

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Львівський національний університет імені Івана Франка**  
**Факультет прикладної математики та інформатики**  
**Кафедра програмування**

**Затверджено**

На засіданні кафедри програмування  
факультету прикладної математики  
та інформатики  
Львівського національного університету  
імені Івана Франка  
(протокол № 1 від 29 серпня 2023 р. )



Зав. кафедри к. ф.-м. н., доц. Ярошко С. А.

**Силабус навчальної дисципліни**  
**«Програмування та підтримка веб застосувань»,**  
**що викладається в межах ОПШ «Інформатика»**  
**першого (бакалаврського) рівня вищої освіти**  
**для здобувачів зі спеціальності 122 Комп'ютерні науки**

**Львів 2023 р.**

<b>Назва дисципліни</b>	Програмування та підтримка веб застосувань
<b>Адреса викладання дисципліни</b>	Львівський національний університет імені Івана Франка, вул. Університетська 1, м. Львів, Україна, 79000
<b>Факультет та кафедра, за якою закріплена дисципліна</b>	Факультет прикладної математики та інформатики, кафедра програмування
<b>Галузь знань, шифр та назва спеціальності</b>	Галузь знань: 12 Інформаційні технології Спеціальність: 122 Комп'ютерні науки
<b>Викладачі дисципліни</b>	Вовк Олександр Володимирович, к. ф.-м. н., асистент кафедри програмування; Васишин Богдан., асистент кафедри інформаційних систем
<b>Контактна інформація викладачів</b>	Електронна пошта: <a href="mailto:oleksandr.vovk@lnu.edu.ua">oleksandr.vovk@lnu.edu.ua</a> , веб-сторінки: <a href="https://ami.lnu.edu.ua/employee/vovk-o-v">https://ami.lnu.edu.ua/employee/vovk-o-v</a>
<b>Консультації з питань навчання по дисципліні відбуваються</b>	Консультації проводять раз на тиждень згідно з оприлюдненим розкладом консультацій викладача або частіше за домовленістю через електронне листування. Можливі онлайн консультації через Microsoft Teams.
<b>Інформація про дисципліну</b>	Курс «Програмування та підтримка веб застосувань» є нормативною дисципліною зі спеціальності 122 Комп'ютерні науки (інформатика) для освітньої програми «Інформатика», яку викладають у п'ятому семестрі.
<b>Коротка анотація дисципліни</b>	Курс містить такі основні складові: Front-end, Back-end та DevOps, які необхідні для забезпечення повного циклу розробки веб застосування починаючи від концепції, закінчуючи реалізацією готового продукту. Front–end (інтерфейс взаємодії системи з користувачем): Html, CSS, JavaScript, JavaScript frameworks, ActionScript , Chrome Developer tools та ін. Back-end (реалізація логіки взаємодії системи з користувачем та даними): Client-server Model, JSON, Layered Architecture, Web Server (Nginx), Application Server and DataBases, Nodejs (NPM, Express), HTTP protocol, HTTPS, WebSocket, архітектурний шаблон MVC, архітектурний стиль REST та ін. DevOps (практики для взаємointegraції робочих процесів): контроль версій (GitHub), генерація документації (Sphinx), автоматизація (Ansible), Continuous Integration (Jenkins), Cloud (MS Azure), Containers (Docker), OSI Model та ін. Більшість технологій вивчається оглядово, однак основна увага акцентується на оволодінні базовими технологіями для розробки трьохшарового веб застосування: Html/ CSS/JS, Client-Server, Web Server (Nginx), Application Server (NodeJS), HTTP/ HTTPS, MVC, REST.
<b>Мета та цілі дисципліни</b>	Метою курсу є вивчення основ та сучасних підходів до розробки, підтримки та розгортання веб застосувань. Ціллю курсу є досягнути достатнього рівня знань та практичних навичок для створення тришарового веб застосунку на належному для сучасних вимог рівні (застосування відомих архітектур, паттернів, фреймворків, засобів захисту інформації, інструментів розгортання та моніторингу, використання cloud технологій та ін.)
<b>Література для вивчення дисципліни</b>	<i>Основна література:</i> <ul style="list-style-type: none"> <li>• Fender J., Young C. Front-End Fundamentals. Leanpub, 2015.</li> <li>• Dean J. Web programming with HTML5, CSS, and JavaScript. Jones &amp; Bartlett Learning, 2019.</li> <li>• Fjordvald M., Nedelcu C. Nginx HTTP Server. Fourth Edition. Packt Publishing Ltd, 2018.</li> <li>• Spruy R. Advanced Game Design with HTML5 and JavaScript. Apress Media, 2015.</li> <li>• Crockford D. JavaScript: The Good Parts. O'Reilly Media, 2008.</li> <li>• Thomas S.-A. SSL&amp;TLS Essentials, John Wiley &amp; Sons, Inc. 2000.</li> <li>• Oppliger R. SSL and TLS Theory and Practice. Artech House, 2009.</li> </ul>

