ЗАКЛАД ВИЩОЇ ОСВІТИ
«УНІВЕРСИТЕТ КОРОЛЯ ДАНИЛА»

Факультет суспільних та прикладних наук

Кафедра інформаційних технологій

ЗАТВЕРДЖУЮ
Проректор з навчально-методичної роботи
_____ Ярослав ШТАНЬКО
“31” серпня 2026 р.

Розробка веб-додатків, веб-дизайн та тестування

СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Галузь знань:	12 Інформаційні технології
Спеціальність:	121 Інженерія програмного забезпечення
Освітньо-професійна (освітньо-наукова) програма:	Розробка та тестування програмного забезпечення
Освітній рівень:	перший (бакалаврський)
Статус дисципліни:	обов’язкова
Мова викладання, навчання та оцінювання:	українська
Семестр викладання	5 семестр (3 курс)

Івано-Франківськ
2026

РОЗРОБНИК:
доцент кафедри ІТ — Дмитро ЛУЦАК
доцент кафедри ІТ — Олександр ІВАНОВ

ЗАТВЕРДЖЕНО:
на засіданні кафедри інформаційних технологій, протокол № 1 від 27.08.2026 р.

Завідувач кафедри — Олександр ІВАНОВ
УЗГОДЖЕНО:

Гарант ОПП/ОНП — Володимир МАКОВИШИН

СХВАЛЕНО:
на засіданні Науково-методичної ради, протокол № 1 від 31.08.2026 р.

e-mail	dmytro.l.lutsak@ukd.edu.ua
Номер аудиторії чи кафедри	206
Посилання на сайт	https://ukd.edu.ua
Сторінка курсу в СДО	https://online.ukd.edu.ua/

ВСТУП

Анотація навчальної дисципліни

Навчальна дисципліна «Розробка веб-додатків, веб-дизайн та тестування» формує навички створення сучасних веб-додатків: від проєктування інтерфейсу до перевірки якості готового продукту. Основними технологіями є HTML, CSS і JavaScript, а середовищем виконання – браузер. Курс охоплює семантичну розмітку HTML, оформлення та компонування веб-інтерфейсів засобами CSS, об’єктну модель документа (DOM) і браузера (BOM), обробку подій за допомогою JavaScript, взаємодію з сервером через Web API та різні підходи до побудови й рендерингу веб-додатків (MPA, SPA, SSR, SSG). Окремий блок присвячено дизайну веб-додатків як складовій частині процесу розробки: здобувачі вивчають основи веб-дизайну, реалізацію адаптивного дизайну засобами CSS, принципи веб-доступності та переваги використання дизайн-систем. Завершальна частина курсу охоплює основи тестування веб-додатків. Навчання побудовано навколо практичних завдань, де здобувачі проводять проєктування, розробку і тестування веб-додатків.

Мета навчальної дисципліни – сформувати у здобувачів вищої освіти теоретичні знання та практичні навички, потрібні для повного циклу розробки сучасних веб-додатків.

Основні завдання навчальної дисципліни

- Ознайомити здобувачів із сучасними підходами до побудови й рендерингу веб-додатків (MPA, SPA, SSR, SSG), їхніми перевагами, обмеженнями та сферами застосування.
- Сформувати вміння створювати семантичну розмітку веб-сторінок засобами HTML, проводити оформлення й компонування веб-інтерфейсів засобами CSS.
- Навчити працювати з об’єктною моделлю документа (DOM) і браузера (BOM), обробляти події користувача та керувати вмістом сторінки за допомогою JavaScript.
- Сформувати навички взаємодії клієнтської частини веб-додатка з сервером через Web API, зокрема надсилання запитів, обробки відповідей і помилок.
- Ознайомити з основами веб-дизайну та процесом проєктування інтерфейсів як складовою частиною розробки веб-додатків.
- Навчити реалізовувати адаптивний дизайн засобами CSS, застосовувати принципи веб-доступності та використовувати дизайн-системи.
- Сформувати знання про принципи, рівні та види тестування веб-додатків, а також навички ручного і автоматизованого тестування.
- Розвинути вміння застосовувати набуті знання на всіх етапах повного циклу розробки веб-додатка: проєктування, розробки та тестування.

