

**ДЕРЖАВНИЙ ТОРГОВЕЛЬНО – ЕКОНОМІЧНИЙ
УНІВЕРСИТЕТ**

Кафедра комп'ютерних наук та інформаційних систем

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

**«Розробка дизайну та програмна реалізація сайту
комп'ютерного клубу»**

Студента 4 курсу, 8 групи
спеціальності
122 «Комп'ютерні науки»

підпис студента

Палієнко Артем
Вікторович

Науковий керівник
кандидат педагогічних наук, доцент

підпис керівника

Дивак Володимир
Валерійович

Гарант освітньої програми
кандидат технічних наук, доцент

підпис керівника

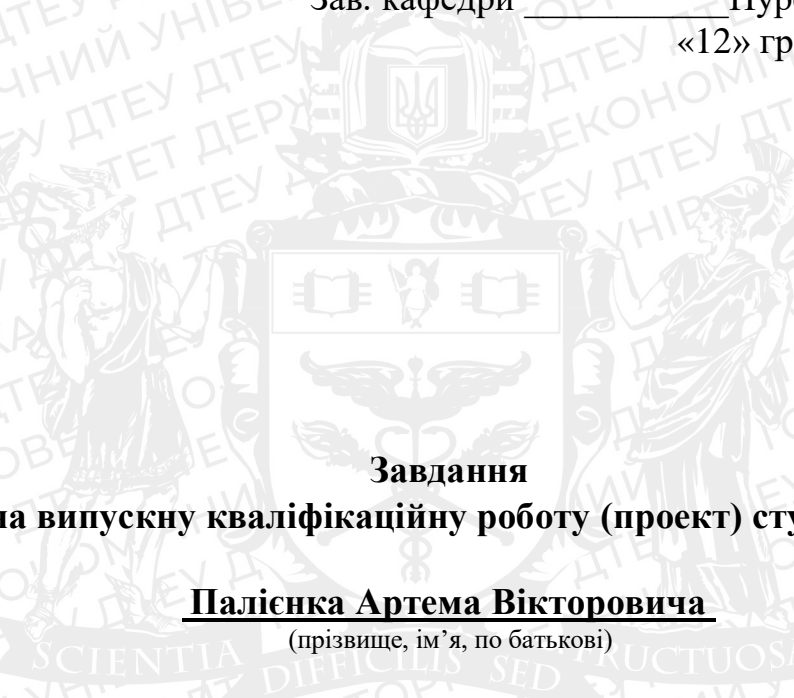
Демідов Павло
Георгійович

Київ 2023

Державний торговельно-економічний університет

Факультет інформаційних технологій
Кафедра комп'ютерних наук та інформаційних систем
Спеціальність 122 «Комп'ютерні науки»

Затверджую
Зав. кафедри _____ Пурський О.І.
«12» грудня 2022р.



Завдання
на випускню кваліфікаційну роботу (проект) студенту

Палієнка Артема Вікторовича

(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної роботи
«Розробка дизайну та програмна реалізація сайту комп'ютерного клубу»
Затверджена наказом ректора від «09» грудня 2022 р. № 3332
2. Строк здачі студентом закінченої роботи 30 травня 2023 року
3. Цільова установка та вихідні дані до роботи
Мета роботи: розробка дизайну та програмна реалізація сайту комп'ютерного клубу»
Об'єкт дослідження: процес розробки дизайну та програмна реалізація сайту комп'ютерного клубу
Предмет дослідження: методи і технологія розробки WEB-сайту комп'ютерного клубу
4. Перелік графічного матеріалу _____

5. Консультанти по роботі із зазначенням розділів, за якими здійснюється консультування:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Дивак В.В	15.12.2022 р.	15.12.2022 р.
2	Дивак В.В	15.12.2022 р.	15.12.2022 р.
3	Дивак В.В	15.12.2022 р.	15.12.2022 р.

6. Зміст випускної кваліфікаційної роботи (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. ТЕОРЕТИЧНІ ПИТАННЯ ФОРМУВАННЯ ДИЗАЙНУ ТА ПРОГРАМНОЇ РЕАЛІЗАЦІЇ САЙТУ КОМП'ЮТЕРНОГО КЛУБУ

1.1. Загальні поняття Web-сайту

1.2. Особливості розробки дизайну та програмної реалізації Web-сайту комп'ютерного клубу

Висновок до 1 розділу

РОЗДІЛ 2. МОДЕЛЬ WEB-САЙТУ КОМП'ЮТЕРНОГО КЛУБУ

2.1. Методи, методології і технології розробки Web-сайту комп'ютерного клубу

2.2. Модель Web-сайту комп'ютерного клубу

Висновок до 2 розділу

РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ WEB-САЙТУ КОМП'ЮТЕРНОГО КЛУБУ

3.1. Програмна реалізація Web-сайту комп'ютерного клубу

3.2. Перевірка Web-сайту комп'ютерного клубу

3.3. Рекомендації з упровадження Web-сайту комп'ютерного клубу

Висновок до 3 розділу

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

7. Календарний план виконання роботи роботи

№ Пор.	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		За планом	фактично
1	2	3	4
1	Вибір теми випускної кваліфікаційної роботи	04.10.2022	04.10.2022
2	Розробка та затвердження завдання на випускну кваліфікаційну роботу	15.12.2022	15.12.2022
3	Вступ	03.02.2023	03.02.2023

4	<i>РОЗДІЛ 1. Теоретичні питання формування дизайну та програмної реалізації сайту комп'ютерного клубу</i>	28.02.2023	28.02.2023
5	<i>РОЗДІЛ 2. Модель Web-сайту комп'ютерного клубу</i>	06.04.2023	06.04.2023
6	<i>РОЗДІЛ 3. Практична реалізація Web-сайту комп'ютерного клубу</i>	12.05.2023	12.05.2023
7	<i>Висновки</i>	15.05.2023	15.05.2023
8	<i>Здача випускної кваліфікаційної роботи на кафедрі науковому керівнику</i>	30.05.2023	30.05.2023
9	<i>Попередній захист випускної кваліфікаційної роботи</i>	31.05.2023- 01.06.2023	31.05.2023- 01.06.2023
11	<i>Виправлення зауважень, зовнішнє рецензування випускної кваліфікаційної роботи</i>	02.06.2023	02.06.2023
12	<i>Представлення готової зшитої випускної кваліфікаційної роботи на кафедрі</i>	05.06.2023	05.06.2023
13	<i>Публічний захист випускної кваліфікаційної роботи</i>	<i>За розкладом роботи ЕК</i>	

8. Дата видачі завдання «15» грудня 2022 р.

Керівник випускної кваліфікаційної роботи

Дивак В.В.

(прізвище, ініціали, підпис)

Гарант освітньої програми

Демідов П.Г.

(прізвище, ініціали, підпис)

Завдання прийняв до виконання студент-дипломник

Палієнко А.В.

(прізвище, ініціали, підпис)

9. Відгук керівника випускної кваліфікаційної роботи (проекту)

Керівник випускної кваліфікаційної роботи (проекту)

30.05.2023 р.

(підпис, дата)

10. Висновок про випускну кваліфікаційну роботу

Випускна кваліфікаційна робота студента

Палієнка А.В.

(прізвище, ініціали)

може бути допущена до захисту в екзаменаційній комісії.

Гарант освітньої програми

Демідов П.Г.

(підпис, прізвище, ініціали)

Завідувач кафедри

Пурський О.І.

(підпис, прізвище, ініціали)

« »

2023 р.

Анотація

У випускній кваліфікаційній роботі було здійснено комплексну розробку Web-сайту комп'ютерного клубу. Було теоретично обґрунтовано вибір теми, її проблематику та сучасний стан досліджень. Розглянуто та теоретично обґрунтовано методи розробки та реалізації сайту комп'ютерного клубу, та запропоновано концепцію розробки Web-сайту комп'ютерного клубу для зручної та кращої роботи з клієнтам та їх інформуванням з приводу послуг комп'ютерного клубу. Досліджено такі питання: аналіз існуючих подібних сайтів, котрі належать іншим клубам, аналіз недоліків сайтів та важливості створення якісного дизайну та зручного інтерфейсу для користувачів, аналіз методів та технологій доступних для розробки та сайту, а також застосування цих технологій для реалізації ідеї створення сайту. У процесі виконання був створений сайт типу «Landing-page», котрий містить усю необхідну інформацію, потрібну користувачеві: перегляд послуг, їх вартість, інформація про клуб, можливість забронювати місце, та інші можливості.

Ключові слова: Web-сайт, HTML, CSS, Java-script, Веб-сторінка, користувач, IDE.

Anotation

In the graduation qualification work, a comprehensive development of a computer club website was carried out. The choice of topic, its issues, and the current state of research were theoretically justified. The methods of development and implementation of the computer club website were considered and theoretically justified, and a concept for the development of a computer club website was proposed for convenient and better work with clients and informing them about the services of the computer club. The following issues were studied: an analysis of existing similar websites belonging to other clubs, an analysis of the shortcomings of websites and the importance of creating a quality design and user-friendly interface for users, an analysis of methods and technologies available for website development, and the application of these technologies to implement the idea of creating a website. In the process of execution, a "Landing-page" type website was created, which contains all the necessary information required by the user: viewing services, their cost, information about the club, the ability to book a place, and other features.

Keywords: Website, HTML, CSS, Java-script, Web page, User, IDE.

Розшифрування абревіатур, використаних у даній роботі:

HTML – Hyper Text Markup Language

CSS – Cascade Style Sheet

JS – JavaScript

DevOps – Development and Operations

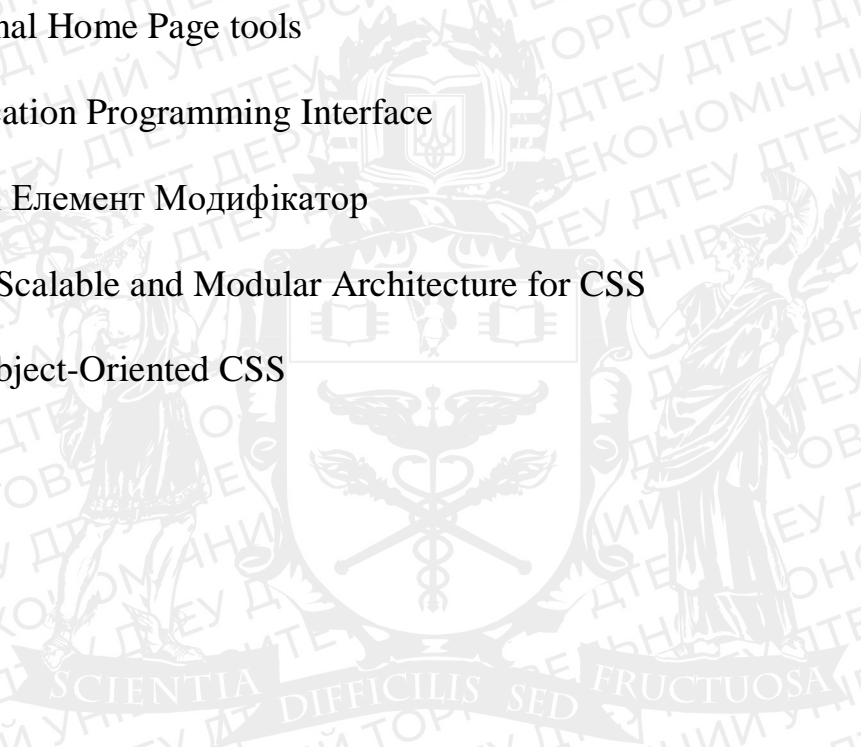
PHP - Personal Home Page tools

API - Application Programming Interface

BEM – Блок Елемент Модифікатор

SMACSS – Scalable and Modular Architecture for CSS

OOCSS – Object-Oriented CSS



ЗМІСТ

ВСТУП	10
РОЗДІЛ 1. ТЕОРЕТИЧНІ ПИТАННЯ ФОРМУВАННЯ ДИЗАЙНУ ТА ПРОГРАМНОЇ РЕАЛІЗАЦІЇ САЙТУ КОМП'ЮТЕРНОГО КЛУБУ	13
1.1 Загальні поняття Web-сайту.....	13
1.2 Особливості розробки дизайну та програмної реалізації Web-сайту комп'ютерного клубу	19
Висновок до розділу 1	21
РОЗДІЛ 2. МОДЕЛЬ WEB-САЙТУ КОМП'ЮТЕРНОГО КЛУБУ	22
2.1 Методи, методології і технології розробки Web-сайту комп'ютерного клубу	22
2.2 Модель web-сайту комп'ютерного клубу	29
Висновок до розділу 2	37
РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ WEB-САЙТУ КОМП'ЮТЕРНОГО КЛУБУ	38
3.1 Програмна реалізація Web-сайту комп'ютерного клубу	38
3.2 Перевірка Web-сайту комп'ютерного клубу.....	50
3.3 Рекомендації з упровадження Web-сайту комп'ютерного клубу	53
Висновок до розділу 3	56
ВИСНОВКИ	57
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	58

ВСТУП

У сучасному світі комп'ютери і Інтернет стали невід'ємною частиною нашого життя. З кожним роком збільшується кількість людей, які користуються комп'ютерами та Інтернетом, а це в свою чергу призводить до зростання популярності комп'ютерних клубів. Такі заклади надають можливість користувачам отримати доступ до потужних комп'ютерів та високошвидкісного Інтернету, а також зустрітися з друзями та провести час весело та цікаво.

