

RAPPORT DE STAGE (07/06 AU 11/06)

SOMMAIRE

Préambule.....	2
Lundi 07 juin 2021.....	2
mardi 08 juin 2021	3
mercredi 09 juin 2021	4
jeudi 10 Juin 2021	9
vendredi 11 juin 2021	13

PREAMBULE

Avant toute chose, il est nécessaire d'indiquer que, pour des raisons de confidentialité, certains travaux ne pourront pas être illustrés (indiqués par un *) ou expliqués dans le détail.

LUNDI 07 JUIN 2021

Activité de la journée : Continuation de l'implémentation de ReactJS

En me renseignant sur le ReactJS et son utilisation avec Laravel 8, je suis tombé sur TypeScript. J'ai vu que ce langage de programmation avait pour but de simplifier l'utilisation du Javascript, sa lisibilité et le débogage. J'ai donc suivi un tutoriel¹ pour l'installer mais après avoir discuté avec le maître de stage, il s'est avéré qu'il ne fallait pas l'utiliser.

J'ai rencontré certaines difficultés, liées au fait que certains dossiers comme « src » n'existaient pas et que mon arborescence de dossiers était différente de celle des tutoriels que je regardais.

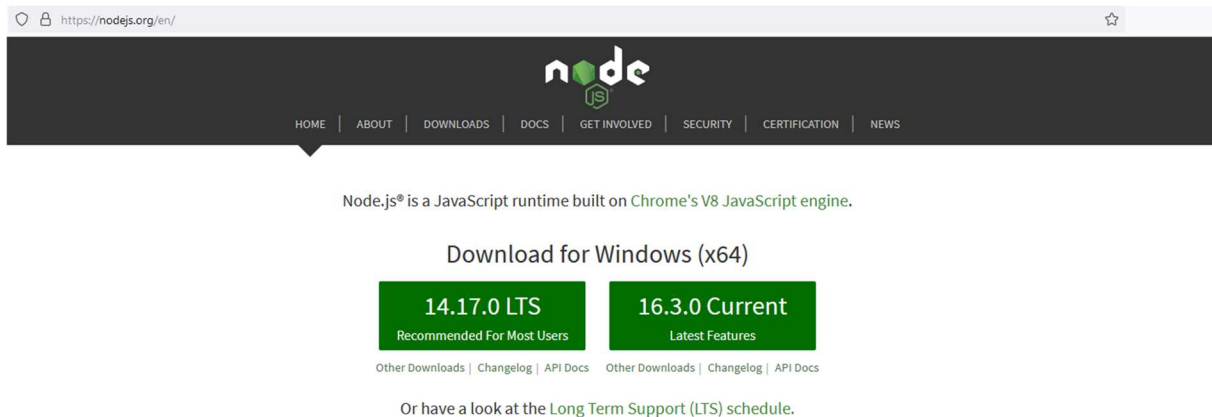
Au final j'ai appris par le maître de stage que j'avais mal interprété la consigne : il ne fallait pas importer du ReactJS dans le projet déjà existant mais créer une nouvelle application ReactJS pour ne pas avoir à se soucier du framework Laravel pour l'instant et apprendre le ReactJS plus aisément.

¹ <https://sadhakbj.medium.com/how-to-setup-reactjs-in-typescript-with-laravel-8-laravel-mix-6-6bf8748befc7>

MARDI 08 JUIN 2021

Activité de la journée : Continuation du développement de l'application en ReactJS

Pour créer l'application ReactJS (avec Visual Studio Code et Laragon), j'ai suivi le guide du site officiel² et je me suis aperçu qu'il fallait que j'installe node.js.



Je me suis rendu le site officiel de node.js et j'ai installé la version recommandée pour la plupart des utilisateurs.

Une fois l'installation terminée j'ai vérifié dans le terminal de Visual Studio Code que cela avait bien fonctionné :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

Testez le nouveau système multiplateforme PowerShell https://aka.ms/pscore6

PS D:\laragon\www\uhreactapp> node -v
v14.17.0
PS D:\laragon\www\uhreactapp> npm -v
6.14.13
Debugger listening on ws://127.0.0.1:57388/716eed78-68ca-43ac-ae30-631169a95d1e
For help, see: https://nodejs.org/en/docs/inspector
Debugger attached.
Waiting for the debugger to disconnect...
PS D:\laragon\www\uhreactapp> 
```

En tapant « node -v » j'ai bien eu la version 14.17.0 et en tapant « npm -v » la version 6.14.13.

J'ai pu ensuite créer mon nouveau projet avec la commande

