

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Львівський національний університет імені Івана Франка**  
**Факультет електроніки та комп'ютерних технологій**  
**Кафедра системного проектування**

**Затверджено**

На засіданні кафедри системного проектування факультету електроніки та комп'ютерних технологій Львівського національного університету імені Івана Франка  
(протокол № 1 від 28.08 2023 р.)

Завідувач кафедри:



Роман ШУВАР

**Силабус з навчальної дисципліни**  
**“Веб програмування на стороні сервера”,**  
**що викладається в межах ОПП “Комп'ютерні науки”**  
**першого (бакалаврського) рівня вищої освіти для здобувачів з**  
**спеціальності 122 – Комп'ютерні науки**

Львів 2023 р.

<b>Назва дисципліни</b>	Веб програмування на стороні сервера
<b>Адреса викладання дисципліни</b>	Корпус факультету електроніки та комп'ютерних технологій, Львівський національний університет імені Івана Франка, вул. Драгоманова 50, м. Львів, 79005, вул. Ген. Тарнавського 107, м. Львів, 79011
<b>Факультет та кафедра, за якою закріплена дисципліна</b>	Факультет електроніки та комп'ютерних технологій Кафедра системного проектування
<b>Галузь знань, шифр та назва спеціальності</b>	12 – Інформаційні технології 122 – Комп'ютерні науки
<b>Викладачі дисципліни</b>	Демків Лідія Степанівна, доцент кафедри системного проектування; Чмихало Олександр Сергійович, асистент кафедри системного проектування; Гусак Олег Васильович, асистент кафедри системного проектування.
<b>Контактна інформація викладачів</b>	<a href="mailto:lidiya.demkiv@lnu.edu.ua">lidiya.demkiv@lnu.edu.ua</a> <a href="https://electronics.lnu.edu.ua/employee/demkiv-l-s/">https://electronics.lnu.edu.ua/employee/demkiv-l-s/</a> <a href="mailto:oleksandr.chmykhalo@lnu.edu.ua">oleksandr.chmykhalo@lnu.edu.ua</a> , <a href="https://electronics.lnu.edu.ua/en/employee/chmykhalo-o/">https://electronics.lnu.edu.ua/en/employee/chmykhalo-o/</a> <a href="mailto:oleh.husak@lnu.edu.ua">oleh.husak@lnu.edu.ua</a> , <a href="https://electronics.lnu.edu.ua/en/employee/husak-o-v/">https://electronics.lnu.edu.ua/en/employee/husak-o-v/</a>
<b>Консультації з питань навчання по дисципліні відбуваються</b>	Консультації в день проведення лекцій/лабораторних занять (за попередньою домовленістю): ауд. 305, корпус факультету електроніки та комп'ютерних технологій, вул. Драгоманова 50, м. Львів
<b>Сторінка курсу</b>	<a href="https://moodle.elct.lnu.edu.ua/course/view.php?id=189">https://moodle.elct.lnu.edu.ua/course/view.php?id=189</a>
<b>Інформація про дисципліну</b>	Дисципліна “Веб програмування на стороні сервера” є нормативною дисципліною з спеціальності 122 – Комп'ютерні науки для освітньої програми “Комп'ютерні науки”, яка викладається в 3-му семестрі в обсязі 5-ох кредитів (за Європейською Кредитно-Трансферною Системою ECTS).
<b>Коротка анотація дисципліни</b>	Навчальну дисципліну розроблено таким чином, щоб надати учасникам необхідні знання, обов'язкові для того, щоб оволодіти основними концепціями на яких ґрунтується Веб програмування на стороні сервера та бути здатними застосовувати ці концепції для вирішення прикладних задач.
<b>Мета та цілі дисципліни</b>	<i>Мета:</i> розвинути у студентів системне базове уявлення, первинні знання, вміння і навички з основ веб програмування на стороні сервера і знань як наукової і прикладної дисципліни, достатні для подальшого продовження освіти та самоосвіти в галузі обчислювальної техніки, інформаційних систем різного призначення; уявлення про клієнт-серверну архітектуру; програмні засоби та програмне забезпечення, які використовують для розробки серверних веб застосунків; про основи протоколу HTTP та інші поширені Веб стандарти для взаємодії між клієнтською та серверною частинами; отримання базового рівня щодо програмування простих серверних Веб застосунків..