	<ul style="list-style-type: none"> <li>• Flanagan D. JavaScript. The Definitive Guide. O'Reilly Media, 2011.</li> <li>• Stefanov S. JavaScript Patterns. O'Reilly Media, Inc, USA, 2010.</li> <li>• Means G. Node for Frond-End Developers, O'Reilly Media, 2012.</li> <li>• Kiessling M. The Node Craftsman Book. An advanced Node.js tutorial. Leanpub, 2015.</li> <li>• Satheesh M., D'mello B.-J., Krol J. Web Development with MongoDB and NodeJS. Second Edition. Packt Publishing Ltd, 2015.</li> <li>• REST: From Research to Practice. Editors: Pautasso C., Wilde E. Springer Science+Business Media, LLC 2011</li> <li>• David Hunter, Kurt Cagle, Chris Dix, Roger Kovack, Jonathan Pinnock, Jeff Rafter - Beginning XML. XML schemas, SOAP, XSLT, DOM, and SAX 2.0-Wrox, 2001.</li> <li>• Robert Daigneau. Service Design Patterns. Fundamental Design Solutions for SOAP WSDL and RESTful Web Services. Addison Wesley Professional, 2011.</li> </ul> <p><i>Допоміжна література:</i></p> <ul style="list-style-type: none"> <li>• Pollard B. HTTP/2 in Action. Manning Publications, 2019.</li> <li>• DuRocher D. HTML &amp; CSS QuickStart Guide. The Simplified Beginners Guide to Developing a Strong Coding Foundation, Building Responsive Websites, and Mastering the Fundamentals of Modern Web Design. 2021.</li> <li>• Belt M.-W. Web Development for beginners. Learn HTML_CSS_Javascript step by step with this Coding Guide, Programming Guide for beginners, Website development. 2020</li> <li>• REST: Advanced Research Topics and Practical Applications. Editors: Pautasso C., Wilde E., Alarcon R. Springer New York Heidelberg Dordrecht London, 2014.</li> <li>• Weikum G., Schek H.-J. Concepts and Applications of Multilevel Transactions and Open Nested Transactions. Department of Computer Science Information Systems – Databases Swiss Federal Institute of Technology Zurich CH-8092 Zurich, Switzerland, 2001.</li> <li>• Encyclopedia of Database Systems. Editors: Liu L., Ozsu M.-T. Springer Science+Business Media, LLC 2009.</li> </ul>
<b>Обсяг курсу</b>	16 години лекцій, 32 години лабораторних занять, 42 години самостійної роботи, всього 90 годин, або 3 кредити ECTS
<b>Очікувані результати навчання</b>	<p>Після завершення цього курсу студент буде:</p> <p><i>знати</i></p> <ul style="list-style-type: none"> <li>• Особливості архітектури клієнт-сервер та її розширень.</li> <li>• Синтаксис та засоби мови програмування Javascript, зокрема «клієнтську» його частину.</li> <li>• Засоби та можливості середовищ для програмування, підтримки, автоматичного розгортання та т.з. «горизонтального розширення» веб застосувачів.</li> <li>• Принципи забезпечення конфіденційності каналу передачі даних в архітектурі клієнт-сервер. Симетричне, асиметричне шифрування, цифровий підпис.</li> <li>• Основи автоматизованої підтримки веб застосувачів (Continuous Integration, Delivery, Deployment. Docker).</li> <li>• Особливості HTTP протоколу та можливості нових його версій.</li> <li>• Основні принципи роботи т.з. балансування навантаження (load balancing) для розподілених веб застосунків.</li> </ul> <p><i>вміти</i></p> <ul style="list-style-type: none"> <li>• Запрограмувати та розгорнути веб застосунок з, як мінімум, трьохшаровою архітектурою.</li> <li>• Забезпечити захищений канал взаємодії з сервером (TLS), в тому числі і повноцінний двонаправлений (web socket).</li> <li>• Використовувати можливості та переваги Html5 стандарту (web worker, web socket, local storage і т.д.)</li> <li>• Забезпечити рівномірне розподілення запитів/навантаження між (як мінімум двома) серверами.</li> <li>• Забезпечити взаємодію веб застосунку з базою даних.</li> </ul>

