

**ДЕРЖАВНИЙ ТОРГОВЕЛЬНО-ЕКОНОМІЧНИЙ  
УНІВЕРСИТЕТ**

**Кафедра комп'ютерних наук та інформаційних технологій**

**ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА**

на тему:

**«Розробка Web-сайту команди групи підтримки»**

Студентки 4 курсу, 9 групи,  
спеціальності  
122 «Комп'ютерні науки»

*підпис студента*

Романько Вікторії  
Володимирівни

Науковий керівник  
кандидат педагогічних наук,  
доцент

*підпис керівника*

Дивак Володимир  
Валерійович

Гарант освітньої програми  
кандидат технічних наук,  
доцент

*підпис керівника*

Демідов Павло  
Георгійович

Київ 2023

# Державний торговельно-економічний університет

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

Зав. кафедри \_\_\_\_\_ **Затверджую**  
**Пурський О.І.**  
«12» грудня 2022р.

## **Завдання** **на випускню кваліфікаційну роботу студентки**

**Романько Вікторії Володимирівни**  
(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної роботи

«Розробка Web-сайту команди групи підтримки»

Затверджена наказом ректора від «09» грудня 2022 р. № 3332

2. Строк здачі студентом закінченої роботи 30 травня 2023 року

3. Цільова установка та вихідні дані до роботи

Мета роботи: розробка функціонального, зручного та інформативного Web-сайту для чирлідінг-команди.

Об'єкт дослідження: розробка Web-сайту команди групи підтримки, зосередженої на чирлідінгу.

Предмет дослідження: методи та інструменти, що використовуються для розробки Web-сайту команди групи підтримки чирлідінгу.

4. Перелік графічного матеріалу \_\_\_\_\_



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

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

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

### ВСТУП

## РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТА ДОСЛІДЖЕННЯ І ОПИС ЙОГО ОСНОВНИХ ПАРАМЕТРІВ І ХАРАКТЕРИСТИК

1.1. Тенденції та традиції створення та просування сайтів в Україні

1.2. Особливості розробки та впровадження Web-сайту для групи підтримки

1.3. Технічне завдання

1.4. Висновки до розділу 1

## РОЗДІЛ 2. ПРОЄКТУВАННЯ WEB-САЙТУ КОМАНДИ ГРУПИ ПІДТРИМКИ

2.1. Опис моделей Web-сайту команди групи підтримки

2.2. Огляд наявних інструментів розробки Web-сайту

2.3. Висновки до розділу 2

## РОЗДІЛ 3. РОЗРОБКА ТА ВПРОВАДЖЕННЯ WEB-САЙТУ КОМАНДИ ГРУПИ ПІДТРИМКИ (ЧИРЛІДИНГУ)

3.1. Функціональні характеристики розробленого Web-сайту

3.2. Опис архітектури розробленого Web-сайту

3.3. Висновки до розділу 3

### ВИСНОВКИ

### СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

7. Календарний план виконання роботи

№ Пор.	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		За планом	фактично
1	2	3	4
1	<i>Вибір теми випускної кваліфікаційної роботи</i>	04.10.2022	04.10.2022
2	<i>Розробка та затвердження завдання на випускну кваліфікаційну роботу</i>	15.12.2022	15.12.2022
3	<i>Вступ</i>	03.02.2023	03.02.2023
4	<i>РОЗДІЛ 1. Аналіз предмета дослідження і опис його основних параметрів і характеристик</i>	28.02.2023	28.02.2023
5	<i>РОЗДІЛ 2. Проектування Web-сайту команди групи підтримки</i>	06.04.2023	06.04.2023
6	<i>РОЗДІЛ 3. Розробка та впровадження Web-сайту команди групи підтримки (чирлідінгу)</i>	12.05.2023	12.05.2023
7	<i>Висновки</i>	15.05.2023	15.05.2023
8	<i>Здача випускної кваліфікаційної роботи на кафедрі науковому керівнику</i>	30.05.2023	30.05.2023
9	<i>Попередній захист випускної кваліфікаційної роботи</i>	31.05.2023 01.06.2023	31.05.2023 01.06.2023
11	<i>Виправлення зауважень, зовнішнє рецензування випускної кваліфікаційної роботи</i>	02.06.2023	02.06.2023
12	<i>Представлення готової зшитої випускної кваліфікаційної роботи на кафедру</i>	05.06.2023	05.06.2023
13	<i>Публічний захист випускної кваліфікаційної роботи</i>	За розкладом роботи ЕК	

8. Дата видачі завдання «15» грудня 2022 р.  
Керівник випускної кваліфікаційної роботи

Гарант освітньої програми

Дивак В.В.

(прізвище, ініціали, підпис)

Демідов П.Г.

(прізвище, ініціали, підпис)





### Анотація

У випускній кваліфікаційній роботі здійснено розробку та реалізацію Web-сайту команди групи підтримки. Було створено зручну у використанні та естетичну за інтерфейсом платформу, на якій доступна вся інформація про чирлідінг. У розробці було використано мову програмування C++, Java, HTML, CSS. Результатом роботи є функціональний, зрозумілий сайт, за допомогою якого популяризується даний вид спорту та можна зареєструватись на заняття.

**Ключові слова:** розробка, реалізація Web-сайту, функціональний, зрозумілий сайт.

### Anotation

In the final qualification work, the development and implementation of a website for a support group was carried out. An easy-to-use and ethetic interface platform was created, on which all information about cheerleading is available. The development used programming language C ++, Java, HTML, CSS. The result of the work is a functional, understandable site, through which this sport is popularized and you can register for classes.

**Keywords:** development, implementation of the website, functional, understandable site.



## ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТА ДОСЛІДЖЕННЯ І ОПИС ЙОГО ОСНОВНИХ ПАРАМЕТРІВ І ХАРАКТЕРИСТИК .....	10
1.1. Тенденції та традиції створення та просування сайтів в Україні .10	
1.2. Особливості розробки та впровадження Web-сайту команди групи підтримки .....	13
1.3. Технічне завдання .....	15
1.4. Висновки до розділу 1.....	24
РОЗДІЛ 2. ПРОЄКТУВАННЯ WEB-САЙТУ КОМАНДИ ГРУПИ ПІДТРИМКИ.....	27
2.1. Опис моделей Web-сайту для групи підтримки .....	27
2.2. Огляд наявних інструментів розробки Web-сайту .....	30
2.3. Висновки до розділу 2.....	36
РОЗДІЛ 3. РОЗРОБКА ТА ВПРОВАДЖЕННЯ WEB-САЙТУ КОМАНДИ ГРУПИ ПІДТРИМКИ (ЧИРЛІДИНГУ) .....	39
3.1. Функціональні характеристики розробленого Web-сайту .....	39
3.2. Опис архітектури розробленого Web-сайту.....	48
3.3. Висновки до розділу 3.....	53
ВИСНОВКИ .....	55
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	60

## ВСТУП

У сучасному світі зростає значення інформаційних технологій і їх вплив на різні сфери нашого життя. Інтернет-ресурси стають необхідними для ефективної комунікації та поширення інформації. Особливо важливою є наявність сайтів для різних груп підтримки, таких як чирлідінг-команди, оскільки вони сприяють формуванню спільноти, обміну досвідом та організації діяльності.

Актуальність розробки веб-сайту команди групи підтримки, зокрема у чирлідінгу, полягає в тому, що це надасть змогу залучити більше людей до цього виду спорту, збільшити обіг інформації між учасниками, забезпечити зручний доступ до необхідної інформації та покращити організацію тренувань і виступів.

Практичне значення даної роботи полягає в розробці функціонального і естетичного веб-сайту, що відповідає потребам чирлідінг-команди. Цей сайт буде включати інформацію про команду, розклад тренувань, результати виступів, фото- та відеоматеріали, контактні дані та інші розділи, необхідні для комунікації та взаємодії учасників.

Метою даної роботи є розробка функціонального, зручного та інформативного Web-сайту для чирлідінг-команди. Цей сайт буде відповідати сучасним вимогам веб-розробки та забезпечувати зручний доступ до інформації для всіх учасників команди.

Об'єктом дослідження є розробка Web-сайту команди групи підтримки, зосередженої на чирлідінгу. Розглядаються основні проблеми та вимоги, що стоять перед розробниками такого сайту, а також специфіка комунікації та взаємодії між учасниками команди.



Предметом дослідження є методи та інструменти, що використовуються для розробки Web-сайту команди групи підтримки чирлідінгу. Розглядаються різні аспекти веб-дизайну, структуру та функціональні можливості сайту, методи забезпечення зручності використання та безпеки інформації.

Методи, які використовуються в даній роботі, включають аналіз вимог користувачів, проектування структури та дизайну веб-сайту, використання сучасних веб-технологій для реалізації функціональності та тестування отриманого результату.

Ця випускна кваліфікаційна робота спрямована на розробку Web-сайту команди групи підтримки, зосередженої на чирлідінгу, з метою поліпшення комунікації та взаємодії учасників команди, підвищення обігу інформації та покращення організації тренувань і виступів. Результати дослідження та розробки веб-сайту можуть бути корисними для будь-якої чирлідінг-команди, яка прагне покращити свою діяльність та забезпечити зручний доступ до інформації для своїх учасників.

## РОЗДІЛ 1.

### АНАЛІЗ ПРЕДМЕТА ДОСЛІДЖЕННЯ І ОПИС ЙОГО ОСНОВНИХ ПАРАМЕТРІВ І ХАРАКТЕРИСТИК

#### 1.1. Тенденції та традиції створення та просування сайтів в Україні

В умовах військових дій в Україні, та пов'язаних з нею девальвацією гривні, зростанням цін на продукти, і зниження купівельної спроможності громадян, будь-яке підприємства, яке хоче вижити і розвиватися, повинно використовувати інноваційні методи залучення покупців і реалізації продукції.

Також, у зв'язку з бойовими діями на сході України, багато підприємців не мають можливості здійснювати торговельну діяльність у своїх містах. У цих умовах великого значення набуває торгівля через глобальну мережу Інтернет - e-commerce, як один з видів організаційних інновацій. Однак на даний момент торгівля через Інтернет магазини, аукціони та сайти оголошень в Україні не має таких масштабів як, наприклад, у Китаї, Індонезії, США. За даними українська асоціація директ маркетингу в 2013 році обсяг ринку e-commerce в Україні склав близько \$ 2 млрд., тобто 2% від усіх торгових операцій. З них найвищий відсоток спостерігався в сегменті «Електроніка та побутова техніка» і склав \$ 1 млрд., тобто 50% всього ринку e-commerce. У той же час, за даними видання Daily Mail в Британії близько 95% населення робили он-лайн покупки, і оборот склав £ 91 млрд [17]. Однією з причин не великих обсягів продажу через Інтернет в Україні є низький рівень його проникнення.



Зокрема, в Україні на початок 2014 року, за даними консалтингової групи Genius, близько 55% жителів України користувалося Інтернетом, в той час як в країнах Північної Америки та Західної Європи цей показник сягав 85%. Станом на 2021 рік, ринок електронної комерції в Україні вже досягнув 4млрд. дол. [25]. Іншою причиною є незнання і невміння працювати з Інтернет інструментами. Традиційно, створення сайту включає комплекс заходів з веб-дизайну, верстці сторінки, програмування, а також налаштування веб-сервера. Однак, для ефективної роботи цього не достатньо. Створення та розміщення сайту в мережі Інтернет дає вихід на глобальний ринок і, відповідно, велику конкуренцію. У цьому випадку підприємцю необхідно використовувати інноваційні методи просування для залучення клієнтів і формування попиту на свій товар. Тому дослідження в напрямку розробки системної моделі просування сайту є актуальними.

