

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЧЕРКАСЬКИЙ ХУДОЖНЬО-ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ
Відділення механіко-технологічне

	СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ "Розробка клієнт-серверних застосувань" Освітньо-професійної програми: <u>«Обслуговування програмних систем і комплексів»</u> (назва освітньо-професійної програми) Спеціальність <u>F3 «Комп'ютерні науки»</u> (шифр та назва спеціальності) Галузь знань: <u>F «Інформаційні технології»</u> (шифр та назва галузі знань) Освітньо-професійний ступінь: фаховий молодший бакалавр
Циклова комісія	Комп'ютерних та землевпорядних дисциплін
Курс, семестр	IV курс, VII, VIII семестр
Обсяг	6 кредитів ЄКТС – 180 годин. Аудиторні заняття: лекційні – 40 год., практичні - 50 год., самостійне опрацювання студентів – 90 год.
Мова викладання	українська
Вимоги до початку вивчення	Необхідні знання та навички з дисциплін «Алгоритми та структури даних» «Бази даних» «Основи веб-розробки»
Відповідність компетентностей та програмних результатів навчання	Загальні компетентності (ЗК): ЗК2. Здатність зберігати та примножувати моральні, культурні, наукові цінності і досягнення суспільства на основі розуміння історії та закономірностей розвитку предметної області, її місця у загальній системі знань про природу і суспільство та у розвитку суспільства, техніки і технологій, використовувати різні види та форми рухової активності для активного відпочинку та ведення здорового способу життя. ЗК3. Здатність до абстрактного мислення, аналізу та синтезу. ЗК4. Здатність застосовувати знання у практичних ситуаціях. ЗК5. Знання та розуміння предметної області та розуміння професійної діяльності. ЗК6. Здатність спілкуватися державною мовою як усно, так і письмово. ЗК7. Здатність спілкуватися іноземною мовою. ЗК9. Здатність до пошуку, оброблення та аналізу інформації з різних джерел. Результати навчання (РН): РН05. Розуміти основні методи і технології об'єктно-орієнтованого та компонентного програмування. РН08. Розробляти застосунки, використовуючи сучасні веб-технології. РН10. Знати методології, методи, моделі, процеси і технології життєвого циклу розробки та тестування програмного забезпечення. РН11. Застосовувати сучасні мови програмування та технології для розробки програмного забезпечення розподілених систем. РН15. Розробляти супровідну документацію на різних етапах процесу життєвого циклу розробки програмного забезпечення. Спеціальні компетентності (СК) СК1. Здатність використовувати основні поняття, ідеї та методи фундаментальних наук під час розв'язання складних спеціалізованих задач з комп'ютерних наук в галузі інформаційних технологій. СК2. Здатність використовувати теоретичні та фундаментальні знання в галузі комп'ютерних наук та інформаційних технологій для вирішення різноманітних проблем.

	СК4. Здатність здійснювати проєктування та розробку програмного забезпечення. СК5. Здатність застосовувати принципи і методи побудови та використання мережових технологій. СК6. Здатність застосовувати методи та засоби захисту програмного забезпечення та даних від несанкціонованого доступу в умовах супроводження та експлуатації програмних систем і комплексів. СК7. Здатність проєктувати, розробляти та обслуговувати веб-застосунки з динамічним контентом, використовуючи веб-технології, технології комп'ютерної графіки та анімації. СК8. Здатність застосовувати сучасні методи, технології та інструментальні засоби проєктування й створення програмних систем та їх супроводження. СК10. Здатність адмініструвати системне та прикладне програмне забезпечення під час реалізації процесів життєвого циклу інформаційних систем. СК11. Здатність застосовувати методи та техніки тестування програмного забезпечення впродовж життєвого циклу розробки програмних систем.
Місце дисципліни в освітньо-професійній програмі	Дисципліна «Розробка клієнт-серверних застосувань» належить до циклу професійної та практичної підготовки освітньо-професійної програми спеціальності F3 «Комп'ютерні науки». Вона забезпечує формування базових теоретичних знань та практичних навичок, необхідних для створення клієнтських і серверних частин програмних систем, роботи з базами даних та побудови розподілених інформаційних сервісів. Зміст навчального матеріалу дисципліни спрямований на розвиток у студентів логічного й аналітичного мислення, умінь проєктувати структуру застосунків, організовувати взаємодію між компонентами, розробляти сервери, мережові модулі та клієнтські інтерфейси, а також вирішувати прикладні задачі у сфері програмування та інформаційних технологій. Вивчення курсу «Розробка клієнт-серверних застосувань» є обов'язковим етапом підготовки фахівця з комп'ютерних наук, оскільки надає фундаментальні компетенції для роботи з сучасними програмними платформами, СУБД, мережевими протоколами, веб-технологіями та мобільними системами.
Що буде вивчатися	У рамках дисципліни "Розробка клієнт-серверних застосувань" студенти вивчатимуть: Основи архітектури клієнт-серверних систем, включаючи моделі взаємодії, принципи розподілу ролей між клієнтом і сервером, поняття сервісів, протоколів обміну даними та модель багаторівневих застосунків. Окрему увагу буде приділено технологіям створення серверних застосунків, роботі з базами даних, обробці запитів та організації віддаленого доступу до даних. Студенти ознайомляться з особливостями роботи середовищ .NET та Java, навчатися створювати клієнтські інтерфейси, мережові модулі, сервери застосунків, а також мобільні застосунки, які взаємодіють із сервером. Курс охоплює принципи обміну даними через протоколи TCP, UDP, HTTP, роботу з REST-сервісами, RMI та іншими технологіями взаємодії у розподілених системах. Також розглядатимуться основи безпеки клієнт-серверних систем, авторизації, автентифікації та захисту передачі даних. Розділ 1. Основи архітектури клієнт-сервер та програмування Java Поняття клієнт-серверної архітектури Типи моделей клієнт-сервер (двошарова, тривірнева, багаторівнева) Ролі клієнта, сервера застосунків і сервера баз даних Основи мови Java Алгоритмічні структури Java Посилальні типи даних Робота з файлами та колекціями Потоки виконання та синхронізація в Java Створення графічних інтерфейсів (AWT/JavaFX) Основи анімації, робота з растровою графікою Обробка подій та створення прикладних Java-додатків Розділ 2. Мережеве програмування засобами .NET та Java

