

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ІНЖЕНЕРНИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ім. Ю.М. Потебні
ЗАПОРІЗЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ
КАФЕДРА ЕЛЕКТРОНІКИ, ІНФОРМАЦІЙНИХ СИСТЕМ ТА
ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Кваліфікаційна робота

перший (бакалаврський)

(рівень вищої освіти)

на тему **Розробка Інтернет-магазину для продажу комп'ютерних товарів з використанням Java Script та PHP**

Виконала: студентка 4 курсу, групи 6.1210-пзс спеціальності 121 Інженерія програмного забезпечення

(код і назва спеціальності)

освітньої програми Програмне забезпечення систем

(код і назва освітньої програми)

Д.В.Нестеренко

(ініціали та прізвище)

Керівник доцент каф. ЕІС та ПЗ, к.ф.-м.н., доцент
І.А.Скрипник

(посада, вчене звання, науковий ступінь, підпис, ініціали та прізвище)

Рецензент директор ТОВ «Дискус»

Р.О. Лютий

(посада, вчене звання, науковий ступінь, підпис, ініціали та прізвище)

Запоріжжя
2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ІНЖЕНЕРНИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ім. Ю.М. Потебні
ЗАПОРІЗЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ

Кафедра електроніки, інформаційних систем та програмного забезпечення

Рівень вищої освіти перший (бакалавський)

Спеціальність 121 Інженерія програмного забезпечення
(код та назва)

Освітня програма Програмне забезпечення систем
(код та назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри Тетяна КРИТСЬКА
“ 01 ” березня 2024 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

Нестеренко Діані Владиславівні

(прізвище, ім'я, по батькові)

1. Тема роботи: Розробка Інтернет-магазину для продажу компютерних товарів з використанням Java Script та PHP

керівник роботи Скрипник Ірина Анатоліївна, к.ф.-м.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від 26.12.2023 № 2215-с

2. Строк подання студентом кваліфікаційної роботи 07.06.2024

3. Вихідні дані кваліфікаційної роботи

- комплект нормативних документів ;
- технічне завдання до роботи.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

- огляд та збір літератури стосовно теми кваліфікаційної роботи;
- огляд та аналіз існуючих рішень та аналогів;
- опис функціональних вимог та архітектури системи;
- створення програмного продукту та його опис;
- перелік технічних та програмних вимог для роботи програми;
- тестування програмної системи та розробка висновків та пропозицій.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

слайдів презентації

6. Консультанти розділів бакалаврської роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата
		Завдання прийняв

7. Дата видачі завдання 01.03.2024

КАЛЕНДАРНИЙ ПЛАН

з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Аналіз предметної області	01.03-10.03.24	виконано
2	Формулювання основної задачі дипломної роботи та узгодження її з науковим керівником	11.03-12.03.24	виконано
3	Аналіз існуючих веб-застосунків для продажу комп'ютерних товарів	13.03-14.03.24	виконано
4	Вибір технологій для розробки веб-застосунку	15.03-20.03.24	виконано
5	Проектування архітектури веб-застосунку	21.03-26.03.24	виконано
6	Узгодження подальших дій з науковим керівником	27.03-28.03.24	виконано
7	Розробка бази даних для застосунку	29.03-13.04.24	виконано
8	Розробка бекенду застосунку з використанням PHP	14.04-16.04.24	виконано
9	Розробка фронтенду застосунку з використанням JavaScript/HTML	17.04-19.04.24	виконано
10	Інтеграція фронтенду та бекенду у веб-застосунок	20.04-07.05.24	виконано
11	Тестування веб-застосунку	08.05-12.05.24	виконано
12	Реалізація користувацького інтерфейсу для веб-застосунку	13.05-20.05.24	виконано
13	Оформлення звіту	21.05-07.06.24	виконано

Студентка _____ Д.В.Нестеренко
(підпис) (прізвище та ініціали)

Керівник роботи _____ І.А. Скрипник
(підпис) (прізвище та ініціали)

Нормоконтроль пройдено

Нормоконтролер _____ І.А. Скрипник
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Сторінок — 72

Рисунків — 29

Таблиць — 2

Джерел — 8

Нестеренко Д. В. Розробка Інтернет-магазину для продажу компютерних товарів з використанням Java Script та PHP: кваліфікаційна робота бакалавра спеціальності 121 «Інженерія програмного забезпечення» / наук. керівник І. А. Скрипник. Запоріжжя : ЗНУ, 2024. 72 с.

У даній роботі представлено розробку та реалізацію веб-додатку для пошуку та придбання комп'ютерних товарів із використанням вподобань користувача. Застосунок використовує OpenServer та MySQL для забезпечення легкого знаходження необхідних товарів, економії часу на пошук та отримання всієї необхідної інформації швидко та зручно.

Мета роботи полягає в розробці Інтернет-магазину який буде приймати заявки від користувачів на купівлю комп'ютерних товарів, збоку адміністрації своєчасна відповідь користувачам.

Завдання, які треба вирішити в роботі:

- Виконати огляд мов програмування та середовищ розробки для створення web-додатку;
- Проаналізувати функціональні можливості web-додатку;
- Реалізувати web-додаток;
- Протестувати web-додаток на помилки та працездатність;
- Написати кваліфікаційну роботу бакалавра по даній темі та зробити висновок.

Результатом роботи є створення web-додатку.

Ключові слова: *PHP, MYSQL, HTML, JAVASCRIPT, PHPSTORM, CSS, WEB, PHPMYADMIN*

ABSTRACT

Pages — 72

Figures — 29

Tables — 2

Sources — 8

Nesterenko D. V. Development of an Online Store for the Sale of Computer Products Using Java Script and PHP: qualification work of a bachelor's degree in specialty 121 "Software engineering" / science. manager I. A. Skrypnyk. Zaporizhzhia : ZNU, 2024. 72 p.

This paper presents the development and implementation of a web application for searching and purchasing computer goods using user preferences. The application uses OpenServer and MySQL to ensure easy finding of the necessary products, saving time on searching and obtaining all the necessary information quickly and conveniently.

The purpose of the work is to develop an online store that will accept applications from users for the purchase of computer goods, from the administration timely response to users.

Tasks to be solved in the work:

- Review programming languages and development environments for creating a web application;
- Analyze the functionality of the web application;
- Implement the web application;
- Test the web application for errors and performance;
- Write a bachelor's thesis on this topic and draw a conclusion.

The result of the work is the creation of a web application.

Keywords: *PHP, MYSQL, HTML, JAVASCRIPT, PHPSTORM, CSS, WEB, PHPMYADMIN*

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1 ТЕОРЕТИЧНІ ВІДОМОСТІ.....	10
1.1 Дослідження основних понять	10
1.2 Аналіз предметної області	13
1.2.1 Принципи побудови веб-сайту.....	13
1.2.2 Коротка характеристика об'єкту управління	14
1.2.3 Опис предметної області.....	15
1.2 Огляд та аналіз існуючих аналогів.....	18
1.3 Висновок до розділу 1	21
РОЗДІЛ 2 ІНСТРУМЕНТИ РОЗРОБКИ ВЕБ-ДОДАТКУ	22
2.1 Засоби розробки	22
2.1.1 HTML	22
2.1.2 PHP	22
2.1.3 CSS.....	26
2.1.4 Javascript.....	27
2.1.5 Mysql	30
2.1.6 Web Server Apache	35
2.1.7 PHPStorm.....	38
2.1.8 PHPMyAdmin.....	42
2.2 Розробка архітектури додатку	43
2.3 Проектування структури бази даних	45
2.4 Вимоги до інформаційних систем.....	47
2.5 UML-діаграми	48
2.6 Моделювання варіантів використання веб-додатку	49
2.7 Висновок до розділу 2	51
РОЗДІЛ 3 РОЗРОБКА ВЕБ-ДОДАТКУ	52
3.1 Процес розробки веб-додатку.	52
3.2 Опис класів	55

3.3 Тестування веб-додатку	65
3.4 Опис функціонування веб-додатку	66
3.5 Вимоги до інтерфейсу	Помилка! Закладку не визначено.
ВИСНОВКИ.....	71
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	72

ВСТУП

Актуальність теми

У контексті стрімкого розвитку ІТ-сектора та зростаючого попиту на комп'ютерну техніку та аксесуари, створення інтернет-магазину для продажу комп'ютерних товарів стає вкрай актуальним. З огляду на те, що конкуренція у цьому сегменті ринку велика, необхідно реалізувати ефективну та зручну для користувачів систему прийому заявок. Це дозволить покупцям легко знаходити необхідні товари, економити час на пошук та отримувати всю необхідну інформацію швидко та зручно.

Мета і задачі розробки

Метою розробки є створення інтернет-магазину для продажу комп'ютерних товарів, який би максимально задовольняв потреби сучасних споживачів.

Методи, засоби та технології розробки

Для реалізації додатку була обрана клієнт-серверна архітектура, яка надасть змогу одночасного доступу до системи багатьох користувачів. Технологією розробки для цього було обрано PHP Storm, на мові PHP за допомогою CSS, HTML, JAVASCRIPT та базу даних MySql.

Практичне значення одержаних результатів

Результатом виконання дипломної роботи є розроблений додаток, який забезпечує подачу заявок користувача по продукції, яка йому потрібна, а збоку компанії, отримування цих заявок та підтвердження в наявності товару.

Апробація результатів

Результати роботи було представлено на XVI університетській науково-практичній конференції студентів, аспірантів, докторантів і молодих вчених «Молода наука-2024» [8]

Глосарій

Asynchronous Javascript and XML(AJAX) — підхід для розгрузки застосунку відправкою невеликих запитів серверу.

Application Programming Interface(API) — спосіб взаємовикористання однієї програми іншою.

Cascading Style Sheets(CSS) — мова розмітки, використовується для задання стилів та розміщення блоків.

ECMAScript(ES) — розширення до мови JavaScript. Не має засобів для виводу та вводу.

Internet Message Access Protocol(IMAP) — протокол який надає доступ до повідомлень електронної пошти.

Internet Protocol (IP) — об'єднує в одну всесвітню мережу окремі комп'ютерні мережі.

JavaScript Object Notation (JSON) — формат для того, щоб обмінюватися даними між браузером та сервером.

Unified Modeling Language(UML) — мова графічного опису для створення бізнес процесів.

eXtensible Markup Language(XML) — розширення до мови HTML, яке допомагає створювати будь які теги нові.

РОЗДІЛ 1 ТЕОРЕТИЧНІ ВІДОМОСТІ

1.1 Дослідження основних понять

1.1.1 Веб-сайт

Web-сайт це інформаційний ресурс, що складається з пов'язаних між собою гіпертекстових документів (web-сторінок) [1].

Web-сторінка являє собою текстовий файл з розширенням * .html. який містить текстову інформацію і спеціальні команди, що визначають в якому вигляді ця інформація буде відображатися у вікні браузера.

У зв'язку з тим, що web-сторінки складаються лише з текстової інформації, то весь графічний, відео та аудіо контент зберігається на web-сервері у вигляді окремих файлів у відповідних форматах, а в кодї сторінки розміщуються лише вказівки на них. На основі цих вказівок клієнтське програмне забезпечення (Web-браузер) обробляє цю інформацію і виводить користувачеві в звичному форматі.

Найбільш популярні web-формати:

- Відео (* .mp4, *. Ogv, *. Webm)
- Аудіо (* .mp3, * .wav, * .mid)
- Графіка (* .gif, * .jpg, * .png)

Web-сайт розташовується на комп'ютерах з встановленим спеціальним програмним забезпеченням і називаються вони Web-серверами.

Зручність використання web-сайту для покупців полягає в тому, що дізнатися інформацію про товар, послуги, ціни, спеціальних акціях і нових позиціях, вони зможуть, просто зайшовши на сайт фірми в Інтернеті. Там же вони зможуть більше дізнатися про компанії, залишити заявку на дзвінок з сайту, записатися на послугу та ін.

Дуже зручний той факт, що на сайті користувач зможе отримати інформацію про товари і послуги цілодобово.

Так само до переваг даного каналу збуту можна віднести: значне зниження витрат, пов'язаних з обміном інформацією за рахунок використання більш дешевих засобів комунікацій, диспетчерську службу і значне збільшення оперативності отримання інформації.

Веб-сайт зможе так само приносити додатковий дохід при продажу продукції через мережу Інтернет покупцям, що знаходяться в інших містах київської області. Компанія націлена на надання послуг в містах України, і здатна зацікавити клієнта, відносно невисокими цінами, широким асортиментом, швидкістю обробки замовлень, швидким виготовленням і доставкою продукції.

У власності компанії є вся необхідне обладнання і додаткові приміщення, необхідний товарний запас, тому введення нової послуги підніме імідж фірми, знизить витрати на одиницю продукції, що продається, забезпечить підйом продажів за рахунок залучення нових клієнтів і забезпечить зручність постійним і потенційним покупцям.

1.1.2 Класифікація веб-сайтів

Кількість сторінок у Всесвітній павутині продовжує зростати. Комерційні та некомерційні організації та навчальні заклади розробляють власні сайти, а звичайним користувачам Інтернету не складає труднощів створення персональних сторінок і повноцінних Інтернет-ресурсів.

Багато роботи було зроблено над класифікацією веб-сайтів, щоб полегшити розробку та визначення їх основних функцій і характеристик. Однак загальноприйнятої системи класифікації в даний час не існує. Розглянемо основні типи сайтів, які характеризуються певним функціоналом.

Типи веб-сайтів залежно від призначення:

Комерційні - «Обличчя» компанії, що взаємодіє з широким колом потенційних клієнтів (покупців, клієнтів і т.д.). Основною метою є досягнення

максимального фінансового прибутку, який може бути виражений як у прямому збільшенні доходу компанії, так і в збільшенні інших комерційних показників.

Некомерційний – веб-сайт, основна мета якого не є комерційною.

Тип веб-сайту за доступністю:

- Відкритий – доступний для всіх відвідувачів Напіввідкритий – для доступу потрібна реєстрація
- Закритий – доступний для вузького кола людей Тип веб-сайту за функціональністю та стилем оформлення:
- Статичний – веб-сайти створюються зі статичних HTML-сторінки. Ці сторінки з'єднані одна з одною за допомогою гіперпосилань. Створена вами сторінка буде збережена на сервері. Користувачі отримують сайт у незмінному вигляді.
- Динаміс – головною функцією є можливість зберігати різні частини сторінки в різних файлах і базах даних. При виведенні користувачеві запитана веб-сторінка генерується в ту ж мить.

Тип веб-сайту на основі фізичного розташування:

- Загальнодоступний – доступний для користувачів Інтернету.
- Локальний – доступний лише в локальній мережі. Це можуть бути корпоративні сторінки вашої організації або особисті сторінки в локальній мережі провайдера.

Типи сайтів за категоріями характеристик відображення інформації та завданнями, які вирішуються:

- Інтернет-представництва організацій
- Інформаційні ресурси

- Веб-сервіси (сайти, що пропонують послуги для виконання конкретних завдань)
- Соціальні мережі (комплексний веб).

Відповідно до характеристик застосованого макета Фіксована ширина - ширина елемента сторінки не масштабується при зміні ширини вікна. Коли використовується тема з фіксованою шириною, ширина сторінок вашого сайту призначається в абсолютних значеннях (у пікселях).

Змінна ширина (гумова версія). Конструкція змінної ширини означає, що ширина сторінки адаптується до ширини вікна. Для цього значення ширини сторінки та її елементів (наприклад, таблиць) задаються у відсотках.

Розтягвані дизайни встановлюють відносні розміри елементів і шрифтів. Під час використання еластичної теми зміна ширини вікна змінює розмір елементів і шрифтів.

Гібридний дизайн. Характерною рисою гібридних конструкцій є те, що деякі ширини елементів визначаються відносними значеннями, а деякі — відсотками. Використовується під час налаштування з кількома стовпцями. При цьому ширина сторінки адаптується до ширини вікна.

