



СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ «ТЕХНОЛОГІЇ .NET У СУЧАСНІЙ ВЕБ-ІНЖЕНЕРІЇ»

Компонента освітньої програми – вибіркова (4 кредити)

Освітньо-професійна програма	Інтелектуальний аналіз даних в комп'ютерних інформаційних системах
Спеціальність	F3 Комп'ютерні науки
Галузь знань	F Інформаційні технології
Рівень вищої освіти	перший (бакалаврський)
Мова навчання	українська
Профайл викладача (-ів)	Томка Юрій Ярославович – кандидат фізико-математичних наук, доцент кафедри комп'ютерних наук https://kkn.chnu.edu.ua/about-kafedra/spivrobitnyky/tomka_yurii/
Контактний тел.	+38095-769-21-91
E-mail:	y.tomka@chnu.edu.ua
Сторінка курсу в Moodle	https://moodle.chnu.edu.ua/course/view.php?id=3387
Консультації	понеділок 09.50 до 11.10

АНОТАЦІЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Метою викладання дисципліни «Технології .NET у сучасній веб-інженерії» є формування у здобувачів практичних навичок побудови надійних, масштабованих та безпечних веб-систем на платформі .NET – від модульного моноліту (Clean Architecture + DDD) до cloud-native мікросервісної системи на базі .NET Aspire, з розгортанням на Amazon EKS.

Основними завданнями курсу «Технології .NET у сучасній веб-інженерії» є: освоєння Clean Architecture та DDD у побудові модульного моноліту; вивчення патернів CQRS, MediatR, Domain Events, Outbox; опанування комунікацій між сервісами (gRPC, REST, RabbitMQ); аутентифікація через Keycloak та Duende IdentityServer; Observability (OpenTelemetry, Prometheus, Grafana); розгортання у Kubernetes на AWS (EKS, ECR, GitHub Actions).

НАВЧАЛЬНИЙ КОНТЕНТ ОСВІТНЬОЇ КОМПОНЕНТИ

Змістовий модуль 1. Модульний моноліт. Clean Architecture + DDD на платформі .NET	
Тема 1	.NET / ASP.NET Core. Еволюція архітектур: від класичного моноліту до cloud-native – Огляд .NET: runtime, SDK, CLR, performance improvements (AOT compilation) – C# : primary constructors, collection expressions, inline arrays, ref readonly parameters – ASP.NET Core pipeline: middleware, DI-контейнер, routing, minimal hosting model – Порівняння архітектур: класичний моноліт → модульний моноліт → мікросервіси – Еволюція архітектури як стратегія: DDD + Clean Architecture як фундамент виокремлення сервісів – Практика: scaffold .NET проєкту, налаштування middleware pipeline та DI

