

# React

A JavaScript Library for building user interfaces

Created By David Yakin 2023

# תוכן עניינים

7	.....	Introduction	➤
10	.....	<u>Virtual DOM</u>	➤
13	.....	<u>Create-react-app</u>	➤
22	.....	<u>React Server</u>	➤
25	.....	<u>React Developer Tools</u>	➤
28	.....	<u>Getting Started</u>	➤
36	.....	<u>Bable.js</u>	➤
40	.....	<u>Component</u>	➤
45	.....	<u>Template</u>	➤
49	.....	<u>Compilation Error</u>	➤
53	.....	<u>Logic</u>	➤
55	.....	<u>Sting interpolation</u>	➤
57	.....	<u>Styles</u>	➤
59	.....	<u>Inline style</u>	➤
61	.....	<u>Styles from module</u>	➤
63	.....	<u>External libraries</u>	➤
65	.....	<u>Props</u>	➤
66	.....	<u>Passing string</u>	➤
70	.....	<u>Passing Object</u>	➤
74	.....	<u>Sending two keys</u>	➤

80	.....	<u>Loops</u>	➤
84	.....	<u>Conditional Rendering</u>	➤
86	.....	<u>Events</u>	➤
87	.....	<u>JAVASCRIPT EVENTS</u>	➤
89	.....	<u>Function invocation with parameters</u>	➤
91	.....	<u>Catching Event</u>	➤
94	.....	<u>Raising Events</u>	➤
98	.....	<u>PropTypes</u>	➤
102	.....	<u>PropTypes Errors</u>	➤
105	.....	<u>Main Types</u>	➤
107	.....	<u>Array &amp; Object of Types</u>	➤
109	.....	<u>oneOfType vs oneOf</u>	➤
112	.....	<u>Exact &amp; isRequired</u>	➤
116	.....	<u>Shape Any &amp; defaultProps</u>	➤
119	.....	<u>node &amp; children</u>	➤
128	.....	<u>Shared Components</u>	➤
129	.....	<u>PageHeader.jsx</u>	➤
133	.....	<u>Static Folder</u>	➤
137	.....	<u>React Hooks</u>	➤
140	.....	<u>useState</u>	➤

152	.....	<u>Layout</u>	➤
160	.....	<u>Error Page</u>	➤
163	.....	<u>React Router Dom</u>	➤
169	.....	<u>Routes</u>	➤
171	.....	<u>BrowserRouter</u>	➤
174	.....	<u>Link &amp; NavLink</u>	➤
181	.....	<u>useNavigate</u>	➤
183	.....	<u>Navigate</u>	➤
187	.....	<u>useParams</u>	➤
193	.....	<u>Nested Routes</u>	➤
198	.....	<u>Life Cycle Hooks</u>	➤
201	.....	<u>Initial rendering</u>	➤
204	.....	<u>useEffect</u>	➤
213	.....	<u>Custom hooks</u>	➤
219	.....	<u>Memoization</u>	➤
221	.....	<u>useCallback</u>	➤
228	.....	<u>useMemo</u>	➤
233	.....	<u>axios</u>	➤
238	.....	<u>cardApiService</u>	➤
244	.....	<u>CardsFeedback.jsx</u>	➤
253	.....	<u>useCards</u>	➤
261	.....	<u>axios interceptors</u>	➤



267	.....	<u>Context</u>	➤
270	.....	<u>example</u>	➤
281	.....	<u>Snackbar</u>	➤
287	.....	<u>Forms</u>	➤
291	.....	<u>Input.jsx</u>	➤
294	.....	<u>FormButton.jsx</u>	➤
297	.....	<u>Form.jsx</u>	➤
301	.....	<u>useForm.js</u>	➤
309	.....	<u>Form implementation</u>	➤
314	.....	<u>Login</u>	➤
316	.....	<u>jwt-decode</u>	➤
318	.....	<u>localStorageService.js</u>	➤
320	.....	<u>UserProvider.jsx</u>	➤
323	.....	<u>usersApiService.js</u>	➤
325	.....	<u>useUsers.js</u>	➤
329	.....	<u>Joi-schema</u>	➤
331	.....	<u>Initial Form Data</u>	➤
333	.....	<u>Axios interceptor</u>	➤
335	.....	<u>LoginPage.jsx</u>	➤
339	.....	<u>Get Current User</u>	➤

347	.....	<u>Logout</u>	➤
348	.....	<u>handleLogout</u>	➤
350	.....	<u>MemuLink.jsx</u>	➤
352	.....	<u>Menu.jsx</u>	➤
357	.....	<u>MenuProvider.jsx</u>	➤
364	.....	<u>Signup</u>	➤
366	.....	<u>initialSignupForm.js</u>	➤
368	.....	<u>Normalize User</u>	➤
370	.....	<u>Joi-schema</u>	➤
371	.....	<u>usersApiService.js</u>	➤
374	.....	<u>useUsers.js</u>	➤
376	.....	<u>SignupPage.jsx</u>	➤
381	.....	<u>MyCardsPage.jsx</u>	➤
385	.....	<u>Delete Card</u>	➤
398	.....	<u>Edit Card</u>	➤
400	.....	<u>mapToModel</u>	➤
406	.....	<u>EditCardPage.jsx</u>	➤
411	.....	<u>Like Card</u>	➤
419	.....	<u>FavCardsPage.jsx</u>	➤
423	.....	<u>Search bar</u>	➤
424	.....	<u>useSearchParams</u>	➤
428	.....	<u>SearchBar implementation</u>	➤