1.2 Аналіз предметної області

1.2.1 Принципи побудови веб-сайту

Дослідження ринку показують, що належним чином створений і розроблений веб-сайт є одним із найважливіших засобів підвищення ефективності ваших торгових інструментів і привернення уваги потенційних покупців, а маркетологи роблять це успішно. В основі використання Інтернет-ресурсів та інших маркетингових інструментів для залучення клієнтів лежить принцип прямого зворотного зв'язку. Тому, щоб створити веб-сайт, метою

якого є щось продати, першою і найважливішою умовою є можливість зацікавити відвідувачів і спонукати їх до подальших дій.

Не дивлячись на очевидний факт, багато веб-розробників ігнорують цей момент. В результаті відвідувачі сайту залишають ресурс після перегляду головної сторінки і не цікавляться її змістом. Тому навіть веб-сайт, вміст якого містить багато корисної інформації (публікації, статті, практичні поради), може ніколи не досягти належного рівня відвідуваності, на який очікують його власники. Тому продажі того чи іншого товару можуть не тільки не зрости, як очікувалося, але й істотно знизитися.

Потрібно лише кілька невеликих, але важливих змін, щоб перетворити ваш звичайний веб-сайт на справді ефективний і надійний маркетинговий інструмент. Кожен кваліфікований веб-розробник повинен усвідомлювати, що в епоху інтерактивних технологій кожен користувач Інтернету, як потенційний споживач, стикається з величезною кількістю інформації різного роду, особливо рекламних повідомлень. Тому в сфері маркетингу існує жорстка конкуренція, і залишатися конкурентоспроможними можна тільки в тому випадку, якщо ви зможете залучити потенційних споживачів, створивши дійсно якісний сайт. Тому сайт повинен мати не тільки привабливий та яскравий дизайн, а й містити корисну та цікаву інформацію для потенційних споживачів.

1.2.2 Коротка характеристика об'єкту управління

У сучасному цифровому світі інтернет-магазини займають ключову позицію в роздрібній торгівлі, особливо у сегменті продажу комп'ютерних товарів. Ці платформи стають все більш популярними завдяки зручності шопінгу онлайн, широкому асортименту продукції та можливості порівняння цін і характеристик товарів без необхідності відвідування фізичних магазинів. Інтернет-магазини, що спеціалізуються на продажу комп'ютерної техніки та аксесуарів, забезпечують важливу інфраструктуру для задоволення зростаючих потреб споживачів у високоякісній та інноваційній продукції.

Ці електронні торгові платформи відіграють вирішальну роль у розвитку електронної комерції, надаючи можливості для малих та середніх підприємств досягати широкого ринку без необхідності значних інвестицій у фізичні торгові точки. З використанням сучасних технологій, таких як JavaScript для фронтенду та PHP для бекенду, інтернет-магазини можуть пропонувати високий рівень інтерактивності та персоналізації, поліпшуючи користувацький досвід та збільшуючи лояльність клієнтів.

Нормативно-правова база для діяльності інтернет-магазинів включає законодавство про захист прав споживачів, електронну комерцію, захист даних та інші регуляторні акти, що забезпечують надійну та прозору взаємодію між покупцями та продавцями.

Основними викликами для розвитку інтернет-магазинів комп'ютерної техніки є забезпечення безпеки транзакцій, ефективне управління ланцюгами поставок, а також надання високоякісного сервісу та підтримки клієнтів. З іншого боку, цифровізація комерційної діяльності відкриває нові можливості для інновацій, оптимізації процесів та розширення ринку.

Враховуючи це, інтернет-магазин для продажу комп'ютерних товарів з використанням JavaScript та PHP повинен не лише задовольняти поточні потреби ринку, але й адаптуватися до швидкозмінних тенденцій у технологіях та споживчих уподобаннях, щоб забезпечити стійке зростання та конкурентоспроможність у майбутньому.

1.2.3 Опис предметної області

Оскільки кожен покупець може купити декілька товарів, то у системі будуть відбуватися такі бізнес-процеси:

- авторизація;
- ідентифікація користувача;
- формування списку замовлень;
- робота в системі;
- ідентифікація користувача;

- відображення замовлень;
- додавання категорій;
- додавання товарів.

Для того, щоб розпочати роботу з додатком, користувач повинен перейти за посиланням. Після переходу безпосередньо у додаток, користувач повинен обрати бажаний товар та його кількість. Після цього він повинен перейти до кошика та оформити замовлення. Додати своє ім'я, прізвище та по-батькові та номер телефону для того щоб власник або оператор приватної молочної компанії зміг зв'язатися з клієнтом. Власник приватної молочної компанії повинен пройти авторизацію на сайті, після цього його буде перенаправлено на сторінку адміністрування на якій він може бачити категорії, товари та список замовлень користувачів. Для підтвердження замовлення він має зателефонувати користувачеві, та підтвердити замовлення.

На рисунку 1.1 зображено діаграму бізнес-процесів системи.

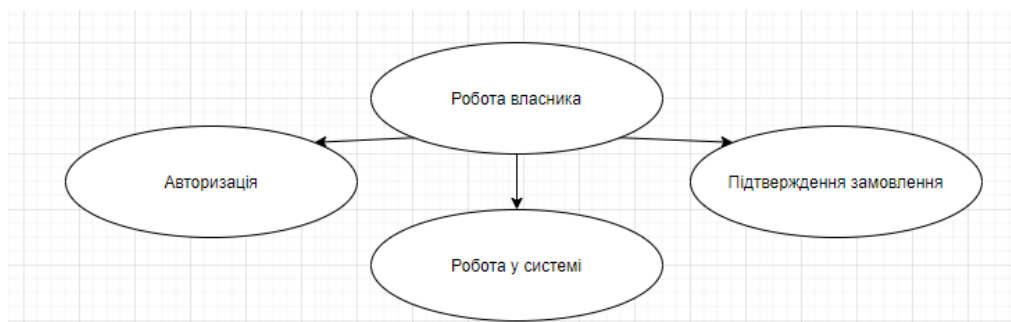


Рисунок 1.1 — Діаграма ієрархії бізнес-процесів предметної області

На рисунку 1.2 зображено діаграму структури функції бізнес-процесу «Авторизація».



Рисунок 1.2 — Діаграма дерева функцій бізнес-процесу «Авторизація»

Характеристика бізнес-процесу «Авторизація» представлена у таблиці 1.1.

Таблиця 1.1 — Характеристика бізнес-процесу «Авторизація»

Назва характеристики	Значення характеристики
Ім'я бізнес-процесу	Авторизація
Основні учасники	Власник
Вхідна подія	Запуск форми входу
Вхідний документ	Логін, пароль
Вихідна подія	Відображення списку замовлень
Вихідні документи	Список замовлень
Клієнт бізнес	Власник

На рисунку 1.3 зображено діаграму структури функції бізнес-процесу «Робота в системі»



Рисунок 1.3 — Діаграма дерева функцій бізнес-процесу «Робота в системі»

Таблиця 1.2 Характеристика бізнес-процесу «Робота в системі»

Назва характеристики	Значення характеристики
Ім'я бізнес-процесу	Робота у системі
Основні учасники	Власник
Вхідна подія	Успішна авторизація
Вхідний документ	Логін, пароль
Вихідна подія	Відображення списку замовлень, редагування товарів, редагування категорій
Вихідні документи	Список замовлень
Клієнт бізнес	Власник

1.2 Огляд та аналіз існуючих аналогів

Через те що аналогів дуже багато я опишу лише декілька з них. На Рисунку 1.4 здійснюю пошук аналогів. Всі аналоги можна знайти за джерелом [17].

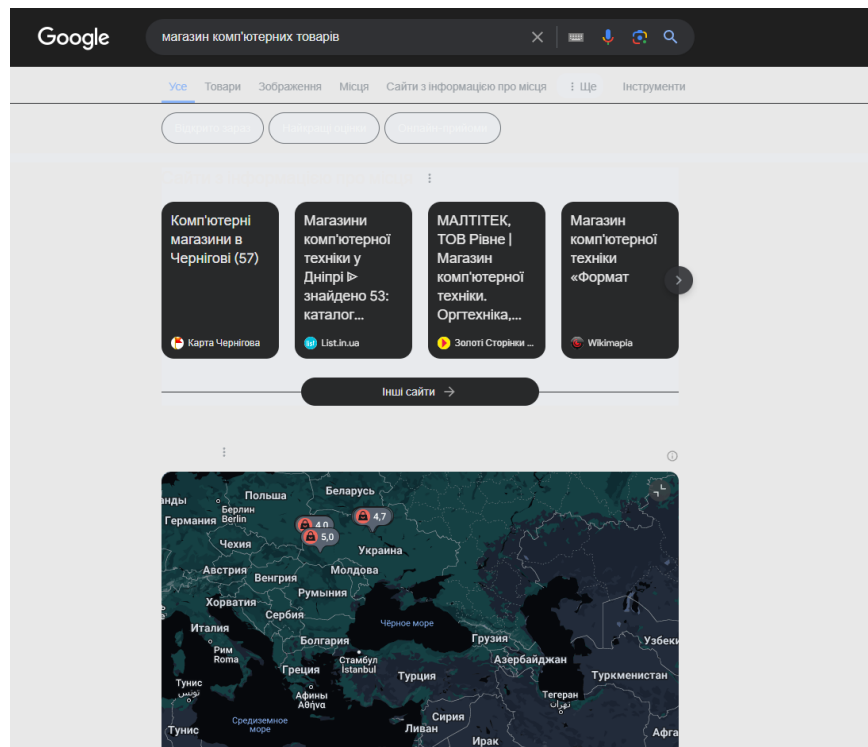


Рисунок 1.4 — Запит для пошуку

1. «CompX»:

CompX — web-додаток для покупки комп'ютерних товарів.

Багатоцільовий web-додаток, основними функціями якого є обробка заявок, власний кабінет та організація продажу товарів [2].

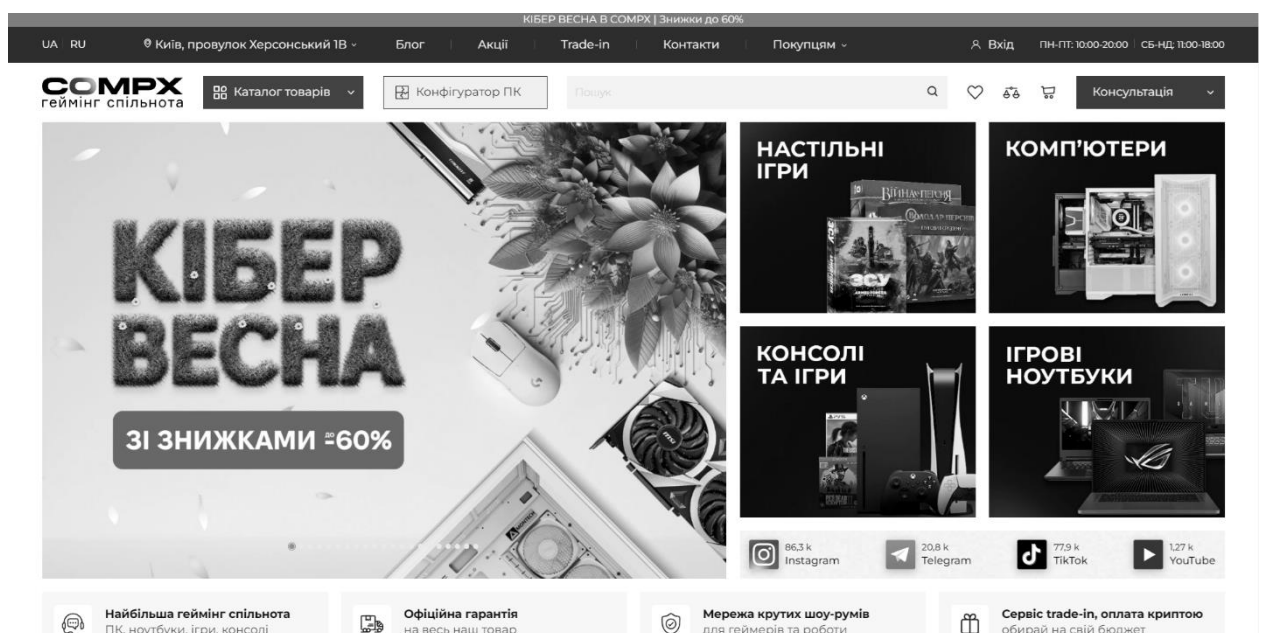


Рисунок 1.5 — Web-додаток «CompX»

2. «Мою»:

Мою — це web-додаток, який дозволяє не тільки купувати комп'ютерні товари, але й інші товари.

Він вважається багатоціловим тому що у нього входить не тільки обробка заявок з приводу комп'ютерних товарів але й заявок різноманітної продукції. Тому у цьому web-додатку є такі можливості як «власний кабінет» тощо [10].

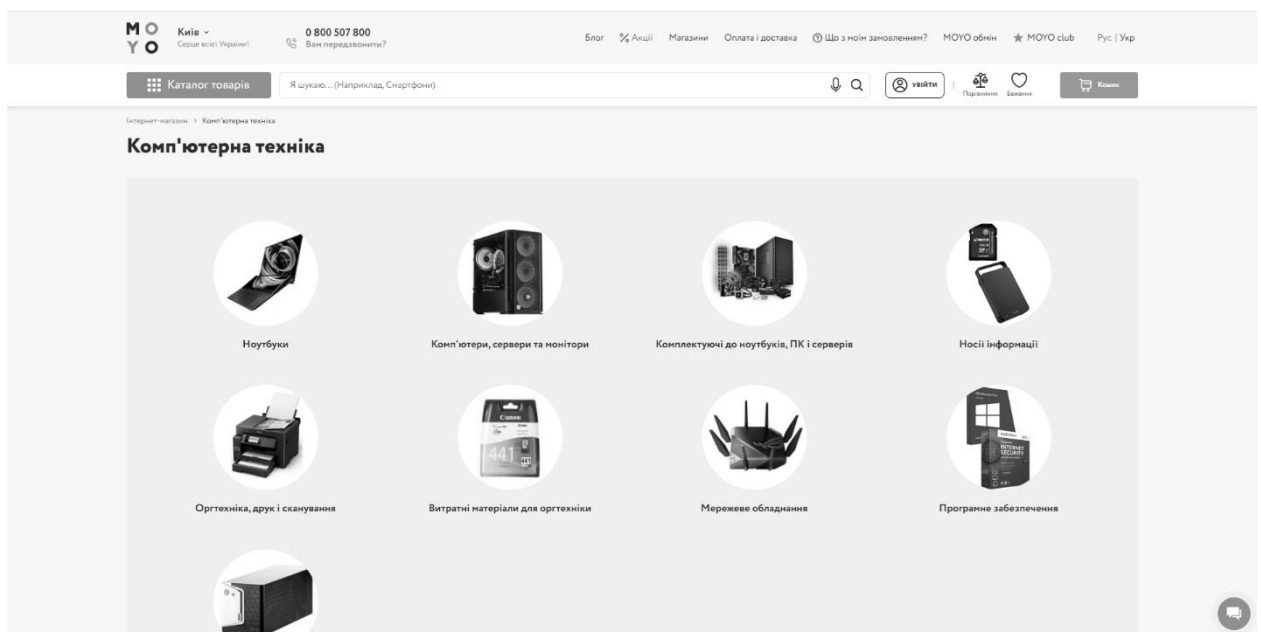


Рисунок 1.6 — Web-додаток «Мою»

3. «Rozetka»:

Rozetka — це дистриб'ютор продукції, який дозволяє отримувати ексклюзивні пропозиції від різноматніх виробників продукції, та створювати заявки, але головний недолік цього web-додатку у тому що немає телефонної підтримки [14].

