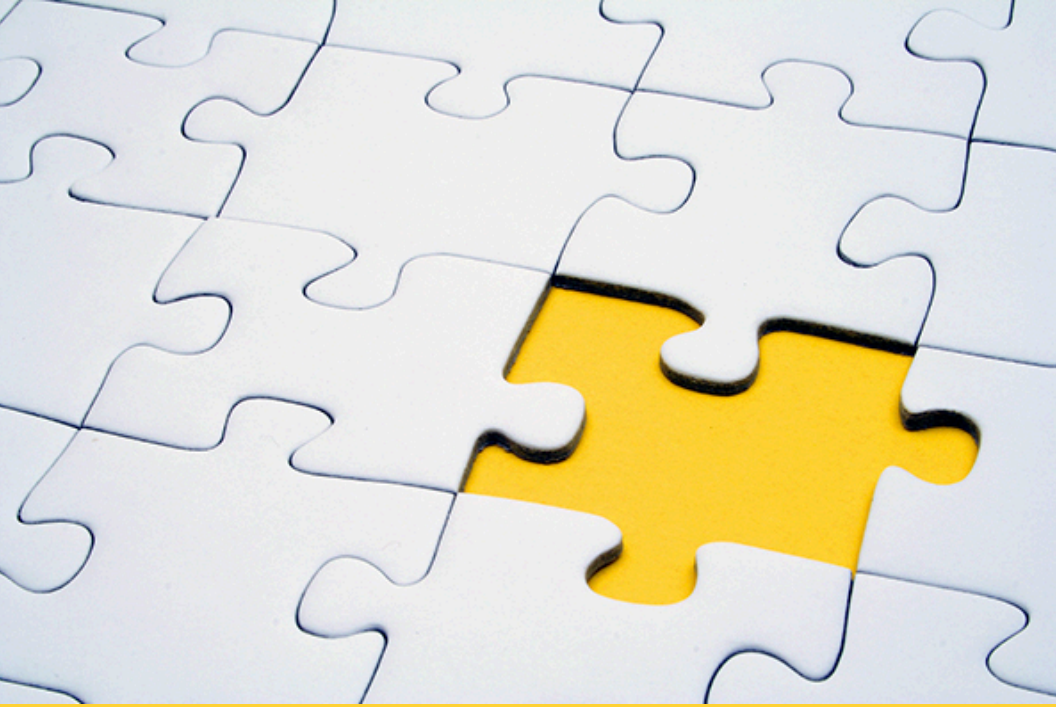


React Native

Full stack *By* keynan perez

npx create-react-app client
cd client
npm start

Lession 1

Props

Parent.js

```
import Demo1ChildComp from "./Demo1Child";

function Demo1ParentComp() {


  return (
    <div style={{ border : "solid 2px green" , width : "400px"}}
className="App">

      <h1>App comp</h1>


      <Demo1ChildComp name="Avi" age={30} address={ {city : "Tel
Aviv", street : "Herzel"}   } />

      <Demo1ChildComp name="Dana" age="40" />
        <h1>App comp Footer</h1>
    </div>
  );
}
export default Demo1ParentComp;
```

Child.js

```
function Demo1ChildComp(props) {


  return (
    <div style={{ border : "solid 2px red" , width : "300px"}}
className="App">

       Name : {props.name}  <br/>
       Age : {props.age}  <br/>
       Street : {props.address?.street}
    </div>
  );
}

export default Demo1ChildComp;
```

Inputs

```
function Demo2Comp() {


  const sayHello = () =>
  {
    alert("Hello");
  }

  const getText = (e) =>
  {
   console.log(e.target.value);
  }

  const getCheck = (e) =>
  {
    console.log(e.target.checked)
  }


   return (
    <div className="App">

     <button onClick={sayHello}>Click Me !</button>  <br/>

     <input type="text" onChange={getText} />  <br/>

    <input type="checkbox" onChange={getCheck} />

    </div>
  );
}
export default Demo2Comp;
```

States

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

function Demo3Comp() {

  //const [name, setName] = useState("Avi");
  //const [age, setAge] = useState(20);
  const [city, setCity] = useState("");

  const [counter, setCounter] = useState(0);


  const [user, setUser] = useState({name : "Avi", age : 20});

  const changeData = () =>
  {
   // setName("Ron");
    //setAge(30);

    //Change ONLY the name
    setUser({...user, name : "Ron"})

    //Change ONLY age
    setUser({...user, age : 30})
  }

 const add1 = () =>
 {
  setCounter(counter+1);
  console.log(counter);
 }

  return (
    <div className="App">

      <button onClick={add1}>+</button>  <br/>

      {counter}  <br/> <br/>



      <button onClick={changeData}>Change Data</button> <br/>


      Name : {user.name}  <br/>
      Age : {user.age}  <br />
      City : <input type="text" onChange={ e =>
setCity(e.target.value)  } /> <br/>
      {city}

    </div>
  );
}

export default Demo3Comp;
```

DHTML Style

JS File

```
import { useState } from 'react';
import './Demo4.css';

function Demo4Comp() {
  const [isRed, setIsRed] = useState(false);
  const [isVisible, setIsVisible] = useState(false);

  return (
    <div className='App'>
      <button onClick={() => setIsRed(!isRed)}>Change Color</
button>

      <h1 style={{ color: isRed ? 'red' : 'blue' }}>Hello World</
h1>
      <br />
      <br />

      <button onClick={() => setIsVisible(!isVisible)}>Show/Hide</
button>

      <h1 className={isVisible ? 'showStyle' : 'hideStyle'}>Hello
World 2</h1>
    </div>
  );
}

export default Demo4Comp;
```

CSS File

```
.showStyle
{
  visibility: visible;
}

.hideStyle
{
  visibility: hidden;
}
```

DHTML CreateDelete

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

function Demo5Comp() {
  const [isExist, setIsExist] = useState(false);

  return (
    <div className='App'>
      <button onClick={() => setIsExist(!isExist)}>Create/Destroy</
button>

      {isExist && <h1>Helo World 3</h1>}

      {isExist ? <h1>Helo World 3</h1> : null}
    </div>
  );
}
export default Demo5Comp;
```

DHTML Repeater

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

function Demo6Comp() {
  const [colors, setColors] = useState(['Red', 'Green', 'Blue']);

  const [products, setProducts] = useState([
    { name: 'PC', price: 100 },
    { name: 'Tablet', price: 200 },
  ]);

  const addColor = () => {
    setColors([...colors, 'Yellow']);
  };

  return (
    <div>
      <table border='1'>
        <tr>
          <th>Name</th>
          <th>Price</th>
        </tr>
        <tbody>
          {products.map((item, index) => {
            return (
              <tr key={index}>
                <td>{item.name}</td>
                <td>{item.price}</td>
              </tr>
            );
```

```
          })}
        </tbody>
      </table>
      <button onClick={addColor}>Add Color</button> <br />
      {colors.map((item) => {
        return <div key={item}>{item}</div>;
      })}
      <br />
      <ul>
        {colors.map((item, index) => {
          return <li key={index}>{item}</li>;
        })}
      </ul>
    </div>
  );
}

export default Demo6Comp;
```

Ex1

Child

```
function Ex1ChildComp(props) {

  const showPhones = () =>
  {
    console.log(props.phones);
  }


  return (
    <div className="App">

    <button style={ {backgroundColor : "red"}  } onClick={ () =>
console.log(props.phones)    }>Show Phones </button>
    <button style={ {backgroundColor : "red"}  }
onClick={ showPhones    }>Show Phones </button>

    </div>
  );
}

export default Ex1ChildComp;
```

Parent

```
import Ex1ChildComp from "./Ex1Child";

function Ex1ParentComp() {


  return (
    <div className="App">

      <Ex1ChildComp phones={ {home : "04-444" , mobile : "052-2222"
, fax : "077-7777"}       } />

    </div>
  );
}

export default Ex1ParentComp;
```

Lession 2

components repeater

רפיטר שעובר על קומפוננטות ומחזיר אותן

App.js

```
import { useState } from 'react';
import ChildComp from './ChildComp';

const App = () => {
  const [names, setNames] = useState(['Ron', 'Gili', 'Dana',
'John']);

  return (
    <div>
      {names.map((name, index) => {name עבור כל איבר שקראנו לו
name תחזיר קומפוננטה שהkey שלה הוא אינדקס ותעביר לה ערך
        return <ChildComp key={index} name={name} />;
      })}
    </div>
  );
};

export default App;
```

ChildComp.js

```
import React from 'react';

const ChildComp = (props) => {
  return (מחזיר דיב עם שם
    <div>
      Name: {props.name}
    </div>
  );
};
export default ChildComp;
```

lifting state up

איך נעביר מידע מבן לאבא באמצעות פונקציה וכתובת

App.js

