React + Ts

HOCS, Route Guard

Wrapper Components/HOCS

```
type TitleProps = {
  children: React.ReactNode;
};

const Title = (props: TitleProps) => {
  const { children } = props;

  return (
    <div className="w-full text-center">
      <h1 className="m-5 text-3xl font-bold text-amber-500">{children}</h1>
    </div>
  );
};
export default Title;
```

- מה הוא ה-Design Pattern שנקרא Higher Order Component?
- איך נוכל להיעזר ב-pattern זה בתוכנית ריאקט?
- מדוע זה נקרא גם Wrapper Component?
- במה זה יכול להקל עלינו?
- מה המשמעות של children בהקשר זה?
- מה ה-type שנרצה לתת ל-children?
- כיצד נגדיר לקומפוננטה שתדרוש children?
- כיצד נוכל להגדיר שזה לא יהיה חובה (אופציונלי)?
- כיצד נעביר children לקומפוננטה?
- מה עוד נוכל לדרוש ב-props מעבר ל-children?

Wrapper Components/HOCS – תרגול – חלק א'

- צרו פרויקט react-flowbite חדש עבור הנושא.
- צרו קומפוננטה חדשה בשם Title.
- הקומפוננטה תדרוש children ב-props.
- הקומפוננטה תחזיר div בעל עיצוב שמתאים לכותרת.
- ה-div יכיל תגית h1 שמכילה את ה-children אשר נשלחו לקומפוננטה.
- צרו קומפוננטה חדשה בשם Text.
- הקומפוננטה תדרוש children ב-props.
- הקומפוננטה תחזיר div בעל עיצוב שמתאים לפיסקה.
- ה-div יכיל תגית p שמכילה את ה-children אשר נשלחו לקומפוננטה.
- צרו מאמר בעל 3 פסקאות.
- את המלל של הכותרת העבירו כ-children לקומפוננטת ה-Title.
- את המלל של הפיסקה העבירו כ-children לקומפוננטת ה-Text.

Title 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam nec turpis et elit tincidunt vestibulum. Donec nec nunc nec elit tincidunt vestibulum. Nullam nec turpis et elit tincidunt vestibulum. Donec nec nunc nec elit tincidunt vestibulum.

Title 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam nec turpis et elit tincidunt vestibulum. Donec nec nunc nec elit tincidunt vestibulum. Nullam nec turpis et elit tincidunt vestibulum. Donec nec nunc nec elit tincidunt vestibulum.

Title 3

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam nec turpis et elit tincidunt vestibulum. Donec nec nunc nec elit tincidunt vestibulum. Nullam nec turpis et elit tincidunt vestibulum. Donec nec nunc nec elit tincidunt vestibulum.

Wrapper Components/HOCS – תרגול – חלק ב'

- צרו קומפוננטה חדשה בשם Flex.
- צרו enum עבור כיוון ה-flex בשם Directions (“row”, “col”).
- צרו enum עבור סידור ה-flex בשם Aligns (“start”, “between”, “center” וכו').
- הקומפוננטה תדרוש children ב-props.
- הקומפוננטה תדרוש גם את ה-props -
  - * direction: Directions
  - * justify: Aligns
  - * items: Aligns
  - * wrap?: boolean
  - * classname?: string
- הקומפוננטה תחזיר תגית div בעלת עיצוב תואם למה שנתקבל ב-props, ה-div יציג את ה-children.
- החליפו את כל האלמנטים מסוג div בתוכנית בקומפוננטת ה-Flex שיצרתם.

```
export enum Directions {
    ROW = "row",
    COL = "col",
};
```

```
export enum Aligns {
    START = "start",
    CENTER = "center",
    END = "end",
    BETWEEN = "between",
    AROUND = "around",
    EVENLY = "evenly",
};
```

```
type FlexProps = {
  children?: React.ReactNode;
  direction: Directions;
  justify: Aligns;
  items: Aligns;
  wrap?: boolean;
  className?: string;
};

const Flex = (props: FlexProps) => {
  const { children, direction, justify, items, wrap, className } = props;

  return (
    <div className={`flex flex-${direction} ${wrap && "flex-wrap"} justify-${justify} items-${items} ${className}`}>
      {children}
    </div>
  );
};
export default Flex;
```

