

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЧЕРНІВЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ЮРІЯ ФЕДЬКОВИЧА**

**Факультет математики та інформатики  
кафедра математичного моделювання**

**Розробка шаблону сайту засобами FlexBox, CSS Grid та їх порівняльна  
характеристика**

**Кваліфікаційна робота  
Рівень вищої освіти – перша(бакалаврський)**

**Виконав:**

**Студент 4 курсу, 407 групи**

**Гусаков Герман Валерійович**

**Керівник:**

**Доцент, канд. фіз.-мат. наук**

**Горбатенко Микола Юрійович**

**До захисту допущено на засіданні кафедри  
протокол № 16 від 30 травня 2023 р  
Зав. кафедрою \_\_\_\_\_ проф. Черевко І.М.**

**Чернівці  
2023**

## **Анотація**

Дана кваліфікаційна робота присвячена дослідженню та побудові шаблонів сайту засобами CSS Grid та FlexBox.

У цій кваліфікаційній роботі були розглянуті такі основні положення: принципи роботи CSS Grid, FlexBox, основи макету сайту, основи створення сайтів.

Порівняльна характеристика макетів створених засобами CSS Grid та FlexBox, визначення кращої технології.

Результати даної роботи можуть бути використані у таких галузях: бізнес, освіта, маркетинг та інші.

Кількість ілюстрацій – 26.

Кількість використаних джерел – 14.

Обсяг роботи - 39 сторінок.

Ключові слова: GRID CSS, FLEXBOX, CSS, HTML, JS, ВЕБ-ДИЗАЙН.

## **Annotation**

This qualification work is devoted to the research and construction of site templates using CSS Grid and FlexBox.

The following basic provisions were considered in this qualification work: the principles of CSS Grid, FlexBox, the basics of site layout, the basics of creating sites.

Comparative characteristics of layouts created by means of CSS Grid and FlexBox, determination of the best technology.

The results of this work can be used in the following fields: business, education, marketing and others.

The number of illustrations is 26.

The number of used sources is 14.

The volume of work is 39 pages.

Keywords: GRID CSS, FLEXBOX, CSS, HTML, JS, WEB DESIGN.

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів наукових досліджень інших авторів мають посилання на відповідне джерело.

\_\_\_\_\_ Г. В. Гусаков  
(підпис)

## Зміст

Вступ .....	6
<b>РОЗДІЛ 1. Поняття та розробка веб-сайту .....</b>	<b>7</b>
<b>1.1 Поняття сайту .....</b>	<b>7</b>
<b>1.2 Етапи створення веб-сайту .....</b>	<b>8</b>
<b>1.2.1 Аналіз та планування.....</b>	<b>8</b>
<b>1.2.2 Розробка концепції дизайну.....</b>	<b>8</b>
<b>1.2.2.1 Кольорова палітра .....</b>	<b>9</b>
<b>1.2.2.2 Шрифт .....</b>	<b>10</b>
<b>1.2.2.3 Графічні елементи .....</b>	<b>10</b>
<b>1.2.2.4 Адаптивний дизайн .....</b>	<b>11</b>
<b>1.2.3 Розробка контенту .....</b>	<b>11</b>
<b>1.2.4 Використання технологій.....</b>	<b>11</b>
<b>1.2.4.1 HTML .....</b>	<b>11</b>
<b>1.2.4.2 CSS .....</b>	<b>12</b>
<b>1.2.4.3 Flexbox .....</b>	<b>12</b>
<b>1.2.4.4 Grid CSS .....</b>	<b>13</b>
<b>1.2.4.5 JavaScript .....</b>	<b>14</b>
<b>1.2.4.6 Backend технології .....</b>	<b>15</b>
<b>1.2.5 Тестування сайту .....</b>	<b>16</b>
<b>1.2.6 Розміщення веб-сайту на хостинг .....</b>	<b>16</b>
<b>1.2.7 Підтримка та оновлення.....</b>	<b>18</b>
<b>1.3 Основні переваги та недоліки використання FlexBox та Grid CSS. ....</b>	<b>19</b>
<b>1.3.1 Простота використання.....</b>	<b>19</b>
<b>1.3.2 Переваги CSS Grid.....</b>	<b>20</b>
<b>1.3.3 Переваги FlexBox .....</b>	<b>20</b>
<b>1.3.4 Висока сумісність.....</b>	<b>21</b>
<b>1.3.5 Недоліки FlexBox та Grid CSS .....</b>	<b>21</b>
<b>РОЗДІЛ 2. ВИКОРИСТАНІ ТЕХНОЛОГІЇ.....</b>	<b>23</b>
<b>2.1 CSS .....</b>	<b>23</b>
<b>2.2 CSS grid .....</b>	<b>24</b>

<b>2.3 Flexbox</b> .....	24
<b>2.4 HTML</b> .....	25
<b>2.5 JavaScript</b> .....	26
<b>2.6 Google Fonts</b> .....	28
<b>2.7 Visual Studio code</b> .....	29
<b>2.8 Github desktop</b> .....	29
<b>2.9 Github pages</b> .....	30
<b>РОЗДІЛ 3. Опис, часткова демонстрація сторінок і порівняльна характеристика макетів CSS Grid та FlexBox.</b> .....	31
<b>3.1 Розроблені сторінки макетів</b> .....	31
<b>3.2 Порівняльна характеристика</b> .....	39
<b>ВИСНОВКИ</b> .....	44
<b>СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ</b> .....	45

## Вступ

В останні роки можна спостерігати значні зміни в Інтернеті. Сайти, які раніше були платформою для розташування статичного контенту, стали багатофункціональними, інтерактивними системами для надання різної інформації.

Актуальність теми дослідження зумовлена необхідністю аналізу існуючих методів розробки для визначення найбільш ефективних шляхів створення корпоративних та інформаційних ресурсів. Метою дослідження є виявлення ефективних методів розробки для правильної побудови бізнесу та інформаційних ресурсів в Інтернеті. Методи створення веб-сайтів включають різноманітні техніки та інструменти розробки. Цей проект має на меті детально розглянути та порівняти можливості, синтаксис та ефективність двох технологій. Порівняльний аналіз дозволить встановити, коли і в яких випадках краще використовувати flexbox або CSS grid, а також визначити їх переваги та недоліки.

В рамках цієї роботи буде розроблено шаблон сайту, демонструючи потужність та гнучкість обох технологій. Застосування цих інструментів дозволить створити зручний та естетичний дизайн, забезпечуючи при цьому легкість утримання та розширення проекту.

Метою кваліфікаційної роботи є не лише створення шаблону сайту, але й глибоке розуміння технологій flexbox та CSS grid, їхньої порівняльної характеристики та оптимального використання.

## **РОЗДІЛ 1. Поняття та розробка веб-сайту**

### **1.1 Поняття сайту**

Веб-сайт (Web-site) - це колекція взаємопов'язаних веб-сторінок, доступних для широкого загального користування, які використовують одне й те ж доменне ім'я. Люди, групи, підприємства або організації можуть створювати та підтримувати веб-сайти з різних причин. Разом, всі загальнодоступні веб-сайти утворюють глобальну мережу.

Веб-сайт, як система електронних документів (файлів даних та коду) може належати приватній особі або організації та бути доступним у комп'ютерній мережі під загальним доменним ім'ям та IP-адресою або локально на одному комп'ютері.

Протокол HTTP був спеціально розроблений для забезпечення прямого доступу клієнтів до сайтів на серверах. Всі ці сайти разом створюють Всесвітню павутину, де комунікація об'єднує різні сегменти інформації світової спільноти в одну єдину базу даних та комунікації на планетарному масштабі.

Розвиток сучасних технологій у створенні веб-сайтів та нескінченні можливості людської уяви спричинили велике розмаїття веб-сайтів, які ми бачимо в Інтернеті під час використання пошукових систем, таких як Bing, Google та інші. Всі вони мають спільні характеристики, які дозволяють користувачеві легко розпізнати їх з точки зору функціональності. Ці характеристики включають головну сторінку (індексну), яку ми бачимо при вході на сайт, меню з посиланнями на основні сторінки або розділи, основний вміст (контент) та нижній колонтитул (футер або підвал) з важливими посиланнями, інформацією про авторські права та іншими відомостями.

Відвідувач може переходити з однієї веб-сторінки на іншу за допомогою гіперпосилань.

Веб-сайти створюються для швидкого та зручного доступу до потрібної

інформації. Інформацію на сторінках розміщують кваліфіковані спеціалісти або любителі у області, в якій створений сайт.

Крім HTML(Hyper Text Markup Language), для стилізації та розташування елементів на сторінці можна використовувати CSS (Cascading Style Sheets). CSS дозволяє задавати розміри, кольори, шрифти, відступи та інші властивості елементів.

## **1.2 Етапи створення веб-сайту**

Етапи створення веб-сайту відіграють дуже важливу роль у процесі розробки, організації та створенні сайту. Вони сприяють структуруванню проекту, забезпечують ефективну співпрацю між учасниками команди та допомагають досягти успішного завершення проекту.

### **1.2.1 Аналіз та планування**

Перший етап у створенні будь-якого веб-сайту - це проведення аналізу та планування. На цьому етапі необхідно визначити цільову аудиторію, мету сторінки, встановити основні елементи контенту, які будуть присутні на сторінці. Також важливо врахувати дизайн сторінки, вибрати кольорову палітру, шрифти та інші візуальні елементи, які відповідатимуть цілям та бренду компанії.

### **1.2.2 Розробка концепції дизайну**

Після аналізу та планування наступним кроком є розробка дизайну та розміщення елементів на сторінці. Дизайн сторінки повинен бути привабливим та відповідати бренду або продукту. Використання візуальних елементів, таких як кольори, шрифти та зображення, допоможе створити привабливий вигляд сторінки.

Розробка концепції дизайну – важливий крок у створенні посадкової сторінки. На даний момент проблеми з інтерфейсом, дизайном і візуальною



привабливістю сайту вирішено. Основна мета — створити дизайн, який ефективно доносить повідомлення до цільової аудиторії та повертає її увагу.

При розробці концепції планування посадки враховуються наступні аспекти:

### **1.2.2.1 Кольорова палітра**

Палітра кольорів веб-дизайну виконує багато важливих функцій у створенні красивого та функціонального веб-сайту. Її використання дає наступний результат:

Гарний зовнішній вигляд - кольори сайту впливають на його сприйняття користувачами. Це дозволяє гармонійно поєднувати та контрастувати, щоб привернути увагу та донести потрібну інформацію.

Брендинг - палітра кольорів може відтворювати кольори бренду компанії, забезпечувати послідовність і підкреслювати ідентичність бренду.

Стиль і атмосфера - кольори мають великий вплив на емоції та настрої користувачів. Залежно від стилю та повідомлення обраного вами веб-сайту, він може справляти враження енергії, спокою, радості чи гармонії.

Навігація та інтерактивність - колір використовується для підкреслення інтерактивних елементів, таких як посилання, кнопки та меню. Це сприяє зручності використання та сприйняттю функціональних елементів веб-сайту.

Читабельність і доступність - вибір кольору впливає на читабельність тексту на вашому веб-сайті. Важливо вибрати колір, який забезпечує достатній контраст між текстом і фоном, щоб він був розбірливим для всіх користувачів, включно з тими з особливими потребами зору.

Створить ідентифікатор - палітра кольорів допомагає вашому веб-сайту виділитися серед конкурентів і створити унікальну ідентичність. Вибір правильних кольорів може зробити ваш сайт більш запам'ятовуваним і впізнаваним.