	Програмування клієнт-серверної взаємодії за протоколом TCP/IP Створення клієнтських і серверних застосунків на основі TCP Обмін повідомленнями через UDP Технологія RMI (Remote Method Invocation) у Java Основи розподілених об'єктів Вступ до Java EE Створення веб-застосунків на основі Servlets Робота з мережею у .NET (TcpClient, TcpListener, HttpClient) Моделювання мережевої взаємодії та налагодження застосунків Розділ 3. Розробка мережевих застосунків для мобільних платформ Встановлення та налаштування Android Studio Створення першого Android-застосунку Життєвий цикл Activity Локалізація, стильове оформлення та робота з кольорами Макети, анімації, структура інтерфейсу Android Робота з віджетами та елементами керування Наміри (Intent), міжкранна взаємодія Підключення Android-клієнтів до серверів Отримання та надсилання даних через HTTP/HTTPS Робота з JSON, REST-API, мережевими бібліотеками Розділ 4. Серверні технології, бази даних та веб-застосунки Архітектура серверів застосунків Робота з базами даних: SQL, підключення, запити, транзакції Фізичне проєктування БД Створення серверної логіки Технології побудови REST-сервісів Сесії, cookies, ідентифікація клієнтів Основи безпеки клієнт-серверних систем Авторизація, автентифікація, робота з токенами Впровадження HTTPS Створення веб-клієнтів і веб-серверів
Чому це цікаво/треба вивчати	Дисципліна «Розробка клієнт-серверних застосувань» є цікавою, тому що дозволяє зрозуміти, як працюють сучасні програмні системи, які ми використовуємо щодня — від простих мобільних додатків до масштабних веб-сервісів, інтернет-магазинів, хмарних платформ та корпоративних систем. Вивчаючи клієнтську й серверну частину застосунків, студенти відкривають для себе принципи обробки даних, мережевої взаємодії, сервіс-орієнтованої архітектури та побудови розподілених систем. Курс дає можливість побачити, як відбувається передача даних між клієнтом і сервером, як працюють API, як організовано бекенд великих онлайн-платформ, соціальних мереж, банківських систем і мобільних застосунків. Цікавість дисципліни також полягає у широких можливостях практичного застосування: від створення власних програм і веб-сайтів до розробки професійних серверних сервісів, мобільних клієнтів, інтеграції з базами даних та побудови безпечних мережевих застосунків. Студенти отримують можливість працювати з сучасними технологіями .NET, Java, Android, мережевими протоколами та СУБД, спостерігаючи, як теоретичні концепції перетворюються на практичні навички, необхідні в сучасній IT-галузі.
Чому можна навчатися	Знання з дисципліни «Розробка клієнт-серверних застосувань» можна буде використовувати для створення, розгортання та підтримки сучасних програмних систем, які працюють за моделлю клієнт-сервер. Студенти зможуть розробляти клієнтські та серверні програми, взаємодіяти з базами даних, проєктувати API, працювати з мережевими протоколами та реалізовувати обмін даними між компонентами застосунків. У сфері програмної інженерії здобуті навички дозволять створювати веб-сервіси, мобільні застосунки, бекенд-системи, мікросервіси та розподілені сервіси для різних галузей — від фінансових технологій до електронної комерції та корпоративних платформ. Студенти навчатися застосовувати сучасні технології .NET, Java, Android та SQL/NoSQL СУБД, забезпечувати безпеку застосунків, оптимізувати роботу серверів та підтримувати ефективну взаємодію між клієнтом і сервером.

