

<https://doi.org/10.31891/2307-5732-2025-355-42>
УДК 004.656.2

ЛАВРИК ВОЛОДИМИР

Київський національний університет технологій та дизайну

<https://orcid.org/0000-0002-6448-2470>

e-mail: lavyryk.vv@knuutd.edu.ua

НІКОНОВ ОЛЕГ

Київський національний університет технологій та дизайну

<https://orcid.org/0000-0002-8878-4318>

e-mail: nikonov.oy@knuutd.edu.ua

НОВАК ДМИТРО

Київський національний університет технологій та дизайну

<https://orcid.org/0000-0002-1796-8857>

e-mail: novak.ds@knuutd.edu.ua

ІНТЕГРАЦІЯ ТРАДИЦІЙНИХ І СУЧАСНИХ ВЕБТЕХНОЛОГІЙ У ПІДГОТОВКУ JUNIOR-ФАХІВЦІВ У СФЕРІ ФРОНТЕНД-РОЗРОБКИ

У статті розглянуто підготовку Junior-фахівців у фронтенд-розробці шляхом інтеграції HTML, CSS і сучасних фреймворків (React, Vue.js, Tailwind CSS). Обґрунтовано змішаний підхід до навчання: від базових технологій — до сучасних інструментів, що забезпечують відповідність потребам IT-ринку.

Ключові слова: HTML, CSS, фронтенд-розробка, React, Vue.js, Tailwind CSS.

LAVRIK VOLODYMYR, NIKONOV OLEG, DMYTRO NOVAK

Kyiv National University of Technologies and Design

INTEGRATION OF TRADITIONAL AND MODERN WEB TECHNOLOGIES IN THE TRAINING OF JUNIOR SPECIALISTS IN THE FIELD OF FRONT-END DEVELOPMENT

The article explores the integration of traditional and modern approaches in the training of Junior specialists in the field of front-end development. Emphasis is placed on the significance of HTML and CSS as foundational technologies that serve as the basis for acquiring core competencies in web development. The traditional methodology, which relies on a consistent, step-by-step mastery of syntax, manual code writing, and basic layout exercises, is analyzed as an effective model for building essential skills at the initial stages of professional training.

However, given the accelerated pace of technological progress and the rising complexity of web applications, the article highlights the growing necessity of incorporating modern front-end tools and frameworks such as React, Vue.js, and Tailwind CSS. These technologies provide enhanced opportunities for building dynamic, scalable, and maintainable web interfaces and are increasingly demanded by employers. A comparative analysis of these frameworks is presented, covering their architecture, design philosophy, syntax, and suitability for learners at the beginner level.

The study supports a blended instructional approach that begins with traditional technologies and progressively integrates contemporary tools. This educational trajectory promotes a deeper understanding of the principles underlying front-end development, while simultaneously equipping learners with the practical skills needed in real-world environments. The article concludes by outlining prospects for further research, emphasizing the need to design adaptive, modular, and scalable educational models that align with evolving industry standards and technological innovations. These findings will be especially relevant to educators, curriculum designers, and institutions aiming to modernize training programs and improve the employability of graduates in the competitive IT labor market.

Keywords: HTML, CSS, front-end development, React, Vue.js, Tailwind CSS.

Стаття надійшла до редакції / Received 22.05.2025

Прийнята до друку / Accepted 26.06.2025

Постановка проблеми у загальному вигляді та її зв'язок із важливими науковими чи практичними завданнями

Подальший розвиток сучасних веб-технологій можливий за умови системного аналізу етапів становлення та основних тенденцій у галузі веб-програмування. З огляду на зростаючі вимоги до функціональності, продуктивності та інтерактивності веб-додатків, особливої актуальності набуває дослідження еволюції підходів, інструментів і архітектурних рішень у розробці веб-систем.

Аналіз наукових праць [1–3] свідчить про те, що веб-програмування пройшло низку важливих етапів — від статичних HTML-сторінок до появи складних архітектур на основі компонентів, бібліотек і фреймворків. На сьогодні широке поширення отримали такі підходи, як JAMstack, використання PWA, WebAssembly, а також інтенсивна інтеграція клієнтських і серверних технологій. Важливими аспектами є також застосування систем автоматизації (Webpack, Vite), оптимізації рендерингу та створення масштабованих інтерфейсів.

