Національний університет «Полтавська політехніка імені Юрія Кондратюка»
(повне найменування вищого навчального закладу)
Навчально-науковий інститут інформаційних технологій та робототехніки
(повна назва факультету)
Кафедра комп'ютерних та інформаційних технологій і систем
(повна назва кафедри)

Пояснювальна записка
до дипломного проекту (роботи)
_____магістра_____
(рівень вищої освіти)
на тему

Розроблення інформаційної системи для ландшафтного дизайну з інтелектуальним пошуком інформації

Виконав: студент _6 курсу, групи _601ТН_
напряму підготовки
_____122 комп'ютерні науки_____
(шифр і назва напряму підготовки)
_____Харченко М.С._
(прізвище та ініціали)
Керівник _____Бесєдін В.Ф._____
(прізвище та ініціали)
Консультант___ Головко Г.В.___
(прізвище та ініціали)
Рецензент _____Некоз І.В._____
(прізвище та ініціали)

Полтава – 2021 року

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ПОЛТАВСЬКА ПОЛІТЕХНІКА ІМЕНІ ЮРІЯ КОНДРАТЮКА»

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА РОБОТОТЕХНІКИ

КАФЕДРА КОМП'ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І СИСТЕМ

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

спеціальність 122 «Комп'ютерні науки»

на тему

«Розроблення інформаційної системи для ландшафтного дизайну з інтелектуальним пошуком інформації**»**

Студент групи 601-ТН__Харченко Максим Сергійович

Керівник роботи
доктор економічних наук,
професор Бесєдін В.Ф.

Консультант
кандидат технічних наук,
доцент Головко Г.В.

Завідувач кафедри
кандидат технічних наук,
доцент Головко Г.В.

Полтава – 2021

РЕФЕРАТ

Пояснювальна записка містить: 84 с., 43 рис., 24 джерела, 1 додаток.

Об’єкт дослідження – діяльність та інформаційна система підприємства з ландшафтного дизайну.

Предмет досліджень – інструментальні засоби розробки інформаційних систем, web-сайтів.

Мета кваліфікаційної роботи – розроблення і впровадження інформаційної системи підприємства з ландшафтного дизайну з елементами асоціативного пошуку.

Методи – технології об’єктно-орієнтованого програмування та проектування, уніфікована мова моделювання UML, теорія баз даних, системний аналіз, технології веб-програмування.

Ключові слова: база даних, онлайн, асоціативний пошук, інформаційна система, система управління базами даних, персональний комп’ютер, сайт, web-сервер.

ABSTRACT

The explanatory note contains: 84 pp., 43 Fig., 24 Sources, 1 appendix.

The object of research is the activity and information system of the enterprise of landscape design.

The subject of research - tools for the development of information systems, web-sites.

The purpose of the qualification work is to develop and implement an information system of the enterprise on landscape design with elements of associative search.

Methods – object-oriented programming and design technologies, unified UML modeling language, database theory, systems analysis, web programming technologies.

Keywords: database, on-line, associative search, informative system, control system by the bases of data, personal computer, web-site, web- server.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

БД – база даних.

ІТ – інформаційні технології.

ПЗ – програмне забезпечення.

ПК – персональний комп’ютер.

СУБД – система управління базами даних. Сукупність програмних і лінгвістичних засобів загального або спеціального призначення, що забезпечують керування створенням і використанням баз даних.

CMS (англ. Content management system) – система управління вмістом.

CRUD (create read update delete) – скорочене іменування чотирьох базових функцій при роботі з персистентними сховищами даних: створення, читання, редагування і видалення.

MVC (Модель-уявлення-контролер) – схема використання декількох шаблонів проектування, за допомогою яких модель даних програми, користувальницький інтерфейс і взаємодія з користувачем розділені на три окремих компонента так, що модифікація одного з компонентів надає мінімальний вплив на інші. Дана схема проектування часто використовується для побудови архітектурного каркаса, коли переходять від теорії до реалізації в конкретній предметній області.

ВСТУП

Інформаційна система (ІС) - це організована система збору, організації, зберігання та передачі інформації. Зокрема, це вивчення додаткових мереж, які люди та організації використовують для збору, фільтрування, обробки, створення та поширення даних.

На сьогоднішній день практично кожна організація має власний веб-сайт. В умовах використання сучасних інформаційних технологій – це необхідний чинник існування, що дозволяє розширити поле рекламної діяльності і привернути тим самим додаткових клієнтів. Також інформаційна система є дуже важливою частиною сучасного підприємства. Досягнення кожного сучасного підприємства в конкретній галузі прямо залежать від комп'ютеризації діяльності та рівня інформаційного забезпечення.

Завдяки асоціативному пошуку, після введення ключового слова система підбирає слова, які асоціюються з уже введеним словом. Після цього здійснюється пошук по комбінації слова-асоціації і вже введеної частини запиту, показуючи популярність пропонованих варіантів. Тобто проводиться пошук не точного входження слів запиту в сторінку, а пошук сторінок, максимально точно описуючих те, що написано в запиті.

Можна виділити такі переваги подачі інформації у формі онлайн інформаційної системи у порівнянні з традиційною очною формою:

– інформаційні системи добре працюють в мережі Інтернет тому, що його кліентам доступна детальна інформація;

– потенційні покупці товару можуть подзвонити або відправити повідомлення по e-mail, щоб отримати більше необхідної інформації;

– учасниками онлайн спілкування можуть бути всі охочі незалежно від місця знаходження.

Формування атмосфери довіри на онлайн інформаційних системах передбачає:

– наявність рейтингової системи;

– незалежну перевірку (верифікацію) дійсності відомостей, надаваних покупцями й продавцями;

– страхування проти шахрайства;

– проведення угод через рахунки типу «эскроу» (умовні документи, які стають документами повної дії після виконання певних умов) для забезпечення поставок оплаченої продукції. Платежі надходять рахунки «эскроу» й переправляються продавцеві після того, як товар доставлено покупцеві.

В даній дипломній роботі необхідно спроектувати та програмно реалізувати інформаційну систему підприємства з ландшафного дизайну з можливістю о пошуку необхідної інформації.

РОЗДІЛ 1
АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКИ ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Поняття веб-сайта

Веб-сайт (англ. website) – це одна або сукупність веб-сторінок, доступних в інтернеті через протоколи HTTP/HTTPS; Веб-сайт – це місце в інтернеті, яке визначається своєю адресою (URL), має свого власника і складається з веб-сторінок, які сприймаються як єдине ціле.

Сукупність всіх загальнодоступних веб-сайтів є Всесвітня павутина. Сторінки веб-сайту об'єднані загальною кореневою адресою, а також звичайно темою, логічною структурою, оформленням або авторством.

Сторінки веб-сайтів – це файли з текстом, розміченим на мові HTML або XHTML. Ці файли, будучи завантаженими відвідувачем на його комп'ютер, обробляються програмою-оглядачем, званою браузером і виводяться на засіб відображення користувача (монітор, екран КПК, принтер або синтезатор мови). Мова HTML/XHTML дозволяє форматувати текст, розрізняти в ньому функціональні елементи, створювати гіпертекстові посилання (гіперпосилання) і вставляти в сторінку, що відображається, зображення, звукозаписи і інші мультимедійні елементи. Відображення сторінки можна змінити додаванням в неї таблиці стилів на мові CSS або сценаріїв на мові JavaScript [5].

Сторінки сайтів можуть бути простими статичними наборами файлів або створюватися спеціальною комп'ютерною програмою на сервері – движком сайту. Движок може бути або зроблений на замовлення для окремого сайту, або готовим продуктом, розрахованим на якийсь клас сайтів. Деякі з движків можуть забезпечити власнику сайту можливість гнучкої настройки структуризації і виведення інформації на веб-сайті; такі називаються системами управління змістом. Веб-сайт може бути розміщений як на одному сервері, так і на декількох (наприклад, портали). Послуги з розміщення сайту

на сервері називаються хостингом і надаються за певну плату за одиницю часу. За умовами надання хостинг часто розділяється на платний і безкоштовний. Недоліками безкоштовного хостингу є обмежений дисковий простір, невелика пропускна спроможність інтернет каналу (низька швидкість завантаження сайту), відсутність обмежена функціональність серверних технологій (Perl, PHP, Python, ASP, Ruby, JSP, MYSQL). Тому безкоштовний хостинг використовується в основному для любительських сайтів [11].

На сьогодні інформаційна система – це обличчя компанії. Інтернет є самим популярним інформаційним ресурсом і залишиться таким в найближчому майбутньому. Ми вводимо запит в Google і отримуємо перелік підприємств, які можуть надати потрібну послугу або товар, заходимо на сайт і вся інформація в наявності. Все просто і зручно. Більше того, відсутність у компанії сайту виглядає несолідно, а сама компанія сприймається несерйозно.

Щоб сайт працював на свою організацію, і приводив нових клієнтів і, відповідно, прибуток, він має бути ефективним.

Сайт повинен мати чітку і зрозумілу структуру. Таку, щоб будь-яка людина, зайшовши на сайт, могла знайти і отримати потрібну інформацію і при цьому не заблукати. Тексти повинні бути максимально прості і легкі до сприйняття.

1.2 Бази даних – основа інформаційних систем

Активна діяльність по пошуку прийнятних способів усуспільнення безупинно зростаючого обсягу інформації привела до створення на початку 60-х років спеціальних програмних комплексів, називаних «Системи управління базами даних» (СУБД) [21].

Основна особливість СУБД – це наявність процедур для введення й збереження не тільки самих даних, але й описів їхньої структури. Файли,

оснащені описом збережених у них даних і знаходяться під управлінням СУБД, стали називати банки даних, а потім «Бази даних» (БД) [21].

Банк даних – це система спеціальним образом організованих даних – баз даних, а також програмних, технічних, мовних і організаційно-методичних засобів, призначених для забезпечення централізованого нагромадження і колективного багатоцільового використання даних [20].

База даних (від грецьк. basic – основа) – іменована сукупність даних, що відображають стан об'єктів і їхніх відносин у розглянутій предметній області [20].

Система управління базами даних – сукупність мовних і програмних засобів, призначених для створення, ведення і конкурентного використання бази даних багатьма користувачами. Структура СУБД визначається використовуваною моделлю даних [21].

База даних організовується так, що дані збираються один раз і централізовано зберігаються (і модифікуються) у виді, доступному всім фахівцям чи системам програмування, що можуть їх використовувати. Особливості організації даних у базах даних забезпечують використання тих самих даних у різних додатках, дозволяють вирішувати різні задачі планування, дослідження й управління. Бази даних зводять до мінімуму дублювання даних, прибігаючи до дублювання тільки для прискорення доступу до даних для забезпечення відновлення бази даних при її руйнуванні. Одна з важливих рис баз даних – незалежність даних від особливостей прикладних програм, що їх використовують, а також можливість створення цих програм, у такій формі, що зміна особливостей збереження, логічної структури значень даних не вимагає зміни програм їхньої обробки. Іншою важливою рисою баз даних є можливість зміни фізичних особливостей збереження даних без зміни їхньої логічної структури [21].

Якість БД, зокрема, вірогідність інформації, що міститься в них, багато в чому визначається оперативністю їхньої актуалізації. Однак, 15% БД обновляються щорічно, приблизно 11% – щоквартalу, 13% – щомісяця, близько

7% – щодня. Лідером підготовки БД, доступних на світовому ринку, є США (більш 5 тис. БД у рік). До країн, у яких підготовляється більш 100 БД у рік, відносяться Росія, Великобританія (641), Канада (480), Австралія (182), Франція (288), Німеччина (342), Японія (153). БД, доступні на світовому ринку, представлені на 29 мовах світу. За станом на 1995 рік у світі існує 1131 БД, що діють у режимі online, із них 99% БД належить США і лише 32 БД(0,03%) – країнам третього світу, що яскраво свідчить про монополізацію науки [21].

Інформація в базі даних зберігається в таблицях**.** *Таблиця* може бути файлом бази даних, а може бути так, що всі таблиці БД зберігаються в одному файлі. Таблиця зберігає неопрацьовані дані. У базі даних може зберігається одна чи більше таблиць. Розміщаючи інформацію в багатьох таблицях у межах однієї бази даних, ми зменшимо працезатрати на підтримку бази даних. У таблиці інформація групується по рядках і стовпцям.

Запис – рядок таблиці. Кожен рядок вважається окремою величиною, до якої можна одержати доступ. Записи звичайно, ідентифікуються по деякій унікальній характеристиці. Інакше запис можна визначити як набір зв'язаних збережених полів.

Поле – стовпець таблиці. Найменша одиниця збережених даних. Кожне поле має визначений тип даних (текст, число, дата і т.д.), довжину й унікальне ім'я, що ідентифікує інформацію, що зберігається в цьому полі. На перетинанні рядка й стовпця знаходиться значення – власне дані. Для добування інформації з бази даних використовується запит**.**

Запит – звертання до бази даних, що містить завдання на пошук, зчитування в базі даних відповідно до деякої умови й видачу інформації користувачу в необхідному виді, можливо, після деякої обробки. Складається мовою запитів. За допомогою запиту можна вибрати і визначити групу записів. Обрані записи називаються динамічним набором.

Функціонування бази даних забезпечується сукупністю мовних і програмних засобів, називаних системою керування базами даних (СУБД).

СУБД забезпечують:

а) визначення даних, підлягаючих збереженню в базі даних (визначення логічних властивостей даних, що відповідають уявленням користувача які називаються структури даних у базі даних, а також фізичну організацію збереження даних, які називаються структурами збереження бази даних);

б) початкове завантаження даних у бази даних;

в) відновлення даних;

г) доступ до даних по різних запитах користувача, добір і добування деякої частини бази даних, редагування витягнутих даних і видачу їх користувачу.

Перераховані дії прийнято називати процесом одержання довідок із бази даних. Спеціальні засоби СУБД забезпечують таємність даних, тобто захист даних від неправомочного впливу, і цілісності даних.

Цілісність даних – захист від непередбаченої взаємодії конкурентних процесів, що приводять до випадкового чи навмисного руйнування даних, а також від відмовлень устаткування.

*Історія***.** Першою моделлю «складу даних» ми зобов'язані власнику престижної премії Алана Тьюринга Чарльзу Вільямові Бахману. Дійсний ветеран комп'ютеру, що активно проробив «на майбутнє» більш 44 років, Бахман знаменитий як творець фундаменту теорії і практики баз даних. З 1960 р. у General Electric він займався дослідженнями і практичною реалізацією системи керування базами даних (СУБД) IDS (Integrated Data Store). Усього за чотири роки (фантастично короткий термін, якщо взяти до уваги новизну розробки, низький рівень технологічних засобів, «корпоративний масштаб» проекту з двох чоловік) IDS була «доведена» до придатного до продажів стану і стала першої комерційний СУБД. Ідеологічна основа (більш «науково» називана моделлю бази даних) IDS, за якої Бахман заслужено був визнаний

гідним у 1973 р. вищої комп'ютерної нагороди ACM, одержала назву «мережний» (network) [20].

Практично одночасно з General Electric розробку СУБД вела і IBM у співдружності з майбутньої не менш знаменитої Rockwell (на початку 60-х вона називалася North American Aviation). Дітище альянсу побачило світло на кілька років пізніше IDS і, незважаючи на схожу абревіатурну назву (IMS, Information Management System), ознаменувало новий період у розвитку «складів для інформації». У IMS застосовувалася ієрархічна (hierarchical) модель бази даних.

Обидві системи, IDS і IMS, дозволяли використовувати свої ресурси з програм, написаних на мовах програмування високого рівня (у першу чергу – Cobol), але тільки за допомогою низькорівневих інтерфейсів.

«Технологічний вибух» назрівав, і в 1970 р. він «прогримів» у всю міць. Едгар Кодд (Edgar F. Codd, так само, як і Бахман, нагороджений Тьюрінгівською премією) опублікував статті, у якій описувалася принципово нова модель бази даних, що одержала назву «реляційної». Таблична модель бази даних Кодда ознаменувала початок самого тривалого етапу в розвитку «сховищ інформації», що триває фактично донині. Більш докладний опис моделей даних, що спочатку використовувалися для створення баз даних ми розглянемо на наступному занятті[20].

Модель даних – це деяка абстракція, що, будучи застосовна до конкретних даних, дозволяє користувачам і розроблювачам трактувати їх уже як інформацію, тобто відомості, що містять не тільки дані, але і взаємозв'язок між ними.

Відповідно до розглянутого раніше трьохрівневою архітектурою ми зіштовхуємося з поняттям моделі даних стосовно кожного рівня. І дійсно, фізична модель даних оперує категоріями, що стосуються організації зовнішньої пам'яті і структур збереження, використовуваних у даному операційному середовищі. В даний момент як фізичні моделі використовуються різні методи розміщення даних, засновані на файлових структурах: це організація файлів прямого і послідовного доступу, індексних файлів і

інвертованих файлів, що використовують різні методи каширування, взаємозалежних файлів. Крім того, сучасні СУБД широко використовують сторінкову організацію даних. Фізичні моделі даних, засновані на сторінковій організації, є найбільш перспективними.

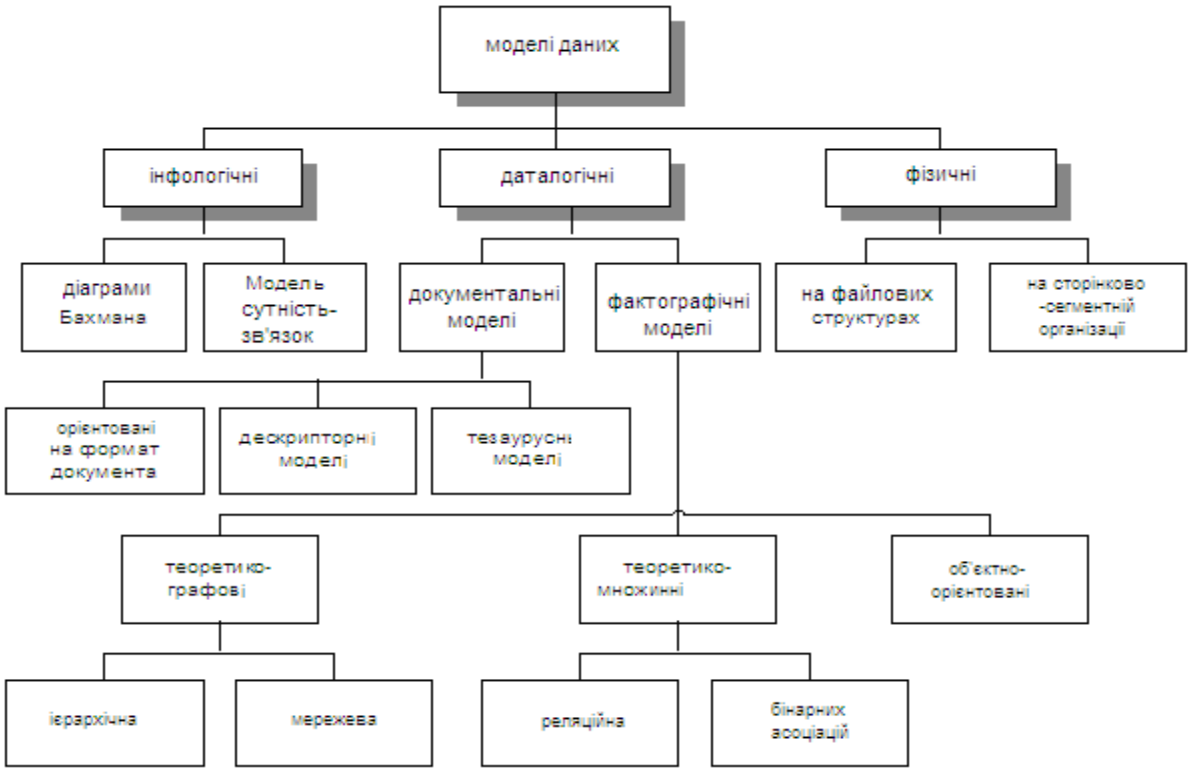

Рисунок 1.1 – Класифікація моделей даних.

Найбільший інтерес викликають моделі даних, які використовуються на концептуальному рівні. Стосовно них зовнішні моделі називаються підсхемами і використовують ті ж абстрактні категорії, що і концептуальні моделі даних.

Крім трьох розглянутих рівнів абстракції при проектуванні БД існує ще один рівень, що передує їм. Модель цього рівня повинна виражати інформацію про предметну область у вигляді, незалежному від використовуваної СУБД. Ці моделі називаються інфологічними, чи семантичними, і відбивають у природній і зручній для розроблювачів і інших користувачів формі інформаційно-логічний рівень абстрагування, зв’язаний з, фіксацією й описом об’єктів предметної області, їхніх властивостей і їхніх взаємозв’язків.

Інфологічні моделі даних використовуються на ранніх стадіях проектування для опису структур даних у процесі розробки додатку, а даталогічні моделі вже підтримуються конкретною СУБД.

Документальні моделі даних відповідають уявленню про слабкоструктуровану інформацію, орієнтовану в основному на вільні формати документів, текстів природною мовою.

Тезаурусні моделі засновані на принципі організації словників, містять визначені мовні конструкції і принципи їхньої взаємодії в заданій граматиці. Ці моделі ефективно використовуються в системах-перекладачах, особливо багатомовних перекладачах. Принцип збереження інформації в цих системах і підкоряється тезаурусним моделям.

Дескрипторні моделі – найпростіші з документальних моделей, вони широко використовувалися на ранніх стадіях використання документальних баз даних. У цих моделях кожному документу відповідав дескриптор – описувач. Цей дескриптор мав тверду структуру й описував документ відповідно до тих характеристик, що вимагаються для роботи з документами в розроблювальної документальний БД.

1.3 Асоціативний пошук

Розробники алгоритму постаралися відтворити динамічну, асоціативну природу людського мислення. Вони виходили з того, що при прийнятті рішень люди, використовують для обробки інформації неієрархічні способи аналізу. Обдумуючи задачу, кожна людина йде до висновку своїм шляхом. Користувачам надається можливість аналізувати дані відповідно до власних розумових процесів.

Традиційні методи пошуку і фільтрації інформації були розроблені для бібліотечних баз даних, тобто для інформаційних ресурсів обмеженого обсягу і заздалегідь відомої структури. Створення глобальної мережі і вихід Web за рамки інтересів наукового співтовариства призвели до того, що число

постачальників інформації стало стрімко рости, при тому, що опублікована ними інформація не мала однорідної структури. Наступний інформаційний вибух став викликом стандартним інформаційним технологіям. Нові масштаби з одного боку зробили аутсайдерами деякі раніше конкурентоспроможні інтелектуальні технології, а з іншого – стимулювали інтенсивні дослідження в області лінгвістичних та імовірнісних методів обробки текстової інформації та нових методів навігації в неоднорідному інформаційному морі [19].

Асоціативний пошук ґрунтується на двох ключових принципах:

- всі дані зберігаються в пам'яті комп'ютера (ОЗУ);
- обчислення виконуються в реальному часі.

Ці архітектурні рішення підкріплюються двома важливими тенденціями в розвитку комп'ютерів. Перша – перехід від 32- до 64-розрядних обчислень, що призвело до експоненціального збільшенню розміру ОЗУ комп'ютера. В даний час можна придбати сервери, де ОЗУ досягає 512 Гбайт, а не так давно, більшість серверів розташовувало ОЗУ ємністю 4 Гбайт. Завдяки збільшенню доступною пам'яті стало можливо перемістити сховище даних з диска безпосередньо в ОЗУ. Друга обставина – повсюдне поширення багатоядерних процесорів. В даний час широко застосовуються сервери, які мають 8 процесорів і в цілому 48 ядер. Зміни привели до серйозного збільшення обчислювальної потужності додатків, орієнтованих на паралельні обчислення. Високопродуктивні сервери пропонуються менш ніж за 50 000 дол. Зовсім недавно для досягнення такого рівня обчислювальних можливостей були потрібні спеціально сконструйовані комп'ютери вартістю сотні тисяч і навіть мільйони доларів [22].

Архітектура зберігання і обробки інформації в оперативній пам'яті (in-memory), дозволяє маніпулювати величезними обсягами даних, підтримуючи високий рівень інтерактивності. Стиснення даних при завантаженні в пам'ять дозволяє зберігати дані в ОЗУ ефективніше, ніж на диску, як це прийнято в традиційних реляційних базах даних. При передачі даних в пам'ять система

будує карту посилань між елементами даних, щоб спростити візуалізацію їх зв'язків .

Завдяки здатності виконувати обчислення в реальному часі можна швидко розраховувати складні заходи і метрики. В новітніх системах закладена можливість розподіляти обчислення між всіма доступними ядрами процесорів і управляти робочим навантаженням одночасно для великої кількості користувачів. Крім того, платформа зберігає в кеш-пам'яті результати всіх запитів, тому типові обчислення виконуються мінімальна кількість разів.

1.4 Огляд існуючих в Інтернеті інформаційних систем про надання послуг ландщафтного дизайну в містах України

Зазвичай подібні сайти створюють з гарним дизайном, але чи достатньо цього користувачеві?! Крім того потрібно добре інформувати користувача, надати максимально інформації про послугу, аби в того не виникало зайвих питань. Сайт має бути доступним та зрозумілим для будь-якого клієнта. Повинна бути проста навігація, щоб кожен зміг замовити деталь без проблем.

Зараз розглянемо приклади деяких сайтів, що стосуються автосервісу автомобілів:

1. «Зеленая Магия» ландшафний дизайн представлений на рис. 1.2, 1.3

Переваги:

- гарна навігація;
- все працює швидко та адекватно;
- зручне меню.

