;
- ієрархічна;
- мережева;
- комбінована.

Лінійна структура – структура, при якій сторінки посилаються один на одного і одночасно на головну. Цей тип структури відмінно підходить для портфолію, презентаційних ресурсів ті інших специфічних Інтернет-майданчиків, мето яких є послідовне ознайомлення користувачів з усім наявним контентом. Увага читача тут переходить від головної сторінки до останньої, проходячи через всі інші. Мінус такої схеми в тому, що успішному просуванню підлягає лише головна сторінка. А значить, залучати нових відвідувачів буде важко. Лінійна структура більше підходить тим, для кого не принципово отримувати трафік з пошуку [24]. На рисунку 3.2 схематично зображена лінійна структура веб-ресурсу.

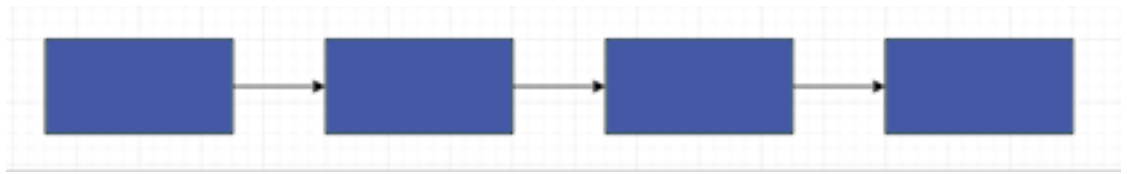


Рисунок 3.2 – Схема лінійної структури

Ієрархічна структура – схожий тип з лінійною структурою. Тільки тут можна застосовувати одночасно кілька продуктів. Найпоширеніший приклад такої схеми - онлайн-бібліотека одного учасника з його творами. Читач тут, знову ж таки, буде рухатися від головної сторінки. Для SEO-просування ця структура сайтів інтернету не підходить[24]. На рисунку 3.3 представлена схема ієрархічної структури веб-ресурсу.

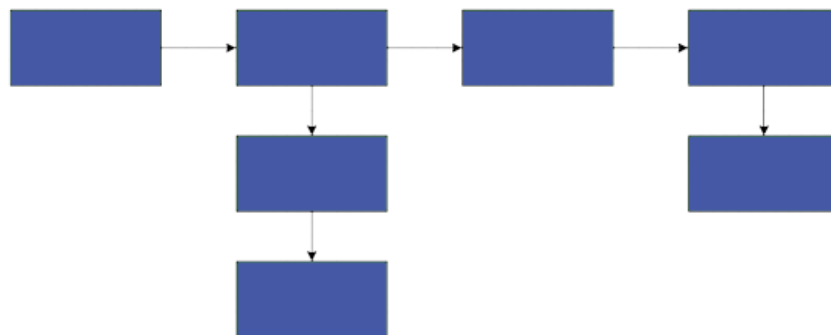


Рисунок 3.3 – Схема ієрархічної структури

Мережева структура – структура, при якій кожна сторінка посилається на кілька інших, рівнозначних між собою. Цю конструкцію можна використовувати для якогось певного продукту, розміщуючи окремі записи з описом його переваги / властивості або сукупностей характеристик. Всі записи тут пов'язані між собою і посилаються на головну, що дозволяє ефективно її просувати. Однак даний підхід вкрай специфічний і підходить далеко не для всіх інтернет-майданчиків [24]. Приклад мережевої схеми веб-ресурсу представлений на рисунку 3.4.

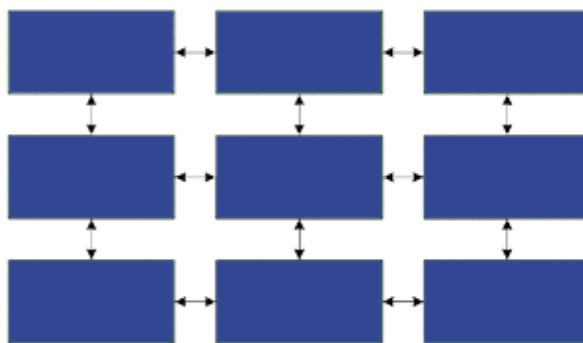


Рисунок 3.4 – Схема мережевої структури

Комбінована структура поєднує кілька структур, тобто веб-ресурс може поєднувати в собі лінійну та ієрархічну структури одночасно.

Для інтернет-магазину «Agrohelf» була обрана мережева структура, схема якої представлена на рисунку 3.5.

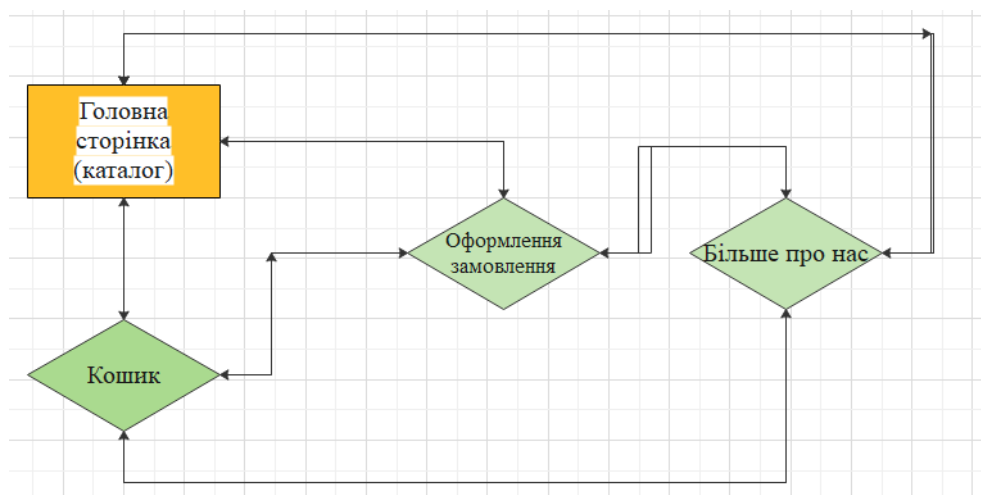


Рисунок 3.5 – Внутрішня структура інтернет-магазину

3.2 Проектування та реалізація бази даних

Після того, як було визначено зовнішню та внутрішню структуру інтернет-магазину, необхідно було вирішити наступне питання – спосіб виведення та збору інформації інтернет-магазину: перелік товарів, інформація про них, інформація про замовлення. Для повноцінної та коректної роботи інтернет-магазину необхідно спроектувати бази даних та виконати її підключення до

інтернет-магазину. База даних інтернет-магазину «Agrohelf» складається із таких таблиць: brands, categories, clients, details, goods, orders. На рисунку 3.6 зображена структура бази даних інтернет-магазину.

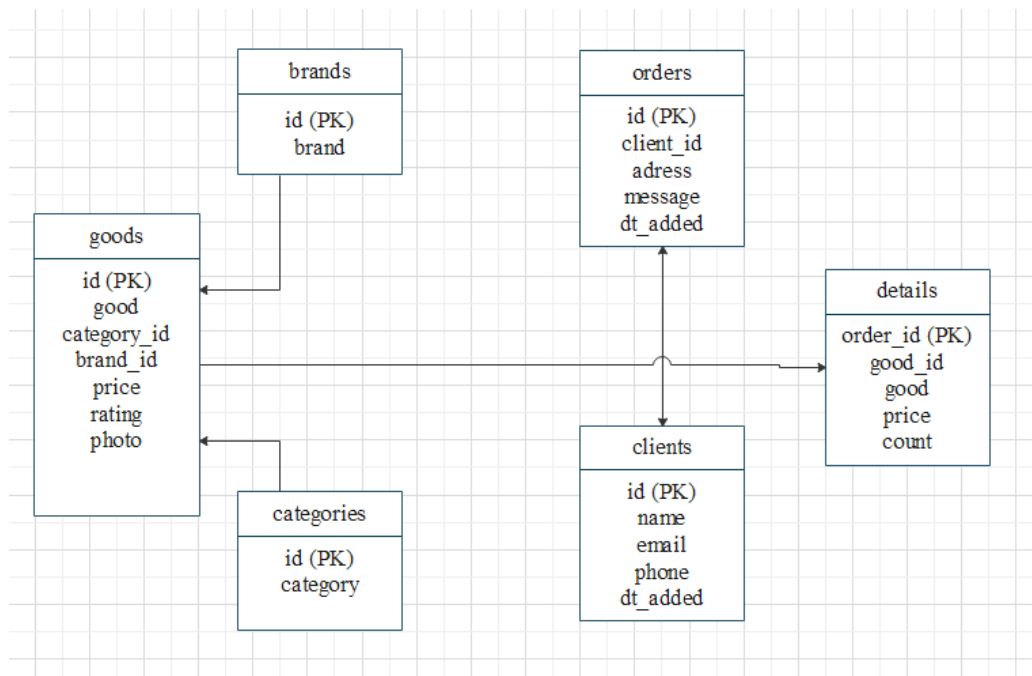


Рисунок 3.6 – Схема бази даних інтернет-магазину «Agrohelf»

Після проектування бази даних була виконана її реалізація засобами MySQL. На рисунку 3.7 представлено таблиці бази даних «agrohelfdata.sql» для інтернет-магазину «Agrohelf»

Таблиця	Действие	Строки	Тип	Сравнение	Размер	Фрагментировано
<input type="checkbox"/> brands	Обзор Структура Поиск Вставить Очистить Удалить	1	InnoDB	utf8_general_ci	16 КБ	-
<input type="checkbox"/> categories	Обзор Структура Поиск Вставить Очистить Удалить	3	InnoDB	utf8_general_ci	16 КБ	-
<input type="checkbox"/> clients	Обзор Структура Поиск Вставить Очистить Удалить	2	InnoDB	utf8_general_ci	16 КБ	-
<input type="checkbox"/> details	Обзор Структура Поиск Вставить Очистить Удалить	2	InnoDB	utf8_general_ci	16 КБ	-
<input type="checkbox"/> goods	Обзор Структура Поиск Вставить Очистить Удалить	6	InnoDB	utf8_general_ci	48 КБ	-
<input type="checkbox"/> orders	Обзор Структура Поиск Вставить Очистить Удалить	2	InnoDB	utf8_general_ci	16 КБ	-
6 таблиц	Всего	16	InnoDB	utf8_general_ci	128 КБ	0 Байт

Рисунок 3.7 – База даних інтернет-магазину

На рисунку 3.8 представлена ER модель бази даних «agrohelfdata.sql», створеної за допомогою Дизайнера phpMyAdmin.

