

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

**Затверджено**

На засіданні  
кафедри прикладної математики  
факультету прикладної математики та  
інформатики  
Львівського національного університету  
імені Івана Франка  
(протокол № 1 від 31 серпня 2023р. )

Завідувач кафедри



Юрій ЯЩУК

**Силабус з навчальної дисципліни  
“Основи web програмування”  
що викладається в межах першого (бакалаврського) рівня  
вищої освіти для здобувачів  
з спеціальності 113 Прикладна математика**

Львів 2023 р.

<b>Назва дисципліни</b>	Основи web програмування
<b>Адреса викладання дисципліни</b>	Головний корпус ЛНУ ім. І. Франка м. Львів, вул. Університетська 1
<b>Факультет та кафедра, за якою закріплена дисципліна</b>	Факультет прикладної математики та інформатики Кафедра прикладної математики
<b>Галузь знань, шифр та назва спеціальності</b>	11 – математика та статистика 113 – прикладна математика
<b>Викладачі дисципліни</b>	Заневич Олег Богданович, асистент кафедри прикладної математики
<b>Контактна інформація викладачів</b>	Oleh.Zanevych@lnu.edu.ua; <a href="https://ami.lnu.edu.ua/employee/oleh-zanevych">https://ami.lnu.edu.ua/employee/oleh-zanevych</a>
<b>Консультації з питань навчання по дисципліні відбуваються</b>	Консультації в день проведення лекцій/лабораторних занять (за попередньою домовленістю).
<b>Сторінка дисципліни</b>	<a href="https://ami.lnu.edu.ua/course/osnovy-web-prohramuvannia-pm">https://ami.lnu.edu.ua/course/osnovy-web-prohramuvannia-pm</a>
<b>Інформація про дисципліну</b>	Дисципліна “Основи web програмування” є дисципліною на вибір зі спеціальності 113 – прикладна математика, яка викладається в 5-му семестрі в обсязі 4 кредитів (за Європейською Кредитно-Трансферною Системою ECTS).
<b>Коротка анотація дисципліни</b>	Курс "Основи web програмування" призначений для здобуття фундаментальних знань та практичних навичок у сфері створення веб-додатків. Протягом курсу студенти дізнаються про основи мови розмітки HTML, методи стилізації CSS, а також глибоко зануряться в світ динамічних веб-застосунків з допомогою JavaScript. Особлива увага приділяється сучасним фронтенд технологіям, зокрема роботі з бібліотекою React.js. У другій частині курсу розглядається розробка серверної частини на основі Node.js та створення RESTful API. Завершивши курс, студенти будуть володіти комплексом знань і практичних навичок для розробки сучасних веб-додатків, від простих статичних веб-сайтів до складних інтерактивних веб-платформ.
<b>Мета та цілі дисципліни</b>	<b>Мета курсу</b> "Основи веб програмування" полягає в наданні студентам комплексного погляду на основні принципи веб-програмування, допомогі зрозуміти й прийняти сучасні підходи до розробки веб-додатків, а також навчити застосовувати вивчені теоретичні знання на практиці. По завершенню курсу, студенти повинні володіти навичками створення і стилізації веб-сторінок, розробки клієнтської частини веб-додатку за допомогою React.js та створення RESTful web service на Node.js. <b>Цілі курсу:</b> <ul style="list-style-type: none"> <li>• Вміти створювати, стилізувати та оптимізувати веб-сторінки з використанням сучасних технологій та підходів.</li> <li>• Розробляти взаємодію на сторінці за допомогою JavaScript та DOM API.</li> <li>• Вміти створювати Single Page Applications (SPA) з використанням React.js, розуміти принципи роботи Virtual DOM та інших особливостей бібліотеки.</li> <li>• Розробляти RESTful web services на Node.js, враховуючи найкращі практики безпеки, оптимізації та структуризації коду.</li> <li>• Успішно інтегрувати клієнтську частину веб-додатку з серверною, забезпечуючи плавну взаємодію між ними.</li> </ul>
<b>Література для вивчення</b>	<b>Інтернет ресурси:</b> 1. HTML <a href="https://developer.mozilla.org/en-US/docs/Web/HTML">https://developer.mozilla.org/en-US/docs/Web/HTML</a>

