Національний технічний університет України
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Кафедра
Штучного
Інтелекту

ВЕБ-ОРІЄНТОВАНА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	***Перший (бакалаврський)***
Галузь знань	*12 Інформаційні технології*
Спеціальність	*122 Комп'ютерні науки* *124 Системний аналіз*
Освітня програма	*28343 Системи і методи штучного інтелекту* *5094 Системний аналіз і управління*
Статус дисципліни	*Вибіркова*
Форма навчання	*очна(денна)*
Рік підготовки, семестр	*3 курс, осінній семестр*
Обсяг дисципліни	*4 кредити ЄКТС/120 годин* *Лекції - 36 год, практ.(комп. практ.) - 18 год, СРС - 66 год*
Семестровий контроль/ контрольні заходи	*залік, МКР*
Розклад занять	*https://schedule.kpi.ua/*
Мова викладання	*Українська*
Інформація про керівника курсу / викладачів	Лектор: *Доцент, PhD, Гуськова Віра Геннадіївна, huskova.vira@lll.kpi.ua* Практичні: *асистент, Лисов Богдан Сергійович, bogukraine@gmail.com*
Розміщення курсу	https://classroom.google.com/

Програма навчальної дисципліни

1. Опис навчальної дисципліни, її мета, предмет вивчання та результати навчання

Web-орієнтована розробка програмного забезпечення — це сучасна галузь, що стрімко розвивається та охоплює різноманітні аспекти створення інтерактивних веб-систем та інформаційних платформ. Web-технології є фундаментом цифрової трансформації, що змінює всі сфери людської діяльності — від бізнесу та освіти до медицини, промисловості та державного управління. Сьогодні web-додатки забезпечують доступ до інформації та послуг у режимі реального часу, є основою електронної комерції, фінансових систем, соціальних мереж та хмарних обчислень.

Вивчення дисципліни дозволяє студентам набути ґрунтовних знань і практичних навичок щодо розробки динамічних сайтів, систем контролю версій, сучасних фреймворків та мов програмування для створення повнофункціональних веб-рішень.

Метою *опанування дисципліни є формування у студентів теоретичних знань і практичних вмінь щодо проектування, розробки та впровадження web-орієнтованих програмних систем, що відповідають сучасним вимогам функціональності, безпеки, масштабованості та ефективності.*

Предметом *навчальної дисципліни є теоретичні основи, методи, технології, стандарти та інструменти, які використовуються при розробці веб-додатків, зокрема побудова структури веб-сторінок, розробка клієнтської частини, використання систем управління версіями, засобів інтерактивної взаємодії, обробки даних та адаптивної верстки.*

Після засвоєння дисципліни ***«Web-орієнтована розробка програмного забезпечення»*** *студенти мають продемонструвати такі компетентності та результати навчання:*

Компетентності:

- *Знання та розуміння предметної області та розуміння професійної діяльності. (ЗК 3)*
- *Здатність генерувати нові ідеї (креативність). (ЗК 8)*
- *Здатність працювати в команді. (ЗК 9)*
- *Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління. (ФК 8)*
- *Здатність реалізувати багаторівневу обчислювальну модель на основі архітектури клієнт-сервер, включаючи бази даних, знань і сховища даних, виконувати розподілену обробку великих наборів даних на кластерах стандартних серверів для забезпечення обчислювальних потреб користувачів, у тому числі на хмарних сервісах. (ФК 9)*
- *Здатність застосовувати методології, технології та інструментальні засоби для управління процесами життєвого циклу інформаційних і програмних систем, продуктів і сервісів інформаційних технологій відповідно до вимог замовника. (ФК 10)*

Результати навчання:

- *Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук. (ПР 9)*
- *Володіти навичками управління життєвим циклом програмного забезпечення, продуктів і сервісів інформаційних технологій відповідно до вимог і обмежень замовника, вміти розробляти проектну документацію (техніко-економічне обґрунтування, технічне завдання, бізнес-план, угоду, договір, контракт). (ПР 11)*
- *Володіти мовами системного програмування та методами розробки програм, що взаємодіють з компонентами комп'ютерних систем, знати мережні технології, архітектури комп'ютерних мереж, мати практичні навички технології адміністрування комп'ютерних мереж та їх програмного забезпечення. (ПР 13)*
- *Розуміти концепцію інформаційної безпеки, принципи безпечного проектування програмного забезпечення, забезпечувати безпеку комп'ютерних мереж в умовах неповноти та невизначеності вихідних даних. (ПР 15)*

2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

Вивчення дисципліни базується на знаннях студентів з основних розділів алгоритмізації, програмування, структурування даних, побудови баз даних, а також основ комп'ютерних мереж, які забезпечують розуміння принципів передачі інформації в мережевих системах. Опанування цих базових курсів дозволяє студентам сформувати фундаментальні знання для подальшого засвоєння технологій проектування, розробки, тестування та впровадження web-орієнтованих програмних систем.