Наявність успішного сайту дозволить підприємству представити продукцію та послуги широкому колу користувачів, а також вийти на міжнародний ринок. Під успішним сайтом ми розуміємо сайт, який за ключовими запитом в пошукових системах Google знаходиться на 1-3 місці. За даними джерела «SEO: Пошукова Оптимізація від А до Я», перші три місця привертають увагу 60-70% людей, близько 30% переходить на сайти, контекстної реклами, і всього кілька відсотків припадає на інші сайти. Таким чином, якщо веб-сайт компанії створено недавно, він не відповідає вимогам пошукових систем і підприємство не користується послугами контекстної реклами, пошукова система залишить його на останніх місцях сайт не дасть ніякого ефекту і споживач не дізнається про цю фірму.

Пошукову оптимізацію можна розділити на три етапи: аудит, внутрішня і зовнішня оптимізація.

Інший метод - просування в соціальних мережах. В останні роки, соціальні мережі стають найбільш відвідуваними сайтами в Інтернеті. За даними досліджень компанії Ebizmba, до трійки найбільш відвідуваних соціальних мереж увійшли Facebook - 2000 мільйонів користувачів на рік, Twitter - 310 мільйонів користувачів на рік і LinkedIn - 255 млн користувачів. Наведені дані свідчать про необхідність пошуку потенційних клієнтів в соціальних мережах. Для цього передбачені різні методи. Так, у Facebook найбільш ефективний метод реклами є створення групи, присвяченій даній компанії. Тим не менш, групі, як і сайту, на початкових етапах складно отримати багато відвідувачів. Група з великою кількістю активних підписників дозволить залучити нових відвідувачів на сайт компанії, інформувати аудиторію про нові товари і послуги, а також дасть можливість дізнаватися та контролювати відгуки про компанії, формуючи громадську думку.

Цільові сторінки, завдяки наявності трьох і більше точок дотику (реєстраційна форма, анкета та інше), дозволяють збирати і використовувати дані потенційних клієнтів для продажу товарів і послуг. Однак даними метод є ефективним тільки при використанні з контекстною рекламою, просуванням в соціальних мережах і SEO. Ефективність цих методів вище при їх комплексній реалізації.

Зростання використання мобільних пристроїв українськими користувачами спонукає до мобільної адаптації сайтів. Розробники та дизайнери активно працюють над створенням мобільно-дружніх інтерфейсів, щоб забезпечити зручний досвід для користувачів, які відвідують сайти зі своїх смартфонів і планшетів.



Трендом в Україні є мінімалістичний дизайн, з фокусом на простоті, чистоті та мінімальному використанні елементів. Це допомагає зосередитися на контенті та полегшує сприйняття інформації для користувачів. Українські сайти все частіше використовують елементи інтерактивності, такі як анімація, графічні ефекти, прокрутка, відео та звукові компоненти. Це дозволяє залучити користувачів та забезпечити більш змістовний досвід взаємодії з сайтом.

Враховуючи популярність соціальних медіа, українські сайти активно використовують інтеграцію з платформами соціальних медіа. Це дозволяє користувачам легко ділитися контентом сайту у своїх соціальних мережах і розповідати про нього. Враховуючи українську мову та культуру, створення сайтів з українською локалізацією є важливим аспектом. Відповідна локалізація забезпечує комфортну сприйняття сайту українськими користувачами та забезпечує більшу привабливість сайту на місцевому ринку. Просування сайтів в пошукових системах (SEO) є важливим елементом веб-розробки в Україні. Розробники та маркетологи зосереджуються на оптимізації вмісту, ключових слів, структури сайту та зовнішніх посилань, щоб підвищити видимість сайту в пошукових системах.

## **1.2. Особливості розробки та впровадження Web-сайту команди групи підтримки**

Розробка і впровадження сайту для групи підтримки вимагає уваги до деяких особливостей, щоб забезпечити ефективність та зручність для користувачів.

Спочатку необхідно з'ясувати, які конкретні потреби група підтримки має і які цілі мають бути досягнуті з допомогою сайту. Наприклад, це може бути надання інформації про групу, взаємодія з учасниками, розміщення ресурсів та інше.

Важливо створити зручний та привабливий дизайн інтерфейсу, який допоможе користувачам швидко зорієнтуватися на сайті. Слід врахувати, що деякі учасники групи підтримки можуть мати обмеження щодо використання комп'ютера або мобільного пристрою, тому дизайн має бути адаптивним та доступним.

Необхідно обрати набір функціональності, який найкраще відповідатиме потребам групи підтримки. Це можуть бути форуми для обговорень, онлайн-чат, блог або новини, розділи з корисними ресурсами, можливість реєстрації та входу для учасників і т. д. Слід забезпечити можливість легкої навігації та пошуку інформації.

Особливу увагу необхідно звернути на захист особистих даних учасників групи підтримки. Має бути забезпечене безпечне зберігання та передача інформації, надійні методи аутентифікації та шифрування даних.

Необхідно створити можливості для взаємодії між учасниками групи підтримки. Це можуть бути коментарі, оцінки та відгуки, можливість обміну досвідом і порадами між користувачами. З урахуванням зростаючого використання мобільних пристроїв, сайт має бути мобільно сумісним. Це забезпечить доступність сайту для користувачів, які використовують смартфони або планшети.

Необхідно забезпечити веб-сайт механізмами аналітики, щоб отримувати інформацію про взаємодію користувачів з сайтом. Це дозволить



аналізувати ефективність сайту, виправляти проблеми та вдосконалювати його з часом. І сайт необхідно також постійно підтримувати та оновлювати.

Ці аспекти є загальними рекомендаціями, і конкретні особливості розробки та впровадження сайту для групи підтримки можуть залежати від унікальних потреб групи та доступних ресурсів.

### **1.3. Технічне завдання**

#### **1. Загальні відомості**

##### **1.1. Найменування системи**

###### **1.1.1. Повне найменування системи - Kolibri**

###### **1.1.2. Скорочене найменування системи - Kolibri**

##### **1.2. Планові терміни початку та закінчення робіт**

Початок робіт: 01.05.2023

Закінчення робіт: 01.06.2023

##### **1.3. Порядок оформлення і пред'явлення результатів робіт**

Розробка сайту повинна відбуватися відповідно до передбачених етапів: аналіз вимог, проектування, розробка, тестування, впровадження. На кожному етапі робіт слід підготувати відповідні результати, які повинні бути представлені замовнику для затвердження. Після затвердження кожного етапу робіт, виконавець забезпечує передачу результатів до наступного етапу або розпочинає наступний етап.

##### **1.4. Головний бенефіціар та потенційні користувачі системи**

Головний бенефіціар – група підтримки чирлідінгу.

Потенційні користувачі системи: члени групи підтримки чирлідінгу, які шукають інформацію, новини та ресурси пов'язані з чирлідінгом; потенційні

нові члени групи підтримки, які хочуть долучитися до команди чирлідінгу та отримати додаткову інформацію про це; тренери, керівники та інші працівники, пов'язані з групою підтримки, які можуть бути зацікавлені у спілкуванні, обміні досвідом та наданні інформації через сайт; зовнішні спостерігачі, які хочуть дізнатися більше про чирлідінг та активності групи підтримки.

## 2. Мета та призначення створення системи

### 2.1. Призначення системи

Система призначена для надання інформаційної підтримки та комунікації для групи підтримки чирлідінгу. Вона повинна забезпечувати доступ до різноманітних ресурсів, новин, інструкцій, календаря подій та іншої важливої інформації, пов'язаної з чирлідінгом. Система також має сприяти взаємодії та обміну досвідом між учасниками групи підтримки, тренерами та іншими зацікавленими особами.

### 2.2. Мета створення системи

Головною метою створення системи є полегшення доступу до інформації та покращення комунікації в групі підтримки чирлідінгу. Створення цього сайту допоможе членам групи швидко знайти актуальну інформацію про тренування, змагання, плани, новини та інші важливі аспекти чирлідінгу. Система також покликана залучити нових учасників та надати їм докладну інформацію про групу підтримки та можливості приєднання. Окрім того, створення системи має на меті покращити комунікацію між учасниками групи, дозволяючи обмінюватися досвідом, порадами та спілкуватися за допомогою форумів, чатів або інших інструментів. Загалом, мета створення системи полягає в покращенні організації та ефективності групи підтримки



чирлїдїнгу через забезпечення зручного та доступного онлайн-середовища для інформаційного обміну та спілкування.

### 3. Вимоги до системи

#### 3.1. Вимоги до системи в цілому

3.1.1. Вимоги до структури та функціонування системи, перелїк підсистем

3.1.1.1. Вимоги до способів і засобів інформаційного обміну між компонентами системи

Система повинна забезпечувати зручні та ефективні способи комунїкації між учасниками групи підтримки. Забезпечити можливість взаємодії через форуми, чати, систему обміну повідомленнями або електронну пошту. Реалїзувати можливість коментування та обговорення контенту сайту. Забезпечити можливість завантаження та обміну файлами (наприклад, фотографїями, відео, документами) між користувачами системи. Забезпечити можливість розсилки повідомлень та сповїщень користувачам системи.

#### 3.1.1.2. Вимоги до режимів функціонування системи

Система повинна працювати безперебїйно та надїйно, забезпечуючи постїйний доступ користувачів до необхідної інформації. Забезпечити швидкий час відгуку системи та мїнїмальні часи завантаження сторїнок. Розробити систему з урахуванням масштабування, щоб вона могла впоратися зі зростанням кїлькості користувачів та обсягу інформації.

#### 3.1.1.3. Вимоги до діагностування системи

Реалїзувати механїзми монїторингу та діагностики системи для виявлення можливих проблем та помилок. Забезпечити систему

журналювання, щоб відстежувати дії користувачів та події системи для аналізу та вирішення проблем.

#### 3.1.1.4. Вимоги до режимів управління системою

Реалізувати адміністративний інтерфейс для керування контентом сайту (додавання, редагування, видалення статей, новин, ресурсів тощо). Забезпечити можливість управління правами доступу користувачів до різних розділів та функцій системи. Розробити механізм резервного копіювання та відновлення даних для забезпечення безпеки та надійності інформації.

#### 3.1.2. Показники призначення

3.1.2.1. Параметри, що характеризують ступінь відповідності системи призначенням

Система повинна повністю задовольняти потребам групи підтримки чирлдингу, надаючи необхідну інформаційну підтримку та комунікацію. Забезпечити зручний та інтуїтивно зрозумілий інтерфейс користувача, що дозволить швидко знаходити необхідну інформацію та взаємодіяти з іншими учасниками групи. Розробити систему з урахуванням специфіки чирлдингу, зокрема, можливість відображення календаря з тренуваннями, змаганнями та іншими подіями. Забезпечити можливість швидкого оновлення інформації та контенту сайту, щоб користувачі завжди отримували актуальну інформацію.

#### 3.1.2.2. Вимоги до пристосовності системи до змін

Система повинна бути гнучкою та легко змінюватися для впровадження нових функціональностей або зміни вимог групи підтримки. Забезпечити модульну архітектуру, що дозволить додавати, видаляти або змінювати компоненти системи без значних затримок та перебудови всієї системи. Забезпечити зручні механізми налаштування та конфігурації системи, щоб адміністраторам було легко вносити зміни без необхідності втручання в код.



3.1.2.3. Вимоги до збереження працездатності системи в різних ймовірних умовах

Система повинна бути стійкою до високого навантаження та забезпечувати швидку відповідь, навіть при одночасному доступі великої кількості користувачів. Забезпечити захист від несанкціонованого доступу, використовуючи механізми аутентифікації, авторизації та шифрування даних. Розробити механізми резервного копіювання та відновлення даних для забезпечення надійності та безпеки інформації навіть у випадку виникнення непередбачуваних ситуацій, таких як відмова обладнання або втрата даних. Забезпечити можливість масштабування системи, щоб вона могла впоратися зі зростанням обсягу даних та кількості користувачів.

