

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ВОЛИНСЬКІЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ЛЕСІ УКРАЇНКИ**

**Кафедра загальної математики та методики навчання інформатики**

На правах рукопису

**ФЕДОРЕНКО МАКСИМ ПЕТРОВИЧ  
МЕТОДИКА ВИВЧЕННЯ МОДУЛЯ “ВЕБ ТЕХНОЛОГІЇ”  
У ЗАКЛАДАХ ЗАГАЛЬНОЇ СЕРЕДНЬОЇ ОСВІТИ**

Спеціальність: 014 «Середня освіта (Інформатика)»

Освітньо-професійна програма Середня освіта. Інформатика

Робота на здобуття освітнього ступеня «Магістр»

Науковий керівник:

**Собчук Оксана Миколаївна**

кандидат педагогічних наук,

доцент кафедри загальної математики

та методики навчання інформатики

**РЕКОМЕНДОВАНО ДО ЗАХИСТУ**

Протокол № \_\_\_\_\_

засідання кафедри загальної математики

та методи навчання інформатики від

\_\_\_\_\_ 2024 року

Завідувач кафедри:

\_\_\_\_\_

**ЛУЦЬК – 2024**

## Анотація

**Федоренко М.П. Методика вивчення модуля “Веб технології” у закладах загальної середньої освіти – Рукопис**

Кваліфікаційна робота за спеціальністю 014 Середня освіта (Інформатика). – Волинський національний університет імені Лесі Українки. Луцьк, 2024.

Робота присвячена дослідженню та вдосконаленню методики викладання вибіркового модуля «Веб-технології» у старшій школі. У процесі дослідження проаналізовано зміст навчальної програми з інформатики для 10–11 класів, визначено місце модуля «Веб-технології» у формуванні сучасних цифрових компетентностей учнів. Особливу увагу приділено теоретичним засадам викладання веб-технологій, що охоплюють підходи, принципи навчання та ключові компетентності, необхідні для успішного опанування матеріалу. Детально розглянуто навчальний зміст модуля – HTML, CSS, JavaScript та основи адаптивного дизайну. Запропоновано методичні рекомендації щодо застосування проєктних і кейс-методів, індивідуальних та групових завдань, а також наведено приклади уроків, практичних завдань і проєктів. Результати апробації представлених методичних підходів свідчать про підвищення мотивації учнів, формування їх критичного мислення, творчих навичок та поглиблення розуміння веб-технологій, що загалом сприяє підвищенню якості навчання.

**Ключові слова:** методика навчання, веб-технології, HTML, CSS, JavaScript, адаптивний дизайн, цифрова грамотність, проєктний метод, кейс-методи, інтерактивне навчання, старша школа, компетентнісний підхід, інформатика, модульне навчання.

## **Annotation**

### **Fedorenko M. P. Methodology of Teaching the "Web Technologies" Module in General Secondary Education Institutions – Manuscript**

Graduate Qualification Work in Specialty 014 Secondary Education (Informatics). – Lesya Ukrainka Volyn National University. Lutsk, 2024.

This diploma thesis is dedicated to researching and improving the methodology of teaching the elective "Web Technologies" module in senior high school. The study analyzes the content of the 10th–11th grade informatics curriculum, identifying the module's role in fostering students' modern digital competencies. Special attention is given to the theoretical foundations of teaching web technologies, including pedagogical approaches, principles, and key competencies required for successful mastery of the subject. The content of the module—HTML, CSS, JavaScript, and fundamentals of responsive design—is examined in detail. The thesis provides methodological recommendations for implementing project-based and case methods, as well as individual and group assignments. Examples of lessons, practical tasks, and projects are also offered. The results of testing these methodological approaches indicate an increase in student motivation, the development of critical thinking and creativity, and a deeper understanding of web technologies, ultimately enhancing the overall quality of learning.

**Keywords:** teaching methodology, web technologies, HTML, CSS, JavaScript, responsive design, digital literacy, project-based method, case methods, interactive learning, high school, competency-based approach, informatics, modular learning

## ЗМІСТ

<b>ВСТУП.....</b>	<b>4</b>
<b>РОЗДІЛ I. ТЕОРЕТИЧНІ ОСНОВИ ВИВЧЕННЯ ВИБІРКОВОГО МОДУЛЯ «ВЕБ-ТЕХНОЛОГІЇ».....</b>	<b>7</b>
1.1. Поняття веб-технологій і їх роль у сучасному світі.....	7
1.2. Аналіз навчальної програми інформатики для 10–11 класів: місце модуля «Веб-технології».....	10
1.3. Значення модуля «Веб-технології» у формуванні цифрової грамотності старшокласників.....	15
1.4. Теоретичні основи викладання веб-технологій: підходи, принципи, ключові компетентності.....	18
1.5. Основний зміст і тематика модуля «Веб-технології».....	22
<b>РОЗДІЛ II. ПРАКТИЧНА МЕТОДИКА ВИКЛАДАННЯ ВИБІРКОВОГО МОДУЛЯ «ВЕБ-ТЕХНОЛОГІЇ».....</b>	<b>25</b>
2.1. Постановка задачі дослідження: основні завдання методичної системи навчання.....	25
2.2. Розробка змістового наповнення модуля.....	27
2.3. Форми та методи викладання модуля.....	36
2.3.1. Переваги використання проектної методики під час вивчення модуля “Веб-технології”.....	37
2.3.2. Кейс-методи у навчанні веб програмуванню.....	39
2.3.3. Інтеграція проектної методики та кейс-методів у навчальний процес.....	41
Очікувані результати впровадження проектної методики та кейс-методів.....	44
2.4. Планування, організація освітнього процесу, приклади завдань і проектів.....	45
2.5. Апробація методики викладання модуля «Веб-технології» та результати дослідження.....	53
<b>ВИСНОВКИ.....</b>	<b>60</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....</b>	<b>65</b>
<b>ДОДАТОК 1.....</b>	<b>69</b>
<b>ДОДАТОК 2.....</b>	<b>73</b>

## ВСТУП

У сучасному світі цифровізація охоплює всі сфери життя, починаючи від освіти і закінчуючи побутом та професійною діяльністю. Ці зміни вимагають від молодого покоління не лише базових навичок користування цифровими технологіями, але й вміння працювати з інформаційними системами, розуміти принципи їх функціонування та створювати власні цифрові продукти. У цьому контексті важливим компонентом шкільної освіти є навчання веб-технологій – основного інструменту для створення веб-ресурсів, які забезпечують комунікацію, доступ до інформації та вирішення багатьох завдань сучасного суспільства.

Вибірковий модуль «Веб-технології», який входить до навчальної програми з інформатики для 10–11 класів, забезпечує учнів старшої школи знаннями про принципи розробки веб-ресурсів, їх оформлення та публікацію в Інтернеті. Навчання цього модуля дозволяє не лише поглибити знання з інформатики, але й розвинути творчі навички, вміння працювати в команді, вирішувати складні проблеми та презентувати результати своєї діяльності.

Актуальність дослідження полягає у необхідності розробки ефективної методики викладання модуля «Веб-технології», яка враховуватиме вікові особливості учнів, рівень їхньої підготовки та сучасні тенденції розвитку інформаційних технологій. Систематизація теоретичних основ і створення практичної методики навчання сприятиме не лише кращому засвоєнню знань, але й підготовці школярів до життя в умовах цифрового суспільства.

**Метою** кваліфікаційної роботи є розробка ефективної методики навчання вибіркового модуля «Веб-технології» для учнів 10–11 класів, яка включатиме оптимальне поєднання теоретичних і практичних занять, форм та методів організації навчального процесу, а також змістового наповнення курсу.

Для досягнення цієї мети поставлено такі **завдання**:

1. Провести аналіз навчальної програми з інформатики для старшої школи та визначити місце модуля «Веб-технології».

2. Обґрунтувати значення вивчення веб-технологій у розвитку цифрової грамотності учнів.
3. Дослідити теоретичні основи викладання веб-технологій.
4. Розробити структуру, зміст і методику навчання модуля «Веб-технології».
5. Провести апробацію розробленої методики в умовах навчального процесу закладу загальної середньої освіти.
6. Сформулювати рекомендації для вчителів щодо ефективного викладання модуля «Веб-технології».

**Об'єктом** дослідження є навчальний процес у рамках вибіркового модуля «Веб-технології» для 10–11 класів.

**Предметом** дослідження є методика викладання цього модуля з акцентом на інтерактивні, проектні та практично орієнтовані форми навчання.

**Наукова новизна** роботи полягає у створенні інтегрованої методики навчання веб-технологій, яка враховує сучасні педагогічні підходи, тенденції розвитку інформаційних технологій і потреби учнів старшої школи.

**Практичне значення** дослідження полягає у можливості застосування розроблених матеріалів для викладання модуля «Веб-технології» у загальноосвітніх школах, ліцеях і коледжах. Результати цього дослідження можуть бути використані для вдосконалення методики викладання вибіркового модуля «Веб-технології» у старших класах, а також для подальших наукових розробок у галузі методики навчання інформатики.

Кваліфікаційна робота складається з двох розділів. У першому розділі розглянуто теоретичні основи викладання вибіркового модуля «Веб-технології»: місце цього модуля в навчальній програмі, значення його вивчення, теоретичні підходи та основний зміст курсу. Другий розділ присвячений розробці практичної методики навчання модуля, включаючи структуру курсу, організацію навчальної діяльності, опис фрагментів уроків і результати апробації.

Результати кваліфікаційної роботи пройшли апробацію під час конференції - VIII Міжнародна науково-практична конференція «Актуальні проблеми

розвитку природничих та гуманітарних наук” Луцьк, Волинський національний університет імені. Лесі Українки, 14 листопада 2024 р.

## РОЗДІЛ I. ТЕОРЕТИЧНІ ОСНОВИ ВИВЧЕННЯ ВИБІРКОВОГО МОДУЛЯ «ВЕБ-ТЕХНОЛОГІЇ»

### 1.1. Поняття веб-технологій і їх роль у сучасному світі

Веб-технології становлять собою комплекс засобів, стандартів і технологій, які дозволяють створювати та забезпечувати функціонування веб-ресурсів, таких як веб-сайти, портали, веб-додатки, онлайн-сервіси та інші інтерактивні платформи, що працюють у середовищі Інтернету. Веб-технології з'явилися як інструмент для передачі інформації в глобальній мережі, однак швидкий розвиток інтернет-інфраструктури, мобільних пристроїв та цифрових технологій зробили їх однією з основних складових сучасного цифрового суспільства. Важливість веб-технологій у сучасному світі обумовлена їх універсальністю, доступністю та можливістю взаємодії між користувачами, пристроями та програмними системами на різних платформах і в різних середовищах.

Основу веб-технологій складають мови розмітки та програмування, які визначають структуру, оформлення і функціональність веб-сторінок. HTML (HyperText Markup Language) є основною мовою розмітки для структурування контенту на веб-сторінках. HTML визначає елементи, такі як заголовки, абзаци, зображення, посилання та форми, які створюють основу будь-якої веб-сторінки. Без HTML неможливо створити повноцінний веб-ресурс, оскільки він є базовою мовою для будь-якого інтернет-сайту. Окрім HTML, мова CSS (Cascading Style Sheets) використовується для оформлення веб-сторінок, що дозволяє задавати шрифти, кольори, відступи, а також інші візуальні аспекти сторінки. Таким чином, CSS додає до веб-сайту естетичний вигляд та забезпечує зручність для користувачів [20].

JavaScript є мовою програмування, яка дозволяє додати інтерактивність до веб-ресурсів, роблячи їх динамічними та адаптивними до дій користувача. Наприклад, JavaScript може обробляти кліки миші, вводити інформацію в форми, працювати з анімаціями та виконувати складні логічні операції, що



робить його основним інструментом для створення сучасних веб-додатків. У поєднанні з бібліотеками та фреймворками, такими як jQuery, React, Vue.js або Angular, JavaScript дозволяє створювати складні та інтерактивні веб-додатки, які є зручними, швидкими і багатофункціональними.

Серверні технології, такі як PHP, Python, Ruby та Node.js, відповідають за обробку інформації на стороні сервера та забезпечують інтеграцію з базами даних, що є необхідним для роботи складних веб-додатків. Завдяки серверним технологіям веб-додатки можуть зберігати дані, виконувати авторизацію користувачів, обробляти форми, генерувати сторінки на основі запитів користувачів, а також здійснювати комунікацію між сервером та клієнтом. Бази даних, такі як MySQL, PostgreSQL та MongoDB, є важливою складовою веб-додатків, оскільки вони забезпечують збереження та обробку великої кількості інформації.

Роль веб-технологій у сучасному світі не обмежується лише інструментами для створення сайтів. Вони є критично важливими для функціонування численних галузей економіки, соціальної сфери, освіти, медицини, торгівлі, розваг та інших аспектів повсякденного життя. У сучасному суспільстві веб-технології стали основою для розвитку таких сфер, як електронна комерція, онлайн-освіта, соціальні мережі, державні сервіси та дистанційна робота. Наприклад, інтернет-магазини та онлайн-платформи надають змогу продавати та купувати товари без необхідності фізичної присутності, що особливо важливо в умовах глобалізації та зростання попиту на дистанційні послуги. Веб-технології відкривають нові можливості для підприємців і стартапів, які можуть досягти глобальної аудиторії за допомогою цифрових платформ.

У галузі освіти веб-технології дозволяють організувати дистанційне навчання, надаючи учням і студентам доступ до навчальних матеріалів з будь-якого місця та в будь-який час. Веб-платформи для навчання, такі як Moodle, Google Classroom та Coursera, забезпечують інтерактивний підхід до навчання та дозволяють педагогам ефективніше взаємодіяти з учнями, а учням

— отримувати знання у зручній формі. Зокрема, під час пандемії COVID-19 веб-технології продемонстрували свою важливість, надавши можливість школам та університетам переходити на дистанційне навчання та забезпечувати безперервний процес освіти.

Медицина також активно використовує веб-технології для організації електронних медичних карток, онлайн-консультацій, дистанційного моніторингу здоров'я пацієнтів та інтеграції систем для обміну інформацією між лікарями та медичними установами. Цифрові медичні сервіси, побудовані на основі веб-технологій, дозволяють пацієнтам отримувати медичну допомогу незалежно від географічного розташування, що значно підвищує доступність медичних послуг.

Ще однією з найбільших галузей, що використовує веб-технології, є сфера фінансів і банківської справи. Інтернет-банкінг, мобільні додатки, платіжні сервіси – все це є прикладами використання веб-технологій для полегшення фінансових операцій та підвищення їх доступності. Онлайн-сервіси надають змогу користувачам виконувати банківські операції, управляти рахунками, оплачувати комунальні послуги, здійснювати перекази та інші фінансові операції без потреби відвідувати відділення банку.

З огляду на актуальність веб-технологій для багатьох сфер життя, попит на фахівців у галузі веб-розробки стрімко зростає. Ринок праці потребує спеціалістів, які володіють не лише базовими навичками, а й здатні працювати з сучасними інструментами та технологіями для створення високоякісних і безпечних веб-додатків. Володіння веб-технологіями відкриває перед молодими спеціалістами численні кар'єрні можливості, такі як веб-розробник, дизайнер, тестувальник, проєктний менеджер та аналітик. Оскільки компанії постійно шукають нові шляхи розвитку своїх онлайн-платформ та цифрових сервісів, знання веб-технологій стає важливою конкурентною перевагою для молодих фахівців на ринку праці.

Включення вивчення веб-технологій до освітніх програм профільної школи є важливим кроком до підготовки учнів до сучасного ринку праці.

Оволодіння веб-технологіями допомагає учням розвивати критичне мислення, креативність, навички вирішення задач, а також здатність до самостійного навчання, що є важливими компетенціями для будь-якої професії у цифрову епоху. Навчання веб-технологій дає учням можливість створювати власні проекти, працювати з реальними інструментами та програмами, що дозволяє їм підготуватися до подальшого навчання або роботи в ІТ-сфері.

## **1.2 Аналіз навчальної програми інформатики для 10–11 класів: місце модуля «Веб-технології»**

Навчальна програма інформатики для 10–11 класів є основою формування інформаційної культури учнів, забезпечуючи їх знаннями та навичками, які відповідають вимогам сучасного суспільства. Одним із ключових елементів програми є модуль «Веб-технології», який займає важливе місце у структурі профільного навчання та реалізується як на рівні стандарту, так і на профільному рівні. Його включення до навчального плану обумовлено зростанням значення інформаційно-комунікаційних технологій у сучасному світі та необхідністю підготовки учнів до вирішення прикладних завдань, пов'язаних із використанням веб-технологій у різних сферах діяльності.