Дисципліна «Web-орієнтована розробка програмного забезпечення» є логічним продовженням дисциплін «Алгоритмізація та програмування», «Алгоритми і структури даних», «Системи баз даних», «Комп'ютерні мережі». Її вивчення формує у студентів комплексне бачення процесів створення сучасних web-додатків з урахуванням особливостей архітектури клієнт-сервер, роботи з мережевими протоколами, управління життєвим циклом програмного продукту.

Отримані знання та навички є основою для успішного вивчення подальших дисциплін освітньої програми, зокрема: «Управління IT-проєктами», «Крос-платформне програмування», «Розробка інтелектуальних систем», «Технології розробки програмного забезпечення», а також для виконання кваліфікаційної бакалаврської роботи.

Компетенції, здобуті під час вивчення дисципліни, застосовуються при розробці реальних web-систем у практичній діяльності та в рамках командних проєктів студентів.

3. Зміст навчальної дисципліни

Розділ 1. Вимоги до розробки веб-застосунку
- *Тема 1.1. Загальні вимоги до створення веб-застосунку*
- *Тема 1.2. Функціональні вимоги*
- *Тема 1.3. Нефункціональні вимоги*
- *Тема 1.4. Система контроля версій/Відслідковування статусу задач задач*

Розділ 2. Основи роботи з HTML: структура, елементи та інтерактивність
- *Тема 2.1. Основи Основи HTML та роль у веб розробці*
- *Тема 2.2. Робота з базовими елементами HTML: текст, посилання, таблиці, зображення*
- *Тема 2.3. Форми та взаємодія з користувачем, Робота з мультимедіа*

Розділ 3. Каскадні таблиці стилів (CSS): основи, техніки та сучасні підходи
- *Тема 3.1. Основи та базові концепції*
- *Тема 3.2. CSS техніки*
- *Тема 3.3. Оптимізація та сучасні методи*
- *Тема 3.4. Інструменти та діагностика*

Розділ 4. Основи, сучасні концепції та практичне застосування
- *Тема 4.1. Базові концепції*
- *Тема 4.2. Просунуті концепції: ООП, модулі, асинхроність, події*
- *Тема 4.3. Робота з браузером, DOM*
- *Тема 4.4. Робота з мережею: HTTP, API*
- *Тема 4.5. Сучасні концепції: ES6+, React*

4. Навчальні матеріали та ресурси

Основна література:

1. *Яценко В.І. Основи веб-програмування. Навч. посіб. – К.: Центр учбової літератури, 2020. – 304 с.*
2. *Гусєв В.В., Забарний О.В. Веб-програмування: HTML, CSS, JavaScript. Навчальний посібник. – Х.: ХНУРЕ, 2022. – 210 с.*
3. *Кузнецова К.О. Сучасні веб-технології: навчальний посібник. – К.: НАУ, 2020. – 185 с.*
4. *Матеріали дистанційного курсу дисципліни «Web-орієнтована розробка ПЗ» на платформі «Сікорський» КПІ.*
5. *Young A., Meck B., Cantelon M., Node.js in Action / Young A., Meck B., Cantelon M. Manning, 2018. 432 p.*

Допоміжна:

1. *Bose T. Modern Full-Stack Development: Using TypeScript, React, Node.js, Webpack, and Docker. – Apress, 2021. – 337 p.*
2. *Official site of AngularJS. – Access mode https://angularjs.org/ Access date: 13.05.2020.*
3. *Flanagan D. JavaScript: The Definitive Guide, 7th Edition. – O'Reilly Media, 2020. – 706 p.*
4. *Official ReactJS website. – Access mode https://uk.reactjs.org/Access date: 13.05.2020.*
5. *The official website of Node.JS. Access mode https://nodejs.org/en/Access date: 13.05.2020.*

Інформаційні ресурси:

1. *MDN Web Docs – https://developer.mozilla.org це Офіційна довідка з HTML, CSS, JavaScript, API та веб-стандартів. Доступно онлайн та рекомендується для постійного використання.*
2. *freeCodeCamp – Responsive Web Design Certification – https://www.freecodecamp.org/learn/responsive-web-design - це введення в HTML, CSS, адаптивну верстку. Пояснення базових понять – перед виконанням лабораторних робіт 2–3.*
3. *JavaScript.info – https://javascript.info/ - це повноцінний підручник з JavaScript. Містить практичні приклади, завдання для самоперевірки.*
4. *Eloquent JavaScript (3rd ed.) – https://eloquentjavascript.net - це онлайн-книга з теорії програмування на JavaScript. Рекомендовано після базового ознайомлення з синтаксисом JS.*
5. *GitHub Docs: Git Basics – https://docs.github.com/en/get-started/using-git - це базова документація з використання Git. Актуально для командної роботи та підготовки web-проєктів. Ознайомлення рекомендовано протягом виконання лабораторних робіт.*

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

Навчальна дисципліна охоплює 36 годин лекційних занять та 18 годин комп'ютерних практичних занять, а також виконання модульної контрольної роботи, яка складається з чотирьох частин відповідно до основних тематичних розділів курсу.

