

ДЕРЖАВНИЙ ТОРГОВЕЛЬНО-ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ

СИСТЕМА УПРАВЛІННЯ ЯКІСТЮ

Система забезпечення якості освітньої діяльності та якості вищої освіти
сертифікована на відповідність ДСТУ ISO 9001:2015 / ISO 9001:2015

Кафедра комп'ютерних наук та інформаційних систем



WEB-ТЕХНОЛОГІЇ/
WEB TECHNOLOGIES

РОБОЧА ПРОГРАМА /
COURSE OUTLINE

освітній ступінь	бакалавр	/	bachelor
галузь знань	<u>12 Інформаційні технології</u>	/	<u>Information Technology</u>
спеціальність	<u>126 Інформаційні системи та технології</u>	/	<u>Information Systems and Technologies</u>
освітня програма	<u>Інформаційні системи та технології</u>	/	<u>Information Systems and Technologies</u>

Київ 2022

Розповсюдження і тиражування без офіційного дозволу ДТЕУ заборонено

Автори: Г. Т. САМОЙЛЕНКО, кандидат фізико-математичних наук, доцент,
А.В. СЕЛІВАНОВА, старший викладач.

Робочу програму розглянуто і затверджено на засіданні кафедри комп'ютерних наук та інформаційних систем 19.07.2022р., протокол № 38

Рецензенти: Т.О. ФІЛІМОНОВА, канд.фіз.-матем. наук, доцент.
С.П. КУДРЯВЦЕВА, канд.техн.н., провідний науковий співробітник Міжнародного науково-навчального центру інформаційних технологій та систем НАН України.

WEB-ТЕХНОЛОГІЇ / WEB TECHNOLOGIES

РОБОЧА ПРОГРАМА / COURSE OUTLINE

освітній ступінь	бакалавр	/	Bachelor
галузь знань	<u>12 Інформаційні технології</u>	/	<u>Information Technology</u>
спеціальність	<u>126 Інформаційні системи та технології</u>	/	<u>Information systems and technologies</u>
освітня програма	<u>Інформаційні системи та технології</u>	/	<u>Information systems and technologies</u>

1. СТРУКТУРА ДИСЦИПЛІНИ ТА РОЗПОДІЛ ГОДИН ЗА ТЕМАМИ (ТЕМАТИЧНИЙ ПЛАН)

Назва теми	Кількість годин				Форми контролю
	Усього годин / кредитів	з них			
		лекції	лабораторні заняття	самостійна робота студентів	
1	2	3	4	5	6
Тема 1. Структура і принципи сучасних Web-технологій.	15	2	2	11	О, ПСР, ПЛР
Тема 2. Введення в CSS. Блочне оформлення структури сайту.	15	2	2	11	О, ПСР, ПЛР
Тема 3. Програмне управління HTML-документом.	15	2	2	11	О, ПСР, ПЛР
Тема 4. Серверні WEB-застосування.	15	2	2	11	О, ПСР, ПЛР
Тема 5. Адаптивна верстка. Технології Flexbox та Grid CSS.	15	2	2	11	О, ПСР, ПЛР
Тема 6. JavaScript.	15	2	2	11	О, ПСР, ПЛР
Тема 7. Графічний редактор Figma.	15	2	2	11	О, ПСР, ПЛР
Тема 8. Розподілена архітектура.	15	2	2	11	О, ПСР, ПЛР
Тема 9. Стандарти розробки front-end.	15	2	2	11	О, ПСР, ПЛР
Тема 10. Стандарти розробки backend	15	2	2	11	О, ПСР, МК, ПЛР
Тема 11. Контейнеризація.	15	2	2	11	О, ПСР, ПЛР
Тема 12. Система керування версіями.	15	2	2	11	О, ПСР, ПЛР
Разом	180/6	24	24	132	
Підсумковий контроль – екзамен					

Умовні позначення: ПСР – перевірка самостійної роботи; МК – модульний контроль; ПЛР – перевірка лабораторної роботи; О – опитування.

2. ТЕМАТИКА ТА ЗМІСТ ЛЕКЦІЙНИХ, ПРАКТИЧНИХ (СЕМІНАРСЬКИХ), ЛАБОРАТОРНИХ ЗАНЯТЬ, САМОСТІЙНОЇ РОБОТИ СТУДЕНТІВ

Результати навчання	Навчальна діяльність*	Робочий час студента, год
1	2	3
<p>Знати: принципи сучасних Web-технологій</p> <p>Вміти: використовувати отримані теоретичні знання про створення динамічних прототипів в практичній діяльності.</p>	<p>Тема 1. Структура і принципи сучасних Web-технологій.</p> <p>Лекція № 1. Структура і принципи сучасних Web-технологій.</p> <p>План лекції:</p> <ol style="list-style-type: none"> 1. Поняття UX і UI. 2. Сучасний дизайн сайту, типи. 3. Проектування логічної структури сайту. Створення динамічних прототипів. Візуальне оформлення сайту. 4. Взаємодія з користувачем. Клієнт-серверна WEB архітектура. 5. Застосування протоколу HTTP. Структура запитів клієнта та відповідей сервера. 6. Загальні поняття про мову гіпертекстової розмітки. Структура HTML-документу. <p style="text-align: center;">Список рекомендованих джерел</p> <p><i>Основний: 1[с.12-35]</i></p> <p><i>Додатковий: 2 [с.10-40], 3 [с.15-68]</i></p> <p><i>Інтернет-ресурси:1,3</i></p>	15
	<p style="text-align: center;">Самостійна робота</p> <p>Вивчення матеріалу до теми «Структура і принципи сучасних Web-технологій», підготовка до лабораторного заняття.</p> <p>Самостійна робота передбачає вивчення окремих питань дисципліни на основі опрацювання літератури та пошуку інформаційних джерел у середовищі Інтернет. Пропонуються такі питання для самостійного опрацювання:</p> <ol style="list-style-type: none"> 1. Елементи, групи тегів та атрибутів HTML-документу. 2. Створення каркасу HTML-сторінки за допомогою таблиць. 3. Використання списків та таблиць в HTML-документі. 4. Створення форм. Блочна верстка HTML-документу за макетом. 5. Створення шаблону за макетом за 	11