	<p><i>Цілі:</i> розуміння принципу взаємодії між Веб клієнтом та Веб сервером та способів передачі даних між ними; функцій та призначення Веб посилань; форматів даних XML та JSON, а також їх прикладного використання протоколу HTTP та вирішення прикладних задач обміну даних з його допомогою; налагодження Веб програм на стороні сервера з допомогою спеціалізованих інструментів; використання мови програмування JavaScript для створення простих Веб додатків на стороні сервера; проектування Веб інтерфейсу для вирішення заданої прикладної задачі.</p>
<p><b>Література для вивчення дисципліни</b></p>	<p>Інтернет-ресурси:</p> <ol style="list-style-type: none"> <li>1) Git Book (український переклад) <a href="https://git-scm.com/book/uk/v2">https://git-scm.com/book/uk/v2</a></li> <li>2) Магія Git (онлайн книга) <a href="http://www-cs-students.stanford.edu/~blynn/gitmagic/intl/uk/">http://www-cs-students.stanford.edu/~blynn/gitmagic/intl/uk/</a></li> <li>3) Сучасний підручник з JavaScript (онлайн книга) <a href="https://uk.javascript.info/">https://uk.javascript.info/</a></li> <li>4) W3schools українською: Node.js Підручник <a href="https://w3schoolsua.github.io/nodejs/index.html#gsc.tab=0">https://w3schoolsua.github.io/nodejs/index.html#gsc.tab=0</a></li> <li>5) W3schools — Online Web Tutorials <a href="https://www.w3schools.com/">https://www.w3schools.com/</a></li> <li>6) Серія навчальних відео «Просто про Web» <a href="https://www.youtube.com/watch?v=ByJl2IcWSfc&amp;list=PLcIcVZ3JhM-gsEmYSzhXy97jV8g5V1O3l">https://www.youtube.com/watch?v=ByJl2IcWSfc&amp;list=PLcIcVZ3JhM-gsEmYSzhXy97jV8g5V1O3l</a></li> <li>7) cURL Documentation <a href="https://curl.se/docs/">https://curl.se/docs/</a></li> <li>8) Fiddler Documentation <a href="https://docs.telerik.com/fiddler-everywhere/introduction">https://docs.telerik.com/fiddler-everywhere/introduction</a></li> <li>9) Postman Documentation <a href="https://learning.postman.com/docs/introduction/overview/">https://learning.postman.com/docs/introduction/overview/</a></li> <li>10) Swagger Documentation <a href="https://swagger.io/docs/">https://swagger.io/docs/</a></li> <li>11) Сайт ExpressJS українською <a href="https://expressjs.com/uk/">https://expressjs.com/uk/</a></li> </ol>
<p><b>Обсяг курсу</b></p>	<p>Загальний обсяг: 150 годин. Аудиторних занять: 64 год., з них 32 год. лекційних та 32 годин лабораторних робіт. Самостійної роботи: 86 год.</p>
<p><b>Очікувані результати навчання</b></p>	<p>У результаті вивчення даного курсу студент буде:</p> <p><b>знати:</b> основні принципи що лежать в основі клієнт-серверної архітектури, основні принципи взаємодії з сервером використовуючи сучасні стандарти Веб, стандартні підходи до організації такої взаємодії. Мати навички створення та налагодження Веб застосунків на стороні сервера. Розуміти підходи до проектування Веб інтерфейсів.</p> <p><b>вміти:</b> застосовувати знання для розв'язування прикладних задач використовуючи сучасні стандарти та протоколи Веб, проектувати Веб інтерфейси для взаємодії з сервером з допомогою протоколу HTTP, створювати та налагоджувати Веб програми на стороні сервера використовуючи мову JavaScript та спеціалізовані інструменти.</p>

