

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЖИТОМИРСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА
Фізико-математичний факультет
Кафедра комп'ютерних наук та інформаційних технологій

Реєстраційний № _____

Дата реєстрації _____

ПРОЄКТНИЙ ПІДХІД ДО НАВЧАННЯ HTML НА ПРИКЛАДІ
СТВОРЕННЯ ВЕБЗАСТОСУНКІВ

Кваліфікаційна (дипломна) робота
здобувача вищої освіти
другого (магістерського) рівня вищої освіти
спеціальності 014 Середня освіта
предметної спеціальності 014.09 Середня
освіта (Інформатика)
освітньо-професійної програми
«Інформатика в закладах освіти»
25Мд-СОінф групи
БАШМАНІВСЬКОГО Максима
Олексійовича

Науковий керівник:
доцент, кандидат педагогічних наук, доцент
КРИВОНОС Олександр Миколайович

Рекомендовано до захисту
рішенням кафедри комп'ютерних наук та інформаційних технологій

Протокол № ____ від «15» листопада 2025 р.

Завідувач кафедри _____ Олена УСАТА

Житомир – 2025

Дата захисту _____

Результати
захисту

Оцінка		
за університетською шкалою	за 100 бальною шкалою	ЄКТС

Голова ЕК

(підпис)

(Власне ім'я ПРІЗВИЩЕ)

Члени ЕК

(підпис)

(Власне ім'я ПРІЗВИЩЕ)

(підпис)

(Власне ім'я ПРІЗВИЩЕ)

(підпис)

(Власне ім'я ПРІЗВИЩЕ)

Секрктар ЕК

(підпис)

(Власне ім'я ПРІЗВИЩЕ)

АНОТАЦІЯ

Башманівський Максим Олексійович. *Проектний підхід до навчання HTML на прикладі створення вебзастосунків* : другий (магістерський) рівень вищої освіти : спеціальність 014 Середня освіта / спеціалізація 014.09 Середня освіта (Інформатика) / Житомирський державний університет імені Івана Франка. Житомир, 2025. 96 с.

Кваліфікаційну роботу присвячено проблемі підвищення ефективності навчання вебтехнологій у закладах загальної середньої освіти. Об'єктом дослідження є процес навчання основ веб-розробки учнів старших класів. Предметом виступають теоретичні та методичні засади використання проектного підходу (PBL) до навчання HTML. Метою роботи є теоретичне обґрунтування та експериментальна перевірка методики навчання HTML через створення вебзастосунків. У ході дослідження використано комплекс методів: теоретичні (аналіз літератури, моделювання), емпіричні (педагогічний експеримент, спостереження, тестування) та методи математичної статистики.

Основним результатом роботи є розроблена модель та методика проектного навчання HTML, яка базується на компетентнісному та діяльнісному підходах. Методика передбачає поетапну організацію роботи над створенням вебпродукту (від ідеї до публікації) з розподілом ролей та чіткими критеріями оцінювання (семантика, валідність, доступність). Ефективність методики підтверджено результатами педагогічного експерименту. Встановлено, що в експериментальній групі, яка навчалася за розробленою методикою, середній бал навчальних досягнень зріс. Також зафіксовано статистично значуще зростання індексу навчальної мотивації (інтересу, самоефективності та навичок співпраці) порівняно з традиційним навчанням.

Наукова новизна полягає в обґрунтуванні цілісної методики навчання HTML, що інтегрує технічні аспекти (валідність, доступність) з розвитком м'яких навичок. Практичне значення має створений комплекс методичних матеріалів, готовий до впровадження у шкільний курс інформатики.

Ключові слова: вебзастосунок, вебтехнології, інформатика, методика навчання, проектна діяльність, проектне навчання, цифрова грамотність, цифрові компетентності, HTML, оцінювання, педагогічний експеримент, навчальний проект, учнівська мотивація.

ABSTRACT

Bashmanivskyi Maksym Oleksiiovych. *Project-Based Approach to Teaching HTML through the Creation of Web Applications* : second (master's) level of higher education : specialty 014 Secondary Education / specialization 014.09 Secondary Education (Informatics) / Zhytomyr Ivan Franko State University. Zhytomyr, 2025. 96 p.

The master's thesis is dedicated to the problem of improving the effectiveness of teaching web technologies in general secondary education institutions. The object of the study is the process of teaching web development basics to senior secondary school students. The subject of the study is the theoretical and methodological foundations of using the project-based learning (PBL) approach to teaching HTML. The purpose of the study is the theoretical substantiation and experimental verification of the methodology for teaching HTML through the creation of web applications. A complex of methods was used in the study: theoretical (literature analysis, modeling), empirical (pedagogical experiment, observation, testing), and methods of mathematical statistics.

The main result of the work is the developed model and methodology of project-based learning of HTML, which is based on competency-based and activity-based approaches. The methodology envisages a stage-by-stage organization of work on creating a web product (from idea to publication) with role distribution and clear assessment criteria (semantics, validity, accessibility). The effectiveness of the methodology was confirmed by the results of the pedagogical experiment. It was established that the average score of learning achievements increased in the experimental group trained according to the developed methodology. A statistically significant increase in the learning motivation index (interest, self-efficacy, and collaboration skills) was also recorded compared to traditional learning.

The scientific novelty lies in the substantiation of a holistic methodology for teaching HTML, which integrates technical aspects (validity, accessibility) with the development of soft skills. The created set of methodological materials, ready for implementation in the school informatics course, has practical value.

Keywords: assessment, digital competencies, digital literacy, HTML, informatics, learning methodology, project activity, project-based learning, student motivation, teaching informatics, web application, web technologies.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1. ТЕОРЕТИКО-МЕТОДОЛОГІЧНІ ЗАСАДИ ПРОЄКТНОГО НАВЧАННЯ ВЕБ-ТЕХНОЛОГІЙ.....	11
1.1. Проєктний підхід як інноваційна педагогічна технологія.....	11
1.2. Специфіка навчання HTML та основ веб-розробки у закладах загальної середньої освіти.....	17
1.3. Аналіз стану використання проєктного підходу при навчанні веб-технологій.....	22
РОЗДІЛ 2. МЕТОДИКА ПРОЄКТНОГО НАВЧАННЯ HTML НА ПРИКЛАДІ СТВОРЕННЯ ВЕБ-ЗАСТОСУНКІВ.....	28
2.1. Модель методики проєктного навчання HTML.....	28
2.2. Етапи та зміст проєктної діяльності учнів при створенні веб-застосунків.....	37
2.3. Діагностика та оцінювання навчальних досягнень учнів.....	60
2.4. Організація та проведення педагогічного експерименту.....	71
РОЗДІЛ 3. ЕКСПЕРИМЕНТАЛЬНА ПЕРЕВІРКА ЕФЕКТИВНОСТІ РОЗРОБЛЕНОЇ МЕТОДИКИ ТА АНАЛІЗ РЕЗУЛЬТАТІВ.....	75
3.1. Дизайн і процедура педагогічного експерименту.....	75
3.2. Результати та статистична обробка.....	78
ВИСНОВКИ.....	85
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ТА ЛІТЕРАТУРИ.....	87
ДОДАТКИ.....	91

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

PBL - Project-Based Learning, проєктне навчання (компетентнісний підхід з орієнтацією на створення кінцевого продукту).

PBL-HTML - модель проєктного навчання основ веб-розробки, сфокусована на створенні учнівського веб-продукту з поетапною організацією, ролями та критеріями якості.

HTML - HyperText Markup Language, мова розмітки для структурування вмісту веб-сторінок; базис подальшого опанування веб-технологій.

ТЗ - технічне завдання; шаблон планування вимог до учнівського продукту (у межах моделі PBL-HTML).

Рубрика - критеріально-орієнтований інструмент оцінювання процесу та продукту (семантика, валідність, доступність тощо).

Чек-лист - перелік мінімальних вимог/критеріїв якості (наприклад, базова доступність, навігація, публікація).

Спринт - короткий відрізок командної роботи з проміжними перевітками якості та взаєморецензуванням.

ВСТУП

В умовах стрімкої цифровізації суспільства та розвитку інформаційних технологій, цифрова грамотність та компетентності у сфері ІТ стають ключовими для успішної професійної діяльності. Веб-технології є невід'ємною частиною сучасної інфраструктури, а мова HTML – фундаментальною основою для створення веб-сторінок та веб-застосунків. Високий попит на веб-розробників зумовлює підвищені вимоги до якості підготовки фахівців, починаючи зі шкільної лави. Традиційні підходи до навчання HTML, що часто зводяться до вивчення тегів поза реальним контекстом, не завжди забезпечують формування глибоких практичних навичок та розуміння принципів створення сучасних веб-продуктів. У цьому контексті актуальним є пошук та впровадження інноваційних педагогічних підходів, здатних підвищити мотивацію здобувачів освіти, розвинути їхню самостійність та вміння застосовувати знання на практиці. Одним із таких підходів є проєктне навчання (Project-Based Learning, PBL), яке передбачає занурення учнів у процес створення конкретного продукту, що сприяє інтеграції знань та розвитку ключових компетентностей. Використання створення веб-застосунків як основи для проєктів при вивченні HTML дозволяє не лише опанувати синтаксис мови, але й зрозуміти її роль у структурі сучасних інтерактивних веб-рішень, роблячи процес навчання більш осмисленим та наближеним до реальних професійних завдань.

Об'єктом дослідження є процес навчання основ веб-розробки (зокрема, мови HTML) здобувачів освіти у старших класах закладів загальної середньої освіти. **Предметом дослідження** виступають теоретичні та методичні засади використання проєктного підходу до навчання HTML на прикладі створення веб-застосунків.

Метою дослідження є теоретичне обґрунтування та експериментальна перевірка ефективності методики використання проєктного підходу до навчання HTML на прикладі створення веб-застосунків.

Для досягнення поставленої мети було визначено такі **завдання дослідження**:

проаналізувати стан дослідженості проблеми застосування проєктного підходу в навчанні веб-технологій та уточнити сутність ключових понять (проєктний підхід, веб-застосунок, методика навчання HTML);

розробити та теоретично обґрунтувати методику проєктного навчання HTML на прикладі створення веб-застосунків (визначити її мету, завдання, принципи, етапи, зміст, форми та методи);

розробити критерії, показники та діагностичний інструментарій для визначення рівнів сформованості знань та вмінь з HTML у здобувачів освіти в умовах проєктного навчання;

експериментально перевірити ефективність розробленої методики проєктного навчання HTML та здійснити аналіз отриманих результатів;

на основі результатів дослідження розробити методичні рекомендації щодо впровадження проєктного підходу до навчання HTML на прикладі створення веб-застосунків у навчальний процес.

Методологічну основу дослідження склав комплекс взаємопов'язаних загальнонаукових та спеціальних методів. Теоретичні методи включали аналіз наукової літератури, стандартів освіти та навчальних програм; порівняльний аналіз підходів до навчання HTML; систематизацію та узагальнення теоретичних положень і практичного досвіду; моделювання методики проєктного навчання. Емпіричні методи охоплювали педагогічний експеримент (констатувальний та формувальний етапи); спостереження за навчальною діяльністю учнів; опитування (анкетування, бесіди); тестування; аналіз продуктів діяльності (створених веб-застосунків). Для обробки даних використовувалися методи математичної статистики (кількісний та якісний аналіз). Застосування такого комплексу методів дозволило забезпечити всебічність, об'єктивність та достовірність дослідження.

Наукова новизна дослідження полягає у тому, що вперше розроблено та теоретично обґрунтовано цілісну методику навчання HTML на основі проєктного підходу через створення веб-застосунків; удосконалено етапи організації проєктної діяльності учнів у цьому контексті; розвинуто питання визначення критеріїв оцінювання навчальних досягнень; визначено та обґрунтовано педагогічні умови ефективної реалізації запропонованого підходу. **Теоретичне значення** роботи полягає у збагаченні теорії та методики навчання інформатики (розділ веб-технологій), внеску в теорію проєктного навчання, обґрунтуванні концептуальних засад використання веб-застосунків як дидактичного інструменту та створенні підґрунтя для подальших досліджень.

Практичне значення отриманих результатів визначається тим, що розроблено методику та комплекс методичних матеріалів (рекомендації, критерії оцінювання), які можуть бути безпосередньо впроваджені в освітній процес закладів загальної середньої, фахової передвищої та вищої освіти. Результати дослідження можуть бути використані при оновленні навчальних програм та розробці електронних навчальних ресурсів. Впровадження запропонованої методики сприятиме підвищенню мотивації учнів, формуванню у них практичних навичок створення веб-продуктів та розвитку ключових компетентностей.

Апробацію основних положень та результатів дослідження здійснено шляхом подання тез доповідей для участі у Всеукраїнській науково-практичній конференції “Інформаційні технології та моделювання системи” (15 травня 2025 року) на тему «Вплив проєктної діяльності на мотивацію студентів при вивченні HTML і CSS» та у I Всеукраїнській науково-практичній конференції “Науковий пошук: сучасні проблеми інформаційних технологій, математики, фізики, астрономії, управління та освіти” (16 травня 2025 року) на тему «Діагностика та оцінювання навчальних досягнень учнів в умовах проєктного навчання HTML».

Структура роботи. Кваліфікаційна робота складається зі вступу, 3 розділів, висновків, списку використаних джерел та додатків. Загальний обсяг роботи становить сторінок 96 сторінок. Робота містить 9 таблиць та 6 рисунків.

РОЗДІЛ 1. ТЕОРЕТИКО-МЕТОДОЛОГІЧНІ ЗАСАДИ ПРОЄКТНОГО НАВЧАННЯ ВЕБ-ТЕХНОЛОГІЙ

1.1. Проєктний підхід як інноваційна педагогічна технологія

Сучасна система освіти перебуває у стані постійних змін, зумовлених швидким розвитком науки, технологій та зростанням інформаційних потоків. Традиційні форми навчання, орієнтовані переважно на репродуктивне засвоєння знань, дедалі частіше не задовольняють потреби суспільства та здобувачів освіти. Вони не забезпечують належного рівня підготовки учнів до життя і професійної діяльності в умовах цифрової економіки, де цінуються здатність до самостійного пошуку інформації, критичне мислення, комунікація та практичне застосування знань.

У цьому контексті постає необхідність використання інноваційних педагогічних технологій, здатних інтегрувати освітній процес із реальними життєвими ситуаціями, мотивувати учнів до активної діяльності та сприяти формуванню ключових компетентностей. Однією з таких технологій є проєктне навчання (Project-Based Learning, PBL) [1], яке передбачає організацію освітнього процесу на основі виконання учнями цілісних практично орієнтованих завдань – проєктів.

Проєктний підхід (Project-Based Learning, PBL) - це інноваційна освітня технологія, що ґрунтується на організації навчальної діяльності учнів через виконання практично орієнтованих завдань, які мають вигляд проєктів. На відміну від традиційних методів, що переважно акцентують увагу на передачі знань від учителя до учня, проєктне навчання передбачає активну діяльність учня, спрямовану на створення конкретного продукту чи розв'язання проблемної ситуації. Сутність PBL полягає у тому, що здобувачі освіти працюють над реальними або максимально наближеними до реальних завданнями, які інтегрують знання з кількох навчальних предметів, сприяють розвитку критичного мислення, комунікативних умінь, креативності та самостійності[2].

Учитель при цьому виступає фасилітатором навчального процесу: він консультує, спрямовує й підтримує, але не домінує.

Основні характеристики проєктного навчання:

- Орієнтація на створення кінцевого продукту (звіт, презентація, вебзастосунок тощо).
- Проблемність і дослідницький характер навчальних завдань.
- Інтеграція знань з різних предметних галузей.
- Співпраця та командна робота учнів.
- Автономність та відповідальність за власне навчання.
- Публічна презентація результатів діяльності.

Традиційна система навчання зазвичай ґрунтується на передачі готових знань від учителя до учня з подальшою перевіркою їх засвоєння у вигляді відтворення матеріалу. Учитель виступає центральною фігурою освітнього процесу, а діяльність учнів часто має репродуктивний характер: вони виконують вправи за зразком, повторюють правила чи алгоритми.

На противагу цьому, проєктне навчання орієнтоване не стільки на засвоєння окремих фактів чи понять, скільки на їх практичне застосування у процесі розв'язання комплексних завдань. Учні отримують можливість самостійно планувати діяльність, шукати інформацію з різних джерел, аналізувати її, інтегрувати набуті знання й презентувати результат у формі проєкту. Ключові відмінності представлені в таблиці 1.1:

Таблиця 1.1

Порівняння традиційного навчання та проєктного підходу (PBL)

Критерій	Традиційне навчання	Проєктне навчання (PBL)
Роль учителя	Головне джерело знань, контролює хід навчання	Наставник, консультант, фасилітатор
Роль учня	Пасивний споживач інформації	Активний учасник, дослідник

Продовження таблиці 1.1

Навчальний процес	Лінійний, орієнтований на підручник і повторення	Відкритий, проблемно-орієнтований
Результат	Відтворення знань у тестах і контрольних роботах	Створення продукту, що має практичну цінність

Проектний підхід до навчання ґрунтується на поєднанні педагогічних і дидактичних принципів, які забезпечують його ефективність та визначають відмінність від традиційних методів. Його ідейне підґрунтя походить з гуманістичної педагогіки та компетентнісного підходу, що орієнтують освітній процес на розвиток особистості учня, формування в нього навичок самостійної діяльності та практично значущих компетентностей.

З педагогічної точки зору проектне навчання спирається на активність учнів, які не отримують готових знань, а самостійно їх шукають, досліджують, обробляють і застосовують. Важливою умовою є співпраця, що передбачає командну роботу, взаємодію з учителем та однокласниками, обмін ідеями й досвідом. Особливе значення має самостійність і відповідальність, адже учні самі визначають шляхи розв'язання поставлених завдань і відповідають за кінцевий результат. Центальною складовою є проблемність: навчання організовується навколо реальної проблеми, яка вимагає залучення знань з різних предметів.

З дидактичної позиції проектний підхід відзначається інтегративністю, оскільки для розв'язання комплексних завдань поєднуються знання з різних галузей. Він завжди має практичну спрямованість, адже результатом є створення конкретного продукту, наприклад застосунку, звіту чи дослідження. Важливим є дослідницький характер навчальної діяльності: учні формують гіпотези, шукають і аналізують інформацію, працюють з різними джерелами. Не менш

суттєвою є рефлексія, що дозволяє осмислити процес власної роботи та результати діяльності всієї команди.

Проектний підхід до навчання HTML реалізується в межах навчального модуля «Вебтехнології» курсу інформатики для 10–11 класів (рівень стандарту). Цей модуль охоплює теми, пов'язані зі створенням і публікацією вебсторінок, гіпертекстовими документами, ергономікою розміщення матеріалів, а також із використанням мультимедійних і графічних елементів. Саме в цьому контексті створення вебзастосунків виступає логічним продовженням тем, присвячених структурі HTML-документа, роботі з гіперпосиланнями, оформленню сторінок засобами CSS та інтеграції елементів взаємодії користувача.

Виконання учнями навчальних мініпроектів - від сайту-візитки до мінікаталогу - дозволяє закріпити теоретичні знання на практиці та сформуванати низку компетентностей, передбачених Державним стандартом базової середньої освіти:

- інформаційно-комунікаційну компетентність - уміння ефективно використовувати цифрові засоби для пошуку, створення та поширення інформації;
- уміння працювати з даними й інформаційними моделями - аналізувати, структурувати й візуалізувати відомості на вебсторінці;
- здатність до співпраці та самостійного навчання - планувати командну роботу над проектом, оцінювати результати та вдосконалювати власний продукт.

Таким чином, модуль «Вебтехнології» є оптимальним освітнім середовищем для впровадження проектного підходу до навчання HTML, оскільки забезпечує зв'язок між змістом курсу, компетентнісними вимогами та практичною діяльністю учнів.

Усе це робить проектне навчання ефективним засобом реалізації компетентнісного підходу в освіті, оскільки воно забезпечує не лише засвоєння

знань, але й розвиток умінь їх застосовувати у реальних і наближених до реальних ситуаціях.

Проектне навчання має низку суттєвих переваг, які роблять його однією з найефективніших інноваційних педагогічних технологій. Насамперед воно сприяє розвитку ключових компетентностей, необхідних у XXI столітті: критичного мислення, уміння аналізувати та оцінювати інформацію, комунікації, співпраці та креативності. У процесі виконання проєктів учні навчаються працювати з різними джерелами, порівнювати дані, робити висновки й приймати самостійні рішення, що формує їхню інформаційну культуру та навички дослідницької діяльності.

Важливою перевагою проєктного підходу є формування практичних умінь і навичок. Учні не лише засвоюють теоретичні знання, а й активно застосовують їх для створення конкретних продуктів: від письмових звітів і презентацій до програмних рішень чи вебзастосунків. Це підвищує мотивацію до навчання, оскільки здобувачі освіти бачать практичну цінність результатів своєї діяльності.

Ще однією сильною стороною є розвиток уміння працювати в команді. Проектне навчання передбачає колективне обговорення ідей, розподіл ролей, спільне прийняття рішень, що вчить учнів співпраці, відповідальності та взаємоповазі. Водночас кожен учасник має можливість виявити індивідуальність, оскільки результат залежить від внеску кожного.

