Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Інформаційних управляючих систем
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження методів розроблення вебсайту інтернет-магазину
(тема)

Виконав:
здобувач 2 курсу, групи ІУСТм-23-1
Заполочний Андрій Дмитрович
(прізвище, ім'я, по батькові)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Інформаційні управляючі системи та технології
(повна назва освітньої програми)

Керівник проф. каф. ІУС Віктор ЛЕВИКІН
(посада, власне ім'я, ПРІЗВИЩЕ)

Допускається до захисту

Зав. кафедри _____ Костянтин ПЕТРОВ
(підпис) (власне ім'я, ПРІЗВИЩЕ)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
Кафедра Інформаційних управляючих систем
Рівень вищої освіти другий (магістерський)
Спеціальність 122 Комп'ютерні науки
(код і повна назва)
Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)
Освітня програма Інформаційні управляючі системи та технології
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
« 09 » грудня 20 24 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Заполочному Андрію Дмитровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів розроблення вебсайту інтернет-магазину затверджена наказом університету від " 27 " листопада 2024 р. № 1249Ст.
2. Термін подання здобувачем роботи до екзаменаційної комісії "18" січня 2025 р.
3. Вихідні дані до роботи Звітні матеріали передатестаційної практики, типові структури вебсайту інтернет-магазину, науково-технічна література, інформація з інтернет-ресурсів стосовно підходів і методики розроблення вебсайту інтернет-магазину.
4. Перелік питань, що потрібно опрацювати в роботі Огляд та аналіз особливостей розроблення вебсайту інтернет-магазину, аналіз вимог до розроблення вебсайту інтернет-магазину, опис наявних підходів розроблення вебсайту інтернет-магазину, постановка задач дослідження, побудова методу розроблення вебсайту інтернет-магазину, обґрунтування використання інструментальних засобів, формування критерію оцінювання методу розроблення вебсайту інтернет-магазину, методика використання методу, оцінювання ефективності методу розроблення вебсайту інтернет-магазину, апробація методу розроблення вебсайту інтернет-магазину.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	09.12.24	Виконано
2	Аналіз предметної галузі та постановка задачі	10.12.24 – 15.12.24	Виконано
3	Огляд і аналіз існуючих технологій, методів розроблення вебсайту інтернет-магазину	16.12.24 – 21.12.24	Виконано
4	Дослідження технологій і методів розроблення вебсайту інтернет-магазину	22.12.24 – 02.01.25	Виконано
5	Апробація модифікованого методу розроблення вебсайту інтернет-магазину	03.01.23 – 07.01.25	Виконано
6	Оформлення пояснювальної записки та графічного матеріалу	08.01.25 – 13.01.25	Виконано
7	Перевірка на плагіат	15.01.25	Виконано
8	Попередній захист кваліфікаційної роботи	18.01.25	Виконано

Дата видачі завдання 09 грудня 2024 р.

Здобувач _____
(підпис)

Керівник роботи _____ проф. каф. ІУС Віктор ЛЕВИКІН
(підпис) (посада, власне ім’я, ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи: 72 с., 22 рис., 5 табл., 1 дод., 27 джерел.

БЕЗПЕКА, ВЕБСАЙТ, ЗАХИСТ ВІД ЗАГРОЗ, ІНТЕРНЕТ-МАГАЗИН, МЕТОДИ РОЗРОБЛЕННЯ, ФРЕЙМВОРК.

Об’єктом дослідження є інформаційні системи вебсайтів інтернет-магазинів.

Предметом дослідження є методи розроблення вебсайту інтернет-магазину.

Метою кваліфікаційної роботи є дослідження наявних методів розроблення вебсайту інтернет-магазину та пропозиції щодо модифікованого методу розроблення, який забезпечує вищий рівень безпеки та функціональності.

У роботі було досліджено методи розроблення вебсайту інтернет-магазину, виконано поставку задач дослідження, надано пропозиції стосовно покращення у вигляді видозміненого методу створення вебсайту. Використання цього методу дає змогу поліпшити захист вебсайту від загроз зламу чи викрадення особистих даних клієнтів і компанії. Також проведено апробацію модифікованого методу розроблення вебсайту інтернет-магазину, зважаючи на заходи безпеки.

Кваліфікаційну роботу виконано згідно з методичними вказівками щодо розробки та оформлення магістерської атестаційної роботи за спеціальністю 122 Комп’ютерні науки (освітня програма «Інформаційні управляючі системи та технології» освітньо-кваліфікаційного рівня «магістр») [1] і ДСТУ 3008:2015 [2].

ABSTRACT

Explanatory note to the qualification work: 72 pages, 22 figures, 5 tables, 1 appendix, 27 sources.

DEVELOPMENT METHODS, FRAMEWORK, ONLINE STORE, SECURITY, THREAT PROTECTION, WEBSITE.

The object of the study is information systems of online shopping websites.

The subject of research is methods of developing an online store website.

The purpose of the qualification work is to study the existing methods of developing an online store website and propose a modified development method that provides a higher level of security and functionality.

In this work, the methods of developing an online store website were investigated, the research objectives were completed, and suggestions for improvement were made in the form of a modified method of creating a website. The use of this method makes it possible to improve the protection of the website from threats of hacking or theft of personal data of customers and the company. The modified method of developing an online store website was also tested, taking into account protection against existing threats.

The qualification work was carried out in accordance with the guidelines for the development and execution of a master's qualification work in the specialty 122 Computer Science (educational program "Information Management Systems and Technologies" of the educational qualification level "Master") [1] and DSTU 3008:2015 [2].

ЗМІСТ

Скорочення та умовні познаки……..8
Вступ……9
1. Огляд та аналіз особливостей розроблення вебсайту інтернет-магазину.10
1.1 Огляд та аналіз особливостей розроблення вебсайту інтернет-магазину……10
1.2 Аналіз вимог до розроблення вебсайту інтернет-магазину……13
1.3 Опис наявних підходів розроблення вебсайту інтернет-магазину…16
1.4 Постановка задач дослідження……..23
2. Дослідження та побудова методу розроблення вебсайту інтернет-магазину, зважаючи на заходи безпеки……24
2.1 Дослідження наявних методів розроблення вебсайту інтернет-магазину……24
2.2 Дослідження можливих загроз під час роботи вебсайту інтернет-магазину……31
2.3 Побудова методу розроблення вебсайту інтернет-магазину, зважаючи на заходи безпеки……...35
2.4 Формування критерію ефективності оцінювання рівня захисту вебсайту від загроз……40
3. Методика застосування методу розроблення вебсайту інтернет-магазину, зважаючи на заходи безпеки……42
3.1 Обґрунтування застосування інструментів для забезпечення захисту вебсайту інтернет-магазину від загроз……42
3.2 Методика стосовно застосування методу розроблення вебсайту інтернет-магазину, зважаючи на заходи безпеки……44
4. Апробація методу розроблення вебсайту інтернет-магазину, зважаючи на заходи безпеки……...47
4.1 Опис вимог для побудови вебсайту інтернет-магазину……..47

4.2 Результати апробації методу розроблення вебсайту інтернет-магазину, зважаючи на заходи безпеки…….53
Висновки……56
Перелік джерел посилання……57
Додаток А Графічний матеріал кваліфікаційної роботи…….60

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

БД – база даних

ІС – інформаційна система

ІТ – інформаційні технології

ПЗ – програмне забезпечення

СУБД – система управління базами даних

CMS – Content Management System

CSRF – Cross-Site Request Forgery

DDoS – Distributed Denial-of-Service

DFD – Data Flow Diagram

DoS – Denial-of-Service attack

ER – Entity-Relationship

HTTP – Hypertext Transfer Protocol

MVS – Model-View-Controller

OO – Object-Oriented

SaaS – Software as a Service

XSS – Cross-Site Scripting

ВСТУП

На сьогодні розроблення вебсайтів інтернет-магазинів є важливим аспектом сучасного бізнесу, оскільки інтернет-комерція стрімко зростає і стає невід'ємною складовою світової економіки та повсякденного життя людей. Це зумовлює високі вимоги до функціональності, зручності використання та безпеки онлайн-платформ, оскільки споживачі очікують не лише широкого асортименту товарів, але й швидкого, простого та безпечного процесу покупки, а бізнеси – зростання продажів, залучення нових клієнтів та автоматизацію деяких процесів [3].

Зі свого боку ІТ-компанії мають організувати розроблення ПЗ (програмного забезпечення), аби втиснутись у терміни, що були поставлені, й оптимізувати етапи розроблення, саме в цьому стають у нагоді методи розроблення проєктів. У кожного з них є особливості, набір етапів розроблення, тому конкретний метод підходить до певного типу вебсайту.

Під час обрання методу для розроблення сайту слід дивитися на такі фактори: масштаб проєкту, складність, мета клієнта, бізнес-ризики тощо. Окрім цього, необхідно вибрати легкий та зрозумілий у використанні інструмент для керування проєктами.

1 ОГЛЯД ТА АНАЛІЗ ОСОБЛИВОСТЕЙ РОЗРОБЛЕННЯ ВЕБСАЙТУ ІНТЕРНЕТ-МАГАЗИНУ

1.1 Огляд та аналіз особливостей розроблення вебсайту інтернет-магазину

Зростання популярності інтернету підтверджується значною кількістю додатків розроблених останнім часом. Ці програми підтримують різноманітні галузі, такі як: торгівля, дослідження та медицина. Особливості вебдодатків, зокрема високий зв’язок, постійна еволюція та орієнтація на вміст, безпосередньо впливають на підходи до їхнього розроблення.

Вебдодатки класифікують за їхньою функціональністю, наприклад:

– інформаційні сторінки (з формами даних чи без них);

– додатки, орієнтовані на базу даних (доступ до даних за запитами або використання профільної інформації);

– додатки для бізнес-транзакцій.

Останні, відомі як інтернет-магазини, створюються з метою підвищення прибутку компаній через онлайн-продажі. Їхнє розроблення базується на бізнес-моделях, які враховують мету та вимоги підприємства [4].

Є три основні бізнес-моделі онлайн-магазинів:

1. Рекламна модель – орієнтована на залучення постійної аудиторії, яка може бути широкою або сегментованою. Доступ до аудиторії продається спонсорам або рекламодавцям. Основні завдання:

– залучення користувачів на вебсайт;

– стимулювання повторних візитів;

– продовження часу, який користувачі проводять на вебсайті;

– активне залучення відвідувачів до життя ресурсу (обговорення різних тем, розіграші, опитування);

– мотивація відвідувачів до розвитку і просування сайту.

2. Підтримка наявного бізнесу – використовується компаніями з реальним бізнесом для збільшення клієнтів та продажів через інтернет. Основні завдання:

– реклама компанії, її товарів та послуг;

– надання актуальної та повної інформації клієнтам;

– створення онлайн-платформи для продажу.

3. Створення нового бізнесу – базується на використанні можливостей інтернету для запуску нових платформ: інтернет-магазинів, торгових майданчиків, сайтів для аукціонів, сервісів, різноманітних бірж та ін.

Усяка модель може мати підмоделі або поєднувати елементи кількох моделей залежно від специфіки бізнесу.

Інтернет-магазини класифікують за такими критеріями:

1. Види роздрібних продажів:

– онлайн-магазини;

– вебвітрини;

– торговельні системи;

– торговельні ряди.

2. Бізнес-модель:

– повністю онлайн-магазини;

– комбіновані моделі (поєднання офлайн- і онлайн-бізнесу).

3. Відносини з постачальниками:

– магазини зі своїм складом;

– магазини, які працюють відповідно до договору з постачальниками.

4. Рівень автоматизації:

– інтернет-магазини;

– вебвітрини;

– торговельні онлайн-системи.

Вебвітрина – це перелік товарів із системою навігації та формуванням замовлень, яке передається менеджеру для подальшої обробки (зв’язок зі складом, організація доставки, контроль оплати).

Інтернет-магазин забезпечує повну автоматизацію оброблення замовлень і дає можливість працювати особисто з кожним відвідувачем.

Торговельні онлайн-системи поєднують можливості інтернет-магазинів з інтеграцією у внутрішній обіг документів організації, забезпечуючи повний торговий цикл. Системи доставки залишаються неавтоматизованими.

Характерну схему функціонування торговельної компанії наведено на рисунку 1.1.

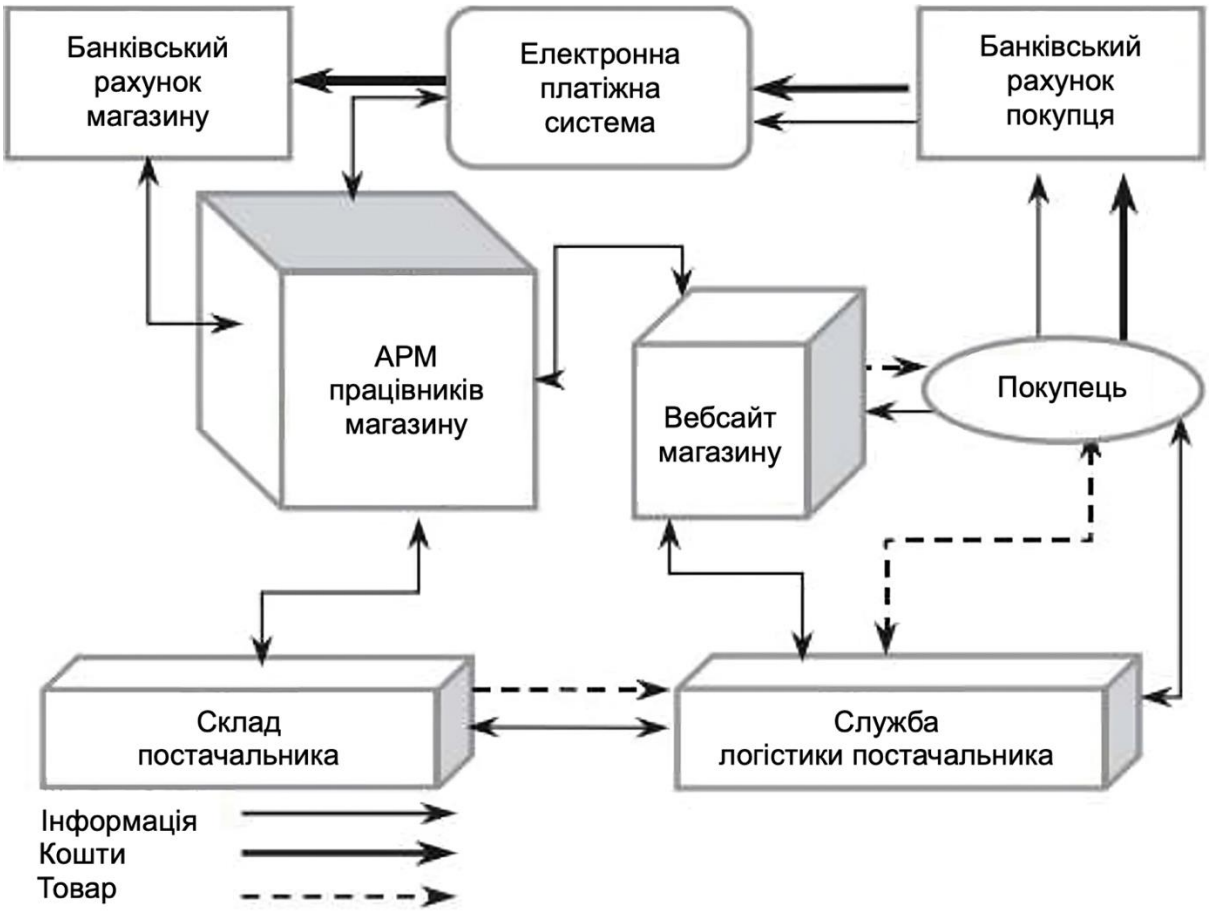

Рисунок 1.1 – Характерна схема функціонування торговельної компанії

Однією з ключових вимог до інформаційних систем (ІС) є забезпечення захисту приватних даних клієнтів і самої компанії, адже сучасні технології відкривають безліч можливостей зламу. Щоб уникнути несанкціонованого доступу до системи та інформації, компанії впроваджують різні захисні функції, які охоплюють усі компоненти ІС, зокрема вебсайти.

