

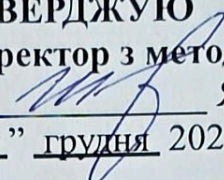
**ЗАКЛАД ВИЩОЇ ОСВІТИ
«УНІВЕРСИТЕТ КОРОЛЯ ДАНИЛА»**

Факультет суспільних і прикладних наук

Кафедра інформаційних технологій

ЗАТВЕРДЖУЮ

Проректор з методичної роботи

 **Ярослав ШТАНЬКО**

“ 19 ” грудня 2025 р.

Принципи розробки з використанням сучасних Python Web Frameworks

СИЛАБУС ТРЕНІНГ-КУРСУ

Галузь знань:	F Інформаційні технології
Спеціальність:	F2 Інженерія програмного забезпечення
Освітньо-професійна (освітньо-наукова) програма:	Розробка та тестування програмного забезпечення
Освітній рівень:	перший (бакалаврський)
Статус дисципліни:	вибіркова
Мова викладання, навчання та оцінювання:	українська

**Івано-Франківськ
2025**

РОЗРОБНИК:
асистент кафедри ІТ

Юрій РИБАК

ЗАТВЕРДЖЕНО:

на засіданні кафедри ІТ
протокол № 5 від 18.12 2025 р.
PhD, заступник завідувача кафедри

Олександр ІВАНОВ

УЗГОДЖЕНО:

Гарант ОП
к.ф-м.н., доцент кафедри

Володимир МАКОВИШИН

СХВАЛЕНО:

на засіданні Науково-методичної ради, протокол № від 19.12 2025

e-mail	yurii.rybak@ukd.edu.ua
Номер аудиторії чи кафедри	Кафедра інформаційних технологій, ауд 206
Посилання на сайт	https://ukd.edu.ua
Сторінка курсу в СДО	<u>Тренінг-курс "Принципи розробки з використанням сучасних Python Web Frameworks"</u>

ВСТУП

Анотація тренінг-курсу «Принципи розробки з використанням сучасних Python Web Frameworks»

Тренінг-курс спрямований на формування у студентів практичних навичок розробки сучасних вебзастосунків та програмних інтерфейсів API з використанням мови програмування Python на сучасних вебфреймворках. У межах тренінг-курсу розглядаються основні принципи побудови backend частини, маршрутизації, асинхронне програмування, а також архітектурні підходи до організації вебсервісів.

Особлива увага приділяється застосування фреймворку FastAPI для створення високопродуктивних RESTful API, формуванню навичок роботи з базами даних через ORM бібліотеки SQLAlchemy та управління міграціями баз даних через Alembic. У навчальному процесі також розглядаються сучасні інструменти типізації, управління залежностями через менеджер пакетів uv та основи контейнеризації Docker. Все це дозволить студентам зрозуміти повний цикл розробки, тестування та розгортання серверних застосунків. Тренінг-курс має практичну спрямованість і передбачає виконання серії практичних завдань та навчального проекту, результатом яких є створення завершеного, задокументованого та масштабованого API. Отримані знання та навички є необхідною основою для подальшого вивчення backend-розробки, мікросервісної архітектури та інженерії програмного забезпечення в цілому.

Мета вивчення тренінг-курсу – формування у здобувачів освіти теоретичних знань та практичних навичок з розробки серверної частини програмних продуктів з використанням мови програмування Python та фреймворку FastAPI. У межах дисципліни здобувачі освіти оволодівають принципами побудови REST API, асинхронної обробки HTTP-запитів, валідації даних та надійної взаємодії з базами даних. Отримані знання дозволяють створювати швидкі, безпечні та сучасні вебсервіси.

Предметом навчальної дисципліни є вивчення теоретичних основ і практичних методів розробки веб-додатків та API у клієнт-серверних архітектурах. Програма дисципліни охоплює основні поняття побудови бекенду, роботу з протоколом HTTP, використання Pydantic для типізації та валідації даних, а також застосування SQLAlchemy для обробки реляційних даних. Дисципліна сприяє формуванню у здобувачів освіти цілісного розуміння ролі backend-частини у сучасних програмних системах та її впливу на загальну продуктивність і стабільність продукту.