<b>Компетентності</b>	<p><i>Інтегральна:</i> Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми у галузі комп'ютерних наук або у процесі навчання, що передбачають застосування теорій та методів інформаційних технологій і характеризується комплексністю та невизначеністю умов.</p> <p><i>Загальні (ЗК):</i></p> <p>ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.  ЗК2. Здатність застосовувати знання у практичних ситуаціях.  ЗК3. Знання та розуміння предметної області та розуміння професійної діяльності.  ЗК6. Здатність вчитися й оволодівати сучасними знаннями  ЗК7. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.  ЗК8. Здатність генерувати нові ідеї (креативність).  ЗК12. Здатність оцінювати та забезпечувати якість виконуваних робіт.</p> <p><i>Спеціальні (фахові, предметні) компетентності (СК):</i></p> <p>СК8. 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.  СК9. 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.  СК10. Здатність застосовувати методології, технології та інструментальні засоби для управління процесами життєвого циклу інформаційних і програмних систем, продуктів і сервісів інформаційних технологій відповідно до вимог замовника.  СК12. Здатність забезпечити організацію обчислювальних процесів в інформаційних системах різного призначення з урахуванням архітектури, конфігурування, показників результативності функціонування операційних систем і системного програмного забезпечення.</p>		
<b>Програмні результати навчання</b>	<p>ПР9. Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук.  ПР10. 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 вебпрограмування  ПР15. 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>		
<b>Ключові слова</b>	<p>Клієнт-сервер архітектура, HTTP, GET method, POST method, Ajax, HTTPS, HTTP 2.0, JSON, YAML, XML, Web Socket, Secure Web Socket, TLS, SSL, public key, private key, host, server, DNS, Cookies, HTML5, об'єктна модель документа (DOM), CSS3, Javascript, JQuery, Createjs, Qooodoo, Nginx, load balancer, NodeJS, npm, Express, REST, MVC, Docker контейнери, хмарні (cloud) технології, MS Azure, IaaS, PaaS, SaaS, SECaaS, MBaaS, Serverless computing, Chrome Developer tools, Continuous Integration, Delivery, Deployment, GitHub, GitLab, Perforce, Ansible, WebPack, Sphinx, Jenkins, PostgreSQL, MongoDB, MS MySQL.</p>		
<b>Формат курсу</b>	Очний		
<b>Теми</b>		Лекції	
	Тиж-день	назва і зміст теми	
	1	<b>Огляд курсу. Огляд Front-end розробки.</b> Клієнт сервер архітектура. HTML/CSS/Javascript/JQuery. Інструменти розробника. TypeScript, AngularJS, WebPack, Qooodoo, Createjs, Action Script 3.0.	
		Самостійна робота	
			К-сть год.
			2
			5