1.2 Аналіз вимог до розроблення вебсайту інтернет-магазину

Розроблення вебсайту складається з кількох важливих етапів, більшість із яких виходять за межі написання коду. Хоча такі технології, як HTML, CSS і JavaScript, визначають зовнішній вигляд та функціональність вебсайту, сам процес створення починається задовго до програмування. Він включає збір даних, планування, проєктування, тестування, обслуговування після запуску тощо [5].

Перший етап розроблення вебсайту для інтернет-магазину – це збір інформації. Він визначає головні цілі, які має виконувати сайт, його аудиторію та основну функціональність. Це допомагає розробити кращу стратегію майбутнього керування проєктом та уникнути можливих проблем на подальших етапах.

Другий етап – планування. Розробляється карта сайту, що описує структуру та взаємозв’язок сторінок. Важливий етап для уникнення непередбачених витрат на внесення змін. На рисунку 1.2 наведено приклад мапи вебсайту інтернет-магазину.

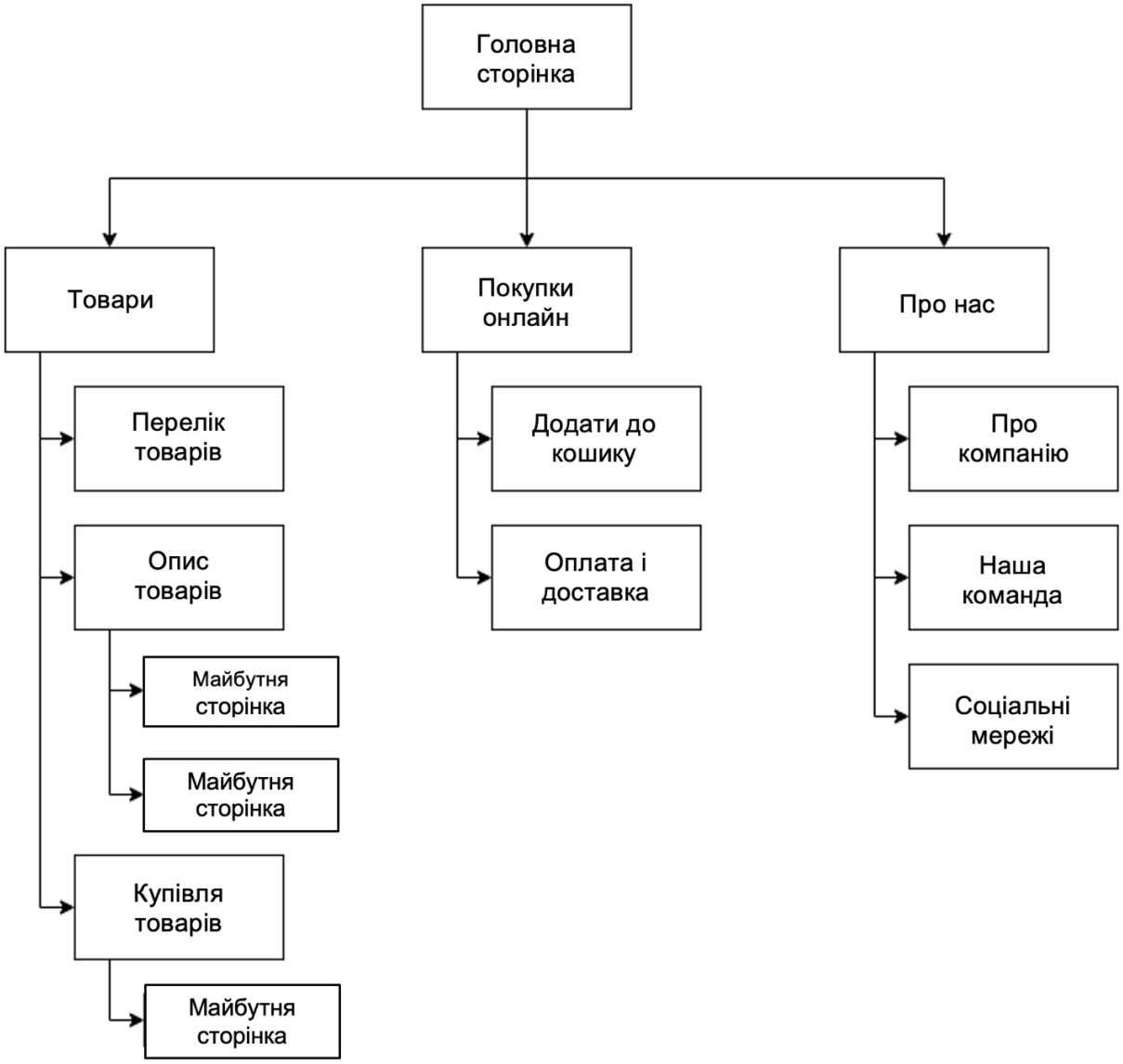

Рисунок 1.2 – Приклад мапи вебсайту інтернет-магазину

Проєктування. На цьому етапі розроблюються візуальні елементи сайту, такі як макети, графіка, кольори та логотипи. Вони мають відповідати уподобанням цільової аудиторії.

Наступний етап – написання та підготовка контенту. Контент формує основу для комунікації з аудиторією. Це включає в себе тексти, заголовки, зображення та заклики до дії. Найкраще, якщо вміст готовий до моменту початку кодування.

Далі йде написання коду. Під час цього етапу сайт перетворюється з макетів у функціональний ресурс. Впроваджуються CMS та інші системи, необхідні для роботи, а також початкові механізми захисту.

Тестування. Перевіряється кожна функція сайту, форми, посилання та захист. Це важливий етап для забезпечення надійної роботи та відповідності вебстандартам. Схему етапів розроблення вебсайту інтернет-магазину показано на рисунку 1.3.

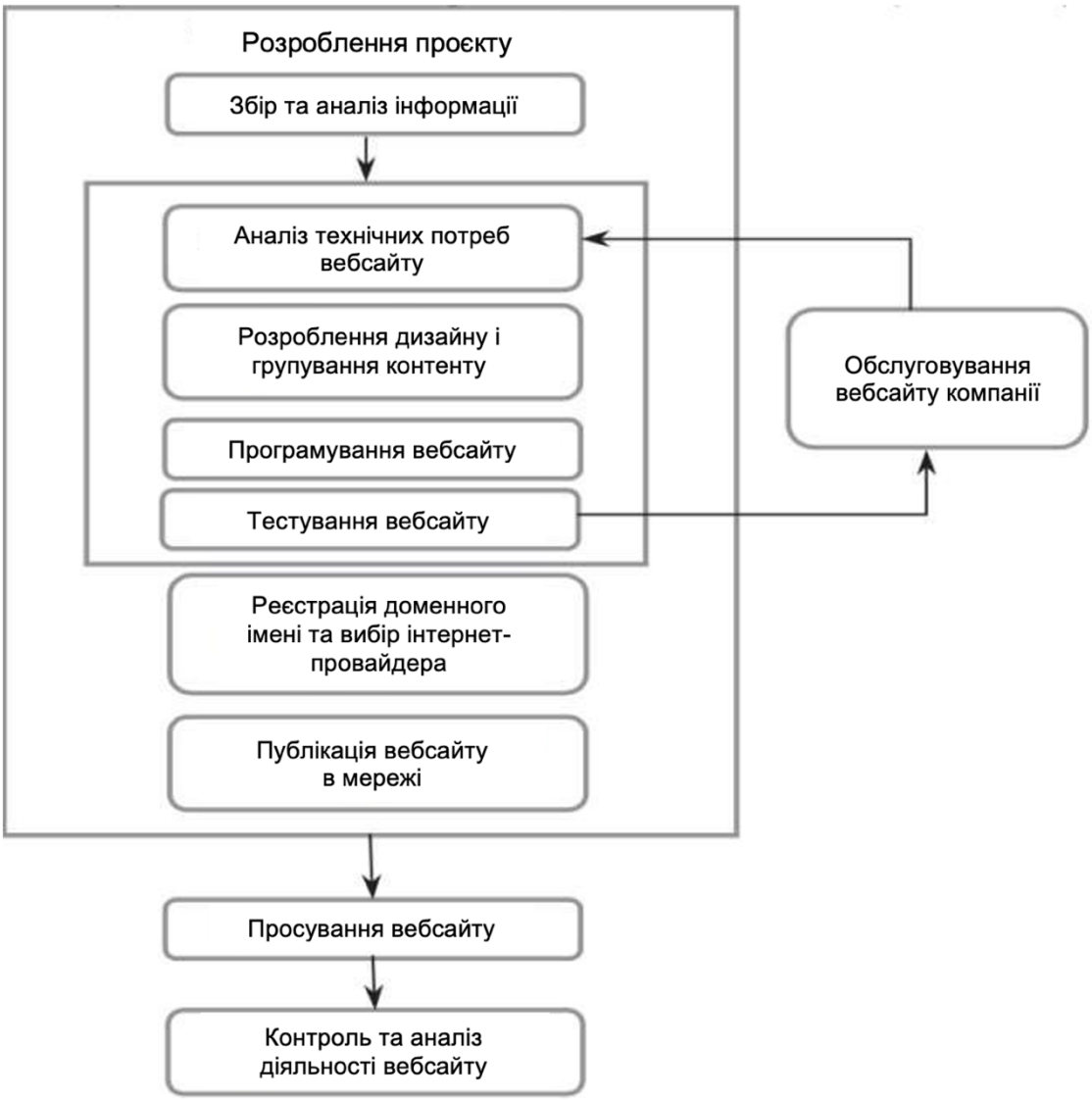

Рисунок 1.3 – Схема етапів розроблення вебсайту інтернет-магазину

Підтримка та обслуговування. Регулярне оновлення контенту та CMS дає змогу зберігати актуальність і безпеку сайту, мінімізуючи ризики технічних збоїв та атак.

Ретельна робота на кожному з цих етапів забезпечує створення ефективного, функціонального та безпечного вебсайту, адаптованого до потреб замовника.

1.3 Опис наявних підходів розроблення вебсайту інтернет-магазину

Є чимало підходів до розроблення вебсайту. Коректний вибір способу розроблення залежить від вимог до самого вебсайту. В ІС компаній нерідко вже наявні готові компоненти, як-от системи постачання або системи оплати. Визначають три основні підходи до розроблення вебсайту [6]:

– із допомогою CMS;

– із застосуванням сучасних фреймворків;

– із застосуванням платформ SaaS.

CMS (Content Management System), або система управління контентом, – це програмний інструмент, що автоматизує процеси створення, редагування, організації та публікації контенту на вебсайті.

Основні характеристики CMS:

– платформа для багатьох користувачів. CMS працює на сервері, забезпечуючи доступ декільком користувачам. Контент зберігається у репозиторії, який може бути частиною серверного програмного пакету або окремим сховищем;

– функціональність для редакторів. CMS дає можливість створювати та змінювати вміст, організовувати процеси редагування і робити його загальнодоступним;

– компонентна структура. Система складається з кількох складових: редагувальний інтерфейс, сховище, механізми публікування та ін. Хоча ці елементи можуть працювати окремо на технічному рівні, для кінцевих користувачів CMS виглядає як одне ціле.

CMS зазвичай написані на PHP, що робить їх сумісними з багатьма існуючими системами компаній. Це зменшує час і економить кошти на інтеграцію сайту в ІС торговельної компанії [7].

Найбільш використовувані системи управління контентом сьогодні включають:

– WordPress – ідеальна для блогів, малих і середніх бізнесів;

– Drupal – підходить для складних і масштабних проєктів;

– Joomla – універсальна CMS з великою кількістю розширень;

– Magento – спеціалізується на інтернет-магазинах.

Типову схему системи управління контентом зображено на рисунку 1.4.

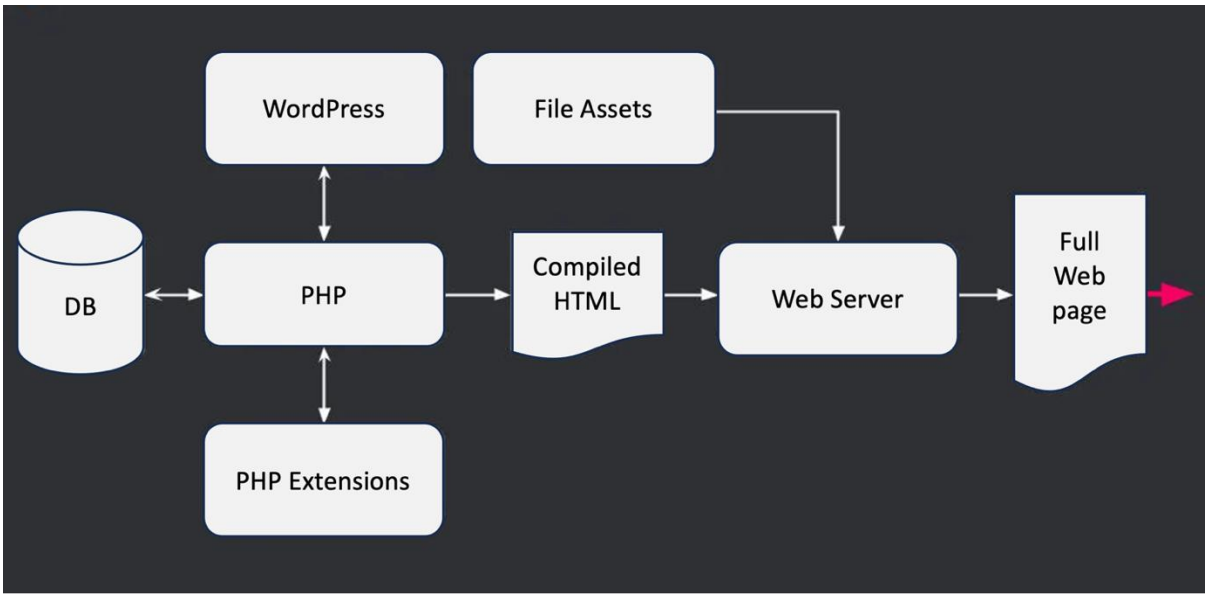

Рисунок 1.4 – Типова структурна схема CMS

Популярним підходом до створення вебсайту інтернет-магазину також є використання фреймворків.

Фреймворк – це програмна основа, яка надає набір інструментів і шаблонів для розроблення вебдодатків. Вони забезпечують базову функціональність, таку як керування сеансами користувачів, робота з базами даних і системи макетів. Використання фреймворку дає змогу розробникам економити час, зосереджуючись на унікальних аспектах проєкту замість створення базової структури з нуля [8].

Фреймворки для мови PHP: Symfony, Laravel, Yii.

Усі ці фреймворки базуються на шаблоні MVC (Model-View-Controller) і є чудовим вибором для створення малих і середніх додатків. Вони добре підходять для інтернет-магазинів через гарний функціонал.

Для веброзроблення використовуються різні мови програмування, і для кожної з них існують популярні фреймворки. Основними мовами для веброзроблення є Python, PHP і Java. Під час вибору фреймворку варто враховувати базовий функціонал, сумісність із БД, мову програмування, що вже використовується в інших компонентах ІС, та рівень захисту, який забезпечує фреймворк.