Культурний вплив - кольори мають різне значення в різних культурах і географічних регіонах. Створюючи веб-сайт для глобальної аудиторії,

важливо враховувати культурні конотації кольору, щоб уникнути помилкових асоціацій і непорозумінь.

Психологічний вплив - колір має психологічний вплив на користувачів і може викликати різні емоції та реакції. Використовуючи переваги психологічного впливу кольору, ви можете досягти бажаних ефектів і вплинути на сприйняття користувачами вашого сайту.

### **1.2.2.2 Шрифт**

Вибір правильних шрифтів і їх розміщення на сторінці має велике значення для читабельності та візуального сприйняття контенту. Важливо підібрати шрифт, який відповідає стилю бренду та забезпечує читабельність тексту на різних пристроях. Використання візуальних елементів, таких як фотографії, ілюстрації, значки та діаграми, допомагає підкреслити повідомлення та зробити сторінку більш привабливою.

### **1.2.2.3 Графічні елементи**

Грамотне розміщення та підбір графічного контенту на веб-сайті мають велике значення для ефективності та привабливості сайту. Графічні елементи допомагають створити естетично привабливий вигляд та передати повідомлення користувачам. Кольорова палітра може відтворювати брендові кольори і створювати єдиний візуальний образ. Правильний підбір та розміщення графічного контенту сприяють поліпшенню користувацького досвіду. Вони допомагають організувати вміст сайту, полегшують навігацію та сприяють швидкому сприйняттю інформації.

Графічні елементи можуть виділяти важливі деталі та підкреслювати їх значення. Крім того, правильний підбір та розміщення графічного контенту сприяють створенню єдності візуального стилю на всьому веб-сайті. Це поліпшує сприйняття бренду та забезпечує консистентний вигляд. Також, графічний контент може впливати на SEO, покращуючи час перебування користувачів на сайті та забезпечуючи якісний контент.

#### **1.2.2.4 Адаптивний дизайн**

Враховуючи різні пристрої, на яких можна переглядати цільову сторінку (наприклад, комп'ютери, планшети, смартфони), для правильного відображення сторінки та її доступу на різних пристроях потрібен адаптивний дизайн. Розробка концепції дизайну вимагає взаємодії з клієнтом або командою проекту, обговорення їх бажань, вимог і бренду.

#### **1.2.3 Розробка контенту**

Третій етап полягає в розробці контенту для сторінки. Це включає в собі: написання привітального тексту, опису продукту, інформації яка буде подана, або послуги, використання візуальних елементів, таких як зображення або відео, які допоможуть привернути увагу відвідувачів. Важливо підготувати контент, який буде коротким, зрозумілим та переконливим.

#### **1.2.4 Використання технологій**

Для створення простої сторінки можна використовувати різні технології, що допомагатимуть забезпечити відповідну функціональність та вигляд сторінки. Однією з ключових технологій є HTML (Hyper Text Markup Language), яка використовується для створення структури сторінки та розміщення контенту. За допомогою HTML можна створити заголовки, абзаци, списки, зображення та інші елементи

##### **1.2.4.1 HTML**

HTML є основою веб-сторінок і використовується для створення їх структури та визначення різних елементів на сторінці. Він складається з тегів, які вказують браузеру, як відображати різні елементи, такі як заголовки, абзаци, списки, зображення та посилання. HTML також дозволяє включати різні метадані, такі як заголовок сторінки, мета-теги, що впливають на пошукову оптимізацію та інші аспекти сторінки.

При розробці сторінки HTML використовується для створення основної структури сторінки, включаючи розмітку контенту, встановлення заголовків, створення посилань та вбудовування мультимедійних елементів.

#### **1.2.4.2 CSS**

CSS використовується для стилізації та оформлення веб-сторінок, надаючи їм вигляд, кольори, розміри, шрифти та інші візуальні аспекти. Він працює в парі з HTML і визначає, які стилі застосовувати до різних елементів на сторінці. CSS дозволяє зручно керувати виглядом елементів, встановлюючи правила для конкретних селекторів. Наприклад, ви можете задати кольори фону, тексту та посилань, встановити шрифти та їх розміри, налаштувати відступи, рамки та інші властивості елементів. При створенні сторінки CSS використовується для стилізації контенту, забезпечуючи привабливий та консистентний вигляд. Він допомагає створити привабливі заголовки, стильні кнопки, просторі між-елементні відступи та інші візуальні ефекти.

#### **1.2.4.3 Flexbox**

Flexbox є технологією CSS, яка дозволяє гнучко розташовувати елементи в одновимірному просторі. Вона дозволяє контролювати розміщення, порядок та відстань між елементами.

Flexbox використовує гнучку модель лейауту, де контейнер (родич) визначається як flex-контейнер, а дочірні елементи - як Flex-елементи. За допомогою властивостей Flexbox, таких як `display`, `Flex-Direction`, `Justify-Content`, `Align-Items`, `Flex-Grow` та інші, можна встановити бажану конфігурацію розташування елементів.

При створенні сторінки Flexbox може бути використаний для розміщення елементів у рядок або колонку, вирівнювання їх по головній або поперечній вісі, розподілу доступного простору між елементами та управління їх поведінкою при розтягуванні екрана.

Гнучкий макет FlexBox пропонує більш динамічний метод організації та розміщення елементів на веб-сайті. З точки зору розміщення та розміру, він пропонує більшу гнучкість елементів. Веб-сайти, яким потрібен виразний дизайн або більш креативний підхід, повинні використовувати FlexBox. FlexBox дозволяє дизайнерам проявити свою творчість і створити унікальний веб-сайт.

FlexBox краще підходить для більш винахідливих і естетично складних веб-сайтів, Grid CSS найкраще підходить для структурованих і насичених вмістом веб-сайтів. FlexBox дає більше контролю над вирівнюванням і розподілом простору між елементами завдяки своїй розмірності та гнучкості. Він працює зі стовпцями або рядками, що робить його ідеальним для невеликих макетів, як-от текст або властивості документа, які використовують стовпці та плаваючі елементи. І навпаки, сітка підтримує двовимірний макет і пропонує гнучку ширину як одиницю довжини. Це подолає обмеження FlexBox.

#### **1.2.4.4 Grid CSS**

Grid CSS - це технологія CSS, яка надає можливість створювати багатовимірні сітки для розміщення елементів на сторінці в двовимірному просторі. Вона дає більш гнучкий та потужний спосіб контролювати розташування елементів порівняно зі стандартними блочними або лінійними моделями макету. Grid CSS дозволяє визначати рядки та колонки у сітці та встановлювати їх розміри, пропорції та розташування. За допомогою властивостей, таких як `grid-template-columns`, `grid-template-rows`, `grid-gap`, `grid-area` та інші, можна точно контролювати розташування елементів у сітці. При створенні сторінки Grid CSS може бути використаний для створення рівномірної сітки, розміщення елементів у відповідних рядках та колонках, підлаштування їх розмірів під контент, а також для створення адаптивного макету, що пристосовується до різних розмірів екрана.

Макет CSS Grid базується на сегментуванні веб-сайту на стовпці та рядки для створення структурованого макета. Завдяки цьому дизайнери можуть чітко та послідовно вирівнювати елементи веб-сайту. Для веб-сайтів із великою кількістю вмісту та інформації, які потрібно організувати логічно, сітка особливо корисна.

#### **1.2.4.5 JavaScript**

JavaScript є мовою програмування, яка часто використовується для динамічної інтерактивності на веб-сайтах. Вона дозволяє додавати функціональність, взаємодію з користувачем, анімацію та інші динамічні ефекти до статичних HTML-сторінок. При створенні сторінки JavaScript може бути використаний для реалізації різноманітних функцій, таких як валідація форм, зміна вмісту сторінки в залежності від дій користувача, візуальні ефекти, анімація, запуск звуків та багато іншого. JavaScript дає можливість маніпулювати HTML-документом, взаємодіяти з його елементами, обробляти події, виконувати асинхронні запити на сервер, зберігати дані в локальному сховищі та багато іншого. Це робить його потужним інструментом для створення динамічних та інтерактивних веб-сайтів.

За допомогою JavaScript можна також інтегрувати сторонні бібліотеки та фреймворки, які розширюють можливості мови та спрощують розробку веб-сайтів. Наприклад, популярні фреймворки, такі як React, Angular та Vue.js, дозволяють побудувати складні динамічні інтерфейси та забезпечують ефективну організацію коду.

Використання JavaScript разом з CSS та HTML дозволяє створити веб-сторінку, яка не тільки виглядає привабливою, але й має розширену функціональність та здатність реагувати на дії користувача, що забезпечує більш залучену та зручну взаємодію з веб-сайтом.

#### 1.2.4.6 Backend технології

Backend-розробка веб-сайту - це складний процес, який вимагає використання різних технологій та компонентів для створення потужного та надійного функціоналу. Він включає в себе використання мов програмування, фреймворків, баз даних, API, серверів та інфраструктури.

Мови програмування, такі як PHP, Python, Ruby, Java і C#, використовуються для написання серверного коду та обробки запитів користувачів. Вони надають потужність та гнучкість для розробки різних функцій і логіки, необхідних для веб-сайту.

Фреймворки, такі як Laravel для PHP, Django для Python, Ruby on Rails для Ruby, Spring для Java, ASP.NET для C#, є набором готових бібліотек, інструментів та стандартів, які спрощують процес розробки backend-функціоналу. Вони дозволяють використовувати передові практики та рекомендації, що сприяє швидкому та ефективному розробці коду.

Бази даних, такі як MySQL, PostgreSQL, MongoDB та SQLite, використовуються для зберігання та організації даних на веб-сайті. Вони дозволяють ефективно та безпечно працювати з даними, виконувати складні запити та забезпечувати цілісність даних.

API (Application Programming Interface) використовується для взаємодії з іншими додатками або сервісами. Вони дозволяють розширювати функціональність веб-сайту, обмінюватися даними і інтегрувати зовнішні ресурси. REST або GraphQL API надають стандартні протоколи та методи для взаємодії з іншими системами.

Сервери та інфраструктура грають важливу роль у роботі веб-сайту. Вони можуть використовувати різні технології, такі як Apache, Nginx або Microsoft IIS, і додаткові інструменти, такі як контейнери, хмарні сервіси або сховища даних. Вони забезпечують швидкість, масштабованість та надійність веб-сайту.

Від правильного вибору мов програмування, фреймворків, баз даних,

API та налаштування серверів залежить ефективність та швидкодія веб-сайту, зручність взаємодії з користувачами та інтеграція з іншими системами.

### **1.2.5 Тестування сайту**

Тестування веб-сайту є невід'ємною складовою процесу його створення. Ця діяльність має вирішальне значення для забезпечення якості та оптимального функціонування веб-сайту.

Перш за все, тестування веб-сайту є необхідним для перевірки правильності роботи всіх його елементів та функцій перед запуском. Це дозволяє виявити та усунути можливі помилки та недоліки, які можуть впливати на коректність роботи сайту та задоволення користувачів. Додатково, тестування спрямоване на забезпечення високої якості користувацького досвіду. Шляхом перевірки навігації, швидкості завантаження сторінок, взаємодії з елементами сайту та його сумісності з різними пристроями і браузерами, можна забезпечити зручність та задоволення від використання сайту.