Аналіз досліджень та публікацій

У роботах [1, 6] наведено дані про ефективність використання проєктно-орієнтованого навчання при викладанні HTML і CSS. Автори досліджували освітній процес у технічних ВНЗ і довели, що традиційні підходи, які базуються на поступовому засвоєнні синтаксису та структури мов розмітки, дозволяють студентам закласти фундаментальні знання для створення веб-інтерфейсів. При цьому, як зазначено в роботі, студенти краще засвоюють матеріал за умов активного залучення в практичні завдання, ніж при використанні виключно теоретичних методик. Це узгоджується з висновками інших дослідників, які наголошують на важливості

реального застосування знань у процесі навчання для розвитку компетенцій у сфері веб-програмування.

Разом з тим, сучасна веб-розробка дедалі більше орієнтується на використання бібліотек і фреймворків. У статтях [2, 3] наведено огляд особливостей фреймворку React, який забезпечує розробку інтерфейсів на основі компонентного підходу. Автори акцентують увагу на тому, що React суттєво змінює парадигму традиційного HTML/CSS-програмування, оскільки дозволяє будувати повторно використовувані модулі інтерфейсу з керованим станом. Впровадження таких рішень у навчальний процес вимагає переосмислення методик викладання, оскільки потребує від студентів не лише знань розмітки, а й володіння концепціями функціонального програмування [3, 7, 12].

У статтях [4, 14] здійснено порівняльний аналіз популярних JavaScript-фреймворків, зокрема React і Vue. Автори підкреслюють, що обидва підходи забезпечують високий рівень продуктивності розробки, але мають різні концептуальні основи. Зокрема, Vue орієнтований на простоту й гнучкість, що робить його придатним для початківців, у той час як React краще підходить для масштабованих проєктів. На думку дослідників, інтеграція таких інструментів у навчальний процес дозволяє формувати в студентів практичні навички, що відповідають сучасним вимогам IT-ринку.

Також у роботах [4, 5, 6] розглянуто використання CSS-фреймворку Tailwind CSS, який надає утилітарний підхід до стилізації інтерфейсів. У дослідженні доведено, що Tailwind значно скорочує час верстки завдяки готовим класам і покращує структуру коду, що особливо важливо у командній роботі. Це свідчить про тенденцію до спрощення та уніфікації стилізації у сучасній фронтенд-розробці. Автори рекомендують включати Tailwind у навчальні курси як інструмент швидкої прототипізації та стилізації, що полегшує перехід від базового HTML/CSS до сучасного підходу в UI/UX розробці.

Таким чином, проведений огляд літератури показує, що традиційне використання HTML і CSS є важливою основою для формування базових знань у сфері веб-програмування. Проте, враховуючи швидкий розвиток технологій, ефективним є доповнення таких курсів вивченням сучасних фреймворків, які формують актуальні навички, затребувані на ринку праці. Подальше дослідження вимагає визначення оптимального балансу між традиційними та сучасними підходами до навчання веб-технологіям.

Формулювання цілей статті

Метою роботи є: дослідити ефективність поєднання традиційних засобів веброзробки (HTML5, CSS3) із сучасними фреймворками (React, Vue.js, Tailwind CSS) у підготовці фахівців початкового рівня.

Завдання:

1. Проаналізувати переваги та обмеження кожного підходу.
2. Обґрунтувати доцільність впровадження комбінованої освітньої моделі, яка відповідатиме сучасним вимогам IT-індустрії.

Виклад основного матеріалу

HTML і CSS є базовими технологіями у сфері веб-розробки, що традиційно розглядаються як основа підготовки фахівців рівня Junior. Згідно з [1], класичний підхід до їх вивчення передбачає поступове освоєння ключових понять, написання коду вручну та виконання типових практичних завдань. Методика побудована на поетапному розкритті матеріалу: від базових конструкцій до складніших аспектів верстки й стилізації.

Однією з визначальних рис є лінійна структура навчання, в межах якої майбутні Junior-фахівці опановують основні теги HTML (заголовки, абзаци, зображення, посилання) і базові властивості CSS (кольори, шрифти, відступи, вирівнювання). Практика написання коду в редакторах на зразок Notepad чи Visual Studio Code сприяє глибшому розумінню синтаксису мов і механізмів їх взаємодії без залучення інструментів автоматизації.

Практичні завдання, які орієнтовані на створення статичних веб-сторінок із текстовим вмістом, таблицями та зображеннями, формують фундаментальні навички верстки. Додатково підкреслюється важливість правильного використання семантичних тегів для підвищення доступності контенту. Також часто застосовуються готові шаблони HTML/CSS для пришвидшення старту роботи [12].