У результаті вивчення дисципліни здобувач повинен знати

- підходи до побудови й рендерингу веб-додатків (MPA, SPA, SSR, SSG), їхні переваги, обмеження та сфери застосування;
- принципи семантичної розмітки HTML і призначення основних елементів веб-сторінки;
- засоби CSS для оформлення та компонування веб-інтерфейсів, зокрема CSS Flexbox, CSS Grid, змінні CSS і медіазапити;
- структуру об’єктної моделі документа (DOM) і браузера (BOM), типи вузлів і способи доступу до них за допомогою JavaScript;
- модель подій браузера: основні категорії подій, фази поширення події, способи обробки та скасування дій за замовчуванням;
- принципи асинхронного виконання коду в JavaScript та механізми взаємодії з сервером через Web API;
- основні етапи проєктування веб-інтерфейсу;
- принципи адаптивного дизайну та засоби його реалізації в CSS;
- призначення та складові частини дизайн-систем, поняття дизайн-токенів;
- основні принципи та рівні тестування, модель піраміди тестування;
- відмінності між ручним і автоматизованим, функціональним і нефункціональним тестуванням, а також основні інструменти для тестування веб-додатків.

У результаті вивчення дисципліни здобувач повинен уміти

- обирати підхід до побудови й рендерингу веб-додатка (MPA, SPA, SSR, SSG) відповідно до його призначення та вимог;
- створювати семантичну та валідну HTML-розмітку веб-інтерфейсів;
- оформлювати й компонувати веб-інтерфейси засобами CSS, зокрема з використанням CSS Flexbox, CSS Grid і змінних CSS;
- знаходити, створювати, змінювати та видаляти елементи DOM за допомогою JavaScript;
- використовувати об’єкти BOM для роботи з вікном браузера, адресою сторінки, історією переходів і локальним сховищем;
- обробляти події користувача та браузера, керувати їх поширенням і діями за замовчуванням;
- надсилати запити до сервера через Web API, обробляти відповіді у форматі JSON і помилки запитів;
- проєктувати інтерфейс веб-додатка: створювати каркаси та макети екранів з урахуванням їхніх станів;
- реалізовувати адаптивний дизайн засобами CSS для різних розмірів екранів;
- використовувати дизайн-токени та компоненти дизайн-системи під час розробки інтерфейсу;
- розробляти та виконувати тести веб-додатків різних рівнів.

Компетентності та результати навчання

Шифри й формулювання наведено відповідно до ОПП «Розробка та тестування програмного забезпечення» у редакції 2024 року. Стовпці містять незалежні переліки компетентностей і програмних результатів навчання для ОК 35.

Шифр та назва компетентності	**Шифр та назва програмних результатів навчання**
ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.	ПРН5. Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об’єктно-орієнтованого аналізу та математичного моделювання для розробки програмного забезпечення.
ЗК2. Здатність застосовувати знання у практичних ситуаціях.	ПРН12. Застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення.
ФК3. Здатність розробляти архітектури, модулі та компоненти програмних систем.	ПРН15. Мотивовано обирати мови програмування та технології розробки для розв’язання завдань створення і супроводження програмного забезпечення.
ФК4. Здатність формулювати та забезпечувати вимоги щодо якості програмного забезпечення у відповідності з вимогами замовника, технічним завданням та стандартами.	
ФК10. Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя.	

Шифр та назва компетентності	Шифр та назва програмних результатів навчання
ФК13. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.	

ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Курс	**3**		
Семестр	**5**		
Кількість кредитів ECTS	**6**		
Аудиторні навчальні заняття		**денна форма**	**заочна форма**
	лекції	**30**	**6**
	практичні	**30**	**6**
Самостійна робота		**120**	**168**
Форма підсумкового контролю	**екзамен**		

Загальний обсяг дисципліни – 180 годин. Для денної форми передбачено 30 годин лекційних, 30 годин практичних занять і 120 години самостійної роботи; для заочної форми – 6 годин, 6 годин та 168 годин відповідно.