Основні переваги створення сайту для бізнесу:

1. Вся інформація зібрана в одному місці

Сайт для компанії – це її інформаційний центр, віртуальний офіс, своєрідний аналог Wikipedia, що дозволяє дати відвідувачам ресурсу структуровані дані про діяльність компанії, її товари або послуги.

- Сайт знайомить клієнта з компанією і ненав'язливо пояснює, чому саме ви варті його довіри.
- Непотрібні витрати на утримання штату консультантів, які будуть безкінечно відповідати на одні і ті ж питання.
- Клієнти в будь-який момент можуть ознайомитися з повним асортиментом товарів або послуг в каталозі.
- Дізнатися адреси торгових представництв і контактні номери, уточнити умови доставки та повернення, написати лист або замовити зворотний дзвінок – ваші потенційні клієнти в кілька кліків отримують доступ до будь-якої інформації, що їх цікавить.

2. Поліпшення репутації

Веб-сайт, оформлений у фірмовому, індивідуальному стилі, – це імідж компанії та її презентабельність. Для будь-якої сучасної компанії він є красномовним свідченням стабільності, успішності, амбітності і впевненості в майбутньому. Професійно зроблений ресурс, трендовий дизайн, бездоганне

юзабіліті – все це говорить про серйозність компанії і про шанобливе ставлення до клієнтів.

3. Більш ефективна реклама

Власний веб-ресурс для компанії є найкращим, однозначно більш вигідним і найбільш результативним способом реклами.

- Сайт можна використовувати як самостійний рекламний майданчик або як актуальне доповнення до масштабних рекламних кампаній в офлайні.
- Ніяких обмежень в подачі інформації та доступу до неї.
- Можливість з максимальною точністю оцінювати ефективність реклами за допомогою свого сайту.
- Висока рентабельність інвестицій. [1]

Актуальність теми дослідження полягає в розробці веб-сайту комп'ютерного клубу, що дозволить користувачам знайти всю необхідну інформацію про заклад, забронювати час на користування комп'ютерами, дізнатися про акції та новини клубу, а також зв'язатися з адміністрацією.

Мета дослідження: розробити дизайн та програмно реалізувати веб-сайту комп'ютерного клубу. Для досягнення поставленої мети необхідно було вирішити наступні завдання:

1. Проаналізувати теоретичні підходи питань розробки веб-сайту комп'ютерного клубу.
2. Розробити модель веб-сайту комп'ютерного клубу;
3. З'ясувати особливості використання програмно-технічних засобів для створення веб-сайту комп'ютерного клубу;
4. Розробити веб-сайт комп'ютерного клубу засобами web розробки: HTML, CSS, JS.

Об'єкт дослідження: процес розробки дизайну та програмна реалізація сайту комп'ютерного клубу.

Предмет дослідження: методи і технологія розробки веб-сайту комп'ютерного клубу.

Методи дослідження: пошуковий, аналітичний, статистичний, емпіричний, метод пошукового аналізу, метод порівняння, а також метод стратегічних пріоритетів, за допомогою яких , було виявлено найзручніший метод розробки сайту.

Для практичного вирішення поставлених задач було використано такі методи: загальнонауковий аналітичний метод; огляд існуючих технологій розробки Web-додатків та мов програмування; методи розробки та підключення бази даних тощо.

Практичне значення дослідження дипломної роботи полягає в наявності актуального та діючого веб-сайту, який може бути використаний у комерційних цілях.

Структура та обсяг випускної кваліфікаційної роботи. Випускна кваліфікаційна робота складається із вступу, трьох розділів, висновків, списку використаних джерел із 19 найменувань, містить 59 сторінок основного тексту і 36 рисунків.

РОЗДІЛ 1. ТЕОРЕТИЧНІ ПИТАННЯ ФОРМУВАННЯ ДИЗАЙНУ ТА ПРОГРАМНОЇ РЕАЛІЗАЦІЇ САЙТУ КОМП'ЮТЕРНОГО КЛУБУ

1.1 Загальні поняття Web-сайту

Web-сайт — це сукупність логічно зв'язаної гіпертекстової інформації, оформленої у вигляді окремих сторінок і доступної в мережі Інтернет. Подібне визначення веб-сайту було правильним на початку існування Інтернету, коли Мережа і веб-сайти використовувалися в основному як розважальна система. До кінця 90-х років веб-сайти дійсно були в основному статичними сторінками. Для створення веб-сайту було потрібне лише знання мови гіпертекстової розмітки — HTML. Якщо ж сторінка надавала якісь програмні засоби — це були виключно засоби, що міг надати сервер, на якому розташований веб-сайт. Про зручність і красу тогочасних веб-сайтів взагалі особливо не доводилося говорити. Час спливає, розвиваються мови програмування, розширюються канали передачі інформації. Зараз Інтернет вже є самодостатньою галуззю економіки, а веб-сайти стали повноправними представництвами фірм в Інтернеті. Мільйони людей вранці встають і йдуть на роботу в «Інтернет». Їх офісами є веб-сайти. [2]

Сучасне визначення Web-сайту говорить, що це сукупність програмних, інформаційних, а також медійних засобів, логічно пов'язаних між собою. По суті ж Web-сайт — це віддзеркалення успішності фірми, її обличчя.

Розглядають різні види веб-сайтів (рис.1.1), котрі виконують такі основні завдання:

- Реклама продукції, послуг, ідей. Правильно зроблений веб-сайт із легкістю приведе клієнта до висновку про необхідність покупки товару, або послуг, або ідей, що пропагуються на ньому;

- Продаж товарів, послуг, інформації, ідей. У сучасної людини немає багато часу для ходіння по магазинах. Тому можливість замовлення товарів і послуг, не відходячи від комп'ютера, значно розширює можливості і клієнта, і продавця;
- Безкоштовне надання інформації або послуг. Насправді надання інформації або послуг — це засіб залучення відвідувачів до даного ресурсу для здобуття, наприклад, статистичної інформації або ж для показу реклами, якщо це рекламний майданчик;
- Підтримка клієнтів.[3]

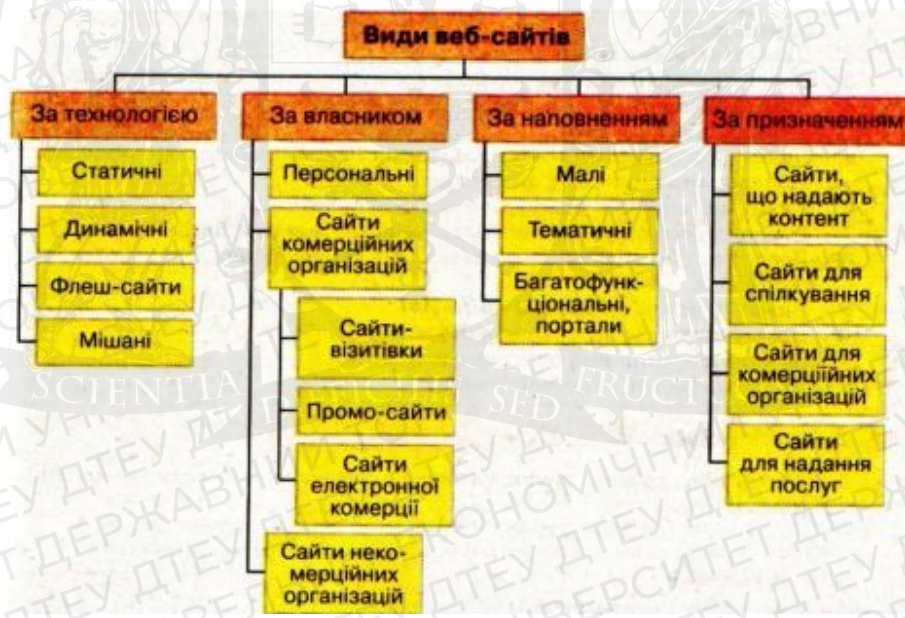


Рис. 1.1 Види веб-сайтів [3]

Це не повний перелік типів сайтів. Динамічний веб-сайт може змінюватися у процесі взаємодії з відвідувачем, а статичний надає всім відвідувачам однакову інформацію.

Сайти можна класифікувати за доступністю сервісів, фізичним розташуванням і призначенням (рис.1.2).

За **доступністю** сервісів їх поділяють на:

- Відкриті – усі сервіси повністю доступні для будь-яких відвідувачів і користувачів;
- Напіввідкриті – для доступу потрібно зареєструватися (зазвичай безкоштовно);
- Закриті – повністю закриті службові сайти організацій (наприклад, корпоративні сайти), особисті сайти приватних осіб. Такі сайти доступні для вузького кола людей. Доступ нових людей можливий через запрошення.

За **фізичним розташуванням**

Якщо сайт доступний користувачам з Інтернету, він вважається зовнішнім, натомість сайт, доступ до якого можуть здійснювати лише користувачі локальної мережі, є внутрішнім. Прикладами внутрішнього сайту можуть бути корпоративний сайт підприємства або сайт приватної особи в локальній мережі провайдера.

За **призначенням** сайти поділяють на :

- Бізнес-сайти – сайти, що містять інформацію про компанії та їхні послуги, здійснюють функцію електронної торгівлі;
- Інформаційні сайти – призначені для інформування відвідувачів, поширення новин, тематичні сайти, енциклопедії, словники тощо;
- Сайти соціальних мереж – інтерактивні багатокористувацькі веб-сайти, які наповнюються самими учасниками мережі. Сайт являє собою автоматизоване соціальне середовище, що дозволяє спілкуватися групі користувачів, об'єднаних спільним інтересом;
- Веб-портали – універсальні сайти, через які можна вийти на інші ресурси Інтернету;

- Сайти сервісів – сайти служб, які існують у мережі Інтернет, зокрема, сайти пошукових служб (Google, Bing), поштові сайти, веб-форуми, он-лайніві сховища даних (Skydrive), сайти служб онлайнного документообігу (Google Docs), зберігання та обробки фотографій (Picnik, ImageShack, Panoramio, Photobucket), зберігання відео (You Tube). [4]



Рис 1.2 Приклади веб-сторінок [3]

Зовнішній вигляд кожного сайту є унікальним, проте в усіх сайтів можна знайти спільні за функціональністю частини. На будь-якому сайті першою відкривається головна сторінка. Її розробці приділяють особливу увагу, оскільки дослідження показали, що люди не здатні читати інформацію, що відображається на моніторі, так уважно, як книжки або журнали. Вони зазвичай лише поверхово переглядають її, наприклад, як рекламу. Якщо головна сторінка містить те, що шукає відвідувач, він читає її далі, а якщо ні — переходить до інших сайтів, яких в Інтернеті дуже багато.

У верхній частині головної сторінки зазвичай розташована так звана шапка, яку дублюють на інших сторінках сайту. Це роблять спеціально, адже ця частина відображається у вікні браузера першою і відвідувач насамперед звертає увагу на неї. Щоб забезпечити швидкий перехід до основних тематичних розділів сайту, створюють меню сайту — список гіперпосилань на його розділи. Горизонтальне меню зазвичай розташовують у шапці, іноді дублюючи його в нижній частині сторінки, а вертикальне — переважно в лівій частині сторінки, у місці, звідки

відвідувач починає її переглядати. Меню є одним із найважливіших компонентів сайту, користувач постійно звертає на нього увагу, і тому вимоги до нього високі. Меню має бути зручним, помітним і зрозумілим, інакше користувач не знатиме, як перейти до потрібного розділу, і покине сайт. Пункти меню мають бути чітко відділені один від одного.

Гіперпосилання, розміщені в тексті чи у вигляді графічних об'єктів, дозволяють переходити на різні сторінки сайту або навіть на інші сайти. На сайтах із дуже великим обсягом інформації є сторінки третього рівня, а якщо необхідно — то й четвертого, п'ятого тощо.

Загалом виділяють три типи структур веб-сайтів — лінійну, деревоподібну та довільну. Подорожуючи сайтом із лінійною структурою, з головної сторінки ви перейдете на другу сторінку, з неї — на третю тощо. На сайті з деревоподібною структурою з головної сторінки можна потрапити на одну зі сторінок другого рівня, звідти – на одну зі сторінок третього рівня тощо. Сайт із довільною структурою видається зовсім неорганізованим, але саме в цьому й полягає принцип його створення. Подорожуючи таким сайтом, ви можете переходити з однієї його сторінки на інші в різні способи, і ваш шлях назад не обов'язково має бути таким самим. [5]

Вибір структури визначається особливостями завдань, що розв'язуються за допомогою вебсайту. Наведемо додатково чотири приклади структур сайту. Кожна із цих структур має свої недоліки та переваги у проектуванні веб-сайту.

