

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ЕКОНОМІКО-ПРАВНИЧИЙ ФАХОВИЙ КОЛЕДЖ
ЗАПОРІЗЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ»**

ЗАТВЕРДЖУЮ

Директор

_____ Олена ГРИБАНОВА

«____» _____ 2025

АРХІТЕКТУРА ВЕБДОДАТКІВ

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

підготовки фахових молодших бакалаврів

галузі знань: 12 інформаційні технології

спеціальності: 121 інженерія програмного забезпечення

освітньо-професійної програми: розробка програмного забезпечення

Укладач/Укладачі: Ірина ІВАНОВА

Обговорено та ухвалено
на засіданні циклової комісії

Протокол № _ від “__” _____ 20__

Голова ЦК _____

(підпис)

(ім'я, прізвище)

Ухвалено методичною радою коледжу

Протокол № _ від “__” _____ 20__

Заступник директора з НМР

Альона ХОДАКОВСЬКА

(підпис)

2025 рік

1. Опис навчальної дисципліни

Найменування показників	Галузь знань, спеціальність, освітньо-професійний ступінь	Характеристика навчальної дисципліни	
		денна форма здобуття освіти	
Кількість кредитів - 3	Галузь знань <i>12 – інформаційні технології</i>	Нормативна/вибіркова	
Розділів – 2	Спеціальність <i>121 – інженерія програмного забезпечення</i>	Рік підготовки: 4	
Загальна кількість годин - 90		7-й сем	8-й сем
		Лекції	
Тижневих аудиторних годин для денної форми здобуття освіти: – 3 год	Освітньо-професійна програма <i>розробка програмного забезпечення</i>	10 год.	год.
		Практичні, семінарські	
		20 год.	год.
		Лабораторні	
	Освітньо-професійний ступінь: <i>фаховий молодший бакалавр</i>	год.	год.
		Самостійна робота	
		60 год.	год.
		Вид контролю:	
залік	-		

2. Мета та завдання навчальної дисципліни

Метою викладання навчальної дисципліни «Бази даних» є оволодіння основними знаннями та набуття навичок та вмінь у галузі проектування баз даних та розробки інформаційних систем, ознайомлення із сучасними поглядами на деякі розділи теорії проектування баз даних, засвоєння головних принципів проектування баз даних, формування навичок та вмінь з проектування баз даних та розробки інформаційних систем.

Основними завданнями вивчення дисципліни є: ознайомлення із сучасними поглядами на методологію проектування баз даних, інформаційних систем; визначення та вирішення основних проблем, які постають при розробці систем управління базами даних (СУБД); вивчення теоретичних засад, що лежать у основі сучасних СУБД.

У разі успішного завершення курсу здобувач освіти набуває **програмні компетентності**:

ЗК3. Здатність спілкуватися державною мовою як усно, так і письмово.

ЗК5. Знання та розуміння предметної області та розуміння професійної діяльності.

ЗК6. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

ЗК7. Здатність застосовувати знання у практичних ситуаціях.

ЗК8. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК9. Здатність ефективно управляти часом та інформацією.

СК1. Здатність алгоритмічно та логічно мислити.

СК4. Здатність дотримуватися стандартів при розробці програмного забезпечення.

СК5. Здатність брати участь у визначенні та формулюванні вимог до програмного забезпечення.

СК6. Здатність брати участь у проектуванні програмного забезпечення.

СК7. Здатність розробляти модулі і компоненти програмного забезпечення за допомогою типових алгоритмів та інструментів.

СК10. Здатність реалізовувати всі етапи життєвого циклу програмного забезпечення.

СК12. Здатність розробляти архітектуру, модулі та компоненти програмного забезпечення.

Програмні результати навчання, які досягаються за допомогою вивчення дисципліни:

РН5. Розробляти та супроводжувати програмне забезпечення.

РН6. Використовувати основні методології та підходи до організації життєвого циклу програмного забезпечення.

PH7. Застосовувати стандарти, специфікації в процесах життєвого циклу програмного забезпечення.

PH8. Аналізувати вимоги до програмного забезпечення.

PH11. Обирати інструментальні засоби, ефективні методи та здійснювати тестування програмних систем.

PH21. Знати основні типи архітектури програмного забезпечення, стандартні архітектурні рішення (патерни проектування) та вміти застосовувати їх під час проектування архітектури складних багатокомпонентних програмних систем.

PH24. Проектувати компоненти архітектурного рішення, використовуючи відповідні програмні засоби.

Міждисциплінарні зв'язки: матеріал дисципліни ґрунтується на знаннях отриманих студентами при вивченні курсів «Алгоритми і структури даних», «Архітектура та проектування програмного забезпечення».

3. Програма навчальної дисципліни

Розділ 1. Основи веб-архітектури та клієнт-серверної взаємодії

Тема 1. Вступ і еволюція вебу. Типи вебзастосунків (SPA/MPA/PWA).