3.1.3. Вимоги до надійності

3.1.3.1. Склад показників надійності до системи в цілому

Максимальний час простою системи, допустимий відповідно до вимог групи підтримки. Частота відновлення роботи системи після виникнення помилок або відмов. Можливість забезпечення резервного копіювання та відновлення даних для забезпечення безпеки та надійності інформації.

3.1.3.2. Вимоги до надійності технічних засобів і програмного забезпечення

Використання стабільних та перевірених технологій і компонентів для забезпечення надійності роботи системи. Відповідність технічних засобів і програмного забезпечення стандартам та вимогам безпеки і надійності. Проведення тестування технічних засобів і програмного забезпечення на стабільність та надійність перед їх впровадженням.

3.1.3.3. Вимоги до методів оцінки і контролю показників надійності на різних стадіях створення системи

Розробка процедур оцінки надійності системи та її компонентів на етапі проектування та розробки. Використання методів тестування та аналізу для перевірки відповідності показників надійності вимогам на різних стадіях розробки. Забезпечення моніторингу та контролю показників надійності під час експлуатації системи для вчасного виявлення та усунення відхилень.

#### 3.1.4. Вимоги до ергономіки та технічної естетики

Розробка зручного та привабливого дизайну інтерфейсу користувача для забезпечення зручності його використання. Врахування принципів уніфікації та логічності в розташуванні та функціональності елементів інтерфейсу. Забезпечення простоти навігації та швидкого доступу до необхідних функцій та інформації.

3.1.5. Вимоги до експлуатації, технічного обслуговування, ремонту і зберігання компонентів системи

Забезпечення легкого встановлення, налаштування та оновлення компонентів системи. Надання зрозумілої та докладної документації, що описує процес експлуатації, технічного обслуговування, ремонту та зберігання системи. Забезпечення підтримки та консультування користувачів у випадку виникнення питань або проблем з експлуатацією системи.

#### 3.1.6 Вимоги до захисту інформації від несанкціонованого доступу

##### 3.1.6.1. Вимоги до інформаційної безпеки

Розробка механізмів аутентифікації та авторизації для контролю доступу до системи та її функцій. Застосування шифрування даних для забезпечення конфіденційності та цілісності інформації. Забезпечення механізмів реєстрації подій та виявлення спроб несанкціонованого доступу.

##### 3.1.6.2. Вимоги до антивірусного захисту



Використання антивірусного програмного забезпечення та регулярне оновлення його баз даних для виявлення та блокування вірусів та шкідливого програмного забезпечення.

#### 3.1.6.3. Розмежування відповідальності ролей при доступі

Встановлення прав доступу відповідно до ролей та обов'язків користувачів для запобігання несанкціонованому доступу до конфіденційної інформації.

#### 3.1.7. Вимоги до захисту від впливу зовнішніх факторів

Застосування заходів захисту від несприятливих погодних умов, електромагнітних перешкод, вторгнень та інших зовнішніх факторів, які можуть вплинути на працездатність системи.

#### 3.1.8. Вимоги безпеки

Дотримання вимог щодо безпеки даних та персональної інформації користувачів згідно з відповідними законодавчими вимогами та стандартами. Застосування механізмів резервного копіювання та відновлення даних для запобігання втраті важливої інформації в разі аварійних ситуацій.

### 3.2. Перелік підсистем системи (при наявності підсистем).

- Користувацький інтерфейс – розробка зручного та інтуїтивно зрозумілого інтерфейсу для користувачів. Реалізація функціональних можливостей, таких як реєстрація користувачів, вхід в систему, профіль користувача, налаштування відображення інформації тощо. Забезпечення можливості відправки запитів на підтримку та отримання відповідей з боку адміністраторів.
- Адміністративний модуль – розробка інтерфейсу для адміністраторів, який дозволяє керувати системою та взаємодіяти з користувачами. Можливість додавання, редагування та видалення інформації

про користувачів. Функціонал для відповіді на запити користувачів, надання допомоги та вирішення проблем.

- Управління контентом – система управління контентом для зручного додавання, редагування та видалення інформації на сайті. Можливість створення статей, новин, розкладу тренувань, фотогалерей та іншого контенту, пов'язаного з діяльністю групи підтримки.
- Інтеграція зі соціальними мережами – можливість публікації інформації з сайту на популярних соціальних мережах для залучення більшої аудиторії та сприяння взаємодії з користувачами.
- Система нотифікацій – розробка системи сповіщень та повідомлень, яка дозволяє надсилати повідомлення користувачам про нові тренування, зміни в розкладі, важливі оголошення та іншу важливу інформацію.
- Аналітичні засоби – впровадження аналітичних засобів для відстеження активності користувачів, аналізу популярності контенту та забезпечення звітності для адміністраторів.

### 3.3. Вимоги до видів забезпечення

#### 3.3.1. Вимоги до математичного забезпечення

Наявність математичних моделей та алгоритмів для вирішення специфічних завдань, пов'язаних з функціональністю та обробкою даних системи.

#### 3.3.2. Вимоги до інформаційного забезпечення

##### 3.3.2.1. Вимоги до складу, структури і способів організації даних в системі

Визначення необхідних типів даних, структур та зв'язків між ними для зберігання та ефективної обробки інформації. Встановлення правил і форматів для валідації та перевірки достовірності введених даних.



3.3.2.2. Вимоги до інформаційного обміну між компонентами системи  
Визначення протоколів та форматів обміну даними між різними компонентами системи для забезпечення їх взаємодії.

3.3.2.3. Вимоги до інформаційної сумісності із суміжними системами  
Забезпечення сумісності та інтеграції з іншими системами, які використовуються в групі підтримки чирлідінгу.

3.3.2.4. Вимоги використання класифікаторів та уніфікованих документів

Використання стандартних класифікаторів та уніфікованих документів для структурування та організації інформації на сайті.

3.3.2.5. Вимоги щодо застосування систем управління базами даних  
Використання платформи або системи управління базами даних для забезпечення ефективного зберігання, доступу та обробки даних.

3.3.2.6. Вимоги до структури процесу збору, обробки, передачі даних в системі представлення даних

Визначення процесів збору, обробки та передачі даних в системі, забезпечення їх ефективності та безпеки.

3.3.2.7. Вимоги до захисту даних від руйнувань при аваріях і збоях в електроживленні системи

Розробка механізмів та стратегій резервного копіювання та відновлення даних в разі аварій, збоїв або втрати електроживлення.

3.3.2.8. Вимоги до контролю, зберігання, оновленню та відновленню даних

Визначення процедур контролю, зберігання, оновлення та відновлення даних для забезпечення їх цілісності та актуальності.

3.3.2.9. Вимоги до процедури надання юридичної сили документам, що продукуються технічними засобами системи

Забезпечення можливості надання юридичної сили документам, що створюються або підписуються через сайт системи підтримки.

#### 4. Вимоги до програмного забезпечення

Розробка веб-додатку для сайту, забезпечення його функціональності, надійності та безпеки.

#### 5. Вимоги до технічного забезпечення

Вибір відповідного хостинг-провайдера або сервера для забезпечення швидкості завантаження, масштабованості та доступності сайту.

#### 6. Вимоги до методичного забезпечення

Розробка документації, посібників або інструкцій для адміністраторів та користувачів системи, щоб забезпечити їх ефективне використання.

Ці вимоги визначають основні параметри і функціональність системи підтримки чирлідінгу та визначають технічні вимоги до математичного, інформаційного, програмного, технічного та методичного забезпечення.

### 1.4. Висновки до розділу 1

Зробивши огляд тенденцій та традицій створення та просування сайтів в Україні, можна зробити такі висновки: мобільна адаптація є необхідною, оскільки користувачі все частіше відвідують сайти зі своїх мобільних пристроїв; простий та мінімалістичний дизайн набуває популярності, забезпечуючи зручну навігацію та чисте сприйняття інформації; використання елементів інтерактивності допомагає залучати користувачів і покращувати їх досвід на сайті; інтеграція з соціальними медіа дозволяє



розповсюджувати контент сайту та залучати більше відвідувачів; локалізація сайтів з урахуванням української мови та культури є важливою для забезпечення приязного сприйняття користувачами в Україні; SEO-оптимізація є необхідною для покращення видимості сайту в пошукових системах та приваблення цільової аудиторії. Загалом, розробка та просування сайтів в Україні спрямовані на забезпечення зручного користувацького досвіду, використання сучасних технологій та врахування місцевих особливостей.

Розробка та впровадження сайту для групи підтримки має свої особливості, що вимагають уваги і спеціального підходу. Перш за все, такий сайт повинен бути зрозумілим та доступним для всіх учасників групи, включаючи тих, хто має обмеження або потребує особливої допомоги. Налаштування інтуїтивного інтерфейсу, використання чітких та зрозумілих піктограм, простота навігації - усе це допоможе забезпечити зручну взаємодію з сайтом. Додатково, сайт для групи підтримки повинен мати можливість обміну інформацією, забезпечувати можливість створення спільних проєктів, а також функції для взаємодії і комунікації між учасниками групи.

У системі підтримки чирлідінгу повинні бути реалізовані функціональності, що забезпечують комунікацію та обмін інформацією між учасниками групи, організацію тренувань, спільне планування подій та управління ресурсами. Метою створення системи є полегшення комунікації та співпраці учасників групи підтримки чирлідінгу, покращення доступу до інформації та оптимізація робочих процесів. Система повинна забезпечувати ефективний інформаційний обмін між компонентами, включаючи можливість завантаження, збереження і організації даних, а також підтримку різних

режимів функціонування і діагностування системи. Також необхідно враховувати вимоги до ергономіки та технічної естетики, забезпечення безпеки та захисту інформації від несанкціонованого доступу, антивірусного захисту, а також захисту системи від впливу зовнішніх факторів. Передбачені підсистеми системи можуть включати функціонал для керування тренуваннями, планування подій, комунікації між учасниками, збереження та обробку даних.

Вимоги до програмного та технічного забезпечення включають розробку веб-додатку, вибір відповідного хостинг-провайдера, забезпечення надійності та безпеки системи. Вимоги до математичного та інформаційного забезпечення включають організацію даних, інформаційний обмін, забезпечення сумісності з іншими системами та захист даних від втрати або пошкодження. Задokumentовані вимоги до програмного забезпечення, технічного забезпечення та методичного забезпечення допоможуть забезпечити успішну розробку та функціонування системи підтримки чирлідінгу.



## РОЗДІЛ 2.

### ПРОЄКТУВАННЯ WEB-САЙТУ КОМАНДИ ГРУПИ ПІДТРИМКИ

#### 2.1. Опис моделей Web-сайту для групи підтримки

Кожна команда має потенціал піднятися або занепасти на основі групи людей, які поділяють однакові захоплення та цілі та працюють разом для досягнення успіху [15]. Цей наратив дуже поширений у елітному спорті, середовищі, яке створює значні проблеми зі здоров'ям і продуктивністю для спортсмена та тих, хто відповідає за його підтримку [4]. Враховуючи, що успіх команд підтримки спортсменів часто вимірюється спортивними результатами [3], дані підтверджують думку про те, що на сучасні досягнення спортсмена може сильно впливати функція команди підтримки спортсмена [10; 24]. Однак, враховуючи величезні проблеми з продуктивністю та здоров'ям, елітним спортивним командам може знадобитися додатковий внесок, окрім традиційних структур тренерського штабу та обмеженої кількості медичного персоналу, щоб вплинути на здоров'я та спортивні результати [5]. Дослідження, що вивчають динаміку роботи команди та ефективність команди в елітному спортивному середовищі, є однією з недооцінених областей, яка може допомогти вирішити цю зростаючу проблему. Природа командної діяльності є складним явищем, яке ще далеке від вирішення [15].