Практичні заняття з дисципліни проводяться з метою закріплення теоретичних положень курсу «Web-орієнтована розробка програмного забезпечення» та набуття студентами практичних навичок у розробці web-додатків, починаючи від побудови структури HTML-документів, стилізації інтерфейсів з використанням CSS до організації інтерактивної поведінки сторінок за допомогою JavaScript, роботи із системами контролю версій.

Під час практичних занять студенти працюють під керівництвом викладача над виконанням індивідуальних та групових завдань, які дозволяють їм набувати досвіду самостійного проектування, розробки, тестування та налагодження web-додатків. Виконання завдань супроводжується обговоренням рішень, аналізом типових помилок, розглядом альтернативних підходів до розробки та оптимізації web-систем.

Лекційні заняття

№ з/п	*Назва теми лекції та перелік основних питань*
1	***Розділ 1. Тема 1.1. Загальні вимоги до створення веб-застосунку*** *Розглядаються основні вимоги до створення веб-застосунків, включаючи визначення цілей, планування архітектури та вибір технологій. Обговорюються принципи адаптивності, кросбраузерності, зручності користувацького інтерфейсу та забезпечення безпеки. Також акцентується увага на оптимізації продуктивності та інтеграції з іншими сервісами.*
2	***Розділ 1. Тема 1.2. Функціональні вимоги*** *Розглядаються функціональні вимоги до програмного забезпечення, які визначають основну поведінку та можливості системи. Обговорюється, як правильно формулювати ці вимоги для забезпечення відповідності очікуванням користувачів і досягнення цілей проєкту.*
3	***Розділ 1. Тема 1.3. Нефункціональні вимоги*** *Розглядаються нефункціональні вимоги, які визначають якісні характеристики системи, такі як продуктивність, безпека, масштабованість і зручність використання. Обговорюється їхня роль у забезпеченні стабільної роботи веб-застосунку та відповідності очікуванням користувачів і бізнес-цілей.* ***Розділ 1. Тема 1.4. Система контроля версій/Відслідковування статусу задач задач*** *Також розглядаються принципи роботи систем контролю версій та інструментів для відстеження статусу задач, які забезпечують ефективне управління розробкою програмного забезпечення, командною співпрацею та контролем змін у коді.*
4	***Розділ 2. Тема 2.1. Основи HTML та роль у веб розробці*** *У цій лекції розглядаються основи HTML, його структура та ключові елементи, які використовуються для створення веб-сторінок. Обговорюється роль HTML у веб-розробці як основи для побудови структури контенту, взаємодії з CSS і JavaScript, а також забезпечення доступності інформації для користувачів і пошукових систем.*
5	***Розділ 2. Тема 2.2. Робота з базовими елементами HTML: текст, посилання, таблиці, зображення*** *Розглядається використання базових елементів HTML для створення та форматування тексту, додавання посилань, створення таблиць і вставлення зображень. Обговорюються особливості їхньої структури, атрибути та практичне застосування для побудови простих веб-сторінок.*

