

МІНІСТЕРСТВО ОСВІТИ НАУКИ УКРАЇНИ
СТРУКТУРНИЙ ПІДРОЗДІЛ «ФАХОВИЙ КОЛЕДЖ ЕКОНОМІКИ ТА
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРАТ «ПВНЗ «ЗІЕІТ»

Циклова комісія з інформаційних технологій

ДО ЗАХИСТУ ДОПУЩЕНИЙ

Голова _____

С.І. Левицький

ВИПУСКНА РОБОТА МОЛОШОГО СПЕЦІАЛІСТА
РОЗРОБКА ВЕБСЕРВІСУ «РОЗРОБКА ІНТЕРНЕТ-МАГАЗИНУ
КОМПЛЕКТУЮЧИХ ДЛЯ ПЕРСОНАЛЬНИХ КОМП'ЮТЕРІВ»

Виконав

ст. гр. ІПЗ - 219к

Файзель О.В.

Науковий керівник

ст. викл.

К.В. Дереза

Запоріжжя

2022

СТРУКТУРНИЙ ПІДРОЗДІЛ «ФАХОВИЙ КОЛЕДЖ ЕКОНОМІКИ ТА
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРАТ «ПВНЗ «ЗІЕІТ»

Циклова комісія з інформаційних технологій

ЗАТВЕРДЖУЮ

Голова

Левицький С.І.

__ . __ . __ р.

З А В Д А Н Н Я
НА ВИПУСКНУ РОБОТУ МОЛОДШОГО СПЕЦІАЛІСТА

Студенту гр. ІПЗ – 219к

Спеціальності: «Інженерія програмного забезпечення»

Файзеля Олександра Вадимовича

_____ (прізвище, ім'я, по батькові)

1. Тема: Розробка веб сервісу «Інтернет-магазин комплектуючих для
_____ персональних комп'ютерів »

_____ затверджена наказом по коледжу № 09.2-20 від 04.03.22 р.

2. Термін здачі студентом закінченої роботи: __ . __ . __ р.

3. Перелік питань, що підлягають розробці:

1. Розглянути предметну область

_____ 2. Розглянути існуючі аналоги та сформулювати задачі проекту

_____ 3. Визначити та оглянути стек технологій

_____ 4. Визначити структуру проекту

_____ 5. Розробити програму

_____ 6. Провести тестування

_____ 7. Оформити інструкції до розгортки

8. Оформити результати роботи у вигляді звіту

4. Календарний графік підготовки випускної роботи молодшого спеціаліста

№ етапу	Зміст	Терміни виконання	Готовність по графіку (%), підпис керівника	Підпис керівника про повну готовність етапу, дата
1	Формулювання (корегування) теми випускної роботи молодшого спеціаліста та збір практичного матеріалу за темою випускної роботи	17.01.22-26.02.22		
2	I атестація I розділ випускної роботи молодшого спеціаліста	28.03.22-02.04.22		
3	II атестація II розділ випускної роботи молодшого спеціаліста	10.05.22-14.05.22		
4	III атестація III розділ випускної роботи молодшого спеціаліста, висновки та рекомендації, додатки, реферат	30.05.22-04.06.22		
5	Перевірка випускної роботи програмою «Антиплагіат»	30.05.22-18.06.22		
6	Доопрацювання випускної роботи молодшого спеціаліста, підготовка презентації, отримання відгуку керівника і рецензії	06.06.22-11.06.22		
7	Попередній захист випускної роботи молодшого	14.06.22-18.06.22		
8	Подача випускної роботи молодшого спеціаліста на кафедру	за 3 дні до захисту		
9	Захист випускної роботи молодшого спеціаліста	20.06.22-25.06.22		

Керівник

К.В. Дереза

(підпис)

(прізвище та ініціали)

« ____ » _____ 2022 р.

Завдання отримав до виконання

О.В. Файзель

« ____ » _____ 2022 р.

РЕФЕРАТ

Випускна робота молодшого спеціаліста містить: 63 с., 26 рис., 2 додатка, 25 джерел, 2 таблиці.

Мета даної роботи полягає в розробці інтернет – магазину комплектуючих для персональних комп'ютерів.

Результатом виконаної роботи буде розроблений інтернет – магазин комплектуючих для персональних комп'ютерів.

Пояснювальна записка складається з вступу, трьох основних розділів та висновку.

У вступі ставиться мета завдання, розглядається поняття електронної комерції.

У першому розділі вивчаємо історію виникнення інтернет – магазинів, розглядаємо переваги та недоліки електронної комерції та розбираємо типові інтернет – магазини.

У другому розділі знайомимося з інструментами які допомагають в розробці інтернет – магазину.

В третьому розділі описуємо розроблений інтернет – магазин та робимо висновки.

У висновку написано про виконану роботу та розроблений інтернет магазин комплектуючих для комп'ютерів.

ЗМІСТ

СПИСОК УМОВНИХ ПОЗНАЧЕНЬ.....	7
ВСТУП.....	8
РОЗДІЛ 1.....	10
ІНТЕРНЕТ – МАГАЗИН ЯК ФОРМА ЕЛЕКТРОННОЇ ТОРГІВЛІ.....	10
1.1 Історія виникнення інтернет – магазинів.....	10
1.2 Переваги та недоліки інтернет – магазинів.....	13
1.2.1 Переваги інтернет – магазину.....	13
1.2.2 Недоліки інтернет – магазину.....	15
1.3 Класифікація інтернет-магазинів.....	16
1.3.1 Класифікація за моделлю інтернет – магазину.....	16
1.3.1.1 Business-to-business (B2B).....	16
1.3.1.2 Business-to-consumer (B2C).....	17
1.3.1.3 Consumer-to-consumer (C2C).....	17
1.3.1.4 Consumer-to-business (C2B).....	17
1.3.2 Класифікація за типом інтернет – магазину.....	18
1.3.2.1 Сайт одного бренду.....	18
1.3.2.2 Інтернет магазини.....	19
1.3.2.3 Торгові майданчики (Маркетплейси).....	19
1.4 Огляд типових інтернет-магазинів.....	20
1.4.1 Інтернет – магазин IT – Blok.....	20
1.4.2 Інтернет – магазин CompX.....	21
1.4.3 Інтернет – магазин Setevuha.....	22
РОЗДІЛ 2.....	23
ІНСТРУМЕНТАЛЬНА ЧАСТИНА ІНТЕРНЕТ МАГАЗИНУ	
КОМП’ЮТЕРНИХ КОМПЛЕКТУЮЧИХ.....	23
2.1 Інтегроване середовище розробки сайтів IDE.....	23

2.1.1 IDE WebStorm.....	25
2.2 Мова програмування JavaScript.....	27
2.3 Мова програмування TypeScript.....	29
2.4 Порівняння JavaScript та TypeScript.....	32
2.5 Бібліотека React.....	33
2.5.1 JSX.....	35
2.6 Мова розмітки HTML.....	36
2.7 Мова стилів CSS.....	37
2.8 Система управління базами даних MySQL.....	39
2.8.1 phpMyAdmin.....	40
2.9 API.....	41
РОЗДІЛ 3.....	44
РОЗРОБКА ІНТЕРНЕТ – МАГАЗИНУ КОМПЛЕКТУЮЧИХ ДЛЯ ПЕРСОНАЛЬНИХ КОМП’ЮТЕРІВ.....	44
3.1 Розробка інтернет – магазину комплектуючих.....	44
3.2 Опис інтернет – магазину.....	44
3.2.1 Головний екран.....	44
3.2.2 Екран товарів.....	47
3.2.3 Екран кошику.....	50
3.2.4 Екран контактів.....	51
ВИСНОВКИ.....	52
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	53
ДОДАТОК.....	55
ДОДАТОК А.....	56
ДОДАТОК Б.....	59

СПИСОК УМОВНИХ ПОЗНАЧЕНЬ

Слово / словосполучення	Скорочення	Умови використання
A		
Application Programming Interface	API	
C		
Continuous Integration Continuous Delivery	CI/CD	
Cascading Style Sheets	CSS	
E		
Electronic Data Interchange	EDI	
G		
Graphical User Interface	GUI	
H		
HyperText Markup Language	HTML	
HyperText Transfer Protocol	HTTP	
I		
Integrated Drive Electronics	IDE	
J		
JavaScript XML	JSX	
S		
Single Page Application	SPA	
Secure Sockets Layer	SSL	
U		
United Parcel Service	UPS	
X		
Extensible Markup Language	XML	

ВСТУП

В даний час відбувається глобальне впровадження інтернет-технологій у всі сфери життя, що у свою чергу формує глобальну інформаційну економіку.

Важливим завданням розвитку країни стає перехід до інформаційної економіки, в якій велика увага приділяється науково-технічному прогресу, масовому поширенню інноваційних технологій та електронному управлінню.

Світова економіка та її головні тенденції пов'язані з сьогоdnішнім формуванням інтернет-торгівлі. Ця тенденція обумовлена низкою факторів: забезпечується розширене відтворення на новій матеріальній основі, надається доступ до інформації та знанням, що трансформуються у товарні форми, в умовах розвитку електронної комерції відбувається формування фінансової економіки. Відмінна риса, якою полягає в тому, що максимізація прибутку відбувається без будь-якого виробництва, тобто підвищується важливість фінансового капіталу. Все вищезгадане характеризує той факт, що відбувається перетворення економічних відносин у новому постіндустріальному рівні. В наш час інтернет-торгівля набирає величезної популярності.

Насамперед інтернет-торгівля — це сфера економіки, яка включає у собі фінансові та торгові транзакції, що здійснюються у свою чергу за допомогою комп'ютерних мереж, та бізнес-процеси, пов'язані з проведенням таких транзакцій. Інтернет став важливим компонентом для ведення успішного бізнесу, у будь-яких організацій чи компаній. Інтернет — це потужний інструмент для дослідження, розвитку торгівлі та ведення бізнесу, впливаючи на аудиторію дистанційно, при цьому немає жодних територіальних обмежень. На сьогоднішній день понад 95% користувачів використовують Інтернет для пошуку інформації, покупки товарів та послуг, створення власних бізнес-проектів, для навчання, для роботи, для розваг і т.д. Інтернет комерція все більше і більше входить у наше життя. Вона відкриває

нові можливості для ведення бізнесу, і більшість компаній вже перейшли на ведення та розвиток свого відносини в Інтернет-середовищі.

Мета даної роботи розробка інтернет – магазину комплектуючих для персональних комп'ютерів.

Завдання для даної роботи:

- Знайти матеріал для розробки інтернет – магазину.
- Знайти та ознайомитись з відповідними інструментами розробки.
- Розробити план розробки інтернет – магазину.
- Розробити інтернет – магазину комплектуючих для персональних комп'ютерів.
- Протестувати інтернет – магазин на помилки та працездатність.
- Написати кваліфікаційну роботу та зробити висновки.

Результатом виконаної роботи буде розроблений інтернет – магазин комплектуючих для персональних комп'ютерів.

РОЗДІЛ 1

ІНТЕРНЕТ – МАГАЗИН ЯК ФОРМА ЕЛЕКТРОННОЇ ТОРГІВЛІ

1.1 Історія виникнення інтернет – магазинів

Історія електронної комерції почалася понад 40 років тому, коли впровадження ранніх технологій, таких як електронний обмін даними (EDI) та телеторгівля у 1970-х роках, проклало шлях для сучасного магазину електронної комерції, яким ми його знаємо сьогодні [1].

Історія електронної комерції тісно переплетена з історією Інтернету. Інтернет-магазини стали можливими, коли Інтернет був відкритий для публіки в 1991 році. Amazon був одним з перших сайтів електронної комерції в США, який почав продавати товари в Інтернеті, і з того часу за ним пішли тисячі компаній.

Онлайн-покупки – це процес пошуку та купівлі товарів чи послуг через Інтернет. Найраніші інтернет-магазини розпочали свою діяльність у 1992 році, а роздрібна онлайн-торгівля зайняла значний сегмент роздрібного ринку протягом першого десятиліття двадцятого століття, коли кількість власників персональних комп'ютерів збільшилася, і авторитетні роздрібні торговці почали пропонувати свої товари через Інтернет [1].

Ідея онлайн-покупок виникла ще до появи всесвітньої мережі. Технологія обробки транзакцій в реальному часі з домашнього телебачення на основі відеотексту була вперше продемонстрована в 1979 Майклом Олдрічем, який спроектував і встановив системи у Великобританії, включаючи першу пілотну систему Tesco в 1984. Перша комп'ютерна мережу B2B створили компанією Thomson Holidays в 1981 году.

В 1990 Тім Бернерс-Лі створив перший сервер і браузер World Wide Web. У 1992 році Чарльз Стек створив перший книжковий інтернет магазин Book Stacks Unlimited (Books.com), за два роки до того, як Джефф Безос запустив Amazon.com (Рис. 1.1). В 1994 відбулися й інші успіхи, такі як

онлайн-банкінг та відкриття інтернет-магазину піци Pizza Hut. У тому ж році Netscape представила SSL-шифрування (Secure Sockets Layer) даних, що передаються в Інтернеті, що стало необхідним для безпечних покупок в Інтернеті. В 1995 Amazon розширив свої онлайн-магазини, а в 1996 з'явився eBay.



Welcome to Amazon.com Books!

*One million titles,
consistently low prices.*

(If you explore just one thing, make it our personal notification service. We think it's very cool!)

