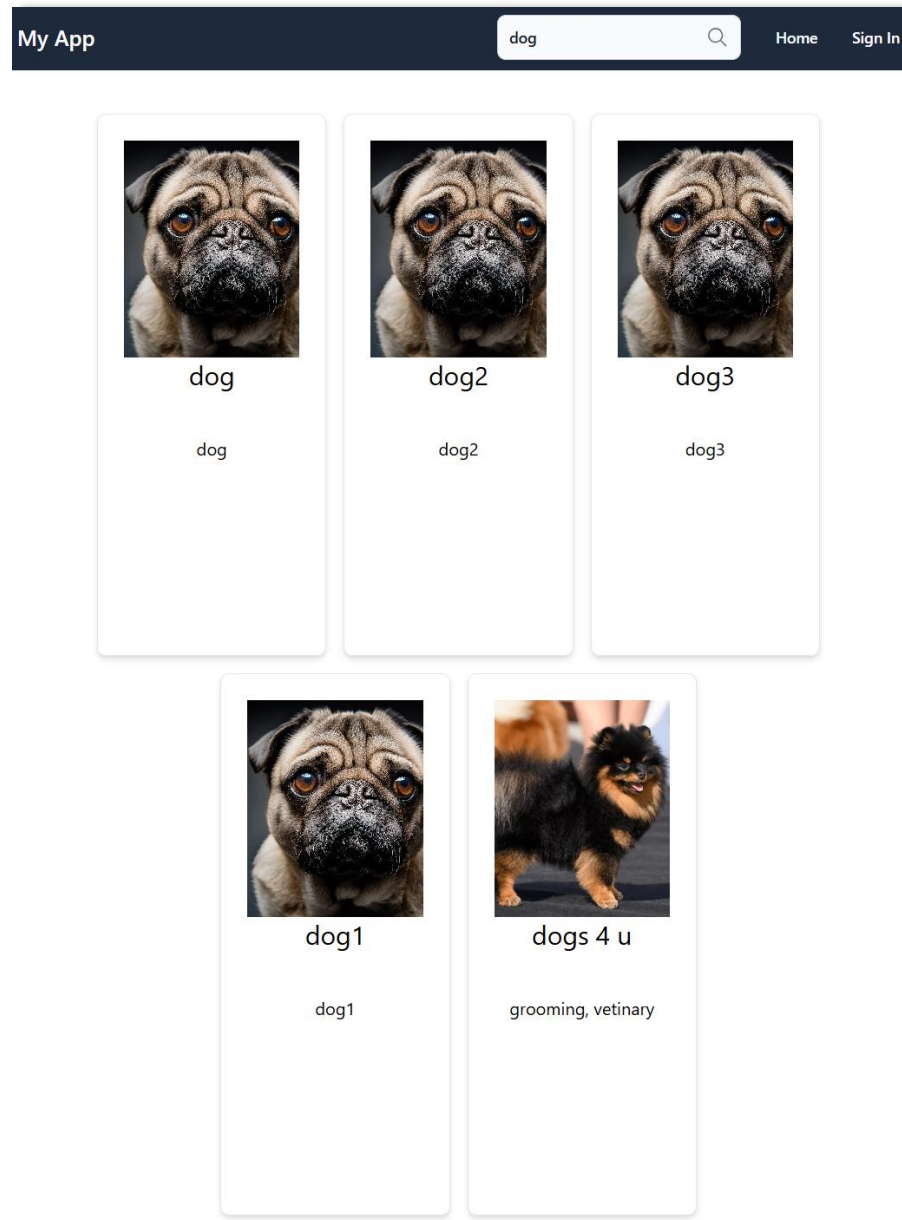


# React + Ts

Search, Favorites & Pagination



## Search

- כיצד נוכל ליצור תיבת חיפוש שהערך שמוקלד ל-input שלה יהיה זמין בכל חלקי האפליקציה בצורה גלובלית?
- לאיזה אירועים נוכל להאזין על מנת להזין את קלט המשתמש ל-state בעת הקלדה?