Переваги традиційного підходу представлені на рисунку.



Рис.1. Переваги HTML/CSS при створенні web-розробок (створення автором)

Таким чином, традиційний підхід до навчання веброзробці є ефективною стартовою моделлю для формування базових професійних компетенцій, оскільки поєднує доступність, простоту впровадження та чітке засвоєння фундаментальних навичок, необхідних для подальшого освоєння сучасних технологій та інструментів у сфері цифрової розробки.

Однак, існують і недоліки, які показані на рисунку 2.

Таким чином, традиційна методика є добрим стартом для підготовки фахівців рівня Junior, однак для ефективної адаптації до сучасних вимог вона має бути доповнена інноваційними технологіями веб-розробки [10].



Рис.2. Недоліки використання HTML/CSS при створенні web- додатків (створення автором)

Сучасна індустрія вимагає від Junior-фахівців знань не лише HTML і CSS, а й навичок використання сучасних фреймворків і бібліотек, що підвищують ефективність та гнучкість розробки. У цьому контексті найбільш релевантними є React, Vue.js і Tailwind CSS [4; 5]. У таблиці 1 наведена характеристика цих технологій та вимоги для засвоєння фахівцями рівня Junior.

Таблиця 1

Порівняльна характеристика сучасних веб-технологій (створення автором)

Характеристика	React	Vue.js	Tailwind CSS
Тип	JavaScript-бібліотека	JavaScript-фреймворк	CSS-фреймворк
Призначення	Розробка інтерфейсів користувача	Розробка інтерфейсів користувача	Стилізація елементів інтерфейсу
Підхід	Компонентний, декларативний	Компонентний, реактивний	Утилітарні CSS-класи
Ключова перевага	Віртуальний DOM	Простий синтаксис	Швидка стилізація
Кривизна навчання	Середня	Низька	Низька
Сфера застосування	SPA, веб-додатки	Будь-які проєкти	Будь-які веб-проєкти
Підтримка екосистеми	Висока (Redux, Router)	Висока (Vuex, Router)	Висока (CLI, плагіни)
Придатність для Junior	Потребує знань JavaScript	Зрозумілий для новачків	Підходить для старту

Проаналізуємо більш ретельно ці застосунки [7, 12, 13]:

React: компонентна модель і віртуальний DOM

React — це JavaScript-бібліотека, орієнтована на розробку користувацьких інтерфейсів за допомогою повторно використовуваних компонентів. Її архітектура передбачає поділ інтерфейсу на ізольовані модулі (компоненти), що сприяє підвищенню масштабованості проєктів. Однією з ключових інновацій є використання віртуального DOM, який дозволяє зменшити кількість звернень до реального DOM і таким чином оптимізує продуктивність. Додаткову гнучкість забезпечують React Hooks — механізм управління станом усередині функціональних компонентів. У межах екосистеми доступні численні інструменти, зокрема React Router (для маршрутизації) та Redux (для глобального управління станом).

Vue.js: прогресивність і спрощена інтеграція

Vue.js позиціонується як прогресивний фреймворк, що забезпечує швидке впровадження та поступове розширення функціональності. Його особливістю є двобічна реактивність, яка синхронізує дані моделі з елементами інтерфейсу в режимі реального часу. Система компонентів у Vue дозволяє структурувати

код згідно з принципами повторного використання, а зрозумілий синтаксис робить фреймворк доступним для початківців. Крім того, Vue демонструє високу сумісність із традиційними HTML-шаблонами, що полегшує інтеграцію у наявні проекти.

Tailwind CSS: утилітарний підхід до стилізації

Tailwind CSS є утилітарним CSS-фреймворком, який підтримує створення дизайну без необхідності формування окремих CSS-файлів. Основною його перевагою є можливість швидкого прототипування інтерфейсів за рахунок використання готових класів для стилізації елементів. Tailwind надає широкі можливості адаптації стилів під конкретні вимоги проекту та підтримує очищення стилів (purging) для зменшення обсягів кінцевого CSS-коду. Крім того, він орієнтований на побудову адаптивного дизайну за допомогою класів для медіазапитів, що сприяє кросплатформеній доступності інтерфейсів.

