

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач випускової кафедри
_____ Аліна САВЧЕНКО
« ____ » _____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА
(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВР
ЗА ОСВІТНЬО-ПРОФЕСІЙНОЮ ПРОГРАМОЮ
«ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ПРОЕКТУВАННЯ»

**Тема: «Веб-застосунок інтернет-магазину взуття з адміністративною
панеллю на основі PHP»**

Виконавець: Олександр ШОСТАКІВСЬКИЙ
Керівник: к.т.н., доцент Олена ТОЛСТІКОВА
Нормоконтролер: к.т.н., доцент Олена ТОЛСТІКОВА

6. Календарний план-графік

№ з/п	Завдання	Термін виконання	Підпис керівника
1.	Огляд та опис теорії розробки веб-застосунків. Написання першого розділу та презентація керівнику.	15.05.2023 - 19.05.2023	
2.	Розробка дизайну та front-end частини застосунку. Написання другого розділу та презентація керівнику.	20.05.2023 - 25.05.2023	
3.	Розробка архітектури бази даних та back-end частини застосунку. Написання третього розділу та презентація керівнику.	26.05.2023 - 05.06.2023	
4.	Проходження перевірки на плагіат.	06.06.2023 - 07.06.2023	
5.	Редагування та друк пояснювальної записки.	08.06.2023	
6.	Проходження нормоконтролю та перепліт пояснювальної записки.	09.06.2023- 13.06.2023	
7.	Написання доповіді та оформлення презентації.	14.06.2023- 18.06.2023	

7. Дата видачі завдання 15.05.2023р.

Керівник кваліфікаційної роботи _____ Олена ТОЛСТІКОВА
(підпис керівника)

Завдання прийняв до виконання _____ Олександр ШОСТАКІВСЬКИЙ
(підпис випускника)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи на тему: «Веб-застосунок інтернет-магазину взуття з адміністративною панеллю на основі PHP» містить: 69 сторінок, 38 рисунків, 20 інформаційних джерел, 5 додатків.

Об'єкт дослідження – веб-застосунок.

Предмет дослідження – веб-застосунок інтернет-магазину взуття з адміністративною панеллю на основі PHP.

Мета кваліфікаційної роботи – розробити веб-застосунок інтернет-магазину взуття на основі PHP.

Методи дослідження – аналітичний, порівняльний, прототипний та тестувальний.

Результати кваліфікаційної роботи рекомендується використовувати для розробки та управління веб-застосунком інтернет-магазину на основі PHP.

Для розробки веб-застосунку інтернет-магазину було встановлено та використано відкритий інструментарій для роботи з образами віртуальних машин Vagrant у парі з Homestead, а також веб-сервер і проксі-сервер nginx, фотографії товарів з доступних джерел, готові матеріали та шаблони.

NGINX, HOMESTEAD, ДИЗАЙН, FIGMA, FRONT-END, ВЕРСТКА, HTML, CSS, JAVASCRIPT, JQUERY, PHPSTORM, SQL, MYSQL, PHP, АРХІТЕКТУРА, РОЗРОБКА, ТЕСТУВАННЯ, ІНТЕРНЕТ-МАГАЗИН.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ	6
ВСТУП.....	7
РОЗДІЛ 1. ЗАГАЛЬНІ ПРИНЦИПИ СТВОРЕННЯ ВЕБ-ЗАСТОСУНКІВ.....	11
1.1. Підходи до розробки веб-застосунків та їх методи	12
1.1.1. Каскадна модель	12
1.1.1. Каскадна модель	14
1.1.2. Ітеративний та інкрементальний підхід	16
1.2. Інструменти для розробки веб-застосунків	17
1.2.1. Менеджери пакетів та виконувачі завдань	18
1.2.2. Мови та платформи	20
1.2.3. Бази даних.....	30
1.2.5. Веб-сервери	36
ВИСНОВКИ ДО РОЗДІЛУ 1	38
РОЗДІЛ 2. ДИЗАЙН ТА FRONT-END ЧАСТИНА ВЕБ-ЗАСТОСУНКУ	39
2.1. Розробка дизайну	39
2.2. Розробка front-end частини.....	44
ВИСНОВОК ДО РОЗДІЛУ 2	47
РОЗДІЛ 3. АРХІТЕКТУРА БАЗИ ДАНИХ ТА BACK-END ЧАСТИНА ВЕБ-ЗАСТОСУНКУ	48
3.1. Розробка архітектури бази даних	48
3.2. Розробка back-end частини застосунку	53
3.2.1. Розробка back-end для користувацької частини застосунку	53
3.2.2. Розробка back-end для адміністративної частини застосунку.....	56
ВИСНОВОК ДО РОЗДІЛУ 3	61
ВИСНОВКИ.....	62
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	64
ДОДАТОК А. КАРТКА ТОВАРУ	66
ДОДАТОК Б. КОШИК ЗАМОВЛЕННЯ.....	67
ДОДАТОК В. КАТАЛОГ ТОВАРІВ	68
ДОДАТОК Г. КЕРУВАННЯ ЛИСТАМИ В АДМІНІСТАТИВНІЙ ПАНЕЛІ	68
ДОДАТОК Ґ. КЕРУВАННЯ ТОВАРАМИ В АДМІНІСТАТИВНІЙ ПАНЕЛІ.....	69

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

Front-end	—	користувацький інтерфейс
Back-end	—	об'єднання сервера з користувачем
Header	—	верхня частина веб-сторінки
Footer	—	секція, розташована внизу веб-сторінки
Слайдер	—	інтерактивний елемент веб-сторінки, який дозволяє відображати і змінювати зображення або контент у вигляді слайдів
DDoS	—	вид кібератаки, в якій зловмисники намагаються перевантажити цільовий сервер або мережу, надсилаючи велику кількість запитів з різних джерел одночасно.
Пул	—	група ресурсів або об'єктів, які об'єднуються разом для спільного використання та ефективного управління.
Коміт	—	збереження змін у репозиторії.
Лінтер	—	інструмент аналізу коду, який використовується в процесі розробки програмного забезпечення для виявлення потенційних проблем, помилок або стилістичних недоліків у вихідному коді.
Фреймворк	—	структурований набір програмних компонентів, бібліотек та інструментів, який надає основу для розробки програмного забезпечення.
IDE	—	інтегроване середовище розробки, що об'єднує різноманітні інструменти та функціональність.
Бургер-меню	—	тип навігаційного меню, який зазвичай використовується на мобільних пристроях або для адаптивного дизайну веб-сайту.

ВСТУП

У сучасному світі Інтернет не просто система комп'ютерних мереж, де зберігається неймовірна кількість різної інформації, а цілий віртуальний всесвіт.

Інтернет-технології суттєво змінили наше життя за останні десятиліття.

Електронна комерція впевнено стає частиною нашого сьогодення та особливо активно розвивається у всіх своїх напрямках. Сучасним торговим каналом є інтернет-магазин — один з пріоритетних напрямків економічної діяльності. Цей інструмент дає можливість продавати товари і послуги величезній аудиторії, що має доступ до інтернету. Його головна ціль — приносити прибуток. Тож, створюючи інтернет-магазин, необхідно аналізувати усі тенденції використання новітніх інструментів та методи їх просування, впроваджувати нові технології для досягнення передових позицій у бізнесі.

Зважаючи на те, що інтернет-магазини повільно, але вірно витісняють своїх прямих конкурентів на ринку, цілком ймовірним є те, що вони зовсім скоро повністю заповнять ринок торгівлі, адже торгівля в інтернеті — це прогресивний бізнес з високою конкуренцією. Тепер те, як виглядає ваш сайт, каталог товарів і наскільки успішна ваша реклама, може сказати про те, наскільки успішний ваш бізнес.

Зростаюча популярність онлайн-магазинів відкрила багато можливостей для підприємців, які прагнуть розширити свій бізнес і залучити нових клієнтів саме до свого ресурсу. Адже, створення комфортних умов для споживача — чи не найголовніше завдання бізнесу. Справжнім трендом серед підприємців стала розробка веб-застосунку власного інтернет-магазину.

Моя кваліфікаційна робота спрямована на розробку веб-застосунку інтернет-магазину взуття з панеллю адміністрування на основі PHP. Цей проект є відповіддю на реальну потребу нашого часу у створенні зручного та повнофункціонального інтернет-магазину. У моїй роботі показано основні аспекти розробки інтернет-магазину з використанням мови програмування

PHP та бази даних MySQL. Проект містить детальний опис функціоналу та можливостей веб-додатків, призначених для полегшення бізнес-процесів.

Особливу увагу приділено адміністративній панелі, яка дозволяє власникам ефективно та зручно керувати товарами, замовленнями та іншими аспектами свого магазину. Інтерфейс адміністративної панелі забезпечує зручний доступ до основних функцій і підвищує ефективність роботи.

Дана кваліфікаційна робота відображає процес створення функціонального, безпечного та прогресивного онлайн-магазину взуття. Проект націлений на розвиток бізнесу, залучення нових клієнтів, позитивний вплив на економіку та ринок праці.

Процес розробки інтернет-магазину на основі PHP, починаючи з аналізу вимог і закінчуючи тестуванням і запуском продукту, вимагає передових навичок і досвіду веб-розробки. Ця робота демонструє мої знання та навички в цій галузі.

Всі матеріали та результати моєї роботи можуть бути корисні іншим розробникам і власникам бізнесу, які планують створити власний інтернет-магазин, оскільки цей проект містить широкі технічні та бізнес-знання для відкриття власного онлайн-магазину та досягнення успіхів на ринку.

Дана кваліфікаційна робота демонструє мої знання та навички програмування, бізнес-планування, а також мою здатність розробляти функціональні та ефективні веб-додатки. Впевнений, що результати цієї роботи допоможуть іншим розробникам і власникам бізнесу, які планують розвивати свою справу за допомогою інтернет-маркетингу та онлайн-продажу.

Актуальність теми кваліфікаційної роботи «Веб-застосунок інтернет-магазину взуття з адміністративною панеллю на основі PHP» у сучасному цифровому світі є дійсно вагомою. Вона поєднує два важливі аспекти - електронну комерцію та веб-розробку, які займають центральне місце в онлайн-торгівлі.

Перш за все, електронна комерція стала надзвичайно популярною і широко використовується по всьому світу. Інтернет-магазини дозволяють

покупцям зручно і швидко здійснювати покупки в Інтернеті, а продавцям - ефективно просувати свої товари та послуги, привертати нових клієнтів та збільшувати прибуток. Застосуванням технологій веб-розробки для створення інтернет-магазину взуття можна створити зручне та привабливе середовище для клієнтів, де вони зможуть переглядати товари, дізнаватися деталі, здійснювати покупки та проводити оплату, все це з комфортом свого дому або офісу.

Об'єктом дослідження кваліфікаційної роботи є веб-застосунок інтернет-магазину.

Предмет дослідження – веб-застосунок інтернет-магазину взуття з адміністративною панеллю на основі PHP.

Мета кваліфікаційної роботи – створити веб-застосунок інтернет-магазину взуття на основі PHP.

Відповідно до поставленої мети роботи визначено основні завдання дослідження:

- дослідити та висвітлити основні підходи та методи розробки веб-застосунків;
- порівняти та проаналізувати популярні інструменти для розробки веб-застосунків;
- розробити макети дизайну веб-застосунку;
- розробити front-end частину застосунку;
- розробити архітектуру бази даних інтернет-магазину;
- розробити back-end для користувацької частини застосунку;
- розробити back-end для адміністративної частини застосунку.

Для досягнення поставленої мети й виконання завдань використано такі методи: аналітичний, порівняльний, прототипний та тестувальний.

Наукова новизна роботи полягає у розробці електронної комерції та внесок у покращення процесу роботи веб-магазинів, висвітленню та аналізу теорії та методологій, оскільки це сприяє кращому розумінню та вибору оптимального підходу для розробки веб-магазину. Важливу роль у новизні

роботи сприяє звернення до принципів дизайну, створення привабливого та зручного інтерфейсу користувача відображає інноваційний підхід до розробки веб-магазинів, з метою поліпшення взаємодії з користувачем.

Практичне значення отриманих результатів цієї кваліфікаційної роботи дають необхідні знання та навички для розробки та управління веб-застосунком інтернет-магазину на основі PHP. Вони дозволять створити функціональний, привабливий та ефективний інтернет-магазин, що забезпечить задоволення потреб цільової аудиторії та сприятиме успішному розвитку бізнесу.

РОЗДІЛ 1

ЗАГАЛЬНІ ПРИНЦИПИ СТВОРЕННЯ ВЕБ-ЗАСТОСУНКІВ

Розробка веб-додатків є дуже важливим процесом у сучасному світі, де веб-технології домінують у більшості аспектів життя та бізнесу. Веб-програми можуть імітувати роль стандартних настільних програм, але з багатьма перевагами: доступ до нього можна отримати з будь-якого місця, де є підключення до Інтернету, не вимагає встановлення додаткового програмного забезпечення та масштабується відповідно до мінливих потреб користувачів.

Однією з головних переваг розробки веб-додатків є їх доступність. Веб-програми дозволяють користувачам взаємодіяти з програмою з будь-якого місця, де є доступ до Інтернету. Це підвищує вашу продуктивність і комфорт під час використання програмного забезпечення. Ще однією важливою перевагою веб-додатків є простота їх розгортання та обслуговування. Користувачам не потрібно встановлювати програмне забезпечення, що скорочує час і зусилля, необхідні для використання програми. Це також значно полегшує розробникам веб-додатків підтримку свого програмного забезпечення та забезпечення його постійної доступності.

З найважливіших характеристик розробки веб-додатків варто виділити їх масштабованість. Веб-програми можна легко модифікувати та розширювати відповідно до мінливих потреб користувачів. Завдяки технологіям веб-розробки, розробники можуть легко збільшувати чи зменшувати ресурси, необхідні для роботи веб-застосунків, що дозволяє підтримувати їх продуктивність та доступність в майбутньому.

Веб-застосунки можуть працювати на будь-якій операційній системі, що підтримує веб-браузер, тому користувачі можуть легко отримати доступ до веб-застосунків на будь-якому пристрої з доступом до Інтернету.

Кафедра КІТ				НАУ 23 38 33 000 ПЗ						
	ПІБ			РОЗДІЛ 1. ЗАГАЛЬНІ ПРИНЦИПИ СТВОРЕННЯ ВЕБ-ЗАСТОСУНКІВ			Літ.	Аркуш	Аркушів	
Розроб.	Шостаківський О.О.								11	28
Керівник	Толстікова О.В.						ТП-415Б – 122			
Н. Контр.	Толстікова О.В.									

Загалом, розробка веб-застосунків дуже важлива та має багато переваг. Веб-застосунки забезпечують доступність, легкість в розгортанні та підтримці, масштабованість та кросплатформеність. Ці переваги дозволяють розробникам створювати додатки, які можуть задовольнити потреби різних користувачів та бізнесів, збільшуючи продуктивність та ефективність роботи.[1, 2]

1.1. Підходи до розробки веб-застосунків та їх методи

Багато традиційних методів розробки систем вважаються непридатними для розробки систем електронної комерції. Інтернет справив величезний вплив на світ бізнесу. Зміни в бізнес-середовищі також впливають на вимоги до підходів до розробки системи. Як розробнику, важливо знати ці аспекти, щоб використовувати правильні методи, підходи до проектування та інструменти в процесі розробки. У цьому розділі описуються характеристики програм електронної комерції та їх вплив на процес розробки, а також аналізуються кілька методів електронної комерції.

Підходи до розробки веб-застосунків можна розбити на декілька груп в залежності від фокусу, підходу та методів, які використовуються в процесі розробки.