На команди, які заохочують і сприяють зусиллям одна одній для досягнення спільної мети, впливають питання лідерства [30], сприятливої командної поведінки [18], організаційного середовища [14] і здатності до адаптації [13]. Команди, які отримали знання про механізми командної

роботи (моніторинг ефективності, адаптація та фасилітаційне лідерство), мають кращі результати [6], особливо коли члени команди могли передбачати поведінку один одного та мали кращі механізми спілкування. Додавання координаційних механізмів, таких як підтримуюча командна поведінка, командне спілкування та орієнтація, є необхідними фасилітаторами командної роботи, щоб команда була успішною [23]. Крім того, високоефективне спортивне середовище створює проблеми для окремих людей, щоб ефективно функціонувати як команда [7]. Незважаючи на підвищений інтерес до концепції командної роботи [16; 21], існують численні та різні концепції командної роботи. У сучасній літературі існує обмежена точка зору щодо взаємозв'язку «командна робота – ефективність команди» [19]. Наскільки відомо авторам, мало робіт описує, що таке вхідні дані та процеси командної роботи, а також описує методології для вимірювання різних впливів і визначення їхньої ролі в оцінці командної роботи відносно результатів у високопродуктивному спорті.

З метою підтримки команди, забезпечення постійної популяризації певного виду спорту, або певного спортсмена, можна підтримувати його не лише на самому стадіоні. Одним із таких методів підтримки є створення веб-сайту для групи підтримки.

Моделі сайту для групи підтримки (чирлідінгу) можуть варіюватися в залежності від потреб та цілей конкретної групи. Однак, основні компоненти, які часто зустрічаються на сайтах груп підтримки, включають наступні елементи:

1. Головна сторінка. Ця сторінка служить вступною сторінкою сайту та надає загальну інформацію про групу підтримки. Вона може містити вітання,



опис місії групи, фотографії членів команди, посилання на інші розділи сайту та контактну інформацію.

2. Про нас. Цей розділ надає детальну інформацію про групу підтримки, включаючи історію, цілі, цінності та принципи. Тут розповідають про досягнення, спеціальності команди.

3. Розклад тренувань. Цей розділ містить інформацію про розклад тренувань групи підтримки. Вказують дні та часи тренувань, місце проведення, інструкції для нових учасників та іншу важливу інформацію.

4. Новини та оголошення. Цей розділ дозволяє оновлювати останні новини, події, зміни в розкладі або будь-яку іншу важливу інформацію.

5. Фотогалерея та відеоматеріали. В цьому розділі розміщують фотографії та відео з тренувань, виступів та інших подій групи підтримки. Це допоможе показати досягнення та створить затишну атмосферу на сайті.

6. Розділ "Питання та відповіді" або форум. Цей розділ надає можливість учасникам задавати питання, обговорювати певні теми та ділитися досвідом. Це створює спільноту і сприяє активній взаємодії між членами групи підтримки.

7. Контактна інформація. Розділ, де вказана контактна інформація групи, така як електронна пошта, телефонний номер або форма зворотного зв'язку, за допомогою якої відвідувачі можуть зв'язатися з групою для отримання додаткової інформації або запитань.

Важливо пам'ятати, що модель сайту може бути налаштована під інші потреби і відповідати вимогам кожної окремої групи підтримки. При розробці сайту варто враховувати цільову аудиторію та забезпечити зручність навігації та доступу до інформації.

## 2.2. Огляд наявних інструментів розробки Web-сайту

На сьогоднішній день існує велика кількість інструментів для розробки веб-сайтів. Вони надають різноманітні можливості, від простих редакторів коду до повноцінних інтегрованих середовищ розробки. Ось найбільш популярних інструментів:

Таблиця 2.1

### Основні інструменти розробки веб-сайту

Призначення інструменту	Інструмент	Характеристика
Редактори коду	Visual Studio Code	Це безкоштовний та потужний редактор коду з великою кількістю розширень, що полегшують роботу розробників.

Продовження таблиці 2.1

Призначення інструменту	Інструмент	Характеристика
Редактори коду	Atom	Ще один безкоштовний редактор коду, який підтримує розширення і має активну спільноту користувачів.
Інтегровані середовища розробки (IDE)	Visual Studio	Можливо, найпопулярніше IDE для розробки програмного забезпечення на мові C#. Воно має потужні можливості для розробки веб-сайтів на ASP.NET.



	JetBrains WebStorm	Інтегроване середовище розробки для веб-розробки з підтримкою JavaScript, HTML і CSS.
Фреймворки та бібліотеки	React	Популярний фреймворк для розробки інтерфейсів користувача з використанням JavaScript.
	Angular	Фреймворк для створення веб-додатків на мові TypeScript, розроблений компанією Google.
	Vue.js	Легкий фреймворк JavaScript для побудови користувацьких інтерфейсів.
Інструменти для управління версіями	Git	Де-факто стандарт для управління версіями коду. Використовується для відстеження змін в коді та спільної роботи над проектами.
Інструменти для автоматизації робочих процесів	Gulp	Інструмент для автоматизації завдань розробки, таких як компіляція CSS або JavaScript, оптимізація зображень і т. д.
	Webpack	Модульний збирач для веб-додатків, який дозволяє збирати, оптимізувати та управляти залежностями веб-сайту.
Інструменти для тестування	Jest	Фреймворк для автоматичного тестування JavaScript-коду з підтримкою відладки і тестування компонентів React.
	Selenium	Інструмент для автоматизованого тестування веб-додатків, який дозволяє записувати і відтворювати дії користувача на веб-сторінках.

Джерело: [1; 2; 8; 9; 11; 12; 20; 22; 26; 27; 28; 29]

Visual Studio Code (VS Code) є безкоштовним редактором коду, розробленим компанією Microsoft. Він підтримує різні мови програмування, включаючи HTML, CSS, JavaScript, Python, Java та багато інших. VS Code має дружній та ергономічний інтерфейс користувача, що дозволяє розробникам

працювати ефективно. Він надає можливості для автодоповнення коду, перетягування та впорядкування рядків коду, швидкого переходу між файлами та багато іншого. Однією з найбільших переваг VS Code є його розширення. Є велика кількість розширень, розроблених спільнотою, які можна встановити безпосередньо з магазину розширень VS Code. Ці розширення дозволяють додати нові функціональні можливості, наприклад, підтримку конкретних мов програмування, інструменти для відлагодження коду, системи контролю версій та багато іншого [26].

Atom є безкоштовним редактором коду, розробленим компанією GitHub. Він підтримує різні мови програмування і має екосистему розширень для розширення його можливостей. Atom має простий та кастомізований інтерфейс користувача. Ви можете налаштувати його за допомогою різних тем і плагінів для створення комфортного робочого середовища. Однією з переваг Atom є його підтримка розширень. У нього є велика спільнота розробників, яка активно створює нові розширення для покращення функціональності редактора. Ви можете встановлювати розширення з вбудованого магазину або створювати власні розширення для виконання специфічних завдань [2].

Visual Studio (VS) є інтегрованим середовищем розробки (IDE), розробленим компанією Microsoft. Це одне з найпопулярніших IDE для розробки програмного забезпечення, зокрема на мові програмування C#. VS надає потужні можливості для розробки веб-сайтів на основі ASP.NET, який є фреймворком веб-розробки, також розробленим Microsoft. Ви можете створювати статичні веб-сторінки, динамічні веб-додатки та великі підприємницькі веб-застосунки з використанням ASP.NET. VS має багато корисних функцій, таких як інтегрований відлагоджувач коду,



автодоповнення, система управління пакетами, підтримка Git та багато інших. Він також має широкий вибір розширень, які дозволяють розширити його функціональність і використовувати його для розробки в інших мовах програмування, не лише C# [27].

JetBrains WebStorm є інтегрованим середовищем розробки, спеціально призначеним для веб-розробки. Воно розроблено компанією JetBrains і надає широку підтримку для роботи з JavaScript, HTML і CSS. WebStorm надає потужні інструменти для розробки веб-додатків, включаючи функції автодоповнення, перевірку синтаксису, відлагодження коду та інструменти для рефакторингу. Воно також має вбудовану підтримку системи керування версіями, таких як Git. Окрім того, WebStorm має інтеграцію з іншими інструментами розробки веб-додатків, такими як фреймворки JavaScript, наприклад React, Angular і Vue.js. Воно підтримує розробку на основі Node.js і має можливості для роботи з серверним JavaScript, таким як Express.js [12].

React є популярним фреймворком для розробки інтерфейсів користувача (UI) з використанням JavaScript. Він розроблений компанією Facebook і використовується для створення динамічних та інтерактивних веб-додатків. Основною концепцією React є віртуальний DOM (Document Object Model), що дозволяє ефективно маніпулювати та оновлювати веб-сторінки. React використовує компонентний підхід, де веб-сторінки розбиваються на невеликі компоненти, які можуть бути повторно використані та легко управляються. React має активну спільноту розробників, багато розширень та документацію, що полегшує розробку з його використанням [20].

Angular є фреймворком для створення веб-додатків, розробленим компанією Google. Він використовує мову програмування TypeScript і надає розширені можливості для створення складних веб-додатків. Angular

пропонує повний набір інструментів для розробки, включаючи систему компонентів, модулі, маршрутизацію, управління станом та багато іншого. Він також підтримує відокремлення шаблону та логіки, що полегшує роботу в команді розробників. Angular має потужну систему директив, яка дозволяє додавати додаткову функціональність до елементів DOM. Він також надає можливості для створення анімацій та взаємодії з користувачем [1].

Vue.js є легким фреймворком JavaScript для побудови користувацьких інтерфейсів. Він має просту API і легко вивчається, що робить його популярним серед початківців та розробників, які шукають простий інструмент для створення веб-додатків.

Vue.js також використовує компонентний підхід, де веб-сторінки розбиваються на окремі компоненти, які можуть бути повторно використані та легко управляються. Він має розширену систему директив, яка дозволяє легко маніпулювати елементами DOM. Один з великих плюсів Vue.js - це його маленький розмір і швидкодія. Він добре поєднується з іншими бібліотеками та інструментами, і може використовуватись як у невеликих проєктах, так і в більших додатках [28].

Git є де-факто стандартом для управління версіями коду і є незамінним інструментом для розробників програмного забезпечення. Git дозволяє відстежувати зміни в коді та зберігати різні версії файлів. Це означає, що ви можете легко переключатися між різними версіями коду, відновлювати попередні стани і відслідковувати, хто і коли вносив зміни. Git сприяє спільній роботі над проєктами, особливо коли працює команда розробників.

Кожен розробник може клонувати (скопіювати) повний репозиторій проєкту на свій комп'ютер і працювати над кодом локально. Після внесення змін розробник може зберегти (зафіксувати) зміни в Git, індексувати (створювати



знімки) стану файлів і потім надіслати (закомітити) ці зміни в центральний репозиторій. Інші члени команди можуть отримати оновлення з центрального репозиторію, щоб мати останню версію коду. Git дозволяє створювати гілки, що представляють окремі шляхи розвитку коду. Гілки дозволяють розробникам працювати над різними функціональностями або розв'язаннями проблем, не впливаючи на головну лінію розвитку. Після завершення роботи над функціональністю розробник може злити (змерджити) гілку з головною лінією розвитку, щоб об'єднати зміни. Це дозволяє уникнути конфліктів інтеграції і зберігати історію змін чистою. Git забезпечує можливість резервного копіювання коду і відновлення попередніх станів. Якщо виникнуть проблеми або помилки, ви можете легко повернутися до попередньої версії коду, що допомагає зменшити ризики втрати даних або неправильних змін [8].

