



Сучасні технології розробки WEB-застосунків на платформі Microsoft.NET

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський)</i>
Галузь знань	<i>12 Інформаційні технології</i>
Спеціальність	<i>126 Інформаційні системи та технології, 121 Інженерія програмного забезпечення, 123 Комп'ютерна інженерія</i>
Освітня програма	<i>Інженерія програмного забезпечення інформаційних систем Інженерія програмного забезпечення комп'ютерних систем Інтегровані інформаційні системи Інформаційні управляючі системи та технології Інформаційне забезпечення робототехнічних систем Комп'ютерні системи та мережі</i>
Статус дисципліни	<i>Вибіркова (вибіркові освітні компоненти)</i>
Форма навчання	<i>Очна(денна)</i>
Рік підготовки, семестр	<i>3 курс, осінній семестр</i>
Обсяг дисципліни	<i>4 кредити 120 годин (36 годин – лекції, 18 годин – лабораторні роботи. 66 годин – СРС)</i>
Семестровий контроль/ контрольні заходи	<i>Залік/ лабораторні роботи, модульна контрольна робота</i>
Розклад занять	<i>https://schedule.kpi.ua</i>
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	<i>Лектор: к.т.н., Крамар Юлія Михайлівна, kramar.yuliia@edu.kpi.ua, моб. +38(050)3420-99-10 Лабораторні: Крамар Юлія Михайлівна</i>
Розміщення курсу	<i>https://classroom.google.com/c/ODOzMTUyNDcxMjky?cjc=2yvc25d3 доступ за запрошенням викладача</i>

Програма навчальної дисципліни

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

Мета навчальної дисципліни полягає у формуванні в студентів системи знань щодо принципів застосування шаблонів проектування та архітектурних підходів під час розробки програмного забезпечення на платформі .NET. Дисципліна спрямована на вивчення шаблонів проектування об'єктно-орієнтованих програм, сучасних архітектур програмних систем, а також

на набуття практичних навичок розробки web-застосунків і сервісів із використанням ASP.NET Core, Web API та архітектурного стилю REST мовою C#.

Предмет навчальної дисципліни — методи, підходи та архітектурні рішення, що застосовуються при проектуванні й розробці програмного забезпечення, зокрема web-застосунків, RESTful сервісів та інформаційних систем на платформі .NET.

Завдання навчальної дисципліни:

- опанування парадигм і методологій програмування, особливостей застосування шаблонів проектування в об'єктно-орієнтованих системах;
- засвоєння шаблонів проектування, принципів SOLID та архітектурних підходів до побудови програмного забезпечення;
- вивчення архітектурних стилів і шаблонів, зокрема багат шарової архітектури, MVC, Clean Architecture, а також клієнт–серверної взаємодії у RESTful Web API;
- набуття знань щодо проектування та реалізації web-застосунків і сервісів в ASP.NET Core;
- розуміння проблем і переваг повторного використання програмного коду при проектуванні програмного забезпечення;
- формування вмінь застосовувати шаблони проектування та архітектурні рішення у професійній діяльності (під час проектування, розробки та командної роботи);
- набуття навичок проектування складного об'єктно-орієнтованого програмного забезпечення, REST API та web-сервісів на платформі .NET;
- отримання практичних навичок використання мови програмування C#, ASP.NET Core та пов'язаних технологій для розробки програмного забезпечення.

У результаті опанування дисципліни формуються такі компетентності:

- здатність проектувати, розробляти та реалізовувати web-застосунки, RESTful сервіси та інформаційні системи на платформі .NET;
- здатність розробляти бізнес-рішення, аналізувати вимоги та оцінювати сучасні технологічні й архітектурні підходи;
- здатність до вибору, проектування та реалізації компонентів інформаційних систем з урахуванням архітектурних принципів.

Очікувані результати навчання:

- знання методологій, технологій і архітектур проектування та реалізації програмного забезпечення на платформі .NET;
- уміння розробляти та використовувати ASP.NET Core Web API відповідно до принципів REST;
- здатність застосовувати сучасні засоби, компоненти та технології для створення web-застосунків і сервісів;
- знання мов і технологій програмування, необхідних для розробки програмного забезпечення та інформаційних систем;
- уміння створювати програмне забезпечення, що входить до складу web- та корпоративних інформаційних систем;
- здатність використовувати сучасні інструментальні та програмні засоби для розробки якісного, масштабованого та підтримуваного програмного забезпечення.

