

Components	CSS
<pre> import React, { Component } from 'react'; import './Header.css' class Header extends Component { constructor(props) { super(props); this.state = { } } render() { return (<header id="header"> <div> <h1>Typing Speed</h1> </div> {this.props.signout?<div className="signout"> Se deconnecter </div>:''} </header>); } } export default Header; </pre>	<pre> #header{ background-color: #000032; color: white; padding: 10px 5px; } .signout{ text-align: right; } </pre>
<pre> import React, { Component } from 'react' import './Authentication.css' class Authentication extends Component { constructor(props) { super(props); this.state = { nom:'', email:'', password:'' } } render() { const Login = <div className="signbox "> <label>Email</label> <input type="text" value={this.state.email} onChange={(ev)=>{ this.setState({email:ev.target.value}) }}/>
</br> <label>Password</label> <input type="password" value={this.state.password} onChange={(ev)=>{ this.setState({password:ev.target.value}) }}/> <button onClick={()=>{ this.props.onLogin(</pre>	<pre> .signview{ width: 30%; margin: 12px auto; } .signbuttons{ display: flex; justify-content: space-evenly; } .signbuttons span{ background-color: white; color: black; padding: 10px 12px; width: 50%; transition: all 0.2s; } .signbuttons span:hover{ background-color: #474545; color: white; } .signbox{ padding: 20px; text-align:left; } </pre>

```

        this.state.email,
        this.state.password
    )
  }}>Login</button>
</div>

const Signup = <div className="signbox">
  <label>Nom</label>
  <input type="text" value={this.state.nom} onChange={(ev)=>{
    this.setState({nom:ev.target.value})
  }}/>
  <br></br>
  <label>Email</label>
  <input type="text" value={this.state.email} onChange={(ev)=>{
    this.setState({email:ev.target.value})
  }}/>
  <br></br>
  <label>Password</label>
  <input type="password" value={this.state.password} onChange={(ev)=>{
    this.setState({password:ev.target.value})
  }}/>
  <button onClick={()=>{
    this.props.onRegister(
      this.state.nom,
      this.state.email,
      this.state.password
    )
  }}>Register</button>
</div>

let classLogin = this.props.login?' active':''
let classSignup = !this.props.login?' active':''
return (<div className="signview">
  <div className="signbuttons shadow">
    <span className={"pointer" +classLogin}
onClick={()=>{this.props.changeView('login')}}>Se Connector</span>
    <span className={"pointer" + classSignup}
onClick={()=>{this.props.changeView('signup')}}>S'enregistrer</span>
  </div>
  <br></br>
  <div className="shadow">
    {this.props.login?Login:Signup}
  </div>
</div>);
}
}

export default Authentification;

```

```

.shadow{
  box-shadow: 2px 2px 12px #3a3a3a;
}
input{
  width: calc(100% - 23px);
  padding: 5px 10px;
}
button{
  padding: 5px;
  width: 100%;
  margin: 5px 0;
}
.active{
  border-bottom: 5px solid black;
}

```

```

import React, { Component } from 'react'
import './UsersList.css'

```

```

#userTable{
  width: 100%;

```

```

class UsersList extends Component {
  constructor(props) {
    super(props);
    this.state = {
      userList:[],
      searchB:''
    }

    this.getData = this.getData.bind(this)
  }
  componentWillMount()
  {
    this.getData()
  }
  getData()
  {
    fetch("http://localhost:3001/")
    .then(res=>res.json())
    .then(res=>{
      this.setState({
        userList:res
      })
    })
  }
  render() {
    return (<div className="container">
      <h1>List des joueurs</h1>
      <div>
        <input value={this.state.searchB} onChange={(ev)=>{
          this.setState({
            searchB:ev.target.value
          })
        }} type="text" placeholder="Search..." id="usersearch" />
        <table id="userTable">
          <thead>
            <tr>
              <th>Nom</th>
              <th>Email</th>
              <th>Score</th>
            </tr>
          </thead>
          <tbody>
            {this.state.userList.map((e)=>
              {
                let data = <tr key={e._id}>
                  <td>{e.nom}</td>
                  <td>{e.email}</td>
                  <td>{e.score}</td>
                </tr>
              }
            )}
          </tbody>
        </table>
      </div>
    </div>
  )}
  }