Тестування веб-сайту також допомагає гарантувати безпеку і захист інформації користувачів. Шляхом перевірки на вразливості та потенційні загрози безпеці, можна вжити заходів для запобігання злому, крадіжці даних або несанкціонованому доступу. Крім того, тестування веб-сайту є важливим для забезпечення сумісності з різними пристроями та платформами. У світі, де веб-сайти відкриваються на різних пристроях, таких як комп'ютери, смартфони, планшети, тестування допомагає впевнитися, що сайт відображається належним чином та працює ефективно на будь-якому з них.

### **1.2.6 Розміщення веб-сайту на хостинг**

Розміщення веб-сайту на хостинг є важливим етапом в процесі його створення та розвитку. Цей процес включає кілька кроків, які потрібно виконати для того, щоб зробити веб-сайт доступним для всього Інтернету.

Перш за все, необхідно вибрати хостинг-провайдера, який надасть



простір на своїх серверах для зберігання веб-сайту. При виборі хостинг-провайдера важливо врахувати такі фактори, як надійність, швидкість, безпека, підтримка та ціна.

Після вибору хостинг-провайдера необхідно зареєструвати доменне ім'я веб-сайту. Доменне ім'я є унікальною адресою, за допомогою якої користувачі зможуть знаходити веб-сайт в мережі. Вибір доменного імені важливий, оскільки він повинен бути легким для запам'ятовування та відображати сутність проекту.

Після того, як доменне ім'я зареєстровано, необхідно налаштувати DNS-записи. Це дозволить пов'язати доменне ім'я з IP-адресою сервера хостинг-провайдера, на якому буде розміщений сам веб-сайт. Цей процес може зайняти деякий час для пропагації DNS-записів.

Далі потрібно завантажити всі необхідні файли веб-сайту на сервер хостинг-провайдера. Це включає HTML-сторінки, CSS-стилі, зображення, скрипти та інші ресурси, які є структурою усього веб-сайту. Зазвичай для цього використовуються FTP-клієнти або панель керування хостингом, яка надається хостинг-провайдером на весь час аренды хостингу.

Після завантаження файлів веб-сайту необхідно перевірити його працездатність та взаємодію з сервером. Це можна зробити, відкривши веб-сайт у веб-браузері та переконавшись, що всі сторінки та функціональність працюють належним чином.

Останнім кроком є забезпечення безпеки веб-сайту. Це включає налаштування захисту від несанкціонованого доступу, резервне копіювання файлів та бази даних, встановлення сертифікату безпеки SSL для захищеного з'єднання та регулярне оновлення веб-додатків і плагінів.

В результаті успішного розміщення веб-сайту на хостинг він стає доступним для відвідувачів з усього світу через інтернет. Хостинг забезпечує надійне зберігання файлів вашого веб-сайту та його безперебійну роботу 24/7. Цей етап створення веб-сайту дозволяє перетворити ваші ідеї та дизайн у функціональну і доступну для користувачів онлайн-присутність.

### **1.2.7 Підтримка та оновлення**

Забезпечення підтримки та оновлення веб-сайту є необхідною складовою його процесу створення та функціонування. Під час розробки веб-сайту використовуються різні технології, мови програмування та фреймворки, які постійно змінюються та вдосконалюються.

Оновлення веб-сайту до нових версій технологій є необхідним для забезпечення безперебійної роботи сайту та його відповідності сучасним стандартам. Одним з аспектів підтримки є оновлення мов програмування та фреймворків, які використовуються для розробки backend-частини сайту. Розробники постійно слідкують за випуском нових версій цих мов та фреймворків, які містять виправлення помилок, покращення безпеки та нові функціональні можливості. Оновлення до нових версій допомагають забезпечити високу безпеку, ефективну роботу та сумісність сайту з оновленою технологічною базою.

Також важливо виконувати оновлення баз даних, які використовуються для зберігання інформації на веб-сайті. Ці бази даних регулярно оновлюються, щоб усунути помилки, покращити продуктивність та додати нові функціональні можливості. Оновлення баз даних забезпечують надійність, цілісність та ефективність роботи сайту. Однак, підтримка та оновлення веб-сайту не обмежуються лише мовами програмування та базами даних. Вона також включає оновлення систем безпеки, що включає в себе встановлення оновлень безпеки для серверного ПЗ, моніторинг захисту від потенційних вразливостей та запобігання злому сайту. Регулярне оновлення системи безпеки є важливим для забезпечення захищеності сайту від потенційних загроз та атак.

Оновлення веб-сайту до нових версій технологій допомагають забезпечити його оптимальну продуктивність, швидкодію та сумісність зі сучасними пристроями та браузерами. Вони також дозволяють використовувати нові функції та можливості, які поліпшують користувацький досвід та сприяють розвитку та конкурентоспроможності веб-сайту.

## 1.3 Основні переваги та недоліки використання FlexBox та Grid CSS

Використання Flex і Grid CSS веб-розробкою має ряд переваг, які роблять їх популярними серед розробників. Але є і деякі недоліки, які будуть розглянуті нижче.

### 1.3.1 Простота використання

FlexBox і Grid CSS мають зрозумілу та просту синтаксичну структуру, що дозволяє розробникам легко використовувати їх у своїх проектах. Ці технології надають інтуїтивно зрозумілі властивості та значення, що спрощує створення та керування розміщенням елементів на веб-сторінках.

На успіх сторінки значно впливає те, наскільки просто нею користуватися. Основна мета полягає в тому, щоб користувачам було просто та зручно переміщатися по сайту, швидко отримувати потрібну інформацію та виконувати потрібні дії, як-от робити покупки чи залишати свою контактну інформацію. Застосовуючи на практиці принцип простоти використання на цільовій сторінці, слід враховувати наступне:

Інтуїтивність - сторінка має бути простою у використанні та зрозумілою для відвідувачів. Розміщення навігаційних елементів, кнопок та інших ключових компонентів має спростити доступ користувачів до них і взаємодіяти з ними.

Чіткість і структура - вміст повинен бути організований логічно та мати чітку структуру на сторінці. Щоб привернути увагу користувачів, важливо привернути увагу до ключових компонентів, таких як заголовки, підзаголовки, повідомлення та привабливі пропозиції. Використання мінімалістичного дизайну робить сторінку чистішою та привабливішою, запобігаючи переповненню сторінки непотрібними елементами.

Важливо обмежити обсяг тексту, використовувати зрозумілу мову та підтримувати зрозумілу структуру сторінки.

Швидкодія – зазвичай користувачі хочуть, щоб сторінка була приємною оку, але одним із найважливіших пунктів є швидкість завантаження сторінки та контенту на ній.

### **1.3.2 Переваги CSS Grid**

Використання CSS Grid робить HTML код більш "чистим", оскільки спрощує розмітку і усуває потребу в багатьох класах і додаткових тегах.

CSS Grid найкраще підходить для створення 2D-макетів з використанням рядків та стовпців.

Так як Grid є частиною нативного CSS, немає ризику його поломки та необхідності перебудови проекту через деякий час. Він є стабільним і надійним. Grid значно спрощує процес побудови макетів, що раніше вимагав багато хаків і поліфілів.

Ідеально підходить для створення великих макетів, де потрібно керувати багатьма елементами. Надає можливість створювати сайти з динамічним контентом, де макет може змінюватися в залежності від змін у вмісті.

### **1.3.3 Переваги FlexBox**

Гнучкість елементів – flexbox дає можливість елементам легко змінювати свої розміри, стискаючись або розтягуючись залежно від заданих правил. Це дозволяє створювати гнучкі та адаптивні макети, де елементи займають необхідний простір.

Вирівнювання - Flexbox забезпечує зручні засоби для вирівнювання елементів по вертикалі та горизонталі, а також вирівнювання за базовою лінією тексту. Це дозволяє точно контролювати розташування елементів на сторінці.

Гнучкість розташування - flexbox може змінювати розташування елементів в HTML-структурі за допомогою CSS. Це дає розробникам більшу свободу при створенні веб-розмітки та полегшує роботу з аспектами верстки.

### **1.3.4 Висока сумісність**

FlexBox і Grid CSS є стандартними модулями CSS, які підтримуються більшістю сучасних веб-браузерів. Це означає, що розробники можуть використовувати ці технології безпосередньо на більшості платформ без необхідності в додаткових поліфіллах або обхідних рішеннях. Керованість та контроль FlexBox і Grid CSS надають розробникам велику кількість властивостей та налаштувань для керування розташуванням елементів на веб-сторінках. Розробники можуть контролювати розміри, відступи, вирівнювання, порядок розташування та інші параметри для кожного елемента на основі їх потреб і вимог дизайну.

Технології FlexBox і Grid CSS є потужними інструментами для створення гнучких та адаптивних макетів на веб-сторінках. Вони надають розробникам можливість легко контролювати розміщення елементів та створювати привабливі та ефективні макети для різних пристроїв і розмірів екранів. У наступних розділах дипломної роботи ми розглянемо деталі та приклади використання Flex і Grid CSS у створенні веб-сайтів.

### **1.3.5 Недоліки FlexBox та Grid CSS**

Основний недолік CSS Grid полягає у його складній синтаксичній структурі. При використанні CSS Grid потрібно докладніше визначити сітку, встановити розміри і положення елементів, що може бути вимогливим для новачків у веб-розробці. Крім того, деякі старі браузери можуть не підтримувати всі функції CSS Grid, що потребує додаткових поліфілів або альтернативних підходів для забезпечення сумісності з усіма браузерами.

Один з найбільших недоліків Flexbox - це відсутність двовимірної сітки. Він призначений для розташування елементів у рядки або стовпці, але не забезпечує таку гнучкість, як CSS Grid для складних макетів з багатьма рядками та стовпцями. Крім того, Flexbox може викликати проблеми з вирівнюванням елементів, особливо якщо їх розміри неоднакові або змінюються динамічно.

Незважаючи на ці недоліки, які можуть потребувати додаткових зусиль і уваги, CSS Grid та Flexbox залишаються потужними інструментами для розробки гнучких та респонсивних веб-інтерфейсів. Вірне використання цих технологій залежить від специфічних потреб проекту та підтримки браузерів, яка вимагає тестування та альтернативних рішень для забезпечення сумісності.

## РОЗДІЛ 2. ВИКОРИСТАНІ ТЕХНОЛОГІЇ

### 2.1 CSS

Використання CSS дозволило стилізувати та анімувати елементи, надаючи їм привабливий зовнішній вигляд, тому були використані наступні правила:

- Height - висота блока, до якого застосовується;
- Font-Weight - альтернативний спосіб задання розміру для шрифту;
- Text-Transform: Uppercase - трансформує літери до верхнього регістру;
- Font-Size - задання розміру шрифту у пікселях, використовується частіше у порівнянні з Font-Weight;
- Color - задає колір будь-чого, від фону до тексту;
- Text-Align - розміщення тексту на сторінці(ліва границя сайту/блоку, центр, права границя сайту/блоку) ;
- Background-Image - використовується щоб помістити будь-яке зображення на фон сторінки, або блоку, якщо прописати його шлях;
- Margin-Top - саме ця властивість робить відступ від верхньої границі; Margin відступи від зовнішніх границь у різних одиницях виміру;
- Padding - робить внутрішні відступи від границь елемента до його вмісту;
- Border-Radius - закруглення кутів будь-якого блоку;
- Z-Index - визначає порядок стеку елементів, тобто можна розміщати елементи поверх один одного, змінюючи індекс;
- Line-Height - відступ між рядками;
- Float - вирівнює елемент по стороні сторінки, і визначає з якої сторони буде обтікання елемента;
- Opacity - прозорість елемента, зазвичай використовується при створенні анімації тексту пр наведенні;
- Width - ширина елемента або блок;

- Background-Color - задає фоновий колір блоку або елемента;

## 2.2 CSS grid

CSS grid надає нові можливості та дозволяє розташовувати елементи одночасно враховувати горизонтальний і вертикальний вільний простір без використання media-запитів, що значно спрощує створення адаптивних веб-сайтів.

