Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

**Лабораторна робота № 2**

з дисципліни «Основи клієнтської розробки»

Тема: «Футбол»

|  |  |
| --- | --- |
| Виконав:  студент групи ІК-34  Могильний Микола  Дата здачі 04.03.2025  Захищено з балом \_\_\_\_\_\_\_\_\_\_\_ | Перевірила:  ст. вик. кафедри ІСТ  Хмелюк Марина Сергіївна |

Київ 2025

**Завдання**

1. Встановити Git на комп’ютер. Створити каталог для проєкту.

Ініціалізувати Git директорію у своєму проєкті (git init).

2. Створити 4 сторінки відповідно до обраної тематики з розширенням

.html. Перша сторінка має назву - index.html. В вказати автора

документа. Кожна сторінка повинна мати назву . Кожна сторінка

повинна мати заголовок .

3. Додати створені сторінки під контроль Git на комп’ютері.

4. Зареєструватись на GitHub. Створити репозиторій. Завантажити проєкт

на віддалений репозиторій (на GitHub).

**Тематика**

27) Футбол

**Теоретичні відомості**

Системи контролю версіями (Version Control Systems, VCS) важливий

аспект розробки сучасного ПЗ. VCS надає такі можливості:

✓ підтримку зберігання файлів у репозиторії;

✓ підтримку історії версій файлів у репозиторії;

✓ знаходження конфліктів при зміні вихідного коду та забезпечення

синхронізації при роботі в багатокористувацькому середовищі

розробки;

✓ відстеження змін авторів.

**Звичайний цикл роботи розробника з СКВ виглядає так**:

- Оновлення робочої копії. Розробник виконує операцію оновлення

робочої копії з сервера на скільки можливо

- Модифікація проекту. Розробник локально модифікує проект,

змінюючи файли, що входять до нього, в робочій копії.

- Фіксація змін. Завершивши черговий етап роботи, розробник фіксує

свої зміни, передаючи їх на сервер. Перед використанням VCS може

вимагати від розробника оновлення.

**Розподілені системи контролю версій (Git) Переваги:**

- Так як кожного разу, коли клієнт забирає свіжу версію файлів, він

створює собі повну копію всіх даних, то у разі збоїв на сервері, через

який йшла робота, будь-який клієнтський репозиторій може бути

скопійовано назад на сервер, щоб відновити базу даних.

- Можливість працювати з кількома віддаленими репозиторіями. Таким

чином, можна одночасно працювати по-різному з різними групами

людей у рамках одного проекту.

Git має три основні стани, в яких можуть знаходитися ваші файли:

