

ширше – бачити проблеми, знаходити рішення й створювати щось нове. Поєднання науки, освіти, міжнародного досвіду та співпраці з компаніями допомагає випускникам не просто знаходити роботу, а змінювати світ навколо себе.

### Список використаних джерел

1. Електронний кампус КПІ ім. Ігоря Сікорського. URL: <https://ecampus.kpi.ua/> (дата звернення: 05.11.2025).
2. КПІ ім. Ігоря Сікорського. Міжнародне співробітництво. URL: <https://kpi.ua/cooperation> (дата звернення: 05.11.2025).
3. Інститут післядипломної освіти КПІ ім. Ігоря Сікорського. Підвищення кваліфікації науково-педагогічних працівників КПІ ім. Ігоря Сікорського в ІПО у 2023–2024 н.р. URL: <https://ipi.kpi.ua/pidvyshhennya-kvalifikatsiyi-naukovo-pedagogichnyh-pratsivnykiv-kpi-im-igorya-sikorskogo-v-ipou-2023-2024-n-r/> (дата звернення: 05.11.2025).
4. Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського». Річний звіт за 2023 рік URL: Київ, 2023. 127 с. URL: <https://kpi.ua/files/2023-report.pdf> (дата звернення: 05.11.2025).

## ПОРІВНЯЛЬНИЙ АНАЛІЗ ІНТЕРНЕТ-ПЛАТФОРМ ДЛЯ ВИВЧЕННЯ ВЕБПРОГРАМУВАННЯ У СТАРШІЙ ШКОЛІ

### Мазур Анастасія Сергіївна

здобувач третього рівня вищої освіти, спеціальність Середня освіта (Інформатика)  
Український державний університет імені М. Драгоманова  
[a.s.mazur@udu.edu.ua](mailto:a.s.mazur@udu.edu.ua)

### Габрусєв Валерій Юрійович

кандидат педагогічних наук, доцент кафедри інформатики та методики її навчання  
Тернопільський національний педагогічний університет імені Володимира Гнатюка  
[gabrusev@fizmat.tnpu.edu.ua](mailto:gabrusev@fizmat.tnpu.edu.ua)

Сучасний етап розвитку освіти характеризується активним впровадженням цифрових технологій у навчальний процес. Особливо актуальним це є для викладання інформатики, а саме вибіркового модуля «Вебтехнології», де інтернет-технології не лише виступають засобом навчання, але й відображають реальне професійне середовище майбутніх фахівців, що є одним із ключових аспектів НУШ та профільної освіти. За даними міжнародних досліджень, попит на веб-розробників зростає щорічно на 13–15 % [3], що робить навчання веб-програмування однією з найважливіших складових підготовки учнів до майбутньої професійної діяльності. У цьому контексті якість шкільної підготовки з веб-програмування набуває особливого значення, оскільки саме в старшій школі формуються базові компетентності, що визначають подальшу професійну траєкторію випускників.

Традиційні методи викладання програмування, що базуються на використанні локальних середовищ розробки та статичних підручників, поступово втрачають свою ефективність. Вони не відповідають темпам розвитку веб-технологій, не забезпечують достатнього рівня інтерактивності та не створюють умов для формування навичок колаборативної роботи. Водночас розвиток інтернет-технологій відкриває нові можливості для організації навчального процесу через використання спеціалізованих онлайн-платформ.

Навчання програмування в сучасній школі переживає період інтенсивної трансформації. Якщо ще десятиліття тому програмування розглядалося як вузькоспеціалізована дисципліна для учнів з особливими здібностями до математики та логіки, то сьогодні воно все більше сприймається як універсальна компетентність, необхідна для успішної адаптації до цифрового суспільства.

Веб-програмування має ряд особливостей, що відрізняють його від інших видів програмування та визначають специфіку його викладання в школі. Веб-програмування є мультидисциплінарним. Воно поєднує в собі програмування (JavaScript), розмітку (HTML), стилізацію (CSS), роботу з графікою, розуміння принципів дизайну та User Experience. Це робить веб-розробку більш різноманітною та цікавою для учнів, але водночас створює виклики щодо комплексності навчання.