<p><b>дисципліни</b></p>	<ol style="list-style-type: none"> <li>2. CSS <a href="https://developer.mozilla.org/en-US/docs/Web/CSS">https://developer.mozilla.org/en-US/docs/Web/CSS</a></li> <li>3. JavaScript <a href="https://developer.mozilla.org/en-US/docs/Web/JavaScript">https://developer.mozilla.org/en-US/docs/Web/JavaScript</a></li> <li>4. React.js <a href="https://react.dev/learn">https://react.dev/learn</a></li> <li>5. Node.js <a href="https://nodejs.org/en/docs">https://nodejs.org/en/docs</a></li> </ol> <p><b>Література:</b></p> <ol style="list-style-type: none"> <li>6. Duckett, J. (2021). HTML and CSS: Design and Build Websites. Wiley.</li> <li>7. Eich, B. (2020). JavaScript: The New Toys. Wiley.</li> <li>8. Mead, A. (2019). The Modern JavaScript Bootcamp. Pearson.</li> <li>9. Seshadri, S., &amp; Teague, J. (2019). Learning React: A Hands-On Guide to Building Web Applications Using React and Redux (2nd ed.). Addison-Wesley.</li> <li>10. Bankras, R. (2018). Beginning Angular with Typescript (updated to Angular 6). Apress.</li> <li>11. Flanagan, D. (2020). JavaScript: The Definitive Guide (7th ed.). O'Reilly Media.</li> <li>12. Wieruch, R. (2018). The Road to React: Your journey to master plain yet pragmatic React.js. Independently published.</li> <li>13. Stearns, B. (2019). Learning React: Modern Patterns for Developing React Apps. O'Reilly Media.</li> <li>14. Cantelon, M., Harter, M., Holowaychuk, T., &amp; Rajlich, N. (2018). Node.js in Action (2nd ed.). Manning Publications.</li> <li>15. Crockford, D. (2020). How JavaScript Works. Virgule-Solidus.</li> <li>16. Haverbeke, M. (2018). Eloquent JavaScript: A Modern Introduction to Programming (3rd ed.). No Starch Press.</li> <li>17. Simpson, K. (2019). You Don't Know JS Yet: Scope &amp; Closures (2nd ed.). Independently published.</li> <li>18. Simpson, K. (2019). You Don't Know JS Yet: ES.Next &amp; Beyond (2nd ed.). Independently published.</li> <li>19. Fogel, K. (2021). Designing Great Web APIs: Creating Business Value Through Developer Experience. O'Reilly Media.</li> <li>20. Griffith, T. (2021). React Up &amp; Running: Building Web Applications. O'Reilly Media.</li> </ol>
<p><b>Обсяг курсу</b></p>	<p>Загальний обсяг: 120 годин (аудиторних занять: 64 год., з них 32 год. лекцій та 32 год. лабораторних робіт; самостійної роботи: 56 год).</p>
<p><b>Очікувані результати навчання</b></p>	<p>Після завершення цього курсу студент буде:</p> <p><b>Знати:</b></p> <ul style="list-style-type: none"> <li>- Основні принципи та стандарти веб-розмітки за допомогою HTML.</li> <li>- Принципи стилізації веб-сторінок з використанням CSS, а також основи адаптивної верстки та сучасних технік розташування елементів.</li> <li>- Основні синтаксичні конструкції, типи даних та особливості роботи мови програмування JavaScript.</li> <li>- Поняття та особливості роботи DOM, способи взаємодії із веб-сторінкою через JavaScript.</li> <li>- Концепцію Single Page Application та основи роботи з бібліотекою React.js.</li> <li>- Основи створення серверної частини веб-додатків на основі Node.js, а також принципи роботи з базами даних та розробка RESTful API.</li> </ul> <p><b>Вміти:</b></p> <ul style="list-style-type: none"> <li>- Розробляти структуровані та семантично правильні веб-сторінки з використанням HTML.</li> <li>- Стилїзувати веб-сторінки, реалізовувати адаптивний дизайн та оптимальне розміщення елементів на сторінці з використанням CSS, Flexbox та Grid.</li> </ul>