	<p>допомогою блочних елементів. Самостійна робота студентів перевіряється з допомогою опитувань, перевірки виконання лабораторних робіт та написання рефератів на питання винесених на самостійне опрацювання.</p> <p>Список рекомендованих джерел <i>Основний: 1[с.12-35]</i> <i>Додатковий: 2 [с.10-40], 3 [с.15-68]</i> <i>Інтернет-ресурси:1,3</i></p>	
	<p>Лабораторне заняття №1. Тема: «Структура і принципи сучасних Web-технологій». <i>Завдання до заняття: створити презентацію, яка ілюструє структуру та принципи сучасних веб-технологій. Робота повинна включати наступне:</i></p> <ol style="list-style-type: none"> <i>1. Огляд технологій: короткий огляд HTML, CSS та JavaScript як основних мов веб-розробки.</i> <i>2. Frontend технології: детальний опис бібліотек та фреймворків для розробки відгукнутого інтерфейсу, таких як React, Angular, Vue.js тощо.</i> <i>3. Backend технології: пояснення серверних технологій, таких як Node.js, Django, Ruby on Rails, та їх роль у створенні та управлінні веб-додатками.</i> <i>4. Бази даних: огляд різних типів баз даних та їхнє використання у веб-розробці, включаючи SQL (MySQL, PostgreSQL) і NoSQL (MongoDB, Firebase).</i> <i>5. Хмарні технології: пояснення ролі хмарних сервісів у розгортанні, масштабуванні та забезпеченні безпеки веб-додатків.</i> <i>6. Мобільна адаптація та безпека: опис методів роботи з різними пристроями та заходів забезпечення безпеки веб-додатків.</i> <i>7. Підсумок: загальний огляд основних принципів, які лежать в основі сучасних веб-технологій та їх вплив на розвиток інтернет-індустрії.</i> <i>8. Підготувати звіт про виконання лабораторної роботи.</i> 	2
<p>Знати: особливості оформлення</p>	<p>Тема 2. Введення в CSS. Блочне оформлення структури сайту. Лекція № 2. Введення в CSS. Блочне</p>	15

<p>структури сайту Вміти: використовувати отримані теоретичні знання про блочне оформлення структури сайту в практичній діяльності.</p>	<p>оформлення структури сайту. План лекції: 1. Технологія CSS та її підтримка браузерями. 2. Синтаксис, правила CSS. 3. Каскадування. 4. Ідентифікації елементів HTML-документів за допомогою селекторів. Список рекомендованих джерел <i>Основний: 1[с.38-60]</i> <i>Додатковий: 2 [с.68-90], 3 [с.15-68]</i> <i>Інтернет-ресурси:1,3</i></p>	2
	<p>Самостійна робота Вивчення матеріалу до теми «Введення в CSS. Блочне оформлення структури сайту», підготовка до лабораторного заняття. Самостійна робота передбачає вивчення окремих питань дисципліни на основі опрацювання літератури та пошуку інформаційних джерел у середовищі Інтернет. Пропонуються такі питання для самостійного опрацювання: 1. Класи. 2. Класифікація макетів. 3. Блочна верстка HTML-документу за макетом. 4. Створення шаблону за макетом за допомогою блочних елементів. Самостійна робота студентів перевіряється з допомогою опитувань, перевірки виконання лабораторних робіт та написання рефератів на питання винесених на самостійне опрацювання. Список рекомендованих джерел <i>Основний: 1[с.38-60]</i> <i>Додатковий: 2 [с.68-90], 3 [с.15-68]</i> <i>Інтернет-ресурси:1,3</i></p>	11
	<p>Лабораторне заняття №2. Тема: «Введення в CSS. Блочне оформлення структури сайту». <i>Завдання до заняття: створити веб-сайт із використанням HTML та CSS, із застосуванням основних принципів блочного оформлення для створення структури сайту.</i> 1. Створити структуру HTML-сторінки, включаючи заголовок, меню, основний вміст та футер. 2. Написати базовий CSS для сайту із застосуванням стилів до заголовку, меню,</p>	2

