

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**КРИВОРІЗЬКИЙ ДЕРЖАВНИЙ ПЕДАГОГІЧНИЙ УНІВЕРСИТЕТ**  
**Факультет фізико-математичний**  
**Кафедра інформатики та прикладної математики**

**РОЗРОБКА ПОЗАШКІЛЬНОГО КУРСУ З ВЕБПРОГРАМУВАННЯ**

Кваліфікаційна робота студента групи Ім-24  
ступінь вищої освіти магістр  
спеціальності 014 Середня освіта (Інформатика)  
Степанюка Євгена Валерійовича  
Керівник Мерзликін Павло Володимирович  
кандидат фізико-математичних наук, доцент

## ЗАПЕВНЕННЯ

Я, Степанюк Євген Валерійович, розумію і підтримую політику Криворізького державного педагогічного університету з академічної доброчесності. Запевняю, що ця кваліфікаційна робота виконана самостійно, не містить академічного плагіату, фабрикації, фальсифікації. Я не надавав і не одержував недозволену допомогу під час підготовки цієї роботи. Використання ідей, результатів і текстів інших авторів мають покликання на відповідне джерело.

Із чинним Положенням про запобігання та виявлення академічного плагіату в роботах здобувачів вищої освіти Криворізького державного педагогічного університету ознайомлений. Чітко усвідомлюю, що в разі виявлення у кваліфікаційній роботі порушення академічної доброчесності робота не допускається до захисту або оцінюється незадовільно.

  
(підпис)

## ЗМІСТ

<b>ВСТУП.....</b>	<b>4</b>
<b>РОЗДІЛ 1 РОЛЬ І МІСЦЕ ВЕБПРОГРАМУВАННЯ В СИСТЕМІ ПОЗАШКІЛЬНОГО НАВЧАННЯ.....</b>	<b>7</b>
1.1. Сучасний стан позашкільного навчання інформатики.....	7
1.2. Аналіз популярності напрямку вебпрограмування серед учнів закладів позашкільної освіти Дніпропетровської області.....	10
1.3. Аналіз наявних позашкільних програм з вебпрограмування.....	13
1.4. Постановка задачі.....	15
Висновки до розділу 1.....	16
<b>РОЗДІЛ 2 РОЗРОБКА ПОЗАШКІЛЬНОГО КУРСУ З ВЕБПРОГРАМУВАННЯ.....</b>	<b>18</b>
2.1. Проектування структури позашкільного курсу.....	18
2.2. Обґрунтування вибору інструментів розробки.....	23
2.3. Особливості програмної реалізації.....	27
Висновки до розділу 2.....	34
<b>ВИСНОВКИ.....</b>	<b>36</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....</b>	<b>38</b>

## ВСТУП

Сучасний етап розвитку суспільства характеризується стрімкою цифровізацією всіх сфер людської діяльності. Інформаційно-комунікаційні технології (ІКТ) стають не лише інструментом виробництва, а й середовищем існування сучасної людини. В умовах постійного зростання попиту на ІТ-послуги вебпрограмування залишається одним із найбільш динамічних та затребуваних напрямів індустрії розробки програмного забезпечення. За даними звіту «IT Ukraine Association» за 2023–2024 роки, навіть в умовах воєнного стану ІТ-галузь України демонструє стійкість та залишається однією з ключових експортних галузей економіки [1].

Водночас система освіти стикається з серйозними викликами. Формальна шкільна освіта, регламентована державними стандартами, часто не встигає за швидкістю оновлення вебтехнологій. На уроках інформатики учні часто отримують лише базові уявлення про HTML та CSS, використовуючи застарілі підходи, тоді як сучасний ринок праці вимагає володіння компонентним підходом, JavaScript-фреймворками (React, Vue, Angular), препроцесорами та системами контролю версій.

У цьому контексті особливого значення набуває позашкільна освіта. Заклади позашкільної освіти мають унікальну можливість бути більш гнучкими, швидше адаптувати навчальні програми до потреб ринку та забезпечувати поглиблене вивчення технологій. Однак аналіз наявних навчальних програм гуртків інформаційно-технічного профілю свідчить про дефіцит методичних розробок, які б поєднували доступність для підлітків 12–16 років із вивченням актуального технологічного стеку (Web 3.0, Single Page Applications). Більшість програм або орієнтовані на застарілі CMS-системи, або є надто складними для сприйняття школярами без попередньої підготовки [7].

**Об’єкт дослідження:** процес позашкільного навчання інформатики.

**Предмет дослідження:** розробка позашкільного курсу з вебпрограмування та його програмна реалізація.

**Мета роботи:** розробити сучасний позашкільний курс з вебпрограмування, що відповідає актуальним вимогам ІТ-галузі та освітнім потребам учнів.

Для досягнення мети необхідно розв'язати такі задачі:

- дослідити сучасний стан позашкільної освіти України в галузі інформатики;
- проаналізувати інтерес учнів до напрямку вебпрограмування, зокрема у Дніпропетровській області;
- проаналізувати чинні навчальні програми позашкільної освіти, що стосуються веброзробки;
- визначити основні вимоги до змісту та результатів позашкільного курсу з вебпрограмування;
- розробити структуру та зміст курсу, адаптованого до потреб учнів та сучасних освітніх стандартів;
- обґрунтувати вибір технологій, інструментів та методичних підходів для реалізації курсу;
- описати особливості програмної реалізації курсу та створених матеріалів.

**Практичне значення роботи** полягає у створенні готового навчального курсу для позашкільних закладів, який може бути використаний керівниками гуртків, учителями інформатики та освітніми центрами для формування у школярів базових і поглиблених навичок веброзробки. Розроблений курс може стати основою для подальшого розширення програм у напрямках frontend, backend та фулстек-розробки.

**Структура роботи.** Кваліфікаційна робота складається зі вступу, двох розділів, висновків до кожного розділу, загального висновку та списку використаних джерел.

## **РОЗДІЛ 1 РОЛЬ І МІСЦЕ ВЕБПРОГРАМУВАННЯ В СИСТЕМІ ПОЗАШКІЛЬНОГО НАВЧАННЯ**

### **1.1. Сучасний стан позашкільного навчання інформатики**

Позашкільна освіта в Україні відіграє важливу роль у формуванні в учнів технічних, інтелектуальних і творчих компетентностей, забезпечуючи можливість здобувати знання й навички поза межами формальної шкільної програми. У контексті розвитку цифрової економіки особливого значення набуває позашкільне навчання інформатики, оскільки саме ця галузь дає учням практичні інструменти для створення сучасних технологічних продуктів, включаючи вебдодатки, мобільні сервіси, мультимедійні проєкти та інші ІТ-розробки.

Позашкільна освіта сьогодні переживає період серйозних змін. Якщо раніше гуртки сприймалися переважно як спосіб організації дозвілля дітей, то зараз вони все більше перетворюються на стартовий майданчик для здобуття майбутньої професії. Особливо це стосується технічного напрямку та інформатики, де шкільна програма просто не встигає за швидким розвитком технологій.

#### **Нормативно-правова основа позашкільної інформатичної освіти**

Функціонування позашкільної освіти регулюється Законом України «Про позашкільну освіту» [2], Законом України «Про освіту» [3], державними стандартами та методичними рекомендаціями Міністерства освіти і науки України [4]. Ці документи визначають організаційні принципи, вимоги до програм, освітні форми та очікувані результати навчання.