Стандартна

Основна веб-сторінка містить посилання на інші документи вебсайту, а документи містять посилання, відповідно, на основну веб-сторінку. Це найпростіший і найпоширеніший спосіб (рис.1.3) організації веб-сайту.

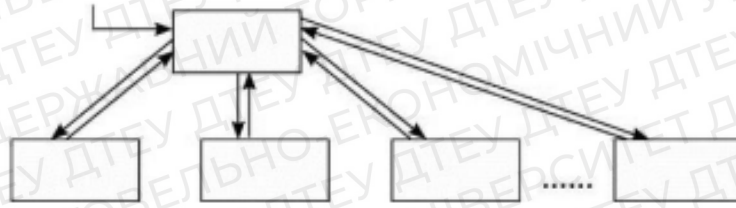


Рис. 1.3 Стандартна структура

Каскад

У цьому випадку посилання в документах задані таким чином, що існує тільки один шлях обходу сторінок вебсайту. За каскадного способу (рис.1.4) організації сторінок відвідувачі сайту можуть переміщуватися тільки в одному з напрямків — вперед або назад.



Рис. 1.4 Каскад

Хмарочос

У цій моделі (рис.1.5) відвідувачі можуть опинитися на деяких сторінках, тільки якщо вони йдуть правильним шляхом. Це нагадує підйом до потрібної кімнати у великому хмарочосі.



Рис. 1.5 Хмарочос

Павутина

У цьому випадку всі сторінки веб-сайту містять посилання на інші сторінки, і користувач може легко перейти з будь-якої сторінки практично на будь-яку іншу. Ця схема (рис.1.6) може перетворитися на лабіринт, якщо вийде з-під контролю, але вона популярна в тих випадках, коли посиланнями на документи користуються не надто часто.

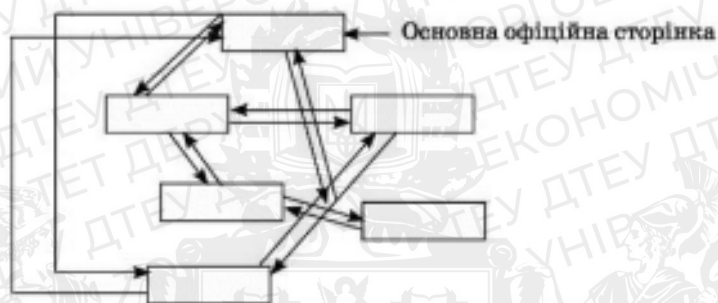


Рис. 1.6 Павутина

1.2 Особливості розробки дизайну та програмної реалізації Web-сайту комп'ютерного клубу

Web-сайт комп'ютерного клубу є важливим елементом, що допомагає клубу залучати нових клієнтів та забезпечує зручний спосіб для вже наявних клієнтів зв'язатися з клубом. Кожна веб сторінка відображає підхід до клієнтів та лице компанії в цілому. При переході на сайти уже відомих комп'ютерних клубів Києва можна бачити не досить приємну картину у вигляді дизайну, котрий застарів десятиліття тому. Після перегляду такої сторінки у користувача пропадає довіра до компанії й бажання використовувати її послуги. [6]

Можна виділити певні елементи, які можуть бути присутні на сторінці одного з таких Web-сайтів:

- Розклад роботи клубу та графік проведення турнірів та інших заходів. Розклад повинен бути простим та зрозумілим, а графік проведення заходів

повинен бути зручним для клієнтів та повинен включати детальну інформацію про турніри, в тому числі їх формат, правила та призи.

- Сторінка зі списком комп'ютерних ігор, що доступні для гри в клубі. Ця сторінка може містити описи ігор, їх характеристики та вартість години гри.
- Контактна інформація - як, наприклад, телефон та електронна пошта клубу, адреса клубу, реквізити для оплати тощо. Для зручності відвідувачів, клуб може додати на сайт онлайн-форму для резервації години гри. Окрім того, веб-сайт клубу може містити інформацію про різні програми та послуги, що надаються клубом, такі як тематичні турніри, навчання грамотному користуванню ПК, а також інші спеціальні пропозиції.
- Новини клубу - для збільшення залучення нових клієнтів, веб-сайт комп'ютерного клубу може містити блог, в якому будуть публікуватися новини клубу, інформація про ігри та їхні оновлення, а також поради щодо вдосконалення гри в різних іграх

Дизайн сайту має бути зручним та привабливим для відвідувачів, включаючи легко зрозумілу навігацію, дружній інтерфейс та адаптив.

Інший важливий аспект веб-сайту клубу - це безпека. Сайт повинен бути захищеним від зломів та атак, а також мати зручну систему резервного копіювання даних, щоб уникнути втрати важливої інформації.

Важливо врахувати, що веб-сайт комп'ютерного клубу має бути відкритим для взаємодії з користувачами. Клієнти повинні мати можливість залишати відгуки та рейтинги ігор, турнірів та інших подій, що відбуваються в клубі. Це може допомогти клубу у поліпшенні своїх послуг та залученні нових клієнтів.

Для успішного функціонування веб-сайту комп'ютерного клубу необхідно також забезпечити підтримку користувачів. Система підтримки повинна бути доступною для клієнтів 24/7, дозволяючи їм отримати допомогу в разі виникнення проблем з доступом до сайту, забування пароля, або будь-яких інших питань.

Одним із найважливіших аспектів веб-сайту є адаптив. Кожен Web-сайт, котрий орієнтований на взаємодію та залучення нових користувачів/клієнтів повинен мати мобільну версію або бути адаптивним до різних розмірів екранів, оскільки все більше людей використовують смартфони та планшети для доступу до інтернету.

Висновок до розділу 1

Веб-сайт комп'ютерного клубу є важливим інструментом для залучення та утримання клієнтів, а також для просування різних послуг та заходів, які надає клуб. При створенні веб-сайту необхідно враховувати потреби користувачів, забезпечувати безпеку та простоту використання, а також бути відкритим для взаємодії з користувачами та їхніх відгуків.

РОЗДІЛ 2. МОДЕЛЬ WEB-САЙТУ КОМП'ЮТЕРНОГО КЛУБУ

2.1 Методи, методології і технології розробки Web-сайту комп'ютерного клубу

Процес розробки програмного забезпечення складається з приблизно однакових етапів. Але як ці етапи будуть реалізовуватися, як вони пов'язані між собою та за якими принципами працюють – визначає методологія розробки ПЗ. Її завдання наступні:

- допомагає зрозуміти, як реалізувати ідею та створити продукт, корисний для бізнесу;
- дає чітке уявлення про методи роботи та план дій, що знижує ймовірність помилок під час розробки;
- допомагає заощадити на етапі розробки, мінімізуючи кількість правок та прискорюючи процес завдяки чіткому плану дій;
- дозволяє розподіляти завдання усередині команди залежно від обраного методу;
- створює послідовність процесів, формує вимоги до продукту.

Необхідність обирати методологію розробки програмного забезпечення обґрунтована націленістю на результат. В іншому випадку можна почати створювати продукт й ніколи його не закінчити, оскільки не буде чіткого плану дій. Але навіть такий метод можна охарактеризувати в методології, хоча вона буде менш ефективною. Існує чимало моделей, деякі компанії пропонують роботу за унікальними методиками. Але за основу завжди беруться 7 базових, перевірених роками та працюючих методик. [7]

1. Waterfall (Каскадна модель)

Каскадна модель є традиційною методологією розробки, в якій процес розробки поділяється на послідовні етапи. Кожен етап (аналіз, проектування, реалізація, тестування, випуск) завершується перед початком наступного. Ця методологія підходить для проектів зі стабільними вимогами, коли заздалегідь відомо, що потрібно реалізувати.

2. Agile (Гнучка методологія)

Agile є ітеративною та інкрементальною методологією, в якій розробка веб-сайту відбувається через короткі ітерації, відомі як спринти. Команда працює над невеликими завданнями протягом спринту, виконує їх, тестує і отримує зворотний зв'язок. Agile підходить для проектів, де вимоги можуть змінюватися або деталі не повністю визначені на початку.

3. Scrum

Scrum є одним з найпопулярніших фреймворків гнучкої розробки. Він передбачає розподіл роботи на короткі ітерації, які називаються спринтами. Кожен спринт триває від 1 до 4 тижнів, після чого виконується огляд і планування наступного спринта. Scrum використовує ролі, такі як Scrum Master і Product Owner, для керування процесом розробки. [8]

4. Lean (Легка методологія)

Lean є методологією, яка спрямована на мінімізацію витрат та максимізацію створення цінності. Вона підкреслює ефективне використання ресурсів, уникнення зайвої роботи та швидке впровадження змін на основі зворотного зв'язку. Lean методологія прагне створювати продукт, який найкращим чином задовольняє потреби клієнта.

5. DevOps

DevOps - це філософія розробки, яка об'єднує розробку та операції в один цілісний процес. Мета DevOps полягає в покращенні співпраці між розробниками та операторами, автоматизації процесів розробки, тестування, випуску та впровадження змін. DevOps сприяє швидкій

розробці, покращенню якості та забезпечує неперервну доставку програмного забезпечення.

6. Spiral (Спіральна модель)

Спіральна модель є ітеративною методологією, яка комбінує елементи каскадної моделі та прототипування. Вона передбачає постійний аналіз ризиків та оцінку ризиків на кожному етапі розробки. Спіральна модель дозволяє гнучко адаптуватися до змін, швидко прототипувати та навчатися на основі зворотного зв'язку.

7. Kanban

Методологія Kanban – це метод управління проєктами, який допомагає візуалізувати потік інтелектуальної роботи та скорочувати число незавершених завдань. Розповідаємо, як налаштувати комунікацію і збільшити продуктивність роботи команди по kanban-методу, що таке kanban-дошки та який сервіс обрати. [9]

Розробка сайту це досить громіздкий процес який включає в себе великий спектр роботи та етапів, котрі ведуть до фінального результату.

Основні етапи, що застосовуються при розробці веб-сайту:

1. Планування та аналіз:

- Визначення мети та цілей веб-сайту комп'ютерного клубу.
- Збір вимог та аналіз конкурентів.
- Створення структури сайту та розробка контенту.

2. Дизайн та макетування:

- Розробка вигляду та стилістики сайту, включаючи кольорову палітру та типографіку.

- Створення макетів інтерфейсу (наприклад, за допомогою графічних редакторів або спеціалізованих інструментів для макетування).

3. Розробка фронтенду:

- Використання мов програмування, таких як HTML, CSS і JavaScript, для створення інтерактивних елементів сайту.

- Впровадження адаптивного дизайну, щоб сайт виглядав добре на різних пристроях та розмірах екрану.

4. Розробка бекенду:

- Використання серверних технологій, таких як PHP, Python, Ruby або Node.js, для розробки функціональності сайту.

- Робота з базами даних (наприклад, MySQL, PostgreSQL або MongoDB) для зберігання та керування даними.

5. Інтеграція:

- Підключення сторонніх сервісів, таких як система управління контентом (наприклад, WordPress або Drupal), системи оплати, соціальні мережі тощо.

- Використання API (Application Programming Interface) для обміну даними між різними додатками та сервісами.

6. Тестування та оптимізація:

- Виконання тестування функціональності та відповідності дизайну.

Існують два основні методи розробки сайтів:

- **Html/CSS/JavaScript**
- **Конструктори сайтів**

З одного боку, конструктори сайтів є легким та доступним способом створення сайту, проте вони не такі гнучкі для створення функціоналу та анімацій й недостатньо безпечні з точки зору використання плагінів, а також засобів захисту.

Щоб досягнути максимальної продуктивності й бажаного функціоналу сайту, рекомендовано використовувати більш професійний шлях повноцінної розробки. Хоч цей шлях і складніший, існують різні фреймворки й бібліотеки, які полегшують життя розробників. Ці технології містять в собі уже готові строки робочого коду, який підходить під ті чи інші потреби розробника. [10]

Більшість фронтенд розробників надають перевагу трьом популярним фреймворкам, якими користуються у різних проектах:

1. **React.js**: React.js є одним з найпопулярніших фреймворків для розробки фронтенду. Він дозволяє розробникам створювати компонентну структуру веб-додатків, яка забезпечує легку перевикористання коду та покращує продуктивність. React.js також пропонує віртуальний DOM, що забезпечує швидку перерендеринг та покращує продуктивність додатків. [11]

2. **Angular**: Angular є повнофункціональним фреймворком, розробленим компанією Google. Він надає розширені можливості для створення складних веб-додатків та забезпечує широкий набір інструментів для розробників. Angular використовує концепцію двостороннього зв'язку даних та пропонує розширений набір функціональності, включаючи маршрутизацію, форми та обробку подій. [12]

2. **Vue.js:** Vue.js є прогресивним фреймворком для розробки фронтенду, який поєднує простоту використання з потужними можливостями. Він дозволяє створювати реактивні компоненти та легко інтегрувати їх у існуючі проекти. Vue.js має легку вагу та швидкодію, що робить його привабливим вибором для багатьох розробників. [13]

Серед популярних бібліотек виділяють наступні:

1. **jQuery:** jQuery є однією з найпопулярніших бібліотек JavaScript. Вона спрощує маніпулювання HTML-документами, обробку подій, анімацію та взаємодію з сервером за допомогою AJAX. jQuery забезпечує короткий та зрозумілий синтаксис, що спрощує розробку фронтенду. [14]

2. **Lodash:** Lodash є утилітною бібліотекою JavaScript, яка надає безліч корисних функцій для роботи з масивами, об'єктами, рядками та іншими типами даних. Вона спрощує маніпулювання даними та забезпечує оптимізацію продуктивності.