	<p><i>основного вмісту та футера.</i></p> <p>3. Застосувати концепцію блочного оформлення для створення різних блоків на сторінці. Наприклад, створити бокову панель, де розміщене меню, та основний блок з контентом.</p> <p>4. Підготувати звіт про виконання лабораторної роботи.</p>	
<p>Знати: основні принципи програмного управління HTML-документом</p> <p>Вміти: використовувати отримані теоретичні знання про використання JavaScript у HTML-документах в практичній діяльності.</p>	<p>Тема 3. Програмне управління HTML-документом.</p> <p>Лекція № 3. Програмне управління HTML-документом.</p> <p>План лекції:</p> <ol style="list-style-type: none"> 1. Функціональне використання JavaScript у HTML-документах. 2. Оголошення змінних. Типи даних. 3. Простір імен. 4. Операнди. Пріоритет операторів. 5. Властивості і методи ключових об'єктів. 6. Визначення і виклик функцій. <p>Список рекомендованих джерел</p> <p><i>Основний: 1[с. 65-90]</i></p> <p><i>Додатковий: 3 [с.90-126]</i></p> <p><i>Інтернет-ресурси:1,3</i></p>	<p>15</p> <p>2</p>
	<p>Самостійна робота</p> <p>Вивчення матеріалу до теми «Програмне управління HTML-документом», підготовка до лабораторного заняття.</p> <p>Самостійна робота передбачає вивчення окремих питань дисципліни на основі опрацювання літератури та пошуку інформаційних джерел у середовищі Інтернет. Пропонуються такі питання для самостійного опрацювання:</p> <ol style="list-style-type: none"> 1. Властивості і методи функцій. 2. Класи. 3. Створення об'єктів. 4. Конструктори. 5. Події та обробка подій. 6. Подання документів у вигляді дерева. 7. Вузли. 8. Пошук елементів у документі. 9. Модифікація документів. <p>Самостійна робота студентів перевіряється з допомогою опитувань, перевірки виконання лабораторних робіт та написання рефератів на питання винесених на самостійне опрацювання.</p>	<p>11</p>

	<p align="center">Список рекомендованих джерел</p> <p align="center"><i>Основний: 1[с. 65-90]</i></p> <p align="center"><i>Додатковий: 3 [с.90-126]</i></p> <p align="center"><i>Інтернет-ресурси:1,3</i></p>	
	<p>Лабораторне заняття №3. Тема: «Програмне управління HTML-документом».</p> <p><i>Завдання до заняття: створити міні-проект, з використанням JavaScript для програмного управління HTML-документом.</i></p> <ol style="list-style-type: none"> <i>1. Створити базову HTML-сторінку з елементами, які будуть взаємодіяти за допомогою JavaScript. Наприклад, кнопки, текстові поля.</i> <i>2. Написати JavaScript-код, який може динамічно додавати та видаляти елементи на сторінці. Наприклад, додавання нового елемента до списку або видалення елемента за допомогою кнопки.</i> <i>3. Розробити функціонал, який дозволяє змінювати стилі елементів на сторінці. Наприклад, змінювати колір тексту, фону чи розміри елементів.</i> <i>4. Реалізувати можливість взаємодії з формами на сторінці за допомогою JavaScript. Наприклад, перевірка введених даних перед відправленням форми чи відображення повідомлення після успішної відправки.</i> <i>5. Підготувати звіт про виконання лабораторної роботи.</i> 	2
<p align="center">Знати: особливості серверних WEB-застосувань</p> <p align="center">Вміти: використовувати отримані теоретичні знання про серверні WEB-застосування в практичній діяльності.</p>	<p align="center">Тема 4. Серверні WEB-застосування.</p> <p align="center">Лекція № 4. Серверні WEB-застосування.</p> <p>План лекції:</p> <ol style="list-style-type: none"> 1. Структура Web-сервера. PHP: препроцесор гіпертексту. 2. Структура програм на мові PHP. 3. Мова розроблення серверних сценаріїв PHP. 4. Типи даних, змінні, оператори. Операції і керуючі конструкції. 5. Цикли. 6. Функції та повторне використання коду. <p align="center">Список рекомендованих джерел</p> <p align="center"><i>Основний: 1[с.92-110]</i></p> <p align="center"><i>Додатковий:3 [с.130-184]</i></p> <p align="center"><i>Інтернет-ресурси:1,3</i></p>	15 2
	Самостійна робота	11

Вивчення матеріалу до теми «Серверні WEB-застосування», підготовка до лабораторного заняття.

Самостійна робота передбачає вивчення окремих питань дисципліни на основі опрацювання літератури та пошуку інформаційних джерел у середовищі Інтернет. Пропонуються такі питання для самостійного опрацювання:

1. Вбудовані функції.
2. Визначення і виклик функцій.
3. Структура Web-сервера.
4. PHP: препроцесор гіпертексту.
5. Структура програм на мові PHP.
6. Мова розроблення серверних сценаріїв PHP.
7. Типи даних, змінні, оператори.
8. Операції і керуючі конструкції.
9. Функції та повторне використання коду.

Самостійна робота студентів перевіряється з допомогою опитувань, перевірки виконання лабораторних робіт та написання рефератів на питання винесених на самостійне опрацювання.

Список рекомендованих джерел

Основний: 1[с.92-110]

Додатковий:3 [с.130-184]

Інтернет-ресурси:1,3

Лабораторне заняття №4. Тема: «Серверні WEB-застосування».

Завдання до заняття: створити простий серверний веб-додаток, який відповідає на запити та повертає інформацію.

