

Міністерство освіти і науки України  
Національний технічний університет  
«Дніпровська політехніка»

Інститут електроенергетики  
(інститут)

Факультет інформаційних технологій  
(факультет)

Кафедра Програмного забезпечення комп'ютерних систем  
(повна назва)

**ПОЯСНЮВАЛЬНА ЗАПИСКА**  
**кваліфікаційної роботи ступеня**  
**бакалавра**

(назва освітньо-кваліфікаційного рівня)

студентки *Попової Єлизавети Сергіївни*  
(ПІБ)

академічної групи *122-17-1*  
(шифр)

спеціальності *122 Комп'ютерні науки*  
(код і назва спеціальності)

освітньої програми *Комп'ютерні науки*  
(назва освітньої програми)

на тему: *Розробка кросбраузерної автоматизованої  
інформаційної системи електронної комерції*

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	<i>доц. Спирінцев В.В</i>			
<b>розділів:</b>				
спеціальний	<i>доц. Спирінцев В.В</i>			
економічний	<i>доц. Касьяненко Л.В.</i>			
<b>Рецензент</b>				
<b>Нормоконтролер</b>	<i>доц. Гуліна І.Г.</i>			

Дніпро  
2021

Міністерство освіти і науки України  
НТУ «Дніпровська політехніка»

**ЗАТВЕРДЖЕНО:**

завідувач кафедри  
програмного забезпечення комп'ютерних  
систем

(повна назва)

І.М. Удовик

(підпис)

(прізвище, ініціали)

«    »                      2021 року

**ЗАВДАННЯ**

**на кваліфікаційну роботу**

*бакалавра*

(назва освітньо-кваліфікаційного рівня)

студентки 122-17-1 Поповій Єлизаветі Сергіївні  
(група) (прізвище та ініціали)

тема кваліфікаційної роботи Розробка кросбраузерної  
автоматизованої інформаційної системи електронної комерції

затверджена наказом ректора НТУ «ДП» від 07.06.2021 р. № 317-с

Розділ	Зміст виконання	Термін виконання
<i>Спеціальний</i>	<i>На основі матеріалів виробничої практики та інших науково-технічних джерел провести аналіз стану рішення проблеми та постановку задачі. Обґрунтувати вибір та здійснити реалізацію методів вирішення проблеми</i>	<i>13.05.2021 р.</i>
<i>Економічний</i>	<i>Провести розрахунок трудомісткості розробки програмного забезпечення, витрат на створення ПЗ й тривалості його розробки</i>	<i>27.05.2021 р.</i>

Завдання видав \_\_\_\_\_ доц. Спирінцев В.В.  
(підпис) (посада, прізвище, ініціали)

Завдання прийняв до виконання \_\_\_\_\_ Попова Є.С.  
(підпис) (прізвище, ініціали)

Дата видачі завдання: 14.01.2021 р.

Термін подання кваліфікаційної роботи до ЕК: 11.06.2021 р.

## ЗМІСТ

РЕФЕРАТ.....	5
ABSTRACT.....	6
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	7
ВСТУП.....	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ	
1.1. Загальні відомості з предметної галузі.....	10
1.2. Призначення розробки та галузь застосування.....	14
1.3. Підстава для розробки.....	14
1.4. Постанова завдання.....	15
1.5. Вимоги до програми або програмного виробу.....	16
1.5.1. Вимоги до функціональних характеристик.....	16
1.5.2. Вимоги до інформаційної безпеки.....	18
1.5.3. Вимоги до складу та параметрів технічних засобів.....	22
1.5.4. Вимоги до інформаційної та програмної сумісності .....	22
РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ.	
2.1. Функціональне призначення системи.....	23
2.2. Опис застосованих математичних методів.....	26
2.3. Опис використаних технологій та мов програмування.....	26
2.4. Опис структури системи та алгоритмів її функціонування .....	40
2.5. Обґрунтування та організація вхідних та вихідних даних програми....	43
2.6. Опис розробленої системи .....	44
2.6.1. Використані технічні засоби.....	44
2.6.2. Використані програмні засоби.....	44
2.6.3. Виклик та завантаження програми.....	44
2.6.4. Опис інтерфейсу користувача.....	46
РОЗДІЛ 3. ЕКОНОМІЧНИЙ РОЗДІЛ.....	53
3.1. Розрахунок трудомісткості та вартості розробки програмного продукту.....	53

3.2. Рахунок витрат на створення програми.....	57
ВИСНОВКИ.....	59
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	61
Додаток А. Код програми.....	63
Додаток Б. Відгук керівника економічного розділу.....	88
Додаток В. Перелік файлів на диску.....	89

## РЕФЕРАТ

Пояснювальна записка: 89 с., 20 рис., 0 табл., 3 дод., 21 джерело.

Об'єкт розробки: кросбраузерна автоматизована інформаційна система електронної комерції.

Мета кваліфікаційної роботи: розробка кросбраузерної автоматизованої інформаційної системи електронної комерції.

У вступі розглядається аналіз та сучасний стан проблеми, конкретизується мета кваліфікаційної роботи та галузь її застосування, наведено обґрунтування актуальності теми та уточнюється постановка завдання.

У першому розділі проведено аналіз предметної галузі, визначено актуальність завдання та призначення розробки, розроблена постановка завдання, задані вимоги до програмної реалізації, технологій та програмних засобів.

У другому розділі виконано аналіз існуючих рішень, обрано платформу для розробки, виконано проектування і розробка програми, наведено опис алгоритму і структури функціонування програми, визначені вхідні і вихідні дані, наведені характеристики складу параметрів технічних засобів, описаний виклик та завантаження програми, описана робота програми.

В економічному розділі визначено трудомісткість розробленої інформаційної системи, проведений підрахунок вартості роботи по створенню програми та розраховано час на його створення.

Практичне значення полягає у створенні сайту, котрий може продавати одяг, взуття та аксесуари.

Актуальність даного програмного продукту визначається попитом на веб-сайти для середніх чи великих компаній, котрі прагнуть мати свій магазин в інтернеті.

Список ключових слів: ІНТЕРНЕТ-МАГАЗИН, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, КОМП'ЮТЕР, ІНФОРМАЦІЙНА СИСТЕМА, АЛГОРИТМ, ІНТЕРФЕЙС, ПРОЕКТУВАННЯ, HTML, CSS, BOOTSTRAP, JQUERY, PHP, JAVASCRIPT.

## ABSTRACT

Explanatory note: 89 p., 20 pic., 0 tables, 3 appendix, 21 sources.

Object of development: cross-browser automated e-commerce information system.

The purpose of the qualification work: development of a cross-browser automated e-commerce information system.

The introduction considers the analysis and current state of the problem, specifies the purpose of the qualification work and its scope, provides a justification for the relevance of the topic and clarifies the problem.

In the first section the analysis of the subject area is carried out, the urgency of the task and purpose of development are defined, the statement of the task is developed, requirements to software realization, technologies and software are set.

The second section analyzes existing solutions, selects a platform for development, performs design and development of the program, describes the algorithm and structure of the program, identifies input and output data, provides characteristics of the parameters of hardware, describes the call and download of the program, describes the program.

The economic section determines the complexity of the developed information system, calculates the cost of work to create a program and calculates the time for its creation.

The practical significance is to create a website that can sell clothes and accessories.

The relevance of this software product is determined by the demand for websites for medium or large companies looking to have their own store online.

Keywords: ONLINE STORE, SOFTWARE, COMPUTER, INFORMATION SYSTEM, ALGORITHM, INTERFACE, DESIGNING, HTML, CSS, BOOTSTRAP, JQUERY, PHP, JAVASCRIPT.

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД – база даних;

ІС – інформаційна система;

ІТ – інформаційні технології;

ПЗ – програмне забезпечення;

ТЗ – технічне завдання;

HTTP – HyperText Transfer Protocol;

HTML – HyperText Markup Language;

CSS – Cascading Style Sheets;

JS – JavaScript;

PHP – Hypertext Preprocessor.

## ВСТУП

Дана розроблена інформаційна система призначена для застосування у сфері електронної комерції.

Електронна комерція - це підприємницька діяльність, яка пов'язана з поширенням, рекламуванням, просуванням, продажом послуг або товарів через Інтернет. Подібно до звичайного магазину, електронний магазин реалізує наступні основні функції: представлення товарів (послуг) покупцю, обробку замовлень, продаж і доставку товарів.

Оскільки в результаті відомих подій минулого року, які призвели до суттєвих економічних потрясінь і перетворення, тенденція зростання ринку електронної комерції багаторазово посилилася. Спостережуване раніше енергійний рух бізнесу в бік освоєння і розвитку свого бренду в мережі, в результаті пандемії стало ще більш активним. І в цьому немає нічого дивного, адже, по-перше, що виникли обмеження, пов'язані із запобіганням поширенню COVID-19, поставили багато компаній в положення, при якому єдино можливим варіантом їх виживання стала інтернет-торгівля. А, по-друге, пандемія вплинула і на поведінкову активність споживача, який в умовах карантину і масочного режиму став менше відвідувати звичні торгові офлайн-майданчики, віддаючи перевагу над ними інтернет-магазини [2].

Тобто розробка інтернет-магазину та його використання є актуальним питанням на сьогоднішній день, оскільки мільйони людей щодня, не виходячи з дому, купують різні товари в електронних магазинах. А інтернет - магазини в свою чергу істотно зменшують витрати виробника, заощадивши на утриманні звичайного магазину, розширюють ринки збуту, так само як і розширюють можливості покупця - купувати будь-який товар у будь-який час в будь-якій країні, в будь-якому місті, у будь-який час доби. Це дає Інтернет-магазинам перевагу перед звичайними магазинами. Цей момент є істотним під час переходу виробників із «звичайної» торгівлі на «електронну». Електронна



комерція в даний час є досить перспективною сферою діяльності для підприємців будь-якого класу. При цьому в якості покупців товарів або послуг можуть виступати як приватні особи, так і організації.

Інтернет-магазин об'єднує елементи прямого маркетингу з образом відвідування традиційного магазину. Відмінною рисою інтернет-магазинів в порівнянні зі звичайною формою торгівлі є те, що інтерактивний магазин може запропонувати значно більшу кількість товарів і послуг, і забезпечити споживачів значно більший обсяг інформації, необхідним для прийняття рішення про покупку.

Метою даної кваліфікаційної роботи є проектування і розробка автоматизованої інформаційної системи електронної комерції. Розроблена система буде працювати в рамках певного підприємства і виконувати функції основного майданчика для ведення комерційної діяльності.

Для досягнення поставленої мети в ході виконання роботи були вирішені такі основні завдання:

- проведення аналізу предметної галузі;
- аналіз теоретичних аспектів створення веб-сайтів;
- аналіз існуючих інтернет-сайтів компаній, що здійснюють продаж вказаних товарів;
- проведення аналізу існуючого ПЗ, мов програмування та засобів створення БД, що використовуються в веб-технологіях для розробки веб-застосунків;
- визначення структури та інтерфейсу інтернет-магазину;
- розробка структури інтернет-магазину;
- створення бази даних для сайту;
- наповнення сайту контентом.

## РОЗДІЛ 1

### АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ

#### 1.1. Загальні відомості з предметної галузі

Електронна комерція – ділова активність з купівлі-продажу товарів і послуг, що передбачає взаємодію сторін на основі інформаційних мереж (без безпосереднього фізичного контакту). Тобто це різновид електронного бізнесу (рис. 1.1), який включає в себе суспільні відносини стосовно купівлі-продажу товарів, послуг та інформації через Інтернет з використанням всіх інструментів, які доступні в мережі.

	Форма	Характеристика
Традиційні форми	Бізнес – споживач (B2C/C2B – Business-to-Consumer/Consumer-to-business)	Форма взаємодії компаній зі споживачами та навпаки, яка надає виробникам додаткові можливості для просування своїх товарів та послуг безпосередньо до споживачів, а споживачам більше можливостей для вибору. Ця форма була однією з перших стратегій е-бізнесу, яка передбачає прямий рух товарів і послуг від виробників до кінцевих споживачів, обминаючи посередників.
	Бізнес – бізнес (B2B – Business-to-Business)	Форма оперативної взаємодії підприємств на всіх рівнях, що орієнтується на діючий бізнес та на відпрацьовані схеми постачання та збуту, сталі внутрішні бізнес-процеси. Вона надає змогу об'єднати внутрішні мережі партнерів для спільного електронного документообігу, створює систему прямого розміщення замовлень з можливістю деталізації їх параметрів та відслідковування їх виконання в режимі реального часу і є новим засобом тіснішої комунікації підприємств. Ця форма набула сьогодні найбільшого поширення.
	Споживач – споживач (C2C – Consumer-to-Consumer)	Взаємодія приватних осіб щодо купівлі-продажу, обміну чи надання товарів, послуг та інформації в кіберпросторі. Прикладами такої форми взаємодії можуть бути дошки оголошень та Інтернет-аукціони.
Альтернативні форми	Бізнес – бізнес (B2B – Business-to-Business)	Всі рівні взаємодії електронними засобами між компаніями, які в такий спосіб можуть ефективно будувати дво- та багатосторонні господарські зв'язки, тобто реалізовувати продукцію, самостійно розміщувати замовлення, здійснювати моніторинг їх виконання за допомогою баз даних постачальників тощо.
	Бізнес – споживач (B2C – Business-to-Consumer)	Електронна роздрібна торгівля, орієнтована на кінцевого споживача. Найхарактернішим прикладом цього напрямку е-комерції є роздрібна торгівля в Інтернеті, спрямована безпосередньо на споживачів.
	Бізнес – адміністрація (B2A – Business-to-Administration)	Взаємодія компаній з адміністративними органами. Охоплює ділові зв'язки комерційних структур з державними та міжнародними організаціями. Цей сегмент е-комерції поки що недостатньо розвинутий, але має перспективи розвитку за умов, що уряди країн активніше використовуватимуть власні можливості для підтримки компаній.
	Споживач – адміністрація (C2A – Consumer-to-Administration)	Взаємодія споживачів з адміністрацією.

Рис. 1.1. Форми електронного бізнесу

Нижче наведено ряд переваг електронної комерції для всіх суб'єктів діяльності:

1) для споживачів:

- нижчі ціни у порівнянні з традиційними магазинами (за рахунок зменшення невиробничих витрат підприємства електронної комерції мають можливість оптимізувати цінову політику);
- доступність інформації про товари, послуги в Інтернет-магазинах у режимі реального часу цілодобово без вихідних.
- використання торговельних платформ, інтернет-магазинів з метою вивчення ринку товарів та послуг, порівняння їх характеристик, цін;
- можливість купувати елітні, рідкісні товари у зарубіжних магазинах, на аукціонах, резервувати місця в готелях різних країн та отримувати інші високоякісні послуги іноземних компаній;
- можливість конфіденційного здійснення покупок;
- можливість отримати швидко, просто безкоштовні зразки та скористатись фірмовою доставкою товарів від виробника;
- цифрові продукти, нематеріальні товари можуть бути одразу доставлені до споживача через мережеві канали;
- можливість обміну відгуками про товари та послуги, а також їх пошуку через соціальні мережі, спільноти, блоги тощо.

2) для виробників:

- зниження обсягу первісних вкладень в бізнес (зникає потреба в купівлі чи оренду великих торговельних площ, торговельного обладнання), невиробничих витрат (витрат на рекламу, витрат, пов'язаних з сервісним обслуговуванням та інформаційною підтримкою споживачів);
- зменшення чисельності персоналу та фонду оплати праці, що обумовлено повною або частковою автоматизацією процесів,

спрощенням комунікації в межах підприємства;

- скорочення циклу виробництва та продажу, підтримка бізнес-процесів у режимі on-line;
- надання нових видів послуг та освоєння нових сегментів ринку, глобальний доступ до світових ринків;
- рівність умов доступу до ринку як для великих корпорацій, так і для невеликих підприємств;
- цілеспрямований адресний вплив на споживацький сегмент за допомогою індивідуальних електронних засобів зв'язку;
- можливість роботи в режимі цілодобового доступу та інтерактивного спілкування із споживачем;
- підвищення рівня прихильності споживачів до торгової марки.

З іншого ж боку, електронна комерція має і негативні аспекти, які були згруповані за таким же принципом, як і вигоди:

1) для споживачів:

- через засоби ідентифікації особи користувача можна здійснювати контроль за нею, перевіряти її діяльність;
- сегмент населення, що не має доступу до мережі залишається неохопленим, до того ж не всі види товарів є представленими в мережі;
- споживачі не мають змоги в повній мірі пересвідчитись у якості товару чи послуги до моменту отримання його.
- необхідність сплати авансу в повному або частковому обсязі від вартості покупки;
- складний порядок повернення товарів у випадку бажання клієнта повернути покупку;
- логістика виконання замовлення має швидкість, або продуктивність, неадекватну швидкості Internet, окрім того зростає вартість невеликих замовлень за рахунок поштових послуг, послуг

перерахунку коштів;

- відсутність мотивів відвідування магазинів, безпосередньо не пов'язаних зі здійсненням купівель.

2) для виробників:

- посилення конкурентної боротьби, її перехід на глобальний рівень, до того ж для країн, що розвиваються, є характерним низький рівень охоплення населення мережею;

- необхідність забезпечення потужної технологічної основи для функціонування е-комерції, її постійного оновлення вимагає значних вкладень фінансових ресурсів;

- складність організації діяльності та розробки онлайн-вітрини (висока вартість, необхідність координації роботи відділу продажів, бухгалтерії, автоматизація обробки платіжних банківських карт і електронних грошей);

- внаслідок відкритості інформації зростає ймовірність порушення прав інтелектуальної власності, плагіату, фінансових шахрайств, тому постає проблема інформаційної безпеки;

- невизначеність та складність організації бухгалтерського обліку, внутрішнього контролю, переорієнтація інформаційних потоків із традиційних паперових носіїв у електронну форму, що за умов необізнаності працівників може ускладнити прийняття управлінських рішень;

- недостатня обізнаність працівників із сучасними інформаційними технологіями та системами, а висококваліфіковані кадри зазвичай вимагають значних витрат на оплату праці;

- необхідність залучення спеціалістів, що будуть здійснювати адміністрування сайту, оновлювати контент інформаційного ресурсу.