1.1.1. Каскадна модель

Перший метод, який хотів би розглянути - це «водоспад» або «каскадна» модель розробки програмного забезпечення. Цей метод дуже простий, лінійний і широко використовується у веб-розробці.

Основна ідея цього методу полягає в розділенні розробки веб-додатку на послідовні фази. Кожна фаза містить певний набір дій, які виконуються по порядку. З цієї причини метод іноді називають «лінійним».

Основні етапи каскадної моделі розробки ПЗ:

- Визначення вимог: Під час цього етапу збираються вимоги до веб-додатків, у тому числі функціональні та нефункціональні вимоги, такі як макети екрана, вимоги безпеки та пропускна здатність;
- Аналіз: На цьому етапі виконується аналіз вимог і створюється детальний план розробки веб-додатку. Розроблено функціональні вимоги до веб-додатку, визначено архітектуру веб-додатку, створено технологічний стек, створено діаграми та специфікації;
- Дизайн: На цьому етапі розробляються дизайн веб-додатку та інтерфейс користувача, включаючи зовнішній вигляд, макет та інші елементи;
- Розробка: На цьому етапі розробляються компоненти веб-додатку та реалізуються функціональні вимоги до веб-додатку;
- Тестування: Під час цього етапу веб-програма перевіряється на наявність помилок і несправностей;
- Деплоймент та експлуатація: На цьому етапі веб-додаток публікується в середовищі виробництва, а також забезпечується його експлуатація та підтримка.

Один з головних недоліків цього методу полягає в тому, що він не передбачає можливості внесення змін після завершення кожного етапу. Тому якщо виявляються помилки на пізніших етапах, то змінити їх може бути важко або навіть неможливо. Також Каскадна модель розробки програмного забезпечення передбачає високі витрати на корекцію помилок на пізніших етапах розробки, оскільки виправлення помилок вимагає зміни вже розробленого коду.

Проте, на користь цього методу можна сказати, що він досить простий для розуміння та виконання, тому є популярним серед розробників веб-додатків. Його також можна успішно використовувати для розробки веб-додатків із відомими вимогами, коли результат кожної фази можна передбачити досить точно.

Крім каскадної моделі, існують методики розробки веб-додатків, такі як «Agile» і «Scrum». Ці методи більш гнучкі, дозволяючи вносити зміни на ранніх етапах розробки та ітераційної розробки веб-додатків. Однак для цього потрібно більше обміну даними між розробниками, менеджерами та клієнтами, що може призвести до більш тривалого часу розробки та підвищення вартості проекту.

1.1.1. Каскадна модель

Гнучка методологія розробки (далі Agile) — підхід до розробки програмного забезпечення, заснований на ітераційному, поетапному підході, а також постійній модифікації та вдосконаленні процесів. Agile прагне забезпечити високу якість програмного забезпечення, швидкість реагування на зміну вимог клієнтів, підвищення рівня задоволеності користувачів і зниження витрат на розробку.

– Гнучка методологія розробки програмного забезпечення використовує такі принципи:

- Індивідуали та інтеракції більш важливі, ніж процеси та інструменти.
- Робота програмного забезпечення більш важлива, ніж докладна документація.
- Співпраця з клієнтом більш важлива, ніж угода про контракт.
- Реагування на зміни більш важливе, ніж дотримання плану.

Agile використовує ітераційний підхід до розробки програмного забезпечення, розділяючи весь процес на менші ітерації, які називаються спринтами. Кожен спринт має фіксовану тривалість і має власний список завдань, які необхідно виконати.

Agile також включає поступовий підхід до розробки програмного забезпечення. У цьому підході програмні продукти розробляються поступово, включаючи нові функції та виправляючи помилки з кожною новою ітерацією. Це дозволяє швидко реагувати на зміну вимог користувачів, покращувати якість продукту та зменшувати ризики та витрати.

Agile включає набір практик і методологій, які сприяють успішній реалізації проекту. Ці методи включають:

- Kanban — це система управління, яка візуально відображає статус проекту. Kanban містить дошку, яка показує завдання, їхній статус і пріоритет, допомагаючи командам візуалізувати процес розробки та легше керувати завданнями;

- XP (Extreme Programming) — це підхід до забезпечення високої якості програмного забезпечення за допомогою набору практик, таких як парне програмування та тестування перед випуском;

- Lean – це підхід, спрямований на максимальну ефективність і мінімальну вартість розробки програмного забезпечення.

Agile є популярним і ефективним підходом у розробці програмного забезпечення, особливо в контексті вимог, що постійно змінюються, і швидкого розвитку технологій. Agile дозволяє командам швидко адаптуватися до змін і покращувати якість програмного забезпечення на кожному етапі розробки. Завдяки ітераційному, покроковому підходу Agile дозволяє швидко досягати результатів і отримувати відгуки від користувачів.

Одним із основних принципів Agile є взаємодія та співпраця між розробниками, тестувальниками та бізнес-клієнтами. Це забезпечує високу якість програмного забезпечення та більш ефективне вирішення завдань. Гнучкість також забезпечує відкритість і прозорість у процесі розробки. Команда розробників регулярно зустрічається з бізнес-клієнтами, щоб уточнити їхні вимоги та переконатися, що продукт відповідає їхнім потребам.

Крім того, Agile дозволяє командам відкрито обговорювати проблеми та рішення. Загалом Agile — це підхід до розробки програмного забезпечення, який забезпечує швидку адаптацію до змін, високу якість продукту та ефективну співпрацю між членами команди. Agile завдяки своїм методам і практикам робить процес розробки програмного забезпечення більш гнучким і ефективним для розробників.

1.1.2. Ітеративний та інкрементальний підхід

Scrum — це ітеративний та поступовий підхід до розробки програмного забезпечення, який дозволяє командам розробників працювати ефективно та швидко. Scrum заснований на принципах Agile і складається з трьох основних компонентів: Ролі, збірки та артефакти.

Ролі Scrum включають власника продукту, Scrum Master і команду розробки. Власник продукту створює та підтримує Беклог продукту та несе відповідальність за визначення пріоритетів і того, які функції слід розробляти в першу чергу. Scrum Master відповідає за забезпечення виконання процесу Scrum і допомагає команді розробників досягти поставлених цілей. Команда розробників складається з кількох професіоналів, які разом працюють над розробкою продукту та досягненням його цілей.

Збори Scrum включають планування спринту, щоденний Scrum, огляд спринту та ретроспективу спринту. Планування спринту — це зустріч, на якій команда розробників і власник продукту обговорюють і приймають рішення про завдання, які потрібно виконати в наступному спринті. Щоденна бійка — це коротка щоденна зустріч, на якій команда розробників обговорює прогрес і наступні кроки. Спринт-огляд — це зустріч, на якій команда розробників представляє завершену роботу власнику продукту та іншим зацікавленим сторонам.

Ретроспектива спринту — це зустріч, на якій команда розробників обговорює, що пройшло добре, а що можна було б покращити в наступному спринті.

Артефакти Scrum включають Backlog продукту та Backlog Sprint. Беклог продукту — це список функцій, які необхідно розробити для продукту. Список може бути змінений, доповнений і оновлений в процесі розробки. Беклог спринту — це список завдань, які команда розробників має виконати під час спринту. Список створюється на нараді з планування спринту та може бути змінений на ретроспективній нараді спринту.

Окрім основних компонентів, Scrum має кілька принципів, які сприяють успіху вашого проекту. Одним із цих принципів є поступовий розвиток. Це означає, що групи розробників працюють над продуктами невеликими ітераціями (спринтами), додаючи нові функції до кожного продукту. Це дозволяє командам розробників швидко й ефективно реагувати на змінені вимоги. Іншим принципом є активний діалог із зацікавленими сторонами, включаючи власників продуктів та інших зацікавлених користувачів. Це дозволяє команді розробників зрозуміти вимоги користувачів і визначити пріоритети розробки.

Scrum також заохочує постійне вдосконалення процесу розробки та команди розробників. На ретроспективі спринту команда розробників обговорює, що вийшло добре, що можна було б покращити, і вносить зміни у свій підхід до розробки.

Підводячи підсумок, Scrum — це ефективний і популярний підхід до розробки програмного забезпечення, який дозволяє командам розробників працювати швидко та ефективно, забезпечуючи при цьому якість і гнучкість розробки продукту.

Кожен із цих підходів передбачає різні методології розробки, зокрема: В. Тестування, контроль версій, автоматизація тощо. Результатом кожного з цих підходів і методів є веб-додаток, який можна використовувати для різних цілей і завдань. [3, 4]

1.2. Інструменти для розробки веб-застосунків

Плюси та мінуси веб-розробника полягають у тому, що Інтернет постійно змінюється. Незважаючи на те, що це захоплююче, це також означає, що веб-розробники завжди повинні бути готові вивчати нові техніки та мови програмування, адаптуватися до змін, а також бути готовими та готовими приймати нові виклики. Це включає такі завдання, як адаптація існуючих структур до потреб бізнесу, тестування веб-сайтів для виявлення технічних проблем, а також оптимізація та масштабування веб-сайтів для кращої роботи

з серверною інфраструктурою. Я підготував список інструментів і ресурсів, які зазвичай використовуються для веб-розробки.

1.2.1. Менеджери пакетів та виконувачі завдань

Виконувачі завдань призначені для автоматизації робочих процесів. Наприклад, ви можете створити завдання для автоматизації мінімізації JavaScript. Потім створюйте та комбінуйте завдання, щоб скоротити час розробки. Менеджер пакунків відстежить всі пакети, якими ви користуєтеся, і переконається, що вони оновлені та мають конкретну версію, яка вам потрібна.

Grunt

Grunt являє собою інструмент автоматизації завдань для розробників програмного забезпечення. Ви можете автоматизувати повторювані завдання, такі як компіляція файлів, тестування, мінімізація та об'єднання, копіювання файлів і розгортання. Grunt підтримує багато плагінів, які можуть легко розширити його функціональність.

Основна перевага Grunt – економічність часу. Grunt миттєво економить ваш час на повторюваних завданнях. Крім того, використання Grunt дозволяє уникнути помилок, які виникають під час запуску завдань вручну.

Grunt використовує конфігураційні файли для опису завдань, які він виконує. Ці файли містять інформацію про використовувані плагіни, параметри та параметри конфігурації.

Однією з особливостей Grunt є можливість використання плагінів, які дозволяють розширити функціональність інструменту за допомогою спеціальних програм.

Ще однією особливістю Grunt є можливість виконувати завдання за певних умов. Наприклад, ви можете запустити завдання, лише якщо файл змінився. Крім того, Grunt дозволяє легко виконувати завдання на кількох платформах, таких як Windows, Linux і Mac OS X.

npm

npm (Node Package Manager) — це менеджер пакетів для платформи Node.js. Розробники Node.js можуть легко керувати залежностями, які використовуються в їхніх проектах.

Основна мета npm — дозволити розробникам легко та швидко встановлювати, оновлювати та видаляти пакети, які використовуються в їхніх проектах. npm забезпечує це через файл `package.json`. Цей файл зберігає інформацію про залежності проекту, як-от імена, версії та іншу інформацію про пакет, необхідну для належного функціонування проекту.

Одним із основних моментів npm є його репозиторій пакетів, який містить понад мільйон пакетів з відкритим кодом, створених розробниками з усього світу. Кожен із цих пакетів можна легко встановити у ваш проект за допомогою команди `npm install`.

Крім того, npm дозволяє розробникам створювати власні пакети та публікувати їх у сховищі для використання іншими розробниками у своїх проектах.

Окрім керування залежностями та публікації пакетів, npm також надає інші корисні функції, такі як виявлення пакетів, оновлення пакетів і версії пакетів. Його також можна використовувати для керування проектами, допомагаючи створювати та запускати проекти з конфігураціями за замовчуванням.

Webpack

Webpack — це інструмент для збирання інтерактивних веб-додатків із кількох файлів, таких як JavaScript, CSS, HTML і зображення. Основна мета Webpack — допомогти вам ефективно обробляти модулі, розбиваючи ваш код на менші, більш керовані частини.

Особливості Webpack:

- Модульність: Webpack дозволяє розбити код на менші частини, якими легше керувати та повторно використовувати;

- Збірка: Webpack дозволяє об'єднувати різні типи файлів в один файл, щоб зменшити кількість запитів, необхідних для завантаження сторінки;
- Зміни коду: Webpack дозволяє застосовувати різні перетворення до вашого коду;
- Підтримка локалізації: Webpack можна налаштувати для підтримки локалізації за допомогою різних плагінів і конфігурацій;

Однією з найкорисніших функцій Webpack є можливість використовувати плагіни для розширення його функціональності. Наприклад, плагіни стиснення зображень допомагають зменшити розмір зображень і прискорити час завантаження сторінки.

1.2.2. Мови та платформи

За кожним інструментом веб-розробки стоїть мова. Мова програмування — це формальна структурована мова, призначена для написання програм, які можуть спілкуватися з комп'ютерами та контролювати їх поведінку. Далі наведено перелік популярних мов програмування та розмітки, які використовуються для веб-розробки.

PHP

PHP розшифровується як «PHP: Hypertext Preprocessor» — це мова програмування сценаріїв, яка використовується для розробки веб-додатків і динамічних веб-сторінок. Вона широко використовується у веб-розробці через її простоту використання та здатність взаємодіяти з різними базами даних та іншими зовнішніми джерелами даних. Відомо.

PHP є мовою з відкритим кодом. Це означає, що вихідний код доступний кожному, хто хоче його використати або змінити. Це дозволяє розробникам додавати нові функції та вдосконалювати мову.

PHP можна використовувати для створення блогів, форумів, соціальних мереж, електронних магазинів та багатьох інших різноманітних веб-додатків. PHP може працювати на будь-якому веб-сервері, який підтримує PHP. Наприклад: Apache або NGINX.

Особливості PHP:

- Динаміка: PHP може створювати динамічні веб-сторінки. Це означає, що вміст сторінки може змінюватися у відповідь на дії користувача чи іншу зовнішню інформацію;
- Розмаїття функцій: PHP має велику кількість вбудованих функцій і бібліотек, що робить його дуже потужним для розробки різних веб-додатків;
- Підтримка бази даних: PHP добре працює з різними базами даних, такими як MySQL, PostgreSQL і Oracle. Це дозволяє розробникам легко зберігати та отримувати дані з баз даних;
- Масштабованість: PHP дозволяє створювати складні веб-додатки та масштабувати їх відповідно до зростаючих потреб користувачів;
- Кросплатформенність: PHP працює в будь-якій операційній системі, включаючи Windows, Linux і macOS, а також може використовуватися з великою кількістю веб-серверів;
- Безпека: PHP має вбудований захист від XSS (міжсайтових сценаріїв), впровадження SQL та інших атак;
- Відкритий код: PHP є мовою з відкритим кодом. Це означає, що він безкоштовний для використання та може бути змінений і вдосконалений спільнотою розробників.

Розглянемо приклад використання PHP для підключення до MySQL бази даних (рис. 1.1):

```
<?php
// Параметри для підключення до бази даних
$server = "localhost"; // сервер бази даних
$username = "username"; // ім'я користувача бази даних
$password = "password"; // пароль користувача бази даних
$db = "database"; // назва бази даних
// Підключення до бази даних
$conn = mysqli_connect($server, $username, $password, $db);
// Перевірка підключення до бази даних
```

```
if (!$connection) {  
    dd("Помилка підключення до бази даних: " . mysqli_connect_error());  
}  
dd("Підключення до бази даних успішно встановлено!");  
?>
```

A screenshot of a terminal window with a black background and green text. The text reads: "Підключення до бази даних успішно встановлено!".