Основними завданнями дисципліни є:

- закріплення базових знань та просунутих концепцій мови програмування Python, необхідних для веброботи;
- вивчення принципів побудови RESTful API та асинхронної моделі обробки запитів;
- освоєння інструментарію FastAPI для створення та документування високопродуктивних вебсервісів;
- вивчення методів жорсткої типізації, валідації та серіалізації даних;

- опанування роботи з реляційними базами даних за допомогою ORM та управління міграціями;
- формування навичок проєктування, структурування, тестування та контейнеризації вебпроєктів;
- реалізація навчальних та проєктних завдань, спрямованих на створення повноцінного backend-застосунку.

У результаті вивчення тренінг-курсу студент повинен **знати:**

- принципи побудови сучасних веб-додатків, клієнт-серверної архітектури та REST API;
- основи та переваги асинхронного програмування в Python;
- можливості, архітектурні особливості та екосистему фреймворку FastAPI;
- методи ефективної взаємодії веб-додатків з базами даних;
- основні підходи до організації структури, тестування та розгортання серверного коду.

У результаті вивчення тренінг-курсу студент повинен **уміти:**

- створювати веб-сервіси та програмні інтерфейси з використанням Python та FastAPI;
- застосовувати асинхронні підходи при розробці бізнес-логіки застосунку;
- розробляти моделі баз даних, налаштовувати зв'язки та виконувати запити за допомогою об'єктно-реляційного відображення;
- реалізовувати валідацію вхідних даних та використовувати автоматичну генерацію інтерактивної документації (Swagger/OpenAPI);
- проєктувати, структурувати та готувати до розгортання завершені серверні застосунки.

ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Курс	3		
Семестр	7		
Кількість кредитів ECTS	3		
Аудиторні навчальні заняття		денна форма	заочна форма
	лекції	0	0
	практичні	20	8
Самостійна робота (в годинах)		70	82
Форма підсумкового контролю	Залік		

Структурно-логічна схема вивчення навчальної дисципліни:

Пререквізити	Постреквізити
Основи програмування	Кваліфікаційна робота
Алгоритми та структури даних	

ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Зміст практичних занять

Тема 1. Сучасний Python для веброзробки: типізація та асинхронність (2 год.)

Огляд ключових інструментів для створення сучасного бекенду. Використання type hints(анотацій типів) у Python як основи для надійної архітектури. Основи асинхронного програмування(async/await, asyncio). Конкурентність та переваги асинхронного вводу/виводу(I/O) для вебзастосунків.

Тема 2. Принципи REST API та створення першого застосунку на FastAPI (4 год.)

Поняття клієнт-серверної архітектури, HTTP-методи та статус-коди. Ініціалізація FastAPI застосунку. Створення базових маршрутів(ендпоінтів). Знайомство з автоматичною генерацією інтерактивної документації(Swagger UI, ReDoc).

Тема 3. Валідація даних(Pydantic) та маршрутизація (2 год.)

Використання бібліотеки Pydantic для створення схем даних. Серіалізація, десеріалізація та суворі валідації вхідних і вихідних запитів. Організація маршрутів у великих проєктах за допомогою APIRouter. Робота з параметрами шляху та параметрами запиту.

Тема 4. Впровадження залежностей(Dependency Injection) та обробка помилок (2 год.)

Концепція Dependency Injection у FastAPI та використання Depends. Застосування залежностей для перевикористання коду(наприклад, отримання сесії бази даних або перевірка прав доступу). Обробка виключень та налаштування відповідей з помилками(HTTPException).

Тема 5. Інтеграція з базами даних: ORM SQLAlchemy (4 год.)

Основи роботи з реляційними базами даних у клієнт-серверних застосунках. Налаштування асинхронного з'єднання з БД. Оголошення моделей даних за допомогою SQLAlchemy. Створення зв'язків між таблицями(One-to-Many, Many-to-Many). Виконання базових CRUD-операцій.

Тема 6. Управління міграціями баз даних за допомогою Alembic (2 год.)

Поняття та необхідність міграцій баз даних у процесі розробки. Ініціалізація Alembic у проєкті. Автоматична генерація міграцій на основі змін у моделях SQLAlchemy. Застосування та відкат міграцій(upgrade/downgrade).