SPOTLIGHT! -- AUGUST 16TH

These are the books we love, offered at Amazon.com low prices. The spotlight moves EVERY day so please come often.

ONE MILLION TITLES

Search Amazon.com's [million title catalog](#) by author, subject, title, keyword, and more... Or take a look at the [books we recommend](#) in over 20 categories... Check out our [customer reviews](#) and the [award winners](#) from the Hugo and Nebula to the Pulitzer and Nobel... and [bestsellers](#) are 30% off the publishers list..

Рис. 1.1 – Один із перших інтернет – магазинів

Більшість перших онлайн-покупців були молодими освіченими чоловіками, знайомими з комп'ютерними технологіями, але 2001 року жінки становили 52,8% онлайн-населення. Інтернет-магазини привернули увагу широкої публіки до 1999 року, як інтернет-стартапи, так і відомі роздрібні торговці відкрили веб-сайти, що пропонують їх товари. Під час сезону різдвяних покупок 1999 року, коли багато споживачів вперше спробували зробити покупки в Інтернеті, роздрібні продавці виявилися не готовими до ефективної обробки та доставки своїх замовлень. Інтернет-магазини покращили обслуговування клієнтів, а транспортні компанії, такі як FedEx та

UPS, розширили свою діяльність, щоб впоратися зі зростаючим трафіком. До грудня 2008 року багато інтернет-магазинів змогли збільшити свої продажі, запропонувавши покупцям в останній момент напередодні Різдва гарантовану доставку вночі [3].

Під час «бульбашки доткомів» 1999–2001 років хвилювання щодо потенціалу онлайн-торгівлі призвело до нереалістичних очікувань бізнесу. Стартапи намагалися продавати такі продукти, як бакалія та корм для собак, через Інтернет, не беручи до уваги надмірно високі витрати на утримання складів та систем доставки. Проте компанії, які мають відкриті роздрібні магазини та продавці спеціалізованих товарів, змогли розширити свою клієнтську базу, використовуючи вже наявну в них інфраструктуру.

Для покупок в Інтернеті потрібен доступ до комп'ютера та будь-яка форма оплати, така як банківський рахунок та дебетова картка, кредитна картка або рахунок PayPal. Згідно з дослідженням, опублікованим в Journal of Electronic Commerce, чим вищий рівень освіти, доходу та роду занять глави домогосподарства, тим сприятливіше сприймаються покупки поза магазинами. Більш широке знайомство з технологіями підвищує можливість розвитку сприятливого ставлення до нових каналів покупок. Покупці в Інтернеті, швидше за все, належать до середнього та вищого класу, але в міру того, як розвиток технологій зробило комп'ютери менш дорогими та доступними для більшої кількості людей, а також спростило підключення до Інтернету, клієнтська база розширилася [4].

Популярність онлайн-покупок – явище глобальне. Опитування інтернет-користувачів показали, що 99 відсотків південнокорейських інтернет-користувачів робили покупки в Інтернеті, за ними слідує 97 відсотків інтернет-користувачів у Німеччині, Японії та Великій Британії. 94% інтернет-користувачів США повідомили, що вони щось купували в Інтернеті. Використання кредитної картки для покупки товарів в Інтернеті особливо привабливе для споживачів на ринках, що розвиваються, які не можуть легко знайти або купити потрібні їм товари в місцевих роздрібних магазинах.

1.2 Переваги та недоліки інтернет – магазинів

1.2.1 Переваги інтернет – магазину

Є ряд помітних і не настільки очевидних переваг ведення бізнесу на онлайн – платформі. Зараз ми озвучимо ці переваги:

1. Низькі фінансові витрати. Однією з переваг електронної комерції є нижча вартість запуску. Фізичні роздрібні магазини мають платити до десяти тисяч гривень за оренду одного зі своїх магазинів. Вони також мають кілька початкових витрат, таких як вивіски магазину, дизайн магазину, покупка інвентарю, торгове обладнання та багато іншого. І давайте не забуватимемо, що власник магазину також повинен наймати персонал для роботи та управління кожним магазином. Їм також може знадобитися найняти працівників служби безпеки в залежності від вартості продукту в магазині. Тим не менш, для магазину прямого постачання електронної комерції стартові витрати оцінюються в тисячу гривень. Залежно від того, який тип бізнесу електронної комерції ви ведете, вам може знадобитися наймати співробітників лише тоді, коли досягнете певного рівня. Оскільки ваш інтернет-магазин знаходиться в мережі, ці співробітники можуть працювати віддалено, що полегшує пошук персоналу, який підійде вашій компанії. Якщо ви оберете пряме постачання через Інтернет, вам не потрібно буде купувати оптом інвентар, що заощадить вам багато грошей [5].
2. Потенційний дохід. Ще однією перевагою електронної комерції є те, що інтернет магазини завжди відкриті для клієнтів. За допомогою реклами у Facebook ви можете залучити когось цілодобово. Поміркуйте про свою аудиторію, яка не працює у звичайний час або надто зайнята, щоб зайти в магазин, щоб щось купити. Інтернет-магазин дозволяє вам залучати тих, хто має нестандартний графік роботи або хто не має часу робити покупки особисто.

3. Продавати на міжнародному рівні. Наступним у списку переваг електронної комерції є те, що новий бренд може легко продаватися клієнтам у всьому світі. Ви можете дізнатися свою аудиторію, де б вона не знаходилася: у Великій Британії, Південній Америці або сусідніх країнах. Якщо ви вибираєте пряму доставку з AliExpress, багато продуктів пропонують доступну доставку ePacket або безкоштовну доставку. Це дозволяє вам встановлювати конкурентоспроможні ціни та постачати свою продукцію для світової аудиторії. Продаж по всьому світу - це великий подвиг, оскільки він допомагає вам створювати свій бренд набагато швидше, експоненційно розширює ваш ринок і дозволяє вам отримувати прибуток набагато раніше, ніж ваші місцеві конкуренти.
4. Доступні працівники. Однією з переваг електронної комерції є доступність найму співробітників, і, як згадувалося раніше, ви можете наймати їх із будь-якої точки світу. Ви можете передати роботу віртуальним помічникам у країнах, де вартість життя набагато нижча. Вам знадобиться менше співробітників у онлайн-бізнесі, ніж у роздрібній торгівлі. Величезною перевагою онлайн-бізнесу є те, що вам не потрібно наймати працівників під час запуску. Ви можете почати та вести бізнес електронної комерції самостійно. Тільки коли ви починаєте рости або коли у вас є капітал, ви можете почати створювати команду.
5. Легше стимулювати імпульсну покупку. Ще одна перевага електронної комерції полягає в тому, що ви можете отримати вигоду з імпульсних покупок, оскільки онлайн-покупці схильні до цього. Якщо у вас є приваблива фотографія товару з яскравими квітами чи людськими емоціями, ви можете створити рекламу, яка стимулює імпульсні покупки. Ви також можете використовувати ряд тактик дефіциту, таких як таймери зворотного відліку або демонстрація обмеженої кількості, щоб спонукати людей купувати. Крім того, якщо у вашому інвентарі є

новий предмет або предмет обмеженого випуску, подумайте, щоб протестувати його на предмет імпульсних покупок, щоб побачити, як люди відреагують на нього.

1.2.2 Недоліки інтернет – магазину

Нижче наведено кілька недоліків електронної комерції, із якими зазвичай стикаються продавці. Зараз ми озвучимо недоліки:

1. Ніхто не може купувати під час краху сайту. Найгірший недолік електронної комерції — це коли ніхто не може купити у вашому магазині, якщо ваш сайт вийде з ладу. Ось чому важливо переконатись, що ваш сайт розміщений на правильній платформі.
2. Клієнти не можуть спробувати перед покупкою. Хоча в даний час це є проблемою для багатьох ритейлерів, це не буде проблемою в довгостроковій перспективі. З доповненою реальністю все більше магазинів починають додавати елементи AR до своїх магазинів, щоб покупці могли приміряти товари. Компанії, що займаються електронною комерцією з доповненою реальністю, пропонують бізнес-рішення, щоб створити більш інтерактивний досвід з вашими клієнтами.
3. Електронна комерція висококонкурентна. Боротьба за клієнтів — ще один із найгірших недоліків електронної комерції. Реальність така, що найкращі ніші часто є найбільш конкурентоспроможними. Ось чому так багато підприємців тягнуться до них. Проте чим конкурентоспроможніша ніша, тим дорожча реклама для цієї ніші. Є кілька способів оминати це. По-перше, ви можете використовувати іншу маркетингову стратегію, ніж ваші конкуренти.
4. Клієнти можуть бути нетерплячими. Якщо покупець має питання в магазині, продавець готовий відповісти на нього. Однак недоліком електронної комерції є те, що більшість підприємств стикаються із

затримкою, відповідаючи на запити клієнтів. Реальність така, що більшість клієнтів очікують на відповідь від компанії протягом години в соціальних мережах. Якщо ви затримаєтеся з відповіддю на їх повідомлення, вони можуть роздратуватися і замість цього робити покупки в іншому місці. Ви повинні бути онлайн 24/7.

5. Час доставки може бути тривалим. Клієнти вважають час доставки одним із найгірших недоліків електронної комерції. Коли клієнт робить покупки особисто, може відразу забрати товар додому. Але при покупках в Інтернеті більшість клієнтів одержують свої товари протягом тижня або більше.
6. Фізична роздрібна торгівля все ще популярніша, незважаючи на спад. Незважаючи на те, що однією з переваг електронної комерції є її зростання, фізичний ритейл, як і раніше, займає більшу частину ринку.

1.3 Класифікація інтернет-магазинів

Класифікувати електронні магазини можна за різними критеріями.

1.3.1 Класифікація за моделлю інтернет – магазину

Найцікавішою класифікацією є класифікація за моделлю бізнесу.

1.3.1.1 Business-to-business (B2B)

Модель електронної комерції B2B структурована саме так, як здається. Це місце, де підприємства продають товари іншим компаніям. Типи продуктів включають все, що покращує ділову практику іншої компанії. Одним із найкращих прикладів бізнесу B2B є компанія-розробник програмного забезпечення. Згадайте Salesforce, HubSpot або SurveyMonkey.

Всі ці продукти призначені для бізнесу, і для типового сімейного споживача немає сенсу вкладати кошти в корпоративне програмне забезпечення [5].

1.3.1.2 Business-to-consumer (B2C)

Business-to-consumer (B2C). Модель електронної комерції "бізнес-споживач" також проста. Це коли бізнес продає товари через Інтернет або мобільний пристрій безпосередньо споживачам. Компанії електронної комерції, орієнтовані на споживача, - це компанії, які ви і я відвідуємо онлайн кожен день, щоб купити такі речі, як одяг, книги, косметика і т.д. До провідних компаній електронної комерції B2C входять такі гіганти, як Amazon, eBay та Alibaba. Також важливо пам'ятати, що компанії B2C не обов'язково мають існувати виключно в Інтернеті, як Amazon та eBay. Компанії B2C можуть мати звичайний магазин і, якщо це має сенс, можуть запустити веб-сайт електронної комерції. Прикладами цього є великі роздрібні торговці модним одягом, такі як Nordstrom, J. Crew, H&M та інші [5].

1.3.1.3 Consumer-to-consumer (C2C)

Consumer-to-consumer (C2C). Бізнес-модель «споживач-споживачеві» - це коли один споживач виробляє та продає продукт іншому споживачеві. Один споживач може використовувати таку платформу, як Amazon, Etsy або eBay для продажу товарів іншому споживачеві. За допомогою онлайн-інструментів електронної комерції, таких як WooCommerce, споживачі можуть навіть налаштувати магазин C2C на власному веб-сайті [5].

1.3.1.4 Consumer-to-business (C2B)

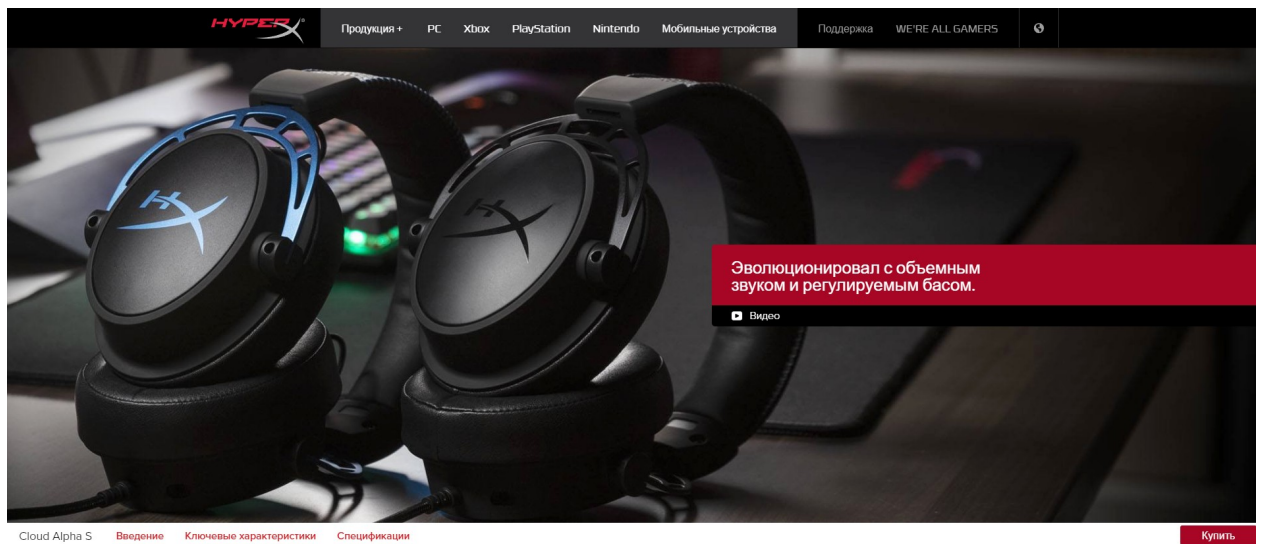
Consumer-to-business (C2B). Останній тип бізнес-моделі електронної комерції це споживач-бізнес. Саме тут споживач чи індивідуальний підприємець продає свої товари чи послуги підприємствам. Хорошими прикладами бізнесу C2B можуть бути бізнес-консультант, позаштатний графічний дизайнер або впливова людина у соціальних мережах із великою аудиторією [5].