Згідно з програмою інформатики для старшої школи, метою навчання є формування компетентностей, які дозволяють учням ефективно використовувати сучасні інформаційно-комунікаційні технології для вирішення навчальних, професійних і повсякденних завдань. Модуль «Веб-технології» спрямований на розвиток у школярів здатності створювати, адмініструвати й оптимізувати веб-ресурси, розвиваючи їхні навички у сфері розробки та дизайну веб-сайтів. Цей модуль є логічним завершенням курсу інформатики, адже він синтезує знання, отримані під час вивчення базових та інших варіативних модулів, зокрема програмування, роботи з даними, графіки та мультимедіа.

Модуль «Веб-технології» забезпечує учням комплексні знання, які охоплюють основи створення веб-сторінок, використання мов розмітки (HTML,

CSS), принципи адаптивного дизайну та інтеграцію мультимедійного контенту. Крім того, учні вивчають основи пошукової оптимізації (SEO), що є критично важливим у сучасних умовах цифрової економіки. Особливе місце у програмі займає робота з системами керування вмістом (CMS), такими як WordPress, Joomla або Wix, які дозволяють створювати функціональні веб-сайти навіть за відсутності глибоких знань програмування. Це відповідає сучасним тенденціям у сфері веб-розробки, де велика частина проектів базується на використанні готових платформ і шаблонів.

Аналізуючи структуру програми, варто зазначити, що модуль «Веб-технології» розміщений у другій половині курсу інформатики, переважно в 11 класі, що дозволяє учням застосовувати вже здобуті знання для виконання практичних завдань. Зокрема, у програмі для 10 класу передбачено вивчення основ алгоритмізації, структур даних, базових мов програмування, а також опрацювання графіки та мультимедіа. Ці знання є основою для роботи з веб-технологіями, оскільки вони забезпечують розуміння принципів функціонування веб-ресурсів та необхідних інструментів для їх створення. Так, графічні елементи, створені під час роботи з мультимедіа, можуть бути інтегровані у веб-сайти, а навички алгоритмізації та програмування використовуються для створення інтерактивних компонентів, таких як форми, калькулятори чи динамічний контент. [18]

У програмі профільного рівня модуль «Веб-технології» має значно ширший обсяг, ніж у стандартній програмі, і включає не лише базові концепції, але й більш складні аспекти веб-розробки. Наприклад, учні знайомляться з основами роботи з базами даних, які інтегруються у веб-додатки, а також вивчають базові аспекти мов програмування для веб-розробки, таких як JavaScript. Це дозволяє створювати не лише статичні, але й динамічні сайти, які реагують на дії користувачів. Вивчення принципів адаптивного дизайну сприяє розумінню важливості оптимізації сайтів для різних пристроїв, що є актуальним у сучасному світі, де більшість користувачів переглядають веб-ресурси з мобільних телефонів або планшетів. [9]

Окрім технічних аспектів, модуль «Веб-технології» має значний вплив на розвиток творчого потенціалу учнів. Створення веб-сайтів вимагає від них не лише технічних знань, але й художнього смаку, уміння знаходити оригінальні рішення для дизайну та структури веб-ресурсів. Наприклад, під час створення веб-сторінки для власного проекту учні мають враховувати її естетичний вигляд, зручність використання, потреби цільової аудиторії. Це сприяє розвитку навичок системного мислення, яке дозволяє аналізувати завдання та знаходити оптимальні шляхи їх вирішення.

Методичний підхід до викладання модуля «Веб-технології» базується на поєднанні теоретичних та практичних занять. На теоретичних заняттях учні знайомляться з основними поняттями та принципами роботи веб-технологій, а на практичних заняттях вони застосовують ці знання для виконання реальних завдань. Це може бути створення персонального сайту, блогу, портфоліо або веб-ресурсу для бізнесу. Важливим елементом є проектна діяльність, яка дозволяє учням працювати над власними проектами, застосовуючи отримані знання у реальних умовах. Такий підхід сприяє розвитку самостійності, творчого підходу до роботи та відповідальності за результат.

Програма також передбачає використання сучасних інструментів для розробки та тестування веб-ресурсів. Наприклад, учні працюють із графічними редакторами, такими як Adobe Photoshop або Figma, для створення дизайну веб-сторінок, а також із текстовими редакторами для написання коду. Використання CMS дозволяє їм швидко створювати функціональні сайти, навіть без глибоких знань програмування, що мотивує до подальшого вивчення веб-технологій.

Важливо зазначити, що модуль «Веб-технології» має міждисциплінарний характер. Він інтегрує знання з інших предметів, таких як математика, англійська мова, мистецтво та навіть економіка. Наприклад, при створенні сайту для бізнес-проекту учні можуть застосовувати знання з економіки для розробки маркетингової стратегії, а знання англійської мови використовуються для

створення англomовного контенту. Така інтеграція сприяє формуванню системного мислення та розуміння ролі веб-технологій у сучасному суспільстві.

Модуль «Веб-технології» відіграє важливу роль у професійній орієнтації учнів. Вивчаючи основи веб-дизайну, програмування та адміністрування сайтів, школярі отримують уявлення про перспективи роботи у сфері інформаційних технологій. Це стимулює їх до подальшого навчання та розвитку в цьому напрямку.

Аналізуючи структуру навчальної програми, слід наголосити, що модуль «Веб-технології» виконує функцію підготовки учнів до практичного використання знань у реальному житті. Одним із ключових аспектів є те, що модуль передбачає не лише ознайомлення з базовими інструментами створення веб-сайтів, але й вивчення сучасних тенденцій у цій сфері. Важливо, що вивчення цього модуля дозволяє поєднати технічні знання з творчим підходом, що розвиває багатогранні навички у школярів.

Особливістю програми є акцент на формуванні інтегративного розуміння процесу створення веб-ресурсів. Наприклад, вивчення HTML забезпечує учням базові знання про структуру веб-сторінок, CSS надає можливість працювати з дизайном та стилізацією, а JavaScript дозволяє додавати інтерактивність. Ці елементи формують фундамент для подальшого освоєння більш складних технологій, таких як робота з серверною частиною або інтеграція веб-сайтів із базами даних.

Ще одним важливим аспектом є вивчення адаптивного дизайну, який є невід'ємною частиною сучасної веб-розробки. Зважаючи на зростаюче використання мобільних пристроїв, учні вчаться створювати веб-сайти, які будуть зручними для перегляду на різних екранах. Це включає вивчення медіа-запитів у CSS, принципів побудови гнучких макетів, а також тестування сайтів на різних пристроях. Такий підхід формує у школярів розуміння потреб сучасної аудиторії та навички адаптації до швидко змінюваних технологічних умов.

Практичні заняття з веб-технологій дозволяють учням застосувати теоретичні знання на практиці. Наприклад, вони можуть створювати прості сайти для шкільних проєктів, онлайн-портфоліо або навіть невеликі інтернет-магазини. Це не лише мотивує учнів до навчання, але й надає їм відчуття досягнення, коли вони бачать результат своєї роботи в реальному житті. Крім того, такі проєкти можуть стати основою для майбутньої професійної діяльності або додаткових курсів і навчання.

Особливу увагу варто звернути на роль вчителя у викладанні модуля «Веб-технології». Вчитель повинен не лише пояснювати технічні аспекти, але й виступати в ролі наставника, який спрямовує учнів і допомагає їм вирішувати проблеми, що виникають під час роботи над проєктами. Це може включати аналіз помилок у коді, поради щодо вибору інструментів або навіть обговорення загальних принципів організації роботи над проєктом. Такий підхід сприяє розвитку самостійності учнів і формує у них навички критичного мислення. [13]

Інтеграція модуля «Веб-технології» з іншими предметами є ще одним важливим аспектом. Наприклад, під час створення сайту на історичну тему учні можуть використовувати знання з історії, працюючи над контентом для сторінок. Знання з математики можуть знадобитися для розробки калькуляторів або інших інтерактивних елементів. Англійська мова стає незамінною при роботі з документацією або створенні англійськомовного контенту. Така міждисциплінарна інтеграція дозволяє учням бачити практичну цінність своїх знань і стимулює їх до подальшого навчання.

Програма також передбачає розвиток у школярів розуміння етики у сфері інформаційних технологій. Наприклад, під час роботи над проєктами обговорюються питання авторського права, конфіденційності даних, етичного використання інформації. Це формує у учнів відповідальне ставлення до своєї діяльності та розуміння важливості дотримання правил і норм у цифровому середовищі.

Важливим елементом є оцінювання навчальних досягнень учнів у рамках модуля «Веб-технології». Оцінювання базується на результатах виконання практичних завдань і проектів. При цьому враховуються не лише технічні аспекти (наприклад, функціональність веб-сайту), але й креативність, якість дизайну, відповідність завданню. Такий підхід дозволяє оцінити всі аспекти роботи учня і сприяє формуванню у них комплексного підходу до вирішення завдань.

Модуль «Веб-технології» відіграє важливу роль у підготовці учнів до подальшого навчання та професійної діяльності. Здобуті знання можуть бути використані як для продовження навчання у вищих навчальних закладах, так і для самостійної роботи у сфері інформаційних технологій. Крім того, вивчення веб-технологій допомагає учням розвивати навички, які можуть бути корисними у багатьох інших сферах, від дизайну до маркетингу.

Загалом, модуль «Веб-технології» є важливою складовою навчальної програми інформатики для старших класів. Він забезпечує учнів практичними знаннями, розвиває їх творчий потенціал і сприяє формуванню ключових компетенцій, необхідних для успішної інтеграції у сучасне інформаційне суспільство.

### **1.3 Значення модуля «Веб-технології» у формуванні цифрової грамотності старшокласників**

Цифрова грамотність стала однією з ключових компетентностей сучасної людини, без якої неможливо повноцінно функціонувати в інформаційному суспільстві. У світі, де технології швидко змінюють спосіб життя, освіти та професійної діяльності, здатність працювати з інформаційними системами, створювати цифровий контент і використовувати веб-ресурси є основою успіху. Саме тому розвиток цифрової грамотності є пріоритетом у шкільній освіті, особливо у старших класах, коли учні наближаються до вибору професійного шляху.



Цифрову грамотність можна визначити як здатність ефективно використовувати інформаційно-комунікаційні технології для пошуку, створення, обробки та обміну інформацією. Вона також включає вміння працювати з великими обсягами даних, адаптувати інформацію до потреб аудиторії, забезпечувати безпеку в онлайн-середовищі та використовувати цифрові інструменти для розв'язання практичних завдань. У контексті шкільної освіти цифрова грамотність виступає важливим компонентом підготовки учнів до майбутнього, оскільки більшість сучасних професій вимагають високого рівня володіння цифровими технологіями.

Модуль «Веб-технології», що входить до навчальної програми інформатики для 10–11 класів, є одним із найважливіших інструментів формування цифрової грамотності. Навчання за цим модулем забезпечує учнів як базовими знаннями про принципи функціонування веб-середовища, так і практичними навичками створення цифрових продуктів. Опанування веб-технологій дає змогу школярам не лише розуміти, як працюють веб-ресурси, але й самостійно створювати власні сайти, веб-додатки або портфоліо, що є надзвичайно актуальним у сучасному світі.

Одним із основних аспектів значення модуля «Веб-технології» є його спрямованість на розвиток технічних навичок. У ході вивчення цього курсу учні знайомляться з основами трьох ключових технологій: HTML, CSS та JavaScript. Кожна з них має своє унікальне призначення і сприяє формуванню комплексного розуміння роботи веб-середовища.

HTML є базовою мовою розмітки, що забезпечує структурування контенту на веб-сторінках. Завдяки вивченню HTML учні вчаться організовувати інформацію у вигляді заголовків, абзаців, таблиць і списків, що дозволяє створювати чіткі та структуровані веб-ресурси. CSS, у свою чергу, дає змогу стилізувати контент, додаючи кольори, шрифти, макети та інші візуальні елементи. Це розвиває у школярів не лише технічні навички, але й естетичний смак, який є важливим при створенні веб-дизайну.

JavaScript забезпечує інтерактивність веб-сторінок, дозволяючи створювати динамічні елементи, такі як кнопки, форми, анімації та спливаючі вікна. Робота з JavaScript сприяє формуванню алгоритмічного мислення, оскільки створення сценаріїв потребує розуміння логіки програмування та здатності аналізувати виконання коду.

Вивчення цих технологій дозволяє учням створювати повноцінні веб-продукти, які відповідають сучасним стандартам. Окрім цього, модуль «Веб-технології» формує у школярів практичні навички, які можуть бути використані як у навчанні, так і в майбутній професійній діяльності.

Навчання за модулем «Веб-технології» сприяє не лише розвитку технічних знань, але й формує у школярів креативність та здатність до нестандартного мислення. Створення веб-дизайну є складним завданням, яке вимагає врахування багатьох аспектів: від підбору кольорової гами та шрифтів до розробки макетів сторінок і інтерактивних елементів.

Розробка веб-ресурсів передбачає адаптацію контенту до потреб цільової аудиторії, що стимулює учнів до аналітичного підходу у прийнятті рішень. Вони навчаються враховувати потреби користувачів, забезпечувати зручність навігації та доступність інформації. Це розвиває їхнє критичне мислення, здатність до планування й оцінки результатів своєї роботи.

Крім того, виконання творчих завдань, таких як створення персонального портфоліо або групових проєктів, мотивує школярів до самовираження через цифрові засоби. Це сприяє не лише їхньому технічному, але й особистісному розвитку.

Однією з головних переваг модуля «Веб-технології» є його практична спрямованість. Усі теоретичні знання, отримані під час уроків, закріплюються через виконання реальних завдань, які відповідають вимогам сучасного ринку праці. Наприклад, учні створюють прості веб-сторінки, додають до них стильові елементи, працюють із анімаціями та інтерактивними кнопками, публікують свої проєкти в Інтернеті.

Важливим аспектом навчання є організація групових проєктів. У процесі роботи над спільним завданням учні вчаться ефективно комунікувати, розподіляти обов'язки, вирішувати конфлікти та досягати поставлених цілей. Це не лише розвиває навички командної роботи, але й готує школярів до майбутніх професій, де такі вміння є обов'язковими.

Модуль «Веб-технології» дає змогу учням адаптуватися до викликів сучасного цифрового середовища. Уміння працювати з веб-ресурсами, створювати інтерактивний контент та аналізувати інформацію є необхідними у багатьох професіях, починаючи від маркетингу й закінчуючи розробкою програмного забезпечення.

Окрім цього, опанування веб-технологій стимулює учнів до подальшої самоосвіти. У процесі навчання школярі знайомляться з різними онлайн-ресурсами, такими як GitHub, CodePen або W3Schools, які дозволяють продовжувати навчання самостійно. Це формує у них здатність до самоосвіти, яка є необхідною у світі, де знання швидко застарівають, а технології постійно розвиваються.

Модуль «Веб-технології» відіграє важливу роль у формуванні цифрової грамотності старшокласників. Він забезпечує учнів знаннями та навичками, які дозволяють розуміти принципи функціонування веб-середовища, створювати власні цифрові продукти й адаптуватися до сучасного інформаційного суспільства. Завдяки своїй практичній спрямованості, креативному потенціалу й орієнтації на сучасні технології, цей модуль є важливим компонентом шкільної програми інформатики, який сприяє підготовці школярів до майбутнього.

#### **1.4 Теоретичні основи викладання веб-технологій: підходи, принципи, ключові компетентності**

Ефективне навчання веб-технологій у шкільному курсі інформатики залежить від поєднання кількох теоретичних і практичних компонентів. Вивчення веб-технологій, таких як HTML, CSS і JavaScript, вимагає не лише засвоєння технічних знань, але й усвідомлення педагогічних підходів, які

лежать в основі навчального процесу. Теоретичні основи викладання веб-технологій включають визначення підходів до навчання, ключових принципів організації освітнього процесу та компетентностей, які повинні формуватися у старшокласників у результаті навчання.