```
import {useState} from 'react';
import ChildComp from './ChildComp'

const App = () => {
  const [data, setData] = useState('')
ניצור משתנה מצב דאטה שיקבל את המידע מהילד בעתיד

  const getDataFromChild = (childValue) => {
    // some code…
    setData(childValue)פונקציה שמקבלת מידע מהילד ושמה אותו בדאטה
  }
  return (
    <>
      <h3>Parent Comp</h3>
      Data From Child: {data}
      <br /> <br />
      <ChildComp callback={getDataFromChild} />נריץ את הילד ונשלח
callback איתו את הכתובת של הפונקציה באבא במשתנה
    </>
  );
};
export default App;
```

ChildComp.js

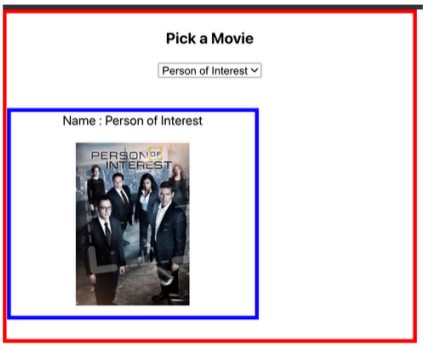
```
import React from 'react';

const ChildComp = (props) => {

  const handleChange = (e) => { פונקציה שמטפלת בכל שינוי שמתבצע
באינפוט
    props.callback(e.target.value)שולחת את האינפוט לפונקציית האבא
  }

  return (
    <>
      <h3>Child Comp</h3>
      Send To Parent: <input type="text" onChange={handleChange} />
handleChange נגדיר שבעת שינוי תפעל הפונקציה
  **למה הפונקציה לא מקבלת כלום? ואין סוגריים פתוחים? כי הוא קיבל גם
כתובת לפונקציה    </>
  );
};
```

```
export default ChildComp;
```

במידה ונרצה לשמור כמה משתנים מכמה אינפוטים שונים ולשלוח אותם רק אחרי לחיצה על כפתור לדוגמא אז נשמור את כל האינפוטים במשתני מצב ולאחר לחיצה על הכפתור נמשוך אותם לפונקציה המטפלת ונשלח אותם

React – Ex – Components Communication

Ex 1



Create a (parent) component with movies data as follows (copy-pase it to your code) :

[{id : 1 , name : "Under the Dome", pic : "https://static.tvmaze.com/uploads/images/medium_portrait/81/202627.jpg"},
{id : 2 , name : "Person of Interest", pic : "https://static.tvmaze.com/uploads/images/medium_portrait/163/407679.jpg"},
{id : 3 , name : "Bitten", pic : "https://static.tvmaze.com/uploads/images/medium_portrait/0/15.jpg"}]

Ex4_1_Parent.js

```
import { useState } from "react";
import Ex4_1_ChildComp from "./Ex4_1_Child";

function Ex4_1_ParentComp() {

  const [movies, setMovies] = useState([{id : 1 , name : "Under the Dome", pic :
                                        "https://static.tvmaze.com/uploads/images/medium_portrait/
81/202627.jpg"},
                                        {id : 2 , name : "Person of Interest", pic :
                                        "https://static.tvmaze.com/uploads/images/medium_portrait/
163/407679.jpg"},
                                        {id : 3 , name : "Bitten", pic :
                                        "https://static.tvmaze.com/uploads/images/medium_portrait/
0/15.jpg"}])


   const [selectedMovie, setSelectedMovie] = useState({name :'',
pic : ''})
ניצור משתנה מצב ריק של הסרט הנבחר

  const selectMovie = (e) =>
  {
    let movieid = e.target.value;

    let movie = movies.find(x => x.id == movieid);

    setSelectedMovie(movie)

  }
הפונקציה תפעל כשאר נבחר סרט מהרשימה,תמשוך את הid מהvalue של האלמנט
find ניצור משתנה movie שיקבל לתוכו סרט ע"י פונקציית

  return (
    <div >

      <h2>Pick A Movie</h2>

       <select onChange={selectMovie}>
       {
        movies.map(item => נרוץ על הסרטים
          {
            return <option  value={item.id} key={item.id}
>{item.name}</option>לכל סרט נחזיר option שהערך הוא שם הסרט והמפתח
הוא id של הסרט
והמשתנה שרשום הוא גם שם הסרט
          })
       }
       </select>  <br/>

       <Ex4_1_ChildComp movieData={selectedMovie} />
נציג את הסרט הנבחר כאשר נעביר לו בפרופס את האובייקט של הסרט שמכיל
שם ותמונה
    </div>
  );
}

export default Ex4_1_ParentComp;
```

Ex4_1_Child.js

```
function Ex4_1_ChildComp(props) {
  return (
    <div >

      Name : {props.movieData.name}  <br/>

      <img src={props.movieData.pic} />

    </div>
  );
}

export default Ex4_1_ChildComp;
```

rest-api

app.js

```
import { useState } from 'react';
import axios from 'axios';

const url = 'https://jsonplaceholder.typicode.com/users';

const App = () => {
  const [users, setUsers] = useState([]);
  const [user, setUser] = useState({});

  const getAllUsers = async () => {
    const resp = await axios.get(url);
    setUsers(resp.data);
  };

  const getUserById = async () => {
    // Option 1
    const resp = await axios.get(`${url}/7`);
    setUser(resp.data);
    // // Option 2 - Destructuring
    // const { data } = await axios.get(`${url}/7`);
    // setUser(data);
  };

  const addUser = async () => {
    const obj = { name: 'Avi', age: 30 };
    const resp = await axios.post(url, obj);
    console.log(resp.data);
  };

  const updateUser = async () => {
    const obj = { name: 'Ron', age: 40 };
    const resp = await axios.put(`${url}/5`, obj);
    console.log(resp.data);
  };

  const deleteUser = async () => {
    const resp = await axios.delete(`${url}/3`);
    console.log(resp.data);
  };

  return (
    <div>
      {/* Get All */}
      <button onClick={getAllUsers}>Get All</button>
      <ul>
        {users.map((user) => {
          return <li key={user.id}>{user.name}</li>;
        })}
      </ul>
```

```
      <br />
      {/* Get By ID */}
      <button onClick={getUserById}>Get By ID</button> <br />
      Username: {user.username}
      <br /> <br />
      {/* Post */}
      <button onClick={addUser}>Add</button> <br />
      {/* Put */}
      <button onClick={updateUser}>Update</button> <br />
      {/* Delete */}
      <button onClick={deleteUser}>Delete</button> <br />
    </div>
  );
};

export default App;
```

rest api utils

App.js

```
import { useState } from 'react';
import { getAll } from './utils';

const usersUrl = 'https://jsonplaceholder.typicode.com/users';

const App = () => {
  const [users, setUsers] = useState([]);

  const getAllUsers = async () => {
    const { data } = await getAll(usersUrl);
    setUsers(data);
  };

  return (
    <div>
      <button onClick={getAllUsers}>Get All Users</button>
      <ul>
        {users.map((user) => {
          return <li key={user.id}>{user.email}</li>;
        })}
      </ul>
    </div>
  );
};

export default App;
```

Utils.js

```
import axios from 'axios';

const getAll = (url) => axios.get(url);

const getById = (url, id) => axios.get(`${url}/${id}`);
// const getById = (url, id) => axios.get(url + '/' + id)

const addItem = (url, obj) => axios.post(url, obj);

const updateItem = (url, id, obj) => axios.put(`${url}/${id}`,
obj);

const deleteItem = (url, id) => axios.delete(`${url}/${id}`);

export { getAll, getById, addItem, updateItem, deleteItem };
```

EXE4.2

App.js

```
import { useState } from 'react';
import ChildComp from './ChildComp';

const App = () => {
  const [persons, setPersons] = useState([]); נייצר סטייט של מערך
אנשים
  const addPerson = (newPer) => {את הפונק הזו נשלח לילד
    setPersons([...persons, newPer]);הפונק מקבלת אדם ומוסיפה למערך
שלושת הנקודות נועדו למנוע דריסה של שאר האנשים
  };
  return (
    <div style={{ backgroundColor: 'yellow', width: '500px' }}>
      <h1>Parent Component</h1>
      <ul>
        {persons.map((per, index) => {per נרוץ על המערך שכל איבר
          return (
            <li key={index}>שניתן לכל שורה key זה ה
              {per.name} is {per.age} years old, lives in
{per.city} and s.he is{' '}
בסוגריים הריקים הגדרנו רווח
              {per.isAdult ? 'an' : 'not an'} adult
זהו תנאי סימו שאלה שאם הוא true החלק הראשון מתקיים
            </li>
          );
        })}
      </ul>
      <ChildComp callback={addPerson} />
    </div>
  );
};
```

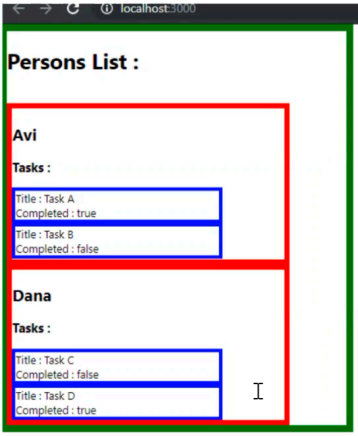
```
export default App;
```

ChildComp

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

const ChildComp = (props) => {
  const [name, setName] = useState('');
  const [age, setAge] = useState('');
  const [city, setCity] = useState('');
  const [isAdult, setIsAdult] = useState(false);
נגדיר את כל המשתנים

  const handleData = () => {הפונקציה הזו תיצור את האובייקט ותעיבר
אותו לאבא
    const obj = { name, age, city, isAdult };כך נגדיר אובייקט
    props.callback(obj);
  };

  return (
    <div style={{ backgroundColor: 'gray' }}>
      <h3>Child Component</h3>
אין צורך ליצור פונקציה לכל משתנה שתכניס את האינפוט למשתנה המצב שלו
לכן נשתמש בפונקציה אנונימית ללא שם

      Name: <input type='text' onChange={(e) =>
setName(e.target.value)} />
      <br />
      Age: <input type='number' onChange={(e) =>
setAge(e.target.value)} />
      <br />
      City:{' '}
      <select defaultValue='' onChange={(e) =>
setCity(e.target.value)}>
        <option value='Haifa'>Haifa</option>
        <option value='Eilat'>Eilat</option>
        <option value='Afula'>Afula</option>
      </select>
      <br />
      Is Adult?{' '}
      <input type='checkbox' onChange={(e) =>
setIsAdult(e.target.checked)} />***************
      <br />
      <button onClick={handleData}>Add</button> כפתור זה יפעיל את
הפונק שתבנה ותשלח את האובייקט לאבא
    </div>
  );
};

export default ChildComp;
```