1. Використати Node.js (або будь-яку іншу платформу) для створення простого веб-сервера.
2. Налаштувати сервер так, щоб він відповідав на запити GET/POST та повертав певну інформацію. Наприклад, повертати текстовий відгук.
3. Додати можливість серверу повертати статичні файли (HTML, CSS, JavaScript тощо), які можна переглянути через браузер.
4. Додати декілька маршрутів для сервера, які реагують на різні URL-адреси та повертають різну інформацію.
5. Забезпечити можливість додатку повертати дані з бази даних (якщо вони

	<p>використовуються), або фіктивні дані змістовної інформації.</p> <p>6. Підготувати звіт про виконання лабораторної роботи.</p>	
<p>Знати: технології Flexbox та Grid CSS</p> <p>Вміти: використовувати отримані теоретичні знання про адаптивну верстку в практичній діяльності.</p>	<p>Тема 5. Адаптивна верстка. Технології Flexbox та Grid CSS.</p> <p>Лекція № 5. Адаптивна верстка. Технології Flexbox та Grid CSS.</p> <p>План лекції:</p> <ol style="list-style-type: none"> 1. Поняття адаптивності сайту, особливості та технології створення адаптивних сторінок. 2. Flexbox. 3. Особливості відображення елементів. 4. Задачі, які вирішуються за допомогою Flexbox. 5. Grid CSS. <p>Список рекомендованих джерел</p> <p><i>Основний: 1[с. 112-135]</i></p> <p><i>Додатковий: 3 [с.188-220]</i></p> <p><i>Інтернет-ресурси: 1, 3</i></p>	<p>15</p> <p>2</p>
	<p>Самостійна робота</p> <p>Вивчення матеріалу до теми «Адаптивна верстка. Технології Flexbox та Grid CSS», підготовка до лабораторного заняття.</p> <p>Самостійна робота передбачає вивчення окремих питань дисципліни на основі опрацювання літератури та пошуку інформаційних джерел у середовищі Інтернет. Пропонуються такі питання для самостійного опрацювання:</p> <ol style="list-style-type: none"> 1. Створення і робота з флекс-контейнерами та флекс-елементами. 2. Задачі, які вирішуються за допомогою Grid CSS. 3. Робота з Grid-контейнерами та Grid-елементами. <p>Самостійна робота студентів перевіряється з допомогою опитувань, перевірки виконання лабораторних робіт та написання рефератів на питання винесених на самостійне опрацювання.</p> <p>Список рекомендованих джерел</p> <p><i>Основний: 1[с. 112-135]</i></p> <p><i>Додатковий: 3 [с.188-220]</i></p> <p><i>Інтернет-ресурси: 1, 3</i></p>	<p>11</p>
	<p><i>Лабораторне заняття №5. Тема: «Адаптивна</i></p>	<p>2</p>

	<p><i>верстка. Технології Flexbox та Grid CSS».</i></p> <p><i>Завдання до заняття: створити веб-сторінку з використанням Flexbox та Grid CSS, яка буде адаптивною для різних розмірів екранів.</i></p> <ol style="list-style-type: none"> <i>1. Створити базову HTML-структуру сторінки, яка включатиме різні секції: шапка, контент, бокова панель, футер тощо.</i> <i>2. Використати Flexbox для організації та компоновання елементів у секціях сторінки. Наприклад, розмістити елементи у шапці або контенті за допомогою Flexbox.</i> <i>3. Використати Grid CSS для створення сітки сторінки (розташувати різні блоки контенту або секції сторінки у вигляді сітки для кращого розташування на екрані)</i> <i>4. Зробити сторінку адаптивною до різних розмірів екранів за допомогою медіа-запитів.</i> <i>5. Підготувати звіт про виконання лабораторної роботи.</i> 	
<p>Знати: основні поняття JavaScript</p> <p>Вміти: використовувати отримані теоретичні знання про використання фреймворків JQuery в практичній діяльності.</p>	<p align="center">Тема 6. JavaScript.</p> <p align="center">Лекція № 6. JavaScript.</p> <p>План лекції:</p> <ol style="list-style-type: none"> 1. Визначення JavaScript. Основні поняття. 2. Способи додання на сторінку. Типи даних. 3. Використання фреймворків JQuery. 4. Правила задання типів даних. 5. Робота з об'єктами і класами. <p>Список рекомендованих джерел</p> <p><i>Основний: 1[с. 140-156]</i></p> <p><i>Додатковий:3 [с.280-360]</i></p> <p><i>Інтернет-ресурси:3,4</i></p>	<p align="center">15</p> <p align="center">2</p>
	<p align="center">Самостійна робота</p> <p>Вивчення матеріалу до теми «JavaScript», підготовка до лабораторного заняття. Самостійна робота передбачає вивчення окремих питань дисципліни на основі опрацювання літератури та пошуку інформаційних джерел у середовищі Інтернет. Пропонуються такі питання для самостійного опрацювання:</p> <ol style="list-style-type: none"> 1. Функції для роботи над даними. 2. Функції вводу та виведення інформації. 3. Умовні вирази. 4. Операції відношень. 5. Цикли. Оператори виходу з циклу. 	<p align="center">11</p>