Поняття вебзастосунку; статичний → динамічний веб; SPA, MPA, PWA — визначення, переваги/обмеження; офлайн-можливості (service worker, manifest); роль CDN і кешування.

Тема 2. Клієнт-серверна модель та HTTP/HTTPS.

URL/URI; запит/відповідь; методи (GET/POST/PUT/PATCH/DELETE) та ідемпотентність; коди 1xx–5xx; заголовки (Accept, Content-Type, Authorization, Cache-Control, ETag); CORS і preflight; TLS/HTTPS.

Тема 3. Front-end архітектура.

HTML/CSS/JS ролі; DOM, CSSOM, критичний шлях рендерингу; компоненти, стан (useState, контекст, reducer), маршрутизація; форми та валідація; доступність (ARIA, ролі, фокус-менеджмент).

Тема 4. Front-end архітектура: від DOM до фреймворків.

HTML/CSS/JS — ролі; DOM, CSSOM, критичний шлях рендерингу; стан і події; огляд сучасних фреймворків (React/Angular/Vue) та шаблонів (контейнер/презентаційні, контекст, reducer); маршрутизація; керування формами та валідацією; доступність (ARIA, ролі, фокус-менеджмент).

Тема 5. Архітектурні патерни веб-систем.

моноліт, модульний моноліт, SOA, мікросервіси, BFF (Backend-for-Frontend); коли який підхід доречний; компроміси команди/продуктивності/складності.

Розділ 2. Дані на клієнті й сервері, безпека, продуктивність і якість

Тема 6. Безпека вебдодатків.

Принципи (найменші привілеї, валідація/санітизація); XSS (reflected/stored/DOM) і контекстно-безпечний рендер; CSRF (SameSite/токени); заголовки безпеки (CSP, HSTS, X-Content-Type-Options); керування сесіями/JWT.

Тема 7. Дані та файли у вебзастосунках.

LocalStorage/SessionStorage/IndexedDB — сфери застосування; Cache API; робота з файлами у браузері (Blob, object URL, прев'ю); завантаження на сервер (multipart/form-data), перевірка MIME, зберігання у /uploads; JSON/CSV як «постійні» дані без БД.

Тема 8. Продуктивність (Core Web Vitals) та оптимізація.

LCP/INP/CLS — цілі та діагностика; оптимізація зображень (AVIF/WebP, srcset/sizes, lazy); шрифти (WOFF2, font-display, preload); JS (code-splitting, defer/async); CSS (критичний CSS); кеш і пріоритети (preload/prefetch/preconnect); Lighthouse/RUM.

Тема 9. Робота з API: REST/GraphQL, інструменти та документація.

Ресурси, колекції, фільтри, пагінація (offset/cursor), стабільні помилки; ідеї GraphQL, коли доречно; OpenAPI/Swagger; Postman/Insomnia (колекції, тести); версіонування й політика депрекейту.

Тема 10. Тестування вебдодатків.

Піраміда тестування; юніт/компонентні (Jest/Vitest, Testing Library); інтеграційні API без БД (Supertest із тимчасовими файлами); E2E (Playwright/Cypress), стабільні локатори; базові перевірки продуктивності (Lighthouse) та доступності (axe).

4. Структура навчальної дисципліни

Назви тематичних розділів і тем	Кількість годин				
	денна форма				
	усього	у тому числі			
		л	с/п	лаб.	сам.роб. інд.завд.
1	2	3	4	5	6
Розділ 1. Основи веб-архітектури та клієнт-серверної взаємодії					
Тема 1. Вступ і еволюція вебу. Типи вебзастосунків (SPA/MPA/PWA)		1	2		6
Тема 2. Клієнт–серверна модель та HTTP/HTTPS		1	2		6
Тема 3. Front-end архітектура		1	2		6
Тема 4. Back-end на JavaScript		1	2		6
Тема 5. Архітектурні патерни веб-систем		1	2		6
Разом за розділом 1		5	10		30
Розділ 2. Дані на клієнті й сервері, безпека, продуктивність і якість					
Тема 6. Безпека вебдодатків		1	2		6
Тема 7. Дані та файли у вебзастосунках		1	2		6
Тема 8. Продуктивність (Core Web Vitals) та оптимізація		1	2		6
Тема 9. Робота з API: REST/GraphQL, інструменти та документація		1	2		6
Тема 10. Тестування вебдодатків		1	2		6
Разом за розділом 2		5	10		30
<i>Усього годин</i>		10	20		60

5. Теми лекційних занять

№ з/п	Назва теми	Кількість годин
1	Вступ до вебархітектури	1
2	Клієнт-серверна архітектура	1
3	Front-end архітектура	1
4	Back-end архітектура	1
5	Архітектурні патерни	1
6	Безпека вебдодатків	1
7	Робота з файлами та зберігання даних у вебдодатках	1
8	Оптимізація продуктивності вебдодатків	1
9	Робота з API	1
10	Основи тестування вебдодатків	1
Разом		10