	2	<b>Огляд Back-end розробки.</b> Трьохшаровий та багат шаровий архітектурні підходи до розробки веб застосунку. Огляд Node.js, розбір коду простого клієнта та сервера. HTTP протокол. Express фреймворк. Nginx сервер та його налаштування. Приєднання до баз даних PostgreSQL, MySQL, MongoDB. Розбір back-end частини простого веб застосунку CRUD типу.	2
		Самостійна робота	5
	3	<b>Огляд DevOps практик та інструментів.</b> Continuous Integration/Delivery/Deployment та їх автоматизація. Ansible. Jenkins. Хмарні (cloud) технології. MS Azure. Load balancing. IaaS, PaaS, SaaS, SECaaS, MBaaS, Serverless computing. Віртуальні машини, контейнери та їхні особливості. Розбір коду для приєднання до хмарного сервісу розпізнавання обличчя. OSI модель.	2
		Самостійна робота	5
	4	<b>Огляд розробки особливостей статичного веб сайту та його розгортання.</b> HTML, особливості HTML5, механізм розбору HTML документу та побудови об'єктної моделі документу DOM. Лінування скриптів, стилів, зображень і т.д. Каскадні таблиці стилів CSS, селектори, media queries, CSS3 – розбір прикладів анімування об'єктів (css3maker.com). Особливості роботи веб сервера зі статичним та динамічним контентом. Властивості HTTP протоколу. Налаштування та оптимізація Nginx сервера для хочтингу статичного контенту. Огляд коду шаблонного веб сайту. Оптимізація для мобільних пристроїв. GET та POST запити. Технологія Ajax.	2
		Самостійна робота	5
	5	<b>Протокол HTTP 1.0/1.1/2.0, HTTPS.</b> Властивості та відмінності. Методи (Get, Post та ін). Розбір вмісту пакету даних з запиту (request) та відповіді (response). Розгляд основних заголовків пакету HTTP. Заголовок Upgrade для оновлення протоколу. Дослідження реального трафіку HTTP пакетів за допомогою Google Dev Tools на прикладі веб застосунку з лекції 2. Оптимізація трафіку та навантаження на сервер. TLS та SSL протоколи різних версій, розбір протоколу «рукостискання» (handshake), симетричне та асиметричне шифрування, цифровий підпис та сертифікат. Тип атаки на канал зв'язку «людина посередині» (man-in-the-middle). Налаштування Nginx сервера для роботи через захищений HTTPS протокол. Приклад генерації ssl сертифікатів за допомогою WIN32 OpenSSL.	2
		Самостійна робота	5
	6	<b>Двонаправлений канал передачі даних за допомогою протоколів Web Socket та Secure Web Socket.</b> Типи напів двонаправленої (half duplex) комунікації: polling, long polling, HTTP Streaming. Огляд особливостей протоколу RFC 6455 web socket. Процедура рукостискання (handshake) для переходу на web socket. Розгляд демонстраційного коду веб застосунку, який використовує Web Socket протокол для чату. Налаштування підтримки сервером Nodejs захищеного Secure Web Socket. Перегляд та дослідження трафіку Web Socket за допомогою інструментів розробки браузерів (на прикладі Chrome dev tools).	2
		Самостійна робота	5
	7	<b>Ядро (Core) JavaScript. Класи та модулі. Клієнтський JavaScript.</b> Динамічна типізація, глобальний об'єкт, об'єкти обгортки, приведення типів, scope chain, функції, розріджені масиви, замикання (closures), метод bind. Класи та прототипи, функції-конструктори, позичання методів, дочірні та абстрактні класи, інкапсуляція, модулі, регулярні вирази. Огляд веб застосунку для обробки регулярних виразів. Об'єкт window,	2
		Самостійна робота	5

	підключення та виконання javascript в html-документі, безпека клієнтського javascript, same-origin policy, cross-site scripting, робота з історією браузера, обробка помилок, фрейми (frames), доступ до елементів DOM моделі та стилів, html форми, обробка подій, XMLHttpRequest об'єкт, JSONP запити, Web Storage, Cookies, доступ до файлової системи та його обмеження.	
	Самостійна робота	6
8	<b>Огляд NodeJS.</b> Веб сервер та клієнт (HTTP модуль), розбір прикладів. Глобальний об'єкт. Buffer та декодування. Обробка подій. Асинхронні виклики. Перехоплення вийнятків та обробка помилок. Модульна система NodeJS (Async, Commander, and Underscore). Потоки та Pipes. Доступ до файлової системи сервера. Сокети та безпека. Хешування паролів. Дочірні процеси – огляд прикладів. Express фреймворк.	2
	Самостійна робота	6
Всього		16
го		42
<b>Лабораторні роботи</b>		
1	Робота з репозиторіями. GitLab, GitHub. Схема gitflow та Continuous Integration. Обговорення організації та оформлення практичних завдань для їхнього індивідуального захисту.	2
2	Обговорення особливостей та вимог завдання про створення тришарового веб застосунку «Інтернет Магазин». Розгляд кількох типів архітектури такого застосунку. Приклади HTML сторінок та приєднання (linking) до них ресурсів. Об'єкт window.	2
3	Вивчення особливостей програмування за допомогою Javascript на прикладах. Особлива увага приділена відмінностям від мов програмування, які були вивчені у попередніх семестрах C++, C#, Java, Python та ін.	2
4	Клієнтський Javascript на прикладах. Розгляд прикладів обмежень безпеки, фрейми. Обговорення обраних студентами підходів та набору (стеку) технологій для розробки застосунку «Інтернет Магазин». Формат серіалізації js-об'єктів JSON.	2
5	HTTP протокол та його особливості. Методи GET та POST. Відправка запитів за допомогою форм та технології Ajax. Об'єкт XMLHttpRequest. Налаштування Nginx сервера для хостингу статичного контенту та проксування запитів на динамічний контент.	2
6-7	NodeJS – програмування back-end частини, модулі для доступу до баз даних. Обробка запитів, генерація html сторінок на основі шаблонів (rendering). Express фреймворк.	4
8-9	Індивідуальний захист розробленого студентами веб застосунку «Інтернет Магазин». Обговорення недоліків архітектури та підходів на прикладах завдань та варіантів їхнього вирішення.	4
10	Обговорення особливостей та вимог завдання про створення веб застосунку для розподілених обчислень. Приклади архітектурного стилю REST.	2
11	Принцип роботи load balancing на основі налаштувань сервера Nginx. Архітектура черги завдань, які очікують виконання.	2
12	Генерація сертифікатів для налаштування захищеного каналу зв'язку HTTPS та приєднання їх до налаштувань сервера на прикладі серверів NodeJS та Nginx. Особливості TLS протоколу.	2
13	Хмарні сервіси, приклади роботи з ними за допомогою Javascript. Налаштування docker контейнерів.	2

