**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ**

**"ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ЛУЦЬКОГО НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ"**

**Циклова комісія комп’ютерних систем та інформаційних технологій**

|  |  |
| --- | --- |
|  | **ЗАТВЕРДЖУЮ**  Заступник директора  з навчальної роботи  \_\_\_\_\_\_\_\_\_\_\_С. БУСНЮК  “29” серпня 2024 року |

ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

**"КЛІЄНТ-СЕРВЕРНА ВЗАЄМОДІЯ"**

Розробник П. ВОВК

Галузь знань 12 Інформаційні технології

Код та назва спеціальності 126 Інформаційні системи та технології, 123 Комп’ютерна інженерія

Освітньо-професійна програма Інформаційні системи та технології, Комп’ютерна інженерія

Статус навчальної дисципліни вибіркова

Мова навчання українська

2024 рік

Програма навчальної дисципліни «Клієнт-серверна взаємодія» для здобувачів фахової передвищої освіти освітньо-професійного ступеня фаховий молодший бакалавр IV курсу спеціальностей 126 Інформаційні системи та технології та 123 Комп'ютерна інженерія денної форми навчання складена на основі ОПП Інформаційні системи та технології та ОПП Комп'ютерна інженерія.

“26” серпня 2024 року - 16 с.

Розробник:Вовк П.Б.

Програма обговорена та схвалена на засіданні циклової комісії комп’ютерних систем та інформаційних технологій

Протокол від “ 26 ” серпня 2024 року № 1

Голова циклової комісії комп’ютерних систем та інформаційних технологій \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ П. ВОВК

Схвалено Педагогічною радою ТФК ЛНТУ

Протокол від “29” серпня 2024 року №1

Робоча програма обговорена та схвалена на засіданні циклової комісії комп’ютерних систем та інформаційних технологій

Протокол від “ ” серпня 202\_\_ року № \_\_\_\_

Голова циклової комісії комп’ютерних систем та інформаційних технологій \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

Схвалено Педагогічною радою ТФК ЛНТУ

Протокол від “ ” серпня 202 року №\_\_\_

Робоча програма обговорена та схвалена на засіданні циклової комісії комп’ютерних систем та інформаційних технологій

Протокол від “ ” серпня 202\_\_ року № \_\_\_\_

Голова циклової комісії комп’ютерних систем та інформаційних технологій \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

Схвалено Педагогічною радою ТФК ЛНТУ

Протокол від “ ” серпня 202 року №\_\_\_

**1. ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

|  |  |  |
| --- | --- | --- |
| **Найменування показників** | **Галузь знань, спеціальність,  освітньо - професійний ступінь** | **Характеристика навчальної дисципліни** |
| Тем – 2 | **Галузь знань:**  12 Інформаційні технології | **Форма навчання** |
| денна |
| **Спеціальність:**  126 Інформаційні системи та технології  123 Комп’ютерна інженерія |
| **Рік підготовки** |
| ІV |
| **Семестр** |
| Загальна кількість годин – 150 | VІI |
| Для денної форми навчання:  аудиторних – 60;  самостійної роботи студента – 90. | **Освітньо-професійний ступінь:**  фаховий молодший бакалавр | **Лекції** |
| 30 год |
| **Практичні** |
| 30 год |
| **Самостійна робота** |
| 90 год |
| **Вид контролю:** |
| диф. залік |

