



Архітектури та проектування програмних систем штучного інтелекту

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Другий (магістерський)</i>
Галузь знань	<i>F Інформаційні технології</i>
Спеціальність	<i>F2 Інженерія програмного забезпечення</i>
Освітня програма	<i>Інженерія програмного забезпечення мультимедійних систем та смарттехнологій Інженерія програмного забезпечення іммерсивних систем та платформ цифрових двійників</i>
Статус дисципліни	<i>Вибіркова</i>
Форма навчання	<i>Очна (денна)</i>
Рік підготовки, семестр	<i>1 рік підготовки, 2 семестр</i>
Обсяг дисципліни	<i>Лекції: 32 год., лабораторні заняття: 14 год., самостійна робота: 104 год.</i>
Семестровий контроль/ контрольні заходи	<i>Екзамен, модульна контрольна робота, календарний контроль</i>
Розклад занять	<i>Згідно розкладу на весняний семестр поточного навчального року (schedule.kpi.ua)</i>
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	<i>Лектор: аспірант, Лісняк Олександр Михайлович, lisnyak.om1010a@gmail.com Лабораторні заняття: аспірант, Лісняк Олександр Михайлович, lisnyak.om1010a@gmail.com</i>
Розміщення курсу	<i>Google Classroom. Доступ надається зареєстрованим студентам.</i>

Програма навчальної дисципліни

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

Вивчення дисципліни «Архітектури та проектування програмних систем штучного інтелекту» спрямоване на формування у здобувачів освіти системного бачення архітектури програмних систем, що використовують компоненти штучного інтелекту, та практичних навичок їх проектування, реалізації, версіонування, контейнеризації й супроводу. Протягом семестру здобувачі послідовно проектують і розвивають власну програмну систему, яка вирішує практичну задачу та містить інтегрований компонент штучного інтелекту.

Метою вивчення дисципліни «Архітектури та проектування програмних систем штучного інтелекту» є формування у здобувачів здатності обґрунтовано проектувати архітектуру програмних систем штучного інтелекту, інтегрувати інтелектуальні компоненти з використанням локальних моделей, власних сервісів або зовнішніх API, організовувати версіонування та відтворюваність програмної системи, застосовувати контейнеризацію й автоматизацію процесу розроблення, а також розуміти базові підходи до моніторингу та оптимізації ШІ-компонентів.

Предметом дисципліни «Архітектури та проектування програмних систем штучного інтелекту» є архітектурні рішення та інженерні практики побудови програмних систем із компонентами штучного інтелекту: декомпозиція системи, способи інтеграції ШІ, програмні інтерфейси взаємодії, надійність, версіонування, відтворюваність, контейнеризація, автоматизація, моніторинг та локальна оптимізація виконання моделей.

Вивчення дисципліни «Архітектури та проектування програмних систем штучного інтелекту» сприяє формуванню у здобувачів освіти **фахових компетентностей**, необхідних для розв'язання практичних задач професійної діяльності, пов'язаних із проектуванням програмних систем штучного інтелекту:

- Здатність обґрунтовано обирати архітектурні шаблони та шаблони інтеграції ШІ-моделей під час проектування мікросервісного програмного забезпечення.
- Здатність впроваджувати процеси автоматизації, локальної контейнеризації ШІ-компонентів та оркестрації взаємодії між ними на комп'ютері розробника.
- Здатність керувати життєвим циклом та версіями моделей машинного навчання за методологією MLOps з використанням локальних open-source інструментів.
- Здатність застосовувати методи профілювання, оптимізації виведення та моніторингу якості моделей для забезпечення стабільної й безпечної роботи програмних систем ШІ.

Вивчення дисципліни «Архітектури та проектування програмних систем штучного інтелекту» сприяє формуванню у студентів наступних **програмних результатів навчання (ПРН)**:

- Розробляти і оцінювати стратегії проектування програмних засобів; обґрунтовувати, аналізувати і оцінювати варіанти проектних рішень з точки зору якості кінцевого програмного продукту, ресурсних обмежень та інших факторів.
- Розробляти і модифікувати архітектуру програмного забезпечення для реалізації вимог замовника.
- Проектувати архітектуру ШІ-застосунків та інтегрувати інтелектуальні компоненти у прикладні програмні системи з урахуванням вимог до модульності, взаємодії сервісів та ресурсних обмежень.
- Організовувати версіонування даних і моделей та відстеження експериментів, метрик і артефактів у процесах розроблення систем машинного навчання.
- Контейнеризувати, розгортати та оптимізовувати виконання моделей і сервісів штучного інтелекту в локальних та розподілених програмних середовищах.
- Здійснювати моніторинг, оцінювання стійкості та виявлення деградації моделей штучного інтелекту, аналізувати зміни характеристик даних і якості моделей.

