

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ЗАТВЕРДЖУЮ

Декан факультету інформаційних технологій

Тетяна ГОВОРУЩЕНКО

Підпис Ім'я, ПРІЗВИЩЕ

2025 р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Програмування веб-сервісів

Назва дисципліни

Призначення Робочої програми

Рівень вищої освіти

Мова навчання

Обсяг дисципліни, кредитів ЄКТС

Статус дисципліни

Факультет

Кафедра

Для освітніх програм різних спеціальностей

Перший (бакалаврський)

Українська

8

Вибіркова

Інформаційних технологій

Комп'ютерної інженерії та інформаційних систем

Форма здобуття освіти	Обсяг дисципліни		Кількість годин						Форма семестрового контролю
			Аудиторні заняття						Самостійна робота (в т.ч. ІРС)
	Кредити ЄКТС	Години	Разом	Лекції	Лабораторні роботи	Практичні заняття	Семінарські заняття		
Д	8	240	98	32	32	34		142	+

Робоча програма складена на основі освітніх програм підготовки бакалавра та стандарту вищої освіти спеціальності.

Робоча програма складена

Підпис

д-р. філософії Олександр МЕЛЬНИЧЕНКО

Науковий ступінь, вчене звання, Ім'я, ПРІЗВИЩЕ

Схвалена на засіданні кафедри комп'ютерної інженерії та інформаційних систем

Протокол від 29.08 2025 № 1.

Зав. кафедри комп'ютерної інженерії та інформаційних систем

Назва

Ольга ПАВЛОВА

Ім'я, ПРІЗВИЩЕ

Хмельницький 2025

3. Пояснювальна записка

Дисципліна «Програмування веб-сервісів» належить до циклу вибіркових дисциплін та характеризується широким міждисциплінарним підходом. Курс спрямований на поглиблення фахових умінь з розроблення програмного забезпечення, зокрема із застосуванням сучасних вебтехнологій та фреймворків. Опанування курсу формує міцну основу для подальшої професійної спеціалізації у сферах розроблення складних програмних систем та сервіс-орієнтованої архітектури.

Мета дисципліни. Оволодіння студентами основними принципами і методами об'єктно-орієнтованого програмування, зокрема, фреймворку розроблення програмного забезпечення ASP. NET CORE, який надалі можуть сприяти його успішному застосуванню в професійній діяльності.

Предмет дисципліни. Застосування фреймворку розроблення вебсервісів ASP. NET CORE, а також розроблення вебсервісів із застосуванням веборієнтованих технологій програмування.

Завдання дисципліни. Формування практичних навичок застосування принципів об'єктно-орієнтованого програмування, а також шаблонів проєктування під час розроблення програмного забезпечення та розроблення вебсервісів та застосунків із використанням фреймворку створення вебсервісів ASP. NET CORE.

Результати навчання. Після вивчення дисципліни студент повинен: *знати* об'єкт, предмет, задачі, проблематику дисципліни та її основні розділи; *володіти* математичними положеннями, що лежать в основі функціонування програмних систем, та базовими поняттями й визначеннями, які використовуються у галузі інформаційних технологій; *вміти* застосовувати основи комп'ютерної інженерії для розв'язування задач синтезу вебсервісів та програмного забезпечення на основі об'єктно-орієнтованого підходу із використанням фреймворку розроблення вебсервісів ASP. NET CORE; *застосовувати* творчі здібності до формування принципово нових ідей щодо розроблення вебсервісів та програмного забезпечення; *демонструвати* вміння розроблювати програмне забезпечення для вбудованих і розподілених застосунків, мобільних і гібридних систем; *оцінювати* результати роботи, захищати, пояснювати та аргументувати розроблений вебсервіс, здійснювати пошук інформації в різних джерелах для розв'язання задач побудови вебсервісів та програмного забезпечення.

4. Структура залікових кредитів дисципліни

Назва розділу (теми)	Кількість годин, відведених на:			
	лекції	лабораторні	практичні	СРС
Тема 1. Основи патернів проєктування. Введення в патерни проєктування.	4	4	4	16
Тема 2. Породжуючі патерни	4	4	4	16
Тема 3. Патерни поведінки	8	8	8	32
Тема 4. Структурні патерни	8	8	8	32
Тема 5. Принципи SOLID	6	6	6	32
Тема 6. Додаткові патерни. Fluent Builder	2	2	2	14
Разом за семестр:	32	32	34	142