Route Guard HOC

```react-router-dom
import { Link } from "react-router-dom";

type RouteGuardProps = {
  children?: React.ReactNode;
  isLogged: boolean;
};

const RouteGuard = (props: RouteGuardProps) => {
  const { children, isLogged } = props;

  return isLogged ? (
    children
  ) : (
    <div className="w-full text-center">
      <h1 className="m-5 text-3xl font-bold text-red-500">
        You are not logged in!
      </h1>
      <Link to="/home" className="text-blue-500">
        Go to Home
      </Link>
    </div>
  );
};

export default RouteGuard;
```

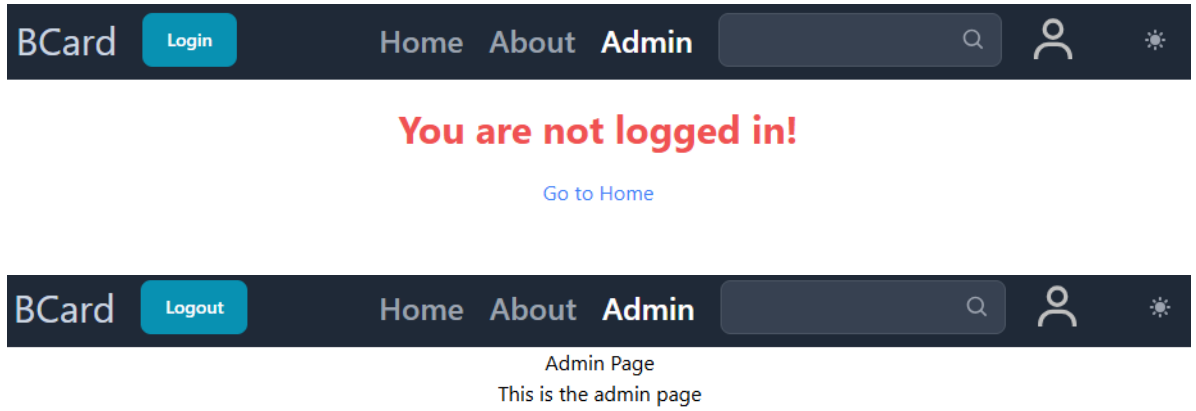
- איזה עוד שימושים יש ל-HOCS מלבד עיצוביים?
- מה הוא Route Guard?
- מדוע שנרצה להגן על נתיבים?
- כיצד נוכל להשתמש ב-HOC על מנת ליצור Route Guard?
- מה נרצה לדרוש ב-props של ה-Route Guard?
- כיצד נוכל לנווט את המשתמש בחזרה לדף הבית במידה והוא לא מורשה לצפות בנתיב מסוים?

You are not logged in!

Go to Home

תרגול - Route Guard

- צרו פרויקט react-flowbite חדש עבור הנושא.
- צרו שני עמודים: Home ו-Admin.
- צרו Navbar תואם בתוך קומפוננטת Header חדשה.
- הוסיפו כפתור login ל-Navbar.
- צרו בקומפוננטה הראשית (App) state בוליאני חדש בשם isLoggedIn, בתור ערך התחלתי תנו לו – false.
- העבירו ל-Header דרך ה-props את ערך ה-state ופונקציית העדכון שלו.
- במידה והמשתמש מחובר (isLoggedIn=true) הכיתוב על הכפתור יהיה logoff.
- כאשר המשתמש לוחץ על כפתור ה-login/logoff ייעשה שימוש בפונקציית העדכון של ה-state על מנת לשנות את isLoggedIn ל-true/false בהתאם.
- צרו RouteGuard אשר עוטף את Admin.
- העבירו לו ב-props את הערך של ה-state.
- התוכן של העמוד Admin יוצג רק למשתמש מחובר, אחרת יועבר בחזרה לדף הבית.




```
<BrowserRouter>
  <Header isLoggedIn={isLoggedIn} toggle={toggle} />

  <Routes>
    {/* Add routes here, the first route is the default route */}
    <Route path="/" element={<Home />} />
    <Route path="/home" element={<Home />} />
    <Route path="/about" element={<About />} />
    <Route
      path="/admin"
      element={
        <RouteGuard isLogged={isLoggedIn}>
          <Admin />
        </RouteGuard>
      }
    />

    <Route path="/*" element={<Error />} />
  </Routes>

</BrowserRouter>
```

שיעורי בית

- פיתחו את פרויקט סוף המודול שלכם.
- צרו wrapper בשם Flex והשתמשו בו בפרויקט שלכם במקום ב-div בכל מקום שיש צורך.

תודה על ההקשבה

אני וצוות המכללה כאן עבורכם