У сучасній педагогіці наголос робиться на тому, що успішне навчання має не лише передавати знання, але й формувати здатність застосовувати їх на практиці. Це особливо важливо для навчання веб-технологій, оскільки їх вивчення орієнтоване на створення реальних цифрових продуктів, які учні можуть використовувати для подальшого розвитку або презентації своїх умінь.

Одним із ключових підходів, який використовується при викладанні веб-технологій, є компетентнісний підхід. У контексті шкільної освіти він передбачає орієнтацію на формування у школярів не лише знань, але й умінь і навичок, які вони можуть застосовувати у реальних життєвих ситуаціях. Це відповідає сучасним вимогам до освіти, які наголошують на важливості практичної спрямованості навчання.

Вивчення веб-технологій у старшій школі сприяє формуванню таких ключових компетентностей:

- Інформаційно-цифрова грамотність. Учні навчаються ефективно працювати з інформаційними ресурсами, створювати та опрацьовувати цифровий контент.
- Навички розв'язання проблем. Робота з веб-технологіями вимагає аналізу завдань, пошуку оптимальних рішень і впровадження їх у цифровому середовищі.
- Командна співпраця. Групові проєкти сприяють розвитку вміння працювати в колективі, що є важливою складовою сучасного професійного життя.
- Самоорганізація та відповідальність. Учні планують свою роботу, виконують завдання у встановлені терміни, презентують результати своєї діяльності.

Компетентнісний підхід дозволяє забезпечити зв'язок між теоретичними знаннями та їх застосуванням у реальних умовах. Він також сприяє формуванню у старшокласників здатності до самоосвіти, яка є необхідною у світі, що швидко змінюється.

Проектне навчання є одним із найефективніших підходів до викладання веб-технологій у школі. Цей метод дозволяє учням працювати над створенням реальних продуктів, таких як веб-сторінки, персональні портфоліо або тематичні сайти. Основною особливістю проектного навчання є його практична спрямованість: учні не просто вивчають теоретичний матеріал, але й застосовують його для досягнення конкретної мети.

Проектне навчання забезпечує:

- Залучення учнів до активної діяльності. Вони самостійно обирають теми для своїх проєктів, планують роботу, аналізують результати.
- Можливість творчого самовираження. Створення дизайну веб-ресурсів стимулює креативність учнів, дозволяючи їм реалізувати свої ідеї.
- Формування навичок самостійної роботи. Учні навчаються самостійно знаходити необхідну інформацію, використовувати ресурси та інструменти для виконання завдань.

Проектне навчання також сприяє інтеграції знань з різних галузей. Наприклад, створення веб-дизайну вимагає врахування основ графічного дизайну, робота з текстом — розуміння основ мовознавства, а програмування інтерактивних елементів — знань алгоритмів і логіки. [17]

Інтерактивні методи є важливим компонентом навчання веб-технологій, оскільки вони дозволяють зробити освітній процес цікавим і ефективним. Використання таких методів, як мозкові штурми, групові дискусії, симуляції або кейс-методи, активізує пізнавальну діяльність учнів, підвищує їхню зацікавленість у темі.

Одним із прикладів інтерактивного навчання є використання онлайн-платформ, які дозволяють учням працювати з кодом у реальному часі. Такі інструменти, як CodePen, JSFiddle або GitHub Pages, надають можливість

практикуватися у написанні HTML, CSS і JavaScript, бачачи результати своєї роботи одразу. Це сприяє закріпленню теоретичних знань і підвищує мотивацію учнів до навчання.

Інтерактивні методи також сприяють розвитку комунікативних навичок, оскільки передбачають активну взаємодію між учнями. Наприклад, під час групових обговорень учні обмінюються ідеями, шукають спільні рішення, допомагають одне одному в розв'язанні проблем. Це формує у них вміння ефективно спілкуватися, що є важливим для майбутнього професійного життя.

Організація навчання веб-технологій базується на кількох принципах, які забезпечують ефективність освітнього процесу. Одним із ключових є принцип поступовості: учні спочатку вивчають основи, такі як структура HTML-документа, і лише після цього переходять до складніших тем, наприклад, створення інтерактивних елементів за допомогою JavaScript.

Принцип доступності передбачає використання навчальних матеріалів, які відповідають рівню підготовки учнів. Наприклад, для старшокласників, які тільки починають вивчати веб-технології, важливо пояснювати основні поняття простою мовою, використовуючи приклади з повсякденного життя.

Ще одним важливим принципом є принцип актуальності. Знання, які учні отримують під час навчання, повинні бути практично застосовними. Наприклад, замість створення статичних веб-сторінок, які сьогодні рідко використовуються, учням варто показувати, як створювати адаптивні сайти, що коректно відображаються на мобільних пристроях.

Принцип індивідуалізації дозволяє враховувати особливості кожного учня, його інтереси та здібності. Наприклад, для учнів, які цікавляться дизайном, можна запропонувати завдання зі створення макетів веб-сторінок, тоді як для тих, хто захоплюється програмуванням, – завдання з написання інтерактивних сценаріїв.

Теоретичні основи викладання веб-технологій формують базу для створення ефективної методики навчання, яка дозволяє поєднати теоретичні знання з практичними навичками. Компетентнісний підхід, проектне навчання

та інтерактивні методи забезпечують різнобічний розвиток учнів, формуючи у них не лише технічні знання, але й творчі, комунікативні та організаційні вміння. Завдяки дотриманню основних педагогічних принципів навчання веб-технологій стає доступним, цікавим і практично орієнтованим, що відповідає вимогам сучасної освіти. [17]

### **1.5 Основний зміст і тематика модуля «Веб-технології»**

Вибірковий модуль «Веб-технології» спрямований на формування в учнів практичних навичок роботи з веб-ресурсами. Його структура побудована таким чином, щоб забезпечити поступове й логічне засвоєння знань: від простих понять до складних тем. Тематика модуля включає три основні компоненти веб-розробки — HTML, CSS і JavaScript, а також адаптивний дизайн як важливий аспект сучасного веб-програмування.

Розробка змісту модуля базується на сучасних педагогічних підходах, які поєднують теоретичне вивчення основ веб-технологій із виконанням практичних завдань. Це забезпечує засвоєння матеріалу через практичне застосування знань. Наприклад, після вивчення теми HTML учні одразу закріплюють знання через створення базових веб-сторінок.

Зміст модуля охоплює такі ключові теми: основи мови розмітки HTML, каскадні таблиці стилів CSS, інтерактивні можливості JavaScript та принципи адаптивного дизайну. Усі ці компоненти створюють цілісну картину розробки веб-ресурсів, формуючи в учнів розуміння, як окремі елементи інтегруються у функціональний веб-продукт.

Мова розмітки HTML (HyperText Markup Language) є основою для створення веб-сторінок. Її головна функція — надання структурної організації веб-документів. У межах модуля учні знайомляться з базовими елементами HTML, вчать створювати структури сторінок, які включають заголовки, абзаци, списки, зображення та посилання.

Заняття, присвячені HTML, починаються з розгляду основних тегів, їхнього синтаксису та атрибутів. Наприклад, учні дізнаються про теги <html>

<head>, <body>, а також вивчають, як використовувати <h1>–<h6> для створення заголовків різного рівня. Для більш глибокого розуміння структури веб-документів учням пропонують створювати сторінки, які включають текст, таблиці, мультимедіа та навігаційні елементи.

Практична частина занять із HTML включає створення персонального веб-портфоліо, де кожен учень може представити свої проєкти. Це завдання дозволяє закріпити теоретичні знання, даючи змогу учням побачити результати своєї роботи в реальному веб-середовищі.

Наступним етапом у вивченні веб-технологій є ознайомлення з каскадними таблицями стилів CSS (Cascading Style Sheets). CSS є інструментом для оформлення веб-сторінок, що дозволяє змінювати кольори, шрифти, розташування елементів та інші візуальні характеристики.

На заняттях учні вивчають принципи зв'язування HTML із CSS, розглядають внутрішні, зовнішні та інлайн-стили. У процесі навчання школярі дізнаються, як використовувати властивості CSS для управління текстом, кольоровою гамою, відступами, межами та вирівнюванням елементів.

Однією з важливих тем є робота з селекторами CSS, які визначають, до яких елементів застосовуються стилі. Учні вчаться використовувати прості селектори, наприклад, `p` чи `h1`, і комбіновані селектори для створення складніших стилів. Крім того, школярі освоюють концепцію класів і ідентифікаторів, що дозволяє гнучко стилізувати елементи.

Практичні завдання на цьому етапі передбачають створення стилізованих веб-сторінок. Учні працюють над проєктами, де вони обирають кольорову гаму, шрифти та макети, щоб розробити гармонійний дизайн веб-сайту. Такі завдання сприяють розвитку візуального мислення й творчих здібностей.

JavaScript є ключовим компонентом веб-розробки, що дозволяє створювати інтерактивні елементи на веб-сторінках. У рамках модуля учні ознайомлюються з основами мови JavaScript, включаючи змінні, функції, цикли й обробку подій.



Навчання починається з розгляду базового синтаксису JavaScript та основних конструкцій, таких як змінні (var, let, const), умови (if-else) і цикли (for, while). Далі школярі вчаться створювати функції, які дозволяють організувати код у логічні блоки.

Особливу увагу приділяють взаємодії JavaScript із HTML і CSS через DOM (Document Object Model). Учні вчаться змінювати вміст сторінки, додавати анімації та інтерактивні елементи, такі як кнопки, форми й меню. Це дозволяє створювати динамічні веб-сторінки, які реагують на дії користувачів.

Практичні завдання включають розробку інтерактивних веб-додатків. Наприклад, учні можуть створювати калькулятори, інтерактивні галереї зображень або прості ігри. Ці завдання сприяють закріпленню знань і розвитку навичок алгоритмічного мислення.

Адаптивний дизайн є важливою темою, оскільки більшість сучасних користувачів використовують мобільні пристрої для доступу до веб-ресурсів. У рамках модуля учні вивчають принципи адаптивного дизайну, які забезпечують коректне відображення веб-сторінок на різних пристроях.

На заняттях розглядаються основи використання медіа-запитів CSS, які дозволяють змінювати стиль залежно від розміру екрана. Учні вчаться створювати адаптивні макети, використовуючи гнучкі сітки та одиниці вимірювання, такі як відсотки, em і rem.

Практична частина включає розробку веб-сайтів, які адаптуються до різних розмірів екранів. Це дозволяє учням зрозуміти, як важливо забезпечувати зручність користування веб-ресурсами для різних категорій користувачів.

Модуль «Веб-технології» охоплює основи HTML, CSS, JavaScript та адаптивного дизайну, що дозволяє учням здобути комплексні знання й навички у сфері веб-розробки. Завдяки чіткій структурі та логічній послідовності тем, цей курс забезпечує поступове засвоєння матеріалу, дозволяючи учням створювати повноцінні веб-продукти. [16]

## РОЗДІЛ II. ПРАКТИЧНА МЕТОДИКА ВИКЛАДАННЯ ВИБІРКОВОГО МОДУЛЯ «ВЕБ-ТЕХНОЛОГІЇ»

### 2.1. Постановка задачі дослідження: основні завдання методичної системи навчання

Модуль «Веб-технології» є важливим компонентом сучасної шкільної програми з інформатики для старших класів. Його викладання має на меті не лише ознайомлення учнів із теоретичними аспектами веб-розробки, але й формування практичних навичок створення веб-ресурсів, розуміння принципів їх функціонування та ролі в сучасному інформаційному середовищі. З огляду на стрімкий розвиток цифрових технологій, актуальність такого навчання постійно зростає, оскільки веб-технології є основою для більшості інтернет-ресурсів, мобільних застосунків та інтерактивних сервісів, які використовуються у повсякденному житті.

Ефективне викладання модуля «Веб-технології» потребує розробки чіткої методичної системи, яка включає визначення змісту курсу, форм і методів навчальної діяльності, а також критеріїв оцінювання результатів. Основним завданням цього дослідження є створення такої методичної системи, яка б відповідала сучасним вимогам до освіти, враховувала вікові особливості учнів і сприяла формуванню в них ключових компетентностей.

Метою дослідження є розробка практичної методики викладання вибіркового модуля «Веб-технології» для учнів 10–11 класів, яка забезпечує гармонійне поєднання теоретичних знань і практичних навичок, необхідних для створення повноцінних веб-ресурсів.

Для досягнення цієї мети було визначено такі основні завдання:

- Визначити змістове наповнення модуля «Веб-технології», що включає основи HTML, CSS, JavaScript та адаптивного дизайну.
- Розробити ефективні форми організації навчальної діяльності, які забезпечують активну участь учнів у навчальному процесі.

- Створити методику проведення практичних занять, спрямованих на закріплення знань через виконання реальних завдань.
- Розробити критерії оцінювання успішності учнів, які враховують як якість виконаних завдань, так і їхню творчість і креативність.
- Провести апробацію розробленої методики в реальних умовах шкільного навчання для оцінки її ефективності.

Розробка методичної системи навчання базується на кількох основних педагогічних підходах:

- Компетентнісний підхід. Основою навчання є формування у школярів практичних компетентностей, які дозволяють ефективно застосовувати знання у реальних життєвих і професійних ситуаціях.
- Проектно-орієнтоване навчання. Учні виконують проекти, які об'єднують теоретичні знання та практичні завдання, наприклад, створення власного веб-сайту.
- Інтерактивний підхід. Навчальний процес включає активну взаємодію між учнями та вчителем, роботу в групах, обговорення, мозкові штурми.
- Особистісно орієнтований підхід. Завдання враховують індивідуальні особливості учнів, їхні інтереси та рівень підготовки.

Методологічна база також включає аналіз сучасних підручників, програмних засобів і онлайн-ресурсів, які можуть бути використані для викладання модуля «Веб-технології».

Методична система викладання модуля «Веб-технології» включає кілька ключових компонентів:

- Змістове наповнення курсу. Воно охоплює базові теми веб-розробки, які вивчаються у логічній послідовності: HTML (структура веб-сторінок), CSS (оформлення та дизайн), JavaScript (інтерактивні елементи) і адаптивний дизайн (пристосування сторінок до різних пристроїв).
- Форми навчальної діяльності. Викладання передбачає поєднання індивідуальних і групових занять, фронтальної роботи та самостійного

виконання завдань. Особлива увага приділяється груповим проектам, які стимулюють командну співпрацю.

- Методи навчання. Основними методами є демонстрація, пояснення, виконання практичних завдань і проектна робота. Крім того, використовуються інтерактивні методи, такі як кейс-методи, робота з онлайн-ресурсами (CodePen, JSFiddle) і обговорення реальних прикладів.
- Оцінювання результатів. Для оцінки успішності навчання розроблено критерії, які враховують якість виконаних завдань, рівень творчості, складність проекту та здатність учня презентувати результати своєї роботи.

Реалізація розробленої методики викладання модуля «Веб-технології» має забезпечити такі результати:

- Поглиблення знань учнів у сфері веб-розробки.
- Формування практичних навичок створення веб-ресурсів, зокрема вміння працювати з HTML, CSS і JavaScript.
- Розвиток творчих здібностей, критичного мислення та вміння працювати в команді.
- Мотивація учнів до подальшого вивчення веб-технологій і використання отриманих знань у професійній діяльності.

Постановка задачі дослідження є важливим етапом у розробці методичної системи навчання. Чітке визначення мети, завдань, методологічної основи та очікуваних результатів дозволяє створити ефективну систему викладання модуля «Веб-технології», яка сприятиме формуванню у старшокласників ключових компетентностей, необхідних для їхньої професійної підготовки й особистісного розвитку.

## **2.2 Розробка змістового наповнення модуля**

Розробка змістового наповнення модуля «Веб-технології» для учнів 10–11 класів передбачає надзвичайно ретельне опрацювання навчального матеріалу, який би у логічній, послідовній і гнучкій формі забезпечував глибинне

розуміння веб-технологій та формування у школярів практичних навичок. Враховуючи стрімкий розвиток інформаційних технологій, ці знання стають не тільки корисними у подальшій професійній діяльності, а й необхідними в контексті повсякденного життя у цифровому світі. При цьому слід орієнтуватися на компетентнісний підхід, інтерактивні та проектні методи навчання, а також на розвиток критичного мислення, креативності, комунікативних навичок і здатності до самостійного пошуку та аналізу інформації.

