

**Чернівецький національний університет імені Юрія Федьковича**  
(повне найменування закладу вищої освіти)  
**Навчально-науковий інститут фізико-технічних та комп'ютерних наук**  
(назва інституту/факультету)  
**Кафедра комп'ютерних наук**  
(назва кафедри)

**СИЛАБУС**  
**навчальної дисципліни**

**Розробка додатків на платформі .NET**

**вибіркова**

<b>Освітньо-професійна програма</b>	<b><u>«Інтелектуальний аналіз даних в комп'ютерних інформаційних системах»</u></b>
<b>Спеціальність</b>	<b><u>№122 «Комп'ютерні науки»</u></b>
<b>Галузь знань</b>	<b><u>12 «Інформаційні технології»</u></b>
<b>Рівень вищої освіти</b>	<u>перший (бакалаврський)</u>  <u>Навчально-науковий інститут фізико-технічних та комп'ютерних наук.</u> <u>Відділ «Комп'ютерних технологій»</u>
<b>Мова навчання:</b>	<u>українська</u>
<b>Розробники:</b>	<u>Томка Ю.Я., к.ф.-м.н., доцент</u>
<b>Профайл викладача:</b>	<u><a href="https://kkn.chnu.edu.ua/about-kafedra/spivrobotnyky/tomka_yurii/">https://kkn.chnu.edu.ua/about-kafedra/spivrobotnyky/tomka_yurii/</a></u>
<b>Контактний тел.</b>	<u>+380957692191</u>
<b>E-mail:</b>	<u><a href="mailto:y.tomka@chnu.edu.ua">y.tomka@chnu.edu.ua</a></u>
<b>Стор-ка курсу в Moodle:</b>	<u><a href="https://moodle.chnu.edu.ua/course/view.php?id=3387">https://moodle.chnu.edu.ua/course/view.php?id=3387</a></u>
<b>Консультації:</b>	<u>На модульних тижнях і перед заліком згідно з затвердженим графіком</u>

## 1. Анотація дисципліни (призначення навчальної дисципліни)

Дисципліна «Розробка додатків на платформі .NET» є частиною освітньо-професійної програми «Інтелектуальний аналіз даних в комп'ютерних інформаційних системах» першого рівня вищої освіти за спеціальністю №122 «Комп'ютерні науки» галузі знань №12 «Інформаційні технології».

Під час вивчення даної дисципліни студенти набудуть:

I. загальних компетентностей:

- **ЗК03.** Знання та розуміння предметної області та розуміння професійної діяльності.
- **ЗК05.** Здатність спілкуватися іноземною мовою.
- **ЗК06.** Здатність вчитися і оволодівати сучасними знаннями.
- **ЗК08.** Здатність генерувати нові ідеї (креативність).
- **ЗК09.** Здатність працювати в команді.

II. спеціальних (фахових, предметних) компетентностей:

- **СК08.** Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальнення, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами та алгоритмами обчислень, структурами даних і механізмами управління.
- **СК12.** Здатність забезпечити організацію обчислювальних процесів в інформаційних системах різного призначення з урахуванням архітектури, конфігурування, показників результативності функціонування операційних систем і системного програмного забезпечення.
- **СК14.** Здатність застосовувати методи та засоби забезпечення інформаційної безпеки, розробляти й експлуатувати спеціальне програмне забезпечення захисту інформаційних ресурсів об'єктів критичної інформаційної інфраструктури.

III. Програмними результатами навчання є:

- **ПР05.** Проектувати, розробляти та аналізувати алгоритми розв'язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислюваних функцій.
- **ПР09.** 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.
- **ПР10.** Використовувати інструментальні засоби розробки клієнт-серверних застосувань, проектувати концептуальні, логічні та фізичні моделі баз даних, розробляти та оптимізувати запити до них, створювати розподілені бази даних, сховища та вітрини даних, бази знань, у тому числі на хмарних сервісах, із застосуванням мов веб-програмування.
- **ПР15.** Розуміти концепцію інформаційної безпеки, принципи безпечного проектування програмного забезпечення, забезпечувати безпеку комп'ютерних мереж в умовах неповноти та невизначеності вихідних даних.