Тема 2	Clean Architecture + DDD. Структура модульного моноліту. Bounded Context. Shared Kernel – DDD стратегічний рівень: Domain, Subdomain, Bounded Context, Ubiquitous Language – Структура Clean Architecture: Domain → Application → Infrastructure → Presentation – Модульний моноліт: Product Module, Basket Module, Order Module (один процес, один деплой) – Shared Kernel: Domain Abstractions, Common Entities, Domain Events – спільні концепції – Внутрішні виклики між модулями: in-process (без мережі, без ґейтвею) – Практика: створення структури модульного моноліту з трьома Bounded Contexts
Тема 3	Entity Framework Core. Repository. Unit of Work. Domain Events. Outbox Pattern – EF Core Code First: DbContext, Entities, Fluent API конфігурація, Migrations, PostgreSQL – Generic Repository та Unit of Work: абстракція доступу до даних – Domain Events: оголошення подій в агрегатах, публікація через MediatR (in-process) – Outbox Pattern: надійна публікація подій через таблицю в БД – гарантована доставка – Транзакційна узгодженість: Unit of Work + Domain Events у межах однієї транзакції – Практика: Domain Events + Outbox Pattern для модуля замовлень
Тема 4	MediatR Pipeline. CQRS. Minimal APIs / Controllers. FluentValidation. AutoMapper. Scrutor – MediatR: IRequest, IRequestHandler, INotification, IPipelineBehavior (logging, validation, caching) – CQRS: Commands (WriteModel) та Queries (ReadModel) – повне розділення моделей – Minimal APIs: endpoint groups, filters, OpenAPI; порівняння з Controllers – FluentValidation: правила, кастомні валідатори, інтеграція через MediatR ValidationBehavior – AutoMapper: маппінг Entity ↔ DTO, Configuration Profiles, IQueryable Projections – Scrutor: assembly scanning, decorator pattern для DI-реєстрації – Практика: CQRS use case CreateOrder з ValidationBehavior, LoggingBehavior та AutoMapper
Тема 5	Аутентифікація та авторизація. Keycloak (SSO). Duende IdentityServer. OAuth 2.0 / OpenID Connect – OAuth 2.0 та OpenID Connect: flows (Authorization Code + PKCE, Client Credentials) – JWT-токени: структура Header.Payload.Signature, claims, підпис (RS256), валідація, refresh – Keycloak: realm, clients, roles, SSO, управління користувачами, PKCE налаштування – Duende IdentityServer: Authorization Server, видача JWT, scopes, consent, PKCE – Service-to-Service автентифікація: Client Credentials flow між внутрішніми сервісами – ASP.NET Core: Bearer token middleware, [Authorize], policy-based authorization – Практика: налаштування Keycloak + захист ASP.NET Core endpoints
Тема 6	Крос-різні можливості. Redis Cache. Polly. Health Checks. Validation. Pagination – Redis: IMemoryCache та IDistributedCache, StackExchange.Redis, cache-aside pattern – Polly v8: ResiliencePipeline, Retry (exponential backoff), CircuitBreaker, Timeout, Hedge – Health Checks: Readiness / Liveness проби, кастомні перевірки, Kubernetes інтеграція – Глобальний ExceptionHandler: IExceptionHandler, ProblemDetails (RFC 7807) – Idempotency: ідемпотентні ключі у запитах, зберігання результатів у Redis – Pagination / Filtering / Sorting: cursor-based та offset pagination через EF Core – Практика: Polly CircuitBreaker + Redis distributed cache у модулі кошика
Тема 7	gRPC у .NET. .proto контракти. Синхронні та стрімінгові виклики. Порівняння з REST – Protocol Buffers (.proto): синтаксис, scalar types, messages, services, code generation – gRPC типи: Unary, Server Streaming, Client Streaming, Bidirectional Streaming – gRPC-dotnet: Grpc.AspNetCore, HttpClientFactory, interceptors (logging, auth, retry) – gRPC для міжсервісної комунікації: синхронні виклики з ProductService до