Недоліки представленого сайту:

- застарілий дизайн;
- неяксне кольорове оформлення;
- невеликі проблеми з пошуком інформації;
- галерея потребує багато ресурсів.

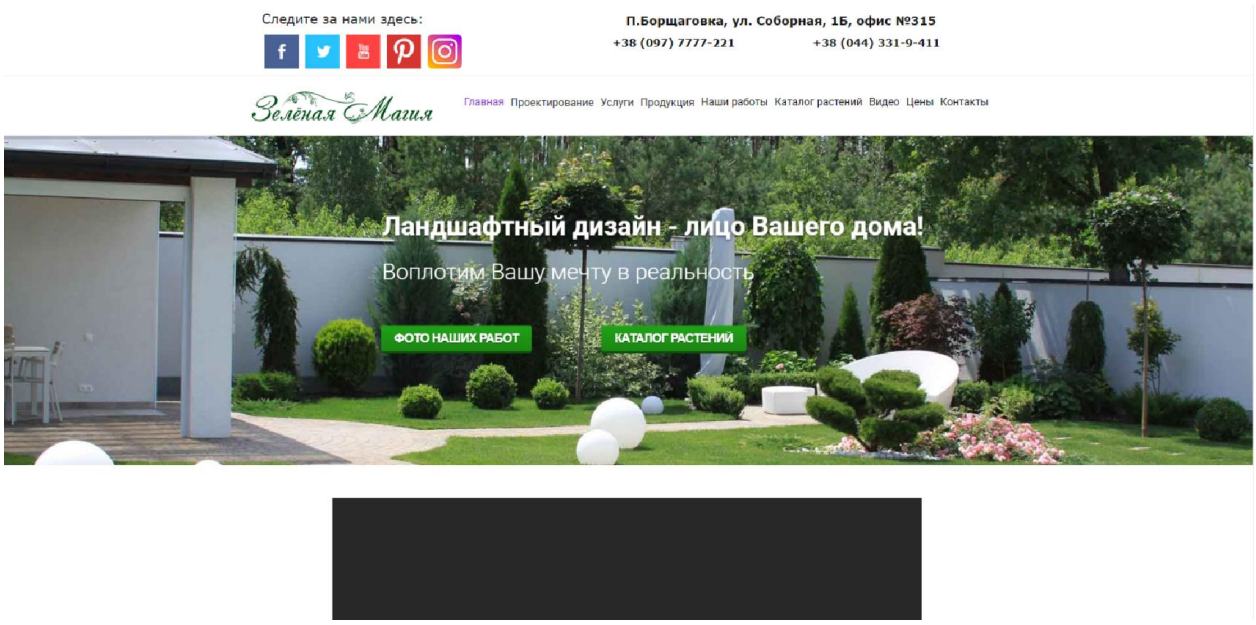

Рисунок 1.2− «Зеленая Магия» ландшафний дизайн

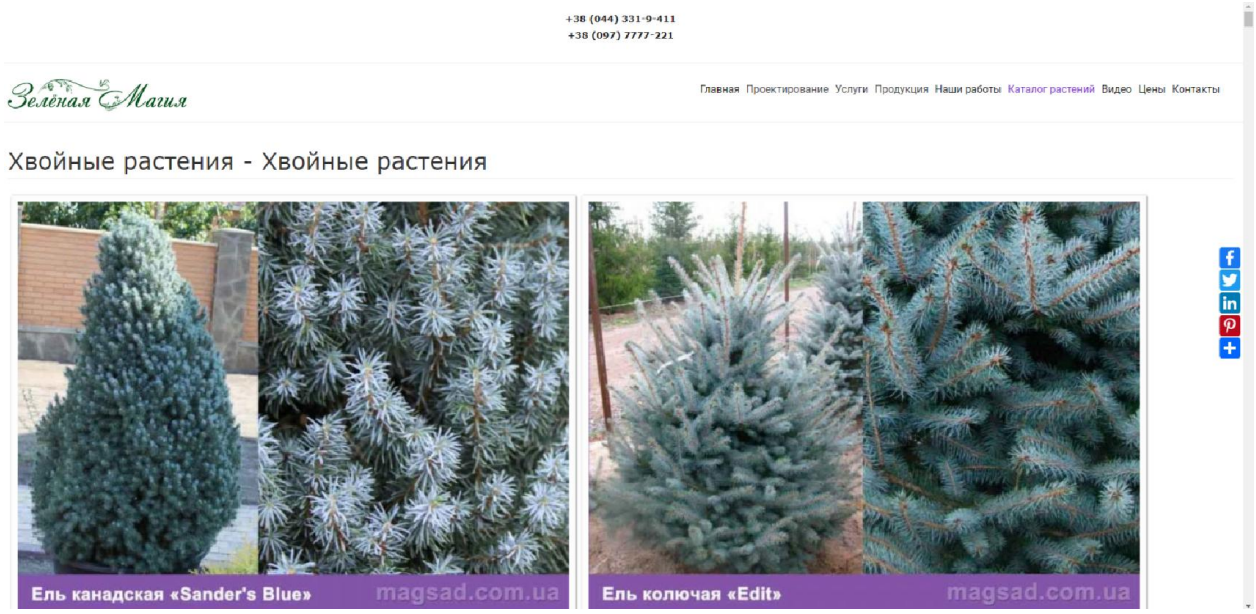

Рисунок 1.3− «Зеленая Магия» ландшафний дизайн

2. «Зелені янголи» ландшафний дизайн у Києві представлений на рис. 1.4

Переваги:

- зручна навиігація;
- легкий та зручний інтерфейс;
- наявність магазину.

Недоліки представленого сайту:

- застарілий дизайн;
- багато непотрібної інформації;
- незручне меню.

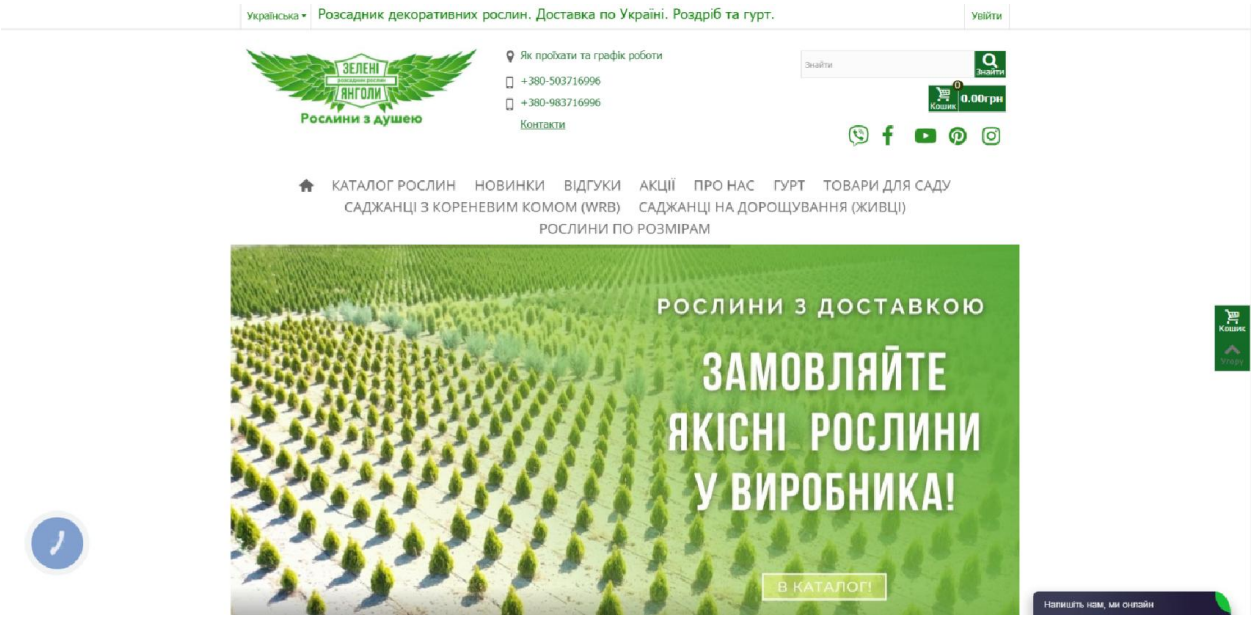

Рисунок 1.4 − «Зелені янголи» ландшафний дизайн у Києві

3. «TRIZIO» ландшафний дизайн у Києві представлений на рис. 1.5

Переваги:

- Сучасний дизайн;
- зручна навиігація;
- гарно описані послуги.

Недоліки представленого сайту:

- незруча таблиця ціноутвоення;
- багато непотрібної інформації.

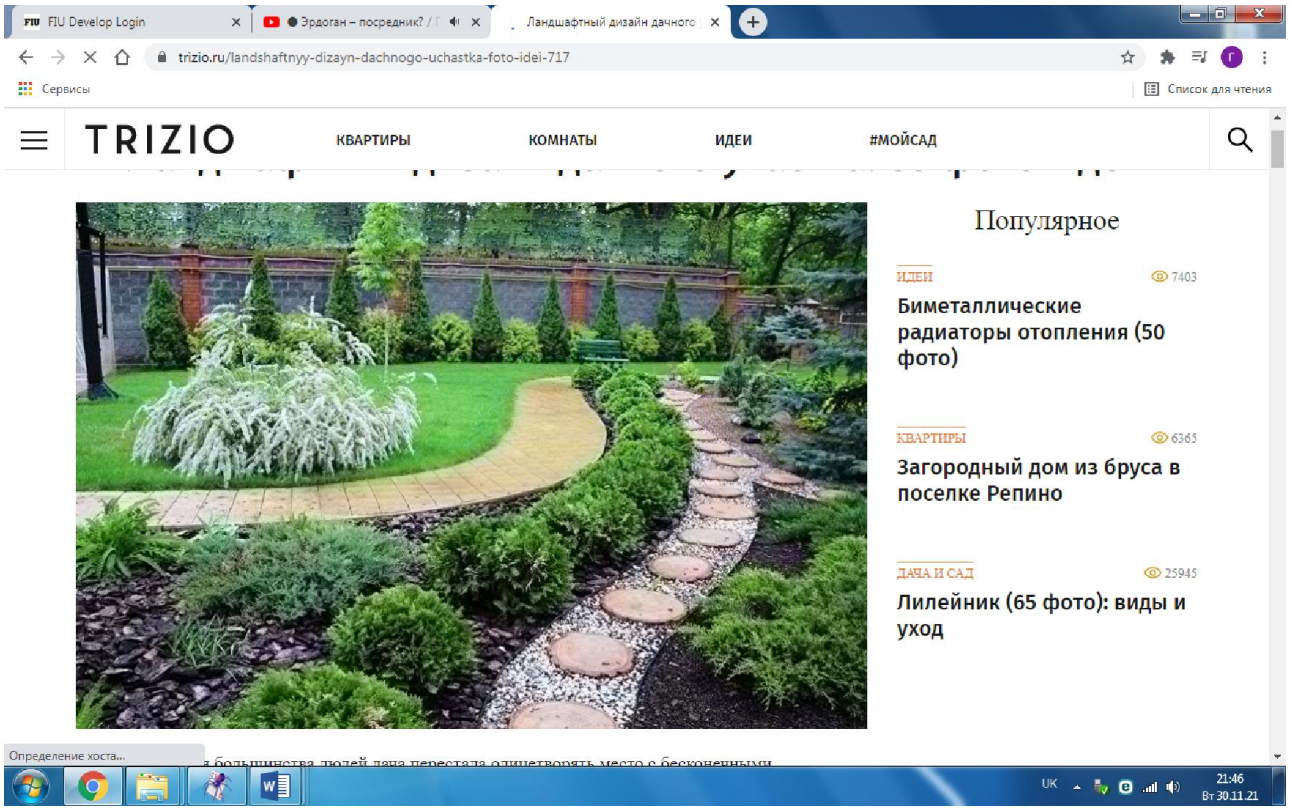

Рисунок 1.5 − «TRIZIO» ландшафний дизайн у Києві

1.5 Постановка задачі

На основі проведеного аналізу була поставлена задача дослідження: спроектувати та програмно реалізувати інформаційну систему підприємства ландшафтного дизайну «THE LANDSCAPER» з системою пошуку інформації (асоціативний пошук), яка повиненна забезпечувати виконання таких функцій:

1. Відображення інформації про підприємство на головній сторінці.
2. Реєстрація нових користувачів.
3. Користувачам:
   - 3.1. переглядати інформацію;
   - 3.2. здійснювати пошук необхідної інформації;
   - 3.3. обмінюватись повідомленнями та файлами;
4. Адміністратору:
   - 4.1. Можливість редагувати дані, блокувати і видаляти користувачів.

4.2. Редагування інформації.

4.3. Управління існуючими ролями, створення і видалення ролей.

4.4. Управління дозволами для ролей.

РОЗДІЛ 2
ТЕОРЕТИЧНА ЧАСТИНА РОЗРОБЛЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Технологія створення веб-сайту

Веб-програмування – галузь веб-розробки і різновид дизайну, в завдання якої входить проектування користувальницьких веб-інтерфейсів для сайтів або веб-додатків. Веб-дизайнери проектують логічну структуру веб-сторінок, продумують найбільш зручні рішення подачі інформації, а також займаються художнім оформленням веб-проекту. В результаті перетину двох галузей людської діяльності грамотний веб-дизайнер повинен бути знайомий з останніми веб-технологіями і володіти відповідними художніми якостями. На сьогоднішній день існують кілька етапів розробки веб-сайту:

1. Передпроектна підготовка.

Визначення тематики майбутнього сайту, з'ясування цілей і завдань, визначення і аналіз цільової аудиторії. Аналіз конкурентів. Розробка структури сайту – ескіз сайту на папері. Створення списку майбутніх тематичних розділів.

2. Розробка дизайну сторінок.

2.1. Дизайн-концепція сайту (креативний дизайн)

Креативна ідея, розробка основної графічної концепції дизайну сайту на прикладі головної та другорядних сторінок. Вибір колірної гамми, художнього стилю. Підготовка макету дизайну.

2.2. Технічний дизайн.

Розробка логічної і фізичної структури ресурсу. Компонування сторінки, верстальної структури. Елементи навігації.

3. Верстка.

3.1. Створення шаблонів сторінок.

3.2. Перевірка правильності написання коду.

3.3. Верстка сторінок сайту на основі затвердженого дизайну типових сторінок.

3.4. Збірка сторінок.

4. Інформаційне наповнення сайту

4.1. Підготовка текстових матеріалів.

4.2. Підготовка графічних матеріалів у растровому форматі, оптимізація картинок.

4.3. Заповнення сторінок.

5. Програмна частина проекту

5.1. Інтеграція сайту з системою управління

Зараз вже жоден сучасний сайт не обходиться без системи управління, оскільки важливою є не лише красива зовнішня оболонка цього сайту, але і можливість зручної роботи з ним. Це особливо актуально для сайтів з розгалуженою структурою і великим об'ємом даних.

В цей етап входить:

- інтеграція з системою управління;
- програмування, налаштування сервера;
- забезпечення безпеки проекту;
- контроль якості.

5.2. Програмування, запуск проекту

На цьому етапі допрацьовується функціонал, що не міститься у стандартному складі системи управління.

6. Тестування сайту в Інтернеті

6.1. Тестування сайту на наявність помилок та коректність функціонування в різних браузерах (Internet Explorer, Netscape, Opera, Safari).

6.2. Перевірка ідентичності відображення сторінок в різних екранних роздільних здатністях в різних браузерах.

7. Розміщення сайту в Інтернеті

7.1. Організація робіт з розміщення проекту в мережі Інтернет.

7.2. Вибір та реєстрація доменного імені.

7.3. Вибір хостинг провайдера, розміщення сайту. Фінальне тестування сайту.

7.4. Навчання персоналу клієнта як працювати з системою управління сайту.

8. Просування сайту.

В даний час вже мало просто розробити якісний сайт із зручною структурою і навігацією, важливим є забезпечення для сайту високої відвідуваності. Спромогтися цього можна не лише розміщенням реклами в засобах масової інформації, але і здійснюючи просування сайту в пошукових системах і каталогах, а також рекламою в Інтернеті.

9. Подальша підтримка сайту

Перші шість етапів відносяться безпосередньо до створення сайту, решта потрібні для подальшого існування сайту.

2.2 Вибір мов і засобів програмування

Існує кілька мов програмування розроблених спеціально для створення веб сторінок та сайтів, зокрема PHP, Perl, Python, Ruby, ASP.NET, Java, Groovy. В свою чергу вони поділяються на клієнтські та серверні.

2.2.1 Вибір клієнтської мови програмування. Технології веб-програмування на стороні клієнта включають в себе набір різних засобів і мов програмування. Перш за все це JavaScript, підтримка якого закладена практично в будь-якому браузер. JavaScript використовується частіше, ніж будь-які інші мови для написання скриптів, що працюють на стороні клієнта. Він є простим, його код легко інтегрувати в код HTML-сторінки, в той же час він надає достатньо багато можливостей.

При виборі клієнтської мови доцільно використовувати HTML оскільки вона підтримується усіма відомими браузерами і працює майже безвідмовно.

HTML (Мова розмітки гіпертекстових документів) – стандартна мова розмітки веб-сторінок в Інтернеті. Більшість веб-сторінок створюються за допомогою мови HTML (або XHTML). Документ HTML обробляється браузером та відтворюється на екрані у зручному для людини вигляді [11].

JavaScript являє собою мову написання сценаріїв на стороні клієнта, яка вносить на Web-сторінки елементи інтерактивності і умовної поведінки. За допомогою JavaScript можна виводити додаткову інформацію про посилання, створювати інтерактивні ефекти при роботі з мишею, змінювати за певних умов вміст сторінок, випадковим чином відображати вміст сторінки, завантажувати вміст в нові вікна браузера і фреймів.

Сценарії JavaScript зазвичай поміщають безпосередньо в документ HTML. Вони можуть перебувати або в заголовку або в тілі документа. В один документ можна помістити кілька сценаріїв.

2.2.2 Вибір серверної мови програмування. Раціонально буде використовувати PHP оскільки вона зручна і підтримується більшістю хостинг – серверів. PHP (гіпертекстовий препроцесор) – скриптова мова програмування, була створена для генерації HTML-сторінок на стороні веб-сервера. PHP є однією з найпоширеніших мов, що використовуються у сфері веб-розробок. PHP інтерпретується веб-сервером у HTML-код, який передається на сторону клієнта.

PHP: Hypertext Preprocessor – скріптова мова програмування, створена для генерації HTML-сторінок на веб-сервері і роботи з базами даних. В даний час підтримується переважною більшістю провайдерів хостингу. Входить в LAMP – «стандартний» набір для створення веб-сайтів (Linux, Apache, MySQL, PHP (Python або Perl)) [2].

У області програмування для Мережі PHP – одна з популярних скріптових мов (разом з JSP, Perl і мовами, використовуваними в ASP.NET) завдяки своїй простоті, швидкості виконання, багатій функціональності і розповсюдженню початкових кодів на основі ліцензії PHP. PHP відрізняється

наявністю ядра і модулів, що підключаються, «розширень»: для роботи з базами даних, сокетами, динамічною графікою, криптографічними бібліотеками, документами формату PDF і т.п. Будь-який охочий може розробити своє власне розширення і підключити його. Існують сотні розширень, проте в стандартне постачання входить лише декілька десятків тих, що добре зарекомендували себе. Інтерпретатор PHP підключається до веб-серверу або через модуль, створений спеціально для цього сервера (наприклад, для Apache або IIS), або як CGI-додаток [2].

Окрім цього, він може використовуватися для вирішення адміністративних завдань в операційних системах UNIX, GNU/Linux, Microsoft Windows, Mac OS X і AmigaOS. Проте в такій якості він не набув поширення, віддаючи пальму першості Perl, Python і VBScript.

Синтаксис PHP подібний синтаксису мови Сі. Деякі елементи, такі як асоціативні масиви і цикл foreach, запозичені з Perl.

Нині PHP використовується сотнями тисяч розробників. Декілька мільйонів сайтів повідомляють про роботу з PHP, що складає більш п’ятої частки доменів Інтернету.

Група розробників PHP складається з безлічі людей, що добровільно працюють над ядром і розширеннями PHP, і суміжними проектами, такими, як PEAR або документація мови.

Назва PHP – рекурсивна абревіатура, що означає «PHP: Hypertext Preprocessor» (раніше акронім розшифровувався як «Personal Home Page Tools»). Спочатку PHP створювався як надбудова над Perl для полегшення розробки веб-сторінок [2].

У 1994 році данський програміст (що нині живе в Канаді) Расмус Лердорф (Rasmus Lerdorf) написав набір скриптів на Perl/CGI для висновку і обліку відвідувачів його онлайн-резюме, оброблювальний шаблони HTML-документів. Лердорф назвав набір Personal Home Page (Особиста Домашня Сторінка). Незабаром функціональності і швидкості Perl – інтерпретатора скриптів – перестало вистачати, і Лердорф написав на мові C новий

інтерпретатор шаблонів PHP/FI (англ. Personal Home Page / Forms Interpreter – «Особиста Домашня Сторінка / Інтерпретатор форм». PHP/FI включав базову функціональність сьогоднішнього PHP: оформлення змінних в стилі Perl ($ім'я_змінної для виведення значення), автоматичну обробку форм і встроєний в HTML-текст і багато що інше. Новонароджена мова відрізнялася від свого пророджича простішим і обмеженим синтаксисом.

У 1997 році після тривалого тестування бети вийшла друга версія обробника, написаного на C – PHP/FI 2.0. Її використовували близько 1 % (приблизно 50 тисяч) всіх інтернет-доменів світу.

PHP 3.0 була першою версією, що нагадує PHP, яким ми знаємо його сьогодні. У 1997 році два ізраїльські програмісти Енді Гутманс (Andi Gutmans) і Зів Сураські (Zeev Suraski), два розробники з ізраїльського інституту технологій (Technion), переписали код з нуля: розробники визнали PHP/FI 2.0 непридатним для розробки додатку електронної комерції, над яким вони працювали для проекту Університету розташованого в Хайфі, Ізраїль. Для спільної роботи над PHP 3.0 за допомогою бази розробників PHP/FI 2.0 Енді, Расмус і Зів вирішили об'єднатися і оголосити PHP 3.0 офіційним наступником PHP/FI, розробка ж PHP/FI була практично повністю припинена [2].

Однією з сильних сторін PHP 3.0 була можливість розширення ядра. Згодом інтерфейс написання розширень привернув до PHP безліч сторонніх розробників, що працюють над своїми модулями, що дало PHP можливість працювати з величезною кількістю баз даних, протоколів, підтримувати велике число API. Фактично, це і був головний ключ до успіху, та варто додати, що важливим кроком виявилася розробка нового, набагато могутнішого і повнішого синтаксису з підтримкою ООП [2].

Абсолютно нова мова програмування одержала нове ім'я. Розробники відмовилися від доповнення про персональне використання, яке було в абревіатурі PHP/FI. Мова була названа просто PHP – абревіатура, що містить рекурсивний акронім (англ. PHP: Hypertext Preprocessor – «PHP: Препроцесор Гіпертексту»).

До кінця 1998 року PHP використовувався десятками тисяч користувачів. Сотні тисяч веб-сайтів повідомляли про те, що вони працюють з використанням цієї мови. У той час PHP 3.0 був встановлений приблизно на 10 % веб-серверів Інтернету.

PHP 3.0 був офіційно випущений в червні 1998 року після 9 місяців публічного тестування.

До зими 1998 роки, практично відразу після офіційного виходу PHP 3.0, Енді Гутманс і Зів Сураські почали переробку ядра PHP. У завдання входило збільшення продуктивності складних додатків і поліпшення модульності базису коду PHP. Розширення дали PHP 3.0 можливість успішно працювати з набором баз даних і підтримувати велику кількість різних API і протоколів, але PHP 3.0 не мав якісної підтримки модулів і додатку працювали неефективно.

Новий «движок», названий Zend Engine (від імен творців, Зіва і Енді, також засновників Zend Technologies), успішно справлявся з поставленими завданнями і вперше був представлений у середині 1999 року. PHP 4.0, заснований на цьому движку і такий, що приніс з собою набір додаткових функцій, офіційно вийшов в травні 2000 року, майже через два роки після виходу свого попередника PHP 3.0. На додаток до поліпшення продуктивності, PHP 4.0 мав ще декілька ключових нововведень, таких як підтримка сесій, буферизація висновку, безпечніші способи обробки інформації, що вводиться користувачем, і декілька нових мовних конструкцій.

П’ята версія PHP була випущена розробниками 13 липня 2004 року. Зміни включають оновлення ядра Zend (Zend Engine 2), що істотно збільшило ефективність інтерпретатора. Введена підтримка мови розмітки XML. Повністю перероблені функції ООП, які стали багато в чому схожі з моделлю, використовуваною в Java. Зокрема, введена деструкція, відкриті, закриті і захищені члени і методи, остаточні члени і методи, інтерфейси і клонування об’єктів. Нововведення, проте, були зроблені з розрахунком зберегти найбільшу сумісність з кодом на попередніх версіях мови. На даний момент

найстабільнішими і часто використовуваними є саме версії 5.xx, навіть не дивлячись на те, що вже є dev-версія PHP 6.

У ній вже зроблено безліч нововведень, як, наприклад, виключення з ядра регулярних виразів POSIX і «довгих» суперглобальних масивів, видалення директив safe_mode, php_magic_quotes і register_globals з конфігураційного файлу php.ini. Також багато уваги приділено підтримці Юнікода. Завантажити поточну версію коду, що розробляється, для GNU/Linux/BSD і скомпільовані версії для Microsoft Windows можна на сайті PHP Snapshots.