Таким чином, React, Vue.js та Tailwind CSS репрезентують різні підходи до організації фронтенд-розробки: від компонентно-орієнтованої архітектури до утилітарного стилювання. Вибір конкретного інструменту залежить від цілей проекту, технічних вимог та рівня підготовки розробника. Застосування таких інструментів сприяє підвищенню ефективності розробки, якості користувацьких інтерфейсів і забезпечує відповідність сучасним стандартам вебтехнологій.

Висновки з даного дослідження

і перспективи подальших розвідок у даному напрямі

Проведене дослідження засвідчує, що традиційні вебтехнології, такі як HTML5 та CSS3, залишаються фундаментом для створення веб-інтерфейсів і відіграють ключову роль у формуванні базових навичок майбутніх розробників. Проте в умовах стрімкої еволюції цифрових продуктів вони вже не можуть забезпечити достатню функціональність для створення інтерактивних та масштабованих вебзастосунків.

Сучасні фреймворки, зокрема React, демонструють істотні переваги щодо продуктивності, повторного використання коду, підтримки компонентного підходу та роботи з віртуальним DOM, що робить їх незамінними інструментами в сучасній фронтенд-розробці. Вони забезпечують високу інтерактивність, адаптивність і ефективність, проте вимагають від користувачів вищого рівня технічної підготовки.

Отже, доцільним є впровадження змішаного підходу в навчання: спершу опановувати базові технології, а згодом — сучасні фреймворки. Такий поступовий перехід сприятиме глибшому розумінню принципів веброзробки, формуванню повноцінних професійних компетенцій та підвищенню конкурентоспроможності майбутніх фахівців на ринку праці.

Перспективи подальших досліджень полягають у розробці інтегрованих освітніх траєкторій, модернізації методик викладання з поетапним впровадженням фреймворків, аналізі впливу сучасних технологій на підготовку Junior-фахівців та створенні адаптивних модульних програм відповідно до вимог IT-індустрії.

Література

1. Гордієнко Т. С., Пилипенко О. В. Тенденції розвитку веб-програмування у 2020-ті роки // Інформаційні технології та комп'ютерна інженерія. – 2021. – № 3(68). – С. 25–31.
2. Новак А. С. Інформаційна система для вибіркових дисциплін. – Київ: КПІ ім. Ігоря Сікорського, 2022. – 63 с.
3. López-Pimentel J. C., Medina-Santiago A., Alcaraz-Rivera M., Del-Valle-Soto C. Sustainable Project-Based Learning Methodology Adaptable to Technological Advances for Web Programming // Sustainability. – 2021. – Vol. 13, No. 15. – P. 8482. DOI: <https://doi.org/10.3390/su13158482>
4. Apriansyah A., Haifaturrahmah N., Sari N., Alaa S. Effect of Project-Based Learning Through Blended Learning on Website Design Skills [Electronic resource] // ResearchGate. – 2024. – Access mode: https://www.researchgate.net/publication/352647061_Effect_of_Project-Based_Learning_Through_Blended_Learning_on_Website_Design_Skills
5. Kim M. C. Learning Early-Stage Web Development at Scale: Exploring Methods to Assess Learning Through Analysis of HTML and CSS // ProQuest Dissertations & Theses. – 2021. – Access mode: <https://search.proquest.com/openview/8dc55704c3e9200d5fad777b6c5415d9/1?cbl=18750&diss=y&pq-origsite=gscholar>
6. Zhou Y., Han J. Machine Learning for Web Development: A Comprehensive Survey // IEEE Access. – 2020. – Vol. 8. – P. 145345–145364. DOI: 10.1109/ACCESS.2020.3014632
7. Розгон Ю., Чортюк М. Цифрові репутаційні перспективи України як фактор конкурентоспроможності в умовах євроінтеграції // Економіка та суспільство. – 2022. – Вип. 38. – С. 45–50.
8. Lindley C. The Front-End Developer/Engineer Handbook 2024 [Electronic resource]. – Frontend Masters, 2024. – Access mode: <https://frontendmasters.com/guides/front-end-handbook/2024/>
9. Детальна програма курсу Frontend розробник від Logos IT Academy [Electronic resource]. – Access mode: <https://frontend.lviv.ua/programa-navchannya>
10. Гошко С. О. Пояснювальна записка [Electronic resource]. – Львів: Львівський національний університет імені Івана Франка, 2023. – Access mode: <https://clio.lnu.edu.ua/wp-content/uploads/2023/03/Hoshko-S.O..pdf>