	BasketService – Порівняння gRPC vs REST: продуктивність (HTTP/2), строга типізація, code-first контракти – gRPC-Web та HTTP Transcoding: RESTful фасад поверх gRPC для браузерних клієнтів – Практика: gRPC Discount сервіс + interceptor для логування та retry
Змістовий модуль 2. Мікросервісна архітектура при розробці Web-додатків на платформі ASP.NET Core	
Тема 8	Мікросервісна архітектура. Декомпозиція моноліту. .NET Aspire. Service Discovery – Принципи мікросервісів: single responsibility, autonomy, loose coupling, deploy independently – Декомпозиція: Strangler Fig Pattern, виокремлення Bounded Context як окремого сервісу – .NET Aspire: AppHost, ServiceDefaults, Components (Redis, PostgreSQL, RabbitMQ) – Service Discovery у .NET Aspire: автоматична реєстрація та знаходження сервісів – Окрема БД per service: PostgreSQL per service, Redis per service – .NET Aspire Dashboard: вбудований UI для Traces, Metrics та Logs у dev-середовищі – Практика: виокремлення Order Service як мікросервісу + .NET Aspire оркестрація
Тема 9	YARP API Gateway. Reverse Proxy. Маршрутизація. Балансування навантаження – YARP (Yet Another Reverse Proxy): Routes, Clusters, конфігурація через appsettings / code – Трансформації запитів/відповідей: заголовки, шляхи, query strings – Rate Limiting: FixedWindow, SlidingWindow, TokenBucket – захист від перевантаження – Load Balancing: RoundRobin, LeastRequests, PowerOfTwoChoices алгоритми – Bearer token forwarding через YARP до downstream сервісів – BFF (Backend for Frontend) та Aggregator патерни через YARP – Практика: YARP Gateway для Catalog/Basket/Order сервісів з rate limiting
Тема 10	Event-Driven Architecture. RabbitMQ. MassTransit. Pub/Sub. Outbox. Idempotency – Event-Driven Architecture: Event Bus, Choreography vs Orchestration – RabbitMQ: exchanges (Direct, Fanout, Topic), queues, bindings, dead-letter queues (DLX) – MassTransit: IConsumer, Publish vs Send, конфігурація, Saga (State Machine) – MassTransit Outbox: транзакційна публікація подій разом із збереженням стану в БД – Idempotency у consumer: обробка дублікатів (at-least-once), зберігання MessageId – Fan-out: OrderCreated event → BasketService (clear cart) + NotificationService – Практика: OrderPlaced event через RabbitMQ + MassTransit між Order та Basket сервісами
Тема 11	Захист мікросервісів. Keycloak SSO. Duende IdentityServer. OAuth 2.0. Service-to-Service – Keycloak як централізований Identity Provider для всіх мікросервісів системи – Authorization Code + PKCE flow: Client (Web/Mobile) → Keycloak → YARP → Services – Client Credentials flow: M2M (machine-to-machine) автентифікація між сервісами – Bearer token validation у кожному мікросервісі: JWT middleware, policy authorization – Duende IdentityServer: кастомний Authorization Server, scopes, claims transformation – API Resource захист: granular permissions, claims mapping, resource-based authorization – Практика: SSO через Keycloak для Catalog, Basket, Order і Discount сервісів
Тема 12	Redis у мікросервісах. Polly. Health Checks. Serilog. Глобальні фільтри та логування – Redis Per Service: окремий Redis instance на кожен сервіс, cache invalidation via events – Polly v8 у мікросервісах: CircuitBreaker між сервісами, Retry з exponential backoff – Health Checks: Readiness/Liveness/Startup Probe, інтеграція з Kubernetes probes – Serilog: structured logging, correlation ID, enrichers, sinks: Console, File, Seq – Global Exception Handling: IExceptionHandler, ProblemDetails (RFC 7807) у всіх сервісах – Практика: Polly + Serilog з correlation ID у всіх сервісах системи

Тема 13	Observability. OpenTelemetry. Prometheus. Grafana. Distributed Tracing – Три стовпи Observability: Logs (Loki/Serilog), Traces (Jaeger), Metrics (Prometheus) – OpenTelemetry у .NET: автоматична інструментація, ActivitySource, OTLP exporter – Distributed Tracing: кореляція запитів між сервісами, TraceId/SpanId propagation – Prometheus: scraping метрик з .NET (dotnet-runtime, ASP.NET Core, custom meters) – Grafana: Dashboards, Alerts, Data Sources – Prometheus, Loki, Tempo, Jaeger – .NET Aspire Dashboard: вбудований трасування та метрики для dev-розробки – Практика: OpenTelemetry + Prometheus + Grafana для повної мікросервісної системи
Тема 14	Docker. Kubernetes. CI/CD на AWS. Amazon EKS. Amazon ECR. Secrets Manager – Docker: multi-stage Dockerfile для .NET 8, оптимізація образу, .dockerignore – Kubernetes: Deployments, Services, ConfigMaps, Secrets, Ingress Controller – Amazon EKS: керований Kubernetes кластер, node groups, IAM Roles for Service Accounts – Amazon ECR: реєстр Docker образів, image scanning, lifecycle policies – CI/CD з GitHub Actions: build → unit tests → push to ECR → kubectl apply to EKS – AWS Secrets Manager та Parameter Store: управління секретами та конфігурацією сервісів – Практика: повний CI/CD pipeline GitHub Actions → Amazon ECR → Amazon EKS
Тема 15	Фінальний проєкт. Від модульного моноліту до мікросервісної системи. Повний CI/CD – Бізнес-система: Product Catalog + Basket + Order + Discount сервіси (реальна доменна область) – Крок 1 – Модульний моноліт: Clean Architecture + DDD, MediatR, EF Core + PostgreSQL, Keycloak – Крок 2 – Мікросервісна система: .NET Aspire, YARP Gateway, gRPC, RabbitMQ + MassTransit – Крос-різні: Redis (per service), Polly v8, Serilog + correlation ID, Health Checks – Observability: OpenTelemetry → Prometheus + Grafana, Distributed Tracing – Розгортання: Docker → Amazon EKS (AWS), CI/CD: GitHub Actions → ECR → EKS – Презентація: демонстрація живої системи + архітектурне обґрунтування прийнятих рішень