2.3 Вибір фреймворка

Фреймворк – програмна платформа, яка визначає структуру програмної системи, програмне забезпечення, що полегшує розробку і об’єднання різних компонентів великого програмного проекту. Вживається також слово «каркас», а деякі автори використовують його в якості основного, в тому числі не базуючись взагалі на англомовному аналогу. Можна також говорити про каркасний підхід як про підхід до побудови програм, де будь-яка конфігурація програми будується з двох частин: перша, постійна частина – каркас, незмінний від конфігурації до конфігурації і несе в собі гнізда, в яких розміщується друга, змінна частина – змінні модулі (або точки розширення) [1].

Існує кілька веб-фреймворків реалізованих на PHP це:

- Zend Framework;
- CakePHP;
- Code Igniter;
- Kohana;
- Symfony;
- Yii;
- Laravel.

Вимоги до фреймворка такі:

- легкість створення нового проекту, читання та пошук документації;

- продуманість об'єктної моделі, можливість розширення;
- споживання пам'яті;
- швидкодія;
- зручність встановлення і налаштування.

Проаналізувавши недоліки і переваги вище перелічених феймворків, а також вимоги до них, був вибраний Laravel.

Laravel - безкоштовний веб-фреймворк з відкритим кодом, призначений для розробки з використанням архітектурної моделі MVC (англ. Model View Controller - модель-представлення-контролер). Laravel випущений під ліцензією MIT. Вихідний код проекту розміщується на GitHub [19].

В результаті опитування sitepoint.com в грудні 2016 про найпопулярніші PHP-фреймворки, Laravel зайняв місце самого багатообіцяючого проекту на 2014 рік.

Переваги Laravel:

- пакети (англ. Packages) - дозволяють створювати і підключати модулі в форматі Composer до додатка на Laravel. Багато додаткових можливостей вже доступні у вигляді таких модулів;
- Eloquent ORM - реалізація шаблону проектування ActiveRecord на PHP. Дозволяє строго визначити відносини між об'єктами бази даних. Стандартний для Laravel будівник запитів Fluent підтримується ядром Eloquent;
- логіка додатка - частина розроблювального додатка, оголошена або за допомогою контролерів, або маршрутів (функцій-замикань). Синтаксис оголошень схожий на синтаксис, використовуваний в каркасі Sinatra;
- зворотня маршрутизація пов'язує між собою посилання і маршрути що генеруються додатком, дозволяючи їх змінювати з автоматичним оновленням пов'язаних посилань. При створенні посилань за допомогою іменованих маршрутів Laravel автоматично генерує кінцеві URL;
- REST-контролери - додатковий шар для розділення логіки обробки GET- і POST-запитів HTTP;

- автозавантаження класів - механізм автоматичного завантаження класів PHP без необхідності підключення файлів їх визначень у include. Завантаження на вимогу запобігає завантаженню непотрібних компонентів; завантажуються тільки ті з них, які дійсно використовуються;
- укладачі уявлень (англ. View composers) - блоки коду, які виконуються при генерації подання (шаблону);
- міграції - система управління версіями для баз даних. Дозволяє пов'язувати зміни в коді програми зі змінами, які потрібно внести в структуру БД, що спрощує розгортання і оновлення програми;
- модульне тестування (юніт-тести) – відіграє дуже велику роль в Laravel, який сам по собі містить велику кількість тестів для запобігання регресій (помилок внаслідок поновлення коду або виправлення інших помилок);
- сторінкове виведення (англ. Pagination) – спрощує генерацію сторінок, замінюючи різні способи вирішення цього завдання єдиним механізмом, вбудованим в Laravel [23].

2.4 Вибір системи управління базами даних

База даних – впорядкований набір логічно взаємопов’язаних даних, що використовуються спільно та призначені для задоволення інформаційних потреб користувачів. У технічному розумінні включно й система керування БД. Головне завдання БД – гарантоване збереження значних обсягів інформації (так звані записи даних) та надання доступу до неї користувачеві або ж прикладній програмі. Таким чином, БД складається з двох частин: збереженої інформації та системи керування нею. З метою забезпечення ефективності доступу записи даних організовують як множину фактів (елемент даних) [20].

Для веб-розробки доцільно використовувати MySQL оскільки це СУБД з відкритим кодом.

MySQL – вільна СУБД (система управління базами даних). Розробку та підтримку MySQL здійснює корпорація Oracle. Продукт поширюється як під GNU General Public License, так і під власною комерційною ліцензією. Крім цього розробники створюють функціональність за замовленням ліцензійних користувачів, саме завдяки такому замовленню майже в найраніших версіях з'явився механізм реплікації [14].

Переваги MySQL:

– простота у встановленні та використанні;

– підтримується необмежена кількість користувачів, що одночасно працюють із БД;

– кількість рядків у таблицях може досягати 50 млн.;

– висока швидкість виконання команд;

– наявність простої і ефективної системи безпеки.

2.5 Вибір середовища програмування

Середовище розробки – це комп’ютерна програма, що допомагає програмістові розробляти нове програмне забезпечення чи модифікувати (удосконалювати) вже існуюче.

Інтегровані середовища розробки зазвичай складаються з редактора сирцевого коду, компілятора або інтерпретатора та засобів автоматизації збірки. Іноді сюди також входять системи контролю версій, засоби для профілювання, а також різноманітні засоби та утиліти для спрощення розробки графічного інтерфейсу користувача. Багато сучасних інтегрованих середовищ розробки також включають оглядач класів, інспектор об’єктів та діаграм ієрархії класів для використання об'єктно-орієнтованого підходу у розробці програмного забезпечення. Сучасні ІСР часто підтримують розробку на декількох мовах програмування.

Для зручної розробки веб-проекту знадобляться такі засоби програмування:

- локальний сервер;
- візуальне середовище програмування;
- програма керування БД.

2.5.1 Вибір локального сервера. Після детального аналізу локальних серверів був вибраний Open Server.

Open Server – це портативна серверна платформа і програмне середовище, створена спеціально для веб-розробників з урахуванням їх рекомендацій і побажань. Програмний комплекс має багатий набір серверного програмного забезпечення, зручний, багатофункціональний продуманий інтерфейс, володіє потужними можливостями з адміністрування та налаштування компонентів. Платформа широко використовується з метою розробки, налагодження і тестування веб-проектів, а так само для надання веб-сервісів в локальних мережах [23].

2.5.2 Вибір візуального середовища програмування. Візуальне середовище програмування – це інтегроване середовище розробки програмних засобів, яке містить редактор вихідного коду, компілятор або інтерпретатор, засоби автоматизації збірки та засоби для спрощення розробки графічного інтерфейсу користувача. Середовища для візуального програмування також надають змогу конструювати програми шляхом оперування графічними об’єктами. Багато сучасних візуальних середовищ програмування використовуються для реалізації принципів об’єктно-орієнтованого підходу у розробці програмного забезпечення [19].

Візуальним середовищем програмування був вибраний PhpStorm (рис. 2.1). PhpStorm являє собою інтелектуальний редактор для PHP, HTML і JavaScript з можливостями аналізу коду на льоту, запобігання помилок у сирцевому коді і автоматизованими засобами рефакторинга для PHP і JavaScript. Автодоповнення коду в PhpStorm підтримує специфікацію PHP 5.3, 5.4, 5.5 та 5.6, включаючи генератори, співпрограми, простори імен, замикання,

типажі і синтаксис коротких масивів. Присутній повноцінний SQL-редактор з можливістю редагування отриманих результатів запитів.

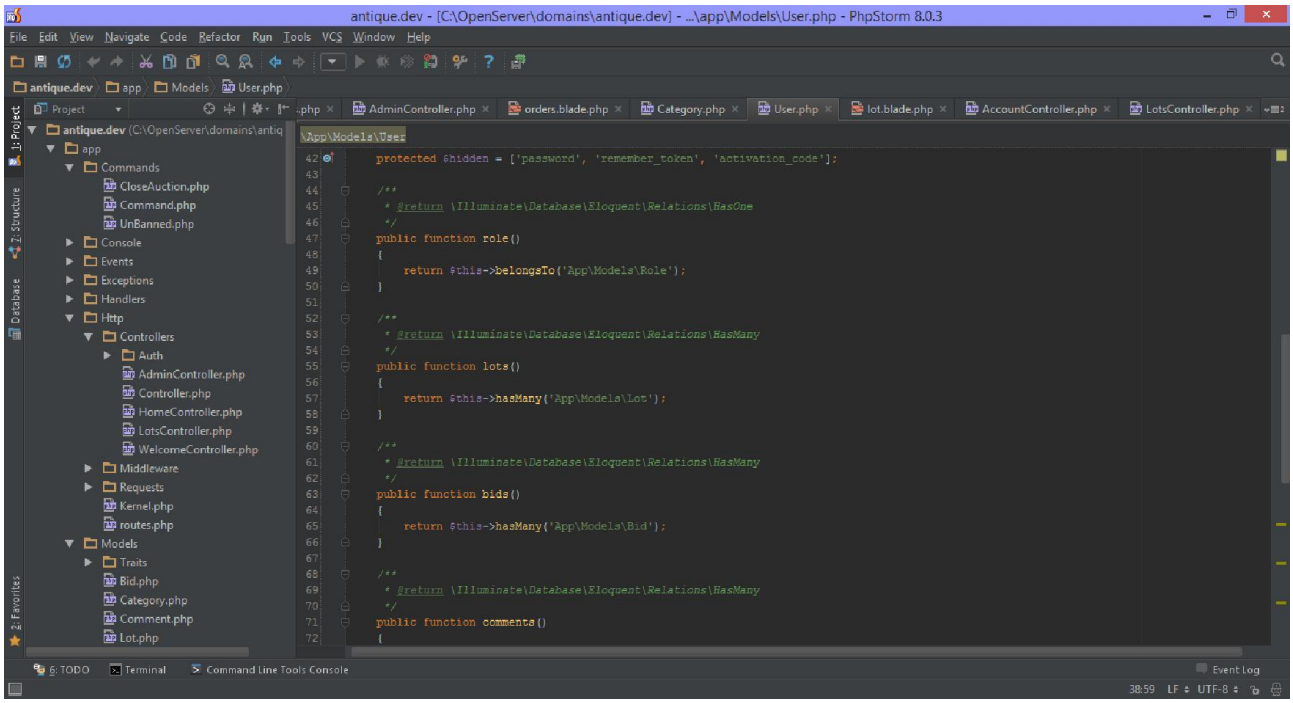

Рисунок 2.1 – Візуальне середовище програмування PhpStorm.

2.5.3 Вибір програми управління БД. HeidiSQL (рис. 2.2), перед тим відома як MySQL-Front – вільний відкритий клієнт, або *фронтенд*, для управління базами даних, розроблений німецьким програмістом Анзґаром Бекером (Ansgar Becker) та кількома іншими розробниками, Написаний на Delphi, підтримує з'єднання та роботу з MySQL, їхні форки, таких як MariaDB та Percona, а також Microsoft SQL Server, починаючи з версії 7.0. Щоб управляти базою даних з HeidiSQL, користувач має увійти на локальний або віддалений сервер MySQL з прийнятним паролем, створивши сесію. В рамках цієї сесії користувач може управляти базами даних MySQL на сервері MySQL, і від'єднатися після закінчення роботи. Можливості програми цілком достатні для більшості операцій із загальними та просунутими базами даних, таблицями та записами, але розробка залишається у активному стані, щоб забезпечити повну функціональність, котра очікується від фронтенду MySQL.

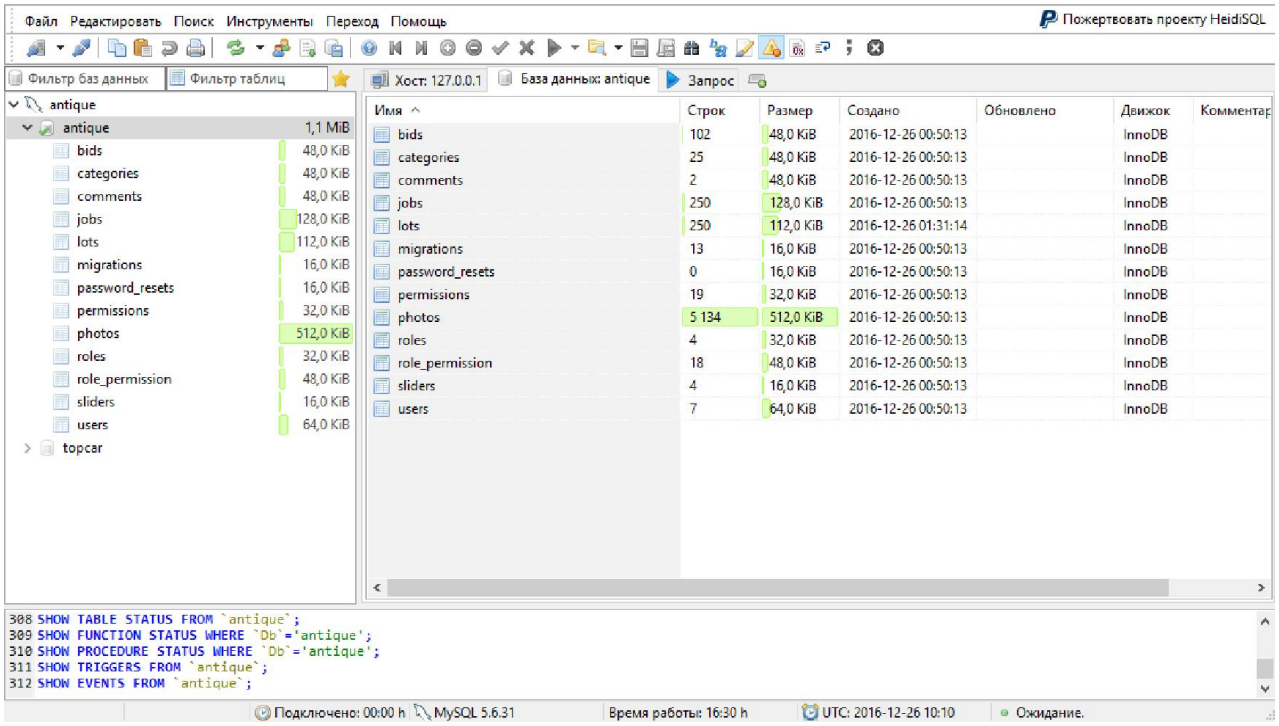

Рисунок 2.2 – програма керування БД HeidiSQL.

2.6 Асоціативний пошук

Розробники постаралися відтворити динамічну, асоціативну природу людського мислення. Вони виходили з того, що при прийнятті рішень люди, використовують для обробки інформації неієрархічні способи аналізу. Обдумуючи задачу, кожна людина йде до висновку своїм шляхом. Користувачам надається можливість аналізувати дані відповідно до власних розумових процесів.

Традиційні методи пошуку і фільтрації інформації були розроблені для бібліотечних баз даних, тобто для інформаційних ресурсів обмеженого обсягу і заздалегідь відомої структури. Створення глобальної мережі і вихід Web за рамки інтересів наукового співтовариства призвели до того, що число постачальників інформації стало стрімко рости, при тому, що опублікована ними інформація не мала однорідної структури. Наступний інформаційний вибух став викликом стандартним інформаційним технологіям. Нові масштаби з одного боку зробили аутсайдерами деякі раніше конкурентоспроможні

інтелектуальні технології, а з іншого – стимулювали інтенсивні дослідження в області лінгвістичних та імовірнісних методів обробки текстової інформації та нових методів навігації в неоднорідному інформаційному морі.

Асоціативний пошук ґрунтується на двох ключових принципах: всі дані зберігаються в пам'яті комп'ютера (ОЗУ); обчислення виконуються в реальному часі. Ці архітектурні рішення підкріплюються двома важливими тенденціями в розвитку комп'ютерів. Перша – перехід від 32- до 64-розрядних обчислень, що призвело до експоненціального збільшенню розміру ОЗУ комп'ютера. В даний час можна придбати сервери, де ОЗУ досягає 512 Гбайт, а не так давно, в 2005 році, більшість серверів розташовувало ОЗУ ємністю 4 Гбайт. Завдяки збільшенню доступною пам'яті стало можливо перемістити сховище даних з диска безпосередньо в ОЗУ. Друга обставина – повсюдне поширення багатоядерних процесорів. В даний час широко застосовуються сервери, які мають 8 процесорів і в цілому 48 ядер. Зміни привели до серйозного збільшення обчислювальної потужності додатків, орієнтованих на паралельні обчислення. Високопродуктивні сервери пропонуються менш ніж за 50 000 дол. Зовсім недавно для досягнення такого рівня обчислювальних можливостей були потрібні спеціально сконструйовані комп'ютери вартістю сотні тисяч і навіть мільйони доларів [19].

Архітектура зберігання і обробки інформації в оперативній пам'яті (in-memory), дозволяє маніпулювати величезними обсягами даних, підтримуючи високий рівень інтерактивності. Стиснення даних при завантаженні в пам'ять дозволяє зберігати дані в ОЗУ ефективніше, ніж на диску, як це прийнято в традиційних реляційних базах даних. При передачі даних в пам'ять система будує карту посилань між елементами даних, щоб спростити візуалізацію їх зв'язків [22].

Завдяки здатності виконувати обчислення в реальному часі можна швидко розраховувати складні заходи і метрики. В новітніх системах закладена можливість розподіляти обчислення між всіма доступними ядрами процесорів і управляти робочим навантаженням одночасно для великої кількості

користувачів. Крім того, платформа зберігає в кеш-пам'яті результати всіх запитів, тому типові обчислення виконуються мінімальну кількість разів.

РОЗДІЛ 3
ПРАКТИЧНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Структура сайту

Структура сайту – це його розділи, підрозділи і сторінки. А також навігація, яка забезпечує доступ до них, тобто різні меню, перехресні посилання і карта сайту. Після ознайомлення з постановкою задачі було прийнято рішення створити сайт з такою структурою.

Структура сайту наведена на рис. 3.1.

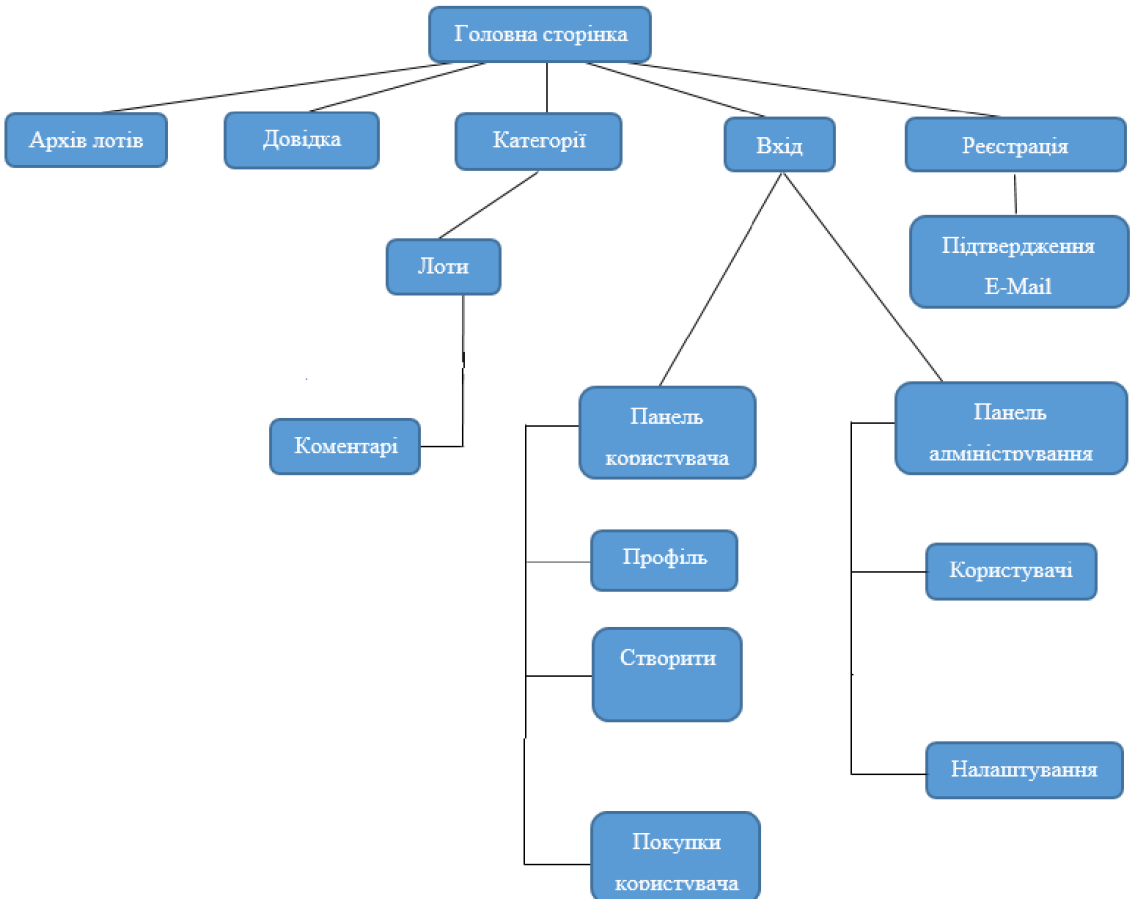

Рисунок 3.1 – Структура сайту.

3.2 База даних

3.2.1 Таблиця користувачів. Для збереження даних користувачів необхідно створити таблицю, users у якій будуть зберігатися логін користувача, прізвище, ім’я, по батькові, адреса електронної пошти, номер телефону, країна і місто проживання, пароль (рис. 3.2).

	#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
☐	1	ID	bigint(20)		UNSIGNED	Нет	*Нет*		AUTO_INCREMENT	✎ ✕ ▾
☐	2	user_login	varchar(60)	utf8mb4_unicode_ci		Нет				✎ ✕ ▾
☐	3	user_pass	varchar(255)	utf8mb4_unicode_ci		Нет				✎ ✕ ▾
☐	4	user_nicename	varchar(50)	utf8mb4_unicode_ci		Нет				✎ ✕ ▾
☐	5	user_email	varchar(100)	utf8mb4_unicode_ci		Нет				✎ ✕ ▾
☐	6	user_url	varchar(100)	utf8mb4_unicode_ci		Нет				✎ ✕ ▾
☐	7	user_registered	datetime			Нет	0000-00-00 00:00:00			✎ ✕ ▾
☐	8	user_activation_key	varchar(255)	utf8mb4_unicode_ci		Нет				✎ ✕ ▾
☐	9	user_status	int(11)			Нет	0			✎ ✕ ▾
☐	10	display_name	varchar(250)	utf8mb4_unicode_ci		Нет				✎ ✕ ▾

Рисунок 3.2 – Таблиця «users»

3.2.2 Таблиці ролей та дозволів. Для збереження даних про ролі і дозволи ролей необхідно створити три таблиці:

– roles – збереження імені та опису ролей (рис. 3.3);

– permissions – збереження імені та опису дозволів (рис. 3.4);

– role_permission – збереження зв’язків між таблицями ролей та дозволів (рис. 3.5).

	#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
☐	1	option_id	bigint(20)		UNSIGNED	Нет	*Нет*		AUTO_INCREMENT	✎ ✕ ▾
☐	2	option_name	varchar(191)	utf8mb4_unicode_ci		Нет				✎ ✕ ▾
☐	3	option_value	longtext	utf8mb4_unicode_ci		Нет	*Нет*			✎ ✕ ▾
☐	4	autoload	varchar(20)	utf8mb4_unicode_ci		Нет	yes			✎ ✕ ▾

Рисунок 3.3 – Таблиця «roles»

3.2.5 Модель схеми БД. Схема баз даних – це структура системи баз даних описана формальною мовою, яка підтримується системою управління баз даних (СУБД) і відноситься до організації даних для створення плану побудови бази даних з розподілом на таблиці. Формально схема баз даних являє собою набір формул (правил), які називаються обмеженнями цілісності. Обмеження цілісності забезпечують сумісність між всіма частинами схеми. Всі обмеження виражаються однією мовою [20].

Поняття схеми бази даних відіграє ту ж роль, що і поняття теорії в численні предикатів. Модель цієї «теорії» точно відповідає базі даних, яку можна побачити в будь-який момент часу як математичний об'єкт. Таким чином, схема може містити формули, що представляють обмеження цілісності спеціально для застосунків і обмеження спеціально для типу бази даних, які виражені на одній мові баз даних. В реляційній базі даних, схема визначає таблиці, поля, відношення, індекси, пакети, процедури, функції, черги, тригери, типи даних, послідовності, матеріалізовані уявлення, синоніми, посилання баз даних, каталоги, Java, XML-схеми та інші елементи.

Схема, як правило, зберігається в словнику даних. Хоча схема визначена в тексті мовою бази даних, цей термін часто використовується для графічного позначення структури бази даних. Іншими словами, схема — це структура бази даних яка визначає об'єкти в базі даних.