Закон «Про позашкільну освіту» окреслює її основні завдання: розвиток творчих здібностей дітей, створення умов для їх професійного самовизначення, підтримку обдарованої молоді, формування соціальних і

технічних компетентностей. Інформатика та інформаційні технології віднесені до науково-технічного напрямку, який включає гуртки з програмування, веброзробки, робототехніки, алгоритмізації, 3D-моделювання та інших STEM-компонентів [2].

Переглядаючи рекомендації Міністерства освіти і науки на 2024/2025 навчальний рік, можна помітити чіткий курс на STEM-освіту (наука, технології, інженерія, математика). Держава прямо рекомендує закладам впроваджувати практичні курси, де учні створюють реальні продукти [5]. Вебпрограмування ідеально вписується у цю концепцію, адже тут поєднується написання коду (Technology), логіка роботи інтерфейсів (Engineering) та візуальне оформлення (Art).

### **Особливості організації позашкільного навчання інформатики**

Система позашкільної інформатичної освіти в Україні відзначається гнучкістю та варіативністю. Навчання проводиться у таких формах, як:

- гурткова робота;
- секції та клуби технічної творчості;
- лабораторні та проєктні заняття;
- дистанційні студії;
- тематичні тренінги, майстер-класи, хакатони;
- участь у конкурсах МАН, олімпіадах, технічних фестивалях.

Завдяки відсутності жорсткої прив'язки до шкільної програми педагог може оперативно оновлювати зміст курсу, впроваджувати сучасні технології, залучати учнів до реальних проєктів.

Проте, аналізуючи реальну ситуацію в гуртках інформатики, я виявив низку проблем, які заважають якісному навчанню:

1. **Застарілі програми.** Багато гуртків досі працюють за програмами десятирічної давнини. Учні вчать верстати сайти за допомогою таблиць або використовувати застарілі CMS, які вже нікому не



потрібні на ринку праці. Це демотивує підлітків, які бачать сучасні сайти і розуміють, що їх вчать чомусь іншому [6].

**2. Проблема з обладнанням.** Не всі державні центри творчості мають потужні комп'ютери. Це створює проблему для вивчення «важких» технологій (наприклад, розробки ігор на Unreal Engine або 3D-моделювання). У цьому контексті веброботка є виграшним варіантом: сучасні інструменти, такі як Vite (який ми обрали для курсу), дуже швидкі й дозволяють комфортно працювати навіть на не дуже нових ПК [6].

**3. Кадрове питання.** Знайти викладача, який є практикуючим програмістом і готовий працювати в школі чи гуртку, дуже важко через різницю в зарплатах. Тому існує велика потреба у якісних навчальних курсах, які були б детально розписані й дозволяли педагогу (навіть без великого досвіду в ІТ) дати дітям актуальні знання [7].

Також варто згадати про дистанційне навчання. В умовах війни багато гуртків працюють онлайн. Згідно з планами уряду щодо розвитку освіти, цифровізація є пріоритетом [8]. Вебпрограмування — це, мабуть, найкращий предмет для онлайну: для навчання потрібен лише браузер і редактор коду, а результат роботи (сайт) можна показати вчителю та друзям просто надіславши посилання.

### **Роль вебпрограмування в сучасній позашкільній освіті**

У межах позашкільного навчання вебпрограмування посідає особливе місце, оскільки:

- поєднує технічні та творчі елементи;
- дає можливість отримати швидкий видимий результат (сайт, сторінка, інтерактивний застосунок);

- забезпечує сильну мотивацію учнів через можливість створити щось для реального життя;
- відповідає тенденціям ринку праці, де вебтехнології мають стабільно високий попит;
- дозволяє застосовувати метод проєктного навчання, що підвищує якість засвоєння матеріалу.

Таким чином, сучасний стан позашкільного навчання інформатики демонструє високу актуальність впровадження програм з вебпрограмування, але також виявляє потребу у розробці якісних, сучасних і методично обґрунтованих навчальних курсів, адаптованих до умов позашкільної освіти. Саме цю прогалину покликаний закрити розроблений мною курс.

## **1.2. Аналіз популярності напрямку вебпрограмування серед учнів закладів позашкільної освіти Дніпропетровської області**

Щоб зрозуміти, наскільки мій курс є актуальним саме зараз і саме в нашому регіоні, я провів комплексний аналіз. Він включав дослідження попиту на ІТ-фахівців, активності учнів у наукових конкурсах та змісту реальних програм, за якими зараз навчають дітей у гуртках області.

Дніпропетровщина традиційно є одним із найпотужніших ІТ-хабів України. Щоб підтвердити це фактами, я звернувся до свіжого рейтингу «Топ-50 ІТ-компаній України» (літо 2025 року), опублікованого на порталі DOU [9]. Статистика чітко показує, що, попри всі виклики, великі сервісні та продуктові компанії продовжують наймати фахівців, і значна частина цих вакансій припадає саме на Дніпро або передбачає віддалену роботу з регіону.

Діти та їхні батьки чудово це бачать. Розуміння того, що вебпрограмування — це реальний соціальний ліфт і можливість отримати

високооплачувану роботу, не виїжджаючи з рідного міста, формує шалений попит на відповідні гуртки.

### **Загальні освітні тенденції в регіоні**

Дніпропетровська область є однією з найбільших освітніх екосистем України, що має розвинену мережу центрів науково-технічної творчості, гуртків інформатики та ІТ-студій. У регіоні активно працюють:

- позашкільні заклади (центри дитячої та юнацької творчості, технічні гуртки, МАН);
- муніципальні ІТ-простори (наприклад, молодіжні хаби, цифрові лабораторії);
- приватні освітні заклади (курси з веброзробки, робототехніки, програмування).

Це створює середовище, в якому попит на напрями, пов'язані з ІТ, формується природним шляхом — через доступність технологій, популярність соціальних мереж, інтерес до цифрової творчості та розуміння перспектив ІТ-сфери.

### **Динаміка інтересів учнів (Аналіз МАН)**

Але чи хочуть діти просто «сидіти за комп'ютером», чи вони прагнуть створювати щось серйозне? Щоб з'ясувати це, я проаналізував архіви обласного конкурсу-захисту науково-дослідницьких робіт Малої академії наук (МАН) за тривалий період (починаючи з 2014 року і до сьогодні) [10].

Переглядаючи списки переможців та теми робіт на сайті Дніпропетровського відділення МАН, я помітив цікаву еволюцію:

- У 2014–2016 роках у секціях «Internet-технології» та «WEB-дизайн» переважали прості статичні сайти-візитки або презентаційні проєкти.

- Починаючи з 2019 року і далі, складність робіт різко зросла. Учні почали презентувати власні вебсервіси, спроби створення соціальних мереж, інтернет-магазинів та навчальних платформ.

Це свідчить про те, що рівень амбіцій та технічної грамотності школярів області значно зріс. Їм вже не цікаво просто верстати сторінки — вони хочуть програмувати складну логіку, що вимагає знання JavaScript та сучасних фреймворків.

