

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
“Проектування web застосунків”
що викладається в межах ОПШ Прикладна математика
першого (бакалаврського) рівня вищої освіти для здобувачів з
спеціальності 113 Прикладна математика**

Львів 2023р.

Назва дисципліни	Проектування web застосунків
Адреса викладання дисципліни	Головний корпус ЛНУ ім. І. Франка м. Львів, вул. Університетська 1
Факультет та кафедра, за якою закріплена дисципліна	Факультет прикладної математики та інформатики Кафедра прикладної математики
Галузь знань, шифр та назва спеціальності	11 – математика та статистика 113 – прикладна математика
Викладачі дисципліни	Заневич Олег Богданович, асистент кафедри прикладної математики
Контактна інформація викладачів	Oleh.Zanevych@lnu.edu.ua; https://ami.lnu.edu.ua/employee/oleh-zanevych
Консультації з питань навчання по дисципліні відбуваються	Консультації в день проведення лекцій/лабораторних занять (за попередньою домовленістю).
Сторінка дисципліни	
інформація про дисципліну	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 5 кредитів (за Європейською Кредитно-Трансферною Системою ECTS).
Коротка анотація дисципліни	<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>
Мета та цілі дисципліни	<p>Мета курсу. 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> <p>Цілі курсу:</p> <ul style="list-style-type: none"> • Освоєння основних концепцій розробки веб-застосунків на платформі Spring. • Набуття практичних навичок у створенні RESTful та GraphQL

	<p>веб-сервісів, зокрема використовуючи реактивний підхід за допомогою Spring WebFlux.</p> <ul style="list-style-type: none"> • Закріплення знань про двосторонню комунікацію в реальному часі через розробку вебсокет-з'єднань. • Ознайомлення із основами фреймворку Angular та його застосуванням у розробці клієнтської частини веб-застосунків. • Розуміння та використання найкращих практик розробки, тестування та оптимізації веб-застосунків. • Розвиток навичок командної роботи та самостійного рішення проблем під час проєктування та розробки веб-додатків.
<p>Література для вивчення дисципліни</p>	<ol style="list-style-type: none"> Spring <ul style="list-style-type: none"> ○ Walls, C. (2019). <i>Spring in Action</i>. Manning Publications. ○ Long, J., & Winch, R. (2020). <i>Spring Security in Action</i>. Manning Publications. Spring WebFlux <ul style="list-style-type: none"> ○ Pivovarit, M. (2020). <i>Reactive Programming with Spring 5</i>. Packt Publishing. ○ Long, J. (2019). <i>Reactive Spring</i>. Leanpub. GraphQL <ul style="list-style-type: none"> ○ Porcello, E., & Banks, A. (2018). <i>Learning GraphQL: Declarative Data Fetching for Modern Web Apps</i>. O'Reilly Media. ○ Bojan, T. (2020). <i>Full-Stack GraphQL Applications with GRANDstack - Essential Excerpts</i>. Leanpub. Websockets <ul style="list-style-type: none"> ○ Pimentel, V., & Lourenço, J. R. (2019). <i>WebSocket: Lightweight Client-Server Communications</i>. O'Reilly Media. Angular <ul style="list-style-type: none"> ○ Freeman, A. (2018). <i>Pro Angular 6</i>. Apress. ○ Staples, M. (2019). <i>Angular for Enterprise-Ready Web Applications</i>. Packt Publishing. ○ Kimmel, R. (2019). <i>Angular Development with TypeScript</i>. Manning Publications. General Web Application Design and Architecture <ul style="list-style-type: none"> ○ Richardson, C., & Smith, R. (2019). <i>Microservices Patterns</i>. Manning Publications. ○ Newman, S. (2018). <i>Building Microservices: Designing Fine-Grained Systems</i>. O'Reilly Media. ○ Fowler, M. (2018). <i>Patterns of Enterprise Application Architecture</i>. Addison-Wesley Professional. Додаткові ресурси та література <ul style="list-style-type: none"> ○ Spring Documentation (Official): https://docs.spring.io/spring/docs/current/spring-framework-reference/ ○ Angular Documentation (Official): https://angular.io/docs ○ Söderberg, B. (2019). <i>Mastering Spring Boot 2.0: Build modern, cloud-native, and distributed systems using Spring Boot</i>. Packt Publishing. ○ Darwin, I. F. (2020). <i>Java Cookbook: Problems and Solutions for Java Developers</i>. O'Reilly Media. ○ Long, J., & Freire, J. L. (2019). <i>Reactive Spring: Build Cloud Native Applications with Reactive Programming</i>.

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.
- Fain, Y., & Moiseev, A. (2019). Angular Development with TypeScript (2nd ed.). Manning Publications.
- Zanon, V. (2021). Microfrontends in Action. Manning Publications.
- Davis, S., & Boje, D. (2019). Pro WebSockets: A Practical Guide to Real-Time Communication with WebSockets. Apress.