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

Для успішного вивчення дисципліни необхідні знання з навчальних дисциплін першого (бакалаврського) рівня освіти спеціальності F2 Інженерія програмного забезпечення та досвід командного розроблення програмного забезпечення. Володіння конкретною мовою програмування, фреймворком або постачальником ШІ-сервісу не вимагається: технологічний стек здобувач обирає самостійно відповідно до потреб власного проекту.

Отримані при засвоєнні дисципліни «Архітектури та проектування програмних систем штучного інтелекту» теоретичні знання та практичні уміння забезпечують успішне виконання магістерських дисертацій за спеціальністю F2 Інженерія програмного забезпечення.

3. Зміст навчальної дисципліни

Дисципліна «Архітектури та проектування програмних систем штучного інтелекту» передбачає вивчення таких тем:

Тема 1. Основи архітектури та проектування програмних систем штучного інтелекту.

Тема 2. Інтеграція та реалізація ШІ-компонентів у прикладних програмних системах.

Тема 3. Версіонування, відтворюваність та керування змінами ШІ-систем.

Тема 4. Контейнеризація та автоматизація процесу розроблення.

Модульна контрольна робота

Екзамен

4. Навчальні матеріали та ресурси

Базова література:

1. Горнійчук І. В., Микитюк А. В., Оніщенко В. О. Сучасні технології програмування [Електронний ресурс] : навч. посіб. для здобувачів ступеня магістра. – Київ : ІСЗІ КПІ ім. Ігоря Сікорського, 2026. – 405 с.
2. Суліма С. В., Скулиш М. А., Курдеча В. В. Технології розробки web-орієнтованих систем [Електронний ресурс] : навч. посіб. – Київ : КПІ ім. Ігоря Сікорського, 2025. – 47 с.
3. Навчальні матеріали кафедри ПЗКС та електронний конспект лекцій з дисципліни «Архітектури та проектування програмних систем штучного інтелекту».

Додаткова література:

1. Huyen C. *Designing Machine Learning Systems: An Iterative Process for Production-Ready Applications*. – O'Reilly Media, 2022.
2. Gift N., Deza A. *Practical MLOps: Operationalizing Machine Learning Models*. – O'Reilly Media, 2021.
3. Офіційна документація Docker, Docker Compose, GitHub Actions / GitLab CI, DVC, MLflow, FastAPI та ONNX Runtime (актуальні версії на момент вивчення дисципліни).

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

№ з/п	Тип навчального заняття	Опис навчального заняття
Тема 1. Основи архітектури та проектування програмних систем штучного інтелекту		
1	Лекція 1. Вступ. Програмна система зі штучним інтелектом	ШІ-модель як компонент програмної системи. Типові складові ШІ-застосунку та їх відповідальність. Життєвий цикл системи. Концепція наскрізного студентського проекту та критерії достатнього обсягу. Завдання на СРС: п. 6, № 1.
2	Лекція 2. Базові принципи архітектури програмних систем	Компоненти та межі відповідальності. Модульність, зв'язність і залежності. Моноліт, модульний моноліт і сервісний підхід. Атрибути якості та компроміси архітектурних рішень. Завдання на СРС: п. 6, № 2.
3	Лекція 3. Архітектурні шаблони інтеграції ШІ-компонентів	Embedded model, Model-as-a-Service та зовнішній AI API. Локальні й хмарні моделі. Порівняння підходів за складністю, вартістю, затримкою, контролем

