**useContext – React**

ה-useContext hook הוא פונקציה ב-React המאפשרת לנו לגשת לנתונים מקומפוננטות Child מבלי להעביר Props בכל רמה של עץ הקומפוננטות.

useContext מספק דרך טובה ונוחה לנהל state בקומפוננטות נפרדות ללא צורך בהעברת Props או בספריות שונות לניהול State.

1. נתחיל במעבר לקומפוננטה ממנה נרצה למשוך את הstate ונעשה import לcreateContext בתוך React:
2. import React, { FC, createContext } from "react";
3. כעת ניצור את הstate שנרצה להעביר וניצור context חדש כך:
4. import React, { FC, createContext } from "react";
5. import styles from "./App.module.scss";
6. import pizzas from "../../data/pizzas.json";
7. import Pizza from "../Pizza/Pizza";
8. import Cart from "../ShoppingCart/ShoppingCart";
9. interface AppProps {}
10. const pizza = [
11. {
12. "id": 1,
13. "name": "Napolitana",
14. "description": "Fresh tomatoes and mozzarella cheese.",
15. "price": 12.99
16. },
17. {
18. "id": 2,
19. "name": "Marinara",
20. "description": "Fresh tomatoes, extra virgin olive oil, oregano and garlic.",
21. "price": 15.99
22. },
23. {
24. "id": 3,
25. "name": "Caprese",
26. "description": "Fresh mozzarella, tomatoes, basil and balsamic vinegar.",
27. "price": 12.99
28. },
29. {
30. "id": 4,
31. "name": "Margherita",
32. "description": "Fresh tomatoes, mozzarella cheese, fresh basil, salt and extra-virgin olive oil.",
33. "price": 17.99
34. }
35. ]
36. export const PizzaContext = createContext<{id:number; name:string; description:string; price:number}[]>(pizza);
37. const App: FC<AppProps> = () => {
38. return (
39. <ul className={styles.App}>
40. <h1>Delicious Pizza</h1>
41. <Cart />
42. {pizzas.map((pizza) => {
43. return <Pizza key={pizza.id} pizza={pizza} />;
44. })}
45. </ul>
46. );
47. };
48. export default App;
49. כעת, על מנת להעביר את הstate הרצוי לקומפוננטה הרצויה, עלינו להצהיר עליה כchild לcontext. זאת אומרת, אילו קומפוננטות יקבלו גישה למידע שאנחנו מעבירים.

נעשה זאת ע"י כך שנעטוף את הקריאה לקומפוננטה בתוך הJsx ב - contextName.Provider.

ונעביר את המשתנה שמחזיק בstate שנרצה להעביר בתוך attribute בשם value (במקרה הזה – pizza).

const App: FC<AppProps> = () => {

  return (

    <ul className={styles.App}>

      <h1>Delicious Pizza</h1>

      <PizzaContext.Provider value={pizza}>

      <Cart />

      </PizzaContext.Provider>

      {pizzas.map((pizza) => {

        return <Pizza key={pizza.id} pizza={pizza} />;

      })}

    </ul>

  );

};

export default App;

\*הערה – יש לעטוף את החלק הקטן ביותר שנוכל על מנת קבל תפקוד תקין של האפליקציה ולהמנע מרינדורים מיותרים בהמשך. אם נעטוף יותר ממה שצריך נקבל התנהות לא צפויה של האפליקציה ובמקרים מסויימים קריסה.