**зафіксовано, змінено і підготовлено**.

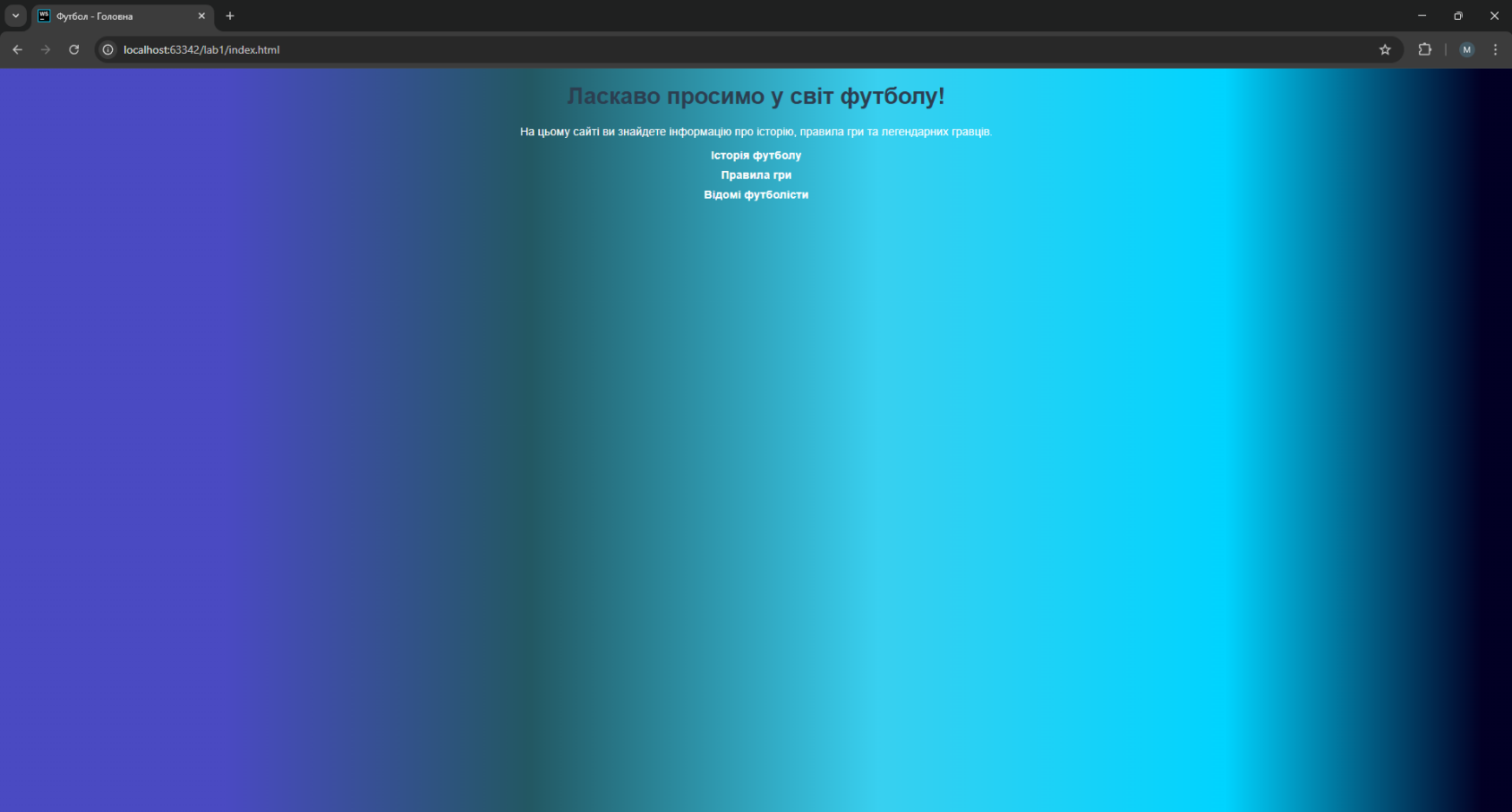
- Зафіксовано (Підтверджено): дані безпечно зберігаються локально.

- Змінено: ще не підтверджено.

