

А.Л. Федорчук<sup>1</sup>, О.Ю. Усата<sup>1</sup>, О.А. Наконечна<sup>2</sup>

<sup>1</sup>Житомирський державний університет імені Івана Франка, Україна

<sup>2</sup>Одеський державний аграрний університет, Україна

## WEB-ДИЗАЙН ТА WEB-ПРОГРАМУВАННЯ У СУЧАСНОМУ ІНТЕРНЕТ-СВІТІ

Досліджуються особливості веб-програмування та веб-дизайну з поєднанням сучасних можливостей Інтернет-середовища. Аналіз проводиться з дослідженням можливостей використання фреймворку Bootstrap, який об'єднує мову гіпертекстової розмітки HTML5, каскадні таблиці стилів CSS3 та мову програмування JavaScript. З використанням мови програмування, її коду та набору допоміжних інструментів при розробці веб-сайту у роботі наводиться алгоритм розробки.

**Ключові слова:** Веб, веб-середовище, веб-технології, фронтенд, фреймворк, Bootstrap, CSS3, HTML5, CCS3, JavaScript.

### Вступ

На сьогодні Інтернет є важливим інструментом, який охоплює громадянські потреби та надає корисні можливості для бізнесу та державних установ. З моменту його появи відбулася низка значних змін, що обумовлюється переходом від найпростішої парадигми Веб 1.0 (яка характеризується статистичною публікацією контенту, в якій користувачі мають змогу лише споживати контент з відсутністю двостороннього зв'язку між користувачами) до Веб 4.0 та Веб 5.0 (які виникли завдяки технологічному прогресу мобільних пристроїв, Інтернету, речей та спеціальних мереж й характеризуються симбіотичною взаємодією між людьми та машинами) [1].

Сучасний веб стає все більше інтерактивним та відкриває безліч можливостей шляхом комбінування двох ключових аспектів – веб-програмування та веб-дизайну, які взаємодіють між собою для створення функціональних веб-додатків з більш розширеними візуальними можливостями для користувачів [2]. У наш час веб-програми можуть виконувати багато роботи, окрім простого відображення статичних даних. Сучасні веб-додатки стали важливою частиною щоденної екосистеми через потреби клієнтів [3].

Одним із факторів, що впливають на оптимізацію попередньої веб-технології, є вибір правильної бібліотеки та передньої частини. Загалом бібліотеки та фреймворки JavaScript відіграють важливу роль з точки зору масштабованості, обслуговування, модульності та підвищення інтерактивності веб-сторінок [3]. Сукупність веб-технологій, які можуть складатися з інформаційних веб-сайтів, веб-сторінок та веб-додатків, як правило, використовуються з новітніми динамічними веб-технологіями для дизайну з метою забезпечення взаємодії з користувачем [4].

У веб-програмуванні HTML є однією з найпоширеніших мов веб-програмування, яка містить

безліч корисних характеристик та можливостей, наприклад, простота та зручність у використанні, швидкість та можливість перегляду на різних платформах. Процес розробки переважно відбувається у браузері, який може містити зручну кросплатформену трансплантацію, складається з вільного доступу клієнтського браузера, ґрунтовно вирішує проблему з різницею в обладнанні розробки, а також підтримує API, CSS3 та логіку програмування [5].

Розвиток інформаційних технологій посприяв значному удосконаленню веб-технологій, де сучасна версія HTML 5.0 на відміну від попередніх версій має змогу перетворювати веб у зрілу платформу для додатків, на якій можна розміщувати різноманітний контент, наприклад, відео, аудіо, зображення, анімації, де взаємодія з комп'ютером цілком стандартизована. Нове покоління технологій HTML 5.0 швидко стало популярним в прикладному програмному забезпеченні браузерів і мережі для управління, що дозволило отримати нові розробки, які допомогли вирішити недоліки попередніх версій [6].

У випадках, коли на веб-сторінці представлена базова структура, яка потребує стилізації та оформлення, під час веб-розробки використовуються інструменти, які допомагають додавати кольори, стиль дизайну та індивідуальний стиль до веб-сайтів. Одним з таких інструментів є каскадні таблиці стилів (CSS), за допомогою чого можна задавати колір, розміри та позиціонування частин веб-сторінки [5–7]. Але існують також більш розвинені та ефективніші методи CSS, які мають змогу стилізувати кожну частину веб-сторінки. Існує три способи застосування CSS під час створення веб-сторінки, які можуть бути вбудованими, внутрішніми або зовнішніми. Додавання атрибуту style до будь-якого об'єкта дає можливість написати власні правила стилізації [8].

Використання фреймворків є абстрактною вер-

цією реалізації, що використовується для допомоги розробникам під час створення веб-сторінок та веб-сайтів. Фреймворки мають визначені класи, які використовуються разом з HTML-кодом. Одним з таких фреймворків є Bootstrap, який є одним з найпопулярніших фреймворків CSS [9]. Bootstrap містить відкритий вихідний код, який розміщений на сервісі обміну кодом та управління версіями GitHub. Окрім цього існують також інші фреймворки, наприклад, Foundation, який орієнтований на адаптивний веб-дизайн; Gumbo, який побудований з коду для розробників-дизайнерів та файлів, за допомогою чого створюється середовище, в якому дизайнери та розробники мають змогу працювати разом з інструментами; Ingrid, який призначений для зменшення кількості класів на об'єктах, що містить дві точки зупинки – одна для мобільних телефонів, а інша для планшетів, але такий фреймворк не містить жодного стилю й характеризується адаптивною сітковою системою, яка є менш функціональною [4, 10].