Схема бази даних сайту наведена на рис. 3.4

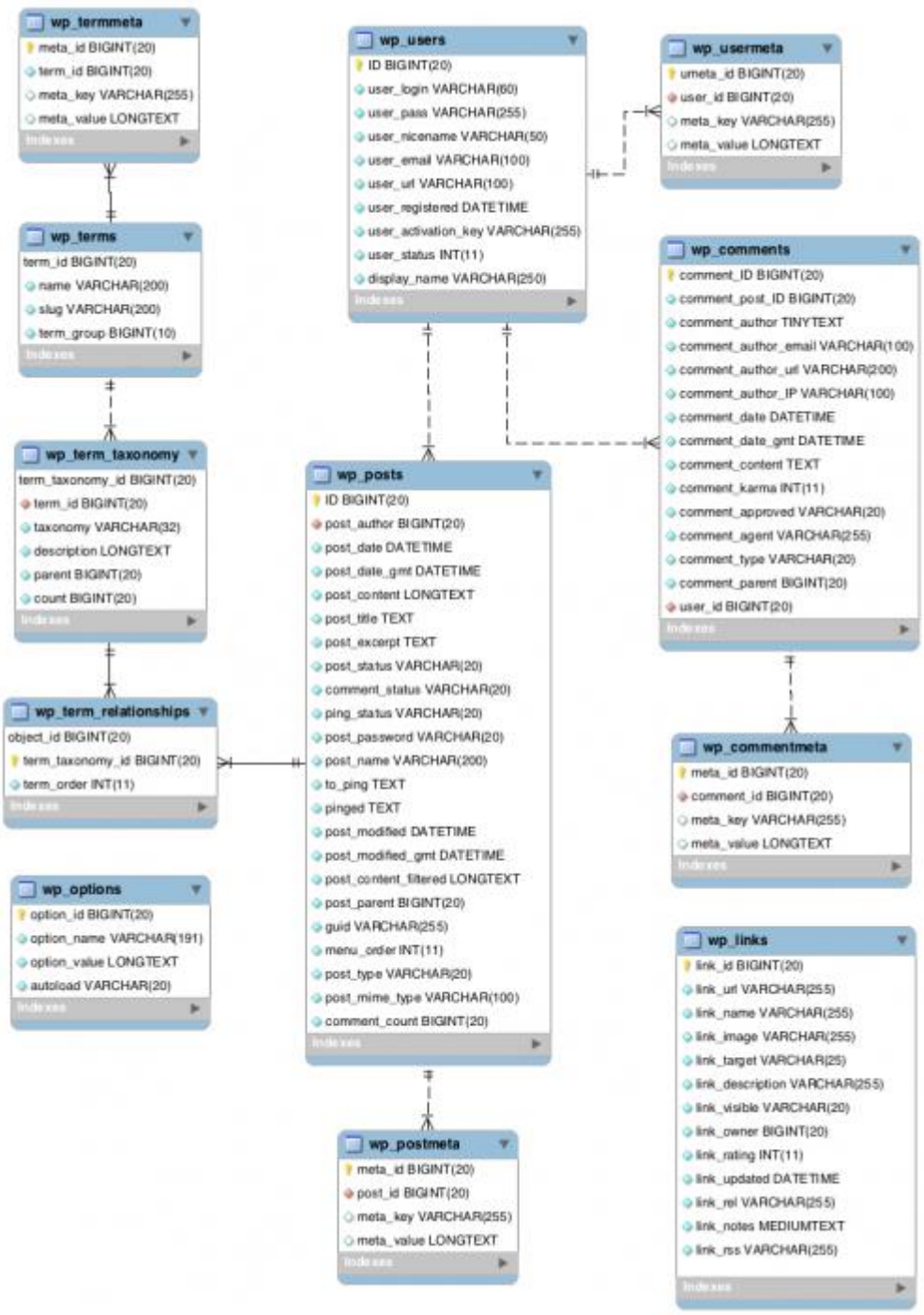

Рисунок 3.4 – Схема бази даних

3.3 Системний аналіз

3.3.1 Діаграми варіантів використання сайту. Діаграма прецедентів — в UML, діаграма, на якій зображено відношення між акторами та прецедентами в системі. Також, перекладається як діаграма варіантів використання.

Діаграма прецедентів є графом, що складається з множини акторів, прецедентів (варіантів використання) обмежених границею системи, асоціацій між акторами та прецедентами, відношень серед прецедентів, та відношень узагальнення між акторами. Діаграми прецедентів відображають елементи моделі варіантів використання [15].

Суть даної діаграми полягає в наступному: проектована система представляється у вигляді безлічі сутностей чи акторів, що взаємодіють із системою за допомогою так званих варіантів використання. Варіант використання (англ. use case) використовують для описання послуг, які система надає актору. Іншими словами, кожен варіант використання визначає деякий набір дій, який виконує система при діалозі з актором. При цьому нічого не говориться про те, яким чином буде реалізована взаємодія акторів із системою.

Діаграми варіантів використання сайту наведені на рис. 3.13.

До базових елементів розглянутої діаграми відносяться варіант використання, актор і інтерфейс.

Варіант використання застосовується для специфікації загальних особливостей поведінки системи або іншої сутності без розгляду її внутрішньої структури (наприклад, оформлення замовлення на купівлю товару, отримання інформації про кредитоспроможність клієнта, відображення графічної форми на екрані монітора).

Актор – це зовнішня по відношенню до моделюється системі сутність, яка взаємодіє з системою і використовує її функціональні можливості для вирішення певних завдань. При цьому актори служать для позначення узгодженого безлічі ролей, які можуть відігравати користувачі в процесі взаємодії з проектованою системою. Ім'я актора має бути достатньо інформативним з точки зору семантики, наприклад клієнт банку, продавець магазину, пасажир авіарейсу, водій автомобіля, стільниковий телефон.

В даному випадку в системі визначені такі актори:

– гість, який може лише переглядати лоти;

– зареєстрований користувач, який може створювати лоти, робити ставки і залишати коментарі;

– модератор, який може блокувати, розблоковувати та переглядати профілі користувачів і модерувати лоти;

– адміністратор має найбільш широкі права: може виконувати всі дії, перераховані вище, а також редагувати та видаляти профілі користувачів, управляти ролями і дозволами, управляти категоріями аренди [15].

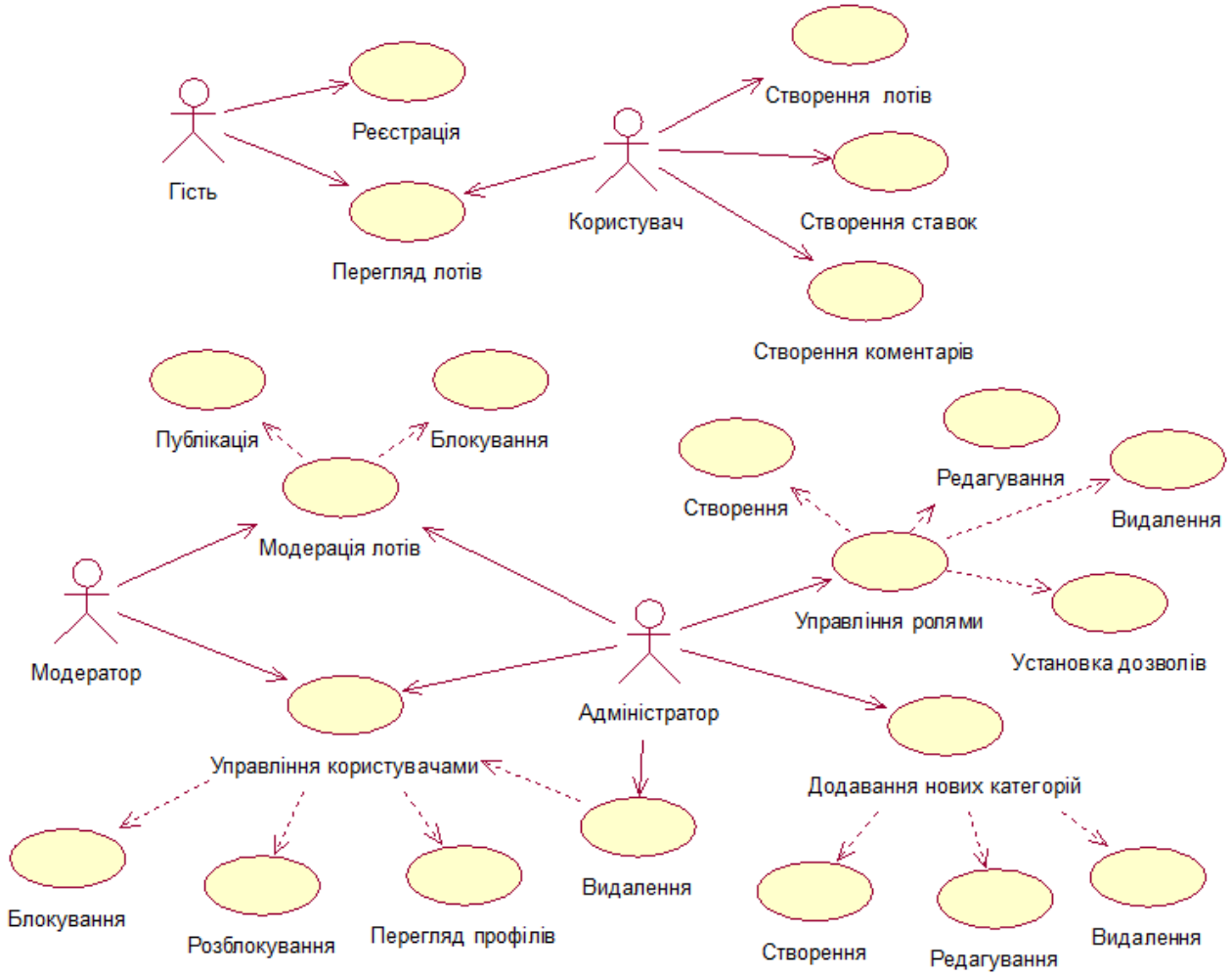

Рисунок 3.13 – Діаграми варіантів використання сайту

Так як в загальному випадку актор завжди знаходиться поза системою, його внутрішня структура ніяк не визначається. Для актора має значення тільки його зовнішнє уявлення, тобто то, як він сприймається з боку системи. Актори взаємодіють з системою за допомогою передачі і прийому повідомлень від варіантів використання. Повідомлення являє собою запит актором сервісу від

системи і отримання цього сервісу. Ця взаємодія може бути виражене за допомогою асоціацій між окремими акторами і варіантами використання або класами. Крім цього, з акторами можуть бути пов'язані інтерфейси, які визначають, яким чином інші елементи моделі взаємодіють з цими акторами.

Інтерфейс служить для специфікації параметрів моделі, які видимі ззовні без вказівки їх внутрішньої структури. У діаграмах варіантів використання інтерфейси визначають сукупність операцій, які забезпечують необхідний набір сервісів або функціональності для акторів. Інтерфейси не можуть містити ні атрибутів, ні станів, ні направлених асоціацій. Вони містять лише операції без вказівки особливостей їх реалізації. Формально інтерфейс еквівалентний абстрактному класу без атрибутів і методів з наявністю тільки абстрактних операцій.

3.3.2 Процес реєстрації. Реєстрація здійснюється перед початком роботи або на стадії замовлення. Для успішної реєстрації необхідно:

1. Активізувати посилання Реєстрація.
2. Після цього користувач потрапляє на спеціальну сторінку Реєстрації
3. У вікні Реєстрація користувач заповняє наступні поля – логін користувача для входу в систему, прізвище, ім’я, по батькові, адресу електронної пошти, номер телефону, країну і місто проживання, пароль. Після цього користувач натисне на кнопку Зареєструватися.
4. У разі правильного виконання дій при реєстрації, система видає повідомлення про успішну реєстрацію і необхідність підтвердити адресу електронної пошти. Для цього на адресу електронної пошти, вказаній при реєстрації, буде відправлене спеціальне повідомлення з посиланням, яке містить код активації, перейшовши по якому профіль користувача активується. Якщо користувач не отримав повідомлення, він має змогу повторного відправлення коду активації.

Діаграма активності процесу реєстрації зображена на рис. 3.14.

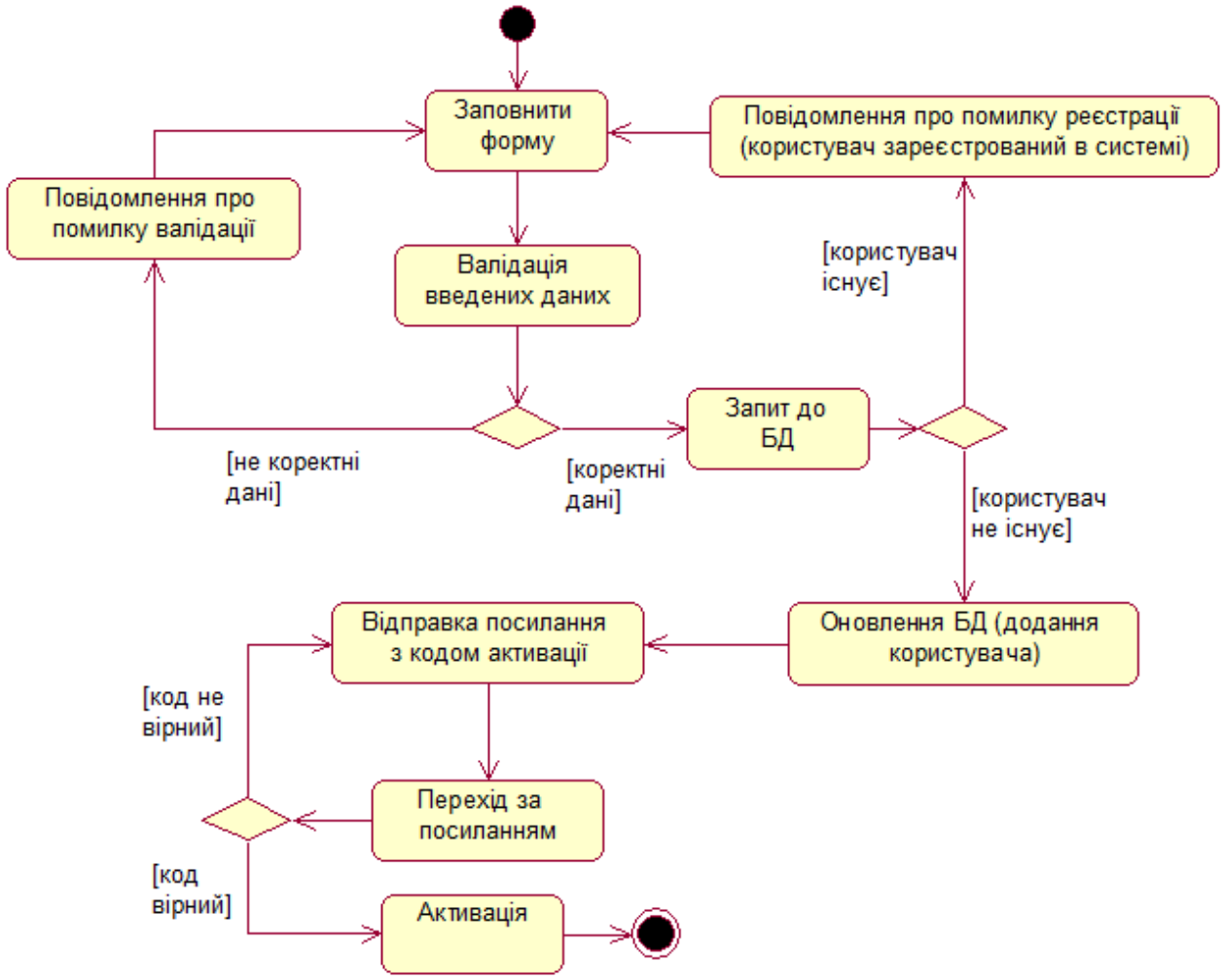

Рисунок 3.14 – Діаграма активності процесу реєстрації

3.3.3 Процес авторизації. Авторизація здійснюється на початку кожного сеансу роботи з онлайн-аренди. Для здійснення авторизації необхідно:

1. Активізувати посилання Вхід.
2. Після цього користувач потрапить на спеціальну сторінку Вхід. У вікні Вхід до системи необхідно ввести ім’я або адресу електронної пошти та пароль. Після цього потрібно натиснути кнопку Вхід.
3. В разі правильного виконання дій, система видає повідомлення про успішну авторизацію сеансу роботи.

Діаграма активності процесу авторизації зображена на рис. 3.15.

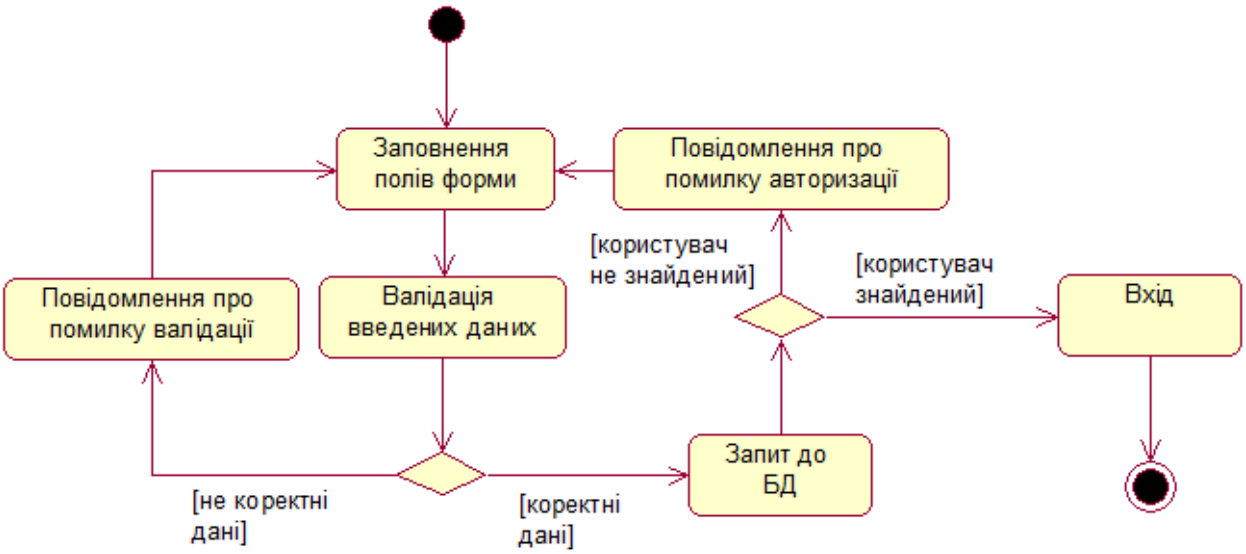

Рисунок 3.15 – Діаграма активності процесу авторизації

3.3.4 Процес створення лоту. Створення лотів доступне тільки авторизованим користувачам. Для створення лоту необхідно:

1. Перейти в панель користувача і активізувати посилання Створити лот.

2. У вікні створення лоту користувач заповнює наступні поля – категорія товару, назва лота, ціна, крок, дата завершення, матеріал, стан, місце знаходження, оплата, доставка, короткий опис, фото лоту. Після цього користувач натисне на кнопку Замовлення.

Діаграма активності процесу створення лоту зображена на рис. 3.16.

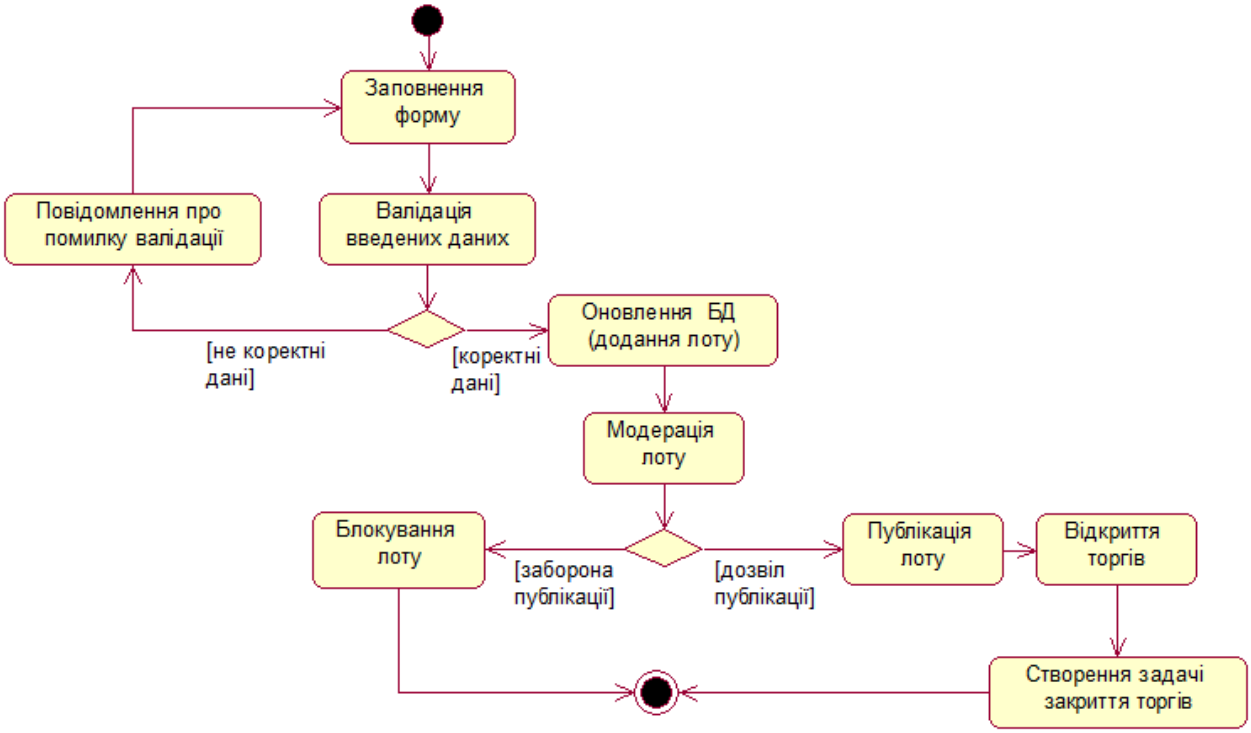

Рисунок 3.16 – Діаграма активності процесу

3.3.5 Процес замовлення.

Замовлення доступне тільки зареєстрованим користувачам. Для створення заявки необхідно:

1. Вибрати необхідний лот і перейти на сторінку детального перегляду.
2. У вікні детального перегляду лоту користувач може переглянути більше інформаціх і вибрати товар, якщо йому підходить.
3. У разі правильного виконання дій, система видає повідомлення про успішне створення замовлення.

Діаграма активності процесу замовлення зображена на рис. 3.17.

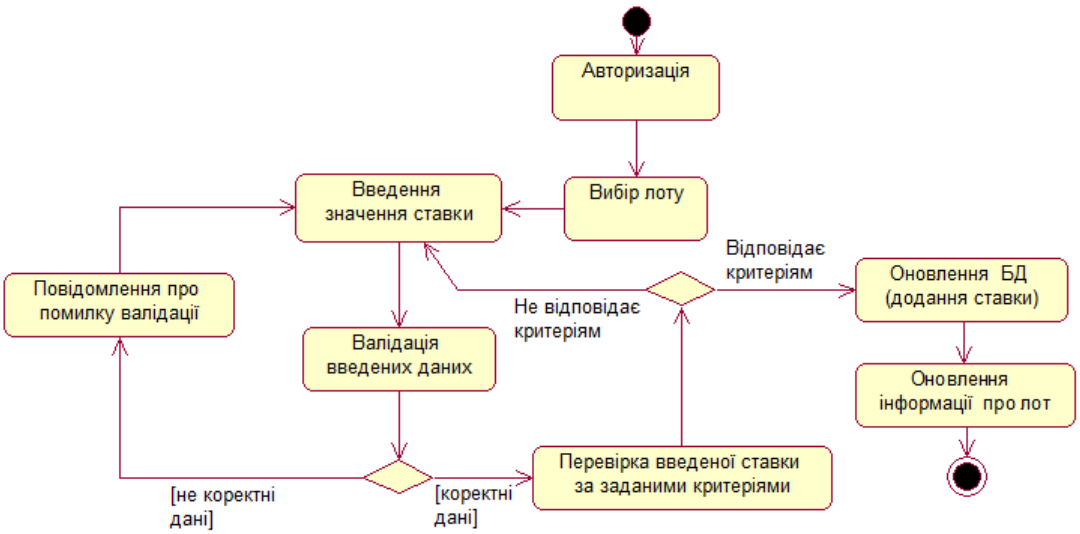

Рисунок 3.17 – Діаграма активності процесу створення замовлення

3.3.6 Процес створення коментаря. Створення коментарів доступне тільки авторизованим користувачам. Для створення коментаря необхідно:

1. Вибрати необхідний лот і перейти на сторінку детального перегляду.
2. У вікні детального перегляду лоту користувач заповнює поле вводу коментаря. Після цього користувач натисне кнопку Відправити.
3. У разі правильного виконання дій при створенні коментаря, система видає повідомлення про успішне створення коментаря.

Діаграма активності процесу створення коментаря зображена на рис. 3.18.

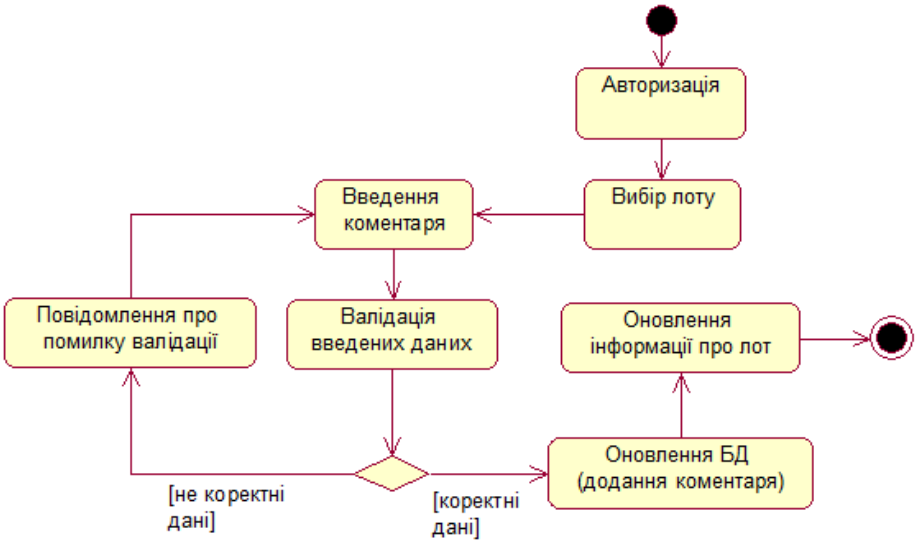

Рисунок 3.18 – Діаграма активності процесу створення коментаря

3.4 Захист інофмації

3.4.1 Використання SSL. Протокол безпечних з'єднань (Secure Sockets Layer, SSL) – це протокол, що забезпечує безпечний зв'язок між сервером і клієнтом. SSL не закриває доступ до передачі даних, а просто їх шифрує. Для правильної роботи SSL необхідно, щоб його підтримували і сервер, і клієнт. Frontpage і Microsoft Internet Explorer, а також Netscape Navigator підтримують SSL. Frontpage також дозволяє створювати посилання, що починаються з https:// замість http://; такі посилання означають безпечне з'єднання по протоколу SSL [4].