Не менш важливою є орієнтація на самостійність та відповідальність. Учні не лише виконують завдання за вказівкою, а й планують свою діяльність, визначають шляхи розв'язання проблем і відповідають за кінцевий результат. Це формує у них навички самоорганізації та готовність до самостійного навчання впродовж життя.

Останньою, але не менш значущою перевагою, виступає публічна презентація результатів. Вона підвищує рівень мотивації, розвиває навички

публічних виступів, аргументації власної позиції та вміння презентувати продукт своєї діяльності широкій аудиторії.

Таким чином, проєктне навчання не лише забезпечує засвоєння навчального матеріалу, а й формує у здобувачів освіти комплекс компетентностей, що відповідають вимогам сучасного інформаційного суспільства.

Попри значні переваги, проєктний підхід має і певні обмеження, які ускладнюють його широке впровадження у закладах освіти. Однією з головних проблем є значні часові витрати. Виконання навчальних проєктів потребує більше часу, ніж традиційні уроки, оскільки учні проходять кілька етапів роботи - від постановки проблеми та планування до реалізації й презентації результатів. У шкільній практиці, де навчальні програми часто перевантажені, це може стати серйозною перешкодою.

Другим викликом є потреба у матеріально-технічному забезпеченні. Для успішної реалізації проєктів учням часто потрібен доступ до комп'ютерів, програмного забезпечення, інтернету, спеціальних лабораторій чи навчальних ресурсів. У закладах освіти з обмеженим фінансуванням створення таких умов може бути складним завданням [4].

Важливим обмеженням є необхідність високої підготовки вчителя. Учитель у PBL виступає не як традиційний викладач, а як фасилітатор і наставник, який координує роботу, мотивує, допомагає знаходити інформацію, але не дає готових відповідей [5]. Така роль потребує зміни педагогічного мислення, володіння сучасними технологіями та методами, а також готовності працювати в умовах непередбачуваності.

Ще однією проблемою є оцінювання результатів проєктної діяльності. На відміну від тестів чи контрольних робіт, результати проєктів мають комплексний характер, включають як індивідуальний внесок учня, так і роботу команди. Об'єктивне оцінювання потребує розробки спеціальних критеріїв, показників і методик, що вимагає додаткових зусиль від учителя.

Не менш значущим викликом є неоднорідність складу учнівських груп. Часто рівень підготовки, мотивації чи відповідальності різних учнів суттєво відрізняється, що може призводити до конфліктів або нерівномірного розподілу роботи. Це потребує від учителя вміння організовувати співпрацю, підтримувати баланс і сприяти справедливому внеску кожного учасника.

Таким чином, хоча проєктне навчання має потужний потенціал для розвитку компетентностей учнів, його впровадження потребує врахування часових, організаційних і методичних обмежень, а також створення належних умов для ефективної роботи учнів та вчителя.

Проєктне навчання (PBL) виступає сучасною інноваційною технологією, яка орієнтує освітній процес на активну діяльність учнів, розвиток критичного мислення, комунікації та здатності застосовувати знання у практичних ситуаціях. На відміну від традиційних методів, воно базується на створенні кінцевого продукту та розв'язанні реальних проблем, забезпечує інтеграцію знань з різних галузей і сприяє формуванню ключових компетентностей, хоча потребує значних ресурсів, часу та високої професійної підготовки вчителя.

1.2. Специфіка навчання HTML та основ веб-розробки у закладах загальної середньої освіти

У сучасних умовах цифровізації суспільства знання основ веб-розробки є важливою складовою цифрової грамотності. HTML (HyperText Markup Language) виступає фундаментальною мовою розмітки, що забезпечує створення та структурування вебсторінок і є відправною точкою для подальшого опанування CSS, JavaScript та інших технологій. Саме тому його вивчення у школі має не лише прикладне значення, а й формує базу для розвитку інформаційно-комунікаційної компетентності учнів.

Засвоєння HTML дозволяє учням зрозуміти принципи побудови сучасних вебресурсів і вебзастосунків, розвиває логічне мислення, навички

структурованого подання інформації, вміння працювати з кодом і використовувати його для створення практичних продуктів. Крім того, навчання веб-розробки підвищує мотивацію учнів, оскільки вони одразу бачать результат своєї діяльності - створену власноруч вебсторінку або невеликий застосунок.

Вивчення HTML також сприяє профорієнтації, адже знайомить школярів із затребуваною сферою ІТ та може стати першим кроком до вибору майбутньої професії. Таким чином, оволодіння основами HTML у школі виходить за межі суто технічного аспекту і стає засобом формування ключових компетентностей, необхідних для успішної діяльності в умовах інформаційного суспільства.

Вивчення основ веб-розробки, зокрема мови розмітки HTML, передбачено навчальними програмами з інформатики у закладах загальної середньої освіти. Згідно з чинним Державним стандартом базової середньої освіти [6], одним із наскрізних завдань шкільного курсу інформатики є формування цифрової компетентності учнів, яка включає вміння створювати, аналізувати та публікувати інформаційні продукти з використанням сучасних технологій. HTML у цьому контексті виступає засобом навчання, що поєднує технічні знання та практичні навички.

У навчальних програмах для 8–9 класів традиційно передбачено ознайомлення учнів з основами вебтехнологій: поняттям вебсторінки, гіперпосилання, структурою HTML-документа [7]. У старшій школі (10–11 класи) ці знання розширюються завдяки вивченню CSS для оформлення та основ JavaScript для інтерактивності, що дозволяє учням створювати більш повноцінні вебзастосунки. Таким чином, HTML виступає відправною точкою для подальшого опанування інших мов і технологій веброзробки.

У межах шкільного курсу інформатики вебзастосунок варто розуміти як простий навчальний вебпродукт, створений учнями з використанням мови розмітки HTML і базових засобів CSS, який реалізує певну навчальну або комунікаційну функцію. Це можуть бути навчальні вебсторінки, мінісайти або

інтерактивні проєкти, що працюють виключно на клієнтському боці браузера - без використання серверної частини чи баз даних.

Шкільний вебзастосунок має відповідати кільком критеріям:

- Семантика HTML - коректне використання тегів для структурування змісту;
- Базова доступність - забезпечення читабельності, наявність альтернативного тексту до зображень;
- Навігація - зрозуміла логічна структура посилань і сторінок;
- Публікація - доступність результату в мережі, наприклад, через GitHub Pages або шкільний сервер;
- Мінімальна інтерактивність - наявність елементів зворотного зв'язку (форми, кнопки, прості динамічні ефекти).

Типовими прикладами є сайт-візитка з теми уроку, мінікаталог навчальних матеріалів із формою зворотного зв'язку або лендінг шкільного заходу. Такі вебзастосунки дають учням можливість засвоїти основи веброзробки, розвивають інформаційну грамотність, навички візуальної комунікації та підтримують реалізацію компетентнісного підходу, передбаченого Державним стандартом базової середньої освіти.

Зміст навчання HTML у школі має не лише предметне, але й метапредметне значення. Робота з тегами, атрибутами та структурою документа сприяє розвитку алгоритмічного та логічного мислення, навчає систематизувати інформацію і бачити її внутрішню організацію [8]. Окрім цього, інтеграція навчання вебтехнологій з іншими предметами (наприклад, створення сайту з історії, біології чи географії) дає можливість реалізовувати міжпредметні зв'язки та формувати цілісне уявлення про використання ІТ у різних сферах.

Навчання HTML у закладах загальної середньої освіти має свої методичні особливості, пов'язані з віковими особливостями учнів, рівнем їхньої підготовки та обмеженим часом, відведеним на вивчення вебтехнологій у програмі. Основна увага приділяється формуванню розуміння структури HTML-документа, базових

тегів та принципів розмітки, що забезпечує ґрунтовну основу для подальшого вивчення інших мов і технологій веброзробки [9].

Важливою методичною умовою є поєднання теоретичного матеріалу з практичними завданнями. Учні повинні мати можливість не лише ознайомитися з тегами та їх призначенням, а й відразу застосувати їх на практиці, створюючи прості вебсторінки. Поступовий перехід від базових завдань до більш комплексних, які передбачають створення мініпроектів або вебзастосунків, сприяє підвищенню мотивації та усвідомленню практичної цінності знань.

Використання візуальних редакторів (наприклад, *Visual Studio Code*, *Notepad++*) може допомогти учням швидше опанувати основи кодування, проте доцільно поєднувати їх із роботою в «чистому коді», щоб забезпечити глибше розуміння структури HTML-документів. Ефективною методичною стратегією є інтеграція завдань з іншими навчальними предметами, наприклад створення вебсторінки для представлення проекту з біології, географії чи історії. Це дозволяє учням побачити реальне застосування технологій і посилює міжпредметні зв'язки.

Особливе значення має використання проєктних завдань, які передбачають командну роботу, розподіл ролей і створення колективного продукту. Такий підхід не лише розвиває технічні навички учнів, а й формує у них навички співпраці, планування діяльності, критичного мислення та самооцінювання результатів роботи.

Навчання HTML у закладах загальної середньої освіти супроводжується низкою труднощів, що впливають на ефективність формування знань і навичок учнів. Одним із ключових викликів є обмеженість навчального часу: у шкільній програмі інформатики вебтехнологіям відводиться лише кілька годин, чого недостатньо для глибокого опанування основ мови розмітки та створення повноцінних проєктів.

Ще однією проблемою є різний рівень підготовки учнів. Частина школярів має досвід роботи з комп'ютерами й зацікавлена у веброзробці, тоді як інші

обмежуються мінімальними знаннями, що ускладнює організацію роботи з класом. Ця неоднорідність вимагає від учителя застосування диференційованих завдань, адаптованих до можливостей учнів.

Важливим викликом є недостатнє технічне забезпечення. Не всі заклади мають сучасні комп'ютери, швидкісний доступ до інтернету чи ліцензійне програмне забезпечення. У результаті учні обмежені в можливостях створювати та публікувати власні вебсторінки. У таких умовах учителі змушені використовувати спрощені середовища чи локальні інструменти, що знижує ефективність навчання.

Крім цього, важливою проблемою є недостатня підготовка вчителів. Викладання вебтехнологій потребує актуальних знань, адже стандарти HTML і практики веброзробки постійно змінюються. Учителі, які не мають досвіду створення вебпроектів, стикаються з труднощами у поясненні сучасних підходів, що може негативно позначатися на якості навчання.

Сучасні тенденції розвитку освіти вимагають нових методів навчання HTML та основ веброзробки, які б відповідали запитам учнів і потребам інформаційного суспільства. Однією з ключових перспектив є активне впровадження проєктного підходу, що дозволяє поєднувати засвоєння теоретичного матеріалу з його практичним застосуванням. Виконуючи проєкти зі створення власних вебсторінок чи простих вебзастосунків, учні набувають не лише технічних знань, а й навичок співпраці, критичного мислення та самостійності.

Ще однією важливою перспективою є інтеграція навчання HTML з іншими предметами. Вебтехнології можуть використовуватися як інструмент для виконання міжпредметних завдань: наприклад, створення сайту з історії України, електронного атласу з географії чи біологічного довідника. Такий підхід сприяє формуванню цілісного бачення використання ІТ у різних сферах діяльності та підвищує мотивацію учнів до навчання.

В умовах поширення STEM-освіти навчання HTML має перспективу стати складовою більш широких освітніх проєктів, де поєднуються елементи програмування, дизайну, інженерії та дослідницької діяльності. Це відповідає сучасним освітнім практикам, орієнтованим на розвиток інноваційності та підготовку учнів до майбутніх професій у сфері цифрових технологій [9].

Крім того, перспективним є використання електронних освітніх ресурсів, інтерактивних середовищ розробки та онлайн-платформ (наприклад, *CodePen*, *JSFiddle*, *Replit*), які забезпечують доступність інструментів для створення і тестування вебсторінок навіть у випадку обмеженого технічного забезпечення школи. Це дає змогу організувати змішане та дистанційне навчання, що особливо актуально в умовах сучасних освітніх викликів.

Узагальнюючи, можна стверджувати, що навчання HTML та основ веброзробки у закладах загальної середньої освіти має важливе значення для формування цифрової компетентності учнів, розвитку їхніх логічних і творчих здібностей, а також підготовки до майбутньої професійної діяльності в умовах цифрового суспільства. Попри низку труднощів - обмежений час, різний рівень підготовки школярів, недостатнє технічне забезпечення та потребу в підвищенні кваліфікації вчителів - цей напрям має значний потенціал. Перспективи впровадження сучасних підходів, зокрема проєктного навчання, міжпредметної інтеграції, STEM-орієнтації та використання онлайн-ресурсів, створюють умови для ефективного та мотивуючого засвоєння HTML, що робить його вивчення актуальним і корисним у сучасній школі.

1.3. Аналіз стану використання проєктного підходу при навчанні веб-технологій

У сучасних умовах швидкого розвитку інформаційних технологій важливим завданням загальної середньої освіти є формування в учнів умінь і навичок, пов'язаних із веброзробкою. Створення вебсторінок, блогів, мінізастосунків та інших цифрових продуктів дає можливість не лише оволодіти

технічними знаннями, а й розвивати критичне мислення, креативність, уміння працювати в команді та презентувати результати діяльності. Саме ці компетентності є ключовими в умовах цифрової економіки та інформаційного суспільства.

У цьому контексті проєктний підхід (Project-Based Learning, PBL) набуває особливого значення, оскільки він забезпечує інтеграцію навчання вебтехнологій із практичною діяльністю. Використання PBL дозволяє перетворити навчання HTML і основ веброзробки на процес створення реальних цифрових продуктів, що мають прикладне значення. Це підвищує мотивацію школярів, оскільки вони одразу бачать результат своєї роботи, і водночас формує міжпредметні зв'язки та практичні вміння.

Разом із тим виникає потреба у дослідженні стану використання проєктного підходу саме в навчанні вебтехнологій. Попри визнану ефективність PBL у світовій практиці, у школах України його застосування все ще обмежено, що пояснюється низкою факторів: перевантаженістю навчальних програм, недостатнім технічним забезпеченням, а також відсутністю достатньої методичної підтримки для вчителів. Аналіз існуючого досвіду дає можливість виявити не лише проблеми, але й перспективи розвитку цього напрямку.

У науково-педагогічній літературі проєктний підхід розглядається як один із найефективніших методів формування ключових компетентностей учнів. Вітчизняні та зарубіжні дослідники відзначають його здатність поєднувати теоретичні знання з практичною діяльністю та формувати у школярів навички, затребувані в сучасному суспільстві. Так, Н. В. Морзе та О. Г. Кузьмінська підкреслюють, що проєктні методи в навчанні інформатики сприяють розвитку інформаційно-комунікаційної компетентності, оскільки передбачають роботу з реальними цифровими продуктами й ситуаціями [10].

У зарубіжних дослідженнях (J. Larmer, J. R. Mergendoller, S. Boss) [11] PBL позиціонується як стратегія, що сприяє глибокому навчанню (deep learning) завдяки створенню учнями продуктів, пов'язаних із реальними життєвими

проблемами. У випадку веброзробки це може бути створення навчальних сайтів, електронних портфоліо, мініблогів чи інтерактивних ресурсів. Такі завдання підвищують мотивацію школярів і водночас відповідають вимогам компетентнісного підходу в освіті.

У сучасних українських школах проєктний підхід поступово інтегрується в навчання вебтехнологій, проте масштаби його застосування залишаються обмеженими. Здебільшого він реалізується у вигляді виконання учнями невеликих проєктів, пов'язаних зі створенням вебсторінок, електронних презентацій або сайтів для шкільних заходів. Наприклад, у межах курсу інформатики учні 8–9 класів часто отримують завдання розробити простий сайт-візитку, а у старшій школі – колективні проєкти, що охоплюють теми міжпредметного характеру (історія, біологія, література), реалізовані у формі вебресурсів. Досвід використання PBL в українських школах підтверджує, що така діяльність сприяє розвитку в учнів ключових компетентностей, передбачених Державним стандартом базової середньої освіти, зокрема уміння працювати з інформацією, здійснювати комунікацію в цифровому середовищі та створювати власні інформаційні продукти.

Водночас на практиці вчителі часто стикаються з труднощами організації таких завдань: обмеженим часом на уроках, неоднорідним рівнем підготовки учнів, нестачею обладнання чи доступу до швидкісного інтернету. Унаслідок цього проєктна діяльність переважно реалізується у формі індивідуальних або групових мінізавдань, тоді як повноцінні довготривалі проєкти зустрічаються рідше.

Попри зазначені виклики, можна спостерігати позитивну тенденцію до поширення проєктного підходу завдяки впровадженню STEM-освіти та цифрових інструментів. Використання онлайн-платформ (*CodePen*, *Replit*, *GitHub Pages*) дозволяє учням створювати та публікувати власні роботи навіть у разі обмежених технічних ресурсів школи. Це свідчить про перспективність

розвитку проєктного навчання вебтехнологій в Україні та необхідність його систематичного методичного забезпечення.

У світовій практиці проєктний підхід є одним із ключових інструментів навчання вебтехнологій у школах. У країнах Європейського Союзу, США та Канади Project-Based Learning широко застосовується як стратегія розвитку цифрової грамотності та підготовки учнів до життя в умовах інформаційного суспільства. Учні залучаються до створення сайтів, блогів, електронних портфоліо, а також простих вебзастосунків, які мають практичне значення - від презентації результатів дослідження до створення навчальних ресурсів для однолітків.

У США проєктний підхід інтегрований у курси з комп'ютерних наук у середній і старшій школі. Наприклад, у межах програм *Computer Science Principles* та *Web Design* учні працюють над тривалими проєктами, результатом яких є створення власних сайтів або інтерактивних застосунків. Вчитель у такій моделі виступає фасилітатором, а навчальний процес побудований навколо реальних проблем і практичних завдань [12].

У країнах ЄС велика увага приділяється міжпредметним проєктам, де вебтехнології виступають інструментом для інтеграції знань із різних дисциплін. Наприклад, у Фінляндії та Естонії учні створюють освітні сайти з історії, природничих наук чи літератури, поєднуючи вивчення HTML і CSS із розвитком навичок дослідження та комунікації. Це відповідає загальній тенденції STEM- та STEAM-освіти, яка поєднує технічні, творчі та гуманітарні аспекти навчання [13]. Міжнародний досвід демонструє, що ефективність PBL у навчанні вебтехнологій значною мірою залежить від забезпечення учнів сучасними інструментами та середовищами для роботи. Онлайн-платформи, такі як *CodePen*, *Glitch*, *GitHub Pages*, широко використовуються у світі як засоби для створення, тестування та публікації учнівських проєктів. Це не лише розширює можливості навчання, але й дозволяє школярам відчувати себе частиною глобальної спільноти розробників.

Аналіз наукових досліджень і практики впровадження проєктного підходу у навчанні вебтехнологій дозволяє виокремити низку проблем і водночас визначити позитивні тенденції розвитку цього напрямку. Однією з головних проблем є обмеженість навчального часу, що не дозволяє повною мірою реалізувати довготривалі й комплексні проєкти. У більшості випадків учні виконують короткі завдання або мініпроєкти, тоді як систематична проєктна діяльність залишається поза межами базової програми.

Ще однією суттєвою проблемою є недостатнє технічне забезпечення закладів освіти. Відсутність сучасних комп'ютерів, стабільного інтернет-з'єднання чи відповідного програмного забезпечення часто обмежує можливості учнів у створенні та публікації вебпроєктів. Додатковим викликом виступає неоднорідність підготовки школярів: у класі можуть бути як учні з високим рівнем цифрової грамотності, так і ті, хто тільки опановує базові навички.

Серед методичних проблем важливим залишається питання підготовки вчителів. Для ефективного використання PBL педагогу потрібні актуальні знання у сфері веброзробки та досвід організації проєктної діяльності, що не завжди забезпечується традиційною системою підвищення кваліфікації. Це створює бар'єри для широкого впровадження сучасних підходів у школах [14].

Водночас спостерігаються і позитивні тенденції. Зокрема, зростає інтерес до STEM- та STEAM-освіти, де проєктна діяльність виступає базовим інструментом формування компетентностей. Активно поширюється використання онлайн-платформ для створення та презентації учнівських робіт (*Replit, CodePen, GitHub Pages*), що частково компенсує технічні обмеження. Крім того, міжнародний досвід поступово інтегрується в українську практику завдяки участі шкіл у міжнародних проєктах та програмі *eTwinning*, яка стимулює міжшкільну співпрацю через цифрові інструменти [15].

Висновки до першого розділу

Аналіз стану використання проєктного підходу в навчанні вебтехнологій показує, що він є ефективним засобом розвитку ключових компетентностей учнів, зокрема критичного мислення, креативності, комунікації та вміння застосовувати знання на практиці. Українська школа лише поступово інтегрує PBL у процес вивчення HTML та веброзробки, переважно у вигляді мініпроєктів і простих колективних завдань, тоді як у міжнародній практиці він широко застосовується як базова стратегія формування цифрової грамотності. Основними проблемами залишаються обмеженість часу, недостатнє технічне забезпечення та потреба у підвищенні кваліфікації вчителів. Водночас спостерігаються позитивні тенденції, зокрема поширення STEM- та STEAM-освіти, використання сучасних онлайн-інструментів і залучення українських шкіл до міжнародних освітніх проєктів.

