

2. Asti P. N. W., Andriyani A. Improving Critical Thinking Ability and Active Learning of Statistics Materials Through Problem Based Learning Model with STEAM Approach. *Formosa Journal of Sustainable Research (FJSR)*, 2022. № 1(2). P. 133–152

3. Chistyakov A. A., Kunitsyna M. L., Yagudina R. Exploring The Characteristics and Effectiveness of Project-Based Learning for Science and STEAM Education. *Eurasia Journal of Mathematics, Science and Technology Education*, 2023. № 19(5).

ОСОБЛИВОСТІ ВИКОРИСТАННЯ ТЕХНОЛОГІЙ АДАПТИВНОЇ ВЕРСТКИ САЙТІВ

Бабій Олег Богданович

здобувач першого (бакалаврського) рівня вищої освіти спеціальності Середня освіта
(Інформатика),

Тернопільський національний педагогічний університет імені Володимира Гнатюка,
babii_ob@fizmat.tnpu.edu.ua

Василенко Ярослав Пилипович

викладач кафедри інформатики та методики її навчання,

Тернопільський національний педагогічний університет імені Володимира Гнатюка,
yava@fizmat.tnpu.edu.ua

У сучасному цифровому світі вебсайти відіграють ключову роль у взаємодії користувачів з інформацією, сервісами та бізнесом. З розвитком мобільних пристроїв адаптивна верстка стала необхідністю для забезпечення зручного перегляду сайтів на різних екранах. Вона дозволяє створювати вебресурси, що відображаються та функціонують на будь-яких пристроях від смартфонів до широкоформатних моніторів.

Збільшення частки мобільного трафіку та різноманіття пристроїв змушує розробників використовувати адаптивні технології, щоб забезпечити якісний користувацький досвід. Відсутність адаптивної верстки може призвести до втрати потенційних клієнтів, погіршення оптимізації та зниження ефективності веб-ресурсу. Саме тому вивчення адаптивної верстки та сучасних фреймворків, які її спрощують, є надзвичайно важливим.

Адаптивна верстка це метод створення вебсторінок, при якому дизайн та елементи інтерфейсу динамічно змінюються відповідно до параметрів пристрою: розміру екрану, роздільної здатності, орієнтації та інших характеристик [1]. Головна мета такого підходу зробити сайт максимально зручним для всіх користувачів, незалежно від того, чи переглядають вони його з мобільного телефону, планшета чи комп'ютера. Принцип роботи адаптивної верстки базується на використанні спеціальних технологій, таких як CSS Media Queries, гнучкі сітки (flexible grids) та адаптивні зображення (responsive images). Завдяки цим методам сайт здатний змінювати своє відображення, зберігаючи при цьому логіку розміщення контенту та зручність навігації [2].

У вебдизайні існує кілька підходів до створення інтерфейсу, кожен із яких має свої особливості та сфери застосування. До основних альтернатив адаптивної верстки належать гумова (гнучка) верстка та мобільна версія сайту. Кожен із цих

підходів має свої переваги та обмеження, що впливають на вибір технології в залежності від цілей проєкту.

В таблиці 1 наведено ключові відмінності кожного методу, зокрема їх особливості реалізації, принципи роботи, а також переваги та обмеження, які варто враховувати при виборі оптимального підходу для розробки вебсайту.

Таблиця 1

Ключові відмінності методів вебпроектування [3-5]

Підхід	Опис	Відмінності від інших методів
Адаптивна верстка (Responsive Design)	Дизайн змінюється залежно від ширини екрану завдяки медіа-запитам	Використовує єдиний HTML-код і CSS, що підлаштовує макет під ширину екрану; не потребує окремої мобільної версії; забезпечує кращу SEO-оптимізацію порівняно з мобільною версією; складніша у розробці, ніж резинова верстка.
Резинова верстка (Fluid Layout)	Всі елементи сайту масштабуються за відсотковими значеннями	Не використовує медіа-запити, а змінює ширину елементів пропорційно екрану; гнучкіша, ніж адаптивна верстка, але може виглядати некоректно на надто великих або маленьких екранах; не дозволяє точного контролю над змінами розмітки для різних пристроїв.
Мобільна версія (Separate Mobile Site)	Окремий сайт для мобільних пристроїв з власним доменом (наприклад, m.example.com)	Вимагає створення та підтримки окремої версії сайту для мобільних пристроїв; має кращу продуктивність на мобільних пристроях у порівнянні з адаптивною версткою; потребує окремого управління контентом і може мати проблеми з дублюванням сторінок для SEO.