Тема 7. Безпека застосунку: автентифікація, авторизація та основи розгортання (4 год.)

Реалізація механізмів безпеки у веб-застосунках. Хешування паролів. Створення, перевірка та декодування JSON Web Tokens(JWT). Захист окремих маршрутів API від неавторизованого доступу за допомогою залежностей. Огляд основ контейнеризації(створення базового Dockerfile) для підготовки FastAPI застосунку до розгортання.

Зміст самостійної роботи здобувачів

Розподіл годин, виділених на вивчення дисципліни:

Найменування видів робіт	Розподіл годин за формами навчання	
	денна	заочна
Самостійна робота, год, у т.ч.:	70	82
Опрацювання матеріалу, викладеного на лекціях		
Підготовка до практичних занять та контрольних заходів		
Підготовка звітів з практичних робіт		
Підготовка до поточного контролю		
Опрацювання матеріалу, винесеного на самостійне вивчення		

ПОЛІТИКА КУРСУ

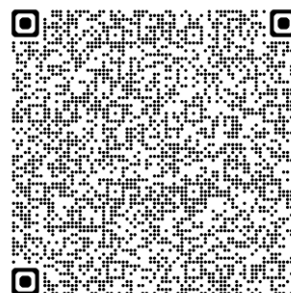
1) щодо системи поточного і підсумкового контролю

Організація поточного та підсумкового семестрового контролю знань студентів, проведення практик та атестації, переведення показників академічної успішності за 100-бальною шкалою в систему оцінок за національною шкалою здійснюється згідно з “Положенням про систему поточного і підсумкового контролю, оцінювання знань та визначення рейтингу здобувачів освіти”. Ознайомитись з документом можна за покликанням.



2) щодо оскарження результатів контрольних заходів

Здобувачі вищої освіти мають право на оскарження оцінки з дисципліни отриманої під час контрольних заходів. Апеляція здійснюється відповідно до «Положення про політику та врегулювання конфліктних ситуацій». Ознайомитись з документом можна за покликанням.



3) щодо відпрацювання пропущених занять

Згідно “Положення про організацію освітнього процесу” здобувач допускається до семестрового контролю з **конкретної навчальної дисципліни (семестрового екзамену, диференційованого заліку)**, якщо він виконав усі види робіт, передбачені на семестр навчальним планом та силабусом/робочою програмою навчальної дисципліни, підтвердив опанування на мінімальному рівні результатів навчання (отримав ≥ 35 бали), відпрацював визначені індивідуальним навчальним планом всі лекційні, практичні, семінарські та лабораторні заняття, на яких він був відсутній. Ознайомитись з документом можна за покликанням.



4) щодо дотримання академічної доброчесності

“Положення про академічну доброчесність” закріплює моральні принципи, норми та правила етичної поведінки, позитивного, сприятливого, доброчесного освітнього і наукового середовища, професійної діяльності та професійного спілкування спільноти Університету, викладання та провадження наукової (творчої) діяльності з метою забезпечення довіри до результатів навчання. Ознайомитись з документом можна за [посиланням](#).



5) щодо використання штучного інтелекту

“Положення про академічну доброчесність” визначає політику щодо використання технічних засобів на основі штучного інтелекту в освітньому процесі. Ознайомитись з документом можна за [покликанням](#). “Положення про систему запобігання та виявлення академічного плагіату, самоплагіату, фабрикації та фальсифікації академічних творів” містить рекомендації щодо використання в академічних текстах генераторів на основі штучного інтелекту. Ознайомитись з документом можна за [покликанням](#).



6) щодо використання технічних засобів в аудиторії та правила комунікації

Використання мобільних телефонів, планшетів та інших гаджетів під час лекційних та практичних занять дозволяється виключно у навчальних цілях (для уточнення певних даних, перевірки правопису, отримання довідкової інформації тощо). На гаджетах повинен бути активований режим «без звуку» до початку заняття. Під час занять заборонено надсилання текстових повідомлень, прослуховування музики, перевірка електронної пошти, соціальних мереж тощо, окрім виробничої необхідності. Під час виконання заходів контролю використання гаджетів заборонено (за винятком, коли це передбачено умовами його проведення). У разі порушення цієї заборони результат анулюється без права перескладання.

