

Національний університет «Полтавська політехніка імені Юрія Кондратюка»
(повне найменування вищого навчального закладу)

Навчально-науковий інститут інформаційних технологій та робототехніки
(повна назва факультету)

Кафедра автоматики, електроніки та телекомунікацій
(повна назва кафедри)

Пояснювальна записка

до кваліфікаційної роботи

бакалавр
(освітній рівень)

на тему Розробка web-застосунку для сайту торгової мережі

Виконав: студент 4 курсу, групи 401-ТТ
спеціальності 172 «Телекомунікації та радіотехніка»
(шифр і назва спеціальності)

Міщенко А.С.

(прізвище та ініціали)

Керівник:

Сокол Г.В.

(прізвище та ініціали)

Рецензент

_____ (прізвище та ініціали)

РЕФЕРАТ

Загальний обсяг текстової частини дипломної роботи складає 71 сторінка формату А4. Вона складається з переліку умовних скорочень, вступу, 3-ох розділів, висновків, списку використаних джерел, 4 додатків. Робота містить 28 рисунків, використано 19 джерел.

Актуальність створення web-застосунку. Потреба у покупках існує завжди, але не завжди є можливість ходити по магазинах, незалежно від сезону та розвитку економіки. Щорічно кількість web-застосунків збільшується, оскільки це дійсно приносить прибуток та зручно для користувачів, не говорячи вже про економію бюджету та часу. Також клієнтами вашого магазину можуть бути не тільки мешканці вашого міста, а й люди усієї планети.

Об'єкт дослідження: Процес функціонування web-застосунку сайту торгової мережі.

Предмет дослідження: Технології розробки web-застосунку сайту торгової мережі.

Метою дипломної роботи є розробка web-застосунку для сайту торгової мережі за допомогою якого користувач може придбати товари та самостійно обрати потрібну послугу не виходячи з дому.

Ключові слова: web-застосунок, база даних, панель адміністрування, web-сервер, код, web-сторінка, дизайн, товар, користувач.

ABSTRACT

The total volume of the text part of the thesis is 71 A4 pages. It consists of a list of abbreviations, introduction, 3 sections, conclusions, a list of sources used, 4 appendices. The work contains 28 drawings, 19 sources were used.

The relevance of creating a web application. The need for shopping always exists, but it is not always possible to go shopping, regardless of the season and economic development. The number of web applications increases every year, because it is really profitable and user-friendly, not to mention budget and time savings. Also, customers of your store can be residents of your city and people around the world.

An object of research: The process of functioning of the web-application of the site of the trade network..

The subject of research: Technologies for developing a web-application of a retail network site.

The thesis project aims to create a web application through which the user can purchase and purchase goods and choose the desired service without leaving home.

Keywords: web application, database, administration panel, web-server, code, web-page, design, product, the user.

Національний університет «Полтавська політехніка імені Юрія Кондратюка»
Інститут Навчально-науковий інститут інформаційних технологій та
робототехніки
Кафедра Автоматики, електроніки та телекомунікацій
Освітній рівень Бакалавр
Спеціальність 172 «Телекомунікації та радіотехніка»

ЗАТВЕРДЖУЮ

**Завідувач кафедри автоматки,
електроніки та телекомунікацій**

_____ О.В. Шефер
“ 11 ” травня 2021 р.

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

Міщенко Артему Сергійовичу

1. Тема роботи «Розробка web-застосунку для сайту торгової мережі»
керівник роботи Сокол Галина Вікторівна, к.т.н., доцент
затверджена наказом вищого навчального закладу від 03.03.2021 року № 158 -
фа
2. Строк подання студентом проекту (роботи) 15.06.2021 р.
3. Вихідні дані до проекту (роботи) стандарт PHP, JavaScript, MySQL, HTML,
CSS.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно
розробити) Аналіз сучасних web-застосунків. Обґрунтування вибору
інструментів для розробки web-застосунків. Розробка web-застосунку торгової
мережі.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових плакатів):
 - 1) узагальнена структура web-застосунку;
 - 2) головна сторінка веб-застосунку «Comfy»;
 - 3) головна сторінка веб-застосунку «МОУО»;
 - 4) головна сторінка веб-застосунку «ITbox»;
 - 5) головна сторінка веб-застосунку «Комп'ютерний всесвіт»;
 - 6) популярні фреймворки та мови програмування;
6. Дата видачі завдання 11.05.2021 р.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	6
ВСТУП.....	7
1. АНАЛІЗ СУЧАСНИХ WEB-ЗАСТОСУНКІВ	9
1.1 Поняття web-застосунку	9
1.2 Класифікація web-застосунків	10
1.3 Технологічні тренди web-розробки.....	11
1.4 Аналітичний огляд аналогів web-застосунку.....	13
1.5 Огляд сучасних технологій для створення web-застосунків	16
1.6 Висновки до першого розділу	18
2. ОБГРУНТУВАННЯ ВИБОРУ ІНСТРУМЕНТІВ ДЛЯ РОЗРОБКИ ЗАСТОСУНКУ	20
2.1 HTML – мова гіпертекстової розмітки	21
2.2 CSS – каскадні таблиці стилів.....	22
2.3 PHP – гіпертекстовий препроцесор.....	23
2.4 JavaScript – прототипна мова програмування	24
2.5 Open Server – локальний web-сервер	25
2.6 MySQL – вільна система управління реляційними базами даних	26
2.7 Висновки до другого розділу	27
3. РОЗРОБКА WEB-ЗАСТОСУНКУ ТОРГОВОЇ МЕРЕЖІ	28
3.1 Дизайн та структура web-застосунку торгової мережі	28
3.2 Розробка архітектури web-застосунку торгової мережі	36
3.3 Розробка діаграми функціональності web-застосунку тогрової мережі	41
3.4 Розробка інтерфейсу web-застосунку торгової мережі.....	43
3.5 Тестування створеного web-застосунку торгової мережі	45
Висновки до розділу	48
ВИСНОВОК.....	49
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	51

	5
ДОДАТОК А.....	53
ДОДАТОК Б.....	63
ДОДАТОК В.....	69
ДОДАТОК Г.....	70

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ПК - персональний комп'ютер.

COM – Component Object Model.

JS – JavaScript.

ОС – операційна система.

SEO – search engine optimization.

СУБД – система управління базами даних.

БД – база даних.

ПЗ – програмне забезпечення.

HTML – HyperText Markup Language.

SPA – Single Page Application.

PWA - Progressive Web Application.

FaaS – Functions as a Service.

ІІІ – штучний інтелект.

W3C – World Wide Web Consortium.

png – portable network graphics.

ВСТУП

Останнім часом створення web-застосунків є досить розповсюдженим заняттям і над зручністю та простотою створення web-застосунків працює досить велика кількість розробників.

В сучасному світі кожна стабільна компанія повинна мати власний web-застосунок в мережі інтернет, який забезпечить інформаційну підтримку бізнесу. За допомогою web-застосунків компанії розв'язують такі задачі, як доставлення своїх товарів, розширення потенціальної аудиторії споживачів, підтримка бренду, інформування громадськості та ін. Компанія СBR провела дослідження поведінки онлайн-покупців під час пандемії коронавірусу. За даними компанії 70% населення країни регулярно користуються інтернетом, хоча в минулому році ця цифра дорівнювала 60%. Тобто в період пандемії рівень користування інтернетом значно виріс. Майже 33% населення країни регулярно здійснює покупки в інтернеті, ця аудиторія поступово зростає. Розробка web-застосунків для компанії являється актуальною і затребуваною сферою діяльності, оскільки становить собою досить недорогий спосіб підвищення продажів, та збільшення аудиторії покупців[1].

Актуальність створення web-застосунку. Потреба у покупках існує завжди, але не завжди є можливість ходити по магазинах, незалежно від сезону та розвитку економіки. Щорічно кількість web-застосунків збільшується, оскільки це дійсно приносить прибуток та зручно для користувачів, не говорячи вже про економію бюджету та часу. Також клієнтами вашого магазину можуть бути не тільки мешканці вашого міста, а й люди усієї планети.

Об'єкт дослідження: Процес функціонування web-застосунку сайту торгової мережі.

Предмет дослідження: Технології розробки web-застосунку сайту торгової мережі.

Метою дипломної роботи: є розробка web-застосунку для сайту торгової мережі за допомогою якого користувач може придбати товари та самостійно обрати потрібну послугу не виходячи з дому.

Постановка задачі. Задачею дослідження є проектування та програмна реалізація web-застосунку, для того, щоб продати та доставити товари.

Призначення web застосунку Функціонал web застосунку, який надається користувачу – це можливість переглядати каталог товарів, а також замовлення цих товарів з можливим доставленням, створення кабінету користувача, для спрощення процесу наступних замовлень. Функціональна частина web-застосунку для адміністрування дає можливість додавання нових послуг, або товарів, з урахуванням відповідних характеристик та редагування наявної на сайті інформації.

1. АНАЛІЗ СУЧАСНИХ WEB-ЗАСТОСУНКІВ

1.1 Поняття web-застосунку

Web-застосунок, або також web-система – це різні програмні продукти, які виконують певні функції, з архітектурою «клієнт-сервер». «Клієнт» використовується для позначення програми, яку застосовують для запуску web-застосунку. Сервером називають додаток що зберігає інформацію. Зазвичай «клієнтом» слугує браузер, сервером web-сервер [2].

Структура web-застосунку в загальному вигляді та напрямки передачі інформації представлена на рис 1.1.

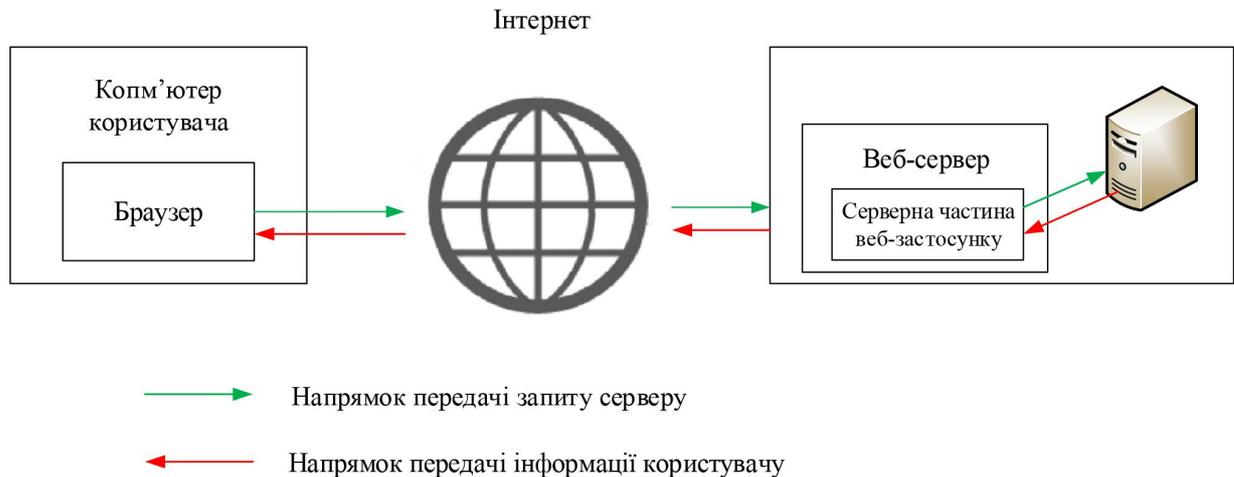


Рисунок 1.1 – узагальнена структура web-застосунку

Багато хто вважає що web-застосунок це звичайний сайт, але це не так. Для мене сайт це в першу чергу щось інформаційне, статичне: візитівка компанії, сайт рецептів або міський портал. Набір завчасно підготовлених HTML-файлів, які лежать на віддаленому web-сервері та виводяться до браузеру по запиту.

Сайти містять різноманітну статистику, котра як і HTML-файл не створюється на ходу. Частіш за все це картинки, CSS-файли, JS-скрипти, але також можуть бути і інші будь-які файли.

А web-застосунки - це щось технічно більш складне. Тут HTML-сторінки генеруються по запиту користувача. Поштові клієнти, соціальні мережі, інтернет магазини, це все web-застосунки.

Статична web-сторінка відображається для всіх користувачів однаково. Як це працює:

1. Користувач вводить в адресному рядку запит, або адресу сторінки.
2. Браузер надсилає його на web-сервер.
3. Сервер аналізує запит, і визначає що ніяких особливих відміток немає.
4. Надсилає web сторінку браузеру не змінюючи ніяких даних на ній.

У випадку з динамічними сторінками схема виглядає наступним чином:

1. Браузер надіслав запит на web-сервер. Наприклад, при цьому надійшла інформація, що у цього користувача є набір ознак, при наявності яких, для нього треба вивести певну інформацію, значить сторінка буде динамічною.
2. Web-сервер надсилає її на сервер застосунків, де спеціальне ПЗ застосує правила та інструкції для додавання особливих змінних. Наприклад користувач авторизований у системі. Для нього виведеться інформація з ПБ та іншою інформацію саме для нього.
3. Сервер забирає готову web-сторінку, надсилає до браузеру, котрий продемонструє її користувачу, який надіслав запит [3].

1.2 Класифікація web-застосунків

Мабуть найбільш корисна класифікація web-застосунків в залежності від їх вигляду при створенні композитних моделей.

Згідно такої класифікації, до першого типу web-застосунків можна віднести програми, які не використовують композитні моделі взагалі. Наприклад, програми написані за допомогою простих скриптових мов програмування.

Другий тип web-застосунків – застосунки, які використовують для розширення функціональності web-серверів та реалізації бізнес-логіки універсальні компонентні моделі, не призначені для створення тільки web-застосунків.

Цей підхід набув широкого розповсюдження на ОС Windows, коли прикладна частина серверного застосунку реалізовувалась за допомогою COM/ActiveX-об'єктів.

Третій вид – web-застосунки, створені з використанням спеціальних композитних моделей. Найбільш популярними такими компонентами є так звані web-компоненти – сервлети і документи написані на мові Java. Ці web-компоненти виконуються під керуванням спеціалізованого контейнера [1].

1.3 Технологічні тренди web-розробки

Single Page Application (застосунки з однією сторінкою) - це web-застосунки, компоненти яких завантажуються один раз на одній сторінці, а контент завантажується по необхідності. І при переході між розділами сторінка не перезавантажується повністю, а тільки необхідні данні, які відображаються на сторінці [5].

Застосунки з однією web-сторінкою сильно виграють по швидкості роботи та зручності використання у класичних web-застосунків. За допомогою SPA можна добитися ефекту роботи застосунку, як десктоп додатку, без перезавантаження сторінки і значних затримок.

Ще кілька років тому додатки з однією web-сторінкою практично не підтримували пошукову оптимізацію і їх використовували переважно для створення особистих кабінетів та панелі адміністрування, то сьогодні створити web-застосунок з однією web-сторінкою, що повністю підтримує пошукову оптимізацію стало набагато простіше. Використовуючи застосунки з серверним рендерингом сьогодні ця проблема повністю зникла. Іншими словами, це все той же SPA, але при першому запиті, сервер генерує не просто дані, а створює готову HTML-сторінку і пошукові системи отримують готові сторінки зі всією розміткою та інформацією.

З розвитком інструментів для створення клієнтських web-застосунків, розробка та перехід на SPA в цьому і наступних роках будуть тільки рости. Якщо у вас застарілий застосунок, що повільно працює, ще й з повним перезавантаженням сторінки при переході між розділами, то можете впевнено оновитися до швидкого застосунку – зараз кращий час, технології уже дозволяють зробити це якісно та швидко [5].

Progressive Web Application (прогресивний web-застосунок) – це продукт спільної революції нативного застосунку та web-сайту. Загалом це web-застосунок, який виглядає та веде себе так, як і реальний нативний застосунок, може отримувати push-повідомлення, працювати в оффлайн режимі і т.д. При цьому користувачеві не потрібно завантажувати застосунок, а досить просто зберегти на робочий стіл [6].

Про те, що потрібно звернути увагу на цю технологію говорить і те що один з самих великих рушіїв для створення e-commerce проектів Magneto в 2018 році запустив версію PWA Studio. Платформа дозволяє створювати фронтенд на базі React для своїх рішень з підтримкою PWA. Для тих у кого вже є інтернет-проект, або ідея створення: не поспішайте писати повноцінний нативний застосунок, для початку зверніть увагу на технологію PWA. Можливо це буде найкраще рішення по відношенню ціни і якості, для вашого проекту [5].

Serverless – це ще один сучасний підхід до розробки. З-за його назви багато хто думає що це дійсно без серверна розробка, писати back-end код не потрібно, а будь-який front-end розробник зможе створити повноцінний застосунок, але це не так [6].

При створення Serverless-застосунку сервер потрібний як і раніше, база даних також. Основна відмінність даного підходу в тому, що back-end код представлений у вигляді хмарних функцій і дозволяє застосунку швидко та легко масштабуватися. При створенні такого застосунку розробник може сфокусуватися на бізнес-задачах та не думати про налаштування масштабу та налаштування інфраструктури, що в подальшому пришвидшує розробку застосунку та зменшує його вартість. Більш того FaaS підхід допоможе зберегти кошти на оренді серверного обладнання, так як використовує стільки ресурсів скільки потрібно для виконання задачі, а якщо немає навантаження, то серверний час взагалі не використовується і не оплачується.

Останні кілька років із-за новизни та своїх обмежень FaaS в основному використовувалася для невеликих проектів, але сьогодні завдяки еволюції ПЗ, універсальності, потужності серверів, з'являються інструменти, що дозволяють

прибрати обмеження, спростити та пришвидшити розробку хмарних застосунків. Це означає, що корпоративні бізнес-сценарії, в яких хмарна модернізація раніше вважалася неможливою, тепер є реальністю.

Голосовий пошук та навігація швидко набирають популярність. Серед драйверів росту – розвиток технологій, практично всі сучасні моделі смартфонів та деякі моделі ноутбуків мають опцію голосового пошуку. Великий вклад приносить і популярність голосових помічників та розумних колонок [6].

Все більше людей спілкуються зі своїми девайсами голосом, а за допомогою сучасних технологій машинного навчання ШІ розпізнає все більш складні запити.

За даними дослідження Global Digital Overview, доля використання голосового пошуку по всьому світу виросла на 9% за 2020 рік, при цьому 43% користувачів у віці від 16 до 64 років шукають інформацію за допомогою голосу мінімум раз у місяць [6].

Ріст популярності голосового пошуку та навігації по web-сторінці за допомогою голосу не викликає здивування, адже він швидко вирішує завдання та простий у використанні. Разом з цим росте попит на оптимізацію web-рішень для голосового пошуку з точки зору SEO та інших технічних стратегій.

1.4 Аналітичний огляд аналогів web-застосунку

Перед тим, як приступити до реалізації проекту, слід проаналізувати існуючі на даний момент web-застосунки подібного характеру та тематики. За допомогою цього ми зможемо зробити висновки про те, як потрібно створити власний web-застосунок на основі недоліків та переваг розглянутих ресурсів.

Серед розглянутих web-застосунків проаналізовані були:

1. Інтернет-магазин «Comfy» рис. 1.2 [7].
2. Інтернет-магазин «МОУО» рис. 1.3 [8].
3. «ITbox» Інтернет-магазин електроніки та комп'ютерної техніки рис. 1.4 [9].
4. «Комп'ютерний світ» Інтернет-магазин електроніки та офісної техніки рис. 1.5 [10].