```
PS D:\laragon\www\uhreactapp> npx create-react-app uhreactapp
```

Pour me familiariser avec les dossiers, fichiers et contenus de ce nouveau projet, j'ai suivi la vidéo de Nord Coders³ sur la création d'une application React où il présente l'arborescence du projet.

² <https://fr.reactjs.org/docs/create-a-new-react-app.html>

³ <https://www.youtube.com/watch?v=nkB0WBQcncE>

MERCREDI 09 JUIN 2021

Activité de la journée : Développement de l'application ReactJS

Afin de comprendre la logique de ReactJS, j'ai décidé de suivre le guide de Nord Coders avec ses 21 vidéos qui présente différents concepts du ReactJS : le JSX, les props, les composants, les hooks, etc.

```
class App extends React.Component {  
  constructor(props) {  
    super(props);  
    this.state = { show: false , title: 'Coucou les amis'};  
  }  
  
  componentDidMount(){  
    this.setState({title: 'Le composant a bien été monté'})  
  }  
  
  componentDidUpdate() {  
    console.log('Composant mis à jour');  
  }  
  
  render() {  
    return <div>  
      <h1 className="text-4xl text-purple-700 my-5">{this.state.title}</h1>  
      <button className="bg-purple-900 text-white rounded py-2 px-3"  
        onClick={() => this.setState({show: !this.state.show}) }  
      >Click!</button>  
      { show ? <Picture /> : null}  
    </div>;  
  }  
}
```

```
export default class Picture extends Component {  
  constructor(props) {  
    super(props);  
    this.state = {timer: null};  
  }  
  
  componentDidMount() {  
    this.setState({  
      timer: setInterval(() => {  
        console.log('Timer appelé');  
      }, 1000)});  
  }  
  
  componentWillUnmount() {  
    console.log('Composant Picture démonté');  
    clearInterval(this.state.timer);  
  }  
  
  render() {  
    return (  
      </img>  
    )  
  }  
}
```

Cela permet de voir l'implémentation d'une classe comme ci-dessus mais aussi une autre façon, celle des fonctions et des composants :

```
1  import React, {useState, useEffect, useRef } from 'react';
2  import Picture from './components/Pictures';
3
4  function App() {
5
6      const [title, setTitle] = useState('Application');
7      const [show, setShow] = useState(false);
8      const isShowInitialize = useRef(false);
9
10     useEffect(() => console.log('Composant app monté'), []);
11     useEffect(() => {
12         if (isShowInitialize.current) {
13             console.log('Show mis à jour')
14         } else {
15             isShowInitialize.current = true;
16         }
17     }, [show]);
18
19     function handleClick() {
20         setShow(!show);
21     }
22
23     return (
24         <div>
25             {console.log('JSX rendered')}
26             <h1 className="text-4xl text-purple-700 my-5">{title}</h1>
27             <button className="bg-purple-900 text-white rounded py-2 px-3"
28                 onClick={handleClick}>Click!</button>
29
30             {show ? <Picture /> : null}
31
32         </div>
33     );
34 }
35
36 export default App;
```

```
1
2 import React, { Component, useEffect, useState, useRef } from 'react'
3 import Picture from './Picture';
4
5
6 export default function Pictures() {
7
8     // const [myTimer, setMyTimer] = useState(null);
9
10    const [images, setImages] = useState([
11        'logo192.png',
12        'logo512.png'
13    ]);
14
15    const [image, setImage] = useState(null);
16
17
18
19    function ImagesComponent() {
20        return images.map((name, index) =>
21            <Picture
22                imageName={name}
23                index={index}
24                key={index}
25                handleRemove={handleRemoveImage}
26            /> );
27    }
28
29    function handleRemoveImage(index) {
30        setImages(images.filter((image, i) => i !== index));
31    }
32
33
34    function handleImageName(event) {
35        setImage(event.target.value);
36    }
37
38    function addImageName() {
39        let newImages = [...images, image];
40        setImages(newImages );
41    }
42
43    useEffect(() => {
44
45        const myTimer = setInterval(() => {
46            console.log('Timer appelé');
47        }, 1000);
```

```

48
49     return () =>clearInterval(myTimer);
50 }, []);
51
52
53
54
55     return (
56         <div className="container mx-auto">
57             <div className="flex items-center justify-between">
58                 <ImagesComponent />
59             </div>
60             <div className="mt-5">
61                 <input type="text" className="border border-gray-600 shadow rounded p-3 mr-2
62                 outline-none" onChange={handleImageName}/>
63                 <button type="submit" className="bg-purple-400 text-white rounded p-3"
64                 onClick={addImageName}>Inscrire le
65                 nom d'une image</button>
66             </div>
67         </div>
68     )
69 }

```

