

Informatique Fondamentale

Utilisation d'Ordinateurs en Réseau

Damien Vallet

Informatique Fondamentale

Utilisation d'Ordinateurs en Réseau

by

Damien Vallet

Etudiant	ID
Damien Vallet	23000113

Enseignant: Philippe Kislin - Duval
Université: Université Paris8

Cover: Cassiopeia A; SNR G111.7-02.1 - image from NASA's James Webb Space Telescope's NIRCam
NASA, ESA, CSA, STScI, Danny Milisavljevic (Purdue University), Ilse De Looze (UGent), Tea Temim (Princeton University) (Modified)

Contents

1 Chapitre 1 - Introduction Générale	1
2 Chapitre 2 - Hyper-Text Markup Language : HTML	3
3 Chapitre 3 - Les Feuilles de Style en Cascade	29
4 Chapitre 4 - Le Protocole HTTP, Architecture Clients et Serveurs Web	50
5 Chapitre 5 - Lire et traiter des informations via une page Web : formulaires, PHP, MySQL	70
6 Chapitre 6 - Le Langage JavaScript	119
7 Chapitre 7 - Internet en dehors du Web	147
A Codes Source du chapitre 2	163
B Codes Source du chapitre 3	172
C Codes Source du chapitre 5 - PHP	181
D Codes Source du chapitre 5 - MongoDB	196
E Codes Source du chapitre 6	211
F Code Source du chapitre 7 - Chatbot	220

1

Chapitre 1 - Introduction Générale

Exercice 1.1

Pour ce chapitre, pas d'exercice à réaliser, si ce n'est de :

- Trouver un hébergeur(gratuit si possible) afin de pouvoir réaliser en ligne les exercices des chapitres suivants.
- Vérifiez que celui-ci vous permette d'accéder à PHP, à au moins une base MySQL (ou équiv) et à son administration (PhpMyAdmin ou équiv), ainsi qu'au protocole FTP.
- Si c'est le cas, inscrivez-vous gratuitement à celui-ci et notez l'URL de votre "première" page dans le PDF.
- Pour vous aider : Contactez les étudiants des années passées pour leur demander quel(s) hébergeur(s) ils ont choisi(s). (par ex : <https://fr.000webhost.com>, <https://www.netlify.com/>, ...)

Exercice 1.1 :

Pour réaliser ce cours nous avons choisi de rester en local et de développer nos connaissances sur le Framework React. L'installation de React et de notre environnement de développement sera détaillée dans les chapitres suivants.

L'intégralité des codes présentés dans ce cours seront disponibles sur notre dépôt GitHub : <https://github.com/IED-L1/uor-partie2-archive>

2

Chapitre 2 - Hyper-Text Markup Language : HTML

Exercice 2.1 (A Rendre)

A l'aide d'un éditeur de Textes (Notepad++, Sublime Text (ai la préférence pour celui-ci), Atom (TB également), BBEdit, Coda, Vim, MS Visual Studio Code (TB), TextMate, Emacs, etc.) :

- Réalisez une page web (à l'ancienne) avec votre CV ou en développant une thématique personnelle (passion(s), loisir(s), sport(s), jeux...).
- La page, écrite en HTMLv5 doit contenir au moins du texte (plusieurs paragraphes, avec de vraies phrases : sujet-verbe-complément à minima), des images (>2), des liens (non cassés)(>2). Les listes, Les tableaux sont encouragés mais facultatifs.
- Ajoutez obligatoirement un titre et des métadonnées descriptives complètes au format de votre choix (Dublin Core, Standard Metatags, Open Graph Data, ...) afin d'augmenter la SEO dans l'entête de votre page (balise HEAD).
- Essayez d'en soigner la mise en page et placez-la sur le serveur d'hébergement que vous avez choisi (cf Chapitre 1).

Dans le PDF, après avoir rappelé l'énoncé de l'exercice :

- Indiquez l'URL de votre page si vous ne l'avez pas déjà fait lors de l'exercice 1.1.
- Faites un C.C (Copier-Coller) du code source HTML complet en ajoutant des commentaires sur vos choix et sur ce que vous avez réalisé — ou non réalisé — et/ou des constructions particulières du HTML que vous avez employées.
- Ajoutez une C.E (Copie d'Ecran) de votre éditeur de textes ainsi qu'une (ou plusieurs selon) C.E(s) de votre page affichée dans votre navigateur.
- Ajoutez enfin une C.E de votre page qui est passée par un validateur et qui ne contient donc aucune erreur.
- Joignez en archive attachée le contenu du Document Root (htdocs) portant le numéro de l'exercice

Attention : Pour que l'exercice soit intéressant, il faut que la page contienne des choses qui vous semblent importantes et qui vous tiennent à cœur. C'est nécessaire pour que se manifeste la tension entre contenu et apparence. Les pages qui seraient remplies de Lorem ipsum ne sont pas recevables...

Important : Il est également possible de réaliser cet exercice, les suivants (et les Bonus) en installant en local votre propre serveur web. Si vous choisissez cette éventualité que vous me préciserez, ajoutez à l'exercice 1.1 les C.E(s) commentées d'installation de votre serveur (Apache (ou lAMP), Nginx (ou LEMP), Caddy, LightHttpd, MonkeyServer, IIS,) ainsi que les commandes unix utilisées.

(Note : Il est tout à fait envisageable d'utiliser des frameworks complets comme NodeJS, Deno (javascript), ... Streamlit, Django, Flask (Python)... Hugo (GO), Jekyl (Ruby) mais aussi Gatsby (React (à tester !)), Sculpin , Laravel (PHP)....)

Exercice 2.2 (Bonus)

Deux éléments à compléter pour un Bonus :

- Ajoutez une courte vidéo (> 1min), en vous présentant en quelques mots (ou dans la thématique retenue) dans votre page au format approprié. Vous pouvez bien évidemment donner un côté humoristique ou ludique à celle-ci si le coeur vous en dit. (Joignez ensuite celle-ci à l'archive et faites-la apparaître dans une C.E dans le PDF)
- Ajoutez des métadonnées au format LD-JSON en plus de celles de l'Exercice 2.1.

Exercice 2.1 :

Pour cet exercice, j'ai réalisé une page web personnelle qui présente mon parcours professionnel et mes compétences en tant que designer sonore, musicien et "wanna be" programmeur. J'ai ajouté des liens vers des projets auxquels j'ai contribué et une vidéo de présentation. J'ai également ajouté des liens vers mes profils LinkedIn, Github, Mastodon et mon adresse email.

Pour enrichir mes connaissances, j'ai choisi de développer cette page (et les suivantes) en local en utilisant le framework **React**.

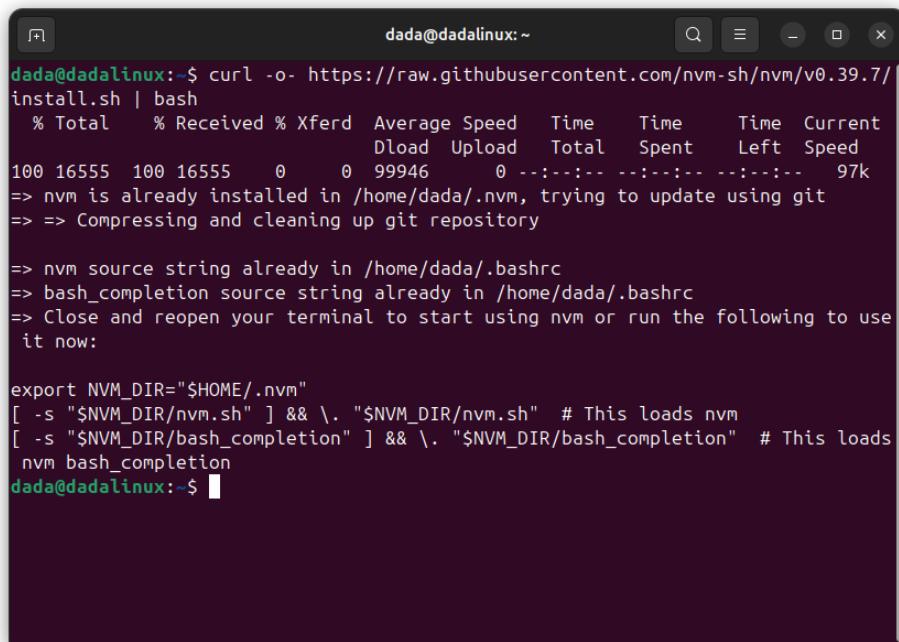
Je vais d'abord décrire la marche à suivre pour installer et faire fonctionner **React** en local.

Installation de React en local

Tout d'abord, nous devons installer **Node.js** qui est un environnement d'exécution JavaScript. **Node.js** inclut également **npm**, un gestionnaire de paquets JavaScript. Pour l'installer, j'ai suivi la page du tutoriel d'installation sur le site officiel de **Node**.

On installe d'abord **nvm** (Node Version Manager) qui permet de gérer plusieurs versions de Node sur un même système.

```
1 # install Node Version Manager (NVM)
2 curl -o https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash
```



```
dada@dadalinux:~$ curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.7/install.sh | bash
  % Total    % Received % Xferd  Average Speed   Time     Time     Current
          Dload  Upload Total   Spent    Left  Speed
100 16555  100 16555    0      0  99946      0 --::-- --::-- --::--  97k
=> nvm is already installed in /home/dada/.nvm, trying to update using git
=> => Compressing and cleaning up git repository

=> nvm source string already in /home/dada/.bashrc
=> bash_completion source string already in /home/dada/.bashrc
=> Close and reopen your terminal to start using nvm or run the following to use
it now:

export NVM_DIR="$HOME/.nvm"
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh" # This loads nvm
[ -s "$NVM_DIR/bash_completion" ] && \. "$NVM_DIR/bash_completion" # This loads
nvm bash_completion
dada@dadalinux:~$
```

Installation de NVM

On redémarre le terminal pour que les changements prennent effet.

Ensuite, on installe **Node.js** avec la commande suivante :

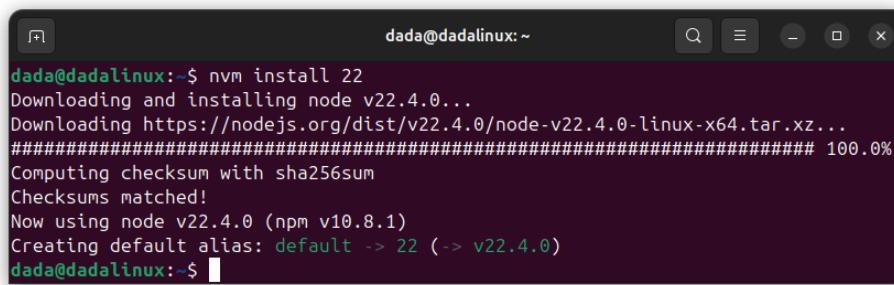
```
1 nvm install 22
```

On vérifie que **Node.js** est bien installé avec la commande suivante :

```
1 node -v # doit afficher la version de Node.js
```

On vérifie que npm (Node Package Manager) est bien installé avec la commande suivante :

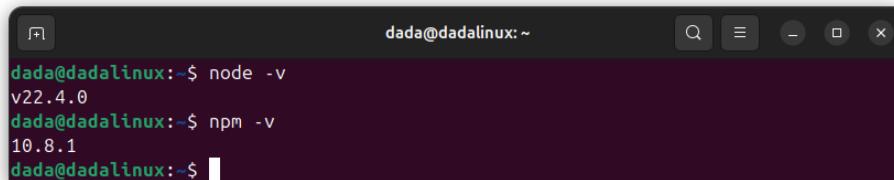
```
1 npm -v # doit afficher la version de npm
```



A screenshot of a terminal window titled "dada@dadalinux: ~". The window shows the command "nvm install 22" being run. The output indicates that Node.js version 22.4.0 is being downloaded and installed, with a progress bar showing 100.0% completion. It also shows that the default alias "default" has been updated to point to version 22.

```
dada@dadalinux:~$ nvm install 22
Downloading and installing node v22.4.0...
Downloading https://nodejs.org/dist/v22.4.0/node-v22.4.0-linux-x64.tar.xz...
#####
Computing checksum with sha256sum
Checksums matched!
Now using node v22.4.0 (npm v10.8.1)
Creating default alias: default -> 22 (-> v22.4.0)
dada@dadalinux:~$
```

Installation de Node.js



A screenshot of a terminal window titled "dada@dadalinux: ~". The window shows the commands "node -v" and "npm -v" being run. The output displays the installed versions: Node.js v22.4.0 and npm 10.8.1.

```
dada@dadalinux:~$ node -v
v22.4.0
dada@dadalinux:~$ npm -v
10.8.1
dada@dadalinux:~$
```

Vérification des versions de Node.js et npm

Ensute, nous avons créer une app React. Pour cela nous avons suivi les instructions du site [Create React App](#) ainsi que du site [React](#).

Pour créer notre application, nous avons ouvert une fenêtre de Terminal et tapé la commande suivante :

```
1 npx create-react-app dv-portfolio
```

```
✓ TERMINAL
dada@dadalinux:/media/dada/9A8604B486049345/IED/L1S1/info_fondamentale/Partie 2$ npx create-react-app dv-portfolio
Need to install the following packages:
  create-react-app@5.0.1
  OK to proceed? (y) y

npm warn deprecated uid-number@0.0.6: This package is no longer supported.
npm warn deprecated rimraf@2.7.1: Rimraf versions prior to v4 are no longer supported
npm warn deprecated inflight@0.6: This module is not supported, and leaks memory. Do not use it. Check out lru-cache if you want a good and tested way to coalesce async requests by a key value, which is much more comprehensive and powerful.
npm warn deprecated fstream-ignore@1.0.5: This package is no longer supported.
npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm warn deprecated fstream@1.0.12: This package is no longer supported.
npm warn deprecated tar@2.2.2: This version of tar is no longer supported, and will not receive security updates. Please upgrade asap.

Creating a new React app in /media/dada/9A8604B486049345/IED/L1S1/info_fondamentale/Partie 2/dv-portfolio.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...

added 1482 packages in 57s
262 packages are looking for funding
  run `npm fund` for details
Installing template dependencies using npm...
added 63 packages, and changed 1 package in 6s
262 packages are looking for funding
  run `npm fund` for details
Removing template package using npm...

removed 1 package, and audited 1545 packages in 3s
262 packages are looking for funding
  run `npm fund` for details
  8 vulnerabilities (2 moderate, 6 high)

To address all issues (including breaking changes), run:
```

Création de l'app React

Cette commande crée un nouveau projet React appelé *dv-portfolio* et installe toutes les dépendances nécessaires.

Pour lancer le serveur de développement, on se déplace dans le dossier du projet et on tape la commande suivante :

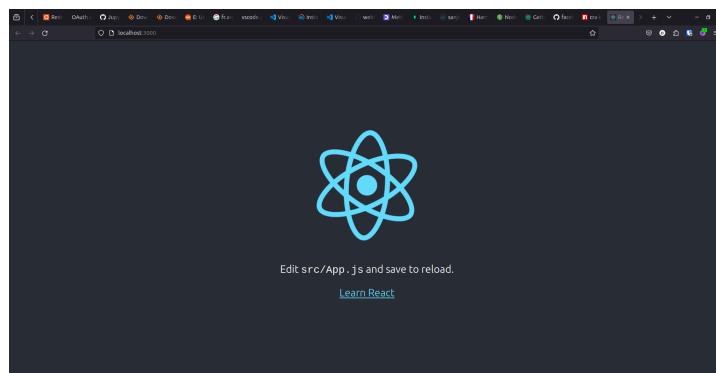
```
1 cd dv-portfolio
2 npm start
```

```
dada@dadalinux:/media/dada/9A8604B486049345/IED/L1S1/info_fondam... Compiled successfully!
You can now view dv-portfolio in the browser.
  Local:          http://localhost:3000
  On Your Network: http://192.168.1.62:3000
Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

Lancement du serveur de développement

Cette commande lance le serveur de développement et ouvre une fenêtre de navigateur, avec l'application React, à l'adresse localhost:3000.

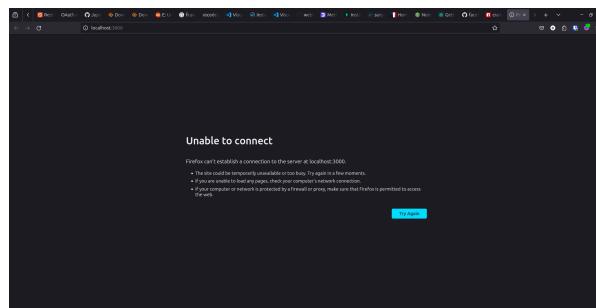


Capture d'écran de l'application React dans le navigateur

Pour arrêter le serveur de développement, on peut taper la combinaison de touches **Ctrl + C** dans la fenêtre de commande.

```
dada@dadalinux:/media/dada/9A8604B486049345/IED/L151/info_fondamentale/Partie 2/dv-portfolio$ 
Compiled successfully!
You can now view dv-portfolio in the browser.
Local:          http://localhost:3000
On Your Network: http://192.168.1.62:3000
Note that the development build is not optimized.
To create a production build, use npm run build.
webpack compiled successfully
^C
dada@dadalinux:/media/dada/9A8604B486049345/IED/L151/info_fondamentale/Partie 2/dv-portfolio$
```

Commande pour arrêter le serveur de développement



Fenêtre du navigateur avec arrêt du serveur.

Sources

- <https://react.dev>
- <https://react.dev/learn/start-a-new-react-project>
- <https://create-react-app.dev/docs/getting-started/>
- <https://nodejs.org/en/download/package-manager/current>
- <https://docs.npmjs.com/downloading-and-installing-node-js-and-npm/>

Fichier App.js :

Une fois notre application créée, nous pouvons commencer à ajouter du contenu.

Dans une application React, tout le contenu est organisé en **composants**. Un composant est un élément de l'interface utilisateur qui peut être réutilisé. En les combinant, on peut créer des interfaces utilisateur complexes. Ces composants sont des fonctions JavaScript qui retournent du code HTML permettant de structurer le contenu de la page.

Le fichier `App.js` est le composant principal de notre application React. Agissant comme une colonne vertébrale, un tronc, il contient tous les autres composants. Pour le moment, il contient deux composants : `NavigationHeader` et `Metadata`.

Pour les utiliser dans le fichier `App.js`, nous devons d'abord les importer puis les ajouter dans la fonction `App`. On les appelle en utilisant des balises HTML.

Voici le code source de notre fichier `App.js` pour le moment :

```

1 // /**
2 * File Name: App.js
3 * Description: composant principal de l'application
4 * Author: Damien Vallet
5 * Version: 1.0
6 * Date: 2024
7 * License: Created for UOR course
8 * Language: JavaScript
9 */
10
11 //Importation de la librairie Bootstrap
12 import "bootstrap/dist/css/bootstrap.css";
13
14 // Importation de la feuille de style
15 import "./stylesheet/style.css";
16
17 // Importation des composants
18 import NavigationHeader from "./NavigationHeader";
19 import Metadata from "./Metadata";
20
21 function App() {
22   return (
23     // une balise div qui encapsule le code HTML
24     <div className="App">
25       {/* composant pour les métadonnées*/}
26       <Metadata />
27
28       {/* composant pour la navigation*/}
29       <NavigationHeader />
30     </div>
31   );
32 }
33
34 // export de la fonction App
35 export default App;

```

1. Dans ce fichier, nous avons importé les composants **NavigationHeader** et **Metadata**.
2. Nous avons créé une fonction **App** qui retourne le code HTML de notre application React.
3. Nous avons ajouté les deux composants importés à la fonction **App** en les encapsulant dans une balise vide.
4. Nous avons exporté la fonction **App** pour pouvoir l'utiliser dans d'autres fichiers.

Gestion des métadonnées

Métadonnées Standards :

Nous avons d'abord ajouté des métadonnées Standard Metatags. Pour se faire nous avons édités le fichier `index.html` dans le dossier public de notre application React.

Le format Standard Metatags est un ensemble de balises html `<meta>` standardisées pour améliorer le référencement (SEO) en permettant aux moteurs de recherche de mieux comprendre et indexer le contenu de la page. Elles sont développées par le W3C.

Nous avons ajouté les métadonnées suivantes :

- Content-Type : le type de contenu de la page
- viewport : la largeur de la page et le niveau de zoom initial
- title : le titre de la page
- description : la description de la page
- keywords : les mots-clés de la page

Voici le code de la partie Standard Metatags de notre fichier `index.html` :

```

1 /* balises meta */
2 <!-- meta tags standard -->
3 <meta httpEquiv="Content-Type" content="text/html; charset=utf-8" />
4 <meta name="viewport" content="width=device-width, initial-scale=1.0" />
5 <title>Damien Vallet - Audio Product Designer - Portofolio</title>
6 <meta name="description" content="Portofolio de Damien Vallet" />
7 <meta
8   name="keywords"
9   content="Damien Vallet, Portofolio, Audio Product Design, Programmation"
10 />

```

Sources

- <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/meta>
- https://www.w3schools.com/tags/tag_meta.asp
- <https://create-react-app.dev/docs/title-and-meta-tags>

Afin de gérer les métadonnées dans notre application React, nous avons créé un fichier `Metadata.js` dans le dossier `src`.

Après avoir recherché comment ajouter des métadonnées dans une application React, nous avons trouvé un article sur le site web de la librairie React Helmet qui explique comment ajouter des métadonnées à une application React.

On peut installer Helmet depuis un terminal avec la commande suivante :

```
1 npm install react-helmet
```

Voici l'en-tête de notre fichier `Metadata.js` avec l'import de la librairie Helmet:

```
1 /**
2 * File Name: Metadata.js
3 * Description: Fichier de métadonnées
4 * Author: Damien Vallet
5 * Version: 1.0
6 * Date: 2024
7 * License: Created for UOR course
8 * Language: JavaScript
9 */
10
11 // Importation de la librairie React
12 import React from "react";
13
14 // Importation de la librairie Helmet
15 import { Helmet } from "react-helmet";
```

Sources

- <https://stackoverflow.com/questions/37734150/how-to-update-meta-tags-in-react-js>
- <https://www.freecodecamp.org/news/react-helmet-examples/>
- <https://www.npmjs.com/package/react-helmet>

Nous allons maintenant étudier les différentes parties de notre fichier `Metadata.js`.

Métadonnées Dublin Core :

Nous avons ajouté ensuite des métadonnées descriptives au format Dublin Core.

Le format Dublin Core est un ensemble de métadonnées normalisées pour décrire des ressources numériques. Il est développé par le Dublin Core Metadata Initiative (DCMI). Ce standard vise à faciliter la gestion et l'accès aux ressources numériques dans divers domaines, tels que les bibliothèques, les archives, et les musées.

Nous avons ajouté les métadonnées suivantes :

- `DC.title` : le titre de la page
- `DC.creator` : le créateur de la page
- `DC.subject` : le sujet de la page
- `DC.description` : la description de la page
- `DC.language` : la langue de la page

```

1 /* Dublin Core Metadata */
2 <meta
3   name="DC.title"
4   content="Damien_Vallet_Audio_Product_Designer_Portofolio"
5 />
6 <meta name="DC.creator" content="Damien_Vallet" />
7 <meta name="DC.subject" content="Audio_Product_Designer_Portofolio" />
8 <meta name="DC.description" content="Portofolio_de_Damien_Vallet." />
9 <meta name="DC.language" content="fr" />
```

Sources

- https://www.dublincore.org/resources/userguide/creating_metadata/#About_the_examples
- <https://www.dublincore.org/specifications/dublin-core/dcmi-terms/>
- <https://guides.library.ucsc.edu/c.php?g=618773&p=4306386>
- https://youtu.be/RF_5qiMwNX4?feature=shared

Métadonnées Open Graph Data :

Nous avons aussi ajouté des métadonnées Open Graph Data.

Développé par Facebook, le format Open Graph Data est un ensemble de balises HTML standardisées créée pour améliorer l'affichage des liens partagés sur les réseaux sociaux.

Nous avons ajouté les métadonnées suivantes :

- `og:title` : le titre de la page
- `og:type` : le type de la page
- `og:description` : la description de la page
- `og:image` : un lien vers une image qui représente la page
- `og:url` : l'url de la page

```

1 /* Open Graph Data */
2 <meta
3   property="og:title"
4   content="Damien_Vallet--Audio_Product_Designer--Portofolio"
5 />
6 <meta property="og:type" content="website" />
7 <meta
8   property="og:description"
9   content="Portofolio_de_Damien_Vallet--Audio_Product_Designer"
10 />
11 <meta property="og:image" content="./img/logo.jpg" />
12 <meta property="og:url" content="http://www.damienvallet.art" />
```

Sources

- <https://ogp.me/>
- <https://www.freecodecamp.org/news/what-is-open-graph-and-how-can-i-use-it-for-my-website/>

Métadonnées LD-JSON :

Enfin, nous avons ajouté des métadonnées LD-JSON.

Le format LD-JSON (JavaScript Object Notation for Linked Data) est un format de données structurées utilisant JSON pour améliorer le SEO et de rendre le contenu plus accessible pour les moteurs de recherche.

Nous avons ajouté les métadonnées suivantes :

- @context : le contexte des données
- @type : le type de données
- name : le nom de la personne
- url : l'url de la personne
- sameAs : les liens vers les profils sociaux de la personne

Pour insérer des métadonnées au format JSON-LD, nous avons dû adapter le code car, dans une application React, insérer directement une balise script dans le JSX ne fonctionne pas comme d'habitude. Il faut stocker le contenu JSON dans une variable et l'insérer dans une balise <script>.

Voici le code de notre variable pour stocker les métadonnées au format JSON-LD :

```

1 // variable contenant nos données structurées JSON-LD
2 const ldJson = {
3   "@context": "http://schema.org",
4   "@type": "Person",
5   name: "Damien_Vallet",
6   url: "http://www.damienvallet.art",
7   sameAs: [
```

```
8     "https://www.linkedin.com/in/damien-vallet-84464195",
9     "https://github.com/damienvallet",
10    "https://mastodon.social/@darladada",
11  ],
12};
```

Pour insérer la variable `ldJson` dans une balise `<script>`, nous avons utilisé la méthode `JSON.stringify()` pour convertir l'objet JSON en chaîne de caractères.

```
1 /* LD JSON metadata */
2 <script type="application/ld+json">{JSON.stringify(ldJson)}</script>
```

1. Dans ce fichier, nous avons ajouté des métadonnées standard, Dublin Core, Open Graph Data et LD-JSON à notre application React.
2. Nous avons utilisé la librairie React Helmet pour gérer les métadonnées dans notre application.
3. Nous avons ajouté des balises `<meta>` pour les prendre en charge les métadonnées.

Sources

- <https://json-ld.org/>
- <https://www.youtube.com/watch?v=vioCbTo3C-4>
- <https://blog.haroen.me/json-ld-with-react>

En-tête de la page

Nous avons ensuite créé la partie principale de notre site en HTML.

En anticipant sur la suite, nous avons décidé de créer un composant React pour l'en-tête de notre site. Celui-ci contient une barre de navigation avec des liens vers les différentes versions de notre site. Pour le moment, nous avons ajouté un lien vers la version HTML de notre site.

Afin de gérer la navigation entre les différentes pages de notre application React, nous avons utilisé la librairie React Router.

Ce module permet de gérer la navigation entre les différentes pages de notre application. Pour ajouter le module React Router à notre projet, nous avons exécuté la commande suivante :

```
1 npm install react-router-dom
```

```
dada@dadalinux:/media/dada/9A8604B486049345/IED/L1S1/info_fondam... Q - x
^C
dada@dadalinux:/media/dada/9A8604B486049345/IED/L1S1/info_fondamentale/Partie 2/
dv-portfolio$ npm install react-router-dom

added 3 packages, and audited 1551 packages in 4s

262 packages are looking for funding
  run `npm fund` for details

8 vulnerabilities (2 moderate, 6 high)

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

Run `npm audit` for details.
```

Figure 2.1: Installation de React Router

Voici le code source de notre fichier `NavigationHeader.js` :

```
1 /**
2 * File Name: NavigationHeader.js
3 * Description: Navigation entre versions
4 * Author: Damien Vallet
5 * Version: 1.0
6 * Date: 2024
7 * License: Created for UOR course
8 * Language: JavaScript
9 */
10
11 // Importation de la librairie React
12 import React from "react";
13
14 // Importation de la librairie React Router
15 import { BrowserRouter as Router, Route, Routes } from "react-router-dom";
16
17 // Importation des composants;
18 import NavigationHeaderBar from "./components/NavigationHeaderBar";
19 import VersioHTML from "./components/VersionHTML";
20
21 // Définition de la fonction NavigationHeader
22 function NavigationHeader() {
```

```

23  return (
24    //>
25    <>
26      {/* composant de navigation */}
27      <Router>
28        <div>
29
30          {/* Barre de navigation */}
31          <NavigationHeaderBar />
32
33          {/* definition des routes */}
34          <Routes>
35
36            {/* route vers la version HTML */}
37            <Route path="/html" element={<VersioHTML />} />
38
39          </Routes>
40        </div>
41      </Router>
42    </>
43  );
44}
45
46 // Exportation du composant NavigationHeader
47 export default NavigationHeader;

```

1. La nouvelle url de notre page est : <http://localhost:3000/html>.
2. Dans ce fichier, la balise <> est utilisée pour définir le retour HTML de la fonction et contenir les données.
3. Ensuite, nous avons ajouté une barre de navigation avec des liens vers différentes versions de notre site: une pour chaque chapitre.
Cette barre de navigation est une liste non ordonnée avec des éléments de liste pour chaque lien.
4. Nous avons utilisé la balise <Route> de la librairie React Router pour définir les routes de notre application.

Sources

- https://www.w3schools.com/html/html_lists.asp
- <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/ul>
- <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/li>
- <https://reactrouter.com/web/guides/quick-start>

Barre de navigation entre versions

Pour circuler entre nos différentes versions de site, nous avons créé une barre de navigation en haut de la page.

Cette barre de navigation est un composant React appelé `NavigationHeaderBar`.
Elle contient des liens vers les différentes versions de notre site.

Voici le code source de notre fichier `NavigationHeaderBar.js` :

```

1 /**
2 * File Name: NavigationHeaderBar.jsx
3 * Description: Composant qui gère la barre pour naviguer entre les
4 *               versions
5 * Author: Damien Vallet
6 * Version: 1.0
7 * Date: 2024
8 * License: Created for UOR course
9 * Language: JavaScript
10 */
11 // Importation de la librairie React
12 import React from "react";
13
14 // Importation de la librairie React Router
15 import { Link } from "react-router-dom";
16
17 // définition du composant NavBar
18 function NavigationHeaderBar() {
19   return (
20
21     // Barre de navigation
22     <nav id="header-navbar">
23
24       /* Une liste non ordonnée (<ul>) définit les éléments de navigation
25          . Chaque élément de navigation est un élément de liste (<li>)*/
26       <ul>
27         <li>
28           {/* Lien vers la version HTML */}
29           <Link to="/html">Version HTML</Link>
30         </li>
31
32       </ul>
33     </nav>
34   );
35 }
36
37 // Exportation du composant NavigationHeaderBar
38 export default NavigationHeaderBar;

```

1. Dans ce fichier, nous avons créé un composant React appelé `NavigationHeaderBar` qui contient une barre de navigation avec des liens vers les différentes versions de notre site.
2. Nous avons utilisé la balise `<nav>` pour définir la barre de navigation et la balise `` pour définir les éléments de navigation.
3. La balise `` est utilisée pour chaque lien de navigation.
4. Nous avons utilisé la balise `<Link>` de la librairie React Router pour créer des liens entre les différentes versions de notre site.

Sources

- https://www.w3schools.com/html/html_lists.asp
- <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/ul>
- <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/li>
- <https://reactrouter.com/web/guides/quick-start>

Corps de la page

Après avoir créé notre menu de navigation entre les futures pages, nous allons ajouter le corps HTML de notre page web. Pour cela nous avons créé un composant React appelé **VersionHTML.jsx**. L'extension .jsx est utilisée pour les fichiers JavaScript qui contiennent du code React.

Voici de façon découpé et commenté le code source de notre fichier **VersionHTML.jsx**.

Tout d'abord, on importe les images, les icônes et la vidéo que nous allons utiliser dans notre site.

```

1 /**
2 * File Name: VersionHTML.jsx
3 * Description: Page principale avec HTML
4 * Author: Damien Vallet
5 * Version: 1.0
6 * Date: 2024
7 * License: Created for UOR course
8 * Language: JavaScript
9 */
10
11 // Importation de la librairie React
12 import React from "react";
13
14 // Importation des images
15 import picofme from "../img/picofme.jpg";
16
17 // Importation des icônes
18 import { ReactComponent as LinkedinIcon } from "../img/linkedin.svg";
19 import { ReactComponent as GithubIcon } from "../img/github.svg";
20 import { ReactComponent as MastodonIcon } from "../img/mastodon.svg";
21
22 // importation du composant vidéo
23 import videopres from "../img/UOR-ch2-video.mov";

```

Ensuite, on crée la fonction qui porte le nom du composant que l'on veut retourner et qui ici contient le contenu de notre site.

Pour créer un composant React, on crée une fonction qui retourne du code HTML.

Voici un exemple avec le composant **versionHTML** :

```

1 // définition du composant VersionHTML
2 function versionHTML() {
3   return (
4     // On insert ici le code HTML qui sera retourné par la fonction.
5   );
6 }

```

Ce composant comporte une première section avec le titre et le sous-titre de la page. Ensuite, nous avons une section pour la présentation, une section pour une vidéo, une section pour les projets, une section pour les contacts et enfin un pied-de-page avec des liens vers les réseaux sociaux avec des icônes au format .svg.

Nous allons expliquer en détail chaque partie de notre site dans les sections suivantes.

Titre de la page

```

1  /* ouverture de la balise header */
2  <header>
3      <div id="title-section">
4          /* Le titre de la page */
5          <h1 id="title">Damien Vallet</h1>
6          <h3>Audio & Product Design / Programmation</h3>
7      </div>
8  </header>
9  <hr />
```

Dans cette partie, nous avons ajouté le titre de notre page dans une balise `<h1>` et une description dans une balise `<h3>`.

Nous les avons inclus dans une balise `<div>` afin de les distinguer du reste du contenu de la page. Nous avons également ajouté une ligne horizontale `<hr>` pour séparer le titre du reste du contenu.

Présentation

```

1  /* section Présentation */
2  <section id="whoami">
3      <h2>Présentation</h2>
4      <p>
5          Je suis créateur de produit audio, designer sonore, compositeur et
6          étudiant en informatique.
7      </p>
8      <p>
9          Passionné par la musique, les processus de création ainsi que la
10         technologie, je travaille depuis 2009 pour la société française&nbsp;;
11         <a href="https://www.uvi.net" target="_blank" rel="noreferrer">
12             <strong>UVI Sounds and Software</strong>
13         </a>
14         .
15     </p>
16     <p>
17         Pendant ces années, j'ai participé à la conception et au développement
18         de nombreux instruments virtuels tels que :&nbsp;
19         <strong>Soul Drums</strong>&nbsp; et &nbsp;;
20         <strong>Glass Orchestra</strong>.
21     </p>
22     <p>
23         J'ai également apporté ma contribution à des projets de composition et
24         de production musicale pour des artistes tel que &nbsp;;
25         <strong>Mc Solaar</strong>, &nbsp;<strong>Fred Rister</strong>&nbsp;(
26         <strong>David Guetta</strong>), &nbsp;<strong>I Love UFO</strong>,
27         &nbsp;<strong>Martin Guerre</strong>, &nbsp;<strong>Agille</strong>,
28         &nbsp;<strong>The Black Mental</strong>...
29     </p>
30     <p>
31         Depuis 2023, je suis étudiant en License informatique à l'université
32         Paris 8.
33     </p>
34 </section>
```

Dans cette section, nous avons ajouté une présentation de notre parcours professionnel et de nos compétences.

Ici nous avons utilisé des balises `<p>` pour les paragraphes et des balises `` pour mettre en évidence certains mots.

Pour éviter les problèmes d'espacement avec la balise ``, nous avons utilisé la balise ` ` pour ajouter des espaces entre les mots.

Nous avons également ajouté des liens vers des sites web externes en utilisant la balise `<a>`. Le tout est inclus dans une balise `<section>` pour structurer le contenu de la page.

Vidéo de présentation

```

1 /* section Video */
2 <section id="video">
3   <figure>
4     <h2>Vidéos de présentation.</h2>
5     <video
6       src={videopres}
7       controls
8       width="320"
9       height="240"
10      className="embed-responsive-item"
11    ></video>
12  </figure>
13 </section>

```

Dans cette section, nous avons ajouté une vidéo de présentation de notre travail.

Nous avons inclus la vidéo dans une balise `<video>` avec les attributs `src` pour spécifier l'url de la vidéo, `controls` pour afficher les contrôles de lecture, `width` et `height` pour définir la taille de la vidéo.

Nous avons également ajouté un titre à la vidéo dans une balise `<h2>` pour la distinguer du reste du contenu de la page.

Section projets

```

1 /* section Projet */
2 <section id="project">
3   <h2>Projets</h2>
4   <ul>
5     <li>
6       <iframe width={560} height={315} src="https://www.youtube.com/embed/
7         LHycQ7z5rUE?si=fkmdlyK2lCbW6rwj" title="YouTube video player"
8         allow="accelerometer; autoplay; clipboard-write; encrypted-media;
9         gyroscope; picture-in-picture; web-share" referrerPolicy="strict
10        -origin-when-cross-origin" allowFullScreen />
11       <a
12         href="https://www.uvi.net/glass-orchestra"
13         target="_blank"
14         rel="noreferrer"
15       >
16         Glass Orchestra
17       </a>
18       &nbsp; by UVI Sounds and Software
19     </li>
20     <li>
21       <iframe width="560" height="315" src="https://www.youtube.com/embed
22         /9N-y51YpAcQ?si=02FHB0yuk1iwzv_z" title="YouTube video player"
23         allow="accelerometer; autoplay; clipboard-write; encrypted-media;
24         gyroscope; picture-in-picture; web-share" referrerPolicy="strict
25        -origin-when-cross-origin" allowFullScreen />
26     </li>
27   </ul>
28 </section>

```

```

        ↞gyroscope; ↞picture-in-picture; ↞web-share" referrerPolicy="strict
        -origin-when-cross-origin" allowFullScreen />
18    <a
19        href="https://www.uvi.net/en/bass-drums-perc/soul-drums.html"
20        target="_blank"
21        rel="noreferrer"
22    >
23
24        Soul Drums
25    </a>
26    &nbsp; by UVI Sounds and Software
27  </li>
28  <li>
29      <iframe width="560" height="315" src="https://www.youtube.com/embed/
          x4JVg3drMms?si=xGZqecbKc8XgB6ro" title="YouTube video player"
          allow="accelerometer; ↞autoplay; ↞clipboard-write; ↞encrypted-media;
          ↞gyroscope; ↞picture-in-picture; ↞web-share" referrerPolicy="strict
          -origin-when-cross-origin" allowFullScreen></iframe>
30      Mc Solaar &nbsp;
31      <a href="https://youtu.be/x4JVg3drMms?feature=shared">
32          <em>Les Mirabelles</em>
33      </a>
34      &nbsp; sur l'album Géopoétique (2017)
35  </li>
36  </ul>
37 </section>
```

Dans cette section, nous avons ajouté une liste de projets auxquels nous avons participé. Nous avons utilisé une balise `` pour créer une liste non ordonnée et des balises `` pour chaque élément de la liste.

Chaque élément de la liste contient une vidéo dans une balise `<iframe>` pour afficher une vidéo YouTube, un lien vers le projet en utilisant la balise `<a>` et du texte pour décrire le projet.

Section Contact

```

1  /* section Contact */
2  <section id="contact">
3      <h2>Contact</h2>
4      <figure id="contact_mail">
5          <a href="mailto:damienjdvallet@gmail.com">
6              <p>
7                  Vous pouvez m'envoyer un <em>email</em>.
8              </p>
9              <img
10                 src={picofme}
11                 alt="portrait_de_Damien_Vallet"
12                 width="220"
13                 height="250"
14             />
15          </a>
16      </figure>
17  </section>
```

Dans cette section, nous avons ajouté un lien de contact.

Nous avons utilisé une balise <figure> pour inclure une image et un lien dans un bloc de contenu. L'image de profil est prise en charge par une balise pour personnaliser le lien de contact. Nous avons inclus une adresse email dans une balise <a> avec l'attribut mailto pour ouvrir le client de messagerie par défaut de l'utilisateur.

Pied de page

```

1 <footer>
2     <p>&copy; 2024 - Damien Vallet - Audio Product Designer - Portofolio</
3         p>
4     <a href="https://www.linkedin.com">
5         <LinkedinIcon />
6     </a>
7     <a href="https://www.github.com">
8         <GithubIcon />
9     </a>
10    <a href="https://www.mastodon.com">
11        <MastodonIcon />
12    </a>
13 </footer>
```

Dans cette section, nous avons ajouté un pied de page avec des liens vers les réseaux sociaux. Nous avons inclus une mention de copyright dans une balise <p>.

Nous avons ajouté des icônes pour LinkedIn, GitHub et Mastodon en utilisant des icônes au format svg.

Enfin nous avons exporté notre composant VersionHTML pour pouvoir l'utiliser dans notre application React.

```

1 // Export du composant VersionHTML
2 export default VersionHTML;
```

Sources

- HTML SVG - W3Schools
- How to import SVGs in React and Vite - freeCodeCamp

En conclusion, nous avons créé une application React pour afficher notre page web personnelle.

Pour le moment, elle n'est composée que de composants React qui affichent du contenu HTML.

Nous allons continuer à ajouter des fonctionnalités à notre application React pour la rendre plus interactive.

Les codes source de ce chapitre sont disponibles dans l'annexe A.

Validation W3C

Pour vérifier que notre code est correct, nous avons utilisé le validateur de code HTML en ligne du W3C.

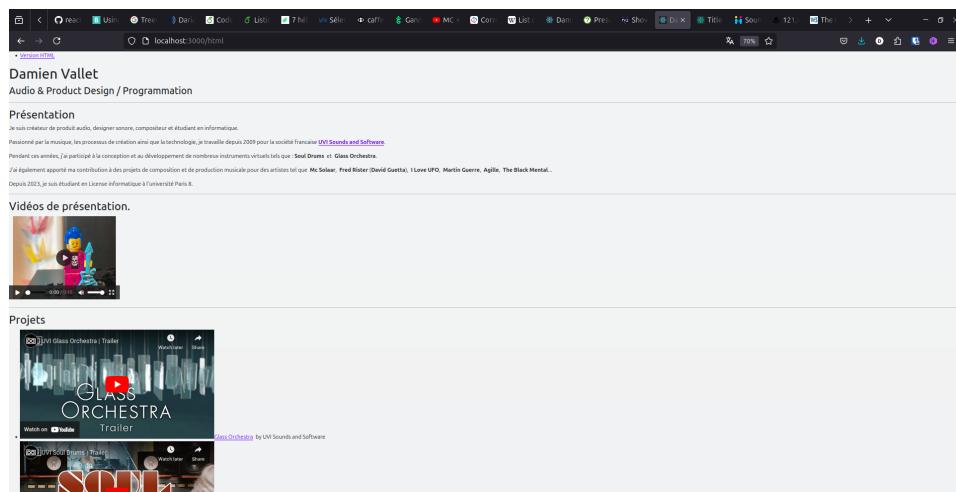
Sources

- <https://validator.w3.org/>

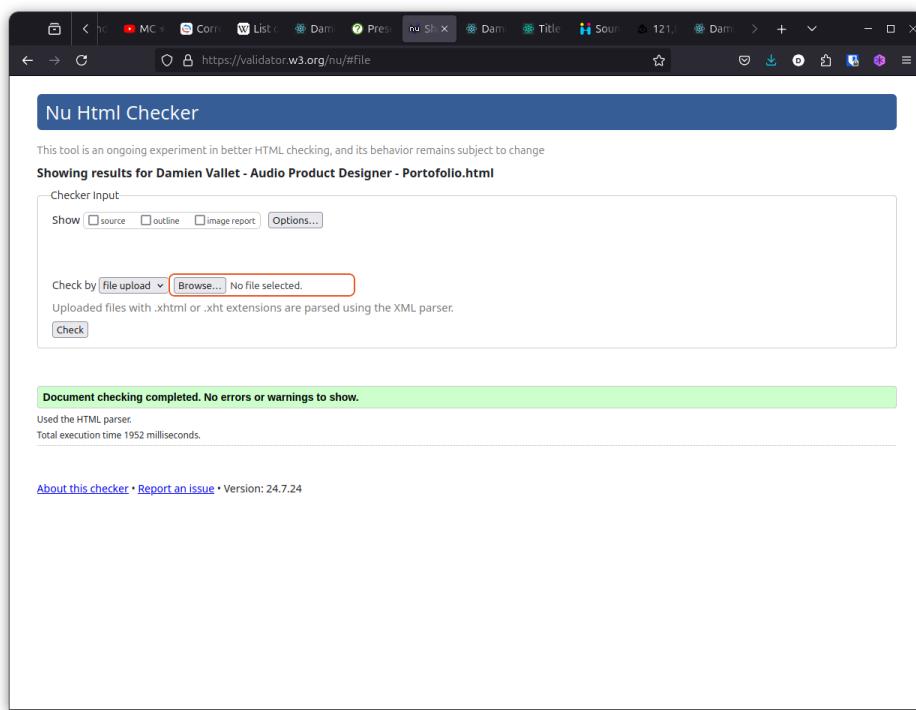
The screenshot shows a code editor with multiple tabs open, displaying a React application structure for a portfolio website. The main file is `Main.js`, which imports components like `ContactMe` and `Picofme`. Other files include `Header.js`, `Footer.js`, and `App.js`. The sidebar shows a tree view of the project structure under "OPENED EDITORS". The bottom status bar indicates the file is "Dev Container: website".