|  |  |
| --- | --- |
| **2. Мета дисципліни, передумови її вивчення та заплановані результати навчання** | |
| Місце дисципліни в освітній програмі: | В сучасному світі взаємодія клієнт-сервер відіграє важливу роль у функціонуванні інформаційних систем. Ця взаємодія базується на принципі розподілених обчислень, де клієнт (користувач) і сервер (потужний комп’ютер або програма) спілкуються один з одним через мережу. Взаємодія клієнт-сервер є основою багатьох сучасних технологій, таких як веб-сайти, електронна пошта, мобільні додатки та інші.  Мета викладання навчальної дисципліни “ Клієнт-серверна взаємодія ” полягає у формування у здобувачів фахової передвищої освіти знань, вмінь та навичок, необхідних для розуміння, проектування та реалізації систем, що базуються на клієнт-серверній архітектурі. Ця дисципліна спрямована на розвиток здатності студентів аналізувати та розробляти ефективні рішення для взаємодії між клієнтськими та серверними компонентами, враховуючи вимоги до масштабованості, надійності, безпеки та продуктивності.  Програму орієнтовано на формування у студентів системного мислення, здатності працювати в команді, вирішувати складні технічні проблеми та адаптуватися до швидко змінюваних технологій у сфері інформаційних технологій. |
| Компетентності загальні або фахові: | ЗК5. Знання та розуміння предметної області і професійної діяльності.  ЗК6. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.  ЗК7. Здатність застосовувати знання у практичних ситуаціях.  СК1. Здатність до алгоритмічного та логічного мислення.  СК2. Здатність накопичувати, обробляти та систематизувати професійні знання в області інформаційних систем та технологій, усвідомлювати важливість навчання протягом усього життя.  СК3. Здатність застосовувати фундаментальні та міждисциплінарні знання для успішного розв’язання завдань у галузі інформаційних систем та технологій.  СК5. Здатність використовувати та управляти сучасними інформаційно-комунікаційними системами та технологіями.  СК7. Здатність використовувати методи і засоби забезпечення інформаційної та функціональної безпеки в області інформаційних систем та технологій.  СК9. Здатність вибору, розгортання, інтегрування, адміністрування та супроводу інформаційних систем та технологій.  СК10. Здатність брати участь в проектуванні, розробці, налагодженні та удосконаленні компонентів інформаційних систем.  СК14. Здатність вибору хмарних платформ та розгортання на них, інтегрування, адміністрування і супроводу інформаційних систем та сервісів.  СК15. Здатність брати участь в проектуванні, розгортанні, налагодженні та удосконаленні компонентів хмарних систем та сервісів. |
| Програмні результати навчання: | РН1. Аналізувати отримане завдання та розробляти алгоритм його вирішення з використанням сучасних інформаційних систем та технологій  РН2. Накопичувати, систематизувати та узагальнювати інформацію про підходи й методики роботи в області інформаційних систем та технологій.  РН3. Знаходити аналогії та застосовувати знання, вміння та навички з суміжних дисциплін для формування та розв’язання професійних задач в області інформаційних систем та технологій.  РН4. Знати способи ідентифікації, формулювання та класифікації вимог до інформаційних систем та технологій і застосовувати їх при прийнятті бізнес-рішень та в процесі аналізу отриманого технічного завдання.  РН5. Обирати оптимальний спосіб вирішення завдань, налаштовувати та користуватись відповідними інструментальними засобами.  РН9. Здійснювати обґрунтований вибір, розгортання, інтегрування, адміністрування й технічну підтримку інформаційних систем та технологій.  РН10. Розробляти та супроводжувати окремі компоненти інформаційних систем.  РН15. Комунікувати з професійних питань українською та іноземною мовою.  РН16. Знати та розуміти предметну область, застосовувати знання у професійній діяльності.  РН17. Аналізувати та узагальнювати необхідну інформацію з різних джерел та ресурсів для вирішення професійних задач з урахуванням сучасних досягнень науки і техніки.  РН18. Здійснювати обґрунтований вибір хмарної платформи та хмарних сервісів.  РН19. Здійснювати розгортання, інтегрування, адміністрування й технічну підтримку хмарних систем та сервісів.  РН20. Розробляти, налагоджувати та супроводжувати окремі компоненти хмарних систем та сервісів. |
| **Передумови для вивчення дисципліни:** | |
| Для вивчення навчальної дисципліни «Клієнт-серверна взаємодія» необхідними є компетентності здобувачів освіти з навчальних дисциплін «Основи програмування», «Комп'ютерні мережі», «Комп'ютерні системи», "Web технології", "Основи інтернету речей (ІОТ)". Навчальна дисципліна забезпечує міжпредметні зв’язки з навчальними дисциплінами «Адміністрування комп’ютерних систем та мереж», «Операційні системи», «Організація баз даних та сховища даних (КР)», «Контролери та сенсори систем ІОТ», " Хмарні платформи та сервіси". | |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **3. Обсяг та структура програми навчальної дисципліни** | | | | | | | | | | | |
| **форма навчання** | | Кредити ЄКТС | **денна (очна)** | | | | | | | |
| **ФОРМА Контролю** | | **Підсумкові оцінки (диф. залік)** | | | | | | | |
| № теми | Назва теми | Кількість годин: | | | | | | | |
| Разом | Самостійна робота | Навчальні заняття: | | | | | |
| Всього | з них: | | | | |
| Лекційні заняття | Семінарські заняття | Практичні заняття | Лабораторні заняття | Індивідуальні заняття |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** |
| 1. | Клієнт – серверна архітектура | 1,9 |  |  |  | 6 |  |  |  |  |
| 2. | Клієнт – серверні технології | 2,1 |  |  |  | 24 |  |  |  |  |
| **Разом з дисципліни:** | | **5** | **150** | **52** | **68** | **30** |  | **30** |  |  |