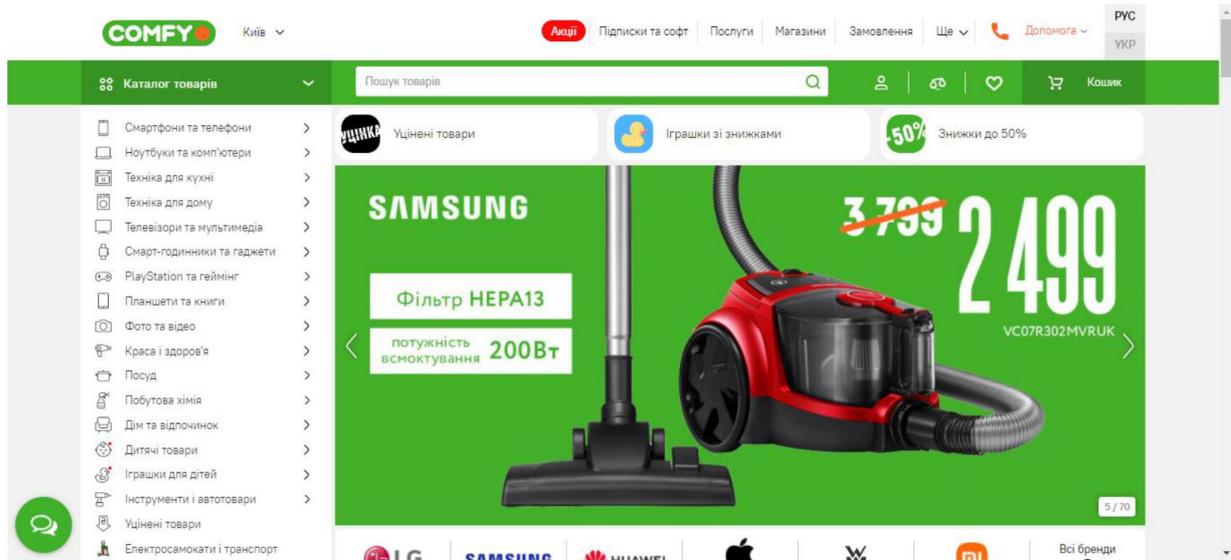


Рисунок 1.2 – Головна сторінка web-застосунку «Comfy»

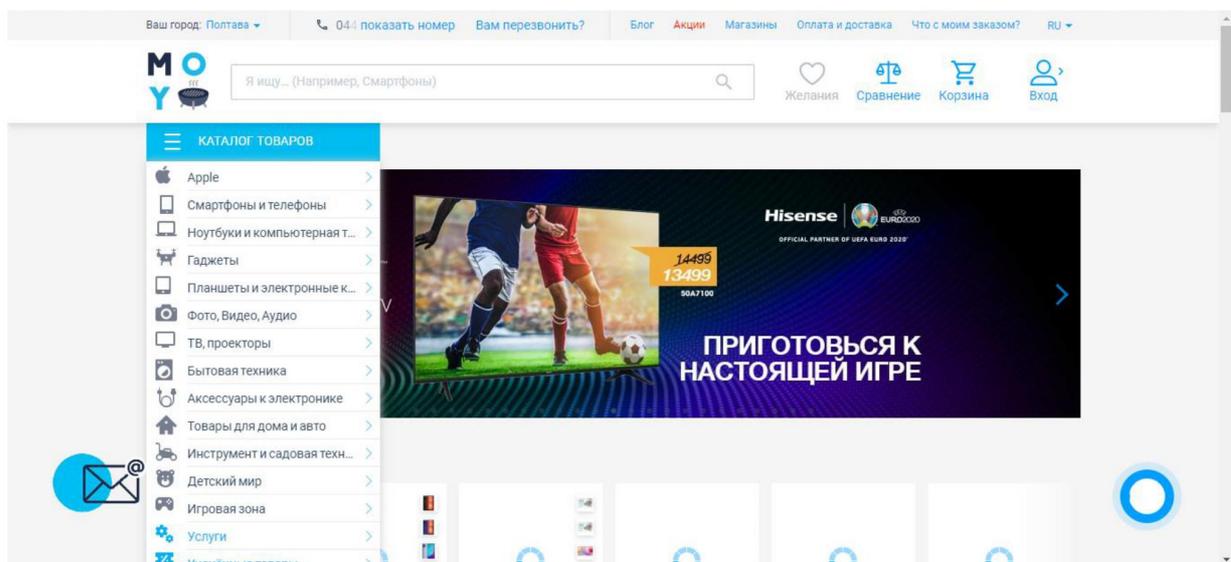


Рисунок 1.3 – Головна сторінка web-застосунку «МОУО»

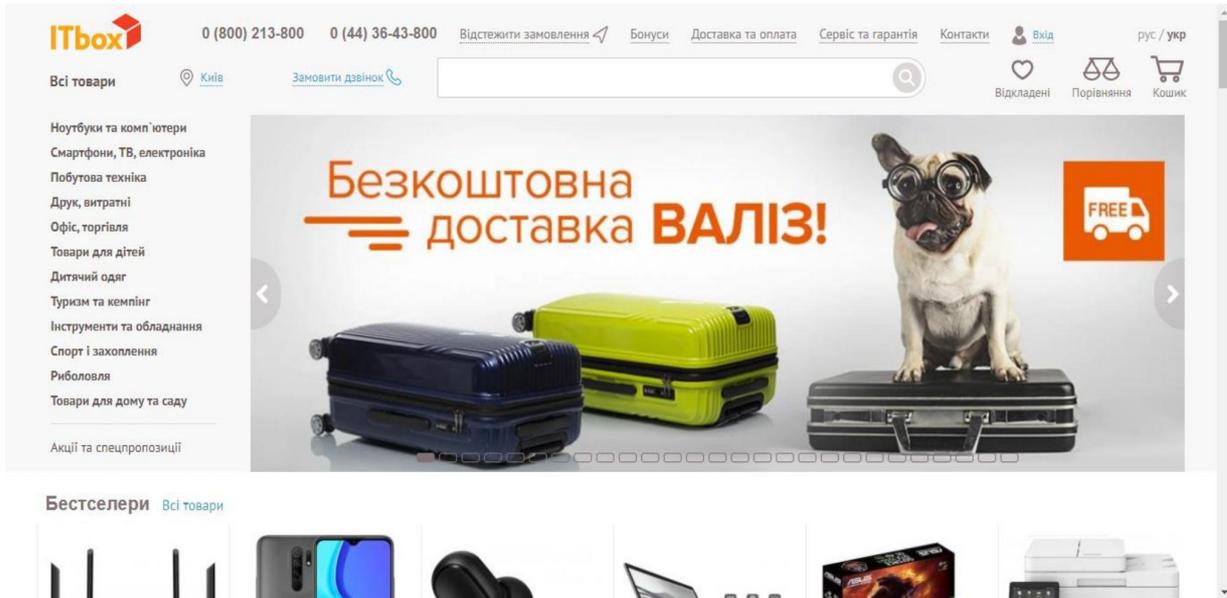


Рисунок 1.4 – Головна сторінка web-застосунку «ITbox»

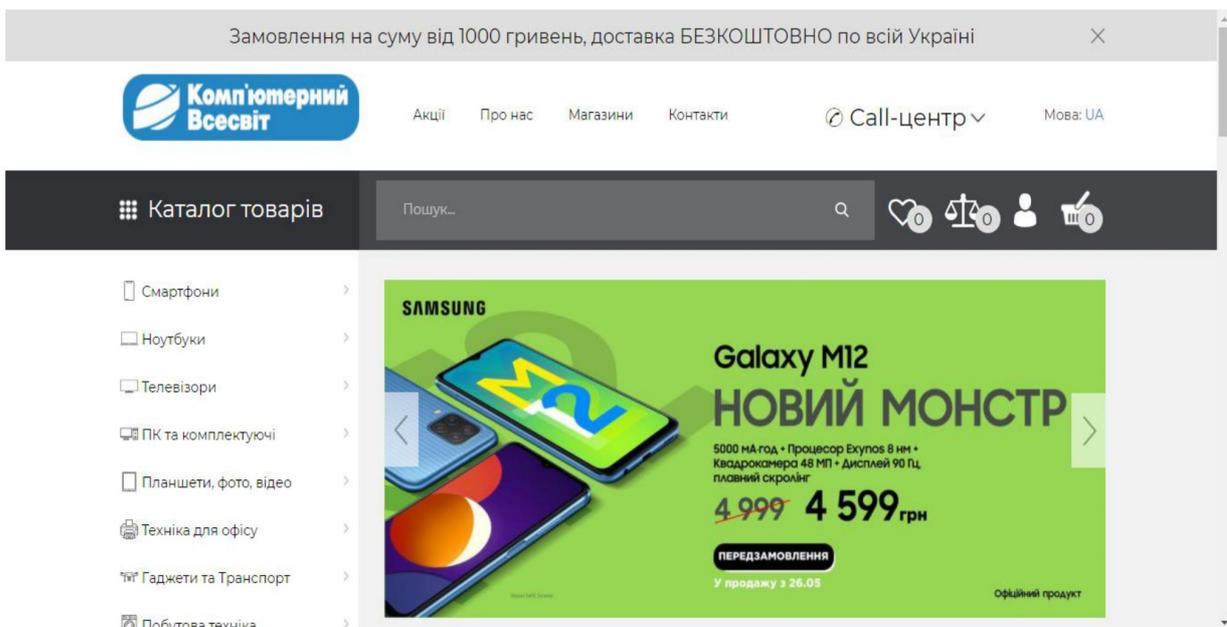


Рисунок 1.5 – Головна сторінка web-застосунку «Комп'ютерний всесвіт»

Проаналізувавши вищезазначені web-застосунки, можемо зробити висновки що, основна мета розробки це представити компанію в Інтернеті, продаж товарів онлайн, сервіс підтримки клієнтів. Web-застосунок повинен розроблятися таким чином, щоб виглядати максимально дружелюбним, зрозумілим та зручним для користувачів.

1.5 Огляд сучасних технологій для створення web-застосунків

В теперішній час перед web-розробниками ставляться багато різних задач – від створення інтерактивних розважальних сайтів до серйозних бізнес проектів, які потребують підвищеної надійності і безпеки від несанкціонованого доступу. Для їх реалізації потрібно правильно підібрати інструменти – мови програмування, фреймворки, або системи керування контентом (CMS – Content Management System).

На даний момент існує безліч мов програмування і з кожним днем їхня кількість змінюється. Переваги кожного з них можуть проявлятися в контексті певної задачі. Вибір мови програмування або фреймворку залежить від IT-відділу (програмістів) компанії, які мови програмування їм відомі та які краще застосувати в тому чи іншому проекті. Слід розмежувати такі поняття, як мова програмування та фреймворк.

Мова програмування – це такий собі базовий синтаксис, за допомогою якого можливе створення застосунків. Фреймворки вони надають розробникам різні набори бібліотек, які значно спрощують роботу [17].

Деякі мови та фреймворки представляють собою одне ціле, а серверні – для програм, що виконуються на стороні серверу. Цей розподіл продемонстровано на рисунку 1.6.

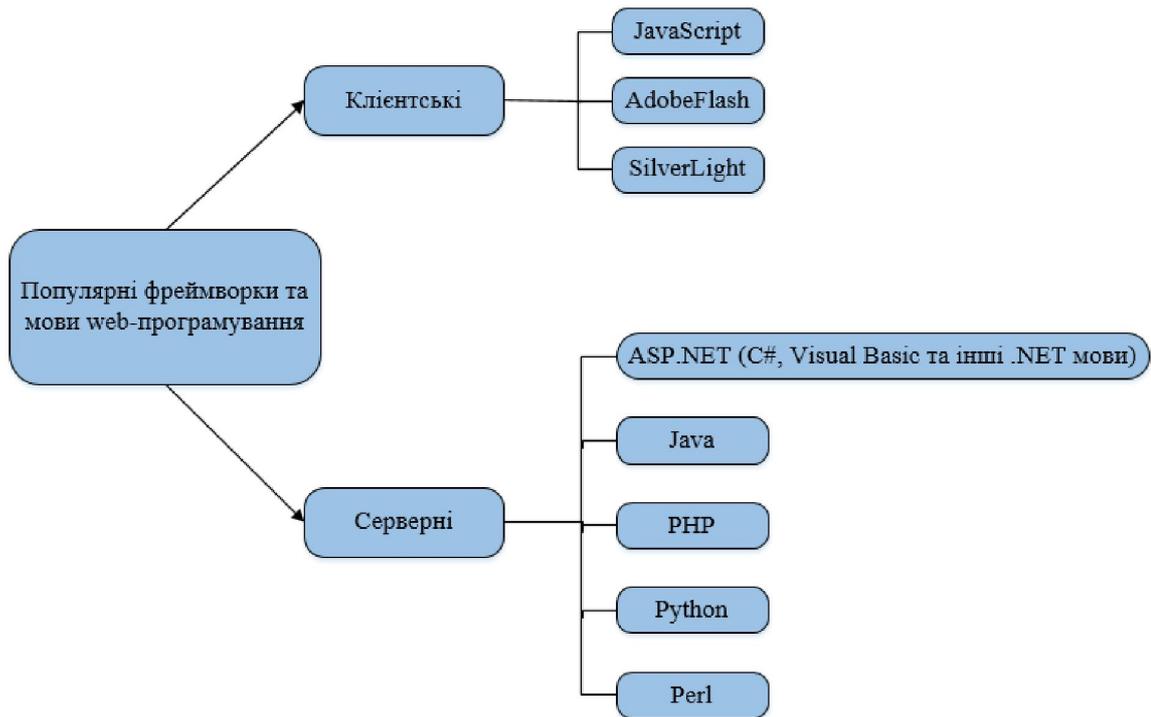


Рис 1.6. Популярні фреймворки та мови програмування

Клієнтські мови програмування - це тип мови обробка, якої здійснюється на стороні клієнта. Вони дають можливість змінювати вміст html-сторінки без перезавантаження її з серверу. Серед клієнтських мов web-програмування можна виділити JS, який так само, як і HTML лежить в основі багатьох web-технологій і вміння користуватися ним відноситься до основних знань web-розробника. До інших популярних мов, а точніше фреймворків належать: XHTML, CSS, SilverLight та будь-які .NET мови [2].

Серверні мови web-програмування виконуються на самому сервері, для чого використовують спеціальні програми. Таким чином на їх роботу не впливає, те з якого браузера виконується вхід, всі обчислювальні процеси виконуються на віддаленому ПК. Доступ до лістингу програми блокується, що гарантує стабільну роботу і безпеку серверу. Користувач бачить лише частину функціоналу, який виводиться до браузеру у вигляді HTML-сторінок. Серверні мови відкривають більше можливостей ніж клієнтські, використовуючи їх можна обмінюватися інформацією з сервером, що неможливо у випадку з клієнтськими мовами програмування. Серед серверних мов web-програмування

можна виділити PHP, Perl, Ruby, Java, C, Python, але найбільш розповсюдженою (більше 5 млн. серверів) серед них є мова PHP, особливістю якої є повна підтримка об'єктно-орієнтованого програмування.

Сучасні динамічні сайти обов'язково використовують базу даних як для зберігання різноманітного контенту, так і для реалізації своїх функцій. На сьогоднішній день набули популярності системи управління базами даних MySQL, PostgreSQL, ORACLE.

Ці бази мають спрощений функціонал, достатній для завдань web-розробки. Найбільш розповсюджена база даних MySQL. Вона досить проста в користуванні і має велику швидкість обробки запитів, що важливо при великій кількості звернень користувачів до серверу, на якому розміщено web-застосунок.

1.6 Висновки до першого розділу

Web-застосунки являють собою особливий тип програм, які створені на основі архітектури клієнт-сервер.

Кожний web-застосунок складається з 3 частин: інформаційної, програмний код та відомостей про конфігурацію. Інформаційна частина включає в себе описання web-сторінки в форматі HTML. Програмний код реалізує бізнес-логіку, оформлену у вигляді процедур обробки даних. Цей код виконується сервером та взаємодіє з динамічними елементами інформаційної частини, дозволяє динамічно формувати вміст web-сторінки, яка передається користувачу. Відомості про конфігурацію містять параметри, що визначають спосіб виконання web-застосунку на сервері, параметри безпеки, реакцію на помилки що виникають і т.д.

Розглянувши декілька сучасних web-застосунків на предмет відповідності їх поставленим цілям, а також коректності вибору інструментів для розробки можна виділити наступні характерні недоліки:

- надлишок або нестача представленої інформації;
- простота використання, користувач протягом 10-15 секунд повинен зрозуміти як користуватися додатком;

- захоплення анімацією, графічними зображеннями та відео фрагментами, які не несуть ніякої корисної інформації, що автоматично тягне за собою зменшення кількості потенціальних користувачів, так як збільшує вимоги до пропускної здатності інтернет-з'єднання та об'єм трафіку;
- некоректний вибір інструментів для розробки, тягне за собою необгрунтоване збільшення вартості готового застосунку.

Підводячи підсумок розділу можна відмітити, що сучасні web-застосунки повинні бути в першу чергу зручними для користувача і орієнтовані на нього, бути багатофункціональними і мати приємний, ненав'язливий дизайн. Більше половини користувачів заходять на web-застосунок за допомогою смартфонів та планшетів, тому адаптивний дизайн відіграє дуже важливу роль в побудові застосунків. Мінімізація flash-анімацій підвищує швидкість завантаження застосунку.

2. ОБГРУНТУВАННЯ ВИБОРУ ІНСТРУМЕНТІВ ДЛЯ РОЗРОБКИ ЗАСТОСУНКУ

Завдяки технічному прогресу та популярності web-розробки, сьогодні на ринку можна знайти багато передових інструментів, які спрощують чи роблять діяльність розробників зручною. Завдяки цьому, сьогодні можна використовувати перевірені бібліотеки, код, написаний іншими розробниками та отримувати більше можливостей, коли справа доходить до більш складних проєктів.

На сьогоднішній день існує безліч різноманітних інструментів, що дозволяють максимально оптимізувати розробку, та зробити всі процеси зручними і зрозумілими. Проте, з урахуванням цього, майже кожен день з'являються нові інструменти для розробки і вибір підходящих та дійсно найкращих стає важким завданням.

При виборі інструментів реалізації проєкту, важливими цілями були: якість майбутнього web-застосунку, сучасний та індивідуальний дизайн і швидкість побудови сторінок.

При виборі інструментів були розглянуті наступні платформи реалізації:

- системи керування вмістом (CMS) – Wordpress.
- PHP-фреймворки.
- HTML, CSS.
- PHP.
- JS.
- OpenServer.
- MySQL.

Для спрощення проєктування можна використати готову CMS або конструктор сайтів, але ці варіанти не підходять так, як мають певні обмеження.

Wordpress був відхилений, тому що він по більшій мірі представляє собою конструктор для сайтів, а для того щоб вивчити у вигляді фреймворків потрібні роки.

Відмова від PHP-фреймворків був обумовлений, малим досвідом використання та тим, що потрібно реалізувати проект в короткі терміни, а вивчення документації зайняло б значну частину часу.

2.1 HTML – мова гіпертекстової розмітки

Основою будь-якої web-сторінки є розмітка. Технологія розмітки HTML визначає структуру і можливе наповнення сторінки. HTML (Hypertext Markup Language) – це код який використовується для структурування і відображення web-сторінки та її контенту [11]. Він дає зрозуміти браузеру, як потрібно відображати завантажену сторінку. Мова складається з тегів – це своєрідні команди, які перетворюються в візуальні об'єкти в браузері користувача. Тегів існує багато і кожен змінює контент, який знаходиться між тегами.

HTML не є мовою програмування це мова розмітки і використовується, щоб повідомляти вашому браузеру, як відображати web-сторінки, які ви відвідуєте. Він може бути складним, або простим, в залежності від того, як захоче розробник. HTML складається з ряду елементів, які використовують, щоб вкладати або огортати різноманітні частини контенту, змусити контент відображатися або діяти певним чином. Теги можуть зробити слово, або зображення посиланням на будь-що, можуть зробити слова курсивом, зробити шрифт більше, або менше і так далі.

Мова HTML має свій власний набір символів, за допомогою яких web-браузери відображають сторінку. Ці символи, називаються дескрипторами, включають в себе елементи, що необхідні для створення гіперпосилань [12].

Одна з особливостей HTML-документів те, що сам документ містить тільки текст, а всі інші об'єкти вбудовуються в документ в момент його відображення браузером за допомогою спеціальних тегів та зберігаються окремо

Як і більшість інструментів, HTML має як і слабкі так і сильні сторони.

Плюси:

- Мова широко використовується з великою кількістю ресурсів і величезною спільнотою.

- Виконується від самого початку в кожному браузері.
- У відкритому доступі і цілковито безкоштовно.
- Чиста і послідовна розмітка.
- Офіційні web-стандарти підтримуються консорціумом World Wide Web.
- Легко інтегрується з базовими мовами, такими як PHP та JS.

Мінуси:

- В основному використовується для статичних web-сторінок. Для динамічної функціональності може знадобитися використання інших мов.
- Не дозволяє користувачеві реалізувати логіку. В результаті чого всі web-сторінки потрібно створювати окремо, навіть якщо вони використовують однакові елементи.
- Деякі браузери приймають нові функції дуже повільно.
- Іноді поведінку браузера важко передбачити.