```

1  import React, { useState } from 'react'
2
3  export default function Picture( { imageName, index, handleRemove } ) {
4      const[showCloseButton, setShowCloseButton] = useState(-1);
5
6      return (
7          <div>
8              <div className="relative" key={index}
9                  onMouseEnter={() => setShowCloseButton(index)}
10                 onMouseLeave={() => setShowCloseButton(-1)}
11             >
12                 <button
13                     className={`bg-white w-5 h-5 pb-1 font-thin rounded-xl text-center flex
14                     justify-center items-center absolute text-red-500 right-0 text-sm
15                     focus:outline-none ${index === showCloseButton ? '' : 'hidden'}}`
16                     onClick={() => handleRemove(index)}
17                 >x</button>
18                 <img className="w-40 mx-auto" src={imageName} alt="logo de react"></img>
19             </div>
20         </div>
21     )
22 }
23

```

Aussi, au cours du tutoriel, j'ai dû installer TailwindCSS ⁴ avec :

```
npm install -D tailwindcss@npm:@tailwindcss/postcss7-compat postcss@^7 autoprefixer@^9
```

J'ai aussi dû remplacer certains paramètres dans le fichier package.json de mon projet.

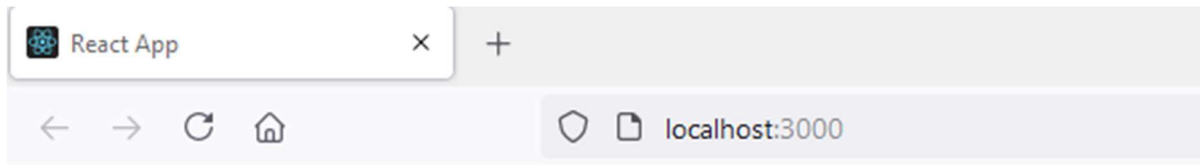
⁴ <https://tailwindcss.com/docs/guides/create-react-app>

```
{  
  // ...  
  "scripts": {  
-    "start": "react-scripts start",  
-    "build": "react-scripts build",  
-    "test": "react-scripts test",  
+    "start": "craco start",  
+    "build": "craco build",  
+    "test": "craco test",  
    "eject": "react-scripts eject"  
  },  
}
```


JEUDI 10 JUIN 2021

Activité de la journée : Continuation du développement de l'application en ReactJS

J'ai pu finir le tutoriel de Nord Coders, ce qui me donne (on démarre l'application avec la commande « npm run start » dans le terminal) :



Application

Click!



Application

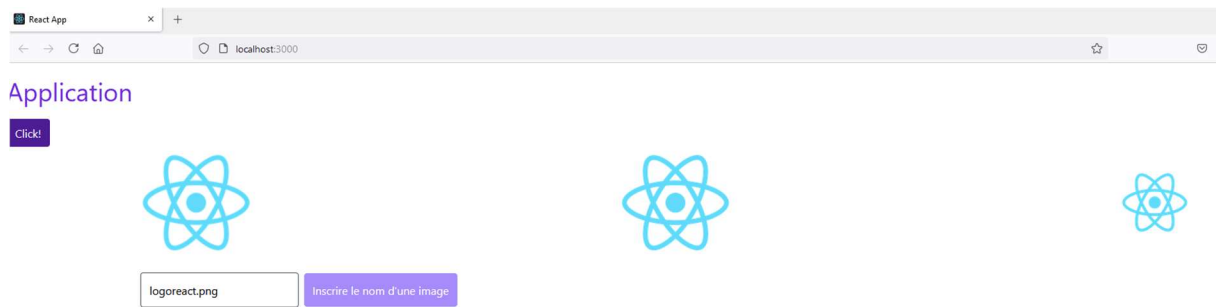
Click!



Application

Click!

En entrant le nom d'une image et en cliquant sur le bouton « Inscrire le nom d'une image » une nouvelle image est ajoutée.



Suite à cela j'ai tenté différentes adaptations du code en suivant d'autres tutoriels^{5 6} que j'ai écrit avec le tutoriel pour créer des formulaires pour addresses, civilstates, etc. Mais cela s'est soldé par plusieurs échecs.

C'est après de longues recherches que j'ai pu me débloquent grâce à ce site⁷.

