

ЗАТВЕРДЖУЮ

Декан факультету ФІТ

Тетяна ГОВОРУЩЕНКО

2025 р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Веб-візуалізація та веб-орієнтовані системи керування

Назва дисципліни

Призначення Робочої програми

Рівень вищої освіти

Мова навчання

Обсяг дисципліни, кредитів ЄКТС

Статус дисципліни

Факультет

Кафедра

Для освітніх програм різних спеціальностей

Перший (бакалаврський)

Українська

8

Вибіркова

Інформаційних технологій

Автоматизації, комп'ютерно-інтегрованих технологій та автоматизації

Форма здобуття освіти	Обсяг дисципліни		Кількість годин						Форма семестрового контролю
			Аудиторні заняття					Самостійна робота (в т.ч. ІРС)	Залік
	Кредити ЄКТС	Години	Разом	Лекції	Лабораторні роботи	Практичні заняття	Семінарські заняття		
Д	8	240	82	32	32	18		158	+

Робоча програма складена на основі освітніх програм підготовки бакалавра та стандарту вищої освіти спеціальності.

Робоча програма складена



канд. техн. наук, доцент Людмила КОРЕЦЬКА

Науковий ступінь, вчене звання, ім'я, ПРІЗВИЩЕ

Схвалена на засіданні кафедри

Автоматизації, комп'ютерно-інтегрованих технологій та автоматизації

Назва

Протокол від 01.09 2025 №1

Зав. кафедри АКІТтаР

Назва



Підпис

Людмила КОРЕЦЬКА

Ім'я, ПРІЗВИЩЕ

Хмельницький 2025

3. Пояснювальна записка

Дисципліна «Веб-візуалізація та веб-орієнтовані системи керування» належить до циклу вибірових фахових дисциплін і спрямована на поглиблення фахових умінь у розробленні веб-застосунків для моніторингу, візуалізації та керування технічними процесами з використанням мови програмування Python та сучасних веб-технологій. Опанування дисципліни сприяє розвитку практичних навичок, формує здатність проєктувати та впроваджувати веб-орієнтовані системи контролю та аналізу даних, а також розвиває аналітичне мислення та технічну грамотність. Дисципліна сприяє формуванню soft skills — уміння працювати в команді, комунікувати, приймати обґрунтовані інженерні рішення та ефективно реалізовувати проєктні завдання у сфері сучасних веб-технологій.

Мета дисципліни. Метою вивчення дисципліни «Веб-візуалізація та веб-орієнтовані системи керування» є формування у здобувачів освіти теоретичних знань і практичних навичок зі створення, розгортання та підтримки веб-застосунків для збору, обробки, візуалізації та керування даними у режимі реального часу з використанням сучасних технологій Python та веб-фреймворків.

Предмет дисципліни. Предметом вивчення є **методи, засоби та технології веб-програмування**, зокрема розроблення веб-інтерфейсів, серверних застосунків і систем обміну даними, а також принципи **побудови веб-орієнтованих систем моніторингу та керування технологічними процесами**.

Завдання дисципліни. Формування практичних навичок розроблення веб-застосунків для візуалізації даних і керування процесами з використанням мови Python та сучасних веб-фреймворків; засвоєння основ веб-програмування, клієнтсько-серверної архітектури, роботи з базами даних і API; опанування методів відображення та оновлення даних у реальному часі, інтеграції з мікроконтролерами (Arduino, ESP) і хмарними сервісами; ознайомлення з процесами тестування, розгортання та безпеки веб-додатків

Результати навчання. Після вивчення дисципліни студент повинен: *знати* основні принципи побудови веб-орієнтованих систем керування, архітектуру клієнт–серверних застосунків, принципи взаємодії через протоколи HTTP і WebSocket; розуміти структуру та можливості фреймворків Flask і бібліотек для веб-візуалізації; *володіти* базовими інструментами Python для розроблення, налагодження та тестування веб-застосунків; *уміти* створювати веб-інтерфейси для моніторингу та керування процесами, реалізовувати динамічне оновлення даних у режимі реального часу; *застосовувати* набуті знання для інтеграції веб-додатків із мікроконтролерами, базами даних і хмарними платформами; *обґрунтовувати* вибір технологій, архітектури та інструментів під час реалізації проєктів; *поєднувати* теоретичні знання з практичними навичками програмування для розв’язання прикладних інженерних задач; *демонструвати* здатність працювати з технічною документацією, проводити аналіз результатів, аргументовано захищати технічні рішення; *виявляти* ініціативність, здатність до самостійного навчання, ефективної комунікації та роботи в команді під час розроблення веб-орієнтованих систем керування.