1. כעת, לאחר שעטפנו את החלק שרצינו להעביר אליו את הstate, נעבור לקומפוננטה המקבלת ונעשה import ל useContext. בנוסף, ניצור משתנה ונמשוך את הstate לתוכו כך:

import React, { FC, useState, useContext, Fragment } from "react";

import { PizzaContext } from "../App/App";

import styles from "./ShoppingCart.module.scss";

interface CartProps {}

interface Flag {

  isOpen: boolean;

}

const Cart: React.FC<CartProps> = () => {

  const [isOpen, setIsOpen] = useState(false);

  const handleClick = (e: React.MouseEvent<HTMLButtonElement>): void => {

    setIsOpen((prevState) => !prevState);

  };

  const pizzas = useContext(PizzaContext);

  const cartSum = pizzas

    .reduce((sum, pizza) => sum + pizza.price, 0)

    .toFixed(2);

  return (

    <div className={styles.ShoppingCart}>

      <button onClick={handleClick}>{pizzas.length} pizza(s)</button>

      <div

        style={{ display: isOpen ? "flex" : "none" }}

        className={styles.listContainer}

      >

        <ul className={styles.cartList}>

          <h4>My Cart ({pizzas.length})</h4>

          <div className={styles.listItems}>

            {pizzas.map((pizza) => {

              return (

                <li key={pizza.id} className={styles.listItem}>

                  <div className={styles.cartPizzaName}>{pizza.name}</div>

                  <div className={styles.cartPizzaName}>{pizza.price}$</div>

                </li>

              );

            })}

          </div>

        </ul>

        <div className={styles.sumContainer}>

          <hr />

          <div className={styles.sumPrice}>

            <h3>Cart Subtotal: </h3>

            <h3>{cartSum}$</h3>

          </div>

        </div>

      </div>

    </div>

  );

};

export default Cart;

1. כעת המשתנה "pizzas"שיצרנו מכיל את הstate של pizza שהעברנו מהקומפוננטה התורמת ונוכל לעשות בו שימוש.

בדוגמא הזו בחרתי למשוך את המידע הרלוונטי על מנת ליצור עגלת קניות שמקבלת ובה פונקציית map שמייצרת <li> שמכיל את המידע הרצוי. שימו לב שיצרתי משתנה שמחזיק את הסכום הכולל של עגלת הקניות ע"י reduce מJavaScript. כמו כן, השתמשתי באורך המערך על מנת לקבוע כמה מוצרים יש לי בעגלה באופן דינמי.

const Cart: React.FC<CartProps> = () => {

  const [isOpen, setIsOpen] = useState(false);

  const handleClick = (e: React.MouseEvent<HTMLButtonElement>): void => {

    setIsOpen((prevState) => !prevState);

  };

  const pizzas = useContext(PizzaContext);

  const cartSum = pizzas

    .reduce((sum, pizza) => sum + pizza.price, 0)

    .toFixed(2);

  return (

    <div className={styles.ShoppingCart}>

      <button onClick={handleClick}>{pizzas.length} pizza(s)</button>

      <div

        style={{ display: isOpen ? "flex" : "none" }}

        className={styles.listContainer}

      >

        <ul className={styles.cartList}>

          <h4>My Cart ({pizzas.length})</h4>

          <div className={styles.listItems}>

            {pizzas.map((pizza) => {

              return (

                <li key={pizza.id} className={styles.listItem}>

                  <div className={styles.cartPizzaName}>{pizza.name}</div>

                  <div className={styles.cartPizzaName}>{pizza.price}$</div>

                </li>

              );