5. Програма навчальної дисципліни

5.1 Зміст лекційного курсу

Номер лекції	Перелік тем лекцій, їхні анотації	Кількість годин
	Тема 1. Основи патернів проєктування. Введення в патерни проєктування.	4
1	Введення в патерни проєктування. Класифікація патернів. Вибір патернів. Літ.: [7] с. 65–88; [9] с. 820–846; [12] с. 600–632.	2
2	Відношення між класами і об'єктами. Інтерфейси або абстрактні класи. Літ.: [7] с. 41–58; [8] р. 35–56; [9] р. 180–212; [12] с. 212–246.	2
	Тема 2. Породжуючі патерни	4
3	Фабричний метод (Factory Method). Абстрактна фабрика (Abstract Factory). Літ.: [7] с. 89–98; [9] р. 845–870; [12] с. 632–660.	2
4	Сінглтон і багатопоточність. Прототип (Prototype). Будівельник (Builder). Літ.: [7] с. 99–108; [8] р. 120–132; [9] р. 870–900; [12] с. 660–692.	2
	Тема 3. Патерни поведінки	8
5	Стратегія (Strategy). Спостерігач (Observer). Команда (Command). Літ.: [7] с. 109–120; [9] р. 900–930; [12] с. 692–724.	2
6	Макрокоманди. Шаблонний метод (Template Method). Літ.: [7] с. 121–126; [9] р. 930–942; [12] с. 724–736.	2
7	Ітератор (Iterator). Стан (State). Ланцюжок обов'язків (Chain of Responsibility). Літ.: [7] с. 55–64; [9] р. 942–972; [12] с. 736–760.	2
8	Інтерпретатор (Interpreter). Посередник (Mediator). Зберігач (Memento). Відвідувач (Visitor). Літ.: [9] р. 972–1010; [12] с. 760–792.	2
	Тема 4. Структурні патерни	8
9	Декоратор (Decorator). Адаптер (Adapter). Літ.: [7] с. 60–72; [9] р. 780–804; [12] с. 580–612.	2
10	Фасад (Facade). Компоновщик (Composite). Літ.: [9] р. 804–826; [12] с. 612–640.	2
11	Замісник (Proxy). Міст (Bridge). Літ.: [9] р. 826–845; [12] с. 640–660.	2
12	Патерн Пристосуванець (Flyweight). Літ.: [9] р. 1010–1022; [12] с. 736–748.	2
	Тема 5. Принципи SOLID	6
13	Single Responsibility Principle (Принцип єдиних обов'язків). Open/Closed Principle (Принцип відкритості / закритості). Літ.: [7] с. 33–48; [9] р. 420–446; [12] с. 520–544.	2
14	Liskov Substitution Principle (Принцип підстановки Лисків). Літ.: [9] р. 446–458; [12] с. 544–556.	2
15	Interface Segregation Principle (Принцип поділу інтерфейсів). Літ.: [9] р. 458–468; [12] с. 556–568.	2
	Тема 6. Додаткові патерни. Fluent Builder	2
16	Патерн проєктування Fluent Builder. Літ.: [7] с. 90–96; [9] р. 888–900; [12] с. 600–616.	2
	Разом за семестр:	32

5.2 Зміст практичних занять

№ п/п	Тема практичного заняття	Кількість годин
1	Введення в ASP.NET Core. Структура проєкту та життєвий цикл запиту. Літ.: [7] с. 5–22; [9] р. 212–238; [10] с. 120–162; [12] с. 500–540.	2
2	Основи ASP.NET Core: Middleware, конфігурація та запуск застосунку. Літ.: [7] с. 23–32; [9] р. 238–260; [10] с. 162–190; [12] с. 540–564.	2
3	Сервіси і Dependency Injection: реєстрація сервісів та їх життєві цикли. Літ.: [7] с. 49–58; [9] р. 468–482; [12] с. 568–584.	2
4	Практичне застосування DI: впровадження залежностей в контролери та сервіси. Літ.: [7] с. 59–64; [9] р. 482–498; [12] с. 584–600.	2
5	Управління станом: робота з Cookies та сесіями. Літ.: [9] р. 260–274; [10] с. 300–332; [11] с. 60–84; [12] с. 356–380.	2
6	Основи маршрутизації та логування в ASP.NET Core. Літ.: [9] р. 274–288; [10] с. 190–210; [11] с. 85–102; [12] с. 540–556.	2
7	Робота з контролерами в ASP.NET Core MVC: методи дій та передача даних. Літ.: [9] р. 288–310; [10] с. 210–252; [11] с. 103–128; [12] с. 556–572.	2
8	Представлення (Views) та Моделі (Models): Razor syntax та ViewModel. Літ.: [9] р. 310–332; [10] с. 252–280; [11] с. 129–160; [12] с. 572–596.	2
9	Реалізація патерну «Декоратор» для розширення функціональності сервісів. Літ.: [7] с. 60–66; [9] р. 780–792; [12] с. 580–598.	2
10	Застосування патерну «Адаптер» для інтеграції зі сторонніми сервісами. Літ.: [7] с. 66–72; [9] р. 792–804; [12] с. 598–612.	2
11	Патерн «Спостерігач» (Observer) для реалізації системи сповіщень. Літ.: [7] с. 112–118; [9] р. 906–918; [12] с. 700–712.	2
12	Патерн «Стратегія» (Strategy) для реалізації гнучких бізнес-правил. Літ.: [7] с. 109–112; [9] р. 900–906; [12] с. 692–700.	2
13	Використання патерну "Фабричний метод" (Factory Method) для створення об'єктів. Літ.: [7] с. 89–94; [9] р. 845–858; [12] с. 632–644.	2
14	Робота з Entity Framework Core: налаштування контексту даних та міграції. Літ.: [9] р. 600–642; [10] с. 332–372; [12] с. 596–612.	2
15	Виконання CRUD-операцій за допомогою Entity Framework Core в MVC. Літ.: [9] р. 642–690; [10] с. 372–408; [12] с. 612–632.	2
16	Створення RESTful API за допомогою ASP.NET Core Web API. Літ.: [9] р. 332–356; [10] с. 280–300; [11] с. 161–180; [12] с. 632–652.	2
Разом за семестр:		34

