

## СИЛАБУС

Базова інформація про дисципліну	
Назва дисципліни	SE052 Front-end програмування з використанням сучасного фреймворка / Front-End Programming with a Modern Framework
Рівень вищої освіти / фахової передвищої освіти	Фаховий молодший бакалавр
Галузь знань	12 Інформаційні технології
Спеціальність	121 Інженерія програмного забезпечення
Освітня програма	Інженерія програмного забезпечення
Семестр	5 семестр (9 кл)
Курс	3 курс (на базі базової середньої освіти)
Анотація курсу	<p>Навчальна дисципліна відноситься до вибіркових дисциплін курсу. Вона орієнтована на студентів, які мають базові знання HTML, CSS та JavaScript і прагнуть поглибити свої навички у створенні сучасних інтерактивних веб-застосунків.</p> <p>У процесі вивчення дисципліни студенти ознайомлюються з концепціями front-end розробки, принципами побудови односторінкових застосунків (SPA) та сучасними підходами до організації клієнтської частини веб-систем. Особлива увага приділяється роботі з популярними фреймворками (React, Angular або Vue), їх архітектурі, компонентному підходу та управлінню станом додатків.</p> <p>На лабораторних заняттях студенти опановують практичні навички створення інтерфейсів користувача, реалізації динамічної взаємодії з сервером через API, роботи з маршрутизацією та оптимізації продуктивності веб-застосунків. Також розглядаються сучасні інструменти розробки, системи збірки проєктів та засоби тестування.</p> <p>Дисципліна передбачає формування практичних компетентностей у розробці масштабованих і зручних веб-інтерфейсів відповідно до сучасних стандартів та вимог користувацького досвіду. Контроль знань здійснюється у формі лабораторних робіт, експрес-контролю та модульних контрольних робіт.</p>
Сторінка курсу в MOODLE	<a href="http://78.137.2.119:2929/course/view.php?id=818">http://78.137.2.119:2929/course/view.php?id=818</a>
Мова викладання	Українська
Лектор курсу	<p>Дмитрюк Валентин Віталійович</p> <p>Канали комунікації:</p> <p>СНД «Moodle»: повідомлення в чаті</p> <p>E-mail: valaimg2218@gmail.com</p>

<b>Місце дисципліни в освітній програмі</b>	
<b>Освітня програма</b>	<a href="http://esbc.edu.ua/documents/otdel/moop_pr24.pdf">http://esbc.edu.ua/documents/otdel/moop_pr24.pdf</a>
<b>Інтегральна компетентність (ІК)</b>	Здатність вирішувати типові спеціалізовані задачі інженерії програмного забезпечення, що вимагає застосування положень і методів відповідних наук (математики, інформатики, інформаційних технологій, тощо) та може характеризуватися певною невизначеністю умов; нести відповідальність за результати своєї діяльності; здійснювати контроль інших осіб у визначених ситуаціях.
<b>Перелік загальних компетентностей (ЗК)</b>	<b>ЗК:</b> <ul style="list-style-type: none"> <li>– Знання та розуміння предметної області та розуміння професійної діяльності.</li> <li>– Здатність до пошуку, оброблення та аналізу інформації з різних джерел.</li> <li>– Здатність застосовувати знання у практичних ситуаціях.</li> </ul>
<b>Перелік спеціальних компетентностей (СК)</b>	<b>СК:</b> <ul style="list-style-type: none"> <li>– Здатність алгоритмічно та логічно мислити.</li> <li>– Здатність вдосконалювати знання і навички в галузі інформаційних технологій та усвідомлення важливості навчання протягом усього життя.</li> <li>– Здатність розробляти модулі і компоненти програмного забезпечення за допомогою типових алгоритмів та інструментів.</li> <li>– Здатність вибирати та використовувати ефективні інструментальні засоби розробки програмного продукту.</li> </ul>
<b>Перелік програмних результатів навчання</b>	<b>РН:</b> <ul style="list-style-type: none"> <li>– Систематизувати та узагальнювати інформацію про підходи, методи та засоби розробки супроводу програмного забезпечення.</li> <li>– Розробляти та супроводжувати програмне забезпечення.</li> <li>– Обирати інструментальні засоби, ефективні методи та здійснювати тестування програмних систем.</li> <li>– Аналізувати та узагальнювати необхідну інформацію з різних джерел та ресурсів для розв'язання професійних задач з урахуванням сучасних досягнень інформаційних технологій.</li> </ul>
<b>Опис дисципліни</b>	
<b>Структура навантаження на студента</b>	Загальна кількість годин – 90 Кількість кредитів – 3 Кількість лекційних годин – 15 Кількість практичних занять – 30 Кількість годин для самостійної роботи студентів – 45 Форма підсумкового контролю – залік
<b>Методи навчання</b>	За подачею навчального матеріалу: методи готових знань, дослідницький метод. З огляду на мету навчання: методи здобуття нових знань, метод формування умінь і навичок, метод застосування знань на практиці, методи закріплення знань, умінь і навичок, методи перевірки і оцінювання знань, умінь і навичок
<b>Зміст дисципліни</b>	
Тема 1. Вступ до фронтенду та Angular	Розглядаються основи front-end розробки та роль клієнтської частини у веб-застосунках. Студенти ознайомлюються з