**4.1 Теми лекцій**

|  |  |  |  |
| --- | --- | --- | --- |
| **№**  **з/п** | **Назва теми** | **Кількість**  **годин** | **Рекомендована література** |
| **Клієнт – серверна архітектура** | | | | |
|  | Вступ до клієнт-серверної архітектури. Основні поняття клієнт-серверної взаємодії. Історія розвитку архітектур. Типові приклади використання. | 2 | 9, С. 7-11; 7, С. 3-9 |
|  | Моделі клієнт-серверної архітектури. Однорівневі та багаторівневі архітектури. Протоколи взаємодії: HTTP, FTP, TCP/IP. Переваги та недоліки різних моделей. | 2 | 7, С. 9-15 |
|  | Протоколи передачі даних. Основи протоколів TCP/IP. Робота з HTTP, HTTPS. Протоколи прикладного рівня (FTP, SMTP, WebSocket) | 2 | 7, С. 3-9 |
| **Клієнт – серверні технології** | | | | |
|  | Серверні технології. Веб-сервери: Apache, Nginx. Сервери додатків. Конфігурація та налаштування серверів. | 2 | 9, С. 11-18; 7, С. 9-15 |
|  | Клієнтські технології. Веб-клієнти: браузери, мобільні додатки. Клієнтські бібліотеки та фреймворки (Ajax, Fetch API). Інтерактивність клієнтів. | 2 | 7, С. 3-9 |
|  | Архітектура RESTful. Принципи REST. Побудова RESTful API. HTTP методи та коди стану. | 2 | 7, С. 9-15 |
|  | Розробка API: практичні аспекти. Проектування API. Документація API (Swagger, OpenAPI). Аутентифікація та авторизація. | 2 | 9, С. 19-30; 7, С.63-78 |
|  | Кешування в клієнт-серверній архітектурі. Основи кешування. Клієнтське та серверне кешування. Використання кешів для оптимізації роботи. | 2 | 9, С. 27-31; 7, С.131-136 |
|  | Безпека клієнт-серверної взаємодії. Основні загрози та вразливості. Шифрування даних (TLS/SSL). Аутентифікація користувачів (OAuth, JWT). | 2 | 9, С. 31-38; 7, С. 15-18 |
|  | Реалізація WebSocket. Порівняння HTTP та WebSocket. Архітектура WebSocket. Приклади використання WebSocket для реального часу. | 2 | 9, С. 38-42; 7, С. 23-33 |
|  | Мікросервісна архітектура. Основи мікросервісів. Переваги та недоліки мікросервісної архітектури. Оркестрація та комунікація між сервісами. | 2 | 9, С. 43-64; 7, С. 33-64 |
|  | Хмарні технології в клієнт-серверній взаємодії. Основи хмарних обчислень. Використання хмарних сервісів для розробки клієнт-серверних систем. AWS, Azure, Google Cloud. | 2 | 9, С. 85-112 7, С. 78-100 |
|  | Моніторинг та логування. Інструменти для моніторингу серверів. Практики логування. Аналіз логів для виявлення проблем. | 2 | 9, С. 235-239; 4, С. 10-87 |
|  | Масштабування та оптимізація. Вертикальне та горизонтальне масштабування. Оптимізація продуктивності. Балансування навантаження. | 2 | 6, С. 239-241 4, С. 101-160 |
|  | Огляд сучасних тенденцій та майбутнє клієнт-серверних систем. Огляд нових технологій (GraphQL, gRPC). Перспективи розвитку клієнт-серверної архітектури. Інтеграція з іншими технологіями (IoT, AI). | 2 | 6, С. 241-246 4, С. 291-310 |
| **Разом** | | **30** |  |