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

const initialState = {
  search: ""
}

const SearchSlice = createSlice({
  name: "search",
  initialState,
  reducers: {
    searchWord: (state: TSearchState, data: PayloadAction<string>) => {
      state.search = data.payload
    }
  }
});

export const searchActions = SearchSlice.actions;
export type TSearchState = typeof initialState;
export default SearchSlice.reducer;
```

```
import { configureStore, combineReducers } from "@reduxjs/toolkit";
import userSlice from "../user.slice";
import searchSlice from "../search.slice";

const store = configureStore({
  reducer: { userSlice, searchSlice },
  middleware: (getDefaultMiddleware) => {
    return getDefaultMiddleware({
      serializableCheck: false,
    })
  },
});

const rootReducer = combineReducers({ userSlice, searchSlice });
export type TRootState = ReturnType<typeof rootReducer>;
export default store;
```

## Search slice – תרגול חלק א'

- פיתחו את הפרויקט מהשיעור הקודם.
- צרו slice חדש עבור החיפוש.
- באובייקט ה-`initialState` הגדירו שדה בשם `search` בעל ערך ריק שהסוג שלו `string`.
- ב-reducer הגדירו פונקציה בשם `searchWord` שמקבלת payload שהוא `string` ומעדכנת את ה-`state` הרלוונטי.
- הוסיפו את ה-slice שייצרתם לקובץ ה-`bigPie`.
- הוסיפו `input` לשם חיפוש בבר הניווט.
- טפלו ב-`input` כך שבעת הקלדה יעודכן הערך השמור ב-`state` של `searchSlice` (יש להשתמש ב-`dispatch`)

```
<Navbar.Brand>
  <TextInput type="search" rightIcon={GoSearch} onChange={search} />
</Navbar.Brand>
```

```
const dispatch = useDispatch();
const search = (e: React.ChangeEvent<HTMLInputElement>) => {
  dispatch(searchActions.searchWord(e.target.value));
};
```

My App



Home

Sign Out

Profile

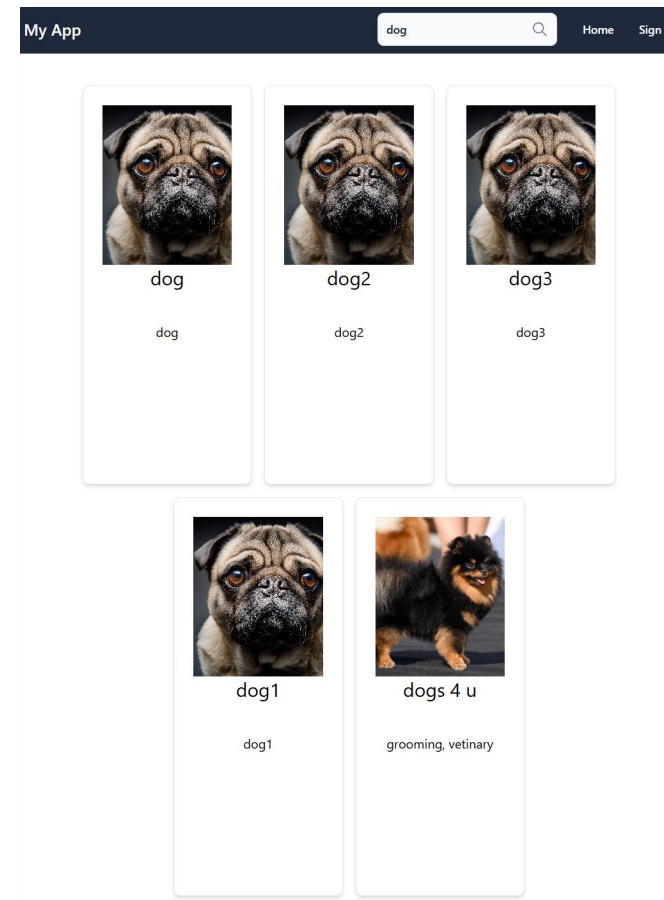
# Search slice – תרגול חלק ב'

```
const search = useSelector((state: TRootState) => state.searchSlice.search);
```

