



DOSSIER PROFESSIONNEL (DP)

<i>Nom de naissance</i>	- Petit
<i>Nom d'usage</i>	- Petit
<i>Prénom</i>	- Antoine
<i>Adresse</i>	- 165 rue du Felibrige 83560 St Julien

Titre professionnel visé

Développeur Web & Web Mobile

MODALITÉ D'ACCÈS :

- Parcours de formation
- Validation des Acquis de l'Expérience (VAE)

DOSSIER PROFESSIONNEL (DP)

Présentation du dossier

Le dossier professionnel (DP) constitue un élément du système de validation du titre professionnel. **Ce titre est délivré par le Ministère chargé de l'emploi.**

Le DP appartient au candidat. Il le conserve, l'actualise durant son parcours et le présente **obligatoirement à chaque session d'examen.**

Pour rédiger le DP, le candidat peut être aidé par un formateur ou par un accompagnateur VAE.

Il est consulté par le jury au moment de la session d'examen.

Pour prendre sa décision, le jury dispose :

1. des résultats de la mise en situation professionnelle complétés, éventuellement, du questionnaire professionnel ou de l'entretien professionnel ou de l'entretien technique ou du questionnement à partir de productions.
2. du **Dossier Professionnel** (DP) dans lequel le candidat a consigné les preuves de sa pratique professionnelle
3. des résultats des évaluations passées en cours de formation lorsque le candidat évalué est issu d'un parcours de formation
4. de l'entretien final (dans le cadre de la session titre).

[Arrêté du 22 décembre 2015, relatif aux conditions de délivrance des titres professionnels du ministère chargé de l'Emploi]

Ce dossier comporte :

- ▶ pour chaque activité-type du titre visé, un à trois exemples de pratique professionnelle ;
- ▶ un tableau à renseigner si le candidat souhaite porter à la connaissance du jury la détention d'un titre, d'un diplôme, d'un certificat de qualification professionnelle (CQP) ou des attestations de formation ;
- ▶ une déclaration sur l'honneur à compléter et à signer ;
- ▶ des documents illustrant la pratique professionnelle du candidat (facultatif)
- ▶ des annexes, si nécessaire.

Pour compléter ce dossier, le candidat dispose d'un site web en accès libre sur le site.

DOSSIER PROFESSIONNEL (DP)

 <http://travail-emploi.gouv.fr/titres-professionnels>

DOSSIER PROFESSIONNEL (DP)

Sommaire

Exemples de pratique professionnelle

Développer la partie front-end d'une application web ou web mobile sécurisée p.6

- Installer et configurer son environnement de travail en fonction du projet web ou web mobile p. 6
- Maquetter des interfaces utilisateur web ou web mobile p. 41
- Réaliser des interfaces utilisateurs statiques web ou web mobile p. 56
- Développer la partie dynamique des interfaces utilisateur web ou web mobile p. 96 & 117

Développer la partie back-end d'une application web ou web mobile sécurisée p. 142

- Mettre en place une base de donnée relationnelle p. 142
- Développer des composants d'accès aux données SQL et NoSQL p. 152
- Développer des composants métier côté serveur p. 152
- Documenter le déploiement d'une application dynamique web ou web mobile p. 175

Titres, diplômes, CQP, attestations de formation (*facultatif*) p. 182

Déclaration sur l'honneur p. 183

Documents illustrant la pratique professionnelle (*facultatif*) p. 184

Annexes (*Si le RC le prévoit*) p. 185

DOSSIER PROFESSIONNEL (DP)

EXEMPLES DE PRATIQUE

PROFESSIONNELLE

DOSSIER PROFESSIONNEL (DP)

Activité-type

1

Développer la partie front-end d'une application web ou web mobile sécurisée

Exemple n°1 - Installer et configurer son environnement de travail en fonction du projet web ou web mobile

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

1. Installation et configuration de Visual Studio Code

Je télécharge Visual Studio Code (VSC) depuis <https://code.visualstudio.com/> et je suis toutes les étapes de l'installation.

DOSSIER PROFESSIONNEL (DP)

The screenshot shows the official Visual Studio Code website. At the top, there's a navigation bar with links for Visual Studio Code, Docs, Updates, Blog, API, Extensions, FAQ, GitHub Copilot, and a search bar labeled "Search Docs". A "Download" button is also present. Below the navigation, a banner invites users to "Join us for VS Code Live: Agent Mode Day on April 16th!". The main feature is a large, bold text: "Your code editor. Redefined with AI.". Below this are two buttons: "Download for Windows" and "Try agent mode". Smaller text below the buttons includes "Web, Insiders edition, or other platforms" and a note about accepting the license and privacy statement. A screenshot of the VS Code interface is shown at the bottom.

The screenshot shows the "Accord de licence" (License Agreement) step in the Visual Studio Code installation process. It features a title "Accord de licence", a note that "Les informations suivantes sont importantes. Veuillez les lire avant de continuer.", and a large text area containing the license terms. To the right is the Microsoft logo. At the bottom, there's a section titled "TERMES DU CONTRAT DE LICENCE LOGICIEL MICROSOFT" with two radio button options: "Je comprends et j'accepte les termes du contrat de licence" (selected) and "Je refuse les termes du contrat de licence".

- Je comprends et j'accepte les termes du contrat de licence
 Je refuse les termes du contrat de licence

Suivant

Annuler

DOSSIER PROFESSIONNEL (DP)

Installation - Microsoft Visual Studio Code (User)

Tâches supplémentaires

Quelles sont les tâches supplémentaires qui doivent être effectuées ?



Sélectionnez les tâches supplémentaires que l'assistant d'installation doit effectuer pendant l'installation de Visual Studio Code, puis cliquez sur Suivant.

Icônes supplémentaires :

- Créer une icône sur le Bureau

Autre :

- Ajouter l'action "Ouvrir avec Code" au menu contextuel de fichier de l'Explorateur Windows
- Ajouter l'action "Ouvrir avec Code" au menu contextuel de répertoire de l'Explorateur Windows
- Incrire Code en tant qu'éditeur pour les types de fichier pris en charge
- Ajouter à PATH (disponible après le redémarrage)

Précédent

Suivant

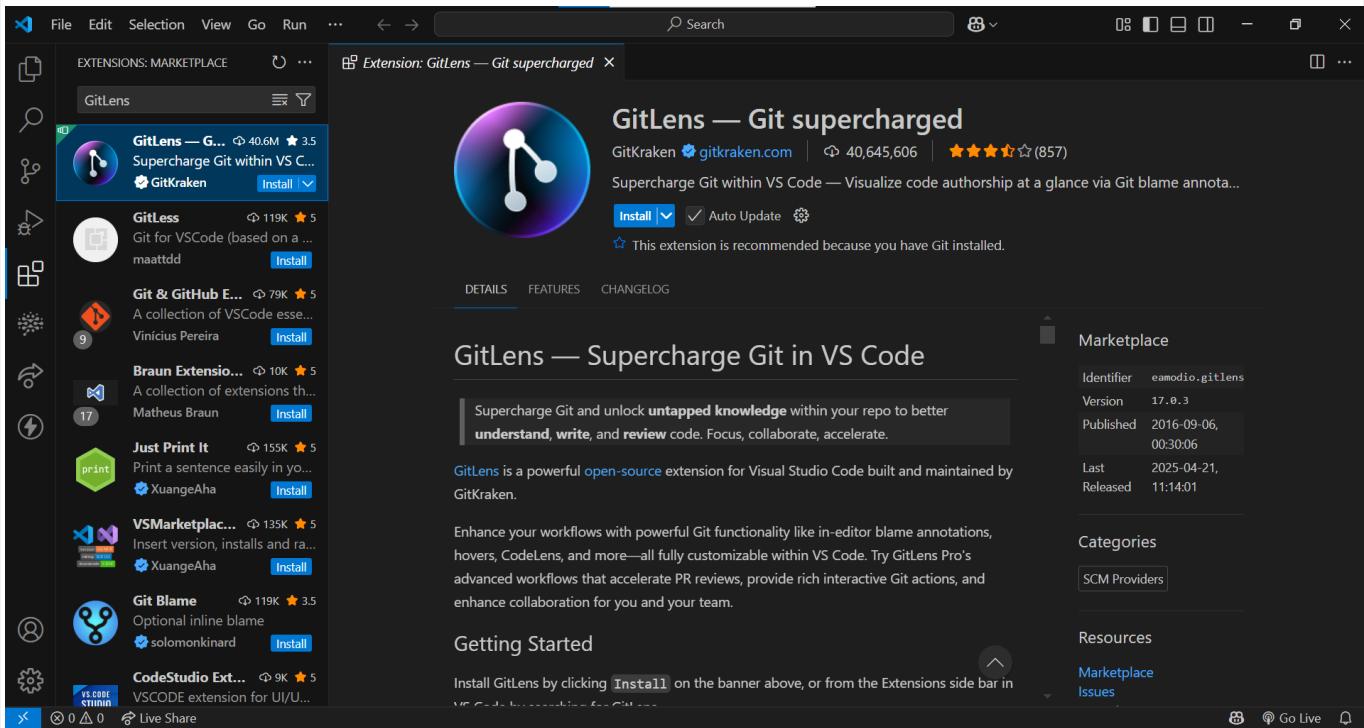
Annuler

J'ai choisi de travailler sur VSC car il est léger, rapide et gratuit. En travaillant avec VSC, j'ai accès à un écosystème d'extensions ultra riche et adaptable à n'importe quel stack web. Parmi ces extensions, certaines sont incontournables, telles que ESLint pour le linting du code, Prettier pour un formatage automatique, Live Server pour prévisualiser le site en temps réel lors du développement, ou bien encore Tailwind CSS, React Snippets, etc. J'ai également accès à une excellente autocomplétion grâce à IntelliSense, ce qui m'aide à gagner du temps, éviter les erreurs et être plus productif. Un autre avantage de VSC est que Git est intégré, ce qui permet une gestion des versions directement dans l'éditeur, ainsi que de committer, pusher et merger sans quitter VSC. Ce dernier offre aussi un excellent debugger pour JavaScript et TypeScript, ainsi que d'autres langages via des extensions, ce qui est parfait pour traquer les bugs côté front comme côté back (Node.js par exemple). VSC dispose d'un support pour le développement web mobile et hybride. Avec les bonnes extensions, j'ai pu travailler avec Ionic, React Native, et simuler des vues mobiles ainsi que des intégrations avec des outils comme Expo ou Flutter si besoin. J'ai utilisé VSC sur Windows, mais il est également disponible sur macOS et Linux, ce qui permet une synchronisation via un compte GitHub ou Microsoft afin de garder mes paramètres peu importe l'appareil sur lequel je travaille. Un important avantage avec VSC est sa grande communauté ; ainsi, en cas de problème, j'ai pu trouver des réponses rapidement sur Stack Overflow, GitHub ou dans la documentation des extensions, et avoir accès à de nombreux templates, snippets et configurations créées par d'autres utilisateurs.

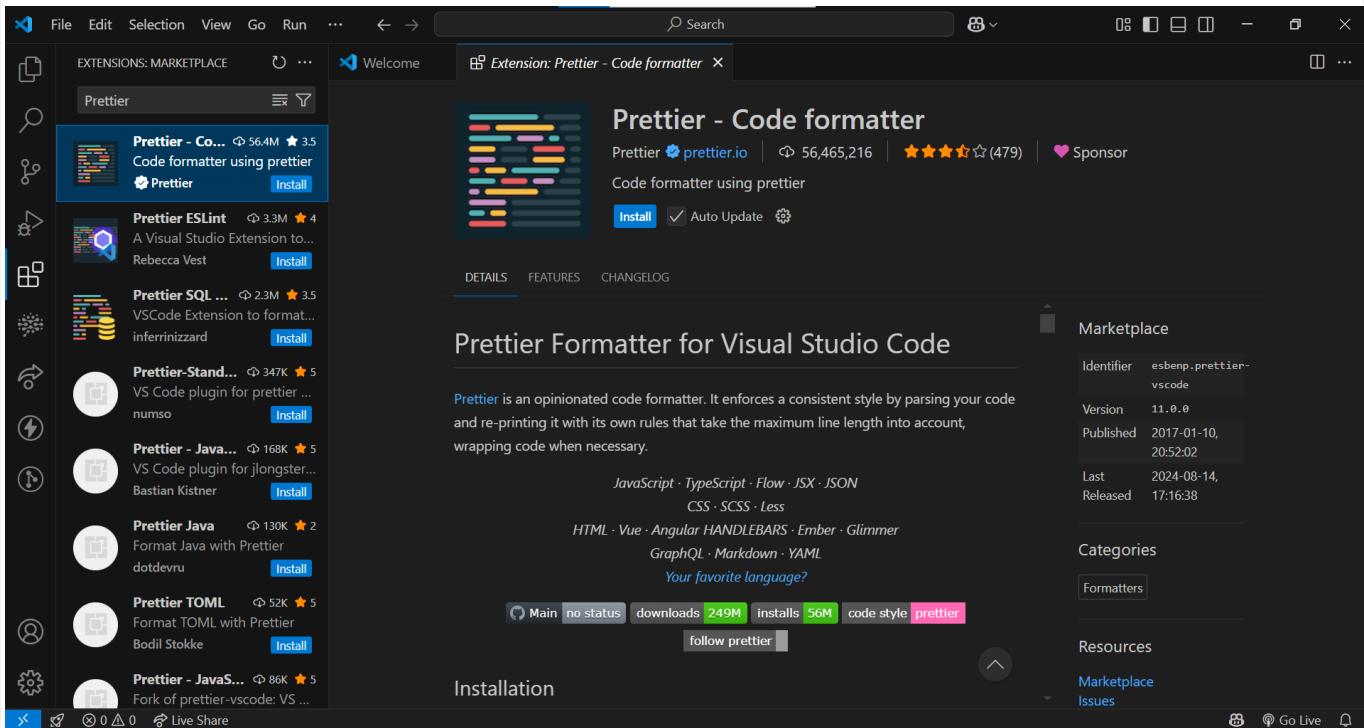
J'ai ensuite installé 4 extensions :

DOSSIER PROFESSIONNEL (DP)

- **GitLens : gestion de Git intégrée**

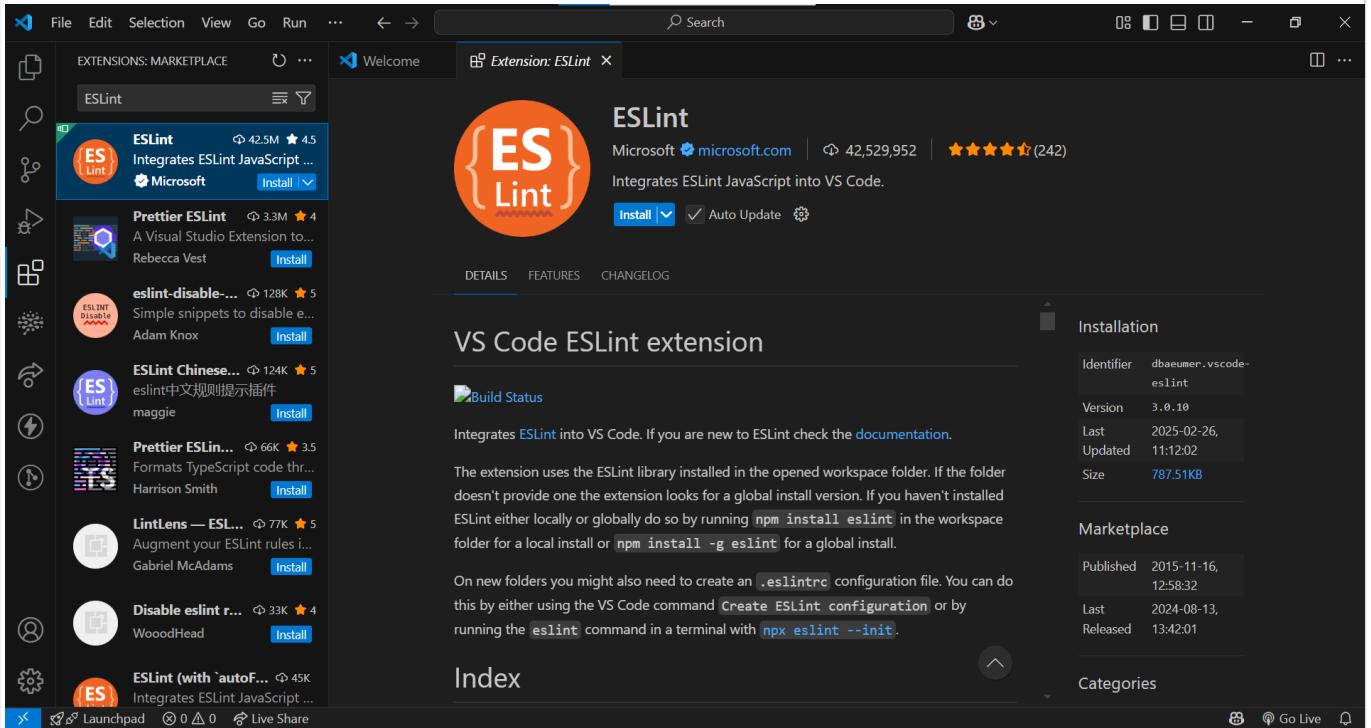


- **Prettier : formatage automatique du code**



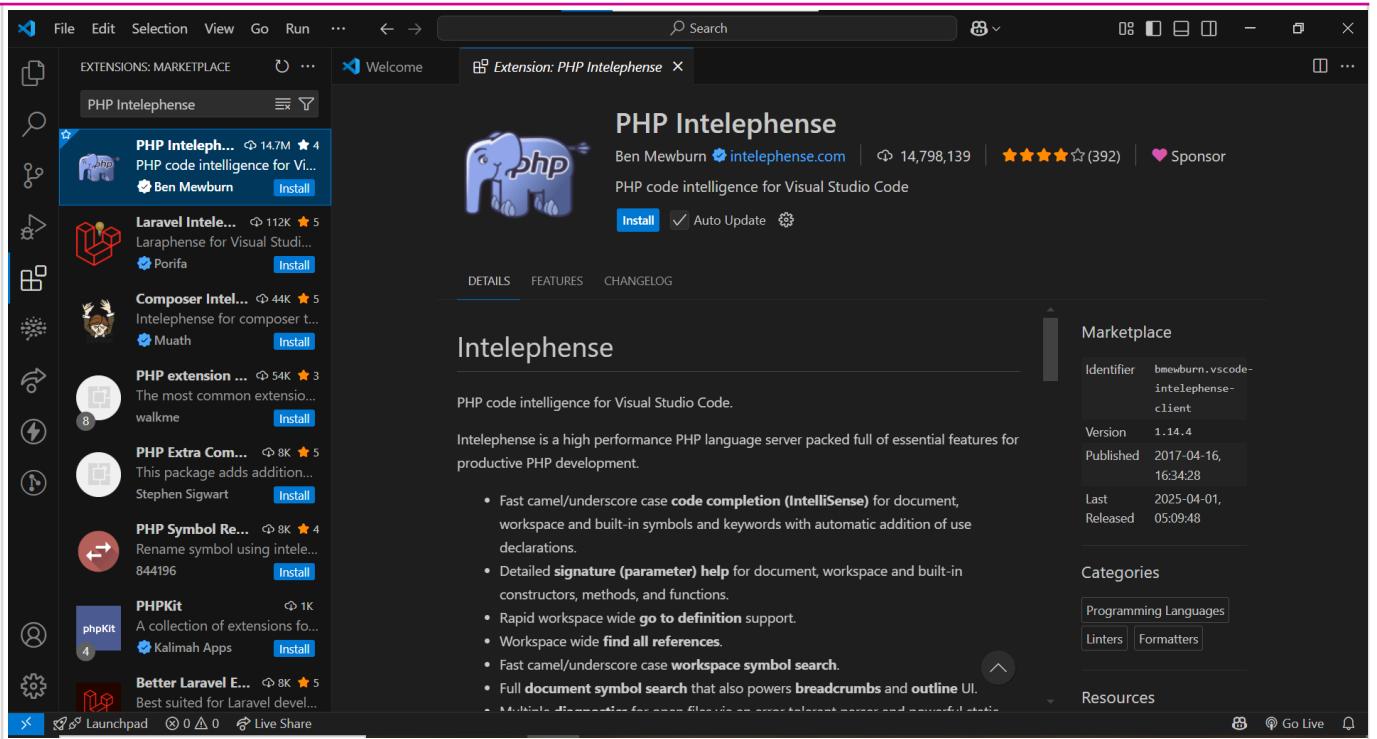
DOSSIER PROFESSIONNEL (DP)

- **ESLint : analyse de code JavaScript**



- **PHP Intelephense : pour PHP**

DOSSIER PROFESSIONNEL (DP)



Pour chacune, je clique simplement sur « Install » puis sur « Trust publisher and install ».