# React Introduction

[https://www.youtube.com/watch?v=XxVg\\_s8xAms](https://www.youtube.com/watch?v=XxVg_s8xAms)



# Q Definition

React is a free and open-source front-end JavaScript library for building user interfaces based on UI components.

It is maintained by Meta (formerly Facebook) and a community of individual developers and companies.

React is only concerned with state management and rendering that state to the DOM

creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality.





# Benefits

Virtual DOM

User Experience

State Handle

Components

No Explicit Data Binding

Life cycle hooks

Open source

# Virtual DOM

החידוש והקונספט המרכזי אותו מביאה  
ריאקט הוא את ה - Virtual DOM  
React תעקוב אחר השינויים בדום וירטואלי  
ותעדכן את הדום האמיתי רק במקומות  
שבהם התרחשו השינויים.  
שיטה זאת יוצרת חיסכון אדיר במשאבים  
ומהירות תגובה גבוהה לכל שינוי.

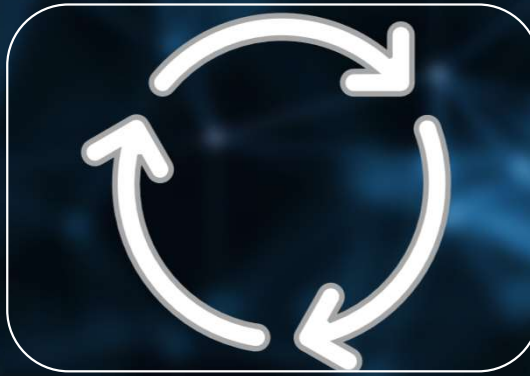
<https://reactjs.org/docs/faq-internals.html>



