



Міністерство освіти та науки України

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»  
Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

## **ЗВІТ**

з дисципліни «Програмування веб-застосунків»  
Комп'ютерний практикум №2

**Виконав:**

*Студент ІІ курсу  
групи ІП-41  
Чередніченко А.О.*

**Перевірив:**

ст. викладач – Ковтунець О.В.

Комп'ютерний практикум №1  
Розробка HTML-структур та візуалізація з допомогою CSS

Мета роботи:

Ознайомлення з мовою візуального оформлення CSS та освоєння сучасних способів верстки вебсторінок.

**Завдання**

1. У вас вже є обраний із 16 варіантів власний варіант виконання завдання відповідно до вашого порядкового номера у списку групи, подальші маніпуляції виконуйте, враховуючи підготовлену головну сторінку вебсайта із завдання комп'ютерного практикуму №1.
2. Засобами HTML і CSS зверстайте нові екземпляри головної вебсторінки вашого вебсайта згідно з вашим варіантом, використовуючи верстку:

А) з допомогою Flexbox CSS,

Б) з допомогою Grid CSS,

В) без використання Flexbox CSS та Grid CSS,

зберігаючи контент вебсторінки, з урахуванням позиціонування та візуалізації окремих елементів вебсторінки. За неможливості виконати котрийсь із пунктів А,Б,В для вашого варіанту поясніть причину. Вкажіть переваги та недоліки видів верстки на основі зверстаних варіантів.

3. Для запису інструкцій CSS використовуйте окремі файли стилів, а у відповідних екземплярах головної вебсторінки забезпечте вказівник на ці файли стилів. Не задавайте жодних інструкцій CSS у файлах вебсторінок.

4. Переконайтеся в коректності функціонування розробки на різних обсягах змістового наповнення та на різних розмірах вікна переглядача.

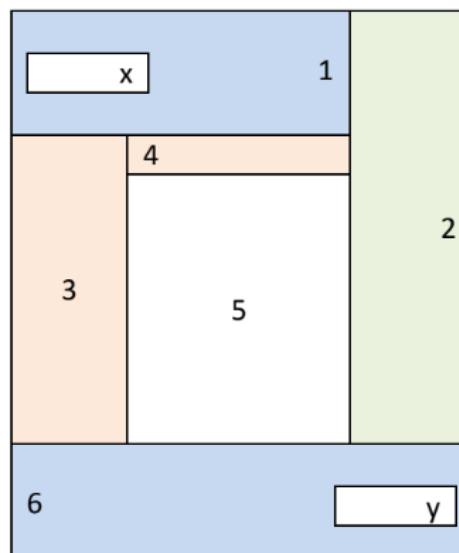
5. Продемонструйте роботу новостворених екземплярів головної вебсторінки через локальну файлову систему.

6. Опублікуйте новостворені екземпляри вебсторінок і файли стилів на GitHub Pages (або обраному вами хостінгу).

7. Продемонструйте роботу екземплярів головної вебсторінки в інтернеті.

Макет обраного мною варіанту №11:

**Варіант 11**



**Виконання роботи:**

Для створення вебсайту було використано Visual Studio Code та встановлене розширення Live Server для симуляції локальної роботи вебсайту у браузері. Для зображення 3 версій розміток у меню було створено додатково 3 посилання на 3 сторінки: Grid, Flexbox та No grid, No flexbox розміток. Головна сторінка в меню – це попередня версія сайту під час виконання Практикуму №1, розмітка якої була виконана комбіновано за допомогою grid та flexbox.(рис.1).



Рис. 1 Вигляд

На рисунку 2 зображено вигляд сайту за допомогою розмітки Flexbox, він є ідентичним до вигляду сторінки, зробленої під час першого практикуму.



Рис. 2 Розмітка з допомогою Flexbox

На рисунку 3 зображено вигляд сайту за допомогою розмітки Grid, він також є ідентичним до вигляду сторінки, зробленої під час першого практикуму.