Таким чином, електронною комерцією можна назвати відносини між економічними суб'єктами, які здійснюються з допомогою інформаційних технологій, а саме мережі Інтернет. Аналіз переваг та недоліків е-комерції

дозволяє зробити висновок, що даний спосіб ведення бізнесу загалом має позитивний ефект для діяльності всіх суб'єктів господарювання.

На сьогоднішній день у зв'язку з розвитком інформаційних технологій і мережі Інтернет, та відповідно до вимог реальності інформаційного суспільства і ринкових умов, з кожним роком з'являється все більше і більше програмного забезпечення, технічних і методологічних засобів, що дозволяють розробляти та розміщувати інформаційні веб-ресурси в Інтернеті. Тому проведений аналіз предметної галузі показав, що зараз електронна комерція є найбільш пріоритетним напрямком розвитку бізнесу, особливо це загострилося в умовах економічної кризи і пандемії COVID-19. І враховуючи всі ці аспекти, в даній кваліфікаційній роботі було вирішено розробляти саме кросбраузерну автоматизовану інформаційну систему електронної комерції, що забезпечить контакт між ключовими сторонами в межах єдиного інформаційного простору, і дозволить не тільки підтримати ефективність бізнесу, но і вивести його на новий рівень.

## **1.2. Призначення розробки та галузь застосування**

Повне найменування системи: «Розробка кросбраузерної автоматизованої інформаційної системи електронної комерції».

Галузь застосування: електронна комерція.

Призначення розробки: здійснення електронної комерції та автоматизації діяльності підприємства.

## **1.3. Підстави для розробки**

Згідно з навчальним планом та відповідно до навчальної програми і графіку навчального процесу, наприкінці навчання студентами виконується кваліфікаційна робота.

Тема кваліфікаційної роботи є узгодженою з його керівником, кафедрою,

а також затверджена наказом ректора. Тобто підставами для розробки та виконання кваліфікаційної роботи виступають:

- наказ ректора Національного технічного університету «Дніпровська Політехніка» № 317-с від 07.06.2021 р.;
- завдання на кваліфікаційну роботу на тему: «Розробка кросбраузерної автоматизованої інформаційної системи електронної комерції».

#### **1.4. Постановка завдання**

Завданням кваліфікаційної роботи є розробка кросбраузерної автоматизованої інформаційної системи електронної комерції.

Для реалізації мети кваліфікаційної роботи необхідно вирішити наступні завдання:

- здійснити аналіз предметної галузі;
- виявити призначення розробки та галузь застосування;
- виявити основні вимоги для програми або програмного виробу (функціональних характеристик, інформаційної безпеки, складу та параметрів технічних засобів, інформаційної та програмної сумісності);
- розглянути функціональне призначення системи;
- здійснити опис використаних технологій та мов програмування;
- здійснити опис структури системи та алгоритмів її функціонування;
- здійснити опис розробленої системи (використаних технічних та програмних засобів, виклику та завантаження програми, опису інтерфейсу користувача);
- здійснити розрахунок трудомісткості та вартості розробки програмного продукту;
- здійснити розрахунок витрат на створення програми.

Після проведення глибокого аналізу завдання, стали зрозумілими основні вимоги до розроблюваної інформаційної системи:

- представлення компанії у мережі інтернет;
- поширення інформації про товари та їх вартість;
- реалізація інструментів продажу товарів;
- реалізація інформаційних модулів магазину;
- реалізація зворотного зв'язку;
- доступність веб-сайту на різних пристроях.

Основними клієнтами компанії, а отже і основною аудиторією розроблюваного сайту є люди, котрі зацікавлені у покупці одягу, взуття та аксесуарів. Виходячи з цих даних, необхідно провести розробку простого, легкого, але водночас привабливого дизайну основних сторінок сайту. Так само важливим фактором стала наявність мобільної версії сайту.

## **1.5. Вимоги до програми або програмного виробу**

### **1.5.1. Вимоги до функціональних характеристик**

Вимоги до функціональних характеристик розроблюваної інформаційної системи:

- вивід всіх товарів на сторінку;
- розміщення контактної інформації;
- наявність горизонтального навігаційного меню;
- навігація по сайту забезпечується за допомогою текстових посилань на його розділи;
- наявність кошика з товарами;
- наявність форми зворотного зв'язку;

Вимоги до структури сайту.

Структура розроблюваного веб-сайту повинна передбачати подальший розвиток за допомогою додавання нових розділів, а також можливість їх зміни, впровадження додаткових блоків і модулів.



Вимоги до навігації сайту.

Графічний інтерфейс користувача повинен мати можливість забезпечувати наочне, інтуїтивно зрозуміле уявлення структури розміщеної на ньому інформації, забезпечувати швидкий перехід до вибраному розділу сайту. Меню навігації має бути зрозумілим для користувача, а саме: гіперпосилання на сторінки повинні мати зрозумілі заголовки, щоб користувач міг без праці перейти до потрібної йому сторінці сайту. Для здійснення переходу між сторінками повинен бути реалізований блок, що складається зі списку гіперпосилань.

Вимоги до дизайну сайту.

Основні вимоги до дизайну сайту:

- графічні елементи сайту повинні бути розроблені з урахуванням специфіки подання інформації на веб-сайті;
- основні елементи меню та інші елементи навігації повинні бути доступні з головної сторінки сайту;
- не повинно бути надлишку інформації на сторінках сайту;
- перехід до обраної категорії повинен здійснюватися за допомогою горизонтального меню;
- дизайн сайту повинен бути сучасним і лаконічним;
- увагу користувача має концентруватися на обраних об'єктах.

При розробці дизайну сайту не повинні бути присутніми:

- елементи, які відволікали б увагу користувачів від перегляду інформації;
- візуально складно структурована інформація, важка для сприйняття;
- розділи, що випадають із загального оформлення сайту.

## 1.5.2. Вимоги до інформаційної безпеки

У зв'язку зі швидким розвитком інформаційних технологій, одним з найважливіших питань в діяльності компанії стало забезпечення інформаційної безпеки.

Інформація - це один з найцінніших активів будь-якої компанії або підприємства, і тому вона повинна бути належним чином захищена.

Інформаційна безпека - це збереження і захист інформації, а також її найважливіших елементів, в тому числі системи та обладнання, призначені для використання, заощадження та передачі цієї інформації. Іншими словами, це набір технологій, стандартів і методів управління, які необхідні для захисту інформаційної безпеки.

Мета забезпечення інформаційної безпеки - захистити інформаційні дані і підтримуючу інфраструктуру від випадкового або навмисного втручання, що може стати причиною втрати даних або їх несанкціонованого зміни. Інформаційна безпека допомагає забезпечити безперервність бізнесу.

Захист веб-ресурсів залишається одним із важливих напрямів ІБ. Щороку кількість веб-застосунків збільшується, зростає також кількість конфіденційної інформації, яка локалізується на серверах віддаленого доступу (особливо із використанням хмарних технологій).

У результаті цього зростають не тільки кількість атак на веб-ресурси, але й економічні наслідки таких атак. Таким чином, удосконалення методів і систем захисту веб-ресурсів від атак залишається актуальною проблемою, особливо з урахуванням постійного вдосконалення методів та інструментів атак і появи нових методів та інструментів. Удосконалення методів захисту веб-ресурсів від атак є також важливою в практичному застосуванні задачею внаслідок зростаючих економічних, соціальних і політичних наслідків від зловмисних дій.

Основні типи загроз ІБ веб-застосунків:

- загроза конфіденційності - несанкціонований доступ до даних;

- загроза цілісності - несанкціоноване спотворення або знищення даних;
- загроза доступності - обмеження або блокування доступу до даних.

Основним джерелом загроз інформаційної безпеки веб-застосунків є зовнішні порушники. Зовнішній порушник - особа, що мотивована, як правило, комерційним інтересом, має можливість доступу до сайту компанії, не володіє знаннями про досліджувану інформаційну систему, та має високу кваліфікацію в питаннях забезпечення мережевої безпеки і великий досвід в реалізації мережевих атак на різні типи інформаційних систем.

Загрози інформаційної безпеки пов'язані з кількома факторами: по-перше, це уразливість веб-додатків, або їх компонентів. По-друге, використання механізмів, що слугують для перевірки ідентифікації. По-третє, існування загроз безпеки, що відносяться до атак на самих користувачів. По-четверте, це витік або розголошення критичної інформації. Та по-п'яте, можливість здійснення логічних атак.

Наслідки, до яких можуть призвести подібні загрози:

- припинення працездатності сайту;
- втрата даних користувачів;
- фінансові та репутаційні втрати компанії;
- використання зловмисником веб-додатку з метою атаки або розсилки спаму на інші ресурси;
- блокуванням браузером, та як наслідок, втрата користувачів;

Аналіз актуальних атак на веб-ресурси (рис. 1.2).

№ за/п	Вид атаки	Вразливість веб-ресурсів, %	Протидія
1	Insufficient transport layer protection	70 %	Використання протоколу HTTPS.
2	Information leakage	56 %	Тестування програмної частини ресурсу, перевірка повідомлень на стороні сервера, моніторинг оповіщень про помилки
3	Cross-site scripting	47 %	Очищення та валідація вхідних даних
4	Brute force	29 %	Використання паролів високої складності, налаштування сервера на аналіз вхідних запитів
5	Content spoofing	26 %	Відмовитися від використання фреймів і не передавати в параметрах абсолютні або локальні шляхи до файлів
6	Cross-site request forgery	24 %	Перевірка вхідних даних з форм
7	URL redirector abuse	16 %	Валідація вхідних даних
8	Predictable resource location	15 %	Контроль доступу до файлів сервера

Рис. 1.2. Класифікація видів атак, їхня розповсюдженість і методи протидії

Найбільш популярною атакою є «Insufficient transport layer protection» - отримання даних під час передавання. Дана атака може бути виконана для 70% ресурсів. Для виключення можливості проведення таких атак достатньо використовувати протокол HTTPS.

Витік інформації («Information leakage»). Дану атаку можна виконати на 56% ресурсів. Витік інформації з додатків виникає в результаті відмови або неправильної роботи програми, а також у разі порушення її логіки. Для виключення можливості проведення атаки необхідно ретельно тестувати програмну частину ресурсу, проводити перевірку повідомлень на стороні сервера, моніторинг оповіщень про помилки.

Атаку «Cross-site scripting» - міжсайтове використання сценаріїв, можливо виконати на 47 % ресурсів. Атака дозволяє передати JavaScript-код на виконання в браузер користувача. Атаки такого роду часто також називають HTML-ін'єк-ціями, адже механізм їхнього впровадження дуже схожий із SQL-ін'єкціями, але на відміну від останніх, впроваджуваний код виконується в

браузері користувача. Для захисту від цього виду атак необхідно проводити очищення та валідацію вхідних даних.

Генерацію великої кількості запитів, або підбір паролів («Brute force») можливо виконати на 29 % ресурсів. Для захисту необхідно забезпечити використання паролів високої складності, налаштування сервера на аналіз вхідних запитів.

Атака «Content spoofing» - підмінна даних через заміну контенту сторінок можлива для 26 % ресурсів. Використовуючи цю техніку, зловмисник змушує користувача повірити, що сторінка згенерована веб-сервером, а не передана із зовнішнього джерела. Для захисту від даного виду атак потрібно відмовитися від використання фреймів і, найголовніше, ніколи не передавати в параметрах абсолютні або локальні шляхи до файлів.

Вид атак на відвідувачів веб-сайтів, який використовує недоліки протоколу HTTP - «Cross-site request forgery». Якщо жертва заходить на сайт, створений зловмисником, від її особи таємно відправляється запит на інший сервер (наприклад, на сервер платіжної системи), який здійснює якусь шкідливу операцію (наприклад, переказ грошей на рахунок зловмисника). Дану атаку можливо виконати на 24 % ресурсів. Для захисту необхідно проводити перевірку вхідних даних з форм, наприклад шляхом додавання унікального доданка.

Перенаправлення на інші сайти через підміну початкових посилань («URL redirector abuse»). Цей вид вразливостей, також як і багато інших перерахованих вище, є різновидом помилок перевірки вхідних даних і можлива на 16 % ресурсів. Вирішенням є валідація вхідних даних.

Ще однією популярною атакою є «Predictable resource location» - знаходження прихованого функціоналу та даних. Доступна на 15% ресурсів і вирішується шляхом контролю доступу до файлів сервера.

Виходячи з наведених даних, можна зробити висновки про те, що для захисту від більшості популярних видів атак достатньо належним чином перевіряти вхідні дані та тестувати програмні частини ресурсів. Також

рекомендовано використовувати шифрований протокол HTTPS. А також для правильної та безпечної роботи інформаційної системи не рекомендується вносити зміни до програмного коду та ресурсів програми.

### **1.5.3. Вимоги до складу та параметрів технічних засобів**

Для нормального функціонування інформаційної системи потрібен клієнтський персональний комп'ютер з такими мінімальними характеристиками:

- центральний процесор: Intel(R) Core(TM) i5-2540M CPU (з тактовою частотою не нижче 2,7 ГГц);
- оперативна пам'ять: 4GB;
- жорсткий диск: 250 GB;
- графічний процесор: Intel Core 2 Quad Q9450 C1 SLAWR 2.66GHz;
- операційна система: Windows 7.

### **1.5.4. Вимоги до інформаційної та програмної сумісності**

Дана інформаційна система може використовуватись на комп'ютерах із встановленню операційною системою Windows 7/8/10.

Також веб-сайт має повну сумісність з такими браузерами: Google Chrome, Firefox, Opera, Internet Explorer 11.

## РОЗДІЛ 2

### ПРОЄКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

#### 2.1. Функціональне призначення системи

Функціональне призначення системи: автоматизація діяльності підприємства, автоматизація та управління системою замовлень, автоматизація процесу обробки замовлення, здійснення взаємодії між користувачем і адміністратором, скорочення часу оформлення замовлення, автоматизація та управління системою відображення контенту, автоматизація та управління системою зворотного зв'язку, отримання адаптивного та інтуїтивно зрозумілого інтерфейсу тощо.

Інформаційна система розроблена як макет сайту електронної комерції з можливістю подальшого використання архітектури для розробки інших інтернет магазинів комерційного застосування у будь-якій сфері підприємницької діяльності. Такий веб-ресурс може бути корисний користувачам, адже він надає зручні інструменти для перегляду каталогу наявних товарів, їх покупки, отримання різного роду інформації щодо магазину. Більш за те, він має потенціал збільшити продажі компанії завдяки можливості опрацювання великої кількості інформації, одночасній обробці запитів багатьох відвідувачів, створенню рекламного майданчика для тих чи інших товарів та іншим корисним підприємству функціям.

Також інтерактивний магазин може запропонувати значно більшу кількість товарів і послуг ніж звичайний фізичний магазин та забезпечити споживачів значно більшим обсягом інформації, необхідним для прийняття рішення про покупку.

Дизайн і структура даного інтернет магазину складається з 5 основних

частин:

- головна сторінка;
- каталог всіх товарів;
- інформаційні сторінки;
- кошик;
- форма оформлення замовлення.

У розробку інформаційної системи входить процес створення макету головної сторінки магазину (рис. 2.1), на яку згодом будуть нанизуватися всі інші елементи. При цьому відбувається формування структурних блоків сайту - відокремлених модулів, кожен з яких відіграє певну роль і відповідає за певний функціонал ресурсу.

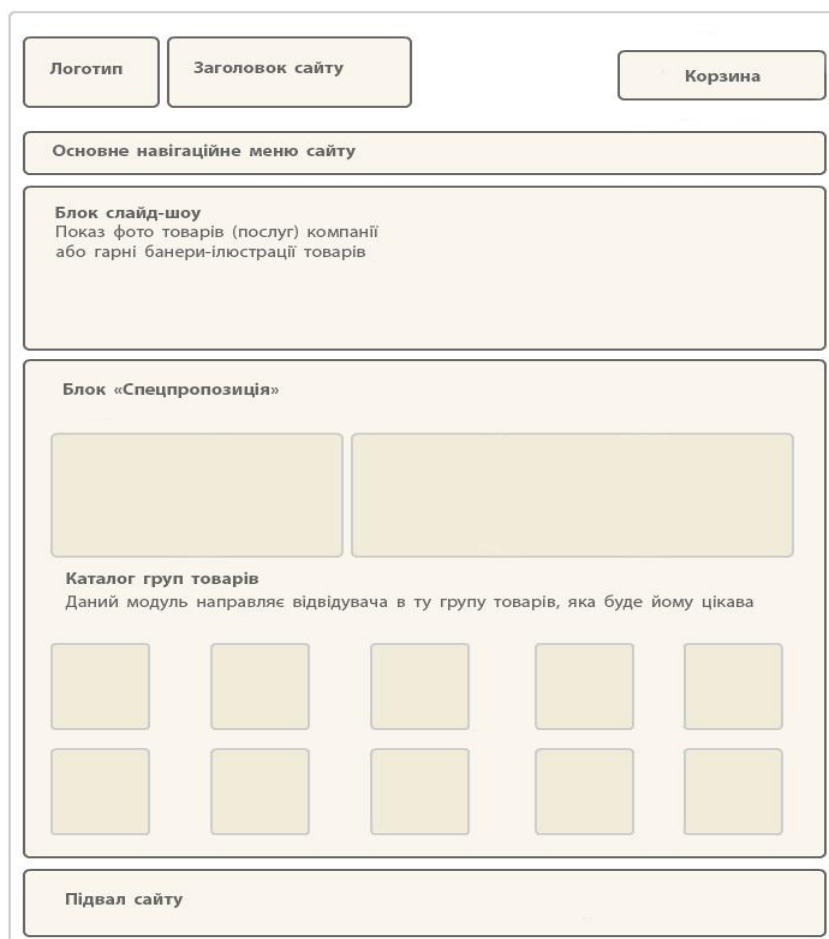


Рис. 2.1. Макет структури головної сторінки інтернет-магазину



Згідно розробленої структури була спроектована головна сторінка веб-сайту. До її складу входять всі основні структурні елементи, перехід за якими здійснюється за допомогою гіперпосилань.