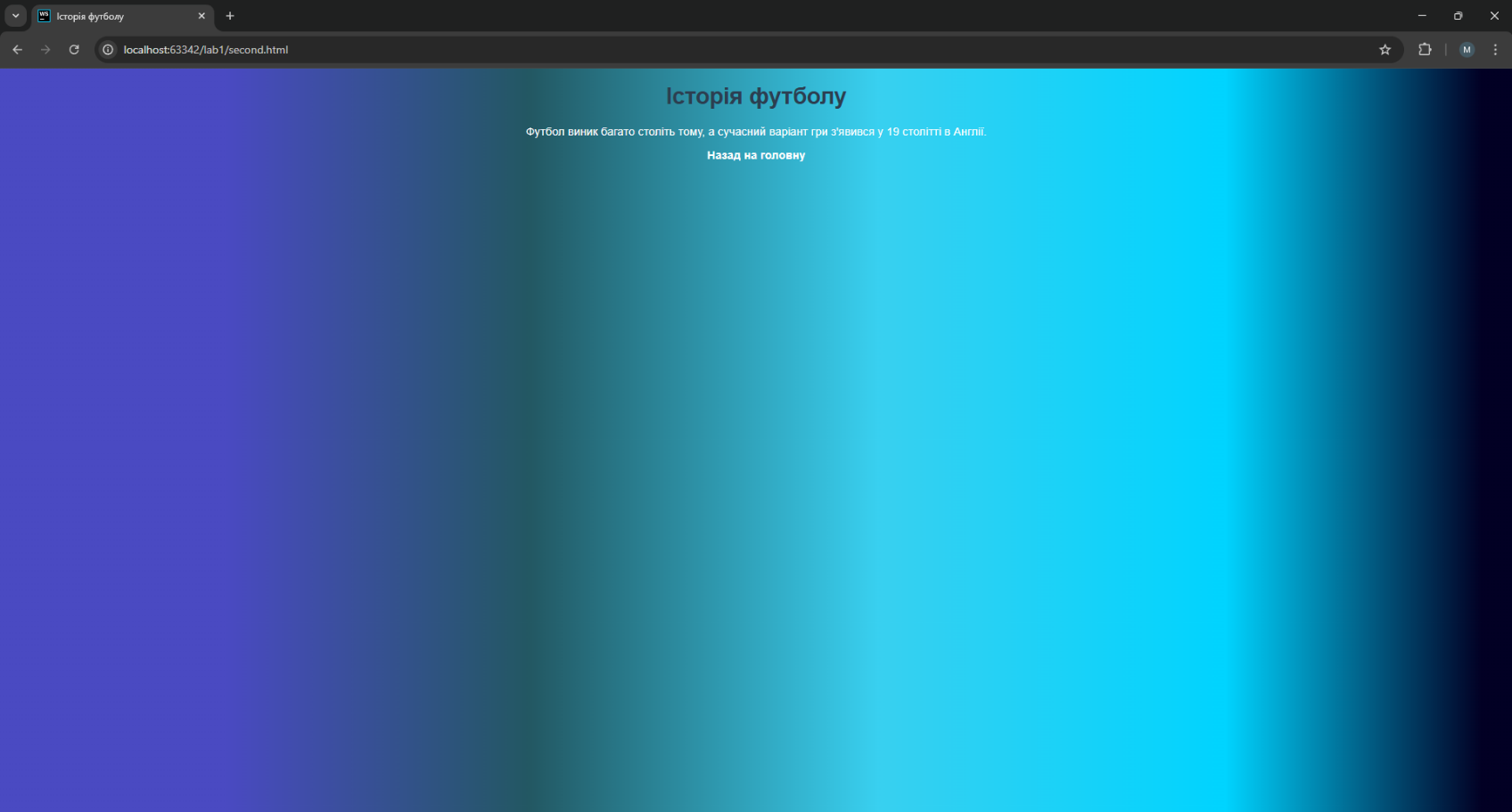
- Підготовлено: позначений для переходу до наступного коміту