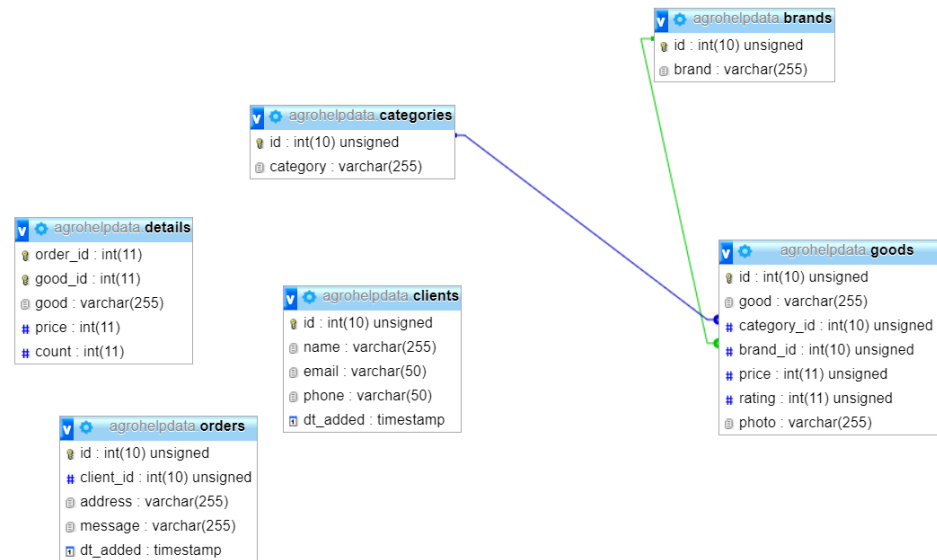


Рисунок 3.8 – ER модель бази даних

3.3 Розробка інтерфейсу

Як і будь-який магазин в інтернет-просторі, інтернет-магазин «Agrohelp» повинен мати власний інтерфейс. Веб-сервіси та системи суттєво спрощують роботу людей. Наприклад, Google Drive, що дозволяє в мережі інтернет зберігати файли, передавати на них посилання і таким чином не передавати флеш-накопичувачі з потрібними даними. Але складний інтерфейс не вплине на спрощення роботи.

Виходить, що програмний інтерфейс не тільки вирішує проблему взаємодії з додатком, а й робить цю взаємодію максимально комфортною. Існує важлива наявність інтерфейсу, що дозволяє при меншій кількості зусиль ознайомитися з можливостями програми та зрозуміти принципи роботи [25].

Головна сторінка «Agrohelp» має мінімалістичний дизайн і максимально просте представлення сайту.

AGROHELP

Магазин добрив. №1 в Україні



[Каталог](#)
[Кошик 2](#)
[Оформлення замовлення](#)
[Більше про нас](#)

[Всі категорії](#)
[Ягоди](#)
[Квіти](#)
[Дерева](#)

Товарів на сторінці:

[1](#)
[2](#)
[3](#)
[>](#)
[»](#)

Показані товари: 1 - 2 из 6

	<p>Артикул 4 Agrohelp4 12 грн</p> <p>Додати в кошик</p>
	<p>Артикул 2 Agrohelp2 13 грн</p> <p>Додати в кошик</p>

<p>Навігація на сайті</p> <p> Каталог Кошик Оформлення замовлення Більше про нас </p>	<p>Контактна інформація:</p> <p> +3801234567 (Vodafone) +38012345678 (Kyivstar) +38012345679 (Lifecell) agrohelp (Skype) </p> <p style="text-align: right;">AGROHELP</p>
---	---

Рисунок 3.9 – Головна сторінка інтернет-магазину

Завдяки JavaScript та CSS реалізовано елементи навігації в інтернет-магазині. За допомогою CSS та HTML були створені кнопки з посиланнями на

сторінки другого рівня. При натисканні на кнопку ми переходимо на інший розділ сайту.

У верхній частині головної сторінки розташована «шапка» сайту, яка дублюється на всіх інших сторінках. Вона створюється за допомогою шаблонів HTML та CSS. «Шапка» інтернет-магазину має простий дизайн та спокійні, трендові кольори.

Для швидкого переходу на головну сторінку достатньо натиснути на логотип сайту, а для переходу до основних розділів було створено меню у вигляді списку гіперпосилань на інші сторінки сайту. Кожне таке посилання називається пунктом меню. Це важливий елемент сайту, так як користувач постійно звертається до нього і тому вимоги до нього дуже високі.

Панель навігації сайту складається з наступних сторінок: «Каталог», «Кошик», «Оформлення замовлення», «Більше про нас». Кожен пункт, при наведенні мишею, змінює колір (рис. 3.10).

AGROHELP

Магазин добрив. №1 в Україні

[Каталог](#) [Кошик 2](#) [Оформлення замовлення](#) [Більше про нас](#)

Рисунок 3.10 – Панель навігації інтернет-магазину

Для досягнення цього ефекту були використані властивості каскадних таблиць стилів, що представлені в таблиця 3.1.

Для перевірки кошика та оформлення замовлення, шапка сайту містить посилання на відповідні форми, які були реалізовані засобами HTML, CSS, JavaScript на стороні клієнта та PHP на стороні сервера.

Сторінка «Кошик» показує користувачу, які товари він обрав для замовлення. На цій сторінці можна побачити загальну вартість замовлення, а також визначити, скільки одиниць одного товару йому потрібно. Після цього, він

натискає на кнопку «Оформити замовлення», яка перенаправляє його на форму для замовлення (рисунок 3.11).

Таблиця 3.1 – Властивості CSS для меню

Властивість	Пояснення
hover	визначає стиль елемента, коли користувач наводить на нього курсора миші, але не обов'язково активованує його, тобто не було натиснута кнопка миші.
transition	встановлює ефект переходу між двома станами елемента, вони можуть бути визначені за допомогою псевдокласу :hover або :active, а також динамічно через JavaScript.
color	визначає колір тексту елемента.
padding	встановлює значення полів навколо вмісту елемента.

Сторінка «Оформлення замовлення» просить від користувача дані, які потрібні для того, щоб визначити місце надсилання замовлення та як зв'язатися з покупцем у випадку, якщо є питання (рис. 3.12).

На головній сторінці існують кнопки для керування каталогом. Користувач може обрати кількість товарів, яка буде показуватись на одній сторінці та зручний спосіб подання інформації про товари. У кожного товару є кнопка «Додати в кошик», яка виконує функцію додавання товару до кошику для подальшого замовлення. На рисунку 3.13 представлені дані функціональні елементи.

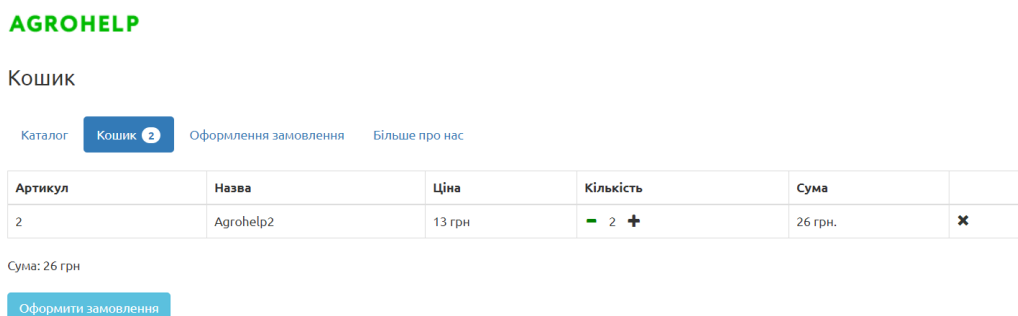


Рисунок 3.11 – Кошик інтернет-магазину

У вашому кошику 2 товар(ів) на загальну суму 26 гривень. Заповніть форму нижче, щоб оформити замовлення.