Lession 3

ex3_6



App.js

```
import Persons from './Persons';

const App = () => {
  return (
    <div>
      <Persons />
    </div>
  );
};

export default App;
```

Persons.js

```
import { useState } from 'react';
import Person from './Person';

const Persons = () => {
  const [persons, setPersons] = useState([
    {
      name: 'Avi',
      tasks: [
        { title: 'Task A', completed: true },
        { title: 'Task B', completed: false },
      ],
    },
     {
      name: 'Dana',
      tasks: [
        { title: 'Task C', completed: false },
        { title: 'Task D', completed: true },
      ],
    },
  ]);

  return (
```

```
    <div style={{ border: '7px solid green', width: '400px' }}>
      <h1>Persons List:</h1>
      {persons.map((per, index) => {id נשתמש באינדקס שאין
        return <Person key={index} person={per} />;
      })}פרסון יקבל אינדקס כי הוא איבר ראשי ברפיטר
    </div>persons זה אובייקט במערך per
  );
};

export default Persons;
```

Person.js

```
import Task from './Task';

const Person = ({ person }) => {במקום לרשום פרופס ניקח ישר את פרסון
  return (
    <div style={{ border: '6px solid red', width: '300px' }}>
      <h3>{person.name}</h3>
      <h4>Tasks:</h4>
      {person.tasks.map((task, index) => {
        return <Task key={index} task={task} />;
נחזיר קומפוננטה של טאסק ובפרופס ניתן את טאסק
      })}
    </div>
  );
};

export default Person;
```

Task.js

```
import React from 'react';

const Task = ({ task }) => { במקום לרשום פרופס ניקח ישר את טאסק
  return (
    <div style={{ border: '5px solid blue', width: '200px' }}>
      Title: {task.title} <br />
      Completed: {task.completed.toString()}
toString הופכת את המשתנה הבולאני למחרוזת
    </div>
  );
};

export default Task;
```

forms

App.js

כאשר נלחץ על הכפתור בform כל האינפוטים וכל הסטייטים בדף יתאפסו לכן נשתמש ב- preventDefault

```
import { useState } from 'react';אין באמת צורך לדעת forms אבל נעבור
על זה
import axios from 'axios';

const usersUrl = 'https://jsonplaceholder.typicode.com/users';

const App = () => {
  const [fname, setFname] = useState('');
  const [lname, setLname] = useState('');

  const handleSubmit = async (e) => {
    e.preventDefault(); מונע מהסטייטים להתאפס
    if (fname) {
      const obj = { firstName: fname, lastName: lname };
      const resp = await axios.post(usersUrl, obj);
      console.log(resp.data);
    } else {
      alert('First Name is mandatory!');
    }
  };

  return (
    <div>
      <form onSubmit={handleSubmit}> ברגע שנלחץ על כפתור מסוג
הפונקציה תופעל submit
        First Name:{' '}
        <input type='text' onInput={(e) =>
setFname(e.target.value)} /> <br />
        Last Name:{' '}
        <input type='text' onInput={(e) =>
setLname(e.target.value)} /> <br />
        <button type='submit'>Send</button>submit כפתור מסוג
      </form>
    </div>
  );
};

export default App;
```

ex5_1

1.

Create a component with a textbox and a button. Entering a user id in the textbox and pressing the button, will present the user's name & email pulled from

https://jsonplaceholder.typicode.com/users by the user id.

If it's name start with "E", also pulled (and present in bullets) his todo's titles

From https://jsonplaceholder.typicode.com/todos

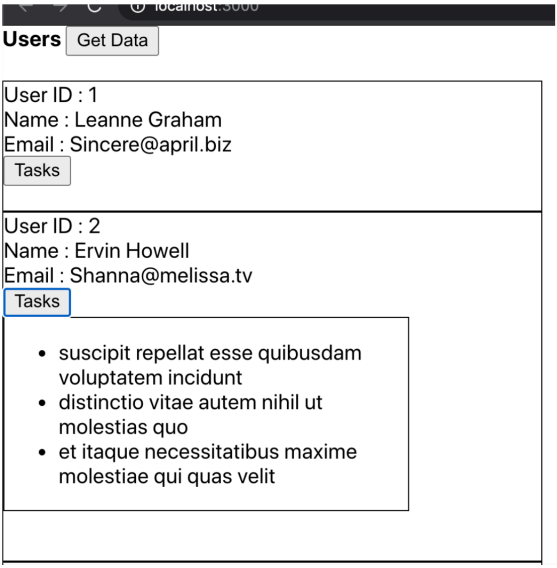
App.js

```
import { useState } from 'react';
import axios from 'axios';

const usersUrl = 'https://jsonplaceholder.typicode.com/users';
const todosUrl = 'https://jsonplaceholder.typicode.com/todos';

const App = () => {
  const [id, setId] = useState('');
  const [user, setUser] = useState({ name: '', email: '' });
  const [todosTitles, setTodosTitles] = useState([]);

  const getUser = async () => {id פונקציה שמקבלת יוזר על ידי
    // // Option 1
    // const resp = await axios.get(`${usersUrl}/${id}`);
    // console.log(resp.data);
    // setUser({ name: resp.data.name, email: resp.data.email });

    // Option 2 - Destructuring
    const { data: user } = await axios.get(`${usersUrl}/${id}`);
    setUser({ name: user.name, email: user.email });

    if (user.name.startsWith('E')) {
      const { data: userTodos } = await axios.get(`${todosUrl}?
userId=${id}`);
      const titles = userTodos.map((todo) => todo.title);
      setTodosTitles(titles);
    } else {
      setTodosTitles([]);
```

```
    }
  };

  return (
    <div>
      User ID: <input type='number' onInput={(e) =>
setId(e.target.value)} />
      <button onClick={getUser}>Get User</button>
      <br /> <br />
      Name: {user.name} <br />
      Email: {user.email} <br />
      <br />
      <ul>
        {todosTitles.map((title, index) => {
          return <li key={index}>{title}</li>;
        })}
      </ul>
    </div>
  );
};

export default App;
```

ex5_3

3. Create a Users component that render “User” child components as follows :



app.js

```
import Users from './Users';

const App = () => {
  return (
    <>
      <Users />
    </>
  );
};

export default App;
```

Users.js

```
import { useState } from 'react';
import { getAll } from './utils';
import User from './User';

const usersUrl = 'https://jsonplaceholder.typicode.com/users';

const Users = () => {
  const [users, setUsers] = useState([]);

  const getUsers = async () => {
    const { data } = await getAll(usersUrl);
    setUsers(data);
  };

  return (
    <>
      <strong>Users</strong> <button onClick={getUsers}>Get Data</
button>
      <br /> <br />
      {users.map((user) => {
        return <User key={user.id} user={user} />;
      })}
    </>
  );
};

export default Users;
```

Utils.js

```
import axios from 'axios';

const getAll = (url) => axios.get(url);

const getUserTasks = async (id, amount) => {
  const todosUrl = 'https://jsonplaceholder.typicode.com/todos';
  const { data: userTodos } = await getAll(`${todosUrl}?userId=$
{id}`);
  const topTodos = userTodos.slice(0, amount);חתוך את הכמות המבוקשת
  return topTodos;
};

export { getAll, getUserTasks };
```

user.js

```
import { useState } from 'react';
import { getUserTasks } from './utils';
import Task from './Task';

const User = ({ user }) => {
  const [tasks, setTasks] = useState([]);

  const showTasks = async () => {פונקציה שמביאה את הטאסקים
    const topTodos = await getUserTasks(user.id, 3);
    setTasks(topTodos);
  };

  return (
    <>
      User ID: {user.id} <br />
      Name: {user.name} <br />
      Email: {user.email} <br />
      <button onClick={showTasks}>Tasks</button>
      <ul>
        {tasks.map((task) => {
          return <Task key={task.id} taskTitle={task.title} />;
        })}
      </ul>
    </>
  );
};

export default User;
```