3. **Axios:** Axios є популярною бібліотекою JavaScript для здійснення HTTP-запитів з браузера або Node.js. Вона надає простий та зрозумілий інтерфейс для взаємодії з веб-серверами та отримання асинхронних даних.

4. **Moment.js:** Moment.js є потужною бібліотекою для роботи з датами та часом у JavaScript. Вона надає зручний спосіб форматування, парсингу, валідації та маніпулювання датами. [15]

5. **React.js:** React.js може також бути розглянутий як бібліотека, яка полегшує розробку фронтенду. Він забезпечує компонентну модель, віртуальний DOM та багато інших функцій для швидкого рендерингу та управління станом веб-додатків. [11]

Написання коду має свої особливості, в залежності від методології CSS, яку обрав розробник. Всі нижче наведені методології забезпечують структурованість, повторне використання та підтримку коду :

1. CSS БЕМ (Блок, елемент, модифікатор)

CSS БЕМ – популярна методологія, що використовує іменування класів для структурування та організації CSS коду. За цією методологією кожен елемент сторінки поділяється на блоки (більші компоненти), елементи (частини блоків) та модифікатори (зміни в стилі). Наприклад, `.header` - це блок, `.header__logo` - елемент, а `.header--dark` - модифікатор. [16]

2. SMACSS (Scalable and Modular Architecture for CSS)

SMACSS - це методологія, яка розділяє CSS код на п'ять категорій: базовий, макет, модульний, стейт та тема. Кожна категорія має свої правила та методики, що допомагають забезпечити модульність, масштабованість та повторне використання коду. SMACSS наголошує на структурі та організації CSS файлів. [16]

3. OOCSS (Object-Oriented CSS)

OOCSS - це методологія, яка використовує об'єктно-орієнтований підхід до написання CSS коду. Вона пропонує виділяти повторювані стилі в окремі класи-об'єкти, які можуть бути використані на різних елементах та компонентах. OOCSS сприяє повторному використанню стилів та підтримці коду. [17]

4. CSS-in-JS

CSS-in-JS - це підхід до написання CSS, коли стилі вбудовуються безпосередньо в JavaScript код. За допомогою CSS-in-JS бібліотек, таких як `Styled Components` або `Emotion`, можна створювати компоненти, які включають CSS стилі без необхідності відокремленого CSS

У процесі створення сайту комп'ютерного клубу використовується саме методологія написання CSS БЕМ, тому що вона зручна для структурування елементів, робить код легким для читання для інших розробників, що грає важливу роль під час написання коду у команді або самотужки.

2.2 Модель web-сайту комп'ютерного клубу

Існує багато типів веб-сайтів, кожен з яких має свою специфіку та призначення. Основні типи веб-сайтів включають:

1. **Інформаційний сайт** - містить інформацію про компанію, продукти чи послуги, новини та іншу корисну інформацію.
2. **Електронний магазин** - містить каталог продуктів, ціни та інформацію про замовлення та оплату товарів.
3. **Блог** - містить статті, коментарі та інформацію, що може зацікавити відвідувачів.
4. **Соціальна мережа** - платформа для обміну інформацією, спілкування та створення спільнот.
5. **Форум** - місце для дискусій та обговорення певної теми чи питання.
6. **Портфоліо** - містить роботи, створені відвідувачем (фото, відео, дизайн, ілюстрації та інші).
7. **Відеохостинг** - платформа для завантаження та перегляду відео.
8. **Пошуковий сайт** - допомагає в знаходженні та відображенні результатів пошуку.
9. **Онлайн-ігри** - сайти, де відвідувачі можуть грати в онлайн-ігри.
10. **Інтернет-портал** - збірна точка для доступу до різних послуг та ресурсів в Інтернеті. [18]

Крім цього, можна виділити ще багато інших типів веб-сайтів, які мають свої унікальні особливості та призначення, наприклад, сайти-агрегатори новин, онлайн-платформи для навчання, медіа-сайти та інші.

Перед створенням моделі web-сайту комп'ютерного клубу потрібно визначитись який з наведених вище типів найбільше підходить замовнику.

У даному випадку найбільше підходить саме інформаційний сайт, тобто "landing-page". Це спеціальний тип веб-сайту, який має на меті залучення максимальної кількості клієнтів чи покупців певного продукту або послуги. Landing-page (надалі лендинг-сторінка) має максимально простий та зрозумілий дизайн, який спрямований на переконання відвідувача в необхідності придбання продукту або послуги.

Такий веб-сайт має конкретну мету - залучення цільової аудиторії та максимальної кількості конверсій (тобто перетворення відвідувачів на клієнтів або покупців). Він може бути створений для запуску рекламних кампаній, просування нового продукту або послуги, просування бренду тощо.

Лендинг-сторінки зазвичай складаються з однієї об'ємної сторінки, де зосереджена вся необхідна інформація про продукт чи послугу, а також з декількох викликів до дії, які мають переконати відвідувачів здійснити покупку. Такими викликами можуть бути кнопки замовлення, форми зворотного зв'язку, пропозиції на отримання безкоштовної консультації тощо.

Особливості якісної лендинг-сторінки:

- Яскравий та привабливий дизайн: Важливо, щоб дизайн сторінки був привабливим та сучасним, він має захоплювати увагу відвідувача та підкреслювати цінність товару або послуги, що пропонуються.
- Зрозумілий та лаконічний контент: Контент лендинг сторінки має бути зрозумілим та лаконічним. Важливо звернути увагу на заголовки, теги та короткі описи, які мають передавати ключову інформацію та заохочувати відвідувача продовжувати досліджувати сторінку.

- Чітка та проста структура: Структура сторінки має бути чіткою та простою. Інформацію слід подавати в послідовності, яка допоможе відвідувачеві розуміти логіку та послугу, яку пропонує компанія.
- Використання відео та графіки: Відео та графіка можуть допомогти зрозуміти та продемонструвати користь продукту або послуги. Якщо використовувати відео, то воно має бути коротким та показати користь, яку можна отримати від використання продукту або послуги.
- Мобільна оптимізація: Оскільки все більше людей використовують мобільні пристрої для перегляду веб-сайтів, лендінг повинен бути оптимізований для мобільних пристроїв. Це забезпечить комфортний перегляд і збереження уваги відвідувача.
- Тестування та аналіз: Успішна лендінг сторінка потребує постійного тестування та аналізу результатів. Тестування може допомогти виявити найбільш ефективні варіанти дизайну та контенту, а аналіз результатів дозволить вдосконалювати та покращувати сторінку з часом. [19]

Лендінг комп'ютерного клубу, котрий має назву «ALTRONE» має інтуїтивно зрозумілий та приємний дизайн. Сторінка сайту має наступні розділи:

- Про нас
- Ігри
- Послуги
- Новини
- Контакти

Під час написання коду використано правила семантики сайту, тобто всі розділи розподілені по секціям, які визначаються окремими тегамі: <header>, <nav>, <main>, <footer>, та іншими проміжними (<section>, <article>)

1. Header веб-сайту

Складається з унікального логотипу (рис.2.1), котрий включає в себе назву комп'ютерного клубу та зображення, яке може асоціюватися з ігровою індустрією. По своїй суті він є простим, але може бути впізнаваним серед цільової аудиторії.



Рис. 2.1 Логотип клубу

2. Навігація

Наступним елементом хедеру є навігація під тегом <nav> (рис.2.2). Кнопки навігації є «якорями», при натисканні на які, користувача переносить на відповідний розділ сайту. Окрім кнопок навігації є функція зміни мови на сторінці.

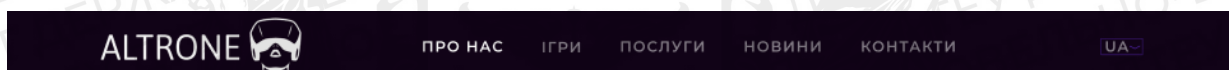


Рис. 2.2 Секція навігації сайту

3. Основна частина хедеру

Після завантаження сайту, користувач буде бачити вище перераховані елементи, а також лаконічну інформацію по сторінці (рис. 2.3), на котру він перейшов, тобто перед користувачем буде:

- назва сайту
- інформація, що це за сайт
- кнопка «забронювати місце»
- спонсори комп'ютерного клубу
- фонове зображення клубу

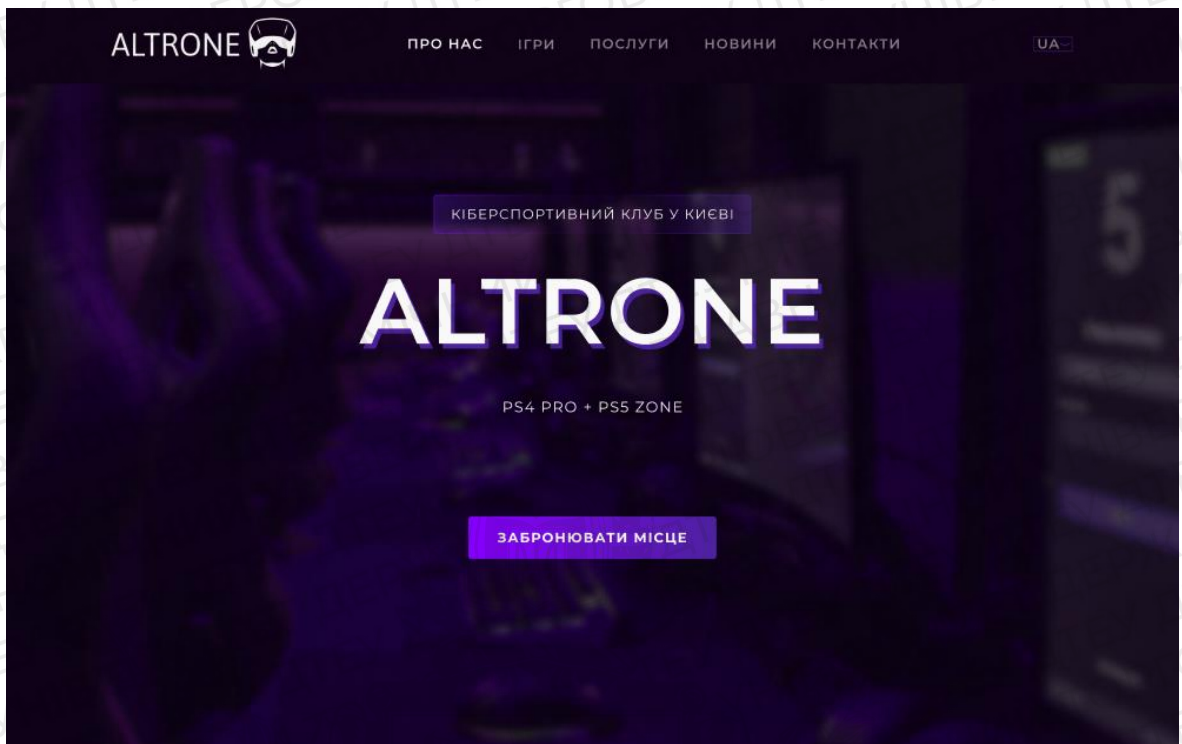


Рис. 2.3 Загальний хедер сайту

4. Секція «Про нас»

Розділ «Про нас» є початком тегу <main>. У цьому блоці (рис. 2.4) будуть короткі тези про переваги клубу та вказані деякі послуги, які будуть надаватись клієнтам.

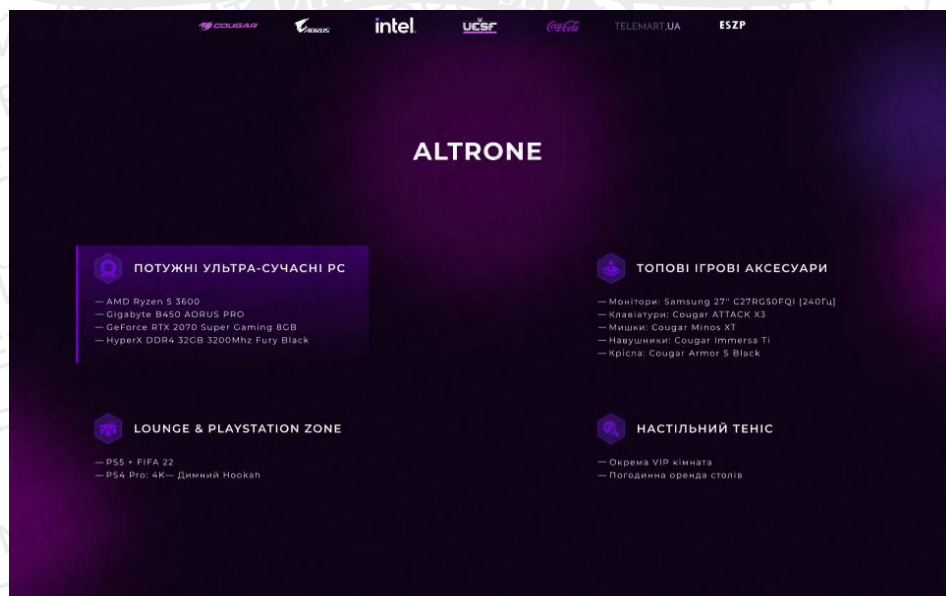


Рис. 2.4 Зображення блоку «Про нас»

5. Секція «Ігри»

Розділ (рис. 2.3) містить в собі деяку кількість найпопулярніших ігор, котрі обирають клієнти в комп'ютерному клубі. При наведенні на картинку, на ній з'являється короткий опис гри для новачків, які раніше не грали. Також, на картинку можна натиснути, щоб перейти на сторінку гри в відомому сервісі по діджитал дистрибуції ігор «Steam», де користувач зможе детальніше ознайомитися з обраною грою. Також у цьому розділі буде ще одна кнопка «Забронювати місце», для спонукання користувачів відвідати саме наш клуб.