Структурно-логічна схема вивчення навчальної дисципліни

Пререквізити	Постреквізити
Основи програмування, об’єктно-орієнтоване програмування, інженерія програмного забезпечення, людино-машинна взаємодія, UI/UX дизайн	Якість програмного забезпечення

Для занять потрібні персональний комп’ютер, сучасний браузер, редактор коду, Node.js, Git і доступ до СДО. Навчальні репозиторії не повинні містити паролів, токенів доступу, а також персональних або конфіденційних даних.

ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Зміст лекційних занять

Тема 1. Глобальна мережа Інтернет. Розробка веб-сайтів і веб-додатків (2 год.)

Історія створення та основи організації глобальної мережі Інтернет; Робота веб-браузера; Основні етапи розробки веб-сайтів та веб-додатків.

Питання для самостійного опрацювання: Міжнародні організації з розробки і підтримки веб-стандартів (W3C, WHATWG, IETF, Ecma International); Пошукові системи: сканування, індексування, ранжування веб-сайтів. Література: [1-3, 5, 10, 18].

Тема 2. Основи HTML (2 год.)

Вступ в HTML. Основні елементи й атрибути HTML. Типова структура HTML-документа.

Питання для самостійного опрацювання: Валідація HTML коду та типові помилки розмітки; Мікророзмітка та структуровані дані Schema.org; Метатеги для соціальних мереж (Open Graph). Література: [2-3, 5, 9-11].

Тема 3. Основи CSS (2 год.)

CSS: призначення та структура. Способи додавання CSS до веб-сторінки. Основи позиціонування елементів веб-сторінки. Адаптивний дизайн веб-сторінки. Використання медіазапитів в CSS.

Питання для самостійного опрацювання: Вкладеність у CSS (nesting), CSS селектор :has(), каскадні шари CSS (@layer). Література: [1-3, 10, 12].

Тема 4. Веб-сторінка: робота з таблицями та веб-формами (2 год.)

Особливості роботи з веб-сторінками, що містять таблиці. Веб-форма: призначення та основні елементи. Взаємодія користувача з веб-формою.

Питання для самостійного опрацювання: Захист веб-форм від спаму та автоматизованого заповнення; Обробка та перевірка даних веб-форм на сервері. Література: [2-3, 6, 12, 18].

Тема 5. Робота з графічними та мультимедійними елементами (2 год.)

Особливості роботи з графічними файлами для веб-сторінок та веб-додатків. Додавання мультимедійних елементів до веб-сторінок та веб-додатків. Використання CSS-анімації.

Питання для самостійного опрацювання: Доступність мультимедійного контенту (субтитри, альтернативний текст, налаштування prefers-reduced-motion); Вбудовування відео зі сторонніх платформ. Література: [2-5, 7, 10, 12, 18].

Тема 6. Стилізація тексту та робота з веб-шрифтами (2 год.)

Основи стилізації тексту за допомогою CSS. Особливості роботи з веб-шрифтами.

Питання для самостійного опрацювання: Варіативні шрифти (variable fonts) та їхні переваги; Іконкові шрифти та SVG-іконки; Особливості типографіки для кирилиці та багатомовних сайтів. Література: [1-4, 12].

Тема 7. Позиціонування елементів за допомогою CSS Flexbox та CSS Grid (4 год.)

Основні концепції CSS Flexbox. Особливості позиціонування елементів за допомогою CSS Flexbox. Переваги і обмеження CSS Flexbox. Основні концепції CSS Grid. Особливості позиціонування елементів за допомогою CSS Grid.

Питання для самостійного опрацювання: Історичні способи верстки макетів (таблиці, float, inline-block); Інструменти налагодження Flexbox і Grid у DevTools; CSS-фреймворки з готовою системою сітки (Bootstrap, Tailwind CSS). Література: [1-3, 12].

Тема 8. Робота JavaScript у браузері (4 год.)

Вступ до JavaScript. JavaScript: особливості роботи з об'єктною моделлю документа (DOM). JavaScript: особливості роботи з об'єктною моделлю браузера (BOM). Використання обробників подій для взаємодії з елементами веб-сторінки.

