

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 мат. та  
інформатики  
Львівського національного університету  
імені Івана Франка  
(протокол № 1 від 27 серпня 2025р. )  
Завідувач кафедри \_\_Ящук  
Ю.О. \_\_\_\_\_

**Силабус з навчальної дисципліни  
“Проектування web застосунків”  
що викладається в межах ОПП “Прикладна математика”  
першого (бакалаврського) рівня вищої освіти для здобувачів з  
спеціальності 113 – прикладна математика**

**Львів 2026 р.**

<b>Назва дисципліни</b>	Проектування web застосунків
<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/практичних занять (за попередньою домовленістю).
<b>Сторінка дисципліни</b>	
<b>інформація про дисципліну</b>	Дисципліна “Проектування web застосунків” є дисципліною на вибір зі спеціальності 113 – прикладна математика для освітньої програми “Прикладна математика”, яка викладається в 6-му семестрі в обсязі 2 кредиту (за Європейською Кредитно-Трансферною Системою ECTS).
<b>Коротка анотація дисципліни</b>	<p>Курс "Проектування web застосунків" присвячений сучасним підходам і технологіям у розробці веб-застосунків. Студенти ознайомляться з процесом створення надійних та швидких RESTful та GraphQL веб-сервісів на платформі Spring у мові програмування Java. Особлива увага приділяється реактивному програмуванню за допомогою Spring WebFlux та розробці вебсокет-з'єднань для забезпечення двосторонньої комунікації між клієнтом та сервером в реальному часі.</p> <p>На фронтенд-частині курсу студенти дізнаються про розробку застосунків за допомогою популярного фреймворку Angular, охоплюючи основні аспекти, такі як компоненти, сервіси, маршрутизація та робота з формами.</p> <p>Крім теоретичної частини, студенти отримають практичний досвід завдяки численним лабораторним роботам, що дозволяє закріпити набуті знання та навички.</p> <p>Після завершення курсу студенти зможуть проектувати та реалізовувати комплексні веб-застосунки, які відповідають сучасним стандартам і вимогам ринку.</p>
<b>Мета та цілі дисципліни</b>	<b>Мета курсу.</b> Ознайомити студентів з сучасними підходами, принципами та технологіями розробки веб-застосунків, забезпечивши глибоке розуміння та практичні навички у створенні високоякісних серверних та клієнтських частин веб-додатків.

	<p><b>Цілі курсу:</b></p> <ul style="list-style-type: none"> <li>• Освоєння основних концепцій розробки веб-застосунків на платформі Spring.</li> <li>• Набуття практичних навичок у створенні RESTful та GraphQL веб-сервісів, зокрема використовуючи реактивний підхід за допомогою Spring WebFlux.</li> <li>• Закріплення знань про двосторонню комунікацію в реальному часі через розробку вебсокет-з'єднань.</li> <li>• Ознайомлення із основами фреймворку Angular та його застосуванням у розробці клієнтської частини веб-застосунків.</li> <li>• Розуміння та використання найкращих практик розробки, тестування та оптимізації веб-застосунків.</li> <li>• Розвиток навичок командної роботи та самостійного рішення проблем під час проектування та розробки веб-додатків.</li> </ul>
<p><b>Література для вивчення дисципліни</b></p>	<ol style="list-style-type: none"> <li>1. <b>Spring</b> <ul style="list-style-type: none"> <li>○ Walls, C. (2019). <i>Spring in Action</i>. Manning Publications.</li> <li>○ Long, J., &amp; Winch, R. (2020). <i>Spring Security in Action</i>. Manning Publications.</li> </ul> </li> <li>2. <b>Spring WebFlux</b> <ul style="list-style-type: none"> <li>○ Pivovarit, M. (2020). <i>Reactive Programming with Spring 5</i>. Packt Publishing.</li> <li>○ Long, J. (2019). <i>Reactive Spring</i>. Leanpub.</li> </ul> </li> <li>3. <b>GraphQL</b> <ul style="list-style-type: none"> <li>○ Porcello, E., &amp; Banks, A. (2018). <i>Learning GraphQL: Declarative Data Fetching for Modern Web Apps</i>. O'Reilly Media.</li> <li>○ Bojan, T. (2020). <i>Full-Stack GraphQL Applications with GRANDstack - Essential Excerpts</i>. Leanpub.</li> </ul> </li> <li>4. <b>Websockets</b> <ul style="list-style-type: none"> <li>○ Pimentel, V., &amp; Lourenço, J. R. (2019). <i>WebSocket: Lightweight Client-Server Communications</i>. O'Reilly Media.</li> </ul> </li> <li>5. <b>Angular</b> <ul style="list-style-type: none"> <li>○ Freeman, A. (2018). <i>Pro Angular 6</i>. Apress.</li> <li>○ Staples, M. (2019). <i>Angular for Enterprise-Ready Web Applications</i>. Packt Publishing.</li> <li>○ Kimmel, R. (2019). <i>Angular Development with TypeScript</i>. Manning Publications.</li> </ul> </li> <li>6. <b>General Web Application Design and Architecture</b> <ul style="list-style-type: none"> <li>○ Richardson, C., &amp; Smith, R. (2019). <i>Microservices Patterns</i>. Manning Publications.</li> <li>○ Newman, S. (2018). <i>Building Microservices: Designing Fine-Grained Systems</i>. O'Reilly Media.</li> <li>○ Fowler, M. (2018). <i>Patterns of Enterprise Application Architecture</i>. Addison-Wesley Professional.</li> </ul> </li> <li>7. <b>Додаткові ресурси та література</b> <ul style="list-style-type: none"> <li>○ Spring Documentation (Official): <a href="https://docs.spring.io/spring/docs/current/spring-framework-reference/">https://docs.spring.io/spring/docs/current/spring-framework-reference/</a></li> <li>○ Angular Documentation (Official): <a href="https://angular.io/docs">https://angular.io/docs</a></li> </ul> </li> </ol>