1.3.2 Класифікація за типом інтернет – магазину

Незалежно від того, який тип бізнес-моделі електронної комерції ви оберете, всі вони мають одну спільну рису. Усі магазини електронної комерції працюють у Інтернеті. Це означає, що вам потрібний сайт. Ось різні типи веб-сайтів електронної комерції, які слід враховувати.

1.3.2.1 Сайт одного бренду

Веб-сайт індивідуального бренду – це сайт, на якому продаються товари чи послуги лише одного підприємства (Рис. 1.2) чи особи. Це також найбільш поширений тип веб-сайту для людей, які відкривають магазин електронної комерції.



Игровая гарнитура HyperX Cloud Alpha S

Рис. 1.2 – Приклад сайту HyperX

1.3.2.2 Интернет магазини

Наступний тип веб-сайту електронної комерції також поширений, особливо великих роздрібних продавців. Інтернет-магазини — це більші магазини, які зазвичай продають товари інших брендів.

TELEMART.UA (Рис. 1.3) – компанія, яка займається комп'ютерами з 2013 року. З 2019 року вони повністю сконцентрувалися на напрямку продажу ПК і всього, що з ними пов'язано. Продають комплектуючі різних виробників.

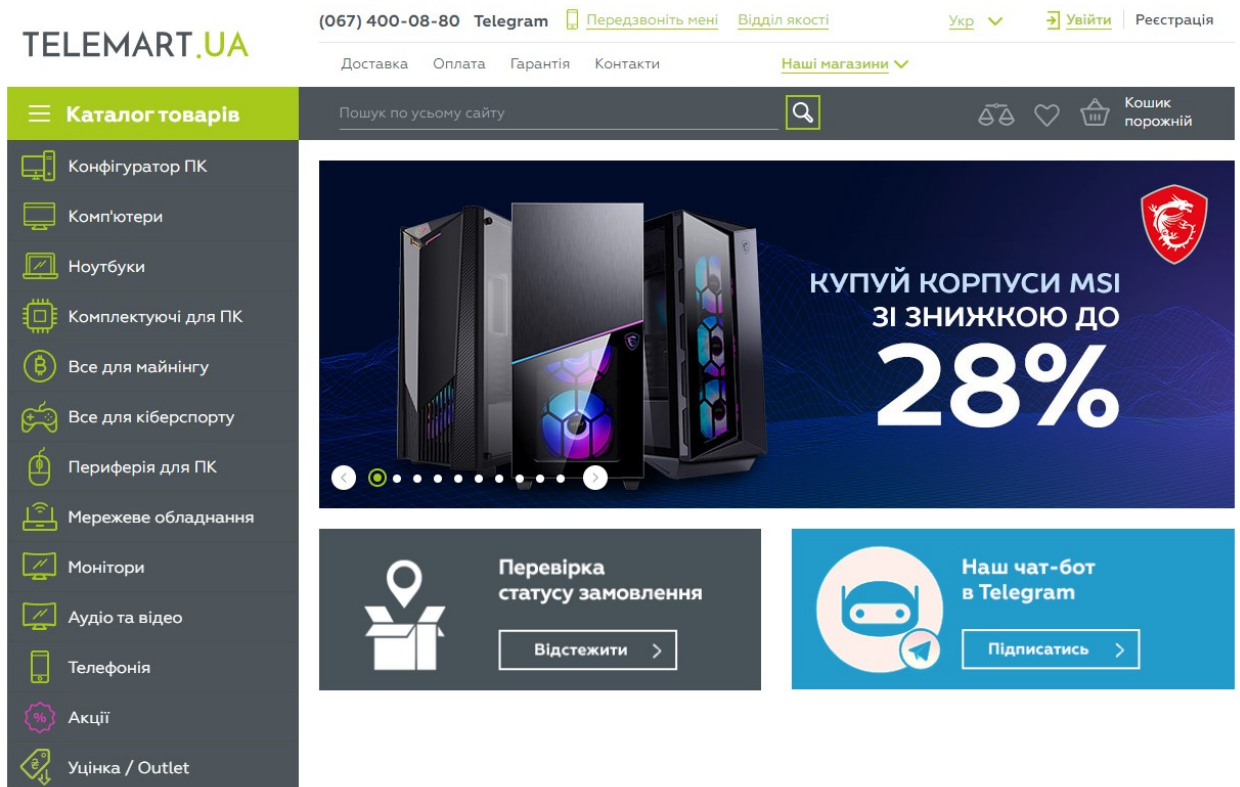


Рис. 1.3 – Приклад інтернет – магазину TELEMART

1.3.2.3 Торгові майданчики (Маркетплейси)

Інший тип веб-сайту електронної комерції – це торговий майданчик. Маркетплейс — це веб-сайт, де індивідуальні підприємці можуть розміщувати свої товари для продажу. Найпопулярніші приклади включають Rozetka та Prom. Rozetka та Prom надають продавцям онлайн-ринок для розміщення та продажу своєї продукції.

Rozetka (Рис. 1.4) це один із найбільших маркетплейсів в Україні. Спочатку Rozetka була невеликим інтернет магазином, який продавав товари різних брендів. Згодом Rozetka дозволила іншим людям виставляти свої товари на продаж, таким чином з інтернет магазину перетворилася на маркетплейс.

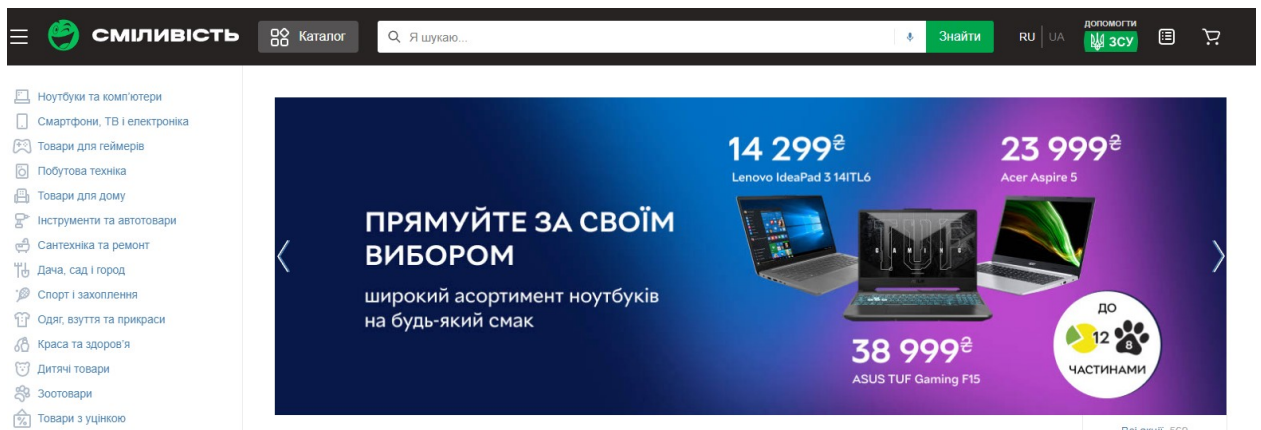


Рис. 1.4 – Маркетплейс Rozetka

1.4 Огляд типових інтернет-магазинів

1.4.1 Інтернет – магазин IT – Blok

Інтернет магазин IT – Blok (Рис. 1.5) займається продажем комп'ютерних комплектуючих та збиранням ігрових комп'ютерів в Україні. На сайті представлений великий вибір комплектуючих. Також можна зібрати комп'ютер онлайн через конфігуратор збирання комп'ютера.

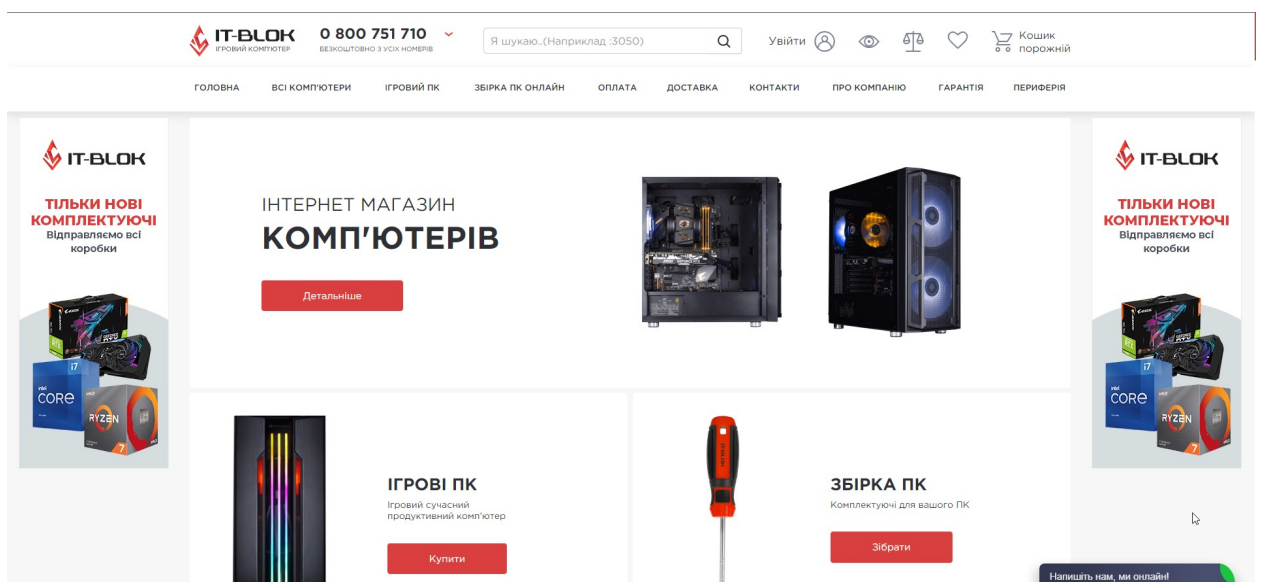


Рис. 1.5 – Інтернет – магазин IT – Blok

1.4.2 Інтернет – магазин CompX

CompX (Рис. 1.6) – це сучасний інтернет – магазин, де можна замовити найактуальніші моделі комп'ютерів, аксесуарів та периферії. В даному інтернет магазині можна знайти індивідуальні складання комп'ютера або замовити складання під себе.

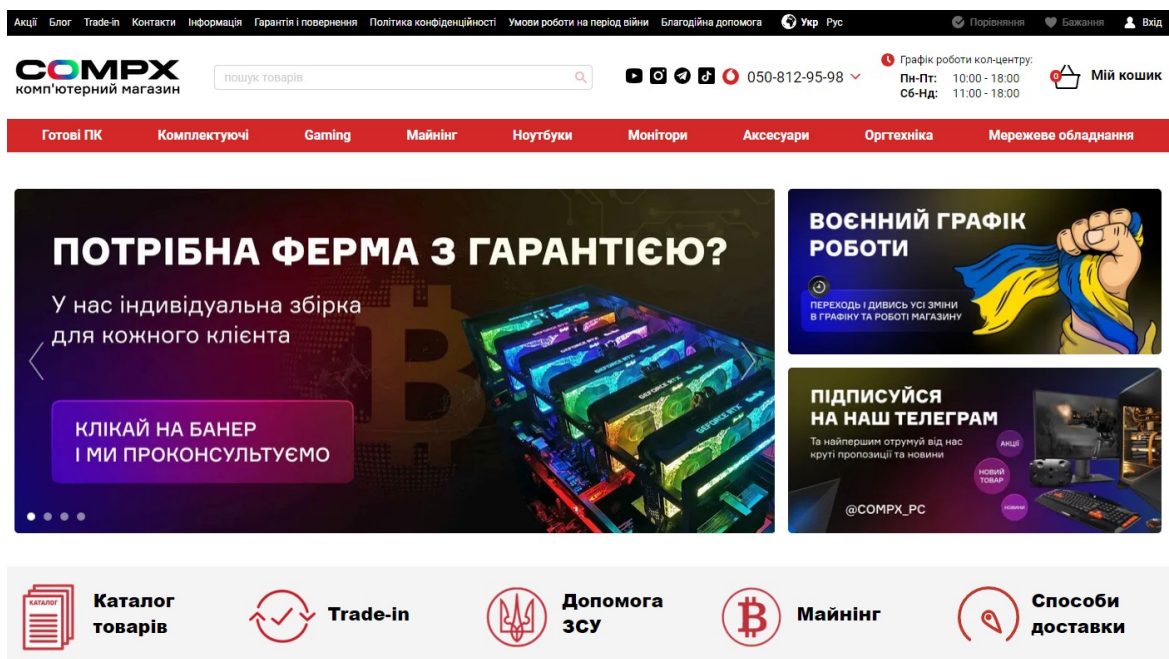


Рис. 1.6 – Інтернет – магазин CompX

1.4.3 Інтернет – магазин Setevuha

Setevuha (Рис. 1.7) – інтернет-гіпермаркет ІТ, з дуже великим вибором обладнання для комп'ютера. На сайті є онлайн допомога, яка зможе допомогти у виборі потрібного товару. Також на сайті можна знайти не тільки аксесуари для комп'ютера, а ще багато чого цікавого з техніки: телевізори, побутова техніка та інше.

