1.----------# Complete the following tasks:
- the file `index.html`, which contains the text `A simple app` should be delivered from the server as static content (0.5 pts);

---------->>>>>>>>>
app.use(express.static('public'))
<<<<<<<<<<----------

- a button with the id `reload` exists in the page and can be clicked (0.5 pts);
- when the button with the id `reload` is clicked with nothing in the filter, all elements are returned(0.5 pts);
- when the button with the id `reload` is clicked with `red` in the filter, elements with color `red` are returned(0.5 pts);
- when the button with the id `reload` is clicked with a filter color which does not match anything an empty list is returned(0.5 pts); (0.5 pts);

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>A simple app</title>
    <script>

        function clickBtn() {
            let table = document.getElementById('main')
            let filter = document.getElementById('filter')

            if(filter.value.length === 0) {
                fetch('/cars').then(res => res.json()).then(data => {
                    let rows = data.map((e) => `
                    <tr>
                        <td>${e.name}</td>
                        <td>${e.color}</td>
                    </tr>
                    `)
                    table.innerHTML = rows.join()
                })
            }

            if (filter.value.length>0 && filter.value==="red"){
                fetch('/cars?filter=red').then(result => result.json()).then(data
=> {
                    let rows = data.map((e) => `
                        <tr>
                            <td>
                                ${e.name}
                            </td>
                            <td>
                                ${e.color}
                            </td>
                        </tr>
                    `)
                    table.innerHTML = rows.join()
                })
            }
        }


        window.onload = async () => {
```

```
            let btnReload = document.getElementById('reload');
            let filter = document.getElementById('filer')
            let tbl = document.getElementById('main')

            btnReload.addEventListener('click', clickBtn)
        }

    </script>
</head>
<body>
    <input type="text" placeholder="filter" id="filter"/>
    <table id="main"></table>
    <button id="reload">RELOAD</button>
</html>


# Subject 1 (2.5 pts)
# TOPIC: Basic servers and clients

# Given the server `server.js` and the file `index.html` in the `public` directory:

2.------------------# Complete the following tasks:
- the file `index.html`, which contains the text `A simple app` should be delivered
from the server as static content (0.5 pts);
      app.use("/",express.static("./public"))
- a button with the id `del` exists in the page and can be clicked (0.5 pts);
- on page load, all elements are loaded in the table with the id `main` with a `tr`
for each car (0.5 pts);
- when the button with the id `del` is clicked, elements with the name specified in
the `name` text input are deleted (0.5 pts);
- elements with the name selected for deletion no longer appear in the table (0.5
pts);
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>A simple app</title>
    <script>

      window.onload = async () => {
            let btn = document.getElementById('del')
            let name = document.getElementById('name')

            btn.onclick = async () => {
                await deleteCar(name.value)
            }

            let deleteCar = async (name) => {


                fetch('/cars/'+name,{method:'DELETE'})
                .then(response=>response.json())
                .then(data=>{
                  load('')
                })

            }

            let tbl = document.getElementById('main')
```

```
            let load = async () => {
                try{
                    let response = await fetch(`/cars`)
                    let data = await response.json()
                    let rows = data.map((e) => `
                        <tr>
                            <td>
                                ${e.name}
                            </td>
                            <td>
                                ${e.color}
                            </td>
                        </tr>
                    `)
                    console.warn(rows)
                    tbl.innerHTML = rows.join('')
                }
                catch(err){
                    console.warn(err)
                }
            }
           load('')
        }


    </script>
</head>
<body>
   A simple app
   <input type="text" placeholder="name" id="name" />
   <table id="main"></table>
   <button id="del">Delete</button>
</body>
</html>
```

3.------------------
Complete the following tasks:
- `index.html` file should be delivered as static content from the `public` directory. It shoudl contain a `paragraph` element with `Webtech app` text (0.5 pts);
- Buttons with `ids` `load` and `delete` should exist in the `html` page and they are not disabled (0.5 pts);
- Clicking the button with the id `load` should load all the elements from `data.json` file and render them inside the table with id `table` with a `tr` for each element and 3 `tds` for each property (0.5 pts);
- Text input with id `name` should exist in the html page (0.5 pts);
- When pressing the button with the id `delete`, the application should erase the element with `name` property equals with the value introduced in the text input with the id `name` (0.5 pts);

JSON----------
```
[
    {
        "name": "John",
        "surname": "Doe",
```

```
        "age": 21
    },
    {
        "name": "Jane",
        "surname": "Dane",
        "age": 23
    }
]
--------------------------------------------------------------------------------
---


<!DOCTYPE html>
<html>
  <head>
    <title>Webtech</title>
  </head>
  <body>
    <table id="table"></table>
    <button id="load">Load</button>
    <button id="delete">Delete</button>
    <input type="text" id="name" />