Питання для самостійного опрацювання: Веб-компоненти та користувацькі елементи (Custom Elements); Налагодження JavaScript-коду в інструментах розробника браузера. Література: [1, 4-6, 13-16].

Тема 9. Асинхронний JavaScript і взаємодія з сервером за допомогою Web API (4 год.)

Синхронний і асинхронний код. Цикл подій у JavaScript. Web API та архітектурний стиль REST. Робота з даними формату JSON.

Питання для самостійного опрацювання: Політика однакового походження та механізм CORS; GraphQL як альтернатива архітектурному стилю REST; Автентифікація під час роботи з API (токени, JWT, OAuth). Література: [1, 4-6, 14-15].

Тема 10. Класифікація веб-додатків за стратегією рендерингу HTML (2 год.)

Особливості рендерингу HTML структури у веб-додатках. Багато-сторінкові веб-додатки (Multi-Page Applications, MPA). Односторінкові веб-додатки (Single-Page Applications, SPA). Веб-додатки з серверним рендерингом (Server-Side Rendering Apps, SSR). Веб-додатки з статичною генерацією (Static Site Generation Apps, SSG).

Питання для самостійного опрацювання: Прогресивні веб-додатки (PWA); Гідратація та архітектура «островів». Література: [1, 5-6, 10].

Тема 11. Проєктування інтерфейсів веб-додатків (2 год.)

Дизайн веб-додатків як складова частина процесу розробки. Типові етапи проєктування інтерфейсів веб-додатків. Дизайн-системи для проєктування інтерфейсів веб-додатків.

Питання для самостійного опрацювання: Принципи доступності веб-інтерфейсів і стандарт WCAG; Темна тема інтерфейсу: принципи та реалізація; Мікровзаємодії в інтерфейсах. Література: [2, 7, 19-20].

Тема 12. Основи-тестування веб-додатків (2 год.)

Мета, принципи та рівні тестування веб-додатків. Ручне і автоматизоване тестування веб-додатків. Функціональне і нефункціональне тестування веб-додатків.

Питання для самостійного опрацювання: Покриття коду тестами та його обмеження; Мутаційне тестування. Література: [1, 8, 19].

Зміст практичних занять

Тема 1. Створення структури вебсайту засобами HTML5 (4 год.)

Розробка багатосторінкового вебсайту з використанням семантичних тегів HTML5; створення навігаційного меню, заголовків, списків, гіперпосилань та структурних блоків сторінки; формування логічної структури вебресурсу.

Питання для самостійного опрацювання: Історія розвитку HTML; стандарти W3C; семантична розмітка та її вплив на SEO.

Тема 2. Створення вебформ, таблиць та мультимедійних елементів (4 год.)

Розробка таблиць для подання структурованих даних; створення форм введення інформації з використанням різних типів елементів керування; розміщення на вебсторінках графічних, аудіо- та відеоматеріалів.

Питання для самостійного опрацювання: Валідація даних у вебформах; формати графічних та мультимедійних файлів для вебзастосунків; питання доступності вебконтенту.

Тема 3. Оформлення вебсторінок засобами CSS (4 год.)

Створення зовнішніх таблиць стилів; форматування тексту, списків, таблиць та навігаційних елементів; використання селекторів, класів та ідентифікаторів; розробка єдиного стилю вебресурсу.

Питання для самостійного опрацювання: Каскадність і специфічність CSS-правил; сучасні підходи до організації CSS-коду; препроцесори CSS.

Тема 4. Розробка адаптивних макетів за допомогою Flexbox та CSS Grid (4 год.)

Побудова одноколонкових та багатоколонкових макетів; використання CSS Flexbox для розташування елементів інтерфейсу; створення складних макетів із застосуванням CSS Grid; адаптація вебсторінок до різних розмірів екранів.

Питання для самостійного опрацювання: Responsive Web Design; медіа-запити CSS; Mobile First підхід до розробки вебінтерфейсів.

Тема 5. Розробка інтерактивних вебсторінок засобами JavaScript (4 год.)