Практичний досвід, набутий у межах дисципліни, включає постановку задач, розробку технічного завдання на проектування програмного забезпечення та RESTful web-сервісів, опис алгоритмів розв'язання задач, розробку програмного коду мовою C# з використанням ASP.NET Core Web API, а також тестування, налагодження й документування програмного забезпечення.

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

Під час вивчення цієї навчальної дисципліни використовуються базові знання студентів з алгоритмів і структур даних, програмування, об'єктно-орієнтованого програмування та баз даних.

Знання й навички, набуті студентами у процесі опанування дисципліни, застосовуються під час вивчення подальших дисциплін, пов'язаних із розробкою програмного забезпечення, а також у ході навчальної та виробничої практики і під час дипломного проєктування.

3. Зміст навчальної дисципліни

Розділ 1.

Тема 1.1 Серіалізація

Тема 1.2 Багатопоточність

Тема 1.3 Асинхронне програмування

Тема 1.4 Робота з базами даних у застосунках

Тема 1.5 Модульне тестування

Розділ 2.

Тема 2.1 Шаблони роботи з даними Репозиторій та Одиниця роботи

Тема 2.2 Введення в архітектурні шаблони. Багаторівнева архітектура

Тема 2.3 Архітектурний шаблон MVC та його модифікації

Розділ 3.

Тема 3.1 Введення в web технології. ASP .Net Core

Тема 3.2 Структура додатку ASP .Net Core. Web API mini

Тема 3.3 Життєвий цикл ASP.NET Core-застосунку.

Тема 3.4 Контролери та методи дії. Маршрутизація

Тема 3.5 ASP.NET Core MVC

Тема 3.6 REST та RESTful сервіси

Тема 3.7 Проєктування REST API

Тема 3.8 Стан застосунку. Кукі, сесії

Тема 3.9 Тестування та розгортання web-застосунків

Тема 3.10 Безпека web-застосунків і сервісів

4. Навчальні матеріали та ресурси

Базова література

1. Дистанційний курс «Сучасні технології розробки WEB-застосунків на платформі Microsoft.NET» для бакалаврів 3-го курсу факультету інформатики та обчислювальної техніки. Адреса розміщення: <https://classroom.google.com/c/ODQzMTUyNDcxMjky?cjc=2yvc25d3> Код курсу: 2yvc25d3
2. Architectural principles [Електронний ресурс] / Microsoft .NET. – 2021–2025. – Режим доступу:
3. <https://learn.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/architectural-principles> (дата звернення: 25.06.2025)

4. Best practices for RESTful web API design [Електронний ресурс] / Microsoft .NET. – 2021–2025. – Режим доступу: <https://learn.microsoft.com/en-us/azure/architecture/best-practices/api-design> (дата звернення: 25.06.2025)
5. Web API implementation [Електронний ресурс] / Microsoft .NET. – 2021–2025. – Режим доступу: <https://learn.microsoft.com/en-us/azure/architecture/best-practices/api-implementation> (дата звернення: 25.06.2025).

Додаткова література

1. Jeffrey Richter CLR via C# (Developer Reference) 4th Edition — Microsoft Press, 2012. — 896 с.
2. ASP.NET Tutorial [Електронний ресурс]. – Режим доступу: <https://www.tutorialspoint.com/asp.net/index.htm> (дата звернення: 25.06.2025)
3. Головна сторінка MSDN [Електронний ресурс]. – Режим доступу: <http://msdn.microsoft.com> (дата звернення: 25.06.2025)
4. ASP.NET Tutorial [Електронний ресурс]. – Режим доступу: <https://dotnet.microsoft.com/en-us/apps/aspnet> (дата звернення: 25.06.2025)
5. Entity Framework Tutorial [Електронний ресурс]. – Режим доступу: <https://www.entityframeworktutorial.net/> (дата звернення: 25.06.2025)

5. Навчальний контент

5.1. Лекційні заняття

Матеріали для вивчення дисципліни розміщені в електронному вигляді на сайті <https://learn.epam.com> та в гугл-класі, посилання надаються студентам на початку семестру. Контент доступний студенту із будь-якого місця в мережі Інтернет.