Ваше ім'я *

Email *

Телефон

Спосіб доставки

Самовивіз (безкоштовно)

В межах міста

Доставка Новою Поштою

Сума доставки 0 гривень. Загальна сума замовлення: 26 + 0 = 26 гривень

Адреса доставки

Коментар

[Надіслати замовлення](#)

Рисунок 3.12 – Форма оформлення замовлення

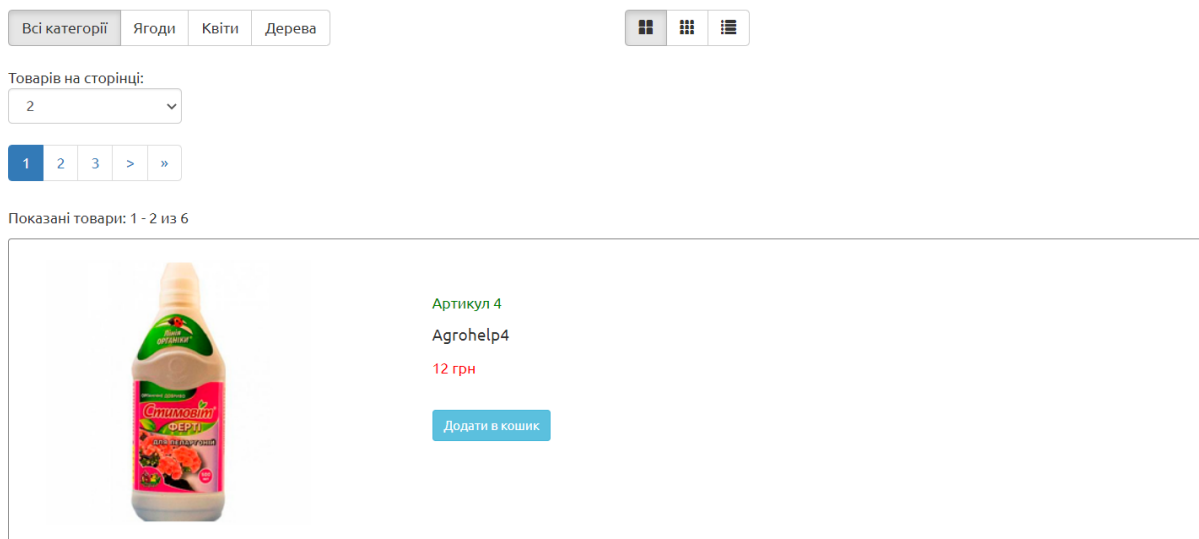


Рисунок 3.13 – Елементи навігації в каталозі

На кожній сторінці присутній так званий «підвал» сайту (footer), що був створений за допомогою HTML та CSS. У ньому дублюються логотип та кнопки

навігації на сайті, а також додано контактну інформацію інтернет магазину. У перспективі до «підвалу» можна додати кнопки, які перенаправлятимуть на соціальні мережі. На рисунку 3.14 представлений footer інтернет-магазину.

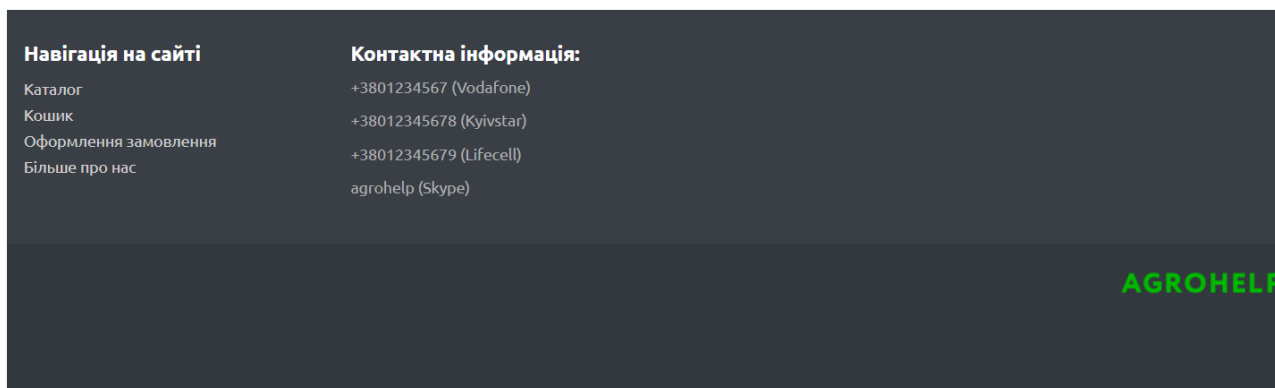


Рисунок 3.14 – footer сайту

У Додатку В представлений супровідний код для реалізації всіх тих процес, що були описані раніше.

3.4 Розробка програмного коду

Інтерактивність формам надає JavaScript, а обробку вхідних даних виконує PHP, після чого повертає результат клієнту та зберігає дані про покупців. Також на всіх сторінках доданий `list-unstyled` – це анімація завантаження каталогу, яка відображається до того моменту, поки сторінка не буде повністю готова для роботи з клієнтом. Фрагменти коду, відповідальні за анімацію завантаження, наведено нижче:

```
//HTML
```

```
<ul id="goods" class="list-unstyled">
  
</ul>
```

```
// Bootstrap
```

```
.list-unstyled{padding-left:0;list-style:none }
```

У каталозі для відображення товарів, відповідно, була використана технологія AJAX. Приведений нижче фрагмент коду відповідає за відправку запиту на сервер та отримання відповіді від нього:

```
function _getData(options) {
    var resetPage = options && options.resetPage,
        options = _getOptions(resetPage);
    $.ajax({
        url: 'scripts/catalog_pag.php',
        data: options,
        type: 'GET',
        cache: false,
        dataType: 'json',
        success: function(response) {
            if (response.code === 'success') {
                _renderCatalog(response.data.goods);
                _renderPagination({
                    page: options.page,
                    limit: options.limit,
                    countAll: response.data.countAll,
                    countItems: response.data.goods.length
                });
            } else {
                console.error('Відбулась помилка');
            }
        }
    });
}

function _renderCatalog(goods) {
    var theme = $('js-theme.active').attr('data-theme');
    ui.$goods.html(goodsTemplate[theme]({ goods: goods }));
}

function _renderPagination(options) {
    var countAll = options.countAll,
        countItems = options.countItems,
        page = options.page,
        limit = options.limit,
        countPages = Math.ceil(countAll / limit),
        start = (page - 1) * limit + 1,
        end = start + countItems - 1;
```

```
var goodsInfoMsg = start + ' - ' + end + ' з ' + countAll;
ui.$goodsInfo.text(goodsInfoMsg);

ui.$pag.html(pagTemplate({
  page: page,
  countPages: countPages
}));
}
```

Після того, як інтернет-магазин був розроблений та протестований, було сформовано інструкцію користувача (Додаток Г) та інструкцію адміністратора (Додаток Д).

ВИСНОВКИ

Під час виконання випускної роботи «Клієнтська частина інтернет-магазину за допомогою JavaScript» був розроблений сучасний інтернет-магазин. Інтернет-магазин був розроблений до вимог, що були зазначені у технічному завданні:

- наявність адаптивної верстки інтернет-магазину;
- можливість перегляду наявного асортименту та його замовлення;
- можливість управління майбутніми замовленнями;
- наявність форми оформлення замовлення;
- наявність розділу «Більше про нас».

Було проведено детальний аналіз предметної області: визначено питання актуальності проблеми, проведено аналіз інтернет-магазинів аналогів, сформовано мету та задачі для реалізації продукту, проведено аналіз існуючих технологій розробки. Технічне завдання до інтернет-магазину представлено у Додатку А.

Також було проведено проектування інтернет-магазину, що дозволило повноцінно та детально вивчити процес розробки інтернет-магазинів з використанням діаграм нотатції IDEF0. На даному етапі роботи було розроблено діаграму варіантів використання, що відображає головних акторів системи та їх сценарії дій з інтернет-магазином. Розробником проекту було проведено планування робіт, що дозволило визначити ключові етапи робіт, їх часові рамки виконання, ризики та сценарії їх мінімізації.

Після того, як був реалізований інтернет-магазин, було проведено його тестування. Результати даного етапу описані нижче:

- адаптивність верстки працює коректно;
- форма кошику та оформлення замовлення містить модулі верифікації даних та має властивість запам'ятовувати дані користувача;
- отримання інформації про замовлення, клієнтів та представлення інформації про товари в каталозі працює коректно.

СПИСОК ЛІТЕРАТУРИ

1. Тенденции развития веб-технологий в 2019 году | PowerCode Academy [Електронний ресурс] – режим доступу: <https://cutt.ly/LtzwzZM>
2. Інтернет магазин UkrSemena [Електронний ресурс] – режим доступу: <https://ukrsemena.com>
3. Інтернет-магазин «Професійні насіння» [Електронний ресурс] – режим доступу: <https://semena.in.ua>
4. Інтернет-магазин Agreemarket [Електронний ресурс] – режим доступу: <https://agreemarket.com.ua>
5. What is HTML? | HyperText Markup Language explained [Електронний ресурс] – режим доступу: <https://bit.ly/2Yw2V84>
6. What is CSS, How Does It Work and What is It Used For? – Skillcrush [Електронний ресурс] – режим доступу: <https://bit.ly/3e3ePg1>
7. Преимущества стилей | htmlbook.ru [Електронний ресурс] – режим доступу: <https://bit.ly/37xbTGe>
8. What is JavaScript? - Learn web development | MDN [Електронний ресурс] – режим доступу: <https://bit.ly/2MVGLGL>
9. javascript.info [Електронний ресурс] – режим доступу: <https://bit.ly/3e1OQpn>
10. Офіційна сторінка jQuery [Електронний ресурс] – режим доступу: <https://jquery.com>
11. Стаття про Bootstrap в енциклопедії «Вікіпедія» [Електронний ресурс] – режим доступу: <https://uk.wikipedia.org/wiki/Bootstrap>
12. PHP: What is PHP? – Manual [Електронний ресурс] – режим доступу: <https://bit.ly/3flKqK1>
13. What is PHP? Write your first PHP Program [Електронний ресурс] – режим доступу: <https://bit.ly/3hu20gM>
14. What is MySQL? - Definition from WhatIs.com [Електронний ресурс] – режим доступу: <https://bit.ly/2MVVpOb>