Laravel – це PHP-фреймворк із відкритим кодом, призначений для розроблення додатків. Серед його особливостей можна виділити: різні способи для доступу до реляційних БД, модульну систему упакування з виділеним менеджером залежностей Composer, орієнтація на синтаксичний цукор та утиліти, що допомагають у розгортанні додатків.

Структурна схема Laravel зображена на рисунку 1.5.

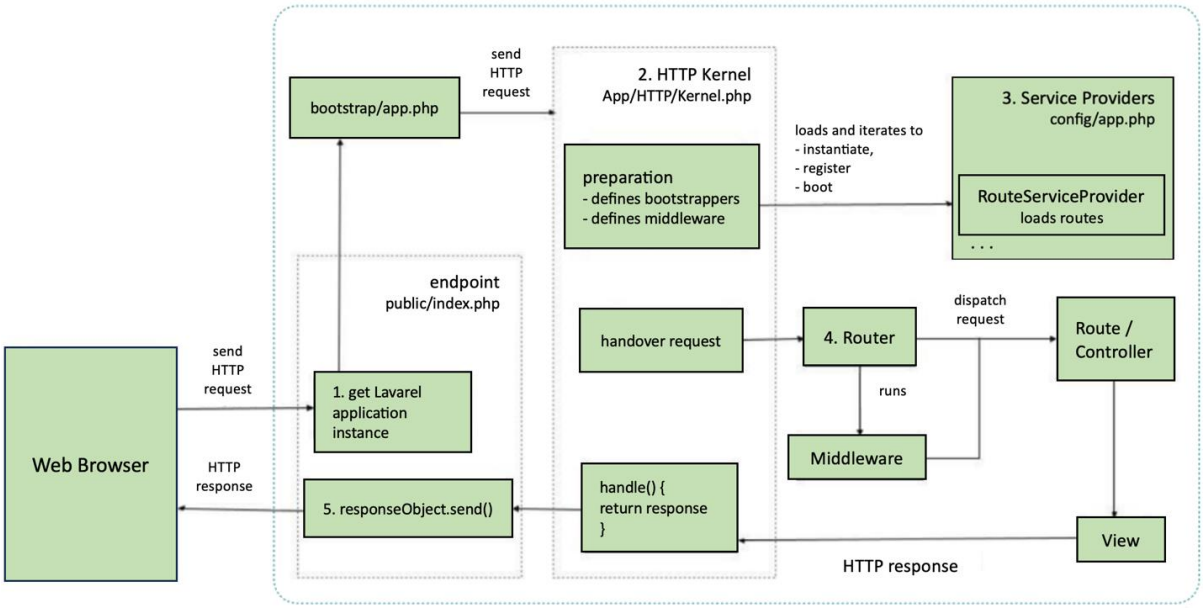

Рисунок 1.5 – Структурна схема Laravel

Фреймворки для Python: Django, Flask.

Django – це повнофункціональний MVC-фреймворк, який завдяки багатому набору вбудованих інструментів і функцій дає змогу значно заощадити час і кошти на розроблення сайту. Django добре підходить для середніх проєктів, таких як інтернет-магазини або торговельні онлайн-системи [9].

Flask – це мікрофреймворк, який надає лише базові сервіси, дозволяючи розробникам розширювати функціонал за допомогою додаткових пакетів. Завдяки компактності й прозорості структури, Flask є чудовим вибором для тих, хто хоче створити налаштовуваний стек, зосереджуючись лише на необхідному функціоналі [10].

Структурні схеми фреймворків Django і Flask зображені на рисунках 1.6 і 1.7 відповідно.

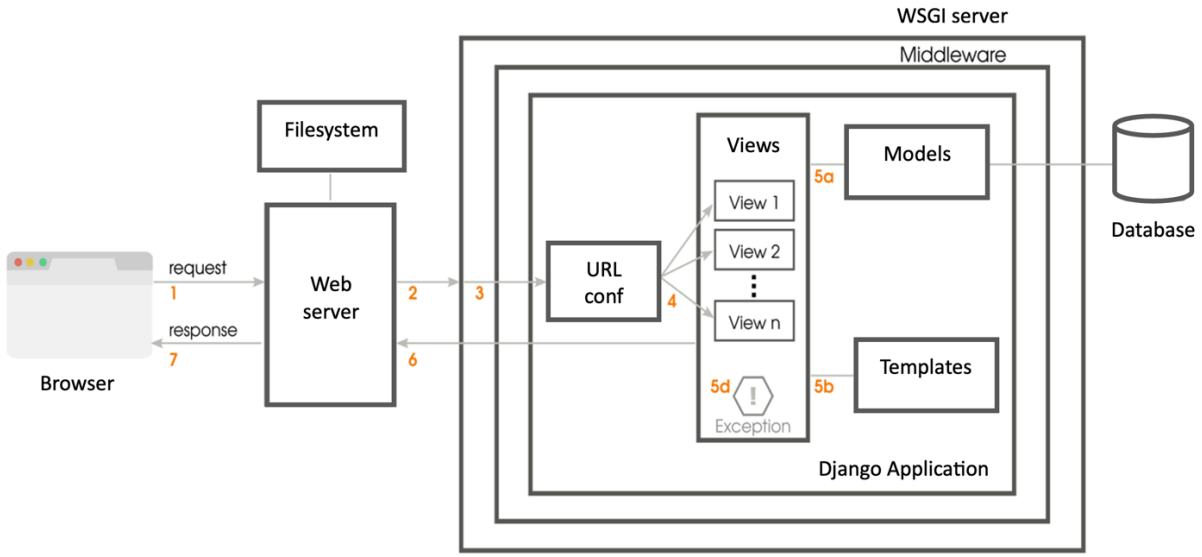

Рисунок 1.6 – Структурна схема Django

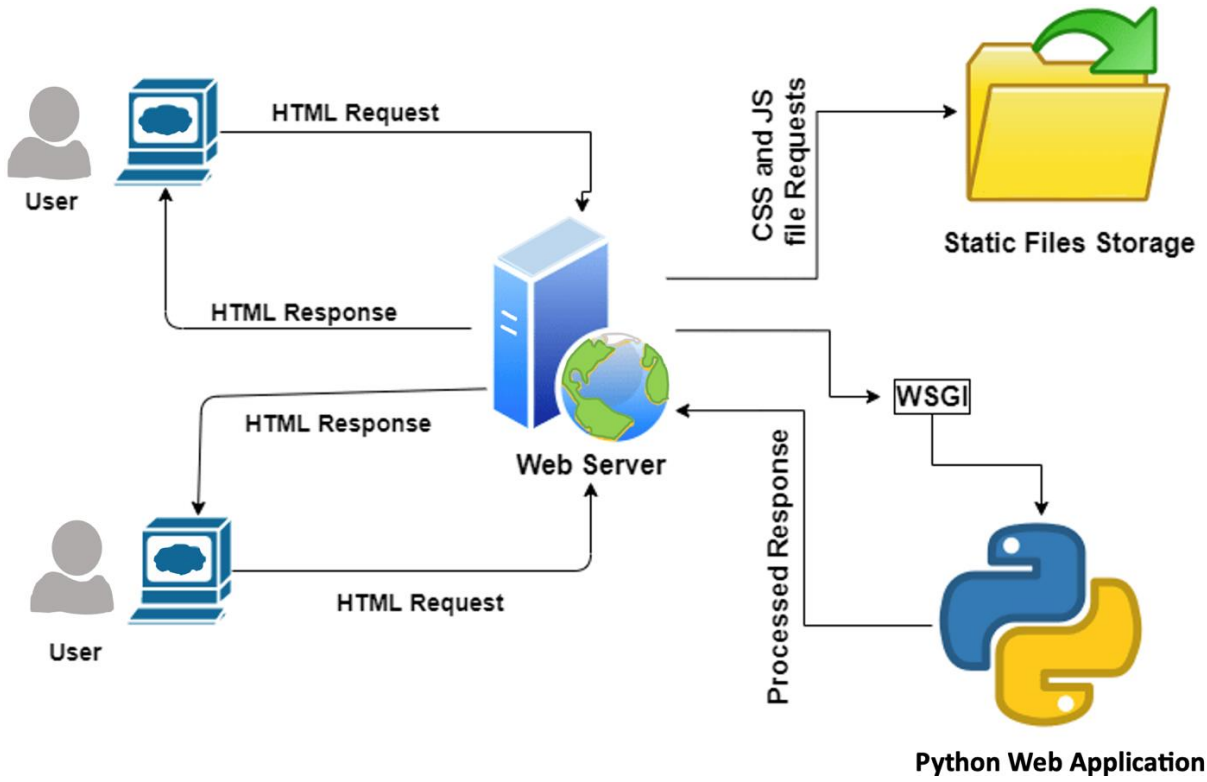

Рисунок 1.7 – Структурна схема Flask

Одним із найпопулярніших фреймворків для мови програмування Java є Spring. Це фреймворк із відкритим кодом, який реалізує принцип інверсії управління та надає широкий набір інструментів для розроблення

додатків на платформі Java. Він використовується для створення як стандартних програм Java, так і вебдодатків. Spring не обмежує розробників якимось одним підходом до програмування, але часто розглядається як альтернатива або доповнення до моделі Enterprise JavaBeans (EJB). Завдяки своїй гнучкості та розширюваності Spring став популярним серед Java-спільноти [11].

SaaS (Software as a Service) – це платформи, які дають змогу швидко і недорого створювати сайти на умовах оренди, без потреби в окремому хостингу. Вони є гарним вибором для запуску простих проєктів, тимчасових сайтів або тестування бізнес-ідей. SaaS-платформи бувають специфічними, тобто для створення лише інтернет-магазинів чи певного типу сайтів, та універсальними – підходять для різних типів вебсайтів [12].

Приклади SaaS-платформ: Shopify, UMI, Setup, WIX, InSales, uCoz.

Переваги SaaS-платформ:

– швидкість запуску. Рішення дають змогу створити сайт у стислі терміни;

– доступність. Невисока вартість реалізації;

– простота в користуванні. Інтуїтивно зрозумілі інструменти, які не потребують програмістських навичок.

Недоліки SaaS-платформ:

– шаблонний дизайн. Вебсайти створюються за стандартними шаблонами, які мають обмежену можливість налаштування. Для проєктів із високими вимогами до унікальності дизайну ці платформи не підходять;

– обмежений функціонал. Якщо платформа не підтримує необхідні функції, їхня реалізація буде неможливою. SaaS не дозволяє гнучко налаштовувати продукти під індивідуальні вимоги;

– низький рівень безпеки. Системи орієнтовані на простоту та швидкість створення сайтів, тому часто мають базові механізми захисту, які підходять тільки для маленьких сайтів.

SaaS-платформи є оптимальним рішенням для проєктів без високих вимог до дизайну та функціоналу. Якщо ж проєкт потребує гнучкості, масштабованості чи специфічних функцій, доцільніше звернути увагу на інші інструменти для розроблення вебсайтів. Структурну схему SaaS-платформи зображено на рисунку 1.8.

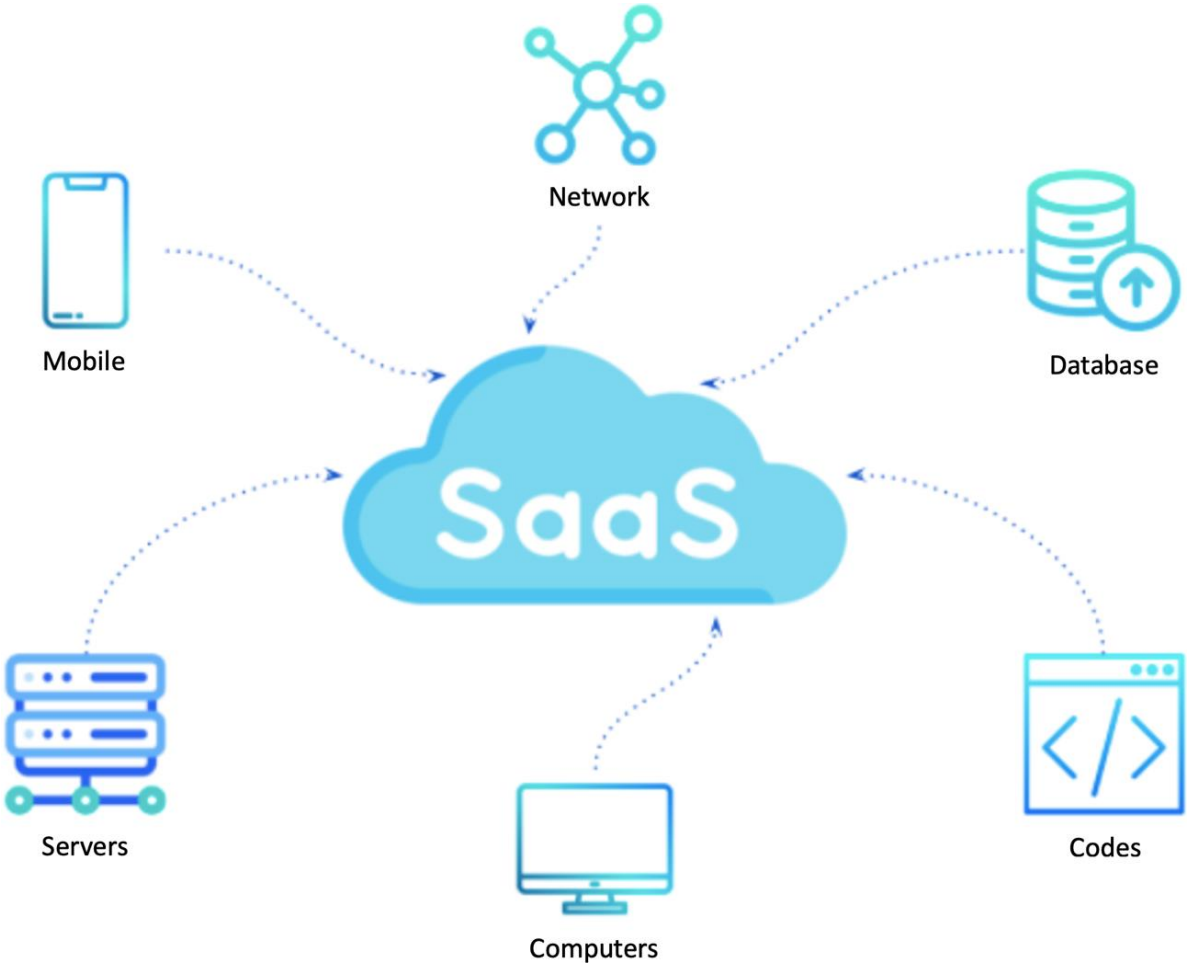

Рисунок 1.8 – Структурна схема SaaS

1.4 Постановка задач дослідження

Об’єкт дослідження – ІС вебсайту інтернет-магазину.

Предмет дослідження – методи розроблення вебсайту інтернет-магазину.

Мета роботи – дослідження методів розроблення, планування, проєктування і тестування вебсайтів інтернет-магазинів та пропозиції щодо поліпшення аспектів цих методів.

Для цього потрібно:

– здійснити аналіз вимог щодо розроблення вебсайту інтернет-магазину з метою оцінки можливостей покращення рівня безпеки проєктних рішень, що існують;

– здійснити дослідження наявних методів розроблення вебсайту інтернет-магазину;

– розробити метод створення вебсайту інтернет-магазину, зважаючи на покращення рівня безпеки на базі досліджень з минулих пунктів;

– обґрунтувати вибір інструментів, що необхідні для реалізації вдосконаленого методу, зважаючи на покращення рівня безпеки;

– сформувати методику застосування модифікованого методу, зважаючи на покращення рівня безпеки;

– здійснити апробацію запропонованого методу розроблення вебсайту інтернет-магазину, зважаючи на покращення рівня безпеки.