РОЗДІЛ 2. МЕТОДИКА ПРОЄКТНОГО НАВЧАННЯ HTML НА ПРИКЛАДІ СТВОРЕННЯ ВЕБ-ЗАСТОСУНКІВ

2.1. Модель методики проєктного навчання HTML

Необхідність створення цілісної моделі методики зумовлена розривом між «лінійним» вивченням синтаксису HTML та реальними вимогами до учнівських результатів у проєктно орієнтованому середовищі. Типові труднощі - обмежений час, неоднорідність підготовки класу, дефіцит критеріїв оцінювання продукту й процесу, а також потреба в чіткому розподілі ролей у команді - вимагають системного рішення. Запропонована модель окреслює логіку від цілей і змісту до процесів та очікуваних результатів, поєднуючи проєктний підхід із компетентнісною, діяльнісною й інтегративною парадигмами навчання інформатики.

Мета моделі - забезпечити кероване впровадження PBL у навчання HTML через зрозумілі етапи та інструменти: постановку проблеми, планування й розробку вебпродукту, проміжні перевірки якості, публічну презентацію та рефлексію. Вона визначає ролі вчителя (фасилітатор, наставник, організатор ресурсів) і учня (дослідник, розробник, презентатор), пропонує мінімальний набір дидактичних засобів (шаблони технічних завдань, рубрики оцінювання, чек-листи якості коду й доступності), а також описує умови реалізації (технічні, організаційні, методичні).

Практична спрямованість моделі полягає у фокусі на створенні завершеного вебпродукту (від статичної сторінки до простого вебзастосунку) з реальним призначенням і зовнішнім користувачем/замовником. Очікуваний ефект - підвищення мотивації, формування предметних (HTML/CSS-основи, структура документа, базові принципи доступності) та надпредметних компетентностей (планування, командна взаємодія, саморефлексія), а також прозорість оцінювання завдяки критеріям, що охоплюють і процес, і результат. Модель слугує «дорожньою картою» для вчителя й учнів, яка дозволяє

масштабувати PBL у межах навчальної програми без втрати керованості та якості.

Вибір створення вебзастосунків як центрального виду діяльності в моделі зумовлений їх відповідністю принципам проєктного навчання та можливістю реалізації повного циклу «від задуму до публічної демонстрації». На відміну від виконання ізольованих вправ із верстки, робота над вебзастосунком інтегрує предметний (HTML/CSS), технологічний (публікація, тестування, валідність) і комунікаційний (презентація, співпраця) аспекти. Такий формат поєднує практичну значущість продукту з реалістичними обмеженнями шкільного середовища: учні створюють завершений цифровий артефакт без складного серверного програмування, але з повною відповідальністю за структуру, зміст і якість подання. Це дозволяє сформувати не лише предметні вміння - семантичну верстку, базову доступність, логіку навігації, - а й ключові компетентності Державного стандарту освіти: уміння працювати з даними, комунікувати в цифровому середовищі, планувати власну діяльність і працювати в команді.

Запропонована модель спирається на поєднання трьох взаємодоповнювальних підходів: компетентнісного, діяльнісного та конструктивістського. Компетентнісний підхід визначає цілі як сформованість у здобувачів не лише предметних знань (структура HTML-документа, семантичні теги, базові засади доступності), а й надпредметних умінь - планування роботи, командна взаємодія, презентація результату, рефлексія. Діяльнісний підхід задає формат навчання через послідовність дій (постановка проблеми → планування → реалізація → тестування/виправлення → публічна презентація → рефлексія), де кожна дія має відчутний продукт і критерії якості. Конструктивізм забезпечує логіку «навчання через створення»: учні конструюють власні знання в контексті розв'язання автентичних задач (реальний замовник або аудиторія, реальні обмеження та вимоги).

На рівні дидактики модель спирається на ключові принципи PBL: навчальна задача є проблемною та осмисленою (автентичний бриф для

вебсторінки/застосунку); діяльність організована як дослідження (збір вимог, аналіз прикладів, вибір технічних рішень); робота колаборативна й ролева (автор контенту, верстальник, тестувальник, презентатор); оцінювання прозоре та багатоаспектне (рубрики для продукту і процесу, само- та взаємооцінювання); рефлексія обов'язкова (короткі ретроспективи після кожного етапу). Окремо підкреслюється принцип поступової ускладненості: від мікропроектів (одна сторінка з коректною семантикою та навігацією) до інтегрованих робіт (мультироздільний сайт/простий вебзастосунок із формами та базовою взаємодією).

Методично модель узгоджується з ТРАСК: учитель поєднує предметний компонент (HTML/CSS-основи, бази доступності та валідності), педагогічний (організація командної роботи, фасилітація дослідження) і технологічний (редактори коду, системи контролю версій, платформи публікації на кшталт GitHub Pages). Це поєднання дозволяє тримати фокус і на якості коду (семантика, читабельність, відповідність стандартам), і на якості навчального процесу (чіткі цілі, проміжні «контрольні точки», підтримка різнорівневих учнів).

Таким чином, теоретичні основи моделі забезпечують цілісність: цілі (компетентності) узгоджені зі змістом (ядро HTML і практики веброзробки), процес - із логікою PBL, а інструменти - з реальними практиками галузі. Це створює умови для керованого, вимірюваного та мотивувального навчання HTML у форматі проєктів.

Запропонована модель вибудовується як цілісна система, у якій цілі навчання визначають зміст, зміст обумовлює процес, а процес приводить до оцінювально-результативного блоку за дотримання певних умов реалізації. На цільовому рівні йдеться про формування поєднання предметних і надпредметних результатів: учні опановують семантику HTML, базові патерни структурування сторінки, принципи доступності та валідності, водночас розвиваючи планування, командну взаємодію, публічну комунікацію та рефлексію. Очікуваний результат формулюється практично: учень уміє сформулювати вимоги до вебпродукту,

обрати адекватні семантичні теги, зібрати каркас сторінки, перевірити валідність і доступність, опублікувати й презентувати роботу.

Змістовий компонент фокусується на ядрі HTML: структурі документа, ключових текстових і структурних елементах, зображеннях і посиланнях, формах, метаданих і мінімальних вимогах до доступності. Цей зміст розкривається у прикладних контекстах - від мікропроектів на одну сторінку до малих багатосторінкових сайтів чи найпростіших вебзастосунків. Матеріал організовано спіралеподібно: повернення до вже відомих понять відбувається на вищому рівні складності, коли учні переходять від простого каркаса до продуманих шаблонів, навігації й перевірок якості.

Процесуальний рівень описує, як саме організовано діяльність. Навчання стартує з постановки проблеми та аналізу прикладів, де клас разом формує короткий бриф: хто є користувачем, яка мета продукту, які сторінки потрібні. Далі команда планує роботу: визначаються ролі, окреслюється беклог задач і розклад коротких спринтів із заздалегідь відомими «контрольними точками». Реалізація розгортається через послідовні ітерації: спочатку створюється каркас і структура навігації, потім наповнюється контент, додаються форми та покращується взаємодія, після чого команда переходить до тестування - валідатори HTML, елементарний аудит доступності, кросбраузерний перегляд - і готує реліз-кандидат. Завершенням слугує публічна презентація з відповідями на запитання аудиторії, а також рефлексія, під час якої кожен учасник фіксує сильні та слабкі сторони власного внеску й команди.

Оцінювально-результативний компонент забезпечує прозорість і справедливість зворотного зв'язку. Якість продукту визначається відповідністю технічному завданню, доречністю семантики, валідністю та базовою доступністю, логікою навігації, охайністю коду та фактом публікації. Якість процесу відстежується через спостереження за плануванням, виконанням ролей, участю у взаємоперевірках і дисципліною спринтів. Важливо, що оцінюється також презентація і рефлексія - здатність аргументувати рішення та осмислювати

результати. Для фіксації прогресу формується невелике портфоліо з посиланням на реліз, коротким технічним описом і висновками.

Функціонування всієї моделі потребує визначених умов. Учитель виступає фасилітатором і ментором, який організовує ресурси й модерує якість, а учні працюють у розподілених ролях із можливою ротацією між спринтами, щоб кожен спробував себе в різних функціях. Необхідні базові інструменти - редактор коду, валідатор HTML, простий механізм публікації на кшталт GitHub Pages або шкільного хостингу, а також готові шаблони технічних завдань, чек-листи якості й рубрики. Часова організація передбачає модуль у межах 8–12 уроків або кілька двотижневих спринтів, де кожна ітерація має чіткий вихід. Диференціація досягається різнорівневими завданнями, парним програмуванням, «експертними столами» та додатковими підказками, а вимоги інклюзії та безпеки регулюють роботу з контентом і персональними даними та підтримують академічну доброчесність.

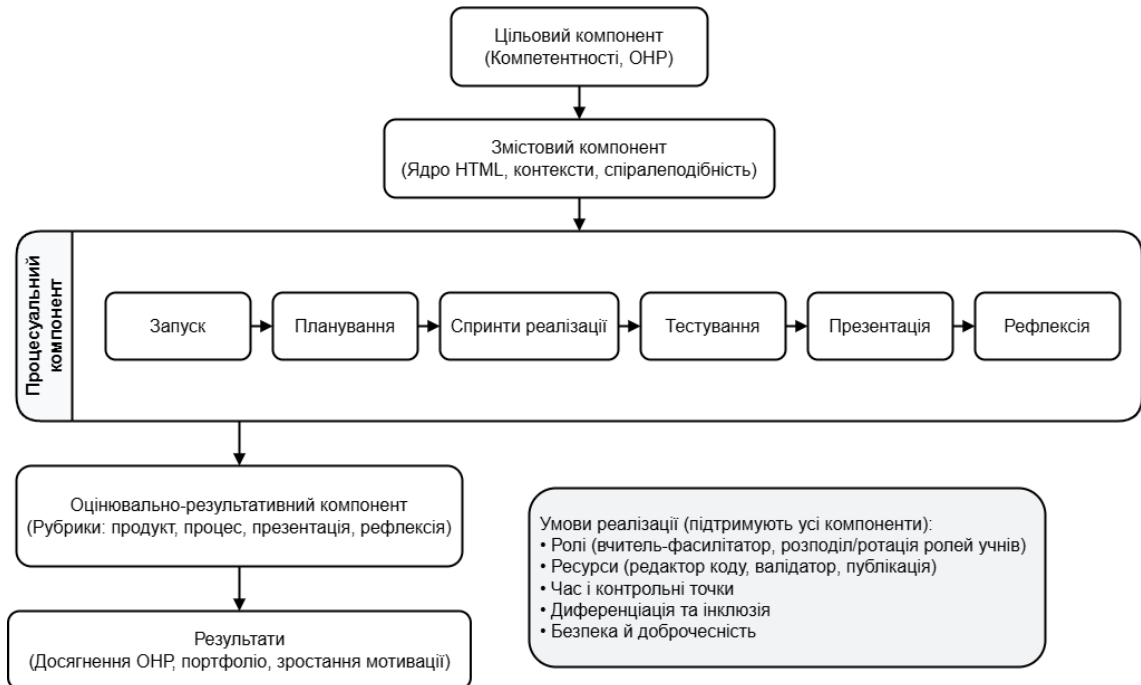


Рисунок 2.1 – Структура моделі методики проєктного навчання HTML

У такому наративному форматі структура моделі зберігає логіку (рис. 2.1) «цілі → зміст → процес → оцінювання → умови» й водночас демонструє, як ці

елементи взаємно підсилюють одне одного: конкретизовані цілі спрямовують добір змісту, зміст задає кроки діяльності, процес виносить продукт на публічну демонстрацію, а оцінювання та рефлексія закріплюють результат і задають траєкторію наступного вдосконалення.

Функціонування моделі передбачає керований цикл «від задуму до публічної демонстрації», де кожен крок має чіткі артефакти, ролі та короткі форматувальні оцінки. Запуск починається з постановки автентичної проблеми й аналізу зразків (2 мікроуроки: семантична структура HTML та критерії якості майбутнього продукту). Клас формує короткий бриф: ціль аудиторії, сценарій використання, перелік сторінок і мінімальні вимоги доступності. Учитель фасилітує, уточнюючи вимірювані результати (ОНР) і узгоджуючи їх із рубрикою.

Далі йде планування: визначаються ролі (верстальник, контент-менеджер, тестувальник, презентатор) і створюється беклог задач, розбитий на 2–3 спринти по 1–2 тижні. На кожен спринт закладаються «контрольні точки»: наприкінці - взаємореview за чек-листом (семантика, валідність, читабельність), коротка самооцінка та оновлення беклогу. Учитель вводить «мікроуроки за запитом» (mini-lessons) рівно під потребу спринту: наприклад, «структура `<header>/<nav>/<main>/<footer>`», «alt-тексти й заголовки рівнів H1–H3», «форми та базова валідація введення».

Спринти реалізації рухаються від каркаса до полірування: спочатку створюють навігацію і шаблон сторінки, далі - наповнюють контент і структурують його семантично, потім - додають форми/просту взаємодію та проводять «аудит доступності 101». Кожен урок починається 3–5-хвилинним стендапом (що зробив / що заважає / що роблю далі) і завершується фіксацією прогресу у спільному трекері (таблиця або kanban) [16]. Для різнорівневих учнів передбачено диференціацію: базова доріжка (чиста семантична верстка, валідність, просте меню) і просунута (форми, таблиці, якорі, легка ARIA, фавікони, метадані).

Після спринтів - етап тестування: учні перевіряють валідність (W3C validator або вбудовані засоби) [17], мінімальні вимоги доступності (наявність альтернативного тексту, логічна ієрархія заголовків, контраст), а також кросбраузерний перегляд. Учитель організовує «критичний перегляд»: інша команда виступає «зовнішнім клієнтом», залишає структурований фідбек за рубрикою. Результатом етапу є реліз-кандидат і короткий технічний опис (що зроблено, як перевірено, що відомо як «технічний борг»).

Публічна презентація відбувається у форматі демо (3–5 хв на команду): показ продукту, обґрунтування вибору семантичних рішень, відповіді на запитання. Оцінюється не лише продукт (відповідність ТЗ, семантика, валідність, доступність, охайність коду, публікація на GitHub Pages/шкільному хостингу), а й комунікація (ясність, логіка, командна участь). Завершує цикл рефлексія: індивідуальний аркуш із самооцінюванням внеску та засвоєних умінь, командна ретроспектива «що вдалося/над чим працювати далі», оновлення портфоліо (посилання, скріншоти, 5–7 речень висновків).

Організаційно модель спирається на регулярний ритм (8–12 уроків або 2–3 спринти), мінімальний набір інструментів (редактор коду, валідатор, платформа публікації, спільний трекер задач) і прозорі правила (рубрики продукту/процесу, чек-листи доступності, шаблони ТЗ). Роль учителя - фасилітувати, вчасно давати «точкові» міні-уроки, підтримувати темп спринтів і культуру взаємоперевірки; роль учнів - брати відповідальність за задачі, підтримувати партнерів і доводити продукт до публікації. Результат - керований проєктний цикл, у якому досягаються ОНР і водночас формується портфель реальних вебпродуктів.

Очікувані результати реалізації моделі охоплюють поєднання предметних, метапредметних і ціннісно-рефлексивних досягнень, які мають бути вимірюваними, спостережуваними та підтвердженими артефактами - кодом, публікацією, презентацією і рефлексією. На предметному рівні учні демонструють коректне структурування HTML-документа з використанням

семантичних тегів і логічної ієрархії заголовків, створюють зрозумілу навігацію без «мертвих» посилань, грамотно працюють із контентом (зображення, списки, таблиці, favicon), дотримуються мінімальних практик доступності (альтернативні описи, читабельні заголовки), забезпечують валідність та охайність коду, а також публікують продукт на шкільному хостингу чи GitHub Pages. Метапредметні результати виявляються у вмінні планувати роботу й виконувати визначені ролі у спринті, підтримувати командну комунікацію через короткі стендапи, брати участь у взаємооцінюванні коду та конструктивно розв'язувати технічні проблеми. Ціннісно-рефлексивний вимір передбачає аргументацію прийнятих рішень під час публічної презентації та усвідомлену рефлексію щодо власного внеску, труднощів і зон подальшого зростання. Нижче подано узгоджені з рубрикою індикатори та рівні досягнень (таблиця 2.1).

Таблиця 2.1

Індикатори та рівні досягнень

Критерій	Базовий	Достатній	Високий
Семантика та структура	Є каркас, поодинокі порушення семантики	Семантика здебільшого коректна, логічні секції/заголовки	Послідовна семантика, продумана структура для контенту
Доступність (мінімум)	Частково заповнені alt, заголовки місцями	Усі зображення з коректним alt, логічні H1–H3	Плюс мінімальні ARIA/фокус, враховано контекст користувача
Валідність/код	Є помилки валідатора, несистемні відступи	Без критичних помилок, читабельний код	Чистий код, відсутні попередження, є короткі коментарі
Навігація	Меню працює частково, трапляються «мертві» лінки	Стабільна навігація, продумана ієрархія	Додаткові зручності (якорі, «до гори», карта сайту)

Продовження таблиці 2.1

Публікація	Проект відкривається локально	Проект опублікований з базовим описом	Публікація + портфоліо (посилання, скріни, опис)
Процес/ команда	Виконує частину ролі, участь нерівномірна	Роль виконано, участь стабільна, фідбек у команді	Ротація ролей, ініціює покращення, якісний фідбек
Презентація	Коротко демонструє продукт	Аргументує ключові рішення, відповідає на питання	Структурована доповідь, демонструє аналіз і рефлексію

Підтвердження досягнень забезпечують артефакти: репозиторій або папка з фінальним кодом і активами, посилання на публікацію з ілюстративними скріншотами, протокол валідації з результатами перевірки та позначений чек-лист доступності, матеріали презентації або запис демо, а також рефлексивний лист з узагальненням внеску, труднощів, рішень і наступних кроків. Вимірювання прогресу поєднує формувальне оцінювання наприкінці кожного спринту (міні-рев'ю коду і коротка самооцінка за кількома критеріями) з підсумковим інтегральним балом за рубрикою, який охоплює продукт, процес, презентацію й рефлексію; для наочності зростання порівнюються перший мікропроект (односторінкова «візитка») і фінальний продукт (багатороздільний сайт або простий вебзастосунок) за тими самими критеріями. Очікуваний вплив на курс інформатики полягає у стійкому підвищенні мотивації завдяки видимому результату та публікації, у закріпленні базової «веб-грамотності» (семантика, доступність, валідність), у формуванні культури співпраці та академічної доброчесності, а також у створенні портфеля робіт, корисного для профорієнтації та подальшого навчання.

Запропонована модель поєднує компетентнісні цілі, ядро змісту HTML, поетапно організований проєктний процес і прозоре багатоканальне оцінювання, спираючись на реалістичні умови реалізації (ролі, ресурси, час, диференціація, безпека). Така конструкція забезпечує керований цикл «цілі → зміст → процес → оцінювання → результати» з вбудованими механізмами зворотного зв'язку, що дозволяє гнучко коригувати навчання та утримувати фокус на створенні завершеного вебпродукту з публічною демонстрацією. Очікуваний ефект - стале зростання предметних умінь (семантика, валідність, базова доступність і публікація), метапредметних компетентностей (планування, командна взаємодія, комунікація) та розвиток рефлексії, підтверджені артефактами в учнівському портфоліо. Модель є масштабованою (від мікропроєктів до міні-застосунків), сумісною з навчальним розкладом (8–12 уроків або 2–3 спринти) і придатною до адаптації під ресурсні обмеження завдяки використанню легкодоступних інструментів (онлайн IDE, валідатори, GitHub Pages).

2.2. Етапи та зміст проєктної діяльності учнів при створенні веб-застосунків

Модуль організовано за логікою проєктного навчання: від постановки автентичної проблеми до публічної демонстрації готового вебпродукту з подальшою рефлексією. Цілі формуються у вимірюваних очікуваних навчальних результатах (учень уміє спроектувати структуру сторінки, застосувати семантичні теги, забезпечити мінімальні вимоги доступності, перевірити валідність коду та опублікувати продукт), а зміст добирається як «ядро HTML» з чітким прикладним призначенням. Процес побудовано ітераційно: короткі цикли розробки із проміжним зворотним зв'язком і корекцією плану, що узгоджується з принципами PBL про глибоке засвоєння через створення значущих продуктів і з сучасними підходами до формувального оцінювання (міні-рев'ю коду, само- та взаємооцінювання, прозорі рубрики).

Оптимальна тривалість модуля в умовах шкільного розкладу - від восьми до дванадцяти уроків, організованих у два-три «спринти» по одному-двох тижнях кожен. Такий ритм дозволяє розвести акценти: у першому циклі учні будують каркас і навігацію, у другому - наповнюють і семантично структурують контент, у третьому - додають форми й перевіряють доступність, після чого відбуваються тестування, публікація та публічне демо. Ітераційність дає можливість регулярно отримувати фідбек і покращувати продукт без перевантаження уроків, а короткі «контрольні точки» наприкінці кожного циклу дисциплінують командну роботу та роблять поступ видимим [18].

Ролі в команді розподіляються так, щоб кожен учень мав конкретну відповідальність (верстальник, контент-менеджер, тестувальник, презентатор) з можливою ротацією між циклами; це підтримує залучення, формує навички співпраці та розвиває метапредметні компетентності. Мінімальний інструментарій - редактор коду, валідатор HTML, простий механізм публікації (наприклад, GitHub Pages [19]) і спільний трекер задач - забезпечує реалістичність і наближає навчання до галузевої практики. Наприкінці модуля кожна команда має портфель артефактів: опублікований проєкт, короткий технічний опис, результати валідації та рефлексивний запис про внесок і наступні кроки, що служить основою для підсумкового оцінювання за рубрикою та відображає прогрес від стартового прототипу до реліз-кандидата.