6	***Розділ 2. Тема 2.3. Форми та взаємодія з користувачем, Робота з мультимедіа*** *Розглядається створення форм в HTML для взаємодії з користувачами, включаючи основні елементи форм, їх атрибути та принципи валідації даних. Також вивчаються способи роботи з мультимедійними елементами, такими як аудіо, відео та інтерактивні компоненти, для створення насиченого користувацького досвіду.*
7	***Розділ 3. Тема 3.1. Основи та базові концепції*** *розглядається використання базових елементів HTML для створення та форматування тексту, додавання посилань, створення таблиць і вставлення зображень. Обговорюються особливості їхньої структури, атрибути та практичне застосування для побудови простих веб-сторінок.*
8	***Розділ 3. Тема 3.2. CSS техніки. Частина 1. Flexbox*** *Розглядається використання Flexbox для побудови гнучких та адаптивних макетів. Обговорюються основні властивості, принципи вирівнювання елементів і налаштування їхньої поведінки у контейнерах, що дозволяє створювати зручні та сучасні інтерфейси.*
9	***Розділ 3. Тема 3.2. CSS техніки. Частина 2. Grid та медіа-запити*** *Ця лекція присвячена технікам роботи з CSS Grid для побудови складних макетів і управління розташуванням елементів у двовимірному просторі. Крім того, обговорюються медіа-запити як інструмент для розробки адаптивних стилів, що забезпечують коректне відображення веб-сторінок на різних пристроях.*
10	***Розділ 3.Тема 3.3. Оптимізація та сучасні методи. Частина 1. Змінні CSS та препроцесори*** *У цій лекції розглядається використання змінних у CSS для спрощення управління стилями та впровадження тематизації, зокрема реалізації dark mode/light mode. Також обговорюються препроцесори Sass і Less, їхні можливості для гніздування стилів, використання міксинів та функцій для оптимізації коду.*
11	***Розділ 3.Тема 3.3. Оптимізація та сучасні методи. Частина 2. Методології CSS та сучасні ефекти*** *Ця лекція присвячена методологіям організації CSS-коду, таким як BEM і SMACSS, що сприяють структурованості та масштабованості проєктів. Крім того, розглядається використання CSS-змішувань (blend modes) для створення ефектів накладання кольорів і зображень.*
12	***Розділ 3. Тема 3.4. Інструменти та діагностика*** *Розглядаються інструменти розробника, зокрема DevTools, для аналізу та оптимізації CSS, а також методи тестування стилів у різних браузерах для забезпечення кросбраузерності. Обговорюється виявлення і виправлення помилок, а також інтеграція CSS із JavaScript для маніпуляції стилями, роботи з DOM і реалізації інтерактивних елементів.*
13	***Розділ 4. Тема 4.1. Базові концепції.*** *У цій лекції розглядаються базові концепції JavaScript, зокрема основи синтаксису, типи змінних і даних, оператори, умови та цикли. Обговорюються різні способи оголошення та використання функцій, включаючи стрілкові функції та замикання. Також вивчається робота з об'єктами та масивами, а саме їх структура, методи обробки та ітерації.*
14	***Розділ 4. Тема 4.2. Просунуті концепції: ООП, модулі, асинхроність, події. Частина 1.*** *У цій лекції розглядаються основи ООП у JavaScript, включаючи класи, прототипи, інкапсуляцію, наслідування та поліморфізм. Також обговорюються модулі, їхнє використання для організації коду через імпорт і експорт у сучасних браузерах і Node.js.*

15	***Розділ 4. Тема 4.2. Просунуті концепції: ООП, модулі, асинхроність, події. Частина 2.*** *Ця лекція присвячена роботі з подіями, їх створенню, обробці та делегуванню для покращення взаємодії з користувачем. Крім того, вивчаються основи асинхронності, використання промісів, async/await, обробка помилок у асинхронному коді та принципи роботи Event Loop.*
16	***Розділ 4. Тема 4.3. Робота з браузером, DOM*** *У цій лекції розглядається робота з браузером, зокрема взаємодія з DOM для навігації, маніпуляції елементами та зміни стилів за допомогою JavaScript. Обговорюються можливості браузерних API, включаючи роботу з LocalStorage, Cookies, геолокацією та WebSockets. Також вивчаються методи валідації даних у формах і їхнє відправлення через JavaScript з використанням fetch або AJAX.*
17	***Розділ 4. Тема 4.4. Робота з мережею: HTTP, API*** *У цій лекції розглядаються основи роботи з мережею в JavaScript, включаючи використання fetch для виконання HTTP-запитів і обробку API-відповідей у форматі JSON. Обговорюється робота з WebSocket для реалізації реального часу, а також інтеграція зі сторонніми API, такими як OpenWeather або Google Maps, з прикладами створення власного міні-API для навчання.*
18	***Розділ 4. Тема 4.5. Сучасні концепції: ES6+, React*** *У цій лекції розглядаються сучасні можливості JavaScript, включаючи ES6+ функціональність, такі як рест- і спред-оператори, деструктуризація, генератори та ітератори. Обговорюються прототипи, нюанси контексту виконання this і методи call, apply, bind. Також дається огляд популярних фреймворків і бібліотек, таких як React, Vue, Angular і jQuery, для роботи з DOM.*

6. Самостійна робота здобувача вищої освіти

Самостійна робота студентів з дисципліни «Web-орієнтована розробка програмного забезпечення» включає підготовку до лекційних та практичних занять, вивчення літературних і електронних джерел, ознайомлення з документацією сучасних web-технологій, підготовку до виконання лабораторних робіт, модульної контрольної роботи та заліку. Студенти самостійно опрацьовують теоретичні основи HTML, CSS, JavaScript, систем контролю версій, готують та оформлюють звіти за лабораторними роботами, а також відпрацьовують практичні навички розробки web-додатків.