		<i>і масштабованістю.</i> <i>Завдання на СРС: п. 6, № 3.</i>
4	<i>Лекція 4. Проектування взаємодії компонентів та API</i>	<i>Контракти між компонентами, REST API, request/response, DTO, валідація та обробка помилок. Синхронна й асинхронна взаємодія на оглядовому рівні. FastAPI як один із прикладів реалізації сервісного інтерфейсу.</i> <i>Завдання на СРС: п. 6, № 4.</i>
5	<i>Лабораторне заняття 1.</i> <i>Лабораторна робота 1.</i> <i>Проектування архітектури власної ШІ-системи</i>	<i>Вибір практичної задачі та концепції наскрізного проекту. Побудова архітектурної схеми, визначення компонентів, їх відповідальності, способу інтеграції ШІ та основних інтерфейсів взаємодії. Код не є обов'язковим.</i> <i>Завдання на СРС: п. 6, № 16.</i>
<i>Тема 2. Інтеграція, реалізація та надійність ШІ-компонентів</i>		
6	<i>Лекція 5. Від архітектури до реалізації</i>	<i>Перетворення архітектурної схеми на структуру програмного коду. Розділення business logic, AI integration та інфраструктурного коду. Конфігурація та керування залежностями. Типові помилки під час реалізації.</i> <i>Завдання на СРС: п. 6, № 5.</i>
7	<i>Лекція 6. Інтеграція зовнішніх ШІ-сервісів</i>	<i>Робота із зовнішніми AI API та SDK. API keys і secrets, конфігурація, timeouts, retries, rate limits та обробка помилок. Залежність від постачальника та способи її зменшення.</i> <i>Завдання на СРС: п. 6, № 6.</i>
8	<i>Лабораторне заняття 2.</i> <i>Лабораторна робота 2. Реалізація системи (частина 1)</i>	<i>Початок реалізації спроектованої системи у довільному технологічному стеку. Реалізація основних компонентів та інтеграції з обраним ШІ-компонентом.</i> <i>Завдання на СРС: п. 6, № 17.</i>
9	<i>Лекція 7. Інтеграція локальних і власних моделей</i>	<i>Вбудовування моделі у застосунок та винесення моделі в окремий сервіс. Межі відповідальності inference-компонента. Ресурсні обмеження, залежності та особливості локального виконання.</i> <i>Завдання на СРС: п. 6, № 7.</i>
10	<i>Лекція 8. Надійність і якість ШІ-систем</i>	<i>Валідація, fallback, caching, timeouts, retries та rate limiting. Некоректні й нестабільні відповіді ШІ. Логування, базові health checks і graceful degradation.</i> <i>Завдання на СРС: п. 6, № 8.</i>
11	<i>Лабораторне заняття 3.</i> <i>Лабораторна робота 2. Реалізація системи (частина 2)</i>	<i>Завершення працездатної версії системи, обробка основних помилок і демонстрація наскрізного сценарію використання. Допускається зміна початкової архітектури з обґрунтуванням причин.</i> <i>Завдання на СРС: п. 6, № 18.</i>
<i>Тема 3. Версіонування, відтворюваність та керування змінами ШІ-систем</i>		
12	<i>Лекція 9. Версіонування у програмних системах зі ШІ</i>	<i>Версіонування коду, конфігурацій, API-контрактів, prompts, моделей і даних. Зв'язок версій</i>

		компонентів. Проблема відтворення попереднього стану системи. Завдання на СРС: п. 6, № 9.
13	Лекція 10. Версіонування даних і моделей	Підходи до зберігання великих артефактів та наборів даних. DVC як приклад інструмента версіонування даних і моделей. Коли спеціалізоване версіонування потрібне, а коли достатньо простіших засобів. Завдання на СРС: п. 6, № 10.
14	Лабораторне заняття 4. Лабораторна робота 3. Версіонування та відтворюваність (частина 1)	Організація версіонування наскрізного проєкту. Визначення артефактів, що потребують контролю версій: код, конфігурації, prompts, дані, моделі або інші складові відповідно до архітектури проєкту. Завдання на СРС: п. 6, № 19.
15	Лекція 11. Експерименти та відтворюваність	Параметри, метрики та артефакти експериментів. Experiment tracking та відтворення результатів. MLflow як приклад інструмента. Практики фіксації змін у системах, що використовують зовнішні AI API. Завдання на СРС: п. 6, № 11.
16	Лабораторне заняття 5. Лабораторна робота 3. Версіонування та відтворюваність (частина 2)	Перевірка можливості відтворити попередній стан системи або її ключових ШІ-артефактів. Документування обраного підходу та демонстрація переходу між версіями. Завдання на СРС: п. 6, № 20.
Тема 4. Контейнеризація, автоматизація та експлуатація програмних систем штучного інтелекту		
17	Лекція 12. Контейнеризація програмних систем	Призначення контейнеризації. Docker image, container, Dockerfile, ports, volumes, environment variables та secrets. Контейнеризація застосунку та ШІ-компонентів. Завдання на СРС: п. 6, № 12.
18	Лекція 13. Багатокомпонентні системи з Docker Compose	Організація локального середовища з декількох компонентів. Мережі, залежності, конфігурація та запуск системи однією командою. Docker Compose як інструмент локальної оркестрації. Завдання на СРС: п. 6, № 13.
19	Лабораторне заняття 6. Лабораторна робота 4. Контейнеризація та автоматизація (частина 1)	Контейнеризація наскрізного проєкту за допомогою Docker. Для багатокомпонентних систем – організація спільного запуску компонентів. Перевірка запуску системи у відтворюваному середовищі. Завдання на СРС: п. 6, № 21.
20	Лекція 14. Автоматизація процесу розроблення	Базові принципи CI/CD. Автоматизація перевірок, тестів і складання. GitHub Actions та GitLab CI як приклади. Git hooks і pre-commit як локальні засоби автоматизації. Завдання на СРС: п. 6, № 14.
21	Лекція 15. Експлуатація, моніторинг та локальна оптимізація ШІ-систем	Health checks, логи, latency та використання ресурсів. Data drift і model degradation на оглядовому рівні. ONNX і квантування як приклади