**Створення сторінок**

Створюємо файл з назвою index.html, що буде головною сторінкою

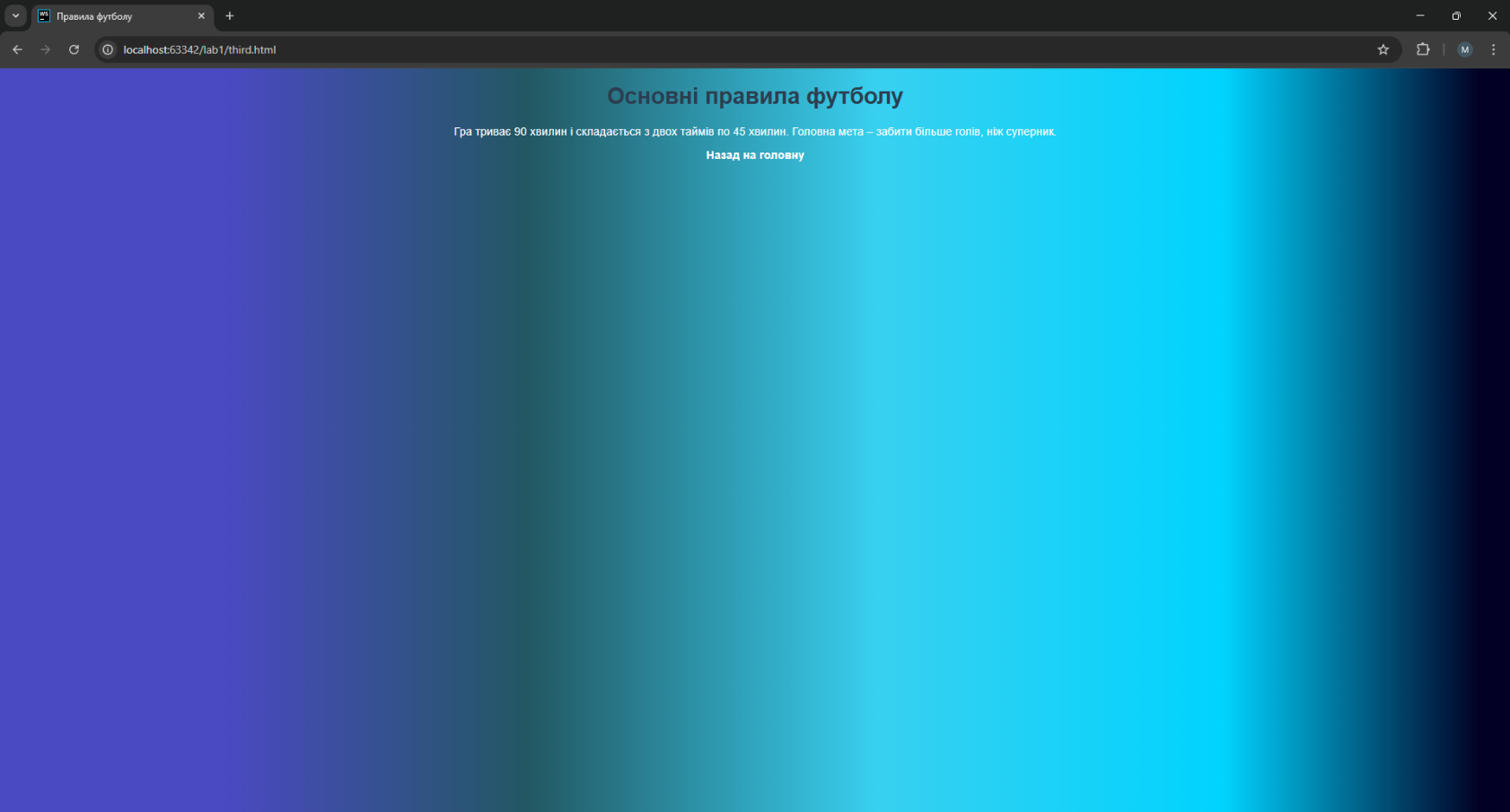
<!DOCTYPE html>  
<html lang="uk">  
<head>  
 <meta charset="UTF-8">  
 <meta name="author">  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <link rel="stylesheet" href="style.css">  
 <title>Футбол - Головна</title>  
</head>  
<body>  
<h1>Ласкаво просимо у світ футболу!</h1>  
<p>На цьому сайті ви знайдете інформацію про історію, правила гри та легендарних гравців.</p>  
<ul>  
 <li><a href="second.html">Історія футболу</a></li>  
 <li><a href="third.html">Правила гри</a></li>  
 <li><a href="fourth.html">Відомі футболісти</a></li>  
</ul>  
</body>  
</html>

Друга сторінка:

<!DOCTYPE html>  
<html lang="uk">  
<head>  
 <meta charset="UTF-8">  
 <meta name="author">  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <link rel="stylesheet" href="style.css">  
 <title>Історія футболу</title>  
</head>  
<body>  
<h1>Історія футболу</h1>  
<p>Футбол виник багато століть тому, а сучасний варіант гри з'явився у 19 столітті в Англії.</p>  
<a href="index.html">Назад на головну</a>  
</body>  
</html>