15. What are the advantages and disadvantages of using MySQL stored procedures? [Электронный ресурс] – режим доступа: <https://bit.ly/2B3tqcX>

16. What is AJAX, How Does It Work, and What Is It Used For? – Skillcrush [Электронный ресурс] – режим доступа: <https://bit.ly/3e30UXd>

17. What is AJAX? – Tutorialspoint [Электронный ресурс] – режим доступа: <https://cutt.ly/Otzw1sk>

18. Постановка целей по SMART (СМАРТ) на примерах [Электронный ресурс] – режим доступа: <https://bit.ly/37AJzCQ>

19. Преимущества Иерархической Структуры Работ (WBS) для менеджеров ИТ проектов / Хабр [Электронный ресурс] – режим доступа: <https://bit.ly/2Y1eQM8>

20. Организационная структура компании [Электронный ресурс] – режим доступа: <https://cutt.ly/Stzw3Dz>

21. Как диаграммы Ганта упрощают работу с проектами / Блог компании Nygger / Хабр [Электронный ресурс] – режим доступа: <https://bit.ly/2XZIE5j>

22. Знакомство с нотацией IDEF0 и пример использования / Блог компании Trinion / Хабр [Электронный ресурс] – режим доступа: <https://bit.ly/37xrnd5>

23. UML — диаграмма вариантов использования (use case diagram) / Хабр [Электронный ресурс] – режим доступа: <https://bit.ly/30EizRc>

24. Структура сайта: схемы, советы по разработке, примеры [Электронный ресурс] – режим доступа: <https://bit.ly/2Yw3ONW>

25. Проектирование пользовательских интерфейсов: что это такое и зачем его заказывать — Блог Live Turing [Электронный ресурс] – режим доступа: <https://bit.ly/2zwDHxR>

ДОДАТОК А ТЕХНІЧНЕ ЗАВДАННЯ

на розробку інтернет-магазину «Agrohelf»

Назва інтернет-магазину «Agrohelf».

Область застосування інтернет-магазину – інтернет-магазин з продажу добрив можуть використовувати як рядові фермери, так і агрономічні підприємства, що мають врожаї.

Об’єкт, у якому використовують інтернет-магазин – агрономічна сфера

1 Основи для розробки

Розробка виконується на основі завдання, виданого викладачами секції Інформатика.

2 Призначення розробки

Розробка повинна надати користувачу можливість зручно придбати добрива.

Назва організації: Прізвище І.Б.

Тема проекту: «Клієнтська частина інтернет-магазину з продажу добрив за допомогою JavaScript»

3 Вимоги до веб-ресурсу

Веб-ресурс повинен бути реалізований у вигляді адаптивного сайту з наявністю форми оформлення замовлення та повинен забезпечувати виконання функціональних можливостей, визначених у пункті 3.2.

3.1 Вимоги до веб-ресурсу

Інтернет-магазин розробляється ітеративно із врахуванням принципів та технологій уніфікованого процесу розроблення сайту. Каркас має бути реалізований мовою HTML з підключенням до CSS. Функції, перелічені в пункті 3.3, мають бути написані мовою JavaScript з використанням технології AJAX.

3.2 Вимоги до функціональних характеристик

Інтернет-магазин повинен забезпечувати виконання наступних функцій:

- надання переліку асортименту добрив;
- вибір потрібних добрив та додавання їх у кошик;

- можливість оформлення замовлення;

4 Перелік програмної документації

- опис проекту продукту із використанням UML-діаграм;
- логічна схема роботи інтернет-магазину;
- програмний код розробки;
- технічне завдання;

5 Порядок виконання робіт і етапи розробки

Стадії та етапи розробки повинні складатися з наступних пунктів:

- оформлення завдання для дипломної роботи;
- планування роботи: розроблення ТЗ, побудова мережевого графіку та діаграми Ганта;
- розроблення логічної схеми інтернет-магазину;
- реалізація інтернет-магазину;
- розроблення логіки роботи інтернет-магазину;
- провести тестування розробки;
- оформлення пояснювальної записки про виконання дипломної роботи;
- задача пояснювальної записки до дипломної роботи та розробленого інтернет-магазину;
- презентація роботи та її захист.

6 Порядок контролю та приймання

Контроль коректності функціонування та придатності веб-ресурсу здійснюється замовником (секцією Інформаційні технології проектування) на основі наданої пояснювальної записки до дипломної роботи та програмних файлів. Контроль ходу виконання проекту здійснюється на основі календарного плану виконання дипломної роботи:

1. Перевірка завдання дипломної роботи.
2. Перевірка ТЗ, мережевого графіка та діаграми Ганта.
3. Перевірка логічної схеми для інтернет-магазину.
4. Перевірка веб-сторінки.

5. Перевірка коректності логіки роботи інтернет-магазину.
6. Перевірка коректності тестування.
7. Здача ПЗ.
8. Презентація.

ДОДАТОК Б

ПЛАНУВАННЯ РОБІТ

Ідентифікація мети ІТ-проекту методом SMART

Результатом випускної роботи є інтернет-магазин «Agrohelf». Для визначення/ідентифікації мети був використаний відомий метод SMART. SMART є сучасним підходом до визначення та постановки бажаних цілей, надає можливість на етапі вибору мети зібрати всю наявну інформацію, визначити терміни роботи, кількість потрібних матеріалів для досягнення мети та структурувати всю інформацію для подальшої роботи команди.

SMART розшифровується як Specific, Measurable, Achievable, Relevant, Time bound. Кожна буква аббревіатури визначає критерій важливості цілей для досягнення мети. Далі ми розглянемо критерії smart більш детально [20].

Результати ідентифікації мети проекту відображено в таблиці Б.1.

Таблиця Б.1 – Ідентифікація проекту

Specific	Створення інтернет-магазину «Agrohelf» та розміщення його в інтернет-просторі.
Measurable	Спрощення придбання добрив виробника «Agrohelf» завдяки простій навігації по сайту та його швидкодії за рахунок новітніх технологій.
Achievable	Реалізація проекту виконується з використанням технології HTML, CSS, JS, PHP, MySQL, AJAX.
Relevant	Для реалізації проекту розробник має достатню кваліфікацію та навички роботи з технологіями.
Time bound	Інтернет-магазин має бути розроблено згідно календарного плану.

Опис фази розробки ІТ-проекту

Ті, хто мають справу зі складними ІТ проектами керівники підтвердять, що поділ завдань на більш дрібні і керовані частини робить робочий процес набагато простіше. Йдеться про ієрархічну структуру робіт WBS (Work Breakdown Structure).

WBS допомагає проектним менеджерам розглянути різні сценарії при виконанні цілей та досягненні мети. Також описується як структура відгалуження, яка структурує етапи проекту в певному порядку. WBS може мати вигляд таблиці завдань і елементів в плані диференціації робіт діаграм Ганта [21]. На рисунку. Б.1 представлена розроблена WBS-діаграма проекту «Agrohelf».

Планування структури організації, для впровадження готового проекту

Організаційна структура проекту або OBS являє собою ієрархічну структуру керування проектом і відображає відносини в команді.

Організаційна структура проекту:

- розробляється на рівні підприємства;
- елементи визначаються на рівні EPS
- дозволяє мати контроль до доступу користувачів до інформації

певного рівня [22].

Розроблена діаграма OBS проекту представлена на рисунку Б.2.

Побудова календарного графіку виконання ІТ – проекту

Останнім етапом в плануванні проекту є розробка календарного плану у вигляді діаграма Ганта та формування PDM-мережі.

Діаграма Ганта є графічним способом відображення планів. Використання графіків практикується для планування проектів в різноманітних сферах. Це зручний спосіб візуалізувати терміни робіт. Gantt Charts також допомагають командам і менеджерам проектів контролювати дати початку і закінчення будь-якого проекту [23].

На рисунку Б.3 представлена розроблена діаграма Ганта. Список робіт із зазначення дат початку та завершення представлений на рисунку Б.4. На рисунках Б.5-Б.8 представлена PDM-мережа