Лекції по дисципліні проводяться із використанням сучасних мультимедійних презентаційних технологій.

Таблиця 1

Лекційні заняття

Лекції	Назва теми лекції та перелік основних питань
1	1.1 Серіалізація Поняття та призначення серіалізації даних. Основні формати серіалізації (JSON, XML, Binary). Використання серіалізації в .NET-застосунках для збереження та передачі даних.
2	1.2 Багатопоточність Основи багатопоточного програмування. Потоки виконання, синхронізація доступу до спільних ресурсів, проблеми конкурентності та способи їх розв'язання в мові C#.
3	1.3 Асинхронне програмування Асинхронні операції та їх роль у підвищенні продуктивності застосунків. Ключові концепції <code>async/await</code> , <code>Task</code> , особливості використання асинхронного програмування в .NET.
4	1.4 Робота з базами даних у застосунках Принципи доступу до даних у програмних застосунках. Основи використання ORM. Entity Framework.
5	1.5 Модульне тестування Призначення та роль модульного тестування в процесі розробки програмного забезпечення. Основні підходи, принципи ізоляції коду, огляд інструментів тестування в

	.NET.
6	2.1 Шаблони роботи з даними: Репозиторій та Одиниця роботи Призначення та особливості шаблонів Repository і Unit of Work. Розділення логіки доступу до даних і бізнес-логіки, переваги використання в прикладних застосунках.
7	2.2 Введення в архітектурні шаблони. Багаторівнева архітектура Поняття архітектурних шаблонів програмного забезпечення. Багаторівнева архітектура, її складові, принципи взаємодії між рівнями та переваги використання.
8	2.3 Архітектурний шаблон MVC та його модифікації Сутність архітектурного шаблону Model–View–Controller. Розподіл відповідальностей між компонентами, огляд модифікацій та сфер застосування MVC.
9	3.1 Введення в web-технології. ASP.NET Core Основи web-розробки, клієнт–серверна архітектура, роль веб-фреймворків. Призначення та можливості платформи ASP.NET Core.
10	3.2 Структура застосунку ASP.NET Core. Web API Minimal Організація проєкту ASP.NET Core. Основні компоненти застосунку, особливості побудови Minimal Web API.
11	3.3 Життєвий цикл ASP.NET Core-застосунку Процес запуску та обробки HTTP-запитів. Middleware, конвеєр обробки запитів, конфігурація та середовище виконання.
12	3.4 Контролери та методи дії. Маршрутизація Призначення контролерів і методів дії. Основи маршрутизації, атрибутна маршрутизація, обробка HTTP-запитів у Web API.
13	3.5 ASP.NET Core MVC Архітектура MVC у ASP.NET Core. Взаємодія моделей, представлень і контролерів, особливості використання MVC у web-застосунках.
14	3.6 REST та RESTful сервіси Принципи архітектурного стилю REST. Ресурси, HTTP-методи, статус-коди та особливості побудови RESTful сервісів.
15	3.7 Проєктування REST API Основні підходи до проєктування REST API. Структура ресурсів, версіонування, обробка помилок, документування API.
16	3.8 Стан застосунку Проблема збереження стану в web-застосунках. Використання cookies, сесій та інших механізмів управління станом.
17	3.9 Безпека web-застосунків і сервісів Основні загрози безпеці web-застосунків. Принципи захисту, автентифікація та авторизація, забезпечення безпеки Web API.
18	3.10 Тестування та розгортання web-застосунків Підходи до тестування web-застосунків і сервісів. Основи розгортання, конфігурація середовищ, підтримка працездатності застосунків.

5.2. Лабораторні заняття

Лабораторні заняття виконуються з використанням мови програмування C# та версій середовищ розробки програм (IDE), які надаються розробниками для навчальних цілей безкоштовно. Вказівки до виконання лабораторних робіт включені в онлайн курс на платформі <https://learn.epam.com>. Практичні завдання в онлайн-курсі складають основу лабораторних робіт. Особливістю форми представлення практичних завдань онлайн-курсу є залучення спеціальної системи Autocode, яка дозволяє автоматизувати перевірку коду. Якщо студент виконав завдання, та Autocode оцінив позитивно код лабораторної роботи та стилістику його написання, студент захищає роботу на занятті з викладачем, відповідаючи на теоретичні запитання. Також викладач визначає академічну добросовісність студента та відсутність плагіату. Для виконання кожної лабораторної роботи студенту надається не більше 2 тижнів.