2. Configuration de Git

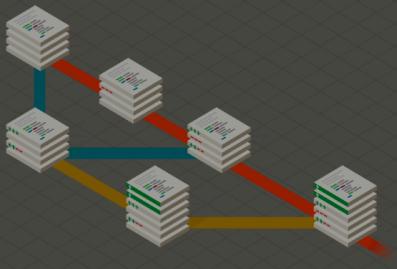
Je télécharge maintenant Git sur www.git-scm.com et l'installe.

DOSSIER PROFESSIONNEL (DP)

 **git** --fast-version-control

Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is easy to learn and has a **tiny footprint with lightning fast performance**. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like **cheap local branching**, **convenient staging areas**, and **multiple workflows**.



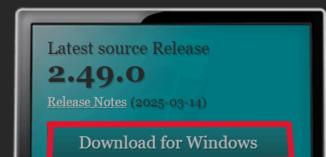
About

The advantages of Git compared to other source control systems.



Documentation

Command reference pages, Pro Git book content, videos and other material.



[Download for Windows](#)

 **git** --everything-is-local

[About](#)

[Documentation](#)

[Downloads](#)

GUI Clients
Logos

[Community](#)

The entire **Pro Git book** written by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).

Download for Windows

[Click here to download](#) the latest **(2.49.0) 64-bit** version of **Git for Windows**. This is the most recent maintained build. It was released **36 days ago**, on 2025-03-17.

Other Git for Windows downloads

Standalone Installer
[32-bit Git for Windows Setup.](#)
[64-bit Git for Windows Setup.](#)

Portable ("thumbdrive edition")
[32-bit Git for Windows Portable.](#)
[64-bit Git for Windows Portable.](#)

Using winget tool
Install **winget tool** if you don't already have it, then type this command in command prompt or Powershell.

```
winget install --id Git.Git -e --source winget
```

The current source code release is version **2.49.0**. If you want the newer version, you can build it from [the source code](#).

DOSSIER PROFESSIONNEL (DP)

Git 2.49.0 Setup

Information

Please read the following important information before continuing.



When you are ready to continue with Setup, click Next.

GNU General Public License

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change

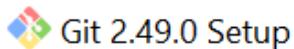
<https://gitforwindows.org/>

Only show new options

Next

Cancel

DOSSIER PROFESSIONNEL (DP)



Select Components

Which components should be installed?



Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

- Additional icons
 - On the Desktop
- Windows Explorer integration
 - Open Git Bash here
 - Open Git GUI here
- Git LFS (Large File Support)
- Associate .git* configuration files with the default text editor
- Associate .sh files to be run with Bash
- Check daily for Git for Windows updates
- (NEW!) Add a Git Bash Profile to Windows Terminal

Current selection requires at least 352,9 MB of disk space.

<https://gitforwindows.org/>

Only show new options

Back

Next

Cancel

DOSSIER PROFESSIONNEL (DP)



Choose a credential helper

Which credential helper should be configured?



Git Credential Manager

Use the [cross-platform Git Credential Manager](#).

See more information about the future of Git Credential Manager [here](#).

None

Do not use a credential helper.

<https://gitforwindows.org/>

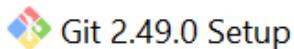
Only show new options

Back

Next

Cancel

DOSSIER PROFESSIONNEL (DP)



Configuring extra options

Which features would you like to enable?



Enable file system caching

File system data will be read in bulk and cached in memory for certain operations ("core.fscache" is set to "true"). This provides a significant performance boost.

Enable symbolic links

Enable [symbolic links](#) (requires the SeCreateSymbolicLink permission). Please note that existing repositories are unaffected by this setting.

<https://gitforwindows.org/>

Only show new options

Back

Install

Cancel

DOSSIER PROFESSIONNEL (DP)

Git 2.49.0 Setup

Choosing the default editor used by Git
Which editor would you like Git to use?



Use Visual Studio Code as Git's default editor

Visual Studio Code is an Open Source, lightweight and powerful editor running as a desktop application. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python, PHP, Go) and runtimes (such as .NET and Unity).

(WARNING!) This will be installed only for this user.

Use this option to let Git use Visual Studio Code as its default editor.

<https://gitforwindows.org/>

Only show new options [Back](#) [Next](#) [Cancel](#)

DOSSIER PROFESSIONNEL (DP)

Git 2.49.0 Setup

Choosing the SSH executable

Which Secure Shell client program would you like Git to use?

Use bundled OpenSSH
This uses ssh.exe that comes with Git.

Use external OpenSSH
NEW! This uses an external ssh.exe. Git will not install its own OpenSSH (and related) binaries but use them as found on the PATH.

<https://gitforwindows.org/>

Only show new options [Back](#) [Next](#) [Cancel](#)

DOSSIER PROFESSIONNEL (DP)



Adjusting your PATH environment

How would you like to use Git from the command line?



Use Git from Git Bash only

This is the most cautious choice as your PATH will not be modified at all. You will only be able to use the Git command line tools from Git Bash.

Git from the command line and also from 3rd-party software

(Recommended) This option adds only some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools. You will be able to use Git from Git Bash, the Command Prompt and the Windows PowerShell as well as any third-party software looking for Git in PATH.

Use Git and optional Unix tools from the Command Prompt

Both Git and the optional Unix tools will be added to your PATH.

Warning: This will override Windows tools like "find" and "sort". Only use this option if you understand the implications.

<https://gitforwindows.org/>

Only show new options

Back

Next

Cancel

DOSSIER PROFESSIONNEL (DP)

Git 2.49.0 Setup

Adjusting the name of the initial branch in new repositories

What would you like Git to name the initial branch after "git init"? 

Let Git decide
Let Git use its default branch name (currently: "master") for the initial branch in newly created repositories. The Git project [intends](#) to change this default to a more inclusive name in the near future.

Override the default branch name for new repositories
NEW! Many teams already renamed their default branches; common choices are "main", "trunk" and "development". Specify the name "git init" should use for the initial branch:

This setting does not affect existing repositories.

<https://gitforwindows.org/>

Only show new options

DOSSIER PROFESSIONNEL (DP)

Git 2.49.0 Setup

Choosing HTTPS transport backend
Which SSL/TLS library would you like Git to use for HTTPS connections?

Use the OpenSSL library
Server certificates will be validated using the ca-bundle.crt file.

Use the native Windows Secure Channel library
Server certificates will be validated using Windows Certificate Stores.
This option also allows you to use your company's internal Root CA certificates distributed e.g. via Active Directory Domain Services.

https://gitforwindows.org/

Only show new options

DOSSIER PROFESSIONNEL (DP)

Git 2.49.0 Setup

Configuring the line ending conversions

How should Git treat line endings in text files?



Checkout Windows-style, commit Unix-style line endings

Git will convert LF to CRLF when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Windows ("core.autocrlf" is set to "true").

Checkout as-is, commit Unix-style line endings

Git will not perform any conversion when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Unix ("core.autocrlf" is set to "input").

Checkout as-is, commit as-is

Git will not perform any conversions when checking out or committing text files. Choosing this option is not recommended for cross-platform projects ("core.autocrlf" is set to "false").

<https://gitforwindows.org/>

Only show new options

Back

Next

Cancel

DOSSIER PROFESSIONNEL (DP)

Git 2.49.0 Setup



Completing the Git Setup Wizard

Setup has finished installing Git on your computer. The application may be launched by selecting the installed shortcuts.

Click Finish to exit Setup.



- Launch Git Bash
- View Release Notes

Only show new options

Finish

Je vérifie l'installation de Git :

J'appuie sur Windows + R et j'écris « powershell » pour accéder au terminal de commande puis j'appuie sur Entrée.

DOSSIER PROFESSIONNEL (DP)

Exécuter



Entrez le nom d'un programme, dossier, document ou ressource Internet, et Windows l'ouvrira pour vous.

Ouvrir :

powershell

OK

Annuler

Parcourir...

Dans PowerShell, j'écris « git --version » et j'appuie sur Entrée pour vérifier l'installation.

Windows PowerShell

```
windows PowerShell  
Copyright (C) Microsoft Corporation. Tous droits réservés.
```

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

```
PS C:\Users\Utilisateur> git --version  
git version 2.49.0.windows.1  
PS C:\Users\Utilisateur>
```

Si j'ai le message « git version » suivi d'une version, c'est que l'installation a réussi. Sinon, je recommence l'installation.

Je configure ensuite mon identité avec :

- `git config --global user.name`
- `git config --global user.email`

Windows PowerShell

```
PS C:\Users\Utilisateur> git config --global user.name "Antoine Petit"  
PS C:\Users\Utilisateur> git config --global user.email "antooine.petit@gmail.com"
```

Je configure ensuite VSC comme éditeur par défaut avec :

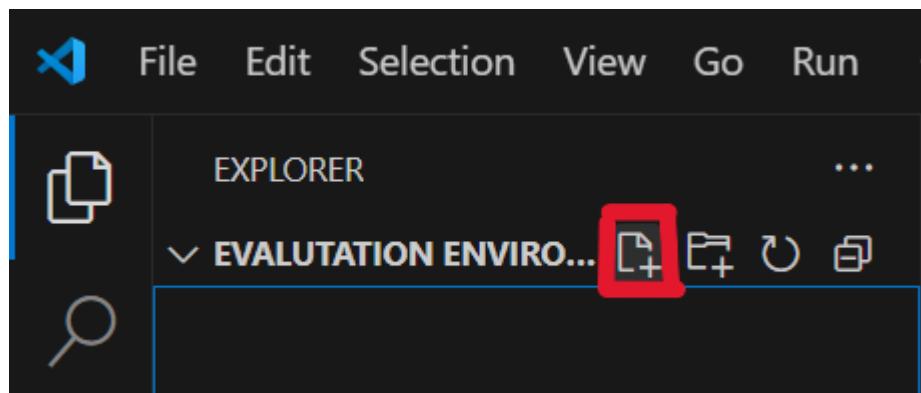
- `git config --global core.editor "code -wait"`

DOSSIER PROFESSIONNEL (DP)

Windows PowerShell

```
PS C:\Users\Utilisateur> git config --global core.editor "code --wait"
```

J'accède ensuite à mon environnement de travail dans VSC et j'y crée un fichier .gitignore qui me servira à exclure des fichiers et dossiers spécifiques de mon repository Git.



J'initialise mon repository Git dans le terminal de VSC avec les commandes appropriées.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS powershell + - ×
● PS C:\Users\Utilisateur\Desktop\DWWM\Evalutation environnement de travail> git init
Initialized empty Git repository in C:/Users/Utilisateur/Desktop/DWWM/Evalutation environnement de travail/.git/
● PS C:\Users\Utilisateur\Desktop\DWWM\Evalutation environnement de travail> git add *
● PS C:\Users\Utilisateur\Desktop\DWWM\Evalutation environnement de travail> git commit -m "Init repo"
[main (root-commit) 252749b] Init repo
 1 file changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 .gitignore
● PS C:\Users\Utilisateur\Desktop\DWWM\Evalutation environnement de travail> git branch -M main
```

3. Mise en place d'un serveur local

Je télécharge WAMP sur www.wampserver.com.

DOSSIER PROFESSIONNEL (DP)

WampServer
Apache, PHP, MySQL sous Windows DÉMARRER TÉLÉCHARGER FORMATION FORUM
HÉBERGEMENT CLOUD

TÉLÉCHARGEMENTS

WampServer est disponible gratuitement (sous licence GPL) en deux versions : 32 et 64 bits. Wampserver 3 n'est pas compatible avec Windows Vista, ni avec Windows Server 2008. Les anciennes versions de Wampserver sont chez [Sourceforge](#).

WAMP SERVER

WAMP SERVER 64 BITS (X64) 3.3.5

- WAMP SERVER 64 BITS (x64) 3.3.5 – Apache 2.4.59 – PHP 7.4.33/8.0.30/8.1.28/8.2.18/8.3.6 – MySQL 8.3.0 – MariaDB 11.3.2 – PhpMyAdmin 5.2.1 – Adminer 4.8.1 – PhpSysInfo 3.4.3.

[changelog](#)

WAMP SERVER 32 BITS (X86) 3.3.0

- WAMP SERVER 32BITS (86) 3.3.0 – Wampserver 3.3.0 32 bit x86 – Apache 2.4.54.2 – PHP 7.4.33/8.0.26/8.1.13/8.2.0 – MySQL 5.7.40 – MariaDB 10.6.5 – PhpMyAdmin 5.2.0 – Adminer 4.8.1 – PhpSysInfo 3.4.2.

[changelog](#)

TÉLÉCHARGER WAMP SERVER 64 BITS (X64) 3.3.5

WampServer est disponible gratuitement (sous licence GPL). Vous pouvez remplir ce formulaire qui nous permettra de vous faire parvenir les actualités formation d'Alter Way, société éditrice, ainsi que toutes les informations liées aux évolutions de WampServer. Si vous ne le souhaitez pas, vous pouvez [passer au téléchargement direct](#).

SOURCEFORGE

Business Software Open Source Software SourceForge Podcast Resources NEXO Earn up to 15% per year [Find App](#) Search for software or

Home / Open Source Software / Internet / HTTP Servers / WampServer / Files

WampServer Files

A Windows Web development environment for Apache, MySQL, PHP databases
Brought to you by: [alterway](#), [hervelecclerc](#), [otomatic](#)

Summary Files Reviews Support Tickets News

Download Latest Version
[wampserver3.3.7_x64.exe \(370.5 MB\)](#)

Get an email when there's a new version of WampServer Enter your email at Next 

Name	Modified	Size	Downloads / Week
WampServer 3	2018-12-09	26,638 	26,638

Totals: 1 Item

Recommended Projects

-  phpMyAdmin A software tool to bring MySQL to the Web
-  EasyPHP A complete WAMP environment for PHP...
-  XAMPP An easy to install Apache distribution containing MySQL...
-  AIOHTTP Asynchronous HTTP client/server framework for...

J'ai utilisé WAMP car il permet de créer un environnement de développement complet incluant Apache (serveur web), MySQL (gestion des bases de données) et PHP (langage serveur). Il existe d'autres serveurs similaires pour d'autres systèmes d'exploitation comme XAMPP (multi-OS) ou MAMP (MacOS).

DOSSIER PROFESSIONNEL (DP)

W Installation - Wampserver64 3.3.7



Accord de licence

Les informations suivantes sont importantes. Veuillez les lire avant de continuer.

Veuillez lire le contrat de licence suivant. Vous devez en accepter tous les termes avant de continuer l'installation.

** WampServer

Créateur : Romain Bourdon
Mise à jour vers 2.5 : Herve Leclerc
Mise à jour 2.5 to 3.0.0 : Otomatic (wampserver@otomatic.net)
<http://forum.wampserver.com/index.php>

LICENCE PUBLIQUE GÉNÉRALE GNU
Version 3, du 29 juin 2007.

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>

Chacun est autorisé à copier et distribuer des copies conformes de ce document de licence, mais toute modification en est proscrite.

Traduction française par Philinne Verdy

- Je comprends et j'accepte les termes du contrat de licence
 Je refuse les termes du contrat de licence

Installation Wampserver 3.3.7 par Inno Setup

Suivant

Annuler

DOSSIER PROFESSIONNEL (DP)

Installation - Wampserver64 3.3.7

Information

Les informations suivantes sont importantes. Veuillez les lire avant de continuer.



Lorsque vous êtes prêt à continuer, cliquez sur Suivant.

Installation de Wampserver

Si vous voulez éviter des problèmes, il faut impérativement lire la totalité de ce message.

AVANT de procéder à l'installation de Wampserver, vous devez vous assurer que certains éléments sont installés sur votre système, sinon, Wampserver ne pourra pas fonctionner, l'installation sera défectueuse et vous devrez supprimer Wampserver AVANT d'installer les éléments qui manquaient.

Les paquetages Microsoft Visual Studio VC10, VC11, VC15, VC17 sont impérativement requis pour Wampserver 3 même si vous n'utilisez que des versions Apache et PHP VC11. À la fin de ce texte, sous la rubrique --- Paquetages VC --- vous trouverez les liens de téléchargement desdits paquetages.

--- N'installez pas Wampserver PAR-DESSUS une version existante, suivez les conseils de :

- Installer une nouvelle version de Wampserver : <http://forum.wampserver.com/read.php?1,119444>

--- Installez Wampserver dans un dossier à la racine d'un disque, par exemple C:

Installation Wampserver 3.3.7 par Inno Setup

Précédent

Suivant

Annuler

Après l'installation, je vais dans le menu Démarrer et je lance Wampserver64.

4. Préparation d'un projet

Node.js

J'ai commencé par télécharger et installer Node.js (version LTS recommandée pour sa stabilité), puis je lance l'installation.

DOSSIER PROFESSIONNEL (DP)

The screenshot shows the official Node.js website. At the top, there's a navigation bar with links for "Apprendre", "À propos", "Téléchargement", "Blog", "Docs", "Contribuer", "Certification", a search bar ("Q. Commencer à écrire... Ctrl + K"), and some icons. The main content area has a dark background with a hexagonal pattern. On the left, there's a large title "Exécuter du JavaScript partout". Below it, a paragraph describes Node.js as a free, open-source, multiplatform environment for executing JavaScript. A red-bordered button labeled "Télécharger Node.js (LTS)" is visible. On the right, there's a code editor window titled "Create an HTTP Server" showing a simple Node.js script to start an HTTP server on port 3000. The code is as follows:

```
1 // server.mjs
2 import { createServer } from 'node:http';
3
4 const server = createServer((req, res) => {
5   res.writeHead(200, { 'Content-Type': 'text/plain' });
6   res.end('Hello World!\n');
7 });
8
9 // starts a simple http server locally on port 3000
10 server.listen(3000, '127.0.0.1', () => {
11   console.log('Listening on 127.0.0.1:3000');
12 });
13
14 // run with `node server.mjs`
```

Below the code editor, there are tabs for "JavaScript" and "Copier dans le Presse-papier".

DOSSIER PROFESSIONNEL (DP)

Node.js Setup

Custom Setup

Select the way you want features to be installed.



Click the icons in the tree below to change the way features will be installed.

- Node.js runtime
- corepack manager
- npm package manager
- Online documentation shortcuts
- Add to PATH

Install the core Node.js runtime (node.exe).

This feature requires 79MB on your hard drive.

Browse...

Reset

Disk Usage

Back

Next

Cancel

DOSSIER PROFESSIONNEL (DP)

Node.js Setup

Tools for Native Modules

Optionally install the tools necessary to compile native modules.



Some npm modules need to be compiled from C/C++ when installing. If you want to be able to install such modules, some tools (Python and Visual Studio Build Tools) need to be installed.

Automatically install the necessary tools. Note that this will also install Chocolatey. The script will pop-up in a new window after the installation completes.

Alternatively, follow the instructions at <https://github.com/nodejs/node-gyp#on-windows> to install the dependencies yourself.

Back

Next

Cancel

J'appuie sur Windows + R, j'écris « powershell » et j'appuie sur Entrée.

Je vérifie l'installation à l'aide des commandes :

- node -v (affiche la version de Node.js)
- npm -v (affiche la version du gestionnaire de paquets)

Windows PowerShell

```
PS C:\Users\Utilisateur> node -v
v22.14.0
PS C:\Users\Utilisateur> npm -v
10.9.2
```

DOSSIER PROFESSIONNEL (DP)

Je crée ensuite mon projet Node.js avec :

- mkdir mon-projet-node
- cd mon-projet-node
- npm init -y
- npm install express

