



Технології програмування користувацьких інтерфейсів (Front-end) Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	Перший (бакалаврський)
Галузь знань	12 Інформаційні технології
Спеціальність	123 Комп'ютерна інженерія
Освітня програма	Комп'ютерні системи та мережі
Статус дисципліни	Вибіркова
Форма навчання	очна(денна)/ заочна
Рік підготовки, семестр	3 курс, осінній семестр
Обсяг дисципліни	4 кредити 120 годин
Семестровий контроль/ контрольні заходи	залік
Розклад занять	//rozklad.kpi.ua
Мова викладання	Українська
Інформація про керівника курсу / викладачів	Лектор: старший викладач, Алещенко Олексій Вадимович alexey.aleshchenko@gmail.com Лабораторні: старший викладач, Алещенко Олексій Вадимович alexey.aleshchenko@gmail.com
Розміщення курсу	//comsys.kpi.ua

Програма навчальної дисципліни

1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

Дисципліна "Технології програмування користувацьких інтерфейсів (Front-end)" спрямована на опанування студентами методів та засобів розробки користувацьких інтерфейсів. Знання основ програмування користувацьких інтерфейсів необхідне для створення програмного забезпечення для комп'ютерних систем, систем реального часу, інтернет-додатків, мобільних пристроїв.

Метою викладання дисципліни "Технології програмування користувацьких інтерфейсів (Front-end)" є:

ЗДАТНІСТЬ: до абстрактного мислення, аналізу та синтезу; працювати в команді; використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення; створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж; забезпечувати захист інформації, що обробляється в комп'ютерних та кіберфізичних системах та мережах з метою реалізації встановленої політики інформаційної безпеки; системно адмініструвати, використовувати, адаптувати та експлуатувати наявні інформаційні технології та системи; ідентифікувати, класифікувати та

описувати роботу програмно-технічних засобів, комп'ютерних та кіберфізичних систем, мереж та їхніх компонентів шляхом використання аналітичних методів і методів моделювання; проектувати, впроваджувати, адмініструвати та обслуговувати глобальні, локальні інтелектуальні програмно - конфігуровані комп'ютерні мережі; розробляти, адаптувати, використати програмне забезпечення для покращення ефективності застосування високопродуктивних комп'ютерних систем; організації обчислювальних процесів в високопродуктивних комп'ютерних системах з різною структурною організацією на основі використання новітніх технологій планування і диспетчеризації та сучасних операційних систем;

ЗНАННЯ: основ управління проектами; для ідентифікації, формулювання і розв'язування технічних задач спеціальності, використовуючи методи, що є найбільш придатними для досягнення поставлених цілей; технічних характеристик, конструктивних особливостей, призначення і правил експлуатації програмно-технічних засобів комп'ютерних систем та мереж для вирішення технічних задач спеціальності;

УМІННЯ: проведення експериментів, збирання даних та моделювання в комп'ютерних системах; застосовувати знання для ідентифікації, формулювання і розв'язування технічних задач спеціальності, використовуючи методи, що є найбільш придатними для досягнення поставлених цілей; розв'язувати задачі аналізу та синтезу засобів, характерних для спеціальності; застосовувати знання технічних характеристик, конструктивних особливостей, призначення і правил експлуатації програмно-технічних засобів комп'ютерних систем та мереж для вирішення технічних задач спеціальності; розробляти програмне забезпечення для вбудованих і розподілених застосувань, мобільних і гібридних систем, розраховувати, експлуатувати, типове для спеціальності обладнання; виконувати експериментальні дослідження за професійною тематикою; виконувати розрахунки параметрів окремих комп'ютерних систем, комп'ютерних мереж; вміти створювати та обслуговувати бази даних.

2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

Необхідні дисципліни: "Програмування. Частина 1. Програмування", "Структури даних та алгоритми".

3. Зміст навчальної дисципліни

Тема 1. HTML.

Тема 2. CSS.

Тема 3. LESS.

Тема 4. SASS.

Тема 5. SCSS.

Тема 6. Bootstrap.

Тема 7. JavaScript.

