

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра інженерії програмного забезпечення

Пояснювальна записка

до бакалаврської роботи
на ступінь вищої освіти бакалавр

на тему: **«РОЗРОБКА СЕРВЕРНОЇ ТА КЛІЄНТСЬКОЇ ЧАСТИНИ
ІНТЕРНЕТ- МАГАЗИНУ МОВОЮ JAVA SCRIPT»**

Виконав: студент 4 курсу, групи ПД–42
спеціальності
121 Інженерія програмного забезпечення
(шифр і назва спеціальності/спеціалізації)

Шлямар М.І.
(прізвище та ініціали)

Керівник _____ Трінтіна Н.А.
(прізвище та ініціали)

Рецензент

(прізвище та ініціали)

Київ –2022

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти -«Бакалавр»

Спеціальність підготовки – 121 «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

Негоденко О.В.

“ ___ ” _____ 2022 року

ЗАВДАННЯ НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТА

ШЛЯМАРА МАКСИМА ІГОРОВИЧА

(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка серверної та клієнтської частини інтернет-магазину мовою java script»

Керівник роботи: Трінтіна Н.А., к.т.н., доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом вищого навчального закладу від «18» лютого 2022 року №__.

2. Строк подання студентом роботи «3» червня 2022 року

3. Вхідні дані до роботи

Методи обробки зображень;

Науково-технічна література з питань, пов'язаних з програмним забезпеченням щодо розробки сервісу інтернет-магазину

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

4.1 Сервіси інтернет-магазинів.

4.2 Вимоги та оцінка якості системи.

4.3 Опис проектування системи.

4.4 Опис використаних технологій.

5. Перелік демонстраційного матеріалу (назва основних слайдів)

1. Мета, об'єкт та предмет дослідження
 2. Аналоги
 3. Порівняння З Аналогами
 4. Технічні Завдання
 5. Програмні Засоби Реалізації
 6. Приклади Роботи
 7. Апробація Результатів Дослідження
 8. Висновки
6. Дата видачі завдання «11» квітня 2022

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	11.04.22	Виконано
2	Вимоги до системи	20.04.22	Виконано
3	Створення та навчання моделі для вилучення полів	28.04.22	Виконано
4	Створення та навчання моделі для вилучення таблиць	8.05.22	Виконано
5	Концепція та архітектура програмного забезпечення	15.05.22	Виконано
6	Вступ, висновки, реферат	21.05.22	Виконано
7	Розробка обов'язкових демонстраційних матеріалів	28.05.22	Виконано
8	Попередній захист роботи	3.06.22	
9	Здача роботи	3.06.22	

Студент _____
(підпис) (прізвище та ініціали)

Керівник роботи _____
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Текстова частина бакалаврської РЕФЕРАТ

Пояснювальна записка до дипломної роботи «Розробка серверної та клієнтської частини інтернет-магазину мовою JavaScript»: 60 с., 21 рис., 8 літературних джерел.

Об'єкт дослідження: веб-розробка.

Предмет дослідження: розробка інтернет-магазину для продажу цифрової техніки на основі WEB.

Мета роботи – Створення і розробка інтернет-магазину для оренди товарів у вигляді WEB- додатку.

Методи дослідження: вибір найефективніших засобів для створення WEB-додатків шляхом порівняльного аналізу.

У роботі проведено аналіз існуючих веб-сервісів, таких як Rozetka і Bodo.

Загальною проблемою цих продуктів є необхідність у постійному з'єднанні з мережею інтернет, що може бути іноді недоступним.

Особливістю методу є можливість побачити дані з бази даних за допомогою серверу та клієнтської частини. Описано архітектуру та основні принципи розробки. За основу було взято технологію NodeJS з серверної сторони та JavaScript з клієнтської сторони для основного функціоналу.

Додаток працює з вже існуючими базами даних. В якості серверу баз даних було взято MongoDB.

Отже, розроблено та описано веб-додаток, завданням якого є сервіс з оренди товарів в даному випадку костюмів.

У якості вихідних даних є відображення інформації на стороні клієнта.

Даний додаток може бути використано у всіх сферах, які потребують веб-додатку з продажу або оренди.

Галузь використання – оренда товарів.

ЗМІСТ

ВСТУП.....	9
1.1. Світовий ринок інтернет-магазинів.	11
1.2. Українська індустрія інтернет-магазинів	12
Висновки до розділу 1.....	15
2.ВИБІР, ОБГРУНТУВАННЯ ТА ОПИС ЗАСОБІВ РЕАЛІЗАЦІЇ.....	16
2.1. Етапи створення та структура сайтів	16
2.2. JAVASCRIPT. Різні версії та ES6.....	19
2.3. Node.js	25
2.4. Express	27
2.5. MongoDB та Mongoose.....	29
Висновки до розділу 2.....	31
3.РОЗРОБКА ІНТЕРНЕТ-МАГАЗИНУ.....	32
3.1. Вступ	32
3.2.NPM, Node.js, MongoDB, Express and EJS	34
3.3.Додаткові пакети Node.....	38
3.4.Програмна логіка та безпека	41
3.5 Демонстрація прототипу інтернет-магазину цифрової техніки	51
Висновки до розділу 3.....	55
ВИСНОВКИ	56
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	57

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

API (англ. application programming interface) - програмний інтерфейс програми, інтерфейс прикладного програмування;

CSRF (англ. Cross-Site Request Forgery, CSRF) - міжсайтова підробка запиту;

CORS (англ. Cross-Origin Resource Sharing) - спільне використання ресурсів з різних джерел;

CSS (англ. Cascading Style Sheets) - каскадні таблиці стилів;

EJS (англ. Embedded JavaScript) - каркас веб-додатків для Node.js;

ES6 (англ. ECMAScript 6) - вбудована розширювана мова програмування яка не має засобів введення-виведення для побудови інших скриптових мов;

HTML (англ. Hyper Text Markup Language) – мова розмітки гіпертексту;

HTTP (англ. Hyper Text Transfer Protocol) - протокол передачі даних, що використовується в комп'ютерних мережах

HTTPS (англ. Hyper Text Transfer Protocol over Secure Socket Layer) - схема URI, що синтаксично ідентична http: схемі, яка зазвичай використовується для доступу до ресурсів Інтернет;

IDE (англ. Integrated Development Environment) - паралельний інтерфейс підключення накопичувачів (жорстких дисків і оптичних приводів) до комп'ютера;

JSON (англ. JavaScript Object Notation) - текстовий формат обміну даними, заснований на JavaScript;

LAMP (англ. Linux Apache MySQL and PHP) - набір відкритого програмного забезпечення, який використовується для створення веб-серверів;

LTS (англ. Long Term Support) - частина життєвого циклу програмного забезпечення (ПЗ);

MVC (англ. Model View Controller) - шаблон програмування, який дозволяє розділити логіку програми на три частини;

MVP (англ. Minimum Viable Product) - мінімально життєздатний продукт;

ВСТУП

Веб-розробка пройшла довгий шлях з моменту створення Інтернету, глобальної комп'ютерної мережі. Для зручності використання розробники повинні розробляти окремі програми для кожної операційної системи та встановлювати їх на локальному комп'ютері. Ця програма називається настільною програмою. З іншого боку, веб-додатки можна легко отримати з веб-браузера, незалежно від операційної системи користувача. Веб-програми пройшли довгий шлях після того, як настільні програми та статичні веб-сайти перейшли на новітні інтерактивні та розумні веб-програми. Користувачі могли читати вміст лише на статичних веб-сайтах. Таким чином, з появою веб-додатків користувачі можуть спілкуватися з сервером, слухати аудіо та дивитися відео та малювати на екрані.

Розробка інтернет-магазину являє собою суміш фронт-енд і бек-енд розробки. Існує кілька мов програмування та методів створення веб-додатків. Спочатку не було мови програмування, яка могла б виконувати розробку додатків фронт-енд і бек-енд.. PHP в основному використовується для бекенда, HTML, CSS і JavaScript є основними стеками для розробки інтерфейсу, а MySQL використовується для бази даних.

Apache діє як веб-сервер в операційній системі Linux у стеку LAMP. Однак, щоб розробляти веб-додатки з використанням стеку LAMP, розробникам необхідно вивчити кілька мов, кожна з яких використовує різні синтаксиси і володіє різними властивостями. Крім того, з появою Node.js як серверної платформи, JavaScript може забезпечити повноцінну розробку програмного забезпечення на одній мові програмування. Хоча Node.js є найбільш часто використовуваною технологією для розробки серверних систем, існує також багато фреймворків, таких як Angular, React, Vue.js або Knockout.js, а також механізми шаблонів, такі як EJS, Handlebars або Pug.

Метою цієї роботи є дослідження різних аспектів веб-розробки з використанням повного стека JavaScript та розробка на його основі прототипу програми. Мета полягає в тому, щоб вивчити використання Node.js, Express,

MongoDB та EJS у розробці повного стека веб-додатків. Крім того, він зосереджується на вивченні різних версій JavaScript і використанні їх на платформі Node.js.

Так, за допомогою Node.js і Express був розроблений інтернет-магазин цифрових технологій з використанням базових пакетів і модулів Node.js і проміжного програмного забезпечення Express. Інше стороннє проміжне програмне забезпечення також використовується для розробки робочих прототипів додатків.

Повнотекстовий JavaScript вважається найкращою технологією для розробки сучасних, масштабованих і безпечних веб-додатків. Однак Node.js не може обробляти складні обчислення даних і алгоритми, що робить його непридатним для розробки великих корпоративних програм.

1. РОЗВИТОК ІНТЕРНЕТ-МАГАЗИНУ В УКРАЇНІ ТА СВІТІ

1.1. Світовий ринок інтернет-магазинів.

Доходи інтернет-ринку ростуть у всіх регіонах світу і ростуть найшвидше - У Східній Азії. Ця частина регіону приносить понад 50% доходу, при цьому Китай перевершує США за обсягами продажів електронної комерції.

Лідери світового ринку - компанії з Китаю, США, Великобританії. Найбільші компанії заробляють мільярди доларів на своїх веб-сайтах. До кінця 2020 року загальний обсяг глобальних продажів інтернет-магазину досяг 2 трлн доларів США. (рис. 1.1)

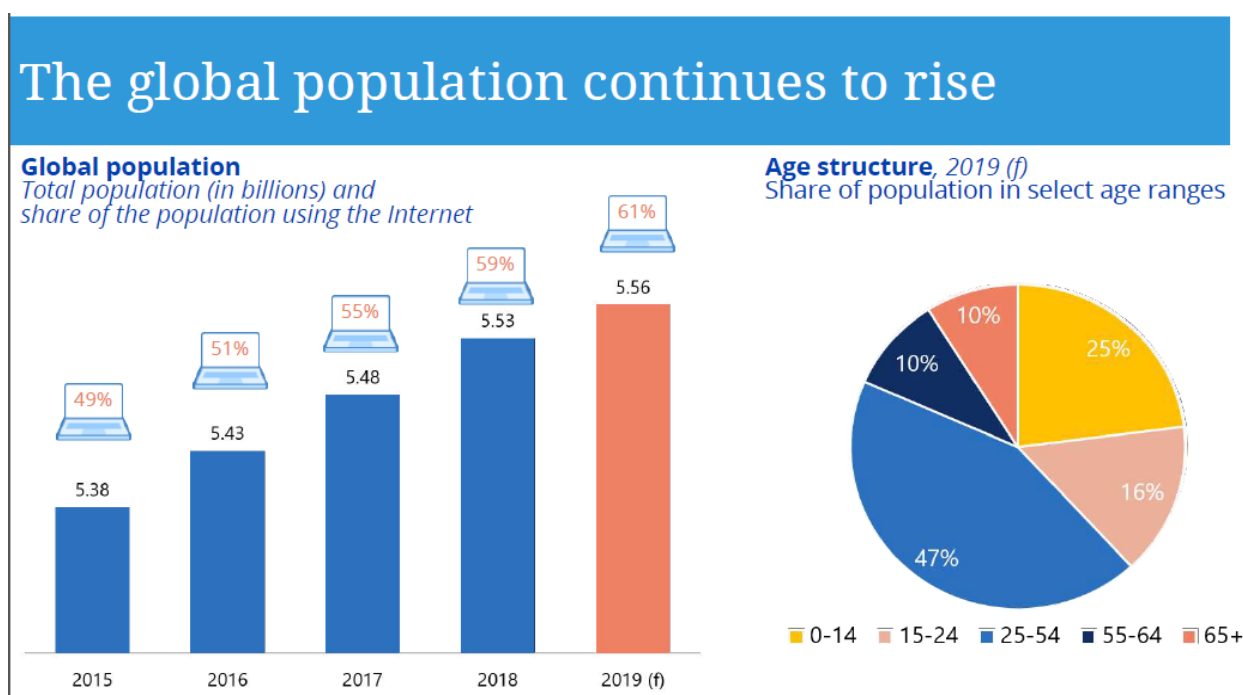


Рис. 1.1. Діаграма розвитку інтернет-торгівлі

У великій ніші ринку найбільший середній чек – у ніші комп’ютерів, телевізорів та мультимедійних пристроїв тут становить 212 доларів. Найнижча середня сума покупки - в ніші ринку квітів і подарунків.