6. Теми семінарських занять (практичних/лабораторних)

№ з/п	Назва теми	Кількість годин
1	React з нуля: компоненти, JSX, пропси й стан	2
2	Події, форми, «підйом» стану і композиція	2
3	Підключаємо бекенд: fetch, життєвий цикл, CORS, файли	2
4	Роутинг і макети (layout)	2
5	Глобальний стан без «важких» бібліотек: Context + useReducer	2
6	Форми з валідацією (клієнт і сервер) + запис у CSV/JSONL	2
7	Списки, пагінація, пошук, віртуалізація (із файлу)	2
8	Аутентифікація та захищені маршрути (JWT у HttpOnly-cookie, users.json)	2
9	Завантаження файлів: прев'ю, прогрес, збереження в /uploads	2
10	Редагування/видалення «каталогу» без БД: оновлюємо courses.json	2
Разом		20

7. Самостійна робота

№ з/п	Назва теми	Кількість годин
1	React з нуля: компоненти, JSX, пропси й стан	6
2	Події, форми, «підйом» стану і композиція	6
3	Підключаємо бекенд: fetch, життєвий цикл, CORS, файли	6
4	Роутинг і макети (layout)	6
5	Глобальний стан без «важких» бібліотек: Context + useReducer	6
6	Форми з валідацією (клієнт і сервер) + запис у CSV/JSONL	6
7	Списки, пагінація, пошук, віртуалізація (із файлу)	6
8	Аутентифікація та захищені маршрути (JWT у HttpOnly-cookie, users.json)	6
9	Завантаження файлів: прев'ю, прогрес, збереження в /uploads	6
10	Редагування/видалення «каталогу» без БД: оновлюємо courses.json	6
Разом		60

Індивідуальні завдання

Створити повноцінний навчальний вебзастосунок у зв'язці React (Vite) + Express з клієнт-серверною взаємодією, базовими сценаріями роботи з даними, елементами безпеки, продуктивності та тестування — без використання БД (персистентність через файли JSON/CSV/JSONL на бекенді або LocalStorage на фронтенді).

Надати єдиний документ (PDF/Doc/Markdown) — «Паспорт проєкту», що деталізує ваш варіант і слугує специфікацією:

1. Назва та короткий опис обраної предметної області (до 150 слів).
2. Користувацькі сценарії (3–5 історій: «як користувач...»).
3. Архітектурна схема (спрощена: клієнт ↔ сервер ↔ файлове сховище).
4. Опис API (ендпоїнти: метод, шлях, параметри, приклади запиту/відповіді).

- коди помилок).
5. Модель даних (структури JSON/CSV/JSONL, поля, типи, валідація).
 6. Вимоги до якості (що саме реалізовано з безпеки/продуктивності/доступності/тестів).
 7. Інструкція запуску (кроки, порти, змінні середовища, тестові користувачі/дані).
 8. Скріншоти/скрінкаст ключових сценаріїв, 2–5 хв відео (за можливості).

8. Види контролю і система накопичення балів

Поточний контроль знань (7 семестр)		Самостійна робота інд. завд.	Залік	Сума
Атестація № 1 (30 балів)	Атестація № 2 (30 балів)			
Розділ 1	Розділ 2			
30	30	20	20	100

Шкала оцінювання: національна та ECTS

<i>12-бальна шкала</i>	<i>30-бальна шкала</i>	<i>100-бальна шкала</i>	<i>Шкала ECTS</i>	<i>4-бальна шкала</i>	<i>Національна шкала екзамен/залік*</i>
1	0-3	0-34	FX	2	Незадовільно
2	4-6	35-49			
3	7-9	50-59	F	3	Задовільно
4	10-12	60-64	E		
5	13-15	65-69	D		
6	16-18	70-74			
7	19-21	75-79	C	4	Добре
8	22-24	80-84			
9	25-27	85-89	B	5	Відмінно
10	28	90-93	A		
11	29	94-97			
12	30	98-100			

*Оцінювання заліку: зараховано/незараховано

9. Рекомендована література

Основна:

1. Banks A., Porcello E. Learning React: Modern Patterns for Developing React Apps. O'Reilly Media, 2020. 307 с.
2. Casciaro M., Mammino L. Node.js Design Patterns. Packt, 2020. 664 с.
3. Geewax J. API Design Patterns. Manning, 2021. 480 с.
4. Newman S. Building Microservices (2nd ed.). O'Reilly Media, 2021. 612 с.
5. Gourley D., Totty B.; Sayer M., Reddy S., Aggarwal A. HTTP: The Definitive Guide. O'Reilly Media, 2002. 656 с.
6. Wagner J. Web Performance in Action: Building Faster Web Pages. Manning, 2017. 376 с.

Інформаційні ресурси

1. Курс «Архітектура вебдодатків». URL : <https://epkmoodle.znu.edu.ua/course/view.php?id=1280>.

