

```
import { useState } from "react";

const tempMovieData = [
  {
    imdbID: "tt1375666",
    Title: "Inception",
    Year: "2010",
    Poster:
      "https://m.media-amazon.com/images/M/MV5BMjAxMzY3Njc5NF5BMl5BanBnXkFtZTcwNTI5OTM0Mw@@._V1_SX300.jpg",
  },
  {
    imdbID: "tt0133093",
    Title: "The Matrix",
    Year: "1999",
    Poster:
      "https://m.media-amazon.com/images/M/MV5BNzQzOTk3OTAtNDQ0Zi00ZTVkLWI0MTEtMD1lZjNkYzNjNTc4L2l1tYwdlXkEyXkFqcGdeQXVyNjU0OTQ0OTY@._V1_SX300.jpg",
  },
  {
    imdbID: "tt6751668",
    Title: "Parasite",
    Year: "2019",
    Poster:
      "https://m.media-amazon.com/images/M/MV5BYWZjMjk3ZTIiODQ2ZC00NTY5LWE0ZDYtZTI3MjcwN2Q5NTVkXkEyXkFqcGdeQXVyODk4OTc3MTY@._V1_SX300.jpg",
  },
];

const tempWatchedData = [
  {
    imdbID: "tt1375666",
    Title: "Inception",
    Year: "2010",
    Poster:
      "https://m.media-amazon.com/images/M/MV5BMjAxMzY3Njc5NF5BMl5BanBnXkFtZTcwNTI5OTM0Mw@@._V1_SX300.jpg",
    runtime: 148,
    imdbRating: 8.8,
    userRating: 10,
  },
  {
    imdbID: "tt0088763",
    Title: "Back to the Future",
    Year: "1985",
  },
];
```

```

    Poster:
      "https://m.media-
amazon.com/images/M/MV5BZmU0M2Y1OGUtZjIxNi00ZjBkLTg1MjgtOWIyNTNiZWlWYjRiXkEyXkFqcGdeQXVyMTQxN
zMzNDI@._V1_SX300.jpg",
      runtime: 116,
      imdbRating: 8.5,
      userRating: 9,
    },
  ];

const average = (arr) =>
  arr.reduce((acc, cur, i, arr) => acc + cur / arr.length, 0);

function NavBar() {
  return (
    <nav className="nav-bar">
      <Logo />
      <Search />
      <NumResults />
    </nav>
  );
}

function Logo() {
  return (
    <div className="logo">
      <span role="img"><img alt="Popcorn icon" data-bbox="278 518 298 533"/></span>
      <h1>usePopcorn</h1>
    </div>
  );
}

function Search() {
  const [query, setQuery] = useState("");

  return (
    <input
      className="search"
      type="text"
      placeholder="Search movies..."
      value={query}
      onChange={(e) => setQuery(e.target.value)}
    />
  );
}

function NumResults() {
  return (
    <p className="num-results">
      Found <strong>X</strong> results
    </p>
  );
}

```

```

    </p>
  );
}

function Main() {
  return (
    <main className="main">
      <ListBox />
      <WatchedBox />
    </main>
  );
}

export default function App() {
  return (
    <>
      <NavBar />
      <Main />
    </>
  );
}

function ListBox() {
  const [isOpen1, setIsOpen1] = useState(true);

  return (
    <div className="box">
      <button
        className="btn-toggle"
        onClick={() => setIsOpen1((open) => !open)}
      >
        {isOpen1 ? "-" : "+"}
      </button>
      {isOpen1 && <MovieList />}
    </div>
  );
}

function MovieList() {
  const [movies, setMovies] = useState(tempMovieData);

  return (
    <ul className="list">
      {movies?.map((movie) => (
        <Movie movie={movie} key={movie.imdbID} />
      ))}
    </ul>
  );
}

```

```

function Movie({ movie }) {
  return (
    <li>
      <img src={movie.Poster} alt={`${movie.Title} poster`} />
      <h3>{movie.Title}</h3>
      <div>
        <p>
          <span>👤</span>
          <span>{movie.Year}</span>
        </p>
      </div>
    </li>
  );
}

```

```

function WatchedBox() {
  const [watched, setWatched] = useState(tempWatchedData);
  const [isOpen2, setIsOpen2] = useState(true);

  return (
    <div className="box">
      <button
        className="btn-toggle"
        onClick={() => setIsOpen2((open) => !open)}
      >
        {isOpen2 ? "-" : "+"}
      </button>
      {isOpen2 && (
        <>
          <WatchedSummary watched={watched} />
          <WatchedMoviesList watched={watched} />
        </>
      )}
    </div>
  );
}

```

```

function WatchedSummary({ watched }) {
  const avgImdbRating = average(watched.map((movie) => movie.imdbRating));
  const avgUserRating = average(watched.map((movie) => movie.userRating));
  const avgRuntime = average(watched.map((movie) => movie.runtime));

  return (
    <div className="summary">
      <h2>Movies you watched</h2>
      <div>
        <p>
          <span>📺</span>
          <span>{watched.length} movies</span>
        </p>

```

```

    <p>
      <span>☆</span>
      <span>{avgImdbRating.toFixed(2)}</span>
    </p>
    <p>
      <span>✪</span>
      <span>{avgUserRating.toFixed(2)}</span>
    </p>
    <p>
      <span>⌚</span>
      <span>{avgRuntime} min</span>
    </p>
  </div>
</div>
);
}

function WatchedMoviesList({ watched }) {
  return (
    <ul className="list">
      {watched.map((movie) => (
        <WatchedMovie movie={movie} key={movie.imdbID} />
      ))}
    </ul>
  );
}

function WatchedMovie({ movie }) {
  return (
    <li>
      <img src={movie.Poster} alt={` ${movie.Title} poster`} />
      <h3>{movie.Title}</h3>
      <div>
        <p>
          <span>☆</span>
          <span>{movie.imdbRating}</span>
        </p>
        <p>
          <span>✪</span>
          <span>{movie.userRating}</span>
        </p>
        <p>
          <span>⌚</span>
          <span>{movie.runtime} min</span>
        </p>
      </div>
    </li>
  );
}

```