### Постановка проблеми

Сучасні можливості веб-технологій пройшли через декілька ключових етапів, за допомогою яких під час процесу реалізації можна було виявити цілий ряд недоліків, які з часом дозволили покращити умови створення якісного веб-контенту. Однак для того, щоб проаналізувати сучасні можливості веб-програмування та відношення сучасного Інтернет-середовища, необхідно насамперед звернути увагу на характерні зміни інформаційних технологій, які посприяли створенню еволюції веб-технологій. Посилаючись на проаналізовані джерела, нами розглянуто ключові аспекти еволюції веб-технологій, які містять ряд переваг та недоліків, на які необхідно звернути увагу під час веб-програмування.

Веб-розробка характеризується фронтенд- та бекенд-розробкою, які один від одного відрізняються мовою програмування з використанням відповідних фреймворків, де виникає необхідність описати ключові особливості при фронтенд-розробці, якщо мова стосується графічного інтерфейсу та дизайну. Використання застарілих комп'ютерів та мобільних пристроїв можуть впливати на відображення веб-контенту, а в деяких випадках унеможлилювати роботу через припинення оновлення браузерів, програмного забезпечення та припинення обслуговування. Перші з версій HTML та CSS були невибагливими у використанні, але містили ряд недоліків, що пов'язане з розробкою веб-контенту, через обмежені можливості. Сучасні технології демонструють значно покращені можливості та перспективи на майбутнє у еволюції веб-технологій, що обумовлюється розвитком штучного інтелекту та використанням Інтернету речей.

Для того, щоб з'ясувати взаємозв'язок між веб-

технологіями та Інтернет-середовищем, у статті розглядалися питання комбінування мови гіпертекстової розмітки HTML5 з каскадними таблицями стилів CSS3, які містяться у фреймворку Bootstrap 5, де особливу увагу було приділено характеристикам та перевагам з недоліками різних версій. В якості порівняння розглядалися ранні версії Bootstrap 3.0 та 5.0, в яких використовуються HTML5 та CSS3. Тому актуальною проблемою даного дослідження є порівняльний аналіз, в якому наводиться характеристика можливостей застарілих версій з сучасними версіями веб-програмування, що дозволяє розробляти та вдосконалювати веб-дизайн та інтерфейс для користувачів у сучасному Інтернеті.

### Аналіз останніх досліджень і публікацій

Оскільки останнім часом Інтернет-технології продовжують розвиватися, дизайн веб-сторінок перейшов від тексту та статичних зображень до динамічних взаємодій, які можуть створюватися за допомогою: мови розмітки HTML5 для відображення веб-сторінок; стилізації та оформлення CSS3 з використанням мови програмування JavaScript; фреймворків, які можуть значно полегшувати та прискорювати розробку веб-сервісів в Інтернеті [11]. У роботі [12] авторами досліджувалося використання програми на основі HTML5, CSS3 та JavaScript для створення веб-сайту, який зміг би забезпечити проведення психологічних експериментів, де HTML5 використовувався для створення елементів дизайну веб-сторінок, а CSS3 слугували в якості кольорового дисплея, щоб дослідити афективну реакцію користувачів на різні зразки кольорів та надати рекомендації дизайнерам щодо покращення веб-сторінок. Однак попередні версії CSS на відміну від CSS3 виражали візуальний дизайн веб-сайту через код й залишалися недостатньо вивченою областю веб-технологій. Незважаючи на те, що CSS3 запропоновано в якості методу додавання шару дизайну до HTML5 документів, вони перейшли з маргінальної позиції до основного використання лише після тривалого часу, коли веб-дизайнерам доводилося працювати над розробкою веб-стандартів.

Безліч робіт приділялося дослідженням можливостей CSS, наприклад, у роботі [13] авторами було з'ясовано, що CSS здатні формувати міст між дизайном та кодом в Інтернеті, зводячи естетичний вигляд веб-сайтів до серії «властивостей» і «значень», розглянувши можливість підходу до веб-дизайну не через інтерфейс, а через текстовий редактор. Однак розвиток та вдосконалення веб-технологій відкрило нові можливості для електронної комерції, що дозволило надавати послуги з продажу товарів. У роботі [14] автором розроблений веб-сайт електронної комерції відносин, використовуючи технологію фронтенд, яка містить HTML,

CSS та JavaScript. Однак варто розрізняти дві різні веб-технології під час розробки веб-сайту, а саме фронтенд (публічна частина веб-додатку або веб-сайту, з якою користувач може взаємодіяти та контактувати напряму) та бекенд (внутрішня, прихована від користувача оболонка сайту або веб-додатку, яка працює на дистанційному сервері, а не в браузері ПК), які відрізняються між собою.

У роботі [14] автором описуються коди для веб-програмування: HTML код – код домашньої сторінки, в якому можна редагувати його у відповідності з особистими вимогами; CSS код – код всього сайту та сторінки входу в систему окремо, де окрім сторінки входу в систему можна редагувати весь CSS сайт; JavaScript код – код розділу «Інформація про продукт».