4. Структура залікових кредитів дисципліни

Назва розділу (теми)	Кількість годин, відведених на:			
	лекції	лабораторні	практичні	СРС
Тема 1. Вступ до веб-розробки та основи Python	4	4	4	18
Тема 2. Створення веб-додатків у Flask	4	4		20
Тема 3. Робота з даними у веб-застосунках	4	4	4	20
Тема 4. Веб-візуалізація даних	4	4		20
Тема 5. API та взаємодія із зовнішніми джерелами даних	4	4	4	20
Тема 6. Веб-орієнтовані системи керування	4	4		20
Тема 7. Безпека та хмарні технології	4	4	6	20
Тема 8. Розгортання та підсумковий проєкт	4	4		20
Разом за семестр:	32	32	18	158

5. Програма навчальної дисципліни

5.1 Зміст лекційного курсу

Номер лекції	Перелік тем лекцій, їх анотації	Кількість годин
1	Вступ до дисципліни Мета та завдання курсу. Поняття веб-візуалізації та веб-орієнтованих систем. Архітектура клієнт-сервер, протоколи HTTP/HTTPS. Приклади сучасних веб-систем моніторингу та керування Літ.: [1, 2]	2
2	Мова програмування Python для веб-розробки Огляд можливостей Python. Основи синтаксису, типи даних, функції. Робота з файлами, бібліотеками, пакетами. Підготовка середовища розробки (Python, pip, venv). Літ.: [2, 5]	2
3	Основи веб-розробки у Flask Концепція мікрофреймворку Flask. Створення простого веб-сервера. Маршрутизація та шаблонізація (Jinja2). Обробка запитів GET і POST Літ.: [2, 8-10]	2
4	HTML, CSS і шаблони Jinja2 у Flask. Основи структури веб-сторінок. Використання CSS для оформлення інтерфейсу. Динамічне оновлення сторінок через Flask + Jinja2. Підключення Bootstrap для створення адаптивного дизайну. Літ.: [3]	2
5	Робота з формами та передачею даних Обробка форм у Flask. Передача даних між сторінками. Використання request, session, redirect. Практичний приклад: форма введення параметрів керування. Літ.: [1,2,3]	2
6	Збереження даних у базі даних Основи SQL та SQLite. ORM SQLAlchemy. Операції CRUD (Create, Read, Update, Delete). Відображення даних із бази у веб-інтерфейсі Літ.: [4]	2
7	Веб-візуалізація даних: графіки у Python Огляд бібліотек Matplotlib, Plotly, Chart.js. Побудова статичних і динамічних графіків. Формування JSON-даних для графіків. Приклад: моніторинг температури у Flask Літ.: [2]	2
8	Інтерактивна візуалізація у браузері Використання JavaScript у поєднанні з Python. Динамічне оновлення графіків через AJAX або WebSocket. Побудова панелі реального часу (реальні дані або емуляція). Літ.: [9, 10]	2
9	Робота з API Поняття REST API. Створення власного API у Flask / FastAPI. Формат обміну даними (JSON, XML). Зчитування даних з API зовнішніх сервісів. Літ.: [9, 10]	2

10	Підключення Python до мікроконтролерів Протоколи обміну: Serial, Wi-Fi (ESP8266, ESP32). Бібліотека pyserial. Зчитування даних з Arduino. Емуляція даних Arduino у Tinkercad Circuits. Літ.: [1, 10]	2
11	Веб-орієнтовані системи керування Принципи побудови SCADA-подібних веб-систем. Структура: контролер-сервер-інтерфейс користувача. Приклад: веб-панель керування температурою або двигуном. Літ.: [10, 11]	2
12	Обмін даними у реальному часі Використання WebSocket у Flask (Flask-SocketIO). Потоки даних від сенсорів до браузера. Відображення оновлень у режимі реального часу. Літ.: [10, 11]	2
13	Безпека веб-додатків Аутентифікація та авторизація користувачів. Захист форм і запитів (CSRF, SQL Injection). Використання Flask-Login. Безпечна передача даних. Літ.: [12]	2
14	Інтеграція з хмарними сервісами Публікація даних у Google Sheets, Firebase, ThingSpeak. Використання MQTT для IoT-проектів. Відображення хмарних даних у веб-інтерфейсі Літ.: [12]	2
15	Розгортання веб-додатків Підготовка застосунку до розгортання. Використання Git, Render, PythonAnywhere, Heroku. Налаштування середовища серверу. Тестування та підтримка веб-сервісу Літ.: [10]	2
16	Підсумковий проєкт Розробка власної веб-системи моніторингу або керування. Демонстрація результатів. Обговорення перспектив розвитку веб-візуалізації. Підбиття підсумків курсу. Літ.: [5]	2
Разом за семестр:		32