```
PS C:\Users\Utilisateur\Desktop\dwww\Node> mkdir mon-projet-node

Répertoire : C:\Users\Utilisateur\Desktop\dwww\Node

Mode                LastWriteTime        Length Name
----                -----          ----  --
d----- 22/04/2025      11:58           0   mon-projet-node

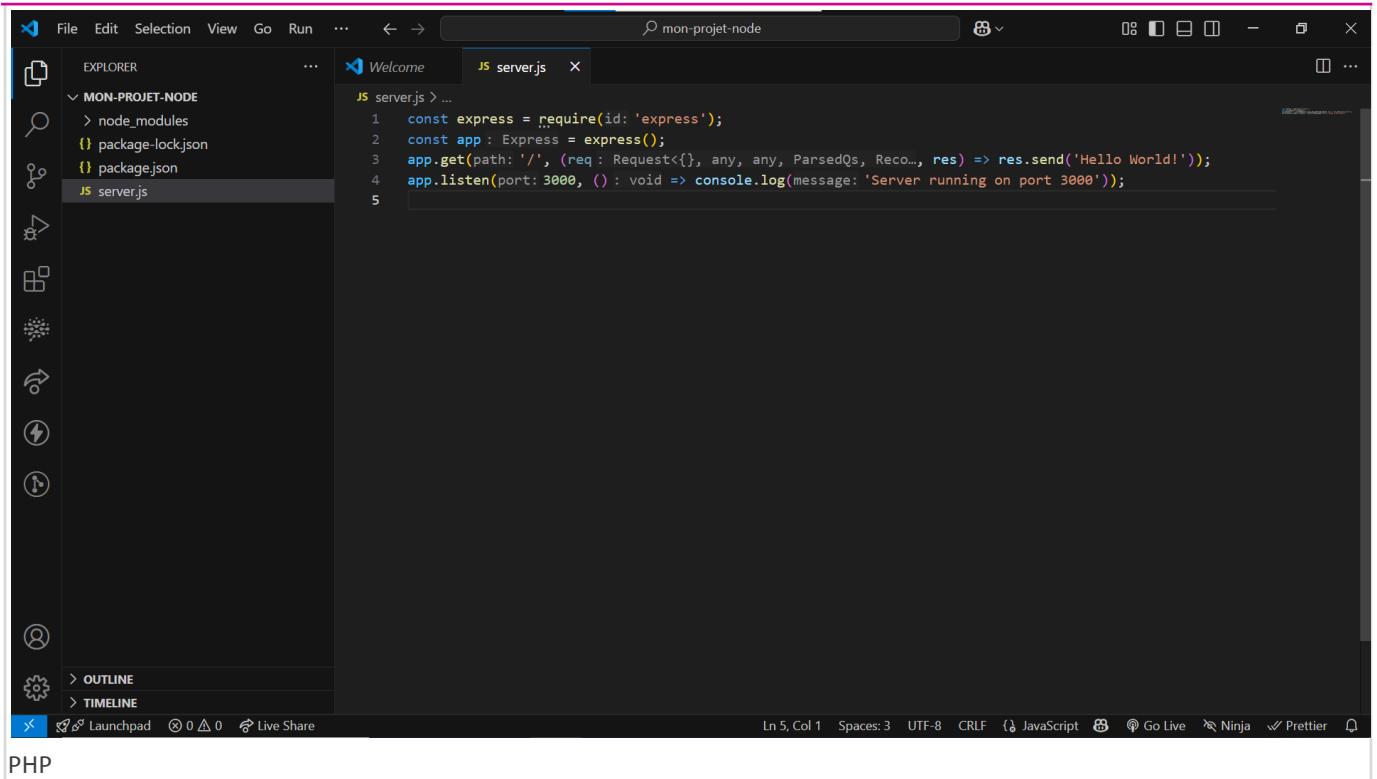
PS C:\Users\Utilisateur\Desktop\dwww\Node> cd mon-projet-node
PS C:\Users\Utilisateur\Desktop\dwww\Node\mon-projet-node> npm init -y
wrote to C:\Users\Utilisateur\Desktop\dwww\Node\mon-projet-node\package.json:

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

added 66 packages, and audited 67 packages in 4s
14 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
PS C:\Users\Utilisateur\Desktop\dwww\Node\mon-projet-node>
```

J'ouvre ensuite VSC et je crée un fichier JavaScript nommé server.js, en suivant la convention. J'y écris le code nécessaire pour définir le port sur lequel sera hébergé mon serveur Node.js.

DOSSIER PROFESSIONNEL (DP)



The screenshot shows the Visual Studio Code interface. On the left is the Explorer sidebar, which displays a project structure for 'MON-PROJET-NODE' containing 'node_modules', 'package-lock.json', 'package.json', and 'server.js'. The main editor area shows the 'server.js' file with the following code:

```
1 const express = require('express');
2 const app : Express = express();
3 app.get(path: '/', (req: Request<{}, any, any, ParsedQs, Recordable>, res) => res.send('Hello World!'));
4 app.listen(port: 3000, () : void => console.log(message: 'Server running on port 3000'));
```

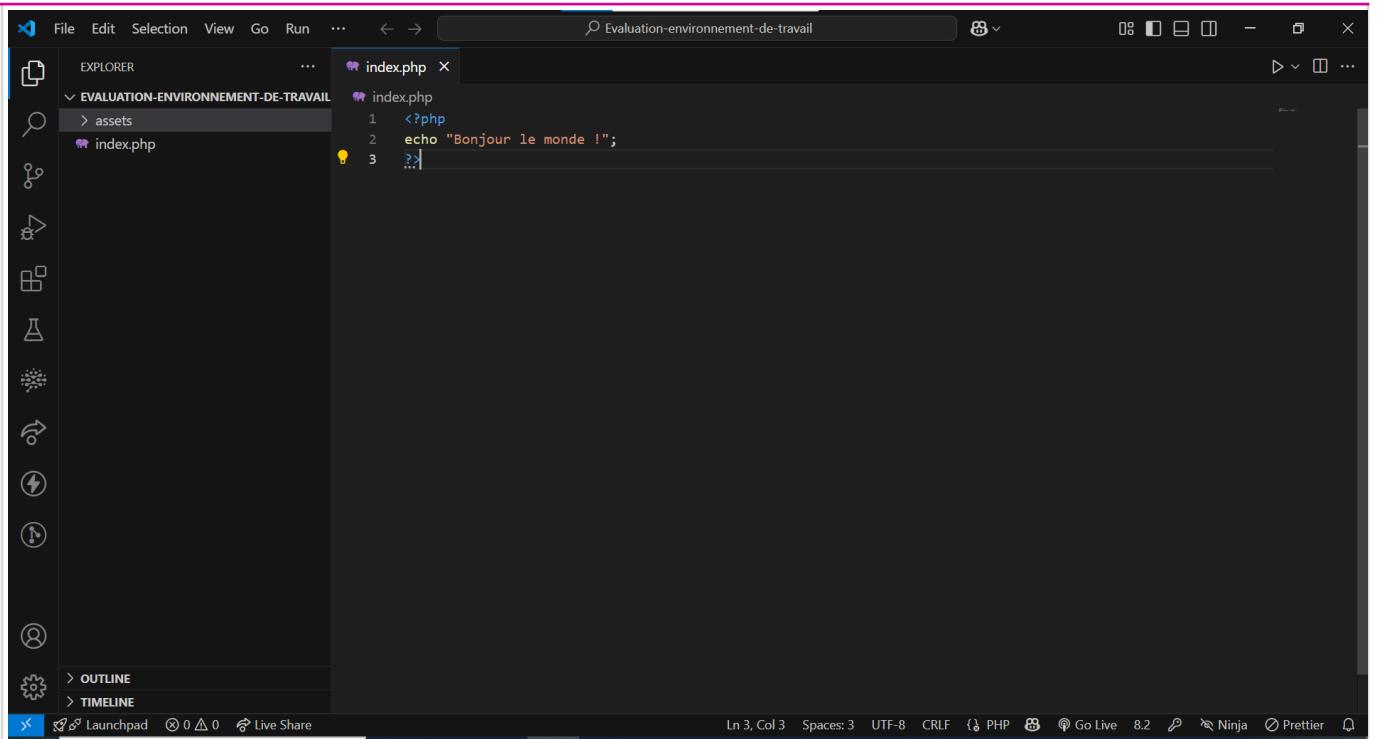
The status bar at the bottom indicates the code is in JavaScript mode, with settings for 5 spaces, UTF-8 encoding, CRLF line endings, and tools like Go Live, Ninja, and Prettier.

PHP

J'accède à mon dossier www dans WAMP.

Pour l'évaluation, je crée un dossier que j'ouvre dans VSC, j'y crée un dossier assets et un fichier index.php dans lequel j'écris mon code PHP.

DOSSIER PROFESSIONNEL (DP)



The screenshot shows a dark-themed code editor interface. On the left is the Explorer sidebar with a tree view showing a folder named 'EVALUATION-ENVIRONNEMENT-DE-TRAVAIL' containing an 'assets' folder and an 'index.php' file. The main workspace shows an 'index.php' file with the following content:

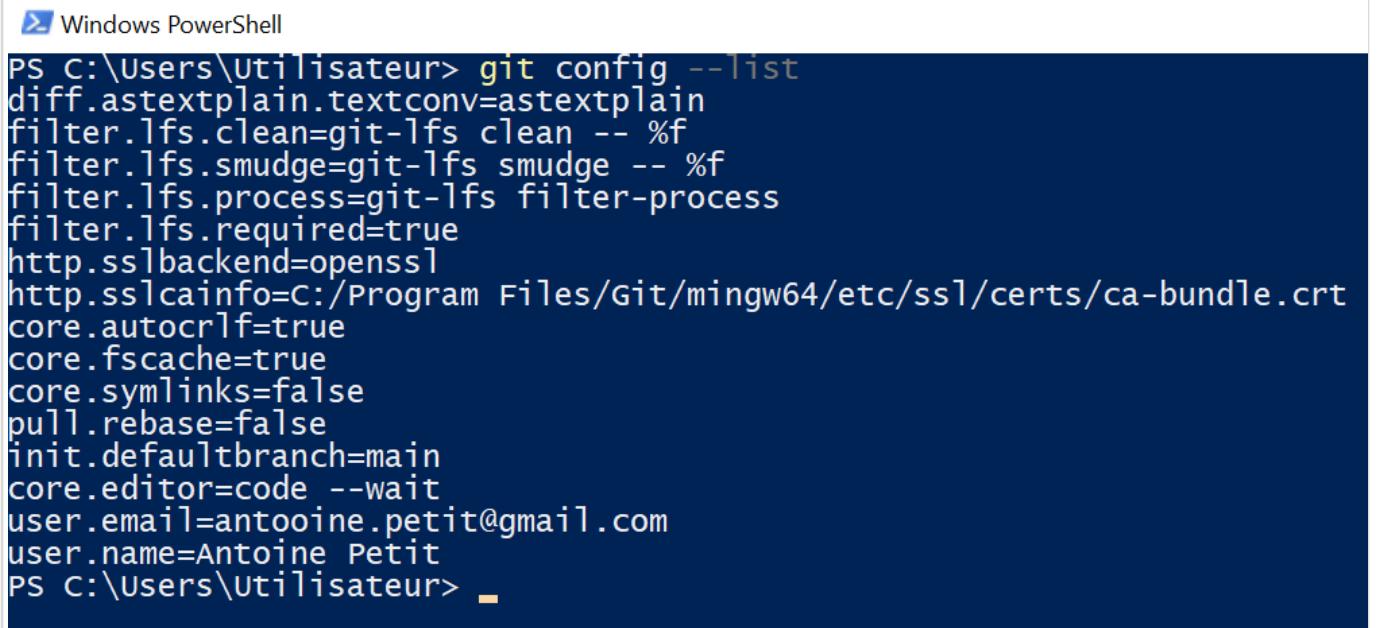
```
<?php  
echo "Bonjour le monde !";
```

The status bar at the bottom displays: Ln 3, Col 3, Spaces: 3, UTF-8, CRLF, PHP, Go Live, Ninja, Prettier.

5. Validation finale

Je vérifie ma configuration Git avec :

- `git config --list`



The screenshot shows a Windows PowerShell window with the following command and its output:

```
PS C:\Users\Utilisateur> git config --list
```

The output lists various Git configuration settings:

```
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
init.defaultbranch=main
core.editor=code --wait
user.email=antooine.petit@gmail.com
user.name=Antoine Petit
PS C:\Users\Utilisateur>
```

DOSSIER PROFESSIONNEL (DP)

Je teste ensuite mon serveur Node.js en exécutant :

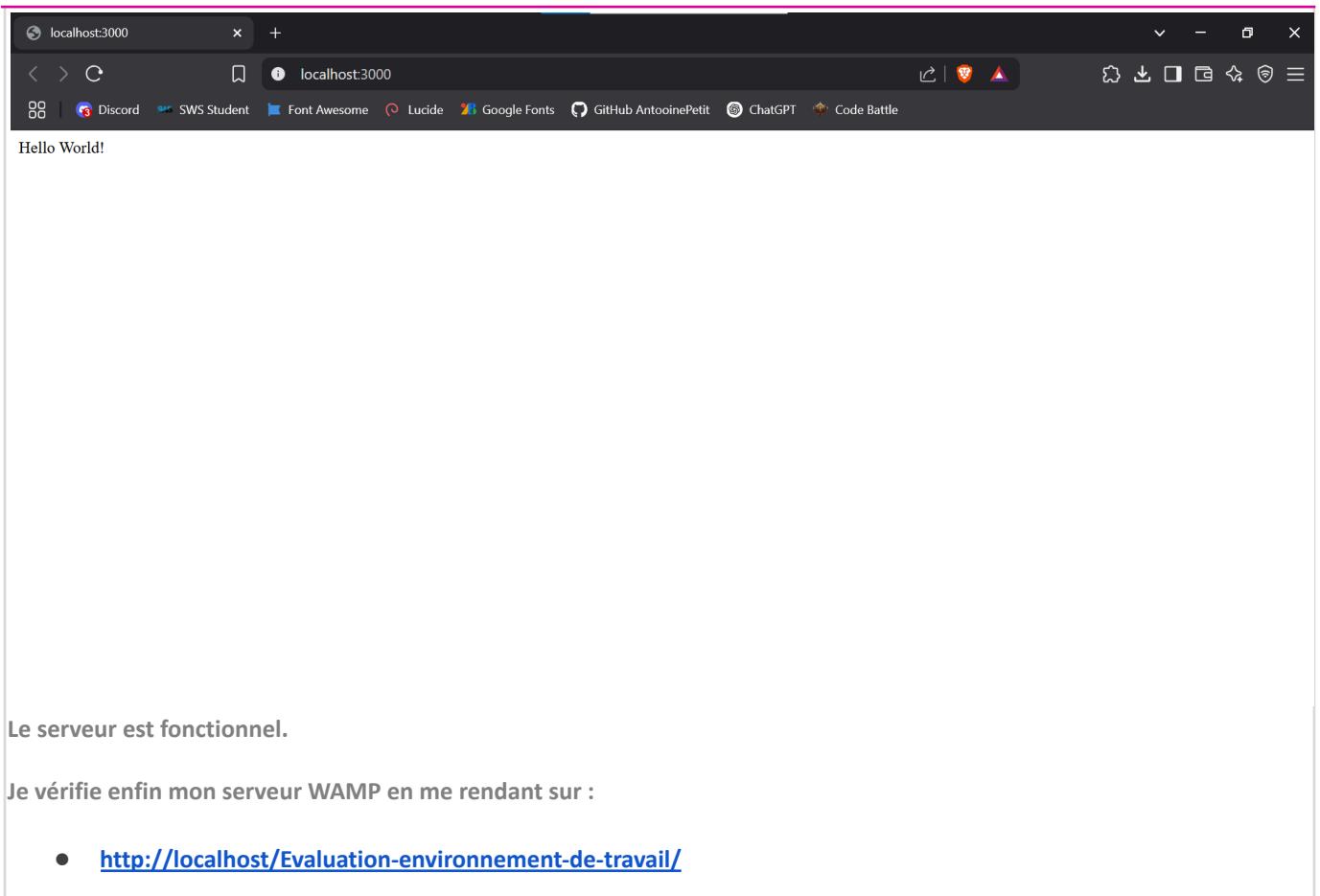
- node .\server.js

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a project named "mon-projet-node" containing "server.js".
- Terminal:** Displays the command "PS C:\Users\Utilisateur\Desktop\DWMM\Evaluation environnement de travail\mon-projet-node> node .\server.js" and the output "Server running on port 3000".
- Status Bar:** Shows file paths like "main*", "Launchpad", "Live Share", and code settings like "Spaces: 3", "UTF-8", "CRLF".

puis en allant sur <http://localhost:3000/>

DOSSIER PROFESSIONNEL (DP)



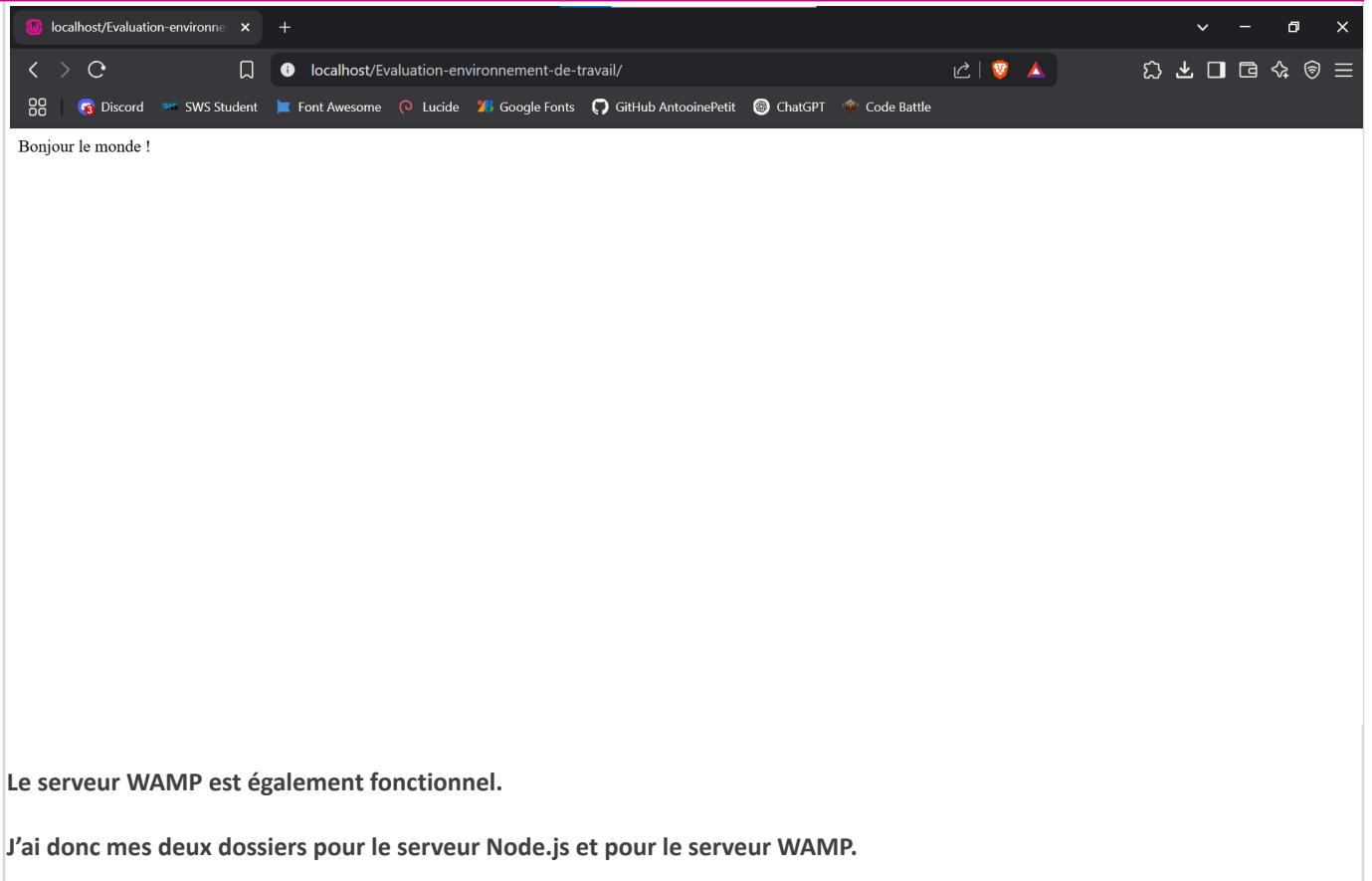
A screenshot of a browser window titled "localhost:3000". The address bar also shows "localhost:3000". The page content area displays the text "Hello World!". The browser interface includes standard navigation buttons (back, forward, search), a tab bar with various icons, and a toolbar with download, refresh, and other functions.

Le serveur est fonctionnel.

Je vérifie enfin mon serveur WAMP en me rendant sur :

- <http://localhost/Evaluation-environnement-de-travail/>

DOSSIER PROFESSIONNEL (DP)



A screenshot of a web browser window. The address bar shows "localhost/Evaluation-environnement-de-travail/". The page content area displays the text "Bonjour le monde !". Below the browser window, there is a text block containing two statements about the WAMP server.

Bonjour le monde !

Le serveur WAMP est également fonctionnel.

J'ai donc mes deux dossiers pour le serveur Node.js et pour le serveur WAMP.

DOSSIER PROFESSIONNEL (DP)

The image shows two instances of Microsoft Visual Studio Code (VS Code) running side-by-side. Both windows have a dark theme.

Top Window (PHP):

- Explorer:** Shows a folder named "EVALUATION-ENVIRONNEMENT-DE-TRAVAIL" containing an "assets" folder and an "index.php" file.
- Editor:** The "index.php" file is open, displaying the following code:

```
<?php  
echo "Bonjour le monde !";  
??
```
- Status Bar:** Shows "Ln 3, Col 3" and "Spaces: 3".

Bottom Window (Node.js):

- Explorer:** Shows a folder named "EVALUTION ENVIRONNEMENT DE TRAV..." containing a "mon-projet-node" folder which includes "node_modules", "package-lock.json", "package.json", "server.js", and ".gitignore".
- Editor:** The "server.js" file is open, displaying the following code:

```
const express = require('express');  
const app = express();  
app.get('/', (req, res) => res.send('Hello World!'));  
app.listen(3000, () => console.log('Server running on port 3000'));
```
- Status Bar:** Shows "Ln 5, Col 1" and "Spaces: 3".

2. Précisez les moyens utilisés :

DOSSIER PROFESSIONNEL (DP)

Dans le cadre de cette évaluation, j'ai travaillé avec Visual Studio Code, WAMP server, Node et Git. Je me suis pleinement servi de mes compétences et connaissances apprises lors de la formation.

3. Avec qui avez-vous travaillé ?

J'ai travaillé en totale autonomie sur cette évaluation.

4. Contexte

Nom de l'entreprise, organisme ou association - La plateforme

Chantier, atelier, service - Formation Développeur Web & Web Mobile

Période d'exercice - Du 22/04/2025 au 22/04/2025

5. Informations complémentaires (*facultatif*)

DOSSIER PROFESSIONNEL (DP)

Cliquez ici pour taper du texte.

DOSSIER PROFESSIONNEL (DP)

Activité-type 1 Développer la partie front-end d'une application web ou web mobile sécurisée

Exemple n°2 - Maquetter des interfaces utilisateur web ou web mobile

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Pour ce projet, nous imaginons une commande fictive.

J'ai été missionné pour concevoir les maquettes d'une application web ou web mobile destinée à la gestion et au suivi de collections de plantes d'intérieur rares. L'application doit permettre aux utilisateurs de cataloguer leurs plantes, suivre leur évolution et recevoir des rappels pour leur entretien. Vous devrez créer des maquettes illustrant l'interface et le parcours utilisateur pour les fonctionnalités essentielles de l'application. Je vais répondre aux besoins décrits dans 9 user stories transmises avec la demande :

Créer une fiche plante

- En tant que collectionneur, je veux pouvoir ajouter une nouvelle plante à ma collection avec son nom, son espèce, sa date d'acquisition et sa photo, afin de la cataloguer facilement.

Consulter le détail d'une plante

- En tant que collectionneur, je veux pouvoir consulter la fiche détaillée d'une plante (historique d'arrosage, de fertilisation, notes de croissance), afin de suivre son évolution dans le temps.

Modifier ou supprimer une plante de ma collection

- En tant que collectionneur, je veux pouvoir mettre à jour ou supprimer les informations d'une plante, afin de corriger ou retirer des plantes de ma collection.

Recevoir des notifications d'entretien

- En tant que collectionneur, je veux recevoir des notifications personnalisées pour l'arrosage et la fertilisation, afin de ne pas oublier les soins de mes plantes.

Configurer les rappels d'entretien

- En tant que collectionneur, je veux pouvoir personnaliser la fréquence et l'heure des rappels d'entretien pour chaque plante, afin d'adapter les notifications à leurs besoins spécifiques.

Ajouter des photos d'évolution

- En tant que collectionneur, je veux pouvoir ajouter régulièrement des photos à la fiche d'une plante, afin de documenter visuellement sa croissance au fil du temps.

Filtrer ma collection de plantes

- En tant que collectionneur, je veux pouvoir filtrer mes plantes par type, besoin en lumière ou date d'acquisition, afin de mieux organiser ma collection.

Trier ma collection de plantes

- En tant que collectionneur, je veux pouvoir trier mes plantes par ordre alphabétique, date d'acquisition ou dernier entretien, afin de naviguer facilement dans ma collection.

DOSSIER PROFESSIONNEL (DP)

Visualiser un historique global d'entretien

- En tant que collectionneur, je veux accéder à une page qui liste l'historique global des entretiens (arrosage, fertilisation, rempotage) pour toutes mes plantes, afin de suivre l'évolution de mes actions sur l'ensemble de ma collection.

À partir des informations qui m'ont été données, je me suis rendu sur Readdy.ai et j'ai demandé à l'intelligence artificielle de me proposer une page d'accueil correspondant à mes attentes. Je lui ai demandé des couleurs proches du vert et du thème de la nature, un header contenant un logo, un menu de navigation, une barre de recherche et un bouton pour accéder à une page de connexion, un main contenant une liste de plante suggérées à l'utilisateur, et un footer regroupant les informations légales, les liens des réseaux sociaux ainsi que les liens utiles. J'ai obtenu ce résultat :

DOSSIER PROFESSIONNEL (DP)

PlantHub Accueil Ma Collection Explorer Guides

Rechercher des plantes...

Connexion

Découvrez le monde des plantes

Collectionnez, apprenez et prenez soin de vos plantes favorites. Rejoignez notre communauté de passionnés de botanique.

Commencer ma collection

Explorer les plantes

Plantes suggérées pour vous

Découvrez notre sélection de plantes parfaites pour débuter votre collection



Monstera Deliciosa

Monstera deliciosa

☀️ Lumière indirecte ⚪ 1-2 fois/semaine



Sansevieria

Sansevieria trifasciata

☀️ Faible à moyenne ⚪ 1 fois/2 semaines



Pothos Doré

Epipremnum aureum

☀️ Lumière indirecte ⚪ 1 fois/semaine



Ficus Lyrata

Ficus lyrata

☀️ Lumière vive indirecte ⚪ 1 fois/semaine



Zamiooculcas

Zamiooculcas zamiifolia

☀️ Faible à moyenne ⚪ 1 fois/2-3 semaines



Philodendron

Philodendron hederaceum

☀️ Lumière indirecte ⚪ 1 fois/semaine

PlantHub

Découvrez, collectionnez et prenez soin de vos plantes favorites. Rejoignez notre communauté de passionnés de botanique.

Liens rapides

Ma Collection
Guides de soins
Communauté

Suivez-nous



DOSSIER PROFESSIONNEL (DP)

Après avoir obtenu ce résultat, j'ai décidé de partir de ce design qui me convenait et je me suis rendu sur figma pour créer une maquette autour de ce design avant d'aller plus loin. J'ai retiré les onglets « Explorer » et « Guides » car je ne comptais pas m'en servir ni les intégrer, ainsi que les niveaux de difficulté (Facile, intermédiaire) sur les plantes qui ne m'intéressaient pas dans ce projet. Après maquettage j'ai obtenu ce résultat (presque identique au design initial) :

The screenshot shows the PlantHub website homepage. At the top, there's a navigation bar with 'PlantHub' and 'Accueil' (Home) links, a search bar containing 'Rechercher des plantes...', and a 'Connexion' (Login) button. The main header features a large image of various green plants and the text 'Découvrez le monde des plantes' (Discover the world of plants). Below the header, a sub-header reads 'Collectionnez, apprenez et prenez soin de vos plantes favorites. Rejoignez notre communauté de passionnés de botanique.' (Collect, learn, and take care of your favorite plants. Join our community of botanical enthusiasts.). There are two buttons: 'Commencer ma collection' (Start my collection) and 'Explore les plantes' (Explore plants). The main content area is titled 'Plantes suggérées pour vous' (Recommended plants for you) and includes a sub-header 'Découvrez notre sélection de plantes parfaites pour débuter votre collection' (Discover our selection of perfect plants to start your collection). Six plant cards are displayed in a grid:

- Monstera Deliciosa**
Monstera Deliciosa
Lumière indirecte | 1-2 fois/semaine
[Voir les détails](#)
- Sansevieria**
Sansevieria trifasciata
Faible à moyenne | 1 fois/2 semaines
[Voir les détails](#)
- Pothos Doré**
Epipremnum aureum
Lumière indirecte | 1 fois/semaine
[Voir les détails](#)
- Ficus Lyrata**
Ficus lyrata
Lumière vive indirecte | 1 fois/semaine
[Voir les détails](#)
- Zamioculcas**
Zamioculcas zamiifolia
Faible à moyenne | 1 fois/2-3 semaines
[Voir les détails](#)
- Philodendron**
Philodendron hederaceum
Lumière indirecte | 1 fois/semaine
[Voir les détails](#)

PlantHub
Découvrez, collectionnez et prenez soin de vos plantes favorites. Rejoignez notre communauté de passionné de botanique.

Liens rapides
[Ma collection](#)
[Guides de soins](#)
[Communauté](#)

Suivez-nous

© 2024 PlantHub. Tous droits réservés.
Mentions légales | Politique de confidentialité | Conditions d'utilisation

J'ai ensuite voulu remplir pré-remplir la deuxième user story en créant les fiches plantes, mais uniquement autour des informations génériques de la plante. Étant sur la page d'accueil, j'accède aux fiches plantes génériques, afin de pouvoir se renseigner sur les informations avant d'ajouter la plante à la collection. Pour la majorité du projet, j'ai jugé

DOSSIER PROFESSIONNEL (DP)

pertinent de travailler avec des modales plutôt qu'avec des pages pleines. Voici donc ma page avec la modale de la fiche plante ouverte :

The screenshot shows a website for "PlantHub". At the top, there's a navigation bar with "PlantHub", "Accueil", "Ma collection", a search bar ("Rechercher des plantes..."), and a "Connexion" button. The main background image is a dense forest of various plants. A central modal window is open, featuring a large image of a Sansevieria plant. The title "Découvrez le monde des plantes" is displayed above the modal. Inside the modal, the plant is identified as "Sansevieria" with its scientific name "Sansevieria trifasciata". Below this, there are sections for "Luminosité" (Lighting) with "Faible à moyenne" (Low to medium), "Arrosage" (Watering) with "1 fois/2 semaines" (Once/two weeks), and a "Description" section stating it's a succulent resistant to neglect and low light. There's also a "Soins" (Care) section mentioning to water only when dry and avoid stagnant water. A "Conseils" (Tips) section lists four points: perfect for beginners, resistant to forgetting, purifies air naturally, and tolerates low light. At the bottom of the modal are two buttons: "Ajouter à ma collection" (Add to my collection) and "Partager" (Share). To the left and right of the modal, there are smaller cards for other plants: "Monstera Deliciosa" and "Ficus Lyrata".

PlantHub

Découvrez, collectionnez et prenez soin de vos plantes favorites. Rejoignez notre communauté de passionnés de botanique.

Liens rapides

Ma collection
Guides de soins
Communauté

Suivez-nous



© 2024 PlantHub. Tous droits réservés.

Mentions légales Politique de confidentialité Conditions d'utilisation

Suite à cette modale, j'ai maquetté la modale de connexion et création de compte, afin de permettre à l'utilisateur d'ajouter la plante à sa collection :

DOSSIER PROFESSIONNEL (DP)

PlantHub Accueil Ma collection

Rechercher des plantes... Connexion

Découvrez le monde des plantes

Collectionnez, apprenez et prenez soin de vos plantes favorites. Rejoignez notre communauté de passionnés de botanique.

Commencer ma collection Explore les plantes

Se connecter/Créer un compte

Email

Mot de passe

[Se connecter/Créer un compte](#)

Monstera Deliciosa
Monstera Deliciosa
Lumière indirecte 1-2 fois/semaine
[Voir les détails](#)

Ficus Lyrata
Ficus lyrata
Lumière vive indirecte 1 fois/semaine
[Voir les détails](#)

Zamioculcas
Zamioculcas zamiifolia
Facile à moyen 1 fois/2-3 semaines
[Voir les détails](#)

Philodendron
Philodendron hederaceum
Lumière indirecte 1 fois/semaine
[Voir les détails](#)

Follow us: [Instagram](#) [Twitter](#) [Facebook](#)

Liens rapides: Ma collection, Guides de soins, Communauté

Suivez-nous: [Instagram](#) [Twitter](#) [Facebook](#)

© 2024 PlantHub. Tous droits réservés.

Mentions légales Politique de confidentialité Conditions d'utilisation

J'ai choisi de garder ce même design de modale pour l'entièreté du projet afin de garder une cohérence graphique. Une fois l'utilisateur connecté, il est renvoyé sur la page d'accueil avec un header modifié, lui permettant de recevoir et de consulter ses notifications. Voici la page d'accueil une fois l'utilisateur connecté :

DOSSIER PROFESSIONNEL (DP)

PlantHub Accueil Ma collection

Rechercher des plantes...   

Découvrez le monde des plantes

Collectionnez, apprenez et prenez soin de vos plantes favorites. Rejoignez notre communauté de passionnés de botanique.

[Commencer ma collection](#) [Explore les plantes](#)

Plantes suggérées pour vous

Découvrez notre sélection de plantes parfaites pour débuter votre collection



Monstera Deliciosa
Monstera Deliciosa
Lumière indirecte  1-2 fois/semaine
[Voir les détails](#)



Sansevieria
Sansevieria trifasciata
Faible à moyenne  1 fois/2 semaines
[Voir les détails](#)



Pothos Doré
Epipremnum aureum
Lumière indirecte  1 fois/semaine
[Voir les détails](#)



Ficus Lyrata
Ficus lyrata
Lumière vive indirecte  1 fois/semaine
[Voir les détails](#)



Zamioculcas
Zamioculcas zamiifolia
Faible à moyenne  1 fois/2-3 semaines
[Voir les détails](#)



Philodendron
Philodendron hederaceum
Lumière indirecte  1 fois/semaine
[Voir les détails](#)

PlantHub

Découvrez, collectionnez et prenez soin de vos plantes favorites. Rejoignez notre communauté de passionnés de botanique.

Liens rapides

[Ma collection](#) [Guides de soins](#) [Communauté](#)

Suivez-nous

Mentions légales | Politique de confidentialité | Conditions d'utilisation

Une fois l'utilisateur connecté, il peut ouvrir la fiche de n'importe quelle plante, ce qui ouvrira la même modale que sur la page d'accueil non connecté. En cliquant sur « Ajouter à ma collection » ou sur le bouton « Commencer ma collection » de la page d'accueil, une nouvelle modale s'ouvre, permettant de renseigner le nom de la plante, l'espèce, la date d'acquisition ainsi qu'ajouter ou non une photo de la plante, validant la user story numéro 1 :

DOSSIER PROFESSIONNEL-Version du 01/06/2016

Page 47

DOSSIER PROFESSIONNEL (DP)

PlantHub Accueil Ma collection

Rechercher des plantes...

Découvrez le monde des plantes

Collectionnez, apprenez et prenez soin de vos plantes favorites. Rejoignez notre communauté de passionnés de botanique.

Ajouter à la collection

Nom de la plante
Sansevieria

Espèce
Sansevieria trifasciata

Date d'acquisition
24 / 07 / 2025

Photographie de la plante

Monstera Deliciosa

Monstera Deliciosa
Lumière indirecte 1-2 h
[Voir les détails](#)

Sansevieria trifasciata

Ficus Lyrata

Ficus lyrata
Lumière vive indirecte 1 h
[Voir les détails](#)

 Ajouter une photo
Si vous n'ajoutez pas de photo, l'image par défaut sera utilisée

[Ajouter à ma collection](#)

1 fois/semaine
[Voir les détails](#)

1 fois/semaine
[Voir les détails](#)

1 fois/semaine
[Voir les détails](#)

PlantHub

Découvrez, collectionnez et prenez soin de vos plantes favorites. Rejoignez notre communauté de passionnés de botanique.

Liens rapides

[Ma collection](#) [Guides de soins](#) [Communauté](#)

Suivez-nous

Mentions légales | Politique de confidentialité | Conditions d'utilisation

Une fois la plante ajoutée, la modale se ferme et une notification de validation s'ouvre en haut à droite. L'utilisateur peut: soit fermer la fenêtre de notification, soit cliquer sur le bouton « Ma collection » pour y accéder.

DOSSIER PROFESSIONNEL (DP)

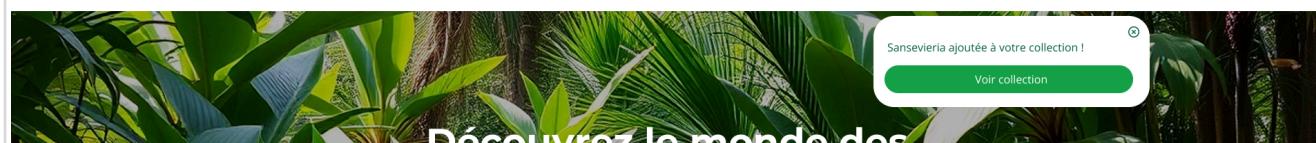
PlantHub Accueil Ma collection

Rechercher des plantes...



Sansevieria ajoutée à votre collection !

Voir collection



Dans le cas où l'utilisateur ferme cette modale et ouvre la modale de notifications, il peut accéder à toutes ses notifications, validant la user story numéro 4 :

PlantHub Accueil Ma collection

Rechercher des plantes...



Il est l'heure d'arroser votre Sansevieria ! 19:00

Ajoutez du fertilisant à votre

Sansevieria !

10:00



L'utilisateur peut ensuite se rendre sur la page « Ma collection » en cliquant sur le lien dédié dans la navigation du header. En arrivant sur cette page, il accède à toutes les plantes de sa collection, qu'il peut filtrer ou bien trier, validant les user stories 7 et 8 :

PlantHub Accueil Ma collection

Rechercher des plantes...



Vos plantes

Retrouvez toutes vos plantes ainsi que leurs suivis

Filtrer Trier



Monstera Deliciosa

Monstera Deliciosa

Lumière indirecte 1-2 fois/semaine

Voir les détails



Sansevieria

Sansevieria trifasciata

Faible à moyenne 1 fois/2 semaines

Voir les détails



Pothos Doré

Epipremnum aureum

Lumière indirecte 1 fois/semaine

Voir les détails



Ficus Lyrata

Ficus lyrata

Lumière vive indirecte 1 fois/semaine

Voir les détails



Zamioculcas

Zamioculcas zamiifolia

Faible à moyenne 1 fois/2-3 semaines

Voir les détails



Philodendron

Philodendron hederaceum

Lumière indirecte 1 fois/semaine

Voir les détails

PlantHub

Découvrez, collectionnez et prenez soin de vos plantes favorites. Rejoignez notre communauté de passionné de botanique.

Liens rapides

Ma collection

Guides de soins

Communauté

Suivez-nous



© 2024 PlantHub. Tous droits réservés.

Mentions légales Politique de confidentialité Conditions d'utilisation

Une fois sur cette page, l'utilisateur peut choisir d'ouvrir une fiche plante afin de voir les détails d'entretien passé, validant la user story numéro 2. L'utilisateur peut également modifier des informations en cas d'erreur ou bien supprimer la plante si il ne l'a plus en sa possession, validant la user story numéro 3 :

DOSSIER PROFESSIONNEL (DP)

PlantHub Accueil Ma collection

Rechercher des plantes...

Plantes suggérées pour vous

Monstera Deliciosa
Monstera Deliciosa
Lumière indirecte 1-2

Sansevieria
Sansevieria trifasciata

Historique de l'entretien

- Arrosage - 24/07/25 10:00
- Ajout d'engrais - 22/07/25 18:00
- Coupe des tiges encombrantes - 21/07/25 15:00
- Arrosage - 21/07/25 11:00

Voir plus

Informations de croissance

- Taille : 73cm
- Age : 264 jours
- Taux de croissance : 0.2cm/semaine

Modifier Modifier Modifier

Ajouter des informations Supprimer de ma collection

Paramètres de notifications

1 fois/semaine
les détails

1 fois/semaine
les détails

PlantHub
Découvrez, collectionnez et partagez vos plantes avec la communauté de passionnés de botanique.

© 2024 PlantHub. Tous droits réservés.

Mentions légales Politique de confidentialité Conditions d'utilisation

Guides de soins Communauté

f

Une fois sur cette modale, l'utilisateur a plusieurs options. Pour le moment, l'utilisateur va régler ses paramètres de notifications, validant ainsi la user story numéro 5 :

DOSSIER PROFESSIONNEL (DP)

The screenshot shows the PlantHub mobile application interface. At the top, there's a navigation bar with 'PlantHub' and 'Accueil' (Home) on the left, and a search bar with 'Rechercher des plantes...' (Search for plants...) and a user icon on the right. Below the navigation is a section titled 'Plantes suggérées pour vous' (Plants suggested for you) with a subtitle 'Découvrez notre sélection de plantes parfaites pour débuter votre collection' (Discover our selection of perfect plants to start your collection). This section displays cards for two plants: 'Monstera Deliciosa' and 'Ficus Lyrata'. A large modal window is centered on the screen, titled 'Paramètres de notifications' (Notification settings). It contains three input fields: 'Fréquence (par jour)' (Frequency (per day)) with the value '2', 'Heures (séparer par une virgule si plusieurs)' (Hours (separate by a comma if multiple)) with the value '10:00, 19:00', and 'Texte de la notification' (Notification text) with the value 'C'est l'heure d'arroser {plante} !' (It's time to water {plant}!). At the bottom of the modal is a green button labeled 'Ajouter des informations' (Add information).

PlantHub

Découvrez, collectionnez et prenez soin de vos plantes favorites. Rejoignez notre communauté de passionné de botanique.

Liens rapides

Ma collection
Guides de soins
Communauté

Suivez-nous



Mentions légales | Politique de confidentialité | Conditions d'utilisation

Une fois qu'il a réglé ses paramètres de notification, l'utilisateur peut revenir sur la fiche de la plante afin d'ajouter des informations concernant sa plante, notamment des photos d'évolution, validant ainsi la user story numéro 6 :

DOSSIER PROFESSIONNEL (DP)

PlantHub Accueil Ma collection

Rechercher des plantes...

Plantes suggérées pour vous

Ajouter des informations

Taille (cm)
73

Age (jours)
264

Date de mesure
24 / 07 / 2025

Photographie de la plante

Ajouter une photo

Monstera Deliciosa
Monstera Deliciosa
Lumière indirecte 1-2
[Voir les détails](#)

Ficus Lyrata
Ficus lyrata
Lumière vive indirecte 1 fo
[Voir les détails](#)

PlantHub
Découvrez, collectionnez et partagez avec la communauté de passionnés de plantes.

Ajouter des informations

Communauté

© 2024 PlantHub. Tous droits réservés.

Mentions légales Politique de confidentialité Conditions d'utilisation

Maintenant que l'utilisateur a ajouté des informations pour mieux suivre l'évolution de sa plante, il se demande ce qu'il a fait exactement à toutes ses plantes cette semaine. En cliquant sur le logo utilisateur en haut à droit de l'écran, il accède à son historique d'action, validant ainsi la user story numéro 9 :

DOSSIER PROFESSIONNEL (DP)

PlantHub Accueil Ma collection



Historique du compte

Retrouvez l'historique complet de vos actions sur votre plante

← 1 / ... →

PlantHub

Découvrez, collectionnez et prenez soin de vos plantes favorites. Rejoignez notre communauté de passionné de botanique.

Liens rapides

Suivez-nous



© 2024 PlantHub. Tous droits réservés.

[Mentions légales](#) [Politique de confidentialité](#) [Conditions d'utilisation](#)

Ainsi, nous remplissons toutes les user stories demandées, en gardant un design épuré et cohérent répondant aux besoins client.

2. Précisez les moyens utilisés :

DOSSIER PROFESSIONNEL (DP)

Pour la création du design, je me suis servi de [ready.ai](#), qui m'a permis d'avoir un point de départ pour le reste de mes éléments. Pour les couleurs j'ai choisi ce qui me semblait cohérent pour un projet traitant de la nature. Pour les choix personnels de design (bouton, modale, historique etc...), je me suis inspiré de projets précédents sur lesquels j'ai pu travailler et de mon ressenti personnel.

3. Avec qui avez-vous travaillé ?

J'ai travaillé en totale autonomie sur ce projet.

4. Contexte

Nom de l'entreprise, organisme ou association - La plateforme

Chantier, atelier, service - Formation Développeur Web & Web Mobile

Période d'exercice - Du 23/07/2025 au 25/07/2025

5. Informations complémentaires (*facultatif*)

DOSSIER PROFESSIONNEL (DP)

Cliquez ici pour taper du texte.

DOSSIER PROFESSIONNEL (DP)

Activité-type 1 Développer la partie front-end d'une application web ou web mobile sécurisée

Exemple n°3 - Réaliser des interfaces utilisateurs statiques web ou web mobile

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Il m'a été demandé de réaliser un site vitrine statique présentant mon profil professionnel, mes compétences, mes projets ainsi que mes coordonnées. Le site devait être conçu uniquement grâce à l'utilisation de HTML5 et CSS3. L'utilisation de JavaScript ou de frameworks et bibliothèques externes était interdite.

Un design était fourni au travers d'un lien figma. Voici la version desktop du design :

DOSSIER PROFESSIONNEL (DP)

Sophie Dubois

Accueil À propos Projets Contact

Sophie Dubois

Développeuse Web Front-End & Designer UI/UX

Je crée des expériences web élégantes, intuitives et performantes avec une attention particulière aux détails.

Voir mes projets



À propos de moi

Mon parcours

Diplômée en développement web et design d'interface, je combine des compétences techniques solides avec une sensibilité esthétique pour créer des sites web et applications qui sont à la fois fonctionnels et visuellement attrayants.

Après 3 ans d'expérience en agence digitale, j'ai développé une expertise dans la création d'interfaces utilisateur intuitives et accessibles, en privilégiant toujours l'expérience utilisateur.

Je suis passionnée par les technologies web modernes et je m'efforce constamment d'améliorer mes compétences pour rester à la pointe des dernières tendances et meilleures pratiques.

En savoir plus sur mon parcours →

Mes compétences



HTML5



CSS3



JavaScript



React



UI/UX Design



Responsive

Mes projets récents

Découvrez une sélection de mes travaux récents qui démontrent mes compétences en développement web et design d'interface.

E-commerce Élégance
Conception et développement d'une boutique en ligne pour une marque de vêtements haut de gamme, avec une interface élégante et une expérience d'achat fluide.

HTML5 CSS3 JavaScript Responsive

[Voir le projet →](#)

Dashboard Analytics Pro
Interface de tableau de bord analytique pour une entreprise SaaS, permettant de visualiser et d'analyser les données clients de manière intuitive.

React CSS3 Data Viz UX Design

[Voir le projet →](#)

[Voir tous les projets](#)

DOSSIER PROFESSIONNEL (DP)

Il était demandé de remplacer les informations présentes sur le design par nos informations personnelles avec autorisation de modifier les typographies et couleurs du projet.

À partir du design fourni, je devais également produire les pages "À propos", "Projets" et "Contact" en gardant le header et footer du design original.

Les demandes pour la page d'accueil étaient les suivantes :

- Section d'introduction : Présentez-vous brièvement.
- Section "À propos" : Résumez votre parcours et vos compétences principales.
- Section "Projets": Mettez en avant 2 à 4 projets réalisés, avec pour chacun une image, un titre et une courte description.
- Section "Contact": Affichez un formulaire statique (non fonctionnel) avec les champs suivants : nom, email, message, case à cocher RGPD.
- Pied de page (footer) : Ajoutez des liens vers vos réseaux sociaux.

Les demandes pour les pages indépendantes :

- about.html : Page dédiée "À propos" avec votre photo et une description détaillée de votre parcours.
- projects.html : Page dédiée "Projets" listant au moins deux projets, chacun illustré et décrit.
- contact.html : Page dédiée "Contact" avec le formulaire statique complet.

Il était également demandé un respect de la sémantique HTML ainsi que de la hiérarchie des titrages et des contenus.

Enfin, il y avait des consignes concernant le design et l'accessibilité. Le site final se devait d'être responsive pour toutes les tailles d'écran à partir de 320px. Je devais utiliser des couleurs cohérentes et des polices d'écriture modernes. Il fallait introduire des effets d'interactivité sur les boutons et les liens au survol. Il était également nécessaire de commenter le code CSS pour expliquer les sections et styles.

Une fois la connaissance des informations nécessaires acquise, j'ai pu commencer à travailler sur un projet vierge. J'ai créé un dossier vierge nommé "Portfolio-CP3-main" dans lequel j'ai ajouté un fichier "index.html", un dossier "css" pour ma feuille de style et un dossier "img" pour mes images. J'ai ensuite créé un nouveau repository github nommé "[Portfolio-CP3](#)" où j'ai initialisé mon projet pour avoir un suivi de mon évolution et sauvegarder régulièrement mon projet pour ne pas risquer de le perdre. J'ai également initialisé github pages pour mon projet, me permettant de rendre la dernière version push disponible directement en version site comme demandé dans les consignes.

Avant de commencer le développement HTML, je me rends dans mon dossier css et je crée un document "style.css" dans lequel je prépare mon document. Pour ça, j'importe mes polices d'écriture depuis Google Fonts (ici Sora et Carlito), j'uniformise mon document, je prépare des variables de couleurs et de police pour mes différents styles et je prépare les styles de titres, paragraphes etc... Je crée également des classes pour certains styles réutilisables comme les boutons, les tags et les conteneurs de contenu. Exemple de commentaire de mon fichier avec l'import des polices, l'uniformisation du document et mes variables :

DOSSIER PROFESSIONNEL (DP)