		локальної оптимізації виконання моделей. Межі застосовності цих підходів для систем із зовнішніми AI API. Завдання на СРС: п. 6, № 15.
22	Лабораторне заняття 7. Лабораторна робота 4. Контейнеризація та автоматизація (частина 2)	Додавання щонайменше одного автоматизованого кроку процесу розроблення: перевірка, тестування, складання, контейнеризація або інший погоджений сценарій за допомогою CI pipeline, Git hook чи іншого інструмента. Демонстрація завершеної системи. Завдання на СРС: п. 6, № 22.
Модульна контрольна робота		

6. Самостійна робота студента/аспіранта

Самостійна робота спрямована на підготовку до лекційних і лабораторних занять, послідовне доопрацювання власного наскрізного проєкту, опрацювання документації обраного технологічного стеку, підготовку до модульної контрольної роботи та екзамену. Загальний обсяг самостійної роботи – 104 години.

№ з/п	Назва теми, що виноситься на самостійне опрацювання	Кількість годин	Література
1	Підготовка до лекції 1	4	Базова та додаткова література, матеріали лекції
2	Підготовка до лекції 2	4	Базова та додаткова література, матеріали лекції
3	Підготовка до лекції 3	4	Базова та додаткова література, матеріали лекції
4	Підготовка до лекції 4	4	Базова та додаткова література, матеріали лекції
5	Підготовка до лекції 5	4	Базова та додаткова література, матеріали лекції
6	Підготовка до лекції 6	4	Базова та додаткова література, матеріали лекції
7	Підготовка до лекції 7	4	Базова та додаткова

			<i>література, матеріали лекції</i>
8	<i>Підготовка до лекції 8</i>	4	<i>Базова та додаткова література, матеріали лекції</i>
9	<i>Підготовка до лекції 9</i>	4	<i>Базова та додаткова література, матеріали лекції</i>
10	<i>Підготовка до лекції 10</i>	4	<i>Базова та додаткова література, матеріали лекції</i>
11	<i>Підготовка до лекції 11</i>	4	<i>Базова та додаткова література, матеріали лекції</i>
12	<i>Підготовка до лекції 12</i>	4	<i>Базова та додаткова література, матеріали лекції</i>
13	<i>Підготовка до лекції 13</i>	4	<i>Базова та додаткова література, матеріали лекції</i>
14	<i>Підготовка до лекції 14</i>	4	<i>Базова та додаткова література, матеріали лекції</i>
15	<i>Підготовка до лекції 15</i>	4	<i>Базова та додаткова література, матеріали лекції</i>
16	<i>Підготовка до лабораторного заняття 1 та доопрацювання наскрізного проєкту</i>	4	<i>Матеріали відповідних лекцій та документація обраного технологічного стеку</i>
17	<i>Підготовка до лабораторного заняття 2 та доопрацювання наскрізного проєкту</i>	4	<i>Матеріали відповідних лекцій та документація обраного технологічного стеку</i>

