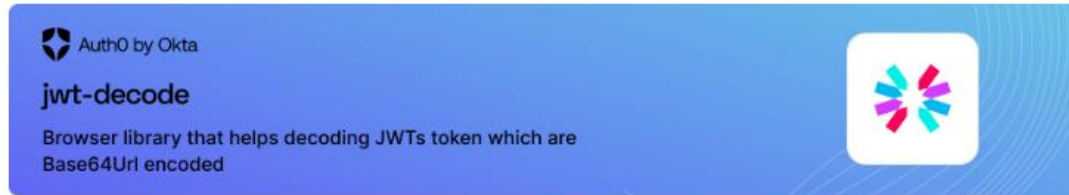


# React + Ts

JWT-Decoder, Route Guard,  
Query Params

04/09/2024

# Jwt-decode



Install

```
> npm i jwt-decode
```

```
import { jwtDecode } from "jwt-decode";

export type DecodedToken = {
  _id: string;
  isBusiness: boolean;
  isAdmin: boolean;
};

export const decode = (token: string) => {
  return jwtDecode(token) as DecodedToken;
};
```

- לשם מה נשתמש בספרייה jwt-decode?
- ממה מורכב JWT-Token?
- מדוע שנרצה לחלץ מידע מתוך token?
- באיזה פונקציה נשתמש על מנת לעשות זאת?
- מדוע שנרצה ליצור type עבור סוג ההחזר של הפונקציה jwtDecode?

# jwt-decode – תרגול חלק א'

▪ פיתחו את הפרויקט של השיעור הקודם.

\* טפלו ב-slice של user כך :

- צרו type עבור user הכולל את כל פרטיו כפי שמתקבלים מה-API בעת שליחת בקשה לקבלת פרטי משתמש.
- ה-state יכיל אובייקט של user במקום שם משתמש.
- טפלו בפונקציות העדכון בהתאם כפי שמופיע בנספח.

```
import { createSlice, PayloadAction } from "@reduxjs/toolkit";

const initialState = {
  isLoggedIn: false,
  user: null as TUser | null,
};

const UserSlice = createSlice({
  name: "user",
  initialState,
  reducers: {
    login: (state: TUserState, data: PayloadAction<TUser>) => {
      state.isLoggedIn = true;
      state.user = data.payload;
    },
    logout: (state: TUserState) => {
      state.isLoggedIn = false;
      state.user = null;
    }
  }
});

export const userActions = UserSlice.actions;
export type TUserState = typeof initialState;
export default UserSlice.reducer;
```

```
export type TUser = {
  _id: string,
  name: {
    first: string,
    middle: string,
    last: string,
  },
  phone: string,
  email: string,
  image: {
    url: string,
    alt: string,
  },
  address: {
    state: string,
    country: string,
    city: string,
    street: string,
    houseNumber: number,
    zip: number,
  },
  isAdmin: boolean,
  isBusiness: boolean,
}
```

# jwt-decode – תרגול חלק ב'

```
const submit = async (form: typeof initialFormData) => {
  try {
    // firstly, we will send a post request to the login endpoint with the form data
    const login = await axios.post(
      "https://monkfish-app-z9uza.ondigitalocean.app/bcard2/users/login",
      form,
    );

    // if the login is successful, we will get a token in the response & we will store it
    const token = login.data;
    localStorage.setItem("token", token);

    // we will set the token in the axios headers
    axios.defaults.headers.common["x-auth-token"] = token;

    // then we will get the user data using the id from the decoded token
    const user = await axios.get(
      "https://monkfish-app-z9uza.ondigitalocean.app/bcard2/users/" +
      decode(token)._id,
    );

    // we will dispatch the user data to the redux store
    dispatch(userActions.login(user.data));
    toast.success("Logged in successfully");
  } catch (error) {
    toast.error("Invalid credentials");
  }
};
```

- צרו helper ל-jwt-decoder בהתאם לדוגמא הקודמת.

- \* טפלו בפונקציית שליחת טופס ההתחברות כך:

- בעת התחברות מוצלחת ה-token ישמר ב-local storage.
- לאחר מכן הוסיפו אותו ל-headers ב-axios תחת המפתח "x-auth-token".
- חלצו את ה-id של המשתמש באמצעות שימוש ב-jwt-decode.
- בצעו קריאת API נוספת לשרת על מנת לייבא את פרטי המשתמש, תצטרכו לשם כך להוסיף את ה-id של המשתמש בסוף כתובת ה-URL.
- עדכנו את ה-store עם פרטי המשתמש.