2 ДОСЛІДЖЕННЯ ТА ПОБУДОВА МЕТОДУ РОЗРОБЛЕННЯ ВЕБСАЙТУ ІНТЕРНЕТ-МАГАЗИНУ, ЗВАЖАЮЧИ НА ЗАХОДИ БЕЗПЕКИ

2.1 Дослідження наявних методів розроблення вебсайту інтернет-магазину

У процесі розвитку вебдодатків фахівці створили немало методів їхнього розроблення. Усі вони мають основну мету – забезпечення управління життєвим циклом ПЗ, включно з розробленням і підтримкою. Ці методи мають поєднувати класичні підходи і принципи розроблення програмного забезпечення із конкретними аспектами інтернету. В цілому, сучасні методи доповнюють традиційні шляхи додавання навігаційної моделі, щоб визначити навігаційні характеристики вебсайту, зокрема з огляду на вимоги певних груп користувачів [13].

Останнім часом були спроби для контексту вебсайтів адаптувати методи, такі як: Data Flow Diagram (DFD), Entity-Relationship (ER) та Object-Oriented (OO). Окрім цього, для швидкого розроблення актуальними є гнучкі методи, оскільки вони орієнтовані на оперативне постачання робочих версій продукту.

У таблиці 2.1 наведені атрибути деяких наявних методів розроблення вебсайту інтернет-магазину.

Таблиця 2.1 – Атрибути методів розроблення сайту інтернет-магазину

Назва методу	Спосіб моделювання	Нотація	Інструменти
1	2	3	4
OOHDM	OO	UML і власна	OOHDM-Web
WebML	OO	UML, ER, власна	WebRatio
UWE	OO	UML і власна	ArgoUWE
OO-H	OO	UML і власна	CASE Tool

Кінець таблиці 2.1

1	2	3	4
OOWS	OO	UML	
WAE	OO	UML і власна	
W2K	OO і ER	UML, HDM і власна	
HDM	ER	ER	
RMM	ER	ER і власна	
SWM	DFD	DFD	ASCENT

Слід зазначити, що ця оцінка може не враховувати кілька важливих аспектів. Різні методи розроблення пропонують різний рівень деталізації й різні інструменти для підтримання процесу. Приміром, деякі з них зосереджуються лише на текстових вказівках, тоді як інші надають графічні інструменти для полегшення ключових етапів розроблення. Для прикладу можна навести метод OOHDM, який включає графічні моделі для етапів архітектурного проєктування, навігації та інтерфейсу. OOHDM забезпечує чітко визначену процедуру розроблення додатків [14].

Hypertext Design Model (HDM), зі свого боку, є моделлю структурованого дизайну гіпертекстових програм. На відміну від моделі процесу, HDM описує лише модель проєктування. HDM використовує принципи поділу на підсистеми та модульність для створення ефективних, масштабованих і зрозумілих архітектур. Водночас OOHDM пропонує повністю структурований підхід до розроблення додатків гіпермедіа.

На рисунку 2.1 зображено основні етапи створення інтернет-магазину з використанням методу OOHDM.

Рисунок 2.1 – Основні етапи створення вебсайту інтернет-магазину з використанням методу OOHDM

Методологія OOHDM охоплює п’ять ключових видів діяльності, як наведено на рисунку 2.1 [15].

Першим йде збір вимог. На цьому етапі збираються вимоги зацікавлених сторін, які представляються з допомогою варіантів використання та діаграм взаємодії користувача.

Далі слідує концептуальний дизайн. Концептуальна модель відповідає об’єктно-орієнтованій моделі й описує сутності дизайну з допомогою UML-нотацій.

Наступний етап – навігаційний дизайн. Він спрямований на створення навігаційної моделі, яка є представленням концептуальної моделі. Вона включає схему навігаційних класів і схему навігаційного контексту.

Абстрактний дизайн є четвертим етапом. Він фокусується на об’єктах інтерфейсу користувача, для його опису використовується метод проєктування абстрактного представлення даних.

Завершальний етап – реалізація, яка полягає в перенесенні розробленого проєкту у фінальну програму.

Ще одним популярним методом є WebML. Це методологія та мова моделювання, призначена для розроблення вебдодатків. WebML орієнтована на структуроване моделювання складних вебсистем та інтерактивних інтерфейсів. Вона забезпечує формальний підхід до проєктування, що дає можливість зручніше розробляти масштабовані та динамічні вебсайти. Вона пропонує графічні, але формалізовані специфікації, що втілюються в цілісний процес проєктування. Цьому процесу можуть допомагати візуальні інструменти, наприклад, WebRatio.

Метод WebML базується на п’яти основних моделях: структурна, концептуальна, композиції, навігаційна, презентаційна [16].

Структурна модель: базується на концептуальних моделях даних, таких як ER, ODMG і UML, якому автори надають перевагу.

Концептуальна модель. Ця модель описує, як можна розширювати структуру за рахунок похідних даних для додавання надлишкової інформації. По-іншому, вона схожа на механізм VIEW у моделюванні БД, як-от у MySQL або Oracle. Для кожної сторінки визначається абстрактна таблиця даних, яка об’єднує дані з кількох інших таблиць. Для запитів використовується мова WebML-OQL (WebML Object Query Language).

Модель композиції. Вона визначає, як вміст розподіляється на сторінках вебдодатку.

Навігаційна модель. Модель описує взаємозв’язки між сторінками через навігаційні посилання – як контекстні, так і неконтекстні. Вона моделює те, як користувач переміщується сторінками вебдодатку.

Презентаційна модель. Цей аспект моделювання відповідає за вигляд і стиль сторінок, визначених у моделі композиції. Вебсторінки, створені за допомогою WebML, візуалізуються згідно з таблицями стилів.

Особливості WebML. WebML пропонує набір методів та інструментів для систематичного проєктування і розроблення сайтів. Ця методологія вирішує численні проблеми завдяки використанню окремих моделей. До того ж, WebML включає компілятори моделей, які автоматично генерують більшу частину логіки та структури сторінок вебдодатку на основі цих моделей.

Обмеження WebML. Незважаючи на численні переваги, метод має певні недоліки. Зокрема, це обмеження в обміні моделями, а також труднощі з представленням таких аспектів, як технологічна незалежність або архітектурні стилі.

Приклад моделі WebML подано на рисунку 2.2.

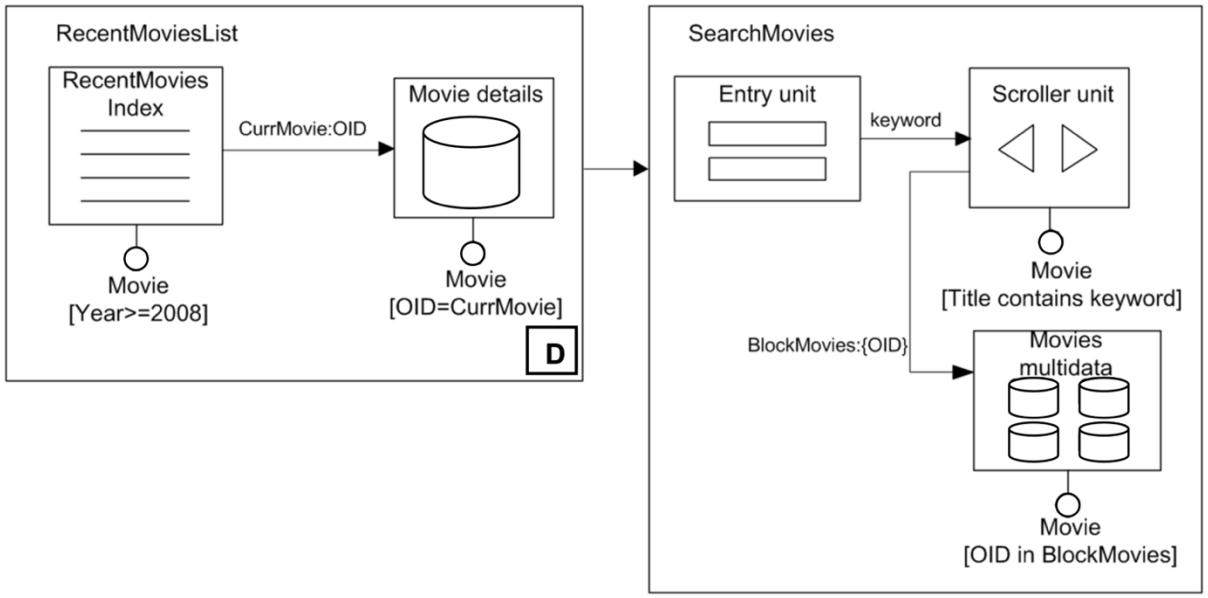

Рисунок 2.2 – Приклад моделі WebML

W2000 (W2K) – це методологія управління проєктами та розроблення програмного забезпечення, що базується на поетапному підході до створення програмних продуктів. Вона поєднує в собі елементи традиційних водоспадних методів та більш гнучких підходів, таких як Agile. Її основою є модель гіпертекстового дизайну (HDM), орієнтована на гіпермедіа та дані, однак W2000 також використовує принципи UML для підтримання бізнес-процесів.

Методологія W2000 дає змогу дизайнерам моделювати всі аспекти додатків – від вебсторінок до складних бізнес-транзакцій – інтегрованим й узгодженим способом. Завдяки підходу, керованому моделями (model-driven approach), дизайнери можуть поступово вдосконалювати свої моделі та легко переходити від етапу специфікації до дизайну.

Структура W2000 складається з чотирьох основних моделей: інформаційної, навігаційної, службової, презентаційної. Інформаційна модель визначає дані, які використовуються додатком і сприймаються користувачем. Навігаційна описує, як організовано вміст для забезпечення коректної реалізації. Службова модель визначає елементи управління

додатком. Презентаційна описує, як дані та сервіси представлені користувачу.

Ієрархічну структуру W2K-методу, яка демонструє його організацію, зображено на рисунку 2.3.

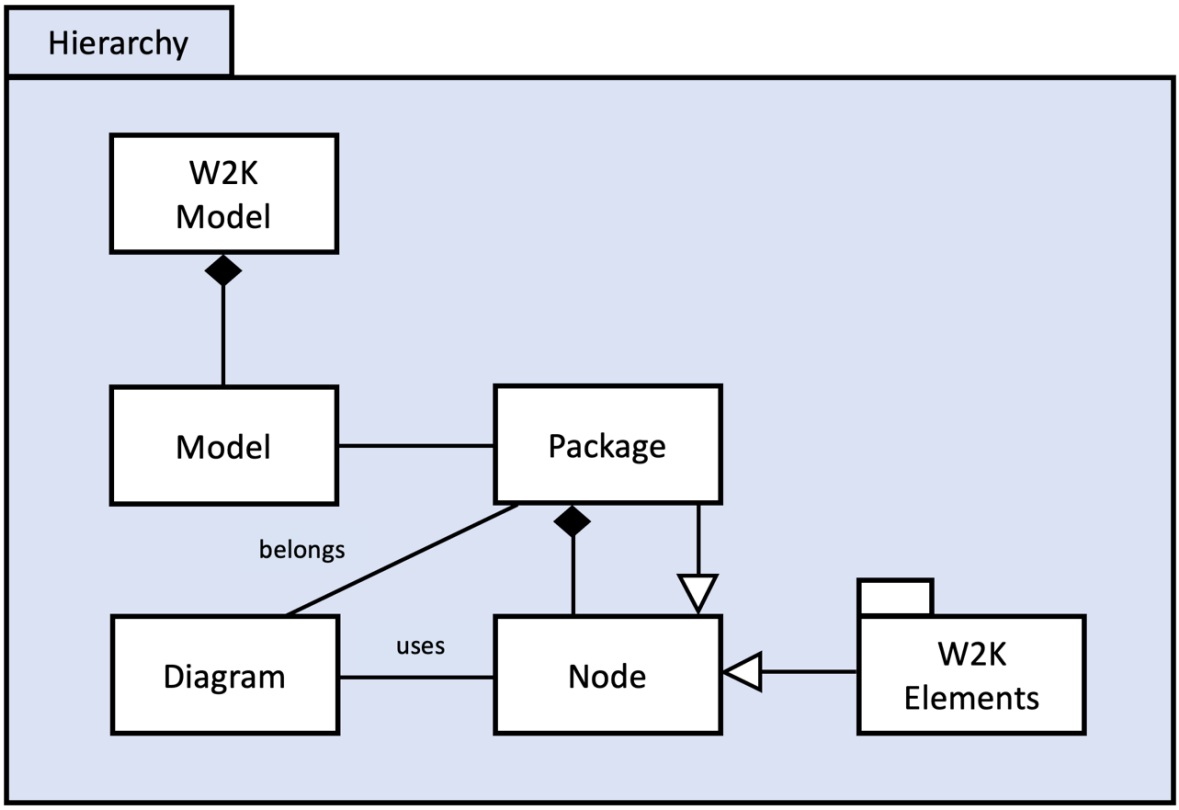

Рисунок 2.3 – Ієрархічна структура W2K-методу

Таблиця 2.2 теж вказує на обмеження багатьох методів розроблення, які надають лише мінімальну підтримку різним етапам створення додатків, особливо тестуванню. Це може призводити до великої кількості помилок під час розроблення, зокрема в аспектах безпеки та коректності функціоналу вебдодатків.

Таблиця 2.2 – Підтримання етапів розроблення вебсайту інтернет-магазину різними методами

Назва методу	Формування вимог	Планування	Аналіз	Проєктування архітектури	Проєктування навігації	Проєктування інтерфейсу	Розроблення	Тестування	Підтримання
OOHDM				+	+	+	+		
WebML					+	+	+		
OO-H			+	+	+	+	+		
UWE			+	+	+	+			
OOWS			+	+	+	+	+		
WAE		+	+	+	+	+	+	+	
W2K			+	+	+	+			
HDM				+	+				
RMM				+	+	+	+		
SWM	+	+	+	+	+	+	+	+	

Аналіз трьох методів розроблення вебсайтів показує, що кожен із них має свої переваги і недоліки та підходить для різних цілей. Проте, ключовою проблемою всіх розглянутих підходів є недостатня увага до функцій безпеки. Через нехтування проєктуванням механізмів захисту на ранніх етапах розроблення можуть виникати серйозні вразливості у системі безпеки. Це створює значний ризик несанкціонованого доступу, викрадення даних клієнтів чи конфіденційної інформації компанії.

Існує чимало видів загроз, однак не всі використовуються для зламу сайту. Вибір інструментів і методів атаки залежить від архітектури додатка та засобів, застосованих під час його розроблення. Тому для захисту інтернет-магазинів важливо ідентифікувати основні загрози й забезпечити відповідні заходи для їхньої нейтралізації.

Якщо ІС, для якої створюється сайт, містить особливі елементи, функції або дані, може виникнути потреба у розробці кастомних механізмів захисту. Також можуть бути використані специфічні функції безпеки, які нечасто використовуються для захисту вебдодатків.

2.2 Дослідження можливих загроз під час роботи вебсайту інтернет-магазину

Іноді відбувається недостатня кількість етапів розроблення, коли основна ціль – створити продукт як можна хутчіше, а не якісніше, початкові стадії проєктування часто відходять на другий план або навіть ігноруються. Як наслідок, заходи безпеки впроваджуються значно пізніше, ніж це необхідно, наприклад, лише на етапі тестування або розгортання вебсайту.