```

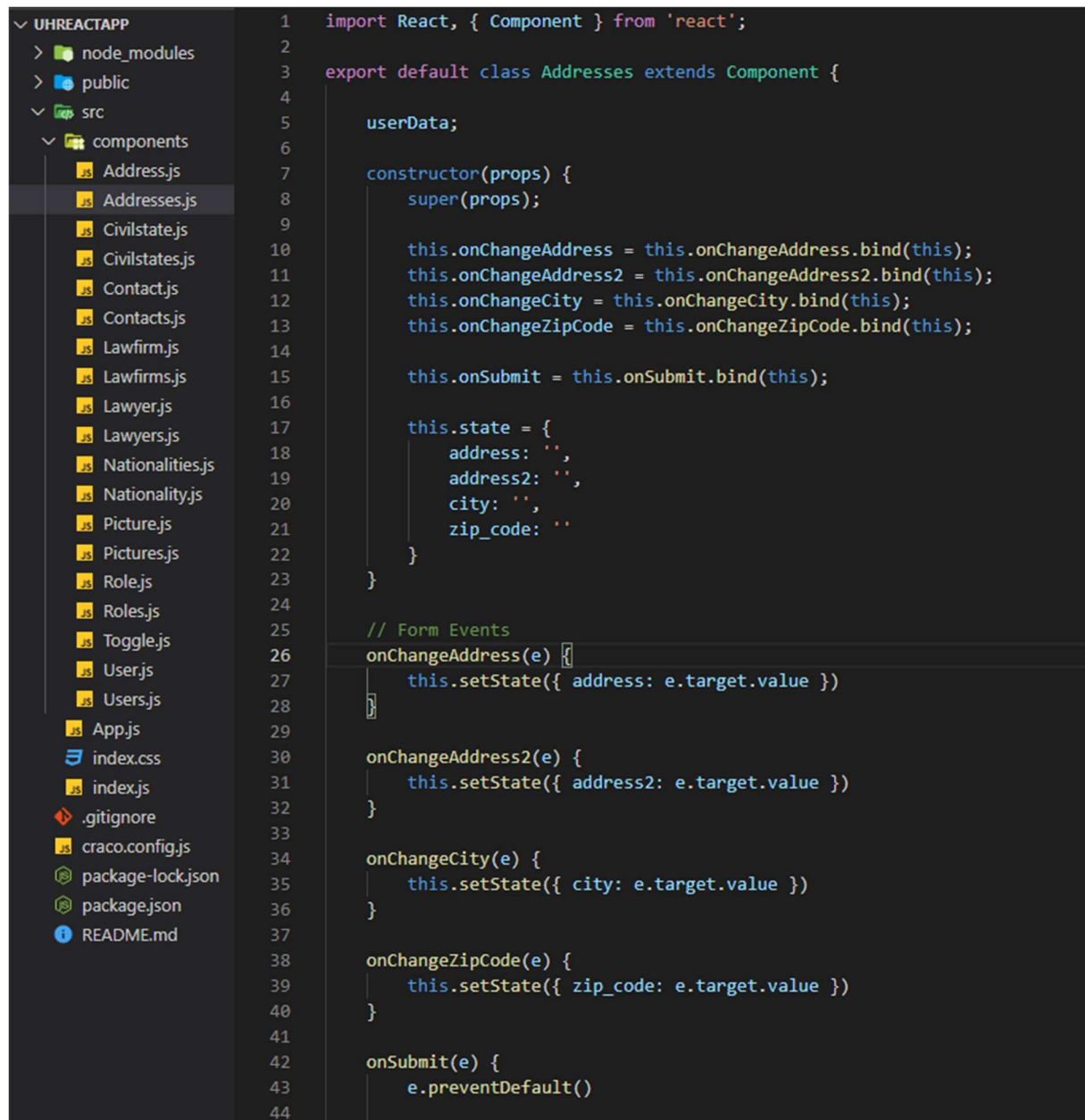
src > App.js > ...
1  import React, {Component} from 'react';
2  import '../node_modules/bootstrap/dist/css/bootstrap.min.css';
3
4  import Toggle from "../components/Toggle";
5
6  import Addresses from '../components/Addresses';
7  import Roles from '../components/Roles';
8
9  class App extends Component {
10
11    constructor(props) {
12      super(props);
13      this.state = { show: false , title: 'Application'};
14    }
15
16    render() {
17      return (
18        <div className="App">
19          <h1>{this.state.title}</h1>
20          <Toggle title="Show Addresses">
21            <Addresses/>
22          </Toggle>
23          <Toggle title="Show Addresses">
24            <Roles />
25          </Toggle>
26        </div>
27      );
28    }
29  }
30
31  export default App;

```

⁵ <https://www.digitalocean.com/community/tutorials/how-to-build-forms-in-react>

⁶ <https://www.golangprograms.com/reactjs-show-hide-component-on-click.html>

⁷ <https://www.c-sharpcorner.com/article/how-to-show-and-hide-component-in-react-application/>