11. І Всеукраїнська науково-практична конференція молодих вчених [Electronic resource]. – Університет імені Альфреда Нобеля, м. Дніпро, 2020. – Access mode: https://duan.edu.ua/images/staff/departments/IT/Files/Conferences/IT_conf_15_May_2020.pdf
12. Mimo. How to Become a Front-End Developer in 2025: The Ultimate Guide [Electronic resource]. – 2024. – Access mode: <https://mimo.org/blog/how-to-become-a-front-end-developer>
13. Skillcrush. 15 Essential Front End Developer Skills (+ Expert Insights) [Electronic resource]. – 2022. – Access mode: <https://skillcrush.com/blog/skills-to-become-a-front-end-developer/>
14. The Software House. State of Frontend 2024 [Electronic resource]. – 2024. – Access mode: <https://tsh.io/state-of-frontend>.

References

1. Hordienko T. S., Pylypenko O. V. Tendentsii rozvytku veb-prohramuvannia u 2020-ti roky // Informatsiini tekhnolohii ta komp'iuterna inzheneriia. – 2021. – № 3(68). – S. 25–31.
2. Novak A. S. Informatsiina systema dlia vybirkovykh dystsyplin. – Kyiv: KPI im. Ihoria Sikorskoho, 2022. – 63 s.
3. López-Pimentel J. C., Medina-Santiago A., Alcaraz-Rivera M., Del-Valle-Soto C. Sustainable Project-Based Learning Methodology Adaptable to Technological Advances for Web Programming // Sustainability. – 2021. – Vol. 13, No. 15. – P. 8482. DOI: <https://doi.org/10.3390/su13158482>
4. Apriansyah A., Haifaturrahmah N., Sari N., Alaa S. Effect of Project-Based Learning Through Blended Learning on Website Design Skills [Electronic resource] // ResearchGate. – 2024. – Access mode: https://www.researchgate.net/publication/352647061_Effect_of_Project-Based_Learning_Through_Blended_Learning_on_Website_Design_Skills
5. Kim M. C. Learning Early-Stage Web Development at Scale: Exploring Methods to Assess Learning Through Analysis of HTML and CSS // ProQuest Dissertations & Theses. – 2021. – Access mode: <https://search.proquest.com/openview/8dc55704c3e9200d5fad777b6c5415d9/1?cbl=18750&diss=y&pq-origsite=gscholar>
6. Zhou Y., Han J. Machine Learning for Web Development: A Comprehensive Survey // IEEE Access. – 2020. – Vol. 8. – P. 145345–145364. DOI: 10.1109/ACCESS.2020.3014632
7. Rozhon Yu., Chortok M. Tsyfrovi reputatsiini perspektyvy Ukrainy yak faktor konkurentospromozhnosti v umovakh yevrointegratsii // Ekonomika ta suspilstvo. – 2022. – Vyp. 38. – S. 45–50.
8. Lindley C. The Front-End Developer/Engineer Handbook 2024 [Electronic resource]. – Frontend Masters, 2024. – Access mode: <https://frontendmasters.com/guides/front-end-handbook/2024/>
9. Detalna prohrama kursu Frontend rozrobnyk vid Logos IT Academy [Electronic resource]. – Access mode: <https://frontend.lviv.ua/programa-navchannya>.
10. Hoshko S. O. Poyasniuvalna zapyska [Electronic resource]. – Lviv: Lvivskyi natsionalnyi universytet imeni Ivana Franka, 2023. – Access mode: <https://clio.lnu.edu.ua/wp-content/uploads/2023/03/Hoshko-S.O..pdf>
11. І Всеукраїнська науково-практична конференція молодих вчених [Electronic resource]. – Університет імені Альфреда Нобеля, м. Дніпро, 2020. – Access mode: https://duan.edu.ua/images/staff/departments/IT/Files/Conferences/IT_conf_15_May_2020.pdf
12. Mimo. How to Become a Front-End Developer in 2025: The Ultimate Guide [Electronic resource]. – 2024. – Access mode: <https://mimo.org/blog/how-to-become-a-front-end-developer>
13. Skillcrush. 15 Essential Front End Developer Skills (+ Expert Insights) [Electronic resource]. – 2022. – Access mode: <https://skillcrush.com/blog/skills-to-become-a-front-end-developer/>
14. The Software House. State of Frontend 2024 [Electronic resource]. – 2024. – Access mode: <https://tsh.io/state-of-frontend>.