Як користуватися набутими знаннями/уміннями	Дисципліна «Розробка клієнт-серверних застосувань» охоплює фундаментальні принципи побудови, роботи та взаємодії програмних систем, що складаються з клієнтських і серверних компонентів. У рамках курсу студенти вивчатимуть архітектуру клієнт-серверних рішень, моделі розподілу функцій між компонентами застосунку, механізми обміну даними, а також основи роботи з серверами застосунків, веб-технологіями та сучасними СУБД. Окрему увагу буде приділено принципам створення серверної логіки, проєктуванню API, роботу з протоколами TCP/IP та HTTP, а також впровадженню мережесервісів і веб-застосунків. Ця дисципліна є ключовою для підготовки фахівців у сфері інформаційних технологій, оскільки поєднує в собі знання про принципи роботи серверних технологій, клієнтських інтерфейсів та засобів організації взаємодії між ними. Студенти здобудуть навички створення клієнтських і серверних застосунків, роботи з базами даних, побудови REST- і мережесервісів, а також застосування сучасних рішень для забезпечення надійності, масштабованості та продуктивності клієнт-серверних систем. Курс також дозволить навчитися розробляти застосунки для мобільних платформ, інтегрувати їх із серверними частинами та оптимізувати взаємодію в розподіленому середовищі.
Заняття	Лекційні заняття: формування теоретичних знань щодо архітектури клієнт-серверних систем, моделей взаємодії, принципів розробки серверних та клієнтських застосунків, роботи з базами даних, мережесервісів та технологій .NET, Java і Android. Використання мультимедійних матеріалів, демонстрація роботи клієнт-серверних систем, обговорення актуальних технічних рішень та підходів у галузі розподілених застосунків. Практичні заняття: закріплення теоретичного матеріалу через розробку реальних клієнтських і серверних програм, створення TCP/UDP застосунків, роботу з REST API, створення веб-сервлетів, виконання запитів до СУБД, розробку Android-клієнтів, тестування та налагодження мережесервісів. Лабораторні заняття: застосування отриманих знань під час створення повноцінних клієнт-серверних рішень: розробка серверної логіки, організація обміну даними, реалізація багаторівневої архітектури, побудова API, інтеграція з базами даних, створення невеликих веб- і мобільних застосунків. Інтерактивні заняття: використання навчальних платформ (Google Classroom, Kahoot, Quizizz) для інтерактивних тестів, задач, перевірки коду, а також платформ онлайн-розробки (Replit, GitHub Classroom) для спільної роботи над клієнтськими та серверними проєктами.
Інформаційне забезпечення	Рекомендовані джерела інформації: Keri Searra, Bert Beyer. Head First Java. — Фабула, 2022. — 720 с. Васильев О. Програмування мовою Java. — Навчальна книга—Богдан, 2020. — 696 с. Herbert Schildt. Java: The Complete Reference. — McGraw-Hill, останнє видання. Mark Richards, Neal Ford. Fundamentals of Software Architecture. — O'Reilly, 2020. Adam Bien. Real World Java EE Patterns. — Prentice Hall, 2019. Andrew Troelsen, Philip Japikse. Pro C# and .NET. — Apress, останнє видання. Microsoft Docs — ASP.NET Core, Web API, Entity Framework Core (онлайн документація). Android Developers Documentation — Android Studio, Jetpack, Networking API. Martin Fowler. Patterns of Enterprise Application Architecture. — Addison-Wesley, 2011. RESTful Web APIs — Leonard Richardson, Mike Amundsen, Sam Ruby. — O'Reilly, 2013. Допоміжна література: Robert C. Martin. Clean Code: A Handbook of Agile Software Craftsmanship. — Prentice Hall, 2008. Robert C. Martin. Clean Architecture. — Prentice Hall, 2017. Kyle Loudon. Mastering Algorithms with C / Java / Python (на вибір, підходить для структур даних і логіки застосунків).

	Steven Feuerstein. Oracle PL/SQL Programming. (для роботи з SQL). W. Richards Stevens. UNIX Network Programming. — Addison-Wesley. Joshua Bloch. Effective Java. — Addison-Wesley, 2018. Tom Hombergs. Get Your Hands Dirty on Clean Architecture. — 2019. Nate Murray, Felipe Coury. Fullstack React / Vue / Angular (для клієнтської частини складних застосунків). Документація MySQL / PostgreSQL / SQLite. GitHub Guides — інструменти командної розробки.
Поточний контроль	Поточний контроль здійснюється викладачем у формі усного спілкування зі студентами, письмового, тестового експрес-контролю (усні відповіді, розв'язання завдань студентами за ПК та на місцях, самостійні роботи, доповіді, презентації виконаних завдань, досліджень, тести тощо) і має за мету перевірку ступеня засвоєння певного навчального матеріалу, а також рівня оволодіння вміннями та навичками.
Семестровий контроль	Підсумковий контроль здійснюється у формі екзамену