Якщо ви подивитеся на рейтинги продажів роздрібних мереж по всьому світу, ви легко помітите, що найбільший прибуток отримують компанії, які працюють і займаються електронною комерцією. (рис 1.2)

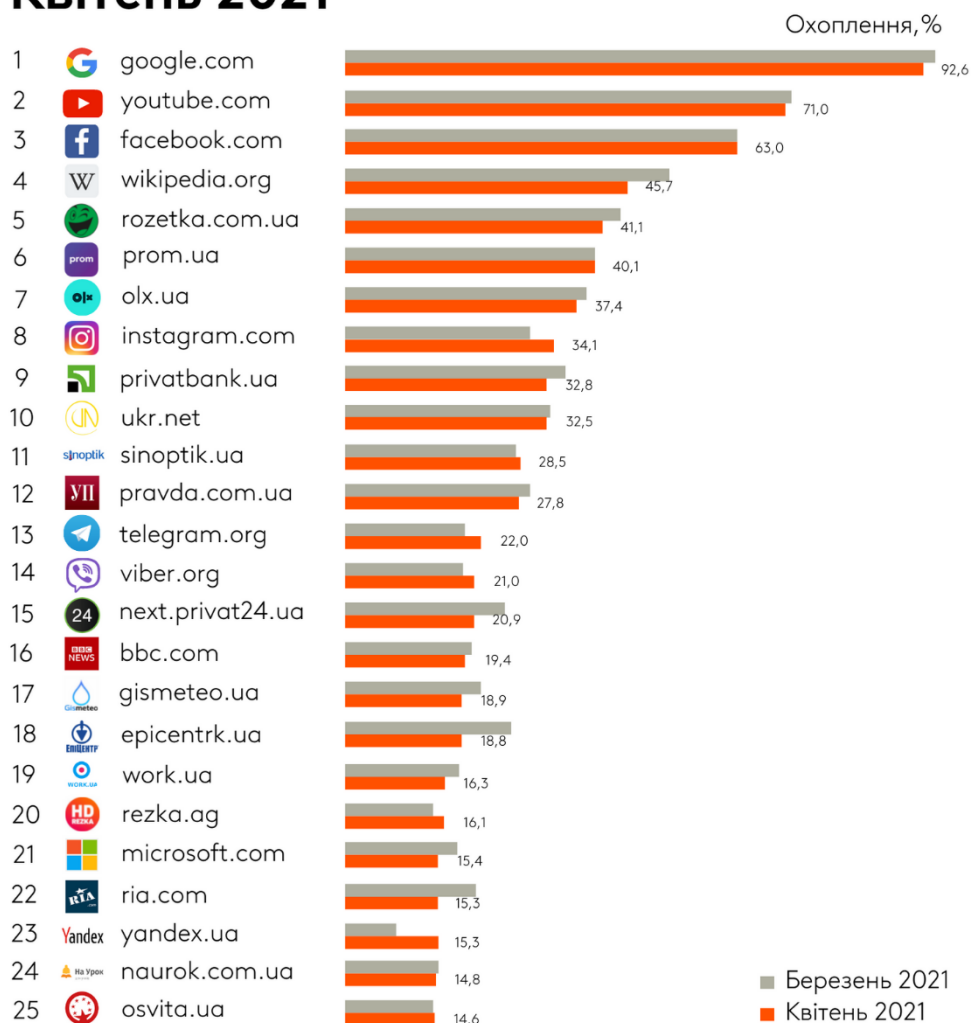
Топ-10 торгових мереж за темпами зростання виручки			
Місце	Бренд	Країна	Зростання обороту за 2014-19 роки, %
1	Alibaba 	Китай	44
2	JD.com 	Китай	35
3	Albertsons Companies 	США	25
4	Amazon.com 	США	22
5	Apple 	США	13
6	China Resources Enterprise 	Китай	13
7	X5 Retail Group 	Росія	10
8	Suning 	Китай	9
9	A.S. Watson 	Гонконг	8
10	Ikea 	Швеція	7

Рис. 1.2. Рейтинг торгових мереж

1.2. Українська індустрія інтернет-магазинів

Зараз сфера інтернет-магазинів дуже стрімко розвивається. Візьмемо наприклад інтернет-магазин Розетка, на якому можна купувати товари онлайн. За статистикою на квітень 2021 року сервіс займає п’яту позицію за популярністю в списку українських сайтів.(рис. 1.3)

ТОП-25 сайтів України Квітень 2021



KANTAR

Дані Kantar CMeter, desktop і mobile інтернет-користувачі у віці 14-70 років, n = 5000
 Дослідження CMeter об'єднує дані з 3-х джерел: Site-centric, Frame-centric, User-centric. Об'єднання даних відбувається в режимі реального часу.

Рис.1.3. Популярність сайтів в Україні

Але, на жаль, через тенденцію розвитку та різноманітні показники існуючих гігантів, наприклад, відсутність адаптивного дизайну, наразі немає жорсткої конкуренції, що приносить незручності, і навіть деякі компанії не мають власних інтернет-магазинів. Як ми бачили, в Інтернеті немає прибутку через побоювання та невизначеність щодо надійності таких транзакцій. Такий висновок можна зробити, аналізуючи відвідуваність українських інтернет-магазинів. (рис.1.4)

Багато компаній або не створюють свій власний інтернет-магазин (чи ні), тому що вони думають, що ринок уже переповнений, або вони створюють свою власну платформу, але конкурують із існуючими сайтами за рекламу.

Кількість відвідувань в місяць

	Кількість відвідувань, млн	% відвідувань з України
1. rozetka.com.ua	42.41	93.54
2. epicentrk.ua	7.76	95.14
3. makeup.com.ua	6.94	92.41
4. allo.ua	6.38	92.84
5. apteka911.com.ua	5.09	88.33
6. add.ua	4.97	92.11
7. comfy.ua	4.65	91.62
8. foxtrot.com.ua	4.33	95.26
9. citrus.ua	3.16	90.6
10. agro-market.net	3.02	94.72
11. moyo.ua	2.86	67.79
12. f.ua	2.61	70.26
13. eldorado.ua	2.1	93.99
14. eva.ua	1.85	96.45
15. intertop.ua	1.78	96.09
16. kasta.ua	1.69	96.8
17. yakaboo.ua	1.61	81.47
18. dok.dbroker.com.ua	1.46	87.31
19. parfums.ua	1.42	93.77
20. stylus.ua	1.22	90.83

Рис.1.4. Популярність українських інтернет-магазинів

Висновки до розділу 1

Підводячи підсумок, можна зробити висновок, що інтернет-магазин є потужним інструментом торгівлі. Безпека та здоров'я дуже важливі, особливо в умовах нинішньої пандемії коронавірусу. Інтернет-магазини є ідеальним засобом для підприємців, які хочуть покращити та пришвидшити свої послуги, отримувати більше прибутку та піклуватися про своїх клієнтів. Український ринок є гарним початком для розвитку, оскільки є місце для створення нових веб-сайтів без багатьох платформ.

Щоб задовольнити ці вимоги, не потрібно створювати велику компанію з сотнями співробітників. Досить менеджерів, кілька дизайнерів, кілька розробників і інвестор. Це оптимізує багато бізнес-процесів і забезпечує зростання доходів.

2. ВИБІР, ОБГРУНТУВАННЯ ТА ОПИС ЗАСОБІВ РЕАЛІЗАЦІЇ

2.1. Етапи створення та структура сайтів

Перш ніж розпочати створення цифрового інтернет-магазину, вам необхідно зрозуміти категоризацію веб-сайту та структуру HTML-сторінок. Після цього ви зосередьтесь на основних етапах створення веб-сайту, поглибите свої знання мови розробки JavaScript за вашим вибором та навчитесь основ використання Node.js.



Рис. 2.1. Класифікація веб-сайтів.

Першим кроком у створенні веб-сайту є проектування. Було обговорено мету та бюджет проекту, з якого сформовано технічні завдання. Наступний етап розробки — дизайн. Результатом є дизайн-макет вашого сайту: графічний файл, створений однією з графічних програм.

Також не забувайте про маркування сторінок. Створюйте шаблонні макети сторінок, написані на HTML, на основі макетів.

Далі потрібно зробити сервіс. Наприклад, якщо у вас є форма реєстрації клієнта.

Останнім кроком є публікація та наповнення сервісу інформацією. Створіть рамку та введіть текстову інформацію.

Головні теги, які застосовуються для створення сайту:

<!DOCTYPE > – це інструкція для веб-браузера про те, у якій версії HTML написана сторінка. Це гарантує, що веб-сторінка аналізується однаково різними веб-браузерами.

< html > - тег `<html>` визначає початок HTML-файлу, всередині нього зберігається заголовок (`<head>`) та тіло документа (`<body>`).

<header> - в HTML використовується для визначення заголовка документа або розділу, оскільки він містить інформацію, пов'язану з заголовком і заголовком пов'язаного вмісту. Елемент `<header>` зазвичай має містити заголовок розділу (елемент `h1-h6` або елемент `<hgroup>`), але це не обов'язково. Його також можна використовувати для обгортання змісту розділу, форми пошуку або будь-яких відповідних логотипів. Тег `<header>` є новим тегом у HTML5 і є тегом-контейнером, тобто він містить початковий тег, вміст і кінцевий тег. В одному документі може бути кілька елементів `<header>`. Цей тег не можна помістити в `<footer>`, `<address>` або інший елемент `<header>`.

<meta> – тег `<meta>` визначає метатеги, які використовуються для зберігання інформації, призначеної для браузерів та пошукових систем. Наприклад, механізми пошукових систем звертаються до метатегів для отримання опису сайту, ключових слів та інших даних. Дозволяється використовувати більш ніж один метатег, вони розміщуються в контейнері `<head>`. Як правило, атрибути будь-якого метатегу зводяться до пар "ім'я=значення", які визначаються ключовими словами `content`, `name` або `http-equiv`.

<title> - тег `<title>` визначає заголовок веб-сторінки, це один з важливих елементів, призначений для вирішення безлічі завдань. В операційній системі Windows текст заголовка відображається у верхньому лівому куті вікна браузера.

<body> - тіло документа `<body>` призначене для розміщення тегів та змістовної частини веб-сторінки.

<div> – Тег `<div>` визначає поділ або розділ у HTML-документі. Тег `<div>`

використовується як контейнер для елементів HTML, який потім оформляється за допомогою CSS або маніпулюється за допомогою JavaScript. Тег `<div>` легко стилізувати за допомогою атрибута `class` або `id`. У тег `<div>` можна помістити будь-який вміст.

`<p>` – Тег `<p>` визначає абзац (параграф) тексту. Якщо тега, що закриває, немає, вважається, що кінець абзацу збігається з початком наступного блокового елемента.

`<h1>`-`<h6>` – HTML пропонує шість текстових заголовків різного рівня, які показують відносну важливість розділу, розташованого після заголовка. Так, тег `<h1>` є найбільш важливим заголовком першого рівня, а тег `<h6>` служить для позначення заголовка шостого рівня і є найменш значним

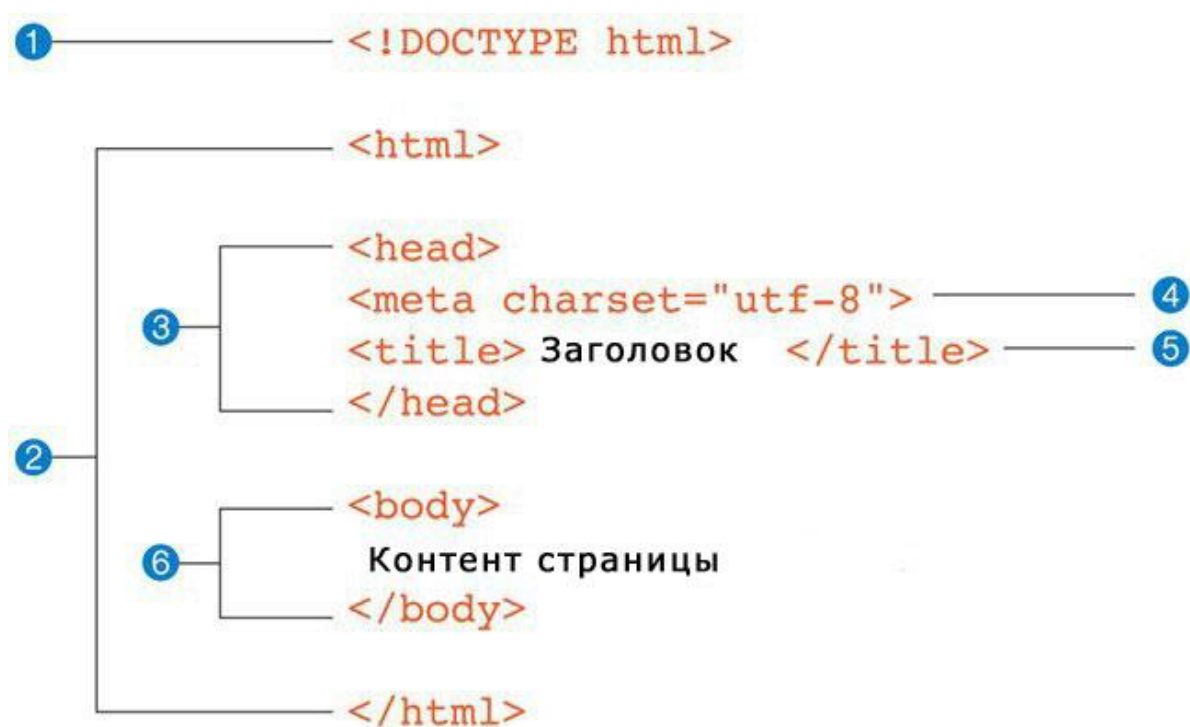


Рис. 2.2. Структура сторінки

2.2. JAVASCRIPT. Різні версії та ES6

JavaScript (JS) — це об'єктно-орієнтована мова програмування, заснована на концепції прототипів . В основному він використовується для впровадження деяких функцій у веб-сторінки, код JavaScript з цих сторінок запускається браузером. Ця мова добре відома тим, що вона використовується для створення веб-сайтів, але вона також використовується для доступу до вбудованих об'єктів в інших програмах. Спочатку він був розроблений Бренданом Ейхом з Netscape Communications Corporation як Mocha, потім LiveScript, і зрештою він отримав назву JavaScript.