- Söderberg, B. (2019). Mastering Spring Boot 2.0: Build modern, cloud-native, and distributed systems using Spring Boot. Packt Publishing.
- Darwin, I. F. (2020). Java Cookbook: Problems and Solutions for Java Developers. O'Reilly Media.
- Long, J., & Freire, J. L. (2019). Reactive Spring: Build Cloud Native Applications with Reactive Programming. O'Reilly Media.
- Sturgeon, P. (2018). Build APIs You Won't Hate: Everyone and their dog wants an API, so you should probably learn how to build them. CreateSpace Independent Publishing Platform.
- Kocer, B. (2021). Angular for Enterprise-Ready Web Applications: Build and deliver production-grade and evergreen Angular apps at cloud-scale. Packt Publishing.
- Banks, A., & Porcello, E. (2019). Learning GraphQL: Declarative Data Fetching for Modern Web Apps. O'Reilly Media.
- Greco, A. (2020). Practical Highcharts with Angular: Your essential guide to creating real-time dashboards. Packt Publishing.
- Hahn, S. (2021). Web Development with Node and Express: Leveraging the JavaScript Stack. O'Reilly Media.
- Mead, A. (2019). The Modern JavaScript Bootcamp. Pearson Education.
- Stoiber, M. (2021). Advanced React: Patterns for Building Reusable Components. Apress.
- Noring, L. (2020). RxJS in Action. Manning Publications.
- Freeman, A. (2019). Pro Angular 9: Build Powerful and Dynamic Web Apps. Apress.
- Dillard, N. (2020). Vue.js: Up and Running: Building Accessible and Performant Web Apps. O'Reilly Media.
- Vlissides, J. M., Gamma, E., Helm, R., & Johnson, R. E. (2018). Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional.
- Reddy, R. (2019). Microservices with Spring Boot and Spring Cloud. Packt Publishing.
- Cantelon, M., Harter, M., Holowaychuk, T., & Rajlich, N. (2021). Node.js in Action (3rd ed.). Manning Publications.
- Di Benedetto, P., Onodi, M., & Randazzo, D. (2020). Pro Spring Security: Securing Spring Framework 5 and Boot 2-based Java Applications. Apress.
- De Rosa, M. (2021). Spring Boot: Up and Running: Building Cloud Native Java and Kotlin Applications. O'Reilly Media.
- Simons, M. (2020). Spring Boot in Practice. Manning Publications.
- Raj, P., & Seshadri, S. (2019). Angular Projects: Build Nine Real-world Applications from Scratch Using Angular 8 and TypeScript. Packt Publishing.
- Porcello, E., & Banks, A. (2020). Learning GraphQL and Relay. Packt Publishing.