Етап 0. Запуск (1-2 уроки). Етап запуску закладає спільне розуміння проблеми, мети й критеріїв якості майбутнього вебпродукту. У фокусі - автентичне завдання (для кого й навіщо створюється сайт), аналіз зразків, первинне формулювання очікуваних результатів і мінімальних вимог доступності та валідності. Учитель коротко презентує рамку PBL і роль команди, а далі фасилітує обговорення, щоб учні самостійно визначили функціональні потреби користувача та перетворили їх на чорнове технічне завдання.

Цілі етапу. Сформуувати спільний «бриф» проєкту, визначити аудиторію, сценарії використання й ключові сторінки/функції; домовитися про понятні,

вимірювані критерії успіху (рубрика); узгодити мінімальне «ядро HTML», без якого продукт не може бути прийнятий (структура документа, семантичний каркас, базова навігація, alt-тексти та валідність).

Запропонований хід уроків передбачає початок із постановки проблеми та окреслення користувача: короткий «гачок» описує ситуацію або замовника, після чого клас колективно формулює мету - що саме має вміти кінцевий результат і які завдання користувача він закриває. Для фокусування очікувань використовується спрощена техніка «Jobs to be done» у формулі «Користувач хоче ____, щоб ____», яка допомагає перетворити загальну ідею на конкретну функціональність [20].

Далі йде аналіз зразків (10–12 хв): учні розглядають два-три приклади простих освітніх сайтів або лендінгів, виокремлюють, що працює, а що ні, помічають семантичні області на кшталт header, nav, main і footer, відстежують логіку заголовків і звертають увагу на наявність коректних alt-текстів. Таке порівняння формує основу для власного чек-листа якості й наочно демонструє взаємозв'язок між структурою сторінки та зручністю користування.

Після цього проводиться міні-урок «ядро HTML»: учні швидко відтворюють базовий скелет документа - `<!doctype html>`, `<html lang="...">`, `<head>` із title та meta, а також `<body>` - і знайомляться з ідеєю семантичних блоків. Увага акцентується на мінімальних вимогах доступності, зокрема на альтернативних описах для зображень і логічній ієрархії заголовків H1–H3, а також на практиці валідації як способі контролю якості розмітки.

Наступний етап - формування «чернетки ТЗ». Команди заповнюють шаблон брифа, де визначають аудиторію й мету, перелік ключових сторінок, базову структуру кожної сторінки, список потрібного контенту та критерії готовності (Definition of Done) [21]. Учитель допомагає перекласти загальні формулювання у вимірювані пункти рубрики: наприклад, «усі зображення мають коректний alt», «на кожній сторінці є рівно один H1», «відсутні помилки валідатора рівня error». Паралельно узгоджуються очікувані артефакти та

наступні кроки: що саме команда здасть наприкінці наступного етапу (прототип каркаса і навігації), як фіксуватиме поступ (скрін) і в якому інструменті лишатиме короткі коментарі-апдейти.

Артефактом цього етапу є власне чернетка ТЗ (бриф). У ній стисло описуються аудиторія й мета, подається перелік сторінок - наприклад, «Головна», «Про нас», «Контакти/Форма» - додається ескіз структури кожної сторінки з послідовністю блоків, формується список контенту (тексти, зображення, таблиці) та фіксуються критерії готовності: семантика, валідність, мінімальна доступність і публікація. Тут же визначаються ролі в команді на перший спринт, аби кожен учень розумів свою відповідальність і очікуваний внесок.

Для старту використовується формувальна міні-рубрика (чотири показники, шкала 0–3 бали кожен), яка не лише оцінює, а й скеровує роботу. Вона охоплює ясність мети та аудиторії, де рух іде від розмитої до чіткої й перевірної формули; якість семантичного каркаса в плані, що переходить від переліку блоків без логіки до продуманої структури з коректною ієрархією H1–H3; вимоги доступності у ТЗ, що еволюціонують від відсутніх до конкретних на кшталт «alt для всіх зображень», «логічні заголовки» та «зрозумілий текст посилань»; а також критерії приймання, що трансформуються від загального «щоб працювало» до вимірюваних умов «0 помилок валідатора» і «наявна навігація між усіма сторінками».

Диференціація та інклюзія закладаються в сам формат завдання. Для класів із різним стартовим рівнем пропонуються два типи брифа: базовий - на дві-три сторінки без форм - і просунутий, що додатково включає сторінку з формою зворотного зв'язку та таблицею розкладу. Учні з меншим досвідом отримують шаблон із підказками, тоді як сильніші працюють у вільнішому форматі й беруть на себе виклик знайти п'ять семантичних помилок у зразковому макеті. В усіх випадках обов'язково враховується доступність контенту: контраст шрифтів і зображень, зрозумілі описи, передбачувана навігація.

Можливі ризики попереджаються завчасно. Розмитий проєкт долається за допомогою наочного шаблону брифа та пари прикладів «правильного» і «сирого» ТЗ, завищені амбіції коригуються звуженням обсягу до MVP, а брак контенту компенсується переліком відкритих бібліотек із чіткими вимогами до ліцензій і атрибуції. Навчальна дисципліна підтримується чітким таймингом, видимими контрольними точками та коротким підсумковим стендапом наприкінці уроку, що забезпечує прозору відповідальність і сталий поступ команд.

Очікуваний вихід етапу. Усі команди мають затверджену чернетку ТЗ з переліком сторінок і семантичним каркасом, базовими вимогами доступності та валідності, розподіленими ролями на перший спринт і чіткою метою наступного кроку - створення прототипу структури та навігації.

Етап 1. Планування (1 урок). Мета етапу - перетворити чернетку ТЗ на робочий план проєкту з чіткими ролями, беклогом задач, таймлайном спринтів і прозорими критеріями якості. Учитель модерує, але команда сама приймає рішення щодо розподілу праці та обсягу мінімально життєздатного продукту (MVP).

Очікувані результати уроку. Наприкінці уроку кожна команда має: 1) затверджені ролі (верстальник, контент-менеджер, тестувальник, презентатор) з короткими зобов'язаннями; 2) беклог задач, розбитий на 2–3 спринти з оцінками складності; 3) визначений MVP (які сторінки/функції точно увійдуть у реліз); 4) рубрику оцінювання продукту й процесу, узгоджену з ОНР; 5) шаблони чек-листів якості (семантика, валідність, доступність) і формат «контрольних точок».

Запропонований хід уроку. Швидкий мікроурок (5–7 хв): нагадування про семантичний каркас і «must-have» доступність (H1–H3, alt, зрозумілі тексти посилань) як критерії для Definition of Done (DoD). Розподіл ролей і зона відповідальності (10 хв): команда фіксує ролі на перший спринт, домовляється про «ротацію» у наступному. Кожна роль отримує перелік типових задач (напр.,

тестувальник - перевірка валідатором, чек-лист доступності; презентатор - конспект демо).

Формування беклогу (15 хв): з чернетки ТЗ виписуються конкретні задачі у форматі «користувацьких історій» (напр., «Користувач може перейти з головної на сторінку “Про нас” через меню») [22]. Кожній задачі - оцінка складності (умовні 1–3) і маркер спринту. Додаються «якірні» задачі: створити каркас сторінок, налаштувати навігацію, додати alt до всіх зображень, пройти валідацію без помилок.

План спринтів і контрольні точки (10 хв): команди розкладають задачі по спринтах, визначають дату міні-рев'ю коду та короткого стендапу (на початку кожного уроку 3–5 хв). Рубрика та чек-листи (10 хв): учитель пропонує базову рубрику (продукт/процес/презентація/рефлексія) та чек-лист доступності; команди адаптують під свій проєкт. Підсумок (3 хв): фіксується MVP і перші задачі Спринту 1 (каркас + навігація); домовленість про формат доказів прогресу (скрін/посилання/коментар у трекері).

Приклад фрагмента беклогу (уривок).

- Каркас «Головна», «Про нас», «Контакти» (складність 2)
- Меню з переходами між усіма сторінками (2)
- alt для всіх зображень (1)
- Єдиний H1 на сторінку, логічні H2/H3 (1)
- Перевірка W3C Validator - 0 помилок (2)
- Публікація на GitHub Pages (2)

Міні-рубрика процесу (формування, 0–2 бали за критерій). План спринтів (реалістичність), ясність беклогу (конкретні, перевірювані задачі), розподіл ролей (зрозумілі зобов'язання), критерії DoD (включають семантику, валідність, доступність), готовність чек-листів.

Диференціація. Для класів з різним рівнем підготовки пропонується два рівні плану: базовий (3 сторінки, без форм) та розширений (додає форму з обов'язковими полями і таблицю). Сильніші учні беруть на себе задачі з

додатковими вимогами (внутрішні якорі, фавікони, базові ARIA-мітки), новачки - базові структурні елементи.

Ризики й запобіжники. Занадто великий обсяг - урізати до MVP; розмиті задачі - переформулювати як «користувацькі історії»; «вигорання» ролей - закласти ротацію; недостатня прозорість - вести спільний трекер (таблиця/Kanban) і короткі стендапи.

Артефакти етапу. Затверджений план спринтів, беклог, призначені ролі, адаптована рубрика, чек-листи якості та визначений MVP; це все зберігається в спільній теці/репозиторії і стане основою контролю на Спринті 1.

Етап 2. Реалізація - Спринт 1 (1–2 уроки). Цілі спринту. Наприкінці спринту команда має прототип із коректним каркасом (`<!doctype>`, `<html lang>`, `<head>` з `title`/метаданими, `<body>`), семантичними областями (`<header>`, `<nav>`, `<main>`, `<footer>`), логічною структурою заголовків (єдиний H1 на сторінку, підзаголовки H2–H3), меню з переходами між усіма сторінками, базовими посиланнями/якорями та початковим контентом-заглушками. Зображення (якщо вже додані) мають коректні alt-описи. Код проходить початкову перевірку валідатором без критичних помилок.

Запропонований хід спринту. Короткий стендап (3–5 хв): що робимо сьогодні, хто за що відповідає. Міні-урок (7–10 хв): семантичний каркас сторінки; принцип «єдиний H1», навігація через `<nav>`; правила хороших посилань (зрозумілий текст замість «натисніть тут»); alt для зображень.

Робота в командах (25–30 хв): створення шаблону сторінок (Головна/Про нас/Контакти чи інші з брифа); додавання меню, внутрішніх/зовнішніх посилань, якірних переходів; первинне наповнення заглушками (Lorem/зразкові зображення з легальними ліцензіями). Міні-рев'ю коду (5–8 хв): взаємоперевірка за чек-листом (семантика, H1–H3, alt, «живі» посилання). Фіксація прогресу (2–3 хв): скрін/посилання у трекері, короткий коментар «що зробили/що далі».

Артефакт спринту - прототип каркаса і навігації. Це мультисторінковий шаблон із працюючим меню, логічними заголовками й базовими

посиланнями/якорями; зображення (за наявності) мають доречні alt; у репозиторії/теці збережено одну «еталонну» сторінку як шаблон.

Чек-лист якості (фрагмент):

- Є `<!doctype html>` і `<html lang="uk">`.
- У `<head>` задано осмислений `<title>` і мінімальні meta.
- На сторінці рівно один H1; ієрархія заголовків послідовна.
- Є `<header>`, `<nav>`, `<main>`, `<footer>`; контент розбитий на `<section>/<article>` за потреби.
- Усі зображення з коректними alt (за змістом, а не «image123»). Посилання мають зрозумілі тексти; немає «мертвих» лінків.
- Початкова перевірка W3C Validator: 0 помилок рівня *error*.

Міні-рубрика (формульовальна, 0–3 бали за критерій). Семантичний каркас; логіка заголовків; навігація/посилання; мінімальна доступність (alt, тексти лінків); валідність коду. Підсумок спринту - короткий фідбек учителя + взаємооцінювання.

Диференціація. Базова доріжка: одна «Головна» + ще 1 сторінка, просте меню, без таблиць/форм. Просунута: 3–4 сторінки, внутрішні якорі, навігація «до гори», фавікони/метадані, початкові `<figure>/<figcaption>`. Слабшим - шаблон HTML та зразок меню; сильнішим - завдання виявити й виправити 5 семантичних огріхів у «дефектному» прикладі.

Ризики й запобіжники. Розфокус на оформленні - заборонити стилізацію до завершення семантики/навігації; «мертві» посилання - перевірка кожного лінка під час рев'ю; дублювання H1 - огляд структури через outline-інструменти; копіювання контенту - нагадування про ліцензії й посилання на джерела.

Очікуваний вихід. Команда демонструє клікабельний прототип із каркасом, меню та базовою доступністю; фіксує результати перевірки валідатором і перелік невеликих правок на початок Спринту 2.

Етап 2. Реалізація - Спринт 2 (1–2 уроки). Фокус спринту: наповнення та семантичне структурування контенту; узгоджені таблиці/цитати/медіа; зміцнення доступності.

Цілі спринту. До кінця спринту команда має контент-версію сторінок: текст упорядкований заголовками H2–H3, смислові блоки оформлені через `<section>/<article>/<aside>`, зображення мають доречні alt і, за потреби, підписи через `<figure>/<figcaption>`. Табличні дані відображено елементами `<table>`, з підписом таблиці `<caption>` і заголовками стовпців/рядків (`<th scope="col/row">`). Цитати розмічені коректно (`<blockquote>`, `<q>`, атрибут `cite` за наявності джерела). За потреби використано медіа-вбудовування з описами. Усі сторінки проходять перевірку валідатором без помилок, а мінімальні вимоги доступності витримані.

Запропонований хід спринту.

Короткий стендап (3–5 хв): що додаємо сьогодні, хто за що відповідає.

Міні-урок (7–10 хв): «контент як дані»: коли таблиця виправдана; підписи до зображень із `<figure>/<figcaption>`; коректні цитати; роль `scope` у заголовках `<th>` для читачів з екранними рідерами.

Робота в командах (25–30 хв): додавання текстового контенту, розставлення рівнів заголовків, оформлення списків, вставлення таблиць/цитат/медіа за потреби; перевірка alt і зрозумілих текстів посилань.

Міні-рев'ю (5–8 хв): взаємоперевірка за чек-листом контенту й доступності; виправлення зауважень.

Фіксація прогресу (2–3 хв): оновлення трекера, короткий коментар «що зроблено/що залишилося».

Артефакт спринту - контент-версія сайту. Усі заплановані сторінки заповнено змістом із коректною семантикою для тексту, зображень, таблиць і цитат; збережено єдиний «стиль» заголовків і блоків по всьому проєкту.

Чек-лист якості (фрагмент).

- Є один H1 на сторінку; підзаголовки H2–H3 послідовні, без «стрибків».

- Блоки контенту структуровані `<section>/<article>`; допоміжний матеріал - у `<aside>` за потреби.
- Зображення: змістовні `alt`; для важливих ілюстрацій - `<figure>/<figcaption>`. Таблиці: доречні (дані, а не «верстка»), мають `<caption>`; заголовки - `<th>` із `scope="col/row"`.
- Цитати/джерела: `<blockquote>` для довгих, `<q>` для коротких; атрибут `cite` або вказівка джерела у тексті.
- Тексти посилань зрозумілі без контексту («Розклад подій» замість «натисніть тут»).
- Перевірка W3C Validator: 0 помилок *error*; виправлені попередження, якщо це можливо.

Диференціація. Базова доріжка: мінімум таблиць (1 невелика), без медіа-вбудовувань; акцент на чистій ієрархії заголовків і `alt`. Просунута: кілька таблиць із підписами й `scope`, цитати з `cite`, використання `<figure>/<figcaption>`, внутрішні посилання-якорі на розділи сторінки. Слабшим учням - шаблон сторінки з підказками до кожного блоку; сильнішим - завдання переробити «анти-приклад» (неправильні таблиці/цитати) у валідну семантику.

Ризики й запобіжники. Надмірне використання таблиць «для сітки» - пояснення різниці між табличними даними і макетуванням (заборонено використовувати `<table>` для верстки); пропущені `alt` - автоматизована перевірка перед комітами; «стрибки» рівнів заголовків - швидкий огляд `outline`; джерела цитат - вимагати посилання/вказівку джерела.

Очікуваний вихід. Команда має заповнені й семантично впорядковані сторінки, підтверджену валідність і мінімальні вимоги доступності; сформований список нюансів для покращення у Спринті 3 (форми, удосконалення доступності та полірування).

Етап 2. Реалізація - Спринт 3 (1–2 уроки). Фокус спринту: форми та покращення доступності; полірування перед тестуванням і публікацією.

Цілі спринту. До завершення спринту команда додає щонайменше одну робочу форму (наприклад, «Зворотний зв'язок» або «Підписка»), забезпечує зв'язок `<label>` ↔ елементи введення, використовує атрибути `required`, `type`, `placeholder` обережно (як підказку, а не заміну мітки), формує зрозумілі повідомлення про помилки (неголосові підказки на сторінці), забезпечує логічний порядок фокусу. Мінімальні практики доступності поглиблюються: є текстові альтернативи, зрозумілі тексти посилань/кнопок, коректні заголовки сторінок і підзаголовки, достатній контраст для елементів інтерфейсу. Код проходить валідацію без помилок, попередження - мінімізовані.

Запропонований хід спринту.

Короткий стендап (3–5 хв): хто робить структуру форми, хто - мітки/доступність, хто - перевірки та стан помилок.

Міні-урок (7–10 хв): кістяк форми: `<form method="post/get" action="#">`, поля `<input type="email/tel/text/date">`, `<textarea>`, `<select><option>`. Обов'язковість пар `<label for="id">-<input id="id">`; роль `name` і `value`; різниця між `placeholder` і видимою міткою; базова вбудована валідація HTML (`required`, `pattern`, `min/max`, `type=email`). Акцент на доступність: опис кнопок (текст/`aria-label` у разі іконок), логічний порядок клавішею `Tab`, групування полів через `<fieldset><legend>`.

Робота в командах (25–30 хв): створення/інтеграція форми на відповідній сторінці, перевірка парності `label-input`, налаштування типів полів, додавання коротких підказок (поруч з полем), верстка сторінки «Підтвердження/Дякуємо» (навіть якщо бекенд відсутній).

Міні-рев'ю доступності (5–8 хв): швидкий «клавіатурний тур» без миші, перевірка `alt`, заголовків, текстів кнопок/посилань, контрасту; виправлення знайденого.

Фіксація прогресу (2–3 хв): нотатка у трекері з переліком зробленого та «хвостів» на етап тестування.

Артефакт спринту - форма з мінімальними перевітками і зрозумілою взаємодією. Поля заповнюються, помилки відображаються поруч і зрозумілі для читача/скрінрідера; усі мітки пов'язані з полями; кнопки мають однозначні написи («Надіслати», «Очистити»).

Чек-лист якості (фрагмент).

- Кожне поле має видиму мітку `<label>` і пов'язаний `for/id`.
- Використано адекватні `type` (`email`, `tel`, `date` тощо) і `required` за потреби.
- Є групування пов'язаних полів через `<fieldset><legend>`, якщо доречно.
- Тексти кнопок і посилань зрозумілі (без «натисніть тут»).
- Клавіатурна навігація послідовна; фокус видно.
- Повідомлення про помилки - текстові, біля поля; не лише колір.
- Валідність: W3C Validator - 0 помилок *error*; попередження перевірені.

Диференціація. Базова доріжка: одна проста форма (ім'я, e-mail, повідомлення) з `required` і типами полів. Просунута: додаткові поля (`<select>`, група `<input type="radio">` із `<fieldset>`), атрибут `pattern` для простої перевірки, якорі «До форми»/«Угору», доступне повідомлення «Дякуємо». Слабшим - готовий шаблон із пропусками (`id/for`, тексти помилок), сильнішим - завдання переробити «анти-приклад» у доступну форму.

Ризики й запобіжники. Покладатися лише на `placeholder` - обов'язково мати видимі мітки; невидимий фокус - додати стилізацію фокусу навіть у базовому CSS або встроєними атрибутами/браузерними стилями; нечитабельні повідомлення - робити їх текстовими, поруч з полями; порушення послідовності Tab - перевірити порядок DOM і відсутність `tabindex>0`.

Очікуваний вихід. Сайт містить працездатну доступну форму, узгоджені підказки та обробку помилок, покращену текстову та навігаційну доступність по всьому проєкту; сформовано короткий перелік технічних боргів для етапу тестування й удосконалення.

Етап 3. Тестування й удосконалення (1–2 уроки). Фокус етапу: формальна перевірка якості продукту перед публікацією: валідність HTML, базова доступність, кросбраузерність і усунення «технічного боргу».

Мета етапу. Переконалися, що всі сторінки проходять валідацію без помилок, ключові вимоги доступності виконано, навігація стабільна, а виявлені дефекти задокументовано й виправлено. Результатом має стати реліз-кандидат і короткий технічний звіт, який піде в портфоліо.

Запропонований хід етапу.

Контрольний стендап (3–5 хв): що перевіряємо, як фіксуємо дефекти, хто виправляє.

