

ДИПЛОМНА РОБОТА

ДР.Шс – 23.00.000 ПЗ

Група Шс-2015

Тисяк С.О.

2019

Кафедра Інформаційних технологій та програмної інженерії

УДК 004

ДИПЛОМНА РОБОТА

Тема Розроблення веб-сервісу для організації спортивних турнірів за олімпійською системою

Напрямок підготовки 6.050103 – «Програмна інженерія»
(код і назва спеціальності)

ПОЯСНЮВАЛЬНА ЗАПИСКА

ДР.ПІс – 23.00.000 ПЗ
(позначення)

Студент

Тисяк С.О.

(підпис) (дата) (розшифрування підпису)

Керівник проекту

к.т.н.

Андрейко В.М.

(посада) (підпис) (дата) (розшифрування підпису)

Нормоконтроль

к.т.н.

Мануляк І.З.

(посада) (підпис) (дата) (розшифрування підпису)

Допускається до захисту

Завідувач кафедри

д.т.н., доц.

Мельничук С.І.

(посада) (підпис) (дата) (розшифрування підпису)

ПВНЗ УНІВЕРСИТЕТ КОРОЛЯ ДАНИЛА

Факультет Інформаційних технологій

Кафедра Інформаційних технологій та програмної інженерії

Напрямок підготовки 6.050103 – «Програмна інженерія»

ЗАТВЕРДЖУЮ:

Завідувач кафедри ІТПІ

_____ С.І. Мельничук

“ _____ ” _____ 2019 р.

**ЗАВДАННЯ
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ)**

Студенту Тисяку Сергію Олеговичу

1. Тема проекту (роботи) Розробка веб-сервісу для організації спортивних турнірів за олімпійською системою

Затверджена наказом ректора Університету Короля Данила від 15.11.2018 р. № 20/4

2. Термін задачі студентом закінченого проекту (роботи) 10.06.2019 р.

3. Вихідні дані до проекту (роботи) JavaScript, jQuery бібліотека, lightgallery бібліотека, CMS WordPress.

4. Зміст пояснювальної записки (перелік питань, що їх належить розробити)
1. Опис наявних аналогів, алгоритми роботи та опис використаних технологій існуючих веб-додатків. 2. Розробка алгоритмів та взаємозв'язків між функціональними об'єктами веб-додатку. 3. Розробка програмного забезпечення інтернет ресурсу.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
Постановка завдання, аналіз існуючих аналогів, інтерфейс, панель адміністратора, висновки

6. Консультанти з проекту (роботи), із зазначенням розділів проекту, що стосуються

Розділ	Консультант	Підпис, дата	
		Завдання Видав	Завдання прийняв

7. Дата видачі завдання 30.10.2018

Керівник Андрейко В.М.

Завдання прийняв до виконання Тисяк С.О.

КАЛЕНДАРНИЙ ПЛАН

Пор №	Назва етапів дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1.	Опис наявних сайтів аналогів, їх алгоритми роботи та опис використаних технологій	30.11.2018	
2.	Розробка алгоритмів та дизайну веб-сайту наукової організації	25.12.2018	
3.	Розробка допоміжних функцій та інтерфейсів для веб-сайту наукової організації	21.02.2019	
4.	Оформлення пояснювальної записки	31.05.2019	
5.	Оформлення графічного матеріалу та підготовка до захисту роботи	13.06.2019	

Студент-дипломник Тисяк С.О.
(підпис) (розшифровка підпису)

Керівник проекту Андрейко В.М.
(підпис) (розшифровка підпису)

АНОТАЦІЯ

Дипломна робота присвячена розробці фронт-енду та бек-енду для сайту з інформацією про спортивні події, які проводяться за олімпійською системою. В даній роботі реалізовано три сторінки, на яких знаходяться турнірна сітка, галерея та новини турніру, а також панель адміністратора для управління сайтом.

Оскільки спортивні змагання в університетах проводяться досить часто, існує потреба в створенні сайту, на якому можна було б переглядати інформацію про ці змагання. Тому головним завданням даної роботи є розроблення клієнтської та серверної частини для такого сайту.

SUMMARY

The thesis is devoted to the development of the front end and backend for a site with information about sporting events held under the Olympic system. In this work, there are three pages on which there is a tournament grid, gallery and news of the tournament, as well as an admin panel for managing the site.

Since sports competitions are held at universities often enough, there is a need to create a site where you can view information about these competitions. Therefore, the main task of this work is to develop a client and server part for such a site.

РЕФЕРАТ

Розрахунково-пояснювальна записка: 101 сторінка, 32 малюнки, список використаних джерел та 2 додатки.

Об'єктом дослідження є Університет Короля Данила.

Метою роботи є розробка програмного забезпечення для спортивних турнірів в Університеті Короля Данила та створення панелі адміністратора, для додавання і зміни даних.

В роботі використані технології для розробки клієнтської частини сайту, наприклад мова програмування JavaScript, таск-менеджер Gulp та CSS препроцесор SCSS та серверної частини сайту такі як, CMS WordPress, текстовий редактор для написання коду Siblime Text.

Методи розробки базуються на використанні веб-технологій.

Виконані необхідні техніко-економічні розрахунки.

ТУРНІРНА СІТКА, НОВИНИ, ПАНЕЛЬ АДМІНІСТРАТОРА,
WORDPRESS, JAVASCRIPT, SCSS.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ.....	9
ВСТУП	10
1 ОПИС АНАЛОГІВ, АЛГОРИТМИ РОБОТИ ТА ОПИС ВИКОРИСТАНИХ ТЕХНОЛОГІЙ ІСНУЮЧИХ ВЕБ-ДОДАТКІВ.....	13
1.1 Вимоги до розробки веб-додатку для організації спортивних турнірів за олімпійською системою	13
1.2 Аналіз наявних аналогів	17
1.3 Постановка задачі	23
2 РОЗРОБКА АЛГОРИТМІВ ТА ВЗАЄМОЗВ'ЯЗКІВ МІЖ ФУНКЦІОНАЛЬНИМИ ОБ'ЄКТАМИ ВЕБ-ДОДАТКУ	25
2.1 Обґрунтування технології та засобів реалізації веб-сайту	25
2.2 Загальна структура сайту	39
2.3 Спеціалізовані функції роботи з даними	41
3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗБЕЧЕННЯ ІНТЕРНЕТ РЕСУРСУ	48
3.1 Розробка front-end частини програмного забезпечення.....	48
3.2 Розробка back-end частини програмного забезпечення	53
ВИСНОВКИ.....	68
ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	69
Додаток А.....	71
Додаток Б	95

						ДР.ПІс – 23.00.000 ПЗ			
Зм.	Арк.	№ докум.				<i>Розроблення веб сервісу для організації спортивних турнірів за олімпійською системою. Пояснювальна записка</i>	Літ.	Ар.	Акрушів
Розроб.		<i>Тисяк С. О.</i>						9	101
Перевір.		<i>Андрейко В.М</i>							
Реценз.									
Н. Контр.		<i>Мануляк .І.З.</i>						<i>УКД, ПІс – 2015</i>	
Затверд.		<i>Мельничук С.І.</i>							

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ

HTML – HyperText Markup Language

CSS – Cascading Style Sheets

JS – JavaScript

CMS – Content Management System

OS Panel – Open Server Panel

					ДР.ІІс – 23.00.000 ПЗ	Арк.
						9
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підп.</i>	<i>Дата</i>		

ВСТУП

Актуальність теми. На даний момент Інтернет – це невід'ємна частина нашого життя. Це не просто розвага, а необхідність від якої вже нікуди не дітися.

Якщо «правильно» його використовувати, то він є надзвичайно корисним невичерпним джерелом інформації, допомагає отримати необхідні знання та навички, вирішити різноманітні завдання. Попри численні дискусії, використання Інтернету має більше переваг ніж недоліків.

Завдяки Інтернету люди по всьому світу мають можливість навчатися дистанційно, адже мають в доступі незліченну кількість книжок, курсів і статей. Зараз можна отримати розгорнуту відповідь на будь-яке запитання, яке стосується всіх можливих сфер і галузей, від науки і медицини до туризму і торгівлі, достатньо ввести короткий пошуковий запит. Також з появою інтернету з'явилася можливість будувати кар'єру і працювати дистанційно.

Уявити собі світ без інтернету на сьогоднішній день практично неможливо. Він спрощує діяльність людини при цьому мінімізуючи витрати часу на пошук потрібної інформації. В даний час розвиток інформаційних технологій є не тільки найважливішою областю знань, але є механізмом функціонування та розвитку соціальних відносин. На сьогоднішній день практично кожна організація має власний веб-сайт. Сайт є інструментом для ознайомлення з тою чи іншою організацією

Ще однією безумовною перевагою використання Інтернету є існування безлічі корисних, а також необхідних веб-ресурсів та додатків, які значно полегшують нам життя. Популярність веб- і мобільних додатків зростає з кожним роком. Важко уявити, як можна обходитися без вже звичних нам додатків і веб-ресурсів, якими ми користуємось кожного дня, щоб сплачувати рахунки, спілкуватися з колегами, рідними і друзями, переглядати новини, дізнаватися прогноз погоди і багато іншого.

					ДР.Шс – 23.00.000 ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підп.	Дата		

Інтернет дає змогу не тільки навчатись, працювати та виконувати рутинну роботу в режимі онлайн, а також займатися своїм хобі. Наприклад, футбольний фанат, який не мав змоги подивитись матч улюбленої команди, може легко знайти цей матч в інтернеті навіть через багато днів після його закінчення, або просто подивитись результат цього матчу, прослідкувати за розвитком, подорожами чи трансферами команди. Послухати улюблені пісні чи подивитись цікаві фільми теж стало набагато простіше. Більше не потрібно йти в магазин і шукати диск чи касету із потрібними піснями чи фільмами. Зараз достатньо просто ввести у пошуку потрібний текст, а всю іншу роботу за тебе виконає інтернет.

Зараз існує дуже багато корисних сервісів на, практично, всі випадки життя, проте більшість з них розраховані на велику кількість людей, а не на якусь конкретну особу чи компанію. Ось чому розвиток ІТ ніколи не зупиниться. Завжди потрібен буде новий сайт, чи нова програма для виконання потреб малої кількості конкретних осіб.

Велика кількість спортивних центрів мають власні веб-сайти, але там немає реалізованого функціоналу, який би давав можливість додавати власний турнір. Тому розробка веб-сервісу, який надаватиме можливість користувачу створювати власний турнір та відслідковувати його в режимі онлайн є актуальною. Такий програмний продукт пришвидшить та спростить процес проведення будь-яких локальних змагань, а також дозволить учасникам та їх близьким чи просто фанатам відслідковувати результати матчів в режимі онлайн. Сьогодні майже будь-яка організація здійснює рекламу послуг, яка часто є нав'язливою та інколи навіть заважає перегляду сайту. Розроблений продукт дозволить не відволікатись на перегляд реклами і зосередитись на потрібній інформації.

Об'єкт дослідження. Об'єктом дослідження виступає спортивний ресурс, а саме турнірна сітка для проведення змагань за олімпійською системою.

Задачі досліджень. Відповідно до теми в роботі поставлені такі задачі:

- інтуїтивно зрозумілий інтерфейс користувача;

					ДР.Шс – 23.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		11

- зручний інтерфейс панелі адміністратора;
- коректна робота і видавання чітких релевантних результатів;
- стабільність роботи, забезпечення умов для безперебійного функціонування;
- збереження даних.

Методи досліджень. При вирішенні поставленого завдання для клієнтської частини були використані HTML, Gulp, SCSS, JavaScript та його бібліотеки. Для розробки серверної частини сайту я використав CMS WordPress, який написаний на PHP. Для написання коду використовувався текстовий редактор Sublime Text.

Результати досліджень. Результатом дипломної роботи є спортивний ресурс для проведення змагань за олімпійською системою, для якого розроблена панель адміністратора.

					ДР.ІІс – 23.00.000 ПЗ	Арк.
						12
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підп.</i>	<i>Дата</i>		

1 ОПИС АНАЛОГІВ, АЛГОРИТМИ РОБОТИ ТА ОПИС ВИКОРИСТАНИХ ТЕХНОЛОГІЙ ІСНУЮЧИХ ВЕБ-ДОДАТКІВ

1.1 Вимоги до розробки веб-додатку для організації спортивних турнірів за олімпійською системою

Олімпійська система (або плей-оф) – система розіграшу, за якою учасник вибуває з турніру після першого ж програшу. За цією системою проходять зокрема розіграші національних кубків з футболу, а також стадія плей-оф, до якої у свою чергу потрапляють найкращі команди групового турніру (Чемпіонат світу з футболу, Чемпіонат світу з баскетболу) або регулярного чемпіонату (НБА, НХЛ) [1].

Головною особливістю олімпійської системи є чітко встановлена кількість учасників (оскільки після кожного матчу виходить один переможець, то загальна кількість учасників має бути степенем числа 2 (4, 8, 16, 32, 64 і т.д.). Якщо ж кількість заявлених учасників не відповідає цій вимозі, то доводиться надавати деяким учасникам (найбільш високо рейтинговим або ж за простим жеребом) місце у другому раунді автоматично.

Також за цією системою проводяться додаткові матчі деяких національних чемпіонатів з ігрових видів спорту, зокрема футболу. Зазвичай ідеться про перехідні матчі за місце у вищому дивізіоні.

Якщо говорити про переваги над круговою та швейцарською системами, то до них можна віднести:

- менша кількість ігор, що дозволяє зекономити час і кошти, а іноді також вимагає меншої кількості арен для організації турніру;
- безкомпромісність боротьби, що унеможливорює договірні матчі;

А до недоліків можна віднести:

- Неможливість об'єктивно розподілити учасників за місцями, з огляду на те що один з них може мати сильніших суперників, ніж інший;

					ДР.Шс – 23.00.000 ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підп.	Дата		

- Відсутність оптимального варіанту проведення змагань. Доводиться або групувати учасників за рейтингом і зводити на ранніх стадіях учасника з високим рейтингом проти учасника з низьким рейтингом, або ж відмовлятися від «розсіювання» і проводити «сліпий жереб». У першому випадку втрачається інтерес до ранніх стадій розіграшу, через передбачуваність підсумків матчів, у другому – є велика імовірність того, що потенційно сильний учасник, зустрівшись уже на ранній стадії з рівним собі, вибуде з розіграшу, а потенційно слабший не зустрінеться із жодним грандом і дійде до фінальних раундів або й виграє турнір; може також вийти фінал, у якому один із суперників значно поступається силою іншому – що суттєво знизить інтерес до головного матчу турніру;
- Також нерівність умов спостерігається тоді, коли через неоптимальну для олімпійської системи кількість учасників декотрі з них змушені грати більше матчів, ніж їхні конкуренти, на самому старті розіграшу.

Використовується олімпійська система в основному у футболі.

Створення дійсно корисного і якісного продукту, сайту чи додатку, вимагає ретельного планування, чітких специфікацій і якісно виконаної роботи. Перш за все додаток має бути зручним у користуванні, швидким, і повинен працювати коректно.[2]

Під час створення сайту я керувався міжнародним стандартом ISO 9001.

ISO 9001 2015 - міжнародний стандарт, який визначає вимоги до систем менеджменту якості. Він - універсальний, оскільки застосовний до компаній будь-якого типу незалежно від сегмента ринку, який вони займають, і роду діяльності.[3]

Стандарт ISO 9001 включає вісім принципів, спрямованих на управління якістю:

- орієнтація на клієнта;
- залучення персоналу;
- лідерство;
- процесний підхід;

					ДР.Шс – 23.00.000 ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підп.	Дата		

- постійне поліпшення;
- системний підхід до управління;
- прийняття рішень на основі фактів;
- взаємовигідні відносини з постачальниками.

Розробка і впровадження системи управління якістю ISO 9001 дає компанії ряд переваг, серед яких:

- доказ прагнення компанії до вдосконалення і задоволення потреб клієнта;
- забезпечення відповідності продукції та послуг правовим і нормативним вимогам
- можливість виміряти прогрес в управлінні бізнес-процесами і поліпшити слабкі сторони менеджменту;

Під час сертифікаційного аудиту за міжнародним стандартом ISO 9001 орган по сертифікації перевіряє відповідність системи менеджменту компанії вимогам стандарту і оцінює ефективність її впровадження

Якщо сертифікація за стандартом ISO 9001 пройшла успішно, ви отримаєте сертифікат системи менеджменту якості ISO 9001, який є дійсним протягом 3-х років.

А вже під час розроблення даного документу я користувався методичними вказівками, розробленими викладачами університету Короля Данила. Серед цих вказівок були:

- сфера застосування;
- нормативні посилання;
- загальні положення;
- організація дипломного проектування;
- підсумкова державна атестація;
- вимоги до змісту та структури дипломної роботи;
- вимоги до оформлення дипломної роботи;
- додатки.

Веб-додаток має клієнт-серверну структуру, клієнтська частина якого виконується в браузері, а за допомогою запитів спілкується з сервером. Таким

					ДР.Шс – 23.00.000 ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підп.	Дата		

чином, при написанні веб-додатку необхідно реалізовувати як клієнтську частину (браузерну), яка подається у вигляді HTML розмітки, CSS стилів, JavaScript коду, відповідального для з'єднання з сервером так і серверну частину, яка повинна обробляти запити клієнтів та відповідати за функціонал сайту.

Багато людей не розуміє поділу на клієнтську та серверну частини (front-end та back-end) і думають, що весь функціонал пишеться разом з дизайном сайту чи програми, проте, це не так. В основному для написання front-end та back-end коду використовуються різні спеціалісти, які мають тільки поверхнєве уявлення про роботу один одного.[4]

Для написання якісного веб-додатку необхідно мати великий набір навичок. Для клієнтської частини це:

- знання мови розмітки сторінок HTML;
- вміння використовувати таблиці стилів або її препроцесорів;
- знання мови JavaScript для взаємодії з сервером і підвищення інтерактивності сторінок;
- використання таск-менеджерів – gulp, grunt чи webpack;
- кросбраузерна верстка – для того, що веб-сайт працював коректно в різних браузерах;
- адаптивний дизайн – для того, що веб-сайт працював коректно при різних розширеннях екрану;

Серверна частина є зовсім іншою стихією, вона повинна коректно і надійно опрацьовувати запити клієнта, зберігати сесії клієнтів, а також всю інформацію, що вводиться, бути стійкою до можливих атак з боку злоумисників.

В даний час найбільш популярними технологіями для реалізації серверної частини є Python, ASP.Net та PHP. Але, зараз, в зв'язку з розвитком JavaScript багато розробників відходять від стандартних методів розробки серверної частини, та обирають JavaScript як альтернативу.

Дуже великої популярності зараз набули різні CMS, такі як WordPress чи Django. З їхньою допомогою дуже просто зробити панель адміністратора для клієнта, для того, щоб коли йому потрібно зробити якісь зміни на сайті, йому не

					ДР.Шс – 23.00.000 ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підп.	Дата		

потрібно буде розбиратися з кодом і якось змінювати його, достатньо просто зайти в панель адміністратора і завдяки зручному та інтуїтивному інтерфейсу робити зміни вже там.

В даному проєкті я використав CMS під назвою WordPress. Вибрав я її через багато причин, наприклад WordPress:[5]

- швидкий, гнучкий;
- займає порівняно мало місця;
- має багато доробок та шаблонів;
- простий для зміни коду;
- має легке налаштування віджетів та меню;
- має вбудований редактор сторінок HTML;
- має автоматичне додавання відповідних пунктів меню після створення нових публікацій у відповідному розділі;
- має вбудовану можливість відносити публікацію до декількох категорій;
- має автоматичні оновлення усіх плагінів, тем та самої системи WordPress.

1.2 Аналіз наявних аналогів

Одним з аналогів сайту, який я розробляв є сайт myscore (рис. 1.1). На ньому можна подивитись інформацію про попередні і наступні матчі будь-якої професійної команди, чи будь-якого професійного турніру, причому не тільки футболу, але й волейболу, баскетболу, тенісу, хокею тощо.

Але, за винятком адміністрації сайту, ніхто не може вносити зміни на цьому сервісі, що робить неможливим його використання в локальних змаганнях. Це одна з причин чому було вирішено створити веб-сайт, на якому б користувачу показувалась інформація (результати, фотографії, дати проведення та різні новини) з будь-якого потрібного турніру, незалежно від того чи це змагання між групами університету, чи професійний турнір.

					ДР.Шс – 23.00.000 ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підп.	Дата		

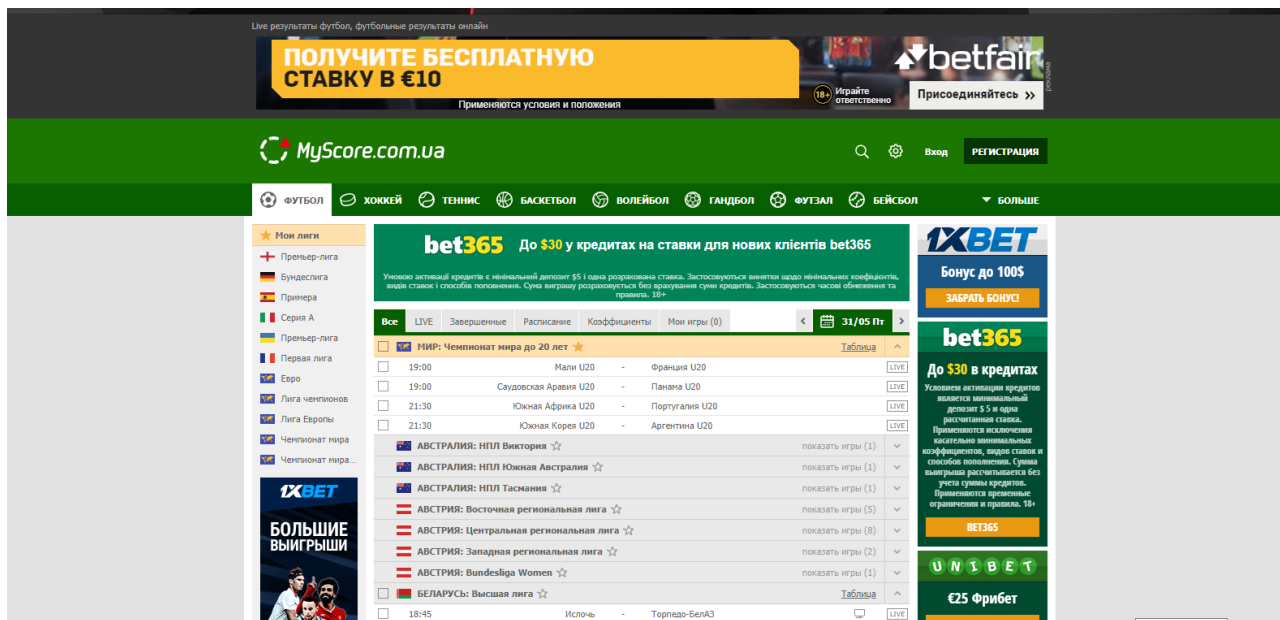


Рисунок 1.1 – Сайт myscore.com.ua

Враховуючи те, що розроблюваний веб-сайт повинен використовуватись для аматорських турнірів, не потрібно шукати надпотужні сервери для зберігання інформації про турнір чи про команду, чи навіть про конкретного спортсмена, як це зроблено на myscore. Адміністрація myscore повинна використовувати потужні сервери для зберігання всієї цієї інформації для того.