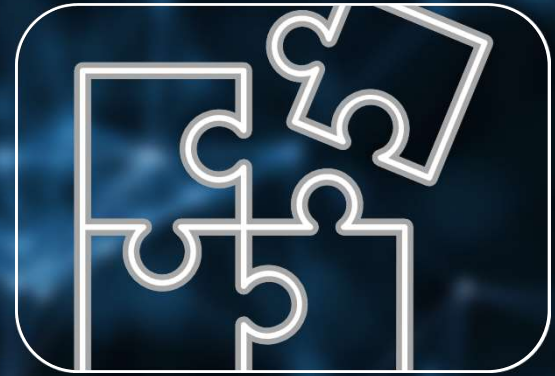
# Virtual DOM implementations



State



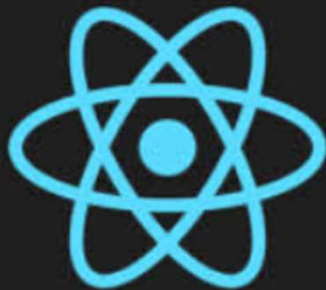
Life cycle  
hooks



Components

# Simple React Snippets

תוסף שיעזור לנו בעבודה עם React בסביבת העבודה של vscode



## Simple React Snippets

burkeholland.simple-react-snippets

Burke Holland

404,829



Repository

Dead simple React snippets you will actually use

Uninstall

*This extension is enabled globally.*





# Create-react-app

כלי שיעזור לנו לפתוח פרויקט חדש ב - React





# Installation

```
npm i -g create-react-app
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

To address all issues (including breaking changes), run:  
`npm audit fix --force`

Run ``npm audit`` for details.

Created git commit.

Success! Created client at C:\Users\DELL\Desktop\HackerU\lecturer-work\Lessons\REACT\REACT-HOOKS\bcard-app\client

Inside that directory, you can run several commands:

`npm start`  
Starts the development server.

`npm run build`  
Bundles the app into static files for production.

`npm test`  
Starts the test runner.

`npm run eject`  
Removes this tool and copies build dependencies, configuration files and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

`cd client`  
`npm start`

Happy hacking!

C:\Users\DELL\Desktop\HackerU\lecturer-work\Lessons\REACT\REACT-HOOKS\bcard-app>

## New React Project

לאחר שהורדנו והתקנו את התוסף  
create-react-app נוכל להשתמש בו  
כדי לפתוח פרויקט חדש ב - React

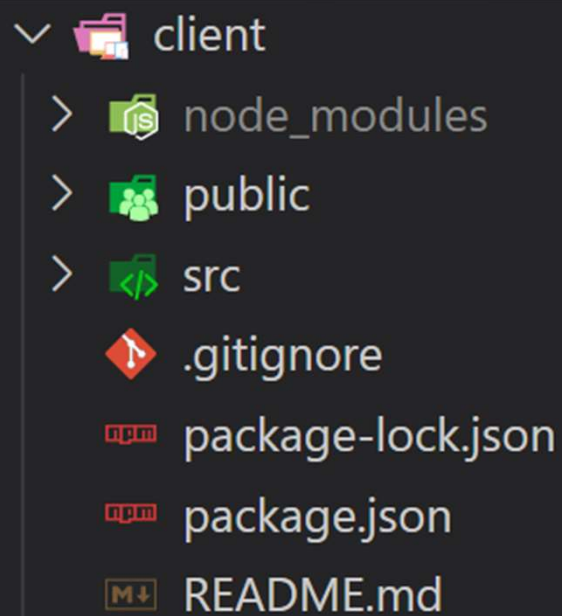
- ניכנס לתוך התיקייה בה אנו  
מעוניינים ליצור פרויקט חדש של  
React

- נקיש בטרמינל את הפקודה  
create-react-app client

- כשפעולת ה - CLI תסתיים ונראה  
את הכיתוב "Happy hacking"

# תשתית הקבצים שה – CLI יצר

- **Node\_modules** – תיקייה השומרת בתוכה את הספריות שהורדנו לפרויקט
- **public** – תיקייה השומרת את הקבצים הסטאטיים
- **src** – תיקייה בה רוב הקוד של הפרויקט ייכתב
- **.gitignore** – קובץ קונפיגורציות של git ובתוכו ההוראות מאילו תיקיות וקבצים להתעלם ולא להעלות ל - github
- **package.json** – קובץ קונפיגורציות הכולל בתוכו פקודות, רשימת תלויות בספריות ועוד
- **package-lock.json** – קובץ השומר את גרסאות הספריות
- **README.md** – קובץ בו ניתן לכתוב פרטים על הפרויקט

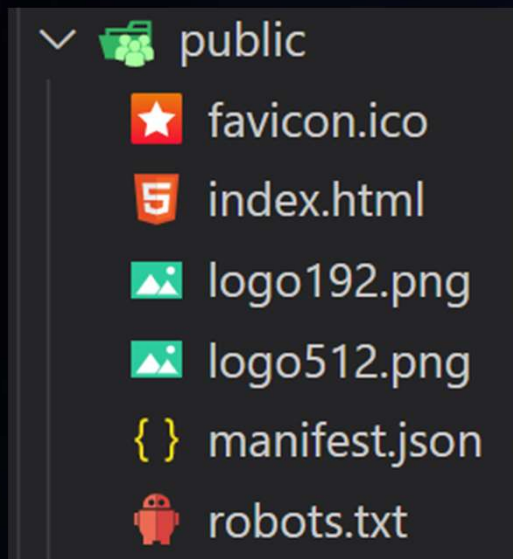


A screenshot of a file explorer window showing the project structure. The root folder is 'client'. Inside 'client', there are four subfolders: 'node\_modules', 'public', 'src', and '.gitignore'. Below the folders are three files: 'package-lock.json', 'package.json', and 'README.md'. Each file has a small icon next to it: a red diamond for '.gitignore', a red square with a white 'M' for 'package-lock.json', and a red square with a white 'M' for 'package.json'.



# Public

- **favicon.ico** – קובץ התמונה שריאקט משתמשת בו באופן דיפולטיבי בפרויקט חדש
- **index.html** – קובץ ה-HTML המרכזי של הפרויקט
- **logo192.png** - הלוגו של ריאקט קטן
- **logo512.png** – הלוגו של ריאקט בגודל יותר
- **manifest.json** – קובץ קונפיגורציות לאפליקציה של מובייל או דסקטופ
- **robots.txt** - קובץ העוזר למנוע החיפוש google בסריקת האפליקציה



! לינק לסרטון המסביר על קובץ robots.txt  
<https://www.youtube.com/watch?v=fzm-zYHjlgY>