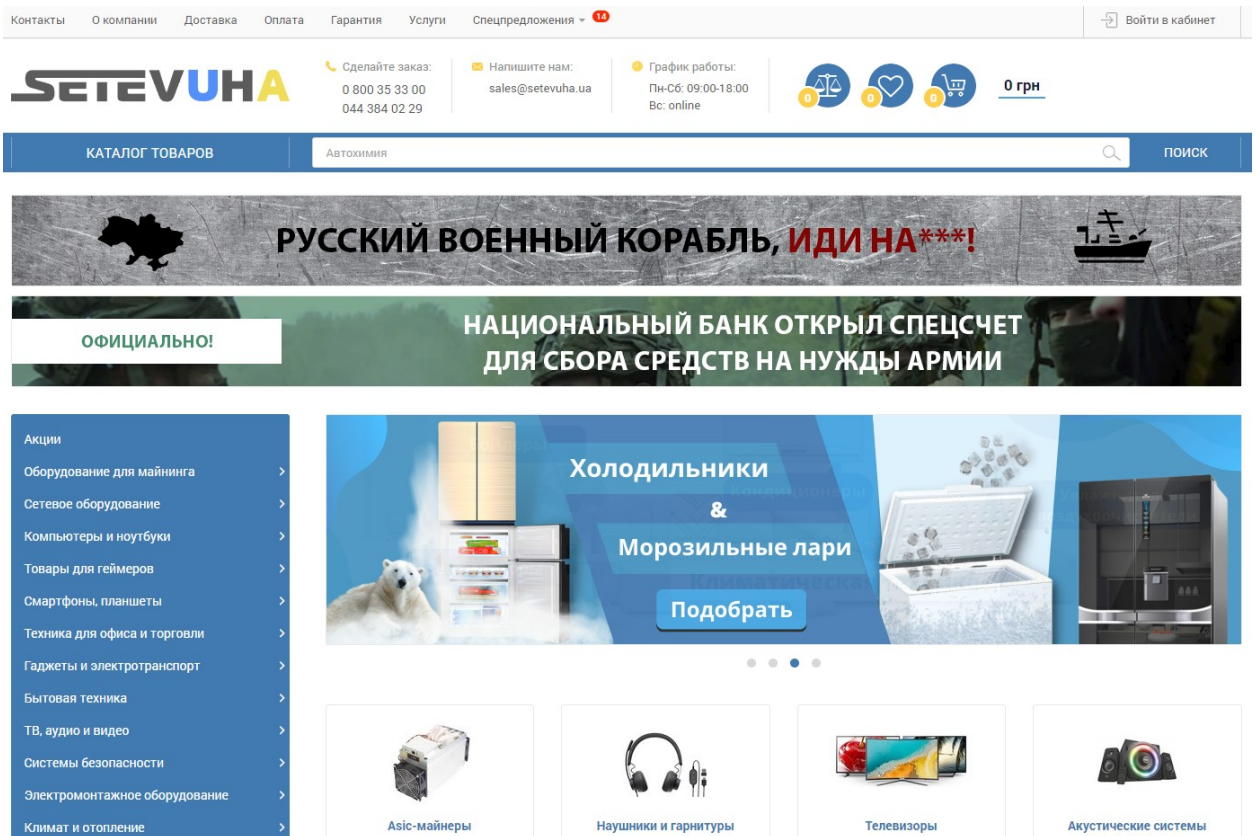


Рис. 1.7 – Интернет – магазин Setevuha

У цьому розділі ми вивчили історію виникнення інтернет-магазину. Ознайомилися з першими інтернет-магазинами як вони створювалися і який мали зовнішній вигляд. Розглянули переваги інтернет-магазинів та вивчили бізнес моделі. Також розібрали типові інтернет-магазини.

РОЗДІЛ 2

ІНСТРУМЕНТАЛЬНА ЧАСТИНА ІНТЕРНЕТ МАГАЗИНУ КОМП'ЮТЕРНИХ КОМПЛЕКТУЮЧИХ

2.1 Інтегроване середовище розробки сайтів IDE

Інтегроване середовище розробки (IDE) – це програмне забезпечення для створення програм (сайтів), яке поєднує звичайні інструменти розробника в єдиний графічний інтерфейс користувача (GUI) [8].

IDE зазвичай складається з:

- Редактор вихідного коду: текстовий редактор, який може допомогти в написанні програмного коду з такими функціями, як підсвічування синтаксису за допомогою візуальних підказок, забезпечення автоматичного заповнення для певної мови та перевірки помилок під час написання коду.
- Автоматизація локальної збірки: утиліти, які автоматизують прості, повторювані завдання в рамках створення локальної збірки програмного забезпечення для використання розробником, наприклад, компіляція вихідного коду комп'ютера в двійковий код, упаковка двійкового коду та запуск автоматизованих тестів.
- Відладчик: програма для тестування інших програм, яка може графічно відображати розташування помилки в оригінальному коді.

IDE дозволяє розробникам швидко починати програмувати нові програми, оскільки не потрібно вручну конфігурувати та інтегрувати кілька утиліт як частину процесу налаштування. Розробникам також не потрібно витратити години окремо, вивчаючи, як використовувати різні інструменти, коли кожна утиліта представлена на одному робочому місці. Це може бути особливо корисно для залучення нових розробників, які можуть покладатися на IDE, щоб освоювати стандартні інструменти та робочі процеси команди. Насправді більшість функцій IDE призначені для економії часу, наприклад інтелектуальне завершення коду та автоматичне генерування коду, що усуває необхідність вводити повні послідовності символів.

Інші поширені функції IDE покликані допомогти розробникам організувати свій робочий процес і вирішувати проблеми. IDE аналізують код так, як він написаний, тому помилки, спричинені людьми, визначаються в режимі реального часу. Оскільки утиліти представлені одним графічним

інтерфейсом, розробники можуть виконувати дії без перемикання між додатками. Підсвічування синтаксису також поширене в більшості IDE, яке використовує візуальні підказки для розрізнення граматики в текстовому редакторі. Деякі IDE додатково включають браузері класів та об'єктів, а також діаграми ієрархії класів для певних мов [8].

Можна розробляти додатки без IDE або для кожного розробника по суті створювати власну IDE, вручну інтегруючи різні утиліти з легким текстовим редактором. Для деяких розробників перевагою цього підходу є ультра налаштування та контроль, які він пропонує. Однак у контексті підприємства економія часу, стандартизація середовища та можливості автоматизації сучасних IDE зазвичай переважають інші міркування.

Сьогодні більшість корпоративних команд розробників вибирають попередньо налаштовану IDE, яка найкраще підходить для їх конкретного випадку використання, тому питання полягає не в тому, чи використовувати IDE, а в тому, яку IDE вибрати.

Існує багато різних технічних та ділових варіантів використання IDE, що також означає, що на ринку є багато власних варіантів IDE з відкритим кодом. Як правило, найважливішими відмінними характеристиками між IDE є:

- Кількість підтримуваних мов: деякі IDE призначені для однієї мови, тому вони краще відповідають певній парадигмі програмування. IntelliJ, наприклад відомий насамперед як Java IDE. Інші IDE мають широкий набір підтримуваних мов в одному, наприклад Eclipse IDE, яка підтримує Java, XML, Python та інші.
- Підтримувані операційні системи: операційна система розробника обмежує, які IDE є життєздатними (якщо IDE не є хмарним), і якщо програма, що розробляється, призначена для кінцевого користувача з конкретною операційною системою (наприклад, Android або iOS), це може бути додатковим обмеженням.

- Функції автоматизації: незважаючи на те, що більшість IDE включають 3 ключові функції текстового редактора, автоматизації збірки та налагодження, багато з них включають підтримку додаткових функцій, таких як рефакторинг, пошук коду та інструменти безперервної інтеграції та безперервного розгортання (CI/CD).
- Вплив на продуктивність системи: слід враховувати обсяг пам'яті IDE, якщо розробник хоче одночасно запускати інші програми, що інтенсивно витрачають пам'ять.
- Плагіни та розширення: деякі IDE включають можливість налаштовувати робочі процеси відповідно до потреб і вподобань розробника.

2.1.1 IDE WebStorm

WebStorm від JetBrains – це інтегроване середовище розробки, розроблене для розробників, які працюють із JavaScript. Він включає в себе розумний редактор та інші можливості, які можуть підвищити ефективність під час написання коду. Інтегровані інструменти розробника WebStorm (Рис. 2.1) включають налагодження JavaScript, розширений контроль версій, локальну історію змін, вбудований клієнт HTTP, модульне тестування та багато іншого. Завдяки розумному редактору WebStorm користувачі можуть отримати допомогу в кожному аспекті написання коду. Функція завершення коду надає відповідні ключові слова та символи, коли користувачі вводять і працюють різними мовами. Крім того, використовуючи аналіз якості, редактор може легко виявити помилки та опечатки в коді. Користувачі також можуть отримати доступ до швидкої документації в WebStorm для посилання на відповідну інформацію про символи [9].

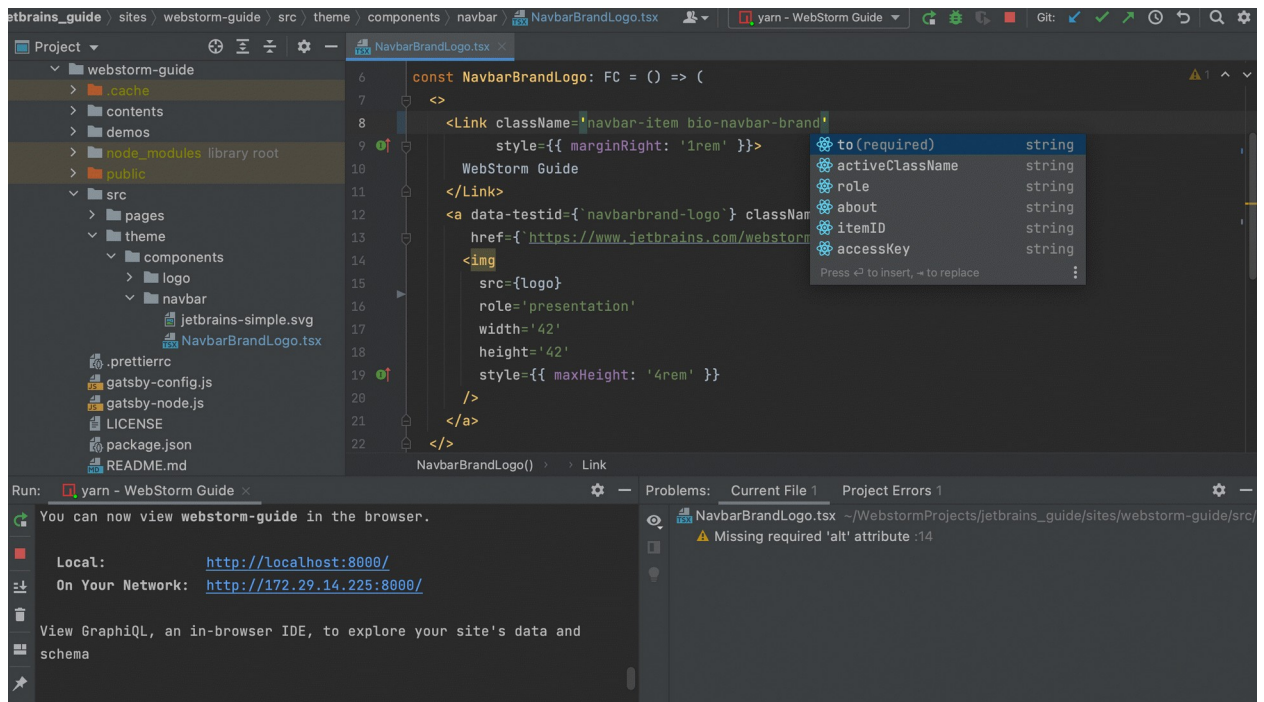


Рис. 2.1 – Інтерфейс WebStorm

Функції WebStorm:

- Інтелектуальна допомога з кодуванням – WebStorm надає розумну допомогу в кодуванні для JavaScript і мов, скомпільованих у JavaScript, Node.js, HTML і CSS.
- Сучасні фреймворки – WebStorm надає допомогу в кодуванні для Angular, React, Vue.js і Meteor.
- Розумний редактор – IDE аналізує ваш проект, щоб забезпечити найкращі результати завершення коду для всіх підтримуваних мов. Сотні вбудованих інспекцій повідомляють про будь – які можливі проблеми прямо під час введення тексту та пропонують варіанти швидкого вирішення.
- Навігація та пошук – WebStorm допомагає ефективніше обійти код та заощадити час під час роботи з великими проектами. Перейдіть до визначення методу, функції чи змінної лише одним клацанням миші або знайдіть варіанти використання.