2.2 CSS – каскадні таблиці стилів

CSS (Cascading Style Sheets) – мова описання зовнішнього вигляду HTML-сторінок. Одна з базових технологій в сучасному інтернеті. Практично ні один web-застосунок не обходиться без CSS тому HTML та CSS діють у одній зв'язці.

Одна з широкого спектру технологій прийнятих консорціумом W3C та отримавших загальну назву «стандарти Web». В 1990-х роках стало ясно що необхідно стандартизувати Web, створити єдині правила, по яким розробники та web-дизайнери проектували б web-застосунки [13].

Як не дивно, CSS вважається одночасно одним з самих простих та одним з самих складних інструментів web-розробника. Безперечно він досить простий на початку – ви визначаєте властивості стилю, значення конкретних елементів і це практично все, що потрібно знати. Проте в великих проектах ситуація стає доволі заплутаною та складною, щоб організувати CSS якимось осмисленим чином. Зміна будь-якого рядка CSS для стилізації елемента на одній сторінці, часто спричиняє непередбачені наслідки для елементів на інших сторінках.

Щоб розібратися в складності властивій CSS, створені найрізноманітніші вигляди передових практик. Проблема в тому, що до цього часу не має єдиної думки, які з них є кращими, а багато з них повністю суперечать один одному.

Найбільш повно підтримують стандарт CSS браузери, що працюють на рушіях Gecko (Mozilla Firefox), WebKit (Safari, Google Chrome) та Presto (Opera) [13].

Ось так виглядає сучасний CSS, з яким можуть виникнути певні складнощі, тому що він продовжує змінюватися і розвиватися швидкими темпами. Важливо пам'ятати історичний контекст, як розвивалася мережа з роками та багато розробників працюють на конкретними інструментами і методологіями, щоб найкращі практики CSS розвивалися разом з web-розробкою.

2.3 PHP – гіпертекстовий препроцесор

Важко в одному реченні дати повне уявлення про цю мову, тому опишу наступними характеристиками:

PHP – це препроцесор гіпертексту (HTML).

PHP – це серверна мова програмування.

PHP – це скриптова інтерпритована мова програмування.

Основне завдання PHP – змусити web-сторінку рухатися, змінюватися. Зазвичай HTML-сторінки статичні. Статичність означає, що після того, як створили сторінку і завантажили на сайт, при кожному звертанні до цієї сторінки браузер продемонструє її будь-якому користувачу в незмінному вигляді. Цього не завжди достатньо. Майже завжди користувачі заходять на web-застосунок за інформацією, яка весь час змінюється і потрібно відображати її актуальний стан. В теперішній час підтримується більшість хостинг-провайдерів та є одним із лідерів серед мов програмування, які використовуються для створення динамічних web-сторінок [14].

Якщо використовувати тільки HTML, вирішити такі завдання не вийде, тут нам і знадобиться PHP. Він отримує вхідний запит від web-серверу, виконує сценарій і повертає web-серверу результат у вигляді готового HTML-коду.

Сервер надсилає цей результат в браузер користувачеві, який в свою чергу, відображає її користувачеві.

PHP дозволяє змінювати web-сторінку на сервері безпосередньо перед тим, як вона буде надіслана браузеру. В ході виконання PHP може змінити, або динамічно створити будь-який HTML-код, який і є результатом виконання сценарію. Потім сервер надсилає цей код браузеру.

PHP досить простий для освоєння, але разом з цим може задовольнити запити професіональних програмістів. Не варто боятися довгого списку можливостей PHP. Хоча він і призначений для web-розробки, область його застосування не обмежується тільки цим.

Він представляє собою мову з відкритим вихідним кодом, для виконання сценаріїв на стороні серверу, що створюють web-сторінки. Окрім незалежності від браузерів пропонує просте і універсальне, незалежне від платформи рішення для електронної комерції і складних web-застосунків, в тому числі тих, які керуються СУБД. Підтримує більшість відомих на сьогодні СУБД, включаючи MySQL, ORACLE, Sybase та DB2.

Результати проведеного аналізу дозволяють зробити висновок про наявність переваг експлуатаційних характеристик PHP-технології. Головними перевагами PHP вбачаємо практичність, легкість у застосуванні, ефективність, продуктивність та гнучкість.

2.4 JavaScript – прототипна мова програмування

JS – це мова програмування, яка вбудована в код сторінки і виконується браузером на локальному комп'ютері користувача. За допомогою різноманітних операторів і функцій JS дозволяє маніпулювати елементами розмітки та її стилями. Такий підхід відкриває можливості по реалізації графічних ефектів та розширення функціоналу web-сторінок, перетворюючи її зі звичайного послідовного набору елементів у взаємодіючий один з одним інтерфейсх. За допомогою JS розробники намагаються досягти однакового відображення web-сторінок у різних браузерах, що на сьогоднішній день є серйозною проблемою, яка стоїть перед web-розробниками та власниками web-застосунків [17].

Якщо сторінки будуть відображатися з помилками в одному з браузерів, це може призвести до втрати, користувачів, які використовують його для доступу до інформації в інтернеті. Розвиток технологій також впливає на JS, останнім часом з'явилося і розвивається велика кількість бібліотек, що містять в собі найрізноманітніші функції, які вирішують типові задачі. Ну, а крім того для бібліотек розробляються плагіни – функціональні модулі, котрі реалізують закінчені рішення.

JS неймовірно універсальний і дружелюбний до новачків. Володіючи великим досвідом, можна створювати ігри, анімовану 2D та 3D графіку, повномасштабні web-застосунки з СУБД та багато іншого. Сам по собі доволі компактний, але досить гнучкий. Розробниками написано велика кількість інструментів поверх основної мови JS, які розблокують велику кількість додаткових функцій [12].

За допомогою цієї мови реалізується можливість реакції сторінки, або окремих її елементів на дії користувача. Сьогодні JS є основною мовою програмування для браузерів. Він повністю сумісний з операційними системами Windows, Linux, Mac OS, а також усіма популярними мобільними платформами.

Використання JS, а також його бібліотек на сайті дозволяє перейти на новий рівень розробки, зробити інтерфейс зручним та приємним для користувача. Вся унікальність даної мови полягає у тому, що він підтримується практично усіма браузерами та повністю інтегрується з ними, а все що можна зробити з його допомогою – робиться дуже просто. Ні одна інша технологія не вміщує в себе всі ці переваги разом.

2.5 Open Server – локальний web-сервер

Для розробки web-застосунків на локальному комп'ютері потрібні локальні сервери. Локальний сервер – це емулятор хостингу. З його допомогою web-застосунок створюється на ПК з можливим переносом на хостинг.

Включає в себе: Apache, nginx, PHP, MySQL, phpMyAdmin, FTP-сервер FileZilla [15].

Open Server – це портативний локальний WAMP/WNMP сервер, що має багатofункціональну керуючу програму, та велику кількість, компонентів що можна підключити. Представлений пакет програм це не просто аматорський збірка, це перший повноцінний професійний інструмент, створений спеціально для web-розробників з урахуванням їх рекомендацій та побажань.

Open Server являє собою WAMP комплекс суть якого, можна зрозуміти розшифрувавши аббревіатуру:

1. Windows – операційна система, для роботи в якій працює сервер.
2. Apache – web-сервер, який вмикається при запуску програми.
3. MySQL – популярна СУБД, яка є обов'язковою умовою роботи багатьох web-застосунків.
4. PHP – інтерпритатор серверної мови програмування.

Проект існує вже більше 10 років, він успішно зарекомендував себе, як першокласний простий та надійний інструмент, необхідний кожному web-розробнику. За роки існування в ньому було реалізовано все найкраще, що є на сьогодні в аналогічних програмних продуктах [6].

2.6 MySQL – вільна система управління реляційними базами даних

MySQL – це реляційна СУБД з відкрити вихідним кодом. В теперішній час одна з найпопулярніших в web-застосунках.

Раніше для довготривалого та безпечного зберігання інформації використовували файли: в них записували декілька рядків, які потім витягувались, для подальшої роботи. Проблема тривалого зберігання інформації достатньо актуальна в процесі розробки web-застосунків. Однак професіональні методи роботи з файлами досить важкі та трудомісткі, так як, потрібно ретельно звертати увагу на занесення інформації до файлів, сортування даних. Перераховані завдання з успіхом вирішують СУБД, котрі самі координують безпеку інформації, її сортування та дають можливість змінювати.

В реляційних базах даних вся інформація зберігається у окремих таблицях, завдяки чому досягається вигреш у швидкості та гнучкості. Таблиці зв'язані між собою за допомогою відношень, це дозволяє об'єднувати дані з декількох

таблиць під час запитів. SQL як частина системи MySQL можна охарактеризувати, як мову структурованих запитів, найбільш розповсюджену мову, що використовується для доступу до баз даних.

MySQL дуже швидка, надійна та легка в користуванні, також володіє рядом зручних можливостей, що розроблені в тісному контакті з користувачами. Спочатку сервер розроблявся для керування великими базами даних з метою забезпечення більш великої швидкості роботи в порівнянні з аналогами. І ось уже протягом декількох років даний сервер використовується в промисловій експлуатації з високими вимогами. Незважаючи на те, що навіть зараз MySQL постійно вдосконалюється, він вже сьогодні забезпечує широкий спектр корисних функцій. Завдяки своїй доступності, швидкості та безпеці дуже добре підходить для доступу до БД. Робота з даною системою не викликає великих труднощів, а підтримка серверу MySQL включена в поставку PHP.

2.7 Висновки до другого розділу

Для розробки застосунку була вибрана такий набір інструментів: PHP, HTML, CSS, JS, MySQL. Кожна з цих технологій має різне призначення, цілі та функції, але велику цінність вони представляють, коли працюють разом, а не окремо.

Фундаментом розробки є мова HTML. CSS є інструментом задання зовнішнього вигляду та позиціонування різноманітних елементів сторінки що дозволяє легко керувати зовнішнім виглядом застосунку.

PHP дозволяє створювати потужні динамічні застосунки, що володіють певною логікою поведінки. PHP також дає нам майже необмежені можливості при розширенні застосунку.

JS виступає в якості важливого, не завжди важливого доповнення. Він здатний значно підвищити зручність користування застосунку та його інтерактивність.

Нарешті MySQL дозволяє нам організувати взаємодію застосунку з БД, що і дозволяє нам у зв'язці з PHP створювати дійсно потужний функціонал web-застосунку.

3. РОЗРОБКА WEB-ЗАСТОСУНКУ ТОРГОВОЇ МЕРЕЖІ

3.1 Дизайн та структура web-застосунку торгової мережі

Розроблений застосунок повинен бути зручним для користувача та адміністратора, котрий буде керувати застосунком. За допомогою представлених нам можливостей, будь-який користувач без особливих професіональних знань та навичок у сфері інформатики та web-програмування повинен легко керувати контентом створеного застосунку.

Одним з етапів досягнення цієї мети є правильне визначення структури і створення дизайну даного web-застосунку. Адже саме на цій стадії ми візуально та технічно уявляємо, що хочемо отримати. Якщо підійти до цих етапів неграмотно та непрофесійно, то проект може вийти невдало.

Існує безліч модульних сіток для дизайну застосунку. Для проекту було обрано табличну сітку, в комірках якої завантажуються текстова, або графічна інформація. Верхню частину застосунку називають шапкою. Зазвичай в шапці розміщують логотип, що представляє продукцію або компанію. В шапці розміщено логотип компанії, головне меню, та метеоінформація. За потребою можна розмістити будь-яку інформацію. По центру розміщено основний контент застосунку, де буде виводитися інформація при натисканні певних посилань. Також там розташоване підменю з інформацією про товари. Нижню частину також називають підвал, в ній розташована інформація про компанію та контакти, зворотній зв'язок. Для створення зручного та красивого дизайну використано каскадні таблиці стилів CSS, за допомогою яких, надається стиль кожному елементу. Структура застосунку представлена на рис. 3.1.

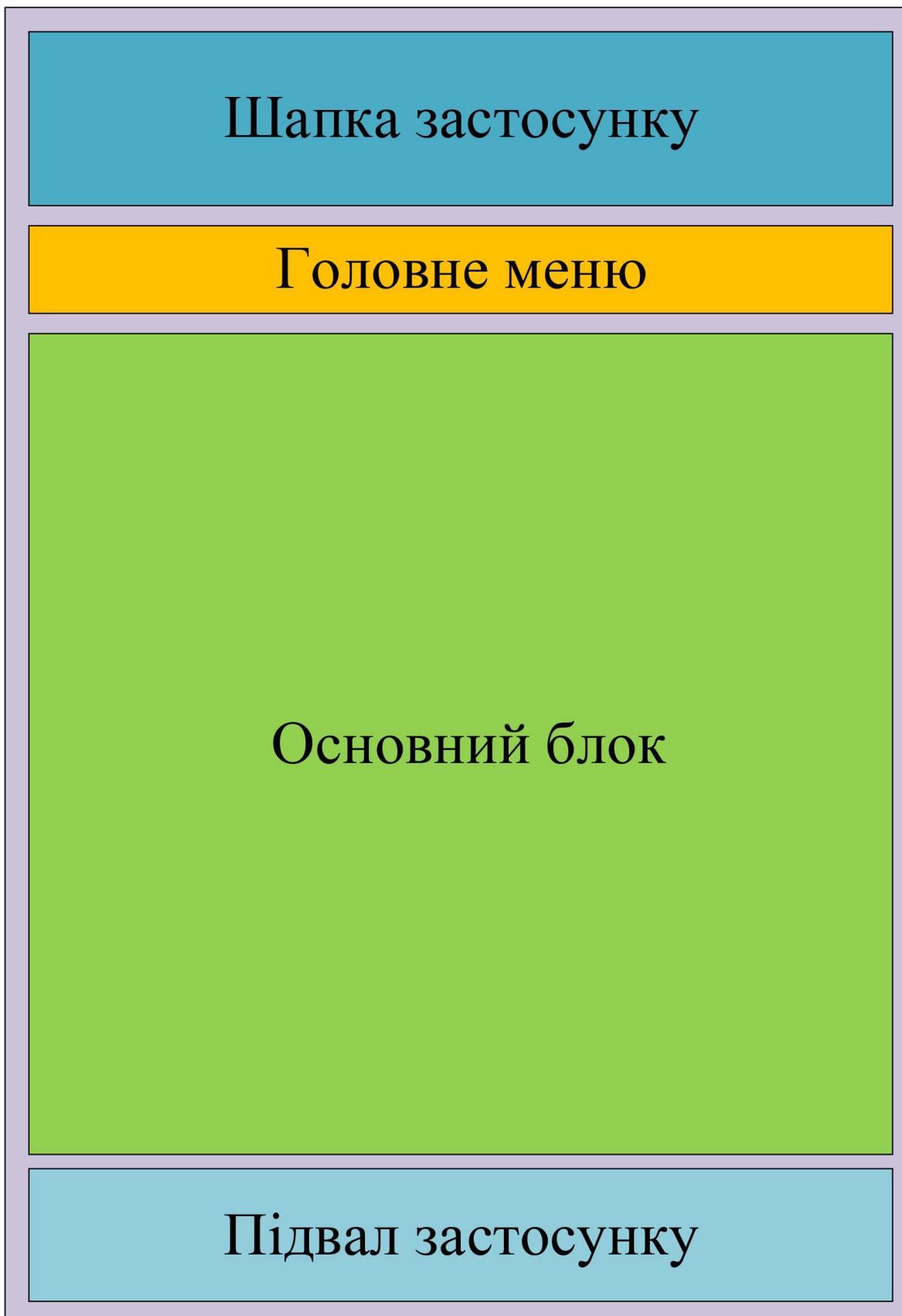


Рис 3.1. Структура web-застосунку торгової мережі

В верхній частині застосунку рис. 3.2 розміщено логотип компанії, зображення з розширенням png, що виводиться за допомогою спеціальної команди. Поряд розташовано інформація про компанію та вітання для користувачів, що змінюється в залежності від того авторизований користувач чи ні. Остання частина це метеоінформація в режимі реального часу, що змінюється при оновленні сторінки. Створена за допомогою JS бібліотек приймає інформацію з застосунку meteorprog [18].

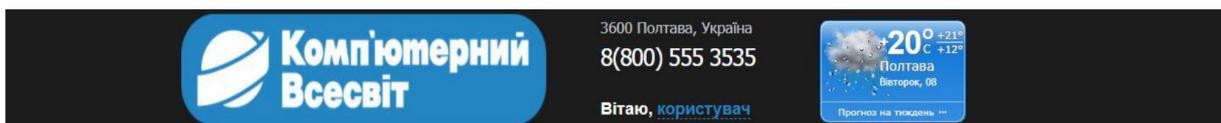


Рис 3.2. Шапка web-застосунку торгової мережі

Оразу під шапкою розміщено головне меню рис. 3.3, де знаходяться посилання на інші сторінки застосунку, а також додаткові функції для зручності використання web-застосунку. Головною функцією меню є вибір сторінок. При виборі будь-якого посилання в центральній частині виводиться інформація що прикріплена до даного розділу. В залежності від того авторизований користувач чи ні меню змінює свій вигляд рис. 3.4.

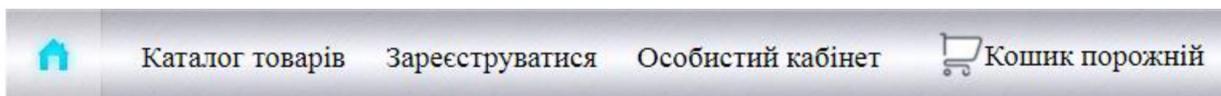


Рисунок 3.3. «Головне меню» web-застосунку торгової мережі для неавторизованого користувача



Рисунок 3.4. «Головне меню» web-застосунку торгової мережі для авторизованого користувача

При натисканні на перше посилання (у вигляді будинку) користувач потрапляє на головну сторінку web-застосунку торгової мережі. На цій сторінці відображена повна інформація про компанію, переваги компанії. Далі йде блок з новинами, в якому користувачі можуть дізнатися новини компанії, акції та інше рис. 3.4.

Мережа «Комп'ютерний Всесвіт» понад 19 років допомагає сотням тисяч людей втілювати свої задуми і здійснювати мрії за допомогою гаджетів!

Робота повинна бути продуктивною, спілкування - приємним, побут - легким і комфортним! Важливі справи залежать від якості речей, якими ми користуємося, а вміння робити правильний вибір on-line - економить час

Тому, чому варто вибрати нас?

- Багаторічний досвід і надійність
- Широкий асортимент
- Лояльна цінова політика
- Безкоштовна доставка в будь-який магазин мережі «Комп'ютерний Світ»
- Швидке обробки замовлень, професійна консультація
- Можливість оплати в розстрочку
- гарантійне і післягарантійне сервісне обслуговування



Що про нас говорять цифри?

- 19 років на ринку
- Більше 80 магазинів по всій території України
- Десятки і сотні брендів
- Кілька тисяч найменувань товарів

А як щодо Сервісу?

Гарантійний? Післягарантійний? Додатковий? - У нас є все! А головне - швидко і якісно.

Кілька слів про нас

Як з'явився «Комп'ютерний Всесвіт»? У дале-е-Еком 2002 році перша зірочка запалилася між кам'яних міста Лева. Слідом за нею - інші, а сьогодні це - велике сузір'я, простягнулося по всій Україні!

Наше гасло: «Безмежний простір твоїх можливостей»

Наша місія: об'єднувати людей навколо Всесвіту технологій і робити майбутнє доступним

Наші цінності:

ВІДДАНІСТЬ. Кожен магазин-зірка - особливий завдяки відданості справі зоряної команди.

ВІДПОВІДАЛЬНІСТЬ. Ми несемо відповідальність перед клієнтами, підтримуємо один одного і долучаємося до ініціатив, які роблять Всесвіт кращим і добрішим.

РОЗВИТОК. Ми орієнтовані на РОЗВИТОК, адже всесвіт технологій не стоїть на місці, тому і ми non-stop рухаємося вперед.

Для «Комп'ютерний Всесвіту» РОЗВИТОК - це:

- зростання команди і збільшення кількості лояльних покупців;
- вдосконалення експертних якостей, щоб гордо декларувати: «Знаю Все у Всесвіті»;
- поширення сучасних гаджетів по всьому Всесвіті і за його межами!