**4.2 Теми практичних занять**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№**  **з/п** | **Назва теми, план.** | **Кількість**  **годин** | **Форма та засоби контролю** | **Рекомендована література** |
|  | Створення VM на Amazon Web Services. Запуск екземпляра EC2 | 2 | виконання завдань, індивідуальне оцінювання | 6, С. 7-80 |
|  | Створення сегмента S3 | 2 | виконання завдань, індивідуальне оцінювання | 6, С. 80-148 2. С. 13-47 |
|  | Використання CloudFront як CDN для веб-сайту | 2 | виконання завдань, індивідуальне оцінювання | 6, С. 80-148 2. С. 13-47 |
|  | Прикріплення тому EBS | 2 | виконання завдань, індивідуальне оцінювання | 6, С. 86-148 2. С. 49-59 |
|  | Вступ до IAM | 2 | виконання завдань, індивідуальне оцінювання | 6, С. 86-148 2. С. 49-59 |
|  | Створення сповіщення CloudWatch | 2 | виконання завдань, індивідуальне оцінювання | 2. С. 76-84 3. С. 56-58 |
|  | Створення екземпляра бази даних Amazon RDS | 2 | виконання завдань, індивідуальне оцінювання | 2. С. 76-84 3. С. 56-58 |
|  | Використання балансувальників навантаження | 2 | виконання завдань, індивідуальне оцінювання | 2. С. 117-171 3. С. 59-67 |
|  | Використання Elastic Beanstalk і CloudFormation | 2 | виконання завдань, індивідуальне оцінювання | 2. С. 117-171 3. С. 59-67 |
|  | Розгортання програм Windows Azure. | 2 | виконання завдань, індивідуальне оцінювання | 2. С. 117-171 3. С. 59-67 |
|  | Створення клієнт-серверних додатків з використанням платформи Windows Azure | 2 | виконання завдань, індивідуальне оцінювання | 6. С. 335-365 3. С. 75-81 |
|  | Створення клієнт-серверних додатків з використанням платформи Google Web Engine | 2 | виконання завдань, індивідуальне оцінювання | 6. С. 256-286 3. С. 81-83 |
|  | Використання технології віртуалізації Hyper-V. | 2 | виконання завдань, індивідуальне оцінювання | 6. С. 283-287 3. С. 88-90 |
|  | Робота з онлайн-офісним пакетом Google Docs. | 2 | виконання завдань, індивідуальне оцінювання | 6. С. 328-333 3. С. 83-87 |
|  | Проектний захист. Презентація розроблених проектів. Оцінка та зворотний зв'язок. | 2 | виконання завдань, індивідуальне оцінювання | 6. С. 328-333 3. С. 87-88 |
| **Разом** | | **30** |  |  |

**4.3 Самостійна робота**

|  |  |  |  |
| --- | --- | --- | --- |
| **№**  **з/п** | **Назва теми** | **Кількість**  **годин** | **Рекомендована література** |
|  | **Архітектура мікросервісів у клієнт-серверній взаємодії**: дослідження основних принципів мікросервісної архітектури, її переваг і недоліків, а також прикладів її використання в сучасних системах. | 10 | 9, С. 27-31;  7, С.131-136  6. С. 7-80 |
|  | **Веб-сокети та їх роль у реальному часі**: вивчення технології веб-сокетів для забезпечення двонаправленого зв'язку між клієнтом і сервером у режимі реального часу, приклади використання у веб-додатках. | 10 | 6. С. 80-150  3. С. 3-20 |
|  | **Безпека в клієнт-серверній взаємодії: SSL/TLS**: дослідження методів забезпечення безпеки передачі даних між клієнтом і сервером, зокрема використання протоколів SSL та TLS. | 10 | 9, С. 7-30;  7, С. 3-136 |
|  | **RESTful API: принципи та використання**: самостійне опрацювання основних концепцій RESTful архітектури, її переваг та способів інтеграції у клієнт-серверні системи. | 10 | 9, С. 7-11;  7, С. 3-9 |
|  | **RPC (Remote Procedure Call) та його варіації**: вивчення механізму віддаленого виклику процедур (RPC), його реалізацій та використання в різних типах клієнт-серверних систем. | 10 | 9, С. 85-239  7, С. 78-100  6, С. 254-280 |
|  | **Кешування в клієнт-серверних системах**: дослідження методів кешування для підвищення продуктивності клієнт-серверних систем, зокрема використання Redis, Memcached та інших рішень. | 10 | 6, С.235-268 |
|  | **Основи хмарних обчислень та їх інтеграція з клієнт-серверною архітектурою**: аналіз того, як хмарні платформи використовуються для розширення можливостей клієнт-серверних систем, приклади SaaS, PaaS, IaaS. | 10 | 6, С. 319-335,  283-293,  307-319 |
|  | **Протокол MQTT та його використання в IoT**: вивчення легковагового протоколу MQTT, який використовується для забезпечення зв'язку між пристроями в Інтернеті речей (IoT). | 10 | 4, С. 10-310 |
|  | **Навантажувальне тестування клієнт-серверних систем**: дослідження інструментів і методів для тестування продуктивності та стійкості клієнт-серверних систем під різним навантаженням (наприклад, JMeter, LoadRunner). | 10 | 6, С. 319-335,  283-293,  307-319 |
| **Разом** | | **90** |  |