Підключення та використання JavaScript у вебдокументах; робота з DOM-моделлю документа; обробка подій користувача; створення інтерактивних елементів інтерфейсу; програмна зміна вмісту вебсторінок.

Питання для самостійного опрацювання: Сучасний стандарт ECMAScript; об'єктна модель браузера (BOM); використання Local Storage у вебзастосунках.

Тема 6. Проєктування та прототипування вебінтерфейсів (4 год.)

Аналіз структури вебдодатків; створення карти сайту; моделювання сценаріїв взаємодії користувачів; розробка wireframe-макетів та інтерактивних прототипів інтерфейсів із використанням сучасних засобів проєктування.

Питання для самостійного опрацювання: Принципи UX/UI-дизайну; дизайн-системи; використання Figma для проєктування вебінтерфейсів.

Тема 7. Тестування вебдодатків та оформлення звітної документації (6 год.)

Розробка тестових сценаріїв для вебдодатків; проведення функціонального тестування; перевірка коректності роботи вебформ та інтерактивних компонентів; тестування адаптивності інтерфейсу; документування виявлених дефектів та підготовка звіту про тестування.

Питання для самостійного опрацювання: Види тестування програмного забезпечення; автоматизоване тестування вебдодатків; системи управління дефектами та баг-трекінг.

За кожною темою здобувач подає вихідний код, контрольні приклади або тести, короткий висновок і посилання на репозиторій, якщо це передбачено завданням. Програма має запускатися за наданою інструкцією, а здобувач – пояснювати її алгоритм і ключові рішення.

Зміст самостійної роботи здобувачів

Розподіл годин, виділених на вивчення дисципліни

Найменування видів робіт	**Розподіл годин**	
	денна форма	**заочна форма**
Самостійна робота, год, у т.ч.:	**120**	**168**
Опрацювання теоретичного матеріалу, поданого в СДО та на практичних заняттях	19	27
Підготовка до практичних занять та контрольних заходів	30	42
Підготовка й оформлення звітів, коду та навчального репозиторію	19	27
Підготовка до поточного та підсумкового контролю	10	14

Найменування видів робіт	Розподіл годин	
	денна форма	заочна форма
Опрацювання матеріалу, винесеного на самостійне вивчення, та виконання підсумкового проєкту	42	58

Самостійна робота включає роботу з документацією, додаткові завдання, тести й звіти, ведення Git-репозиторію та інтегрований вебпроєкт. Конкретні завдання і строки виконання публікуються в СДО.

ПОЛІТИКА КУРСУ

Коротко, з покликанням на відповідну нормативну базу УКД, висвітлити питання.[1]

1) щодо системи поточного і підсумкового контролю

Організація поточного та підсумкового семестрового контролю знань студентів, проведення практик та атестації, переведення показників академічної успішності за 100-бальною шкалою в систему оцінок за національною шкалою здійснюється згідно з "Положенням про систему поточного і підсумкового контролю, оцінювання знань та визначення рейтингу здобувачів освіти". Ознайомитись з документом можна за посиланням.

2) щодо оскарження результатів контрольних заходів

Здобувачі вищої освіти мають право на оскарження оцінки з дисципліни отриманої під час контрольних заходів. Апеляція здійснюється відповідно до «Положення про політику та врегулювання конфліктних ситуацій». Ознайомитись з документом можна за покликанням.

3) щодо відпрацювання пропущених занять

Згідно "Положення про організацію освітнього процесу" здобувач допускається до семестрового контролю з ***конкретної навчальної дисципліни (семестрового екзамену, диференційованого заліку), якщо він виконав усі види робіт, передбачені на семестр навчальним планом та силабусом/робочою програмою навчальної дисципліни, підтвердив опанування на мінімальному рівні результатів навчання (отримав ≥35 бали), відпрацював визначені індивідуальним навчальним планом всі лекційні, практичні, семінарські та***

[1] зміст пунктів може редагуватись з огляду на особливості курсу

лабораторні заняття, на яких він був відсутній. *Ознайомитись з документом можна за покликанням.*

4) щодо дотримання академічної доброчесності



"Положення про систему забезпечення академічної доброчесності" закріплює моральні принципи, норми та правила етичної поведінки, позитивного, сприятливого, доброчесного освітнього і наукового середовища, професійної діяльності та професійного спілкування спільноти Університету, викладання та провадження наукової (творчої) діяльності з метою забезпечення довіри до результатів навчання. Ознайомитись з документом можна за покликанням.

