МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

КРЕМЕНЧУЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ІМЕНІ МИХАЙЛА ОСТРОГРАДСЬКОГО

Навчально-науковий інститут електричної інженерії

та інформаційних технологій

КАФЕДРА АВТОМАТИЗАЦІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

ЗВІТ

З ЛАБОРАТОРНИХ РОБІТ

З НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

«Frontend-розробка»

Виконав студент групи КН-23-1

Полинько Ігор Миколайович

Перевірив старший викладач кафедри АІС Притчин О. С.

КРЕМЕНЧУК 2025

**Лабораторна робота № 2**

**Тема:** Робота з компонентом React з використанням JSX та стилів.

**Мета:** Познайомитись і отримати головні навички роботи з компонентом React з використанням JSX та стилів.

**Виконання завдання лабораторної роботи:**

**Завдання 1.** Створіть файл для нового компоненту в папці components. Памʼятайте, що бажано аби він мав розширення .jsx.

Створіть ваш компонент. Структура може бути довільною, але він повинен мати щонайменше 2-3 розділювача, кілька будь-яких заголовків h, кілька параграфів p та один чи кілька списків з типом на вибір. Також можна додати будь-які html теги також на вибір.

Кнопки та інші теги можна поки не додавати. React має свій особливий механізм для надання сторінці інтерактивності. Ми поговоримо про це у наступній лабораторній роботі.

Задайте тегам стилі аби вони мали якийсь відступ від країв сторінки та були впорядкованими і сторінка мала гарний вигляд.

Для цього можна створити однойменний файл в папці components з розширенням .css. Або такий же файл в окремій папці styles користуючись принципами описаними вище.

Для імпортування файлу з іншої папки треба прописати шлях з .. в імпорті. Кожне .. підіймає шлях на папку в гору. Таким чином наприклад, щоб вказати шлях до папки styles з папки components на треба такий шлях

import '../styles/myComponent.css';

Тут ми, спочатку, підіймаємось в папку src через .. і потім вказуємо вже шлях до файлу зі стилями з папки src.

В файлі зі стилями має бути щонайменше 3-5 стилів з 2-3 різними селекторами.

Після завершення написання компоненту зі стилями поверніться до файлів App.jsx та App.css.

Спочатку відредагуйте стилі в файлі App.css щоб div з класом App займав всю висоту та ширину сторінки.

Далі, в файлі App.jsx імпортуйте ваш компонент та вставте його в головний div.

Збережіть всі файли та запустіть проєкт.

Перевірте результат та виправте стилі, структуру та код компонента якщо в цьому буде потреба.

**App.jsx:**

import "./App.css";

import MyInfo from "./components/MyInfo";

function App() {

  return (

    <div className="App">

      <MyInfo />

    </div>

  );

}

export default App;

**MyInfo.jsx:**

function MyInfo() {

  return (

    <div className="my-info">

      <h1>Привіт, я Ігор!</h1>

      <p>Це перший параграф у власному компоненті.</p>

      <p>А це другий параграф.</p>

      <h2>Мої улюблені речі:</h2>

      <ul>

        <li>Фотографія</li>

        <li>Ігри</li>

        <li>Formula 1</li>

      </ul>

      <h2>План на день:</h2>

      <ol>

        <li>Намагатися прокинутися</li>

        <li>Зробити лабораторну роботу по React</li>

        <li>Зробити перерву</li>

      </ol>

    </div>

  );

}

export default MyInfo;

**MyInfo.css:**

.my-info {

  margin: 20px;

  padding: 15px;

  border: 2px solid #4caf50;

  border-radius: 10px;

  background-color: #f9f9f9;

}

.my-info h1 {

  color: #2e7d32;

  margin-bottom: 10px;

}

.my-info h2 {

  color: #388e3c;

  margin-top: 15px;

}

.my-info p {

  margin: 8px 0;

  font-size: 16px;

  line-height: 1.4;