Тема 8. TypeScript.

Тема 9. jQuery.

Тема 10. React.

Тема 11. Virtual DOM.

Тема 12. Redux.

Тема 13. Angular.

Тема 14. Vue.

4. Навчальні матеріали та ресурси

Базова:

1. Караєва, Н. В. *WEB - та мобільні технології в економіці [Електронний ресурс] : конспект лекцій / Н. В. Караєва ; НТУУ «КПІ ім. Ігоря Сікорського»*. – Електронні текстові дані (1 файл: 2,02 МБ). – Київ : НТУУ «КПІ ім. Ігоря Сікорського», 2017. – 158 с. – Назва з екрана. <https://ela.kpi.ua/handle/123456789/18947>
2. *Технології програмування користувацьких інтерфейсів (Front-end). Методичні вказівки до виконання лабораторних робіт для здобувачів ступеня бакалавра за спеціальностями: 121 Інженерія програмного забезпечення, 123 Комп'ютерна інженерія, 126 Інформаційні системи та технології всіх форм навчання / Уклад.: Алещенко О.В. – К.: КПІ ім. Ігоря Сікорського, 2022. [Електронний ресурс]. Ухвалено кафедрою обчислювальної техніки (протокол № 10 від 25.05.2022 р.). Погоджено Методичною комісією ФІОТ (протокол № 10 від 09.06.2022 р.). <https://comsys.kpi.ua/metodichni-vkazannya-po-disciplinam>.*

Додаткова:

3. Marijn Haverbeke. *Eloquent JavaScript, 3rd Edition: A Modern Introduction to Programming* — No Starch Press, 2018. — 472 p. Also available online <https://eloquentjavascript.net/>
4. David Flanagan. *JavaScript: The Definitive Guide: Master the World's Most-Used Programming Language*. — 7th Edition — O'Reilly Media, 2020. — 706 p.
5. Boris Cherny. *Programming TypeScript: Making Your JavaScript Applications Scale*. — O'Reilly Media, 2019. — 324 p.
6. 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.
7. Alex Banks, Eve Porcello. *Learning React: Modern Patterns for Developing React Apps*. — 2nd Edition O'Reilly Media, 2020. — 310 p.
8. 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.

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

Очна форма навчання

Назви розділів і тем	Кількість годин				
	Всього	у тому числі			
		Лекції	Практичні (семінарські)	Лабораторні роботи	СРС
Тема 1. HTML.	11	4	-	3	4
Тема 2. CSS.	11	4	-	3	4
Тема 3. SASS.	5	1	-	-	4
Тема 4. SCSS.	5	1	-	-	4
Тема 5. LESS.	5	1	-	-	4
Тема 6. Bootstrap.	5	1	-	-	4
Тема 7. JavaScript.	11	4	-	3	4
Тема 8. TypeScript.	5	1	-	-	4
Тема 9. jQuery.	5	1	-	-	4
Тема 10. React.	11	4	-	3	4

Тема 11. Virtual DOM.	6	2	-	-	4
Тема 12. Redux.	6	2	-	-	4
Тема 13. Angular.	14	6	-	3	5
Тема 14. Vue.	12	4	-	3	5
МКР	4	-	-	-	4
Залік	4	-	-	-	4
Всього в семестрі:	120	36	-	18	66

Заочна форма навчання

Назви розділів і тем	Кількість годин				
	Всього	у тому числі			
		Лекції	Практичні (семінарські)	Лабораторні роботи	СРС
Тема 1. HTML.	8	1	-	1	6
Тема 2. CSS.	8	1	-	1	6
Тема 3. SASS.	6,25	0,25	-	-	6
Тема 4. SCSS.	6,25	0,25	-	-	6
Тема 5. LESS.	6,25	0,25	-	-	6
Тема 6. Bootstrap.	6,25	0,25	-	-	6
Тема 7. JavaScript.	8	1	-	1	6
Тема 8. TypeScript.	6,25	0,25	-	-	6
Тема 9. jQuery.	7,25	0,25	-	-	7
Тема 10. React.	10	1	-	2	7
Тема 11. Virtual DOM.	7,25	0,25	-	-	7
Тема 12. Redux.	7,25	0,25	-	-	7
Тема 13. Angular.	10	1	-	2	7
Тема 14. Vue.	9	1	-	1	7
МКР	7	-	-	-	7
Залік	7	-	-	-	7
Всього в семестрі:	120	8	-	8	104