5.3 Зміст лабораторних занять

№ п/п	Тема лабораторного заняття	Кількість годин
1	Введення в ASP.NET Core. Основи ASP.NET Core. Літ.: [7] с. 5–22; [9] р. 212–260; [10] с. 120–170; [12] с. 500–540.	4
2	Сервіси і Dependency Injection. Конфігурація. Літ.: [7] с. 49–64; [9] р. 468–498; [12] с. 568–596.	4
3	Стан застосунків. Куки. Сесії. Логування. Маршрутизація. Літ.: [9] р. 260–288; [10] с. 170–210; [11] с. 60–102; [12] с. 540–572.	4
4	ASP.NET Core MVC. Контролери. Представлення. Моделі. Літ.: [9] р. 288–332; [10] с. 210–280; [11] с. 103–160; [12] с. 556–596.	4
5	Маршрутизація в ASP.NET Core MVC. Літ.: [9] р. 274–310; [10] с. 190–240; [11] с. 140–160; [12] с. 540–572.	4
6	Помічники HTML. Помічники тегів. View Component. Літ.: [9] р. 310–320; [10] с. 240–268; [11] с. 120–140; [12] с. 572–588.	4
7	Метадані та валідація моделі. Літ.: [9] р. 320–332; [10] с. 268–300; [11] с. 161–176; [12] с. 588–604.	4
8	Робота з даними в Entity Framework в MVC. Літ.: [9] р. 600–690; [10] с. 332–408; [12] с. 604–632.	4
Разом за семестр:		32

5.4 Зміст самостійної роботи здобувача вищої освіти

Самостійна робота студентів полягає у систематичному опрацюванні програмного матеріалу з відповідних джерел інформації, підготовці до практичних та лабораторних занять, виконанні індивідуальних завдань (ІЗ), підготовці до підсумкової контрольної роботи (ПКР). До послуг студентів сторінка кафедри у Модульному середовищі для навчання, де розміщені Робоча програма дисципліни та необхідні матеріали з її навчально-методичного забезпечення та контролю результатів навчання.

Номер тижня	Вид самостійної роботи	Кількість годин
1–2	Опрацювання лекційного матеріалу. Підготовка до виконання та захисту ЛР №1 та підготовка до ПЗ №1–2.	14
3–4	Опрацювання лекційного матеріалу. Підготовка до виконання та захисту ЛР №2 та підготовка до ПЗ №3–4. Виконання ІЗ №1 “Розроблення вебзастосунку «Персональний блог» з системою автентифікації та коментарів”.	20
5–6	Опрацювання лекційного матеріалу. Підготовка до виконання та захисту ЛР №3 та підготовка до ПЗ №5–6.	12
7–8	Опрацювання лекційного матеріалу. Підготовка до виконання та захисту ЛР №4 та підготовка до ПЗ №7–8. Виконання ІЗ №2 “Реалізація системи оброблення замовлень з використанням патернів "Стратегія" та "Спостерігач””.	20
9–10	Опрацювання лекційного матеріалу. Підготовка до виконання та захисту ЛР №5 та підготовка до ПЗ №9–10.	12
11–12	Опрацювання лекційного матеріалу. Підготовка до виконання та захисту ЛР №6 та підготовка до ПЗ №11–12. Виконання ІЗ №3 “Створення RESTful API для управління завданнями (Task Manager) з документацією Swagger/OpenAPI”.	20
13–14	Опрацювання лекційного матеріалу. Підготовка до виконання та захисту ЛР №7 та підготовка до ПЗ №13–14. Підготовка до ПКР.	20
15–16	Опрацювання лекційного матеріалу. Підготовка до виконання та захисту ЛР №8 та підготовка до ПЗ №15–16. Виконання ІЗ №4 “Розроблення та впровадження власного Middleware для моніторингу запитів в ASP.NET Core”. Захист ІЗ.	24
Разом за семестр:		142

Примітки: ЛР – лабораторна робота, ПЗ – практичне заняття, ІЗ – індивідуальне завдання.

На самостійне опрацювання здобувачів виносяться ІЗ з відповідних тем, що сформовані в Модульному середовищі для навчання MOODLE. Керівництво самостійною роботою та контроль за виконанням ІЗ здійснюється викладачем, згідно з розкладом консультацій у позаурочний час.

Вимоги до виконання ІЗ викладені в Модульному середовищі для навчання на сторінці навчальної дисципліни.

6. Технології та методи навчання

Процес навчання з дисципліни ґрунтується на використанні традиційних та сучасних технологій та методів навчання, зокрема: *лекції* (з використанням проблемного й інтерактивного навчання, аналізу архітектурних діаграм, демонстрацій коду в реальному часі (live-coding) та інформаційно-комунікаційних технологій); *практичні заняття* (з використанням інструктування, аналізу реальних кейсів проектування вебсервісів, рефакторингу коду для застосування патернів проектування та принципів SOLID, елементів групової дискусії); *лабораторні заняття* (з фокусом на практичній реалізації вебсервісів та вебзастосунків з використанням фреймворку ASP.NET Core); *самостійна робота* (опрацювання теоретичного матеріалу, підготовка до виконання лабораторних робіт, підготовка до підсумкового контролю, виконання та захист ІЗ), з використанням інформаційно-комп’ютерних технологій та технологій дистанційного навчання.

7. Методи контролю

Поточний контроль здійснюється під час аудиторних практичних та лабораторних занять, а також у дні проведення контрольних заходів, встановлених робочою програмою і графіком освітнього процесу, зокрема, з використанням Модульного середовища для навчання. Водночас використовуються такі методи поточного контролю:

- оцінювання результатів роботи на практичних заняттях (опитування теоретичного матеріалу, розв’язування задач, участь у обговоренні ситуацій);
- оцінювання результатів захисту лабораторних робіт;
- захист та оцінювання результатів виконання ІЗ.

Підсумкова семестрова оцінка виставляється за результатами поточного контролю. Здобувач вищої освіти, який набрав з будь-якого виду навчальної роботи, суму балів нижчу за 60 відсотків від максимального балу, вважається таким,

який *має* академічну заборгованість. Ліквідація академічної заборгованості із семестрового контролю здійснюється у період екзаменаційної сесії або за графіком, встановленим деканатом, відповідно до «Положення про контроль і оцінювання результатів навчання здобувачів вищої освіти у ХНУ».