	<ul style="list-style-type: none"> <li>○ Fain, Y., &amp; Moiseev, A. (2019). Angular Development with TypeScript (2nd ed.). Manning Publications.</li> <li>○ Zanon, V. (2021). Microfrontends in Action. Manning Publications.</li> <li>○ Davis, S., &amp; Boje, D. (2019). Pro WebSockets: A Practical Guide to Real-Time Communication with WebSockets. Apress.</li> <li>○ Bhambhani, M. (2020). Building RESTful Web Services with Spring 5: Leverage the power of Spring 5.0, Java SE 9, and Spring Boot 2.0. Packt Publishing.</li> </ul>
<p><b>Обсяг курсу</b></p>	<p>Загальний обсяг: 150 годин (аудиторних занять: 64 год., з них 32 год. лекцій та 32 год. лабораторних робіт; самостійної роботи: 86 год).</p>
<p><b>Очікувані результати навчання</b></p>	<p>Після завершення цього курсу студент буде:</p> <p><b>Знати:</b></p> <ol style="list-style-type: none"> <li>1. <b>Основи веб-застосунків:</b> <ul style="list-style-type: none"> <li>○ Архітектуру веб-застосунків та розрізнення між frontend та backend частинами.</li> <li>○ Основи протоколу HTTP та його ролі в веб-застосунках.</li> </ul> </li> <li>2. <b>Spring і Spring WebFlux:</b> <ul style="list-style-type: none"> <li>○ Основи Spring як контейнера зворотнього зв'язку та його основних компонентів.</li> <li>○ Принципи реактивного програмування та як їх застосовувати в Spring WebFlux.</li> </ul> </li> <li>3. <b>GraphQL:</b> <ul style="list-style-type: none"> <li>○ Що таке GraphQL та в чому його переваги порівняно з традиційними RESTful сервісами.</li> <li>○ Як створювати схеми GraphQL та використовувати їх для обробки запитів.</li> </ul> </li> <li>4. <b>Websockets:</b> <ul style="list-style-type: none"> <li>○ Основи протоколу Websockets та його відмінності від традиційного HTTP.</li> <li>○ Як імплементувати взаємодію в реальному часі в веб-застосунках.</li> </ul> </li> <li>5. <b>Angular:</b> <ul style="list-style-type: none"> <li>○ Основи фреймворку Angular, його архітектуру та основні компоненти.</li> <li>○ Як розробляти SPA (Single Page Application) за допомогою Angular.</li> </ul> </li> </ol> <p><b>Вміти:</b></p> <ol style="list-style-type: none"> <li>1. <b>Розробляти веб-застосунки:</b> <ul style="list-style-type: none"> <li>○ Проектувати та розробляти повноцінні веб-застосунки, об'єднуючи frontend та backend частини.</li> </ul> </li> <li>2. <b>Використовувати Spring і Spring WebFlux:</b> <ul style="list-style-type: none"> <li>○ Створювати RESTful сервіси на Spring.</li> <li>○ Розробляти реактивні веб-сервіси за допомогою Spring WebFlux.</li> </ul> </li> </ol>