Міні-інструктаж (5–7 хв): як читати звіти валідатора, що вважати критичною помилкою, які мінімальні перевірки доступності робимо вручну: альтернативні описи, логіка заголовків H1–H3, зрозумілі тексти посилань і кнопок, порядок фокусу, видимість фокусу. Нагадування про кросбраузерний перегляд (щонайменше 2 різні браузері / 2 різні розміри екрана).

Перевірки команди (25–30 хв):

Валідність: проганяємо сторінки через W3C Validator, фіксуємо помилки/попередження, виправляємо критичне (errors = 0).

Доступність (ручний чек): alt для кожного зображення; один H1 на сторінку, послідовні H2–H3; тексти лінків/кнопок зрозумілі поза контекстом; клавіатурний «тур» Tab - фокус іде логічно, видимий; у формах усі label зв'язані з input, є повідомлення про помилки.

Кросбраузерність/розміри: відкриваємо у двох браузерах; перевіряємо дрібні «зсуви» контенту, роботу меню/якорів, поведінку форми.

Критичний огляд «парою» (5–8 хв): інша команда грає роль зовнішнього клієнта, проходить чек-лист, залишає структурований фідбек за рубрикою (продукт/процес/презентація).

Усунення дефектів і фіналізація (10–15 хв): виправляємо список критичних і високих за пріоритетом, решту позначаємо як «технічний борг» у звіті.

Артефакт етапу - реліз-кандидат і технічний звіт. Реліз-кандидат - зібрана версія сайту з нульовими помилками валідатора, підтвердженими мінімальними вимогами доступності та відтестованою навігацією. Технічний звіт на ½–1 сторінку: що перевірено, які помилки виявлено і виправлено, що залишено як борг, посилання на валідатор/чек-лист.

Чек-лист якості (короткий).

- Validator: 0 errors; попередження переглянуті та, де можливо, усунені.
- Доступність: alt у всіх ; один H1; логічні H2–H3; зрозумілі тексти лінків/кнопок; фокус видимий; послідовність Tab правильна; у формах label–input пов’язані; є текстові повідомлення про помилки; за потреби <fieldset><legend>.
- Навігація: усі посилання «живі»; якірні переходи працюють; є шлях назад на головну.
- Кросбраузерність/розміри: переглянуто щонайменше у 2 браузерах/2 ширинах, критичних зсувів немає.
- Добросовісність контенту: джерела зображень/цитат вказані або мають відкриту ліцензію.

Диференціація. Базова доріжка - ручний чек-лист + валідатор. Просунута - додатково перевірка з клавіатури, гіперпосилання-якорі, <figure>/<figcaption>, прості ARIA-атрибути там, де доречно (наприклад, aria-label для іконок-кнопок). Слабшим - готовий шаблон звіту про тестування; сильнішим - «чорний ящик»: дати чужу сторінку з навмисними порушеннями та попросити «довести до релізу».

Ризики й запобіжники. «Повалені» дедлайни - обмежуємо виправлення до критичного, решту в борг; «сліпота» до власних помилок - обов’язковий перехресний огляд іншою командою; надмірна залежність від кольору в повідомленнях - додаємо текстові пояснення.

Очікуваний вихід. Готовий реліз-кандидат з нульовими помилками валідатора, підтвердженою базовою доступністю, стабільною навігацією і

коротким технічним звітом - усе це переходить на наступний етап публікації та публічної презентації.

Етап 4. Публічна презентація (1 урок). Фокус етапу: показати готовий вебпродукт аудиторії, обґрунтувати ключові рішення (семантика, доступність, навігація), отримати зовнішній фідбек і зафіксувати його для портфоліо.

Мета етапу. Навчити учнів стисло й переконливо презентувати результат, демонструючи відповідність ТЗ/рубриці та усвідомленість технічних рішень; відпрацювати навички публічної комунікації, відповідей на запитання та приймання конструктивної критики.

Структура уроку (орієнтовно). Налаштування (3–4 хв): нагадування критеріїв оцінювання виступів і регламенту: 3–5 хв на команду + 1–2 хв запитання.

Демо команд (≈30–35 хв): кожна команда показує опублікований сайт (GitHub Pages/шкільний хостинг), коротко пояснює цілі, аудиторію, ключові сторінки, семантичний каркас, мінімальні рішення з доступності (alt, H1–H3, тексти лінків, фокус), результати валідації, специфіку форми та повідомлень про помилки. Демонстрація готового продукту є підсумковим етапом проектного навчання, що дає змогу учням презентувати результати, усвідомити власний внесок і отримати зворотний зв'язок [23].

Q&A та фідбек (після кожного демо): аудиторія (клас/«зовнішні клієнти») ставить 1–2 запитання; однокласники заповнюють міні-форми фідбеку за рубрикою (продукт/комунікація).

Підсумок (5 хв): учитель фіксує типові сильні рішення та зони росту (узагальнений фідбек) і пояснює, що саме треба додати до портфоліо.

Артефакти етапу:

- Опублікований проєкт (посилання).
- Слайд/шпаргалка демо (3–5 ключових тез).
- Лист фідбеку (2–3 рядки від однолітків/гостей за рубрикою).
- Коротка нотатка «що поліпшити» - піде в розділ «Рефлексія».

Таблиця 2.1

Рубрика публічної презентації (уришок).

Рівень	S1. Структура/таймінг	S2. Ясність і демонстрація (що/чому/як)
3	Чіткий вступ-мета-хід-висновки; вклалися у час	Пояснено рішення, обмеження, альтернативи; живе демо ключових фіч
2	Структура загалом є; невеликий овертайм	Є демо, але не всі аргументи чіткі
1	Структура слабка; суттєвий овер/андер-тайм	Демо поверхове; причини рішень не пояснені
0	Хаотично; таймінг зірваний	Нема релевантного демо

Шаблон сценарію демо (орієнтир 5 хв).

- 20–30 с: ціль/аудиторія, коротко про ТЗ.
- 90–120 с: ключові сторінки → навігація → семантичний каркас (де Н1, як структуровано секції).
- 60–90 с: доступність (alt, фокус, тексти лінків), форма (label–input, повідомлення про помилки).
- 20–30 с: валідація (скрін/звіт), що вдалося найкраще.
- 20–30 с: що плануємо поліпшити.

Диференціація. Базова доріжка - коротке живе демо сторінок і форми. Просунута - додатково: міні-лог «що зламали й полагодили на тестуванні», 1–2 слайди з порівнянням до/після, короткий показ клавіатурної навігації. Учням, яким складно виступати, дати роль демонстратора/читача шпаргалки; учням-лідерам - роль модератора Q&A.

Ризики й запобіжники. Перевищення регламенту - таймер і «жовта картка» за хвилину до кінця; «порожнє» демо без аргументації - обов'язкова шпаргалка з 5 запитань («для кого? що вирішує? де H1? які alt? де звіт валідатора?»); технічні збої - завчасно зберегти локальний архів і скріни ключових перевірок.

Очікуваний вихід. Кожна команда провела стисле, аргументоване демо опублікованого продукту, отримала структурований фідбек і підготувала нотатку з поліпшеннями, яку використає на етапі рефлексії та в портфоліо.

Етап 5. Рефлексія (0,5 уроку). Фокус етапу: усвідомити індивідуальні та командні результати, зафіксувати уроки, перевести фідбек у план наступних кроків і завершити портфоліо.

Мета етапу. Сформувати звичку системної само- й взаєморефлексії після публічної презентації: що вдалося, що завадило, які рішення спрацювали, які варто змінити; закріпити здобуті вміння та окреслити «зони росту» кожного учня і команди загалом.

Запропонований хід ($\approx 20\text{--}25$ хв).

Тиха індивідуальна рефлексія (7–8 хв): коротка анкета «Що зробив(ла) я? Який вплив? Яку проблему розв'язав(ла)? Що зроблю інакше наступного разу?». Рефлексія є обов'язковим компонентом кожної діяльності. Вона забезпечує усвідомлення особистісних досягнень, труднощів і шляхів самовдосконалення, сприяє формуванню навичок самооцінки та саморегуляції [24].

Додається самооцінка за 3–4 критеріями рубрики (продукт/процес/презентація/доступність) у шкалі 0–3.

Мікроретроспектива команди (7–8 хв): формат *Start–Stop–Continue* або *Mad–Sad–Glad*; команда узгоджує 2 конкретні покращення процесу (напр., частіші короткі рев'ю, чіткіші описи задач).

Трансфер у план дій (5–7 хв): кожен формулює один персональний «наступний крок» (мікроціль), команда - короткий список технічного боргу/ідей для версії 1.1 (якщо курс триває). Учитель збирає аркуші, позначає пріоритети та дає стислий коментар.

Диференціація й інклюзія. Учням, яким важко структурувати думки, дати шаблон із підказками та прикладами формулювань; сильнішим - додаткове завдання: «перепиши один фрагмент коду, покращуючи семантику/доступність, і коротко обґрунтуй». За потреби рефлексія може бути усною зі стенограмою або відео-/аудіовідповіддю.

Очікуваний вихід. Завершений пакет портфоліо з індивідуальною та командною рефлексією, чітко визначеними наступними кроками та зафіксованими домовленостями щодо покращень. Рефлексивні матеріали стають підставою для підсумкового оцінювання та планування наступних модулів.

Методичні прийоми підтримки на кожному етапі. Щоб тримати темп і якість усього модуля, варто запроєктувати «опори» для учнів і вчителя: короткі міні-уроки за запитом, регулярні мікроформати взаємодії та прості інструменти фіксації прогресу. Нижче - компактний набір прийомів, які інтегруються саме в ті місця циклу, де вони дають найбільший ефект, і підсилюють формувальне оцінювання.

1) Міні-уроки «під потребу спринту» (7–10 хв).

Запуск: «Скелет HTML-документа і семантичні області»; «Що таке вимірювані критерії якості (DoD)».

Спринт 1: «Єдиний H1 і логіка H2–H3»; «Зрозумілі тексти посилань»; «alt для зображень - як писати змістовно».

Спринт 2: «Коли таблиця доречна: <caption>, <th> і scope»; «<figure>/<figcaption> і роль підписів».

Спринт 3: «Зв'язок label–input, required vs placeholder, клавіатурна навігація й фокус»; «Повідомлення про помилки - текстом, не кольором».

Тестування: «Як читати звіт валідатора і пріоритезувати виправлення».

Контент і доступність (Спринт 2): alt змістовні; таблиці мають <caption> і scope; цитати через <blockquote>/<q> із джерелом.

Форми (Спринт 3): кожне поле має <label> (зв'язок через for/id); доречні тегі; є текстові підказки й повідомлення про помилки; фокус видимий; Tab іде по порядку.

2) Ритуали командної роботи (дуже короткі, але регулярні).

Щоуроку: стендап 3–5 хв («що зробив / що заважає / що роблю далі»).

Наприкінці кожного спринту: міні-рев'ю коду 5–8 хв у парах за чек-листом; мікросамооцінка 3–4 критерії рубрики (0–3 бали).

Обов'язкова ротація ролей між спринтами, щоб кожен спробував інші види діяльності.

3) Фіксація прогресу й прозорість плану.

Спільний трекинг задач (таблиця або Kanban): статуси «To do / In progress / Review / Done», 1–2 скріни або посилання після уроку.

«Контрольні точки» у середині та наприкінці спринту з коротким протоколом (дата, що готово, що блокує, рішення/хто робить).

Шаблони документів у спільній теці: бриф/ТЗ, чек-листи, рубрика, звіт тестування, рефлексія.

4) Педагогічні мікроприйоми для диференціації.

Базові/розширені доріжки завдань: усі роблять семантику й валідність; охочі додають таблиці, форму, підписи до зображень, якорі.

Парне програмування: сильніший із новачком - чіткий розподіл «водій/навігатор», ротація всередині уроку.

Експертні столи: учні, які випередили, на 10 хв стають «експертами» для сусідніх команд (відповідають за 1–2 критерії чек-листа).

Шаблони та анти-прикладі: слабшим - налаштовані шаблони сторінок/форм; сильнішим - дефектні фрагменти коду для переробки.

5) Формувальне оцінювання як «вбудований» зворотний зв'язок.

Кожен спринт завершується короткою рубрикою (продукт/процес/доступність/комунікація, 0–2 бали), яка дає чіткий сигнал «що підтягнути».

На тестуванні - список дефектів із пріоритетами (Critical/High/Medium), де «Critical» виправляється обов'язково, решта потрапляє в «технічний борг».

Після презентації - міні-форми фідбеку від однолітків за 2–3 критеріями, що входять у підсумкову рубрику.

6) Мінімальний інструментарій і правила безпеки.

Інструменти: редактор коду (локальний або онлайн), W3C Validator, проста публікація (GitHub Pages/шкільний хостинг), спільний трекер задач.

Безпека й доброчесність: не публікувати персональні дані; використовувати зображення з відкритими ліцензіями з вказанням джерела; у презентації називати запозичення коректно.

Результати, які утворюються завдяки цим прийомам: бриф/ТЗ, беклог і план спринтів, проміжні прототипи, чек-листи з відмітками, звіт валідатора, список дефектів і технічного боргу, опублікований реліз, презентаційна шпаргалка, індивідуальна та командна рефлексія. Сукупно вони роблять навчальний поступ вимірюваним і прозорим, що відповідає логіці PBL і сучасним підходам до оцінювання.

Ризики й шляхи пом'якшення. Ефективність модуля залежить від передбачення типових ризиків і заздалегідь спроектованих «запобіжників». Нижче подано узагальнені проблеми, що найчастіше виникають у проєктному навчанні HTML, та практичні способи їхнього пом'якшення на рівні змісту, процесу й організації. Як зазначає Є. С. Полат, успіх проєктної методики значною мірою залежить від здатності педагога передбачити типові труднощі та створити систему запобіжних дій на етапі проєктування [25].

Брак часу та «розповзання» обсягу

Класична пастка - амбіційні ідеї, які не вміщаються в 8–12 уроків. Лікується чітким MVP: список мінімальних обов'язкових артефактів (каркас, навігація, валідність, базова доступність, одна форма), фіксованими «контрольними точками» наприкінці кожного спринту та жорстким правилом - спершу семантика й доступність, потім прикраси. Якщо команда відстає, вчитель

скорочує вимоги до «версії 1.0», решту переносить у «технічний борг» (версія 1.1).

Нерівномірний рівень підготовки в групі
Сильні учні «витягують» усе на себе, слабші випадають. Вирівнюємо за рахунок ротації ролей, парного програмування «водій/навігатор», двох доріжок складності (база/просунута), шаблонів сторінок/форм для старту й «анти-прикладів» для відпрацювання помилок. Оцінювання внеску - окремо від оцінювання продукту (індивідуальний журнал задач/мітки авторства).

Фокус на оформленні замість змісту
Учні хочуть «красиво» раніше, ніж «правильно». На перших спринтах блокуємо стилізацію: лише семантика, навігація, валідність, доступність. Дизайн дозволяємо мінімально, коли пройдено валідатор і чек-лист доступності. Мікроуроки підкреслюють користь семантики для пошуку/доступності, що підвищує внутрішню мотивацію.

«Мертві» посилання, дубльовані заголовки. Це типові технічні огріхи. Вбудовуємо короткі чек-листи у кінець кожного уроку (5 хв на взаємоперевірку), а також «клавіатурний тур» Tab наприкінці Спринту 3. Перед комітом - швидка перевірка валідатором, правило: 0 помилок рівня error.

Слабка командна взаємодія. Конфлікти або «мовчазні пасажери» руйнують темп. Допмагають щоденні 3–5-хв стендапи з трьома питаннями, прозорий трекер задач (To do/In progress/Review/Done), короткі протоколи міні-рев'ю та обов'язкова участь щонайменше двох учасників у публічному демо. Домовленості фіксуємо письмово (правила фідбеку, дедлайни, критерії готовності).

Обмежені ресурси (техніка/інтернет/хостинг). План «В»: онлайн-IDE (або портативні редактори), парна робота за одним комп'ютером із ротацією кожні 10–12 хв, офлайн-прототипування сторінок на папері на старті, публікація на шкільному сервері чи локальне демо зі скрінами в портфолію, якщо доступу назовні немає.

Академічна доброчесність і ліцензії. Недопустимі копіювання текстів/зображень без атрибуції та публікація персональних даних. Вводимо короткий кодекс: джерела вказуємо, зображення - з відкритими ліцензіями, у формах не збираємо ПД. У презентації команда окремо згадує всі запозичення [26].

Перевантаження вчителя мікроконсультаціями. Фасилітатор «тоне» у повтореннях. Допомагає банк мікроуроків (відео-GIF/1-сторінкові пам'ятки), «експертні столи» (учні-помічники на 10 хв), шаблони (бриф, рубрики, чек-листи) і спільний FAQ у класному документі.

Стрес під час публічної презентації. Тремтячі руки - звична справа. Готуємо шпаргалку сценарію на 5 хв («мета/аудиторія → сторінки/навігація → семантика/доступність → валідація → що поліпшити»), дозволяємо альтернативні формати (коротке відео-демо), чіткий тайм-менеджмент і дружній фідбек за рубрикою.

Втрата мотивації на середині модуля. Щоб уникнути «просідання», закладаємо видимі проміжні результати: наприкінці Спринту 1 - клікабельний каркас, наприкінці Спринту 2 - контент-версія, наприкінці Спринту 3 - форма й «аудит доступності 101». Кожен крок - маленьке демо й запис у портфолію.

Таблиця 2.2

Коротка матриця «ризик → запобіжник»

Ризик	Що робимо заздалегідь	Що робимо, якщо вже сталося
Брак часу / «розповзання»	MVP, контрольні точки, правило «спершу семантика»	Обрізаємо до основного, переносимо зайве в борг
Нерівний рівень	Ротація ролей, пари «водій/навігатор», шаблони	Перерозподіляємо задачі, даємо анти-приклад для виправлення
Фокус на дизайні	Заборона стилізації до валідності/доступності	«Заморожуємо» CSS, повертаємося до чек-листа семантики

Продовження таблиці 2.2

Технічні огріхи	Чек-листи + валідатор наприкінці уроку	Гаряче виправлення перед демо, фіксуємо причину в звіті
Слабка взаємодія	Стендапи, трекер, правила фідбеку	Перезбір ролей, «експертні столи», фасилітація вчителя
Немає ресурсів	Онлайн-IDE, парна робота, офлайн-прототипи	Локальне демо + скріни в портфоліо, публікація пізніше
Порушення добročесності	Інструктаж, шаблон посилань на джерела	Переробка з атрибуцією, зниження балів за рубрикою процесу
Перевантаження вчителя	Банк міні-уроків, FAQ, учні- експерти	Обмежуємо «1-на-1» до тайм-слотів, скеровуємо до FAQ
Стрес на демо	Сценарій, альтернативні формати, таймер	Підтримка ролями, дозволяємо повторну спробу короткого фрагмента
Просідання мотивації	Маленькі перемоги кожен спринт, публічність	Додаткові «швидкі перемоги» (якір, favico, caption)

Такий набір запобіжників робить модуль стійким до типових збоїв і підтримує головну мету: довести кожную команду до якісного, валідного й доступного опублікованого продукту з усвідомленою аргументацією рішень.

Запроєктований модуль поетапної діяльності - від запуску й планування до реалізації в трьох спринтах, тестування, публічної презентації та рефлексії - забезпечує керований цикл створення учнівського вебпродукту й робить навчання HTML змістовним, прикладним і вимірюваним. Чітко визначені очікувані результати, MVP і критерії якості (семантика, валідність, мінімальна доступність, публікація) утримують фокус на суті, а ритм коротких спринтів із мікроуроками «за запитом», стендапами й взаємореєв'ю дає регулярний зворотний зв'язок і підсилює формувальне оцінювання. Інтегровані методичні опори - шаблони ТЗ, чек-листи, рубрики, трекер задач - роблять поступ

прозорим, а артефакти (код, звіти валідатора, опублікований реліз, презентація, рефлексія) забезпечують доказовість досягнень.

Диференціація та інклюзія закладені у зміст, процес і продукт: дві доріжки складності, ротація ролей, парне програмування, мультимодальні пояснення та адаптації оцінювання дозволяють кожному учневі досягти базових результатів і водночас створюють простір для просунутих викликів. Передбачені ризики (брак часу, нерівний рівень підготовки, технічні обмеження, стрес презентації) компенсуються наперед визначеними запобіжниками: пріоритизацією MVP, контрольними точками, «експертними столами», офлайн/онлайн альтернативами та чіткими правилами доброчесності. У підсумку розділ 2.2 демонструє цілісну, масштабовану й життєздатну методику, що водночас формує предметні вміння веброзмітки й метапредметні компетентності співпраці, планування, комунікації та рефлексії, підтверджені публічним результатом.

2.3. Діагностика та оцінювання навчальних досягнень учнів

Оцінювання в модулі PBL-HTML покликане (1) підтвердити досягнення очікуваних результатів навчання з HTML та мінімальної доступності; (2) забезпечити безперервний зворотний зв'язок для корекції навчання під час спринтів; (3) зафіксувати індивідуальний внесок і командну взаємодію; (4) мотивувати до якісного, публічно придатного продукту й свідомої рефлексії.

Ключові принципи. Вирівнювання з цілями (alignment). Кожен критерій напряму відповідає ОНР: семантика/структура, валідність, базова доступність, навігація/посилання, публікація, презентація, рефлексія.