Загальні принципи побудови змісту модуля:

- логічна послідовність;
- компетентнісний та діяльнісний підхід;
- поступовий перехід від керованої діяльності до самостійної творчості;
- адаптивність до індивідуальних потреб та рівнів підготовки.

Матеріал модуля поділяється на тематичні блоки, які послідовно розкриваються від простого до складного. Спочатку учні знайомляться з основами HTML як базовою мовою розмітки веб-сторінок, потім опановують CSS для оформлення та стилізації контенту, після цього вивчають основи JavaScript для додавання інтерактивності й динамічної взаємодії з користувачем, а наприкінці знайомляться з концепціями адаптивного дизайну, що дозволяє забезпечити коректне відображення веб-ресурсів на різних пристроях.

Зміст модуля орієнтований на формування здатності застосовувати отримані знання у практичних ситуаціях. Учні не тільки заучують теоретичні поняття, але й реалізують їх у власних проектах, виконують реальні практичні завдання, аналізують помилки, вдосконалюють свої рішення. Це сприяє розвитку стійких навичок та впевненості у власних можливостях.

Спочатку навчання відбувається під безпосереднім керівництвом учителя. Виконуються типові завдання з детальними інструкціями, учні знайомляться з прикладами коду та базовими прийомами. З часом ступінь самостійності зростає: учні опрацьовують складніші проекти, створюють власні веб-сторінки, підбирають дизайн та функціональність відповідно до обраної теми,

використовуючи наявні знання й консультуючись з учителем за потреби. Зрештою, наприкінці курсу вони реалізують повноцінні проекти фактично без детальних інструкцій, проявляючи власну ініціативу.

Зміст курсу можна підлаштовувати під різні рівні підготовки учнів. Для більш підготовлених учнів можна запропонувати складніші завдання, наприклад, розробку складних анімацій, інтеграцію сторонніх бібліотек, покращення продуктивності чи оптимізацію для SEO. Учні з меншими навичками можуть зосередитися на базових аспектах і поступово рухатись вперед.

Модуль умовно поділяється на чотири основні тематичні блоки:

- Основи HTML
- CSS та основи веб-дизайну
- Основи JavaScript та інтерактивність
- Адаптивний дизайн та підсумковий проект

В межах кожного з блоків будемо виділяти основні теоретичні аспекти, перелік завдань для практичного виконання учнями та тематику міні-проектів для розвитку набутих компетентностей.

У тематичному блоці “Основи HTML” учні знайомляться з наступними теоретичними аспектами модуля: концепцією гіпертексту, історією виникнення та роллю HTML у створенні веб-сторінок. Розглядаються поняття тегів, елементів, атрибутів, структура стандартного HTML-документа (декларація типу документа, елементи `<html>`, `<head>`, `<body>`). Вивчаються базові теги форматування тексту, створення заголовків `<h1>`–`<h6>`, абзаців `<p>`, списків `<ul>`, `<ol>`, `<li>`, посилань `<a>` та зображень `<img>`. Далі учні знайомляться з таблицями `<table>`, `<tr>`, `<td>`, формами `<form>`, `<input>`, `<textarea>`, `<select>` та іншими елементами, що дозволяють структурувати дані і взаємодіяти з користувачем.

При цьому доцільно виконувати такі практичні завдання: створення простої веб-сторінки зі структурою HTML; додавання заголовків, абзаців,

зображень, гіперпосилань на внутрішні та зовнішні ресурси; формування простих форм для введення даних користувача.

В якості міні-проектів даного змістового блоку пропонуємо: розробку стартової сторінки для шкільного сайту, наприклад, сторінки домашнього завдання, або інформаційної сторінки про шкільний захід (конференція, виставка). Учні працюють над версткою цієї сторінки, додаючи контент, забезпечуючи логічну структуру та надаючи основну інформацію.

Після цього доцільно переходити до опанування змістовим блоком “CSS та основи веб-дизайну”, де вивчається призначення CSS як каскадних таблиць стилів для відокремлення змісту від презентації. Учні дізнаються про різні способи підключення CSS: внутрішні стилі (всередині `<style>`), зовнішні стилі (файл `.css`), вбудовані стилі (через атрибути), а також про поняття каскаду, специфічності селекторів. Розглядаються різні селектори (елементів, класів, ідентифікаторів, атрибутів, псевдокласів, псевдоелементів), принципи використання шрифтів, кольорів, фону, відступів, меж, а також складніші інструменти побудови макетів за допомогою Flexbox і CSS Grid.

В якості практичних завдань, які виконуються на уроках, можна запропонувати: створення зовнішнього файлу стилів і підключення його до HTML-сторінки. Встановлення фонового зображення, підбір кольорової гами та шрифтів. Верстка простого макету сторінки із використанням `display: flex;` чи `display: grid;`. Вирівнювання елементів, додавання стилів для навігаційного меню, оформлення текстових блоків та заголовків.

При виконанні міні-проектів варто продовжити попередні завдання, звернувши увагу на оформлення попередньо створеної HTML-сторінки за допомогою CSS у єдиному корпоративному стилі. Наприклад, учні можуть створити тематичну сторінку «Віртуальний музей школи» з оформленням у певній кольоровій гамі, застосувати шрифти, які підкреслюють стилістику сторінки. Створення макету блогу чи невеликого онлайн-журналу, де кожен елемент має гармонійний зовнішній вигляд.

У теоретичному блоці “Основи JavaScript та інтерактивність” учні знайомляться з роллю JavaScript у житті сучасного вебу. Вивчаються змінні, типи даних, базові оператори, функції, умовні конструкції, цикли, а також поняття подій у браузері (onClick, onSubmit, onMouseOver тощо). Розглядається робота з DOM (Document Object Model), пошук елементів сторінки за допомогою `document.getElementById()`, `document.querySelector()`, зміна вмісту елементів, CSS-класів, стилів. Важливим аспектом є валідація форм, прості маніпуляції з даними, інтерактивні ефекти, анімації та динамічне завантаження контенту.

При цьому виконуються такі практичні завдання: додавання на сторінку інтерактивних елементів, наприклад, кнопки, яка змінює колір фону при натисканні, або реалізація простих слайдерів зображень, перевірка полів форми перед надсиланням, створення списку завдань (ToDo List) з можливістю додавання та видалення елементів.

Темами міні-проектів можуть служити: Розробка інтерактивної галереї фотографій, де зображення змінюються при натисканні на стрілки чи мініатюри. Створення простої онлайн-гри (наприклад, «Вгадай число» або «Пам’ятай порядок»), де користувач взаємодіє із сторінкою, а результати гри відображаються на екрані. Створення інтерактивних форм реєстрації/авторизації, які перевіряють правильність введених даних у реальному часі.

Завершується вивчення модуля темою “Адаптивний дизайн та підсумковий проєкт”, де учні дізнаються про принципи адаптивного та респонсивного дизайну, медіа-запити (media queries), підходи до оптимізації сторінок під екрани різного розміру. Розглядаються найкращі практики для мобільних пристроїв, застосування відносних одиниць вимірювання, простота навігації, оптимізація зображень та шрифтів.

Практичні завдання: створення верстки сторінки, яка змінює свій вигляд при зменшенні чи збільшенні ширини вікна; налаштування меню, яке



перетворюється у випадające при вузькому екрані; оптимізація зображень для швидкого завантаження.

В якості підсумкового проєкту з модуля учням пропонується розробка повноцінного міні-сайту або інтернет-сторінки інформаційного ресурсу на основі набутих знань та навичок. Учні самі обирають тему: це може бути сайт шкільного гуртка, інтернет-візитка учня, невеликий онлайн-каталог з продукцією, блог про технології чи мистецтво. Головна умова – використання HTML, CSS та JavaScript, забезпечення адаптивного дизайну під різні пристрої. Проєкт може включати декілька сторінок, навігаційне меню, інтерактивні елементи, форми з валідацією та стилізацією. Учні презентують свій проєкт перед класом: розповідають про ідею, застосовані технології, складнощі, з якими зіткнулися, та шляхи їх подолання.

Зупинимось детальніше на характеристиці практичних завдань, які доцільно використовувати в межах вивчення модуля. Ми рекомендуємо при вивченні Веб технологій застосування практико орієнтованого підходу, коли виконання практичних завдань супроводжує вивчення теоретичного матеріалу. Учні не просто читають та слухають пояснення, а одразу закріплюють їх, застосовуючи знання на практиці. Завдання можуть мати різну складність та формат:

Базові завдання допомагають закріпити нову інформацію відразу після пояснення. Наприклад, додати на сторінку HTML два абзаци тексту, зображення з підписом, створити базові стилі для заголовку чи застосувати просту функцію JavaScript, яка виводить повідомлення.

Середні за складністю завдання учні виконують після опрацювання певного тематичного блоку. Це може бути створення макету веб-сторінки, застосування кількох CSS-селекторів, зміна стилів при наведенні курсора або розробка простої інтерфейсної взаємодії за допомогою JavaScript (наприклад, при натисканні кнопки змінюється вміст елемента).

Наступний рівень – творчі та дослідницькі завдання. Мета таких завдань – стимулювати учнів до самостійного пошуку інформації, креативності та

нестандартного підходу. Наприклад, учням пропонується створити унікальний дизайн сторінки, вигадати цікаву тему для сайту, вбудувати інтерактивний елемент, який вони раніше не розглядали на уроці, або випробувати можливості CSS-анімацій. Це сприяє розвитку інноваційного мислення та глибшому розумінню веб-технологій.

Кульмінацією навчального процесу є проєктна діяльність, яка дозволяє об'єднати в єдине ціле увесь вивчений матеріал. Проєкт може бути індивідуальним або груповим. Перевага групових проєктів полягає у розвитку комунікативних навичок, здатності розподілити обов'язки, співпрацювати, приймати колективні рішення та долати конфлікти. Під час проєктної роботи учні самостійно обирають тему, аналізують реальні потреби користувачів (наприклад, створюють сайт з порадами для старшокласників), працюють над дизайном, інформаційною структурою, функціональністю. Вони можуть використовувати додаткові онлайн-інструменти, галереї ідей, форуми для вирішення проблем з версткою чи JavaScript.

На завершення проєкту передбачена презентація результатів перед класом. Учні демонструють готовий сайт, розповідають про особливості реалізації, обґрунтовують вибір технологій, пояснюють, як вони вирішували проблеми. Такий виступ сприяє формуванню навичок публічних виступів, аргументації, вмінню коректно відповідати на запитання, приймати критику та надавати пропозиції щодо покращення.

Приклади можливих проєктів:

- Сайт-каталог з описом книжок, фільмів або музики, що подобаються учням. Кожен елемент каталогу містить зображення, текстовий опис, посилання на сторонні ресурси.
- Інтерактивний сайт-шпаргалка з формулами з математики чи фізики з можливістю перевірки правильності рішень простих задач за допомогою JavaScript.

- Міні-сайт шкільного фестивалю або культурного заходу з програмою виступів, галереєю фотографій, формами реєстрації гостей, адаптивним дизайном для перегляду з телефону.
- Прототип односторінкового веб-додатку (landing page) для уявного стартапу, що пропонує інноваційний продукт або послугу.

Для реалізації змісту модуля рекомендовано використовувати сучасні засоби навчання та інструменти розробки:

- Онлайн-середовища для тестування коду: CodePen, JSFiddle, Replit – ці платформи дозволяють учням експериментувати з HTML, CSS і JavaScript у режимі реального часу, спостерігати результати роботи коду миттєво, ділитися посиланнями на свої напрацювання з учителем і однокласниками.
- Онлайн-курси та відеоуроки: Безкоштовні матеріали на платформах типу Codecademy, W3Schools, Khan Academy, Udemy (безкоштовні курси), а також освітні YouTube-канали, де автори детально пояснюють різноманітні аспекти веб-розробки. Учні можуть переглядати ці матеріали для додаткового опрацювання та поглиблення знань.
- Готові приклади коду та бібліотеки: Учитель може демонструвати учням приклади CSS-фреймворків (Bootstrap, Tailwind) або прості бібліотеки JavaScript для вирішення конкретних завдань. Хоча у шкільному курсі може бути обмежено використання сторонніх інструментів, ознайомлення з ними допоможе зрозуміти, як у реальній практиці розробники спрощують собі роботу.
- Посібники та книжки: Використання актуальної літератури з веб-розробки. Наприклад, книги з основ HTML5 та CSS3, JavaScript для початківців. Учні можуть мати додаткові джерела для поглибленого вивчення та підготовки.

Розробляючи змістове наповнення модуля, варто паралельно передбачити систему оцінювання, яка б враховувала: технічну коректність коду, завершеність та функціональність розроблених проєктів, якість дизайну та адаптивність,

креативність та оригінальність, здатність аргументувати свої рішення та презентувати роботу. Характеристику основних показників, які дозволяють оцінити якість виконаних завдань наводимо в Таблиці 2.1.

Таблиця 2.1.

### Критерії оцінювання практичних завдань та проєктів

Критерій оцінювання практичного завдання	Основні показники критерію
Технічна коректність коду	Відсутність синтаксичних помилок, коректне використання тегів, властивостей CSS, правильно оголошені й використані змінні JavaScript.
Завершеність та функціональність	Наскільки створені сторінки або проєкти відповідають поставленим вимогам, чи всі елементи інтерфейсу працюють, чи інтерфейс зрозумілий для користувача.
Якість дизайну та адаптивність	Чи сторінка має привабливий та логічний дизайн, чи коректно відображається на різних пристроях, чи зрозуміло структуровано інформацію, чи зручно користуватись навігацією
Креативність та оригінальність	Чи застосовували учні власні ідеї, нестандартні підходи до вирішення завдань, чи використовували додаткові інструменти і прийоми, які не були детально розглянуті на уроці.

Здатність аргументувати свої рішення та презентувати роботу:	Учні повинні вміти пояснити, чому вони обрали саме такий дизайн, структуру чи метод, як працює той чи інший фрагмент коду, які труднощі виникали під час роботи над проектом і як вони їх подолали.
--------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Отже, розробка змістового наповнення модуля «Веб-технології» передбачає створення цілісної системи, в якій кожен тематичний блок логічно доповнює попередній, а набуті знання відразу закріплюються через практичну діяльність. Зміст охоплює фундаментальні поняття та технології створення веб-сайтів (HTML, CSS, JavaScript), розширюється адаптивним дизайном і завершується комплексним проектом, що інтегрує всі здобуті вміння. Такий підхід сприяє формуванню в учнів всебічної компетентності у галузі веб-розробки, розвиває їхню самостійність, творчий потенціал, уміння працювати в команді та публічно презентувати результати власної діяльності. Це не лише готує старшокласників до можливих професійних перспектив у сфері ІТ, але й надає важливі життєві навички, необхідні у сучасному високотехнологічному суспільстві.

### 2.3 Форми та методи викладання модуля

Викладання модуля «Веб-технології» у старших класах школи є одним із ключових напрямів сучасної інформатичної освіти, оскільки веб-технології сьогодні становлять основу інформаційного суспільства. Уміння створювати веб-ресурси, розуміти принципи адаптивного дизайну, застосовувати мови розмітки та стилізації (HTML, CSS), а також інтерактивні скрипти (JavaScript) є важливими навичками, які можуть знадобитися учням як у майбутній професійній діяльності, так і в повсякденному житті. Проте навчання веб-технологій не може обмежуватися суто теоретичною площиною чи

традиційними лекційно-репродуктивними методами. Для досягнення якісних результатів потрібен комплексний підхід, що стимулює творче мислення, критичний аналіз, самостійний пошук рішень, практичну діяльність і командну співпрацю.

У цьому контексті особливого значення набуває використання проектної методики та кейс-методів. Ці дві педагогічні технології доповнюють одна одну, створюючи динамічне навчальне середовище, де учні не просто опановують певні технічні аспекти веб-розробки, а й вчаться мислити цілісно, орієнтуватися в сучасному інформаційному просторі, приймати рішення, аргументувати свої точки зору, співпрацювати з однолітками та прагнути вдосконалення. Застосування проектної методики та кейс-методів забезпечує розвиток так званих «м'яких навичок» (soft skills), які є не менш важливими за технічні компетенції.