До складу веб-сторінки входить:

1. Шапка сайту.

Місце, в якому зазвичай розташовується шапка сайту - верхня частина сторінки. В даному блоці розміщується:

- назва сайту (логотип);
- головне навігаційне меню;
- модуль кошику.

Всі ці елементи використовуються користувачем набагато частіше, ніж будь-які інші на сайті, тому вони і знаходяться в шапці, щоб до них завжди був доступ, з будь-якої сторінки сайту.

2. Основна частина сайту (контент).

Тут зазвичай розташована вся важлива інформація, здатна зацікавити відвідувача і підвести його до рішення здійснити покупку.

В даному блоці розміщується:

- текстовий контент;
- графічний контент.

До цих пунктів відносяться: рекламні банери/слайдери, діючі акції, графічно оформлені каталоги зі спеціальними пропозиціями, інформація про новинки, розміщення карточок популярних товарів тощо.

3. Підвал сайту.

Зазвичай підвалом називають ту область сайту, яка розташована під усіма іншими блоками. В даному блоці розміщується:

- дублікат логотипу;
- дублікат головного навігаційного меню;
- дублікат головної інформації про сайт;
- іконки соціальних мереж.

## 2.2. Опис застосованих математичних методів

Оскільки особливості предметної галузі розв'язуваної задачі не передбачають застосування математичних методів, при розробці кросбраузерної автоматизованої інформаційної системи електронної комерції математичні методи не використовувалися.

## 2.3. Опис використаних технологій та мов програмування

В рамках даної роботи під розробкою сайту мається на увазі створення прототипу сайту, а саме діючі сторінки: головна сторінка сайту, сторінка картки товару, корзина товарів, сторінка замовлення, сторінка підтримки, сторінка контактів, сторінка доставки. Прототип сайту не має власного серверного підґрунтя. А запити опрацьовуються за допомогою мови програмування JavaScript та локальної бази даних (API JSON).

HTML (англ. HyperText Markup Language, укр. Мова розмітки гіпертекстових документів) – стандартна мова розмітки веб-сторінок в Інтернеті (рис. 2.2). Більшість веб-сторінок створюються за допомогою мови HTML (або XHTML). Документ HTML оброблюється браузером та відтворюється на екрані у звичному для людини вигляді.



Рис. 2.2. Логотип HTML

HTML є похідною мовою від SGML, успадкувавши від неї визначення типу документа та ідеологію структурної розмітки тексту.

Попри те, що HTML – штучна комп'ютерна мова, вона не є мовою програмування.

HTML разом із каскадними таблицями стилів та вбудованими скриптами – це три основні технології побудови веб-сторінок.

HTML впроваджує засоби для:

- створення структурованого документа шляхом позначення структурного складу тексту: заголовки, абзаци, списки, таблиці, цитати та інше;
- отримання інформації із Всесвітньої мережі через гіперпосилання; створення інтерактивних форм;
- включення зображень, звуку, відео, та інших об'єктів до тексту.

CSS (англ. Cascading Style Sheets, укр. Каскадні таблиці стилів) – спеціальна мова, що використовується для опису зовнішнього вигляду сторінок, написаних мовами розмітки даних (рис. 2.3).

Найчастіше CSS використовують для візуальної презентації сторінок, написаних HTML та XHTML, але формат CSS може застосовуватися до інших видів XML-документів.

Специфікації CSS були створені та розвиваються Консорціумом Всесвітньої мережі.



Рис. 2.3. Логотип CSS

CSS має різні рівні та профілі. Наступний рівень CSS створюється на основі попередніх, додаючи нову функціональність або розширюючи вже наявні функції. Рівні позначаються як CSS1, CSS2 та CSS3. Профілі – сукупність правил CSS одного або більше рівнів, створені для окремих типів пристроїв або інтерфейсів. Наприклад, існують профілі CSS для принтерів, мобільних пристроїв тощо.

CSS (каскадна або блочна верстка) прийшла на заміну табличній верстці веб-сторінок. Головна перевага блочної верстки – розділення змісту сторінки (даних) та їхньої візуальної презентації.

CSS використовується авторами та відвідувачами веб-сторінок, щоб визначити кольори, шрифти, верстку та інші аспекти вигляду сторінки. Одна з головних переваг – можливість розділити зміст сторінки (або контент, наповнення, зазвичай HTML, XML або подібна мова розмітки) від вигляду документу (що описується в CSS).

Таке розділення може покращити сприйняття та доступність контенту, забезпечити більшу гнучкість та контроль за відображенням контенту в різних умовах, зробити контент більш структурованим та простим, прибрати повтори тощо. CSS також дозволяє адаптувати контент до різних умов відображення (на екрані монітора, мобільного пристрою (КПК), у роздрукованому вигляді, на екрані телевізора, пристроях з підтримкою шрифту Брайля або голосових браузерів та ін.).

Один і той самий HTML або XML документ може бути відображений по-різному залежно від використаного CSS. Стили для відображення сторінки можуть бути:

Стили автора (інформація надана автором сторінки): зовнішні таблиці стилів (англ. stylesheet), найчастіше окремий файл або файли .css внутрішні таблиці стилів, включені як частина документу або блоку стилі для окремого елемента.

Стилі користувача локальний .css-файл, вказаний користувачем для використання на сторінках і вказаний в налаштуваннях браузера (наприклад Opera)

Стилі переглядача (браузера) стандартний стиль переглядача, наприклад стандартні стилі для елементів, визначені браузером, використовуються коли немає інформації про стиль елемента або вона неповна.

Стандарт CSS визначає порядок та діапазон застосування стилів, тобто, в якій послідовності і для яких елементів застосовуються стилі. Таким чином, використовується принцип каскаду, коли для елементів вказується лише та інформація про стилі, що змінилася або не визначена загальнішими стилями.

JavaScript (JS) – динамічна, об'єктно-орієнтована прототипна мова програмування. Реалізація стандарту ECMAScript (рис. 2.4).



Рис. 2.4. Логотип JS

Найчастіше використовується для створення сценаріїв веб-сторінок, що надає можливість на стороні клієнта (пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінки.

JavaScript класифікують як прототипну (підмножина об'єктно-орієнтованої), скриптову мову програмування з динамічною типізацією. Окрім прототипної, JavaScript також частково підтримує інші парадигми програмування (імперативну та частково функціональну) і деякі відповідні

архітектурні властивості, зокрема: динамічна та слабка типізація, автоматичне керування пам'яттю, прототипне наслідування, функції як об'єкти першого класу.

Мова JavaScript використовується для:

- написання сценаріїв веб-сторінок для надання їм інтерактивності;
- створення одно сторінкових веб-застосунків (ReactJS, AngularJS, Vue.js);
- програмування на стороні сервера (Node.js);
- стаціонарних застосунків (Electron, NW.js);
- мобільних застосунків (React Native, Cordova);
- сценаріїв в прикладному ПЗ (наприклад, в програмах зі складу Adobe Creative Suite чи Apache JMeter);
- всередині PDF-документів тощо.

Незважаючи на схожість назв, мови Java та JavaScript є двома різними мовами, що мають відмінну семантику, хоча й мають схожі риси в стандартних бібліотеках та правилах іменування. Синтаксис обох мов отриманий «у спадок» від мови C, але семантика та дизайн JavaScript є результатом впливу мов Self та Scheme.

jQuery – це фреймворк, бібліотека, створена для взаємодії мови програмування JavaScript і мови гіпертекстової розмітки HTML (рис. 2.5) [14].



Рис. 2.5. Логотип jQuery

Представлений громадськості він був у 2006 році на одній з конференцій, присвячених розробкам в галузі соцмереж і браузерів, яка проходила в Нью-Йорку. Автор проекту – молодий програміст Джон Резиг, який очолює проект і нині.

Бібліотека дозволяє звертатися абсолютно до будь-якого елементу DOM для зміни його вмісту, структури, параметрів і навіть оформлення.

Ясність у розумінні попереднього речення внесе визначення DOM – це універсальний багато платформний (не залежить від операційної системи і використовуваного мови програмування) користувальницький інтерфейс, за допомогою якого програми і міні-програми (скрипти) здатні відкривати HTML і XML-файли для їх коригування.

При цьому правила, умови чи обмеження фактично відсутні, що відкриває перед юзером необмежені можливості по обробці і зміни вмісту HTML. Будь-відомий документ представляється у вигляді ієрархічного дерева, в якому кожна гілка і її дочірні елементи – це атрибути, графічні або текстові об'єкти.

JDOM – наступний крок в еволюції об'єктної моделі документа, створена з урахуванням всіх особливостей Java, jQuery поки не використовується, принаймні громадськості це не представлено.

jQuery – це написана на JavaScript бібліотека, заснована на взаємодії вищезазначеного мультипарадигмової мови програмування з HTML-документами і файлами XML відповідного вмісту.

За допомогою фреймворку або каркасу, як його частіше називають вітчизняні розробники, можна легко отримати доступ до гіпертекстової сторінці та маніпулювати її структурою та елементами без будь-яких обмежень.

Основні можливості каркасу:

- звернення до будь-яких елементів DOM для їх зміни і обробки;
- сумісність з AJAX;

- величезна бібліотека плагінів, особливо стосуються візуального оформлення сторінок;
- підтримка роботи з подіями.

Для тих користувачів, які вже знайомі з JavaScript, цього цілком достатньо. Вони чудово знають, наскільки трудомістким є операція вибору елементів в документі для їх зміни.

Це не тільки довгий процес, але ще і дуже незручний навіть для досить досвідчених розробників. Бібліотека ж дозволить в рази скоротити вихідний код програми, спростивши його до межі. Новачкам слід на практиці побачити всі переваги фреймворку.

Методологія БЕМ (Блок, Елемент, Модифікатор) - компонентний підхід у веб-розробці. В його основі лежить принцип розподілу інтерфейсу на незалежні блоки. Він дозволяє легко і швидко розробляти інтерфейси будь-якої складності і повторно використовувати вже існуючий код, уникаючи «Сору-Paste».

Блок - функціонально незалежний компонент сторінки, який може бути повторно використаний. В HTML блоки представлені атрибутом class.

Особливості:

- Назва блоку характеризує сенс («що це?» - «меню»: menu, «кнопка»: button), а не стан («який, як виглядає?» - «червоний»: red, «великий»: big).

- Блок не повинен впливати на своє оточення, тобто блоку не слід задавати зовнішню геометрію (у вигляді відступів, кордонів, що впливають на розміри) і позиціонування.

- В CSS по БЕМ методології також не рекомендується використовувати селектори за тегами або по id.

Таким чином забезпечується незалежність, при якій можливе повторне використання або перенесення блоків з місця на місце.

Вкладеність:

- блоки можна вкладати один в інший;



– допустима будь-яка вкладеність блоків. Наприклад:

```
<! - Блок `header` ->
<header class = "header">
  <! - Вкладений блок `logo` ->
  <div class = "logo"> </ div>
  <! - Вкладений блок `search-form` ->
  <form class = "search-form"> </ form>
</ header>
```

Елемент - складова частина блоку, яка не може використовуватися у відриві від нього.

Особливості:

– Назва елемента характеризує сенс («що це?» - «пункт»: item, «текст»: text), а не стан («який, як виглядає?» - «червоний»: red, «великий»: big).

– Структура повного імені елемента відповідає схемі: ім'я-блока\_\_імя-елемента. Ім'я елемента відокремлюється від імені блоку двома підкресленнями (\_\_).

Вкладеність:

– елементи можна вкладати один в одного;  
– допустима будь-яка вкладеність елементів;  
– елемент - завжди частина блоку, а не іншого елемента. Це означає, що в назві елементів не можна прописувати ієрархію виду block\_\_elem1\_\_elem2.

Блок створюється для повторного використання коду, а тому він повинен бути незалежним від інших компонентів сторінки. Якщо ж ділянка коду не може бути використаний без блоку-батька, ми називаємо його як елемент. Елементи - це внутрішня реалізація блоку, але вони повинні бути не у всіх блоків. Наприклад:

```
<! - Блок "sidebar" ->
<div class = "sidebar">
```

```

<! - Елемент "ul" блоку "sidebar" ->
<ul class = "sidebar__list">
  <! - Елемент "li" блоку "sidebar" ->
  <li class = "sidebar__item"> first </ li>
</ ul>
</ div>

```

Модифікатор - сутність, що визначає зовнішній вигляд, стан або поведінку блоку або елемента.

Особливості:

- Назва модифікатора характеризує зовнішній вигляд («який розмір?», «Яка тема?» тощо - «розмір»: size\_s, «тема»: theme\_islands), стан («чим відрізняється від інших?» - «відключений»: disabled, «сфокусований»: focused) і поведінку («як поводить?»), «як взаємодіє з користувачем?» - «напрямок»: directions\_left-top).

- Ім'я модифікатора відділяється від імені блоку або елемента одним підкресленням ( ), або двійним дефісом (-).

Структура повного імені модифікатора відповідає схемі:

- ім'я-блока\_ім'я-модифікатора;
- ім'я-блока\_\_ім'я-елемента\_ім'я-модифікатора.

Неможливо одночасно використовувати два однакових модифікатора з різними значеннями.

Згідно з методологією БЕМ, модифікатор не можна використовувати самостійно у відриві від виділяється блоку або елемента. Він повинен змінювати вид, поведінку або стан компонента, а не замінювати його.

Наприклад:

```

<div class=" sidebar ">
  <!-- Елемент "ul" блоку " sidebar " -->
  <ul class=" sidebar__list ">
    <!-- Елемент "li" блоку " sidebar " -->
    <li class=" sidebar__item sidebar__item--lite">first</li>

```

</ul>

</div>

Мікс - прийом, що дозволяє використовувати різні БЕМ-суті на одному DOM-вузлі.

Мікси дозволяють:

- поєднувати поведінку і стилі декількох сутностей без дублювання коду;
- створювати семантично нові компоненти інтерфейсу на основі наявних.

Bootstrap - це відкритий і безкоштовний HTML, CSS і JS фреймворк, який використовується веб-розробниками для швидкої верстки адаптивних дизайнів сайтів та веб-додатків [15].

Основна область його застосування - це фронтенд розробка сайтів та інтерфейсів адмінок. Серед аналогічних систем (Foundation, UIKit, Semantic UI, InK і ін.) Фреймворк Bootstrap є найпопулярнішим.

Фреймворк Bootstrap - це набір CSS і JavaScript файлів. Щоб його використовувати ці файли необхідно просто підключити до сторінки. Після цього вам стануть доступні інструменти даного фреймворка: колоночная система (сітка Bootstrap), класи і компоненти.

Bootstrap складається з:

- інструментів для створення макета (обгортковий контейнерів, потужної системи сіток, гнучких медіа-об'єктів, адаптивних утілітних класів);
- класів для стилізації базового контенту: тексту, зображень, коду, таблиць і figure;
- готових компонентів: кнопок, форм, горизонтальних і вертикальних навігаційних панелей, слайдерів, випадаючих списків, акордеонів, модальних вікон, спливаючих підказок тощо;
- утілітних класів для вирішення традиційних завдань найбільш часто виникають перед веб-розробниками: вирівнювання тексту, відображення та приховування елементів, завдання кольору, фону, margin і

padding відступів тощо.

Переваги Bootstrap при його використанні для frontend розробки сайтів:

- висока швидкість створення якісної адаптивної верстки навіть початківцями веб-розробниками (досягається це завдяки використанню готових класів і компонентів, створених професіоналами);
- кросбраузерність і кросплатформеність (коректне відображення та робота сайту у всіх підтримуваних цим фреймворком браузерах і операційних системах);
- наявність великої кількості готових добре продуманих компонентів, протестованих величезним співтовариством веб-розробників на різних пристроях;
- можливість налаштування під свій проект, досягається це за допомогою зміни SCSS змінних і використання Міксини (можна змінити кількість колонок, кольору, радіус заокруглень, відступи між колонками тощо);
- низький поріг входження; для роботи з фреймворком не обов'язково мати «глибокі» знання з HTML, CSS, JavaScript і jQuery (досить знати тільки основи цих технологій);
- однорідність дизайну і його узгодженість між різними компонентами (в Bootstrap всі компоненти виконані в єдиному стилі);
- наявність величезної кількості спільнот і навчальних матеріалів; при бажанні це допоможе не тільки добре розібратися в фреймворку, але і знайти відповіді практично на будь-які виникають у вас питання.

Bootstrap, як і більшість подібних фреймворків, має недоліки. Серед них можна відзначити наступні:

- більш великий розмір кінцевих css і js-файлів проекту, ніж вони вийшли, якби ми все це створювали самостійно (це пов'язано з тим, що стилі фреймворка і його js-код містять універсальний код (на всі випадки життя), а за фактом для конкретного проекту з усього цього може знадобитися тільки частина);

– складність використання Bootstrap для верстки сайтів з унікальним дизайном, тому що розробка в цьому випадку буде супроводжуватися значним переписуванням його коду і простими налаштуваннями Bootstrap змінних тут вже не обійтись.

Перший недолік можна трохи «згладити», якщо виконати самостійну збірку проекту з вихідних кодів і включити в неї тільки ті компоненти, які нам потрібні.