5.2 Зміст лабораторних занять

№ п/п	Тема лабораторного заняття	Кількість годин
1	Основи Python	4
2	Робота з даними у Python	4
3	Основи Flask	4
4	Форми та передача даних у Flask	4
5	Бази даних у Flask	4
6	Візуалізація даних у Flask	4
7	Взаємодія з мікроконтролерами	4
8	Розгортання веб-застосунку	4
Разом:		32

5.3 Зміст практичних занять

№ п/п	Тема практичного заняття	Кількість годин
1	Розроблення структури веб-застосунку	4
2	Система збору та візуалізації даних	4
3	Створення веб-орієнтованої системи керування	4
4	Підсумковий проєкт	4
5	Узагальнююче заняття	2
Разом за семестр		18

5.3 Зміст самостійної (у т.ч. індивідуальної) роботи здобувача вищої освіти

Самостійна робота студентів усіх форм здобуття освіти полягає у систематичному опрацюванні програмного матеріалу з відповідних джерел інформації, підготовці до виконання лабораторних робіт та практичних занять, тестування. До послуг студентів сторінка кафедри у Модульному середовищі для навчання, де розміщені Робоча програма дисципліни та необхідні матеріали з її навчально-методичного забезпечення та контролю результатів навчання.

Номер тижня	Вид самостійної роботи	Кількість годин
1-2	Опрацювання лекційного матеріалу. Підготовка до ЛР1 та ПЗ1. Підготовка до захисту ЛР1.	18
3-4	Опрацювання лекційного матеріалу. Підготовка до ЛР2 та ПЗ2. Підготовка до захисту ЛР2.	20
5-6	Опрацювання лекційного матеріалу. Підготовка до ЛР3 та ПЗ3. Підготовка до захисту ЛР3.	20
7-8	Опрацювання лекційного матеріалу. Підготовка до ЛР4 та ПЗ4. Підготовка до захисту ЛР4.	20
9-10	Опрацювання лекційного матеріалу. Підготовка до ЛР5 та ПЗ5. Підготовка до захисту ЛР5.	20
11-12	Опрацювання лекційного матеріалу. Підготовка до ЛР6 та ПЗ6. Підготовка до захисту ЛР6.	20
13-14	Опрацювання лекційного матеріалу. Підготовка до ЛР7 та ПЗ7. Підготовка до захисту ЛР7.	20
15-16	Опрацювання лекційного матеріалу. Підготовка до ЛР8 та ПЗ8. Підготовка до захисту ЛР8.	20
Разом:		158

Примітки: ЛР – лабораторна робота, ПЗ– практичне заняття.

6. Технології та методи навчання

Процес навчання з дисципліни ґрунтується на використанні традиційних та сучасних технологій та методів навчання, зокрема: лекції (з використанням методів візуалізації, проблемного й інтерактивного навчання, мотиваційних прийомів, інформаційно-комунікаційних технологій); лабораторні заняття (з використанням методів комп'ютерного моделювання, методів проєктної діяльності, тренінгових вправ, аналіз проблемних ситуацій, пояснення, дискусія тощо); практичні заняття (з використанням інструктування, демонстрування, розв'язування типових і прикладних задач, аналізу кейсів, ситуаційних завдань, елементів дискусії тощо); самостійна робота (опрацювання теоретичного матеріалу, підготовка до виконання практичних робіт, поточного та підсумкового контролю, виконання індивідуальних та домашніх завдань), з використанням інформаційно-комп'ютерних технологій та технологій дистанційного навчання.

7. Методи контролю

Поточний контроль здійснюється під час аудиторних практичних та лабораторних занять, а також у дні проведення контрольних заходів, встановлених робочою програмою і графіком освітнього процесу, в т.ч. з використанням Модульного середовища для навчання. При цьому використовуються такі методи поточного контролю:

- оцінювання результатів роботи на практичних заняттях (опитування теоретичного матеріалу, розв'язування задач, участь у обговоренні ситуацій);
- тестовий контроль засвоєння теоретичного та практичного матеріалу з теми;
- оцінювання результатів захисту лабораторних робіт;

Підсумкова семестрова оцінка виставляється за результатами поточного контролю. Здобувач вищої освіти, який набрав з будь-якого виду навчальної роботи, суму балів нижчу за 60 відсотків від максимального балу, вважається таким, який *має* академічну заборгованість. Ліквідація академічної заборгованості із семестрового контролю здійснюється у період екзаменаційної сесії або за графіком, встановленим деканатом відповідно до «Положення про контроль і оцінювання результатів навчання здобувачів вищої освіти у ХНУ».