Рисунок Б.1 – WBS-діаграма дипломного проекту



Рисунок Б.2 – WBS-діаграма дипломного проекту

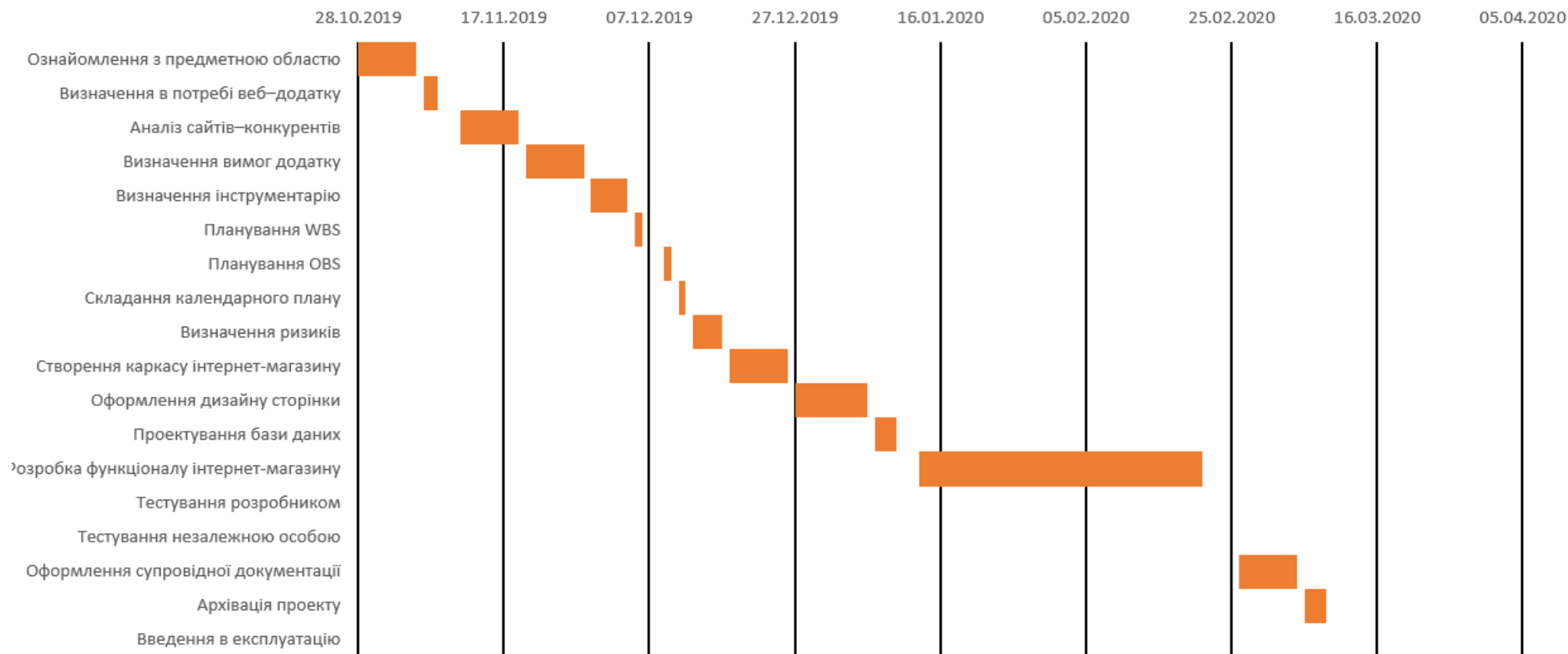


Рисунок Б.3 – Діаграма Ганта для проекту «Agrohelp»

Task	Start Date	End Date	Duration
Ознайомлення з предметною областю	28.10.2019	05.11.2019	8
Визначення в потребі веб-додатку	06.11.2019	08.11.2019	2
Аналіз сайтів-конкурентів	11.11.2019	19.11.2019	8
Визначення вимог додатку	20.11.2019	28.11.2019	8
Визначення інструментарію	29.11.2019	04.12.2019	5
Планування WBS	05.12.2019	06.12.2019	1
Планування OBS	09.12.2019	10.12.2019	1
Складання календарного плану	11.12.2019	12.12.2019	1
Визначення ризиків	13.12.2019	17.12.2019	4
Створення каркасу інтернет-магазину	18.12.2019	26.12.2019	8
Оформлення дизайну сторінки	27.12.2019	06.01.2020	10
Проектування бази даних	07.01.2020	10.01.2020	3
Розробка функціоналу інтернет-магазину	13.01.2020	21.02.2020	39
Тестування розробником	24.02.2020	24.02.2020	0
Тестування незалежною особою	25.02.2020	25.02.2020	0
Оформлення супровідної документації	26.02.2020	05.03.2020	8
Архівація проекту	06.03.2020	09.03.2020	3
Введення в експлуатацію	10.03.2020	10.03.2020	0

Рисунок Б.4 – Список робіт для діаграми Ганта



Рисунок Б.5 – PDM-мережа проекту

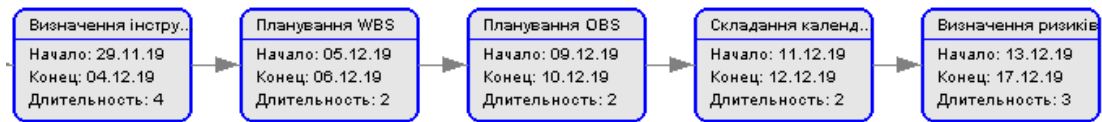


Рисунок Б.6 – Продовження PDM-мережі

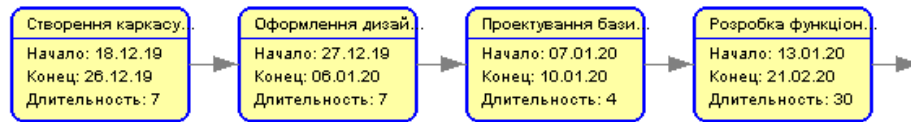


Рисунок Б.7 – Продовження PDM-мережі

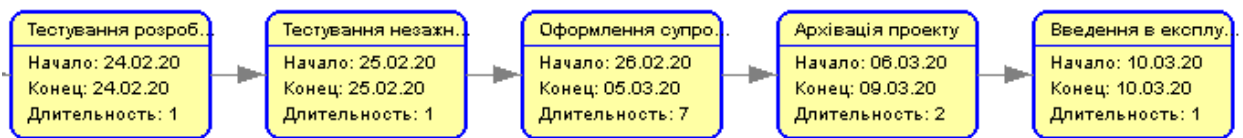


Рисунок Б.8 – Продовження PDM-мережі

Керування ризиками. Ризик є ймовірнісною подією, яка може мати різний вплив: як позитивний, так і негативний. Процес ідентифікації ризиків дозволяє визначити такі події, що здатні вплинути на хід виконання проекту та його реалізацію.

Ідентифікацію ризиків потрібно проводити регулярно з метою ефективної оцінки різних сценаріїв та їх наслідків.

Класифікація ризиків:

1. За ймовірністю виникнення:
 - слабо ймовірнісні;
 - мало ймовірнісні;
 - ймовірні;
 - досить ймовірні;
 - майже ймовірні.
2. За величиною втрат:
 - мінімальна;
 - низька;
 - середня;

- висока;
- максимальна.

На основі цих даних була проведена класифікація ризиків для даного проекту, що наведена в таблиці Б.2.

Таблиця Б.2 – Класифікація ризиків

№	Назва ризику	Ймовірність	Величина втрат
1	Некоректно складене ТЗ	2	4
2	Недотримання календарного плану	1	3
3	Некоректна робота програмного забезпечення	4	4
4	Некоректна робота апаратного забезпечення	4	4
5	Хвороба розробника	2	2
6	Некоректне тестування	2	1
7	Пошкодження файлу	4	5

Використовуючи дану класифікацію, була побудована матриця ризиків, що представлена на рисунку Б.9.

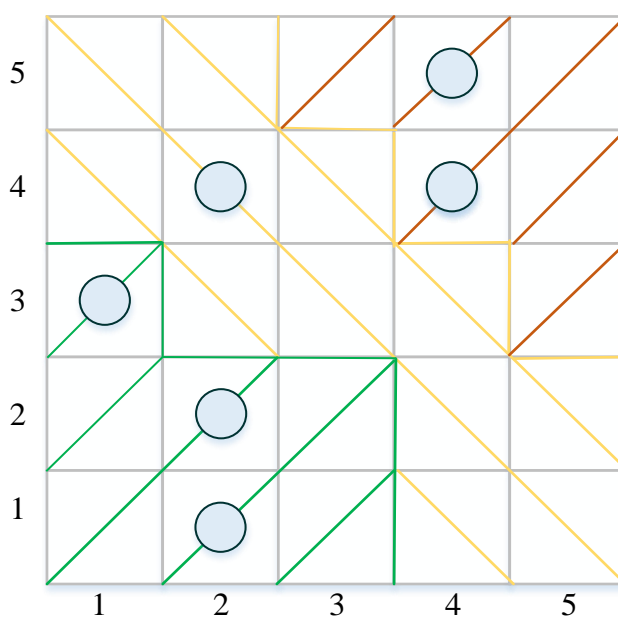


Рисунок Б.9 – Матриця ризиків

ДОДАТОК В

ФАЙЛИ ПРОГРАМНОГО КОДУ

Main.js

```
'use strict';

var app = (function($) {
  var $body = $('body'),
      page = $body.data('page'),
      options = {
        elAddToCart: '.js-add-to-cart',
        attrId: 'data-id',
        attrName: 'data-name',
        attrPrice: 'data-price',
        attrDelta: 'data-delta',
        elCart: '#cart',
        elTotalCartCount: '#total-cart-count',
        elTotalCartSumma: '#total-cart-summa',
        elCartItem: '.js-cart-item',
        elCartCount: '.js-count',
        elCartSumma: '.js-summa',
        elChangeCount: '.js-change-count',
        elRemoveFromCart: '.js-remove-from-cart'
      },
      optionsCatalog = _.extend({
        renderCartOnInit: false,
        renderMenuCartOnInit: true
      }, options),
      optionsCart = _.extend({
        renderCartOnInit: true,
        renderMenuCartOnInit: true
      }, options),
      optionsOrder = _.extend({
        renderCartOnInit: false,
        renderMenuCartOnInit: true
      }, options);

  function init() {
    if (page === 'catalogPag') {
      catalogPag.init();
    }
  }
}
```

```

        cart.init(optionsCatalog);
    }
    if (page === 'cart') {
        cart.init(optionsCart);
    }
    if (page === 'order') {
        order.init();
        cart.init(optionsOrder);
    }
}
return {
    init: init
}
})(jQuery);

```

```
jQuery(document).ready(app.init);
```

Cart.js

```

'use strict';
var cart = (function($) {
    var cartData,
        opts = {};

    function init(options) {
        _initOptions(options);
        updateData();
        if (opts.renderCartOnInit) {
            renderCart();
        }
        if (opts.renderMenuCartOnInit) {
            renderMenuCart();
        }
        _bindHandlers();
    }

    function _initOptions(options) {
        var defaultOptions = {
            renderCartOnInit: true,
            renderMenuCartOnInit: true,
            elAddToCart: 'js-add-to-cart',
            attrId: 'data-id',