Лекційні заняття

Очна форма навчання

№ лекції	Назва теми лекції, перелік основних питань та завдання на СРС
1	«HTML». Загальна характеристика мови HTML. Основи синтаксису HTML. СРС: Частково або повністю виконати першу лабораторну роботу даного кредитного модуля, використовуючи мову HTML.
2	«HTML. Продовження». Елементи та атрибути HTML5. Глобальні атрибути. Атрибути користувача. Структура документа HTML5 та його валідація. Мета-теги. СРС: Створити документ HTML5, використовуючи (окрім іншого) глобальні атрибути та мета-теги. Провалідувати створений документ.
3	«CSS». Каскадні таблиці стилів. Приєднання стилів до сторінки. Селектори. CSS властивості.

	CPC: Частково або повністю виконати другу лабораторну роботу даного кредитного модуля, використовуючи технологію CSS.
4	«CSS. Продовження». Поточна робота над специфікаціями CSS і як взяти в ній участь. Псевдо-класи та псевдо-елементи. Модальне вікно. CPC: Спробувати в другій лабораторній роботі даного кредитного модуля реалізувати модальність та використати псевдо-класи та псевдо-елементи.
5	«SASS. SCSS». Скриптова метамова. Підвищення рівня абстракції коду та спрощення файлів CSS. Синтаксис SASS та SCSS. Вкладені правила. Домішки (міксини). Цикли. Успадкування. CPC: Спробувати переробити другу лабораторну роботу даного кредитного модуля з використанням технологій SASS або SCSS.
6	«LESS. Bootstrap». Динамічна мова стилів. Вкладена метамова. Змінні, вкладені правила, міксини, оператори та функції. Bootstrap шаблони CSS та HTML для типографіки, форм, кнопок, навігації та інших компонентів інтерфейсу. CPC: Спробувати переробити другу лабораторну роботу даного кредитного модуля з використанням технологій LESS та Bootstrap.
7	«JavaScript». Основи JavaScript. Реалізація стандарту ECMAScript. Змінні та константи. Типи даних. Операції. Обробка помилок. Робота з браузером та BOM. Робота з DOM. CPC: Частково або повністю виконати третю лабораторну роботу даного кредитного модуля на мові програмування JavaScript.
8	«JavaScript. Продовження». JSON. Promise, async і await. Ajax. Fetch API. Функціональне програмування. Об'єктно-орієнтоване програмування. Додаткові сфери застосування JS. CPC: Спробувати створити програму на мові програмування JavaScript з використанням promise, async і await, Ajax та Fetch API.
9	«TypeScript. jQuery». Основи TypeScript. Змінні та константи. Типи даних. Функції. Тип функції та стрілочні функції. Селектори та фільтри jQuery. Маніпуляція елементами jQuery. Ефекти та анімація в jQuery. CPC: Спробувати переробити третю лабораторну роботу в рамках даного кредитного модуля з використанням мови TypeScript та бібліотеки jQuery.
10	«React». Основи React. Рендеринг елементів. Компоненти. Props. Події. Стан. Управління компонентами-класами. Життєвий цикл компоненту. CPC: Частково або повністю виконати четверту лабораторну роботу даного кредитного модуля, використовуючи React.
11	«React. Продовження». Хуки. Управління функціональними компонентами. Маршрутизація. CPC: Спробувати переробити четверту лабораторну роботу в рамках даного кредитного модуля з використанням хуків та функціональних компонент.
12	«Virtual DOM». Бібліотека ReactDOM. Узгодження. Тінювий DOM. React Fiber. CPC: Спробувати розібратися із алгоритмом узгодження віртуального та реального DOM та усвідомити, як саме в цьому приймаються участь ключі елементів списку.
13	«Redux». Сховище. Дії. Творці дій. Reducer. CPC: Спробувати переробити четверту лабораторну роботу в рамках даного кредитного модуля з використанням Redux.
14	«Angular». Основи Angular. Компоненти. Модулі. Стили та шаблони компонента. Прив'язка даних. Взаємодія між компонентами.