### Мета дослідження

Метою даного дослідження є вивчення та аналіз ключових аспектів веб-програмування та веб-дизайну з поєднанням сучасних можливостей Інтернет-середовища. Для проведення аналізу веб-дизайну проаналізовано можливості використання фреймворку Bootstrap, який об'єднує та містить в собі мову гіпертекстової розмітки HTML5, каскадні таблиці стилів CSS3 та мову програмування JavaScript. Для досягнення поставленої мети необхідно проаналізувати: характерні особливості під час фронтенд- та бекенд-розробки з притаманними їм відмінностями; проаналізувати етапи розвитку та характеристику можливостей веб-технологій для того, щоб визначити ключові аспекти, які посприяли інформаційним технологіям; провести літературний огляд з дослідженням застарілих та сучасних версій Bootstrap, HTML та CSS та проаналізувати їх можливості; на основі проаналізованої літератури визначити основні характеристики, переваги та недоліки веб-технологій для розробки веб-сторінок та веб-сайту; на основі проаналізованих джерел сформулювати послідовний алгоритм розробки веб-сторінки та веб-сайту.

### Виклад основного матеріалу

Для того, щоб проаналізувати сучасні можливості веб-програмування, було розглянуто основні аспекти веб-технологій, а саме: вивчення характеристики можливостей веб-розробок за допомогою сучасного Інтернету, де проаналізовані можливості фреймворку Bootstrap для фронтенд розробки з використанням мови гіпертекстової розмітки HTML5 та таблиць каскадних стилів CSS3. В якості аналізу були розглянуті основні можливості та недоліки попередніх версій для порівняння характеристик з сучасними можливостями.

**Характеристика можливостей веб-розробок за допомогою сучасного Інтернету.** Для того, щоб з'ясувати та дослідити сучасні можливості для веб-

програмування та веб-дизайну, насамперед необхідно проаналізувати покоління науково-технологічного прогресу у сфері веб-технологій та Інтернету. Як відомо, веб-технології розвивалися послідовно, так як протягом останніх двадцяти років технологічний прогрес досяг значних результатів з появою потужних комп'ютерів та розвитком інформаційних технологій. Розглянемо нижче кожне покоління веб-технологій з описом та характеристикою розвитку можливостей [6].

**Веб 0.0 Десктоп (Ера настільних комп'ютерів) 1980–1990 роки.** Інтернет, який вперше з'явився на початку 90-х років минулого століття, забезпечив способи публікації та доступу до документів в Інтернеті. Перші веб-стандарты стосувалися того, як документи можуть відображатися браузером, як ці документи можуть бути передані через Інтернет для читання користувачами. Вважається, що протокол передачі гіпертексту (HTTP) і мова розмітки гіпертексту (HTML) зробили значний внесок в успіх Інтернету.

**Веб 1.0 (Ранній Веб) 1989–2004 роки.** Версія для читання, в якій Інтернет 1.0 був статичним та одностороннім (споживання контенту). Інформація та контент подавалися у вигляді опису в брошурах, який надавався організаціями за допомогою Інтернету (характерні технології: FTP, IRC, Usenet, електронна пошта, SGML, SQL, MacOS і файлові сервери). У 2000–2010 роках розвивався Веб 2.0 (Flash, XML, http, P2P, AJAX, RDF, Java). Сайти епохи Веб 1.0 здебільшого були зроблені в одній з ранніх специфікацій HTML 2.0, HTML 3.0 та HTML 4.0, які мали ширину від 700 до 900 пікселів, що було пов'язано з розміром тодішніх моніторів з електронно-променевою трубкою (15 дюймів), і вони часто створювалися на основі таблиць або так званих фреймів.

**Веб 2.0 (Пост-епоха, соціальний Веб) 2004–2016 роки.** Версія пов'язана з розвитком соціальних та інших мереж, двосторонньої комунікації, різноманітних технологій з контентом. Веб 2.0 обумовлюється об'єднанням людей і створенням технологій, ефективних для людей, де встановлюються взаємовигідні відносини між користувачем і постачальником Інтернет-послуг.

**Веб 3.0 (Семантичний Веб) 2016–2020 роки.** Фундаментальна ідея полягає у охарактеризуванні інформації, її структуризації та пов'язанням таким чином, щоб зробити більш переконливою для розкриття, роботизації, інкорпорації та повторного або перехресного використання в різних додатках. Це дозволило об'єднувати, координувати та досліджувати інформацію з різних інформаційних секцій з метою отримання нового потоку даних, що дозволило покращити інформування керівників, підвищити доступність портативного Інтернету, імітувати інновації та розвиток, а також посилити фактор глобалізації та підвищити рівень задоволеності серед користувачів.

**Веб 4.0 (Мобільний та симбіотичний Веб)**

**2020 рік й до сьогодні.** Технологія Веб 4.0 є мобільною версією Веб 2.0 і катализатором розвитку Веб 2.0, в якому різні пристрої з'єднуються між собою в режимі реального часу. Розвиток Веб 4.0 спрямований на підключення до мережі якомога більшої кількості пристроїв. Революційним фактором Веб 4.0 є новий тип комунікації, як людей з об'єктами, так і об'єктів між собою (M2M-комунікація, машина з машиною). Інтернет речей можна розглядати як глобальну інфраструктуру інформаційного суспільства, що дозволяє надавати передові послуги, поєднуючи фізичні та віртуальні речі.