            })}

          </div>

        </ul>

        <div className={styles.sumContainer}>

          <hr />

          <div className={styles.sumPrice}>

            <h3>Cart Subtotal: </h3>

            <h3>{cartSum}$</h3>

          </div>

        </div>

      </div>

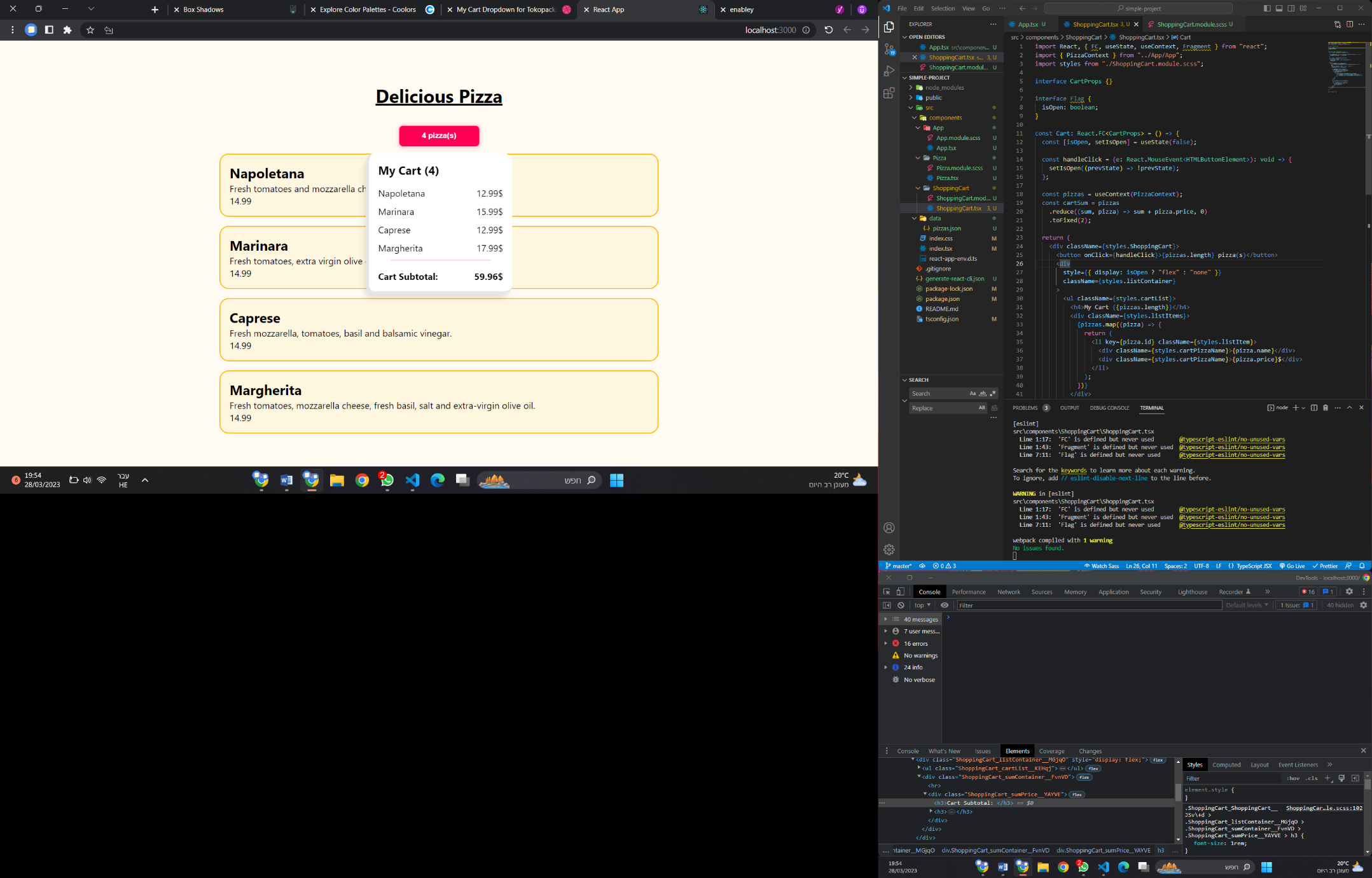
    </div>

  );

};

**זהו**

יצרנו context בקומפוננטה תורמת ומשכנו בקומפוננטה מקבלת. ולאחר שקיבלנו את הערכים עשינו בהם שימוש ללא צורך בהעברת props לכל ילד בעץ הקומפוננטות וללא רינדורים מיותרים. האפליקציה יעילה ונוחה לפיתוח ושינויים.

זוהי התוצאה (כל הנתונים בתוך MyCart הועברו באופן דינמי) :