

Базова інформація про дисципліну	
Назва дисципліни	SE017 / Front-end програмування на React JS / Front-end-programming on React JS
Рівень вищої освіти / фахової передвищої освіти	Фахова передвища
Семестр	I семестр
Факультет /відділення	Інженерії програмного забезпечення
Анотація курсу	<p>Даний курс розрахований на студентів з базовим досвідом програмування на JavaScript і бажанням навчатися новим технологіям для розробки сучасних додатків.</p> <p>При проходженні даної дисципліни, студенти ознайомляться з поняттям «front-end розробка», сучасними інструментами розробки веб-застосунків, підходами створення багатофункціональних систем з веб-інтерфейсом. На лабораторних заняттях опанують бібліотеки та інструментальні засоби для розробки клієнтської частини веб-застосунків, зокрема Angular, React, Vue. Студенти також ознайомляться з перспективними технологіями, що підтримуються сучасними браузерами та описані останніми версіями стандартів. В курсі передбачений контроль якості отриманих знань у вигляді експрес-контрольних та модульних контрольних робіт.</p>
Сторінка курсу в MOODLE	http://78.137.2.119:1919/m72/course/view.php?id=1182
Мова викладання	Українська

Лектор курсу	Сухенко Андрій Сергійович канали комунікації: СДН «Moodle»: повідомлення в чаті E-mail: andrii.suhenko@gmail.com Telegram: https://t.me/andrii_sukhenko
Місце дисципліни в освітній програмі	
Перелік загальних компетентностей (ЗК)	<p>ЗК05. Знання та розуміння предметної області та розуміння професійної діяльності.</p> <p>ЗК06. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.</p> <p>ЗК07. Здатність застосовувати знання у практичних ситуаціях.</p>
Перелік спеціальних компетентностей (СК)	<p>СК01. Здатність алгоритмічно та логічно мислити.</p> <p>СК02. Здатність вдосконалювати знання і навички в галузі інформаційних технологій та усвідомлення важливості навчання протягом усього життя.</p> <p>СК07. Здатність розробляти модулі і компоненти програмного забезпечення за допомогою типових алгоритмів та інструментів.</p> <p>СК09. Здатність вибирати та використовувати ефективні інструментальні засоби розробки програмного продукту.</p>
Перелік програмних результатів навчання	<p>РН02. Систематизувати та узагальнювати інформацію про підходи, методи та засоби розробки супроводу програмного забезпечення.</p> <p>РН05. Розробляти та супроводжувати програмне забезпечення.</p> <p>РН11. Обирати інструментальні засоби, ефективні методи та здійснювати тестування програмних систем.</p> <p>РН15. Аналізувати та узагальнювати необхідну інформацію з різних джерел та ресурсів для розв'язання професійних задач з урахуванням сучасних досягнень інформаційних технологій.</p>
Опис дисципліни	
Структура навантаження на студента	<p>Загальна кількість годин – 180</p> <p>Кількість кредитів – 6</p> <p>Кількість лекційних годин – 30</p> <p>Кількість практичних занять – 30</p>

	<p>Кількість годин для самостійної роботи студентів – 120</p> <p>Форма підсумкового контролю – залік</p>
--	--

Методи навчання	<p>За подачею навчального матеріалу: методи готових знань, дослідницький метод.</p> <p>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</p>
Зміст дисципліни	
Тема 1. Основи бібліотеки React	<p>Перешкоди та труднощі. React як бібліотека. Новий синтаксис ECMAScript. Популярність функціонального JavaScript. Майбутнє бібліотеки React. Робота з файлами. Файловий репозиторій. Інструменти React-розробника. Встановлення Node.js.</p>
Тема 2. Новий синтаксис JavaScript	<p>Оголошення змінних в ES6 (const, let, шаблонні рядки, Параметри за замовчуванням). Стрілкові функції. Транспіляція ES6. Об'єкти та масиви ES6. Проміси. Класи. Модулі ES6. CommonJS</p>
Тема 3. Функціональне програмування і застосуванням JavaScript	<p>Значення поняття функціональності. Порівняння імперативності з декларативністю. Функціональні концепції. Незмінність. Чисті функції. Перетворення даних. Функції вищого порядку. Рекурсія. Композиція.</p>
Тема 4. Чистий React	<p>Налаштування сторінки. Віртуальний DOM. Елементи React. ReactDOM. Дочірні елементи. Конструювання елементів з даними. Компоненти React. React.createClass. React.Component. Функціональні компоненти, що не мають стану. Відображення DOM.</p>
Тема 5. React з JSX	<p>Елементи React у вигляді коду JSX. Поради щодо застосування JSX. Вкладені компоненти. Відображення масивів на JSX. Введення в Webpack. Завантажувачі Webpack.</p>