### **2.3.1. Переваги використання проектної методики під час вивчення модуля “Веб-технології”**

Проектна методика належить до інноваційних педагогічних технологій, які ставлять учня в центр навчального процесу. В основі проектного підходу лежить ідея, що знання стають справді цінними тоді, коли вони застосовуються для розв'язання реальних задач. Таким чином, учні, які навчаються за проектною методикою, не обмежуються пасивним сприйняттям інформації, а беруть активну участь у процесі творення. Проекти у сфері веб-технологій можуть охоплювати дуже широкий спектр тем: від створення простих інформаційних сторінок до розробки функціональних порталів, інтерактивних інструментів, міні-ігор, адаптивних лендингів і навіть невеликих веб-додатків.

Основні принципи проектної методики:

- Спрямованість на кінцевий продукт.
- Інтеграція знань та міжпредметні зв'язки.
- Планування і системне мислення.
- Командна робота та розподіл ролей.

— Творчість та самостійність.

Результатом роботи над проектом є цілком реальний веб-продукт – сайт чи сторінка, яку можна переглянути у браузері, протестувати, відкрити для доступу іншим користувачам. Це може бути навчальний сайт для учнів молодших класів, особисте портфоліо, сайт-візитка для молодіжного об'єднання, цифровий шкільний журнал чи навіть інтернет-магазин із базовою функціональністю. Учні бачать свій прогрес у вигляді конкретних змін на екрані, що підвищує мотивацію та задоволення від процесу навчання.

Проектна діяльність часто виходить за межі суто інформатики. Створюючи веб-сайт, учні можуть залучати знання з дизайну, малювання, мовознавства (при підготовці текстового наповнення), маркетингу (для уявлення цільової аудиторії сайту), навіть історії чи географії (якщо тематика сайту пов'язана з культурною чи історичною спадщиною, географічними об'єктами тощо). Така інтеграція робить навчання веб-технологій органічною частиною освітнього процесу, а не ізольованою дисципліною.

Реалізація проекту вимагає попереднього планування: визначення цілей, складання сценаріїв, структурування контенту, продумування навігації та дизайну. Учні вчать системно мислити, виокремлювати ключові елементи, розуміти взаємозв'язки між компонентами веб-сайту, оцінювати їхній вплив на зручність використання.

Більш складні веб-проекти доцільно виконувати у групах: одні учні відповідають за верстку, інші за стилізацію, ще інші – за інтерактивні функції чи наповнення контенту. Такий підхід привчає до командної взаємодії, виробляє вміння обговорювати ідеї, домовлятися, допомагати одне одному. Учні отримують досвід колективної відповідальності за кінцевий продукт, що надзвичайно важливо в реальних професійних умовах.

Проектна діяльність заохочує учнів до експериментів, пошуку нестандартних рішень. Навіть базові знання HTML і CSS дозволяють проявити творчість у дизайні та структурі сторінок, а додавання JavaScript стимулює розробку інтерактивних елементів. Учні можуть самостійно знаходити корисні

фреймворки, бібліотеки, інструменти, підбирати колірні схеми, шрифти, форми подання інформації. У процесі пошуку вони вдосконалюють свої навички інформаційної грамотності, вчаться критично оцінювати знайдені дані.

Використання проектної методики на уроках інформатики в школі дозволяє досягти таких дидактичних результатів:

- Підвищення мотивації учнів:
- Формування професійних навичок:
- Розвиток відповідальності і самоконтролю:

Реальний результат діяльності збільшує зацікавленість школярів. Вони розуміють, що створюють щось корисне і видиме, а не просто механічно виконують вправи. [23]

Проектний підхід готує учнів до майбутньої роботи у сфері ІТ, де більшість завдань вирішуються у форматі проектів. Учні навчаються керувати власним часом, дотримуватися дедлайнів, самостійно шукати ресурси та вирішувати проблеми.

### **2.3.2. Кейс-методи у навчанні веб програмуванню**

Кейс-методи доповнюють проектну методику, оскільки вони фокусуються на розв'язанні конкретних проблемних ситуацій, близьких до реальної практики веб-розробника. На відміну від проектів, що часто мають довгостроковий характер і спрямовані на створення цілісного продукту, кейси можуть бути відносно короткостроковими завданнями, зосередженими на певних проблемах.

Наприклад, кейс може передбачати: оптимізацію швидкості завантаження сторінки через використання кешування, мінімізації скриптів, стиснення зображень; усунення проблем у верстці, через які сайт неправильно відображається у певних браузерах; адаптацію дизайну під мобільні пристрої, що вимагатиме застосування медіа-запитів та перевірки відображення сайту на різних екранах; відлагодження JavaScript-коду, який відповідає за інтерактивні елементи, наприклад, випадаючі меню, слайдери зображень, галереї, валідацію форм.



Кейс – це опис певної ситуації чи проблеми, яку потрібно вирішити, використовуючи наявні знання та навички. Такий підхід допомагає учням побачити, як теорія застосовується на практиці, та відчувати себе на місці реального веб-розробника.

Розглядаючи кейс, учні повинні проаналізувати умови, розібратися в суті проблеми, оцінити різні варіанти рішень, вибрати оптимальний. Це розвиває вміння мислити критично, логічно обґрунтовувати свої кроки, перевіряти гіпотези.

Кейс-методи дозволяють учням навчитися швидко реагувати на зміни умов, розуміти, що у реальній роботі ніколи не буває абсолютно ідеальних сценаріїв, а технічні проблеми чи несподівані нюанси – це норма. Учні вчаться не розгублюватися, а шукати креативні рішення, експериментувати з інструментами та методами.

Часто кейс-методи передбачають публічне обговорення результатів. Учні презентують своє рішення, пояснюють вибір, захищають його перед іншими. Це допомагає розвивати комунікаційні навички, уміння працювати з критикою, відстоювати власну позицію.

Таким чином, особливостями кейс-технологій порівняно з іншими методами навчання є:

- Конкретність і практична спрямованість
- Критичне мислення й аналітичні здібності
- Гнучкість та адаптивність
- Розвиток навичок аргументації і презентації рішень

Перевагами застосування кейс-методів є максимальне наближення до реальності, різноманітність ситуацій, формування навичок проблемного навчання.

Учні розуміють, з якими викликами може стикатися веб-розробник у професійній діяльності. Кейсова робота може охоплювати технічні, дизайнерські, користувацькі проблеми, що робить навчання багатограним.

Учні вчатья виявляти корінь проблеми, а не лише усувати поверхневі недоліки.  
[23]

### **2.3.3. Інтеграція проектної методики та кейс-методів у навчальний процес**

Поєднання проектної методики та кейс-методів забезпечує комплексний підхід до вивчення веб-технологій:

#### 1. Початковий етап навчання.

Учні знайомляться з основами HTML та CSS, вчатья створювати прості сторінки, застосовувати стилі. На цьому етапі можна використовувати кейси для відлагодження базових помилок у кодї, аналізу структури сторінки. Це допоможе учням зрозуміти, наскільки важливо дотримуватися стандартів веб-розробки.

#### 2. Базовий етап (впровадження JavaScript, адаптивний дизайн).

Коли учні опанують основи, настав час переходити до динамічних елементів, інтерактивності та адаптивності. Проектна методика стане у пригодї, коли учні розробляють сайт із можливістю перемикання теми оформлення, створюють галереї зображень, форми з валідацією даних, меню, що розкриваються при натисканні. Кейс-методи допоможуть розглянути конкретні проблеми: наприклад, чому сайт коректно працює на ПК, але «ламається» на мобільному, і як це виправити.

#### 3. Просунутий етап (інтеграція сторонніх бібліотек, оптимізація, юзабіліті)

На старших етапах учні можуть навчитися використовувати CSS-фреймворки, підключати бібліотеки JavaScript, опанувувати інструменти автоматизації. Проектні завдання можуть охоплювати розробку тематичних портфоліо чи невеликих веб-додатків. Кейс-методи дозволять опрацювати питання оптимізації продуктивності, безпеки, доступності. Наприклад, кейс може бути присвячений поліпшенню навігації для людей із порушеннями зору, що розширить горизонти учнівського мислення.

#### 4. Підсумкові проекти та комплексні кейси:

Наприкінці курсу можна організувати масштабний проект, у якому учні застосовують усі здобуті знання. Одночасно їм пропонуються кейси, які моделюють реальні виробничі ситуації: «Замовник незадоволений швидкістю сайту», «Необхідно забезпечити крос-браузерну сумісність», «Потрібно додати функцію пошуку продуктів у каталозі». Такий інтегрований підхід сприятиме формуванню у учнів універсальних навичок проблемного підходу.

Окрім проектної методики та кейс-методів, ефективним інструментом є індивідуальні завдання. Вони дозволяють врахувати особисті особливості та інтереси кожного учня. Наприклад, один учень може отримати завдання створити сайт-візитку про своє хобі (фотографія, музика, спорт), інший – адаптувати макет для мобільних пристроїв, третій – реалізувати нескладну гру на JavaScript. Такий підхід дозволяє учням сфокусуватися на тих аспектах, які їх найбільше цікавлять, розвиваючи як технічні, так і творчі здібності.

Індивідуальні завдання доповнюють проектну роботу та кейси, оскільки дають можливість учням:

- Закріпити знання в практичних умовах.
- Працювати у власному темпі, приділяючи більше уваги складним для них темам.
- Виявляти і реалізовувати власні ідеї, не обмежуючись рамками групового проекту.

У процесі застосування проектної методики та кейс-методів доцільно використовувати сучасні освітні інструменти:

- Онлайн-редактори коду (CodePen, JSFiddle, Replit): Дозволяють експериментувати з HTML, CSS і JavaScript у реальному часі, миттєво бачити результати, обмінюватися кодом з однокласниками та вчителем.
- Системи контролю версій (Git, GitHub): За допомогою Git учні можуть навчитися командній розробці, відслідковуванню змін у коді, вирішенню конфліктів злиття.
- Фреймворки та бібліотеки (Bootstrap, Tailwind, jQuery): Дозволяють учням швидко створювати адаптивні та стилізовані сторінки, розуміти, як

працюють популярні інструменти, застосовувані у професійній веб-розробці.

- Мультимедійні матеріали та онлайн-курси: Доступ до відеоуроків, інтерактивних посібників, воркшопів допомагає учням самостійно поглиблювати знання, знаходити нові рішення, навчатися у зручний для них час.

#### **2.3.4. Роль учителя у впровадженні проектної методики та кейс-методів**

Учитель у цій системі виступає більше як фасилітатор, ментор, консультант, а не просто джерело інформації. Він:

- Допомагає учням обрати тему проекту, сформулювати завдання, визначити терміни.
- Пропонує кейси, що відповідають рівню підготовки класу, поступово ускладнюючи завдання.
- Надає методичну підтримку, радить інструменти, ресурси, найкращі практики.
- Сприяє взаємодії між учнями, організовує дискусії, «мозкові штурми».
- Здійснює оцінювання, враховуючи як технічну якість результатів, так і креативність, логіку рішень, вміння працювати в команді.

Оцінювання можна організувати за кількома критеріями: технічна коректність коду, наявність адаптивності, швидкість завантаження, зручність навігації, оригінальність дизайну, здатність учня презентувати свій продукт, аргументувати вибір технологій. Такий підхід робить оцінювання об'єктивним і стимулює учнів працювати над усіма аспектами проекту чи кейсу.

## **Очікувані результати впровадження проектної методики та кейс-методів**

Поєднання проектної методики та кейс-методів дозволяє сформувати у учнів такі якості:

- Глибоке розуміння веб-технологій: Замість поверхневого ознайомлення учні справді занурюються в матеріал, оскільки постійно застосовують його на практиці.
- Комплексне мислення: Учні вчать бачити веб-сторінку не як суму тегів чи стилів, а як систему, у якій треба враховувати дизайн, функціональність, зручність для користувача, продуктивність.
- Критичний аналіз і аргументація рішень: Робота над кейсами вчить шукати оптимальні рішення, оцінюючи їхні переваги та недоліки, логічно обґрунтовувати вибір певного підходу.
- Творчість і креативність: Проектна діяльність заохочує нестандартне мислення, пошук оригінальних ідей, вихід за рамки шаблонних рішень.
- Комунікація та співпраця: Командні проекти та групове обговорення кейсів сприяють розвитку комунікативних навичок, уміння слухати інших та працювати разом.
- Підготовка до майбутньої професійної діяльності: Учні, які оволоділи основами веб-розробки через проекти й кейси, краще підготовлені до реалій ІТ-галузі, де необхідно швидко реагувати на проблеми, постійно вчитися новому, ефективно працювати у команді.

Використання проектної методики та кейс-методів у викладанні модуля «Веб-технології» є інноваційним кроком, що дозволяє перетворити навчальний процес на захопливу, осмислену та продуктивну діяльність. Такий підхід стимулює учнів до активної роботи, розвиває у них самостійність, креативність, відповідальність і готовність до викликів сучасного інформаційного світу. За допомогою проектної методики та кейс-методів можна досягти глибшого засвоєння теорії, впровадити практичні навички та сформувати ключові компетентності, необхідні для успішного майбутнього. Це інвестиція в освіту,

яка допоможе виховати покоління учнів, здатних не лише споживати технології, але й творити їх, обирати найкращі рішення та критично оцінювати результати своєї діяльності у цифровій епосі.

#### **2.4. Планування, організація освітнього процесу, приклади завдань і проектів**

Ефективне викладання модуля «Веб-технології» значною мірою залежить від структури кожного заняття, яка має забезпечувати гармонійне поєднання теоретичної та практичної частин. Фрагменти уроків розробляються з урахуванням вікових особливостей учнів, рівня їхньої підготовки та завдань, які ставляться перед ними на кожному етапі вивчення модуля. Планування уроку має чітко визначені цілі, які спрямовані на поступове освоєння матеріалу, закріплення знань та їхнє застосування через інтерактивні завдання і проекти.

Наприклад, перше заняття з мови HTML, що є основою веб-розробки, може бути зосереджене на темі «Знайомство з HTML і створення першої веб-сторінки». Цей урок орієнтований на формування початкових навичок роботи з HTML та розуміння структури веб-документа. Заняття починається з короткої презентації, яка демонструє учням реальні приклади веб-сторінок та пояснює, як їх можна створити за допомогою HTML.

Після вступу вчитель пояснює основні елементи HTML-документа: заголовки, абзаци, списки, зображення та гіперпосилання. Для наочності використовується інтерактивна демонстрація, під час якої вчитель створює веб-сторінку прямо на уроці, поступово додаючи нові елементи та пояснюючи кожен із них. На практичному етапі учням пропонується створити свою першу веб-сторінку, яка має містити заголовок, кілька абзаців тексту, список улюблених фільмів або книг, а також зображення, які вони можуть знайти в Інтернеті чи завантажити з готового набору. Завдання завершується тим, що учні зберігають свою сторінку у форматі .html і відкривають її у браузері, що дає їм можливість одразу побачити результат своєї роботи.

Для учнів, які впоралися із завданням швидше, можна запропонувати додаткові задачі, наприклад, створити таблицю, що містить розклад їхніх занять, або додати кнопки для переходу на інші сторінки, які в майбутньому вони зможуть заповнити. Такий підхід дозволяє адаптувати урок під різні рівні підготовки учнів, забезпечуючи індивідуальний підхід до кожного.

Уроки, присвячені CSS, мають на меті навчити учнів стилізувати веб-сторінки, роблячи їх більш привабливими та зручними для користувачів. Наприклад, урок «Основи CSS: кольори, шрифти та вирівнювання тексту» починається з демонстрації базових властивостей CSS, таких як `color`, `font-family`, `font-size`, `text-align`. Учням показують, як ці властивості впливають на вигляд тексту на сторінці. Теоретичний блок завершується поясненням способів підключення CSS до HTML: через внутрішні стилі, зовнішній файл стилів і інлайн-стилі.

Практична частина уроку включає завдання на застосування CSS до створеної раніше HTML-сторінки. Наприклад, учням пропонується змінити кольори заголовків і тексту, вирівняти абзаци по центру, застосувати декоративний шрифт для заголовків і збільшити розмір тексту для окремих абзацив. Завдання можна ускладнити, запропонувавши учням створити рамки навколо елементів або додати фоновий колір для сторінки.

Наступні уроки CSS орієнтовані на роботу зі складнішими властивостями. Наприклад, урок «Макети сторінок за допомогою CSS» знайомить учнів із властивостями для розташування елементів, такими як `position`, `display`, `flexbox`. На практиці учням пропонується створити макет веб-сторінки, що містить заголовок, основний текстовий блок і бічну панель. Вони вчаться розташовувати елементи у вигляді гнучких сіток, що дозволяє створювати більш професійні дизайни.