```
1  import React, { Component } from 'react';
2
3  export default class Addresses extends Component {
4
5      userData;
6
7      constructor(props) {
8          super(props);
9
10         this.onChangeAddress = this.onChangeAddress.bind(this);
11         this.onChangeAddress2 = this.onChangeAddress2.bind(this);
12         this.onChangeCity = this.onChangeCity.bind(this);
13         this.onChangeZipCode = this.onChangeZipCode.bind(this);
14
15         this.onSubmit = this.onSubmit.bind(this);
16
17         this.state = {
18             address: '',
19             address2: '',
20             city: '',
21             zip_code: ''
22         }
23     }
24
25     // Form Events
26     onChangeAddress(e) {
27         this.setState({ address: e.target.value })
28     }
29
30     onChangeAddress2(e) {
31         this.setState({ address2: e.target.value })
32     }
33
34     onChangeCity(e) {
35         this.setState({ city: e.target.value })
36     }
37
38     onChangeZipCode(e) {
39         this.setState({ zip_code: e.target.value })
40     }
41
42     onSubmit(e) {
43         e.preventDefault()
44     }
```

```

> OPEN EDITORS      src > components > Addresses.js > Addresses > onChangeAddress
UHREACTAPP
  > node_modules
  > public
  > src
    > components
      Address.js
      Addresses.js
      Civilstate.js
      Civilstates.js
      Contact.js
      Contacts.js
      Lawfirm.js
      Lawfirms.js
      Lawyer.js
      Lawyers.js
      Nationalities.js
      Nationality.js
      Picture.js
      Pictures.js
      Role.js
      Roles.js
      Toggle.js
      User.js
      Users.js
  App.js
  index.css
  index.js
  .gitignore
  craco.config.js

44
45      this.setState({
46          address: '',
47          address2: '',
48          city: '',
49          zip_code: ''
50      })
51  }
52
53  // React Life Cycle
54  componentDidMount() {
55      this.userData = JSON.parse(localStorage.getItem('user'));
56
57      if (localStorage.getItem('user')) {
58          this.setState({
59              address: this.userData.address,
60              address2: this.userData.address2,
61              city: this.userData.city,
62              zip_code: this.userData.zip_code
63          })
64      } else {
65          this.setState({
66              address: '',
67              address2: '',
68              city: '',
69              zip_code: ''
70          })
71      }
72  }
73
74  UNSAFE_componentWillUpdate(nextProps, nextState) {
75      localStorage.setItem('user', JSON.stringify(nextState));
76  }
77

```

```

> OPEN EDITORS      src > components > Addresses.js > Addresses > onChangeAddress
UHREACTAPP
  > node_modules
  > public
  > src
    > components
      Address.js
      Addresses.js
      Civilstate.js
      Civilstates.js
      Contact.js
      Contacts.js
      Lawfirm.js
      Lawfirms.js
      Lawyer.js
      Lawyers.js
      Nationalities.js
      Nationality.js
      Picture.js
      Pictures.js
      Role.js
      Roles.js
      Toggle.js
      User.js
      Users.js
  App.js
  index.css
  index.js
  .gitignore
  craco.config.js