**5. Засоби діагностики результатів навчання, інструменти, обладнання та програмне забезпечення, використання яких передбачає навчальна дисципліна**

Засобами оцінювання та методами демонстрування результатів навчання є диференційований залік, практичні завдання за комп’ютером, реальних об’єктах (сервер та його складові), аналітичні звіти, реферати, презентації результатів виконаних завдань та досліджень, письмове виконання ІНДЗ, виступи на наукових заходах, Використовуються відео лекції, практичні заняття з індивідуальними завданнями, самостійна робота здобувача фахової передвищої освіти з навчальною та довідковою літературою, самостійне виконання завдань, консультації. Використовуються лабораторний навчальний комп’ютер, інструктивні картки для практичних робіт, картки з індивідуальними завданнями для практичних робіт. Використовується доступ до мережі інтернет, хмарні сервіси.

|  |  |  |  |
| --- | --- | --- | --- |
| **6. Порядок та критерії оцінювання результатів навчання** | | | |
| **6.1. Порядок оцінювання результатів навчання** | | | |
| Форма контролю | | Порядок проведення контролю | |
| Поточний контроль | | Усне опитування, домашні завдання, захист звітів з практичних робіт, виступи на практичних заняттях, практичні та письмові роботи оцінюються за 4-бальною шкалою | |
| Підсумковий контроль | | Оцінка за залік визначається як середнє арифметичне оцінок за практичні роботи. | |
| **6.2. Критерії оцінювання результатів навчання** | | | |
| Оцінювання за національною шкалою: | | | Критерії та визначення оцінювання |
| рівень компетентності | оцінка: | |
| 4-бальна | |  |
| **1** | **2** | | **3** |
| Високий  (творчий) | 5  (відмінно) | | Здобувач освіти вільно володіє програмовим матерiалом, виявляє здiбностi, вмiє самостійно поставити мету дослiдження, вказує шляхи її реалiзацiї, робить аналiз та висновки.  Здобувач освіти вiльно володiє вивченим матерiалом, умiло послуговується науковою термiнологiєю, вмiє опрацьовувати наукову iнформацiю (знаходити новi фак­ти, явища, iдеї, самостiйно використовувати їх вiдповiдно до поставленої мети тощо).  Здобувач освіти вiльно володiє вивченим матерiалом, умiло послуговується науковою термiнологiєю, вмiє опрацьовувати наукову iнформацiю (знаходити новi фак­ти, явища, iдеї, самостiйно використовувати їх вiдповiдно до поставленої мети тощо). |
| Достатній  (конструктивно-варіативний) | 4  (добре) | | Здобувач освіти вiльно володiє вивченим матерiалом у стандартних ситуацiях, наводить приклади його практичного застосування та аргументи на пiдтвердження власних думок. Здобувач освіти умiє пояснювати роботу вузлів клієнт-серверної архітектури, аналізувати, узагальнювати знання, систематизувати їх, зi сторонньою допомогою (викладача, одногрупників тощо) ро­бити висновки. Здобувач освіти може пояснювати роботу вузлів клієнт-серверної архітектури, виправляти допущенi неточностi, виявляє знання i розумiння основних протоколів клієнт-серверної взаємодії (функціонування, характеристики, особливості застосування). |
| Середній  (репродуктивний) | 3  (задовільно) | | Здобувач освіти може зi сторонньою допомогою пояснювати призначення вузлів клієнт-серверної архітектури, виправляти допущенi неточностi (власнi, iнших уч­нiв), виявляє елементарнi знання основних по­ложень клієнт-серверної взаємодії (законiв, понять, формул). Здобувач освіти описує явища, вiдтворює значну частину навчального матерiалу, знає складові вузлів клієнт-серверної архітектури, їх характеристики, записує основнi співвідношення, рiвняння i закони. Здобувач освіти за допомогою викладача описує вузли клієнт-серверної архітектури, без пояснень наводить приклади, що ґрунтуються на його власних спостереженнях чи матерiалi пiдручника, розповiдях викладача тощо. |
| Початковий  (рецептивно-продуктивний) | 2  (незадовільно) | | Здобувач освіти за допомогою викладача описує вузли клієнт-серверної архітектури або його частини у незв’язаному вигляді без пояснень вiдповiдних причин, називає параметри та характеристики складових, розрiзняє призначення окремих складових вузлів клієнт-серверної архітектури. |
| Здобувач освіти описує роботу клієнт-серверної взаємодії на ос­новi свого попереднього досвiду, за допомогою викладача вiдповiдає на запитання, що потребують однослiвної вiд­по­вiдi. |
| Здобувач освіти володiє навчальним мате­рiа­лом на рiвнi­ розпiзнавання складових клієнт-серверної архітектури, за допомогою викладача вiдповiдає на запитання, що потребують вiдповiдi “так” чи “нi”. |