## 2. Мета навчальної дисципліни

Метою викладання дисципліни «Технології розробки .NET» є підготовка спеціалістів із розробки веб-застосунків на основі платформи ASP.NET Core та мови C#.

## 3. Завдання

Основними завданнями курсу «Технології розробки .NET» є формування у студента теоретико-практичного базису щодо сучасних підходів і методик розробки різного роду веб-застосунків на платформі .NET в розрізі найбільш актуальних для ринку ентерпрайз архітектур та систем контролю версій.

#### 4. Пререквізити:

- ППО5. «Веб-технології та веб-дизайн»
- ППО6. «Об'єктно-орієнтоване програмування»
- ППО8. «Організація баз даних та знань»
- ППО11. «Паттерни проектування»

#### 5. Результати навчання

В результаті вивчення курсу студент повинен:

**знати:**

- відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізів та математичного моделювання для розробки програмного забезпечення (ПР05 ОПП «Інженерія програмного забезпечення»);
- базові теоретичні поняття, що пов'язані із розробкою та забезпеченням безпеки Web-застосунків;
- основні технологічні можливості розробки Web-застосунків із використанням актуальних версій платформи ASP.NET;
- способи аутентифікації, авторизації при розробці безпечних ASP.NET-застосунків;
- основні можливості інструментарію ADO.NET, LINQ та Entity Framework при роботі із базами даних;
- практичні аспекти найбільш актуальних і поширених архітектурних рішень при розробці веб-додатків;
- можливості фреймворку Blazor при розробці UI складової веб-застосунків на основі ASP.NET;
- теоретико-практичні особливості роботи із Ocelot;
- теоретико-практичні особливості роботи із Redis;
- теоретико-практичні особливості роботи із Polly;
- теоретико-практичні особливості роботи із gRpc;
- теоретико-практичні особливості роботи із ElasticSearch;
- теоретико-практичні особливості роботи із Identity Server;
- теоретико-практичні особливості роботи із GraphQL;
- можливості методології розробки програмного забезпечення DevOps.

**вміти:**

- імплементувати серверні ASP.NET застосунки з допомогою ASP.NET Core на основі наступних архітектурних рішень: N-layer, Clean Architecture, Microservices.
- використовувати найбільш поширені шаблони проектування та парадигми;
- працювати із ADO.NET та MS SQL Server;
- працювати із Entity Framework;
- реалізовувати аутентифікацію та авторизацію для мікросервісної архітектури;
- імплементувати можливості моніторингу, стійкості відповідних рішень на базі мікросервісної архітектури.
- імплементувати високоефективний міжсервісний зв'язок за допомогою gRPC та меседж брокерів;
- здійснювати контейнеризацію та оркестрацію веб-застосунків для мікросервісної архітектури;

## 6. Опис навчальної дисципліни

### 6.1. Загальна інформація

Назва навчальної дисципліни _ Технології розробки .NET												
Форма навчання	Рік підготовки	Семестр	Кількість			Кількість годин						Вид підсумкового контролю
			кредитів	годин	змістових модулів	лекції	практичні	семінарські	лабораторні	самостійна робота	індивідуальні завдання	
Денна	3	5	4	120	2	30	-	-	30	60	-	залік
Заочна	-	-	-	-	-	-	-	-	-	-	-	-

### 6.2. Дидактична карта навчальної дисципліни

Назви змістових модулів і тем	Кількість годин													
	денна форма							заочна форма						
	усього	у тому числі					усього	у тому числі						
		л	п	лаб	інд	с.р.		л	п	лаб	інд	с.р.		
1	2	3	4	5	6	7	8	9	10	11	12	13		
<b>Теми лекційних занять</b>	<b>Змістовий модуль 1. Розробка веб-застосунків із використанням ASP.NET CORE. Back-End</b>													
Тема 1. Вступ та основи ASP.NET Core. Принципи побудови БД для мікросервісів	8	2	-	2	-	4								
Тема 2. <b>N-tier architecture.</b> Багатошарова архітектура на основі ADO.NET&Dapper.	8	2	-	2	-	4								
Тема 3. <b>N-tier architecture.</b> Багатошарова архітектура на основі Entity Framework Core. Data Access Layer.(LINQ to Entities. Паттерни Generic Repository/Unit of Work)	8	2	-	2	-	4								
Тема 4. <b>N-tier architecture.</b> Багатошарова архітектура на основі Entity Framework Core. Business Logic Layer та Web-Layer.	8	2	-	2	-	4								
Тема 5. <b>DDD та Clean Architecture.</b> CQRS. MediatR.	8	2	-	2	-	4								
Тема 6. Питання аутентифікації та авторизації. ASP.NET Core	8	2	-	2	-	4								