Task.js

```
import React from 'react';

const Task = ({ taskTitle }) => {
  return (
    <>
      <li>{taskTitle}</li>
    </>
  );
};

export default Task;
```

Lession 4

component lifecycle

App.js

useEffect מבצע פעולות אוטומטיות כתלות למשהו שקורה בקומפונט(לדוגמא תעשה משהו בכל פעם שהקומפוננט מתרנדר

```
import { useState, useEffect } from 'react';
import axios from 'axios';

const url = 'https://jsonplaceholder.typicode.com/users';

const App = () => {
  const [users, setUsers] = useState([]);
  const [counter, setCounter] = useState(0);

  // Will run after every render
  useEffect(() => {ירוץ אחרי כל פעם שהקומפונטנט מתרנדר
    console.log('Hello World');
  });

  // Will run once - at the component creation (Mounting)
  useEffect(() => { נשאיר את התלויים ריקים
    console.log('At mounting');
    // Option 1 - async-await
    const fetchData = async () => {
      const { data } = await axios.get(url);
      setUsers(data);
    };
    fetchData();
    // // Option 2 - then
    // axios.get(url).then(({ data }) => setUsers(data));
  }, []); // [] - Dependency List

  // Will run only if one or more from the dependency list updates
  useEffect(() => {
    console.log('Counter:', counter);כולם ירוצו פעם אחת לפחות כאשר
הדף נוצר
  }, [counter]); ירוץ רק אם אחד או יותר מהתלויים יתעדכן

  return (
    <div>
      <h1>Hello World</h1>
```

```
      User: {users[0]?.name} נשתמש בסימן שאלה לבדוק האם האובייקט
קיים מכיוון שאם איננו קיים תיווצר שגיאה
      <br /> <br />
      Counter: {counter}
      <br />
      <button onClick={() => setCounter(counter + 1)}>Add 1</
button>
    </div>
  );
};

export default App;
```

ex7_1

App.js

```
import { useState } from 'react';
import ChildComp from './ChildComp';

const App = () => {
  const [userId, setUserId] = useState('');

  return (
    <div>
      User ID:{' '}
      <input type='number' onInput={(e) =>
setUserId(e.target.value)} />
      <br />
      <ChildComp userId={userId} />
    </div>
  );
};

export default App;
```

ChildComp.js

```
import { useState, useEffect } from 'react';
import axios from 'axios';

const todosUrl = 'https://jsonplaceholder.typicode.com/todos';

const ChildComp = ({ userId }) => {
```

```
  const [taskTitles, setTaskTitles] = useState([]);

  useEffect(() => {
    const fetchData = async () => {
      const { data: userTodos } = await axios.get(
        `${todosUrl}?userId=${userId}`
      );
       const top5titles = userTodos.slice(0, 5).map((todo) =>
todo.title);
      setTaskTitles(top5titles);
    };
    fetchData();
  }, [userId]);

  return (
    <div>
      <ul>
        {taskTitles.map((title, index) => {
          return <li key={index}>{title}</li>;
        })}
      </ul>
    </div>
  );
};

export default ChildComp;
```

routing

npm i react-router-dom

index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import App from './App';

import { BrowserRouter } from ’react-router-dom‘;
app את שיעטוף browserrouter את נייבא

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <BrowserRouter>
    <App />כך זה יראה
  </BrowserRouter>
);
```

App.js

```
import { Routes, Route, Link } from 'react-router-dom';
נייבא את כל הספריות החשובות
import Homepage from './pages/Homepage';
import About from './pages/About';
import Contact from './pages/Contact';
import ContactEmail from './pages/ContactEmail';
import ContactPhone from './pages/ContactPhone';
import Products from './pages/Products';
import Product from './pages/Product';
import User from './pages/User';
import UserData from './pages/UserData';

const App = () => {
  return (
    <div>
      <h1>Welcome to React Router</h1>
                        נייצר לינקים שיעבירו אותנו לדף אחר ע"י לחיצה
                       לינק לא מרפרש את הדף ומוחק את הסטייט כמו href
      <Link to='/'>Homepage</Link> <br />
      <Link to='/about'>About page</Link> <br />
      <Link to='/contact'>Contact page</Link> <br />
      <Link to='/products'>Products page</Link> <br />
      <Link to='/user'>User page</Link> <br />
      <br />
      <Routes> route ובפנים נכריז על כל routes נפתח
        <Route path='/' element={<Homepage />} /> בדף הראשי נטען את
homepage
        <Route path='/about' element={<About />} />

        {/* Nested Routes */}
        <Route path='/contact' element={<Contact />}> עמוד בתוך
(יש לי ראוט פותח וראוט סוגר בניגוד לראוט שפותח וסוגר את עצמו) עמוד
הראוט הראשון הפותח הוא הראוט הראשי
          <Route path='email' element={<ContactEmail />} />
אין צורך לתת סלש לאימייל הוא מקבל אוטומטית
          <Route path='phone' element={<ContactPhone />} />
        </Route>הראוט הסוגר

        {/* Dynamic Routing - Params */}
העברת פרמטרים בלינק עצמו לפי מאסטר דיטייל
המסטר הוא הדף עם כל המוצרים
הדיטטיל הוא הדף עם המוצר היחיד וכל המידע עליו
        <Route path='/products' element={<Products />} />
        <Route path='/product/:id/:name' element={<Product />} />
שנראה משהו מהתבנית הזו נעבור לדף זה

        {/* Session Storage */}   העברת מידע באמצעות סשיין סטורג'
        <Route path='/user' element={<User />} />
        <Route path='/user-data' element={<UserData />} />
      </Routes>
    </div>
  );
};
```

```
export default App;
```

FOLDER PAGES

homepage.js

```
import { Link, useNavigate } from 'react-router-dom';
נועד לבצע מעבר מדף לדף ואין צורך לייבא את הדף אליו usenavigate hook
                        נרצה לעבור מכיוון שבנינו אינסטנט של navigate
const Homepage = () => {
  const navigate = useNavigate();navigate אינסטנט של

  return (
    <div style={{ backgroundColor: 'red', width: '400px', height:
'400px' }}>
      <h1>Homepage</h1>
      {/* Option 1 */} מוגבל רק ללינק
      <Link to='/about'>About page</Link> <br />
      {/* Option 2 */} ניתן להשתמש בכל פעולה שנרצה
      <button onClick={() => navigate('/about')}>About Page</
button>
    </div>
  );
};

export default Homepage;
```

Product.js

```
import { useParams } from 'react-router-dom';
כדי למשוך את המידע ששלחנו בכתובת עצמה

const Product = () => {
  const { id, name } = useParams();נמשוך את המידע

  return (
    <div>
      <h2>Watch Page</h2>
      ID: {id} <br />
      Name: {name}
    </div>
  );
};
```

```
export default Product;
```

Products.js

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

const Products = () => {
  return (
    <div>
      <h1>Products Page</h1>
      <Link to='/product/123/Watch'>Watch page</Link> <br />
    </div>
  );
};

export default Products;
```

User.js

```
import { useState } from 'react';
import { useNavigate } from 'react-router-dom';

const User = () => {

  const [name, setName] = useState('');
נגדיר סטייט של ניים כמו שאנחנו מכירים
  const navigate = useNavigate();
ניצור אינסטנט של נביגייט כדי לעבור עמוד מתוך עמוד

  const next = () => { נפעיל את הפונקציה נקסט
    sessionStorage['name'] = name;נשים בסשיין את השם
    navigate('/user-data');נעבור לדף אחר
  };

  return (
    <div style={{ backgroundColor: 'salmon', width: '400px',
height: '400px' }}>
      Name: <input type='text' onInput={(e) =>
setName(e.target.value)} />
      <br />
      <button onClick={next}>User Data</button>
    </div>
  );
};

export default User;
```

UserData.js

```
import React from 'react';

const UserData = () => {
  return (
    <div>
      User Data: {sessionStorage['name']}מתוך הסשיין סטורג ניגש לשם
    </div>
  );
};
export default UserData;
```

Contact.js

```
import { Link, Outlet } from 'react-router-dom';
כאשר נשתמש בנסטד ראוטס נהיה חייבים להשתמש בקומפוננט אאוטלט
const Contact = () => {
  const info = {
    name: 'My Company',
  };

  return (
    <div style={{ backgroundColor: 'pink', width: '400px', height:
'400px' }}>
      <h1>Contact Us</h1>
      <Link to='email'>By Email</Link> <br />
אם נשים סלש לפני האימייל זה יהפוך להיות יחסית לרוט הראשי ולא ביחס
לקונטקט
      <Link to='phone'>By Phone</Link> <br />
      {/* Option 1 */}
      {/* <Outlet /> */}
      {/* Option 2 - with context */}
נריץ את אאוטלט עם אינפורמצינ כלשהי שנרצה להעביר
פה הגדרנו למעלה אובייקט בשם info עם פרופרטי של שם
כך בעצם נעביר מידע מאבא לבן
המידע יעבור לכל קומפוננט שמקונן שם
      <Outlet context={info} /> איפה שאני מריץ את האאוטלט שם ירוץ
הקומפוננט המקונן של קונטקט
    </div>
  );
};

export default Contact;
```