    <script>
        window.onload = async () => {
        let table = document.getElementById("table");

        let p = document.createElement("p");
        p.innerText = "Webtech app";
        document.body.appendChild(p);

        let btLoad = document.getElementById("load");

        btLoad.addEventListener("click", async () => {
          try {
            table.innerHTML = ""; // Clear the table before loading new elements
            let response = await fetch("data.json");
            let data = await response.json();
            for (let element of data) {
              let tr = document.createElement("tr");
              let td1 = document.createElement("td");
              let td2 = document.createElement("td");
              let td3 = document.createElement("td");
              tr.setAttribute("id", element.name);
              td1.innerText = element.name;

              td2.innerText = element.surname;
              td3.innerText = element.age;

              tr.appendChild(td1);
              tr.appendChild(td2);
              tr.appendChild(td3);

              table.appendChild(tr);
            }
          } catch (e) {
            console.warn(e);
          }
        });
```

```
        let btDelete = document.getElementById("delete");
        btDelete.addEventListener("click", () => {
          let nameValue = document.getElementById("name");
          let value = nameValue.value;
          let ta = document.getElementById(`${value}`);
          ta.parentNode.removeChild(ta);
        });
      };
          </script>
  </body>
</html>



# Subject 1 (2.5 pts)
# TOPIC: Basic servers and clients



4.-----------------
# Complete the following tasks:
- the static content in public directory is delivered by the server(0.5 pts);
-  `profil.json` has the structure required in the test (0.5 pts)
- the page `index.html` has a first rang title containing the text `Profil
Influencer` (0.5 pts)
- profile details (name, instagram, youtube) are displayed in separate paragraphs
in the div-ul with id=content (0.5 pts)
- the button with will convert instagram followers into milions (ex 5M); this is
executed only on the client side  (0.5 pts)



<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>A simple app</title>
    <script>
            let load = async () => {
                try{
                    let response = await fetch(`/profile.json`)

                    let object = await response.json()
                    let content = ''
                    content+= `
                        <p id="p1"> ${object.name} </p>
                        <p id="p2"> ${object.instagram} </p>
                        <p id="p3"> ${object.youtube} </p>
                    `
                    document.getElementById("content").innerHTML = content;

                }
                catch(err){
                    console.warn(err)
                }
            }

            let convert = () => {
```

```
                let instagram = document.getElementById('p2').innerHTML
                document.getElementById("p2").innerHTML = instagram/10000000 + "M";
            }

            document.addEventListener('DOMContentLoaded', load)
    </script>
</head>
<body>
    <h1>Profil influencer</h1>
    A simple app
    <div id="content"></div>
    <input type="button" value="convert" id="convert" onclick="convert()" />
</body>
</html>
```

```
-------------V3 2020---------------------
SIMPLE:

# Having the class `Queue` from file `index.js` implement the following tasks:
- Class `Queue` should contain a property called `items`, of type `Array` that will
be initialized with an empty array (0.5 pts);
- Implement method `insert` that accepts `element` as an argument, which will be
added in the array, according to the queue's principle;
- The method `insert` will allow only `string` elements to be added into the queue
and will throw an Error with the text `Invalid Type` for other types.
- Implement method `extract` that will return an `element` from the array,
according to the queue's principle;
- If the array is empty and the `extract` method is called, it will throw an Error
with the text `Invalid Operation`;
<!DOCTYPE html>
<html>
  <head>
    <title>Webtech</title>
  </head>
  <body>
    <table id="table"></table>
    <button id="load">Load</button>
    <button id="delete">Delete</button>
    <input type="text" id="name" />

    <script>
            window.onload = async () => {
              let table = document.getElementById("table");

              let p = document.createElement("p");
              p.innerText = "Webtech app";
              document.body.appendChild(p);

              let btLoad = document.getElementById("load");

              btLoad.addEventListener("click", async () => {
                try {
                  let response = await fetch("data.json");
                  let data = await response.json()
                  for (let element of data) {
```

```
                    let tr = document.createElement("tr");
                    let td1 = document.createElement("td");
                    let td2 = document.createElement("td");
                    let td3 = document.createElement("td");
                    tr.setAttribute("id",element.name)
                    td1.innerText = element.name;

                    td2.innerText = element.surname;
                    td3.innerText = element.age;

                    tr.appendChild(td1);
                    tr.appendChild(td2);
                    tr.appendChild(td3);

                    table.appendChild(tr);
                  }
                } catch (e) {
                  console.warn(e);
                }
              });

              let btDelete = document.getElementById("delete");
              btDelete.addEventListener("click",()=>{
                  let nameValue=document.getElementById("name")
                  let value=nameValue.value;
                  let ta=document.getElementById(`${value}`)
                  ta.parentNode.removeChild(ta);
              })
            };
    </script>
  </body>
</html>
```

----------V0 2020----------

Given the server `app.js` and the file `index.html` in the `public` directory:

Complete the following tasks:
- the file `index.html`, which contains the text `A simple app` should be delivered from the server as static content (0.5 pts);
- a button with the id `load` exists in the page and can be clicked (0.5 pts);
- when the button with the id `load` is clicked, a list of cars should be fetched from the server; cars with the color `red` loaded in the table with the id `main` with a `tr` for each car (0.5 pts);
- the table contains a `tr` for each car loaded from the server (0.5 pts);
- only `red` cars are shown (0.5 pts);