	<p>Після вивчення даного курсу здобувачі набувають таких Загальних (ЗК), Спеціальних/Фахових (СК) компетентностей та Програмних результатів навчання (ПР):</p> <p>ЗК2. Здатність застосовувати знання у практичних ситуаціях.</p> <p>ЗК12. Здатність оцінювати та забезпечувати якість виконуваних робіт.</p> <p>СК 3. Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем.</p> <p>СК 8. Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.</p> <p>СК 9. Здатність реалізувати багаторівневу обчислювальну модель на основі архітектури клієнт-сервер, включаючи бази даних, знань і сховища даних, виконувати розподілену обробку великих наборів даних на кластерах стандартних серверів для забезпечення обчислювальних потреб користувачів, у тому числі на хмарних сервісах.</p> <p>СК 16. Здатність реалізовувати високопродуктивні обчислення на основі хмарних сервісів і технологій, паралельних і розподілених обчислень при розробці й експлуатації розподілених систем паралельної обробки інформації.</p> <p>ПР 10. Використовувати інструментальні засоби розробки клієнт-серверних застосувань, проектувати концептуальні, логічні та фізичні моделі баз даних, розробляти та оптимізувати запити до них, створювати розподілені бази даних, сховища та вітрини даних, бази знань, у тому числі на хмарних сервісах, із застосуванням мов веб-програмування.</p> <p>ПР 11. Володіти навичками управління життєвим циклом програмного забезпечення, продуктів і сервісів інформаційних технологій відповідно до вимог і обмежень замовника, вміти розробляти проектну документацію (техніко-економічне обґрунтування, технічне завдання, бізнес-план, угоду, договір, контракт).</p> <p>ПР 14. Застосовувати знання методології та CASE-засобів проектування складних систем, методів структурного аналізу систем, об'єктно-орієнтованої методології проектування при розробці і дослідженні функціональних моделей організаційно-економічних і виробничо-технічних систем.</p> <p>ПР 15. Розуміти концепцію інформаційної безпеки, принципи безпечного проектування програмного забезпечення, забезпечувати безпеку комп'ютерних мереж в умовах неповноти та невизначеності вихідних даних.</p>
<b>Ключові слова</b>	Клієнт, сервер, сервіс, хост, запит, URL, JSON, XML, JavaScript, функція зворотнього виклику, Node.JS, HTTP, npm, curl, Fiddler, Postman, Web API, REST, автентифікація, сесія, TLS, Swagger
<b>Формат курсу</b>	Очний
<b>Теми</b>	Див. <b>Схема курсу</b>
<b>Підсумковий контроль,</b>	Екзамен у кінці семестру

<b>форма</b>	
<b>Пререквізити</b>	Для вивчення даного курсу студентам потрібні базові знання з курсів: <ul style="list-style-type: none"> <li>- основи програмування;</li> <li>- алгоритми і структури даних;</li> <li>- об'єктно-орієнтоване програмування.</li> </ul>
<b>Навчальні методи та техніки, які будуть використовуватися під час викладання курсу</b>	Інформаційні методи (лекції, презентації, лабораторні роботи, виконання індивідуальних завдань, робота у групі, командна робота, обговорення, консультації для поглибленого розуміння тем, бесіда, ілюстрація, демонстрація), дедуктивні методи на основі узагальнень, евристичні методи (проблемна лекція), інтерактивні методи (дискусія).
<b>Необхідне обладнання</b>	<p>Для проведення лекційних занять:</p> <ul style="list-style-type: none"> <li>• монітор TFT 23";</li> <li>• системний блок (процесор Intel i5-6500, 8GB оперативної пам'яті, HDD 256GB) ;</li> <li>• мультимедійне обладнання (проектор, проекційний екран, дошка настінна, звуковий підсилювач та аудіосистема);</li> <li>• комутатор мережевий для доступу до мережі Internet.</li> </ul> <p>Для проведення лабораторних занять:</p> <ul style="list-style-type: none"> <li>• комп'ютерна лабораторія з 12-14 робочими місцями;</li> <li>• монітори TFT 23";</li> <li>• системні блоки (процесор Intel i5-6500, 8GB оперативної пам'яті, HDD 256GB);</li> <li>• мультимедійне обладнання (проектор, проекційний екран, дошка настінна, звуковий підсилювач та аудіосистема);</li> <li>• комутатор мережевий для доступу до мережі Internet.</li> </ul> <p>Необхідне програмне забезпечення:</p> <ul style="list-style-type: none"> <li>• операційна система ОС Ubuntu 16 LTS чи Microsoft Windows 10 ;</li> <li>• середовище розробки на мові програмування JavaScript: VSCode (версії 1.8x чи вище);</li> <li>• Node JS (версії 9.x чи вище);</li> <li>• cURL (версії 8.x чи вище);</li> <li>• Git (версії 2.41.x чи вище);</li> <li>• Fiddler (версії 5);</li> <li>• Postman</li> </ul>
<b>Критерії оцінювання (окремо для кожного виду навчальної діяльності)</b>	<p>Оцінювання проводиться за 100-бальною шкалою. Бали нараховуються за наступним співвідношенням:</p> <ul style="list-style-type: none"> <li>• Змістовий модуль: до 20 балів за виконання модульного завдання – відповіді на теоретичні питання з переліку.</li> <li>• Виконання лабораторних робіт: до 30 балів.</li> <li>• Екзамен: до 50 балів.</li> </ul> <p>Підсумкова максимальна кількість балів 100.</p> <p><b>Змістовий модуль</b> – Відповіді на теоретичні питання з переліку складеного за темами лекційних занять. Кожне питання має бути розкрито в обсязі необхідному для того, щоб продемонструвати що студент розуміє матеріал. Як правило до теоретичної інформації мають бути наведені приклади. Питання для змістових модулів див. у розділі</p>