Баланс формувального та підсумкового оцінювання. Короткі «мікрооцінки» після спринтів (чек-листи, міні-рубрики) + підсумковий інтегральний бал за продукт, процес, презентацію й рефлексію.

Критеріальність і прозорість. Оцінювання лише за наперед відомими критеріями й дескрипторами рівнів; жодних «вражень» чи бонусів за оформлення, що не пов'язане зі змістом.

Доказовість (evidence-based). Кожне судження підтверджене артефактами: кодом, звітом валідатора, чек-листом доступності, посиланням на публікацію, матеріалами демо, рефлексією.

Надійність і справедливість. Єдині рубрики для всіх команд; «якірні» приклади робіт; за можливості - перехресна перевірка артефактів між командами.

Аутентичність. Завдання та критерії відображають реальні практики веб-розробки початкового рівня: семантика, валідність, мінімальна доступність, публікація й аргументація рішень.

Інклюзивність і доступність. Альтернативні способи демонстрації результату (живе демо/запис екрана/серія скрінів), прості формулювання інструкцій, можливість додаткового часу для окремих учнів.

Добросовісність. Чесність щодо авторства: позначення запозичень і джерел контенту, заборона на публікацію персональних даних.

Своєчасність фідбеку. Короткий, конкретний зворотний зв'язок «що працює / що покращити / як це зробити» одразу після контрольних точок.

Масштабованість і економія часу. Мінімально достатній набір інструментів (1–2 стор. рубрик + короткі чек-листи), щоб не перевантажувати уроки.

Операціоналізація принципів (хто? що? коли? як?).

- Що: предметні критерії (семантика, валідність, доступність, навігація, публікація) + процесні (планування, роль, взаємодія, дотримання дедлайнів) + комунікаційні (презентація, аргументація) + рефлексія.
- Коли: пре-діагностика на старті; формувальні міні-оцінки наприкінці кожного спринту; підсумок на тестуванні/демо; пост-діагностика (за потреби).

- Хто: учитель (основне оцінювання), учні (самооцінка/взаємооцінка за скороченими формами), «зовнішній клієнт» (стислий відгук під час демо).
- Як: рубрики (0–2/0–3 бали за критерій з чіткими дескрипторами), чек-листи (так/ні), короткі журнали внеску в трекері, портфоліо артефактів.

Таблиця 2.3

Матриця узгодження (ОНР → критерій → артефакт → індикатор).

ОНР (учень уміє...)	Критерій оцінювання	Артефакт-доказ	Індикатор/межа прийнятності
Семантично структурувати сторінку	Семантика й заголовки	HTML-код	1×H1/стор., послідовні H2–H3, логічні секції
Забезпечити мінімальну доступність	Доступність	Чек-лист, код	alt у всіх , видимий фокус, зрозумілі тексти лінків
Підтвердити валідність	Валідність	Звіт валідатора	0 помилок (errors)
Реалізувати навігацію	Навігація/посилання	Живе демо/посилання	Усі посилання «живі», є шлях на головну
Опублікувати продукт	Публікація	Посилання, скріни	Доступна сторінка (GitHub Pages/шк. хостинг)
Пояснити рішення	Презентація	Демо/шпаргалка	Аргументація семантики/доступності, Q&A
Усвідомити внесок/зони росту	Рефлексія/процес	Рефлексивний лист	Конкретні приклади й наступні кроки

Такий набір цілей і принципів забезпечує прозору, справедливу й доказову систему оцінювання, що не лише фіксує підсумок, а й керує навчальним процесом упродовж усього модуля PBL-HTML.

Оцінювальна система вибудовується як прозорий цикл «цілі → критерії → збирання доказів → зворотний зв'язок → корекція плану», у якому кожен учасник має визначені обов'язки, інструменти та артефакти. Архітектура системи:

- Рівень критеріїв: предметні (семантика, валідність, доступність, навігація/посилання, публікація) і процесні (планування, роль/відповідальність, співпраця, комунікація, рефлексія).
- Рівень інструментів: рубрики продукту й процесу (короткі, 1–2 стор.), чек-листи якості, пре/пост-тест, карти оцінювання презентації, форми само- та взаємооцінювання, портфоліо.
- Рівень процедур: коли і як саме збираємо докази (кінці спринтів, тестування, презентація, рефлексія), як надаємо фідбек (усно, письмово, у трекері), як фіксуємо рішення (план правок).

Таблиця 2.4

Ролі та відповідальність.

Роль	Основні дії	Інструменти	Артефакти-докази	Коли
Учитель (фасилітатор оцінювання)	Налаштовує критерії/рубрики, проводить пре/пост-тест, перевіряє артефакти, надає формувальний фідбек	Рубрики, чек-листи, тести, трекер задач	Оцінювальні листи, протоколи міні-рев'ю, зведена відомість	Старт (пре), кінець кожного спринту, тестування, демо, пост
Учні (команда)	Самооцінка/взаємооцінка, підготовка доказів, фіксація внеску, приймання фідбеку й корекція плану	Форми само/взаємооцінки, чек-листи, трекер	Портфоліо: код, звіти валідатора, публікація, слайди, рефлексія	Наприкінці кожного уроку/спринту; демо; рефлексія

Продовження Таблиці 2.4

«Зовнішній клієнт» (однолітки/запрошені)	Короткий незалежний фідбек на демо за 2–3 критеріями	Карта фідбеку презентації	Лист відгуку (2–3 пункти)	Під час публічної презентації
---	--	---------------------------	---------------------------	-------------------------------

Канали зворотного зв'язку:

- Миттєвий усний (стендап 3–5 хв, міні-рев'ю коду 5–8 хв).
- Письмовий короткий (чек-лист + 2–3 речення «що добре / що покращити / як»).
- У трекері задач (коментар до задачі, зміна статусу, план правок).
- Підсумковий (оціночний лист за рубрикою + нотатка для портфолію).
- Потік даних (як це працює).
- На старті - пре-діагностика (короткий тест + міні-практика) і презентація критеріїв.
- Після кожного спринту - формувальна мікрооцінка: взаємооцінка за чек-листом, самооцінка (0–2 бали/критерій), короткий фідбек учителя; оновлення беклогу.
- На етапі тестування - верифікація артефактів: звіт валідатора (errors = 0), чек доступності, список дефектів із пріоритетами.
- На презентації - оцінка демо (рубрика презентації) + зовнішній фідбек.
- Після демо - рефлексія (індивідуальна й командна) і підсумкове зведення (інтегральний бал + коментар).
- Правила справедливості й прозорості.
- Один набір рубрик для всіх; перед початком - розбір 1–2 «якірних» прикладів із виставленням балів.
- Окремий облік індивідуального внеску (журнал задач/мітки авторства), щоб групова оцінка не «поглинула» роботу окремих учнів.

- Доказовість будь-якого балу: у підсумковому листі є посилання на конкретний артефакт (код/звіт/слайд/нотатка).
- Мінімальний пакет документів (що має бути в теці групи). Рубрики продукту та процесу, шаблони чек-листів (семантика/доступність/форми), бланки само- й взаємооцінювання, форма відгуку «зовнішнього клієнта», зразок зведеної відомості, інструкція «як оформити портфоліо артефактів».

Така структура робить оцінювання керованим і економним за часом: критерії зрозумілі наперед, ролі розподілені, докази збираються «по дорозі», а фідбек одразу впливає на наступні кроки команди. Нижче подано перелік критеріїв із чіткими показниками вимірювання. Вони узгоджені з ОНР та артефактами проекту (код, звіти валідатора, чек-листи доступності, публікація, демо, рефлексія). Рівні досягнень (Базовий/Достатній/Високий) використовуються однаково для всіх критеріїв.

Таблиця 2.5

Предметні критерії

Критерій	Показники (індикатори)	Доказ
Семантика та структура	На кожній сторінці рівно 1×<h1>; послідовні H2–H3; присутні <header>, <nav>, <main>, <footer>; змістові блоки виділено <section>/<article>;	HTML-код; скрін структури; короткий коментар
Навігація та посилання	Меню працює на всіх сторінках; відсутні «мертві» лінки; тексти посилань зрозумілі без контексту;	Живе демо; перелік перевіраних лінків
Доступність (мінімум)	Усі мають змістовний alt; логічна ієрархія заголовків; у формах - пари label–input, текстові повідомлення про помилки	Чек-лист доступності; коротке відео «клавіатурного туру»

Продовження таблиці 2.5

Валідність HTML	0 помилок (errors) у W3C Validator; попередження проаналізовано	Звіт валідатора (посилання/скрін)
Робота з контентом	Зображення з доречними підписами через <figure>/<figcaption> (де це критично); таблиці лише для даних, з <caption> і scope у <th>; коректні цитати <blockquote>/<q> із джерелом	Код + скріни фрагментів
Форми (якщо є у ТЗ)	Доречні type (email, tel, тощо), required за потреби;	Демо заповнення форми; код
Публікація	Опублікована версія доступна за посиланням; базовий опис (readme/сторінка «Про проєкт»)	Посилання; скріншоти

Таблиця 2.6

Метапредметні критерії

Критерій	Показники (індикатори)	Доказ
Планування та відповідальність	Є чіткий беклог; задачі розподілені й виконані в строк; зафіксовані «контрольні точки»	Трекер задач; протоколи міні-рев'ю
Робота в ролях і співпраця	Ролі визначено; була ротація; участь у стендапах і взаєморев'ю;	Відмітки в трекері; нотатки команди
Комунікація та презентація	Структуроване демо; аргументовані відповіді на запитання	Слайд/шпаргалка; відео/спостереження
Рефлексія	Конкретні приклади внеску; аналіз труднощів і рішень; сформульовані «наступні кроки»	Рефлексивний лист (½–1 стор.)

Рубрикатор рівнів (єдина шкала для всіх критеріїв):

- Базовий: вимога виконана частково або з помітними вадами (є структурні/логічні огріхи; окремі індикатори відсутні).
- Достатній: вимога виконана повністю на базовому рівні якості (індикатори здебільшого дотримано; незначні неточності не впливають на використання).
- Високий: вимога виконана якісно й послідовно (усі індикатори дотримано; є докази продуманості рішень і стабільності в різних кейсах).

Нижче подано уніфіковану шкалу рівнів і дескрипторів, придатну для всіх критеріїв із попереднього пункту (предметні та метапредметні). Шкалу можна використовувати у 3-бальній (0–2) або 4-рівневій (0–3) інтерпретації. Формулювання дескрипторів зроблено «спостережуваними» - їх легко звірити з артефактами.

Таблиця 2.7

Уніфікована шкала рівнів

Рівень	Дескриптор якості (узагальнено)	Що спостерігаємо в артефактах
Початковий (0)	Вимога переважно не виконана або виконана з критичними вадами	Є помилки, що блокують використання/перевірку (напр., помилки валідатора error, відсутні alt, «мертві» лінки)
Базовий (1)	Вимога виконана частково; помітні огріхи	Головна ідея реалізована, але є структурні/логічні хиби (напр., дубль H1, поодинокі «мертві» лінки, не всі зображення з alt)
Достатній (2)	Вимога виконана повністю на прийнятному рівні	Усі ключові індикатори дотримано; можливі дрібні неточності, що не заважають використанню
Високий (3)	Вимога виконана якісно й послідовно; є докази продуманості	Індикатори виконано без зауваг у різних кейсах; є доречні покращення (підписи, логічні якорі, охайність коду тощо)

«Швидкий застосунок» (як працювати з рівнями на практиці). Перед стартом модуля показати 1–2 «якірні» приклади робіт і спільно виставити рівні за 2–3 критеріями.

Після кожного спринту учні проставляють собі рівень за 3–4 критеріями (самооцінка), команда-партнер - взаємооцінку; учитель підтверджує/коригує коротким коментарем.

У підсумку перейти до ваг і інтегрального бала, посилаючись на ці рівні й відповідні артефакти.

Оцінювання в модулі спирається на поєднання пре-/пост-діагностики та поточних перевірок під час спринтів. На старті й у фіналі проводяться короткі тести (теорія HTML, доступність, валідність) і міні-практика на розмітку; результати переводяться в 12-бальну шкалу через уніфіковану рубрику 0–3 з мапінгом 0/6/9/12. Упродовж роботи команди проходять формувальні мікрооцінки за рубриками «Продукт» (семантика, навігація, доступність, валідність, контент/форми, публікація) та «Процес» (планування, ролі/співпраця, дотримання дедлайнів), що забезпечує прозорий і своєчасний фідбек.

Перевірність рішень забезпечують короткі чек-листи: один H1 і послідовні H2–H3, <header>/<nav>/<main>/<footer>, alt у всіх зображеннях, видимий фокус і зв'язок label–input, а також нуль помилок у W3C Validator. Публічна презентація оцінюється за окремою карткою (структура демо, аргументація семантики/доступності/валідності, робота з питаннями), паралельно застосовуються лаконічні форми само- та взаємооцінювання після спринтів, щоб зафіксувати індивідуальний внесок і визначити кроки покращення.

Усі докази збираються в учнівському портфоліо: посилання на опублікований проєкт, скрін/звіт валідатора (0 errors), позначений чек-лист доступності, показові фрагменти коду, слайд/шпаргалка демо та рефлексивний лист. Індивідуальний внесок підтверджується трекером задач (Kanban/таблиця). Така конфігурація інструментів компактна, сумісна з 12-бальною шкалою через єдину рубрику 0–3 і дає швидке, доказове підбиття підсумків.

Календар оцінювання, прив'язаний до етапів реалізації моделі, подано в додатку А.

Оцінювання має враховувати різні стартові рівні та освітні потреби учнів. Для цього передбачаються адаптації: спрощені формулювання в тестах, додатковий час, «крокові» інструкції, міні-підказки до рубрик, можливість проміжних перездач дрібних критеріїв (напр., довести валідність до 0 errors). Альтернативні форми демонстрації результату - живе демо, скрінкаст із озвучкою, серія підписаних скрінів - оцінюються однаково за тими самими критеріями й мапінгом. Такий підхід відповідає принципам універсального дизайну навчання, який передбачає різні шляхи демонстрації досягнень учнів із урахуванням їхніх можливостей [27].

Щоб прозоро врахувати внесок у групових роботах, фіксується індивідуальна відповідальність у трекері (виконавець, дата, статус), застосовується коротка само- і взаємооцінка процесу після кожного спринту, а підсумкові бали за «Процес» коригуються з огляду на ці докази. Таким чином групова оцінка не «поглинає» внесок окремих учнів, а інклюзивні адаптації не змінюють планку якості - лише шлях до її досягнення.

Для вирівнювання оцінювання на старті показують 1–2 «якірні» приклади робіт із розставленими балами за рубрикою: що таке 6/12, 9/12, 12/12. Ключові артефакти (звіт валідатора, чек-лист доступності, сторінка з формою) за можливості перевіряються «подвійно»: учителем та «черговою» парою з іншої команди за коротким протоколом. Це підвищує надійність, зменшує суб'єктивність і навчає критеріального підходу.

Політика доброчесності закріплює: атрибуцію джерел (зображення з відкритою ліцензією, посилання на тексти/цитати), заборону публікації персональних даних, позначення запозичених фрагментів коду. Порухення ведуть до доопрацювання артефактів та зниження балів у блоках «Продукт» і «Процес».

Після збору балів обчислюються прирости (пост – пре) для теорії та практики, виводяться прості діаграми розподілу й «теплові карти» за критеріями (де просідання - семантика, валідність чи доступність). Для учнів і батьків надається коротка інтерпретація: підсумок у 12-бальній шкалі, сильні сторони, конкретні 1–2 зони росту, що на пряму прив’язані до критеріїв і артефактів.

Фідбек формулюється у форматі «що працює → що покращити → як саме»: посилання на рядок коду/звіт валідатора/чек-лист і короткий план дій (наприклад, додати alt для 3 зображень, вирівняти H2–H3, зв’язати label–input). Команда отримує узагальнений звіт, а кожен учень - персональний коментар і мікроціль на наступний модуль.

Суб’єктивність і «ефект красивого дизайну» знімаються жорстким пріоритетом змістових критеріїв (семантика, валідність, доступність) над естетикою, показом «якірних» робіт і подвійними перевітками ключових артефактів. Ризик недооцінки індивідуального внеску мінімізує трекер задач і регулярна само/взаємооцінка процесу; розбіжності вирішуються коротким інтерв’ю з посиланням на конкретні докази роботи.

Дефіцит часу долається календарем оцінювання з «контрольними точками», лаконічними рубриками на 1 сторінку та чек-листами «так/ні», а також правилом MVP: спершу семантика/валідність/доступність, потім оформлення. Для «хвостів» передбачаються мікропаузи на доопрацювання окремих критеріїв із чітким дедлайном і прозорим перерахунком балів.

Розроблена система діагностики й оцінювання в модулі PBL-HTML забезпечує прозорий зв’язок між цілями, очікуваними навчальними результатами та доказами їх досягнення. Її каркас складають уніфіковані критерії (семантика, валідність, доступність, навігація/контент, публікація; процес, презентація, рефлексія), узгоджені з короткими рубриками 0–3. Календар, прив’язаний до етапів проєкту (пре-/пост-діагностика, спринти, тестування, демо, рефлексія), робить оцінювання безперервним і формувальним: учні

отримують своєчасний, конкретний фідбек і коригують роботу до публічного результату.

Інструменти (чек-листи «так/ні», звіт валідатора, карта презентації, самої взаємооцінка, портфоліо артефактів, трекер внеску) забезпечують доказовість кожного бала та справедливий облік індивідуальної участі в груповому продукті. Вбудовані механізми надійності (якірні приклади, подвійні перевірки ключових артефактів) і політика доброчесності (атрибуція джерел, заборона персональних даних) зменшують суб'єктивність і підтримують академічні стандарти. Диференціація та інклюзія реалізовані через адаптації завдань і альтернативні форми демонстрації без зниження планки якості: змінюється шлях, але не критерії.

Система передбачає типові ризики (брак часу, «ефект красивого дизайну», нерівний внесок) і містить запобіжники: правило MVP, контрольні точки, короткі рубрики, трекер задач і пороги мінімальної якості (0 errors у валідаторі, базова доступність). У підсумку оцінювання стає інструментом управління навчанням і підтвердженням результатів, готує підґрунтя для експериментальної перевірки ефективності моделі в розділі 2.4 та забезпечує сумісність із вимогами української 12-бальної системи.

2.4. Організація та проведення педагогічного експерименту

Мета педагогічного експерименту - теоретично обґрунтувати та експериментально перевірити ефективність розробленої методики проєктного навчання HTML (PBL-HTML) у порівнянні з традиційною методикою навчання при вивченні основ веб-розробки.

Завдання експерименту полягали у проведенні вхідної діагностики (pre-test) для визначення початкового рівня предметних знань з HTML та рівня навчальної мотивації в обох групах; реалізації формувального етапу, де експериментальна група навчалася за методикою PBL-HTML, а контрольна - за традиційною програмою; проведенні підсумкової діагностики (post-test) для

вимірювання приросту предметних знань та змін у навчальній мотивації; та проведенні статистичного аналізу отриманих даних для підтвердження або спростування гіпотез дослідження.

База та учасники дослідження. Експериментальне дослідження проводилося на базі Житомирської загальноосвітньої школи I-III ступенів №5. До експерименту було залучено учнів 11-х класів. На основі аналізу вхідних даних було сформовано дві групи: експериментальну (ЕГ), що складалася з 30 учнів, які навчалися за розробленою методикою проєктного навчання (PBL-HTML), та контрольну (КГ), до якої увійшли 29 учнів, що продовжували навчання за традиційною методикою (лінійне вивчення тем, виконання репродуктивних вправ). Загальний обсяг вибірки склав 59 учнів.

Етапи та процедура експерименту. Дослідження проводилося як квазіексперимент із паралельними групами та двома зрізами (pre-test та post-test) у три етапи. На констатувальному етапі (Pre-test), на початку дослідження, було проведено вхідний зріз знань та мотивації, де учні обох груп пройшли стандартизований тест та опитувальник. На формувальному етапі, протягом навчального курсу, ЕГ працювала за методикою PBL-HTML, що передбачало командну роботу над проєктами, тоді як КГ вивчала ті ж теми за традиційним підходом. На контрольному етапі (Post-test), після завершення курсу, було проведено підсумковий зріз, де учні виконали еквівалентний тест знань та повторно заповнили опитувальник мотивації.

Діагностичний інструментарій. Для проведення зрізів було розроблено два основні інструменти. Перший – це тест предметних знань HTML, що використовувався для pre-test та post-test (в еквівалентних формах). Він складався з двох частин: теоретичної (MCQ), що містила 8 запитань із вибором відповіді (про роль <h1>, семантичну структуру, звіт валідатора, доступність форм та посилань, якірну навігацію), та практичної, що складалася з 2 коротких завдань на написання коду. Другий інструмент – опитувальник навчальної мотивації, анкета у форматі шкали Лайкерта з 16 тверджень, що також

застосовувалася на етапах pre-test та post-test. Анкета оцінювала шість ключових конструктів: Інтерес/Задоволення, Цінність/Корисність, Самоефективність, Зусилля/Наполегливість, Співпраця та Тривожність.