№ з/п	Вид самостійної роботи	Кількість годин СРС
1	Опрацювання навчального матеріалу, викладеному на лекціях 1-3	4
2	Опрацювання навчального матеріалу лекцій 1-3, винесеного на самостійне вивчення	4
3	Підготовка до лабораторного заняття 1	3
4	Оформлення звітів до лабораторної роботи 1	2
5	Опрацювання навчального матеріалу, викладеному на лекціях 4-12	8
6	Опрацювання навчального матеріалу лекцій 4-12, винесеного на самостійне вивчення	8

7	*Підготовка до лабораторного заняття 2,3*	6
8	*Оформлення звітів до лабораторної роботи 2,3*	4
9	*Опрацювання навчального матеріалу, викладеному на лекціях 12-18*	4
10	*Опрацювання навчального матеріалу лекцій 12-18, винесеного на самостійне вивчення*	4
11	*Підготовка до лабораторного заняття 4*	3
12	*Оформлення звітів до лабораторної роботи 4*	2
13	*Підготовка до Фінального тесту*	6
14	*Підготовка до заліку*	8

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Відвідування занять. *Пропуски лекційних і практичних занять не впливають безпосередньо на кількість набраних балів, оскільки підсумкова оцінка базується на результатах виконаних завдань і контрольних заходів. Водночас, активна участь у роботі на заняттях, демонстрація виконаних завдань, обговорення результатів, індивідуальні відповіді, а також захист лабораторних робіт оцінюються під час аудиторної взаємодії з викладачем.*

Участь у практичних заняттях. *Студент зобов'язаний готуватись до практичних занять, ознайомлюватися з матеріалами теми, попередньо вивчати приклади коду та документацію. Участь у заняттях передбачає презентацію результатів лабораторних робіт, аргументований захист рішень, а також обговорення технічних питань.*

Пропущені контрольні заходи. *Студенти, які з поважної причини (медичні довідки, академічна мобільність тощо) пропустили контрольну роботу або захист лабораторної, мають право відпрацювати ці заходи у визначений викладачем термін. Детальніше: https://kpi.ua/files/n3277.pdf*

Процедура оскарження результатів. *Студент має право подати обґрунтовану апеляцію на результати будь-якого контрольного заходу, якщо він не згоден з виставленою оцінкою. Звернення розглядаються відповідно до встановленої процедури в межах кафедри чи освітньої програми.*

Календарний контроль. *Проводиться двічі на семестр для моніторингу прогресу студентів у виконанні вимог дисципліни, своєчасного виявлення труднощів у засвоєнні матеріалу та коригування індивідуальних планів навчання.*

Критерій	***Перший календарний контроль***	***Другий календарний контроль***
Термін календарного контролю	*Тиждень 8*	*Тиждень 14*
Умови отримання позитивної оцінки	*Поточний рейтинг ≥ 20 балів*	*Поточний рейтинг ≥ 40 балів*

Академічна доброчесність. *Політика та принципи академічної доброчесності визначені у розділі 3 Кодексу честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського». Студенти зобов'язані дотримуватись академічної доброчесності під час виконання лабораторних, контрольних робіт, захисту індивідуальних завдань та підсумкової атестації. Детальніше: https://kpi.ua/code.*

Норми етичної поведінки. *Норми етичної поведінки студентів і працівників визначені у розділі 2 Кодексу честі НТУУ «КПІ ім. Ігоря Сікорського». Студенти зобов'язані дотримуватись взаємоповаги, коректності під час навчального процесу, зокрема при роботі в командах над спільними web-проєктами. Детальніше: https://kpi.ua/code.*

Інклюзивне навчання. *Засвоєння знань та умінь з дисципліни є доступним для більшості здобувачів з особливими освітніми потребами. Особливі умови можуть бути погоджені індивідуально за необхідності.*

Навчання іноземною мовою. *В рамках виконання завдань студентам може бути рекомендовано використовувати англомовну технічну документацію, стандарти, документацію web-фреймворків, онлайн-курси та навчальні матеріали.*

Призначення заохочувальних та штрафних балів. *Відповідно до Положення про систему оцінювання результатів навчання, сума всіх заохочувальних балів не може перевищувати 10% рейтингової шкали оцінювання. Заохочувальні бали можуть нараховуватись за участь у хакатонах, конкурсах, технічних конференціях, виконання додаткових індивідуальних завдань за погодженням із викладачем.*

Заохочувальні бали

Критерій	***Заохочувальні бали***
Написання тез, статей, оформлення проєктної роботи у вигляді наукової роботи або практичного web-проєкту для участі у конкурсах студентських наукових робіт (за тематикою дисципліни)	*5*
Участь у хакатонах, міжнародних, всеукраїнських конкурсах, конференціях, виставках або семінарах, пов'язаних із тематикою web-розробки	*5*

Підготування до семінарських занять та контрольних заходів здійснюється під час самостійної роботи студентів з можливістю консультування з викладачем у визначений час консультацій або за допомогою електронного листування (електронна пошта, месенджери, гуглкласрум).