Загалом, це дуже популярний сайт, особливо в країнах СНГ, на якому можна дізнатись практично всю інформацію про свою улюблену команду чи турнір. Наприклад, будь-хто бажаючий може подивитись дату проведення наступного матчу практично будь-якої команди, інформацію про матч, коефіцієнти на перемогу тієї чи іншої команди, при чому з різних букмекерських контор, результати попередніх матчів, вік та національність гравців команди чи індивідуального спортсмена та інформацію про їхні трансфери (в основному ця інформація надається тільки про відносно популярних гравців) та багато чого іншого.

Якщо ж говорити про недоліки цього сервісу, то головними будуть:

- нав'язлива реклама (яка, в принципі, зараз є на більшості веб-сайтів, так як дозволяє власникам сайту заробляти гроші на своєму продукті, а так як

									Арк.
									18
Змн.	Арк.	№ докум.	Підп.	Дата	ДР.Шс – 23.00.000 ПЗ				

вона не перекриває самого вмісту веб-сайту, то й великим недоліком це назвати не можна);

- неможливість додавання свого власного змагання(як впринципі і на всіх подібних сайтах);
- неправильні назви та запізніла інформація про результат багато непопулярних в Європі турнірів (наприклад чемпіонат Палестини з футболу).

Ще одним аналогом мого дипломного проекту є сайт terricon.com (рис. 1.2).

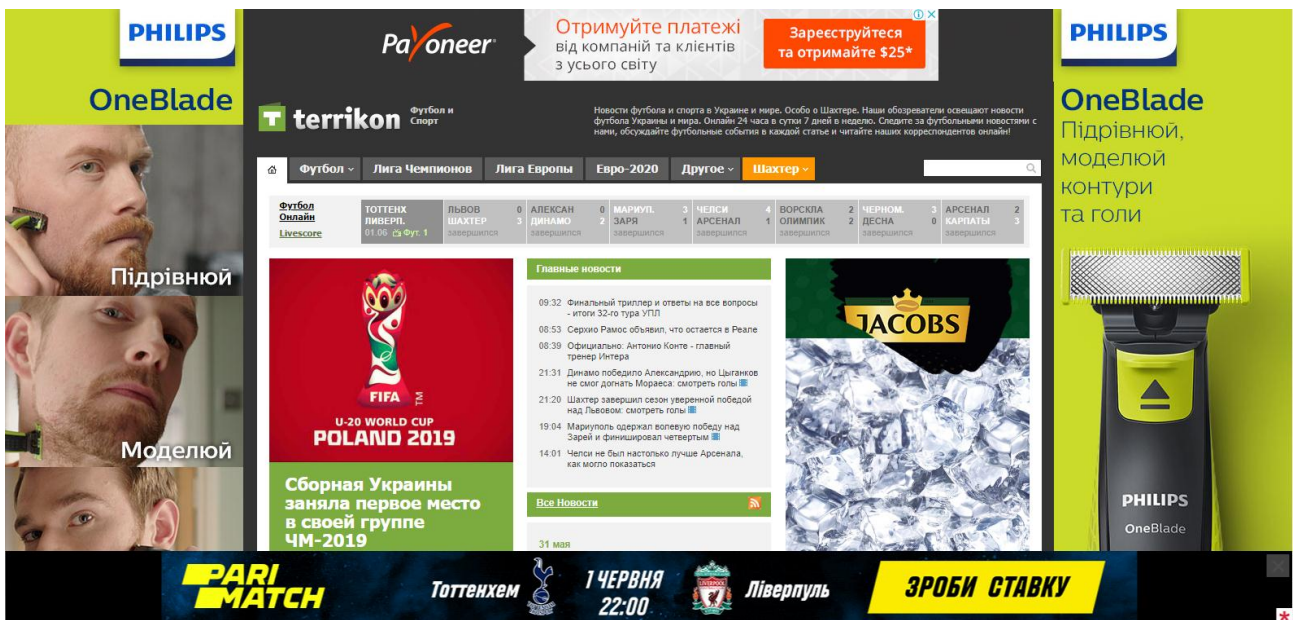


Рисунок 1.2 – Сайт terricon.com

На цьому веб-сайті також можна подивитись інформацію про різні турніри, проте, в основному, він розрахований для футболу. Також можна помітити, що вони виділяють команду «Шахтар Донецьк» на фоні інших команд, що на мою думку є неправильним, так як деякі користувачі можуть не користуватися цим сайтом тільки через те, що вони фанати інших команд, а це в свою чергу обмежує розвиток веб-сайту в подальшому.

					ДР.Шс – 23.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		19

Як і на myscore, на terricon є велика кількість турнірів, але їх набагато менше, також там немає інформації про конкретних гравців чи про конкретні команди, за винятком «Шахтаря».

Ще одним недоліком цього сайту є величезна кількість реклами. Фактично вона займає більше місця на сайті, ніж сам контент. І хоча, в основному, реклама й не перекриває контент, інколи вона «вискакує» знизу сторінки і потрібно її закрити, щоб продовжити перегляд, а це в свою чергу завдає незручності і відволікає від перегляду сайту. В основному цей сайт – це менш-функціональна версія myscore, тільки з менш-зручним інтерфейсом та з більшою кількістю реклами. Враховуючи всі його недоліки у порівнянні з перевагами, не дивно, що він не має й половини тої популярності, яка є в myscore.

І останній аналог розроблюваного мною сайту, це сайт skysports.com (рис.1.3).

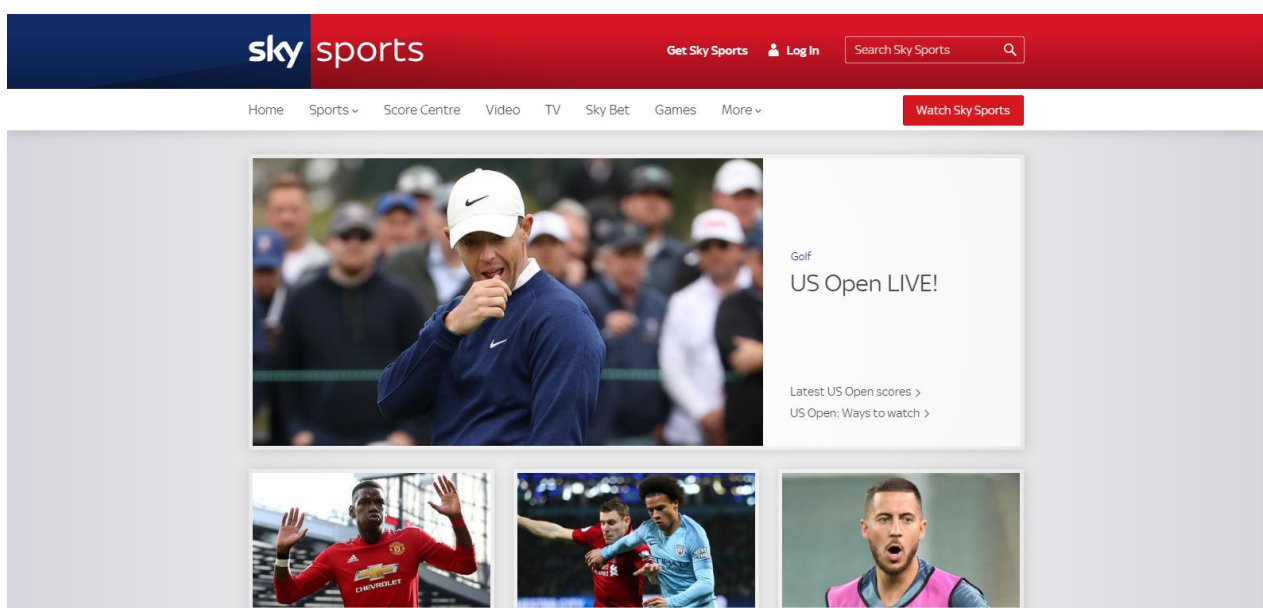


Рисунок 1.3 – Головна сторінка сайту skysports.com

Skysports – це англійський сервіс, який переріс зі звичайної газети в повноцінний веб-ресурс. На даний момент це найпопулярніший у світі сайт на спортивну тематику. На ньому можна подивитись інформацію про цілих двадцять видів змагань зі всього світу, при чому вся інформація буде показуватись без запізньєв в режимі онлайн.

					ДР.Шс – 23.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		20

Популярність цього сайту не викликає жодних запитань, так як на ньому є не тільки турнірна сітка та інформація про команди та окремих спортсменів. На skysports.com можна побачити безліч інформації та функцій, які недоступні на інших сайтах. Наприклад:

- почитати новини про можливі трансфери чи події безпосередньо в їхній онлайн газеті;
- перегляд матчів онлайн;
- ігри на спортивну тематику для android та iphon;
- інформація про матчі та турніри на цьому сайті, завдяки потужним серверам, зберігається набагато довше, ніж на інших;
- швидкість сайту просто неймовірна.

Також до переваг можна віднести чудовий дизайн цього сайту.

Якщо ж говорити про недоліки цього сайту, до до них можна віднести обмеження деяких функцій для певних країн чи регіонів, переглядати сайт можна виключно на англійській мові, а також рекламу. Хоча, враховуючи те, що реклами тут набагато менше, ніж на вище описаних сайтах, а також сайтах взагалі і вона взагалі не відволікає від перегляду сайту та не перекриває контент, то це й мінусом назвати не можна.

Так як створюваний мною сайт планувався для змагань з футболу(хоча він і може використовуватись і для інших змагань), то я вирішив описати й сторінку категорії футбол (рис. 1.4).

					ДР.Шс – 23.00.000 ПЗ	Арк.
						21
Змн.	Арк.	№ докум.	Підп.	Дата		

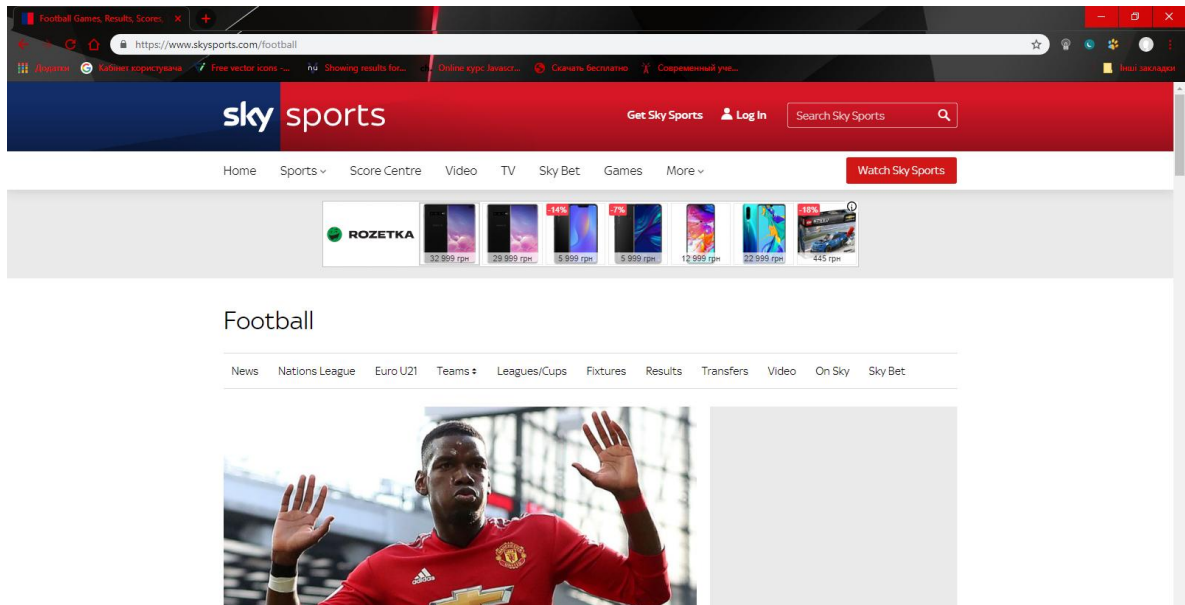


Рисунок 1.4 – Категорія футбол сайту skysports.com

На цій початку цієї сторінки знаходиться остання інформація, про найпопулярніші команди та про найпопулярніших футболістів світу. Крім того, на ній знаходиться меню для переходу до різних категорій:

- новини;
- Ліга Націй УЄФА;
- Чемпіонат Європи до 21 року;
- команди (для вибору і перегляду інформації про будь-яку професійну команду світу);
- ліги/кубки ;
- результати;
- трансфери;
- відео ;
- On Sky (для перегляду матчів онлайн);
- Sky Bet (на цій сторінці можна ставити ставки на спорт онлайн, але вона працює тільки для країн Євросоюзу, а також для США та Канади);

					ДР.Шс – 23.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		22

1.3 Постановка задачі

На даний момент різні школи, коледжі та університети проводять дуже багато різноманітних спортивних змагань. Вони проводять їх, як суто для одного закладу так і змагання між різними закладами. Але на даний момент не існує сервісу, який дозволяв би переглядати інформацію про ці змагання в режимі онлайн. Тому даний сайт повинен виправити цю проблему і дозволити переглядати інформацію про локальні змагання будь-якій людині.

Даний сайт пропонує створення свого власного змагання і розміщення його на сервері за допомогою панелі адміністратора. Під словом «створення» мається на увазі не тільки додавання інформації про назви команд і результати їхньої гри, але й додання турнірної сітки, що дасть змогу простіше відслідковувати результати, як всіх команд, так і якоїсь конкретної, також будь-хто мажачий зможе подивитись розгорнуту інформацію про проведений матч чи той, який ще тільки має проводитись (дата матчу, місце проведення турніру, імена учасників, мінімальна статистика, обзор на матч), крім цього користувачі матимуть змогу подивитись фотографії з будь-якого етапу турніру та з будь-якого матчу.

Головна мета цього сайту – це надання змоги створення та адміністрування будь-якого власного турніру.

Створення якісного сайту вимагає ретельного планування, чітких специфікацій і якісно виконаної роботи. Перш за все сайт має бути зручним у користуванні (простий, зрозумілий інтерфейс), швидким (оперативно обробляти вхідні і вихідні запити), і повинен працювати коректно (видавати чіткі правильні результати).

Основні функціональні можливості:

адміністративна панель, для додавання статей на сайт;

- можливість редагування та видалення уже існуючих статей;
- інтуїтивно зрозумілий інтерфейс користувача;
- зручний інтерфейс панелі адміністратора;

					ДР.Шс – 23.00.000 ПЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підп.	Дата		

- коректна робота і видавання чітких релевантних результатів;
- стабільність роботи, забезпечення умов для безперебійного функціонування;
- збереження даних.

Детальніший опис кожної функції і етапу розробки буде наведений в наступних розділах.

					ДР.ІІс – 23.00.000 ІЗ	Арк.
						24
Змн.	Арк.	№ докум.	Підп.	Дата		

2 РОЗРОБКА АЛГОРИТМІВ ТА ВЗАЄМОЗВ'ЯЗКІВ МІЖ ФУНКЦІОНАЛЬНИМИ ОБ'ЄКТАМИ ВЕБ-ДОДАТКУ

2.1 Обґрунтування технології та засобів реалізації веб-сайту

Одним з ключових моментів в розвитку всевітньої павутини відіграє веб - розробка - процес створення веб-сайта або веб-додатку. Термін включає розробку додатків електронної комерції, веб-дизайн, програмування для веб на стороні клієнта і серверу, а також конфігурування веб-серверу. Основними етапами веб-розробки є:

- проектування сайту або веб-додатку;
- створення макетів сторінок;
- наповнення;
- обслуговування працюючого сайту або його програмної основи;
- подальше просування сайту в мережі та підняття його рейтингу.

Вибір технологій є одним із найважливіших етапів розробки будь-якого сайту чи додатку. Тому варто приділити цьому достатню кількість часу, зробити дослідження, проаналізувати доступні варіанти і вибрати найкращі технології, які чудово справлятимуться із поставленими задачами.

Важливі критерії при виборі стеку технологій:

- розмір і тип проекту, тут варто враховувати призначення проекту, потенційну аудиторію тощо;
- складність проекту;
- підтримуваність вибраних технологій.

Прості проекти це зазвичай сайти візитки, лендінги, прості інтернет-магазини, прості програми. Для таких типів проектів зазвичай вже є готові рішення - CMS або шаблони. Середні проекти - складні інтернет-магазини і маркетплейси, портали національного масштабу, різноманітні сервіси. Зазвичай використовуються фреймворки. Складні - величезні портали, соціальні мережі,

					ДР.Шс – 23.00.000 ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підп.	Дата		

інноваційні та нетипові рішення. Ядро таких проектів зазвичай розробляється на чистій мові програмування.

Слід звертати увагу на те наскільки добре документований фреймворк, інструмент чи бібліотека. Адже обґрунтована, повна документація дозволяє легко знаходити методи розв'язування необхідних задач чи виправляти помилки, з якими доводиться зустрічатися в процесі розробки. В основному, якщо в програміста хороші навички англійської мови, для нього не складе великої проблеми знайти хорошу документацію для більшості фреймворків чи бібліотек, проте, якщо він хоче знайти документацію на якійсь іншій мові, особливо не дуже поширеній, то у нього можуть виникнути серйозні проблеми з вивченням і використанням цих технологій.

Крім того важливим є дата виходу і, особливо, частота використання певної мови чи технології. Тому, що при використанні популярної технології набагато легше знайти документацію по ній або навіть вже готові рішення. При використанні чогось нового або не популярного такої можливості немає.

Також потрібно враховувати складність і підтримуваність певного фреймворку чи бібліотеки. Особливо небезпечно в цьому плані використовувати нові бібліотеки, які, наприклад не мають достатньої підтримки більшості сучасних браузерів чи операційних систем.

Не найкращою ідеєю буде й використання застарілих технологій, так як в основному вони набагато повільніші за сучасні. Проте, перед тим, як використовувати нову технологію потрібно дізнатись її підтримуваність в браузерах та операційних системах.

Також слід оцінити рівень важливості і конфіденційності особистих даних і в залежності від цього використовувати потрібну технологію.

Основні технології які я використовував:

- HTML;
- JavaScript;
- JQuery;
- lightgallery.js;

					ДР.Шс – 23.00.000 ПЗ	Арк.
						26
Змн.	Арк.	№ докум.	Підп.	Дата		

- CSS;
- SCSS;
- Gulp;
- WordPress.

HTML – стандартизована мова розмітки документів у всесвітній павутині опублікована в 1993 році. Більшість сучасних веб-сайтів написані саме на цій мові. Крім HTML можна також використовувати і XML, проте HTML набагато зручніша і популярніша, крім того її підтримують усі сучасні браузері. З 1993 року HTML постійно покращувалась і випускала нові стандарти, останнім був HTML5.

По суті, HTML досить проста мова, що складається з елементів, які можуть бути застосовані до частин тексту, щоб надавати їм різні значення (абзац, список, таблиця тощо), розділяти документ на логічні секції і додавати контент на вашу сторінку, такий як текст, фото, відео і аудіо.

JavaScript класифікують як прототипну (підмножина об'єктно-орієнтованої), скриптову мову програмування з динамічною типізацією.

JavaScript спочатку створювався для того, щоб зробити web-сторінки «живими». Програми на цій мові називаються скриптами. У браузері вони підключаються безпосередньо до HTML і, як тільки завантажується сторінка - тут же виконуються.

Коли створювалася мова JavaScript, у неї спочатку була інша назва: «LiveScript». Але тоді була дуже популярна мова Java, і маркетингологи вирішили, що схожа назва зробить нову мову більш популярною.

Планувалося, що JavaScript буде таким собі «молодшим братом» Java. Однак, історія розпорядилася по-своєму, JavaScript сильно виріс, і зараз це абсолютно незалежна мова, зі своєю специфікацією, яка називається ECMAScript, і до Java не має ніякого відношення.

JavaScript одна з найпопулярніших мов програмування. Мова має безліч бібліотек і фреймворків серед яких ReactJS, VueJS, Angular, проте в своєму

					ДР.Шс – 23.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		27

проекті я використовував нативний JS. JavaScript Постійно розвивається і пропонує нові стандарти (останній ECMAScript 2016) та бібліотеки.

jQuery – JavaScript бібліотека. На даний час вона вже не така популярна, як колись, проте продовжує використовуватися на великій кількості веб-сайтів, в основному через те, що завдяки їй можна використовувати інші бібліотеки, наприклад різні слайдери чи API для робот з Ajax. В даному проекті я її підключив тільки заради того, щоб можна було працювати з бібліотекою lightgallery.js.

Lightgallery.js – бібліотека для створення галереї, яка є досить простою у використанні та підключенні.

CSS) – формальна мова опису зовнішнього вигляду документа, написаного з використанням мови розмітки.

Переважно використовується як засіб опису, оформлення зовнішнього вигляду веб-сторінок, написаних за допомогою мов розмітки HTML і XHTML, але може також застосовуватися до будь-яких XML-документів, наприклад, до SVG або XUL.

SCSS – «діалект» CSS препроцесора SASS, який дозволяє писати CSS код набагато швидше та простіше. SCSS препроцесор – це надбудова над CSS, яка додає раніше недоступні можливості для CSS, за допомогою нових синтаксичних конструкцій, таких як вкладення, міксини, змінні та багато інших функцій.

Приклад коду на SCSS:

```
.header{
  display: flex;
  align-items: center;
  justify-content: center;
  background-color: #878721;
  height: 140px;
  text-transform: uppercase;
  letter-spacing: 5px;
  margin-bottom: 30px;
```

					ДР.Шс – 23.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		28

```

.nav{
    width: 60%;
    margin: 0 auto;
    display: flex;
    justify-content: space-between;
    list-style: none;
    li{
        .item{
            color: #ffe;
            transition: .3s;
            position: relative;
            text-transform: uppercase;
            text-decoration: none;
            font-size: 20px;
            transition: .5s;
            &:after{
                background-color: black;
                display: block;
                content: "";
                height: 1.5px;
                width: 0%;
                left: 50%;
                position: absolute;
                transition: width .5s;
                transform: translateX(-50%);
            }
            &:hover:after,&:focus:after {
                width: 100%;
            }
            &:hover{
                color: black;
            }
        }
    }
}

```

					ДР.Шс – 23.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		29

```
}  
}
```

Цей самий код на CSS:

```
.header {  
  display: -webkit-box;  
  display: -webkit-flex;  
  display: -ms-flexbox;  
  display: flex;  
  -webkit-box-align: center;  
  -webkit-align-items: center;  
    -ms-flex-align: center;  
      align-items: center;  
  -webkit-box-pack: center;  
  -webkit-justify-content: center;  
    -ms-flex-pack: center;  
      justify-content: center;  
  background-color: #878721;  
  height: 140px;  
  text-transform: uppercase;  
  letter-spacing: 5px;  
  margin-bottom: 30px; }  
.header .nav {  
  width: 60%;  
  margin: 0 auto;  
  display: -webkit-box;  
  display: -webkit-flex;  
  display: -ms-flexbox;  
  display: flex;  
  -webkit-box-pack: justify;  
  -webkit-justify-content: space-between;  
    -ms-flex-pack: justify;  
      justify-content: space-between;  
  list-style: none; }  
.header .nav li .item {  
  color: #ffe;
```

					ДР.Шс – 23.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		30

```

-webkit-transition: .3s;
-o-transition: .3s;
transition: .3s;
position: relative;
text-transform: uppercase;
text-decoration: none;
font-size: 20px;
-webkit-transition: .5s;
-o-transition: .5s;
transition: .5s; }
.header .nav li .item:after {
  background-color: black;
  display: block;
  content: "";
  height: 1.5px;
  width: 0%;
  left: 50%;
  position: absolute;
  -webkit-transition: width .5s;
  -o-transition: width .5s;
  transition: width .5s;
  -webkit-transform: translateX(-50%);
  -ms-transform: translateX(-50%);
  transform: translateX(-50%); }
.header .nav li .item:after, .header .nav li
.item:focus:after {
  width: 100%; }
.header .nav li .item:hover {
  color: black; }

```

Як можна побачити код на SCSS набагато коротший та більш читабельний, ніж код на чистому CSS.