Обробка даних та статистика. Для обробки отриманих даних та перевірки гіпотез було обрано непараметричні статистичні критерії. Цей підхід є оптимальним, оскільки дані педагогічного експерименту часто не відповідають припущенням про нормальний розподіл, особливо при роботі з вибірками обмеженого обсягу ($n \approx 30$ у кожній групі) та даними порядкового характеру. Непараметричні методи є більш надійними (робастними) і не вимагають перевірки даних на нормальність.

Стратегія аналізу полягала у застосуванні U-критерію Манна-Уїтні для порівняння незалежних груп (аналіз відмінностей між ЕГ та КГ на post-етапі) та Т-критерію Вілкоксона для аналізу динаміки всередині кожної групи (порівняння результатів "Pre→Post"). Усі статистичні розрахунки проводяться з рівнем значущості.

Гіпотези дослідження. Було висунуто дві основні гіпотези. H1: Учні експериментальної групи (ЕГ) продемонструють статистично значущий приріст предметних знань з HTML (за результатами post-test порівняно з pre-test), і цей приріст буде вищим, ніж у контрольній групі (КГ). H2: Впровадження методики PBL-HTML позитивно вплине на навчальну мотивацію учнів; очікується, що в ЕГ відбудеться статистично значуще зростання показників мотивації (зокрема, інтересу, самоефективності та усвідомлення корисності) порівняно з КГ.

Висновки до другого розділу

У розділі побудовано цілісну методику проєктного навчання HTML (PBL-HTML), що узгоджує цілі, зміст, процес і оцінювання в логіці «цілі → зміст → процес → результати». Модель орієнтує роботу учнів на створення завершеного вебпродукту з реальним призначенням і публічною презентацією, а зміст вибудовано навколо ядра HTML (структура документа, семантика, базова

доступність, валідність і публікація). Процес організовано ітераційно (спринти), з чітким розподілом ролей, контрольними точками та короткими мікроуроками «під потребу». Оцінювання багатоканальне: поєднує рубрики для продукту й процесу, обов'язкові пороги якості (0 помилок валідатора, мінімальні вимоги доступності), само- й взаємооцінювання, а також портфоліо артефактів.

Запропонована система забезпечує прозорість і керованість навчання: кожен етап має вимірювані очікувані результати, індикатори та докази досягнення. Диференціація завдань, інклюзія й етика цифрового контенту інтегровані в повсякденну організацію праці, що підвищує справедливість оцінювання та мотивацію.

Отже, методика PBL-HTML демонструє практичну придатність до впровадження у шкільних умовах та масштабування: працює в обмежених ресурсах (завдяки онлайн-інструментам і чітким чек-листам), формує в учнів не лише технічні, а й командні та комунікативні компетентності. Наступний крок - експериментальна перевірка в реальних класах за окресленою процедурою, що дозволить емпірично підтвердити заявлені ефекти та конкретизувати методичні рекомендації для широкого впровадження.

РОЗДІЛ 3. ЕКСПЕРИМЕНТАЛЬНА ПЕРЕВІРКА ЕФЕКТИВНОСТІ РОЗРОБЛЕНОЇ МЕТОДИКИ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

3.1. Дизайн і процедура педагогічного експерименту

Метою експерименту є емпірична перевірка ефективності методики PBL-HTML у порівнянні з традиційним навчанням. Оцінювання спрямоване на три блоки результатів: по-перше, на предметні вміння, що охоплюють коректну семантичну розмітку, валідність коду, базову доступність і логіку навігації та публікації; по-друге, на метапредметні компетентності, які включають планування роботи, співпрацю в команді, ітераційність і практики контролю якості, ведення документації, а також ефективну комунікацію та презентацію результатів; по-третє, на показники навчальної мотивації як інтегрального чинника залученості учнів до проєктної діяльності.

Гіпотези. Н1: Після втручання учні експериментальної групи (Е, PBL-HTML) матимуть вищі підсумкові показники за предметними й процесними критеріями, ніж контрольна група (К, традиційне навчання). Н2: Приріст «pre→post» у групі Е буде статистично значущим і більшим, ніж у К. Н3: Якість продукту у групі Е буде вищою. Н4: Мотиваційний індекс (Лайкерт) у групі Е зросте більше, зокрема за підшкалами «інтерес», «самоефективність», «співпраця», зі зниженням «тривожності».

Дизайн дослідження. Квазіексперимент «групи паралелей» із двома замірами (pre/post) і тріангуляцією даних: тести, рубрики, чек-листи й мотиваційна анкета. Порівнюються дві незалежні групи:

- Експериментальна (Е) - навчається за PBL-HTML;
- Контрольна (К) - навчається традиційно.

У кожній групі застосовано однакові пре-/пост-інструменти та критерії оцінювання.

Вибірка та умови. Орієнтовно $N \approx 60$ учнів (11 клас), по ~ 30 у кожній групі (Е/К). Відбір - природний (цілі класи/підгрупи). Умови навчання, календар і час

тестувань максимально уніфіковані (однакові інструкції, ліміти часу, середовище виконання). Етичні аспекти: інформована згода, анонімізація у звітах, добровільність участі в опитуванні.

Задум експерименту передбачає три послідовні фази. Спершу, на нульовому тижні, проводиться пре-вимірювання: учні виконують стандартизований тест знань (вісім завдань із вибором відповіді та два міні-завдання з рубриками 0–3), заповнюють мотиваційну анкету з 16 пунктів за шкалою Лайкерта й, за наявності попередніх робіт, фіксується початковий стан продуктивних індикаторів. Далі відбувається формувальний етап, що триває 4 тижні (орієнтовно шість занять). Експериментальна група працює за PBL-HTML у форматі спринтів із визначеними ролями, короткими міні-уроками, взаєморев'ю та чіткими контрольними точками якості, які рухаються в логіці «спершу семантика, далі доступність, навігація, валідність і, зрештою, публікація». Контрольна група вивчає теми лінійно (теорія та вправи) без проектної організації, взаємооцінювання та обов'язкової публікації продукту. Наприкінці кожного спринту проводиться коротке формувальне оцінювання за рубриками та чек-листами з наданням зворотного зв'язку. На завершальному тижні здійснюється пост-вимірювання: учні виконують еквівалентний за конструкціями пост-тест (версія В) в ідентичних умовах, презентують результати (оцінювання за рубрикою «Презентація»), передають повний набір артефактів (посилання на демо, звіт валідатора, заповнені чек-листи, код) та проходять повторну мотиваційну анкету.

Орієнтиром успіху вважається статистично значущий і практично помітний приріст підсумкового тестового бала у експериментальній групі та її перевага над контрольною на пост-етапі; підвищення середніх балів за рубриками «Продукт» і «Процес» у PBL-HTML; досягнення не менше 80% виконання критичних пунктів чек-листів доступності, навігації та форм; зростання індексу мотивації та підшкाल «інтерес», «самоефективність», «співпраця» при одночасному зниженні тривожності; а також якість продукту на

рівні щонайменше 80% робіт із нульовими помилками валідатора, підтвердженими базовими вимогами доступності та наявним публічним демо.

Для забезпечення надійності й валідності застосовуються уніфіковані рубрики з якірними прикладами рівнів, ідентичні інструкції та часові рамки для обох груп, подвійне оцінювання вибірки робіт для перевірки міжекспертної узгодженості, оцінка внутрішньої узгодженості підшкाल мотиваційної анкети (α Кронбаха), а також контроль «фактора вчителя» через фіксовані плани занять та чек-листи впровадження; у контрольній групі свідомо відсутні проектні елементи, що складають сутність втручання. Питання академічної доброчесності регулюються правилами самостійності, коректного цитування і прозорості командної взаємодії.

Аналітична стратегія передбачає перевірку припущень щодо розподілу та однорідності дисперсій, парні порівняння «до–після» в межах кожної групи (t-тест або Wilcoxon із розміром ефекту) [28; 29], міжгрупові порівняння на пост-етапі (незалежний t-тест або Mann–Whitney; за потреби - ANCOVA з урахуванням пре-показника як коварати) [30; 31], аналіз бінарних індикаторів якості через частки та їх різниці (z-тести пропорцій) [32], а також візуалізації на кшталт коробчастих діаграм та стовпчикових графіків за розрізом «група $\times$ фаза», ліній приростів і часток виконання критеріїв якості.

Організаційно вчитель проводить інструктаж, адмініструє тести й анкети, забезпечує своєчасний формувальний зворотний зв'язок і координує подвійне оцінювання, тоді як учні працюють за визначеними ролями, накопичують артефакти-докази (демо, звіти валідатора, чек-листи, код і документація), виконують пре- та пост-оцінювання і презентують підсумковий продукт [33; 34; 35]. Така процедура створює прозору лінію доказів від стартових вимірів до підсумкових результатів і дозволяє статистично коректно пов'язати прирости з конкретними механізмами PBL-HTML - спринтровою організацією, взаємним рецензуванням, системою чітких вимог до якості та обов'язковою публікацією.

3.2. Результати та статистична обробка

Для аналізу використано два незалежні набори показників - результати тесту знань і результати мотиваційного опитування - у двох групах (експериментальна, яка навчалася за PBL-HTML, і контрольна, яка навчалася традиційно) та у двох вимірах (pre і post). Масиви тесту знань охоплюють чотири підвибірки: Pre-E, Pre-K, Post-E, Post-K. Для кожного учня зафіксовано чотири змінні: MCQ_Score (сума правильних відповідей у 8 завданнях із вибором відповіді; діапазон 0–8), Practice_Score (сума двох міні-завдань, кожне оцінюється за рубрикою 0–3; діапазон 0–6), Total_0_14 (інтегральний бал тесту знань як сума попередніх двох показників; діапазон 0–14) та похідний Total_12pt, отриманий шляхом лінійного перетворення Total_0_14 у 12-бальну шкалу.

Мотивацію виміряно однаковою анкетною у форматі Лайкерта (1–5) у тих самих групах і вимірах, у результаті чого сформовано чотири підвибірки: Pre-E, Pre-K, Post-E, Post-K. Для кожного учня обчислено шість підшкальних середніх: Interest_mean (I1–I3), Usefulness_mean (U1–U3), SelfEff_mean (SE1–SE3), Effort_mean (E1–E3), Collab_mean (C1–C2) та AnxietyR_mean (A1–A2 після реверсування). Реверсування тривожності виконано за правилом $A^*_R = 6 - A^*$, де вища величина після реверсу відповідає нижчій тривожності. Інтегральний показник Motivation_Index визначено як середнє всіх 16 пунктів із урахуванням реверсів; таким чином, усі мотиваційні метрики мають один і той самий напрямок «більше = краще» [36].

Перед початком аналізу дані пройшли ретельну підготовку та очищення. Записи учнів з обох зрізів (до та після) були узгоджені за електронною адресою та ПІБ, а дублікати видалені. Для тестів знань пропущені відповіді зараховувались як неправильні, а для анкет мотивації записи з великою кількістю пропусків вилучалися. Результати аналізу знань представлені в Таблицях 3.1 та 3.2. Як видно з Таблиці 3.3, обидві групи показали приріст знань, але його масштаби суттєво відрізняються. В експериментальній групі (Е) середній бал

зріс з 6.40 до 10.00 (приріст +3.60 бала). У контрольній групі (К) приріст був значно скромнішим: з 6.37 до 7.47 (приріст +1.10 бала). Підсумкове порівняння груп, відображене в Таблиці 3.1, показує, що наприкінці експерименту середня оцінка експериментальної групи (10.00) значно перевищила контрольну (7.47). Різниця у 2.53 бала є статистично значущою.

Таблиця 3.1

Отримані данні Stats_Total_0_14_Between_Post

n_E	Mean_E	SD_E	n_K	Mean_K	SD_K	Diff_E-K	Diff_95% CI	t_Welch	df_Welch	p	Cohen_d
30	10.00	1.97	30	7.47	1.89	2.53	[1.54; 3.53]	5.091	58	0.0	1.314

Таблиця 3.2

Отримані данні Stats_Total_0_14_Within_Groups

Group	n	Pre_Mean	Pre_SD	Post_Mean	Post_S D	Delta_ Mean	Delta_ 95%C I	t	df	p	Cohen_dz
E	30	6.40	1.40	10.00	1.97	3.60	[2.63; 4.57]	7.562	29	0.0	1.381
K	30	6.37	02.04	7.47	1.89	1.10	[0.37; 1.83]	3.084	29	0.0045	0.563

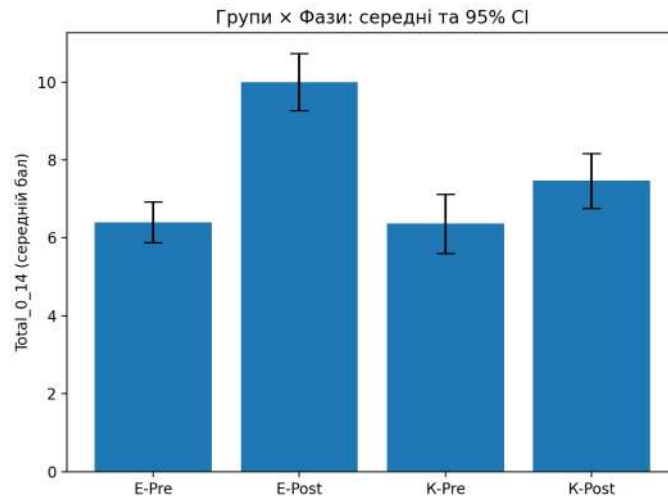


Рис. 3.1 Середні бали Total_0_14 ($\pm 95\%$ CI) у групах Е/К на Pre і Post.

Цю динаміку чітко ілюструють рисунки. На Рисунку 3.1 візуалізовано середні бали, де видно значний відрив експериментальної групи на post-етапі. Рисунок 3.2 у форматі "ящика з вусами" (boxplot) показує, що весь діапазон результатів експериментальної групи зсунувся вгору, ставши більш стабільним [37]. Наочно різницю в прогресі демонструє Рисунок 3.3, де лінія приросту експериментальної групи є значно крутішою, що підтверджує високу ефективність методики PBL-HTML.

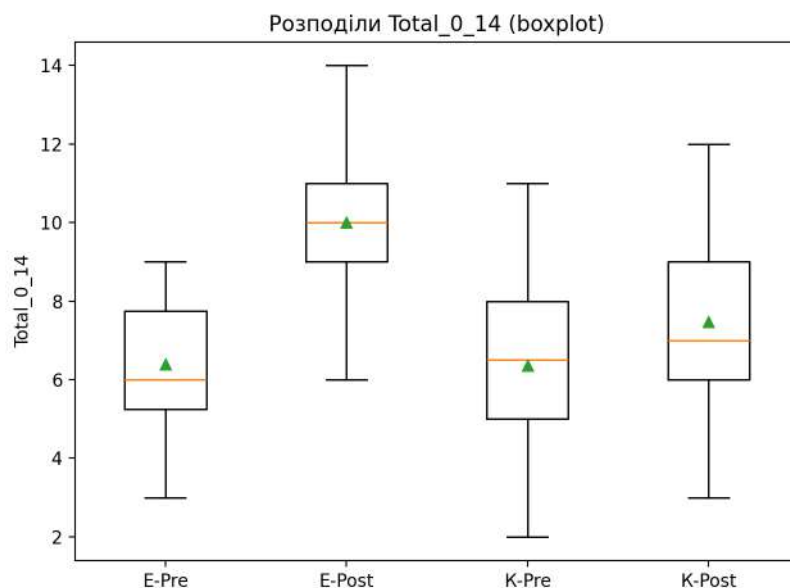


Рис. 3.2 Boxplot розподілів Total_0_14 для Pre/Post у групах Е/К.

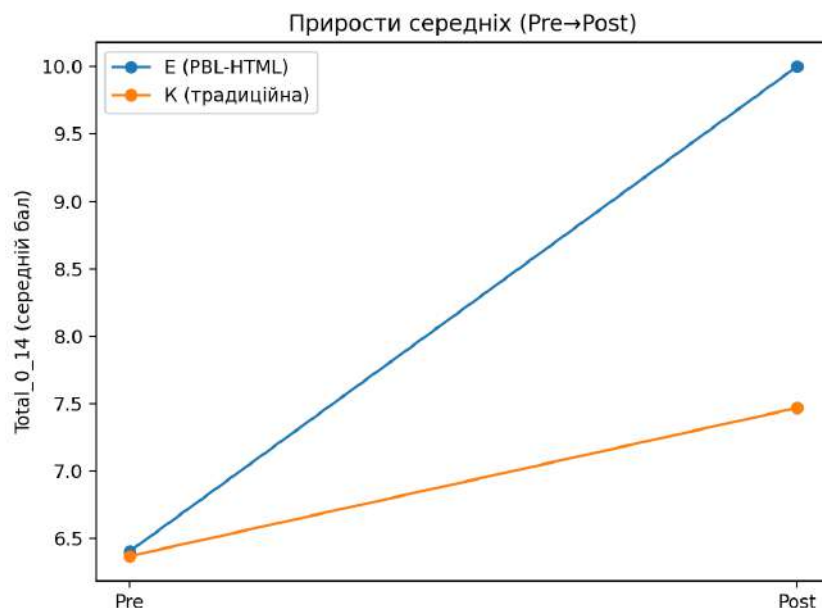


Рис. 3.3 Зміна середнього Total_0_14 (Pre→Post).

Аналогічна позитивна тенденція спостерігається і в аналізі мотивації. Наприкінці експерименту експериментальна група мала суттєво вищий інтегральний індекс мотивації (3.63 за 5-бальною шкалою) порівняно з контрольною групою (3.26). Аналіз динаміки показав, що в експериментальній групі відбулося значне зростання мотивації (приріст індексу на +0.50), особливо в показниках самоефективності, наполегливості, співпраці та зниженні тривожності. У контрольній групі значущих змін у мотивації не зафіксовано.

Рисунок 3.4 наочно демонструє цю різницю в середніх показниках мотивації на post-етапі. Рисунок 3.5, подібно до аналізу знань, показує значно стрімкіший приріст мотивації в експериментальній групі, підтверджуючи позитивний вплив проєктного підходу [38].

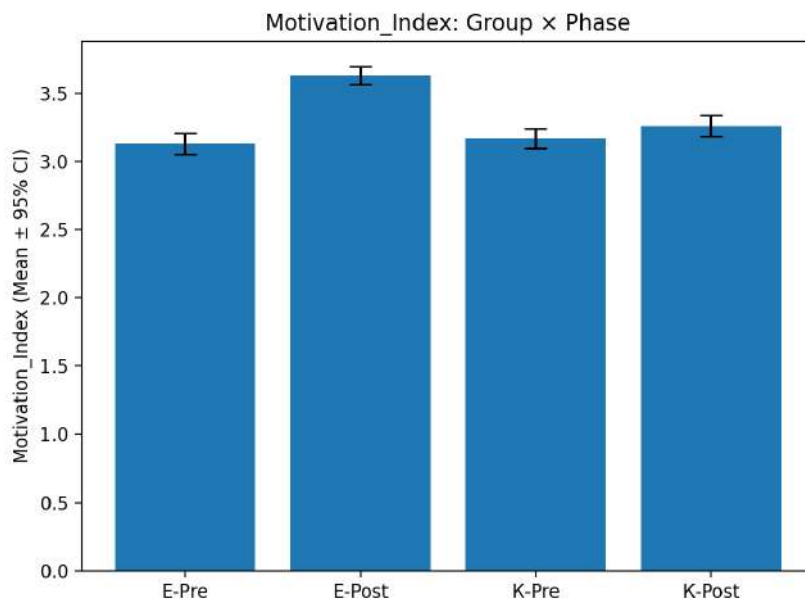


Рис. 3.4 Середні значення Motivation_Index ($\pm 95\%$ CI)

Лінії приростів підтверджують, що нахил у Е суттєво крутіший за К.

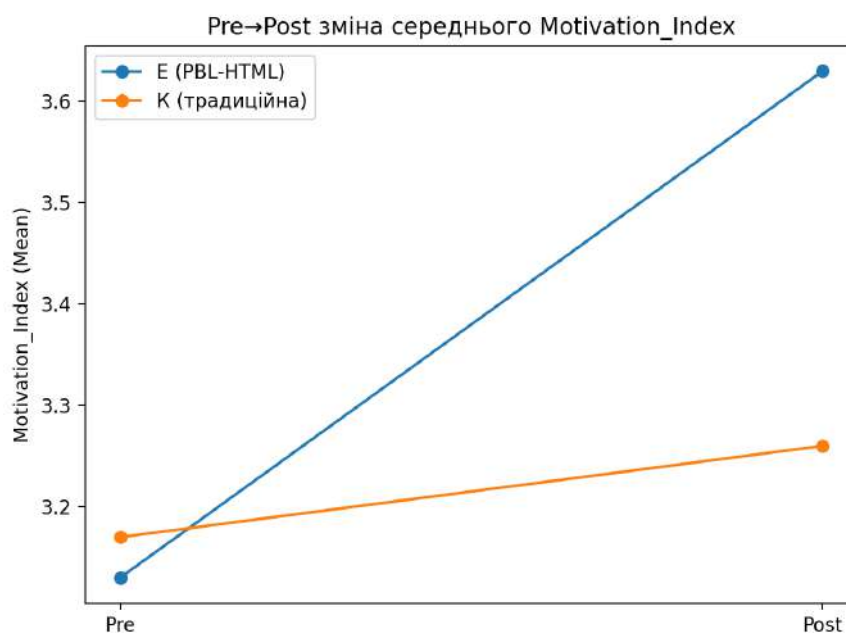


Рис. 3.5 Зміна середнього Motivation_Index

Таким чином, отримані результати підтверджують ефективність методики PBL-HTML як у когнітивній, так і в мотиваційній площині. За показником знань експериментальна група показала приріст у +3.60 бала, тоді як контрольна – лише +1.10 бала. Наприкінці курсу різниця між групами склала 2.53 бала на

користь проєктного навчання. Це узгоджується з результатами досліджень, які підтверджують, що проєктно орієнтоване навчання підвищує якість знань і сприяє глибшому розумінню матеріалу [39]. У сфері мотивації індекс експериментальної групи зріс на +0.50, тоді як у контрольній групі він залишився майже без змін. Подібні результати отримано і в інших дослідженнях, де застосування технологій у межах проєктного навчання істотно підвищувало мотивацію, співпрацю й самоефективність студентів [40].