Git - це система контролю версій файлів з вихідним кодом програми. Написавши повний код або його частину, Git зберігає поточний стан файлу. Далі розробник продовжує роботу над кодом і розуміє, що зробив помилку або не оптимізований код. Щоб повернутися в попереднє збережений стан, не потрібно видаляти ділянки некоректного коду, переглядаючи його повністю. Досить ввести спеціальну команду в Git, і система поверне розробника до попереднього стану файлу [16].

З точки зору програмного забезпечення Git буває трьох видів:

- Консольна утиліта, яка потребує знання текстових команд для управління репозиторієм.
- Графічне додаток для ПК.
- Онлайн-версія, яка називається GitHub - саме тут зберігаються репозиторії більшості розробників, які діляться своїм дітищем з усіма користувачами інтернету.

Git використовується як для командної розробки, так і одним програмістом. Кожен розробник команди створює довірену йому частину проекту. Дослідне особа переглядає всі версії файлів, виконаних молодшими розробниками, робить правки і здійснює збірку проекту з частин.

Репозиторій - це початкова точка для ведення проекту. У репозиторії зберігаються файли проекту і виробляються над ними потрібні операції: створюються гілки для різних розробників і файлів, проводиться розподіл на версії, пишуться коментарі, здійснюється складання кінцевого результату в готовий файл проекту. Головна майстер-гілка здійснює остаточний збір

проекту.

Сховище складається з наступних елементів:

- Папка. У репозиторій потрапляє папка з файлами проекту, над якими буде вестися подальша робота.
- Гілка. Кожен розробник команди створює гілку для себе. Якщо розробник один, то її можна не створювати.
- Коміт. Це команда для збереження поточного стану файлу. Всього в репозиторії можна створити 10 комітів. Розробник може повернутися до будь-якого з 10 станів файлу для редагування коду. Після збереження стану файлу, над ним можна продовжити роботу і зберегти знову (закомітити).

Версії Git:

1. Локальна система. Файли зберігаються на одному комп'ютері. Це робить безліч обмежень для командної розробки.
2. Централізована. Використовує загальний сервер, надаючи команді хмарне сховище файлів. Сервер забезпечує віддалену роботу команди над проектом. Однак даний спосіб не безпечний і сервер може піддатися збою або злому, що може привести до пошкодження файлів.
3. Розподілена система. Створена для віддаленої та локальної роботи. Пошкоджені файли легко відновити. Є найдорожчою, але найнадійнішою організацією Git.

PHP (від англійського Hypertext Preprocessor - «препроцесор гіпертексту») - це скриптова мова, на якій створюють сайти і веб-додатки. Мова інтегрується з більшістю веб-серверів і працює з усіма поширеними операційними системами. У PHP зрозумілий синтаксис і низький поріг входу для вивчення [17].

Переваги:

- Висока продуктивність. Для веб-розробки швидкість виконання програм - ключовий параметр. У цьому PHP обходить більшість мов, включаючи Python.
- Робота з різними платформами. У веб-розробці важливо

забезпечити стабільну якість, незалежно від рішення на стороні серверу операційної системи або веб-сервера. PHP підтримує Oracle, MySQL, Apache, Windows, Unix, Linux і інші платформи.

- Популярність. 79% сайтів написані на PHP, ця мова використовують поширені системи управління сайтами, наприклад WordPress, Drupal, Magento.

- Велике ком'юніті. Спільнота PHP-розробників більше, ніж у Python, тому і вибір бібліотек в області веб-розробки більш широкий. У ком'юніті простіше знайти потрібну керівництво або отримати відповідь на питання.

Недоліки:

- Несистемний синтаксис. Наприклад, назви функцій можуть бути схожі, але виконувати зовсім різні операції. Також, коли мову допрацьовували, використовували мови C і Java, тому можна зустріти їх синтаксис. Початківець розробник може заплутатися, а досвідчений, навпаки, побачить в цьому перевага, тому що йому буде легше перейти на нові мови.

- Негативна слава. PHP створювався як мова, яку може використовувати людина без підготовки. Тому його поверхнево освоювали недосвідчені програмісти і кидалися виконувати замовлення. Сайти ламалися, а горе-фахівці нічого не могли зробити. Хоча мова вдосконалюється і сучасний PHP нічим не гірше молодих мов, все ще можна почути негативні думки.

- Можливість помилок. Якщо в коді буде помилка, то мова дозволить використати її. Якщо недолік буде очевидним, помилку буде дуже складно знайти. Тому, хоча PHP - проста і гнучка мова, неуважний розробник може створити проблеми собі і колегам.

PHP - мова, що інтерпретується з динамічною типізацією. Застосовується переважно для розробки веб-додатків. Цією мовою написані популярні движки для сайтів і фреймворки.

Open Server - це програмна платформа для веб-фахівців, що забезпечує створення і функціонування в середовищі Windows повноцінного сервера. Це

збірка всіх необхідних компонентів, за допомогою якої можна створювати на своєму комп'ютері сайти будь-якої складності [18].

Local server - Локальний сервер - це емулятор хостингу, спеціальна програма, що дозволяє веб-розробникам розробляти сайт на локальному комп'ютері, без необхідності виходу в Інтернет.

Open Server є повноцінним професійним інструментом, найбільш затребуваним серед розробників. Має простий і зрозумілий інтерфейс і навігацію, зручну керуючу програму. Відрізняється великим вибором підключаються компонентів.

У повну версію включено необхідний набір програм для моментального старту: Opera, Sublime, WinSCP, Simplewall, Jcpicker, Bright Master. Сервер регулярно оновлюється, що говорить про стабільність його роботи.

Користувачам пропонується на вибір кілька HTTP-модулів (Apache або Nginx), різні версії модулів PHP і СУБД. До того ж між ними працює швидке переключення.

Установка стаціонарна і портативна.

переваги:

- Багатофункціональність.
- Інтерфейс багатомовний.
- Є детальна інструкція по роботі з сервером.
- Доступ до доменів здійснюється в один клік.
- Можливість роботи через командний рядок.
- Швидкість запуску і зупинки служби.
- Можливість перегляду логів в реальному часі.

Недолік один - не передбачено автоматичне оновлення серверу.

## **2.4. Опис структури системи та алгоритмів її функціонування**

Нижче представлений короткий опис структури системи (рис. 2.6):

Папка Willberries – головна (коренева) папка проекту, в якій зберігаються



всі файли та директорії.

Папка `css` – дана директорія містить весь CSS код, а також деякі плагіни, що використовуються для стилізації контенту сайту.

Папка `db` – дана директорія містить файл `json` з локальною базою даних.

Папка `favicon` – дана директорія містить іконки (для кожного браузера окрема), що супроводжують сайт в інтернет-середовищі. Ці іконки з'являються на вкладках, в результатах пошуку і допомагають сайту бути більш впізнаваним.

Папка `img` – дана директорія містить всі зображення, що використовуються на сайті.

Папка `js` – дана директорія містить весь JavaScript код, який використовується для здійснення логіки сайту.

Папка `phpmailer` – дана директорія містить папку з плагіном, що в свою чергу реалізує відправку листів і файлів на пошту.

Файл `.gitignore` – текстовий файл, що потрібен для приховування файлів і папок від системи контролю версій Git.

Файл `index.html` – даний файл містить контент домашньої (головної) сторінки, які люди бачать, коли вперше потрапляють на сайт.

Файл `aboutUs.html` – даний файл містить контент сторінки, на якій розміщено більш детальну інформацію щодо сайту.

Файл `contacts.html` – даний файл містить контент сторінки, на якій розміщена контактна інформація сайту.

Файл `location.html` – даний файл містить контент сторінки, на якій розміщена інформація про доставку.

Файл `support.html` – даний файл містить контент сторінки зворотного зв'язку сайту.

Файл `sendmail.php` – даний файл здійснює налаштування плагіну та відправку листів на пошту.

Файл `server.php` – локальний сервер сайту.

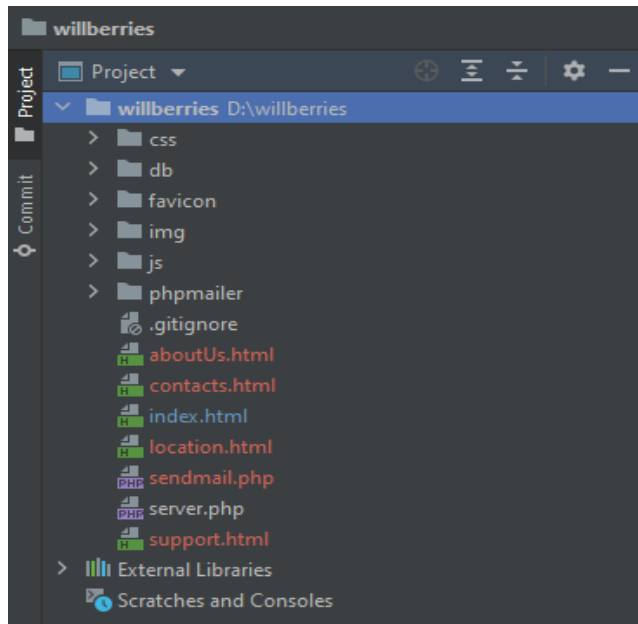


Рис. 2.6. Структура системи

Діаграма алгоритму функціонування інформаційної системи представлена нижче (рис. 2.7):

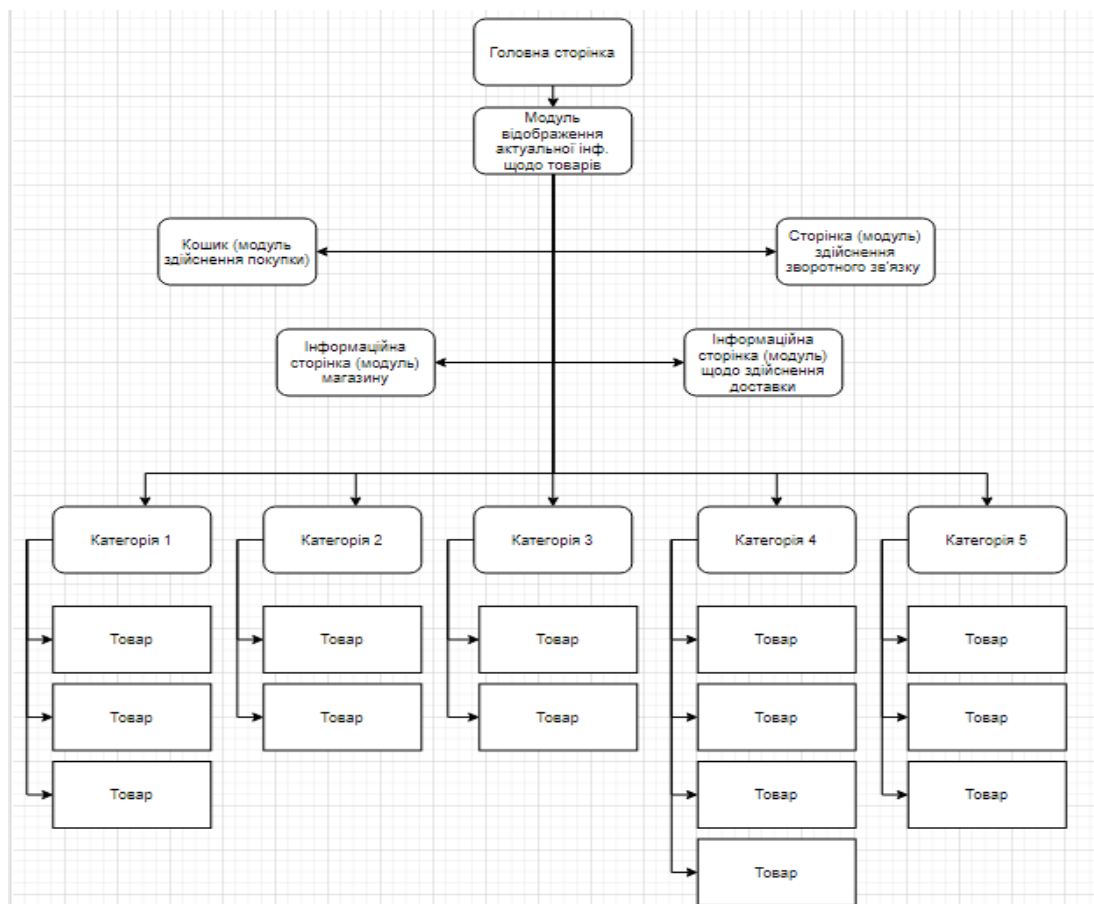


Рис. 2.7. Алгоритм функціонування сайту електронної комерції

## **2.5. Обґрунтування та організація вхідних та вихідних даних програми**

Вхідними даними для системи є:

- інформація про клієнта, що надається їм в процесі оформлення замовлення;
- інформація про товарні позиції (окремі товари);
- інформація щодо успішності оформлення замовлення;
- інформація щодо здійснення доставки замовлення клієнту;
- інформація щодо організації і роботи інтернет-магазину;
- інформація, що надається клієнтом в службу підтримки;
- інша статистична інформація щодо роботи магазину.

Вихідними даними для системи є:

- рахунок, який виставляється системою клієнту для сплати його замовлення;
- список активних кошиків і відповідні їм списки товарних позицій;
- повідомлення про виконання замовлення товару(-ів) в інтернет-магазині;
- список товарів, які перебувають в процесі доставки з інтернет-магазину на склад;
- кількість товарів, які перебувають в процесі доставки з інтернет-магазину на склад;
- повідомлення щодо здійснення зворотного зв'язку з клієнтом при здійсненні покупки в інтернет магазині;
- повідомлення щодо здійснення зворотного зв'язку з клієнтом при зверненні на сторінку підтримки.

## **2.6. Опис розробленої системи**

### **2.6.1. Використані технічні засоби**

Для функціонування системи необхідна клієнтська персональна комп'ютера з наступними мінімальними характеристиками:

1. центральний процесор: Intel(R) Core(TM) i7-2620M CPU @ 2.70GHz 2.70 GHz;
2. графічний процесор: AMD Radeon HD 6470M;
3. жорсткий диск: SSD накопичувач PATRIOT Burst 480 GB;
4. оперативна пам'ять: DDR3-1333 8 GB;
5. материнська плата: HP EliteBook 8560p (S-G2, QM67, DDR3, HD 6470M 1GB);
6. мережева карта: Intel(R) 82579LM Gigabit Network Connection;
7. BIOS: Hewlett-Packard 68SCF Ver. F.08, 26/08/11;
8. операційна система: Windows 10.

### **2.6.2. Використані програмні засоби**

Під час виконання кваліфікаційної роботи були використані такі програмні засоби:

1. операційна система Microsoft Windows 10 Pro;
2. браузер: Google Chrome, Opera;
3. редактори коду: Visual Studio Code, IDE PhpStorm;
4. графічний редактор: Photoshop;
5. сервіс для розробки інтерфейсів: Figma;
6. локальний сервер: OpenServer.

### **2.6.3. Виклик та завантаження програми**

Для початку роботи з сайтом потрібно відкрити браузер, у вікні написати

адресу сайту URL: <http://willberries> і перейти на головну сторінку сайту (рис. 2.8).

При переході за адресою веб-браузер виконує наступні дії:

- Браузер посилає запит на DNS-сервер і отримує справжню адресу серверу на якому зберігається потрібний вам сайт.
- Браузер надсилає HTTP-запит на сервер з проханням надати доступ до файлів сайту і копіює їх. Цей запит і всі дані, що пересилаються між клієнтом і сервером, передаються в мережі за допомогою протоколу TCP/IP.
- Якщо сервер вирішує задовольнити запит, він відсилає клієнту повідомлення "200 OK" і передає файли веб-сайту браузеру частинами, згрупованими у маленькі набори, які називаються пакетами даних.
- Браузер збирає до купи всі пакети і дістає з них елементи сайту, а потім відтворює їх у вигляді готової веб-сторінки.

IP-адреса сайту представляє унікальний ідентифікатор, що вказує на місце зберігання сайту в мережі. Винайдені DNS-сервери – це спеціальні сервери, що зіставляють веб-адресу, що вводиться у браузер (наприклад, "mozilla.org") зі справжньою (IP) адресою сайту.

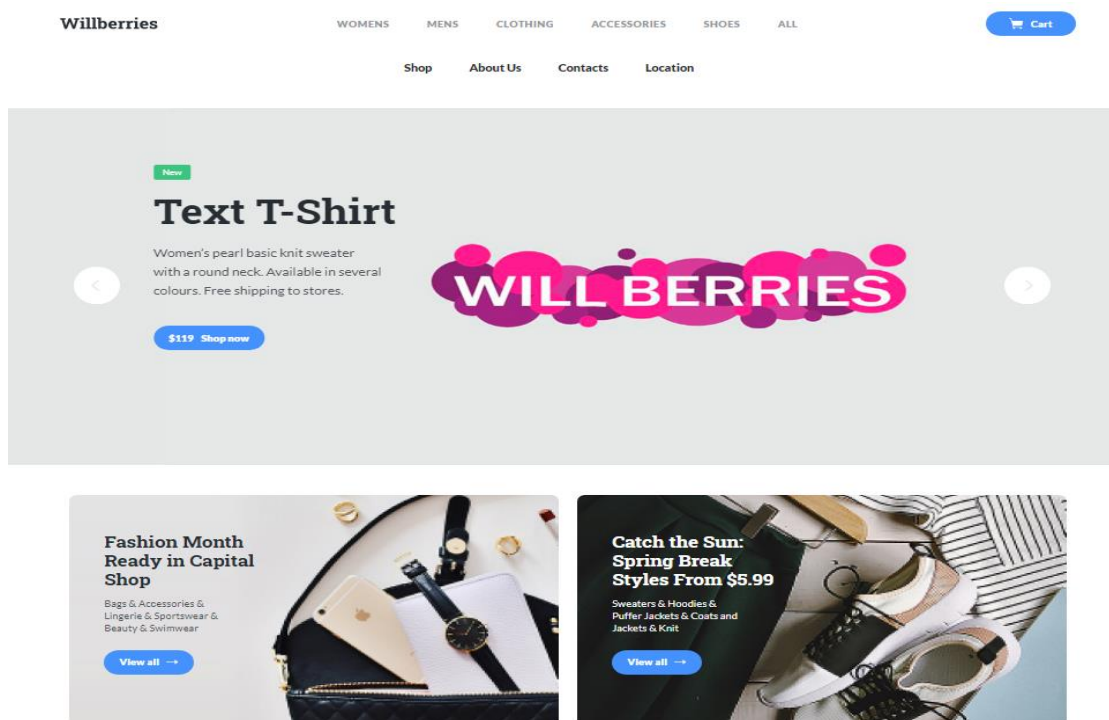


Рис. 2.8. Головна сторінка інтернет-магазину

## 2.6.4. Опис інтерфейсу користувача

Після того як користувач відкрив браузер, ввів URL адресу сайту, і опинився на головній сторінці сайту, він має змогу ознайомитися з інтерфейсом інтернет магазину.