2.2 Мова програмування JavaScript

JavaScript є простою та легкою для вивчення мовою програмування в порівнянні з іншими мовами, такими як C++, Ruby і Python. Це інтерпретована мова високого рівня, яку можна легко вбудувати в такі мови, як HTML. Його розробили Netscape Communications Corporation, Mozilla Foundation та ECMA International. Брендан Аїх відомий як творець або дизайнер мови програмування JavaScript [11].

JavaScript – це текстова, об'єктно – орієнтована мова програмування, яка використовується, щоб зробити веб – сторінки та програми більш динамічними та інтерактивними для відвідувачів.

```
1 var button = document.getElementById("enter");
2 var input = document.getElementById("userinput");
3 var ul = document.querySelector("ul");
4
5 function inputLength() {
6     return input.value.length;
7 }
8
9 function createListElement() {
10     var li = document.createElement("li");
11     li.appendChild(document.createTextNode(input.value));
12     ul.appendChild(li);
13     input.value = "";
14 }
15
16 function addListAfterClick() {
17     if (inputLength() > 0) {
18         createListElement();
19     }
20 }
21
22 function addListAfterKeypress(event) {
23     if (inputLength() > 0 && event.keyCode === 13) {
24         createListElement();
25     }
26 }
27
28 button.addEventListener("click", addListAfterClick);
29
30 input.addEventListener("keypress", addListAfterKeypress);
```

Рис. 2.2 – Приклад коду JavaScript

Використовуваний як на стороні клієнта, так і на сервері, JavaScript (Рис. 2.2) пропонує розширені можливості, які не можуть дати базові мови, такі як HTML і CSS, наприклад оновлення каналу Twitter, вбудованих відео

YouTube тощо. Іншими словами, веб – сторінки з підтримкою JavaScript є більш динамічними та орієнтованими на користувачів, що може спонукати користувачів повертатися на сайт [12].

Тим не менш, JavaScript призначений для використання з іншими мовами програмування, а не як їх заміна.

Наприклад, веб – сайтам та додаткам все одно потрібно буде використовувати HTML для базової структури, тоді як CSS потрібен для налаштувань форматування та макета, а також керування тим, як відобразатиметься текст, зображення та інший вміст на веб – сторінці. Звідси JavaScript може покращити ці елементи, перетворивши їх із статичних функцій на інтерактивні та захоплюючі.

JavaScript дозволяє веб – сторінкам ставати більш інтерактивними багатьма способами, включаючи відображення анімації, створення функціональних спадних меню, збільшення та зменшення зображень веб – сторінки або навіть зміну кольору кнопок, коли користувач наводить курсор миші на неї [13].

Розробники часто використовують фреймворки JavaScript, такі як Vue, React та Angular, під час створення веб та мобільних додатків, оскільки вони дозволяють використовувати рутинні завдання та функції, спільні для багатьох програм. Ці функції включають кнопку пошуку та параметри вибору категорій.

JavaScript широко використовується в розробці ігор, особливо для нових розробників, які хочуть практикувати та вдосконалювати свої навички.

Деякі розробники також вирішують використовувати Node.js, внутрішню інфраструктуру JavaScript, для створення базових веб – сервісів і розбудови інфраструктури сайту.

Переваги використання JavaScript:

- Інтерактивність: веб – сайти JavaScript є більш інтерактивними та привабливими для користувачів, оскільки мова створює багатші та привабливіші інтерфейси.

- Швидкість: веб – сайти завантажуватимуться та працюватимуть з більш високою швидкістю, оскільки JavaScript можна запускати негайно у браузері, що зменшує попит на сервері.
- Простота: це просто реалізувати, та часто для використання потрібно лише базовий рівень знань.
- Інтеграція: JavaScript добре працює з більшістю інших мов програмування, що дає змогу інтегруватися в безліч різних мов програм.
- Популярність: він неймовірно популярний – більшість веб – сайтів у тій чи іншій формі використовують JavaScript, незалежно від природи веб – сайту.

У використанні JavaScript не так багато недоліків. Основна проблема полягає в тому, що існує ймовірність порушень безпеки на стороні клієнта в коді JavaScript, оскільки він доступний для перегляду користувачем. Незважаючи на це, JavaScript залишається найпопулярнішою мовою програмування на даний момент, та враховуючи, що це рідна мова програмування для веб – браузерів, новачкам – програмістам варто вивчати основи та використовувати їх у проектах.

2.3 Мова програмування TypeScript

Історично склалося так, що JavaScript був основною мовою для написання сценаріїв веб – сторінок та інтернет – додатків, а також зручною можливістю використовувати його як у фронтальній, так і серверній частині таких фреймворків, як Node.js та Deno. Це здорово і таке інше, але звичайна веб – сторінка перетворилася на велику та складну систему, для якої JavaScript просто не призначений [15].

TypeScript – це мова програмування, вперше розроблена компанією Microsoft у 2012 році. Її головне завдання – підвищити продуктивність розробки складних додатків. TypeScript (Рис. 2.4) був розроблений Андерсом

Хейлсбергом у Microsoft. Обмеження JavaScript для розробки великомасштабних додатків Microsoft та серед їх зовнішніх клієнтів призвели до створення TypeScript. Через складність роботи зі складним кодом JavaScript виникла потреба в спеціальних інструментах для полегшення розробки компонентів JavaScript.

Це мова з відкритим вихідним кодом, розроблена як над набір JavaScript. TypeScript компілюється у JavaScript. Простими словами це означає, що будь – який код, дійсний у JavaScript, також є цінним для TypeScript [17].

Статична типізація допомагає вирішити масу проблем при навігації за складними, великомасштабними системами, написаними на JavaScript. Це також спрощує рефакторинг коду, не ламаючи систему повністю і не натикаючись на приховані помилки, такі як горезвісне «undefined is not a function». Будь – який гідний програміст стикався з цим скільки завгодно разів.

Фактично 15% всіх помилок JavaScript можна виявити за допомогою TypeScript. Динамічна типізація може призвести до помилок, які можуть призвести до повної зупинки коду на стороні сервера, але з TypeScript письмовий аспект дає йому явну перевагу при масштабуванні до великих кодових баз комерційних та бізнес – додатків.

Однак мова вводить додаткові функції, такі як введення тексту або об'єктно – орієнтоване програмування. Щоб скористатися цими функціями, бібліотека не потрібна. Замість цього просто скористайтеся інструментом компіляції TypeScript, щоб транспілювати його (компілювати вихідний код однієї мови на іншу) у JavaScript. Таким чином, виконаний код буде JavaScript еквівалентом скомпільованого коду TypeScript [16].

```

type filterFn = (item: string) => bool;

// The higher-order-function takes an array and a function as arguments
function filterItems(arr: string[], fn: filterFn): string[] {
  const newArray: string[] = [];
  arr.forEach(item => {
    if(fn(item)){
      newArray.push(item);
    }
  });
  return newArray;
}

function checkNameLength(name: string) {
  return name.length >= 10;
}

const doctorList = ["DoctorOne", "DoctorTwo", "DoctorThree", "DoctorFour"];

// We are passing the array and a function as arguments to filterItems method.
const output = filterItems(doctorList, checkNameLength);

console.log(output); // ["DoctorThree", "DoctorFour"]

```

Рис. 2.5 – Приклад коду мови програмування TypeScript

Дизайн мови (Рис. 2.5) це те, що робить TypeScript відносно унікальним серед аналогів. Це чиста, елегантна мова програмування, яка, незважаючи на те, що вона вимагає деякого часу для вивчення, є фантастичною для роботи – порівнянню з такими мовами програмування, як Go, Kotlin та Rust.

З точки зору розробки програмного забезпечення, TypeScript пропонує багато переваг перед JavaScript:

- Необов’язкова статична типізація. JavaScript – це мова з динамічною типізацією, що означає, що типи перевіряються, а помилки типів даних виявляються лише під час виконання. Це може бути дуже небезпечним та може призвести до помилок під час виробництва. TypeScript вводить необов’язкову строгу статичну типізацію: після оголошення змінна не змінює свого типу та може набувати лише певних значень.
- Читання. Завдяки додаванню строгих типів та інших елементів, які роблять код виразнішим, ви можете побачити задум розробників, які спочатку написали код. Це особливо важливо для розподілених команд, які працюють над одним проектом. Код, який сам за себе

говорить, може компенсувати відсутність прямого спілкування між членами команди.

- Підтримка IDE. Інформація про типи робить редактори та інтегровані середовища розробки набагато кориснішими. Вони можуть пропонувати такі функції, як навігація за кодом та автозаповнення, надаючи точні пропозиції.
- Сила об'єктної орієнтації. TypeScript підтримує концепції об'єктно – орієнтованого програмування, такі як класи, інтерфейси, успадкування тощо.
- Велика підтримка. TypeScript має відкритий вихідний код та підтримується Google та Microsoft – це 2 з 4 найбільших титанів технологічної індустрії. З такою величезною підтримкою, а також постійним розвитком немає причини не переключитися на мову програмування, яка стане найпопулярнішою у недалекому майбутньому.

2.4 Порівняння JavaScript та TypeScript

Таблиця 2.1

Параметр	JavaScript	TypeScript
Що таке	Потужна система типів, включаючи дженерики та функції JS	Легка, інтерпретована, об'єктно-орієнтована мова з першокласними функціями
Прив'язка даних	TypeScript використовує такі поняття, як типи та інтерфейси, для опису використовуваних даних.	JavaScript така концепція недоступна.
Екосистема	Екосистема досить потужна та інтуїтивно зрозуміла. Таким чином, він дозволяє статично типізувати різні типи ідіоматичних функцій JavaScript, таких як типи	JavaScript пропонує можливість досліджувати та створювати код без етапу складання.

	об'єднання, перетин, розмічене об'єднання.	
NPM-пакет	З Typescript багато пакетів прт поставляються або зі статичними визначеннями типів, або із зовнішніми, які легко встановити.	JavaScript пропонує можливість досліджувати та створювати код без етапу складання.
Крива навчання	Тверда крива навчання. Потрібне попереднє знання сценаріїв.	Гнучка і проста в освоєнні мова сценаріїв.
Прототипування	Typescript має функцію прототипування.	JavaScript не має цієї функції.
Спільнота	Typescript не має великої спільноти розробників.	JavaScript має величезну спільноту розробників
Збірник	Код TypeScript має бути скомпільований	Немає необхідності компілювати JavaScript.
Анотація	Щоб отримати максимальну віддачу від функцій TypeScript, розробники повинні постійно анотувати код.	Анотації не потрібні для JavaScript.

2.5 Бібліотека React

React – це декларативна, ефективна та гнучка бібліотека JavaScript для зовнішнього інтерфейсу. Вона використовується для розробки інтерфейсу користувача спеціально для одно сторінкових сайтів та додатків [19].

React – це бібліотека представлень на основі компонентів (Рис. 2.6) з відкритим вихідним кодом, яка відповідає за розробку рівня представлення веб – додатків та мобільних програм. За допомогою React ви можете розробляти веб – додаток, власний мобільний додаток для Android та IOS.

Розробники у Facebook почали стикатися з проблемами супроводу коду ще у 2011 році. У міру того, як розширювалися можливості програм Facebook Ad, потреба компанії в більшій кількості членів команди зростала. Звичайно, у компанії почалися проблеми з обробкою зростаючого числа функцій додатків та членів команди [19].

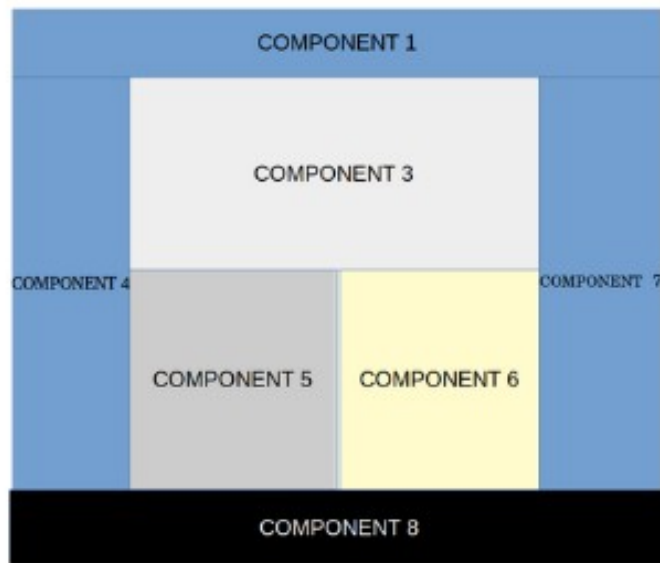


Рис. 2.6 – Приклад макету додатку на основі компонентів

Незабаром стала очевидною потреба в оновленій структурі для забезпечення більш плавної роботи. Ціль полягала в тому, щоб створити структуру, яка дозволила б розробникам розбивати складні компоненти і повторно використовувати коди для прискорення процесу завершення коду.

Розробник Facebook Джордан Уоке успішно створив прототип, який набагато ефективніше керував усім процесом, та саме так з'явився React.js. Хоча спочатку фреймворк був створений тільки для iOS, з ним швидко послідувала підтримка Android, перш ніж він став загальнодоступним у 2015 році.