```
const filterBySearch = () => {
  if (cards) {
    return cards.filter(
      (card) =>
        card.title.toLowerCase().includes(search.toLowerCase()) ||
        card.subtitle.toLowerCase().includes(search.toLowerCase()),
    );
  }
  return cards;
};
```

```
return (
  <div className="mt-10 flex flex-wrap justify-center gap-4">
    {cards &&
      filterBySearch()?.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>
);
```

- ייבאו את ערך ה-search הנוכחי לעמוד Home.
- טפלו בעמוד Home כך שתבוצע פעולת סינון (באמצעות המתודה filter) למערך הכרטיסים בהתאם לערך החיפוש הנוכחי טרם הרינדור.



PATCH like/unlike a card

https://monkfish-app-z9uza.ondigitalocean.app/bcard2/cards/6601941691640a0b5122cdd5

- Please note that this EndPoint will unlike an already liked card
- You will need to provide a token to get an answer from this api

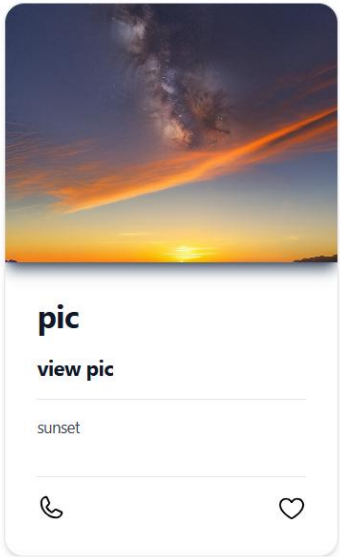
HEADERS

x-auth-token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2NTNhNjA4ZDFjN2NkODBiMWZkMjc1MzliLCJpc0J1c2luZXNzIj09bWVILCJpc0FkbWlulj09bWVILCJpc0E3MTQ5MDE3NTF9.2z5kfoALae61tXb-POAd\_DSYfK6Ve5TrEkhkmHbZegg

טיפול בכרטיסים מועדפים

- כיצד נוכל לדעת מי עשה "לייק" לכרטיס ביקור מסויים?
- כיצד נוכל לסמן כרטיס מסויים כמועדף?
- כיצד נוכל להראות למשתמש רק את הכרטיסים שהוא עשה להם "לייק"?
- כיצד נוכל לבצע קריאת API לשם סימון כרטיס כמועדף על ידי משתמש?
- מה יש לשלוח ב-header?
- כיצד נשיג את ה-token שהמשתמש התחבר איתו לשם שליחה ב-header?



```
const isLoggedIn = useSelector(
  (state: TRootState) => state.userSlice.isLoggedIn,
);
```