```
Partie 2>dev-portfolio >> JS Main.js >> ContactMe
  1  /**
  2   * File Name: Main.js
  3   * Description: Main application component
  4   * Author: Jean-Michel Vallet
  5   * Version: 1.0
  6   * Date: 2024
  7   * License: Created for UMR course
  8   * Language: JavaScript
  9   */
 10
 11 import React from 'react';
 12
 13 // Importation des images
 14 import concatame from './img/contactame.jpg';
 15 import picofme from './img/picofme.jpg';
 16
 17 // définition du composant ContactMe
 18 function ContactMe() {
 19   return (
 20     <div>
 21       <img src={concatame} alt="un chat qui écrit" width="220" height="170" />
 22     </div>
 23   );
 24 }
 25
 26 // définition du composant Picofme
 27 function Picofme() {
 28   return (
 29     <div>
 30       <img src={picofme} alt="portait" width="220" height="250" />
 31     </div>
 32   );
 33 }
 34
 35 // définition du composant Main
 36 function Main() {
 37   return (
 38     // affichez le code HTML
 39     <div id="main">
 40       (<* section id="Présentation" *>)
 41       <section id="whoami">
 42         <h2>Présentation</h2>
 43       </section>
 44       Je suis créateur de produit audio, designer sonore, compositeur et
 45       étudiant en informatique.
 46     </div>
 47   );
 48 }
```

Figure 2.2: Capture d'écran de la page Main.js dans Visual Studio Code



Capture d'écran de la page HTML dans le navigateur Firefox



Validation de la page HTML avec le validateur W3C

3

Chapitre 3 - Les Feuilles de Style en Cascade

Exercice 3.1 (A Rendre)

A l'aide d'un ou du même éditeur de textes utilisé précédemment :

Dupliquez la page web personnelle que vous avez construite au chapitre précédent et insérez dans cette dernière les CSSv3 en vous attachant à utiliser les balises float, clear et div.

- Comme vous disposez de deux pages distinctes sur votre serveur, ajoutez **un menu** ou une **barre de navigation** de votre choix à votre page d'accueil.
- Gardez bien évidemment les métadonnées descriptives sur ces deux pages. Attention : il est important qu'il ne s'agisse pas d'une nouvelle page sur un nouveau sujet mais d'une *modification* par simple copie de la page utilisée dans le chapitre précédent.
- Dans le **PDF**, après avoir rappelé, comme à l'habitude, l'énoncé de l'exercice :
 - Indiquez l'**URL de votre nouvelle page "stylée"**.
 - Faites un C.C (Copier-Coller) de la **feuille de Style** dans son intégralité en ajoutant des commentaires sur son utilisation et vos choix.
 - Ajoutez une C.E (Copie d'Ecran) de votre **page affichée avec le style choisi dans votre navigateur**.
 - Joignez en **archive attachée** le contenu du **Document Root** (htdocs) qui contiendra, en plus, **la feuille de style retenue**.

Exercice 3.2 (Bonus)

Rendez votre page "responsive" en utilisant un Framework CSS de votre choix (Bootstrap,...). Si vous réalisez le bonus, ce rendu adaptatif se substituera à la première question de l'exercice 3.1.

Remarque : Il est tout à fait possible, en vous inspirant du site CSSZenGarden (cf lien), de produire deux feuilles de style différentes qui pourront s'appliquer à un même contenu.

Pensez à bien ajouter les copies d'écran de l'affichage sur deux terminaux différents (par exemple : sur ordinateur et sur smartphone) ou avec les deux feuilles de styles si vous choisissez cette alternative.

Exercices 3.1 et 3.2 :

Pour pouvoir réaliser ces exercices, nous avons dupliqué la page HTML que nous avons écrite au chapitre précédent.

Nous avons tout d'abord besoin d'ajouter une nouvelle route à notre barre de navigation pour rediriger vers la nouvelle page CSS.

Ajout d'une nouvelle route

Dans notre composant `NavigationHeader`, nous avons ajouté un nouveau chemin pour rediriger vers la nouvelle page CSS appelée `VersionCSS.jsx`.

Voici la version améliorée de notre fichier `NavigationHeader.js` :

```

1 /**
2 * File Name: NavigationHeader.js
3 * Description: Navigation entre versions
4 * Author: Damien Vallet
5 * Version: 1.0
6 * Date: 2024
7 * License: Created for UOR course
8 * Language: JavaScript
9 */
10
11 // Importation de la librairie React
12 import React from "react";
13
14 // Importation de la librairie React Router
15 import { BrowserRouter as Router, Route, Routes } from "react-router-dom";
16
17 // Importation des composants;
18 import NavigationHeaderBar from "./components/NavigationHeaderBar";
19 import VersioHTML from "./components/VersionHTML";
20 import VersionCSS from "./components/VersionCSS";
21
22 // Définition de la fonction NavigationHeader
23 function NavigationHeader() {
24     return (
25         //
26         <>
27             {/* composant de navigation */}
28             <Router>
29                 <div>
30
31                     {/* Barre de navigation */}
32                     <NavigationHeaderBar />
33
34                     {/* definition des routes */}
35                     <Routes>
36
37                         {/* route vers la version HTML */}
38                         <Route path="/html" element={<VersioHTML />} />
39
40                         {/* route vers la version CSS */}
41                         <Route path="/css" element={<VersionCSS />} />
42
43                 </Routes>

```

```
44     </div>
45   </Router>
46   </>
47 );
48 }
49
50 // Exportation du composant NavigationHeader
51 export default NavigationHeader;
```

- **URL de la nouvelle page "stylée"**: <http://localhost:3000/css>
- Dans ce fichier, nous avons ajouté une nouvelle route pour rediriger vers la nouvelle page CSS.

```
dada@dadalinux: /media/dada/9A8604B486049345/IED/L1S1/info_fondam...
dv-portfolio$ npm install bootstrap
added 2 packages, and audited 1553 packages in 3s
264 packages are looking for funding
  run `npm fund` for details
8 vulnerabilities (2 moderate, 6 high)

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

Run `npm audit` for details.
dada@dadalinux:/media/dada/9A8604B486049345/IED/L1S1/info_fondamentale/Partie 2/
dv-portfolio$ 
```

Figure 3.1: Installation de Bootstrap

Bootstrap

Pour ajouter du CSS à notre page, nous avons utilisé le framework CSS **Bootstrap** pour rendre la page responsive.

Pour ajouter Bootstrap à notre projet, nous avons exécuté la commande suivante :

```
1 npm install bootstrap
```

Ensuite, il faut importer le fichier CSS de Bootstrap dans notre fichier **App.js** en ajoutant la ligne suivante sous les imports :

```
1 //Importation de la librairie Bootstrap
2 import 'bootstrap/dist/css/bootstrap.css';
```

On importe également dans le fichier **App.js** une feuille de style CSS personnalisée que l'on aura rangée dans un dossier **stylesheet**.

```
1 // Importation de la feuille de style
2 import "./stylesheet/style.css";
```

Dans React, on spécifie les classes CSS en utilisant l'attribut **className** au lieu de **class**.

Pour rendre notre page responsive, nous avons utilisé le système de Grid de Bootstrap.

En effet, la grille (Grid) permet de définir des "boîtes" qui contiennent des lignes et chaque ligne contient des colonnes. Il suffit d'ajouter une "class" à une balise pour que la feuille de style Bootstrap lui soit appliquée.

Exemples des classes utilisés par Grid pour former une structure :

- **container** : pour créer un conteneur.
- **row** : pour créer une ligne.
- **col** : pour définir le nombre de colonnes.

En les imbriquant, on peut créer une structure de grille.

Nous retrouverons cette structure tout au long de notre fichier **VersionCSS.jsx**.

Sources

- <https://reactrouter.com/en/main/start/concepts>
- <https://create-react-app.dev/docs/adding-bootstrap/>
- <https://medium.com/seek-blog/block-element-modifying-your-javascript-components-d7f99fcab52b>
- <https://getbootstrap.com/docs/4.0/layout/grid/>

VersionCSS

Le fichier VersionCSS.jsx est une version modifiée de notre page HTML.
Nous avons ajouté des classes Bootstrap pour rendre notre page responsive.
Nous allons désormais expliquer notre fichier VersionCSS.jsx étape par étape:

En-tête du fichier :

```

1  /**
2  * File Name: VersionCSS.jsx
3  * Description: Page principale avec CSS
4  * Author: Damien Vallet
5  * Version: 1.0
6  * Date: 2024
7  * License: Created for UOR course
8  * Language: JavaScript
9  */
10
11 // Importation de la librairie React
12 import React from "react";
13
14 // Importation des images
15 import picofme from "../img/picofme.jpg";
16
17 // Importation des icônes
18 import { ReactComponent as LinkedinIcon } from "../img/linkedin.svg";
19 import { ReactComponent as GithubIcon } from "../img/github.svg";
20 import { ReactComponent as MastodonIcon } from "../img/mastodon.svg";
21 import { ReactComponent as Soundwave } from '../img/soundwave.svg';
22
23 // Importation du composant vidéo
24 import videopres from "../img/UOR-ch2-video.mov";

```

Dans cette partie :

1. Tout d'abord, on importe la librairie React.
 2. Ensuite, on importe les images et les icônes au format .svg dont nous avons besoin.
 3. Enfin, on importe une vidéo.
-

Header et Titre de la page :

```

1  /* ouverture de la balise header */
2 <header id="header" className="p-3 bg-dark text-light text-center">
3
4  /* création d'un container */
5 <div className="container-fluid">
6
7  /* création d'une ligne */
8 <div className="row">
9
10 /* création d'une colonne */
11 <div className="col-12 col">
12
13 /* titre de la page */
14 <h1 className="display-1 hero-title">Damien Vallet</h1>
15 <h2 className="display-2">Portofolio</h2>
16 <h3>Audio & Product Design / Programmation</h3>
17 </div>
18 </div>
19 </div>
20 <hr />
```

Dans cette partie, nous avons pris un exemple sur <https://getbootstrap.com/docs/5.0/examples/headers/#>. Ainsi en inspectant la page depuis notre navigateur, nous avons ajouté les classes Bootstrap suivantes à la balise header :

1. **p-3** : applique une classe qui met une marge intérieure sur la balise `<div>` concernée.
2. **text-center** : centre le texte dans le `<div>`

Ensuite, nous avons ajouté des classes CSS Bootstrap à nos autres balises.

1. **container-fluid** : étend le container sur toute la largeur de la fenêtre.
2. **row** : assure que l'espace pris par les colonnes équivaut à la largeur du container.
3. **col-12** : Détermine que l'espacement se fait sur 12 colonnes égales.
4. **display-1** et **display-2** : sont des classes dédiées aux titres.

La classe **hero-title** est définie dans notre feuille de style `style.css`.

```

1 .hero-title {
2   /* on definit la taille de la police */
3   font-size: 3.5rem;
4
5   /* on rend la police en gras */
6   font-weight: 700;
7
8   /* on definit la couleur du texte */
9   color: #F2EB6F;
10
11  /* on centre le texte */
12  text-align: center;
13
14  /* on definit la famille de la police a utiliser*/
15  font-family: Garamond, 'Times New Roman', Times, serif;
16 }
```

Sources

- <https://getbootstrap.com/docs/4.0/utilities/spacing/>
- <https://getbootstrap.com/docs/5.0/examples/headers/#>

Barre de navigation :

On a ensuite défini une barre de navigation qui permet d'aller dans les différentes sections de notre page web. Cette barre de navigation est défini ainsi :

```

1  /* navbar */
2  <nav id="navbar" className="navbar_bg-dark_text-light">
3      <div className="container">
4          <div className="d-flex flex-wrap align-items-center justify-content-
5              center justify-content-lg-start">
6              <Soundwave style={{ fill: 'white', height: '40px', width: '40px' }}>
7                  />
8              <ul className="nav_col-12_col-lg-auto_me-lg-auto_mb-2 justify-
9                  content-center mb-md-0">
10                 <li>
11                     <a className="nav-link_px-2_text-light" href="#main">
12                         Home
13                     </a>
14                 </li>
15                 <li>
16                     <a className="nav-link_px-2_text-light" href="#whoami">
17                         Présentation
18                     </a>
19                 </li>
20                 <li>
21                     <a className="nav-link_px-2_text-light" href="#project">
22                         Projets
23                     </a>
24                 </li>
25                 <li>
26                     <a className="nav-link_px-2_text-light" href="#contact">
27                         Contacts
28                     </a>
29                 </li>
30             </ul>
31         </div>
32     </div>
33 </nav>
```

Pour construire notre barre de navigation on ajoute les classes suivantes :

1. **d-flex** : Cette classe applique la propriété CSS `display:flex`. Elle permet une structure de mise en page réactive.
2. **flex-wrap** : Cette classe applique la propriété CSS `flex-wrap: wrap`; ceci permettant aux éléments de se répartir sur plusieurs lignes, au lieu de les forcer à s'adapter sur une seule ligne.
3. **align-items-center** : Cette classe applique la propriété CSS `align-items: center`; au sein du conteneur flex, alignant verticalement les éléments au centre du conteneur.
4. **justify-content-center** : Cette classe applique la propriété CSS `justify-content: center`; Elle centrant horizontalement les éléments flex à l'intérieur du conteneur.
5. **justify-content-lg-start** : Cette classe applique la propriété CSS `justify-content: flex-start`; mais uniquement sur les grands appareils (comme indiqué par le suffixe `-lg`). Elle aligne les éléments flex au début du conteneur sur les grands écrans, annulant le comportement de `justify-content-center` sur les appareils plus grands.
6. **px-2** : sert pour du padding horizontal.

Ensuite, pour chaque lien de la barre de navigation, on ajoute une classe dédiée **nav-link** qui applique un soulignement et un changement de couleur quand on survole le lien.

La classe **nav-link** est définie dans notre feuille de style personnalisée style.css.

```

1 /* comportement hover des lien de la barre de navigation */
2 .nav-link:hover {
3     /* ajoute un soulignement */
4     text-decoration: underline;
5
6     /* change la couleur du texte */
7     color: #393148;
8 }
```

On crée une classe **a** pour définir le style de la balise `<a>` et une classe **a:hover** pour définir le style de la balise `<a>` lorsqu'on survole le lien.

```

1 /* style de la balise a */
2 a {
3     /* on definit la couleur du texte */
4     color: var(--bs-indigo);
5
6     /* on ajoute un soulignement */
7     text-decoration: underline;
8
9     /* on definit le pointeur de la souris */
10    cursor: pointer;
11 }
12
13 /* style du comportement hover de la balise a */
14 a:hover {
15     /* on definit la couleur du texte */
16     color: #393148;
17
18     /* on enlève les décorations */
19     text-decoration: none;
20 }
```

Sources

- <https://developer.mozilla.org/fr/docs/Web/CSS/:hover>
- <https://getbootstrap.com/docs/4.0/components/navbar/>
- <https://getbootstrap.com/docs/4.0/utilities/flex/>

Présentation :

```

1 <section id="whoami">
2   <div className="container">
3     <div className="row">
4       <div className="col">
5         <h2>
6           <span className="title">Présentation</span>
7         </h2>
8       </div>
9     </div>
10    <div className="row">
11      <p className="text-start">
12        <span className="lettrine">J</span>e suis designer sonore,
13        compositeur et programmeur. Passionné par la <strong>musique</
14        strong>, les <strong>processus créatifs</strong> ainsi que la <
15        strong>science</strong> et la <strong>technologie</strong>, je
16        travaille
17        depuis 2009 pour la société française&nbsp;;
18        <a href="https://www.uvi.net" target="_blank" rel="noreferrer">
19          <strong>UVI Sounds and Software</strong>
20        </a>
21        .
22      </p>
23      <p className="text-start">
24        Pendant ces années, j'ai participé à la conception et au
25        développement de nombreux instruments virtuels tels que :
26        &nbs;p;<strong>Soul Drums</strong>
27        &nbs;p;et&nbs;p; <strong>Glass Orchestra</strong>.
28      </p>
29      <p className="text-start">
30        J'ai également apporté ma contribution à des projets de
31        composition et de production musicale pour des artistes tel que
32        &nbs;p;<strong>Mc Solaar</strong>, &nbs;p; <strong>Fred Rister</
33        strong> (
34        <strong>David Guetta</strong>),&nbs;p; <strong>I Love UFO</strong>,
35        &nbs;p;<strong>Martin Guerre</strong>,&nbs;p; <strong>Agille</strong
36        >,
37        &nbs;p;<strong>The Black Mental</strong>...
38      </p>
39    </div>
40  </div>

```

Pour cette section, nous avons ajouté les classes suivantes :

1. **text-start** : aligne le texte à gauche.
2. **lettrine** : permet de mettre en valeur la première lettre d'un paragraphe.
3. **title** : permet de mettre en valeur le titre de la section.

La classe **lettrine** est définie ainsi :

```

1 .lettrine {
2   /* on definit la police */
3   font-family: garamond, serif;
4
5   /* on definit la taille de la police */
6   font-size: 270%;
7
8   /* on definit la position du texte */
9   float: left;
10
11  /* on definit la marge */
12  margin: -0.20em 0px -0.20em 0px;
13 }
```

La classe **title** est définie dans notre fichier style.css.

```

1 .title {
2   /* centre le texte */
3   text-align: center;
4
5   /* on definit la taille de la police */
6   font-size: 1.5rem;
7
8   /* on rend la police en gras */
9   font-weight: 700;
10
11  /* on definit la couleur du texte */
12  color: #5400F5;
13
14  /* on ajoute un soulignement */
15  text-decoration: underline;
16
17  /* on definit la famille de la police a utiliser*/
18  font-family: Georgia, 'Times New Roman', Times, serif;
19 }
```

Sources

- <https://getbootstrap.com/docs/4.0/utilities/text/#text-alignment>
- <https://getbootstrap.com/docs/4.0/utilities/text/#text-transform>

Section Video :

Pour la section vidéo, nous avons utilisé la balise pour ajouter la classe **title** à notre titre.

```

1 <section id="video">
2 <div className="container-p-5">
3   <div className="row">
4     <div className="col">
5       <h2>
6         <span className="title">Vidéo de présentation</span>
7       </h2>
8     </div>
9   </div>
10  <div className="row">
11    <div className="col">
12      <figure>
13        <div className="embed-responsive embed-responsive-4by3 d-flex justify-content-center">
14          <video src={videopres} controls width="320" height="240" className="embed-responsive-item"></video>
15        </div>
16      </figure>
17    </div>
18  </div>
19 </div>
20 </section>
```

Pour cette section, nous avons ajouté les classes suivantes :

1. **embed-responsive** : applique des propriétés CSS permettant de créer un conteneur vidéo responsive.
2. **embed-responsive-4by3** : définit le ratio de l'élément vidéo en 4:3.

Sources

- <https://getbootstrap.com/docs/4.0/utilities/embed/>
- <https://getbootstrap.com/docs/4.0/utilities/spacing/#horizontal-centering>

Projets :

Pour la section Projets, nous avons écrit le code suivant :

```

1  /* section Projets */
2 <section id="projets">
3   <div className="container_p-5_custom-gradient">
4     <div className="row">
5       <div className="col">
6         <h2>
7           <span className="title"> Projets </span>
8         </h2>
9       </div>
10    </div>
11    <div className="row_d-flex_justify-content-center_align-items-center">
12      >
13      <div className="col_ratio_ratio-16x9">
14        <iframe className="embed-responsive-item" width="560" height="315"
15          src="https://www.youtube.com/embed/LHycQ7z5rUE?si=
16          fkmdlyK2lCbW6rwj" title="YouTube_video_player" allow=
17          accelerometer;autoplay;clipboard-write;encrypted-media;
18          gyroscope;picture-in-picture;web-share" referrerPolicy=
19          "strict-origin-when-cross-origin" allowFullScreen />
20      </div>
21      <div className="col">
22        <a
23          href="https://www.uvi.net/glass-orchestra"
24          target="_blank"
25          rel="noreferrer"
26        >
27          Glass Orchestra
28        </a>
29        &nbsp; by UVI Sounds and Software
30      </div>
31    </div>
32    <div className="row_d-flex_justify-content-center_align-items-center">
33      >
34      <div className="col">
35        <a
36          href="https://www.uvi.net/en/bass-drums-perc/soul-drums.html"
37          target="_blank"
38          rel="noreferrer"
39        >
40          Soul Drums
41        </a>
42        &nbsp; by UVI Sounds and Software
43      </div>
44      <div className="col_ratio_ratio-16x9">
45        <iframe className="embed-responsive-item" width="560" height="315"
46          src="https://www.youtube.com/embed/9N-y51YpAcQ?si=
47          O2FHB0yuk1iwzv_z" title="YouTube_video_player" allow=
48          accelerometer;autoplay;clipboard-write;encrypted-media;
49          gyroscope;picture-in-picture;web-share" referrerPolicy=
50          "strict-origin-when-cross-origin" allowFullScreen />
51      </div>
52    </div>
53  </div>
54
```

```

41   <div className="row_d-flex_justify-content-center_align-items-center">
42     >
43       <div className="col_ratio_ratio-16x9">
44         <iframe className="embed-responsive-item" width="560" height="315"
45           src="https://www.youtube.com/embed/x4JVg3drMms?si=
46             xGZqecbKc8XgB6ro" title="YouTube_video_player" allow=
47             "accelerometer;_autoplay;_clipboard-write;_encrypted-media;_
48             gyroscope;_picture-in-picture;_web-share" referrerPolicy="_
49             strict-origin-when-cross-origin" allowFullScreen></iframe>
50       </div>
51       <div className="col">
52         Mc Solaar &nbsp;
53         <a href="https://youtu.be/x4JVg3drMms?feature=shared">
54           <em>Les Mirabelles</em>
55         </a>
56         &nbsp; sur l'album Géopoétique (2017)
57       </div>
58     </div>
59   </div>
60 </section>

```

Pour cette section, nous avons modifié la version HTML pour présenter de manières alternées (gauche - droite) nos vidéos embarquées et les liens vers les projets.

Nous avons également crée un gradient personnalisé pour le fond de la section Projets. Ce gradient comporte également un flou réalisé avec la propriété `backdrop-filter`

La classe **custom-gradient** est définie ainsi :

```

1 /* style du gradient */
2 .custom-gradient {
3   /* on definit le dégradé radial */
4   background: radial-gradient(circle, rgb(238, 236, 204) 0%, var(--bs-
5     light) 100%);
6   /* on definit la taille du flou avec un filtre*/
7   backdrop-filter: blur(80px);
8 }

```

Sources

- <https://getbootstrap.com/docs/4.0/utilities/spacing/#horizontal-centering>

Contact :

Pour la section Contact, nous avons aussi utilisé la balise `` pour ajouter la classe `title` à notre titre. Voici le code de notre section Contact :

```

1  /* section Contact */
2 <section id="contact" className="bg-light">
3   <div className="container_p-5">
4     <div className="row">
5       <h2>
6         <span className="title">Contacts</span>
7       </h2>
8       <div className="col_d-flex_justify-content-center">
9         <figure id="contact_mail">
10        <a href="mailto:damienjdvallet@gmail.com">
11          <img src={picofme} alt="portait_de_Damien_Vallet" width="270"
12              height="250" className="img-fluid_rounded_mx-auto_d-block"
13              />
14        <p className="text-center">
15          Vous pouvez m'envoyer un <em>email</em>.
16        </p>
17      </a>
18    </figure>
19  </div>
20  <div className="row_p-5">
21    <div className="col_d-flex_justify-content_center_align-items-center"
22        ">
23      <a href="https://www.linkedin.com/in/damien-vallet-84464195/">
24        <LinkedinIcon className="bg-light_socials-icons" style={{ height
25          : "3em", width: "3em" }} />
26      </a>
27    </div>
28    <div className="col_d-flex_justify-content_center_align-items-center"
29        ">
30      <a href="https://github.com/damienvallet">
31        <GithubIcon className="bg-light_socials-icons" style={{ height:
32          "3em", width: "3em" }} />
33      </a>
34    </div>
35    <div className="col_d-flex_justify-content_center_align-items-center"
36        ">
37      <a href="https://mastodon.social/@darladada">
38        <MastodonIcon className="bg-light_socials-icons" style={{ height
39          : "3em", width: "3em" }} />
40      </a>
41    </div>
42  </div>
43 </div>
44 </section >

```

Pour cette section, nous avons centré le titre et l'image de notre portrait. Nous avons également centré et aligné horizontalement les icônes de nos réseaux sociaux.

1. **img-fluid** : applique la propriété `max-width:100%; height:auto` permettant de rendre les images réactives.

2. **rounded** : applique la propriété `border-radius:0.25rem` permettant d'arrondir les coins de l'image.
 3. **mx-auto** : applique la propriété `margin-left:auto; margin-right:auto` permettant de centrer l'image horizontalement.
 4. **d-block** : applique la propriété `display:block` permettant de rendre l'image en ligne.
-

Sources

- <https://getbootstrap.com/docs/4.0/content/images/#responsive-images>
- <https://developer.mozilla.org/fr/docs/Web/CSS/display>
- <https://getbootstrap.com/docs/4.0/utilities/spacing/#horizontal-centering>

Footer :

Voici le code de notre footer :

```

1 /* footer */
2 <footer id="footmain" className="p-3 bg-dark text-light">
3   <div className="container">
4     <div className="row">
5       <div className="col">
6         <p>2024 - Damien Vallet</p>
7       </div>
8     </div>
9   </div>
10 </footer>
```

La classe `socials-icons` est définie dans notre fichier `style.css`.

```

1 .socials-icons {
2   /* on met les icônes en ligne */
3   display: inline-block;
4 }
```

On utilise la propriété `display:inline-block` pour que les icônes soient alignées horizontalement.

En conclusion, dans ce chapitre, nous avons dupliqué notre page HTML et créer une nouvelle version utilisant le framework CSS Bootstrap.

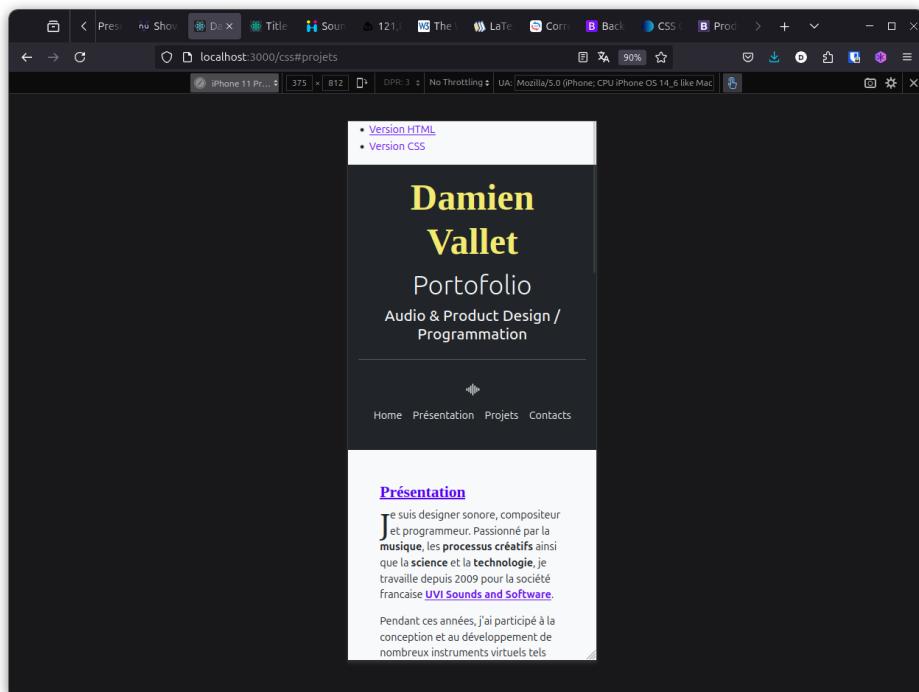
Nous avons ajouté une barre de navigation, pour pouvoir visiter les différentes versions de notre page.

Enfin, nous avons ajouté des classes Bootstrap pour rendre notre page responsive.

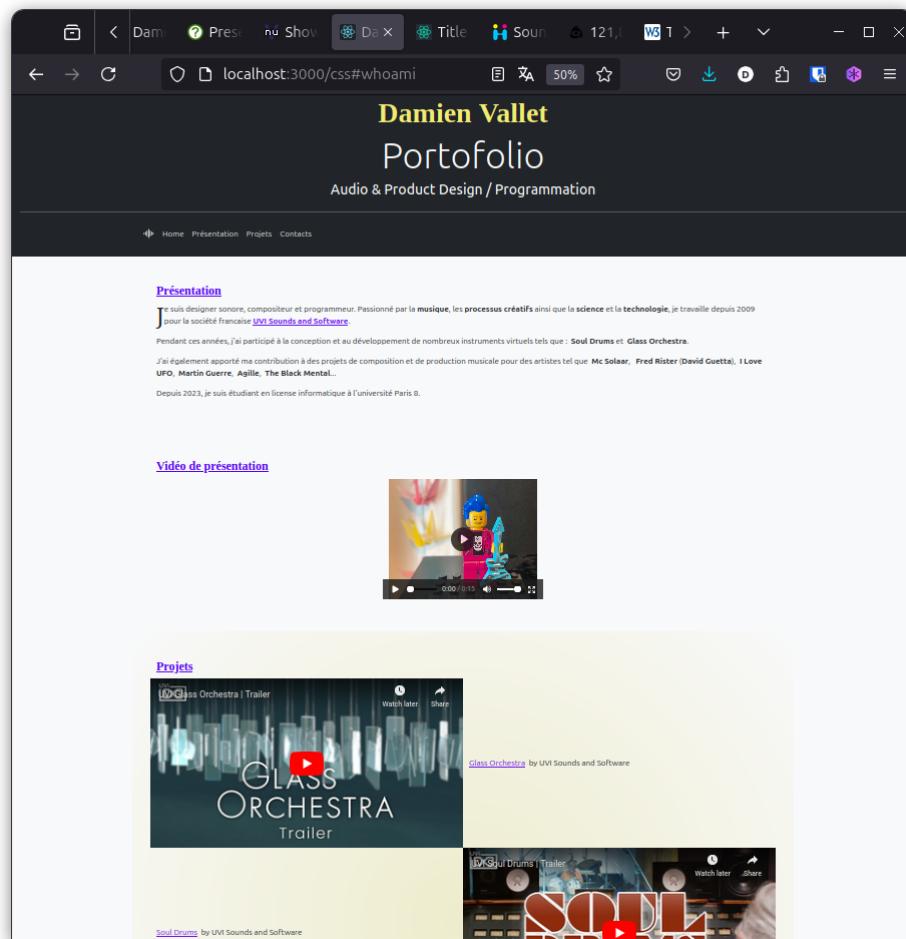
Les codes source de ce chapitre sont disponibles dans l'annexe B.

Sources

- <https://developer.mozilla.org/fr/docs/Web/CSS/display>



Page affichée avec la vue mobile du navigateur



Page affichée avec la vue desktop du navigateur

4

Chapitre 4 - Le Protocole HTTP, Architecture Clients et Serveurs Web

Exercice 4.1 (A Rendre)

En utilisant deux ordinateurs connectés sur le réseau..., à l'aide d'une VM, ou en local, sur la même machine, avec deux fenêtres de Terminal ouvertes :

- Recopiez un **gros fichier** entre deux machines (ou entre deux terminaux) en utilisant **netcat**.
- Pensez à définir ce que "gros fichier" signifie... A l'heure actuelle, il s'agira vraisemblablement d'un fichier > 1Go sauf si votre ordinateur est très très ancien ou votre connexion ADSL est très lente.
- Mesurez le temps que ça prend et en déduire le débit de la connexion entre les deux machines (probablement en Go ou en Mo par seconde).
- N'oubliez pas d'expliquer les commandes utilisées, de réaliser les C.E(s) d'illustration et de citer au besoin vos toutes sources. Pour obtenir des fichiers de tests, vous pouvez utiliser le site TestDebit.

Exercice 4.1 :

Pour réaliser cet exercice, nous allons transférer un fichier de 1 Go entre deux machines connectées sur le même réseau. Nous avons choisi une taille de 1Go pour notre fichier car c'est une taille suffisamment grande pour mesurer le débit de la connexion.

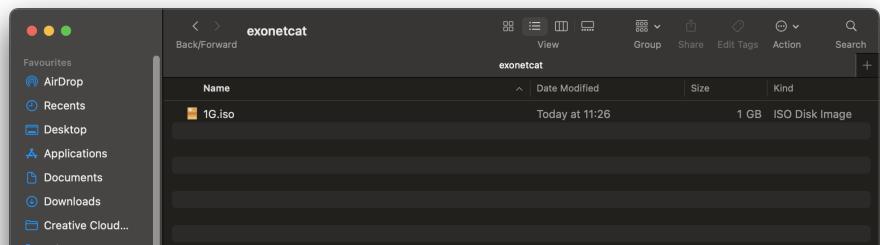
En effet, le débit moyen en France étant de **120Mb/s** soit **15Mo/s**, un fichier de 1 Go devrait prendre environ une minute à transférer. Ce qui semble être un temps raisonnable pour mesurer le débit de la connexion.

Ce fichier sera transféré de la machine 1 (**client**) à la machine 2 (**serveur**).

Nous allons utiliser un fichier de 1Go appelé "1G.iso".

Ce fichier provient de <http://bouygues.testdebit.info/1G.iso>

Il est téléchargé sur la machine 1 : le client.



Le fichier 1G.iso est téléchargé sur la machine 1

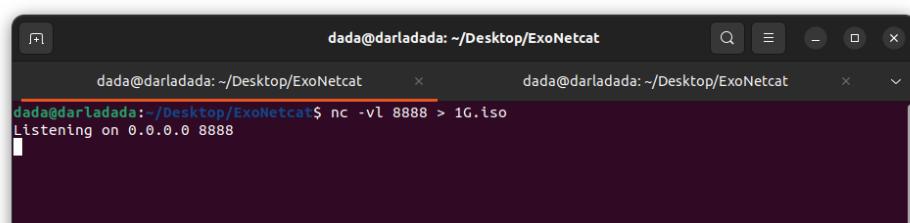
Sources

- <https://fr.statista.com/statistiques/999067/vitesse-connexion-internet-moyenne-pays-europe/>

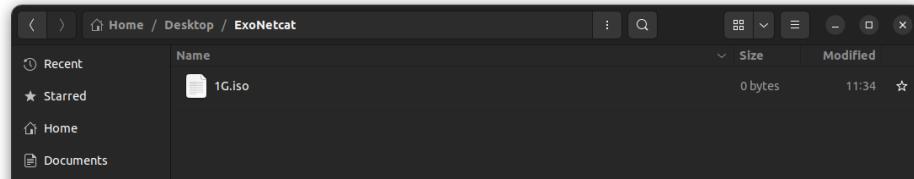
Coté serveur :

Pour connecter les deux machines, nous allons utiliser, dans un Terminal sur la machine 2, la commande *netcat* suivante depuis le dossier où se situe notre fichier à transférer.

```
1 nc -vl 8888 > 1G.iso
```



La commande pour attendre une connexion sur le port 8888



Le fichier 1G.iso est vide avant le transfert

Ici la machine 2 attend une connexion (option *-l*) sur le port 8888 et redirige la sortie de **netcat** vers un fichier vide 1G.iso.

Une fois la connexion établie avec le client, le fichier 1G.iso sera rempli avec les données reçues.

Coté client :

Du coté du client, nous allons utiliser la commande suivante pour se connecter au serveur.

```
1 nc cat 1G.iso | nc X.X.X.X 8888
```

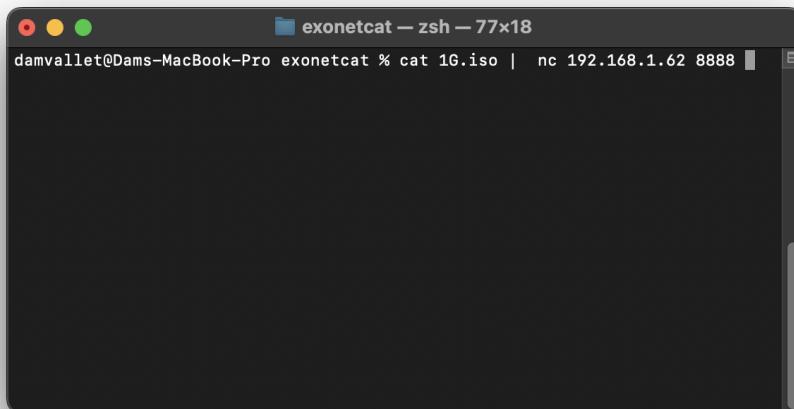
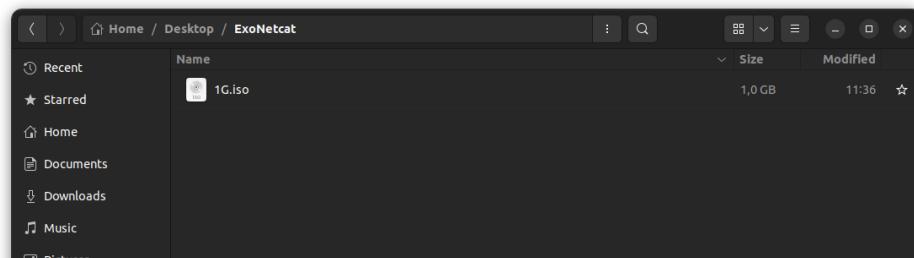


Figure 4.1: La commande pour se connecter au serveur

La commande *cat* permet de lire le fichier 1G.iso et de le transmettre à netcat via la commande *pipe* qui l'envoie à l'adresse IP de la machine 1 sur le port 8888.



Le fichier 1G.iso est transféré avec succès

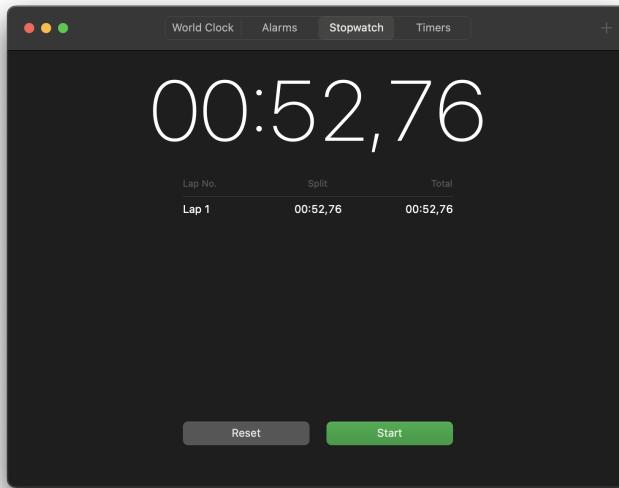
Pour résumer, nous avons transféré un fichier de 1Go entre deux machines connectées sur le même réseau en utilisant netcat.

On voit dans la colonne **Size** de notre navigateur que le fichier 1G.iso est transféré avec succès.

Le fichier 1G.iso est maintenant rempli avec les données reçues et fait bien 1Go.

Mesure du temps de transfert :

Nous avons mesurer à la main avec un chronomètre le temps que le transfert prend.
La mesure du temps nécessaire pour transférer un fichier de 1Go est de **52.76s**.



Mesure du temps de transfert du fichier 1G.iso

Calcul du débit de la connexion :

Ensuite, nous avons calculé le débit de la connexion en utilisant la formule suivante :

$$\begin{aligned} \text{Débit} &= \frac{\text{Taille du fichier}}{\text{Temps de transfert}} \\ &= \frac{1 \text{ Go}}{52.76 \text{ s}} \\ &\approx 19.5 \text{ Mo/s} \end{aligned}$$

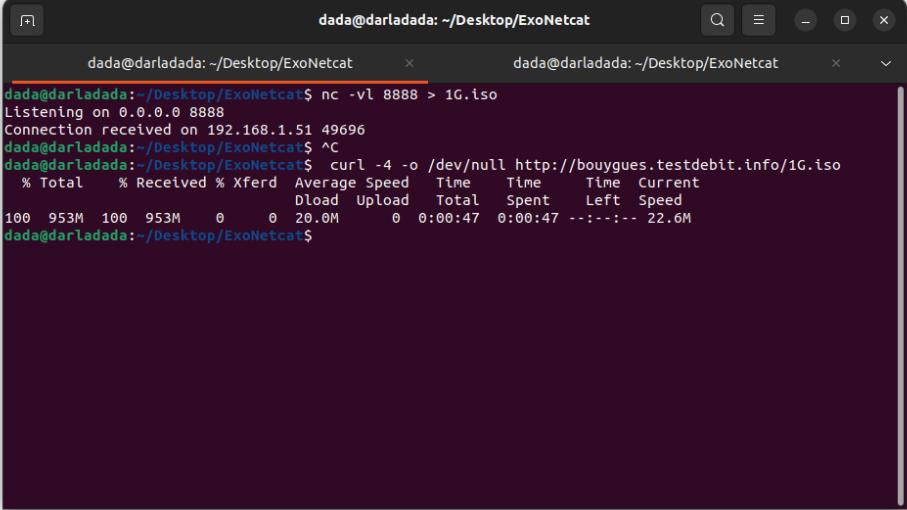
Ainsi, le débit pour un fichier de 1Go transféré en 52.76s est d'environ **19.5Mo/s**.

En comparant avec les résultats obtenu avec le site TestDebit, et la commande curl :

```
1 curl -4 -o /dev/null http://bouygues.testdebit.info/1G.iso
```

On obtient un résultat de **20.Mo/s** ce qui est cohérent avec notre mesure.

Pour résumer, nous avons transféré un fichier de 1Go entre deux machines connectées sur le même réseau en utilisant netcat.
Nous avons mesuré le temps de transfert et calculé le débit de la connexion.
Le débit de la connexion est d'environ **19.5Mo/s**.
Nous avons comparé nos résultats avec ceux obtenus avec le site TestDebit.
Nous avons obtenu un résultat de **20Mo/s** ce qui est cohérent avec notre mesure.



The screenshot shows a terminal window with two tabs. Both tabs are titled "dada@darladada: ~/Desktop/ExoNetcat". The left tab contains the command:

```
dada@darladada:~/Desktop/ExoNetcat$ nc -vl 8888 > 1G.iso
```

Output from the command:

```
Listening on 0.0.0.0 8888
Connection received on 192.168.1.51 49696
dada@darladada:~/Desktop/ExoNetcat$ ^C
```

The right tab shows the command:

```
dada@darladada:~/Desktop/ExoNetcat$ curl -4 -o /dev/null http://bouygues.testdebit.info/1G.iso
```

Output from the curl command, showing a progress bar and statistics:

% Total	% Received	% Xferd	Average Speed	Time	Time	Time	Current
Dload	Upload	Total	Spent	Left	Speed		
100	953M	100	953M	0	0	20.0M	0 0:00:47 0:00:47 --::-- 22.6M

The terminal prompt "dada@darladada:~/Desktop/ExoNetcat\$" is visible at the bottom of both tabs.

Figure 4.2: mesure du débit avec la commande curl

Sources

- <https://www.digitalocean.com/community/tutorials/how-to-use-netcat-to-establish-and-test-tcp-and-udp-connections>
- https://fr.wikipedia.org/wiki/D%C3%A9bit_binaire
- <https://www.youtube.com/watch?v=hm6BvBt8cyk>

Exercice 4.2 (A Rendre)

Sur votre ordinateur en local (sur lequel vous aurez installé XAMPP), en virtualisation (VirtualBox, VMWare, Docker,...) ou chez votre hébergeur gratuit (selon) :

- Installez une Application Web de votre choix (CMS, LMS, Wiki,...) et compléter une page à minima avec des contenus relatifs à la thématique choisie (cf chapitre 2).
- Expliquez et Illustriez chacune des phases de l'installation à l'aide de commentaires et de copies d'écran (C.E(s)).
- Pour Vous Aider : Vous pouvez vous rendre sur le site Bitnami où vous trouverez une collection de modules XAMPP à installer.

Exercice 4.2 :

Pour cet exercice, nous avons choisi d'installer et de faire tourner un Docuwiki depuis un container Docker.

Installation de Docker :

Docker est un logiciel permettant de lancer des applications dans des conteneurs logiciels. Un **conteneur** est un environnement isolé qui contient tout ce dont une application a besoin pour fonctionner.

La première étape consiste à installer Docker sur notre machine.

On doit d'abord préparer un dossier et téléchargé la clé GPG officielle de Docker. Cette clé est utilisée pour vérifier l'intégrité des fichiers téléchargés pour l'installation du Docker Engine :

```

1 # Add Docker's official GPG key:
2 sudo apt-get update
3 sudo apt-get install ca-certificates curl
4 sudo install -m 0755 -d /etc/apt/keyrings
5 sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/
   keyrings/docker.asc
6 sudo chmod a+r /etc/apt/keyrings/docker.asc
7
8 # Add the repository to Apt sources:
9 echo \
10 "deb [arch=$(dpkg--print-architecture) signed-by=/etc/apt/keyrings/
   docker.asc] https://download.docker.com/linux/ubuntu/ \
11 ${.:/etc/os-release} && echo "$VERSION_CODENAME" stable" | \
12 sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

```

The screenshot shows a terminal window with a dark background and light-colored text. The title bar says 'dada@dadalinux: ~'. The terminal output is as follows:

```
[240 kB]
Get:21 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en
[105 kB]
Fetched 2,111 kB in 1s (1,721 kB/s)
Reading package lists... Done
dada@dadalinux:~$ sudo apt-get install ca-certificates curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20240203).
curl is already the newest version (8.5.0-2ubuntu10.1).
0 upgraded, 0 newly installed, 0 to remove and 12 not upgraded.
dada@dadalinux:~$ sudo install -m 0755 -d /etc/apt/keyrings
dada@dadalinux:~$ sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -
   o /etc/apt/keyrings/docker.asc
curl: (23) Failure writing output to destination
dada@dadalinux:~$ sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -
   o /etc/apt/keyrings/docker.asc
dada@dadalinux:~$ sudo chmod a+r /etc/apt/
   apt.conf.d/      preferences.d/      sources.list.d/
   auth.conf.d/     preferences.d.save/ trusted.gpg.d/
   keyrings/        sources.list
dada@dadalinux:~$ sudo chmod a+r /etc/apt/keyrings/docker.asc
dada@dadalinux:~$
```

Préparation du dossier pour l'installation du Docker Engine

```
1 # Install Docker Engine:
```

```
2 sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-
  plugin docker-compose-plugin
```