8. Політика дисципліни

Політика навчальної дисципліни загалом визначається системою вимог до здобувача вищої освіти, що передбачені чинними положеннями Університету про організацію і навчально-методичне забезпечення освітнього процесу. Зокрема, проходження інструктажу з техніки безпеки; відвідування занять з дисципліни є обов'язковим. За об'єктивних причин (підтверджених документально) теоретичне навчання за погодженням із лектором може відбуватись в он-лайн режимі. Успішне опанування дисципліни і формування фахових компетентностей і програмних результатів навчання передбачає необхідність підготовки до практичних та лабораторних занять (вивчення

теоретичного матеріалу з теми, попередню підготовку протоколу роботи, підготовку до усного опитування для допуску до заняття (наведені у Методичних рекомендаціях до лабораторних занять)), активно працювати на занятті, брати участь у дискусіях щодо прийнятих рішень при виконанні здобувачами лабораторних робіт.

Здобувачі вищої освіти мають дотримуватися встановлених термінів виконання всіх видів навчальної роботи відповідно до робочої програми навчальної дисципліни. Термін захисту лабораторної роботи вважається своєчасним, якщо студент захистив її на наступному після виконання роботи занятті. Пропущене практичне і лабораторне заняття студент зобов'язаний відпрацювати у встановлений викладачем термін, але не пізніше, ніж за два тижні до кінця теоретичних занять у семестрі.

Засвоєння студентом теоретичного матеріалу з дисципліни оцінюється за результатами опитування під час практичних занять, тестування.

Здобувач вищої освіти, виконуючи самостійну роботу або індивідуальну роботу з дисципліни, має дотримуватися політики доброчесності (заборонені списування, плагіат (в т.ч. із використанням мобільних девайсів)). У разі виявлення порушення політики академічної доброчесності в будь-яких видах навчальної роботи здобувач вищої освіти отримує незадовільну оцінку і має повторно виконати завдання з відповідної теми (виду роботи), що передбачені робочою програмою. Будь-які форми порушення академічної доброчесності **не допускаються**.

У межах вивчення навчальної дисципліни здобувачам вищої освіти передбачено визнання і зарахування результатів навчання, набутих шляхом неформальної освіти, що розміщені на доступних платформах, які сприяють формуванню компетентностей і поглибленню результатів навчання, визначених робочою програмою дисципліни, або забезпечують вивчення відповідної теми та/або виду робіт з програми навчальної дисципліни (детальніше у Положенні про порядок визнання та зарахування результатів навчання здобувачів вищої освіти у ХНУ).

9. Оцінювання результатів навчання студентів у семестрі

Оцінювання академічних досягнень здобувача вищої освіти здійснюється відповідно до «Положення про контроль і оцінювання результатів навчання здобувачів вищої освіти у ХНУ». При поточному оцінюванні виконаної здобувачем роботи з кожної структурної одиниці і отриманих ним результатів викладач виставляє йому певну кількість балів із встановлених Робочою програмою для цього виду роботи. При цьому кожна структурна одиниця навчальної роботи може бути зарахована, якщо здобувач набрав не менше 60 відсотків (мінімальний рівень для позитивної оцінки) від максимально можливої суми балів, призначеної структурній одиниці.

При оцінюванні результатів навчання здобувачів вищої освіти з будь-якого виду навчальної роботи (структурної одиниці) рекомендується використовувати наведені нижче узагальнені критерії:

Таблиця – Критерії оцінювання навчальних досягнень здобувача вищої освіти

Оцінка та рівень досягнення здобувачем запланованих ПРН та сформованих компетентностей	Узагальнений зміст критерія оцінювання
Відмінно (високий)	Здобувач вищої освіти глибоко і у повному обсязі опанував зміст навчального матеріалу, легко в ньому орієнтується і вміло використовує понятійний апарат; уміє пов'язувати теорію з практикою, вирішувати практичні завдання, впевнено висловлювати і обґрунтовувати свої судження. Відмінна оцінка передбачає логічний виклад відповіді мовою викладання (в усній або у письмовій формі), демонструє якісне оформлення роботи і володіння спеціальними приладами та інструментами, прикладними програмами. Здобувач не вагається при видозміні запитання, вміє робити детальні та узагальнюючі висновки, демонструє практичні навички з вирішення фахових завдань. При відповіді допустив дві–три несуттєві <i>похибки</i> .
Добре (середній)	Здобувач вищої освіти виявив повне засвоєння навчального матеріалу, володіє понятійним апаратом, орієнтується у вивченому матеріалі; свідомо використовує теоретичні знання для вирішення практичних задач; виклад відповіді грамотний, але у змісті і формі відповіді можуть мати місце окремі неточності, нечіткі формулювання правил, закономірностей тощо. Відповідь здобувача вищої освіти будується на основі самостійного мислення. Здобувач вищої освіти у відповіді допустив дві–три <i>несуттєві помилки</i> .
Задовільно (достатній)	Здобувач вищої освіти виявив знання основного програмного матеріалу в обсязі, необхідному для подальшого навчання та практичної діяльності за професією, справляється з виконанням практичних завдань, передбачених програмою. Як правило, відповідь здобувача вищої освіти будується на рівні репродуктивного мислення, здобувач вищої освіти має слабкі знання структури навчальної дисципліни, допускає неточності і <i>суттєві помилки</i> у відповіді, вагається при відповіді на видозмінене запитання. Разом з тим, набув навичок, необхідних для виконання нескладних практичних завдань, які відповідають мінімальним критеріям оцінювання і володіє знаннями, що дозволяють йому під керівництвом викладача усунути неточності у відповіді.