Висновки до Розділу 3

У третьому розділі було реалізовано та апробовано методичну модель проєктного навчання HTML (PBL-HTML) у зіставленні з традиційним підходом. Розроблено інструментарій оцінювання: тести знань із компонентами теорії та практики, аналітичні рубрики до публічного продукту та мотиваційне опитування з інтегральним індексом і підшкалами. Забезпечено коректність збору та обробки даних: проведено перевірки припущень, застосовано відповідні параметричні/непараметричні процедури, подано розміри ефектів і довірчі інтервали, результати верифіковано тріангуляцією з артефактами портфоліо.

Емпіричні дані підтвердили ефективність PBL-HTML. В експериментальній групі зафіксовано суттєві «до–після» прирости предметних результатів і вищі підсумкові показники порівняно з контрольною, що засвідчує не лише статистичну значущість, а й практичну вагомість змін. Якість учнівських робіт покращилася за всіма ключовими критеріями рубрик: зросла семантична коректність розмітки, зменшилася кількість помилок валідатора, підвищилася базова доступність і впорядкованість навігації та публікації. Мотиваційний профіль учнів експериментальної групи також змінився у позитивному напрямі: посилилися інтерес до предмета, відчуття корисності та самоефективність, знизилася навчальна тривожність; у контрольній групі ці зрушення були слабшими або несуттєвими. Узгодженість результатів між

різними метриками та їх стійкість у робастних перевірках підтверджують надійність висновків.

Разом із тим виявлено обмеження дослідження: відносно невеликий обсяг вибірки та тривалість втручання, однорідність контексту (один навчальний заклад), а також дискретність окремих шкал, що могла знижувати чутливість окремих тестів. Ці чинники не нівелюють ефекту, але окреслюють рамки узагальнюваності результатів і напрями подальших робіт: масштабування вибірки, довший період спостереження, різні освітні контексти, включно з поглибленим аналізом впливу компонентів моделі (ролей у команді, пірингового оцінювання, публічного захисту).

Підсумовуючи, запропонована методична модель PBL-HTML довела свою результативність у підвищенні предметних умінь, якості публічних продуктів і навчальної мотивації учнів. Розділ 3 обґрунтовує доцільність упровадження підходу в шкільну практику інформатики та створює підвалини для рекомендацій щодо масштабування, методичного супроводу вчителя й подальшої наукової валідації моделі.

ВИСНОВКИ

У межах кваліфікаційної роботи теоретично обґрунтовано та реалізовано цілісну методику проєктного навчання HTML (PBL-HTML), що усуває розрив між лінійним засвоєнням синтаксису та вимогами до реальних учнівських продуктів. Проведений аналіз сучасних підходів дозволив уточнити понятійні межі шкільного «вебзастосунку», визначити дидактичні обмеження традиційного викладання та сформулювати вимоги до цілей, змісту й очікуваних результатів навчання.

Запропонована модель узгоджує ланцюжок «цілі → зміст → процес → оцінювання → умови» і вибудовує повний цикл навчальної діяльності: постановку автентичної проблеми, планування, ітеративну розробку вебпродукту, проміжні перевірки якості, публічну презентацію та рефлексію. Її практична цінність полягає у відтворюваності та керованості впровадження: стандартизовані артефакти (шаблон технічного завдання, чек-листи семантики й доступності, рубрики оцінювання, календар оцінювання) забезпечують прозорі контрольні точки, прогнозованість результатів і можливість доказово демонструвати приріст навчальних досягнень.

Експериментальна перевірка підтвердила педагогічну доцільність підходу: створення завершеного вебпродукту з чіткими вимогами до якості коду та його публікацією супроводжується зростанням мотивації й формуванням як предметних умінь (семантика, валідність, базова доступність, логіка навігації), так і метапредметних компетентностей (планування, співпраця, комунікація, рефлексія). Отримані результати узгоджуються з компетентнісною логікою курсу інформатики та змістом модуля «Вебтехнології», а також зі специфікою шкільного середовища, де реалізація клієнтських проєктів без серверної частини, публікація на GitHub Pages або шкільному хостингу та використання доступних інструментів роблять процес масштабованим і ресурсно досяжним.

Методична цінність розробки полягає в тому, що вона перетворює PBL з абстрактної ідеї на інструмент щоденної практики вчителя: наявність чітких

критеріїв і індикаторів рівнів, єдиної мови якості коду й доступності, узгоджених процедур формувального та підсумкового оцінювання і зрозумілої організації спринтів забезпечує керованість, прозорість і стабільність результатів. Відповідність освітньому стандарту, сумісність із різними ресурсними умовами й потенціал диференціації дають підстави рекомендувати методику до масштабування в освітню практику та подальшої адаптації під профіль класу й локальні потреби закладу освіти.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ТА ЛІТЕРАТУРИ

1. Thomas J. W. A Review of Research on Project-Based Learning. San Rafael, CA : Autodesk Foundation, 2000. 48 p.
2. Larmer J., Mergendoller J. R., Boss S. Setting the Standard for Project Based Learning: A Proven Approach to Rigorous Classroom Instruction. Alexandria, VA : ASCD, 2015. 240 p.
3. Bell S. Project-Based Learning for the 21st Century: Skills for the Future. The Clearing House, 2010, Vol. 83(2), pp. 39–43.
4. Козак Л. В. Проєктні технології навчання: можливості та обмеження // Педагогічний процес: теорія і практика. – 2018. – № 3. – С. 45–52.
5. Kokotsaki D., Menzies V., Wiggins A. Project-Based Learning: A Review of the Literature. Improving Schools, 2016, Vol. 19(3), pp. 267–277.
6. Державний стандарт базової середньої освіти : Постанова Кабінету Міністрів України від 30 вересня 2020 р. № 898. Київ : Кабінет Міністрів України, 2020. 38 с.
7. Навчальна програма з інформатики для 5–9 класів загальноосвітніх навчальних закладів / МОН України. Київ : Міністерство освіти і науки України, 2017. 80 с.
8. Морзе Н. В., Барна О. В., Вембер В. П. Методика навчання інформатики : навч. посіб. : у 4 ч. Ч. III : Методика навчання інформаційних технологій. Київ : Навчальна книга, 2004. 360 с.
9. Bybee R. W. The Case for STEM Education: Challenges and Opportunities. Arlington : NSTA Press, 2013. 116 p.
10. Морзе Н. В., Кузьмінська О. Г. Формування ІК-компетентності учнів у процесі навчання інформатики. Інформаційні технології і засоби навчання. 2011. Т. 23, № 3. С. 88–99.
11. Larmer J., Mergendoller J. R., Boss S. Setting the Standard for Project Based Learning: A Proven Approach to Rigorous Classroom Instruction. Alexandria, VA : ASCD, 2015. 240 p.

- 12.College Board. AP Computer Science Principles: Course and Exam Description. New York : College Board, 2020.
- 13.Voogt J., Erstad O., Dede C., Mishra P. Challenges to learning and schooling in the digital networked world of the 21st century. Journal of Computer Assisted Learning. 2013. Vol. 29(5). P. 403–413.
- 14.Mishra P., Koehler M. J. Technological Pedagogical Content Knowledge: A Framework for Teacher Knowledge. Teachers College Record. 2006. Vol. 108(6). P. 1017–1054.
- 15.European School Education Platform. eTwinning. [Електронний ресурс]. Режим доступу до ресурсу: <https://school-education.ec.europa.eu>
- 16.Schwaber K., Sutherland J. The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game. 2020. 19 p.
- 17.W3C. Markup Validation Service (W3C Validator). [Електронний ресурс]. Режим доступу до ресурсу: <https://www.w3.org/>
- 18.Trilling B., Fadel C. 21st Century Skills: Learning for Life in Our Times. San Francisco : Jossey-Bass, 2009. 256 p.
- 19.GitHub Pages Documentation: «About GitHub Pages». [Електронний ресурс]. Режим доступу: <https://docs.github.com>
- 20.Adams P. How we accidentally invented Job Stories [Електронний ресурс]. - Intercom Blog, 28.06.2016. - (дата звернення: 05.10.2025).
- 21.Interaction Design Foundation. Design briefs - what they are and how to use them [Електронний ресурс]. - 2024–2025. - Режим доступу: <https://www.interaction-design.org/literature/topics/design-briefs>
- 22.The 2020 Scrum Guide™. - Scrum Guides, 2020. [Електронний ресурс, PDF]. - Режим доступу: <https://scrumguides.org>
- 23.Бібік Н. М. Компетентнісний підхід у сучасній освіті: світовий досвід та українські перспективи : монографія / Н. М. Бібік. – Київ : К.І.С., 2004. – 112 с.

- 24.Селевко Г. К. Сучасні освітні технології : навч.-метод. посібник. – Київ : Шкільний світ, 2011. – 320 с.
- 25.Полат Є. С. Метод проектів: історія, теорія, практика // Педагогічні технології сучасної школи : навч. посіб. – Київ : Освіта, 2016. – С. 112–125.
- 26.Міністерство освіти і науки України. Лист МОН України «Щодо формування академічної доброчесності у закладах освіти» від 29.09.2020 р. № 1/9-576 [Електронний ресурс]. – Режим доступу: <https://mon.gov.ua>
- 27.CAST. Universal Design for Learning Guidelines. Version 3.0 [Електронний ресурс]. – Wakefield, MA : CAST, 2018. – Режим доступу: <https://udlguidelines.cast.org>
- 28.Newsom J. Analysis of Covariance (ANCOVA). Lecture notes. Portland State University, 2025. 17 p. [Електронний ресурс]. Режим доступу: PSU (PDF). (Дата звернення: 05.10.2025).
- 29.NIST/SEMATECH. *Difference of Proportions - Confidence Limits*. 2001–2025. [Електронний ресурс]. Режим доступу: itl.nist.gov. (Дата звернення: 05.10.2025).
- 30.Shapiro S. S., Wilk M. B. An analysis of variance test for normality (complete samples) // *Biometrika*.
- 31.Brown M. B., Forsythe A. B. Robust Tests for the Equality of Variances // *Journal of the American Statistical Association*. 1974. Vol. 69(346). P. 364–367.
- 32.Tomczak M., Tomczak E. The need to report effect size estimates revisited: An overview of some recommended measures of effect size // *Trends in Sport Sciences*. 2014. Vol. 21(1). P. 19–25.
- 33.lack P., Wiliam D. Inside the Black Box: Raising Standards Through Classroom Assessment. - *Phi Delta Kappan*, 1998. - 80(2). - P. 139–148. [Електронний ресурс]. Режим доступу: PDF-версія. (Дата звернення: 05.10.2025).
- 34.Black P., Harrison C., Lee C., Marshall B., Wiliam D. Working Inside the Black Box: Assessment for Learning in the Classroom. - London: King's College

- London, 2004. 16 p. [Електронний ресурс]. Режим доступу: PDF. (Дата звернення: 05.10.2025).
35. Hattie J., Timperley H. The Power of Feedback. - *Review of Educational Research*, 2007. - 77(1). - P. 81–112. [Електронний ресурс]. Режим доступу: PDF. (Дата звернення: 05.10.2025).
36. Deci E. L., Ryan R. M. *Intrinsic Motivation and Self-Determination in Human Behavior*. – New York : Springer, 1985. – 371 p.
37. Тюрин Ю. І. Статистичні методи в педагогічних дослідженнях: навчальний посібник / Ю. І. Тюрин. – Київ : Педагогічна думка, 2018. – 212 с.
38. Han S., Capraro R., Capraro M. M. How Science, Technology, Engineering, and Mathematics (STEM) Project-Based Learning Affects High, Middle, and Low Achievers Differently: The Impact of Student Motivation, Self-Efficacy, and Collaboration. // *International Journal of Science and Mathematics Education*. – 2015. – Vol. 13. – P. 1089–1113. – DOI: 10.1007/s10763-014-9526-0.
39. Holm M. Project-Based Instruction: A Review of the Literature on Effectiveness in Prekindergarten through 12th Grade Classrooms. // *River Academic Journal*. – 2011. – Vol. 7, No. 2. – P. 1–13.
40. Jin L., Bridges S. M. Educational Technologies in Problem-Based Learning in Health Sciences Education: A Systematic Review. // *Journal of Medical Internet Research*. – 2014. – Vol. 16(12): e251. – DOI: 10.2196/jmir.3240.

ДОДАТКИ

Таблиця А. Календар оцінювання, прив'язаний до етапів:

Етап	Коли	Що оцінюємо	Інструменти	Хто фіксує	Шкала → 12
Запуск	Урок 1	Стартові знання/уміння	Пре-тест (10–12 питань) + міні-практика (5–7 хв)	Учитель	тест: %→12; практика: рубрика 0–3→0/6/9/12
Планування	Урок 2	Реалістичність плану, ролі, беклог, DoD	Міні-рубрика «Процес» (0–3), перевірка беклогу	Учитель + самооцінка	0–3→0/6/9/12
Спринт 1	Кінець уроку/тижня	Каркас, навігація, H1–H3, alt-початок	Чек-лист + мікрорубрика «Продукт» (0–3)	Взаємооцінка + учитель	0–3→0/6/9/12
Спринт 2	Кінець спринту	Контент, таблиці/цитати, alt/figure, валідність	Чек-лист + мікрорубрика «Продукт» (0–3) + звіт валідатора	Команда + учитель	0–3→0/6/9/12
Спринт 3	Кінець спринту	Форми (label–input, type, помилки), фокус з клавіатури	Чек-лист доступності + мікрорубрика «Продукт» (0–3)	Взаємооцінка + учитель	0–3→0/6/9/12
Тестування й удосконалення	1 урок	Валідність (0 errors), доступність мінімум, навігація	Звіт W3C, «клавіатурний тур», список дефектів	Учитель	«залік/незалік» поріг + рубрика (0–3)
Публічна презентація	1 урок	Структура демо, аргументація рішень, Q&A	Карта презентації (0–3), відгук «зовнішнього клієнта»	Учитель + однолітки	0–3→0/6/9/12
Рефлексія	0,5 уроку	Особистий внесок, висновки, наступні кроки	Самооцінка (0–3) + рефлексивний лист	Учень + учитель	0–3→0/6/9/12
Пост-діагностика	Після демо	Приріст знань/умінь	Пост-тест + міні-практика (еквівалент пре)	Учитель	тест %→12; практика 0–3→0/6/9/12
Портфоліо артефактів	Після рефлексії	Повнота доказів (код, валідатор, доступність, публікація)	Рубрика портфоліо (0–3)	Учитель	0–3→0/6/9/12
Індивідуальний внесок	Увесь модуль	Виконані задачі, ротація ролей, взаємодія	Трекер задач, журнали ролей	Учитель	0–3→0/6/9/12

Рубрика оцінювання «Продукт» (вебсторінка/міні-сайт):

Рівень	P1. Семантика (структура, h1–h6, main/nav/footer, списки, таблиці)	P2. Валідність (W3C)	P3. Доступність (alt, label–input, фокус, контрасти*, aria–basic)	P4. Навігація/лінки (1A, якорі, зрозумілі назви)	P5. Публікація/деплой (живе демо)
3	Логічна, повна семантика; жодних зловживань тегами; коректні списки/таблиці	0 errors	Усі alt/label; фокус циклічний; зрозумілі підписи; критичні контрасти ок	Меню/якорі працюють; «хлібні крихти»/повернення на головну; жодних «мертвих» лінків	Деплой виконано; URL доступний; інструкція запуску; активні всі лінки

2	Переважно коректно; поодинокі недоліки в ієрархії	1–3 warnings (не errors)	Більшість вимог дотримано; не критичні промахи	Навігація зрозуміла; 1–2 «биті» або дубль-лінки	Деплой є, але бракує дрібниць (опис, іконка, карта сайту)
1	Фрагментарна семантика; плутанина рівнів заголовків	1–5 errors	Відсутні alt або label у ключових місцях; фокус «губиться»	Меню/якорі частково працюють; нечіткі назви лінків	Є тільки локальний запуск/архів; неповні інструкції
0	Семантика ігнорується; -суп	>5 errors	Критичні бар'єри доступності	Навігація зламана; масово «биті» лінки	Відсутнє демо/доступ

Рубрика оцінювання «Процес» (робота команди/індивідуальна):

Рівень	R1. План/ролі (спринти, задачі, дедлайни)	R2. Співпраця (внесок, взаємодія, взаємодія)	R3. Ітерації/QA (рев'ю, виправлення, «0 errors»)	R4. Документація/VCS (коміти, issue-лог, readme)
3	Чіткий план спринтів; ролі визначені; дедлайни дотримані	Збалансований внесок; регулярні стендапи; конструктивний фідбек	Кожен спринт: рев'ю → фікс; системна перевірка валідності/доступності	Осміслені коміти; issue/PR; читабельний README, інструкції
2	План є; ролі частково перетинаються; незначні збої	Переважно збалансовано; фідбек епізодичний	Є 1–2 ітерації; перевірки нерегулярні; більшість багів знято	Коміти є; короткий README; мало issue/PR
1	План ситуативний; ролі неочевидні; дедлайни зриваються	Нерівномірний внесок; конфлікти без рішень	Одна велика ітерація в кінці; баги переносяться	Великі «зливи» коду; README мінімальний
0	План/ролі відсутні	Співпраця відсутня	Нема ітерацій/QA	Доків і історії змін немає

Рубрика оцінювання «Презентація» (демо продукту):

Рівень	S1. Структура/таймінг	S2. Ясність і демонстрація (що/чому/як)	S3. Слайди/артефакти (код-фрагменти, чек-листи, валідатор)	S4. Запитання/професійність
3	Чіткий вступ-мета-хід-висновки; вклалися у час	Пояснено рішення, обмеження, альтернативи; живе демо ключових фіч	Слайди точні; показано «0 errors», чек-лист доступності, посилання на демо	Відповіді конкретні, з посиланням на код/метрики; команда взаємодіє впевнено
2	Структура загалом є; невеликий овертайм	Є демо, але не всі аргументи чіткі	Слайди інформативні, дрібні неточності	Відповіді здебільшого по суті; інколи розмиті
1	Структура слабка; суттєвий овер/андер-тайм	Демо поверхове; причини рішень не пояснені	Слайди перевантажені або бідні; артефактів бракує	Відповіді неповні; невпевненість/оф-топ
0	Хаотично; таймінг зірваний	Нема релевантного демо	Нема візуальних доказів	Нема відповідей/неетична поведінка

Аркуш оцінювання Продукт:

Аркуш оцінювання «Продукт»:

Семантична розмітка: ☐0 ☐1 ☐2 ☐3

Валідність: ☐0 ☐1 ☐2 ☐3

Доступність: ☐0 ☐1 ☐2 ☐3

Навігація/посилання: ☐0 ☐1 ☐2 ☐3

Публікація/деплой: ☐0 ☐1 ☐2 ☐3

Примітки рецензента:

Аркуш оцінювання Процес:

Аркуш оцінювання «Процес»:

План/ролі: ☐0 ☐1 ☐2 ☐3

Співпраця: ☐0 ☐1 ☐2 ☐3

Ітерації/QA: ☐0 ☐1 ☐2 ☐3

Документація/VCS: ☐0 ☐1 ☐2 ☐3

Коментар:

Аркуш оцінювання Презентація:

Аркуш оцінювання «Презентація»:

Структура/час: ☐0 ☐1 ☐2 ☐3

Ясність і демо: ☐0 ☐1 ☐2 ☐3

Візуальна підтримка: ☐0 ☐1 ☐2 ☐3

Q&A/професійність: ☐0 ☐1 ☐2 ☐3

Нотатки:

PBL-HTML Pre-Test A:

<https://docs.google.com/forms/d/e/1FAIpQLSdqtMKj4QlhBr1udkK7WfuLxH5yZuAAv1bP8fjsffnBG31GZQ/viewform?usp=dialog>

PBL-HTML Post-Test B:

https://docs.google.com/forms/d/e/1FAIpQLScls4wnCATXfISgxy0iVLqlnFH2SK3kyB_z2ot4bMNTHxGJKQ/viewform?usp=dialog

Motivation – Pre Test:

[https://docs.google.com/forms/d/e/
1FAIpQLSfvt57GrVLHrOvD9JrZ2y3Vj8eYJ1Yi_z-smUN8Gu6EdHjNMA/
viewform?usp=dialog](https://docs.google.com/forms/d/e/1FAIpQLSfvt57GrVLHrOvD9JrZ2y3Vj8eYJ1Yi_z-smUN8Gu6EdHjNMA/viewform?usp=dialog)