	фреймворком Angular, його можливостями, структурою та перевагами у створенні сучасних односторінкових застосунків.
Тема 2. Основи TypeScript	Вивчаються ключові особливості мови TypeScript як надбудови над JavaScript. Розглядаються типізація, інтерфейси, класи та інші інструменти, що дозволяють підвищити надійність і підтримуваність коду.
Тема 3. Архітектура Angular-додатку	Описуються принципи побудови Angular-застосунків, їх модульна структура та організація проєкту. Розглядаються основні елементи архітектури, зокрема модулі, компоненти та сервіси.
Тема 4. Компоненти, шаблони, data binding	Вивчається створення компонентів як основних будівельних блоків Angular-додатків. Розглядаються шаблони та механізми зв'язування даних (data binding), що забезпечують взаємодію між логікою та інтерфейсом.
Тема 5. Сервіси та Dependency Injection	Розглядається використання сервісів для організації бізнес-логіки та повторного використання коду. Вивчається механізм Dependency Injection, який забезпечує гнучке управління залежностями в застосунку.
Тема 6. RxJS: Observables, оператори	Ознайомлення з бібліотекою RxJS для роботи з асинхронними потоками даних. Розглядаються Observables, основні оператори та їх застосування для обробки подій і HTTP-запитів.
Тема 7. Інтеграція з API та використання Gemini CLI	Вивчаються методи взаємодії Angular-додатку із серверною частиною через API. Розглядається використання Gemini CLI для автоматизації розробки, генерації коду та інтеграції можливостей штучного інтелекту у процес створення веб-застосунків.
<b>Політика дисципліни</b>	
<b>Політика відвідування</b>	Регулярне відвідування всіх видів занять, своєчасність виконання самостійної роботи. За об'єктивних причин (наприклад, хвороба, міжнародне стажування) навчання зорганізується в он-лайн формі за погодженням із керівником курсу.
<b>Політика щодо дедлайнів та перескладання</b>	Роботи, які здаються із порушенням термінів без поважних причин, оцінюються на нижчу оцінку.
<b>Академічна доброчесність</b>	У випадку недотримання політики академічної доброчесності (плагіат, самоплагіат, фабрикація, фальсифікація, списування, обман, хабарництво) передбачено повторне проходження оцінювання.
<b>Використання ШІ</b>	Використання ШІ під час виконання завдань регламентується Політикою «Використання ШІ в освітньому процесі ЧДБК» Завдання мають маркування регламенту використання ШІ.

### Система оцінювання

Система оцінювання підсумкової успішності студентів поділяється на **поточний контроль** та **семестровий контроль**.

**Поточний контроль** здійснюється протягом семестру і охоплює всі види аудиторної роботи (практичні заняття) та виконання індивідуальних завдань. Максимальна кількість балів, яку студент може набрати за цей вид контролю, становить 100.

#### Підсумковий контроль

Відбувається у формі диференційного заліку (у формі підсумкового модульного контролю).

#### Розрахунок підсумкової оцінки

Підсумкова оцінка базується виключно на балах, накопичених протягом семестру (S). Ваговий коефіцієнт у цьому випадку становить 1.

Формула:  $O=S \times 1$

Види навчальної роботи	Загальна кількість балів
Усні відповіді за темами 1 по 1 балу	5
Практична робота за темами 1 по 5 балів	55
Модульні контрольні роботи (2 к.р. по 10 балів)	20
Індивідуальна самостійна робота (проєкт)	20
Разом	100

### Критерії оцінювання для кожного виду навчальної

#### Критерії оцінювання усних відповідей:

1 б. – Студент відповів правильно на запитання.

0,5 б. – Студент частково правильно відповів на запитання.

0 б. – Студент не зміг відповісти на запитання.

#### Критерії практичних робіт

5 б. – Студент виконав всі завдання практичної роботи без помилок, а також правильно оформив звіт.