```
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>A simple app</title>
    <script>
      window.onload = async () => {
        try {
```

```
          const response = await fetch(`http://localhost:8080/cars`);
          const responseBody = await response.json();
          console.log(responseBody);
          const button = document.createElement("button");
          button.setAttribute("id", "load");
          document.body.appendChild(button);
          let table = document.getElementById("main");
          button.addEventListener("click", () => {
            for (let data of responseBody) {
              if (data.color === "red") {
                console.log(data);
                let tr = document.createElement("tr");
                let td1 = document.createElement("td");
                let td2 = document.createElement("td");
                td1.innerText = data.name;
                td2.innerText = data.color;
                tr.appendChild(td1);
                tr.appendChild(td2);
                table.appendChild(tr);
              }
            }
          });
        } catch (err) {
          console.log(err);
        }
      };
    </script>
  </head>
  <body>
    A simple app
    <table id="main"></table>
  </body>
</html>
```

```
--------------V1 2020--------------
# Complete the following tasks:
- the file `index.html`, which contains the text `A simple app` should be delivered
from the server as static content (0.5 pts);
- a button with the id `reload` exists in the page and can be clicked (0.5 pts);
- when the button with the id `reload` is clicked with nothing in the filter, all
elements are returned(0.5 pts);
- when the button with the id `reload` is clicked with `red` in the filter,
elements with color `red` are returned(0.5 pts);
- when the button with the id `reload` is clicked with  a filter color which does
not match anything an empty list is returned(0.5 pts); (0.5 pts);

<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>A simple app</title>
    <script>
```

```
      window.onload = async () => {
        let btn = document.getElementById("reload");
        let filter = document.getElementById("filter");

        let tbl = document.getElementById("main");

        btn.addEventListener("click", async () => {
          {
            load(filter);
          }
        });

        let load = async (filter) => {
          try {
            const response = await fetch(
              `http://localhost:8080/cars?filter=${filter.value}`
            );
            const responseBody = await response.json();
            const data = responseBody;
            let rows = data.map(
              (e) => `
                        <tr>
                            <td>
                                ${e.name}
                            </td>
                            <td>
                                ${e.color}
                            </td>
                        </tr>
                    `