	<p>6. Типи об'єктів. 7. Робота з елементами документа. 8. Робота з подіями. Робота з DOM.</p> <p>Самостійна робота студентів перевіряється з допомогою опитувань, перевірки виконання лабораторних робіт та написання рефератів на питання винесених на самостійне опрацювання.</p> <p style="text-align: center;">Список рекомендованих джерел</p> <p style="text-align: center;"><i>Основний: 1[с. 140-156]</i> <i>Додатковий:3 [с.280-360]</i> <i>Інтернет-ресурси:3,4</i></p>	
	<p>Лабораторне заняття №6. Тема: «JavaScript». Завдання до заняття: створити To-Do список, який дозволить користувачам додавати, видаляти та позначати завдання як виконані, використовуючи JavaScript.</p> <ol style="list-style-type: none"> 1. Створити основний HTML для відображення To-Do списку. Додати форму для введення нових завдань, список для відображення завдань та кнопки для видалення або позначення завдань. 2. Написати JavaScript-код, який додає можливість додавання нових завдань у список, видалення завдань та позначення їх як виконані. Використати обробники подій для кнопок та форми. 3. Забезпечити зберігання даних про завдання локально, використовуючи <code>localStorage</code>, щоб при оновленні сторінки список завдань не зникав. 4. Додати CSS, щоб стилізувати To-Do список. Додати функціонал, такий як редагування завдань, сортування за різними параметрами або фільтрація виконаних завдань. 5. Підготувати звіт про виконання лабораторної роботи. 	2
<p style="text-align: center;">Знати: особливості створення макетів</p> <p style="text-align: center;">Вміти: використовувати отримані теоретичні знання про створення макетів в практичній діяльності.</p>	<p style="text-align: center;">Тема 7. Графічний редактор Figma. Лекція № 7. Графічний редактор Figma.</p> <p>План лекції:</p> <ol style="list-style-type: none"> 1. Графічний редактор Figma. 2. Робота з базовими фігурами та ефектами. 3. Створення макету. 4. Фотографії та монтажні області. <p style="text-align: center;">Список рекомендованих джерел</p> <p style="text-align: center;"><i>Основний: 1[с. 140-156]</i></p>	15 2

	<p><i>Додатковий: 3 [с.420-485]</i> <i>Інтернет-ресурси:2,3</i></p>	
	<p align="center">Самостійна робота</p> <p>Вивчення матеріалу до теми «Графічний редактор Figma», підготовка до лабораторного заняття.</p> <p>Самостійна робота передбачає вивчення окремих питань дисципліни на основі опрацювання літератури та пошуку інформаційних джерел у середовищі Інтернет. Пропонуються такі питання для самостійного опрацювання:</p> <ol style="list-style-type: none"> 1. Робота з текстами. 2. Прототипування та автоматизація. 3. Авторозмітка та компоненти. 4. Інтерактивний прототип. <p>Самостійна робота студентів перевіряється з допомогою опитувань, перевірки виконання лабораторних робіт та написання рефератів на питання винесених на самостійне опрацювання.</p> <p align="center">Список рекомендованих джерел</p> <p><i>Основний: 1[с. 140-156]</i> <i>Додатковий: 3 [с.420-485]</i> <i>Інтернет-ресурси:2,3</i></p>	11
	<p><i>Лабораторне заняття №7. Тема: «Графічний редактор Figma».</i></p> <p><i>Завдання до заняття: розробити дизайн лендінгу для веб-сайту за допомогою графічного редактора Figma.</i></p> <ol style="list-style-type: none"> 1. <i>Обрати тему для лендінгу - це може бути будь-що, від продукту або послуги до події чи промо-сторінки.</i> 2. <i>Створити макет лендінгу, використовуючи інструменти Figma. Розмістити елементи так, щоб вони були зручні для користувача та естетично зорієнтовані.</i> 3. <i>Додати текстовий контент та мультимедійні елементи (зображення, відео тощо) для наочного представлення обраного продукту чи послуги.</i> 4. <i>Визначити кольорову схему та шрифти, які будуть використані для створення єдиної стилістики для лендінгу.</i> 5. <i>Створити мобільну версію лендінгу, враховуючи особливості відображення на малих екранах.</i> 	2

	<i>6. Підготувати звіт про виконання лабораторної роботи.</i>	
<p>Знати: особливості розподілених архітектур</p> <p>Вміти: використовувати отримані знання про розподілені архітектури в практичній діяльності.</p>	<p>Тема 8. Розподілена архітектура. Лекція № 8. Розподілена архітектура.</p> <p>План лекції:</p> <ol style="list-style-type: none"> 1. Клієнт-Серверна архітектура (Client-Server Architecture). 2. Мікросервісна архітектура (Microservices Architecture). 3. Хмарна архітектура (Cloud-Based Architecture). <p>Список рекомендованих джерел <i>Основний: 2 [с.80-112]</i> <i>Додатковий: 2 [с. 35-60]</i> <i>Інтернет-ресурси: 4,5,6</i></p>	<p>15</p> <p>2</p>
	<p>Самостійна робота</p> <p>Вивчення матеріалу до теми «Розподілена архітектура», підготовка до лабораторного заняття.</p> <p>Самостійна робота передбачає вивчення окремих питань дисципліни на основі опрацювання літератури та пошуку інформаційних джерел у середовищі Інтернет. Пропонуються такі питання для самостійного опрацювання:</p> <ol style="list-style-type: none"> 1. Системи, засновані на повідомленнях (Message-Oriented Middleware). 2. Peer-to-Peer (P2P) архітектура. 3. Сервісно-орієнтована архітектура (Service-Oriented Architecture, SOA). 4. Event-Driven Architecture (EDA). 5. Space-Based Architecture (Space-Based Computing). 6. Data-Centric Architecture. <p>Самостійна робота студентів перевіряється з допомогою опитувань, перевірки виконання лабораторних робіт та написання рефератів на питання винесених на самостійне опрацювання.</p> <p>Список рекомендованих джерел <i>Основний: 2 [с.80-112]</i> <i>Додатковий: 2 [с. 35-60]</i> <i>Інтернет-ресурси: 4,5,6</i></p>	11
	<p>Лабораторне заняття №8. Тема: «Розподілена архітектура».</p> <p><i>Завдання до заняття: розробити концепцію системи електронної комерції з використанням</i></p>	2