**Веб 5.0 (Симбіотичний Веб, емоційний та чутливий Веб) теперішній час.** Розробка персональних серверів, які можуть надавати інформацію поблизу будинку та інформацію про розміщення у мережі за допомогою смарт-комутатора. В іншому розумінні це дає можливість залучати розумні технології та штучний інтелект, що дозволить приймати рішення самостійно. Веб 5.0 використовує нейро-технології, які можуть інтерпретувати окремі біометричні показники та зчитувати емоції користувачів, завдяки чому веб-додатки зможуть змінювати вираз обличчя, у режимі реального часу.

**Веб 6.0 (Незалежне існування) майбутні перспективи.** Проаналізувавши можливості Веб 5.0, можна зробити припущення, що майбутній Веб 6.0 може слугувати в якості самостійної істоти з незалежним існуванням, яка міститиме ознаки автономного та незалежного інтелекту. У той час, коли у випадку Веб 4.0 і Веб 5.0 існують концепції штучного інтелекту та віртуального агента, то у випадку Веб 6.0 вони прагнуть бути незалежними, до такої міри, що їх не можна описати як «штучні». Веб 6.0 може набути форми «іншого інтелекту» або окремої сутності, яка буде функціонувати в екосистемі Інтернету в залежності від наявності електроімпульсів, без необхідності «прив'язки» до довготривалого носія даних.

**Фреймворк Bootstrap** у поєднанні з HTML, CSS та JavaScript фреймворками дозволяє створювати мобільний та адаптивний веб-сайт, де характерною особливістю є швидкість та зручність під час розробки. Bootstrap дозволяє легко та швидко розробити веб-сайт, оскільки він містить безліч корисних інструментів форматування, а також передбачає постійне оновлення з виправленнями у попередніх версіях. Завдяки постійному оновленню щороку випускаються дедалі новіші версії цього фреймворку, в яких враховуються недоліки з попередніх версій, що робить його більш сучасним у веб-програмуванні з веб-дизайном [9]. Варто зазначити, що Bootstrap у комплексі з HTML, CSS та JavaScript відноситься до фронтенд-розробки. На рис. 1 показані три рівні з метою визначення основних відмінностей між фронтенд- та бекенд-розробками [10].

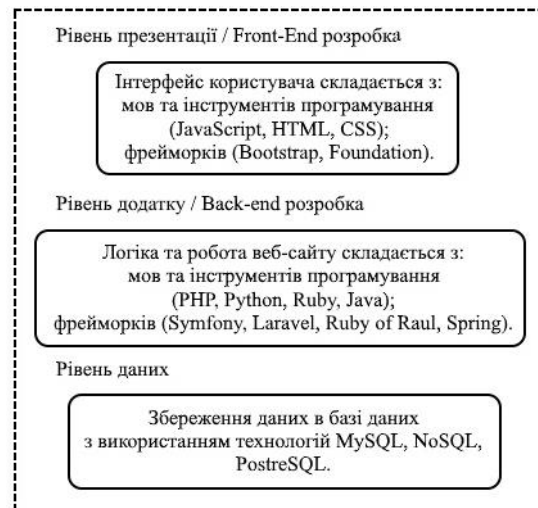


Рис. 1. Бекенд- та фронтенд-розробки з рівнями програмування та базою даних  
Джерело: сформовано авторами на основі [10]

Як показано на рис. 1, рівень презентації відрізняється від рівня додатку використанням різних мов програмування, однак рівень даних може містити збережені дані з двох цих рівнів на стороні серверу. Для того, щоб проаналізувати можливості користувачької сторони (сторони клієнта) та сторони серверу, на рис. 2 показана взаємозалежність обміну даних, однак клієнт на стороні розробленого фронтенду може взаємодіяти з елементами веб-розробки, а на стороні серверу ці властивості можуть бути обмеженими, що називається технологією поділу [12].



Рис. 2. Технологія поділу веб-додатків  
Джерело: сформовано авторами на основі [11]



Проаналізувавши обидва рисунки, можна припустити, що фреймворк Bootstrap разом з HTML, CSS та JavaScript можуть міститися на стороні клієнта з видимою оболонкою, що дозволяє взаємодіяти та змінювати дані у процесі розробки з веб-середовищем. Розглянемо основні характеристики фреймворку Bootstrap 5.0:

- орієнтований на мобільні пристрої з jsDelivr та шаблоном стартової сторінки (орієнтованих на мобільні пристрої по всій бібліотеці, а не в окремих бібліотеках);

- для роботи компонентів необхідне використання JavaScript з плагінами (один із наведених `<script>` символів у кінці сторінки, перед закриванням ставиться тег `<body>` для активації), що містить в собі вбудовану підтримку JavaScript для різних компонентів (відкриваюче меню, модальні вікна, вкладки тощо) для полегшення клієнтської логіки;

- легкий та зручний початок роботи, який передбачає знання лише HTML5 та CSS3 (не потрібно вручну налаштовувати багато CSS стилів через спрощений CSS), й надає консистентний та сучасний дизайн «з коробки»;

- адаптивність (адаптивний дизайн CSS Bootstrap), що містить сітку (grid system), яка дозволяє легко створювати адаптивні веб-сторінки, які добре виглядають на різних розмірах екранів з налаштуванням на комп'ютери, планшети та мобільні пристрої, забезпечуючи безперебійну роботу користувача;

- відкритий вихідний код, який забезпечує більш плавний, гнучкий та публічний процес розробки з багатьма доступними ресурсами;

- розширення та налаштування передбачає розширювання та налаштування стилів з компонентами, щоб вони відповідали унікальним потребам проекту;

- підтримка та документація, що характеризується спільнотою розробників та добре документованим API для допомоги з вирішення проблем та швидкого вивчення фреймворку.