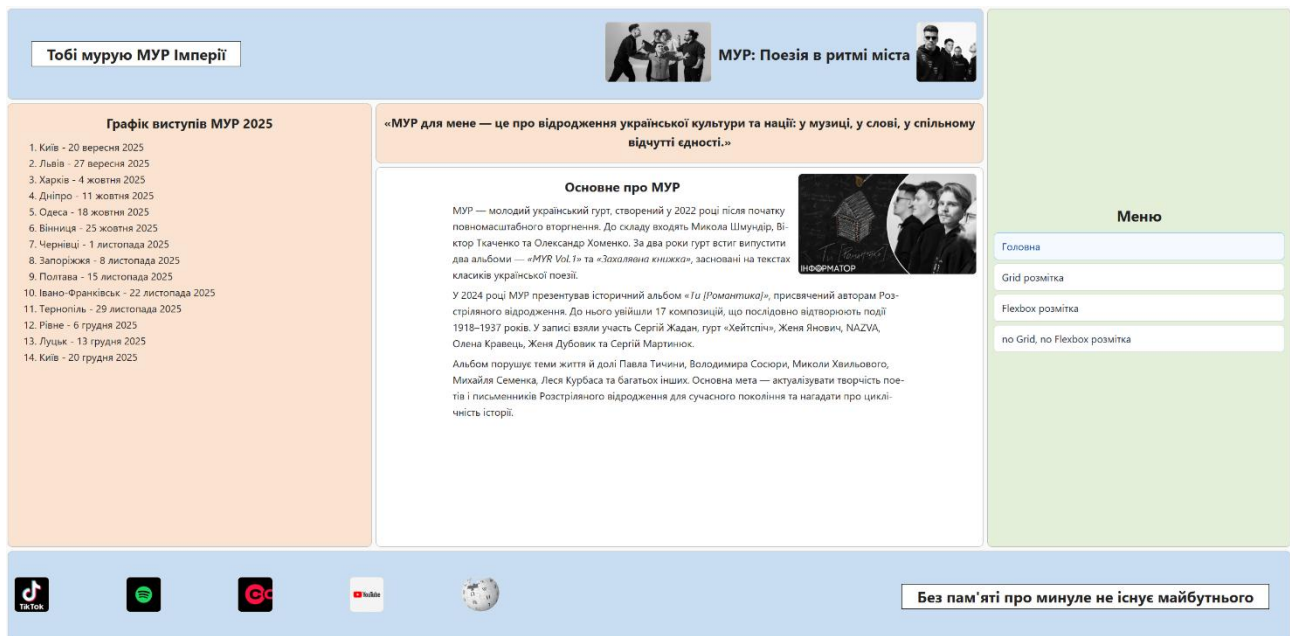


Рис. 3 Розмітка з допомогою Grid

На рисунку 4 зображено вигляд сайту без використання Grid та Flexbox. Особисто для мене це був найважчий вид верстки, адже досягти первинного варіанту було практично неможливо через свою складність у реалізації розташування блоків, а саме важке вертикальне вирівнювання та “рівні колонки”. Перевірювши як це виглядає на різних моніторах, я впевнився, що сенсу адаптувати під різні роздільні здатності немає, тому в моєму випадку реалізовано для монітора з роздільною здатністю 2560 \* 1440. Адаптації, що стосуються розміру вікна також були мінімально проведені, окрім основного блоку, адже заважає інший блок із цитатою над ним та ламає його розміри.



Рис. 4 Розмітка без використання Grid та Flex

Було складено таблицю порівняння варіантів верстки зі вказанням недоліків та переваг

Без Flex/Grid	Flexbox	CSS Grid
<ul style="list-style-type: none"> <li>• Переваги — працює навіть у дуже старих браузерях.</li> <li>• Переваги — достатньо для елементарних лінійних макетів.</li> <li>• Недоліки — потребує хаків (float, clearfix, проміжки в inline-block).</li> <li>• Недоліки — важке вертикальне вирівнювання та “рівні колонки”.</li> <li>• Недоліки — майже неможливо міняти порядок/робити адаптив без перестановок у DOM.</li> <li>• Недоліки — крихіткість при зміні контенту; багато медіа-запитів.</li> </ul>	<ul style="list-style-type: none"> <li>• Переваги — ідеальний для 1-D потоків (ряди/колонки, меню, тулбари).</li> <li>• Переваги — легке центрування і вирівнювання, рівні висоти, підтримка gap.</li> <li>• Переваги — зручний адаптив: flex-wrap, order, авто-марджини без зайвих медіа-запитів.</li> <li>• Недоліки — не 2-D: повноцінні сітки сторінки будувати складно.</li> <li>• Недоліки — при переносі рядків висоти елементів між рядками не синхронізуються.</li> <li>• Недоліки — плутанина flex-basis/width і дефолтних min-width може ламати розмітку.</li> </ul>	<ul style="list-style-type: none"> <li>• Переваги — справжній 2-D лейаут: керування рядками, колонками й областями.</li> <li>• Переваги — потужні інструменти: fr, minmax, auto-fit/auto-fill, gap, (де є) subgrid.</li> <li>• Переваги — менше медіа-запитів: сітка сама підлаштовується під ширину.</li> <li>• Переваги — точне позиціонування та вирівнювання, можливість накладання шарів.</li> <li>• Недоліки — вищий поріг входу; надмірний для простих одновимірних компонентів.</li> <li>• Недоліки — неявні треки та auto-висоти можуть давати несподівані результати.</li> </ul>

На мою думку, найдоцільніше обирати:

- Grid — каркас сторінки, лендінги, дашборди, двовимірні макети.
- Flexbox — компоненти: хедери, навігація, картки в ряд, групи кнопок.
- Комбінація — Grid для сторінкової сітки, Flex всередині блоків.
- Без Flex/Grid — лише для дуже простих або legacy/фолбек-кейсів

Для запису інструкцій CSS використано окремі файли стилів(рис. 5)

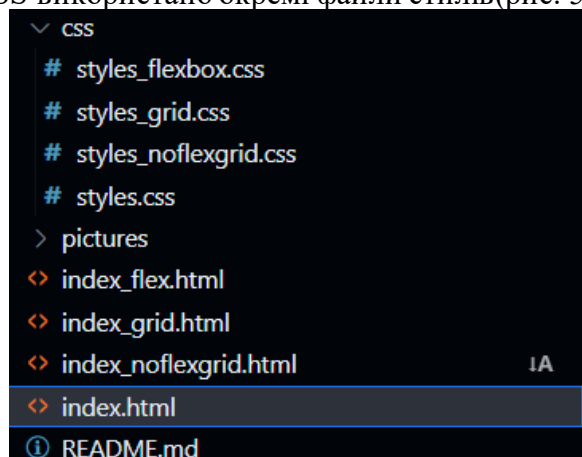


Рис. 5 Окремі файли css для розмітки

Посилання на Гіт Хаб Репозиторій: <https://github.com/blaz3xx/Web-Workshop2>  
На Git Hub Pages: <https://blaz3xx.github.io/Web-Workshop2/>

**Висновок:** У результаті виконання лабораторної роботи було опрацьовано сучасні способи верстки вебсторінок за допомогою Flexbox, Grid та базових інструментів CSS. Порівняння показало, що Flexbox зручний для одновимірних макетів, Grid — для двовимірних, а верстка без них є обмеженою. Сторінки були протестовані на адаптивність і опубліковані на GitHub Pages, що підтвердило досягнення мети роботи.