Identity. JWT-токен.													
Тема 7. Клієнт-серверні застосунки на платформі Asp.Net та GraphQL	8	2		2		4							
Разом за ЗМ1	<b>56</b>	<b>14</b>	-	<b>14</b>	-	<b>28</b>							
<b>Теми лекційних занять</b>	<b>Змістовий модуль 2. Мікросервісна архітектура при розробці Web-додатків на платформі ASP.NET Core</b>												
Тема 8. <b>Microservice architecture.</b> Основні поняття та концепції.	6	2	-	-	-	4							
Тема 9. <b>Microservice architecture.</b> In-memory & distributed (Redis) caching in Asp.net Core	8	2	-	2	-	4							
Тема 10. <b>Microservice architecture.</b> Побудова Api Gateway із використанням Ocelot. Дизайн-патерн «Агрегатор»	8	2	-	2	-	4							
Тема 11. <b>Microservice architecture.</b> Використання gRPC для побудови високопродуктивного міжмікросервісного зв'язку на платформі .Net	8	2	-	2	-	4							
Тема 12. <b>Microservice architecture.</b> Microservices Event Driven Architecture на базі RabbitMQ та MassTransit	8	2	-	2	-	4							
Тема 13. <b>Microservice architecture.</b> Захист мікросервісів за допомогою Identity Server4, OAuth 2.0 та OpenId Connect під управлінням Ocelot Api Gateway	10	2	-	4	-	4							
Тема 14. <b>Microservice architecture.</b> Microservices Observability, Resilience, Monitoring on .Net	8	2	-	2	-	4							
Тема 15. Основи розгортання .Net додатків на мікросервісній архітектурі у Kubernetes-кластер	8	2	-	2	-	4							
Разом за ЗМ 2	<b>64</b>	<b>16</b>	-	<b>16</b>	-	<b>32</b>							
<b>Усього годин</b>	<b>120</b>	<b>30</b>	-	<b>30</b>	-	<b>60</b>							

### 6.3. Зміст завдань для самостійної роботи

№	Назва теми
1	<p>Вступ та основи ASP.NET Core:</p> <ul style="list-style-type: none"> <li>Загальна структура проекту;</li> <li>Основи конфігурації</li> <li>Dependency injection;</li> <li>Логірування;</li> <li>Маршрутизація;</li> <li>Представлення та компоновка;</li> <li>Контролери та дії</li> </ul>
2	<p>Багатошарова архітектура на основі ADO.NET/Dbapper.</p> <ul style="list-style-type: none"> <li>Поняття багатошарової архітектури;</li> <li>DAL;</li> <li>BLL;</li> <li>WEB;</li> <li>CRUD – операції;</li> <li>Підеднаний режим роботи ADO.NET;</li> <li>SQL;</li> <li>Dbapper</li> </ul>
3	<p>Багатошарова архітектура на основі Entity Framework Core. Data Access Layer. LINQ to Entities. Паттерни Generic Repository/Unit of Work:</p> <ul style="list-style-type: none"> <li>Linq to Objects;</li> <li>Linq to Entities;</li> <li>POCO Objects;</li> <li>Fluent API;</li> <li>Generic Repository;</li> <li>UOW;</li> <li>DbSet;</li> <li>DbContext;</li> <li>Робота із міграціями</li> </ul>
4	<p>Багатошарова архітектура на основі Entity Framework Core. Business Logic Layer та Web-Layer:</p> <ul style="list-style-type: none"> <li>User Stories;</li> <li>Use Cases;</li> <li>Сервіса/Менеджери;</li> <li>DTO;</li> <li>Мапінг даних;</li> <li>Dependency Injection;</li> <li>AutoMapper;</li> <li>Маршрутизація;</li> </ul>
5	<p>Чиста архітектура. CQRS. MediatR.</p> <ul style="list-style-type: none"> <li>Детальний аналіз кожного шару;</li> <li>Загальна структура проекту;</li> <li>Value Objects;</li> <li>CQRS;</li> <li>MediatR;</li> <li>Fluent Api;</li> </ul>
6	<p>Питання аутентифікації та авторизації. ASP.NET Core Identity. JWT-токена.</p> <ul style="list-style-type: none"> <li>Вступ до ASP.NET Core Identity</li> <li>Основні класи в ASP.NET Core Identity</li> <li>Додавання Identity до проекту з нуля</li> </ul>