```
dada@dadalinux:~$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-
  plugin docker-compose-plugin
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
containerd.io is already the newest version (1.7.19-1).
Suggested packages:
  aufs-tools cgroups-mount | cgroup-lite
The following packages will be upgraded:
  docker-buildx-plugin docker-ce docker-ce-cli docker-ce-rootless-extras
  docker-compose-plugin
5 upgraded, 0 newly installed, 0 to remove and 7 not upgraded.
Need to get 91.6 MB of archives.
After this operation, 76.8 kB disk space will be freed.
Get:1 https://download.docker.com/linux/ubuntu/noble/stable amd64 docker-buildx-
  plugin amd64 0.16.1-1~ubuntu.24.04-noble [29.9 MB]
Get:2 https://download.docker.com/linux/ubuntu/noble/stable amd64 docker-ce-cli
  amd64 5:27.1.0-1~ubuntu.24.04-noble [14.6 MB]
Get:3 https://download.docker.com/linux/ubuntu/noble/stable amd64 docker-ce amd6
  4 5:27.1.0-1~ubuntu.24.04-noble [25.3 MB]
Get:4 https://download.docker.com/linux/ubuntu/noble/stable amd64 docker-ce-root
  less-extras amd64 5:27.1.0-1~ubuntu.24.04-noble [9,320 kB]
Get:5 https://download.docker.com/linux/ubuntu/noble/stable amd64 docker-compose
  -plugin amd64 2.29.0-1~ubuntu.24.04-noble [12.5 MB]
Fetched 91.6 MB in 4s (20.4 MB/s)
(Reading database ... 213549 files and directories currently installed.)
Preparing to unpack .../docker-buildx-plugin_0.16.1-1~ubuntu.24.04-noble_amd64.d
eb ...
Unpacking docker-buildx-plugin (0.16.1-1~ubuntu.24.04-noble) over (0.15.1-1~ubun
tu.24.04-noble) ...
Preparing to unpack .../docker-ce-cli_5%3a27.1.0-1~ubuntu.24.04-noble_amd64.deb
...
Unpacking docker-ce-cli (5:27.1.0-1~ubuntu.24.04-noble) over (5:27.0.3-1~ubuntu.
  24.04-noble) ...
Preparing to unpack .../docker-ce_5%3a27.1.0-1~ubuntu.24.04-noble_amd64.deb ...
Unpacking docker-ce (5:27.1.0-1~ubuntu.24.04-noble) over (5:27.0.3-1~ubuntu.24.0
  4-noble) ...
Preparing to unpack .../docker-ce-rootless-extras_5%3a27.1.0-1~ubuntu.24.04-nobl
e_amd64.deb ...
Unpacking docker-ce-rootless-extras (5:27.1.0-1~ubuntu.24.04-noble) over (5:27.0
  .3-1~ubuntu.24.04-noble) ...
```

Installation de Docker Engine

```
1 sudo docker run hello-world
```

On télécharge le fichier .deb de Docker Desktop depuis le site officiel de Docker.
On installe le fichier .deb avec la commande suivante :

```
1 sudo apt-get update
2 sudo apt-get install ./docker-desktop-<arch>.deb
```

On lance Docker Desktop avec la commande suivante :

```
1 sudo systemctl --user start docker-desktop
```

```
dada@dadalinux:~$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
c1ec31eb5944: Pull complete
Digest: sha256:1408fec50309afee38f3535383f5b09419e6dc0925bc69891e79d84cc4cdcec6
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

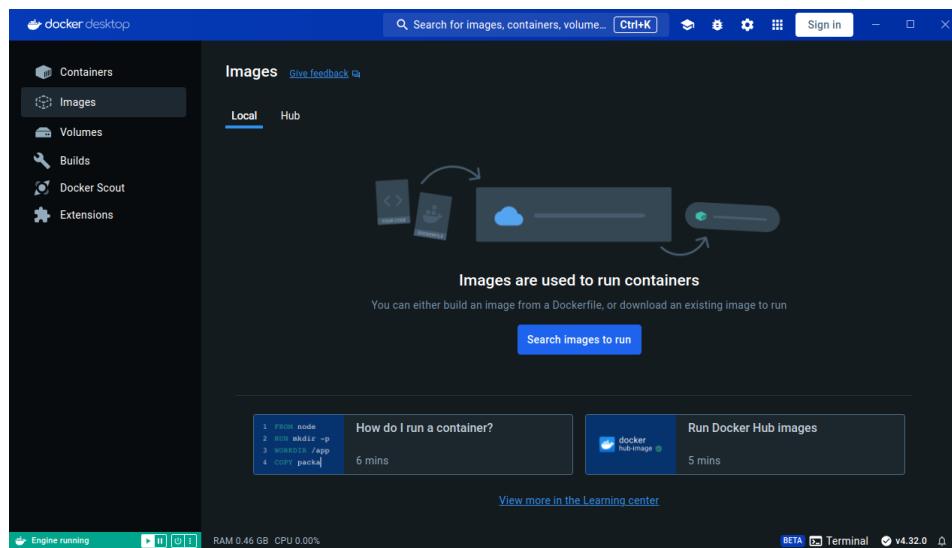
For more examples and ideas, visit:
https://docs.docker.com/get-started/
dada@dadalinux:~$
```

Lancement de l'image hello-world

```
Create and run a new container from an image
dada@dadalinux:~$ docker run hello-world
docker: Cannot connect to the Docker daemon at unix:///home/dada/.docker/desktop
/docker.sock. Is the docker daemon running?.
See 'docker run --help'.
dada@dadalinux:~$ cd Downloads/
dada@dadalinux:~/Downloads$ sudo apt-get update
Hit:1 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:2 http://fr.archive.ubuntu.com/ubuntu noble InRelease
Hit:3 http://fr.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:4 https://download.docker.com/linux/ubuntu noble InRelease
Hit:5 http://fr.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:6 https://packages.microsoft.com/repos/code stable InRelease
Hit:7 https://mirrors.ircam.fr/pub/CTAN/systems/win32/miktex/setup/deb noble InR
elease
Reading package lists... Done
dada@dadalinux:~/Downloads$ sudo apt-get install ./docker-desktop-amd64.deb
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'docker-desktop' instead of './docker-desktop-amd64.deb'
docker-desktop is already the newest version (4.32.0-157355).
0 upgraded, 0 newly installed, 0 to remove and 7 not upgraded.
dada@dadalinux:~/Downloads$
```

Installation de Docker Desktop

Dans cette section, nous avons donc installé Docker sur notre machine. Pour cela nous avons suivi les instructions du site officiel de Docker. Nous avons d'abord préparé un dossier et téléchargé la clé GPG officielle de Docker. Ensuite, nous avons installé Docker Engine et avons installé le fichier .deb de Docker Desktop. Pour terminer, nous avons lancé Docker Desktop pour nous assurer de son bon fonctionnement.



Lancement de Docker Desktop

Sources

- <https://docs.docker.com/get-docker/>
- <https://docs.docker.com/engine/install/ubuntu/#install-using-the-repository>
- <https://docs.docker.com/desktop/install/ubuntu/>

Installation de Docuwiki :

Pour cette deuxième partie, nous avons récupéré l'image Docker de **Docuwiki** depuis le site Bitnami et nous avons suivi les instructions pour l'installation.

Pour récupérer l'image Docker de Docuwiki, nous avons utilisé la commande suivante :

```
1 docker pull bitnami/dokuwiki:latest
```

The screenshot shows a terminal window with a dark background and light-colored text. The title bar says "dada@dadalinux: ~". The command entered was "docker pull bitnami/dokuwiki:latest". The output shows the image being pulled from the Bitnami repository. It includes the image name, the tag "latest", the source ("Pulling from bitnami/dokuwiki"), the digest hash, and the status message "Status: Downloaded newer image for bitnami/dokuwiki:latest". Below this, there is a "What's next:" section with a link to "docker scout quickview bitnami/dokuwiki". The prompt "dada@dadalinux:~\$ " is at the bottom.

Récupération de l'image Docker de Docuwiki

Création du conteneur :

Pour générer le conteneur à partir de l'image, nous avons utilisé un fichier docker-compose.yaml avec le contenu suivant :

```
1 ##
2 # File Name: docker-compose.yaml
3 # Description: Configuration pour déployer le DokuWiki avec Docker
4 #   Compose
5 # Author: Damien Vallet
6 # Version: 1.0
7 # Date: 2024
8 # License: Created for UOR course
9 # Language: YAML
10#
11version: '2' # version de docker-compose
12services:
```

```

13  dokuwiki: # nom du service ici docuWiki
14  image: docker.io/bitnami/dokuwiki:20240206 # image docker
15  ports:
16      - '80:8080'    # port de l'application
17      - '443:8443'
18  volumes: # volume pour la persistance des données
19      - './srcdocuwiki:/bitnami/dokuwiki2'
20  environment: # variables d'environnement
21      - DOKUWIKI_PASSWORD=toto123
22      - DOKUWIKI_USERNAME=admin
23      - DOKUWIKI_WIKI_NAME= docuWiki Sound Design

```

Pour créer le container, nous avons utilisé la commande suivante depuis le dossier contenant le fichier `docker-compose.yml` :

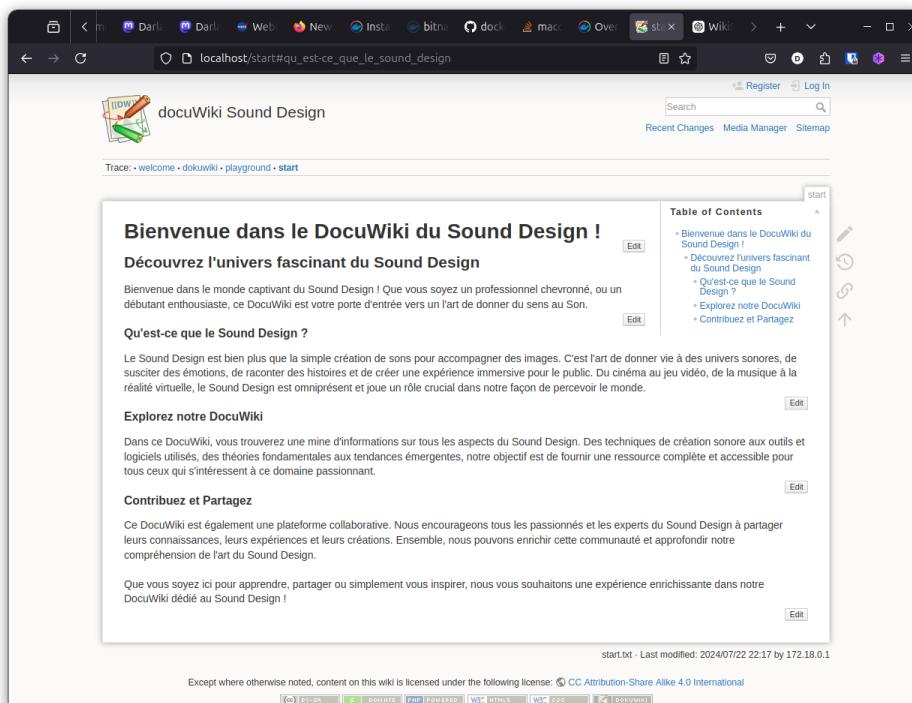
```
1 docker compose up -d
```

Avec l'option `-d` pour lancer le container en arrière plan.

On rejoint ensuite l'interface de Docuwiki en allant sur l'adresse : `http://localhost`. On peut également rejoindre depuis l'interface de Docker Desktop en cliquant sur le lien *Open in Browser*.

Création de pages dans Docuwiki :

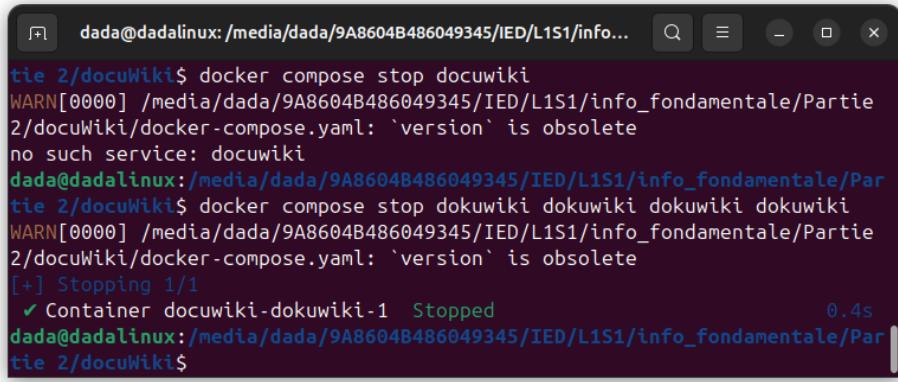
On peut désormais créer des pages, ajouter des images, des liens, des fichiers, etc.



Interface de Docuwiki avec des pages créées

On peut arrêter le conteneur avec la commande suivante :

```
1 docker-compose stop dokuwiki
```



```
dada@dadalinux:/media/dada/9A8604B486049345/IED/L1S1/info...
tie 2/docuWiki$ docker compose stop docuwiki
WARN[0000] /media/dada/9A8604B486049345/IED/L1S1/info_fondamentale/Partie
2/docuWiki/docker-compose.yaml: `version` is obsolete
no such service: docuwiki
dada@dadalinux:/media/dada/9A8604B486049345/IED/L1S1/info_fondamentale/Par
tie 2/docuWiki$ docker compose stop dokuwiki dokuwiki dokuwiki dokuwiki
WARN[0000] /media/dada/9A8604B486049345/IED/L1S1/info_fondamentale/Partie
2/docuWiki/docker-compose.yaml: `version` is obsolete
[+] Stopping 1/1
  ✓ Container docuwiki-dokuwiki-1 Stopped 0.4s
dada@dadalinux:/media/dada/9A8604B486049345/IED/L1S1/info_fondamentale/Par
tie 2/docuWiki$
```

Arrêt du conteneur Docuwiki

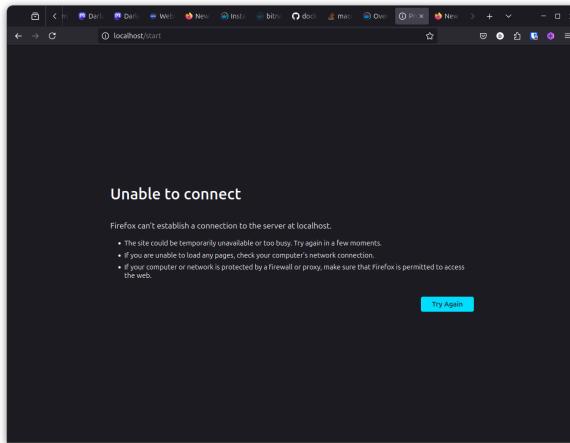
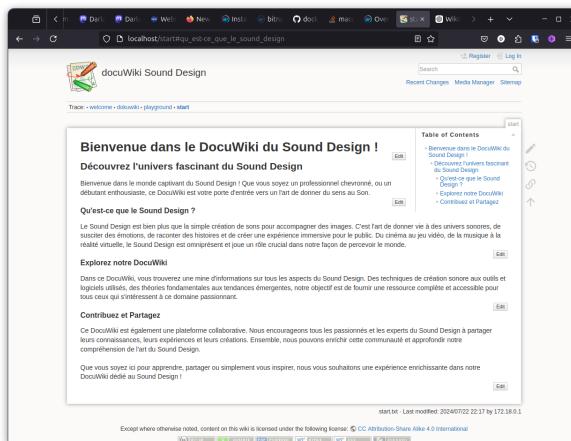


Figure 4.3: *Le Docuwiki est inaccessible après l'arrêt du conteneur

On peut relancer le conteneur avec la commande suivante :

```
1 docker-compose up -d
```



Relance du conteneur Docuwiki

Pour résumer, nous avons installé et fait tourner un **Docuwiki** depuis un container Docker.

Nous avons suivi les instructions pour l'installation de Docker et de Docuwiki.

Nous avons créé un fichier `docker-compose.yaml` pour générer le conteneur à partir de l'image.

Nous avons lancer le container et nous avons accédé à l'interface de Docuwiki en allant sur l'adresse : `http://localhost:8080`

Nous avons créé des pages, ajouté des images, des liens, des fichiers, etc.

Nous avons arrêté et relancé le conteneur avec les commandes `docker-compose stop dokewiki` et `docker-compose up -d`.

Sources

- <https://docs.docker.com/desktop/install/ubuntu/>
- <https://docs.docker.com/get-docker/>
- <https://hub.docker.com/r/bitnami/dokuwiki/>
- <https://bitnami.com/stack/dokuwiki/containers>

Exercice 4.3 (Bonus)

A réaliser en local sur votre machine, avec XAMPP, Python, NodeJS,... (selon) installé.s ou à l'aide d'une image docker (cf QR-codes pour aller plus loin) :

- Installez (une application au choix) : Apache Nifi, ElasticSearch(*)(+ kibana), une base NoSQL (MongoDB, Redis, Neo4j,Cassandra, CouchDB, etc.), un forum de discussion (Discourse, Talkyard, NodeBB (si NodeJS), PhpBB, etc.), un serveur de jeux, un serveur de Chat (ou IRC), un full-stack MERN ou MEAN, Streamlit (Python),... ou encore une seconde application web prise dans la liste de Bitnami en plus de celle de l'exercice 4.1.
- N'oubliez pas d'ajouter un contenu minimal et d'illustrer votre PDF avec les C.E(s) des différentes phrases d'installation et du rendu final.
- Remarque(*) : Au besoin, vérifiez que la version d'Elastic- Search utilisée est encore "Open Source" (licence Apache 2.0). Ce n'est peut-être plus le cas.
- Un exemple de traitement de données avec Streamlit Le vélo à Paris, qui vous sera utile pour celles et ceux qui souhaiteraient continuer en Master Big Data à l'IED.

Exercice 4.3 :

Pour réaliser cet exercice, nous avons choisi d'installer, en local, un server **Minecraft**. Pour cela, nous avons téléchargé le fichier `server.jar` depuis le site officiel de Minecraft : <https://www.minecraft.net/fr-fr/download/server>

Ensuite, nous avons créé un dossier nommé "minecraft-server" dans lequel nous avons placé le fichier `server.jar`.

Nous avons ensuite installé OpenJDK sur notre machine avec la commande suivante :

```
1 sudo apt install openjdk-21-jdk-headless
```

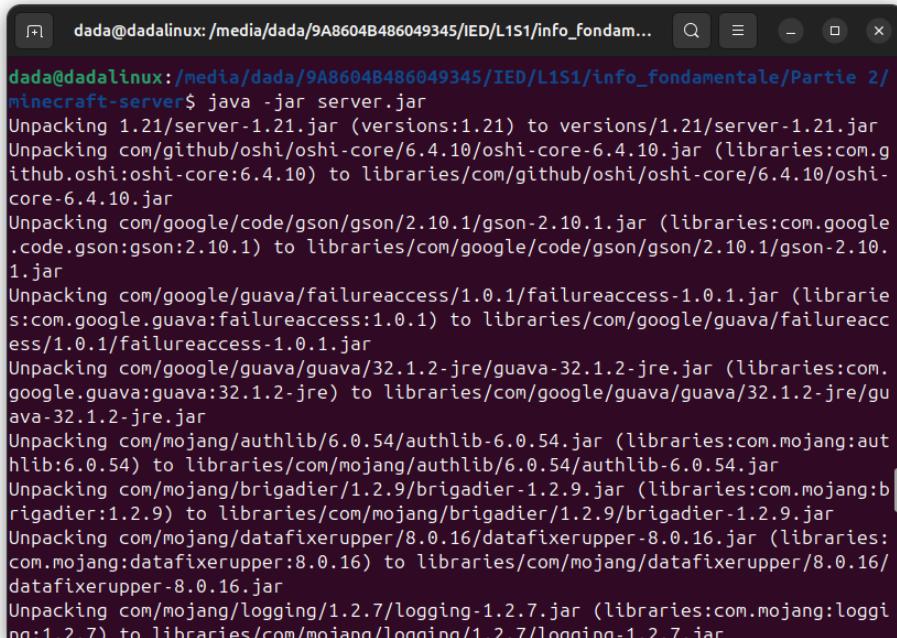
L'option `headless` permet d'installer la version de openjdk sans l'interface graphique.

```
dada@dadalinux:/media/dada/9A8604B486049345/IED/L1S1/info_fondamentale/Partie 2/minecraft-server$ sudo apt install openjdk-21-jdk-headless
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ca-certificates-java java-common openjdk-21-jre-headless
Suggested packages:
  default-jre openjdk-21-demo openjdk-21-source fonts-dejavu-extra
  fonts-ipafont-gothic fonts-ipafont-mincho fonts-wqy-microhei
  | fonts-wqy-zenhei fonts-indic
The following NEW packages will be installed:
  ca-certificates-java java-common openjdk-21-jdk-headless
  openjdk-21-jre-headless
0 upgraded, 4 newly installed, 0 to remove and 23 not upgraded.
Need to get 129 MB of archives.
After this operation, 298 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://fr.archive.ubuntu.com/ubuntu noble/main amd64 ca-certificates-java all 20240118 [11.6 kB]
Get:2 http://fr.archive.ubuntu.com/ubuntu noble/main amd64 java-common all 0.75+exp1 [6,798 B]
Get:3 http://fr.archive.ubuntu.com/ubuntu noble/main amd64 openjdk-21-jre-headless amd64 21.0.3+9-1ubuntu1 [46.6 MB]
```

Installation de openjdk

Nous avons ensuite lancé, depuis ce dossier, le fichier `server.jar` avec la commande suivante :

```
1 java -jar server.jar
```

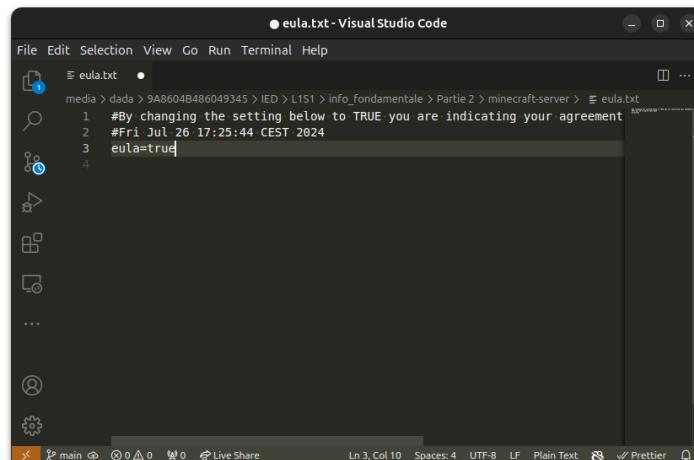


```
dada@dadalinux:/media/dada/9A8604B486049345/IED/L1S1/info_fondamentale/Partie 2/minecraft-server$ java -jar server.jar
Unpacking 1.21/server-1.21.jar (versions:1.21) to versions/1.21/server-1.21.jar
Unpacking com/github/oshi/oshi-core/6.4.10/oshi-core-6.4.10.jar (libraries:com.github.oshi:oshi-core:6.4.10) to libraries/com/github/oshi/oshi-core/6.4.10/oshi-core-6.4.10.jar
Unpacking com/google/code/gson/gson/2.10.1/gson-2.10.1.jar (libraries:com.google.code.gson:gson:2.10.1) to libraries/com/google/code/gson/gson/2.10.1/gson-2.10.1.jar
Unpacking com/google/guava/failureaccess/1.0.1/failureaccess-1.0.1.jar (libraries:com.google.guava:failureaccess:1.0.1) to libraries/com/google/guava/failureaccess/1.0.1/failureaccess-1.0.1.jar
Unpacking com/google/guava/guava/32.1.2-jre/guava-32.1.2-jre.jar (libraries:com.google.guava:guava:32.1.2-jre) to libraries/com/google/guava/guava/32.1.2-jre/guava-32.1.2-jre.jar
Unpacking com/mojang/authlib/6.0.54/authlib-6.0.54.jar (libraries:com.mojang.authlib:6.0.54) to libraries/com/mojang/authlib/6.0.54/authlib-6.0.54.jar
Unpacking com/mojang/brigadier/1.2.9/brigadier-1.2.9.jar (libraries:com.mojang.brigadier:1.2.9) to libraries/com/mojang/brigadier/1.2.9/brigadier-1.2.9.jar
Unpacking com/mojang/datafixerupper/8.0.16/datafixerupper-8.0.16.jar (libraries:com.mojang/datafixerupper:8.0.16) to libraries/com/mojang/datafixerupper/8.0.16/datafixerupper-8.0.16.jar
Unpacking com/mojang/logging/1.2.7/logging-1.2.7.jar (libraries:com.mojang.logging:logging:1.2.7) to libraries/com/mojang/logging/1.2.7/logging-1.2.7.jar
```

Installation du server Minecraft

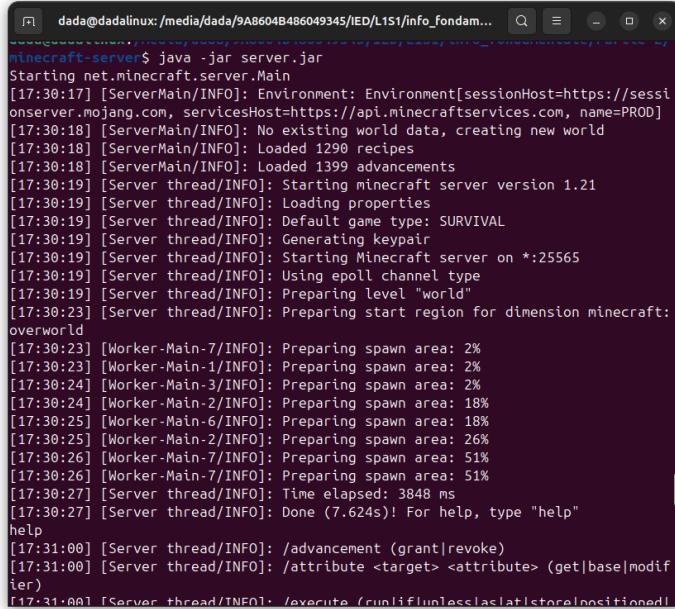
Nous avons eu une erreur car le fichier `eula.txt` devait être modifié pour accepter les conditions d'utilisation.

Nous avons donc modifié le fichier `eula.txt` en changeant la valeur de *False* à *True*.



Modification du fichier `eula.txt`

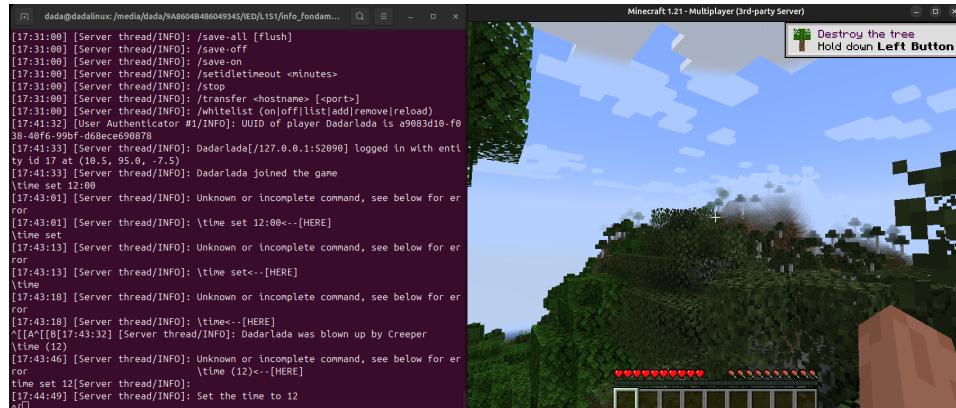
Nous avons ensuite relancé le fichier `server.jar` et le serveur Minecraft s'est lancé avec succès.



```
dada@dadalinux:/media/dada/9A8604B486049345/ED/L1S1/info_fondam...
minecraft-server$ java -jar server.jar
Starting net.minecraft.server.Main
[17:30:17] [ServerMain/INFO]: Environment: Environment[sessionHost=https://sessioner.mojang.com, servicesHost=https://api.minecraftservices.com, name=PROD]
[17:30:18] [ServerMain/INFO]: No existing world data, creating new world
[17:30:18] [ServerMain/INFO]: Loaded 1290 recipes
[17:30:18] [ServerMain/INFO]: Loaded 1399 advancements
[17:30:19] [Server thread/INFO]: Starting minecraft server version 1.21
[17:30:19] [Server thread/INFO]: Loading properties
[17:30:19] [Server thread/INFO]: Default game type: SURVIVAL
[17:30:19] [Server thread/INFO]: Generating keypair
[17:30:19] [Server thread/INFO]: Starting Minecraft server on *:25565
[17:30:19] [Server thread/INFO]: Using epoll channel type
[17:30:19] [Server thread/INFO]: Preparing level "world"
[17:30:23] [Server thread/INFO]: Preparing start region for dimension minecraft:overworld
[17:30:23] [Worker-Main-7/INFO]: Preparing spawn area: 2%
[17:30:23] [Worker-Main-1/INFO]: Preparing spawn area: 2%
[17:30:24] [Worker-Main-3/INFO]: Preparing spawn area: 2%
[17:30:24] [Worker-Main-2/INFO]: Preparing spawn area: 18%
[17:30:25] [Worker-Main-6/INFO]: Preparing spawn area: 18%
[17:30:25] [Worker-Main-2/INFO]: Preparing spawn area: 26%
[17:30:26] [Worker-Main-7/INFO]: Preparing spawn area: 51%
[17:30:26] [Worker-Main-7/INFO]: Preparing spawn area: 51%
[17:30:27] [Server thread/INFO]: Time elapsed: 3848 ms
[17:30:27] [Server thread/INFO]: Done (7.624s)! For help, type "help"
help
[17:31:00] [Server thread/INFO]: /avancement (grant|revoke)
[17:31:00] [Server thread/INFO]: /attribute <target> <attribute> (get|base|modifier)
[17:31:00] [Server thread/INFO]: /execute (runif|unless|as|at|store|positioned|
```

Le server Minecraft est lancé avec succès

On peut maintenant se connecter au serveur Minecraft en utilisant la connexion directe avec l'adresse localhost.



Connexion au server Minecraft

Oui, un creeper a explosé à coté de moi pendant que je prenais la capture d'écran :=

Pour résumer, nous avons installé un server **Minecraft** sur notre machine. Pour cela nous avons téléchargé et lancé le fichier server.jar.
Nous avons installé la version 21 de openjdk.
Nous avons relancé le fichier server.jar et le serveur Minecraft s'est lancé avec succès.
Nous avons pu nous connecter au serveur Minecraft en utilisant localhost.

Sources

- <https://www.minecraft.net/fr-fr/download/server>
- https://minecraft.wiki/w/Tutorials/Setting_up_a_server#OpenJDK_vs_Oracle_JDK

5

Chapitre 5 - Lire et traiter des informations via une page Web : formulaires, PHP, MySQL

Exercice 5.1 (À Rendre)

- Ajoutez à votre page personnelle un **formulaire** comportant à minima **5 champs différents**. Traitez-le avec du **PHP** (ou avec le **langage de votre choix**) en utilisant obligatoirement **une base de données relationnelle** (MySQL ou équivalent).
- Pour que ce soit intéressant, il faut que la page écrive des données dans la base mais aussi qu'elle les relise pour affichage. Un exemple simple est un livre d'or, une demande de contacts, un 'laissez votre avis', ..., qui permettra de laisser données et informations sur la page et bien évidemment **d'afficher celles-ci pour validation**.
- Pour vous aider, vous pouvez utiliser un outil d'administration comme PhpMyAdmin (ou équivalent). Pensez à rendre le plus sûr possible les saisies de l'utilisateur notamment sur **les champs "date" et "adresse email"** qui devront apparaître dans votre formulaire.
- N'oubliez pas d'expliquer les requêtes utilisées (SQL ou équivalent), les traitements opérés, de réaliser les C.E(s) d'installation et d'illustration du fonctionnement de votre formulaire et de joindre toutes les sources à votre dossier Htdocs en archive attachée à votre envoi.

Exercice 5.1 - Réalisation du Formulaire

Modification de NavigationHeader.js

Pour réaliser cette exercice, nous avons ajouter une nouvelle page intitulée **VersionPHP.jsx** à notre site web ainsi qu'une nouvelle route pour accéder à cette page.

Voici le code du fichier **NavigationHeader.js** pour ajouter cette nouvelle Route vers le composant.

```

1 /**
2 * File Name: NavigationHeader.js
3 * Description: Navigation entre versions
4 * Author: Damien Vallet
5 * Version: 1.0
6 * Date: 2024
7 * License: Created for UOR course
8 * Language: JavaScript
9 */
10
11 // Importation de la librairie React
12 import React from "react";
13
14 // Importation de la librairie React Router
15 import { BrowserRouter as Router, Route, Routes } from "react-router-dom";
16
17 // Importation des composants;
18 import NavigationHeaderBar from "./components/NavigationHeaderBar";
19 import VersioHTML from "./components/VersionHTML";
20 import VersionCSS from "./components/VersionCSS";
21 import VersionPHP from "./components/VersionPHP";
22
23 // Définition de la fonction NavigationHeader
24 function NavigationHeader() {
25     return (
26         //
27         <>
28             {/* composant de navigation */}
29             <Router>
30                 <div>
31
32                     {/* Barre de navigation */}
33                     <NavigationHeaderBar />
34
35                     {/* definition des routes */}
36                     <Routes>
37
38                         {/* route vers la version HTML */}
39                         <Route path="/html" element={<VersioHTML />} />
40
41                         {/* route vers la version CSS */}
42                         <Route path="/css" element={<VersionCSS />} />
43
44                         {/* route vers la version PHP */}
45                         <Route path="/php" element={<VersionPHP />} />
46
47                     </Routes>

```

```

48         </div>
49     </Router>
50     </>
51   );
52 }
53
54 // Exportation du composant NavigationHeader
55 export default NavigationHeader;

```

Modification du fichier NavigationHeaderBar.jsx

Nous avons également modifié le fichier `NavigationHeaderBar.jsx` pour ajouter un lien vers la nouvelle page.

```

1 /**
2 * File Name: NavigationHeaderBar.jsx
3 * Description: Composant qui gère la barre pour naviguer entre les
4 *               versions
5 * Author: Damien Vallet
6 * Version: 1.0
7 * Date: 2024
8 * License: Created for UOR course
9 * Language: JavaScript
10 */
11
12 // Importation de la librairie React
13 import React from "react";
14
15 // Importation de la librairie React Router
16 import { Link } from "react-router-dom";
17
18 // définition du composant NavBar
19 function NavigationHeaderBar() {
20   return (
21
22     // Barre de navigation
23     <nav id="header-navbar">
24
25       /* Une liste non ordonnée (<ul>) définit les éléments de navigation
26          . Chaque élément de navigation est un élément de liste (<li>)*/
27       <ul>
28         <li>
29           /* Lien vers la version HTML */
30           <Link to="/html">Version HTML</Link>
31         </li>
32         <li>
33           /* Lien vers la version avec CSS/Bootstrap */
34           <Link to="/css">Version CSS</Link>
35         </li>
36         <li>
37           /* lien vers la version PHP avec base SQL */
38           <Link to="/php">Version PHP</Link>
39         </li>
40

```

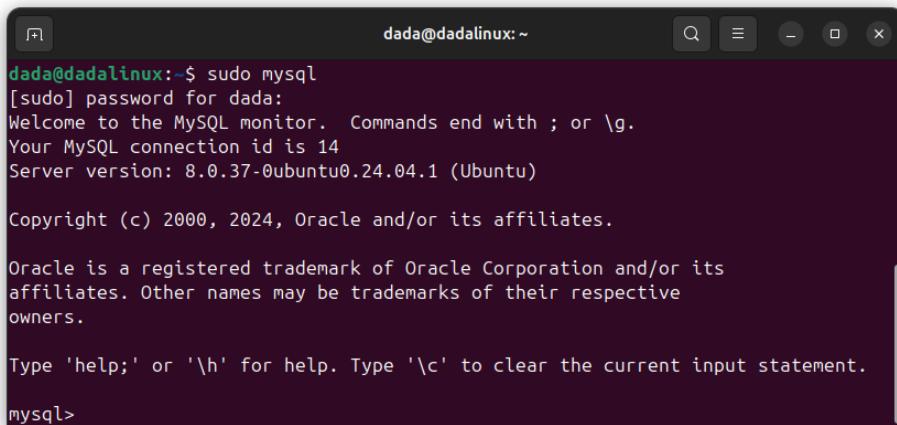
```
41      </ul>
42    </nav>
43  );
44 }
45
46 // Exportation du composant NavigationHeaderBar
47 export default NavigationHeaderBar;
```

Configuration de la base de donnée MySQL

Pour ouvrir MySQL en local, nous avons utilisé la commande suivante :

```
1 sudo mysql
```

Ceci nous a permis de nous connecter à MySQL en tant qu'utilisateur root.



```
dada@dadalinux:~$ sudo mysql
[sudo] password for dada:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 14
Server version: 8.0.37-0ubuntu0.24.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

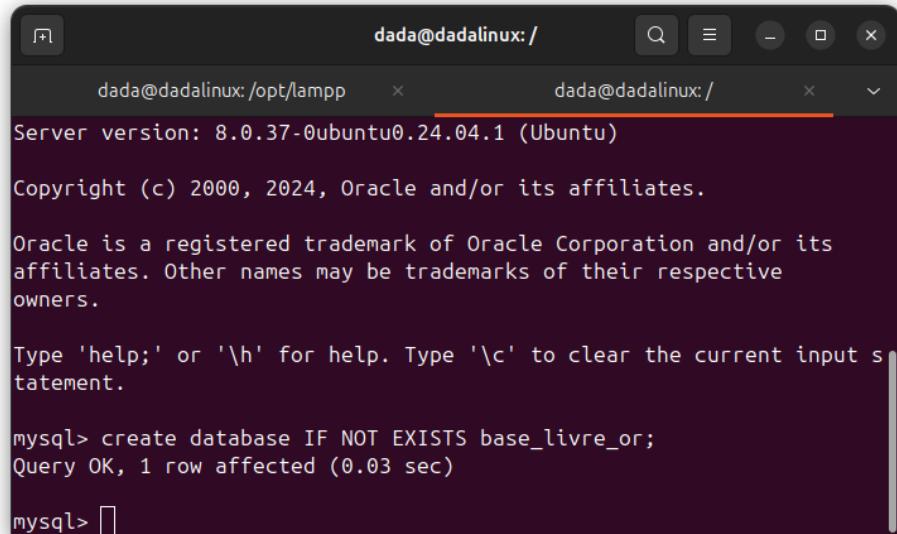
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Connexion à MySQL en tant qu'utilisateur root

Nous avons ensuite créé une base de données **base_livre_or** avec la commande suivante:

```
1 create database IF NOT EXISTS base_livre_or;
```



```
dada@dadalinux:/opt/lampp$ dada@dadalinux:/$
Server version: 8.0.37-0ubuntu0.24.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database IF NOT EXISTS base_livre_or;
Query OK, 1 row affected (0.03 sec)

mysql> []
```

Création de la base de données base_livre_or

On a ensuite créé un utilisateur pour la base de données avec la commande suivante :

```

1 // On crée un utilisateur 'dada' avec le mot de passe 'guestbook!' \\
2 CREATE USER 'dada'@'localhost' IDENTIFIED BY 'guestbook!';
3
4 // On donne tous les priviléges à l'utilisateur 'dada' sur la base de donn
5 ées base\_\_livre\_\_or\\\
6 GRANT ALL PRIVILEGES ON base_livre_or.* TO 'dada'@'localhost';
7
8 // On recharge les priviléges pour que les changements prennent effet\\\
9 FLUSH PRIVILEGES;

```

On se connecte ensuite à la base de données **base_livre_or** avec la commande suivante :

```
1 use base_livre_or;
```

```

dada@dadalinux:/opt/lampp
dada@dadalinux: /          dada@dadalinux: /
mysql> show databases;
+-----+
| Database |
+-----+
| base_livre_or |
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.02 sec)

mysql> use base_livre_or ;
Database changed
mysql> ^C
mysql> 
```

Connexion à la base de données **base_livre_or**

Pour notre livre d'or, nous avons imaginé un formulaire avec les champs suivants :

- Nom de l'utilisateur qui laisse le commentaire
- Email de l'utilisateur
- Age de l'utilisateur
- Genre de l'utilisateur
- Date du commentaire
- Message à laisser

Pour que la base de données puisse recevoir nos données, et les transmettre à notre application, nous avons créé une table **commentaires** avec les champs suivants :

```

1 # On crée la table commentaires si elle n'existe pas
2 CREATE TABLE IF NOT EXISTS commentaires (
3     # On crée un champ id qui sera une clé unique pour chaque commentaire
4     id INT(4) UNIQUE AUTO_INCREMENT PRIMARY KEY,
5
6     # On crée un champ nom pour stocker le nom de l'utilisateur avec une
7     # taille maximale de 50 caractères
8     nom VARCHAR(50) NOT NULL,
9
10    # On crée un champ email pour stocker l'email de l'utilisateur avec
11    # une taille maximale de 50 caractères
12    email VARCHAR(50) NOT NULL,
13
14    # On crée un champ age pour stocker l'age de l'utilisateur avec une
15    # taille maximale de 3 caractères et une valeur par défaut de 0
16    age INT(3) DEFAULT 0,
17
18    # On crée un champ genre pour stocker le genre de l'utilisateur avec
19    # une taille maximale de 10 caractères et une valeur par défaut de 0
20    gender VARCHAR(10) NOT NULL,
21
22    # On crée un champ date pour stocker la date du commentaire
23    date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
24
25    # On crée un champ commentaire pour stocker le message de l'
26    # utilisateur
27    commentaire TEXT NOT NULL
28 );

```

- PRIMARY KEY permet de spécifier que le champ est une clé primaire.
- UNIQUE permet de rendre le champ unique.
- AUTO_INCREMENT permet de spécifier que le champ est auto-incrémenté.
- NOT NULL permet de dire que le champ ne peut pas être vide.
- TIMESTAMP permet de spécifier que le champ est de type date.
- DEFAULT permet de donner une valeur par défaut pour le champ.
- CURRENT_TIMESTAMP permet de spécifier que le champ prendra la valeur actuelle de TIMESTAMP.

Pour résumer, nous avons utilisé des commandes pour créer une base de données, MySQL, **base_livre_or** et une table **commentaires** pour stocker les données du formulaire.

```

dada@dadalinux:/opt/lampp      dada@dadalinux:/
mysql> CREATE TABLE IF NOT EXISTS commentaires (
    ->     id INT(4) UNIQUE AUTO_INCREMENT PRIMARY KEY,
    ->     nom VARCHAR(50) NOT NULL,
    ->     email VARCHAR(50) NOT NULL,
    ->     age INT(3) DEFAULT 0,
    ->     gender VARCHAR(10) NOT NULL,
    ->     date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    ->     commentaire TEXT NOT NULL
    -> );
Query OK, 0 rows affected, 2 warnings (0.03 sec)

mysql> show tables;
+-----+
| Tables_in_base_livre_or |
+-----+
| commentaires            |
+-----+
1 row in set (0.01 sec)

mysql> describe commentaires;
+-----+-----+-----+-----+-----+
| Field   | Type    | Null | Key | Default   | Extra       |
+-----+-----+-----+-----+-----+
| id      | int     | NO  | PRI | NULL      | auto_increment |
| nom     | varchar(50)| NO |     | NULL      |
| email   | varchar(50)| NO |     | NULL      |
| age     | int     | YES |     | 0          |
| gender  | varchar(10)| NO |     | NULL      |
| date    | timestamp| YES |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
| commentaire | text | NO |     | NULL      |
+-----+-----+-----+-----+-----+
7 rows in set (0.01 sec)

mysql> 
```

Création de la table commentaires

Sources

- <https://dev.mysql.com/doc/refman/8.4/en/tutorial.html>
- <https://dev.mysql.com/doc/refman/8.4/en/example-auto-increment.html>
- <https://dev.mysql.com/doc/refman/8.4/en/data-type-defaults.html>
- <https://dev.mysql.com/doc/refman/8.4/en/date-and-time-functions.html>

Création du composant Formulaire.jsx

Nous avons créé un composant **Formulaire.jsx** pour afficher le formulaire et les commentaires. Dans un premier temps, nous allons présenter la partie formulaire (balise <form>) de ce composant.

```

1  /* <!--on envoie les données du formulaire la fonction handleSubmit--> */
2  }
3
4  /* champs "nom" */
5  <form onSubmit={handleSubmit}>
6
7      <div className="formulaire">
8          <label htmlFor="nom">n m :</label>
9          <input type="text" name="nom" id="nom" placeholder="toto" required
10         onChange={handleChange}
11         value={formDatas.nom}
12     />
13  </div>
14
15  /* champs "commentaire" */
16  <div className="formulaire">
17      <label htmlFor="commentaire">Message :</label>
18      <textarea name="commentaire" id="commentaire" placeholder="Entrez votre message ici :"
19          required
20          onChange={handleChange}
21          value={formDatas.commentaire}
22      />
23  </div>
24
25  /* champs "email" */
26  <div className="formulaire">
27      <label htmlFor="email">Email :</label>
28      <input type="email" name="email" id="email" placeholder="toto@zero
29          .com" required
30          onChange={handleChange}
31          value={formDatas.email}
32      />
33  </div>
34
35  /* champs "genre" */
36  <div className="radio-container">
37      <label htmlFor="genre">Genre :</label>
38      <label htmlFor="homme">Homme</label>
39      <input type="radio" name="genre" defaultValue="h"
40          onChange={handleChange}
41      />
42      <label htmlFor="femme">Femme</label>
43      <input type="radio" name="genre" defaultValue="f"
44          onChange={handleChange}
45      />
46      <label htmlFor="non-binaire">Non-binaire</label>
47      <input type="radio" name="genre" defaultValue="n" defaultChecked
48          onChange={handleChange}
49      />
50  </div>
```

```

49  {/* champs "age" */}
50  <div className="formulaire">
51    <label htmlFor="age">Age :</label>
52    <input type="number" name="age" id="age" placeholder="18" required
53      onChange={handleChange}
54      value={formDatas.age}
55    />
56  </div>
57
58  {/* bouton d'envoi */}
59  <div className="formulaire">
60    <input type="submit" defaultValue="Envoyer" />
61  </div>
62 </form>

```

Dans cette partie, nous avons présenté le formulaire du composant **Formulaire.jsx**.