Розробка React (Рис. 2.7) використовує функції JavaScript для багатьох своїх шаблонів, таких як JSX, що спрощує написання та додавання HTML у React. Основна мета React – звести до мінімуму виникнення помилок у процесі розробки інтерфейсу користувача.

Хоча розробникам React не обов'язково використовувати JSX для React, він працює як корисний наочний посібник для розробників, що дозволяє покращити роботу з інтерфейсом користувача всередині коду

JavaScript. Це також допомагає React більш ефективно ідентифікувати попереджувальні повідомлення та помилки.

```
Navbar.js ×
src > screens > App > components > Navbar.js > Navbar
1 import React from 'react'; 8.4K (gzipped: 3.4K)
2 import PropTypes from 'prop-types'; 1.7K (gzipped: 869)
3 import { Link } from 'react-router-dom'; 26.5K (gzipped: 8.9K)
4
5 function Navbar({ navItems }) {
6   ...return <ul className="App-nav-list">
7     .....{
8     .....  navItems.map(navItem =>
9     .....    <li className="App-nav-item">
10    .....    <Link to={navItem.route}>{navItem.name}</Link>
11    .....    </li>
12    .....  )
13    ..... }
14  </ul>
15 }
16
17 Navbar.propTypes = {
18   ... navItems: PropTypes.arrayOf(PropTypes.string)
19 };
20
21 export default Navbar;
```

Рис. 2.7 – Приклад коду React

2.5.1 JSX

У React ви ніколи не торкаєтесь власне HTML. Ви працюєте з розширенням синтаксису, яке справді є однією з найчудовіших частин екосистеми React: JSX. JSX настільки схожий на HTML, що ви можете думати про нього як про JavaScript із присмаком HTML. Що добре в JSX (Рис. 2.8), то це те, що якщо ви знаєте HTML і CSS, ви інтуїтивно розумієте, як працювати з JSX [20].

JSX – це абстракція поверх API `React.createElement()`. Одна з причин, чому це життєво важливо для бібліотеки – та чому команда React вирішила використовувати її в першу чергу – полягає в тому, що API буде занадто громіздким для використання з точки зору масштабування. Потенційно можна було б використовувати `React.createElement()` для створення всієї програми, однак це було б не більш ефективно, ніж використання власне

HTML. Спочатку може здатися, що ми зробили крок назад, додавши нашу розмітку в логіку нашого шаблону, проте кілька швидких хвилин з JSX, та ви зачепите за стиль.

```
render() {
  const { name } = this.state;
  return (
    <div className="App">
      <header className="App-header">
        <img src={logo} className="App-logo" alt="logo" />
        <h1 className="App-title">Welcome to React</h1>
      </header>
      <input type="text" onChange={this.handleChange} />
      <br />
      <b>{name}</b>

      <Home />
    </div>
  );
}
```

Рис. 2.8 – Приклад коду JSX

2.6 Мова розмітки HTML

HTML – це мова гіпертекстової розмітки. Гіпертекст – це, по суті, текст усередині тексту, а мова розмітки – це мова, зрозуміла комп’ютерам, призначена для опису веб – сторінок та підвищення інтерактивності використаного вами тексту [22].

HTML був винайдений Тімом Бернерсом – Лі, Робертом Кайо та іншими ще в 1989 як мову публікації в інтернеті, та кажучи дуже просто, це перший будівельний блок при створенні веб – сторінки. Коли ви вперше зіткнетесь зі світом HTML, ви ймовірно зіткнетесь з цілим набором термінів та фраз, які також дещо збивають з пантелику. Три основні – це елементи, теги та атрибути.

HTML підтримується кожним браузером та встановлений практично на кожній існуючій веб – сторінці. Вам не потрібні ліцензії, вам не потрібно платити. Якщо ми можемо порівняти веб – сторінку з людським тілом, то HTML це кістки тіла [22].

```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD
  XHTML 1.0 Transitional//EN"
2 "http://www.w3.org/TR/xhtml1/DTD/
  xhtml1-transitional.dtd">
3
4 <html xmlns="http://www.w3.org/1999/
  xhtml">
5   <head>
6     <meta http-equiv="Content-
  Type" content=
7       "text/html; charset=us-
  ascii" />
8     <script type="text/
  javascript">
9       function reDo() {top.
  location.reload();}
10      if (navigator.appName ==
  'Netscape') {top.onresize = reDo;}
11      dom=document.
  getElementById;
12    </script>
13  </head>
14  <body>
15  </body>
16 </html>

```

Рис. 2.9 – Приклад коду HTML 5

HTML 5 (Рис. 2.9) – це остання «версія» HTML, та знання в цій галузі – тенденція, що росте на ринку праці. За великим рахунком, він має ту ж функціональність, що і стандартний HTML, але набагато динамічніший та використовує набагато менше коду для створення чогось фантастичного.

По суті, HTML використовується для створення основного вмісту веб – сторінки, надання їй структури. Ви починаєте з написання слів, потім використовуєте теги або елементи до цих слів. Потім веб – браузер читає це та може зрозуміти заголовок сторінки, будь – які абзаци, а також, де сторінка починається та закінчується, таким чином заповнюючи вашу веб – сторінку контентом.

2.7 Мова стилів CSS

CSS – це аббревіатура від Cascade Styling Sheets. Коротше кажучи, це мова стилів аркуша, тип мови, яку ви можете використовувати для опису

представлення мови розмітки – в даному випадку для опису рухів HTML. Він ефективно визначає, як будівельні блоки, закладені в HTML, оформляються та надаються користувачеві [23].

```
34  /* A reference to a type */
35  span.ts span.type-ref {
36      color: ■ rgb(175, 0, 219) !important;
37  }
38
39  /* Signature details */
40  div.signature > table {
41      border-collapse: collapse;
42      border: thin ■ darkgray solid;
43      width: 60%;
44  }
```

Рис. 2.10 – Приклад коду CSS

CSS був вперше створений приблизно в 1996 році, щоб зрозуміти HTML та зробити веб – сторінку приголомшливою.

Якщо HTML це кістки тіла, то CSS (Рис. 2.10) це оболонка, яка його покриває. Він використовується для кольору фону, стилю макета, меж, затінювання – всіх основних елементів дизайну, які роблять веб – сторінку гладкою та розумною. CSS дозволяє розрізняти уявлення та зміст, змінюючи дизайн та відображення елементів HTML.

Презентація та простота використання – це пара основних речей, які CSS привніс до веб – дизайну, перевірши те, як контент виглядає на веб – сторінці, і те, що ще доповнює цей контент. Хоча він часто використовується у поєднанні з HTML, насправді він незалежний від нього та може використовуватись з будь – якою мовою розмітки на основі XML.

Багато людей вважають, що HTML та CSS – це те саме, але насправді це зовсім різні мови. Хоча вони часто використовуються разом у кодуванні та веб – дизайні, їх можна використовувати та окремо, та обидва мають безліч різноманітних застосувань.

Погляньмо на деякі ключові відмінності між HTML та CSS:

- HTML визначає зміст та структуру веб – сторінки, тоді як CSS переводить відображення та дизайн HTML – елементів.
- В результаті HTML може включати код CSS у свої файли, але CSS не може містити HTML у своїх аркушах.
- HTML – це один встановлений тип або метод, тоді як, залежно від вимог, CSS може бути або внутрішнім, або зовнішнім.
- HTML не використовується для представлення або візуалізації, як CSS.
- Коли справа доходить до написання мови, HTML має досить ясний та простий для розуміння синтаксис, у той час як CSS може бути трохи складнішим та заплутаним.
- HTML використовує «теги» по обидва боки від вмісту, тоді як CSS використовує «селектори», за якими слідує блок оголошень.
- Код що використовується HTML, набагато більше, ніж код, що використовується CSS, що дає CSS менше часу завантаження веб – сторінки.

2.8 Система управління базами даних MySQL

MySQL – це система управління реляційними базами даних із відкритим вихідним кодом. Як і в інших реляційних базах даних MySQL зберігає дані в таблицях, що складаються з рядків та стовпців. Користувачі можуть визначати, маніпулювати, контролювати та запитувати дані за допомогою мови структурованих запитів, біль відомої як SQL. Ім'я MySQL є комбінацією «Му», імені дочки творця MySQL Майкла Віденіуса, та «SQL» [24].

Гнучка та потужна програма MySQL (Рис. 2.11) є найпопулярнішою системою баз даних з відкритим вихідним кодом у світі. Будучи частиною широко використовується стека технологій LAMP (який складається з операційної системи на базі Linux, веб-сервера Apache, бази даних MySQL і

PHP для обробки), він використовується для зберігання та вилучення даних у найрізноманітніших популярних додатках, веб – сайтах та сервісах.

Спочатку MySQL була запущена ще в 1995 році. З того часу вона зазнала кількох змін у власності / управлінні, перш ніж у 2010 році опинилася у корпорації Oracle.

MySQL вибирає підхід, який називається реляційною базою даних. У реляційній базі даних ваші дані розбиваються на кілька окремих областей зберігання, які називаються таблицями, а не об'єднуються в одну велику одиницю зберігання [25].

В даний час ми використовуємо системи управління реляційними базами даних для зберігання та управління величезним обсягом даних. Це називається реляційною базою даних, тому що всі дані зберігаються в різних таблицях, а відносини встановлюються за допомогою первинних ключів або інших ключів, відомих як зовнішні ключі.

Система управління реляційною базою даних – це програмне забезпечення яке:

- Дозволяє реалізувати базу даних із таблицями, стовпцями та індексами.
- Гарантує цілісність посилань між рядками різних таблиць.
- Оновлює індекси автоматично.
- Інтерпретує запит SQL та поєднує інформацію з різних таблиць.

2.8.1 phpMyAdmin

PhpMyAdmin – це веб – програма, яка дозволяє вам керувати базами даних MySQL за допомогою веб – браузера. PhpMyAdmin представлений 9 вересня 1998 року та написаний на PHP. Він пропонує простий у використанні інтерфейс, який дозволяє запускати команди MySQL та операції з баз даних.

Ви також можете використовувати його для перегляду та редагування таблиць, рядків та полів бази даних. Він також дозволяє імпортувати, експортувати або видаляти всі дані у базі даних.

Це найпопулярніша програма для управління базою даних MySQL. PhpMyAdmin також підтримує широкий спектр операцій, таких як управління базами даних, відносинами, таблицями, стовпцями, індексами, дозволами, користувачами тощо в MySQL та MariaDB. Ці операції можна використовувати через інтерфейс (Рис. 2.12) користувача, при цьому у нас все ще є можливість виконати будь – який оператор SQL.

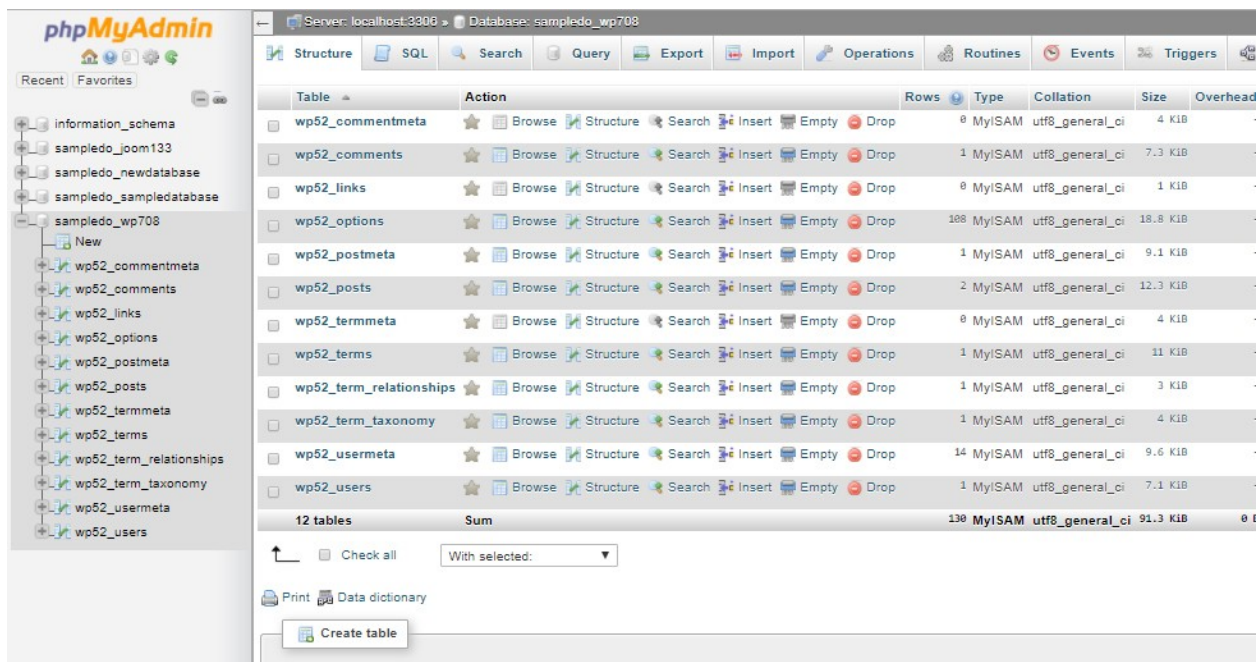


Рис. 2.12 – Інтерфейс PhpMyAdmin

PhpMyAdmin перекладено 72 мовами, а також підтримує мови RTL та LTR, так що широке коло людей може легко використовувати це програмне забезпечення. Ми можемо запускати запити MySQL, відновлювати, оптимізувати, перевіряти таблиці та виконувати інші команди управління базою даних. PhpMyAdmin також можна використовувати для виконання завдань, таких як створення бази даних, виконання запитів.