Нижче представлений короткий опис можливостей сторінок:

Головна сторінка – показано декілька популярних товарів, блоги з підбіркою найцікавіших товарів, розділи товарів, банери, новини.

Продукт – назва, малюнок товару, опис.

Кошик – назва товару, кількість, ціна, загальна сума.

Замовлення – телефон, ім'я.

Вдала покупка – виводиться повідомлення про вдалу покупку.

Невдала покупка – виводиться повідомлення про невдалу покупку та вид помилки.

Сторінка підтримки – форма зворотного зв'язку.

Сторінка контактів – інформація щодо контактних телефонів і час роботи.

Сторінка місцезнаходження – інформація щодо доставки.

Сторінка про магазин – характерна інформація щодо сайту.

На головній сторінці (рис. 2.12) користувач може ознайомитися з найпопулярнішими товарами магазину, перейти до каталогу всіх товарів (рис. 2.11), впорядкувати їх за допомогою фільтрів (рис. 2.9), що знаходяться в шапці головної сторінки магазину, згідно вподобань, відкрити модальне вікно кошику і форми замовлень, та при бажанні безпосередньо здійснити замовлення обраних товарів. З цієї сторінки користувач за допомогою навігаційного меню, що знаходиться у шапці (рис. 2.9) і підвалі (рис. 2.10) сторінки, може перейти на такі сторінки, як: «Про нас», «Місцезнаходження», «Підтримка», де він зможе дізнатися більше інформації, щодо роботи магазину. Також у підвалі сторінки користувач зможе знайти посилання на соц. мережі інтернет магазину (рис. 2.10).

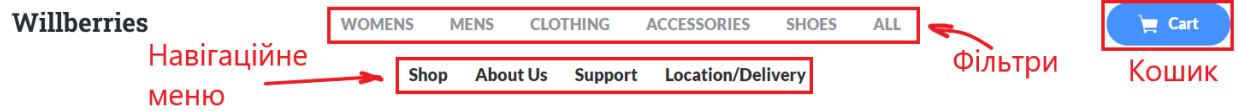


Рис. 2.9. Шапка головної сторінки

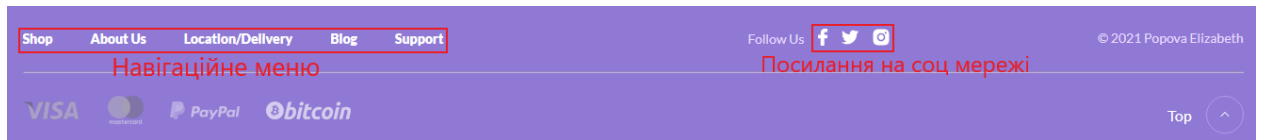


Рис. 2.10. Підвал головної сторінки

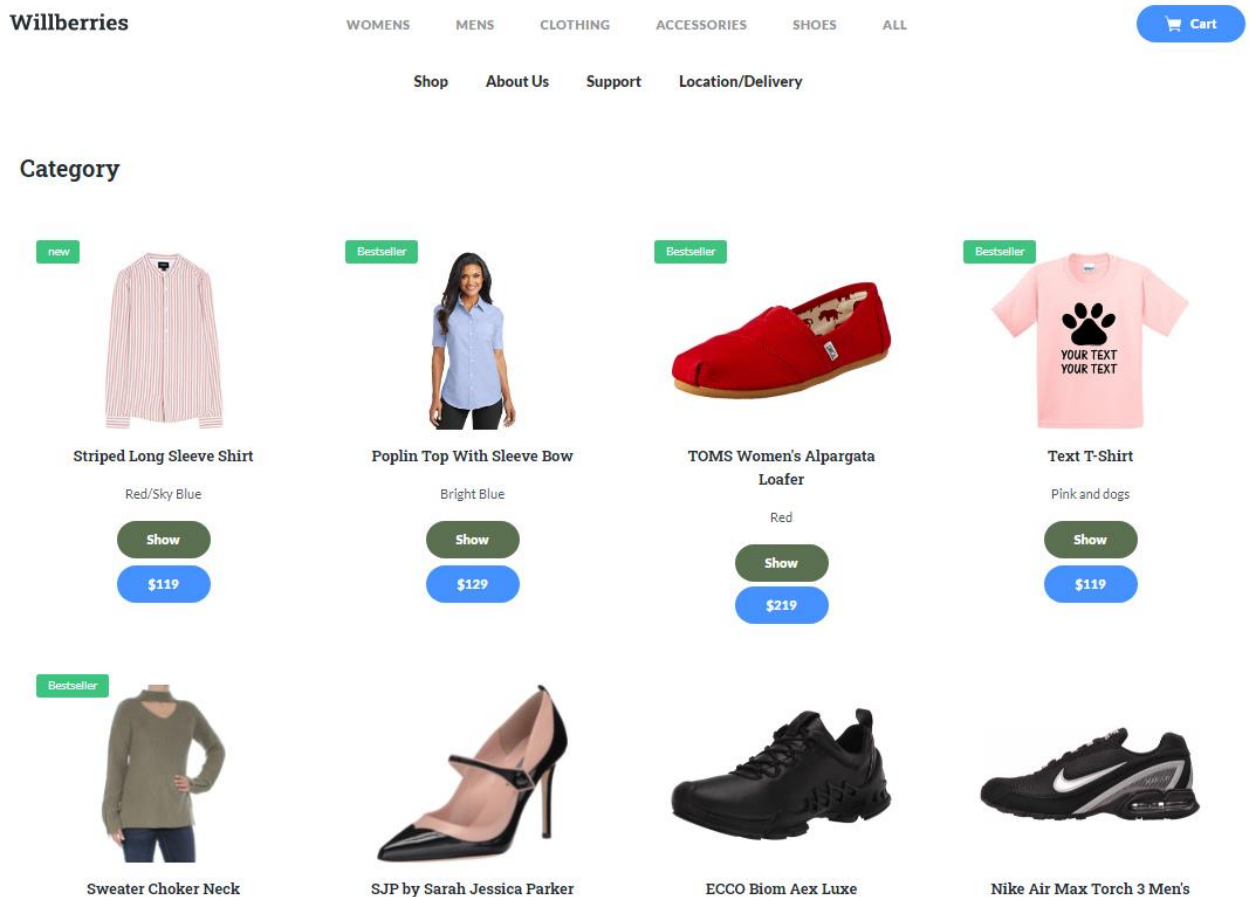


Рис. 2.11. Каталог товарів

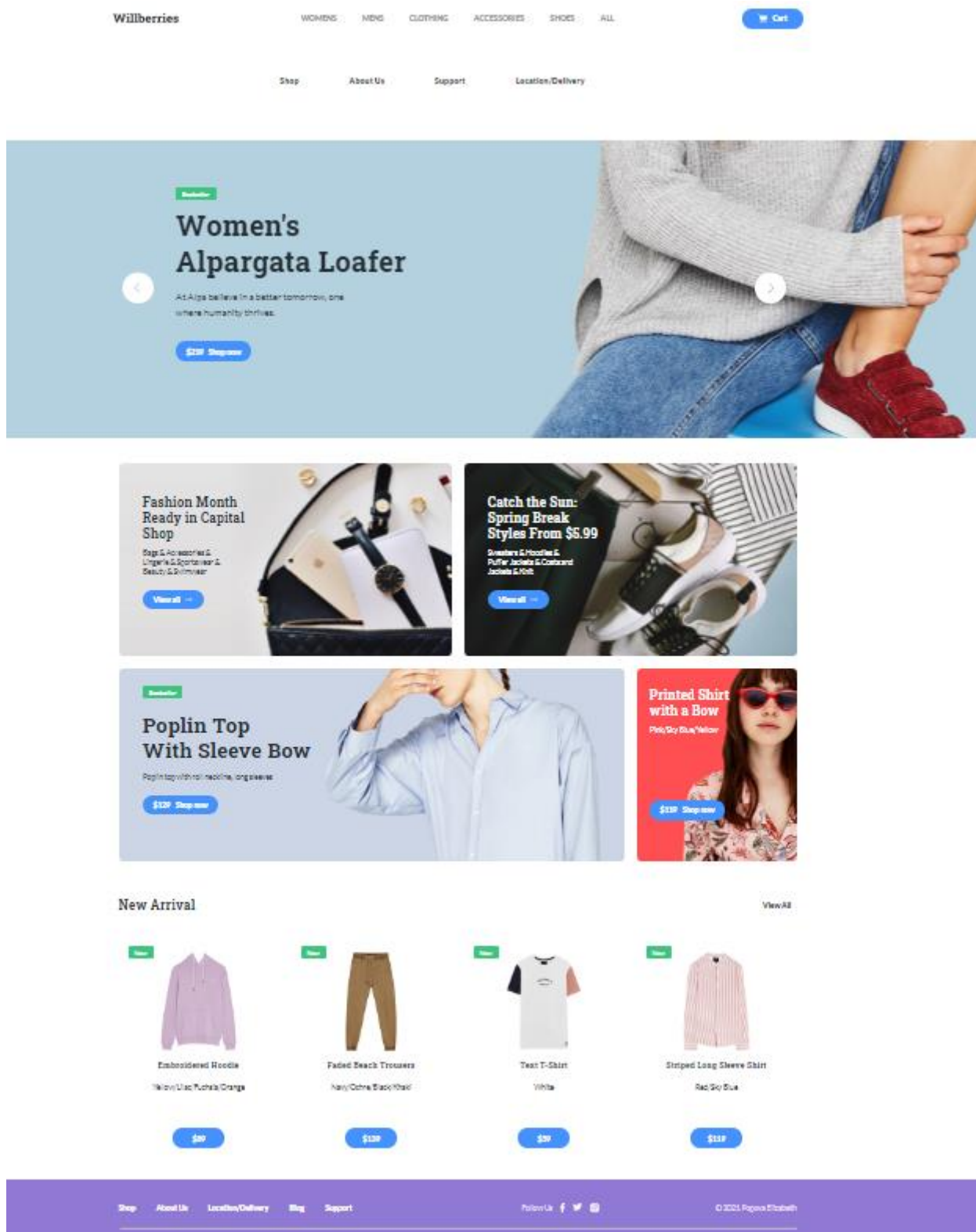


Рис. 2.12. Загальний вигляд головної сторінки інтернет-магазину

Працюючи з модальним вікном кошику і форми замовлення, користувач може повернутися на головну сторінку, або продовжити покупку, і



заповнивши інформаційні поля форми, здійснити замовлення товарів (рис. 2.13).

При вдалому оформленні покупки, виводиться модальне вікно з характерною інформацією, щодо успішного замовлення.

При невдалому оформленні – модальне вікно з характерною інформацією щодо помилки. При закритті модального вікна оформлення покупки, користувач автоматично опиняється на головній сторінці.



Рис. 2.13. Корзина і форма замовлення

Коли користувач знаходиться на сторінці товару, він має змогу переглянути характеристики товару більш детально, та повернутися на головну сторінку, щоб додати цей товар в корзину або ж для перегляду чи вибору інших товарів (рис. 2.14).

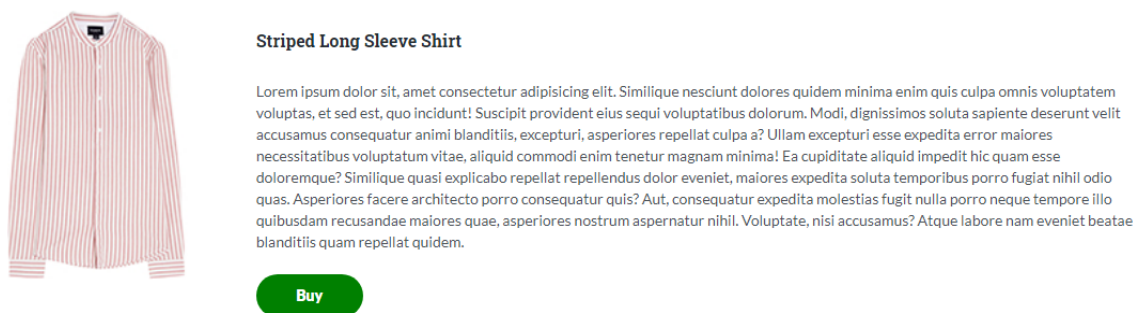


Рис. 2.14. Сторінка товару

Переходячи на сторінку "Про Нас" користувач може отримати більш детальну інформацію щодо організації магазину: скільки років існує, які товари і послуги надає, про їхню якість, а також про політику і стратегії інтернет-магазину (рис. 2.15). При бажанні користувач може натиснути кнопку «графіку роботи колл центру», у верхньому правому куті сторінки, і перейти на сторінку контактної інформації call-center (рис. 2.16).

Willberries

WOMENS

MENS

CLOTHING

ACCESSORIES

SHOES

ALL

Call-centre work  
schedule

Shop

About Us

Contacts

Location



**Lorem ipsum dolor sit amet.**

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Alias at consectetur dicta distinctio dolores eligendi explicabo hic id ipsa iste maxime nam, natus, nihil odit officii perspicatis placeat sed tenetur totam unde? Doloribus ex placeat possimus ullam! Assumenda eius et exercitationem id itaque laborum omnis praesentium sint veniam vitae, voluptatibus.

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Alias at consectetur dicta distinctio dolores eligendi explicabo hic id ipsa iste maxime nam, natus, nihil odit officii perspicatis placeat sed tenetur totam unde? Doloribus ex placeat possimus ullam! Assumenda eius et exercitationem id itaque laborum omnis praesentium sint veniam vitae, voluptatibus.

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Alias at consectetur dicta distinctio dolores eligendi explicabo hic id ipsa iste maxime nam, natus, nihil odit officii perspicatis placeat sed tenetur totam unde? Doloribus ex placeat possimus ullam! Assumenda eius et exercitationem id itaque laborum omnis praesentium sint veniam vitae, voluptatibus.

Рис. 2.15. Сторінка «Про нас»

## Contact phone numbers

### Consultations & orders

[+38 044 537-02-22](#)

[+38 044 503-80-80](#)

### Other sellers

[+38 044 502-10-79](#)

### Call-centre work schedule

Weekdays: 8:00 - 21:00

Saturday: 9:00 - 20:00

Sunday: 10:00 - 19:00

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Voluptas consequuntur exercitationem adipisci nostrum voluptates error commodi non, illum deleniti labore dolorem laborum sapiente unde sequi officis impedit reiciendis quisquam dignissimos! Debitis amet alias, minima quia sit quibusdam voluptatum voluptatem iure possimus, sed voluptas, odio iusto consequatur. Minima dicta facere est laudantium, necessitatibus beatae eaque doloribus allquam doloremque explicabo magni quia. Dignissimos tempore eaque commodi, ullam suscipit dolor hic, placeat alias inventore maxime dicta quis ducimus fuga nostrum numquam quam culpa corrupti! At eum perferendis quod expedita corporis nemo error similique. Unde consequatur voluptatibus, commodi provident allquam iure et expedita est porro delectus esse dolor totam ipsam saepe reiciendis, dicta beatae quae? Eveniet ad laboriosam quos molestias nihil, neque parlatur consequatur? Fuga, distinctio reprehenderit! Explicabo illum dolorum cum quod eveniet nam non. Excepturi molestias saepe soluta enim necessitatibus vitae nulla id? Rem blanditilis explicabo iusto recusandae aliquid modi aut animi dolorem.

Рис. 2.16. Сторінка контактної інформації

Якщо користувачеві знадобиться допомога у різного роду питаннях стосовно роботи сайту, він може перейти на сторінку підтримки, і заповнивши відповідні поля форми, з ним зв'яжеться відповідальна за це людина, щоб надати консультацію щодо питання користувача (рис. 2.17). На цій сторінці також присутня кнопка «графіку роботи колл центру», по якій, при бажанні, користувач може перейти.

## Support

Name:

Email:

Phone:

Message:

SUBMIT

Рис. 2.17. Сторінка підтримки

Переходячи на сторінку "Місцезнаходження/Доставка" користувач може отримати більш детальну інформацію щодо регіонів, способів та умов здійснення доставки магазином. Переглянути діючі офлайн точки самовивозу товару користувач може за допомогою розміщеної гугл карти (рис. 2.18). Також на цій сторінці присутня кнопка «графіку роботи колл центру», по якій, при бажанні, користувач може перейти.