```
const likeOrUnlike = async (e: React.MouseEvent<SVGElement>) => {
  const cardId = e.currentTarget.id;
  try {
    if (isLikedCard(cardId)) {
      axios.defaults.headers.common["x-auth-token"] =
        localStorage.getItem("token");
      await axios.patch(
        `https://monkfish-app-z9uza.ondigitalocean.app/bcard2/cards/${cardId}`,
      );
      toast.success("Unliked Card");
    } else {
      await axios.patch(
        `https://monkfish-app-z9uza.ondigitalocean.app/bcard2/cards/${cardId}`,
      );
      toast.success("Liked Card");
    }
  } catch (error) {
    toast.error("Error Liking Card");
  } finally {
    setCards(
      cards.map((card) => {
        if (card._id === cardId) {
          if (isLikedCard(cardId)) {
            card.likes = card.likes.filter((like) => like !== user.user!._id);
          } else {
            card.likes.push(user.user!._id);
          }
        }
        return card;
      })
    );
  }
};
```

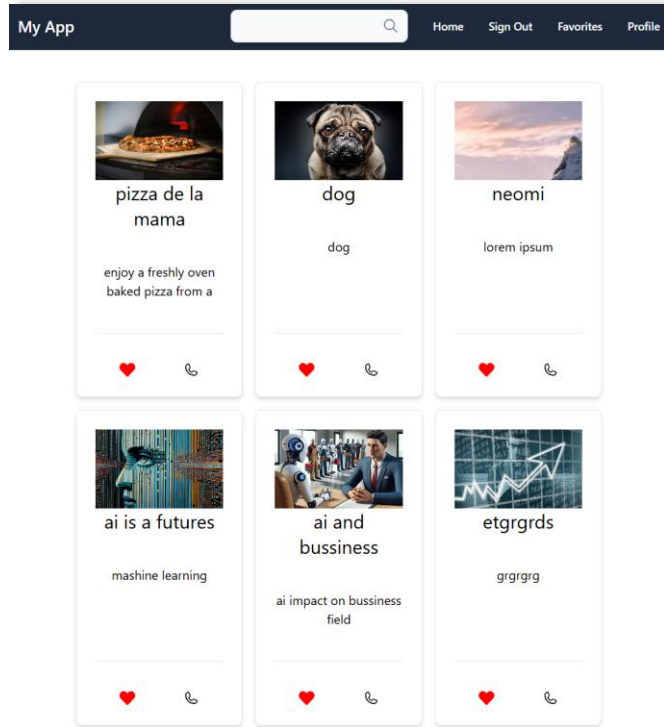
## טיפול בכרטיסים מועדפים – תרגול חלק א'

- ייבאו אייקון בצורת לב לאפליקציה באמצעות שימוש בספרייה react-icons.
- טפלו בעמוד הבית כך שיוצג האייקון בתחתית כל כרטיס ביקור רק במידה ויש משתמש מחובר.
- במידה וה-id של המשתמש נמצא במערך ה"לייקים" של כרטיס ביקור כלשהו (כלומר שמדובר בכרטיס שהמשתמש עשה לו "לייק" בעבר) – הציגו את האייקון בצבע אדום.
- צרו פונקציה בשם likeOrUnlike שתקבל id של כרטיס ותבצע קריאת API שתגרום לסימון "like/unlike" על ידי המשתמש לאותו הכרטיס ותעדכן את ה-state של הכרטיסים.
- לחיצה על האייקון תקרא לפונקציה likeOrUnlike.

```
<div className="flex">
  {user.isLoggedIn && (
    <FaHeart
      size={20}
      className="m-auto cursor-pointer"
      color={isLikedCard(card._id) ? "red" : "black"}
      id={card._id}
      onClick={likeOrUnlike}
    />
  )}
  <PiPhone size={20} className="m-auto" />
</div>
```

# טיפול בכרטיסים מועדפים – תרגול חלק ב'