<p>Тема 6. Властивості, стан та дерево компонентів</p>	<p>Перевірка властивостей. Перевірка властивостей під час використання createClass. Властивості, які використовуються за замовчуванням. Перевірка властивостей, що настраюється. Класи ES6 та функціональні компоненти, не мають стану. Зворотний потік даних. Посилання у функціональних компонентах, які мають стан. Управління станом React. Впровадження стану компонента. Ініціалізація стану із властивостей. Стан в середині дерева компонента. Передача властивостей вниз по дереву компонентів. Передача даних по дереву компонентів</p>
<p>Тема 7. Удосконалення компонентів</p>	<p>Життєві цикли компонентів. Життєвий цикл установки. Життєвий цикл оновлення, React.Children. Підключення бібліотек JavaScript. Створення запитів за допомогою Fetch. Підключення D3 Timeline. Компоненти вищого порядку. Управління станом за межами React. Реалізації Flux.</p>
<p>Тема 8. Redux</p>	<p>Стан. Дії. Перетворювачі. Перетворювач кольору. Перетворювач кольорів. Перетворювач сортування. Збереження в localStorage. Функції проміжної ланки.</p>
<p>Тема 9. Використання технології React Redux</p>	<p>React Redux явна передача сховища. Передача сховища через контекст. Порівняння презентаційних та контейнерних компонентів. Провайдер React Redux. Функція connect бібліотеки React Redux .</p>
<p>Тема 10. React та сервер</p>	<p>Порівняння ізоморфізму з універсалізмом. Код React, який відображається на сервері. Універсальний організатор кольорів. Універсальний Redux. Універсальна маршрутизація. Обмін даними з сервером. Виконання дій на сервері. Дії з Redux Thunks.</p>
<p>Політика дисципліни</p>	

Політика відвідування	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.
--	---

Політика щодо дедлайнів та перескладання	Роботи, які здаються із порушенням термінів без поважних причин, оцінюються на нижчу оцінку.
---	--

Академічна доброчесність	У випадку недотримання політики академічної доброчесності (плагіат, самоплагіат, фабрикація, фальсифікація, списування, обман, хабарництво) передбачено повторне проходження оцінювання.
---------------------------------	--

Система оцінювання

Поточний контроль здійснюється протягом семестру під час проведення практичних, семінарських та інших видів занять і оцінюється сумою набраних балів (максимальна сума – 100 балів; мінімальна сума, що дозволяє студенту отримати атестацію з предмету – 60 балів); підсумковий/ семестровий контроль, проводиться у формі заліку, відповідно до графіку навчального процесу.

Підсумкова оцінка за умови заліку виставляється як загальна сума балів, набраних за результатами поточного контролю.

Накопичування рейтингових балів з навчальної дисципліни

Види навчальної роботи	Мах кількість балів
Аудиторна	
Практичні завдання (10 тем)	60
Тестування	10
Індивідуальна	
Захист практичних завдань до самостійної роботи	30
РАЗОМ	100

Шкала оцінювання

ECTS	Бали	Зміст
-------------	-------------	--------------

A	90-100	Бездоганна підготовка в широкому контексті
B	80-89	Повні знання, міцні вміння
C	70-79	Хороші знання та вміння
D	65-69	Задовільні знання, стереотипні вміння
E	60-64	Виконання мінімальних вимог діяльності в стандартних умовах

FX	35-59	Слабкі знання, відсутність умінь
F	1-34	Необхідний повторний курс

Список рекомендованих джерел

1. Marijn Haverbeke. Eloquent JavaScript, 3rd Edition: A Modern Introduction to Programming — No Starch Press, 2018. — 472 p. Also available online <https://eloquentjavascript.net/>
2. David Flanagan. JavaScript: The Definitive Guide: Master the World's Most-Used Programming Language. — 7th Edition — O'Reilly Media, 2020. — 706 p.
3. Boris Cherny. Programming TypeScript: Making Your JavaScript Applications Scale. — O'Reilly Media, 2019. — 324 p.
4. Aristeidis Bampakos, Pablo Deeleman. Learning Angular: A no-nonsense beginner's guide to building web applications with Angular 10 and TypeScript. — 3rd Edition — Packt Publishing, 2020. — 430 p.
5. Alex Banks, Eve Porcello. Learning React: Modern Patterns for Developing React Apps. — 2nd Edition O'Reilly Media, 2020. — 310 p.
6. Heitor Ramon Ribeiro. Vue.js 3 Cookbook: Discover actionable solutions for building modern web apps with the latest Vue features and TypeScript. — Packt Publishing , 2020. — 562 p.