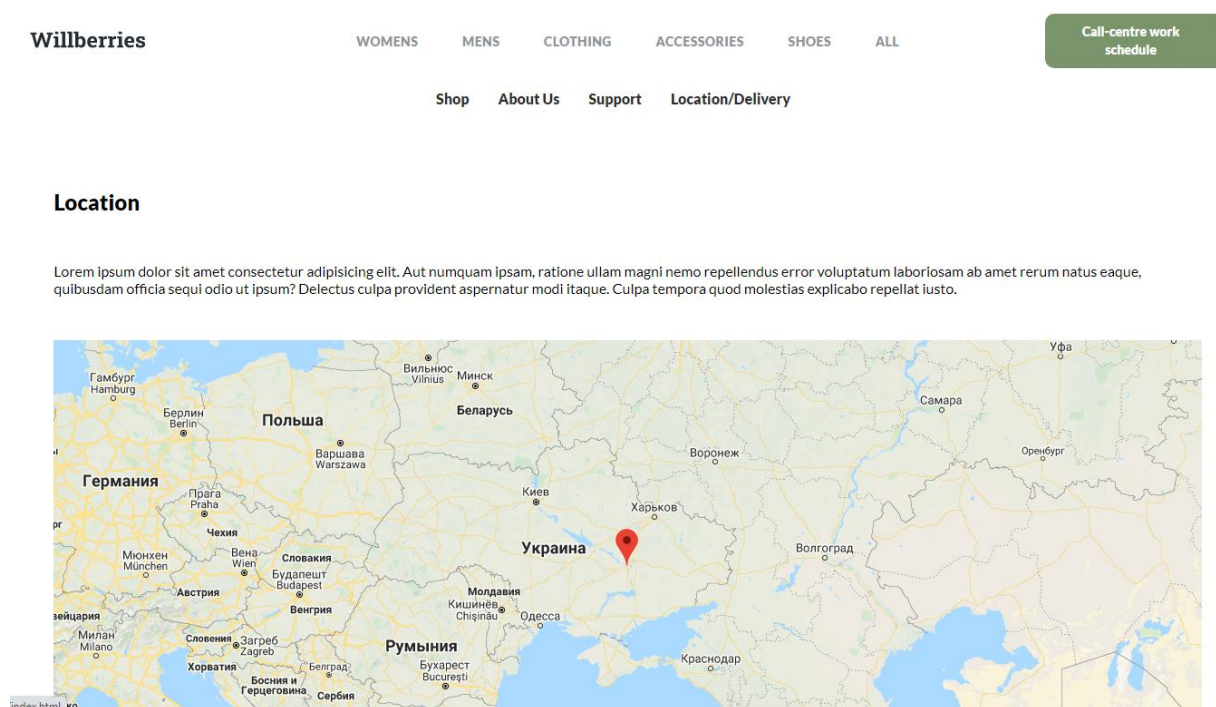


Рис. 2.18. Сторінка доставки

## РОЗДІЛ 3

### ЕКОНОМІЧНИЙ РОЗДІЛ

При розробці програмного забезпечення важливими етапами є визначення трудомісткості розробки ПЗ і розрахунок витрат на створення програмного продукту.

#### **3.1. Визначення трудомісткості розробки програмного забезпечення**

Задані дані:

1. передбачуване число операторів – 2774;
2. коефіцієнт складності програми – 1,7;
3. коефіцієнт корекції програми в ході її розробки – 0,05;
4. годинна заробітна плата програміста, грн/год – 125;

Середня годинна зарплата Junior FE developer в Україні була вираховувати виходячи з даних «Української спільноти програмістів (DOU)» [20]. Станом на кінець 2020 року зарплата Junior FE розробника простягається від 500\$ до 1100\$. Вирахувавши середню заробітну плату програміста маємо плату 800\$ у місяць. При курсі валют НБУ на початок червня 2021 року один американський долар дорівнює 27,34 грн, тому середня зарплата в гривнях дорівнює 21920 грн. При стандартному графіку (176 годин/місяць) зарплата за годину буде становити близько 125 грн.

5. коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі – 1,2;
6. коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності – 1,4;
7. вартість машино-години ЕОМ – 21 грн/год ;

Оскільки для цього проекту потрібна велика потужність ПК для back-end та підняття великої кількості даних на локальному сервері, гарним рішенням буде оренда комп'ютера на час розробки додатку.

Вартість оренди комп'ютера на місяць 1300 грн (монітор) та 2400 грн (системний блок). Загалом на місяць оренда коштуватиме 3700 грн. При стандартному графіку (176 годин/місяць ) вартість машино-години ЕОМ за годину роботи буде становити 21 грн. В цю вартість входить ремонт за гарантією та базовий комплект гарнітури (клавіатура та миша) [21].

Нормування праці в процесі створення ПЗ істотно ускладнено в силу творчого характеру праці програміста. Тому трудомісткість розробки ПЗ може бути розрахована на основі системи моделей з різною точністю оцінки.

Трудомісткість розробки ПЗ можна розрахувати за формулою:

$$t = t_o + t_u + t_a + t_n + t_{oml} + t_d, \text{ людино-годин,} \quad (3.1)$$

де  $t_o$  – витрати праці на підготовку й опис поставленої задачі (приймається 50);

$t_u$  – витрати праці на дослідження алгоритму рішення задачі;

$t_a$  – витрати праці на розробку блок-схеми алгоритму;

$t_n$  – витрати праці на програмування по готовій блок-схемі;

$t_{oml}$  – витрати праці на налагодження програми на ЕОМ;

$t_d$  – витрати праці на підготовку документації.

Складові витрати праці визначаються через умовне число операторів у ПЗ, яке розробляється.

Умовне число операторів (підпрограм):

$$Q = q * C * (1 + p), \quad (3.2)$$

де  $q$  – передбачуване число операторів;

$C$  – коефіцієнт складності програми;

$p$  – коефіцієнт корекції програми в ході її розробки.

$$Q = 2774 * 1,7 * (1 + 0,05) = 4952,6 \text{ людино-годин} \quad (3.3)$$

Витрати праці на вивчення опису задачі  $t_u$  визначається з урахуванням уточнення опису і кваліфікації програміста:

$$t_u = \frac{Q*B}{(75..85)*k}, \text{ людино-годин,} \quad (3.4)$$

де  $B$  – коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі. Прийmemo збільшення витрат праці внаслідок недостатнього опису завдання не більше 50% ( $B = 1,2$ );

$k$  – коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності. При стажі роботи від 5 до 8 років він складає 1,4.

$$t_u = \frac{4952,6*1,2}{75*1,4} = \frac{5943,12}{105} = 56,6 \text{ людино-годин} \quad (3.5)$$

Витрати праці на розробку алгоритму рішення задачі:

$$t_a = \frac{Q}{(20..25)*k}, \text{ людино-годин.} \quad (3.6)$$

$$t_a = \frac{4952,6}{23*1,4} = 153,8 \text{ людино-годин} \quad (3.7)$$

Витрати на складання програми по готовій блок-схемі:

$$t_n = \frac{Q}{(20..25)*k}, \text{ людино-годин.} \quad (3.8)$$

$$t_n = \frac{4952,6}{21*1,4} = 168,45 \text{ людино-годин} \quad (3.9)$$

Витрати праці на налагодження програми на ЕОМ:

- за умови автономного налагодження одного завдання:

$$t_{\text{отл}} = \frac{Q}{(4..5)*k}, \text{ ЛЮДИНО-ГОДИН.} \quad (3.10)$$

$$t_{\text{отл}} = \frac{4952,6}{5*1,4} = 707,5 \text{ ЛЮДИНО-ГОДИН} \quad (3.11)$$

- за умови комплексного налагодження завдання:

$$t_{\text{отл}}^k = 1,5 * t_{\text{отл}}, \text{ ЛЮДИНО-ГОДИН.} \quad (3.12)$$

$$t_{\text{отл}}^k = 1,5 * 707,5 = 1061,25 \text{ ЛЮДИНО-ГОДИН} \quad (3.13)$$

Витрати праці на підготовку документації:

$$t_{\delta} = t_{\delta p} + t_{\delta o}, \text{ ЛЮДИНО-ГОДИН,} \quad (3.14)$$

де  $t_{\delta p}$  – трудомісткість підготовки матеріалів і рукопису.

$$t_{\delta p} = \frac{Q}{15..20*k}, \text{ ЛЮДИНО-ГОДИН.} \quad (3.15)$$

$$t_{\delta p} = \frac{4952,6}{18*1,4} = 196,5 \text{ ЛЮДИНО-ГОДИН} \quad (3.16)$$

$t_{\delta o}$  – трудомісткість редагування, печатки й оформлення документації

$$t_{\delta o} = 0,75 * t_{\delta p}, \text{ ЛЮДИНО-ГОДИН.} \quad (3.17)$$

$$t_{\delta o} = 0,75 * 196,5 = 147,375 \text{ ЛЮДИНО-ГОДИН} \quad (3.18)$$

$$t_{\delta} = 196,5 + 147,375 = 343,9 \text{ ЛЮДИНО-ГОДИН} \quad (3.19)$$



Тепер розрахуємо трудомісткість ПЗ:

$$t = 50 + 56,6 + 153,8 + 168,45 + 707,5 + 343,9 = 1480, \quad \text{людино-годин.} \quad (3.20)$$

### 3.2. Витрати на створення програмного забезпечення

Витрати на створення ПЗ  $K_{\text{по}}$  включають витрати на заробітну плату виконавця програми  $Z_{\text{зп}}$  і витрат машинного часу, необхідного на налагодження програми на ЕОМ.

$$K_{\text{по}} = Z_{\text{зп}} + Z_{\text{мв}}, \text{ грн.} \quad (3.21)$$

Заробітна плата виконавців визначається за формулою:

$$Z_{\text{зп}} = t * C_{\text{пр}}, \text{ грн,} \quad (3.22)$$

де:  $t$  – загальна трудомісткість, людино-годин;

$C_{\text{пр}}$  – середня годинна заробітна плата програміста, грн/година

$$Z_{\text{зп}} = 1480 * 125 = 185\,000, \text{ грн.} \quad (3.23)$$

Вартість машинного часу, необхідного для налагодження програми на ЕОМ:

$$Z_{\text{мв}} = t_{\text{отл}} * C_{\text{мч}}, \text{ грн,} \quad (3.24)$$

де  $t_{\text{отл}}$  – трудомісткість налагодження програми на ЕОМ, год.

$$Z_{\text{мв}} = 707,5 * 21 = 14\,858, \text{ грн.} \quad (3.25)$$

Визначені в такий спосіб витрати на створення програмного забезпечення є частиною одноразових капітальних витрат на створення АСУП.

$$K_{\text{по}} = 185\,000 + 14\,858 = 199\,585, \text{ грн.} \quad (3.26)$$

Очікуваний період створення ПЗ:

$$T = \frac{t}{B_k * F_p}, \text{ міс,} \quad (3.27)$$

де  $B_k$  – число виконавців;

$F_p$  – місячний фонд робочого часу (при 40 годинному робочому тижні  $F_p=176$  годин).

$$B_k = 1$$

$$T = \frac{1480}{1 * 176} = 8,4 \text{ міс} \quad (3.28)$$

**Висновок:** програмне забезпечення розроблено для забезпечення доступу користувачів до основних програм та пропозицій компанії продажу одягу та аксесуарів, ефективної взаємодії між потенційними споживачами та компанією, створення позитивного іміджу компанії, підвищення продажу товарів. Вартість даного програмного забезпечення оцінюється в 199 585, грн. Очікуваний час розробки становить – приблизно 8,4 міс. Цей термін пов'язаний зі значним числом операторів, і включає час на дослідження і розробку алгоритму вирішення поставленого завдання, програмування по готовому алгоритму, налагодження програми і підготовку документації.

## ВИСНОВКИ

Під час виконання кваліфікаційної роботи було поставлено завдання розробити кросбраузерну автоматизовану інформаційну систему електронної комерції.

Розроблене програмне забезпечення призначене для надання універсального інструменту для відображення та управління контенту системи електронної комерції. Практичне значення даної системи полягає в забезпеченні відвідувачам сайту простого і комфортного доступу до каталогу товарів за рахунок оптимальних параметрів візуалізації його вмісту, що зробить процес покупки швидким і звучним, і підвищить ефективність роботи інформаційної системи.

В процесі виконання даного проекту були вирішені наступні задачі:

- вивчено предметну галузь розв'язуваної задачі;
- створено алгоритм для реалізації поставленого завдання;
- розглянуті і застосовані сучасні технології та мови програмування;
- вивчено сучасне програмне забезпечення;
- створена база даних та реалізована клієнтська частина;
- проведений аналіз працездатності системи.

Інформаційна система розроблена за допомогою таких мов web-програмування, як: PHP, JavaScript, а також HTML5, Bootstrap, CSS3 та jQuery.

Розроблене програмне забезпечення має ряд відмінних рис:

- сучасний дизайн та інтуїтивно зрозумілий інтерфейс;
- гнучку систему налаштування відображення інформації;
- формування головної сторінки на основі шаблону з використанням контенту з бази даних;
- кросбраузерність розробленої системи;
- розроблений модуль зворотного зв'язку.

Також у кваліфікаційній роботі було визначено трудомісткість розробленої системи, на базі середньої зарплати розробника, і проведений

підрахунок вартості роботи по створенню програми, який складає 199 585 грн, та розраховано час на створення інформаційної системи – 1480 людино-годин, тобто 8,4 місяців.

Виходячи з матеріалів кваліфікаційної роботи можна впевнено сказати, що поставлені завдання, сформульовані у вступі, повністю вирішені. Мета кваліфікаційної роботи досягнута.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Методичні рекомендації до виконання кваліфікаційних робіт бакалаврів напряму підготовки 122 «Комп'ютерні науки / Л.М. Коротенко, О.С. Шевцова; Нац. гірн. ун-т. – Д: НТУ «Дніпровська політехніка», 2018. – 65 с.
2. Електронна комерція в 2021 році. URL: <https://vc.ua/trade/243482-elektronnaya-kommerciya-v-2021-godu-kak-pandemiya-vliyaet-na-onlayn-rynok> (дата звернення: 02.06.2021).
3. Програмування Web-сторінок / С.В. Глушаков, І.А. Жакін, Т.С. Хачиров; Худож.-оформ. А.С. Юхтман Яків Петрович. - М.: ТОВ «Видавництво АСТ»; Харків: «Фоліо», 2003. - 387 с.
4. Борисов Р. И. Постройте профессиональный сайт сами. - СПб., 2013. -304 с.
5. Веллинг Л., Томсон Л. Разработка веб-приложений с помощью PHP и MySQL. - М.: Вильямс - 2014. - 848 с.
6. Дронов В.А. Разработка современных Web-сайтов. - СПб.: БХВ-Петербург, 2013. – 414 с.
7. Мальчук Е.В. HTML и CSS. Самоучитель / Е. В. Мальчук – М.: Вильямс, 2008. – 416 с.
8. Зольников Д.С. PHP 5. Как самостоятельно создать сайт любой сложности. 2 изд. - М.: ИТ Пресс, 2014. - 272 с.
9. Кожемякин А. А. HTML и CSS в примерах. Создание Web-страниц - М.: Альтекс-А, 2014. - 416 с.
10. Федорчук Д. А. Разработка WEB приложений на PHP и MySQL - СПб.: Корона-принт, 2013. – 340 с.
11. Ковальов А., Курдюмов І. та ін. Управління проектом по створенню інтернет-сайту - М.: Паблішер, 2001. - 337 с.
12. Колисниченко Д.Н. Joomla! Руководство пользователя. - М.: Диалектика, 2013. - 256 с.

13. Яргер Р.Дж., Риз, Дж. Кинг. MySQL и mSQL: Базы данных для небольших предприятий и Интернета. - СПб: Символ-Плюс, 2013. -340 с.
14. jQuery. URL: <https://uk.photo-555.com/9399353-what-is-jquery> (дата звернення: 05.06.2021).
15. Bootstrap. URL: <https://itchief.ua/bootstrap/introduction> (дата звернення: 07.06.2021).
16. Git. URL: <https://www.levina.team/sistema-kontrolya-versiy-git> (дата звернення: 07.06.2021).
17. PHP. URL: <https://blog.skillfactory.ua/php-vs-python-что-выбрать/> (дата звернення: 07.06.2021).
18. OpenServer. URL: <https://sysadmin-note.ua/article/obzor-lokalnykh-serverov-wamp-2020-goda> (дата звернення: 07.06.2021).
19. Методичні вказівки з виконання економічного розділу в дипломних проектах студентів спеціальності «Комп'ютерні системи» / О.Г. Вагонова, О.Б. Нікітіна, Н.Н. Романюк; М-во освіти і науки України, ДВНЗ «Нац. гірн. ун- т». – Д.: НГУ, 2013. – 11 с.
20. Вартість оренди ноутбуку почесово. URL: <https://notebooksbu.com/garantiya-3-goda/> (дата звернення: 07.06.2021).
21. Середня заробітна плата Junior FE developer в Україні станом на початок 2021 року. URL: <https://dou.ua/lenta/articles/salary-report-devs-june-2020/> (дата звернення: 07.06.2021).