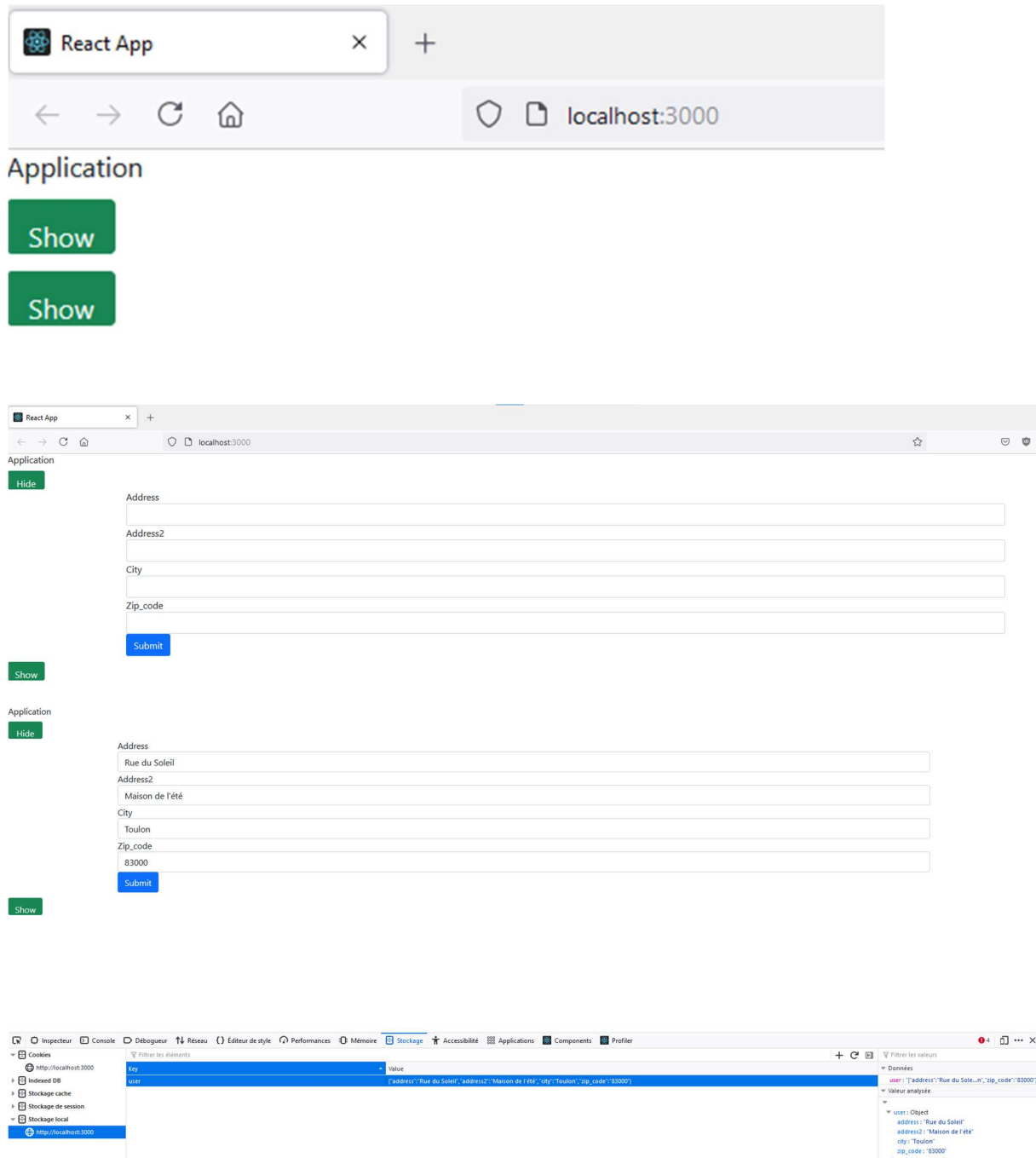
77
78      render() {
79          return (
80              <div className="container">
81                  <form onSubmit={this.onSubmit}>
82                      <div className="form-group">
83                          <label>Address</label>
84                          <input type="text" className="form-control" value={this.state.address} onChange={this.onChangeAddress} />
85                      </div>
86                      <div className="form-group">
87                          <label>Address2</label>
88                          <input type="text" className="form-control" value={this.state.address2} onChange={this.onChangeAddress2}/>
89                      </div>
90                      <div className="form-group">
91                          <label>City</label>
92                          <input type="text" className="form-control" value={this.state.city} onChange={this.onChangeCity} />
93                      </div>
94                      <div className="form-group">
95                          <label>Zip_code</label>
96                          <input type="text" className="form-control" value={this.state.zip_code} onChange={this.onChangeZipCode} />
97                      </div>
98                      <button type="submit" className="btn btn-primary btn-block">Submit</button>
99                  </form>
100              </div>
101          )
102      }
103  }

```

VENDREDI 11 JUIN 2021

Activité de la journée : Continuation du développement de l'application ReactJS

Mon code actuel m'a permis d'avoir le rendu suivant :



Tout ce qui est entré dans les champs sera conservé dans le localStorage, ce qui signifie que rafraîchir la page ne va pas effacer les données rentrées. Cependant la persistance de ces données laisse à désirer car une fois le submit cliqué elles ne sont plus conservées.