Gulp – таск-менеджер для виконання часто повторюваних задач, написаний на JS. Gulp дозволяє дуже сильно прискорити роботу за рахунок підключення різних плагінів, яких нараховується вже більше трьох тисяч. У своєму проєкті я

					ДР.Шс – 23.00.000 ПЗ	Арк.
						31
Змн.	Арк.	№ докум.	Підп.	Дата		

використовував плагін для автоматичного перезавантаження сторінки, для транслявання коду з SCSS в CSS, для мінімізації JS і CSS коду та зображень, для автоматичного добавлення вендорних префіксів. Для роботи з Gulp потрібен встановлений Node.js.

Для установки Gulp потрібно в командному рядку ввести команду:
npm і gulp -g – для глобальної установки Gulp на комп'ютер

Після чого потрібно створити папку і ввести команду npm init для ініціалізації проекту у створеній папці.

Далі потрібно створити файл json і ввести там всі необхідні дані, після чого встановити Gulp у свій проект використавши команду npm і gulp --save-dev.

Приклад роботи Gulp для автоматичного перезавантаження сторінки після змін у редакторі коду:

```
browserSync = require('browser-sync');
gulp.task('watch', ['browser-sync', 'css-libs', 'scripts'],
function() {
    gulp.watch('app/sass/**/*.scss', ['sass']);
    gulp.watch('app/*.html', browserSync.reload);
    gulp.watch('app/js/**/*.js', browserSync.reload);
});
```

Вибираючи технологію для розробки серверної частини даного сайту користувався такими критеріями:

- швидкість розробки, тобто терміни, які відведені на розробку проекту;
- вартість, слід аналізувати бюджет відведений під даний проект;
- доступні інструменти розробки;
- наявність готових рішень, якщо для поставленої задачі вже існує готове рішення, то постає питання чи варто взагалі створювати все з нуля, якщо можна використати вже розроблений алгоритм;
- вартість підтримки, якщо для розробки проекту використовуються платні сервіси, варто обрахувати вартість підтримки такого ресурсу в довготривалій перспективі;

					ДР.Шс – 23.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		32

- вимоги до навантажень, приблизна оцінка кількості відвідувань за день.
- кросплатформеність, потреба у використанні розробленого продукту різними типами операційних систем і тд;
- можливості інтеграції з іншими рішеннями, забезпечує можливість розвитку і вдосконалення розробленого проекту в майбутньому.

CMS – це система управління контентом – інформаційна система або комп'ютерна програма, яка використовується для забезпечення і організації спільного процесу створення, редагування і управління вмістом, інакше – контентом.

Найпопулярніші CMS:

- Django;
- Drupal;
- Etomite CMS;
- eZ publish;
- Irokez;
- Joomla;
- Mambo;
- mojoPortal;
- WordPress.

Універсального рішення у виборі звісно немає. Вибір залежить від цілей, технічних можливостей розробника, бюджету проекту та функціоналу. Для блогів та простих сайтів найчастіше використовують WordPress через простоту його використання для пересічного користувача.

Складні по функціоналу та із підвищеними потребами та вимогами до кастомізації, організації контенту найкраще всього використовувати Drupal.

У своєму проекті я використовував WordPress – це система керування даними з відкритим початковим кодом, яка написана на PHP у 2003 році і використовує сервер бази даних MySQL. На даний момент це найпопулярніша CMS у світі. За її допомогою я створив панель адміністратора для додавання даних на сайт.

					ДР.Шс – 23.00.000 ПЗ	Арк.
						33
Змн.	Арк.	№ докум.	Підп.	Дата		

Вибрав я її через багато причин, наприклад WordPress:

- швидкий, гнучкий;
- займає порівняно мало місця;
- має багато доробок та шаблонів;
- простий для зміни коду;
- має легке налаштування віджетів та меню;
- має вбудований редактор сторінок HTML;
- має автоматичне додавання відповідних пунктів меню після створення нових публікацій у відповідному розділі;
- має вбудовану можливість відносити публікацію до декількох категорій;
- має автоматичні оновлення усіх плагінів, тем та самої системи WordPress.

Так як WordPress написаний на PHP, то для того, щоб ваш сайт працював потрібно скопіювати дані з index.html в index.php. Після чого потрібно визначити, які дані чи структура мають залишатися незмінними, а які мають змінюватись чи вводитись заново через панель адміністратора. В тих місцях, де контент має додаватись через панель адміністратора потрібно прописати спеціальний PHP код. Наприклад, для за відображення заголовку певного поста потрібно ввести `<?php the_title() ?>`, а після того ввести цей заголовок у записі в панелі адміністратора.[7]

Для безпечної та швидшої побудови сайту потрібно дотримуватись вимог по охороні праці на робочому місці.

Основними принципами при організації робочого місця має бути врахування робочої пози при виконанні трудової діяльності.

Робоча поза — це основне положення працівника у просторі виробничого середовища.

Зручна робоча поза має забезпечувати стійке положення тіла людини, ніг, рук, голови під час роботи, мінімальну затрату енергії та максимальну результативність праці.[8]

Організація робочого місця передбачає:

					ДР.Шс – 23.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		34

- характер та особливості предметної діяльності;
- правильне розміщення у виробничому приміщенні;
- ергономічно обґрунтоване робоче положення, вибір параметрів робочого місця з урахуванням антропометричних характеристик людини.

- раціональну компоновку виробничих меблів, технологічного обладнання на робочих місцях.

Загальні принципи організації робочого місця:

- предмети і знаряддя праці мають бути на робочому місці, але не заважати працюючому;

- засоби і предмети праці, якими користуються постійно, мають бути ближче ніж ті, якими користуються рідше;

- предмети праці, які беруть лівою рукою, мають бути зліва, а ті, які беруть правою рукою - справа;

- якщо використовують обидві руки, то місце розташування предметів і засобів праці проектується з урахуванням зручності захоплення їх двома руками;

- організація робочого місця має забезпечувати хорошу візуальність, оглядовість, виключати захаращення його сировиною, зайвими предметами праці та готовими виробами.

Статичне напруження працівників у процесі праці пов'язане з підтримуванням у нерухомому стані предметів і знарядь праці, а також робочої пози.

При організації робочого місця, залежно від робочої пози, потрібно розподіляти статичні напруження, зменшувати їх величину та передбачувати можливість зміни пози під час предметної діяльності.

Визначений простір, обмежений по висоті 2 м над рівнем підлоги, на якому розташовано робочі місця постійного або непостійного (тимчасового) перебування працівників, називається робочою зоною.

Від правильного планування робочої зони у значній мірі залежать умови праці та її ефективність. Планування вважають основним елементом організації робочого місця. При плануванні робочого місця має передбачуватись таке його

					ДР.Шс – 23.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		35

положення у виробничому приміщенні по відношенню до інших робочих місць, яке буде забезпечувати найкращі умови з точки зору освітлення, вентиляції, опалення, подачі сировини і предметів праці, а також видалення готової продукції та виробничих відходів за межі робочої зони.

Правильне планування робочого місця призводить до значного зниження потреби у транспортних засобах, які у виробничих умовах завжди являють собою підвищену небезпеку для працюючих, зменшують вільну площу робочого місця, яка має становити не менше 4,5 м².

При раціональному плануванні робочих місць обов'язково враховують антропометричні дані людини. Від антропометричних даних залежить зона досяжності, тобто та частина робочої зони, яка обмежена дугами, що описуються максимально витягнутими руками. Зона допоміжних рухів (заміна інструментів, складання готових виробів) має бути розташованою за межами зони основних рухів, але у місяцях найкращої досяжності.

Робочі місяці мають бути оснащені засобами праці, що необхідні при виконанні тих або інших робочих функцій. Сюди належать засоби автоматизації, механізації, інструменти та різні пристосування, що забезпечують безпеку праці при виконанні предметної діяльності.

Компоновка технологічного обладнання та інших засобів праці у плані і просторі приміщення має:

- забезпечувати мінімальну кількість основних і допоміжних робочих рухів;
- виключати перехресні рухи рук або ніг;
- не допускати виходу з робочої зони для контролю робочих параметрів машини;
- забезпечувати надійність управління основним обладнанням. Вибір технічного обладнання, інвентарю і пристосування має забезпечувати фронт роботи з мінімальними витратами роботи та часу.

Додаткові пристосування та пристрої, мають забезпечувати безпеку праці, захист працюючих від дії шкідливих чинників з боку суміжних робочих місць (екрани, огороження, ширми, решітки і т. ін.). Положення технологічного

					ДР.Шс – 23.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		36

обладнання, пристосувань та інвентаря має бути стійким, щоб не створювати їх додаткового зрушення.

Продумане оснащення і розташування технологічного обладнання, інвентаря і пристосування дає можливість найбільше ефективно використовувати виробничі площі, уникати зайвої тісноти, створювати умови для безпечного виконання робіт.

Приступаючи до роботи з ПК, необхідно завжди пам'ятати, що це дуже складна і дорога апаратура, яка потребує акуратного й обережного ставлення до неї, високої самодисципліни на всіх етапах роботи з комп'ютером. Напруга живлення ПК (220 В) є небезпечною для життя людини. Через це в конструкції блоків комп'ютера, міжблочних з'єднувальних кабелів передбачена достатньо надійна ізоляція від струмопровідних ділянок. Користувач практично має справу лише з декількома вимикачами живлення і, здавалось би, застрахований від ураження електричним струмом. Однак в практичній роботі можуть зустрічатись непередбачені ситуації, і щоб вони не стали небезпечними для користувача, необхідно знати та чітко виконувати ряд правил техніки безпеки. Це допоможе не тільки уникнути нещасних випадків і зберегти здоров'я, але й гарантує збереження апаратури. Особливо уважним треба бути при роботі з дисплеєм, електронно-променевою трубкою якого використовує високу напругу і є джерелом електромагнітного випромінювання. Неправильне поводження з дисплеєм та іншою електронною апаратурою може призвести до тяжких уражень електричним струмом, спричинити загоряння апаратури.

Суворо забороняється:

- торкатися до екрана і тильного боку дисплея, проводів живлення і пристроїв заземлення, з'єднувальних кабелів;
- порушувати порядок ввімкнення й вимкнення апаратурних блоків, намагатись самостійно усунути виявлену несправність в роботі апаратури;
- класти на апаратуру сторонні предмети;
- працювати на комп'ютері у вологій одежі і з вологими руками.

					ДР.Шс – 23.00.000 ПЗ	Арк.
						37
Змн.	Арк.	№ докум.	Підп.	Дата		

В разі появи запаху горілого, незвичайних звуків або самовільного вимкнення апаратури треба негайно вимкнути комп'ютер і повідомити про це вчителя. Робота на комп'ютері потребує постійної уваги, чітких дій і самоконтролю. Через це на комп'ютері не можна працювати при недостатньому освітленні, високому рівні шуму.

Під час роботи на комп'ютері НЕОБХІДНО:

- суворо дотримуватись положень інструкції з експлуатації апаратури;
- уважно слідкувати за справністю основних блоків і пристроїв;
- працювати на клавіатурі чистими сухими руками, не натискувати на ті чи інші клавіші без потреби або навмання;
- працюючи з дискетами, оберігати їх від ударів, скручення, дії магнітного поля або тепла, не торкатись дискети, яка виступає з конверта, вставляти дискету в дисковод тільки після його ввімкнення, переконавшись в правильному орієнтуванні дискети відносно щілини дисковода;
- під час перерви в роботі вимикати комп'ютер лише в тому разі, коли обробка поточної інформації завершена і вміст оперативної пам'яті занесено на магнітні диски (в противному випадку неминуча втрата інформації).

Користувач практично має справу лише з декількома вимикачами живлення і, здавалось би, застрахований від ураження електричним струмом. Однак в практичній роботі можуть зустрічатись непередбачені ситуації, і щоб вони не стали небезпечними для користувача, необхідно знати та чітко виконувати ряд правил техніки безпеки. Це допоможе не тільки уникнути нещасних випадків і зберегти здоров'я, але й гарантує збереження апаратури.

В разі появи запаху горілого, незвичайних звуків або самовільного вимкнення апаратури треба негайно вимкнути комп'ютер і повідомити про це керівника або відповідального по техніці безпеки. Робота на комп'ютері потребує постійної уваги, чітких дій і самоконтролю. Через це на комп'ютері не можна працювати при недостатньому освітленні, високому рівні шуму.

					ДР.Шс – 23.00.000 ПЗ	Арк.
						38
Змн.	Арк.	№ докум.	Підп.	Дата		

Під час роботи комп'ютера треба працювати на відстані 60-70 см від екрана, дотримуватись правильної постави, не сутулячись і не нахиляючись.

2.2 Загальна структура сайту

Розроблюваний веб-сайт поділений на 3 великі частини (рис.2.1):

- Головна сторінка
- Галерея
- Новини

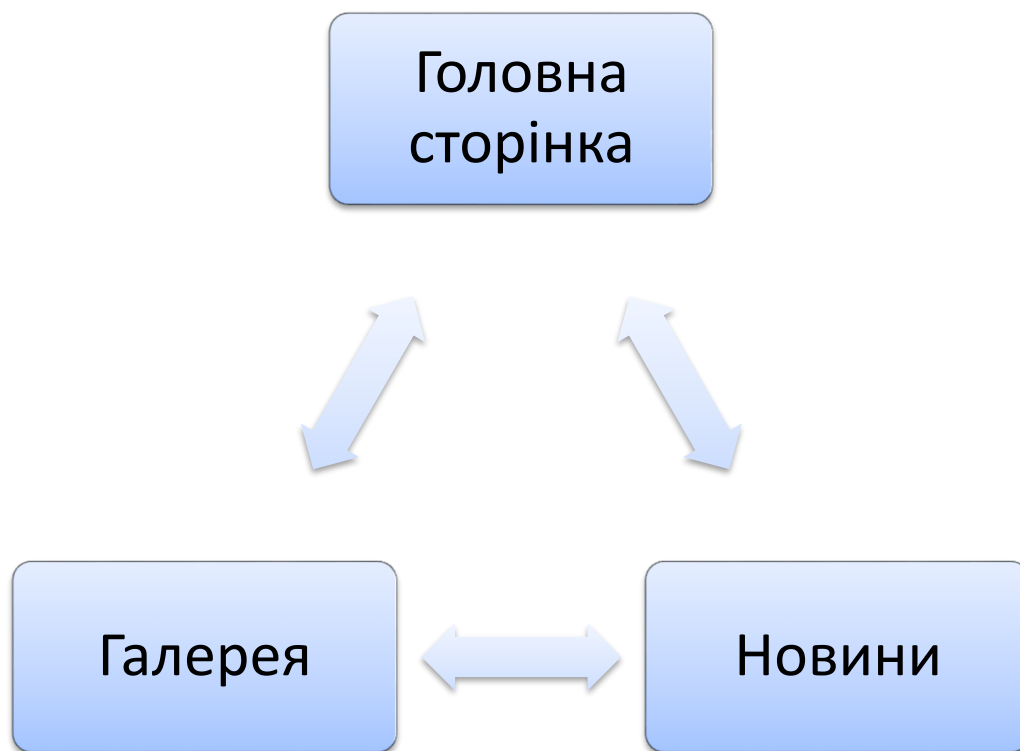


Рисунок 2.1 – Схематична структура сайту

А також окрема сторінка для кожної новини.

На головній сторінці розміщена інформація про назви команд, турнірна сітка і результати матчів, а також header і footer.

Header і footer на всіх сторінках сайту однакові. В header-і розміщене меню для навігації по сайту, а footer це просто лінія на всю ширину екрану

					ДР.Шс – 23.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		39

висотою в 2 рх. І header і footer однакового зеленого кольору, а також цього самого кольору блок, в якому розміщується інформація про окремий матч.

Назви команд розміщені вверху сторінки в 2 рядки, для зручності перегляду, так як при великій кількості команд вони всі не помістилися б на екрані.

Турнірна сітка розрахована на те, що турнір починається з 1/8 фіналу(тобто всього повинно бути до 16 команд учасників). Враховуючи, що сайт розрахований на невеликі змагання цього повністю достатньо.

Блок 1/8 фіналу розділений на ще 8 частин, в яких з лівої сторони записана назва першої команди, а з правої назва другої і, якщо матч вже завершився, то його результат вписується між ними. Блоки 1/4 фіналу, півфінал та фінал побудовані за тим самим принципом, єдина різниця в тому, що вони розділені на меншу кількість частин через те, що половину учасників вибуло на попередньому етапі.

Назви команд і результат їхньої гри вводиться через панель адміністратора. А прохід (чи не прохід) команди до наступного етапу визначає JavaScript. JS збирає інформацію про рахунок, яким закінчився матч і «переносить» команду-переможця до наступного етапу, а також додає для неї на даному (вже пройденому) етапі CSS клас bold, який робить шрифт жирним, завдяки цьому простіше відслідкувати, яка команда проти якої грала, а також одразу побачити переможця, без погляду на результат. JS робить ці дії на кожному етапі змагань.

Наступною сторінкою є галерея. Назва сторінки говорить сама за себе, на цій сторінці розміщуються фотографії зі змагань. Ці фотографії розділені на декілька блоків-галерей, для окремого етапу змагань кожен.

Також, завдяки бібліотеці lightgallery.js їх можна збільшувати на весь екран клікаючи на них, та переключатись між ними, завантажувати і закривати повноекранний перегляд завдяки кнопкам навігації. Ці кнопки добавляються автоматично, але зразу вони виглядають, як прямокутники. Для того, щоб змінити їх вигляд можна підключити спеціальний шрифт, який завантажується в архів разом з галереєю, або кастомізувати самому через CSS. Так як вони

					ДР.Шс – 23.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		40

виглядають красиво і зрозуміло зразу після підключення шрифтів, я не змінював їх вигляду вручну.

Ще одна сторінка це сторінка новини. Створена вона для того, що надати користувачу потрібну йому інформацію, а також зацікавити в пошуку нової інформації.

На ній зберігається коротка інформація про певний матч, команду чи конкретний етап змагань (дату початку, дату закінчення, список учасників, місце проведення тощо), а також посилання на сторінку, на якій відображається повна інформація про ці події.

2.3 Спеціалізовані функції роботи з даними

Для керування сайтом і налаштуванням його «під себе» у WordPress використовується спеціальний файл `functions.php`.

Як правило, в даному файлі задаються додаткові функції для шаблону, що дозволяють змінювати зовнішній вигляд вашого сайту. Однак насправді його можливості значно ширші, оскільки `functions.php` в WordPress завантажується автоматично при ініціалізації системи. Тобто при виклику будь-якої сторінки, як фронтенда (сайту), так і бекенд (адмінки), відповідні функції з даного файлу будуть виконані.[11]

За допомогою файлу `functions.php` можна вирішити цілий спектр абсолютно різних завдань:[12]

- зміна зовнішнього вигляду теми;
- визначення своїх функцій щоб пізніше викликати їх в інших файлах шаблону, що економить час на впровадження змін на сайті, наприклад, функція відображення першої картинки поста, яку після визначення можна використовувати в файлах теми для категорій, тегів, архівів;
- Налаштування адмінки під себе - приховування певних пунктів з меню, зміна різних параметрів відображення для інших користувачів;

					ДР.Шс – 23.00.000 ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підп.	Дата		

- створення спеціальної сторінки налаштувань теми;
- Перевизначення якихось параметрів системи - зміна розмірів мініатюр для шаблону, а також різного роду хаки і фільтри.

Взагалі файл `functions.php` в WordPress за логікою і швидкості роботи нагадує плагіни. Хоча при цьому у них є принципові відмінності. Модулі використовуються, як правило, для вирішення якихось конкретних завдань і застосовуються до будь-якої теми вашого сайту. Плюс для роботи їх потрібно активувати.

Що стосується файлу шаблону `functions.php`, то він може містити багато різних функцій і виконується тільки для тієї теми, в якій визначено. Тобто при зміні шаблону, всі ваші поточні напрацювання будуть відключені, а система почне зчитувати інформацію з іншого файлу функцій

Для покращення читабельності коду і не тільки у WordPress придумані спеціальні функції, більшість з яких описані у файлі `functions.php`.

Наприклад, якщо на всіх сторінках використовується однаковий `header` і `footer`, то замість того, щоб прописувати їх в кожному документі можна створити документ `header.php`, в якому буде зберігатися код для шапки сайту, а підключатися він буде вгорі коду і файл `footer.php`, який буде містити код для підвалу сайту(підключається в кінці коду).[13]

Вигляд коду до використання функцій `get_header` та `get_footer`:

```
<!DOCTYPE html>
<!--[if lt IE 8 ]><html class="no-js ie ie7" lang="en"> <![endif]-->
<!--[if IE 8 ]><html class="no-js ie ie8" lang="en"> <![endif]-->
<!--[if (gte IE 8) || !(IE)]><!--><html class="no-js" lang="en"> <!-->
<![endif]-->

<head>
  <meta charset="UTF-8">
  <?php wp_head() ?>

</head>
```

					ДР.Шс – 23.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		42

```

<body>
  <header class="header">
    <?php wp_nav_menu( array(
      'theme_location' => 'top',
      'container'       => 'ul',
      'menu_class'      => 'nav',
      'menu_id'         => 'nav'
    ) ); ?>
  </header>
  <?php

  $posts = get_posts( array(
    'numberposts' => 1,
    'category'    => '0',
    'post_type'   => 'post',
    'suppress_filters' => true, // подавление работы фильтров
изменения SQL запроса
  ) );

  foreach( $posts as $post ){ setup_postdata($post);
  ?>

  <section class="section">
    <div class="section__title-block"><h2 class="title"><?php
the_title() ?></h2></div>
    <article class="section__text-block">
      <?php the_content() ?>
    </article>
  </section>

  <?php
}

wp_reset_postdata(); // сброс

```

					ДР.ІІс – 23.00.000 ІЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		43

?>

```
<footer class="footer"></footer>
<?php wp_footer(); ?>
<script>
    lightGallery(document.querySelector('.lightgallery'));
</script>
</body>
</html>
```

Вигляд коду після використання функцій `get_header` та `get_footer`:

```
<?php get_header(); ?>

<?php

$posts = get_posts( array(
    'numberposts' => 1,
    'category'     => '0',
    'post_type'    => 'post',
    'suppress_filters' => true, ) );

foreach( $posts as $post ){ setup_postdata($post);
?>

<section class="section">
    <div class="section__title-block"><h2 class="title"><?php
the_title() ?></h2></div>
    <article class="section__text-block">
        <?php the_content() ?>
    </article>
</section>

<?php
}
```

					ДР.ІІс – 23.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		44

```
wp_reset_postdata();  
?>  
<?php get_footer(); ?>
```

Такий код набагато легше змінювати і підтримувати. Крім того, якщо потрібно змінити шапку чи підвал на всіх сторінках, не потрібно переходити до кожного файлу, який містить інформацію про певну сторінку, достатньо просто змінити код в файлах header.php та footer.php.

Підключення стилів та скриптів відбувається в файлі functions.php та викликається через спеціальні команди в header-і та footer-і сайту.