```

```

border: 2px solid black;
margin: 8px 0;
border-radius: 8px;
}
#userTable tbody tr td{
  padding: 8px 4px;
}
#userTable tbody tr:nth-child(odd)
{
  background-color: rgb(211, 211,
211);
}
#usersearch{
  border-radius: 5px;
}

```

```

                if(e.score == -1)
                    return;
                if(this.state.searchB === '')
                {
                    return data
                }
                else{
                    if(e.nom.indexOf(this.state.searchB) > -1 ||
e.email.indexOf(this.state.searchB) > -1)
                        {
                            return data
                        }
                    }
                }
            })
        </tbody>
    </table>
</div>
</div>;
}
}

export default UsersList;

```

```

import React, { Component } from 'react'
import './Test.css'
import UsersList from './UsersList';
class Test extends Component {
    constructor(props) {
        super(props);
        this.state = {
            text:'Sed clita amet ipsum dolor no sadipscing ea ut labore. Et dolor elittr tempor
nonumy aliquyam tempor dolor. At elittr.',
            input:'',
            h:0,
            m:0,
            s:0,
            total:0,
            page:'test',
            startTimer:false
        }
        this.changePage = this.changePage.bind(this)
        this.startTimer = this.startTimer.bind(this)
        this.stoptTimer = this.stoptTimer.bind(this)
        this.handelTimer = this.handelTimer.bind(this)
        this.handelTextChange = this.handelTextChange.bind(this)
    }
    changePage()
    {
        this.setState({

```

```

        page: 't'
    })
}
handelTimer()
{
    if(true)
    {
        this.setState({
            s: this.state.s+1,
            total: this.state.total+1
        })
        if(this.state.s>=60)
        {
            this.setState({
                s: 0,
                m: this.state.m+1
            })
        }
        if(this.state.m>=60)
        {
            this.setState({
                s: 0,
                m: 0,
                h: this.state.h+1
            })
        }
    }
}
startTimer(e)
{
    this.handelTextChange(e)
    if(this.state.startTimer)
    {
        return
    }
    this.setState({
        startTimer: true
    })
    let timer = setInterval(()=>{
        if(this.state.startTimer)
            this.handelTimer()
        else
            clearInterval(timer)
    },1000)
}
handelTextChange(e)
{
    this.setState({
        input: e.target.value
    })
}

```

```

      setTimeout(()=>{
        if(this.state.input === this.state.text)
        {
          this.stoptTimer()

        }
      },100)
      console.log(this.state.input)
    }
    stoptTimer()
    {
      if(this.state.startTimer)
      {
        if(this.state.input === this.state.text)
        {

          fetch('http://localhost:3001/stoptimer',{
            headers:{
              'Content-Type':'application/json'
            },
            method:'post',
            body:JSON.stringify({
              id:this.props.userid,
              score:this.state.total
            })
          })
          .then(res=>res.json())
          .then(res=>{
            alert("Your score is "+this.state.total)
            this.setState({
              s:0,
              m:0,
              h:0,
              total:0,
              input:'',
              startTimer:false
            })
          })
        }
      }
    }
  }
  render() {
    const test = <div className="container black">
      <div className="textView"><p>{this.state.text}</p></div>
      <div className="input">
        <textarea value={this.state.input} onChange={this.startTimer} placeholder="Text
ici" className="textarea"></textarea>
      </div>

```

```

        <div className="buttonsvew">
            <button>Reset</button>
            <button onClick={this.changePage} >Scores</button>
            <div>
                {this.state.h<10?'0'+this.state.h:this.state.h}:
                {this.state.m<10?'0'+this.state.m:this.state.m}:
                {this.state.s<10?'0'+this.state.s:this.state.s}
            </div>
        </div>
    </div>
    return (<div>
        {this.state.page=='test'?test:<UsersList/>}
    </div>);
}
}

export default Test;

