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
Львівського національного університету
імені Івана Франка
(протокол № 1 від 31 серпня 2021 р.)

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

Силабус навчальної дисципліни

«Програмування та підтримка веб застосувань»,

що викладається в межах ОПП “Інформатика”

першого (бакалаврського) рівня вищої освіти

для здобувачів зі спеціальності 122 Комп’ютерні науки

Львів 2021 р.

Назва дисципліни	Програмування та підтримка веб застосувань
Адреса викладання дисципліни	Львівський національний університет імені Івана Франка, вул. Університетська 1, м. Львів, Україна, 79000
Факультет та кафедра, за якою закріплена дисципліна	Факультет прикладної математики та інформатики, кафедра програмування
Галузь знань, шифр та назва спеціальності	Галузь знань: 12 Інформаційні технології Спеціальність: 122 Комп'ютерні науки Спеціалізація: Інформатика
Викладачі дисципліни	Вовк Олександр Володимирович, к. ф.-м. н., асистент кафедри програмування; Василишин Богдан., асистент кафедри інформаційних систем
Контактна інформація викладачів	Електронна пошта: oleksandr.vovk@lnu.edu.ua, веб-сторінки: https://ami.lnu.edu.ua/employee/vovk-o-v
Консультації з питань навчання по дисципліні відбуваються	Консультації проводять раз на тиждень згідно з оприлюдненим розкладом консультацій викладача або частіше за домовленістю через електронне листування. Можливі онлайн консультації через Microsoft Teams.
Інформація про дисципліну	Курс «Програмування та підтримка веб застосувань» є нормативною дисципліною зі спеціальності 122 Комп'ютерні науки (інформатика) для освітньої програми «Інформатика», яку викладають у пятому семестрі.
Коротка анотація дисципліни	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 взаємоінтеграції 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.
Мета та цілі дисципліни	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 (застовування відомих архітектур, паттернів, фреймворків, засобів захисту інформації, інструментів розгортання та моніторингу, використання cloud технологій та ін.)
Література для вивчення дисципліни	*Основна література:* • Fender J., Young C. Front-End Fundamentals. Leanpub, 2015. • Dean J. Web programming with HTML5, CSS, and JavaScript. Jones & Bartlett Learning, 2019. • Fjordvald M., Nedelcu C. Nginx HTTP Server. Fourth Edition. Packt Publishing Ltd, 2018. • Spruy R. Advanced Game Design with HTML5 and JavaScript. Apress Media, 2015. • Crockford D. JavaScript: The Good Parts. O'Reilly Media, 2008.

	• Thomas S.-A. SSL&TLS Essentials, John Wiley & Sons, Inc. 2000. • Oppliger R. SSL and TLS Theory and Practice. Artech House, 2009. • Flanagan D. JavaScript. The Definitive Guide. O'Reilly Media, 2011. • Stefanov S. JavaScript Patterns. O'Reilly Media, Inc, USA, 2010. • Means G. Node for Frond-End Developers, O'Reilly Media, 2012. • Kiessling M. The Node Craftsman Book. An advanced Node.js tutorial. Leanpub, 2015. • Satheesh M., D'mello B.-J., Krol J. Web Development with MongoDB and NodeJS. Second Edition. Packt Publishing Ltd, 2015. • REST: From Research to Practice. Editors: Pautasso C., Wilde E. Springer Science+Business Media, LLC 2011 *Допоміжна література:* • Pollard B. HTTP/2 in Action. Manning Publications, 2019. • DuRocher D. HTML & 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. • 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 • REST: Advanced Research Topics and Practical Applications. Editors: Pautasso C., Wilde E., Alarcon R. Springer New York Heidelberg Dordrecht London, 2014. • 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. • Encyclopedia of Database Systems. Editors: Liu L., Ozsu M.-T. Springer Science+Business Media, LLC 2009.
Обсяг курсу	16 години лекцій, 32 години лабораторних занять
Очікувані результати навчання	Після завершення цього курсу студент буде: *знати* ● Особливості архітектури клієнт-сервер та її розширень. ● Синтаксис та засоби мови програмування Javascript, зокрема «клієнтську» його частину. ● Засоби та можливості середовищ для програмування, підтримки, автоматичного розгортання та т.з. «горизонтального розширення» веб застосувань. ● Принципи забезпечення конфіденційності каналу передачі даних в архітектурі клієнт-сервер. Симетричне, асиметричне шифрування, цифровий підпис. ● Основи автоматизованої підтримки веб застосувань (Continuous Integration, Delivery, Deployment. Docker). ● Особливості HTTP протоколу та можливості нових його версій. ● Основні принципи роботи т.з. балансування навантаження (load balancing) для розподілених веб застосунків. *вміти* ● Запрограмувати та розгорнути веб застосунок з, як мінімум, трьохшаровою архітектурою. ● Забезпечити захищений канал взаємодії з сервером (TLS), в тому чисті і повноцінний двонаправлений (web socket). ● Використовувати можливості та переваги Html5 стандарту (web worker, web socket, local storage і т.д.) ● Забезпечити рівномірне розподілення запитів/навантаження між (як мінімум двома) серверами. ● Забезпечити взаємодію веб застосунку з базою даних.