2.9 API

API означає інтерфейс прикладного програмування, який є набором визначень і протоколів для створення та інтеграції прикладного програмного забезпечення.

API дозволяють вашому продукту або послуги взаємодіяти з іншими продуктами та послугами (Рис. 2.13), не знаючи, як вони реалізовані. Це може спростити розробку додатків, заощадивши час та гроші. Коли ви розробляєте нові інструменти та продукти або керуєте наявними, API дають вам:

- гнучкість
- спростити дизайн
- адміністрування
- використання
- надати можливості для інновацій.

Оскільки API інтерфейси спрощують інтеграцію нових компонентів додатків у існуючу архітектуру, вони допомагають бізнесу та IT-командам співпрацювати. Потреби бізнесу часто швидко змінюються у відповідь на цифрові ринки, що постійно змінюються, де нові конкуренти можуть змінити всю галузь за допомогою нового додатка. Щоб залишатися конкурентоспроможними, важливо підтримувати швидку розробку та розгортання інноваційних послуг. Розробка хмарних програм — це очевидний спосіб підвищити швидкість розробки, заснований на підключенні архітектури програм мікросервісів через API.

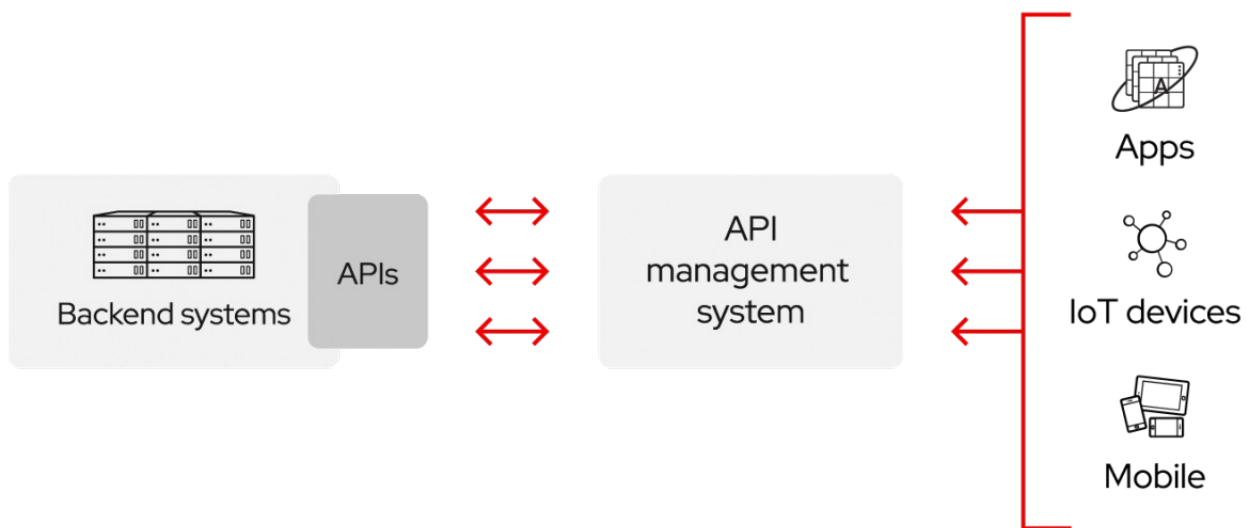


Рис. 2.13 – Схема роботи API

API – це спрощений спосіб підключення вашої власної інфраструктури за допомогою розробки хмарних програм, але вони також дозволяють вам ділитися своїми даними з клієнтами та іншими зовнішніми користувачами. Загальнодоступні API – інтерфейси є унікальною цінністю для бізнесу, оскільки вони можуть спростити та розширити способи зв'язку з вашими партнерами, а також потенційно монетизувати ваші дані (популярним прикладом є API – інтерфейс Google Maps).

Кожен API містить і реалізується викликами функцій – операторами мови, які вимагають програмне забезпечення для виконання певних дій та послуг. Виклики функцій – це фрази, що складаються з дієслів та іменників, наприклад:

- Почати або завершити сеанс
- Отримайте зручності для одномісного номера
- Відновлення або отримання об'єктів із сервера.

Коротше кажучи, інтерфейси API дозволяють відкрити доступ до ваших ресурсів, зберігаючи при цьому безпеку і контроль. Як ви відкриваєте доступ і кому вирішувати лише вам. Безпека API — це правильне керування API, зокрема використання шлюзу API. Підключення до API та створення програм, що використовують дані або функціональні можливості, що

надаються API, можна здійснити за допомогою розподіленої інтеграційної платформи, яка з'єднує всі, включаючи застарілі системи та Інтернет речей (IoT).

В даному розділі були розглянуті інструменти за допомогою яких розробник може розробляти свій власний веб – сайт. Це середовище розробки, та мова програмування. Також були розглянуті ресурси які надають розробнику дані та інструмент за допомогою якого він робить веб – сайт.

РОЗДІЛ 3

РОЗРОБКА ІНТЕРНЕТ – МАГАЗИНУ КОМПЛЕКТУЮЧИХ ДЛЯ ПЕРСОНАЛЬНИХ КОМП'ЮТЕРІВ

3.1 Розробка інтернет – магазину комплектуючих

В даному розділі можна детально ознайомитись зі структурою інтернет магазину для персональних комп'ютерів, розглянути та розібрати код та зрозуміти як працює цей інтернет-магазин.

3.2 Опис інтернет – магазину

3.2.1 Головний екран

На цьому екрані розміщено навігаційне меню інтернет – магазину (Рис. 3.1) з такими категоріями:

- Home;

- Products;
- On sale;
- Contact us.
- Корзина



Рис. 3.1 – Навігаційне меню інтернет – магазину

На Рис. 3.2, є код який виводить це навігаційне меню на екран.

```
import React from "react";
import { Nav, NavLink, Bars, NavMenu, NavBtn, NavBtnLink } from "./NavbarElements";

const Navbar = () => {
  return (
    <>
      <Nav>
        <NavLink to="/">
          <img src={require("../images/game.png")} alt="Fizel games" style={{height:60}} /> <span style={{color:"white", padding:2}} />
        </NavLink>
        <Bars />
        <NavMenu>
          <NavLink to="/home">
            <span style={{fontSize:22, fontFamily:"montserrat,sans-serif"}}> Home </span>
          </NavLink>
          <NavLink to="/products">
            <span style={{fontSize:22, fontFamily:"montserrat,sans-serif"}}> Products </span>
          </NavLink>
          <NavLink to="/sale">
            <span style={{fontSize:22, fontFamily:"montserrat,sans-serif"}}> On sale </span>
          </NavLink>
          <NavLink to="/contact">
            <span style={{fontSize:22, fontFamily:"montserrat,sans-serif"}}> Contact us </span>
          </NavLink>
        </NavMenu>
        <NavBtn>
          <NavBtnLink to="/cart">  </NavBtnLink>
        </NavBtn>
      </Nav>
    </>
  )
}

export default Navbar
```

Рис. 3.2 – Код навігаційної панелі

Також цей сайт виконано як (SPA), тобто генерується один HTML документ, а потім деякі його компоненти просто змінюються.

```
return (  
  // <div className="App">  
    <Router>  
      <Navbar/>  
      <Switch>  
        <Route path="/" exact component={Home}/>  
        <Route path="/home" component={Home}/>  
        <Route path="/contact" component={Contact}/>  
        <Route path="/cart" component={Cart}/>  
        <Route path="/products" component={Products}/>  
      </Switch>  
    </Router>  
  );
```

Рис. 3.3 – Код SPA сайту

Далі йде пункт категорій (Рис. 3.4), тут можна обрати потрібний продукт. На рисунку показано такі категорії:

- Gamepads;
- Keyboards;
- Consoles.

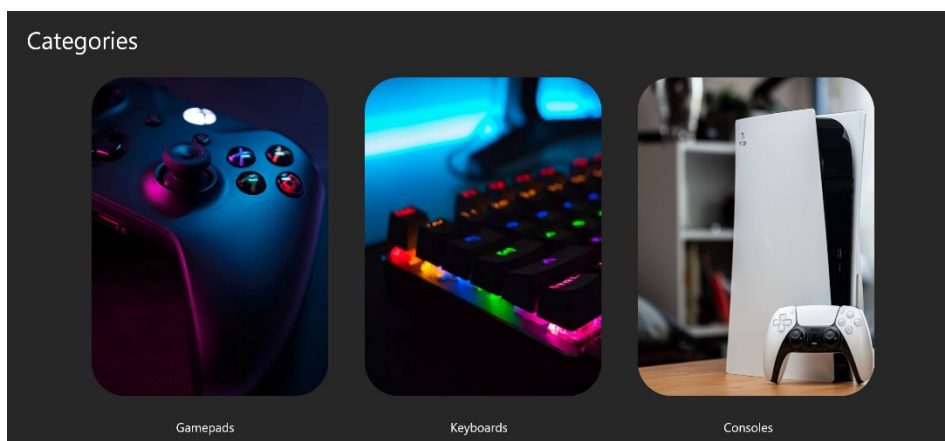


Рис. 3.4 – Пункт «Категорії»

Далі рекламний банер. За ним пункт із найкращими пропозиціями (Рис. 3.5). Тут виводяться товари, які найчастіше купують.

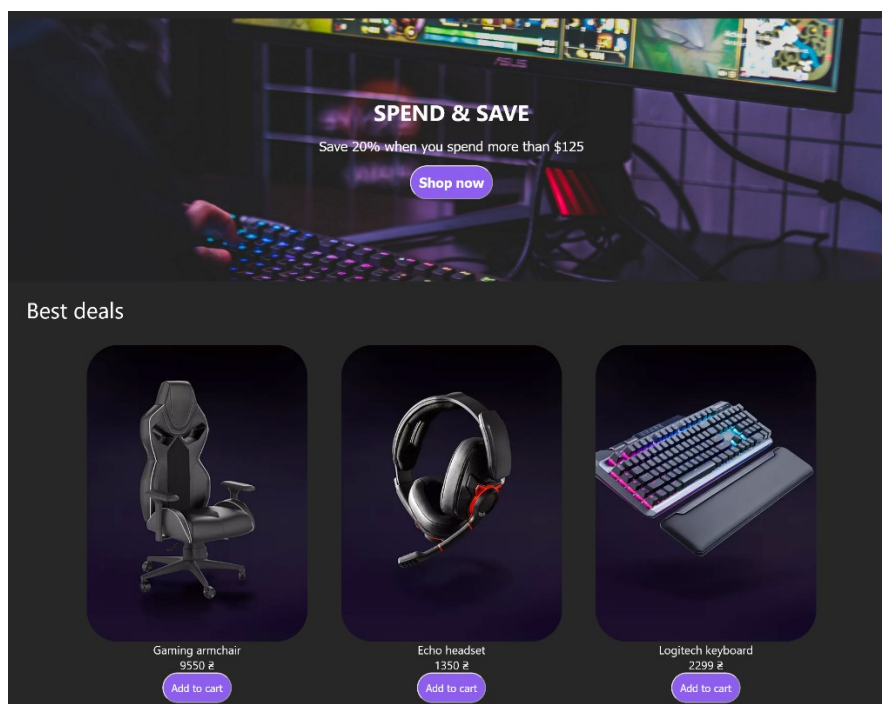


Рис. 3.5 – Рекламний банер та найкращі товари

Та в самому низу йде графа, де можна підписатись на повідомлення від інтернет магазину.

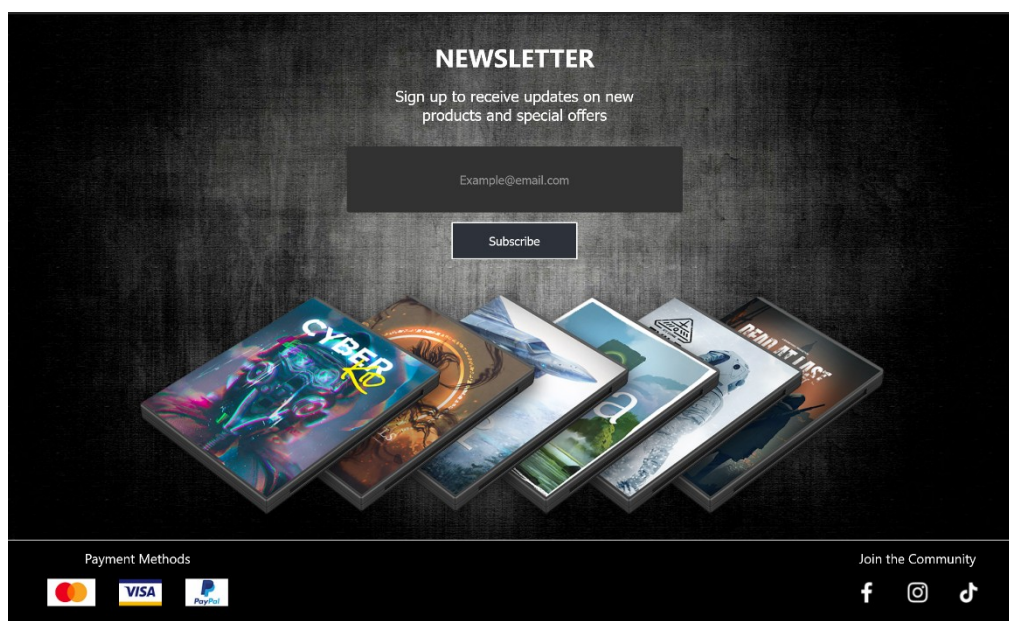



Рис. 3.6 – Графа підписки на повідомлення

3.2.2 Екран товарів

Щоб потрапити на екран товарів, потрібно натиснути кнопку "Products". Після цього сайт перекидає на цей екран. На цьому екрані представлено всю продукцію сайту. Зліва є невеликий фільтр, в якому можна відсортувати потрібний за характеристиками товар.