5) щодо використання штучного інтелекту



"Політика використання штучного інтелекту" визначає політику щодо використання технічних засобів на основі штучного інтелекту в освітньому процесі. Ознайомитись з документом можна за покликанням.[2] *"Положення про систему забезпечення академічної доброчесності" містить рекомендації щодо використання в академічних текстах генераторів на основі штучного інтелекту. Ознайомитись з документом можна за покликанням.*

6) щодо використання технічних засобів в аудиторії та правила комунікації



Використання мобільних телефонів, планшетів та інших гаджетів під час лекційних та практичних занять дозволяється виключно у навчальних цілях (для уточнення певних даних, перевірки правопису, отримання довідкової інформації тощо). На гаджетах повинен бути активований режим «без звуку» до початку заняття. Під час занять заборонено надсилання текстових повідомлень, прослуховування музики, перевірка електронної пошти, соціальних мереж тощо. Під час виконання заходів контролю використання гаджетів заборонено (за винятком, коли це передбачено умовами його проведення). У разі порушення цієї заборони результат анульовується без права перескладання. Ознайомитись з відповідним документом можна за посиланням.
Комунікація відбувається через електронну пошту і сторінку дисципліни в Moodle.

7) щодо зарахування результатів навчання, здобутих шляхом формальної/інформальної освіти



Процедури визнання результатів навчання, здобутих шляхом формальної/інформальної освіти визначаються «Положенням про порядок визнання результатів навчання, здобутих шляхом неформальної та / або інформальної освіти». Ознайомитись з документом можна за покликанням.[3]

---

[2] визначається політика використання ШІ в навчальній дисципліні - дозволене/заборонене, правила використання

[3] визначається перелік електронних та інших ресурсів та умови перезарахування

НАВЧАННЯ

Навчання здійснюється через короткі пояснення й демонстрації, кероване програмування, поступове ускладнення задач, аналіз помилок, взаємне рецензування коду та проєктну роботу. На практичних заняттях здобувачі планують алгоритм, реалізують його, перевіряють за допомогою контрольних даних, налагоджують і пояснюють результат.

Результат навчання	Метод навчання	Метод оцінювання
ПРН7. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.	Демонстрація мовних конструкцій, кероване програмування, налагодження у браузері й Node.js, робота з офіційною документацією.	Практичні задачі, пояснення коду, тести з основ мови та інструментів.
ПРН13. Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.	Алгоритмічні вправи, трасування, декомпозиція, робота з масивами й об’єктами, тестування граничних випадків.	Перевірка алгоритмів і коду, захист практичних робіт, контрольні задачі.
ПРН15. Мотивовано обирати мови програмування та технології розробки для розв’язання завдань створення і супроводження програмного забезпечення.	Порівняння інструментів і підходів, Git-практикум, код-рев’ю, інтегрований вебпроєкт і рефлексія щодо вибору технологій.	Репозиторій і історія змін, проєкт, усний захист, комп’ютерне тестування.

ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ

Процедура та регламент проведення контрольних заходів

Процедура проведення контрольних заходів регулюється «Положенням про систему поточного та підсумкового контролю, оцінювання знань та визначення рейтингу студентів». Фіксація поточного контролю здійснюється в «Електронному журналі обліку успішності академічної групи» за чотирибальною шкалою («2», «3», «4», «5»). У разі відсутності студента на занятті виставляється індекс «н». За результатами семестрової роботи в Журналі автоматично визначається кумулятивна підсумкова оцінка поточної роботи та ведеться облік пропусків.

Правило трьох тижнів: Усі пропущені заняття (лекційні, практичні) та отримані негативні оцінки («2») студенти зобов'язані відпрацювати **впродовж трьох наступних тижнів** з моменту виникнення заборгованості. У разі недотримання зазначеного терміну замість «н» виставляється оцінка **«0» (нуль балів) без права подальшого перездавання**. Відпрацьоване лекційне заняття фіксується літерою «в».