Комунікація відбувається через електронну пошту і сторінку дисципліни в курсі СДО.

7) щодо зарахування результатів навчання, здобутих шляхом формальної/інформальної освіти

Процедури визнання результатів навчання, здобутих шляхом формальної/інформальної освіти визначаються «Положенням про порядок визнання результатів навчання, здобутих шляхом неформальної та / або інформальної освіти». Ознайомитись з документом можна за [покликанням](#).



Окремі теми можуть бути зараховані як вивчені за рахунок публікації одноосібних або у співавторстві статей, тез, виступів на конференціях. Можливість зарахування попередньо узгоджується з викладачем з позицій актуальності теми, журналу, конференції, тощо.

МЕТОДИ НАВЧАННЯ

При вивченні дисципліни застосовується комплекс методів для організації навчання студентів з метою розвитку їх логічного та абстрактного мислення, творчих здібностей, підвищення мотивації до навчання та формування особистості майбутнього фахівця.

ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ

Вид	Зміст	% від загальної оцінки	Бал	
			min	max
Поточні контрольні заходи	всього	60	35	60
Підсумкові контрольні заходи	екзамен	40	24	40
Всього:	-	100	60	100

Процедура проведення контрольних заходів, а саме поточного контролю знань протягом семестру та підсумкового семестрового контролю, регулюється «Положенням про систему поточного та підсумкового контролю оцінювання знань та визначення рейтингу студентів».

Фіксація **поточного** контролю здійснюється в “Електронному журналі обліку успішності академічної групи” на підставі чотирибальної шкали – “2”; “3”; “4”; “5”. У разі відсутності студента на занятті виставляється “н”. За результатами поточного контролю у Журналі, автоматично визначається підсумкова оцінка, здійснюється підрахунок пропущених занять.

Усі пропущені заняття, а також негативні оцінки студенти зобов'язані відпрацювати впродовж трьох наступних тижнів. У випадку недотримання цієї норми, замість “н” в журналі буде виставлено “0” (нуль балів), без права перездачі. Відпрацьоване лекційне заняття в електронному журналі позначається літерою «в».

Критерії оцінювання:

«незадовільно»	Студент володіє матеріалом лише на рівні розпізнавання і відтворення окремих фактів, елементів та об'єктів, що виражаються окремими словами чи реченнями; володіння матеріалом обмежується елементарним рівнем засвоєння, викладення уривається речення; здатний висловити думку на елементарному рівні; володіє матеріалом на рівні окремих фрагментів, що становлять незначну частину навчального матеріалу; не може розпізнати або відтворити матеріал практичних.
«задовільно»	студент володіє матеріалом на початковому рівні, значну частину матеріалу відтворює на репродуктивному рівні; володіє матеріалом на рівні вищому за початковий, за допомогою викладача може логічно відтворити значну його частину; може відтворити значну частину теоретичного матеріалу, виявляючи знання і розуміння основних положень; за допомогою викладача може аналізувати навчальний матеріал, порівнювати, робити висновки та виправляти помилки; може розпізнати або відтворити за прикладом матеріал практичних занять.
«добре»	Студент здатний застосовувати вивчений матеріал у стандартних ситуаціях, частково контролювати власні навчальні дії та наводити окремі власні приклади на підтвердження певних тверджень; вміє порівнювати, узагальнювати та систематизувати інформацію під керівництвом викладача, в цілому самостійно застосовувати її на практиці; контролює власну діяльність, виправляє помилки та добирає аргументи на підтвердження певних думок під керівництвом викладача; вільно володіє вивченим обсягом матеріалу та вміє застосовувати його на практиці; вільно розв'язує задачі в стандартних ситуаціях, самостійно виправляє помилки та добирає переконливі аргументи на підтвердження вивченого матеріалу.
«відмінно»	студент виявляє початкові творчі здібності, самостійно визначає окремі цілі власної навчальної діяльності та оцінює нові факти, явища і ідеї; знаходить джерела інформації та самостійно використовує їх відповідно до цілей, поставлених викладачем; здатний вільно дискутувати на теми, пов'язані з матеріалом навчальної дисципліни, висловлювати власні думки та визначати програму особистої діяльності; самостійно оцінює різноманітні явища і факти, виявляючи особисту позицію щодо них; без