	<ul style="list-style-type: none"> <li>- Реалізувати динамічну взаємодію на сторінці, використовуючи JavaScript.</li> <li>- Створювати додатки на React.js, реалізувати взаємодію між компонентами та управляти станом додатку.</li> <li>- Розробляти RESTful API на Node.js, забезпечуючи взаємодію сервера із клієнтською частиною та базою даних.</li> <li>- Аналізувати вимоги до веб-додатку, розробляти функціональність відповідно до специфікацій та впроваджувати найкращі практики розробки для забезпечення якості та безпеки продукту.</li> </ul>																																					
<b>Ключові слова</b>	HTML, CSS, JavaScript (JS), Flexbox, Grid, псевдо класи, псевдоелементи, Media queries, Mobile first, BEM методологія, ECMAScript, DOM, Event listeners, Single page application (SPA), React.js, Virtual DOM, Context, Node.js, Express.js, REST API, Middleware, CRUD, Knex.js, Асинхронність, Promises, Життєвий цикл компонент, Event loop, Autoboxing, Unboxing, Колекції даних, Клієнт-серверна архітектура																																					
<b>Формат курсу</b>	Очний. Проведення лекцій, лабораторних робіт та консультацій																																					
<b>Теми</b>	Подано нижче у таблиці Схема курсу “Основи web програмування ”																																					
<b>Підсумковий контроль, форма</b>	Залік																																					
<b>Пререквізити</b>	<ul style="list-style-type: none"> <li>• Основи інформатики та програмування: знання основних концепцій програмування, таких як змінні, цикли, умовні оператори.</li> <li>• Базове розуміння роботи Інтернету: розуміння того, як функціонують веб-сервери та браузеры, основи HTTP.</li> <li>• Основи роботи з операційною системою: здатність встановлювати та налаштовувати програмне забезпечення, розуміння файлової системи.</li> <li>• Базове знайомство з командним рядком: вміння виконувати прості команди в командному рядку або терміналі вашої операційної системи.</li> <li>• Англійська мова на технічному рівні: більшість технічних матеріалів, документації та ресурсів доступні на англійській мові, тому базове розуміння технічного англійського є обов'язковим.</li> </ul>																																					
<b>Навчальні методи та техніки, які будуть використовуватись під час викладання курсу</b>	Презентації, лекції, семінари. Домашні та індивідуальні завдання																																					
<b>Необхідне обладнання</b>	Комп'ютер із доступом до інтернету																																					
<b>Критерії оцінювання (окремо для кожного виду навчальної діяльності)</b>	<p><b>Оцінювання</b> проводиться за 100-бальною шкалою.</p> <table border="1"> <thead> <tr> <th colspan="2" rowspan="2">Оцінка за шкалою ECTS</th> <th rowspan="2">Оцінка в балах</th> <th colspan="3">Оцінка за національною шкалою</th> </tr> <tr> <th colspan="2">Екзамен, диференційований залік</th> <th>залік</th> </tr> </thead> <tbody> <tr> <td>A</td> <td>Відмінно</td> <td>100 - 90</td> <td>Відмінно</td> <td>5</td> <td rowspan="5">зараховано</td> </tr> <tr> <td>B</td> <td>Дуже добре</td> <td>81 - 89</td> <td rowspan="2">Добре</td> <td rowspan="2">4</td> </tr> <tr> <td>C</td> <td>Добре</td> <td>71 - 80</td> </tr> <tr> <td>D</td> <td>Задовільно</td> <td>61 - 70</td> <td rowspan="2">Задовільно</td> <td rowspan="2">3</td> </tr> <tr> <td>E</td> <td>Достатньо</td> <td>51 - 60</td> </tr> <tr> <td>FX (F)</td> <td>Незадовільно</td> <td>0 - 50</td> <td>Незадовільно</td> <td>2</td> <td>не зараховано</td> </tr> </tbody> </table>	Оцінка за шкалою ECTS		Оцінка в балах	Оцінка за національною шкалою			Екзамен, диференційований залік		залік	A	Відмінно	100 - 90	Відмінно	5	зараховано	B	Дуже добре	81 - 89	Добре	4	C	Добре	71 - 80	D	Задовільно	61 - 70	Задовільно	3	E	Достатньо	51 - 60	FX (F)	Незадовільно	0 - 50	Незадовільно	2	не зараховано
Оцінка за шкалою ECTS					Оцінка в балах	Оцінка за національною шкалою																																
		Екзамен, диференційований залік		залік																																		
A	Відмінно	100 - 90	Відмінно	5	зараховано																																	
B	Дуже добре	81 - 89	Добре	4																																		
C	Добре	71 - 80																																				
D	Задовільно	61 - 70	Задовільно	3																																		
E	Достатньо	51 - 60																																				
FX (F)	Незадовільно	0 - 50	Незадовільно	2	не зараховано																																	