### **Фактори, що стимулюють популярність вебпрограмування в області**

1. **Потужний локальний ІТ-ринок.** Дніпропетровська область має потужну ІТ-екосистему. Саме в цьому регіоні учні найчастіше обирають веброзробку як напрям для початку знайомства з ІТ [11].
2. **Доступність професійних інструментів.** Використання безкоштовних платформ (Code.org, Replit, GitHub Pages, Vercel) робить веброзробку максимально доступною навіть у громадах з обмеженими ресурсами.
3. **Простий поріг входу.** Початок роботи з HTML/CSS не потребує складних математичних або алгоритмічних знань, що дозволяє швидко побачити результат — власну вебсторінку.

### **Аналітичний висновок**

Популярність вебпрограмування серед учнів позашкільних закладів Дніпропетровщини стабільно зростає завдяки:

- високому попиту на ІТ-професію;
- доступності сучасних навчальних ресурсів;
- розвитку локальної освітньої та ІТ-екосистеми;
- інтересу молоді до створення цифрового контенту та персональних вебпроектів.

Це підтверджує доцільність розробки сучасного позашкільного курсу з вебпрограмування, який відповідатиме вимогам ринку та освітнім потребам учнів регіону.

### **1.3. Аналіз наявних позашкільних програм з вебпрограмування**

У цьому підпункті розглянемо існуючі в Україні позашкільні програми, присвячені вебпрограмуванню, проаналізуємо їхній зміст, цільову аудиторію, переваги та недоліки. Досвід цих програм важливий, щоб на базі цього зробити власний курс, який буде конкурентоспроможним та ефективним.

#### **Державні навчальні програми**

##### **1. Навчальна програма Міністерства освіти та науки України**

«Навчальні програми з позашкільної освіти» від Міносвіти, у якому є модуль, присвячений вебпрограмуванню: PHP, клієнт-серверні додатки, робота з базами даних, CMS (Joomla, uCoz) — саме ці компетенції зазначено як результат навчання [12].

- **Переваги:** формалізований підхід, чіткі компетенції, орієнтація на практичні навички (сервер, БД, CMS).
- **Недоліки:** програма, ймовірно, побудована на старіших технологіях (Joomla, uCoz), може не відповідати найсучаснішим тенденціям (наприклад, фреймворки JS, React, Node.js).

##### **2. Навчальна програма «Сучасні технології програмування» (позашкільна освіта).** Програма, покриває науково-технічний напрямок для гуртків, зокрема модулі програмування [13].

- Фокус на загальні технології програмування, можливість тонко налаштувати курс відповідно до потреб гуртка.
- Може бути гнучкою, але не завжди орієнтована саме на веб (можливо, більше загальна програмістська база).

3. **Навчальна програма «Основи програмування» (позашкільна освіта).** Ще один офіційний документ, який може бути базовим для курсу [14].

- Підходить для початкового рівня — діти, які лише знайомляться з алгоритмами, логікою програмування.
- Менш глибоко охоплює веб-технології, ніж програма Міносвіти, але дає фундамент для подальшого розвитку.

4. **Навчальна програма з позашкільної освіти «Основи веб-технологій».** Авторська програма, що використовується в закладах позашкільної освіти для гуртків інформаційно-технічного профілю. Основний акцент зроблено на вивченні мов розмітки, каскадних таблиць стилів та роботі з графічними редакторами для створення дизайну сайтів [15].

- **Переваги:** комплексний підхід, який поєднує технічну складову (код) із творчою (вебдизайн, робота з графікою); доступність для учнів без попередньої підготовки; орієнтація на створення завершеного продукту (власного сайту).
- **Недоліки:** програма здебільшого орієнтована на створення статичних сайтів і не охоплює створення складних інтерактивних вебдодатків; відсутній глибокий розгляд мови JavaScript та сучасних інструментів розробки (React, TypeScript, Git), які є стандартом у сучасній IT-індустрії.

Наступним етапом дослідження став аналіз пропозицій комерційного сектору. Приватні IT-школи швидше реагують на зміни ринку, тому їхні програми зазвичай більш актуальні. Я проаналізував діяльність п'яти провідних освітніх платформ та шкіл, які пропонують курси для дітей та підлітків: IT Step Academy, IT Talented Kids, IT Club, IT Generation та Academy UA.

Таблиця 1.1

## Порівняльна характеристика приватних курсів з вебпрограмування

Назва закладу / Курсу	Цільова аудиторія / Технології	Переваги	Недоліки
<b>Мала Комп'ютерна Академія (IT Step)</b> [16]	Вік: 9–14 років. Стек: Комплексна програма (все в одному: робототехніка, 3D, вебдизайн, HTML/CSS, Python).	<ul style="list-style-type: none"> <li>Системний підхід та потужна інфраструктура.</li> <li>Гейміфікація процесу навчання (система MyStat).</li> <li>Визнаний бренд та сертифікація.</li> </ul>	<ul style="list-style-type: none"> <li>Висока вартість та тривалість навчання (кілька років).</li> <li>Веброботка є лише одним із модулів, учень не може обрати <i>тільки</i> веб.</li> <li>Програма часто стандартизована і негнучка.</li> </ul>
<b>IT Talented Kids (курс «Web Start»)</b> [17]	Вік: 11–14 років. Стек: HTML5, CSS3, основи JavaScript, публікація сайтів.	<ul style="list-style-type: none"> <li>Чіткий фокус саме на вебтехнологіях.</li> <li>Проектний підхід (створення власного сайту).</li> <li>Робота в міні-групах.</li> </ul>	<ul style="list-style-type: none"> <li>Курс є вступним («Start»), не охоплює поглиблені фреймворки (React) або типізацію (TypeScript).</li> <li>Платна основа. (3000 грн/міс.)</li> </ul>
<b>IT Club</b> [18]	Вік: школярі різного віку. Стек: HTML, CSS, конструктори сайтів, базовий JS.	<ul style="list-style-type: none"> <li>Доступність філій (часто працюють у школах).</li> <li>Практична спрямованість.</li> <li>Демократичніша ціна порівняно з великими академіями.</li> </ul>	<ul style="list-style-type: none"> <li>Часто використовують спрощені інструменти (конструктори) замість чистого коду.</li> <li>Рівень викладання сильно залежить від конкретного локального інструктора.</li> <li>Платна основа. (3500 грн/міс.)</li> </ul>
<b>IT Generation</b> [19]	Вік: підлітки. Стек: Сучасний Frontend (HTML/CSS/JS), Python.	<ul style="list-style-type: none"> <li>Використання сучасних методик викладання.</li> <li>Акцент на Soft Skills та презентації проєктів.</li> <li>Можливість онлайн-навчання.</li> </ul>	<ul style="list-style-type: none"> <li>Відсутність глибокого занурення в архітектуру складних додатків (SPA) у дитячих курсах.</li> <li>Комерційна вартість навчання.</li> </ul>
<b>Academy UA (Online)</b> [20]	Вік: підлітки та студенти. Стек: Web Design, Frontend (адаптивна верстка).	<ul style="list-style-type: none"> <li>Зручний дистанційний формат.</li> <li>Сучасні матеріали, доступ 24/7.</li> <li>Орієнтація на створення портфоліо.</li> </ul>	<ul style="list-style-type: none"> <li>Дистанційний формат підходить не всім дітям (потребує високої самодисципліни).</li> <li>Менше живої взаємодії з викладачем («hands-on» допомоги) порівняно з офлайн-гуртками.</li> </ul>

#### 1.4. Постановка задачі

Аналіз сучасного стану позашкільної інформатичної освіти, тенденцій розвитку вебтехнологій та наявних навчальних програм показав, що попри високий інтерес учнів до вебпрограмування, існує низка суттєвих проблем. Значна частина наявних позашкільних програм застаріла, використовує технології попередніх поколінь або подає матеріал фрагментарно. Багато закладів пропонують базові заняття з HTML і CSS, однак не охоплюють сучасні інструменти веброзробки, не приділяють достатньої уваги практичній проектній діяльності, роботі з JavaScript, інтерактивністю, хмарними сервісами та принципами розробки вебзастосунків.