Згідно з даними IBM, виправлення помилок, що були виявлені після запуску продукту, обходиться в 4 – 5 разів дорожче, ніж виправлення на етапі проєктування, і аж до 100 разів дорожче, ніж виявлення помилок під

час обслуговування. Дослідження Stake показують те, що в середньому компанії виявляють тільки 25% вразливостей у своїх програмних продуктах. У типовому корпоративному програмному забезпеченні зазвичай виявляють близько семи значних помилок. Виправлення таких дефектів на етапі тестування коштує приблизно у 7 разів дешевше, ніж після розгортання. Дослідження також підтверджують, що впровадження безпеки на етапі проєктування дає 21% повернення інвестицій у безпеку IT (ROSI), тоді як відкладення цього до етапу впровадження знижує показник до 15%, а до етапу тестування – до 12%. Впровадження безпеки на початкових стадіях життєвого циклу розроблення дає можливість створювати надійніші й більш захищені програми за менших витрат [17].

Ігнорування безпеки на початкових етапах призводить до появи численних слабких місць, які шахраї можуть використати, аби отримати доступ до конфіденційних даних клієнтів, таких як паролі, коди, адреси, документи, номери телефонів та ін.

Однією з найпопулярніших атак є CSRF (Cross-Site Request Forgery). Цей тип кібератаки передбачає зловмисне використання довіри програми до браузера користувача. Під час неї користувач обманом змушується надіслати запит, якого він не мав наміру виконувати. Наприклад, це можуть бути запити через приховані форми, теги зображень або JavaScript, які працюють без відома користувача [18].

На противагу крос-сайтового сценарію XSS (Cross-Site Scripting), який експлуатує довіру людини до вебсайту, CSRF використовує довіру вебсайту до браузера. Результатом атаки можуть бути витік конфіденційних даних, зміна сеансу користувача чи маніпуляції з їхніми акаунтами [19]. Діаграму роботи атаки CSRF показано на рисунку 2.4.

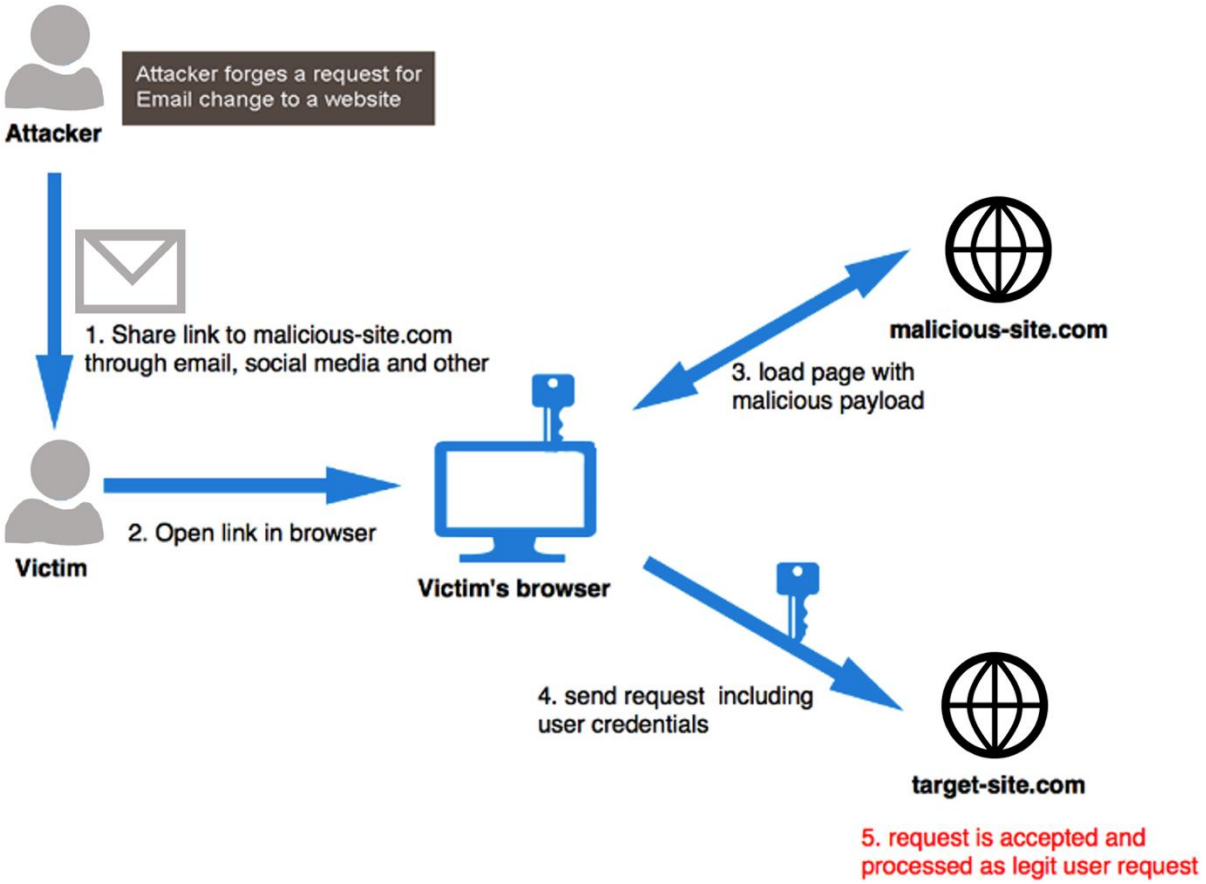

Рисунок 2.4 – Діаграма роботи атаки CSRF

XSS-атаки дають можливість хакерам впроваджувати скрипт з боку клієнта до браузерів інших користувачів. Це може відбуватися через збереження шкідливого коду в БД, який згодом відображається іншим користувачам, або через спонукання користувачів перейти за спеціальним покликанням, що призводить до спрацювання шкідливого JavaScript у браузері. Джерелом таких кібератак можуть бути не лише бази даних, але й інші ненадійні джерела даних, як-от файли cookie чи вебслужби, якщо дані недостатньо перевіряються перед відображенням на сторінці.

SQL-ін’єкції – це атака, що дає змогу зловмисникам виконувати довільні SQL-запити у БД. Вона може призводити до викрадення конфіденційної інформації, видалення записів або навіть компрометації всієї бази даних [20].

Клікджекінг (перехоплення кліків) – це маніпуляція, під час якої зловмисники додають невидиме віконце з прихованою командою поверх видимого контенту. Користувач, нічого не підозрюючи, клікає на це приховане віконце, виконуючи дії, задумані зловмисником. Наприклад, користувач може ненавмисно надати доступ до облікового запису чи здійснити транзакцію [21].

DoS-атака (Denial of Service, відмова в обслуговуванні) спрямована на виведення сервера з ладу, змушуючи його не відповідати на запити. Цей вид атак зазвичай не вбачає викрадення конфіденційних даних, проте може бути допоміжним інструментом для інших. Приміром, помилки у програмному забезпеченні можуть призводити до виконання шкідливого коду під час аварійного вимикання сервісу. Кібератаки лавинного типу також викликають перевантаження сервера через величезну кількість запитів [22].

DDoS-атака (Distributed Denial of Service) має таку саму ціль, як і DoS, проте виконується з багатьох комп’ютерів одночасно. DDoS-атаки організовуються через мережу комп’ютерів, кожен із яких виконує DoS-атаку на цільову систему. Це створює величезне навантаження на сервер і часто робить його недоступним. DDoS використовують там, де звичайні DoS-атаки виявляються неефективними, оскільки система захисту може блокувати запити, що надходять із однієї IP-адреси [23].

У таблиці 2.3 представлено загрози, що можуть з’явитися через недоліки дизайну вебсайту.

Таблиця 2.3 – Загрози, які можуть з’явитися через недоліки дизайну вебсайту

Категорія	Можливі загрози
1	2
Перевірка даних введення	SQL-ін’єкції, міжсайтовий скриптинг, переповнення буферу, канонізація

Кінець таблиці 2.3

1	2
Аутентифікація	Відмова у виконанні операції користувачем, використання сайту без змоги відстеження шахрая
Авторизація	Отримання приватних даних, підвищення привілеїв, підроблення даних, фішингова атака
Конфігурація	Несанкціонований доступ до адміністраторського інтерфейсу, доступ до репозиторію конфігураційних даних
Сеанс	Перехоплення сесії, повторна активація викраденого сеансу
Приватні дані	Викрадення приватної інформації через злам
Маніпулювання з параметрами	Маніпулювання з вебзапитами, cookie, підроблення HTTP-заголовків
Криптографія	Погане генерування ключів або керування ними, слабке шифрування
Ситуації-винятки	Отримання конфіденційної інформації, відмова з обслуговуванням

2.3 Побудова методу розроблення вебсайту інтернет-магазину, зважаючи на заходи безпеки

Вдосконалений метод розроблення пропонує включити всі базові етапи створення програмного забезпечення, такі як формування вимог, планування, аналіз, із впровадженням заходів безпеки на кожному з них. Це дає змогу врахувати потенційні вразливості вебсайту, ефективно спланувати їх усунення, а також провести повноцінне тестування.

Вдосконалений метод сполучає ключові етапи моделювання з розглянутих раніше методів.

Етап 1. Парадигма моделювання. Використовується парадигма гіпертексту та БД із методології WebML, адже в багатьох вебсайтах інформація зберігається в БД для адміністрування чи прийняття рішень, а гіпертекст забезпечує доступ для користувачів.

Етап 2. Моделювання вимог. Специфікація вимог до ПЗ та методи аналізу структурованих систем дають можливість визначити головні бізнес-вимоги. Методології OOHDM і W2K пропонують багато інструментів для цього.

Етап 3. Моделювання вмісту. Методологія WebML забезпечує детальну фіксацію вмісту. Її унікальні особливості включають підтримку робочих процесів, які залучають кількох користувачів, а також можливість синхронізації відповідних гіпертекстових представлень і контролю окремих типів потоку діяльності через призначення об’єктів.

Етап 4. Моделювання навігації. WebML чітко визначає структуру класів, що підтримують створені навігаційні структури. Методології UWE, OO-H і WebML вводять концептуальні діаграми процесів, які підтримують як структуру, так і навігацію. OOHDM розширює свої навігаційні моделі для підтримання нових концепцій. Навігаційна діаграма OOHDM забезпечує більш гнучке та детальне моделювання навігації.

Схема алгоритму для підвищення безпеки вебсайту представлено на рисунку 2.5.

Рисунок 2.5 – Схема алгоритму для підвищення безпеки вебсайту

Оцінювання безпеки та моделювання загроз є важливими етапами як під час створення нових сайтів, так і під час вдосконалення тих, що вже існують. Задля забезпечення безпеки під час розроблення нового сайту необхідно вже на початкових етапах виконати моделювання загроз і

визначити вимоги до безпеки. Якщо ж мета полягає в підвищенні захисту сайту, що вже є, слід здійснити оцінку його поточних безпекових функцій.

Для розроблення надійного вебдодатку, стійкого до зламів, потрібно застосувати систематичний і комплексний підхід до захисту на трьох рівнях: мережевому, рівні хоста та рівні додатку. На кожному з цих рівнів слід враховувати специфічні питання безпеки. На рисунку 2.6 зображено категорії конфігурації безпеки, що використовуються для налаштування параметрів захисту хоста та мережі, а також категорії вразливостей додатків, які допомагають структурувати аналіз потенційних ризиків.

Цілісний підхід до безпеки сайтів передбачає комплексний аналіз та впровадження захисту на всіх рівнях системи, оскільки зловмисники можуть скористатися слабкими місцями на будь-якому з них. Саме тому заходи безпеки мають охоплювати мережевий рівень, рівень хоста та рівень додатків. Підхід до безпеки сайту ілюструється на рисунку 2.6.

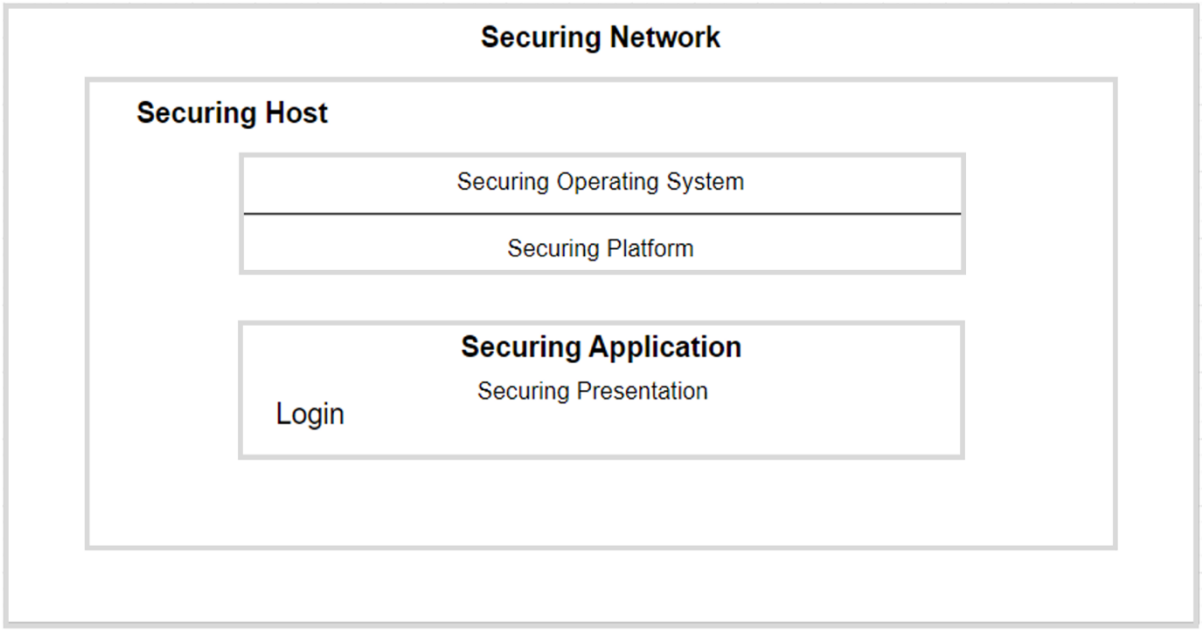

Рисунок 2.6 – Цілісний підхід до безпеки сайту

Хост складається з операційної системи, .NET Framework та відповідних служб і компонентів. Не дивлячись на роль сервера (сервер з

IIS, сервер додатків із Enterprise Services чи сервер БД із SQL Server), є загальна методологія, яка охоплює різні типи серверів.

Для створення безпечних сайтів та оцінки тих, що існують, застосовуються категорії вразливостей вебдодатків, що охоплюють широкий спектр технологій і компонентів у мультирівневій архітектурі. Ці категорії враховують такі аспекти: ануляція введення, аутентифікація, авторизація, захист приватних даних, управління сеансами, управління конфігурацією, маніпуляція параметрами, криптографія і управління винятками.

Сайти створюють низку проблем через специфіку протоколу HTTP, який не зберігає стан. До прикладу, додаток має відстежувати сеанси користувачів і забезпечувати аутентифікацію. Через те, що всі подальші дії авторизації залежать від ідентифікації користувача, важливо забезпечити надійність процесів аутентифікації й управління сеансами.

Крім того, оскільки вебдодатки працюють через загальнодоступні мережі, критично важливо запобігати маніпулювання з параметрами, витоку приватних даних та іншим ризикам.

Зазвичай безпеку сприймали як мережеву проблему, де фаєрвол (брандмауер) виконував роль головного захисту. Інший підхід полягав у блокуванні хостів системними адміністраторами. Однак ці методи мають значні обмеження. Хоча фаєрволи обмежують трафік до HTTP, але він може мати команди, що експлуатують вразливості додатків. Також підхід блокування хостів не враховує атак на рівні додатків, які стають все більш поширеними.

Ще однією зоною ризику є процес розгортання. Нерідка ситуація: програма не працює в захищеному середовищі, змушуючи адміністратора знижувати безпекові налаштування, що створює нові вразливості. Також небезпеку становить відсутність чіткої політики безпеки.