Також це легка мова програмування, яку веб-розробники зазвичай використовують для створення більш динамічних взаємодій під час розробки веб-сторінок, програм, серверів і навіть ігор. Розробники зазвичай використовують JavaScript поряд з HTML і CSS. Мова сценаріїв добре працює з CSS у формативанні елементів HTML. Однак JavaScript все ще підтримує взаємодію з користувачем, чого CSS не може зробити сам. Реалізація JavaScript в Інтернеті, мобільних додатках та розробці ігор робить мову сценаріїв гідною вивчення. Ви можете зробити це за допомогою навчальних платформ, таких як BitDegree, або досліджуючи безкоштовні шаблони та програми JavaScript на платформах для розміщення коду, як-от GitHub.

У JavaScript є різні програми, з якими, ймовірно, стикався кожен, хто відвідував інтерактивні веб-сторінки або мобільні додатки. Хоча розробка веб-сайтів, включаючи додавання таких елементів, як інтерактивні форми та анімація, є найбільш традиційним використанням JavaScript, вона також знаходить застосування у:

- Ігри на основі веб-браузера, включаючи 2D та 3D ігри
- Розробка мобільних додатків — вигідна, оскільки не залежить від телефонної платформи
- Презентації — створення веб-анімованих слайдів Хоча більшість програм JavaScript є клієнтськими, JavaScript також корисний у додатках на стороні

сервера, таких як створення веб-серверів.

Незважаючи на схожість назви та синтаксису, зв'язку між JavaScript та Java немає. Як і Java, JavaScript має синтаксис, близький до синтаксису мови C, але він має більше спільного з мовою Self, ніж з Java.

Зміна назви з LiveScript на JavaScript була здійснена приблизно в той же час, коли Netscape включив підтримку технології Java у веб-браузер Netscape Navigator. Microsoft реалізувала JavaScript як JScript з низкою змін і розширень у реалізації Netscape. На платформі Windows JScript є однією з виконуваних мов Windows Script, тому його можна використовувати для створення сценаріїв програм, які підтримують Windows Script, таких як Internet Explorer або навіть операційна система Windows. ECMA стандартизована мова JavaScript / JScript як ECMAScript (стандарт ECMA-262).

Чим JavaScript відрізняється від Java? Мова програмування JavaScript, розроблена компанією Netscape, Inc., не є частиною платформи Java. JavaScript не створює аплети або окремі програми. У своїй найпоширенішій формі JavaScript знаходиться всередині HTML-документів і може забезпечити рівні інтерактивності для веб-сторінок, які неможливо досягти за допомогою простого HTML. Основні відмінності між Java та JavaScript: Java є мовою програмування ООП, тоді як JavaScript є мовою сценаріїв ООП. Java створює програми, які запускаються у віртуальній машині або браузері, тоді як код JavaScript виконується лише у браузері. Код Java має бути скомпільований, поки весь код JavaScript є текстом. Для них потрібні різні плагіни.

Включення JavaScript покращує роботу веб-сторінки, перетворюючи її зі статичної сторінки в інтерактивну. Нагадаю, JavaScript додає поведінку веб-сторінкам. JavaScript має ряд переваг перед використанням різних мов для розробки бекенда та інтерфейсу. Це набір чудових технологій для розробки динамічних високопродуктивних додатків. Спільне використання та повторне використання коду допомагають зменшити кількість рядків коду до 40% під час спільного використання фрагментів коду або навіть шаблонів, бібліотек і шаблонів. Це робить обслуговування або рефакторинг дуже простим і економічним. Крім

того, використання однієї мови допомагає зробити розробку команд більш ефективнішою та зменшує відмінності між ними, оскільки всі команди, які використовують програму, краще розумітимуть, що роблять інші. Крім того, вам не доведеться турбуватися про відмінності в синтаксисі, оскільки проекти використовують різні мови.

JavaScript також покращує швидкість і продуктивність веб-додатків. PayPal переходить із Java Node.js як платформи сервера додатків і публікує звіт про те, як JavaScript і Node.js можуть покращити продуктивність програми та значно зменшити кількість написаного ними коду. Згідно зі звітом, додаток будується вдвічі швидше, ніж Java, яка має невелику кількість членів команди.

Найбільш популярні технології для створення веб-сторінок, за результатами опитування Stack Overflow:

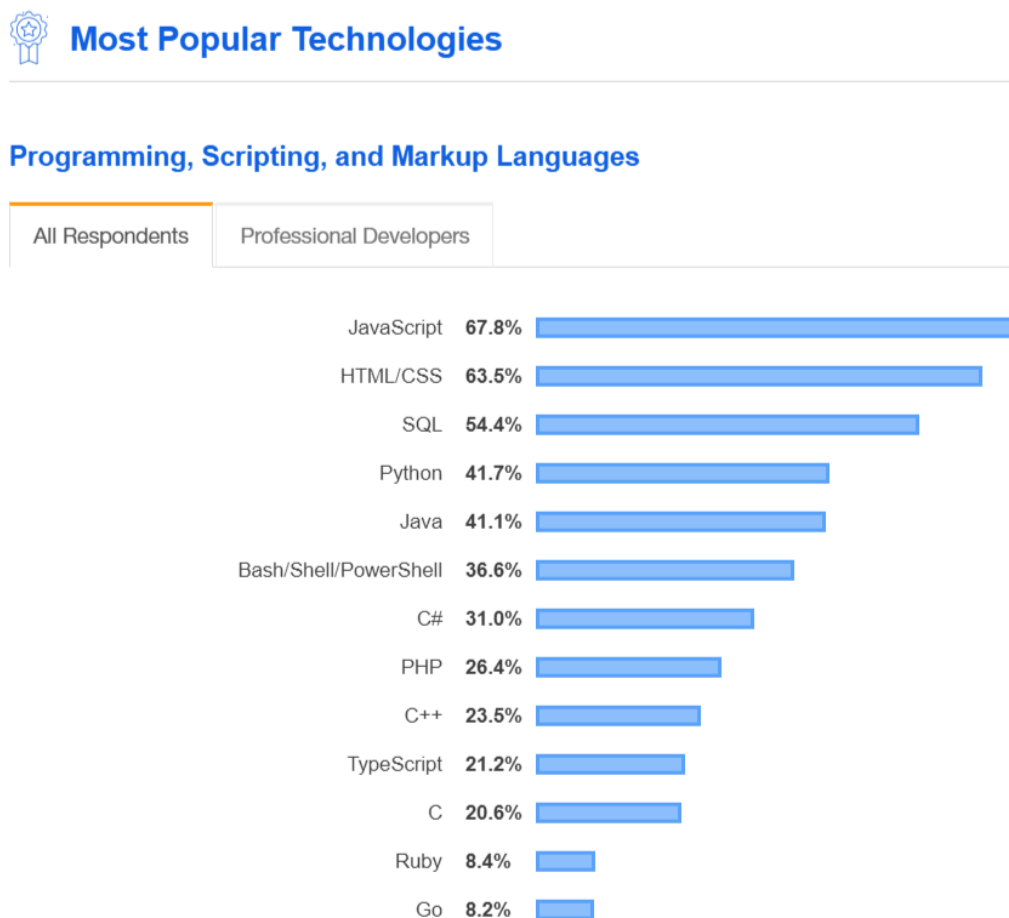


Рис. 2.3. Результати дослідження

Як можна побачити на рис. 2.3 69,8% від загальної кількості людей, які були опитані, вказали JavaScript як технологію, яку вони часто використовують. Це говорить про те, що JavaScript є однією з найпопулярніших мов для створення веб-додатків.

JavaScript сертифікований Netscape 1996 ECMA (Ecma International - Європейський Союз для систем регулювання даних і комунікацій), організації, що відповідає за стандарти IT та ІКТ. Глобальна робота ECMA дозволяє модернізувати JavaScript відповідно до поточних технічних стандартів. З тих пір були випущені різні версії ECMAScript, особливо EcmaScript6 (ES6). З моменту свого основного випуску в 2015 році ECMA планує випускати нові моделі щороку. Тому ECMA змінила назву ECMAScript з вихідного номера на номер року. JavaScript зараз дуже популярний і відомий як мова Інтернету.

Для його запуску потрібне середовище, наприклад браузер або Node.js. Мови програмування, такі як Java, можна компілювати, і з них можна створити виконуваний файл. Але для роботи мови сценаріїв, як-от JavaScript, потрібна підтримка середовища хостингу. У 1993 році Національний центр суперкомп'ютерних додатків (NCSA) випустив браузер NCSA Mosaic. Це був перший популярний графічний веб-браузер. У 1994 році в Маунтін-В'ю, Каліфорнія, була заснована нова компанія під назвою Mosaic Communications. Вони найняли багатьох оригінальних авторів NCSA Mosaic для створення нового браузера під назвою Mosaic Netscape.

Внутрішня кодова назва браузера — Mozilla, що є комбінацією Mosaic та Godzilla. Новий веб-браузер Mosaic Netscape 0.9 був випущений в 1994 році. Протягом 4 місяців Mosaic Netscape зайняв 75% ринку браузерів. Щоб уникнути проблеми з торговельною маркою з NCSA, браузер було перейменовано в Netscape Navigator, а Mosaic Communications було перейменовано в Netscape Communications. Netscape Communications зрозуміла, що браузеру потрібна мова, щоб зробити його більш динамічним. У 1995 році компанія найняла Брендана Айха з метою вбудувати мову програмування Scheme в Netscape Navigator.

ECMAScript – це стандарт, на якому базується JavaScript, і його часто називають ES. ECMAScript 2015, також відомий як ES6, є фундаментальною версією стандарту ECMAScript. Опублікований через 4 роки після останньої версії стандарту ECMAScript 5.1, він також позначив перехід із номера видання на номер року

ECMAScript 4 (ES 4) версія JavaScript, яка ніколи не існувала. Для цього оновлення заплановано різні нові функції, включаючи класи, простори імен, інтерфейси, визначення типів тощо. Основна мета цього випуску — зробити мову придатною для великомасштабної розробки програмного забезпечення, з акцентом на розробку бібліотеки ES4, і зробити її сумісною з ES3. Однак у липні 2008 року від ES4 відмовилися через розбіжності між двома командами, які працювали над новими функціями мови. Натомість вони погодилися випустити додаткові оновлення ECMAScript 5.

П'ята версія ECMAScript була випущена в грудні 2009 року після випуску ES4. Це допомагає покращити кодування, методи пошуку та обробки полів, таких як функції та атрибути карти, зменшення indexOf фільтрів і forEach та JSON..

JavaScript ES6 — це новіша версія JavaScript, представлена в 2015 році. ECMAScript – це стандарт, який використовує мову програмування JavaScript. ECMAScript надає специфікацію того, як має працювати мова програмування JavaScript.

ES6 відноситься до версії 6 мови програмування ECMA Script. ECMA Script це стандартизована назва для JavaScript, а версія 6 — наступна версія після версії

5, яка була випущена в 2011 році. Це значне покращення мови JavaScript і додає багато інших функцій, призначених для полегшення великомасштабної розробки програмного забезпечення. ECMAScript, або ES6, був опублікований у червні 2015 року. Згодом його було перейменовано на ECMAScript 2015. Підтримка повної мови веб-браузером ще не завершена, хоча основні частини підтримуються. Основні веб-браузери підтримують деякі функції ES6. Однак можна використовувати програмне забезпечення, відоме як транспілер, для перетворення коду ES6 в ES5, який краще підтримується в більшості браузерів.

Кожного, хто проходив співбесіду з JavaScript протягом останніх 2 років, запитали щонайменше 1671 раз, що таке ES6 і які функції він пропонує. Але хтось помітив, що JS, очевидно, раптом не зупинився на ES6 чи ES7.

Для кращого розуміння історичну послідовність розвитку JavaScript та його стандартів дивіться таблицю нижче:

Даты релиза различных версий Javascript

Издание	Официальное название	Дата публикации
ES8	ES2017	июнь 2017
ES7	ES2016	июнь 2016
ES6	ES2015	июнь 2015
ES5.1	ES5.1	июнь 2011
ES5	ES5	декабрь 2009
ES4	ES4	отменен
ES3	ES3	декабрь 1999
ES2	ES2	июнь 1998
ES1	ES1	июнь 1997

2.3. Node.js

Node.js – це кросплатформна платформа програмування на стороні сервера з відкритим вихідним кодом, побудована на JavaScript Google Chrome V8, яка дозволяє виконувати JavaScript за межами браузера. Розроблено Райаном Далем у 2009 році. Node.js — це неблокуюча асинхронна система вводу-виводу, керована подіями. Тобто він викликає наступний виклик API, не чекаючи завершення одного виклику API. Замість цього використовуйте функцію зворотного виклику, щоб повернути попередню подію відразу після наступної події, зазначеної раніше.