Ключові слова	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, Qooxdoo, 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.
Формат курсу	Проведення лекцій, лабораторних робіт та консультацій в приміщеннях університету (в умовах карантину – онлайн на платформі Microsoft Teams)
Теми	

	Лекції	
Тиж-день	назва і зміст теми	К-сть год.
1	**Огляд курсу. Огляд Front-end розробки.** Клієнт сервер архітектура. HTML/CSS/Javascript/JQuery. Інстременти розробника. TypeScript, AngularJS, WebPack, Qooxdoo, Createjs, Action Script 3.0.	2
2	**Огляд Back-end розробки.** Трьохшаровий та багатошаровий 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
3	**Огляд DevOps практик та інструментів.** 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
4	**Огляд розробки особливостей статичного веб сайту та його розгортання.** HTML, особливості HTML5, механізм розбору HTML документу та побудови об'єктної моделі документу DOM. Лінкування 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
5	**Протокол HTTP 1.0/1.1/2.0, HTTPS.** Властивості та відмінності. Методи (Get, Post та ін). Розбір вмісту пакету даних з запиту (request) та відповіді (response).	2

		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.	
	6	**Двонаправлений канал передачі даних за допомогою протоколів Web Socket та Secure Web Socket.** 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 бравзера (на прикладі Chome dev tools).	2
	7	**Ядро (Core) JavaScript. Класи та модулі. Клієнтський JavaScript.** 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, підключення та виконання javascript в html-документі, безпека клієнтського javascript, same-origin policy, cross-site scripting, робота з історією бравзера, 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.	2
	8	**Огляд NodeJS.** 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
	Всього		16
	Лабораторні роботи		

	1	Робота з репозиторіями. GitLab, GitHub. Схема gitflow та Continuous Integration. Обговорення організації та оформлення практичних завдань для їхнього індивідувального 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 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

<table>
<tr><td></td><td>

Оцінювання:

<table>
<tr><td>Назва</td><td>Кількість балів</td></tr>
<tr><td>Колоквіуми за лекційним курсом</td><td>50</td></tr>
<tr><td>Практичні завдання</td><td>50</td></tr>
<tr><td>Всього</td><td>100</td></tr>
</table>

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 прямопропорційно приведено до 50-ти бальної шкали оцінювання.

</td></tr>
<tr><td>Підсумковий контроль, форма</td><td>залік в кінці семестру</td></tr>
<tr><td>Пререквізити</td><td>Для вивчення курсу студенти потребують базових знань з дисциплін “Програмування”, “Алгоритми та структури даних”, “Бази даних”, “Комп’ютерні мережі”.</td></tr>
<tr><td>Навчальні методи та техніки, які використовують під час викладання курсу</td><td>Презентації, лекції, демонстрації основних технологій та сервісів, виконання та оцінювання індивідуальних завдань, самостійна робота з вивчення переліку електронних ресурсів.</td></tr>
<tr><td>Необхідне обладнання</td><td>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 демонтсрації слайдів.</td></tr>
<tr><td>Критерії оцінювання (окремо для кожного виду навчальної діяльності)</td><td>Академічна доброчесність: 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 незарахуванння викладачем, незалежно від масштабів плагіату чи обману.
Відвідання занять є важливою складовою навчання. Очікується, що всі студенти відвідають усі лекції та практичні зайняття курсу. Студенти повинні інформувати викладача про неможливість відвідати заняття. Альтернативою відвідування лекційних та лабораторних занять в університеті може бути дистанційна онлайнова робота за розкладом проведення занять за погодженням з викладачем. У будь-якому випадку студенти зобов’язані дотримуватися термінів визначених для виконання всіх видів робіт та індивідуальних завдань, передбачених курсом.
Політика виставлення балів. Враховуються бали поточної успішності, самостійної роботи та бали підсумкового тестування. При цьому обов’язково враховуються списування та плагіат; несвоєчасне виконання поставленого завдання тощо.</td></tr>
<tr><td>Опитування</td><td>Анкету-оцінку з метою оцінювання якості курсу буде надано після завершення курсу.</td></tr>
</table>