```
1  /* Import fonts */
2
3  @import url("https://fonts.googleapis.com/css2?family=Sora:wght@100..800&display=swap");
4  @import url("https://fonts.googleapis.com/css2?family=Carlito:ital,wght@0,400;0,700;1,400;1,700&display=swap");
5
6  /* Uniformisation document */
7
8  * {
9    margin: 0;
10   transition: all 0.3s ease;
11   scroll-behavior: smooth;
12 }
13
14 :root {
15  /* Fonts */
16  --title: "Sora", sans-serif;
17  --text: "Carlito", sans-serif;
18
19  /* Colors */
20  --white: #ffffff;
21  --dark-text: #4b5563;
22  --light-grey: #f9fafb;
23  --grey: #d1d5db;
24  --dark-grey: #9ca3af;
25  --primary-color: #dd6e42;
26  --secondary-color: #2cda9d;
27 }
```

Pour ce qui est du style de mes boutons et de certains de mes liens, je règle leurs paddings (zone cliquable), leurs formes avec le border radius, leurs couleurs de texte et de fond et leurs effets de survol. Pour les liens “.link”, je règle la position en relative, car sur l’effet de hover, j’ai besoin de changer la position de l’icône présente dans le lien pour une animation. L’animation permettra de déplacer l’icône sur la droite pendant 0.5s et qui maintiendra l’icône à cette position tant que le hover est maintenu grâce au animation-fill-mode en forwards. L’effet de hover de la classe “.button” change simplement la taille en l’augmentant de 5%, et change le curseur de la souris en pointeur.

DOSSIER PROFESSIONNEL (DP)

```
94  .button {  
95    padding: 14px 32px;  
96    background: var(--secondary-color);  
97    color: var(--white);  
98    border-radius: 8px;  
99    display: block;  
100   width: fit-content;  
101  
102   &:hover {  
103     cursor: pointer;  
104     scale: 1.05;  
105   }  
106 }  
107  
108 .link {  
109   color: var(--secondary-color);  
110   position: relative;  
111  
112   &:hover {  
113     i {  
114       position: absolute;  
115       top: 3px;  
116       animation: 0.5s slideOut;  
117       animation-fill-mode: forwards;  
118     }  
119   }  
120 }
```

DOSSIER PROFESSIONNEL (DP)

```
146     /* Animations */
147
148     @keyframes slideOut {
149         0% {
150             right: -17px;
151         }
152         100% {
153             right: -117px;
154         }
155     }
```

Une fois cette étape passée, je me suis rendu sur mon fichier index.html pour démarrer le développement du projet. J'ai commencé par le doctype et le head de ma page. J'y nomme mon document et j'y lie ma feuille de style "style.css" et mon script Font Awesome. Font Awesome me permet d'intégrer des icônes et des logos dans mon code.

```
1  <!DOCTYPE html>
2  <html lang="fr">
3      <head>
4          <meta charset="UTF-8" />
5          <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6          <title>Accueil - Portfolio CP3</title>
7          <link rel="stylesheet" href="css/style.css" />
8          <script
9              src="https://kit.fontawesome.com/2bb90c57b5.js"
10             crossorigin="anonymous"
11         ></script>
12     </head>
```

Une fois le head terminé, j'ouvre la balise body et je prépare mes différentes parties en plaçant les balises header, puis main et enfin footer à l'intérieur du body. Je continue en reproduisant le design, en commençant par le header.

Pour mon header, je commence par délimiter une zone de contenu à l'aide d'une div, dans laquelle je mets mon logo cliquable qui mène à la page d'accueil et mon menu de navigation dans une balise nav, avec les liens vers mes futures pages indépendantes. J'y intègre également l'icône de "burgermenu" qui servira lors du responsive. Pour le moment, j'utilise la feuille de style pour le cacher.

DOSSIER PROFESSIONNEL (DP)

```
13    <body>
14        <header>
15            <div class="container-1540">
16                <a href="/" class="logo">Antoine Petit</a>
17
18                <nav>
19                    <ul>
20                        <li><a href="/">Accueil</a></li>
21                        <li><a href="./about.html">À propos</a></li>
22                        <li><a href="./projects.html">Projets</a></li>
23                        <li><a href="./contact.html">Contact</a></li>
24                    </ul>
25                </nav>
26
27                <i class="fa-solid fa-bars" id="burgermenubar"></i>
28            </div>
29        </header>
```

Une fois le header réalisé, je me rends à nouveau sur ma feuille de style pour le styliser en m'appuyant toujours sur le design Figma fourni. Je mets le background voulu au header, et je mets un display flex à ma div de contenu. Je répartis les éléments à l'opposé l'un de l'autre dans le conteneur et les aligne horizontalement. Je mets également un display flex à mon menu de navigation tout en retirant les icônes de liste. Ensuite je change la taille des liens de mon menu pour qu'ils fassent la taille de mon header, ça me sert à créer un effet de survol où tout l'arrière du header au niveau de mon lien change de couleur.

DOSSIER PROFESSIONNEL (DP)

```
196 header {  
197     background: var(--light-grey);  
198     box-shadow: 0px 1px 2px 0px #0000000d;  
199  
200     .container-1540 {  
201         display: flex;  
202         justify-content: space-between;  
203         align-items: center;  
204         padding: 15px;  
205  
206         .logo {  
207             color: var(--primary-color);  
208         }  
209  
210         nav {  
211             ul {  
212                 display: flex;  
213                 padding: 0;  
214                 list-style: none;  
215  
216                 li {  
217                     a {  
218                         padding: 20px;  
219                         font-weight: 500;  
220  
221                         &:hover {  
222                             background: var(--secondary-color);  
223                             color: var(--white);  
224                         }  
225                 }  
226             }  
227         }  
228     }  
229 }
```

DOSSIER PROFESSIONNEL (DP)

Une fois le style réalisé, le résultat se rapproche du design initial.

Antoine Petit

Accueil À propos Projets Contact

Avant de m'intéresser au développement du main, je préfère réaliser le footer. Mon footer est composé du même logo cliquable que dans le header, de logos cliquables menants sur mes réseaux sociaux, d'une courte description et enfin des mentions légales et copyright.

Je réalise mon footer d'une manière similaire au header avec une div pour le contenu. Je découpe mon contenu par ligne, la première contenant mon logo et mes réseaux sociaux, et la deuxième ligne contenant la description et les mentions légales. Il aurait été possible de découper également en colonnes mais j'ai choisi de le faire en lignes pour ce projet.

```
192    <footer>
193        <div class="container-1540">
194            <div>
195                <a href="/" class="logo">Antoine Petit</a>
196                <div id="socials">
197                    <a
198                        href="https://www.linkedin.com/in/antoine-petit-323587347/"
199                        target="_blank"
200                        ><i class="fa-brands fa-linkedin-in"></i>
201                    </a>
202                    <a href="https://github.com/AntooinePetit" target="_blank"
203                        ><i class="fa-brands fa-github"></i>
204                    </a>
205                    <a
206                        href="https://www.frontendmentor.io/profile/AntooinePetit"
207                        target="_blank"
208                        ><i class="fa-brands fa-dev"></i>
209                    </a>
210                </div>
211            </div>
212            <div>
213                <p>Développeur Web Front-End & Designer UI/UX</p>
214                <p>© 2025 Antoine Petit. Tous droits réservés.</p>
215            </div>
216        </div>
217    </footer>
```

On peut maintenant passer au CSS. La div de conteneur est très utile ici car j'ai une couleur de fond que je veux sur toute la largeur de la page, mais je veux que mon contenu soit limité en largeur. J'utilise "> div" pour indiquer que le style ne concerne que les enfants directs de l'élément ".container-1540". J'utilise également "&:nth-child(2)" pour indiquer que le style concerne dans cette balise le deuxième enfant direct. Voici la feuille de style. Pour mes réseaux sociaux, je crée une bulle autour de mon icône en changeant le padding de mon lien et en stylisant l'icône elle-même. Au survol, mes icônes de réseaux sociaux augmentent leur taille de 10%.

DOSSIER PROFESSIONNEL (DP)

```
238 footer {
239   background: var(--primary-color);
240   padding: 50px 0;
241   .container-1540 {
242     .logo {
243       color: var(--white);
244     }
245     > div {
246       display: flex;
247       justify-content: space-between;
248       align-items: center;
249       #socials {
250         display: flex;
251         align-items: center;
252         a {
253           padding: 15px;
254           i {
255             font-size: 20px;
256             color: var(--white);
257             background: var(--dark-text);
258             padding: 15px 16px;
259             border-radius: 100000px;
260           }
261           &:nth-child(2) {
262             i {
263               padding: 15px;
264             }
265           }
266           &:hover {
267             scale: 1.1;
268           }
269         }
270       }
271       &:nth-child(2) {
272         p {
273           color: var(--white);
274           &:nth-child(2) {
275             font-size: 14px;
276           }
277         }
278       }
279     }
280   }
281 }
```

DOSSIER PROFESSIONNEL (DP)

Le résultat de ce code est le suivant :



Maintenant que notre header et notre footer sont finis, je peux désormais m'intéresser au main.

Pour le main, je découpe le contenu global en sections. J'en arrive aux sections "presentation", "about", "projects" et "contact".