}

.my-info ul,

.my-info ol {

  margin-left: 20px;

}

**App.css:**

.App {

  width: 100%;

  height: 100vh;

  display: flex;

  justify-content: center;

  align-items: flex-start;

  background-color: #eef2f3;

}

.App-logo {

  height: 40vmin;

  pointer-events: none;

}

@media (prefers-reduced-motion: no-preference) {

  .App-logo {

    animation: App-logo-spin infinite 20s linear;

  }

}

.App-header {

  background-color: #282c34;

  min-height: 100vh;

  display: flex;

  flex-direction: column;

  align-items: center;

  justify-content: center;

  font-size: calc(10px + 2vmin);

  color: white;

}

.App-link {

  color: #61dafb;

}

@keyframes App-logo-spin {

  from {

    transform: rotate(0deg);

  }

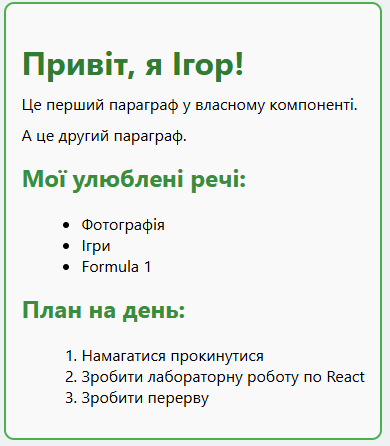
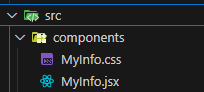
  to {

    transform: rotate(360deg);

  }

}

**Результат:**

** **

**Висновок:** на цій лабораторній роботі ми познайомились і отримали головні навички роботи з компонентом React з використанням JSX та стилів.

**Контрольні питання:**

1. **Що таке JSX?**

JSX (JavaScript XML) — це синтаксичне розширення JavaScript, яке дозволяє писати HTML-подібний код прямо в JS. Потім він компілюється у звичайні виклики React.createElement.

1. **Як написати багатостроковий JSX?**

Обгорнути в дужки і переносити рядки:

return (

<div>

<h1>Привіт</h1>

<p>Це багатостроковий JSX</p>

</div>

);

1. **Чи можна в JSX вставити JavaScript вираз?**

Так, можна.

1. **Як вставити в JSX JavaScript вираз?**

Через фігурні дужки { }:

const name = "Ігор";

return <h1>Привіт, {name}!</h1>;

1. **Що таке умовний рендеринг?**

Це можливість рендерити різні елементи залежно від умови. Наприклад:

{isLoggedIn ? <p>Вітаю!</p> : <p>Будь ласка, увійдіть</p>}

1. **Як підключити файл зі стилями до файлу з React компонентом?**

Просто імпортувати CSS у компонент:

import './App.css';

1. **Які види експорту/імпорту ви знаєте?**

* Експорт за замовчуванням (default export)
* Іменований експорт (named export)
* Збірний імпорт (import \* as …).

1. **Що таке експорт за умовчанням?**

Це коли з файлу можна експортувати тільки одну основну сутність:

export default App;

І під час імпорту можна дати будь-яке ім’я:

import MyApp from './App';

1. **Які види іменованого експорту ви знаєте?**

* Окремий:

export const Button = () => {};

* Груповий:

const Input = () => {};

const Label = () => {};

export { Input, Label };

**10. Які види імпорту ви знаєте?**

* Імпорт за замовчуванням:

import App from './App';

* Іменований імпорт:

import { Button } from './UI';

* Збірний імпорт:

import \* as UI from './UI';

1. **Що таке збірний імпорт?**

Це коли всі експорти з файлу імпортуються під одним простором імен:

import \* as Utils from './utils';

Utils.formatDate();

1. **Як вставити свій компонент в JSX іншого компонента?**

function Header() {

return <h1>Заголовок</h1>;

}

function App() {

return (

<div>

<Header />

</div>

);

}