Під час виконання кваліфікаційної роботи використовувались такі властивості CSS grid:

- Grid-Area – за допомогою якого я задавав назву кожному grid блоку, щоб далі правильно їх розміщати.
- Grid-Template-Columns – ця властивість задає кількість та розміри колонок у Grid сітці.
- Grid-Template-Rows – робить те саме, що і попередня тільки замість стовпців визначає розмір і кількість рядків сітки.
- Grid-Template-Areas – за допомогою цієї властивості я розміщав Grid блоки на веб-сайті за назвами, які я задавав у Grid-Area.
- Grid-Gap – задає відступи між grid блоками.

## 2.3 Flexbox

CSS модуль розкладки Flexible Box, зазвичай його називають флексбокс, flexbox або просто flex (флекс), був розроблений як модель одномірно-спрямованого макета і як один із методів розподілу простору між елементами в інтерфейсі, з потужними можливостями вирівнювання.

Під час виконання кваліфікаційної роботи використовувались такі властивості Flexbox:

- Display:flex – відображає блок як flex елемент.
- Flex-Direction – CSS властивість, що вказує на те, як flex-елементи розміщуються у flex-контейнері по головній осі та напрямку.



- Flex-Wrap – задає правила виведення flex-елементів в один рядок або кілька, з перенесенням блоків. Якщо перенесення дозволено, то можна встановити напрямок, в якому виводяться блоки.
- Flex – ця властивість дозволяє задавати значення Flex-Grow, Flex-Shrink та Flex-Basis. Зазвичай їй задають значення flex:1, що дозволяє flex-елементу розтягуватися на весь доступний простір.

## 2.4 HTML

Під час розробки кваліфікаційної роботи були використані основні теги з мови гіпертекстової розмітки задля створення необхідного простору для технологій, які описані у пунктах вище. Нижче наведені теги, які були використані:

- Html – так званий контейнер, який тримає в собі весь вміст веб сторінки.
- Head – використовується для підключення файлів зі стилями, та розміщення тегу з назвою сторінки, а саме title.
- Body – «тіло» веб-сайту в якому далі розміщуються всі наступні теги, та вся сторінка.
- Header, Main, Footer – теги в яких знаходиться інформація верхньої, середньої та нижньої частини сайту, такий розподіл веб-сторінки використовується майже на кожному макеті, через зручність.
- Div – універсальний блочний тег, який використовується для групування елементів документа з метою зміни вмісту через стилі.
- Class – атрибут елемента, який прив'язує його до будь-якого класу, який описується у CSS.
- Ul, Li – завдяки цим тегам створюються список та його елементи.
- Link – завдяки цьому тегу можна підключати зовнішні файли до HTML документа.
- Img – всередині розміщується url посилання на зображення, яке буде розміщено на сторінці.

- P – визначає параграф чи текстовий абзац. Зазвичай використовується як контейнер для тексту, який необхідно стилізувати.
- Footer – спеціальний тег, який створений для виділення нижньої частини сайту(підвалу), зазвичай там міститься контактна інформація або інформація про авторське право.
- Header – такий самий тег як і footer, тільки використовується для виділення вмісту верхньої частини веб-сайту. Найчастіше містить в собі блок з меню, логотипом.
- Main – тег, що зберігає в собі основну інформацію веб сторінки.
- Script – використовується для вбудовування JavaScript коду, без підключення файлу з розширенням .js. Зазвичай це робиться, якщо до сайту необхідно підключити невеликий шматочок коду без створення окремого файлу.
- Br – тег, яким можна перенести рядок в тексті сторінки. Він робить розрив рядка, а не ділить текст абзацами.
- Strong – використовується для виділення важливого тексту.

## 2.5 JavaScript

Також я використовував мову програмування JavaScript (або JS) – це мова програмування, яка може додати інтерактивності веб-сайту. В розробці сайтів вона поєднується разом з HTML та CSS. HTML – визначає розмітку сторінки, CSS – зовнішній вигляд, а JavaScript вносить інтерактивність. Тобто з її допомогою можна описати, як розмітка і стилі відреагують на ті чи інші дії користувача. Коротко описати принцип роботи JS можна наступним чином:

Користувач щось зробив на сторінці, ці дії(клік миші, скрол сторінки та інші) запускають подію, далі запускається JS код, котрий призначений на цю подію і як результат на сторінці щось відбулось.

Наприклад, користувач натиснув на маленьке зображення, спрацювала подія OnClick, далі запустилась функція, яка була назначена на клік по цьому зображенню, і відкрилась збільшена копія зображення.

Саме в моїй роботі JS була використана для створення слайдера, головними перевагами JavaScript у цьому пункті є:

**Динамічність:** JavaScript дає змогу створювати слайдери з можливістю динамічної зміни зображень, тексту та інших елементів, які з'являються на слайді. Ця стратегія створює живий і захоплюючий вміст, який покращує взаємодію користувача з веб-сайтом.

**Гнучкість:** JavaScript дозволяє легко змінювати ряд налаштувань повзунка, наприклад швидкість слайда, ефекти анімації та автоматичне або ручне перемикавання слайдів. Це дає змогу забезпечити ідеальну взаємодію з користувачем, налаштувавши повзунок відповідно до вимог конкретного проекту.

JavaScript є широко поширеною мовою програмування, тому слайдер, створений з його допомогою, працюватиме в більшості сучасних веб-браузерів. У результаті можна вільно надавати одну функціональність повзунка різним користувачам.

Функціональність слайдера легко розширити завдяки можливостям JavaScript. JavaScript можна використовувати для додавання додаткових функцій, таких як автоматичне змінення розміру слайдера у відповідь на зміну розміру вікна браузера, підтримку жестів для пристроїв із сенсорним екраном та інтеграцію з іншими бібліотеками та плагінами.

Проста інтеграція HTML та CSS - використання JavaScript для створення слайдера дозволяє легко інтегрувати його з уже написаним кодом HTML і стилізувати його за допомогою CSS. Це дає вам велику свободу змінювати вигляд повзунка та його взаємодію з іншими елементами сторінки.

## 2.6 Google Fonts

На сайті для застосовано одна із найпопулярніших бібліотек для підбору шрифтів, а саме Google Fonts. Для коректної роботи необхідно підключити шрифти у html файлі за допомогою спеціального посилання, яке можна отримати безпосередньо на сайті, і далі прописати у файлі стилізації потрібні властивості.

Чому Google Fonts є таким популярним вибором для веб-розробників: Великий вибір шрифтів: Google Fonts має вражаючий асортимент шрифтів, доступних для безкоштовного використання. Ви можете знайти шрифти різних стилів і налаштувань, що дозволяє підібрати відповідний шрифт для своїх потреб.

Легкість використання: Підключення Google Fonts до веб-сторінки є дуже простим. Вам потрібно лише додати кілька рядків коду в HTML-файл, і ви зможете використовувати шрифти на своєму веб-сайті.

Швидкість завантаження: Google Fonts пропонує оптимізовані шрифти, які швидко завантажуються на веб-сторінку. Це важливо для поліпшення продуктивності вашого веб-сайту та забезпечення швидкого завантаження сторінок.

Кросбраузерна сумісність: Google Fonts працює на багатьох популярних веб-браузерах, що забезпечує єдність відображення шрифтів на різних пристроях та платформах. Безкоштовне використання - Google Fonts надає шрифти для веб-розробників безкоштовно. Ви можете використовувати їх як у комерційних, так і в некомерційних проектах, не потребуючи платежів або ліцензування.

Загалом, Google Fonts є зручним та доступним інструментом для веб-розробників, який дозволяє використовувати різноманітні шрифти на їх веб-сайтах з мінімальними зусиллями. Вибір великого асортименту шрифтів та його легке використання роблять Google Fonts популярним вибором серед веб-розробників у всьому світі.

## 2.7 Visual Studio code

Для розробки кваліфікаційної роботи я вирішив обрати даний редактор коду тому, що:

- Він є безкоштовним, крос-платформним та має відкритий код.
- Visual Studio Code підтримує широкий спектр мов програмування та технологій, що використовуються у веб-розробці, таких як HTML, CSS, JavaScript, TypeScript, Python, PHP та інші. Вбудовані функції, такі як підсвічування синтаксису, автодоповнення та перевірка помилок, полегшують роботу з цими мовами.
- Забезпечує зручну інтеграцію з популярними інструментами розробки, такими як Git, веб-серверами та відладчиками. Це дозволяє розробникам працювати з улюбленими інструментами безпосередньо у редакторі, що підвищує продуктивність та ефективність роботи.
- VS Code має велику та активну спільноту розробників, яка вносить внесок до його розвитку. Це означає, що завжди є ресурси, допомога та додаткові розширення, які можна знайти та використовувати
- Редактор підсвічує та виділяє код на багатьох мовах програмування.
- Плагін LiveServer – один із найзручніших додатків, який дозволяє запустити власний локальний сервер у браузері, та спостерігати зміни на сторінці в реальному режимі, без перезавантаження сторінки.
- Зручне налаштування візуальної частини, що робить редактор більш приємним у використанні.

## 2.8 Github desktop

У кваліфікаційній роботі я використовував GitHub для збереження коду свого проекту, але саме desktop версія була потрібна для:

Зручного підключення та інтеграції потрібного репозиторія одразу у середовище розробки(Visual Studio Code).

Своєчасного збереження та оновлення коду та необхідних файлів у репозиторії.

Зручної роботи з GitHub не виходячи з зони комфорту.

## **2.9 Github pages**

Також використовувався дуже зручна функція на GitHub, а саме GitHub pages. Це безкоштовний хостинг на GitHub, який дозволяє розміщати веб-сторінки(html, css, js файли) на їх власний хостинг та створювати власний домен, завдяки чому абсолютно без плати за це можна отримати повністю робочий сайт, звісно якщо у репозиторії користувача наявні всі необхідні файли для запуску.