8. Види контролю та рейтингова система оцінювання результатів навчання (РСО)

Семестровий контроль проводиться у вигляді заліку. Для оцінювання результатів навчання застосовується 100-бальна рейтингова система та університетська шкала оцінювання.

Поточний контроль *включає: виконання та захист лабораторних робіт, участь у практичних заняттях, індивідуальні завдання, самостійне виконання підготовчих робіт, заключний тест за матеріалами курсу.*

Календарний контроль *проводиться двічі на семестр з метою моніторингу виконання студентами вимог силабусу:*

- *Перший календарний контроль — на 8 тижні (поточний рейтинг ≥ 20 балів).*
- *Другий календарний контроль — на 14 тижні (поточний рейтинг ≥ 40 балів).*

Семестровий контроль *— залік.*

Таблиця відображає систему оцінювання в дисципліні «Web-орієнтована розробка ПЗ», що базується на п'яти контрольних заходах:

№	Контрольний захід	Дедлайн виконання	Мін. оцінка	Макс. оцінка
1	**Лабораторна робота 1.** Розробка концепції веб-застосунку	26.09.2025	5	10
2	**Лабораторна робота 2.** Верстка основної структури веб-сторінки	17.10.2025	10	20
3	**Лабораторна робота 3.** Стилізація веб-сторінки за допомогою CSS	14.11.2025	15	20
4	**Лабораторна робота 4.** Реалізація інтерактивності за допомогою JavaScript	12.12.2025	15	20
5	Заключний тест за матеріалами курсу	19.12.2025	15	30
Разом:			**60**	**100**

Зверніть увагу, що за несвоєчасне подання звіту лабораторної роботи, оцінка може бути знижена, але не більше ніж на 20% від загальної оцінки по роботі.

Оцінювання контрольних заходів

- ***Лабораторні роботи*** *оцінюються за якістю реалізації завдань, повнотою функціональності, коректністю коду, оформленням звітів, дотриманням термінів виконання.*
- ***Заключний тест*** *містить теоретичні питання, тестові завдання та практичні завдання з HTML, CSS, JavaScript, архітектури web-додатків.*

Розгорнуті ***критерії оцінювання*** *для контрольних заходів дисципліни «Web-орієнтована розробка програмного забезпечення». Критерії враховують обсяг, складність і ваговий бал кожної роботи.*

Назва заходу	Кількість балів	Критерії оцінювання

Лабораторна робота 1. Розробка концепції веб-застосунку	*10 балів*	***0–5 балів** — робота не здана або дуже поверхневий рівень: відсутній аналіз функцій, немає структурованої специфікації.* ***6 балів** — базовий опис і приклади вимог, часткова відповідність технічному завданню.* ***7 балів** — є структура функціональних і нефункціональних вимог, наявна спроба сегментації користувачів.* ***8 балів** — повна, структурована концепція із базовим UI-прототипом.* ***9–10 балів** — глибокий аналіз вимог, обґрунтовані рішення, якісна презентація з урахуванням UX-факторів.*
Лабораторна робота 2. Верстка основної структури веб-сторінки	*20 балів*	***0–11 балів** — неповна верстка, відсутні семантичні теги, порушена структура документа.* ***12–13 балів** — базова структура побудована, але семантика неповна, є помилки в розмітці.* ***14–15 балів** — верстка структурована, використані основні теги, частково дотримано принципи доступності.* ***16–17 балів** — адаптивна структура, коректне форматування, додано базову навігацію.* ***18–20 балів** — повна семантична, адаптивна верстка з урахуванням SEO та A11Y; чистий код.*
Лабораторна робота 3. Стилізація веб-сторінки за допомогою CSS	*20 балів*	***0–11 балів** — стилізація мінімальна, не використано базові властивості оформлення.* ***12–13 балів** — часткове стилізування, але відсутня адаптивність чи правильна сітка.* ***14–15 балів** — нормальне оформлення, є базовий responsive, застосовані Flex/Grid.* ***16–17 балів** — якісна адаптивна стилізація, структурована система класів, застосовані медіа-запити.* ***18–20 балів** — комплексна стилізація з кастомним дизайном, ефектами (анімації/теми), відповідність макету.*
Лабораторна робота 4. Реалізація	*20 балів*	***0–11 балів** — код непрацюючий або містить суттєві помилки.*