```

```

    attrName: 'data-name',
    attrPrice: 'data-price',
    attrDelta: 'data-delta',
    elCart: '#cart',
    elTotalCartCount: '#total-cart-count',
    elTotalCartSumma: '#total-cart-summa',
    elCartItem: '.js-cart-item',
    elCartCount: '.js-count',
    elCartSumma: '.js-summa',
    elChangeCount: '.js-change-count',
    elRemoveFromCart: '.js-remove-from-cart'
  };
  _.defaults(options || {}, defaultOptions);
  opts = _.clone(options);
}
function _bindHandlers() {
  _onClickAddBtn();
  _onClickChangeCountInCart();
  _onClickRemoveFromCart();
}
function updateData() {
  cartData = JSON.parse(localStorage.getItem('cart')) || [];
  return cartData;
}
function getData() {
  return cartData;
}
function saveData() {
  localStorage.setItem('cart', JSON.stringify(cartData));
  return cartData;
}
function clearData() {
  cartData = [];
  saveData();
  return cartData;
}
function getById(id) {
  return _.findWhere(cartData, {id: id});
}

```

```
}

function add(item) {
  var oldItem;
  updateData();
  oldItem = getById(item.id);
  if(!oldItem) {
    cartData.push(item);
  } else {
    oldItem.count = oldItem.count + item.count;
  }
  saveData();
  return item;
}

function remove(id) {
  updateData();
  cartData = _.reject(cartData, function(item) {
    return item.id === id;
  });
  saveData();
  return cartData;
}

function changeCount(id, delta) {
  var item;
  updateData();
  item = getById(id);
  if(item) {
    item.count = item.count + delta;
    if (item.count < 1) {
      remove(id);
    }
    saveData();
  }
  return _.findWhere(cartData, {id: id}) || {};
}

function getCount() {
  return _.size(cartData);
}
```

```

function getCountAll() {
    return _.reduce(cartData, function(sum, item) {return sum + item.count}, 0);
}
function getSumma() {
    return _.reduce(cartData, function(sum, item) {return sum + item.count * item.price}, 0);
}
function renderCart() {
    var template = _.template($('#cart-template').html()),
        data = {
            goods: cartData
        };
    $(opts.elCart).html(template(data));
    renderTotalCartSumma();
}
function renderMenuCart() {
    var countAll = getCountAll();
    $(opts.elTotalCartCount).html(countAll !== 0 ? countAll : '');
}
function renderTotalCartSumma() {
    $(opts.elTotalCartSumma).html(getSumma());
}

function findCartElemById(id) {
    return $(opts.elCartItem + '[' + opts.attrId + '=' + id + ']');
}
function _onClickAddBtn() {
    $('body').on('click', opts.elAddToCart, function(e) {
        var $this = $(this);
        add({
            id: +$this.attr(opts.attrId),
            name: $this.attr(opts.attrName),
            price: +$this.attr(opts.attrPrice),
            count: 1
        });
        renderMenuCart();
        alert('Товар додано в кошик');
    });
}

```

```

    });
}
function _onClickChangeCountInCart() {
    $('body').on('click', opts.elChangeCount, function(e) {
        var $this = $(this),
            id = +$this.attr(opts.attrId),
            delta = +$this.attr(opts.attrDelta),
            $cartElem = findCartElemById(id),
            cartItem = changeCount(id, delta);
        if (cartItem.count) {
            $cartElem.find(opts.elCartCount).html(cartItem.count);
            $cartElem.find(opts.elCartSumma).html(cartItem.count * cartItem.price);
        } else {
            $cartElem.remove();
        }
        renderMenuCart();
        renderTotalCartSumma();
    });
}
function _onClickRemoveFromCart() {
    $('body').on('click', opts.elRemoveFromCart, function(e) {
        if(!confirm('Видалити товар з кошику?')) return false;
        var $this = $(this),
            id = +$this.attr(opts.attrId),
            $cartElem = findCartElemById(id);
        remove(id);
        $cartElem.remove();
        renderMenuCart();
        renderTotalCartSumma();
    });
}
return {
    init: init,
    update: updateData,
    getData: getData,
    save: saveData,
    clearData: clearData,
    getById: getById,

```

```

    add: add,
    remove: remove,
    changeCount: changeCount,
    getCount: getCount,
    getCountAll: getCountAll,
    getSumma: getSumma
  }
})(jQuery);

```

CatalogPag.js

```

'use strict';
var catalogPag = (function($) {
  var ui = {
    $categoryBtn: $('.js-category'),
    $themeBtn: $('.js-theme'),
    $limit: $('#pages-limit'),
    $pag: $('#pagination'),
    $goods: $('#goods'),
    $goodsInfo: $('#goods-info')
  };
  var goodsTemplate = {
    big: _.template($('#goods-template-big').html()),
    compact: _.template($('#goods-template-compact').html()),
    list: _.template($('#goods-template-list').html())
  },
  pagTemplate = _.template($('#pagination-template').html());
  function init() {
    _setTheme();
    _getData({
      resetPage: true
    });
    _bindHandlers();
  }
  function _setTheme() {
    var theme = localStorage.getItem('theme') || 'compact';
    $('.js-theme[data-theme="' + theme + '"]').addClass('active');
  }
  function _bindHandlers() {
    ui.$categoryBtn.on('click', _changeCategory);
  }

```



```

    ui.$themeBtn.on('click', _changeTheme);
    ui.$limit.on('change', _changeLimit);
    ui.$pag.on('click', 'a', _changePage);
  }
  function _changeCategory(e) {
    var $category = $(e.target);
    ui.$categoryBtn.removeClass('active');
    $category.addClass('active');
    _getData({
      resetPage: true
    });
  }
  function _changeTheme(e) {
    var $theme = $(e.target).closest('button'),
        theme = $theme.attr('data-theme');
    ui.$themeBtn.removeClass('active');
    $theme.addClass('active');
    _getData({
      resetPage: false
    });
    localStorage.setItem('theme', theme);
  }
  function _changeLimit() {
    _getData({
      resetPage: true
    });
  }
  function _changePage(e) {
    e.preventDefault();
    e.stopPropagation();
    var $page = $(e.target).closest('li');
    ui.$pag.find('li').removeClass('active');
    $page.addClass('active');
    _getData();
  }
  function _getOptions(resetPage) {
    var categoryId = +$('.js-category.active').attr('data-category'),
        page = !resetPage ? +ui.$pag.find('li.active').attr('data-page') : 1,

```

```

        limit = +ui.$limit.val();
    return {
        category: categoryId,
        page: page,
        limit: limit
    }
}
function _getData(options) {
    var resetPage = options && options.resetPage,
        options = _getOptions(resetPage);
    $.ajax({
        url: 'scripts/catalog_pag.php',
        data: options,
        type: 'GET',
        cache: false,
        dataType: 'json',
        success: function(response) {
            if (response.code === 'success') {
                _renderCatalog(response.data.goods);
                _renderPagination({
                    page: options.page,
                    limit: options.limit,
                    countAll: response.data.countAll,
                    countItems: response.data.goods.length
                });
            } else {
                console.error('Відбулась помилка');
            }
        }
    });
}
function _renderCatalog(goods) {
    var theme = $('<div>.js-theme.active</div>').attr('data-theme');
    ui.$goods.html(goodsTemplate[theme]({goods: goods}));
}
function _renderPagination(options) {
    var countAll = options.countAll,
        countItems = options.countItems,

```

```

        page = options.page,
        limit = options.limit,
        countPages = Math.ceil(countAll / limit),
        start = (page - 1) * limit + 1,
        end = start + countItems - 1;
var goodsInfoMsg = start + ' - ' + end + ' z ' + countAll;
ui.$goodsInfo.text(goodsInfoMsg);
ui.$pag.html(pagTemplate({
    page: page,
    countPages: countPages
}));
}
return {
    init: init
}
})(jQuery);

```

Order.js

```

'use strict';
var order = (function($) {
    var ui = {
        $orderForm: $('#order-form'),
        $messageCart: $('#order-message'),
        $orderBtn: $('#order-btn'),
        $alertValidation: $('#alert-validation'),
        $alertOrderDone: $('#alert-order-done'),
        $orderMessageTemplate: $('#order-message-template'),
        $fullSumma: $('#full-summa'),
        $delivery: {
            type: $('#delivery-type'),
            summa: $('#delivery-summa'),
            btn: $('.js-delivery-type'),
            alert: $('#alert-delivery')
        }
    };
    var freeDelivery = {
        enabled: false,
        summa: 10000
    };

```

```

function init() {
    _renderMessage();
    _checkCart();
    _initDelivery();
    _bindHandlers();
}
function _renderMessage() {
    var template = _.template(ui.$orderMessageTemplate.html()),
        data;
    cart.update();
    data = {
        count: cart.getCountAll(),
        summa: cart.getSumma()
    };
    ui.$messageCart.html(template(data));
}
function _checkCart() {
    if (cart.getCountAll() === 0) {
        ui.$orderBtn.attr('disabled', 'disabled');
    }
}
function _changeDelivery() {
    var $item = ui.$delivery.btn.filter(':checked'),
        deliveryType = $item.attr('data-type'),
        deliverySumma = freeDelivery.enabled ? 0 : +$item.attr('data-summa'),
        cartSumma = cart.getSumma(),
        fullSumma = deliverySumma + cartSumma,
        alert =
            freeDelivery.enabled
                ? 'Будь-ласка, оберіть спосіб доставки'
                :
                    'Сума доставки ' + deliverySumma + ' гривень.' +
                    'Загальна сума замовлення: ' +
                    cartSumma + ' + ' + deliverySumma + ' = ' + fullSumma + ' гривень';
    ui.$delivery.type.val(deliveryType);
    ui.$delivery.summa.val(deliverySumma);
    ui.$fullSumma.val(fullSumma);
    ui.$delivery.alert.html(alert);
}