	CPC: Частково або повністю виконати п'яту лабораторну роботу даного кредитного модуля, використовуючи <i>Angular</i> .
15	«Angular. Продовження». Прив'язка до подій дочірнього компонента. Життєвий цикл компоненту. Взаємодія між модулями. Директиви. Сервіси та <i>dependency injection</i> . CPC: Спробувати додати до реалізації п'ятої лабораторної роботи в рамках даного кредитного модуля використання методів життєвого циклу компоненту, взаємодії між модулями, директив, сервісів та <i>dependency injection</i> .
16	«Angular. Продовження». Робота з формами. HTTP та взаємодія з сервером. Маршрутизація. <i>Pipes</i> . CPC: Спробувати додати до реалізації п'ятої лабораторної роботи в рамках даного кредитного модуля використання роботи з формами, HTTP та взаємодії з сервером, маршрутизація та <i>pipes</i> .
17	«Vue». Основи <i>Vue 3</i> . Об'єкт <i>Vue</i> . Прив'язка даних. Обробка введення користувача. Події. Двостороння прив'язка. <i>Refs</i> та управління HTML-елементами. CPC: Частково або повністю виконати шосту лабораторну роботу даного кредитного модуля, використовуючи <i>Vue</i> .
18	«Vue. Продовження». Життєвий цикл <i>Vue</i> . Умовний рендеринг та робота з масивами. Робота з формами. Компоненти. Слоти. Маршрутизація. CPC: Спробувати додати до реалізації шостої лабораторної роботи в рамках даного кредитного модуля роботу з формами, умовний рендеринг, роботу з масивами, компоненти, слоти та маршрутизацію.

Заочна форма навчання

№ лекції	Назва теми лекції, перелік основних питань та завдання на CPC
1	«HTML». Загальна характеристика мови HTML. Основи синтаксису HTML. Елементи та атрибути HTML5. Глобальні атрибути. Атрибути користувача. Структура документа HTML5 та його валідація. Мета-теги. CPC: Частково або повністю виконати першу лабораторну роботу даного кредитного модуля, використовуючи мову HTML. Створити документ HTML5, використовуючи (окрім іншого) глобальні атрибути та мета-теги. Провалідувати створений документ. «CSS». Каскадні таблиці стилів. Приєднання стилів до сторінки. Селектори. CSS властивості. Поточна робота над специфікаціями CSS і як взяти в ній участь. Псевдо-класи та псевдо-елементи. Модальне вікно. CPC: Частково або повністю виконати другу лабораторну роботу даного кредитного модуля, використовуючи технологію CSS. Спробувати в другій лабораторній роботі даного кредитного модуля реалізувати модальність та використати псевдо-класи та псевдо-елементи.
2	«SASS. SCSS». Скриптова метамова. Підвищення рівня абстракції коду та спрощення файлів CSS. Синтаксис SASS та SCSS. Вкладені правила. Домішки (міксини). Цикли. Успадкування. CPC: Спробувати переробити другу лабораторну даного кредитного модуля з використанням технологій SASS або SCSS. «LESS. Bootstrap». Динамічна мова стилів. Вкладена метамова. Змінні, вкладені правила, міксини, оператори та функції. <i>Bootstrap</i> шаблони CSS та HTML для типографіки, форм, кнопок, навігації та інших компонентів інтерфейсу.