	<p>Реєстрація та створення користувачів у Identity  Авторизація користувачів у Identity  Керування користувачами  Зміна пароля  Валідація пароля  Валідація користувача  Управління ролями  Ініціалізація БД ролями та користувачами  --  Введення в аутентифікацію та авторизацію  Аутентифікація за допомогою JWT-токенів  Авторизація за допомогою JWT-токенів у клієнті JavaScript  Аутентифікація за допомогою куки  HttpContext.User, ClaimPrincipal та ClaimsIdentity  ClaimPrincipal та об'єкти Claim  Авторизація за ролями  Авторизація на основі Claims  Створення обмежень для авторизації</p>
7	<p>Клієнт-серверні застосунки на платформі Asp.Net та GraphQL  Building a basic GraphQL server API.  Controlling nullability.  Understanding GraphQL query execution and DataLoader.  GraphQL schema design approaches.  Understanding middleware.  Adding complex filter capabilities.  Adding real-time functionality with subscriptions.  Testing the GraphQL server.</p>
8	<p>Microservice architecture. Основні поняття та концепції.</p>
9	<p>Microservice architecture. In-memory &amp; distributed (Redis) caching in Asp.net Core</p> <ul style="list-style-type: none"> <li>Data types</li> <li>Using Redis</li> <li>Managing Redis</li> <li>Redis Stack</li> </ul>
10	<p>Microservice architecture. Побудова Api Gateway із використанням Ocelot.  Дизайн-патерн «Агрегатор»  Routing</p> <ul style="list-style-type: none"> <li>Authentication</li> <li>Authorization</li> <li>Websockets</li> <li>Administration</li> <li>Rate Limiting</li> <li>Caching</li> <li>Quality of Service</li> <li>Headers Transformation</li> <li>HTTP Method Transformation</li> <li>Claims Transformation</li> <li>Logging</li> <li>Tracing</li> <li>Request Id / Correlation Id</li> <li>Middleware Injection and Overrides</li> <li>Load Balancer</li> <li>Delegating Handlers</li> <li>Http Error Status Codes</li> </ul>