Перш ніж рухатися далі, ви повинні з'ясувати, чи підтримує SSL сервер, з яким ви збираєтеся працювати. Зробити це можна декількома способами:

- довідайтеся це в адміністратора сервера. Адміністратор може відключити підтримку SSL, так що бажано з'ясувати це заздалегідь.
- якщо ви використовуєте Microsoft Internet Information Server чи один із серверів Netscape (Commerce Server, FastTrack Server чи Enterprise Server), то вам, швидше за все, турбуватися нема про що. Однак і в даному

випадку вам не зашкодить з'ясувати це в адміністратора (майте на увазі, що Microsoft Personal Web Server не підтримує SSL)

- якщо ви хочете перевірити це самостійно, то створіть новий сайт у Провіднику Frontpage. Укажіть сервер, задайте назву сайта й установіть прапорець Connect Using SSL. Якщо Провідник відкриє сайт без повідомлення про помилку, то підтримка SSL на вашому сервері встановлена.
- Якщо на сервері й у Frontpage активізований SSL, то зв'язок між клієнтом Frontpage і сервером, включаючи будь-які команди від чи Провідника Редактора, є безпечним. Це означає, що весь потік інформації між Frontpage і Web-сервером, де б вони не були розташовані, шифрується. Такий захист зручний у декількох ситуаціях:
- Якщо ви знаходитеся в шляху і вам потрібно внести зміни в сайт, що знаходиться на сервері у вашому будинку, то ви можете відкрити сайт, зробити необхідні зміни і зберегти їх на сервері;
- Якщо ваша організація має більше одного офісу і тільки один Web-сервер, то можна вносити зміни в сайт із віддаленого офісу;
- Якщо ваш корпоративний чи персональний Web-сайт розташований на сервері провайдера послуг Internet (як правило, такі сервери обслуговують декілька Web-сайтів), то ви зможете робити зміни у вашому сайті за допомогою Frontpage, не побоюючись, що інформація буде перехоплена і переглянута яким-небудь хакером, оскільки всі передані дані зашифровані [4].

3.4.2 Повноваження і ролі користувачів. Ролі дають можливість створити декілька груп користувачів (адміністратори, модератори, звичайний користувач). Повноваження дають вам можливість установки для ролей прав доступу до вашого сайту. З їхньою допомогою ви можете визначати коло людей, що мають привілеї перегляду, чи авторства адміністрування сайту незалежно від того, для якої мережі – Intranet чи Internet – ви розробляєте сайт.

Якщо ви установили дозвіл доступу до певної частини сайту тільки для адміністратора, то очевидно, користувачі з іншими ролями доступу до вашого сайту мати не будуть. Наприклад, ви можете визнати за необхідне обмежити на час розробки доступ до сайту; коли ваш сайт буде готовий, то ви просто зміните повноваження користувачів і надасте доступ до сайту всім бажаючим.

Для користувачів сайту були створенні ролі і права доступу, тобто тільки користувачі з певними ролями мають доступ до відповідних розділів сайту. Паролі користувачів шифруються MD5

3.5 Сторінки інформаційної системи

3.5.1 Головна сторінка. Головна сторінка містить такі основні вкладки:

- головна – головна сторінка;
- нові пропозиції;
- аккаунт – сторінка реєстрації та авторизації;
- меню навігації.

На головній сторінці розміщені рядок пошуку, слайдер та лоти які були додані останніми (рис. 3.20).

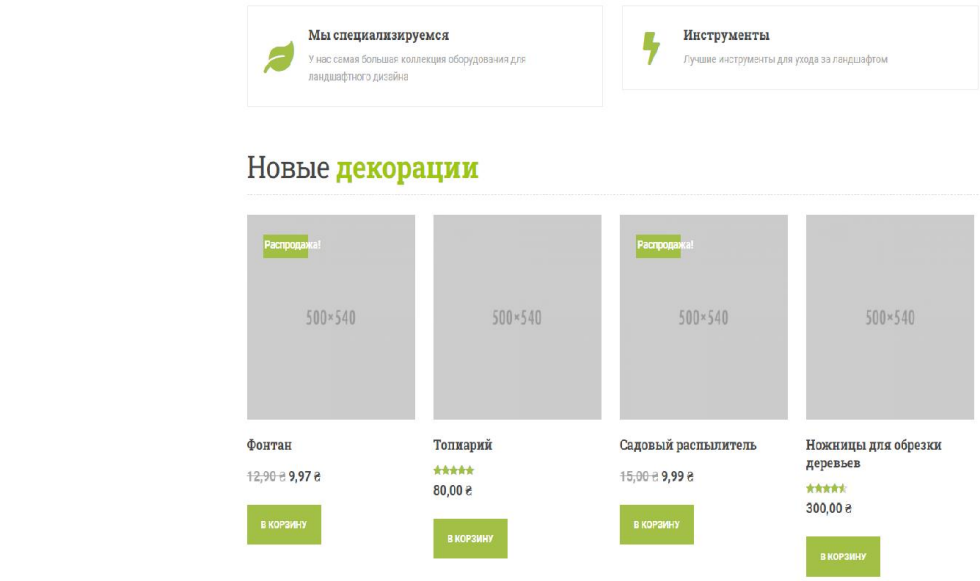

Рисунок 3.20 – Головна сторінка

Співробітництво (рис. 3.21)

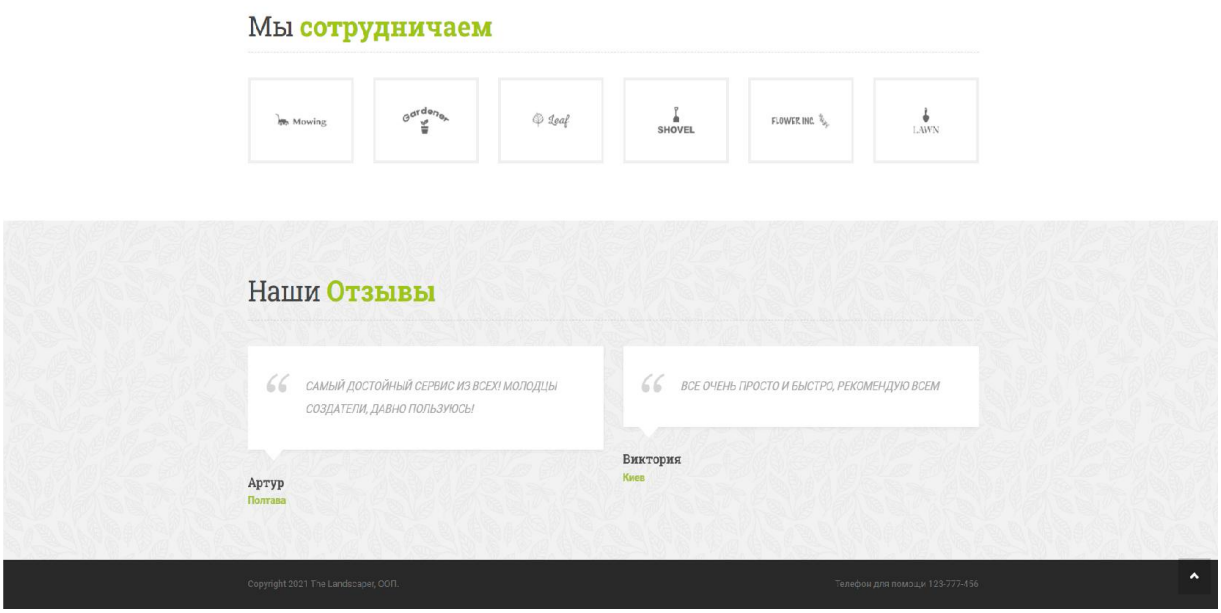

Рисунок 3.21 – Співробітництво

3.5.2 Меню навігації. Бокове меню зі списком категорій (рис. 3.22), при виборі певної категорії користувач перейде на сторінку даної категорії (рис. 3.23).

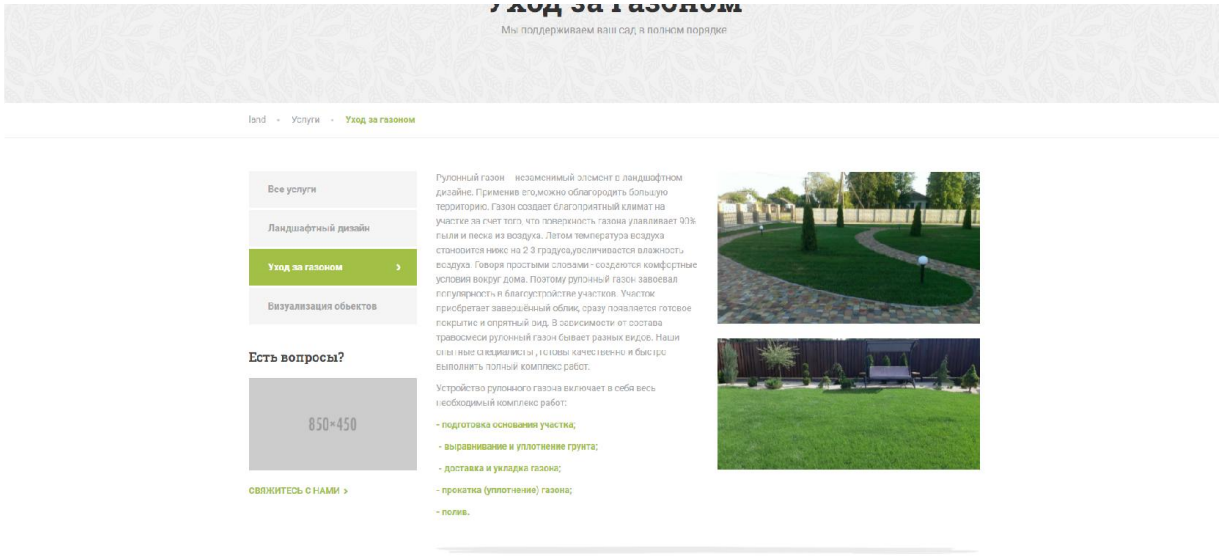

Рисунок 3.22 – Меню категорій

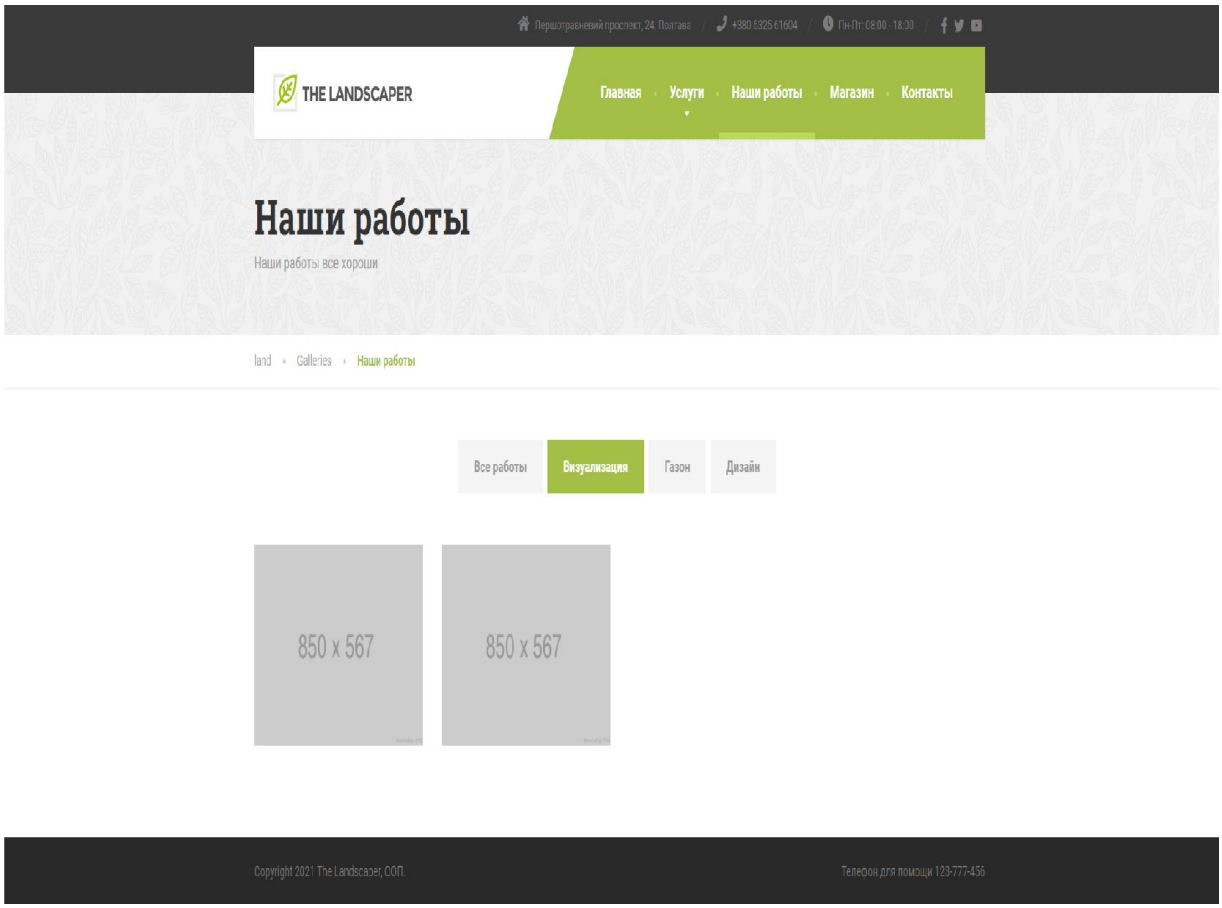

Рисунок 3.23 – Список послуг

Пошук необхідних продуктів показаний на рис. 3.24., рис. 3.25.,

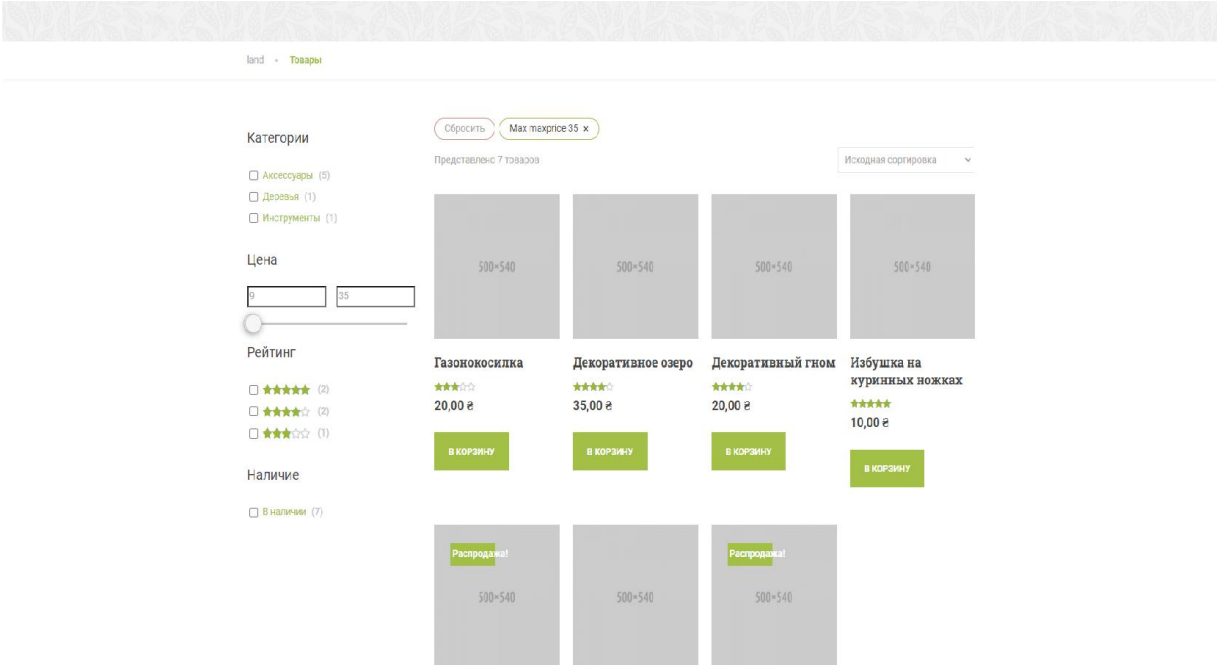

Рисунок 3.24 – Пошук необхідних продуктів

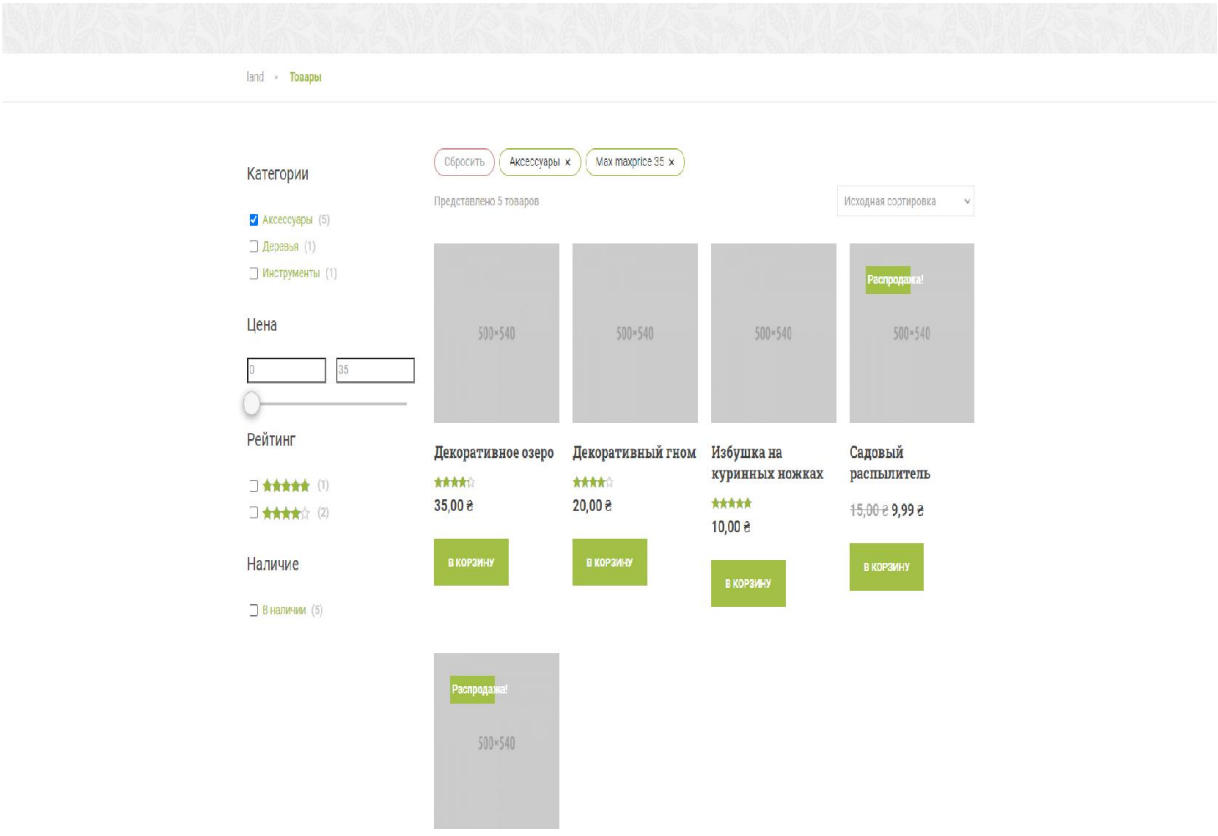

Рисунок 3.25 – Пошук необхідних продуктів

3.5.3 Сторінки реєстрації та авторизації. На сторінці реєстрації потенційний користувач може зареєструватися без проблем (рис. 3.24), на сторінці авторизації зареєстрований користувач може авторизуватися на сайті (рис. 3.25).

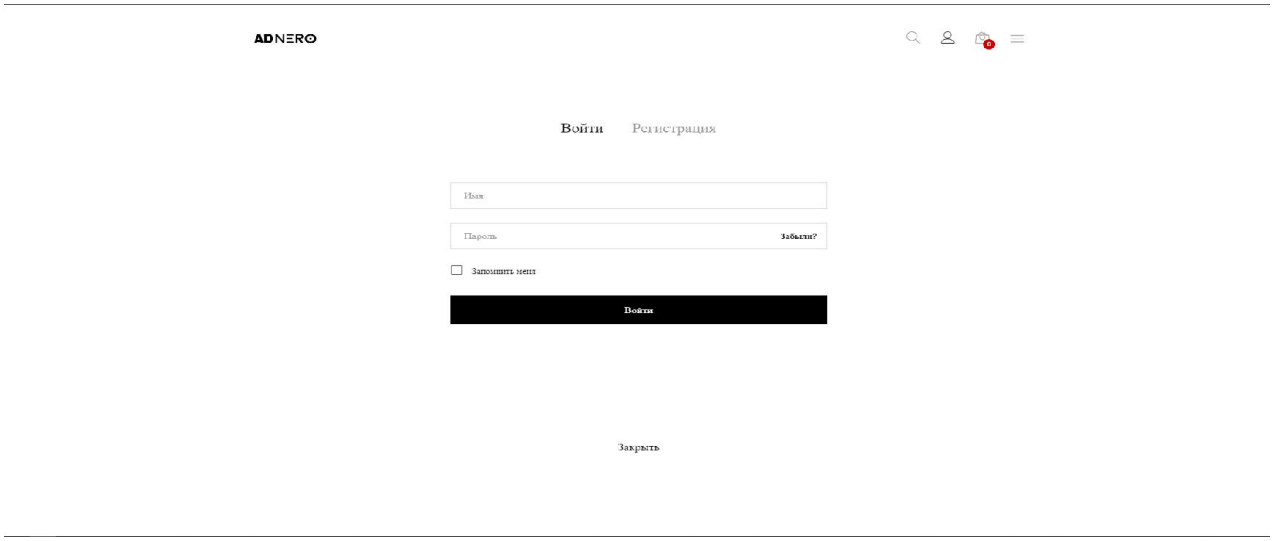

Рисунок 3.24 – Сторінка авторизації

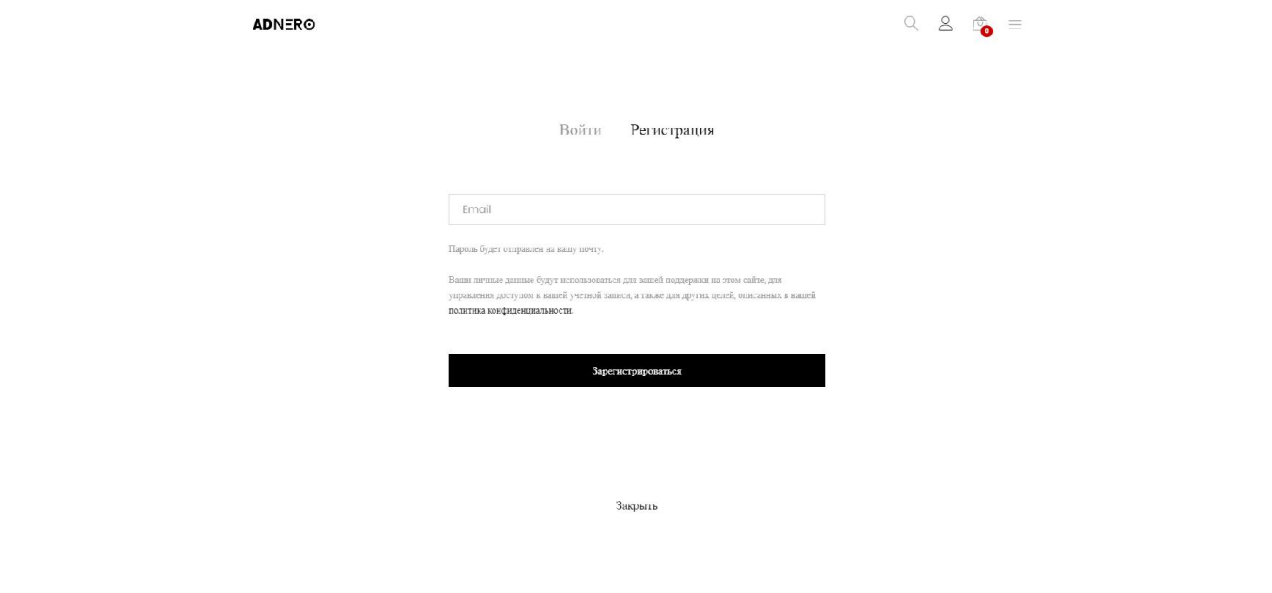

Рисунок 3.25 – Сторінка регистрации

3.5.4 Панель користувача. Головна сторінка панелі користувача дозволяє редагувати його дані (рис. 3.27).