2.4 Формування критерію ефективності оцінювання рівня захисту вебсайту від загроз

Визначення типів атак на сайти дає можливість сформувати критерії оцінки ефективності запропонованого методу. Щоб це зробити, необхідно встановити рівні захисту сайту, які представлені в таблиці 2.4.

Таблиця 2.4 – Рівні захисту вебсайту

1-й рівень	Система забезпечує захист вебсайту менше, ніж від 30% видів атак
2-й рівень	Система забезпечує захист вебсайту від 30 – 50% видів атак
3-й рівень	Система забезпечує захист вебсайту від 50 – 70% видів атак
4-й рівень	Система забезпечує захист вебсайту від 70 – 90% видів атак
5-й рівень	Система забезпечує захист вебсайту від 90%+ видів атак

Для вибору потрібного рівня захисту сайту інтернет-магазину потрібно обчислити коефіцієнт ефективності захисної системи (E), що вираховується за формулою 2.1:

$$E = \frac{L}{C * t}, \tag{2.1}$$

де L – захисний рівень;

C – вартість розроблення в доларах США;

t – час розроблення у годинах.

Крім того, слід ідентифікувати загрози, які потенційно можуть вплинути на безпеку додатка ІС компанії. Хоча є безліч методів зламу ПЗ і додатків, далеко не всі вони актуальні для вебсайтів. Тому прагнення захистити вебсайт від усіх можливих загроз є нераціональним, оскільки це вимагатиме значних ресурсів.

Разом з тим, вибір мінімального рівня захисту теж є помилковим рішенням. Це знижує витрати на розроблення, але суттєво підвищує ризик успішного зламу сайту. Оптимальний підхід передбачає баланс між ефективністю захисту та ресурсами, необхідними для його забезпечення.

Рівень захисту вебсайту залежить від: наявних компонентів системи, мови програмування й інших інструментів, складності системи та її архітектури.

Ці фактори безпосередньо впливають на час і вартість розроблення, а також визначають доцільність використання тих чи інших засобів захисту. Ретельний аналіз дає змогу оптимізувати витрати та забезпечити необхідний рівень безпеки.

3 МЕТОДИКА ЗАСТОСУВАННЯ МЕТОДУ РОЗРОБЛЕННЯ ВЕБСАЙТУ ІНТЕРНЕТ-МАГАЗИНУ, ЗВАЖАЮЧИ НА ЗАХОДИ БЕЗПЕКИ

3.1 Обґрунтування застосування інструментів для забезпечення захисту вебсайту інтернет-магазину від загроз

На всіх етапах розроблення вебсайту застосовуються різноманітні інструментальні засоби, які допомагають досягти цілей кожного з етапів. Для розроблення сайту, який буде стійким до зламу, потрібно правильно вибрати такі засоби. Найпопулярнішими серед них для розроблення вебсайту інтернет-магазину є системи керування контентом і фреймворки.

CMS вибирають через їхню легкість у використанні. Завдяки їм можна розробити вебсайт без написання програмного коду, адже CMS мають базову структуру, що дає змогу легко ставити теми, плагіни та інші розширення, не впливаючи на функції додатку. Такі системи часто базуються на модулях, що спрощує процес розроблення вебсайту і робить їх доступними навіть без залучення програмістів.

Однак CMS мають і недоліки, серед яких найбільшим є низький безпековий рівень. Оскільки більшість CMS є ПЗ із відкритим кодом, зловмисники можуть аналізувати цей код, знаходити вразливості та використовувати їх для зламу вебсайтів. До того ж, завантаження чи оновлення розширень із ненадійних джерел може призвести до компрометації безпеки.

Ще одним недоліком є обмежені бюджети. Компанії, що обирають CMS через їхню низьку вартість, часто економлять на всіх етапах розробки, включно з впровадженням функцій захисту. Замість комплексного проєктування безпеки зазвичай використовують декілька розширень для захисту від основних видів кібератак, уникаючи створення кастомних функцій для зменшення витрат.

Фреймворки пропонують значно вищий рівень безпеки. Чимало з них обмежують доступ до вихідного коду, що ускладнює завдання хакерам. Вебсайти на базі фреймворків зазвичай мають спеціальний код, що робить їх менш уразливими до атак.

На сьогодні більшість фреймворків включають вбудовані функції безпеки, такі як захист від CSRF, XSS та SQL-ін'єкцій, шифрування інформації тощо. Наприклад:

– Laravel надає хешування паролів, захист маршрутів, механізми для роботи з CSRF-атаками, безпечне шифрування AES, валідацію даних користувачів тощо;

– Django забезпечує захист від XSS-атак, CSRF, SQL-ін'єкцій, перехоплення кліків, а також має вбудовану валідацію заголовків вузлу.

Фреймворки часто використовуються компаніями, що мають високі вимоги до безпеки, зберігають конфіденційні дані або потребують створення унікального функціоналу. Такі проєкти вимагають ретельного підходу до безпеки та дозволяють впроваджувати кастомні функції, недоступні для CMS.

Для забезпечення безпеки будь-який сучасний вебсайт має працювати через протокол HTTPS. Це запобігає перехопленню даних між клієнтом і сервером, захищаючи облікові дані та іншу інформацію від зловмисників. Натепер майже всі фреймворки підтримують використання цього протоколу.

Основною метою запропонованого методу є покращення рівня захисту вебсайту інтернет-магазину. Це досягається шляхом впровадження функцій безпеки на початкових етапах розроблення та використання відповідних інструментальних засобів. З огляду на це, для реалізації модифікованого методу розроблення вебсайту краще використовувати один із фреймворків, що були розглянуті вище.

3.2 Методика стосовно застосування методу розроблення вебсайту інтернет-магазину, зважаючи на заходи безпеки

Спосіб використання методу розроблення вебсайту інтернет-магазину зі врахуванням заходів безпеки здійснюється у декілька послідовних етапів, а саме: формування вимог; аналіз загроз, що існують; концептуальний та навігаційний дизайни вебсайту; архітектурне проєктування; розроблення інтерфейсу; розроблення вебсайту; тестування і підтримання вебсайту.

Формування вимог. На цьому етапі розробники та працівники компанії-замовника обговорюють потреби щодо майбутнього вебсайту. Визначаються ключові вимоги до функціональності та системи захисту додатка. У результаті створюється документ ТЗ, який затверджується представниками замовника.

Аналіз загроз. Команда, відповідальна за безпеку, аналізує можливі загрози на основі вимог, визначених раніше. Вибираються засоби захисту, а ТЗ доповнюється відповідними вимогами.

Концептуальний вебдизайн. Розробляються концептуальні моделі вебсайту, зважаючи на заходи безпеки. Окрім цього, створюються UML-діаграми, що відображають основні сутності.

Навігаційний вебдизайн. Формується навігаційна модель вебсайту разом із концептуальними діаграмами.

Проєктування архітектури. Команда архітекторів проєктує архітектуру сайту, враховуючи існуючі компоненти ІС замовника. Головний архітектор проводить зустрічі з командами розробників для узгодження деталей. У результаті створюються архітектурні діаграми, які затверджуються замовником.

Розроблення інтерфейсу. Під час цього етапу розробляються вимоги до дизайну інтерфейсу, створюються шаблони дизайну, які затверджуються

замовником. Після цього розпочинається програмування дизайну інтерфейсу.

Розроблення вебсайту. Йде вибір ПЗ та інструментів, що забезпечують належний рівень безпеки. Після узгодження цих інструментів розпочинається розроблення функціоналу вебсайту та впровадження систем захисту. Після завершення розроблення функціонал затверджується замовником.

Тестування. Створюється контрольний список для перевірки функціоналу та системи захисту. Після тестування робиться звіт із переліком знайдених помилок, які надсилаються розробникам для внесення поправок.

Підтримання вебсайту. На останньому етапі виконується технічна підтримка: дрібні доопрацювання, оновлення системних елементів (пакетів, розширень), усунення проблем із функціоналом та захистом.

На рисунку 3.1 зображена детальна схема роботи модифікованого методу розроблення вебсайту інтернет-магазину із врахуванням заходів безпеки.

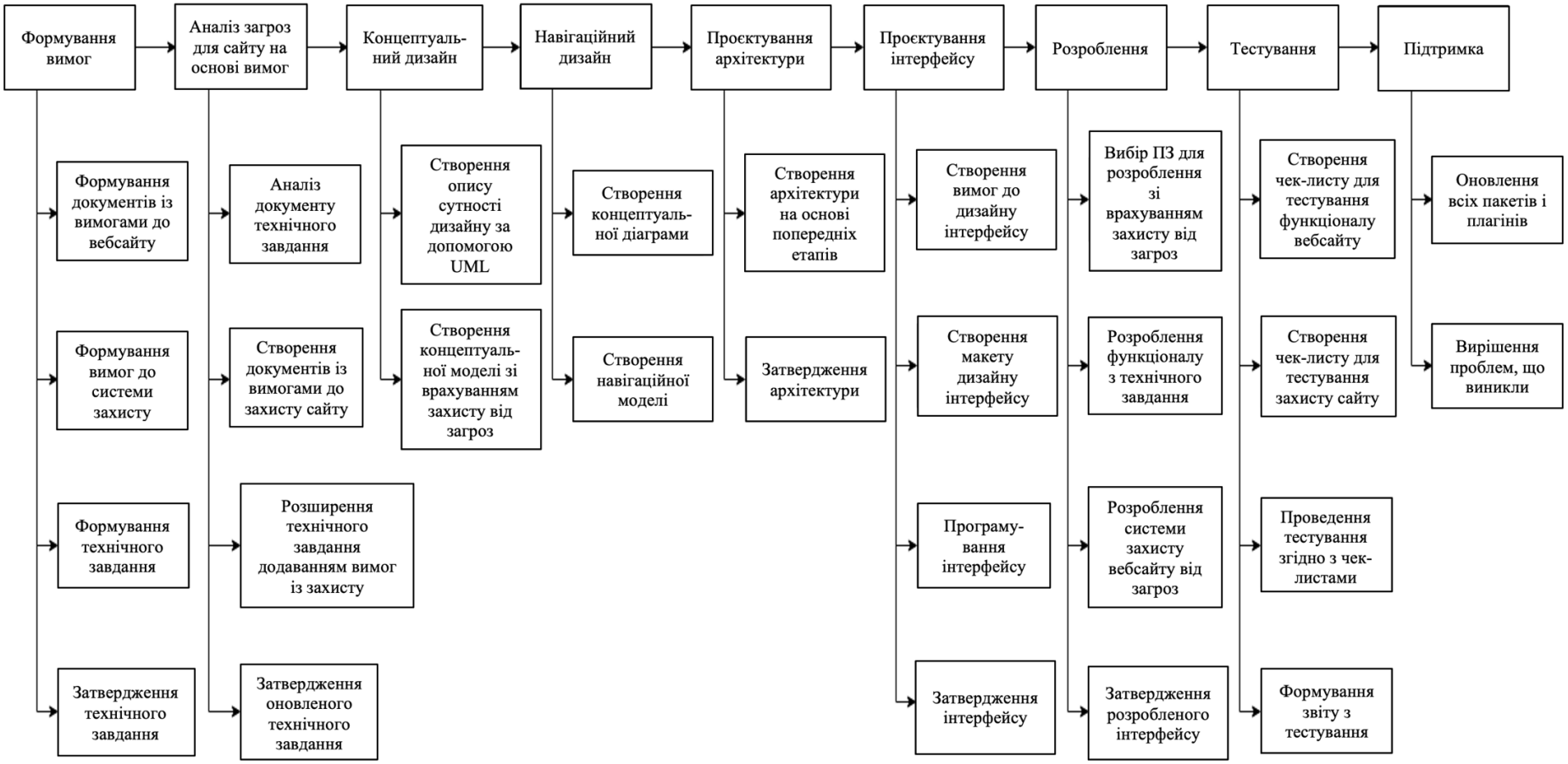

Рисунок 3.1 – Детальна схема роботи модифікованого методу розроблення вебсайту інтернет-магазину із врахуванням заходів безпеки

4 АПРОБАЦІЯ МЕТОДУ РОЗРОБЛЕННЯ ВЕБСАЙТУ ІНТЕРНЕТ-МАГАЗИНУ, ЗВАЖАЮЧИ НА ЗАХОДИ БЕЗПЕКИ

4.1 Опис вимог для побудови вебсайту інтернет-магазину годинників

Перевірка методу буде здійснюватися на прикладі вебсайту інтернет-магазину годинників.

Пропонується така послідовність вимог до розроблення вебсайту інтернет-магазину годинників зі захистом від можливих загроз:

– провести аналіз наявних загроз для інтернет-магазину, пов’язаних із продажем годинників;

– скласти перелік загроз, від яких необхідно убезпечити вебсайт;

– розробити концептуальний вебдизайн інтернет-магазину;

– створити навігаційний вебдизайн сайту;

– розробити інтерфейс користувача для сайту;

– визначити архітектуру вебсайту, зважаючи на необхідний рівень захисту;

– обрати ПЗ для створення вебсайту, зважаючи на заходи безпеки.

Як зазначено в минулих пунктах, існує велика кількість загроз, пов’язаних зі зламом сайтів, проте не всі їхні види можуть бути використані для атак на додатки. Тому необхідно окреслити найпоширеніші загрози, що застосовуються для зламування інтернет-магазину, і забезпечити проти них ефективний захист. Перелік загроз може змінюватися залежно від різних факторів, приміром, у разі необхідності впровадження складного чи незвичайного функціоналу, для якого може знадобитися унікальний захист від зламу.

У таблиці 4.1 наведено перелік загроз, від яких потрібно захистити вебсайт інтернет-магазину.

Таблиця 4.1 – Перелік загроз, від яких потрібно захистити вебсайт інтернет-магазину годинників

Категорія	Найменування типових загроз
Захист даних	Переповнення буферу, міжсайтовий скриптинг, SQL-ін’єкції
Захист форм авторизації	Отримання приватних даних, підроблення даних, фішингова атака
Захист сесії	Перехоплення сесії, повторна активація викраденого сеансу
Захист конфігурації	Несанкціонований доступ до адміністраторського інтерфейсу, доступ до репозиторію конфігураційних даних
Шифрування приватних даних	Викрадення приватної інформації через злам
Захист від маніпулювання параметрами	Маніпулювання з вебзапитами, cookie, підроблення HTTP-заголовків
Криптографія	Погане генерування ключів або керування ними, слабке шифрування
Захист сервера	Атаки DoS та DDoS

Концептуальну діаграму вебсайту інтернет-магазину годинників приведено на рисунку 4.1.