За умови навчання у форматі online заняття, у тому числі контрольні заходи, проводяться з використанням сервісу Google, MS Teams та цифрової платформи Learn.

Вихідний код лабораторних робіт має бути розміщеним кожним студентом особисто у глобальному репозиторії <https://autocode.git.epam.com>.

Таблиця 2

Лабораторні заняття

№ з/п	Назва лабораторної роботи (комп'ютерного практикуму)	Ресурс	Кількість ауд. годин
1	Дослідження технології серіалізації	Autocode	2
2	Дослідження багатопоточності та асинхронного програмування	Autocode	2
3	Модульне тестування	Autocode	2
4	Робота з даними за технологією Entity Framework	Autocode	2
5	Дослідження багаторівневої архітектури web-застосунків	Метод.вказ.	4
6	Вивчення технології розробки web api сервісів	Метод.вказ.	6
		Разом	18

5.3. Самостійна робота студента

До самостійної роботи студента відноситься виконання індивідуальних завдань з тематики, яка виноситься на лабораторні роботи, а також опрацювання теоретичного матеріалу за наданими текстами лекцій та додатковою літературою (згідно таблиць 1 та 2). Усі навчальні матеріали (тексти лекцій, методичні вказівки до виконання лабораторних робіт) розміщені в електронному вигляді на сайті (<https://learn.epam.com/study/>), а також в гугл-класі. Навчальний контент доступний із будь-якого місця в мережі Інтернет.

На самостійну роботу студент має витратити кількість годин, що співмірна із кількістю годин, проведених ним на аудиторних заняттях.

Таблиця 3

Самостійна робота

№ з/п	Назва тем самостійної роботи	Год.
1	Підготовка до лабораторних робіт	30
2	Опрацювання матеріалу лекційних занять	30
3	Підготовка до модульної контрольної роботи	6

	Разом	66
--	-------	----

5.4 Модульна контрольна робота

Метою модульної контрольної роботи є закріплення та перевірка теоретичних знань із освітнього компонента, набуття студентами практичних навичок самостійного вирішення завдань.

Модульна контрольна робота (МКР) виконується в кінці курсу на останньому тижні.

6. Політика навчальної дисципліни (освітнього компонента) Як викладач, так і студент зобов'язані дотримуватись [Кодексу честі Національного технічного університету України «Київський політехнічний інститут»](#).

Основні положення політики:

- відвідування лекційних та лабораторних занять є обов'язковою (за можливості) складовою вивчення матеріалу;
- студент повинен вивчати дисципліну самостійно та вчасно;
- впродовж занять студенти можуть задавати питання стосовно матеріалу, що викладається; студенти мають можливість підняти будь-яке питання, яке стосується процедури проведення та оцінювання контрольних заходів;
- студенти мають право оскаржити результати контрольних заходів, аргументовано пояснивши з яким критерієм не погоджуються відповідно до критеріїв оцінки та/або зауважень;
- у випадку виявлення факту академічної недобросесної роботи лабораторна робота не зараховується.

Відвідування є обов'язковим (за можливості) (за винятком випадків, коли існує поважна причина, наприклад, хвороба чи дозвіл працівників деканату). Якщо студент не може бути присутнім на заняттях, він все одно несе відповідальність за опрацювання теоретичного матеріалу та виконання лабораторних робіт згідно графіку.

Політика перескладань. Пропущені (за наявності поважних причин) роботи можна здати після дедлайну без втрати балів, але не пізніше останнього лабораторного заняття в семестрі і не більше одної роботи за раз. Якщо студент не набрав достатню кількість балів, то він може бути допущений до основної сесії (матиме багато додаткових питань на заліку), але не допущений до перескладань на додатковій сесії. Такі обмеження стимулюють студента організувати систематичне виконання завдань та не допускати значного накопичення не зданих лабораторних робіт на кінець семестру. Перенесення вивчення дисципліни на наступний семестр як додаткової послуги неможливе.

