

## СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

<b>Назва дисципліни</b>	<b>WEB-ПРОГРАМУВАННЯ</b>
<b>Рівень вищої освіти</b>	Перший (бакалаврський)
<b>Форма здобуття освіти</b>	Денна
<b>Семестр</b>	V
<b>Пререквізити дисципліни</b>	Передбачається, що студент має присвоєні кредити з дисципліни «Основи програмування», «Алгоритми та структури даних»
<b>Обсяг дисципліни</b>	кредити – 4,0 загальний обсяг годин – 120 лекцій – 10
<b>Форма звітності</b>	Залік
<b>Викладачі</b>	Командирчик Андрій Васильович, викладач
<b>Контактна інформація</b>	Телефон: +38 (096) 885-29-51 E-mail: andreykomandirchuck@gmail.com

### ЗАГАЛЬНИЙ ОПИС ДИСЦИПЛІНИ

Навчальна дисципліна «Веб-програмування» надає студентам глибокі знання та практичні навички у створенні веб-додатків і веб-сайтів. Курс охоплює основи фронтенд і бекенд розробки, включаючи HTML, CSS, JavaScript, а також серверні технології та бази даних, що дозволяють студентам розуміти весь цикл розробки веб-програм.

### МЕТА НАВЧАННЯ

Основними цілями курсу є: – якісна підготовка студентів у галузі веб-програмування; – знайомство з основними технологіями та інструментами для створення веб-додатків; – розуміння архітектури веб-додатків та принципів їх роботи; – розвиток навичок роботи з сучасними фреймворками та бібліотеками.

Очікувані результати навчання: знати: – основні технології веб-програмування (HTML, CSS, JavaScript); – принципи роботи клієнт-серверної архітектури; – основи роботи з базами даних та мовою SQL; – популярні фреймворки для фронтенд та бекенд розробки (наприклад, React, Node.js).

вміти: – створювати та оформлювати веб-сторінки з використанням HTML і CSS; – реалізовувати інтерактивність на веб-сторінках за допомогою JavaScript; – розробляти прості серверні додатки; – використовувати бази даних для зберігання та обробки даних.

## ТЕМАТИЧНИЙ ПЛАН

1. Вступ до веб-програмування
  - 1.1. Що таке веб-програмування?
  - 1.2. Основні технології веб-розробки.
  - 1.3. Клієнт-серверна архітектура.
2. Фронтенд технології
  - 2.1. Основи HTML: структура веб-сторінки.
  - 2.2. Стилзація за допомогою CSS: селектори, властивості, медіа-запити.
  - 2.3. JavaScript: синтаксис, типи даних, функції.
  - 2.4. DOM-маніпуляції та події.
3. Основи веб-дизайну
  - 3.1. Принципи адаптивного дизайну.
  - 3.2. Використання фреймворків CSS (Bootstrap).
  - 3.3. Основи UX/UI дизайну.
4. Бекенд технології
  - 4.1. Основи серверного програмування (Node.js).
  - 4.2. Основи роботи з API.
  - 4.3. Робота з базами даних: MongoDB, SQL.
5. Веб-фреймворки
  - 5.1. Основи роботи з React (фронтенд).
  - 5.2. Основи роботи з Express (бекенд).
  - 5.3. Інтеграція фронтенду та бекенду.
6. Безпека веб-додатків
  - 6.1. Основи веб-безпеки.
  - 6.2. Запобігання атакам: XSS, CSRF, SQL Injection.

### 6.3. Аутентифікація та авторизація користувачів.

#### **ВИДИ РОБІТ І ЗАВДАНЬ. ІНСТРУМЕНТАРІЙ НАВЧАННЯ**

Курс включає лекційні та лабораторні заняття, а також самостійну роботу. Студенти отримують бали за засвоєння теоретичного матеріалу на лекціях, виконання лабораторних завдань, підготовку звітів з виконаних робіт, захист проектів та індивідуальні завдання.

Основним інструментальним забезпеченням є:

- Текстові редактори (Visual Studio Code, Sublime Text).
- Браузери для тестування (Chrome, Firefox).
- Серверне програмне забезпечення (Node.js, Express).
- Бази даних (MongoDB, MySQL).

#### **ПОЛІТИКА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

Відвідування занять є одним із способів накопичення балів. Проте, значна увага приділятиметься насамперед самостійній роботі, під час якої здобувачі освіти матимуть більше можливостей розширити, поглибити та удосконалити знання й уміння, набуті під час аудиторних занять та консультацій чи самостійного опанування адаптованих викладачем навчальних матеріалів – конспектів лекцій, методичних роз'яснень щодо ходу виконання практичних завдань. Формами звітності за самостійну роботу виступатимуть звіти з результатами виконаних завдань. Вимоги до самостійної роботи студента включають систематичність, вчасність та якість виконання завдань. Здобувачі освіти повинні дотримуватися встановлених термінів здачі робіт та виконувати настанови викладача щодо їх структури та формату.

Для забезпечення політики академічної доброчесності, студентам необхідно самостійно виконувати завдання, уникаючи плагіату, фальсифікації даних та списування. Використання інформаційних джерел повинно бути належним чином оформлене згідно з вимогами наукового стилю та стандартами цитування. В разі виявлення порушень академічної доброчесності можуть застосовуватися відповідні санкції, включаючи зниження оцінки або навіть скасування результатів роботи.

## СИСТЕМА ОЦІНЮВАННЯ

Студент може отримати до 45 балів за виконання й захист завдань лабораторного практикуму, до 25 балів за виконання й захист завдань самостійної роботи і до 30 балів під час модульного контролю.

Оцінка за кожну лабораторну та самостійну роботу включає такі складові:

- виконання (наскільки повно й правильно виконані завдання роботи);
- захист (наскільки вільно студент володіє матеріалом і здатен представити результати виконаної роботи);
- звіт (наскільки грамотно, якісно й повно задокументовано у звіті результати виконаної роботи);
- дотримання графіка виконання (студент одержить максимальні бали за цією складовою за умови вчасного виконання роботи; у випадку відхилень від графіка кількість балів може знижуватися).