ContactPhone.js

```
import React from 'react';

const ContactPhone = () => {
  return (
    <div style={{ backgroundColor: 'brown', width: '300px', height:
'200px' }}>
```

```
      <h1>Phone</h1>
      050-555-5555
    </div>
  );
};

export default ContactPhone;
```

ContactEmail.js

```
import { useOutletContext } from 'react-router-dom';
כדי להשתמש במידע שהעברנו מהאאוטלט צריך את useOutletContext

const ContactEmail = () => {

  const info = useOutletContext(); ניצור אינסטנס שלו

  return (
    <div style={{ backgroundColor: 'purple', width: '300px',
height: '200px' }}>
      <h1>Email</h1>
      our.email@ggl.com
      <br /> <br />
      Name: {info.name} כך נשתמש בו
    </div>
  );
};
export default ContactEmail;
```

About.js

```
import React from 'react';

const About = () => {
  return (
    <div style={{ backgroundColor: 'green', width: '400px', height:
'400px' }}>
      <h1>About Page</h1>
    </div>
  );
};
export default About;
```

ex8_3

Create a "wizard application" which consists of 4 stages :

Stage 1 : "Welcome page" with a button "Start" that will navigate to stage 2

Stage 2 : A display with a FORM containing 2 text boxes for providing "First name" & "Last Name" and a button "Next" that will navigate to stage 3 only if the "First Name" which is mandatory has been supplied.

Stage 3 : A display with a drop down for selecting the city and a button "Next" that will navigate to the last stage (stage 4)

Stage 4 : A display that will present all the data provided so far.

App.js

```
import { Routes, Route } from 'react-router-dom';
import Stage1 from './pages/Stage1';
import Stage2 from './pages/Stage2';
import Stage3 from './pages/Stage3';
import Stage4 from './pages/Stage4';

const App = () => {
  return (
    <div>
      <Routes>
        <Route path='' element={<Stage1 />} />
        <Route path='stage2' element={<Stage2 />} />
        <Route path='stage3' element={<Stage3 />} />
        <Route path='stage4' element={<Stage4 />} />
      </Routes>
    </div>
  );
};

export default App;
```

PAGES FOLDER

Stage1.js

```
import { useNavigate } from 'react-router-dom';

const Stage1 = () => {
  const navigate = useNavigate();

  return (
    <div>
      <h4>Welcome</h4>
      <button onClick={() => navigate('/stage2')}>Start</button>
    </div>
  );
};

export default Stage1;
```

Stage2.js

```
import { useState } from 'react';
import { useNavigate } from 'react-router-dom';

const Stage2 = () => {
  const [user, setUser] = useState({ fname: '', lname: '' });
  const navigate = useNavigate();

  const handleChange = (e) => {
    const { name, value } = e.target;
    setUser({ ...user, [name]: value });
  };

  const handleSubmit = (e) => {
    e.preventDefault();
    sessionStorage['fname'] = user.fname;
    sessionStorage['lname'] = user.lname;
    navigate('/stage3');
  };

  return (
    <div>
      <form onSubmit={handleSubmit}>
        First Name:{' '}
        <input type='text' name='fname' onChange={handleChange}
required />
        <br />
        Last Name: <input type='text' name='lname'
onChange={handleChange} />
        <br />
        <button type='submit'>Next</button>
      </form>
    </div>
  );
};

export default Stage2;
```

Stage3.js

```
import { useState } from 'react';
import { useNavigate } from 'react-router-dom';

const Stage3 = () => {
  const [city, setCity] = useState('');
  const navigate = useNavigate();

  const handleChange = (e) => {
    setCity(e.target.value);
  };

  const next = () => {
    sessionStorage['city'] = city;
    navigate('/stage4');
  };

  return (
    <div>
      City:{' '}
      <select defaultValue='' onChange={handleChange}>
        <option value='' disabled>
          Choose a City
        </option>
        <option value='Haifa'>Haifa</option>
        <option value='Afula'>Afula</option>
        <option value='Eilat'>Eilat</option>
      </select>
      <br /> <br />
      <button onClick={next}>Next</button>
    </div>
  );
};

export default Stage3;
```

Stage4.js

```
const Stage4 = () => {
  return (
    <div>
      <strong>First Name:</strong> {sessionStorage['fname']} <br />
      <strong>Last Name:</strong> {sessionStorage['lname']} <br />
      <strong>City:</strong> {sessionStorage['city']}
    </div>
  );
};

export default Stage4;
```

Lession 5

ex8_1_2

App.js

```
import { Routes, Route } from 'react-router-dom';
import Users from './pages/Users';
import User from './pages/User';
import Posts from './pages/Posts';
import Todos from './pages/Todos';

const App = () => {
  return (
    <div>
      <h1>Master-Details</h1>
      <Routes>
        <Route path='/' element={<Users />} />
        <Route path='/:id' element={<User />}>
          <Route path='posts' element={<Posts />} />מקונן
          <Route path='todos' element={<Todos />} />מקונן
        </Route>
      </Routes>
    </div>
  );
};

export default App;
```

utils.js

```
import axios from 'axios';

const getAll = (url) => axios.get(url);

const getItem = (url, id) => axios.get(`${url}/${id}`);

// ex8_2
const getUserItems = async (url, userId) => {
  const { data } = await getAll(`${url}?userId=${userId}`);
  const titles = data.map((item) => item.title);
  return titles;
};

export { getAll, getItem, getUserItems };
```

PAGES FOLDER

User.js

```
import { useState, useEffect } from 'react';
import { useParams, Link, Outlet } from 'react-router-dom';
import { getItem } from '../utils';

const usersUrl = 'https://jsonplaceholder.typicode.com/users';

const User = () => {

  const { id } = useParams();
  const [user, setUser] = useState({});

  useEffect(() => {
    const fetchData = async () => {
      const { data } = await getItem(usersUrl, id);תמשוך את הנתונים
עם הid
      setUser(data);
    };
    fetchData();
  }, [id]); id -תרוץ שוב כאשר יש שינוי ב

  return (
    <div>
      <h3>{user.username}'s Details</h3>
      <strong>Name:</strong> {user.name} <br />
```

```
      <strong>Email:</strong> {user.email} <br /> אם אין מייל הוא מתעלם
      <strong>City:</strong> {user.address?.city} <br />אם אין אדרס וגם סיטי תהיה שגיאה כי אדרס בכלל לא קיים
      <br />
      {/* ex8_2 */}
      <Link to='posts'>Posts</Link> <Link to='todos'>Todos</Link>
      <Outlet />
    </div>
  );
};

export default User;
```

Users.js

```
import { useState, useEffect } from 'react';
import { Link } from 'react-router-dom';
import { getAll } from '../utils';

const usersUrl = 'https://jsonplaceholder.typicode.com/users';

const Users = () => {

  const [users, setUsers] = useState([]);

  useEffect(() => {יפעל כאשר הדף יעלה ולא יפעל שוב
    const fetchData = async () => { מביא את כל היוזרים ושומר
      const { data } = await getAll(usersUrl);
      setUsers(data);
    };
    fetchData();נפעיל את הפונקציה
  }, []);

   return (
    <div>
      <h2>Usernames</h2>
      <ul>
        {users.map((user) => {
          return (
            <li key={user.id}> key ניתן לו
              <Link to={`/${user.id}`}>{user.username}</Link> נייצר לינקים עם מאפ שכל קישור יעביר אותנו לדף אחר עי איידי
בגלל שאנחנו צריכים לתת מחרוזת ב- to נשתמש בבק טיק או ב- toString להפוך את המספר למחרוזת
            </li>
          );
        })}
      </ul>
    </div>
  );
};

export default Users;
```

Todos.js

```
import { useState, useEffect } from 'react';
import { useParams } from 'react-router-dom';
import { getUserItems } from '../utils';

const todosUrl = 'https://jsonplaceholder.typicode.com/todos';

const Todos = () => {
  const { id } = useParams();idעדיין ניתן למשוך את ה
  const [todos, setTodos] = useState([]);

  useEffect(() => {
    const fetchData = async () => {
      const titles = await getUserItems(todosUrl, id);
      setTodos(titles);
    };
    fetchData();
  }, [id]);,תעדכן את הקומפוננט כאשר האיידי משתנה

  return (
    <div>
      <h4>Todos:</h4>
      <ul>
        {todos.map((title, index) => {
          return <li key={index}>{title}</li>;
        })}
      </ul>
    </div>
  );
};

export default Todos;
```