Однак попередні версії фреймворку містили недоліки, які супроводжувалися перерозподілом стилів, що могло спричинити велику кількість надлишкового коду на веб-сайті з повільним завантаженням. Тому використання фреймворку Bootstrap 5 у комплексі з JavaScript, HTML5, CSS3 призначене для сучасних комп'ютерів з великим обсягом пам'яті та відповідними системними вимогами, що значно відрізняється від попередніх версій.

**CSS3 та HTML5.** Використання CSS є потужним інструментом для визначення вигляду та стилю веб-сторінок, проте вони також можуть містити ряд проблем, які можуть ускладнювати розробку та підтримку веб-сайтів. У табл. 1 показані основні недоліки та проблеми, які можуть містити CSS. Для розробників важливо бути уважними при написанні CSS й використовувати найкращі практики та інструменти, за допомогою яких можливе коментування та полегшення коду. Не менш важливо також стежити за оновленнями веб-стандартів та виправляти проблеми з помилками, що виникають у процесі розробки й спрямовані на забезпечення оптимальної продуктивності та вигляду веб-сайту на різних пристроях в браузері.

Таблиця 1

Проблеми та недоліки застарілих CSS під час розробки веб-сторінки та веб-сайту

Специфічність селекторів	Каскади CSS визначають, які стилі застосовуються до конкретних елементів, але якщо використовувати селектори з високою специфічністю (наприклад, <code>id</code> селектори або вкладені селектори), це може призвести до непередбачуваного стилю та складнощів в усуненні при полегдженні.
Загальний CSS код	Якщо веб-сайт складається з багатьох сторінок і повторюються одні й ті самі стилі на кожній сторінці, це може ускладнювати підтримку та розширення.
Кросбраузерна сумісність	Окремі браузери можуть по-різному інтерпретувати CSS, що може призвести до зміни вигляду веб-сайту на різних платформах.
Пріоритети стилів	CSS має систему пріоритетів з визначенням, який стиль буде застосований у разі, якщо конфліктують різні правила, що може стати складним у великих проєктах.
Довгий CSS код	Великий обсяг CSS може бути недоступним для перегляду під час обслуговування, особливо якщо він не організований та не коментований належним чином.
Надмірне використання «!important»	Використання «!important» для нав'язування стилів може зробити код менш передбачуваним і важким для розуміння.
Видимість полегдження	Деякі стилі можуть бути додані або перезаписані за допомогою JavaScript, що може робити полегдження та виправлення помилок складним.

Джерело: сформовано авторами на основі [15–16]

У сучасному веб-дизайні важливо відокремити вміст (написаний на мові HTML5) від макета (визначеного таблицею стилів CSS3). Це та сама концепція, яку дослідники застосовують щодня, використовуючи LaTeX, де макет і графічне представлення визначаються таблицею стилів, наданою видавцем, і зазвичай описується у файлі (.sty). Покращення, які прийшли з третьою версією CSS3 перебувають на етапі вдосконалення й можуть змінюватися в залежності від виявлених недоліків. Нові модулі CSS3 допомагають створювати привабливі сторінки, впроваджуючи нові методи для фону, границь, ефектів та анімації.

Незважаючи на те, що CSS3 та HTML5 є достатньо потужними інструментами для веб-програмування, вони можуть також містити ряд недоліків. До недоліків CSS3 відносяться:

- сумісність браузера, де різні браузери можуть інтерпретувати CSS3 правила по-різному, що може призвести до різниці у відображенні веб-сторінок на різних платформах;

- навантаження на браузер, де використання складних CSS3 анімацій та трансформацій може призвести до погіршення продуктивності, особливо для старих комп'ютерів та мобільних пристроїв;

- CSS3 має велику кількість властивостей та правил, що може зробити код менш зрозумілим та важким для обслуговування;

- неможлива підтримка в застарілих браузерах, що обумовлюється наявністю застарілих браузерів, які не оновлювалися й не здатні підтримувати CSS3, що так само може обмежувати можливості для розробників.

До недоліків HTML5 відносяться:

- сумісність браузера, так як різні браузери можуть підтримувати різні функції та елементи HTML5 по-різному;

- недостатня підтримка старими браузерами. Деякі застарілі версії не здатні підтримувати HTML5, що може вимагати додаткового коду або заміни функцій для забезпечення сумісності;

- збільшений обсяг коду, де використання нових елементів та функцій HTML5 може призвести до збільшення обсягу коду сторінки, що впливає на завантаження та продуктивність;

- неможлива підтримка деякими мобільними пристроями, так як застарілі моделі мобільних пристроїв не можуть підтримувати повну функціональність HTML5, що обмежує можливості розробників для створення мобільних додатків.