Gulp є інструментом для автоматизації завдань розробки веб-сайтів. Він дозволяє розробникам визначати та виконувати різні завдання, такі як компіляція CSS або JavaScript, оптимізація зображень, відстеження змін у файловій системі та багато іншого. Gulp використовує концепцію потоків (streams) для обробки файлів. Розробники можуть визначити різні завдання і ланцюжки обробки, де вхідні файли пройшли через послідовні етапи обробки та виводяться у вигляді оброблених файлів. Gulp має велику кількість плагінів, які розширюють його можливості. Це дозволяє розробникам використовувати готові рішення для типових задач або створювати власні плагіни для специфічних потреб [9].

Webpack є модульним збирачем для веб-додатків і дозволяє розробникам управляти залежностями, бандлити (збирати) код і оптимізувати роботу веб-сайту. Одна з ключових концепцій Webpack - це модульність. Він

дозволяє розробникам використовувати модулі, що містять код, стилі, зображення та інші ресурси, і автоматично розпізнавати та обробляти їх залежності. Webpack може обробляти різні типи файлів, включаючи JavaScript, CSS, зображення та інші. Він має можливості зведення коду, оптимізації розміру файлів, кешування та багато інших функцій, що сприяють покращенню продуктивності веб-сайту. Крім того, Webpack дозволяє розробникам використовувати різні режими збирання (development, production) для відповідної конфігурації та оптимізації збірки залежно від потреб проекту [29].

Якщо використовувати Gulp і Webpack разом, можна отримати потужну комбінацію для автоматизації та управління розробкою веб-сайту. Gulp може бути використаний для загальної автоматизації завдань розробки, таких як компіляція та оптимізація файлів, водночас як Webpack забезпечує модульну організацію коду і керування залежностями.

### **2.3. Висновки до розділу 2**

Основні моделі сайту для групи підтримки (чирлідінгу) включають головну сторінку, розділ "Про нас", розклад тренувань, новини та оголошення, фотогалерею та відеоматеріали, розділ "Питання та відповіді" або форум, а також контактну інформацію. Ці компоненти спрямовані на забезпечення ефективної комунікації та інформаційної підтримки для учасників групи. Вони допомагають залучити і утримати учасників, забезпечують зручну навігацію та доступ до необхідної інформації. Розділ "Про нас" дає можливість представити групу, її цілі та цінності. Розклад тренувань дозволяє учасникам бути в курсі графіку занять та подій. Розділ



новин та оголошень допомагає тримати учасників оновленими про останні події і зміни. Фотогалерея та відеоматеріали дозволяють демонструвати досягнення та створюють затишну атмосферу. Розділ "Питання та відповіді" або форум забезпечує можливість взаємодії між учасниками, обговорення тем та обмін досвідом. Контактна інформація надає можливість зв'язку з групою для отримання додаткової інформації або запитань. У кінцевому підсумку, модель сайту для групи підтримки (чирлідінгу) має за мету створити інформаційний центр та спільноту, де учасники можуть отримати необхідну підтримку, обмінятися досвідом та залишатися оновленими про події.

Visual Studio Code і Atom є популярними редакторами коду, що надають розширені можливості та мають активну спільноту користувачів. Visual Studio є потужним інтегрованим середовищем розробки, зокрема для розробки веб-сайтів на ASP.NET. JetBrains WebStorm - це інтегроване середовище розробки з фокусом на веб-розробку, з підтримкою JavaScript, HTML і CSS. React є популярним фреймворком для розробки інтерфейсів користувача з використанням JavaScript. Angular - це фреймворк для створення веб-додатків на мові TypeScript, розроблений компанією Google. Vue.js - це легкий фреймворк JavaScript для побудови користувацьких інтерфейсів. Git є стандартом для управління версіями коду і спільної роботи над проектами, забезпечуючи зручність відстеження змін та відновлення попередніх станів. Gulp є інструментом для автоматизації завдань розробки, таких як компіляція CSS або JavaScript, оптимізація зображень і багато іншого. Webpack є модульним збирачем для веб-додатків, дозволяючи збирати, оптимізувати та управляти залежностями веб-сайту. Ці інструменти та фреймворки є незамінними для розробки веб-сайтів і допомагають розробникам збільшити продуктивність, полегшити роботу з кодом і досягати

кращих результатів. Кожен з цих інструментів має свої особливості і варто вибрати той, який найкраще відповідає вашим потребам і вимогам проекту.





## **РОЗДІЛ 3. РОЗРОБКА ТА ВПРОВАДЖЕННЯ WEB-САЙТУ КОМАНДИ ГРУПИ ПІДТРИМКИ (ЧИРЛІДИНГУ)**

### **3.1. Функціональні характеристики розробленого Web-сайту**

В ході проведеного дослідження був розроблений веб-сайт Kolibri для чирлідінгу. Даний сайт виконує наступні функції:

1. Реєстрація користувачів. Дозволяє відвідувачам сайту створювати облікові записи, щоб мати доступ до додаткових функцій або можливостей.
2. Автентифікація користувачів. Забезпечує механізми перевірки ідентифікації користувачів, щоб гарантувати доступ тільки авторизованим особам.
3. Профілі користувачів. Дозволяє користувачам створювати та керувати своїми профілями, включаючи інформацію про себе та зміну налаштувань облікового запису.
4. Форум або спільнота: Забезпечує можливість спілкування між користувачами, обміну досвідом, питаннями та відповідями, що сприяють підтримці і спільноті групи Kolibri.
5. Розділ новин і оголошень. Забезпечує майданчик для розміщення актуальної інформації про події, зустрічі, тренування та інші новини, які можуть бути цікавими для членів групи підтримки.
6. Календар подій. Надає можливість планування і відстеження подій, зустрічей, тренувань та інших активностей групи Kolibri.
7. Ресурси та матеріали. Забезпечує доступ до різноманітних ресурсів, матеріалів, інструкцій, відео, фотографій тощо, які допомагають учасникам групи розвиватися та покращувати свої навички.

8. Пошук. Надає можливість швидкого пошуку інформації на сайті, що допомагає користувачам знаходити потрібні ресурси, новини, повідомлення та інше.

9. Контактна інформація. Дозволяє користувачам зв'язуватися з адміністрацією сайту або представниками групи Kolibri для отримання додаткової інформації або підтримки.

Веб-сайт містить 4 контролери:

1. Контролер GalleryController веб-додатка на основі фреймворку ASP.NET Core. Контролер відповідає за обробку запитів, пов'язаних з галереєю файлів.



```
1 using Kolibri.Models;
2 using Kolibri.Models;
3 using Microsoft.AspNetCore.Mvc;
4
5 namespace Kolibri.Controllers
6 {
7     public class GalleryController : Controller
8     {
9         ApplicationContext _context;
10        IWebHostEnvironment _appEnvironment;
11
12        public GalleryController(ApplicationContext context, IWebHostEnvironment appEnvironment)
13        {
14            _context = context;
15            _appEnvironment = appEnvironment;
16        }
17
18        public IActionResult Index()
19        {
20            return View(_context.Files.ToList());
21        }
22        [HttpGet]
23        public IActionResult AddFile()
24        {
25            return View();
26        }
27        [HttpPost]
28        public async Task AddFile(IFormFile uploadedFile)
29        {
30            if (uploadedFile != null)
31            {
32                string path = "/files/" + uploadedFile.FileName;
33                using (var fileStream = new FileStream(_appEnvironment.WebRootPath + path, FileMode.Create))
34                {
35                    await uploadedFile.CopyToAsync(fileStream);
36                }
37                FileModel file = new FileModel { Name = uploadedFile.FileName, Path = path };
38                _context.Files.Add(file);
39                _context.SaveChanges();
40            }
41            return RedirectToAction("Index");
42        }
43    }
44 }
45
46
47
48
```

Рис. 3.1. Контролер GalleryController

• У рядках 1-5 відбувається імпорт необхідних просторів із використанням директив **using**. Ці простори інкапсулюють класи та методи, які необхідні для роботи контролера.



- У рядках 7-17 оголошується клас **GalleryController**, який успадковує базовий клас **Controller**.

- Контролер має два приватних поля: **ApplicationContext \_context** і **\_appEnvironment** (рядки 10-11). **ApplicationContext** є контекстом бази даних, який використовується для взаємодії з таблицею файлів. **\_appEnvironment** є інтерфейсом для доступу до середовища веб-додатка.

- У рядках 13-18 визначається конструктор **GalleryController**, який отримує залежності (**context** та **appEnvironment**) і зберігає їх в приватних полях для подальшого використання.

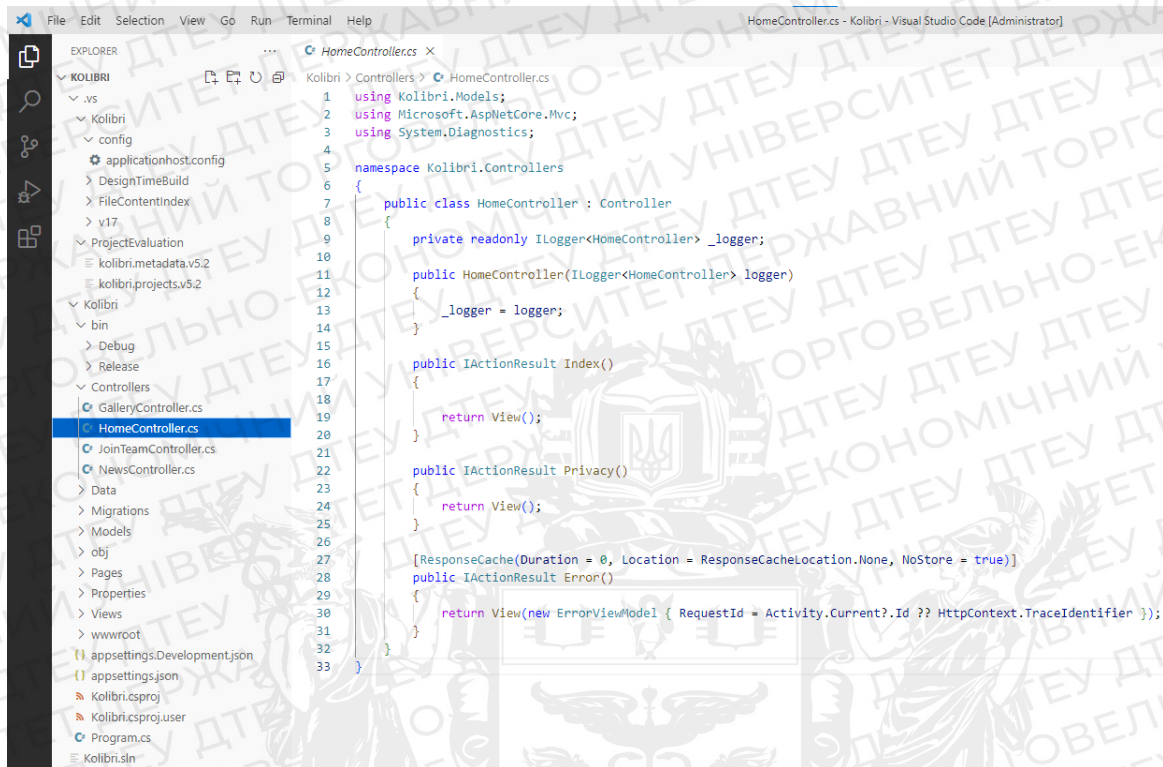
- У рядках 20-28 описано метод **Index()**, який повертає представлення **View** зі списком файлів, отриманих з бази даних за допомогою **\_context.Files.ToList()**.