4 б. – Студент виконав всі завдання практичної роботи, але допустився помилки в одному з них та в оформленні звіту.

3 б. – Студент виконав всі завдання практичної роботи, але допустився помилки в двох з них та є недоліки в оформленні звіту.

2 б. – Студент виконав лише частину завдань практичної роботи, але має помилки при виконанні, а також є недоліки в оформленні звіту.

1 б. – Студент намагався виконати практичну роботу, але в завданнях є помилки та звіт оформлено невірно.

0 б. – студент не виконав практичної роботи та не здав звіт.

#### Критерії оцінювання модульних робіт

10 б. – виконано всі завдання без помилок, відповіді повні й обґрунтовані.

9 б. – виконано майже всі завдання, допущено декілька незначних помилок.

8 б. – виконано більшу кількість завдань, але є окремі помилки та недоречності у відповідях.

7 б. – виконано три завдання, але з помітними помилками.

6 б. – виконано два завдання повністю та половину третього, але частина з них має помилки.

5 б. – виконано два завдання, але продемонстровано розуміння основного матеріалу.

4 б. – виконано деякі завдання правильно, але більшість з помилками.

3 б. – робота має лише деякі правильні елементи у відповідях.

2 б. – виконано мінімальний обсяг завдань, знання без глибокого розуміння.

1 б. – студент намагався виконати завдання, але відповіді містять помилки й потребують корекції.

#### Критерії оцінювання індивідуальних робіт (проєктів)

20 б. – завдання виконано повністю, без жодної помилки; звіт правильно й акуратно оформлений, відповіді повні, логічні та аргументовані.

19 б. – усі завдання виконані, але є 1–2 несуттєві неточності у відповідях чи оформленні.

18–15 б. – виконано більшість завдань правильно, звіт загалом оформлений належно; наявні окремі помилки у змісті або дрібні недоліки в структурі/оформленні.

4–13 б. – завдання виконані частково, відповіді містять суттєві неточності чи неповноту; у звіті є помилки в оформленні або бракує аргументації.

12–10 б. – виконано половину чи трохи більше завдань, відповіді часто неправильні або поверхові; звіт має помітні недоліки у змісті та структурі.

9–7 б. – зроблено спробу виконати більшість завдань, проте більшість відповідей неправильні або неповні; звіт оформлено формально, із значними помилками.

6–4 б. – завдання виконані частково, правильних відповідей небагато; звіт майже не відповідає вимогам оформлення.

3–1 б. – зроблено лише символічну спробу виконати завдання; відповіді в основному неправильні; звіт оформлено вкрай слабо.

0 б. – завдання не виконано, звіт відсутній.

ECTS	Бали	Зміст
<b>A</b>	90-100	Бездоганна підготовка в широкому контексті
<b>B</b>	80-89	Повні знання, міцні вміння
<b>C</b>	70-79	Хороші знання та вміння
<b>D</b>	65-69	Задовільні знання, стереотипні вміння
<b>E</b>	60-64	Виконання мінімальних вимог діяльності в стандартних умовах
<b>FX</b>	35-59	Слабкі знання, відсутність умінь
<b>F</b>	1-34	Необхідний повторний курс

#### Список рекомендованих джерел

1. Freeman A. Pro Angular. 16th ed. New York: Apress, 2024. 850 p.
2. Green D., Seshadri S. Angular: Up and Running. 2nd ed. Sebastopol: O'Reilly Media, 2022. 350 p.
3. Cherny M. Programming TypeScript. Sebastopol: O'Reilly Media, 2023. 324 p.
4. Banks A., Porcello E. Learning React. 2nd ed. Sebastopol: O'Reilly Media, 2023. 350 p.
5. Choi B. RxJS in Action. Shelter Island: Manning Publications, 2021. 300 p.
6. Fowler M. Patterns of Enterprise Application Architecture. Boston: Addison-Wesley, 2021. 560 p.
7. Gamma E., Helm R., Johnson R., Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software. Boston: Addison-Wesley, 2021. 395 p.

#### Інтернет ресурси

1. Angular Documentation. URL: <https://angular.io>.
2. TypeScript Documentation. URL: <https://www.typescriptlang.org>.
3. RxJS Documentation. URL: <https://rxjs.dev>.
4. MDN Web Docs. URL: <https://developer.mozilla.org>.
5. Angular University. URL: <https://angular-university.io>.
6. Stack Overflow. URL: <https://stackoverflow.com>.
7. Gemini CLI Documentation. URL: <https://ai.google.dev>.