```
31      <main>
32    >      <section id="presentation">...
45      </section>
46
47    >      <section id="about" class="container-1540">...
107    </section>
108
109   >      <section id="projects">...
158    </section>
159
160   >      <section id="contact" class="container-670">...
189    </section>
190      </main>
```

Je développe maintenant ma section "presentation" avec mes informations personnelles et une photo personnelle. Pour ce qui est de la sémantique, j'utilise un article pour regrouper les informations et liens qui seront sur la gauche de l'écran, puis je mets une balise img avec le lien de mon image présente dans mon dossier "img" pour la droite de l'écran. J'utilise une balise H1 pour définir le titre principal de ma page, des balises p pour mes paragraphes et une balise a pour mon lien. Ce lien est une ancre menant à la section "projects" sur la même page. J'inclue évidemment un texte alternatif clair à mon image pour des besoins d'accessibilité :

DOSSIER PROFESSIONNEL (DP)

```
32      <section id="presentation">
33          <div class="container-1540">
34              <article>
35                  <h1>Antoine Petit</h1>
36                  <p>Développeur Web Front-End & Designer UI/UX</p>
37                  <p>
38                      Je crée des expériences web élégantes, intuitives et performantes
39                      avec une attention particulière aux détails.
40                  </p>
41                  <a href="#projects" class="button">Voir mes projets</a>
42              </article>
43              
44          </div>
45      </section>
```

Pour le style, le fond contient une image légèrement obscurcie. Pour réaliser ce fond, je règle la hauteur de ma section à 100vh, soit la hauteur de l'écran de l'utilisateur car je veux que cette section soit entièrement affichée à l'ouverture du site quoi qu'il arrive. Ensuite, je change le fond en y mettant d'abord un dégradé de noir avec une opacité suffisamment faible pour faire un effet de filtre obscurcissant sur mon image, puis mon image pour la placer en dessous du filtre. Je centre mon fond et je fais en sorte que l'image couvre tout le fond :

```
#presentation {
    height: 100vh;
    padding: 0;
    background: linear-gradient(
        90deg,
        rgba(45, 48, 71, 0.9) 0%,
        rgba(45, 48, 71, 0.7) 50%,
        rgba(45, 48, 71, 0) 100%
    ),
    url(..../img/background-presentation.jpg);
    background-position: center;
    background-size: cover;
```

Ensuite, je mets un display grid à mon conteneur, afin de pouvoir placer mes deux éléments (article & img) de part et d'autre du conteneur, alignés au centre du fond :

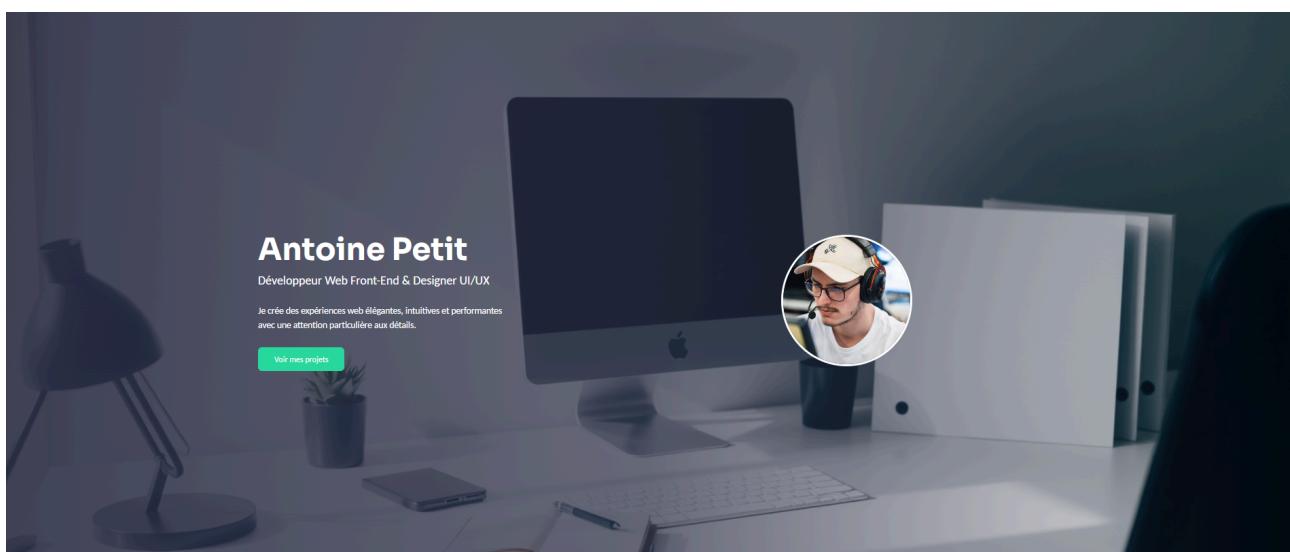
DOSSIER PROFESSIONNEL (DP)

```
306   .container-1540 {  
307     height: 100%;  
308     display: grid;  
309     grid-template-columns: 1fr 1fr;  
310     align-items: center;
```

Enfin, pour afficher mon image comme dans une bulle à contour blanc, je lui définis une taille carrée, un border radius infini pour assurer la forme parfaitement arrondie et enfin une bordure blanche. Je centre mon image dans sa moitié de la grille et pour qu'elle couvre toute la hauteur définie :

```
336   img {  
337     width: 250px;  
338     height: 250px;  
339     border-radius: 10000px;  
340     object-fit: cover;  
341     justify-self: center;  
342     border: 4px solid var(--white);  
343 }
```

Voilà le résultat pour cette section :



Je passe maintenant à la section “about”. Comme elle n'a pas un fond sur l'entièreté de la page, je lui attribue la classe de conteneur directement pour limiter la largeur du contenu. J'y inclus directement un H2 pour nommer la section, puis ouvre une div pour mes deux articles. Dans le premier article, je place un H3 pour nommer l'article,

DOSSIER PROFESSIONNEL (DP)

puis des balises p pour présenter mon parcours et enfin une balise a menant à la page “À propos” aussi accessible depuis le header :

```
47    <section id="about" class="container-1540">
48        <h2>À propos de moi</h2>
49        <div>
50            <article>
51                <h3>Mon parcours</h3>
52                <p>
53                    Étudiant en développement web et web mobile, je construis
54                    progressivement mon expertise en combinant rigueur technique et
55                    curiosité active. Après deux premières périodes de formation en
56                    2016 et 2019, j'ai repris le chemin du code avec une réelle
57                    volonté de professionnalisation.
58                </p>
59                <p>
60                    Je me forme en profondeur aux langages fondamentaux du web : HTML,
61                    CSS et JavaScript. Actuellement, je m'intéresse aux animations
62                    web, que j'explore comme un moyen d'apporter du dynamisme aux
63                    interfaces tout en préservant la performance.
64                </p>
65                <p>
66                    Orienté code avant tout, je cherche à développer des interfaces
67                    propres, maintenables et accessibles, avec un vrai souci de
68                    structure et de clarté. Mon objectif actuel est simple : obtenir
69                    mon diplôme tout en consolidant mes bases et en expérimentant dès
70                    que possible sur des projets personnels.
71                </p>
72                <a href="../about.html" class="link">
73                    En savoir plus sur mon parcours
74                    <i class="fa-solid fa-arrow-right"></i>
75                </a>
76            </article>
```

Pour le second article, je présente mes compétences. Pour ça, je mets d'abord un H3, puis une div dans laquelle je place mes compétences dans des span incluant une icône Font Awesome et une balise p incluant le nom de la compétence. Une bonne pratique aurait été d'utiliser des div à la place des span, et possiblement des H4 à la place des p.

DOSSIER PROFESSIONNEL (DP)

```
77      <article>
78          <h3>Mes compétences</h3>
79          <div>
80              <span>
81                  <i class="fa-brands fa-html5"></i>
82                  <p>HTML5</p>
83              </span>
84              <span>
85                  <i class="fa-brands fa-css3"></i>
86                  <p>CSS3</p>
87              </span>
88              <span>
89                  <i class="fa-brands fa-js"></i>
90                  <p>JavaScript</p>
91              </span>
92              <span>
93                  <i class="fa-brands fa-react"></i>
94                  <p>React</p>
95              </span>
96              <span>
97                  <i class="fa-regular fa-circle"></i>
98                  <p>UI/UX Design</p>
99              </span>
100             <span>
101                 <i class="fa-regular fa-circle"></i>
102                 <p>Responsive design</p>
103             </span>
104         </div>
105     </article>
```

Pour styliser, je commence par créer un espace entre le H2 et le reste de mon contenu pour alléger visuellement le site. Ensuite, de la même manière que la première section, je fais un display grid en colonnes pour répartir mes contenus de façon égale à gauche et à droite. J'établis un écart de 40px en cas de plus petit écran d'ordinateur pour que les contenus ne se collent pas. Je les centre horizontalement. Ensuite je règle la taille des mes titres et les écarts entre mes éléments avec du margin.

DOSSIER PROFESSIONNEL (DP)

```
347  /* À propos */
348
349  #about {
350    h2 {
351      margin-bottom: 25px;
352    }
353
354    div {
355      display: grid;
356      grid-template-columns: 1fr 1fr;
357      gap: 40px;
358      align-items: center;
359
360      article {
361        h3,
362        h2 {
363          font-size: 24px;
364          margin-bottom: 25px;
365        }
366
367        p {
368          margin-bottom: 25px;
369        }
370      }
```

Pour ce qui est du second article, j'ai opté pour un display flex en no wrap. Ainsi, lorsque le contenu est censé dépasser du conteneur, il revient automatiquement à la ligne, ce qui rend le responsive plus simple à réaliser. Je règle l'écart entre les éléments. Ensuite, j'aligne tout au centre de mes span, et je règle les flex pour avoir 3 éléments par ligne; je change la couleur, la taille de mes icônes et l'épaisseur de mes textes.

DOSSIER PROFESSIONNEL (DP)

```
371   div {  
372     display: flex;  
373     flex-wrap: wrap;  
374     gap: 20px;  
375   }  
376   span {  
377     justify-self: center;  
378     text-align: center;  
379     flex: 1 0 30%;  
380   }  
381   i {  
382     color: var(--primary-color);  
383     font-size: 50px;  
384     margin-bottom: 20px;  
385   }  
386   p {  
387     font-weight: 500;  
388   }  
389 }  
390 }  
391 }
```

Voilà le résultat :

À propos de moi

Mon parcours

Étudiant en développement web et web mobile, je construis progressivement mon expertise en combinant rigueur technique et curiosité active. Après deux premières périodes de formation en 2016 et 2019, j'ai repris le chemin du code avec une réelle volonté de professionnalisation.

Je me forme en profondeur aux langages fondamentaux du web : HTML, CSS et JavaScript. Actuellement, je m'intéresse aux animations web, que j'explore comme un moyen d'apporter du dynamisme aux interfaces tout en préservant la performance.

Orienté code avant tout, je cherche à développer des interfaces propres, maintenables et accessibles, avec un vrai souci de structure et de clarté. Mon objectif actuel est simple : obtenir mon diplôme tout en consolidant mes bases et en expérimentant dès que possible sur des projets personnels.

[En savoir plus sur mon parcours →](#)

Mes compétences

 HTML5	 CSS3	 JavaScript
 React	 UI/UX Design	 Responsive design

Voilà le lien vert avec l'effet de hover, la flèche est déplacée sur la droite comme prévu dans l'animation plus haut :

[En savoir plus sur mon parcours →](#)

DOSSIER PROFESSIONNEL (DP)

Pour ce qui est de la section “projects”, je commence par faire une div pour le contenu comme dans la première section, je nomme la section avec un H2 et un paragraphe d’introduction. Ensuite, j’ai une div dans laquelle je liste les projets, ici 2 projets. Chaque projet est un article, composé d’une image suivie d’une div contenant toutes les informations (H3, paragraphe de description, div contenant les tags correspondant au projet, lien permettant d’accéder au projet si existant). À la fin de la section, on retrouve un lien menant vers la page “Projets” contenant plus de projets.

```
109 <section id="projects">
110   <div class="container-1540">
111     <h2>Mes projets récents</h2>
112     <p>
113       Découvrez une sélection de mes travaux récents qui démontrent mes
114       compétences en développement web et design d'interface.
115     </p>
116     <div class="liste-projets">
117       <article>
118         
119         <div>
120           <h3>Vividly</h3>
121           <p>
122             Développement d'un site de recherche d'artisan de qualité,
123             avec expérience personnalisé selon la localisation
124           </p>
125           <div>
126             <p class="tag">HTML5</p>
127             <p class="tag">CSS3</p>
128             <p class="tag">Responsive design</p>
129             <p class="tag">Mobile first</p>
130           </div>
131           <a href="#" class="link"
132             >Voir le projet <i class="fa-solid fa-arrow-right"></i>
133           ></a>
134         </div>
135       </article>
136     <article> ...
137     </article>
138   </div>
139   <a href=".//projects.html" class="button">Voir tous les projets</a>
140 </div>
141 </section>
```

Pour le style de cette section, j’ai créé un style réutilisable pour l’affichage des projets, afin de pouvoir le modifier sur la page “Projets” sans avoir à le réécrire. Pour ce style réutilisable, je choisis un display grid avec un écart de 25px. Ensuite, je mets un background blanc à chaque article, ainsi qu’un léger border radius pour arrondir et rendre l’article plus joli, puis une ombre portée juste en dessous de l’article pour le faire mieux ressortir du fond. Je limite la taille de mon image et je choisis de n’afficher que le haut de l’image (donc le haut du projet). Je mets le même

DOSSIER PROFESSIONNEL (DP)

border radius au deux coins supérieurs de mon image pour qu'elle colle à l'article. Ensuite, je mets un padding à la div enfant directe pour que le contenu ne soit pas collé aux bords de l'image ou de l'article. Je crée un léger écart entre les éléments avec du margin bottom. Enfin, je mets un display flex à mes tags pour un affichage en ligne.

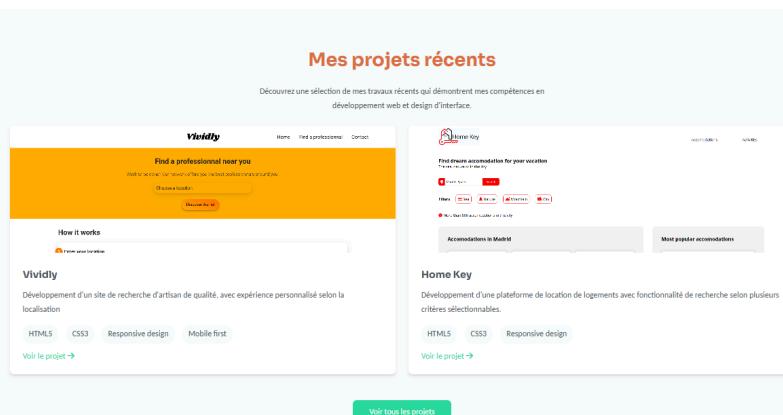
```
157 /* Eléments réutilisables */
158
159 .liste-projets {
160   display: grid;
161   gap: 25px;
162
163   article {
164     background: var(--white);
165     border-radius: 8px;
166     box-shadow: 0px 2px 4px -2px #0000001a, 0px 4px 6px -1px #0000001a;
167
168     img {
169       height: 250px;
170       object-fit: cover;
171       object-position: top;
172       border-top-left-radius: 8px;
173       border-top-right-radius: 8px;
174     }
175
176   > div {
177     padding: 25px;
178
179     h3,
180     > p,
181     div {
182       margin-bottom: 15px;
183     }
184
185     div {
186       display: flex;
187       gap: 15px;
188       flex-wrap: wrap;
189     }
190   }
191 }
192 }
```

Pour le style de la section en elle-même sur cette page, je mets un background à la section, je limite le contenu avec mon conteneur, et je règle la taille maximale du texte d'introduction en plus de l'aligner au mieux de l'écran. Je limite l'affichage de mes projets à 2 éléments par ligne et enfin je centre mon bouton menant à la page "Projets".

DOSSIER PROFESSIONNEL (DP)

```
396  /* Projets récents */  
397  
398  #projects {  
399    background: var(--light-grey);  
400  
401    .container-1540 {  
402      > p:nth-child(2) {  
403        text-align: center;  
404        max-width: 600px;  
405        margin: 25px auto;  
406      }  
407  
408      .liste-projets {  
409        grid-template-columns: 1fr 1fr;  
410      }  
411  
412      .button {  
413        margin: auto;  
414        margin-top: 50px;  
415      }  
416    }  
417  }
```

Voilà le résultat :



DOSSIER PROFESSIONNEL (DP)

Il ne nous reste plus qu'à réaliser la section "Contact" sur cette page. Je commence par un H2 comme à chaque début de section, un paragraphe d'introduction et enfin un formulaire. Pour le formulaire, le site étant statique, je ne remplis pas le "action" et je ne définis pas de "method" d'envoi du formulaire. Ensuite, pour faciliter le style, je regroupe tous les labels et leur input lié dans des span. Je lie les <label> et les <input>, en prenant le soin de mettre le bon type d'input à chaque fois (type email pour l'email, checkbox pour les RGPD et type submit pour l'envoi). Je n'oublie pas de mettre le "value" dans l'input submit pour afficher correctement le texte.