Адаптивна верстка (Responsive Design). Дизайн змінюється залежно від ширини екрану завдяки медіа-запитам використовує єдиний HTML-код і CSS, що підлаштовує макет під ширину екрану; не потребує окремої мобільної версії; забезпечує кращу SEO-оптимізацію порівняно з мобільною версією; складніша у розробці, ніж резинова верстка.

Гумова верстка (Fluid Layout). Всі елементи сайту масштабуються за відсотковими значеннями еє використовує медіа-запити, а змінює ширину елементів пропорційно екрану; гнучкіша, ніж адаптивна верстка, але може виглядати некоректно на надто великих або маленьких екранах; не дозволяє точного контролю над змінами розмітки для різних пристроїв.

Мобільна версія (Separate Mobile Site). Окремий сайт для мобільних пристроїв з власним доменом (наприклад, m.example.com) вимагає створення та підтримки окремої версії сайту для мобільних пристроїв; має кращу продуктивність на мобільних пристроях у порівнянні з адаптивною версткою; потребує окремого управління контентом і може мати проблеми з дублюванням сторінок для SEO.

Кожен із підходів до верстки має свої особливості та сфери застосування. Адаптивна верстка є найбільш універсальним і сучасним методом, що дозволяє створювати сайти, які однаково добре працюють на будь-яких пристроях, не потребуючи окремої мобільної версії. Резинова верстка забезпечує простішу

реалізацію, але не дозволяє точно контролювати вигляд сайту на різних екранах. Мобільна версія сайту може бути корисною для високонавантажених проєктів, але потребує додаткових ресурсів на підтримку та може викликати проблеми з SEO через дублювання контенту. Вибір відповідного підходу залежить від специфіки проєкту, ресурсів розробника та потреб кінцевих користувачів.

Основними технологіями для створення адаптивних вебдизайнів є Media Queries, Flexbox і CSS Grid. Media Queries дозволяють змінювати стилі залежно від ширини екрану, Flexbox забезпечує зручну роботу з рядковими та стовпцевими макетами, а CSS Grid надає потужні можливості для створення складних макетів із сіткою. Комбіноване використання цих методів дає змогу створювати гнучкі, зручні та сучасні веб-сайти, які коректно відображаються на будь-яких пристроях.

Адаптивна верстка вимагає не лише знань CSS, а й ефективних інструментів для її реалізації. Фреймворки значно спрощують процес розробки, забезпечуючи готові компоненти, сітки та стилі. Серед найпопулярніших рішень для створення адаптивних сайтів виділяють Bootstrap, Foundation та Tailwind CSS.

Bootstrap – це найкращий вибір для швидкої розробки завдяки найбільшому набору готових UI-компонентів, простоті використання та широкій підтримці. Foundation більше підходить для мобільно-орієнтованих інтерфейсів і забезпечує гнучкість у кастомізації, проте має меншу популярність. Tailwind CSS є оптимальним вибором для кастомних рішень, де потрібен повний контроль над стилями, але вимагає більше часу на освоєння та не містить готових UI-компонентів. Вибір фреймворку залежить від конкретних потреб проєкту та досвіду розробника.