ДРАЙВ. Наш двигун - внутрішній **ДРАЙВ**. Кидаємо і приймаємо виклики в боротьбі за самовдосконалення, розум - відкритий до нового, серця сповнені енергією. **ГУМОР.** Що за політ до нових галактик без реактивного палива - **ГУМОРУ?** Ми прагнемо бути дотепними і керуватися принципами позитивного світогляду і здорового почуття гумору! Команда мережі «Комп'ютерний Всесвіт» - це dream team! Ми - за поєднання драйву, продуктивності і командної роботи! Відкривати нові можливості і магазини - ось, що ми любимо і вміємо робити! Наші цінності та бачення збігаються? Тоді ласкаво просимо в Команду. До речі, поки це читали, ми вже відкрили черговий магазин!

1-11-26

ти показали унікальну ірину»

иль ЗА3-2301 «Снага»
динила в себе несовместимые
! Данная грузопассажирская
ль была разработана в
дине 90-х годов прошлого века.
га» наделена внешностью,
ная отдалено напоминает всем
:тную «Газель».

2019-11-20

Maserati тестирует среднеторный спорткар

По неофициальным данным,
итальянцы трудятся над бензиновой
«шестеркой». Недавно всплывший
проект трехлитрового рядного
мотора FCA напрашивается для
сопоставления.

2021-05-31

Artem

qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq
qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq
qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq

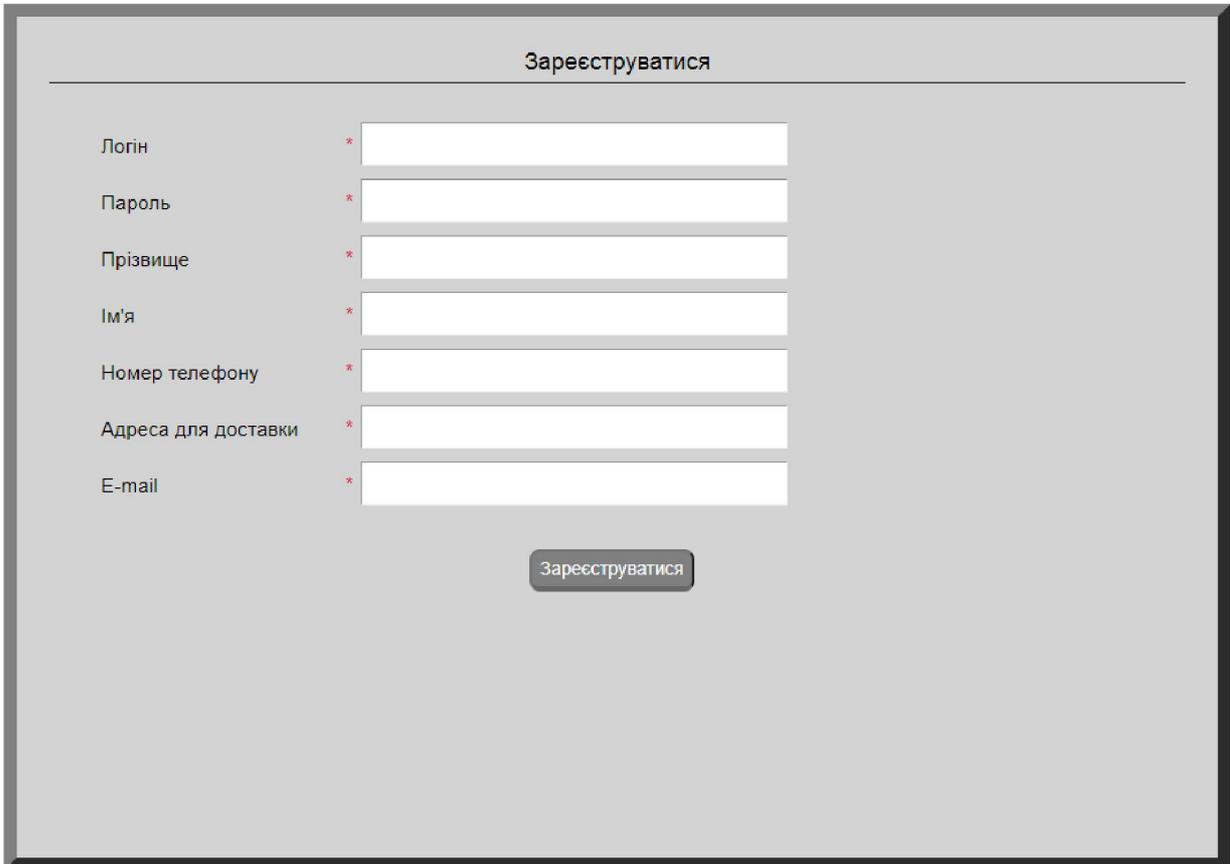
2019

В Се «Тас

Моде
объе,
вещи
моде
серед
«Сна
котор
извек

Рисунок 3.4. «Головна» сторінка web-застосунку торгової мережі

Наступний розділ це посилання на реєстрацію. При натисканні відображається сторінка з формою в якій користувачеві потрібно ввести деякі данні про себе та натиснути кнопку «Зареєструватися» після цього користувач буде зареєстрований у web-застосунку та зможе авторизуватися рис. 3.6.



The image shows a registration form with the title "Зареєструватися" (Register) centered at the top. Below the title, there are seven input fields, each with a red asterisk indicating a required field. The fields are labeled as follows: "Логін" (Login), "Пароль" (Password), "Прізвище" (Surname), "Ім'я" (Name), "Номер телефону" (Phone number), "Адреса для доставки" (Delivery address), and "E-mail". At the bottom center of the form, there is a button labeled "Зареєструватися" (Register).

Рисунок 3.6 сторінка «Зареєструватися» web-застосунку торгової мережі

Якщо ж користувач вже зареєстрований йому потрібно авторизуватися, для цього необхідно вказати логін та пароль, що був вказаний під час реєстрації. Далі натиснути кнопку вхід, після цього користувач буде направлений на головну сторінку з привітанням рис.3.7.

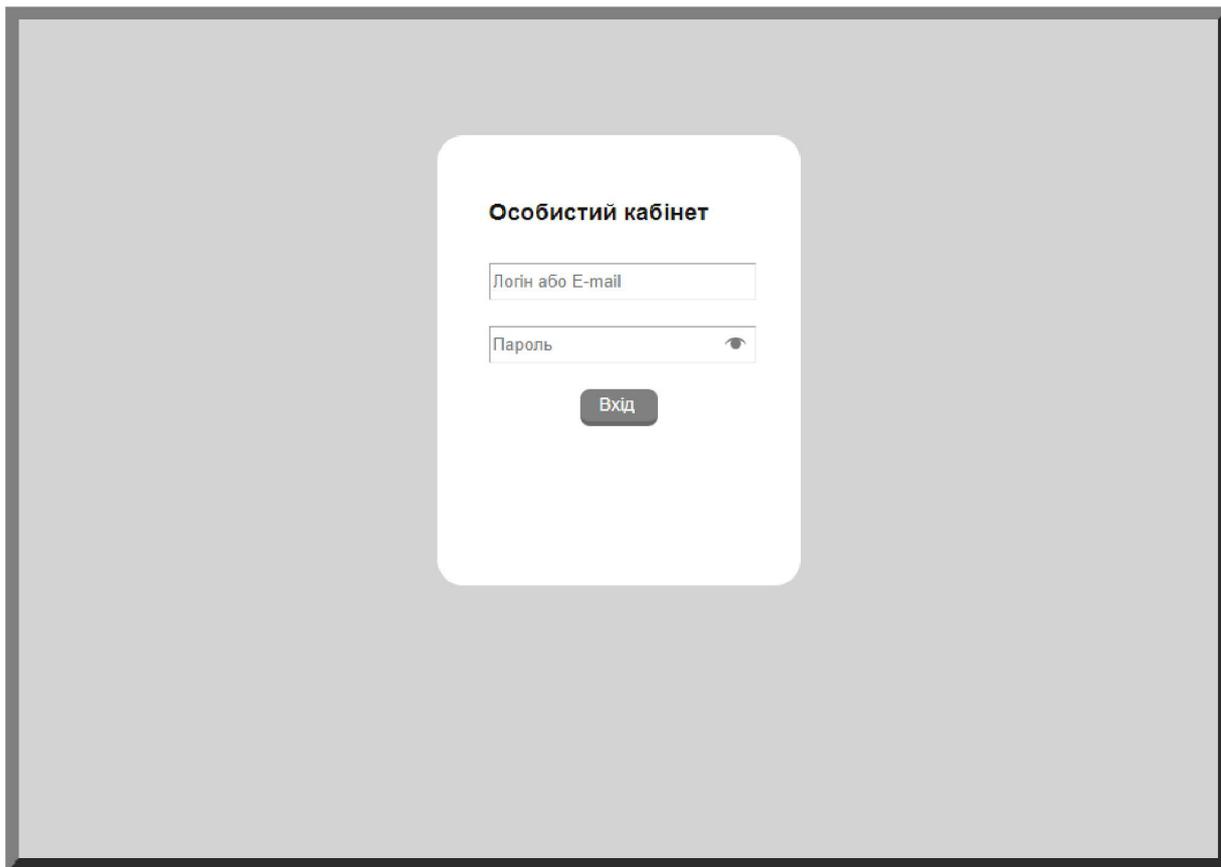


Рисунок 3.7 сторінка «Авторизуватися» web-застосунку торгової мережі

Останнє посилання це сторінка кошик товарів в якій відображаються обрані користувачем товари та відображена інформація для доставки товарів рис. 3.8.

1. Кошик товарів → 2. Контактна інформація → 3. Завершення

Крок 1 з 3 Очистити

Зображення	Назва товару	Кількість	Ціна	
	Xiaomi Redmi Note 8 Екран: 6.67", 2400x1080 (20:9), 395 ppi Пам'ять: 64 ГБ, ОЗП 6 ГБ Процесор: 8 ядер(а), 2.3 ГГц Камера: 4 модулі Відео: fullHD 60 к/с, ultraHD 4K	- 1 +	1 x 5 800 5 800 грн	
	Samsung Galaxy A12 Екран: 6.67", 2400x1080 (20:9), 395 ppi Пам'ять: 64 ГБ, ОЗП 6 ГБ Процесор: 8 ядер(а), 2.3 ГГц Камера: 4 модулі Вага: 209 г	- 1 +	1 x 5 199 5 199 грн	
	Samsung Galaxy A12 Екран: 6.67", 2400x1080 (20:9), 395 ppi Пам'ять: 64 ГБ, ОЗП 6 ГБ Відео: fullHD 60 к/с, ultraHD 4K Ємність батареї: 5020 мАгод Вага: 209 г	- 1 +	1 x 9 000 9 000 грн	
	Apple iPhone 12 mini Екран: 6.67", 2400x1080 (20:9), 395 ppi Пам'ять: 64 ГБ, ОЗП 6 ГБ Процесор: 8 ядер(а), 2.3 ГГц Ємність батареї: 5020 мАгод Вага: 209 г	- 1 +	1 x 30 199 30 199 грн	
			Разом 50 198 грн	

Далі

Рисунок 3.8 сторінка кошик товарів web-застосунку торгової мережі
Якщо користувач авторизований в головному меню буде відображено ще одне посилання. Сторінка профіль на якій знаходиться форма редагування

профілю на якій користувач зможе редагувати данні введені при реєстрації рис. 3.9.

The image shows a web form for updating a user profile. The title is 'Зміна даних профілю'. The form contains several input fields, each with a red asterisk indicating a required field. The fields are: 'Поточний пароль' (Current password) with a masked password '.....'; 'Новий пароль' (New password) which is empty; 'Прізвище' (Surname) with the value '12d21d12'; 'Ім'я' (Name) with the value '4124124124'; 'Номер телефону' (Phone number) with the value 'rqwqwrqwr'; 'Адреса для доставки' (Delivery address) with the value 'ewfewfwefewf' and a second empty line below it; and 'E-mail' with the value 'artem170500gmailcom'. A 'Зберегти' (Save) button is located at the bottom of the form.

Рисунок 3.9 сторінка для зміни даних профілю користувача

3.2 Розробка архітектури web-застосунку торгової мережі

Вибір архітектури здійснюється з урахуванням функціональності та характеристик інформації, виходячи з цього була обрана архітектура клієнт-сервер рис 3.10.

На рівні серверу можна досить чітко виділити наступні компоненти:

- рівень представлення, призначений для роботи з даними користувача;
- рівень бізнес-логіка, на якому розташовані модулі застосунку;
- рівень бази даних, призначений для зберігання даних.

Таким чином була визначена модель взаємодії клієнт-сервер.

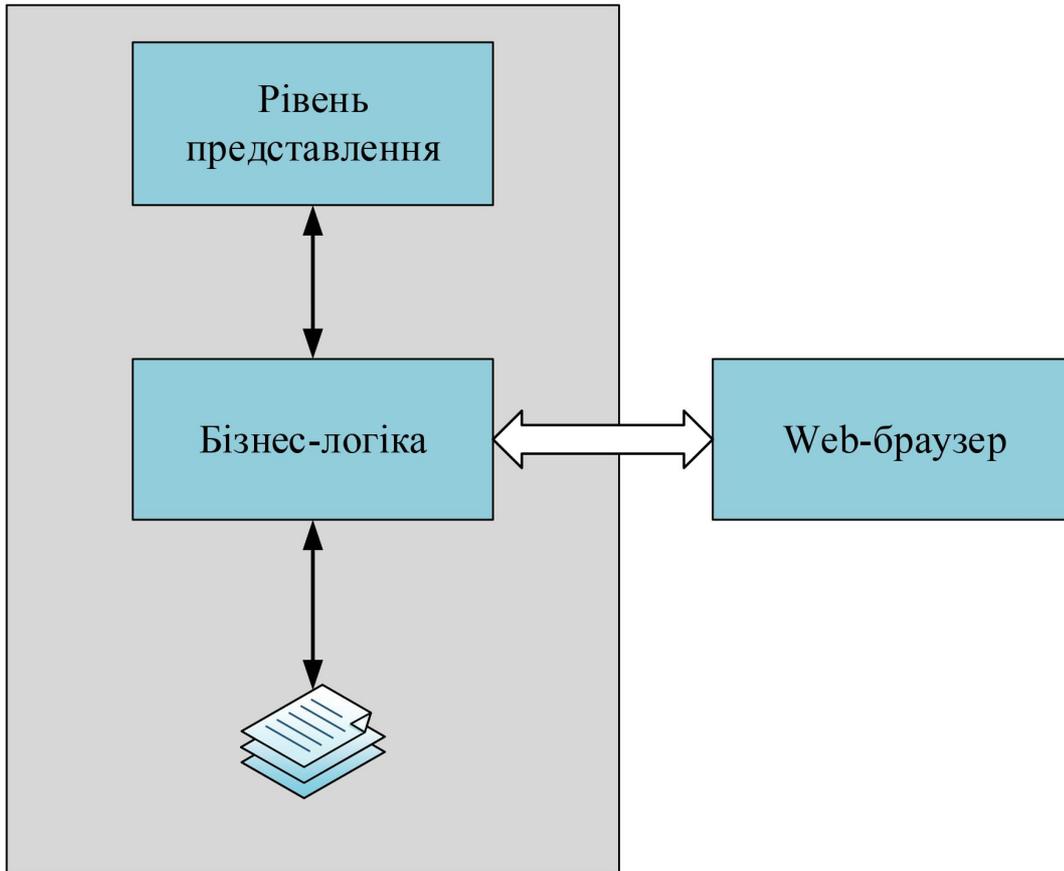


Рисунок 3.10. Архітектура клієнт-сервер

Для збереження даних в системі в таблицях БД розроблена логічна діаграма приведена у ДОДАТКУ А.

На рисунках 3.11-3.19 показані налаштування таблиць створених в РНРМуAdmin для БД.

В таблиці даних `cars_brand` зберігається інформація про категорії товарів: порядковий номер; тип товару та назва бренду.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1	id	int(11)		Нет	Нет		AUTO_INCREMENT	
<input type="checkbox"/>	2	type	varchar(25)	utf8_general_ci	Нет	Нет			
<input type="checkbox"/>	3	brand	varchar(50)	utf8_general_ci	Нет	Нет			

Рисунок 3.11 Структура таблиці даних `cars_brand`

В таблиці даних `table_cars` зберігається інформація про товар: назва; ціна одного екземпляру; бренд; короткий опис товару для відображення в каталозі; повний опис товару; коротка характеристика для відображення у каталозі; повна

характеристика товару; тип товару; кількість переглядів товару; номер категорії в якій знаходиться товар; зображення товару.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1	id	int(11)		Нет	Нем		AUTO_INCREMENT	
<input type="checkbox"/>	2	title	varchar(255)	utf8_general_ci	Нет	Нем			
<input type="checkbox"/>	3	price	int(11)		Нет	Нем			
<input type="checkbox"/>	4	brand	varchar(40)	utf8_general_ci	Нет	Нем			
<input type="checkbox"/>	5	mini_descript	text	utf8_general_ci	Нет	Нем			
<input type="checkbox"/>	6	descript	text	utf8_general_ci	Нет	Нем			
<input type="checkbox"/>	7	mini_features	text	utf8_general_ci	Нет	Нем			
<input type="checkbox"/>	8	features	text	utf8_general_ci	Нет	Нем			
<input type="checkbox"/>	9	type_product	varchar(255)	utf8_general_ci	Нет	Нем			
<input type="checkbox"/>	10	brand_id	int(11)		Нет	Нем			
<input type="checkbox"/>	11	image	varchar(255)	utf8_general_ci	Нет	Нем			
<input type="checkbox"/>	12	visible	int(11)		Нет	1			
<input type="checkbox"/>	13	count	int(11)		Нет	0			

Рисунок 3.12 Структура таблиці даних table_cars

В таблиці user_reg зберігається інформація про зареєстрованих користувачів: порядковий номер; логін; пароль; прізвище та ім'я; номер телефону; домашня адреса; e-mail; час реєстрації та ір-адреса.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1	id	int(11)		Нет	Нем		AUTO_INCREMENT	
<input type="checkbox"/>	2	login	varchar(100)	utf8_general_ci	Нет	Нем			
<input type="checkbox"/>	3	pass	varchar(100)	utf8_general_ci	Нет	Нем			
<input type="checkbox"/>	4	surname	varchar(100)	utf8_general_ci	Нет	Нем			
<input type="checkbox"/>	5	name	varchar(100)	utf8_general_ci	Нет	Нем			
<input type="checkbox"/>	6	numbphone	varchar(100)	utf8_general_ci	Нет	Нем			
<input type="checkbox"/>	7	address	varchar(100)	utf8_general_ci	Нет	Нем			
<input type="checkbox"/>	8	email	varchar(100)	utf8_general_ci	Нет	Нем			
<input type="checkbox"/>	9	datetime	datetime		Нет	Нем			
<input type="checkbox"/>	10	ip	varchar(50)	utf8_general_ci	Нет	Нем			

Рисунок 3.13 Структура таблиці даних user_reg

Таблиця cart містить в собі інформацію про товари додані до кошику для покупок: номер товару; ціна за один екземпляр; кількість товарів; дата; час та ір-адреса.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 cart_id	int(11)			Нет	Нет		AUTO_INCREMENT	
<input type="checkbox"/>	2 cart_id_product	int(11)			Нет	Нет			
<input type="checkbox"/>	3 cart_price	int(11)			Нет	Нет			
<input type="checkbox"/>	4 cart_count	int(11)			Нет	1			
<input type="checkbox"/>	5 cart_datetime	datetime			Нет	Нет			
<input type="checkbox"/>	6 cart_ip	varchar(100)	utf8mb4_unicode_ci		Нет	Нет			

Рисунок 3.14 Структура таблиці даних cart

В таблиці даних table_reviews зберігається інформація про відгуки користувачів: номер товару; позитивний відгук; негативний; коментар; дата та час відгуку; статус перевірки адміністратором.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 rev_id	int(11)			Нет	Нет		AUTO_INCREMENT	
<input type="checkbox"/>	2 id	int(11)			Нет	Нет			
<input type="checkbox"/>	3 name	varchar(100)	utf8mb4_unicode_ci		Нет	Нет			
<input type="checkbox"/>	4 good_rev	text	utf8mb4_unicode_ci		Нет	Нет			
<input type="checkbox"/>	5 bad_rev	text	utf8mb4_unicode_ci		Нет	Нет			
<input type="checkbox"/>	6 comment	text	utf8mb4_unicode_ci		Нет	Нет			
<input type="checkbox"/>	7 date_rev	date			Нет	Нет			
<input type="checkbox"/>	8 moderat	int(11)			Нет	Нет			