```
{user.isLoggedIn && (
  <>
    <Navbar.Brand className="text-white">Sign Out</Navbar.Brand>
    <Navbar.Brand>
      <Link to={"/favorites"} className="text-white">
        Favorites
      </Link>
    </Navbar.Brand>
    <Navbar.Brand>
      <Link to={"/profile"} className="text-white">
        Profile
      </Link>
    </Navbar.Brand>
  </>
)}
```



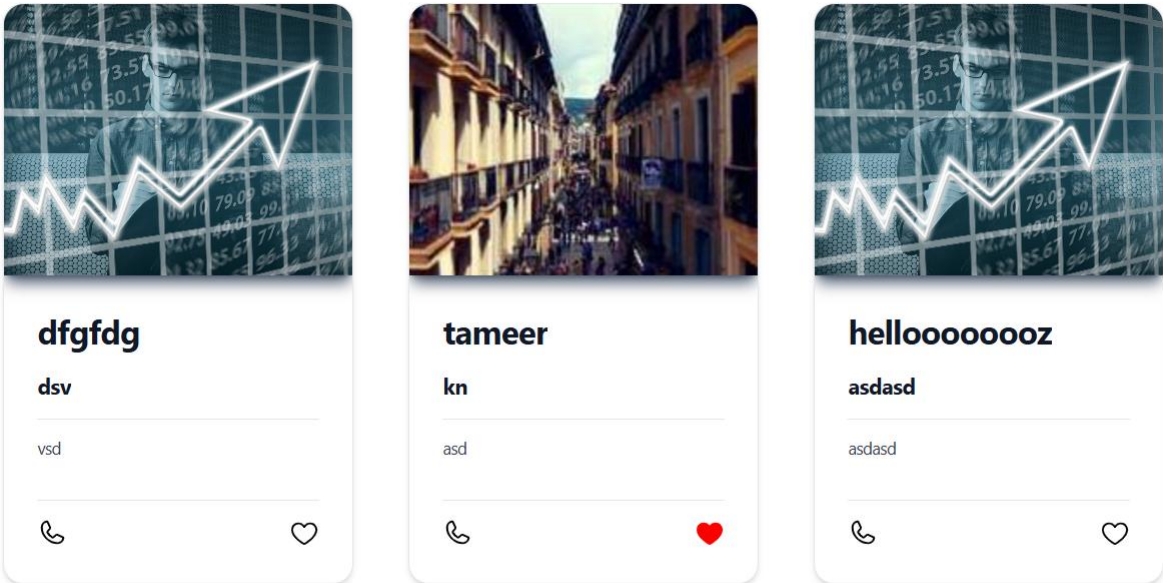
- צרו עמוד מועדפים בשם Favorites.
- צרו נתיב וקישור תואם.
- טפלו בעמוד כך שיוצג רק למשתמש מחובר (באמצעות שימוש ב-RouteGuard).
- הציגו בעמוד זה בעת הטעינה (באמצעות שימוש ב-useEffect ו-useState) רק כרטיסים שהמשתמש המחובר עשה להם "לייק" (תזדקקו למתודה filter לשם כך).
- טפלו בעמוד כך שתבוצע פעולת סינון (באמצעות המתודה filter) למערך הכרטיסים בהתאם לערך החיפוש הנוכחי טרם הרינדור.
- גירמו ל-useEffect לרוץ בכל פעם שמשתנה מילת החיפוש.
- עדכנו את ה-state של מערך הכרטיסים בכל פעם שמשתמש ילחץ על אייקון הלב בהתאם.

\* שימו לב, כרטיס שכבר מסומן ב"לייק" מופיע בדף זה, לחיצה על האייקון תגרום לכרטיס כבר לא להיות מסומן ב"לייק", כך שלאחר הלחיצה הכרטיס צריך להיעלם מעמוד המועדפים של משתמש זה.

# עימוד/Pagination



- מה זה עימוד\ Pagination?
- מה הם היתרונות שבשימוש?
- האם ל-Flowbite יש קומפוננטה ייעודית לשם כך?
- איזה props יש לקומפוננטה זו שעלינו להכיר?
- איזה state עלינו לנהל בעצמנו לשם השימוש?
- כיצד נוכל להציג את הכרטיסים בהתאם לעמוד שבו אנחנו נמצאים?





# Pagination – תרגול

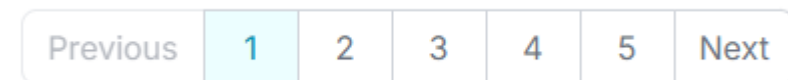
```
const [currPage, setCurrPage] = useState<number>(1);
```

```
const onPageChange = (page: number) => {
  setCurrPage(page);
};
```

```
const filterByPage = () => {
  const start = (currPage - 1) * 9;
  const end = start + 9;
  return filterBySearch()?.slice(start, end);
};
```