```

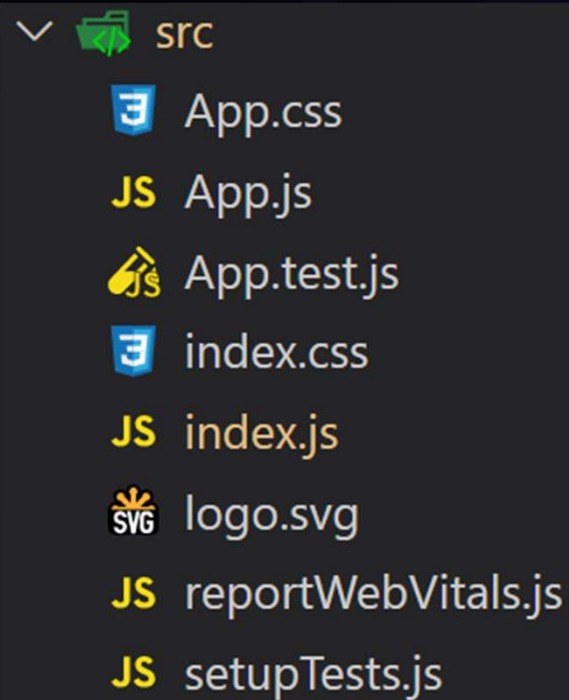
index.html M X
client > public > index.html > ...
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8" />
5     <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
6     <meta name="viewport" content="width=device-width, initial-scale=1" />
7     <meta name="theme-color" content="#000000" />
8     <meta
9       name="description"
10      content="Web site created using create-react-app"
11    />
12    <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
13    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
14    <title>React App</title>
15  </head>
16  <body>
17    <noscript>You need to enable JavaScript to run this app.</noscript>
18    <div id="root"></div>
19  </body>
20 </html>

```

# index.html

- קישור למיקום תמונת ה favicon
- קישור למיקום התמונה ל – apple
- קישור מיקום קובץ manifest.json
- כותרת האפליקציה
- האלמנט אליו יוזרקו כל שאר הקומפוננטות

# src



• App.css – קובץ העיצוב של הקומפוננט app.js

• App.js – קובץ הלוגיקה של הקומפוננט app

• App.test.js – קובץ הבדיקות של הקומפוננט app

• index.css – קובץ העיצוב של הקומפוננט index

• index.js – קובץ הלוגיקה של קומפוננט index

• logo.svg – קובץ הלוגו של ריאקט

• reportWebVitals.js – בדיקת אופטימיזציה של

האפליקציה

• setupTests.js – קובץ הבדיקות המרכזי של האפליקציה

! לינק לסרטון שמסביר על webVitals

<https://www.youtube.com/watch?v=00RoZf1FE34>

JS App.js



client > src > JS App.js > ...

```
1  import logo from './logo.svg';
2  import './App.css';
3
4  function App() {
5    return (
6      <div className="App">
7        <header className="App-header">
8          <img src={logo} className="App-logo" alt="logo" />
9          <p>
10             Edit <code>src/App.js</code> and save to reload.
11          </p>
12          <a
13             className="App-link"
14             href="https://reactjs.org"
15             target="_blank"
16             rel="noopener noreferrer"
17          >
18             Learn React
19          </a>
20        </header>
21      </div>
22    );
23  }
24
25  export default App;
```

# App.js

- ייבוא קובץ ה- logo לקומפוננט
- ייבוא קובץ העיצוב לקומפוננט
- יצירת הקומפוננט App
- התצוגה לגולש (מה שהפונקציה מחזירה)



# index.js

קובץ הלוגיקה המרכזי של האפליקציה

• ייבוא מודולים:

- מופע מהספרייה React לקומפוננט
- קובץ העיצוב של הקומפוננט
- קומפוננט App
- קובץ בדיקות האופטימיזציה לקומפוננט

• יצירת קבוע בשם root שערכו שווה

ערך להפעלת מטודת

ReactDOM.createRoot שתתפוס את האלמנט HTML ב - DOM עם id=root (האלמנט הזה נמצא בתוך הקובץ index.html בתיקיה public)

• הפעלת מטודת root.render אשר תזריק לתוך האלמנט שתפסנו root את הקומפוננטות (ידובר בהרחבה בהמשך)

• עטיפת האפליקציה בקומפוננט של React שייכפה strictMode על הקוד שייכתב בתוכו (ידובר על כך בהרחבה בהמשך המצגת)

• הצבת קומפוננט App כך שתוצג לגולש

• הפעלת מטודת reportWebVitals

JS index.js M X

client > src > JS index.js > ...