- У рядках 30-34 описано метод **AddFile()** з атрибутом **[HttpGet]**, який повертає представлення **View** для додавання файлу.

- У рядках 36-53 описано метод **AddFile()** з атрибутом **[HttpPost]**, який приймає завантажений файл (**IFormFile uploadedFile**) і зберігає його на сервері. Файл копіюється у визначену шляху (**path**) та додається до бази даних (**\_context.Files.Add(file)**). Після цього зміни зберігаються (**\_context.SaveChanges()**), і виконується перенаправлення на метод **Index()**.

Загальна функціональність цього контролера полягає в тому, що він відповідає за відображення списку файлів у галереї (**Index()**) і додавання нового файлу (**AddFile()**).

2. Контролер **HomeController** веб-додатка на основі фреймворку **ASP.NET Core**. Контролер відповідає за обробку запитів, пов'язаних з головною сторінкою та приватністю.



```
File Edit Selection View Go Run Terminal Help
HomeController.cs - Kolibri - Visual Studio Code [Administrator]

EXPLORER
KOLIBRI
  .vs
  Kolibri
    config
    applicationhost.config
    DesignTimeBuild
    FileContentIndex
    v17
    ProjectEvaluation
      kolibri.metadata.v5.2
      kolibri.projects.v5.2
    Kolibri
      bin
        Debug
        Release
      Controllers
        GalleryController.cs
        HomeController.cs
        JoinTeamController.cs
        NewsController.cs
      Data
      Migrations
      Models
      obj
      Pages
      Properties
      Views
      wwwroot
      appsettings.Development.json
      appsettings.json
      Kolibri.csproj
      Kolibri.csproj.user
      Program.cs
      Kolibri.sln

HomeController.cs
1 using Kolibri.Models;
2 using Microsoft.AspNetCore.Mvc;
3 using System.Diagnostics;
4
5 namespace Kolibri.Controllers
6 {
7     public class HomeController : Controller
8     {
9         private readonly ILogger<HomeController> _logger;
10
11         public HomeController(ILogger<HomeController> logger)
12         {
13             _logger = logger;
14         }
15
16         public IActionResult Index()
17         {
18             return View();
19         }
20
21         public IActionResult Privacy()
22         {
23             return View();
24         }
25
26         [ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]
27         public IActionResult Error()
28         {
29             return View(new ErrorViewModel { RequestId = Activity.Current?.Id ?? HttpContext.TraceIdentifier });
30         }
31     }
32 }
33
```

Рис. 3.2. Контролер HomeController

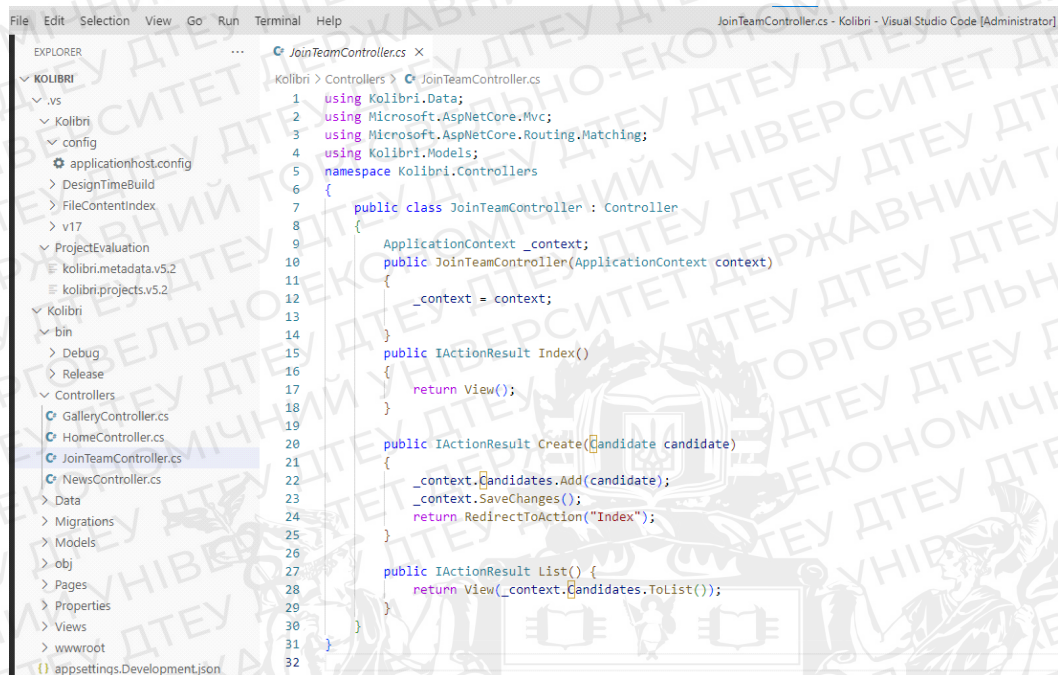
- У рядках 1-4 відбувається імпорт необхідних просторів із використанням директив **using**. Ці простори інкапсулюють класи та методи, які необхідні для роботи контролера.
- У рядках 6-16 оголошується клас **HomeController**, який успадковує базовий клас **Controller**.
- Контролер має приватне поле **\_logger** (рядок 8), яке представляє інтерфейс **ILogger<HomeController>**. Це поле використовується для журналювання подій та діагностики.
- У рядках 10-15 визначається конструктор **HomeController**, який отримує залежність **logger** і зберігає його в приватному полі **\_logger** для подальшого використання.



- У рядках 17-22 описано метод **Index()**, який повертає представлення **View**. Цей метод використовується для відображення головної сторінки додатка.
- У рядках 24-27 описано метод **Privacy()**, який повертає представлення **View**. Цей метод використовується для відображення сторінки приватності.
- У рядках 29-34 описано метод **Error()**, який повертає представлення **View** для обробки помилок. Він також створює новий екземпляр **ErrorViewModel** і передає його до представлення, яке може використовувати його для відображення додаткової інформації про помилку.

Цей контролер відповідає за базові дії на головній сторінці та сторінці приватності, а також обробляє помилки, якщо вони виникають. Він використовує інтерфейс **ILogger** для журналювання подій.

3. Контролер **JoinTeamController** веб-додатка на основі фреймворку **ASP.NET Core**. Контролер відповідає за обробку запитів, пов'язаних з приєднанням до команди.



```
File Edit Selection View Go Run Terminal Help
JoinTeamController.cs - Kolibri - Visual Studio Code [Administrator]

EXPLORER
KOLIBRI
  .vs
  Kolibri
  config
  applicationhost.config
  DesignTimeBuild
  FileContentIndex
  v17
  ProjectEvaluation
  kolibri.metadata.v5.2
  kolibri.projects.v5.2
  Kolibri
    bin
    Debug
    Release
    Controllers
      GalleryController.cs
      HomeController.cs
      JoinTeamController.cs
      NewsController.cs
    Data
    Migrations
    Models
    obj
    Pages
    Properties
    Views
    wwwroot
  appsettings.Development.json

JoinTeamController.cs
Kolibri > Controllers > JoinTeamController.cs
1 using Kolibri.Data;
2 using Microsoft.AspNetCore.Mvc;
3 using Microsoft.AspNetCore.Routing.Matching;
4 using Kolibri.Models;
5 namespace Kolibri.Controllers
6 {
7     public class JoinTeamController : Controller
8     {
9         ApplicationContext _context;
10        public JoinTeamController(ApplicationContext context)
11        {
12            _context = context;
13        }
14
15        public IActionResult Index()
16        {
17            return View();
18        }
19
20        public IActionResult Create(Candidate candidate)
21        {
22            _context.Candidates.Add(candidate);
23            _context.SaveChanges();
24            return RedirectToAction("Index");
25        }
26
27        public IActionResult List() {
28            return View(_context.Candidates.ToList());
29        }
30    }
31 }
32 }
```

Рис. 3.3. Контролер JoinTeamController

- У рядках 1-6 відбувається імпорт необхідних просторів із використанням директив **using**. Ці простори інкапсулюють класи та методи, які необхідні для роботи контролера.
- У рядках 8-18 оголошується клас **JoinTeamController**, який успадковує базовий клас **Controller**.
- Контролер має приватне поле **\_context** (рядок 10), яке представляє контекст бази даних **ApplicationContext**. Це поле використовується для взаємодії з таблицею кандидатів.
- У рядках 12-16 визначається конструктор **JoinTeamController**, який отримує залежність **context** і зберігає його в приватному полі **\_context** для подальшого використання.



- У рядках 20-25 описано метод **Index()**, який повертає представлення **View**. Цей метод використовується для відображення сторінки з формою приєднання до команди.
- У рядках 27-34 описано метод **Create(Candidate candidate)**, який приймає об'єкт **Candidate** в якості параметра. Він додає кандидата до контексту бази даних (**\_context.Candidates.Add(candidate)**) і зберігає зміни (**\_context.SaveChanges()**). Після цього виконується перенаправлення на метод **Index()**.
- У рядках 36-41 описано метод **List()**, який повертає представлення **View** зі списком кандидатів, отриманих з бази даних за допомогою **\_context.Candidates.ToList()**.

Цей контролер відповідає за сторінку приєднання до команди. Він має методи для відображення форми, додавання кандидата до бази даних та відображення списку кандидатів.

4. Контролер **NewsController** веб-додатка на основі фреймворку **ASP.NET Core**. Контролер відповідає за обробку запитів, пов'язаних з новинами.

NewsController.cs

Kolibri > Controllers > NewsController.cs

```
1 using Kolibri.Data;
2 using Kolibri.Models;
3 using Microsoft.AspNetCore.Mvc;
4
5 namespace Kolibri.Controllers
6 {
7     public class NewsController : Controller
8     {
9         ApplicationContext _context;
10        public NewsController(ApplicationContext context)
11        {
12            _context = context;
13        }
14
15        public IActionResult Index()
16        {
17            return View(_context.News.ToList());
18        }
19
20        public IActionResult Create()
21        {
22            return View();
23        }
24
25        [HttpPost]
26        public IActionResult Create(News news)
27        {
28            _context.News.Add(news);
29            _context.SaveChanges();
30            return RedirectToAction("Index");
31        }
32
33        public IActionResult Details(int id)
34        {
35            if(_context.News.Any(x=>x.Id == id))
36            {
37
38                News news = _context.News.First(x => x.Id == id);
39                return View(news);
40            }
41            return View();
42        }
43    }
44 }
45
46
```

Рис. 3.4. Контролер NewsController



- У рядках 1-5 відбувається імпорт необхідних просторів із використанням директив **using**. Ці простори інкапсулюють класи та методи, які необхідні для роботи контролера.
- У рядках 7-17 оголошується клас **NewsController**, який успадковує базовий клас **Controller**.
- Контролер має приватне поле **\_context** (рядок 10), яке представляє контекст бази даних **ApplicationContext**. Це поле використовується для взаємодії з таблицею новин.
- У рядках 12-16 визначається конструктор **NewsController**, який отримує залежність **context** і зберігає його в приватному полі **\_context** для подальшого використання.
- У рядках 19-25 описано метод **Index()**, який повертає представлення **View** зі списком новин, отриманих з бази даних за допомогою **\_context.News.ToList()**.
- У рядках 27-31 описано метод **Create()**, який повертає представлення **View** для створення нової новини.
- У рядках 33-39 описано метод **Create(News news)** з атрибутом **[HttpPost]**, який приймає об'єкт **News** в якості параметра. Він додає новину до контексту бази даних (**\_context.News.Add(news)**) і зберігає зміни (**\_context.SaveChanges()**). Після цього виконується перенаправлення на метод **Index()**.
- У рядках 41-51 описано метод **Details(int id)**, який приймає ідентифікатор новини в якості параметра. Він перевіряє, чи існує новина з таким ідентифікатором в базі даних (**\_context.News.Any(x => x.Id == id)**). Якщо така новина існує, вона отримується з бази даних (**\_context.News.First(x => x.Id == id)**) і передається до представлення **View**

для відображення деталей новини. Якщо новина не знайдена, повертається пусте представлення.