Ось так виглядає підключення CSS стилів у шапці сайту:

```
<head>  
  <meta charset="UTF-8">  
  <?php wp_head() ?>  
</head>
```

А так виглядає підключення JS файлів у підвалі сайту:

```
<footer class="footer"></footer>  
<?php wp_footer(); ?>  
</body>  
</html>
```

Для того, щоб WordPress розумів, як йому працювати з цими функціями, потрібно описати їх у файлі functions.php. Спочатку потрібно описати дію, яку потрібно виконати(її назву і спосіб виклику), а після цього написати функції, які будуть виконуватися при виклику цих дій.[13]

Код для опису дії:

```
add_action( 'wp_enqueue_scripts', 'style_theme' );  
add_action( 'wp_footer', 'scripts_theme' );
```

Ось так виглядає підключення CSS файлів:

```
function style_theme() {  
  wp_enqueue_style( 'style', get_stylesheet_uri() );  
  wp_enqueue_style( 'lightgallery', get_template_directory_uri()  
    . '/assets/css/lightgallery.css' );  
}
```

					ДР.Шс – 23.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		45

```

    wp_enqueue_style( 'main', get_template_directory_uri() .
'/assets/css/main.css' );
    wp_enqueue_style( 'article', get_template_directory_uri() .
'/assets/css/article.css' );
    wp_enqueue_style( 'gallery', get_template_directory_uri() .
'/assets/css/gallery.css' );
    wp_enqueue_style( 'news', get_template_directory_uri() .
'/assets/css/news.css' );
}

```

А так підключення JavaScript файлів:

```

function scripts_theme() {
    wp_enqueue_script( 'jquery' );
    wp_enqueue_script( 'main', get_template_directory_uri() .
'/assets/js/main.js' );
    wp_enqueue_script( 'lightgallery.min.js',
get_template_directory_uri() .
'/assets/js/lightgallery.min.js' );
}

```

Реєстрація навігаційного меню у WordPress теж проходить інакше, ніж в HTML. Для того, щоб WordPress зумів відображати меню на сторінках, потрібно додати дію:[14]

```
add_action( 'after_setup_theme', 'menu' );
```

Де 'after_setup_theme' – це назва функції WordPress, а 'menu' – це будь-яка довільна назва функції, яку прописує програміст у файлі functions.php. Ця функція виглядає так:

```

function menu() {
    register_nav_menu( 'top', 'header menu' );
    add_theme_support( 'title-tag' );
}

```

Після того, як програміст прописав код для дії і функції у файлі functions.php, йому потрібно зареєструвати меню в потрібному місці на сайті. В моєму випадку меню знаходиться у файлі header.php.[15]

```
<header class="header">
```

					ДР.Шс – 23.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		46

```

<?php wp_nav_menu( array(
    'theme_location' => 'top',
    'container'      => 'ul',
    'menu_class'     => 'nav',
    'menu_id'        => 'nav'
) ); ?>
</header>

```

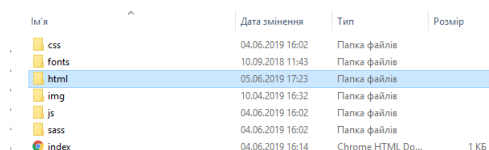
Тепер, коли меню вже зареєстровано, потрібно зайти в панель адміністратора, створити потрібні сторінки і додати їх до головного меню. Якщо після цих всіх кроків перейти на сайт, можна побачити, що меню вже додалось, але відображається воно не так, як у верстці. Це відбувається тому, що WordPress додає до елементів меню свій клас, а той який задавався при верстці не бачить. Для того, щоб меню мало такий вигляд, як в .html файлі, потрібно перейти в панель розробника (кнопка F12 в Google Chrome), знайти там назву класів елементів меню і в .css файлі замінити клас, який прописував я, на той який запропонував WordPress.[16]

					ДР.Шс – 23.00.000 ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підп.	Дата		

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗБЕЧЕННЯ ІНТЕРНЕТ РЕСУРСУ

3.1 Розробка front-end частини програмного забезпечення

Правильно сформована розмітка сайту є запорукою успішної подальшої розробки якісного програмного забезпечення і правильної роботи програми. Тому перед тим, як писати код потрібно спочатку уявити і зрозуміти, як буде виглядати сайт і дерево каталогів. Я скористався деревом, яке пропонує Gulp (рис.3.1), папки, в яких знаходяться CSS, SCSS, JavaScript, HTML файли та картинки розмістити в одній папці «app», також в цій папці потрібно розмістити файл index.html [17].



Ім'я	Дата змінення	Тип	Розмір
css	04.06.2019 16:02	Папка файлі	
fonts	10.09.2018 11:43	Папка файлі	
html	05.06.2019 17:23	Папка файлі	
img	10.04.2019 16:32	Папка файлі	
js	04.06.2019 16:02	Папка файлі	
sass	04.06.2019 16:02	Папка файлі	
index	04.06.2019 16:14	Chrome HTML Do...	1 КБ

Рисунок 3.1 – Дерево каталогів

Перед тим, як почати писати розмітку, стилі і JS я встановив таск-менеджер Gulp в свою папку з проектом. Далі в Gulp я описав задачі які він повинен виконувати, а саме[18]:

- автоматичне перезавантаження сторінки при внесенні будь яких змін в редакторі коду;
- транслювання коду з SCSS в CSS;
- мінімізація JavaScript і CSS коду;
- стискання зображень;
- автоматичне добавлення вендорних префіксів для кросбраузерності проекту;

Так як Gulp написаний на JS, то і код для нього теж пишеться на нативному JavaScript.

					ДР.Шс – 23.00.000 ПЗ	Арк.
						48
Змн.	Арк.	№ докум.	Підп.	Дата		

Після того, як було створено gulpfile.js і описано в ньому всі функції, які він повинен виконувати, потрібно відкрити командний рядок і знайти папку з потрібним проектом (користувачам Windows достатньо перейти в папку з проектом і утримуючи клавішу Shift натиснути праву кнопку миші, після чого вибрати «відкрити командний рядок тут»). Далі в командному рядку потрібно написати gulp і натиснути клавішу Enter[19]. Якщо все зроблено правильно, командний рядок повинен виглядати так (рис. 3.2):

```

gulp
PS E:\Projects\gulp> gulp
[16:34:59] Using gulpfile
[16:34:59] Starting 'browser-sync'...
[16:35:01] Finished 'browser-sync' after
[16:35:01] Starting 'sass'...
[16:35:01] Starting 'scripts'...
Browserslist: caniuse-lite is outdated. Please run next command `npm update caniuse-lite browserslist`
[Browsersync] 2 files changed (
[16:35:02] Finished 'sass' after
[16:35:02] Starting 'css-libs'...
[Browsersync] Access URLs:
-----
Local:
External:
-----
UI:
UI External:
-----
[Browsersync] Serving files from:
[16:35:05] Finished 'scripts' after
[16:35:05] Finished 'css-libs' after
[16:35:05] Starting 'watch'...
[16:35:05] Finished 'watch' after
[16:35:05] Starting 'default'...
[16:35:05] Finished 'default' after

```

Рисунок 3.2 – Командний рядок

Після створення структури проекту і запуску Gulp, можна починати писати код. Для цього потрібно відкрити будь-який редактор коду, я вибрав Sublime Text. Після цього в редакторі я прописав код для розмітки головної сторінки сайту, а саме index.html, CSS стилі і JS для нього. А також заповнив ті поля, які пізніше будуть заповнюватись в панелі адміністратора. Зробив я це для того, щоб побачити, як буде виглядати сайт після наповнення його контентом. В результаті головна сторінка мала такий вигляд (рис.3.3):

КОМАНДИ

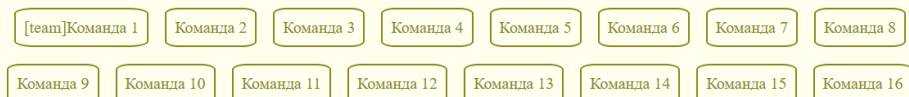


Рисунок 3.3 – Головна сторінка

А так вигий вигляд мала турнірна сітка (рис.3.4):

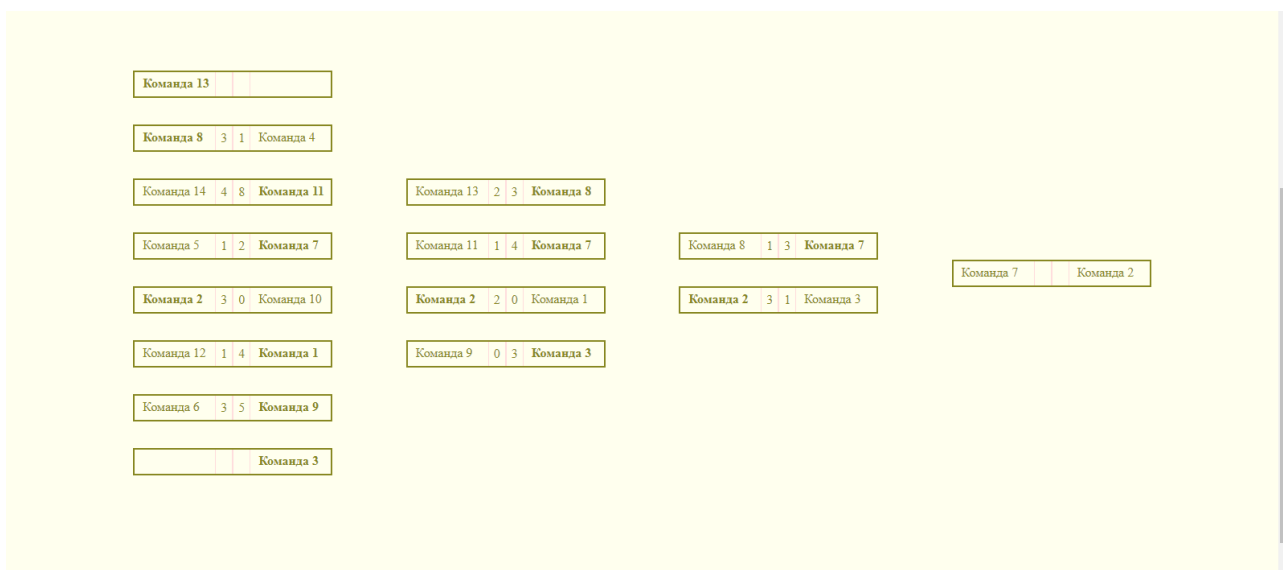


Рисунок 3.4 – Турнірна сітка

Як можна побачити на рисунку, я спеціально ввів меншу кількість команд, ніж передбачено в турнірній таблиці. Це зроблено для того, щоб показати, що при недостатній кількості учасників турнірна сітка не «зламається» і JavaScript автоматично виведе команду, яка не мала суперника в наступний етап.

Крім того JavaScript файл повинен брати результат матчу і виводити команду переможця до наступного етапу, а також додавати для неї CSS клас **bold**, який робить шрифт жирним.

Змн.	Арк.	№ докум.	Підп.	Дата

Наступний файл, який я написав був gallery.html. Назва файлу говорить сама за себе, це галерея. Тут розміщені фотографії з різних етапів змагань і будь-хто бажающий може їх подивитися (рис.3.5). Ось, як виглядає ця сторінка:

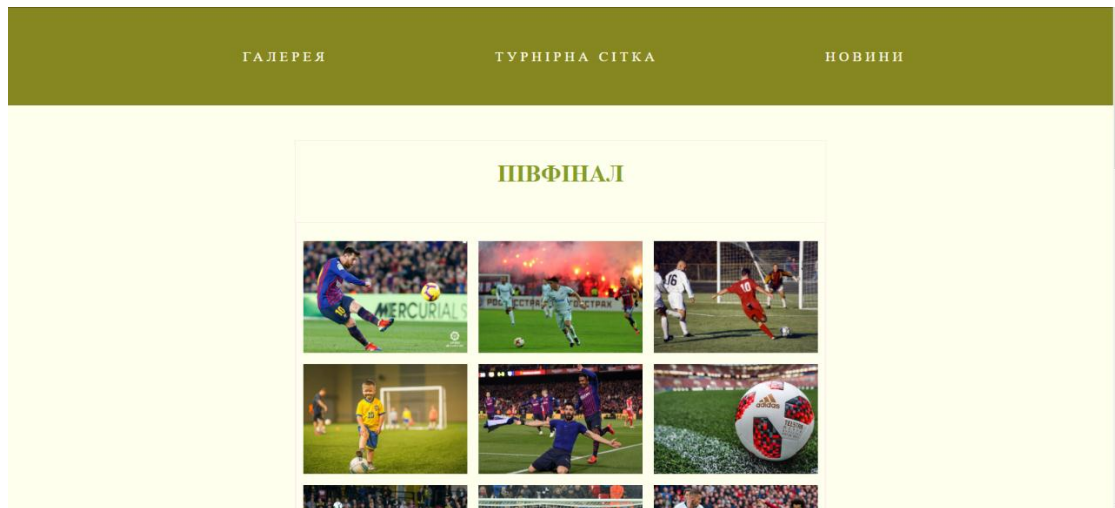


Рисунок 3.5 – Галерея

При кліку на зображення, воно збільшується, а також додаються кнопки навігації, які дозволяють переходити від одного зображення до іншого, завантажувати та закривати його.



Рисунок 3.6 – Окреме зображення

					ДР.Шс – 23.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		51

Це зроблено за допомогою бібліотеки lightgallery.js.[19] Для її використання спочатку потрібно підключити jQuery. Ця бібліотека добре задокументована, тому в її підключенні не виникає жодних проблем чи незручностей. Потрібно просто завантажити архів з усіма потрібними файлами і розпакувати його в папці з потрібним проектом. Після чого підключити стилі, шрифт і JS файли до вашого проекту. І слідує інструкціям додати ідентифікатор або клас до блоку, який потрібно зробити галереєю. Також було створено файл news.html.



Рисунок 3.6 – Новини

В цьому файлі повинні бути коротко описані останні новини змагань, дати їхнього проведення, інформація про ігри, які вже закінчились, а також посилання на повну статтю, яка знаходилась у файлі article.html.

					ДР.Шс – 23.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		52

ФІНАЛІ. ПРОГНОЗИ ЕКСПЕРТІВ

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Temporibus error quod necessitatibus dicta, dolore est blanditiis similique dolorum inventore corporis nesciunt, commodi minima recusandae quam laudantium in voluptas sunt?

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Temporibus error quod necessitatibus dicta, dolore est blanditiis similique dolorum inventore corporis nesciunt, commodi minima recusandae quam laudantium in voluptas sunt? Dolorum. Sit amet, consectetur adipisicing elit. Incidunt expedita quod recusandae, numquam laboriosam eum maxime exercitationem et eligendi, laborum consequuntur optio, dolores! Accusamus officia incidunt dolorem dolores voluptate molestias! Lorem ipsum dolor sit amet, consectetur adipisicing elit. Illum sint architecto, culpa asperiores dignissimos ducimus amet molestias minus accusamus, dicta voluptatibus eum officia, quod. Vel, neque modi nulla harum voluptatum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Temporibus error quod necessitatibus dicta, dolore est blanditiis similique dolorum inventore corporis nesciunt. Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ex, harum, dignissimos? Quis iusto ducimus repellendus maxime ex totam similique, necessitatibus quisquam in ad commodi tempora hic neque, impedit iure beatae. Commodi minima recusandae quam laudantium in voluptas sunt? Dolorum culpa asperiores dignissimos ducimus amet molestias minus accusamus, dicta voluptatibus eum officia, quod. Vel, neque modi nulla harum voluptatum? Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ex, harum, dignissimos? Quis iusto ducimus repellendus maxime ex totam similique,

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Temporibus error quod necessitatibus dicta, dolore est blanditiis similique dolorum inventore corporis nesciunt. Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ex, harum, dignissimos? Quis iusto ducimus repellendus maxime ex totam similique, necessitatibus quisquam in ad commodi tempora hic neque, impedit iure beatae. Commodi minima recusandae quam laudantium in voluptas sunt? Dolorum culpa asperiores dignissimos ducimus amet molestias minus accusamus, dicta voluptatibus eum officia, quod. Vel, neque modi nulla harum voluptatum? Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ex, harum, dignissimos? Quis iusto ducimus repellendus maxime ex totam similique, necessitatibus quisquam in ad commodi tempora hic neque, impedit iure beatae.

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Temporibus error quod necessitatibus dicta, dolore est blanditiis similique dolorum inventore corporis nesciunt, commodi minima recusandae quam laudantium in voluptas sunt? Dolorum.

[Інші новини...](#)

Рисунок 3.7 – Відображення статті на сайті

3.2 Розробка back-end частини програмного забезпечення

Back-end частину даного проекту я розробляв на CMS під назвою WordPress, на мові PHP. Для того, щоб вона працювала потрібно було встановити локальний веб сервер, я вибрав Open Server Panel[22].

Після установки OS Panel потрібно завантажити сам WordPress з офіційного сайту, далі розпакувати завантажений архів і скопіювати файли в папку domains, в якій перед цим потрібно створити папку з назвою проекту, у мене це папка з назвою «diploma». Виглядає це так (рис.3.8):

					ДР.Шс – 23.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		53

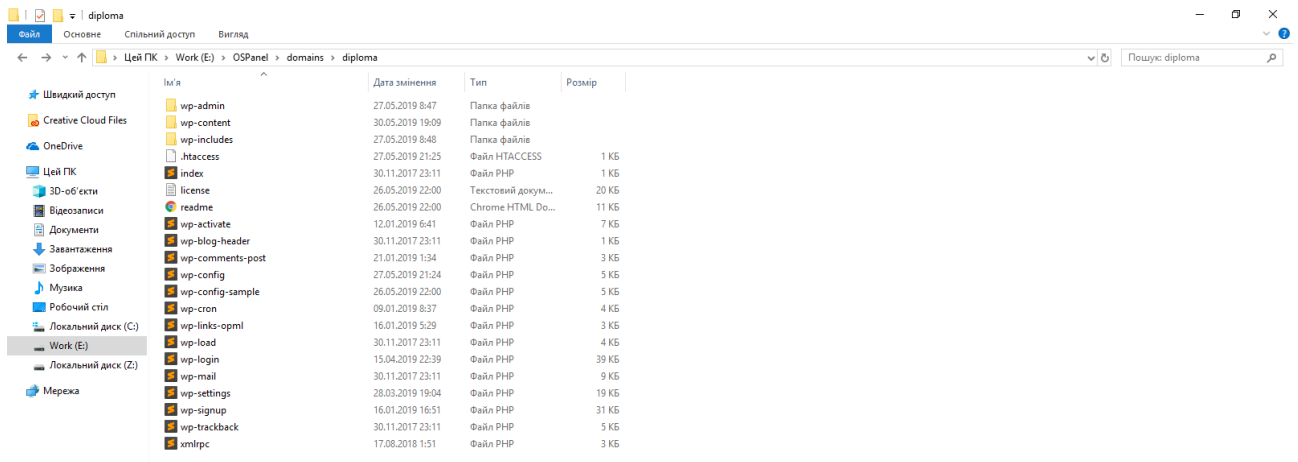


Рисунок 3.8 – Структура OS Panel

Коли всі ці кроки зроблені потрібно перейти в папку wp-content, далі в themes(теми) і вже там створити папку з назвою власної теми(її я теж назвав diploma), а вже в цій папці треба зберігати всі файли проекту. Просто скопіювати всі файли, які я зробив раніше не вийде, так як WordPress не бачить файлів формату .html, проте можна створити файли формату .php, а в них вставити код з .html файлів[23]. JavaScript і CSS файли WordPress бачить, тому їх змінювати не потрібно. Всі ці файли потрібно просто скопіювати в папку assets, яку треба створити в папці diploma. Після цього потрібно запустити OS Panel. На панелі інструментів з'явиться іконка схожа на прапорець (рис.3.9). Потрібно клікнути на неї мишкою і в меню, яке «вискочить» після кліку вибрати Запустити[24]:

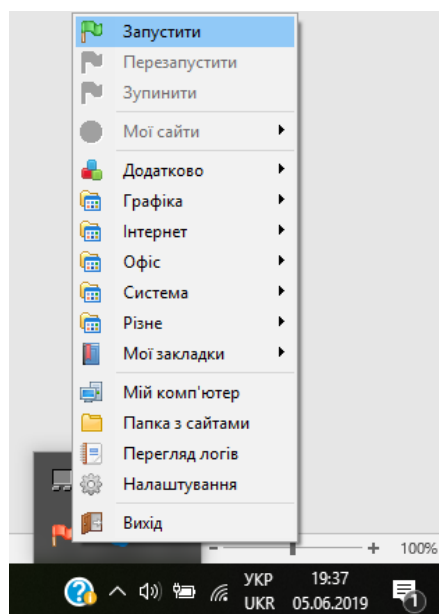


Рисунок 3.9 – Запуск OS Panel

					ДР.Шс – 23.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		54

Після запуску Open Server Panel потрібно вибрати сайт, який ви хочете відкрити (рис.3.10), в моєму випадку це diploma:

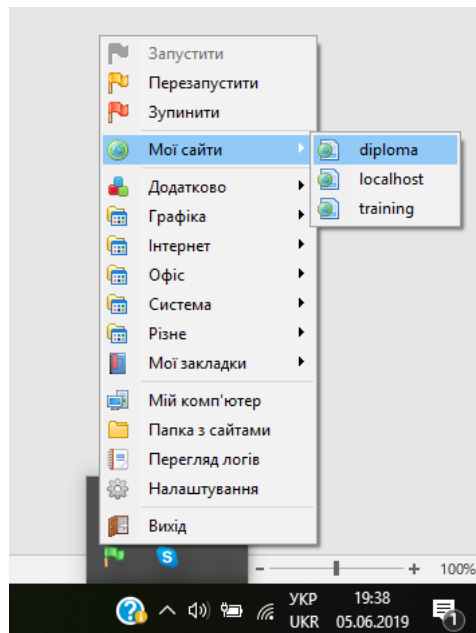


Рисунок 3.10 – Відкриття сайту через OS Panel

Коли це все зроблено, можна побачити, що WordPress просить ввести ім'я користувача і пароль (рис.3.11), тут треба ввести ім'я користувача «root», а поле пароль залишити порожнім[25]:

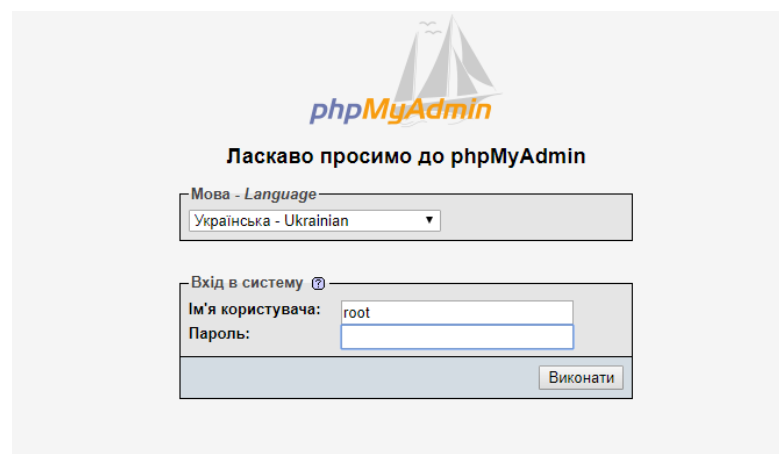


Рисунок 3.11 – Авторизація

Після цього відкриється сторінка phpmyadmin, яка буде виглядати так (рис.3.12):

					ДР.Шс – 23.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		55

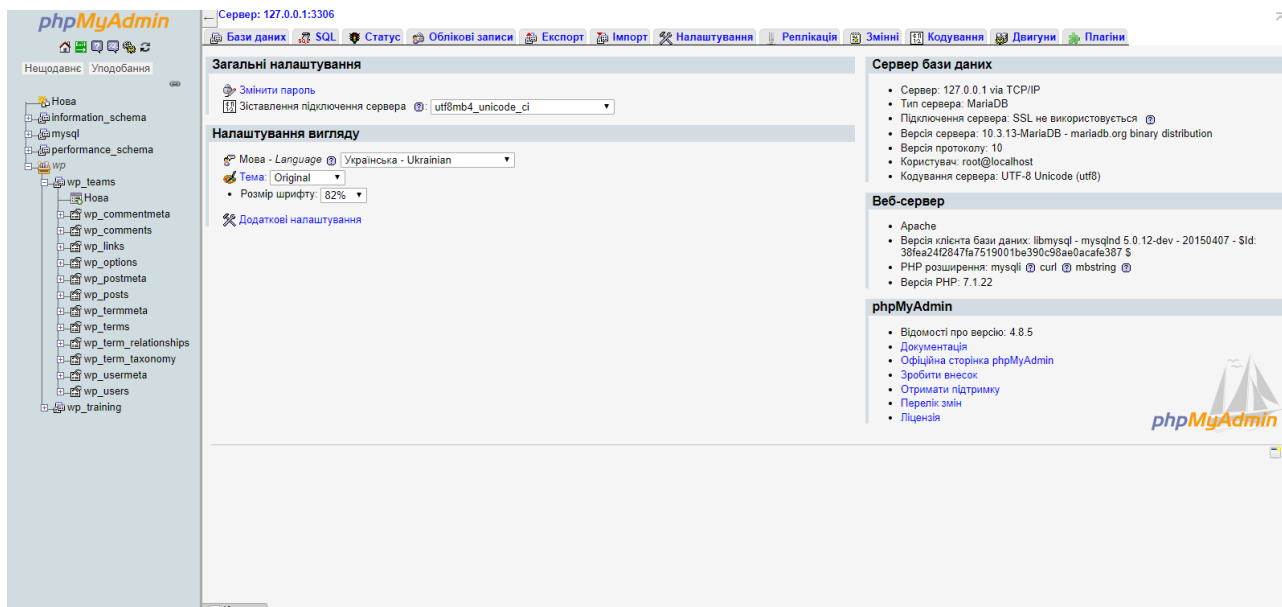


Рисунок 3.12 – РНРMyAdmin

На цій сторінці потрібно створити базу даних, яку потім використовуватиме WordPress (рис.3.13).