Node.js є масштабованим, легким та ефективним, що робить його ідеальним для розробки веб-додатків у режимі реального часу. Node.js використовує основний модуль HTTP для самостійного створення сервера, моніторингу вхідні запити, обробляти ці запити, перевіряти точність введених даних, Просто підключіться до бази даних і поверніть файл JSON або відобразіть сторінку HTML, і вона поверне відповідь.

Node (або формальніше Node.js) - кросплатформенна середовище виконання з відкритим вихідним кодом, яка дозволяє розробникам створювати всілякі серверні інструменти та програми використовуючи мову JavaScript. Середовище виконання призначене для використання поза контекстом браузера (тобто виконується безпосередньо на комп'ютері або серверній ОС). Таким чином, середовище виключає JavaScript для браузера і додає підтримку більш традиційних OS API-інтерфейсів, включаючи бібліотеки HTTP і файлових систем.

Рис. 2.4 Показує потік односторонніх неблокуючих подій вводу-виводу в Node.js. Платформа створює цикл подій на основі одного потоку даних для виконання кожного запиту, який надходить до сервера. Це не блокує введення-виводу і покращує швидкість і продуктивність таких програм.

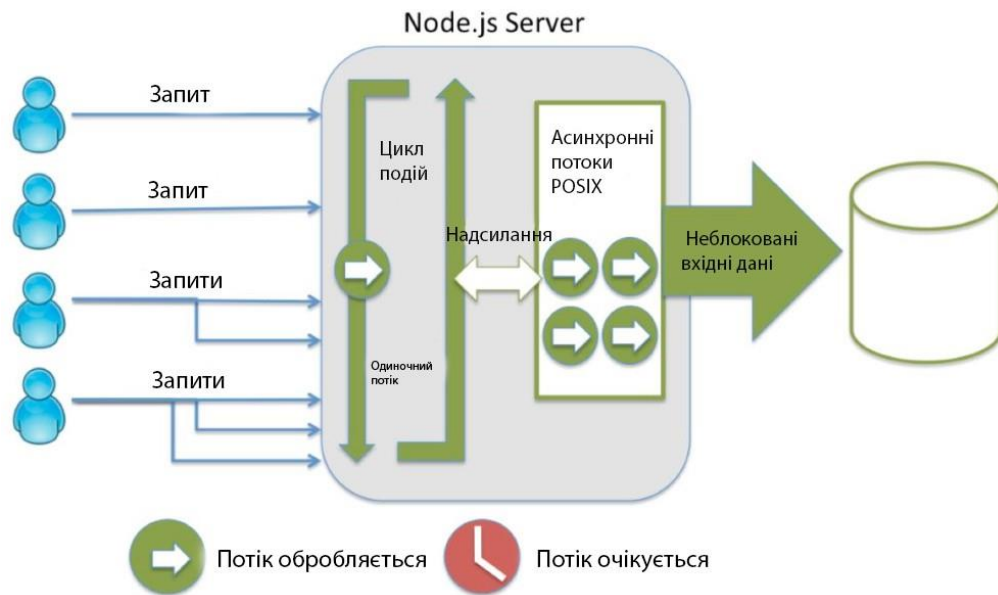


Рис. 2.4. Виконання однопоточкового вводу-виводу Node.js

Сервер Node.js обробляє декілька запитів одночасно. Ці запити обробляються одним потоком, утворюючи цикл подій. Цей цикл делегується асинхронному потоку для неблокуючого виконання вводу-виводу.

Стабільність Node.js регулярно покращується, а нові функції JavaScript із щорічного випуску ECMAScript були інтегровані в нещодавно випущені випуски Node.js.

2.4. Express

Express.js – це швидкий і легкий фреймворк, який використовується в основному для розробки веб-додатків, і розробники Node.js по всьому світу повністю закохані в цю платформу. Express.js надає всі функції веб-програм, не затьмарюючи можливості Node.js. За допомогою цієї статті я дам вам повне уявлення про основи Express.js, що допоможе вам розпочати роботу з ним. Express JS — це фреймворк Node.js, розроблений для швидкого створення кросплатформних мобільних додатків API для веб-програм і полегшення роботи вузла js. Express — це структура веб-додатків node js, яка надає широкі можливості для створення веб- та мобільних додатків. Він використовується для створення односторінкової, багатосторінкової та гібридної веб-додатки. Це шар, побудований на вершині Node js, який допомагає керувати серверами та маршрутами.

Express - Це фрагмент проміжного інструменту, який виконується зверху вниз у циклі запиту/відповіді. Окремі посередники мають доступ до бар'єрних запитань і відповідей, а також наступних функцій, які передаються від одного медіа до іншого. Посередник отримує запит, виконує внутрішній код, модифікує запит і об'єкти відповіді, викликає наступні функції та активує проміжних посередників у найближчій черзі. Швидкі програми можуть отримати доступ до проміжного програмного забезпечення рівня програми, проміжного програмного забезпечення рівня маршрутизатора, проміжного програмного забезпечення для обробки помилок, мікропрограми та проміжного програмного забезпечення сторонніх розробників.

Проміжне програмне забезпечення на рівні програмного забезпечення використовує функції `app.use()` і `app.method()` для з'єднання програмних об'єктів. де "метод" - це виконувана команда HTTP. Програмне забезпечення середнього рівня маршрутизатора відрізняється від середнього рівня програмного забезпечення лише тим, що воно підключається до швидкого екземпляра маршрутизатора. Загальна програма Еллера пов'язана з програмним об'єктом. і зазвичай має чотири параметри: запит, відповідь і помилка. і Наступні об'єкти.

Вбудовані посередники, такі як `express.static` і `express.json`, обслуговують

статичні файли та аналізують майбутні запити у файли JSON відповідно. Крім того, деяке проміжне програмне забезпечення сторонніх розробників розширює функціональність програм Express шляхом аналізу файлів cookie або аналізу файлів cookie.

Крім того, маршрутизатор у експрес-режимі розділяє програму на кілька міні-експрес-додатків, які об'єднуються, щоб утворити експрес-додаток. Експрес-маршрутизатори складаються з команд HTTP (GET, POST, PUT, DELETE) і шляхів або розташування ресурсів. Крім того, функції зворотного виклику також вказуються в методі маршрутизації, де як параметри можна використовувати кілька функцій зворотного виклику. Такі функції використовують функцію `next()`, переходять до іншої функції зворотного виклику та викликають наступний метод.

2.5. MongoDB та Mongoose

MongoDB є найпопулярнішою з нової породи нереляційних баз даних NoSQL. Зокрема, це база даних документів, яку також називають документоорієнтованою базою даних або сховищем документів.

Структури документів зазвичай узгоджуються з об'єктами, з якими розробники працюють у кодї, що є більш гнучким підходом, ніж таблична структура реляційної бази даних, орієнтована на рядки і стовпці. Розробники можуть переробляти структури документів (даних) у міру того, як вимоги їх додатків змінюються з часом.

Завдяки такому підходу структури даних стають подібними до коду — обидві знаходяться під контролем розробників. Бази даних документів популярні серед інших платформ для електронної комерції та торгівлі цінними паперами, оскільки вони добре масштабуються на кількох серверах для підтримки великих обсягів даних і трафіку.

Модель документа MongoDB проста для вивчення та використання, але при цьому вона надає всі можливості, необхідні для задоволення найскладніших вимог у будь-якому масштабі. Ми надаємо драйвери для понад 10 мов, а спільнота створила ще десятки.

MongoDB — програма керування базами даних NoSQL з відкритим вихідним кодом. NoSQL використовується як заміна традиційних реляційних баз даних. Бази даних NoSQL корисні для обробки великих обсягів розподілених даних. MongoDB — це інструмент, який може керувати документально-орієнтованою інформацією, зберігати або отримувати інформацію.

MongoDB Atlas — це рішення хмарних баз даних для сучасних додатків, доступне по всьому світу. Ця найкраща у своєму класі автоматизація та усталені методи пропонують розгортати повністю керовану MongoDB в AWS, Google Cloud та Azure. Він також забезпечує доступність, масштабованість та відповідність найсуворішим вимогам безпеки та конфіденційності даних.

MongoDB Cloud — це уніфікована платформа даних, яка включає глобальну хмарну базу даних, пошук, озеро даних, мобільні служби та служби додатків.

MongoDB зберігає записи даних як документи BSON. BSON — це двійкове

представлення документів JSON, хоча воно містить більше типів даних, ніж JSON. Документи, в свою чергу, збираються в колекції. Якщо ви вже знайомі з реляційними базами даних, ви можете вважати колекцію еквівалентною таблиці, але без схеми. На відміну від записів у реляційній таблиці, документи в колекції можуть мати різні поля, хоча зазвичай усі документи в колекції мають схоже або пов'язане призначення. Колекції існують у межах даної бази даних.

Більшість компаній використовують MongoDB як розподілену базу даних на кількох географічно розкиданих серверах у конфігурації, яка називається кластером. Кластери дозволяють базі даних MongoDB горизонтально масштабуватися на багатьох серверах із розподілом (автобалансування). Вони також дозволяють програмам реплікувати дані між серверами, щоб забезпечити високу доступність за допомогою функції, яку MongoDB викликає набори реплік, таким чином підвищуючи загальну продуктивність і надійність кластера MongoDB.

Хоча MongoDB є неструктурованою базою даних і покладається на дисципліну розробників програмного забезпечення для підтримки структури файлів програмного забезпечення, багато з них потребують організованих даних для належного функціонування. Мангуст покликаний вирішити цю проблему. MongoDB підтримує транзакції ACID з кількома документами, навіть між наборами реплік і розподіленими кластерами. Це означає, що якщо з'єднання розривається до завершення транзакції, або якщо будь-яка команда в транзакції не вдається, то база даних відкочує всі зміни, внесені під час транзакції. Відповідність ACID також є ключовою перевагою для реляційних баз даних.

Щоб отримати дані в бази даних і вивести з них, MongoDB використовує мову запитів MongoDB (MQL). Він використовує той самий синтаксис, що й документи, що полегшує роботу розробникам, але він далеко не такий інтуїтивно зрозумілий, як SQL, стандартна мова запитів для реляційних баз даних, і багато аналітиків заперечують використання «інтуїтивно зрозумілих» і «SQL» у те саме речення.

Висновки до розділу 2

Елементи веб-розробки швидко змінюються, розробляються нові мови та методи розробки. Розробка веб-сайтів – це поєднання проміжної та внутрішньої розробки. Ви можете вибрати з широкого спектру мов програмування та систем веб-розробки. Раніше не існувало мов програмування, яка б сприяла розробці програмного забезпечення на середньому та локальному рівні. Однак з появою NodeJS як серверної платформи JavaScript може забезпечити повну розробку програмного забезпечення на одній мові програмування. Тому для цього проекту було обрано JavaScript та Node.js.

3. РОЗРОБКА ІНТЕРНЕТ-МАГАЗИНУ

3.1. Вступ

Прототип Porydbay.com був розроблений, щоб довести, що для розробки веб-додатка використовувався цілий стек JavaScript. Розроблена програма є мінімально життєздатним продуктом (MVP) для купівлі та продажу продуктів через Інтернет. Пізніше MVP може бути розроблено для охоплення інших аспектів програми електронної комерції в Інтернеті.

Porydbay.com - це онлайн-додаток, яка сприяє онлайн-маркетингу. На пристрої не повинен входити хтось, крім адміністратора та користувача. Керівники повинні надавати інформацію бази даних, описи продуктів і дані бази даних. Крім того, ви можете редагувати все, переміщатися в центр обробки даних, додавати та створювати нотатки.

Розробка програми MVC (Model View Controller). Серверна частина складається з модуля і адміністратора, а дизайн реалізується на стороні клієнта. Програма, розроблена у форматі MVC, розділена на три окремі частини: модель, контролер і дисплей, які працюють разом.

Шаблон – це об’єкт, який містить дані програми. Модель може отримувати дані від контролера, а також відправляти дані на дисплей. Контролер є основою програми, яка містить всю логіку, необхідну для запуску програми. Такі завдання додають, змінюють, видаляють і видаляють дані з модулів, оброблених на контролері. Перегляд – це те, що бачить і з чим взаємодіє користувач. Користувачі обмінюються даними візуально.

Рис. 3.1. зображує модель інтерфейсу і контролера сервісу MVC, а також те, як вони використовують один одного.

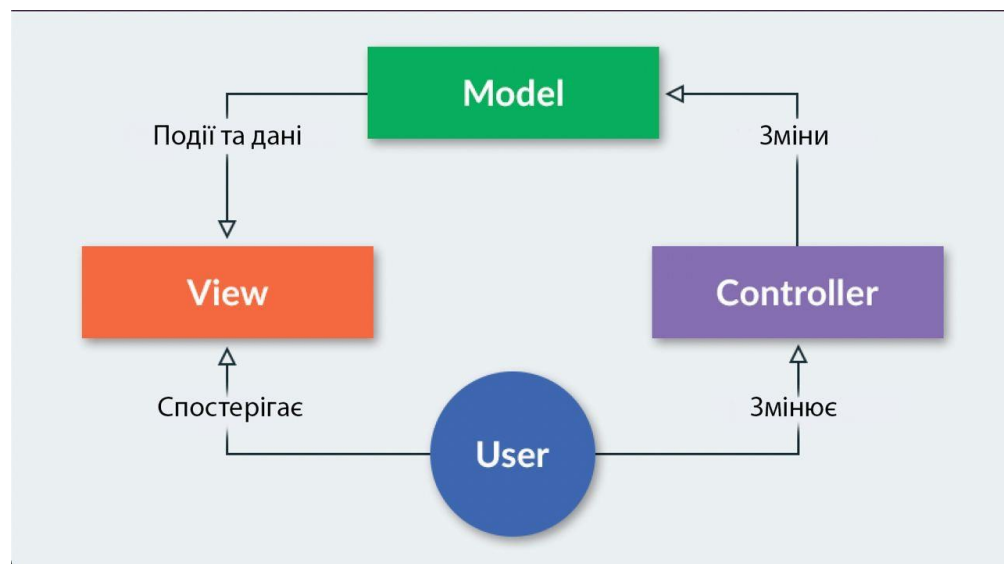


Рис.3.1. Область моделі, інтерфейсу і контролера програми MVC

Як ви бачите на рис.3.1., view збирає вхідні дані від клієнта і відправляє їх у controller. Далі controller збирає дані і змінює model. Згодом, view отримує нові дані з моделі і показує відредаговані дані в view.