Цей контролер відповідає за сторінку новин. Він має методи для відображення списку новин, створення нової новини та відображення деталей окремої новини.

### 3.2. Опис архітектури розробленого Web-сайту

Розроблений веб-сайт Kolibri має наступну архітектуру:

- Клієнтська сторона (Front-end):
- HTML, CSS та JavaScript: Веб-сторінка реалізовані за допомогою HTML для структури, CSS для стилізації та JavaScript для динамічної взаємодії з користувачем (рис. 3.4).



```

1  @([
2  ViewData["Title"] = "Home Page";
3  ])
4
5
6  <!--? slider Area Start-->
7  <div class="slider-area slider-height position-relative" data-background="assets/img/hero/h1_hero.png">
8  <div class="slider-active">
9  <!-- Single Slider -->
10 <div class="single-slider">
11 <div class="slider-cap-wrapper">
12 <div class="hero-caption">
13 @*<span>Kolibri Team</span>*@
14 <h1 data-animation="fadeInLeft" data-delay=".4s">Kolibri Team</h1>
15 <p data-animation="fadeInLeft" data-delay=".6s">
16 Дух команди - ключ до найкращих результатів!
17 </p>
18 <!-- Hero Btn -->
19 <a href="/JoinTeam" class="btn" data-animation="fadeInLeft" data-delay=".9s">Приєднатися
20 </div>
21 <div class="hero-img">
22 
24 </div>
25 </div>
26 </div>
27 <!-- Video icon -->
28 <div class="video-icon">
29 <a class="popup-video btn-icon" href="https://www.youtube.com/watch?v=X-EUxJVtskE"><i class="fas fa-p
30 </div>
31 </div>
32 <!-- slider Area End-->
33 <!--? Services Area Start -->
34 <section class="services-section section-padding30 fix">
35 <div class="container">
36 <div class="row justify-content-center">
37 <div class="col-xl-10 col-lg-10 col-md-12">
38 <!-- Section Tittle -->
39 <div class="section-tittle text-center mb-70">
40 <span>Наша команда</span>
41 <h2>Тренування роблять нас кращими, змагання - непереможними.</h2>
42 </div>
43 </div>
44 </div>
45 <div class="services-active">
46 <div class="single-services section-over1 text-center">

```

Рис. 3.4. Клієнтська сторінка Kolibri

- Фреймворк або бібліотека для фронтенду: Використання популярних фреймворків або бібліотек, таких як React, Angular або Vue.js, для побудови динамічних компонентів та реактивного інтерфейсу користувача.
- Серверна сторона (Back-end) (рис. 3.5):
- ASP.NET Core: Фреймворк ASP.NET Core використовується для створення серверної частини сайту. Він забезпечує обробку запитів, маршрутизацію, взаємодію з базою даних та інші серверні функції.





```

1  B SolutionFileName Kolibri.sln LangName ru-RU
2  Configuration Debug LangID 1049 DesignTimeBuild true
3  SolutionDir C:\Users\maxbo\source\repos\Kolibri\
4  sln BuildingInsideVisualStudio DefineExplicitDefaults Platform AnyC
5  DriverData C:\Windows\System32\Drivers\DriverDataR EFC_6712S ForceIdentityAuthenticationT
6  Windows_NT Path C:\Program Files\Microsoft MPI\Bin\;C:\Windows\system32;C:\Windows;C:\Window
7  SystemRoot
8  C:\Windows TEMP C:\Users\maxbo\AppData\Local\Temp ThreadedWaitDialogDpiContext
9  USERDOMAIN USERDOMAIN RoamingProfile USERNAME maxbo USERPROFILE C:\U
10 VSAPPIDIR VSAPPIDNAME
11 devenv.exe VSLANG VSSKUEDITION Community Windir MSBuildToolsPath32
12 VsInstallRoot C:\Program Files\Microsoft Visual Studio\2022\Community\MSBuildToolsRoot
13 VCTargetsPath C:\Program Files\Microsoft Visual Studio\2022\Community\MSBuild\Microsoft\VC\v170\
14 NuGetProps C:\Program Files\Microsoft Visual Studio\2022\Community\MSBuild\Microsoft\NuGet\17.0\Micro
15 JavaScript MSBuildUseVisualStudioDirectoryLayout ([MSBuild]::IsRunningFromVisualStudio())
16 C:\Program Files\Microsoft Visual Studio\2022\Community\MSBuild\Current\Bin\..\..\Common7\IDE\Co
17 C:\Program Files\Microsoft Visual Studio\2022\Community\Common7\IDE\CommonExtensions\Microsoft\NuGet\M
18 Debug;Release Platforms
19 OutputType Exe C:\Program Files\dotnet\sdk\7.0.102\Sdks\Microsoft.NET.Sdk.Web.ProjectSystem\tar
20 FileAlignment 512+ ErrorReport prompt-AssemblyName $(MSBuildProjectName)/
21 RootNamespace $(MSBuildProjectName.Replace(" ", "_"))1
22 Deterministic Optimize3 false4 $(Configuration) == 'Debug' 5 & PlatformWithoutCon
23 IsPackable WarnOnPackingNonPackableProject MSBuildProjectDefaultTargets Build
24 _IsExecutable C:\Program Files\dotnet\sdk\7.0.102\Sdks\Microsoft.NET.Sdk\targets\Microsoft.NET.Sdk.Co
25 VersionPrefix 1.0.0 C:\Program Files\dotnet\sdk\7.0.102\Sdks\Microsoft.NET.Sdk\targets\Micros
26 $(Authors)
27 AssemblyTitle Product PlatformName C:\Program Files\dotnet\sdk\7.0.102\Sdks\Microsoft
28 OutputPath bin\Debug\net7.0\$(OutputPath)\$(TargetFramework.ToLowerInvariant())\C:\Pr
29 bin\Debug\
30 $(OutputPath) PublishProfileImported C:\Program Files\dotnet\sdk\7.0.102\Sdks\Microsoft.NET.S
31 Properties PublishProfileRootFolder C:\Users\maxbo\source\repos\Kolibri\Kolibri\Properties\P
32 EnableDefaultWindowsPlatform $(TargetFrameworkIdentifier) == '.NETCoreApp' and $(MSBuild
33 !AppendTargetFrameworkToOutputPath $(UsingNETSdkDefaults) == 'true' ProduceReferenceAs
34 SelfContained $(TargetFrameworkIdentifier) == '.NETCoreApp' and $(HasRuntimeOutput) == 'true'!
35 UseAppHost# DefaultAppHostRuntimeIdentifier $(NETCoreSdkRuntimeIdentifier)% $(Default
36 $(RuntimeIdentifier) == '' and
37 (('UseAppHost' == 'true' and $(ProjectDepsFilePath) == '') or
38 $('EnableComHosting' == 'true' and $_IsExecutable) != 'true') or
39 $('UseIJSHost' == 'true')& AppendRuntimeIdentifierToOutputPath Check
40 PublishDir bin\Debug\net7.0\publish\6 $(OutputPath)\$(PublishDirName)\7 NuGetFallbackFol
41 NETCOREAPP $(TargetFrameworkIdentifier.Replace('.', '')) ToUpperInvariant() GenerateDocumentati
42 TargetRuntime Managed AlwaysUseNumericalSuffixInItemNames C:\Program Files\Microsoft V
43
44 GenerateCompiledExpressionsTempFile;
45
46 DesignTimeXamlMarkupCompilation;

```

Рис. 3.5. Серверна сторона Kolibri



□ База даних: Використання реляційної бази даних Microsoft SQL Server для зберігання даних сайту. Взаємодія з базою даних здійснена за допомогою ORM (Object-Relational Mapping) бібліотеки, такої як Entity Framework.

□ Мова програмування: Код серверної сторони написаний мовою C#.

□ Web-сервер: Додаток розгорнутий на веб-сервері, такому як Internet Information Services (IIS), який оброблює HTTP-запити та передає їх до додатку ASP.NET Core.

□ Бізнес-логіка:

□ Контролери: Контролери відповідають за обробку запитів та взаємодію з користувачем. Вони приймають запити, обробляють їх і відправляють відповіді.

□ Моделі: Моделі відображають структуру та дані, з якими працює сайт. Вони включають класи, які відображають таблиці бази даних та інші об'єкти даних, необхідні для виконання операцій.

□ Сервіси: Сервіси містять бізнес-логіку та виконують різноманітні операції, такі як звернення до бази даних, обробка даних, валідація тощо. Вони можуть бути використані контролерами для виконання певних дій.

□ Зв'язок між клієнтською та серверною сторонами:

□ HTTP-протокол: Клієнтська сторона взаємодіє з серверною стороною за допомогою HTTP-протоколу. Клієнтські запити надсилаються на сервер, а сервер відповідає на запити з необхідною інформацією або даними.

□ AJAX: Для реалізації асинхронних запитів і оновлення частини сторінки без перезавантаження використовується технологія AJAX.



□ Формат передачі даних: Дані можуть передаватись між клієнтом і сервером у форматі JSON або XML.

### 3.3. Висновки до розділу 3

Розроблений сайт має на меті підтримку групи Kolibri (чирлідінг) і її членів. Сайт надає можливість реєстрації користувачів та автентифікації для забезпечення безпеки та доступу до додаткових функцій. На сайті може бути наявна спільнота або форум, де учасники групи Kolibri можуть спілкуватися, обмінюватися досвідом та отримувати підтримку. Сайт може містити розділ новин і оголошень, де публікуються актуальні події, тренування та інші новини групи Kolibri. Календар подій на сайті допомагає учасникам групи планувати та відстежувати розклад тренувань, зустрічей та інших активностей. Сайт може надавати різноманітні ресурси та матеріали, які допомагають учасникам групи розвиватися та вдосконалювати свої навички. Наявність пошукової функції на сайті сприяє швидкому знаходженню потрібної інформації. Сайт надає контактну інформацію, щоб учасники могли звертатися до адміністрації сайту або представників групи Kolibri для отримання додаткової підтримки.

Архітектура розробленого сайту базується на фреймворку ASP.NET Core і включає клієнтську та серверну сторони. Клієнтська сторона використовує HTML, CSS та JavaScript для створення зручного та привабливого інтерфейсу користувача.

Серверна сторона використовує ASP.NET Core для обробки запитів користувачів, взаємодії з базою даних та виконання бізнес-логіки. Архітектура включає розділення на контролери, моделі та сервіси.

Контролери відповідають за обробку запитів і взаємодію з користувачем, моделі представляють дані, з якими працює сайт, а сервіси містять бізнес-логіку та виконують різноманітні операції. Зв'язок між клієнтською та серверною сторонами здійснюється за допомогою HTTP-протоколу. Клієнтські запити надсилаються на сервер, який обробляє їх і повертає відповіді з необхідною інформацією або даними. Ця архітектура дозволяє розбити розробку сайту на окремі компоненти, що спрощує розвиток, тестування та підтримку додатку. Вона також забезпечує гнучкість і масштабованість, що дозволяє додавати нові функціональні можливості та розширювати функціонал існуючих компонентів. Загалом, архітектура розробленого сайту на базі ASP.NET Core забезпечує стабільну та ефективну роботу, надійність та безпеку, а також дозволяє швидко реагувати на зміни вимог та розширювати функціонал сайту.



