

МІНІСТЕРСТВО ОСВІТИ ТА 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 2025 р.)



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

Силабус з навчальної дисципліни
“Архітектура та дизайн Web-сторінок”
що викладається в межах ОПП “Середня освіта (Інформатика)”
першого (бакалаврського) рівня вищої освіти
для здобувачів зі спеціальності
A4.09 Середня освіта (Інформатика)

Назва дисципліни	Архітектура та дизайн Web-сторінок
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 програмування
Галузь знань, шифр та назва спеціальності	Галузь знань: А Освіта/ Педагогіка Спеціальність: А4.09 Середня освіта(Інформатика)
Викладачі дисципліни	Малець Романна Богданівна, к. ф.-м. н., доцент, доцент кафедри програмування
Контактна інформація викладачів	romanna.malets@lnu.edu.ua ; https://ami.lnu.edu.ua/employee/malets-r-b ; Головний корпус ЛНУ ім. І. Франка, каб. 263. м. Львів, вул. Університетська, 1
Консультації з питань навчання по дисципліні відбуваються	Консультації в день проведення лекцій/практичних занять(за попередньою домовленістю).
Сторінка дисципліни	https://ami.lnu.edu.ua/academics/bachelor/curriculum-education
інформація про дисципліну	Дисципліна “Архітектура та дизайн Web-сторінок” є дисципліною на вибір зі спеціальності А4 Середня освіта (Інформатика) для освітньої програми “ Середня освіта (Інформатика), яка викладається в 6-му семестрі в обсязі 4,5 кредитів (за Європейською Кредитно-Трансферною Системою ECTS).
Коротка анотація дисципліни	Курс "Архітектура та дизайн Web-сторінок" 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 міжклієнтом та сервером в реальному часі.
Мета та цілі дисципліни	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. Цілі курсу: <ul style="list-style-type: none"> • Освоєння основних концепцій розробки веб-застосунків на платформі Spring. • Набуття практичних навичок у створенні RESTful та GraphQL веб-сервісів, зокрема використовуючи реактивний підхід за допомогою Spring WebFlux. • Закріплення знань про двосторонню комунікацію в реальному часі через розробку вебсокет-з'єднань.

	<ul style="list-style-type: none"> • 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 проектування та розробки веб-додатків.
<p style="text-align: center;">Література для вивчення дисципліни</p>	<ol style="list-style-type: none"> 1. 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. 2. Spring WebFlux <ul style="list-style-type: none"> ○ Pivovarov, M. (2020). <i>Reactive Programming with Spring 5</i>. Packt Publishing. ○ Long, J. (2019). <i>Reactive Spring</i>. Leanpub. 3. 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. 4. Websockets <ul style="list-style-type: none"> ○ Pimentel, V., & Lourenço, J. R. (2019). <i>WebSocket: Lightweight Client-Server Communications</i>. O'Reilly Media. 5. 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. 6. 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. 7. Additional Resources <ul style="list-style-type: none"> ○ Spring Documentation (Official): https://docs.spring.io/spring/docs/current/spring-framework-reference/

	<p>o Angular Documentation (Official): https://angular.io/docs</p>
Обсяг курсу	<p>Загальний обсяг: 135 годин. Аудиторних занять: 64 год., з них 32 год. лекційних та 32 годин лабораторних робіт. Самостійної роботи: 71 год.</p>
Очікувані результати навчання	<p>Після завершення цього курсу студент буде:</p> <p>Знати:</p> <ol style="list-style-type: none"> Основи веб-застосунків: <ul style="list-style-type: none"> Архітектуру веб-застосунків та розрізнення міжfrontend та backend частинами. Основи протоколу HTTP та його ролю в веб-застосунках. 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 їхзастосовувати в Spring WebFlux. 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 використовуватиїх для обробки запитів. Websockets: <ul style="list-style-type: none"> Основи протоколу Websockets та його відмінностейвід традиційного HTTP. Як імплементувати взаємодію в реальному часі ввеб-застосунках. Angular: <ul style="list-style-type: none"> Основи фреймворку Angular, його архітектуру таосновні компоненти. Як розробляти SPA (Single Page Application) за допомогою Angular. <p>Вміти:</p> <ol style="list-style-type: none"> Розробляти веб-застосунки: <ul style="list-style-type: none"> Проектувати та розробляти повноцінні веб-застосунки, об'єднуючи frontend та backend частини. Використовувати Spring і Spring WebFlux: <ul style="list-style-type: none"> Створювати RESTful сервіси на Spring. Розробляти реактивні веб-сервіси за допомогоюSpring WebFlux. Розробляти GraphQL сервіси: <ul style="list-style-type: none"> Створювати GraphQL сервери на Spring. Опрацьовувати та відповідати на GraphQL запити. Впроваджувати Websockets: <ul style="list-style-type: none"> Імплементувати функціональність в реальному часі в веб-застосунках за допомогою Websockets. Розробляти застосунки на Angular:

	<ul style="list-style-type: none"> ○ Створювати, тестувати та розгорнути 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 Схема курсу “Архітектура та дизайн Web-сторінок”
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 (окремо для кожного виду навчальної діяльності)	<p>Оцінювання проводиться за 100-бальною шкалою. Студенти набирають 100 балів протягом семестру за виконання індивідуальних завдань.</p> <p>Студент виконає 14 індивідуальних лабораторних робіт. Кожна робота оцінюється максимально у 7 балів.</p> <ul style="list-style-type: none"> • 100% – умови завдання виконано повністю, автор відповідає на всі запитання щодо використаних підходів, чітко інтерпретує отримані результати, немає ознак недоброчесності; • 80% – наведено логічно правильну послідовність розв'язування, алгоритми складено правильно, бракує

	<p>окремих коментарів, автор не досить повно пояснює використані підходи, немає ознак недоброчесності;</p> <ul style="list-style-type: none">• 60% – у правильній послідовності розв’язування допущено окремі помилки, які автор уміє виправити після зауваження викладача, бракує коментарів, на запитання щодо використаних підходів автор відповідає з помилками, немає ознак недоброчесності;• 40% – у правильній послідовності розв’язування пропущено окремі етапи, завдання виконано частково, автор не розуміє недоліків поданої роботи, не вміє їх виправити, немає ознак недоброчесності;• 20% – завдання виконано частково, автор не може самостійно інтерпретувати отримані результати, виправити помилки, немає ознак недоброчесності;• 0% – завдання не виконано, або ж виявлено ознаки недоброчесності: запозичення, автор не володіє відповідним теоретичним матеріалом тощо; <p>Додаткові бали студент може одержати за активну участь під час лекційного заняття.</p> <p>Академічна доброчесність: Очікується, що роботи студентів будуть їх оригінальними дослідженнями чи міркуваннями.</p> <p>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>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 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 “ Архітектура та дизайн Web-сторінок ”

Лекційні заняття

1. Вступ до веб-застосунків та Spring:

- Огляд веб-застосунків, вступ до Spring Framework та його екосистеми.

2. Основи RESTful веб-сервісів:

- Що таке REST, основні принципи, створення простого RESTful API на Spring.

3. Поглиблене вивчення REST на Spring:

- Обробка помилок, валідація, HATEOAS, Spring Data REST.

4. Вступ до реактивного програмування та Spring WebFlux:

- Що таке реактивне програмування, основи Reactor, реактивний Spring WebFlux.

5. Поглиблене вивчення Spring WebFlux:

- Реактивні контролери, реактивні репозиторії, обробка помилок в реактивних API.

6. Вступ до GraphQL і Spring GraphQL:

- Що таке GraphQL, як відрізняється від REST, перші кроки з Spring GraphQL.

7. Поглиблене вивчення Spring GraphQL:

- Створення складних запитів, мутації, директиви, обробка помилок.

8. Вебсокети та їх реалізація на Spring:

- Основи вебсокетів, створення вебсокет-сервера на Spring, підписки.

9. Вступ до Angular та його екосистеми:

- Як працює Angular, архітектура, основні концепції.

10. Компоненти та модулі в Angular:

- Створення компонентів, взаємодія між компонентами, основи модулів.

11. Сервіси та Dependency Injection в Angular:

- Як створити сервіс, DI в Angular, використання провайдерів.

12. Маршрутизація в Angular:

- Налаштування маршрутів, lazy-loading, обробка помилок під час маршрутизації.

13. Форми в Angular:

- Template-driven та Reactive форми, валідація, динамічні форми.

14. З'єднання Angular з RESTful та GraphQL сервісами:

- Використання HttpClient для RESTful сервісів, інтеграція з GraphQL.

15. Реактивний підхід в Angular з RxJS:

- Основи RxJS, як використовувати Observables, оператори.

16. Заключна лекція: кращі практики та підсумки курсу:

- Розгляд найкращих практик для проектування веб-застосунків, підсумки та додаткові ресурси для вивчення.

Практичні заняття

1. Встановлення та налаштування середовища:

- Встановлення JDK, Spring Boot, Angular CLI. Створення базового Spring Boot проекту.

2. Перше RESTful API на Spring:

- Створення простого RESTful API, тестування за допомогою Postman.

3. Розширення RESTful API:

- Додавання валідації, обробки помилок і реалізації HATEOAS.

4. Вступ до реактивної розробки з Spring WebFlux:

- Створення реактивного контролера, тестування реактивних ендпоінтів.

5. Розширення реактивного API з Spring WebFlux:

- Додавання реактивних репозиторіїв, обробка помилок у реактивному стилі.

6. Початкове знайомство з GraphQL на Spring:

- Створення базового GraphQL сервера, виконання запитів за допомогою GraphQL Playground.

7. Розширення GraphQL сервера:

- Додавання складних запитів, мутацій, директив.

8. Реалізація вебсокетів на Spring:

- Створення простого вебсокет-сервера, тестування з WebSocket клієнтом.

9. Основи Angular: Створення простого застосунку:

- Використання Angular CLI, створення основних компонентів, модулів.

10. Сервіси і DI в Angular:

- Створення сервісу, впровадження залежностей, взаємодія між компонентами.

11. Маршрутизація в Angular:

- Налаштування базової маршрутизації, lazy-loading, canActivate guards.

12. Робота з формами в Angular:

- Створення template-driven та reactive форм, валідація.

13. З'єднання Angular з backend:

- Використання HttpClient для з'єднання з RESTful та GraphQL API.

14. Фінальний проект:

- Студенти створюють повноцінний веб-застосунок, що об'єднує всі навички, отримані під час курсу.