Визнання результатів навчання, набутих у неформальній/інформальній освіті. Порядок визнання таких результатів регламентується Положенням про визнання результатів навчання, набутих у неформальній / інформальній освіті (<https://osvita.kpi.ua/index.php/node/179>). Можуть бути зараховані окремі змістовні модулі або теми дисципліни. В такому разі здобувач звільняється від виконання відповідних завдань, отримуючи за них максимальний бал відповідно до рейтингової системи оцінювання.

Політика академічної поведінки та доброчесності: конфліктні ситуації мають відкрито обговорюватись в академічних групах з викладачем, необхідно бути взаємно толерантним, поважати думку іншого. Плагіат та інші форми нечесної роботи неприпустимі. Всі завдання студент має виконати самостійно із використанням рекомендованої літератури й отриманих знань та навичок. Цитування в письмових роботах допускається тільки із відповідним посиланням на авторський текст. Недопустимі підказки і списування у ході захисту робіт, на тестах, на заліку.

Обов'язковою умовою виконання завдань з освітньої компоненти є дотримання політики та принципів академічної доброчесності (<https://kpi.ua/academic-integrity>), які, у тому числі, викладено у Кодексі честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (<https://kpi.ua/code>), Положенні про систему

запобігання академічному плагіату в КПІ ім. Ігоря Сікорського (<https://osvita.kpi.ua/node/47>). У разі виявлення дублювання робіт, плагіату роботи здобувачі отримують нульовий рейтинг.

Норми академічної етики: дисциплінованість; дотримання субординації; чесність; відповідальність; робота в аудиторії з відключеними мобільними телефонами. Повага один до одного дає можливість ефективніше досягати поставлених результатів. Якщо студент використовує свій ноутбук чи телефон для аудіо- чи відеозапису, необхідно заздалегідь отримати дозвіл викладача.

Дотримання академічної доброчесності студентів й викладачів регламентується кодексом честі Національного технічного університету України «Київський політехнічний інститут», положенням про організацію освітнього процесу в КПІ ім. Ігоря Сікорського. За порушення принципів академічної доброчесності, зокрема плагіат лабораторних робіт, студент втрачає всі бали за дану роботу. Студент має вивчати дисципліну протягом семестру, дотримуючись календарного плану. Усі завдання студент має виконувати самостійно і вчасно.

Політика використання штучного інтелекту. Використання штучного інтелекту (далі, ШІ) регламентується «Політикою використання штучного інтелекту для академічної діяльності в КПІ ім. Ігоря Сікорського» (<https://osvita.kpi.ua/node/1225>). Усі навчальні завдання з дисципліни мають бути результатом власної оригінальної роботи здобувача.

Для перевірки як теоретичних знань, так і практичних навичок в семестрі передбачена модульна контрольна робота (МКР), яка може складатися з двох частин, які проводяться під час семестрових атестацій.

Завдання та запитання модульної контрольної роботи – це тренування студента до запитань та завдань екзаменаційного білету. Дату та час проведення робіт оголошує лектор на початку семестру, відмічає її у журналі та нагадує про проведення роботи на лекційних заняттях. Переписати модульну контрольну роботу неможливо.

Оцінювання студентів здійснюється згідно рейтингової оцінки рівня підготовки студентів з дисципліни. Поточний стан успішності студенти можуть бачити в системі «Електронний кампус». Рейтингова система оцінювання з кредитного модуля описана у наступному розділі робочої програми.

7. Види контролю та рейтингова система оцінювання результатів навчання (PCO)

7.1. Поточний контроль

Поточний контроль успішності засвоєння знань виконується шляхом виконання ними завдань лабораторних робіт та МКР.

Семестровий рейтинг студента з дисципліни складається з балів, що він отримує у підсумку за:

- лабораторні роботи;
- модульну контрольну роботу.

7.2. Система рейтингових (вагових) балів та критерії оцінювання

Лабораторні роботи

Вагові бали кожної лабораторної роботи наведені у Таблиці 4. Сумарний ваговий бал за даний контрольний захід (*r_з*) складає **42 бали**.

Критерії оцінювання лабораторних робіт включають якість її виконання та захисту (Таблиця 4)

Таблиця 4

Вагові бали та критерії оцінювання лабораторних робіт.