ВНАСЛІДОК ПОШКОДЖЕНЬ ІНФРАСТРУКТУРИ ПІД ЧАС ОБСТРІЛУ ТЕРМІНИ ВІДПРАВКИ МОЖУТЬ БУТИ ЗБІЛЬШЕНІ

ROZETKA Каталог Я ищу... Найти UA RU

Ноутбуки и компьютеры
 Смартфоны, ТВ и электроника
 Товары для геймеров
 Бытовая техника
 Товары для дома
 Инструменты и автотовары
 Сантехника и ремонт
 Дача, сад и огород
 Спорт и увлечения
 Одежда, обувь и украшения
 Красота и здоровье
 Детские товары
 Зоотовары
 Офис, школа, книги
 Алкогольные напитки и продукты
 Бытовая химия
 Продолжаем уценку до -50%

Справочный центр
 Чат с ROZETKA
 Магазины Rozetka

Никита nikita@99@gmail.com
 Персональные предложения
 Слоски желаний
 Корзина

Офіційний партнер

Це легко. Це сильно. Це M3.
 Тільки до 24 березня

В оплату частинами на 15 платежів,
 всього від 3 567 ₪/міс.

MacBook Air

Все акції 252

Последние просмотренные товары

Бойлер Midea D100-20ED0 (D) Wi-Fi Prime 15 362 ₪	Verbatim BD-R DL 50 GB 6x Cake 10 шт (43746) 1 629 ₪ Нет в наличии	Verbatim BD-R SL Datalife 25 GB 6x Spindle 1.660 1 479 ₪ Нет в наличии	Комплект Georplast фильтров для туалета 120 ₪	Закрытый туалет с фильтром для котлов 516 ₪ Нет в наличии	Подвесная оконная кровать для кота 799 ₪

Рисунок 1.7 — Web-додаток «Rozetka»

1.3 Висновок до розділу 1

1. Наведено коротку характеристику об'єкту дослідження та визначено склад функцій, що входять до основних бізнес-процесів предметної області та повинні бути автоматизованими.
2. Проведено огляд та аналіз кількох веб-систем аналогів, що реалізують схожі функції предметної області.
3. Проведено етап аналізу вимог до розроблюваного додатку.

РОЗДІЛ 2 ІНСТРУМЕНТИ РОЗРОБКИ ВЕБ-ДОДАТКУ

2.1 Засоби розробки

2.1.1 HTML

Мова розмітки гіпертексту, або більш відома як HTML, є комп'ютерною мовою, яка є основою Всесвітньої павутини. За допомогою мови HTML будь-який текст можна розмітити, перетворити на гіпертекст, а потім опублікувати в Інтернеті.

Мова HTML має власний набір символів, які веб-браузери використовують для відображення сторінок. Ці символи, прийняті як дескриптори і містять елементи, які потрібні для створення гіперпосилання [3, с. 20].

Сам цей документ зберігає лиш текст, а усе інше має зберігатися під час відображення браузером(це решта об'єктів, які особливими тегами вбудовані в документ). Це і є однією з вігоміших рис документів HTML. Коли ви зберігаєте HTML-файл, там де розташовується документ створюється папка, у якій розміщуються супровідні елементи графічного оформлення [2, с. 14].

2.1.2 PHP

По-перше, PHP використовується для написання скриптів, які працюють на стороні сервера, власне, тому він і був придуманий. PHP може вирішувати ті ж завдання, що й який завгодно CGI-скрипт, включаючи обробку даних у html-формах, динамічну генерацію html-сторінок тощо. Але PHP можна використовувати і в інших сферах.

Друга область полягає у створенні сценаріїв, які запускаються в командному рядку. Тобто за допомогою PHP можна створити такий скрипт, який можна буде виконати на конкретному комп'ютері незалежно від веб-сервера або браузера. Останньою сферою є створення GUI-додатків (графічних інтерфейсів), які відбуваються на стороні клієнта [3, с. 58].

З 1995 року почалася Історія PHP, тоді Расмус Лердорф створив просту програму Perl для аналізу користувачів, які переглядають його резюме на веб-сайті. Пізніше, коли додатком вже користувалися кілька людей, а бажаючих його отримати ставало все більше, Лердорф назвав свій витвір Personal English Home Page Tool. Profile Tools версія 1. Яка стала доступна для безкоштовного завантаження. З цього моменту популярність PHP зросла як ніколи раніше.

Але як правило, швидко потрібні зміни та нові доповнення. Щоб реалізувати їх, Расмус зробив нову версію пакета, але вже написану мовою C. Одержаний такий інструмент отримав робочу назву PHP/FI (укр. Персональна домашня сторінка/інтерпретатор форм), у майбутньому також називатиметься PHP/FI (укр. Персональна домашня сторінка/інтерпретатор форм) під назвою PHP 2. Ця версія стала більш родібною до тієї яка зараз є у PHP. Має синтаксис як у стилі мови Perl і спосіб іменування змінних. Версія впроваджує оператори PHP у HTML-код сторінки, автоматично інтерпретувати форми та інтегруватися з базами даних. Крім того, оскільки PHP скомпільований у веб-сервер Apache, усе працює доволі непогано та моторно. Раніше PHP використовувався на 50 000 доменах до 1997 року(менше 99% від усіх веб-серверів) [4].

У тому ж 1997 році Зеєв Сураскі та Енді Гутманс приєдналися до проекту PHP. Вони навчаючись у Техніон-Ізраїльському технологічному інституті, спробували використати PHP/FI для бізнес-проекту в університеті. У той же час вони натрапили на труднощі та обмеження технології. Зробили заключення під час вивчення вихідного коду PHP 2, що PHP необхідно вдосконалити, а також, суттєво змінити дещо в частині синтаксису мови. За кілька місяців Зев та Енді чудово впоралися з цими питаннями[4].

Після завершення роботи Зев і Енді погодилися співпрацювати з Расмусом у сфері розвитку та вдосконалення мови. З того моменту виникла PHP Group - команда однодумців, присвячених розвитку технології PHP. Кінцевий продукт був випущений у 1998 році як PHP 3.

А найголовнішою перевагою PHP 3 стала можливість розширення ядра, на що звернули увагу багато сторонніх розробників, які проектували вузькопрямовані модулі для розробки PHP. Їх інтеграція надає PHP можливість взаємодіяти з вагомою кількістю баз даних, протоколів і підтримувати велику кількість API. До закінчення 1998 року кількість споживачів PHP зросла до 100000, і PHP був впроваджений на більш ніж 10% серверів Інтернету. У той же час публікації в електронній пресі та видання посібників з PHP також відобразили вплив на широке розповсюдження мови.

Енді Гутманс і Зев Сураскі почали робити редизайн ядра одразу коли відбувся випуск PHP. По-перше, потрібно було вирішити питання з ростом продуктивності. Продукт, що щойно вийшов під назвою Zend Engine (від імен його творців: Zeev і Andi) чудово впорався з цим завданням, тому його реалізували в 1999 році. Головною метою є вміння компілювати скрипти у виконувани модулі, завдяки чому продуктивність може бути покращена в рази.

PHP 3 та PHP 4

1997 року Зев Сураскі та Ані Гутманс змінили синтаксичний аналізатор та лягли в основу PHP 3, переробивши назву мови на рекурсивну аббревіатуру PHP: Hypertext Preprocessor. Відтоді PHP 3 розпочало активне тестування та було офіційно випущено в червні 1998 року. Згодом Сураскі та Гутманс зробили перекодування ядра PHP і випустили Zend Engine у 1999 році. В Ізраїлі Рамат-Гані так само випустили Zend Technologies.

PHP 4 був впроваджений 22 травня 2000 року з підтримкою Zend Engine 1.0. Результатом на серпень 2008 року ця гілка здобула версію 4.4.9. PHP 4 наразі не підтримується, і жодних оновлень безпеки не випущено.

PHP 5

14 липня 2004 PHP 5 був оприлюднений з новим Zend Engine II. PHP 5 містить нові можливості, такі як оновлена підтримка об'єктно-орієнтованого програмування, розширення PHP Data Objects (PDO), які забезпечують зрозумілий і узгоджений інтерфейс для доступу до баз даних, а також нараховуються покращення продуктивності. PHP 5 стає єдиною стабільною

версією у 2008 році, що має постійні розробки. Пізніше статичне зв'язування не було наявне в PHP і було додано у версії 5.3.

Станом на 5 лютого 2008 року численна кількість відомих проектів з відкритим вихідним кодом припинили підтримку PHP 4 у створенні коду через ініціативу GoPHP5, запущену Альянсом розробників PHP для полегшення переходу від PHP 4 до PHP 5.

Через деякий час інтерпретатори PHP відкрили доступ багатьом існуючим 32-розрядним і 64-розрядним операційним системам, створених із вихідного коду PHP або з використанням заздалегіть зібраних двійкових файлів. Для PHP версій 5.3 і 5.4 одним лише існуючим двонаправленим дистрибутивом Microsoft Windows є 32-розрядна версія x86, яка потребує 32-розрядного режиму сумісності з Windows під час потребування інформаційних служб Інтернету (IIS) на 64-розрядних платформах Windows. Версія PHP 5.5 дає 64-розрядній версії x86-64 можливість для використання Microsoft Windows.

PHP 6 та Unicode

Через проблему нестачі підтримки Unicode на рівні базової мови PHP отримав різні відгуки, що не дуже позитивні. У 2005 році під керівництвом Андрія Змієвського було розпочато проект із впровадження підтримки Unicode у PHP способом вбудовування бібліотеки International Components for Unicode (ICU) і впровадження текстових рядків як UTF-16. Через те, що це включатиме значні зміни у внутрішню структуру мови та код користувача, її мали запровадити як версію 6.0 мови поміж з окремими основними функціями, що знаходились на етапі становлення.

Однак затримки проекту були спричинені відсутністю розробників, які усвідомлювали складності продуктивності, необхідні для перетворення на UTF-16 (який використовується не часто у веб-середовищі). Тому версія PHP 5.3 була випущена в 2009 році з різними функціями, відмінними від Unicode, відновленими з PHP 6, включаючи простори імен. Проект у його поточній формі було залишено повністю у березні 2010 року, і було запропоновано

випуск PHP 5.4, який мав у своєму складі більшість випущених не-Юнікодових функцій у PHP 6, таких як тире та прив'язка до закриття. Спочатку мали надію розробити новий план інтеграції з Unicode, але станом на 2014 рік це ще не було впроваджено у дію.

PHP 7

Упродовж 2014 і 2015 років була написана свіжа головна версія PHP під номером PHP 7. Відлік цього релізу поклав початок суперечкам[5]. Принаймі версія PHP 6 з Юнікодом ніколи не була опрелюднена, деякі статті та назви книг згадують назву PHP 6, що може завдавати заплутування, якщо нова версія це хоч раз згадує назву [6]. З назвою визначились лише після голосування і вона була позначена як PHP 7 [7].

2.1.3 CSS

Коли згадується вираз CSS ми усвідомлюємо фразу «каскадні таблиці стилів». Вони реалізовані в коді HTML і не потребують спеціальних редакторів і компіляторів, оскільки інтерпретуються звичиними для нас в повсякденності браузерами.

Каскадні таблиці стилів — це описи всіляких елементів HTML, які робляться для створення розширень атрибутів останніх. Спочатку цей стиль був згаданий консорціумом WWW задля експлуатації специфікації HTML 3.0, але його фактично не погоджували до 1997 року. Тільки задоволені володарі таких браузерів, як Netscape Navigator 4.0 і Internet Explorer 4.0, мали шанс на користування CSS.

Щодо цього питання, дозвольте мені відразу перейти до суті, поєднавши історичні факти. На цьому для нас CSS відчиняє новітні, передовсім незнайомі та недоступні кордони. Зі вдяки ним ми зараз можемо з легкістю зробити наш сайт більш елегантним, надавши йому новий функціонал і гарний зовнішній вигляд.

Вивчити синтаксис CSS дуже легко, тому оволодіти каскадними стилями є дуже легким завданням, проте насамперед одним найцінніших на

етапі розробки зовнішнього вигляду застосунку. Щоб не залишитися безмовним, хотілося б навести нечисленний приклад. Давайте візьмемо звичну таблицю HTML і спробуємо встановимо атрибут `border = "2"`. Після того, як браузер інтерпретує цей код, ми отримуємо найпоширеніші результати, до яких звик наш погляд. Друга річ, що такі таблиці не сумісний ні з яким більш-менш хорошим дизайном. Отже, для чого ця межа підходить? CSS в цьому може допомогти, завдяки ньому можна досить просто й легко оформити будь-яку таблицю, що забажаєте.

2.1.4 Javascript

Прийнято називати JavaScript однією з широко використовуваних прототипно-орієнтованих мов програмування сценаріїв. Цей код можна виконати як на стороні клієнта, так і на стороні сервера. Практично немає ніякого новітнього сайту, який би не використовував цю мову програмування. JavaScript був запроваджений Netscape, щоб додати інтерактивність веб-сторінок.

Найпопулярніші області застосування для програм JavaScript:

- Використання сценаріїв для динамічного створення документів;
- Попередня перевірка введених користувачем даних перед передачею на сервер;
- Створення динамічних HTML-сторінок;
- Взаємодія з користувачами при вирішенні «місцевих» завдань.

Поява в Netscape

У 1995 році Компанія Netscape мала на меті завдання впровадити мову програмування Scheme або «щось подібне» в браузер Netscape. Тому робітники запросили американського розробника Брендан Аїк, який спеціалізується на системному програмуванні. Крім того, щоб пришвидшити розвиток, Netscape почав співпрацювати з Sun Microsystems.

Через деякий час концепція впровадженної мови програмування розширилася до перспективи її використання прямо в HTML-кодi сторінки. Мета двох компаній полягала в тому, щоб випустити мову, яка могла б з'єднати різні елементи веб-сайту: зображення, аплети Java, об'єктну модель документа. Метою мови мала стати зручність для веб-дизайнерів і недосвідчених програмістів. Назва нової мови, яка було в розробці на той момент становила Mocha, яку пізніше замінили на LiveScript у перших двох бета-версіях браузера Netscape 2.0. Незабаром після цього LiveScript було перейменовано на JavaScript, а саме через перебування на піку бренду Java, і була випущена третя бета-версія Netscape 2.0 (2.0B3) під сучасною назвою [6, 7]. Щоб таке впровадити компанії довелося придбати ліцензію у компанії Sun Microsystems, яка саме і була володарем бренду Java.

Пізніше компанія Nombas у 1992 році впровадила скриптову мову програмування Cmm (англ. C-minus-minus), якій згодом змінили назву на ScriptEase та яка вмiла вбудовуватися в вебсторінки. Тримається така некоректна думка, що JavaScript розробили під впливом Cmm. Але в дійсності Брендан Аїк казав, що раніше не помічав про Cmm до того, як він створив LiveScript[8]. Трохи згодом, Nombas залишили розробку Cmm та розпочали приймати участь у групі зі стандартизації JavaScript й використовувати її.

Стандартизація

У листопаді 1996 року компанія Netscape оголосила, що подала JavaScript до Ecma International для дослідження в якості галузевого стандарту. Подальша праця призвела до появи стандартизованої мови під назвою ECMAScript У червні 1997 року Ecma International оприлюднила перше видання специфікації ECMA-262 Потім були затосовані певні редагування, щоб привести специфікацію у відповідність до стандарту ISO/IEC-16262 і Друге видання було опубліковане через рік, у червні 1998 року. Третє видання було опубліковане в грудні 1999 року.

Робота над четвертою редакцією стандарту ECMAScript так і не була завершена, і четверта редакція так і не була випущена [10]. Однак, п'ята редакція була опублікована в грудні 2009 року.

Шоста версія була випущена в червні 2015 року [11], після чого комітет ECMAScript вирішив робити оновлення кожного року, і остання на той час версія отримала назву ES2015. У цій версії було впроваджено чимало нововведень, зокрема об'єкт Promise для асинхронного виконання коду, деструктурування присвоєнь, шаблонні рядки, функції зі стрілкою, генеруючі функції, оператори оголошення змінних let і const тощо.

Версія ES2016, випущена в червні 2016 року [11], додала інноваційні перспективи, такі як оператор піднесення до степеня** та метод Array.prototype.includes, що має на меті перевірку, чи переданий аргумент міститься в масиві.

У версії ES2017 [11], випущеній у червні 2017 року, було додано змогу користуватися асинхронними функціями, "висячі" коми в параметрах функцій, об'єкти Atomics та кілька інших подальших методів для того щоб якісно проводити роботу з рядками.