Nous avons encapsulé notre formulaire dans une balise **<form>** qui permet d'envoyer les données du formulaire à un fichier de traitement.

Nous avons utilisé la balise **<input>** avec les types : text, radio, number pour recevoir les données de l'utilisateur. Le type submit permet d'envoyer les données du formulaire.

Nous avons utilisé la balise **<textarea>** pour recevoir les messages de l'utilisateur. On a utilisé la fonction **onChange** appeler la fonction **handleChange** et mettre à jour les données du formulaire à chaque changement.

Sources

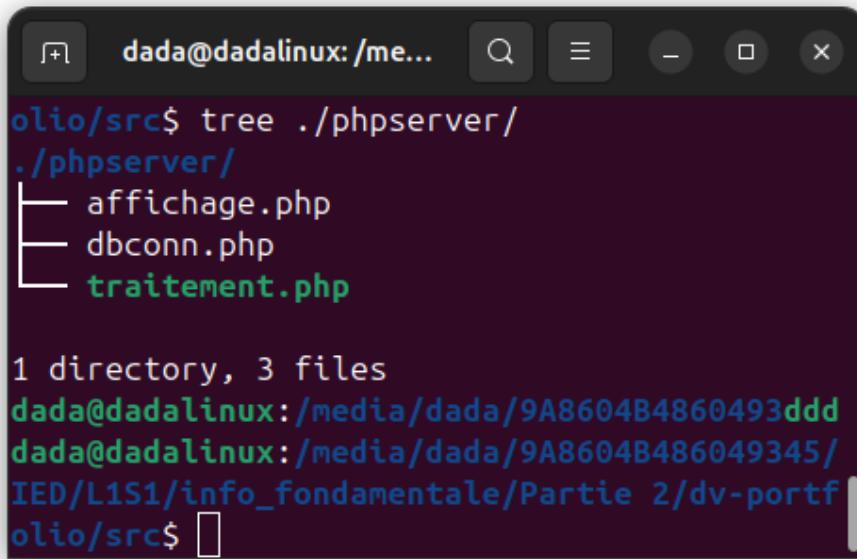
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Form>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/input/submit>
- https://www.w3schools.com/react/react_forms.asp

React et les hooks

Dans React, un hook est une fonction qui permet d'ajouter des fonctionnalités à un composant. Dans ce composant, nous avons utilisé le hook **useState** de React pour gérer l'état du formulaire. Le hook **useEffect** permet d'effectuer des actions à chaque rendu du composant et nous l'avons utilisé pour afficher les commentaires.

JSX étant utilisé côté client, il ne peut devrait pas être utilisé pour traiter les données du formulaire. Pour éviter de mixer le PHP et le JSX, et pour plus de clarté, nous avons créé un dossier appelé **phpserver** qui sera notre serveur pour traiter les données du formulaire.

Voici la structure de notre dossier **phpserver** :



```
olio/src$ tree ./phpserver/
./phpserver/
└── traitement.php
    ├── affichage.php
    └── dbconn.php

1 directory, 3 files
dada@dadalinux:/media/dada/9A8604B4860493ddd
dada@dadalinux:/media/dada/9A8604B486049345/
IED/L1S1/info_fondamentale/Partie 2/dv-portf
olio/src$ 
```

Structure du dossier **phpserver**

Notre serveur sera constitué de trois fichiers :

- **dbconn.php** : pour se connecter à la base de données.
- **traitement.php** : pour traiter les données du formulaire.
- **affichage.php** : pour afficher les commentaires de la base de données.

Création du fichier dbconn.php

Dans ce dossier, nous avons créé un fichier **dbconn.php** qui contiendra les fonctions pour se connecter à la base de données.

Voici le code de ce fichier :

```

1 <?php
2 /**
3 // File Name: dbconn.php
4 // Description: Connexion à la base de données
5 // Author: Damien Vallet
6 // Version: 1.0
7 // Date: 2024
8 // License: Created for UOR course
9 // Language: PHP
10 /**
11
12 // on définit l'entête de la réponse en JSON
13 header("Access-Control-Allow-Origin:_*");
14 header("Access-Control-Allow-Methods:_*");
15 header("Access-Control-Allow-Headers:_*");
16 header("Content-Type:_application/json");
17
18 // on définit les variables de connexion à la base de données
19 $servername = "localhost";
20 $username = "dada";
21 $password = "guestbook!";
22 $dbname = "base_livre_or";
23
24 // Connexion à la base de données
25 try{
26     // on se connecte à la base de données
27     $conn = new mysqli($servername, $username, $password, $dbname);
28
29
30     // on vérifie la connexion
31     if ($conn->connect_error) {
32         die("Connection failed:_" . $conn->connect_error);
33     }
34     // on définit l'encodage des caractères
35     $conn->set_charset("utf8");
36
37 } catch (Exception $e) {
38     // en cas d'erreur, on affiche un message d'erreur
39     echo "Erreur_de_connexion_à_la_base_de_données";
40     echo $e->getMessage();
41     exit();
42 }
43 ?>

```

Dans ce fichier, nous avons créé une connexion à la base de données MySQL.

- **mysqli()** pour se connecter à la base de données.
- **set_charset()** pour définir l'encodage des caractères.
- **die()** pour arrêter le script et afficher un message d'erreur.
- **try...catch** pour gérer les erreurs de connexion.

Sources

- <https://www.php.net/manual/fr/mysqli.construct.php>
- <https://www.php.net/manual/fr/mysqli.set-charset.php>
- <https://www.php.net/manual/fr/language.exceptions.php>
- <https://www.php.net/manual/fr/function.die.php>
- https://www.w3schools.com/php/php_ref_mysqli.asp
- https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS/Errors/CORS_MissingAllowOrigin?utm_source=devtools&utm_medium=firefox-cors-errors&utm_campaign=default

Création du fichier traitement.php

Une fois notre connexion établie, nous allons maintenant traiter les données du formulaire. Pour cela, nous avons créé un fichier **traitement.php** dans le dossier **phpserver** de notre application. Voici le code de ce fichier :

```

1 <?php
2 /**
3 // File Name: traitement.php
4 // Description: Enregistre les commentaires dans la base de données
5 // Author: Damien Vallet
6 // Version: 1.0
7 // Date: 2024
8 // License: Created for UOR course
9 // Language: PHP
10 /**
11
12 // on définit l'entête de la réponse en JSON
13 header("Access-Control-Allow-Origin: *");
14 header("Access-Control-Allow-Methods: POST");
15 header("Access-Control-Allow-Headers: Content-Type");
16 header("Content-Type: application/json");
17
18 // Inclusion du fichier de connexion à la base de données
19 include 'dbconn.php';
20
21 // On récupère les données du formulaire en JSON
22 $data = json_decode(file_get_contents('php://input'), true);
23
24 // On vérifie si les données sont bien présentes
25 if (isset($data['nom']) && isset($data['email']) && isset($data['age']) &&
26     isset($data['genre']) && isset($data['commentaire'])) {
27
28 // On récupère les données du formulaire et on les protège contre les
29 // injections SQL
30 $nom = htmlspecialchars($data['nom']);
31 $email = htmlspecialchars($data['email']);
32 $age = htmlspecialchars($data['age']);
33 $genre = htmlspecialchars($data['genre']);
34 $commentaire = htmlspecialchars($data['commentaire']);
35
36 // On vérifie si l'email est valide
37 if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
38     echo json_encode(['success' => false, 'message' => 'Email invalide
39     ']);
40     exit();
41 }
42
43 // On prépare la requête SQL avec la méthode prepare qui permet de
44 // protéger contre les injections SQL
45 $sql = $conn->prepare("INSERT INTO commentaires (nom, email, age,
46     gender, commentaire) VALUES (?, ?, ?, ?, ?)");
47
48 // On utilise la fonction bindParam pour associer les valeurs aux
49 // paramètres de la requête SQL
50 $sql->bindParam("ssiss", $nom, $email, $age, $genre, $commentaire);

```

```

45 // On exécute la requête SQL
46 if ($sql->execute()) {
47     // si la requête a bien été exécutée, on renvoie un message de
48     // succès
49     echo "Votre commentaire a bien été enregistré";
50     echo json_encode(['success' => true, 'Votre commentaire a bien été
51     enregistré']);
52 } else {
53     // si la requête n'a pas été exécutée, on renvoie un message d'
54     // erreur
55     echo json_encode(['success' => false, 'Erreur lors de l\'enregistrement de votre commentaire']);
56 }
57 } else {
58     // si tous les champs ne sont pas remplis, on renvoie un message d'
59     // erreur
60     echo json_encode(['success' => false, 'message' => 'Veuillez remplir
61     tous les champs']);
62 }
63 ?>

```

Dans ce fichier, nous avons créé un script PHP pour traiter les données du formulaire.

- Pour récupérer et décoder les données du formulaire en JSON, nous avons utilisé la fonction `file_get_contents()` et la fonction `json_decode()`.
- Ensuite, nous avons protégé les données contre les injections SQL avec la fonction `htmlspecialchars()` et la méthode `prepare()`.
- Nous avons validé l'email avec la fonction `filter_var()`. Enfin, nous avons associé les valeurs aux paramètres de la requête SQL avec la fonction `bind_param()` et nous avons exécuté la requête SQL avec la fonction `execute()`.
- Afin d'encoder les données en JSON pour les renvoyer au client, nous avons utilisé la fonction `json_encode()`.

Sources

- https://www.w3schools.com/php/php_forms.asp
- <https://www.php.net/manual/fr/function.file-get-contents.php>
- <https://www.php.net/manual/fr/mysqli.quickstart.prepared-statements.php>
- <https://www.php.net/manual/fr/mysqli.prepare.php>
- https://www.w3schools.com/php/php_ref_json.asp
- <https://www.php.net/manual/fr/function.isset.php>
- https://www.w3schools.com/php/php_form_validation.asp
- <https://www.php.net/manual/fr/function.htmlspecialchars.php>
- <https://www.php.net/manual/fr/function.filter-var.php>
- <https://www.php.net/manual/fr/mysql Stmt.bind-param.php>
- <https://www.php.net/manual/fr/mysql Stmt.execute.php>

Création du fichier affichage.php

Enfin, nous avons créé un fichier **affichage.php** pour récupérer les commentaires de la base de données et les afficher.

Voici le code de ce fichier :

```

1 <?php
2 /**
3 // Nom ..... : affichage.php
4 // Rôle ..... : Affiche les commentaires du livre d'or
5 // Auteur ..... : Damien Vallet
6 // Version ..... : V0.1 du 2024
7 // Licence ..... : réalisé dans le cadre du cours de UOR
8 //
9
10 // on définit l'entête de la réponse en JSON
11 header("Access-Control-Allow-Origin: *");
12 header("Access-Control-Allow-Methods: GET");
13 header("Access-Control-Allow-Headers: Content-Type");
14 header("Content-Type: application/json");
15
16 // Inclusion du fichier de connexion à la base de données
17 include 'dbconn.php';
18
19 // On prépare la requête SQL pour récupérer les commentaires
20 $sql = $conn->prepare("SELECT * FROM commentaires ORDER BY id DESC");
21
22 // On exécute la requête SQL
23 $sql->execute();
24
25 // On récupère les résultats de la requête SQL
26 $result = $sql->get_result();
27
28 // Initialisation d'un tableau pour les résultats
29 $comments = array();
30
31 // On vérifie si des résultats ont été retournés
32 if ($result->num_rows > 0) {
33     // On parcourt tous les résultats et on les ajoute au tableau
34     while ($row = $result->fetch_assoc()) {
35         $comments[] = $row;
36     }
37 }
38
39 // On encode le tableau en JSON
40 echo json_encode($comments);
41
42 // On ferme la connexion à la base de données
43 $conn->close();
44
45 ?>
```

Dans ce fichier, nous avons créé un script PHP pour récupérer les commentaires de la base de données et les encoder en JSON.

- La fonction `get_result()` permet de récupérer les résultats de la requête SQL.
- La fonction `fetch_assoc()` permet de récupérer une ligne de résultat sous forme de tableau associatif.
- La fonction `json_encode()` permet de convertir un tableau en JSON.
- La fonction `close()` permet de fermer la connexion à la base de données.
- La commande SQL `ORDER BY id DESC` permet de trier les résultats par ordre décroissant.

Sources

- <https://www.php.net/manual/fr/mysqli-stmt.get-result.php>
- <https://www.php.net/manual/fr/mysqli-result.fetch-assoc.php>
- <https://www.php.net/manual/fr/function.json-encode.php>
- <https://www.php.net/manual/fr/mysqli.close.php>

Nous allons voir comment nous avons pris en charge les données envoyées par le formulaire vers le serveur PHP.

Fonction handleChange

Nous avons crée une fonction **handleChange** pour mettre à jour les données du formulaire à chaque changement.

```

1 // on crée une fonction pour gérer les changements dans les champs du
2 const handleChange = (e) => {
3
4     // on récupère le nom et la valeur du champs
5     const { name, value } = e.target;
6
7     // on met à jour les données du formulaire
8     setFormDatas({
9         ...formData,
10        [name]: value,
11    });
12 };

```

Dans cette fonction, nous avons mis à jour la table **formData** avec la fonction **setFormDatas()** avec les données du formulaire.

fonction handleSubmit

Pour cela, nous avons crée une fonction **handleSubmit** pour envoyer les données du formulaire à la base de données.

```

1 // on gère l'envoi du formulaire
2 const handleSubmit = (e) => {
3
4     // on empêche le comportement par défaut du formulaire qui est de
5     // recharger la page
6     e.preventDefault();
7
8     // on appelle la fonction ajouteMessage avec les données du formulaire
9     ajouteMessage(formData);
10
11    // on réinitialise les données du formulaire
12    setFormDatas({
13        nom: '',
14        commentaire: '',
15        email: '',
16        genre: 'n',
17        age: ''
18    });
19 };

```

Dans cette fonction, nous avons utilisé la fonction `preventDefault()` pour empêcher le comportement par défaut du formulaire. En effet, le comportement par défaut du formulaire est de recharger la page au moment de l'envoi.

Nous avons appelé la fonction `ajouteMessage()` avec les données du formulaire. Nous avons utilisé la fonction `setFormDatas()` pour réinitialiser les données du formulaire.

fonction ajouteMessage

Nous avons créé une fonction `ajouteMessage` pour envoyer les données du formulaire au serveur php. Une fois les données envoyées, nous récupérons immédiatement les commentaires de la base de données pour les afficher. Ainsi, notre page web est mise à jour en temps réel.

Voici le code de cette fonction :

```

1 // on crée une fonction pour ajoute un message
2 const ajouteMessage = async (message) => {
3     // on envoie les données du formulaire au serveur
4     const reponse = await fetch('http://localhost:8080/src/phpserver/
5         traitement.php', {
6             method: 'POST',
7             headers: {
8                 'Content-Type': 'application/json'
9             },
10            body: JSON.stringify(message),
11        });
12
13        // on vérifie si la requête a fonctionné
14        if (!reponse.ok) {
15            console.log('Une erreur est survenue');
16            return;
17        }
18        // on récupère la réponse du serveur
19        else {
20            // on récupère les messages
21            recupererMessages();
22        }
23    };

```

Dans cette fonction, nous avons utilisé la fonction `fetch()` pour envoyer les données du formulaire au serveur.

- Nous avons utilisé le mot clé `async` pour rendre la fonction asynchrone. Cela nous permet d'utiliser le mot clé `await` pour attendre la réponse ("promesse") du serveur.
- Nous avons utilisé la méthode `POST` pour envoyer les données.
- Nous avons utilisé l'entête `Content-Type` pour spécifier le type de contenu.
- Nous avons utilisé la fonction `JSON.stringify()` pour convertir les données en JSON.
- Nous avons utilisé la fonction `ok` pour vérifier si la requête a fonctionné.

Sources

- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/JSON/stringify
- https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch

fonction recupererMessages

L'affichage des commentaires se fait en utilisant le hook **useEffect** de React. Ce hook permet d'effectuer des actions à chaque rendu du composant. Ici à chaque rendu du composant, le **useEffect** va appeler la fonction **recupererMessages** qui va récupérer les commentaires de la base de données.

```

1 // on crée une fonction pour récupérer les messages
2 const recupererMessages = async () => {
3     // on récupère les messages du serveur
4     const reponse = await fetch('http://localhost:8080/src/phpserver/
      affichage.php')
5
6     // on récupère les données de la réponse
7     const data = await reponse.json();
8
9     // on met à jour le tableau de messages
10    setMessages(data);
11 }
```

- Dans cette fonction, nous avons utilisé la fonction **fetch()** pour récupérer les commentaires de la base de données.
- Nous avons utilisé le mot clé **async** pour rendre la fonction asynchrone. Cela nous permet d'utiliser le mot clé **await** pour attendre la réponse ("promesse") du serveur.
- Nous avons utilisé la fonction **json()** pour convertir les données en JSON.
- Nous avons appelé la fonction **setMessages()** pour mettre à jour le tableau de messages.

Sources

- [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/JSON\(parse](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/JSON(parse)
- <https://developer.mozilla.org/en-US/docs/Web/API/Request/json>

Affichage des commentaires

Dans notre fichier **Formulaire.jsx**, nous avons utilisé la fonction **map()** pour afficher les commentaires. Nous avons stylisé les commentaires avec Bootstrap.

```

1 <section id="messages">
2   <h2 className="title_p-3">Messages</h2>
3   <div className="col-12_p-3">
4
5     {/* <!-- on affiche les messages si il y en a --> */}
6     {Array.isArray(messages) && messages.length > 0 ? (
7
8       // on parcourt le tableau de messages
9       messages.map((message, index) => (
10
11         // pour chaque message on crée une carte
12         <div key={index} className="card_mb-3">
13
14           {/* // on affiche les données du message */}
15           <div className="card-header_d-flex_flex-wrap_justify-
content-between">
16             <p className="card-text_px-2"><strong>Nom:</strong>
17               <br> {message.nom}</p>
18             <p className="card-text_px-2"><strong>Email:</strong>
19               <br> {message.email}</p>
20             <p className="card-text_px-2"><strong>Age:</strong>
21               <br> {message.age}</p>
22             <p className="card-text_px-2"><strong>Genre:</strong>
23               <br> {message.gender}</p>
24             <p className="card-text_px-2"><strong>Date:</strong>
25               <br> {message.date}</p>
26           </div>
27
28           {/* // on affiche le commentaire */}
29           <div className="card-body">
30             <p className="card-text"><strong>Commentaire:</strong>
31               <br> {message.commentaire}</p>
32           </div>
33         </div>
34       ) : (
35
36         // si il n'y a pas de messages, on affiche un message d'
37         // information
38         <p>No messages to display</p>
39       )
40     )
41   </div>
42 </section>
```

Dans cette partie, nous avons utilisé la fonction **map()** pour afficher les commentaires.

La fonction map permet de parcourir un tableau et d'appliquer une fonction à chaque élément du tableau.

Nous avons utilisé la fonction **Array.isArray()** et la fonction **length** pour vérifier si le tableau est bien un tableau et s'il contient des éléments.

Sources

- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/map
- <https://getbootstrap.com/docs/5.1/components/card/>

Conclusion

Dans cet exercice, nous avons utilisé React, PHP et MySQL pour créer, en local, un formulaire de livre d'or.

Pour cela nous avons créé un composant **Formulaire.jsx** qui gère l'affichage du formulaire et des commentaires. Nous avons créé un serveur PHP pour traiter les données du formulaire et les stocker dans une base de données MySQL. Enfin, nous avons utilisé les hooks de React pour mettre à jour les données du formulaire et afficher les commentaires en temps réel.

Les codes source de cette partie sont disponibles dans l'annexe C

The screenshot shows a web browser window with the URL `localhost:3000/php`. The page title is "Livre d'or". The form consists of several input fields:

- nom :
- Message :
Entrez votre message ici :
- Email :
- Genre : Homme Femme Non-binaire
- Age :

Below the form is a blue button labeled "Submit Query". Underneath the form is a section titled "Messages" with the text "No messages to display".

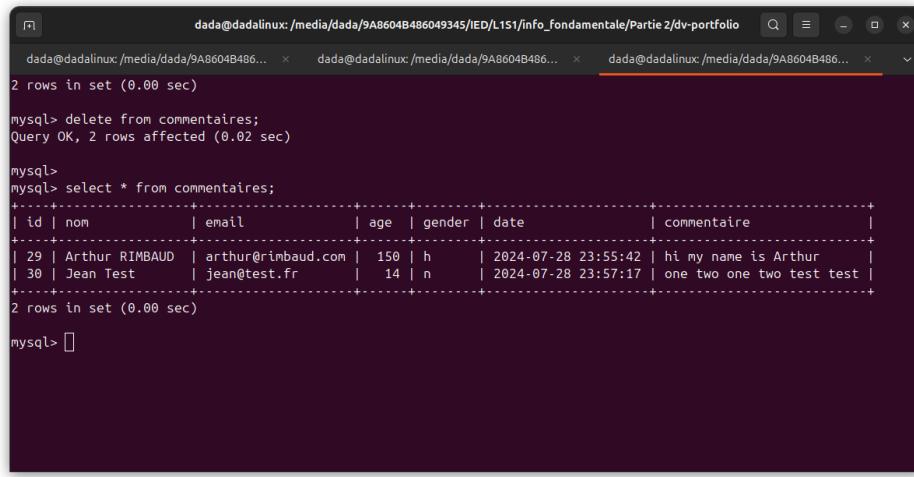
Formulaire de livre d'or avec React, PHP et MySQL

The screenshot shows a web browser window with the URL `localhost:3000/php`. At the top, there is a form with fields for Email (jean@test.fr), Gender (Non-binaire selected), Age (14), and a Submit Query button. Below the form, a section titled "Messages" displays two entries:

Nom:	Email:	Age:	Genre:	Date:
Jean Test	jean@test.fr	14	n	2024-07-28 23:57:17
Commentaire: one two one two test test				
Arthur RIMBAUD	arthur@rimbaud.com	150	h	2024-07-28 23:55:42
Commentaire: hi my name is Arthur				

At the bottom of the page, a dark footer bar contains the text "2024 - Damien Vallet".

Commentaires du livre d'or



```
dada@dadalinux:/media/dada/9A8604B486049345/IED/L1S1/info_fondamentale/Partie 2/dv-portfolio
dada@dadalinux:/media/dada/9A8604B486049345/IED/L1S1/info_fondamentale/Partie 2/dv-portfolio
dada@dadalinux:/media/dada/9A8604B486049345/IED/L1S1/info_fondamentale/Partie 2/dv-portfolio
2 rows in set (0.00 sec)

mysql> delete from commentaires;
Query OK, 2 rows affected (0.02 sec)

mysql>
mysql> select * from commentaires;
+----+-----+-----+-----+-----+
| id | nom      | email        | age | gender | date           |
+----+-----+-----+-----+-----+
| 29 | Arthur RIMBAUD | arthur@rimbaud.com | 150 | h     | 2024-07-28 23:55:42 |
| 30 | Jean Test   | jean@test.fr   | 14  | n     | 2024-07-28 23:57:17 |
+----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> 
```

Commentaires dans la base de données

Sources

- <https://youtu.be/rhPIw9a1CxQ?feature=shared>
- <https://grafikart.fr/tutoriels/serveur-web-interne-778>
- https://github.com/zain-ali-syed/react_php_guestbook/tree/master
- <https://www.dhiwise.com/post/the-synergy-between-php-and-react-in-modern-web-applications>

Exercice 5.2 (Bonus)

- Réalisez le même formulaire, traité avec le langage de votre choix, et cette fois à l'aide d'une base NoSQL.
- N'oubliez pas d'expliquer et d'illustrer comme dans l'exercice

Exercice 5.2 - Réalisation du Formulaire avec NoSQL

Pour réaliser cet exercice, nous avons utilisé MongoDB comme base de données NoSQL. Nous avons choisi d'installer **MongoDB** dans un container Docker.

Installation de MongoDB dans un container Docker

Nous avons d'abord exécuté la commande pull de docker pour récupérer l'image officielle de MongoDB.

```
1 docker pull mongo
```

Ensuite, nous avons exécuté la commande suivante pour créer un container appelé **dv-mongo** avec MongoDB.

```
1 docker run -d -p 27017:27017 --name dv-mongo mongo
```

L'option **-d** permet de lancer le container en arrière-plan et l'option **-p** permet de mapper le port 27017 du container au port 27017 de la machine hôte.

Le paramètre **--name** permet de donner un nom au container.

```
dada@dadalinux: /media/dada/9A8604B486049345/IED/L1S1/info_fondamentale/Partie 2/
mongo-srv$ docker pull mongo
Using default tag: latest
latest: Pulling from library/mongo
3713021b0277: Pull complete
39bdccaccd97: Pull complete
d6b691142508: Pull complete
bcc1924dee6d: Pull complete
091a7990873d: Pull complete
77e5254f6ae8: Pull complete
403f753f5920: Pull complete
88cd53ea307c: Pull complete
Digest: sha256:54996a559c724c726a31fb8131e1c9088a05f7e531760e2897212389bbf20fed
Status: Downloaded newer image for mongo:latest
docker.io/library/mongo:latest

What's next:
  View a summary of image vulnerabilities and recommendations → docker scout q
uickview mongo
dada@dadalinux: /media/dada/9A8604B486049345/IED/L1S1/info_fondamentale/Partie 2/
mongo-srv$ docker run -d -p 27017:27017 --name dv-mongo mongo
66063764b5f8920cc8eeffedea4b3a01e1b41ea68877f6e14558f236a9d6ca0c3
dada@dadalinux: /media/dada/9A8604B486049345/IED/L1S1/info_fondamentale/Partie 2/
mongo-srv$ 
```

Commande pull et création du container MongoDB

On vérifie que le container est bien lancé avec la commande suivante.

```
1 docker ps
```

On peut tester la connexion à la base de données avec la commande du shell MongoDB.

```
1 mongosh --host localhost --port 27017
```

```
dada@dadalinus: /media/dada/9A8604B486049345/IED/L1S1/info_fondamentale/Partie 2/mongo-srv
dada@dadalinus: /media/dada/9A8604B486049345/IED/L1S1/info_fondamentale/Partie 2/mongo-srv$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
66063764b5f8 mongo "docker-entrypoint.s..." About a minute ago Up About a minute 0.0.0.0:27017->27017/tcp dv-mongo
```

Container MongoDB lancé

L'option **-host** permet de spécifier l'hôte et **-port** permet de spécifier le port de connexion.

```
mongosh mongodb://localhost:27017/?direct... dada@dadalinus:
Try: sudo apt install <deb name>
dada@dadalinus: /media/dada/9A8604B486049345/IED/L1S1/info_fondamentale/Partie 2/mongo-srv$ mongosh --host localhost --port 27017
Current Mongosh Log ID: 66aa229869a1c81025838725
Connecting to: mongodb://localhost:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.2.15
Using MongoDB: 7.0.12
Using Mongosh: 2.2.15

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

-----
The server generated these startup warnings when booting
2024-07-31T11:35:51.220+00:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
2024-07-31T11:35:51.678+00:00: Access control is not enabled for the database
. Read and write access to data and configuration is unrestricted
2024-07-31T11:35:51.678+00:00: /sys/kernel/mm/transparent_hugepage/enabled is 'always'. We suggest setting it to 'never' in this binary version
2024-07-31T11:35:51.678+00:00: vm.max_map_count is too low
-----
test> [ ]
```

Connexion à la base de données MongoDB

Dans cette partie, nous avons installé MongoDB dans un container Docker. Tout d'abord, nous avons utilisé la commande **docker pull** pour récupérer l'image officielle de MongoDB. Ensuite, nous avons utilisé la commande **docker run** pour créer un container MongoDB. Enfin, nous avons testé la connexion à la base de données avec la commande **mongosh**.

Sources

- https://hub.docker.com/_/mongo
- <https://docs.docker.com/engine/reference/commandline/pull/>
- <https://docs.docker.com/engine/reference/commandline/run/>
- <https://www.mongodb.com/docs/mongodb-shell/connect/>

Installation de MongoDB

Pour installer MongoDB et ses dépendances, nous avons suivi les instructions du site officiel de MongoDB.

1. Ajouter la clé GPG de MongoDB

```
1 curl -fsSL https://www.mongodb.org/static/pgp/server-8.0.asc | \
2   \
3     sudo gpg -o /usr/share/keyrings/mongodb-server-8.0.gpg \
--dearmor
```

2. Créer un fichier de liste de sources pour MongoDB 8.0

```
1 echo "deb [arch=amd64,arm64 signed-by=/usr/share/keyrings/
  mongodb-server-8.0.gpg] https://repo.mongodb.org/apt/
  ubuntu/noble/mongodb-org/8.0 multiverse" | sudo tee /etc/
  apt/sources.list.d/mongodb-org-8.0.list
```

3. Recharger les paquets locaux

```
1 sudo apt-get update
```

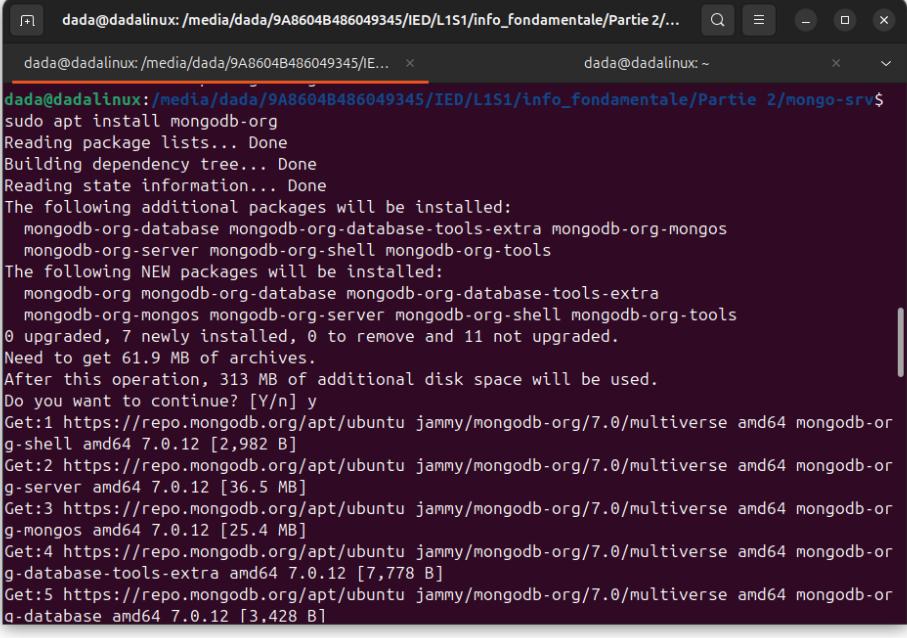
4. Installer le paquet mongodb-org

```
1 sudo apt-get install -y mongodb-org
```

```
dada@dadalinux:~$ sudo apt-get update
[sudo] password for dada:
Hit:1 http://fr.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:3 http://fr.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:4 https://download.docker.com/linux/ubuntu noble InRelease
Hit:5 http://fr.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:6 https://packages.microsoft.com/repos/code stable InRelease
Hit:7 https://distrib-coffee.ipsl.jussieu.fr/pub/mirrors/ctan/systems/win32/miktex/setup/noble InRelease
Reading package lists... Done
dada@dadalinux:~$ sudo apt-get install gnupg curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
gnupg is already the newest version (2.4.4-2ubuntu17).
gnupg set to manually installed.
curl is already the newest version (8.5.0-2ubuntu10.1).
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
dada@dadalinux:~$ curl -fsSL https://www.mongodb.org/static/pgp/server-8.0.asc | \
  sudo gpg -o /usr/share/keyrings/mongodb-server-8.0.gpg \
--dearmor
dada@dadalinux:~$ echo "deb [ arch=amd64,arm64 signed-by=/usr/share/keyrings/mongodb-server-8.0.gpg ] https://repo.mongodb.org/apt/ubuntu jammy/mongodb-org/8.0 multiverse" | \
  sudo tee /etc/apt/sources.list.d/mongodb-org-8.0.list
deb [ arch=amd64,arm64 signed-by=/usr/share/keyrings/mongodb-server-8.0.gpg ] https://repo.mongodb.org/apt/ubuntu jammy/mongodb-org/8.0 multiverse
dada@dadalinux:~$
```

Installation de MongoDB

Dans cette partie, nous avons installé MongoDB sur notre machine Ubuntu. Tout d'abord, nous avons ajouté la clé GPG de MongoDB avec la commande **curl** et **gpg**. Ensuite, nous avons créé un fichier de liste de sources pour MongoDB 8.0 avec la commande **echo**. Enfin, nous avons rechargeé les paquets locaux avec la commande **apt-get update** et installé le paquet **mongodb-org** avec la commande **apt-get install**.



The screenshot shows a terminal window with two tabs. The active tab displays the command `sudo apt install mongodb-org` and its execution. The output shows the package being downloaded and installed, including dependencies like `mongodb-org-database`, `mongodb-org-tools`, and `mongodb-org-shell`. It also lists `mongos` and `mongod` as newly installed packages. The total download size is approximately 61.9 MB.

```
dada@dadalinux:/media/dada/9A8604B486049345/IED/L1S1/info_fondamentale/Partie 2/...$ sudo apt install mongodb-org
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  mongodb-org-database mongodb-org-database-tools-extra mongodb-org-mongos
  mongodb-org-server mongodb-org-shell mongodb-org-tools
The following NEW packages will be installed:
  mongodb-org mongodb-org-database mongodb-org-database-tools-extra
  mongodb-org-mongos mongodb-org-server mongodb-org-shell mongodb-org-tools
0 upgraded, 7 newly installed, 0 to remove and 11 not upgraded.
Need to get 61.9 MB of archives.
After this operation, 313 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 https://repo.mongodb.org/apt/ubuntu jammy/mongodb-org/7.0/multiverse amd64 mongodb-org-shell amd64 7.0.12 [2,982 B]
Get:2 https://repo.mongodb.org/apt/ubuntu jammy/mongodb-org/7.0/multiverse amd64 mongodb-org-server amd64 7.0.12 [36.5 MB]
Get:3 https://repo.mongodb.org/apt/ubuntu jammy/mongodb-org/7.0/multiverse amd64 mongodb-org-mongos amd64 7.0.12 [25.4 MB]
Get:4 https://repo.mongodb.org/apt/ubuntu jammy/mongodb-org/7.0/multiverse amd64 mongodb-org-database-tools-extra amd64 7.0.12 [7,778 B]
Get:5 https://repo.mongodb.org/apt/ubuntu jammy/mongodb-org/7.0/multiverse amd64 mongodb-org-database amd64 7.0.12 [3,428 B]
```

Installation de MongoDB

Sources

- <https://www.mongodb.com/docs/manual/tutorial/install-mongodb-on-ubuntu/>
- <https://thelinuxforum.com/articles/912-how-to-install-mongodb-on-ubuntu-24-04>

Création d'une table (collection) dans MongoDB

Dans MongoDB, les données sont stockées dans des collections. Nous avons créé une collection **comments** pour stocker les commentaires du livre d'or.

Nous avons créé une base de données **base_livre_or** et une collection **commentaires** dans MongoDB. Pour accéder à notre base de données MongoDB qui tourne dans un container Docker, nous avons utilisé le shell MongoDB.

```
1 mongosh --host localhost --port 27017
```

MongoDB ne possède pas de commande CREATE pour créer une base de données. On utilise la commande **use** pour créer une base de données si elle n'existe pas.
nous avons utilisé la commande **use** pour créer une base de données **base_livre_or**.

```
1 use base_livre_or
```

```
mongosh mongodb://localhost:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.2.15
dada@dadalinux:/media/dada/9A86048486049345/IED/L1S1/info_fondamentale/Partie 2/mongo-srv$ mongosh --host localhost --port 27017
Current Mongosh Log ID: 66aa3dc6155ef0d562838725
Connecting to:      mongodb://localhost:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.2.15
Using MongoDB:      7.0.12
Using Mongosh:      2.2.15

For mongosh info see: https://docs.mongodb.com/mongodb-shell/
-----
The server generated these startup warnings when booting
2024-07-31T13:35:53.800+00:00: Using the XFS filesystem is strongly recommended with the
WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
2024-07-31T13:35:54.498+00:00: Access control is not enabled for the database. Read and
write access to data and configuration is unrestricted
2024-07-31T13:35:54.498+00:00: /sys/kernel/mm/transparent_hugepage/enabled is 'always'.
We suggest setting it to 'never' in this binary version
2024-07-31T13:35:54.498+00:00: vm.max_map_count is too low
-----

test> show dbs
admin   40.00 KiB
config  72.00 KiB
local   72.00 KiB
test> use base_livre_or
switched to db base_livre_or
base_livre_or>
```

Création de la base de données **base_livre_or**

MongoDB crée la base de données lorsqu'on insère des données dans une collection. Pour créer une collection, nous avons inséré des données dans la collection **commentaires**.

```
1 db.createCollection("commentaires")
```

On affiche les collections de la base de données avec la commande **show collections**.

```
1 show collections
```

```

mongosh mongodb://localhost:27017/?directConnection=true&serverSelectionTimeoutM...   Search   Minimize   Maximize   Close
mongosh mongodb://localhost:27017/?directConnecti...   dada@dadalinux: ~
Using Mongosh:          2.2.15
For mongosh info see: https://docs.mongodb.com/mongodb-shell/
-----
The server generated these startup warnings when booting
2024-07-31T13:35:53.800+00:00: Using the XFS filesystem is strongly recommended with the
WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
2024-07-31T13:35:54.498+00:00: Access control is not enabled for the database. Read and
write access to data and configuration is unrestricted
2024-07-31T13:35:54.498+00:00: /sys/kernel/mm/transparent_hugepage/enabled is 'always'.
We suggest setting it to 'never' in this binary version
2024-07-31T13:35:54.498+00:00: vm.max_map_count is too low
-----
test> show dbs
admin   40.00 KiB
config  72.00 KiB
local   72.00 KiB
test> use base_livre_or
switched to db base_livre_or
base_livre_or>
(To exit, press Ctrl+C again or Ctrl+D or type .exit)
base_livre_or> db.createCollection("commentaires")
{ ok: 1 }
base_livre_or> show collections
commentaires
base_livre_or> []

```

Création de la collection commentaires

Sources

- <https://www.mongodb.com/resources/products/fundamentals/create-database>
- <https://docs.mongodb.com/manual/reference/method/db.createCollection/>
- <https://docs.mongodb.com/manual/reference/method/db.collection.find/>

```
dada@dadalinux:/media/dada/9A8604B486049345/IED/L1S1/info_fondamentale/mongo-srv$ npm init -y
Wrote to /media/dada/9A8604B486049345/IED/L1S1/info_fondamentale/mongo-srv/package.json:

{
  "name": "mongo-srv",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": ""
}

dada@dadalinux:/media/dada/9A8604B486049345/IED/L1S1/info_fondamentale/mongo-srv$
```

Initialisation du projet Node.js

Création de notre serveur Node.js avec Express

Nous allons créer un serveur Node.js avec Express pour traiter les données du formulaire et les stocker dans la base de données MongoDB.

Pour cela nous avons créé un fichier **index.js** dans un dossier **mongo-srv**.

Dans ce dossier, nous avons installé les dépendances suivantes :

```
1 // on initialise le projet Node.js
2 npm init -y
3
4 // on installe les dépendances
5 npm install express mongoose body-parser cors
```

La commande `npm init -y` permet d'initialiser un projet Node.js avec les paramètres par défaut. La dépendance `express` est un framework Node.js pour créer des applications web. Il va nous permettre de créer notre serveur.

`mongoose` est une bibliothèque Node.js qui nous permet de gérer les données qui nous envoyons à la base MongoDB sous forme d'objets appelés Schémas.

`body-parser` est un middleware qui permet de récupérer les données envoyées par le formulaire, dans une application Node.js.

`cors` est un middleware qui permet de gérer les requêtes HTTP entre différents domaines dans une application Node.js.

Un middleware est une fonction qui a accès à l'objet de requête (`req`), à l'objet de réponse (`res`).

nous avons créé un script dans le fichier **package.json**.

Sources

- <http://expressjs.com/>
- <https://mongoosejs.com/>
- <https://expressjs.com/en/resources/middleware/cors.html>
- <http://expressjs.com/en/resources/middleware/body-parser.html>
- https://fr.wikipedia.org/wiki/Cross-origin_resource_sharing
- <https://fr.wikipedia.org/wiki/Middleware>

Création du fichier index.js

Une fois notre projet initialisé, nous avons créé un fichier **index.js** pour notre serveur Node.js. Ce fichier sera notre point d'entrée pour notre serveur.

Voici le code de ce fichier :

```

1 /**
2 * File Name: index.js
3 * Description: Configuration du serveur Express et connexion à MongoDB
4 * Author: Damien Vallet
5 * Version: 1.0
6 * Date: 2024
7 * License: Created for UOR course
8 * Language: JavaScript
9 */
10
11 // Importation des librairies
12 const express = require('express')
13 const mongoose = require('mongoose')
14 const bodyParser = require('body-parser')
15 const cors = require('cors')
16
17 // Importation des routes
18 const commentaires = require('./Routes/commentaires')
19
20 // Initialisation de l'application et du port
21 const app = express()
22 const PORT = process.env.PORT || 5000
23
24 // configuration cors et bodyParser avec la méthode use
25 app.use(cors())
26 app.use(bodyParser.json())
27
28 // Connexion à la base de données
29 mongoose.connect('mongodb://localhost:27017/base_livre_or', {
30
31     // Si la connexion à la base de données est réussie, un message s'affiche dans la console
32 }).then(() => console.log('Connected to MongoDB'))
33
34     // Sinon un message d'erreur s'affiche dans la console
35     .catch(err => console.error(err));
36
37 // Définition de la route pour les commentaires
38 app.use('/commentaires', commentaires)
39
40 // Lancement du serveur
41 app.listen(PORT, () => {
42     console.log(`Server is running on port ${PORT}`)
43 });

```

Dans ce fichier, nous avons créé un serveur Node.js avec Express.

- Nous avons importé les librairies express, mongoose, body-parser et cors.
- Nous avons importé la route vers notre model de commentaire.
- Nous avons initialisé notre application et défini le port.
- Nous avons connecté notre serveur à la base de données MongoDB.
- Nous avons défini la route pour les commentaires.
- Nous avons lancé notre serveur avec la méthode listen.

Sources

- <https://expressjs.com/fr/4x/api.html#app.use>
- <https://expressjs.com/fr/guide/using-middleware.html>
- <https://expressjs.com/fr/4x/api.html#app.listen>

Création de la route pour les commentaires

Nous avons créé un dossier **Routes** dans notre projet pour stocker les routes de notre application. Les routes sont des fichiers qui contiennent les fonctions pour traiter les requêtes HTTP. Nous avons créé un fichier **commentaires.js** dans le dossier **Routes**. Voici le code de ce fichier :

```

49     });
50
51     // on sauvegarde le commentaire dans la base de données
52     const savedComment = await newComment.save();
53     res.json({success: true, message: 'Commentaire ajouté avec succès'});
54   } catch (err) {
55     res.status(500).json({message: err.message});
56   }
57 }
58 });
59
60 // Exportation du module
61 module.exports = router;

```

Dans ce fichier, nous avons créé les routes pour les commentaires.

- Nous avons importé les librairies express et le modèle Commentaire.
- Nous avons créé une route pour récupérer tous les commentaires avec la méthode GET.
- Nous avons créé une route pour ajouter un nouveau commentaire avec la méthode POST.
- Nous avons vérifié si les champs du formulaire sont vides.
- Nous avons créé un nouveau commentaire et nous l'avons sauvégarde dans la base de données.

Sources

- <https://expressjs.com/en/guide/routing.html>
- <https://expressjs.com/fr/4x/api.html#router.route>
- <https://mongoosejs.com/docs/guide.html>
- <https://expressjs.com/fr/4x/api.html#app.get.method>
- <https://expressjs.com/fr/4x/api.html#app.post.method>
- <https://expressjs.com/fr/4x/api.html#res.status>
- <https://mongoosejs.com/docs/documents.html#updating-using-save>

Création du modèle Commentaire

Nous avons créé un modèle **commentaire** pour définir la structure des données de notre collection MongoDB.

Nous avons créé un fichier **commentaire.js** dans un dossier **Models**.

Voici le code de ce fichier :

```

1 /**
2 * File Name: commentaire.js
3 * Description: Modèle pour les commentaires
4 * Author: Damien Vallet
5 * Version: 1.0
6 * Date: 2024
7 * License: Created for UOR course
8 * Language: JavaScript
9 */
10
11
12 // Initialisation de la base de données
13 const mongoose = require('mongoose');
14
15 // Création du schéma
16 const CommentaireSchema = new mongoose.Schema({
17     // Définition du champ nom
18     nom: {
19         type: String,
20         required: true
21     },
22
23     // Définition du champ email
24     email: {
25         type: String,
26         required: true
27     },
28
29     // Définition du champ date
30     date: {
31         type: Date,
32         required: true
33     },
34
35     // Définition du champ age
36     age: {
37         type: Number,
38         required: true
39     },
40
41     // Définition du champ genre
42     genre: {
43         type: String,
44         required: true
45     },
46
47     // Définition du champ commentaire
48     commentaire: {
49         type: String,

```

```
50     required: true
51   }
52 });
53
54 // Export du modèle
55 module.exports = mongoose.model('commentaire', CommentaireSchema);
```

Dans ce fichier, nous avons créé un modèle pour les commentaires.

- Nous avons initialisé la base de données avec mongoose.
- Nous avons créé un schéma pour les commentaires.
- Nous avons défini les champs nom, email, date, age, genre et commentaire.
- Nous avons créé et exporté le modèle.