№	Назва роботи	Бали		
		Виконання	Захист	Сума

1	Дослідження технології серіалізації	7		7
2	Дослідження багатопоточності та асинхронного програмування	7		7
3	Модульне тестування	7		7
4	Робота з даними за технологією Entity Framework	7		7
5	Дослідження багаторівневої архітектури web-застосунків	7	8	15
6	Вивчення технології розробки web api сервісів	18	24	42
	Модульна контрольна робота	15		15
	Разом	68	32	100

Критерії оцінювання лабораторних робіт 1-4 (оцінюються системою автоматичної перевірки завдань Autocode):

- виконане завдання з автоматичною перевіркою на Autocode – 7 балів;
- роботи 5-6 виконуються за індивідуальним варіантом, наданим викладачем, код демонструється викладачу і захищається на занятті (лабораторна робота зараховується студенту тільки при умові її захисту).

Критерії оцінювання лабораторної роботи 5:

- “**відмінно**”, повна відповідь (не менше 95% потрібної інформації); - 17-18 балів
- “**добре**”, достатньо повна відповідь (не менше 75% потрібної інформації, або незначні неточності); - 15-16 балів
- “**задовільно**”, неповна відповідь (не менше 60% потрібної інформації та деякі помилки); - 13-14 балів
- “**незадовільно**”, незадовільна відповідь. - < 13 балів

Критерії оцінювання лабораторної роботи 6:

- “**відмінно**”, повна відповідь (не менше 95% потрібної інформації); - 41-44 балів
- “**добре**”, достатньо повна відповідь (не менше 75% потрібної інформації, або незначні неточності); - 33-40 балів
- “**задовільно**”, неповна відповідь (не менше 60% потрібної інформації та деякі помилки); - 26-32 балів
- “**незадовільно**”, незадовільна відповідь. - < 26 балів

Модульна контрольна робота

Ваговий бал МКР –15 балів.

Критерії оцінювання кожної частини МКР:

- “**відмінно**”, повна відповідь (не менше 95% потрібної інформації); - 14-15 балів
- “**добре**”, достатньо повна відповідь (не менше 75% потрібної інформації, або незначні неточності); - 11-13 балів

- “*задовільно*”, неповна відповідь (не менше 60% потрібної інформації та- 12-13 балів деякі помилки);
- “*незадовільно*”, незадовільна відповідь. - <9-11 балів

Умови позитивної проміжної атестації

Для отримання “зараховано” з першої проміжної атестації (7 тиждень) студент повинен мати не менше ніж 14 балів (за умови, якщо на початок 7 тижня згідно з календарним планом контрольних заходів “ідеальний” студент має отримати 21 бал).

Для отримання “зараховано” з другої проміжної атестації (13 тиждень) студент повинен мати не менше ніж 30 балів (за умови, якщо на початок 13 тижня згідно з календарним планом контрольних заходів “ідеальний” студент має отримати 43 бали).

Семестровий контроль

Семестровий контроль результатів навчання проводиться у вигляді заліку. Студенти, які виконали всі умови допуску до заліку та набрали 60 і більше балів, отримують відповідну рейтингову оцінку без потреби проходження заходу семестрового контролю.

Студенти, які виконали всі умови допуску до заліку, але набрали менше 60, проходять захід семестрового контролю, при цьому семестровий контроль оцінюється в 100 балів.

Якщо студент склав залік та за його результатами отримав меншу кількість балів за кількість балів, отриманих за результатами заходів поточного контролю, рейтингова оцінка визначається за результатами заліку.

Відповідність рейтингових балів оцінкам за університетською шкалою наведена в таблиці 5.

Таблиця 5. Відповідність рейтингових балів оцінкам за університетською шкалою

Кількість балів	Оцінка
95 - 100	Відмінно
85 - 94	Дуже добре
75 - 84	Добре
65 - 74	Задовільно
60 - 64	Достатньо
менше 60	Незадовільно

Робочу програму навчальної дисципліни (силабус):

Складено доцентом, к.т.н, доцентом, Крамар Юлією Михайлівною.

Ухвалено кафедрою ІІІ (протокол №2/1 від 10.10.2025р.)

Погоджено Методичною комісією факультету (протокол №3 від 17.10.2025р.)