## **Питання до модульного контролю.**

**Академічна доброчесність:** Очікується, що роботи студентів будуть оригінальними дослідженнями чи міркуваннями. Списування та втручання в роботу інших студентів становлять, але не обмежують, приклади можливої академічної недоброчесності. Виявлення ознак академічної недоброчесності в написанні завдань є підставою для її незарахування викладачем, незалежно від масштабів плагіату чи обману. Жодні форми порушення академічної доброчесності не толеруються.

**Відвідання занять** є важливою складовою навчання. Очікується, що всі студенти відвідають усі лекції та лабораторні заняття курсу. Студенти повинні інформувати викладача про неможливість відвідати заняття. У будь-якому випадку студенти зобов'язані дотримуватися термінів визначених для виконання всіх видів робіт, передбачених курсом.

**Література.** Уся література, яку надає викладач перебуває у відкритому доступі в мережі інтернет.

**Політика виставлення балів.** Враховуються бали, набрані при поточному контролі, бали за виконання лабораторних робіт, а також бали за підсумковий іспит.

**Оцінювання лабораторних робіт** (7 лабораторних робіт, максимальна кількість балів: 30) відбувається шляхом оцінки роботи студента під час проведення лабораторної роботи в аудиторії (2 бали за перші дві роботи, 4 бали за лабораторні 3-6, 10 балів за 7 лабораторну роботу) та захисту звіту по виконаній лабораторній роботі. Складність лабораторних робіт поступово зростає. Оцінка ставиться за наступними критеріями:

- володіння навчальним матеріалом, відповіді на теоретичні питання по темі;
- оформлення звіту до лабораторної роботи;
- функціонування коду програми відповідно до умов завдання:

**Оцінювання змістових модулів** (1 змістових модулів, 20 балів ) — за результатами відповіді на 4 питання.

Кожне з 4 питань оцінюється окремо. Бали оцінювання нараховуються за наступним співвідношенням:

5 - розглянута тема відтворюється в повному обсязі, правильно, обґрунтовано, логічно, містить аналіз і систематизацію, аргументовані висновки. Засвідчено глибоке володіння матеріалом. Наведені приклади відповідають темі. Можуть бути присутні несуттєві помилки та невідповідності;

4 - відтворюється значна частина розглянутої теми. Виявлено знання і розуміння основних положень навчальної дисципліни, проте присутні неточності та/або невідповідності основній темі. Наведені приклади не дуже точні, проте в загальному відповідають темі;

3 - відстежується загальне розуміння розглянутої теми. Виявлені множинні неточності та невідповідності, пояснення відсутні;

2 – студент погано розуміє розглянуту тему. Виявлені суттєві неточності та невідповідності. Наведені приклади і код з суттєвими недоліками, або