Окрім цього, не всі програми адаптовані до сучасної освітньої логістики — змішаного формату, дистанційних платформ, використання онлайн-репозиторіїв, інтерактивних середовищ розробки. Це знижує їхню ефективність та привабливість для учнів. Зростаючий попит на IT-компетентності серед молоді Дніпропетровської області, а також активний розвиток локального IT-ринку підсилюють потребу у створенні сучасного, практикоорієнтованого позашкільного курсу з вебпрограмування.

Виходячи з цього, постає необхідність розробити навчальний курс, який:

- відповідає актуальним тенденціям розвитку вебтехнологій;
- орієнтований на учнів 12–16 років, які мають різний стартовий рівень знань;
- поєднує базові знання з HTML/CSS/JavaScript із сучасними підходами до створення вебінтерфейсів;
- забезпечує практичну підготовку через роботу над реальними проектами;



- використовує гнучку модульну структуру, придатну для позашкільної освіти;
- передбачає використання доступних інструментів, хмарних середовищ та сучасних сервісів для розгортання вебпроектів.

Таким чином, **основним завданням дослідження** є розробка сучасного позашкільного курсу з вебпрограмування, який забезпечує учням можливість якісно та послідовно опановувати основи веброзробки, створювати власні вебпроекти, працювати з онлайн-інструментами та формувати практичні компетентності, затребувані в сучасній ІТ-сфері.

### **Висновки до розділу 1**

У першому розділі було проаналізовано сучасний стан позашкільної інформатичної освіти та визначено місце вебпрограмування в системі навчання учнів. Аналіз показав, що позашкільна освіта відіграє важливу роль у формуванні цифрових компетентностей, оскільки характеризується гнучкістю, практичною спрямованістю та можливістю швидко адаптувати зміст навчання до змін у сфері ІТ.

Аналіз нормативної бази засвідчив, що позашкільна освіта має чітко окреслену місію — сприяти розвитку технічної творчості та забезпечувати умови для набуття сучасних знань, зокрема у галузі інформаційних технологій. Упродовж останніх років МОН та ІМЗО активно оновлюють методичні рекомендації, розширюють STEM-напрямок, підтримують конкурси та проєктну діяльність, що створює сприятливе середовище для розвитку вебпрограмування.

Дослідження популярності веброзробки серед учнів Дніпропетровської області виявило стійку позитивну динаміку: підвищується зацікавленість підлітків у створенні вебсайтів та цифрових проєктів, зростає кількість гуртків, інтенсивів та конкурсних робіт,

пов'язаних із вебтехнологіями. Найбільшу активність демонструє вікова група 12–16 років, яка сприймає веброзробку як доступний спосіб реалізувати творчі ідеї, створити власний продукт або почати формувати шлях до майбутньої ІТ-професії.

Аналіз існуючих позашкільних програм засвідчив певні суперечності між сучасними вимогами вебіндустрії та змістом багатьох навчальних курсів. Значна частина офіційних програм використовує застарілі технології або подає матеріал фрагментарно, тоді як приватні ІТ-школи пропонують більш сучасний і практичний зміст, орієнтований на реальні проекти. Це свідчить про потребу в оновленні підходів до навчання веброзробки у позашкільних закладах, розробці сучасних гнучких програм і адаптації методичного матеріалу до актуальних технологій.

Загалом результати розділу вказують на актуальність створення **сучасного позашкільного курсу з вебпрограмування**, який урахує інтереси учнів, орієнтується на реальні потреби ринку та використовує сучасні вебтехнології. Саме ця потреба визначила постановку задачі та стала основою для подальшої розробки авторського курсу, що буде представлено в наступному розділі.

## РОЗДІЛ 2 РОЗРОБКА ПОЗАШКІЛЬНОГО КУРСУ З ВЕБПРОГРАМУВАННЯ

### 2.1. Проєктування структури позашкільного курсу

Проєктування структури навчального курсу є ключовим етапом розробки освітнього продукту, оскільки визначає логіку подання матеріалу, навчальне навантаження, послідовність формування компетентностей та можливості практичного застосування знань учнями. Для позашкільної освіти особливо важливо створити гнучку, модульну та практикоорієнтовану структуру, яка дозволяє адаптувати курс під різний рівень підготовки учнів та забезпечує високий рівень залученості.

#### Методологічні підходи до розробки структури курсу

У процесі проєктування курсу застосовано такі методологічні принципи:

1. **Модульність** — курс поділений на окремі тематичні модулі, які можуть викладатися послідовно або автономно залежно від рівня підготовки групи.
2. **Практична спрямованість** — кожен модуль містить проєктні завдання, а завершення курсу передбачає створення учнями власного вебпроєкту.
3. **Доступність навчального матеріалу** — використання інструментів, які не потребують складного програмного забезпечення (Replit, CodePen, GitHub Pages).
4. **Поступове ускладнення** — від основ HTML і CSS до інтерактивності на JavaScript та роботи з простими API.
5. **Гнучкість формату** — курс розрахований на очну, дистанційну або змішану форму навчання.

6. **Проектне навчання** — ключовим результатом є створення вебсайту або міні-застосунку.

### **Цільова аудиторія та вимоги до структури**

Курс орієнтований на учнів 12–16 років, які можуть мати різний початковий рівень підготовки — від повної відсутності досвіду до базових навичок користування HTML. Відповідно структура курсу повинна:

- забезпечувати швидкий старт та мотивацію;
- включати завдання різної складності;
- пропонувати чітко зрозумілі інструкції та візуальний результат на кожному етапі;
- поступово готувати учнів до самостійної роботи над проектом.

### **Структура курсу**

Загальна тривалість курсу: 36–48 годин, що повністю відповідає типовому семестровому позашкільному курсу [21].

Курс складається із 5 основних модулів, кожен з яких завершується практичним завданням.

### **Модуль 1. Основи веброзробки (теоретичний фундамент)**

**Тривалість:** 6 годин

#### **Цілі модуля:**

- Ознайомити учнів із принципами роботи Інтернету, браузера та вебсторінки.
- Навчити базової структури HTML-документа.
- Сформувати навички роботи з онлайн-середовищами (Replit, CodePen).

#### **Основні теми:**

1. Як працює Інтернет: клієнт–серверна модель у простому поясненні.
2. Що таке HTML, CSS, JS — ролі мов у веброзробці.
3. Структура HTML-документа: теги, атрибути, семантика.

4. Вступ до GitHub: для чого потрібні репозиторії.