```
160    <section id="contact" class="container-670">
161        <h2>Contactez-moi</h2>
162        <p>
163            Vous avez un projet en tête ou une question ? N'hésitez pas à me
164            contacter, je vous répondrai dans les plus brefs délais.
165        </p>
166        <form action="#">
167            <span>
168                <label for="name">Nom</label>
169                <input type="text" name="name" id="name" />
170            </span>
171            <span>
172                <label for="email">Email</label>
173                <input type="email" name="email" id="email" />
174            </span>
175            <span>
176                <label for="message">Message</label>
177                <textarea name="message" id="message"></textarea>
178            </span>
179            <span>
180                <input type="checkbox" name="check" id="check" />
181                <label for="check">
182                    >J'accepte que mes données soient traitées conformément à la
183                    politique de confidentialité. Elles ne seront utilisées que pour
184                    répondre à ma demande.</label>
185                >
186            </span>
187            <input type="submit" value="Envoyer le message" class="button" />
188        </form>
189    </section>
```

Pour le style de la section, la section est limitée dans un plus petit conteneur. Je centre le texte d'introduction et mets une marge pour créer un espace entre "l'en-tête" de la section et le formulaire. Pour le formulaire, je mets un fond et un border radius pour faire ressortir le formulaire avec une forme douce. Je choisis un display grid en 2 colonnes avec 20px de séparation et 20px de padding pour créer une séparation avec le bord du formulaire. Pour les span, je mets un display flex en direction colonne pour que l'input prenne l'entièreté de la largeur disponible et je sépare le label et l'input de 10px. Je sélectionne le 3ème et 4ème span et je leur donne la taille complète de ma grille en précisant que ces deux spans démarrent au début de la première colonne et se terminent au début de la troisième colonne (donc la taille maximale car 2 colonnes). Pour les RGPD, je réduis légèrement la taille du texte mais le rend plus gras pour une meilleure mise en avant. Enfin je définis les styles de mes inputs et textarea.

DOSSIER PROFESSIONNEL (DP)

```
419  /* Contact */
420
421  #contact {
422    p {
423      text-align: center;
424      margin-top: 25px;
425      margin-bottom: 60px;
426    }
427
428    form {
429      background: var(--light-grey);
430      border-radius: 16px;
431      display: grid;
432      grid-template-columns: 1fr 1fr;
433      gap: 20px;
434      padding: 20px;
435    }
436
437    span {
438      display: flex;
439      flex-direction: column;
440      gap: 10px;
441
442      &:nth-child(3),
443      &:nth-child(4) {
444        grid-column: 1 / 3;
445      }
446
447      &:nth-child(4) {
448        flex-direction: row;
449        align-items: start;
450
451        label {
452          font-weight: 400;
453          font-size: 14px;
454        }
455      }
456
457      input,
458      textarea {
459        padding: 15px;
460        border-radius: 8px;
461        background: var(--white);
462        border: 1px solid var(--grey);
463        font-size: 16px;
464      }
465
466      textarea {
467        min-height: 146px;
468        resize: none;
469      }
470    }
```

DOSSIER PROFESSIONNEL (DP)

Voilà donc le résultat final :

Contactez-moi

Vous avez un projet en tête ou une question ? N'hésitez pas à me contacter, je vous répondrai dans les plus brefs délais.

Nom Email

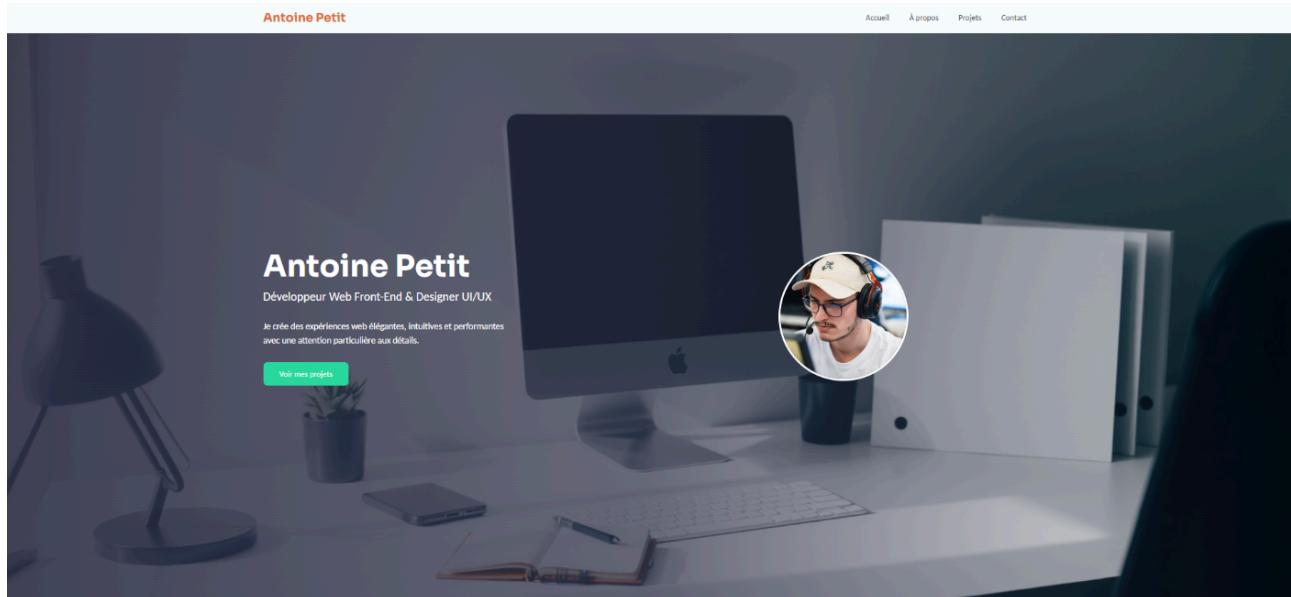
Message

J'accepte que mes données soient traitées conformément à la politique de confidentialité. Elles ne seront utilisées que pour répondre à ma demande.

Envoyer le message

Ceci complète donc la page index. La voici au complet :

DOSSIER PROFESSIONNEL (DP)



À propos de moi

Mon parcours

Étudiant en développement web et web mobile, je construis progressivement mon expertise en combinant l'aspect technique et curiosité active. Après deux premières périodes de formation en 2016 et 2019, j'ai repris le chemin du code avec une réelle volonté de professionnalisation.

Je me forme en profondeur aux langages fondamentaux du web : HTML, CSS et JavaScript. Actuellement, je m'intéresse aux animations web, que j'explique comme un moyen d'apporter de la dynamisme aux interfaces tout en préservant la performance.

Ordonné code avant tout, je cherche à développer des interfaces propres, maintenables et accessibles, avec un vrai souci de structure et de clarté. Mon objectif actuel est simple : obtenir mon diplôme tout en consolidant mes bases et en expérimentant dès que possible sur des projets personnels.

[En savoir plus sur mon parcours →](#)

Mes compétences



HTML5



CSS3



JavaScript



React



UI/UX Design



Responsive design

Mes projets récents

Découvrez une sélection de mes travaux récents qui démontrent mes compétences en développement web et design d'interface.

Two screenshots of recent projects. The first project, "Vividly", is a platform for finding professionals near you. The second project, "Vidive Key", is a platform for finding accommodations for your vacation. Both projects are built with HTML5, CSS3, and responsive design.

Contactez-moi

Vous avez un projet en tête ou une question ? N'hésitez pas à me contacter, je vous répondrai dans les plus brefs délais.

Nom

Email

Message

J'accepte que mes données soient traitées conformément à la politique de confidentialité. Elles ne seront utilisées que pour répondre à ma demande.

DOSSIER PROFESSIONNEL (DP)

Pour les autres pages, le style ne change globalement que très peu, et j'ai regroupé tous les styles sur une seule feuille de style pour plus facilement récupérer des éléments de style. La page "À propos" récupère le style de la deuxième section, en y ajoutant l'image de la première section. Voici le style et le résultat :

```
489  /* Page à propos */
490
491  #about-page {
492      #about {
493          div {
494              img {
495                  max-width: 400px;
496                  height: 400px;
497                  border-radius: 1000px;
498                  object-fit: cover;
499                  justify-self: center;
500              }
501
502          article:nth-child(1) {
503              grid-column: 1 / 3;
504
505              h3 {
506                  text-align: center;
507              }
508          }
509      }
510  }
511 }
```

DOSSIER PROFESSIONNEL (DP)

Antoine Petit

Accueil À propos Projets Contact

À propos de moi

Mon parcours

Étudiant en développement web et web mobile, je poursuis actuellement une formation professionnalisante avec l'objectif clair de construire des solutions web efficaces, bien structurées et maintenables. Mon parcours mêle rigueur scientifique, passion du numérique et esprit compétitif, ce qui forge aujourd'hui ma façon d'aborder les défis du développement.

Titulaire d'un baccalauréat scientifique, spécialité Informatique et Sciences du Numérique, j'ai toujours eu un attrait pour les technologies, le raisonnement logique et la construction de systèmes fonctionnels. Cette base scientifique me permet d'aborder le code avec méthode et d'avoir une compréhension claire de ce que je développe.

Entre 2019 et 2022, j'ai été joueur compétiteur sur Rainbow Six Siege, un engagement qui m'a appris la gestion du stress, la communication en équipe, la recherche constante de performance, ainsi qu'un esprit d'analyse stratégique. Ces qualités sont aujourd'hui des atouts dans ma formation, où je m'efforce d'écrire un code propre, efficace et pensé pour des usages concrets.

De 2022 à 2024, j'ai traversé une période de réflexion professionnelle : un temps nécessaire pour me recentrer, explorer différentes voies et trouver une formation qui correspond à mes compétences, mes envies et ma vision du métier. C'est ainsi que j'ai intégré le parcours de développement web & web mobile que je suis actuellement avec détermination.

Je m'investis principalement dans l'apprentissage du HTML, CSS et JavaScript, avec un intérêt croissant pour les animations web et la création d'interfaces dynamiques. Je reste résolument orienté code, en accord avec ma volonté de bâtir des fondations techniques solides avant d'élargir mes compétences.

Côté personnel, je suis passionné par les jeux vidéo, l'esport et le sport automobile, autant de domaines qui cultivent chez moi la précision, la réactivité et le goût de la compétition. Le tennis, que je pratique également, m'apporte un équilibre entre concentration et endurance mentale, deux qualités précieuses dans le domaine du développement.

Mes compétences



HTML5



CSS3



JavaScript



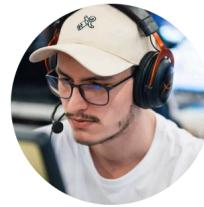
React



UI/UX Design



Responsive design



© 2025 Antoine Petit. Tous droits réservés.

La page "Projets" ne change que dans le nombre d'article par ligne :

```
477  /* Page projet */  
478  
479  #projects-page {  
480      #projects {  
481          .container-1540 {  
482              .liste-projets {  
483                  grid-template-columns: repeat(3, 1fr);  
484              }  
485          }  
486      }  
487  }
```

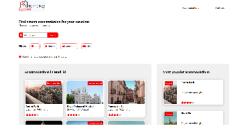
DOSSIER PROFESSIONNEL (DP)

Antoine Petit

Accueil À propos Projets Contact

Mes projets

Découvrez une sélection de mes travaux récents qui démontrent mes compétences en développement web et design d'interface.

 <p>Vividly Développement d'un site de recherche d'artisan de qualité, avec expérience personnalisée selon la localisation. HTML5 CSS3 Responsive design Mobile first Voir le projet →</p>	 <p>Home Key Développement d'une plateforme de location de logements avec fonctionnalité de recherche selon plusieurs critères sélectionnables. HTML5 CSS3 Responsive design Voir le projet →</p>	 <p>Ako Développement d'un site e-commerce de ventes de vêtements et accessoires adaptées à tous. HTML5 CSS3 Responsive design Voir le projet →</p>
 <p>Boum Boum Conception et développement d'un site vitrine pour une agence d'événements. HTML5 CSS3 Responsive design Voir le projet →</p>	 <p>Freshly Baked Goodness Développement d'un site vitrine d'une boutique de pâtisserie, avec présentation des produits et prix. HTML5 CSS3 Responsive design Voir le projet →</p>	 <p>Delights Développement d'un site vitrine d'une boutique de pâtisserie, avec présentation des produits et prix. HTML5 CSS3 Responsive design Voir le projet →</p>
 <p>Nomade Jewelry Développement d'un site e-commerce d'une bijouterie, avec présentation des produits. HTML5 CSS3 Responsive design Voir le projet →</p>	 <p>Digital Marketing Agency Conception d'un site vitrine d'une entreprise de développement web, avec informations de contact et présentation des partenaires. HTML5 CSS3 Responsive design Voir le projet →</p>	

Antoine Petit

Développeur Web Front-End & Designer UI/UX



© 2025 Antoine Petit. Tous droits réservés.

Enfin, la page contact est exactement la même que la section “Contact” de l’index :

DOSSIER PROFESSIONNEL (DP)

Antoine Petit

Accueil À propos Projets Contact

Contactez-moi

Vous avez un projet en tête ou une question ? N'hésitez pas à me contacter, je vous répondrai dans les plus brefs délais.

Nom

Email

Message

J'accepte que mes données soient traitées conformément à la politique de confidentialité. Elles ne seront utilisées que pour répondre à ma demande.

Antoine Petit

Développeur Web Front-End & Designer UI/UX



© 2025 Antoine Petit. Tous droits réservés.

Pour le responsive tablette, je fais un media query pour tous les écrans inférieurs à 1025px. Les seuls changements concernent la première section de la page index et la page "Projets". Ils réduisent chacun de 1 le nombre de

DOSSIER PROFESSIONNEL (DP)

colonnes, donc une seule colonne pour la page index et deux colonnes pour la page “Projets” :

```
513  /* Responsive tablette */
514
515  @media screen and (max-width: 1024px) {
516      /* À propos */
517      #about {
518          div {
519              grid-template-columns: 1fr;
520          }
521      }
522
523      /* Page projet */
524
525      #projects-page {
526          #projects {
527              .container-1540 {
528                  .liste-projets {
529                      grid-template-columns: repeat(2, 1fr);
530                  }
531              }
532          }
533      }
534 }
```

Pour le responsive mobile, je fais un media query pour tous les écrans inférieurs à 768px. Il y a du changement sur le header, le footer, l’animation, l’élément réutilisable d’affichage des projets, la première section de l’index, le formulaire de contact et enfin la page “À propos”.

Pour le header, je fais disparaître le menu de navigation pour afficher le burgermenu. La navigation n'est donc plus possible, et du JavaScript est nécessaire pour la rendre de nouveau fonctionnel.

DOSSIER PROFESSIONNEL (DP)

```
536  /* Responsive mobile */
537
538  @media screen and (max-width: 767px) {
539      /* Header */
540
541      header {
542          .container-1540 {
543              nav {
544                  display: none;
545              }
546
547              #burgermenu {
548                  display: block;
549              }
550          }
551      }
```

Antoine Petit



Pour le footer, je change la direction du flex en colonne pour que les éléments se superposent en une seule colonne.

DOSSIER PROFESSIONNEL (DP)

```
553     /* Footer */  
554  
555     footer {  
556         .container-1540 {  
557             > div {  
558                 flex-direction: column;  
559                 text-align: center;  
560             }  
561         }  
562     }
```

Antoine Petit



Développeur Web Front-End & Designer UI/UX

© 2025 Antoine Petit. Tous droits réservés.

Pour l'animation, je réduis simplement la distance de l'animation, sinon la flèche sort de l'écran sur les tailles d'écrans les plus petites.

DOSSIER PROFESSIONNEL (DP)

```
564     /* Animation */  
565  
566     @keyframes slideOut {  
567         0% {  
568             right: -17px;  
569         }  
570         100% {  
571             right: -47px;  
572         }  
573     }
```

Pour l'élément réutilisable, je réduis simplement le nombre de colonne à une seule pour toutes les pages. J'utilise “!important” pour forcer le style à tous les éléments concernés.

```
575     /* Eléments réutilisables */  
576  
577     .liste-projets {  
578         grid-template-columns: 1fr !important;  
579     }
```

Pour la première section de l'index, je réduis la hauteur pour simplement contenir le contenu (peu importe la hauteur de l'écran), je règle le padding pour avoir un peu d'espace en haut et en bas, je réduis le nombre de colonne à une seule. Enfin je mets des marges autour de mon image pour créer un espace autour.

DOSSIER PROFESSIONNEL (DP)

```
581     /* Présentation */  
582  
583     #presentation {  
584         height: fit-content;  
585         padding: 25px 0;  
586         .container-1540 {  
587             grid-template-columns: 1fr;  
588  
589             img {  
590                 margin: 20px 0;  
591             }  
592         }  
593     }
```

DOSSIER PROFESSIONNEL (DP)

Antoine Petit

Développeur Web Front-End & Designer UI/UX

Je crée des expériences web élégantes, intuitives et performantes avec une attention particulière aux détails.

[Voir mes projets](#)



Pour la section et la page “Contact”, je réduis le nombre de colonne du formulaire à une seule et je m’assure que tous les span sont bien dans la première colonne.

DOSSIER PROFESSIONNEL (DP)

```
595     /* Contact */  
596  
597     #contact {  
598         form {  
599             grid-template-columns: 1fr;  
600  
601             span {  
602                 grid-column: 1 !important;  
603             }  
604         }  
605     }
```

DOSSIER PROFESSIONNEL (DP)

Contactez-moi

Vous avez un projet en tête ou une question ? N'hésitez pas à me contacter, je vous répondrai dans les plus brefs délais.

Nom

Email

Message

J'accepte que mes données soient traitées conformément à la politique de confidentialité. Elles ne seront utilisées que pour répondre à ma demande.

[Envoyer le message](#)

Et enfin pour la page à propos, je réduis le nombre de colonne à une seule et m'assure que tous les éléments sont bien dans la première colonne.

DOSSIER PROFESSIONNEL (DP)

```
607  /* Page à propos */  
608  
609  #about-page {  
610      #about {  
611          div{  
612              grid-template-columns: 1fr;  
613  
614              article:nth-child(1){  
615                  grid-column: 1 !important;  
616              }  
617  
618              img{  
619                  grid-row: 2;  
620              }  
621          }  
622      }  
623  }  
624 }  
625 }
```

DOSSIER PROFESSIONNEL (DP)

Antoine Petit



À propos de moi

Mon parcours

Étudiant en développement web et web mobile, je poursuis actuellement une formation professionnalisaante avec l'objectif clair de construire des solutions web efficaces, bien structurées et maintenables. Mon parcours mêle rigueur scientifique, passion du numérique et esprit compétitif, ce qui forge aujourd'hui ma façon d'aborder les défis du développement.

Titulaire d'un baccalauréat scientifique, spécialité Informatique et Sciences du Numérique, j'ai toujours eu un attrait pour les technologies, le raisonnement logique et la construction de systèmes fonctionnels. Cette base scientifique me permet d'aborder le code avec méthode et d'avoir une compréhension claire de ce que je développe.

Entre 2019 et 2022, j'ai été joueur compétitif sur Rainbow Six Siege, un engagement qui m'a appris la gestion du stress, la communication en équipe, la recherche constante de performance, ainsi qu'un esprit d'analyse stratégique. Ces qualités sont aujourd'hui des atouts dans ma formation, où je m'efforce d'écrire un code propre, efficace et pensé pour des usages concrets.

De 2022 à 2024, j'ai traversé une période de réflexion professionnelle : un temps nécessaire pour me recentrer, explorer différentes voies et trouver une formation qui corresponde à mes compétences, mes envies et ma vision du métier. C'est ainsi que j'ai intégré le parcours de développement web & web mobile que je suis actuellement avec détermination.

Je m'investis principalement dans l'apprentissage du HTML, CSS et JavaScript, avec un intérêt croissant pour les animations web et la création d'interfaces dynamiques. Je reste résolument orienté code, en accord avec ma volonté de bâtir des fondations techniques solides avant d'élargir mes compétences.

Côté personnel, je suis passionné par les jeux vidéo, l'esport et le sport automobile, autant de domaines qui cultivent chez moi la précision, la réactivité et le goût de la compétition. Le tennis, que

DOSSIER PROFESSIONNEL (DP)

Voici donc comment j'ai répondu aux demandes et attentes des consignes qui m'ont été transmises.

2. Précisez les moyens utilisés :

Pour ce projet, j'ai utilisé la maquette Figma qui m'a été transmise afin de prendre connaissance du design et de ses spécificités ainsi que du responsive. La maquette m'a permis de définir plusieurs classes renouvelables comme les boutons, les liens et les conteneurs de différentes tailles. Pour le choix des couleurs et des polices, j'ai simplement suivi mes goûts personnels et mon ressenti. J'ai utilisé la bibliothèque d'icône Font Awesome pour les différentes icônes présentes sur mon projet. J'ai régulièrement push mes modifications sur Github afin d'avoir le meilleur suivi possible de l'évolution du projet, afin de pouvoir faire un rollback en cas de problème avec le projet. Le projet a été entièrement intégré en HTML et CSS.

3. Avec qui avez-vous travaillé ?

Pour ce projet, j'ai travaillé en totale autonomie.

4. Contexte

Nom de l'entreprise, organisme ou association - La plateforme

Chantier, atelier, service - Formation Développeur Web & Web Mobile
Période d'exercice - Du 22/05/2025 au 23/05/2025

5. Informations complémentaires (*facultatif*)

DOSSIER PROFESSIONNEL (DP)

Cliquez ici pour taper du texte.

DOSSIER PROFESSIONNEL (DP)

Activité-type 1 Développer la partie front-end d'une application web ou web mobile sécurisée

Exemple n°4 - Développer la partie dynamique des interfaces utilisateur web ou web mobile

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Pour ce projet, les consignes m'ont été fournies au travers du README lié à ce repository GitHub :

<https://github.com/JackAdamsJenkins/GeniArtHub>. Le repository contient également deux dossiers :

- Le dossier “back” contenant le back du projet. Il était interdit de manipuler les fichiers présents dans le dossier. Les deux seules actions que je devais réaliser était d’ouvrir un terminal avec le back (clic droit sur le dossier back -> ouvrir dans un terminal intégré) et de taper la commande “npm install” une fois puis “node server” à chaque session de travail sur le projet.
- Le dossier “front” contenant la page index et la page product toutes deux remplies avec des données de démonstration, une feuille de style remplies avec le style du header, footer, la page index et la page produit et enfin les images du projet (icônes et logo). Il contient également un dossier “js” avec 3 fichiers JavaScript vierge.

Les consignes demandent de :

- rendre la page d'accueil dynamique en chargeant les données produits grâce à une connexion au back
- rendre la page produit dynamique en chargeant les données d'un produit unique
- gérer l'ajout d'un produit au panier et le stockage du panier dans localstorage
- créer la page panier du projet, la stylisé et la rendre dynamique en chargeant les informations du panier
- permettre l'envoi du panier au back pour récupérer un numéro de commande généré par le back

Je commence donc par la page d'accueil, la plus simple. Je supprime les articles fictifs présents dans le code HTML en prenant soit d'en copier un et de le coller sur mon fichier JavaScript en guise de template pour plus tard.