Sublime Text — це програмне забезпечення для розробки додатків, яке допомагає підприємствам керувати рефакторингом коду, налагодженням, редагуванням на кількох моніторах, виділенням синтаксису тощо з уніфікованої платформи. Це дозволяє співробітникам використовувати функцію автозаповнення для створення пропозицій коду та автоматичного надання посилань на визначення.

Sublime Text, безсумнівно, є одним із найпотужніших текстових редакторів. Про це свідчить кількість задоволених користувачів. Якщо ви дослідите його, ви врешті-решт побачите, наскільки красиво його потужні функції приховані за простим та елегантним інтерфейсом. Керування версіями - програмування для відстеження змін переважно у вихідному коді, за допомогою програмістів, які допомагають розробникам із зміями, заповненими конвертами багатьма людьми в різний час.

Системи контролю версій поділяються на: локальний, що дозволяє зберігати дані лише на локальному комп'ютері (наприклад, SCCS і RCS) централізований, заснований на архітектурі клієнт-сервер (наприклад, CVS, Subversion) розподілений, заснований на архітектурі P2P (наприклад, BitKeeper, Code Co-op, Git, svk). Перша група зберігає лише версії файлів з локального комп'ютера. У централізованих рішеннях є один центральний репозиторій, з яким усі користувачі системи синхронізують свої зміни. Розподілені рішення дозволяють запускати рівні, незалежні гілки, які можна вільно синхронізувати один з одним, наприклад, електронною поштою (Code Co-op). Вибір відповідного типу системи контролю версій залежить від потреб проекту.

3.2.NPM, Node.js, MongoDB, Express and EJS

NPM - менеджер пакетів за замовчуванням для середовища Node.js, також можна використовувати для керування інтерфейсним рівнем веб-додатка. NPM також є сховищем пакетів (зване реєстром npm) і назвою компанії (npm, Inc.), яка ним керує. NPM повністю написаний на JavaScript, започаткований Ісааком З. Шлютером і натхненний менеджером пакетів PHP PEAR і Perl CPAN. NPM є

менеджером пакетів за замовчуванням для середовища виконання JavaScript Node.js. Він складається з клієнта командного рядка, який також називається npm, і онлайнвої бази загальнодоступних і платних пакетів, що називається реєстром npm. Реєстр доступний через клієнт, а доступні пакети можна шукати та шукати на веб-сайті npm. Менеджером пакетів і реєстром керує npm, Inc.

```
sudo npm install -express -save -g
```

Ми використовували Express Generator для створення стандартної структури папок програми, в якій найбільш часто використовуване проміжне програмне забезпечення вже включено в програму. Наступна команда використовується для встановлення Express Generator у вашій програмі.

```
sudo npm install express-generator -g
```

Різні версії npm, Node.js і Express, які використовуються при розробці програми, показано на Рис 3.2.

```
C:\Users\mshly>cd C:\WEB  
  
C:\WEB>npm -v  
8.5.5  
  
C:\WEB>node -v  
v16.15.0
```

Рис. 3.2. Перевірка версій NPM, NODE та EXPRESS

Для створення сервісу використовується версія Node.js версії 16.15.0, як показано на рис. 3.2., яка була останньою версією (LTS) на момент створення цього сервісу. Крім того, при створенні сервісу використовувалися npm версії 8.5.5.

MongoDB може бути встановлений локально на комп'ютері, на якій створюється сервіс або сразу на MongoDB Atlas в якості онлайн сховища. Аккаунт було зареєстровано на веб-сайті MongoDB. Після створення проекту було створено новий безкоштовний кластер MongoDB. Крім того, створюється користувач із доступом для читання та запису до бази даних, яка використовується для читання та запису даних у програмі Node.js. Крім того, поточна IP-адреса комп'ютера додається до списку IP-адрес, щоб програми могли взаємодіяти з сервером. Нарешті, драйвер MongoDB було використано у програмі за допомогою команди "npm install mongodb - save".

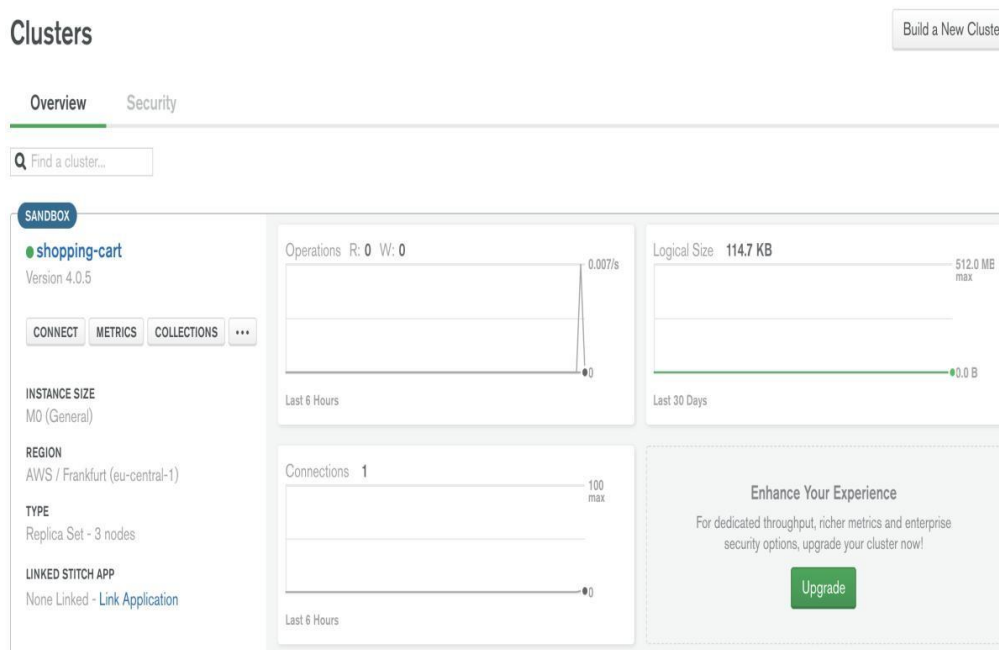


Рис. 3.3. Огляд кластера MongoDB, який використовується для програми.

Кластер MongoDB відображає геоданні, де знаходиться сервер, операції зчитування і запису, кількість активних з'єднань на сервері і розмір бази даних, як ми можемо побачити на рис.3.3.

Mongoose – це об'єктне моделювання MongoDB, розроблене для роботи в асинхронному середовищі. Давайте подивимося правді в очі, написання шаблону перевірки MongoDB, кастингу та бізнес-логіки — це непросто. Тому ми написали «Мангуст». Mongoose забезпечує просте рішення на основі схеми для моделювання даних вашої програми і включає в себе вбудований приведення типів, перевірку, побудову запитів, гачки бізнес-логіки та багато іншого.

EJS — це проста мова шаблонів, яка дозволяє створювати HTML-розмітку за допомогою звичайного JavaScript. Ніякої релігійності щодо того, як організувати речі. Немає заново винаходу ітерації та потоку керування. Це просто звичайний JavaScript. EJS ми можемо додати за допомогою команди `npm install ejs -save`.

3.3.Додаткові пакети Node

Крім основних пакетів та інструментів, описаних раніше, в проєкті були встановлені численні інші пакети для забезпечення додаткових функцій програми. Деякі з них обговорюються нижче.

Nodemailer — це модуль Node.js, який дозволяє з легкістю надсилати електронні листи з вашого сервера. Незалежно від того, чи хочете ви спілкуватися зі своїми користувачами чи просто сповіщати себе, коли щось пішло не так, одним із варіантів для цього є електронна пошта.

Щоб перевірити електронні листи, надіслані за допомогою Nodemailer, ми будемо використовувати Mailtrap, онлайн-інструмент для комплексного тестування електронної пошти в середовищі попереднього виробництва. Він ловить наші повідомлення, відображатиме, як вони мають виглядати в реальному поштовому клієнті, а також допомагатиме аналізувати та налагоджувати їх. Mailtrap також надає параметри тестування прихованої копії та дозволяє ділитися результатами тестування електронної пошти з іншими членами команди, а також пересилати електронні листи на справжні підтверджені адреси. Навіть якщо у вас ще немає облікового запису, весь процес налаштування займе всього пару хвилин. Mailtrap інтегрується як звичайний SMTP-сервер. Швидко зареєструйтеся (це безкоштовно), перейдіть на вкладку налаштувань SMTP у папці «Вхідні», скопіюйте необхідні налаштування та вставте їх у сценарій програми.

PDFKit — це бібліотека для створення PDF-документів для Node і браузера, яка полегшує створення складних багатосторінкових документів для друку. API охоплює ланцюжок і включає як функції низького рівня, так і абстракції для функціональності вищого рівня. API PDFKit розроблено, щоб бути простим, тому генерувати складні документи часто так само просто, як кілька викликів функцій. Його

було встановлено в проєкт за допомогою команди "npm install pdfkit -save".
Рахунок-

фактура був створений з динамічними даними користувача. PDF-рахунок, створений за допомогою "pdffit", показано на рис.3.4.

Prydbay.com
Invoice

	Product	Quantity	Rate	Amount
1	Oneplus 6	1	€549.99	€549.99
2	Computer	2	€599.99	€1199.98
Total Sum:				€1749.97

Рис. 3.4. Рахунок-фактура

Рахунок-фактура – це документ, у якому перераховуються продукти та послуги, які компанія надає клієнту, і встановлюється зобов'язання клієнта сплатити підприємство за ці продукти та послуги. Виставлення рахунків має вирішальне значення для малого бізнесу, оскільки рахунки є діловими документами, які дозволяють компаніям отримувати оплату за свої послуги. Популярним способом розпочати роботу з рахунками-фактурами є використання шаблонів рахунків-фактур. Шаблони рахунків-фактур заощаджують ваш час, тому вам не доведеться повторно заповнювати ту саму інформацію щоразу, коли ви створюєте рахунок-фактуру. Якщо ви хочете стати трохи більш просунутим, FreshBooks також створює програмне забезпечення для рахунків-фактур, яке до смішного простого у використанні.

Рахунки-фактури служать важливою метою як для підприємства, що надсилає рахунок, так і для клієнта, який його отримує. Для малого бізнесу рахунок-фактура допомагає прискорити процес оплати, надаючи клієнтам сповіщення про платіж, який підлягає сплаті. Для клієнтів рахунки-фактури забезпечують організований запис витрат із детальними деталями та можуть допомогти у веденні обліку.

Найпростіший спосіб керувати рахунками-фактурами та відстежувати їх через онлайн-систему виставлення рахунків. Програмне забезпечення для виставлення рахунків дозволяє швидко та легко надсилати професійні рахунки-

фактури прямо з комп'ютера чи мобільного пристрою. Клієнти можуть оплатити лише одним кліком, а ви можете відстежувати, хто заплатив, а хто запізнюється на одній централізованій панелі інструментів. Традиційно ручні завдання, такі як ведення бухгалтерії та нумерація рахунків-фактур, виконуються на задній панелі, тому легко стежити за своїм грошовим потоком.

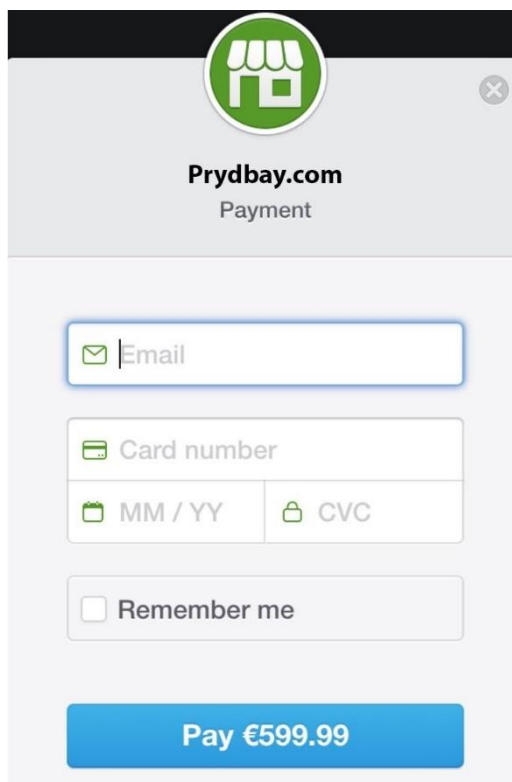
The image shows a mobile payment interface. At the top, there is a green circular logo with a white storefront icon. Below the logo, the text "Prydbay.com" and "Payment" are displayed. The form contains several input fields: an "Email" field with an envelope icon, a "Card number" field with a card icon, and two smaller fields for "MM / YY" (with a calendar icon) and "CVC" (with a lock icon). There is also a "Remember me" checkbox. At the bottom, a prominent blue button contains the text "Pay €599.99".