Версія ES2018 була випущена в червні 2018 року [12] і включає в себе змогу для асинхронної ітерації, інструмент Spread(...) для зручного користування об'єктами та масивами, було додано деякі засоби для роботи з регулярними виразами, а також метод Promise.prototype.finally, який викликається, коли Promise досягає статусу "завершено".

Версія ES2019 була випущена в червні 2019 року [13], і її оновлення включають новий тип даних Symbol, деякі методи обробки рядків і масивів, а також можливість переробити об'єкт у масив за допомогою Entries і навпаки.

Поточною версією є ES2020, що була випущена в червні 2020 року [14]. У цій версії введено новий тип даних BigInt, оператор, який позначається двома знаками питання, що слугує для перевірок на нулі та невизначеність, об'єкт globalThis, спроба використовувати необов'язкові значення в об'єктах, динамічний імпорт, метод String.prototype.matchAll для того щоб легко за

допомогою регулярних виразів здійснювати пошук рядків, а також до Promise було додано метод `allSettled`, який викликається після виконання всіх Промісів.

Актуальний стан

На сьогоднішній день JavaScript вважається дуже розповсюдженою мовою програмування в Інтернеті [15]. На початку свого існування більша кількість з усіх професійних програмістів ставилися до мови з настороженістю, а до складу її цільової аудиторії входили програмісти-аматори [16]; поява AJAX виправила обставини і привернула до мови увагу професійної публіки, а наступні зміни відповідно до стандарту ES6+ ефективно ввели безліч корисних можливостей, котрих раніше було брак для ефективного програмування. Тому підсумовуючи було розроблено та вдосконалено багато практик роботи з JavaScript (включаючи тестування та налагодження), впроваджено бібліотеки та фреймворки, а використання JavaScript за межами браузера набуло широкого розповсюдження.

2.1.5 Mysql

MySQL це така база даних з відкритим вихідним кодом, яка зараз на слуху у багатьох (дехто вважає її найкращою у світі). В дійсності, MySQL все більше конкурує з коштовними великими базами даних, такими як Oracle і Microsoft SQL Server.

MySQL була розроблена і нині має підтримку від шведської компанії MySQL AB (веб-сайт www.mysql.com). певною мірою MySQL розвинулася з раніше розробленої бази даних mSQL, популярність якої вже пройшла, але все ще підтримується . Цікаво, що жодній з багатьох книзі не пояснюється, як тлумачиться аббревіатура MySQL. З кінцівкою назви, "SQL", все зрозуміло. Це мова структурованих запитів, мова, яка призначена для роботи з більшістю баз даних які існують зараз. Однак початок назви "My" публічно не має розшифровки, та й навіть всередині компанії [3, с. 10].

MySQL - це реляційна система управління базами даних (СКБД). Якщо поглянути з технічної сторони, MySQL - це програма, яка керує файлами, що створюють базу даних, але в багатьох випадках термін "база даних" має застосування не лише до програми, а й до набору файлів. База даних - це лише набір даних (текст, числа, час життя), які поєднані один з одним, за збереження та організацію яких бере участь СУБД.

Існує безліч типів баз даних, починаючи з не складних, де дані підтримується у плоских файлах, закінчуючи реляційними та об'єктно-орієнтованими базами даних. Реляційні бази даних мають змогу зберігати інформацію одразу про декілька таблиць одночасно. До початку 70-х років бази даних були схожі на гігантські електронні таблиці, де зберігалося усе до купи, але на допомогу була розроблена ця концепція. Реляційні бази даних вимагають більше сил та наполегливості при розробці, але окупність є значною через підвищену надійність і цілісність даних.

MySQL, як і PHP та деякі версії UNIX, є програмою з відкритим вихідним кодом. Це дає розуміння, що програму можна встановити і запустити без вкладень, а вихідний код можна змінювати (як і саму MySQL, її можна запустити з Інтернету). Однак є випадки, коли вам може знадобитися оплатити ліцензію на MySQL, наприклад, якщо ви хочете інтегрувати MySQL у свій продукт з метою отримання прибутку. Для отримання додаткової інформації про це див. Ліцензійні умови MySQL [3, с. 11].

MySQL має у своєму складі декілька частин, включаючи сервер MySQL (програма MySQL, яка фактично керує базою даних), клієнт MySQL (програма MySQL, яка надає інтерфейс до сервера) та низку утиліт, які використовуються для підтримання бази даних та інших цілей. Для роботи з MySQL можна використовувати розповсюджені мови програмування, такі як PHP, Perl та Java.

Ключові моменти в розвитку MySQL

Початкова розробка розпочалася в 1994 році MySQL Майклом Віденіусом Девідом Аксмарком (Michael Widenius David Axmark).

Вперше оприлюднений запуск становив 23 травня 1995 року.

Версія MySQL для операційних систем Windows була опублікована для Windows 95 і NT 8 січня 1998 року.

Реліз версії 3.23 відбувся в січні 2001 року.

Версія 4.0 була випущена в березні 2003 року.

Реліз версії 4.01 відбувся у серпні 2003 року.

Реліз версії 4.1 відбувся у жовтні 2004 року.

Версія 5.0 була випущена у жовтні 2005 року.

26 лютого 2008 року Sun Microsystems придбала компанію MySQL AB.

Реліз версії 5.1 відбувся 27 листопада 2008 року.

Версія 5.4 бета - квітень 2009 року не була випущена.

Версія версії 5.5 перестала випускатися в грудні 2010 року.

Версія 6.0 в даний час знаходиться на етапі розробки. Нова шоста версія MySQL в даний час створюється і тестується і буде включати в себе

Цілісність зв'язаних даних з підтримкою зовнішнього ключа для всіх типів сховищ даних

Додаткова підтримка символів Unicode (десь 65 536 символів у BMP (Basic Multilingual Plane)).

З'явився новий тип таблиці для зберігання даних - Falcon (доступний на сайті MySQL).

У жовтні 2005 року Oracle викупила фінську компанію Innobase Oy, яка виробляла InnoDB - універсальну підсистему керування сховищами даних для MySQL.

Після придбання у прес-релізі Oracle було заявлено, те що усі контракти компанії на програмне забезпечення, що були доступні для MySQL AB, будуть поновлені у 2006 р. На конференції користувачів MySQL у квітні 2006 р. основоположник MySQL випустив прес-реліз, в якому MySQL та Innobase Oy оприлюднили, що вони дали згоду про безстрокову ліцензійну угоду.

У лютому 2006 року Oracle купила програмне забезпечення Sleepcat - механізм баз даних, що лежить в основі іншого механізму зберігання даних MySQL.

У жовтні 2009 року корпорація Oracle підписала угоду про придбання з Sun Microsystems, теперішнім власником інтелектуальної власності на MySQL.

Де і коли використовується MySQL

При створенні веб-проекту, який є переважним елементом, більшість користувачів, безсумнівно, будуть використовувати MySQL для різних типів даних.

MySQL доволі широко використовується в поєднанні з PHP.

Значна частина веб-додатків використовується MySQL як програмним компонентом; звісно що популярність веб-додатків насамперед пов'язана з багаторазовим використанням PHP і його підтримкою MySQL.

З гордістю можна сказати, що велика кількість веб-додатків з високим трафіком (Flickr, Facebook, Wikipedia, Google, Nokia, YouTube тощо) використовують MySQL для зберігання даних і реєстрації користувачів.

Найпопулярнішим способом встановлення MySQL є платформа Linux, ширше використовується саме там, але база даних також може бути встановлена на інших платформах, таких як Microsoft Windows.

MySQL може використовуватися на інших платформах у різних системах, включаючи AIX. BSDI. freeBSD. HP-UX. Linux. mac OS X. NetBSD. Nowel NetWare. OpenBSD. OpenSolaris. eComStation. OS/2. Warp. qnx IRIX. Solaris. Symbian. SunOS. SCO OpenServer. SCO UnixWare. Sanos. Tru64. Microsoft Windows.

Існує також порт MySQL для OpenVMS.

Усі мови програмування в яких підтримується API (Application Programming Interface) підтримують бібліотеку для того щоб отримати доступ до бази даних MySQL.

Більше того, інтерфейс ODBC (Open Database Connectivity) під назвою MyODBC дає змогу допоміжним мовам програмування, наприклад ASP і ColdFusion, користуватися базою даних MySQL для підтримки і покращення інтерфейсу ODBC. Сервер MySQL і значна кількість бібліотек є ANSI.

Базою даних MySQL можна керувати за допомогою інструментів командного рядка (команди MySQL і mysqladmin). Користувачі можуть завантажити такі утиліти адміністрування з офіційного вебсайту MySQL.

GUI (графічний інтерфейс користувача) MySQL Administrator MySQL Migration Toolkit і MySQL Query Browser; наразі інструменти GUI згруповані в єдиний пакет під назвою MySQL GUI Tools MySQL Migration Toolkit і MySQL Query Browser.

Хотілось б зауважити, що веб-інтерфейс для адміністрування баз даних MySQL, розроблений на PHP, - phpMyAdmin безкоштовний веб-додаток з відкритим вихідним кодом.

З phpMyAdmin браузер користувача може легко керувати сервером, виконувати SQL-команди і переглядати склад таблиць і даних.

Додаток набув застребуваність серед веб-розробників завдяки тому, що доступ до інтерфейсу дає змогу управляти SQL без необхідності вводити прямо SQL-команди.

Хоча ми вже майже підійшли до кінця статті, щодо систем керування базами даних, будь-який веб-дизайнер, напевно, ознайомлений із веб-інтерфейсом для керування системою керування базами даних MySQL під назвою phpMyAdmin.

Після встановлення програмного забезпечення систему управління базами даних MySQL та phpMyAdmin можна використовувати як на сервері, де була створена база даних, так і на комп'ютері, де наявна можливість створення подібного сервера через програмний пакет (я використовую Denver). phpMyAdmin надає можливість без усіляких зусиль додавати бази даних і користувачів, копіювати таблиці тощо.

2.1.6 Web Server Apache

Apache — це кросплатформне програмне забезпечення, яке має підтримку таких операційних систем, як Linux, BSD, Mac OS і Microsoft Windows.

Одними з вагоміших переваг Apache враховується надійність і гнучкість конфігурації. Ви можете підключатися до зовнішніх модулів, щоб надавати дані, використовувати СУБД для автентифікації користувачів, змінювати повідомлення про помилки тощо.

Ядро Apache містить такі головні функції, наприклад обробка конфігураційних файлів, протокол HTTP та система завантаження модулів. Ядро (не так як модулі) цілком написано Apache Software Foundation без залучення сторонніх веб-розробників [3, с. 42].

Чити в теорії, само ядро Apache має змогу працювати в чистому вигляді без використання модулів. Проте, функціональність таких рішень дуже низька.

Ядро Apache було повністю розроблено мовою C.

Система конфігурації Apache базується на текстових конфігураційних файлах. Існує три рівні конфігурації умов:

1. конфігурація сервера (httpd.conf).
2. конфігурація віртуальних хостів (httpd.conf починаючи з версії 2.2extra / httpd-vhosts.conf).
3. конфігурація на рівні каталогу (.htaccess).

Є власна мова конфігураційних файлів, що базується на основі блоків директив; Майже всі параметри ядра, включаючи управління MPM, можна перетворити за допомогою файлів конфігурації. Більшість модулів мають свої власні параметри.

Apache має в собі рушій віртуального хосту. Це дозволяє підтримувати багато веб-сайтів (доменів) повністю на одній IP-адресі, кожен з яких відображає відповідний контент.

Зараз є ряд модулів, які додають підтримку різних мов програмування та систем розробки в Apache.

До них належать:

1. PHP (mod_php).
2. Python (mod_python)
3. Ruby (apache-ruby)
4. Perl (mod_perl).
5. ASP (Apache-Asp).

Більше того, також Apache вбудовує механізми CGI та FastCGI, і може виконувати програми майже на будь-якій мові програмування, включаючи C, C++, SH та Java.

Веб-сервер Apache ідеально підходить для запуску веб-сайту на стабільній та багатоцільовій платформі, тому це чудовий вибір. Однак він також має свої недоліки і переваги, про які слід знати.

Почнемо з переваг:

- Безкоштовний для комерційного використання.
- Надійне та незмінне програмне забезпечення.
- Регулярні виправлення безпеки, які часто оновлюються.
- Гнучкість за рахунок модульної конструкції.
- Легко налаштувати та підходить для новачків.
- Мульти/багато-платформенний (будь де чудово працює на серверах хоч це Unix, хоч Windows).
- Підтримує сайти WordPress одразу після інсталяції без додаткового налаштування.
- Велика спільнота та легкодоступна підтримка, якщо у вас виникнуть питання.

Тепер перейдемо трішки до недоліків:

- Виникають питання з продуктивністю на зайнятих веб-сайтах.
- Забагато параметрів конфігурації може стати проблемою з вразливістю системи безпеки.

HTTP-сервер Apache був анонсований у 1995 році розробником з Університету Іллінойсу в Урбана-Шампейн (UIUC) Робертом Маккалоу. Цей сервер з'явився як покращена версія іншого клієнта HTTP, до того моменту написаного Робертом, NCSA HTTPd 1.3.

Ціллю для виправлення стала низка «патчів» або програмних «патчів» для NCSA. Це оригінальне походження назви патчі (а не від індіанського племені апачі). Що тлумачиться як «сервер із патчем».

З 1999 року цей продукт розробляється та підтримується Apache Software Foundation (ASF), спільнотою професійних ентузіастів у всьому світі. Ця ж неприбуткова організація створила офіційну ліцензію на програмне забезпечення — Apache License.

У 2000 році ASF опублікувало нову версію Apache 2.0 в якій було усунуто код NCSA і повністю перероблено архітектуру. Відтоді веб-сервери перетворилися на дві основні гілки: 1.x і 2.x.

Apache побудовано за допомогою ядра та динамічної системи модулів. Параметри системи перетворюються за рахунок конфігураційних файлів.

Ядро Apache було розроблено на Смові - Apache Software Foundation. Його головними функціями є обробка конфігураційного файлу, протоколи HTTP/HTTPS і підвантаження модулів. Ядро має змогу виконуватися без модулів, але з обмеженою функціональністю.

Модулі — це незалежні файли, які поширюють первинну функціональність ядра. Модулі можуть бути залучені в програмне забезпечення під час початкової інсталяції або додані пізніше шляхом зміни конфігураційних файлів.

Численна кількість з них створені для конкретних аспектів обробки запитів клієнтів, такі як підтримка багатьох мов програмування, безпека, кешування та автентифікація. Тому великі завдання поділяються на кілька етапів, кожна з яких виконується у своїй вузькоспеціалізованій фазі.

Apache має понад 500 модулів. Велика кількість сучасних веб-додатків швидко аноншуються як модулі в Apache. Прикладами є ISPmanager і VDSmanager.

Текстові файли система конфігурації Apache обробляє з заданими настройками. Сама система розділена на три рівні умов, кожен з яких має власний файл конфігурації.

Рівень конфігурації сервера (файл `httpd.conf`) є основним файлом конфігурації і охоплює весь двигун веб-сервера.

Рівень каталогу (файли `.htaccess`) являє собою додаткові файли конфігурації. Цей крок стосується лише каталогу, де знаходиться файл, і його підкаталогів.

Рівень віртуального хосту (файл `httpd.conf` або додатковий `/httpd-vhosts.conf`).

Файли конфігурації Apache часто розташовані в папці `conf`, а додаткові файли конфігурації знаходяться вкладеними в додаткову папку. Для того, щоб щось змінити треба віредагувати певний файлу або через командний рядок.

Веб-хост — це серверний компонент, який надає послуги окремому об'єкту, який на ньому розмістили (сайтами, віртуальними серверами). Система віртуального хосту Apache пропонує можливість одночасного запуску з однієї IP-адреси декількох проектів.