Рис. 1.1. Використання PHP

Спочатку визначаються параметри для підключення, такі як сервер бази даних (`$server`), ім'я користувача (`$username`), пароль користувача (`$password`) та назва бази даних (`$db`). Потім функція `mysqli_connect()` використовується для підключення до бази даних з заданими параметрами, і повертає об'єкт з'єднання. Якщо підключення не вдалося, виконується команда `dd()` з повідомленням про помилку, що містить текстову помилку, повернену функцією `mysqli_connect_error()`. Якщо підключення вдалося, виконується команда `dd()` з повідомленням про успішне підключення до бази даних.

Загалом PHP є потужною мовою програмування, яка справила великий вплив на веб-розробку. Він простий у використанні, має широкий набір функцій і дозволяє створювати динамічні та масштабовані веб-програми. PHP — чудовий вибір, якщо ви плануєте створювати веб-додатки.

JavaScript

JavaScript — це мова програмування, яка зазвичай використовується для розробки динамічних веб-сайтів та інтерактивних програм. Мова може використовуватися як на стороні клієнта, так і на стороні сервера.

Повертаючись в історію, варто зауважити, що JavaScript був розроблений у 1995 році Бренданом Айхом для Netscape. Спочатку мова називалася Mocha, потім перейменована на LiveScript і, нарешті, на JavaScript. Сьогодні JavaScript є однією з найпопулярніших мов програмування, яка використовується для розробки веб-сайтів, мобільних додатків, настільних додатків і багатьох інших додатків.

Особливості JavaScript:

– Синтаксис: Синтаксис JavaScript досить простий для вивчення та розуміння. Багато елементів, таких як функції, змінні та об'єкти, використовуються в JavaScript з тими ж іменами, що й в інших мовах програмування;

– Взаємодія з HTML і CSS: JavaScript може взаємодіяти зі сторінками HTML і CSS, дозволяючи розробникам створювати динамічні ефекти та змінювати зовнішній вигляд сторінки на основі дій користувача;

– Об'єктно-орієнтований підхід: JavaScript — це мова програмування, яка підтримує об'єктно-орієнтований підхід;

– Функції вищого порядку: JavaScript підтримує функції вищого порядку, тому ви можете передавати функції як параметри та повертати їх як результати інших функцій;

– Не сувора типізація: JavaScript — це мова з вільною типізацією. Це означає, що тип змінних може змінюватися під час виконання програми;

– Асинхронність: JavaScript підтримує асинхронний підхід. Це дозволяє створювати програми, які можуть виконувати багато операцій одночасно, не блокуючи інші процеси;

– Мультиплатформенність: JavaScript можна використовувати на багатьох платформах, включаючи браузер, сервери, мобільні пристрої.

Наведено приклад використання JavaScript для реалізації сценарію, що відображає відомості про браузер, який використовується (рис. 1.2):

```
const browserInfo = `Browser: ${navigator.appName} ${navigator.appVersion}  
Language: ${navigator.language}  
Platform: ${navigator.platform}`;  
alert(browserInfo);
```

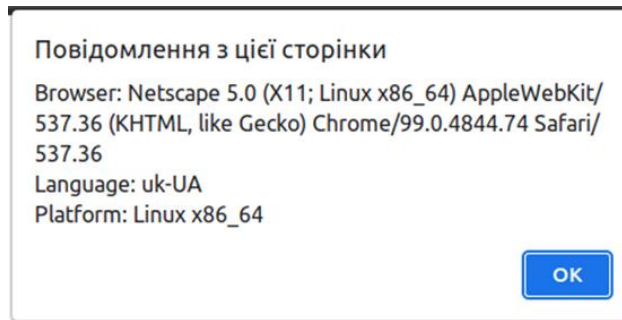


Рис. 1.2. Використання JavaScript

Цей код створює змінну `browserInfo`, яка містить інформацію про браузер, мову та платформу, на якій запущено скрипт. Для отримання цієї інформації використовується об'єкт `navigator`, який містить різні властивості, що описують характеристики браузера та операційної системи.

У першому рядку коду використовується рядкова шаблонна літералка (backtick), щоб створити рядок `browserInfo`. У цьому рядку використовуються рядкові змінні `${navigator.appName}`, `${navigator.appVersion}`, `${navigator.language}` та `${navigator.platform}`, які містять відповідну інформацію про браузер та операційну систему. У другому рядку коду використовується функція `alert()`, яка виводить діалогове вікно зі значенням змінної `browserInfo`.

Отже, коли цей код виконується в браузері, він виводить інформацію про браузер, мову та платформу на якій запущено скрипт у вікні спливаючого повідомлення.

Таким чином, JavaScript — це потужна мова програмування, яка дозволяє розробникам створювати динамічні веб-сайти та інтерактивні програми, які працюють на різних платформах. Такі функції, як взаємодія HTML і CSS, об'єктно-орієнтований підхід і функції високого рівня, дозволяють розробникам створювати складні та гнучкі програми.

HTML5

HTML5 (Hypertext Markup Language, version 5) є останньою версією стандарту мови розмітки гіпертексту, що використовується для створення веб-сторінок та додатків. HTML5 був прийнятий як рекомендація World Wide Web Consortium у жовтні 2014 року і замінив попередню версію HTML 4.01.

Основне призначення HTML5 полягає в тому, щоб забезпечити веб-розробникам більш розширені можливості в розробці динамічних та багатофункціональних веб-додатків, що працюють на різних пристроях. HTML5 надає нові теги та атрибути, що забезпечують підтримку відео та аудіо, графіки, анімації, геолокації, локального сховища, офлайн-роботи, мультимедійних ефектів та інших функцій.

Основні особливості HTML5 полягають у:

- Підтримка відео та аудіо без додаткових плагінів: HTML5 надає можливість вставляти відео та аудіо без додаткових плагінів, таких як Adobe Flash. Це покращує продуктивність і безпеку;

- Нові теги та атрибути: HTML5 містить нові теги й атрибути, які розширюють функціональність веб-сторінок;

- Покращена підтримка графіки: HTML5 має кращу підтримку графіки, що дозволяє створювати більш складні та динамічні ефекти;

- Покращена підтримка мультимедіа: HTML5 забезпечує не тільки підтримку відео та аудіо, але й можливість маніпулювати мультимедійними ефектами, такими як тіні та переходи;

- Підтримка розташування: HTML5 дозволяє веб-додаткам отримувати місцезнаходження користувача за допомогою GPS або іншими методами. Для прикладу розглянуто приклад використання HTML:

```
<!DOCTYPE html>
<html>
<head>
  <title>Приклад HTML5 - Шостаківський</title>
</head>
<body>
  <h1>Привіт, світ!</h1>
  <p>Це мій простий приклад документу HTML5.</p>
  
</ul>
```

```
<li>Елемент списку 1</li>
<li>Елемент списку 2</li>
<li>Елемент списку 3</li>
</ul>
<a href="https://www.example.com">Посилання на приклад</a>
</body>
</html>
```

Аналізуючи даний код варто виділити основні елементи:

- `<!DOCTYPE html>`: Це декларація типу документа і вказує, що ми використовуємо HTML5.
- `<html>`: Цей елемент є корневим елементом документа HTML.
- `<head>`: Цей елемент містить метадані документа, такі як заголовок вікна браузера, підключення зовнішніх файлів CSS або JavaScript, метатеги та інші налаштування.
- `<title>`: Цей елемент встановлює заголовок документа, який відображається в рядку заголовка вікна браузера або в закладці.
- `<body>`: Цей елемент містить весь вміст веб-сторінки, який буде відображений у вікні браузера.
- `<h1>`: Цей елемент визначає заголовок першого рівня. У даному випадку він відобразить текст «Привіт, світ!».
- `<p>`: Цей елемент визначає абзац. У даному прикладі він містить текст «Це мій перший документ HTML5.».
- ``: Цей елемент вставляє зображення на сторінку. Атрибут `src` вказує шлях до зображення «`image.jpg`», атрибут `alt` встановлює альтернативний текст, який буде відобразитися, якщо зображення не завантажеться або недоступне.
- ``: Цей елемент визначає нумерований список. У даному прикладі він містить три елементи списку ``, які відобразяться як пункти списку.

– ``: Ці елементи визначають пункти списку. У даному прикладі ми маємо три `` елементи, які відображаються як окремі пункти списку. Текст вкладений всередину кожного `` елемента відображається як зміст пункту списку.

– `<a>`: Цей елемент створює посилання на іншу веб-сторінку або ресурс. Атрибут `href` вказує цільовий URL, до якого буде вести посилання. У даному прикладі посилання спрямовує на `<https://www.example.com>`. Текст всередині `<a>` елемента відображається як текст посилання.

Цей простий приклад коду демонструє основні елементи HTML5 для створення веб-сторінки. Він містить заголовок, параграф, зображення, нумерований список та посилання (рис. 1.3).

Привіт, світ!

Це мій простий приклад документу HTML5.



- Елемент списку 1
- Елемент списку 2
- Елемент списку 3

[Посилання на приклад](#)

Рис. 1.3. Використання HTML5

Таким чином, HTML5 пропонує веб-розробникам більш гнучкі та розширені можливості для розробки крос-пристроїв веб-додатків із більш складною функціональністю. Це дозволяє підвищити продуктивність і ефективність ваших веб-додатків і забезпечити кращий досвід для ваших користувачів.

SQL

SQL (Structured Query Language) - це мова запитів до баз даних, що дозволяє зберігати, змінювати та отримувати дані зі структурованих джерел. SQL є стандартом в області управління базами даних та використовується в багатьох системах управління базами даних, таких як MySQL, Oracle, Microsoft SQL Server, PostgreSQL та інші.

SQL дозволяє створювати таблиці для зберігання даних, виконувати запити для вибірки, оновлення, видалення та вставки даних, створювати та модифікувати індекси та забезпечувати безпеку даних з допомогою механізму автентифікації та авторизації користувачів.

Особливості SQL:

– Декларативний характер: SQL є декларативною мовою, що означає, що користувач описує, що потрібно зробити, а не як це зробити. Наприклад, для вибірки даних з бази даних, користувач описує, які дані він хоче отримати, а не як вони повинні бути вибрані.

– Реляційна структура даних: SQL працює зі структурованими даними у вигляді таблиць з реляційними зв'язками між ними. Кожна таблиця складається зі стовпців та рядків, де кожен запис містить значення для кожного атрибута.

– Розширені можливості: SQL дозволяє виконувати запити до даних, що об'єднують таблиці, вибірки з декількох джерел, агрегування даних, фільтрацію, сортування, обмеження результатів запиту, вставку, оновлення та видалення даних.

– Можливості оптимізації: SQL має можливості оптимізації запитів, що дозволяє покращити продуктивність запитів та зменшити навантаження на базу даних.

Приклад коду на SQL для створення таблиці користувачів та додавання декількох записів (рис. 1.4):

```

-- Створюємо таблицю користувачів
CREATE TABLE users (
  id INT PRIMARY KEY,
  name VARCHAR(50),
  email VARCHAR(50)
);
-- Додаємо записи в таблицю
INSERT INTO users (id, name, email)
VALUES
  (1, 'Антон Дмитренко', 'dmytrenko@example.com'),
  (2, 'Валерія Шульга', 'shulga@example.com'),
  (3, 'Володимир Федін', 'fedin@example.com');
-- Переглядаємо результат
SELECT * FROM users;

```

The screenshot shows a MySQL IDE interface. On the left, there is a code editor with the following SQL code:

```

1 -- Створюємо таблицю користувачів
2 CREATE TABLE users (
3   id INT PRIMARY KEY,
4   name VARCHAR(50),
5   email VARCHAR(50)
6 );
7 -- Додаємо записи в таблицю
8 INSERT INTO users (id, name, email)
9 VALUES
10  (1, 'Антон Дмитренко', 'dmytrenko@example.com'),
11  (2, 'Валерія Шульга', 'shulga@example.com'),
12  (3, 'Володимир Федін', 'fedin@example.com');
13 -- Переглядаємо результат
14 SELECT * FROM users;

```

On the right side, there is an 'Output' window showing the result of the SQL execution:

id	name	email
1	Антон Дмитренко	dmytrenko@example.com
2	Валерія Шульга	shulga@example.com
3	Володимир Федін	fedin@example.com

Below the table, it says: [Execution complete with exit code 0]

Рис. 1.4. Використання SQL

Підсумовуючи, варто зауважити, що SQL є мовою запитів до баз даних, яка дозволяє зберігати, змінювати та отримувати дані зі структурованих джерел. SQL є декларативним і має розширені функції для обробки структурованих даних у формі таблиць із реляційними зв'язками між ними, запитів даних із різних джерел і виконання оптимізації запитів для підвищення продуктивності. SQL є стандартом керування базами даних і використовується багатьма системами керування базами даних.

1.2.3. Бази даних

База даних — це структурований набір даних, що зберігається в комп'ютерній системі, з якого дані можна швидко й ефективно отримувати, оновлювати, видаляти та вставляти. Бази даних використовуються для зберігання великих обсягів даних і забезпечення доступу до цих даних із різних джерел. База даних може зберігати різноманітні дані. В. Текстові документи, зображення, відео, аудіо тощо.

MySQL

MySQL - це безкоштовна система керування базами даних з відкритим вихідним кодом, яка дозволяє зберігати, оновлювати та запитувати великі обсяги даних. MySQL створена для використання в сучасних веб-додатках, де необхідно зберігати велику кількість даних, та для інших застосувань, де вимоги до продуктивності дуже високі.

Основні особливості MySQL включають:

Висока продуктивність: MySQL забезпечує швидкий доступ до даних за рахунок ефективної організації даних в пам'яті та на диску, оптимізації запитів та інших методів. Масштабованість: MySQL може працювати з великими обсягами даних і може бути розміщений на декількох серверах.

- Безпека: MySQL має вбудовану систему автентифікації та авторизації, що дозволяє забезпечити безпеку даних;
- Простота використання: MySQL має простий і зрозумілий інтерфейс для розробників і адміністраторів;
- Функціональність: MySQL має широкий спектр функцій, включаючи можливість зберігати дані в різних форматах, підтримку транзакцій і запитів для отримання даних;
- Підтримка: MySQL має велику спільноту користувачів і розробників, що забезпечує постійний розвиток системи та підтримку користувачів.

MySQL є однією з найпопулярніших систем керування базами даних, яка використовується багатьма веб-додатками, такими як WordPress, Facebook, Twitter і YouTube.

MongoDB

MongoDB — це документно-орієнтована база даних, яка зберігає дані у вигляді документів у форматі JSON (JavaScript Object Notation). MongoDB зазвичай використовується для зберігання великих обсягів даних із швидким доступом до даних і масштабованістю.

Ключові особливості MongoDB:

1. Гнучкість: MongoDB не вимагає суворої схеми даних. Будь-які дані можна зберігати в будь-якому форматі, що полегшує розробку та масштабування додатків.

2. Швидкість: MongoDB має кілька механізмів для прискорення доступу до даних, включаючи індекси та кешування. Більшість операцій з базою даних можна виконувати в реальному часі.

3. Масштабованість: MongoDB розроблено для легкого масштабування. Нові сервери зберігання можна додати, щоб збільшити потужність бази даних без значних витрат на розробку або реконструкцію.

4. Висока доступність: MongoDB може копіювати дані, тож ви можете зберігати копії своїх даних на кількох серверах.

5. Гнучкість запитів: MongoDB має потужну мову запитів, яка дозволяє виконувати складні запити до бази даних.

MongoDB використовується в різних галузях, включаючи фінанси, телекомунікації, роздрібну торгівлю та онлайн-рекламу. Підходить для веб-додатків, які вимагають великих обсягів даних, швидкості та гнучкості. MongoDB підтримує кілька платформ, таких як Windows, Linux і Mac OS, і має велику спільноту розробників і користувачів, які можуть отримати підтримку та знайти відповіді на ваші запитання.