	<ul style="list-style-type: none"> ○ 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.
Обсяг курсу	<p>Загальний обсяг: 150 годин (аудиторних занять: 64 год., з них 32 год. лекцій та 32 год. лабораторних робіт; самостійної роботи: 86 год).</p>
Очікувані результати навчання	<p>Після завершення цього курсу студент буде:</p> <p>Знати:</p> <ol style="list-style-type: none"> 1. Основи веб-застосунків: <ul style="list-style-type: none"> ○ Архітектуру веб-застосунків та розрізнення між frontend та backend частинами. ○ Основи протоколу HTTP та його ролі в веб-застосунках. 2. Spring і Spring WebFlux: <ul style="list-style-type: none"> ○ Основи Spring як контейнера зворотнього зв'язку та його основних компонентів. ○ Принципи реактивного програмування та як їх застосовувати в Spring WebFlux. 3. GraphQL: <ul style="list-style-type: none"> ○ Що таке GraphQL та в чому його переваги порівняно з традиційними RESTful сервісами. ○ Як створювати схеми GraphQL та використовувати їх для обробки запитів. 4. Websockets: <ul style="list-style-type: none"> ○ Основи протоколу Websockets та його відмінності від традиційного HTTP. ○ Як імплементувати взаємодію в реальному часі в веб-застосунках. 5. Angular: <ul style="list-style-type: none"> ○ Основи фреймворку Angular, його архітектуру та основні компоненти. ○ Як розробляти SPA (Single Page Application) за допомогою Angular. <p>Вміти:</p> <ol style="list-style-type: none"> 1. Розробляти веб-застосунки: <ul style="list-style-type: none"> ○ Проєктувати та розробляти повноцінні веб-застосунки, об'єднуючи frontend та backend частини. 2. Використовувати Spring і Spring WebFlux: <ul style="list-style-type: none"> ○ Створювати RESTful сервіси на Spring. ○ Розробляти реактивні веб-сервіси за допомогою Spring WebFlux. 3. Розробляти GraphQL сервіси: <ul style="list-style-type: none"> ○ Створювати GraphQL сервери на Spring. ○ Опрацьовувати та відповідати на GraphQL запити. 4. Впроваджувати Websockets: <ul style="list-style-type: none"> ○ Імплементувати функціональність в реальному часі в веб-застосунках за допомогою Websockets. 5. Розробляти застосунки на Angular:

	<ul style="list-style-type: none"> ○ Створювати, тестувати та розгортати веб-додатки на Angular. ○ Інтегрувати Angular застосунки з backend сервісами. 																												
Ключові слова	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																												
Формат курсу	Очний. Проведення лекцій та лабораторних занять																												
Теми	Подано нижче у таблиці Схеми курсу “Проектування web застосунків”																												
Підсумковий контроль, форма	Залік																												
Пререквізити	<ol style="list-style-type: none"> 1. Основи програмування: Знання основних концепцій програмування, таких як змінні, цикли, умовні оператори, функції тощо. 2. Основи веб-технологій: Розуміння, як працює Інтернет, основних принципів HTTP та різниці між клієнтом та сервером. 3. Основи баз даних: Розуміння основ реляційних баз даних, можливість створення простих SQL-запитів. 4. Основи ООП: Розуміння принципів об'єктно-орієнтованого програмування, таких як інкапсуляція, наслідування, поліморфізм та абстракція. 5. HTML та CSS: Знання основ створення веб-сторінок за допомогою HTML, стилізація за допомогою CSS. 6. Основи JavaScript: Знання основ JavaScript для взаємодії з DOM та реалізації клієнтської логіки. 7. Основи розробки серверних застосунків: Поняття про сервер, запуск серверних застосунків, робота з серверними фреймворками на базовому рівні. 																												
Навчальні методи та техніки, які будуть використовуватись під час викладання курсу	Презентації, лекції, семінари. Домашні та індивідуальні завдання																												
Необхідне обладнання	Комп'ютер із доступом до інтернету																												
Критерії оцінювання (ок-ремо для кожного виду навчальної діяльності)	<p>Оцінювання 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="2">Оцінка за національною шкалою</th> </tr> <tr> <th>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> </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> </tbody> </table> <p style="text-align: right;">зараховано</p>	Оцінка за шкалою ECTS		Оцінка в балах	Оцінка за національною шкалою		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
Оцінка за шкалою ECTS					Оцінка в балах	Оцінка за національною шкалою																							
		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																											

FX (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 задачу і демонструє, що він намагався виконати поставлене завдання. Однак, наявна реалізація не зодоволяє 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 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.