Apache дозволяє конфігурувати параметри модуля та ядра, а також вводити обмеження споживання ресурсів сервера (трафік, оперативна пам'ять, ЦП) для кожного віртуального хоста індивідуально. Що складає технічну основу всієї системи веб-хостингу.

2.1.7 PHPStorm

JetBrains PhpStorm — комерційне кроссплатформене інтегроване середовище розробки на PHP, написане російською компанією JetBrains на базі платформи IntelliJ IDEA.

PhpStorm — це інтелектуальний редактор для PHP, HTML і JavaScript з оперативним аналізом коду, уникненням помилок у сирому коді та інструментами автоматичного рефакторингу для PHP і JavaScript. Автоматичне завершення коду PhpStorm підтримує специфікації PHP 5.3/5.4/5.5/5.6/7.0/7.1 (найновіші та старіші проекти), маючи в складі генератори, загальні програми, простори імен, закриття, типи та синтаксис невеликих послідовностей.

Має повноцінний редактор SQL, який дозволяє вносити правки в результати вхідних запитів.

PhpStorm випущено на основі платформи IntelliJ IDEA, яка в свою чергу була розроблена на Java. Люди, які мають змогу користуватися ним можуть розширити функціональні можливості свого середовища розробки, встановивши й в подальшому використовуючи плагіни, спеціально розроблені для платформи IntelliJ, але також є можливість самому створити власні плагіни.

Усі функції WebStorm включені в PhpStorm. PhpStorm є досить коштовним і інтелектуальним редактором коду для PHP з підкресленням та виділенням синтаксису, розширеним форматуванням коду, перевіркою помилок у ту ж секунду та інтелектуальним автозавершенням.

Він підтримує PHP 5.3, 5.4 та 5.5 (генератори, загальні програми, простори імен, замикання, типи, короткий синтаксис послідовностей, доступ до членів класу під час створення екземпляра, розіменування послідовностей у викликах функцій, двійкові літерали, статика (включаючи вирази у викликах тощо). Нині ним можна користуватися не лише для сучасних, але й для традиційних PHP-проектів.

Автоматичне завершення коду автоматично може розпізнати і завершити клас, метод, імена змінних, ключові слова PHP, а також загальноживані імена полів і змінних на основі їх типу.

- Підтримка стандартів розробки коду (PSR1/PSR2, Drupal, Symfony2, Zend).

- Підтримка PHPDoc.PhpStorm забезпечує належне автоматичне завершення коду на основі анотацій `@property`, `@method` і `@var`.
- Виявлення дублікатів коду.
- Сніффер коду PHP (`phpcs`), який перевіряє ваш код протягом тих самих секунд.
- Рефакторинг (перейменування, введення змінної/константи/поля, вбудовування змінних).
- Підтримка помливості змінити шаблони Smarty, включаючи підкреслення синтаксичних помилок, автоматичне завершення функцій і атрибутів Smarty, а також автоматичне вставлення подвійних дужок, лапок і закриваючих тегів.
- Види MVC для фреймворків Symfony2 та Yii.
- Виявлення коду, упакованого в архіви PHAR.

Середовище розробки

- Підтримка SQL і бази даних, включаючи рефакторинг схем бази даних, створення сценаріїв міграції схеми, експорт результатів запиту у файл або буфер обміну та редагування збережених процедур.
- Віддалене розгортання програми та постійна синхронізація з FTP, SFTP, FTPS .
- Системи контролю версій (Git — включає особливі функції для роботи з GitHub, Subversion, Mercurial, Perforce, CVS і TFS) дають змогу виконувати безліч дій безпосередньо з PhpStorm, таких як коміти, злиття та відмінності (відстежувати все).
- Діаграма класів UML для коду PHP, яка надає доступ до прямого виклику з діаграми.
- Підтримка Phing (забезпечує автозаповнення, перевірку тегу за замовчуванням, властивості, цільового імені та значень атрибутів шляху у файлах збірки).

- Інтеграція з системами відстеження помилок.
- Підтримка Vagrant, консоль SSH і віддалені засоби.
- Google App Support Engine для PHP.
- PhpStorm також пропонує використовувати різні комбінації клавіш для підвищення продуктивності[2].

Зневадження і тестування

- Досить просто настроювані візуальні відладчики (Xdebug, Zend Debugger) дозволяють бачити контекст локальних змінних і настроюваних об'єктів трасування (включаючи масиви та складні об'єкти) і маніпулювати їхніми значеннями на швидкому рівні.
- Інтеграція Profiler: сценарії можна профілювати безпосередньо з PhpStorm за рахунок Xdebug або Zend Debugger. Доступні зведені(паketні) звіти. Користувачі мають змогу переходити безпосередньо від статистики роботи з функціями у своєму коді PHP.
- Інтеграція з фреймворком модульного тестування PHPUnit із покриттям коду (дають можливість розробляти тести PHPUnit в PhpStorm і негайно запускати з каталогу, файлу чи класу через контекстне меню).

Робота з JavaScript, CSS і HTML

Усі функції, наявні для доступу в WebStorm, включені в PhpStorm. А саме:

- Автоматичне завершення для JavaScript, HTML і CSS коду (теги, ключові слова, мітки, змінні, параметри, функції).
- Підтримка HTML5.
- Редагування в реальному часі: виправлення коду можуть бути негайно відображені в браузері без необхідності на перезавантаження сторінки.
- Підтримка CSS/SASS/SCSS/LESS (автоматичне завершення коду, пікреслення помилок, перевірка тощо).
- Zen Coding.

- Навігація по коду та пошук використання (перехід до декларації/ідентифікатора, використання пошуку).
- Підтримка ECMAScript Harmony.
- Рефакторинг JavaScript (перейменування, призначення змінних/функцій, вбудовані змінні/функції, переміщення/копіювання, безпечне видобування), вилучення вбудованих сценаріїв розділяти файли).
- Інтеграція з дизасемблером JavaScript та фреймворками модульного тестування JavaScript.

2.1.8 PHPMyAdmin

phpMyAdmin — це веб-програма з відкритим вихідним кодом, написана на PHP із графічним веб-інтерфейсом для керування базами даних MySQL або MariaDB. За допомогою phpMyAdmin ви можете керувати вашим сервером MySQL, виконувати SQL-запити, переглядати і редагувати вміст таблиць бази даних з вашого браузера. Ця програма дуже популярна серед веб-програмістів. phpMyAdmin зараз широко використовується на практиці. Адже це пов'язано з тим, що вона дозволяє керувати базами даних MySQL без встановлення додаткового програмного забезпечення та без введення команд SQL з будь-якого комп'ютера, підключеного до Інтернету, через зручний всім інтерфейс.

Ще одна причина популярності, що розробники інтенсивно розробляли продукт з урахуванням усіх інновацій СУБД MySQL. Більшість українських провайдерів користуються цим додатком в якості панелі управління, що дозволяє керувати базами даних, закріпленими за клієнтами.

Оскільки ця програма поширюється за Загальною публічною ліцензією GNU, інші розробники, такі як XAMPP у Денвері, також включили її у свої розробки.

Він був започаткований німецьким програмістом Тобіасом Ратшіллером (Tobias Ratschiller) і, як і phpPgAdmin, заснований на схожому додатку MySQL-Webadmin, що з'явився в 1997 році.

Проект локалізовано 85 мовами.

2.2 Розробка архітектури додатку

Мій додаток буде розроблятися на основі архітектури «клієнт-сервер». Котра є однією з моделей архітектури програмного забезпечення, який є найбільш широко використовуваним поняттям у створенні розподілених мережових додатків і має на меті взаємодію та обмін даними між розподіленими мережевими додатками. Вона включає в себе такі основні компоненти [3]:

- Набір серверів, який надає інформаційні та інші послуги додаткам, які мають доступ до них.
- Група клієнтів, які користуються послуги, що надають їм сервери.
- Мережа, що дозволяє взаємодіяти і обмінюватися інформацією між клієнтами та серверами.

І сервер, і клієнт працюють незалежно і паралельно один від одного. Між клієнтом і сервером немає фіксованого зв'язування. Така ситуація більш імовірна, зазвичай сервер обробляє запити від різних клієнтів в один й той самий момент часу. Клієнти, з іншого боку, можуть підключатися до будь-якого сервера. Вони повинні розуміти й знати про доступні сервери, але не повинні знати про те що щенаявні інші клієнти [3].

На рисунку 2.1 наведено діаграму, яка візуалізує архітектуру розроблюваного web-додатку.

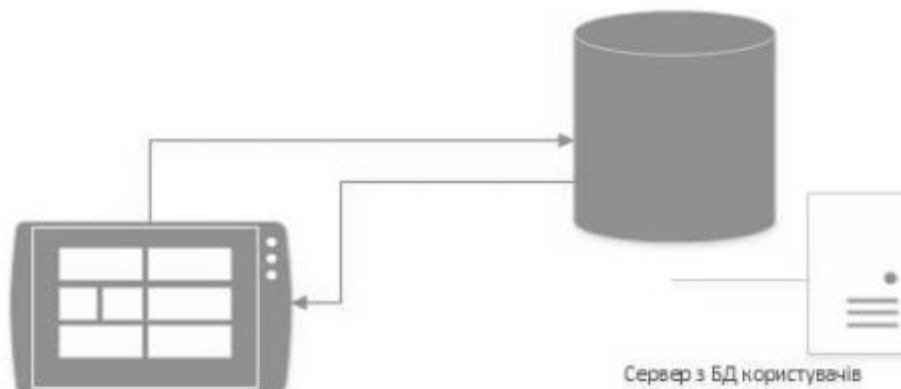


Рисунок 2.1 — Архітектура web-додатку

Система розробки використовує підключення до сервера. По-перше, це база даних зареєстрованих користувачів (покупців і адміністраторів), які програма використовує під час автентифікації (для повернення звітів про те, чи зареєстрований цей користувач у системі та чи є він). Так, які у вас є права доступу (користувач чи адміністратор), зареєстрований користувач у системі чи ні?

Потім схема роботи розробленого додатку детально описана, щоб адміністратор міг переглянути список погляд. Щоб розмістити будь-яке замовлення, користувач повинен спочатку оформити замовлення в системі, вказавши своє ППП, номер телефону, email та товар. Інакше список не буде повним.

Для використання програми користувач (адміністратор або покупець) повинен оформити замовлення (на Рисунку 2.2 наведена діаграма стану процесу «Авторизація»).



Рисунок 2.2 — Діаграма станів процесу «Авторизація»

Після успішної авторизації система, якщо користувач ідентифікований як адміністратор, виведе на екран товари, категорії та список замовлень,, якщо ж він ідентифікований як покупець, то виведе на екран таблицю його замовлень. Далі адміністратору потрібно вибрати товар зі списку, і система покаже йому повну інформацію про товари. На рисунку 2.3 зображено діаграма станів процесу «Перегляд списку замовлень».

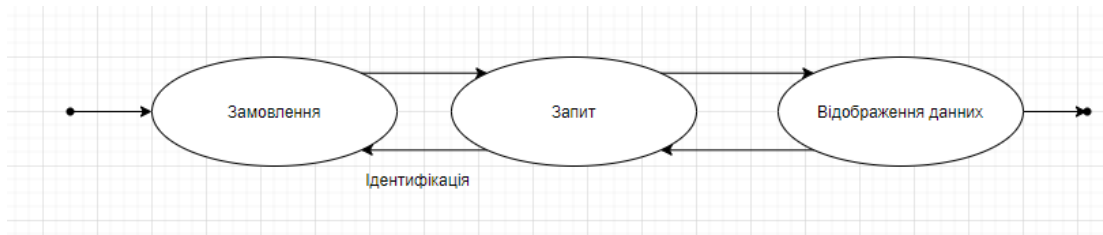


Рисунок 2.3 — Діаграма станів процесу «Перегляд списку замовлень»

Після цього адміністратор може вибрати будь-яке замовлення.

2.3 Проектування структури бази даних

Необхідним кроком у розробці системи є проектування бази даних, яка надійно зберігає дані та дозволяє виконувати операції з даними. Перебуваючи на цьому кроці потрібно вирішити, які дані хочемо зберігати та які операції з ними потрібно виконати. Сформулювати структуру бази даних, включаючи визначення основних сутностей, відповідно до зв'язків. Потім після формування основних блоків між ними повинні бути встановлені логічні зв'язки, щоб уникнути помилок та проблем під час збереження та обробки даних.

Під час автентифікації система намагається відтворити підключення до двох різних серверів. Спочатку знайде користувача в базі, де є інформація про усіх зареєстрованих користувачів. Далі сервер повертає звіт про те, чи зареєстрований цей користувач, який намагається здійснити вхід у систему, і

якщо звіт повернув позитивну відповідь, то далі треба дізнатися з нього які права доступу він має.

Якщо програмі не вдається автентифікувати користувача в базі даних користувачів (оскільки користувач може ще не бути зареєстрованим у цій програмі), і якщо ідентифікація користувача в базі даних користувачів успішна. Використовується діаграма потоку даних (DFD). Подібно до IDEF0, DFD надає змодельовану систему як мережу взаємопов'язаних завдань і може використовуватися для більш наглядного представлення моделей IDEF0. Проаналізувати поточні операції документообігу в корпоративних системах обробки інформації, визначити, як кожне завдання перетворює вхідні дані на виходи, і визначити зв'язки між цими завданнями.

Далі на Рисунку 2.4 формується таблиця ідентифікаторів, яка представляє атрибути, їх тип і розмірність, які будуть в відношеннях.

Ідентифікатори

#	ім'я	тип	сортвання	атрибути	тип	по умовчально	коментарий	доповнительно	действие
1	id	int(11)			Нет	Нет		AUTO_INCREMENT	✎ ✕ ▾
2	user_name	varchar(255)	utf8_general_ci		Нет	Нет			✎ ✕ ▾
3	user_phone	varchar(255)	utf8_general_ci		Нет	Нет			✎ ✕ ▾
4	user_comment	text	utf8_general_ci		Нет	Нет			✎ ✕ ▾
5	user_id	int(11)			Да	NULL			✎ ✕ ▾
6	date	timestamp			Нет	current_timestamp()			✎ ✕ ▾
7	products	text	utf8_general_ci		Нет	Нет			✎ ✕ ▾
8	status	int(11)			Нет	1			✎ ✕ ▾

Рисунок 2.4 — Таблица Users

1	id	int(11)			Нет	Нет		AUTO_INCREMENT	✎ ✕ ▾
2	name	varchar(255)	utf8_general_ci		Нет	Нет			✎ ✕ ▾
3	category_id	int(11)			Нет	Нет			✎ ✕ ▾
4	code	int(11)			Нет	Нет			✎ ✕ ▾
5	price	float			Нет	Нет			✎ ✕ ▾
6	availability	int(11)			Нет	Нет			✎ ✕ ▾
7	brand	varchar(255)	utf8_general_ci		Нет	Нет			✎ ✕ ▾
8	description	text	utf8_general_ci		Нет	Нет			✎ ✕ ▾
9	is_new	int(11)			Нет	0			✎ ✕ ▾
10	is_recommended	int(11)			Нет	0			✎ ✕ ▾
11	status	int(11)			Нет	1			✎ ✕ ▾

Рисунок 2.5 — Items

<input type="checkbox"/>	1	id	int(11)		Нет	Нет		AUTO_INCREMENT			
<input type="checkbox"/>	2	name	varchar(255)	utf8_general_ci	Нет	Нет					
<input type="checkbox"/>	3	sort_order	int(11)		Нет	0					
<input type="checkbox"/>	4	status	int(11)		Нет	1					

Рисунок 2.6— Таблиця *Category*

Відповідно до даних, які ми отримали створюється реляційна модель даних. Вона дає презентацію у формі ERD і вказує на відносини та зв'язки між ними. ERD на рівні програмного забезпечення розроблено з кодом, який по логіці точно пов'язує дані, необхідні для роботи, і надійно зберігає дані в таблицях. ERD показано на рисунку 2.10. Фізична реалізація є частиною дизайну бази даних. Фізичне проектування передбачає створення ERD за рахунок самого коду. Для фізичного дизайну я обрала базу даних MySQL, як я вже раніше писала, оскільки вона слугує більш правильним рішенням, бо є безоплатною та будеслугувати в якості сервера розробленої системи.