**Алгоритм розробки веб-сторінки та веб-сайту.** Використовуючи сучасні можливості мови програмування, її коду та набору допоміжних інструментів при розробці веб-сайту, у роботі наводиться алгоритм розробки, де кожен етап містить окремі процеси та операції під час розробки веб-

сайту, що обумовлює унеможливлення існування один без одного. Використовуючи сучасні можливості мови програмування, її коду та набору допоміжних інструментів при веб-розробці, розглянемо алгоритм для побудови веб-сторінки. Під час розробки веб-сторінки кожен етап містить окремі процеси та операції, що обумовлює унеможливлення існування один без одного. Наприклад, на етапі планування та аналізу необхідно завчасно з'ясувати, яку мову програмування доцільніше використовувати для фронтенд- та бекенд-розробок, які фреймворки відповідають для реалізації таких розробок, залучення верстки та розмітки та інше.

1. Планування та аналіз:

1.1 визначення мети (визначення мети та завдання щодо створення веб-сайту для розуміння його функціональності та доцільного призначення);

1.2 дослідження аудиторії (необхідно врахувати потреби користувачів, визначити чисельність аудиторії, яку складають користувачі для створення функціональності та необхідного дизайну);

2. Дизайн та прототип:

2.1 для графічного дизайну використовуються інструменти, за допомогою яких можливо створювати дизайн веб-сторінки, інтерфейсу та елементів (Adobe XD, Figma, Sketch);

2.2 для визначення структури та взаємодії веб-сторінок необхідно створити прототип.

3. Веб-програмування:

3.1 вибір технології з визначенням мови програмування та набору інструментів для розробки веб-сайту (HTML5, CSS3, JavaScript) з використанням фреймворків (Bootstrap, React, Angular);

3.2 бекенд-розробка призначена для розробки серверної частини й може залучати мови програмування (Python, Node.js) та фреймворки (Ruby);

3.3 розробка бази даних для зберігання інформації.

4. Верстка та розмітка з використанням кодових редакторів (використання кодових редакторів забезпечує надання функцій з підсвічування синтаксису, автодоповнення та інші корисні інструменти для розробників):

4.1 HTML5 та CSS3 для створення структури веб-сторінок, їх стилізації та розмітки. HTML5 містить безліч нових та покращених елементів (наприклад, <header>, <nav>, <section>, <video>, <audio> та інші), які дозволяють краще структурувати сторінки. Каскадні таблиці стилів 3-го покоління (CSS3) використовуються для створення складного та красивого дизайну (задання стилів, анімацій, переходів, тіней та інших ефектів), що дозволяє захоплювати більшу аудиторію користувачів;

4.2 Flexbox та CSS Grid для створення респонсивних та еластичних макетів за допомогою потужних CSS-модулів з розміщенням елементів на веб-

сторінці. Дозволяють полегшити розміщення елементів у веб-дизайні.

#### 5. Розробка функціоналу:

5.1 для створення інтерактивних елементів із додаванням функціональності до веб-сторінок використовується мова програмування JavaScript;

5.2 розробка фронтенд та бекенд повинна забезпечувати взаємодію за допомогою AJAX-запитів або API.

#### 6. Оптимізація та тестування:

6.1 перевірка розробленого веб-сайту на сумісність з різними браузерами (Google Chrome, Mozilla Firefox, Opera тощо) і на різних пристроях (операційні системи Windows, Android, IOS);

6.2 оптимізація веб-сайту для швидкого завантаження зі зменшенням розміру зображень за допомогою веб-кешу та інших методів.

#### 7. Реліз та підтримка:

7.1 реліз обумовлюється розміщенням веб-сайту на веб-сервері з налаштуванням домену;

7.2 підтримка й оновлення. Після релізу надалі необхідно підтримувати та оновлювати веб-сайт згідно з потребами користувачів та змінами в Інтернет-середовищі.

#### 8. Інструменти та фреймворки:

8.1 використання фреймворків забезпечує швидку розробку веб-сайту й надає готові компоненти, сітки та стилі, які можна з легкістю налаштувати (React, Angular, Vue.js, Bootstrap). Bootstrap робить респонсивну верстку та перетворює користувацький інтерфейс на більш спрощений та легкодоступний;

8.2 інструменти для автоматизації завдань, такі як компіляція CSS та JavaScript для оптимізації зображень (Gulp, Webpack).

Таким чином, засоби при проектуванні та верстки веб-сайтів з використанням сучасних технологій можуть значно спростувати процес розробки, що може створювати більш якісний веб-контент. Тому важливо враховувати те, що кожен проект може мати певні особливості з підбором засобів та технологій, де необхідно враховувати конкретні вимоги та потреби згідно проекту.

### **Висновки і перспективи подальших досліджень**

Веб-програмування у результаті своєї еволюції зазнало значних змін завдяки технологічному розвитку інформаційних технологій, коли вперше з'явилися комп'ютери, а разом з ними невдовзі й мова програмування. Кожен етап еволюції містить значний внесок у розвиток інформаційних технологій та веб-розробок, за допомогою чого людство почало досліджувати нові можливості комунікацій та обміну інформації. Сучасні технології характеризуються значним технологічним проривом, що обумовлено розвитком штучного інтелекту, Інтернету

речей та сучасних Інтернет-технологій, коли штучний інтелект матиме змогу слугувати в якості окремої та незалежної істоти з самостійним вибором.