Особливо цікаві уроки, присвячені JavaScript, оскільки вони додають веб-сторінкам інтерактивності. Наприклад, на уроці з теми «Обробка подій у JavaScript» учням пояснюють, як створювати сценарії для кнопок, які виконують різні дії. Під час практичного завдання учні створюють сторінку з

двома кнопками: одна змінює колір фону сторінки, а друга додає новий елемент до списку. Це дозволяє їм побачити, як JavaScript взаємодіє з HTML і CSS, і зрозуміти, як додавати динамічні функції.

Для закріплення знань із JavaScript учням можна запропонувати створити міні-додаток, наприклад, калькулятор для обчислення базових арифметичних операцій. Такий проект не лише закріплює знання про змінні, функції та обробники подій, але й стимулює розвиток алгоритмічного мислення, оскільки учні мають продумати логіку роботи калькулятора.

Уроки з адаптивного дизайну є важливою частиною модуля, оскільки вони знайомлять учнів із сучасними підходами до створення сайтів. Наприклад, урок «Використання медіа-запитів для адаптивного дизайну» починається з пояснення, як працюють медіа-запити CSS, та демонстрації прикладів сайтів, які адаптуються до різних розмірів екрану. На практиці учням пропонується створити веб-сторінку, яка має два різних макети: один для настільних комп'ютерів, а другий для смартфонів. Це завдання дозволяє учням зрозуміти, як зробити сайт доступним для широкого кола користувачів.

Проектна діяльність є інтегруючим компонентом модуля «Веб-технології». Наприклад, підсумковий проект може включати створення багатосторінкового сайту, присвяченого обраній темі, наприклад «Екологія» або «Історичні пам'ятки моєї країни». Учні працюють у групах, кожен відповідає за свою частину проекту: дизайн, програмування або контент. У процесі роботи вони використовують усі знання, здобуті на попередніх уроках, що дозволяє їм закріпити матеріал і побачити практичне застосування своїх умінь.

Кожен урок завершується етапом рефлексії, під час якого учні обговорюють свої успіхи, труднощі та способи їх подолання. Це дозволяє не лише оцінити рівень засвоєння матеріалу, але й стимулює учнів до подальшого навчання, оскільки вони бачать результати своєї роботи.

На більш просунутому етапі вивчення модуля «Веб-технології» увага учнів зосереджується на інтеграції знань, створенні більш складних веб-ресурсів і розробці взаємодії між різними компонентами веб-додатків. Для



досягнення цих цілей використовуються комбіновані уроки, на яких одночасно застосовуються HTML, CSS та JavaScript.

Одним із прикладів такого заняття є урок «Створення інтерактивної галереї зображень». Заняття починається з пояснення концепції інтерактивних елементів на веб-сторінках і їхньої важливості для покращення користувацького досвіду. Під час теоретичної частини вчитель демонструє, як використовувати JavaScript для динамічного додавання елементів на сторінку та їхньої заміни. Практична частина уроку передбачає створення веб-сторінки, на якій є кілька ескізів зображень, що змінюються на повнорозмірні варіанти при натисканні.

Завдання можна ускладнити, додавши функцію автоматичної прокрутки зображень або кнопки для переходу між ними. Для цього учні вивчають роботу з таймерами в JavaScript (`setInterval` та `setTimeout`), що дозволяє реалізувати автозміну картинок із заданими інтервалами. Такі уроки стимулюють інтерес до програмування, дозволяючи учням побачити результати своєї роботи одразу.

Ще одним прикладом інтегрованого уроку є заняття на тему «Створення адаптивного сайту портфоліо». Цей урок спрямований на підсумкове використання всіх знань, отриманих учнями під час вивчення модуля. На теоретичному етапі вчитель пояснює ключові принципи адаптивного дизайну, такі як гнучкі сітки, відносні одиниці вимірювання та медіа-запити. Для демонстрації вчитель використовує приклади сайтів, які коректно відображаються на пристроях із різними розмірами екранів.

На практичному етапі учням пропонується розробити багатосторінковий сайт, який містить такі розділи, як «Про мене», «Проекти» та «Контакти». Важливо, щоб сайт мав три рівні адаптації: для настільних комп'ютерів, планшетів і смартфонів. Учні самостійно визначають, як саме змінюватиметься дизайн для кожного з пристроїв. Наприклад, вони можуть використовувати гнучкі сітки для забезпечення правильної структури сторінки, а також змінювати розміри шрифтів і кнопок залежно від ширини екрану.

Практичні завдання такого типу дозволяють учням інтегрувати знання про HTML, CSS та JavaScript, створюючи повноцінні веб-продукти. Учні

працюють як самостійно, так і в групах, що сприяє формуванню навичок співпраці та обміну знаннями.

Важливим компонентом викладання модуля є використання завдань, які мають відношення до реальних викликів у сфері веб-розробки. Наприклад, кейс-завдання «Оптимізація сайту» включає аналіз створеного веб-ресурсу на предмет швидкості завантаження та виявлення проблем. Учні вчаться оптимізувати зображення, зменшувати розмір файлів CSS і JavaScript, а також використовувати техніки кешування.

Ще одним кейсом може бути завдання «Додати функцію пошуку на веб-сторінку». Учням пояснюють принцип роботи поля пошуку, яке дозволяє користувачам швидко знаходити потрібну інформацію на сайті. Практична частина включає створення простого сценарію на JavaScript, який шукає ключові слова в тексті сторінки та підсвічує знайдені збіги. Це завдання дозволяє учням зрозуміти, як реалізуються базові алгоритми пошуку та їхня інтеграція у веб-ресурси.

Окремої уваги заслуговують уроки, присвячені реалізації інтерактивних функцій, які використовуються в сучасних додатках. Наприклад, урок «Реалізація інтерактивного списку справ» знайомить учнів із принципами динамічного додавання, редагування та видалення елементів на сторінці. На практиці учні створюють веб-додаток, який дозволяє додавати завдання до списку, позначати їх як виконані та видаляти.

Учням також пропонуються завдання на роботу з локальним збереженням даних, наприклад, із використанням LocalStorage. У ході уроку вони вчаться зберігати стан веб-додатка навіть після оновлення сторінки, що є важливим аспектом сучасних веб-ресурсів.

Проектна діяльність на заключних етапах модуля спрямована на створення комплексних продуктів, які можуть бути представлені на виставках або конкурсах. Наприклад, учням пропонують створити сайт для реального або уявного клієнта. Це завдання включає всі етапи веб-розробки: від визначення вимог клієнта та планування структури сайту до його створення, тестування й

презентації. Учні мають змогу показати свої роботи батькам, учителям та іншим учням, що підвищує їхню мотивацію.

Крім того, учням можна запропонувати створити інноваційний проект, наприклад, інтерактивну карту шкільної бібліотеки, яка відображає доступні книги та їхні місця на полицях. Такий проект не лише закріплює технічні знання, але й демонструє, як веб-технології можуть бути використані для вирішення реальних проблем.

Завершальні уроки модуля можуть включати обговорення найкращих практик веб-розробки, таких як дотримання стандартів доступності, забезпечення безпеки даних та оптимізація роботи сайтів для різних браузерів. Учні розглядають реальні приклади успішних веб-ресурсів, аналізують їхні сильні та слабкі сторони, а також обговорюють можливості для подальшого навчання та кар'єри у сфері інформаційних технологій.

Під час рефлексії учні оцінюють свої досягнення, обговорюють труднощі, які вони подолали, та пропонують ідеї для покращення своїх робіт. Це стимулює їх до подальшого вдосконалення знань і вмінь, а також формує впевненість у власних силах.

Для більш глибокого вивчення веб-технологій важливо пропонувати учням завдання, які не лише закріплюють основні технічні поняття, але й дають змогу проявити креативність. Наприклад, урок із теми «Анімації в CSS» допомагає учням зрозуміти, як створювати динамічні ефекти на веб-сторінках, використовуючи ключові кадри та переходи. Теоретична частина уроку включає пояснення властивостей `transition` і `animation`, демонстрацію прикладів їх застосування.

Практичне завдання на цьому уроці може полягати в тому, щоб створити веб-сторінку з анімованими елементами. Наприклад, учні можуть додати ефекти плавного зникнення чи появи тексту при наведенні мишки, або анімацію кнопок, які змінюють колір чи форму. Учні також можуть створити просту анімовану картинку за допомогою `@keyframes`, наприклад, логотип, який

обертається чи змінює розмір. Завдання завершується презентацією робіт, де учні пояснюють, які анімаційні ефекти вони використали і чому саме ці.

Іншим цікавим заняттям є урок «Використання API для інтеграції зовнішніх даних у веб-додатки». Це завдання актуальне для учнів, які хочуть дізнатися більше про взаємодію між веб-ресурсами. На теоретичному етапі учні знайомляться з поняттям API (інтерфейсу програмування додатків) і його роллю у веб-розробці. Учням демонструють, як використовувати JavaScript для надсилання запитів до зовнішніх серверів і отримання даних у форматі JSON.

Практична частина уроку може включати завдання створити веб-сторінку, яка показує поточну погоду в заданому місті, використовуючи безкоштовне API, таке як OpenWeatherMap. Учні пишуть код, який надсилає запит до API, обробляє отримані дані та відображає їх на сторінці. Завдання можна ускладнити, додавши функцію вибору міста або показу прогнозу погоди на кілька днів. Це завдання розвиває аналітичне мислення учнів і дає їм змогу зрозуміти, як веб-додатки взаємодіють із зовнішніми джерелами даних.

Цікавим підходом є створення інтерактивних навчальних веб-додатків. Наприклад, урок «Розробка вікторини на JavaScript» може бути присвячений створенню додатка, який пропонує користувачеві запитання та оцінює правильність відповідей. Учні вивчають, як працювати з масивами для зберігання запитань і відповідей, а також як реалізувати логіку перевірки відповідей та підрахунку балів. Практичне завдання передбачає створення веб-сторінки, яка показує запитання, дає можливість вибрати правильну відповідь і підраховує кількість балів наприкінці вікторини. Завдання можна ускладнити, додавши таймер, який обмежує час на відповідь.

Окремої уваги заслуговують уроки, присвячені темі «Розробка односторінкових додатків (SPA)». Це просунутий рівень навчання, який дає учням змогу зрозуміти, як створюються сучасні динамічні веб-додатки. Теоретична частина включає пояснення концепції SPA, де контент оновлюється без перезавантаження сторінки, завдяки використанню JavaScript і технологій, таких як Fetch API. На практиці учні створюють простий односторінковий

додаток, наприклад, список справ, де користувач може додавати, видаляти та редагувати записи.

Для уроків з адаптивного дизайну можна використати завдання на розробку веб-сайту з інтерактивною картою. Наприклад, учням пропонують створити сторінку, на якій відображається карта з можливістю змінювати масштаб і переглядати інформацію про вибрані місця. Для цього учні використовують Google Maps API або Leaflet.js. Такі завдання дають учням змогу застосувати знання з HTML, CSS і JavaScript у контексті створення сучасних веб-додатків.

Проектна діяльність може включати завдання на створення реальних або вигаданих комерційних сайтів. Наприклад, проект «Сайт для кафе» включає такі етапи: розробка структури сайту, створення макетів сторінок, додавання меню страв із цінами, інтерактивної форми для бронювання столиків та інтеграції карти для показу місця розташування. Учні вчаться продумувати користувацький досвід і реалізовувати функції, які є важливими для реального бізнесу.

Ще один приклад проекту — «Особистий фінансовий трекер». Це веб-додаток, який дозволяє користувачам вводити свої доходи та витрати, відображаючи підсумковий баланс. Учні вчаться працювати з формами, обробкою даних на JavaScript, а також створюють графіки або діаграми для візуалізації фінансової інформації.

Для старших учнів або тих, хто проявляє особливий інтерес до програмування, можна запропонувати завдання з використанням серверних технологій, наприклад, Node.js. Урок «Основи роботи з сервером» може включати створення простого серверного додатка, який обробляє запити від клієнтів і повертає дані у форматі JSON. Практичне завдання може передбачати розробку API для зберігання та отримання даних про користувачів.

## **2.5 Апробація методики викладання модуля «Веб-технології» та результати дослідження**

Апробація розробленої методики викладання модуля «Веб-технології» є надзвичайно важливим етапом, який дозволяє не тільки перевірити її ефективність у реальних умовах шкільного навчання, а й виявити сильні та слабкі сторони, оцінити динаміку засвоєння матеріалу учнями, їхню мотивацію, розвиток ключових компетентностей і готовність до творчого застосування здобутих знань. Такий підхід спрямований на формування свідомих і компетентних користувачів інформаційних технологій, які здатні впевнено орієнтуватися у цифровому середовищі, створювати власні веб-продукти та розуміти принципи роботи Інтернету.

### **Мета, завдання та умови апробації**

Основною метою дослідження було визначити, наскільки розроблена методика сприяє глибокому опануванню учнями базових веб-технологій (HTML, CSS, JavaScript, адаптивний дизайн), здатності застосовувати їх на практиці, а також формує такі важливі компетентності як критичне мислення, креативність, вміння працювати в команді, здатність до самостійного розв'язання нетипових задач і висока навчальна мотивація. Серед завдань дослідження було:

1. Виявити початковий рівень знань учнів у сфері веб-технологій, інформаційних систем та основ програмування.
2. Перевірити ефективність застосування поєднання різних форм і методів навчання: проектної методики, кейс-методів, індивідуальних завдань, інтерактивних занять.
3. Оцінити, наскільки запропоновані завдання та проекти стимулюють розвиток ключових компетентностей: комунікативних навичок, вміння планувати діяльність, аналізувати проблеми та творчо їх розв'язувати.
4. Виявити труднощі, з якими стикаються учні під час освоєння веб-технологій, та визначити шляхи їх подолання.

5. Одержати зворотний зв'язок від учнів щодо цікавості та корисності модуля, а також їхньої готовності продовжувати вивчення цієї сфери.

Апробація проводилася в одній із загальноосвітніх шкіл, де експериментальну групу склали 14 учнів 10-го класу. Вік учнів (15–16 років) та їхня загальна ІТ-підготовка були типовими для старшокласників, які мають базові знання з інформатики, але здебільшого не знайомі з веб-розробкою. Навчальні заняття відбувалися у комп'ютерному класі з доступом до Інтернету та використанням сучасних технічних засобів, що дало змогу забезпечити комфортні умови для практичної діяльності. Наявність швидкого інтернет-з'єднання дозволила учням працювати з онлайн-редакторами коду (CodePen, JSFiddle), ознайомлюватися з довідковими матеріалами (W3Schools, MDN Web Docs) та переглядати мультимедійні навчальні ресурси.

### **Попередня діагностика і вибір стратегії навчання**

Перед початком занять було проведено вступне діагностичне тестування, яке складалося з теоретичної та практичної частин. Теоретичні питання стосувалися базових понять ІТ-сфери (що таке HTML, як відрізнити CSS від HTML, які бувають типи файлів), а практичні завдання передбачали створення простого текстового файлу в редакторі та мінімальну взаємодію з файлами на комп'ютері. Результати діагностики засвідчили, що більшість учнів мають фрагментарні знання про структуру веб-сторінки та поняття «гіпертекст», а безпосередніх навичок роботи з HTML або CSS у них фактично немає. Це дало підстави побудувати програму навчання з поступовим ускладненням: від базових тегів HTML до основ стилізації, далі – до інтерактивності з JavaScript та адаптивного дизайну.

### **Організація навчального процесу і зміст уроків**

Навчальний курс складався зі шести уроків (по 2 академічні години кожний), протягом яких учні проходили шлях від ознайомлення з найпростішими HTML-тегами до створення підсумкових проєктів. Кожний урок мав чітко сформульовану мету, поєднував теоретичний блок з

практичними завданнями, а також містив елементи проектної діяльності та кейс-методів.