інтерактивності за допомогою JavaScript		***12–13 балів** — реалізовані прості події, базова логіка, але є баги.* ***14–15 балів** — функціонал працює, використані події, змінні, цикли; DOM-маніпуляції обмежені.* ***16–17 балів** — реалізовано взаємодію з формами, DOM API, збереження у localStorage.* ***18–20 балів** — реалізовано інтерактивний функціонал з promise, модулями, валідацією; чистий, читабельний код.*
Заключний тест за матеріалами курсу	*30 балів*	***0–17 балів** — неуспішне проходження: менше 60% правильних відповідей, не пройдені ключові теми.* ***18–21 балів** — засвоєно базовий рівень матеріалу (60–70% правильних відповідей).* ***22–24 балів** — твердий рівень знань, правильні відповіді з більшості тем.* ***25–27 балів** — висока обізнаність, успішне проходження теорії та практичних кейсів.* ***28–30 балів** — відмінне засвоєння всіх тем: складні логічні задачі, повне розуміння взаємозв'язків.*

** Оцінювання здійснюється на основі об'єктивних критеріїв. Студент має право на зворотній зв'язок та апеляцію згідно з політикою університету.*

Для отримання заліку «автоматом» необхідно:

- *набрати не менше 60 балів за семестр;*
- *успішно захистити всі лабораторні роботи;*
- *скласти заключний тест;*
- *виконати індивідуальні завдання, якщо вони були задані.*

Студенти, які не набрали необхідного мінімуму або бажають підвищити оцінку, мають можливість скласти додаткову залікову контрольну роботу у вигляді письмового тестування або виконання комплексного практичного завдання з розробки невеликого web-проєкту.

Шкала переведення рейтингових балів в оцінки:

Кількість балів	***Оцінка***
100–95	*Відмінно*
94–85	*Дуже добре*
84–75	*Добре*

74–65	*Задовільно*
64–60	*Достатньо*
Менше 60	*Незадовільно*

9. Додаткова інформація з дисципліни (освітнього компонента)

Перелік питань, що виносяться на семестровий контроль, наведено у Додатку А.

Методи та форми навчання включають не лише традиційні університетські лекції та лабораторні заняття, але й командну роботу, обговорення кейсів, презентації результатів проєктів, активні дискусії. В курсі широко використовуються стратегії активного навчання, зокрема:

- ***проектно-орієнтоване навчання*** *(створення web-додатків у рамках реальних або змодельованих технічних завдань),*
- ***дослідницький підхід*** *(аналіз сучасних web-технологій, тестування інструментів),*
- ***кейс-методи*** *(розбір вимог до створення веб-продукту),*
- ***візуалізація*** *через електронні презентації, діаграми, розробку інтерфейсів,*
- ***інформаційно-комунікаційні технології*** *(редактори коду, браузерні інструменти розробника, системи керування версіями).*

Комунікація з викладачем здійснюється через ***інформаційну систему "Електронний кампус"****, платформу дистанційного навчання* ***"Гугл класрум"****, а також за допомогою електронної пошти, Telegram або веб-ресурсу викладача.*

Матеріально-технічне забезпечення дисципліни

Для ефективного вивчення дисципліни ***«Web-орієнтована розробка програмного забезпечення»*** *необхідне наступне матеріально-технічне забезпечення:*

- ***Комп'ютери або ноутбуки*** *з доступом до Інтернету;*
- ***Браузери*** *з інструментами розробника (Google Chrome, Mozilla Firefox, тощо);*
- ***Середовище розробки коду****: Visual Studio Code (безкоштовне);*
- ***Інструменти для веб-розробки****: Live Server, Prettier, ESLint (плагіни для VS Code);*
- ***Графічний редактор*** *для прототипування (наприклад, Figma – безкоштовний тариф);*
- ***Системи контролю версій****: Git, GitHub (безкоштовний доступ для студентів);*
- ***Платформа дистанційного навчання****: "Гугл класрум" та "Електронний кампус КПІ".*

Використовується ***виключно безкоштовне програмне забезпечення або таке, що має відкриту ліцензію****, або те, до якого студенти мають гарантований доступ у рамках інституційної політики.*

Факультативне навчання.
Для поглиблення знань з web-розробки студентам можуть рекомендуватися наступні онлайн-курси:

- *Bootstrap 5 Exercises - https://www.w3schools.com/bootstrap5/bootstrap_exercises.php*
- *JavaScript Algorithms and Data Structures (freeCodeCamp) - https://www.freecodecamp.org/learn/javascript-algorithms-and-data-structures/*
- *Frontend Development with React (Coursera) - https://www.coursera.org/professional-certificates/meta-front-end-developer*
- *Responsive Web Design Certification (freeCodeCamp) - https://www.freecodecamp.org/learn/2022/responsive-web-design/*