Після відкриття цього екрана користувачу показується певна кількість товару. Щоб інтернет-магазин показав ще товару, потрібно натиснути кнопку "Load more", тоді сайт провантажить ще певну кількість товару.

FIZEL GAMES Home Products On sale Contact us 

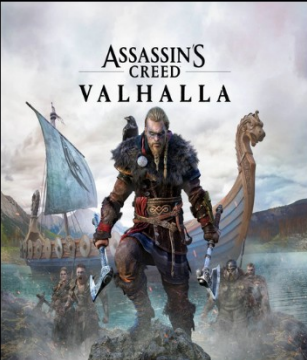
SHOP ALL

Filter


- Games
- Peripherals
- Consoles
- Chairs
- Gamepads




Watch Dogs 2
1200 ₴




Assassin's Creed Valhalla
1200 ₴




Deep Rock Galactic
800 ₴




Grand Theft Auto 5
700 ₴




Dishonored 2
200 ₴




Deep Rock Galactic
800 ₴



Rainbow Six Siege
599 ₴




Wolfenstein: New Order
699 ₴



Satisfactory
499 ₴

[Load more](#)

Payment Methods



Join the Community

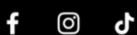


Рис. 3.7 – Экран товаров

3.2.3 Екран кошику

Після вибору товару потрібно натиснути на значок "Кошки" і товар потрапить на екран кошика. На цьому екрані показується весь товар, який вибрав покупець та хоче купити.

Ліворуч показується обраний товар:

- Фото;
- Назва;
- Вартість.

Справа сума всіх товарів та вартість доставки даного товару. Потім йде підсумкова вартість товару та кнопка яка підтверджує покупку.

Також є можливість видалити не потрібний товар. Для цього треба натиснути на хрестик.

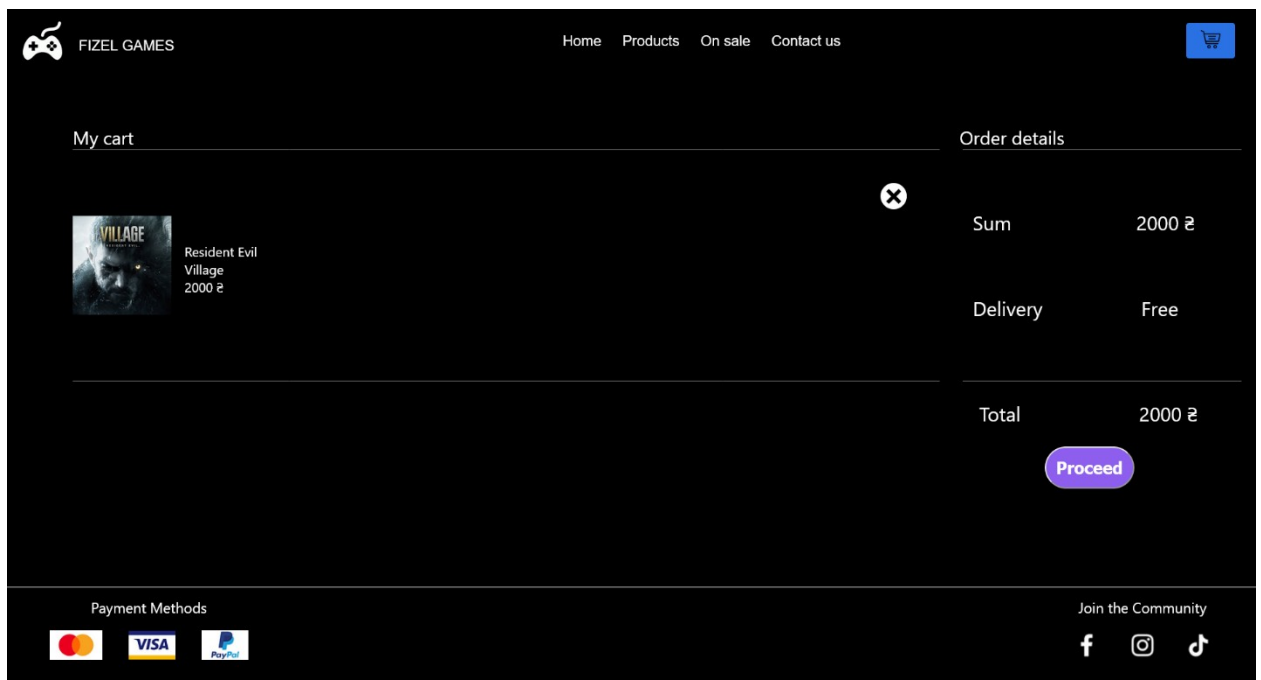


Рис. 3.8 – Кошик інтернет – магазину

3.2.4 Екран контактів

На даному екрані відображається контактна інформація інтернет-магазину:

- Електронна пошта;
- Номер телефону;
- Адреса;
- Карта, яка показує де знаходиться головний офіс.

Справа є форма куди можна написати повідомлення з питання, що цікавить.

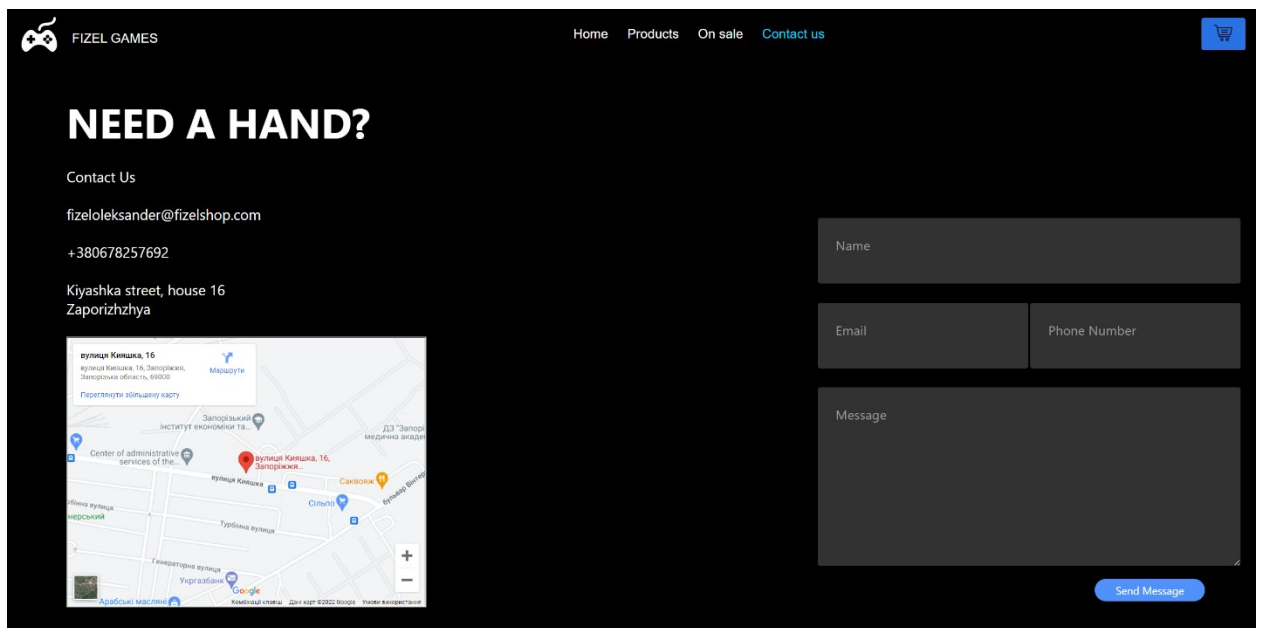


Рис. 3.9 – Екран контактів інтернет – магазину

ВИСНОВКИ

Під час написання дипломної роботи, був розроблений та протестований інтернет – магазин комплектуючих для персональних комп'ютерів. Під час розробки, були виконані такі завдання:

- Знайшов матеріал для розробки інтернет – магазину.
- Знайшов та ознайомився з відповідними інструментами розробки.
- Розробив план розробки інтернет – магазину.
- Розробив інтернет – магазину комплектуючих для персональних комп'ютерів.
- Протестував інтернет – магазин на помилки та працездатність.
- Написав кваліфікаційну роботу та зробив висновки.

Даний інтернет-магазин має простий і зрозумілий інтерфейс. Завдяки фільтру, користувач може знайти потрібний товар і купити його через кошик.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Артур Мазітов. Створення та просування інтернет-магазину. Як відкрити та заробити. 2020. 250 с.
2. Сэмюэл Грингард. Інтернет речей. Майбутнє вже тут. 2017. 188 с.
3. Що таке електронна комерція [Електронний ресурс] – Режим доступу: [www. URL: https://www.oracle.com/cis/cx/ecommerce/what-is-ecommerce/](https://www.oracle.com/cis/cx/ecommerce/what-is-ecommerce/)
4. Все про комерцію в інтернеті [Електронний ресурс] – Режим доступу: [www. URL: https://wezom.com.ua/blog/elektronnaya-kommertsiya](https://wezom.com.ua/blog/elektronnaya-kommertsiya)
5. Електронна комерція в 2022 році. [Електронний ресурс] – Режим доступу: [www. URL: https://ecommerce-platforms.com/ru/glossary/ecommerce](https://ecommerce-platforms.com/ru/glossary/ecommerce)
6. Бред Стоун. The Everything Store: Джефф Безос та ера Amazon. 2013. 916 с.
7. Колін Брайар, Білл Карр. Працюючи навпаки. Інсайти та секрети від топ-менеджерів Amazon. 2022. 416 с.
8. WebStorm. [Електронний ресурс] – Режим доступу: [www. URL: https://www.jetbrains.com/ru-ru/webstorm/](https://www.jetbrains.com/ru-ru/webstorm/)
9. Опис WebStorm. [Електронний ресурс] – Режим доступу: [www. URL: https://itpro.ua/product/jetbrains-webstorm/?tab=description](https://itpro.ua/product/jetbrains-webstorm/?tab=description)
10. Що таке API. [Електронний ресурс] – Режим доступу: [www. URL: https://techrocks.ru/2018/05/29/api-simple-explanation/](https://techrocks.ru/2018/05/29/api-simple-explanation/)
11. Посібник по JavaScript [Електронний ресурс] – Режим доступу: [www. URL: https://uk.javascript.info/](https://uk.javascript.info/)
12. Документація по JavaScript [Електронний ресурс] – Режим доступу: [www. URL: https://developer.mozilla.org/ru/docs/Web/JavaScript](https://developer.mozilla.org/ru/docs/Web/JavaScript)
13. Чиннатхамбі До. JavaScript з нуля. 2021. 400 с.
14. Дронов В.А. JavaScript. 20 уроків для початківців. 2021. 352 с.

- 15.Документація по TypeScript [Електронний ресурс] – Режим доступу: [www. URL: https://www.typescriptlang.org/docs/](https://www.typescriptlang.org/docs/)
- 16.Ведення в TypeScript [Електронний ресурс] – Режим доступу: [www. URL: https://metanit.com/web/typescript/1.1.php](https://metanit.com/web/typescript/1.1.php)
- 17.Гайди по TypeScript [Електронний ресурс] – Режим доступу: [www. URL: https://devblogs.microsoft.com/typescript/](https://devblogs.microsoft.com/typescript/)
- 18.Початок роботи з React [Електронний ресурс] – Режим доступу: [www. URL: https://developer.mozilla.org/ru/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/React_getting_started](https://developer.mozilla.org/ru/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/React_getting_started)
- 19.Документація React [Електронний ресурс] – Режим доступу: [www. URL: https://ru.react.js.org/docs/getting-started.html](https://ru.react.js.org/docs/getting-started.html)
- 20.Документація по JSX [Електронний ресурс] – Режим доступу: [www. URL: https://ru.reactjs.org/docs/introducing-jsx.html](https://ru.reactjs.org/docs/introducing-jsx.html)
- 21.Робсон Елізабет та Фрімен Ерік. Вивчаємо HTML, XHTML та CSS 2-ге вид. 2013. 720 с.
- 22.Джон Дакетт. HTML та CSS. Розробка та дизайн веб-сайтів. 2022. 478 с.
- 23.Основи CSS. [Електронний ресурс] – Режим доступу: [www. URL: https://developer.mozilla.org/ru/docs/Learn/Getting_started_with_the_web/CSS_basics](https://developer.mozilla.org/ru/docs/Learn/Getting_started_with_the_web/CSS_basics)
- 24.Зайцев Петро, Ткаченко Вадим, та Шварц Берон. MySQL по максимуму. 3-тє видання. 2018. 864 с.
- 25.Що таке MySQL. [Електронний ресурс] – Режим доступу: [www. URL: https://freehost.com.ua/faq/wiki/что-такое-mysql/](https://freehost.com.ua/faq/wiki/что-такое-mysql/)