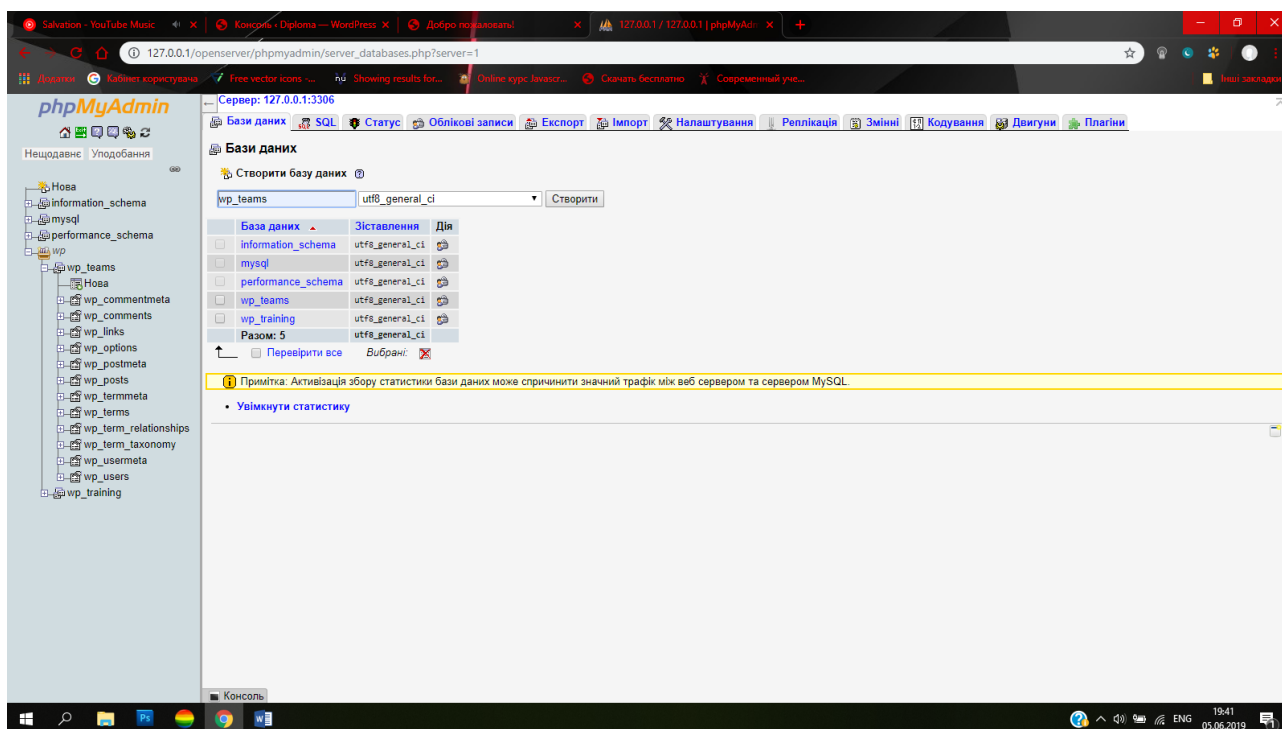


Рисунок 3.13 – База даних WordPress

Тепер, щоб мати права для зміни даних в цій таблиці потрібно створити нового користувача і надати йому привілеї для керування сторінкою(рис.3.14).

					ДР.Шс – 23.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		56

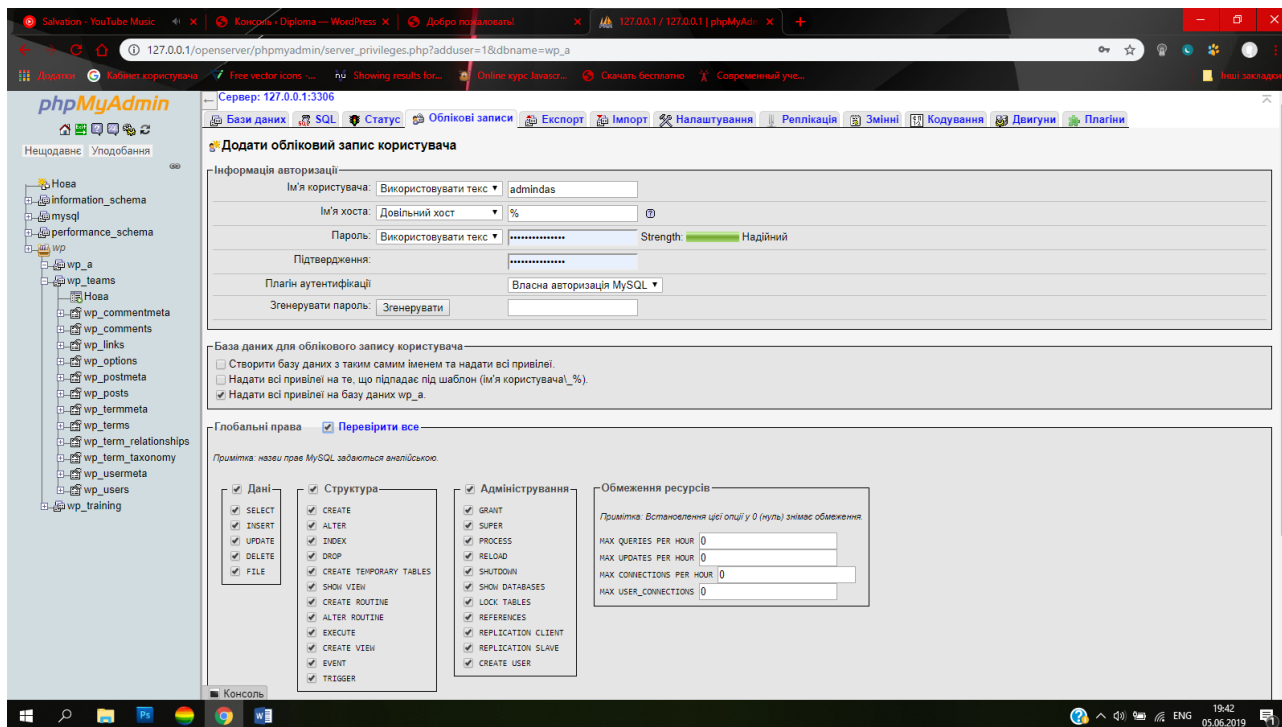


Рисунок 3.14 – Створення нового користувача

Після створення нового користувача і надання йому привілей сторінки, переходимо на наш сайт але, як бачимо на ньому зовсім не та інформація, яка нам потрібна (рис.3.15).

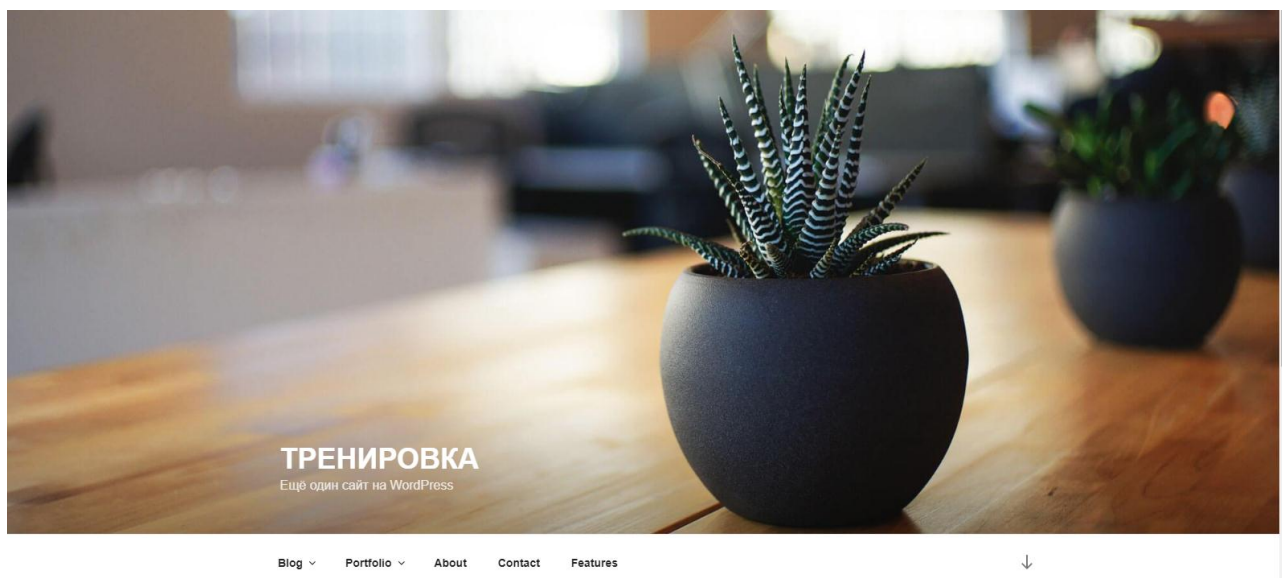


Рисунок 3.15 – Тема WordPress

					ДР.Шс – 23.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		57

Це все відбулось через те, що показ контенту у WordPress залежить від активної теми, а за умовчуванням активна тема Twenty Seventeen. Для того, щоб змінити цю тему і встановити потрібну, для початку потрібно перейти в консоль(панель адміністратора)[28].

Для цього можна натиснути на “консоль” в верхньому лівому кутку сайту, або в рядку де пишеться адреса сайту ввести, адреса сайту/admin. Перевага такого входу полягає в, тому що на сайті не потрібно створювати ніяких посилань для входу в адмін-панель і відповідно відкрити форму входу може тільки та особа яка знає необхідне слово, у випадку цього сайту це Admin, але обране слово може бути будь-яким в залежності від уподобань власника сайту чи адміністратора[29]. Готово. Тепер ми в панелі адміністратора (рис.3.16).

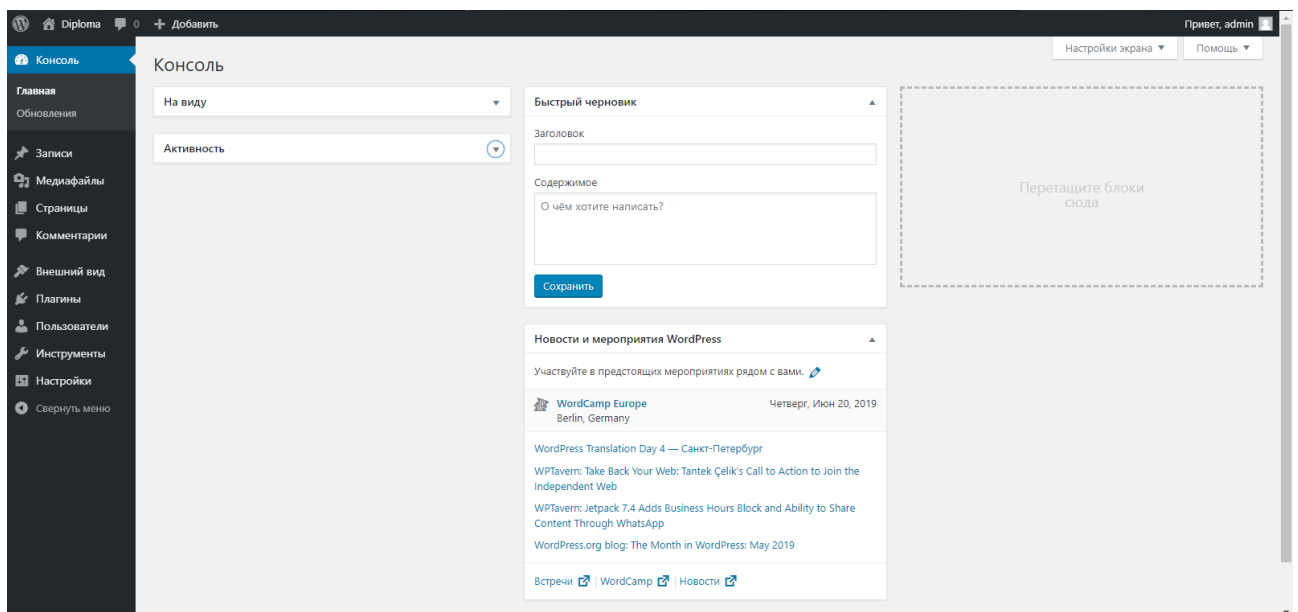


Рисунок 3.16 – Панель адміністратора

Тепер потрібно перейти в меню зліва натиснути на розділ зовнішній вид і там вибрати теми (рис.3.17).

					ДР.Шс – 23.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		58

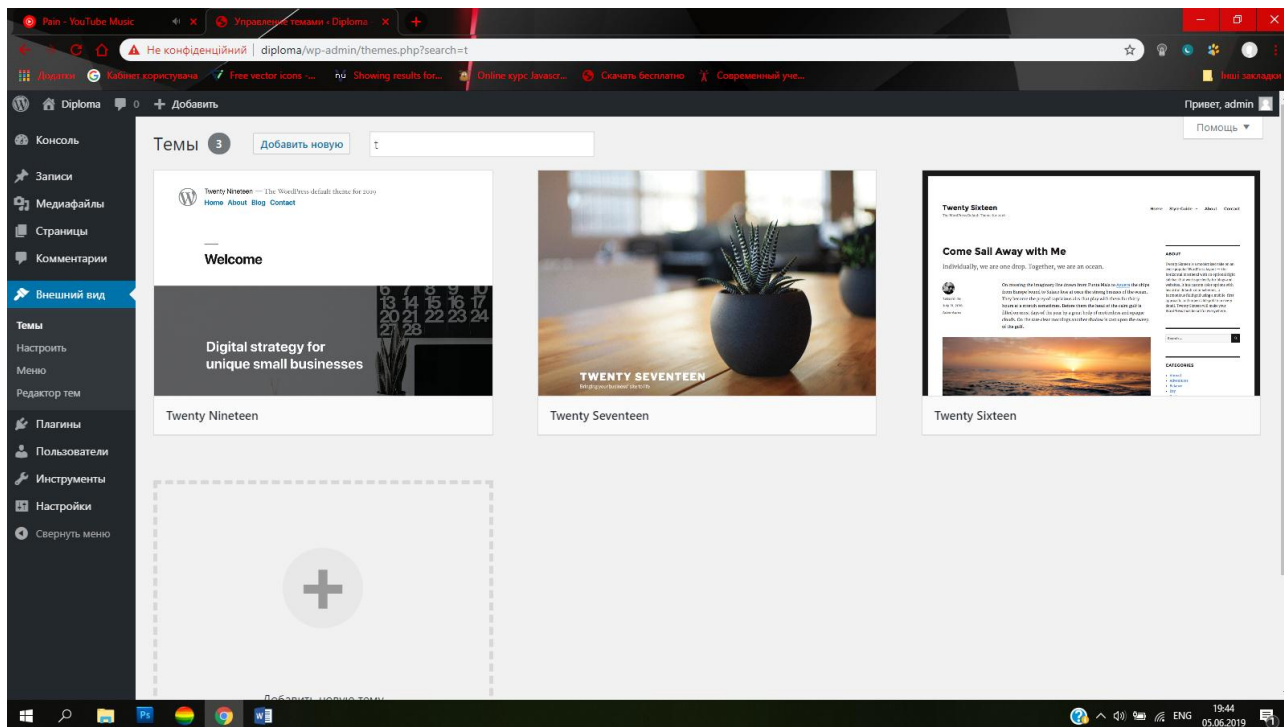


Рисунок 3.17 – Стандартні теми WordPress

Можна побачити, що є три теми, але жодна з них мені не підходить. Для того, щоб створити свою тему потрібно зробити 2 кроки. Перший крок створити файли `index.php` та `style.css`. Другий крок вписати на початку CSS файлу такий КОД:

```
/*
Theme Name: Diploma
*/
```

Для того, щоб додати зображення теми потрібно зробити скрін сайту (або просто взяти будь-яку картинку) перейменувати його в `screenshot.png` і перемістити в кореневу папку проекту[30]. Тепер на вкладці теми з'явилась нова тема, яку я створив (рис.3.18).

					ДР.Шс – 23.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		59

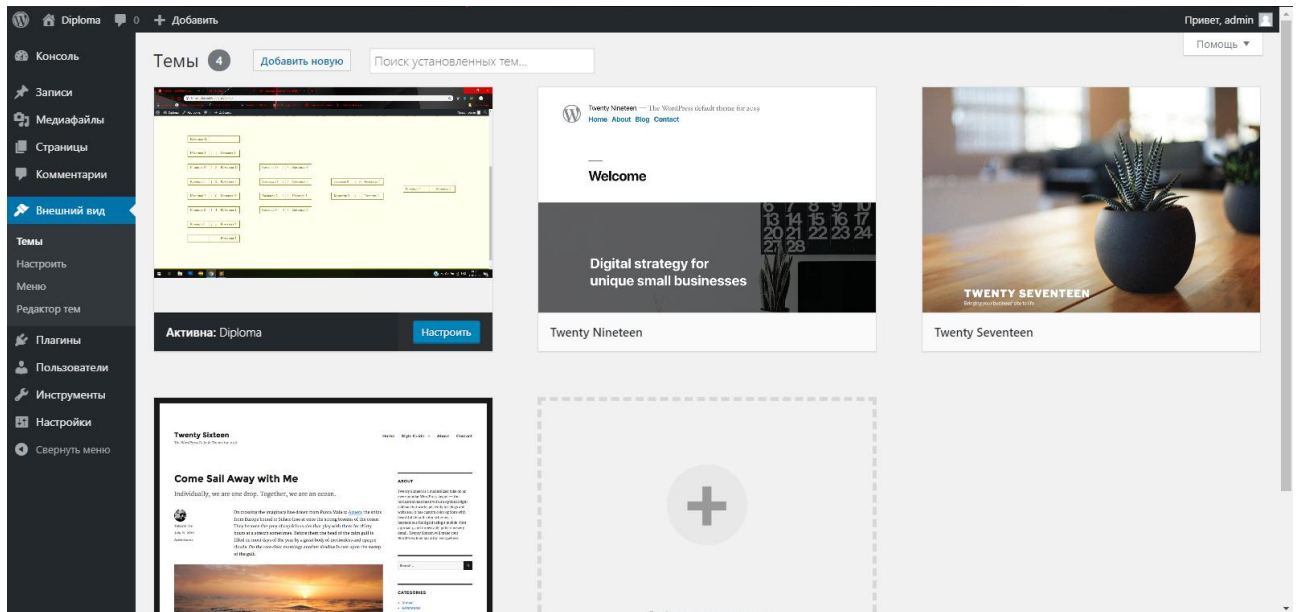


Рисунок 3.18 – Теми WordPress

Тепер потрібно навести мишкою на тему і клікнути на кнопку «активировать». Тепер активна тема це тема, яку я створив. Тепер, для того, щоб моя верстка коректно відображалась, потрібно змінити шлях до CSS та JavaScript файлів, а також створити файли page.php замість news.html, single.php замість article.html та gallery.php замість gallery.html, а також header.php та footer.php для відображення шапки і підвалу сайту.

Також потрібно створити файли 404.php для показу помилки, якщо користувач ввів неправильну адресу та functions.php для того, щоб підключати різні WordPress функції для сайту, такі як підключення стилей, скриптів, меню шорткодів та зображень.

Можна помітити, що header і footer виглядають однаково на всіх сторінках сайту, тому можна вирізати весь код в усіх .php (.html) до закриваючого тегу header і вставити його в header.php. Також можна вирізати увесь код починаючи з відкриваючого тегу footer і вставити його в footer.php. Тепер можна видалити аналогічний код на всіх сторінках, а замість нього прописати php код `<?php get_header?>` на початку сторінки та, відповідно, `<?php get_footer(); ?>` вкінці, щоб підключити header і footer..

					ДР.Шс – 23.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		60

Також, можна видалити всі теги script і link з header.php і footer.php і підключити їх в functions.php:

```
<?php
add_action( 'wp_enqueue_scripts', 'style_theme' );
add_action( 'wp_footer', 'scripts_theme' );
function style_theme() {
    wp_enqueue_style( 'style', get_stylesheet_uri() );
    wp_enqueue_style( 'lightgallery', get_template_directory_uri()
. '/assets/css/lightgallery.css' );
    wp_enqueue_style( 'main', get_template_directory_uri() .
'/assets/css/main.css' );
    wp_enqueue_style( 'article', get_template_directory_uri() .
'/assets/css/article.css' );
    wp_enqueue_style( 'gallery', get_template_directory_uri() .
'/assets/css/gallery.css' );
    wp_enqueue_style( 'news', get_template_directory_uri() .
'/assets/css/news.css' );
}
function scripts_theme() {
    wp_enqueue_script( 'jquery' );
    wp_enqueue_script( 'main', get_template_directory_uri() .
'/assets/js/main.js' );
    wp_enqueue_script( 'lightgallery.min.js',
get_template_directory_uri() . '/assets/js/lightgallery.min.js' );
}
```

Тепер сайт буде виглядати саме так, як і потрібно, але для того, що додавати нову інформацію, текст, картинки чи команду потрібно буде міняти код, а так як це важко для людини яка ніколи не стикалась з програмуванням, цей варіант не підходить. Тому потрібно прописати у файлах .php функції, які дозволять змінювати дані на сайті прямо з панелі адміністратора.