<p>допомоги викладача знаходить джерела інформації та використовує одержані відомості відповідно до мети й завдань власної пізнавальної діяльності; використовує набуті знання і вміння в нестандартних ситуаціях, виявляє вміння знаходити альтернативні шляхи для вирішення завдань та здобути нові знання самостійно.</p>

До підсумкового контролю допускаються студенти які за результатами поточного контролю отримали не менше 35 балів. Усі студенти, що отримали 34 балів і менше, не допускаються до складання підсумкового контролю і на підставі укладання додаткового договору, здійснюють повторне вивчення дисципліни впродовж наступного навчального семестру. За результатами підсумкового контролю (екзамен) студент може отримати 40 балів. Студенти, які під час підсумкового контролю отримали 24 бали і менше, вважаються такими, що не здали екзамен і повинні йти на перездачу.

Загальна семестрова оцінка з дисципліни, яка виставляється в екзаменаційних відомостях оцінюється в балах (згідно з **Шкалою оцінювання знань за ЄКТС**) і є сумою балів отриманих під час поточного та підсумкового контролю.

Підсумковий контроль з тренінг-курсу “Принципи розробки з використанням сучасних Python Web Frameworks” проводиться у формі заліку. Залік виконується у вигляді письмової роботи, яку студенти готують протягом усього тренінг-курсу відповідно до виданих завдань.

Шкала оцінювання знань за ЄКТС:

Оцінка за національною шкалою	Рівень досягнень, %	Шкала ECTS
Національна диференційована шкала		
Відмінно	90 – 100	A
Добре	83 – 89	B
	75 – 82	C
Задовільно	67 – 74	D
	60 – 66	E
Незадовільно	35 – 59	FX
	0 – 34	F

Національна недиференційована шкала		
Зараховано	60 – 100	-
Не зараховано	0 – 59	-

Студенти, які не з'явилися на підсумковий контроль без поважних причин, вважаються такими, що одержали незадовільну оцінку.

Критерії оцінювання підсумкового проєкту

1. Якість коду та архітектурні рішення: правильне використання type hints та принципів асинхронного програмування; логічний поділ коду на модулі та використання маршрутизації; відсутність дублювання коду та дотримання стандартів написання коду на Python(PEP 8).

2. Реалізація бізнес-логіки та роботи з даними: створення повноцінного RESTful API із суворою перевіркою та серіалізацією вхідних даних. Налаштування коректної взаємодії з реляційними базами даних через ORM: виконання CRUD-операцій та побудова зв'язків між таблицями. Ефективне управління структурою бази даних за допомогою міграцій. Надійна імплементація механізмів безпеки: хешування паролів та застосування JWT токенів доступу.

3. Стабільність, розгортання та документація: Коректне перехоплення HTTPException для чіткого інформування клієнта про статус помилок. Наявність інтерактивної та зрозумілої документації для API на основі OpenAPI. Базова підготовка проєкту до деплою за допомогою контейнеризації Docker.

4. Захист проєкту: Здатність здобувача освіти аргументовано пояснити обрані архітектурні підходи та логіку алгоритмів. Демонстрація розуміння власного коду та готовність відповідати на питання викладача щодо реалізації проєкту.

Підсумковий проєкт передбачає створення завершеного, задокументованого та масштабованого API. Проєкт реалізується за ітераційним принципом протягом курсу. Таким чином, залікова робота є результатом послідовної та систематичної роботи студента під час аудиторних та самостійних занять.

Шкала оцінювання проєкту

Відмінно. Продемонстровано глибоке розуміння технологій. Проєкт має ідеальну структуру, асинхронність та типізація застосовані бездоганно. Всі ендпоінти працюють стабільно, зв'язки в БД спроектовані правильно, безпека повністю функціонує. Проєкт супроводжується вичерпною документацією та

має готовий Docker-контейнер. На захисті студент впевнено аргументує свої рішення та легко орієнтується в коді.

Добре. Основні вимоги курсу виконані, проте існують незначні недоліки. Проєкт є функціональним, але присутні неоптимальні архітектурні рішення, часткове дублювання коду або дрібні похибки у складних SQL- чи ORM-запитах. Документація та Docker-файл наявні в базовому вигляді. Під час захисту студент загалом розуміє свій код, але пояснення складних деталей потребує додаткового часу.