8. Політика дисципліни

Політика навчальної дисципліни загалом визначається системою вимог до здобувача вищої освіти, що передбачені чинними положеннями Університету про організацію і навчально-методичне забезпечення освітнього процесу. Зокрема, проходження інструктажу з техніки безпеки; відвідування занять з дисципліни є обов'язковим. За об'єктивних причин (підтверджених документально) теоретичне навчання за погодженням із лектором може відбуватись в онлайн режимі. Успішне опанування дисципліни і формування фахових компетентностей і програмних результатів навчання передбачає необхідність підготовки до практичних та лабораторних занять (вивчення теоретичного матеріалу з теми, попередню підготовку звіту роботи, підготовку до усного опитування для допуску до заняття), активно працювати на занятті, брати участь у дискусіях щодо прийнятих рішень при виконанні здобувачами лабораторних робіт та виконати ІЗ.

Здобувачі вищої освіти мають дотримуватися встановлених термінів виконання всіх видів навчальної роботи, відповідно до робочої програми навчальної дисципліни. Термін захисту лабораторної роботи вважається своєчасним, якщо студент захистив її на наступному після виконання роботи занятті. Пропущене практичне і лабораторне заняття студент зобов'язаний відпрацювати у встановлений викладачем термін, але не пізніше, ніж за два тижні до кінця теоретичних занять у семестрі.

Засвоєння студентом теоретичного матеріалу з дисципліни оцінюється за результатами опитування під час практичних занять та написання ПКР.

Здобувач вищої освіти, виконуючи самостійну роботу або індивідуальну роботу з дисципліни, має дотримуватися політики доброчесності (заборонені списування, плагіат (зокрема, із використанням мобільних девайсів)). У разі виявлення порушення політики академічної доброчесності в будь-яких видах навчальної роботи здобувач вищої освіти отримує незадовільну оцінку і має повторно виконати завдання з відповідної теми (виду роботи), що передбачені робочою програмою. Будь-які форми порушення академічної доброчесності **не допускаються**.

У межах вивчення навчальної дисципліни здобувачам вищої освіти передбачено визнання і зарахування результатів навчання, набутих через неформальну освіту, що розміщені на доступних платформах, які сприяють формуванню компетентностей і поглибленню результатів навчання, визначених Робочою програмою дисципліни, або забезпечують вивчення відповідної теми та/або виду робіт з програми навчальної дисципліни (детальніше у «Положенні про порядок визнання та зарахування результатів навчання здобувачів вищої освіти у ХНУ»). Так, окремі результати вивчення навчальної дисципліни можуть бути зараховані у випадку отримання студентом результатів навчання у неформальній освіті, що підтверджені відповідним документом (сертифікат, свідоцтво, освітня програма тощо):

– Як результат виконання ЛР № 1–2 зараховується відкритий онлайн-курс:

<https://learn.microsoft.com/en-us/training/paths/aspnet-core-fundamentals/>

– Як результат виконання ЛР № 3–4 зараховується онлайн-курс: <https://learn.microsoft.com/en-us/training/paths/aspnet-core-web-app/>

– Як результат виконання ЛР № 5–6 зараховується онлайн-курс: <https://www.freecodecamp.org/news/learn-asp-net-core-mvc-net-6/>

– Як результат виконання ЛР № 7–8 зараховується онлайн-курс: <https://www.freecodecamp.org/news/how-to-build-an-aspnet-core-mvc-web-app/>

9. Оцінювання результатів навчання студентів у семестрі

Оцінювання академічних досягнень здобувача вищої освіти здійснюється, відповідно до «Положення про контроль і Оцінювання академічних досягнень здобувача вищої освіти здійснюється, відповідно до «Положення про контроль і оцінювання результатів навчання здобувачів вищої освіти в ХНУ». При поточному оцінюванні виконаної здобувачем роботи з кожної структурної одиниці та отриманих ним результатів викладач виставляє йому певну кількість балів із встановлених Робочою програмою для цього виду роботи. Водночас кожна структурна одиниця навчальної роботи може бути зарахована, якщо здобувач набрав не менше 60 відсотків (мінімальний рівень для позитивної оцінки) від максимально можливої суми балів, призначеної структурній одиниці.

При оцінюванні результатів навчання здобувачів вищої освіти з будь-якого виду навчальної роботи (структурної одиниці) рекомендується використовувати наведені нижче узагальнені критерії:

Таблиця – Критерії оцінювання навчальних досягнень здобувача вищої освіти

Оцінка та рівень досягнення здобувачем запланованих програмних результатів навчання та сформованих компетентностей	Узагальнений зміст критерія оцінювання
Відмінно (високий)	Здобувач вищої освіти глибоко і у повному обсязі опанував зміст навчального матеріалу, легко в ньому орієнтується і вміло використовує понятійний апарат; уміє пов'язувати теорію з практикою, вирішувати практичні завдання, впевнено висловлювати і обґрунтовувати свої судження. Відмінна оцінка передбачає логічний виклад відповіді мовою викладання (в усній або у письмовій формі), демонструє якісне оформлення роботи і володіння спеціальними приладами та інструментами, прикладними програмами. Здобувач не вагається при видозміні запитання, вміє робити детальні та узагальнюючі висновки, демонструє практичні навички з вирішення фахових завдань. При відповіді допустив дві–три несуттєві <i>помилки</i> .
Добре (середній)	Здобувач вищої освіти виявив повне засвоєння навчального матеріалу, володіє понятійним апаратом, орієнтується у вивченому матеріалі; свідомо використовує теоретичні знання для вирішення практичних задач; виклад відповіді грамотний, але у змісті і формі відповіді можуть мати місце окремі неточності, нечіткі формулювання правил, закономірностей тощо. Відповідь здобувача вищої освіти будується на основі самостійного мислення. Здобувач вищої освіти у відповіді допустив дві–три <i>несуттєві помилки</i> .
Задовільно (достатній)	Здобувач вищої освіти виявив знання основного програмного матеріалу в обсязі, необхідному для подальшого навчання та практичної діяльності за професією, справляється з виконанням практичних завдань, передбачених програмою. Як правило, відповідь здобувача вищої освіти будується на рівні репродуктивного мислення, здобувач вищої освіти має слабкі знання структури навчальної дисципліни, допускає неточності і <i>суттєві помилки</i> у відповіді, вагається при відповіді на видозмінене запитання. Разом з тим, набув навичок, необхідних для виконання нескладних практичних завдань, які відповідають мінімальним критеріям оцінювання і володіє знаннями, що дозволяють йому під керівництвом викладача усунути неточності у відповіді.
Незадовільно (недостатній)	Здобувач вищої освіти виявив розрізнені, безсистемні знання, не вміє виділяти головне і другорядне, допускається помилок у визначенні понять, перекручує їхній зміст, хаотично і невпевнено викладає матеріал, не може використовувати знання при вирішенні практичних завдань. Як правило, оцінка «незадовільно» виставляється здобувачеві вищої освіти, який не може продовжити навчання без додаткової роботи з вивчення навчальної дисципліни.

Структурування дисципліни за видами навчальної роботи і оцінювання результатів навчання студентів денної форми здобуття освіти у семестрі

Аудиторна робота											Контрольні заходи	Самостійні на робота	Семестровий контроль
Лабораторні роботи №:								Практичні заняття			Підсумкова контрольна робота (ПКР)	Індивідуальне завдання (ІЗ)	Залік
1*	2	3	4	5	6	7	8	1	2	3			
Кількість балів за кожний вид навчальної роботи (мінімум-максимум)													
3-5	3-5	3-5	3-5	3-5	3-5	3-5	3-5	6-10	6-10	6-10	6-10	12-20	За рейтингом
24-40								18-30			6-10	12-20	60-100

Примітки: *за набрану з будь-якого виду навчальної роботи з дисципліни кількість балів, нижче встановленого мінімуму, здобувач отримує незадовільну оцінку і має її перездати у встановлений викладачем (деканом) термін. Інституційна оцінка встановлюється, відповідно до таблиці «Співвідношення інституційної шкали оцінювання і шкали оцінювання ЄКТС».

Оцінювання на практичних заняттях

Оцінка, яка виставляється за практичне заняття, складається з таких елементів: усне опитування студентів на знання теоретичного матеріалу з теми; вільне володіння студентом термінологією дисципліни й уміння професійно обґрунтувати прийняті рішення при розв'язуванні задач; результати самостійних робіт.

При оцінюванні практичного заняття викладач керується узагальненими критеріями, наведеними у таблиці «**Критерії оцінювання навчальних досягнень здобувача вищої освіти**» (мінімальний позитивний бал – 6 балів, максимальний – 10 балів).

Оцінювання результатів захисту лабораторної роботи

Лабораторна робота, що виконана й оформлена студентом, відповідно до вимог, які визначені в Модульному середовищі для навчання MOODLE, комплексно оцінюється викладачем під час її захисту з урахуванням таких критеріїв: самостійність та коректність програмної реалізації архітектури та функціоналу вебсервісу; повнота відповіді та знання теоретичних основ (наприклад, принципів REST, патернів проектування, роботи фреймворку); якість програмного коду, дотримання принципів SOLID та обґрунтованість обраних архітектурних рішень.

Результат виконання і захисту здобувачем вищої освіти кожної лабораторної роботи оцінюється, відповідно до таблиці «**Критерії оцінювання навчальних досягнень здобувача вищої освіти**» (мінімальний позитивний бал – 3 бали, максимальний – 5 балів).

У випадку виявлення здобувачем рівня знань, нижчого ніж 60 відсотків від максимального балу, встановленого Робочою програмою для кожної структурної одиниці, лабораторна робота йому не зараховується і для її захисту він має детальніше опрацювати матеріал з теми роботи, методику її виконання, виправити грубі помилки та повторно вийти на її захист у призначений для цього викладачем час.

Оцінювання підсумкової контрольної роботи

ПКР передбачає виконання трьох завдань (два теоретичних питання та одне практичне завдання, що передбачає вирішення задачі з розроблення або конфігурації вебсервісу). Під час оцінювання ПКР враховуються: повнота та коректність відповіді, якість вирішення практичного завдання. Кожне теоретичне завдання оцінюється у 2 бали, практичне завдання – у 6 балів; загальна сума балів на позитивну оцінку становить від 6 до 10 балів.

Кількість правильних відповідей	1	2	3
Відсоток правильних відповідей	0-59	60	100
Кількість отриманих балів	3	6	10

У випадку отримання студентом негативної оцінки, ПКР варто перездати до завершення залікового тижня.

Оцінювання результатів виконання індивідуального завдання

ІЗ виконується студентом та оформлюється, відповідно до вимог, які визначені в Модульному середовищі для навчання MOODLE, а також комплексно оцінюється викладачем з урахуванням таких критеріїв: самостійність виконання; правильність розв'язання поставлених задач; обґрунтованість вибору методів розв'язання; повнота пояснень та аргументованість відповідей; якість оформлення звіту та програмного коду.