	<p>Студенти набирають 100 балів протягом семестру за виконання індивідуальних завдань. Загалом студенти отримують 5 індивідуальних завдань, кожне з яких оцінюється у 20 балів. При оцінюванні завдань беруться до уваги 2 критерія: реалізація функціоналу згідно з вимог поставленими у завданні (15 балів) та якість коду реалізації (5 балів). Нижче детальніше подано оцінювання кожного з критеріїв.</p> <p><b>Функціонал згідно з вимогами (максимум 15 балів)</b></p> <p><b>1-бал.</b> Студен практично нічого не зробив, але прийшов на задачу і демонструє, що він намагався виконати поставлене завдання. Однак, наявна реалізація не зодоволяє вимоги завдання.</p> <p><b>1-5 балів.</b> Мінімальна реалізація вимог. Було зроблено менше 30% з запланованого функціоналу або функціонал працює з помилками, що критично впливають на роботу застосунку.</p> <p><b>6-9 балів.</b> Часткова реалізація вимог. Виконано від 30% до 70% запланованого функціоналу. Присутні дрібні помилки, які не впливають на загальну роботу застосунку.</p> <p><b>10-13 балів.</b> Майже повна реалізація вимог. Виконано від 70% до 90% запланованого функціоналу, з мінімальними помилками або недоліками.</p> <p><b>14-15 балів.</b> Повна реалізація вимог. Всі заплановані функції реалізовані правильно та без помилок.</p> <p><b>Якість коду (максимум 5 балів)</b></p> <p><b>1 бал.</b> Код містить значні помилки, погано організований, важко читабельний, не дотримано конвенцій кодування.</p> <p><b>2-3 бали.</b> Код добре організований, але містить деякі недоліки у читабельності або недотримання конвенцій кодування. Присутня спроба організації коду, але потребує оптимізації.</p> <p><b>4 бали.</b> Код добре організований і читабельний, з мінімальними недоліками. Дотримано конвенцій кодування, код ефективно структурований.</p> <p><b>5 балів.</b> Відмінна якість коду. Код чистий, добре організований, оптимізований, без помилок. Повне дотримання конвенцій кодування і стандартів розробки.</p> <p><b>Академічна доброчесність:</b> Очікується, що роботи студентів будуть їх оригінальними дослідженнями чи міркуваннями. Відсутність посилань на використані джерела, фабрикування джерел, списування, втручання в роботу інших студентів становлять, але не обмежують, приклади можливої академічної недоброчесності. Виявлення ознак академічної недоброчесності в письмовій роботі студента є підставою для її незарахування викладачем, незалежно від масштабів плагіату чи обману.</p> <p><b>Відвідання</b> занять є важливою складовою навчання. Очікується, що всі студенти відвідають усі лекції та лабораторні заняття курсу, якщо інше не регламентується правилами узгодженими з деканатом факультету прикладної математики та інформатики. Студенти повинні інформувати викладача про неможливість відвідати заняття. У будь-якому випадку студенти зобов'язані дотримуватися термінів визначених для виконання домашніх та індивідуальних завдань, передбачених курсом.</p> <p><b>Література.</b> Уся література, яку студенти не зможуть знайти самостійно, буде надана викладачем виключно в освітніх цілях без права її передачі третім особам. Студенти заохочуються до використання також й іншої літератури та джерел, яких немає серед рекомендованих.</p>
Опитування	Оцінювання якості курсу буде доступне у системі "Dekanat" після завершення курсу.