Визнання результатів неформальної освіти.
За бажанням студента результати проходження сертифікованих онлайн-курсів можуть бути враховані у системі оцінювання згідно з Положенням про порядок визнання результатів навчання, набутих студентами КПІ ім. Ігоря Сікорського у неформальній/неформальній освіті: https://osvita.kpi.ua/node/179

Робочу програму навчальної дисципліни (силабус):
Складено:
доцент кафедри штучного інтелекту,
PhD, Гуськова Віра Геннадіївна
асистент кафедри штучного інтелекту,
Лисов Богдан Сергійович

***Ухвалено** кафедрою штучного інтелекту (протокол №14 від 24.06.2025)*
***Погоджено** Методичною комісією НН ІПСА (протокол №7 від 25.06.2025)*

Перелік питань, які виносяться на семестровий контроль:
Додаток А

Тема 1. Розробка концепції веб-застосунку

1. *Що таке веб-застосунок і як він відрізняється від веб-сайту?*
2. *Які існують типи веб-застосунків?*
3. *Поясніть клієнт-серверну архітектуру у веб-розробці.*
4. *Основні етапи життєвого циклу веб-додатку.*
5. *Як сформулювати функціональні та нефункціональні вимоги до веб-застосунку?*
6. *Що таке специфікація вимог?*
7. *Аналіз цільової аудиторії: навіщо і як?*
8. *Відмінності між UX та UI.*
9. *Що таке user flow та як його побудувати?*
10. *Які засоби використовують для створення прототипів?*
11. *Які існують принципи побудови логіки веб-застосунку?*
12. *Яка роль бізнес-логіки в архітектурі проєкту?*
13. *Як визначити обсяг MVP та пріоритети реалізації функцій?*

Тема 2. Верстка основної структури веб-сторінки

9. *Базова структура HTML-документа.*
10. *Семантичні HTML-елементи: приклади і роль.*
11. *Основні HTML-елементи.*
12. *Організація навігації на сторінці.*
13. *Призначення атрибутів: id, class, href, alt.*
14. *Блочні vs. рядкові елементи.*
15. *Вкладеність HTML-елементів.*
16. *Як використовуються таблиці та списки в HTML?*
17. *Роль форм і їх елементів (input, select, textarea).*
18. *Як організувати доступність (a11y) у HTML?*
19. *Які елементи використовують для мультимедіа (audio, video)?*
20. *Що таке HTML-шаблони та як їх застосовують?*
21. *Як створити SEO-дружню структуру сторінки?*

Тема 3. Стилізація веб-сторінки за допомогою CSS

17. *Що таке каскадність і специфічність у CSS?*
18. *Основні типи селекторів у CSS.*
19. *Flexbox та Grid Layout — порівняння.*
20. *Media queries і адаптивність.*
21. *Box Model: складові та приклади.*
22. *Псевдокласи і псевдоелементи.*
23. *Вбудовані, внутрішні та зовнішні стилі.*
24. *Темізація інтерфейсу (dark/light).*
25. *CSS-змінні (custom properties).*
26. *Анімації в CSS: transition, keyframes.*
27. *Як працює позиціонування елементів?*
28. *Роль z-index та контексту накладання.*
29. *Способи приховування елементів (display, visibility, opacity).*

Тема 4. Реалізація інтерактивності за допомогою JavaScript

27. *DOM і його маніпуляція.*
28. *Обробка подій через addEventListener.*
29. *var, let, const: різниця.*
30. *Замикання, функції, стрілкові функції.*
31. *fetch API: приклади.*
32. *Promise та async/await.*
33. *localStorage, sessionStorage.*
34. *Валідація форм.*
35. *Обробка помилок (try/catch).*
36. *Маніпуляції класами через JS.*
37. *Делегування подій.*
38. *JSON і взаємодія з бекендом.*
39. *Основи об'єктно-орієнтованого підходу в JavaScript.*

Тема 5. Підсумкове проєктування та оцінювання знань

39. *Як реалізувати структуру MVP?*
40. *Методи деплою веб-додатку.*
41. *Основи Git (init, commit, push).*
42. *Якість коду: валідатори, ESLint.*
43. *Критерії ефективності (швидкість, доступність).*
44. *Кросбраузерність.*
45. *Тестування UI/UX.*
46. *PWA — особливості.*
47. *Популярні фреймворки (React, Vue): переваги.*
48. *Інструменти оптимізації швидкодії сторінки.*
49. *Основи безпеки фронтенду (CORS, XSS, CSRF).*
50. *Що таке веб-доступність (WCAG)?*
51. *Як працювати з devtools браузера для налагодження?*