Результат виконання здобувачем вищої освіти кожного ІЗ оцінюється, відповідно до таблиці «**Критерії оцінювання навчальних досягнень здобувача вищої освіти**» (мінімальний позитивний бал – 12 балів, максимальний – 20 балів). За підсумками захисту присвоюється відповідна сума балів.

У разі, якщо здобувач вищої освіти виявив рівень знань і виконання ІЗ, що нижчий ніж 60 відсотків від максимальної кількості балів, встановленої Робочою програмою для цієї структурної одиниці, завдання не зараховується. У такому випадку студент має повторно опрацювати зміст завдання, усунути помилки та здати на перевірку доопрацьоване ІЗ у терміни, погоджені з викладачем.

Підсумкова семестрова оцінка

Підсумкова семестрова оцінка за інституційною шкалою і шкалою ЄКТС визначається в автоматизованому режимі після внесення викладачем результатів оцінювання у балах з усіх видів навчальної роботи до електронного журналу. Співвідношення інституційної шкали оцінювання і шкали оцінювання ЄКТС наведені нижче у таблиці «**Співвідношення інституційної шкали оцінювання і шкали оцінювання ЄКТС**».

Таблиця – Співвідношення інституційної шкали оцінювання і шкали оцінювання ЄКТС

Оцінка ЄКТС	Рейтингова шкала балів	Інституційна оцінка (рівень досягнення здобувачем вищої освіти запланованих результатів навчання з навчальної дисципліни)	
		Залік	Іспит/диференційований залік
A	90-100	Зараховано	Відмінно/Excellent – високий рівень досягнення запланованих результатів навчання з навчальної дисципліни, що свідчить про безумовну готовність здобувача до подальшого навчання та/або професійної діяльності за фахом
B	83-89		Добре/Good – середній (максимально достатній) рівень досягнення запланованих результатів навчання з навчальної дисципліни та готовності до подальшого навчання та/або професійної діяльності за фахом
C	73-82		

D	66-72	Незараховано	Задовільно/Satisfactory – Наявні мінімально достатні для подальшого навчання та/або професійної діяльності за фахом результати навчання з навчальної дисципліни
E	60-65		
FX	40-59		Незадовільно/Fail – Низка запланованих результатів навчання з навчальної дисципліни відсутня. Рівень набутих результатів навчання є недостатнім для подальшого навчання та/або професійної діяльності за фахом
F	0-39		Незадовільно/Fail – Результати навчання відсутні

Семестровий залік виставляється на останньому занятті за умови, якщо загальна сума балів, яку накопичив здобувач з дисципліни за результатами поточного контролю, перебуває у межах від 60 до 100 балів. Водночас за інституційною шкалою ставиться оцінка «зараховано», а за шкалою ЄКТС – буквене позначення оцінки, що відповідає набраній студентом сумі балів, відповідно до таблиці «Співвідношення». Присутність здобувача у цьому випадку не є обов'язковою.

10. Питання для самоконтролю результатів навчання

1. Життєвий цикл HTTP-запиту в ASP.NET Core.
2. Роль та структура файлу Program.cs у застосунках на базі мінімальних API.
3. Концепція конвеєра оброблення запитів (request pipeline) та його конфігурація.
4. Призначення та механізм роботи проміжного програмного забезпечення (Middleware).
5. Створення та реєстрація власного Middleware у конвеєрі.
6. Принцип інверсії залежностей (Dependency Injection) як основа архітектури .NET.
7. Життєві цикли сервісів: Singleton, Scoped, Transient та їх відмінності.
8. Механізми конфігурації застосунку (appsettings.json, змінні середовища, секрети користувача).
9. Структура та призначення контролерів у патерні MVC та в Web API.
10. Атрибутна маршрутизація для декларативного визначення ендпоінтів.
11. Конвенційна маршрутизація та її налаштування для традиційних MVC-застосунків.
12. Використання обмежень маршрутів (route constraints) для валідації параметрів URL.
13. Механізм прив'язки моделей (Model Binding) до параметрів методів дій.
14. Джерела прив'язки даних ([FromRoute], [FromBody], [FromQuery], [FromHeader]).
15. Ієрархія та призначення типів результатів дій (ActionResult, ActionResult<T>).
16. Фундаментальні принципи RESTful API.
17. Правильне використання HTTP-методів (GET, POST, PUT, DELETE) у REST-архітектурі.
18. Стандартні HTTP-коди стану та їх семантичне застосування у відповідях API.
19. Роль класу DbContext як шлюзу до бази даних в Entity Framework Core.
20. Підхід до розроблення баз даних Code-First та його переваги.
21. Процес створення та застосування міграцій для управління схемою БД.
22. Реалізація базових CRUD-операцій (Create, Read, Update, Delete) за допомогою EF Core.
23. Налаштування зв'язків між сутностями: "один-до-багатьох" та "багато-до-багатьох".
24. Використання асинхронних методів (async/await) для роботи з базою даних.
25. Формування запитів до бази даних за допомогою LINQ (Language Integrated Query).
26. Призначення та відмінність між доменною моделлю та DTO (Data Transfer Object).
27. Основи синтаксису Razor для створення динамічних HTML-представлень.
28. Використання строго типізованих моделей (@model) у представленнях.
29. Призначення та використання майстер-сторінок (_Layout.cshtml) для уніфікації інтерфейсу.
30. Застосування часткових представлень (Partial Views) для повторного використання розмітки.
31. Роль та створення компонентів представлення (View Components) для інкапсуляції UI-логіки.
32. Використання тег-хелперів (Tag Helpers) для спрощення генерації HTML-форм та елементів.
33. Валідація моделей на стороні сервера за допомогою анотацій даних.
34. Перевірка стану валідації в контролері за допомогою об'єкта ModelState.
35. Реалізація власної (кастомної) логіки валідації шляхом створення атрибутів.
36. Принцип єдиної відповідальності (SRP) в контексті сервісів та контролерів.
37. Принцип відкритості/закритості (OCP) при розширенні функціоналу системи.
38. Принцип підстановки Лісков (LSP) у побудові ієрархій класів.
39. Принцип розділення інтерфейсу (ISP) для створення гнучких залежностей.
40. Принцип інверсії залежностей (DIP) та його зв'язок з DI-контейнером.
41. Патерн "Стратегія" для реалізації взаємозамінних алгоритмів.
42. Патерн "Декоратор" для динамічного додавання нової функціональності до об'єктів.
43. Патерн "Адаптер" для інтеграції несумісних інтерфейсів сторонніх бібліотек.
44. Патерн "Фабричний метод" для делегування створення об'єктів підкласам.
45. Патерн "Спостерігач" для організації слабкозв'язаної системи сповіщень.
46. Концепція та типи фільтрів (Action, Exception, Authorization, Result) в ASP.NET Core.