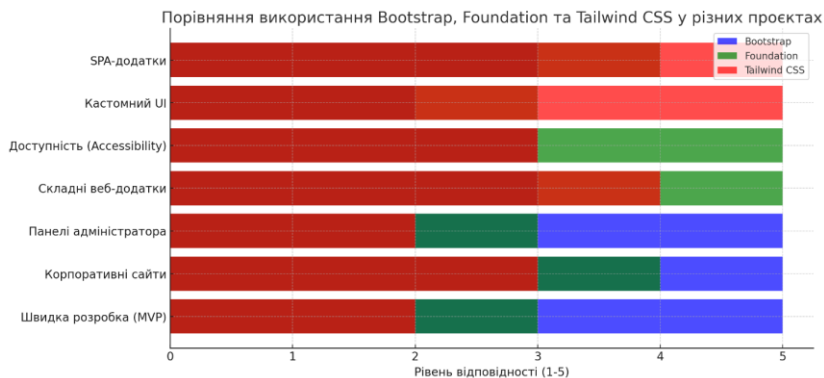


Рис. 1. Використання Bootstrap, Foundation та Tailwind CSS у різних проєктах

Вибір технологій та фреймворків для адаптивної верстки залежить від типу проєкту, його складності та вимог до кастомізації. У таблиці 2 подано рекомендації для різних категорій вебпроєктів.

Таблиця 2

Рекомендації щодо вибору технологій та фреймворків

Тип проєкту	Рекомендовані технології	Рекомендовані фреймворки
Корпоративний сайт	Media Queries, Flexbox	Bootstrap, Foundation
Інформаційний портал	Media Queries, Grid	Bootstrap, Foundation
Інтернет-магазин	Flexbox, Grid	Bootstrap, Foundation
Односторінковий додаток (SPA)	Flexbox, Tailwind CSS	Tailwind CSS

Лендінг	Media Queries, Flexbox	Bootstrap
Панель адміністратора	Flexbox, Grid	Bootstrap, Foundation
Складний веб-додаток	Grid, Foundation	Foundation, Tailwind CSS
Дизайнерський сайт / Кастомний UI	Tailwind CSS, Flexbox	Tailwind CSS

У ході даного дослідження було проведено аналіз сучасних технологій адаптивної верстки, їх застосування у вебдизайні та вебпрограмуванні, а також розглянуто популярні фреймворки, що використовуються для створення адаптивних інтерфейсів. На основі дослідження було визначено, що адаптивна верстка є ключовим елементом сучасного вебдизайну, оскільки забезпечує коректне відображення вебсайтів на різних пристроях та покращує користувацький досвід. Основні технології адаптивної верстки Media Queries, Flexbox та CSS Grid мають свої сфери застосування та найкраще підходять для різних типів макетів.

Було також проведено аналіз та порівняння популярних фреймворків, таких як Bootstrap, Foundation та Tailwind CSS. Дослідження показало, що:

- Bootstrap є найкращим вибором для швидкої розробки вебпроектів завдяки великій кількості готових UI-компонентів та широкій підтримці в спільноті;

- Foundation підходить для складних вебпроектів та рішень, орієнтованих на мобільні пристрої та доступність;

- Tailwind CSS забезпечує максимальну кастомізацію, що робить його оптимальним вибором для дизайнерських проектів та сучасних SPA-додатків.

На основі проведеного аналізу були розроблені рекомендації щодо вибору технологій і фреймворків для різних типів вебпроектів. Було створено таблицю відповідності технологій та фреймворків для різних задач, що дозволяє ефективно визначати оптимальне рішення залежно від специфіки проекту. Таким чином, результати дослідження дозволяють зробити висновок, що правильний вибір технологій адаптивної верстки та відповідного фреймворку суттєво впливає на швидкість розробки, гнучкість дизайну та кінцеву якість вебпроекту.

Список використаних джерел

1. GeeksForGeeks. Understanding Fluid Layouts in Web Design. URL: <https://www.geeksforgeeks.org/understanding-fluid-layouts-in-web-design-how-to-use-it> (дата звернення: 20.03.2025).
2. Expans. Адаптивна верстка сайту або мобільна версія? URL: <https://expans.ua/blog/adaptyvna-verstka-saytu-abo-mobilna-versiya> (дата звернення: 20.03.2025).
3. Ramotion. Responsive vs Adaptive Web Design. URL: <https://www.ramotion.com/blog/responsive-vs-adaptive-web-design> (дата звернення: 20.03.2025).
4. SendPulse. Адаптивний та респонсивний дизайн: у чому різниця? URL: <https://sendpulse.ua/blog/responsive-and-adaptive-design> (дата звернення: 20.03.2025).
5. W3Schools. CSS Responsive Web Design. URL: https://www.w3schools.com/css/css_rwd_intro.asp (дата звернення: 20.03.2025).