```
1 import React from "react";
2 import ReactDOM from "react-dom/client";
3 import "./index.css";
4 import App from "./App";
5 import reportWebVitals from "./reportWebVitals";
6
7 const root = ReactDOM.createRoot(document.getElementById("root"));
8 root.render(
9   <React.StrictMode>
10     <App />
11   </React.StrictMode>
12 );
13
14 reportWebVitals();
```



# React Server

בעזרת הפקודה `npm start` בטרמינל נפעיל  
את השרת הזמני של ריאקט ונוכל לראות את  
התצוגה הראשונית של האפליקציה



# npm start

נפעיל את הסרבר הזמני של React שגם יאזין לשינויים בקוד וייתן לנו תצוגת אמת של הקוד שלנו

- נכנס לתיקייה הרלוונטית בה נמצא פרויקט ה – React
- נקליד את הפקודה `npm start`
- ה – CLI יודיע לנו ובמידה ולא תהיה שגיאה בתהליך `compile` נראה את הכיתוב הבא
- ה – CLI יעדכן אותנו כי אנו מאזינים לפורט הדיפולטיבי של React שהוא פורט 3000
- ה – CLI יעדכן אותנו כי אנחנו בסביבת עבודה של `development`

! כמו כן ה – CLI יפתח את הדפדפן בכתובת ובפורט הרשומים

Compiled successfully!←

You can now view `client` in the browser.

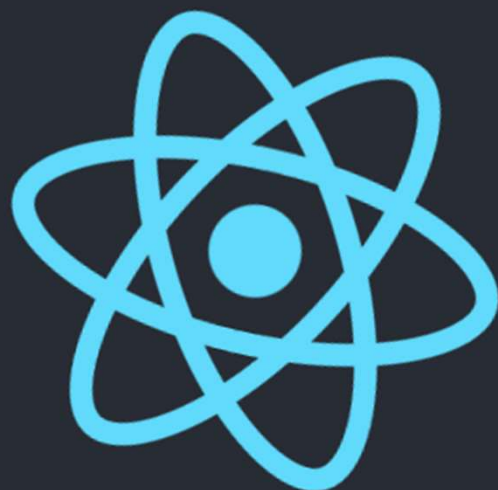
Local: http://localhost:3000 ←

On Your Network: http://10.0.0.8:3000

Note that the development build is not optimized.←  
To create a production build, use `npm run build`.

webpack compiled successfully





Edit src/App.js and save to reload.

[Learn React](#)

## התוצאה בדפדפן

אם לא התקבלה שגיאה והכול תקין אנו  
אמורים לראות בדפדפן את התצוגה  