## Схема курсу “Основи web програмування”

### Лекційні заняття

№ п/п	Тема	Тези лекції	Література та інтернет ресурси	Завдання, год
1.	Вступ до веб-розробки	<ul style="list-style-type: none"><li>• Історія Інтернету і веб-розробки.</li><li>• Основи клієнт-серверної архітектури.</li><li>• Як працює браузер.</li></ul>	[1, 6]	2
2.	Основи HTML	<ul style="list-style-type: none"><li>• Структура документа, DOCTYPE.</li><li>• Блочні та інлайнні елементи.</li><li>• Атрибути елементів. Робота з гіпертекстовими посиланнями, зображеннями.</li></ul>	[1, 6]	2
3.	Форми та таблиці у HTML	<ul style="list-style-type: none"><li>• Створення форм, їх елементи (input, textarea, select тощо).</li><li>• Таблиці та їх структуризація.</li></ul>	[1, 6]	2
4.	Введення у CSS. Селектори та базова стилізація	<ul style="list-style-type: none"><li>• Що таке CSS і як його підключити.</li><li>• Селектори, властивості та значення.</li><li>• CSS Box model.</li></ul>	[2, 6]	1
5.	Поглиблене вивчення CSS. Позиціонування та адаптивність	<ul style="list-style-type: none"><li>• Позиціонування елементів (static, relative, absolute, fixed).</li><li>• Flexbox.</li><li>• Grid.</li><li>• Media queries.</li></ul>	[2, 6]	1
6.	Поглиблена стилізація з CSS. Псевдокласи, псевдоелементи	<ul style="list-style-type: none"><li>• Селектори по атрибутах, псевдокласи, псевдоелементи.</li><li>• Транзиції та анімації.</li></ul>	[2, 6]	2
7.	Базові концепції JavaScript (JS):	<ul style="list-style-type: none"><li>• Вступ до JS: синтаксис, змінні, типи даних.</li><li>• Умовні оператори, цикли.</li></ul>	[3, 7]	1
8.	Функції та об'єкти у JS	<ul style="list-style-type: none"><li>• Декларація та виклик функцій.</li><li>• Об'єкти, створення та робота з ними.</li></ul>	[3, 7-11]	1