2.4 Вимоги до інформаційних систем.

Для коректного працювання сайту треба виконувати декілька пунктів, а саме:

- Як усі сучасні інтернет-додатки сайт потребує сервер або хостинг
- Сайт не може працювати без інтернету або на локальному ресурсі без підтримки PHP, MySQL.

У зв'язку з цим було обрано саме ці програмні та апаратні забезпечення для функціонування додатку:

- Операційна система: Windows Server (7-10), Linux;
- Процесор: Intel Xeon E5450 (3.0Ghz);
- HDD/SSD: 100-150 GB;
- ОЗУ: 1GB DDR3;
- Відео карта: -.

Додаток було протестовано на такій системі при використуванні OpenServer:

- Операційна система: Windows 10, Linux;
- Процесор: AMD Ryzen 5 3600 (3.6 BOOST 4.2 GHz);
- HDD/SSD: 1 TB;
- ОЗУ: 32GB;
- Відео карта: ASUS TUF GAMING RTX3060 TI (8GB)

2.5 UML-діаграми

На Рисунку 2.7 зображено логотип Уніфікованої мови моделювання.



Рисунок 2.7 — Логотип UML

UML (Unified Modeling Language) — це уніфікована мова моделювання, яка має призначення в парадигмах об'єктно-орієнтованого програмування. Це невід'ємна частина інтегрованого процесу написання програмного забезпечення. UML взагалі це мова загального призначення та відкритий стандарт, в якому застосовуються графічні нотації для моделювання абстрактних моделей систем, які як правило прийнято називати моделями UML. UML був розроблений головним чином для визначення, візуалізації, проектування та документування систем програмного забезпечення. Хоча UML не полягає мовою програмування, генерація коду можлива шляхом запуску моделей UML як інтерпретованого коду [15].

UML-діаграми можна розрізнити так:

- Діаграма прецедентів:

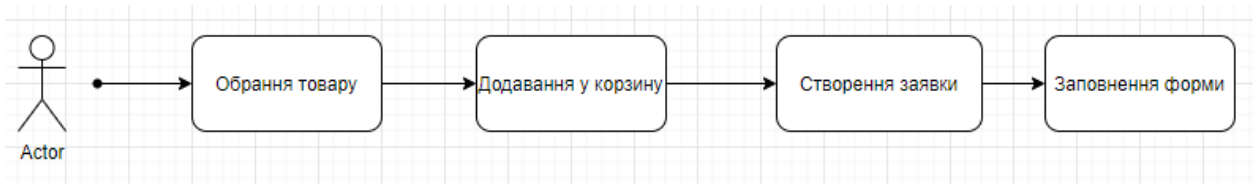


Рисунок 2.8 — Діаграма прецедентів web-додатку

На Рисунку 2.8 зображено як виконується взаємодія користувача та web-додатку. Користувач обирає товар який йому потрібен, а потім переходить у корзину та замовляє його. Потім адміністратор у панелі перевіряє це замовлення.

Створити UML-діаграму можливо за допомогою джерела [23].

2.6 Моделивання варіантів використання веб-додатку

Сьогодні в інтернет-магазинах є багато варіантів, але нам потрібна лише одна категорія. Це розпродаж.

Минулого року понад 35% українців робили покупки онлайн, і ця тенденція зростає. Відкрийте інтернет-магазин і продавайте товари, в яких ви добре розбираєтеся. Ви можете зробити це на своєму веб-сайті, продавати через Landingham або навіть через соціальні мережі. Або скористайтеся сторонніми сервісами, такими як Avito, Ebay або Amazon. Різні інтернет-магазини, категорії продажів: Електроніка б'є всі рекорди продажів в Інтернеті. Якщо ви можете конкурувати з великими мережами та знайти спосіб мінімізувати ризик пошкодження обладнання під час транспортування, це хороший варіант.

Одяг. Багато речей можна забрати, не приміряючи, і вони активно купуються в Інтернеті. Продаж через соціальні мережі дозволяє мінімізувати витрати та запропонувати доступні ціни.

Книги. Так, так, це папір. Звучить старомодно, але зараз люди віддають перевагу замовляти на сайтах, оскільки ціни в офлайн-магазинах зросли.

Позиції з Китаю. Абсолютно все підійде: замовляйте одягу, аксесуарів і гаджетів у китайських магазинах і перепродавайте онлайн у своєму регіоні — швидша доставка дає вам перевагу перед AliExpress. Важливо лише знайти надійного постачальника, щоб неякісна продукція не призвела до збитків.

Спонсоровані спільноти та акаунти. Звичайно, перш за все ви повинні їх заохотити. Вирішувати вам, чи хочете ви купити ботів і продати їх дешево, чи чесно набрати людей і продати їх дорого. Звертайтеся до нас для «білої» SMM реклами.

Органічна їжа. Ведіть переговори з місцевими фермерами, садівниками та збирачами грибів, знайдіть покупців в Інтернеті та організуйте доставку.

Антикваріат. Підходить, лише якщо ви це розумієте або хочете зрозуміти. Завжди існував сегмент населення любителів старовини. Важливо достукатися до них і запропонувати відповідні продукти.

Товари для тварин. Як не дивно, люди можуть відкладати на собі, але не на котах чи собаках.

Предмети для дітей. Ще одна високоприбуткова ринкова ніша.

Предмети першої необхідності. Той, хто може раз на місяць доставити людям миючий засіб, бритви, серветки, шампунь, знайде вдячних клієнтів.

І вам потрібно обов'язково вибрати правильну CMS для свого інтернет-магазину.

Партнерський маркетинг. Продавати товари інших компаній за певну плату. Такі операції на партнерських умовах вимагають мінімальних вкладень.

Кібернетична окупація. Зареєструйте та перепродайте свій домен.

Товари для хоббі. Люди, які захоплюються скрапбукінгом, вишиванням, різьбленням по дереву, риболовлею або полюванням точно знають, що їм потрібно — тому легко і швидко готові зробити онлайн-покупки.

Косметика. Її легко продати з невеликими інвестиціями, легко й дешево доставити, він має тривалий термін зберігання, що робить його практичним

продуктом для онлайн-торгівлі. Ви можете відкрити інтернет-магазин в Instagram.

Б/з товари. Купуємо та продаємо вживану техніку, одяг, побутові речі в хорошому стані.

Оскільки предметом моєї роботи є розробка веб-додатку для обробки заявок для особистих молочних ферм, то даний веб-додаток відноситься до категорії продажу.

Розроблений веб-додаток можна налаштувати під різні види продажу, а не тільки молочної продукції. Цю програму можна вважати багатоцільовою, оскільки її можна легко адаптувати для інших цілей зі знанням мови програмування.

Тому ви можете використовувати його для власних цілей.

2.7 Висновок до розділу 2

Побудовані діаграми діяльності програмного продукту, для наочного зображення основних процесів та дії, які виконуватиме система.

Побудовані діаграми потоків даних DFD розроблюваної системи, які відображають роботу системи при виконанні тих чи інших операцій.

Спроектвана ER діаграма бази даних додатку, яка дозволяє розробнику переглянути, які дані мають зберігатися в базі даних і, як вони повинні взаємодіяти між собою

Проаналізовано засоби розробки додатку.

РОЗДІЛ 3 РОЗРОБКА ВЕБ-ДОДАТКУ

3.1 Процес розробки веб-додатку

Для того щоб почати розробку веб-додатку потрібно завантажити середовище розробки таке як PHPStorm, яке зображено на Рисунку 3.1. Це середовище дозволяє вести роботу одразу з декількома файлами проекту.

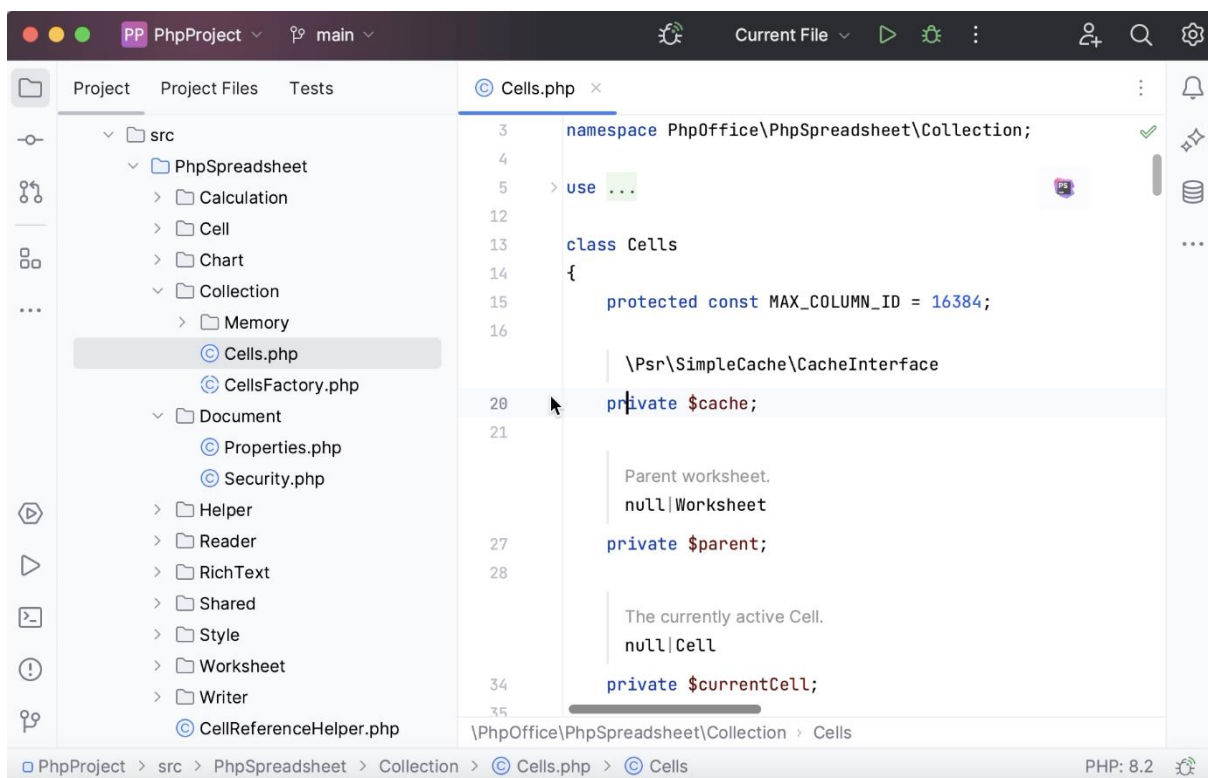


Рисунок 3.1 — Середовище розробки PHPStorm

Для успішного функціонування веб-додатку треба було розробити базу даних MySQL. На Рисунку 3.2 показано базу, яка зберігає дані про товари та про категорії, а також для зберігання користувачів, а саме адміністраторів.

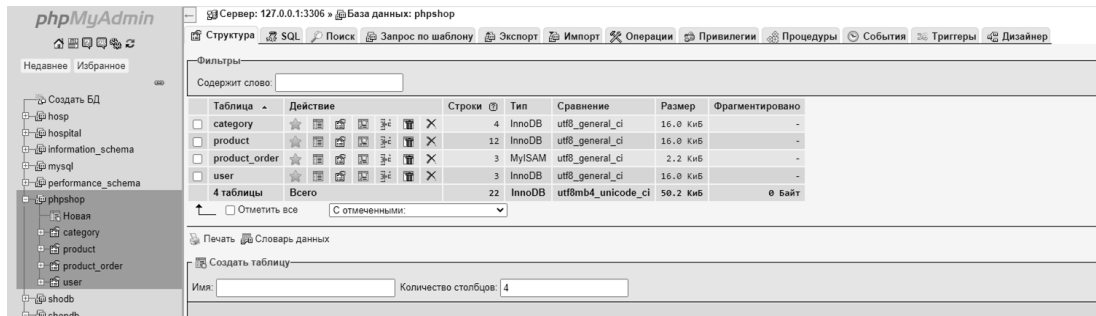


Рисунок 3.2 — Система керування базами даних PhpMyAdmin

Спочатку було розроблено таблицю, зображену на Рисунку 3.3, яка відповідала за зберігання користувачів.

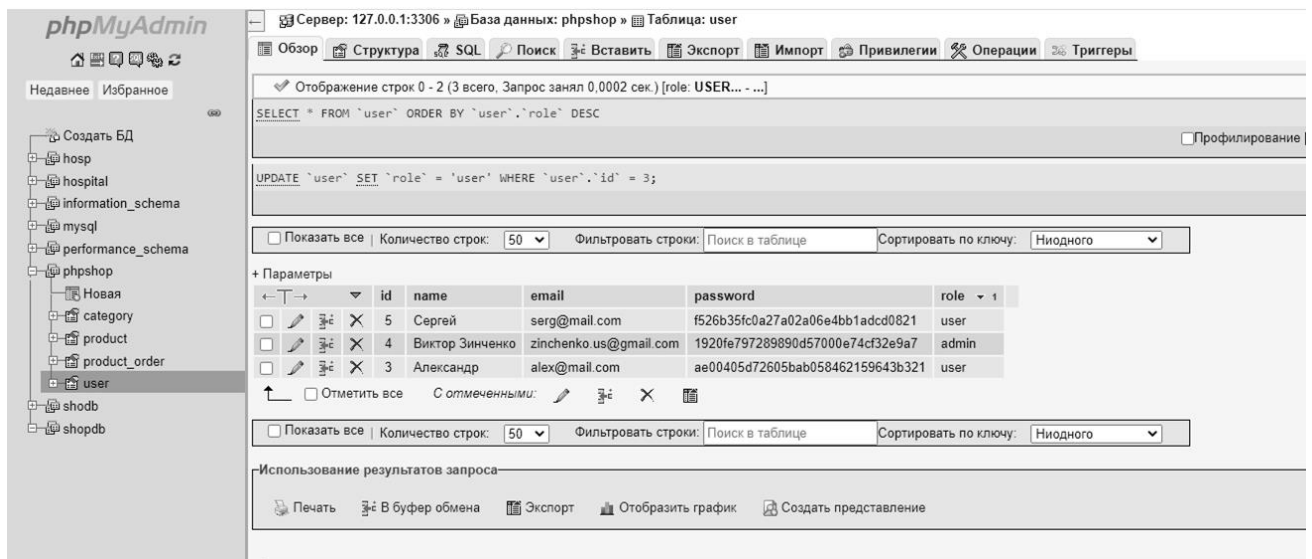


Рисунок 3.3 — Таблиця користувачів.

Після цього було реалізовано такі таблиці, як категорія товарів, яка зображена на Рисунку 3.4, таблиця товарів, яка зображена на Рисунку 3.5, а також таблиця замовлення товару на Рисунку 3.6.

Сервер: 127.0.0.1:3306 » База данных: phpshop » Таблица: category

Отображение строк 0 - 3 (4 всего, Запрос занял 0,0002 сек.)

```
SELECT * FROM `category`
```

Показать все | Количество строк: 50 | Фильтровать строки: Поиск в таблице | Сортировать по ключу: Ниодного

Параметры

id	name	sort_order	status
13	Ноутбуки	1	1
14	Планшеты	2	1
15	Мониторы	3	1
16	Игровые компьютеры	4	1

Использование результатов запроса

Рисунок 3.4 — Таблица категорії.