Незадовільно (недостатній)	Здобувач вищої освіти виявив розрізнені, безсистемні знання, не вміє виділяти головне і другорядне, допускається помилок у визначенні понять, перекручує їх зміст, хаотично і невпевнено викладає матеріал, не може використовувати знання при вирішенні практичних завдань. Як правило, оцінка «незадовільно» виставляється здобувачеві вищої освіти, який не може продовжити навчання без додаткової роботи з вивчення навчальної дисципліни.
-------------------------------	---

Структурування дисципліни за видами навчальної роботи і оцінювання результатів навчання студентів денної форми здобуття освіти у семестрі

Аудиторна робота												Контрольні заходи	Семестровий контроль
Лабораторні заняття №:								Практичні заняття (мінімум 2 контрольні точки)				Тестовий контроль	Залік
1*	2	3	4	5	6	7	8	1	2	3	4	T1	
Кількість балів за кожний вид навчальної роботи (мінімум-максимум)													
3-5	3-5	3-5	3-5	3-5	3-5	3-5	3-5	6-10	6-10	6-10	6-10	12-20	За рейтингом
24-40								24-40				12-20	60-100**

Примітки: *За набрану з будь-якого виду навчальної роботи з дисципліни кількість балів, нижче встановленого мінімуму, здобувач отримує незадовільну оцінку і має її перездати у встановлений викладачем (деканом) термін. Інституційна оцінка встановлюється відповідно до таблиці «Співвідношення інституційної шкали оцінювання і шкали оцінювання ЄКТС».

Оцінювання на практичних заняттях

Оцінка, яка виставляється за практичне заняття, складається з таких елементів: усне опитування студентів на знання теоретичного матеріалу з теми; вільне володіння студентом термінологією і уміння професійно обґрунтувати прийняті рішення при розв'язуванні задач; результати самостійних робіт.

При оцінюванні практичного заняття викладач керується узагальненими критеріями, наведеними у таблиці «Критерії оцінювання навчальних досягнень здобувача вищої освіти» (мінімальний позитивний бал – 6 балів, максимальний – 10 балів).

Оцінювання результатів захисту лабораторної роботи

Виконана й оформлена відповідно до встановлених Методичними рекомендаціями вимог лабораторна робота комплексно оцінюється викладачем при її захисті з урахуванням таких критеріїв: самостійність та правильність виконання; повнота відповіді та знання методики виконання та фахової термінології з дисципліни тощо.

Результат виконання і захисту здобувачем вищої освіти кожної лабораторної роботи оцінюється відповідно до таблиці Критеріїв оцінювання навчальних досягнень здобувача вищої освіти.

У випадку виявлення здобувачем рівня знань, нижчого ніж 60 відсотків від максимального балу, встановленого Робочою програмою для кожної структурної одиниці, лабораторна робота йому **не зараховується** і для її захисту він має детальніше опрацювати матеріал з теми роботи, методику її виконання, виправити грубі помилки та повторно вийти на її захист у призначений для цього викладачем час.

Оцінювання результатів тестового контролю

Кожний з тестів, передбачених Робочою програмою, складається із 20 тестових завдань, кожне з яких є рівнозначним.

Відповідно до таблиці структурування видів робіт за тестовий контроль здобувач залежно від кількості правильних відповідей може отримати від 12 до 20 балів.

Розподіл балів в залежності від наданих правильних відповідей на тестові завдання

Кількість правильних відповідей	0-11	12-13	14	15-16	17	18-20
Відсоток правильних відповідей	0-59	60-65	66-72	73-82	83-89	90-100
Кількість отриманих балів	0	12	14	16	18	20

На тестування відводиться 30-40 хвилин. Студент проходить тестування в он-лайн режимі у Модульному середовищі для навчання. Також, студент може проходити тестування письмово, записуючи правильні відповіді у талоні відповідей. При отриманні негативної оцінки тест слід перездати до терміну **наступного** контролю.