## РОЗДІЛ 3. Опис, часткова демонстрація сторінок і порівняльна характеристика макетів CSS Grid та FlexBox.

### 3.1 Розроблені сторінки макетів

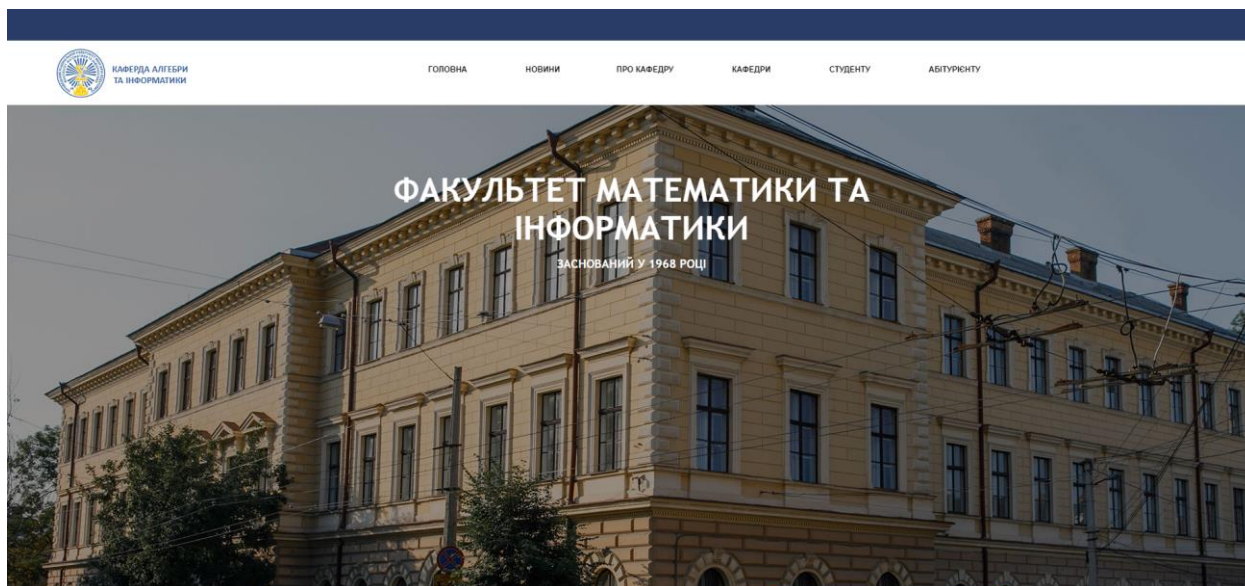


Рис.3.1 – Перша частина головної сторінки макету Grid CSS

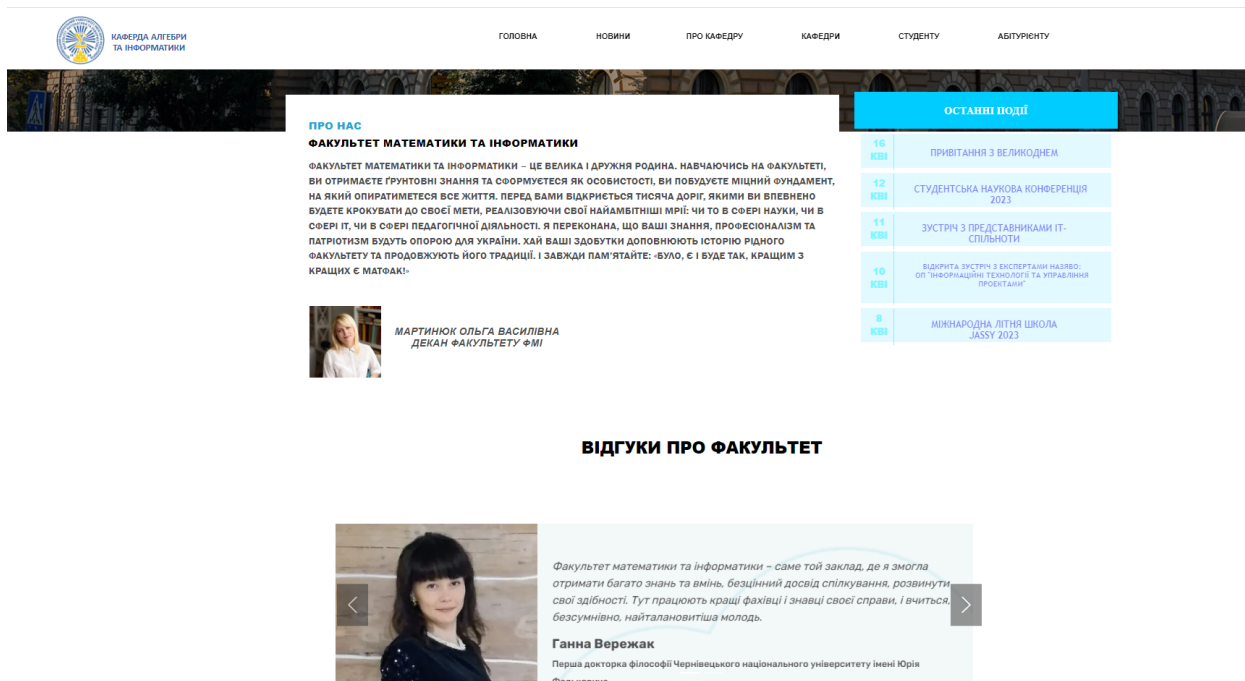


Рис.3.2 – Друга частина головної сторінки макету Grid CSS

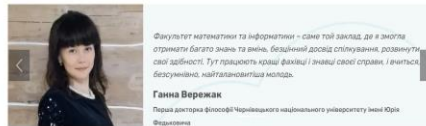


Рис.3.3 – Третя частина головної сторінки макету Grid CSS

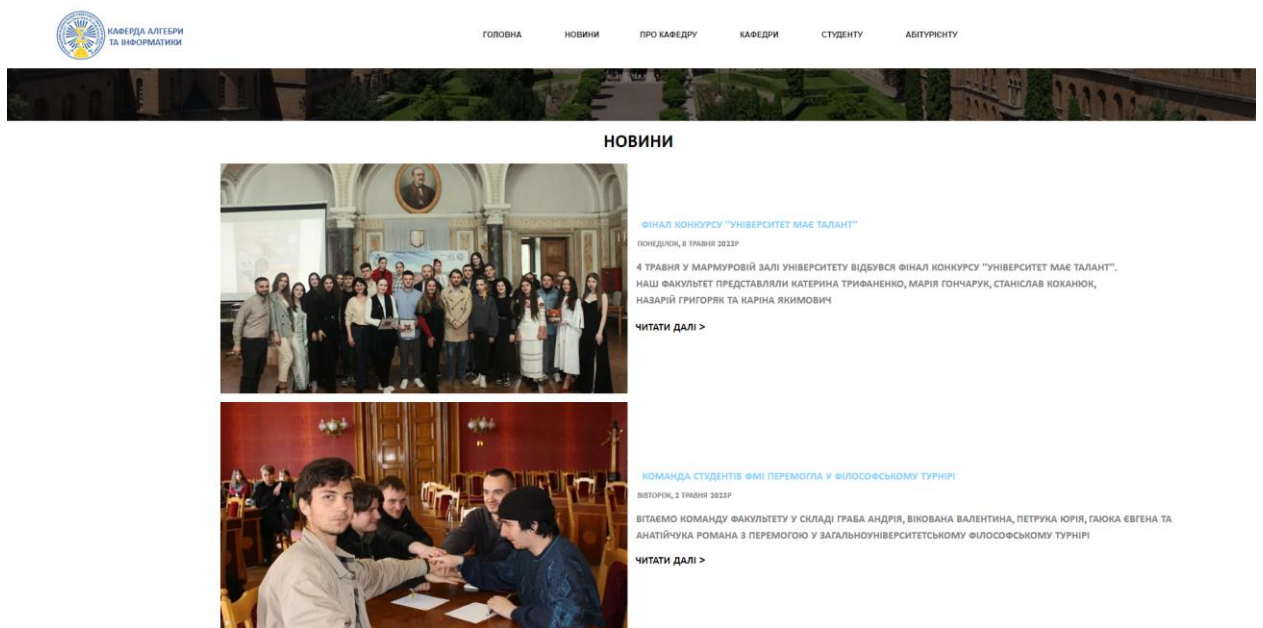


Рис. 3.4 – Сторінка з новинами Grid CSS

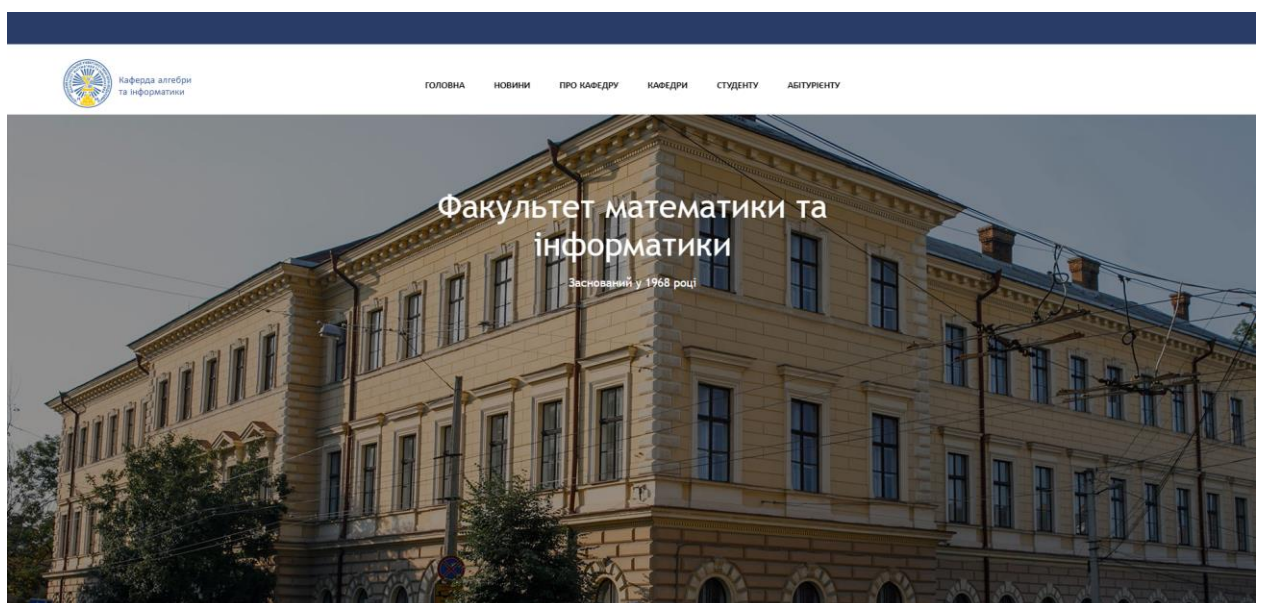


Рис. 3.5 – Перша частина головної сторінки макету FlexBox



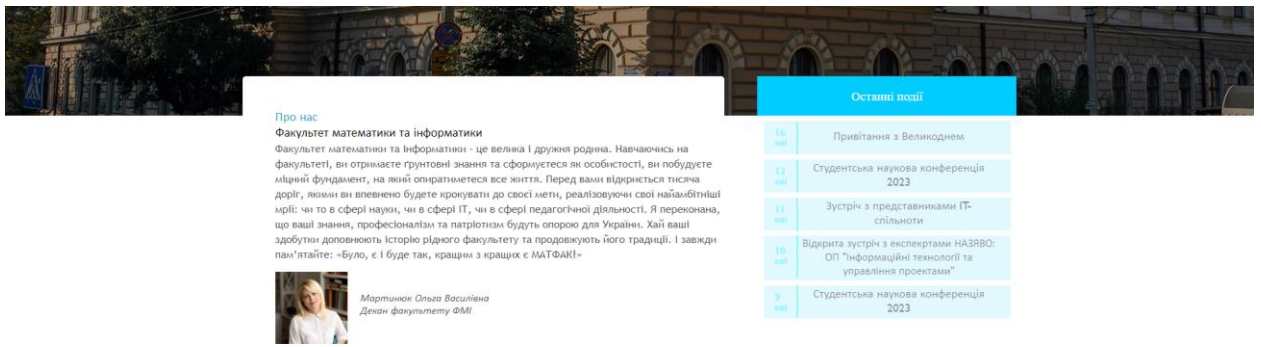


Рис.3.6 – Друга частина головної сторінки макету FlexBox



Рис.3.7 – Третя частина головної сторінки макету FlexBox

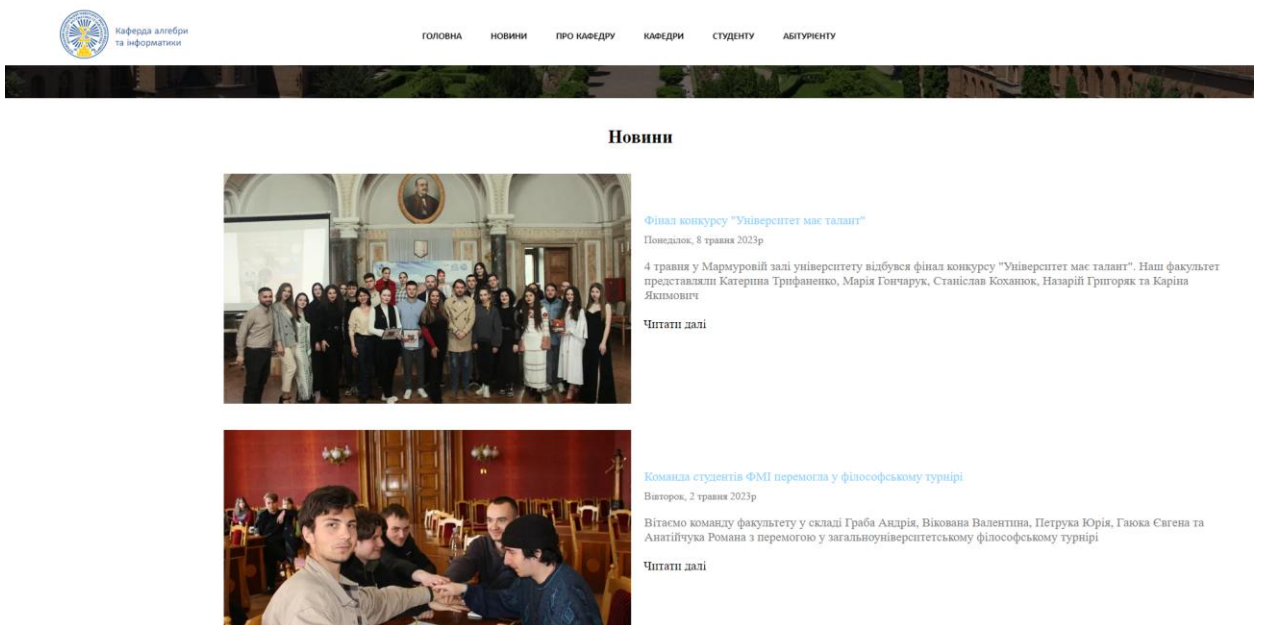
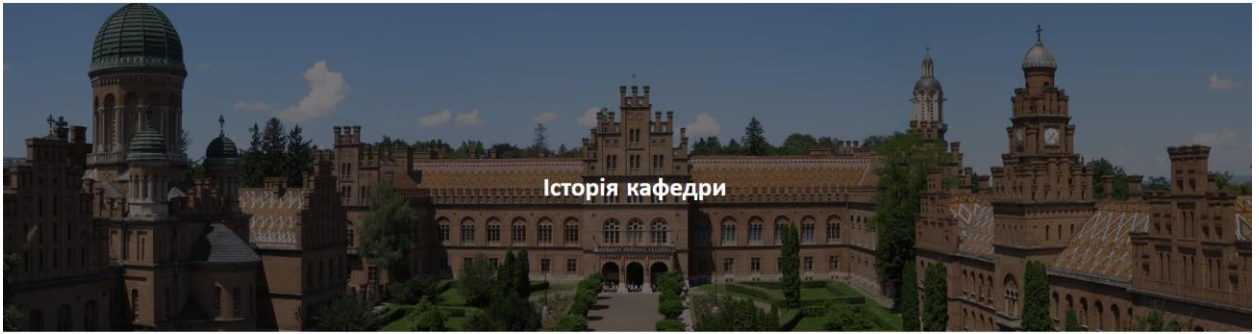


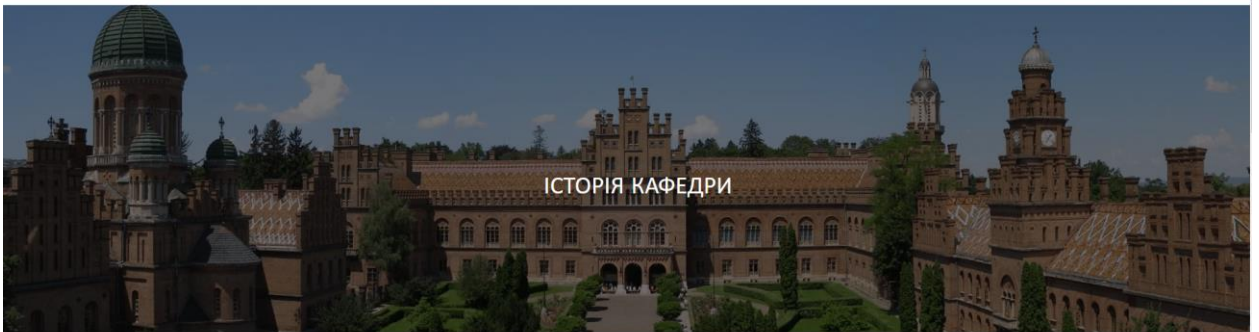
Рис. 3.8 – Сторінка з новинами FlexBox



### Історія кафедри

У 1940 році у складі фізико-математичного факультету Чернівецького державного університету була утворена кафедра алгебри та геометрії. Важливу роль в організації кафедри та обранні її наукового напрямку відіграли академіки Микола Миколайович Боголюбов (1909-1992), Борис Володимирович Гнеденко (1912-1995) та заслужений діяч науки УРСР, професор Олександр Степанович Смогоржевський (1896-1969). Першим завідувачем кафедри було призначено доцента Юрія Олександровича Кікеша (1863-1969), а з 1945 року кафедрою завідував учень Олександра Смогоржевського Микола Григорович Бєляєв (1904–1981). За направленням у лютому 1945 року він прибув на роботу в Чернівецький університет. У 1952 році захистив кандидатську дисертацію й наполегливо працював над проблемами диференціальної геометрії ліній та поверхонь неевклідових просторів нормалізації гіперповерхонь, оснащених полем прямих, опублікував понад 30 наукових праць.

Рис. 3.9 – Історія кафедри FlexBox