Рис. 3.5. Платіжна форма Stripe

На рис. 3.5 показана платіжна форма компанії Stripe, яка збирає інформацію про кредитну картку клієнта. Форма відправляється в stripe скрипт разом із загальною нараховуваною сумою і ключем API Stripe, наданим додатку.

3.4. Програмна логіка та безпека

Середовище створення проекту, всі пакети вузлів та інструменти були встановлені. Далі був етап написання програмного коду html, css, js. Структура файлової системи проекту показана на рис. 3.6.

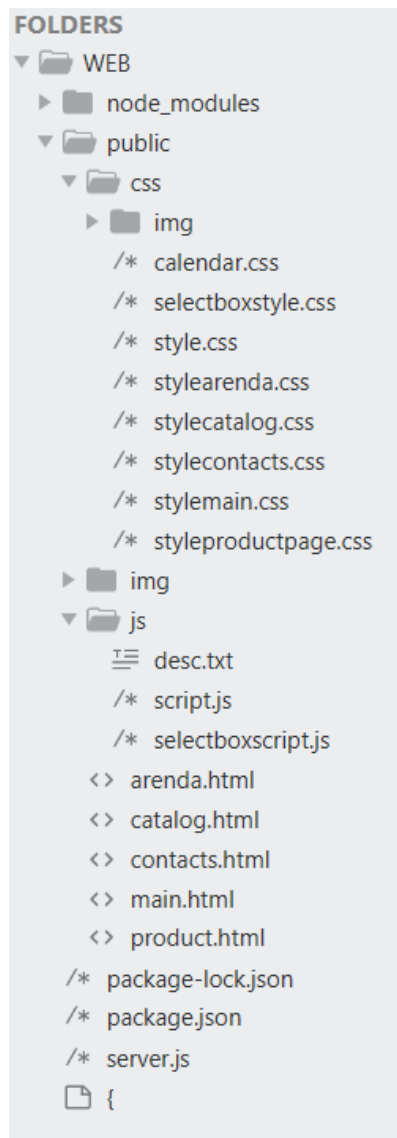


Рис. 3.6. Файлова структура програми

Сервіс складається з різних папок в яких знаходяться html, css, js файли та зображення, як можна побачити на рис. 3.6, для реалізації структури та сервісу програми. Файли знаходяться в різних папках, утворюючи структуру.

Структура повинна залишатись зрозумілою і простою. Моделі, контролери та види програми розміщуються в різних теках, які належать структурі MVC. Каталог "node_modules" містить всі залежності серверної сторони. Папка "public" містить css, js файли та зображення, які поділені на папки з відповідною назвою.

Є кілька задач, які найбільш ефективно вирішуються з використанням файлу .gitignore: Не турбуватись з вибором потрібних файлів для індексації (которая git add). У великому проекті часто буває багато файлів, які не підлягають версії. Своєю лептою вносяться редактори та засоби розробки, компілятори, дебагери, інструменти для пошуку та сама операційна система.

Зробити локальний конфіг, який не зачепить pull-ом. Припустимо, для роботи вашого проекту потрібний конфіг. Значення в ньому залежать від місця виконання, погоди та настрою розробника. Варіант 1: зробити local.conf та вносити зміни за необхідності. Якщо ви його раптом додасте та закомітите, то у вас будуть конфлікти під час pull/push. Або з пуллом до вас прийде чужий конфіг. Варіант 2: local.conf.example, що версіонується, і local.conf, що ігнорується. У роботі використовується другий, для формування вручну або автоматично використовується перший. Добре, зручно та не перешкоджає автоматизації.

```

main.html x
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="css/style.css">
  <link rel="stylesheet" href="css/stylemain.css">
  <link rel="shortcut icon" href="img/testfavicon2.png" type="image/png">
  <title>Cosplay Shop</title>
</head>
<body>
  <div class="backgroundimg"></div>
  <header>
    <div class="headerlogo">
      
    </div>
    <div class="outer">
      <div class="headerblocks inner" onclick="changePage(id)" id='main'>Главная</div>
    </div>
    <div class="outer">
      <div class="headerblocks inner" onclick="changePage(id)" id='catalog'>Каталог</div>
    </div>
    <div class="outer">
      <div class="headerblocks inner" onclick="changePage(id)" id='arenda'>Условия Аренды</div>
    </div>
    <div class="outer">
      <div class="headerblocks inner" onclick="changePage(id)" id='contacts'>Контакты</div>
    </div>
  </header>
  <div class="secondmainline">
    <div class="firstmaincolumn">
      
      
      
    </div>
    <div class="secondmaincolumn">
      <div class="maintext">

```

Рис. 3.8. HTML структура головної сторінки

На малюнку 3.8 можемо побачити структуру головної сторінки сервісу. Саме на цю сторінку попадає клієнт в перший раз. В head ми підключаємо стилі css до нашої веб-сторінки, задаємо назву та іконку сайту. В body в нас іде вже сама структура сайту. Основний тег який можна побачити на малюнку це тег div, тег блоків завдяки яким ми створюємо структуру сайту, в них ми можемо всунути усілякий контент (зображення, текст, тощо)

Файл «server.js» є основним файлом сервісу та запуском серверу. Він імпортує все необхідне проміжне програмне забезпечення, налаштовує сервер який відображає веб-сторінку, створює об'єкт програми та експортує цей об'єкт із модуля. Шматок коду із файлу «server.js» наведено на рис. 3.10.

```
1  const express = require('express');
2  const path = require('path');
3
4  const app = express();
5  const port = process.env.PORT || 3000;
6
7  // sendFile will go here
8  app.use(express.static(__dirname + '/public'));
9
10 app.get('/', function(req, res) {
11   res.sendFile(path.join(__dirname, 'public/main.html'));
12 });
13
14 app.listen(port);
15 console.log('Server started at http://localhost:' + port);|
```

Рис. 3.10. Фрагмент коду server.js

На мал. 3.10, в javascript файл «server.js» в перших двох рядках імпортуються бібліотеки express та path. Далі ми присвоюємо функцію express() до змінної app та присвоюємо порт 3000 який ми будемо далі використовувати в адресі локального веб сайту. Також ми відправляємо на сервер файл main.html як головну сторінку та через express.static відправляємо супроводжуючі файли такі як css, js та зображення.

У файлі «package-lock.json» вказана детальна інформація про проект, а також необхідні залежності проекту. Він може містити ключові слова, домашню сторінку, помилки, ліцензію, автора, співавторів, сховище тощо, на додаток до основної інформації, такої як назва, версія та опис. Файл «package-lock.json», який використовується в проекті, представлений на рис. 3.11.

Рис. 3.11. Файл «package-lock.json»

Як показано на рис. 3.11, файл 'package.json' включає ім'я, опис та номер версії програми, залежності, необхідні для програми, та номер їх версії, а також сценарії для запуску або зупинки програми.

Безпека є дуже важливим елементом у всіх веб-застосунках. Вразливість веб-застосунків дуже швидко зростає через збільшення кількості кібератак. Хакери дуже часто використовують різні методи, такі як фішинг і збирання даних для входу в аккаунти, а також даних від кредитної картки. Окрім цього, не добре захищені маршрути можуть забезпечити доступ хакерів до сервісу. З цього слідує, що веб-сервіси мають бути дуже захищені. Окрім цього, веб-сервіси з електронними платежами мають значну кількість грошових операцій через покупку та продаж товарів, ці сервіси мають створювати та вдосконалювати безпеку найвищим пріоритетом. Не значне порушення безпеки сервісу призводить до неочікуваних збитків для клієнтів і для компанії.

Аутентифікація — це перевірка автентичності частини даних, перевірених суб'єктом (установою, особою або системою). Ідентифікація — це опис дії, яка доводить або демонструє ідентичність особи чи речі; аутентифікація — це процес фактичної перевірки цієї особистості. Цей процес включає перевірку особи за допомогою перевірки документів, що посвідчують особу, перевірку автентичності веб-сайту за допомогою цифрового сертифіката, визначення віку конструкції за допомогою радіовуглецевого датування або перевірку продукту, подивившись на його упаковку та етикетку. Іншими словами, аутентифікація зазвичай включає перевірку принаймні одного типу ідентифікації особи.

Є також таке поняття як авторизація — це визначення та надання прав доступу до ресурсів інформаційної безпеки чи комп'ютерної безпеки, особливо до контролю доступу. Дієслово «авторизувати» може означати визначення політики доступу та прийняття доступу. Наприклад, персонал кадрів часто має право доступу до записів співробітників, і ця політика часто формалізується як правила контролю доступу в комп'ютерних системах. Під час виконання система використовує визначені правила контролю доступу, щоб вирішити, приймати чи відхиляти аутентифіковані запити доступу. Можуть бути авторизовані різноманітні ресурси, включаючи, але не обмежуючись, один або декілька файлів або даних, комп'ютерні програми, комп'ютерне обладнання та функції, що надаються комп'ютерним програмним забезпеченням. Натомість авторизованим є користувач

комп'ютера, комп'ютерне програмне забезпечення та/або інше обладнання.

Безпека веб-сайту вимагає пильності в усіх аспектах дизайну та використання веб-сайту. Ця вступна стаття не зробить вас гуру безпеки веб-сайтів, але вона допоможе вам зрозуміти, звідки надходять загрози, і що ви можете зробити, щоб захистити веб-програму від найпоширеніших атак.

Інтернет - небезпечне місце! З великою регулярністю ми чуємо про те, що веб-сайти стають недоступними через атаки відмови в обслуговуванні або показ зміненої (і часто шкідливої) інформації на своїх домашніх сторінках. В інших резонансних випадках мільйони паролів, адрес електронної пошти та даних кредитних карток були опубліковані в суспільному надбанні, піддаючи користувачів веб-сайту як особистому збентеженню, так і фінансовому ризику.

Ефективна безпека веб-сайту вимагає зусиль щодо дизайну всього веб-сайту: у вашому веб-додатку, конфігурації веб-сервера, ваших політиках створення та оновлення паролів і кодів на стороні клієнта. Хоча все це звучить дуже зловісно, хороша новина полягає в тому, що якщо ви використовуєте веб-фреймворк на стороні сервера, він майже напевно увімкне «за замовчуванням» надійні та добре продумані механізми захисту від ряду найбільш поширених атак. . Інші атаки можна пом'якшити за допомогою конфігурації веб-сервера, наприклад, увімкнувши HTTPS. І, нарешті, є загальнодоступні інструменти сканування вразливостей, які можуть допомогти вам з'ясувати, чи не зробили ви очевидних помилок.

XSS – це термін, який використовується для опису класу атак, які дозволяють зловмиснику вводити клієнтські сценарії через веб-сайт у браузері інших користувачів. Оскільки введений код надходить у браузер із сайту, код є надійним і може виконувати такі дії, як надсилання зловмиснику файлу cookie авторизації сайту користувача. Коли зловмисник має файл cookie, він може увійти на сайт, як ніби він є користувачем, і робити все, що може користувач, наприклад отримати доступ до даних своєї кредитної картки, переглянути контактні дані або змінити паролі.

Відображена вразливість XSS виникає, коли вміст користувача, який

передається на сервер, повертається негайно і без змін для відображення у браузері. Будь-які сценарії в оригінальному вмісті користувача будуть запущені під час завантаження нової сторінки. Наприклад, розглянемо функцію пошуку на сайті, де пошукові терміни закодовані як параметри URL-адреси, і ці терміни відображаються разом із результатами. Зловмисник може створити пошукове посилання, яке містить шкідливий сценарій як параметр і електронною поштою іншому користувачеві. Якщо цільовий користувач натисне це «цікаве посилання», скрипт буде виконано, коли відобразяться результати пошуку. Як обговорювалося раніше, це надає зловмиснику всю необхідну інформацію, щоб увійти на сайт як цільовий користувач, потенційно робити покупки як користувач або ділитися своєю контактною інформацією.

Постійна вразливість XSS виникає, коли шкідливий сценарій зберігається на веб-сайті, а потім знову відображається без змін, щоб інші користувачі могли ненароком виконати. Наприклад, дошка обговорень, яка приймає коментарі, що містять незмінений HTML, може зберегти шкідливий сценарій від зловмисника. Коли відображаються коментарі, скрипт виконується і може надіслати зловмиснику інформацію, необхідну для доступу до облікового запису користувача. Цей вид атаки надзвичайно популярний і потужний, оскільки зловмисник може навіть не мати прямої взаємодії з жертвами.

Хоча дані із запитів POST або GET є найпоширенішим джерелом уразливостей XSS, будь-які дані з браузера є потенційно вразливими, як-от дані файлів cookie, які відображаються браузером, або файли користувача, які завантажуються та відображаються.

Процес зміни даних користувача, щоб їх не можна було використовувати для запуску сценаріїв або іншим чином впливати на виконання коду сервера, відомий як очищення введення. Багато веб-фреймворків за замовчуванням автоматично очищають введення користувача з форм HTML.

Уразливості ін'єкції SQL дозволяють зловмисникам виконувати довільний код SQL у базі даних, дозволяючи отримувати доступ до даних, змінювати або видаляти їх незалежно від дозволів користувача. Успішна атака ін'єкції може

підробити ідентифікатори, створити нові посвідчення з правами адміністрування, отримати доступ до всіх даних на сервері або знищити/змінити дані, щоб зробити їх непридатними. Типи ін'єкції SQL включають SQL-ін'єкцію на основі помилок, ін'єкцію SQL на основі логічних помилок і ін'єкцію SQL на основі часу.