	<p>3. <b>Розробляти GraphQL сервіси:</b></p> <ul style="list-style-type: none"> <li>○ Створювати GraphQL сервери на Spring.</li> <li>○ Опрацьовувати та відповідати на GraphQL запити.</li> </ul> <p>4. <b>Впроваджувати Websockets:</b></p> <ul style="list-style-type: none"> <li>○ Імплементувати функціональність в реальному часі в веб-застосунках за допомогою Websockets.</li> </ul> <p>5. <b>Розробляти застосунки на Angular:</b></p> <ul style="list-style-type: none"> <li>○ Створювати, тестувати та розгортати веб-додатки на Angular.</li> <li>○ Інтегрувати Angular застосунки з backend сервісами.</li> </ul>			
<b>Ключові слова</b>	Web Development, RESTful Services, Spring Framework, Spring WebFlux, Reactive Programming, GraphQL, Spring GraphQL, Websockets, Real-time Communication, Angular, SPA, Frontend Development, Backend Development, HTTP Protocol, Java			
<b>Формат курсу</b>	Очний. Проведення лекцій та лабораторнихзанять			
<b>Теми</b>	Подано нижче у таблиці Схема курсу “ Проектування web застосунків”			
<b>Підсумковий контроль, форма</b>	Залік			
<b>Пререквізити</b>	<ol style="list-style-type: none"> <li>1. <b>Основи програмування:</b> Знання основних концепцій програмування, таких як змінні, цикли, умовні оператори, функції тощо.</li> <li>2. <b>Основи веб-технологій:</b> Розуміння, як працює Інтернет, основних принципів HTTP та різниці між клієнтом та сервером.</li> <li>3. <b>Основи баз даних:</b> Розуміння основ реляційних баз даних, можливість створення простих SQL-запитів.</li> <li>4. <b>Основи ООП:</b> Розуміння принципів об'єктно-орієнтованого програмування, таких як інкапсуляція, наслідування, поліморфізм та абстракція.</li> <li>5. <b>HTML та CSS:</b> Знання основ створення веб-сторінок за допомогою HTML, стилізація за допомогою CSS.</li> <li>6. <b>Основи JavaScript:</b> Знання основ JavaScript для взаємодії з DOM та реалізації клієнтської логіки.</li> <li>7. <b>Основи розробки серверних застосунків:</b> Поняття про сервер, запуск серверних застосунків, робота з серверними фреймворками на базовому рівні.</li> </ol>			
<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 (ок-</b>	<p><b>Оцінювання</b> проводиться за 100-бальною шкалою.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%;"></td> <td style="width: 33%;"></td> <td style="width: 33%; text-align: center;"><b>Оцінка за національною шкалою</b></td> </tr> </table>			<b>Оцінка за національною шкалою</b>
		<b>Оцінка за національною шкалою</b>		

ремо для кожного виду навчальної діяльності)	Оцінка за шкалою ECTS		Оцінка в балах	Екзамен, диференційований залік		залік
	A	Відмінно	100 - 90	Відмінно	5	
	B	Дуже добре	81 - 89	Добре	4	
	C	Добре	71 - 80			
	D	Задовільно	61 - 70	Задовільно	3	
	E	Достатньо	51 - 60			
	FХ (F)	Незадовільно	0 - 50	Незадовільно	2	не зараховано

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.

**Функціонал згідно з вимогами (максимум 15 балів)**

**1-бал.** Студент практично нічого не зробив, але прийшов на здачу і демонструє, що він намагався виконати поставлене завдання. Однак, наявна реалізація не зодовольє вимоги завдання.

**1-5 балів.** Мінімальна реалізація вимог. Було зроблено менше 30% з запланованого функціоналу або функціонал працює з помилками, що критично впливають на роботу застосунку.

**6-9 балів.** Часткова реалізація вимог. Виконано від 30% до 70% запланованого функціоналу. Присутні дрібні помилки, які не впливають на загальну роботу застосунку.

**10-13 балів.** Майже повна реалізація вимог. Виконано від 70% до 90% запланованого функціоналу, з мінімальними помилками або недоліками.

**14-15 балів.** Повна реалізація вимог. Всі заплановані функції реалізовані правильно та без помилок.

**Якість коду (максимум 5 балів)**

**1 бал.** Код містить значні помилки, погано організований, важко читабельний, не дотримано конвенцій кодування.

**2-3 бали.** Код добре організований, але містить деякі недоліки у читабельності або недотримання конвенцій кодування. Присутня спроба організації коду, але потребує оптимізації.

**4 бали.** Код добре організований і читабельний, з мінімальними недоліками. Дотримано конвенцій кодування, код ефективно структурований.

**5 балів.** Відмінна якість коду. Код чистий, добре організований, оптимізований, без помилок. Повне дотримання конвенцій кодування і стандартів розробки.

**Академічна доброчесність:** Очікується, що роботи студентів будуть їх оригінальними дослідженнями чи міркуваннями. Відсутність посилань на

	<p>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>
<b>Опитування</b>	Оцінювання якості курсу буде доступне у системі “Dekanat” після завершення курсу.