11	<p>Microservice architecture. Використання gRPC для побудови високопродуктивного міжмікросервісного зв'язку на платформі .Net</p> <ul style="list-style-type: none"> <li>Служби gRPC у ASP.NET Core</li> <li>Підтримувані платформи</li> <li>Використання gRPC у програмах на основі браузера</li> <li>Параметр Configuration</li> <li>Аутентифікація та авторизація</li> <li>Перехоплювачі gRPC</li> <li>Ведення журналу та діагностика</li> <li>Зауваження щодо безпеки</li> <li>Оптимальні методи підвищення продуктивності</li> <li>Міжпроцесна взаємодія</li> <li>Перекодування gRPC JSON</li> <li>Клієнти та служби з підтримкою Code First</li> <li>Перевірка працездатності gRPC</li> <li>Управління посиланнями protobuf за допомогою .NET gRPC</li> <li>Тестування служб за допомогою gRPCurl</li> <li>Міграція з C-core на gRPC для .NET</li> <li>Навіщо виконувати міграцію WCF на ASP.NET Core gRPC</li> <li>Порівняння служб gRPC із API-інтерфейсами HTTP</li> <li>Приклади</li> <li>Усунення несправностей</li> </ul>
12	<p>Microservice architecture. Microservices Event Driven Architecture на базі RabbitMQ та MassTransit</p> <ul style="list-style-type: none"> <li>Introducing RabbitMQ</li> <li>Introducing RabbitMQ Exchanges</li> <li>Understanding Queueing Architecture</li> <li>Implementing Microservices and Message Queuing</li> <li>Installing and Configuring RabbitMQ</li> <li>Implementing the Queuing Code</li> </ul>
13	<p>Microservice architecture. Захист мікросервісів за допомогою Identity Server4, OAuth 2.0 та OpenId Connect під управлінням Ocelot Api Gateway</p> <ul style="list-style-type: none"> <li>Protecting an API using Client Credentials</li> <li>Interactive Applications with ASP.NET Core</li> <li>ASP.NET Core and API access</li> <li>Using EntityFramework Core for configuration and operational data</li> <li>Using ASP.NET Core Identity</li> <li>Building JavaScript client applications</li> <li>Building Blazor WASM client applications</li> </ul>
14	<p>Microservice architecture. Microservices Observability, Resilience, Monitoring on .Net</p> <ul style="list-style-type: none"> <li>Microservices Observability with Distributed Logging using ElasticSearch</li> <li>Microservices Resilience and Fault Tolerance with Applying Retry and Circuit-Breaker patterns using Polly</li> <li>Microservices Monitoring with Health Checks using WatchDog</li> <li>Microservices Tracing with OpenTelemetry using Zipkin</li> </ul>
15	<p>Основи розгортання .Net додатків на мікросервісній архітектурі у Kubernetes-кластер</p> <ul style="list-style-type: none"> <li>Основи Kubernetes</li> <li>Підняття простого K8s Cluster на Windows</li> <li>Підняття K8s Cluster в AWS - Amazon Web Services</li> <li>Підняття K8s Cluster у GCP - Google Cloud Platform</li> </ul>



Підняття K8s Cluster Для Навчання - Безкоштовно в Інтернеті Створення DockerImage з Dockerfile та Завантаження в DockerHub Головні Об'єкти K8s Створення та керування - Pods Створення та керування - Deployments Створення та керування - Services Створення та керування - Ingress Controllers Створення та управління - Helm Charts
---

## 7. Система контролю та оцінювання

Поточний контроль (ПК) здійснюється під час проведення лекційних, практичних та індивідуально-консультативних занять з метою перевірки рівня засвоєння теоретичних знань та практичних навичок студента. ПК проводиться у формі написання письмових робіт, проміжних тестувань та активності й влучності обговорення відповідних тем під час навчальних занять. Згідно з навчальним планом семестровий контроль з дисципліни «Дискретна математика» відбувається у формі іспиту.

*Відвідування занять.* Відсутність на аудиторному занятті не передбачає нарахування штрафних балів, оскільки фінальний рейтинговий бал студента формується виключно на основі оцінювання результатів навчання. Разом з тим, обговорення результатів виконання тематичних завдань, а також презентація / публічний виступ та участь у обговореннях та доповнення на практичних заняттях оцінюватимуться під час аудиторних занять.

*Оцінювання пропущених контрольних заходів.* Кожен студент має право відпрацювати пропущені з поважної причини (лікарняний, мобільність тощо) заняття за рахунок самостійної роботи.

*Процедура оскарження результатів контрольних заходів оцінювання.* Студент може підняти будь-яке питання, яке стосується процедури контрольних заходів та очікувати, що воно буде розглянуто згідно із наперед визначеними процедурами Положенням про апеляцію на результати підсумкового семестрового контролю знань студентів Чернівецького національного університету імені Юрія Федьковича ([http://www.budarch.chnu.edu.ua/res//budarch/Docs/Polozhennia/11\\_Polozh\\_pro\\_apeliatsiy\\_2020.pdf](http://www.budarch.chnu.edu.ua/res//budarch/Docs/Polozhennia/11_Polozh_pro_apeliatsiy_2020.pdf)).

*Академічна доброчесність.* Політика та принципи академічної доброчесності визначені Етичним кодексом Чернівецького національного університету імені Юрія Федьковича ([https://drive.google.com/file/d/1CB4AIMVXSAykF\\_CepI-k98GPc9E8KznQ/view](https://drive.google.com/file/d/1CB4AIMVXSAykF_CepI-k98GPc9E8KznQ/view)).