	14-16	Індивідуальний захист веб застосунку для розподілених обчислень. Обговорення недоліків архітектури та підходів на прикладах завдань та варіантів їхнього вирішення.	6
	Всього		32
<b>Підсумковий контроль, форма</b>	залік в кінці семестру		
<b>Пререквізити</b>	Для вивчення курсу студенти потребують базових знань з дисциплін “Програмування”, “Алгоритми та структури даних”, “Бази даних”, “Комп’ютерні мережі”.		
<b>Навчальні методи та техніки, які використовують під час викладання курсу</b>	Презентації, лекції, демонстрації основних технологій та сервісів, виконання та оцінювання індивідуальних завдань, самостійна робота з вивчення переліку електронних ресурсів.		
<b>Необхідне обладнання</b>	Комп’ютер із операційною системою Windows/Linux, програми для роботи з мовами розмітки Notepad++, встановлений офісний пакет Microsoft Office/OpenOffice, доступ до мережі Інтернет, проектор для демонстрації слайдів.		
<b>Критерії оцінювання (окремо для кожного виду навчальної діяльності)</b>	<b>Оцінювання:</b>		
	Назва	Кількість балів	
	Колоквіуми за лекційним курсом	50	
	Практичні завдання	50	
	Всього	100	
<p>Основним принципом системи оцінювання є рівноправність теоретичної та практичної частин, кожна з яких оцінюється від 0 до 50 балів (50-ти бальна шкала). Варто відзначити, що практичні завдання та теоретична частина, задля зручності, можуть оцінюватись за довільною бальною шкалою, однак результат, у підсумку, буде прямо пропорційно приведено до 50-ти бальної шкали оцінювання. Наприклад, кожному з 14 лабораторних робіт можна оцінювати від 0 до 5, а набрану суму балів <math>\times 5/7</math>.</p> <p><b>Академічна доброчесність:</b> Очікується, що роботи студентів будуть їх оригінальними дослідженнями чи міркуваннями. Списування, втручання в роботу інших студентів становлять, але не обмежують, приклади можливої академічної недоброчесності. Виявлення ознак академічної недоброчесності в індивідуальних завданнях студента є підставою для її незарахування викладачем, незалежно від масштабів плагіату чи обману. Жодні форми порушення академічної доброчесності не толеруються.</p> <p><b>Відвідання занять</b> є важливою складовою навчання. Очікується, що всі студенти відвідають усі лекції та практичні заняття курсу. Студенти повинні інформувати викладача про неможливість відвідати заняття.</p> <p><b>Політика виставлення балів.</b> Враховуються бали поточної успішності, самостійної роботи та бали підсумкового тестування. При цьому обов’язково враховуються списування та плагіат; несвоєчасне виконання поставленого завдання тощо.</p>			
<b>Опитування</b>	Анкету-оцінку з метою оцінювання якості курсу буде надано після завершення курсу.		