	<p>CPC: Спробувати переробити другу лабораторну даного кредитного модуля з використанням технологій LESS та Bootstrap.</p> <p>«JavaScript». Основи JavaScript. Реалізація стандарту ECMAScript. Змінні та константи. Типи даних. Операції. Обробка помилок. Робота з браузером та BOM. Робота з DOM. JSON. Promise, async і await. Ajax. Fetch API. Функціональне програмування. Об'єктно-орієнтоване програмування. Додаткові сфери застосування JS.</p> <p>CPC: Частково або повністю виконати третю лабораторну роботу даного кредитного модуля на мові програмування JavaScript. Спробувати створити програму на мові програмування JavaScript з використанням promise, async і await, Ajax та Fetch API.</p>
3	<p>«TypeScript. jQuery». Основи TypeScript. Змінні та константи. Типи даних. Функції. Тип функції та стрілочні функції. Селектори та фільтри jQuery. Маніпуляція елементами jQuery. Ефекти та анімація в jQuery.</p> <p>CPC: Спробувати переробити третю лабораторну роботу в рамках даного кредитного модуля з використанням мови TypeScript та бібліотеки jQuery.</p> <p>«React». Основи React. Рендеринг елементів. Компоненти. Props. Події. Стан. Управління компонентами-класами. Життєвий цикл компоненту. Хуки. Управління функціональними компонентами. Маршрутизація.</p> <p>CPC: Частково або повністю виконати четверту лабораторну роботу даного кредитного модуля, використовуючи React. Спробувати переробити четверту лабораторну роботу в рамках даного кредитного модуля з використанням хуків та функціональних компонент.</p> <p>«Virtual DOM». Бібліотека ReactDOM. Узгодження. Тіньовий DOM. React Fiber.</p> <p>CPC: Спробувати розібратися із алгоритмом узгодження віртуального та реального DOM та усвідомити, як саме в цьому приймаються участь ключі елементів списку.</p> <p>«Redux». Сховище. Дії. Творці дій. Reducer.</p> <p>CPC: Спробувати переробити четверту лабораторну роботу в рамках даного кредитного модуля з використанням Redux.</p>
4	<p>«Angular». Основи Angular. Компоненти. Модулі. Стили та шаблони компонента. Прив'язка даних. Взаємодія між компонентами. Прив'язка до подій дочірнього компонента. Життєвий цикл компоненту. Взаємодія між модулями. Директиви. Сервіси та dependency injection. Робота з формами. HTTP та взаємодія з сервером. Маршрутизація. Pipes.</p> <p>CPC: Частково або повністю виконати п'яту лабораторну роботу даного кредитного модуля, використовуючи Angular. Спробувати додати до реалізації п'ятої лабораторної роботи в рамках даного кредитного модуля використання методів життєвого циклу компоненту, взаємодії між модулями, директив, сервісів та dependency injection. Спробувати додати до реалізації п'ятої лабораторної роботи в рамках даного кредитного модуля використання роботи з формами, HTTP та взаємодії з сервером, маршрутизація та pipes.</p> <p>«Vue». Основи Vue 3. Об'єкт Vue. Прив'язка даних. Обробка введення користувача. Події. Двостороння прив'язка. Refs та управління HTML-елементами. Життєвий цикл Vue. Умовний рендеринг та робота з масивами. Робота з формами. Компоненти. Слоти. Маршрутизація.</p> <p>CPC: Частково або повністю виконати шосту лабораторну роботу даного кредитного модуля, використовуючи Vue. Спробувати додати до реалізації шостої</p>

лабораторної роботи в рамках даного кредитного модулю роботу з формами, умовний рендеринг, роботу з масивами, компоненти, слоти та маршрутизацію.

Лабораторні заняття

Очна форма навчання

№ з/п	Назва лабораторної роботи	Кількість ауд. годин
1	HTML (Тема 1)	3
2	CSS (Тема 2)	3
3	JavaScript (Тема 7)	3
4	React (Тема 10)	3
5	Angular (Тема 13)	3
6	Vue (Тема 14)	3
	Разом:	18

Заочна форма навчання

№ з/п	Назва лабораторної роботи	Кількість ауд. годин
1	HTML (Тема 1)	1
2	CSS (Тема 2)	1
3	JavaScript (Тема 7)	1
4	React (Тема 10)	2
5	Angular (Тема 13)	2
6	Vue (Тема 14)	1
	Разом:	8