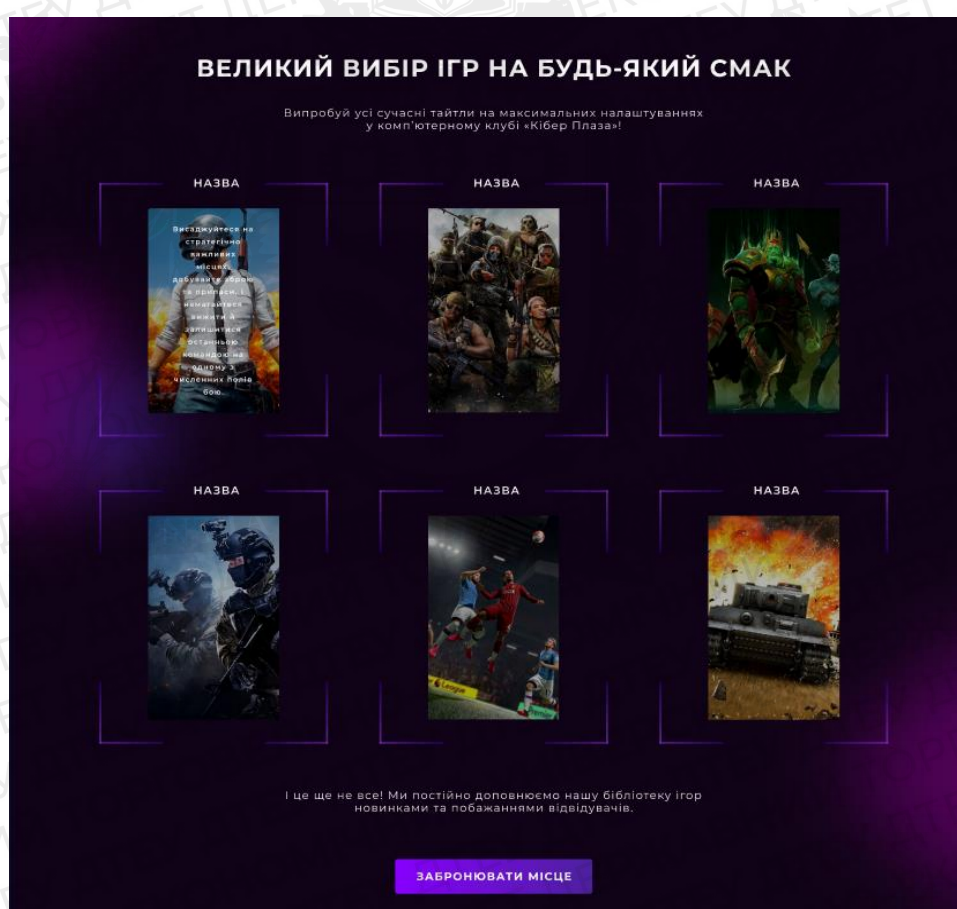
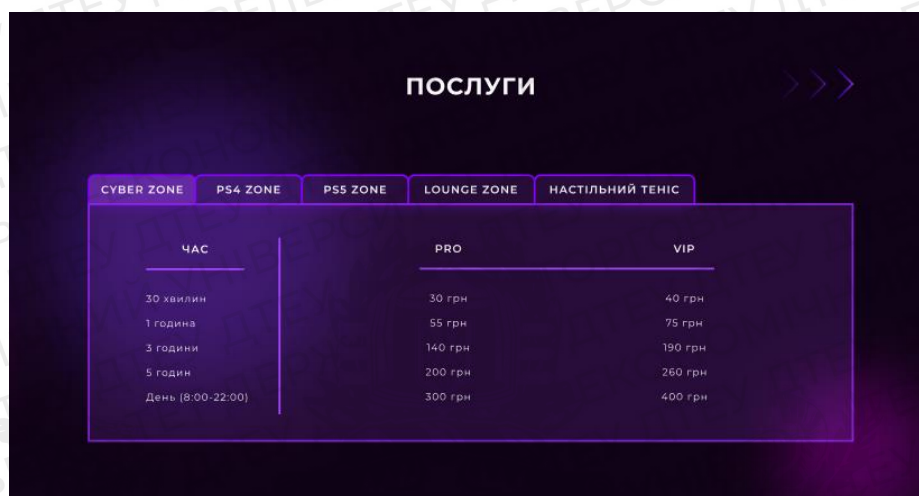


Рис. 2.5 Зображення блоку «Ігри»

6. Секція «Послуги»

Один із головних розділів, який є причиною чому більшість користувачів шукають подібні веб-сайти. Цей розділ (рис. 2.6) містить в собі інформацію про наявні послуги, та ціни на надання цих послуг.

Послуги поділені на маленькі секції, кожна з яких є активною кнопкою перемикання таблиць з цінами, де розписано ціни, відповідно до часу користування, наприклад PS5 чи окремим ігровим місцем.



The screenshot shows a website interface for 'ПОСЛУГИ' (Services). At the top, there are five tabs: 'CYBER ZONE', 'PS4 ZONE', 'PS5 ZONE', 'LOUNGE ZONE', and 'НАСТІЛЬНИЙ ТЕНІС'. The 'PS5 ZONE' tab is selected. Below the tabs is a table with three columns: 'ЧАС' (Time), 'PRO', and 'VIP'. The table lists prices for different durations: 30 minutes, 1 hour, 3 hours, 5 hours, and a full day (8:00-22:00).

ЧАС	PRO	VIP
30 хвилин	30 грн	40 грн
1 година	55 грн	75 грн
3 години	140 грн	190 грн
5 годин	200 грн	260 грн
День (8:00-22:00)	300 грн	400 грн

Рис. 2.6 Зображення послуг та цін комп'ютерного клубу

7. Секція «Новини»

Особливістю клубу є тісна взаємодія зі спонсорами, тому наявність турнірів збільшить кількість відвідувачів та бажаючих позмагатися у тій чи іншій кібердисципліні, щоб перевірити свої навички, а також отримати неабиякі емоції від гри. У розділі «Новини» (рис. 2.7) публікується актуальна інформація з приводу проведення турнірів, а також інша важлива інформація, така як: акції, оновлення, набір в команду, та інші цікаві статті.

Дизайн стрічки новин дуже лаконічний, а динаміка додана за рахунок анімації наведення.

Кнопка «Детальніше» перенесе користувача на сторінку, пов'язану з доданою новиною.

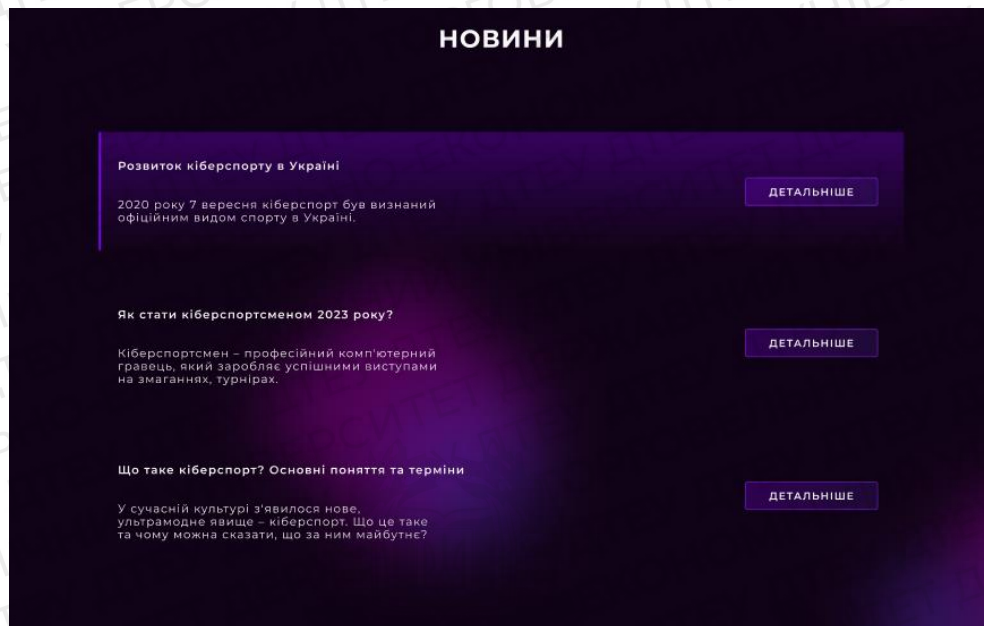


Рис. 2.7 Зображення розділу «Новини»

8. Футер сайту

Футер (рис. 2.8) складається з наступних елементів:

- Логотип комп'ютерного клубу.
- Кнопки навігації (це додає зручності користування сайтом, адже з футера користувач завжди може перейти на розділ, який його цікавить).
- Контактна інформація (адреса, електронна пошта, номер телефону).
- Соціальні мережі клубу, представлені в вигляді зображень, на які можна натиснути, щоб перейти на обрану соціальну мережу.
- Кнопка «Вгору», яка дозволяє користувачу одразу переміститися до хедеру сайту, ця кнопка завжди знаходиться в правому нижньому куті екрану.

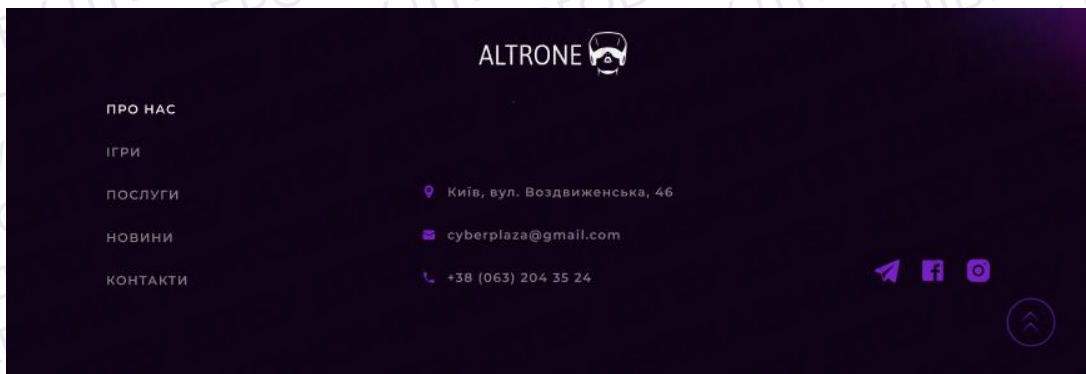


Рис. 2.8 Зображення футеру сайту

Висновок до розділу 2

Дизайн виконано в темному кольорі, щоб зменшити навантаження на очі користувачів. Поєднання чорного з білим завжди було хорошим мінімалістичним рішенням при створенні сайтів, у даному випадку використано чорний з фіолетовим відтінком та білим шрифтом Montserrat. Створено анімації об'єктів, щоб додати більше динаміки сайту. Задній фон сторінки, не включаючи картинку хедера, створений та стилізований тільки за рахунок CSS коду.

Під час розробки сайту використано методологію Scrum, а код написано на методології БЕМ, для кращої структуризації елементів. Використана бібліотека jQuery для реалізації плавного переходу по розділах навігації. Якщо замовник забажає додати більше динаміки та, наприклад особистий кабінет для користувачів, це можна буде виконати на React.js. За допомогою JavaScript повністю локалізовано сайт.

РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ WEB-САЙТУ КОМП'ЮТЕРНОГО КЛУБУ

3.1 Програмна реалізація Web-сайту комп'ютерного клубу

1. Header сайту

У блоці header реалізовано навігаційну панель яка зберігає свою позицію при скролінгу web-сторінки. Код (рис. 3.1) написано на JavaScript

```
76 // Sticky-header
77 (function () {
78     const header = document.querySelector('.header');
79     window.onscroll = () => {
80         if (window.pageYOffset > 50) {
81             header.classList.add('header_active');
82         }
83         else {
84             header.classList.remove('header_active');
85         }
86     };
87 }());
```

Рис. 3.1 Навігаційна панель на JavaScript

Функція працює наступним чином:

1. Задано змінну “header”, яка визначається селектором “.header”
2. Вказано подію «window.onscroll»
3. Додано умови для події: якщо сторінка скролиться вниз на 50 пікселів, то до елемента із класом “.header” додається ще один клас “.header_active”

4. “.header_active” стилізується (рис. 3.2) в CSS файлі й використовується при активації скрол функції. Фон змінюється з прозорого на колір заданий кольором #110317. Також додається тінь для всього блоку.

```
.header_active{
  background: #110317;
  box-shadow: 0 0 20px #110317;
}
```

Рис. 3.2 Стилізація “.header_active”

5. Вмістом else є видалення класу “.header_active”.

2. Випадаючий список для зміни мови

Простий випадаючий список під тегом <select> неможливо унікально стилізувати тільки за допомогою CSS, тому цей список реалізовано за допомогою звичайних блоків, а стилізація описана в CSS та JavaScript файлах.

1. Створено розмітку (рис. 3.3) для випадаючого списку в HTML, котрий містить дві мови: українську та англійську

```
52 <div class="language">
53 <div class="select" id="select">
54 <div class="select_header">
55 <span class="select_current">ua</span>
56 <div class="select_icon">&rang</div>
57 </div>
58 <div class="select_body">
59 <div class="select_item" id="ua" >ua</div>
60 <div class="select_item" id="en" >en</div>
61 </div>
62 </div>
63 </div>
```

Рис. 3.3 Розмітка випадаючого списку

2. Стилiзований список (рис. 3.4) в CSS

```
174 .select{
175   position: relative;
176   display: flex;
177   width: 75px;
178   height: 25px;
179   align-items: center;
180   font-weight: 600;
181   font-size: 22px;
182   text-transform: uppercase;
183   color: #F9F9F9;
184   opacity: .5;
185 }
186 .select.is-active .select_body{
187   display: block;
188 }
189 .select_header{
190   border: 1px solid #4D229E;
191   cursor: pointer;
192   display: flex;
193   width: 80px;
194   height: 25px;
195   padding-left: 4px;
196   box-shadow: 0px 4px 4px rgba(0, 0, 0, 0.25);
197 }
198 .select_current{
199   font-weight: 600;
200   font-size: 22px;
201   text-transform: uppercase;
202   padding-left: 4px;
203 }
204 .select_header:hover>.select_icon{
205   color: #F9F9F9;
206   transition: .3s;
207 }
208 .select_header:hover,.select_header:active,.select_header:focus{
209   background-color: #4D229E;
210   transition: .3s;
211 }
212 .select_icon{
213   align-items: center;
214   display: flex;
215   flex-shrink: 0;
216   justify-content: center;
217   transform: rotate(90deg);
218   height: 15px;
219   padding-bottom: 2px;
220   margin: auto auto;
221   text-align: center;
222   color: #4D229E;
223 }
224 .select_body{
225   border: 1px solid #4D229E;
226   border-top: 0;
227   position: absolute;
228   left: 0;
229   right: 0;
230   top: 100%;
231   display: none;
232 }
233 .select_item{
234   padding-left: 8px;
235   cursor: pointer;
236 }
237 }
238 .select:hover,.select:active,.select:focus{
239   opacity: 1;
240   transition: .3s;
241 }
242 .select_item:hover,.select_item:active,.select_item:focus{
243   background-color: #4D229E;
244   transition: .3s;
```

Рис. 3.4 Стилiзацiя списку

3. Реалiзацiя списку (рис. 3.5) за допомогою JavaScript

1. Встановлено обробники подiй на елементах управлiння вибором. Обробники подiй виконуються при натисканнi на елементи вибору.
2. Функцiя **selectToggle** викликається при натисканнi на елемент з класом **“.select_header”**. Вона переключає клас **“.is-active”** для батькiвського елемента, щоб показувати або приховувати випадаючий список.
3. Функцiя **selectChoose** викликається при натисканнi на елемент з класом **“.select_item.”** Вона оновлює вибране значення в елементi **“.select_current”**, приховує випадаючий список та зберiгає вибране значення в хешi адресної строки. Пiсля цього сторiнка перезавантажується.


```

15     selectHeader.forEach(item => {
16         item.addEventListener('click', selectToggle);
17     });
18
19     selectItem.forEach(item => {
20         item.addEventListener('click', selectChoose);
21     });
22
23     function selectToggle() {
24         this.parentElement.classList.toggle('is-active');
25     }
26
27     function selectChoose() {
28         let text = this.innerText,
29             select = this.closest('.select'),
30             currentText = select.querySelector('.select__current');
31         currentText.innerText = text;
32         select.classList.remove('is-active');
33
34         let itemId = this.getAttribute('id');
35         window.location.hash = itemId;
36         window.location.reload();
37
38         // Збереження значення id у localStorage
39         localStorage.setItem('selectedItemId', itemId);
40     }

```

Рис. 3.5 JavaScript реалізація списку

4. Локалізація сайту реалізована за допомогою окремого JavaScript файлу, котрий містить масив (рис. 3.6) з перекладом усіх елементів на сторінці. Елементи згруповані в масив, кожен елемент має свій ключ, тобто частину селектора “.lng-“+ hash.

```

1  const langArr = {
2      "header-about": {
3          "en": "About us",
4          "ua": "Про нас"
5      },
6      "header-games": {
7          "en": "Games",
8          "ua": "Ігри"
9      },
10     "header-services": {
11         "en": "Services",
12         "ua": "Послуги"
13     },

```

Рис. 3.6 Мультимовний масив

5. Hash виконує ключову роль у алгоритмі зміни мови (рис. 3.7), оскільки в залежності від вибраного елемента “.select__current” в адресну строку буде виводитись hash, тобто id елемента з класом “.select__item”, який і визначає якою мовою буде відображатися веб-сторінка.

```
41 ..... window.addEventListener('DOMContentLoaded', function () {
42 .....     let hash = window.location.hash.substring(1);
43 .....     if (hash) {
44 .....         let selectedItem = document.getElementById(hash);
45 .....         if (selectedItem) {
46 .....             let selectedText = selectedItem.innerHTML;
47 .....             selectCurrent.innerHTML = selectedText;
48 .....             // Збереження значення id у localStorage при завантаженні
49 .....             // сторінки
49 .....             localStorage.setItem('selectedItemId', hash);
50 .....         } else {
51 .....             // Якщо хеш не відповідає жодному елементу select__item,
52 .....             // встановлюємо дефолтне значення id
52 .....             window.location.hash = 'ua';
53 .....             selectCurrent.innerHTML = 'UA';
54 .....             localStorage.setItem('selectedItemId', 'ua');
55 .....             window.location.reload();
56 .....         }
57 .....     }
58 .....     // Зміна мови
59 .....     for (let key in langArr) {
60 .....         document.querySelector('.lng-' + key).innerHTML = langArr[key]
61 .....         [hash];
62 .....         let elem = document.querySelector('.lng-' + key);
63 .....         if (elem) {
64 .....             elem.innerHTML = langArr[key][hash];
65 .....         }
66 .....         if (key === undefined) {
67 .....             elem.innerHTML = langArr[key][hash];
68 .....             window.location.reload();
69 .....         }
70 .....     }
71 ..... }
72 ..... });
73 ..... select();
```

Рис. 3.7 Реалізація зміни мови

1. При завантаженні сторінки, подія **DOMContentLoaded** викликається для виконання коду в середині обробника подій.
2. Функція отримує хеш з адресної строки за допомогою **window.location.hash.substring(1)**. Якщо хеш існує, вона знаходить елемент зі відповідним **id** і оновлює вибране значення в елементі **.select__current**.

3. Якщо хеш не відповідає жодному елементу `.select_item`, за замовчуванням встановлюється значення `ua` і перезавантажується сторінка.

4. Після завершення перевірки хешу, функція виконує цикл `for...in`, який пробігає через всі ключі об'єкту `langArr`. В циклі відбувається оновлення вмісту елементів з класом `.lng-<key>` залежно від значення хешу. Якщо елемент з відповідним класом не знайдено, перезавантажується сторінка.

Ця функція дозволяє вибирати значення зі списку, зберігати вибране значення в хеші адресної строки та оновлювати вміст сторінки відповідно до вибраної мови.

4. Плавний scroll web-сторінки

Для зручності та лаконічності написання коду, а також одним із рішень для усунення конфлікту хешів адресної строки, підключено бібліотеку jQuery та спеціальний плагін плавного скролу сторінки.

1. Встановлено відповідні id в атрибут `href=""` (рис. 3.8) для елементів навігаційної панелі, логотипу (в шапці сайту), а також для кнопки «наверх», яка зафіксована в правому нижньому куті сторінки

```
24 ..... <ul class="header_list" >
25 ..... <li class="header_item">
26 ..... <a href="#main" rel="m_PageScroll2id"
27 ..... class="header_item-link js-scroll lng-header-about">
28 ..... Про нас
29 ..... </a>
30 ..... </li>
31 ..... <li class="header_item">
32 ..... <a href="#games" rel="m_PageScroll2id"
33 ..... class="header_item-link js-scroll lng-header-games">
34 ..... Ігри
35 ..... </a>
36 ..... </li>
37 ..... <li class="header_item">
38 ..... <a href="#services" rel="m_PageScroll2id"
39 ..... class="header_item-link js-scroll lng-header-services">
40 ..... Послуги
41 ..... </a>
```

Рис. 3.8 Шляхи для якорів сайту

2. Підключено бібліотеку jQuery (рис. 3.9), а також плагін плавного скролу сторінки

```
605 <script src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.0/jquery.min.js"></script>
606 <script src="./libs/PageScroll2id.min.js"></script>
```

Рис. 3.9 Підключення jQuery

3. У файлі main.js (загальний файл для написання функцій web-сторінки) прописано уже готову функцію (рис. 3.10), яка міститься у раніше підключеній бібліотеці jQuery:

```
jQuery(document).ready(function (){
    $("a[rel='m_PageScroll2id']").mPageScroll2id({
        offset:120
    });
});
```

Рис. 3.10 Підключення функції плавного скролу

4. Встановлено значення “.m_PageScroll2id” (рис. 3.11) для атрибута “rel” елементів навігації, логотипу і кнопки, щоб привести у дію плагін

```
24 <ul class="header_list" >
25 <li class="header_item">
26 <a href="#main" rel="m_PageScroll2id"
   class="header_item-link js-scroll lng-header-about">
27     Про нас
28 </a>
29 </li>
30 <li class="header_item">
31 <a href="#games" rel="m_PageScroll2id"
   class="header_item-link js-scroll lng-header-games">
32     Ігри
33 </a>
34 </li>
35 <li class="header_item">
36 <a href="#services" rel="m_PageScroll2id"
   class="header_item-link js-scroll lng-header-services">
37     Послуги
38 </a>
```

Рис. 3.11 Встановлення атрибута rel

5. Додано “offset: 120” (рис. 3.12, рис. 3.13), щоб прибрати недолік анімованої шапки профіля, яка за рахунок position: fixed, виходить з потоку елементів й перекриває блок при автоматичному скролі сторінки

а) Відсутній offset

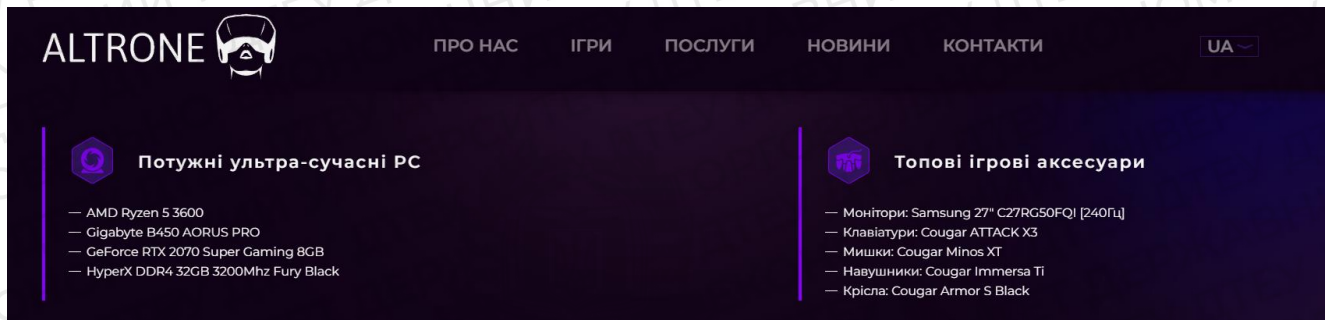


Рис. 3.12 Offset: 0

б) Наявний offset

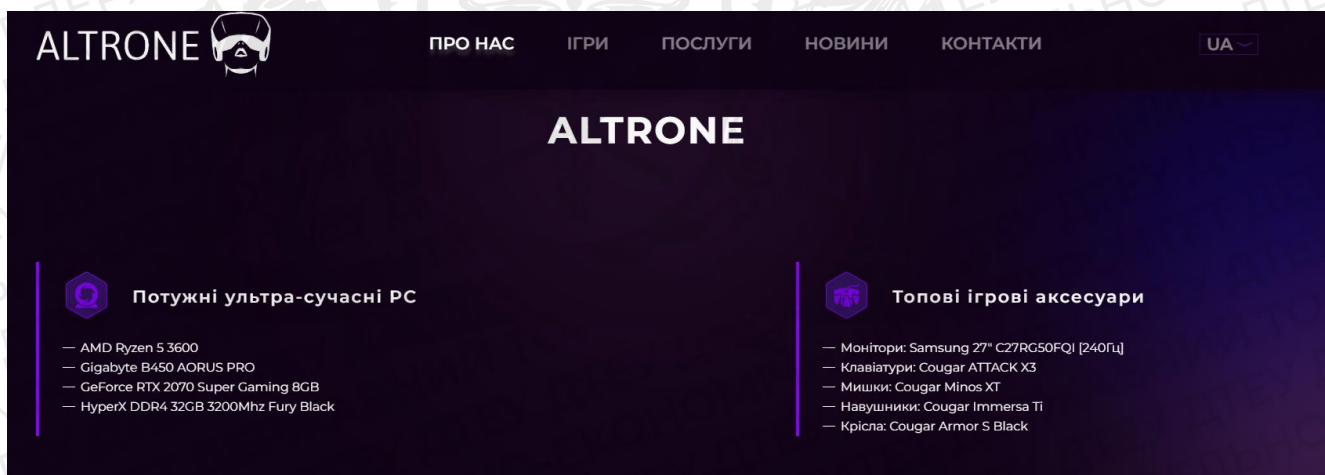


Рис. 3.13 Offset: 120

4. Блок «Ігри»

Даний блок реалізовано за допомогою grid та flex властивостей та анімованими блоками в CSS.