```

```

}
function _initDelivery() {
    freeDelivery.enabled = (cart.getSumma() >= freeDelivery.summa);
    ui.$delivery.btn.on('change', _changeDelivery);
    _changeDelivery();
}
function _bindHandlers() {
    ui.$orderForm.on('click', '.js-close-alert', _closeAlert);
    ui.$orderForm.on('submit', _onSubmitForm);
}
function _closeAlert(e) {
    $(e.target).parent().addClass('hidden');
}
function _validate() {
    var formData = ui.$orderForm.serializeArray(),
        name = _.find(formData, {name: 'name'}).value,
        email = _.find(formData, {name: 'email'}).value,
        isValid = (name !== "") && (email !== "");
    return isValid;
}
function _getCartData() {
    var cartData = cart.getData();
    _.each(cart.getData(), function(item) {
        item.name = encodeURIComponent(item.name);
    });
    return cartData;
}

function _orderSuccess(response) {
    console.info('response', response);
    ui.$orderForm[0].reset();
    ui.$alertOrderDone.removeClass('hidden');
}
function _orderError(response) {
    console.error('response', response);
}
function _orderComplete() {
    ui.$orderBtn.removeAttr('disabled').text('Надіслати замовлення');
}

```

```

}
function _onSubmitForm(e) {
    var isValid,
        formData,
        cartData,
        orderData;
    e.preventDefault();
    ui.$alertValidation.addClass('hidden');
    isValid = _validate();
    if (!isValid) {
        ui.$alertValidation.removeClass('hidden');
        return false;
    }
    formData = ui.$orderForm.serialize();
    cartData = _getCartData();
    orderData = formData + '&cart=' + JSON.stringify(cartData);
    ui.$orderBtn.attr('disabled', 'disabled').text('Іде обробка замовлення...');
    $.ajax({
        url: 'scripts/order.php',
        data: orderData,
        type: 'POST',
        cache: false,
        dataType: 'json',
        error: _orderError,
        success: function(response) {
            if (response.code === 'success') {
                _orderSuccess(response);
            } else {
                _orderError(response);
            }
        },
        complete: _orderComplete
    });
}
return {
    init: init
}
})(jQuery);

```

Common.php

```

<?php
define('DB_HOST', 'localhost');
define('DB_USER', 'root');
define('DB_PASSWORD', '');
define('DB_NAME', 'agrohelpdata');
function connectDB() {
    $errorMessage = 'Невозможно присоединиться до БД';
    $conn = new mysqli(DB_HOST, DB_USER, DB_PASSWORD, DB_NAME);
    if (!$conn)
        throw new Exception($errorMessage);
    else {
        $query = $conn->query('set names utf8');
        if (!$query)
            throw new Exception($errorMessage);
        else
            return $conn;
    }
}

```

Catalog_pag.php

```

<?php
include_once './common.php';
function getOptions() {
    $categoryId = (isset($_GET['category'])) ? (int)$_GET['category'] : 0;
    $page = (isset($_GET['page'])) ? (int)$_GET['page'] : 1;
    $limit = (isset($_GET['limit'])) ? (int)$_GET['limit'] : 5;

    return array(
        'category_id' => $categoryId,
        'page' => $page,
        'limit' => $limit
    );
}
function getGoods($options, $conn) {
    $page = $options['page'];
    $limit = (int)$options['limit'];
    $start = ($page - 1) * $limit;
    $categoryId = $options['category_id'];
}

```

```

$categoryWhere =
    ($categoryId !== 0)
        ? " g.category_id = $categoryId and "
        : "";
$queryBase = "
    select
        g.id as good_id,
        g.good as good,
        g.category_id as category_id,
        b.brand as brand,
        g.price as price,
        g.rating as rating,
        g.photo as photo
    from
        goods as g,
        brands as b
    where
        $categoryWhere
        g.brand_id = b.id
";
$queryCountAll = 'select count(*) count_all from (' . $queryBase . ') as tmp';
$data = $conn->query($queryCountAll);
$row = $data->fetch_assoc();
$countAll = (int)$row['count_all'];
$queryTotal = $queryBase . "
    order by price asc
    limit $start, $limit
";
$data = $conn->query($queryTotal);
$goods = $data->fetch_all(MYSQLI_ASSOC);
return array(
    'countAll' => $countAll,
    'goods' => $goods
);
}
try {
    $conn = connectDB();
    $options = getOptions();

```



```

    $data = getGoods($options, $conn);
    echo json_encode(array(
        'code' => 'success',
        'data' => $data
    ));
}
catch (Exception $e) {
    echo json_encode(array(
        'code' => 'error',
        'message' => $e->getMessage()
    ));
}

```

Order.php

```

<?php
define('DB_HOST', 'localhost');
define('DB_USER', 'root');
define('DB_PASSWORD', '');
define('DB_NAME', 'agrohelpdata');
function connectDB() {
    $errorMessage = 'Невозможно соединиться с БД';
    $conn = new mysqli(DB_HOST, DB_USER, DB_PASSWORD, DB_NAME);
    if (!$conn)
        throw new Exception($errorMessage);
    else {
        $query = $conn->query('set names utf8');
        if (!$query)
            throw new Exception($errorMessage);
        else
            return $conn;
    }
}
function getParam($param, $conn, $default = '') {
    return (isset($_POST[$param])) ? mysqli_real_escape_string($conn, $_POST[$param]) :
    $default;
}
function getData($conn) {
    return array(

```

```

'name' => getParam('name', $conn, 'noname'),
'email' => getParam('email', $conn, 'unknown email'),
'phone' => getParam('phone', $conn),
'address' => getParam('address', $conn),
'message' => getParam('message', $conn),
'delivery_type' => getParam('delivery_type', $conn),
'delivery_summa' => getParam('delivery_summa', $conn),
'full_summa' => getParam('full_summa', $conn),
'cart' => isset($_POST['cart']) ? stripslashes($_POST['cart']) : ''
);
}
function addClient($data, $conn) {
    $query = sprintf(
        "insert into clients (`name`, `email`, `phone`) values ('%s', '%s', '%s'",
        $data['name'],
        $data['email'],
        $data['phone']
    );
    $conn->query($query);
    return $conn->insert_id;
}
function addOrder($data, $conn) {
    $query = sprintf(
        "insert into orders (`client_id`, `address`, `message`) values (%d, '%s', '%s'",
        $data['client_id'],
        $data['address'],
        $data['message']
    );
    $conn->query($query);
    return $conn->insert_id;
}
function addDetails($data, $conn) {
    $cart = json_decode($data['cart'], true);
    $orderId = $data['order_id'];
    $values = array();
    foreach($cart as $cartItem) {
        $value = sprintf(
            "(%d, %d, '%s', %d, %d)",

```

```

        $orderId,
        $cartItem['id'],
        mysqli_real_escape_string($conn, $cartItem['name']),
        $cartItem['price'],
        $cartItem['count']
    );
    array_push($values, $value);
}
$query = sprintf(
    "insert into details (`order_id`, `good_id`, `good`, `price`, `count`) values %s",
    implode(',', $values)
);
$conn->query($query);
}
try {
    $conn = connectDB();
    $data = getData($conn);
    $clientId = addClient($data, $conn);
    $data['client_id'] = $clientId;
    $orderId = addOrder($data, $conn);
    $data['order_id'] = $orderId;
    addDetails($data, $conn);
    echo json_encode(array(
        'code' => 'success'
    ));
}
catch (Exception $e) {
    echo json_encode(array(
        'code' => 'error',
        'message' => $e->getMessage()
    ));
}
}

```

ДОДАТОК Г

ІНСТРУКЦІЯ КОРИСТУВАЧА

Для доступу на інтернет-магазин «Agrohelf» необхідно запустити браузер і ввести в адресний рядок наступну адресу: <http://agrohelp.com.ua>

Після завершення завантаження, зазвичай це займає близько 3-4 секунд, відкриється головна сторінка. Для повнофункціонального перегляду сайту потрібно використовувати наступні браузери:

- MS Edge;
- Google Chrome 3.0 і вище;
- Opera 6.0 і вище;
- Mozilla Firefox 1.0;
- Mozilla 1.7.

У верхній частині головної сторінки (рис.Г.1) розташовується «шапка», яка є незмінною на всіх сторінках сайту і складається з логотипу, панелі навігації та панелі кнопок керування каталогом.

Для того, щоб придбати товар(и), користувач спочатку повинен натиснути на кнопку «Додати у кошик» під товаром(ами), який(і) йому потрібні.

AGROHELP

Магазин добрив. №1 в Україні

Каталог

Кошик **2**

Оформлення замовлення

Більше про нас

Всі категорії

Ягоди

Квіти

Дерева



Товарів на сторінці:

2



1

2

3

>

»

Показані товари: 1 - 2 из 6



Артикул 4

Agrohlp4

12 грн

Додати в кошик



Артикул 2

Agrohlp2

13 грн

Додати в кошик

Навігація на сайті

Каталог

Кошик

Оформлення замовлення

Більше про нас

Контактна інформація:

+3801234567 (Vodafone)

+38012345678 (Kyivstar)

+38012345679 (Lifecell)

agrohelp (Skype)

AGROHELP

Рисунок Г.1 – Головна сторінка

Після успішного додавання товарів у кошик, користувач може перейти на сторінку «Кошик», натиснувши на певне посилання на панелі навігації. На рис.Г.2 зображено сторінку «Кошик».