Проаналізувавши дві технології веб-розробок фронтенд та бекенд, можна зробити висновок, що рівень презентації (фронтенд-розробка) відповідає стороні клієнта, з якою клієнт може взаємодіяти й на якій можуть додаватися динамічні елементи до HTML-сторінки за допомогою JavaScript з візуальним визначенням веб-сторінки за допомогою CSS. У випадку бекенд-розробки рівень додатку відповідає стороні серверу, з якою клієнт не може взаємодіяти, так як оболонка знаходиться у прихованому вигляді.

У даному дослідженні на основі проведеного аналізу еволюції веб-технологій було з'ясовано, що технології веб-програмування розвиваються й на сьогодні, що характеризується оновленням програмного забезпечення, виявленням та усуненням технічних помилок під час веб-програмування. На основі огляду літератури проаналізовано можливості розвитку застарілих інструментів та мов програмування, що ґрунтується на використанні фреймворку Bootstrap, який складається з мови гіпертекстової розмітки тексту HTML п'ятого покоління та каскадних таблиць стилів CSS третього покоління з використанням мови програмування JavaScript. Кожна з описаних технологій містить ряд переваг та недоліків, які пов'язані з оновленням та відсутністю відповідного обладнання та можливостей через застарілі системи інформаційних технологій.

Проаналізувавши фреймворк Bootstrap 5 у комплексі з HTML5, CSS3 та JavaScript можна зробити наступний висновок, що даний фреймворк орієнтований на мобільні пристрої з використанням JavaScript плагінів, що дозволяє легко адаптувати під будь-які розміри екрану з налаштуванням. За допомогою HTML5 та CSS3 можна легко задавати стилі для оформлення через спрощений CSS. Однак попередні версії фреймворку містили недоліки, які супроводжувалися перерозподілом стилів, що могло спричинити велику кількість надлишкового коду на веб-сайті з повільним завантаженням. Тому використання фреймворку Bootstrap 5 у комплексі з JavaScript, HTML5, CSS3 призначене для сучасних комп'ютерів з великим обсягом пам'яті та відповідними системними вимогами, що значно відрізняються від попередніх версій.

На основі проведеного аналізу побудований алгоритм розробки веб-сторінки та веб-сайту, який характеризується наступними етапами: планування та аналіз, дизайн та прототип, веб-програмування, розробка бази для зберігання даних та розробка функціоналу, оптимізація та тестування, реліз та підтримка, а також інструменти та фреймворки. Кожен етап містить окремі процеси та операції під час розробки веб-сторінки, що обумовлює неможли-

вість існування один без одного. Наприклад, на етапі планування та аналізу необхідно завчасно з'ясувати, яку мову програмування доцільніше використовувати для фронтенд- та бекенд-розробок, які фреймворки відповідають для реалізації таких розробок, залучення верстки та розмітки тощо.

### References

- Almeida, F., & Monteiro, J. (2017). The Role of Responsive Design in Web Development. *Webology*, 14(2), 157. Retrieved from <http://www.webology.org/2017/v14n2/a157.pdf>
- Ricci, D., Nicastro, L., & Pio, M. A. (2013). Modern Web 2.0 tools for educational astronomy. *Proceedings of the 7th International Technology, Education and Development Conference (INTED2013)* (pp. 2998–3007). International Association of Technology, Education and Development (IATED). Retrieved from <http://ross.iasfbo.inaf.it/GLORIA-WP2/D2.3/nicastro-proceedings-inted2013.pdf>
- Lazuardy, M. F. S., & Anggraini, D. (2022). Modern Front End Web Architectures with React.Js and Next.Js. *International Research Journal of Advanced Engineering and Science*, 7(1), 132–141. Retrieved from <http://irjaes.com/wp-content/uploads/2022/02/IRJAES-V7N1P162Y22.pdf>
- Wang, L. (2016). Research on the UI and UE Design Techniques and the Status in HTML5 Website Development. *Proceedings of the 2016 2nd International Conference on Social Science and Technology Education (ICSSTE 2016)* (pp. 853–858). Atlantis Press. DOI: [10.2991/icsste-16.2016.154](https://doi.org/10.2991/icsste-16.2016.154)
- Sotnik, S., Shakurova, T., & Lyashenko, V. (2023). Development Features Web-Applications. *International Journal of Academic and Applied Research (IJAAAR)*, 7(1), 79–85. Retrieved from <https://openarchive.nure.ua/handle/document/21600>
- Król, K. (2020). Evolution of online mapping: from Web 1.0 to Web 6.0. *Geomatics, Landmanagement and Landscape*, (1), 33–51. DOI: [10.15576/GLL/2020.1.33](https://doi.org/10.15576/GLL/2020.1.33)
- Bader, W. I., & Hammouri, A. I. (2016). Responsive Web Design Techniques. *International Journal of Computer Applications*, 150(2), 18–27. DOI: [10.5120/ijca2016911463](https://doi.org/10.5120/ijca2016911463)
- Önday, Ö. (2019). Web 6.0: Journey From Web 1.0 To Web 6.0. *Journal of Media & Management*, 1(1), 1–6. DOI: [10.47363/JMM/2019\(1\)102](https://doi.org/10.47363/JMM/2019(1)102)
- Mohd, T. K., Thompson, J., Carmine, A., & Reuter, G. (2022). Comparative Analysis on Various CSS and JavaScript Frameworks. *Journal of Software*, 17(6), 282–291. DOI: [10.17706/jsw.17.6.282-291](https://doi.org/10.17706/jsw.17.6.282-291)
- Laaziri, M., Benmoussa, K., Khouliji, S., Larbi, K. M., & El Yamami, A. (2019). Analyzing bootstrap and foundation front-end frameworks: a comparative study. *International Journal of Electrical and Computer Engineering (IJECE)*, 9(1), 713–722. DOI: [10.11591/ijece.v9i1.pp713-722](https://doi.org/10.11591/ijece.v9i1.pp713-722)
- Kar, S., Islam, M. M., & Rahaman, M. (2020). State-of-the-Art Reformation of Web Programming Course Curriculum in Digital Bangladesh. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 11(3), 193–201. DOI: [10.14569/IJACSA.2020.0110323](https://doi.org/10.14569/IJACSA.2020.0110323)
- Gupta, P., & Agarwal, D. (2022). Webapp service for providing information about NGOs using HTML5, CSS3, JavaScript, Node JS, MongoDB. *Proceedings of the 2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)* (pp. 2262–2266). Institute of Electrical and Electronics Engineers (IEEE).
- Wilson, D., Hassan, S.-U., Aljohani, N. R., Visvizi, A., & Nawaz, R. (2023). Demonstrating and negotiating the adoption of web design technologies: Cascading Style Sheets and the CSS Zen Garden. *Internet Histories*, 7(1), 27–46. DOI: [10.1080/24701475.2022.2055274](https://doi.org/10.1080/24701475.2022.2055274)
- Sharma, A. K. (2021). BIG BUY (E-Commerce website) by using Frontend Web Development. *International Journal for Modern Trends in Science and Technology*, 7(11), 201–208. Retrieved from <https://www.ijmtst.com/volume7/issue11/34.IJMTST0711095.pdf>
- Biçer, M. S., & Diri, B. (2016). Defect prediction for Cascading Style Sheets. *Applied Soft Computing*, 49, 1078–1084. DOI: [10.1016/j.asoc.2016.05.038](https://doi.org/10.1016/j.asoc.2016.05.038)
- Bos, B., Çelik, T., Hickson, I., & Lie, H. W. (Eds.). (2005). *Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification*. World Wide Web Consortium. Retrieved from <https://www.w3.org/TR/2005/WD-CSS21-20050613/css2.pdf>