1. Розмітка Html (рис. 3.14) має наступний вигляд:

а) Картка знаходить в обгортці з класом “.wgraper”, яка використовується у всьому файлі, щоб обмежувати довжину блоків та однаково вирівнювати їх по центру.

- б) У обгортці знаходить блок “.games__container”, котрий буде відображатися flex-елементом.
- в) Наступним дочернім елементом є “.games__cards”, який буде містити сітку з 6 карток.
- г) Карткам надано: заголовок, посилання та картинку.

```

156 ..... <section class="games">
157 .....   <div class="games__info" id="games">
158 .....     <h2 class="games__title lng-games-title">
159 .....       Великий вибір ігор на будь-який смак
160 .....     </h2>
161 .....     <p class="games__subtitle lng-games-subtitle">
162 .....       Випробуй усі сучасні тайтли на максимальних налаштуваннях
163 .....       комп'ютерному клубі «Altrone»!
164 .....     </p>
165 .....   </div>
166 .....   <div class="main_lights-violet--4">
167 .....   </div>
168 .....   <div class="main_lights-violet--5">
169 .....   </div>
170 .....   <div class="main_lights-violet--6">
171 .....   </div>
172 .....   <div class="main_lights-blue--2">
173 .....   </div>
174 .....   <div class="wrapper">
175 .....     <div class="games__container">
176 .....       <div class="games__cards">
177 .....         <div class="games__card">
178 .....           <h3 class="games__card-title lng-card-1">
179 .....             pubg
180 .....           </h3>
181 .....           <a class="games__card-link" target="_blank"
182 .....             href="https://store.steampowered.com/app/578080/
183 .....             PUBG_BATTLEGROUNDS/">
184 .....             
186 .....             <p class="games__card-desc lng-card-1-desc">
187 .....               Висаджуйтеся на стратегічно важливих місцях,
188 .....               добувайте зброю та припаси,
189 .....               й залишитися останньою
190 .....               командою на одному з численних полів бою.
191 .....             </p>
192 .....           </a>
193 .....         </div>
194 .....       </div>
195 .....     </div>
196 .....   </div>
197 ..... </section>

```

Рис. 3.14 Принцип розмітки карток блоку «Ігри»

- д) Задані стилі (рис. 3.15) для блоку «Ігри» та анімації наведення, при яких з'являється короткий опис обраної гри.

```

468 .games__container{
469     display: flex;
470     justify-content: space-evenly;
471     margin-bottom: 90px;
472 }
473
474 .games__cards{
475     display: grid;
476     grid-template-columns: repeat(3,2fr);
477     column-gap: 105px;
478     row-gap: 206px;
479 }
480
481 .games__card{
482     position: relative;
483     width: 452px;
484     height: 518px;
485     background: url(../img/line-games.png) no-repeat left ;
486 }
487 .games__card-link:visited,
488 .games__card-link:active,
489 .games__card-link:focus {
490     color: transparent;
491     outline: none;
492 }
493 .games__cards:hover > .games__card:not(:hover){
494     transform: scale(.9);
495     transition: 1s;
496 }
497 .games__cards:hover > .games__card:not(:hover) .games__card-item,
498 .games__cards:hover > .games__card:not(:hover) .games__card-title{
499     transform: scale(.9);
500     filter:brightness(0.5) saturate(0) contrast(1) blur(20px);
501     transition: 1s;
502 }

```

Рис. 3.15 Стилізація карток блоку «Ігри»

5) Таблиця блоку «Послуги»

1. Таблиця (рис. 3.17) блоку послуги написана за допомогою flex-елементів. Перемикання вкладок таблиці реалізовано на JavaScript. На рисунках нижче наведено приклад розмітки лише вкладки над таблицею, а також розмітку першої таблиці (рис. 3.16). Решта таблиць описано у такий самим спосіб.

```
<section class="services" id="services">
  <div class="wrapper">
    <h3 class="services_title lng-services">
      Послуги
    </h3>
    <ul class="services_list">
      <li class="services_list-item page1 lng-services-cyber">кібер зона</li>
      <li class="services_list-item page2 lng-services-ps4">зона PS4</li>
      <li class="services_list-item page3 lng-services-ps5">зона PS5</li>
      <li class="services_list-item page4 lng-services-lounge">зона відпочинку</li>
    </ul>
    <div class="services_table first">
      <div class="services_table-left">
        <div class="services_tablemaintitle-left">
          <h4 class="services_tabletitle-left lng-services-time-cyber">
            час
          </h4>
        </div>
        <ul class="services_tablelist-left">
          <li class="services_tablelist-item lng-services-time-1">30 хвилин</li>
          <li class="services_tablelist-item lng-services-time-2">1 година</li>
          <li class="services_tablelist-item lng-services-time-3">3 години</li>
          <li class="services_tablelist-item lng-services-time-4">5 годин</li>
          <li class="services_tablelist-item lng-services-time-5">День (8:00-22:00)</li>
        </ul>
      </div>
      <div class="services_table-right">
        <div class="services_tablemaintitle-right">
          <h4 class="services_tabletitle-right">
            про
          </h4>
        </div>
        <div class="services_tabletitle-right">
          <h4 class="services_tabletitle-right">
            vip
          </h4>
        </div>
        <div class="services_tablelist-right">
          <ul class="list-left">
            <li class="services_tablelist-item lng-services-price-1">30 грн</li>
            <li class="services_tablelist-item lng-services-price-2">55 грн</li>
            <li class="services_tablelist-item lng-services-price-3">140 грн </li>
            <li class="services_tablelist-item lng-services-price-4">200 грн</li>
            <li class="services_tablelist-item lng-services-price-5">300 грн</li>
          </ul>
          <ul class="list-right">
            <li class="services_tablelist-item lng-services-price-6">100 грн</li>
            <li class="services_tablelist-item lng-services-price-7">180 грн</li>
            <li class="services_tablelist-item lng-services-price-8">270 грн </li>
            <li class="services_tablelist-item lng-services-price-9">250 грн</li>
            <li class="services_tablelist-item lng-services-price-10">300 грн</li>
          </ul>
        </div>
      </div>
    </div>
  </div>
</section>
```

Рис. 3.16 Розмітка таблиці

2. Розміщення елементів таблиці можливе за рахунок flex-елементів. Усього є 3 списки, вони поділені на 2 групи: список зліва (".services__table-left) і списки зправа (".services__tablelist-right), де кожен список також поділено на два з класами ".list-left" і ".list-right" відповідно. Ці три блоки обгорнені в контейнер ".services__table", який розділяє дочірні списки на дві частини. Ця структура утворює імпровізовану таблицю.

ПОСЛУГИ

КІБЕР ЗОНА
ЗОНА PS4
ЗОНА PS5
ЗОНА ВІДПОЧИНКУ

ЧАС	PRO	VIP
30 Хвилин	30 Грн	100 Грн
1 Година	55 Грн	180 Грн
3 Години	140 Грн	270 Грн
5 Годин	200 Грн	400 Грн
День (8:00-22:00)	300 Грн	600 Uah

Рис. 3.17 Результат розмітки і стилізації таблиці

3. Таблиця реалізована на JavaScript. Цей код (рис. 3.18) є досить легким, але заплутаний з точки зору алгоритму:

```

88 // servicesTable-change
89 (function () {
90     const firstTable = document.querySelector('.first');
91     const secondTable = document.querySelector('.second');
92     const thirdTable = document.querySelector('.third');
93     const fourthTable = document.querySelector('.fourth');
94     const firstPage = document.querySelector('.page1');
95     const secondPage = document.querySelector('.page2');
96     const thirdPage = document.querySelector('.page3');
97     const fourthPage = document.querySelector('.page4');
98     ...
99
100     firstPage.addEventListener('click', () => {
101         firstPage.classList.add('page1-background');
102         secondPage.classList.remove('page1-background');
103         thirdPage.classList.remove('page1-background');
104         fourthPage.classList.remove('page1-background');
105
106         firstTable.classList.remove('first-page');
107         firstTable.classList.add('first');
108         secondTable.classList.add('second');
109         thirdTable.classList.add('third');
110         fourthTable.classList.add('fourth');
111     });
112
113     secondPage.addEventListener('click', () => {
114         firstPage.classList.remove('page1');
115         firstPage.classList.remove('page1-background');
116         thirdPage.classList.remove('page1-background');
117         fourthPage.classList.remove('page1-background');
118         secondPage.classList.add('page1-background');
119
120         secondTable.classList.remove('second');
121         firstTable.classList.add('first-page');
122         thirdTable.classList.add('third');
123         fourthTable.classList.add('fourth');
124     });
125
126     thirdPage.addEventListener('click', () => {
127         firstPage.classList.remove('page1');
128         firstPage.classList.remove('page1-background');
129         secondPage.classList.remove('page1-background');
130         thirdPage.classList.add('page1-background');
131         fourthPage.classList.remove('page1-background');
132
133         thirdTable.classList.remove('third');
134         firstTable.classList.add('first-page');
135         secondTable.classList.add('second');
136         fourthTable.classList.add('fourth');
137     });
138
139     fourthPage.addEventListener('click', () => {
140         firstPage.classList.remove('page1');
141         firstPage.classList.remove('page1-background');
142         secondPage.classList.remove('page1-background');
143         thirdPage.classList.remove('page1-background');
144         fourthPage.classList.add('page1-background');
145
146         firstTable.classList.add('first-page');
147         secondTable.classList.add('second');
148         thirdTable.classList.add('third');
149         fourthTable.classList.remove('fourth');
150     });
151
152 })()
153
154

```

Рис. 3.18 Програмна реалізація перемикачів

У CSS створено 7 класів з властивостями `display` та значеннями `none`, один пустий з заглушкою у вигляді `content:" "`, а також два класа для задання фону вкладкам (клас `“.page1-background”` встановлює значення за умовчужанням для виділення першої вкладки при завантаженні сторінки).

Функція передбачає встановлення слухача подій, який перевіряє наявність `display:none` й `background` та видаляє клас з цією властивістю під час вибору потрібної вкладки.

3.2 Перевірка Web-сайту комп'ютерного клубу

Важливою частиною завершення розробки сайту є валідація, компресія, та оптимізація коду. Після завершення написання розмітки, стилів, а також алгоритмів код пройшов повторну валідацію за межами VSC, адже середовище розробки може пропускати деякі помилки посилаючись на довершеність браузера, який пробачить невеликі помилки під час написання коду.

1. Щоб код був ідеально написаний без помилок, використано сторонні валідатори (рис. 3.19), такі як: **W3.org HTML/CSS Validator** та **Esprima Syntax Validator** для JS.

The image shows a screenshot of the W3.org HTML/CSS Validator interface. On the left, there is a code editor with JavaScript code. On the right, the 'Showing results for contents of text-input area' section displays the HTML document structure. Below the code editor, a green banner states 'Code is syntactically valid.' Below the HTML preview, another green banner states 'Document checking completed. No errors or warnings to show.' At the bottom, there is a blue banner with the W3C logo and text: 'сервис проверки CSS от W3C', 'W3C результаты проверки CSS для TextArea (CSS3 + SVG)', and 'Перейти к: Преду'. Below this, there is a green banner with the text: 'W3C результаты проверки CSS для TextArea (CSS3 + SVG)', 'Поздравляем! Ошибок не обнаружено.', and 'Этот документ проходит проверку по стандарту CSS3 + SVG!'.