ОСВІТНІ ТЕХНОЛОГІЇ, ФОРМИ ТА МЕТОДИ МЕТОДИ НАВЧАННЯ

У процесі вивчення навчальної дисципліни використовуються інноваційні освітні технології: інформаційно-комунікаційні, технології студентоцентрованого навчання; проєктна діяльність; традиційні та інтерактивні форми і методи навчання, серед яких: лекція-візуалізація, проблемна лекція, семінар-дискусія, семінар-діалог, самостійно-дослідницька робота, аналіз і рішення ситуативних професійних психолого-педагогічних задач (Case study) та ін.

ФОРМИ Й МЕТОДИ КОНТРОЛЮ ТА ОЦІНЮВАННЯ

Поточний контроль: усне та письмове опитування, тестування, демонстрація виконаних практичних робіт, захист наскрізного проєкту (модульний моноліт → мікросервіси → деплой на Amazon EKS).

Підсумковий контроль – залік.

КРИТЕРІЇ ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ

Оцінювання програмних результатів навчання здобувачів освіти здійснюється за шкалою європейської кредитно-трансферної системи (ECTS).

Критерієм успішного оцінювання є досягнення здобувачем вищої освіти мінімальних порогових рівнів (балів) за кожним запланованим результатом навчання

ПОЛІТИКА ЩОДО АКАДЕМІЧНОЇ ДОБРОЧЕСНОСТІ

Дотримання політики щодо академічної доброчесності учасниками освітнього процесу при вивченні навчальної дисципліни регламентовано такими документами:

- ✓ «Етичний кодекс Чернівецького національного університету імені Юрія Федьковича» <https://www.chnu.edu.ua/media/jxdfs0zb/etychnyi-kodeks-chernivets-koho-natsionalnoho-universytetu.pdf> ;
- ✓ «Положенням про виявлення та запобігання академічного плагіату у Чернівецькому національному університету імені Юрія Федьковича» <https://www.chnu.edu.ua/media/n5nbzwb/polozhennia-chnu-pro-plahi-at-2023plusdodatky-31102023.pdf> .

ІНФОРМАЦІЙНІ РЕСУРСИ

1. Price M.J. C# 12 and .NET 8 – Modern Cross-Platform Development Fundamentals. – Packt Publishing, 2023. – 823 p.
2. Price M.J. Apps and Services with .NET 8. – Packt Publishing, 2024. – 828 p.
3. Troelsen A., Japikse P. Pro C# 12 with .NET 8: Foundational Principles and Practices in Programming. – Apress, 2023. – 1712 p.
4. Microsoft .NET Architecture Guides [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://dotnet.microsoft.com/en-us/learn/dotnet/architecture-guides>.
5. .NET Aspire Documentation [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://learn.microsoft.com/en-us/dotnet/aspire/>.
6. Duende IdentityServer v7 Documentation [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://docs.duendesoftware.com/identityserver/v7>
7. Keycloak Documentation [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://www.keycloak.org/documentation>.
8. YARP (Yet Another Reverse Proxy) Documentation [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://microsoft.github.io/reverse-proxy/>.
9. .NET Microservices: Architecture for Containerized .NET Applications [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://docs.microsoft.com/en-us/dotnet/architecture/microservices/>.
10. MassTransit Documentation [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://masstransit.io/documentation/>.
11. Enterprise grade .NET & Cloud tutorials for every level [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://opentelemetry.io/docs/languages/dotnet/>.

Детальна інформація щодо вивчення курсу «Технології .NET у сучасній веб-інженерії» висвітлена у робочій програмі навчальної дисципліни: https://kkn.chnu.edu.ua/wp-content/uploads/2024/02/2024_vk_robpr_%D0%9F%D0%9F%D0%925_NET.pdf