## Схема курсу “Проектування web застосунків”

### Лекційні заняття

Тема	Тези лекції	Література та інтернет ресурси
<b>Вступ до web-застосунків та Spring:</b>	Огляд веб-застосунків, вступ до Spring Framework та його екосистеми	[1]
<b>Основи RESTful веб-сервісів</b>	Що таке REST, основні принципи, створення простого RESTful API на Spring.	[6, 7]
<b>Поглиблене вивчення REST на Spring</b>	Обробка помилок, валідація, HATEOAS, Spring Data REST	[1]
<b>Вступ до реактивного програмування та Spring WebFlux</b>	Що таке реактивне програмування, основи Reactor, реактивний Spring WebFlux.	[2]
<b>Поглиблене вивчення Spring WebFlux</b>	Реактивні контролери, реактивні репозиторії, обробка помилок в реактивних API	[2]
<b>Вступ до GraphQL і Spring GraphQL</b>	Що таке GraphQL, як відрізняється від REST, перші кроки з Spring GraphQL	[2, 3]
<b>Поглиблене вивчення Spring GraphQL</b>	Створення складних запитів, мутації, директиви, обробка помилок	[2, 3]
<b>Вебсокети та їх реалізація на Spring</b>	Основи вебсокетів, створення вебсокет-сервера на Spring, підписки	[2, 4]
<b>Вступ до Angular та його екосистеми:</b>	Як працює Angular, архітектура, основні концепції.	[5]
<b>Компоненти та модулі в Angular</b>	Створення компонентів, взаємодія між компонентами, основи модулів	[5]
<b>Сервіси та Dependency Injection в Angular:</b>	Як створити сервіс, DI в Angular, використання провайдерів	[5]

<b>Маршрутизація в Angular</b>	Налаштування маршрутів, lazy-loading, обробка помилок під час маршрутизації	[5]
<b>Форми в Angular</b>	Template-driven та Reactive форми, валідація, динамічні форми	[5]
<b>З'єднання Angular з RESTful та GraphQL сервісами</b>	Використання HttpClient для RESTful сервісів, інтеграція з GraphQL	[5]
<b>Реактивний підхід в Angular з RxJS</b>	Основи RxJS, як використовувати Observables, оператори.	[5, 6]
<b>Заключна лекція: кращі практики та підсумки курсу</b>	Розгляд найкращих практик для проектування веб-застосунків, підсумки та додаткові ресурси для вивчення.	[6, 7]

## Лабораторні заняття

<b>Тема</b>	<b>Короткий опис лабораторного заняття</b>	<b>Література та інтернет ресурси</b>
<b>Встановлення та налаштування середовища</b>	Встановлення JDK, Spring Boot, Angular CLI. Створення базового Spring Boot проекту.	[1]
<b>Перше RESTful API на Spring</b>	Створення простого RESTful API, тестування за допомогою Postman	[1]
<b>Розширення RESTful API</b>	Додавання валідації, обробки помилок і реалізації HATEOAS.	[1, 6, 7]
<b>Вступ до реактивної розробки з Spring WebFlux</b>	Створення реактивного контролера, тестування реактивних ендпоінтів	[1, 2]
<b>Розширення реактивного API з Spring WebFlux</b>	Додавання реактивних репозиторіїв, обробка помилок у реактивному стилі.	[1, 2]
<b>Початкове знайомство з GraphQL на Spring</b>	Створення базового GraphQL сервера, виконання запитів за допомогою GraphQL Playground.	[1, 2, 3]
<b>Розширення GraphQL сервера</b>	Додавання складних запитів, мутацій, директив	[1, 2, 3]
<b>Реалізація вебсокетів на Spring:</b>	Створення простого вебсокет-сервера, тестування з WebSocket клієнтом	[4]
<b>Основи Angular: Створення простого застосунку</b>	Використання Angular CLI, створення основних компонентів, модулів	[5]
<b>Сервіси і DI в Angular</b>	Створення сервісу, впровадження залежностей, взаємодія між компонентами	[5]
<b>Маршрутизація в Angular</b>	Налаштування базової маршрутизації, lazy-loading, canActivate guards.	[5]
<b>Робота з формами в Angular</b>	Створення template-driven та reactive форм, валідація	[5]
<b>З'єднання Angular з backend:</b>	Використання HttpClient для з'єднання з RESTful та GraphQL API	[5]

<b>Фінальний проект</b>	Студенти створюють повноцінний веб-застосунок, що об'єднує всі навички, отримані під час курсу	
<b>Розгортання проекту в хмарі</b>	Студенти розгортають попередньо розроблений веб застосунок в AWS	
<b>Залікове заняття</b>	Завершення приймання індивідуальних завдань. Виставлення оцінок. Консультація, щодо талону (на вимогу).	