*Інклюзивне навчання.* Засвоєння знань та умінь в ході вивчення дисципліни «Дискретна математика» може бути доступним для більшості осіб з особливими освітніми потребами, окрім здобувачів з серйозними вадами зору, які не дозволяють виконувати завдання за допомогою персональних комп'ютерів, ноутбуків та/або інших технічних засобів.

*Навчання іноземною мовою.* У ході виконання завдань студентам може бути рекомендовано звернутися до англомовних джерел.

Підготовка до практичних 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).

*Неформальна освіта.* При наявності сертифікатів про проходження професійно-спрямованих психолого-педагогічних курсів, тренінгів, майстер-класів з неформальної освіти, участь у неформальній освіті студентам зараховувалось до 10% змістового матеріалу, що відповідає прослуханому матеріалу, за умови підготовки ними презентацій та нотаток за матеріалами прослуханого курсу, чи веб-заходу та їх публічного захисту на практичних заняттях.

Також, як можливості неформальної освіти студентам під час вивчення курсу «Розробка додатків на платформі .NET» пропонується проходження курсів з отриманням сертифікатів, як індивідуальне завдання (ІНДЗ). На один модуль не більше 5 балів при 100%

(46 – 80-99%, 36 – 60-79% тощо) проходженні курсів на віртуальних платформах (Prometheus, Coursera та інші).

### Критерії оцінювання результатів навчання з навчальної дисципліни

Залік проводиться у формі, визначеній кафедрою. Заліковий білет включає теоретичні і практичні завдання та містить 4 завдання (2 теоретичних питання, 2 практичні задачі. Результати заліку оцінюються за національною чотирибальною шкалою. Максимальна кількість балів на заліку за шкалою ВНЗ становить 40 балів.

Загальна підсумкова оцінка з дисципліни складається з суми балів за результатами ПМК та за виконання завдань, що виносяться на залік.

Якщо студент на заліку отримав незадовільну оцінку, то це вважається як академічна заборгованість і набрані бали не заносяться до відомості. За графіком деканату студент перескладає залік і його результати заносяться до окремої відомості.

За результатами складання заліку студент отримує:

27,0-30,0 бала – дана розгорнута вичерпна відповідь на теоретичні питання та правильно виконане практичне завдання;

24,0-26,0 бала – студентом допущені незначні помилки у відповіді на теоретичні питання чи практичне завдання;

21,0-23,0 бала – студент допустив значні помилки у відповіді на одне з теоретичних питань чи в практичному завданні;

18,0-20,0 бала – студент, допустивши значні помилки, не дав чіткої відповіді на теоретичні питання, не в повному обсязі виконав практичне завдання;

15,0-17,0 бала – студент не дав відповіді на одне із теоретичних питань, практичне завдання виконане не в повному обсязі;

10,5-14,0 бала – студент не дав відповіді на одне із теоретичних питань, не виконане практичне завдання.

0,1-10,3 бала – студент виконав частину одного з теоретичних питань, практичне завдання не виконане.

0 балів – студент не з'явився на езалік.

### Шкала оцінювання результатів заліку

Підсумкова кількість балів за залік	Оцінка за національною шкалою	Оцінка за шкалою ECTS
<b>27 - 30</b>	зараховано	<b>A</b> (відмінно)
<b>24 – 26</b>	зараховано	<b>B</b> (дуже добре)
<b>21 – 23</b>		<b>C</b> (добре)
<b>18 – 20</b>	зараховано	<b>D</b> (задовільно)
<b>15 – 17</b>		<b>E</b> (достатньо)
<b>10,5 – 14</b>	не зараховано з можливістю повторного складання	<b>FX</b>
<b>0 – 10,4</b>	не зараховано з обов'язковим повторним курсом	<b>F</b>

### Шкала оцінювання результатів навчальних досягнень при вивченні дисципліни

Кількість балів за 100 бальною шкалою (max- 100 балів)	Підсумкова оцінка за національною шкалою (max- 5 балів)	Підсумкова оцінка за шкалою ECTS
90-100	зараховано	A
80-89	зараховано	B

70-79	зараховано	C
60-69	зараховано	D
50-59	зараховано	E
35-49	не зараховано з можливістю повторного складання	FX
1-34	не зараховано з обов'язковим повторним курсом	F

### Розподіл балів, які отримують студенти

Поточне оцінювання (аудиторна та самостійна робота)															ІНДЗ	Кількість балів (залік)	Загальна кількість балів
Змістовний модуль 1							Змістовний модуль 2								-	30	100
T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15			
2	5	4	4	6	4	8	4	4	4	5	5	5	4	6			
33							37										

\*T1, T2 ... T15 – теми змістових модулів.