**7. Рекомендована література**

**Основна**

1. James F. Kurose, Keith W. Ross. Computer Networking: A Top-Down Approach.
2. Fundamentals of Web Development. - Randy Connolly, Ricardo Hoar
3. Lee Atchison. Architecting for Scale.
4. Artur Ejsmont. Web Scalability for Startup Engineers.
5. Ilya Grigorik. High Performance Browser Networking
6. Leonard Richardson, Mike Amundsen, Sam Ruby. RESTful Web APIs.
7. Комп’ютерні мережі: [Книга 1. Технології комп’ютерних мереж]: Навчальний посібник / Євсеєв С.П., Дженюк Н.В., Толкачов М.Ю та ін. – Харків, – Львів: Видавництво ПП «Новий Світ – 2000», 2024. – 471 с.

**Допоміжна**

1. Andrew S. Tanenbaum, David J. Wetherall. Computer Networks, 5th Edition. – Prentice Hall, Indian International Ed., 2010. – 960 p
2. Організація комп’ютерних мереж [Електронний ресурс]: підручник: для студ. Спеціальності 121 «Інженерія програмного забезпечення»та 122 «Комп’ютерні науки» / КПІ ім. Ігоря Сікорського; Ю.А. Тарнавський, І.М. Кузьменко. – Електронні текстові дані (1 файл: 45,7 Мбайт). – Київ: КПІ ім. Ігоря Сікорського, 2018. – 259 с
3. Комп’ютерні мережі.Конспект лекцій /Укл.: Зав’ялець Ю.А. – Чернівці, 2015. – 183 с.
4. Комп’ютерні мережі : навчальний посібник / О. С. Городецька, В. А. Гикавий, О. В. Онищук. − Вінниця : ВНТУ, 2017. − 129 с.
5. Карпенко М. Ю. Конспект лекцій з курсу «Комп’ютерні мережі» (для студентів усіх форм навчання спеціальностей 122 – Комп’ютерні науки, 151 – Автоматизація та комп’ютерно-інтегровані технології, 126 – Інформаційні системи та технології) / М. Ю. Карпенко, Н. В. Макогон; Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. – Харків : ХНУМГ ім. О. М. Бекетова, 2019. – 99 с.

**8. Інформаційні ресурси**

1. Курс "Computer Networks" від Університету Вашингтона пропонує базові та просунуті знання з комп'ютерних мереж, включаючи клієнт-серверну архітектуру, протоколи TCP/IP, HTTP : <https://www.coursera.org/learn/illinois-tech-computer-networking>
2. Курс "Web Applications for Everybody" від Університету Мічигану охоплює теми, пов'язані з веб-програмуванням, RESTful API та клієнт-серверною взаємодією: <https://www.coursera.org/specializations/web-applications>
3. Розробка програмного забезпечення та вебтехнології. Linux Professional Institute Inc. <https://learning.lpi.org/uk/learning-materials/030-100/031/>
4. REST API як спосіб спілкування компонент веб-додатків. Навчальний центр FoxmindEd: <https://foxminded.ua/shcho-take-rest-api/>
5. Протокол TCP/IP або як працює Інтернет (для початківців). Центр підготовки IT-фахівців. <https://qalight.ua/baza-znaniy/protokol-tcp-ip-abo-yak-praczyu%D1%94-internet-dlya-pochatkivcziv/>