PostgreSQL

PostgreSQL — це високопродуктивна об'єктно-реляційна система керування базами даних з відкритим кодом. Він включає високий рівень надійності, масштабованості та розширюваності, і може використовуватися як для малих, так і для дуже великих баз даних.

Основна мета PostgreSQL — зберігати та впорядковувати дані, що відрізняє його від традиційних систем зберігання файлів. PostgreSQL надає функції для збереження, оновлення, видалення та отримання даних у базі даних. Він підтримує різні типи даних, такі як текст, числа, дати, геодані, XML і JSON.

Особливості PostgreSQL включають:

1. Транзакційна безпека: PostgreSQL забезпечує можливість робити зміни в базі даних в рамках транзакцій, що гарантує їх безпеку та відмінність;
2. Підтримка стандартів: PostgreSQL підтримує багато стандартів, такі як SQL, ACID та OLAP;
3. Масштабованість: PostgreSQL може бути налаштований для роботи зі збільшенням кількості даних та користувачів;
4. Розширюваність: PostgreSQL може бути розширений за допомогою плагінів та власних функцій;
5. Гнучкість: PostgreSQL надає можливість налаштування різних параметрів, що дозволяє оптимізувати його роботу для різних вимог.

У загальному, PostgreSQL є потужною та надійною системою управління базами даних, яка підходить для різних проектів від невеликих веб-сайтів до великих корпоративних додатків.

1.2.4. Редактори тексту та коду

Незалежно від того, чи потрібно занотувати, писати код чи розмітку, хороший текстовий редактор є частиною повсякденного життя.

PhpStorm

PhpStorm — це інтегроване середовище розробки (IDE) для мови програмування PHP, розроблене JetBrains. PhpStorm пропонує широкий спектр інструментів для розробки програмного забезпечення PHP, включаючи: В. Підтримка синтаксису, автозаповнення, рефакторинг, налагодження, тестування тощо (рис. 1.5).

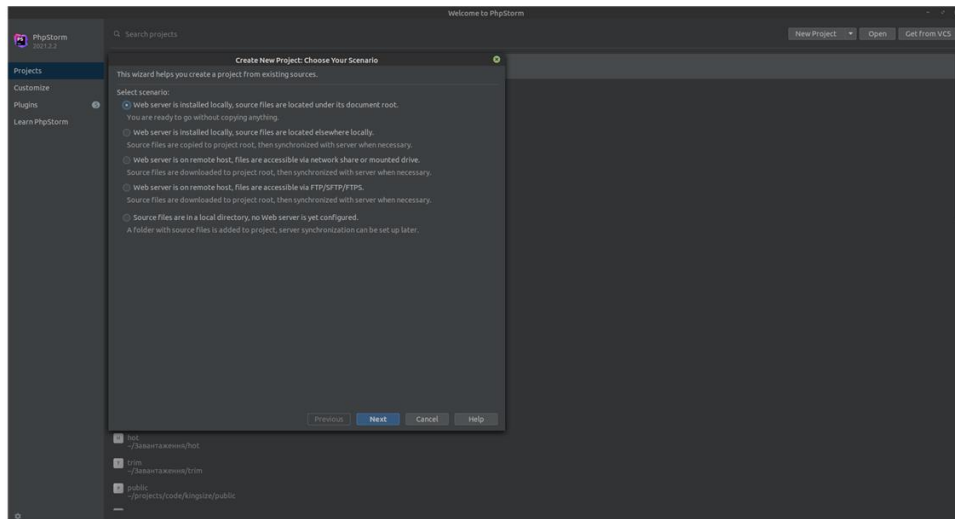


Рис. 1.5. Програма PhpStorm

Основні особливості PhpStorm:

1. Підтримка багатьох фреймворків, таких як Symfony, Laravel, Yii, Zend тощо, забезпечує зручну розробку на основі цих фреймворків.
2. Автозавершення, яке дозволяє швидше писати код і зменшує кількість помилок, пов'язаних із неправильним синтаксисом.
3. Рефакторинг, який дозволяє ефективно переписувати код, зменшувати дублювання коду, витягувати функції в окремі методи, витягувати інтерфейси та класи, перейменовувати змінні та методи тощо.
4. Підтримка тестування, яка полегшує створення та виконання тестів для коду PHP.
5. Налаштовувач, який дозволяє легко налагоджувати код PHP і відстежувати значення змінних у реальному часі.
6. Підтримка контролю версій полегшує роботу з такими сховищами, як Git, SVN і Mercurial.
7. Підтримка інтеграції з іншими інструментами розробки, такими як Docker, Vagrant, Composer.

PhpStorm використовується для розробки веб-додатків, веб-сайтів, мобільних додатків, програмного забезпечення для обробки даних і багатьох інших завдань. Це допомагає розробникам PHP створювати високоякісний код, зменшує загальну кількість помилок і підвищує продуктивність

розробника. Потужний інструмент розробки програмного забезпечення PHP, PhpStorm пропонує широкий спектр функцій і інструментів, які допомагають розробникам створювати чистий, структурований і ефективний код. Його інтуїтивно зрозумілий інтерфейс і кросфреймворкова підтримка інтегруються зі стандартними інструментами розробки, такими як Git, Docker, Vagrant і Composer. PhpStorm допомагає розробникам працювати швидше та з меншою кількістю помилок. Це зменшує час, витрачений на налагодження та тестування коду, і покращує якість кінцевого продукту.

Visual Studio Code

Visual Studio Code — це безкоштовний, відкритий і кросплатформенний редактор коду, розроблений Microsoft. Він дуже популярний серед програмістів і має багато корисних функцій, які дозволяють ефективно працювати з кодом.

Основна мета VS Code — дозволити програмістам писати, редагувати та налагоджувати код, який використовується для створення веб-додатків, мобільних додатків та інших додатків.

Особливості Visual Studio Code:

1. Підтримка кількох мов програмування, таких як JavaScript, Python, PHP, C++, Java тощо;
2. Зручний інтерфейс з багатьма інструментами для роботи з кодом, такими як автодоповнення, підсвічування синтаксису, згорання коду тощо;
3. Можливість налагодження коду за допомогою вбудованого відладчика; Підтримка плагінів, що дозволяють розширити функціональність редактора;
4. Інтеграція з системами контролю версій, такими як Git;
5. Підтримка використання контейнерів Docker;
6. Можливість задавати різні теми оформлення для зручності користувача.

Крім того, VS Code є легким та швидким редактором, який працює на багатьох платформах, включаючи Windows, macOS та Linux, що робить його дуже популярним серед програмістів у всьому світі.

Atom

Atom — це безкоштовний редактор із відкритим кодом, створений GitHub для підтримки розробки програмного забезпечення. Atom підтримує багато мов програмування, таких як JavaScript, HTML, CSS, Node.js, і може використовуватися як для створення веб-сайтів, так і для розробки додатків.

Основні можливості редактора Atom:

1. Модульність: Atom побудовано на модульній архітектурі, яка дозволяє розширити його функціональність за допомогою різних плагінів і програм.

2. Підсвічування синтаксису: Atom має підсвічування синтаксису, яке можна використовувати, щоб полегшити розуміння та редагування коду.

3. Автозаповнення: Atom підтримує автоматичне завершення коду, щоб зменшити кількість помилок і підвищити продуктивність.

4. Відстеження змін: Atom підтримує систему контролю версій Git, що дозволяє відстежувати зміни в коді та співпрацювати з іншими розробниками.

5. Підтримка плагінів: Atom має велику кількість плагінів, які дозволяють додавати нові функціональні можливості та розширювати функціональність редактора.

6. Інтеграція з іншими інструментами: Atom інтегрується з багатьма іншими інструментами, включаючи системи збірки, літери, налагоджувальники та багато інших.

Atom має також інтегровану підтримку для роботи з GitHub, що дозволяє розробникам працювати з репозиторіями, виконувати коміти, робити пули та багато іншого, не виходячи з редактора.

Крім того, Atom має вбудований редактор для відлагодження коду, що дозволяє розробникам легко відстежувати та виправляти помилки в коді. Atom доступний для різних операційних систем, включаючи Windows, macOS та

Linux, і має активну спільноту розробників, що допомагає підтримувати та розвивати редактор.

Загалом, Atom є потужним та гнучким редактором, який може бути використаний для розробки різноманітних проєктів, від веб-сайтів до додатків. Його модульна архітектура дозволяє розширювати його можливості за допомогою плагінів та додатків, що дозволяє розробникам налаштувати редактор до своїх потреб.

1.2.5. Веб-сервери

Веб-сервер, який в кінцевому підсумку використовується, зазвичай залежатиме від комбінації особистих уподобань, функціональності чи вже існуючої інфраструктури. Nginx і Apache є двома найпоширенішими веб-серверами, однак є й інші варіанти.

Nginx

Nginx - це веб-сервер та проксі-сервер з відкритим вихідним кодом, який зазвичай використовується для розгортання веб-додатків та обслуговування високої навантаженості. Однією з особливостей Nginx є його здатність ефективно обробляти багато одночасних з'єднань, що робить його популярним вибором для великих веб-сайтів та додатків з високим рівнем навантаження.

Деякі з основних функцій Nginx:

1. Веб-сервер: Nginx може виконувати функції веб-сервера, тобто обслуговувати веб-сторінки та інші статичні ресурси.
2. Проксі-сервер: Nginx може виконувати функції проксі-сервера, тобто пересилати запити на інші сервери та повертати відповіді клієнту.
3. Балансувальник навантаження: Nginx може взяти на себе функції балансувальника навантаження.
4. Захист від DDoS атак: Nginx має вбудований захист від атак DDoS, що робить його корисним для веб-додатків, які піддаються цим типам атак.

Більш детально Nginx, цей веб-сервер і проксі-сервер є одним із найпопулярніших рішень для розгортання веб-додатків і високого навантаження. Він дуже ефективний, може обробляти велику кількість одночасних підключень і мало ресурсомісткий.

Крім того, Nginx має вбудований захист від DDoS-атак і може діяти як балансир навантаження та проксі-сервер. Усі ці функції роблять Nginx обов'язковим інструментом для веб-розробників і системних адміністраторів.

Apache

Apache — це безкоштовний веб-сервер, який використовується для публікації веб-сайтів. Цей веб-сервер є одним із найпопулярніших веб-серверів у світі та використовується більшістю веб-сайтів, у тому числі такими популярними веб-сайтами, як Facebook, Twitter і Wikipedia.

Основне призначення Apache — створювати й обробляти веб-сторінки, надсилати їх користувачам за запитом і керувати даними на сервері. Ви можете обслуговувати різні типи веб-вмісту, наприклад HTML, CSS, JavaScript, зображення, аудіо- та відеофайли. Крім того, він захищає ваш сервер від зловмисних атак, таких як DDoS-атаки.

Apache характеризується своєю надійністю та багатими можливостями налаштування. Apache працює на різних операційних системах, таких як Windows, Linux і macOS, і підтримує різні мови програмування, такі як PHP, Python і Perl. Він також пропонує можливість встановлення додаткових модулів, які розширюють його функціональність.

Apache має велику спільноту користувачів, яка надає постійну підтримку та оновлення програмного забезпечення. Крім того, Apache має відкритий код, тому користувачі можуть безкоштовно змінювати вихідний код і використовувати його у своїх проектах. Загалом, Apache — це потужний і надійний веб-сервер, який можна налаштувати та підтримує різні мови програмування, щоб ви могли розширити його можливості.

ВИСНОВКИ ДО РОЗДІЛУ 1

В даному розділі розглянуто різноманітні підходи та інструменти, що використовуються в процесі розробки веб-застосунків. Підходи до розробки веб-застосунків включають каскадну модель, гнучку методологію, ітеративний та інкрементальний підхід. Кожен з цих підходів має свої переваги та недоліки і може бути використаний в залежності від вимог проекту та ресурсів, доступних для розробки.

Каскадна модель є класичним підходом, що передбачає послідовну роботу над кожним етапом розробки. Гнучка методологія дозволяє більш гнучко адаптуватися до змін у вимогах проекту і акцентує на співпраці замовника та самоорганізації команди розробників. Ітеративний та інкрементальний підходи передбачають поетапну розробку і впровадження функціональності, дозволяючи швидко отримувати результати та зворотний зв'язок від користувачів.

У розділі також розглянуті різні інструменти для розробки веб-застосунків. Менеджери пакетів та виконувачі завдань допомагають управляти залежностями та автоматизувати рутинні процеси розробки. Мови програмування та фреймворки використовуються для створення веб-застосунків. Бази даних грають важливу роль у зберіганні та обробці інформації. Редактори тексту та коду дозволяють розробникам зручно працювати з програмним кодом. Веб-сервери відповідають за обробку запитів користувачів та надання їм необхідної відповіді.

У процесі розробки веб-застосунків важливо враховувати особливості проекту, вимоги замовника та доступні ресурси. Вибір підходу та інструментів залежить від конкретної ситуації та може бути змінений у ході розробки. Правильне використання підходів та інструментів сприяє ефективній розробці веб-застосунків та досягненню поставлених цілей проекту.

РОЗДІЛ 2

ДИЗАЙН ТА FRONT-END ЧАСТИНА ВЕБ-ЗАСТОСУНКУ

2.1. Розробка дизайну

В якості середовища розробки макетів дизайну свого майбутнього інтернет-магазину мною було обрано досить популярний інструмент для дизайну інтерфейсу Figma. Це професійний інструмент розробки дизайну, який дозволяє створювати високоякісні макети інтерфейсів веб-сайтів, програм та інших цифрових продуктів.

Для розробки спочатку було проаналізовано взуттєвий ринок і в якості основних кольорів вибрано чорний і золотий. Ці кольори дуже популярні та стильні, ідеальні для брендів, які прагнуть привернути увагу своїх клієнтів.

Наступним кроком було визначення назви бренду. Мною було обрано назву «King Size». Назва бренду важлива, оскільки вона передає перше враження про ваш бренд і допомагає відрізнити вас від ваших конкурентів.

Визначившись з основними кольорами та назвою бренду, вирішено було створити та зареєструвати логотип компанії. Користуючись генератором логотипів <https://looka.com> обрано ключові елементи логотипу, кольори, та слоган «Step by step». Після цього штучний інтелект застосунку запропонував можливі варіанти логотипів (рис. 2.1).

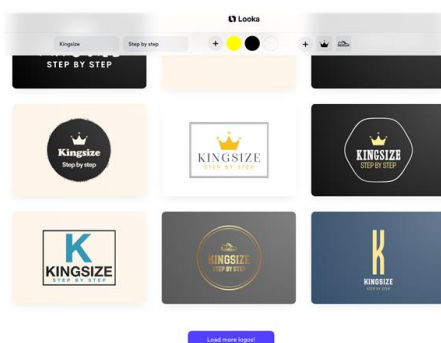


Рис. 2.1. Варіанти логотипу

Кафедра КІТ				НАУ 23 38 33 000 ПЗ			
	ПІБ			РОЗДІЛ 2. ДИЗАЙН ТА FRONT-END ЧАСТИНА ВЕБ- ЗАСТОСУНКУ	Літ.	Аркуш	Аркушів
Розроб.	Шостаківський О.О.					39	9
Керівник	Толстікова О.В.				ТП-415Б – 122		
Н. Контр.	Толстікова О.В.						

Мій вибір зупинився на одному з варіантів (рис. 2.2).