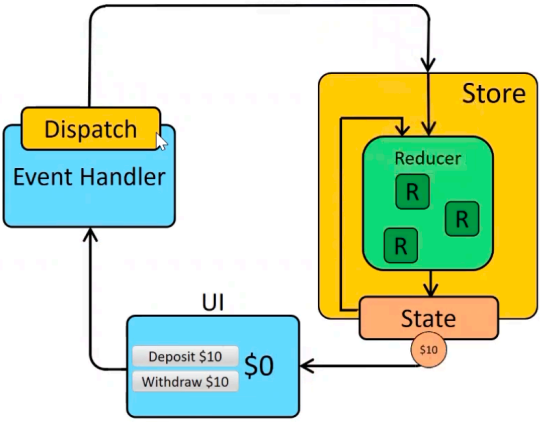
Posts.js

```
import { useState, useEffect } from 'react';
import { useParams } from 'react-router-dom';
import { getUserItems } from '../utils';

const postsUrl = 'https://jsonplaceholder.typicode.com/posts';

const Posts = () => {
  const { id } = useParams();
  const [posts, setPosts] = useState([]);
```

```
  useEffect(() => {
    const fetchData = async () => {
      const titles = await getUserItems(postsUrl, id);
      setPosts(titles);
    };
    fetchData();
  }, [id]);

  return (
    <div>
      <h4>Posts:</h4>
      <ul>
        {posts.map((title, index) => {
          return <li key={index}>{title}</li>;
        })}
      </ul>
    </div>
  );
};

export default Posts;
```

redux

כאשר נרצה להעביר מידע בין קומפוננטות שהן לא אבא ובן
כולם יעבדו מול ה- store שיהיה מאין דאטה בייס לכל
הקומפוננטות
ה- reducer מעדכן לי את הסטייט עי action שהוא מקבל
האקשן אומר לרדוסר איך הוא רוצה לעדכן את הסטור וכל פעם
שמשהו מתעדכן כל הקומפוננט שלוקח ממנו מידע מתעדכן
נפתח תקיית redux ששם יהיה הקובץrootReducer והוא יהיה
הראשי ויאגד לתוכו עוד רדוסרים אם יש
כל פונקציה בקובץ היא סוג של רדוסר שמקבל שני
פרמטרים(הסטייט הנוכחי והאקשיין)



App.js

```
import CounterViewer from './components/CounterViewer';
import CounterChanger from './components/CounterChanger';

const App = () => {
  return (
    <div>
      <CounterViewer />
      <CounterChanger />
    </div>
  );
};

export default App;
```

index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import App from './App';

import { legacy_createStore as createStore } from 'redux'; זה כבר
ישן לא משתמשים בזה ובגלל זה הלגאסי
זה החדש יותר toolkit

import { Provider } from 'react-redux';
מידלוואר שנקרא פרוביידר שתפקידו לספק רת הסטור לכל האפליקציה שלי
import rootReducer from './redux/rootReducer';

const store = createStore(rootReducer);  ניצור סטור ונספק לו את
הרדוסר

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <Provider store={store}>ניצור סטור
    <App />
  </Provider>
);
```

REDUX FOLDER

rootReducer.js

```
const initialValue = { המשתנה ההתחלתי שניתן לסטייט מכיוון שבהתחלה
הוא ריק
  counter: 0,
};

// state - current state
// action - { type: 'WHAT_TO_DO', [payload: value] }
const applyCounterChanger = (state = initialValue, action) => {
  switch (action.type) {
    case 'INCREMENT':
      return { ...state, counter: state.counter + action.payload };
נשתמש בספרייד אופרייטור כדי לעדכן רק את הcounter במידה ויש עוד
משתנים באובייקט ולקאונטר נוסיף את המשתנה הנוכחי counter.state ועוד
מה שקיבלנו מהאקשיין
    case 'DECREMENT':
      return { ...state, counter: state.counter - action.payload };

    default:
      return state;במידה והוא לא מצא קייס. נכון
  }
};

export default applyCounterChanger;
```

COMPONENTS FOLDER

CounterChanger.js

```
import { useDispatch } from 'react-redux'; על מנת לתקשר עם הסטור
ולשלוח את סוג הרדוסר

const CounterChanger = () => {


  const dispatch = useDispatch(); נגדיר את דיספץ'
שהוא זה שפונה לרוט

  const increment = () => {
    dispatch({ type: 'INCREMENT', payload: 1 });
  };

  const decrement = () => {
    dispatch({ type: 'DECREMENT', payload: 1 });
  };

  return (
    <div
      style={{ backgroundColor: 'cyan', width: '200px', textAlign:
'center' }}
    >
      <h2>Counter Changer</h2>
      <button onClick={increment}>+</button>
      <button onClick={decrement}>-</button>
    </div>
  );
};

export default CounterChanger;
```

CounterViewer.js

```
import { useSelector } from 'react-redux'; ככה נמשוך את המידע
מהסטור

const CounterViewer = () => {

  const counter = useSelector((state) => state.counter);
נמשוך את המידע מהסטור ונשים בקאונטר
מקבל את כל הסטייט useselector(כלומר ממש את כל המשתנים בסטור ומחזיר
את counter.state בלבד

  return (
    <div
      style={{ backgroundColor: 'gold', width: '200px', textAlign:
'center' }}
```

```
    >
      <h2>Counter Viewer</h2>
      {counter}
    </div>
  );
};

export default CounterViewer;
```

Lession 6

Redux

index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';

import {createStore} from 'redux'
import {Provider} from 'react-redux'

//import {appReducer} from './Ex9_1/ex9_1_reducer'
import {appReducer} from './Ex9_2/ex9_2_reducer'


const appStore = createStore(appReducer)


const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <Provider store={appStore}>
    <App />
    </Provider>

);

// If you want to start measuring performance in your app, pass a
function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/
CRA-vitals
reportWebVitals();
```

Demo1.js

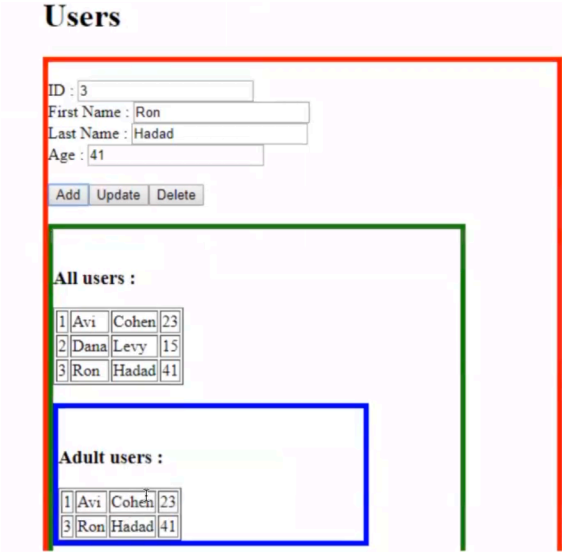
```
import Button from '@mui/material/Button';

function Demo1Comp() {
  return (
    <div className="App">
            <Button color="secondary" onClick={() =>
alert("Hello")} variant="contained">Contained</Button>
    </div>
  );
}
export default Demo1Comp;
```

App.js

```
import logo from './logo.svg';
import './App.css';
import UsersEditorComp from './Ex9_1/UsersEditor';
import ProductsHostComp from './Ex9_2/ProductsHost';
import Demo1Comp from './Demo1_MaterialUI/Demo1';

function App() {
  return (
    <div className="App">
  <Demo1Comp />
< UsersEditorComp />
```

הזה יפעיל את הקומפוננטות הראשיות של כל appה
תרגיל אך כל אחת בנפרד ולא שלושתן ביחד
usersEditor בתרגיל הראשון יפעיל רק את

```
< ProductsHostComp />
    </div>
  );
}

export default App;
```



Ex9_1

ex9_1_reducer.js

```
export function appReducer(state = { users : [] },action)
{נגדיר את הרדוסר שהסטייט הראשוני שלו הוא מערך ריק של יוזרים
    switch(action.type)
    {

        case "LOAD":
            return {...state, users : action.payload}
בחלק הראשון של הריטרן נחזיר סטייט עם שלוש נקודות מה שאומר שזה כל מה
שיש בסטור ללא שינוי ובחלק השני נחזיר את יוזרס עם המערך שקיבלנו

        case "ADD":
            return {...state, users : [...state.users,
action.payload]}
בחלק הראשון של הריטרן נחזיר סטייט עם שלוש נקודות מה שאומר שזה כל מה
שיש בסטור ללא שינוי ובחלק השני נחזיר יוזר אחד שקיבלנו
אך נוסיך גם ליוזרס שלוש נקודות כדי לא לשנות את שאר היוזרים

        case "UPDATE":

        let arr = […state.users];נמשוך את כל היוזרים למערך
        let index = arr.findIndex(x => x.id == action.payload.id);
נמצא את האינדקס הנכון
        if(index >= 0)
        {
            arr[index] = action.payload נחליף את היוזר באינדקס
ביוזר חדש ונכניס למערך הזמני
        }

        return {...state, users : arr}נחליף את המקור במערך הזמני



        case "DELETE":

            let arr2 = [...state.users];
            let index2 = arr2.findIndex(x => x.id ==
action.payload);
            if(index2 >= 0)
            {
                arr2.splice(index2,1)
            }

            return {...state, users : arr2}

        default:
            return state
    }
}
```