	<p>не відповідають темі; 1 – 0 – студент взагалі не розуміє розглянуту тему. Тему не розкрито.</p> <p><b>Критерії оцінювання результатів неформальної освіти:</b> Нарахування балів відбувається за написання студентом тез доповідей на конференціях, наукових статей, участь у діяльності наукових гуртків, участь у наукових семінарах та круглих столах, конкурсах, участь у заходах неформальної освіти за отримання сертифікатів про проходження навчання на різних освітніх платформах (Coursera, Prometheus тощо), курсах на провідних ІТ компаніях за тематикою навчальної дисципліни. Кількість балів визначається відсотком покриття результатів відповідної активності до вимог результатів навчання з навчальної дисципліни.</p>
<p><b>Питання до модульного контролю</b></p>	<ol style="list-style-type: none"> <li>1. Що таке клієнт-серверна архітектура? Поясніть основні принципи на яких вона побудована?</li> <li>2. Поясніть такі поняття: клієнт, сервер, сервіс, хост. Наведіть приклади.</li> <li>3. Як працює інтерфейс командного рядка? Чому вміти користуватись командним рядком є важливо для розробника?</li> <li>4. Які є базові команди для переміщення файловою системою та перегляду файлів з командного рядка Linux/Unix? Поясніть їх роботу на наведіть приклади для кожної з них</li> <li>5. Які є базові команди командного рядка Linux/Unix для створення / видалення файлів та тек? Поясніть їх роботу на наведіть приклади для кожної з них.</li> <li>6. Що таке система контролю версій і чому вона незамінна у роботі розробника? Наведіть приклади її використання.</li> <li>7. Як виглядає для розробника робота з системою контролю версій Git? Навіщо потрібен віддалений репозиторій і як розробник з ним взаємодіє?</li> <li>8. Дайте визначення і наведіть приклади для таких понять: репозиторій, коміт, стейджінг (staging area), гілка (branch)?</li> <li>9. Як створити і підготувати репозиторій для роботи? Наведіть приклади команд які для цього використовують</li> <li>10. Поясніть на прикладі як відбувається запит до сервера?</li> <li>11. Що таке URL, навіщо і як його використовують?</li> <li>12. Дайте визначення основних частин URL. Як ці частини вони виділені в URL?</li> <li>13. Як працює відсоткове кодування та навіщо його використовують? Як перевести текст у відсоткове кодування і назад? Наведіть</li> </ol>

приклади

14. Що таке JSON і як його використовують? Наведіть приклади.
15. Який зв'язок між JSON і JavaScript? Як співвідноситься JSON з типами даних JavaScript? Як працювати з JSON у JavaScript?
16. Як оголошуються змінні у JavaScript? Як задається тип даних для змінних у JavaScript? В чому різниця між let та const? Що таке примітиви і які вони бувають?
17. Що таке масив у JavaScript? Як працювати з масивами у JavaScript? Поясніть і наведіть приклади
18. Що таке об'єкт у JavaScript? Як працювати з об'єктами у JavaScript? Поясніть і наведіть приклади
19. Як визначаються функції у програмі JavaScript? Який є зв'язок між функціями та об'єктами?
20. Що таке функції зворотнього виклику (callback) та як їх використовують? Наведіть приклади.
21. Що таке функціональні вирази та стрілкові функції JavaScript? Як їх використовують? Наведіть приклади
22. Що таке модулі та як їх використовують? Які типи модулів бувають і яка між ними різниця?
23. Що таке синхронне та асинхронне виконання операцій? Порівняйте та наведіть приклади. Навіщо потрібні та як використовують асинхронні операції?
24. Що таке Node.JS та як його використовують? Який зв'язок між Node.JS і асинхронною обробкою операцій?
25. Що таке HTTP? Розкажіть про основні принципи роботи, застосування протоколу та наведіть приклади
26. Опишіть структуру HTTP запиту і структуру HTTP відповіді, наведіть приклади використання кожної з структурних частин запиту\відповіді
27. Розкажіть про основні методи HTTP та як вони застосовуються?
28. Які є групи статусних кодів HTTP? Поясніть призначення кодів кожної з груп та наведіть приклади.
29. Що таке Node.js Package Manager і як його використовують? Навіщо потрібен файл package.json та яку інформацію він може містити?
30. Що таке XML і як його використовують? Що таке теги і атрибути? Наведіть приклади.