## ВИСНОВКИ

В результаті проведеного дослідження були зроблені наступні висновки:

1. Зробивши огляд тенденцій та традицій створення та просування сайтів в Україні, можна зробити такі висновки: мобільна адаптація є необхідною, оскільки користувачі все частіше відвідують сайти зі своїх мобільних пристроїв; простий та мінімалістичний дизайн набуває популярності, забезпечуючи зручну навігацію та чисте сприйняття інформації; використання елементів інтерактивності допомагає залучати користувачів і покращувати їх досвід на сайті; інтеграція з соціальними медіа дозволяє розповсюджувати контент сайту та залучати більше відвідувачів; локалізація сайтів з урахуванням української мови та культури є важливою для забезпечення приязного сприйняття користувачами в Україні; SEO-оптимізація є необхідною для покращення видимості сайту в пошукових системах та приваблення цільової аудиторії. Загалом, розробка та просування сайтів в Україні спрямовані на забезпечення зручного користувацького досвіду, використання сучасних технологій та врахування місцевих особливостей.

2. Розробка та впровадження сайту для групи підтримки має свої особливості, що вимагають уваги і спеціального підходу. Перш за все, такий сайт повинен бути зрозумілим та доступним для всіх учасників групи, включаючи тих, хто має обмеження або потребує особливої допомоги. Налаштування інтуїтивного інтерфейсу, використання чітких та зрозумілих піктограм, простота навігації - усе це допоможе забезпечити зручну взаємодію з сайтом.

Додатково, сайт для групи підтримки повинен мати можливість обміну інформацією, забезпечувати можливість створення спільних проєктів, а також функції для взаємодії і комунікації між учасниками групи.

3. У системі підтримки чирлідінгу повинні бути реалізовані функціональності, що забезпечують комунікацію та обмін інформацією між учасниками групи, організацію тренувань, спільне планування подій та управління ресурсами. Метою створення системи є полегшення комунікації та співпраці учасників групи підтримки чирлідінгу, покращення доступу до інформації та оптимізація робочих процесів. Система повинна забезпечувати ефективний інформаційний обмін між компонентами, включаючи можливість завантаження, збереження і організації даних, а також підтримку різних режимів функціонування і діагностування системи. Також необхідно враховувати вимоги до ергономіки та технічної естетики, забезпечення безпеки та захисту інформації від несанкціонованого доступу, антивірусного захисту, а також захисту системи від впливу зовнішніх факторів. Передбачені підсистеми системи можуть включати функціонал для керування тренуваннями, планування подій, комунікації між учасниками, збереження та обробку даних. Вимоги до програмного та технічного забезпечення включають розробку веб-додатку, вибір відповідного хостинг-провайдера, забезпечення надійності та безпеки системи. Вимоги до математичного та інформаційного забезпечення включають організацію даних, інформаційний обмін, забезпечення сумісності з іншими системами та захист даних від втрати або пошкодження. Задokumentовані вимоги до програмного забезпечення, технічного забезпечення та методичного забезпечення допоможуть забезпечити успішну розробку та функціонування системи підтримки чирлідінгу.



4. Основні моделі сайту для групи підтримки (чирлідінгу) включають головну сторінку, розділ "Про нас", розклад тренувань, новини та оголошення, фотогалерею та відеоматеріали, розділ "Питання та відповіді" або форум, а також контактну інформацію. Ці компоненти спрямовані на забезпечення ефективної комунікації та інформаційної підтримки для учасників групи. Вони допомагають залучити і утримати учасників, забезпечують зручну навігацію та доступ до необхідної інформації. Розділ "Про нас" дає можливість представити групу, її цілі та цінності. Розклад тренувань дозволяє учасникам бути в курсі графіку занять та подій. Розділ новин та оголошень допомагає тримати учасників оновленими про останні події і зміни. Фотогалерея та відеоматеріали дозволяють демонструвати досягнення та створюють затишну атмосферу. Розділ "Питання та відповіді" або форум забезпечує можливість взаємодії між учасниками, обговорення тем та обмін досвідом. Контактна інформація надає можливість зв'язку з групою для отримання додаткової інформації або запитань. У кінцевому підсумку, модель сайту для групи підтримки (чирлідінгу) має за мету створити інформаційний центр та спільноту, де учасники можуть отримати необхідну підтримку, обмінятися досвідом та залишатися оновленими про події.

5. Visual Studio Code і Atom є популярними редакторами коду, що надають розширені можливості та мають активну спільноту користувачів. Visual Studio є потужним інтегрованим середовищем розробки, зокрема для розробки веб-сайтів на ASP.NET. JetBrains WebStorm - це інтегроване середовище розробки з фокусом на веб-розробку, з підтримкою JavaScript, HTML і CSS. React є популярним фреймворком для розробки інтерфейсів користувача з використанням JavaScript.

Angular - це фреймворк для створення веб-додатків на мові TypeScript, розроблений компанією Google. Vue.js - це легкий фреймворк JavaScript для побудови користувацьких інтерфейсів. Git є стандартом для управління версіями коду і спільної роботи над проектами, забезпечуючи зручність відстеження змін та відновлення попередніх станів. Gulp є інструментом для автоматизації завдань розробки, таких як компіляція CSS або JavaScript, оптимізація зображень і багато іншого. Webpack є модульним збирачем для веб-додатків, дозволяючи збирати, оптимізувати та управляти залежностями веб-сайту. Ці інструменти та фреймворки є незамінними для розробки веб-сайтів і допомагають розробникам збільшити продуктивність, полегшити роботу з кодом і досягати кращих результатів. Кожен з цих інструментів має свої особливості і варто вибрати той, який найкраще відповідає вашим потребам і вимогам проекту.

6. Розроблений сайт має на меті підтримку групи Kolibri (чирлідінг) і її членів. Сайт надає можливість реєстрації користувачів та автентифікації для забезпечення безпеки та доступу до додаткових функцій. На сайті може бути наявна спільнота або форум, де учасники групи Kolibri можуть спілкуватися, обмінюватися досвідом та отримувати підтримку. Сайт може містити розділ новин і оголошень, де публікуються актуальні події, тренування та інші новини групи Kolibri. Календар подій на сайті допомагає учасникам групи планувати та відстежувати розклад тренувань, зустрічей та інших активностей.

Сайт може надавати різноманітні ресурси та матеріали, які допомагають учасникам групи розвиватися та вдосконалювати свої навички. Наявність пошукової функції на сайті сприяє швидкому знаходженню потрібної інформації.



Сайт надає контактну інформацію, щоб учасники могли звертатися до адміністрації сайту або представників групи Kolibri для отримання додаткової підтримки.

7. Архітектура розробленого сайту базується на фреймворку ASP.NET Core і включає клієнтську та серверну сторони. Клієнтська сторона використовує HTML, CSS та JavaScript для створення зручного та привабливого інтерфейсу користувача. Серверна сторона використовує ASP.NET Core для обробки запитів користувачів, взаємодії з базою даних та виконання бізнес-логіки. Архітектура включає розділення на контролери, моделі та сервіси. Контролери відповідають за обробку запитів і взаємодію з користувачем, моделі представляють дані, з якими працює сайт, а сервіси містять бізнес-логіку та виконують різноманітні операції. Зв'язок між клієнтською та серверною сторонами здійснюється за допомогою HTTP-протоколу. Клієнтські запити надсилаються на сервер, який обробляє їх і повертає відповіді з необхідною інформацією або даними. Ця архітектура дозволяє розбити розробку сайту на окремі компоненти, що спрощує розвиток, тестування та підтримку додатку. Вона також забезпечує гнучкість і масштабованість, що дозволяє додавати нові функціональні можливості та розширювати функціонал існуючих компонентів. Загалом, архітектура розробленого сайту на базі ASP.NET Core забезпечує стабільну та ефективну роботу, надійність та безпеку, а також дозволяє швидко реагувати на зміни вимог та розширювати функціонал сайту.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- Angular. URL: <https://angular.io/>
- Atom. URL: <https://atom.en.uptodown.com/windows>
- Cruickshank A, Collins D. Change management: the case of the elite sport performance team. *J Chang Manag.* 2012;12(2):209–29.
- Dijkstra HP, Pollock N, Chakraverty R, Alonso JM. Managing the health of the elite athlete: a new integrated performance health management and coaching model. *Br J Sports Med.* 2014;48(7):523–31.
- Drust B, Green M. Science and football: evaluating the influence of science on performance. *J Sports Sci.* 2013;31:1377–82.
- Entin EE, Serfaty D. Adaptive team coordination. *Hum Factors.* 1999;41(2):312–25.
- Eubank M, Nesti M, Cruickshank A. Understanding high performance sport environments: impact for the professional training and supervision of sport psychologists. *Sport Exerc Psychol Rev.* 2014;10(2):30–6.
- Git. URL: <https://git-scm.com/>
- Gulp. URL: <https://gulpjs.com/>
- Guzmán I. Measuring efficiency and sustainable growth in Spanish football teams. *Eur Sport Manag Q.* 2006;6(3):267–87.
- Jest. URL: <https://jestjs.io/uk/>
- JetBrains WebStorm. URL: <https://www.jetbrains.com/webstorm/>
- Kozlowski S, Gully S, Nason E, Smith E. The changing nature of work performance: implications for staffing, personnel actions, and development. *SIOP Frontiers Series.* San Francisco: Jossey-Bass; 1999. p. 240–92.



- Malcolm D, Scott A. Professional relations in sport healthcare: workplace responses to organisational change. *Soc Sci Med.* 2011;72(4):513–20.
- McEwan D, Beauchamp MR. Teamwork in sport: a theoretical and integrative review. *Int Rev Sport Exerc Psychol.* 2014;7(1):229–50.
- McEwan D, Ruissen GR, Eys MA, Zumbo BD, Beauchamp MR. The effectiveness of teamwork training on teamwork behaviors and team performance: a systematic review and meta-analysis of controlled interventions. *PLoS ONE.* 2017;12(1):e0169604-e.
- Online shopping 20 years on: It started with the secure purchase of a Sting album...now 95% of British people buy goods via internet retailers. URL: <http://www.dailymail.co.uk/news/article-2722631/Online-shopping-20-years-It-started-securepurchase-Sting-album-95-British-people-buy-goods-internet-retailers.html>
- Porter CO. Goal orientation: effects on backing up behavior, performance, efficacy, and commitment in teams. *J Appl Psychol.* 2005;90(4):811.
- Porter COLH, Gogus CI, Yu RC-F. When does teamwork translate into improved team performance? A resource allocation perspective. *Small Group Res.* 2010;41(2):221–48.
- React. URL: <https://react.dev/>
- Schmutz JB, Meier LL, Manser T. How effective is teamwork really? The relationship between teamwork and performance in healthcare teams: a systematic review and meta-analysis. *BMJ Open.* 2019;9(9):e028280.
- Selenium. URL: <https://www.selenium.dev/>
- Shiflett S. Toward a general model of small group productivity. *Psychol Bull.* 1979;86(1):67.

- Smith ACT, Stewart B. The special features of sport: a critical revisit. Sport Manag Rev. 2010;13(1):1–13.
- The e-commerce market in Ukraine reached \$4 billion. URL: <https://soulpartners.com.ua/news/tpost/x2dve03v71-rinok-elektronno-komerts-v-ukrandosyagn>.
- Visual Studio Code. URL: <https://code.visualstudio.com/>
- Visual Studio. URL: <https://visualstudio.microsoft.com/>
- Vue.js. URL: <https://vuejs.org/>
- Webpack. URL: <https://webpack.js.org/>
- Zaccaro SJ, Rittman AL, Marks MA. Team leadership. Leadersh Q. 2001;12(4):451–83.