

УДК 004.8

Тарас Х. - ст. гр. СН-43

Тернопільський національний технічний університет імені Івана Пулюя

ДИЗАЙНЕРСЬКІ КОМПЕТЕНЦІЇ ПРОГРАМІСТА У КОНТЕКСТІ ВИКОРИСТАННЯ TAILWIND CSS

Науковий керівник: к.т.н, доцент Дмитроца Л.П.

Taras K.

Ternopil Ivan Puluj National Technical University

DESIGN COMPETENCES OF A PROGRAMMER IN THE CONTEXT OF USING TAILWIND CSS

Supervisor: PhD, Associate Professor Dmytrotsa L.P.

Ключові слова: вебінтерфейс, фронтенд-розробка, UI/UX дизайн

Keywords: web interface, front-end development, UI/UX design

У сучасному світі веброзробки все частіше стираються межі між ролями, які ще донедавна вважалися окремими професійними спеціалізаціями. Програміст більше не є виключно технічним виконавцем, а дизайнер – не лише творцем естетичного вигляду інтерфейсів. У нових умовах ринку інформаційних технологій дедалі більшу цінність набувають фахівці, які здатні поєднувати аналітичне мислення з візуальною грамотністю, створюючи не лише функціональні, а й візуально привабливі цифрові продукти.

Фронтенд-розробка переживає активну трансформацію, що зумовлена зростанням вимог до користувацького досвіду та естетики вебінтерфейсів. Якщо раніше розробник здебільшого відповідав за функціональність і коректну роботу елементів інтерфейсу, то сьогодні його зона відповідальності значно ширша. Користувачі очікують не просто працюючий сайт, а привабливе, інтуїтивно зрозуміле середовище, у якому взаємодія з елементами інтерфейсу відбувається природно і легко. Це, у свою чергу, вимагає від фронтенд-розробника не лише знання мов програмування, бібліотек і фреймворків, а й базових навичок дизайну та розуміння принципів UI/UX.

Зміна ринку праці підтверджує цю тенденцію [1]. У більшості вакансій на позицію frontend developer серед обов'язкових або бажаних вимог зазначаються знання системи дизайну, досвід роботи з Figma, Adobe XD, а також розуміння принципів побудови інтерфейсів. Такий підхід обумовлений необхідністю оптимізувати командну роботу, зменшити кількість ітерацій при погодженні макетів і спростити адаптацію дизайну під різні пристрої та екрани.

Крім того, в умовах зростання кількості стартапів та невеликих проєктів, де часто немає окремих дизайнерів, саме на програміста покладається обов'язок створення візуальної частини продукту. У таких випадках його здатність до естетичного мислення, чуття пропорцій, контрасту, ритму та гармонії стає не менш важливою, ніж технічна грамотність. Таким чином, сучасний фронтенд – це синтез технологій і дизайну, де програміст дедалі частіше виконує роль повноцінного творця інтерфейсу.

З-поміж сучасних CSS-інструментів особливе місце займає Tailwind CSS – фреймворк, що реалізує концепцію utility-first [2], за якої розробник оперує

низькорівневими класами без необхідності створювати кастомні стилі. Унікальність Tailwind CSS полягає в тому, що він дозволяє будувати інтерфейс безпосередньо в HTML-структурі, використовуючи зрозумілі, логічно названі класи. Це робить його надзвичайно привабливим для програмістів, які не мають професійного дизайнерського досвіду.

Фреймворк орієнтований на швидкість і гнучкість. Замість того, щоб писати власні стилі або адаптовувати готові компоненти з дизайн-систем, розробник може «збирати» інтерфейс із будівельних блоків, орієнтуючись на технічну логіку. Tailwind забезпечує точний контроль над кожним елементом, а також передбачає систему кольорових палітр, відступів, розмірів і шрифтів, які вже оптимізовані для зручного візуального сприйняття.

Завдяки Tailwind CSS програмісти отримали інструмент, який не лише пришвидшує процес розробки, а й дозволяє створювати візуально якісні інтерфейси навіть без глибокого занурення у дизайнерські концепції. Tailwind значно знижує бар'єр входу у візуальну частину розробки, пропонуючи певну «дизайн-підказку» через систему класів, що базуються на перевірених UI-практиках.

Однак така уніфікована система не гарантує створення продуманого й ефективного дизайну. Tailwind CSS – це лише інструмент, і те, наскільки вдало він буде використаний, залежить саме від компетенцій розробника, зокрема його вміння бачити інтерфейс не лише як набір блоків, а як цілісний продукт із власною логікою взаємодії, естетикою та користувацькою цінністю.