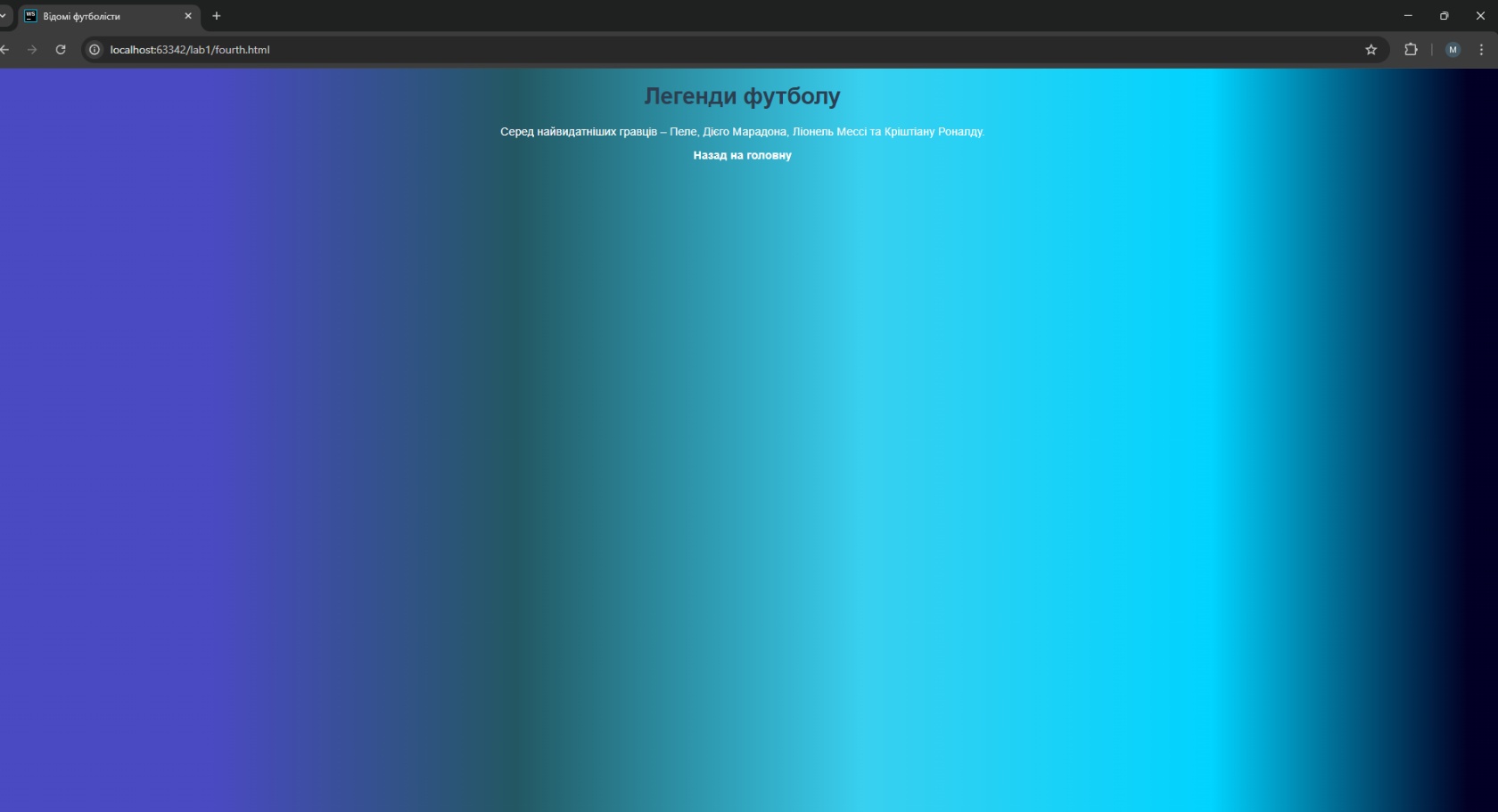
Третя сторінка:

<!DOCTYPE html>  
<html lang="uk">  
<head>  
 <meta charset="UTF-8">  
 <meta name="author">  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <link rel="stylesheet" href="style.css">  
 <title>Правила футболу</title>  
</head>  
<body>  
<h1>Основні правила футболу</h1>  
<p>Гра триває 90 хвилин і складається з двох таймів по 45 хвилин. Головна мета – забити більше голів, ніж суперник.</p>  
<a href="index.html">Назад на головну</a>  
</body>  
</html>



Четверта сторінка:

<!DOCTYPE html>  
<html lang="uk">  
<head>  
 <meta charset="UTF-8">  
 <meta name="author">  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <link rel="stylesheet" href="style.css">  
 <title>Відомі футболісти</title>  
</head>  
<body>  
<h1>Легенди футболу</h1>  
<p>Серед найвидатніших гравців – Пеле, Дієго Марадона, Ліонель Мессі та Кріштіану Роналду.</p>  
<a href="index.html">Назад на головну</a>  
</body>  
</html>

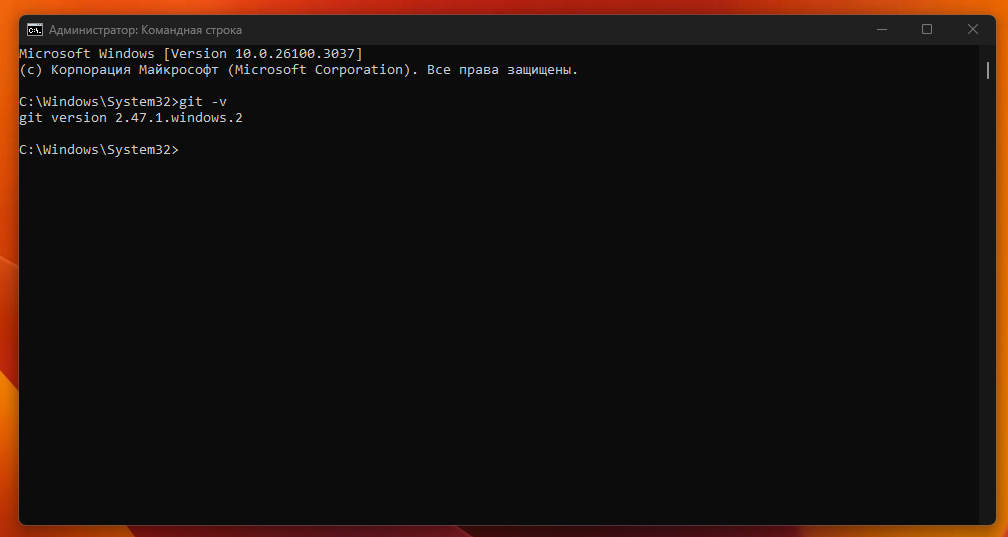


І останнє CSS:

body {  
 font-family: Arial, sans-serif;  
 margin: 0;  
 padding: 0;  
 color: white;  
 background: rgb(74,74,194);  
 background: linear-gradient(90deg, rgba(74,74,194,1) 15%, rgba(35,88,99,1) 35%, rgba(57,208,240,1) 58%, rgba(0,212,255,1) 81%, rgba(2,0,36,1) 98%);  
 text-align: center;  
}  
.container {  
 width: 80%;  
 max-width: 800px;  
 margin: 50px auto;  
 background: white;  
 padding: 20px;  
 border-radius: 10px;  
 box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);  
}  
h1 {  
 color: #2c3e50;  
}  
a {  
 color: white;  
 text-decoration: none;  
 font-weight: bold;  
}  
a:hover {  
 text-decoration: none;  
}  
ul {  
 list-style: none;  
 padding: 0;  
}  
ul li {  
 margin: 10px 0;  
}  
button {  
 background-color: #3498db;  
 color: white;  
 border: none;  
 padding: 10px 15px;  
 font-size: 16px;  
 cursor: pointer;  
 border-radius: 5px;  
 transition: background 0.3s;  
}  
button:hover {  
 background-color: #2980b9;  
}