**Практичне завдання:** створення першої вебсторінки «Про мене».

### **Модуль 2. HTML та структура вебсторінки**

**Тривалість:** 10 годин

**Цілі модуля:**

- Навчити учнів розмітці контенту.
- Ознайомити з семантичними елементами сучасного HTML5.
- Використовувати таблиці, списки, форми та медіа.

**Основні теми:**

1. Заголовки, параграфи, списки, цитати
2. Зображення, відео, аудіо
3. Посилання, навігація, якірні посилання
4. Таблиці та правильне їх оформлення
5. Основи форм (input, textarea, select)

**Практичне завдання:** створити структуру сайту «Мoje хобі» або «Профіль користувача».

### **Модуль 3. CSS та вебдизайн**

**Тривалість:** 12 годин

**Цілі модуля:**

- Навчити базам стилізації вебсторінок.
- Пояснити принципи адаптивного дизайну.
- Удосконалити навички роботи з макетами.

**Основні теми:**

1. Базові CSS-властивості
2. Flexbox та Grid
3. Кольори, шрифти, іконки
4. Адаптивність: media queries
5. Псевдокласи та анімації

**Практичне завдання:** розробити дизайн повноцінної лендінг-сторінки.

#### **Модуль 4. JavaScript: логіка та інтерактивність**

**Тривалість:** 12 годин

**Цілі модуля:**

- Сформувати розуміння алгоритмів та базових конструкцій JS.
- Навчити працювати з DOM.
- Навчити взаємодії з простими API.

**Основні теми:**

1. Змінні, типи даних, масиви, об'єкти
2. Умови, цикли, функції
3. Робота з DOM (події, маніпуляції елементами)
4. Fetch API: отримання даних з інтернету
5. Створення інтерактивних компонентів

**Практичне завдання:** розробити невеликий інтерактивний вебзастосунок (наприклад, галерею, калькулятор, міні-чат).

#### **Модуль 5. Основи TypeScript для початківців**

**Тривалість:** 6 годин

**Цілі модуля:**

- Ознайомити учнів з ідеєю типізації та її перевагами.
- Показати різницю між JavaScript та TypeScript у простих прикладах.
- Навчити працювати з базовими типами, інтерфейсами та простими функціями.
- Підготувати ґрунт для подальшого вивчення фреймворків (React, Angular), де TypeScript уже став стандартом.

**Чому TypeScript важливий для сучасного курсу?**

- Він *завжди* використовується в комерційній веброзробці (особливо у фронтенді).
- Дозволяє учням усвідомлено писати код, а не інтуїтивно.

- Рано формує “правильні” навички — робота зі структурами даних, інтерфейсами, строгими правилами.
- Добре підходить для командних проєктів, навіть невеликих учнівських.

**Основні теми модуля:**

1. Що таке TypeScript і навіщо він потрібен
2. Типи даних: number, string, boolean, any, unknown
3. Масиви та об'єкти з типами
4. Інтерфейси та типи (interface, type)
5. Типізація функцій
6. Простий tooling: TypeScript Playground, tsc, інтеграція в Replit
7. Перетворення простого JS-проєкту на TS-проєкт

**Практичне завдання:**

- Переписати частину JS-коду попереднього модуля (наприклад, калькулятор або галерею) на TypeScript.
- Додати інтерфейс для структури даних (картка товару, користувач, запис блогу тощо).

**Результат модуля:** Учні розуміють базову типізацію, не бояться TypeScript і можуть читати прості TS-файли. Чудове підґрунтя для подальшої роботи з фреймворками.

**Модуль 6. Створення фінального вебпроєкту**

**Тривалість:** 6–8 годин

**Цілі модуля:**

- Узагальнити отримані знання.
- Навчити працювати над довготривалим проєктом.
- Оформити портфоліо.

**Проектні теми (на вибір):**

- Особистий сайт-портфоліо

- Тематичний односторінковий сайт (мікроблог, каталог, інформаційна сторінка)
- Простий вебдодаток
- Мультимедійний сайт

#### **Фінальні етапи:**

- Завантаження проєкту на GitHub Pages
- Презентація роботи перед групою
- Призначення менторського фідбеку

#### **Додаткові елементи курсу**

- **Міні-олімпіади**, тижневі челенджі
- **Командні проєкти** для старших груп
- **Консультації онлайн** (за потреби та можливостями закладу)
- **Проміжні тести** у форматі Google Forms
- **Використання інтерактивних платформ** (codewars-lite для дітей, CodinGame Kids)

Структура курсу побудована таким чином, щоб забезпечити послідовний перехід від базових понять до створення повноцінних проєктів, сформувати інтерес до веброзробки та надати учням навички, релевантні сучасній ІТ-галузі. Модульність і практична спрямованість дозволяють адаптувати курс до рівня підготовки учнів та умов конкретного позашкільного закладу.

## **2.2. Обґрунтування вибору інструментів розробки**

Створення сучасного позашкільного курсу з вебпрограмування потребує використання інструментів, які поєднують доступність для учнів, актуальність для ІТ-ринку та технологічну гнучкість для викладача. Оскільки курс орієнтований на формування практичних навичок веброзробки, було прийнято рішення розробляти навчальний проєкт на



стеку React + TypeScript + Vite з використанням модульної стилізації на основі SCSS modules та локального JSON-сховища для даних.

Такий набір технологій дозволяє створити навчальний продукт, який:

- відповідає сучасним стандартам фронтенд-розробки;
- є безпечним і легким для розгортання;
- не потребує складної серверної інфраструктури;
- дозволяє учням працювати з тими інструментами, які реально використовуються у комерційних проєктах.

### **Вибір фреймворку React**

React був обраний як основа курсу з таких причин:

1. **Найпопулярніший фронтенд-фреймворк у світі.** React стабільно очолює рейтинги StackOverflow, GitHub та State of JS, а знання React вважається базовою вимогою на ринку праці. Знайомство з ним у позашкільному курсі дає учням шанс працювати зі справжніми технологіями сучасної розробки.
2. **Компонентний підхід.** Компонентна структура інтерфейсу дозволяє учням мислити блоками, а це спрощує розуміння складних вебзастосунків [22].
3. **Низький поріг входу при правильній подачі.** Початкові приклади React можуть бути дуже простими, що робить його доступним навіть для новачків.
4. **Масштабованість.** Фреймворк підходить як для маленьких учнівських проєктів, так і для складних систем.

### **Використання TypeScript**

Вибір TypeScript як основної мови проєкту зумовлений:

1. **Строгою типізацією,** яка допомагає учням зрозуміти структуру даних та уникати типових помилок.

2. **Індустріальним стандартом** — практично всі сучасні React-проекти пишуться з TypeScript.
3. **Покращеною читабельністю та підтримуваністю коду**, що важливо в командних проєктах, які можуть виконувати учні [23].
4. **Інтеграцією з редакторами**, які підсвічують помилки ще до запуску коду — зручна функція для новачків.

### **Вибір інструмента збірки Vite**

Vite став основним інструментом розробки завдяки:

1. **Миттєвому старту проєкту** — він працює набагато швидше за Webpack [24].
2. **Простій конфігурації** — мінімум налаштувань, що важливо в навчальному середовищі.
3. **Вбудованій підтримці TypeScript і React**, що дозволяє учням швидко перейти до практики.
4. **Можливості працювати в будь-якому середовищі**, включно з локальними комп'ютерами позашкільних закладів.