9.	<b>Поглиблений JS: колекції, асинхронність, ES6+ особливості</b>	<ul style="list-style-type: none"> <li>• Масиви, Set, Map.</li> <li>• Promises, асинхронні функції.</li> <li>• ES6+ основні можливості (стрілкові функції, деструктуризація тощо).</li> </ul>	[3, 7-11]	2
10.	<b>JavaScript і DOM</b>	<ul style="list-style-type: none"> <li>• Що таке DOM. Робота з елементами сторінки.</li> <li>• Обробка подій.</li> </ul>	[3, 7-11]	2
11.	<b>Вступ до React.js</b>	<ul style="list-style-type: none"> <li>• Що таке React та його переваги.</li> <li>• Створення компонентів, props, state.</li> </ul>	[4, 13]	2
12.	<b>React.js: Життєвий цикл компоненту, хуки</b>	<ul style="list-style-type: none"> <li>• Життєвий цикл класового компоненту.</li> <li>• Використання основних хуків: useState, useEffect та ін.</li> </ul>	[4]	2
13.	<b>Розширений React: роутинг, контекст, використання API</b>	<ul style="list-style-type: none"> <li>• React Router для роутингу.</li> <li>• Використання контексту для передачі даних між компонентами.</li> </ul>	[4]	2
14.	<b>Основи Node.js та Express.js:</b>	<ul style="list-style-type: none"> <li>• Що таке Node.js, його переваги та особливості.</li> <li>• Вступ до Express.js. Створення простого веб-сервера.</li> </ul>	[5, 15]	2
15.	<b>Робота з базами даних на прикладі MongoDB</b>	<ul style="list-style-type: none"> <li>• Що таке NoSQL бази даних.</li> <li>• Вступ до MongoDB, основні операції.</li> </ul>	[15]	2
16.	<b>Завершальний огляд та подальші кроки у веб-розробці</b>	<ul style="list-style-type: none"> <li>• Безпека у веб-розробці.</li> <li>• Оптимізація продуктивності та доступності веб-сторінок.</li> <li>• Вступ до інших фреймворків та інструментів.</li> </ul>		

### Лабораторні заняття

№ п/п	Тема	Короткий опис лабораторного заняття	Література та інтернет ресурси	Завдання, год
1.	<b>Створення базової веб-сторінки</b>	Студенти створюють свою першу HTML-сторінку з текстом, зображенням та посиланням	[1, 6]	2

2.	<b>Робота з формами і таблицями</b>	Створення простої контактної форми та таблиці для відображення даних	[1, 6]	2
3.	<b>Основи CSS: стилізація веб-сторінки</b>	Додавання стилів до попередньо створеної веб-сторінки: зміна фону, шрифтів, відступів.	[2, 6]	2
4.	<b>Адаптивний дизайн з медіа-запитами</b>	Адаптація попередньої веб-сторінки для різних типів екранів (мобільних, планшетів, десктопів).	[2, 6]	2
5.	<b>Анімації та транзиції в CSS</b>	Додавання анімацій до елементів веб-сторінки, експериментування з транзиціями	[2, 6]	2
6.	<b>Вступ до JavaScript: інтерактивність на сторінці</b>	Додавання простих JS-скриптів для зміни контенту сторінки, реакції на клік користувача.	[3, 7-11]	2
7.	<b>Робота з об'єктами та масивами в JS</b>	Створення та модифікація об'єктів та масивів, використання циклів для обробки даних.	[3, 7-11]	2
8.	<b>Асинхронність в JS: завдання з Promises</b>	Виконання асинхронних завдань, робота з web APIs, використання Promises	[3, 7-11]	2
9.	<b>DOM маніпуляції та обробка подій</b>	Додавання, видалення та зміна елементів DOM. Відслідковування подій користувача на сторінці.	[3, 7-11]	2
10.	<b>Створення простого додатка на React</b>	Створення базового React-дodatка з декількома компонентами, передача даних між компонентами.	[4, 13]	2
11.	<b>Реалізація роутингу в React-дodatку</b>	Додавання роутингу до React-дodatка, створення переходів між сторінками.	[4]	2
12.	<b>Робота з зовнішніми API в React</b>	Інтеграція React-дodatка з зовнішнім API, отримання та відображення даних	[4]	2
13.	<b>Вступ до Node.js: створення простого сервера</b>	Розробка простого сервера на Express.js, обробка запитів та відправка відповідей	[15]	2
14.	<b>З'єднання сервера з базою даних</b>	Інтеграція сервера на Node.js з MongoDB, збереження та отримання даних з бази.	[15]	2
15.	<b>Розробка доступних web-застосунків: основи Accessibility</b>	Студенти оптимізують веб-аплікацію таким чином, щоб нею могли користуватися люди з обмеженими можливостями на основі принципів WCAG (Web Content Accessibility Guidelines)		2
16.	<b>Здача індивідуальних завдань</b>	Завершення приймання індивідуальних завдань. Виставлення оцінок. Консультація, щодо талону (на вимогу).		