Підсумкова семестрова оцінка за інституційною шкалою і шкалою ЄКТС визначається в автоматизованому режимі після внесення викладачем результатів оцінювання у балах з усіх видів навчальної роботи до електронного журналу. Співвідношення інституційної шкали оцінювання і шкали оцінювання ЄКТС наведені нижче у таблиці «Співвідношення».

Семестровий залік виставляється на останньому занятті за умови якщо загальна сума балів, яку накопичив здобувач з дисципліни (іншого освітнього компонента) за результатами **поточного** контролю, знаходиться у межах від 60 до 100 балів. При цьому за інституційною шкалою ставиться оцінка «**зараховано**», а за шкалою ЄКТС – буквене позначення оцінки, що відповідає набраній студентом сумі балів відповідно до таблиці Співвідношення. Присутність здобувача у цьому випадку не є обов'язковою.

Таблиця – Співвідношення інституційної шкали оцінювання і шкали оцінювання ЄКТС

Оцінка ЄКТС	Рейтингова шкала балів	Інституційна оцінка (рівень досягнення здобувачем вищої освіти запланованих результатів навчання з навчальної дисципліни)	
		Залік	Іспит/диференційований залік
A	90-100	Зараховано	Відмінно/Excellent – високий рівень досягнення запланованих результатів навчання з навчальної дисципліни, що свідчить про безумовну готовність здобувача до подальшого навчання та/або професійної діяльності за фахом
B	83-89		Добре/Good – середній (максимально достатній) рівень досягнення запланованих результатів навчання з навчальної дисципліни та готовності до подальшого навчання та/або професійної діяльності за фахом
C	73-82		Задовільно/Satisfactory – Наявні мінімально достатні для подальшого навчання та/або професійної діяльності за фахом результати навчання з навчальної дисципліни
D	66-72		
E	60-65		
FX	40-59	Незараховано	Незадовільно/Fail – Низка запланованих результатів навчання з навчальної дисципліни відсутня. Рівень набутих результатів навчання є недостатнім для подальшого навчання та/або професійної діяльності за фахом
F	0-39		Незадовільно/Fail – Результати навчання відсутні

10. Питання для самоконтролю результатів навчання

1. Що таке веб-візуалізація та які її основні завдання?
2. Що розуміють під терміном «веб-орієнтована система»?
3. У чому полягає різниця між клієнтською та серверною частинами веб-додатку?
4. Які основні етапи обміну даними в архітектурі «клієнт–сервер»?
5. Що таке протокол HTTP і чим він відрізняється від HTTPS?
6. Наведи приклади сучасних веб-систем моніторингу або керування (SCADA, IoT тощо).
7. Які основні типи даних є в мові Python?
8. Як створити та використовувати функції у Python?
9. Що таке бібліотека та пакет у Python, як їх встановлювати?
10. Для чого використовується `pip` і `env`?
11. Як відбувається робота з файлами у Python?
12. Назви популярні бібліотеки Python для веб-розробки.
13. Що таке Flask і чому його називають «мікрофреймворком»?
14. Як створити простий веб-сервер на Flask?
15. Для чого у Flask використовується маршрутизація (routing)?
16. Що таке шаблонізація і яку роль відіграє Jinja2?
17. Як Flask обробляє HTTP-запити GET та POST?
18. Яка основна структура HTML-документа?
19. Для чого використовується CSS у веб-розробці?
20. Як динамічно вставляти дані з Flask у HTML через Jinja2?
21. Що таке Bootstrap і які його основні переваги?
22. Як зробити веб-інтерфейс адаптивним під різні пристрої?
23. Як у Flask реалізується обробка HTML-форм?
24. Для чого використовується об'єкт `request` у Flask?
25. Як передати дані між сторінками у Flask?
26. Що таке `session` і `redirect`, і для чого вони потрібні?
27. Наведи приклад простої форми керування параметрами у Flask.
28. Що таке SQL і для чого використовується SQLite?
29. Які основні операції CRUD і що вони означають?
30. Що таке ORM і як працює SQLAlchemy?
31. Як відображати дані з бази у веб-інтерфейсі Flask?
32. Як у Flask створити таблицю бази даних через SQLAlchemy?
33. Які бібліотеки Python використовуються для побудови графіків (Matplotlib, Plotly, Chart.js)?
34. Як передати дані з Flask у JavaScript для побудови графіка?
35. Чим відрізняються статичні та динамічні графіки?
36. Як створити Flask-додаток для моніторингу температури?
37. Що таке AJAX і як він використовується для оновлення даних на сторінці?
38. Яку роль відіграє WebSocket у веб-системах реального часу?
39. Що таке REST API і як його створити у Flask або FastAPI?
40. У якому форматі зазвичай передаються дані через API (JSON, XML)?
41. Як читати дані з зовнішнього API у Python?
42. Які основні протоколи обміну даними між ПК і мікроконтролерами (Serial, Wi-Fi)?
43. Як працює бібліотека `pyserial`?