Сучасний ринок освітніх технологій пропонує широкий спектр інтернет-платформ для навчання веб-програмуванню: від простих онлайн-редакторів коду до комплексних систем управління навчанням. Кожна з цих платформ має свої особливості, переваги та обмеження. Однак відсутність системного порівняльного аналізу цих інструментів за критеріями, релевантними саме для шкільної освіти, створює значні труднощі для вчителів інформатики при виборі оптимального навчального інструментарію.

Інтернет-платформа має забезпечувати можливість вивчення тем, передбачених чинною навчальною програмою з інформатики у розрізі вибіркового модуля «Веб-технології» для старшої школи. Згідно з навчальною програмою, учні мають оволодіти основами HTML (структура веб-сторінок, семантична розмітка, форми), CSS (селектори, блокова модель, позиціонування, адаптивний дизайн) та JavaScript (змінні, типи даних, умовні конструкції, цикли, функції, робота з DOM, обробка подій).

Важливо, щоб платформа підтримувала сучасні стандарти веб-технологій (HTML5, CSS3, ES6+), оскільки навчання застарілих технологій не має практичного сенсу. Водночас платформа не повинна бути надто складною, включаючи просунуті фреймворки та бібліотеки, які виходять за межі шкільної програми та можуть перевантажити учнів. Огляд Веб-платформ для Навчання Веб-програмуванню.

Вибір ефективної веб-платформи є ключовим для організації сучасного навчального процесу з веб-програмування. Наведений нижче аналіз порівнює шість основних інструментів – від простих онлайн-редакторів коду (CodePen, JSFiddle) до повноцінних хмарних IDE з функціями управління класом (Replit) та професійних систем контролю версій (GitHub Classroom). Кожна платформа має унікальні переваги та недоліки щодо інтерактивності, автоматичної перевірки, наявності освітніх інструментів та кривої навчання, що є критичним при виборі оптимального рішення для викладання в школі.

CodePen є однією з найпопулярніших платформ («ігровий майданчик для фронтенд-розробників»), заснованою у 2012 році. Її основна перевага – це миттєвий візуальний зворотний зв'язок завдяки функції live preview. CodePen пропонує потужний редактор коду з автодоповненням та підтримкою препроцесорів. Він ідеально підходить для швидких вправ, демонстрацій на уроках та створення портфоліо учнів. Платформа має потужну соціальну складову, що підвищує мотивацію, проте безкоштовна версія має обмеження на приватні проєкти та не містить повноцінних інструментів для вчителя.

JSFiddle, запущений у 2009 році, є піонером серед онлайн-редакторів коду. Він більш мінімалістичний та утилітарний, популярний для створення мінімальних прикладів коду (minimal reproducible examples) для тестування та діагностики. На відміну від CodePen, JSFiddle вимагає натискання кнопки «Run» для оновлення результату. Його переваги включають простоту, швидкість та зручну систему версіонування. Однак, він має застарілий дизайн, мінімальні соціальні функції та майже повну відсутність інструментів для вчителя.

Replit – це потужне хмарне інтегроване середовище розробки (IDE), що підтримує понад 50 мов програмування. Його ключова перевага для освіти – версія «Teams for Education». Ця версія надає вчителям найкращий серед усіх платформ набір інструментів: створення класів, призначення завдань зі стартовим кодом, автоматичну перевірку за допомогою юніт-тестів та моніторинг прогресу учнів. Replit дозволяє створювати багатофайлові, повноцінні веб-проекти та підтримує командну розробку (мультиплеєр-режим). Водночас, інтерфейс може бути складним для початківців, а повноцінна освітня версія є платною.

FreeCodeCamp – це безкоштовна некомерційна платформа, що є повноцінною навчальною програмою з інтерактивними вправами та проектами. Вона пропонує структуровані сертифікаційні курси, наприклад, «Responsive Web Design» та «JavaScript Algorithms and Data Structures». Методика навчання базується на інтерактивних вправах з автоматичною перевіркою. Це чудове джерело якісного, безкоштовного та структурованого контенту. Основний недолік для школи – орієнтація на самостійне навчання, повна відсутність інструментів для управління класом та неможливість адаптувати обсяг матеріалу під шкільну програму.