6. Самостійна робота студента

Очна форма навчання

№ з/п	Назва теми, що виноситься на самостійне опрацювання	Кількість годин СРС
1	«HTML». Частково або повністю виконати першу лабораторну роботу даного кредитного модуля, використовуючи мову HTML.	2
2	«HTML. Продовження». Створити документ HTML5, використовуючи (окрім іншого) глобальні атрибути та мета-теги. Провалідувати створений документ.	2
3	«CSS». Частково або повністю виконати другу лабораторну роботу даного кредитного модуля, використовуючи технологію CSS.	2
4	«CSS. Продовження». Спробувати в другій лабораторній роботі даного кредитного модуля реалізувати модальність та використати псевдо-класи та псевдо-елементи.	2
5	«SASS. SCSS». Спробувати переробити другу лабораторну даного кредитного модуля з використанням технологій SASS або SCSS.	8

6	«LESS. Bootstrap». Спробувати переробити другу лабораторну даного кредитного модуля з використанням технологій LESS та Bootstrap.	8
7	«JavaScript». Частково або повністю виконати третю лабораторну роботу даного кредитного модуля на мові програмування JavaScript.	2
8	«JavaScript. Продовження». Спробувати створити програму на мові програмування JavaScript з використанням promise, async і await, Ajax та Fetch API.	2
9	«TypeScript. jQuery». Спробувати переробити третю лабораторну роботу в рамках даного кредитного модуля з використанням мови TypeScript та бібліотеки jQuery.	8
10	«React». Частково або повністю виконати четверту лабораторну роботу даного кредитного модуля, використовуючи React.	2
11	«React. Продовження». Спробувати переробити четверту лабораторну роботу в рамках даного кредитного модуля з використанням хуків та функціональних компонент.	2
12	«Virtual DOM». Спробувати розібратися із алгоритмом узгодження віртуального та реального DOM та усвідомити, як саме в цьому приймаються участь ключі елементів списку.	4
13	«Redux». Спробувати переробити четверту лабораторну роботу в рамках даного кредитного модуля з використанням Redux.	4
14	«Angular». Частково або повністю виконати п'яту лабораторну роботу даного кредитного модуля, використовуючи Angular.	3
15	«Angular. Продовження». Спробувати додати до реалізації п'ятої лабораторної роботи в рамках даного кредитного модуля використання методів життєвого циклу компоненту, взаємодії між модулями, директив, сервісів та dependency injection.	1
16	«Angular. Продовження». Спробувати додати до реалізації п'ятої лабораторної роботи в рамках даного кредитного модуля використання роботи з формами, HTTP та взаємодії з сервером, маршрутизація та pipes.	1
17	«Vue». Частково або повністю виконати шосту лабораторну роботу даного кредитного модуля, використовуючи Vue.	3
18	«Vue. Продовження». Спробувати додати до реалізації шостої лабораторної роботи в рамках даного кредитного модуля роботу з формами, умовний рендеринг, роботу з масивами, компоненти, слоти та маршрутизацію.	2
19	«Підготовка до МКР»	4
20	«Підготовка до заліку»	4
	Разом:	66

Заочна форма навчання

№ з/п	Назва теми, що виносить на самостійне опрацювання	Кількість годин СРС
1	«HTML». Загальна характеристика мови HTML. Основи синтаксису HTML. Елементи та атрибути HTML5. Глобальні атрибути. Атрибути користувача. Структура документа HTML5 та його валідація. Мета-теги.	6