Рисунок 3.15 Структура таблиці даних table_reviews

Таблиця news містить в собі інформацію про новини; що відображаються у web-застосунку: номер новини; заголовок; текст новини; дата додавання новини.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 id	int(11)			Нет	Нет		AUTO_INCREMENT	
<input type="checkbox"/>	2 title	varchar(255)	utf8_general_ci		Нет	Нет			
<input type="checkbox"/>	3 text	text	utf8_general_ci		Нет	Нет			
<input type="checkbox"/>	4 date	date			Нет	Нет			

Рисунок 3.16 Структура таблиці даних news

В таблиці даних orders зберігається інформація про замовлення: номер замовлення; дата та час замовлення; підтвердження замовлення адміністратором; тип доставки; статус оплати; прізвище та ім'я; тип оплати; номер телефону замовника; примітки до замовлення; e-mail замовника та домашня адреса.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1	order_id	int(11)		Нет	Нет		AUTO_INCREMENT	
<input type="checkbox"/>	2	order_date	datetime		Нет	Нет			
<input type="checkbox"/>	3	order_confirm	varchar(10)	utf8mb4_unicode_ci	Нет	no			
<input type="checkbox"/>	4	order_arrive	varchar(255)	utf8mb4_unicode_ci	Нет	Нет			
<input type="checkbox"/>	5	order_pay	varchar(50)	utf8mb4_unicode_ci	Нет	Нет			
<input type="checkbox"/>	6	order_fio	text	utf8mb4_unicode_ci	Нет	Нет			
<input type="checkbox"/>	7	order_type_pay	varchar(100)	utf8mb4_unicode_ci	Нет	Нет			
<input type="checkbox"/>	8	order_phone	text	utf8mb4_unicode_ci	Нет	Нет			
<input type="checkbox"/>	9	order_note	text	utf8mb4_unicode_ci	Нет	Нет			
<input type="checkbox"/>	10	order_email	text	utf8mb4_unicode_ci	Нет	Нет			
<input type="checkbox"/>	11	order_address	text	utf8mb4_unicode_ci	Нет	Нет			

Рисунок 3.17 Структура таблиці даних orders

В таблиці даних orders зберігається інформація про придбані товари: номер покупки; номер замовлення; номер товару; кількість товарів.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1	buy_id	int(11)		Нет	Нет		AUTO_INCREMENT	
<input type="checkbox"/>	2	buy_id_order	int(11)		Нет	Нет			
<input type="checkbox"/>	3	buy_id_product	int(11)		Нет	Нет			
<input type="checkbox"/>	4	buy_count_product	int(11)		Нет	Нет			

Рисунок 3.18 Структура таблиці даних orders

Таблиця news містить в собі інформацію про зареєстрованих адміністраторів та доступ до перегляду інформації: порядковий номер адміністратора; логін; пароль; прізвище та ім'я адміністратора; посада; e-mail; номер телефону; доступ на перегляд, додавання та видалення замовлень; доступ на додавання та видалення товарів; доступ на додавання та видалення коментарів; доступ на перегляд та видалення користувачів; доступ на додавання та видалення новин; доступ на додавання та видалення категорій; доступ на перегляд адміністраторів.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1	id			Нет	Нет		AUTO_INCREMENT	
<input type="checkbox"/>	2	login	utf8mb4_unicode_ci		Нет	Нет			
<input type="checkbox"/>	3	pass	utf8mb4_unicode_ci		Нет	Нет			
<input type="checkbox"/>	4	name	utf8mb4_unicode_ci		Нет	Нет			
<input type="checkbox"/>	5	surname	utf8mb4_unicode_ci		Нет	Нет			
<input type="checkbox"/>	6	role	utf8mb4_unicode_ci		Нет	Нет			
<input type="checkbox"/>	7	email	utf8mb4_unicode_ci		Нет	Нет			
<input type="checkbox"/>	8	phone	utf8mb4_unicode_ci		Нет	Нет			
<input type="checkbox"/>	9	view_orders			Нет	0			
<input type="checkbox"/>	10	accept_orders			Нет	0			
<input type="checkbox"/>	11	delete_orders			Нет	0			
<input type="checkbox"/>	12	add_tovar			Нет	0			
<input type="checkbox"/>	13	edit_tovar			Нет	0			
<input type="checkbox"/>	14	delete_tovar			Нет	0			
<input type="checkbox"/>	15	accept_review			Нет	0			
<input type="checkbox"/>	16	delete_review			Нет	0			
<input type="checkbox"/>	17	view_client			Нет	0			
<input type="checkbox"/>	18	delete_client			Нет	0			
<input type="checkbox"/>	19	add_news			Нет	0			
<input type="checkbox"/>	20	delete_news			Нет	0			
<input type="checkbox"/>	21	add_cat			Нет	0			
<input type="checkbox"/>	22	delete_cat			Нет	0			
<input type="checkbox"/>	23	view_admin			Нет	0			

Рисунок 3.19 Структура таблиці даних reg_admin

3.3 Розробка діаграми функціональності web-застосунку торгової мережі

Функціонал застосунку поділений на дві частини:

1. Користувач – може використовувати навігацію по каталогах, зареєструватися або авторизуватися, переглядати товари, залишити відгук, надіслати товари до кошику та оформити замовлення.
2. Адміністратор – має такий самий функціонал, що і користувач, додатково володіє функціями по керуванню web-застосунком: управління каталогами; управління товарами; управління загальною інформацією; управління замовленнями; налаштування web-застосунку.

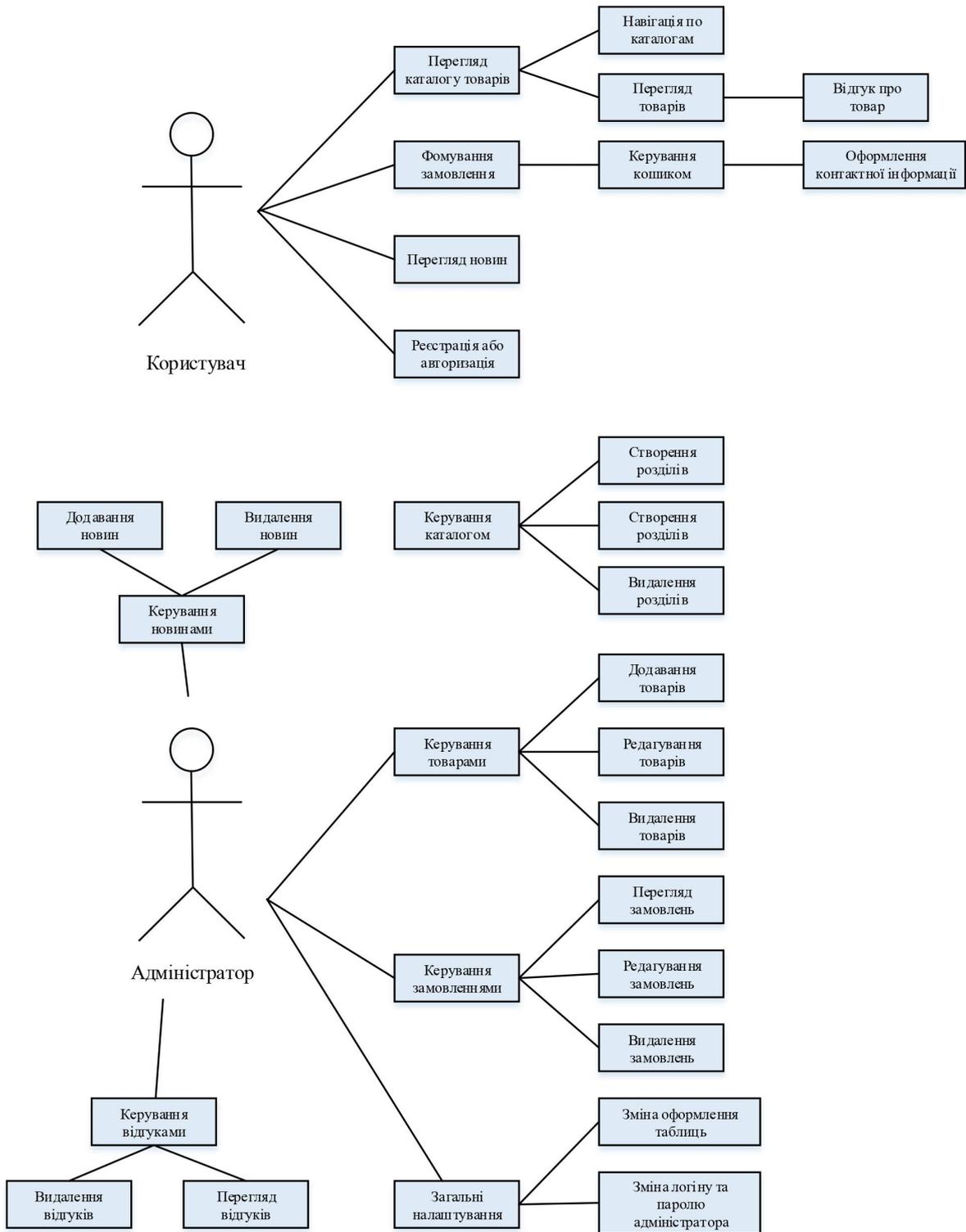


Рисунок 3.20 Діаграма функціональності

3.4 Розробка інтерфейсу web-застосунку торгової мережі

Створення якісного інтерфейсу передбачає реалізацію принципу «інтереси користувача понад усе» та відповідну методологію розробки всього програмного продукту. Інтерфейс повинен бути простим. При цьому мається на увазі забезпечення легкості в його вивченні та користуванні. Крім цього, він повинен надавати доступ до всього переліку функціональних можливостей передбачених web-застосунком. Один з можливих шляхів підтримання простоти – представлення на екрані інформації мінімально необхідної користувачу для виконання чергового кроку по замовленню товару.

Інтерфейс – в широкому сенсі – визначений стандартами кордон між взаємодіючими незалежними об'єктами. Інтерфейс задає параметри, процедури і характеристики взаємодії об'єктів [19].

Інтерфейс користувача – елементи та компоненти програми, які здатні впливати на взаємодію користувача з програмним забезпеченням. В тому числі:

- засоби відображення інформації, інформацію що відображається, формати і коди;
- керуючі режими, мова користувач-інтерфейс;
- пристрої та технології введення даних;
- діалоги, взаємодія і транзакції між користувачем і комп'ютером;
- зворотній зв'язок з користувачем;
- підтримка прийняття рішень в конкретній предметній області;
- порядок використання програми та документація на неї.

Ще декілька простих правил, що дозволять інтерфейсу бути зрозумілим користувачу:

1. Чим простіше, тим краще. Це зовсім не означає, що повинен бути тільки текст і контактна інформація, але не повинно бути марної інформації, шрифт повинен легко читатися. Графічні елементи повинні бути чіткими, виразними та швидко завантажуватися. Не варто використовувати

анімацію та мелодії, які довго завантажуються та відволікають увагу користувача.

Користувач переглядає сторінки зверху вниз. Найбільшу увагу зосереджують на верхній частині, як правило там розміщують більш корисна інформація.

Чим простіше виконана верхня частина, тим легше запам'ятати назву застосунку та саму компанію.

Типовою помилкою багатьох застосунків є завантаження інформацією верхню частину, серед якої ледве помітно назву компанії та назву застосунку. В результаті користувач переглядаючи дану частину сторінки не встигає помітити потрібну інформацію. Відбувається перенасичення інформацією.

2. Користувач повинен без труднощів знаходити товар, яким цікавиться та мати можливість отримати повну інформацію про нього (описання у вигляді тексту плюс зображення).
3. Товари повинні бути розподілені по групам. Необхідно забезпечити можливість пошуку товарів по назві бренду та типу. Для кожного товару зазвичай передбачені: короткий опис та повний опис плюс зображення.
4. При перегляді товарів у користувача повинна бути можливість сортувати товар по ціні, або бренду, типу.
5. При оформленні замовлення покупець повинен ввести конкретну інформацію: логін, пароль, ім'я, адресу, номер телефону та ін. Після реєстрації користувач має змогу авторизуватися.
6. У web-застосунку можуть бути і інформаційні розділи:
 - з даними про застосунок (сфера діяльності, адреса, телефони та ін.);
 - з інформацією про доставку;
 - з інформацією про акції та знижки;
 - новинами
 - інша корисна інформація

7. Зворотною, прихованою стороною web-застосунку є система керування. Вхід в систему керування здійснюється тільки після того, як адміністратор введе логін та пароль. Адміністратор має можливість повністю керувати вмістом web-застосунку:
- додавати чи видаляти товари, опис, зображення, змінювати ціну;
 - редагувати розділи (новини, статті, відгуки);
 - редагувати контактну інформацію;
 - переглядати історію замовлень та статистику користувачів.
8. Обережна робота з кольором. Правильно застосований колір, може, наприклад передавати незначні відмінності між однорідними елементами. Неправильно застосований колір може заважати роботі з застосунком.

3.5 Тестування створеного web-застосунку торгової мережі

Тестування застосунку розпочинається з реєстрації нового користувача. При переході за посилання користувач може побачити форму реєстрації з полями: логін, пароль, ім'я, прізвище, номер телефону, адреса для доставки та e-mail. Після введення всіх даних користувачу виведеться повідомлення про успішну реєстрацію і користувач буде зареєстрований рис. 3.21.

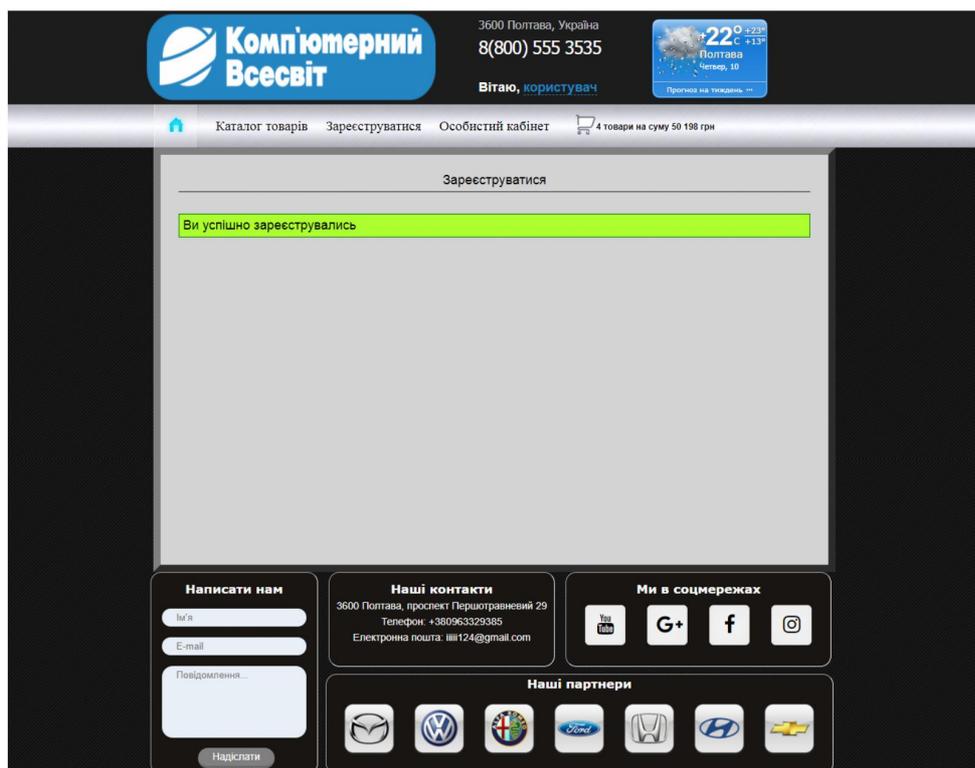


Рисунок 3.21 Тестування форми реєстрації - успішна реєстрація
 Якщо ж користувач не вводив дані або логін вже зареєстрований, користувач побачить повідомлення про помилки, та прохання ввести коректні дані рис. 3.22.

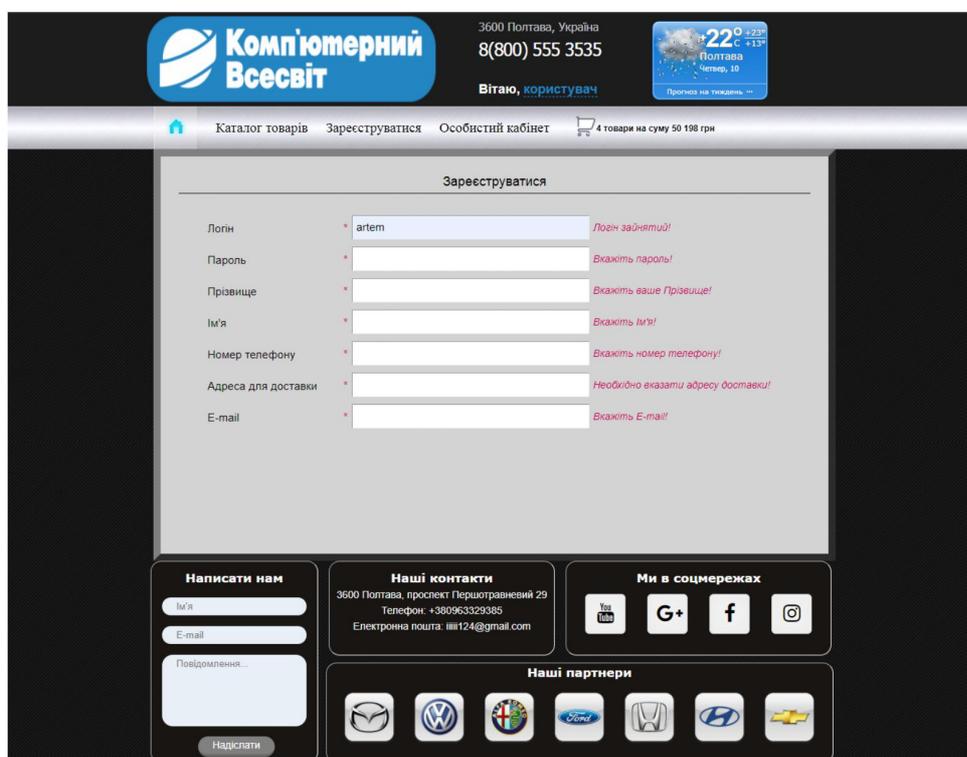


Рисунок 3.22 Тестування форми реєстрації – невдала реєстрація

Після реєстрації користувач зможе авторизуватися після натискання на посилання особистий кабінет. Далі потрібно ввести логін та пароль вказані при реєстрації, якщо дані введені вірно користувач буде направлений на каталог товарів та в шапці сайту буде привітання з ім'ям користувача введеним під час реєстрації рис. 3.23.