### **SCSS та модульні стилі (styles.module.scss)**

Модульні SCSS-стилі були обрані з таких причин:

1. **Ізоляція стилів** — учні не заплутуються у конфліктах класів.
2. **Підтримка змінних, міксинів, вкладеності**, що формує правильну культуру роботи зі стилями [25].
3. **Сучасний підхід до стилізації**, який широко застосовується у React-проектах.
4. **Простота переходу від базового CSS до SCSS** — учні поступово освоюють складніші інструменти.

### **Використання react-router-dom**

Бібліотека react-router-dom додає можливість [26]:

- створювати багатосторінкові навчальні інтерфейси;

- навчати учнів концепції маршрутизації;
- демонструвати реальні механізми навігації у вебзастосунках.

### **Використання локального JSON як бази даних**

Для зберігання всієї інформації курсу використовується локальний JSON-файл. Це рішення прийнято з педагогічних та технічних причин:

1. **Відсутність потреби у бекенді.** Учні не витрачають час на серверну частину, зосереджуючись на фронтенді.
2. **Спрощення розробки.** JSON легко редагувати, копіювати, переносити, що підходить для навчальних завдань.
3. **Демонстрація роботи з даними.** Це дозволяє навчити учнів працювати з масивами, об'єктами, читанням та відображенням інформації.
4. **Можливість швидко розгорнути проєкт на GitHub Pages.** Оскільки немає бекенду, застосунок можна розгорнути без додаткових сервісів.

### **Вибір ESLint та TypeScript ESLint**

ESLint та пов'язаний із ним набір плагінів обрані для того, щоб:

- навчити учнів писати чистий і структурований код;
- автоматично виявляти помилки стилю та синтаксису;
- забезпечити однаковий стиль написання коду у всіх учнівських роботах;
- сформувати професійні навички з перших етапів навчання.

### **Використання бібліотек @types**

Оскільки проєкт працює з TypeScript, важливим є використання:

- **@types/react**,
- **@types/react-dom**,
- **@types/node**.

Ці пакети забезпечують коректну типізацію всіх компонентів та інтеграцій, що критично важливо для стабільної роботи застосунку в навчальному середовищі.

### **Використання sass-embedded**

Це сучасна високошвидкісна реалізація Sass, яка [25]:

- швидше компілює стилі;
- має кращу сумісність з Vite;
- дозволяє писати структурні та масштабовані стилі.

### **Підсумок**

Вибраний стек React + TypeScript + Vite + SCSS modules + JSON база є оптимальним для позашкільного курсу, оскільки:

- відповідає сучасним вимогам ІТ-галузі;
- забезпечує легкість входу для учнів;
- дозволяє швидко створювати функціональні та стильні вебзастосунки;
- не потребує серверної інфраструктури;
- дозволяє учням працювати зі справжніми інструментами веброзробки;
- забезпечує чисту архітектуру та підтримуваність коду.

## **2.3. Особливості програмної реалізації**

Програмна реалізація навчального вебкурсу базується на сучасному стеку технологій React + TypeScript + Vite, зі стилізацією через SCSS-модулі та зберіганням навчального контенту у локальному JSON-файлі. Такий підхід дозволяє забезпечити гнучку архітектуру, високу продуктивність застосунку, простоту супроводу та відсутність потреби у серверній інфраструктурі, що є важливим для позашкільної освіти.

### **Вибір середовища розробки: Visual Studio Code**

Під час створення авторського вебпроєкту використовувалося середовище Visual Studio Code з таких причин [27]:

1. **Безкоштовність та доступність** — важливо для навчальних закладів.
2. **Потужна екосистема розширень:**
  - Prettier
  - ESLint
  - React/TypeScript snippets
  - Sass support
  - GitHub integration
3. **Зручна робота з терміналом** та підтримка швидкого запуску команд Vite.
4. **Інтегрована система підсвічування помилок TypeScript** — критично для навчального процесу.
5. **Підтримка Git**, що дозволяє учням поступово освоювати систему контролю версій.

VS Code є стандартом де-факто у веброзробці, і його застосування повністю відповідає вимогам до створення сучасного освітнього вебкурсу.

### **Архітектура проєкту**

Застосунок побудований за компонентною структурою React, що дозволяє логічно розділити окремі частини інтерфейсу, спростити підтримку і дає можливість учням зрозуміти принципи модульного проєктування.

Структура проєкту:

```
project/  
| — public/  
| — src/  
|   | — assets/
```

```
| |— components/
| | |— LessonViewer/
| | |— ProgressBar/
| | |— Sidebar/
| | |— Task/
| | |— ui/
| | | |— Button/
| | | |— Scrollable/
| |— constants/
| |— context/
| | |— CourseContext.ts
| |— data/
| | |— course.json
| |— hooks/
| | |— useLocalStorage.ts
| | |— useProgress.ts
| |— pages/
| | |— Course/
| | |— Home/
| | |— Lesson/
| |— types/
| |— utils/
| |— App.css
| |— App.tsx
| |— index.css
| |— main.tsx
|— package.json
|— tsconfig.json
```

| └─ vite.config.ts

### Завантаження даних з JSON

Оскільки бекенд у проєкті не використовується, навчальний контент курсу зберігається в одному файлі: `src/data/course.json`

Структура JSON може включати:

- список модулів;
- теми кожного модуля;
- короткий опис уроків;
- практичні завдання;
- додаткові матеріали;
- посилання на ресурси.

Приклад:

```
{  
  "modules": [  
    {  
      "id": "web-fundamentals",  
      "title": "Модуль 1. Основи веброзробки",  
      "lessons": [  
        {  
          "id": "internet-basics",  
          "title": "Як працює Інтернет",  
          "content": "...",  
          "tasks": ["..."]  
        }  
      ]  
    }  
  ]  
}
```

Завантаження даних у React здійснюється за допомогою:

```
import courseData from "../data/course.json";
```

JSON-файл виконує роль *локальної бази даних*, що дає можливість:

- швидко змінювати контент;
- зберігати структуру курсу окремо від логіки застосунку;
- працювати без API та серверів;
- створювати легку портативну версію проєкту.

### Маршрутизація застосунку

Для маршрутизації використовується react-router-dom, який дозволяє створити багатосторінкову структуру [26]:

- / – головна сторінка
- /course – список уроків
- /course/:moduleId/:lessonId – сторінка конкретного уроку
- \* – сторінка 404

Маршрутна конфігурація:

```
<Routes>
```

```
<Route path="/" element={<Home />} >
```

```
<Route path="/course" element={<Course />} >
```

```
<Route path="/course/:moduleId/:lessonId" element={<Lesson />} >
```

```
<Route path="*" element={<Navigate to="/" />} >
```

```
</Routes>
```

Це дозволяє учням на практиці вивчати:

- принципи навігації у вебзастосунках;
- роботу з параметрами URL;
- структуру SPA.

### Компонентний підхід

Проєкт використовує:

### Компоненти сторінок (pages/)



Відповідають за логіку відображення окремих частин застосунку.

### **Компоненти інтерфейсу (components/ui/)**

Кнопки, інпути, модальні вікна, картки — елементи, які можна перевикористовувати.