הבאה



# React Developer Tools

כלי שיעזור לנו בשלבי הפיתוח ב - React



## React Developer Tools

 Featured

★★★★★ 1,402 ⓘ | [Developer Tools](#) | 3,000,000+ users

<https://reactjs.org/blog/2015/09/02noitallatsni#lmth.sloot-repoleved-tcaer-wen>



# Components

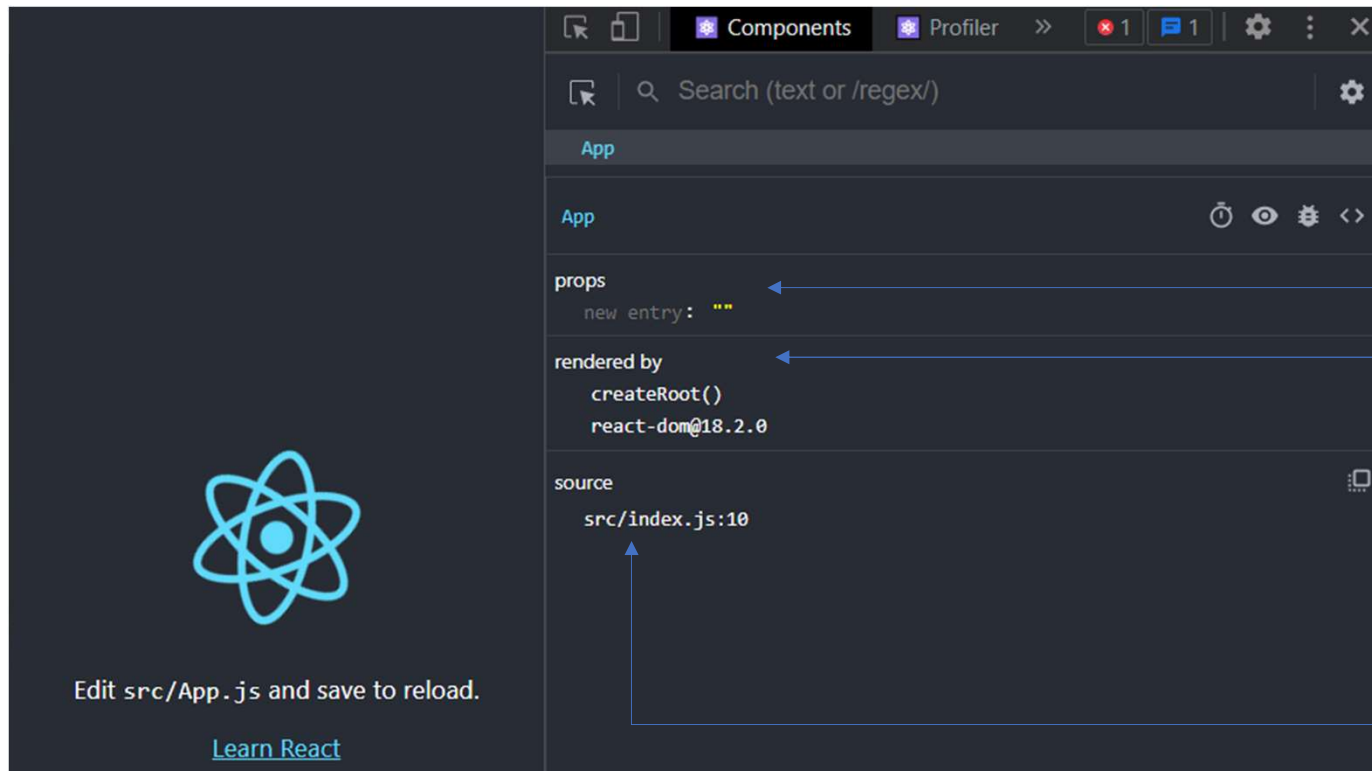
לשונית זאת מראה לנו נתונים נוספים על קומפוננט נבחרת

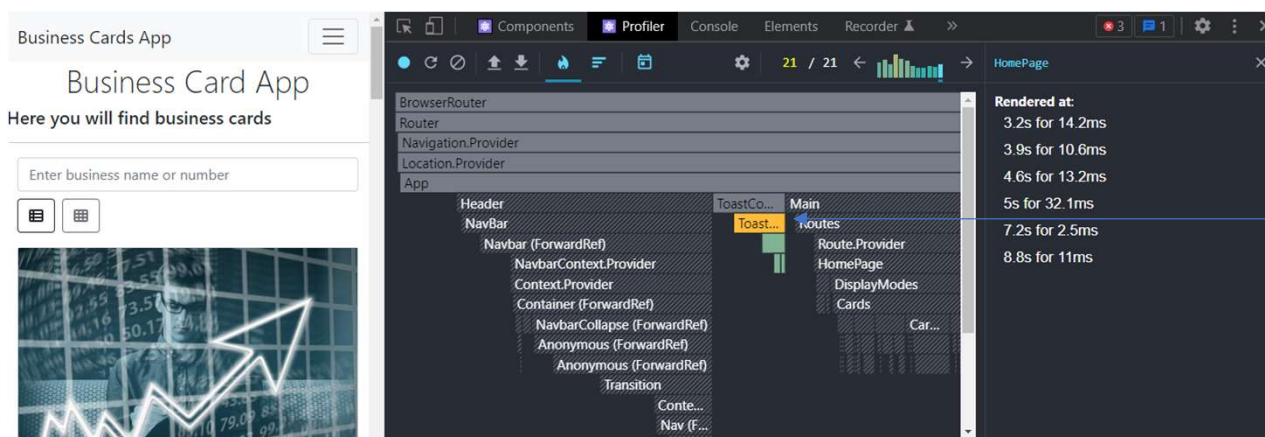
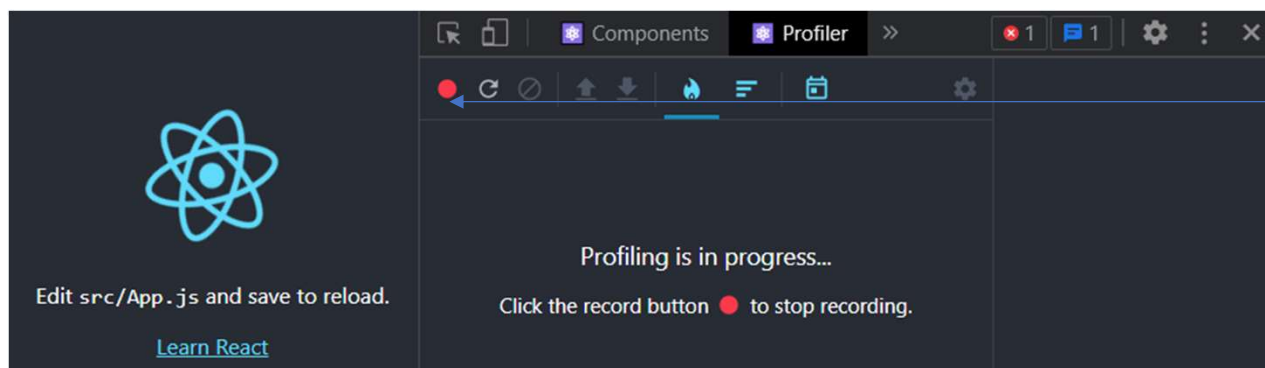
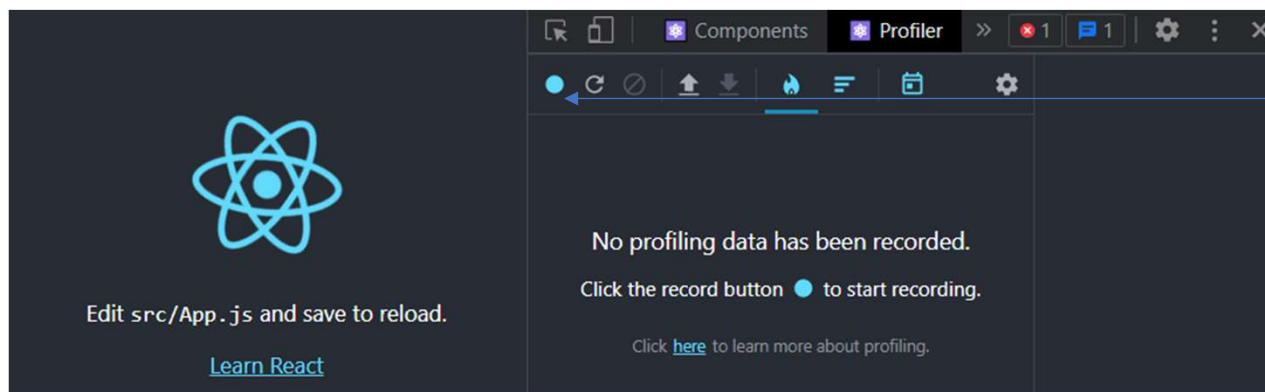
- בחלק זה של הדף נראה איפה עומדת הקומפוננט בהיררכיה של האפליקציה

- החלק השני מדבר על הקומפוננט עצמה ובו ניתן לראות נתונים כמו:

- props – נתונים שהועברו לקומפוננט
- state – משתנים ש React תגיב לשינויים בערכיהם
- rendered by – הפונקציה שאחראית על טעינה וטעינה ומחדש של הקומפוננט
- source – קובץ המקור

! בגלל שאנו לא מנהלים state בקומפוננט הזה אני לא רואה את הנתונים הללו





# Profiler

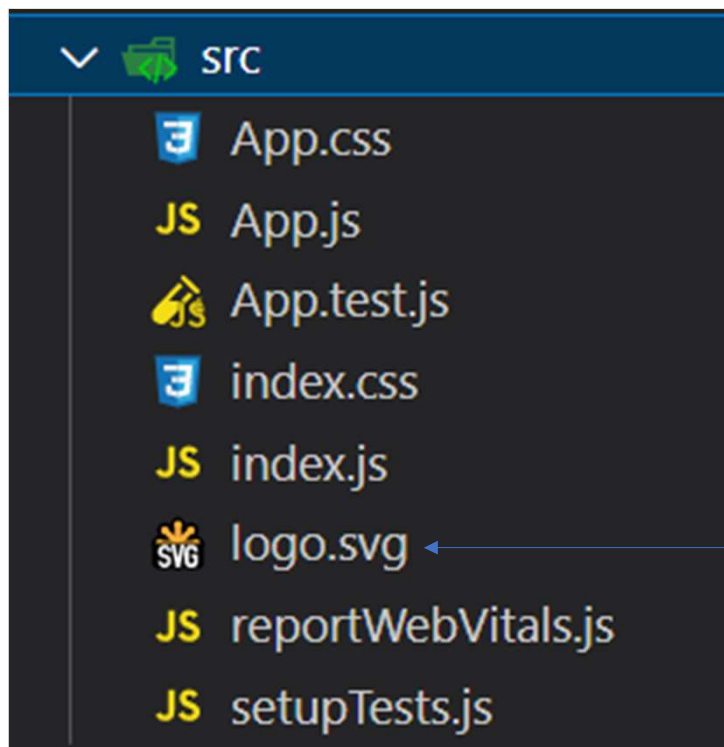
בלשונית זאת נוכל לעשות בדיקות אופטימיזציה

- לחיצה על כפתור record תתחיל להאזין לאירועים שקורים ב – DOM
- לחיצה נוספת תעצור את ההקלטה ובמידה ויהיו נתונים להצגה (כמו משך הזמן שלקח לכל אירוע לפעול) החלקים הללו יוצגו לנו.
- לחיצה על אחד מהאירועים תיתן לנו פרטים עליו

# תחילת עבודה

ניקוי ראשוני של האפליקציה מתמונות וערכים דיפולטיביים של  
create-react-app





# src

- מחיקת קובץ הלוגו של ריאקט



# App.js

- מחיקת ייבוא קובץ הלוגו של ריאקט
- מחיקת תוכן האלמנט div עם המחלקה העיצובית App
- התוצאה