- \* טפלו בעמוד Profile כך שיוצג שם המשתמש האמיתי.

# Route Guard HOC

```
type RouteGuardProps = {
  children: React.ReactNode;
  bizOnly?: boolean;
  adminOnly?: boolean;
};

const RouteGuard = (props: RouteGuardProps) => {
  const { children, bizOnly, adminOnly } = props;
  const userState = useSelector(
    (state: TRootState) => state.userSlice,
  ) as TUserState;

  const user = userState.user!;

  if (!userState.isLoggedIn) {
    return <Navigate to="/signin" />;
  }
  if (bizOnly && !user.isBusiness) {
    return <Navigate to="/" />;
  }
  if (adminOnly && !user.isAdmin) {
    return <Navigate to="/" />;
  }

  return <>{children}</>;
};

export default RouteGuard;
```

- 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?

# תרגול - Route Guard

- צרו RouteGuard בתוכנית שלכם בהתאם לדוגמא הקודמת.
- ב-Route אשר מפנה ל-Profile, עיטפו את הקומפוננטה ב-RouteGuard כך שרק משתמש מחובר יוכל לצפות בעמוד הפרופיל.

```
<Routes>
  <Route path="/" element={<Home />} />
  <Route path="/signin" element={<SignIn />} />
  <Route
    path="/profile"
    element={
      <RouteGuard>
        <Profile />
      </RouteGuard>
    }
  />
</Routes>
```

# Query Params

```
<Route path="/card/:id" element={<CardDetails />} />
```

dog

dog



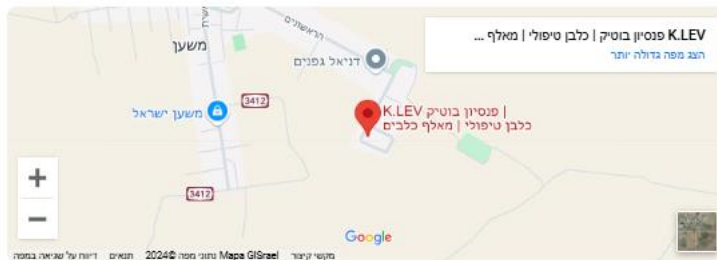
dog

Email: dog@gmail.com

Phone: <https://www.google.com>

Website: <https://www.google.com>

Address: dog 11, dog, dog



- מה הם Query Params?
- למה הם משמשים?
- כיצד נוכל להגדיר כתובת דינאמית לניתוב באפליקציה שלנו באמצעות שימוש ב-Query Params?
- מה התועלת שבדבר?

# Query Params – תרגול

```
<Route path="/card/:id" element={<CardDetails />} />
```

```
export type TCard = {
  _id: string;
  title: string;
  subtitle: string;
  description: string;
  phone: string;
  email: string;
  web: string;
  image: {
    url: string;
    alt: string;
  };
  address: {
    state: string;
    country: string;
    city: string;
    street: string;
    houseNumber: number;
    zip: number;
  };
  bizNumber: number;
  likes: string[];
  user_id: string;
};

const Home = () => {
  const [cards, setCards] = useState<TCard[]>();
  const nav = useNavigate();

  const navToCard = (e: React.MouseEvent<HTMLDivElement>) => {
    nav(`/card/${e.currentTarget.id}`);
  };

  useEffect(() => {
    (async () => {
      try {
        const res = await axios.get(
          "https://monkfish-app-z9uza.ondigitalocean.app/bcard2/cards",
        );
        setCards(res.data);
      } catch (error) {
        console.log("No Cards Found");
      }
    })();
  }, []);
};
```

- בעמוד הבית באפליקציה, יבאו והציגו את כל הכרטיסים מה-API בהתאם לנספח.
- צרו type עבור card הכולל את כל פרטיו כפי שמתקבלים מה-API בעת שליחת בקשה לקבלת פרטי כרטיס ביקור.
- צרו עמוד פרטי כרטיס בשם CardDetails וצרו עבורו נתיב דינאמי עבור הצגת פרטי כרטיס בהתאם לדוגמא שבשקף הקודם.
- טפלו בכרטיסים כך שבעת לחיצה יתבצע ניווט לעמוד הכרטיס עם ה-id התואם.