### ІСТОРІЯ КАФЕДРИ

У 1940 році у складі фізико-математичного факультету Чернівецького державного університету була утворена кафедра алгебри та геометрії. Важливу роль в організації кафедри та обранні її наукового напрямку відіграли академіки Микола Миколайович Боголюбов (1909-1992), Борис Володимирович Гнеденко (1912-1995) та заслужений діяч науки УРСР, професор Олександр Степанович Смогоржевський (1896-1969). Першим завідувачем кафедри було призначено доцента Юрія Олександровича Кікеша (1863-1969), а з 1945 року кафедрою завідував учень Олександра Смогоржевського Микола Григорович Бєляєв (1904–1981). За направленням у лютому 1945 року він прибув на роботу в Чернівецький університет. У 1952 році захистив кандидатську дисертацію й наполегливо працював над проблемами диференціальної геометрії ліній та поверхонь неевклідових просторів нормалізації гіперповерхонь, оснащених полем прямих, опублікував понад 30 наукових праць.

З 1953 по 1956 рік Микола Бєляєв працював на посаді проректора з навчальної роботи ЧДУ, з 1956 по 1966 рік – декан фізико-математичного факультету, з 1956 по 1973 рік – завідувач кафедри алгебри та геометрії. Доклав багато зусиль для встановлення наукових контактів кафедри алгебри та геометрії з провідними математичними центрами колишнього СРСР, приділяв значну увагу безпосередній роботі з молоддю, майстерно читав лекції з основних та спеціальних курсів.

Рис. 3.10 – Історія кафедри Grid CSS



**Мартинюк Ольга Василівна**

Декан факультету математики та інформатики. Доктор фізико-математичних наук, професор кафедри алгебри та інформатики



**Кушнірчук Василь Йосипович**

Заступник декана з навчально-виховної роботи. Кандидат фізико-математичних наук, доцент кафедри математичного моделювання

Рис. 3.11 – Деканат FlexBox



**МАРТИНЮК ОЛЬГА ВАСИЛІВНА**

ДЕКАН ФАКУЛЬТЕТУ МАТЕМАТИКИ ТА ІНФОРМАТИКИ, ДОКТОР ФІЗИКО-МАТЕМАТИЧНИХ НАУК, ПРОФЕСОР КАФЕДРИ АЛГЕБРИ ТА ІНФОРМАТИКИ



**КУШНІРЧУК ВАСИЛЬ ЙОСИПОВИЧ**

ЗАСТУПНИК ДЕКАНА З НАВЧАЛЬНО-ВИХОВНОЇ РОБОТИ. КАНДИДАТ ФІЗИКО-МАТЕМАТИЧНИХ НАУК, ДОЦЕНТ КАФЕДРИ МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ

Рис. 3.12 – Деканат GridCSS



Магістри



День вишиванки



Факультету - 50



Математична майстерня

Рис. 3.13 – Галерея FlexBox



МАГІСТРИ



ДЕНЬ ВИШИВАНКИ



ФАКУЛЬТЕТУ - 50



МАТЕМАТИЧНА МАЙСТЕРНЯ

Рис. 3.14 – Галерея GridCSS

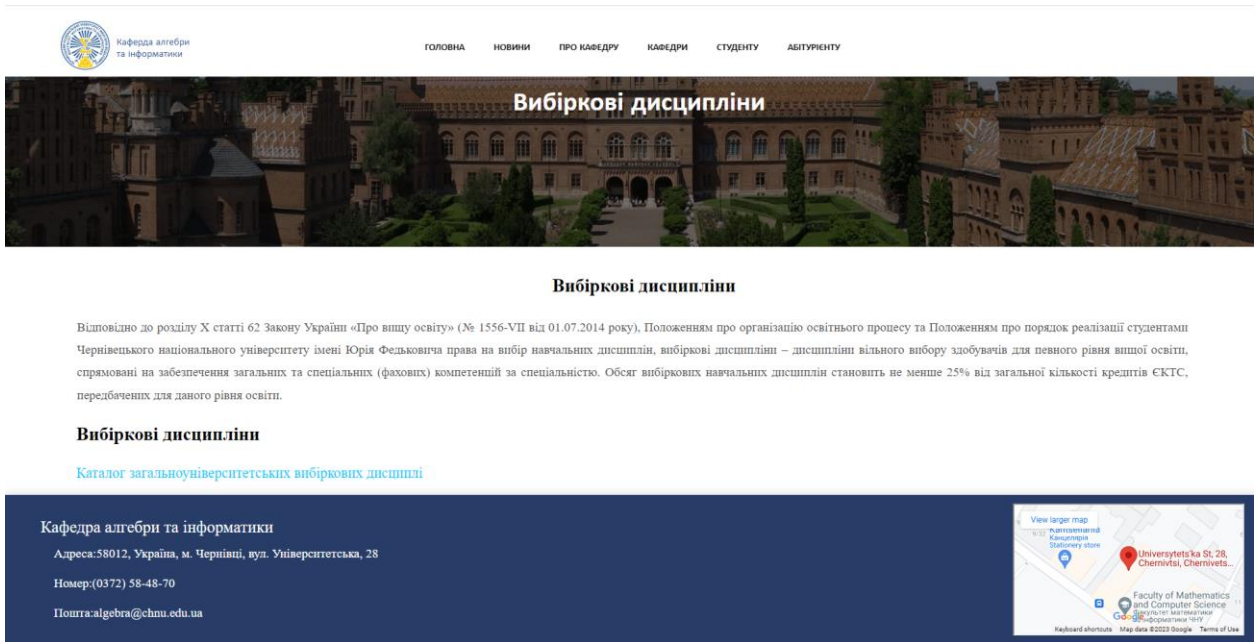


Рис. 3.15 – Вибіркові дисципліни FlexBox

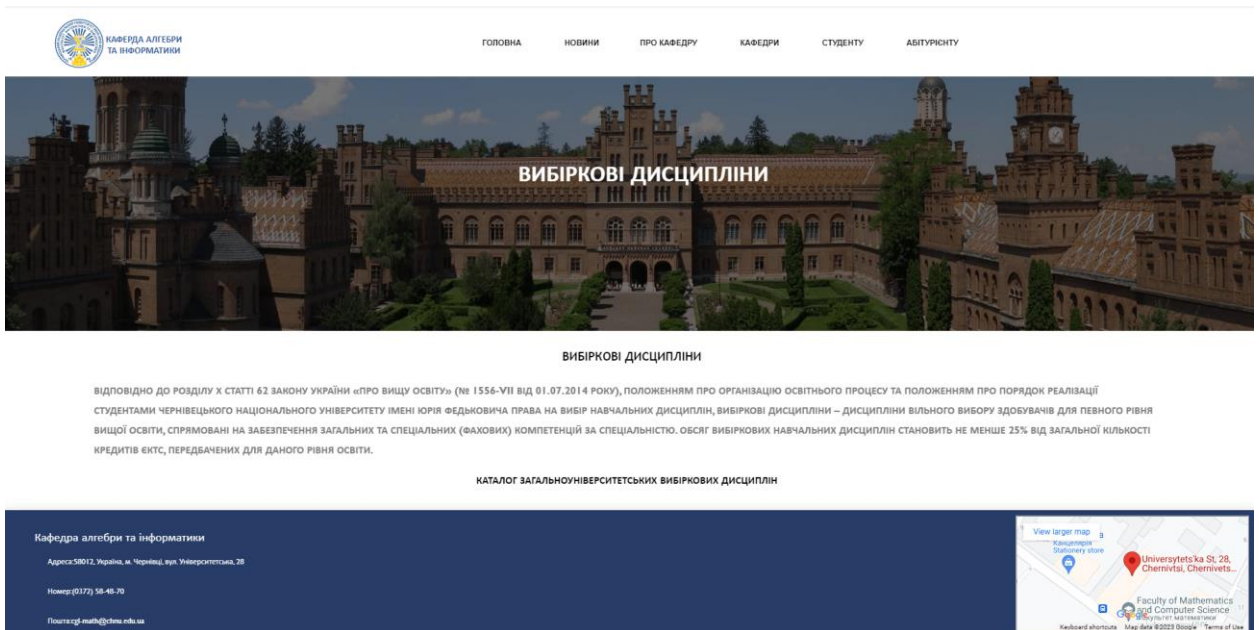


Рис. 3.16 – Вибіркові дисципліни GridCSS

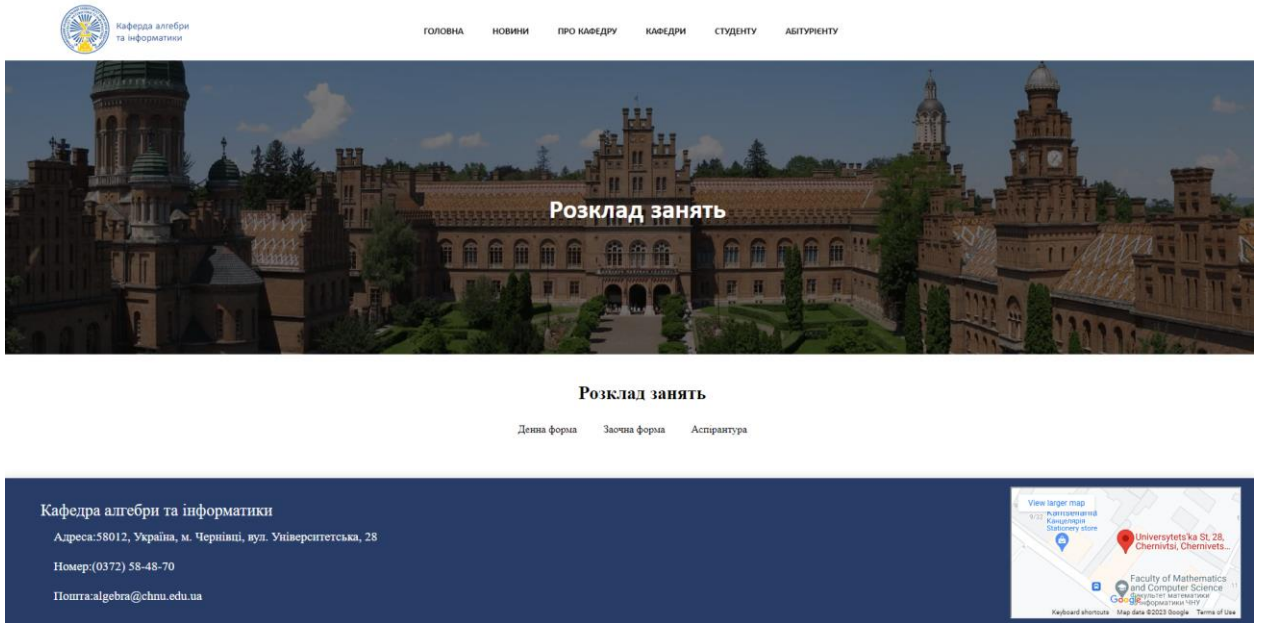


Рис. 3.17 – Розклад занять FlexBox

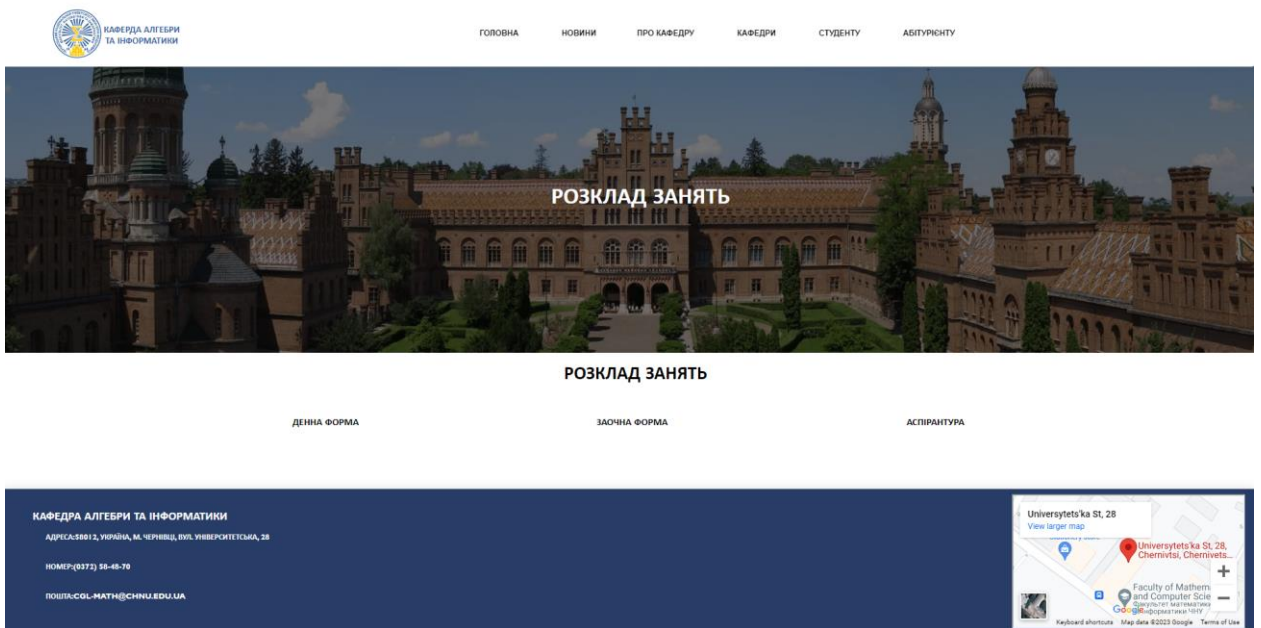


Рис. 3.18 – Розклад занять GridCSS

Дані сторінки були розроблені під час виконання кваліфікаційної роботи.

### 3.2 Порівняльна характеристика

Візуально макети, які створені за допомогою CSS Grid та FlexBox майже не відрізняються, лише на деяких сторінках видно, що елементи розміщені трішки по різному, але значна різниця буде помітна, якщо перейти до коду сторінок, а саме до файлу стилів(CSS). Для сторінок, які були створені з використанням flexbox, потрібно буде написати більше рядків коду, адже потребує більшої пильності при розміщені елементів у блоках на сторінці.

Також можна звернути увагу на розташування елементів, на двох макетах буде помітно, що однакові елементи можуть бути розташовані трохи по різному (зміщені на кілька пікселів), все це відбувається через особливості тої чи іншої технології.

Наочний приклад двох однакових макетів, але з різними технологіями:

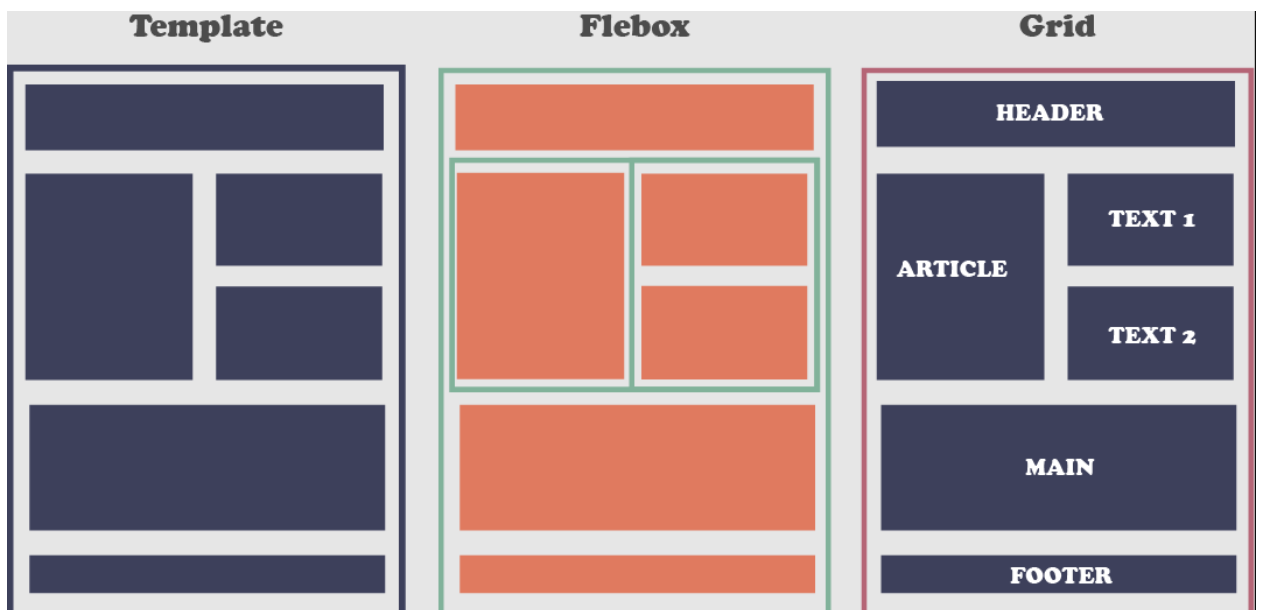


Рис. 3.19 – Однаковий макет – дві технології

Порівняння:

Швидкодія одна за найпотрібніших функцій при створенні будь-якого веб-сайту. Клієнти та просто відвідувачі певно, що віддадуть перевагу сайту, з більш простим дизайном, але хорошою швидкодією та меншою ресурсозатратністю, аніж сторінці з важким для системи дизайном, та повільним завантаженням.