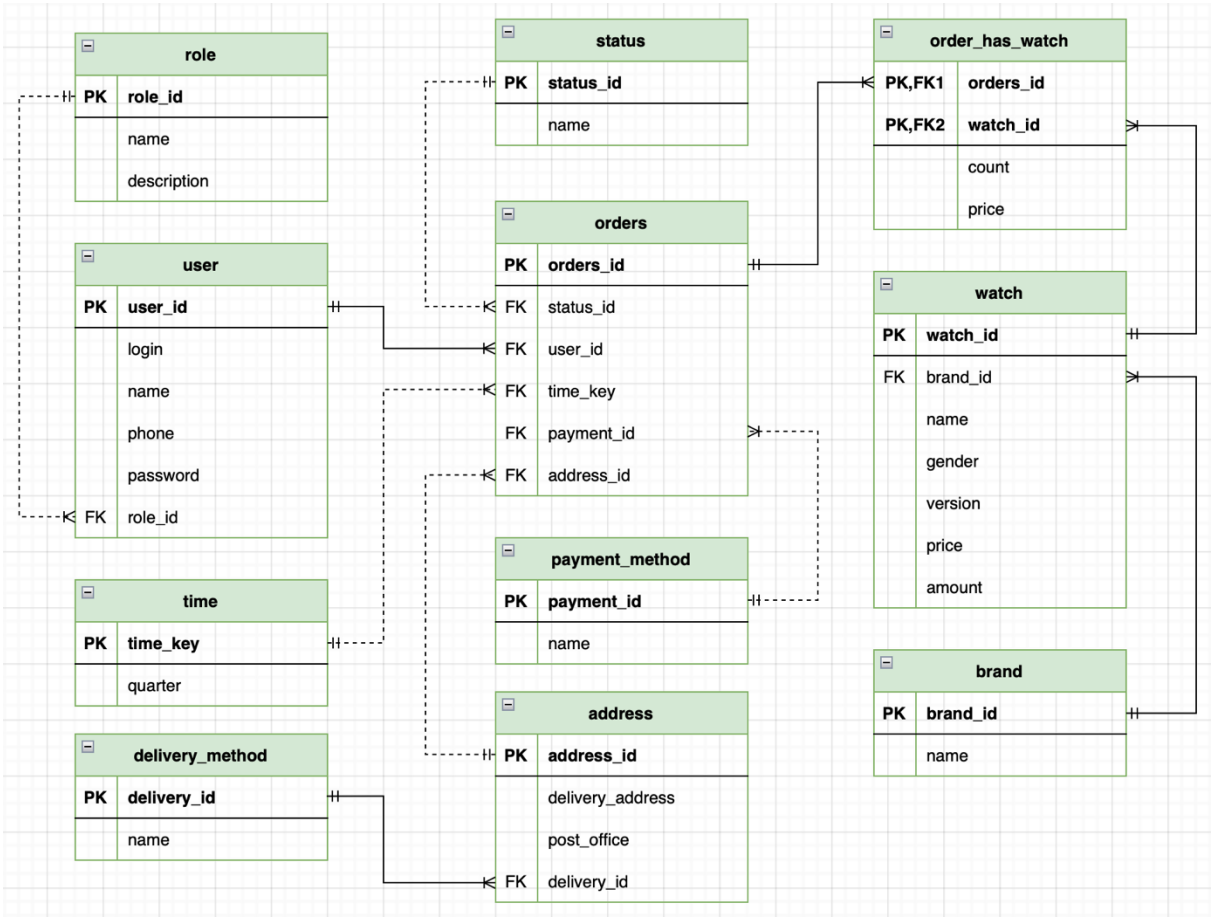

Рисунок 4.1 – Концептуальна діаграма вебсайту інтернет-магазину годинників

Навігаційний дизайн вебсайту інтернет-магазину годинників показано на рисунку 4.2.

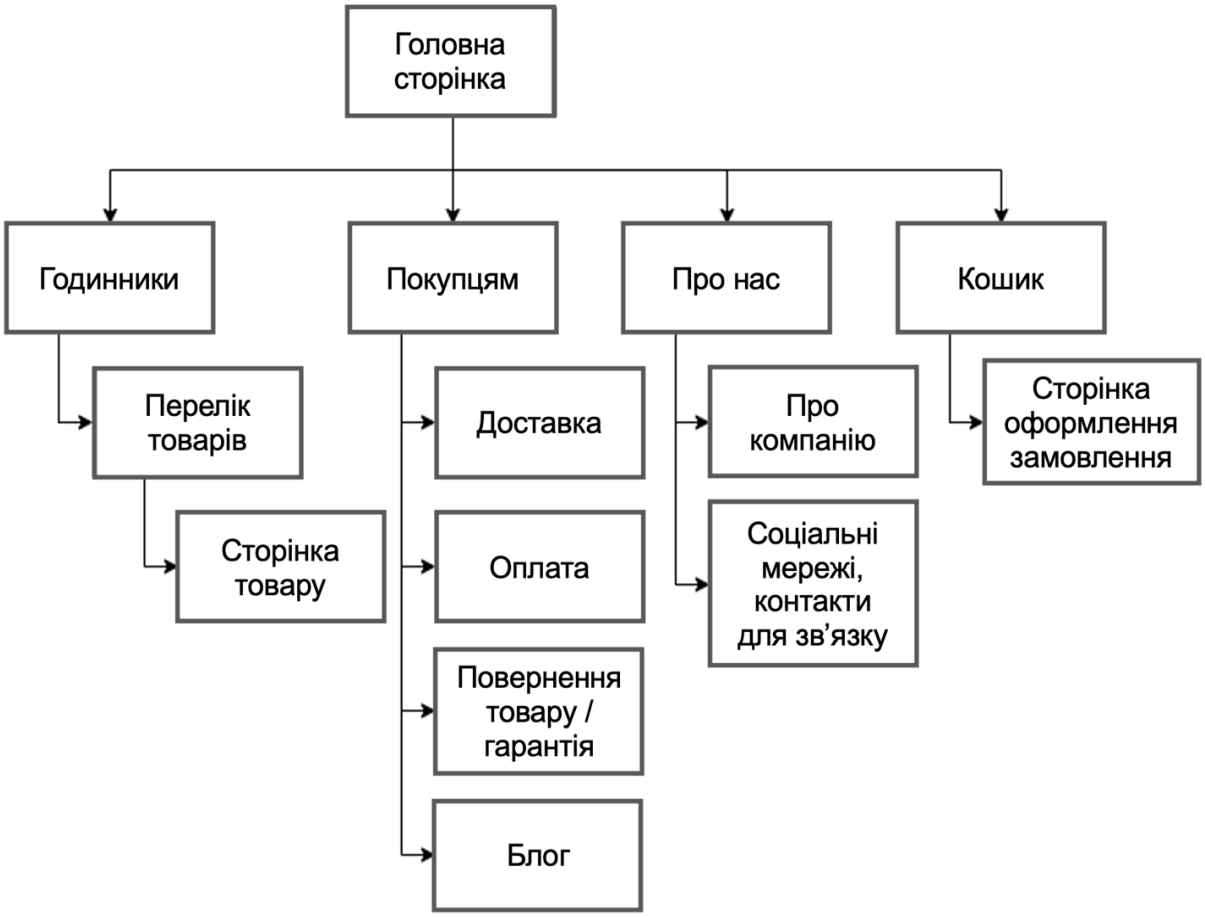

Рисунок 4.2 – Навігаційний дизайн вебсайту інтернет-магазину годинників

Щоб створити вебсайт інтернет-магазину годинників доцільно використовувати MVC-архітектуру, оскільки сучасні фреймворки для розроблення підтримують саме її. Використання MVC дає змогу спрощено інтегрувати функції захисту вебсайту, а також зменшує час і витрати на розроблення системи захисту. На рисунку 4.3 представлена схема MVC-архітектури для вебсайту інтернет-магазину.

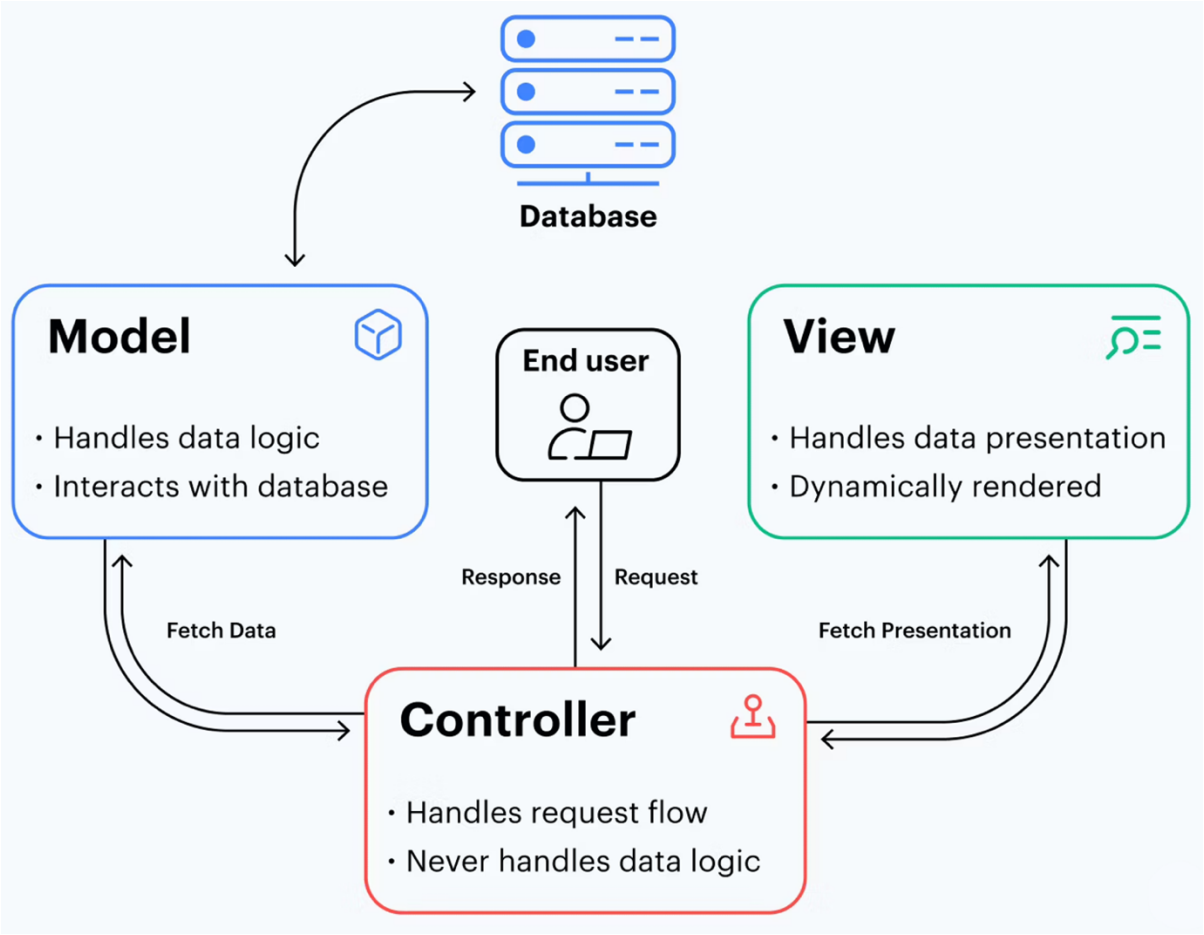

Рисунок 4.3 – Схема MVC-архітектури для вебсайту інтернет-магазину

Для створення вебсайту найкраще підійдуть фреймворки на основі Python або PHP, оскільки вони пропонують великий набір інструментів для реалізації MVC-архітектури та впровадження механізмів захисту.

Було обрано фреймворк Django, тому що він був створений для забезпечення максимальної швидкості розроблення додатків. Це дає змогу зробити процес розробки оперативнішим, оскільки є шаблонізатори та міграції баз даних. Фреймворк забезпечує захисту від XSS-атак, CSRF, SQL-ін’єкцій тощо.

Програмування на певній мові вимагає наявності відповідного середовища розробки. Для мови Python таким середовищем є IDE PyCharm, що призначена для розроблення сучасного програмного забезпечення. Ця IDE включає зручний інструментарій для створення програмних додатків

різного рівня складності. Також це середовище дає змогу розробити зручний інтерфейс користувача, динамічно прослідкувати хід виконання програми, надає розробникові можливість навігації за кодом [24].

Система управління базами даних (СУБД) MySQL була взята за основу для серверної частини додатку. MySQL відомий у всьому світі як найбезпечніша та найнадійніша СУБД, що використовується в таких популярних вебдодатках, як YouTube, WordPress, Joomla, Yahoo!, Google, X (Twitter) та Facebook [25]. Захист даних та підтримання обробки транзакцій, які супроводжують останню версію MySQL, можуть мати велику користь для будь-якого бізнесу.

Для простішої взаємодії з БД застосовується MySQL Workbench. Привабливою властивістю MySQL є те, що це вільна СУБД із відкритим кодом. Ця система створена у якості альтернативи до комерційних. Вона призначена для створення динамічних вебсторінок і має великий набір функцій, що допомагають забезпечити безпеку. Система також має високу швидкість, стійкість і легкість використання.

Django дає можливість застосовувати міграції для взаємодії з базою даних. Таблиці БД, які використовуються в ІС магазину годинників, були прописані, як моделі Django. Вони описують структуру даних, що використовуються. Ці дані зберігаються в БД, а взаємодія з нею здійснюється за допомогою моделей. За замовчуванням СУБД SQLite використовується як база даних для Django [26], проте для цієї інформаційної системи було вирішено використовувати MySQL.

4.2 Результати апробації методу розроблення вебсайту інтернет-магазину, зважаючи на заходи безпеки

Під час розроблення вебсайту інтернет-магазину годинників застосовано метод, що враховує захист від можливих загроз. У процесі роботи наведено приклади діаграм, які використовуються при розробці сайтів із застосуванням цього методу. Окрім цього, було визначено перелік основних ймовірних атак на вебсайт інтернет-магазину та впроваджено відповідні заходи захисту, наведено приклад обрання ПЗ для створення додатку. Вибір здійснювався на основі таких критеріїв, як зручність розроблення, наявність функціональних можливостей, підтримання різних програмних компонентів, БД та вбудованих механізмів захисту від зламу.

До того ж, було обрано архітектуру сайту, представлено її діаграму та враховано можливості розроблення системи безпеки для інтернет-магазину. Для забезпечення правильного доступу до ресурсу, є можливість розподілити ролі користувачів та обмежити доступ у випадку, якщо права доступу не відповідають певній ролі. Приклади сторінок вебсайту інтернет-магазину годинників наведено на рисунках 4.4 – 4.7.