Умови допуску до екзамену: До складання семестрового екзамену допускаються здобувачі, які за підсумками поточного контролю набрали **не менше 35 балів** та відпрацювали всі пропущені аудиторні заняття. Студенти, які набрали **34 бали і менше**, не допускаються до складання підсумкового контролю та на підставі укладання додаткового договору зобов'язані пройти повторний курс вивчення дисципліни.

За підсумками складання екзамену здобувач максимально може набрати 40 балів. Мінімальний поріг успішного складання екзамену становить **25 балів**. Здобувачі, які під час екзамену отримали 24 бали і менше, вважаються такими, що не склали екзамен (отримують підсумковий статус FX), та повинні з'явитися на академічне перескладання згідно з графіком деканату.

Критерії оцінювання

«Відмінно» (90–100 балів, A): Здобувач демонструє всебічні, глибокі та системні знання теоретичного матеріалу з розробки вебдодатків, вебдизайну та тестування; вільно володіє технологіями HTML, CSS і JavaScript, принципами адаптивного дизайну, вебдоступності та взаємодії з Web API; самостійно проєктує інтерфейси вебдодатків, обґрунтовано обирає архітектурні підходи (MPA, SPA, SSR, SSG), створює якісний, валідний та структурований код;

безпомилково виконує практичні завдання, аргументовано пояснює прийняті технічні рішення та впевнено проводить тестування створених веб-додатків.

«Добре» (75–89 балів, B–C): Здобувач упевнено володіє навчальним матеріалом; правильно застосовує HTML, CSS і JavaScript для розв’язання типових практичних завдань; уміє створювати адаптивні вебінтерфейси, працювати з DOM, використовувати Web API та проводити базове тестування веб-додатків; допускає окремі несуттєві помилки в коді, проєктуванні або документації, які здатний самостійно виявити та виправити після зауважень.

«Задовільно» (60–74 бали, D–E): Здобувач володіє навчальним матеріалом на мінімально достатньому рівні; знає основні концепції веброзробки та тестування, створює нескладні вебсторінки й вебдодатки за зразком або з допомогою викладача; відчуває труднощі під час проєктування інтерфейсів, реалізації динамічної поведінки та тестування програмних рішень, проте здатний виконувати базові практичні завдання і пояснювати основні принципи їх роботи.

«Незадовільно» з можливістю повторного складання (35–59 балів, FX): Здобувач володіє лише окремими фрагментами навчального матеріалу; допускає суттєві помилки під час створення вебінтерфейсів і програмування клієнтської логіки; не може самостійно реалізувати або протестувати веб-додаток відповідно до поставлених вимог; має академічну заборгованість, але виконав мінімальний обсяг поточної роботи, необхідний для отримання права на повторне складання підсумкового контролю.

«Незадовільно» з обов’язковим повторним вивченням (0–34 бали, F): Здобувач не засвоїв основні поняття та методи розробки веб-додатків; не володіє базовими знаннями HTML, CSS і JavaScript; практичні завдання не виконані або виконані на неприйнятному рівні; не може продемонструвати досягнення програмних результатів навчання, не допущений до підсумкового контролю та підлягає повторному вивченню навчальної дисципліни.

Шкала оцінювання за ЄКТС

Шкала оцінювання знань за ЄКТС:

Оцінка за національною шкалою	Рівень досягнень, %	Шкала ECTS
Національна диференційована шкала		
Відмінно	90 – 100	A
Добре	83 – 89	B
	75 – 82	C
Задовільно	67 – 74	D
	60 – 66	E
Незадовільно	35 – 59	FX
	0 – 34	F
Національна недиференційована шкала		
Зараховано	60 – 100	-
Не зараховано	0 – 59	-

Студенти, які не з'явилися на заліки/екзамени без поважних причин, вважаються такими, що одержали незадовільну оцінку.