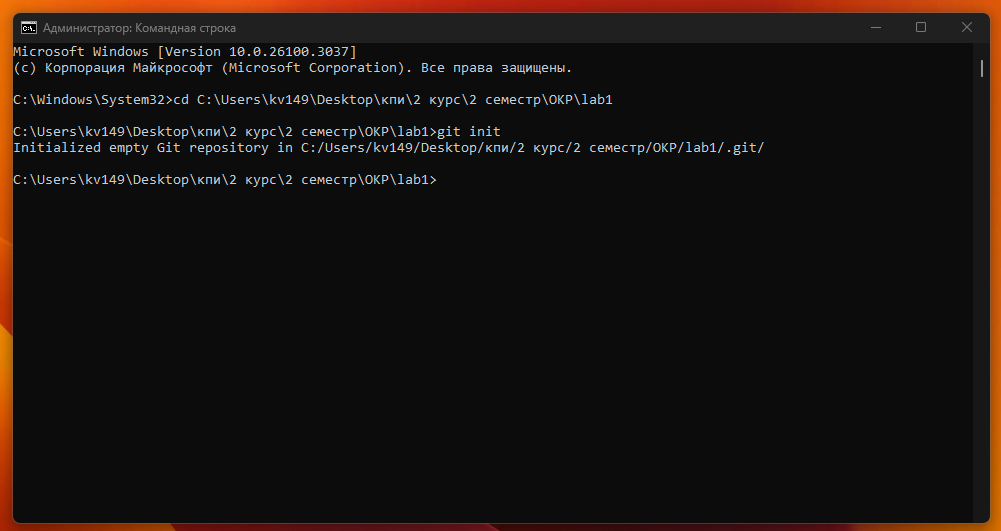
**Встановлення Git**

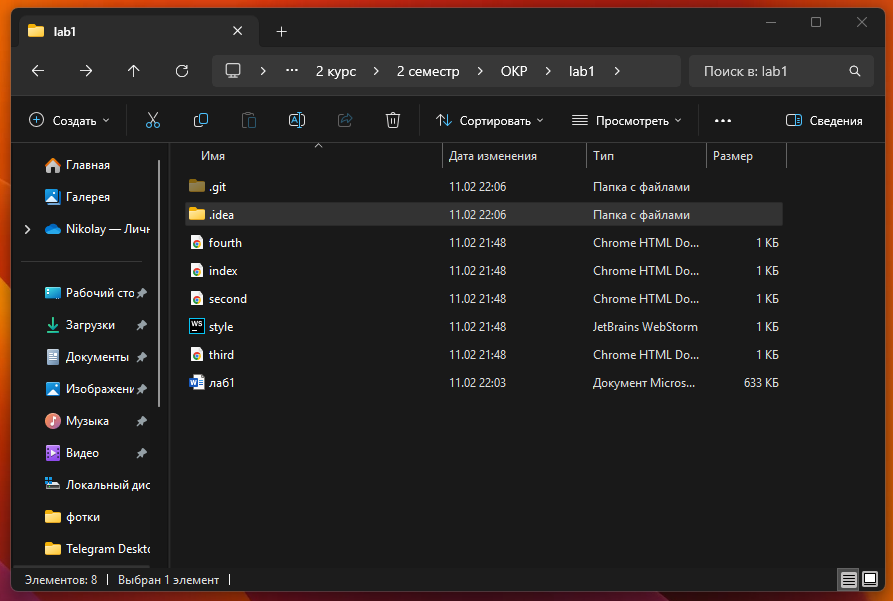
Оскільки Git вже було встановлено на комп’ютер раніше, тому просто перевіримо його версію через командну стрічку