	CPC: Частково або повністю виконати першу лабораторну роботу даного кредитного модуля, використовуючи мову HTML. Створити документ HTML5, використовуючи (окрім іншого) глобальні атрибути та мета-теги. Провалідувати створений документ.	
2	«CSS». Каскадні таблиці стилів. Приєднання стилів до сторінки. Селектори. CSS властивості. Поточна робота над специфікаціями CSS і як взяти в ній участь. Псевдо-класи та псевдо-елементи. Модальне вікно. CPC: Частково або повністю виконати другу лабораторну роботу даного кредитного модуля, використовуючи технологію CSS. Спробувати в другій лабораторній роботі даного кредитного модуля реалізувати модальність та використати псевдо-класи та псевдо-елементи.	6
3	«SASS. SCSS». Скриптова метамова. Підвищення рівня абстракції коду та спрощення файлів CSS. Синтаксис SASS та SCSS. Вкладені правила. Домішки (міксини). Цикли. Успадкування. CPC: Спробувати переробити другу лабораторну даного кредитного модуля з використанням технологій SASS або SCSS.	12
4	«LESS. Bootstrap». Динамічна мова стилів. Вкладена метамова. Змінні, вкладені правила, міксини, оператори та функції. Bootstrap шаблони CSS та HTML для типографіки, форм, кнопок, навігації та інших компонентів інтерфейсу. CPC: Спробувати переробити другу лабораторну даного кредитного модуля з використанням технологій LESS та Bootstrap.	12
5	«JavaScript». Основи JavaScript. Реалізація стандарту ECMAScript. Змінні та константи. Типи даних. Операції. Обробка помилок. Робота з браузером та BOM. Робота з DOM. JSON. Promise, async і await. Ajax. Fetch API. Функціональне програмування. Об'єктно-орієнтоване програмування. Додаткові сфери застосування JS. CPC: Частково або повністю виконати третю лабораторну роботу даного кредитного модуля на мові програмування JavaScript. Спробувати створити програму на мові програмування JavaScript з використанням promise, async і await, Ajax та Fetch API.	6
6	«TypeScript. jQuery». Основи TypeScript. Змінні та константи. Типи даних. Функції. Тип функції та стрілочні функції. Селектори та фільтри jQuery. Маніпуляція елементами jQuery. Ефекти та анімація в jQuery. CPC: Спробувати переробити третю лабораторну роботу в рамках даного кредитного модуля з використанням мови TypeScript та бібліотеки jQuery.	13
7	«React». Основи React. Рендеринг елементів. Компоненти. Props. Події. Стан. Управління компонентами-класами. Життєвий цикл компоненту. Хуки. Управління функціональними компонентами. Маршрутизація. CPC: Частково або повністю виконати четверту лабораторну роботу даного кредитного модуля, використовуючи React. Спробувати переробити четверту лабораторну роботу в рамках даного кредитного модуля з використанням хуків та функціональних компонент.	7
8	«Virtual DOM». Бібліотека ReactDOM. Узгодження. Тінювий DOM. React Fiber. CPC: Спробувати розібратися із алгоритмом узгодження віртуального та реального DOM та усвідомити, як саме в цьому приймаються участь ключі елементів списку.	7

9	«Redux». Сховище. Дії. Творці дій. Reducer. СРС: Спробувати переробити четверту лабораторну роботу в рамках даного кредитного модулю з використанням Redux.	7
10	«Angular». Основи Angular. Компоненти. Модулі. Стилі та шаблони компонента. Прив'язка даних. Взаємодія між компонентами. Прив'язка до подій дочірнього компонента. Життєвий цикл компоненту. Взаємодія між модулями. Директиви. Сервіси та dependency injection. Робота з формами. HTTP та взаємодія з сервером. Маршрутизація. Pipes. СРС: Частково або повністю виконати п'яту лабораторну роботу даного кредитного модуля, використовуючи Angular. Спробувати додати до реалізації п'ятої лабораторної роботи в рамках даного кредитного модулю використання методів життєвого циклу компоненту, взаємодії між модулями, директив, сервісів та dependency injection. Спробувати додати до реалізації п'ятої лабораторної роботи в рамках даного кредитного модулю використання роботи з формами, HTTP та взаємодії з сервером, маршрутизація та pipes.	7
11	«Vue». Основи Vue 3. Об'єкт Vue. Прив'язка даних. Обробка введення користувача. Події. Двостороння прив'язка. Refs та управління HTML-елементами. Життєвий цикл Vue. Умовний рендеринг та робота з масивами. Робота з формами. Компоненти. Слоти. Маршрутизація. СРС: Частково або повністю виконати шосту лабораторну роботу даного кредитного модуля, використовуючи Vue. Спробувати додати до реалізації шостої лабораторної роботи в рамках даного кредитного модулю роботу з формами, умовний рендеринг, роботу з масивами, компоненти, слоти та маршрутизацію.	7
12	«Підготовка до МКР»	7
13	«Підготовка до заліку»	7
	Разом:	104