Задовільно. Виконано лише мінімально необхідний обсяг роботи(базовий CRUD). Код може бути недостатньо структурованим, асинхронність застосована лише формально. Складні інструменти працюють з перебоями або відсутні. Налаштування Docker не виконано. На захисті студент плутається в деталях і не здатний самостійно модифікувати програму.

Незадовільно. Проєкт містить критичні помилки, через які сервер падає або не запускається. Студент не здатний пояснити логіку написаних рядків, що викликає підозру щодо самостійності виконання роботи.

РЕКОМЕНДОВАНІ ДЖЕРЕЛА ІНФОРМАЦІЇ

Основна література

1. Lubanovic, B. (2023). *FastAPI: Modern Python Web Development. Sebastopol: O'Reilly Media.* 368 с.
2. De Luca, G. (2024). *FastAPI Cookbook. Birmingham: Packt Publishing.* 436 с.
3. Peters, T. (2024). *Practical Python Backend Programming: Build Flask and FastAPI Applications. GitforGits.* 273 с.
4. Maxwell, M. J. (2025). *Python and SQLAlchemy: A Hands-On Introduction to Database Mastery. Amazon Digital Services.* 306 с.

Додаткова література

5. Dawson, M. (2022). *Python Programming for the Absolute Beginner (4th Edition). Boston: Cengage Learning.* 672 с.
6. Percival, H., & Gregory, B. (2020). *Architecture Patterns with Python: Enabling Test-Driven Development, Domain-Driven Design, and Event-Driven Microservices. Sebastopol: O'Reilly Media.* 306 с.
7. Fowler, M. (2022). *Python Concurrency with asyncio. Shelter Island: Manning Publications.* 392 с.
8. Anaya, M. (2021). *Clean Code in Python: Develop maintainable and efficient code (2nd Edition). Birmingham: Packt Publishing.* 344 с.

Електронні інформаційні ресурси

9. Programming with Mosh. *Python Full Course for Beginners* [відео]. YouTube <https://www.youtube.com/watch?v=K5KVEU3aaeQ> (Дата доступу: 12.12.2025).

10. freeCodeCamp.org. *Automate with Python – Full Course for Beginners* [відео]. YouTube <https://www.youtube.com/watch?v=PXMJ6FS7llk> (Дата доступу: 12.12.2025).

11. freeCodeCamp.org та CS50. *Harvard CS50's Introduction to Programming with Python – Full University Course* [відео]. YouTube https://www.youtube.com/watch?v=nLRL_NcnK-4 (Дата доступу 12.12.2025).

12. Школа програмування. *Повний Курс Python / Вивчення в одному відео для початківців з нуля* [відео]. YouTube <https://www.youtube.com/watch?v=Gye2Na7C8jg> (Дата доступу 12.12.2025).

13. Tech With Tim. *Learn Fast API With This ONE Project* [відео]. YouTube <https://www.youtube.com/watch?v=SR5NYCdZKkc> (Дата доступу: 12.12.2025).

14. NeuralNine. *SQLAlchemy Crash Course - Master Databases in Python* [відео]. YouTube <https://www.youtube.com/watch?v=529LYDgRTgQ> (Дата доступу: 12.12.2025).

15. ArjanCodes. *SQLAlchemy: The BEST SQL Database Library in Python* [відео]. YouTube <https://www.youtube.com/watch?v=aAy-B6KPld8> (Дата доступу: 12.12.2025).

16. Code with Josh. *FastAPI SQLAlchemy Tutorial 2025 — Build a REST API with SQL* [відео]. YouTube <https://www.youtube.com/watch?v=xq1Snezb1rs> (Дата доступу: 12.12.2025).

17. freeCodeCamp.org. *Docker Tutorial for Beginners* [відео]. YouTube <https://www.youtube.com/watch?v=pTFZFXd4hOI> (Дата доступу: 12.12.2025).

18. ArjanCodes. *Asynchronous Programming in Python / Asyncio* [відео]. YouTube <https://www.youtube.com/watch?v=Qb9s3IiMHAk> (Дата доступу: 12.12.2025).