### **Компоненти оформлення (components/)**

- Layout
- LessonViewer
- ProgressBar
- Sidebar
- Task

### **Хуки (hooks/)**

Наприклад:

- `useLocalStorage()` — робота з `localStorage` браузера
- `useProgress()` — розраховує і повертає інформацію про прогрес проходження курсу

Хуки допомагають учням зрозуміти сучасний підхід до логіки в React.

### **Стилізація через `styles.module.scss`**

Використання SCSS-модулів дозволяє [25]:

- ізолювати стилі компонентів;
- використовувати змінні, міксини та вкладеність;
- уникати конфліктів класів;
- структурувати стилі за компонентами.

Приклад:

```
import styles from "./styles.module.scss";  
export const Button = () => (  
  <button className={styles.button}>  
    {children}  
  </button>
```

);

### **Інструменти контролю якості коду**

У проєкті застосовано:

- **ESLint** — перевірка стилю коду [28];
- **typescript-eslint** — інтеграція ESLint з TS [29];
- **eslint-plugin-react-hooks** — контроль правильного використання React-хуків [30];
- **Prettier (опційно)** — автоформатування коду.

Це дозволяє учням працювати у середовищі з чіткими правилами написання коду та уникати поширених помилок.

### **Збірка та запуск застосунку**

Завдяки Vite розробка працює максимально швидко:

- запуск проєкту: `npm run dev`
- збірка для продакшену: `npm run build`

Швидкість HMR дозволяє працювати комфортно навіть на слабких комп'ютерах навчальних закладів.

### **Розгортання проєкту**

Проєкт легко розгортається на [31]:

- **GitHub Pages**
- **Netlify**
- **Vercel**

Оскільки backend не використовується, розгортання займає 1–2 хвилини.

### **Підсумок**

Програмна реалізація вебкурсу побудована таким чином, щоб забезпечити максимальну практичність, простоту та відповідність сучасним стандартам веброботи. Використання React, TypeScript, Vite, SCSS-модулів та JSON-бази дозволяє створити повністю автономний

проект, який легко розгортати, підтримувати і розширювати. Таке технічне рішення відповідає потребам позашкільної освіти та створює комфортне середовище для навчання вебпрограмуванню.

## **Висновки до розділу 2**

У другому розділі здійснено комплексну розробку авторського позашкільного курсу з вебпрограмування, що враховує актуальні тенденції ІТ-галузі, вікові особливості учнів та специфіку позашкільної освіти. На основі проведеного аналізу було спроектовано сучасну, модульну та практикоорієнтовану структуру курсу, яка забезпечує послідовне формування знань — від основ веброзробки до створення повноцінного вебзастосування.

Побудована структура передбачає шість навчальних модулів: основи веброзробки, HTML, CSS, JavaScript, TypeScript та створення фінального проєкту. Така модель дозволяє здійснювати поступове ускладнення матеріалу, а також забезпечує можливість адаптації темпів навчання залежно від рівня підготовки групи. Особливу увагу приділено практичним завданням, проєктному навчанню та роботі з сучасними інструментами розробки, що сприяє формуванню у здобувачів стійких практичних компетентностей.

Обґрунтовано вибір технологічного стеку React + TypeScript + Vite, який є стандартом фронтенд-розробки у сучасних ІТ-компаніях. Використання SCSS-модулів, ESLint, бібліотек React Router, а також зберігання контенту у JSON-файлі забезпечує оптимальний баланс між простотою впровадження, функціональністю та відповідністю реальним умовам веброзробки. Такий набір інструментів дозволяє створити продуктивний, легкий у підтримці та розширюванні навчальний проєкт.

У межах програмної реалізації було сформовано архітектуру застосунку, розроблено логіку роботи з даними, структуру компонентів, маршрутизацію та сервіси, що забезпечують повноцінну роботу навчального вебкурсу без використання бекенда. Використання середовища Visual Studio Code сприяло ефективній організації розробки та відповідає сучасним стандартам професійної практики.

Підсумовуючи, розділ демонструє, що розроблений авторський курс відповідає вимогам сучасної вебіндустрії, є методично доцільним та технічно реалізованим, а також здатен забезпечити учням якісну підготовку у сфері вебпрограмування в умовах позашкільної освіти.

## ВИСНОВКИ

У кваліфікаційній роботі було здійснено комплексне дослідження стану позашкільної освіти у сфері вебпрограмування, проведено аналіз сучасних освітніх трендів, розглянуто наявні навчальні програми та визначено їхні сильні й слабкі сторони. На основі виявлених проблем і потреб учнівської аудиторії розроблено авторський позашкільний курс, що відповідає вимогам сучасної ІТ-галузі та освітніх стандартів.

У першому розділі було визначено роль вебпрограмування в системі позашкільної освіти та обґрунтовано важливість його впровадження як інструмента формування цифрових компетентностей. Аналіз популярності напряму серед учнів Дніпропетровської області засвідчив зростання інтересу молоді до веброзробки, що потребує оновлення методичної бази та створення сучасних педагогічних продуктів. Вивчення існуючих програм показало, що значна частина з них не відображає актуальних тенденцій розвитку вебтехнологій, що зумовило необхідність розробки нового, більш сучасного курсу.

Другий розділ присвячено проектуванню структури курсу та його технічній реалізації. Було створено модульну систему, яка охоплює ключові етапи навчання: основи веброзробки, HTML, CSS, JavaScript, TypeScript та фінальний проєкт. Така структура забезпечує послідовність засвоєння знань, формує в учнів практичні навички та стимулює творче мислення. Обґрунтовано вибір технологічного стеку React + TypeScript + Vite, який відповідає стандартам сучасної фронтенд-розробки та дозволяє створити швидкий, легкий у розгортанні та підтримці навчальний застосунок. Особливу увагу приділено використанню SCSS-модулів, ESLint та локального JSON-файлу як джерела навчального контенту, що забезпечує автономність та доступність проєкту для позашкільних закладів.

У процесі створення програмної реалізації було застосовано професійні підходи до структурування коду, організації компонентів та роботи з даними. Проєкт розроблено в середовищі Visual Studio Code, що є загальноновизнаним інструментом веброзробників і дозволяє учням працювати в реальних умовах сучасної ІТ-практики.

Аналіз та реалізована розробка підтвердили, що створення сучасного позашкільного курсу з вебпрограмування є не лише актуальним, але й необхідним для забезпечення конкурентних цифрових навичок молоді. Авторський курс може бути використаний у позашкільних навчальних закладах, адаптований під різні вікові групи та доповнений новими модулями залежно від потреб учнів.