W3Schools – один із найстаріших та найпопулярніших веб-ресурсів для вивчення веб-технологій, що функціонує як великий довідник та колекція туторіалів. Його ключова особливість – вбудований інтерактивний редактор «Try it Yourself» для експериментів з кодом. Він надає найбільш повний та простий довідковий матеріал з HTML, CSS та JavaScript. W3Schools ідеально підходить як довідковий ресурс та джерело прикладів для швидких демонстрацій. Недоліками є відсутність інструментів для вчителів, спрощений підхід до матеріалу та наявність реклами.

GitHub Classroom – це безкоштовний сервіс для організації та управління завданнями з програмування, що використовує професійні інструменти Git та GitHub. Він дозволяє вчителям створювати індивідуальні та групові завдання, налаштовувати автоматичне тестування (через GitHub Actions) та проводити Code Review. Головна перевага – навчання учнів професійним навичкам контролю версій та колаборації. Однак, Git та GitHub мають високу криву навчання і не рекомендуються для учнів, які тільки починають вивчати HTML, тому його варто впроваджувати лише у старших класах (11 клас).

Резюме для шкільного навчання: CodePen є найкращим інструментом для швидкої практики та візуальних демонстрацій. freeCodeCamp – чудове структуроване доповнення для самостійної роботи. Replit – оптимальний вибір для організації повноцінного курсу завдяки професійному IDE та потужним інструментам для вчителя. GitHub Classroom варто впроваджувати поступово, починаючи з 11-го класу, для підготовки до професійної діяльності.

### Список використаних джерел

1. Деревенко А. М., Ільїна Т. В., Ібрагімова Л. А. Використання цифрових платформ для підвищення якості професійної освіти. *Академічні візії*, 2024. № 31. С. 1–12.
2. Гнатишин М. Аналіз сучасних тенденцій розвитку технологій веб-розробки. *Природничі та гуманітарні науки. актуальні питання* : матеріали І Міжнар. студ. наук.-техн. конф. 2023. С. 132.
3. Зріз ринку цифрової розробки України за 2024 рік. URL: <https://it-rating.ua/snapshot-web-development-market-ukraine-2024>. (дата звернення: 02.11.2025р.).

## ОПТИМІЗАЦІЯ ПРОДУКТИВНОСТІ ІГОР, СТВОРЕНИХ З ВИКОРИСТАННЯМ РУШІЯ GODOT ТА C#

### Мельник Петро Петрович

здобувач другого рівня вищої освіти, спеціальність Комп'ютерні науки  
Тернопільський національний педагогічний університет імені Володимира Гнатюка  
[melnuk\\_pp@fizmat.tnpu.edu.ua](mailto:melnuk_pp@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](mailto:yava@fizmat.tnpu.edu.ua)

У сучасній індустрії відеоігор продуктивність – не просто «приємна властивість», а фундаментальна вимога до успішного проєкту: гравці очікують плавної роботи, швидкого завантаження і відсутності затримок. Особливо це стосується 3D-ігор, мобільних платформ та кросплатформених рішень, де апаратні ресурси можуть бути обмеженими. Враховуючи це, оптимізація ігрового двигуна, рендерингу, скриптової логіки та алгоритмів стає критичною складовою процесу розробки.

Рушій Godot (версії 3.x/4.x) активно розвивається як відкрите рішення для 2D– і 3D-ігор: він пропонує зручну сценно-вузлову архітектуру, підтримку декількох мов програмування (зокрема рідною мовою GDScript та C#) і орієнтований на мультиплатформеність [1].

Однією з сильних сторін Godot є розвинена документація щодо загальних підходів до оптимізації: – наприклад, офіційна стаття «General optimization tips» описує кроки щодо організації проєкту, правильного імпорту ресурсів, налаштування рендерингу, управління світлом і матеріалами [2].

Програмування ігрової логіки на C# у Godot відкриває додаткові можливості: статична типізація, відомі розробникам бібліотеки.NET, кращі засоби під час роботи з більшими проєктами. Зокрема, огляд «Godot C# vs GDScript» звертає увагу на те, що C# «часто перевершує GDScript у виконанні складних обчислень або великих проєктів» – хоча і з певними застереженнями [3].

Отже, тема оптимізації продуктивності при розробці ігор із використанням Godot + C# є надзвичайно актуальною. Вона охоплює кілька вимірів:

– технічний-архітектурний – як правильно структурувати сцени, управляти вузлами (nodes), обирати підходящий рендер-режим;