18	Підготовка до лабораторного заняття 3 та доопрацювання наскрізного проєкту	4	Матеріали відповідних лекцій та документація обраного технологічного стеку
19	Підготовка до лабораторного заняття 4 та доопрацювання наскрізного проєкту	4	Матеріали відповідних лекцій та документація обраного технологічного стеку
20	Підготовка до лабораторного заняття 5 та доопрацювання наскрізного проєкту	4	Матеріали відповідних лекцій та документація обраного технологічного стеку
21	Підготовка до лабораторного заняття 6 та доопрацювання наскрізного проєкту	4	Матеріали відповідних лекцій та документація обраного технологічного стеку
22	Підготовка до лабораторного заняття 7 та доопрацювання наскрізного проєкту	4	Матеріали відповідних лекцій та документація обраного технологічного стеку
23	Підготовка до модульної контрольної роботи	6	Матеріал лекцій 1–15
24	Підготовка до екзамену	10	Весь матеріал семестру

Разом: 104 години.

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Відвідування лекційних занять є рекомендованим та необхідним для послідовного засвоєння матеріалу. Відсутність на занятті сама по собі не передбачає штрафних балів; рейтинговий бал формується за результатами виконання контрольних заходів.

Лабораторні роботи утворюють один наскрізний проєкт. Здобувач самостійно обирає предметну область, мову програмування, фреймворки, бази даних, спосіб інтеграції ШІ та джерело ШІ-компонента: власну або локальну модель, безкоштовний публічний API, власну підписку на зовнішній сервіс чи інший технічно обґрунтований варіант. Проєкт має бути завершеною програмною системою, що вирішує практичну задачу; тривіальний застосунок, єдиною функцією якого є передавання запиту до зовнішнього ШІ API та відображення відповіді, не вважається достатнім. Технологічний стек сам по собі не є предметом оцінювання.

Правила поведінки на заняттях ґрунтуються на взаємній повазі, академічній доброчесності та дотриманні Кодексу честі КПІ ім. Ігоря Сікорського.

Використання засобів штучного інтелекту під час виконання лабораторних робіт дозволяється, зокрема для пошуку інформації, роботи з документацією, генерації та рецензування коду, підготовки тестових даних і пошуку помилок. Здобувач повинен розуміти та вміти пояснити всі рішення й артефакти, які подає на оцінювання, а використання ШІ та його роль у виконанні роботи мають бути задекларовані у звіті. Під час модульної контрольної роботи та екзамену використання генеративних засобів ШІ заборонено, якщо викладачем не визначено інше.

У разі повітряної тривоги, відсутності електропостачання або інших обставин непереборної сили строки виконання робіт можуть бути скориговані за погодженням з викладачем.

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

Семестровий рейтинг складається з результатів виконання та захисту 4 лабораторних робіт, що виконуються протягом 7 лабораторних занять, і модульної контрольної роботи. Максимальний семестровий рейтинг – 50 балів. Екзамен – 50 балів. Максимальна підсумкова оцінка – 100 балів.

Лабораторні роботи: ЛР1 – 4 бали; ЛР2 – 8 балів; ЛР3 – 8 балів; ЛР4 – 8 балів. Максимальна кількість балів за лабораторні роботи: 28 балів.

Критерії оцінювання лабораторних робіт орієнтовані насамперед на досягнення заявленого результату та працездатність системи; вибір мови програмування, фреймворків і постачальника ШІ не впливає на оцінку сам по собі.

- ЛР1 – до 4 балів: до 3 балів за повноту, логічність і обґрунтованість архітектури; до 1 бала за захист та пояснення прийнятих рішень.*
- ЛР2–ЛР4 – по 8 балів: до 6 балів за працездатність, повноту виконання поставленої задачі та коректність інтеграції компонентів; до 2 балів за захист, розуміння реалізації та здатність пояснити прийняті рішення.*

Модульна контрольна робота: максимум 22 бали. Робота проводиться у тестовій формі та складається з 22 запитань за матеріалом лекцій 1–15. Кожна правильна відповідь оцінюється в 1 бал; неправильна або відсутня відповідь – 0 балів.

Рейтингова шкала протягом семестру: $RC = 28 \text{ балів (лабораторні роботи)} + 22 \text{ бали (МКР)} = 50 \text{ балів}$.