```
1 import logo from './logo.svg';
2 import './App.css';
3
4 function App() {
5   return (
6     <div className="App">
7       <header className="App-header">
8         <img src={logo} className="App-logo" alt="logo" />
9         <p>
10           Edit <code>src/App.js</code> and save to reload.
11         </p>
12         <a
13           className="App-link"
14           href="https://reactjs.org"
15           target="_blank"
16           rel="noopener noreferrer"
17         >
18           Learn React
19         </a>
20       </header>
21     </div>
22   );
23 }
24
25 export default App;
```

```
1 import './App.css';
2
3 function App() {
4   return <div className="App"></div>;
5 }
6
7 export default App;
```

# index.css

- מחיקת תוכן הקובץ
- יצירה של מחלקות עיצוביות משלנו
- מחיקת התוכן של הקובץ App.css

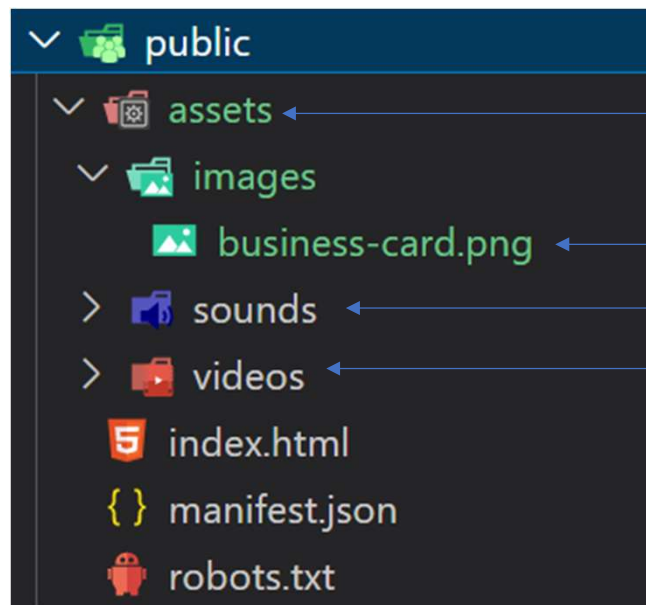
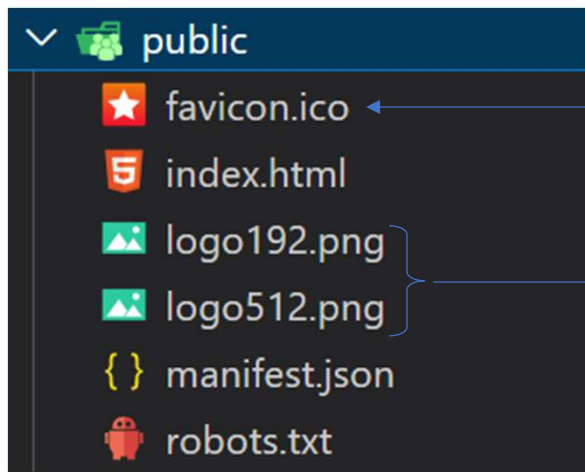
```
index.css X
src > index.css > body
1 body {
2   margin: 0;
3   font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',
4     'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',
5     sans-serif;
6   -webkit-font-smoothing: antialiased;
7   -moz-osx-font-smoothing: grayscale;
8 }
9
10 code {
11   font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',
12     monospace;
13 }
```