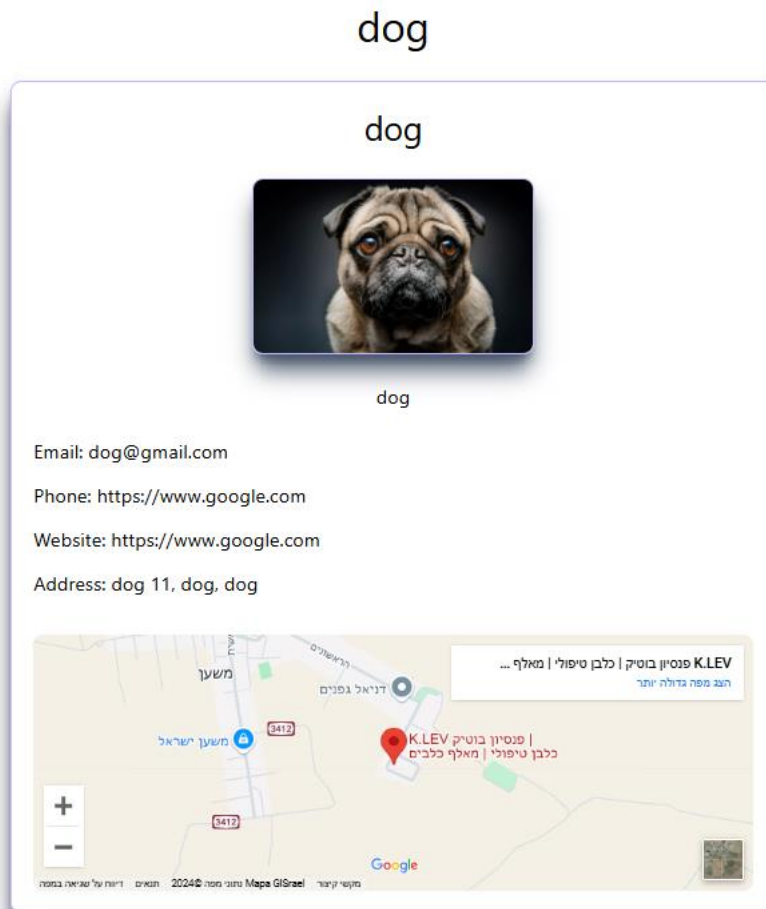
```
<div className="mt-10 flex flex-wrap justify-center gap-4">
  {cards &&
    cards.map((card) => (
      <Card
        key={card._id}
        id={card._id}
        className="h-[500px] w-1/4"
        onClick={navToCard}
      >
        <div className="flex h-full flex-col">
          <img
            src={card.image.url}
            alt={card.image.alt}
            className="h-[200px] object-cover"
          />
          <h1 className="text-center text-2xl">{card.title}</h1>
          <p className="mt-10 text-center">{card.description}</p>
        </div>
      </Card>
    ))
  }
</div>
```



# useParams

```
const { id } = useParams<{ id: string }>();
```

- לשם מה נועד ה-Hook המיוחד useParams אשר מגיע מהספרייה react-router-dom?
- כיצד נוכל להשתמש בו על מנת לייבא פרמטר ספציפי מהכתובת?
- איזה תועלת יש בדבר?