Сервер: 127.0.0.1:3306 » База данных: phpshop » Таблица: product

Отображение строк 0 - 11 (12 всего, Запрос занял 0,0001 сек.)

```
SELECT * FROM `product`
```

Показать все | Количество строк: 50 | Фильтровать строки: Поиск в таблице | Сортировать по ключу: Ниодного

Параметры

id	name	category_id	code	price	availability	brand	description	is_new	is_recommended	status
34	Ноутбук Asus X200MA (X200MA-KX315D)	13	1839707	395	1	Asus	Экран 11.6" (1366x768) HD LED, гляцевый / Intel P...	0	0	1
35	Ноутбук HP Stream 11-d050nr	13	2343847	305	0	Hewlett Packard	Экран 11.6" (1366x768) HD LED, матовый / Intel Cel...	1	1	1
36	Ноутбук Asus X200MA White	13	2028027	270	1	Asus	Экран 11.6" (1366x768) HD LED, гляцевый / Intel C...	0	1	1
37	Ноутбук Acer Aspire E3-112-C65X	13	2019487	325	1	Acer	Экран 11.6" (1366x768) HD LED, матовый / Intel Ce...	0	1	1
38	Ноутбук Acer TravelMate TMB115	13	1953212	275	1	Acer	Экран 11.6" (1366x768) HD LED, матовый / Intel Ce...	0	0	1
39	Ноутбук Lenovo Flex 10	13	1602042	370	0	Lenovo	Экран 10.1" (1366x768) HD LED, сенсорный, гляцева...	0	0	1
40	Ноутбук Asus X751MA	13	2028367	430	1	Asus	Экран 17.3" (1600x900) HD+ LED, гляцевый / Intel ...	0	1	1
41	Samsung Galaxy Tab S 10.5 16GB	14	1129365	780	1	Samsung	Samsung Galaxy Tab S создан для того, чтобы сделат...	1	1	1
42	Samsung Galaxy Tab S 8.4 16GB	14	1128670	640	1	Samsung	Экран 8.4" Super AMOLED (2560x1600) емкостный Mult...	0	0	1
43	Gazer Tegra Note 7	14	683364	210	1	Gazer	Экран 7" IPS (1280x800) емкостный Multi-Touch / NV...	0	0	1
44	Монитор 23" Dell E2314H Black	15	355025	175	1	Dell	С расширением Full HD Вы сможете рассмотреть мелы...	0	0	1
45	Компьютер Everest Game	16	1563832	1320	1	Everest	Everest Game 9085 — это компьютеры премиум класса...	0	0	1

Использование результатов запроса

Рисунок 3.5 — Таблица товарів.

Сервер: 127.0.0.1:3306 » База данных: phpshop » Таблица: product_order

Затронута 1 строка. (Запрос занял 0,0003 сек.)

Отображение строк 0 - 2 (3 всего, Запрос занял 0,0002 сек.)

```
SELECT * FROM `product_order`
```

Показать все | Количество строк: 50 | Фильтровать строки: Поиск в таблице | Сортировать по ключу: Ниодного

Параметры

id	user_name	user_phone	user_comment	user_id	date	products	status
45	fsdfs	1	123123123	4	2015-05-14 12:54:45	{ "1": "1", "2": "1", "3": "2" }	3
46	САША1	g3424242342		4	2015-05-18 18:34:42	{ "4": "3", "43": "3" }	1

Использование результатов запроса

Рисунок 3.6 — Таблица замовлення.

Програмна частина для «костяку» була реалізована на мові програмування PHP, JavaScript, HTML, CSS.

Реалізація класів для роботи веб-додатку.

Також було реалізовано корзину, для того щоб користувач зміг вибрати певну кількість необхідного йому товару. Для того щоб веб-додаток зміг використовувати базу даних треба було прописати з'єднання з цією базою.

3.2 Опис класів

1. Клас AdminBase

Клас AdminBase є абстрактним класом, який забезпечує загальну логіку для контролерів, що використовуються в панелі адміністратора. Цей клас, який представлено в Лістингу 3.1, включає метод для перевірки, чи є користувач адміністратором.

Лістинг 3.1 — *Перевірка користувача на наявність адмін. Доступу*

```
<?php

/**
 * Абстрактний клас AdminBase містить загальну логіку для контролерів, які
 * використовуються в панелі адміністратора
 */
abstract class AdminBase
{

    /**
     * Метод, який перевіряє користувача на те, чи він є адміністратором
     * @return boolean
     */
    public static function checkAdmin()
    {
        // Перевірити чи авторизований користувач. Якщо ні, він буде
        переадресований
        $userId = User::checkLogged();
```

```

// Отримуємо інформацію про поточного користувача
$user = User::getUserById($userId);

// Якщо роль поточного користувача "admin", пускаємо його до адмінпанелі
if ($user['role'] == 'admin') {
    return true;
}

// Інакше завершуємо роботу з повідомленням про закритий доступ
die('Access denied');
}
}

```

Метод `checkAdmin`

Метод `checkAdmin` є статичним і призначений для перевірки, чи має поточний користувач права адміністратора. Він виконує такі дії:

1. Перевіряє, чи авторизований користувач за допомогою методу `User::checkLogged()`. Якщо користувач не авторизований, його буде перенаправлено.
2. Отримує інформацію про поточного користувача за допомогою методу `User::getUserById($userId)`.
3. Якщо роль поточного користувача – "admin", метод повертає `true`.
4. Якщо роль користувача відрізняється від "admin", метод завершує роботу з повідомленням про закритий доступ (`die('Access denied')`).

Цей метод забезпечує безпеку, гарантує, що тільки авторизовані адміністратори можуть отримати доступ до відповідних розділів панелі адміністратора.

2. Клас `Cart`

Клас `Cart`, який предсталено в Лістингу 3.2, призначений для керування товарами у кошику покупок, використовуючи сесії PHP. Він забезпечує додавання товарів до кошика та підрахунок кількості товарів у кошику.

Лістинг 3.2 — Реалізація класу корзина

```

<?php

/**
 * Клас Cart
 * Компонент для роботи кошиком
 */
class Cart
{

    /**
     * Додавання товару до кошика (сесії)
     * @param int $id <p>id товар</p>
     * @return integer <p>Кількість товарів у кошику</p>
     */
    public static function addProduct($id)
    {
        // Приводим $id к типу integer
        $id = intval($id);

        // Порожній масив для товарів у кошику
        $productsInCart = array();

        // Якщо у кошику вже є товари (вони зберігаються у сесії)
        if (isset($_SESSION['products'])) {
            // То заповним наш масив товарами
            $productsInCart = $_SESSION['products'];
        }

        //Перевіряємо, чи є вже такий товар у кошику
        if (array_key_exists($id, $productsInCart)) {
            // Якщо такий товар є у кошику, але був доданий ще раз, збільшимо
            кількість на 1
            $productsInCart[$id] ++;
        } else {
            // Якщо ні, додаємо id нового товару до кошика з кількістю 1
            $productsInCart[$id] = 1;
        }

        // Записуємо масив із товарами у сесію
        $_SESSION['products'] = $productsInCart;

        // Повертаємо кількість товарів у кошику
        return self::countItems();
    }

    /**
     * Підрахунок кількості товарів у кошику (у сесії)
     * @return int <p>Кількість товарів у кошику</p>
     */
}

```

```

public static function countItems()
{
    // Перевірка наявності товарів у кошику
    if (isset($_SESSION['products'])) {
        // Якщо масив із товарами є
        // Підраховуємо та повернемо їх кількість
        $count = 0;
        foreach ($_SESSION['products'] as $id => $quantity) {
            $count = $count + $quantity;
        }
        return $count;
    } else {
        // Якщо товарів немає, повернемо 0
        return 0;
    }
}

/**
 * Повертає масив з ідентифікаторами та кількістю товарів у кошику<br/>
 * Якщо товарів немає, повертає false;
 * @return mixed: boolean or array
 */
public static function getProducts()
{
    if (isset($_SESSION['products'])) {
        return $_SESSION['products'];
    }
    return false;
}

/**
 * Отримуємо загальну вартість переданих товарів
 * @param array $products <p>Масив з інформацією про товари</p>
 * @return integer <p>Загальна вартість</p>
 */
public static function getTotalPrice($products)
{
    // Отримуємо масив з ідентифікаторами та кількістю товарів у кошику
    $productsInCart = self::getProducts();

    // Підраховуємо загальну вартість
    $total = 0;
    if ($productsInCart) {
        // Якщо у кошику не порожньо
        // Проходимо по переданому метод масиву товарів
        foreach ($products as $item) {
            // Знаходимо загальну вартість: ціна товару * кількість товару
            $total += $item['price'] * $productsInCart[$item['id']];
        }
    }
}

```

```

        return $total;
    }

    /**
     * Очищає кошик
     */
    public static function clear()
    {
        if (isset($_SESSION['products'])) {
            unset($_SESSION['products']);
        }
    }

    /**
     * Видаляє товар із зазначеним id із кошика
     * @param integer $id <p>id товар</p>
     */
    public static function deleteProduct($id)
    {
        // Отримуємо масив з ідентифікаторами та кількістю товарів у кошику
        $productsInCart = self::getProducts();

        // Видаляємо з масиву елемент із зазначеним id
        unset($productsInCart[$id]);

        // Записуємо масив товарів з віддаленим елементом у сесію
        $_SESSION['products'] = $productsInCart;
    }
}

```

Метод addProduct

Метод addProduct(\$id) є статичним методом, який виконує додавання товару до кошика. Ось ключові кроки цього методу:

1. Перетворення переданого ID товару у ціле число.
2. Ініціалізація масиву для зберігання товарів у кошику.
3. Перевірка, чи є вже товари у кошику (зберігаються у сесії).
4. Якщо товар вже є у кошику, збільшує кількість цього товару на 1.
5. Якщо товару ще немає у кошику, додає його з кількістю 1.
6. Зберігає оновлений масив товарів у сесії.
7. Повертає загальну кількість товарів у кошику за допомогою методу countItems().

Метод countItems

Метод countItems() є статичним методом, який підраховує загальну кількість товарів у кошику. Ось основні етапи цього методу:

1. Перевірка, чи існує масив товарів у сесії.
2. Якщо масив товарів існує, підраховує загальну кількість товарів.
3. Повертає підраховану кількість товарів.

Цей клас забезпечує базову функціональність для роботи з кошиком покупок у веб-додатках, дозволяючи додавати товари до кошика та відслідковувати їх кількість.

3. Клас Router

Клас Router, який представлено в Лістингу 3.3, використовується для керування маршрутизацією в веб-додатку. Він перенаправляє HTTP-запити на відповідний контролер і дію залежно від URI запити.

Лістинг 3.3 — Реалізація класу БД

```
<?php

/**
 * Клас Db
 * Компонент для роботи з базою даних
 */
class Db
{
    /**
     * Встановлює з'єднання з базою даних
     * @return \PDO <p>Об'єкт класу PDO для роботи з БД</p>
     */
    public static function getConnection()
    {
        // Отримуємо параметри підключення з файлу
        $paramsPath = ROOT . '/config/db_params.php';
        $params = include($paramsPath);

        // Встановлюємо з'єднання
        $dsn = "mysql:host={$params['host']};dbname={$params['dbname']}";
        $db = new PDO($dsn, $params['user'], $params['password']);

        // Задаємо кодування
```

```

        $db->exec("set names utf8");

        return $db;
    }
}

```

Властивості:

- `private $routes`: Масив маршрутів, завантажений із конфігураційного файлу.

Методи:

- **Конструктор (`__construct`)**: Ініціалізує об'єкт Router, встановлює шлях до файлу маршрутів і завантажує маршрути з файлу.
- **getURI**: Повертає URI запиту, видаляючи початкові та кінцеві слеші.
- **run**: Обробляє вхідний HTTP-запит, перевіряє наявність маршруту, що відповідає URI запиту, і перетворює зовнішній шлях у внутрішній згідно з правилом.

Цей клас є важливою частиною MVC-фреймворку, де він відповідає за маршрутизацію запитів до відповідних контролерів і дій

4. Клас CartController

Клас CartController використовується для оформлення покупки, підрахунку вартості товарів у кошику, та саме для додавання й видалення товару з нашого кошика в веб-додатку.

Розглянемо метод **actionCheckout**, представлений в Лістингу 3.4. Він забезпечує формування покупки. Після того як користувач натиснув на кнопку оформлення замовлення, викликається цей метод. Він перевіряє чи є замовник гостем. Якщо клієнт вже здійснив раніше авторизацію, то ми автоматично підтягуємо ім'я користувача з бази даних. Після успішного вводу користувачем телефону та пошти інформація про замовлення відправляється на пошту до адміністратора.

Лістинг 3.4 — Реалізація оформлення замовлення

```

/**
 * Action для сторінки "Оформлення покупки"
 */
public function actionCheckout()
{
    // Отримаємо дані з кошика
    $productsInCart = Cart::getProducts();

    // Якщо товарів немає, відправляємо користувачі шукати товари на головну
    if ($productsInCart == false) {
        header("Location: /");
    }

    // Список категорій для лівого меню
    $categories = Category::getCategoriesList();

    // Знаходимо загальну вартість
    $productsIds = array_keys($productsInCart);
    $products = Product::getProductsByIds($productsIds);
    $totalPrice = Cart::getTotalPrice($products);

    // ККількість товарів
    $totalQuantity = Cart::countItems();

    //Поля для форми
    $userName = false;
    $userPhone = false;
    $userComment = false;

    // Статус успішного оформлення замовлення
    $result = false;

    // ПЧи перевіряємо користувач гостем
    if (!User::isGuest()) {
        // Якщо користувач не гість
        // Отримуємо інформацію про користувача з БД
        $userId = User::checkLogged();
        $user = User::getUserById($userId);
        $userName = $user['name'];
    } else {
        // Якщо гість, поля форми залишаться порожніми
        $userId = false;
    }

    // Обробка форми
    if (isset($_POST['submit'])) {
        // Якщо форму відправлено
        // Отримуємо дані з форми
        $userName = $_POST['userName'];
    }
}

```

```

$userPhone = $_POST['userPhone'];
$userComment = $_POST['userComment'];

// Прапор помилок
$errors = false;

// Валідація полів
if (!User::checkName($userName)) {
    $errors[] = 'Неправильное имя';
}
if (!User::checkPhone($userPhone)) {
    $errors[] = 'Неправильный телефон';
}

if ($errors == false) {
    // Якщо помилок немає
    // Зберігаємо замовлення у базі даних
    $result = Order::save($userName, $userPhone, $userComment,
        $userId, $productsInCart);

    if ($result) {
        // Якщо замовлення успішно збережено
        // Сповіщаємо адміністратора про нове
замовлення    ПОШТОЮ
        $adminEmail = 'support@mycompshop.ua';
        $message = '<a
href="https://ycompshop.ua/admin/orders">Список заказов</a>';
        $subject = 'Новый заказ!';
        mail($adminEmail, $subject, $message);

        // Очищаємо кошик
        Cart::clear();
    }
}

// Підключаємо вигляд
require_once(ROOT . '/views/cart/checkout.php');
return true;
}

```

5. Клас SiteController

Клас SiteController виконує роль контролера в MVC. Цей клас містить методи, які відповідають за обробку різних дій (actions) на сайті. Кожен метод контролера відповідає за певну сторінку або функціональність сайту. Розглянемо детальніше призначення кожного методу в цьому класі:

Метод `actionIndex()`:

Призначення: Метод відповідає за головну сторінку сайту.

Функціональність:

- Отримує список категорій для відображення в лівому меню.
- Отримує список останніх товарів (обмеження на 6 товарів).
- Отримує список рекомендованих товарів для слайдера.
- Підключає відповідний вид (`view`), що відображає головну сторінку сайту.

Метод `actionContact()`, представлений в Лістингу 3.5, забезпечує:

Призначення: Метод відповідає за сторінку "Контакти".

Функціональність:

- Ініціалізує змінні для форми контактів.
- Обробляє дані, які користувач вводить у форму (`email` та текст повідомлення).
- Валідує `email` за допомогою методу `User::checkEmail()`.
- У разі відсутності помилок відправляє повідомлення адміністратору через `email`.
- Підключає відповідний вид (`view`), що відображає сторінку контактів.