	<p><i>розподіленої архітектури.</i></p> <ol style="list-style-type: none"> 1. <i>Описати функціональні та нефункціональні вимоги до системи електронної комерції. Визначити, які сервіси та функції повинні бути доступні.</i> 2. <i>Спроектувати розподілену архітектуру для системи. Розглянути використання мікросервісної архітектури, шаблону CQRS (Command Query Responsibility Segregation), Event Sourcing або інших відомих шаблонів.</i> 3. <i>Описати механізми комунікації між сервісами вашої системи. Розглянути використання API, інтеграційні технології, потоки подій тощо.</i> 4. <i>Розглянути аспекти безпеки системи та можливості масштабування різних компонентів архітектури.</i> 5. <i>Визначити технологічний стек, який буде використовуватися для реалізації кожного компонента системи.</i> 6. <i>Створити схематичне представлення розподіленої архітектури, використовуючи засоби моделювання або діаграми.</i> 7. <i>Підготувати звіт про виконання лабораторної роботи.</i> 	
<p>Знати: поняття стандартів розробки front-end</p> <p>Вміти: використовувати отримані теоретичні знання про стандарти розробки front-end в практичній діяльності.</p>	<p>Тема 9. Стандарти розробки front-end. Лекція № 9. Стандарти розробки front-end. План лекції:</p> <ol style="list-style-type: none"> 1. ECMAScript та JavaScript-стандарти. 2. Використання стандарту ECMAScript та дотримання сучасних практик розробки JavaScript. 3. Дотримання семантичних правил HTML, використання HTML5-елементів та атрибутів для підтримки функціоналу та структуризації документів. 4. Використання CSS-препроцесорів та дотримання CSS-стандартів для підтримки рефакторингу. <p>Список рекомендованих джерел <i>Основний: 2 [с.80-112] Додатковий: 2 [с.60-82], 3 [с. 35-60] Інтернет-ресурси: 3</i></p>	<p>15</p> <p>2</p>
	<p>Самостійна робота Вивчення матеріалу до теми «Стандарти розробки front-end», підготовка до</p>	<p>11</p>

	<p>лабораторного заняття.</p> <p>Самостійна робота передбачає вивчення окремих питань дисципліни на основі опрацювання літератури та пошуку інформаційних джерел у середовищі Інтернет. Пропонуються такі питання для самостійного опрацювання:</p> <ol style="list-style-type: none"> 1. Розробка з використанням компонентів (Component-Based Development). 2. Responsive Web Design (RWD): розробка веб-інтерфейсу, що ефективно працює на різних пристроях та екранах. 3. Accessibility (доступність). 4. Performance Optimization (оптимізація продуктивності). 5. Code Versioning (керування версіями коду). 6. Використання систем керування версіями для ведення історії змін коду та спільної роботи в команді. <p>Самостійна робота студентів перевіряється з допомогою опитувань, перевірки виконання лабораторних робіт та написання рефератів на питання винесених на самостійне опрацювання.</p> <p style="text-align: center;">Список рекомендованих джерел</p> <p style="text-align: center;"><i>Основний: 2 [с.80-112]</i></p> <p style="text-align: center;"><i>Додатковий: 2 [с.60-82], 3 [с. 35-60]</i></p> <p style="text-align: center;"><i>Інтернет-ресурси: 3</i></p>	
	<p>Лабораторне заняття №9. <i>Тема: «Стандарти розробки front-end».</i></p> <p><i>Завдання до заняття: розробити набір стилів та компонентів, використовуючи встановлені стандарти front-end розробки.</i></p> <ol style="list-style-type: none"> 1. <i>Ознайомитись зі стандартами кодування, семантикою HTML, методологіями CSS (наприклад, BEM), правилами форматування та назвами класів.</i> 2. <i>Розробити компоненти, такі як кнопки, форми, картки і т. д., дотримуючись обраних стандартів.</i> 3. <i>Створити CSS-стили для цих компонентів, використовуючи підходи, зазначені в стандартах, і дотримуючись їхньої структури та назв класів.</i> 4. <i>Зробити компоненти адаптивними для різних екранів.</i> 5. <i>Перевірити компоненти наскільки вони</i> 	2