Рис. 2.2. Логотип бренду

Наступним етапом розробки дизайну мого майбутнього інтернет-магазину була розробка макету головної сторінки застосунку. Розробка макета в Figma дозволяє продумати кожну деталь і зробити інтерфейс інтуїтивно зрозумілим і простим у використанні. Для цього було використано функціональні елементи, такі як фільтри, рядки пошуку тощо у своєму макеті, щоб зробити пошук взуття максимально простим і швидким. Також важливо враховувати візуальну частину, таку як розміщення елементів на сторінці, колірні схеми та інші фактори, які впливають на естетику та сприйняття вашого сайту.

Аналізуючи сайти-аналоги та плануючи блоки, які повинні складати майбутню головну сторінку розроблено було таку структуру (рис. 2.3).



Рис. 2.3. Структура головної сторінки

Головну сторінку умовно можна розділити на 7 частин: header (шапку) сайту, візитівку, особливості, галерею, відгуки, новини та footer (підвал) сайту.

Дотримуючись кольорової гама було розроблено дизайн шапки сайту (рис. 2.4). Вона включає в себе: логотип бренду, в якому, в подальшому, буде закладено посилання на головну сторінку та навігаційне меню по застосунку. Меню в свою чергу міститиме посилання на основні блоки та сторінки майбутнього інтернет-магазину, а саме: головна, товари (Додаток В), про нас, доставка та оплата.



Рис. 2.4. Шапка сайту

Наступним блоком є візитівка сайту, в якій описано слоган та кредо компанії, а саме «Все починається, коли відкриваєш взуттєву коробку. Просто натисніть і замов». Також мною вирішено додати фільтр для пошуку необхідної моделі по всіх категоріях товарів. Дизайнерським рішенням було об'єднати header та візитівку в один візуальний блок у вигляді суцільної фігури (рис. 2.5).

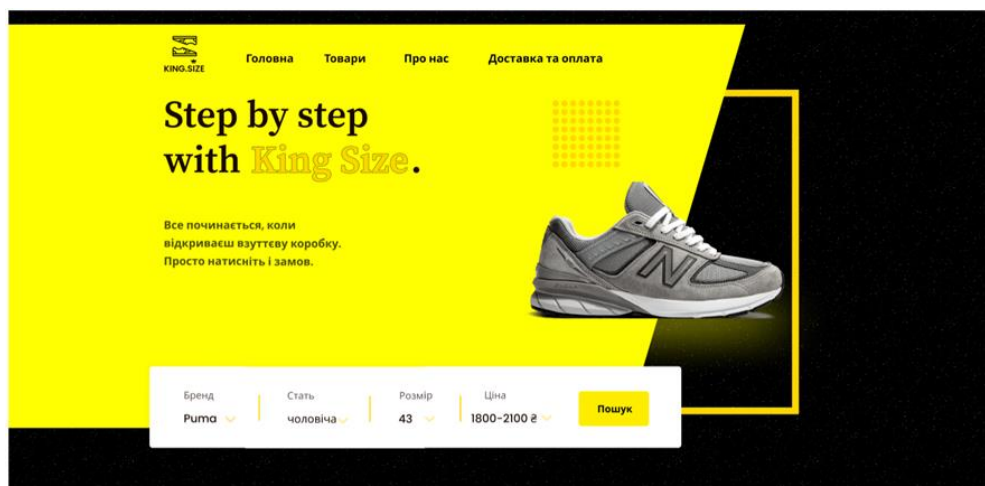


Рис. 2.5. Перший екран головної сторінки

Після візитівки розташовано блок особливостей компанії, якими вона виділяється серед конкурентів та пропонує користувачеві (рис. 2.6). До таких особливостей можна віднести: якість продукції, гнучкість замовлення та швидкість доставки, що дійсно цінується в інтернет-магазинах.

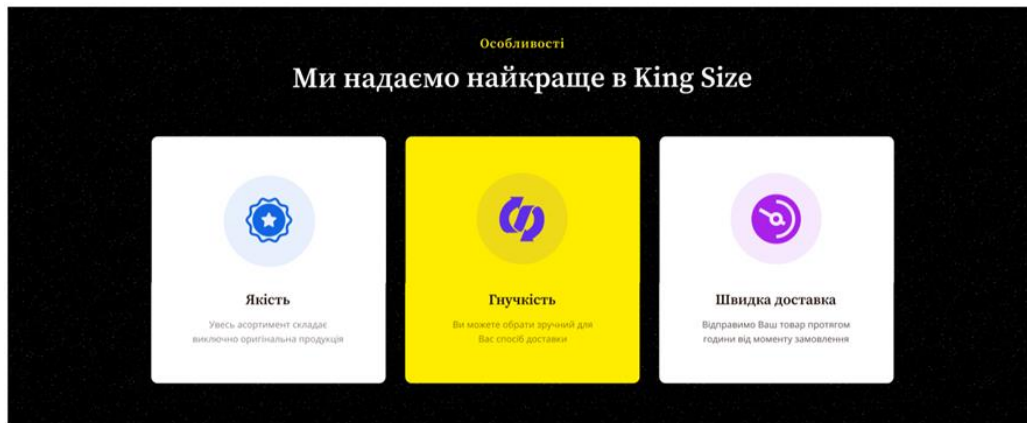


Рис. 2.6. Блок особливостей компанії

Ще одним ключовим блоком головної сторінки інтернет-магазину є галерея популярних товарів з асортименту. Саме такий блок реалізовано наступним. Він включає в себе навігацію по категоріям та слайдер з коротким описом товарів (Додаток А), де вказано назву, доступні розміри, ціну та фото товару (рис. 2.7).

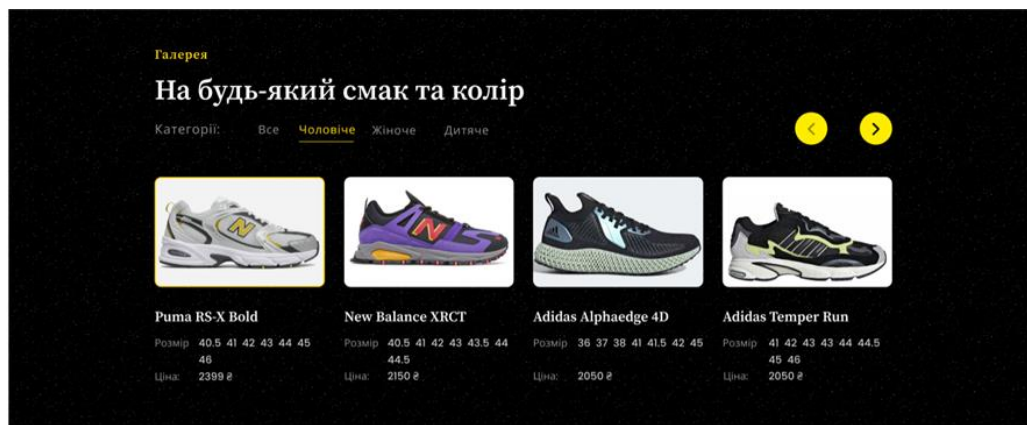


Рис. 2.7. Блок галереї товарів

Важко переоцінити важливість продемонструвати відгуки реальних покупців задля довіри майбутніх. З таких міркувань мною було розроблено блок відгуків, який являє собою слайдер з аватаркою клієнтів, їх іменами та, безпосередньо, відгуками (рис. 2.8).



Рис. 2.8. Блок відгуків покупців

Гарною практикою ведення онлайн бізнесу є збереження електронних адрес покупців задля подальшої роботи з ними. Таким чином можна в майбутньому повідомляти клієнтів про новинки товарів, акції та спеціальні пропозиції. Саме з таким міркувань створено блок, в якому покупець може вказати свій email (рис. 2.9).

Останнім блоком головної сторінки даного веб-застосунку інтернет-магазину взуття є footer (рис. 2.9). В ньому розташовано посилання на сторінку політики конфіденційності, де можна буде ознайомитися з умовами, нормами та правилами співпраці. Також вказано адресу контактної електронної пошти, по якій можна зв'язатися та посилання на соціальні мережі компанії.

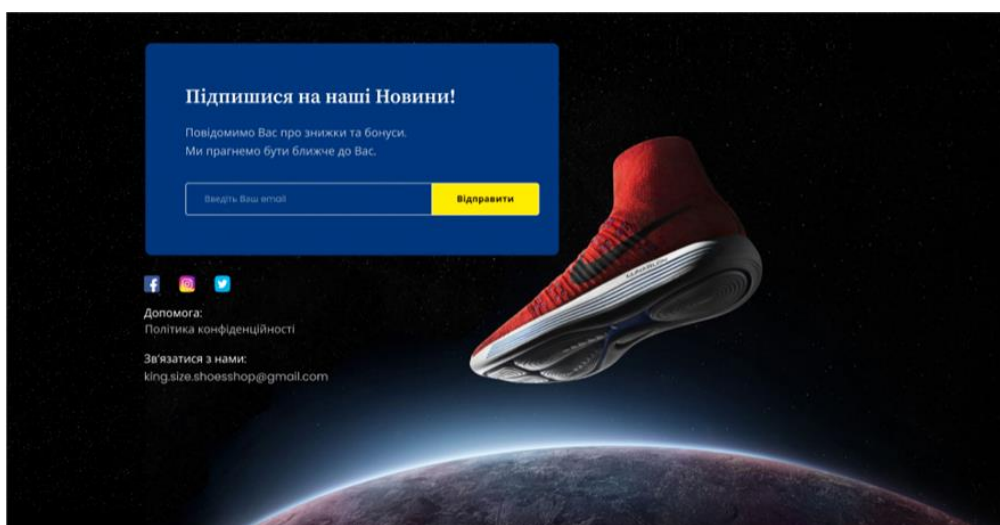


Рис. 2.9. Останній екран головної сторінки

2.2. Розробка front-end частини

Розробка front-end частини інтернет-магазину по готовим макетам - це процес створення веб-інтерфейсу магазину, що включає в себе візуальну структуру та поведінку веб-сторінок на клієнтській стороні.

Основні кроки процесу розробки front-end частини інтернет-магазину по готовим макетам можуть включати наступні етапи:

1. Розбивка на компоненти: Макети включають різні блоки та елементи, тому потрібно розбити макет на окремі компоненти для подальшого кодування (рис. 2.10).

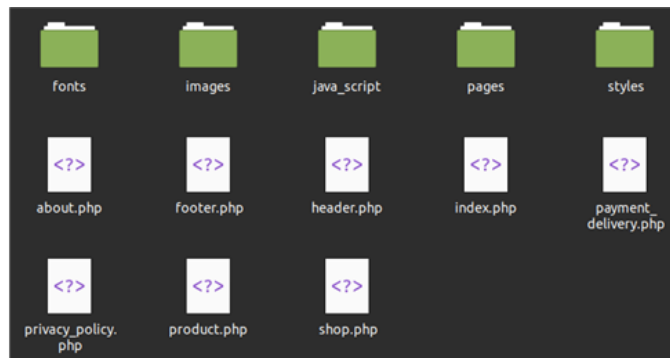


Рис. 2.10. Структура проекту

2. HTML-розмітка: На цьому етапі перекладається дизайн макету в розмітку HTML. Кожен компонент конвертується в HTML-код, з урахуванням структури сторінки та розташування елементів (рис. 2.11).

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6
7 <meta name="viewport" content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">
8 <meta http-equiv="X-UA-Compatible" content="ie=edge">
9 <title>King Size</title>
10 <link rel="shortcut icon" href="/images/favicon.ico" type="image/x-icon">
11 <link rel="stylesheet" href="/index_files/style.css">
12 <style></style>
13 <style></style>
14 <style></style><object class="chrome-extension://jffafkigfgmjafhpkoibhfeafeabmccg/" style="display: none; visibility: hidden;"></object>
15 </head>
16
17 <body>
18 <div class="main-cont content">
19
20 <header>
21 <div class="content header pc" id="head" style="padding-top: 30px; margin-top: -1px;">
22 <div class="menu" id="header" style="filter: none;">
23 <div class="logo"><a href="/index.php"></a></div>
24
25 <div class="menu-item"><a class="item" href="/shop.php?category_id=all&page=1&page=1&sort=date_up">Все</a></div>
26 <div class="menu-item"><a class="item" href="/shop.php?category_id=1&page=1&page=1&sort=date_up">Чоловіче взуття</a></div>
27 <div class="menu-item"><a class="item" href="/shop.php?category_id=2&page=1&page=1&sort=date_up">Жіноче взуття</a></div>
28 <div class="menu-item"><a class="item" href="/shop.php?category_id=3&page=1&page=1&sort=date_up">Одяг</a></div>
29 <div class="menu-item"><a class="item" href="/shop.php?category_id=4&page=1&page=1&sort=date_up">Акcesуари</a></div>
30 </div>
31 </div>
```

Рис. 2.11. Фрагмент HTML коду сторінки

3. CSS-стилізація: Слід застосувати CSS для стилізації HTML-коду. Встановити кольори, розміри, шрифти, вирівнювання, анімації та інші властивості, щоб забезпечити візуальну привабливість та відповідність макету (рис. 2.12).

```
index.html | style.css
body {
  color: #1A0A03;
  font-weight: 500;
  font-family: CenturyGothic;
  background-color: #000000;
  background-image: url(../images/back.png);
}

ul {
  list-style-type: none;
}

a {
  text-decoration: none;
}

.main-cont {
  width: 100%;
}

.container {
  width: 100%;
}

header {
  width: 100%;
  max-width: 1240px;
  position: fixed;
  z-index: 100000;
}

.content {
  max-width: 1240px;
  margin: 0 auto;
}

.header_mob{
  display: none;
}

#head{
  padding-top: 30px;
}

#header {
  float: left;
  width: 1189px;
  height: 100px;
  max-width: 1240px;
  padding-right: 80px;
  background-size: 100%;
  background-image: url('../images/header.png');
}
```

Рис. 2.12. Фрагмент CSS коду сторінки

4. JavaScript-розробка: На цьому етапі додаються інтерактивність та функціональність до сторінок. Застосовуються JavaScript-бібліотеки та фреймворки для реалізації таких функцій, як валідація форм, динамічне завантаження контенту, слайдери тощо (рис. 2.13).