	<p>31. Порівняйте XML та JSON: що є спільного та відмінного?</p> <p>32. Розкажіть про curl і особливості його використання. Наведіть приклади.</p> <p>33. Розкажіть про Fiddler і особливості його використання. Наведіть приклади.</p> <p>34. Розкажіть про Postman і особливості його використання. Наведіть приклади.</p> <p>35. Як передати дані на сервер і отримати дані з серверу використовуючи HTTP? Розкажіть та наведіть приклади.</p> <p>36. Розкажіть про формат x-www-form-urlencoded і його застосування.</p> <p>37. Розкажіть про формат multipart/form-data і його застосування.</p> <p>38. Що фреймворк Express.JS дає розробнику у порівнянні з використанням лише Node.js для розробки веб-сервера? Зробіть огляд особливостей Express.JS.</p> <p>39. Дайте визначення таким поняттям: інтерфейс, API, Web API, веб ресурс. Наведіть приклади.</p> <p>40. Які є правила “хорошого тону” для найменування веб ресурсів для створення REST Web APIs? Що означає абревіатура CRUD в контексті Web APIs?</p> <p>41. Які є правила “хорошого тону” для використання методів HTTP для створення REST Web APIs?</p> <p>42. Що таке автентифікація? Як автентифікація відрізняється від авторизації? Як сигналізувати помилку автентифікації чи авторизації на сервері?</p> <p>43. Назвіть кілька способів автентифікації та порівняйте їх між собою</p> <p>44. Як працює базова HTTP автентифікація? Наведіть приклад</p> <p>45. Як працює автентифікація на базі сесій? Наведіть приклад</p> <p>46. Як працює автентифікація на базі JWT токена? Наведіть приклад</p> <p>47. Що таке сесії та куки? Як вони використовуються?</p> <p>48. Що таке HTTPS? Що таке TLS? Як вони працюють?</p> <p>49. Що таке симетричне та асиметричне шифрування? Який зв'язок шифрування з алгоритмами цифрового підпису? Що таке хешування і де воно використовується?</p> <p>50. Що таке Swagger? Як він працює навіщо його використовують?</p>
<b>Опитування</b>	Анкету-оцінку з метою оцінювання якості курсу буде надано по завершенню курсу.

**Схема курсу “ Веб програмування на стороні сервера ”  
для студентів спеціальності 122 – Комп'ютерні науки**

Тиж.	Тема, план, короткі тези	Форма діяльності (заняття) **лекція, самостійна, дискусія, групова робота)	Література. * Ресурси в інтернеті	Завдання, год	Термін виконання
1	<b>Базові визначення. Командний рядок на елементарному рівні.</b> Огляд основних визначень: клієнт, сервер, сервіс, хост. Знайомство з інтерфейсом командного рядка. Переміщення файловою системою та маніпуляції з файлами з командного рядка.	Лекція	[*6], Сайт курсу – <a href="https://moodle.elct.lnu.edu.ua/course/view.php?id=303">https://moodle.elct.lnu.edu.ua/course/view.php?id=303</a>	2	кінець поточного тижня
1	Вступне заняття. Техніка безпеки, перевірка обладнання та налаштування програмного забезпечення.	Лабораторна робота	Сайт курсу - <a href="https://moodle.elct.lnu.edu.ua/course/view.php?id=303">https://moodle.elct.lnu.edu.ua/course/view.php?id=303</a>	2	кінець поточного тижня
2	<b>Принцип роботи системи контролю версій Git. Базовий алгоритм роботи з Git.</b> Знайомство з системами контролю версій. Знайомство з Git, встановлення, налаштування та базовий алгоритм роботи з ним. Основні поняття.	Лекція	[*1], [*2], Сайт курсу <a href="https://moodle.elct.lnu.edu.ua/course/view.php?id=303">https://moodle.elct.lnu.edu.ua/course/view.php?id=303</a>	2	кінець поточного тижня
2	<b>Лабораторна робота 1:</b> Базова робота з Git. Запуск простої програми на JavaScript.	Лабораторна робота	Сайт курсу - <a href="https://moodle.elct.lnu.edu.ua/course/view.php?id=303">https://moodle.elct.lnu.edu.ua/course/view.php?id=303</a>	2	кінець поточного місяця
3	<b>Запити до сервера. Uniform Resource Locators</b> Виконання запитів до сервера з допомогою програми cURL. Структура URL та їх використання. Відсоткове кодування.	Лекція	[*6], [*7] Сайт курсу - <a href="https://moodle.elct.lnu.edu.ua/course/view.php?id=303">https://moodle.elct.lnu.edu.ua/course/view.php?id=303</a>	2	кінець поточного тижня
3	<b>Лабораторна робота 2:</b> Робота з URL	Лабораторна робота	Сайт курсу - <a href="https://moodle.elct.lnu.edu.ua/course/view.php?id=303">https://moodle.elct.lnu.edu.ua/course/view.php?id=303</a>	2	кінець поточного місяця
4	<b>Типи, змінні та функції у JavaScript. JSON.</b> Оголошення змінних JavaScript. Оголошення функцій у JavaScript. Типи даних у JavaScript. JSON та його зв'язок з JavaScript. Перетворення текстових рядків у JSON та назад. Вибрані конструкції мови.	Лекція	[*6], [*3], [*4] Сайт курсу - <a href="https://moodle.elct.lnu.edu.ua/course/view.php?id=303">https://moodle.elct.lnu.edu.ua/course/view.php?id=303</a>	2	кінець поточного тижня
4	<b>Лабораторна робота 3:</b> Робота з JSON та читання/запис файлів.	Лабораторна робота	Сайт курсу - <a href="https://moodle.elct.lnu.edu.ua/course/view.php?id=303">https://moodle.elct.lnu.edu.ua/course/view.php?id=303</a>	4	кінець поточного місяця