## 8. Рекомендована література

### 8.1. Базова (основна)

1. Mark J. Price. C# 11 and .NET 7 – Modern Cross-Platform Development Fundamentals: Start building websites and services with ASP.NET Core 7, Blazor, and EF Core 7 / Mark J. Price. – Birmingham, United Kingdom: Packt Publishing Limited. – 818 p.
2. Mark J. Price. Apps and Services with .NET 7: Build practical projects with Blazor, .NET MAUI, gRPC, GraphQL, and other enterprise technologies / Mark J. Price. – Birmingham, United Kingdom: Packt Publishing. – 814 p.
3. Andrew Troelsen. Pro C# 10 with .NET 6: Foundational Principles and Practices in Programming / Andrew Troelsen, Phil Japikse. – Berkley, United States: Apress, 2022. – 1705 p.
4. Ian Griffiths. Programming C# 10: Build Cloud, Web, and Desktop Applications / Ian Griffiths, 2022. – 833 p.
5. Edmund Krogman. Learning Oop With C#: Tutorial And Explains Deep Core Concepts Of Oop / Edmund Krogman, 2022. – 408 p.
6. Raihan Taher. Hands-On Object-Oriented Programming with C#: Build maintainable software with reusable code using C# / Raihan Taher., 2019. – 290 p.
7. Troelsen A. Pro C# 8 with .NET Core 3 Foundational Principles and Practices in Programming / A. Troelsen, J. Japikse. – Berkley, United States: Apress, 2020. – 1160 p.
8. Freeman A. Pro ASP.NET Core 3 (Develop Cloud-Ready Web Applications Using MVC 3, Blazor, and Razor Pages) / Adam Freeman., 2020. – 1400 c.
9. Bipin J. Beginning Database Programming Using ASP.NET Core 3: With MVC, Razor Pages, Web API, jQuery, Angular, SQL Server, and NoSQL / Joshi Bipin. – Berkley, United States: Apress, 2019. – 481 c.
10. Baptista G. Hands-On Software Architecture with C# 8 and .NET Core 3: Architecting software solutions using microservices, DevOps, and design patterns for Azure Cloud / Gabriel Baptista. – Birmingham, UK: Packt Publishing, 2019. – 598 c.