Лістинг 3.5 — Реалізація форми зворотнього зв'язку

```
/**
 * Action для сторінки "Контакти"
 */
public function actionContact()
{

    // Змінні для форми
    $userEmail = false;
    $userText = false;
    $result = false;

    // Обробка форми
    if (isset($_POST['submit'])) {
        // Якщо форма отправлена
        // Отримуємо данні з форми
        $userEmail = $_POST['userEmail'];
```



```

$userText = $_POST['userText'];

// Флаг помилок
$errors = false;

// Валідація полів
if (!User::checkEmail($userEmail)) {
    $errors[] = 'Неправильный email';
}

if ($errors == false) {
    // Якщо помилок немає
    // Відправляємо лист адміністратору
    $adminEmail = 'support@mycompshop.ua';
    $message = "Текст: {$userText}. От {$userEmail}";
    $subject = 'Тема письма';
    $result = mail($adminEmail, $subject, $message);
    $result = true;
}
}

// Підключаємо вид
require_once(ROOT . '/views/site/contact.php');
return true;
}

```

3.3 Тестування веб-додатку

Щоб протестувати функціонал веб-додатку було використано:

- Додаток та браузер Chrome;
- Особистий комп'ютер в якості серверу для веб-додатку;
- OpenServer в якості програми запуску;
- Веб-додаток.

Тестування проходило таким чином. Користувач заходив на сайт обирав необхідний йому товар додавав до кошику та робив замовлення.

Для того щоб користувач міг вільно працювати з веб-додатком було вирішено заповнити його випадковим контентом.

Під час тестування було виявлено декілька помилок, які будуть вирішені пізніше.

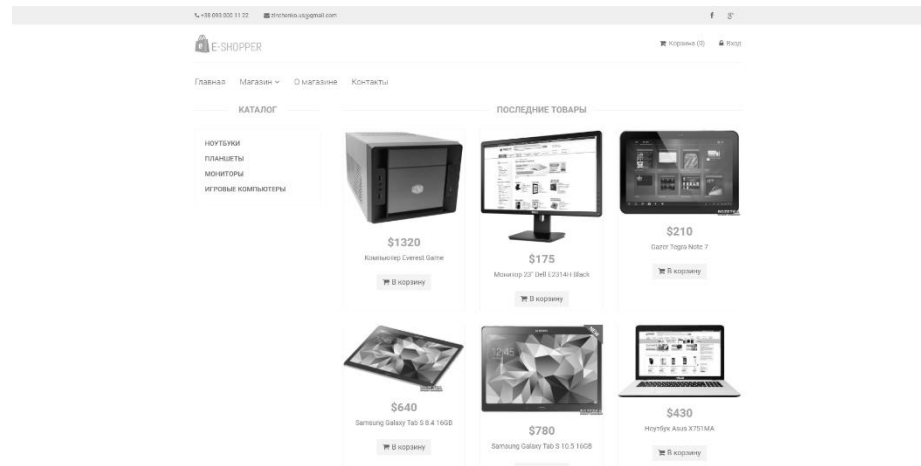


Рисунок 3.7 — Перевірка працездатності веб-додатку.

3.4 Опис функціонування веб-додатку

Функціонування веб-додатку залежить від працездатності серверу та його коректного налаштування. Якщо сервер не буде працювати то веб-додаток буде працювати не правильно або зовсім не буде працювати.

Для перевірки сайту на працездатність було прийнято рішення використати безкоштовний сервіс uptime. Який дозволяє провести корисну навантаження на сайт задля перевірки його працездатності.

Але так як сайт у розробці робота над ним буде продовжуватися і далі. Тому зовнішній вигляд або функціонал його може змінюватися.

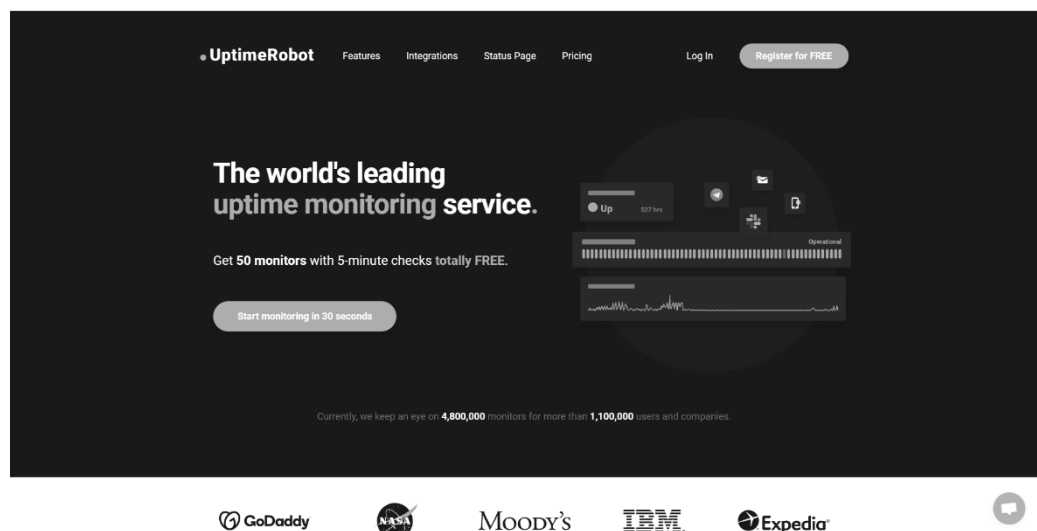


Рисунок 3.8 — Веб-додаток UPTIME.

2.2 Вимоги до інтерфейсу

Відповідно до функціональних вимог була розроблена клієнтська програма, де реалізовані всі зазначені функції.

На рисунку 3.9 показана авторизація контролює доступ користувачів до ресурсів системи після успішного проходження ними аутентифікації

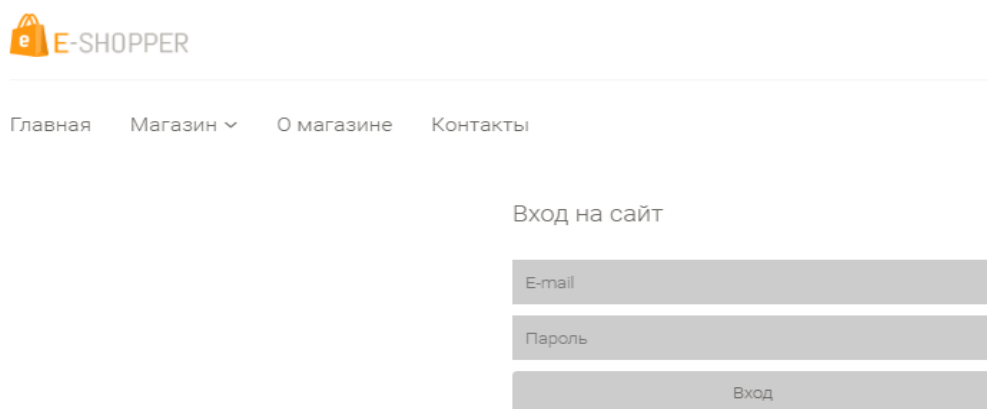


Рисунок 3.9 — Вікно аутентифікації

Після проходження авторизації користувачу системи відкривається головна сторінка застосунку (див. Рисунок 3.10).

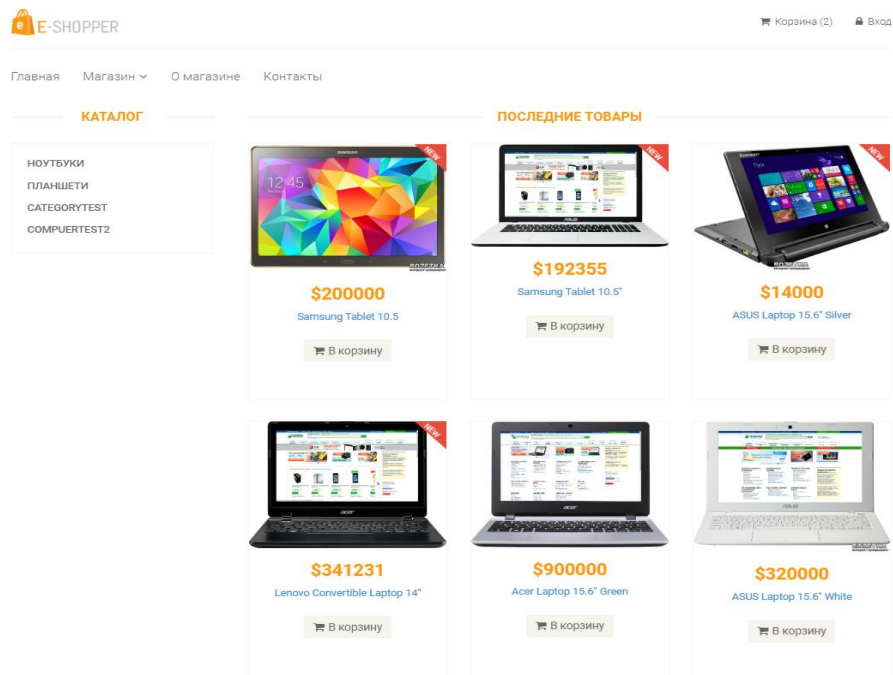


Рисунок 3.10 — Головна сторінка

Після вибору товарів, які користувач бажає придбати, їх можна додати до кошика. Як показано на Рисунку 3.11 обрані товари можна переглянути, натиснувши на пункт меню «Переглянути кошик».

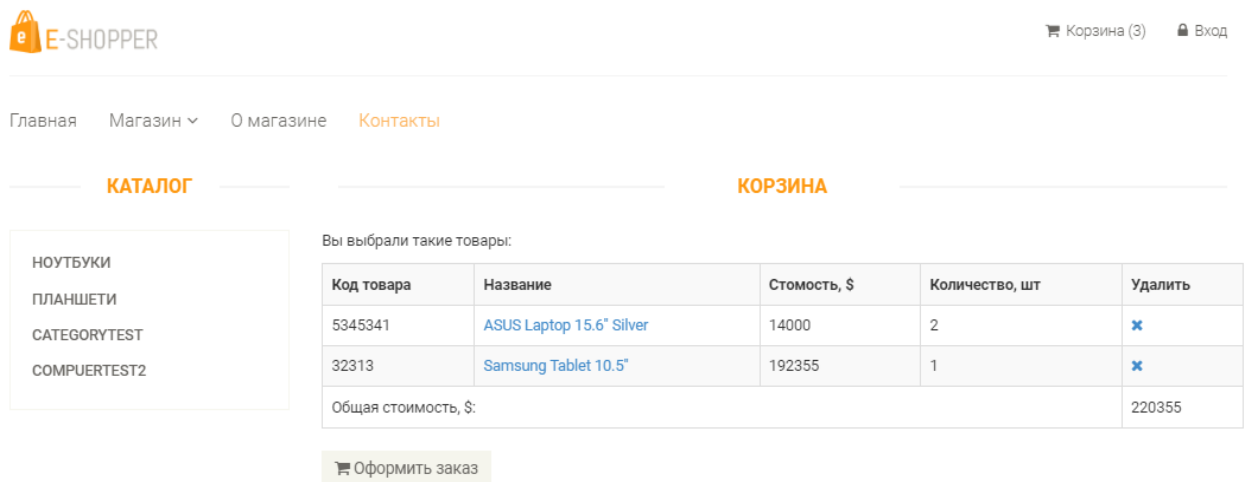


Рисунок 3.11 — Перегляд обраних товарів у кошику

Якщо у покупця виникають якісь питання, у застосунку, як показано на Рисунку 3.12 є зворотній зв'язок. Можна надіслати повідомлення, вказавши електронну адресу. Яке прийде на пошту адміністратору

Обратная связь

Есть вопрос? Напишите нам

Ваша почта

E-mail

Сообщение

Сообщение

Отправить

Рисунок 3.12 — Форма зворотнього зв'язку

Якщо до системи заходить адміністратор, він потрапляє до адмін панелі. Де є можливість Керувати товарами, категоріями та замовленнями. На Рисунок 3.13 зображена форма редагування товарами, де адміністратор може додати новий товар або редагувати вже існуючий.

[+ Добавить товар](#)

Список товаров


















ID товара	Артикул	Название товара	Цена		
33	1231234	Acer Laptop 15.6	250000		
34	12311234	ASUS Laptop 14	300000		
35	13132121	HP Laptop 15.6" Blue	500000		
36	13132124	ASUS Laptop 15.6" White	320000		
37	747474	Acer Laptop 15.6" Green	900000		
38	56463452	Lenovo Convertible Laptop 14"	341231		
39	5345341	ASUS Laptop 15.6" Silver	14000		
40	32313	Samsung Tablet 10.5"	192355		
41	312351	Samsung Tablet 10.5	200000		

Рисунок 3.13 — Форма керування товарами

На Рисунку 3.14 зображена форма управління замовлень , де адміністратор може редагувати статус замовлення, відносно його готовності.

Редактировать заказ #48

Имя клиента
Diana

Телефон клиента
+380966915186

Комментарий клиента

Дата оформления заказа
2024-06-11 12:55:39

Статус

В обработке	▼
Новый заказ	
В обработке	
Доставляется	
Закрыт	

Рисунок 3.14 — Форма управління замовленнями

ВИСНОВКИ

В ході виконання кваліфікаційної роботи було досягнуто поставлених цілей та задач, а саме було проведено дослідження літератури, направленої на розробку web-додатку з продажу комп'ютерних товарів, та було проведено огляд існуючих аналогів на ринку.

Отримано наступні наступні результати:

- Створено web-додаток з функціоналом продажу комп'ютерних товарів.
- Виконано огляд мов програмування та середовищ розробки для створення web-додатку.
- Проаналізовано функціональні можливості web-додатку.
- Протестовано web-додаток на помилки та працездатність.
- Зроблено висновки та рекомендації щодо застосування програмного застосунку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Martin Fowler. Refactoring: Improving the Design of Existing Code. 2nd Edition. Addison-Wesley Professional, 2018. 464 p.
2. Alessandro Del Sole. Visual Studio Code Succinctly. SyncFusion Inc., 2016. 128p.
3. Grady Booch, James Rumbaugh, Ivar Jacobson. The Unified Modeling Language User Guide. 2nd Edition. Addison-Wesley Professional, 2005. 496 p.
4. Документація до UML-діаграм: веб сайт. URL: <https://creately.com/blog/diagrams/uml-diagram-types-examples/>. (дата звернення: 08.04.2024).
5. Документація до недоліків та переваг JavaScript: веб сайт. URL: <https://hackit-ukraine.com/627-the-advantages-and-disadvantages-of-javascript>. (дата звернення: 18.04.2024).
6. Документація до створення UML-діаграм: веб сайт. URL: <https://slickplan.com/blog/how-to-make-a-uml-diagram>. (дата звернення: 22.04.2024).
7. Joseph W. Lowery. Macromedia Dreamweaver MX 2004 Bible. Wiley, 2004. 1000 p.
8. Нестеренко Д. В., Скрипник І. А., доцент, к.ф.-м.н. Розробка Інтернет-магазину для продажу комп'ютерних товарів з використанням Java Script та PHP. Збірник наукових праць студентів, аспірантів, докторантів і молодих вчених «Молода наука-2024» / Запорізький національний університет. Запоріжжя: ЗНУ, 2024. Т.5. С. 191-192.

Декларація
академічної доброчесності
здобувача ступеня вищої освіти ЗНУ

Я, Нестеренко Діана Владиславівна студентка 4 курсу, форми навчання денної, Інженерного навчально-наукового інституту, спеціальність 121 Інженерія програмного забезпечення, адреса електронної пошти ipz20bd-106@stu.zsea.edu.ua, — підтверджую, що написана мною кваліфікаційна робота на тему **«Розробка Інтернет-магазину для продажу комп'ютерних товарів з використанням Java Script та PHP»** відповідає вимогам академічної доброчесності та не містить порушень, що визначені у ст.42 Закону України «Про освіту», зі змістом яких ознайомлений.

- заявляю, що надана мною для перевірки електронна версія роботи є ідентичною її друкованій версії;

згодна на перевірку моєї роботи на відповідність критеріям академічної доброчесності у будь-який спосіб, у тому числі за допомогою інтернет-системи, а також на архівування моєї роботи в базі даних цієї системи.

Дата 17.06.2024 Підпис _____ Нестеренко Діана Владиславівна
(студентка)

Дата 17.06.2024 Підпис _____ Скрипник Ірина Анатоліївна
(науковий керівник)