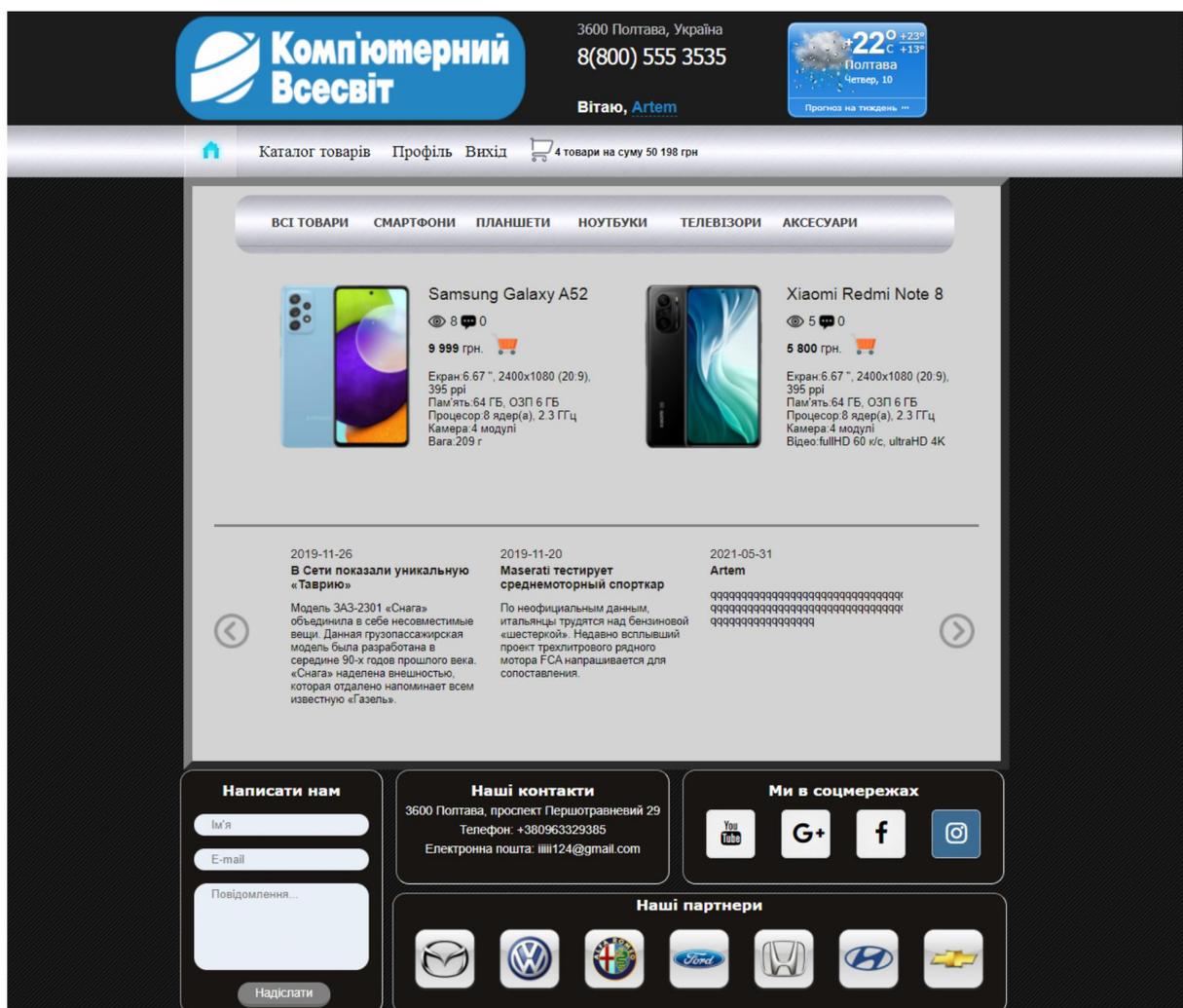


Рисунок 3.23 Тестування форми авторизації – успішний вхід

У випадку якщо данні введені не вірно користувачу буде повідомлено про це та прохання ввести коректні дані рис. 3.24

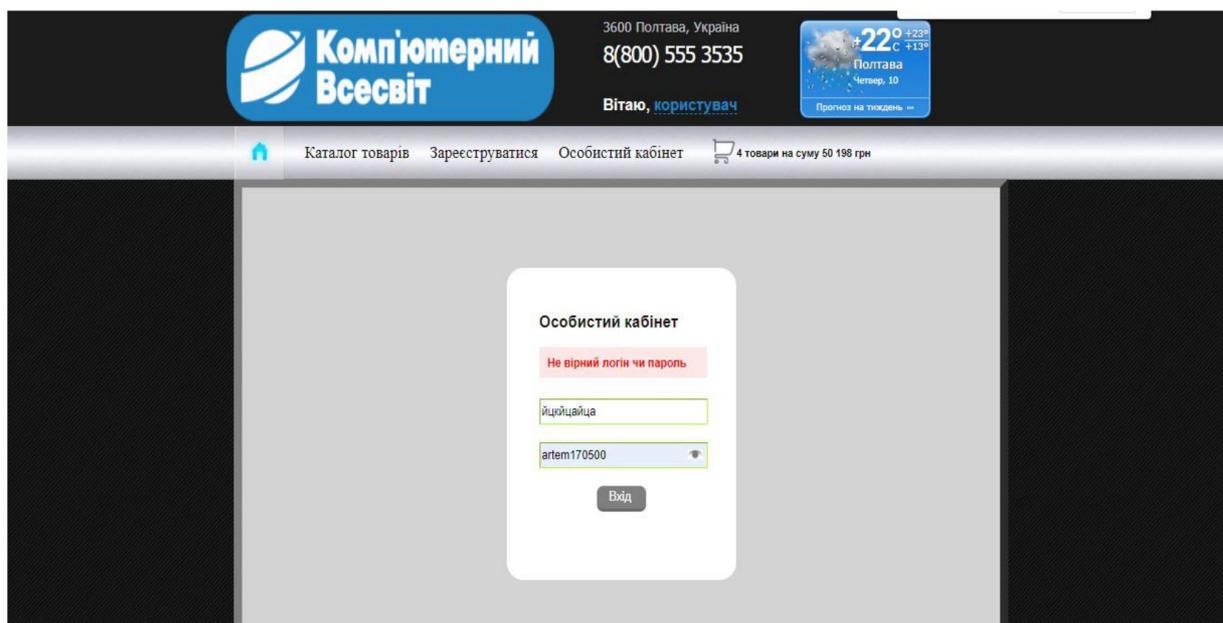


Рисунок 3.24 Тестування форми авторизації – невдалий вхід

Висновки до розділу

Керуючись даними принципами розробки інтерфейсу, було вирішено зробити ставку на простоту та інформативність, щоб користувач потрапляючи на сторінку, отримав чітку інформацію про товари. Також про те, як він зможе оплатити замовлення, які умови та терміни доставки та ін.

У web-застосунку повинен бути реалізований швидкий та зручний пошук товару, що необхідний користувачу, так як не всі мають необмежений час та доступ в інтернет.

ВИСНОВОК

В процесі виконання було розроблено web-застосунок підприємства, який призначений для продажу різних товарів через мережу Інтернет.

Областю призначення даного застосунку є інтернет-торгівля.

Створений застосунок є складним програмним комплексом, побудованим за технологією клієнт-сервер. Його ядром проекту є СУБД, рушієм написаний на мові програмування PHP, інтерфейс користувача виконаний у вигляді web-застосунку торгової мережі. Розділи формуються динамічно, вбудовані в відповідну структуру і дизайн застосунку, тобто всі зображення завантажуються одноразово при відкриванні стартової сторінки. Розробка проекту відбувалася «з нуля», та на етапі проектування було переглянуто декілька реальних програмних продуктів. Це дозволило створити справжній комерційний продукт, що задовольняє реальні потреби компанії.

В роботі розглянуті актуальні технології розробки універсального застосунку для продажу товарів, який пропонує користувачам переглянути та придбати товари. Застосунок пропонує користувачу переглянути та вибрати товари, котрі можна придбати не виходячи з дому. Створення застосунку дозволить просувати та продавати товари, зменшити витрати на рекламу, здобути новий ринок збуту товару, а також збільшити клієнтську аудиторію і як наслідок, збільшення прибутку.

Web-застосунок має зручний і зрозумілий інтерфейс, завдяки якому не потрібно багато зусиль в освоєнні роботи з застосунком.

До числа основних відмінних рис можна віднести наступні:

- Невеликий розмір файлів з кодом web-сторінок, що забезпечує швидке завантаження з мережі на ПК користувача.
- Блок адміністраторів, для керування вмістом, за допомогою якого зручно працювати навіть звичайним користувачам.
- Гнучкість, відкритість і модифікація, за допомогою простих інструментів.

Результати тестування показали, що система працює без помилок, є можливість роботи web-застосунку у всіх популярних браузерях.

Застосунок має великий потенціал, в подальшому можлива розробка додаткових функцій, для підвищення продажів, та просування компанії в мережі інтернет.

Результати дипломної роботи можуть бути використані для дослідження за даною тематикою при розробці web-застосунків, в подальшому можлива розробка додаткових функцій (підключити системи онлайн сплати, налаштувати акції та знижки для зареєстрованих користувачів, можливість онлайн відстеження доставки товарів та ін.), які дозволять збільшити кількість користувачів та підвищити продажі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Дослідження: Як змінювалися звички українських інтернет-покупців під час пандемії [інфографіка]. – Режим доступу:
<https://itc.ua/news/doslidzhennya-yak-zminuyutsya-zvichki-ukra%D1%97nskih-internet-pokupcziv-pid-chas-pandemi%D1%97/>.
2. Клієнт-серверна архітектура та ролі серверів. – Режим доступу:
<https://medium.com/@IvanZmerzlyi/клієнт-серверна-архітектура-та-ролі-серверів-9893d8048229> .
3. Що таке статичні і динамічні сторінки. – Режим доступу:
<https://qipu.ru/uk/mobilnye-sovety/chto-takoe-sticheskie-i-dinamicheskie-web-stranicy-sticheskaya-stranica-v.html>.
4. Классификация web-приложения по виду используемых при создании компонентных моделей. – Режим доступу: <http://galleo.ru/articles/w168>.
5. Технологические тренды web-разработки. – Режим доступу:
<https://habr.com/ru/post/451572/>
6. Ключевые тренды web-разработки в 2021 году. – Режим доступу:
<https://vc.ru/azoft/190597-klyuchevye-trendy-veb-razrabotki-v-2021-godu>
7. Головна сторінка web-застосунку «Comfy». – Режим доступу:
<https://comfy.ua/ua>
8. Головна сторінка web-застосунку «МОУО». – Режим доступу:
<https://www.mojo.ua/>
9. Головна сторінка web-застосунку «ITBOX». – Режим доступу:
<https://www.itbox.ua/>
10. Головна сторінка web-застосунку «KVSHOP». – Режим доступу:
<https://kvshop.com.ua/>
11. Поняття про мову розмітки, гіпертекстовий документ та його елементи. – Режим доступу: <https://naurok.com.ua/urok-ponyattya-pro-movu-rozmitki-gipertekstoviy-dokument-ta-yogo-elementi-html-fayli-200638.html>

12. Джамса Крис. Эффективный самоучитель по креативному Web-дизайну. HTML, XHTML, CSS, JavaScript, PHP, ASP, ActiveX. Текст, графика, звук и анимация. Пер с англ. / Крис Джамса, Конрад Кинг, Энди Андерсон - М.: ООО "ДиаСофтЮП", 2005.
13. Дэйв Ши, Молли Е. Хольцшлаг. Философия CSS-дизайна. / Пер. с англ. А. Слинкина. — М.: ИТ Пресс, 2005. - 312 с.
14. Веллинг, Л. Разработка web-приложений с помощью PHP и MySQL [Текст] / Л. Веллинг, Л. Томсон. - Вильямс 2010 - 848 с.
15. Чиртик, А.В. Популярный самоучитель HTML [Текст] / А.В. Чиртик. - СПб.: Питер, 2012.
16. Гудман Д. JavaScript и DHTML. Сборник рецептов. Для профессионалов, / Питер, 2004 г.
17. Языки web-программирования – какой выбрать? – Режим доступа: <https://developer.guru.net/post/web-programming-languages/>
18. Погода в Україні. – Режим доступа: <https://www.meteorprog.ua/ua/>
19. Дужий В.И., Волкова А.А., Галькевич А.А., Годунов А.С. - Программное обеспечение для компьютерных систем и сетей. Разработка технического задания. друк. Учебное пособие. - Х.: Нац. аерокосмічний ун-т "Харк. авіац. ін-т". - 2007.
20. Сокол Г.В. Аналіз сучасних мов програмування для створення веб-застосунків / Г.В. Сокол, А.С. Міщенко// Тези 73-ї наукової конференції професорів, викладачів, наукових працівників, аспірантів та студентів Національного університету «Полтавська політехніка імені Юрія Кондратюка». Том 1. (Полтава, 21 квітня – 13 травня 2021 р.) – Полтава: Національний університет імені Юрія Кондратюка, 2021. – С. 15-16.
21. G. Sokol Prospects for the use of MIMO based on IEEE 802.11ac IN IoT technologies / G. Sokol, P. Podhornyi, A. Mishenko, R. Piskovyi, D. Herasymov // // Системи управління, навігації та зв'язку, – 2020. – Вип. 4 (62). – С. 126-131.– doi: 10.26906/SUNZ.2020.4.126

CHAPTER 1 ANALYTICAL PART

1.1 The concept of web application

A web application, or web system, is a variety of software products that perform certain functions with a client-server architecture. The client is used to indicating the program used to run the web application. A server is an application that stores information. Usually, the "client" is a browser, a web server [2]. The structure of the web interface in a general view and the information transmission channels are showing in Many people think that a web application is a regular site, but it is not. For me, the site is primarily something informative, static: a business card, a recipe site, or a city portal. A set of pre-prepared HTML files that lie on a remote web server and are displayed, to the browser on request.fig. 1.1.

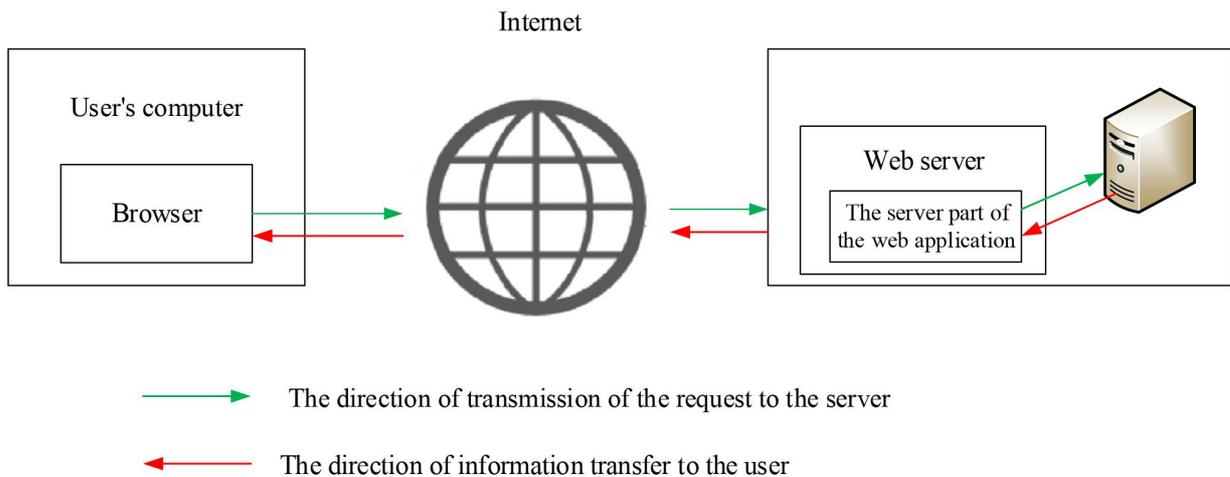


Figure 1.1 - structure of the web application in general

Many people think that a web application is a regular site, but it is not. For me, the site is primarily something informative, static: a business card, a recipe site, or a city portal. A set of pre-prepared HTML files that lie on a remote web server and are displayed, to the browser on request.

Sites contain a variety of statistics, which, like the HTML file is not created on the fly. Most often, it is pictured, CSS-files, JS-scripts, but there can also be any other files.

In addition, web applications are technically more complex. Here HTML pages are generated at the request of the user. Email clients, social networks, online stores, these are all web applications.

A static web page is displayed the same for all users. How it works:

1. The user enters a query or page address in the address bar.
2. The browser sends it to the webserver.
3. The server analyzes the request and determines that there are no special marks.
4. Sends the web page to the browser without changing any data on it.

In the case of dynamic pages, the scheme is as follows:

1. The browser sent a request to the webserver. For example, information was received that this user has a set of features, in the presence of which, for him to display certain information, then the page will be dynamic.
2. The web server sends it to the application server, where special software applies rules and instructions to add special variables. For example, the user is authorized in the system. Information from his name and other information for him will be displayed for him.
3. The server picks up the finished web page, sends it to the browser, which will show it to the user who sent the request.

1.2 Classification of web applications

Perhaps the most useful classification of web applications depending on their type when creating composite models.

According to this classification, the first type of web application can include programs that do not use composite models at all. For example, programs are written using simple scripting languages.

The second type of web application is an application that uses universal component models to extend the functionality of web servers and implement business logic, not just to create web applications. This approach became widespread on Windows when the application part of the server application was implemented using COM / ActiveX objects.

The third type - web applications created using special composite models. The most popular such components are the so-called web components - servlets and documents written in Java. These web components are executed under the control of a specialized container [1].

1.3 Technological trends of web-development

Single Page Applications are web applications whose components are downloaded once on one page and the content is downloaded as needed. And when moving between sections, the page does not reload completely, but only the necessary data that is displayed on the page [5].

Applications with one web page greatly benefit from the speed and ease of use of classic web applications. With SPA you can achieve the effect of the application as a desktop application, without reloading the page and significant delays.

Until a few years ago, applications with one web page did not support search engine optimization and they were used mainly to create personal accounts and administration panels, today it is much easier to create a web application with one web page that fully supports the search engine optimization. Using server-side rendering applications, this problem has completely disappeared today. In other words, it's still the same SPA, but on the first request, the server generates not just data but creates a ready-made HTML page and search engines get ready-made pages with all the markup and information.

With the development of tools for creating client web applications, the development and transition to SPA in this and subsequent years will only grow. If you have an outdated application that runs slowly, even with a full-page reload when switching between sections, you can confidently upgrade to a fast application - now is the best time, technology already allows you to do it quickly and efficiently [5].

Progressive Web Application is a product of the joint revolution of native applications and websites. In general, this is a web application that looks and behaves like a real native application, can receive push messages, work offline, etc. In this case, the user does not need to download the application but simply save it to the desktop [6].

The fact that one of the biggest drivers for the creation of e-commerce projects Magneto in 2018 launched a version of PWA Studio says that you need to pay attention to this technology. The platform allows you to create a frontend based on React for your solutions with PWA support. For those who already have an Internet project or an idea to create: do not rush to write a full-fledged native application, first pay attention to PWA technology. Perhaps this will be the best solution in terms of price and quality for your project [5].

Serverless is another modern approach to development. Because of its name, many people think that it really is without server development, you do not need to write back-end code, and any front-end developer will be able to create a full-fledged application, but this is not the case [6].

When creating a serverless application, the server is still needed, the database as well. The main difference of this approach is that the back-end code is presented in the form of cloud functions and allows the application to scale quickly and easily. When creating such an application, the developer can focus on business tasks and not think about scalability and infrastructure settings, which further speeds up the development of the application and reduces its cost. Moreover, the FaaS approach will help save money on renting server equipment, as it uses as many resources as needed to perform the task, and if there is no load, the server time is not used at all and is not paid.

In the last few years, due to its novelty and limitations, FaaS has been used mainly for small projects, but today, thanks to the evolution of software, versatility, server power, tools are emerging to remove limitations, simplify and speed up cloud applications. This means that corporate business scenarios, in which cloud modernization was previously considered impossible, are now a reality.

Voice search and navigation are fast gaining popularity. Among the drivers of growth - the development of technology, almost all modern models of smartphones and some models of laptops have the option of voice search. The popularity of voice assistants and smart speakers also makes a great contribution [6].

More and more people are communicating with their devices by voice, and with the help of modern machine learning technologies, AI recognizes increasingly complex requests.

According to the Global Digital Overview study, the share of voice search worldwide has grown by 9% in 2020, with 43% of users aged 16 to 64 searching for information by voice at least once a month [6].

The growing popularity of voice search and navigation on the web page by voice is not surprising, because it solves problems quickly and is easy to use. At the same time, there is a growing demand for web-optimization solutions for voice search in terms of SEO and other technical strategies.

1.3 Analytical review of web application analogues

Before starting the project, it is necessary to analyze the existing web applications of a similar nature and subject. With this we will be able to draw conclusions about how to create your own web application based on the disadvantages and advantages of these resources.

Among the considered web applications were analyzed:

1. Comfy online store (Fig. 1.2, access: <https://comfy.ua/ua/>).
2. Online store "MOYO" (Fig. 1.3, access: <https://www.moyo.ua/>).
3. "ITbox" Online store of electronics and computer equipment (Fig. 1.4, access: <https://www.itbox.ua/>).
4. "Computer Universe" Online store of electronics and office equipment (Fig. 1.5, access: <https://kvshop.com.ua/>).