```
66     if ($("#models").length) {
67         tns({
68             container: '#models',
69             items: 4,
70             slideBy: "page",
71             autoplay: true,
72             nav: false,
73             touch: true,
74             controls: true,
75             controlsText: ["", ""],
76             autoplayTimeout: 7000,
77             autoplayButtonOutput: false,
78         });
79     }
80     if ($("#models-mob").length) {
81         tns({
82             container: '#models-mob',
83             items: 1,
84             slideBy: "page",
85             autoplay: true,
86             nav: true,
87             navAsThumbnails: true,
88             navContainer: "#models_mob_nav",
89             touch: true,
90             controls: true,
91             controlsText: ["", ""],
92             autoplayTimeout: 7000,
93             autoplayButtonOutput: false,
94         });
95     }

```

Рис. 2.13. Фрагмент JS коду оголошення слайдера

5. Тестування: Після завершення розробки front-end частини, проводиться тестування для перевірки правильності відображення, коректності роботи функціоналу та забезпечення зручного користування. Провівши аналіз інтерфейсу було прийнято додати іконку кошику замовлень (Додаток Б) та бургер-меню сторінок (рис. 2.14).

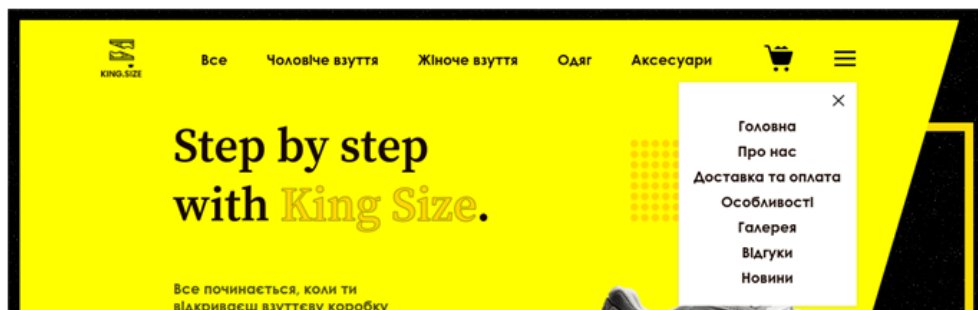


Рис. 2.14. Тестування front-end частини

ВИСНОВОК ДО РОЗДІЛУ 2

У даному розділі проведено детальний аналіз і виконана робота з розробки дизайну та front-end частини майбутнього інтернет-магазину. Для розробки дизайну був обраний інструмент Figma, який дозволив створити високоякісні макети інтерфейсу. Кольорова палітра на основі чорного та золотого кольорів була обрана з урахуванням популярності та контрасту. Назва бренду «King Size» була вибрана для передачі сильного першого враження та відмінності від конкурентів.

Розроблено та зареєстровано логотип компанії за допомогою генератора логотипів. Слоган «Step by step» був обраний для підкреслення послідовного розвитку. Розробка макету головної сторінки інтернет-магазину включала такі елементи, як шапка сайту, візитівка, особливості, галерея, відгуки, новини та підвал сайту. У підрозділі «Розробка front-end частини» було проведено розбивку на компоненти, HTML-розмітку, CSS-стилізацію та JavaScript-розробку. Ці етапи дозволили створити функціональну та естетичну front-end частину застосунку. Також було проведено тестування, щоб переконатися в коректному функціонуванні розробленої системи. Завдяки ретельному аналізу, професійному дизайну та розробці front-end частини, майбутній інтернет-магазин «King Size» буде мати привабливий та функціональний інтерфейс, що допоможе залучити та зберегти увагу клієнтів. Результати роботи в цьому розділі стануть основою для подальшого розвитку проекту, його імплементації та успішної роботи на ринку. Розробка дизайну та front-end частини відіграють важливу роль у створенні позитивного користувацького досвіду, викликаючи інтерес та задоволення у відвідувачів інтернет-магазину.

Отримані результати можуть бути використані як підґрунтя для подальших досліджень і розвитку в галузі розробки інтерфейсів та front-end рішень. Вироблені знання та навички дозволять впроваджувати сучасні стандарти та кращі практики у майбутніх проектах.

РОЗДІЛ 3

АРХІТЕКТУРА БАЗИ ДАНИХ ТА ВАСК-END ЧАСТИНА ВЕБ-ЗАСТОСУНКУ

3.1. Розробка архітектури бази даних

Серед різних варіантів для розробки бази даних інтернет-магазину «King Size» мною було обрано MySQL з кількох причин.

По-перше, MySQL є однією з найпопулярніших та надійних реляційних систем управління базами даних на сьогоднішній день. Ця система має довгу історію використання великими організаціями та підтримується широким співтовариством розробників, що забезпечує стабільність, безпеку та надійність.

По-друге, MySQL є відкритою системою з великим набором функціональності. Вона підтримує мови запитів SQL, які дозволяють зручно та ефективно маніпулювати даними. Також, MySQL має можливості для роботи з транзакціями, індексуванням, збереженими процедурами та іншими функціями, що роблять її ідеальним вибором для веб-додатків, таких як інтернет-магазин.

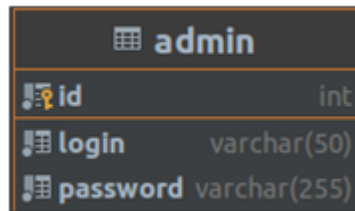
Першим етапом розробки бази даних є її створення. Для цього застосовано SQL запит: CREATE DATABASE kingsizeshoes.

Після створення бази даних kingsizeshoes, у ній будуть створені різні таблиці, такі як «admin» (адміністратори), «categories» (категорії), «news» (новини), «orders» (замовлення) та інші. Кожна таблиця буде мати свої стовпці для зберігання відповідних даних.

Таблиця admin відповідає за дані адміністраторів застосунку (рис. 3.1). Вона складається з стовпців id, що є первинним ключем та ідентифікатором

Кафедра КІТ				НАУ 23 38 33 000 ПЗ							
	<i>ПІБ</i>			РОЗДІЛ 3. АРХІТЕКТУРА БАЗИ ДАНИХ ТА ВАСК-END ЧАСТИНА			<i>Літ.</i>	<i>Аркуш</i>	<i>Аркушів</i>		
<i>Розроб.</i>	Шостаківський О.О.								48	14	
<i>Керівник</i>	Толстікова О.В.						ТП-415Б – 122				
<i>Н. Контр.</i>	Толстікова О.В.										

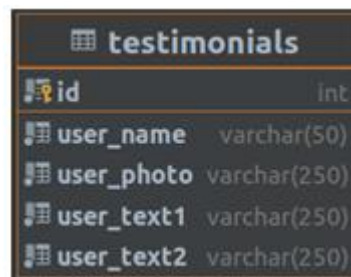
адміністратора, login та password, які відповідають за логін та пароль адміністратора відповідно.



admin	
id	int
login	varchar(50)
password	varchar(255)

Рис. 3.1. Таблиця admin

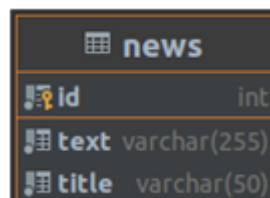
Таблиця testimonials відповідає за відгуки покупців (рис. 3.2). Вона складається зі стовпців id, що є первинним ключем та ідентифікатором відгуку, user_name — ім'я користувача, user_photo — аватар користувача, user_text1 — головна думка коментаря та user_text2 — основний текст коментаря.



testimonials	
id	int
user_name	varchar(50)
user_photo	varchar(250)
user_text1	varchar(250)
user_text2	varchar(250)

Рис. 3.2. Таблиця testimonials

Таблиця news відповідає за новини та розсилки листів покупцям, які залишили свій email на сайті (рис. 3.3). Дана таблиця складається зі стовпців id, що є первинним ключем та ідентифікатором листа, стовпець text відповідає за основний текст листа та стовпець title — це тема листа.



news	
id	int
text	varchar(255)
title	varchar(50)

Рис. 3.3. Таблиця news

У таблиці `users_email` зберігаються залишені користувачами застосунку електронні адреси (рис. 3.4). Її основою є два стовпці: `id_email`, що є первинним ключем та ідентифікатором запису та, безпосередньо, `email`.

users_email	
id_email	int
email	varchar(50)

Рис. 3.4. Таблиця `users_email`

Таблиця `categories` відповідає за категорії до яких відносяться продукти (рис. 3.5). Таблиця складається з стовпців `id`, що є первинним ключем та ідентифікатором категорії та `categor_name`, яка відповідає за назву категорії.

categories	
id	int
categor_name	varchar(15)

Рис. 3.5. Таблиця `categories`

Таблиця `products` є таблицею товарів (рис. 3.6). Дана таблиця містить поля, які характеризують продукт:

- `id` — аидентифікатор;
- `product_art` — артикль товару;
- `product_name` — ім'я товару;
- `product_price` — ціна;
- `product_img` — головне фото товару;
- `tags` — ключові слова(теги), які знадобляться для пошуку;
- `description` — опис товару.

products	
id	int
product_art	varchar(10)
product_name	varchar(250)
product_price	int
product_img	varchar(250)
tags	varchar(250)
description	varchar(750)

Рис. 3.6. Таблиця products

Також для зв'язку певних характеристик товару мною було прийнято рішення створити декілька зв'язуючих таблиць. Таких як:

categoryOfProduct — дана таблиця пов'язує таблицю товарів products та categories (рис. 3.7). Для цього використано зв'язок «один до багатьох».



Рис. 3.7. Таблиця categoryOfProduct

Ця таблиця для зв'язку використовує ключі category_id та product_id, що є відповідними ідентифікаторами записів з таблиць.

galleryProduct таблиця зв'язує товари з таблиці products та відповідні фото (рис. 3.8). Для цього використано зв'язок «багато до одного».

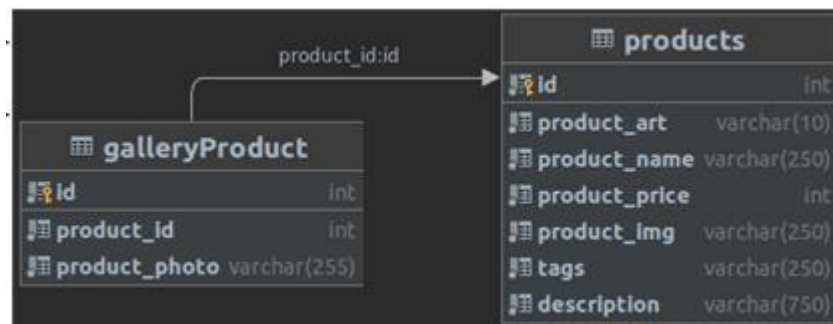


Рис. 3.8. Таблиця galleryProduct

Також однією з ключових таблиць є таблиця sizeProduct, яка, по аналогії з попередньою, товари з таблиці products та відповідні їх розміри в наявності (рис. 3.9). В таблиці також застосовується зв'язок «багато до одного».

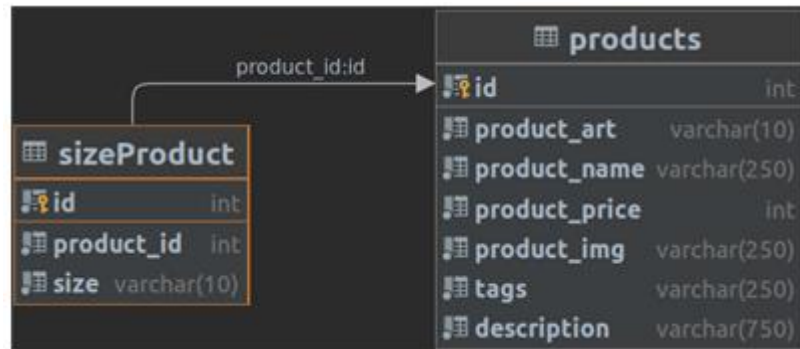


Рис. 3.9. Таблиця sizeProduct

Таблиця orders містить інформацію про замовлення та складається з наступних полів: розмір, спосіб оплати, спосіб доставки, номер телефону покупця, ПІБ покупця, адреса доставки, статус замовлення, ідентифікатор товару, який є зовнішнім ключем з таблиці products та дата замовлення (рис. 3.10).

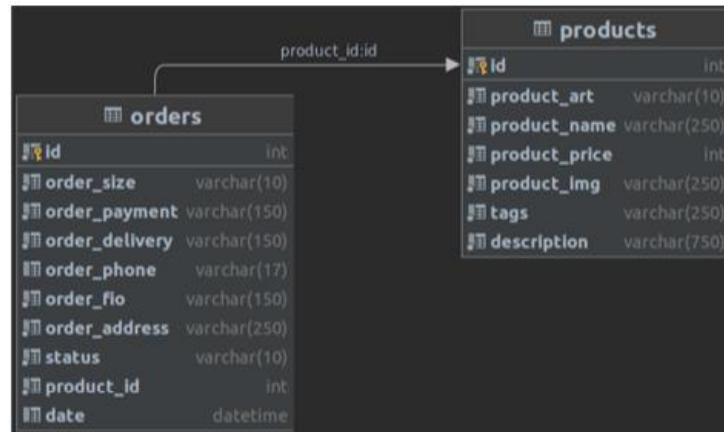


Рис. 3.10. Таблиця sizeProduct

Загальне дерево архітектури бази даних веб-застосунку інтернет-магазину взуття «King Size» матиме наступний вигляд (рис. 3.11):

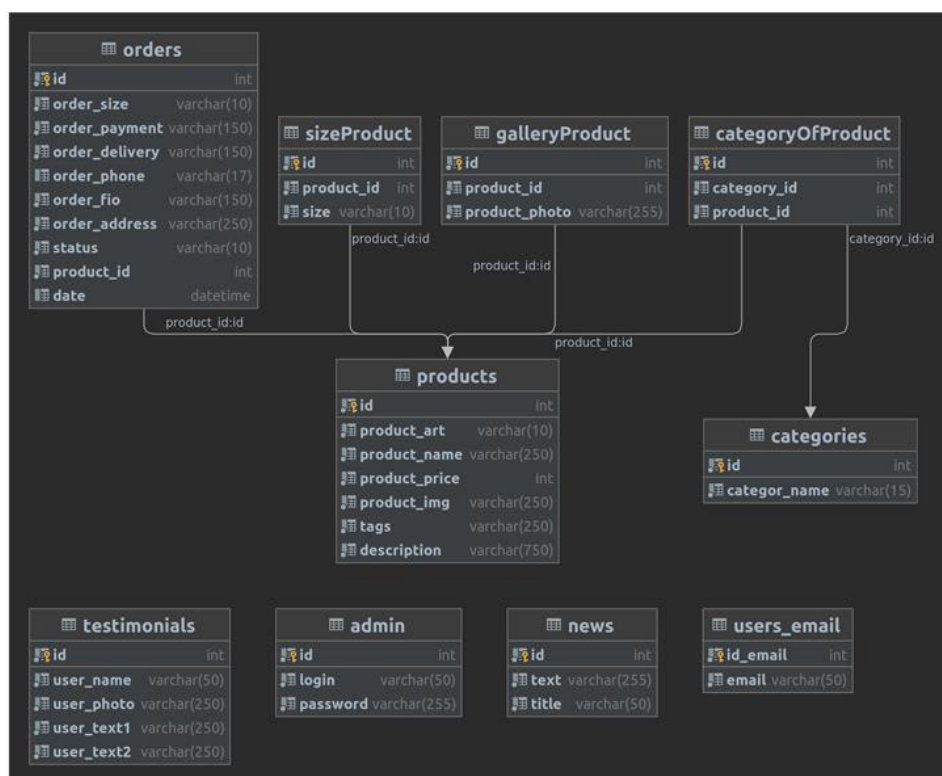


Рис. 3.11. Архітектура бази даних

3.2. Розробка back-end частини застосунку

Розробка back-end частини для інтернет-магазину взуття KingSize на PHP - це процес створення та налаштування серверної частини програмного забезпечення, яка забезпечує функціональність, логіку та обробку даних для магазину.

Back-end частина інтернет-магазину відповідає за обробку логіки додатку, збереження та взаємодію з даними, а також забезпечення безпеки та швидкодії системи. PHP - це мова програмування, яка широко використовується для розробки back-end частини веб-додатків.

3.2.1. Розробка back-end для користувацької частини застосунку

Одним з основних блоків головної сторінки кожного інтернет-магазину є блок галереї товарів. Розробка блоку каруселі товарів на PHP включає кілька етапів, включаючи налаштування серверної частини, отримання даних, генерацію HTML-коду.

Спочатку потрібно отримати дані про товари, які будуть відображатися в каруселі. Це можна зробити шляхом запиту до бази даних за допомогою SQL-запиту (рис. 3.12). Запит повинен вибрати потрібні дані, такі як назва товару, зображення, опис, ціна тощо.

```
<?php
global $connection;
$query = "SELECT * FROM `products`";
$products = $connection->query($query);
$productsMob = $connection->query($queryMob);
?>
```

Рис. 3.12. SQL-запит діставання товарів для галереї

Наступним кроком є генерація HTML-коду для відображення товарів у каруселі. Можна використовувати цикл для проходження через дані товарів та створення відповідних елементів HTML для кожного товару (рис. 3.13).