			<a href="#">d=303</a>		
5	<b>Функції зворотнього виклику у JavaScript. Модулі: CommonJS та ESM</b> Поняття про функції зворотнього виклику. Функціональні визази та стрілкові функції у JavaScript. Що таке модулі та як їх використовують. Підключення вбудованих модулів Node JS у програмі.	Лекція	[*3], [*4] Сайт курсу - <a href="https://moodle.ct.lnu.edu.ua/course/view.php?id=303">https://moodle.ct.lnu.edu.ua/course/view.php?id=303</a>	2	кінець поточного тижня
6	<b>Обробка подій у Node.JS. Структура HTTP запити. Найпростіший HTTP server на Node.JS.</b> Поняття про обробку подій та асинхронність у Node JS. Структура HTTP запити та відповіді. Методи HTTP та статусні коди. Приклад найпростішого серверу на Node JS	Лекція	[*4], [*5] Сайт курсу - <a href="https://moodle.ct.lnu.edu.ua/course/view.php?id=303">https://moodle.ct.lnu.edu.ua/course/view.php?id=303</a>	2	кінець поточного тижня
6	<b>Лабораторна робота 4:</b> Створення найпростішого серверу.	Лабораторна робота	Сайт курсу - <a href="https://moodle.ct.lnu.edu.ua/course/view.php?id=303">https://moodle.ct.lnu.edu.ua/course/view.php?id=303</a>	4	кінець поточного місяця
7	<b>Деякі інструменти для налагодження запитів. Основні типи запитів HTTP та їх приклади з curl.</b> Втористання сURL. Використання Fiddler. Використання Postman. Приклади.	Лекція	[*6], [*8], [*9], Сайт курсу - <a href="https://moodle.ct.lnu.edu.ua/course/view.php?id=303">https://moodle.ct.lnu.edu.ua/course/view.php?id=303</a>	2	кінець поточного тижня
8	<b>Встановлення сторонніх модулів. прп та файл package.json. XML та робота з ним.</b> Знайомство з Node.js Package Manager. Структура файлу package.json. XML та його використання. Робота з XML у JavaScript. Порівняння XML та JSON.	Лекція	[*6], [*8], [*9], Сайт курсу - <a href="https://moodle.ct.lnu.edu.ua/course/view.php?id=303">https://moodle.ct.lnu.edu.ua/course/view.php?id=303</a>	2	кінець поточного тижня
8	<b>Лабораторна робота 5:</b> Робота з XML.	Лабораторна робота	Сайт курсу - <a href="https://moodle.ct.lnu.edu.ua/course/view.php?id=303">https://moodle.ct.lnu.edu.ua/course/view.php?id=303</a>	4	кінець поточного місяця
9	<b>Використання Express.JS.</b> Ставорення веб серверу з використанням ExpressJS. Приклади обробки різних запитів.	Лекція	[*11] Сайт курсу - <a href="https://moodle.ct.lnu.edu.ua/course/view.php?id=303">https://moodle.ct.lnu.edu.ua/course/view.php?id=303</a>	2	кінець поточного тижня
10	<b>Поняття про Web API.</b> Визначення понять інтерфейс, API, Web API, веб ресурс. Поняття про REST. Правила “хорошого тону” для найменування веб ресурсів для створення REST Web API	Лекція	[*6] Сайт курсу - <a href="https://moodle.ct.lnu.edu.ua/course/view.php?id=303">https://moodle.ct.lnu.edu.ua/course/view.php?id=303</a>	2	кінець поточного тижня
10	<b>Лабораторна робота 6:</b> Створення веб серверу з ExpressJS.	Лабораторна робота	Сайт курсу - <a href="https://moodle.ct.lnu.edu.ua/course/view.php?id=303">https://moodle.ct.lnu.edu.ua/course/view.php?id=303</a>	4	кінець поточного місяця
11	<b>Поняття про автентифікацію.</b> Що таке автентифікація і як автентифікація відрізняється від	Лекція	[*6], Сайт курсу - <a href="https://moodle.ct.lnu.edu.ua/course/view.php?id=303">https://moodle.ct.lnu.edu.ua/course/view.php?id=303</a>	2	кінець поточного тижня