Порядок відпрацювання та підсумкового контролю

Пропущені заняття й негативні оцінки необхідно відпрацювати впродовж трьох наступних тижнів. За тривалої відсутності з поважних причин, підтверджених деканатом, строк відраховується з моменту повернення. За недотримання строку замість «н» у журналі виставляється «0» без права перездачі цього пропуску; відпрацьована лекція позначається «в».

Підсумковий контроль проводиться в СДО та/або комп’ютерній аудиторії. Залік у першому семестрі охоплює базові алгоритми, типи даних, розгалуження, цикли й масиви. Екзамен у другому семестрі охоплює функції, об’єкти, класи, асинхронність, Git і веброзробку та може містити комп’ютерне тестування й практичне завдання з написання або аналізу коду. Тривалість, дозволені засоби й критерії оприлюднюються в СДО до контролю.

РЕКОМЕНДОВАНІ ДЖЕРЕЛА ІНФОРМАЦІЇ

Основна література

1. Sinha A., Ranjan A., Battewad R. *JavaScript for Modern Web Development: Building a Web Application Using HTML, CSS, and JavaScript.* New Delhi : BPB Publications, 2023. 440 p.

2. Duckett J. *HTML & CSS: Design and Build Websites*. Indianapolis : John Wiley & Sons, 2022. 512 p.

3. Meyer E. A., Weyl E. *CSS: The Definitive Guide. Web Layout and Presentation.* 5th ed. Sebastopol: O'Reilly Media, 2023.

4. Flanagan D. *JavaScript: The Definitive Guide*. 7th ed. Sebastopol : O'Reilly Media, 2021. 704 p.

5. Purewal S. *Learning Web App Development: Build Quickly with Proven JavaScript Techniques.* Sebastopol : O'Reilly Media, 2020. 303 p.

6. Haverbeke M. *Eloquent JavaScript*. 4th ed. San Francisco : No Starch Press, 2024.

7. Robbins J. N. *Learning Web Design*. 6th ed. Sebastopol : O'Reilly Media, 2023.

8. Andrews M., Whittaker J. A. *How to Break Web Software: Functional and Security Testing of Web Applications and Web Services*. Upper Saddle River : Addison-Wesley, 2019. 240 p.

Електронні інформаційні ресурси

9. WHATWG. *HTML Living Standard: Edition for Web Developers* [Електронний ресурс]. URL: https://html.spec.whatwg.org/dev/ (дата звернення: 14.08.2026).

10. GeeksforGeeks. *Web Development Technologies.* [Електронний ресурс]. URL: https://www.geeksforgeeks.org/web-tech/web-technology (дата звернення: 14.08.2026).

11. MDN Web Docs. *Learn HTML: Structuring the Web* [Електронний ресурс]. URL: https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/Structuring_content (дата звернення: 14.08.2026).

12. MDN Web Docs. *CSS Layout* [Електронний ресурс]. URL: https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/CSS_layout (дата звернення: 14.08.2026).

13. Сучасний підручник з JavaScript [Електронний ресурс]. URL: https://uk.javascript.info/ (дата звернення: 14.08.2026).

14. Ecma International. *ECMAScript® 2026 Language Specification* [Електронний ресурс]. URL: https://tc39.es/ecma262/2026/multipage/ (дата звернення: 14.08.2026).

15. MDN Web Docs. *JavaScript Guide* [Електронний ресурс]. URL: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide (дата звернення: 14.08.2026).

16. OpenJS Foundation. *Node.js API Documentation* [Електронний ресурс]. URL: https://nodejs.org/api/ (дата звернення: 14.08.2026).

17. Git Project. *Git Reference* [Електронний ресурс]. URL: https://git-scm.com/docs (дата звернення: 14.08.2026).

18. MDN Web Docs. *Accessibility on the Web* [Електронний ресурс]. URL: https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/Accessibility (дата звернення: 14.08.2026).

19. Microsoft. *Playwright Documentation* [Електронний ресурс]. URL: https://playwright.dev/ (дата звернення: 14.08.2026).

20. Google. *Material Design 3* [Електронний ресурс]. URL: https://m3.material.io/ (дата звернення: 14.08.2026).