Структура сторінок панелі користувача.

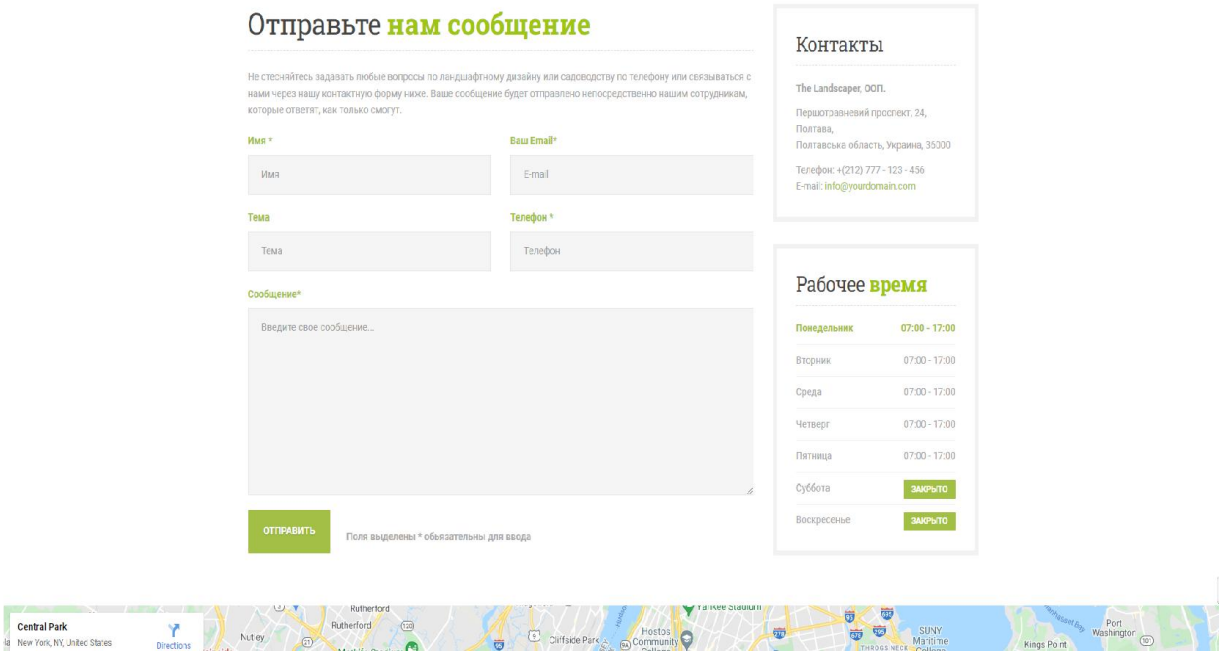

Рисунок 3.27 – Головна сторінка панелі користувача

Сторінка замовлення (корзина) (рис. 3.28).

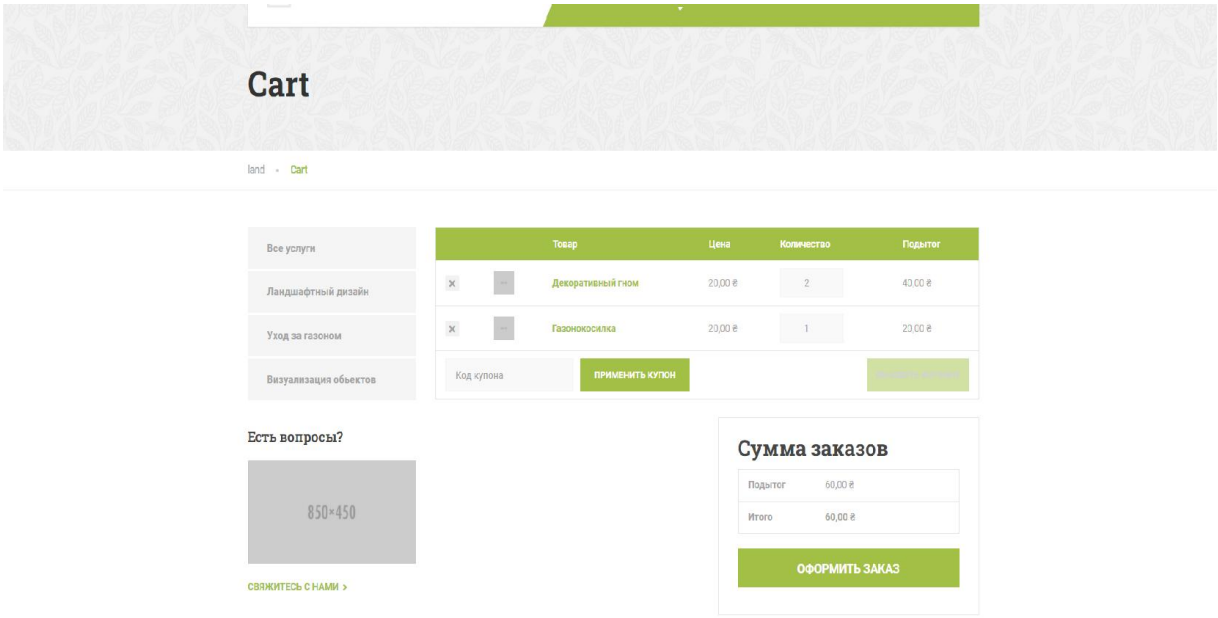

Рисунок 3.28 – Сторінка корзини

Сторінка оформлення замовлення (рис. 3.29).

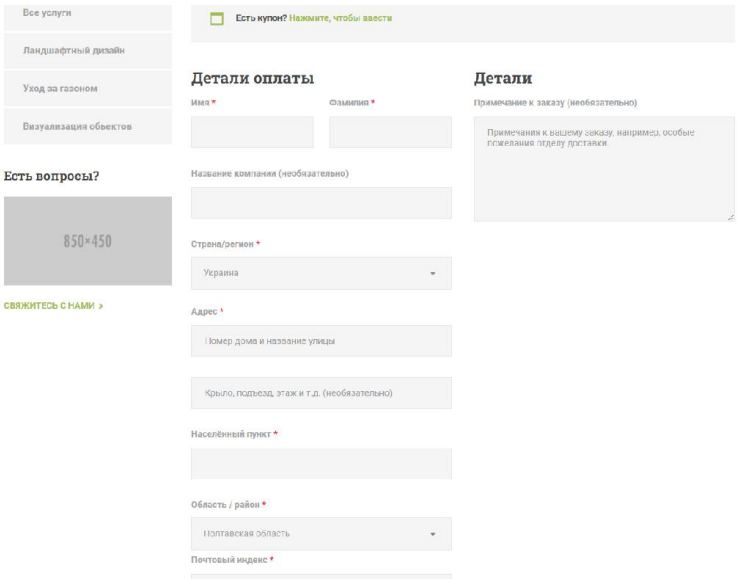

Рисунок 3.29 – Сторінка оформлення замовлення

Сторінка списку товарів в адмін частині (рис. 3.30).

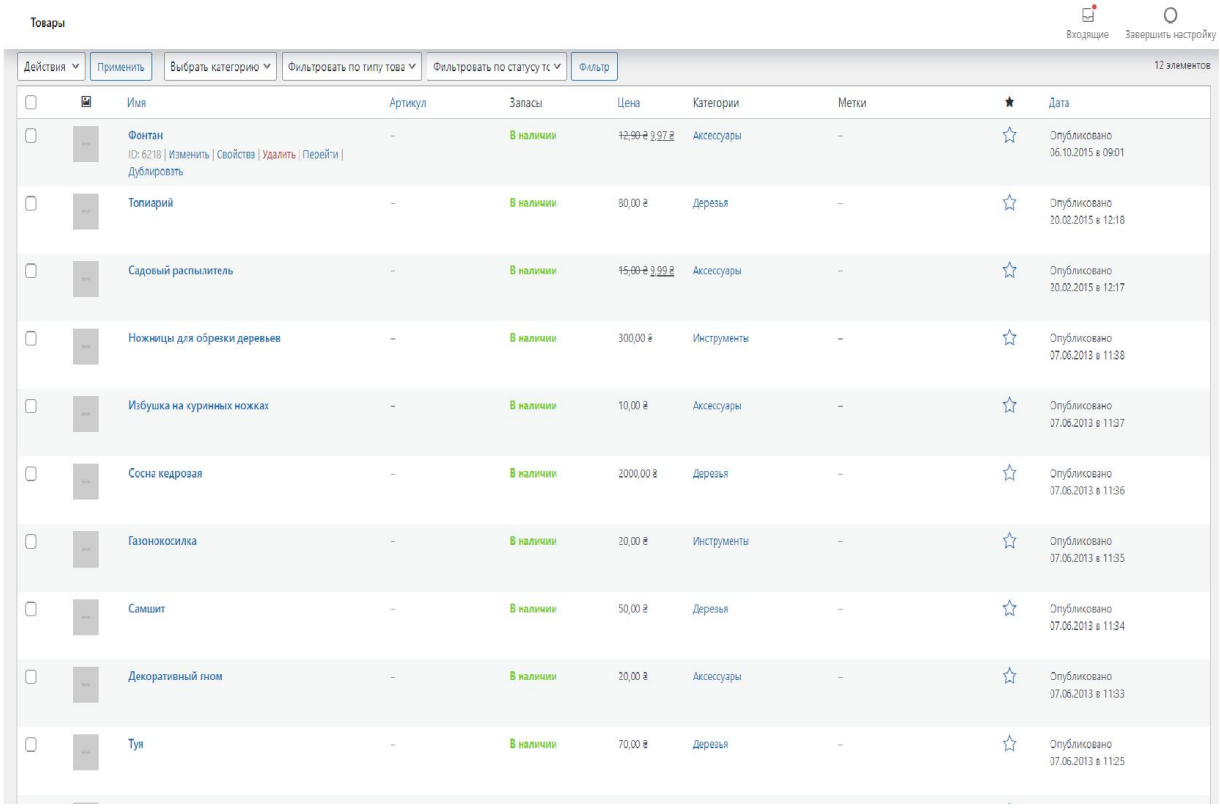

Рисунок 3.30 – Сторінка списку товарів в адмін частині

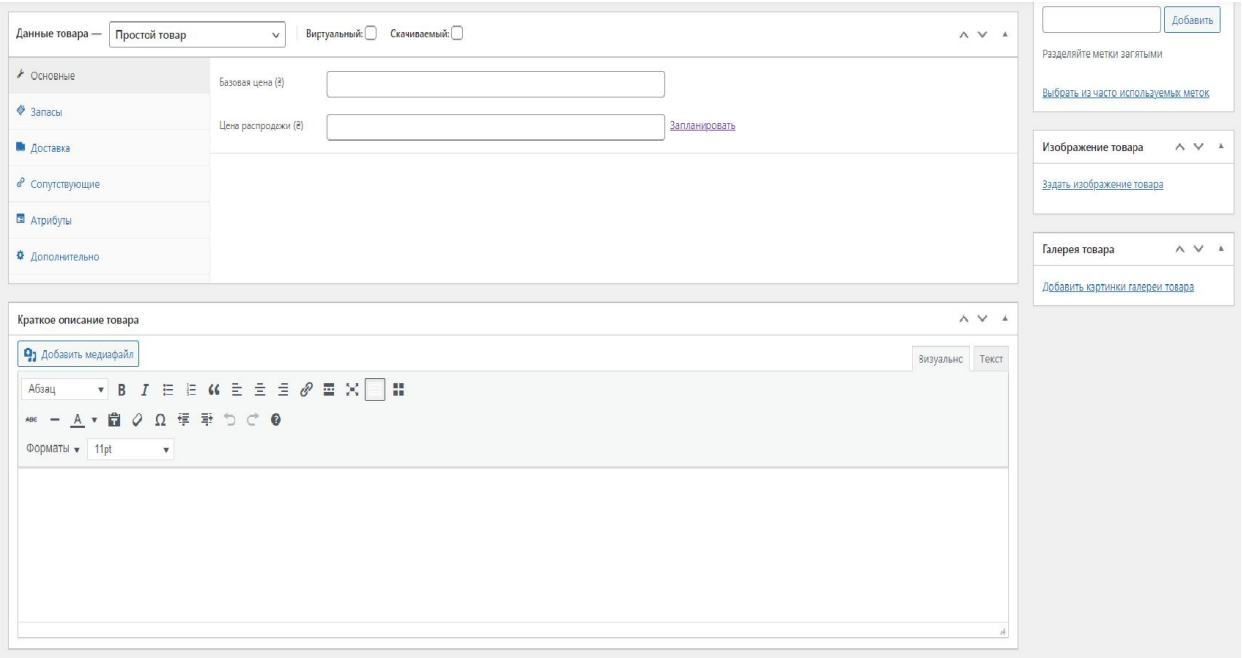

Рисунок 3.30 – Сторінка списку товарів в адмін частині

3.5.6 Панель управління сайтом. Панель управління – це сторінка з якої виконується управління. На сторінці мають бути відображені вкладки для керування користувачами та контентом.

Структура сторінок панелі адміністратора:

- головна;
- користувачі;
- налаштування;
- товари.

На сторінці користувачі відображаються всі зареєстровані користувачі (рис. 3.32). Тут також можна переглянути (рис. 3.33), редагувати та призначити роль (рис. 3.34), заблокувати (рис. 3.35) та видалити профіль користувача (рис. 3.36), відфільтрувати список користувачів по статусу E-Mail, встановити кількість одночасно відображуваних користувачів та виконати пошук по списку користувачів.

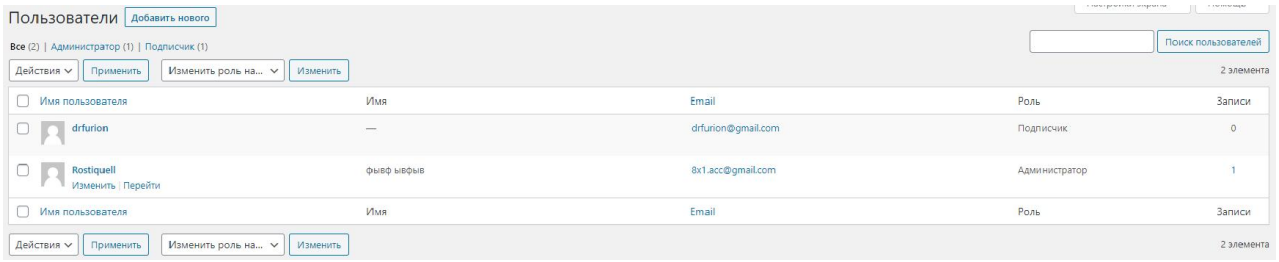

Рисунок 3.32 – Сторінка користувачі

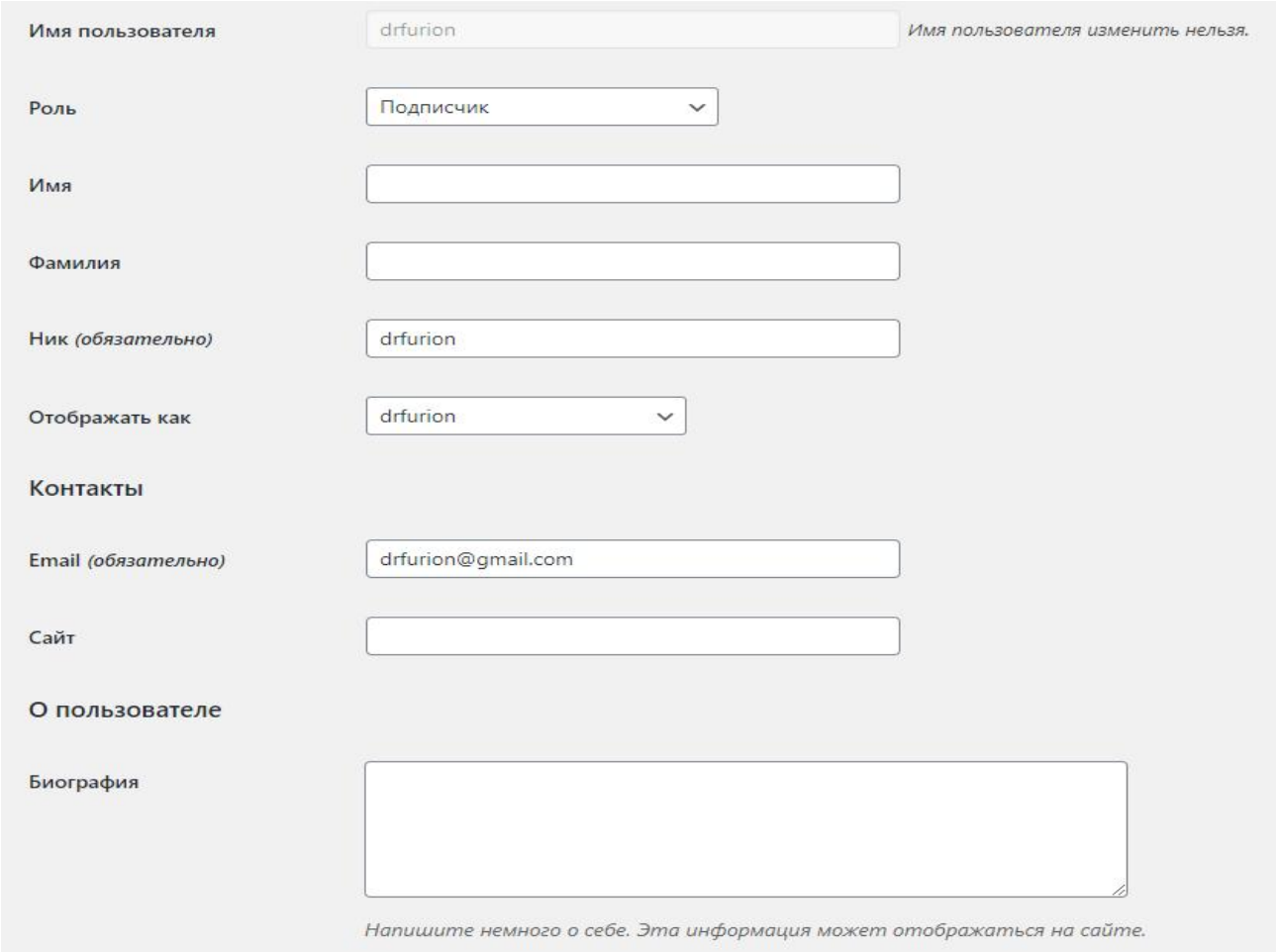

Рисунок 3.34 – Сторінка редагування профілю користувача

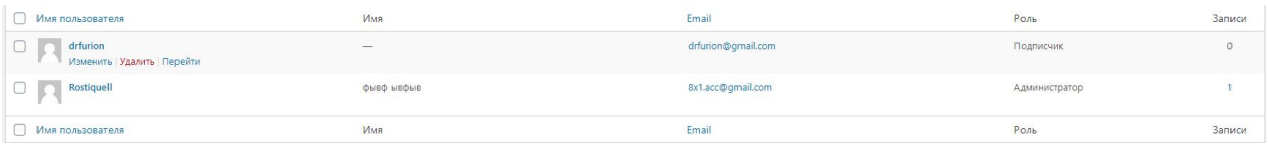

Рисунок 3.36 – Видалення профілю користувача

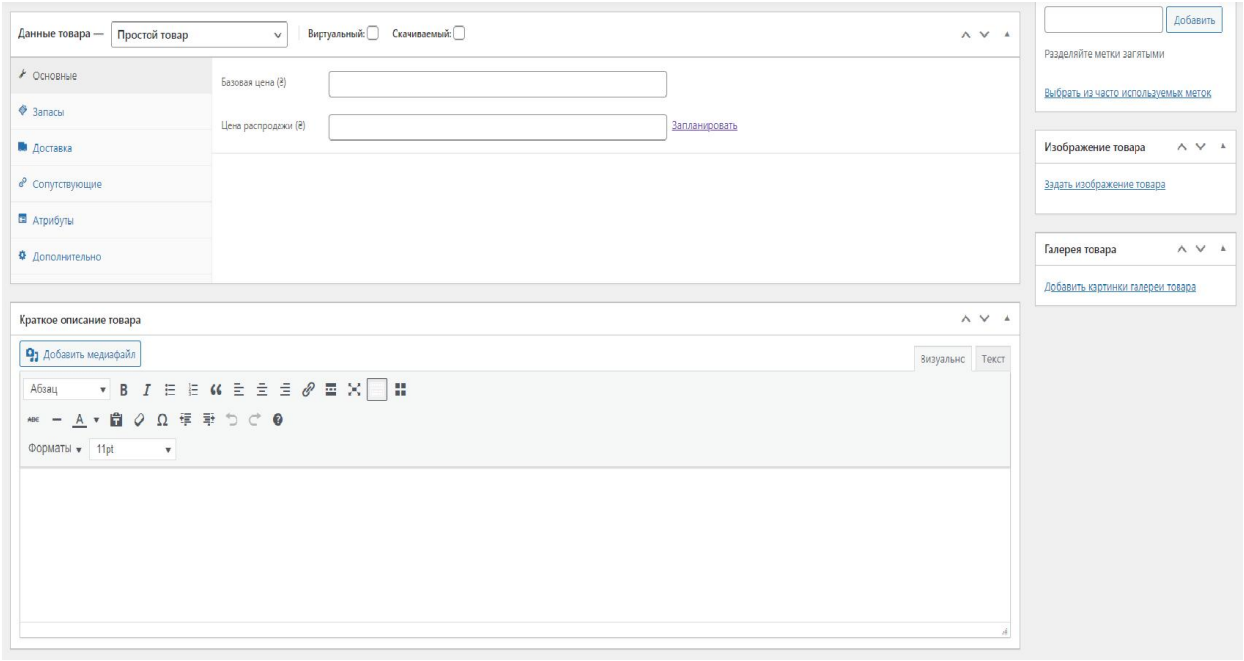

Рисунок 3.36 – редагування товарів

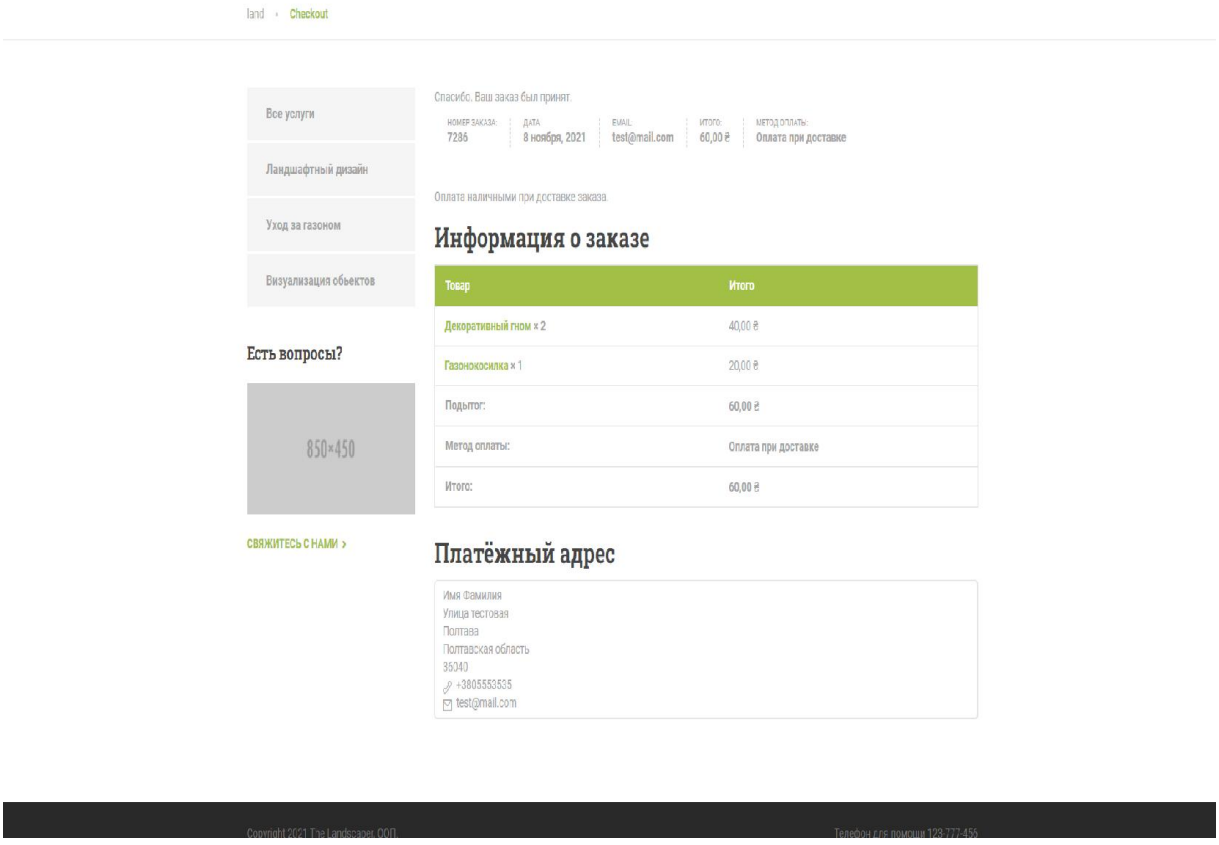

Рисунок 3.36 – оформлений товар

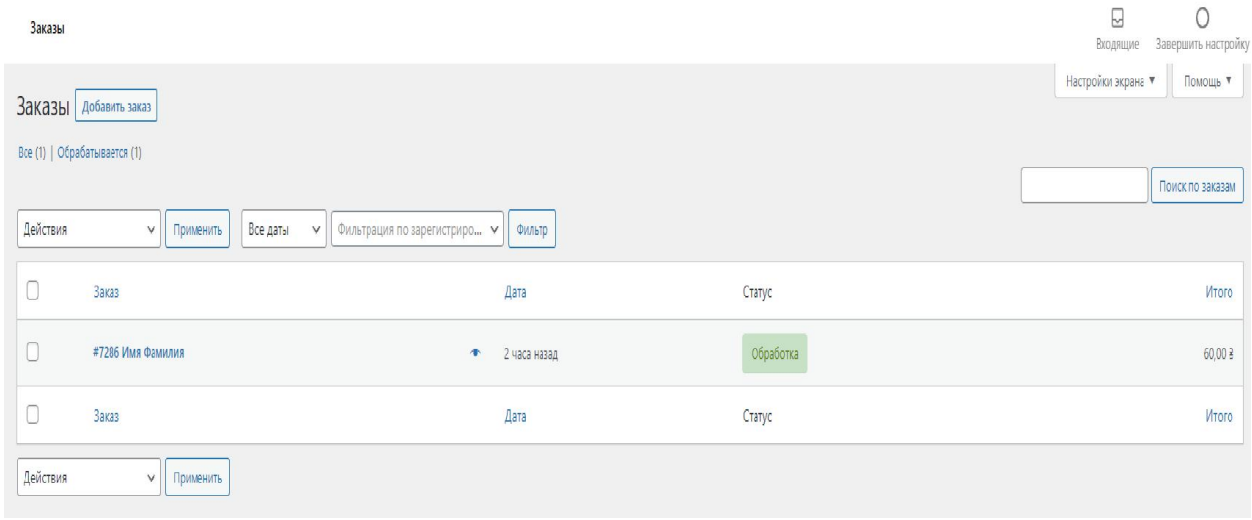

Рисунок 3.36 – оформлений товар на сторінці адміністратора

РОЗДІЛ 4
ТЕСТУВАННЯ

4.1 Тестування ІС

Якість програмного продукту характеризується набором властивостей, що визначають, наскільки продукт «хороший» з точки зору зацікавлених сторін, таких як замовник продукту, спонсор, кінцевий користувач, розробники і тестувальники продукту, інженери підтримки, співробітники відділів маркетингу, навчання і продажів. Кожен з учасників може мати різне уявлення про продукт і те, наскільки він хороший чи поганий, тобто про те, наскільки висока якість продукту. Таким чином, постановка задачі забезпечення якості продукту виливається у завдання визначення зацікавлених осіб, їх критеріїв якості і потім знаходження оптимального рішення, що задовольняє цим критеріям. Тестування є одним з найбільш усталених способів забезпечення якості розробки програмного забезпечення і входить в набір ефективних засобів сучасної системи забезпечення якості програмного продукту.