# useParams – תרגול

- בעמוד פרטי הכרטיס (CardDetails), ייבאו את ה-id של הכרטיס באמצעות שימוש ב-useParams.
- באמצעות שימוש ב-useEffect ו-useState, הציגו את פרטי הכרטיס המלאים, יש לבצע קריאת API לשם כך.
- במידה והתקבל error מהשרת, הציגו הודעה תואמת במקום פרטי הכרטיס בתוך תגית H1 ממורכזת.

```
const CardDetails = () => {
  const { id } = useParams<{ id: string }>();
  const [card, setCard] = useState<TCard>();
  const [error, setError] = useState<string>();

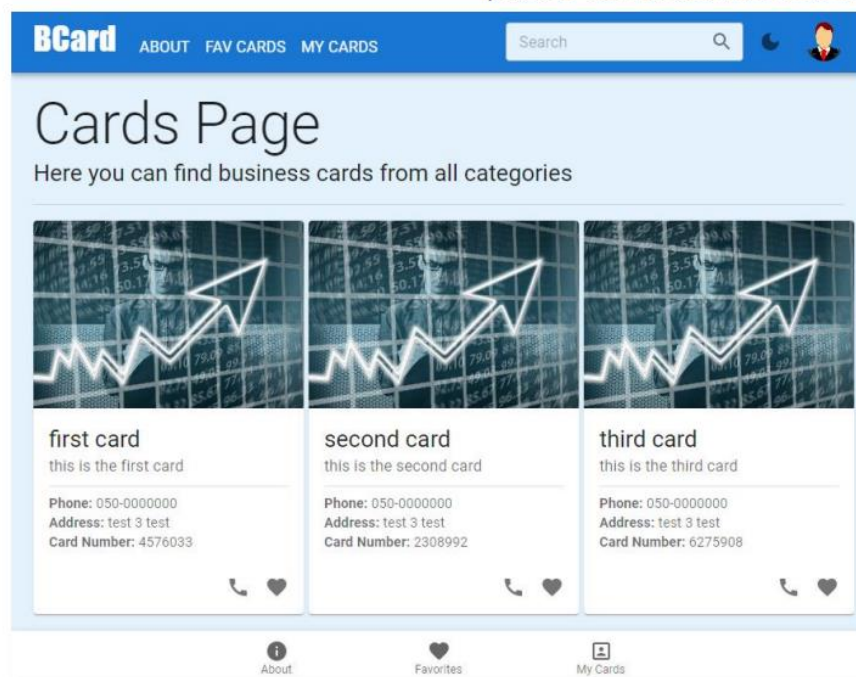
  useEffect(() => {
    (async () => {
      try {
        const res = await axios.get(
          `https://monkfish-app-z9uza.ondigitalocean.app/bcard2/cards/${id}`,
        );
        setCard(res.data);
      } catch (error) {
        setError(`Card with Id:${id} Not Found`);
      }
    })();
  }, [id]);

  return card ? (
    <Card title={card.title}>
      <img src={card.image.url} alt={card.image.alt} />
      <h2>{card.subtitle}</h2>
      <p>{card.description}</p>
      <p>{card.phone}</p>
      <p>{card.email}</p>
      <p>{card.web}</p>
      <p>
        {card.address.country}, {card.address.city}, {card.address.street}{" "}
        {card.address.houseNumber}
      </p>
    </Card>
  ) : (
    <h1>{error}</h1>
  );
};
```

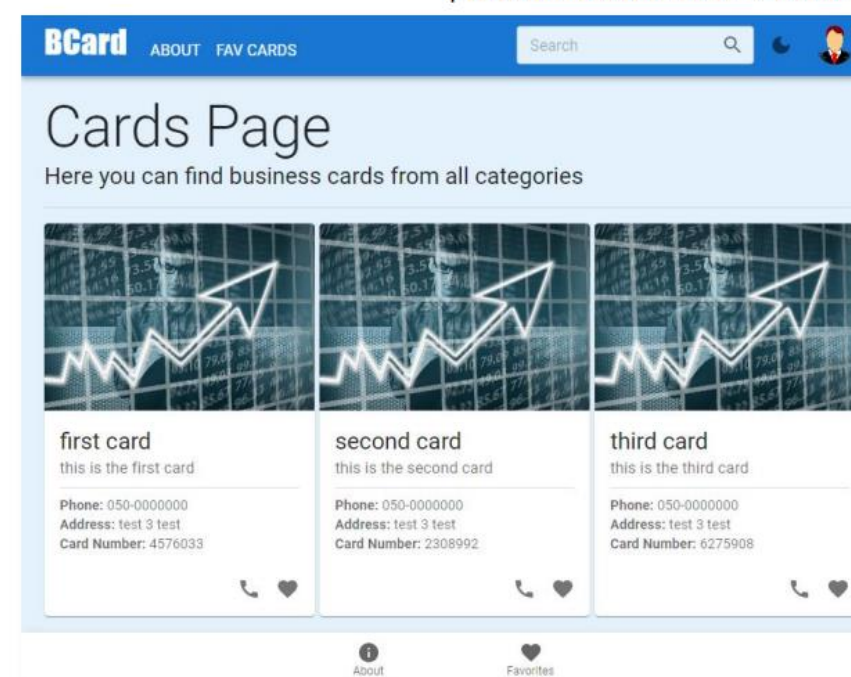
# שיעורי בית

- יש לחזור ולסיים את כל התרגילים שבמצגת.
- פיתחו את פרויקט סוף המודול שלכם.
- טפלו ויישמו את מה שלמדנו בשיעור:

תפריט ניווט טעם משתמש מחובר מסוג עסקי



תפריט ניווט עם משתמש מחובר שאינו עסקי



# תודה על ההקשבה

אני וצוות המכללה כאן עבורכם