            );
            tbl.innerHTML = rows.join();
          } catch (err) {
            console.warn(err);
          }
        };
        load("");
      };
    </script>
  </head>
  <body>
    A simple app
    <input type="text" placeholder="filter" id="filter" />
    <table id="main"></table>
    <button id="reload">Reload</button>
  </body>
</html>
```

------V2 2020------

Complete the following tasks:

- the file `index.html`, which contains the text `A simple app` should be delivered from the server as static content (0.5 pts);
- a button with the id `del` exists in the page and can be clicked (0.5 pts);
- on page load, all elements are loaded in the table with the id `main` with a `tr` for each car (0.5 pts);
- when the button with the id `del` is clicked, elements with the name specified in the `name` text input are deleted (0.5 pts);

- elements with the name selected for deletion no longer appear in the table (0.5 pts);

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>A simple app</title>
    <script>
        window.onload = async () => {
            let btn = document.getElementById('del')
            let name = document.getElementById('name')


            btn.onclick = async () => {
                await deleteCar(name.value)
            }

            let deleteCar = async (name) => {
                try{
                              await fetch(`http://localhost:8080/cars/${name}`, {
                                    method : 'delete'
                        })
                        load()
                            }
                            catch(err){
                                  console.warn(err)
                            }
            }

            let tbl = document.getElementById('main')

            let load = async () => {
                try{
                    let response = await fetch(`http://localhost:8080/cars`)
                    let data = await response.json()
                    let rows = data.map((e) => `
                        <tr id= "${e.name}">
                            <td>
                                ${e.name}
                            </td>
                            <td>
                                ${e.color}
                            </td>
                        </tr>
                    `)
                    //console.warn(rows)
                    tbl.innerHTML = rows.join('')
                }
                catch(err){
                    console.warn(err)
                }
            }

            load()
        }
    </script>
</head>
<body>
```

```
    A simple app
    <input type="text" placeholder="name" id="name" />
    <table id=main></table>
    <button id="del">Del</button>
</body>
</html>



------V4 2020-------------
# Given the server `app.js` and the file `index.html` in the `public` directory:

# Complete the following tasks:
- the static content in public directory is delivered by the server(0.5 pts);
-  `profil.json` has the structure required in the test (0.5 pts)
- the page `index.html` has a first rang title containing the text `Profil
Influencer` (0.5 pts)
- profile details (name, instagram, youtube) are displayed in separate paragraphs
in the div-ul with id=content (0.5 pts)
- the button with will convert instagram followers into milions (ex 5M); this is
executed only on the client side  (0.5 pts)


----------------------------------------JSON-----------------------------------

    {
        "name": "influencer",
        "instagram": 1000000,
        "youtube": 2000000
    }
--------------------------------------------------------------------------------

<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>A simple app</title>
    <script>
        window.onload = async () => {
            let btnConvert = document.getElementById('convert')
            let data
            let response

            btnConvert.onclick = () => {
                convert()
            }

            let load = async () => {
                try {
                     response = await fetch(`/profile.json`)
                     data = await response.json();

                    let div = document.getElementById("content");
                    for (let elem in data) {
                        let p = document.createElement('p')
                        p.innerText = data[elem]
                        p.setAttribute('id', elem)
                        div.append(p)
                    }
```

```
                }
                catch (err) {
                    console.warn(err)
                }
            }

            let convert = () => {
                let p=document.getElementById("instagram")
                let value=(parseInt(p.innerText)/1000000)
                let newP=value+"M"
                p.innerText=newP;
            }

            load()
        }


    </script>
</head>
<body>
    <h1>Profil influencer</h1>
    <div id="content"></div>
    <input type="button" value="convert" id="convert" onclick="convert()" />
</body>
</html>
```

```
-----------------------------------------------
2021-------------------------------------------
# Having the files `app.js` and `index.html` from `public` directory:

# Complete the following tasks:
- `index.html` file should be delivered as static content from the `public` directory. It shoudl contain a `paragraph` element with `Webtech app` text (0.5 pts);
- Buttons with `ids` `load` and `delete` should exist in the `html` page and they are not disabled (0.5 pts);
- Clicking the button with the id `load` should load all the elements from `data.json` file and render them inside the table with id `table` with a `tr` for each element and 3 `tds` for each property (0.5 pts);
- Text input with id `name` should exist in the html page (0.5 pts);
- When pressing the button with the id `delete`, the application should erase the element with `name` property equals with the value introduced in the text input with the id `name` (0.5 pts);



<!DOCTYPE html>
<html>
  <head>
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
    <script src="app.js"></script>
  </head>
  <body>
    <p>Webtech app</p>
    <input id="name" type="text">
```

```
    <button id="load">Load</button>
    <button id="delete">Delete</button>
    <table id="table">
      <thead>
        <tr>
          <th>Name</th>
          <th>Age</th>
          <th>City</th>
        </tr>
      </thead>
      <tbody id="data-rows">
      </tbody>
    </table>

    <script>

        const loadButton = document.getElementById("load");
const deleteButton = document.getElementById("delete");
const nameInput = document.getElementById("name");
const table = document.getElementById("table");

loadButton.addEventListener("click", () => {
  fetch("data.json")
    .then(response => response.json())
    .then(data => {
      data.forEach(element => {
        const tr = document.createElement("tr");
        tr.innerHTML = `<td>${element.name}</td><td>${element.age}</td><td>$
{element.city}</td>`;
        table.appendChild(tr);
      });
    });
});

deleteButton.addEventListener("click", () => {
  const name = nameInput.value;
  const rows = table.getElementsByTagName("tr");
  for (let i = 0; i < rows.length; i++) {
    const row = rows[i];
    if (row.children[0].innerHTML === name) {
      table.deleteRow(i);
      break;
    }
  }
});


    </script>
  </body>
</html>
```

Complete the following tasks:

- the file `index.html`, which contains the text `A simple app` should be delivered

from the server as static content (0.5 pts);
- a button with the id `load` exists in the page and can be clicked (0.5 pts);
- when the button with the id `load` is clicked, a list of cars should be fetched from the server; cars with the color `red` loaded in the table with the id `main` with a `tr` for each car (0.5 pts);
- the table contains a `tr` for each car loaded from the server (0.5 pts);
- only `red` cars are shown (0.5 pts);

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>A simple app</title>
    <script>
        document.addEventListener('DOMContentLoaded', (e)=>{
            let but=document.getElementById('load')
            but.addEventListener('click',async(e)=>{
                let prom=await fetch('http:/cars')
                let cars=await prom.json()

                let cont=cars.filter(e=>e.color=='red').map(e=>`<tr><td>$
{e.name}</td></tr>`)
                document.getElementById('main').innerHTML=cont.join('')

            })
        } )
    </script>
</head>
<body>
    A simple app
    <table id=main></table>
    <input type="button" id="load"/>
</body>
</html>
```