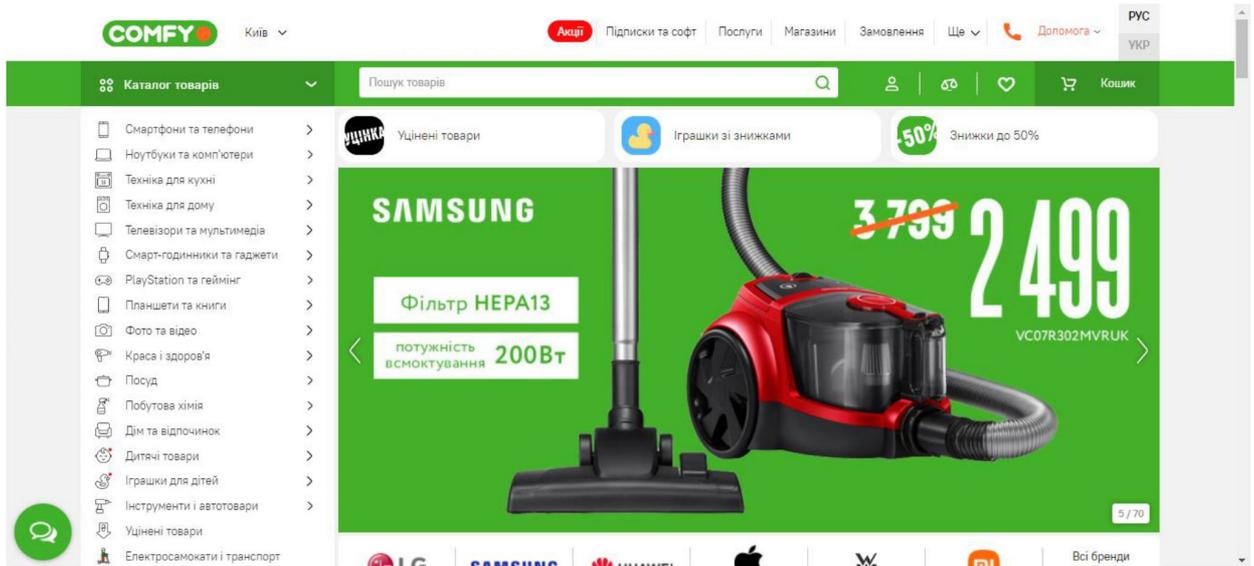


Figure 1.2 - Comfy web application home page

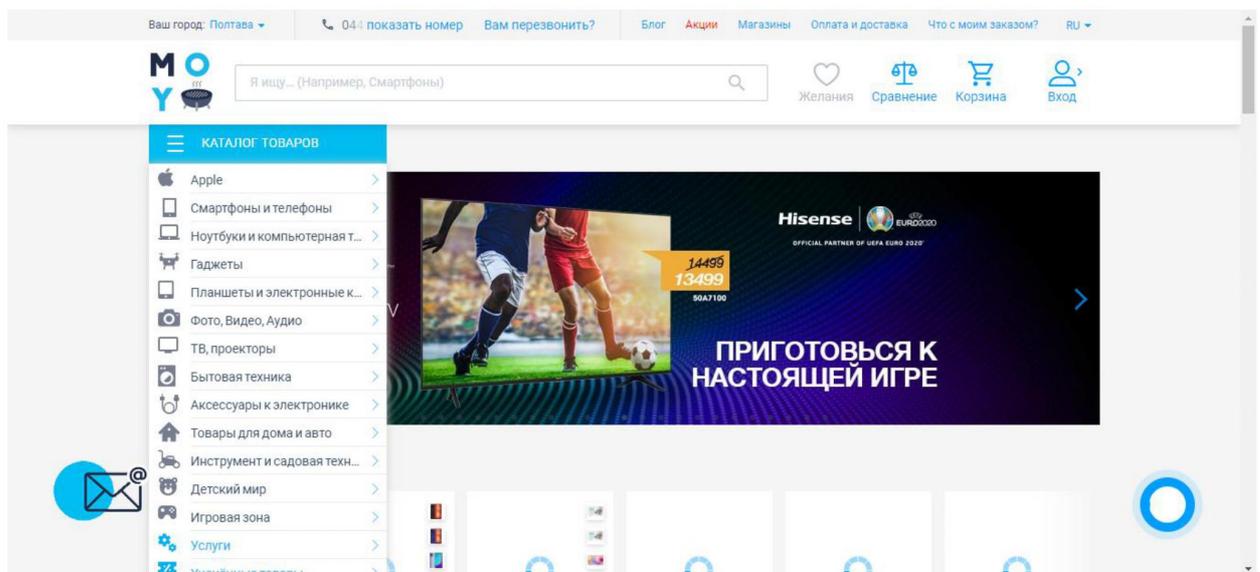


Figure 1.3 - Main page of the MOYO web application

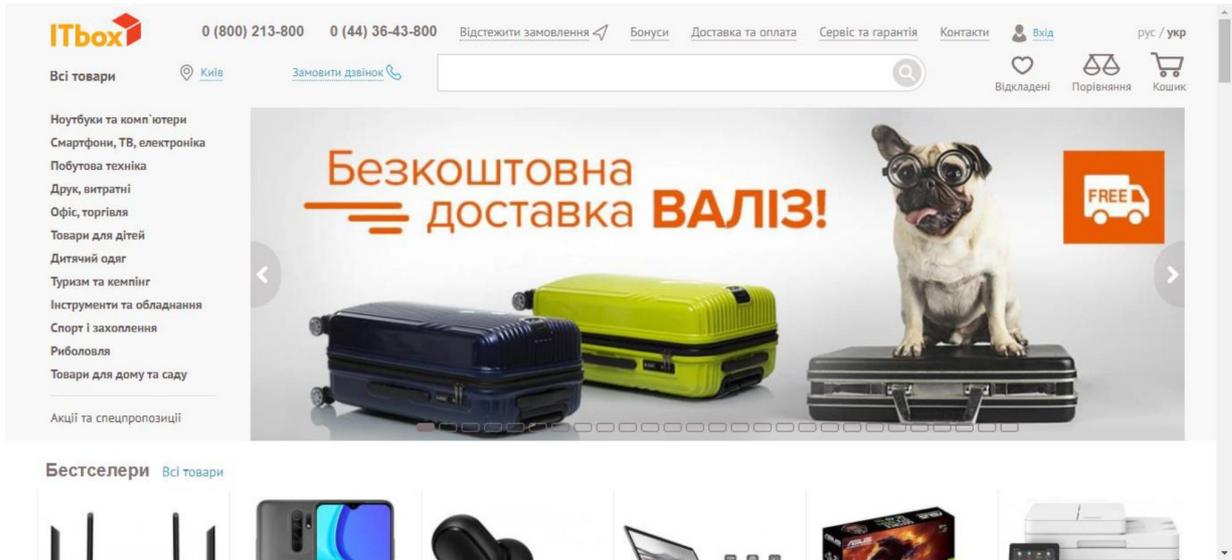


Figure 1.4 - Home page of the web application "ITbox"

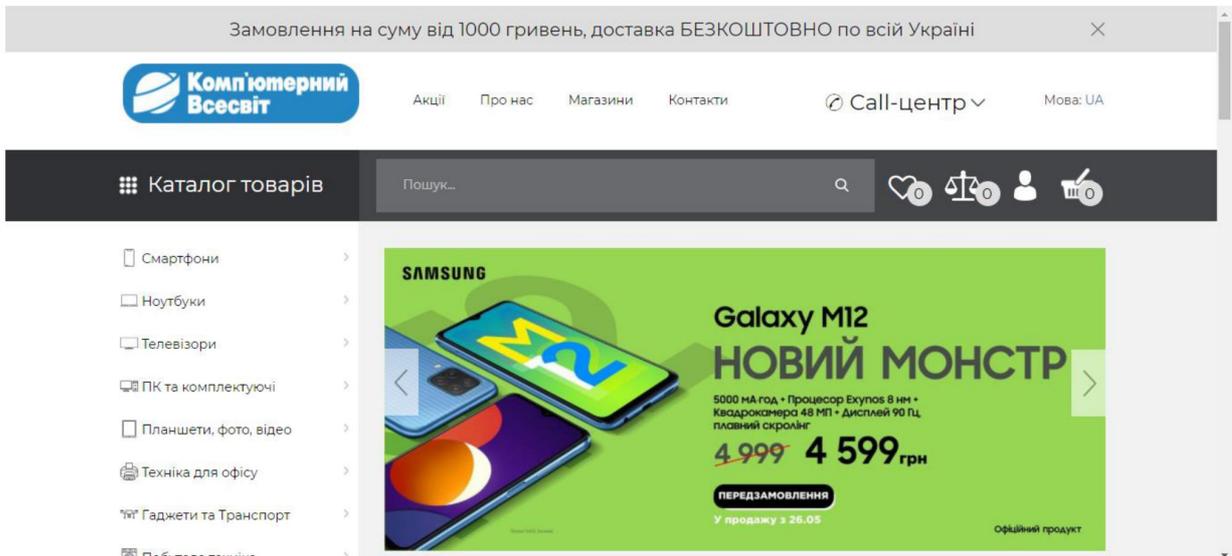


Figure 1.5 - The main page of the computer application "Computer Universe"

After analyzing the above web applications, we can conclude that the main purpose of development is to present the company on the Internet, selling goods online, customer support services. The web application should be designed to look as user-friendly, understandable, and user-friendly as possible.

1.4 An overview of modern technologies for creating web applications

Nowadays, web developers face many different tasks - from creating interactive entertainment sites to serious business projects that require increased reliability and security from unauthorized access. To implement them, you need to choose the right

tools - programming languages, frameworks, or content management systems (CMS - Content Management System).

Currently, there are many programming languages and their number is changing every day. The advantages of each of them can be manifested in the context of a particular task. The choice of programming language or framework depends on the IT department (programmers) of the company, which programming languages they know, and which are better to use in a project. It is necessary to distinguish between such concepts as programming language and framework [2].

A programming language is a basic syntax that allows applications to be created. Frameworks provide developers with different sets of libraries, which greatly simplifies the work. Some languages and frameworks are one, and server-side ones are for server-side programs. This distribution is shown in Figure 1.6.

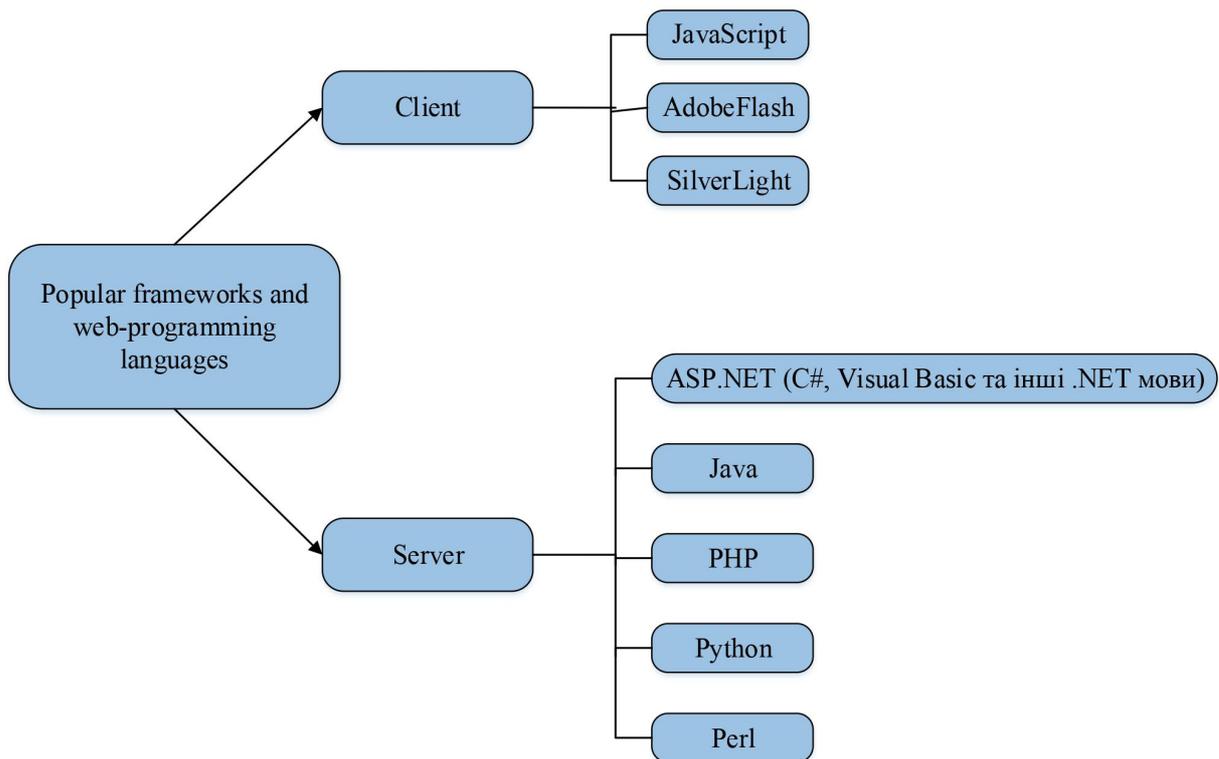


Figure 1.6 - Popular frameworks and web programming languages

Client programming languages are a type of processing language that is performed on the client-side. They allow you to change the content of the HTML page without reloading it from the server. Among the client languages of web programming

are JS, which, like HTML, is the basis of many web technologies, and the ability to use it is the basic knowledge of a web developer. Other popular languages, namely frameworks include XHTML, CSS, Silverlight, and any .NET languages.

Server web programming languages are executed on the server itself, for which special programs are used. Thus, their work is not affected by which browser is logged in, all computational processes are performed on a remote PC. Access to the program listing is blocked, which guarantees stable operation and security of the server. The user sees only part of the functionality that is displayed to the browser in the form of HTML pages. Server languages open more opportunities than client languages; using them, you can exchange information with the server, which is not possible in the case of client programming languages. Among the server-side web programming languages are PHP, Perl, Ruby, Java, C, Python, but the most common (more than 5 million servers) among them is the PHP language, which features full support for object-oriented programming.

Modern dynamic sites are sure to use a database both to store a variety of content and to implement their functions. To date, MySQL, PostgreSQL, ORACLE database management systems have become popular. These databases have simplified functionality, sufficient for web development tasks. The most common MySQL database. It is quite easy to use and has a high speed of request processing, which is important for a large number of user requests to the server on which the web application is hosted [3].

1.6 Conclusions to the first section

Web applications are a special type of program that is based on a client-server architecture.

Each web application consists of 3 parts: information, program code, and configuration information. The information part includes a description of the web page in HTML format. The program code implements the business logic, designed in the form of data processing procedures. This code is executed by the server and interacts with the dynamic elements of the information part, allows you to dynamically generate the content of the web page, which is transmitted to the user. Configuration information

contains parameters that determine how the web application runs on the server, security settings, response to errors that occur, etc.

Considering several modern web applications in terms of compliance with their objectives, as well as the correctness of the choice of tools for development, we can identify the following characteristic shortcomings:

- excess or lack of information provided;
- ease of use, the user within 10-15 seconds must understand how to use the application;
- fascination with animations, graphics, and videos that do not carry any useful information, which automatically entails a reduction in the number of potential users, as it increases the bandwidth requirements of the Internet connection and the amount of traffic;
- incorrect choice of tools for development entails an unreasonable increase in the cost of the finished application.

Summing up, we can say that modern web applications should be primarily user-friendly and user-oriented, multifunctional, and have a nice, unobtrusive design. More than half of users access the web application using smartphones and tablets, so adaptive design plays a very important role in building applications. Minimizing flash animations increases the download speed of the application.