Попри функціональність і зручність Tailwind CSS, його використання не є гарантією створення якісного та ефективного дизайну. Фреймворк надає набір готових інструментів, але не замінює дизайнерське бачення. Без розуміння основ композиції, принципів візуальної ієрархії, типографіки, балансу кольору та ритму інтерфейс, зібраний за допомогою Tailwind, ризикує залишитися лише технічно коректним, але візуально слабким або навіть незручним для користувача.

Програміст, який не володіє дизайнерськими навичками, може легко створити перевантажений або дезорганізований інтерфейс, де важливі елементи губляться, а користувачеві важко зорієнтуватися.

Наприклад, недотримання контрастності може ускладнити читання тексту, а надлишок декоративних елементів – відволікти від основної функції сторінки. У реальних проектах такі помилки можуть призвести до зниження зручності користування, погіршення досвіду користувача та втрати клієнтів.

Ще одним важливим аспектом є розуміння контексту взаємодії користувача з інтерфейсом. Tailwind не дає відповіді на питання, які дії очікуються від користувача, які елементи повинні бути помітними в першу чергу або як адаптувати інтерфейс до різних сценаріїв використання. Ці питання належать до сфери UX-дизайну [3], який виходить за межі лише візуальної стилізації. Без урахування таких аспектів навіть естетично вивірений інтерфейс не зможе забезпечити зручну та ефективну взаємодію з користувачем.

Таким чином, Tailwind CSS може стати потужним інструментом у руках програміста, але лише за наявності хоча б базових дизайнерських компетенцій. Без них використання фреймворку перетворюється на механічне компонування елементів, позбавлене логіки, глибини і стратегічного бачення інтерфейсу як цілісної системи.

Здатність програміста мислити візуально й володіти основами дизайну відкриває перед ним нові можливості та переваги в професійному середовищі.

По-перше, розробник із дизайнерським мисленням краще розуміє потреби користувача. Він здатен проектувати інтерфейс не лише з погляду функціональності, а й з урахуванням логіки взаємодії, зручності сприйняття інформації та естетичних

очікувань аудиторії. Це особливо важливо у фронтенд-розробці, де саме візуальний контакт є першим пунктом взаємодії з продуктом.

По-друге, дизайнерські навички програміста сприяють підвищенню його автономності. У невеликих командах або фріланс-проектах, де немає окремого UI/UX-дизайнера, вміння самостійно створювати макети, компоновати інтерфейс і приймати стилістичні рішення стає великою перевагою. Tailwind CSS у цьому випадку працює як ефективний інструмент реалізації візуальних ідей без потреби писати CSS з нуля.

По-третє, програмісти з дизайнерським досвідом мають кращу комунікацію з командами дизайнерів. Вони швидше розуміють вимоги до верстки, логіку адаптивного дизайну, специфіку кольорових схем та типографіки, а також можуть запропонувати адекватні технічні рішення для реалізації креативних ідей.

У результаті, програміст із дизайнерськими навичками перетворюється на універсального фахівця, здатного створювати повноцінні продукти – від логіки до візуальної оболонки. Tailwind CSS лише підсилює цей ефект, виступаючи зручним середовищем для реалізації дизайну «зсередини коду».

Висновок. У сучасній фронтенд-розробці, де інтерфейс є ключовим елементом взаємодії з користувачем, дизайнерські компетенції програміста набувають особливої ваги. Tailwind CSS значно спрощує процес створення інтерфейсів, але сам по собі не замінює дизайнерського мислення. Здатність програміста орієнтуватися в принципах візуальної композиції, кольорів, типографіки та логіки UX дозволяє не лише ефективніше використовувати Tailwind, а й створювати більш цілісні, зручні та естетично привабливі продукти. Таким чином, синергія технічних і дизайнерських навичок стає запорукою професійної гнучкості та конкурентоспроможності розробника у цифровій екосистемі.

Література

1. Скороход Д. Що має знати Senior Front-end Developer. Результати аналізу вакансій в Україні та Каліфорнії. DOU.UA. URL: <https://dou.ua/lenta/articles/knowledge-for-senior-front-end-developer/> (дата звернення: 11.04.2025).
2. Szymanski M. Utility-first approach. TW Elements. URL: <https://tw-elements.com/learn/te-foundations/tailwind-css/utility-first/> (дата звернення: 12.04.2025).
3. Stevens E. What Is User Experience (UX) Design? Everything You Need to Know. CareerFoundry. 22.01.2025. URL: <https://careerfoundry.com/en/blog/ux-design/what-is-user-experience-ux-design-everything-you-need-to-know-to-get-started/> (дата звернення: 12.04.2025).