Express js — це мінімальний і гнучкий фреймворк веб-додатків Node.js, який надає надійні функції для розробки веб- та мобільних додатків. Ми можемо створювати веб-API за допомогою цього фреймворка дуже швидко, і він має величезну підтримку баз даних NoSQL, таких як MongoDB. Ви можете дізнатися більше про Node js Express js Framework в його офіційних документах.

Атаки CSRF використовуються через викрадення сеансів у авторизованого користувача і відсилання запитів з заміною реальних даних на сервер. Ця атака не викрадає дані; але вона маніпулює і змінює дані клієнта, наприклад як електронна адреса, пароль та може пересилати гроші.

Node.js — це кросплатформне середовище для запуску серверних і мережеских програм із відкритим кодом. Програми написані на JavaScript і можуть працювати в середовищі Node.js Microsoft Windows, GNU / Linux, macOS і IBM. Node.js забезпечує архітектуру, керовану подіями, і неблокуючу систему прикладного програмування вводу-виводу (API), яка оптимізує продуктивність і масштабованість програми. Ці технології часто використовуються для мережеских додатків реального часу.

Всіма зовнішніми пакетами можливо управляти через npm.

Node Package Manager (NPM) — це інструмент командного рядка, який встановлює, оновлює або видаляє пакети Node.js у вашій програмі. Це також онлайн-сховище для пакетів Node.js з відкритим кодом. Спільнота вузлів у всьому світі створює корисні модулі та публікує їх як пакети в цьому репозиторії.

Файл cookie – це пара ключ-значення, яка зберігається у браузері. Браузер додає файли cookie до кожного HTTP-запиту, який надсилається на сервер. У файлі cookie ви не можете зберігати багато даних. Файли cookie не можуть зберігати будь-які облікові дані користувача або секретну інформацію. Якби ми це зробили, хакер міг би легко отримати цю інформацію та вкрати особисті дані для

зловмисної діяльності. З іншого боку, дані сеансу зберігаються на стороні сервера, тобто в базі даних або сховищі сеансів. Таким чином, він може вмістити більші обсяги даних. Щоб отримати доступ до даних на стороні сервера, сеанс аутентифікується за допомогою секретного ключа або ідентифікатора сеансу, який ми отримуємо з файлу cookie під час кожного запиту.

Express-session - серверна платформа HTTP, яка використовується для створення проміжного програмного забезпечення сеансу та керування ним. Цей підручник присвячений сеансам. Таким чином, в центрі уваги буде бібліотека Express-session.

Захист транспортного рівня (TLS) — це стандартний протокол Internet Engineering Task Force (IETF), який забезпечує аутентифікацію, конфіденційність і цілісність даних між двома взаємодіючими комп'ютерними програмами. Це найпоширеніший протокол безпеки, який використовується сьогодні, і найкраще підходить для веб-браузерів та інших програм, які потребують безпечного обміну даними через мережу. Це включає сеанси перегляду веб-сторінок, передачу файлів, підключення до віртуальної приватної мережі (VPN), сеанси віддаленого робочого столу та передачу голосу через IP (VoIP). Зовсім недавно TLS інтегрується в сучасні технології стільникового транспорту, включаючи 5G, для захисту функцій основної мережі в мережі радіодоступу (RAN).

TLS розвинувся на основі протоколу Secure Sockets Layer від Netscape Communications Corp. і значною мірою витіснив його, хоча терміни SSL або SSL/TLS досі іноді використовуються як взаємозамінні. IETF офіційно взяв на себе протокол SSL, щоб стандартизувати його за допомогою відкритого процесу, і випустив версію 3.1 SSL у 1999 році як TLS 1.0. Протокол було перейменовано в TLS, щоб уникнути юридичних проблем з Netscape, яка розробила протокол SSL як ключову частину свого оригінального веб-браузера. Відповідно до специфікації протоколу, TLS складається з двох рівнів: протоколу запису TLS і протоколу рукоштовання TLS. Протокол запису забезпечує безпеку підключення, тоді як протокол рукоштовання дозволяє серверу і клієнту аутентифікувати один одного та узгоджувати алгоритми шифрування та криптографічні ключі перед обміном

будь-якими даними.

Остання версія TLS, 1.3, була офіційно завершена IETF у 2018 році. Основною перевагою порівняно з попередніми версіями протоколу є додані механізми шифрування під час встановлення зв'язку між клієнтом і сервером. Хоча попередні версії TLS також пропонують шифрування, TLS вдається встановити зашифрований сеанс раніше в процесі рукоштовування. Крім того, зменшується кількість кроків, необхідних для завершення рукоштовування, що істотно скорочує час, необхідний для завершення рукоштовування та початку передачі чи отримання даних між клієнтом і сервером. Іншим покращенням TLS 1.3 є те, що кілька криптографічних алгоритмів, які використовуються для шифрування даних, було видалено, оскільки вони вважалися застарілими і не рекомендовані для безпечного транспортування. Крім того, тепер потрібні деякі функції безпеки, які раніше були необов'язковими. Наприклад, криптографічні хеші алгоритму дайджесту повідомлень 5 (MD5) більше не підтримуються, потрібна ідеальна пряма секретність (PFS), а узгодження Rivest Cipher 4 (RC4) заборонено. Це виключає ймовірність того, що сеанс із шифруванням TLS використовує відомий небезпечний алгоритм або метод шифрування у TLS версії 1.3.

Переваги TLS очевидні при обговоренні використання TLS та його відсутності. Як зазначалося вище, сеанс із шифруванням TLS забезпечує безпечний механізм аутентифікації, шифрування даних і перевірку цілісності даних. Однак, якщо порівнювати TLS з іншим набором протоколів безпечної аутентифікації та шифрування, таким як Internet Protocol Security, TLS пропонує додаткові переваги і є причиною того, що IPsec замінюється TLS у багатьох ситуаціях розгортання на підприємстві.

3.5 Демонстрація прототипу інтернет-магазину цифрової техніки

В розробці програми ми використовували такі технології, як NodeJS та Express для серверної частини та HTML, CSS, JS для клієнтської частини. Для написання коду ми використовували Sublime Text3, дуже простий та зручний

текстовий редактор у який можна встановлювати різні плагіни для кожної мови програмування.

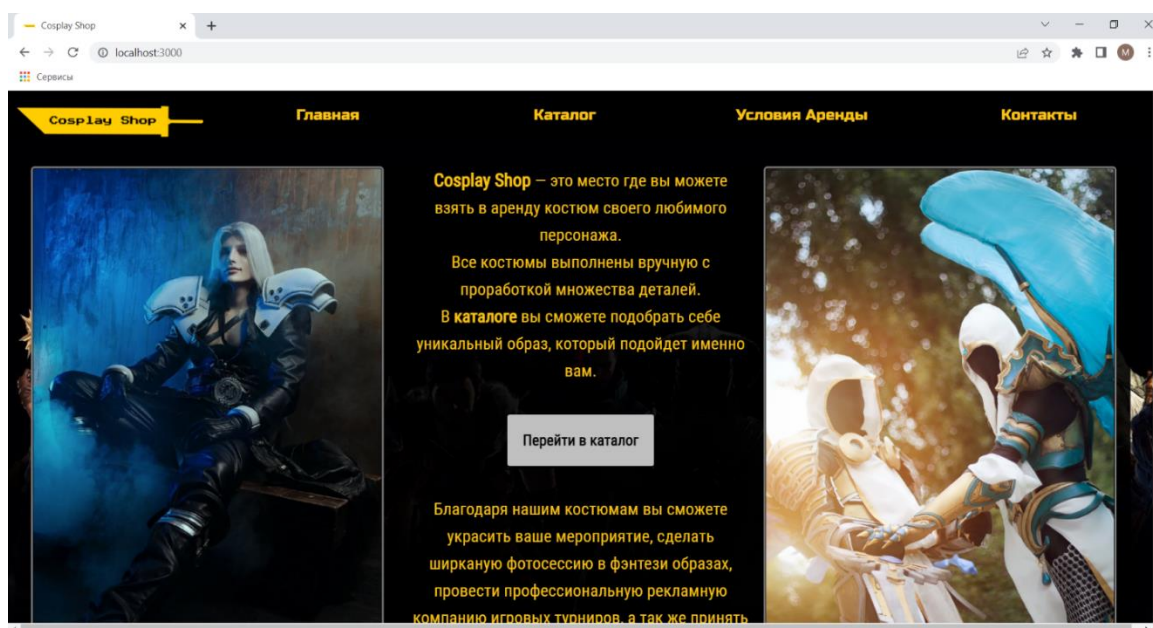


Рис. 3.13. Початкова сторінка веб-сервісу

Початкова сторінка програми для користувачів показана на рис 3.13. Вона має зверху панель для переходу до різних сторінок. На ній також знаходиться основна інформація про призначення веб-додатку та в центрі виділяється кнопка для переходу у каталог. Переход на наступну сторінку виконується завдяки функціям розробленими на Javascript у файли script.js та підтянуті до html коду сторінки.

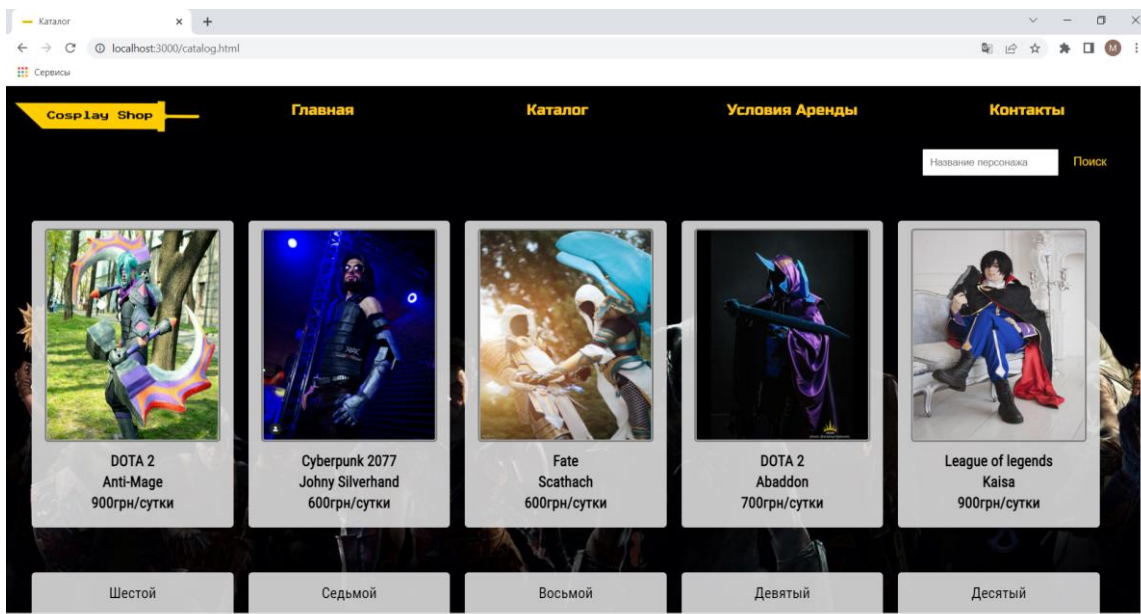


Рис. 3.14. Сторінка товарів

На сторінці відображаються продукти які достаються з бази даних через сервер, як показано на рис 3.14. Кількість продуктів в одному рядку залежить від ширини екрану, веб-сторінка автоматично підлаштовується під розмір екрану для того щоб у користувача не з'їжджали блоки та не були дуже маленькими. Вся адаптивність реалізується завдяки тегу `@media` в файлах стилю веб-додатку.

На цій сторінці можна також перейти на будь-яку іншу основну сторінку завдяки панелі зверху, обрати товар та перейти на його сторінку або скористуватись пошуком товарів, який шукає за ім'ям або джерелом (відеогра, фільм тощо).