Календарний контроль проводиться двічі на семестр як моніторинг поточного стану виконання вимог силабусу. Умовою позитивного календарного контролю є виконання запланованих на відповідний момент складових наскрізного проєкту та не менше 50 % від максимально можливої кількості балів, доступної на момент проведення контролю.

Умови допуску до екзамену: виконані та захищені всі 4 лабораторні роботи; виконана модульна контрольна робота; семестровий рейтинг не менше 25 балів.

Структура екзамену (50 балів):

- Тестова частина – 25 балів: 25 тестових запитань за матеріалом дисципліни, по 1 балу за правильну відповідь.*
- Практична частина – 25 балів: аналіз архітектурного кейсу за тематикою лабораторного проєкту з вибором і обґрунтуванням структури системи, способу інтеграції ШІ, версіонування, контейнеризації, автоматизації та базових заходів надійності.*

Критерії оцінювання практичного кейсу: до 15 балів за коректність та повноту архітектурного рішення; до 5 балів за врахування надійності, експлуатації та можливих відмов; до 5 балів за аргументацію рішень і аналіз компромісів.

Заохочувальні бали можуть нараховуватися за технічно обґрунтоване розширення наскрізного проекту понад обов'язкові вимоги: розгортання у хмарному середовищі, розширений CI/CD pipeline, моніторинг, оптимізацію моделі, нетривіальні архітектурні рішення або іншу погоджену з викладачем функціональність. Максимум – 5 заохочувальних балів за семестр; семестровий рейтинг з їх урахуванням не може перевищувати 50 балів.

Результати неформальної та інформальної освіти можуть визнаватися відповідно до чинного положення КПІ ім. Ігоря Сікорського, якщо їх зміст відповідає окремим результатам навчання дисципліни. Рішення про зарахування конкретної складової приймає викладач на основі наданих студентом підтверджувальних матеріалів.

Підсумкова оцінка визначається як сума семестрового рейтингу та балів за екзамен: $R = RC + RE$. Для позитивної оцінки за екзаменаційну складову студент має набрати не менше 30 балів із 50.

Кількість балів	Оцінка
95...100	Відмінно
85...94	Дуже добре
75...84	Добре
65...74	Задовільно
60...64	Достатньо
Менше 60	Незадовільно
Не виконані умови допуску	Не допущено

Робочу програму навчальної дисципліни (силабус):

Складено: аспірант, Лісняк О.М.

Ухвалено кафедрою ПЗКС (протокол № 12 від 11.02.2026 р.)

Погоджено Методичною комісією факультету програмних систем та прикладної математики (протокол № 7 від 23.02.2026 р.)

Додаток 1. Орієнтовний перелік питань, які виносяться на семестровий контроль

- 1. ШІ-модель як компонент програмної системи: основні відмінності між моделлю та завершеним програмним продуктом.*
- 2. Основні компоненти програмної системи зі штучним інтелектом та їх відповідальність.*
- 3. Модульність, зв'язність, залежності та межі компонентів у програмній архітектурі.*
- 4. Embedded model, Model-as-a-Service та зовнішній AI API: переваги, недоліки й критерії вибору.*
- 5. Проектування API-контрактів між компонентами ШІ-системи.*
- 6. Підходи до інтеграції зовнішніх AI API та керування secrets.*
- 7. Інтеграція локальної або власної моделі у програмну систему.*
- 8. Timeouts, retries, rate limiting, caching і fallback у ШІ-системах.*
- 9. Версіонування коду, конфігурацій, prompts, моделей і даних.*
- 10. Призначення DVC та випадки, коли він доцільний.*
- 11. Поняття відтворюваності та experiment tracking у ШІ-системах.*
- 12. Призначення MLflow та основні сутності experiment tracking.*
- 13. Принципи контейнеризації та базові поняття Docker.*
- 14. Багатокомпонентна система та призначення Docker Compose.*
- 15. Базові принципи CI/CD і автоматизації процесу розроблення.*
- 16. Git hooks та CI pipeline: відмінності й сфери застосування.*
- 17. Базові підходи до експлуатації та моніторингу ШІ-систем.*
- 18. Поняття data drift і model degradation.*
- 19. ONNX та квантування як підходи до локальної оптимізації моделей.*
- 20. Проектування наскрізної архітектури ШІ-системи з урахуванням надійності, ресурсних обмежень і технологічних компромісів.*