UsersEditor.js

```
import { useEffect, useState } from "react";
import { useDispatch } from "react-redux";
import UsersTableComp from "./UsersTable";
import axios from 'axios';
הקומפוננטה האבא הגבוהה ביותר שבה ניתן  לערוך יוזרים
function UsersEditorComp() {

  const dispatch = useDispatch() כדי להשתמש חובה להכריז עליו

  useEffect(() => כאשר היא עולה נמשוך מהשרת את כל היוזרים
  {
    async function getUsers()
    {
      let resp = await axios.get("https://
jsonplaceholder.typicode.com/users");
      let users = resp.data;

      let finalUsers = users.map(user =>
                        {נערוך את היוזרים שקיבלנו באופן שמתאים
לאובייקט מסוג יוזר כפי שאנחנו נרצה שהוא יראה
                          let obj = {};ניצור אובייקט ריק
                          obj.id = user.id;id נכניס לו
                          let arr = user.name.split(" ")
נפרק את השם שמגיע צמוד לשם פרטי ומשפחה עי ספליט על הרווח
                          obj.fname = arr[0];
                          obj.lname = arr[1];
                          obj.age = 18;

                          return obj
                        })

        dispatch({type : "LOAD", payload : finalUsers})נפעיל את
load אקשן עם המידע שבמערך אובייקטים מסוג משתמש

    }
    getUsers();נפעיל את הפונקציה
  },[])

 const [user, setUser] = useState({id : 0, fname: '', lname : '' ,
age : 0}) נייצור סטייט ליוזר

 const add = () => פונקציית add שתפעיל אקשיין add ותשלח יוזר
 {
  dispatch({type : "ADD", payload : user})
 }
 const update = () =>
```

```
 {
  dispatch({type : "UPDATE", payload : user})

 }
 const deleteUser = () =>
 {
  dispatch({type : "DELETE", payload : user.id})

 }

  return (
    <div style={{width : "600px", border: "solid 4px red"}}>

      <h2>Users Editor</h2>

      ID : <input type="text" onChange={e => setUser({...user, id :
+e.target.value}) } />  <br/>
      First Name : <input type="text" onChange={e =>
setUser({...user, fname : e.target.value}) } />  <br/>
      Last Name : <input type="text" onChange={e =>
setUser({...user, lname : e.target.value}) } />  <br/>
      Age : <input type="text" onChange={e => setUser({...user, age
: +e.target.value}) } />  <br/>
כל אינפוט יכניס את המידע לסטייט מסוג פרסון
    <button onClick={add}>Add</button>
    <button onClick={update}>Update</button>
    <button onClick={deleteUser}>Delete</button>

    <br/>
    <br/>
      <UsersTableComp />

    </div>
  );
}

export default UsersEditorComp;
```

UsersTable.js

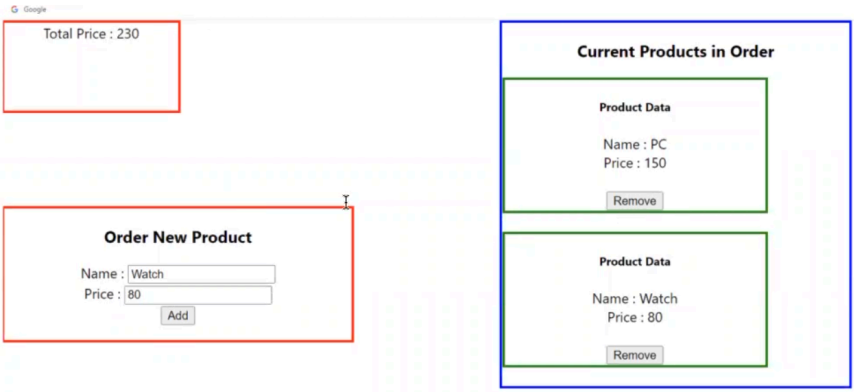
```
import { useSelector } from "react-redux";
import AdultUsersTableComp from "./AdultUsersTable";

function UsersTableComp() {

  const storeData = useSelector(state => state) נמשוך את כל המידע
מהסטור

  return (
    <div style={{width : "400px",border: "solid 4px green"}}>
      <h2>Users Table</h2>

      <table border={1}>
        <thead>
          <tr><th>ID</th><th>First Name</th><th>Last Name</
th><th>Age</th></tr>
        </thead>
        <tbody>
          {
            storeData.users.map(item => הדאטה שמשכנו מהסטור נקודה
יוזרס
              {
                return <tr key={item.id}>
                  <td>{item.id}</td>
                  <td>{item.fname}</td>
                  <td>{item.lname}</td>
                  <td>{item.age}</td>
                </tr>
              })
          }
        </tbody>
      </table>  <br/><br/>

      <AdultUsersTableComp />
    </div>
  );
}

export default UsersTableComp;
```

AdultUsersTable.js

בדיוק אותו דבר כמו הקודם רק עם תנאי על המשתמשים

```
import { useSelector } from "react-redux";

function AdultUsersTableComp() {

  const storeData = useSelector(state => state)

  return (
    <div style={{width : "300px",border: "solid 4px blue"}}>
      <h2>Adult Users Table</h2>

      <table border={1}>
        <thead>
          <tr><th>ID</th><th>First Name</th><th>Last Name</
th><th>Age</th></tr>
        </thead>
        <tbody>
          {
            storeData.users.filter(x => x.age >= 18).map(item =>
              {18נעשה פילטר ואז map כדי לפלטר את המשתמשים הגדולים מ
                return <tr key={item.id}>
                  <td>{item.id}</td>
                  <td>{item.fname}</td>
                  <td>{item.lname}</td>
                  <td>{item.age}</td>
                </tr>
              })
          }
        </tbody>
      </table>  <br/><br/>

    </div>
  );
}

export default AdultUsersTableComp;
```

Ex9_2



ex9_2_reducer.js

```
export function appReducer(state = [] ,action) פה הסטייט הוא מערך
ריק ולא ג'ייסון
{
    switch(action.type)
    {
        case "ADD":
            return [...state, action.payload]

        case "DELETE":
            let arr = […state];מחיקה לפי שם המוצר

            let index = arr.findIndex(x => x.name ==
action.payload);

            if(index >= 0)
            {
                arr.splice(index,1)
            }

            return arr

        default:
            return state
    }
}
```

ProductsHost.js

הקומפוננט הראשי

```
import AddProductsComp from "./AddProducts";
import PriceComp from "./Price";
import ProductsComp from "./Products";

function ProductsHostComp() {
  return (
    <div className="App">

      <div style={{width: "49%", float:"left"}} >
        <PriceComp />  <br/> <br/>
        <AddProductsComp />נשים את שני אלה בשמאל
      </div>

      <div style={{width: "49%", float:"right"}} >
        <ProductsComp />את זה בימין
      </div>



    </div>
  );
}

export default ProductsHostComp;
```

AddProducts.js

```
import { useState } from "react";
import { useDispatch } from "react-redux";

function AddProductsComp() {

  const disptach = useDispatch();

  const [product, setProduct] = useState({name : '', price : 0})

  const add = () =>
  {
    disptach({type : "ADD", payload : product})
  }

  return (
    <div style={{width : "500px", border: "solid 4px red"}}>
    <h2>Add Product</h2>

    Name : <input type="text" onChange={e =>
setProduct({...product, name : e.target.value}) } />  <br/>
    Price : <input type="text" onChange={e =>
setProduct({...product, price : +e.target.value}) } />  <br/>

    <button onClick={add}>Add</button>
    </div>
```

```
  );
}

export default AddProductsComp;
```

Price.js

```
import { useEffect, useState } from "react";
import { useSelector } from "react-redux";

function PriceComp() {

  const [price, setPrice] = useState(0); נגדיר את המחיר ל0

  const storeData = useSelector(state => state);נמשוך את כל המידע
מהסטור

  useEffect(() =>נפעיל את היוס אפקט כל פעם שיש שינוי בסטור
  {וככה כל פעם נעדכן את הסכום
    let total = 0;
    storeData.forEach(prod => total += prod.price)
לאחר שמשכנו את כל המוצרים נחשב את הסכום
    setPrice(total);
  },[storeData])

  return (
    <div style={{width : "500px", border: "solid 4px orange"}}>

      <h2>Total Price :  {price} </h2>



    </div>
  );
}

export default PriceComp;
```

Products.js

```
import { useSelector } from "react-redux";
import ProductComp from "./Product";

function ProductsComp() {

  const storeData = useSelector(state => state);נמשוך את כל המערך

  return (
    <div style={{width : "600px", border: "solid 4px green"}}>
    <h2>Products</h2>
    {
      storeData.map((item,index) =>נרוץ על המערך
      {
        return <ProductComp key={index} prodData={item} />
      })בפרופס item נפעיל קומפוננט מוצר ינשלח כל פעם
    }
    </div>
  );
}

export default ProductsComp;
```