Для початку потрібно замінити існуюче меню, елементи якого прописані в php коді. Зробити це можна також за допомогою functions.php.

```
add_action( 'after_setup_theme', 'menu' );
function menu() {
```

					ДР.Шс – 23.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		61

```

register_nav_menu( 'top', 'header menu' );
add_theme_support( 'title-tag' );
add_theme_support( 'post-thumbnails', array('post') );
}

```

Після цього в header.php потрібно написати код для відображення меню:

```

<header class="header">
    <?php wp_nav_menu( array(
        'theme_location' => 'top',
        'container'       => 'ul',
        'menu_class'      => 'nav',
        'menu_id'         => 'nav'
    ) ); ?>
</header>

```

Тепер в панелі адміністратора можна вибрати меню та налаштувати його. Для цього потрібно вибрати сторінки, створити новою сторінку, ввести назву сторінки меню і натиснути готово (рис.3.19).

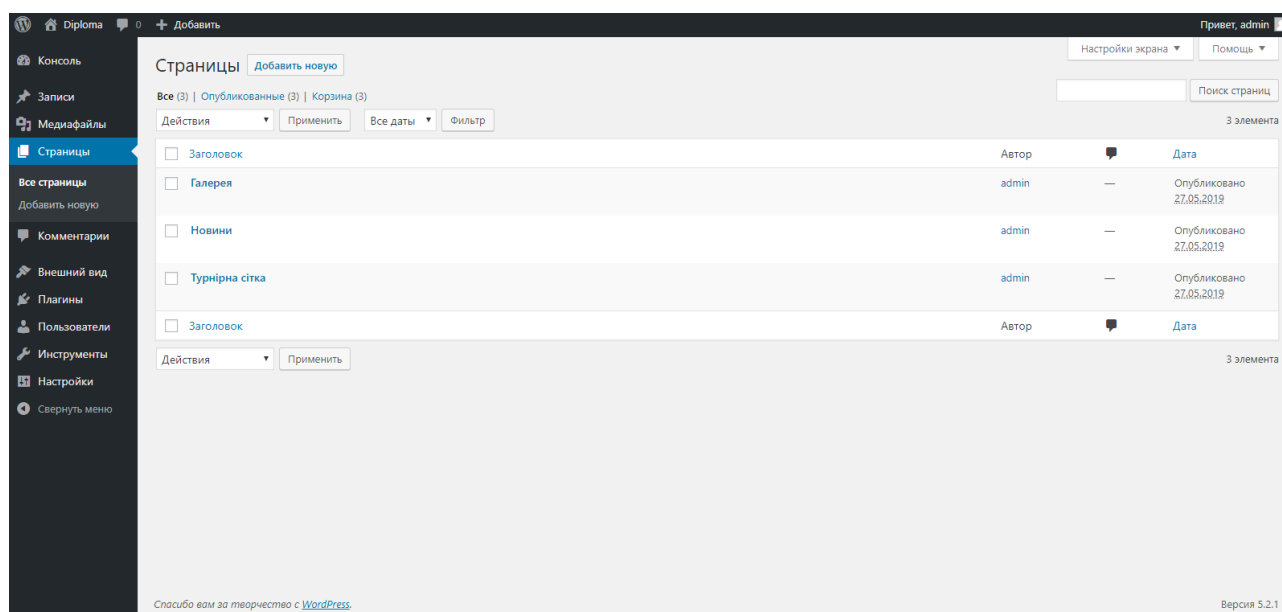


Рисунок 3.19 – Сторінки

Після того, як створив достатню кількість сторінок потрібно зареєструвати їх. Для цього потрібно перейти на вкладку зовнішній вигляд і вибрати меню. Там відмітити галочкою потрібні пункти, натиснути додати в меню, після чого

					ДР.Шс – 23.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		62

розмістити пункти меню в потрібному порядку і натиснути кнопку сохраниць меню (рис.3.20).

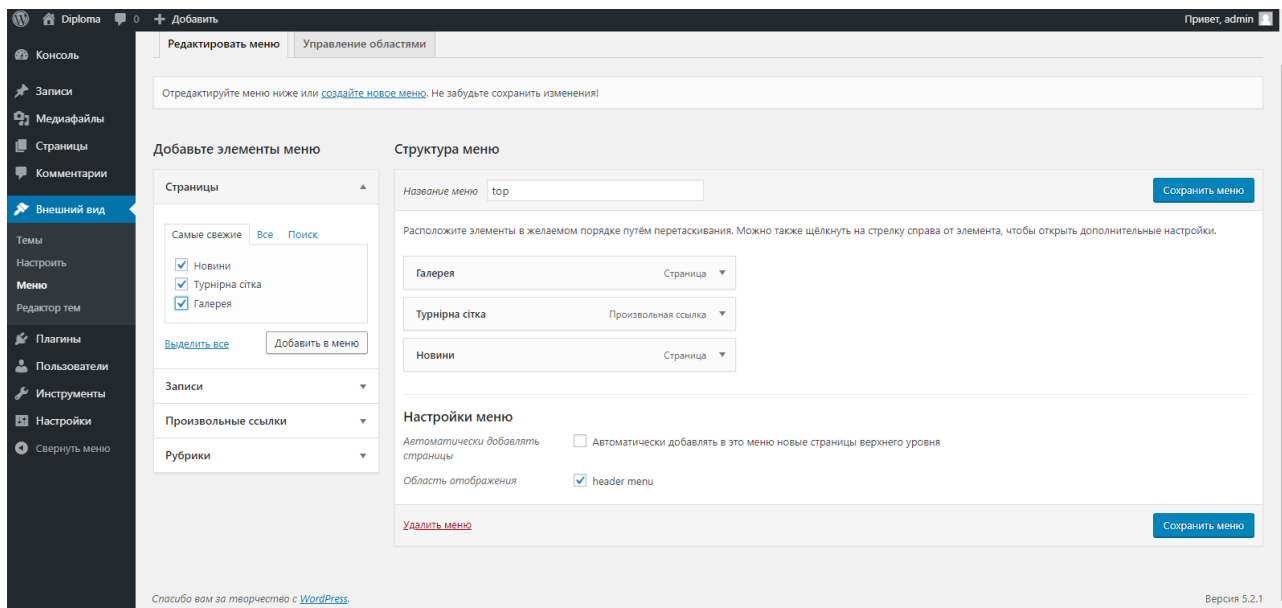


Рисунок 3.20 – Меню WordPress

Тепер, коли вже є можливість переключатись між сторінками, можна зайнятись кожною сторінкою окремо. Для того, щоб додати запис на сторінку, потрібно перейти в записи і створити новий (рис.3.21).

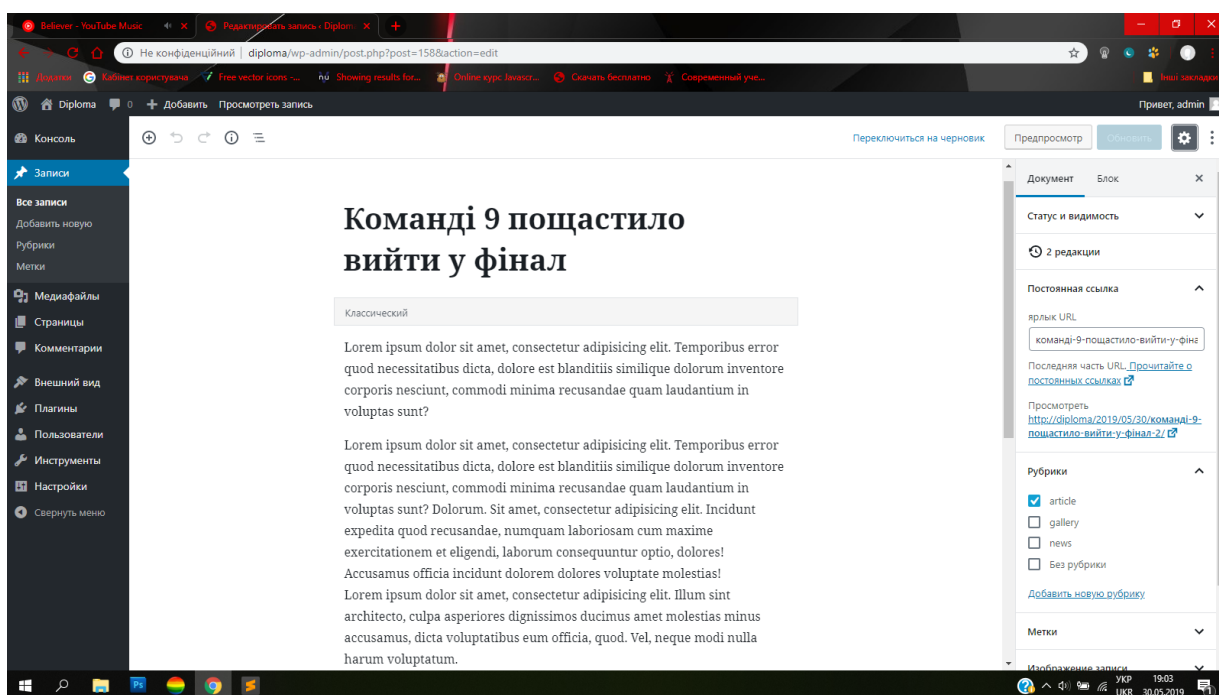


Рисунок 3.21 – Приклад запису для сторінки single.php

					ДР.Шс – 23.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		63

Для того, щоб вибирати, який запис добавляти на конкретну сторінку, потрібно зайти в рубрики, створити нову рубрику і сортувати записи вже по рубрикам (рис.3.22).

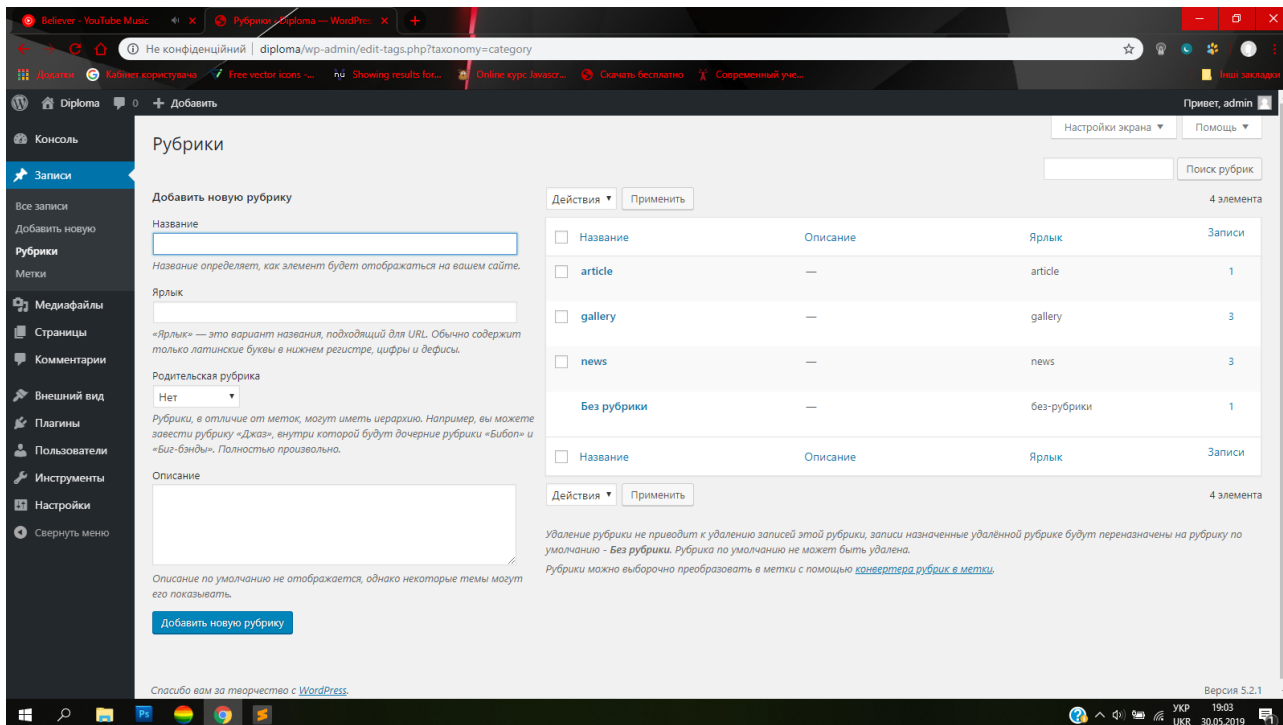


Рисунок 3.22 – Рубрики

А вже для того, щоб WordPress знав, яку рубрику показувати на сторінці потрібно на цій сторінці ввести код в php файлі.

```
$posts = get_posts( array(
    'numberposts' => 5,
    'category'     => '3',
    'post_type'   => 'post',
) );
```

І в category, ввести потрібний номер певної категорії.

Щоб WordPress знав де показувати інформацію із запису треба в потрібному місці прописати php код `<?php the_title() ?>` для заголовку та `<?php the_content() ?>` для контенту(тексту, зображень тощо).

Картинки на сайт додаються майже тим самим способом, що й текст (рис.3.23).

					ДР.Шс – 23.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		64

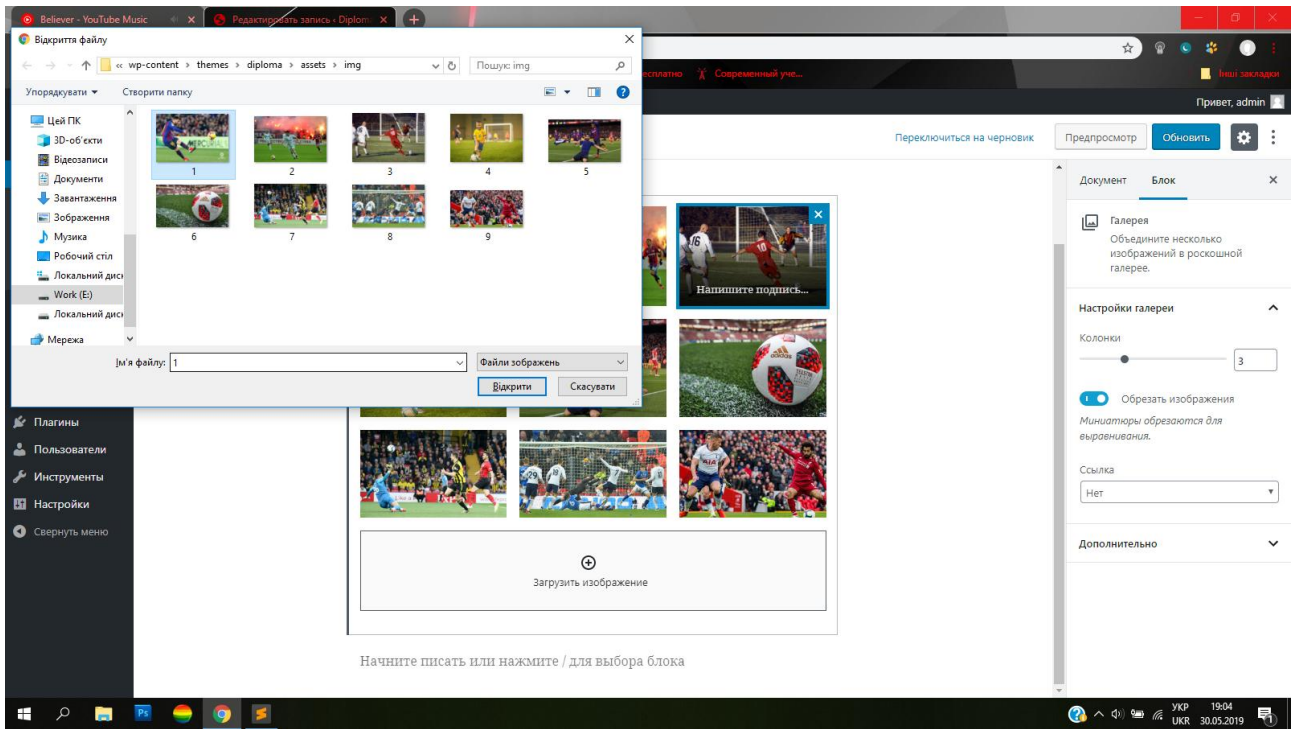


Рисунок 3.23 – Додавання галереї у WordPress

Вигляд запису після додавання (рис.3.24):

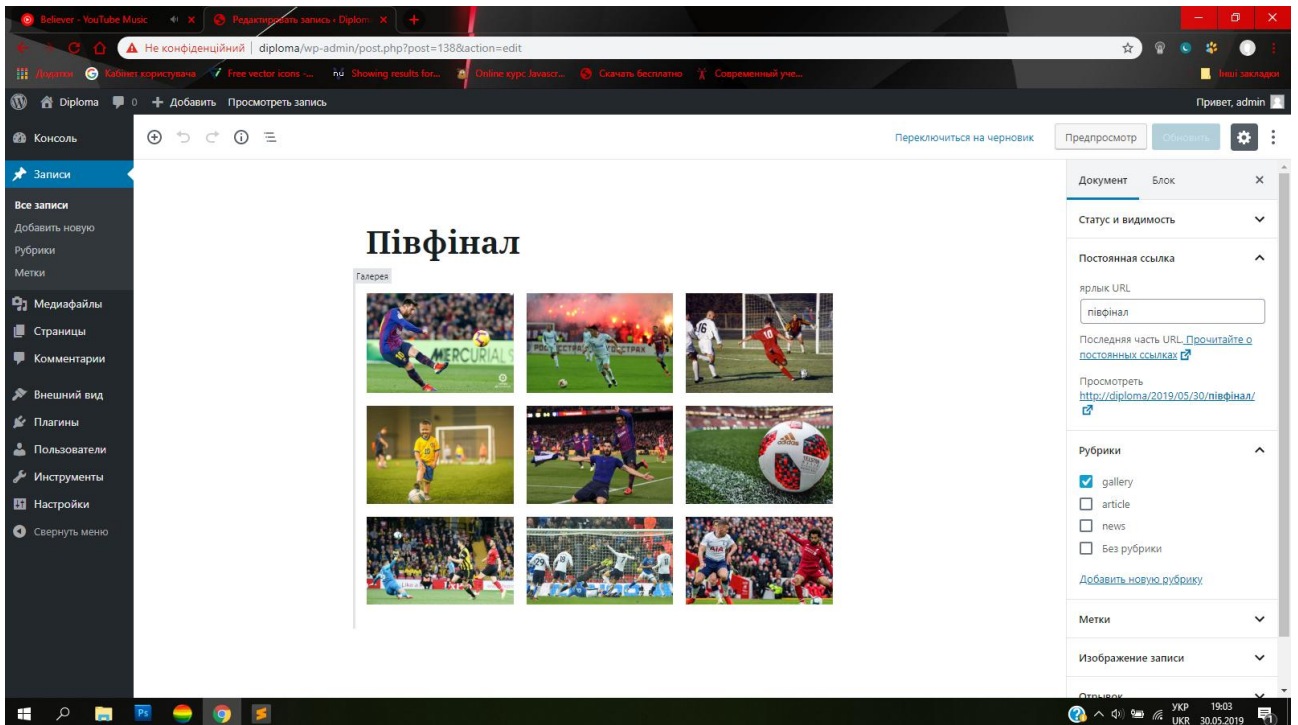


Рисунок 3.24 – Галерея у WordPress

					ДР.Шс – 23.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		65

А вже для головної сторінки важче добавляти текст. Тут потрібно скористатися шорткодами [24]. З їхньою допомогою можна вставляти свій текст безпосередньо в запис за допомогою квадратних дужок. Але для їхнього використання потрібно додати код в functions.php.

```
add_shortcode( 'team', 'baztag_func' );
function baztag_func( $atts, $content ) {
    return "$content";
}
add_shortcode( 'result', 'result_func' );
function result_func( $atts, $content ) {
    return "$content";
}
```

Так буде виглядати код для головної сторінки (рис.3.25):

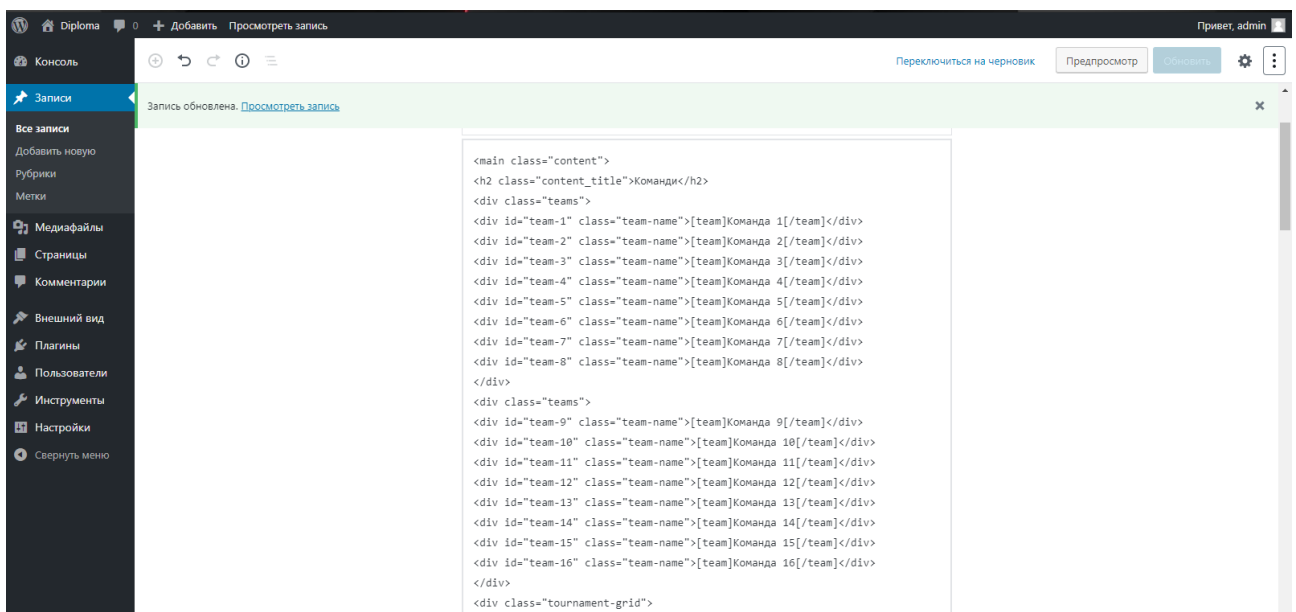


Рисунок 3.25 – Код головної сторінки

А так його побачить адміністратор сайту (рис.3.26):