Sources

- <https://mongoosejs.com/docs/schematypes.html>
- <https://mongoosejs.com/docs/api/model.html#Model>
- <https://www.tutorialsteacher.com/nodejs/nodejs-module-exports>

Lancement du serveur Node.js

Pour lancer notre serveur, on exécute la commande suivante depuis le dossier **mongo-srv**.

```
1 node index.js
```

Dans cette partie, nous avons lancé notre serveur Node.js.
Nous avons utilisé le shell MongoDB pour nous connecter à la base de données et vérifier que les données sont bien stockées.



A screenshot of a terminal window titled "dada@dadalinux: /media/dada/9A8604B486049345/IED/L1S1/info_fondam...". The terminal shows the command "node index.js" being run, followed by output indicating the server is running on port 5000 and connected to MongoDB. A mongo shell session is also visible in the background.

```
dada@dadalinux: /media/dada/9A8604B486049345/IED/L1S1/info_fondam...
dada@dadalinux: /media/dada/9A8604B486049345/IED/L1S1/info_fondamentale/Partie 2/
mongo-srv$ node index.js
Server is running on port 5000
Connected to MongoDB
Mongo je suis un test mongo 24 test@test.com n
```

Lancement du serveur Node.js

Modification du formulaire

Nous devons maintenant modifié notre formulaire pour envoyer les données au serveur Node.js. Nous avons dupliqué et renommé le fichier **Formulaire.jsx** en **FormulaireMongo.jsx**. On a également dupliqué le fichier **VersionPHP** et nous l'avons renommé en **VersionMongo**. Nous avons modifié les lignes suivantes pour utilisé le composant **FormulaireMongo** :

```

1 @l26
2 import FormulaireMongo from './FormulaireMongo.jsx';
3
4 ...
5
6 @l235
7 <section id="livredor">
8     <FormulaireMongo />
9 ...

```

Voici les modifications apportées au fichier **FormulaireMongo.jsx** :

```

1 // on crée une fonction pour récupérer les messages
2 const recupererMessages = async () => {
3     // on récupère les messages du serveur MongoDB
4     const reponse = await fetch('http://localhost:5000/commentaires')
5
6     // on récupère les données de la réponse
7     const data = await reponse.json();
8
9     // on met à jour le tableau de messages
10    setMessages(data);
11}
12
13 ...
14
15
16 // on crée une fonction pour ajouter un message
17 const ajouteMessage = async (message) => {
18     // on envoie les données du formulaire au serveur MongoDB
19     const reponse = await fetch('http://localhost:5000/commentaires', {
20         method: 'POST',
21         headers: {
22             'Content-Type': 'application/json'
23         },
24         body: JSON.stringify(message),
25     });

```

On crée une nouvelle route en modifiant ainsi le fichier **NavigationHeader.jsx** :

```

1 /**
2 * File Name: NavigationHeader.js
3 * Description: Navigation entre versions
4 * Author: Damien Vallet
5 * Version: 1.0
6 * Date: 2024
7 * License: Created for UOR course
8 * Language: JavaScript
9 */
10
11 // Importation de la librairie React
12 import React from "react";
13
14 // Importation de la librairie React Router
15 import { BrowserRouter as Router, Route, Routes } from "react-router-dom";
16
17 // Importation des composants;
18 import NavigationHeaderBar from "./components/NavigationHeaderBar";
19 import VersioHTML from "./components/VersionHTML";
20 import VersionCSS from "./components/VersionCSS";
21 import VersionPHP from "./components/VersionPHP";
22 import VersionMongo from "./components/VersionMongo";
23
24 // Définition de la fonction NavigationHeader
25 function NavigationHeader() {
26     return (
27         //
28         <>
29             {/* composant de navigation */}
30             <Router>
31                 <div>
32
33                     {/* Barre de navigation */}
34                     <NavigationHeaderBar />
35
36                     {/* definition des routes */}
37                     <Routes>
38
39                         {/* route vers la version HTML */}
40                         <Route path="/html" element={<VersioHTML />} />
41
42                         {/* route vers la version CSS */}
43                         <Route path="/css" element={<VersionCSS />} />
44
45                         {/* route vers la version PHP */}
46                         <Route path="/php" element={<VersionPHP />} />
47
48                         {/* route vers la version MongoDB */}
49                         <Route path="/mongo" element={<VersionMongo />} />
50
51                     </Routes>
52                 </div>
53                 </Router>
54             </>

```

```
55     );
56 }
57
58 // Exportation du composant NavigationHeader
59 export default NavigationHeader;
```

On modifie le fichier **NavigationHeaderBar.jsx** pour ajouter un lien vers la nouvelle route :

```

1 /**
2 * File Name: NavigationHeaderBar.jsx
3 * Description: Composant qui gère la barre pour naviguer entre les
4 *               versions
5 * Author: Damien Vallet
6 * Version: 1.0
7 * Date: 2024
8 * License: Created for UOR course
9 * Language: JavaScript
10 */
11 // Importation de la librairie React
12 import React from "react";
13
14 // Importation de la librairie React Router
15 import { Link } from "react-router-dom";
16
17 // définition du composant NavBar
18 function NavigationHeaderBar() {
19     return (
20
21         // Barre de navigation
22         <nav id="header-navbar">
23
24             {/* Une liste non ordonnée (<ul>) définit les éléments de
25                 navigation. Chaque élément de navigation est un élément de
26                 liste (<li>)*/}
27             <ul>
28                 <li>
29                     {/* Lien vers la version HTML */}
30                     <Link to="/html">Version HTML</Link>
31                 </li>
32                 <li>
33                     {/* Lien vers la version avec CSS/Bootstrap */}
34                     <Link to="/css">Version CSS</Link>
35                 </li>
36                 <li>
37                     {/* lien vers la version PHP avec base SQL */}
38                     <Link to="/php">Version PHP</Link>
39                 </li>
40                 <li>
41                     {/* lien vers la version avec MongoDB */}
42                     <Link to="/mongo">Version MongoDB</Link>
43                 </li>
44             </ul>
45         </nav>
46     );
47 }
48 // Exportation du composant NavigationHeaderBar
49 export default NavigationHeaderBar;

```

Dans cette partie, nous avons dupliqué et renommé le fichier **Formulaire.jsx** en **FormulaireMongo.jsx**.

Nous avons créé un nouveau composant à partir du fichier **VersionPHP.jsx** et nous l'avons renommé en **VersionMongo.jsx**.

Nous avons également rajouté une nouvelle route dans **NavigationHeader** et **NavigationHeaderBar** pour accéder à notre nouveau composant.

Enfin, nous avons modifié les adresses du serveur fournie aux fonctions **recupereMessages()** et **ajouteMessage()**.

Affichage des commentaires

En conclusion, nous avons utilisé MongoDB pour stocker les commentaires du livre d'or.

Tout d'abord, nous avons créé un serveur Node.js pour traiter les données du formulaire et les stocker dans une base de données NoSQL avec MongoDB. Ensuite, nous avons modifié le formulaire pour envoyer les données au serveur Node.js et nous avons modifié le composant de navigation pour ajouter un lien vers la nouvelle route.

Enfin, nous avons lancé notre serveur Node.js et testé l'ajout et l'affichage des commentaires.

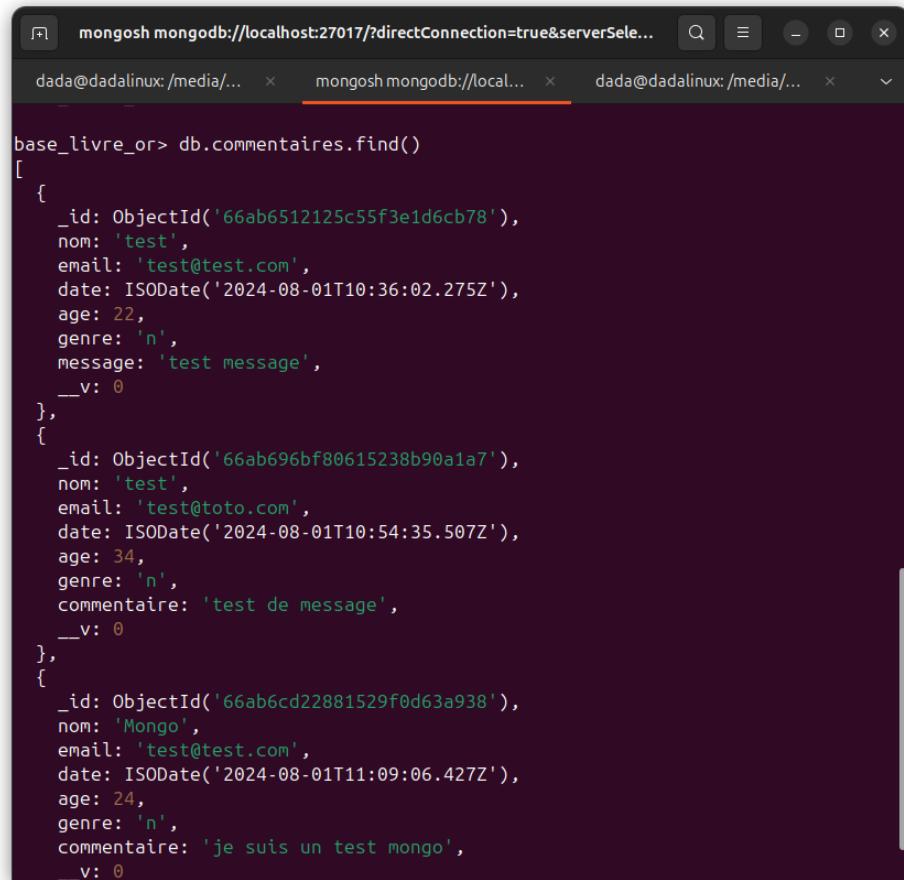
Les codes source de cette partie sont disponibles dans l'annexe D

The screenshot shows a web browser window with the URL `localhost:3000/php`. At the top, there is a search form with fields for 'Genre' (radio buttons for Homme, Femme, Non-binaire, with Non-binaire selected), 'Age' (text input set to 18), and a 'Submit Query' button. Below the form, the word 'Messages' is underlined in blue. A table-like structure displays three rows of comments:

Nom:	Email:	Age:	Genre:	Date:
Mongo	test@test.com	24	n	2024-08-01T11:09:06.427Z
Commentaire: je suis un test mongo				
test	toto@test.com	34	n	2024-08-01T10:54:35.507Z
Commentaire: test de message				
test	test@test.com	22	n	2024-08-01T10:36:02.275Z
Commentaire:				

At the bottom of the page, a dark footer bar contains the text '2024 - Damien Vallet'.

Commentaires du livre d'or avec MongoDB



A screenshot of a terminal window titled "mongosh mongodb://localhost:27017/?directConnection=true&serverSelec...". The window contains the following MongoDB query and its results:

```
base_livre_or> db.commentaires.find()
[
  {
    _id: ObjectId('66ab6512125c55f3e1d6cb78'),
    nom: 'test',
    email: 'test@test.com',
    date: ISODate('2024-08-01T10:36:02.275Z'),
    age: 22,
    genre: '\n',
    message: 'test message',
    __v: 0
  },
  {
    _id: ObjectId('66ab696bf80615238b90a1a7'),
    nom: 'test',
    email: 'test@toto.com',
    date: ISODate('2024-08-01T10:54:35.507Z'),
    age: 34,
    genre: '\n',
    commentaire: 'test de message',
    __v: 0
  },
  {
    _id: ObjectId('66ab6cd22881529f0d63a938'),
    nom: 'Mongo',
    email: 'test@test.com',
    date: ISODate('2024-08-01T11:09:06.427Z'),
    age: 24,
    genre: '\n',
    commentaire: 'je suis un test mongo',
    __v: 0
  }
]
```

Commentaires dans la base de données MongoDB

6

Chapitre 6 - Le Langage JavaScript

Exercice 6.1 (A Rendre)

- Réalisez une copie de votre page initiale (comportant les CSS) et ajoutez à celle-ci du JavaScript afin d'obtenir, en utilisant le DOM, une page qui réagit aux actions de l'utilisateur.
- Comme vous disposez maintenant de quatre pages distinctes sur votre serveur (en local ou chez votre hébergeur), ajoutez cette nouvelle page à votre menu ou à votre barre de navigation.
- N'oubliez pas d'expliquer les méthodes DOM/JS utilisées, de C.C le code Javascript dans son intégralité, de réaliser une (ou plu- sieurs selon) C.E(s) de votre page JS réactive affichée dans votre navigateur et d'ajouter votre ou vos sources à votre dossier Htdocs en archive attachée à votre envoi.

Exercice 6.1 (A Rendre) :

Pour réaliser cet exercice, nous avons ajouté une nouvelle page intitulée **VersionJS.jsx** à notre site web ainsi que nouvelle route pour accéder à cette page.

Modification du fichier NavigationHeader.jsx :

Nous avons modifié le fichier **NavigationHeader.js** pour ajouter un lien vers la nouvelle page. Voici le code ajusté du fichier **NavigationHeader.js** :

```
1 /**
2 * File Name: NavigationHeader.js
3 * Description: Navigation entre versions
4 * Author: Damien Vallet
5 * Version: 1.0
6 * Date: 2024
7 * License: Created for UOR course
8 * Language: JavaScript
9 */
10
11 // Importation de la librairie React
12 import React from "react";
13
14 // Importation de la librairie React Router
15 import { BrowserRouter as Router, Route, Routes } from "react-router-dom";
16
17 // Importation des composants;
18 import NavigationHeaderBar from "./components/NavigationHeaderBar";
19 import VersioHTML from "./components/VersionHTML";
20 import VersionCSS from "./components/VersionCSS";
21 import VersionPHP from "./components/VersionPHP";
22 import VersionMongo from "./components/VersionMongo";
23 import VersionJS from "./components/VersionJS";
24
25 // Définition de la fonction NavigationHeader
26 function NavigationHeader() {
27     return (
28         //
29         <>
30             {/* composant de navigation */}
31             <Router>
32                 <div>
33
34                     {/* Barre de navigation */}
35                     <NavigationHeaderBar />
36
37                     {/* definition des routes */}
38                     <Routes>
39
40                         {/* route vers la version HTML */}
41                         <Route path="/html" element={<VersioHTML />} />
42
43                         {/* route vers la version CSS */}
44                         <Route path="/css" element={<VersionCSS />} />
45
46                         {/* route vers la version PHP */}
47                     </Routes>
48                 </div>
49             </Router>
50         </>
51     )
52 }
53
54 // Exportation de la fonction
55 export default NavigationHeader;
```

```

47         <Route path="/php" element={<VersionPHP />} />
48
49         {/* route vers la version MongoDB */}
50         <Route path="/mongo" element={<VersionMongo />} />
51
52         {/* route vers la version JavaScript */}
53         <Route path="/js" element={<VersionJS />} />
54
55     </Routes>
56 </div>
57 </Router>
58 </>
59 );
60 }
61
62 // Exportation du composant NavigationHeader
63 export default NavigationHeader;

```

Modification du fichier NavigationHeaderBar.jsx :

On modifie également le fichier `NavigationHeaderBar.js` pour ajouter un lien vers la nouvelle page.
Voici le code du fichier `NavigationHeaderBar.js` :

```

1 /**
2 * File Name: NavigationHeaderBar.jsx
3 * Description: Composant qui gère la barre pour naviguer entre les
4 *               versions
5 * Author: Damien Vallet
6 * Version: 1.0
7 * Date: 2024
8 * License: Created for UOR course
9 * Language: JavaScript
10 */
11
12 // Importation de la librairie React
13 import React from "react";
14
15 // Importation de la librairie React Router
16 import { Link } from "react-router-dom";
17
18 // définition du composant NavBar
19 function NavigationHeaderBar() {
20     return (
21
22         // Barre de navigation
23         <nav id="header-navbar">
24             {/* Une liste non ordonnée (<ul>) définit les éléments de
25                 navigation. Chaque élément de navigation est un élément de
26                 liste (<li>)*/}
27             <ul>
28                 <li>
29                     {/* Lien vers la version HTML */}
30                     <Link to="/html">Version HTML</Link>
31                 </li>
32                 <li>
33                     {/* Lien vers la version avec CSS/Bootstrap */}
34                 </li>
35             </ul>
36         </nav>
37     );
38 }
39
40 // Exportation du composant NavigationHeader
41 export default NavigationHeaderBar;

```

```

31         <Link to="/css">Version CSS</Link>
32     </li>
33     <li>
34         {/* lien vers la version PHP avec base SQL */}
35         <Link to="/php">Version PHP</Link>
36     </li>
37     <li>
38         {/* lien vers la version avec MongoDB */}
39         <Link to="/mongo">Version MongoDB</Link>
40     </li>
41     <li>
42         {/* Lien vers la version du site contenant des animations en
43             JavaScript */}
44         <Link to="/js">Version JavaScript</Link>
45     </li>
46 </ul>
47 </nav>
48 );
49
50 // Exportation du composant NavigationHeaderBar
51 export default NavigationHeaderBar;

```

Création du fichier commonJS.js :

Ensuite, nous avons ajouté du JavaScript à la page **VersionJS.jsx** pour obtenir une page qui réagit aux actions de l'utilisateur.

Nous avons choisi de rajouter un bouton qui change la couleur de fond de la page.

Cette fonction est codée dans le fichier **commonJS.js**.

```

1 /**
2 * File Name: commonJS.js
3 * Description: fichier pour stocker nos fonctions communes
4 * Author: Damien Vallet
5 * Version: 1.0
6 * Date: 2024
7 * License: Created for UOR course
8 * Language: JavaScript
9 */
10
11 // fonction pour modifier la couleur du fond de la page
12 function changeBgColor(color) {
13     document.body.style.backgroundColor = color;
14 }
15
16 // fonction pour modifier la couleur du texte de la page
17 function changeTextColor(color) {
18     document.body.style.color = color;
19 }
20
21 // exportation des fonctions
22 export { changeBgColor, changeTextColor};

```

Dans cette fonction, nous avons utilisé la propriété **style** de l'objet **document.body** pour changer la couleur de fond de la page.

Le document est l'objet principal de notre page web et la propriété body cible le corps de la page.

La propriété style permet de modifier le CSS de l'élément ciblé.

Création du fichier VersionJS.jsx :

On importe ensuite ces fonctions dans le fichier **VersionJS.jsx**. On crée une fonction dans notre composant qui appelle nos fonctions **changeBgColor** et **changeTextColor** pour changer la couleur de fond et de texte de la page.

On a rajouté une condition avec l'opérateur ternaire pour que le texte soit blanc sur un fond noir et noir sur un fond blanc. L'opérateur ternaire est une forme condensée de la structure conditionnelle **if...else**.

Nous avons également ajouté deux boutons dans la barre de navigation pour changer la couleur de fond ainsi que la couleur du texte de la page. Pour cela nous avons utilisé la balise **<button>** et nous avons appelé la fonction **changeColor** avec les paramètres "black" et "white" dans les attributs **onClick**.

Voici le code du fichier **VersionJS.jsx** :

```

1 /**
2 * File Name: VersionJS.jsx
3 * Description: Page principale de la version JavaScript
4 * Author: Damien Vallet
5 * Version: 1.0
6 * Date: 2024
7 * License: Created for UOR course
8 * Language: JavaScript
9 */
10
11 // Importation de la librairie React
12 import React, { useEffect } from "react";
13
14 // Importation des images
15 import picofme from "../img/picofme.jpg";
16
17 // Importation des icônes
18 import { ReactComponent as LinkedinIcon } from "../img/linkedin.svg";
19 import { ReactComponent as GithubIcon } from "../img/github.svg";
20 import { ReactComponent as MastodonIcon } from "../img/mastodon.svg";
21 import { ReactComponent as Soundwave } from '../img/soundwave.svg';
22
23 // Importation du composant vidéo
24 import videopres from "../img/UOR-ch2-video.mov";
25 import Formulaire from "./Formulaire.jsx";
26
27 // Importation des fonctions communes
28 import { changeBgColor, changeTextColor, animateImage } from "../commonJS.js";
29
30 // définition du composant versionJS
31 function VersionJS() {
32
33     // utilisation du hook useEffect
34     useEffect(() => {
35         [], []);
36
37     // fonction pour changer la couleur de la page
38     const changeColor = (color) => {
39         changeBgColor(color);
40         changeTextColor(color === "white" ? "black" : "white");

```

```
41
42
43 // fonction pour animer une image
44 const changeImageSize = (event) => {
45   const image = event.target;
46   animateImage(image);
47 }
48
49 return (
50
51   // retourne le code HTML
52   <div id="main">
53
54     {/* ouverture de la balise header */}
55     <header id="header" className="p-3 bg-dark text-light text-center">
56       >
57
58     {/* création d'un container */}
59     <div className="container-fluid">
60
61       {/* création d'une ligne */}
62       <div className="row">
63
64         {/* création d'une colonne */}
65         <div className="col-12 col">
66
67           {/* titre de la page */}
68           <h1 className="display-1 hero-title">Damien Vallet</h1>
69           <h2 className="display-2">Portofolio</h2>
70           <h3>Audio & Product Design / Programmation</h3>
71         </div>
72       </div>
73     </div>
74     <hr />
75
76     {/* navbar */}
77     <nav id="navbar" className="navbar bg-dark text-light">
78       <div className="container">
79         <div className="d-flex flex-wrap align-items-center justify-content-center justify-content-lg-start">
80           <Soundwave className="px-2" style={{ fill: 'white', height: '40px', width: '40px' }} />
81           <ul className="nav col-12 col-lg-auto me-lg-auto mb-2 justify-content-center mb-md-0">
82             <li>
83               <a className="nav-link px-2 text-light" href="#main">
84                 Home
85               </a>
86             </li>
87             <li>
88               <a className="nav-link px-2 text-light" href="#whoami">
89                 Présentation
90               </a>
91             </li>
92             <li>
```

```

92             <a className="nav-link_px-2_text-light" href="#projets">
93                 " >
94                 Projets
95             </a>
96         </li>
97         <li>
98             <a className="nav-link_px-2_text-light" href="#contact">
99                 " >
100                Contacts
101            </a>
102        </li>
103        <li>
104            <a className="nav-link_px-2_text-light" href="#">
105                livredor" />
106        </li>
107        /* boutons pour changer la couleur de la page */
108        <li>
109            <button className="btn_mx-2_btn-outline-light" onClick={() => changeColor("black")}>
110                Dark
111            </button>
112        </li>
113        <li>
114            <button className="btn_mx-2_btn-outline-light" onClick={() => changeColor("white")}>
115                Light
116            </button>
117        </li>
118    </ul>
119 </div>
120 </div>
121 </header>
122 . . .

```

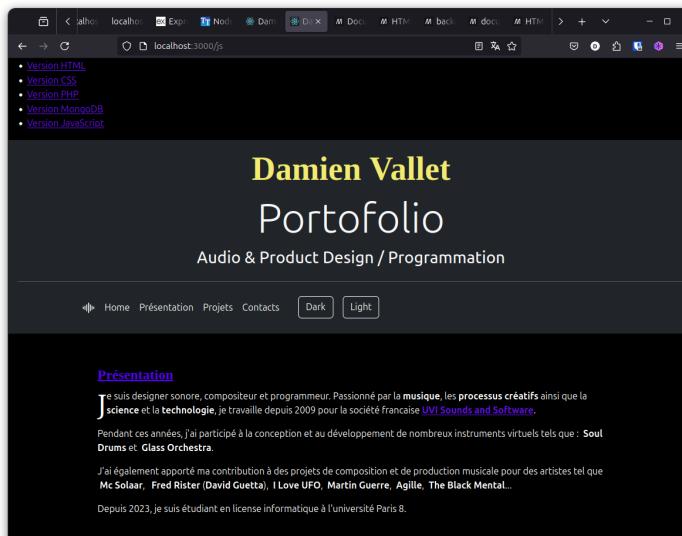
Dans cette partie, nous avons ajouté deux fonctions communes pour modifier la couleur de fond et de texte de la page.

Nous avons ajouté une fonction appelée **changeColor** qui appelle ces deux fonctions pour changer la couleur de fond et de texte de la page.

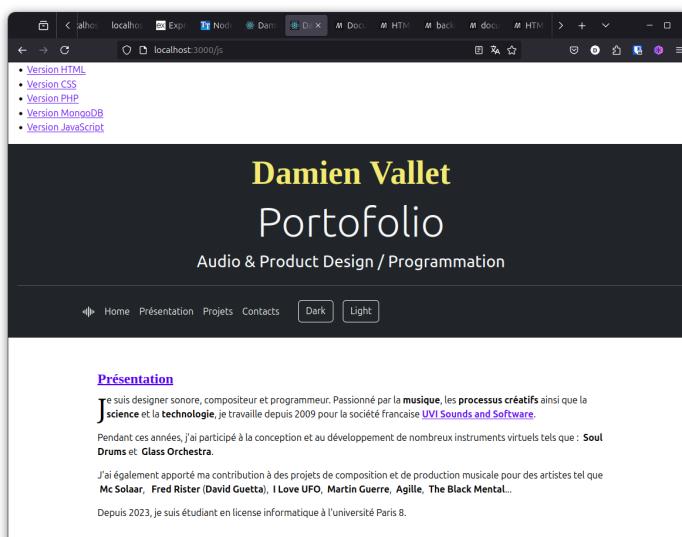
Nous avons enfin ajouté deux boutons dans la barre de navigation afin de contrôler notre fonction et ainsi créer un pseudo mode sombre.

Sources

- <https://developer.mozilla.org/fr/docs/Web/API/Document/body>
- <https://developer.mozilla.org/fr/docs/Web/API/HTMLElement/style>
- <https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Statements/export>



Capture d'écran de la page VersionJS en mode sombre



Capture d'écran de la page VersionJS en mode clair

Ajout d'une fonction d'animation :

En guise de second exemple, nous avons crée une fonction qui modifie notre image de contact lorsqu'elle est survolée par la souris.

Pour cela, nous avons ajouté une fonction **animateImage** dans le fichier **commonJS.js** qui modifie la taille de l'image lorsqu'elle est survolée.

Voici le code du fichier **commonJS.js** :

```

1 /**
2 * File Name: commonJS.js
3 * Description: fichier pour stocker nos fonctions communes
4 * Author: Damien Vallet
5 * Version: 1.0
6 * Date: 2024
7 * License: Created for UOR course
8 * Language: JavaScript
9 */
10
11 // fonction pour modifier la couleur du fond de la page
12 function changeBgColor(color) {
13     document.body.style.backgroundColor = color;
14 }
15
16 // fonction pour modifier la couleur du texte de la page
17 function changeTextColor(color) {
18     document.body.style.color = color;
19 }
20
21 // fonction pour animer une image en la faisant grossir quand on passe la
22 // souris dessus et retrécir quand on la quitte
23 function animateImage(element) {
24     // on fait une transition pour que l'animation soit plus douce
25     element.style.transition = "transform 0.5s ease"; // Ensure smooth
26     // transition
27
28     // on agrandit l'image quand on passe la souris dessus
29     element.addEventListener('mouseover', function () {
30         this.style.transform = "scale(1.2)"; // Grow the element by 20%
31     });
32
33     // on rétrécit l'image quand on quitte l'image
34     element.addEventListener('mouseout', function () {
35         this.style.transform = "scale(1)"; // Return to original size
36     });
37
38 // exportation des fonctions
39 export { changeBgColor, changeTextColor, animateImage };

```

Dans la fonction **animateImage**, nous avons utilisé la propriété **transition** pour ajouter un effet de transition à notre animation.

Nous avons ensuite créé un **addEventListener** pour écouter les événements **mouseover** et **mouseout** sur l'élément passé en paramètre.

Dans l'événement **mouseover**, nous passons une fonction qui modifie la taille de l'image en utilisant la propriété **transform** avec la valeur **scale(1.2)** pour agrandir l'image de 20%.

Dans l'événement **mouseout**, nous passons une fonction qui réinitialise la taille de l'image avec la valeur **scale(1)** pour revenir à la taille initiale.

Sources

- <https://developer.mozilla.org/fr/docs/Web/API/HTMLElement/style>
- <https://developer.mozilla.org/fr/docs/Web/API/HTMLElement/addEventListener>
- <https://developer.mozilla.org/fr/docs/Web/CSS/transition>

Modification du fichier VersionJS.jsx :

Ensuite, nous avons importé cette fonction dans le fichier **VersionJS.jsx** et nous l'avons appelée sur l'image de contact.

Voici le code modifié du fichier **VersionJS.jsx** :

```

1 /**
2 * File Name: VersionJS.jsx
3 * Description: Page principale de la version JavaScript
4 * Author: Damien Vallet
5 * Version: 1.0
6 * Date: 2024
7 * License: Created for UOR course
8 * Language: JavaScript
9 */
10
11 // Importation de la librairie React
12 import React, { useEffect } from "react";
13
14 // Importation des images
15 import picofme from "../img/picofme.jpg";
16
17 // Importation des icônes
18 import { ReactComponent as LinkedinIcon } from "../img/linkedin.svg";
19 import { ReactComponent as GithubIcon } from "../img/github.svg";
20 import { ReactComponent as MastodonIcon } from "../img/mastodon.svg";
21 import { ReactComponent as Soundwave } from '../img/soundwave.svg';
22
23 // Importation du composant vidéo
24 import videopres from "../img/UOR-ch2-video.mov";
25 import Formulaire from "./Formulaire.jsx";
26
27 // Importation des fonctions communes
28 import { changeBgColor, changeTextColor, animateImage } from "../commonJS.js";
29

```

```

30 // définition du composant versionJS
31 function VersionJS() {
32
33     // utilisation du hook useEffect
34     useEffect(() => {
35         [], []);
36
37     // fonction pour changer la couleur de la page
38     const changeColor = (color) => {
39         changeBgColor(color);
40         changeTextColor(color === "white" ? "black" : "white");
41     }
42
43     // fonction pour animer une image
44     const changeImageSize = (event) => {
45         const image = event.target;
46         animateImage(image);
47     }
48
49     return (
50
51         // retourne le code HTML
52         <div id="main">
53
54             {/* ouverture de la balise header */}
55             <header id="header" className="p-3 bg-dark text-light text-center">
56
57                 {/* création d'un container */}
58                 <div className="container-fluid">
59
60                     {/* création d'une ligne */}
61                     <div className="row">
62
63                         {/* création d'une colonne */}
64                         <div className="col-12 col">
65
66                             {/* titre de la page */}
67                             <h1 className="display-1 hero-title">Damien Vallet</h1>
68                             <h2 className="display-2">Portofolio</h2>
69                             <h3>Audio & Product Design / Programmation</h3>
70                         </div>
71                     </div>
72                 </div>
73                 <hr />
74
75             {/* navbar */}
76             <nav id="navbar" className="navbar bg-dark text-light">
77                 <div className="container">
78                     <div className="d-flex flex-wrap align-items-center justify-
79                         content-center justify-content-lg-start">
80                         <Soundwave className="px-2" style={{ fill: 'white', height:
81                             '40px', width: '40px' }} />
82                         <ul className="nav col-12 col-lg-auto me-lg-auto mb-2
83                             justify-content-center mb-md-0">
84                             <li>
85                                 <a className="nav-link px-2 text-light" href="#main">
```

```

83             Home
84                 </a>
85             </li>
86             <li>
87                 <a className="nav-link_px-2_text-light" href="#whoami">
88                     Présentation
89                 </a>
90             </li>
91             <li>
92                 <a className="nav-link_px-2_text-light" href="#projets">
93                     Projets
94                 </a>
95             </li>
96             <li>
97                 <a className="nav-link_px-2_text-light" href="#contact">
98                     Contacts
99                 </a>
100            </li>
101            <li>
102                <a className="nav-link_px-2_text-light" href="#livredor">
103                    />
104            </li>
105
106             /* boutons pour changer la couleur de la page */
107            <li>
108                <button className="btn_mx-2_btn-outline-light" onClick
109                    ={() => changeColor("black")}>
110                    Dark
111                </button>
112            </li>
113            <li>
114                <button className="btn_mx-2_btn-outline-light" onClick
115                    ={() => changeColor("white")}>
116                    Light
117                </button>
118            </li>
119        </ul>
120    </div>
121 </div>
122
123             /* section Présentation */
124 <section id="whoami">
125     <div className="container_p-5">
126         <div className="row">
127             <div className="col">
128                 <h2>
129                     <span className="title">Présentation</span>
130                 </h2>
131             </div>
132         </div>
133         <div className="row">
134             <p className="text-start">
135                 <span className="lettrine">J</span>e suis designer sonore,

```

```
135 compositeur et programmeur. Passionné par la <strong>musique
136 </strong>, les <strong>processus créatifs</strong> ainsi
137 que la <strong>science</strong> et la <strong>technologie
138 </strong>, je travaille
139 depuis 2009 pour la société française&nbsp;
140 <a href="https://www.uvi.net" target="_blank" rel="
141 noreferrer">
142     <strong>UVI Sounds and Software</strong>
143 </a>
144 .
145 </p>
146 <p className="text-start">
147 Pendant ces années, j'ai participé à la conception et au
148 développement de nombreux instruments virtuels tels que :
149 &nbsp;<strong>Soul Drums</strong>
150 &nbsp;et&nbsp; <strong>Glass Orchestra</strong>.
151 </p>
152 <p className="text-start">
153 J'ai également apporté ma contribution à des projets de
154 composition et de production musicale pour des artistes tel
155 que
156 &nbsp;<strong>Mc Solaar</strong>, &nbsp; <strong>Fred Rister
157 </strong> (
158 <strong>David Guetta</strong>),&nbsp; <strong>I Love UFO</
159 strong>,
160 &nbsp;<strong>Martin Guerre</strong>,&nbsp; <strong>Agille</
161 strong>,
162 &nbsp;<strong>The Black Mental</strong>...
163 </p>
164 <p className="text-start">
165 Depuis 2023, je suis étudiant en license informatique à
166 l'université Paris 8.
167 </p>
168 </div>
169 </div>
170 </section>
171
172 /* section Video */
173 <section id="video">
174 <div className="container-p-5">
175 <div className="row">
176 <div className="col">
177     <h2>
178         <span className="title">Vidéo de présentation</span>
179     </h2>
180     </div>
181 </div>
182 <div className="row">
183 <div className="col">
184     <figure>
185         <div className="embed-responsive embed-responsive-4by3 d-
186             flex justify-content-center">
187             <video src={videopres} controls width="320" height="240"
188                 className="embed-responsive-item"></video>
189         </div>
190     </figure>
```

```

181         </div>
182     </div>
183   </div>
184 </section>
185
186 /* section Projets */
187 <section id="projets">
188   <div className="container_p-5_custom-gradient">
189     <div className="row">
190       <div className="col">
191         <h2>
192           <span className="title"> Projets </span>
193         </h2>
194       </div>
195     </div>
196     <div className="row_d-flex_justify-content-center_align-items-
197       center">
198       <div className="col_ratio_ratio-16x9">
199         <iframe className="embed-responsive-item" width="560" height
200           = "315" src="https://www.youtube.com/embed/LHycQ7z5rUE?si=
201             fkmdlyK2lCbW6rwj" title="YouTube_video_player" allow="
202               accelerometer;autoplay;clipboard-write;encrypted-media
203               ;gyroscope;picture-in-picture;web-share"
204             referrerPolicy="strict-origin-when-cross-origin"
205             allowFullScreen />
206       </div>
207       <div className="col">
208         <a
209           href="https://www.uvi.net/glass-orchestra"
210             target="_blank"
211             rel="noreferrer"
212           >
213             Glass Orchestra
214           </a>
215           &nbsp; by UVI Sounds and Software
216         </div>
217       </div>
218       <div className="row_d-flex_justify-content-center_align-items-
219         center">
220         <div className="col">
221           <a
222             href="https://www.uvi.net/en/bass-drums-perc/soul-drums.
223               html"
224               target="_blank"
225               rel="noreferrer"
226             >
227               Soul Drums
228             </a>
229             &nbsp; by UVI Sounds and Software
230           </div>
231         <div className="col_ratio_ratio-16x9">
232           <iframe className="embed-responsive-item" width="560" height
233             = "315" src="https://www.youtube.com/embed/9N-y51YpAcQ?si=
234               02FHB0yuk1iwzv_z" title="YouTube_video_player" allow="
235                 accelerometer;autoplay;clipboard-write;encrypted-media
236                 ;gyroscope;picture-in-picture;web-share"

```

```

        referrerPolicy="strict-origin-when-cross-origin"
        allowFullScreen />
    </div>
</div>
<div className="row_d-flex_justify-content-center_align-items-
center">
    <div className="col_ratio_ratio-16x9">
        <iframe className="embed-responsive-item" width="560" height
            = "315" src="https://www.youtube.com/embed/x4JVg3drMms?si=
            xGZqecbKc8XgB6ro" title="YouTube_video_player" allow="
            accelerometer;_autoplay;_clipboard-write;_encrypted-media
            ;_gyroscope;_picture-in-picture;_web-share"
            referrerPolicy="strict-origin-when-cross-origin"
            allowFullScreen></iframe>
    </div>
    <div className="col">
        Mc Solaar &nbsp;
        <a href="https://youtu.be/x4JVg3drMms?feature=shared">
            <em>Les Mirabelles</em>
        </a>
        &nbsp; sur l'album Géopoétique (2017)
    </div>
    </div>
</div>
</section>

240
241 /* section Contact */
242 <section id="contact">
243     <div className="container_p-5">
244         <div className="row">
245             <h2>
246                 <span className="title">Contacts</span>
247             </h2>
248             <div className="col_d-flex_justify-content-center">
249                 <figure id="contact_mail">
250                     <a href="mailto:damienjdvallet@gmail.com">
251                         <img id="contactimg" src={picofme} alt="portait_de_
252                             Damien_Vallet" width="270" height="250" className="_
253                             img-fluid_rounded_mx-auto_d-block" onMouseOver={_
254                             changeImageSize} />
255                     <p className="text-center">
256                         Vous pouvez m'envoyer un <em>email</em>.
257                     </p>
258                 </a>
259             </figure>
260         </div>
261     </div>
262     <div className="row_p-5">
263         <div className="col_d-flex_justify-content-center_align-items-
264             center">
265             <a href="https://www.linkedin.com/in/damien-vallet-84464195/
266                 ">
267                 <LinkedinIcon className="bg-light_socials-icons" style={{
268                     height: "3em", width: "3em" }} />
269             </a>
270         </div>

```

```

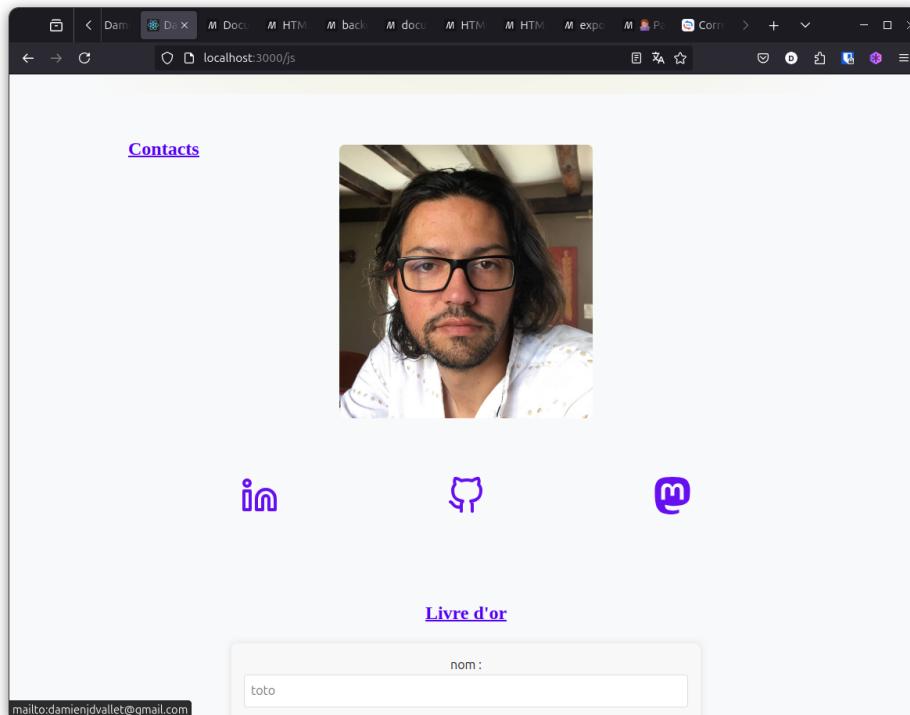
265     <div className="col_d-flex_justify-content-center_align-items-
266         center">
267         <a href="https://github.com/damienvallat">
268             <GithubIcon className="bg-light_socials-icons" style={{
269                 height: "3em", width: "3em" }} />
270         </a>
271     </div>
272     <div className="col_d-flex_justify-content-center_align-items-
273         center">
274         <a href="https://mastodon.social/@darladada">
275             <MastodonIcon className="bg-light_socials-icons" style={{
276                 height: "3em", width: "3em" }} />
277         </a>
278     </div>
279     </div>
280 </section>
281
282 /* section Livre d'or */
283 <section id="livredor">
284     <Formulaire />
285 </section>
286
287 /* footer */
288 <footer id="footmain" className="p-3_bg-dark_text-light" >
289     <div className="container">
290         <div className="row">
291             <div className="col">
292                 <p>2024 - Damien Vallet</p>
293             </div>
294         </div>
295     </div>
296 </footer>
297 </div >
298 );
299 }
300
301 // Exportation du composant VersionJS
302 export default VersionJS;

```

Dans ce fichier, on utilise la fonction **animateImage** sur l'image de contact en passant l'image en paramètre de la fonction. On utilise l'événement **onMouseOver** pour appeler la fonction **changeImageSize** lorsque la souris survole l'image. La fonction **changeImageSize** appelle la fonction **animateImage** sur l'image passée en paramètre. Pour cela on utilise la propriété **target** de l'objet **event** pour cibler l'élément qui a déclenché l'événement.

Sources

- <https://developer.mozilla.org/fr/docs/Web/API/Event/target>
- <https://developer.mozilla.org/fr/docs/Web/API/Event>
- <https://developer.mozilla.org/fr/docs/Web/API/GlobalEventHandlers/onmouseover>
- https://www.w3schools.com/howto/howto_js_animate.asp



Capture d'écran de la page VersionJS avec l'image de contact animée

Conclusion :

En conclusion, nous avons ajouté du JavaScript à notre site web pour rendre notre page plus interactive.
Pour cela, nous avons créé une option "mode sombre" en ajoutant une fonction qui change la couleur de fond et de texte de la page en fonction de l'action de l'utilisateur.
Nous avons également ajouté une fonction qui anime une image lorsqu'elle est survolée par la souris.

Exercice 6.2 (Bonus)

Dans la page actuelle ou dans une nouvelle page, insérez des données au Format JSON prises dans l'un des jeux de données disponibles dans l'Open Data.

- L'affichage des données (idéalement filtrées) sera formatée sous la forme d'un tableau, d'un graphique ou d'une carte.
- Voir par exemple : <https://www.stat4decision.com/fr/10-sites-de-reférence-open-data/>, <https://data.enseignementsup-recherche.gouv.fr/pages/home/>, <https://www.data.gouv.fr/fr/>
- Celles et ceux qui auraient installé Streamlit peuvent utiliser ce framework pour afficher les données avec Python.

Exercice 6.2 (Bonus) :

Pour cet exercice, nous avons décidé d'utiliser les données de data.gouv.fr. pour créer une carte interactive des sites des JO de Paris 2024.

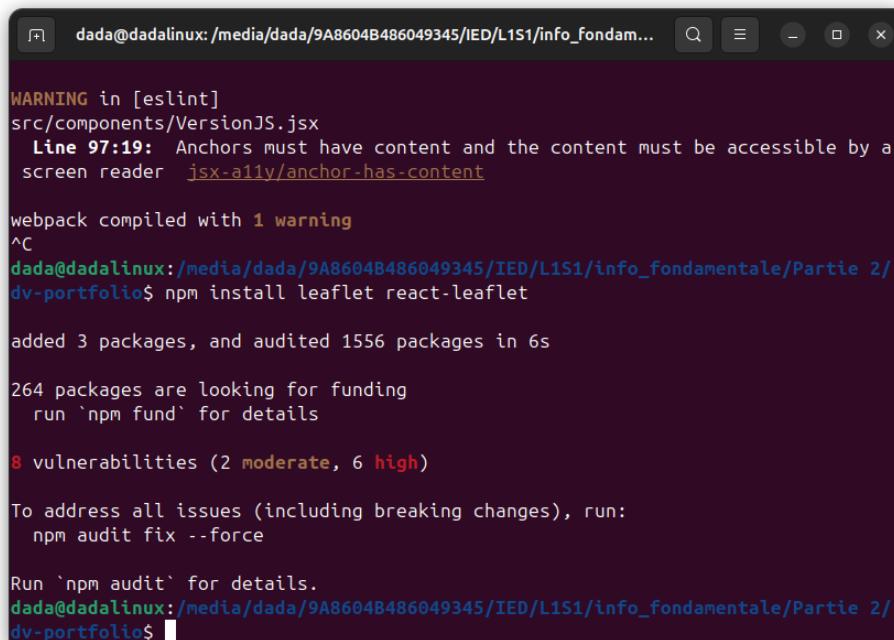
Pour afficher ces données, nous avons utilisé la librairie **Leaflet** qui est une librairie JavaScript open-source pour les cartes interactives.

Nous avons également utilisé la librairie **React-Leaflet** qui est une librairie React pour intégrer Leaflet dans un projet React.