47. Глобальна оброблення винятків за допомогою фільтрів або спеціалізованого Middleware.
48. Основи механізмів автентифікації та авторизації для захисту ендпоінтів.
49. Інтеграція Swagger/OpenAPI для автоматичного документування та тестування API.
50. Управління станом клієнта за допомогою Cookies та Session.

11. Навчально-методичне забезпечення

Освітній процес з дисципліни «Програмування веб-сервісів» повністю і в достатній кількості забезпечений необхідною навчально-методичною літературою, перелік якої наведено у розділі 13.

12. Матеріально-технічне та програмне забезпечення дисципліни

Для проведення лекційних, практичних та лабораторних занять використовуються спеціалізовані аудиторії та комп'ютерні класи кафедри, оснащені сучасною комп'ютерною технікою та мультимедійним обладнанням. Програмне забезпечення включає операційні системи (Windows, Linux), інтегровані середовища розроблення (наприклад, Visual Studio 2022 Community Edition, JetBrains Rider), платформу .NET SDK, систему контролю версій Git, а також інструменти для тестування API (наприклад, Postman). Усе програмне забезпечення, що використовується в межах вивчення дисципліни, є ліцензійним або вільно розповсюджуваним.

13. Рекомендована література

1. Criticality assessment of wind turbine defects via multispectral UAV fusion and fuzzy logic / P. Radiuk, B. Rusyn, O. Melnychenko, T. Perzynski, A. Sachenko, S. Svystun, O. Savenko. *Energies*. 2025. Vol. 18, no. 17. P. 4523. URL: <https://doi.org/10.3390/en18174523> (date of access: 26.08.2025).
2. Enhanced fire hazard detection in solar power plants: an integrated UAV, AI, and SCADA-based approach / A. Lysyi, A. Sachenko, P. Radiuk, M. Lysyi, O. Melnychenko, O. Ishchuk, O. Savenko. *Radioelectronic and computer systems*. 2025. Vol. 2025, no. 2. P. 99–117. URL: <https://doi.org/10.32620/reks.2025.2.06> (date of access: 28.07.2025).
3. Determining the criticality assessment of defects on wind turbine components detected by UAV sensors / S. Svystun, O. Melnychenko, P. Radiuk, A. Lysyi, A. Sachenko. *Proceedings of the 6th International Workshop on Intelligent Information Technologies & Systems of Information Security (IntelITSIS 2025)* : CEUR-Workshop Proceedings, Khmelnytskyi, Ukraine, 4 April 2025 / ed. by T. Hovorushchenko et al. Aachen, 2025. P. 351–362. URL: <https://ceur-ws.org/Vol-3963/paper28.pdf> (date of access: 28.07.2025).
4. DyTAM: Accelerating wind turbine inspections with dynamic UAV trajectory adaptation / S. Svystun, L. Scislo, M. Pawlik, O. Melnychenko, P. Radiuk, O. Savenko, A. Sachenko. *Energies*. 2025. Vol. 18, no. 7. P. 1823. URL: <https://doi.org/10.3390/en18071823> (date of access: 28.07.2025).
5. Dynamic trajectory adaptation for efficient UAV inspections of wind energy units / S. Svystun, O. Melnychenko, P. Radiuk, O. Savenko, A. Sachenko, A. Lysyi. *Proceedings of the 2024 14th International Conference on Dependable Systems, Services and Technologies (DESSERT)* : Proceedings, Athens, Greece, 11–13 October 2024 / ed. by V. Kharchenko et al. New York, NY, USA, 2025. P. 1–7. URL: <https://doi.org/10.1109/DESSERT65323.2024.11122243> (date of access: 21.08.2025).
6. Intelligent integrated system for fruit detection using multi-UAV imaging and deep learning / O. Melnychenko, L. Scislo, O. Savenko, A. Sachenko, P. Radiuk. *Sensors*. 2024. Vol. 24, no. 6. P. 1913. URL: <https://doi.org/10.3390/s24061913> (date of access: 22.05.2025).
7. Каплун В.А. Основи web-програмування. Теорія і практика : електронний навчальний посібник комбінованого (локального та мережного) використання [Електронний ресурс] / Каплун В. А., Ціхоцький М. С., Лукічов В. В. Вінниця : ВНТУ, 2023. 128 с.
8. Albahari J. C# 12.0 Pocket Reference: Instant Help for C# 12.0 Programmers 1st Edition. O'Reilly Media, 2023. 284 p.
9. Albahari J. C# 10 in a Nutshell. The Definitive Reference. O'Reilly Media, 2022. 1058 p.
10. Двірничук К. В., Вацек Д. О. Веб-програмування та веб-дизайн : навч. посіб. Чернівці : Чернівець. нац. ун-т ім. Ю. Федьковича, 2022. 472 с.
11. Босько В.В., Константинова Л.В., Марченко К.М., Улічев О.С. Web-програмування. Частина 1 (frontend) : навч. посіб. Кропивницький: ЦНТУ, 2022. 208 с.
12. Грін Дж., Стеллман Е.. Head First C#: Посібник для навчання з програмування в реальному світі за допомогою C# та .NET Core 4th Edition. O'Reilly Media, 2021. 800 с.