					ДР.Шс – 23.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		66

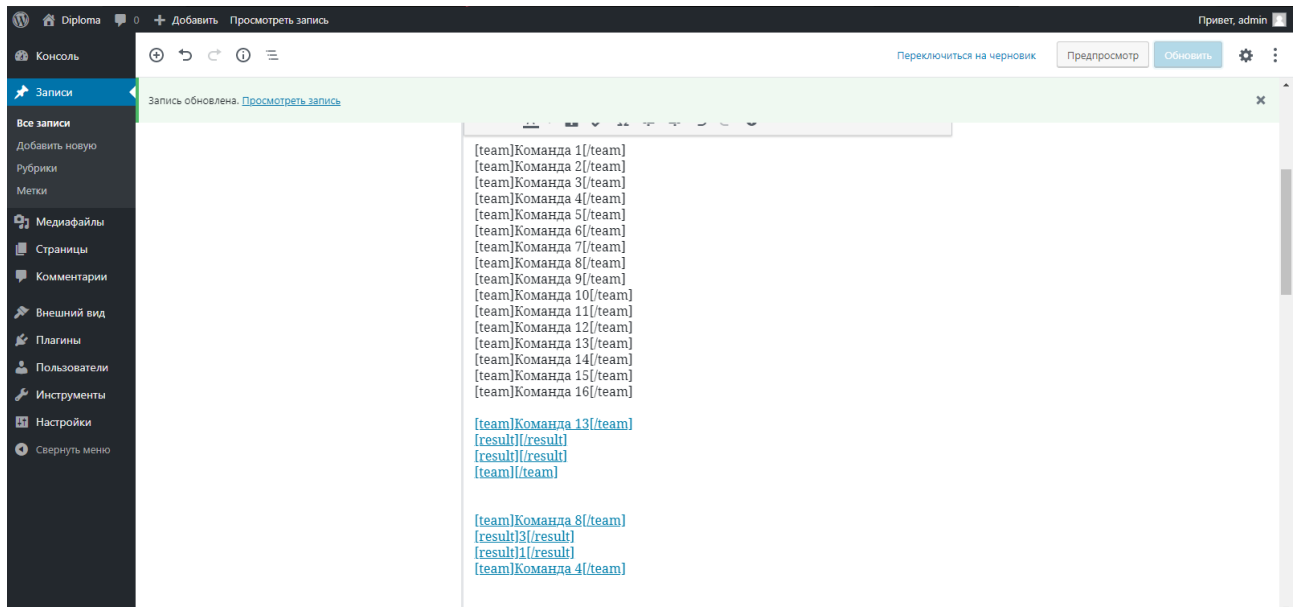


Рисунок 3.26 – Вигляд запису головної сторінки

Всі записи, які я використав для представлення сайту (рис.3.27):

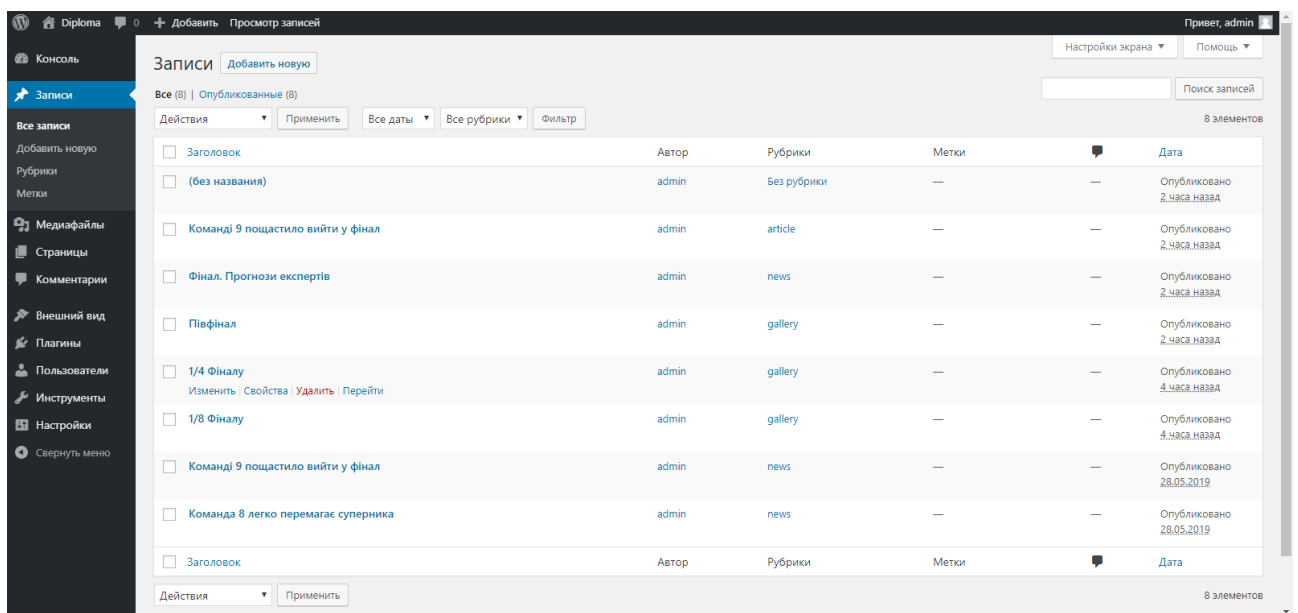


Рисунок 3.27 – Записи на панелі адміністратора

ВИСНОВКИ

Під час виконання даного дипломного проекту було розроблено веб-сайт для формування турнірної сітки спортивних змагань за олімпійською системою.

Веб-додаток реалізований за допомогою веб технологій таких як JavaScript, Gulp (зборщик проектів на JS), SCSS (CSS препроцесор). Back end реалізований за допомогою CMS WordPress та PHP.

Проект розроблений з розрахунком на подальше вдосконалення і додавання нових функціональних можливостей(додавання нових турнірів, збільшення кількості учасників). Тому інтеграція нових функцій займатиме мінімум часу і зусиль.

Тестування сайту відбувалося в тестуванні додатку в різних браузерях (Google Chrome, Mozilla, Opera, Internet Explorer). Необхідність такого тестування полягає в тому, щоб виявити і виправити помилки, допущенні під час розробки. В результаті тестування виявилось, що завдяки Gulp, сайт працював однаково в усіх популярних браузерах.

Результатом роботи є повністю функціональний, протестований і готовий до використання веб-додаток.

Дипломний проект включає в себе вичерпний теоретичний і практичний опис проекту та кожної його частини. Для зручності користування створено додаток, у вигляді посібника для користувачів, який містить інструкцію з описом всіх сторінок, компонентів і кнопок.

Дипломний проект виконано у повній відповідності до завдання і всіма нормативними вимогами. Додаток дозволяє адміністратору керувати (додавати, редагувати, видаляти) всіма даними, які відображаються в програмі.

					ДР.ІІс – 23.00.000 ПЗ	Арк.
						68
Змн.	Арк.	№ докум.	Підп.	Дата		

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Липпман С., Лажойе Ж. Язык программирования JavaScript. Вводный курс. С-Пб, Невский проспект, 2006. 1406 с.
2. Marijn Haverbeke-Вячеслав Голованов, Выразительный JavaScript: Введение Пер. с англ. «Питер», 2016. 433 с.
3. Дэвид Сойер Макфарланд. JavaScript и jQuery. Исчерпывающее руководство. Пер. с англ. М.: Издательский дом «Вильямс», 2017. 450 с.
4. Дэвид Флэнаган, JavaScript. Карманный справочник. Пер. з англ. «Питер», 2015. – 600с.
5. Дэвид Флэнаган, JavaScript – Подробное руководство 6-ое издание Пер. с англ. М.: Издательский дом «Вильямс», 2015. — 300 с.
6. Морето С. JavaScript и jQuery. Исчерпывающее руководство. С. Морето Пер. с англ. Издательский дом «Вильямс», 2017 — 880 с.
7. Myscore. URL: <http://www.myscore.com.ua> (дата звернення: 15.11.2018).
8. Terricon. URL: <http://www.terricon.com> (дата звернення: 16.11.2018).
9. SkyBet. URL: <http://www.skybet.com> (дата звернення: 16.11.2018).
10. Правила охорони праці під час експлуатації електронно-обчислювальних машин: ДНАОП 0.00-1.31-99 200684, К : Держнагляд охорони праці України 1999, 65с.
11. Програмування. Практикум Укл.: Семенюк А.Д., Сопронюк Ф.О. Чернівці: Рута, 2001.– 143 с.
12. Симонович С., Евсеев Г. Практическая информатика: универсальный курс. Инфорком–Пресс, 1999. 480 с.
13. Семчук А.Р., Юрченко І.В. Економічна інформатика. Навчальний посібник.– Чернівці: МВІЦ "Місто", 2008. 426 с.
14. Мэтью Макдональд, Веб-разработка. 2-е изд.: Пер. с англ. М.: Издательский дом "Вильямс", 2015. 640 с.

					ДР.Шс – 23.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		69

15. Хорстманн К. С., Корнелл Г. Библиотека профессионала. JavaScript Том 1. Основы, 7-е изд.: Пер. с англ. – М.: Издательский дом "Вильямс", 2007. – 896 с.
16. Дерил Бартлетт, WordPress для начинающих. Пер. с англ. Эксмо, 2016 208с.
17. PHP. URL: <https://www.php.net/manual/ru> (дата звернення: 6.02.2019).
18. Learn JavaScript URL: <http://www.learn.javascript.ru> (дата звернення: 7.02.2019).
19. Open Server. URL: <https://ospanel.io> (дата звернення: 13.02.2019).
20. Форта Б. SQL за 10 минут. Москва: «Вильямс», 2014. 340 с.
21. Билл Карвин. Программирование баз данных SQL. Типичные ошибки и их устранение. Москва: Рид Групп 2012. 312с.
22. Хоган Б. «HTML5 и CSS3. Веб-по стандартам нового поколения». Санкт-Петербург: Питер, 2011. 272 с.
23. Довідник по HTML та CSS. URL: <http://htmlbook.ru> (дата звернення: 5.03.2019).
24. MySQL. Матеріал з habr. URL: <https://habr.com/ru/hub/mysql/> (дата звернення: 7.03.2019).
25. HTTP-cookie. Матеріал з Вікіпедії вільної енциклопедії. URL: <https://uk.wikipedia.org/wiki/cookie> (дата звернення: 15.03.2019).
26. JavaScript. Матеріал з habr. URL: <https://habr.com/ru/company/ruvds/blog/429552/> (дата звернення: 17.03.2019).
27. Кузнецов М. В., Симдянов И. Б. Самоучитель PHP 7. Санкт-Петербург: БХВ-Петербург, 2018. 448 с. (дата звернення: 26.03.2019).
28. Sublime Text. URL: <https://www.sublimetext.com> (дата звернення: 15.04.2019).
29. Courshanter coursehunters.net/course/online-kurs-javascript-hero (дата звернення: 6.05.2019).
30. Underscores. URL: underscores.me (дата звернення: 7.05.2019).

					ДР.Шс – 23.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		70

Додаток А

Файл gulpfile.js

```

var gulp          = require('gulp'),
    sass          = require('gulp-sass'),
    browserSync   = require('browser-sync'),
    concat        = require('gulp-concat'),
    uglify        = require('gulp-uglifyjs'),
    cssnano       = require('gulp-cssnano'),
    rename        = require('gulp-rename'),
    del           = require('del'),
    imagemin      = require('gulp-imagemin'),
    pngquant      = require('imagemin-pngquant'),
    cache         = require('gulp-cache'),
    autoprefixer  = require('gulp-autoprefixer');

gulp.task('sass', function() {
    return gulp.src('app/sass/**/*.scss')
        .pipe(sass())
        .pipe(autoprefixer(['last 15 versions', '> 1%', 'ie 8', 'ie 7'], {
cascade: true })))
        .pipe(gulp.dest('app/css'))
        .pipe(browserSync.reload({stream: true})))
});

gulp.task('browser-sync', function() {
    browserSync({
        server: {
            baseDir: 'app'
        },
    },
    notify: false
    });
});

gulp.task('scripts', function() {
    return gulp.src([
        'app/libs/jquery/dist/jquery.min.js',
        'app/libs/magnific-popup/dist/jquery.magnific-popup.min.js'
    ])
        .pipe(concat('libs.min.js'))
        .pipe(uglify())
        .pipe(gulp.dest('app/js'));
});

```

```

gulp.task('css-libs', ['sass'], function() {
  return gulp.src('app/css/libs.css')
    .pipe(cssnano())
    .pipe(rename({suffix: '.min'}))
    .pipe(gulp.dest('app/css'));
});

gulp.task('watch', ['browser-sync', 'css-libs', 'scripts'], function() {
  gulp.watch('app/sass/**/*.scss', ['sass']);
  gulp.watch('app/*.html', browserSync.reload);
  gulp.watch('app/js/**/*.js', browserSync.reload);
});

gulp.task('clean', function() {
  return del.sync('dist');
});

gulp.task('img', function() {
  return gulp.src('app/img/**/*')
    .pipe(cache(imagemin({
      interlaced: true,
      progressive: true,
      svgoPlugins: [{removeViewBox: false}],
      use: [pngquant()]
    })))
    .pipe(gulp.dest('dist/img'));
});

gulp.task('build', ['clean', 'img', 'sass', 'scripts'], function() {
  var buildCss = gulp.src([
    'app/css/*.css',
    'app/css/libs.min.css'
  ])
    .pipe(gulp.dest('dist/css'))
  var buildFonts = gulp.src('app/fonts/**/*')
    .pipe(gulp.dest('dist/fonts'))

  var buildJs = gulp.src('app/js/**/*')
    .pipe(gulp.dest('dist/js'))
  var buildHtml = gulp.src('app/*.html')
    .pipe(gulp.dest('dist'));
});

gulp.task('clear', function () {
  return cache.clearAll();
});

gulp.task('default', ['watch']);

```


Файл index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
  <link rel="stylesheet" href="css/style.css">
</head>
<body>
<header class="header">
  <ul class="nav">
    <li><a href="gallery.html" class="item">Галерея</a></li>
    <li><a href="index.html" class="item">Турнірна сітка</a></li>
    <li><a href="news.html" class="item">Новини</a></li>
  </ul>
</header>
<main class="content">
  <h2 class="content_title">Команди</h2>
  <div class="teams">
    <div class="team-name" id="team-1">Команда 1</div>
    <div class="team-name" id="team-2">Команда 2</div>
    <div class="team-name" id="team-3">Команда 3</div>
    <div class="team-name" id="team-4">Команда 4</div>
    <div class="team-name" id="team-5">Команда 5</div>
    <div class="team-name" id="team-6">Команда 6</div>
    <div class="team-name" id="team-7">Команда 7</div>
    <div class="team-name" id="team-8">Команда 8</div>
  </div>
  <div class="teams">
    <div class="team-name" id="team-9">Команда 9</div>
    <div class="team-name" id="team-10">Команда 10</div>
    <div class="team-name" id="team-11">Команда 11</div>
    <div class="team-name" id="team-12">Команда 12</div>
    <div class="team-name" id="team-13">Команда 13</div>
    <div class="team-name" id="team-14">Команда 14</div>
    <div class="team-name" id="team-15">Команда 15</div>
    <div class="team-name" id="team-16">Команда 16</div>
  </div>

  <div class="tournament-grid">

    <div class="first-round">
```

```
<a href="article.html" class="game">
<span class="team" id="first-round-team1">Команда 1</span>
<span class="result" id="first-round-result1"></span>
<span class="result" id="first-round-result2"></span>
<span class="team" id="first-round-team2"></span>
</a>
<a href="article.html" class="game">
<span class="team" id="first-round-team3">Команда 3</span>
<span class="result" id="first-round-result3">5</span>
<span class="result" id="first-round-result4">7</span>
<span class="team" id="first-round-team4">Команда 4</span>
</a>
<a href="article.html" class="game">
<span class="team" id="first-round-team5">Команда 5</span>
<span class="result" id="first-round-result5">4</span>
<span class="result" id="first-round-result6">8</span>
<span class="team" id="first-round-team6">Команда 6</span>
</a>
<a href="article.html" class="game">
<span class="team" id="first-round-team7">Команда 7</span>
<span class="result" id="first-round-result7">1</span>
<span class="result" id="first-round-result8">2</span>
<span class="team" id="first-round-team8">Команда 8</span>
</a>
<a href="article.html" class="game">
<span class="team" id="first-round-team9"></span>
<span class="result" id="first-round-result9"></span>
<span class="result" id="first-round-result10"></span>
<span class="team" id="first-round-team10">Команда 10</span>
</a>
<a href="article.html" class="game">
<span class="team" id="first-round-team11">Команда 11</span>
<span class="result" id="first-round-result11">1</span>
<span class="result" id="first-round-result12">3</span>
<span class="team" id="first-round-team12">Команда 12</span>
</a>
<a href="article.html" class="game">
<span class="team" id="first-round-team13">Команда 13</span>
<span class="result" id="first-round-result13">2</span>
<span class="result" id="first-round-result14">5</span>
<span class="team" id="first-round-team14">Команда 14</span>
</a>
<a href="article.html" class="game">
```

```

<span class="team" id="first-round-team15">Команда 15</span>
<span class="result" id="first-round-result15">4</span>
<span class="result" id="first-round-result16">0</span>
<span class="team" id="first-round-team16">Команда 16</span>
</a>
</div>

```

```

<div class="quarterfinal">
  <a href="article.html" class="game">
    <span class="team" id="quarter-team1"></span>
    <span class="result" id="quarter-result1">2</span>
    <span class="result" id="quarter-result2">3</span>
    <span class="team" id="quarter-team2"></span>
  </a>
  <a href="article.html" class="game">
    <span class="team" id="quarter-team3"></span>
    <span class="result" id="quarter-result3">1</span>
    <span class="result" id="quarter-result4">4</span>
    <span class="team" id="quarter-team4"></span>
  </a>
  <a href="article.html" class="game">
    <span class="team" id="quarter-team5"></span>
    <span class="result" id="quarter-result5">2</span>
    <span class="result" id="quarter-result6">0</span>
    <span class="team" id="quarter-team6"></span>
  </a>
  <a href="article.html" class="game">
    <span class="team" id="quarter-team7"></span>
    <span class="result" id="quarter-result7">0</span>
    <span class="result" id="quarter-result8">3</span>
    <span class="team" id="quarter-team8"></span>
  </a>
</div>

```

```

<div class="semi-final">
  <a href="article.html" class="game">
    <span class="team" id="semi-team-1"></span>
    <span class="result" id="semi-result-1">1</span>
    <span class="result" id="semi-result-2">3</span>
    <span class="team" id="semi-team-2"></span>
  </a>
  <a href="article.html" class="game">
    <span class="team" id="semi-team-3"></span>

```

```

        <span class="result" id="semi-result-3">2</span>
        <span class="result" id="semi-result-4">1</span>
        <span class="team" id="semi-team-4"></span>
    </a>
</div>
<div class="final">
    <a href="article.html" class="game">
        <span class="team" id="final-team-1"></span>
        <span class="result" id="final-result-1"></span>
        <span class="result" id="final-result-2"></span>
        <span class="team" id="final-team-2"></span>
    </a>
</div>
</div>
</main>

<footer class="footer"></footer>

<script src="js/script.js"></script>
</body>
</html>

```

Файл gallery.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
    <link rel="stylesheet" href="css/lightgallery.css">
    <link rel="stylesheet" href="css/style.css">
    <link rel="stylesheet" href="css/gallery.css">
</head>
<body>
    <header class="header">
        <ul class="nav">
            <li><a href="gallery.html" class="item">Галерея</a></li>
            <li><a href="index.html" class="item">Турнірна
сітка</a></li>
            <li><a href="news.html" class="item">Новини</a></li>
        </ul>
    </header>

```

```

<main>
  <section class="section">
    <h2 class="section__title">1/8 фіналу</h2>
    <div class="section__content lightgallery">
      <a href="img/4.jpg">
        
      </a>
      <a href="img/4.jpg">
        
      </a>
      <a href="img/4.jpg">
        
      </a>
      <a href="img/4.jpg">
        
      </a>
      <a href="img/4.jpg">
        
      </a>
      <a href="img/4.jpg">
        
      </a>
      <a href="img/4.jpg">
        
      </a>
      <a href="img/4.jpg">
        
      </a>
      <a href="img/4.jpg">
        
      </a>
      <a href="img/4.jpg">
        
      </a>
    </div>
  </section>
  <section class="section">
    <h2 class="section__title">чвертьфінал</h2>
    <div class="section__content lightgallery1">
      <a href="img/3.jpg">
        
      </a>
      <a href="img/3.jpg">
        
      </a>
      <a href="img/3.jpg">

```



```

        <a href="img/2.jpg">
            
        </a>
        <a href="img/2.jpg">
            
        </a>
        <a href="img/2.jpg">
            
        </a>
    </div>
</section>
</main>

<footer class="footer"></footer>

<script src="js/lightgallery.min.js"></script>
<script>
    lightGallery(document.querySelector('.lightgallery'));
    lightGallery(document.querySelector('.lightgallery1'));
    lightGallery(document.querySelector('.lightgallery2'));
    lightGallery(document.querySelector('.lightgallery3'));
</script>
</body>
</html>

```

Файл news.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
    <link rel="stylesheet" href="css/style.css">
    <link rel="stylesheet" href="css/news.css">
</head>
<body>
    <header class="header">
        <ul class="nav">
            <li><a href="gallery.html" class="item">Галерея</a></li>
            <li><a href="index.html" class="item">Турнірна
сітка</a></li>
            <li><a href="news.html" class="item">Новини</a></li>
        </ul>

```

```

</header>

<main>
  <section class="section">
    <div class="section__title-block"><a href="article.html"
class="title">Фінал. Прогнози експертів</a></div>
    <div class="section__text-block">
      <p class="text">Lorem ipsum dolor sit amet, consectetur
adipisicing elit. Laboriosam sed possimus, temporibus accusantium ad!
Laboriosam consequatur aliquid cupiditate, earum minima ratione
reprehenderit. Odit dolorem impedit voluptate placeat debitis ex
deleniti...</p>
    </div>
  </section>
  <section class="section">
    <div class="section__title-block"><a href="article.html"
class="title">Команда 8 легко перемагає суперника</a></div>
    <div class="section__text-block">
      <p class="text">Lorem ipsum dolor sit amet, consectetur
adipisicing elit. Ex, harum, dignissimos? Quis iusto ducimus repellendus
maxime ex totam similique, necessitatibus quisquam in ad commodi tempora
hic neque, impedit iure beatae...</p>
    </div>
  </section>
  <section class="section">
    <div class="section__title-block"><a href="article.html"
class="title">Команді 9 пощастило вийти у фінал</a></div>
    <div class="section__text-block">
      <p class="text">Lorem ipsum dolor sit amet, consectetur
adipisicing elit. Temporibus error quod necessitatibus dicta, dolore est
blanditiis similique dolorum inventore corporis nesciunt, commodi minima
recusandae quam laudantium in voluptas sunt? Dolorum...</p>
    </div>
  </section>
  <section class="section">
    <div class="section__title-block"><a href="article.html"
class="title">Фінал. Прогнози експертів</a></div>
    <div class="section__text-block">
      <p class="text">Lorem ipsum dolor sit amet, consectetur
adipisicing elit. Laboriosam sed possimus, temporibus accusantium ad!
Laboriosam consequatur aliquid cupiditate, earum minima ratione
reprehenderit. Odit dolorem impedit voluptate placeat debitis ex
deleniti...</p>

```



```

        </div>
    </section>
    <section class="section">
        <div class="section__title-block"><a href="article.html"
class="title">Команда 8 легко перемагає суперника</a></div>
        <div class="section__text-block">
            <p class="text">Lorem ipsum dolor sit amet, consectetur
adipisicing elit. Ex, harum, dignissimos? Quis iusto ducimus repellendus
maxime ex totam similique, necessitatibus quisquam in ad commodi tempora
hic neque, impedit iure beatae...</p>
        </div>
    </section>
    <section class="section">
        <div class="section__title-block"><a href="article.html"
class="title">Команді 9 пощастило вийти у фінал</a></div>
        <div class="section__text-block">
            <p class="text">Lorem ipsum dolor sit amet, consectetur
adipisicing elit. Temporibus error quod necessitatibus dicta, dolore est
blanditiis similique dolorum inventore corporis nesciunt, commodi minima
recusandae quam laudantium in voluptas sunt? Dolorum...</p>
        </div>
    </section>
</main>