**Рецензент:** д-р техн. наук, проф. В.В. Сабадаш, Національний університет «Львівська політехніка», Україна.

**Автор:** ФЕДОРЧУК Анна Леонідівна  
кандидат педагогічних наук, доцент, доцент кафедри комп'ютерних наук та інформаційних технологій Житомирський державний університет імені Івана Франка  
E-mail – [anna.l.fedorchuk@gmail.com](mailto:anna.l.fedorchuk@gmail.com)  
ID ORCID: <https://orcid.org/0000-0001-8227-3210>

**Автор:** УСАТА Олена Юріївна  
кандидат педагогічних наук, доцент, доцент кафедри комп'ютерних наук та інформаційних технологій Житомирський державний університет імені Івана Франка  
E-mail – [o.y.usata@gmail.com](mailto:o.y.usata@gmail.com)  
ID ORCID: <https://orcid.org/0000-0002-0610-7007>

**Автор:** НАКОНЕЧНА Оксана Андріївна  
кандидат технічних наук, доцент кафедри інформаційних технологій Одеський державний аграрний університет  
E-mail – [Nakonechnaya@ukr.net](mailto:Nakonechnaya@ukr.net)  
ID ORCID: <https://orcid.org/0000-0001-5547-130X>



## WEB DESIGN AND WEB PROGRAMMING IN THE MODERN INTERNET WORLD

A. Fedorchuk<sup>1</sup>, O. Usata<sup>1</sup>, O. Nakonechna<sup>2</sup>

<sup>1</sup>Zhytomyr Ivan Franko State University, Ukraine

<sup>2</sup>Odesa State Agrarian University, Ukraine

*Modern web content development capabilities are undergoing characteristic changes due to the improvement of web technologies in programming. The origin of web technologies began in the early 80s of the last century when the Hypertext Transfer Protocol (HTTP) and the Hypertext Markup Language (HTML) made a significant contribution to the development of Internet capabilities. The web evolution, characterised by scientific and technological progress in information technology, has opened up new opportunities for the development of web programming, which has led to the emergence of programming languages, frameworks, cascading style sheets, and hypertext markup languages. This research aims to study and analyse the key aspects of web programming and web design with a combination of modern features of the Internet environment. To analyse web design, the article considers the possibilities of using the Bootstrap framework, which combines and contains the HTML5 hypertext markup language, CSS3 cascading style sheets, and the JavaScript programming language. Having analysed two web development technologies—front-end and back-end—we can conclude that the presentation level (front-end development) corresponds to the client side, with which the client can interact and where dynamic elements can be added to an HTML page using JavaScript with the visual definition of a web page using CSS. In the case of back-end development, the application layer corresponds to the server side, with which the client cannot interact, as the shell is hidden. Using the modern capabilities of the programming language, its code, and a set of auxiliary tools in website development, the paper presents a development algorithm where each stage contains separate processes and operations during website development, which makes it impossible for them to exist independently. Thus, tools for designing and laying out websites using modern technologies can greatly simplify the development process, which can create higher-quality web content.*

**Keywords:** Web, web environment, web technologies, front-end, framework, Bootstrap, CSS3, HTML5, CCS3, JavaScript.