	авторизації. Базова HTTP автентифікація. Автентифікація на базі сесій. Автентифікація на базі JWT токена. Сесії та куки.		<a href="https://moodle.ct.lnu.edu.ua/course/view.php?id=303">ct.lnu.edu.ua/course/view.php?id=303</a>		
12	<b>Використання Swagger.</b> Знайомство зі Swagger. Використання Swagger для документування Web API.	Лекція	[*10] Сайт курсу - <a href="https://moodle.ct.lnu.edu.ua/course/view.php?id=303">https://moodle.ct.lnu.edu.ua/course/view.php?id=303</a>	2	кінець поточного тижня
12	<b>Лабораторна робота 7:</b> Розробка веб-серверу для системи інвентаризації	Лабораторна робота	Сайт курсу - <a href="https://moodle.ct.lnu.edu.ua/course/view.php?id=303">https://moodle.ct.lnu.edu.ua/course/view.php?id=303</a>	8	кінець поточного тижня
13	<b>Протокол HTTPS</b> Симетричне та асиметричне шифрування. Алгоритм цифрового підпису. Протокол TLS. Безпека HTTP з'єднання.	Лекція	[*6] Сайт курсу - <a href="https://moodle.ct.lnu.edu.ua/course/view.php?id=303">https://moodle.ct.lnu.edu.ua/course/view.php?id=303</a>	2	кінець поточного тижня
14	<b>Розгортання Веб серверів.</b> Хостинги. Розгортання Веб серверу. Доменні імена. Поняття про масштабування і основні проблеми які виникають при масштабуванні.	Лекція	[*6] Сайт курсу - <a href="https://moodle.ct.lnu.edu.ua/course/view.php?id=303">https://moodle.ct.lnu.edu.ua/course/view.php?id=303</a>	2	кінець поточного тижня
15	<b>Стиль програмування та патерни, які використовують при створенні Веб сервера.</b> Принципи програмування: DRY, KISS, YAGNI. Обрані архітектурні патерни.	Лекція	Сайт курсу - <a href="https://moodle.ct.lnu.edu.ua/course/view.php?id=303">https://moodle.ct.lnu.edu.ua/course/view.php?id=303</a>	2	кінець поточного тижня
16	<b>Підсумкове заняття.</b>	Лекція	Сайт курсу - <a href="https://moodle.ct.lnu.edu.ua/course/view.php?id=303">https://moodle.ct.lnu.edu.ua/course/view.php?id=303</a>	2	кінець поточного тижня
16	<b>Консультація перед екзаменом.</b>	Лабораторна робота	Сайт курсу - <a href="https://moodle.ct.lnu.edu.ua/course/view.php?id=303">https://moodle.ct.lnu.edu.ua/course/view.php?id=303</a>	2	кінець поточного тижня