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

- правила захисту лабораторних робіт:
 - студент надає програмний код до лабораторної роботи,
 - студент захищає наданий програмний код шляхом співбесіди;
- політика дедлайнів та перескладань:
 - студент має право здавати та перездавати лабораторні роботи до дня заліку включно,
 - обмеження на кількість перескладань відсутнє;
- політика щодо академічної доброчесності:
 - студент має право захищати програмний код, який був написаний не лише ним, але в цьому випадку студент має розуміти «що» та «як» виконує цей код, вміти використати його як основу для внесення модифікацій, які можуть бути запропоновані викладачем.

8. Види контролю та рейтингова система оцінювання результатів навчання (PCO)

Види контролю з навчальної дисципліни «Технології програмування користувацьких інтерфейсів (Front-end)» включають:

Лабораторні роботи:

Заплановано самостійне виконання шести лабораторних робіт. Темі лабораторних робіт узгоджені у часі та за змістом з темами лекцій. Виконання лабораторних робіт у повному обсязі дозволяє набутти практичних навичок програмування користувацьких інтерфейсів.

Поточний контроль:

Передбачено проведення контрольної роботи (МКР)

Семестровий контроль:

Залік проводиться у вигляді співбесіди зі студентом для об'єктивного визначення рівня знань, умінь та практичних навичок, отриманих за семестр

Семестровий рейтинг студента складається з балів, які він отримує за види робіт відповідно до таблиці 1.

Таблиця 1

Оцінювання окремих видів навчальної роботи студента (у балах)

Вид навчальної роботи	Всього за видом роботи
Виконання та захист лабораторної роботи № 1	15
Виконання та захист лабораторної роботи № 2	15
Виконання та захист лабораторної роботи № 3	15
Виконання та захист лабораторної роботи № 4	15
Виконання та захист лабораторної роботи № 5	15
Виконання та захист лабораторної роботи № 6	15
МКР Rк	10
Rп	100
Залік (Rз) за бажанням	20

Індивідуальний поточний рейтинг студента (**Rп**) складається з балів, які він отримує за виконання лабораторних робіт (**Rл**) і МКР (**Rк**). Протягом семестру студенти виконують 6 лабораторних робіт. Максимальна кількість балів за кожну лабораторну роботу – 15. Бали нараховуються за:

- теоретична складова – 7 балів,
- практична складова – 8 балів.

Максимальний можливий бал за лабораторну роботу – 15 балів. Максимальна кількість балів за всі лабораторні роботи $6 \times 15 = 90$ балів.

Розрахунок розміру шкали (R) рейтингу.

Сума вагових балів контрольних заходів протягом семестру становить:

Rп = Rл + Rк, де Rп – семестровий рейтинг студента (МКР, лабораторні роботи).

Необхідною умовою допуску студента до заліку є його індивідуальний семестровий рейтинг (Rp), не менший, ніж 40 балів, та відсутність заборгованості з лабораторних робіт та МКР. При невиконанні згаданих вимог студент до заліку не допускається.

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

Кількість балів	Оцінка
100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно
64-60	Достатньо
Менше 60	Незадовільно
Не виконані умови допуску	Не допущено

9. Додаткова інформація з дисципліни (освітнього компонента)

В рамках вивчення дисципліни «Технології програмування користувацьких інтерфейсів (Front-end)» допускається зарахування балів, одержаних в результаті дистанційних курсів за умови погодження програми даного курсу з викладачем.

Робочу програму навчальної дисципліни (силабус):

Складено старшим викладачем кафедри обчислювальної техніки Алещенком Олексієм Вадимовичем

Ухвалено кафедрою обчислювальної техніки (протокол № 10 від 25.05.2022)

Погоджено Методичною комісією факультету (протокол № 10 від 09.06.2022)