З технічної точки зору тестування полягає у виконанні програми на деякій множині вихідних даних м звірці одержуваних результатів із заздалегідь відомими (еталонними) з метою встановити відповідність різних властивостей і характеристик програми замовленим властивостями.

4.2 Тест-план

Тест план-це документ описує весь обсяг робіт з тестування, починаючи з опису об'єкта, стратегії, розкладу, критеріїв початку і закінчення тестування, до необхідного в процесі роботи обладнання, спеціальних знань, а також оцінки ризиків з варіантами їх вирішення.

Тест план повинен як мінімум відповідати на такі питання:

1. Що треба тестувати (об'єкт тестування: система, додаток, обладнання).

2. Що будете тестувати (список функцій і компонент тестується системи).
3. Як будете тестувати (стратегія тестування - види тестування і їх застосування по відношенню до тестованого об'єкту).
4. Коли будете тестувати (послідовність проведення робіт: підготовка, тестування, аналіз результатів, в розрізі запланованих фаз розробки проекту).
5. Критерії початку і закінчення тестування.

Тестування проводилося в 3 найпопулярніших браузерах. Результати проведення фінального тестування верстки зображена в таблиці 4.1

Таблиця 4.1 – Тестування верстки

	Mozilla Firefox	Google Chrome	Internet Explorer
Коректне відображення картинок	Успішно	Успішно	Успішно
Клікабельність посилань	Успішно	Успішно	Успішно
Відображення колірної гами всіх елементів	Успішно	Успішно	Успішно
Коректність скролла	Успішно	Успішно	Успішно
Коректне розміщення банерів	Успішно	Успішно	Успішно
Коректне маштабування сторінок	Успішно	Успішно	Успішно
Відображення в різних розширеннях	Успішно	Успішно	Успішно
Коректна зміна розміру текстових полів	Успішно	Успішно	Успішно

Після завершення тестування верстки було проведено тестування функціоналу. Результати тестування наведені в таблиці 4.2.

Таблиця 4.2 – Тестування функціоналу

Клікабельність кнопок	Успішно	Успішно	Успішно
Коректне відображення картинок	Успішно	Успішно	Успішно
Меню	Успішно	Успішно	Успішно
Локалізація	Успішно	Успішно	Успішно
Клікабельність посилань	Успішно	Успішно	Успішно
Коректне заповнення бланка	Успішно	Успішно	Успішно

зворотнього зв'язку (перевірка Email)			

Також було розроблено та виконано тест-кейс, для перевірки роботи системи. Виконані тест кейси наведені в таблиці 4.3.

Таблиця 4.3 – Тест-кейси

Тест-кейс №1. Тестування при невірному введенні логіну	
Дії	**Очікуваний результат**
1. Відкриваємо інформаційну систему від імені адміністратора	З'явилося вікно з формою авторизації
2. Вводимо незареєстрований логін	Логін введений
3. Натискаємо кнопку "Вхід"	З'явилося повідомлення про те, що логін невірний
Тест-кейс №2. Тестування при невірному введенні пароля	
Дії	**Очікуваний результат**
1. Відкриваємо інформаційну систему від імені адміністратора	З'явилося вікно з формою авторизації
2. Вводимо зареєстрований логін	Логін введений
3. Вводимо невірний пароль	Пароль введений
4. Натискаємо кнопку "Вхід"	З'явилося повідомлення про те, що пароль невірний
Тест-кейс №3. Тестування при вірному введенні логіна та пароля	
Дії	**Очікуваний результат**
1. Відкриваємо інформаційну систему від імені адміністратора	З'явилося вікно з формою авторизації
2. Вводимо зареєстрований логін	Логін введений
3. Вводимо вірний пароль	Пароль введений
4. Натискаємо кнопку "Вхід"	Вхід в інформаційну систему від імені адміністратора

Провівши тестування, були знайдені та усунені помилки. Інформаційна система працює коректно. Фунціонал повністю та безпомилково виконує поставлені задачі.

ВИСНОВКИ

В даній дипломній роботі був спроектований та програмно реалізований інформаційна система підприємства для проведення надання послуг з ландшафтного дизайну.

Розроблена система забезпечить:

− можливість приймати участь в роботі та замовленню послуг незалежно від географічного положення;

− пошук необхідних послуг;

− пошук товару.

Переваги такого сайту – незалежність від географічного положення, що збільшить потенціальний потік користувачів.

Проект розроблений за допомогою CMS WordPress, що забезпечує надійність збереження даних користувачів, швидкодію та невисоке навантаження на сервер, можливість розширення функціоналу сайту.

Враховуючи вище сказане можна зробити висновок, що всі поставлені завдання в дипломній роботі виконані у повному обсязі.

Таким чином, впровадження цього проекту значною мірою поліпшує збереження цілісності даних, перегляд та пошук необхідної інформації.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Программирование на yii [Електронний ресурс]. – Режим доступу: http://www.yiiframework.com/;
2. Руководство по PHP [Електронний ресурс]. – Режим доступу: http://php.net/manual/ru/ ;
3. Учебник CSS [Електронний ресурс]. – Режим доступу: http://ru.html.net/tutorials/css/;
4. Карасик І. Програмні та апаратні засоби захисту інформації для персональних комп'ютерів / / КомпьютерПресс № 3, 1995
5. HTML та CSS бібліотека [Електронний ресурс]. – Режим доступу: http://htmlbook.ru/
6. Анісімов А.В. Інформаційні системи та бази даних: Навчальний посібник для студентів факультету комп’ютерних наук та кібернетики. / Анісімов А.В., Кулябко П.П. − Київ. – 2017. – 110 с.
7. Антоненко В. М. Сучасні інформаційні системи і технології: управління знаннями : навч. посібник / В. М. Антоненко, С. Д. Мамченко, Ю. В. Рогушина. – Ірпінь : Нац. університет ДПС України, 2016. – 212 с.
8. Воронін А. М. Інформаційні системи прийняття рішень: навчальний посібник. / Воронін А. М., Зіатдінов Ю. К., Климова А. С. − К. : НАУ-друк, 2009. − 136с.
9. Галузинський Г. П. Інформаційні системи у бізнесі. Практикум для індивідуальної роботи: навч.- метод. посіб. для самост. вивч. Дисципліни. / Галузинський Г. П., Денісова О. О., Писаревська Т. А. − К. : КНЕУ, 2008. − 524с.
10. Годун В.М. Інформаційні системи і технології в статистиці: навч. посіб. / В.М. Годун, Н.С. Орленко, М. А. Сендзюк; за ред. В.Ф. Ситника. – К.: КНЕУ, 2003. – 267 сКомп'ютерні та Інформаційні технології і системи [Електронний ресурс] – режим доступу: http://fitts.pntu.edu.ua/ua/kafedry/inf-tekh-i-sy-m

11. XHTML [Електронний ресурс] – режим доступу: http://htmlbook.ru/ xhtml
12. ANGULAR [Електронний ресурс] – режим доступу: https://metanit.com/ web/angular/
13. APACHE [Електронний ресурс] – режим доступу: http://www.apache.ru/
14. MYSQL [Електронний ресурс] – режим доступу: https://www.mysql.com/
15. Діаграма прецедентів [Електронний ресурс] – режим доступу: http://www.planerka.info/item/Diagramma-precedentov-(variantov-ispolzovaniya)-UML
16. Діаграма діяльності [Електронний ресурс] – режим доступу: http://it-gost.ru/articles/view_articles/96
17. Контекстна діаграма [Електронний ресурс] – режим доступу: http://techtrend.com.ua/index.php
18. Діаграма декомпозиції [Електронний ресурс] – режим доступу: http://studopedia.com.ua/1_162873_diagrami-dekompozitsii.html
19. Розробка інформаційної системи [Електронний ресурс] – режим доступу: http://bukvar.su/informatika_programmirovanie
20. Аналіз баз даних [Електронний ресурс] – режим доступу: https://www.osp.ru/os/2002/03/181272/
21. Порівняльний аналіз СУБД [Електронний ресурс] – режим доступу: https://sibac.info/studconf/tech/xxxvi/43413
22. Порівняльний аналіз серверів [Електронний ресурс] – режим доступу: http://www.it-ic.ru/sravnenie-veb-serverov-apache-iis
23. Огляд web-серверів [Електронний ресурс] – режим доступу: http://info-comp.ru/sisadminst/221-popular-web-servers.html
24. Переваги Laravel [Електронний ресурс] – режим доступу: https://wezom.com.ua/blog/17-preimuschestv-ispolzovanija-laravel-v-it-industrii

ДОДАТКИ

Додаток А. Код програми

\myaccount\form-edit-account.php

```
<?php
/**
 * Edit account form
 *
 * This template can be overridden by copying it to yourtheme/woocommerce/myaccount/form-edit-
account.php.
 *
 * HOWEVER, on occasion WooCommerce will need to update template files and you
 * (the theme developer) will need to copy the new files to your theme to
 * maintain compatibility. We try to do this as little as possible, but it does
 * happen. When this occurs the version of the template file will be bumped and
 * the readme will list any important changes.
 *
 * @see https://docs.woocommerce.com/document/template-structure/
 * @package WooCommerce/Templates
 * @version 3.5.0
 */

defined( 'ABSPATH' ) || exit;

do_action( 'woocommerce_before_edit_account_form' ); ?>

<form class="woocommerce-EditAccountForm edit-account" action="" method="post" <?php do_action(
'woocommerce_edit_account_form_tag' ); ?> >

        <?php do_action( 'woocommerce_edit_account_form_start' ); ?>

        <p class="woocommerce-FormRow woocommerce-FormRow--first form-row form-row-first">
                <input type="text" class="woocommerce-Input woocommerce-Input--text input-text"
placeholder="<?php esc_html_e( 'Имя', 'unero' ); ?>" name="account_first_name" autocomplete="given-
name" id="account_first_name" value="<?php echo esc_attr( $user->first_name ); ?>" />
        </p>
        <p class="woocommerce-FormRow woocommerce-FormRow--last form-row form-row-last">
                <input type="text" class="woocommerce-Input woocommerce-Input--text input-text"
placeholder="<?php esc_html_e( 'Фамилия', 'unero' ); ?>" name="account_last_name"
autocomplete="family-name" id="account_last_name" value="<?php echo esc_attr( $user->last_name );
?>" />
        </p>
        <div class="clear"></div>

        <p class="woocommerce-form-row woocommerce-form-row--wide form-row form-row-wide">
                <label for="account_display_name"><?php esc_html_e( 'Никнейм', 'unero' );
?> <span class="required">*</span></label>
                <input type="text" class="woocommerce-Input woocommerce-Input--text input-text"
name="account_display_name" id="account_display_name" value="<?php echo esc_attr( $user-
>display_name ); ?>" /> <span><em><?php esc_html_e( 'Так будет отображаться ваше имя в разделе
аккаунта и в обзорах.', 'unero' ); ?></em></span>
        </p>
        <div class="clear"></div>

        <p class="woocommerce-FormRow woocommerce-FormRow--wide form-row form-row-wide">
                <input type="email" class="woocommerce-Input woocommerce-Input--email input-text"
placeholder="<?php esc_html_e( 'Email', 'unero' ); ?>" name="account_email" autocomplete="email"
id="account_email" value="<?php echo esc_attr( $user->user_email ); ?>" />
        </p>

        <fieldset>
                <p class="woocommerce-FormRow woocommerce-FormRow--wide form-row form-row-wide">
                        <input type="password" class="woocommerce-Input woocommerce-Input--
password input-text" placeholder="<?php esc_html_e( 'Старый пароль', 'unero' ); ?>"
name="password_current" autocomplete="off" id="password_current" />
                </p>
                <p class="woocommerce-FormRow woocommerce-FormRow--wide form-row form-row-wide">
                        <input type="password" class="woocommerce-Input woocommerce-Input--
password input-text" placeholder="<?php esc_html_e( 'Новый пароль', 'unero' ); ?>" name="password_1"
autocomplete="off"  id="password_1" />
                </p>
                <p class="woocommerce-FormRow woocommerce-FormRow--wide form-row form-row-wide">
                        <input type="password" class="woocommerce-Input woocommerce-Input--
password input-text" placeholder="<?php esc_html_e( 'Подтвердите новый пароль', 'unero' ); ?>"
name="password_2" autocomplete="off" id="password_2" />
```

```
                </p>
        </fieldset>
        <div class="clear"></div>

        <?php do_action( 'woocommerce_edit_account_form' ); ?>

        <p class="edit-btn">
                <?php wp_nonce_field( 'save_account_details', 'save-account-details-nonce' ); ?>
                <input type="submit" class="woocommerce-Button button"
name="save_account_details" value="<?php esc_attr_e( 'Сохранить', 'unero' ); ?>" />
                <input type="hidden" name="action" value="save_account_details" />
        </p>

        <?php do_action( 'woocommerce_edit_account_form_end' ); ?>
</form>

<?php do_action( 'woocommerce_after_edit_account_form' ); ?>
```

myaccount\dashboard.php

```
<?php
/**
 * My Account Dashboard
 *
 * Shows the first intro screen on the account dashboard.
 *
 * This template can be overridden by copying it to yourtheme/woocommerce/myaccount/my-account-
dashboard.php.
 *
 * HOWEVER, on occasion WooCommerce will need to update template files and you
 * (the theme developer) will need to copy the new files to your theme to
 * maintain compatibility. We try to do this as little as possible, but it does
 * happen. When this occurs the version of the template file will be bumped and
 * the readme will list any important changes.
 *
 * @see         https://docs.woothemes.com/document/template-structure/
 * @author      WooThemes
 * @package     WooCommerce/Templates
 * @version     2.6.0
 */

if ( ! defined( 'ABSPATH' ) ) {
        exit; // Exit if accessed directly
}
?>

<div class="row">
        <div class="col-md-4 col-sm-4 col-left">
                <div class="myaccount-sidebar">
                        <?php
                        $user = get_user_by( 'ID', get_current_user_id() );
                        if ( $user ) {
                                ?>
                                <ul>
                                        <li>
                                                <?php echo get_avatar(
get_current_user_id(), 125 ); ?>
                                        </li>
                                        <li>
                                                <?php printf( '<span class="m-title">%s
%s</span>', esc_html__( 'Привет!', 'unero' ), $user->display_name ); ?>
                                        </li>
                                        <li>
                                                <?php printf( '<span>%s:</span>%s',
esc_html__( 'Имя', 'unero' ), get_user_meta( get_current_user_id(), 'billing_first_name', true ) . '
' . get_user_meta( get_current_user_id(), 'billing_last_name', true ) ); ?>
                                        </li>
                                        <li>
                                                <?php printf( '<span>%s:</span>%s',
esc_html__( 'Email', 'unero' ), $user->user_email ); ?>
                                        </li>
                                        <li>
                                                <?php printf( '<span>%s:</span>%s',
esc_html__( 'Телефон', 'unero' ), get_user_meta( get_current_user_id(), 'billing_phone', true ) );
?>
                                        </li>
                                        <li>
                                                <?php
                                                $country = get_user_meta(
get_current_user_id(), 'billing_country', true );
```

```
                                                        if( $country && function_exists( 'WC' ) ) {
                                                                $country = WC()->countries->countries[ $country ];
                                                        }
                                                        ?>
                                                        <?php printf( '<span>%s:</span>%s', esc_html__( 'Страна', 'unero' ), $country )  ?>
                                                </li>
                                                <li>
                                                        <?php printf( '<span>%s:</span>%s', esc_html__( 'Индекс', 'unero' ), get_user_meta( get_current_user_id(), 'billing_postcode', true ) ); ?>
                                                </li>
                                                <li>
                                                        <?php printf( '<a href="%s" class="m-button">%s</a>', esc_url( wc_get_endpoint_url( 'edit-account' ) ), esc_html__( 'Редактировать профиль', 'unero' ) ); ?>
                                                </li>
                                        </ul>
                                <?php } ?>
                        </div>
                </div>
                <div class="col-md-8 col-sm-8">
                        <?php
                        /**
                         * My Account dashboard.
                         *
                         * @since 2.6.0
                         */
                        do_action( 'woocommerce_account_dashboard' );

                        /**
                         * Deprecated woocommerce_before_my_account action.
                         *
                         * @deprecated 2.6.0
                         */
                        do_action( 'woocommerce_before_my_account' );

                        /**
                         * Deprecated woocommerce_after_my_account action.
                         *
                         * @deprecated 2.6.0
                         */
                        do_action( 'woocommerce_after_my_account' );
                        ?>
                </div>
</div>
```

myaccount\my-address.php

```
<?php
/**
 * My Addresses
 *
 * This template can be overridden by copying it to yourtheme/woocommerce/myaccount/my-address.php.
 *
 * HOWEVER, on occasion WooCommerce will need to update template files and you
 * (the theme developer) will need to copy the new files to your theme to
 * maintain compatibility. We try to do this as little as possible, but it does
 * happen. When this occurs the version of the template file will be bumped and
 * the readme will list any important changes.
 *
 * @see     https://docs.woothemes.com/document/template-structure/
 * @author  WooThemes
 * @package WooCommerce/Templates
 * @version 2.6.0
 */

if ( ! defined( 'ABSPATH' ) ) {
        exit; // Exit if accessed directly
}

$customer_id = get_current_user_id();

if ( ! wc_ship_to_billing_address_only() && wc_shipping_enabled() ) {
        $get_addresses = apply_filters( 'woocommerce_my_account_get_addresses', array(
                'billing'  => esc_html__( 'Оплата', 'unero' ),
                'shipping' => esc_html__( 'Доставка', 'unero' )
```

```
        ), $customer_id );
} else {
        $get_addresses = apply_filters( 'woocommerce_my_account_get_addresses', array(
                'billing' => esc_html__( 'Оплата', 'unero' )
        ), $customer_id );
}

$oldcol = 1;
$col    = 1;
?>

<h2 class="billing-title"><?php esc_html_e( 'Данные оплаты и доставки', 'unero' ); ?></h2>

<p>
        <?php echo apply_filters( 'woocommerce_my_account_my_address_description', esc_html__(
'Следующие адреса будут использоваться на странице оформления заказа по умолчанию.', 'unero' ) ); ?>
</p>

<?php if ( ! wc_ship_to_billing_address_only() && wc_shipping_enabled() ) {
        echo '<div class="u-columns woocommerce-Addresses col2-set addresses">';
} ?>

<?php foreach ( $get_addresses as $name => $title ) : ?>

    <div class="u-column<?php echo ( ( $col = $col * - 1 ) < 0 ) ? 1 : 2; ?> col-<?php echo ( (
$oldcol = $oldcol * - 1 ) < 0 ) ? 1 : 2; ?> woocommerce-Address">
        <header class="woocommerce-Address-title title">
            <h3><?php echo wp_kses_post( $title ); ?></h3>
        </header>
        <address>
                            <?php
                            $address = apply_filters(
'woocommerce_my_account_my_address_formatted_address', array(
                                    'first_name' => get_user_meta( $customer_id, $name .
'_first_name', true ),
                                    'last_name'  => get_user_meta( $customer_id, $name .
'_last_name', true ),
                                    'company'    => get_user_meta( $customer_id, $name .
'_company', true ),
                                    'address_1'  => get_user_meta( $customer_id, $name .
'_address_1', true ),
                                    'address_2'  => get_user_meta( $customer_id, $name .
'_address_2', true ),
                                    'city'       => get_user_meta( $customer_id, $name . '_city',
true ),
                                    'state'      => get_user_meta( $customer_id, $name . '_state',
true ),
                                    'postcode'   => get_user_meta( $customer_id, $name .
'_postcode', true ),
                                    'country'    => get_user_meta( $customer_id, $name .
'_country', true )
                            ), $customer_id, $name );

                            $formatted_address = WC()->countries->get_formatted_address( $address
);

                            if ( ! $formatted_address ) {
                                    _e( 'ВЫ ЕЩЕ НЕ НАСТРОИЛИ ЭТО.', 'unero' );
                            } else {
                                    echo wp_kses_post( $formatted_address );
                            }
                            ?>
        </address>
        <footer class="woocommerce-Address-edit">
                            <?php
                            $edit_title = esc_html__( 'Редактировать способ оплаты', 'unero' );
                            if ( $name != 'billing' ) {
                                    $edit_title = esc_html__( 'Редактировать адресс доставки',
'unero' );
                            }
                            ?>
            <a href="<?php echo esc_url( wc_get_endpoint_url( 'edit-address', $name ) ); ?>"
               class="edit"><?php echo esc_attr( $edit_title ); ?></a>
        </footer>
    </div>

<?php endforeach; ?>

<?php if ( ! wc_ship_to_billing_address_only() && wc_shipping_enabled() ) {
        echo '</div>';
```

```
} ?>
```