Кошик

[Каталог](#)
[Кошик 2](#)
[Оформлення замовлення](#)
[Більше про нас](#)

Артикул	Назва	Ціна	Кількість	Сума	
2	Agrohelp2	13 грн	- 2 +	26 грн.	✕

Сума: 26 грн

[Оформити замовлення](#)

Рисунок Г.2 – Кошик

Після того, як користувач перевірів, чи все вірно він обрав, він натискає на кнопку «Оформити замовлення» та потрапляє на сторінку з формою для замовлення. На рис.Г.3 можна побачити, як виглядає форма.

В ній користувач вказує свої дані для відправки, обирає спосіб доставки (сайт відразу підсумовує вартість з різними видами доставки) та після того, як він усе заповнив, з'являється повідомлення «Дякуємо за замовлення! Незабаром ми з вами зв'яжемося» (рис.Г.4).

У вашому кошику 2 товар(ів) на загальну суму 26 гривень. Заповніть форму нижче, щоб оформити замовлення.

Ваше ім'я *

Email *

Телефон

Спосіб доставки

Самовивіз (безкоштовно)

В межах міста

Доставка Новою Поштою

Сума доставки 0 гривень. Загальна сума замовлення: 26 + 0 = 26 гривень

Адреса доставки

Коментар

Надіслати замовлення

Рисунок Г.3 – сторінка «Оформлення замовлення»

Сума доставки 0 гривень. Загальна сума замовлення: 26 + 0 = 26 гривень

Адреса доставки

Коментар

Дякуємо за замовлення! Незабаром ми з вами зв'яжемося ×

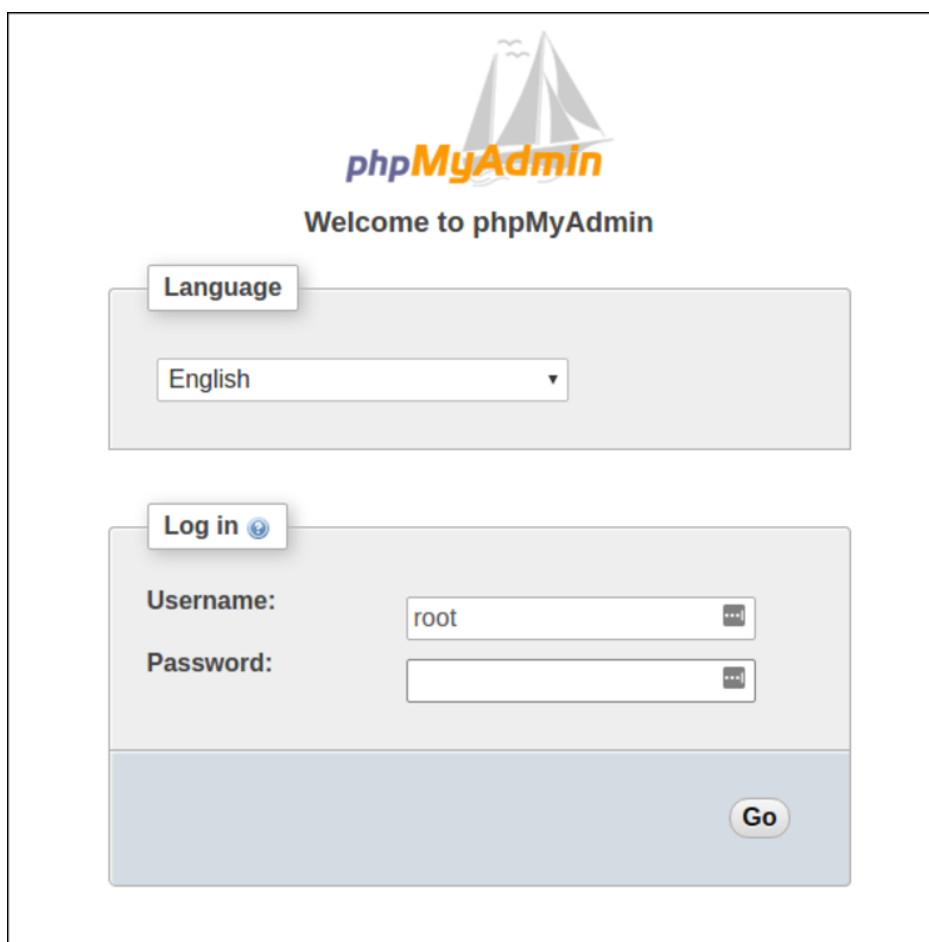
Рисунок Г.4 – повідомлення про успішне замовлення

В розділі «Більше про нас» клієнт має змогу переглянути інформацію про виробника. Поки що використовується типова заглушка «Lorem ipsum», але в перспективі можна додати зображення, точне місцезнаходження офісу виробника та контакти.

ДОДАТОК Д

ІНСТРУКЦІЯ АДМІНІСТРАТОРА

Адміністратор має змогу переглядати замовлення у базі даних та керувати асортиментом за допомогою phpMyAdmin. Достатньо на сервері прописати шлях до цього інструменту і з'явиться вікно авторизації (рис.Д.1).



The image shows the phpMyAdmin login interface. At the top, there is a logo of a sailboat with the text 'phpMyAdmin' and 'Welcome to phpMyAdmin'. Below the logo, there is a 'Language' dropdown menu currently set to 'English'. Underneath the language menu is a 'Log in' button with a blue eye icon. The login form consists of two input fields: 'Username:' with the value 'root' and 'Password:'. A 'Go' button is located at the bottom right of the form area.

Рисунок Д.1 – Вхід до адміністративної панелі керування базами даних

Після успішної авторизації потрапляємо до головної сторінки phpMyAdmin (Рисунок Д.2).

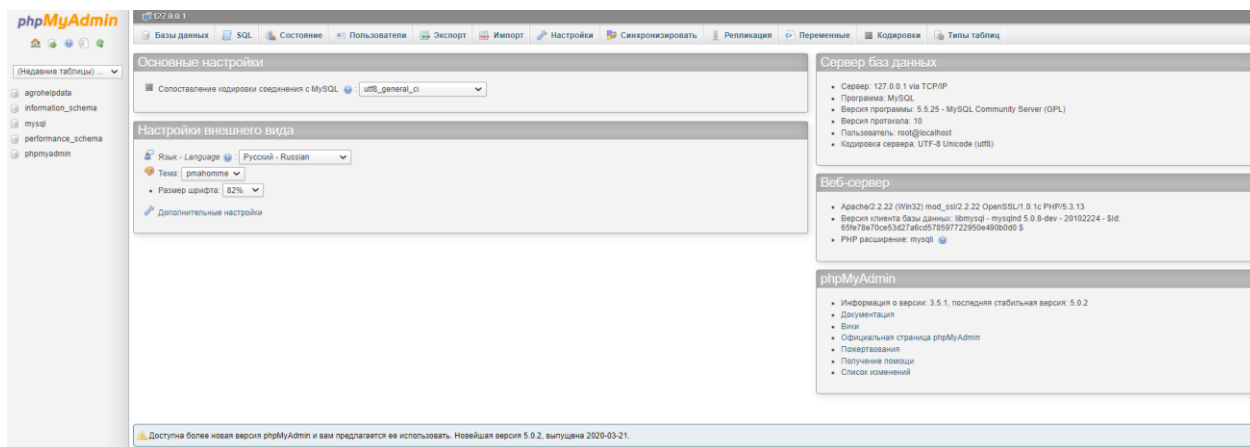


Рисунок Д.2 – Адміністративна панель

Обираємо базу даних «agrohelp». Переглядаємо усі наявні таблиці, які вже були описані. На рис.Д.3 зображено, яка таблиця за що відповідає. phpMyAdmin дозволяє експортувати таблиці і це можна використати для того, щоб переглядати їх в іншому ПЗ.

Таблица	Действие	Строки	Тип	Сравнение	Размер	Фрагментировано	
<input type="checkbox"/> brands	← Обзор Структура Поиск Вставить		InnoDB	utf8_general_ci	16 КБ	-	
<input type="checkbox"/> categories	← Обзор Структура Поиск Вставить		InnoDB	utf8_general_ci	16 КБ	-	
<input type="checkbox"/> clients	← Обзор Структура Поиск Вставить Очистить Удалить	3	InnoDB	utf8_general_ci	16 КБ	-	
<input type="checkbox"/> details	← Обзор Структура Поиск Вставить		DB	utf8_general_ci	16 КБ	-	
<input type="checkbox"/> goods	← Обзор Структура Поиск Вставить		DB	utf8_general_ci	48 КБ	-	
<input type="checkbox"/> orders	← Обзор Структура Поиск Вставить Очистить Удалить	3	InnoDB	utf8_general_ci	16 КБ	-	
6 таблиц	Всего		19	InnoDB	utf8_general_ci	128 КБ	0 Байт

Annotations in the image:

- A blue box labeled 'Таблиці каталогу' has arrows pointing to the 'brands' and 'categories' rows.
- A white box labeled 'Таблиці замовлень' has arrows pointing to the 'orders' and 'goods' rows.

Рисунок Д.3 – Наявні таблиці в базі даних «agrohelpdata»