Таким чином, поставлені у роботі завдання виконано повністю. Результатом дослідження є цілісний, методично обґрунтований і технічно реалізований освітній продукт, який сприяє розвитку цифрової грамотності, творчих здібностей та інтересу учнів до сучасних інформаційних технологій.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1) Digital Tiger: the power of Ukrainian IT – 2023. *IT Ukraine Association*. URL: <https://itukraine.org.ua/digital-tiger-the-power-of-ukrainian-it-2023/> (дата звернення: 03.11.2025).
- 2) Про позашкільну освіту: Закон України від 22.06.2000 №1841-III. Відомості Верховної Ради України, 2000, №46. URL: <https://zakon.rada.gov.ua/laws/show/1841-14> (дата звернення: 03.11.2025).
- 3) Про освіту: Закон України від 22.06.2017 №2657-VIII. Відомості Верховної Ради України, 2017, №38-39. URL: <https://zakon.rada.gov.ua/laws/show/2145-19> (дата звернення: 03.11.2025).
- 4) Методичні матеріали за напрямками позашкільної освіти Міністерства освіти і науки України. URL: <https://mon.gov.ua/tag/metodichni-rekomendatsii?&tag=metodichni-rekomendatsii> (дата звернення: 03.11.2025).
- 5) Методичні рекомендації щодо розвитку STEM-освіти в закладах загальної середньої та позашкільної освіти у 2024/2025 навчальному році : Лист ІМЗО від 12.08.2024 № 21/08-1242. *Інститут модернізації змісту освіти*. URL: <https://imzo.gov.ua/2024/08/13/lyst-imzo-vid-12-08-2024-21-08-1242-metodychni-rekomendatsii-shchodo-rozvytku-stem-osvity-v-zakladakh-zahal-noi-seredn-oi-ta-pozashkil-noi-osvity-u-2024-2025-navchal-nomu-rotsi> // (дата звернення: 03.11.2025).
- 6) Використання інформаційних технологій у практиці роботи початкової школи. URL: [https://library.udpu.edu.ua/library\\_files/430895.pdf](https://library.udpu.edu.ua/library_files/430895.pdf) (дата звернення: 03.11.2025).
- 7) Збірник матеріалів VI Всеукраїнської науково-практичної конференції молодих учених. URL: <https://dspace.udpu.edu.ua/bitstream/123456789/10638/2/V%D0%86%20%D0%92%D0%A1%D0%95%D0%A3%D0%9A%D0%A0%D0%90%D0%87%D0%9D%D0%A1%D0%AC%D0%9A%D0%90%20%D0%9D%D0%90%D0%A3%D0%9A%D0%9E%D0%92%D0%9E-%D0%9F%D0%A0%D0%90%D0%9A%D0%A2%D0%98%D0%A7%D0%9D%D0>

[%90%20%D0%9A%D0%9E%D0%9D%D0%A4%D0%95%D0%A0%D0%95%D0%9D%D0%A6%D0%86%D0%AF%20%D0%9C%D0%9E%D0%9B%D0%9E%D0%94%D0%98%D0%A5%20%D0%92%D0%A7%D0%95%D0%9D%D0%98%D0%A5%20%C2%AB%D0%9D%D0%90%D0%A3%D0%9A%D0%9E%D0%92%D0%90%20%D0%9C%D0%9E%D0%9B%D0%9E%D0%94%D0%AC-2018%2016.11.pdf#page=13&8](#) (дата звернення: 03.11.2025).

- 8) Про схвалення Концепції розвитку природничо-математичної освіти (STEM-освіти) : Розпорядження Кабінету Міністрів України від 05.08.2020 № 960-р. *Верховна Рада України*. URL: <https://zakon.rada.gov.ua/laws/show/960-2020-p#Text> (дата звернення: 03.11.2025).
- 9) Топ-50 ІТ-компаній України, літо 2025: нові лідери та тенденції ринку. *DOU.ua*. URL: <https://dou.ua/lenta/articles/top-50-summer-2025/> (дата звернення: 06.11.2025).
- 10) Архіви конкурсів-захистів науково-дослідницьких робіт учнів-членів МАН України. *Дніпропетровське відділення Малої академії наук України*. URL: <https://dvman.dnepredu.com/uk/site/konkurs-zakhist.html> (дата звернення: 06.11.2025).
- 11) Дослідження «Dnipro IT Ecosystem Research 2025». URL: <https://www.dnipro.media/yak-projshla-it-dnipro-conference/> (дата звернення: 06.11.2025).
- 12) Навчальна програма з позашкільної освіти науково-технічного напрямку «Основи вебтехнологій». *Мала академія наук України*. URL: <https://api.man.gov.ua/api/assets/man/ab5bc975-c01c-409a-a689-67f2521c7aaf/> (дата звернення: 11.11.2025).
- 13) Навчальна програма з позашкільної освіти «Сучасні технології програмування». URL: [https://zpo.ucoz.ua/Programu\\_/Nayk\\_tehnik/suchasni\\_tekhnologiji\\_programuvannya.pdf](https://zpo.ucoz.ua/Programu_/Nayk_tehnik/suchasni_tekhnologiji_programuvannya.pdf) (дата звернення: 11.11.2025).



- 14) Навчальна програма з позашкільної освіти «Основи програмування». URL:  
[https://zpo.ucoz.ua/Programu\\_/Nayk\\_tehnik/osnovi\\_programuvannja.pdf](https://zpo.ucoz.ua/Programu_/Nayk_tehnik/osnovi_programuvannja.pdf)  
(дата звернення: 11.11.2025).
- 15) Навчальна програма з позашкільної освіти «Основи веб-технологій». URL:  
[https://drive.google.com/file/d/11Ka\\_v7fkJj3RHgZbMH0nJIVPYqC48XQz/view](https://drive.google.com/file/d/11Ka_v7fkJj3RHgZbMH0nJIVPYqC48XQz/view) (дата звернення: 11.11.2025).
- 16) Навчання для дітей. *IT Step Academy*. URL:  
<https://cloud.itstep.org/education-children> (дата звернення: 12.11.2025).
- 17) Курс Web Start. *IT Talented Kids*. URL:  
<https://ittalentedkids.com.ua/web> (дата звернення: 12.11.2025).
- 18) Навчальні курси. *IT Club*. URL: <http://www.itclub.in.ua> (дата звернення: 12.11.2025).
- 19) ІТ-освіта для дітей та підлітків. *IT Generation*. URL:  
<https://it-generation.com.ua> (дата звернення: 12.11.2025).
- 20) Онлайн курси для дітей. *Academy UA*. URL:  
<https://online.academyua.com> (дата звернення: 12.11.2025).
- 21) Web-course. URL: <https://github.com/eugenestepaniuk/web-course>
- 22) React Documentation — Official React 19 Guidelines. URL:  
<https://react.dev> (дата звернення 18.11.2024).
- 23) TypeScript Documentation — Microsoft. URL:  
<https://www.typescriptlang.org> (дата звернення 20.11.2024).
- 24) Vite — Official Documentation. URL: <https://vitejs.dev> (дата звернення 18.11.2024).
- 25) SCSS (Sass) Official Documentation. URL:  
<https://sass-lang.com/documentation> (дата звернення 18.11.2024).
- 26) React Router Documentation — Official Guide. URL:  
<https://reactrouter.com> (дата звернення 18.11.2024).
- 27) Visual Studio Code — Official Documentation. URL:  
<https://code.visualstudio.com/docs> (дата звернення 18.11.2024).

- 28) ESLint — Official Documentation. URL: <https://eslint.org/docs/latest/> (дата звернення 18.11.2024).
- 29) Typescript-eslint — Official Documentation. URL: <https://typescript-eslint.io/getting-started/> (дата звернення 18.11.2024).
- 30) React Router Documentation — eslint-plugin-react-hooks. URL: <https://react.dev/reference/eslint-plugin-react-hooks> (дата звернення 18.11.2024).
- 31) GitHub Education — Student Program Documentation. URL: <https://education.github.com> (дата звернення 18.11.2024).