Тому у сьогоднішніх реаліях гарний дизайн та швидкодію потрібно комбінувати.

Результати порівняння:

Макет створений з використанням Css grid



Рис. 3.20 – Швидкодія Grid

Макет створений з використанням Flexbox

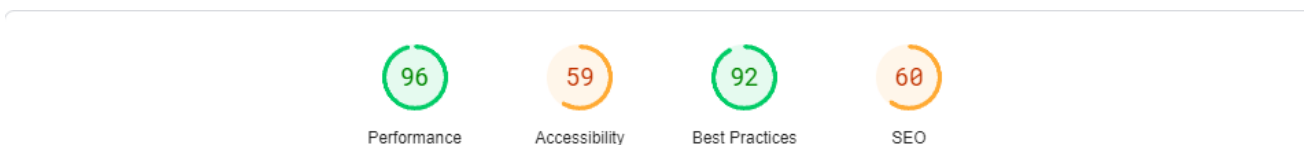


Рис. 3.21 – Швидкодія Flex

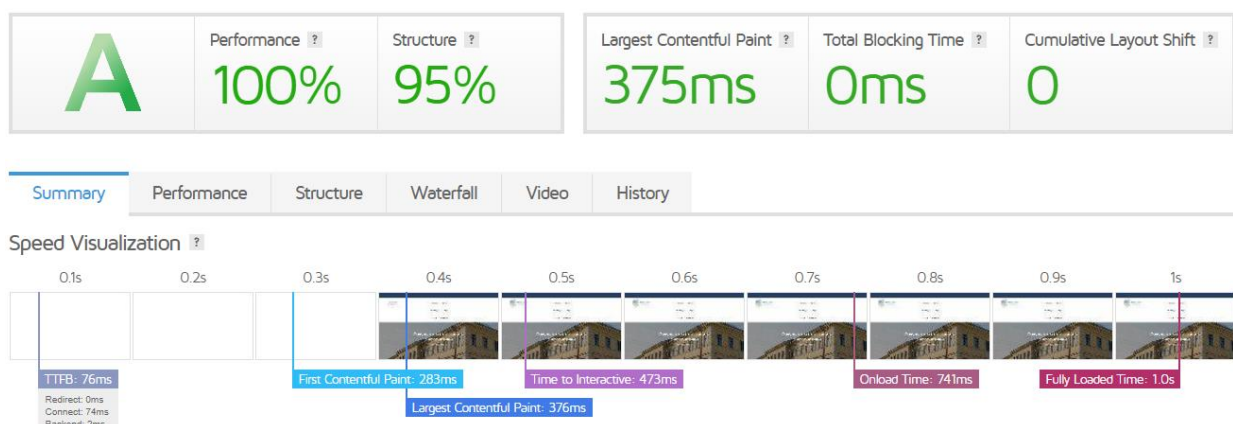


Рис. 3.22 – Тест швидкості завантаження контенту Grid CSS



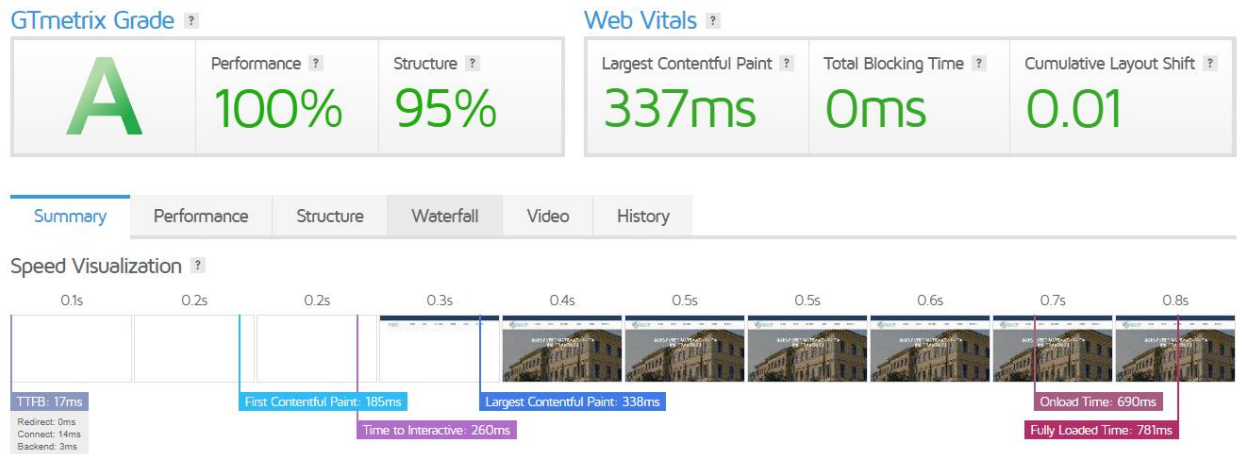


Рис. 3.23 – Тест швидкості завантаження контенту FlexBox

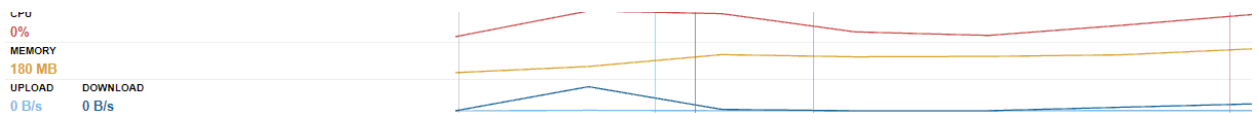


Рис. 3.24 – Графік використання процесора(CPU) та пам'яті при завантаженні сторінки FlexBox

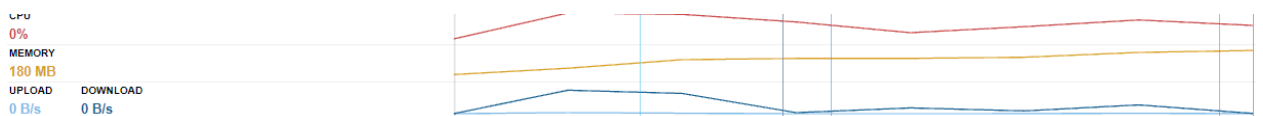


Рис. 3.25 – Графік використання процесора(CPU) та пам'яті при завантаженні сторінки Grid CSS

Тест був проведений з головними сторінками обох макетів. Сайт створений з використанням технології FlexBox показав кращі результати у порівнянні з Grid CSS(див. рис. 3.22, 3.23).

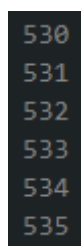
Незважаючи на те, що візуальний вигляд сторінок був однаковим, було помічено, що макет, розроблений за допомогою FlexBox, продемонстрував кращі результати порівняно з макетом, який створювався з використанням Grid CSS. Це може свідчити про те, що FlexBox більш підходить для сайтів, де швидкість завантаження є важливою, що стосується більшості сучасних сайтів, тоді як Grid CSS може бути кращим вибором для сайтів, де більше

уваги приділяється візуальному оформленню, а швидкість завантаження не є головним пріоритетом.

Додатково були проведені тестування затратності комп'ютерних ресурсів при завантаженні обох сторінок. Порівнявши два графіка (див. Рис. 3.24 та Рис. 3.25) можна зробити висновок, що завдяки технології FlexBox сайт використовує меншу кількість ресурсів комп'ютера та завантажується швидше, ніж Grid CSS.

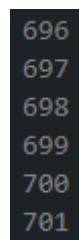
Відмінності між цими двома підходами можуть бути помічені і в коді. Під час розробки двох макетів від самого початку, було помічено, що для сторінок, де використовувався FlexBox, потрібно було додавати більше коду, щоб досягти подібності з першим сайтом. З особистого досвіду під час виконання кваліфікаційної роботи було встановлено, що CSS Grid є більш зручним для створення макетів сайту завдяки його простоті, доступності та широкому набору властивостей, які полегшують роботу.

Різниця у кількості коду в файлі стилів:



```
530
531
532
533
534
535
```

Рис.3.26 – Grid Cсс



```
696
697
698
699
700
701
```

Рис.3.27 – FlexBox

Як можна зауважити, у файлі стилів макету, у якому використовується технологія FlexBox рядків коду, а саме правил css більше, це свідчить про те,

що для ідентичної стилізації сторінки необхідно використати більше правил та докласти більше зусиль.

Ще однією відмінністю полягає в розміщенні елементів. Порівнюючи з Grid CSS, на макеті, де використовувався FlexBox, було складніше досягти бажаного розташування елементів, наприклад, створити три колонки однакового розміру.

Крім того, витрачений час на розробку кожного макету був різним. Розробка макету з використанням Grid CSS зайняла багато більше часу через необхідність приділення увагу дрібницям, які можуть вплинути на вигляд сторінки при неправильному використанні.

Для даного макету краще підходить використання CSS Grid, однак поєднання обох технологій може бути ефективною стратегією для досягнення кращих результатів у верстці сайту. Кожна з цих технологій має свої унікальні можливості та переваги, і використання їх разом може доповнювати одна одну та забезпечувати більшу гнучкість у розміщенні елементів та контролі над макетом.

## **ВИСНОВКИ**

Під час виконання кваліфікаційної роботи на тему «Розробка шаблону сайту засобами FlexBox, Grid CSS та їх порівняльна характеристика» я вивчив такі технології як FlexBox, Grid CSS, покращив знання у Html, CSS, JavaScript та актуальних технологій для створення веб-сайту. Створив два макети, використовуючи дві різні технології та провів їх порівняльну характеристику, що допомогло мені краще зрозуміти, коли і як їх потрібно поєднувати.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Grid layout in css interface layout for the Web / Eric A. Meyer // [с. 7, 13, 24]  
[Електронна книга]
2. <https://developer.mozilla.org/en-US/docs/Web/CSS> [Електронний ресурс]
3. <https://developer.mozilla.org/ru/docs/Web/HTML> [Електронний ресурс]
4. <https://habr.com/ru/post/656245/> [Електронний ресурс]
5. <https://developer.mozilla.org/ru/docs/Web/CSS/@media> [Електронний ресурс]
6. Html and Css design and build websites / John Dacket // с. 100-107.  
[Електронна книга]
7. <https://github.com/tresqblock/diplomafinal/> - мій репозиторій з кодом.
8. <https://tresqblock.github.io/diplomafinal/flex.html> - макет FlexBox, який завантажений на онлайн хостинг та доступний для перегляду.
9. <https://tresqblock.github.io/diplomafinal/mainpage.html> - макет Grid CSS, який завантажений на онлайн хостинг та доступний для перегляду.  
[Електронний ресурс]
10. <https://web.dev/learn/css/flexbox/> [Електронний ресурс]
11. [https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\\_Flexible\\_Box\\_Layout/Typical\\_Use\\_Cases\\_of\\_Flex\\_box](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Flexible_Box_Layout/Typical_Use_Cases_of_Flex_box) [Електронний ресурс]
12. [https://en.wikipedia.org/wiki/CSS\\_grid\\_layout](https://en.wikipedia.org/wiki/CSS_grid_layout) [Електронний ресурс]
13. <https://fmi.chnu.edu.ua/> - сайт, на основі якого були створені макети для порівняння.
14. <https://pagespeed.web.dev/> [Електронний ресурс]