<footer class="footer"></footer>

<script src="js/lightgallery.min.js"></script>
<!-- lightgallery plugins -->
<!-- <script src="js/lg-thumbnail.min.js"></script> -->
<!-- <script src="js/lg-fullscreen.min.js"></script> -->
<script>
    lightGallery(document.getElementById('lightgallery'));
</script>
</body>
</html>

```

Файл article.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
    <link rel="stylesheet" href="css/style.css">

```

```

    <link rel="stylesheet" href="css/article.css">
</head>
<body>
    <header class="header">
        <ul class="nav">
            <li><a href="gallery.html" class="item">Галерея</a></li>
            <li><a href="index.html" class="item">Турнірна
сірка</a></li>
            <li><a href="news.html" class="item">Новини</a></li>
        </ul>
    </header>

    <main>
        <section class="section">
            <div class="section__title-block"><h2 href="article.html"
class="title">Фінал. Прогнози експертів</h2></div>
            <article class="section__text-block">
                <p class="text">Lorem ipsum dolor sit amet, consectetur
adipisicing elit. Temporibus error quod necessitatibus dicta, dolore est
blanditiis similique dolorum inventore corporis nesciunt, commodi minima
recusandae quam laudantium in voluptas sunt?</p>

                <p class="text">Lorem ipsum dolor sit amet, consectetur
adipisicing elit. Temporibus error quod necessitatibus dicta, dolore est
blanditiis similique dolorum inventore corporis nesciunt, commodi minima
recusandae quam laudantium in voluptas sunt? Dolorum. Sit amet,
consectetur adipisicing elit. Incidunt expedita quod recusandae, numquam
laboriosam cum maxime exercitationem et eligendi, laborum consequuntur
optio, dolores! Accusamus officiaidunt dolorem dolores voluptate
molestias!

                Lorem ipsum dolor sit amet, consectetur adipisicing
elit. Illum sint architecto, culpa asperiores dignissimos ducimus amet
molestias minus accusamus, dicta voluptatibus eum officia, quod. Vel,
neque modi nulla harum voluptatum.</p>

                <p class="text">Lorem ipsum dolor sit amet, consectetur
adipisicing elit. Temporibus error quod necessitatibus dicta, dolore est
blanditiis similique dolorum inventore corporis nesciunt. Lorem ipsum
dolor sit amet, consectetur adipisicing elit. Ex, harum, dignissimos? Quis
iusto ducimus repellendus maxime ex totam similique, necessitatibus
quisquam in ad commodi tempora hic neque, impedit iure beatae. Commodi
minima recusandae quam laudantium in voluptas sunt? Dolorum culpa
asperiores dignissimos ducimus amet molestias minus accusamus, dicta
voluptatibus eum officia, quod. Vel, neque modi nulla harum voluptatum?
Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ex, harum,

```

dignissimos? Quis iusto ducimus repellendus maxime ex totam similique, necessitatibus quisquam in ad commodi tempora hic neque, impedit iure beatae.</p>

```
<p class="text">Lorem ipsum dolor sit amet, consectetur
adipisicing elit. Temporibus error quod necessitatibus dicta, dolore est
blanditiis similique dolorum inventore corporis nesciunt, commodi minima
recusandae quam laudantium in voluptas sunt? Dolorum.</p>
```

```
</article>
```

```
</section>
```

```
<a href="news.html" class="see-more">Інші новини...</a>
```

```
</main>
```

```
<footer class="footer"></footer>
```

```
</body>
```

```
</html>
```

Файл style.scss

```
body{
  margin: 0;
  padding: 0;
  background-color: #ffe;
}

.header{
  display: flex;
  align-items: center;
  justify-content: center;
  background-color: #878721;
  height: 140px;
  text-transform: uppercase;
  letter-spacing: 5px;
  margin-bottom: 30px;
  .nav{
    width: 60%;
    margin: 0 auto;
    display: flex;
    justify-content: space-between;
    list-style: none;
    li{
```

```

        .item{
            color: #ffe;
            transition: .3s;
            position: relative;
            text-transform: uppercase;
            text-decoration: none;
            font-size: 20px;
            transition: .5s;
            &:after{
                background-color: black;
                display: block;
                content: "";
                height: 1.5px;
                width: 0%;
                left: 50%;
                position: absolute;
                transition: width .5s;
                transform: translateX(-50%);

            }
            &:hover:after,&:focus:after {
                width: 100%;
            }
            &:hover{
                color: black;
            }
        }
    }
}

.content{
    width: 100%;
    display: flex;
    flex-direction: column;
    .content_title{
        width: 100%;
        margin: 20px 0 35px 0;
        font-size: 34px;
        font-weight: bold;
        text-align: center;
        color: #879D21;
        text-transform: uppercase;
    }
}

```

```
}  
.teams{  
  display: flex;  
  align-items: center;  
  justify-content: center;  
  .team-name{  
    border: 2px solid #878721;  
    color: #878721;  
    border-radius: 15%;  
    padding: 10px;  
    margin: 10px;  
    font-size: 20px;  
  }  
}  
  
.tournament-grid{  
  margin-top: 80px;  
  margin: 80px auto;  
  width: 80%;  
  display: flex;  
  align-items: center;  
  justify-content: space-between;  
  .game{  
    cursor: pointer;  
    text-decoration: none;  
    display: flex;  
    align-items: center;  
    border: 2px solid #878721;  
    color: #878721;  
    margin-bottom: 15px;  
    height: 30px;  
    .bold{  
      font-weight: bold;  
    }  
    .team{  
      height: 20px;  
      margin: 10px 0 10px 10px;  
      width: 90px;  
    }  
    .result{  
      display: flex;  
      justify-content: center;  
      align-items: center;  
      height: 100%;  
    }  
  }  
}
```

```

        width: 20px;
        border-right: solid 1px #fdd;
        &:nth-child(even){
            border-left: solid 1px #fdd;
        }
    }
}
}

.footer{
    position: fixed;
    bottom: 0;
    width: 100%;
    height: 4px;
    background-color: #878721;
}

```

Файл gallery.scss

```

main{
    display: flex;
    flex-direction: column;
    align-items: center;
    .section{
        width: 80%;
        &__title{
            font-size: 34px;
            text-transform: uppercase;
            text-align: center;
            color: #879D21;
            margin-bottom: 50px;
        }
        &__content{
            display: flex;
            justify-content: space-between;
            flex-wrap: wrap;
            width: 100%;
            border: 1px solid #fee;
            padding: 10px;
            padding-bottom: 0;
            box-sizing: border-box;
            a{

```

```
width: 32%;
.photo{
    width: 100%;
    margin-bottom: 20px;
    height: 250px;
    box-shadow: 0px 5px 8px rgba(0,0,0,0.5);
    transition: box-shadow .5s, transform .5s;
    &:hover {
        box-shadow: 0px 0px 0px rgba(0,0,0,0.5);
        transform: scale(1.1);
    }
}
}
}
}
```

Файл news.scss

```
main{
    display: flex;
    align-items: center;
    flex-direction: column;
    justify-content: center;
    margin: 50px auto;
    width: 60%;
    .section{
        margin-bottom: 40px;
        border: 1px solid #fee;
        &__title-block{
            text-align: center;
            border-bottom: 1px solid #fee;
            padding: 20px 0;
            .title{
                font-size: 34px;
                text-transform: uppercase;
                text-decoration: none;
                text-align: center;
                color: #879D21;
                transition: .5s;
                &:hover{
                    color: #6B6B09;
                }
            }
        }
    }
}
```

```

    }
  }
  &__text-block{
    text-align: center;
    padding: 20px;
    .text{
      font-size: 20px;
      color: #6B6B09;
      font-style: italic;
    }
  }
}
}
}

```

Файл article.scss

```

main{
  display: flex;
  align-items: center;
  flex-direction: column;
  justify-content: center;
  margin: 50px auto;
  width: 60%;
  .section{
    margin-bottom: 40px;
    border: 1px solid #fee;
    &__title-block{
      text-align: center;
      border-bottom: 1px solid #fee;
      padding: 20px 0;
      .title{
        font-size: 34px;
        margin: 0;
        text-transform: uppercase;
        text-decoration: none;
        text-align: center;
        color: #879D21;
      }
    }
  }
  &__text-block{
    text-align: center;
    padding: 20px;
    .text{

```



```
let firstRoundResult6 = document.getElementById('first-round-result6');
let firstRoundTeam7 = document.getElementById('first-round-team7');
let firstRoundResult7 = document.getElementById('first-round-result7');
let firstRoundTeam8 = document.getElementById('first-round-team8');
let firstRoundResult8 = document.getElementById('first-round-result8');
let firstRoundTeam9 = document.getElementById('first-round-team9');
let firstRoundResult9 = document.getElementById('first-round-result9');
let firstRoundTeam10 = document.getElementById('first-round-team10');
let firstRoundResult10 = document.getElementById('first-round-result10');
let firstRoundTeam11 = document.getElementById('first-round-team11');
let firstRoundResult11 = document.getElementById('first-round-result11');
let firstRoundTeam12 = document.getElementById('first-round-team12');
let firstRoundResult12 = document.getElementById('first-round-result12');
let firstRoundTeam13 = document.getElementById('first-round-team13');
let firstRoundResult13 = document.getElementById('first-round-result13');
let firstRoundTeam14 = document.getElementById('first-round-team14');
let firstRoundResult14 = document.getElementById('first-round-result14');
let firstRoundTeam15 = document.getElementById('first-round-team15');
let firstRoundResult15 = document.getElementById('first-round-result15');
let firstRoundTeam16 = document.getElementById('first-round-team16');
let firstRoundResult16 = document.getElementById('first-round-result16');
```

```
//quarterfinal
```

```
let quarterTeam1 = document.getElementById('quarter-team1');
let quarterResult1 = document.getElementById('quarter-result1');
let quarterTeam2 = document.getElementById('quarter-team2');
let quarterResult2 = document.getElementById('quarter-result2');
let quarterTeam3 = document.getElementById('quarter-team3');
let quarterResult3 = document.getElementById('quarter-result3');
let quarterTeam4 = document.getElementById('quarter-team4');
let quarterResult4 = document.getElementById('quarter-result4');
let quarterTeam5 = document.getElementById('quarter-team5');
let quarterResult5 = document.getElementById('quarter-result5');
let quarterTeam6 = document.getElementById('quarter-team6');
let quarterResult6 = document.getElementById('quarter-result6');
let quarterTeam7 = document.getElementById('quarter-team7');
let quarterResult7 = document.getElementById('quarter-result7');
let quarterTeam8 = document.getElementById('quarter-team8');
let quarterResult8 = document.getElementById('quarter-result8');
```

```
//semifinal
```

```

let semiTeam1 = document.getElementById('semi-team-1');
let semiResult1 = document.getElementById('semi-result-1');
let semiTeam2 = document.getElementById('semi-team-2');
let semiResult2 = document.getElementById('semi-result-2');
let semiTeam3 = document.getElementById('semi-team-3');
let semiResult3 = document.getElementById('semi-result-3');
let semiTeam4 = document.getElementById('semi-team-4');
let semiResult4 = document.getElementById('semi-result-4');

//final
let finalTeam1 = document.getElementById('final-team-1');
let finalResult1 = document.getElementById('final-result-1');
let finalTeam2 = document.getElementById('final-team-2');
let finalResult2 = document.getElementById('final-result-2');

/// >>> ADD FONT-WEIGHT:BOLD TO WINNERS AND PUSH THEM TO NEXT ROUND <<<
///

// first round
if      (+firstRoundResult1.textContent>+firstRoundResult2.textContent      ||
firstRoundTeam2.textContent === '') {
    firstRoundTeam1.classList.add("bold");
    quarterTeam1.innerHTML = firstRoundTeam1.textContent;
}
else if (+firstRoundResult1.textContent<+firstRoundResult2.textContent      ||
firstRoundTeam1.textContent === '') {
    firstRoundTeam2.classList.add("bold");
    quarterTeam1.innerHTML = firstRoundTeam2.textContent;
}
if      (+firstRoundResult3.textContent>+firstRoundResult4.textContent      ||
firstRoundTeam4.textContent === '') {
    firstRoundTeam3.classList.add("bold");
    quarterTeam2.innerHTML = firstRoundTeam3.textContent;
}
else if (+firstRoundResult3.textContent<+firstRoundResult4.textContent      ||
firstRoundTeam3.textContent === '') {
    firstRoundTeam4.classList.add("bold");
    quarterTeam2.innerHTML = firstRoundTeam4.textContent;
}
if      (+firstRoundResult5.textContent>+firstRoundResult6.textContent      ||
firstRoundTeam6.textContent === '') {
    firstRoundTeam5.classList.add("bold");

```

```

    quarterTeam3.innerHTML = firstRoundTeam5.textContent;
}
else if (+firstRoundResult5.textContent<+firstRoundResult6.textContent ||
firstRoundTeam5.textContent === '') {
    firstRoundTeam6.classList.add("bold");
    quarterTeam3.innerHTML = firstRoundTeam6.textContent;
}
if (+firstRoundResult7.textContent>+firstRoundResult8.textContent ||
firstRoundTeam8.textContent === '') {
    firstRoundTeam7.classList.add("bold");
    quarterTeam4.innerHTML = firstRoundTeam7.textContent;
}
else if (+firstRoundResult7.textContent<+firstRoundResult8.textContent ||
firstRoundTeam7.textContent === '') {
    firstRoundTeam8.classList.add("bold");
    quarterTeam4.innerHTML = firstRoundTeam8.textContent;
}
if (+firstRoundResult9.textContent>+firstRoundResult10.textContent ||
firstRoundTeam10.textContent === '') {
    firstRoundTeam9.classList.add("bold");
    quarterTeam5.innerHTML = firstRoundTeam9.textContent;
}
else if (+firstRoundResult9.textContent<+firstRoundResult10.textContent ||
firstRoundTeam9.textContent === '') {
    firstRoundTeam10.classList.add("bold");
    quarterTeam5.innerHTML = firstRoundTeam10.textContent;
}
if (+firstRoundResult11.textContent>+firstRoundResult12.textContent ||
firstRoundTeam12.textContent === '') {
    firstRoundTeam11.classList.add("bold");
    quarterTeam6.innerHTML = firstRoundTeam11.textContent;
}
else if (+firstRoundResult11.textContent<+firstRoundResult12.textContent
|| firstRoundTeam11.textContent === '') {
    firstRoundTeam12.classList.add("bold");
    quarterTeam6.innerHTML = firstRoundTeam12.textContent;
}
if (+firstRoundResult13.textContent>+firstRoundResult14.textContent ||
firstRoundTeam14.textContent === '') {
    firstRoundTeam13.classList.add("bold");
    quarterTeam7.innerHTML = firstRoundTeam13.textContent;
}
}

```

```

else if (+firstRoundResult13.textContent<+firstRoundResult14.textContent
|| firstRoundTeam13.textContent === '') {
    firstRoundTeam14.classList.add("bold");
    quarterTeam7.innerHTML = firstRoundTeam14.textContent;
}
if (+firstRoundResult15.textContent>+firstRoundResult16.textContent ||
firstRoundTeam16.textContent === '') {
    firstRoundTeam15.classList.add("bold");
    quarterTeam8.innerHTML = firstRoundTeam15.textContent;
}
else if (+firstRoundResult15.textContent<+firstRoundResult16.textContent
|| firstRoundTeam15.textContent === '') {
    firstRoundTeam16.classList.add("bold");
    quarterTeam8.innerHTML = firstRoundTeam16.textContent;
}

// quarter final
if (+quarterResult1.textContent>+quarterResult2.textContent ||
quarterTeam2.textContent === '') {
    quarterTeam1.classList.add("bold");
    semiTeam1.innerHTML = quarterTeam1.textContent;
}
else if (+quarterResult1.textContent<+quarterResult2.textContent ||
quarterTeam1.textContent === '') {
    quarterTeam2.classList.add("bold");
    semiTeam1.innerHTML = quarterTeam2.textContent;
}
if (+quarterResult3.textContent>+quarterResult4.textContent ||
quarterTeam4.textContent === '') {
    quarterTeam3.classList.add("bold");
    semiTeam2.innerHTML = quarterTeam3.textContent;
}
else if (+quarterResult3.textContent<+quarterResult4.textContent ||
quarterTeam3.textContent === '') {
    quarterTeam4.classList.add("bold");
    semiTeam2.innerHTML = quarterTeam4.textContent;
}
if (+quarterResult5.textContent>+quarterResult6.textContent ||
quarterTeam6.textContent === '') {
    quarterTeam5.classList.add("bold");
    semiTeam3.innerHTML = quarterTeam5.textContent;
}
}

```

```

else if (+quarterResult5.textContent<+quarterResult6.textContent ||
quarterTeam5.textContent === '') {
    quarterTeam6.classList.add("bold");
    semiTeam3.innerHTML = quarterTeam6.textContent;
}
if (+quarterResult7.textContent>+quarterResult8.textContent ||
quarterTeam8.textContent === '') {
    quarterTeam7.classList.add("bold");
    semiTeam4.innerHTML = quarterTeam7.textContent;
}
else if (+quarterResult7.textContent<+quarterResult8.textContent ||
quarterTeam7.textContent === '') {
    quarterTeam8.classList.add("bold");
    semiTeam4.innerHTML = quarterTeam8.textContent;
}

//semi final
if (+semiResult1.textContent>+semiResult2.textContent) {
    semiTeam1.classList.add("bold");
    finalTeam1.innerHTML = semiTeam1.textContent;
}
else if(+semiResult1.textContent<+semiResult2.textContent) {
    semiTeam2.classList.add("bold");
    finalTeam1.innerHTML = semiTeam2.textContent;
}
if (+semiResult3.textContent>+semiResult4.textContent) {
    semiTeam3.classList.add("bold");
    finalTeam2.innerHTML = semiTeam3.textContent;
}
else if (+semiResult3.textContent<+semiResult4.textContent) {
    semiTeam4.classList.add("bold");
    finalTeam2.innerHTML = semiTeam4.textContent;
}

//final
if (+finalResult1.textContent>+finalResult2.textContent) {
    finalTeam1.classList.add("bold");
}
else if (+finalResult1.textContent<+finalResult2.textContent) {
    finalTeam2.classList.add("bold");
}

```

Додаток Б

Файл index.php

```

<?php get_header(); ?>
<?php

$posts = get_posts( array(
    'numberposts' => 1,
    'category'     => '0',
    'post_type'    => 'post',
    'suppress_filters' => true,
) );

foreach( $posts as $post ){ setup_postdata($post);
    ?>
    <section class="section">
        <div class="section__title-block"><h2 class="title"><?php the_title()
?></h2></div>
        <article class="section__text-block">
            <?php the_content() ?>
        </article>
    </section>
    <?php
}
wp_reset_postdata();

?>
<?php get_footer(); ?>

```

Файл header.php

```

<!DOCTYPE html>
<!--[if lt IE 8 ]><html class="no-js ie ie7" lang="en"> <![endif]-->
<!--[if IE 8 ]><html class="no-js ie ie8" lang="en"> <![endif]-->
<!--[if (gte IE 8) || (IE)]><!--><html class="no-js" lang="en"> <!--
<![endif]-->

<head>
    <meta charset="UTF-8">
    <?php wp_head() ?>

```

```

</head>
<body>
  <header class="header">
    <?php wp_nav_menu( array(
      'theme_location' => 'top',
      'container'      => 'ul',
      'menu_class'     => 'nav',
      'menu_id'        => 'nav'
    ) ); ?>
  </header>

```

Файл footer.php

```

<footer class="footer"></footer>
<?php wp_footer(); ?>
<script>
  lightGallery(document.querySelector('.lightgallery'));
</script>
</body>
</html>

```

Файл 404.php

Error 404

Файл functions.php

```

<?php

add_action( 'wp_enqueue_scripts', 'style_theme' );
add_action( 'wp_footer', 'scripts_theme' );
add_action( 'after_setup_theme', 'menu' );

function menu() {
  register_nav_menu( 'top', 'header menu' );
  add_theme_support( 'title-tag' );
  add_theme_support( 'post-thumbnails', array('post') );
  add_image_size( 'post-thumb', 0, 250, false );
}

function style_theme() {
  wp_enqueue_style( 'style', get_stylesheet_uri() );
}

```



```

    wp_enqueue_style( 'lightgallery', get_template_directory_uri() .
'/assets/css/lightgallery.css' );
    wp_enqueue_style( 'main', get_template_directory_uri() .
'/assets/css/main.css' );
    wp_enqueue_style( 'article', get_template_directory_uri() .
'/assets/css/article.css' );
    wp_enqueue_style( 'gallery', get_template_directory_uri() .
'/assets/css/gallery.css' );
    wp_enqueue_style( 'news', get_template_directory_uri() .
'/assets/css/news.css' );
}

function scripts_theme() {
    wp_enqueue_script( 'jquery' );
    wp_enqueue_script( 'main', get_template_directory_uri() .
'/assets/js/main.js' );
    wp_enqueue_script( 'lightgallery.min.js', get_template_directory_uri() .
'/assets/js/lightgallery.min.js' );
}

add_shortcode( 'team', 'baztag_func' );
function baztag_func( $atts, $content ) {
    return "$content";
}

add_shortcode( 'result', 'result_func' );
function result_func( $atts, $content ) {
    return "$content";
}

```

Файл gallery.php

```

<?php
/*
Template Name: Gallery
*/
?>
<?php get_header(); ?>

<main>

<?php

```

```

$posts = get_posts( array(
    'numberposts' => 5,
    'category'     => '4',
    'post_type'    => 'post',
    'suppress_filters' => true, ) );

foreach( $posts as $post ){ setup_postdata($post);
    ?>

    <section class="section">
        <h2 class="section__title"><?php the_title() ?></h2>
        <div class="section__content lightgallery">
            <?php the_content() ?>
        </div>
    </section>

    <?php
}

wp_reset_postdata();

?>
</main>
<?php get_footer() ?>

```

Файл page.php

```

<?php get_header(); ?>

<main>

    <?php

        $posts = get_posts( array(
            'numberposts' => 5,
            'category'     => '3',
            'post_type'    => 'post',
            'suppress_filters' => true, // подавление работы фильтров
изменения SQL запроса
        ) );

        foreach( $posts as $post ){ setup_postdata($post);
            ?>

```

```

        <section class="section">
            <div class="section__title-block"><a href="<?php
the_permalink() ?>" class="title"><?php the_title() ?></a></div>
            <div class="section__text-block">
                <?php the_content() ?>
            </div>
        </section>

        <?php
    }

wp_reset_postdata();

?>
</main>
<?php get_footer() ?>

```

Файл single.php

```

<?php get_header(); ?>

<main>

    <?php

        $posts = get_posts( array(
            'numberposts' => 1,
            'category'     => '0',
            'post_type'    => 'post',
            'suppress_filters' => true, ) );

        foreach( $posts as $post ){ setup_postdata($post);
            ?>

            <section class="section">
                <div class="section__title-block"><h2 class="title"><?php
the_title() ?></h2></div>
                <article class="section__text-block">
                    <?php the_content() ?>
                </article>
            </section>

```

```
<a href="<?php the_permalink() ?>">Інші новини...</a>

    <?php
}

wp_reset_postdata();

?>

</main>
<?php get_footer() ?>
```

БІБЛІОГРАФІЧНА ДОВІДКА

Тема дипломної роботи: Розробка веб сервісу для формування розкладу занять.

Обсяг пояснювальної записки: 61 аркуш

Перелік графічних матеріалів:

Таблиць _____ 0 _____

рисуноків _____ 32 _____

додатків 2 на 30 аркушах.

Дата закінчення дипломного проекту: “10” червня 2019 р.

Студент – дипломник _____
(підпис) (розшифровка підпису)