myaccount\form-login.php

```
<?php
/**
 * Login Form
 *
 * This template can be overridden by copying it to yourtheme/woocommerce/myaccount/form-login.php.
 *
 * HOWEVER, on occasion WooCommerce will need to update template files and you
 * (the theme developer) will need to copy the new files to your theme to
 * maintain compatibility. We try to do this as little as possible, but it does
 * happen. When this occurs the version of the template file will be bumped and
 * the readme will list any important changes.
 *
 * @see     https://docs.woocommerce.com/document/template-structure/
 * @package WooCommerce/Templates
 * @version 3.6.0
 */

if ( ! defined( 'ABSPATH' ) ) {
        exit; // Exit if accessed directly.
}

?>

<?php do_action( 'woocommerce_before_customer_login_form' ); ?>

<div class="customer-login">

        <div class="row">

                <div class="col-md-6 col-sm-6 col-md-offset-3 col-sm-offset-3 col-login">
                        <div class="unero-tabs">
                                <ul class="tabs-nav">
                                        <li class="active"><a href="#" class="active"><?php
esc_html_e( 'Войти', 'unero' ); ?></a></li>
                                        <?php if ( get_option(
'woocommerce_enable_myaccount_registration' ) === 'yes' ) : ?>
                                                <li><a href="#"><?php esc_html_e(
'Регистрация', 'unero' ); ?></a></li>
                                        <?php endif; ?>
                                </ul>
                                <div class="tabs-content">

                                        <div class="tabs-panel active">

                <form class="woocommerce-form woocommerce-form-login login" method="post">

                                                <?php do_action(
'woocommerce_login_form_start' ); ?>

                                                <p class="woocommerce-FormRow
woocommerce-FormRow--wide form-row form-row-wide">
                                                        <input type="text"
placeholder="<?php esc_html_e( 'Имя', 'unero' ); ?>" class="woocommerce-Input woocommerce-Input--
text input-text" name="username" id="username" autocomplete="username" value="<?php if ( ! empty(
$_POST['username'] ) ) {
                                                                echo esc_attr(
$_POST['username'] );
                                                        } ?>" />
                                                </p>

                                                <p class="woocommerce-FormRow
woocommerce-FormRow--wide form-row form-row-wide form-row-password">
                                                        <input
placeholder="<?php esc_html_e( 'Пароль', 'unero' ); ?>" class="woocommerce-Input woocommerce-Input--
text input-text" type="password" name="password" id="password" autocomplete="current-password" />
                                                        <a class="lost-password"
href="<?php echo esc_url( wp_lostpassword_url() ); ?>"><?php esc_html_e( 'Забыли?', 'unero' );
?></a>
                                                </p>

                                                <?php do_action(
'woocommerce_login_form' ); ?>

                                                <p class="form-row">
                                                        <label for="rememberme"
class="inline rememberme">
```

```
<input
class="woocommerce-Input woocommerce-Input--checkbox" name="rememberme" type="checkbox"
id="rememberme" value="forever" /><span class="label"> <?php esc_html_e( 'Запомнить меня', 'unero'
); ?></span>
</label>
<?php wp_nonce_field(
'woocommerce-login', 'woocommerce-login-nonce' ); ?>
<input type="submit"
class="woocommerce-Button button" name="login" value="<?php esc_attr_e( 'Войти', 'unero' ); ?>" />

</p>

<?php do_action(
'woocommerce_login_form_end' ); ?>

</form>
</div>

<?php if ( get_option(
'woocommerce_enable_myaccount_registration' ) === 'yes' ) : ?>

<div class="tabs-panel">


<form method="post"
class="woocommerce-form woocommerce-form-register register">

<?php do_action(
'woocommerce_register_form_start' ); ?>

<?php if ( 'no' ===
get_option( 'woocommerce_registration_generate_username' ) ) : ?>

<p
class="woocommerce-FormRow woocommerce-FormRow--wide form-row form-row-wide">

<input type="text" placeholder="<?php esc_html_e( 'Username', 'unero' ); ?>"
autocomplete="username" class="woocommerce-Input woocommerce-Input--text input-text" name="username"
id="reg_username" value="<?php if ( ! empty( $_POST['username'] ) ) {
ec
}
?>" />
</p>

<?php endif; ?>

<p class="woocommerce-
FormRow woocommerce-FormRow--wide form-row form-row-wide">
<input
type="email" placeholder="<?php esc_html_e( 'Email', 'unero' ); ?>"  autocomplete="email"
class="woocommerce-Input woocommerce-Input--text input-text" name="email" id="reg_email"
value="<?php if ( ! empty( $_POST['email'] ) ) {
echo
esc_attr( $_POST['email'] );
} ?>" />
</p>

<?php if ( 'no' ===
get_option( 'woocommerce_registration_generate_password' ) ) : ?>

<p
class="woocommerce-FormRow woocommerce-FormRow--wide form-row form-row-wide">

<input type="password" placeholder="<?php esc_html_e( 'Password', 'unero' ); ?>"
autocomplete="new-password" class="woocommerce-Input woocommerce-Input--text input-text"
name="password" id="reg_password" />
</p>

<?php else : ?>

<p><?php esc_html_e( 'Пароль будет отправлен на вашу почту.',
'unero' ); ?></p>

<?php endif; ?>

<?php do_action(
'woocommerce_register_form' ); ?>
<?php do_action(
'register_form' ); ?>
```

```
                                                                    <p class="woocomerce-
FormRow form-row">
                                                                        <?php
wp_nonce_field( 'woocommerce-register', 'woocommerce-register-nonce' ); ?>
                                                                        <input
type="submit" class="woocommerce-Button button" name="register" value="<?php esc_attr_e(
'Зарегистрироваться', 'unero' ); ?>" />
                                                                    </p>

                                                                    <?php do_action(
'woocommerce_register_form_end' ); ?>

                                                                </form>

                                                            </div>

                                                        <?php endif; ?>
                                                    </div>
                                                </div>
                                            </div>
                                        </div>
</div>
<?php do_action( 'woocommerce_after_customer_login_form' ); ?>
```

woocommerce\woocommerce.php

```
<?php
/**
 * Plugin Name: WooCommerce
 * Plugin URI: https://woocommerce.com/
 * Description: An eCommerce toolkit that helps you sell anything. Beautifully.
 * Version: 4.2.0
 * Author: Automattic
 * Author URI: https://woocommerce.com
 * Text Domain: woocommerce
 * Domain Path: /i18n/languages/
 *
 * @package WooCommerce
 */

defined( 'ABSPATH' ) || exit;

if ( ! defined( 'WC_PLUGIN_FILE' ) ) {
        define( 'WC_PLUGIN_FILE', __FILE__ );
}

// Load core packages and the autoloader.
require __DIR__ . '/src/Autoloader.php';
require __DIR__ . '/src/Packages.php';

if ( ! \Automattic\WooCommerce\Autoloader::init() ) {
        return;
}
\Automattic\WooCommerce\Packages::init();

// Include the main WooCommerce class.
if ( ! class_exists( 'WooCommerce', false ) ) {
        include_once dirname( WC_PLUGIN_FILE ) . '/includes/class-woocommerce.php';
}

/**
 * Returns the main instance of WC.
 *
 * @since  2.1
 * @return WooCommerce
 */
function WC() { // phpcs:ignore WordPress.NamingConventions.ValidFunctionName.FunctionNameInvalid
        return WooCommerce::instance();
}

// Global for backwards compatibility.
$GLOBALS['woocommerce'] = WC();
```

woocommerce\cart\mini-cart.php

```
<?php
/**
 * Mini-cart
 *
 * Contains the markup for the mini-cart, used by the cart widget.
```

```
 *
 * This template can be overridden by copying it to yourtheme/woocommerce/cart/mini-cart.php.
 *
 * HOWEVER, on occasion WooCommerce will need to update template files and you
 * (the theme developer) will need to copy the new files to your theme to
 * maintain compatibility. We try to do this as little as possible, but it does
 * happen. When this occurs the version of the template file will be bumped and
 * the readme will list any important changes.
 *
 * @see     https://docs.woocommerce.com/document/template-structure/
 * @author  WooThemes
 * @package WooCommerce/Templates
 * @version 3.7.0
 */

if ( ! defined( 'ABSPATH' ) ) {
	exit; // Exit if accessed directly
}

?>

<?php do_action( 'woocommerce_before_mini_cart' ); ?>

<ul class="cart_list product_list_widget <?php echo esc_attr( $args['list_class'] ); ?>">

	<?php if ( ! WC()->cart->is_empty() ) : ?>

		<?php
		foreach ( WC()->cart->get_cart() as $cart_item_key => $cart_item ) {
			$_product   = apply_filters( 'woocommerce_cart_item_product', $cart_item['data'], $cart_item, $cart_item_key );
			$product_id = apply_filters( 'woocommerce_cart_item_product_id', $cart_item['product_id'], $cart_item, $cart_item_key );

			if ( $_product && $_product->exists() && $cart_item['quantity'] > 0 && apply_filters( 'woocommerce_widget_cart_item_visible', true, $cart_item, $cart_item_key ) ) {
				$product_name      = apply_filters( 'woocommerce_cart_item_name', $_product->get_name(), $cart_item, $cart_item_key );
				$thumbnail         = apply_filters( 'woocommerce_cart_item_thumbnail', $_product->get_image(), $cart_item, $cart_item_key );
				$product_price     = apply_filters( 'woocommerce_cart_item_price', WC()->cart->get_product_price( $_product ), $cart_item, $cart_item_key );
				$product_permalink = apply_filters( 'woocommerce_cart_item_permalink', $_product->is_visible() ? $_product->get_permalink( $cart_item ) : '', $cart_item, $cart_item_key );
				?>
			<li class="<?php echo esc_attr( apply_filters( 'woocommerce_mini_cart_item_class', 'mini_cart_item', $cart_item, $cart_item_key ) ); ?>">
					<?php
					$remove_url = '';
					if ( function_exists( 'wc_get_cart_remove_url' ) ) {
						$remove_url = wc_get_cart_remove_url( $cart_item_key );
					} else {
						$remove_url = WC()->cart->get_remove_url( $cart_item_key );
					}
					echo apply_filters(
						'woocommerce_cart_item_remove_link',
sprintf(
						'<a href="%s" class="remove" title="%s" data-item_key="%s" data-product_id="%s" data-product_sku="%s">&times;</a>',
						esc_url( $remove_url ),
						__( 'Remove this item', 'unero' ),
						esc_attr( $cart_item_key ),
						esc_attr( $product_id ),
						esc_attr( $_product->get_sku() )
					), $cart_item_key
					);
					?>
				<div class="un-mini-cart-thumbnail">
					<?php if ( ! $_product->is_visible() ) : ?>
						<?php echo str_replace( array( 'http:', 'https:' ), '', $thumbnail ) ?>
					<?php else : ?>
					<a href="<?php echo esc_url( $product_permalink ); ?>">
						<?php echo str_replace( array( 'http:', 'https:' ), '', $thumbnail ) ?>
					</a>
```

```
                                                    <?php endif; ?>
                    </div>
                    <div class="un-mini-cart-content">
                                                    <?php if ( ! $_product->is_visible() ) : ?>
                                                            <?php echo wp_kses_post(
$product_name ); ?>
                                                    <?php else : ?>
                                <a href="<?php echo esc_url( $product_permalink ); ?>">
                                                                    <?php echo wp_kses_post(
$product_name ); ?>
                                </a>
                                                    <?php endif; ?>
                                                    <?php
                                                    if ( function_exists(
'wc_get_formatted_cart_item_data' ) ) {
                                                            echo
wc_get_formatted_cart_item_data( $cart_item );
                                                    } else {
                                                            echo WC()->cart->get_item_data(
$cart_item );
                                                    }
                                                    ?>

                                                    <?php echo apply_filters(
'woocommerce_widget_cart_item_quantity', '<span class="quantity">' . sprintf( '%s: %s %s',
esc_html__( 'Кол.', 'unero' ), $cart_item['quantity'], $product_price ) . '</span>', $cart_item,
$cart_item_key ); ?>
                    </div>

                </li>
                                        <?php
                                    }
                                }
                                ?>

            <?php else : ?>

            <li class="empty"><?php esc_html_e( 'No products in the cart.', 'unero' ); ?></li>

            <?php endif; ?>

</ul><!-- end product list -->

<?php if ( ! WC()->cart->is_empty() ) : ?>
    <div class="un-cart-panel-footer">
        <p class="total">
            <strong><?php esc_html_e( 'Общая сумма', 'unero' ); ?>:</strong> <?php echo WC()->cart-
>get_cart_subtotal(); ?>
        </p>

                    <?php do_action( 'woocommerce_widget_shopping_cart_before_buttons' ); ?>

        <p class="buttons">
            <a href="<?php echo esc_url( wc_get_cart_url() ); ?>"
               class="button wc-forward"><?php esc_html_e( 'Просмотреть корзину', 'unero' ); ?></a>
            <a href="<?php echo esc_url( wc_get_checkout_url() ); ?>"
               class="button checkout wc-forward"><?php esc_html_e( 'К оплате', 'unero' ); ?></a>
        </p>
    </div>

<?php endif; ?>

<?php do_action( 'woocommerce_after_mini_cart' ); ?>
```

woocommerce\myaccount\dashboard.php

```
<?php
/**
 * My Account Dashboard
 *
 * Shows the first intro screen on the account dashboard.
 *
 * This template can be overridden by copying it to yourtheme/woocommerce/myaccount/my-account-
dashboard.php.
 *
 * HOWEVER, on occasion WooCommerce will need to update template files and you
 * (the theme developer) will need to copy the new files to your theme to
 * maintain compatibility. We try to do this as little as possible, but it does
 * happen. When this occurs the version of the template file will be bumped and
 * the readme will list any important changes.
```

```
 *
 * @see         https://docs.woothemes.com/document/template-structure/
 * @author      WooThemes
 * @package     WooCommerce/Templates
 * @version     2.6.0
 */

if ( ! defined( 'ABSPATH' ) ) {
        exit; // Exit if accessed directly
}
?>

<div class="row">
        <div class="col-md-4 col-sm-4 col-left">
                <div class="myaccount-sidebar">
                        <?php
                        $user = get_user_by( 'ID', get_current_user_id() );
                        if ( $user ) {
                                ?>
                                <ul>
                                        <li>
                                                <?php echo get_avatar(
get_current_user_id(), 125 ); ?>
                                        </li>
                                        <li>
                                                <?php printf( '<span class="m-title">%s
%s</span>', esc_html__( 'Привет!', 'unero' ), $user->display_name ); ?>
                                        </li>
                                        <li>
                                                <?php printf( '<span>%s:</span>%s',
esc_html__( 'Имя', 'unero' ), get_user_meta( get_current_user_id(), 'billing_first_name', true ) . '
' . get_user_meta( get_current_user_id(), 'billing_last_name', true ) ); ?>
                                        </li>
                                        <li>
                                                <?php printf( '<span>%s:</span>%s',
esc_html__( 'Email', 'unero' ), $user->user_email ); ?>
                                        </li>
                                        <li>
                                                <?php printf( '<span>%s:</span>%s',
esc_html__( 'Телефон', 'unero' ), get_user_meta( get_current_user_id(), 'billing_phone', true ) );
?>
                                        </li>
                                        <li>
                                                <?php
                                                $country = get_user_meta(
get_current_user_id(), 'billing_country', true );
                                                if( $country && function_exists( 'WC' ) ) {
                                                        $country = WC()->countries-
>countries[ $country ];
                                                }
                                                ?>
                                                <?php printf( '<span>%s:</span>%s',
esc_html__( 'Страна', 'unero' ), $country )  ?>
                                        </li>
                                        <li>
                                                <?php printf( '<span>%s:</span>%s',
esc_html__( 'Индекс', 'unero' ), get_user_meta( get_current_user_id(), 'billing_postcode', true ) );
?>
                                        </li>
                                        <li>
                                                <?php printf( '<a href="%s" class="m-
button">%s</a>', esc_url( wc_get_endpoint_url( 'edit-account' ) ), esc_html__( 'Редактировать
профиль', 'unero' ) ); ?>
                                        </li>
                                </ul>
                        <?php } ?>
                </div>
        </div>
        <div class="col-md-8 col-sm-8">
                <?php
                /**
                 * My Account dashboard.
                 *
                 * @since 2.6.0
                 */
                do_action( 'woocommerce_account_dashboard' );

                /**
                 * Deprecated woocommerce_before_my_account action.
                 *
```

```
                     * @deprecated 2.6.0
                     */
                    do_action( 'woocommerce_before_my_account' );

                    /**
                     * Deprecated woocommerce_after_my_account action.
                     *
                     * @deprecated 2.6.0
                     */
                    do_action( 'woocommerce_after_my_account' );
                    ?>
            </div>
</div>
```

woocommerce\single-product\meta.php

```
<?php
/**
 * Single Product Meta
 *
 * This template can be overridden by copying it to yourtheme/woocommerce/single-product/meta.php.
 *
 * HOWEVER, on occasion WooCommerce will need to update template files and you
 * (the theme developer) will need to copy the new files to your theme to
 * maintain compatibility. We try to do this as little as possible, but it does
 * happen. When this occurs the version of the template file will be bumped and
 * the readme will list any important changes.
 *
 * @see             https://docs.woocommerce.com/document/template-structure/
 * @author          WooThemes
 * @package         WooCommerce/Templates
 * @version         3.0.0
 */

if ( ! defined( 'ABSPATH' ) ) {
            exit;
}

global $product;

$product_meta = unero_get_option( 'show_product_meta' );
if ( empty( $product_meta ) ) {
            return;
}
?>
<div class="product_meta">

            <?php do_action( 'woocommerce_product_meta_start' ); ?>
            <?php if ( in_array( 'sku', $product_meta ) ) { ?>
                    <?php if ( wc_product_sku_enabled() && ( $product->get_sku() || $product-
>is_type( 'variable' ) ) ) : ?>

              <span class="sku_wrapper"><strong><?php esc_html_e( 'Артикул:', 'unero' ); ?></strong>
                                        <span class="sku">
                      <?php if ( $sku = $product->get_sku() ) {
                                echo wp_kses_post( $sku );
                      } else {
                                esc_html_e( 'N/A', 'unero' );
                      } ?>
                  </span>
              </span>

                    <?php endif; ?>
            <?php } ?>

            <?php if ( in_array( 'categories', $product_meta ) ) { ?>
                    <?php echo wc_get_product_category_list( $product->get_id(), ', ', '<span
class="posted_in"><strong>' . _n( 'Город:', 'Categories:', count( $product->get_category_ids() ),
'unero' ) . ' </strong>', '</span>' ); ?>
            <?php } ?>

            <?php if ( in_array( 'tags', $product_meta ) ) { ?>
                    <?php echo wc_get_product_tag_list( $product->get_id(), ', ', '<span
class="tagged_as"><strong>' . _n( 'Tag:', 'Tags:', count( $product->get_tag_ids() ), 'unero' ) . '
</strong>', '</span>' ); ?>
            <?php } ?>

            <?php do_action( 'woocommerce_product_meta_end' ); ?>
```

```
</div>
```

unero\functions.php

```
<?php
/**
 * Drfuri Core functions and definitions
 *
 * @package Unero
 */

/**
 * Sets up theme defaults and registers support for various WordPress features.
 *
 * @since  1.0
 *
 * @return void
 */
function unero_setup() {
	// Sets the content width in pixels, based on the theme's design and stylesheet.
	$GLOBALS['content_width'] = apply_filters( 'unero_content_width', 840 );

	// Make theme available for translation.
	load_theme_textdomain( 'unero', get_template_directory() . '/lang' );

	// Theme supports
	add_theme_support( 'woocommerce' );
	add_theme_support( 'automatic-feed-links' );
	add_theme_support( 'title-tag' );
	add_theme_support( 'post-thumbnails' );
	add_theme_support( 'post-formats', array( 'audio', 'gallery', 'video' ) );
	add_theme_support(
		'html5', array(
			'comment-list',
			'search-form',
			'comment-form',
			'gallery',
		)
	);


	if( unero_fonts_url() ) {
		add_editor_style( array( 'css/editor-style.css', unero_fonts_url(),
get_template_directory_uri() . '/css/linearicons.min.css' ) );
	} else {
		add_editor_style( 'css/editor-style.css' );
	}

	// Load regular editor styles into the new block-based editor.
	add_theme_support( 'editor-styles' );

	// Load default block styles.
	add_theme_support( 'wp-block-styles' );

	// Add support for responsive embeds.
	add_theme_support( 'responsive-embeds' );

	add_theme_support( 'align-wide' );

	add_theme_support( 'align-full' );

	// Register theme nav menu
	register_nav_menus(
		array(
			'primary'     => esc_html__( 'Primary Menu', 'unero' ),
			'user_logged' => esc_html__( 'User Logged Menu', 'unero' ),
		)
	);

	$image_sizes = unero_get_option( 'image_sizes_default' );
	if ( is_null( $image_sizes ) || ! is_array( $image_sizes ) ) {
		$image_sizes = array(
			'blog_large',
			'blog_normal',
			'blog_masonry',
			'blog_grid',
			'post_normal',
			'categories_grid',
			'shop_masonry',
			'portfolio_masonry',
```

```
                        'portfolio_carousel',
                );
        }

        // Image Size
        if ( in_array( 'blog_large', $image_sizes ) ) {
                add_image_size( 'unero-blog-large', 1170, 360, true );
        }

        if ( in_array( 'blog_normal', $image_sizes ) ) {
                add_image_size( 'unero-blog-normal', 800, 397, true );
        }

        if ( in_array( 'blog_masonry', $image_sizes ) ) {
                add_image_size( 'unero-blog-masonry', 370, 588, false );
        }

        if ( in_array( 'post_normal', $image_sizes ) || in_array( 'blog_grid', $image_sizes ) ||
in_array( 'shop_masonry', $image_sizes ) ) {
                add_image_size( 'unero-product-masonry-normal', 370, 370, true );
        }

        if ( in_array( 'categories_grid', $image_sizes ) ) {
                add_image_size( 'unero-product-cat-normal', 360, 422, true );
                add_image_size( 'unero-product-cat-long', 800, 422, true );
        }

        if ( in_array( 'shop_masonry', $image_sizes ) ) {
                add_image_size( 'unero-product-masonry-large', 370, 499, true );
                add_image_size( 'unero-product-masonry-long', 270, 364, true );
        }

        if ( in_array( 'portfolio_masonry', $image_sizes ) ) {
                add_image_size( 'unero-portfolio-masonry', 540, 670, false );
        }

        if ( in_array( 'portfolio_carousel', $image_sizes ) ) {
                add_image_size( 'unero-portfolio-carousel', 1170, 644, true );
        }

        // Initialize

        global $unero_woocommerce;
        $unero_woocommerce = new Unero_WooCommerce;

        if ( is_admin() ) {
                new Unero_Meta_Box_Product_Data;
        }
}

add_action( 'after_setup_theme', 'unero_setup' );


/**
 * Register widgetized area and update sidebar with default widgets.
 *
 * @since 1.0
 *
 * @return void
 */
function unero_register_sidebar() {
        // Register primary sidebar
        $sidebars = array(
                'blog-sidebar'      => esc_html__( 'Blog Sidebar', 'unero' ),
                'menu-sidebar'      => esc_html__( 'Menu Sidebar', 'unero' ),
                'catalog-sidebar'   => esc_html__( 'Catalog Sidebar', 'unero' ),
                'catalog-filter'    => esc_html__( 'Catalog Filter', 'unero' ),
                'portfolio-sidebar' => esc_html__( 'Portfolio Sidebar', 'unero' ),
                'page-sidebar'      => esc_html__( 'Page Sidebar', 'unero' ),
                'product-sidebar'   => esc_html__( 'Product Sidebar', 'unero' ),
        );

        // Register footer sidebars
        for ( $i = 1; $i <= 6; $i ++ ) {
                $sidebars["footer-sidebar-$i"] = esc_html__( 'Footer', 'unero' ) . " $i";
        }

        // Register sidebars
        foreach ( $sidebars as $id => $name ) {
                register_sidebar(
```

```
                        array(
                                'name'          => $name,
                                'id'            => $id,
                                'before_widget' => '<div id="%1$s" class="widget %2$s">',
                                'after_widget'  => '</div>',
                                'before_title'  => '<h4 class="widget-title">',
                                'after_title'   => '</h4>',
                        )
                );
        }
}

add_action( 'widgets_init', 'unero_register_sidebar' );

/**
 * Load theme
 */


// customizer hooks
require get_template_directory() . '/inc/customizer/customizer.php';

require get_template_directory() . '/inc/frontend/woocommerce.php';

require get_template_directory() . '/inc/functions/entry.php';

require get_template_directory() . '/inc/functions/media.php';


if ( is_admin() ) {
        require get_template_directory() . '/inc/libs/class-tgm-plugin-activation.php';
        require get_template_directory() . '/inc/backend/plugins.php';
        require get_template_directory() . '/inc/backend/meta-boxes.php';
        require get_template_directory() . '/inc/mega-menu/class-mega-menu.php';
        require get_template_directory() . '/inc/backend/product-meta-box-data.php';
        require get_template_directory() . '/inc/backend/importer.php';
        require get_template_directory() . '/inc/backend/editor.php';
} else {
        // Frontend functions and shortcodes
        require get_template_directory() . '/inc/functions/header.php';
        require get_template_directory() . '/inc/functions/layout.php';
        require get_template_directory() . '/inc/functions/breadcrumbs.php';
        require get_template_directory() . '/inc/functions/nav.php';
        require get_template_directory() . '/inc/functions/class-menu-walker.php';
        require get_template_directory() . '/inc/mega-menu/class-mega-menu-walker.php';
        require get_template_directory() . '/inc/functions/footer.php';
        require get_template_directory() . '/inc/functions/options.php';

        // Frontend hooks
        require get_template_directory() . '/inc/frontend/layout.php';
        require get_template_directory() . '/inc/frontend/header.php';
        require get_template_directory() . '/inc/frontend/nav.php';
        require get_template_directory() . '/inc/frontend/entry.php';
        require get_template_directory() . '/inc/frontend/comments.php';
        require get_template_directory() . '/inc/frontend/footer.php';
}
```

\unero\index.php

```
<?php
/**
 * The main template file.
 *
 * This is the most generic template file in a WordPress theme
 * and one of the two required files for a theme (the other being style.css).
 * It is used to display a page when nothing more specific matches a query.
 * E.g., it puts together the home page when no home.php file exists.
 * Learn more: http://codex.wordpress.org/Template_Hierarchy
 *
 * @package Unero
 */

get_header(); ?>

<div id="primary" class="content-area <?php unero_content_columns() ?>">
        <?php do_action( 'unero_before_content' ); ?>
        <main id="main" class="site-main">
                <?php if ( have_posts() ) : ?>

                        <?php $blog_view = 'blog-layout-' . unero_get_option( 'blog_view' ); ?>
```

```
                        <div id="unero-site-content" class="<?php echo esc_attr( $blog_view );
?>">
                                <div class="unero-post-list">
                                        <?php /* Start the Loop */ ?>
                                        <?php while ( have_posts() ) : the_post(); ?>

                                                <?php
                                                /* Include the Post-Format-specific
template for the content.
                                                 * If you want to override this in a child
theme, then include a file
                                                 * called content-___.php (where ___ is the
Post Format name) and that will be used instead.
                                                 */
                                                get_template_part( 'template-
parts/content', get_post_format() );
                                                ?>

                                        <?php endwhile; ?>
                                </div>
                        </div>

                        <?php unero_paging_nav(); ?>

                <?php else : ?>
                        <?php get_template_part( 'template-parts/content', 'none' ); ?>
                <?php endif; ?>

        </main>


        <!-- #main -->
</div><!-- #primary -->

<?php get_sidebar(); ?>
<?php get_footer(); ?>
```