ДЕМОНСТРАЦІЙНИЙ МАТЕРІАЛ

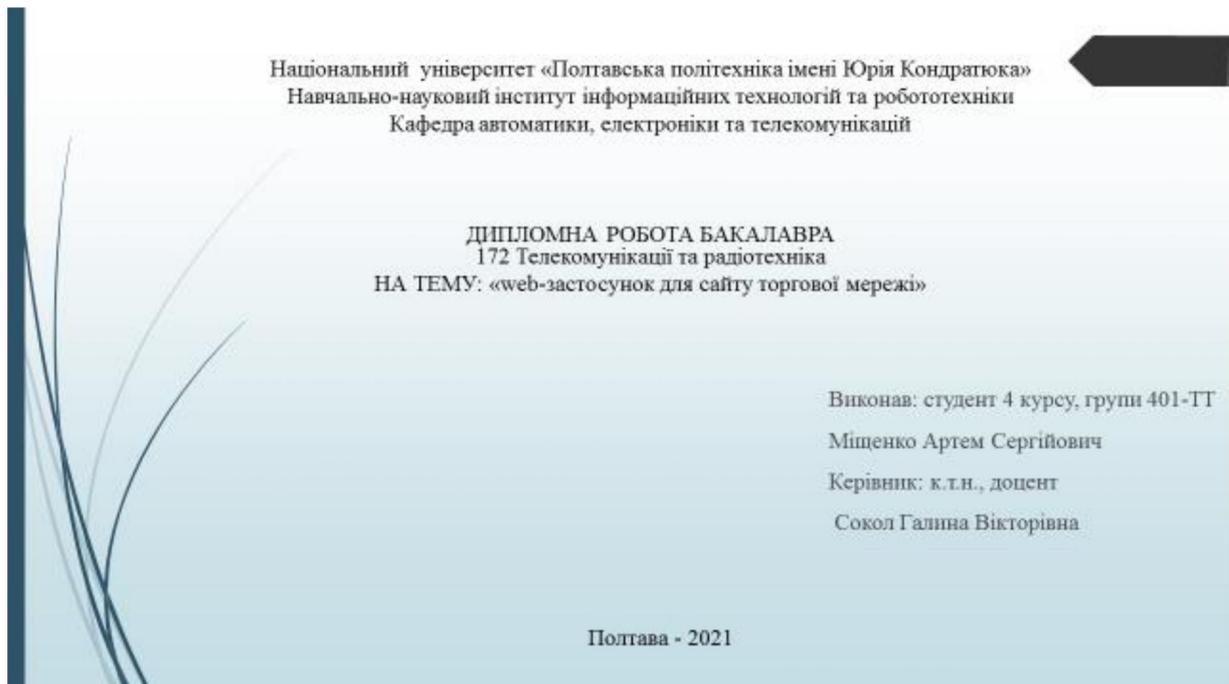


Рис. Б.1 – слайд № 1

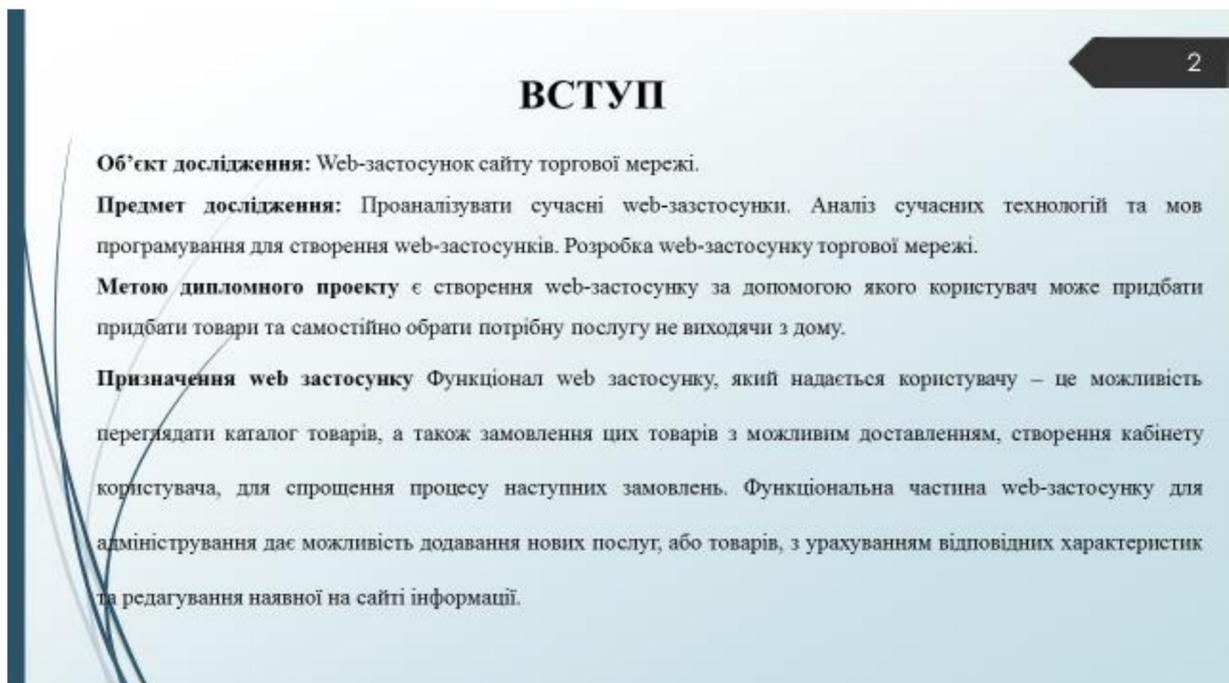


Рис. Б.2 – слайд № 2



Рис. Б.3 – слайд № 3

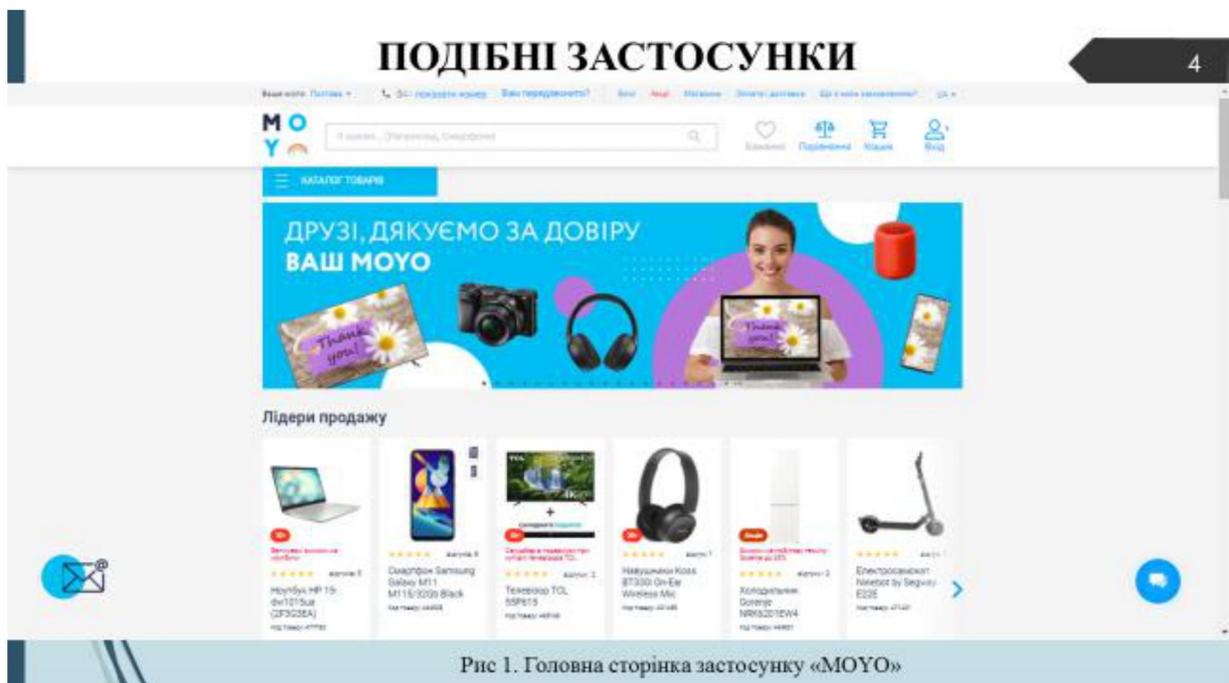


Рис. Б.4 – слайд № 4

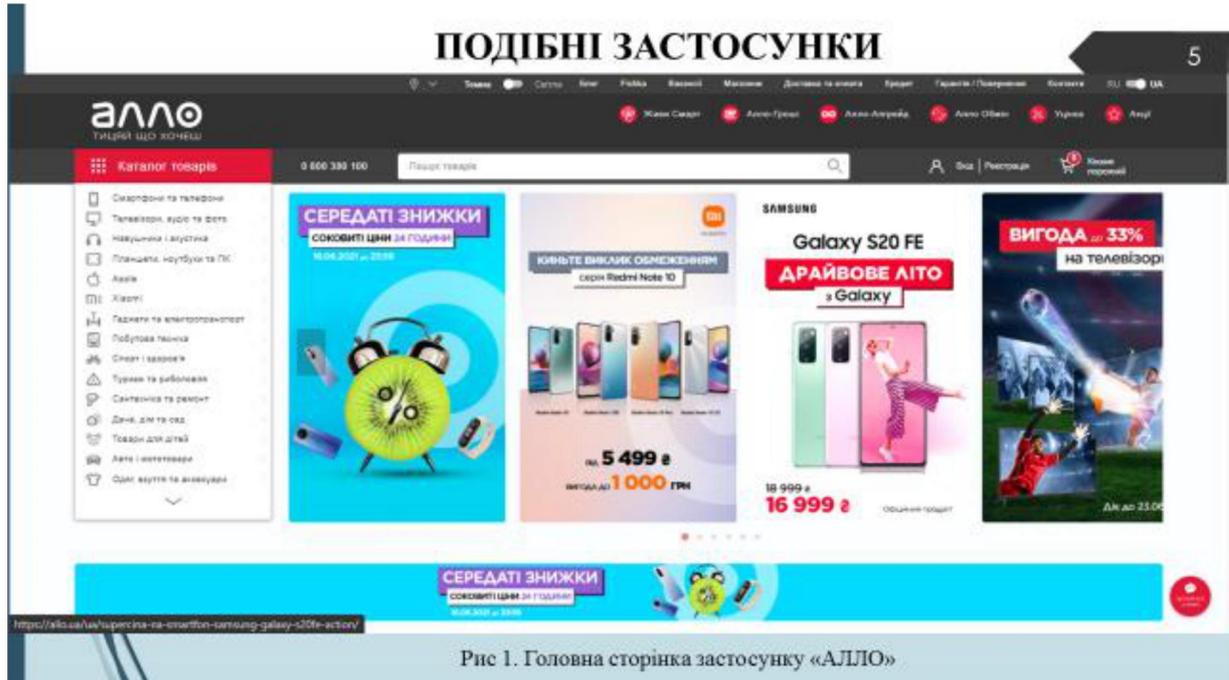


Рис. 1. Головна сторінка застосунку «АЛЛО»

Рис. Б.5 – слайд № 5



Рис. Б.6 – слайд № 6

ФУНКЦІОНАЛ АДМІНІСТРАТОРА

Функції адміністратора в панелі адміністраторів

Функція	Виконавець
ВИДАЛЕННЯ. Видалення користувачів із системи.	Адміністратор
РЕДАГУВАННЯ. Редагування даних користувачів.	Адміністратор
ПЕРЕГЛЯД. Перегляд всіх користувачів в системі.	Адміністратор
ВИДАЛЕННЯ. Видалення товарів із системи.	Адміністратор
РЕДАГУВАННЯ. Редагування товарів.	Адміністратор
ДОДАВАННЯ. Додавання товарів.	Адміністратор
ПЕРЕГЛЯД. Перегляд всіх новин в системі.	Адміністратор
ДОДАВАННЯ. Додавання новин в системі.	Адміністратор
ВИДАЛЕННЯ. Видалення новин в системі.	Адміністратор
ВИДАЛЕННЯ. Видалення категорій із системи.	Адміністратор
ДОДАВАННЯ. Додавання категорій	Адміністратор
ОБРОБКА. Обробка замовлень.	Адміністратор
ВИДАЛЕННЯ. Видалення відгуків.	Адміністратор
РЕДАГУВАННЯ. Редагування відгуків.	Адміністратор

Рис. Б.7 – слайд № 7

ФУНКЦІОНАЛ КОРИСТУВАЧА

Функції користувача у застосунку

Функція	Виконавець
РЕЄСТРАЦІЯ. Зареєструвати профіль.	Клієнт
АВТОРИЗАЦІЯ. Авторизуватися у застосунку	Клієнт
РЕДАГУВАННЯ. Редагування профілю.	Клієнт
ПЕРЕГЛЯД. Перегляд профілю.	Клієнт
ПЕРЕГЛЯД. Перегляд товарів	Клієнт
ПЕРЕГЛЯД. Перегляд категорій.	Клієнт
ДОДАВАННЯ. Додати товар до кошику замовлень.	Клієнт
ВИДАЛЕННЯ. Видалити товар з кошика замовлень.	Клієнт
ВІДГУК. Залишити відгук про товар.	Клієнт
ЗАМОВЛЕННЯ. Створити замовлення	Клієнт
ПЕРЕГЛЯД. Перегляд новин.	Клієнт

Рис. Б.8 – слайд № 8

АПРОБАЦІЇ ТА ПУБЛІКАЦІЇ

10

Результати апробовані в рамках 73-ї наукової конференції професорів, викладачів, наукових працівників, аспірантів та студентів:

«Prospects for the use of MIMO based on IEEE 802.11ac IN IoT technologies = Перспективи застосування MIMO на основі IEEE 802.11ac в технологіях IoT».

Результати, які мають відношення до теми дипломної роботи опубліковані в матеріалах тез конференції Том 1 (21 квітня – 13 травня 2021):

«Аналіз сучасних мов програмування для створення веб-застосунків. К.т.н., доц. Г.В. Сокол, студент гр.401-ТТ А.С. Міщенко.»

Рис. Б.9 – слайд № 9

ДЕМОНСТРАЦІЯ РОБОТИ WEB-ЗАТСОСУНКУ

9



Рис. Б.10 – слайд № 10

ВИСНОВОК

В процесі проектування дипломного проекту був розроблений web-застосунок торгової мережі з продажу комп'ютерної техніки.

В процесі виконання було розроблено web-застосунок який призначений для різних товарів для пролажу через мережу інтернет

Web-застосунок призначений для продажу різноманітних товарів через мережу Інтернет. Областю призначення даного застосунку є інтернет-торгівля.

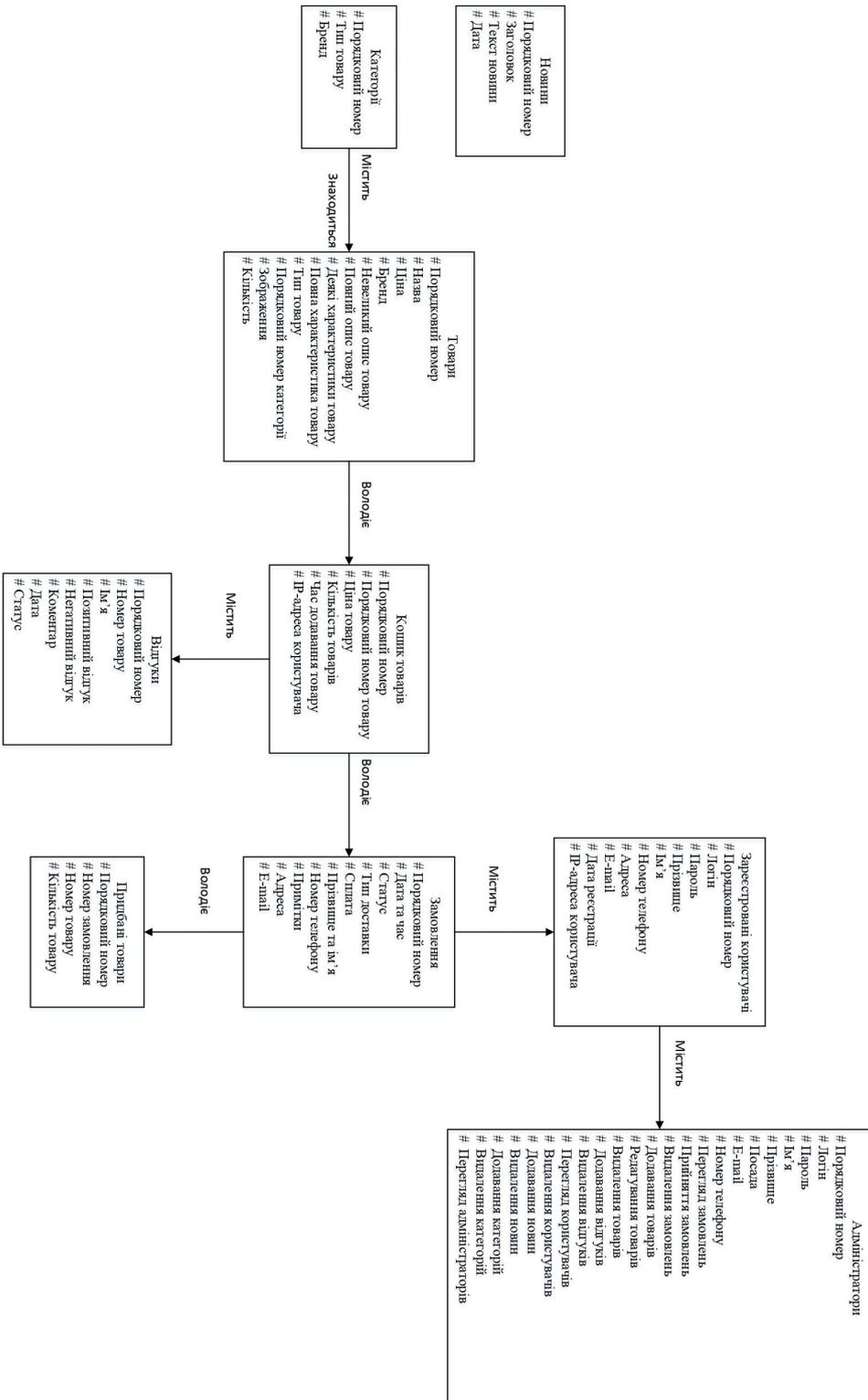
До числа основних відмінних рис можна віднести наступні:

- Невеликий розмір файлів з кодом web-сторінок, що забезпечує швидке завантаження з мережі на ПК користувача.
- Блок адміністраторів, для керування вмістом, за допомогою якого зручно працювати навіть звичайним користувачам.
- Гнучкість, відкритість і модифікація, за допомогою простих інструментів.

Рис. Б.10 – слайд № 10

ДОДАТОК В

ЛОГІЧНА МОДЕЛЬ БАЗИ ДАНИХ



АПРОБАЦІЇ ТА ПУБЛІКАЦІЇ

УДК 621.396

А.С. Мищенко, студент гр. 401ТТ,
Г.В. Сокол, к.т.н., доцент
Національний університет
«Полтавська політехніка імені Юрія Кондратюка»

АНАЛІЗ СУЧАСНИХ МОВ ПРОГРАМУВАННЯ ДЛЯ СТВОРЕННЯ ВЕБ-ЗАСТОСУВАНЬ

В сучасному світі для створення веб-застосунків використовується безліч різних мов програмування. Вибрати мову програмування потрібно враховуючи певні фактори та обставини. Кожна з мов має свої переваги та недоліки, вибір повинен залежати від поставленої задачі. Якщо це особистий проект, розробник обирає ту мову, яку знає краще.

Розглянемо найбільш популярні мови програмування, які використовуються для створення веб-застосунків:

- SQL.
- JavaScript.
- Python.
- PHP.

SQL – це основа багатьох СУБД (систем управління базами даних). Вважається важливою частиною програмування, перша і основна мова для роботи з базами даних, що набула широкого розповсюдження.

~ 15 ~

JavaScript – це мова сценаріїв, використовується в основному в інтернеті. Одна з найпопулярніших мов програмування серед розробників. JavaScript інтерпретована мова, таким чином їй не потрібно компілювати. Вона використовується як одна з основних технологій, що дозволяє створити інтерактивні веб-сайти.

Python – один з самих популярних та простих мов програмування і застосовується майже всюди, в тому числі і для розробки веб-застосунків. Для нього є багато фреймворків, частина з яких не просто полегшує розробку, а й дозволяє зрозуміти, як працює готовий сайт.

PHP – скриптова мова програмування, створена для генерації HTML-сторінок на стороні веб-сервера. На сьогоднішній день PHP є найпоширенішою мовою програмування для створення веб-застосунків. Вона вважається однією з базових мов програмування. Головним чинником мови є практичність. Мова PHP знайома програмістам, що працюють в різних областях. Ця мова дуже популярна і пов'язано це з тим, що її використовують на одній з найпоширеніших платформ для створення сайтів Wordpress і понад 80% найбільш відвідуваних сайтів у світі створені за допомогою PHP.

Література

1. SQL [WWW документ]. URL <https://uk.wikipedia.org/wiki/SOL>.
2. JavaScript [WWW документ]. URL https://developer.mozilla.org/uk/docs/Learn/Getting_started_with_the_web/JavaScript_basics.
3. Python [WWW документ]. URL <http://www.plugin.org.ua/documentation/about-python>.
4. PHP [WWW документ]. URL <https://uk.wikipedia.org/wiki/PHP>.

G. Sokol, P. Podhorneyi, A. Mishenko, R. Piskoviyi, D. Herasymov

National University "Yuri Kondratyuk Poltava Polytechnic", Poltava, Ukraine

PROSPECTS FOR THE USE OF MIMO BASED ON IEEE 802.11ac IN IoT TECHNOLOGIES

Abstract. Over time, the requirements on the network and the level of Internet traffic become more demanding. From standard cable connection with a speed of 100 Mbps to the use of Wi-Fi at speeds up to 2.3 Gbps per channel 160 MHz (802.11ac2) with simultaneous provision of multiple input mode - multiple output with multiple users (Multi User Multiple Input Multiple Output, MU-MIMO). In addition, the introduction of 5G mobile communication systems has significantly increased the level of mobile internet traffic. The development of the next generation of 6G with speeds from 100 Gbps to 1 Tbit / s has already begun in the world when controlling the artificial intelligence network, which is planned for 2026-30. In its perspective, the expansion of the range of services based on cloud computing has given a significant impetus to the development of new areas of Internet use. As a result, various concepts based on the Internet of Things (IoT) are being rapidly implemented. As you know, this approach allows you to connect to the Internet any object equipped with sensors or sensors that can receive, transmit and process information, as well as apply different actions depending on the information received. The quality of examples can include "smart home", "smart city" and others. Implementation of communication between objects of opportunities via the Internet using Bluetooth, Wi-Fi, ZigBee, etc. Another, quite new direction is the concept of the Internet of BioNanoThings (IoBNT). As you know, a living cell is already a kind of nanomachine that is controlled by a genetic program. The cell's nucleus is the processor and memory, the mitochondria are the batteries, and the receptor proteins on the cell's surface are sensors that can receive signals, and their slit contacts with other cells can send signals. It only remains to reprogram them for your needs. At the same time, it is necessary to organize communication not only between the cells themselves, but also with the Internet. The cell can transmit bitwise information depending on the state, but at a very low speed. One of the options for implementing communications may be MIMO.

Keywords: Internet, Internet of Things, Internet of Bio-Nano things, MIMO, Wi-Fi, QAM.

Перспективи застосування MIMO на основі IEEE 802.11ac в технологіях ІОТ

Г. В. Сокол, П. А. Подгорний, А. С. Міщенко, Р. В. Пісковий, Д. О. Герасимов

Анотація. Вимоги до мережі рівень трафіку Інтернет стають все більш вибагливі. Від стандартного підключення кабелем зі швидкістю 100 Мбіт/с до використання Wi-Fi зі швидкістю до 2,3 Гбіт/с на канал 160 МГц (802.11ac2) з одночасним забезпеченням режиму множинного входу – множинного виходу з кількох користувачами (Multi User Multiple Input Multiple Output, MU-MIMO). Крім того, впровадження систем мобільного зв'язку 5G дозволило значно підвищити рівень трафіку мобільного Інтернету. В світі вже почалися розробки наступного покоління 6G зі швидкостями від 100 Гбіт/с до 1 Тбіт/с при управлінні мережі штучним інтелектом, яке планують на 2026-30 рр. В свою чергу, розширення номенклатури сервісів на базі хмарних обчислень дало значний поштовх до розвитку нових напрямків застосування Інтернету. Як наслідок, швидко впроваджуються різноманітні концепції на основі Інтернету речей (Internet of Things, IoT). Як відомо, такий підхід надає можливість підключення до Інтернету будь-якого об'єкту, оснащеного датчиками або сенсорами, що можуть отримувати, передавати та обробляти інформацію, і виконувати певні дії в залежності від отриманої інформації. В якості прикладів можна вказати «розумний будинок», «розумне місто», та ін. Реалізація зв'язку між об'єктами можлива через Інтернет з використанням технологій Bluetooth, Wi-Fi, ZigBee та ін. Ще одним, досить новим напрямком є концепція Інтернету біонаноречей (Internet of BioNanoThings, IoBNT). Як відомо, жива клітина – вже є своєрідною наномашинною яка керується генетичною програмою. Процесором і пам'яттю виступає ядро клітини, мітохондрії служать батарейкою а білки-рецептори на поверхні цієї клітини – це сенсори, які можуть сприймати сигнали, а їх щільні контакти з іншими клітинами, можуть відправляти сигнали. Залишається лише перепрограмувати їх для свої потреб. При цьому, потрібно ще організувати зв'язок не тільки між самими клітинами, але і з мережею Інтернет. Клітина може передавати побігову інформацію в залежності від стану, але з дуже низькою швидкістю. Одним з варіантів реалізації комунікацій може стати MIMO.

Ключові слова: Інтернет, Інтернет речей, Інтернет біонано речей, MIMO, Wi-Fi, QAM