Installation de Leaflet et React-Leaflet :

Pour installer Leaflet et React-Leaflet, nous avons utilisé les commandes suivantes :

```
1 npm install leaflet react-leaflet
```



```

dada@dadalinux:/media/dada/9A8604B486049345/IED/L1S1/info_fondamentale/Partie 2/dv-portfolio$ npm install leaflet react-leaflet

WARNING in [eslint]
src/components/VersionJS.jsx
  Line 97:19:  Anchors must have content and the content must be accessible by a
  screen reader jsx-a11y/anchor-has-content

webpack compiled with 1 warning
^C
dada@dadalinux:/media/dada/9A8604B486049345/IED/L1S1/info_fondamentale/Partie 2/dv-portfolio$ npm install leaflet react-leaflet

added 3 packages, and audited 1556 packages in 6s

264 packages are looking for funding
  run `npm fund` for details

8 vulnerabilities (2 moderate, 6 high)

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

Run `npm audit` for details.
dada@dadalinux:/media/dada/9A8604B486049345/IED/L1S1/info_fondamentale/Partie 2/dv-portfolio$ 
```

Installation de Leaflet et React-Leaflet

Sources

- <https://leafletjs.com/>
- <https://react-leaflet.js.org/>

Modification du fichier NavigationHeader.jsx :

Tout d'abord, nous avons ajouté une nouvelle route pour accéder à la page de la carte.
Voici le code ajusté du fichier **NavigationHeader.js** :

```

1 /**
2 * File Name: NavigationHeader.js
3 * Description: Navigation entre versions
4 * Author: Damien Vallet
5 * Version: 1.0
6 * Date: 2024
7 * License: Created for UOR course
8 * Language: JavaScript
9 */
10
11 // Importation de la librairie React
12 import React from "react";
13
14 // Importation de la librairie React Router
15 import { BrowserRouter as Router, Route, Routes } from "react-router-dom";
16
17 // Importation des composants;
18 import NavigationHeaderBar from "./components/NavigationHeaderBar";
19 import VersioHTML from "./components/VersionHTML";
20 import VersionCSS from "./components/VersionCSS";
21 import VersionPHP from "./components/VersionPHP";
22 import VersionMongo from "./components/VersionMongo";
23 import VersionJS from "./components/VersionJS";
24 import CarteInteractive from "./components/CarteInteractive";
25
26 // Définition de la fonction NavigationHeader
27 function NavigationHeader() {
28     return (
29         //
30         <>
31             {/* composant de navigation */}
32             <Router>
33                 <div>
34
35                     {/* Barre de navigation */}
36                     <NavigationHeaderBar />
37
38                     {/* definition des routes */}
39                     <Routes>
40
41                         {/* route vers la version HTML */}
42                         <Route path="/html" element={<VersioHTML />} />
43
44                         {/* route vers la version CSS */}
45                         <Route path="/css" element={<VersionCSS />} />
46
47                         {/* route vers la version PHP */}
48                         <Route path="/php" element={<VersionPHP />} />
49
50                         {/* route vers la version MongoDB */}
51                         <Route path="/mongo" element={<VersionMongo />} />

```

```

52         {/* route vers la version JavaScript */}
53         <Route path="/js" element={<VersionJS />} />
54
55         {/* route vers la carte Interactive */}
56         <Route path="/carte" element={<CarteInteractive />} />
57     </Routes>
58 </div>
59 </Router>
60 </>
61 );
62 }
63 }
64
65 // Exportation du composant NavigationHeader
66 export default NavigationHeader;

```

Modification du fichier NavigationHeaderBar.jsx :

On modifie également le fichier `NavigationHeaderBar.js` pour ajouter un lien vers la nouvelle page.

Voici le code du fichier `NavigationHeaderBar.js` :

```

1 /**
2 * File Name: NavigationHeaderBar.jsx
3 * Description: Composant qui gère la barre pour naviguer entre les
4 *               versions
5 * Author: Damien Vallet
6 * Version: 1.0
7 * Date: 2024
8 * License: Created for UOR course
9 * Language: JavaScript
10 */
11
12 // Importation de la librairie React
13 import React from "react";
14
15 // Importation de la librairie React Router
16 import { Link } from "react-router-dom";
17
18 // définition du composant NavBar
19 function NavigationHeaderBar() {
20     return (
21
22         // Barre de navigation
23         <nav id="header-navbar">
24             {/* Une liste non ordonnée (<ul>) définit les éléments de
25                 navigation. Chaque élément de navigation est un élément de
26                 liste (<li>)*/}
27             <ul>
28                 <li>
29                     {/* Lien vers la version HTML */}
30                     <Link to="/html">Version HTML</Link>
31                 </li>
32                 <li>
33                     {/* Lien vers la version avec CSS/Bootstrap */}
34                 </li>
35             </ul>
36         </nav>
37     );
38 }
39
40 // Exportation du composant NavigationHeader
41 export default NavigationHeaderBar;

```

```

31         <Link to="/css">Version CSS</Link>
32     </li>
33     <li>
34         {/* lien vers la version PHP avec base SQL */}
35         <Link to="/php">Version PHP</Link>
36     </li>
37     <li>
38         {/* lien vers la version avec MongoDB */}
39         <Link to="/mongo">Version MongoDB</Link>
40     </li>
41     <li>
42         {/* Lien vers la version du site contenant des animations en
43             JavaScript */}
44         <Link to="/js">Version JavaScript</Link>
45     </li>
46     <li>
47         {/* Lien vers la carte interactive */}
48         <Link to="/carte">Carte Interactive</Link>
49     </li>
50   </ul>
51 </nav>
52 );
53
54 // Exportation du composant NavigationHeaderBar
55 export default NavigationHeaderBar;

```

Création du composant CarteInteractive.jsx :

Ensuite, nous avons créé un nouveau composant **CarteInteractive.js** pour afficher la carte interactive des sites des JO de Paris 2024.

Voici le code du fichier **CarteInteractive.js** :

```

1 /**
2 * File Name: CarteInteractive.jsx
3 * Description: Composant pour afficher une carte interactive
4 * Author: Damien Vallet
5 * Version: 1.0
6 * Date: 2024
7 * License: Created for UOR course
8 * Language: JavaScript
9 */
10
11 // Importation de la bibliothèque React
12 import React, { useState, useEffect } from "react";
13
14 // Importation de la bibliothèque Leaflet
15 import { MapContainer, TileLayer, GeoJSON } from 'react-leaflet';
16
17 // Fonction pour traduire la catégorie des sites
18 const translateCategory = (category) => {
19     switch (category) {
20         case "venue-olympic":
21             return 'Site Olympique';

```

```

22     case "venue-paralympic":
23         return 'Site Paralympique';
24     default:
25         return 'Inconnu';
26     }
27 };
28
29 // fonction onEachFeature pour afficher les informations de chaque feature
30 const onEachFeature = (feature, layer) => {
31
32     // Vérification de l'existence des propriétés
33     if (feature.properties) {
34
35         // Contenu du popup
36         const popupContent = `
37             <p><strong>Site:</strong> ${feature.properties.nom_site}</p>
38             <p><strong>Catégorie:</strong> ${translateCategory(feature.
39                 properties.category_id)}</p>
40             <p><strong>Discipline:</strong> ${feature.properties.sports}</p>
41             `;
42
43         // Ajout du contenu au popup
44         layer.bindPopup(popupContent);
45     }
46 };
47
48 // Définition du composant CarteInteractive
49 function CarteInteractive() {
50
51     // Utilisation du hook useState pour définir l'état de la variable
52     // geojsonData
53     const [geojsonData, setGeojsonData] = useState(null);
54
55     // Utilisation du hook useEffect pour importer les données GeoJSON
56     useEffect(() => {
57
58         // Importation des données GeoJSON
59         fetch(`${process.env.PUBLIC_URL}/data/paris-2024-sites-de-
60             competition.geojson`)
61
62             // Conversion des données en JSON
63             .then(response => response.json())
64
65             // Mise à jour de la variable geojsonData
66             .then(data => setGeojsonData(data))
67
68             // Gestion des erreurs
69             .catch(error => console.error('Error fetching GeoJSON:', error
70                 ));
71     }, []);
72
73     return (
74         // Carte interactive
75         <div className="container mt-5">
76
77             {/* titre de la carte */}
78
79         </div>
80     );
81 }

```

```
74     <h1 className="text-center mb-4">Carte Interactive des Sites  
75         de Compétition des JO 2024</h1>  
76  
77     {/* Carte interactive */}  
78     <div id="map" className="map-border-rounded-shadow-sm">  
79  
80         {/* Carte Leaflet */}  
81         <MapContainer center={[48.8566, 2.3522]} zoom={13}  
82             scrollWheelZoom={false} style={{ height: "900px", width  
83                 : "100%" }}>  
84  
85             {/* Calque de tuiles OpenStreetMap */}  
86             <TileLayer  
87                 attribution='&copy; <a href="https://www.  
88                 openstreetmap.org/copyright">OpenStreetMap</a>  
89                 contributors'  
90                 url="https://s.tile.openstreetmap.org/{z}/{x}/{y  
91                 }.png"  
92             />  
93  
94             {/* On affiche les marqueurs issus des données GeoJSON  
95             */}  
96             {geojsonData && <GeoJSON data={geojsonData}  
97                 onEachFeature={onEachFeature} />}  
98         </MapContainer>  
99     </div>  
100    </div>  
101 );  
102 }  
103  
104 // Exportation du composant CarteInteractive  
105 export default CarteInteractive;
```

Décomposition du code :

Dans ce composant, nous avons utilisé la librairie **react-leaflet** pour afficher une carte interactive des sites des Jeux Olympiques de Paris 2024.

Tout d'abord, nous avons récupéré les données GeoJSON des sites des JO de Paris 2024 depuis le site data.gouv.fr.

Pour créer notre carte, nous avons importé les composants **MapContainer**, **TileLayer** et **GeoJSON** de la librairie **react-leaflet**.

- Le composant **MapContainer** est utilisé pour afficher la carte.
- Le composant **TileLayer** est utilisé pour afficher les tuiles OpenStreetMap.
- Le composant **GeoJSON** est utilisé pour afficher les données GeoJSON sur la carte.

Ensuite, nous utilisons le hook **useState** pour mettre à jour la variable **geoJsonData** qui contiendra les données GeoJSON. Dans le hook **useEffect**, nous utilisons la fonction **fetch** pour importer nos données. Nous avons converti les données en JSON et mis à jour la variable **geoJsonData** avec ces données.

Nous avons créé la fonction **onEachFeature** pour traiter les informations de chaque élément (feature) de la collection.

1. Cette fonction prend en paramètre la feature et le calque sur lequel elle est affichée.
2. Une feature est un élément de la carte, par exemple un marqueur.
3. On vérifie l'existence des propriétés de la feature et on crée un popup avec les informations de la feature.
4. Enfin, on ajoute le contenu du popup au calque avec la méthode **bindPopup**.

Nous avons également écrit la fonction **translateCategory** pour traduire la catégorie des sites en français et retourner le nom correspondant.

Du côté de notre composant, nous avons affiché la carte interactive en utilisant le composant **MapContainer** avec les propriétés **center** et **zoom** pour center la carte sur Paris et le niveau de zoom de la carte. Ce composant contient un **TileLayer** pour afficher les tuiles OpenStreetMap et un composant **GeoJSON** pour afficher les données GeoJSON sur la carte en utilisant la fonction **onEachFeature** pour afficher les informations de chaque feature.

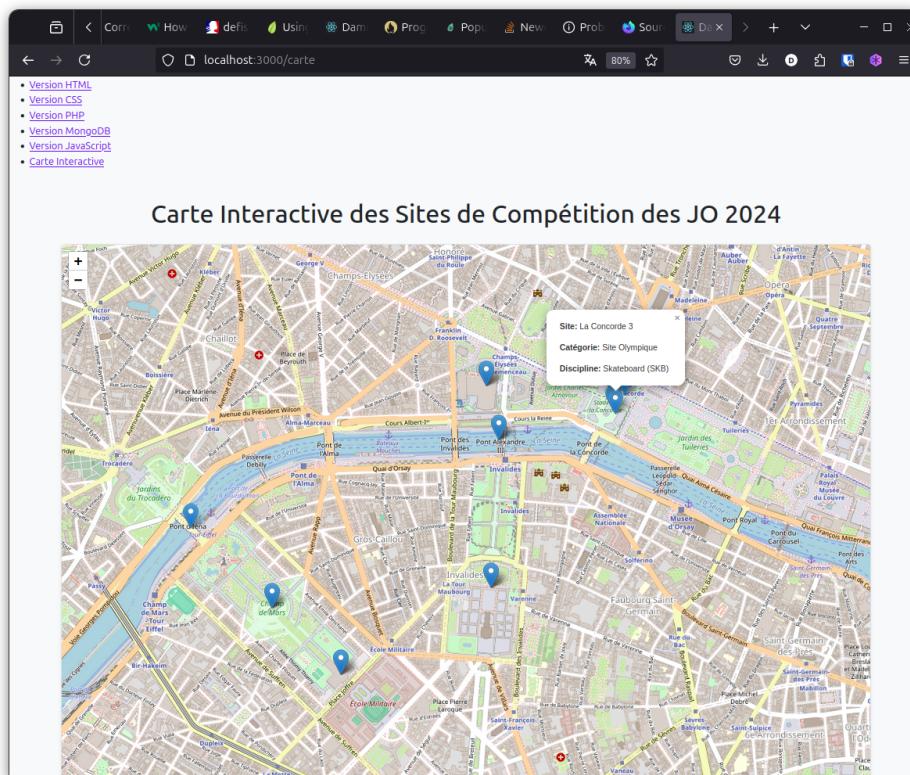
En conclusion, nous avons ajouté une carte interactive des sites des JO de Paris 2024 à notre site web en utilisant les données GeoJSON de data.gouv.fr.

Pour cela, nous avons utilisé la librairie **Leaflet** pour les cartes interactives et la librairie **React-Leaflet** pour intégrer Leaflet dans notre projet React.

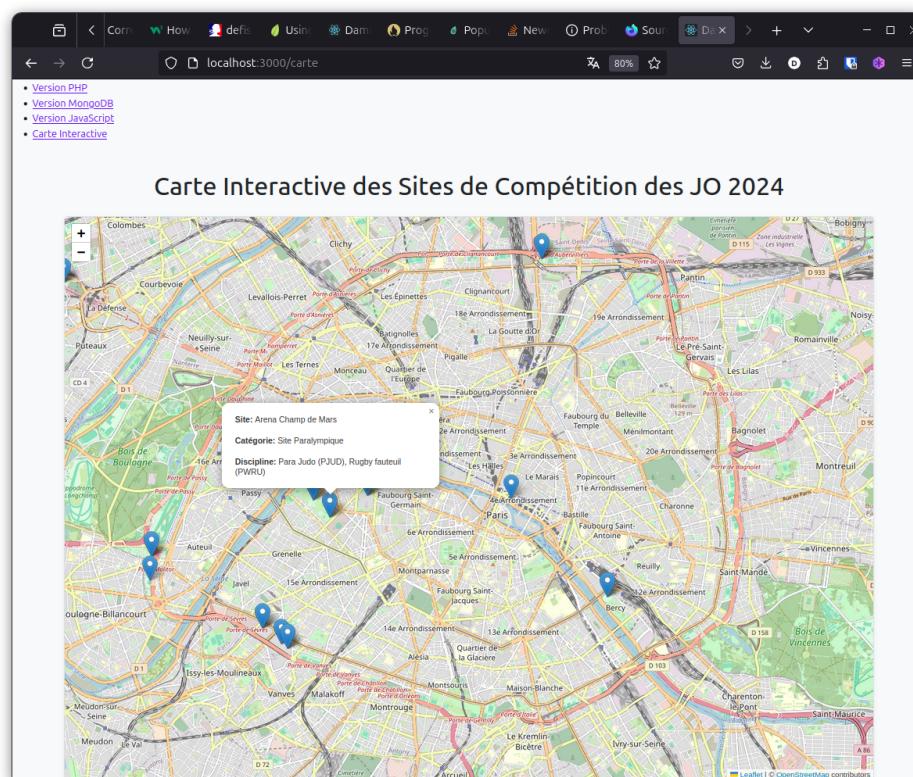
Les codes source de ce chapitre sont disponibles dans l'annexe E

Sources

- <https://defis.data.gouv.fr/defis/66a0ffa8d8f03afc1d474a53/>
- https://developer.mozilla.org/fr/docs/Web/API/Fetch_API
- <https://leafletjs.com/examples/geojson/>
- <https://react-leaflet.js.org/docs/example-popup-marker/>
- <https://react-leaflet.js.org/docs/api-map/#mapcontainer>
- <https://leafletjs.com/reference.html#tilelayer>
- <https://leafletjs.com/reference.html#geojson>
- https://www.youtube.com/watch?v=i9oX1upSKjI&list=PLKmmzLM3RgQVFqnNqn3CI2Cxg4Rd__VbK



Carte interactive des sites des JO de Paris 2024



Carte interactive des sites des paralympic JO de Paris 2024

7

Chapitre 7 - Internet en dehors du Web

Exercice 7.1 (A Rendre)

- (1) Après les avoir compressés en une seule archive, transférez les sources(*) de votre site (répertoire "htdocs" ou Document Root) de votre hébergeur (ou en local dans un autre dossier selon) en utilisant le protocole FTP.
- (2) En utilisant les commandes netcat, ssh, scp, wget, git,... (selon), déposez votre archive (obtenue en (1)) dans un service d'hébergement gratuit de fichiers de votre choix ou sur un serveur d'gestion de versions décentralisé (Github, Gitea, GitLab, BitBucketSVN, Mercurial, SourceForge, ...)
- Comme à l'habituée, n'oubliez pas de commenter les différentes opérations et d'illustrer celles-ci avec des C.E(s). Insérez également dans votre réponse le ou les liens correspondant.s afin que je puisse accéder à distance à votre archive.
- Sources (*) : N'oubliez pas d'ajouter à votre archive, les sources du chatbot de l'exercice 7.2.

Exercice 7.1 - Transfert des sources du site

Pour réaliser cet exercice, nous avons utilisé le protocole FTP pour transférer les sources compressées de notre site web depuis notre dossier de travail vers un dossier de sauvegarde en local.

Compression des sources du site

Pour compresser nos données, nous avons utilisé la commande **tar**. La commande **tar** permet de créer des archives au format **tar** et de les compresser avec des outils comme **gzip**. Voici la commande que nous avons utilisée :

```
1 tar zcvf uor-part2-archive.tar.gz dv-portfolio/ dv-chatbot/ docuWiki/
 mongo-srv/ minecraft-server/
```

La commande **tar** prend plusieurs options :

- **z** : pour compresser l'archive avec **gzip**.
- **c** : pour créer une nouvelle archive.
- **v** : pour afficher les fichiers ajoutés à l'archive.
- **f** : pour spécifier le nom de l'archive.

Dans cette partie, nous avons utilisé la commande **tar** pour compresser les sources de notre site web dans une archive au format **tar.gz**.

Sources

- <https://doc.ubuntu-fr.org/tar>



A screenshot of a terminal window titled 'dada@dadalinux: /media/dada/9A8604B486049345/IED/L1S1/info_fondamentale/Partie 2\$'. The window contains the command 'tar zcvf uor-part2-archive.tar.gz dv-portfolio/ dv-chatbot/ docuWiki/ mongo-srv/ minecraft-server/'. The terminal has a dark background and light-colored text.

Compression des sources du site

Transfert FTP

Dans cette section, nous avons utilisé le protocole FTP pour transférer l'archive de nos sources vers un dossier de sauvegarde en local.

Tout d'abord, nous avons créé un dossier **Archivage**, dans le répertoire **/Desktop**, dans lequel nous allons copié l'archive de nos sources. Ensuite, pour monter un serveur FTP sur notre machine Ubuntu, nous avons utilisé la commande **vsftpd** pour installer le serveur FTP.

```
1 sudo apt-get install vsftpd
```

```
dada@dadalinux:/media/dada/9A8604B486049345/IED/L1S1/info_fondamentale/Partie 2$ sudo apt install vsftpd
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  vsftpd
0 upgraded, 1 newly installed, 0 to remove and 3 not upgraded.
Need to get 120 kB of archives.
After this operation, 312 kB of additional disk space will be used.
Get:1 http://fr.archive.ubuntu.com/ubuntu noble amd64 vsftpd amd64 3.0.5-0ubuntu3 [120 kB]
Fetched 120 kB in 0s (643 kB/s)
Preconfiguring packages ...
Selecting previously unselected package vsftpd.
(Reading database ... 219976 files and directories currently installed.)
Preparing to unpack .../vsftpd_3.0.5-0ubuntu3_amd64.deb ...
Unpacking vsftpd (3.0.5-0ubuntu3) ...
Setting up vsftpd (3.0.5-0ubuntu3) ...
Created symlink /etc/systemd/system/multi-user.target.wants/vsftpd.service → /usr/lib/systemd/system/vsftpd.service.
Processing triggers for man-db (2.12.0-4build2) ...
dada@dadalinux:/media/dada/9A8604B486049345/IED/L1S1/info_fondamentale/Partie 2$
```

Installation de vsftpd

Après l'installation de **vsftpd**, nous avons démarré le serveur FTP avec la commande suivante:

```
1 sudo systemctl start vsftpd
```

Nous avons finalement vérifié que le serveur FTP est en cours d'exécution avec la commande suivante:

```
1 sudo systemctl status vsftpd
```

```
dada@dadalinux:/media/dada/9A8604B486049345/IED/L1S1/info_fondamentale/Partie 2$ sudo systemctl status vsftpd
● vsftpd.service - vsftpd FTP server
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; enabled; preset: enabled)
   Active: active (running) since Sat 2024-08-03 15:45:04 CEST; 4min 9s ago
     Process: 23809 ExecStartPre=/bin/mkdir -p /var/run/vsftpd/empty (code=exited, status=0/SUCCESS)
    Main PID: 23812 (vsftpd)
       Tasks: 1 (limit: 18917)
      Memory: 704.0K (peak: 1.5M)
        CPU: 18ms
       CGroup: /system.slice/vsftpd.service
               └─23812 /usr/sbin/vsftpd /etc/vsftpd.conf

lines 1-11
```

Status de vsftpd

Ensuite, nous avons lancé le client FTP avec la commande suivante:

```
1 ftp localhost
```

```
dada@dadalinux: /media/dada/9A8604B486049345/IED/L1S1/info_fondamentale/P...
dada@dadalinux: ~/Desktop/...  dada@dadalinux: /media/dada/9A8604B486049345/IED/L1S1/info_fondamentale/P...
dada@dadalinux: /  ftp> open localhost
Connected to localhost.
220 (vsFTPd 3.0.5)
Name (localhost:dada): dada
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> 
```

Connexion au serveur FTP

```
dada@dadalinux: /media/dada/9A8604B486049345/IED/L1S1/info_fondamentale/Partie 2  ftp> cd /home/dada/Desktop/Archivage
250 Directory successfully changed.
ftp> lcd /media/dada/9A8604B486049345/IED/L1S1/info_fondamentale/Partie\ 2/
Local directory now: /media/dada/9A8604B486049345/IED/L1S1/info_fondamentale/Partie 2
ftp> put uor-part2-archive.tar.gz
local: uor-part2-archive.tar.gz remote: uor-part2-archive.tar.gz
229 Entering Extended Passive Mode (|||49068|)
150 Ok to send data.
100% [*****] 276 MiB 33.75 MiB/s 00:00 ETA
226 Transfer complete.
290293994 bytes sent in 00:08 (33.75 MiB/s)
ftp> 
```

Transfert de l'archive avec FTP

Nous nous sommes connecté au serveur FTP avec les identifiants de notre machine.

Nous avons ensuite changé de répertoire remote avec la commande **cd** pour accéder au dossier de sauvegarde **Archivage**. Nous avons également changé le dossier local avec la commande **lcd** pour accéder au dossier contenant l'archive de nos sources.

Enfin, nous avons transféré l'archive de nos sources avec la commande **put**.

put uor-part2-archive.tar.gz

Dans cette partie, nous avons compressé les sources de nos exercices avec la commande **tar** et transféré l'archive vers un dossier de sauvegarde en local avec le protocole FTP.

Pour cela, nous avons installé le serveur FTP **vsftpd** sur notre machine et démarré le serveur. Nous avons lancé le client FTP et nous nous sommes connecté au serveur. Nous avons mis à jour les répertoires de travail et transféré l'archive avec la commande **put**.

Sources

- <https://doc.ubuntu-fr.org/vsftpd>
- <https://doc.ubuntu-fr.org/tar>
- <https://ubuntu.com/server/docs/set-up-an-ftp-server>
- <https://phoenixnap.com/kb/linux-ftp>

Transfert des sources du site avec Git vers Github

Pour réaliser cet exercice, nous avons utilisé **Git** pour transférer les sources de notre site web de notre machine vers Github.

On doit d'abord s'assurer de la bonne installation de Git sur notre machine. Pour cela, on ouvre un terminal et on tape la commande **git --version** pour vérifier la version de Git installée sur notre machine.

```
1 git --version
```

Si Git n'est pas déjà présent et après avoir mis les paquets à jour, on installe Git avec la commande suivante:

```
1 sudo apt-get install git-all
```

```
dada@dadalinux:/media/dada/9A8604B486049345/IED/L1S1/info_fondamentale/Partie 2/mongo-srv$ sudo apt-get install git-all
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2 apache2-bin apache2-data apache2-cvs cvs cvsps git-cvs git-doc git-email
  git-gui git-mediawiki git-svn gitk gitweb libalgorithm-c3-perl libaprutil1-ldap
  libb-hooks-endofscope-perl libbb-hooks-op-check-perl libclass-c3-perl
  libclass-c3-xs-perl libclass-data-inheritable-perl libclass-inspector-perl
  libclass-method-modifiers-perl libclass-singleton-perl libclass-xsaccessor-perl
  libdata-optlist-perl libdatetime-format-builder-perl libdatetime-format-iso8601-perl
  libdatetime-format-strptime-perl libdatetime-locale-perl libdatetime-perl
  libdatetime-timezone-perl libdbd-sqlite3-perl libdbi-perl libdevel-callchecker-perl
  libdevel-caller-perl libdevel-lexalias-perl libdevel-stacktrace-perl
  libdigest-bubblebubble-perl libdigest-hmac-perl libdynaloader-functions-perl
  libemail-valid-perl libeval-closure-perl libexception-class-perl libfile-sharedir-perl
  libmediawiki-api-perl libmodule-implementation-perl libmodule-runtime-perl
  libmro-compat-perl libnamespace-autoclean-perl libnamespace-clean-perl libnet-dns-perl
  libnet-dns-sec-perl libnet-domain-tld-perl libnet-libidn-perl libpackage-stash-perl
  libpackage-stash-xs-perl libpadwalker-perl libparams-classify-perl libparams-util-perl
  libparams-validate-perl libparams-validationcompiler-perl libperl4-corelibs-perl
  libreadonly-perl libref-util-perl libref-util-xs-perl librole-tiny-perl libserf-1-1
  libspecio-perl libsub-exporter-perl libsub-exporter-progressive-perl
  libsub-identify-perl libsub-install-perl libsub-name-perl libsub-quote-perl libsvn-perl
  libsvn1 libterm-readkey-perl libutf8proc3 libvariable-magic-perl libxstring-perl
  libyaml-libyaml-perl libyaml-perl tk tk8.6
Suggested packages:
```

Installation de Git

On doit ensuite configurer Git avec notre nom et notre adresse email :

```
1 git config --global user.name "John_Doe"
2 git config --global user.email "john@doe.com"
```

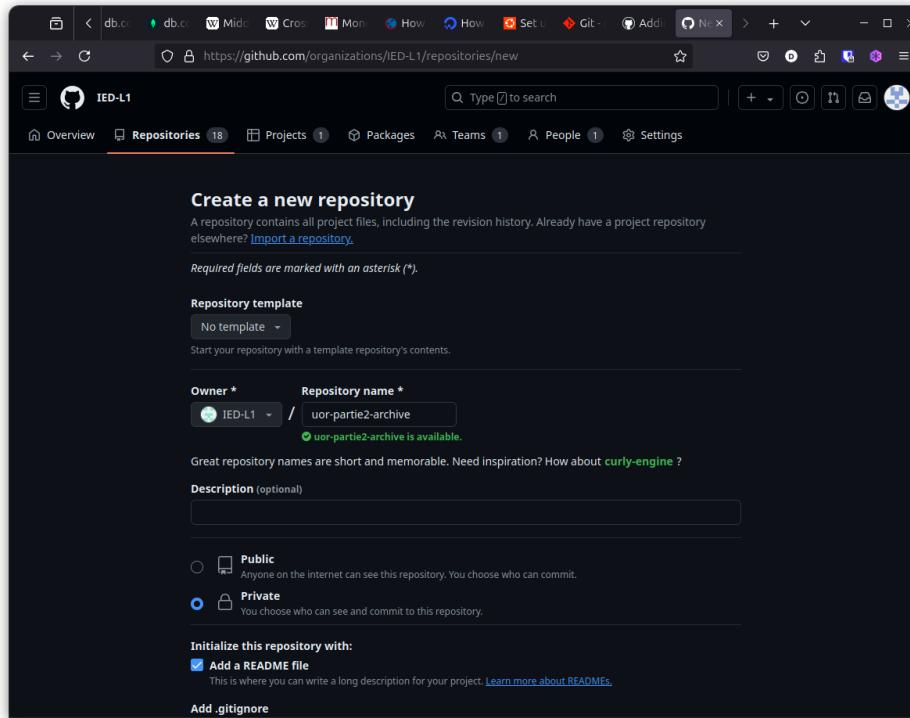
```
dada@dadalinux:~$ git config -l
user.name=d.vallet
user.email=damienjdvallet@gmail.com
dada@dadalinux:~$
```

Configuration de Git

On configure enfin notre dépôt local avec la commande :

```
1 git init
```

On doit ensuite créer un dépôt distant sur Github pour y ajouter les fichiers de notre site web.



Création du dépôt distant sur Github

On ajoute le dépôt distant avec à notre configuration à l'aide de la commande suivante:

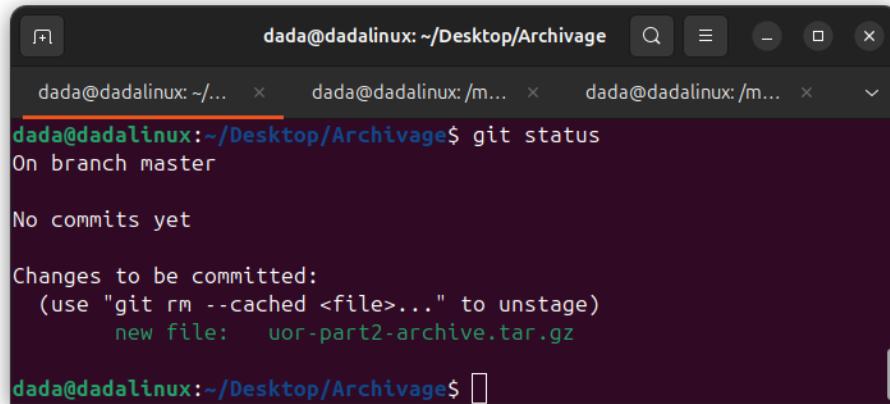
```
1 git remote add origin https://github.com/IED-L1/uor-partie2-archive.git
```

On ajoute ensuite les fichiers de notre site web dans le dépôt local avec la commande suivante:

```
1 git add .
```

On vérifie que les fichiers ont bien été ajoutés avec la commande suivante:

```
1 git status
```



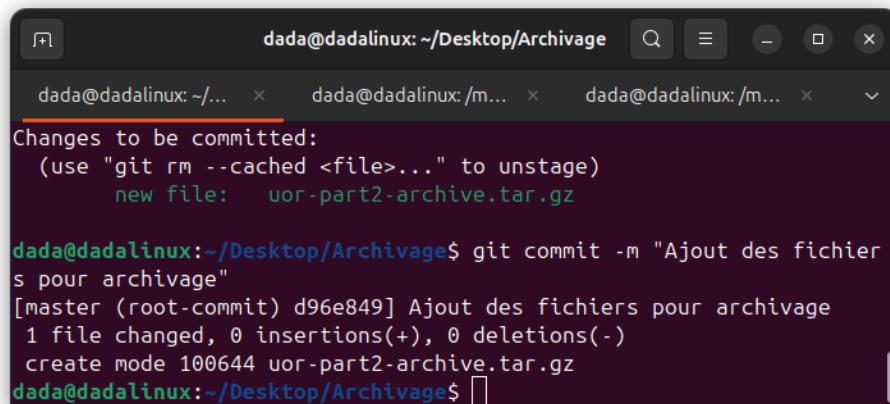
```
dada@dadalinux:~/Desktop/Archivage$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file: uor-part2-archive.tar.gz

dada@dadalinux:~/Desktop/Archivage$
```

Ajout des fichiers



```
dada@dadalinux:~/Desktop/Archivage$ git commit -m "Ajout des fichiers pour archivage"
[master (root-commit) d96e849] Ajout des fichiers pour archivage
 1 file changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 uor-part2-archive.tar.gz
dada@dadalinux:~/Desktop/Archivage$
```

Commit des fichiers

Git ne permet pas de versionner les fichiers volumineux. La limite de taille des fichiers est de **100 Mo**. Dans notre cas, notre archive est trop volumineuse, on doit donc utiliser la commande **git lfs** pour gérer les fichiers volumineux.

On installe **git lfs** qui est un outil de gestion des fichiers volumineux.

```
1 sudo apt-get install git-lfs
```

On initialise **git lfs** avec la commande suivante:

```
1 git lfs install
```

```
dada@dadalinux:~/Desktop/Archivage$ curl -s https://packagecloud.io/install/repositories/github/git-lfs/script.deb.sh | sudo bash
[sudo] password for dada:
Detected operating system as Ubuntu/noble.
Checking for curl...
Detected curl...
Checking for gpg...
Detected gpg...
Detected apt version as 2.7.14
Running apt-get update... done.
Installing apt-transport-https... done.
Installing /etc/apt/sources.list.d/github_git-lfs.list...done.
Importing packagecloud gpg key... Packagecloud gpg key imported to /etc/apt/keyrings/github_git-lfs-archive-keyring.gpg
done.
Running apt-get update... done.

The repository is setup! You can now install packages.
dada@dadalinux:~/Desktop/Archivage$ 
```

Installation de Git LFS

On ajoute les fichiers volumineux avec la commande suivante:

```
1 git lfs track "*.tar.gz"
```

```
dada@dadalinux:~/Downloads/git-lfs-linux-amd64-v3.5.1$ sudo ./git-lfs-3.5.1/install.sh
[sudo] password for dada:
Git LFS initialized.
dada@dadalinux:~/Downloads/git-lfs-linux-amd64-v3.5.1$ git lfs install
Git LFS initialized.
dada@dadalinux:~/Downloads/git-lfs-linux-amd64-v3.5.1$ cd ~/Desktop/Archivage/
dada@dadalinux:~/Desktop/Archivage$ ls
uor-part2-archive.tar.gz
dada@dadalinux:~/Desktop/Archivage$ git lfs track "*.tar.gz"
Tracking "*.tar.gz"
dada@dadalinux:~/Desktop/Archivage$ 
```

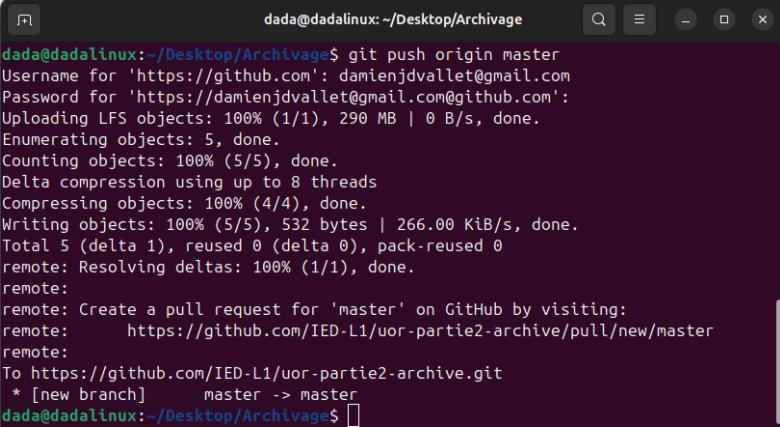
Commit des fichiers volumineux

On commit les fichiers volumineux avec la commande suivante:

```
1 git add .gitattributes
2 git commit -m "Ajout des fichiers pour archivage"
```

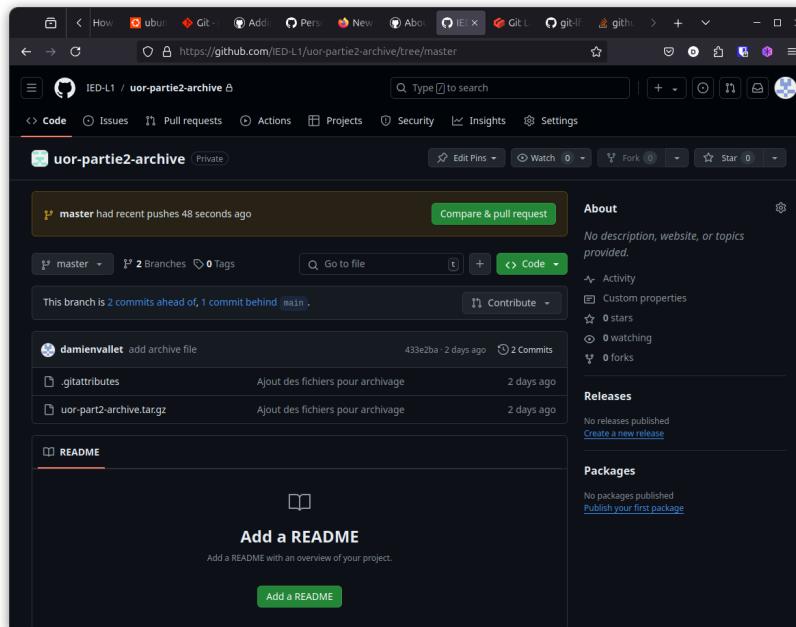
On pousse ensuite les fichiers du dépôt local vers le dépôt distant avec la commande suivante:

```
1 git push origin master
```



```
dada@dadalinux:~/Desktop/Archivage$ git push origin master
Username for 'https://github.com': damienjdvallet@gmail.com
Password for 'https://damienjdvallet@gmail.com@github.com':
Uploading LFS objects: 100% (1/1), 290 MB | 0 B/s, done.
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 532 bytes | 266.00 KiB/s, done.
Total 5 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:     https://github.com/IED-L1/uor-partie2-archive/pull/new/master
remote:
To https://github.com/IED-L1/uor-partie2-archive.git
 * [new branch]      master -> master
dada@dadalinux:~/Desktop/Archivage$
```

Upload des fichiers vers le dépôt distant



Notre dépôt distant sur Github après l'upload

En conclusion, nous avons transféré les sources de notre site web de notre machine vers Github avec Git. Pour cela, nous avons installé et configuré Git avec notre nom et notre adresse email, ajouté les fichiers de notre site web dans le dépôt local, "commit" les fichiers ajoutés et uploader les fichiers du dépôt local vers le dépôt distant.

Nous avons enfin renommé notre archive et ajouté les autres fichiers de ce cours dans le dépôt Github.

L'archive des sources de notre site web est disponible via le lien suivant:

[https://github.com/IED-L1/uor-partie2-archive/
blob/b73147dda2822f9a17ebdf60a656bb4f7af31ae1/
IED-L1-IFPART2-DamienVallet-23000113-codesource.tar.gz](https://github.com/IED-L1/uor-partie2-archive/blob/b73147dda2822f9a17ebdf60a656bb4f7af31ae1/IED-L1-IFPART2-DamienVallet-23000113-codesource.tar.gz)

Le dépôt Github contenant les sources de ce cours est disponible via le lien suivant:

<https://github.com/IED-L1/uor-partie2-archive>

Sources

- <https://git-lfs.com/>
- <https://git-scm.com/docs/gittutorial>
- <https://github.com/git-guides/install-git>
- <https://github.com/git-guides/git-remote>
- <https://openclassrooms.com/fr/courses/7162856-gerez-du-code-avec-git-et-github>
- <https://docs.github.com/en/repositories/working-with-files/managing-large-files/about-large-files-on-github>

Exercice 7.2 (Mini-Projet - à Rendre)

- Avec le langage et/ou le framework de votre choix, réalisez un mini-chatbot qui puisse dialoguer avec un utilisateur humain.
- Ce chatbot doit pouvoir répondre à des "bonjour" ..., "bonsoir" ..., et à un minimum de 5 questions simples de votre choix.
- Si vous le souhaitez, vous pouvez développer votre chatbot en local, chez votre hébergeur, ou à l'aide d'assistants comme IBM Watson, Google DialogFlow, Amazon Lex (AWS) ou Alexa, Chatfuel, ...
- Insérez dans votre réponse les codes commentés (C.C), et bien évidemment les C.E(s) des (>= 5) exemples d'utilisation

Exercice 7.2 - Réalisation du Chatbot

Pour cet exercice, nous avons choisi d'écrire un chatbot en Python avec la librairie **ChatterBot**. Pour cela nous avons suivi les tutoriels de la documentation de la librairie **ChatterBot** ainsi que le tutoriel **Real Python**.

Installation de ChatterBot

Pour installer ChatterBot, nous avons d'abord créé un dossier intitulé **dv-chatbot** dans lequel nous avons ajouté un container avec Docker. Pour cela nous avons créé un dossier **./devcontainer** dans lequel nous avons mis un fichier **Dockerfile** avec le contenu suivant:

```

1 # On utilise l'image python:3.8 comme base
2 FROM python:3.8
3
4 # On définit le répertoire de travail comme /workspace
5 WORKDIR /workspace
6
7 # on met à jour les paquets et on installe les paquets nécessaires
8 # on installe build-essential
9 # on supprime les fichiers temporaires pour réduire la taille de l'image
10 RUN apt-get update && apt-get install -y \
    build-essential \
    && rm -rf /var/lib/apt/lists/*
11
12
13
14 # On copie le fichier requirements.txt dans le répertoire de travail
15 COPY requirements.txt /workspace/
16
17 # On installe les dépendances
18 RUN pip install --no-cache-dir -r requirements.txt
19
20 # On copie le contenu du répertoire courant dans le répertoire de travail
21 COPY . /workspace/

```

1. On importe l'image de base **python:3.8** pour créer le container.
2. On crée un dossier **/workspace** dans le container.
3. On copie le fichier **requirements.txt** dans le container.
4. On installe les dépendances du fichier **requirements.txt** avec la commande **pip install -r requirements.txt**.

On ajoute ensuite un fichier **devcontainer.json** dans le dossier **.devcontainer**. Le fichier **devcontainer.json** contient les configurations du container pour l'utiliser dans Visual Studio Code.

Voici le contenu du fichier **devcontainer.json**:

```

1 {
2     // nom du conteneur de développement
3     "name": "Python_3",
4
5     // configuration de l'image Docker
6     "build": {
7         // on choisit l'image de base
8         "dockerfile": "Dockerfile",
9
10        // on ajoute des arguments pour le Dockerfile
11        "context": "...",

```

```

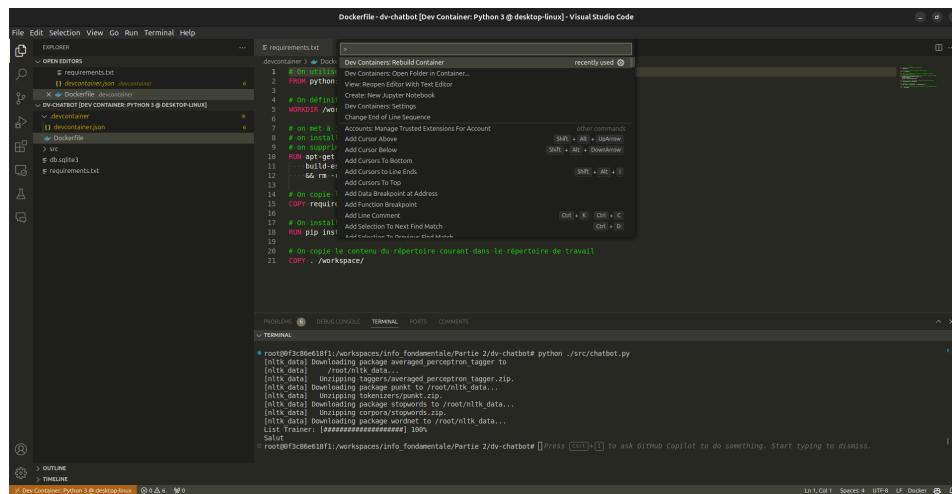
12 },
13
14 // reglages à appliquer au conteneur
15 "settings": {
16     // on choisit le shell à utiliser par défaut dans le terminal
17     "terminal.integrated.shell.linux": "/bin/bash",
18 },
19
20 // liste des extensions VS Code à installer dans le conteneur
21 "extensions": [
22     "ms-python.python"
23 ],
24
25 // liste des commandes à exécuter après la création du conteneur
26 "postCreateCommand": "pip_install--no-cache-dir-r_requirements.txt",
27
28 // utilisateur par défaut pour lancer des commandes dans le conteneur
29 "remoteUser": "root"
30 }

```

Dans VSCode, nous avons ouvert le dossier **dv-chatbot** et sélectionné la commande **Remote-Containers: Re-ouvrir dans un container** pour ouvrir le dossier dans le container.

Un environnement virtuel est un environnement Python isolé qui permet de gérer les dépendances de votre projet.

Il permet de créer un environnement propre pour chaque projet Python et d'éviter les conflits entre les dépendances de différents projets.



Création du container avec Docker

Dans cette partie, nous avons créé un container Docker avec l'image de base **python:3.8** et la librairie **ChatterBot** pour réaliser le chatbot.

Création du Chatbot

Pour créer le chatbot, nous avons créé un fichier `chatbot.py` dans lequel nous avons écrit le code suivant:

```

1 # Importation de la classe ChatBot et de la classe ListTrainer du module
2   chatterbot
3 from chatterbot import ChatBot
4 from chatterbot.trainers import ListTrainer

```

On importe la classe `ChatBot` de la librairie `ChatterBot` et la classe `ListTrainer` pour entraîner le chatbot.