Відвідання занять є важливою складовою навчання. Очікується, що всі студенти відвідають усі лекції та лабораторні заняття курсу, якщо інше не регламентується правилами узгодженими з деканатом

	<p>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>Література. 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 “Проектування web застосунків”

Лекційні заняття

№ п/п	Тема	Тези лекції	Література та інтернет ресурси	Завдання, год
1.	Вступ до web-застосунків та Spring:	Огляд веб-застосунків, вступ до Spring Framework та його екосистеми	[1]	2
2.	Основи RESTful веб-сервісів	Що таке REST, основні принципи, створення простого RESTful API на Spring.	[6, 7]	2
3.	Поглиблене вивчення REST на Spring	Обробка помилок, валідація, HATEOAS, Spring Data REST	[1]	2
4.	Вступ до реактивного програмування та Spring WebFlux	Що таке реактивне програмування, основи Reactor, реактивний Spring WebFlux.	[2]	2
5.	Поглиблене вивчення Spring WebFlux	Реактивні контролери, реактивні репозиторії, обробка помилок в реактивних API	[2]	2
6.	Вступ до GraphQL і Spring GraphQL	Що таке GraphQL, як відрізняється від REST, перші кроки з Spring GraphQL	[2, 3]	2
7.	Поглиблене вивчення Spring GraphQL	Створення складних запитів, мутації, директиви, обробка помилок	[2, 3]	2
8.	Вебсокети та їх реалізація на Spring	Основи вебсокетів, створення вебсокет-сервера на Spring, підписки	[2, 4]	2
9.	Вступ до Angular та його екосистеми:	Як працює Angular, архітектура, основні концепції.	[5]	2
10.	Компоненти та модулі в Angular	Створення компонентів, взаємодія між компонентами, основи модулів	[5]	2
11.	Сервіси та Dependency Injection в Angular:	Як створити сервіс, DI в Angular, використання провайдерів	[5]	2
12.	Маршрутизація в Angular	Налаштування маршрутів, lazy-loading, обробка помилок під час маршрутизації	[5]	2
13.	Форми в Angular	Template-driven та Reactive форми, валідація, динамічні форми	[5]	2
14.	З'єднання Angular з RESTful та GraphQL сервісами	Використання HttpClient для RESTful сервісів, інтеграція з GraphQL	[5]	2
15.	Реактивний підхід в Angular з RxJS	Основи RxJS, як використовувати Observables, оператори.	[5, 6]	2

16.	Заключна лекція: кращі практики та підсумки курсу	Розгляд найкращих практик для проектування веб-застосунків, підсумки та додаткові ресурси для вивчення.	[6, 7]	2
-----	--	---	--------	---

Лабораторні заняття

№ п/п	Тема	Короткий опис лабораторного заняття	Література та інтернет ресурси	Завдання, год
1.	Встановлення та налаштування середовища	Встановлення JDK, Spring Boot, Angular CLI. Створення базового Spring Boot проекту.	[1]	3
2.	Перше RESTful API на Spring	Створення простого RESTful API, тестування за допомогою Postman	[1]	3
3.	Розширення RESTful API	Додавання валідації, обробки помилок і реалізації HATEOAS.	[1, 6, 7]	3
4.	Вступ до реактивної розробки з Spring WebFlux	Створення реактивного контролера, тестування реактивних ендпоінтів	[1, 2]	3
5.	Розширення реактивного API з Spring WebFlux	Додавання реактивних репозиторіїв, обробка помилок у реактивному стилі.	[1, 2]	3
6.	Початкове знайомство з GraphQL на Spring	Створення базового GraphQL сервера, виконання запитів за допомогою GraphQL Playground.	[1, 2, 3]	3
7.	Розширення GraphQL сервера	Додавання складних запитів, мутацій, директив	[1, 2, 3]	5
8.	Реалізація вебсокетів на Spring:	Створення простого вебсокет-сервера, тестування з WebSocket клієнтом	[4]	3
9.	Основи Angular: Створення простого застосунку	Використання Angular CLI, створення основних компонентів, модулів	[5]	3
10.	Сервіси і DI в Angular	Створення сервісу, впровадження залежностей, взаємодія між компонентами	[5]	5
11.	Маршрутизація в Angular	Налаштування базової маршрутизації, lazy-loading, canActivate guards.	[5]	4
12.	Робота з формами в Angular	Створення template-driven та reactive форм, валідація	[5]	3
13.	З'єднання Angular з backend:	Використання HttpClient для з'єднання з RESTful та GraphQL API	[5]	3
14.	Фінальний проект	Студенти створюють повноцінний веб-застосунок, що об'єднує всі навички, отримані під час курсу		5
15.	Розгортання проекту в хмарі	Студенти розгортають попередньо розроблений веб застосунок в AWS		5
16.	Приймання індивідуальних завдань	Завершення приймання індивідуальних завдань. Виставлення оцінок. Консультація, щодо талону (на вимогу).		