44. Як отримати дані з Arduino у Flask-додатку?
45. Як можна емулювати дані з Arduino у Tinkercad Circuits?
46. Що таке SCADA і як реалізується подібна архітектура у Flask?
47. Як реалізувати потік даних у Flask за допомогою WebSocket (Flask-SocketIO)?
48. Як відображати оновлення даних у браузері без перезавантаження сторінки?
49. Які основні загрози для веб-додатків (CSRF, SQL Injection)?
50. Як відбувається розгортання Flask-додатку на Heroku, Render або PythonAnywhere?

11. Навчально-методичне забезпечення

Освітній процес з дисципліни «Веб-візуалізація та веб-орієнтовані системи керування» повністю і в достатній кількості забезпечений необхідною навчально-методичною літературою.

12. Матеріально-технічне та програмне забезпечення дисципліни (за потреби)

Інформаційна та комп'ютерна підтримка: ПК, планшет, смартфон або інший мобільний пристрій, проектор. Програмне забезпечення: програми Microsoft Office або аналогічні, доступ до мережі Інтернет, робота з презентаціями.

13. Рекомендована література:

Основна:

1. Коваль Ю. В., Ставровський А. Б. Інформаційні мережі: навчальний посібник. Київ, 2021. 84 с
2. Васильєв О.М. Програмування в Python. Теорія і практика : навч. посіб. Київ : Видавництво Ліра-К, 2023. 462 с. ISBN 978-617-520-513-6.
3. Готинчан Т.І. Основи веброзробки: HTML і CSS. Частина 1 : навчальний посібник. Чернівці : Чернівецький національний університет, 2023. 208 с.
4. Доценко С. І. Організація та системи керування базами даних: Навч. посібник. Харків: УкрДУЗТ, 2023. 117 с.
5. Томка Ю.Я., Талах М.В., Ушенко Ю.О. Python та Django Full Stack веб-розробка. Чернівці: Чернівецький нац. ун-т ім. Ю.Федьковича, 2022. 248с.
6. Бен С. Python. Збірник вправ. Введення в мову Python із завданнями та рішеннями. Харків, 2021. 238 с.
7. Висоцька В., Оборська О. Python. Алгоритмізація та програмування. Н.Світ 2000. 2021. 514 с.
8. Маттес Е. Пришвидшений курс Python. Львів.: Видавництво Старого Лева, 2021. 600 с.
9. Чичкарьов Є.А., Зінченко О.В., Єльченко С.В. Прикладне програмування на Python. Частина 1. Основи програмування на Python. Навчальний посібник. Київ: ДУТ, 2022. 160 с.
10. Палагін В. В., Палагіна О. А., Зорін О. С. Основи Python та програмування електронних систем : навч. посіб. Черкаси : ЧДТУ, 2024. 216 с.
11. SCADA-системи. Практикум. Частина 2 [Електронний ресурс] : навчальний посібник / уклад.: В. Г. Здоренко, С. М. Лісовець. Київ : КПІ ім. Ігоря Сікорського, 2024. 52 с. Назва з екрана.
12. Hoffman A. Web Application Security Exploitation and Countermeasures for Modern Web Application 2021, 328 p.

Додаткова:

1. Gutttag John V. Introduction to Computation and Programming Using Python with Application to Understanding Data. 2016. 591 p
2. Яковенко А. В. Основи програмування. Python. Частина 1. Київ: КПІ ім. Ігоря Сікорського, 2018. 195 с.
3. Крєневич А.П. Python у прикладах і задачах. Частина 2. Об'єктно-орієнтоване програмування. Навчальний посібник. К.: ВПЦ "Київський Університет", 2020. 152 с.
4. Титенко С. В. Web-орієнтовані інформаційні системи. Методичні вказівки до вивчення дисципліни для студентів напряму підготовки 6.050103 «Програмна інженерія». К.: НТУУ «КПІ» 2015. 51с.