## КОД ПРОГРАМИ

```

index.html // верстка головної сторінки

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Willberries</title>
  <link rel="preconnect" href="https://fonts.gstatic.com">
  <link rel="apple-touch-icon" sizes="180x180" href="favicon/apple-touch-icon.png">
  <link rel="icon" type="image/png" sizes="32x32" href="favicon/favicon-32x32.png">
  <link rel="icon" type="image/png" sizes="16x16" href="favicon/favicon-16x16.png">
  <link rel="mask-icon" href="favicon/safari-pinned-tab.svg" color="#5bbad5">
  <meta name="msapplication-TileColor" content="#da532c">
  <meta name="theme-color" content="#ffffff">
  <link href="https://fonts.googleapis.com/css2?family=Lato:wght@400;900&display=swap" rel="styleshee
t">
  <link href="https://fonts.googleapis.com/css2?family=Roboto+Slab:wght@700&display=swap" rel="style
sheet">
  <link rel="stylesheet" href="css/bootstrap-grid.min.css">
  <link rel="stylesheet" href="css/swiper-bundle.min.css"/>
  <link rel="stylesheet" href="css/style.css">

</head>
<!-- class="show-goods" -->
<body id="body">
  <header class="container header px-4 px-md-0">
    <div class="row justify-content-between align-items-center">
      <div class="col-lg-2 col-6">
        <a href="index.html" class="logo-link">
          
        </a>
      </div>
      <div class="col-lg-6 d-none d-lg-block">
        <nav>
          <ul class="navigation d-flex justify-content-around">
            <li class="navigation-item">
              <a href="#" class="navigation-link" data-field="gender">Womens</a>
            </li>
            <li class="navigation-item">
              <a href="#" class="navigation-link" data-field="gender">Mens</a>
            </li>
            <li class="navigation-item">
              <a href="#" class="navigation-link" data-field="category">Clothing</a>
            </li>
            <li class="navigation-item">
              <a href="#" class="navigation-link" data-field="category">Accessories</a>
            </li>
          </ul>
        </nav>
      </div>
    </div>
  </header>

```

```

        </li>
        <li class="navigation-item">
            <a href="#" class="navigation-link" data-field="category">Shoes</a>
        </li>
        <li class="navigation-item">
            <a href="#" class="navigation-link view-all" >All</a>
        </li>

    </ul>
</nav>
</div>
<div class="col-lg-2 col-6 d-flex justify-content-end">
    <button class="button button-cart">
        
        <span class="button-text">Cart</span>
        <span class="button-text cart-count"></span>
    </button>
</div>
</div>
<!-- /.row -->
<div class="menu__container">
    <div class="menu__body wrap">
        <nav class="menu_nav">
            <a href="index.html" class="menu_nav-item">Shop</a>
            <a href="aboutUs.html" class="menu_nav-item">About Us</a>
            <a href="support.html" class="menu_nav-item">Support</a>
            <a href="location.html" class="menu_nav-item">Location/Delivery</a>
            <a href="#" class="menu_nav-item icon">&#9776;</a>
        </nav>
    </div>
</div>
</header>
<section class="slider swiper-container col-lg-10">
    <div class="swiper-wrapper">
        <section class="slide slide-1 swiper-slide">
            <div class="container">
                <div class="row">
                    <div class="col-lg-4 col-10 offset-lg-1">
                        <span class="label">Bestseller</span>
                        <h2 class="slide-title">Women's Alpargata Loafer</h2>
                        <p class="slide-
description">At Alpa believe in a better tomorrow, one where humanity thrives.</p>
                        <button class="button add-to-cart" data-id="003">
                            <span class="button-price">$219</span>
                            <span class="button-text">Shop now</span>
                        </button>
                    </div>
                </div>
            <!-- /.col-4 offset-lg-1 -->
        </div>
    <!-- /.row -->
</div>

```



```

        <!-- /.container -->
    </section>
    <section class="slide slide-2 swiper-slide">
        <div class="container">
            <div class="row">
                <div class="col-lg-4 col-10 offset-lg-1">
                    <span class="label">New</span>
                    <h2 class="slide-title">Text T-Shirt</h2>
                    <p class="slide-
description">Women’s pearl basic knit sweater with a round neck. Available in several colours. Free shipping to sto
res.</p>
                    <button class="button add-to-cart" data-id="004">
                        <span class="button-price">$119</span>
                        <span class="button-text">Shop now</span>
                    </button>
                </div>
                <!-- /.col-4 offset-lg-1 -->
            </div>
            <!-- /.row -->
        </div>
    <!-- /.container -->
</section>
<section class="slide slide-1 swiper-slide">
    <div class="container">
        <div class="row">
            <div class="col-lg-4 col-10 offset-lg-1">
                <span class="label">Bestseller</span>
                <h2 class="slide-title">Sweater Choker Neck</h2>
                <p class="slide-
description">Women’s pearl basic knit sweater with a round neck. Available in several colours. Free shipping to sto
res.</p>
                <button class="button add-to-cart" data-id="005">
                    <span class="button-price">$319</span>
                    <span class="button-text">Shop now</span>
                </button>
            </div>
            <!-- /.col-4 offset-lg-1 -->
        </div>
        <!-- /.row -->
    </div>
    <!-- /.container -->
</section>
</div>
<!-- /.swiper-wrapper -->
<div class="slider-nav">
    <div class="container">
        <div class="row justify-content-between">
            <div class="col-1">
                <button class="slider-button slider-button-prev">
                    
                </button>

```

```

</div>
<!-- /.col-1 -->
<div class="col-1">
  <button class="slider-button slider-button-next">
    
  </button>
</div>
<!-- /.col-1 -->
</div>
<!-- /.row -->
</div>
<!-- /.container -->
</div>
<!-- /.slider-nav -->
</section>
<!-- /.slider -->
<section id="blog" class="special-offers container pt-5 pb-4">
  <div class="row mb-4">
    <div class="col-xl-6">
      <div class="card card-1 mb-4">
        <h3 class="card-title">Fashion Month Ready in Capital Shop</h3>
        <p class="card-text">Bags & Accessories & Lingerie & Sportswear & Beauty & Swimwear</p>
        <button class="button show-accessories">
          <span class="button-text">View all</span>
          
        </button>
      </div>
      <!-- /.card -->
    </div>
    <!-- /.col-6 -->
    <div class="col-xl-6">
      <div class="card card-2 mb-4">
        <h3 class="card-title text-light">Catch the Sun: Spring Break Styles From $5.99</h3>
        <p class="card-text text-
light">Sweaters & Hoodies & Puffer Jackets & Coats and Jackets & Knit</p>
        <button class="button show-clothing">
          <span class="button-text">View all</span>
          
        </button>
      </div>
      <!-- /.card -->
    </div>
    <!-- /.col-6 -->
    <div class="col-xl-9 col-lg-6 mb-4">
      <div class="card card-3">
        <span class="label">Bestseller</span>
        <h3 class="card-title large">Poplin Top <br>With Sleeve Bow</h3>
        <p class="card-text large">Poplin top with roll neckline, long sleeves</p>
        <button class="button add-to-cart" data-id="002">
          <span class="button-price">$129</span>
          <span class="button-text">Shop now</span>
        </button>
      </div>
    </div>
  </div>
</section>

```

```

        </button>
    </div>
    <!-- /.card -->
</div>
<!-- /.col-9 -->
<div class="col-xl-3 col-lg-6">
    <div class="card card-4">
        <h3 class="card-title text-light mw-160">Printed Shirt with a Bow</h3>
        <p class="card-text text-light">Pink/Sky Blue/Yellow</p>
        <button class="button add-to-cart button-four" data-id="009">
            <span class="button-price">$119</span>
            <span class="button-text">Shop now</span>
        </button>
    </div>
    <!-- /.card -->
</div>
<!-- /.col-3 -->
</div>
<!-- /.row -->
<div class="row align-items-center mb-4">
    <div class="col-9">
        <h2 class="section-title">New Arrival</h2>
    </div>
    <!-- /.col-9 -->
    <div class="col-3 d-flex justify-content-end">
        <a href="#" class="more view-all">View All</a>
    </div>
    <!-- /.col-3 -->
</div>
<!-- /.row -->
<div class="short-goods row">
    <div class="col-lg-3 col-sm-6">
        <div class="goods-card">
            <span class="label">New</span>
            <!-- /.label -->
            <h3 class="goods-title">Embroidered Hoodie</h3>
            <!-- /.goods-title -->
            <p class="goods-description">Yellow/Lilac/Fuchsia/Orange</p>
            <!-- /.goods-description -->
            <!-- /.goods-price -->
            <button class="button goods-card-btn add-to-cart" data-id="012">
                <span class="button-price">$89</span>
            </button>
        </div>
        <!-- /.goods-card -->
    </div>
    <!-- /.col-3 -->
    <div class="col-lg-3 col-sm-6">
        <div class="goods-card">
            <span class="label">New</span>

```

```

        <!-- /.label -->
        <h3 class="goods-title">Faded Beach Trousers</h3>
        <!-- /.goods-title -->
        <p class="goods-description">Navy/Ochre/Black/Khaki</p>
        <!-- /.goods-description -->
        <button class="button goods-card-btn add-to-cart" data-id="011">
            <span class="button-price">$139</span>
        </button>
        <!-- /.goods-price -->
    </div>
    <!-- /.goods-card -->
</div>
<!-- /.col-3 -->
<div class="col-lg-3 col-sm-6">
    <div class="goods-card">
        <span class="label">New</span>
        <!-- /.label -->
        <h3 class="goods-title">Text T-Shirt</h3>
        <!-- /.goods-title -->
        <p class="goods-description">White</p>
        <!-- /.goods-description -->
        <button class="button goods-card-btn add-to-cart" data-id="010">
            <span class="button-price">$59</span>
        </button>
        <!-- /.goods-price -->
    </div>
    <!-- /.goods-card -->
</div>
<!-- /.col-3 -->
<div class="col-lg-3 col-sm-6">
    <div class="goods-card">
        <span class="label">New</span>
        <!-- /.label -->
        <h3 class="goods-title">Striped Long Sleeve Shirt</h3>
        <!-- /.goods-title -->
        <p class="goods-description">Red/Sky Blue</p>
        <!-- /.goods-description -->
        <button class="button goods-card-btn add-to-cart" data-id="001">
            <span class="button-price">$119</span>
        </button>
        <!-- /.goods-price -->
    </div>
    <!-- /.goods-card -->
</div>
<!-- /.col-3 -->
</div>
<!-- /.row -->
</section>
<!-- /.container -->

```

```

<section class="long-goods">
  <div class="container">
    <div class="row align-items-center mb-4">
      <div class="col-12">
        <h2 class="section-title">Category</h2>
      </div>
      <!-- /.col-12 -->
    </div>
    <div class="row long-goods-list">
      <div class="col-lg-3 col-sm-6">
        <div class="goods-card">
          <span class="label">New</span>
          <!-- /.label -->
          <h3 class="goods-title">Embroidered Hoodie</h3>
          <!-- /.goods-title -->
          <p class="goods-description">Yellow/Lilac/Fuchsia/Orange</p>
          <!-- /.goods-description -->
          <!-- /.goods-price -->
          <button class="button goods-card-btn add-to-cart" data-id="007">
            <span class="button-price">$89</span>
          </button>
        </div>
        <!-- /.goods-card -->
      </div>
      <!-- /.col-3 -->
      <div class="col-lg-3 col-sm-6">
        <div class="goods-card">
          <span class="label">New</span>
          <!-- /.label -->
          <h3 class="goods-title">Faded Beach Trousers</h3>
          <!-- /.goods-title -->
          <p class="goods-description">Navy/Ochre/Black/Khaki</p>
          <!-- /.goods-description -->
          <button class="button goods-card-btn add-to-cart" data-id="007">
            <span class="button-price">$139</span>
          </button>
          <!-- /.goods-price -->
        </div>
        <!-- /.goods-card -->
      </div>
      <!-- /.col-3 -->
      <div class="col-lg-3 col-sm-6">
        <div class="goods-card">
          <span class="label">New</span>
          <!-- /.label -->
          <h3 class="goods-title">Text T-Shirt</h3>
          <!-- /.goods-title -->
          <p class="goods-description">White</p>
          <!-- /.goods-description -->

```

```

        <button class="button goods-card-btn add-to-cart" data-id="007">
            <span class="button-price">$59</span>
        </button>
        <!-- /.goods-price -->
    </div>
    <!-- /.goods-card -->
</div>
<!-- /.col-3 -->
<div class="col-lg-3 col-sm-6">
    <div class="goods-card">
        <span class="label">New</span>
        <!-- /.label -->
        <h3 class="goods-title">Striped Long Sleeve Shirt</h3>
        <!-- /.goods-title -->
        <p class="goods-description">Red/Sky Blue</p>
        <!-- /.goods-description -->
        <button class="button goods-card-btn add-to-cart" data-id="001">
            <span class="button-price">$119</span>
        </button>
        <!-- /.goods-price -->
    </div>
    <!-- /.goods-card -->
</div>
<!-- /.col-3 -->
<div class="col-lg-3 col-sm-6">
    <div class="goods-card">
        <span class="label">New</span>
        <!-- /.label -->
        <h3 class="goods-title">Embroidered Hoodie</h3>
        <!-- /.goods-title -->
        <p class="goods-description">Yellow/Lilac/Fuchsia/Orange</p>
        <!-- /.goods-description -->
        <!-- /.goods-price -->
        <button class="button goods-card-btn add-to-cart" data-id="007">
            <span class="button-price">$89</span>
        </button>
    </div>
    <!-- /.goods-card -->
</div>
<!-- /.col-3 -->
<div class="col-lg-3 col-sm-6">
    <div class="goods-card">
        <span class="label">New</span>
        <!-- /.label -->
        <h3 class="goods-title">Faded Beach Trousers</h3>
        <!-- /.goods-title -->
        <p class="goods-description">Navy/Ochre/Black/Khaki</p>
        <!-- /.goods-description -->
        <button class="button goods-card-btn add-to-cart" data-id="007">

```

```

        <span class="button-price">$139</span>
    </button>
    <!-- /.goods-price -->
</div>
<!-- /.goods-card -->
</div>
<!-- /.col-3 -->
<div class="col-lg-3 col-sm-6">
    <div class="goods-card">
        <span class="label">New</span>
        <!-- /.label -->
        <h3 class="goods-title">Text T-Shirt</h3>
        <!-- /.goods-title -->
        <p class="goods-description">White</p>
        <!-- /.goods-description -->
        <button class="button goods-card-btn add-to-cart" data-id="007">
            <span class="button-price">$59</span>
        </button>
        <!-- /.goods-price -->
    </div>
    <!-- /.goods-card -->
</div>
<!-- /.col-3 -->
<div class="col-lg-3 col-sm-6">
    <div class="goods-card">
        <span class="label">New</span>
        <!-- /.label -->
        <h3 class="goods-title">Striped Long Sleeve Shirt</h3>
        <!-- /.goods-title -->
        <p class="goods-description">Red/Sky Blue</p>
        <!-- /.goods-description -->
        <button class="button goods-card-btn add-to-cart" data-id="001">
            <span class="button-price">$119</span>
        </button>
        <!-- /.goods-price -->
    </div>
    <!-- /.goods-card -->
</div>
<!-- /.col-3 -->
<div class="col-lg-3 col-sm-6">
    <div class="goods-card">
        <span class="label">New</span>
        <!-- /.label -->
        <h3 class="goods-title">Embroidered Hoodie</h3>
        <!-- /.goods-title -->
        <p class="goods-description">Yellow/Lilac/Fuchsia/Orange</p>
        <!-- /.goods-description -->
        <!-- /.goods-price -->
        <button class="button goods-card-btn add-to-cart" data-id="007">

```

```

        <span class="button-price">$89</span>
    </button>
</div>
<!-- /.goods-card -->
</div>
<!-- /.col-3 -->
<div class="col-lg-3 col-sm-6">
    <div class="goods-card">
        <span class="label">New</span>
        <!-- /.label -->
        <h3 class="goods-title">Faded Beach Trousers</h3>
        <!-- /.goods-title -->
        <p class="goods-description">Navy/Ochre/Black/Khaki</p>
        <!-- /.goods-description -->
        <button class="button goods-card-btn add-to-cart" data-id="007">
            <span class="button-price">$139</span>
        </button>
        <!-- /.goods-price -->
    </div>
    <!-- /.goods-card -->
</div>
<!-- /.col-3 -->
<div class="col-lg-3 col-sm-6">
    <div class="goods-card">
        <span class="label">New</span>
        <!-- /.label -->
        <h3 class="goods-title">Text T-Shirt</h3>
        <!-- /.goods-title -->
        <p class="goods-description">White</p>
        <!-- /.goods-description -->
        <button class="button goods-card-btn add-to-cart" data-id="007">
            <span class="button-price">$59</span>
        </button>
        <!-- /.goods-price -->
    </div>
    <!-- /.goods-card -->
</div>
<!-- /.col-3 -->
<div class="col-lg-3 col-sm-6">
    <div class="goods-card">
        <span class="label">New</span>
        <!-- /.label -->
        <h3 class="goods-title">Striped Long Sleeve Shirt</h3>
        <!-- /.goods-title -->
        <p class="goods-description">Red/Sky Blue</p>
        <!-- /.goods-description -->
        <button class="button goods-card-btn add-to-cart" data-id="001">
            <span class="button-price">$119</span>

```



```

        </button>
        <!-- /.goods-price -->
    </div>
    <!-- /.goods-card -->
</div>
<!-- /.col-3 -->
</div>
</div>

</section>
<footer class="footer">
    <div class="container">
        <div class="row">
            <div class="col-xl-7 mb-4 mb-xl-0">
                <nav>
                    <ul class="footer-menu d-sm-flex">
                        <li class="footer-menu-item"><a href="index.html" class="footer-menu-
link">Shop</a></li>
                        <li class="footer-menu-item"><a href="aboutUs.html" class="footer-menu-
link">About Us</a></li>
                        <li class="footer-menu-item"><a href="location.html" class="footer-menu-
link">Location/Delivery</a></li>
                        <li class="footer-menu-item"><a href="#blog" class="footer-menu-link">Blog</a></li>
                        <li class="footer-menu-item"><a href="support.html" class="footer-menu-
link">Support</a></li>
                    </ul>
                </nav>
            </div>
            <!-- /.col-7 -->
            <div class="col-lg-3 d-flex align-items-center">
                <span class="footer-text">Follow Us</span>
                <span class="footer-social d-inline-flex align-items-center">
                    <a href="#" class="social-link">
                        
                    </a>
                    <a href="#" class="social-link">
                        
                    </a>
                    <a href="#" class="social-link">
                        
                    </a>
                </span>
            </div>
            <!-- /.col-3 -->
            <div class="col-xl-2 col-lg-3 d-flex justify-content-lg-end mt-4 mt-lg-0">
                <span class="footer-text">© 2021 Popova Elizabeth</span>
            </div>
            <!-- /.col-2 -->
        </div>
    <!-- /.row -->
    <hr class="footer-line mt-4 mb-4">