```
<div class="models-container">
  <div class="my-slider" id="models">
    <?php
    $a = 0;
    while ($product = $products->fetch_assoc()) {
      $idProduct = $product['id'];
      $sizeProduct = $connection->query(
        "SELECT * FROM `sizeProduct` WHERE product_id = '$idProduct'";
      );
      if ($a > 15) {
        break;
      } ?>
      <div class="galery-element">
        <a href="product.php?product_id=?= $product['id'] ??">
          
        </a>
        <div class="txt1">
          <a href="product.php?product_id=?= $product['id'] ??">=?= $product['product_name']; ?</a>
        </div>
        <div class="txt2" id="galery-size"><p>Розмір:</p>
        <?php
        while ($size = $sizeProduct->fetch_assoc()) {
          if ($size['size'] != null) {
            ?>
            <span>=?= $size['size'] ?</span>
            <?php
            } else { ?>
            <span class="txt2">Немає в наявності</span>
            <?php
            }
          } ?>
        </div>
        <div class="txt2" id="galery-price">
          Ціна:<span>=?= $product['product_price']; ?</span></div>
        </div>
      <?php
      $a++;
    }
  ?>
</div>
```

Рис. 3.13. Цикл виводу галереї товарів

Після генерації HTML-коду, його потрібно вивести на сторінку. Це можна зробити, вставляючи згенерований код у відповідне місце на сторінці,

наприклад, в <div> контейнер або в певний елемент зі сторінки, який буде служити контейнером для каруселі (рис. 3.14).

```
<div class="models-container">
  <div class="my-slider" id="models">
    <?php
      $a = 0;
      while ($product = $products->fetch_assoc()) {
        $idProduct = $product['id'];
        $sizeProduct = $connection->query(
          "SELECT * FROM 'sizeProduct' WHERE product_id = '$idProduct';"
        );
        if ($a > 15) {
          break;
        }
        <div class="galery-element">
          <a href="product.php?product_id=<?> $product['id'] ?>"></a>
          <div class="txt1"><a href="product.php?product_id=<?> $product['id'] ?>"><?> $product['product_name']; ?></a></div>
          <div class="txt2" id="galery-size"><p>Розмір:</p>
          <?php
            while ($size = $sizeProduct->fetch_assoc()) {
              if ($size['size'] != null) {
                ?>
                <span><?> $size['size'] ?></span>
              }
            }
          </div>
          <div class="txt2" id="galery-price">Ціна:<span><?> $product['product_price']; ?> </span></div>
        </div>
        <?php
          $a++;
        }
      }
    </div>
  </div>
```

Рис. 3.14. Код каруселі продуктів

Деталі даного рішення:

1. Підключення до бази даних відбувається в імпортованому файлі config.php (рис. 3.15). Код починається з оголошення глобальної змінної \$connection, яка вказує на підключення до бази даних. Попередньо, потрібно налаштувати з'єднання з базою даних, щоб отримати доступ до даних товарів.

```
<?php
define("DBHOST" , "localhost");
define("DBUSER" , "homestead");
define("DBPASS" , "secret");
define("DB" , "kingsizeshoes");

$connection = @mysqli_connect( DBHOST, DBUSER, DBPASS, DB) or die("No connection to BD");
mysqli_set_charset($connection , "utf8") or die("No charset");
?>
```

Рис. 3.15. Код підключення до бази даних kingsizeshoes

2. Далі отримано дані товарів за допомогою запиту SQL \$query вибирає всі записи з таблиці products. Результат запиту зберігається у змінну \$products за допомогою методу query().

3. Після чого слід відобразити карусель товарів. Відображення товарів відбувається всередині контейнера з класом "models-container". Карусель

створюється за допомогою елемента з класом "my-slider" і ідентифікатором "models".

4. Оскільки товар не один, слід використати цикл відображення товарів. Використовуючи цикл while, дані про кожен товар отримуються з результату запиту. Кожен товар відображається в блоках з класом "galery-element". В цьому блоку відображаються зображення товару, назва, розміри, ціна тощо.

5. Змінна \$a слугує лічильником, що обмежує кількість товарів, відображених у каруселі. Якщо \$a перевищує 15, цикл припиняється за допомогою break.

6. Для кожного товару виводяться його зображення, назва, розміри та ціна. Зображення та назва товару мають посилання на сторінку товару.

7. Для кожного товару виводяться розміри, що взяті з таблиці sizeProduct за допомогою запиту, який використовує product_id. Якщо розмір недоступний (\$size['size'] дорівнює null, то виводиться повідомлення "Немає в наявності". У протилежному випадку, розмір виводиться відповідно.

8. Ціна товару виводиться в блоку з класом "galery-price", разом із відповідним значенням з таблиці products.

В результаті отримано карусель з товарів, дані яких дістаються з бази даних (рис. 3.16).

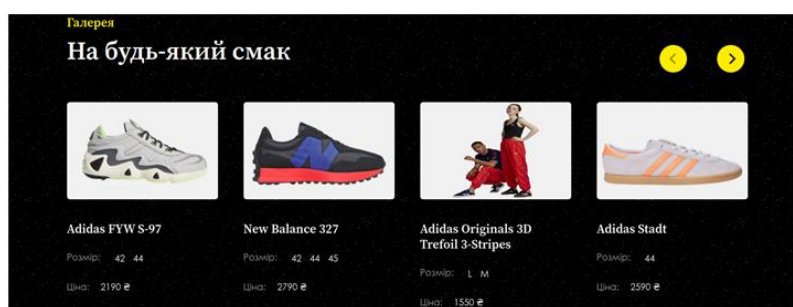


Рис. 3.16. Карусель продуктів

3.2.2. Розробка back-end для адміністративної частини застосунку

Адміністративна панель є незамінним інструментом для ефективного управління та функціонування інтернет-магазину. Вона виконує низку

важливих функцій, які допомагають власникам і адміністраторам магазину управляти та контролювати всі аспекти їх бізнесу.

Крім того, завдяки панелі є можливість додавати, редагувати та видаляти товари, встановлювати їх ціни, описи, зображення та інші характеристики. Це дозволяє легко оновлювати асортимент продуктів і контролювати їх доступність для покупців.

Панель адміністратора також допомагає відстежувати замовлення, обробляти платежі, керувати доставкою та відстежувати стан виконання замовлень. Вона надає зручний інтерфейс для управління процесом замовлення від початку до кінця.

Першим кроком створення адміністративної панелі інтернет-магазину є авторизація (рис. 3.17). Вона необхідна для обмеження кола доступу до даних та інструментів керування застосунку.

```
global $connection;
$adminTable = $connection->query("SELECT * FROM `admin`");
$admin = $adminTable->fetch_assoc();

$adminPass = $admin['password'];
if ($login == '' || $login != $admin['login']) {
    $_SESSION['error_login'] = "Некоректно введений логін!";
    redirect();
}elseif ($pass == '' || md5($pass) != $adminPass) {
    $_SESSION['error_pass'] = "Некоректно введений пароль!";
    redirect();
}else{
    $_SESSION['error_login'] = '';
    $_SESSION['error_pass'] = '';
    $_SESSION['login'] = true;
    header("Location: admin_orders.php?status=Активне");
    exit;
}
```

Рис. 3.17 Авторизація адміністратора

Наведений код демонструє простий процес авторизації адміністратора на PHP.

Перелік кроків реалізації:

1. Код починається з підключення до бази даних. Змінна `$connection` повинна містити об'єкт підключення до бази даних, щоб виконувати запити до таблиці `admin`.

2. Запит до таблиці `admin` отримує всі дані адміністратора. Це включає логін та хешований пароль.

3. Код перевіряє, чи введений логін не порожній і співпадає з логіном адміністратора з бази даних. Якщо логін некоректний, зберігається повідомлення про помилку у змінній `$_SESSION['error_login']`, і функція `redirect()` перенаправляє користувача на сторінку авторизації.

4. Код перевіряє, чи введений пароль не порожній і співпадає з хешованим паролем адміністратора з бази даних. Якщо пароль некоректний, зберігається повідомлення про помилку у змінній `$_SESSION['error_pass']`, і користувач знову перенаправляється на сторінку авторизації.

5. Якщо логін та пароль коректні, зберігаються необхідні дані в сесії, такі як `$_SESSION['login'] = true`, і користувач перенаправляється на сторінку керування замовленнями з параметром `status=Активне`. Функція `exit` зупиняє подальше виконання коду.

Після успішної авторизації адміністратор потрапляє на сторінку керування замовленнями у розділ активних (рис. 3.18). В шапці адміністративної панелі знаходяться посилання на головну сторінку сайту, сторінку замовлень, керування товарами (Додаток Г), листами (Додаток Г) та відгуками. Також передбачена кнопка виходу, при кліку на яку з сесії видаляється ключ авторизації. Крайнім елементом `header` є бургер-меню з основними сторінками сайту.

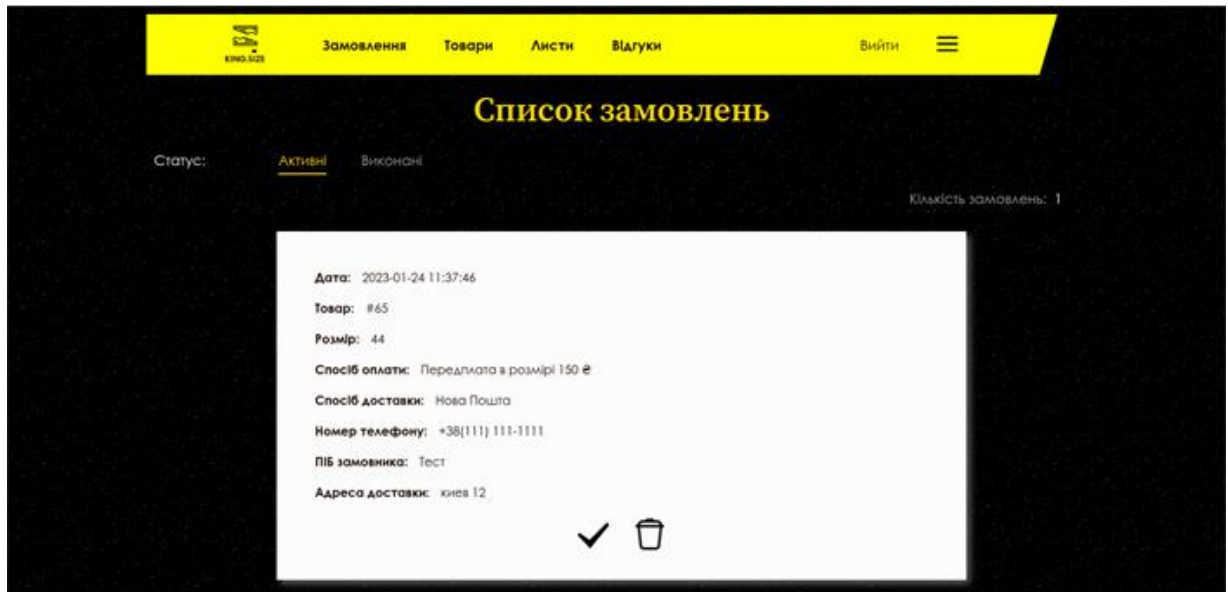


Рис. 3.18. Сторінка керування замовленнями

Код починається з ініціалізації змінної `$statusId`, яка визначає поточний статус замовлень, отриманий з параметра URL-адреси. Перевіряється також наявність значення `$statusId`. Якщо значення відсутнє, виконання коду припиняється (рис. 3.19).