14. Інформаційні ресурси

1. Модульне середовище для навчання. URL : <https://msn.khmnmu.edu.ua/>
2. Електронна бібліотека ХНУ. URL: <http://library.khmnmu.edu.ua/>
3. Інституційний репозитарій ХНУ. URL : <https://elar.khmnmu.edu.ua/home>

Тип дисципліни	Вибіркова
Рівень вищої освіти	Перший (бакалаврський)
Мова викладання	Українська
Семестр	—
Кількість призначених кредитів ЄКТС	8,0
Форми здобуття освіти, для яких викладається дисципліна	Очна (денна)

Результати навчання. Після вивчення дисципліни студент повинен: *знати* основні принципи побудови веб-орієнтованих систем керування, архітектуру клієнт–серверних застосунків, принципи взаємодії через протоколи HTTP і WebSocket; *розуміти* структуру та можливості фреймворків Flask і бібліотек для веб-візуалізації; *володіти* базовими інструментами Python для розроблення, налагодження та тестування веб-застосунків; *уміти* створювати веб-інтерфейси для моніторингу та керування процесами, реалізовувати динамічне оновлення даних у режимі реального часу; *застосовувати* набуті знання для інтеграції веб-додатків із мікроконтролерами, базами даних і хмарними платформами; *обґрунтовувати* вибір технологій, архітектури та інструментів під час реалізації проєктів; *поєднувати* теоретичні знання з практичними навичками програмування для розв’язання прикладних інженерних задач; *демонструвати* здатність працювати з технічною документацією, проводити аналіз результатів, аргументовано захищати технічні рішення; *виявляти* ініціативність, здатність до самостійного навчання, ефективної комунікації та роботи в команді під час розроблення веб-орієнтованих систем керування.

Зміст навчальної дисципліни. Вступ до веб-розробки та основи Python. Створення веб-додатків у Flask. Робота з даними у веб-застосунках. Веб-візуалізація даних. API та взаємодія із зовнішніми джерелами даних. Веб-орієнтовані системи керування. Безпека та хмарні технології. Розгортання та підсумковий проєкт.

Запланована навчальна діяльність: Мінімальний обсяг навчальних занять в одному кредиті ЄКТС навчальної дисципліни для *першого* (бакалаврського) рівня вищої освіти за денною формою здобуття освіти становить 10 годин на 1 кредит ЄКТС.

Форми (методи) навчання: лекції (з використанням методів візуалізації, проблемного й інтерактивного навчання, мотиваційних прийомів, інформаційно-комунікаційних технологій); лабораторні заняття (з використанням методів комп’ютерного моделювання, методів проєктної діяльності, тренінгових вправ, аналіз проблемних ситуацій, пояснення, дискусія тощо), практичні заняття (з використанням інструктування, демонстрування, розв’язування типових і прикладних задач, аналізу кейсів, ситуаційних завдань, елементів дискусії тощо); самостійна робота (опрацювання теоретичного матеріалу, підготовка до виконання практичних робіт, поточного та підсумкового контролю, виконання індивідуальних та домашніх завдань), з використанням інформаційно-комп’ютерних технологій та технологій дистанційного навчання.

Форми оцінювання результатів навчання: оцінювання лабораторних та практичних робіт; виконання; тестування.

Вид семестрового контролю: залік.

Навчальні ресурси:

1. Васильєв О.М. Програмування в Python. Теорія і практика : навч. посіб. Київ : Видавництво Ліра-К, 2023. 462 с. ISBN 978-617-520-513-6
2. Томка Ю.Я., Талах М.В., Ушенко Ю.О. Python та Django Full Stack веб-розробка. Чернівці: Чернівецький нац. ун-т ім. Ю.Федьковича, 2022. 248с.
3. Бен С. Python. Збірник вправ. Введення в мову Python із завданнями та рішеннями. Харків, 2021. 238 с.
4. Висоцька В., Оборська О. Python. Алгоритмізація та програмування. Н.Світ 2000. 2021. 514 с.
5. Маттес Е. Пришвидшений курс Python. Львів.: Видавництво Старого Лева, 2021. 600 с.
6. Чичкарьов Є.А., Зінченко О.В., Єльченко С.В. Прикладне програмування на Python. Частина 1. Основи програмування на Python. Навчальний посібник. Київ: ДУТ, 2022. 160 с.
7. Палагін В. В., Палагіна О. А., Зорін О. С. Основи Python та програмування електронних систем : навч. посіб. Черкаси : ЧДТУ, 2024. 216 с.
8. Модульне середовище для навчання. Доступ до ресурсу: <https://msn.khmnu.edu.ua/>
9. Електронна бібліотека ХНУ. Доступ до ресурсу: <http://library.khmnu.edu.ua/>

Викладачі: канд. техн. наук, доцент Корецька Л.О.