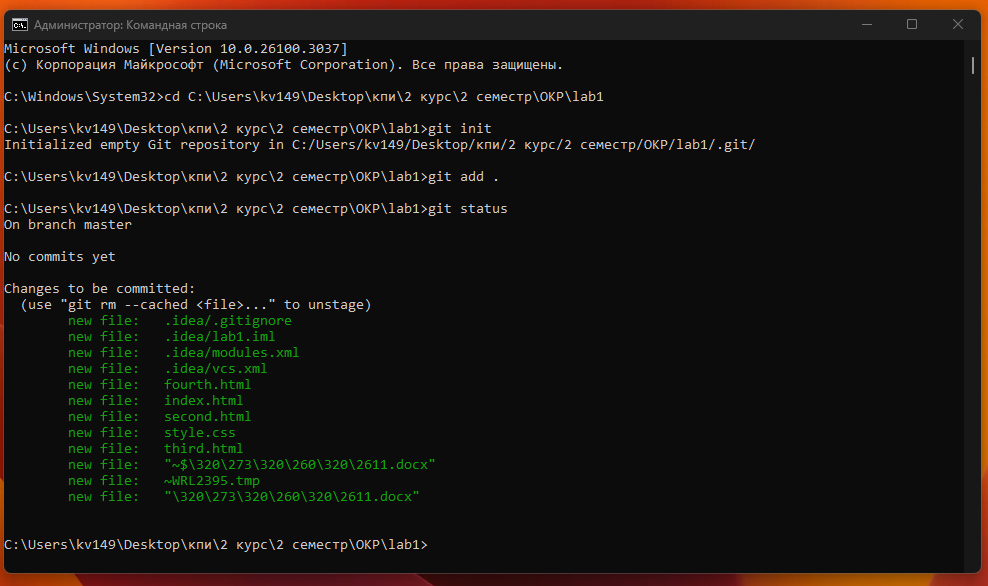


Перед початком роботи, потрібно проініціалізувати Git директорію у своєму

проєкті. Це робиться за допомогою команди: git init і маємо приховану папку







Спочатку перевіряємо стан командою git status та бачимо, що комітів нема,

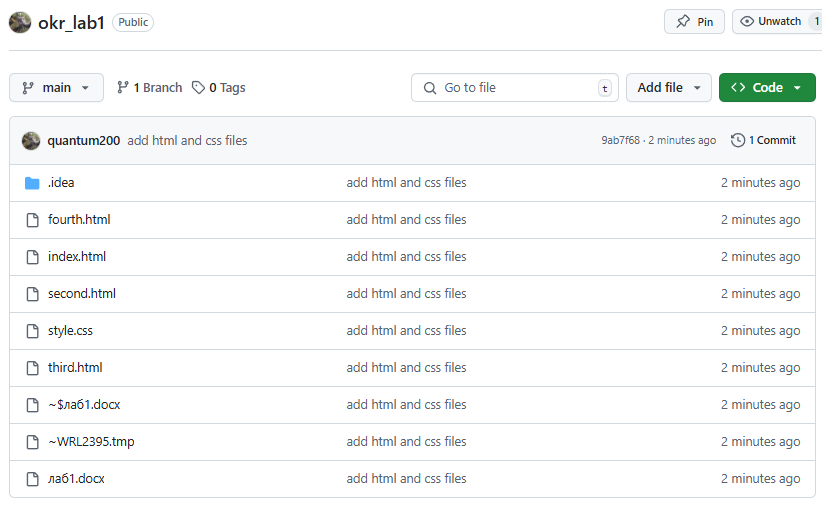
далі за допомогою команди git add . додаємо файли “.” означає, що додаємо

всі файли, а не кожен окремо). Далі додаємо коміт командою git commit та

додаємо повідомлення -m. Після цього публікуємо зміни в репозиторій

Гітхабу за допомогою команди git push -u origin main до гілки мейн та

перевіряємо наявність файлів у гітхабі.



**Висновок**

У ході цієї лабораторної роботи ми познайомились з системою управління версіями Git, створили репозиторій на віддаленому сервері Github та опублікували 4 сторінки, написані на HTML.