Рис. 3.19 Результати валідації HTML, CSS, JS

2. Перед валідацією CSS файлу, розставлено вендорні префікси (рис. 3.20), для кращої кросбраузерності.



Рис. 3.20 Використання CSS Autoprefixer

3. Після успішної валідації усі файли, що знаходяться в папці проекту пройшли компресію (рис. 3.21, рис. 3.22), тобто були перетворені в текст, котрий займає одну строку (видалено всі відступи з файлів)



Рис. 3.21 Результати компресії HTML

```
1 jQuery(document).ready(function(){$("a[rel='m_PageScroll2id']").mPageScroll2id({offset:120});let select=function(){let e=document.querySelectorAll(".select__header"),s=document.querySelectorAll(".select__item"),t=document.querySelector(".select__current");function a(){this.parentElement.classList.toggle("is-active")}function c(){let e=this.innerText,s=this.closest(".select");s.querySelector(".select__current").innerText=e,s.classList.remove("is-active");let t=this.getAttribute("id");window.location.hash=t,window.location.reload(),localStorage.setItem("selectedItemId",t)}e.forEach(e=>{e.addEventListener("click",a)}),s.forEach(e=>{e.addEventListener("click",c)}),window.addEventListener("DOMContentLoaded",function(){let e=window.location.hash.substring(1);if(e){let s=document.getElementById(e);if(s){let a=s.innerText;t.innerText=a,localStorage.setItem("selectedItemId",e)}else window.location.hash="ua",t.innerText="UA",localStorage.setItem("selectedItemId","ua"),window.location.reload()}for(let c in langArr){document.querySelector(".lng-"+c).innerHTML=langArr[c][e];let r=document.querySelector(".lng-"+c);r&&(r.innerHTML=langArr[c][e]),void 0===c&&(r.innerHTML=langArr[c][e],window.location.reload())}}});select(),function(){let e=document.querySelector(".header");window.onscroll=(()=>{window.pageYOffset>50?e.classList.add("header_active"):e.classList.remove("header_active")}()),function(){let e=document.querySelector(".first"),s=document.querySelector(".second"),t=document.querySelector(".third"),a=document.querySelector(".fourth"),c=document.querySelector(".page1"),r=document.querySelector(".page2"),l=document.querySelector(".page3"),i=document.querySelector(".page4");c.addEventListener("click",()=>{c.classList.add("page1-background"),r.classList.remove("page1-background"),l.classList
```

Рис. 3.22 Результати компресії JS

Над CSS файлом проведено таку саму процедуру.

4. Останній крок це завантаження сайту у мережу. Один з таких сайтів, котрий надає безкоштовний домен для web-сайту це **Netlify**.

- Папку з проектом завантажено на Netlify.
- Створено унікальне посилання на web-сторінку: <https://altrone.netlify.app/>

3.3 Рекомендації з упровадження Web-сайту комп'ютерного клубу

1. Визначення ідеї

Для початку розробки сайту нам важливо розуміти, чим саме займається компанія-замовник, яка специфіка його продукту або послуги. Цю інформацію потрібно конкретизувати якомога детальніше, з усіма нюансами.

Можливо, якась особливість, яка здається незначною, зіграє ключову роль в позиціонуванні сайту. Крім того, важливо сформулювати ідею майбутнього сайту: яку думку, почуття, меседж він повинен донести.

2. Тип сайту

Потрібно визначитися, який сайт ви хочете отримати.

Тип сайту слід обрати виходячи з того, які функції він буде виконувати. Якщо його призначення - це просто вистава однієї послуги, курсу або конференції, буде достатньо лише однієї сторінки Лендінг. Якщо потрібен багатосторінковий сайт, де необхідно розписати всі послуги та інформацію про компанію - корпоративний.

Якщо хочете продавати широкий асортимент товарів з онлайн-оплатою, вам потрібен інтернет-магазин. Інформаційний портал необхідний для активного і регулярного донесення певної інформації (наприклад, новин).

3. Технічне завдання

При розробці сайту обов'язково складіть з розробниками технічне завдання з докладним зазначенням вимог до майбутнього сайту.

Саме на нього фахівці спиратимуться на всіх етапах розробки.

4. Цільова аудиторія

Ви повинні чітко визначити, хто та цільова аудиторія, яка стане користувачами сайту, кого саме він зацікавить і чому. Від статі, віку, статків і соціального статусу аудиторії багато в чому буде залежати стиль сайту, його наповнення та структура.

Рішення, оптимальні для молоді, неприпустимі для людей старшого покоління, що добре для багатьох жінок, навряд чи сподобається чоловікам середнього достатку.

5. Функціонал

Продумайте, який функціонал повинен бути на сайті. Це може бути: особистий кабінет, онлайн-калькулятор, імпорту товарів через XML-файл і все те, з чим ви і відвідувачі будуть взаємодіяти.

Також обов'язково вкажіть мовні версії сайту. Наприклад, якщо ви плануєте працювати на українському ринку, вам неодмінно потрібна версія українською мовою, тому що цього тепер вимагає законодавство.

6. Структура

Підготуйте весь список послуг, що надаються або категорій товарів. Від цього буде залежати зручність навігації по сайту і структура меню.

Коли структура готова, її аналізує SEO-спеціаліст і юзабіліті-експерт. Вони оцінюють її з точки зору подальшого просування сайту та зручності, вносять свої корективи, якщо це необхідно.

7. Контент

Подумайте про контент, який буде на сайті. Звідки ви будете брати тексти, фото і відео?

8. Референс

Підберіть кілька прикладів сайтів з вашої тематики, які подобаються. Дивіться не тільки локальні сайти конкурентів, а й колег з інших країн, особливо з Західної Європи і США.

9. Дизайн

Якщо у компанії вже є готовий логотип і фірмовий стиль, це стане дуже хорошою новиною для дизайнерів. Якщо немає, потрібно як мінімум визначитися із загальною стилістикою і квітами, які будуть використовуватися в оформленні. Сайту знадобиться загальна візуальна концепція, колірні рішення, шрифти та інші графічні елементи, які будуть виглядати естетично, добре поєднуватися із загальним дизайном і між собою.

10. Просування

Навіть якщо ваш сайт готовий і дуже хороший, людям потрібно якимось про нього дізнатися. Існує безліч методів просування сайту для залучення користувачів.

Для просування зазвичай використовують такі канали:

- SEO-просування
- Контекстна реклама
- SMM-просування
- Контент-маркетинг
- Розміщення на сторонніх майданчиках і агрегаторах

11. Домен і хостинг

Заздалегідь потрібно визначитися з доменом сайту, особливо, якщо це стосується домену рівня .UA. Отримати такий домен можливо тільки після реєстрації торгової марки, а це може зажадати значного часу.

Визначитися також, на якому хостингу буде розміщуватися сайт. Для невеликого корпоративного сайту вистачить звичайного хостингу, ціна яких стартує від 5\$ в місяць.

Інтернет-магазин з десятками тисяч товарів і планованої великою відвідуваністю вимагає виділеного сервера і додаткових витрат на настройку.

Висновок до розділу 3

У середовищі розробки VSC було реалізовано наступну динаміку сайту:

- навігація у шапці та футері web-сторінки
- функція плавного скролу сторінки
- збереження положення header при скролі сторінки
- кнопка і функція зміни локалізації сайту на англійську мову
- таблиця та функція перемикавання вкладок таблиці

Функції написані на JavaScript, а алгоритми їх роботи описано у розвернутій формі, також використана бібліотека jQuery для реалізації функції плавного скролу від навігації до відповідного блоку.

Файли проекту успішно пройшли валідацію на помилки, а також компресію тексту коду, що зменшило розмір файлів, котрі завантажено на сервіс розміщення сайту в мережу.

Створено унікальний домен та посилання на сайт, щоб користувачі змогли знайти його через строку пошуку в браузері.

Для розробки якісного web-ресурсу потрібно розуміти особливості створення сайту, які включають:

- розробку ідеї
- визначення типу сайту
- написання ТЗ
- визначення функціоналу
- структура сайту

А також мати загальне уявлення про просування сайту в мережі. Усі ці особливості описані в пункті 3 розділу 3.

ВИСНОВКИ

У дипломній роботі проведено теоретичне дослідження про якісне створення сайту, написання коду та використання різних бібліотек та фреймворків, після якого визначено тип сайту, методи і методології розробки.

Виконано розробку web-сайту комп'ютерного ігрового клубу. Метою проекту було створення функціонального та привабливого веб-інтерфейсу для користувачів, який надає чітке представлення про заклад та всю необхідну інформацію для потенційного відвідувача. Користувач може зайти на web-сторінку клубу щоб отримати коротку, але зрозумілу інформацію про клуб, встановлені на комп'ютерах ігри, послуги та останній актуальні новини чи турніри.

Для досягнення цієї мети було використано сучасні технології та інструменти розробки веб-сайтів, зокрема HTML, CSS, JavaScript та бібліотека jQuery.

Результатом роботи є функціональний web-сайт комп'ютерного ігрового клубу зі зручним інтерфейсом. На сайті реалізовано можливість зміну мови, для зручності користування сайтом іноземцями.

Розроблений web-сайт має потенціал для подальшого розширення та вдосконалення, зокрема шляхом додавання нових функцій та можливостей для взаємодії між користувачами. Крім того, його можна пристосувати для використання в інших сферах, які вимагають організації турнірів та змагань.

У процесі розробки виникли певні труднощі, зокрема пов'язані з локалізацією сайту на англійську мову та оптимізацією продуктивності. Проте, завдяки систематичному підходу до роботи та використанню сучасних технологій, вдалося успішно розробити функціональний та ефективний веб-сайт.

Отже, дипломна робота доводить, що розробка web-сайту комп'ютерного ігрового клубу є актуальною та перспективною задачею. Результати роботи можуть бути використані в реальних умовах для створення привабливих та зручних веб-інтерфейсів для користувачів, що зацікавлені у геймінгових турнірах та змаганнях.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. AG Marketing: Навіщо бізнесу потрібен сайт [Електронний ресурс]. – Режим доступу: <https://ag.marketing/blog/navishcho-biznesu-potriben-sayt/>
2. Web Tec: Веб-сайт: визначення й застосування [Електронний ресурс]. – Режим доступу: <http://www.webtec.com.ua/uk/articles/index/view/2011-05-05/web-site>
3. NDU: Поняття, структура та різновиди веб-сайтів. Автоматизоване розроблення веб-сайтів [Електронний ресурс]. – Режим доступу: <http://www.ndu.edu.ua/liceum/web.pdf>
4. ApriCode: Види сайтів [Електронний ресурс]. – Режим доступу: <https://ApriCode.Com/Uk/Razrobotka/Vidy-Sajtov/>
5. VPU20: Структура веб-сайтів [Електронний ресурс]. – Режим доступу: <http://www.vpu20.lviv.ua/images/library/informatyka/okg02.pdf>
6. Pidru4niki: Системи і форми електронної торгівлі [Електронний ресурс]. – Режим доступу: https://pidru4niki.com/13710106/marketing/sistemi_formi_elektronnoyi_torgivli
7. Wezom: Методології розробки [Електронний ресурс]. Режим доступу: <https://wezom.com.ua/ua/blog/metodologija-razrobotki-programmnogo-obespechenija>
8. Scrum: Introduction to Scrum [Електронний ресурс]. – Режим доступу: <https://www.scrum.org/>
9. Український Спектр: Методологія Kanban [Електронний ресурс]. – Режим доступу: <https://uaspectr.com/2021/01/26/shho-take-kanban/>
10. Luxnet: Популярні бібліотеки [Електронний ресурс]. – Режим доступу: <https://luxnet.io/uk/blog/popular-frameworks-for-front-end-development-2020>
11. UK Legacy: Інструкція по React.js [Електронний ресурс]. – Режим доступу: <https://uk.legacy.reactjs.org/tutorial/tutorial.html>
12. Angular: Документація [Електронний ресурс]. – Режим доступу: <https://angular.io/docs>

13. Vue.js: Документація [Електронний ресурс]. – Режим доступу: <https://ua.vuejs.org/guide/introduction.html>
14. JQuery: Documentation [Електронний ресурс]. – Режим доступу: <https://api.jquery.com/>
15. Moment.js: Documantation [Електронний ресурс]. – Режим доступу: <https://momentjs.com/docs/>
16. Linkomania: CSS методології. CSS БЕМ, SMACSS [Електронний ресурс]. – Режим доступу: <http://linkomania.com.ua/css-metodologiyi-css-bem-smacss-ecss/>
17. Основи об'єктно-орієнтованого програмування: навчальний посібник / О. В. Щербаков, Ю. Е. Парфьонов, В. М. Федорченко. – Харків : ХНЕУ ім. С. Кузнеця, 2019. – 237 с.
18. Сучасні інформаційно-комунікаційні технології: Навчальний посібник. / Г.Г.Швачич, В.В.Толстой, Л.М.Петречук, Ю.С.Івашенко, О.А.Гуляєва, Соболенко О.В. – Дніпро: НМетАУ, 2017. –230 с.
19. Аналіз методів тестування зручності використання веб-сайтів / С. С. Кухарчук О.В. Романюк – Вінниця: ВНТУ.