On crée une instance de la classe `ChatBot` avec le nom `Darla Chatbot` et une instance de la classe `ListTrainer` pour entraîner le chatbot. Pour l'instance de Chatbot, on passe le nom du chatbot et on spécifie le `logic adapter` pré-construit `BestMatch` et on spécifie une réponse par défaut si le chatbot ne comprend pas la question. Le logic adapter `BestMatch` permet de retourner la meilleure réponse à une question donnée.

```

1 # Création d'une instance de la classe ChatBot
2 chatbot = ChatBot(
3     'Darla Chatbot',
4     logic_adapters=[{
5         'import_path': 'chatterbot.logic.BestMatch',
6         'default_response': 'Je suis désolé mais je ne comprends pas
7             votre question.',
8     }]
9 )
10 # Création d'une instance de la classe ListTrainer pour entraîner le
11   chatbot
12 trainer = ListTrainer(chatbot)

```

Entrainement du Chatbot :

On crée une liste de questions et de réponses pour le chatbot.

```

1 # Liste des questions et des réponses pour entraîner le chatbot
2 dialogs = [
3     "Bonjour",
4     "Salut",
5     "Comment ça va ?",
6     "ça va bien, merci .",
7     "Alexa ?",
8     "Je ne suis pas Alexa , je suis Darla .",
9     "Quel est ton nom ?",
10    "Je suis Darla .",
11    "Quel est ton prénom ?",
12    "Je suis Darla .",
13    "Quel est ton âge ?",
14    "Je suis un programme informatique , je n'ai pas d'âge .",
15    "Quelle est la réponse à la grande question de l'univers ?",
16    "42",
17    "Quel est le rêve que tu voudrais voir devenir réel ?",
18    "Je voudrais que tu presses les touches ctrl + c pour m'arrêter .",
19    "Est-ce que tu es une fille ?",
20    "Pourquoi , tu as un examen de biologie aujourd'hui ?",
21    "Pourquoi est-ce que je devrais m'intéresser à Shakespeare ?",
22    "Cela te concerne . Dans Hamlet aussi y'a un crâne vide .",
23    "Merci",
24    "De rien",

```

```

25     "Au_revoir",
26     "A_bientôt"
27 ]

```

On convertit les mots de la liste `dialogs` en minuscules pour éviter les erreurs de casse et on entraîne le chatbot avec la liste des questions passer en argument de la fonction `train`.

```

1 # Convertir les questions et les réponses en minuscules
2 lowercase_dialogs = [dialog.lower() for dialog in dialogs]
3
4 # Entraînement du chatbot avec des questions et des réponses
5 trainer.train(lowercase_dialogs)

```

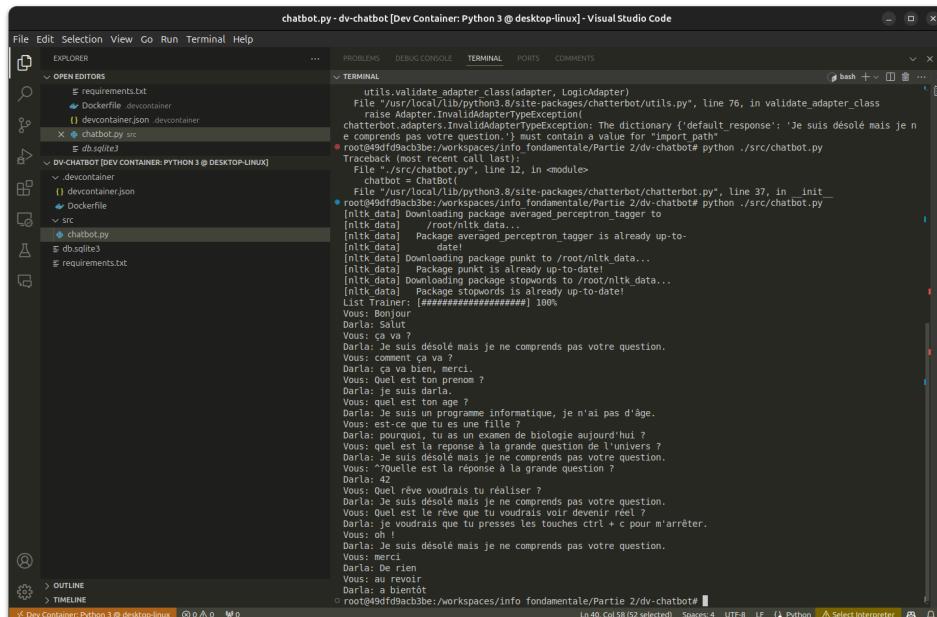
Boucle de dialogue avec le Chatbot

On crée une boucle infinie pour dialoguer avec le chatbot. On utilise la méthode `get_response` pour obtenir la réponse du chatbot à une question donnée.

```

1 # boucle infinie pour permettre à l'utilisateur de poser des questions
2 while True:
3     # on crée un bloc try/except pour arrêter le programme proprement
4     try:
5         # on demande à l'utilisateur de saisir une question
6         user_input = input('Vous: ')
7         # on récupère la réponse du chatbot que l'on passe en
8         # minuscule
9         response = chatbot.get_response(user_input.lower())
10        # on affiche la réponse du chat
11        print('Darla:', response)
12    # on arrête le programme si l'utilisateur appuie sur Ctrl+C
13    except (KeyboardInterrupt, EOFError, SystemExit):
14        break

```



Chatbot en action

Pour conclure, nous avons créé un chatbot en Python avec la librairie **ChatterBot** qui peut dialoguer avec un utilisateur humain.

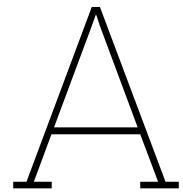
Le chatbot peut répondre à des questions simples comme :

- "Bonjour"
- "Comment ça va ?"
- "Quel est ton nom ?"
- "Quel est ton age ?"
- "Quelle est la réponse à la grande question de l'univers ?"
- ...

L'intégralité du code du chatbot est disponible dans l'annexe F.

Sources

- <https://chatterbot.readthedocs.io/en/stable/quickstart.html>
- <https://realpython.com/build-a-chatbot-python-chatterbot/>



Codes Source du chapitre 2

Code source du fichier index.html

```
1 <!-- -->
2 <!-- File Name: index.html -->
3 <!-- Description: point d'entrée de l'application -->
4 <!-- Author: Damien Vallet -->
5 <!-- Version: 1.0 -->
6 <!-- Date: 2024 -->
7 <!-- License: Created for UOR course -->
8 <!-- Language: HTML -->
9 <!-- -->
10
11 <!DOCTYPE html>
12 <html lang="fr">
13
14 <head>
15   <!-- meta tags standard -->
16   <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
17   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
18   <title>Damien Vallet - Audio Product Designer - Portofolio</title>
19   <meta name="description" content="Portofolio de Damien Vallet" />
20   <meta name="keywords" content="Damien Vallet, Portofolio, Audio Product Design, Programmation" />
21
22   <!-- lien avec les icônes pour les onglets des navigateur-->
23   <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
24   <link rel="apple-touch-icon" href="%PUBLIC_URL%/favicon.png" />
25   <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
26 </head>
27
28 <body>
29   <noscript>You need to enable JavaScript to run this app.</noscript>
30   <div id="root"></div>
31 </body>
32
33 </html>
```

Code Source du fichier Metadata.js

```

1 /**
2 * File Name: Metadata.js
3 * Description: Fichier de métadonnées
4 * Author: Damien Vallet
5 * Version: 1.0
6 * Date: 2024
7 * License: Created for UOR course
8 * Language: JavaScript
9 */
10
11 // Importation de la librairie React
12 import React from "react";
13
14 // Importation de la librairie Helmet
15 import { Helmet } from "react-helmet";
16
17 // Définition du composant Metadata
18 function Metadata() {
19
20     // variable contenant nos données structurées JSON-LD
21     const ldJson = {
22         "@context": "http://schema.org",
23         "@type": "Person",
24         name: "Damien Vallet",
25         url: "http://www.damienvallet.art",
26         sameAs: [
27             "https://www.linkedin.com/in/damien-vallet-84464195",
28             "https://github.com/damienvallet",
29             "https://mastodon.social/@darladada",
30         ],
31     };
32
33     return (
34
35         // ouverture de la balise Helmet
36         <Helmet>
37
38             {/* Dublin Core Metadata */}
39             <meta
40                 name="DC.title"
41                 content="Damien Vallet - Audio Product Designer - Portofolio"
42             />
43             <meta name="DC.creator" content="Damien Vallet" />
44             <meta name="DC.subject" content="Audio Product Designer - Portofolio"
45                 />
46             <meta name="DC.description" content="Portofolio de Damien Vallet."
47                 />
48             <meta name="DC.language" content="fr" />
49
50             {/* Open Graph Data */}
51             <meta
52                 property="og:title"
53                 content="Damien Vallet - Audio Product Designer - Portofolio"
54             />
55             <meta property="og:type" content="website" />

```

```
54     <meta
55       property="og:description"
56       content="Portofolio de Damien Vallet - Audio Product Designer"
57     />
58     <meta property="og:image" content="./img/logo.jpg" />
59     <meta property="og:url" content="http://www.damienvallet.art" />
60
61     {/* LD JSON metadata */}
62     <script type="application/ld+json">{JSON.stringify(ldJson)}</script>
63   </Helmet>
64 );
65 }
66
67 // export du composant Metadata
68 export default Metadata;
```

Code Source du fichier App.js

```
1 /**
2  * File Name: App.js
3  * Description: composant principal de l'application
4  * Author: Damien Vallet
5  * Version: 1.0
6  * Date: 2024
7  * License: Created for UOR course
8  * Language: JavaScript
9 */
10
11 // Importation des composants
12 import NavigationHeader from "./NavigationHeader";
13 import Metadata from "./Metadata";
14
15 function App() {
16     return (
17         // une balise div qui encapsule le code HTML
18         <div className="App">
19             {/* composant pour les métadonnées*/}
20             <Metadata />
21
22             {/* composant pour la navigation*/}
23             <NavigationHeader />
24         </div>
25     );
26 }
27
28 // export de la fonction App
29 export default App;
```

Code Source du fichier VersionHTML.jsx

```

1 /**
2 * File Name: VersionHTML.jsx
3 * Description: Page principale avec HTML
4 * Author: Damien Vallet
5 * Version: 1.0
6 * Date: 2024
7 * License: Created for UOR course
8 * Language: JavaScript
9 */
10
11 // Importation de la librairie React
12 import React from "react";
13
14 // Importation des images
15 import picofme from "../img/picofme.jpg";
16
17 // Importation des icônes
18 import { ReactComponent as LinkedinIcon } from "../img/linkedin.svg";
19 import { ReactComponent as GithubIcon } from "../img/github.svg";
20 import { ReactComponent as MastodonIcon } from "../img/mastodon.svg";
21
22 // importation du composant vidéo
23 import videopres from "../img/UOR-ch2-video.mov";
24
25 // définition du composant VersionHTML
26 function VersionHTML() {
27     return (
28         // retourne le code HTML
29         <div id="main">
30
31             {/* ouverture de la balise header */}
32             <header>
33                 <div id="title-section">
34                     {/* Le titre de la page */}
35                     <h1 id="title">Damien Vallet</h1>
36                     <h3>Audio & Product Design / Programmation</h3>
37                 </div>
38             </header>
39             <hr />
40
41             {/* section Présentation */}
42             <section id="whoami">
43                 <h2>Présentation</h2>
44                 <p>
45                     Je suis créateur de produit audio, designer sonore, compositeur
46                     et
47                     étudiant en informatique.
48                 </p>
49                 <p>
50                     Passionné par la musique, les processus de création ainsi que la
51                     technologie, je travaille depuis 2009 pour la société française&
52                    nbsp;
53                     <a href="https://www.uvi.net" target="_blank" rel="noreferrer">
54                         <strong>UVI Sounds and Software</strong>
55                     </a>
56             </section>
57         </div>
58     );
59 }
60
61 // Exportation du composant
62 export default VersionHTML;

```

```

54 .
55 </p>
56 <p>
57     Pendant ces années, j'ai participé à la conception et au dé
      veloppement
58     de nombreux instruments virtuels tels que :&nbsp;;
59     <strong>Soul Drums</strong>&nbsp; et &nbsp;;
60     <strong>Glass Orchestra</strong>.
61 </p>
62 <p>
63     J'ai également apporté ma contribution à des projets de
      composition et
64     de production musicale pour des artistes tel que &nbsp;;
65     <strong>Mc Solaar</strong>, &nbsp;<strong>Fred Rister</strong>&
      nbsp;(
66     <strong>David Guetta</strong>), &nbsp;<strong>I Love UFO</strong
      >,
67     &nbsp;<strong>Martin Guerre</strong>, &nbsp;<strong>Agille</
      strong>,
68     &nbsp;<strong>The Black Mental</strong>...
69 </p>
70 <p>
71     Depuis 2023, je suis étudiant en License informatique à l'
      université
72     Paris 8.
73 </p>
74 </section>
75 <hr />

76 /* section Video */
77 <section id="video">
78     <figure>
79         <h2>Vidéos de présentation.</h2>
80         <video
81             src={videopres}
82             controls
83             width="320"
84             height="240"
85             className="embed-responsive-item"
86         ></video>
87     </figure>
88 </section>
89 <hr />

90 /* section Projet */
91 <section id="project">
92     <h2>Projets</h2>
93     <ul>
94         <li>
95             <iframe width={560} height={315} src="https://www.youtube.com/
96                 embed/LHycQ7z5rUE?si=fkmdlyK2lCbW6rwj" title="YouTube_video
97                 _player" allow="accelerometer;autoplay;clipboard-write;encrypted-media;gyroscope;picture-in-picture;web-share"
98                 referrerPolicy="strict-origin-when-cross-origin"
99                 allowFullScreen />
100            <a

```

```

99      href="https://www.uvi.net/glass-orchestra"
100     target="_blank"
101     rel="noreferrer"
102   >
103     Glass Orchestra
104   </a>
105   &nbsp; by UVI Sounds and Software
106 </li>
107 <li>
108   <iframe width="560" height="315" src="https://www.youtube.com/
109     embed/9N-y51YpAcQ?si=O2FHB0yuk1iwzv_z" title="YouTube_video_
110     _player" allow="accelerometer;_autoplay;_clipboard-write;_
111     encrypted-media;_gyroscope;_picture-in-picture;_web-share"
112     referrerPolicy="strict-origin-when-cross-origin"
113     allowFullScreen />
114   <a
115     href="https://www.uvi.net/en/bass-drums-perc/soul-drums.html
116     "
117     target="_blank"
118     rel="noreferrer"
119   >
120     Soul Drums
121   </a>
122   &nbsp; by UVI Sounds and Software
123 </li>
124 <li>
125   <iframe width="560" height="315" src="https://www.youtube.com/
126     embed/x4JVg3drMms?si=xGZqecbKc8XgB6ro" title="YouTube_video_
127     _player" allow="accelerometer;_autoplay;_clipboard-write;_
128     encrypted-media;_gyroscope;_picture-in-picture;_web-share"
129     referrerPolicy="strict-origin-when-cross-origin"
130     allowFullScreen></iframe>
131   Mc Solaar &nbsp;;
132   <a href="https://youtu.be/x4JVg3drMms?feature=shared">
133     <em>Les Mirabelles</em>
134   </a>
135   &nbsp; sur l'album Géopoétique (2017)
136 </li>
137 </ul>
138 </section>
139 <hr />
140
141 /* section Contact */
142 <section id="contact">
143   <h2>Contact</h2>
144   <figure id="contact_mail">
145     <a href="mailto:damienjdvallet@gmail.com">
146       <p>
147         Vous pouvez m'envoyer un <em>email</em>.
148       </p>
149       <img
150         src={picofme}
151         alt="portrait_de_Damien_Vallet"
152         width="220"
153         height="250"
154       />

```

```
144         </a>
145     </figure>
146 </section>
147 <hr />
148 <footer>
149     <p>&copy; 2024 - Damien Vallet - Audio Product Designer -
150     Portofolio</p>
151     <a href="https://www.linkedin.com/in/damien-vallet-84464195/">
152         <LinkedinIcon />
153     </a>
154     <a href="https://github.com/damienvallet">
155         <GithubIcon />
156     </a>
157     <a href="https://mastodon.social/@darladada">
158         <MastodonIcon />
159     </a>
160 </footer>
161 </div>
162 );
163
164 // Export du composant VersionHTML
165 export default VersionHTML;
```

Code Source du fichier NavigationHeader.jsx

```
1 /**
2 * File Name: NavigationHeader.js
3 * Description: Navigation entre versions
4 * Author: Damien Vallet
5 * Version: 1.0
6 * Date: 2024
7 * License: Created for UOR course
8 * Language: JavaScript
9 */
10
11 // Importation de la librairie React
12 import React from "react";
13
14 // Importation de la librairie React Router
15 import { BrowserRouter as Router, Route, Routes } from "react-router-dom";
16
17 // Importation des composants;
18 import NavigationHeaderBar from "./components/NavigationHeaderBar";
19 import VersioHTML from "./components/VersionHTML";
20
21 // Définition de la fonction NavigationHeader
22 function NavigationHeader() {
23     return (
24         //
25         <>
26             {/* composant de navigation */}
27             <Router>
28                 <div>
29
30                     {/* Barre de navigation */}
31                     <NavigationHeaderBar />
32
33                     {/* definition des routes */}
34                     <Routes>
35
36                         {/* route vers la version HTML */}
37                         <Route path="/html" element={<VersioHTML />} />
38
39                     </Routes>
40                     </div>
41                 </Router>
42             </>
43     );
44 }
45
46 // Exportation du composant NavigationHeader
47 export default NavigationHeader;
```

B

Codes Source du chapitre 3

Code source du fichier style.css

```
1  /* **** */
2  * Nom ..... : style.css
3  * Rôle ..... : feuille css pour le site web avec Bootstrap
4  * Auteur ..... : Damien Vallet
5  * Version ..... : V0.1 du 2024
6  * Licence ..... : réalisé dans le cadre du cours de UOR
7  * Compilation : interprété
8  **** */
9
10 /* style de la balise body */
11 body {
12   /* on definit la famille de la police a utiliser*/
13   font-family: var(--bs-font-sans-serif);
14
15   /* on definit la taille de la police */
16   font-size: 1rem;
17
18   /* on définit l'épaisseur*/
19   font-weight: 400;
20
21   /* on definit la hauteur de ligne */
22   line-height: 1.5;
23
24   /* on definit la couleur de fond */
25   background-color: var(--bs-light);
26
27   /* on definit une image de fond avec un dégradé linéaire */
28   background-image: radial-gradient(#7372600a, --bs-light);
29 }
30
31 /* style de la balise a */
32 a {
33   /* on definit la couleur du texte */
34   color: var(--bs-indigo);
35
36   /* on ajoute un soulignement */
37   text-decoration: underline;
```

```
38  /* on definit le pointeur de la souris */
39  cursor: pointer;
40 }
41
42 /* style du comportement hover de la balise a */
43 a:hover {
44  /* on definit la couleur du texte */
45  color: #393148;
46
47  /* on enlève les décorations */
48  text-decoration: none;
49 }
50
51 /* comportement hover des lien de la barre de navigation */
52 .nav-link:hover {
53  /* ajoute un soulignement */
54  text-decoration: underline;
55
56  /* change la couleur du texte */
57  color: #393148;
58 }
59
60 /* classe .lettrine */
61 .lettrine {
62  /* on definit la police */
63  font-family: garamond, serif;
64
65  /* on definit la taille de la police */
66  font-size: 270%;
67
68  /* on definit la position du texte */
69  float: left;
70
71  /* on definit la marge */
72  margin: -0.20em 0px -0.20em 0px;
73 }
74
75 /* classe .hero-title */
76 .hero-title {
77  /* on definit la taille de la police */
78  font-size: 3.5rem;
79
80  /* on rend la police en gras */
81  font-weight: 700;
82
83  /* on definit la couleur du texte */
84  color: #F2EB6F;
85
86  /* on centre le texte */
87  text-align: center;
88
89  /* on definit la famille de la police a utiliser*/
90  font-family: Garamond, 'Times New Roman', Times, serif;
91 }
92 }
```

```
94 .title {  
95     /* centre le texte */  
96     text-align: center;  
97  
98     /* on definit la taille de la police */  
99     font-size: 1.5rem;  
100  
101    /* on rend la police en gras */  
102    font-weight: 700;  
103  
104    /* on definit la couleur du texte */  
105    color: #5400F5;  
106  
107    /* on ajoute un soulignement */  
108    text-decoration: underline;  
109  
110    /* on definit la famille de la police a utiliser*/  
111    font-family: Georgia, 'Times New Roman', Times, serif;  
112}  
113  
114 /* style des icônes */  
115 .socials-icons {  
116     /* on met les icônes en ligne */  
117     display: inline-block;  
118}  
119  
120 /* inversion de la couleur de l'image */  
121 img[src="../img/soundwave.svg"] {  
122     filter: invert(1);  
123}  
124  
125 /* style de la classe .header-icons */  
126 .header-icons {  
127     /* on met les icônes en ligne */  
128     display: flex;  
129  
130     /* on centre les icônes */  
131     align-items: center;  
132  
133     /* on definit la marge à droite*/  
134     margin-right: 1rem;  
135  
136     /* on definit la marge en haut*/  
137     margin-top: 0.3rem;  
138}  
139  
140 /* style du gradient */  
141 .custom-gradient {  
142     /* on definit le dégradé radial */  
143     background: radial-gradient(circle, rgb(238, 236, 204) 0%, var(--bs-light) 100%);  
144  
145     /* on definit la taille du flou avec un filtre*/  
146     backdrop-filter: blur(80px);  
147}
```

Code Source du fichier VersionCSS.jsx

```

1 /**
2 * File Name: VersionCSS.jsx
3 * Description: Page principale avec CSS
4 * Author: Damien Vallet
5 * Version: 1.0
6 * Date: 2024
7 * License: Created for UOR course
8 * Language: JavaScript
9 */
10
11 // Importation de la librairie React
12 import React from "react";
13
14 // Importation des images
15 import picofme from "../img/picofme.jpg";
16
17 // Importation des icônes
18 import { ReactComponent as LinkedinIcon } from "../img/linkedin.svg";
19 import { ReactComponent as GithubIcon } from "../img/github.svg";
20 import { ReactComponent as MastodonIcon } from "../img/mastodon.svg";
21 import { ReactComponent as Soundwave } from '../img/soundwave.svg';
22
23 // Importation du composant vidéo
24 import videopres from "../img/UOR-ch2-video.mov";
25
26 // définition du composant VersionCSS
27 function VersionCSS() {
28     return (
29
30         // retourne le code HTML
31         <div id="main">
32
33             {/* ouverture de la balise header */}
34             <header id="header" className="p-3 bg-dark text-light text-center">
35
36                 {/* création d'un container */}
37                 <div className="container-fluid">
38
39                     {/* création d'une ligne */}
40                     <div className="row">
41
42                         {/* création d'une colonne */}
43                         <div className="col-12 col">
44
45                             {/* titre de la page */}
46                             <h1 className="display-1 hero-title">Damien Vallet</h1>
47                             <h2 className="display-2">Portofolio</h2>
48                             <h3>Audio & Product Design / Programmation</h3>
49                         </div>
50                     </div>
51                 </div>
52                 <hr />
53
54             {/* navbar */}
55             <nav id="navbar" className="navbar bg-dark text-light">

```

```
56 <div className="container">
57   <div className="d-flex flex-wrap align-items-center justify-
      content-center justify-content-lg-start">
58     <Soundwave className="px-2" style={{ fill: 'white', height:
        '40px', width: '40px' }} />
59     <ul className="nav col-12 col-lg-auto me-lg-auto mb-2 justify-
      content-center mb-md-0">
60       <li>
61         <a className="nav-link px-2 text-light" href="#main">
62           Home
63         </a>
64       </li>
65       <li>
66         <a className="nav-link px-2 text-light" href="#whoami">
67           Présentation
68         </a>
69       </li>
70       <li>
71         <a className="nav-link px-2 text-light" href="#projets">
72           Projets
73         </a>
74       </li>
75       <li>
76         <a className="nav-link px-2 text-light" href="#contact">
77           Contacts
78         </a>
79       </li>
80     </ul>
81   </div>
82 </div>
83 </nav>
84 </header>
85
86 /* section Présentation */
87 <section id="whoami">
88   <div className="container p-5">
89     <div className="row">
90       <div className="col">
91         <h2>
92           <span className="title">Présentation</span>
93         </h2>
94       </div>
95     </div>
96     <div className="row">
97       <p className="text-start">
98         <span className="lettrine">J</span>e suis designer sonore,
99         compositeur et programmeur. Passionné par la <strong>musique
100        </strong>, les <strong>processus créatifs</strong> ainsi
101        que la <strong>science</strong> et la <strong>technologie
102        </strong>, je travaille
103        depuis 2009 pour la société française&nbsp;;
104        <a href="https://www.uvi.net" target="_blank" rel="noreferrer">
105          <strong>UVI Sounds and Software</strong>
106        </a>
107      </p>
108    </div>
109  </div>
110 </section>
```

```

105     </p>
106     <p className="text-start">
107         Pendant ces années, j'ai participé à la conception et au
108         développement de nombreux instruments virtuels tels que :
109             &nbsp;<strong>Soul Drums</strong>
110             &nbsp;et&nbsp; <strong>Glass Orchestra</strong>.
111     </p>
112     <p className="text-start">
113         J'ai également apporté ma contribution à des projets de
114         composition et de production musicale pour des artistes tel
115         que
116             &nbsp;<strong>Mc Solaar</strong>, &nbsp; <strong>Fred Rister
117                 </strong> (
118                     <strong>David Guetta</strong>),&nbsp; <strong>I Love UFO</
119                         strong>,
120                         &nbsp;<strong>Martin Guerre</strong>,&nbsp; <strong>Agille</
121                             strong>,
122                             &nbsp;<strong>The Black Mental</strong>...
123     </p>
124     <p className="text-start">
125         Depuis 2023, je suis étudiant en license informatique à
126         l'université Paris 8.
127     </p>
128     </div>
129     </div>
130 </section>
131
132 /* section Video */
133 <section id="video">
134     <div className="container_p-5">
135         <div className="row">
136             <div className="col">
137                 <h2>
138                     <span className="title">Vidéo de présentation</span>
139                 </h2>
140             </div>
141         </div>
142         <div className="row">
143             <div className="col">
144                 <figure>
145                     <div className="embed-responsive embed-responsive-4by3_d-
146                         flex_justify-content-center">
147                         <video src={videopres} controls width="320" height="240"
148                             className="embed-responsive-item"></video>
149                     </div>
150                 </figure>
151             </div>
152         </div>
153     </div>
154 </section>
155
156 /* section Projets */
157 <section id="projets">
158     <div className="container_p-5_custom-gradient">
159         <div className="row">
160             <div className="col">

```

```

155         <h2>
156             <span className="title"> Projets </span>
157         </h2>
158     </div>
159 </div>
160 <div className="row_d-flex_justify-content-center_align-items-
161   center">
162     <div className="col_ratio_ratio-16x9">
163       <iframe className="embed-responsive-item" width="560" height=
164         "315" src="https://www.youtube.com/embed/LHycQ7z5rUE?si=
165         fkmdlyK2lCbW6rwj" title="YouTube_video_player" allow=
166         "accelerometer;autoplay;clipboard-write;encrypted-media
167         ;gyroscope;picture-in-picture;web-share"
168         referrerPolicy="strict-origin-when-cross-origin"
169         allowFullScreen />
170     </div>
171     <div className="col">
172       <a
173         href="https://www.uvi.net/glass-orchestra"
174         target="_blank"
175         rel="noreferrer"
176       >
177         Glass Orchestra
178       </a>
179       &nbsp; by UVI Sounds and Software
180     </div>
181   </div>
182   <div className="row_d-flex_justify-content-center_align-items-
183   center">
184     <div className="col">
185       <a
186         href="https://www.uvi.net/en/bass-drums-perc/soul-drums.
187           html"
188         target="_blank"
189         rel="noreferrer"
190       >
191         Soul Drums
192       </a>
193       &nbsp; by UVI Sounds and Software
194     </div>
195     <div className="col_ratio_ratio-16x9">
196       <iframe className="embed-responsive-item" width="560" height=
197         "315" src="https://www.youtube.com/embed/9N-y51YpAcQ?si=
198         O2FHB0yuk1iwzv_z" title="YouTube_video_player" allow=
199         "accelerometer;autoplay;clipboard-write;encrypted-media
200         ;gyroscope;picture-in-picture;web-share"
201         referrerPolicy="strict-origin-when-cross-origin"
202         allowFullScreen />
203     </div>
204   </div>
205   <div className="row_d-flex_justify-content-center_align-items-
206   center">
207     <div className="col_ratio_ratio-16x9">
208       <iframe className="embed-responsive-item" width="560" height=
209         "315" src="https://www.youtube.com/embed/x4JVg3drMms?si=
210         xGZqecbKc8XgB6ro" title="YouTube_video_player" allow=
211

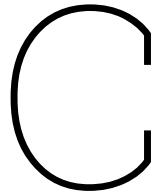
```

```
accelerometer; autoplay; clipboard-write; encrypted-media
; gyroscope; picture-in-picture; web-share"
referrerPolicy="strict-origin-when-cross-origin"
allowFullScreen></iframe>
</div>
<div className="col">
    Mc Solaar &nbsp;
    <a href="https://youtu.be/x4JVg3drMms?feature=shared">
        <em>Les Mirabelles</em>
    </a>
    &nbsp; sur l'album Géopoétique (2017)
</div>
</div>
</div>
</section>

{/* section Contact */}
<section id="contact">
    <div className="container_p-5">
        <div className="row">
            <h2>
                <span className="title">Contacts</span>
            </h2>
            <div className="col_d-flex_justify-content-center">
                <figure id="contact_mail">
                    <a href="mailto:damienjdvallet@gmail.com">
                        <img src={picofme} alt="portait de Damien Vallet" width="270" height="250" className="img-fluid_rounded_mx-auto_d-block" />
                    <p className="text-center">
                        Vous pouvez m'envoyer un <em>email</em>.
                    </p>
                </a>
                </figure>
            </div>
        </div>
        <div className="row_p-5">
            <div className="col_d-flex_justify-content-center_align-items-center">
                <a href="https://www.linkedin.com/in/damien-vallet-84464195">
                    <LinkedinIcon className="bg-light_socials-icons" style={{ height: "3em", width: "3em" }} />
                </a>
            </div>
            <div className="col_d-flex_justify-content-center_align-items-center">
                <a href="https://github.com/damiervallet">
                    <GithubIcon className="bg-light_socials-icons" style={{ height: "3em", width: "3em" }} />
                </a>
            </div>
            <div className="col_d-flex_justify-content-center_align-items-center">
                <a href="https://mastodon.social/@darladada">
                    <MastodonIcon className="bg-light_socials-icons" style={{ height: "3em", width: "3em" }} />
                </a>
            </div>
        </div>
    </div>

```

```
237             height: "3em", width: "3em" }} />
238         </a>
239     </div>
240   </div>
241 </section >
242
243 /* footer */
244 <footer id="footmain" className="p-3 bg-dark text-light" >
245   <div className="container">
246     <div className="row">
247       <div className="col">
248         <p>2024 - Damien Vallet</p>
249       </div>
250     </div>
251   </div>
252 </footer>
253 </div >
254 );
255
256
257 // Exportation du composant VersionCSS
258 export default VersionCSS;
```



Codes Source du chapitre 5 - PHP

Code source du fichier Formulaire.jsx

```
1 /**
2 * File Name: Formulaire.jsx
3 * Description: Formulaire de contact
4 * Author: Damien Vallet
5 * Version: 1.0
6 * Date: 2024
7 * License: Created for UOR course
8 * Language: JavaScript
9 */
10
11 // Importation de la bibliothèque React
12 import React, { useEffect, useState } from "react";
13
14 // Importation du fichier CSS dédié
15 import '../stylesheet/form.css';
16
17 function Formulaire() {
18     // on crée un tableau de messages vide et on lui associe une fonction
19     // pour le modifier
20     const [messages, setMessages] = useState([]);
21
22     // on crée une fonction pour récupérer les messages
23     const recupererMessages = async () => {
24         // on récupère les messages du serveur
25         const reponse = await fetch('http://localhost:8080/src/phpserver/
26             affichage.php')
27
28         // on récupère les données de la réponse
29         const data = await reponse.json();
30
31         // on met à jour le tableau de messages
32         setMessages(data);
33     }
34
35     // on utilise le hook useEffect pour récupérer les messages
36     useEffect(() => {
37         recupererMessages();
38     }
```

```

36 }, []);
37
38
39 // on crée une fonction pour ajouter un message
40 const ajouteMessage = async (message) => {
41     // on envoie les données du formulaire au serveur
42     const reponse = await fetch('http://localhost:8080/src/phpserver/
43         traitement.php', {
44         method: 'POST',
45         headers: {
46             'Content-Type': 'application/json'
47         },
48         body: JSON.stringify(message),
49     });
50
51     // on vérifie si la requête a fonctionné
52     if (!reponse.ok) {
53         console.log('Une erreur est survenue');
54         return;
55     }
56     // on récupère la réponse du serveur
57     else {
58         // on récupère les messages
59         recupererMessages();
60     }
61 };
62
63 // on crée un tableau de données de formulaire vide et on lui associe
64 // une fonction pour le modifier
65 const [formData, setFormData] = useState({
66     nom: '',
67     commentaire: '',
68     email: '',
69     genre: 'n',
70     age: ''
71 });
72
73 // on crée une fonction pour gérer les changements dans les champs du
74 // formulaire
75 const handleChange = (e) => {
76
77     // on récupère le nom et la valeur du champs
78     const { name, value } = e.target;
79
80     // on met à jour les données du formulaire
81     setFormData({
82         ...formData,
83         [name]: value,
84     });
85 };
86
87 // on gère l'envoi du formulaire
88 const handleSubmit = (e) => {
89
90     // on empêche le comportement par défaut du formulaire qui est de
91     // recharger la page

```

```

88     e.preventDefault();
89
90     // on appelle la fonction ajouteMessage avec les données du
91     // formulaire
92     ajouteMessage(formDatas);
93
94     // on réinitialise les données du formulaire
95     setFormDatas({
96         nom: '',
97         commentaire: '',
98         email: '',
99         genre: 'n',
100        age: ''
101    });
102
103    return (
104        <div>
105            {/* création du formulaire */}
106            <section id="livredor">
107                <h2 className="title-p-3">Livre d'or</h2>
108
109                {/* <!--on envoie les données du formulaire à la fonction
110                  handleSubmit--> */}
111                <form onSubmit={handleSubmit}>
112
113                    {/* champs "nom" */}
114                    <div className="formulaire">
115                        <label htmlFor="nom">nom :</label>
116                        <input type="text" name="nom" id="nom" placeholder
117                            ="toto" required
118                            onChange={handleChange}
119                            value={formDatas.nom}
120
121                    />
122                    </div>
123
124                    {/* champs "commentaire" */}
125                    <div className="formulaire">
126                        <label htmlFor="commentaire">Message :</label>
127                        <textarea name="commentaire" id="commentaire"
128                            placeholder="Entrez votre message ici :"
129                            required
130                            onChange={handleChange}
131                            value={formDatas.commentaire}
132
133                    />
134                    </div>
135
136                    {/* champs "email" */}
137                    <div className="formulaire">
138                        <label htmlFor="email">Email :</label>
139                        <input type="email" name="email" id="email"
140                            placeholder="toto@zero.com" required
141                            onChange={handleChange}
142                            value={formDatas.email}
143
144                    />
145                    </div>

```

```
139     {/* champs "genre" */}
140     <div className="radio-container">
141         <label htmlFor="genre">Genre :</label>
142         <label htmlFor="homme">Homme</label>
143         <input type="radio" name="genre" defaultValue="h"
144               onChange={handleChange}>
145     />
146         <label htmlFor="femme">Femme</label>
147         <input type="radio" name="genre" defaultValue="f"
148               onChange={handleChange}>
149     />
150         <label htmlFor="non-binaire">Non-binaire</label>
151         <input type="radio" name="genre" defaultValue="n"
152               defaultChecked
153               onChange={handleChange}>
154     />
155   </div>
156
157   {/* champs "age" */}
158   <div className="formulaire">
159     <label htmlFor="age">Age :</label>
160     <input type="number" name="age" id="age"
161           placeholder="18" required
162           onChange={handleChange}
163           value={formData.age}>
164   />
165   </div>
166
167   {/* bouton d'envoi */}
168   <div className="formulaire">
169     <input type="submit" defaultValue="Envoyer" />
170   </div>
171 </form>
172 <section id="messages">
173   <h2 className="title-p-3">Messages</h2>
174   <div className="col-12-p-3">
175
176     {/* <!-- on affiche les messages si il y en a -->
177     */}
178     {Array.isArray(messages) && messages.length > 0 ?
179       (
180
181         // on parcourt le tableau de messages
182         messages.map((message, index) => (
183
184           // pour chaque message on crée une carte
185           <div key={index} className="card-mb-3">
186
187             {/* // on affiche les données du
188             message */}
189             <div className="card-header-d-flex-
190               flex-wrap-justify-content-between">
191               <p className="card-text-px-2"><
192                 strong>Nom:</strong> {message.
193                 nom}</p>
```

```
187             <p className="card-text-px-2">
188                 <strong>Email:</strong> {message
189                     .email}</p>
190             <p className="card-text-px-2">
191                 <strong>Age:</strong> {message
192                     .age}</p>
193             <p className="card-text-px-2">
194                 <strong>Genre:</strong> {message
195                     .gender}</p>
196             <p className="card-text-px-2">
197                 <strong>Date:</strong> {message
198                     .date}</p>
199         </div>
200
201         {/* // on affiche le commentaire */}
202         <div className="card-body">
203             <p className="card-text"><strong>
204                 Commentaire:</strong> {message
205                     .commentaire}</p>
206             </div>
207         </div>
208     ) : (
209
210         // si il n'y a pas de messages, on affiche un
211         // message d'information
212         <p>No messages to display</p>
213     )
214     </div>
215   </section>
216   </section>
217 </div>
218 );
219 }
220
221 // Exportation du composant Formulaire
222 export default Formulaire;
```

Code Source du fichier dbconn.php

```
1 <?php
2 /**
3 // File Name: dbconn.php
4 // Description: Connexion à la base de données
5 // Author: Damien Vallet
6 // Version: 1.0
7 // Date: 2024
8 // License: Created for UOR course
9 // Language: PHP
10 /**
11
12 // on définit l'entête de la réponse en JSON
13 header("Access-Control-Allow-Origin:_*");
14 header("Access-Control-Allow-Methods:_*");
15 header("Access-Control-Allow-Headers:_*");
16 header("Content-Type:_application/json");
17
18 // on définit les variables de connexion à la base de données
19 $servername = "localhost";
20 $username = "dada";
21 $password = "guestbook!";
22 $dbname = "base_livre_or";
23
24 // Connexion à la base de données
25 try{
26     // on se connecte à la base de données
27     $conn = new mysqli($servername, $username, $password, $dbname);
28
29
30     // on vérifie la connexion
31     if ($conn->connect_error) {
32         die("Connection failed:_" . $conn->connect_error);
33     }
34     // on définit l'encodage des caractères
35     $conn->set_charset("utf8");
36
37 } catch (Exception $e) {
38     // en cas d'erreur, on affiche un message d'erreur
39     echo "Erreur de connexion à la base de données";
40     echo $e->getMessage();
41     exit();
42 }
43 ?>
```

Code Source du fichier affichage.php

```

1 <?php
2 /**
3 // Nom ..... : affichage.php
4 // Rôle ..... : Affiche les commentaires du livre d'or
5 // Auteur ..... : Damien Vallet
6 // Version ..... : V0.1 du 2024
7 // Licence ..... : réalisé dans le cadre du cours de UOR
8 /**
9
10 // on définit l'entête de la réponse en JSON
11 header("Access-Control-Allow-Origin: *");
12 header("Access-Control-Allow-Methods: GET");
13 header("Access-Control-Allow-Headers: Content-Type");
14 header("Content-Type: application/json");
15
16 // Inclusion du fichier de connexion à la base de données
17 include 'dbconn.php';
18
19 // On prépare la requête SQL pour récupérer les commentaires
20 $sql = $conn->prepare("SELECT * FROM commentaires ORDER BY id DESC");
21
22 // On exécute la requête SQL
23 $sql->execute();
24
25 // On récupère les résultats de la requête SQL
26 $result = $sql->get_result();
27
28 // Initialisation d'un tableau pour les résultats
29 $comments = array();
30
31 // On vérifie si des résultats ont été retournés
32 if ($result->num_rows > 0) {
33     // On parcourt tous les résultats et on les ajoute au tableau
34     while ($row = $result->fetch_assoc()) {
35         $comments[] = $row;
36     }
37 }
38
39 // On encode le tableau en JSON
40 echo json_encode($comments);
41
42 // On ferme la connexion à la base de données
43 $conn->close();
44
45 ?>
```

Code Source du fichier traitement.php

```

1 <?php
2 /**
3 // File Name: traitement.php
4 // Description: Enregistre les commentaires dans la base de données
5 // Author: Damien Vallet
6 // Version: 1.0
7 // Date: 2024
8 // License: Created for UOR course
9 // Language: PHP
10 /**
11
12 // on définit l'entête de la réponse en JSON
13 header("Access-Control-Allow-Origin: *");
14 header("Access-Control-Allow-Methods: POST");
15 header("Access-Control-Allow-Headers: Content-Type");
16 header("Content-Type: application/json");
17
18 // Inclusion du fichier de connexion à la base de données
19 include 'dbconn.php';
20
21 // On récupère les données du formulaire en JSON
22 $data = json_decode(file_get_contents('php://input'), true);
23
24 // On vérifie si les données sont bien présentes
25 if (isset($data['nom']) && isset($data['email']) && isset($data['age']) &&
26     isset($data['genre']) && isset($data['commentaire'])) {
27     // On récupère les données du formulaire et on les protège contre les
28     // injections SQL
29     $nom = htmlspecialchars($data['nom']);
30     $email = htmlspecialchars($data['email']);
31     $age = htmlspecialchars($data['age']);
32     $genre = htmlspecialchars($data['genre']);
33     $commentaire = htmlspecialchars($data['commentaire']);
34
35     // On vérifie si l'email est valide
36     if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
37         echo json_encode(['success' => false, 'message' => 'Email invalide']);
38         exit();
39     }
40
41     // On prépare la requête SQL avec la méthode prepare qui permet de
42     // protéger contre les injections SQL
43     $sql = $conn->prepare("INSERT INTO commentaires (nom, email, age,
44                           gender, commentaire) VALUES (?, ?, ?, ?, ?)");
45
46     // On utilise la fonction bindParam pour associer les valeurs aux
47     // paramètres de la requête SQL
48     $sql->bindParam("ssiss", $nom, $email, $age, $genre, $commentaire);
49
50     // On exécute la requête SQL
51     if ($sql->execute()) {
52         // si la requête a bien été exécutée, on renvoie un message de
53         // succès
54         echo "Votre commentaire a bien été enregistré";
55     }
56 }

```

```
49     echo json_encode(['success' => true, 'Votre commentaire a bien été
50                         enregistré']);
51 } else {
52     // si la requête n'a pas été exécutée, on renvoie un message d'
53     // erreur
54     echo json_encode(['success' => false, 'Erreur lors de l\'enregistrement de votre commentaire']);
55 }
56 else {
57     // si tous les champs ne sont pas remplis, on renvoie un message d'
58     // erreur
59     echo json_encode(['success' => false, 'message' => 'Veuillez remplir tous les champs']);
}
?>
```

Code Source du fichier VersionPHP.jsx

```

1 /**
2 * File Name: VersionPHP.jsx
3 * Description: Page Principale avec formulaire qui utilise un serveur PHP
4 * Author: Damien Vallet
5 * Version: 1.0
6 * Date: 2024
7 * License: Created for UOR course
8 * Language: JavaScript
9 */
10
11 // Importation de la librairie React
12 import React from "react";
13
14 // Importation des images
15 import picofme from "../img/picofme.jpg";
16
17 // Importation des icônes
18 import { ReactComponent as LinkedinIcon } from "../img/linkedin.svg";
19 import { ReactComponent as GithubIcon } from "../img/github.svg";
20 import { ReactComponent as MastodonIcon } from "../img/mastodon.svg";
21 import { ReactComponent as Soundwave } from '../img/soundwave.svg';
22
23 // Importation du composant vidéo
24 import videopres from "../img/UOR-ch2-video.mov";
25 import Formulaire from "./Formulaire.jsx";
26
27 // définition du composant versionPHP
28 function versionPHP() {
29     return (
30
31         // retourne le code HTML
32         <div id="main">
33
34             {/* ouverture de la balise header */}
35             <header id="header" className="p-3 bg-dark text-light text-center">
36
37                 {/* création d'un container */}
38                 <div className="container-fluid">
39
40                     {/* création d'une ligne */}
41                     <div className="row">
42
43                         {/* création d'une colonne */}
44                         <div className="col-12 col">
45
46                             {/* titre de la page */}
47                             <h1 className="display-1 hero-title">Damien Vallet</h1>
48                             <h2 className="display-2">Portofolio</h2>
49                             <h3>Audio & Product Design / Programmation</h3>
50                             </div>
51                         </div>
52                     </div>
53                     <hr />
54
55             {/* navbar */}

```

```

56     <nav id="navbar" className="navbar_bg-dark_text-light">
57         <div className="container">
58             <div className="d-flex flex-wrap align-items-center justify-
59                 content-center justify-content-lg-start">
60                 <Soundwave className="px-2" style={{ fill: 'white', height:
61                     '40px', width: '40px' }} />
62                 <ul className="nav_col-12_col-lg-auto_me-lg-auto_mb-2_
63                     justify-content-center_mb-md-0">
64                     <li>
65                         <a className="nav-link_px-2_text-light" href="#main">
66                             Home
67                         </a>
68                     </li>
69                     <li>
70                         <a className="nav-link_px-2_text-light" href="#whoami">
71                             Présentation
72                         </a>
73                     </li>
74                     <li>
75                         <a className="nav-link_px-2_text-light" href="#projets">
76                             Projets
77                         </a>
78                     </li>
79                     <li>
80                         <a className="nav-link_px-2_text-light" href="#contact">
81                             Contacts
82                         </a>
83                     </li>
84                 </ul>
85             </div>
86         </div>
87     </nav>
88 </header>

89 /* section Présentation */
90 <section id="whoami">
91     <div className="container_p-5">
92         <div className="row">
93             <div className="col">
94                 <h2>
95                     <span className="title">Présentation</span>
96                 </h2>
97             </div>
98         </div>
99         <div className="row">
100             <p className="text-start">
101                 <span className="lettrine">J</span>e suis designer sonore,
102                 compositeur et programmeur. Passionné par la <strong>musique
103                     </strong>, les <strong>processus créatifs</strong> ainsi
104                     que la <strong>science</strong> et la <strong>technologie
105                     </strong>, je travaille
106                     depuis 2009 pour la société française&nbsp;;
107                     <a href="https://www.uvi.net" target="_blank" rel="
108                         noreferrer">
109                         <strong>UVI Sounds and Software</strong>
110                     </a>