```
index.css M X
src > index.css > ...
1 * {
2   margin: 0;
3   padding: 0;
4   box-sizing: border-box;
5 }
6
7 center {
8   display: flex;
9   justify-content: center;
10  align-items: center;
11 }
12
13 cursor {
14   cursor: pointer;
15 }
```

App.css M X

src > App.css

1



# public

- מחיקת קובץ favicon.ico של ריאקט
- מחיקת הלוגואים של react
- הוספת תיקייה בשם assets ובתוכה שלוש תיקיות:
  - Images
    - נוריד מאתר pixabay אייקון מתאים לאפליקציה שלנו
  - sounds
  - videos

```

5 index.html M X
public > 5 index.html > html > head > link
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8" />
5     <link rel="icon" href="%PUBLIC_URL%/assets/images/business-card.png" />
6     <meta name="viewport" content="width=device-width, initial-scale=1" />
7     <meta name="theme-color" content="#000000" />
8     <meta
9       name="description"
10      content="Web site created using create-react-app"
11    />
12    <link
13      rel="apple-touch-icon"
14      href="%PUBLIC_URL%/assets/images/business-card.png"
15    />
16    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
17    <title>Business Cards App</title>
18  </head>
19  <body>
20    <div id="root"></div>
21  </body>
22 </html>

```

# index.html

- החלפת ה icon לתמונת ה – icon שייבאנו לפרויקט
- החלפת ה – icon למקרה ומשתמשים ב – apple
- שינוי הכיתוב באלמנט ה – title לשם האפליקציה

{ } manifest.json M X

public > { } manifest.json > ...

```
1  {
2    "short_name": "Business cards app",
3    "name": "Business card application for business and clients",
4    "icons": [
5      {
6        "src": "./assets/images/business-card.png",
7        "sizes": "64x64 32x32 24x24 16x16",
8        "type": "image/x-icon"
9      },
10     {
11       "src": "./assets/images/business-card.png",
12       "type": "image/png",
13       "sizes": "192x192"
14     },
15     {
16       "src": "./assets/images/business-card.png",
17       "type": "image/png",
18       "sizes": "512x512"
19     }
20   ],
21   "start_url": ".",
22   "display": "standalone",
23   "theme_color": "#000000",
24   "background_color": "#ffffff"
25 }
```

# manifest.json

- שינוי השם המקוצר של האפליקציה
- קביעת השם המלא של האפליקציה
- שינוי מיקום התמונות בשביל ה - icons



## משימת initial app



## Business-cards-app

- הורד את ה – CLI של create-react-app באופן גלובלי
- פתח פרויקט חדש בעזרת create-react-app בשם client
- הכן את הפרויקט לעבודה על ידי ניקוי הקבצים והתיקיות הלא רלוונטיות לפרויקט
- הוסף קבצים ותיקיות שיידרשו לפרויקט כמו שמופיע בשקפים הקודמים