	<p><i>відображаються правильно та відповідають встановленим стандартам.</i></p> <p>6. Підготувати звіт про виконання лабораторної роботи.</p>	
<p>Знати: особливості стандартів розробки backend</p> <p>Вміти: використовувати отримані знання про стандарти розробки в практичній діяльності.</p>	<p>Тема 10. Стандарти розробки backend. Лекція № 10. Стандарти розробки backend. План лекції:</p> <ol style="list-style-type: none"> 1. RESTful API Design: використання REST або подібного до REST підходу для створення API. 2. Захист даних та забезпечення аутентифікації та авторизації. 3. Використання протоколів HTTPS для шифрування з'єднань. 4. Використання Систем Керування Базами Даних (Database Management Systems - DBMS). <p>Список рекомендованих джерел <i>Основний: 2[с.120-142]</i> <i>Додатковий: 2 [с.60-82]</i> <i>Інтернет-ресурси: 4,7</i></p>	<p>15</p> <p>2</p>
	<p>Самостійна робота</p> <p>Вивчення матеріалу до теми «Стандарти розробки backend», підготовка до лабораторного заняття.</p> <p>Самостійна робота передбачає вивчення окремих питань дисципліни на основі опрацювання літератури та пошуку інформаційних джерел у середовищі Інтернет. Пропонуються такі питання для самостійного опрацювання:</p> <ol style="list-style-type: none"> 1. Логування (Logging): Логування подій та помилок для полегшення відладки та моніторингу. 2. Документація коду (Code Documentation). 3. Масштабованість (Scalability). 4. Архітектурні патерни (Architectural Patterns). 5. Керування Залежностями (Dependency Management). 6. Стандартизація форматів обміну даними. <p>Самостійна робота студентів перевіряється з допомогою опитувань, перевірки виконання лабораторних робіт та написання рефератів на питання винесених на самостійне опрацювання.</p>	<p>11</p>

	<p align="center">Список рекомендованих джерел</p> <p align="center"><i>Основний: 2[с.120-142]</i></p> <p align="center"><i>Додатковий: 2 [с.60-82]</i></p> <p align="center"><i>Інтернет-ресурси: 4,7</i></p>	
	<p>Лабораторне заняття №10. Тема: «Стандарти розробки backend».</p> <p>Завдання до заняття: розробити API з використанням стандартів розробки backend.</p> <ol style="list-style-type: none"> 1. Визначення функціональності API: описати, які операції та функції повинно виконувати API. Це може бути система управління користувачами, CRUD операції для деяких даних, аутентифікація та авторизація тощо). 2. Вибір технологій: обрати мову програмування та фреймворк для розробки API. 3. Розробка API: створити реалістичну реалізацію API з урахуванням RESTful принципів. 4. Документація API: за допомогою засобів (наприклад, Swagger, Postman тощо), задокументувати API, зазначаючи кожен ендпоінт, методи, параметри та відповіді. 5. Тестування та перевірка працездатності. 6. Підготувати звіт про виконання лабораторної роботи. 	2
<p>Знати: поняття контейнеризації</p> <p>Вміти: використовувати отримані теоретичні знання про Docker Compose в практичній діяльності.</p>	<p align="center">Тема 11. Контейнеризація.</p> <p align="center">Лекція № 11. Контейнеризація.</p> <p>План лекції:</p> <ol style="list-style-type: none"> 1. Контейнеризація. 2. Віртуальні машини і контейнери. 3. Платформа Docker та її компоненти. 4. Образ і контейнер. 5. Реєстр образів. Базові команди Docker. <p align="center">Список рекомендованих джерел:</p> <p align="center"><i>Основний: 2[с.120-142]</i></p> <p align="center"><i>Додатковий: 2 [с.60-82]</i></p> <p align="center"><i>Інтернет-ресурси: 4,6</i></p>	15
	<p align="center">Самостійна робота</p> <p>Вивчення матеріалу до теми «Контейнеризація», підготовка до лабораторного заняття.</p> <p>Самостійна робота передбачає вивчення окремих питань дисципліни на основі опрацювання літератури та пошуку інформаційних джерел у середовищі Інтернет. Пропонуються такі питання для самостійного</p>	11

	<p>опрацювання:</p> <ol style="list-style-type: none"> 1. Dockerfile. Побудова Docker-образу. 2. Інструмент Docker Compose. 3. Використання оркестратора Docker Compose для запуску багатоконтейнерних додатків. <p>Самостійна робота студентів перевіряється з допомогою опитувань, перевірки виконання лабораторних робіт та написання рефератів на питання винесених на самостійне опрацювання.</p> <p style="text-align: center;">Список рекомендованих джерел</p> <p style="text-align: center;"><i>Основний: 2[с.120-142]</i></p> <p style="text-align: center;"><i>Додатковий: 2 [с.60-82]</i></p> <p style="text-align: center;"><i>Інтернет-ресурси: 4,6</i></p>	
	<p>Лабораторне заняття №11. Тема «Контейнеризація»</p> <p><i>Завдання до заняття: створити файл конфігурації Docker Compose для розгортання та керування декількома сервісами.</i></p> <ol style="list-style-type: none"> 1. <i>Визначення сервісів: обрати декілька сервісів, які можна запустити в контейнерах, наприклад, веб-додаток, базу даних та кеш-сервер.</i> 2. <i>Написання Dockerfile: для кожного обраного сервісу написати Dockerfile, щоб можна було створити відповідний контейнер.</i> 3. <i>Створення конфігурації Docker Compose: створити файл docker-compose.yml, визначити всі сервіси, їх залежності, мережі, обсяги даних.</i> 4. <i>Запуск та тестування: запустити контейнери за допомогою docker-compose up та переконатись, що всі сервіси працюють коректно.</i> 5. <i>Конфігурація середовища: задати параметри середовища для кожного сервісу через docker-compose.yml та змінити їх без перезапуску контейнера.</i> 6. <i>Моніторинг та логування: додати конфігурацію для збору логів та метрик зі своїх контейнерів.</i> 7. <i>Підготувати звіт про виконання лабораторної роботи.</i> 	2
<p>Знати: особливості системи керування версіями</p>	<p>Тема 12. Система керування версіями. Лекція № 12. Система керування версіями. План лекції:</p>	15