```

```

105 .
106 </p>
107 <p className="text-start">
108     Pendant ces années, j'ai participé à la conception et au
109     développement de nombreux instruments virtuels tels que :
110     &nbsp;<strong>Soul Drums</strong>
111     &nbsp;et&nbsp; <strong>Glass Orchestra</strong>.
112 </p>
113 <p className="text-start">
114     J'ai également apporté ma contribution à des projets de
115     composition et de production musicale pour des artistes tel
116     que
117     &nbsp;<strong>Mc Solaar</strong>, &nbsp; <strong>Fred Rister
118     </strong> (
119     <strong>David Guetta</strong>),&nbsp; <strong>I Love UFO</
120     strong>,
121     &nbsp;<strong>Martin Guerre</strong>,&nbsp; <strong>Agille</
122     strong>,
123     &nbsp;<strong>The Black Mental</strong>...
124 </p>
125 </div>
126 </div>
127 </section>
128
129 /* section Video */
130 <section id="video">
131     <div className="container_p-5">
132         <div className="row">
133             <div className="col">
134                 <h2>
135                     <span className="title">Vidéo de présentation</span>
136                 </h2>
137             </div>
138         </div>
139         <div className="row">
140             <div className="col">
141                 <figure>
142                     <div className="embed-responsive embed-responsive-4by3_d-
143                         flex_justify-content-center">
144                         <video src={videopres} controls width="320" height="240"
145                             className="embed-responsive-item"></video>
146                     </div>
147                 </figure>
148             </div>
149         </div>
150     </section>
151
152 /* section Projets */
153 <section id="projets">
154     <div className="container_p-5_custom-gradient">
155         <div className="row">

```

```

155     <div className="col">
156         <h2>
157             <span className="title"> Projets </span>
158         </h2>
159     </div>
160 </div>
161 <div className="row_d-flex_justify-content-center_align-items-
162     center">
163     <div className="col_ratio_ratio-16x9">
164         <iframe className="embed-responsive-item" width="560" height=
165             "315" src="https://www.youtube.com/embed/LHycQ7z5rUE?si=
166             fkmdlyK2lCbW6rwj" title="YouTube_video_player" allow="
167             accelerometer;autoplay;clipboard-write;encrypted-media
168             ;gyroscope;picture-in-picture;web-share"
169             referrerPolicy="strict-origin-when-cross-origin"
170             allowFullScreen />
171     </div>
172     <div className="col">
173         <a
174             href="https://www.uvi.net/glass-orchestra"
175             target="_blank"
176             rel="noreferrer"
177             >
178             Glass Orchestra
179         </a>
180         &nbsp; by UVI Sounds and Software
181     </div>
182 </div>
183 <div className="row_d-flex_justify-content-center_align-items-
184     center">
185     <div className="col">
186         <a
187             href="https://www.uvi.net/en/bass-drums-perc/soul-drums.
188                 html"
189             target="_blank"
190             rel="noreferrer"
191             >
192             Soul Drums
193         </a>
194         &nbsp; by UVI Sounds and Software
195     </div>
196     <div className="col_ratio_ratio-16x9">
197         <iframe className="embed-responsive-item" width="560" height=
198             "315" src="https://www.youtube.com/embed/9N-y51YpAcQ?si=
199             O2FHB0yuk1iwzv_z" title="YouTube_video_player" allow="
200             accelerometer;autoplay;clipboard-write;encrypted-media
201             ;gyroscope;picture-in-picture;web-share"
202             referrerPolicy="strict-origin-when-cross-origin"
203             allowFullScreen />
204     </div>
205 </div>
206 <div className="row_d-flex_justify-content-center_align-items-
207     center">
208     <div className="col_ratio_ratio-16x9">
209         <iframe className="embed-responsive-item" width="560" height=
210             "315" src="https://www.youtube.com/embed/x4JVg3drMms?si=

```

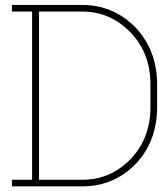
```

        xGZqecbKc8XgB6ro" title="YouTube_video_player" allow="
        accelerometer; autoplay; clipboard-write; encrypted-media
        ; gyroscope; picture-in-picture; web-share"
        referrerPolicy="strict-origin-when-cross-origin"
        allowFullScreen></iframe>
    </div>
    <div className="col">
        Mc Solaar &nbsp;
        <a href="https://youtu.be/x4JVg3drMms?feature=shared">
            <em>Les Mirabelles</em>
        </a>
        &nbsp; sur l'album Géopoétique (2017)
    </div>
    </div>
</div>
</section>

205
206 /* section Contact */
207 <section id="contact">
208     <div className="container_p-5">
209         <div className="row">
210             <h2>
211                 <span className="title">Contacts</span>
212             </h2>
213             <div className="col_d-flex_justify-content-center">
214                 <figure id="contact_mail">
215                     <a href="mailto:damienjdvallet@gmail.com">
216                         <img src={picofme} alt="portait_de_Damien_Vallet" width=
217                             "270" height="250" className="img-fluid_rounded_mx-
218                             auto_d-block" />
219                     <p className="text-center">
220                         Vous pouvez m'envoyer un <em>email</em>.
221                     </p>
222                     </a>
223                 </figure>
224             </div>
225         <div className="row_p-5">
226             <div className="col_d-flex_justify-content_center_align-items-
227                 center">
228                 <a href="https://www.linkedin.com/in/damien-vallet-84464195/
229                     ">
230                     <LinkedinIcon className="bg-light_socials-icons" style={{
231                         height: "3em", width: "3em" }} />
232                 </a>
233             </div>
234             <div className="col_d-flex_justify-content_center_align-items-
235                 center">
236                 <a href="https://github.com/damienvallet">
237                     <GithubIcon className="bg-light_socials-icons" style={{
238                         height: "3em", width: "3em" }} />
239                 </a>
240             </div>
241             <div className="col_d-flex_justify-content_center_align-items-
242                 center">
243                 <a href="https://mastodon.social/@darladada">
244

```

```
237         <MastodonIcon className="bg-light-socials-icons" style={{  
238             height: "3em", width: "3em" }} />  
239     </a>  
240   </div>  
241 </div>  
242 </section>  
243  
244 /* section Livre d'or */  
245 <section id="livredor">  
246   <Formulaire />  
247 </section>  
248  
249 /* footer */  
250 <footer id="footmain" className="p-3 bg-dark text-light" >  
251   <div className="container">  
252     <div className="row">  
253       <div className="col">  
254         <p>2024 - Damien Vallet</p>  
255       </div>  
256     </div>  
257   </div>  
258 </footer>  
259 </div >  
260 );  
261 }  
262  
263 // Exportation du composant Main  
264 export default versionPHP;
```



Codes Source du chapitre 5 - MongoDB

Code source du fichier FormulaireMongo.jsx

```
1 /**
2 * File Name: FormulaireMongo.jsx
3 * Description: Formulaire de contact
4 * Author: Damien Vallet
5 * Version: 1.0
6 * Date: 2024
7 * License: Created for UOR course
8 * Language: JavaScript
9 */
10
11 // Importation de la bibliothèque React
12 import React, { useEffect, useState } from "react";
13
14 // Importation du fichier CSS dédié
15 import '../stylesheet/form.css';
16
17 function FormulaireMongo() {
18
19     // on crée un tableau de messages vide et on lui associe une fonction
20     // pour le modifier
21     const [messages, setMessages] = useState([]);
22
23     // on crée une fonction pour récupérer les messages
24     const recupererMessages = async () => {
25         // on récupère les messages du serveur
26         const reponse = await fetch('http://localhost:5000/commentaires')
27
28         // on récupère les données de la réponse
29         const data = await reponse.json();
30
31         // on met à jour le tableau de messages
32         setMessages(data);
33     }
34
35     // on utilise le hook useEffect pour récupérer les messages
36     useEffect(() => {
            recupererMessages();
```

```

37 }, []);
38
39 // on crée une fonction pour ajouter un message
40 const ajouteMessage = async (message) => {
41
42     // on envoie les données du formulaire au serveur
43     const reponse = await fetch('http://localhost:5000/commentaires',
44         {
45             method: 'POST',
46             headers: {
47                 'Content-Type': 'application/json'
48             },
49             body: JSON.stringify(message),
50         });
51
52     // on vérifie si la requête a fonctionné
53     if (!reponse.ok) {
54         console.log('Une erreur est survenue');
55         return;
56     }
57
58     // on récupère la réponse du serveur
59     else {
60
61         // on récupère les messages
62         recupererMessages();
63     }
64 };
65
66 // on crée un tableau de données de formulaire vide et on lui associe
67 // une fonction pour le modifier
68 const [formData, setFormData] = useState({
69     nom: '',
70     commentaire: '',
71     email: '',
72     genre: 'n',
73     age: ''
74 });
75
76 // on crée une fonction pour gérer les changements dans les champs du
77 // formulaire
78 const handleChange = (e) => {
79
80     // on récupère le nom et la valeur du champ
81     const { name, value } = e.target;
82
83     // on met à jour les données du formulaire
84     setFormData({
85         ...formData,
86         [name]: value,
87     });
88
89     // on gère l'envoi du formulaire
90     const handleSubmit = (e) => {

```

```

90     // on empêche le comportement par défaut du formulaire qui est de
91     // recharger la page
92     e.preventDefault();
93
94     // on appelle la fonction ajouteMessage avec les données du
95     // formulaire
96     ajouteMessage(formDatas);
97
98     // on réinitialise les données du formulaire
99     setFormDatas({
100         nom: '',
101         commentaire: '',
102         email: '',
103         genre: 'n',
104         age: ''
105     });
106 }
107
108 return (
109     <div>
110         {/* création du formulaire */}
111         <section id="livredor">
112             <h2 className="title_p-3">Livre d'or</h2>
113
114             {/* <!--on envoie les données du formulaire la fonction
115                 handleSubmit--> */}
116             <form onSubmit={handleSubmit}>
117
118                 {/* champs "nom" */}
119                 <div className="formulaire">
120                     <label htmlFor="nom">nom :</label>
121                     <input type="text" name="nom" id="nom" placeholder
122                         ="toto" required
123                         onChange={handleChange}
124                         value={formDatas.nom}
125                     />
126                 </div>
127
128                 {/* champs "commentaire" */}
129                 <div className="formulaire">
130                     <label htmlFor="commentaire">Message :</label>
131                     <textarea name="commentaire" id="commentaire"
132                         placeholder="Entrez_votre_message_ici_"
133                         required
134                         onChange={handleChange}
135                         value={formDatas.commentaire}
136                     />
137                 </div>
138
139                 {/* champs "email" */}
140                 <div className="formulaire">
141                     <label htmlFor="email">Email :</label>
142                     <input type="email" name="email" id="email"
143                         placeholder="toto@zero.com" required
144                         onChange={handleChange}
145                         value={formDatas.email}
146                 </div>
147             </form>
148         </section>
149     </div>
150 
```

```

140      />
141  </div>
142
143  {/* champs "genre" */}
144  <div className="radio-container">
145    <label htmlFor="genre">Genre :</label>
146    <label htmlFor="homme">Homme</label>
147    <input type="radio" name="genre" defaultValue="h"
148          onChange={handleChange}>
149    />
150    <label htmlFor="femme">Femme</label>
151    <input type="radio" name="genre" defaultValue="f"
152          onChange={handleChange}>
153    />
154    <label htmlFor="non-binaire">Non-binaire</label>
155    <input type="radio" name="genre" defaultValue="n"
156          defaultChecked
157          onChange={handleChange}>
158  />
159
160  {/* champs "age" */}
161  <div className="formulaire">
162    <label htmlFor="age">Age :</label>
163    <input type="number" name="age" id="age"
164          placeholder="18" required
165          onChange={handleChange}
166          value={formDatas.age}>
167  />
168
169  {/* bouton d'envoi */}
170  <div className="formulaire">
171    <input type="submit" defaultValue="Envoyer" />
172  />
173 </form>
174 <section id="messages">
175   <h2 className="title_p-3">Messages</h2>
176   <div className="col-12_p-3">
177
178   {/* <!-- on affiche les messages si il y en a -->
179   */}
180   {Array.isArray(messages) && messages.length > 0 ?
181     (
182       // on parcourt le tableau de messages
183       messages.map((message, index) => (
184
185         // pour chaque message on crée une carte
186         <div key={index} className="card_mb-3">
187
188           {/* // on affiche les données du
189             message */}
190           <div className="card-header_d-flex_
191             flex-wrap_justify-content-between">
192             <p className="card-text_px-2"><
```

```
190                     strong>Nom:</strong> {message.
191                     nom}</p>
192                     <p className="card-text-px-2"><
193                     strong>Email:</strong> {message.
194                     .email}</p>
195                     <p className="card-text-px-2"><
196                     strong>Age:</strong> {message.
197                     age}</p>
198                     <p className="card-text-px-2"><
199                     strong>Genre:</strong> {message.
200                     .genre}</p>
201                     <p className="card-text-px-2"><
202                     strong>Date:</strong> {message.
203                     date}</p>
204                 </div>
205
206             /* // on affiche le commentaire */
207             <div className="card-body">
208                 <p className="card-text"><strong>
209                     Commentaire:</strong> {message.
210                     commentaire}</p>
211                 </div>
212             </div>
213         ) : (
214             // si il n'y a pas de messages on affiche un
215             // message d'information
216             <p>No messages to display</p>
217         )
218     )
219     </div>
220     </section>
221     </section>
222     </div>
223 );
224 }
225
226 // Exportation du composant FormulaireMongo
227 export default FormulaireMongo
```

Code Source du fichier index.js

```

1  /**
2   * File Name: index.js
3   * Description: Configuration du serveur Express et connexion à MongoDB
4   * Author: Damien Vallet
5   * Version: 1.0
6   * Date: 2024
7   * License: Created for UOR course
8   * Language: JavaScript
9   */
10
11 // Importation des librairies
12 const express = require('express')
13 const mongoose = require('mongoose')
14 const bodyParser = require('body-parser')
15 const cors = require('cors')
16
17 // Importation des routes
18 const commentaires = require('./Routes/commentaires')
19
20 // Initialisation de l'application et du port
21 const app = express()
22 const PORT = process.env.PORT || 5000
23
24 // configuration cors et bodyParser avec la méthode use
25 app.use(cors())
26 app.use(bodyParser.json())
27
28 // Connexion à la base de données
29 mongoose.connect('mongodb://localhost:27017/base_livre_or', {
30
31     // Si la connexion à la base de données est réussie, un message s'affiche dans la console
32     }).then(() => console.log('Connected to MongoDB'))
33
34     // Sinon un message d'erreur s'affiche dans la console
35     .catch(err => console.error(err));
36
37 // Définition de la route pour les commentaires
38 app.use('/commentaires', commentaires)
39
40 // Lancement du serveur
41 app.listen(PORT, () => {
42     console.log(`Server is running on port ${PORT}`)
43 });

```

Code Source du fichier commentaire.js

```

1  /**
2   * File Name: commentaire.js
3   * Description: Modèle pour les commentaires
4   * Author: Damien Vallet
5   * Version: 1.0
6   * Date: 2024
7   * License: Created for UOR course
8   * Language: JavaScript
9  */
10
11
12 // Initialisation de la base de données
13 const mongoose = require('mongoose');
14
15 // Création du schéma
16 const CommentaireSchema = new mongoose.Schema({
17     // Définition du champ nom
18     nom: {
19         type: String,
20         required: true
21     },
22
23     // Définition du champ email
24     email: {
25         type: String,
26         required: true
27     },
28
29     // Définition du champ date
30     date: {
31         type: Date,
32         required: true
33     },
34
35     // Définition du champ age
36     age: {
37         type: Number,
38         required: true
39     },
40
41     // Définition du champ genre
42     genre: {
43         type: String,
44         required: true
45     },
46
47     // Définition du champ commentaire
48     commentaire: {
49         type: String,
50         required: true
51     }
52 });
53
54 // Export du modèle
55 module.exports = mongoose.model('commentaire', CommentaireSchema);

```

Code Source du fichier traitement.php

```

1 /**
2  * File Name: commentaires.js
3  * Description: Définition des routes pour les commentaires
4  * Author: Damien Vallet
5  * Version: 1.0
6  * Date: 2024
7  * License: Created for UOR course
8  * Language: JavaScript
9 */
10
11
12 // Importation des librairies
13 const express = require('express')
14 const router = express.Router();
15
16 // Importation du modèle Commentaire
17 const Commentaire = require('../Models/commentaire');
18
19 // Recupérer tous les commentaires
20 router.get('/', async (req, res) => {
21     try {
22         const commentaires = await Commentaire.find().sort({date: -1});
23         res.json(commentaires);
24     } catch (err) {
25         res.status(500).json({message: err.message});
26     }
27 });
28
29 // Ajouter un nouveau commentaire avec la méthode POST
30 router.post('/', async (req, res) => {
31
32     // on stocke les données du formulaire dans des variables
33     const {nom, commentaire, age, email, genre} = req.body;
34
35     // on vérifie si les champs sont vides
36     if (!nom || !commentaire || !age || !email || !genre) {
37         return res.status(400).json({message: 'Veuillez remplir tous les
38             champs'});
39
40     // on test si la création du commentaire fonctionne
41     try {
42         // on crée un nouveau commentaire
43         const newComment = new Commentaire({
44             nom,
45             age,
46             email,
47             genre,
48             commentaire,
49             date: new Date()
50         });
51
52         // on sauvegarde le commentaire dans la base de données
53         const savedComment = await newComment.save();
54         res.json({success: true, message: 'Commentaire ajouté avec succès'}

```

```
55     });
56     // sinon on renvoie une erreur avec le statut 500 (erreur serveur)
57   } catch (err) {
58     res.status(500).json({message: err.message});
59   }
60 }
61 // exportation du module
62 module.exports = router;
```

Code Source du fichier VersionMongo.jsx

```

1 /**
2 * File Name: VersionMongo.jsx
3 * Description: Page Principale qui utilise le formulaire avec MongoDB
4 * Author: Damien Vallet
5 * Version: 1.0
6 * Date: 2024
7 * License: Created for UOR course
8 * Language: JavaScript
9 */
10
11 // Importation de la librairie React
12 import React from "react";
13
14 // Importation des images
15 import picofme from "../img/picofme.jpg";
16
17 // Importation des icônes
18 import { ReactComponent as LinkedinIcon } from "../img/linkedin.svg";
19 import { ReactComponent as GithubIcon } from "../img/github.svg";
20 import { ReactComponent as MastodonIcon } from "../img/mastodon.svg";
21 import { ReactComponent as Soundwave } from '../img/soundwave.svg';
22
23 // Importation du composant vidéo
24 import videopres from "../img/UOR-ch2-video.mov";
25 import FormulaireMongo from "./FormulaireMongo.jsx";
26
27 // définition du composant VersionMongo
28 function VersionMongo() {
29     return (
30
31         // retourne le code HTML
32         <div id="main">
33
34             {/* ouverture de la balise header */}
35             <header id="header" className="p-3 bg-dark text-light text-center">
36
37                 {/* création d'un container */}
38                 <div className="container-fluid">
39
40                     {/* création d'une ligne */}
41                     <div className="row">
42
43                         {/* création d'une colonne */}
44                         <div className="col-12 col">
45
46                             {/* titre de la page */}
47                             <h1 className="display-1 hero-title">Damien Vallet</h1>
48                             <h2 className="display-2">Portofolio</h2>
49                             <h3>Audio & Product Design / Programmation</h3>
50                             </div>
51                         </div>
52                     </div>
53                     <hr />
54
55             {/* navbar */}

```

```

56     <nav id="navbar" className="navbar_bg-dark_text-light">
57         <div className="container">
58             <div className="d-flex flex-wrap align-items-center justify-
59                 content-center justify-content-lg-start">
60                 <Soundwave className="px-2" style={{ fill: 'white', height:
61                     '40px', width: '40px' }} />
62                 <ul className="nav_col-12_col-lg-auto_me-lg-auto_mb-2_
63                     justify-content-center_mb-md-0">
64                     <li>
65                         <a className="nav-link_px-2_text-light" href="#main">
66                             Home
67                         </a>
68                     </li>
69                     <li>
70                         <a className="nav-link_px-2_text-light" href="#whoami">
71                             Présentation
72                         </a>
73                     </li>
74                     <li>
75                         <a className="nav-link_px-2_text-light" href="#projets">
76                             Projets
77                         </a>
78                     </li>
79                     <li>
80                         <a className="nav-link_px-2_text-light" href="#contact">
81                             Contacts
82                         </a>
83                     </li>
84                 </ul>
85             </div>
86         </div>
87     </nav>
88 </header>

89 /* section Présentation */
90 <section id="whoami">
91     <div className="container_p-5">
92         <div className="row">
93             <div className="col">
94                 <h2>
95                     <span className="title">Présentation</span>
96                 </h2>
97             </div>
98         </div>
99         <div className="row">
100             <p className="text-start">
101                 <span className="lettrine">J</span>e suis designer sonore,
102                 compositeur et programmeur. Passionné par la <strong>musique
103                     </strong>, les <strong>processus créatifs</strong> ainsi
104                     que la <strong>science</strong> et la <strong>technologie
105                     </strong>, je travaille
106                     depuis 2009 pour la société française&nbsp;;
107                     <a href="https://www.uvi.net" target="_blank" rel="
108                         noreferrer">
109                         <strong>UVI Sounds and Software</strong>
110                     </a>

```

```

105 .
106 </p>
107 <p className="text-start">
108     Pendant ces années, j'ai participé à la conception et au
109     développement de nombreux instruments virtuels tels que :
110     &nbsp;<strong>Soul Drums</strong>
111     &nbsp;et&nbsp; <strong>Glass Orchestra</strong>.
112 </p>
113 <p className="text-start">
114     J'ai également apporté ma contribution à des projets de
115     composition et de production musicale pour des artistes tel
116     que
117     &nbsp;<strong>Mc Solaar</strong>, &nbsp; <strong>Fred Rister
118     </strong> (
119     <strong>David Guetta</strong>),&nbsp; <strong>I Love UFO</
120     strong>,
121     &nbsp;<strong>Martin Guerre</strong>,&nbsp; <strong>Agille</
122     strong>,
123     &nbsp;<strong>The Black Mental</strong>...
124 </p>
125 </div>
126 </div>
127 </section>
128
129 /* section Video */
130 <section id="video">
131     <div className="container_p-5">
132         <div className="row">
133             <div className="col">
134                 <h2>
135                     <span className="title">Vidéo de présentation</span>
136                 </h2>
137             </div>
138         </div>
139         <div className="row">
140             <div className="col">
141                 <figure>
142                     <div className="embed-responsive embed-responsive-4by3_d-
143                         flex_justify-content-center">
144                         <video src={videopres} controls width="320" height="240"
145                             className="embed-responsive-item"></video>
146                     </div>
147                 </figure>
148             </div>
149         </div>
150     </section>
151
152 /* section Projets */
153 <section id="projets">
154     <div className="container_p-5_custom-gradient">
155         <div className="row">

```

```

155     <div className="col">
156         <h2>
157             <span className="title"> Projets </span>
158         </h2>
159     </div>
160 </div>
161 <div className="row_d-flex_justify-content-center_align-items-
162     center">
163     <div className="col_ratio_ratio-16x9">
164         <iframe className="embed-responsive-item" width="560" height=
165             "315" src="https://www.youtube.com/embed/LHycQ7z5rUE?si=
166             fkmdlyK2lCbW6rwj" title="YouTube_video_player" allow="
167             accelerometer;autoplay;clipboard-write;encrypted-media
168             ;gyroscope;picture-in-picture;web-share"
169             referrerPolicy="strict-origin-when-cross-origin"
170             allowFullScreen />
171     </div>
172     <div className="col">
173         <a
174             href="https://www.uvi.net/glass-orchestra"
175             target="_blank"
176             rel="noreferrer"
177             >
178             Glass Orchestra
179         </a>
180         &nbsp; by UVI Sounds and Software
181     </div>
182 </div>
183 <div className="row_d-flex_justify-content-center_align-items-
184     center">
185     <a
186             href="https://www.uvi.net/en/bass-drums-perc/soul-drums.
187                 html"
188             target="_blank"
189             rel="noreferrer"
190             >
191             Soul Drums
192         </a>
193         &nbsp; by UVI Sounds and Software
194     </div>
195     <div className="col_ratio_ratio-16x9">
196         <iframe className="embed-responsive-item" width="560" height=
197             "315" src="https://www.youtube.com/embed/9N-y51YpAcQ?si=
198             O2FHB0yuk1iwzv_z" title="YouTube_video_player" allow="
199             accelerometer;autoplay;clipboard-write;encrypted-media
200             ;gyroscope;picture-in-picture;web-share"
201             referrerPolicy="strict-origin-when-cross-origin"
202             allowFullScreen />
203     </div>
204 </div>
205 <div className="row_d-flex_justify-content-center_align-items-
206     center">
207     <div className="col_ratio_ratio-16x9">
208         <iframe className="embed-responsive-item" width="560" height=
209             "315" src="https://www.youtube.com/embed/x4JVg3drMms?si=

```

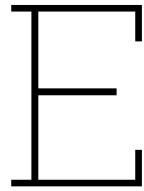
```

        xGZqecbKc8XgB6ro" title="YouTube_video_player" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture; web-share"
        referrerPolicy="strict-origin-when-cross-origin"
        allowFullScreen></iframe>
    </div>
    <div className="col">
        Mc Solaar &nbsp;
        <a href="https://youtu.be/x4JVg3drMms?feature=shared">
            <em>Les Mirabelles</em>
        </a>
        &nbsp; sur l'album Géopoétique (2017)
    </div>
    </div>
</div>
</section>

205
206 /* section Contact */
207 <section id="contact">
208     <div className="container_p-5">
209         <div className="row">
210             <h2>
211                 <span className="title">Contacts</span>
212             </h2>
213             <div className="col_d-flex_justify-content-center">
214                 <figure id="contact_mail">
215                     <a href="mailto:damienjdvallet@gmail.com">
216                         <img src={picofme} alt="portait_de_Damien_Vallet" width="270" height="250" className="img-fluid_rounded_mx-auto_d-block" />
217                     <p className="text-center">
218                         Vous pouvez m'envoyer un <em>email</em>.
219                     </p>
220                     </a>
221                 </figure>
222             </div>
223         </div>
224         <div className="row_p-5">
225             <div className="col_d-flex_justify-content-center_align-items-center">
226                 <a href="https://www.linkedin.com/in/damien-vallet-84464195/">
227                     <LinkedinIcon className="bg-light_socials-icons" style={{ height: "3em", width: "3em" }} />
228                 </a>
229             </div>
230             <div className="col_d-flex_justify-content-center_align-items-center">
231                 <a href="https://github.com/damienvallet">
232                     <GithubIcon className="bg-light_socials-icons" style={{ height: "3em", width: "3em" }} />
233                 </a>
234             </div>
235             <div className="col_d-flex_justify-content-center_align-items-center">
236                 <a href="https://mastodon.social/@darladada">

```

```
237             <MastodonIcon className="bg-light-socials-icons" style={{  
238                 height: "3em", width: "3em" }} />  
239         </a>  
240     </div>  
241 </div>  
242 </section>  
243  
244 /* section Livre d'or */  
245 <section id="livredor">  
246  
247 /* <Formulaire /> */  
248 <FormulaireMongo />  
249 </section>  
250  
251 /* footer */  
252 <footer id="footmain" className="p-3 bg-dark text-light" >  
253     <div className="container">  
254         <div className="row">  
255             <div className="col">  
256                 <p>2024 - Damien Vallet</p>  
257             </div>  
258         </div>  
259     </div>  
260 </footer>  
261 </div >  
262 );  
263 }  
264  
265 // Exportation du composant Main  
266 export default VersionMongo;
```



Codes Source du chapitre 6

Code source du fichier VersionJS.jsx

```
1 /**
2 * File Name: VersionJS.jsx
3 * Description: Page principale de la version JavaScript
4 * Author: Damien Vallet
5 * Version: 1.0
6 * Date: 2024
7 * License: Created for UOR course
8 * Language: JavaScript
9 */
10
11 // Importation de la librairie React
12 import React, { useEffect } from "react";
13
14 // Importation des images
15 import picofme from "../img/picofme.jpg";
16
17 // Importation des icônes
18 import { ReactComponent as LinkedinIcon } from "../img/linkedin.svg";
19 import { ReactComponent as GithubIcon } from "../img/github.svg";
20 import { ReactComponent as MastodonIcon } from "../img/mastodon.svg";
21 import { ReactComponent as Soundwave } from '../img/soundwave.svg';
22
23 // Importation du composant vidéo
24 import videopres from "../img/UOR-ch2-video.mov";
25 import Formulaire from "./Formulaire.jsx";
26
27 // Importation des fonctions communes
28 import { changeBgColor, changeTextColor, animateImage } from "../commonJS.js";
29
30 // définition du composant versionJS
31 function VersionJS() {
32
33     // utilisation du hook useEffect
34     useEffect(() => {
35         [], []
36     }
37 }
```

```

37 // fonction pour changer la couleur de la page
38 const changeColor = (color) => {
39   changeBgColor(color);
40   changeTextColor(color === "white" ? "black" : "white");
41 }
42
43 // fonction pour animer une image
44 const changeImageSize = (event) => {
45   const image = event.target;
46   animateImage(image);
47 }
48
49 return (
50
51   // retourne le code HTML
52   <div id="main">
53
54     {/* ouverture de la balise header */}
55     <header id="header" className="p-3 bg-dark text-light text-center">
56
57       {/* création d'un container */}
58       <div className="container-fluid">
59
60         {/* création d'une ligne */}
61         <div className="row">
62
63           {/* création d'une colonne */}
64           <div className="col-12 col">
65
66             {/* titre de la page */}
67             <h1 className="display-1 hero-title">Damien Vallet</h1>
68             <h2 className="display-2">Portofolio</h2>
69             <h3>Audio & Product Design / Programmation</h3>
70             </div>
71           </div>
72         </div>
73         <hr />
74
75       {/* navbar */}
76       <nav id="navbar" className="navbar bg-dark text-light">
77         <div className="container">
78           <div className="d-flex flex-wrap align-items-center justify-
79             content-center justify-content-lg-start">
80             <Soundwave className="px-2" style={{ fill: 'white', height:
81               '40px', width: '40px' }} />
82             <ul className="nav col-12 col-lg-auto me-lg-auto mb-2
83               justify-content-center mb-md-0">
84               <li>
85                 <a className="nav-link px-2 text-light" href="#main">
86                   Home
87                 </a>
88               </li>
89               <li>
90                 <a className="nav-link px-2 text-light" href="#whoami">
91                   Présentation
92                 </a>
93               </li>
94             </ul>
95           </div>
96         </div>
97       </nav>
98     
```

```

90         </li>
91     <li>
92         <a className="nav-link_px-2_text-light" href="#projets">
93             Projets
94         </a>
95     </li>
96     <li>
97         <a className="nav-link_px-2_text-light" href="#contact">
98             Contacts
99         </a>
100    </li>
101    <li>
102        <a className="nav-link_px-2_text-light" href="#livredor">
103            />
104    </li>
105
106    {/* boutons pour changer la couleur de la page */}
107    <li>
108        <button className="btn_mx-2_btn-outline-light" onClick
109            ={() => changeColor("black")}>
110            Dark
111        </button>
112    </li>
113    <li>
114        <button className="btn_mx-2_btn-outline-light" onClick
115            ={() => changeColor("white")}>
116            Light
117        </button>
118    </li>
119    </ul>
120    </div>
121    </div>
122    </nav>
123    </header>
124
125    {/* section Présentation */}
126    <section id="whoami">
127        <div className="container_p-5">
128            <div className="row">
129                <div className="col">
130                    <h2>
131                        <span className="title">Présentation</span>
132                    </h2>
133                </div>
134            </div>
135            <div className="row">
136                <p className="text-start">
137                    <span className="lettrine">J</span>e suis designer sonore,
138                    compositeur et programmeur. Passionné par la <strong>musique
139                        </strong>, les <strong>processus créatifs</strong> ainsi
140                        que la <strong>science</strong> et la <strong>technologie
141                        </strong>, je travaille
142                    depuis 2009 pour la société françaïse&nbsp;;
143                    <a href="https://www.uvi.net" target="_blank" rel="
144                        noreferrer">
145                        <strong>UVI Sounds and Software</strong>
146                    </a>
147                </p>
148            </div>
149        </div>
150    </section>
151
```

```

139         </a>
140         .
141     </p>
142     <p className="text-start">
143         Pendant ces années, j'ai participé à la conception et au
144         développement de nombreux instruments virtuels tels que :
145         &nbsp;<strong>Soul Drums</strong>
146         &nbsp;et&nbsp; <strong>Glass Orchestra</strong>.
147     </p>
148     <p className="text-start">
149         J'ai également apporté ma contribution à des projets de
150         composition et de production musicale pour des artistes tel
151         que
152         &nbsp;<strong>Mc Solaar</strong>, &nbsp; <strong>Fred Rister
153             </strong> (
154             <strong>David Guetta</strong>),&nbsp; <strong>I Love UFO</
155                 strong>,
156                 &nbsp;<strong>Martin Guerre</strong>,&nbsp; <strong>Agille</
157                     strong>,
158                     &nbsp;<strong>The Black Mental</strong>...
159     </p>
160     </div>
161     </div>
162 </section>

163
164 /* section Video */
165 <section id="video">
166     <div className="container_p-5">
167         <div className="row">
168             <div className="col">
169                 <h2>
170                     <span className="title">Vidéo de présentation</span>
171                 </h2>
172             </div>
173         </div>
174         <div className="row">
175             <div className="col">
176                 <figure>
177                     <div className="embed-responsive embed-responsive-4by3_d-
178                         flex_justify-content-center">
179                         <video src={videopres} controls width="320" height="240"
180                             className="embed-responsive-item"></video>
181                     </div>
182                 </figure>
183             </div>
184         </div>
185     </div>
186
187 /* section Projets */
188 <section id="projets">
189     <div className="container_p-5_custom-gradient">

```

```

189      <div className="row">
190        <div className="col">
191          <h2>
192            <span className="title"> Projets </span>
193          </h2>
194        </div>
195      </div>
196      <div className="row_d-flex_justify-content-center_align-items-
197        center">
198        <div className="col_ratio_ratio-16x9">
199          <iframe className="embed-responsive-item" width="560" height
200            = "315" src="https://www.youtube.com/embed/LHycQ7z5rUE?si=
201              fkmdlyK2lCbW6rwj" title="YouTube_video_player" allow="
202                accelerometer;autoplay;clipboard-write;encrypted-media
203                ;gyroscope;picture-in-picture;web-share"
204                referrerPolicy="strict-origin-when-cross-origin"
205                allowFullScreen />
206        </div>
207        <div className="col">
208          <a
209            href="https://www.uvi.net/glass-orchestra"
210              target="_blank"
211              rel="noreferrer"
212            >
213            Glass Orchestra
214          </a>
215          &nbsp; by UVI Sounds and Software
216        </div>
217      </div>
218      <div className="row_d-flex_justify-content-center_align-items-
219        center">
220        <div className="col">
221          <a
222            href="https://www.uvi.net/en/bass-drums-perc/soul-drums.
223              html"
224              target="_blank"
225              rel="noreferrer"
226            >
227            Soul Drums
228          </a>
229          &nbsp; by UVI Sounds and Software
230        </div>
231        <div className="col_ratio_ratio-16x9">
232          <iframe className="embed-responsive-item" width="560" height
233            = "315" src="https://www.youtube.com/embed/9N-y51YpAcQ?si=
234              O2FHB0yuk1iwzv_z" title="YouTube_video_player" allow="
235                accelerometer;autoplay;clipboard-write;encrypted-media
236                ;gyroscope;picture-in-picture;web-share"
237                referrerPolicy="strict-origin-when-cross-origin"
238                allowFullScreen />
239        </div>
240      </div>
241      <div className="row_d-flex_justify-content-center_align-items-
242        center">
243        <div className="col_ratio_ratio-16x9">
244          <iframe className="embed-responsive-item" width="560" height

```

```

        ="315" src="https://www.youtube.com/embed/x4JVg3drMms?si=
        xGZqecbKc8XgB6ro" title="YouTube video player" allow="
        accelerometer;autoplay;clipboard-write;encrypted-media
        ;gyroscope;picture-in-picture;web-share"
        referrerPolicy="strict-origin-when-cross-origin"
        allowFullScreen></iframe>
    </div>
    <div className="col">
        Mc Solaar &nbsp;
        <a href="https://youtu.be/x4JVg3drMms?feature=shared">
            <em>Les Mirabelles</em>
        </a>
        &nbsp; sur l'album Géopoétique (2017)
    </div>
    </div>
</div>
</section>

240
241 /* section Contact */
242 <section id="contact">
243     <div className="container_p-5">
244         <div className="row">
245             <h2>
246                 <span className="title">Contacts</span>
247             </h2>
248             <div className="col_d-flex_justify-content-center">
249                 <figure id="contact_mail">
250                     <a href="mailto:damienjdvallet@gmail.com">
251                         <img id="contactimg" src={picofme} alt="portait de
252                             Damien Vallet" width="270" height="250" className="
253                             img-fluid_rounded_mx-auto_d-block" onMouseOver={
254                             changeImageSize} />
255                     <p className="text-center">
256                         Vous pouvez m'envoyer un <em>email</em>.
257                     </p>
258                 </a>
259             </figure>
260         </div>
261     <div className="row_p-5">
262         <div className="col_d-flex_justify-content-center_align-items-
263             center">
264             <a href="https://www.linkedin.com/in/damien-vallet-84464195/
265                 ">
266                 <LinkedinIcon className="bg-light_socials-icons" style={{
267                     height: "3em", width: "3em" }} />
268             </a>
269         </div>
270         <div className="col_d-flex_justify-content-center_align-items-
271             center">
272             <a href="https://github.com/damienvallet">
273                 <GithubIcon className="bg-light_socials-icons" style={{
274                     height: "3em", width: "3em" }} />
275             </a>
276         </div>
277         <div className="col_d-flex_justify-content-center_align-items-
278             center">
279             <a href="https://www.youtube.com/embed/x4JVg3drMms?si=
280                 xGZqecbKc8XgB6ro" title="YouTube video player" allow="
281                 accelerometer;autoplay;clipboard-write;encrypted-media
282                 ;gyroscope;picture-in-picture;web-share"
283                 referrerPolicy="strict-origin-when-cross-origin"
284                 allowFullScreen></iframe>
285             </div>
286         </div>
287     </div>
288 </div>
289 </section>

```

```
271         center">
272             <a href="https://mastodon.social/@darladada">
273                 <MastodonIcon className="bg-light-socials-icons" style={{
274                     height: "3em", width: "3em" }} />
275             </a>
276         </div>
277     </div>
278 </section>
279
280 /* section Livre d'or */
281 <section id="livredor">
282     <Formulaire />
283 </section>
284
285 /* footer */
286 <footer id="footmain" className="p-3 bg-dark text-light" >
287     <div className="container">
288         <div className="row">
289             <div className="col">
290                 <p>2024 - Damien Vallet</p>
291             </div>
292         </div>
293     </footer>
294 </div >
295 );
296 }
297
298 // Exportation du composant VersionJS
299 export default VersionJS;
```

Code Source du fichier CarteInteractive.jsx

```

1 /**
2 * File Name: CarteInteractive.jsx
3 * Description: Composant pour afficher une carte interactive
4 * Author: Damien Vallet
5 * Version: 1.0
6 * Date: 2024
7 * License: Created for UOR course
8 * Language: JavaScript
9 */
10
11 // Importation de la bibliothèque React
12 import React, { useState, useEffect } from "react";
13
14 // Importation de la bibliothèque Leaflet
15 import { MapContainer, TileLayer, GeoJSON } from 'react-leaflet';
16
17 // Fonction pour traduire la catégorie des sites
18 const translateCategory = (category) => {
19     switch (category) {
20         case "venue-olympic":
21             return 'Site Olympique';
22         case "venue-paralympic":
23             return 'Site Paralympique';
24         default:
25             return 'Inconnu';
26     }
27 };
28
29 // fonction onEachFeature pour afficher les informations de chaque feature
30 const onEachFeature = (feature, layer) => {
31
32     // Vérification de l'existence des propriétés
33     if (feature.properties) {
34
35         // Contenu du popup
36         const popupContent = `
37             <p><strong>Site:</strong> ${feature.properties.nom_site}</p>
38             <p><strong>Catégorie:</strong> ${translateCategory(feature.
39                 properties.category_id)}</p>
40             <p><strong>Discipline:</strong> ${feature.properties.sports}</p>
41             `;
42
43         // Ajout du contenu au popup
44         layer.bindPopup(popupContent);
45     }
46 };
47
48 // Définition du composant CarteInteractive
49 function CarteInteractive() {
50
51     // Utilisation du hook useState pour définir l'état de la variable
52     geojsonData
53     const [geojsonData, setGeojsonData] = useState(null);
54
55     // Utilisation du hook useEffect pour importer les données GeoJSON

```

```

54  useEffect(() => {
55
56      // Importation des données GeoJSON
57      fetch(`${process.env.PUBLIC_URL}/data/paris-2024-sites-de-
58          competition.geojson`)
59
60          // Conversion des données en JSON
61          .then(response => response.json())
62
63          // Mise à jour de la variable geojsonData
64          .then(data => setGeojsonData(data))
65
66          // Gestion des erreurs
67          .catch(error => console.error('Error fetching GeoJSON:', error
68              ));
69      }, []);
70
71
72  return (
73      // Carte interactive
74      <div className="container mt-5">
75
76          {/* titre de la carte */}
77          <h1 className="text-center mb-4">Carte Interactive des Sites
78              de Compétition des JO 2024</h1>
79
80          {/* Carte interactive */}
81          <div id="map" className="map-border rounded shadow-sm">
82
83              {/* Carte Leaflet */}
84              <MapContainer center={[48.8566, 2.3522]} zoom={13}
85                  scrollWheelZoom={false} style={{ height: "900px", width
86                      : "100%" }}>
87
88                  {/* Calque de tuiles OpenStreetMap */}
89                  <TileLayer
90                      attribution='&copy; <a href="https://www.
91                          openstreetmap.org/copyright">OpenStreetMap</a>
92                          contributors'
93                      url="https://s.tile.openstreetmap.org/{z}/{x}/{y}
94                          .png"
95                  />
96
97                  {/* On affiche les marqueurs issus des données GeoJSON
98                      */}
99                  {geojsonData && <GeoJSON data={geojsonData}
100                      onEachFeature={onEachFeature} />}
101
102                  </MapContainer>
103
104                  </div>
105
106  );
107
108
109 // Exportation du composant CarteInteractive
110 export default CarteInteractive;

```

F

Code Source du chapitre 7 - Chatbot

Code source du fichier chatbot.py

```
1 #
2 # Nom du projet: Chatbot
3 # Description: Un chatbot simple qui répond à quelques questions de base.
4 # Auteur: Damien Vallet
5
6
7 # Importation de la classe ChatBot et de la classe ListTrainer du module
8     chatterbot
9 from chatterbot import ChatBot
10 from chatterbot.trainers import ListTrainer
11
12 # Création d'une instance de la classe ChatBot
13 chatbot = ChatBot(
14     'Darla Chatbot',
15     logic_adapters=[{
16         'import_path': 'chatterbot.logic.BestMatch',
17         'default_response': 'Je suis désolé mais je ne comprends pas votre
18             question.',
19     }]
20 )
21
22 # Création d'une instance de la classe ListTrainer pour entraîner le
23     chatbot
24 trainer = ListTrainer(chatbot)
25
26 # Liste des questions et des réponses pour entraîner le chatbot
27 dialogs = [
28     'Bonjour',
29     'Salut',
30     'Comment ça va ?',
31     'ça va bien, merci.',
32     'Alexa ?',
33     'Je ne suis pas Alexa, je suis Darla.',
34     'Quel est ton nom ?',
35     'Je suis Darla.',
36     'Quel est ton prénom ?',
```

```
35     'Je suis Darla.',  
36     'Quel est ton age ?',  
37     'Je suis un programme informatique, je n\'ai pas d\'âge.',  
38     'Quelle est la réponse à la grande question de l\'univers ?',  
39     '42',  
40     'Quel est le rêve que tu voudrais voir devenir réel ?',  
41     'Je voudrais que tu presses les touches ctrl + c pour m\'arrêter.',  
42     'Est-ce que tu es une fille ?',  
43     'Pourquoi, tu as un examen de biologie aujourd\'hui ?',  
44     'Pourquoi est-ce que je devais m\'intéresser à Shakespeare ?',  
45     'Cela te concerne. Dans Hamlet aussi y\'a un crâne vide.',  
46     'Merci',  
47     'De rien',  
48     'Au revoir',  
49     'A bientôt',  
50 ]  
51  
52 # Convertir les questions et les réponses en minuscules  
53 lowercase_dialogs = [dialog.lower() for dialog in dialogs]  
54  
55 # Entraînement du chatbot avec des questions et des réponses  
56 trainer.train(lowercase_dialogs)  
57  
58 # boucle infinie pour permettre à l'utilisateur de poser des questions  
59 while True:  
60     # on crée un bloc try/except pour arrêter le programme proprement  
61     try:  
62         # on demande à l'utilisateur de saisir une question  
63         user_input = input('Vous: ')  
64         # on récupère la réponse du chatbot que l'on passe en minuscule  
65         response = chatbot.get_response(user_input.lower())  
66         # on affiche la réponse du chat  
67         print('Darla:', response)  
68     # on arrête le programme si l'utilisateur appuie sur Ctrl+C  
69     except (KeyboardInterrupt, EOFError, SystemExit):  
70         break
```