```

```

<div class="row justify-content-between">
  <div class="col-lg-4 col-sm-9">
    
    
    
    
  </div>
  <!-- /.col-4 -->
  <div class="col-lg-2 col-sm-1 d-flex justify-content-end">
    <a href="#body" class="scroll-link top-link d-flex align-items-center">
      <span class="top-link-text">Top</span>
      
    </a>
  </div>
  <!-- /.col-2 -->
</div>
<!-- /.row -->
</div>
<!-- /.container -->
</footer>
<div class="overlay" id="modal-cart">
  <div class="modal">
    <header class="modal-header">
      <h2 class="modal-title">Cart</h2>
      <button class="modal-close">x</button>
    </header>
    <table class="cart-table">
      <colgroup>
        <col class="col-goods">
        <col class="col-price">
        <col class="col-minus">
        <col class="col-qty">
        <col class="col-plus">
        <col class="col-total-price">
        <col class="col-delete">
      </colgroup>
      <thead>
        <tr>
          <th>Good(s)</th>
          <th>Price</th>
          <th colspan="3">Qty.</th>
          <th colspan="2">Total</th>
        </tr>
      </thead>
      <tbody class="cart-table__goods">
        <tr class="cart-item" data-id="003">
          <td>Sweater</td>
          <td>250$</td>
          <td><button class="cart-btn-minus">-</button></td>
          <td>2</td>
          <td><button class="cart-btn-plus">+</button></td>
        </tr>
      </tbody>
    </table>
  </div>
</div>

```

```

        <td>500$</td>
        <td><button class="cart-btn-delete">x</button></td>
    </tr>
    <tr class="cart-item" data-id="003">
        <td>Sweater</td>
        <td>250$</td>
        <td><button class="cart-btn-minus">-</button></td>
        <td>2</td>
        <td><button class="cart-btn-plus">+</button></td>
        <td>500$</td>
        <td><button class="cart-btn-delete">x</button></td>
    </tr>
    <tr class="cart-item" data-id="003">
        <td>Sweater</td>
        <td>250$</td>
        <td><button class="cart-btn-minus">-</button></td>
        <td>2</td>
        <td><button class="cart-btn-plus">+</button></td>
        <td>500$</td>
        <td><button class="cart-btn-delete">x</button></td>
    </tr>
</tbody>
<tfoot>
<tr>
    <th colspan="5">Total:</th>
    <th class="card-table__total" colspan="2">1500$</th>
</tr>
</tfoot>
</table>
<button class="button btn-danger">
    <span class="button-text">Clear All</span>
</button>
<form class="modal-form" action="">
    <input class="modal-input" type="text" placeholder="Имя" name="nameCustomer">
    <input class="modal-input" type="tel" placeholder="Телефон" name="phoneCustomer">
    <button class="button cart-buy" type="submit">
        <span class="button-text">Checkout</span>
    </button>
</form>
</div>
</div>
<script src="js/swiper-bundle.min.js"></script>
<script src="js/jquery-3.6.0.min.js"></script>
<script src="js/main.js" type="module"></script>
<script src="js/inputmask.min.js"></script>
<script src="js/jquery.js"></script>
<script src="js/jquery.inputmask.min.js"></script>
</body>
</html>

```

**aboutUs.html** // верстка інформаційної сторінки

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>AboutUs</title>
  <link href="https://fonts.googleapis.com/css2?family=Lato:wght@400;900&display=swap" rel="stylesheet"
t">
  <link href="https://fonts.googleapis.com/css2?family=Roboto+Slab:wght@700&display=swap" rel="style
sheet">
  <link rel="stylesheet" href="css/bootstrap-grid.min.css">
  <link rel="stylesheet" href="css/style.css">
</head>
<body id="body">

  <header class="container header px-4 px-md-0">
    <div class="row justify-content-between align-items-center">
      <div class="col-lg-2 col-6">
        <a href="index.html" class="logo-link">
          
        </a>
      </div>
      <div class="col-lg-6 d-none d-lg-block">
        <nav>
          <ul class="navigation d-flex justify-content-around">
            <li class="navigation-item">
              <a href="index.html" class="navigation-link" data-field="gender">Womens</a>
            </li>
            <li class="navigation-item">
              <a href="index.html" class="navigation-link" data-field="gender">Mens</a>
            </li>
            <li class="navigation-item">
              <a href="index.html" class="navigation-link" data-field="category">Clothing</a>
            </li>
            <li class="navigation-item">
              <a href="index.html" class="navigation-link" data-field="category">Accessories</a>
            </li>
            <li class="navigation-item">

```

```

        <a href="index.html" class="navigation-link" data-field="category">Shoes</a>
    </li>
    <li class="navigation-item">
        <a href="index.html" class="navigation-link view-all" >All</a>
    </li>
</ul>
</nav>
</div>
<div class="col-lg-2 col-6 d-flex justify-content-end">
    <a class="schedule-link" href="contacts.html" vlink="none">
        <button class="button button-schedule">
            <span class="button-text">Call-centre work schedule</span>
            <span class="button-text cart-count"></span>
        </button>
    </a>
</div>
</div>
<!-- /.row -->
<div class="menu__container">
    <div class="menu__body wrap">
        <nav class="menu_nav">
            <a href="index.html" class="menu_nav-item">Shop</a>
            <a href="aboutUs.html" class="menu_nav-item">About Us</a>
            <a href="support.html" class="menu_nav-item">Support</a>
            <a href="location.html" class="menu_nav-item">Location/Delivery</a>
            <a href="#" class="menu_nav-item icon">&#9776;</a>
        </nav>
    </div>
</div>
</header>
<main class="main__container">
    <div class="about__container">
        <div class="about__images">
            
            
        </div>
        <div class="about__info"></div>
        <h2>Lorem ipsum dolor sit amet.</h2>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Alias at consectetur dicta distinctio do
lores eligendi explicabo hic id ipsa iste maxime nam, natus, nihil odit officiis perspiciatis placeat sed tenetur totam u
nde? Doloribus ex placeat possimus ullam! Assumenda eius et exercitationem id itaque laborum omnis praesentium
sint veniam vitae, voluptatibus.</p>

```

<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Alias at consectetur dicta distinctio do  
lores eligendi explicabo hic id ipsa iste maxime nam, natus, nihil odit officiis perspiciatis placeat sed tenetur totam u  
nde? Doloribus ex placeat possimus ullam! Assumenda eius et exercitationem id itaque laborum omnis praesentium  
sint veniam vitae, voluptatibus.</p>

<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Alias at consectetur dicta distinctio do  
lores eligendi explicabo hic id ipsa iste maxime nam, natus, nihil odit officiis perspiciatis placeat sed tenetur totam u  
nde? Doloribus ex placeat possimus ullam! Assumenda eius et exercitationem id itaque laborum omnis praesentium  
sint veniam vitae, voluptatibus.</p>

</div>

</div>

</main>

<footer class="footer">

<div class="container">

<div class="row">

<div class="col-xl-7 mb-4 mb-xl-0">

<nav>

<ul class="footer-menu d-sm-flex">

<li class="footer-menu-item"><a href="index.html" class="footer-menu-  
link">Shop</a></li>

<li class="footer-menu-item"><a href="#" class="footer-menu-link">About Us</a></li>

<li class="footer-menu-item"><a href="location.html" class="footer-menu-  
link">Location/Delivery</a></li>

<li class="footer-menu-item"><a href="support.html" class="footer-menu-  
link">Support</a></li>

</ul>

</nav>

</div>

<!-- /.col-7 -->

<div class="col-lg-3 d-flex align-items-center">

<span class="footer-text">Follow Us</span>

<span class="footer-social d-inline-flex align-items-center">

<a href="#" class="social-link">



</a>

<a href="#" class="social-link">



</a>

<a href="#" class="social-link">



</a>

</span>

```

    </div>
    <!-- /.col-3 -->
    <div class="col-xl-2 col-lg-3 d-flex justify-content-lg-end mt-4 mt-lg-0">
      <span class="footer-text">© 2021 Popova Elizabeth</span>
    </div>
    <!-- /.col-2 -->
  </div>
<!-- /.row -->
<hr class="footer-line mt-4 mb-4">
<div class="row justify-content-between">
  <div class="col-lg-4 col-sm-9">
    
    
    
    
  </div>
  <!-- /.col-4 -->
  <div class="col-lg-2 col-sm-1 d-flex justify-content-end">
    <a href="#body" class="scroll-link top-link d-flex align-items-center">
      <span class="top-link-text">Top</span>
      
    </a>
  </div>
  <!-- /.col-2 -->
</div>
<!-- /.row -->
</div>
<!-- /.container -->
</footer>
<script src="js/main.js" type="module"></script>
</body>
</html>
main.js // головний файл функцій
import smoothScroll from './module/smoothScroll.js';
document.addEventListener('DOMContentLoaded', () => {
  smoothScroll();

  var mySwiper = new Swiper('.swiper-container', {
    loop: true,

    // Navigation arrows
    navigation: {

```

```

        nextEl: '.slider-button-next',
        prevEl: '.slider-button-prev',
    },
});

// cart

const buttonCart = document.querySelector('.button-cart');
const modalCart = document.querySelector('#modal-cart');
// const modalClose = document.querySelector('.modal-close');

const viewAll = document.querySelectorAll('.view-all');
const navigationLink = document.querySelectorAll('.navigation-link:not(.view-all)');
const longGoodsList = document.querySelector('.long-goods-list');
const showAccessories = document.querySelectorAll('.show-accessories');
const showClothing = document.querySelectorAll('.show-clothing');
const cartTableGoods = document.querySelector('.cart-table__goods');
const cardTableTotal = document.querySelector('.card-table__total');
const cartCount = document.querySelector('.cart-count');
const btnDanger = document.querySelector('.btn-danger');

const getGoods = async function () {
    const result = await fetch('db/db.json');
    if (!result.ok){
        throw 'Ошибка вышла:' + result.status;
    }
    return result.json();
};

const cart = {
    cartGoods: JSON.parse(localStorage.getItem('cart')) || [],
    updateLocalStorage(){
        localStorage.setItem('cart', JSON.stringify(this.cartGoods));
    },
    getCountCartGoods(){
        return this.cartGoods.length;
    },
    countQuantity(){
        const counter = this.cartGoods.reduce((sum, item) => {
            return sum + item.count;
        }, 0)
        if (counter === 0) {

```



```

        cartCount.textContent = "";
    } else {
        cartCount.textContent = counter;
    }
},
clearCart(){
    this.cartGoods.length = 0;
    this.countQuantity();
    this.renderCart();
    this.updateLocalStorage();
},
renderCart(){
    cartTableGoods.textContent = "";
    this.cartGoods.forEach(({ id, name, price, count }) => {
        const trGood = document.createElement('tr');
        trGood.className = 'cart-item';
        trGood.dataset.id = id;

        //(buttons => + data-id="${id}")
        trGood.innerHTML = `
            <td>${name}</td>
            <td>${price}</td>
            <td><button class="cart-btn-minus">-</button></td>
            <td>${count}</td>
            <td><button class="cart-btn-plus">+</button></td>
            <td>${price * count}</td>
            <td><button class="cart-btn-delete">x</button></td>
        `;
        cartTableGoods.append(trGood);
    });

    const totalPrice = this.cartGoods.reduce((sum, item) => {
        return sum + item.price * item.count;
    }, 0);

    cardTableTotal.textContent = totalPrice + '$';
},
deleteGood(id) {
    this.cartGoods = this.cartGoods.filter(item => id !== item.id);
    this.renderCart();
    this.countQuantity();
    this.updateLocalStorage();
}

```

```

    },
    minusGood(id) {
      for (const item of this.cartGoods) {
        if (item.id === id) {
          if (item.count <= 1) {
            this.deleteGood(id);
          } else {
            item.count--;
          }
          break;
        }
      }
      this.renderCart();
      this.updateLocalStorage();
      this.countQuantity();
    },
    plusGood(id) {
      for (const item of this.cartGoods) {
        if (item.id === id) {
          item.count++;
          break;
        }
      }
      this.renderCart();
      this.updateLocalStorage();
      this.countQuantity();
    },
    addCartGoods(id){
      const goodItem = this.cartGoods.find(item => item.id === id);
      if(goodItem){
        this.plusGood(id);
      } else {
        getGoods()
        .then(data => data.find(item => item.id === id))
        .then(({ id, name, price }) => {
          this.cartGoods.push({
            id,
            name,
            price,
            count: 1
          });
        });
        this.updateLocalStorage();
      }
    }
  }
}

```

```

        this.countQuantity();
    });
}

}

}

btnDanger.addEventListener('click', cart.clearCart.bind(cart)); //()=>{cart.clearCart();}

...

style.css // файл стилів
* {
    box-sizing: border-box;
}

body {
    margin: 0;
    font-family: 'Lato', sans-serif;
}

img {
    max-width: 100%;
    height: auto;
}

.header {
    padding: 15px 0;
}

.navigation {
    list-style: none;
    padding: 0;
    margin: 0;
}

.navigation-link {
    text-decoration: none;
    text-transform: uppercase;
    color: #939699;
    font-family: 'Lato', sans-serif;
    font-style: normal;
    font-weight: 900;
    font-size: 14px;
}

```

```
    line-height: 20px;
    letter-spacing: 0.5px;
}

button {
    cursor: pointer;
}

.button {
    padding: 10px 20px;
    background: #4592FF;
    border-radius: 20px;
    border: none;
    color: #fff;
    font-family: 'Lato',
    sans-serif;
    font-style: normal;
    font-weight: 900;
    font-size: 14px;
    line-height: 20px;
    display: flex;
    align-items: center;
}

.button-schedule{
    background: #7a956b;
    border-radius: 10px;
    padding: 10px 20px;
    font-weight: 700;
}

.button-icon, .button-price {
    margin-right: 10px;
}

.button-icon {
    margin-left: 10px;
}

.button-icon:last-child {
    margin-right: 0;
}
```

```

}
.btn-danger{
  padding: 5px 10px;
  background-color: tomato;
  display: block;
  margin-left: auto;
  margin-bottom: 7px;
}
.schedule-link{
  text-decoration: none;
}
}
.cart-count {
  margin-left: 10px;
}

.show-goods .slider,
.show-goods .special-offers {
  display: none;
}

.long-goods {
  display: none;
}

.show-goods .long-goods {
  display: block;
}

.slide {
  padding-top: 90px;
  padding-bottom: 110px;
  min-height: 568px;
}

.slide-1 {
  background: #B4D1E0 url('../img/slide-image-1.png') no-repeat top right 30%;
}

.slide-2 {
  background: #e5e7e6 url('../img/banner1.jpg') no-repeat top right 30%;
}

```

```
.label {  
  background-color: #3DC47E;  
  border-radius: 3px;  
  padding: 5px 12px;  
  color: #fff;  
  font-size: 12px;  
  line-height: 15px;  
}
```

```
.slide-title {  
  color: #262D33;  
  font-size: 55px;  
  line-height: 70px;  
  font-family: 'Roboto Slab',  
  serif;  
  margin-top: 20px;  
  margin-bottom: 15px;  
}
```

```
.slide-description {  
  font-size: 18px;  
  line-height: 30px;  
  color: #4B5157;  
  max-width: 335px;  
  margin-bottom: 40px;  
}
```

```
.slider {  
  position: relative;  
}
```

```
.slider-nav {  
  position: absolute;  
  top: 50%;  
  width: 100%;  
  transform: translateY(-50%);  
  z-index: 9;  
}
```

```
.slider-button {  
  background-color: transparent;
```

```

border: none;
cursor: pointer;
}

.card {
  min-height: 370px;
  border-radius: 10px;
  padding-top: 60px;
  padding-left: 46px;
  padding-right: 46px;
  padding-bottom: 80px;
}

.card-1 {
  background: #E3E2E3 url('../img/card-image-1.jpg') no-repeat top right;
  background-size: cover;
}

.card-2 {
  background: #000 url('../img/card-image-2.jpg') no-repeat top right;
  background-size: cover;
}

.card-3 {
  padding-top: 35px;
  background: #CCD5E6 url('../img/card-image-3.png') no-repeat top right 10%;
  background-size: contain;
}

.card-4 {
  display: flex;
  flex-direction: column;
  align-items: flex-start;
  padding-top: 35px;
  padding-left: 25px;
  background: #FF4F52 url('../img/card-image-4.png') no-repeat top right;
  background-size: contain;
}

```

...

**ВІДГУК**  
**керівника економічного розділу**  
**на кваліфікаційну роботу бакалавра**  
**на тему:**  
**«Розробка кросбраузерної автоматизованої інформаційної системи**  
**електронної комерції»**  
**студентки групи 122-17-1 Попової Єлизавети Сергіївни**

**Керівник економічного розділу**  
**доцент каф. ПЕП та ПУ, к.е.н**

**Л. В. Касьяненко**



## ПЕРЕЛІК ФАЙЛІВ НА ДИСКУ

Ім'я файлу	Опис
Пояснювальні документи	
Диплом_Попова.doc	Пояснювальна записка до кваліфікаційної роботи. Документ Word.
Диплом_Попова.pdf	Пояснювальна записка до кваліфікаційної роботи в форматі PDF.
Програма	
Program.rar	Архів. Містить коди програми і откомпільовану програму
Презентація	
Презентація_Попова.ppt	Презентація кваліфікаційної роботи.