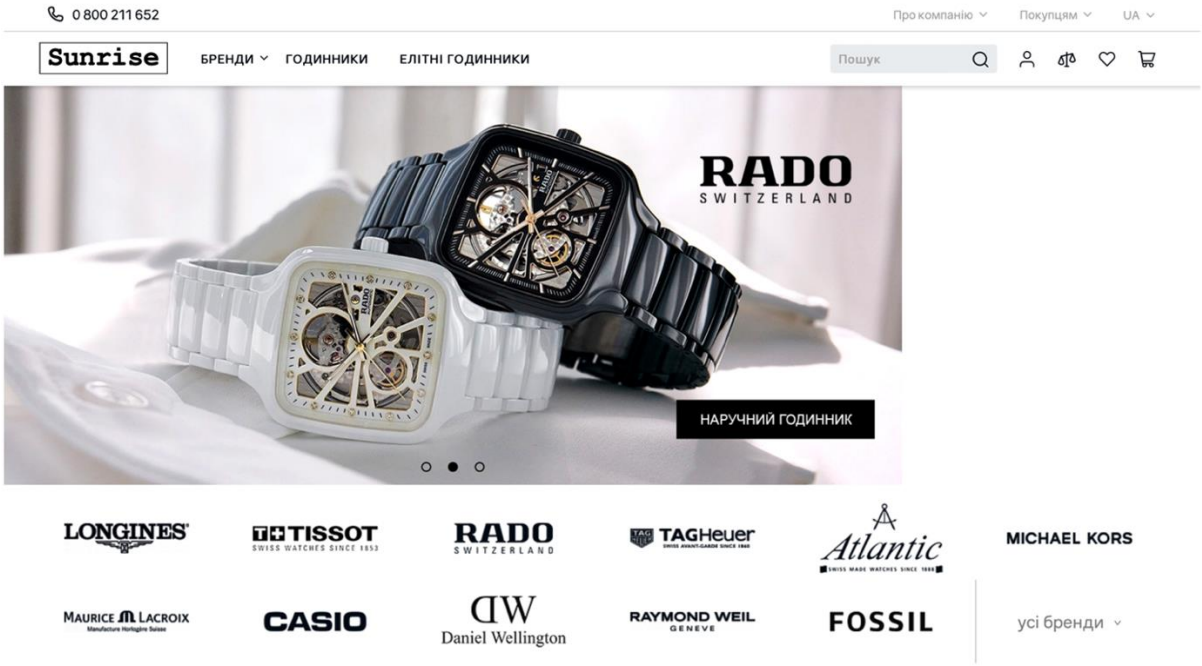

Рисунок 4.4 – Головна сторінка

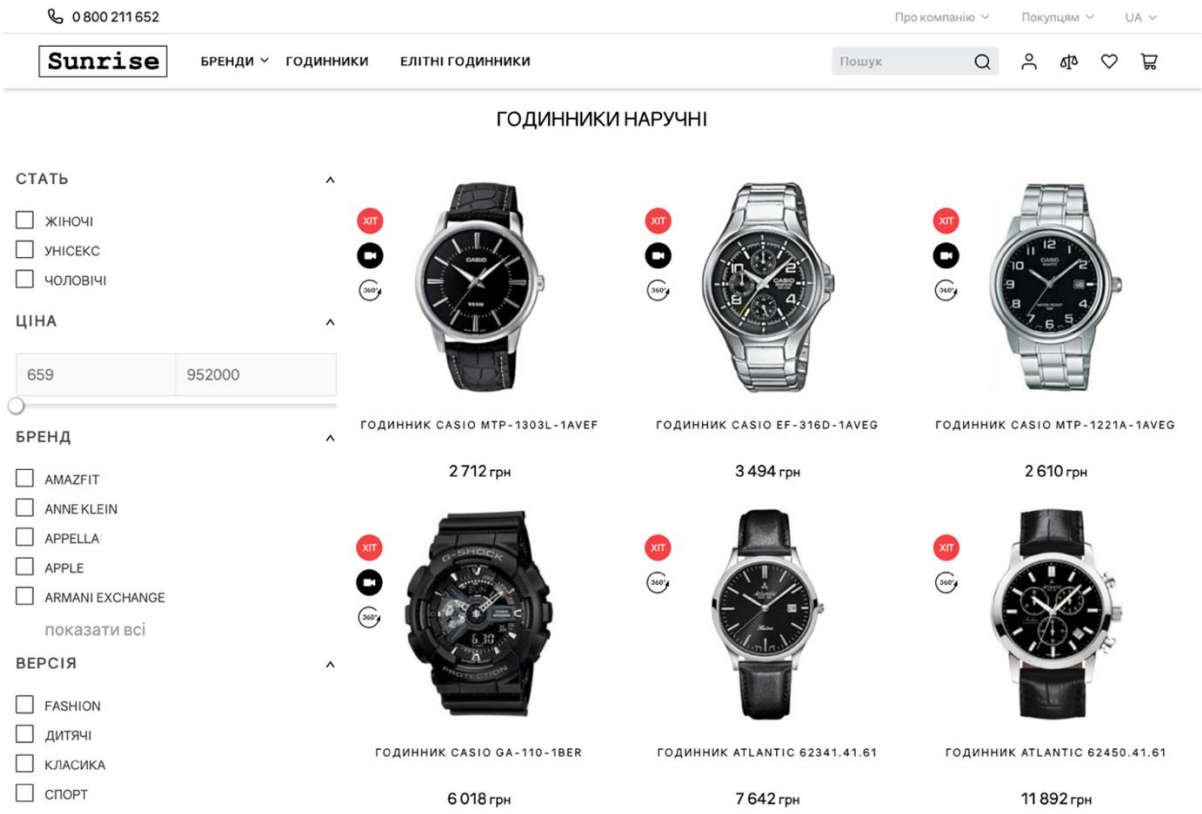

Рисунок 4.5 – Сторінка «Годинники»

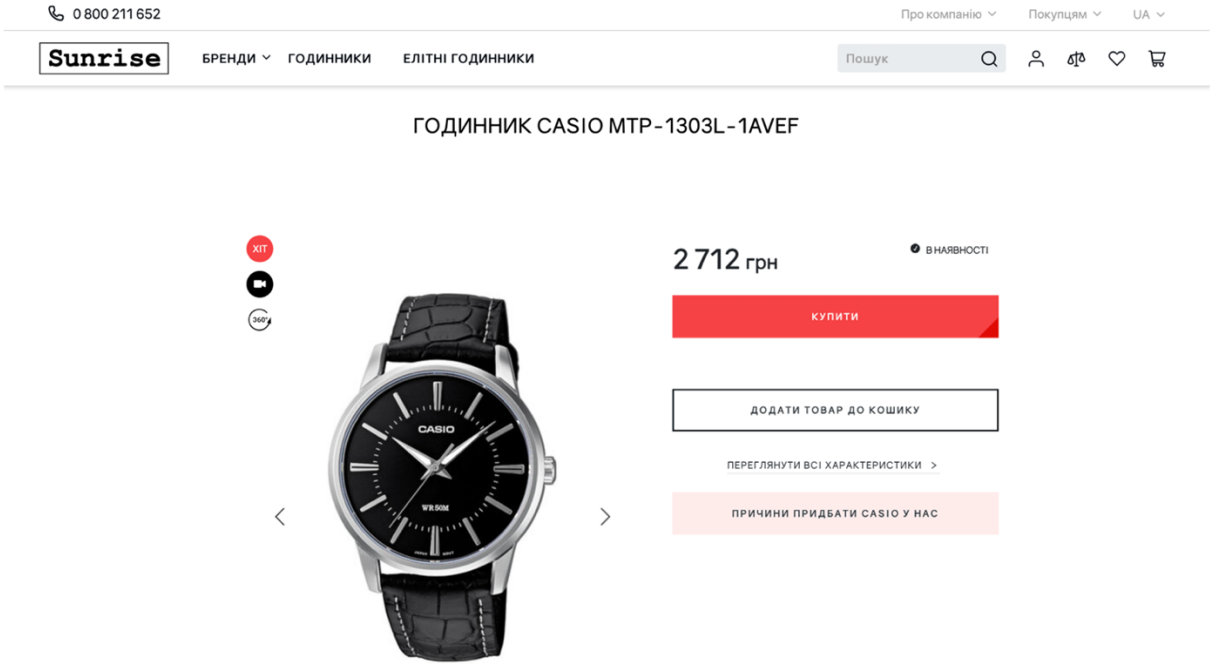

Рисунок 4.6 – Сторінка товару

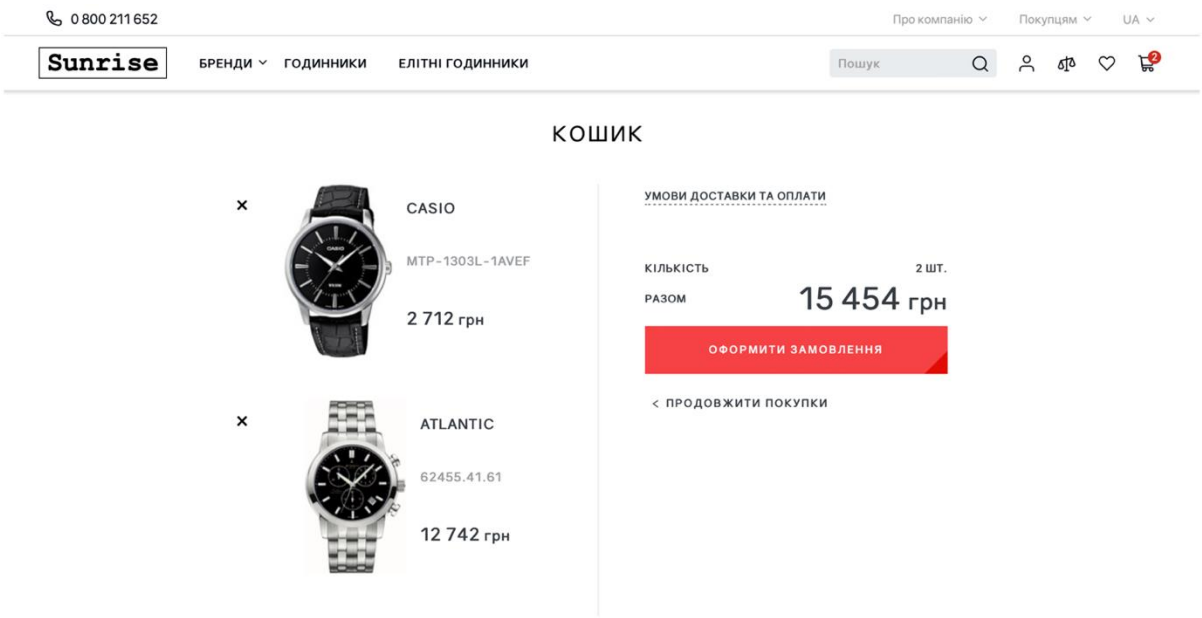

Рисунок 4.7 – Сторінка кошику

ВИСНОВКИ

Виконання кваліфікаційної роботи було присвячене дослідженню методів розроблення вебсайту інтернет-магазину.

Мета роботи – дослідити наявні методи розроблення вебсайту інтернет-магазину, запропонувати модифікований метод розроблення, що забезпечує вищий рівень безпеки та функціональності.

Для досягнення поставленої мети було проведено детальний аналіз наявних методів розроблення вебсайтів, розглянуто сучасні підходи до інтеграції функцій захисту в процесі розроблення. Окрему увагу було приділено дослідженню найпоширеніших типів кібератак, які становлять серйозну загрозу для ІС торговельних компаній.

У роботі створено модифікований метод розроблення вебсайту інтернет-магазину, що враховує безпекові аспекти на всіх етапах: від формування вимог до етапу підтримки. Проведено аналіз і вибір інструментальних засобів для розроблення, які забезпечують інтеграцію сучасних функцій безпеки.

Крім того, було запропоновано концептуальний дизайн вебсайту, створено навігаційний дизайн і обрано архітектурну модель, яка дає змогу забезпечити гнучкість, масштабованість і високу продуктивність системи.

Апробація модифікованого методу була виконана на прикладі створення вебсайту інтернет-магазину годинників. У ході реалізації методу розроблено функціональну структуру, протестовано впроваджені заходи захисту, а також оцінено ефективність обраних інструментальних засобів.

Загалом, результати роботи підтвердили, що запропонований метод розроблення дає змогу значно підвищити рівень захисту ІС, знизити ризик витоку даних і створити функціональну, надійну платформу для бізнесу. Отримані результати можуть бути використані як основа для розроблення інших вебсайтів із підвищеними вимогами до безпеки.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Методичні вказівки щодо розробки та оформлення магістерської атестаційної роботи за спеціальністю 122 Комп'ютерні науки (освітня програма «Інформаційні управляючі системи та технології» освітньо-кваліфікаційного рівня «магістр») / Упоряд.: Петров К. Е., Левикін В. М., Чалий С. Ф., Євланов М. В., Саєнко В. І., Міхнов Д. К., Міхнова А. В., Чала О. В. – Харків: ХНУРЕ, 2022. – 28 с.

2. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура і правила оформлювання. – Чинний від 22.06.2015. – Київ: ДП «УкрНДНЦ», 2016. – 31 с.

3. 27-й Міжнародний молодіжний форум «Радіоелектроніка та молодь у XXI столітті». Збірник матеріалів форуму. Т. 6., Ч. І. – Харків: ХНУРЕ. 2023. – 420 с.

4. Шалева О. І. Електронна комерція [Навчальний посібник] // Центр учбової літератури, Київ. – 2011. – 15 с.

5. Svetlana Gordiyenko. Website Development Process: Full Guide in 7 Steps. XBSoftware. URL: https://xbsoftware.com/blog/website-development-process-full-guide (дата звернення: 15.12.2024).

6. Катерина Войнова. Методи розробки Web-сайтів. Google Sites. URL: https://sites.google.com/site/tz5103voinovakateryna/metodi-rozrobki-web-sajtiv (дата звернення: 16.12.2024).

7. Deane Barker. Web Content Management [Текст] / Deane Barker // O'REILLY, Sebastopol, – 2016. – 45 с.

8. Miguel Grinberg. Flask Web Development [Текст] / Miguel Grinberg // O'REILLY, Sebastopol, – 2014. – 20 с.

9. Django Web Framework (Python). MDN Web Docs. URL: https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django (дата звернення: 22.12.2024).

10. Technology. What is Flask? BairesDevBlog. URL: https://www.bairesdev.com/blog/what-is-flask/ (дата звернення: 22.12.2024).

11. Java Code Geeks. Spring Framework Cookbook [Текст] / Java Code Geeks // Exelixis Media. – 2017. – 76 с.

12. Kathleen Casey. What is Software as a Service (SaaS)? TechTarget. URL: https://www.techtarget.com/searchcloudcomputing/definition/Software-as-a-Service (дата звернення: 23.12.2024).

13. Levykin, V., Ievlanov, M., Neumyvakina, O., Levykin, I., Nakonechny, A. (2024). Estimation of IT-project efforts for information system creation in the conditions of re-use of its functions. Eastern-European Journal of Enterprise Technologies, 2 (2 (128)), 6–19. https://doi.org/10.15587/1729-4061.2024.3012272.

14. Заполочний А. Д. Дослідження методів розроблення вебсайту інтернет-магазину // 13-та Міжнародна науково-технічна конференція «Інформаційні системи та технології ІСТ-2024». Збірник матеріалів форуму. Ч. 2. – Харків: ХНУРЕ, 2024. – 30 с.

15. André L.S. Domingues. A Comparison Study of Web Development Methods / André L.S., Sandro L. Bianchini, Reginaldo Ré, Fabiano C. Ferrari // WebMedia, Brazil. – 2008. – 4 с.

16. Dr. Kovita. Modelling Techniques of Web Architecture for Improvement of Web Applications [Текст] / Dr. Kovita, Sonia Sachdeva // International Journal of Emerging Technology and Advanced Engineering. – 2017. – 225с.

17. Ingle D.R. Hybrid Analysis and Design Model for Building Web Information // International Journal of Computer Science Issues, 2012. – 518с.

18. Cross-Site Request Forgery (CSRF). Owasp. URL: https://owasp.org/www-community/attacks/csrf (дата звернення: 25.12.2024).

19. Cross-Site Request Scripting (XSS). Owasp. URL: https://owasp.org/www-community/attacks/xss/ (дата звернення: 25.12.2024).

20. SQL-ін’єкції та захист від них. Foxminded. URL:

https://foxminded.ua/sql-iniektsii/ (дата звернення: 26.12.2024).

21. Clickjacking. CQR Company. URL: https://cqr.company/web-vulnerabilities/clickjacking/ (дата звернення: 26.12.2024).

22. DoS-атака. Wikipedia. URL: https://uk.wikipedia.org/wiki/DoS-атака (дата звернення: 27.12.2024).

23. Що таке DDoS-атака. Microsoft. URL: https://www.microsoft.com/uk-ua/security/business/security-101/what-is-a-ddos-attack (дата звернення: 27.12.2024).

24. PyCharm Features overview. JetBrains. URL: https://www.jetbrains.com/pycharm/features/ (дата звернення: 03.01.2025).

25. Що таке MySQL. FREEhost.com.ua. URL: https://freehost.com.ua/ukr/faq/wiki/chto-takoe-mysql/ (дата звернення: 04.01.2025).

26. Django 4. SQL Lite. Створення моделей та міграція БД. S-нотатник. URL: https://stas.net.ua/django-sql-lite/ (дата звернення: 05.01.2025).

27. ДСТУ 8302:2015. Інформація та документація. Бібліографічні посилання. Загальні положення та правила складання. – Чинний від 04.03.2016. – Київ: ДП «УкрНДНЦ», 2016. – 20 с.