```

```

import React, { Component } from 'react';
import Header from './Components/Header'
import Auth from './Components/Authentication'
import Test from './Components/Test'
import './App.css'
class App extends Component {
    constructor(props){
        super(props)

        this.state={
            signout:false,
            login:true,
            isLogin:false,
            id:0
        }

        this.onsignouthandler = this.onsignouthandler.bind(this)
        this.loginHandler = this.loginHandler.bind(this)
        this.registerHandler = this.registerHandler.bind(this)
        this.changeAuthView = this.changeAuthView.bind(this)
    }
    onsignouthandler()
    {
        this.setState({
            signout:true
        })
    }
    loginHandler(email,password){
        fetch("http://localhost:3001/login",{
            headers:{
                'Content-Type':'application/json'
            },
            method:'post',

```

```

*{
    margin: 0;
    padding: 0;
}
.App {
    text-align: center;
}

.pointer{
    cursor: pointer;
}
.container{
    width: 30%;
    margin: 12px auto;
    padding: 10px;
}
.container.black{
    background-color: #5f5d5d;
    color: white;
    border: 3px solid black;
}

```

```

        body:JSON.stringify({
            email,password
        })
    })
    .then(res=>res.json())
    .then(res=>{
        if(res.success)
        {
            this.setState({
                isLogin:true,
                id:res.id
            })
        }
        else{
            alert("Mot de passe ou email pas correct")
        }
    })
}
registerHandler(nom,email,password)
{
    fetch("http://localhost:3001/signup",{
        headers:{
            'Content-Type':'application/json'
        },
        method:'post',
        body:JSON.stringify({
            nom,email,password,score:-1
        })
    })
    .then(res=>res.json())
    .then(res=>{
        alert(res.msg)
        window.location.reload()
    })
}
changeAuthView(view)
{
    if(view == 'login')
    {
        this.setState({
            login:true
        })
    }
    else{
        this.setState({
            login:false
        })
    }
}
render() {

```



```

    return (
      <div className="App">
        <Header signout={this.state.signout} onSignout={this.onsignouthandler}/>
        {!this.state.isLogin?
          <Auth login={this.state.login} onLogin={this.loginHandler}
onRegister={this.registerHandler} changeView={this.changeAuthView}/>
          :<Test userid={this.state.id}/>}
        </div>
      );
    }
  }
}

export default App;

```

Exercise 2 :

```

const app = require("express")()
const parser = require("body-parser")
const mongo = require("mongodb").MongoClient
const cors = require('cors')
const port = 3001

app.use(cors())
app.use(parser.json())
app.use(parser.urlencoded({extended: false}))
var db;
mongo.connect("mongodb://127.0.0.1",{useNewUrlParser:true},(err,client)=>{
  if(err) console.log(err)
  else{
    db = client.db("exam")
    app.listen(port,()=>{
      console.log("Start listening")
    })

    app.post('/login',(req,res)=>{
      db.collection('users').find({email:req.body.email,password:req.body.password}).toArray((err,data)=>{
        if(data == null)
        {
          res.send({success:false})
        }
        else
          res.send({success:true,id:data[0]._id})
      })
    })

    app.post('/signup',(req,res)=>{
      db.collection('users').insert(req.body,(err,data)=>{
        if(err){
          res.json({msg:err})
        }
      })
    })
  }
})

```

```

        else{
            res.send({
                msg:'Votre compte a ete crée avec success'
            })
        }
    })
})

app.post('/stoptimer',(req,res)=>{
    db.collection('users').updateOne({_id:req.id},{ $set:{
        score:req.score
    }},(err,data)=>{
        if(err)
            throw err
        if(data)
        {
            res.json({success:true})
        }
        else{
            res.json({success:false})
        }
    })
})

app.get('/',(req,res)=>{
    db.collection('users').find().toArray((err,data)=>{
        if(err)
            throw err
        res.json(data)
    })
})

}

})

```