- **Перший урок:** Ознайомлення з HTML. Учитель пояснив структуру веб-сторінки, принципи роботи тегів `<html>`, `<head>`, `<body>`, роль заголовків, абзаців, списків, гіперпосилань і зображень. Практичне завдання полягало у створенні простої веб-сторінки з описом улюбленого фільму або книги. Учні працювали індивідуально, отримуючи консультації, які допомагали подолати початкові труднощі. [Додаток 1]
- **Другий урок:** Основи CSS. Учитель продемонстрував, як підключати CSS, які бувають селектори і властивості, як змінювати кольори, шрифти, відступи та розташування елементів. Кейс-метод у цьому уроці полягав у тому, що учням було запропоновано поліпшити зовнішній вигляд сторінки з попереднього уроку, перетворивши її з «чорнової версії» на більш привабливий веб-документ. Таким чином, вони могли одразу побачити різницю між простим HTML та сторінкою, стилізованою за допомогою CSS. [Додаток 2]
- **Третій урок:** Вступ до JavaScript. Учні вперше зіткнулися з концепціями змінних, операторів, функцій та подій. Учитель пояснив, як за допомогою JavaScript можна змінювати вміст сторінки на льоту, реагувати на дії користувача (клік, наведення курсору, введення даних). Практичне завдання: створити сторінку з кнопками, які змінюють колір фону або тексту, а також проста форма для введення даних, що перевіряє правильність заповнення полів. Деякі учні спершу відчували складність розуміння логіки програмування, але за допомогою додаткових пояснень та прикладів поступово оволоділи початковими навичками.
- **Четвертий урок:** Адаптивний дизайн і медіа-запити. Учитель продемонстрував, як створити сторінку, що коректно відображається на різних пристроях. Учні дослідили, як змінювати макет за допомогою `@media` та відповідних CSS-властивостей, протестували свої сторінки в емуляторах мобільних пристроїв. Це допомогло їм усвідомити практичну



цінність адаптивності, оскільки сучасний веб-глобально використовується на смартфонах, планшетах і моніторах різних розмірів.

- **П'ятий та шостий уроки:** Підсумкові проекти. На цих заняттях учні об'єдналися у три групи по 4–5 осіб (у одній групі було 5, в інших – по 4), та мали створити міні-сайт на тему, яку вони обирали самостійно. Одна з груп взяла тему екології та створила сайт з інформацією про проблеми довкілля, використовуючи інтерактивні елементи для перегляду статистики. Інша – сайт про шкільні гуртки, де можна було ознайомитися з розкладом занять, переглянути фотоальбом, залишити заявку на вступ. Третя група вирішила створити невеличкий портал з порадами для підготовки до іспитів, інтегруючи форму зворотного зв'язку та невеликий інтерактивний тест.

Під час роботи над проектами учні активно використовували отримані знання: структурували контент за допомогою HTML, оформлювали його CSS-стилями, створювали інтерактивні елементи на JavaScript, забезпечували адаптивність макету. Учитель виступав у ролі ментора, консультував щодо технічних питань, допомагав долати складнощі організаційного та технологічного характеру.

### **Результати апробації і методи оцінювання**

Наприкінці курсу було проведено підсумкове тестування, яке включало теоретичні питання (визначення понять, призначення певних тегів, властивостей і функцій) та практичні завдання (виправлення помилок у коді, додавання певного елемента до сторінки, не порушуючи існуючу структуру). Переважна більшість учнів показала суттєвий прогрес порівняно з початковим рівнем: вони впевненіше застосовували базові теги HTML, вміли підключати та налаштовувати CSS, розуміли принципи адаптивного дизайну і могли написати прості скрипти на JavaScript.

Аналіз результатів проектних робіт виявив, що учні навчилися самостійно орієнтуватися в коді, застосовувати різні технології на практиці та вирішувати завдання, які не були детально розібрані на уроках, але логічно впливали з

освоєного матеріалу. Проекти були доволі різноманітними за структурою та дизайном, і хоча не всі елементи виглядали ідеально з технічної або естетичної точки зору, помітний був самостійний пошук рішень, креативний підхід до оформлення, намагання покращити юзабіліті сайту.

Рефлексія учнів, проведена у формі анкетування та відкритого обговорення, показала, що більшість старшокласників отримала позитивні враження від курсу. Вони оцінили практичну спрямованість занять, можливість власноруч створювати веб-сторінки та цілі міні-сайти. Дехто відзначав, що робота в групах навчила їх домовлятися, розподіляти обов'язки, узгоджувати дизайнерські та технічні рішення. Частина учнів зізналася, що найбільше складнощів викликала логіка JavaScript, однак поступове введення прикладів, індивідуальні консультації та робота над власним кодом допомогли подолати первинні труднощі.

### **Аналіз труднощів і шляхи вдосконалення**

Апробація виявила декілька аспектів, які потребують уваги та вдосконалення:

#### **1. Складність JavaScript для початківців:**

Порівняно з HTML та CSS, логіка написання скриптів здалася учням складнішою. Надалі можна запропонувати більше покрокових прикладів, інтерактивних тренажерів, простих завдань на початку (наприклад, зміна тексту за кліком), поступово ускладнюючи задачі.

#### **2. Розподіл ролей у групових проектах:**

Деякі учні відчували труднощі з тим, щоб визначити, хто за що відповідає у команді. Рішенням може бути надання учням попередньо узгоджених сценаріїв організації проектної діяльності, поради щодо ролей (дизайнер, верстальник, програміст, контент-редактор), а також проміжний контроль прогресу, щоб група рівномірно розподіляла роботу.

#### **3. Брак часу на деталізацію деяких аспектів:**

В умовах шкільної програми час обмежений, тому не завжди можна детально розглянути всі нюанси. Можливим виходом стане створення

супровідних навчальних матеріалів (конспектів, відеоуроків, інструкцій), які учні можуть використовувати самостійно, а також поширення позакласних активностей (факультативи, гуртки) для поглибленого вивчення теми.

#### **4. Врахування різних рівнів підготовки та інтересів:**

Навіть у одній групі учнів рівень підготовки та інтерес до веб-технологій може суттєво різнитися. Варто гнучко підходити до індивідуальних завдань, пропонувати більш складні чи спрощені варіанти, заохочувати сильніших учнів до наставництва над тими, хто відстає.

#### **Висновки та перспективи**

Апробація методики викладання модуля «Веб-технології» засвідчила її ефективність у підвищенні рівня технічної грамотності учнів, розвитку їхніх творчих та комунікативних навичок, формуванні зацікавленості у тематиці веб-розробки. Учні навчилися не просто сприймати інформацію, а й створювати власні продукти, експериментувати з кодом, шукати відповіді у довідкових ресурсах, самостійно виправляти помилки та оптимізувати сторінки.

Отриманий досвід дозволяє стверджувати, що інтеграція проектної методики, кейс-методів, індивідуальних завдань і практичних прикладів в єдину навчальну систему є виправданою і перспективною. Така стратегія забезпечує більш глибоке засвоєння матеріалу, розкриває потенціал учнів, робить процес навчання цікавим, мотивуючим і близьким до реальних умов сучасного ІТ-середовища.

У подальшому можна розширити зміст модуля, додати елементи роботи з простими фреймворками чи інструментами автоматизації, запровадити міні-хакатони для вирішення кейсів у обмежений час. Такі заходи ще більше підвищать мотивацію, допоможуть учням відчувати себе справжніми розробниками та стимулюватимуть подальший саморозвиток у сфері інформаційних технологій.

Таким чином, апробація підтвердила актуальність, доцільність і результативність обраної методики, її здатність сприяти формуванню

необхідних у сучасному цифровому суспільстві компетентностей та підготувати учнів до активної й успішної діяльності у сфері веб-технологій.

## ВИСНОВКИ

У процесі виконання даної дипломної роботи було розроблено й обґрунтовано комплексну методику викладання модуля «Веб-технології» для учнів старшої школи, що інтегрує сучасні підходи до навчання, спрямовані на формування в учнів широкого спектра ключових компетентностей. Особливість цієї методики полягає у поєднанні традиційного теоретичного матеріалу з систематичним залученням проектної діяльності, кейс-методів та індивідуальних завдань, що уможливорює високий ступінь залученості учнів у навчальний процес і дає змогу формувати у них не лише технічні знання та вміння, а й розвивати креативність, самостійність, критичне мислення й комунікативні навички.

Запропонована методика відповідає сучасним трендам цифровізації освіти та вимогам інформаційного суспільства, де знання веб-технологій набувають все більшого значення. Веб-технології сьогодні — це не просто інструменти для створення веб-сторінок чи сайтів, а ключові елементи глобальної цифрової екосистеми. Опанування основами HTML, CSS, JavaScript, адаптивного дизайну й інтерактивних підходів до розробки користувацького інтерфейсу дозволяє учням не тільки зрозуміти принципи функціонування сучасного Інтернету, а й усвідомити логіку веб-розробки, навчитися працювати з кодом, аналізувати помилки, оптимізувати сторінки та користувацький досвід. Таким чином, залучення учнів до активного вивчення веб-технологій слугує важливим кроком на шляху їхньої цифрової грамотності та підготовки до життя у високотехнологічному світі.

Апробація розробленої методики у реальному навчальному середовищі показала, що її впровадження сприяє підвищенню навчальної мотивації, активності та зацікавленості учнів. Порівняно з традиційними формами навчання, використання проектної методики дало змогу учням працювати над реальними завданнями, бачити практичну цінність опанованих знань. Проектно-орієнтований підхід стимулював учнів до пошуку самостійних

рішень, генерування власних ідей, вибору оптимальних технологічних засобів та дизайнерських рішень. Залучення кейс-методів, що передбачають розв'язання конкретних проблемних ситуацій, наближених до реальних умов професійної діяльності веб-розробника, допомогло учням зрозуміти, що програмний код — це не лише абстрактні команди, а інструмент для створення корисних, функціональних і якісних веб-ресурсів.

Окремо слід відзначити значення індивідуальних завдань, що враховують різнорівневі здібності та інтереси учнів. Такий підхід дозволив більш підготовленим старшокласникам поглиблювати свої знання та експериментувати з додатковими інструментами, фреймворками чи бібліотеками, тоді як учні з меншим досвідом мали змогу поступово засвоювати базові концепти. Водночас індивідуальні завдання стимулювали почуття успіху, впливали на підвищення самооцінки, мотивації та впевненості у власних силах.

Результати дослідження підтвердили, що така комплексна методика навчання веб-технологій сприяє формуванню важливих компетентностей:

- **Критичне мислення:** Учні навчилися аналізувати код, знаходити та виправляти помилки, порівнювати різні технічні рішення, оцінювати їхню ефективність та доцільність.
- **Креативність:** Робота над проектами, можливість самостійно обирати тематику веб-сторінок, експериментувати з дизайном, анімованими елементами та інноваційними підходами до взаємодії з користувачем стимулювала творчий потенціал.
- **Комунікаційні навички:** Виконання групових проектів і кейсів вимагало від учнів активної взаємодії, вміння аргументувати власну точку зору, слухати інших, співпрацювати для досягнення спільної мети. Це призвело до розвитку здатності до конструктивного діалогу, компромісів та підвищення якості командної діяльності.
- **Самостійність та відповідальність:** Індивідуальні завдання та проектна робота спонукали учнів планувати власну діяльність, розподіляти час, шукати інформацію, використовувати довідкові ресурси (MDN Web Docs,

W3Schools), самостійно приймати рішення щодо архітектури сторінки чи стилізації елементів. Такий підхід стимулював вироблення відповідальності за кінцевий результат.

- **Інформаційна грамотність:** Учні, працюючи з веб-технологіями, вчилися критично ставитися до інформації, знаходити потрібні ресурси, робити висновки з документації та прикладів коду, аналізувати різні підходи до верстки та програмування. Це загартувало їх у вмінні орієнтуватися у великому масиві цифрових даних.

Важливим аспектом дослідження стало виявлення труднощів, з якими стикалися учні під час засвоєння матеріалу. До них належить складність логіки програмування на JavaScript, поява питань щодо адаптивної верстки для різних типів пристроїв, проблеми з оптимізацією та швидкістю завантаження сторінок. Однак ці труднощі були подолані завдяки додатковим поясненням, демонстрації покрокових прикладів, впровадженню кейсів, спрямованих на розв'язання конкретних проблем, а також індивідуальним консультаціям. Таким чином, труднощі стали стимулом для більш глибокого занурення у тему, розуміння взаємозв'язків між різними технологічними складовими веб-розробки.

Отримані результати мають вагоме практичне значення. Їх можна використати під час вдосконалення навчальних програм з інформатики у старших класах, введення вибіркового модулю для поглибленого вивчення ІТ-технологій та при підготовці учителів. Викладачі можуть адаптувати методику під власні умови, розширювати чи скорочувати обсяг матеріалу, пропонувати інші кейси та проекти, доповнювати завдання новими інструментами (наприклад, CSS-фреймворками, бібліотеками для анімацій чи простими інструментами CI/CD для інтернет-проектів). Існує перспективність розробки подібних модулів для поглибленого навчання фреймворків і бібліотек, а також вивчення основ безпеки веб-додатків.

У широкому контексті цифровізації освіти дане дослідження узгоджується з основними документами й нормативними актами, прийнятими в Україні та на міжнародному рівні (Концепція розвитку цифрових

компетентностей в Україні, Digital Education Action Plan ЄС, рекомендації ЮНЕСКО щодо цифрової грамотності та ін.). Це підтверджує, що створення таких модулів навчання має стратегічне значення для підготовки молодого покоління до успішної діяльності в умовах динамічного цифрового середовища. Здатність швидко опановувати нові інструменти, адаптуватися до технологічних змін, працювати над проектами різної складності — це навички, які стануть ключовими перевагами майбутнього спеціаліста не лише в ІТ-сфері, а й у багатьох інших галузях.

Крім того, запропонована методика слугує прикладом інтеграції різнопланових педагогічних інструментів. Поєднання проектної діяльності (що стимулює творчість і самостійність), кейс-методів (які розвивають критичне мислення та аналітичні здібності) та індивідуальних завдань (що дозволяють врахувати диференційовані потреби учнів) забезпечує багатогранний розвиток особистості школяра. Такий підхід набагато краще відповідає викликам сучасної освіти, ніж традиційні лекційно-репродуктивні методи, оскільки він робить процес навчання гнучким, динамічним, орієнтованим на розвиток життєво важливих компетентностей.

Перспективи подальших досліджень можуть полягати у масштабнішій апробації методики, залученні більшої кількості навчальних закладів і учнів, проведенні порівняльного аналізу результатів з іншими підходами до навчання веб-технологій. Можливе також удосконалення змісту модуля, зокрема, включення сучасних технологій веб-розробки (препроцесорів, інструментів для збірки проектів, роботи з API), а також розширення тематики проектів, інтеграція питань інформаційної безпеки та етики цифрового контенту. Ще одним напрямком є розробка онлайн-вих супровідних курсів, методичних рекомендацій для вчителів, які хочуть впровадити дану методику у своїй практиці, створення форумів чи спільнот для обміну досвідом і матеріалами.