```

<div class="orders-cont">
  <div class="main-text2">Список замовлень</div>
  <div class="categories">
    <?php
      global $connection;
      if (empty($_GET['status'])) {
        return;
      }
      $statusId = $_GET['status'];
      $activeAct = ($_GET['status'] == 'Активні' ? 'active_cat' : '');
      $activeDone = ($_GET['status'] == 'Виконані' ? 'active_cat' : '');
      $orders = $connection->query("SELECT * FROM `orders` WHERE status = '$statusId'");
    </?php>
    <div class="categories-txt">Статус:</div>
    <a class="categories-txt" <?=$activeAct?> href="/admin/admin_orders.php?status=Активні">Активні</a>
    <a class="categories-txt" <?=$activeDone?> href="/admin/admin_orders.php?status=Виконані">Виконані</a>
    <div class="categories-txt">Кількість замовлень:<span><?=$orders->num_rows?></span></div>
  </div>
  <?php
    while($order = $orders->fetch_assoc()){
      <?php
        <div class="itm">
          <div class="text">Дата:<span><?=$order['date']?></span></div>
          <div class="text">Товар:<span><?=$order['product_id']?></span></div>
          <div class="text">Розмір:<span><?=$order['order_size']?></span></div>
          <div class="text">Спосіб оплати:<span><?=$order['order_payment']?></span></div>
          <div class="text">Спосіб доставки:<span><?=$order['order_delivery']?></span></div>
          <?php
            if ($order['order_phone'] != '') {
              <div class="text">Номер телефону:<span><?=$order['order_phone']?></span></div>
              <?php }>
              <div class="text">ПІБ замовника:<span><?=$order['order_fio']?></span></div>
              <div class="text">Адреса доставки:<span><?=$order['order_address']?></span></div>
              <div class="done-order-cont">
                <a class="done-order" id="done" href="/admin/update_order.php?action=changeStatus&status=Виконані&id=?=$order['id']?></a>
                <a class="done-order" id="delete" href="/admin/delete_order.php?action=deleteOrder&id=?=$order['id']?></a>
              </div>
            </?php>
          </div>
        </?php>
      }
    }
  </?php>
</div>

```

Рис. 3.19. Код сторінки керування замовленнями

Запит до бази даних, використовуючи об'єкт `$connection`, призначений для отримання замовлень з відповідним статусом. Запит використовує значення `$statusId` для фільтрації результатів.

За допомогою циклу `while` перебираються рядки результатів запити і виводяться відповідні значення замовлень. Кожне замовлення відображається у своєму блоку `<div class="itm">`, де вказані різні деталі, такі як дата, товар, розмір, спосіб оплати, спосіб доставки тощо.

Для кожного замовлення додаються кнопки з посиланнями, що дозволяють змінити статус замовлення (`<a class="done-order" id="done" href="/admin/update_order.php?action=changeStatus&status=Виконане&id=<?=$order['id']?>">`) або видалити його (`<a class="done-order" id="delete" href="/admin/delete_order.php?action=deleteOrder&id=<?=$order['id']?>">`).

ВИСНОВОК ДО РОЗДІЛУ 3

Даний розділ кваліфікаційної роботи представляє значний внесок у створення функціональності та ефективності мого веб-застосунку інтернет-магазину взуття.

У першій частині наведено приклад розробки функціональності блоку галереї товарів на головній сторінці магазину. Основний етап розробки полягав у налаштуванні серверної частини, що включало в себе забезпечення взаємодії з базою даних та обробку запитів. Для отримання необхідних даних про товари використовувався SQL-запит до бази даних, який вибирав необхідні параметри, такі як назва товару, зображення, опис і ціна. Наступним кроком була генерація HTML-коду для відображення товарів у каруселі. Це досягалося за допомогою циклу, який перебирав дані товарів і створював відповідні елементи HTML для кожного товару. Нарешті, згенерований HTML-код був виведений на сторінку, шляхом вставки у відповідне місце на сторінці, наприклад, в контейнер `<div>` або інший визначений елемент.

У другій частині проведена розробка адміністративної панелі, яка є ключовим інструментом для ефективного управління та функціонування інтернет-магазину. Адміністративна панель надає власникам та адміністраторам магазину можливість додавати, редагувати та видаляти товари, встановлювати ціни, описи, зображення та інші характеристики. Мною була розроблена можливість створювати та розсилати покупцям електронні листи з новинками та пропозиціями. Також у адміністративній панелі передбачене створення та редагування відгуків покупців. На прикладі одного з основних елементів адміністративної панелі — авторизації, було розглянуто функціональну логіку. Авторизація виконується для обмеження доступу до даних та інструментів керування застосунком лише авторизованим користувачам, таким як власникам або адміністраторам магазину. Після успішної авторизації, користувач потрапляє на сторінку керування замовленнями у розділі активних.

ВИСНОВКИ

Під час виконання кваліфікаційній роботі було проведено дослідження та розробка веб-застосунку інтернет-магазину взуття з використанням PHP та адміністративною панеллю. Результати цієї роботи важливі для веб-розробників та компаній, які прагнуть створити ефективний та функціональний інтернет-магазин.

У першому розділі роботи розглядаються різні підходи до розробки веб-застосунків та їх методи. Зокрема, описується каскадна модель, гнучка методологія, ітеративний та інкрементальний підхід. Крім того, в цьому розділі розглядаються інструменти для розробки веб-застосунків, включаючи менеджери пакетів та виконувачі завдань, мови та платформи, бази даних, редактори коду та веб-сервери.

Загалом, у першому розділі висвітлюються різних підходи до розробки веб-застосунків та їх методи. Демонструється, що існує багато способів розробки веб-застосунків, і вибір підходу залежить від конкретних потреб проекту. Також в розділі підкреслено важливість використання відповідних інструментів для ефективної розробки веб-застосунків.

У другому розділі роботи детально розглядається розробка дизайну та front-end частини майбутнього веб-застосунку. Розробка дизайну включає визначення естетичного вигляду та композиції елементів інтерфейсу користувача. Завдяки цьому дослідженню, було ознайомлено з принципами дизайну, такими як колірна гама, шрифти, композиція, візуальні ефекти тощо, які сприяють створенню привабливого та зручного для використання інтерфейсу.

Також розглядається розробка front-end частини веб-застосунку, де використовуються HTML, CSS та JavaScript. HTML використано для створення структури та маркування елементів веб-сторінки. CSS дозволяє задавати стилі та вигляд елементів, що знаходяться на сторінці. А за допомогою JavaScript забезпечено інтерактивність та динаміку веб-застосунку.

Загальною метою цього розділу було створення привабливого та функціонального інтерфейсу, який забезпечує зручну взаємодію користувачів з веб-застосунком.

У третьому розділі розглядається розробка архітектури бази даних та back-end частини веб-застосунку. Розробка архітектури бази даних включає проектування та створення структури бази даних, таблиць, зв'язків між ними та визначення правил збереження даних. Важливо планувати та оптимізувати базу даних для ефективного збереження та отримання інформації.

Також розглядається розробка back-end частини веб-застосунку, яка включає реалізацію логіки та функціональності серверної частини. Це охоплює обробку запитів користувачів, валідацію даних, взаємодію з базою даних та інші операції, необхідні для правильної роботи веб-застосунку.

Цей розділ роботи зосереджений на розробці архітектури бази даних та back-end логіки веб-застосунку. Він підкреслює важливість правильного планування та оптимізації бази даних для збереження та отримання інформації. Також він наголошує на важливості розробки back-end функціональності та розробки адміністративної панелі для управління магазином.




Загальні висновки даної кваліфікаційної роботи підкреслюють важливість глибокого розуміння теоретичних аспектів розробки веб-застосунків та використання практичних інструментів та технологій для їх реалізації. Також підкреслюється значення дизайну та front-end розробки для створення зручного та привабливого користувацького інтерфейсу. Крім того, важливим етапом є архітектура бази даних та розробка back-end логіки, які визначають функціональність та ефективність інтернет-магазину. В цілому, ця робота надає цінну інформацію та рекомендації для розробки веб-застосунків, зокрема інтернет-магазинів, з використанням PHP та адміністративною панеллю.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. «Web Application Development Methodologies: A Comparative Study». Saroj Kaushik, International Journal of Advanced Research in Computer Science, Volume 10, No. 4, 2019.
2. Modern tools and current trends in web-development. Indonesian Journal of Electrical Engineering and Computer Science / Bhubaneswar, Gothapatha, India, 2021.
3. «Web Application Development Methodologies: A Comparative Review». Muhammad Shahid Nadeem, Sajjad Ahmad Baig, Imran Haider, International Journal of Computer Science and Network Security, Volume 19, No. 7, 2019.
4. [Електронний ресурс] Agile Alliance. Agile Manifesto. URL: <https://www.agilealliance.org/agile101/the-agile-manifesto>.
5. [Електронний ресурс] KeyCDN: Понад 100 чудових інструментів і ресурсів веб-розробки. URL: <https://www.keycdn.com/blog/web-development-tools>
6. Кузьмін, О. В. Розробка веб-додатків на основі HTML, CSS, JavaScript та PHP / О. В. Кузьмін, І. І. Кузьмін. - Київ: Видавничий дім «КМ-Букс», 2018.
7. «Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5», Robin Nixon, O'Reilly Media, 2018.
8. Вишневецький, В. Веб-дизайн: курс лекцій / В. Вишневецький. - Київ: Вид-во Національного університету «Києво-Могилянська академія», 2019.
9. [Електронний ресурс] UX Design. «Figma Review – A Design Tool For 2021». URL: <https://uxdesign.cc/figma-review-a-design-tool-for-2021-32e139fc1892>
10. [Електронний ресурс] Smashing Magazine. «Figma vs Sketch: Which One Is Better?» URL: <https://www.smashingmagazine.com/2019/09/figma-sketch-better/>

11. «Веб-розробка на основі HTML, CSS, JavaScript, PHP та MySQL» - Ірина Семенів, Олександр Семенів, Видавництво "Альтерпрес", 2018.
12. «Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics» by Jennifer Niederst Robbins, O'Reilly Media, 2018.
13. «PHP. Об'єктно-орієнтований підхід» - Олексій Скороход, Видавництво "Бібліотека програміста", 2020.
14. «Web-дизайн і розробка веб-сайтів» - Олег Комісаренко, Видавництво "Самміт-книга", 2021.
15. «Веб-дизайн і HTML-верстка» - Володимир Застосов, Видавництво "Клуб Сімейного Дозвілля", 2019.
16. «PHP Objects, Patterns, and Practice» by Matt Zandstra, Apress, 2018.
17. «JavaScript: The Definitive Guide: Activate Your Web Pages» by David Flanagan, O'Reilly Media, 2020.
18. «User Experience Design: Creating Designs Users Really Love» by Gavin Allanwood and Peter Beare, Bloomsbury Publishing, 2019.
19. «Responsive Web Design with HTML5 and CSS3: Build responsive and future-proof websites to meet the demands of modern web users» by Ben Frain, Packt Publishing, 2019.


ДОДАТОК А. КАРТКА ТОВАРУ

Все Чоловіче взуття Жіноче взуття Одяг Акcesуари  

New Balance 574

Артикул: ML574EGG

Ціна: 2890 €




Опис:

New Balance 574 - класичні бігові кросівки, розроблені на зорі становлення однієї з маловідомих американських субкультур. Вперше представлені наприкінці 80-х, 574 миттєво здобули популярність і досі залишаються ключовим силуетом у класичній лінійці New Balance. Про історію легендарного силуету, дизайн та різновиди кросівок New Balance 574 розповідає Снікер Енциклопедія мережі магазинів Street Beat.





Розміри в наявності:


43 44 45

Купити 

ДОДАТОК Б. КОШИК ЗАМОВЛЕННЯ

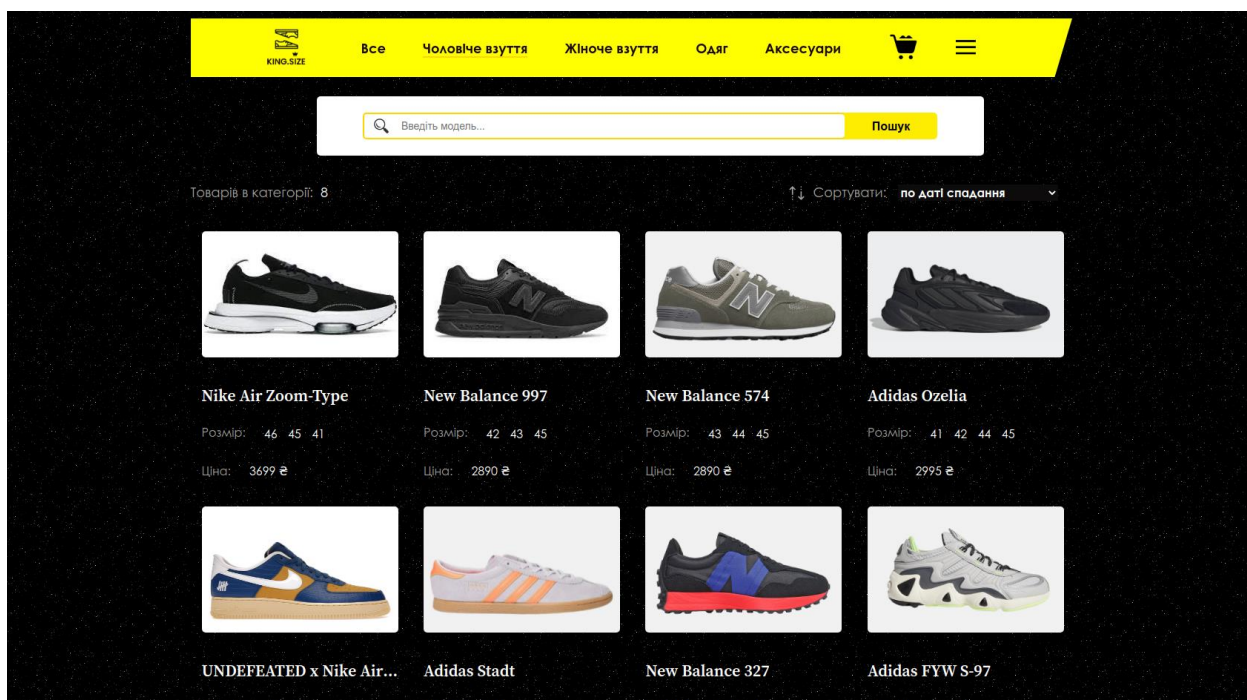
Обрані товари

Назва	Артикул	Розмір	Ціна	
 New Balance 574	art. ML574EGG	44	2890€	✗
 Adidas Originals 3D Trefoil 3-Stripes	art. GE6249	M	1550€	✗
 UNDEFEATED x Nike Air Force 1	art. DM8462-400	41	4175€	✗
 New Balance 327	art. MS327CPB	45	2790€	✗

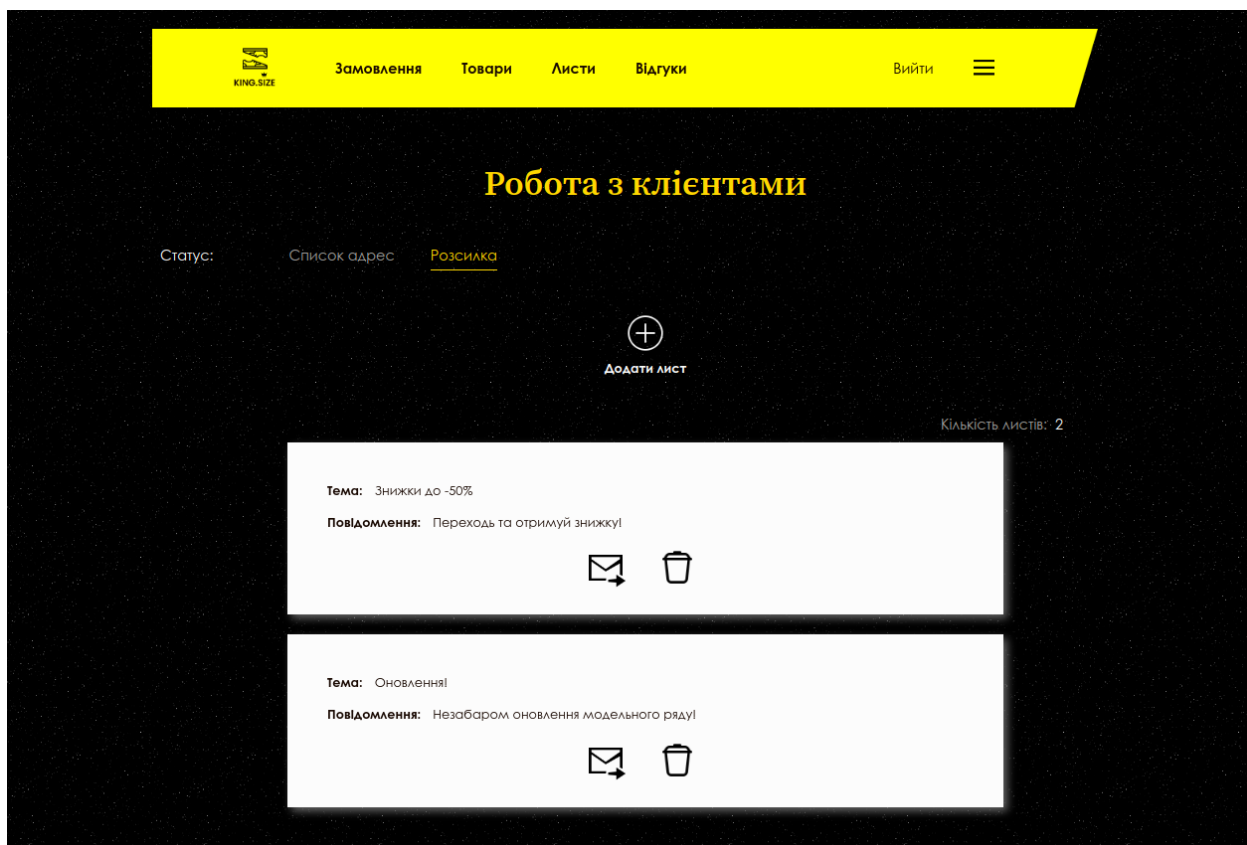
[До покупок](#) 

[Оформити замовлення](#)



ДОДАТОК В. КАТАЛОГ ТОВАРІВ



ДОДАТОК Г. КЕРУВАННЯ ЛИСТАМИ В АДМІНІСТАТИВНІЙ ПАНЕЛІ




ДОДАТОК Г. КЕРУВАННЯ ТОВАРАМИ В АДМІНІСТАТИВНІЙ ПАНЕЛІ

Замовлення Товари Листи Відгуки Вийти 

Робота з товарами

Розділ:


Товари


Додати товар


Пошук

Кількість товарів: 9

ID товару: #72
Артикул: CJ2033-001
Категорія: Жіноче взуття Чоловіче взуття
Назва: Nike Air Zoom-Type
Ціна: 3699 ₴
Розміри: 46 45 41
Головне фото:



Галерея:



Опис:
Чоловічі кросівки, що гарантують пружність під час носки та довговічність всіх матеріалів. Верх виконаний з дихаючої текстильної сітки. Верхня частина поєднує два види плетіння, а формовані вставки забезпечують чудовий комфорт. Отвори для шнурків укріплені еластичними накладками. Кросівки мають дуже широку підошву з амортизаційною технологією Zoom Air, у передній частині видно два балони, що надають пружність рухам. В основі Zoom Air лежить безліч міцних синтетичних волокон, затиснутих між пластиковими частинами, що забезпечують гнучку амортизацію.

Теги: піке найк айр айр айр Zoom-Type мужские чоловічі жіночі женские кроссовки кроси кросівки кожаныє шкіряні