14. Інформаційні ресурси

1. Модульне середовище для навчання. URL: <https://msn.khmnu.edu.ua/course/view.php?id=7544>
2. Електронна бібліотека ХНУ. URL: <http://library.khmnu.edu.ua/>
3. Інституційний репозитарій ХНУ. URL : <https://elar.khmnu.edu.ua/home>

ПРОГРАМУВАННЯ ВЕБ-СЕРВІСІВ

Тип дисципліни	Вибіркова
Рівень вищої освіти	Перший (бакалаврський)
Мова викладання	Українська
Семестр	—
Кількість призначених кредитів ЄКТС	8,0
Форми здобуття освіти, для яких викладається дисципліна	Очна (денна)

Результати навчання. Після вивчення дисципліни студент повинен: *знати* об'єкт, предмет, задачі, проблематику дисципліни та її основні розділи; *володіти* математичними положеннями, що лежать в основі функціонування програмних систем, та базовими поняттями й визначеннями, які використовуються у галузі інформаційних технологій; *вміти* застосовувати основи комп'ютерної інженерії для розв'язування задач синтезу вебсервісів та програмного забезпечення на основі об'єктно-орієнтованого підходу із використанням фреймворку розроблення вебсервісів ASP.NET CORE; *застосовувати* творчі здібності до формування принципово нових ідей щодо розроблення вебсервісів та програмного забезпечення; *демонструвати* уміння розроблювати програмне забезпечення для вбудованих і розподілених застосунків, мобільних і гібридних систем; *оцінювати* результати роботи, захищати, пояснювати та аргументувати розроблений вебсервіс, здійснювати пошук інформації в різних джерелах для розв'язання задач побудови вебсервісів та програмного забезпечення.

Зміст навчальної дисципліни. Вивчення фундаментальних принципів та практичних підходів до розроблення сучасних вебсервісів на платформі .NET. Основні теми включають: архітектуру ASP.NET Core (конвеєр оброблення запитів, Middleware, Dependency Injection), розроблення вебзастосунків за патерном MVC (контролери, моделі, представлення Razor), створення RESTful Web API (ендпоінти, HTTP-методи, коди стану, DTO). Розглядається робота з базами даних за допомогою Entity Framework Core (DbContext, міграції, LINQ-запити). Особлива увага приділяється ключовим механізмам, таким як маршрутизація, прив'язка моделей, валідація та фільтри. Курс охоплює застосування принципів SOLID та основних патернів проектування (породжуючих, структурних, поведінкових) для створення якісного та підтримуваного коду.

Запланована навчальна діяльність. Мінімальний обсяг навчальних занять в одному кредиті ЄКТС навчальної дисципліни для *першого* (бакалаврського) рівня вищої освіти за денною формою здобуття освіти становить 10 годин на 1 кредит ЄКТС.

Форми (методи) навчання: *лекції* (з використанням проблемного й інтерактивного навчання, аналізу архітектурних діаграм, демонстрацій коду в реальному часі (live-coding) та інформаційно-комунікаційних технологій); *практичні заняття* (з використанням інструктування, аналізу реальних кейсів проектування вебсервісів, рефакторингу коду для застосування патернів проектування та принципів SOLID, елементів групової дискусії); *лабораторні заняття* (з фокусом на практичній реалізації вебсервісів та вебзастосунків з використанням фреймворку ASP.NET Core); *самостійна робота* (опрацювання теоретичного матеріалу, підготовка до виконання лабораторних робіт, підготовка до підсумкового контролю, виконання та захист ІЗ), з використанням інформаційно-комп'ютерних технологій та технологій дистанційного навчання.

Форми оцінювання результатів навчання: оцінювання виконання практичних завдань; оцінювання виконання та захисту лабораторних робіт; оцінювання підсумкової контрольної роботи; оцінювання виконання та захисту індивідуального завдання.

Вид семестрового контролю: залік.

Навчальні ресурси:

1. Каплун В.А. Основи web-програмування. Теорія і практика : електронний навчальний посібник комбінованого (локального та мережного) використання [Електронний ресурс] / Каплун В. А., Ціхоцький М. С., Лукічов В. В. Вінниця : ВНТУ, 2023. 128 с.
2. Albahari J. C# 12.0 Pocket Reference: Instant Help for C# 12.0 Programmers 1st Edition. O'Reilly Media, 2023. 284 p.
3. Двірничук К. В., Вацек Д. О. Веб-програмування та веб-дизайн : навч. посіб. Чернівці : Чернівець. нац. ун-т ім. Ю. Федьковича, 2022. 472 с.
4. Модульне середовище для навчання. URL: <https://msn.khmnmu.edu.ua/course/view.php?id=7544>
5. Електронна бібліотека ХНУ. Доступ до ресурсу: <http://library.khmnmu.edu.ua/>

Викладачі: д-р. філософії, ст. викл. Мельниченко О.В.