Таким чином, ця дипломна робота продемонструвала, що впровадження інноваційних методів навчання веб-технологій здатне зробити уроки інформатики не тільки пізнавальними, а й захопливими, мотивуючими та



перспективними для майбутнього професійного розвитку учнів. Розвинені у процесі навчання компетентності виходять за межі суто технічних знань і стають фундаментом для успішної діяльності у сучасному світі, де інформація, технології та комунікація відіграють ключову роль.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. **CodePen** [Електронний ресурс] – онлайн-редактор фронтенд-коду. URL: <https://codepen.io/>
2. **Codecademy** [Електронний ресурс] – інтерактивна платформа навчання програмуванню. URL: <https://www.codecademy.com/>
3. **JSFiddle** [Електронний ресурс] – інструмент для тестування та спільного використання коду. URL: <https://jsfiddle.net/>
4. **Khan Academy** [Електронний ресурс] – освітня платформа з різних галузей знань. URL: <https://www.khanacademy.org/>
5. **Replit** [Електронний ресурс] – середовище розробки в браузері. – Режим доступу: <https://replit.com/>
6. **Udemy** [Електронний ресурс] – онлайн-курси від професіоналів у різних сферах. URL: <https://www.udemy.com/>
7. **W3Schools** [Електронний ресурс] – довідник та навчальні матеріали з веб-технологій. URL: <https://www.w3schools.com/>
8. **Використання освітніх онлайн-матеріалів: сучасні підходи і технології нової української школи** / Державна установа "Український інститут розвитку освіти". URL: <https://uied.org.ua/wp-content/uploads/2023/03/metodychni-rekomendacziyi.pdf>
9. **Інформатика (профільний рівень). Підручник для 10 класу закладів загальної середньої освіти** / Автори: В. Д. Руденко, Н. В. Речич, В. О. Потієнко.
10. **Календарно-тематичне планування модулю «Веб-технології»**. Освітній проект «На Урок» для вчителів. URL: <https://naurok.com.ua/kalendarno-tematichne-planuvannya-modulyu-veb-tehnologi-122140.html>

11. **Карабін О.Й., Петрів Х.Б.** Формування основ вебдизайну в учнів старших класів закладів загальної середньої освіти. URL: [http://www.innovpedagogy.od.ua/archives/2022/53/part\\_1/9.pdf](http://www.innovpedagogy.od.ua/archives/2022/53/part_1/9.pdf)
12. **Методика навчання інформатики [Електронний ресурс]** / Львівський національний університет імені Івана Франка. URL: <https://pedagogy.lnu.edu.ua/wp-content/uploads/2016/11/Навчальний-посібник-1.pdf>
13. **Методичні рекомендації щодо викладання інформатики в основній та старшій школі [Електронний ресурс]** / Запорізький національний університет. URL: <https://moodle.znu.edu.ua/mod/resource/view.php?id=225754>
14. **Методичні рекомендації щодо викладання інформатики у 5-11 класах у 2022/2023 навчальному році [Електронний ресурс]** / Міністерство освіти і науки України. URL: [https://znavshov.com/FR/16401/13-15\\_684-686-lipen-serpen-2022-5-8.pdf](https://znavshov.com/FR/16401/13-15_684-686-lipen-serpen-2022-5-8.pdf)
15. **Млавець Ю.Ю.** Методика навчання інформатики (конспект лекцій). URL: <https://www.uzhnu.edu.ua/uk/infocentre/get/45516>
16. **Мосіюк О.О.** **WEB-ТЕХНОЛОГІЇ Частина 1. Верстка.** URL: [http://eprints.zu.edu.ua/32361/1/Web\\_ost.pdf](http://eprints.zu.edu.ua/32361/1/Web_ost.pdf)
17. **Навчальний посібник. Шкільний курс інформатики та методика його викладання.** URL: <http://dspace.pnpu.edu.ua/bitstream/123456789/3982/1/barbolina1.pdf>
18. **Речич Н.В.** Інформатика: вебтехнології (вибірковий модуль для 10–11 класів, рівень стандарту) : Папер./електрон. кн. Українна, 2020. 160 с.
19. **Ривкінд Й.Я., Лисенко Т.І., Чернікова Л.А., Шакотько В.В.** Інформатика 10-11 клас. Рівень стандарту. URL: [https://kpdі.edu.ua/biblioteka/I/Інформатика 10\(11\) клас Ривкінд Й.Я..pdf](https://kpdі.edu.ua/biblioteka/I/Інформатика 10(11) клас Ривкінд Й.Я..pdf)
20. **Собчук О., Федоренко М.** Особливості викладання веб-технологій у профільній школі / Актуальні проблеми розвитку природничих та гуманітарних наук: збірник матеріалів VIII Міжнар. наук. практ. конф. (14

- листопада 2024 р.) Луцьк: ВНУ ім. Лесі Українки, 2024. – С.150-152. URL: [https://volnu-my.sharepoint.com/personal/science-dep\\_vnu\\_edu\\_ua/\\_layouts/15/onedrive.aspx?id=/personal/science-dep\\_vnu\\_edu\\_ua/Documents/конференція%20РМВ/Збірник\\_РМВ\\_2024.pdf&parent=/personal/science-dep\\_vnu\\_edu\\_ua/Documents/конференція%20РМВ&ga=1](https://volnu-my.sharepoint.com/personal/science-dep_vnu_edu_ua/_layouts/15/onedrive.aspx?id=/personal/science-dep_vnu_edu_ua/Documents/конференція%20РМВ/Збірник_РМВ_2024.pdf&parent=/personal/science-dep_vnu_edu_ua/Documents/конференція%20РМВ&ga=1).
21. **Суцук А.** Застосування сучасних web-орієнтованих технологій в освітньому процесі. Національний університет водного господарства та природокористування. URL: <https://core.ac.uk/download/pdf/304295664.pdf>
22. Сучасні цифрові технології та інноваційні методики навчання: досвід, тенденції, перспективи. Тернопільський нац. пед. ун-т. URL: [https://lib.iitta.gov.ua/id/eprint/740555/1/Збірник\\_Тернопіль\\_2024.pdf](https://lib.iitta.gov.ua/id/eprint/740555/1/Збірник_Тернопіль_2024.pdf)
23. **Трошкі Н.В.** Робоча програма навчальної дисципліни викладання інформатики в школі. URL: <https://www.uzhnu.edu.ua/en/infocentre/get/44631>
24. **Євтеєв В. М., Семеріков С. О., Теплицький І. О.** Досвід вивчення інтерактивних Web-технологій в середній школі та педагогічному ВНЗ. URL: [https://elibrary.kdpu.edu.ua/bitstream/0564/783/1/Evteev\\_Semerikov\\_Teplitskiy.pdf](https://elibrary.kdpu.edu.ua/bitstream/0564/783/1/Evteev_Semerikov_Teplitskiy.pdf).
25. **Стовбик А. Р.** Методика навчання вибіркового модуля «Веб-технології» у шкільному курсі інформатики. URL: [https://archer.chnu.edu.ua/bitstream/handle/123456789/7094/math\\_2022\\_049.pdf?sequence=1&isAllowed=y](https://archer.chnu.edu.ua/bitstream/handle/123456789/7094/math_2022_049.pdf?sequence=1&isAllowed=y).
26. Наказ Міністерства освіти і науки України № 614 від 21.06.2010 "Про затвердження вимог до специфікації навчального комп'ютерного комплексу для кабінетів інформатики та інформаційно-комунікаційних технологій навчальних закладів системи загальної середньої освіти". URL: <https://zakon.rada.gov.ua/rada/show/v0614290-10>.

27. **Морзе Н.В.** Інформатика: підруч. для 10 кл. загальноосвітніх навчальних закладів / **Н.В. Морзе, В.П. Вембер, О.Г. Кузьмінська.** – К.: Школяр, 2010. – 304 с.
28. **Жиденко, Т. А.** Використання методу проектів на уроках інформатики / **Т.А. Жиденко** // Інформатика в школі. – 2009. – № 6. – С. 2–7.
29. **Куліковський О. Д.** Модуль “Web-технології” (збірник практичних завдань для 10-11 класів). URL: <https://urok-ua.com/modul-web-tehnologiyi-zbirnyk-praktychnyh-zavdan-dlya-10-11-klasiv/>.
30. **М.М Глибовець, І.О Завадський.** Вступ до вебтехнологій Посібник для 10–12 класів закладів загальної середньої освіти та вищих навчальних закладів. Київ. 160с
31. **Стрілецька Н. М.** Методика навчання інформатики (у початковій школі): навчально-методичний посібник / **Н. М. Стрілецька.** – Чернігів: Чернігівський національний педагогічний університет імені Т. Г. Шевченка, 2014. – 240 с.

## ДОДАТОК 1

### План-конспект уроку з інформатики

**Тема:** Ознайомлення з HTML. Створення простої веб-сторінки

**Клас:** 10

**Тип уроку:** Комбінований (пояснення нового матеріалу + практична робота)

**Тривалість:** 45 хвилин

#### Мета уроку

1. **Навчальна:** Ознайомити учнів з основами HTML, структурою веб-сторінки та принципами роботи тегів.
2. **Розвивальна:** Формувати навички створення базових веб-сторінок, критичного мислення та самостійної роботи.
3. **Виховна:** Розвивати інтерес до веб-технологій та вміння працювати індивідуально.

#### Очікувані результати:

Після уроку учні повинні:

- Розуміти структуру HTML-документа.
- Знати функції основних тегів: `<html>`, `<head>`, `<body>`, заголовки `<h1>`-`<h6>`, абзаци `<p>`, списки `<ul>`, `<ol>`, гіперпосилання `<a>` та зображення `<img>`.
- Вміти створювати прості веб-сторінки, використовуючи вивчені теги.

#### Обладнання та ресурси:

- Комп'ютери з текстовим редактором (наприклад, VSCode, Notepad++) та браузером.
- Проектор або інтерактивна дошка.
- Відео- або PDF-матеріали для демонстрації роботи HTML.

- Доступ до інтернету для перегляду готових прикладів.

## Хід уроку

### 1. Організаційний момент (2 хвилини)

- Привітання.
- Перевірка присутності.
- Оголошення теми та мети уроку.

### 2. Актуалізація знань (5 хвилин)

- **Запитання до класу:**
  1. Що таке веб-сторінка?
  2. Які сайти ви часто відвідуєте?
  3. Чи знаєте, з чого складається веб-сторінка?
- **Мотивація:** Пояснення, як базові знання HTML можуть стати основою для створення власного сайту.

### 3. Вивчення нового матеріалу (15 хвилин)

#### 1. Структура HTML-документа:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Назва сторінки</title>
  </head>
  <body>
    <!-- Основний вміст сторінки -->
  </body>
</html>
```

#### 2. Пояснення основних частин:

- `<html>` – кореневий тег.

— `<head>` – містить мета-інформацію, назву сторінки.

— `<body>` – містить основний контент сторінки.

### 3. Основні теги HTML:

— **Заголовки:** `<h1>...<h6>`

— **Абзаци:** `<p>`

— **Списки:**

○ Ненумерований: `<ul>`

○ Нумерований: `<ol>`

○ Пункти: `<li>`

— **Гіперпосилання:** `<a href="URL">Текст посилання</a>`

— **Зображення:** ``

### 4. Демонстрація прикладу на екрані:

— Створення простої сторінки в реальному часі з текстом та зображеннями.

## Практична робота (20 хвилин)

### Завдання:

Створити веб-сторінку з описом улюбленого фільму або книги, використовуючи вивчені теги.

### Структура сторінки:

1. Назва сторінки – `<title>`.
2. Заголовок сторінки – `<h1>`.
3. Короткий опис (абзац) – `<p>`.
4. Список головних героїв або авторів – `<ul>` або `<ol>`.
5. Додавання зображення – `<img>`.
6. Додавання гіперпосилання – `<a>`.

### Покрокові інструкції:

1. Відкрити текстовий редактор.
2. Зберегти файл з розширенням `.html`.



3. Написати код за зразком.
4. Переглянути результат у браузері.

**Підтримка вчителя:** Індивідуальні консультації для тих, хто стикається з труднощами.

### 5. Підсумок уроку (3 хвилини)

- Обговорення виконаних робіт.
- Короткий повтор основних тегів.
- Підведення підсумків:
  - Що нового ви дізналися сьогодні?
  - Які теги вам запам'яталися?

### 6. Домашнє завдання (2 хвилини)

- **Обов'язкове завдання:** Додати на створену веб-сторінку ще одне гіперпосилання та список із 3 пунктів.
- **Додаткове завдання:** Ознайомитися з іншими базовими тегами (наприклад, `<hr>`, `<br>`, `<strong>`, `<em>`).

## ДОДАТОК 2

### План-конспект уроку з інформатики

**Тема:** Основи CSS. Підключення стилів та застосування базових властивостей для веб-сторінок

**Клас:** 10

**Тип уроку:** Комбінований (вивчення нового матеріалу + практична робота)

**Тривалість:** 45 хвилин

#### Мета уроку

1. **Навчальна:** Ознайомити учнів з основами CSS, способами підключення стилів, типами селекторів та основними властивостями (кольори, шрифти, відступи, розташування елементів).
2. **Розвивальна:** Формувати вміння аналізувати та покращувати структуру веб-сторінки.
3. **Виховна:** Розвивати естетичний смак та творчий підхід до створення веб-документів.

#### Очікувані результати

Після уроку учні повинні:

- Розуміти принципи застосування CSS для веб-сторінок.
- Знати способи підключення CSS (inline, internal, external).
- Вміти використовувати базові селектори та властивості CSS для стилізації веб-сторінок.

#### Обладнання та ресурси

- Комп'ютери з текстовим редактором (VSCode, Notepad++) та браузером.
- Проектор або інтерактивна дошка.
- Приклади HTML-кодів і CSS-стилів для демонстрації.
- Доступ до інтернету (за потреби).

## Хід уроку

### 1. Організаційний момент (2 хвилини)

- Привітання.
- Оголошення теми та мети уроку.

### 2. Актуалізація знань (5 хвилин)

- Повторення основних елементів HTML: структура сторінки, теги `<h1>`, `<p>`, `<ul>`, `<img>`, `<a>`.
- **Запитання до класу:**
  - Як виглядала ваша веб-сторінка з минулого уроку?
  - Чи достатньо використання одного лише HTML для створення привабливих сайтів?
- **Мотивація:** Сьогодні ми навчимося робити ваші сторінки стильними та привабливими за допомогою CSS.

### 3. Вивчення нового матеріалу (15 хвилин)

#### 1. Що таке CSS?

- CSS (Cascading Style Sheets) – це мова стилів, що використовується для оформлення веб-сторінок.

#### 2. Способи підключення CSS:

- **Inline CSS:** стиль безпосередньо в тегу.  
`<p style="color: red;">Це червоний текст.</p>`
- **Internal CSS:** у секції `<style>` всередині `<head>`.  
`<html>`  
`<head>`  
`<style>`  
`p { color: blue; font-size: 18px; }`

```

</style>
</head>
<body>
<p>Це текст з внутрішніми стилями.</p>
</body>
</html>

```

- **External CSS:** у зовнішньому файлі .css.

#### **HTML-документ:**

```

<html>
<head>
<link rel="stylesheet" href="styles.css">
</head>
<body>
<p>Це текст зі стилями з зовнішнього файлу.</p>
</body>
</html>

```

#### **Файл styles.css:**

```

p {
color: green;
font-size: 20px;
text-align: center;
}

```

### **3. Основні селектори CSS:**

- **Тег:** p { color: blue; }
- **Клас:** .my-class { font-size: 16px; }
- **Ідентифікатор:** #my-id { background-color: yellow; }

### **4. Основні властивості CSS:**

- **Колір тексту:** color

- **Колір фону:** background-color
- **Шрифти:** font-family, font-size, font-weight
- **Відступи:** margin (зовнішній), padding (внутрішній)
- **Рамки:** border
- **Розташування:** text-align

#### 4. Демонстрація застосування CSS (5 хвилин)

- Учитель показує на екрані веб-сторінку без стилів і ту ж сторінку з використанням CSS.

#### 5. Практична робота (18 хвилин)

##### **Завдання:**

Поліпшити зовнішній вигляд веб-сторінки з попереднього уроку, застосовуючи CSS.

##### **Інструкції:**

1. **Створіть зовнішній файл стилів** – styles.css.

2. **Підключіть його до сторінки:**

```
<link rel="stylesheet" href="styles.css">
```

3. **Застосуйте стилі:**

- Змініть колір фону сторінки.
- Встановіть інший шрифт для заголовка та абзацу.
- Вирівняйте текст по центру.
- Додайте рамки або тінь для зображень.

##### **Приклад CSS:**

```
body {  
background-color: #f0f8ff;  
font-family: Arial, sans-serif;  
}
```

```
h1 {  
  color: #333;  
  text-align: center;  
}  
p {  
  font-size: 16px;  
  color: #555;  
}  
img {  
  border: 2px solid #000;  
  margin: 10px;  
  box-shadow: 2px 2px 5px #888;  
}
```

**Підтримка:** Учитель консулює учнів, допомагаючи виправити помилки в коді.

#### 6. Підсумок уроку (3 хвилини)

- Коротке обговорення результатів роботи.
- **Запитання до учнів:**
  - Що ви сьогодні навчилися?
  - Чи побачили різницю між сторінкою без CSS і зі стилями?

#### 7. Домашнє завдання (2 хвилини)

1. Додати нові стилі до веб-сторінки:
  - Встановити інший колір для заголовків та посилань.
  - Використати різні відступи (`margin`, `padding`).
2. Ознайомитися з новими властивостями `border-radius`, `text-shadow` та спробувати їх застосувати.

## Рефлексія уроку

Учні оцінюють свій прогрес, відповідаючи на запитання:

- Що сподобалося найбільше?
- Які труднощі виникли?
- Що б ви хотіли додатково навчитися у CSS?