```
<Pagination
  currentPage={currPage}
  totalPages={Math.ceil(cards.length) / 9}
  onPageChange={onPageChange}
  className="mt-16 flex w-full justify-center"
/>
```

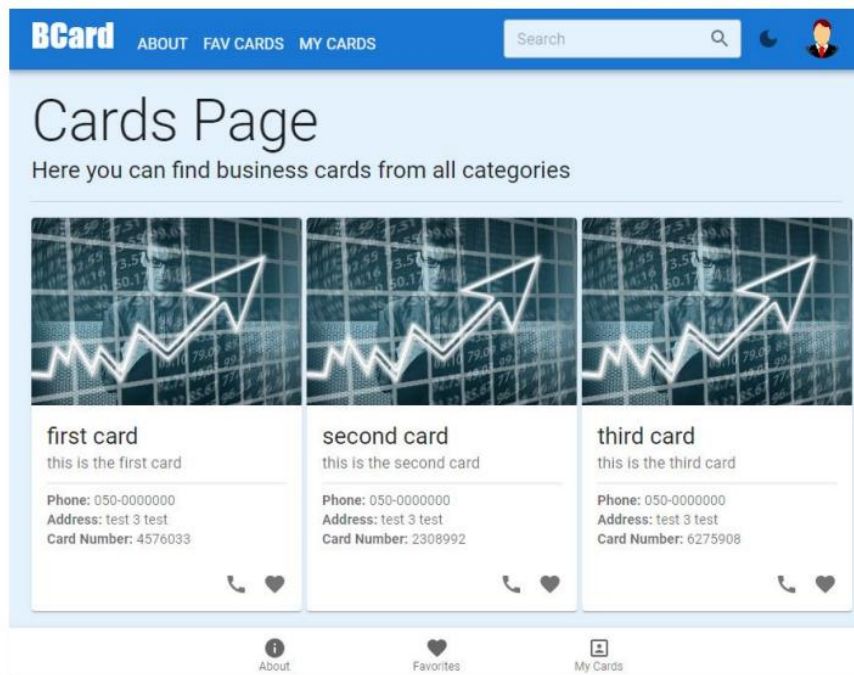
- ייבאו את Pagination לעמוד Home.
- צרו state מספרי שהערך שהוא מחזיק הוא העמוד הנוכחי, כערך התחלתי תנו לו 1.
- צרו פונקציה לשם דיפדוף, הפונקציה תעדכן את ה-state בהתאם.
- כ-totalPages תנו את האורך של מערך הכרטיסים.
- סננו את המערך של הכרטיסים בהתאם לעמוד שבו המשתמש נמצא טרם רינדור (הציגו 9 כרטיסים בעמוד).
- חיזרו על התרגיל בקומפוננטה Favorites.



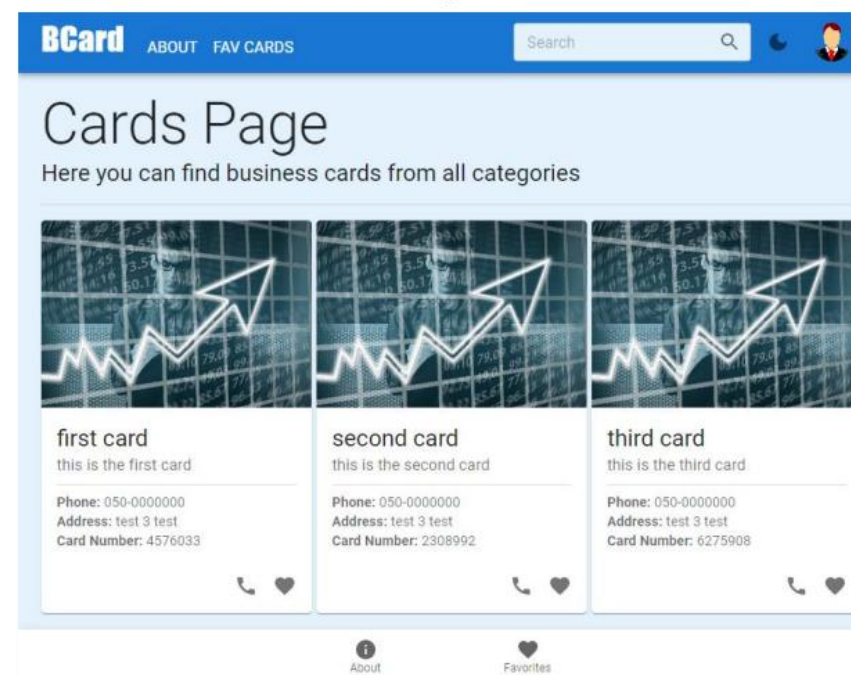
# שיעורי בית

- יש לחזור ולסיים את כל התרגילים שבמצגת.
- טפלו בחיפוש בהתאם למה שלמדתם.
- צרו עמוד מועדפים.
- צרו עימוד ואפשרו חיפוש בכל העמודים שמציגים מערך של כרטיסי ביקור.

תפריט ניווט טעם משתמש מחובר מסוג עסקי



תפריט ניווט עם משתמש מחובר שאינו עסקי



# תודה על ההקשבה

אני וצוות המכללה כאן עבורכם