Product.js

```
import { useDispatch } from "react-redux";

function ProductComp(props) {

 const dispatch = useDispatch();

  return (
    <div>
    <div style={{width : "400px", border: "solid 4px blue"}}>

      Name : {props.prodData.name} <br/>
      Price : {props.prodData.price} <br/>

      <button onClick={ () => dispatch({type : "DELETE", payload :
props.prodData.name}) }>Remove</button>

    </div>
    <br/>
    </div>
  );
}

export default ProductComp;
```

Firebase

לא עברתי זמן 02:20:00

App.js

```
import firebase from './firebaseApp'
import {useState} from 'react'

function App() {

  const [persons, setPersons] = useState([])
  const [person, setPerson] = useState({})


 const getPersons =async  () =>
 {
   let data = await
firebase.firestore().collection('persons').get()

   let personsData = [];
   data.forEach(doc =>
    {
      let obj = { id : doc.id, name : doc.data().name, age :
doc.data().age, city : doc.data().city};
      personsData.push(obj);
    })

    setPersons(personsData)

 }


 const getPerson =async  () =>
 {
   let per = await
firebase.firestore().collection('persons').doc('QDRvy05DmbGxBMGVNvm
W').get()


      let obj = { id : per.id, name : per.data().name, age :
per.data().age, city : per.data().city};

    setPerson(obj)

 }

 const createPerson =async  () =>
 {
  let obj = {  name : 'Gil', age :30, city : 'Eilat'};


  await firebase.firestore().collection('persons').add(obj)
```

```
   alert('Created')

 }

 const updatePerson =async  () =>
 {
  let obj = {  name : 'Gil1', age :31, city : 'Eilat1'};


  await
firebase.firestore().collection('persons').doc('F8wzRFuiewz1AoUOX00
I').set(obj);

   alert('Updated')

 }


 const deletePerson =async  () =>
 {


  await
firebase.firestore().collection('persons').doc('F8wzRFuiewz1AoUOX00
I').delete();

   alert('Deleted')

 }

  return (
    <div className="App">

      <button onClick={getPersons}>Get Persons</button>  <br/>
      <button onClick={getPerson}>Get Person</button>  <br/>
      <button onClick={createPerson}>Create Person</button>  <br/>
      <button onClick={updatePerson}>Update Person</button>  <br/>
      <button onClick={deletePerson}>Delete Person</button>  <br/>


      {person.name}  <br/>

      <table border={1}>
        {
          persons.map(item =>
            {
              return <tr key={item.id}>
                <td>{item.name}</td>
                <td>{item.age}</td>
                <td>{item.city}</td>
              </tr>
            })
        }

      </table>
    </div>
  );
}
```

```
export default App;
```

FirebaseApp.js

```
// Import the functions you need from the SDKs you need
import firebase from "firebase/app";
import 'firebase/firestore'
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
const firebaseConfig = {
  apiKey: "AIzaSyB3KBjwjKYlMcCn26JVYUiZWO7PMKudg5c",
  authDomain: "fs39-6ace9.firebaseapp.com",
  projectId: "fs39-6ace9",
  storageBucket: "fs39-6ace9.appspot.com",
  messagingSenderId: "73101360529",
  appId: "1:73101360529:web:cb6e66b870329b49c96e30"
};

// Initialize Firebase
firebase.initializeApp(firebaseConfig)

export default firebase
```

Lession 7

App.js

```
import Demo1ParentComp from './Demo1_React.Memo/Demo1Parent';
import Demo3ParentComp from './Demo3_useCallback/Demo3Parent';

function App() {
  return (
    <div className="App">
     <Demo3ParentComp />
    </div>
  );
}

export default App;
```

שיפורי ביצועים React.Memo

נגיד ויש לנו קומפוננט אבא ואנחנו מעדכנים בו משהו וזה גורם לרינדור שלו,אנו לא רוצים שגם כל הבנים שלו יתרנדרו ואפילו ישלחו בקשות לשרתים למרות שלא היה בהם שינוי

Demo1Child.js

```
import React from 'react';
function Demo1ChildComp(props) {
  console.log("child")

  return (
    <div className="App">
      <h3>Child Comp</h3>

      <h3>Child Footer</h3>
    </div>
  );
}
export default React.memo(Demo1ChildComp); אם נשנה את האבא הילד לא
props יתרנדר אלא אם נעביר לו
```

אם האבא שינה סטייט אצלו שהוא גם נשלח לבן רק אז הבן מתרנדר
אם האבא שינה סטייט אחר שלא נשלח לבן אבל הוא שולח סטייט אחר לא יקרה לבן כלום
עדיף להשתמש רק אם ההיררכיה כבדה (עבודה עם שרת ופונקציות מורכבות)

Demo1Parent.js

```
import { useState } from 'react';
import Demo1ChildComp from './Demo1Child';

function Demo1ParentComp() {

  const [counter, setCounter] = useState(0);
  const [text, setText] = useState("A");


  console.log("Parent")

  return (
    <div className="App">
      <h1>Parent Comp</h1>

      <button onClick={() => setCounter(counter+1) }>+</button>
      <button onClick={() => setText(text + "A") }>Change Text</
button>


      <Demo1ChildComp counter={counter} />

      <h1>Parent Footer</h1>
    </div>
  );
}

export default Demo1ParentComp;
```

useMemo

WithOutUseMemo.js
בלי להשתמש

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

function WithOutUseMemoComp() {

  const [number,setNumber] = useState(1);
  const [counter,setCounter] = useState(0);

  const result = someHeavyFunction(number)

  return (
    <div className="App">
      <button onClick={() => setCounter(counter+1) }>+</button>
כפתור שמשנה את הסטייט של הקאונטר אך לא נרצה שבגלל השינוי הזה כל
הקופוננט ירוץ והפונקציה הכבדה תרוץ שוב ללא צורך כי המספר לא השתנה
   <input type="number" onChange={e =>setNumber(e.target.value)} />
הפונקציה הזו מקבלת מספר ושמה בסטייט שלו
מה שגורם לטעינת הקומפוננט מחדש והפעלת הפונקציה הכבדה שמשתמשת במספר
      {result}

    </div>
  );
}

function someHeavyFunction(num)
{
  //Does somethong "Heavy" with that number

  //return value
}

export default WithOutUseMemoComp;
```

WithUseMemo.js

עם השימוש

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

function WithUseMemoComp() {

  const [number,setNumber] = useState(1);
  const [counter,setCounter] = useState(0);

  const result =  useMemo(() => someHeavyFunction(number),[number])
גם פה יש דיפנדנסי רק אם המספר משתנה תריץ את הפונקציה ממה כמו יוס
אפקט
  return (
   <div className="App">
      <button onClick={() => setCounter(counter+1) }>+</button>
כפתור שמשנה את הסטייט של הקאונטר אך לא נרצה שבגלל השינוי הזה כל
הקופוננט ירוץ והפונקציה הכבדה תרוץ שוב ללא צורך כי המספר לא השתנה
   <input type="number" onChange={e =>setNumber(e.target.value)} />
הפונקציה הזו מקבלת מספר ושמה בסטייט שלו
מה שגורם לטעינת הקומפוננט מחדש והפעלת הפונקציה הכבדה שמשתמשת במספר
      {result}

    </div>
  );
}

function someHeavyFunction(num)
{
  //Does somethong "Heavy" with that number

  //return value
}

export default WithUseMemoComp;
```

useCallback

Demo3Parent.js

```
import { useCallback, useState } from "react";
import Demo3ChildComp from "./Demo3Child";

function Demo3ParentComp() {

  console.log('Parent');

  const [counter, setCounter] = useState(0)
  //Instead of....
  /*
 const getDataFromChild = () =>
 {

 }
 */
למרות שהשתמשנו בusememo בילד עדיין בשינוי של הקאונטר בילד נטען שוב
הסיבה היא שהפונקציה שמועברת לילד נוצרת מחדש כל פעם שהאבא מתרנדר
ולכן יש לה מקום חדש בזיכרון והוא עובר לבן אז זה כמו פרופס חדש
usecallback אומר לפונקציה לא לשנות את הכתובת בעת רינדור חדש
 const getDataFromChild = useCallback(() =>
 {

 },[counter])נוכל להוסיף תלויים שיגרמו לפונקציה כן לשנות כתובת אם
הם משתנים



  return (
    <div className="App">
     <h1>Parent Comp</h1>


     <button onClick={() => setCounter(counter+1) }>+</button>

     <Demo3ChildComp callback={ data => getDataFromChild(data)} />

     <h1>Parent Footer</h1>
    </div>
  );
}

export default Demo3ParentComp;
```

Demo3Child.js

```
import React from 'react'

function Demo3ChildComp(props) {

  console.log('Child');

  return (
    <div className="App">
     <h3>Child Comp</h3>

     <button onClick={() => props.callback(“hello from child”)}
>Send data to parent</button>

     <h3>Child Footer</h3>
    </div>
  );
}

export default React.memo(Demo3ChildComp);
```