<p>Вміти: використовувати отримані теоретичні знання про керування версіями в практичній діяльності.</p>	<ol style="list-style-type: none"> 1. Базові поняття. Створення репозиторію. 2. Операції Git при внесенні змін. 3. Робота з гілками. Запити на злиття гілки. Видалення гілки. Коміти злиття. 4. Конфлікти злиття та їх вирішення. 5. Виключення відстеження файлів. <p>Список рекомендованих джерел: <i>Основний: 2 [с.145-170]</i> <i>Додатковий: 2 [с.60-82]</i> <i>Інтернет-ресурси: 4,7</i></p>	<p>2</p>
	<p style="text-align: center;">Самостійна робота</p> <p>Вивчення матеріалу до теми «Система керування версіями», підготовка до лабораторного заняття.</p> <p>Самостійна робота передбачає вивчення окремих питань дисципліни на основі опрацювання літератури та пошуку інформаційних джерел у середовищі Інтернет. Пропонуються такі питання для самостійного опрацювання:</p> <ol style="list-style-type: none"> 1. Збереження незавершених змін (git stash). 2. Навігація по історії. 3. Скасування або внесення змін до вже зроблених комітів. 4. Злиття гілок (у власну гілку). <p>Самостійна робота студентів перевіряється з допомогою опитувань, перевірки виконання лабораторних робіт та написання рефератів на питання винесених на самостійне опрацювання.</p> <p style="text-align: center;">Список рекомендованих джерел <i>Основний: 2 [с.145-170]</i> <i>Додатковий: 2 [с.60-82]</i> <i>Інтернет-ресурси: 4,7</i></p>	<p>11</p>
	<p>Лабораторне заняття №12. Тема: «Система керування версіями»</p> <p><i>Завдання до заняття: використовуючи Git, створити та об'єднати гілки для розробки функціоналу та вирішення конфліктів під час об'єднання.</i></p> <ol style="list-style-type: none"> 1. Створити новий репозиторій на сервісі хостингу (наприклад, GitHub або GitLab). 2. Клонувати репозиторій на локальний комп'ютер. 3. Створити дві гілки - одну для розробки 	<p>2</p>

	<p><i>нового функціоналу (наприклад, feature/awesome-feature) та іншу для виправлення помилок (наприклад, bugfix/fix-bug).</i></p> <p>4. <i>Внести зміни у кожен з гілок - додати файли, внести зміни в існуючі файли.</i></p> <p>5. <i>Змінити один і той самий рядок або регіон коду в обох гілках, щоб викликати конфлікт.</i></p> <p>6. <i>Об'єднати гілки, вирішивши конфлікти, застосовуючи зміни з обох гілок.</i></p> <p>7. <i>Відправити зміни до віддаленого репозиторію та створити пулл-реквест для об'єднання робочої гілки з головною гілкою.</i></p> <p>8. <i>Підготувати звіт про виконання лабораторної роботи.</i></p>	
	Разом	180

* +20% інтерактиву – зазначені курсивом

3. СПИСОК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ

Основний:

1. Брюханова Г. В. Комп'ютерні дизайн-технології: навчальний посібник для студентів вищих навчальних закладів / Г.В. Брюханова. – К.: Центр учбової літератури, 2018. – 180 с.
2. Методи та новітні підходи до проектування, управління і застосування високопродуктивних ІТ-інфраструктур: монографія / Ю. В. Бойко, В. М. Волохов, М. М. Глибовець, С. В. Єршов, С. Л. Кривий, С. Д. Погорілий, О. І. Ролік, С. Ф. Теленик, М. В. Ясочка // за ред. проф. А. В. Анісімова. – К.: ВПЦ «Київський університет», 2016. – 447 с.

Додатковий:

1. Воробієнко П.П., Нікітюк Л.А., П.І. Резніченко. Телекомунікаційні та інформаційні мережі: Підручник / П.П. Воробієнко, Л.А. Нікітюк, П.І. Резніченко. - К.: САММІТ-Книга, 2010. – 708 с.
2. Анісімов А.В. Інформаційні системи та бази даних: Навчальний посібник для студентів факультету комп'ютерних наук та кібернетики. / Анісімов А.В., Кулябко П.П. – Київ. – 2017. – 110 с.
3. Ford R. Web Design. The Evolution of the Digital World 1990 –Today (multilingual Edition) (MI: MIDI) / TASCHEM; Multilingual edition (December 7, 2019) – 640 p

Інтернет-ресурси:

1. Організація комп'ютерних мереж [Електронний ресурс] : підручник: для студ. спеціальності 121 «Інженерія програмного забезпечення» та 122 «Комп'ютерні науки» / КПІ ім. Ігоря Сікорського; Ю. А. Тарнавський, І. М. Кузьменко. – Електронні текстові дані. – Київ : КПІ ім. Ігоря Сікорського, 2018. – 259 с

2. FigmaHelpCenter. Режим доступу:<https://help.figma.com/hc/en-us>
3. Material design palette– Режим доступу:<https://www.materialpalette.com>.
4. <https://refactoring.guru/uk/design-patterns/catalog>
5. <https://kubernetes.io/>
6. <https://www.docker.com/>
7. <https://git-scm.com/>

**- Курсивом виділені джерела, що є в бібліотеці ДТЕУ*