**9. 1 ПЕРЕЛІК ОРІЄНТОВНИХ ПИТАНЬ ПІДГОТОВКИ ДО ЗАЛІКУ**

1. Що таке клієнт-серверна архітектура і які її основні принципи?
2. Які основні відмінності між однорівневою та багаторівневою архітектурою?
3. Яка роль протоколу HTTP у клієнт-серверній взаємодії?
4. Поясніть основні функції протоколу TCP/IP.
5. У чому полягають переваги та недоліки трирівневої архітектури?
6. Що таке RESTful API, і як воно використовується в клієнт-серверних додатках?
7. Порівняйте мікросервісну архітектуру з монолітною. Які є переваги та недоліки кожної?
8. Як забезпечується безпека передачі даних у клієнт-серверних системах?
9. Що таке веб-сокети, і як вони використовуються для реального часу в веб-додатках?
10. Поясніть процес аутентифікації та авторизації у клієнт-серверних системах.
11. Які основні компоненти складають трирівневу архітектуру?
12. Що таке FTP і яка його роль у клієнт-серверній взаємодії?
13. Опишіть механізм кешування в клієнт-серверних системах.
14. Як працює модель "клієнт-сервер" у контексті хмарних обчислень?
15. Що таке навантажувальне тестування і для чого воно потрібне в клієнт-серверних системах?
16. Які є основні протоколи передачі даних у клієнт-серверній архітектурі?
17. Як використовуються SSL/TLS для захисту клієнт-серверних з’єднань?
18. Які є методи для оптимізації продуктивності клієнт-серверних систем?
19. Що таке RPC (Remote Procedure Call) і як він реалізується?
20. Які є можливі проблеми при масштабуванні клієнт-серверних систем?
21. Як функціонує протокол FTP і які його основні команди?
22. Що таке DNS і яка його роль у клієнт-серверних системах?
23. Як клієнт-серверна архітектура впливає на дизайн баз даних?
24. Що таке "безстатевість" у контексті протоколу HTTP, і як це впливає на розробку додатків?
25. Як забезпечити високу доступність у клієнт-серверних системах?
26. Які основні підходи до захисту від атак типу "людина посередині" (MITM) у клієнт-серверній архітектурі?
27. Що таке Session Management і як воно реалізується в клієнт-серверних додатках?
28. Як працює модель "публікація-підписка" (Publish-Subscribe) у клієнт-серверній взаємодії?
29. Що таке CDN (Content Delivery Network) і як воно впливає на клієнт-серверну взаємодію?
30. Які існують методи для розподілу навантаження в клієнт-серверних системах?
31. Поясніть поняття "стейтлес" і "стейтфул" у контексті серверних додатків.
32. Що таке API Gateway і яку роль він відіграє в клієнт-серверній архітектурі?
33. Як реалізується обробка помилок у клієнт-серверних системах?
34. Які основні типи даних передаються через HTTP-запити (GET, POST, PUT, DELETE)?
35. Як функціонує протокол WebSocket і в чому його переваги перед HTTP?
36. Що таке OAuth і як він використовується для управління доступом до ресурсів у клієнт-серверних системах?
37. Як реалізувати безпечне зберігання сесій в багатокористувацьких системах?
38. Які є підходи до забезпечення безперервної інтеграції (CI/CD) у клієнт-серверних додатках?
39. Як працює технологія віддаленого виклику методів (RMI) у клієнт-серверних додатках?
40. Що таке "серіалізація" і "десеріалізація" у контексті передачі даних між клієнтом і сервером?
41. Які є переваги використання GraphQL у порівнянні з традиційними RESTful API?
42. Як функціонує MQTT-протокол і де він використовується?
43. Як реалізується балансування навантаження в розподілених системах?
44. Що таке "щільність даних" і як вона впливає на продуктивність клієнт-серверних систем?
45. Як реалізується механізм шифрування даних на рівні транспортного протоколу (TLS)?
46. Що таке API Rate Limiting і як він захищає сервери від перевантаження?
47. Як забезпечити сумісність між різними версіями API у клієнт-серверних системах?
48. Які є методи моніторингу та логування у клієнт-серверних системах?
49. Як працює гібридна архітектура (змішана модель) у клієнт-серверних системах?
50. Що таке "демаркаційна точка" в контексті клієнт-серверних систем і яку роль вона відіграє в мережевій архітектурі?