### 8.2. Допоміжна

1. **Томка**, О. Burchinska, V. Dvorzhak, E. Vatamanitsa, A. Dovgun Development of an information and exchange service for advertising promotion using .NET platform and the C#, Python // Strategic business analysis in cross-platform decision support systems. Monography. / Yuriy Ushenko, Serhiy Ostapov, Serhiy Golub.(eds), LAP LAMBERT Academic Publishing, 2023. р. 74-136. ISBN: 978-620-5-64024-1. <https://drive.google.com/file/d/16zkL7sqNbjTR7TjtjvMWOvhCLfDf02y/view?usp=sharing>
2. Збірник тестових завдань з дисциплін «Об'єктно-орієнтоване програмування» та «Розробка додатків на платформі .NET». Частина 1. / **Ю.Я. Томка**. – Чернівці: Технодрук, 2022. – 264с. <https://archer.chnu.edu.ua/xmlui/handle/123456789/6706>
3. Збірник тестових завдань з дисциплін «Об'єктно-орієнтоване програмування» та «Розробка додатків на платформі .NET». Частина 2. / **Ю.Я. Томка**. – Чернівці: Технодрук, 2022. – 184с. <https://archer.chnu.edu.ua/xmlui/handle/123456789/6707>
4. **Томка Ю.Я.** Об'єктно-орієнтоване програмування: навч. посіб. Чернівці: Чернівецький нац. ун-т ім. Ю.Федьковича, 2022. 502с. <https://archer.chnu.edu.ua/xmlui/handle/123456789/6377>
5. Windows Presentation Foundation: практичні рецепти / **Ю.Я. Томка**, О.М. Яцько – Чернівці: Технодрук, 2023. – 368с. <https://archer.chnu.edu.ua/xmlui/handle/123456789/6705>
6. **Томка Ю.Я.**, Ушенко Ю.О. Основи ASP.NET MVC 5: навчальний посібник. Чернівці, Чернівецький нац. ун-т, 2018. 730 с. <https://archer.chnu.edu.ua/xmlui/handle/123456789/6381>
7. Senthilvel G. Enterprise Application Architecture with .NET Core: An architectural journey into the Microsoft .NET open source platform / G. Senthilvel, O. Khan, H. Qureshi. – Birmingham, UK: Packt Publishing, 2017. – 566 с.
8. C. Martin R. Clean Architecture: A Craftsman's Guide to Software Structure and Design, First Edition / Robert C. Martin. – London, England: Pearson, 2017. – 428 с.
9. Gammelgaard C. Microservices in .NET Core: with examples in Nancy / Christian Horsdal Gammelgaard. – Shelter Island, New York: Manning Publications, 2017. – 344 с.
10. Mohapatra B. Microservices by examples using .NET Core: A book with lot of practical and architectural styles for Microservices using .NET Core / B. Mohapatra, B. Banerjee, G. Arora. – India: BPB PUBLICATIONS, 2019. – 239 с.
11. Himschoot P. Blazor Revealed: Building Web Applications in .NET / Peter Himschoot. – Berkley, United States: Apress, 2019. – 247 с.
12. Litvinavicius T. Exploring Blazor Creating Hosted, Server-side, and Client-side Applications with C# / Taurius Litvinavicius. – Berkley, United States: Apress, 2019. – 199 с.
13. Hewage M. A Practical Guide to Azure DevOps: Learn by doing / Milindanath Hewage. – India: Independently published, 2020. – 185 с.

## 9. Інформаційні ресурси

1. ASP.NET documentation [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-3.1>.
2. Руководство по ASP.NET Core 3 [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://metanit.com/sharp/aspnet5/>
3. ASP.NET CORE [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://www.c-sharpcorner.com/technologies/Asp-Net-core>.
4. ASP.NET Core Series [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://code-maze.com/net-core-series/>.
5. ASP.NET Core in Action [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://livebook.manning.com/book/asp-net-core-in-action/appendix-b/1.6>
6. .NET Tutorials [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://dotnettutorials.net>
7. JASON WATMORE'S BLOG [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://jasonwatmore.com/>.

8. Clean Architecture with .NET Core: Getting Started [Электронный ресурс]. – 2022. – Режим доступа до ресурсу: <https://jasontaylor.dev/clean-architecture-getting-started/>.
9. .NET Microservices: Architecture for Containerized .NET Applications [Электронный ресурс]. – 2022. – Режим доступа до ресурсу: <https://docs.microsoft.com/en-us/dotnet/architecture/microservices/>.
10. Azure DevOps documentation [Электронный ресурс]. – 2022. – Режим доступа до ресурсу: <https://docs.microsoft.com/en-us/azure/devops/?view=azure-devops>.
11. Learn Entity Framework Core [Электронный ресурс]. – 2022. – Режим доступа до ресурсу: <https://www.learnentityframeworkcore.com/>.
12. Enterprise grade .NET & Cloud tutorials for every level [Электронный ресурс]. – 2020. – Режим доступа до ресурсу: <https://www.youtube.com/c/Elfocrash/playlists>.
13. fullstackmark.com [Электронный ресурс]. – 2022. – Режим доступа до ресурсу: <https://fullstackmark.com/>.
14. Dapper Tutorial [Электронный ресурс]. – 2022. – Режим доступа до ресурсу: <https://dapper-tutorial.net/dapper>
15. Material Design components for Blazor [Электронный ресурс]. – 2022. – Режим доступа до ресурсу: <https://www.matblazor.com/>.