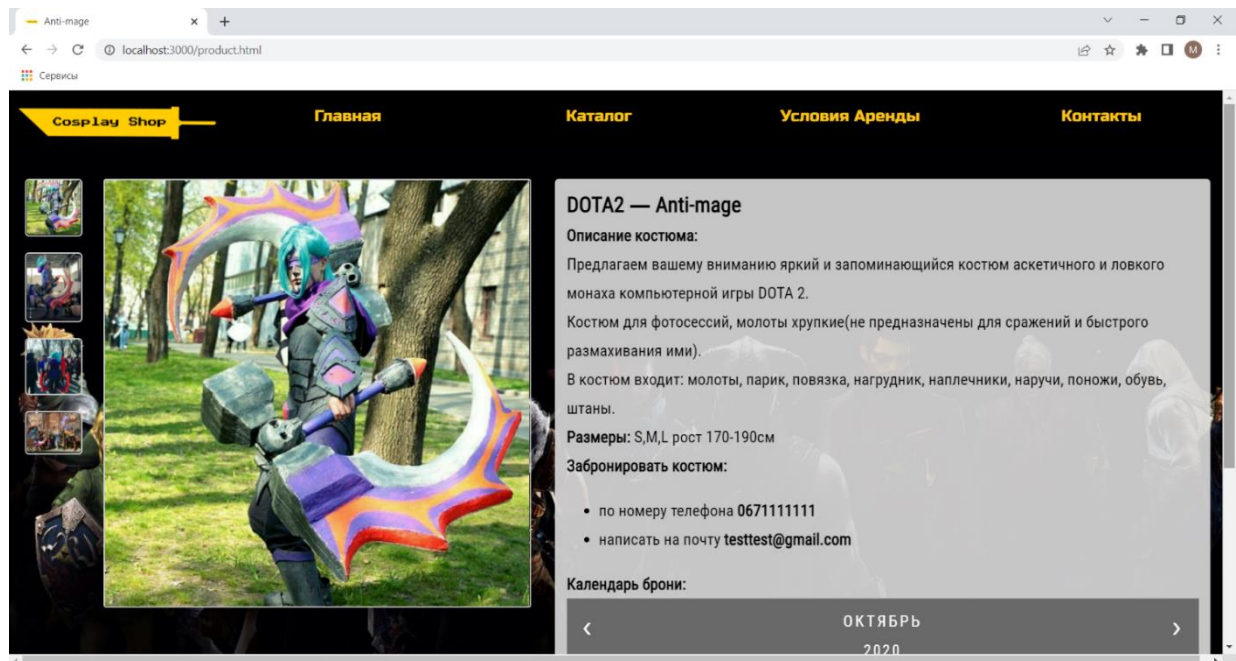


Рис. 3.15. Сторінка конкретного товару

На цій сторінці можна побачити детальний опис товару, що входить в нього, розмір, контакти та календар бронювання, який підтягується з бази даних, все це можна побачити на рис. 3.15. Також тут є список фотографій товару, при натисканні на кожну з них вона відкривається замість основної. При натисканні на основну фотографію вона зменшує або збільшує свої розміри, це зроблено для того, щоб можна було розгледіти фотографію або звільнити більше місця під опис товару та календар бронювання.

JS виявився однією з найкращих мов для створення сервісу як і для клієнтської частини, так і для серверної.

Також, Express, NodeJS та MongoDB являються перевіреними засобами та застосовуються у різних не масштабних компаніях. Також легко отримати фідбек у разі виникнення запитання на різних форумах в інтернеті.

Окрім всіх цих переваг, є декілька різних недоліків, котрі потрібно обговорити. В першу чергу, важко використовувати базою даних, так як MongoDB заснована на об'єктній моделі документа, повністю відрізняючись від стандартних реляційних баз даних. Вона не гарантує теж самий рівень технологій та можливостей, як реляційні бази даних.

Висновки до розділу 3

В цьому розділі ми розглянули логіку роботи та створення веб-сервісу. Розроблено інтерфейс для клієнтської частини та був створена серверна частина. З'єднано клієнтську та серверну частину завдяки NodeJS та технології Express. Також створена база даних MongoDB. Інтерфейс клієнтської частини розробили завдяки html, css, js

ВИСНОВКИ

Були досліджені різні можливості розробки веб-сайтів та сервісів з бронювання товару. Також був створений серверний застосунок для можливості підтримувати веб-сайт в доступі для клієнтів та виводу товарів. Було реалізовано графічні елементи, дизайн та адаптивність веб-сторінки на різних пристроях.

Головною метою дипломної роботи була розробка сервісу інтернет-магазину з оренди костюмів, покращення своїх навичок у розробці та звітуванні на мові Javascript. Також покращились навички в використуванні NodeJS, Express, MongoDB та певний час зайняло вивчення цих технологій. Були розглянуті переваги та недоліки створення веб-застосунків в яких використовувалися ці технології та порівняно з іншими подібними технологіями. Окрім цього, було оцінено швидкість роботи програми.

З досвіду накопиченого під час написання дипломної роботи я визначив, що JavaScript є однією із найкращих мов для створення веб-додатків, так як він забезпечує всі необхідні ресурси для створення веб-сервісів, такі як NodeJS для серверної частини, Express, React та Angular та інші для клієнтської сторони. Сервіс з використанням NodeJS має високу швидкість. Окрім цього, він один із найкращих для створення великих та об'ємних веб-сервісів. Значна кількість бібліотек та фреймворків NodeJS розширює можливості та спрощує їх використання, що надаються NodeJS. JavaScript дуже часто оновлюється, додаються та пришвидшуються технології.

Але JavaScript - не найбільш краща мова для створення всіх видів веб-застосунків. NodeJS не підтримує дуже важку обробку даних. З цього можна вважати, що високотехнологічні підприємства та компанії, використовуючі машинне навчання або алгоритми, не використовують цю технологію.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. JavaScript [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/JavaScript>
2. Express.js [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/Express.js>
3. Session Management in Node.js using ExpressJS and Express Session [Електронний ресурс] – Режим доступу до ресурсу: <https://www.section.io/engineering-education/session-management-in-nodejs-using-expressjs-and-express-session/>
4. Transport Layer Security (TLS) [Електронний ресурс] – Режим доступу до ресурсу: <https://www.techtarget.com/searchsecurity/definition/Transport-Layer-Security-TLS>
5. Website security [Електронний ресурс] – Режим доступу до ресурсу: https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps/Website_security
6. HTML5 Tag [Електронний ресурс] – Режим доступу до ресурсу: <https://www.geeksforgeeks.org/html5-header-tag/>
7. DOCTYPE Declaration [Електронний ресурс] – Режим доступу до ресурсу: <https://www.freecodecamp.org/news/what-is-the-doctype-declaration-in-html/>
8. A Basic Introduction to JS for Beginners [Електронний ресурс] – Режим доступу до ресурсу: <https://www.hostinger.com/tutorials/what-is-javascript>

ДОДАТОК



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
 НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
 ТЕХНОЛОГІЙ
 КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



Розробка серверної та клієнтської частини інтернет магазину мовою JavaScript

ВИКОНАВ СТУДЕНТ 4 КУРСУ

ГРУПИ ПД-42

ШЛЯМАР МАКСИМ ІГОРОВИЧ

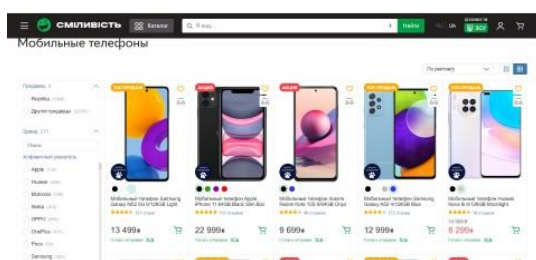
КЕРІВН ІК РОБОТИ

к.т.н, доц. ТРИНТИНА Н.А.

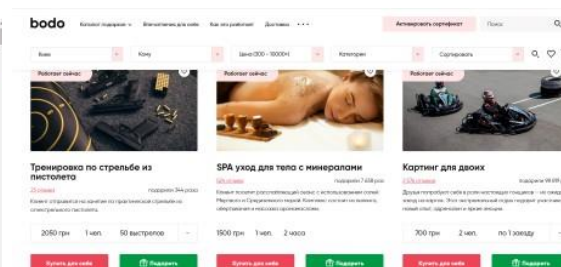
Київ – 2022

АНАЛОГІ

Інтернет-магазин Rozetka



Інтернет-магазин подарунків Bodo



ПОРІВНЯННЯ ЗАНАЛОГАМИ

	CosplayShop	Rozetka	Bodo
Адаптивність	+	+	-
Бронювання	+	-	+
Відгуки	-	+	+
Контактна інформація	+	+	+

3

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Мета роботи – Створити клієнтську та серверну частину інтернет-магазину використовуючи мову Javascript.

Об'єкт дослідження – Застосунок який дає змогу клієнтам замовляти чи бронювати товар.

Предмет дослідження – Створення інтернет магазину.

4

ТЕХНІЧНІ ЗАВДАННЯ

Дослідити можливості розробки веб-сайтів та сервісів з бронювання товару.

Розробити серверний застосунок для можливості підтримувати веб-сайт в доступі для клієнтів та виводу товарів.

Реалізувати графічні елементи, дизайн та адаптивність веб-сторінки на різних пристроях.

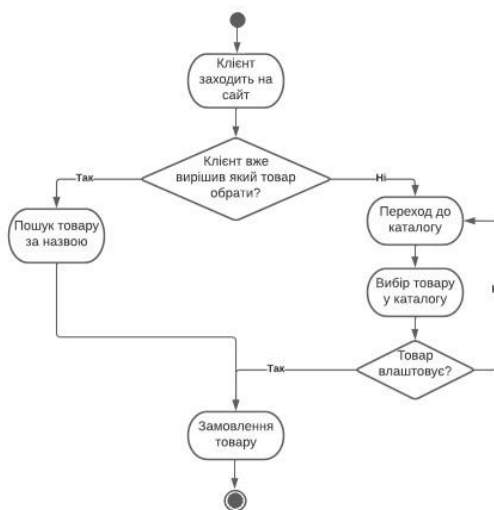
5



ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ

6

ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ



7

АРХІТЕКТУРА САЙТУ

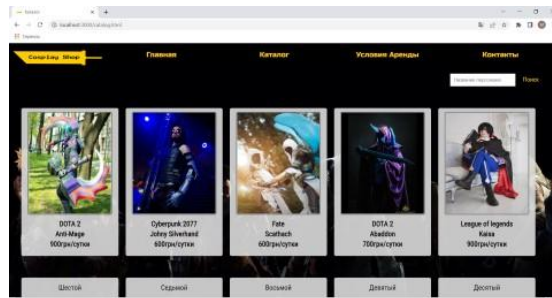


8

ЭКРАННИ ФОРМИ

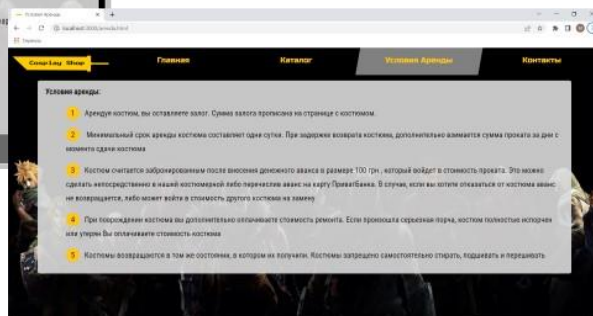
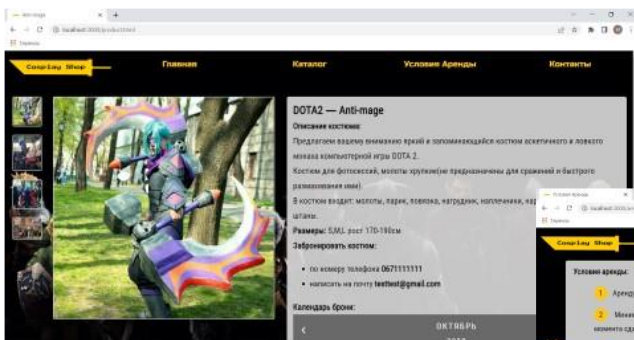
```

C:\Users\mahly>cd C:\WEB
C:\WEB>npm start
> start
> node server.js
Server started at http://localhost:3000
  
```



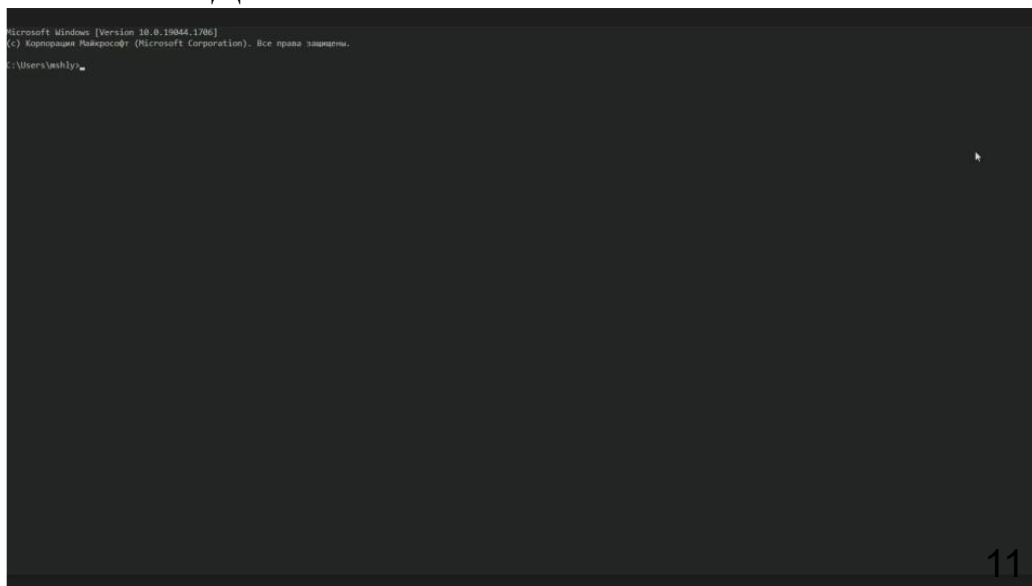
9

ЭКРАННИ ФОРМИ



10

ВІДЕО РОБОТИ ЗАСТОСУНКУ



АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

Шлямар М.І., / Науково-технічна конференція «Застосування програмного забезпечення в інфокомунікаційних технологіях». Збірник тез. – К.: ДУТ, 2022

ВИСНОВКИ

1. Досліджено можливості розробки веб-сайтів та сервісів з бронювання товару.
2. Розроблено серверний застосунок для можливості підтримувати веб-сайт в доступі для клієнтів та виводу товарів.
3. Реалізовано графічні елементи, дизайн та адаптивність веб-сторінки на різних пристроях.



ДЯКУЮ ЗА УВАГУ!