```
front > js > JS index.js
  1  /* <article>
● 2   |   
  3   |   <a href="product.html">Buy bird</a>
  4   |   </article> */
  5   |
```

Ensuite, je commence ma connexion au back au travers d'une fonction asynchrone “getDatas”. Je fais un “try catch” pour gérer les erreurs, et je réalise un “fetch” des données grâce aux endpoints fournis dans le README du projet. Je ne précise pas la méthode pour le “fetch”, c'est en GET par défaut. Je mets un “await” devant mon “fetch” et devant le “req.json()” pour être sûr de bien renvoyer les données et non une promesse.

DOSSIER PROFESSIONNEL (DP)

```
front > js > JS index.js > ...
1 // Récupération des données globales de tous les produits
2 async function getDatas() : Promise<any> {
3   try {
4     const req : Response = await fetch(input: "http://localhost:3000/api/products/");
5     return await req.json();
6   } catch (e) {
7     console.error(e);
8   }
9 }
```

Ensuite, je crée une seconde fonction asynchrone “populateDatas” qui va me permettre de traiter et d'afficher les données reçues grâce à ma première fonction. Je récupère donc mes données en appelant la fonction “getDatas”, encore précédée d'un await. Une fois les données reçues, je sélectionne la section où je souhaite afficher mes données grâce à un “querySelector”. Je fais maintenant un “try catch” pour l'affichage des données. Si les données ont été reçues et existent, j'utilise la méthode “forEach” afin de réaliser le même affichage pour chaque produit récupéré automatiquement grâce au template récupéré plus tôt, et je l'insère dans le HTML avant la fin de la section. Si les données n'existent pas, à cause d'une erreur du back par exemple, j'affiche un message d'erreur à l'utilisateur indiquant qu'aucun produit n'a été trouvé. Enfin, j'appelle ma fonction “populateDatas” au chargement de la page.

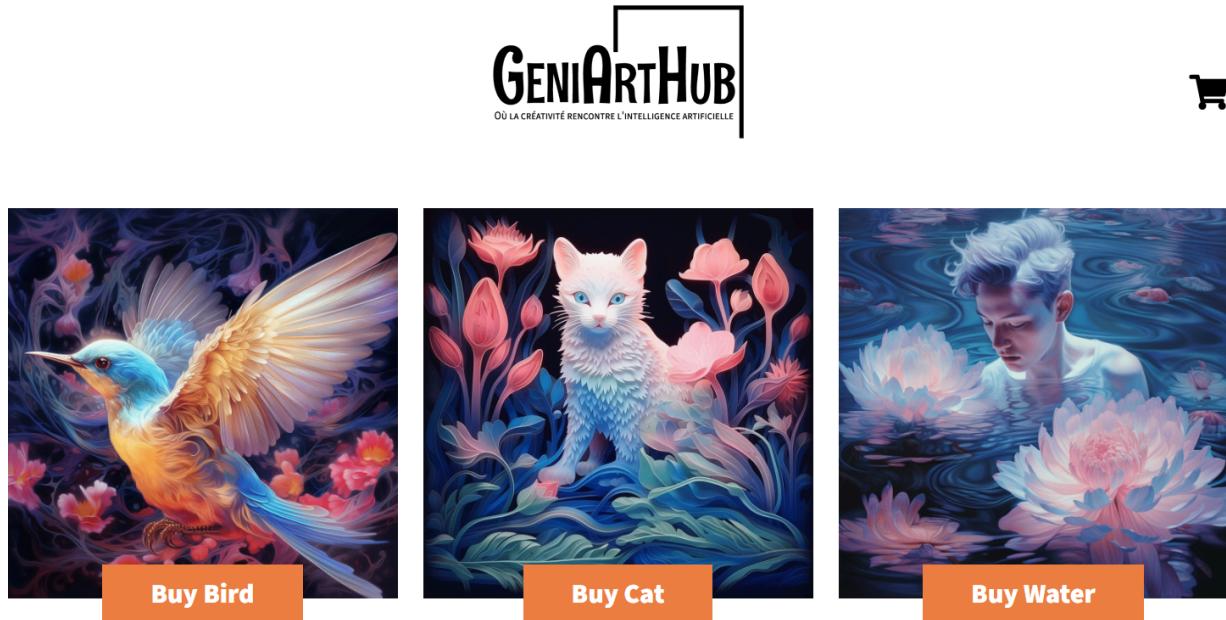
```
11 // Affichage des données sur la page      You, 4 months ago • feat(front): improve produc
12 async function populateDatas() : Promise<void> {
13   const datas : any = await getDatas();
14   const sectionProducts : Element | null = document.querySelector(selectors: ".products");
15   try {
16     datas.forEach((data : any) : void => {
17       const { _id, shorttitle, titre, image } = data;
18       const template = `
19         <article>
20           
21           <a href="product.html?id=${_id}">Buy ${shorttitle} </a>
22         </article>
23       `;
24       sectionProducts.insertAdjacentHTML(position: "beforeend", template);
25     });
26   } catch (e) {
27     console.error(e);
28     sectionProducts.innerHTML =
29       '<p class="error">Erreur ! Aucun article trouvé !</p>';
30   }
31 }
32
33 populateDatas();
```

On peut voir que j'utilise une déstructuration de données à la ligne 17 pour pouvoir manipuler les données plus facilement. Une alternative aurait été d'écrire “data.” devant chaque donnée à utiliser (ex. “data.image”, “data._id”). Dans la même logique, au lieu d'utiliser des backticks `` et d'insérer les données grâce à \${}, il aurait été

DOSSIER PROFESSIONNEL (DP)

possible d'utiliser la concaténation (ex. "<img src = ' " + image + " ' "). Je préfère la méthode avec les backticks car je la trouve plus lisible et l'insertion de données dans la chaîne de caractères est plus pratique.

Ainsi, je termine la page d'accueil et je valide l'affichage des données de manière dynamique.



Je peux maintenant réaliser l'affichage des données pour la page product. Pour ça, de la même façon que pour la page d'accueil, je récupère le code HTML de l'affichage de démonstration et le colle dans mon fichier JavaScript en guise de futur template.

```
front > js > JS product.js
1  /* <article>
2    <figure>
3      
4    </figure>
5    <div>
6      <h1>Éclat Éthéré : Bird</h1>
7      <p>Plongez dans l'univers mystique de 'Bird', une œuvre d'art captivante qui transcende les limites de la réalité. Réalisée dans le style éthéré et spectral, cette pièce évoque la présence d'un oiseau envoûtant qui semble flotter dans l'au-delà. </p>
8      <div class="price">
9        <p>Acheter pour</p>
10       <span class="showprice">35.25€</span>
11     </div>
12     <div class="declinaison">
13       <input type="number" name="quantity" id="quantity" placeholder="1" value="1" minlength="1">
14       <select name="format" id="format">
15         </select>
16     </div>
17     <a class="button-buy" href="#">Buy bird</a>
18   </div>
19 </article>
20
21 <aside>
22   <h2>Description de l'œuvre : Éclat Éthéré : Bird</h2>
23 </aside> */
```

Comme pour la page d'accueil, je crée une fonction "getDatas" mais cette fois, je lui attribue un paramètre "id". Ce paramètre me permettra de charger uniquement les données du produit voulu. J'utilise cet "id" dans mon fetch comme indiqué dans les endpoints du back.

DOSSIER PROFESSIONNEL (DP)

```
1 // Récupération des données spécifique du produit
2 async function getDatas(id : any) : Promise<any> {
3   try {
4     const req : Response = await fetch(input: `http://localhost:3000/api/products/${id}`);
5     return await req.json();
6   } catch (e) {
7     console.error(e);
8   }
9 }
```

Ensuite, encore une fois comme pour la page d'accueil, je crée une fonction "populateDatas". Pour commencer, je récupère l'id de mon produit dans l'URL grâce à "new URLSearchParams(document.location.search)" puis un "params.get("id")". Ces deux lignes me permettent d'abord de récupérer tous les paramètres présents dans l'URL de la page, puis de récupérer l'élément "id". Une fois l'id récupéré, je lance ma fonction "getDatas" avec l'id.

```
11 // Affichage des données du produit
12 async function populateDatas() : Promise<void> {
13   try {
14     const params = new URLSearchParams(document.location.search);
15     const productId : string | null = params.get(name: "id");
16     const datas : any = await getDatas(productId);
```

Je rédige directement le "catch" dans le cas d'une erreur de base de donnée ou bien d'une mauvaise URL qui mènerait à essayer de charger un article qui n'existe pas.

```
71 } catch (e) {
72   console.error(e);
73   document.querySelector(selectors: ".detailoeuvre").innerHTML =
74     '<p class="error">Erreur ! Aucun article trouvé !</p>';
75 }
76 }
```

Une fois la sécurité réalisée, je prépare l'affichage des données. Avec un querySelector, je sélectionne ma section d'affichage. Je réalise ensuite une nouvelle déstructuration de données. Ensuite, je prépare une version raccourcie de la description du produit grâce à "split("\n")". Après avoir analysé les données reçues, je me suis rendu compte que les différents paragraphes de la description sont séparés d'un "\n", je me serai donc de cette information et de la méthode "split" pour séparer ma description en un tableau contenant chaque paragraphe indépendamment. Je modifie ensuite le titre de l'onglet pour un affichage plus clair du produit choisi. J'utilise ensuite mon template dans lequel j'inclue toutes les informations. Enfin, je l'insère avant la fin de ma section (ici, étant donné qu'il n'y a qu'un seul élément dans la section, je peux utiliser "beforeend" ou "afterbegin" et j'obtiendrais le même résultat).

DOSSIER PROFESSIONNEL (DP)

```
17  const detailoeuvre : Element | null = document.querySelector(selectors: ".detailoeuvre");
18  const { _id, titre, description, image, declinaisons } = datas;
19  const shortdescription : any = description.split("\n");
20  document.title = `${titre} - GeniArtHub`;
21  const template = `<article>
22    <figure>
23      
24    </figure>
25    <div>
26      <h1>${titre}</h1>
27      <p>${shortdescription[0]}</p>
28      <div class="price">
29        <p>Acheter pour</p>
30        <span class="showprice">${declinaisons[0].prix}€</span>
31      </div>
32      <div class="declinaison">
33        <input type="number" name="quantity" id="quantity" placeholder="1" value="1" minlength="1">
34        <select name="format" id="format">
35          </select>
36        </div>
37        <a class="button-buy" href="#">Buy bird</a>
38        <p style="color:red;" id="message-d-info"> </p>
39      </div>
40    </article>
41
42    <aside>
43      <h2>Description de l'oeuvre : ${titre} </h2>
44      <p> ${description} </p>
45    </aside>`;
46    detailoeuvre.insertAdjacentHTML(position: "beforeend", template);
```

DOSSIER PROFESSIONNEL (DP)



GENIARTHUB
OÙ LA CRÉATIVITÉ RENCONTRE L'INTELLIGENCE ARTIFICIELLE



Douce Énigme : Cat

Laissez-vous transporter dans l'univers fascinant de 'Cat', une œuvre d'art qui incarne la quintessence de l'éthétré. Ce tableau vous plonge dans un monde où la beauté des fleurs et des objets en 3D lumineux s'entrelace avec la réalité et le mystère. La représentation hyperréaliste de l'eau dans 'Cat' est tout simplement évoquante, capturant la lumière environnante de manière époustouflante. Les teintes lumineuses, vibrantes comme une symphonie visuelle, confèrent à cette œuvre une aura magique qui ne manquera pas de captiver votre regard. Chaque détail, minutieusement dessiné au crayon de couleur, apporte une dimension organique et vivante à l'ensemble. La forme organique au cœur de l'œuvre évoque des mystères et des réflexions, vous invitant à explorer ses contours et ses nuances. Les paysages lumineux de 'Cat' vous transportent vers des horizons inexplorés, où chaque recoin émet une lueur énigmatique. Cette œuvre, plus qu'un simple tableau, est une expérience artistique qui transcende la réalité et vous invite à plonger dans un monde de lumière et d'ombre. 'Douce Énigme : Cat' n'est pas seulement une œuvre à admirer, c'est une invitation à la contemplation et à la découverte. Offrez-vous cette pièce unique qui éveillera votre imagination et ajoutera une touche de mystère à votre espace de vie. Plongez dans l'inconnu avec 'Cat' et laissez-vous emporter par son charme insaisissable.

Acheter pour

9.99€

1

20 x 20



Buy bird

Description de l'œuvre : Douce Énigme : Cat

Laissez-vous transporter dans l'univers fascinant de 'Cat', une œuvre d'art qui incarne la quintessence de l'éthétré. Ce tableau vous plonge dans un monde où la beauté des fleurs et des objets en 3D lumineux s'entrelace avec la réalité et le mystère. La représentation hyperréaliste de l'eau dans 'Cat' est tout simplement évoquante, capturant la lumière environnante de manière époustouflante. Les teintes lumineuses, vibrantes comme une symphonie visuelle, confèrent à cette œuvre une aura magique qui ne manquera pas de captiver votre regard. Chaque détail, minutieusement dessiné au crayon de couleur, apporte une dimension organique et vivante à l'ensemble. La forme organique au cœur de l'œuvre évoque des mystères et des réflexions, vous invitant à explorer ses contours et ses nuances. Les paysages lumineux de 'Cat' vous transportent vers des horizons inexplorés, où chaque recoin émet une lueur énigmatique. Cette œuvre, plus qu'un simple tableau, est une expérience artistique qui transcende la réalité et vous invite à plonger dans un monde de lumière et d'ombre. 'Douce Énigme : Cat' n'est pas seulement une œuvre à admirer, c'est une invitation à la contemplation et à la découverte. Offrez-vous cette pièce unique qui éveillera votre imagination et ajoutera une touche de mystère à votre espace de vie. Plongez dans l'inconnu avec 'Cat' et laissez-vous emporter par son charme insaisissable.

Suite à ça, je m'occupe de l'affichage dynamique du prix en fonction de la taille sélectionnée par l'utilisateur. Pour ça, je commence par insérer les différentes tailles disponibles en tant qu'`<option>` dans mon `select` à ligne 34. Étant donné que j'ai récupéré les différentes déclinaisons dans ma déstructuration plus tôt, je fais un `forEach` pour ajouter chaque déclinaison. On peut voir que j'ai au préalable initialisé une variable "`i`" qui va me servir à définir la valeur de chacune des balises `<option>` pour mieux pouvoir manipuler les valeurs. Je n'oublie pas d'incrémenter la variable dans mon `forEach`.

Maintenant que mes `<option>` sont toutes implémentées, je peux réaliser l'affichage dynamique du prix. Je fais un "addEventListerner" sur mon élément `<select>` en "change". Cela signifie que dès que la valeur du `select` changera, la fonction sera appelée. Dans cette fonction, je récupère l'affichage du prix avec un autre "querySelector" et je

DOSSIER PROFESSIONNEL (DP)

change directement le contenu textuel avec “.textContent”. Ainsi, en cas de changement, le contenu prend automatiquement la valeur du prix lié à la déclinaison.

```
48 // Affichage dynamique des prix en fonction de la taille sélectionnée
49 const select : Element | null = document.querySelector(selectors: "#format");
50 let i : number = 0;
51 declinaisons.forEach((declinaison : any) : void => {
52     select.insertAdjacentHTML(
53         position: "beforeend",
54         string: `<option value="${i}"> ${declinaison.taille} </option>` 
55     );
56     i++;
57 });
58 select.addEventListener(type: "change", (e : Event) : void => {
59     document.querySelector(selectors: ".showprice").textContent = `${
60         declinaisons[select.value].prix
61     }€`;
62 });
```

Affichage du prix différent après le changement de déclinaison.



Maintenant, je dois gérer l'ajout de produit au panier. Je sais que je dois stocker mes produits sous la forme d'une liste d'objets dans le localstorage. Par conséquent, j'initialise une constante “panier” et je lui attribue, à l'aide d'une ternaire, soit la valeur de l'élément “panier” du local storage si il existe, soit un tableau vide si l'élément n'existe pas. Ainsi, je peux soit récupérer un panier existant, soit créer un nouveau panier à remplir.

```
126 // Récupération du panier dans le local storage s'il existe, sinon création d'un panier vide
127 const panier : any = localStorage.getItem(key: "panier")
128 ? JSON.parse(localStorage.getItem(key: "panier"))
129 : [];
```

Également, je dois m'assurer que l'utilisateur ne puisse pas avoir plus de 100 unités du même produit dans son panier, et qu'il ne puisse pas ajouter 0 unité ou un nombre négatif d'unité. Pour ça, je créer une fonction d'affichage d'erreur que je pourrais appeler si l'utilisateur tente ces actions. Cette fonction me permet également d'afficher le succès de l'ajout au panier si aucune erreur n'a été détectée lors de l'ajout. Ainsi, je définis en paramètre une variable “isError” à laquelle j'attribue la valeur “true” par défaut. Je sélectionne ensuite mon paragraphe d'affichage

DOSSIER PROFESSIONNEL (DP)

avec un `querySelector`. Maintenant je fais un “if / else” sur la valeur de “`isError`”, si elle est “true”, j’affiche le message d’erreur en rouge, sinon (donc si elle est “false”), j’affiche le message de succès en vert.

```
114 // Fonction d'affichage d'erreur
115 function showError(isError = true) : void{
116   const paragrapheErreur : Element | null = document.querySelector(selectors: '#message-d-info')
117   if(isError === true){
118     paragrapheErreur.innerHTML = "Vous ne pouvez avoir qu'entre 1 et 100 unités dans votre <a href='cart.html'>panier</a>."
119     paragrapheErreur.style.color = 'red'
120   } else {
121     paragrapheErreur.textContent = 'Article ajouté au panier'
122     paragrapheErreur.style.color = 'green'
123   }
124 }
```

Enfin, je fais ma fonction d’ajout au panier. Elle prend en paramètres l’ID unique du produit, la déclinaison choisie ainsi que la quantité voulue. Le prix n’est pas enregistré dans le localstorage pour des raisons de sécurité. Je commence par la sécurité sur la quantité: si elle est supérieure à 100 ou inférieure à 1, j’appelle ma fonction d’erreur et je mets fin à la fonction. Si la quantité est valide, je crée un identifiant unique à partir de l’ID du produit et de la valeur de “`declinaison`”, qui correspond à un nombre entier défini par “`i`” plus tôt. Ensuite, il m’est demandé de ne pas ajouter des duplicita de produits dans le panier. Par conséquent, je cherche si le produit existe déjà dans mon panier avec la méthode “`.find`” sur l’identifiant que je viens de créer. Si le produit existe, je le stocke dans une variable le temps de le manipuler. Je vérifie ensuite si le total d’unité, soit la quantité déjà présente dans le panier + la quantité que l’utilisateur souhaite ajouter, est bien compris entre 1 et 100 unités. Comme plus tôt, si cette valeur est hors des limites définies, j’appelle ma fonction d’erreur et mets fin à la fonction. Sinon, je définis la nouvelle quantité à mon élément et je sauvegarde le panier dans le localstorage avec “`localStorage.setItem`”, qui me permet d’enregistrer mon élément en tant que chaîne de caractères sous l’identifiant “`panier`”. J’appelle ensuite ma fonction d’affichage avec la valeur “`false`” pour donner un retour à l’utilisateur sur l’ajout du produit au panier. Ensuite, dans le cas où mon produit n’existe pas dans mon panier, je crée un objet avec les différentes données du produit, et je l’ajoute dans mon tableau panier avec la méthode “`.push`”. Je le sauvegarde dans le localstorage comme effectué plus tôt et j’affiche la confirmation à l’utilisateur.

DOSSIER PROFESSIONNEL (DP)

```
78 // Envoie des produits dans le panier
79 function addProductToCart(productId : any, declinaison : any, quantite : any) : void {
80     // Vérification des quantités
81     if(quantite > 100 || quantite < 1){
82         showError()
83         return
84     }
85     const idPanier = `${productId}-${declinaison}`;
86     // Si le produit existe déjà dans le panier, on met à jour sa quantité
87     if (panier.find((produit : any) : boolean => produit.idCart === idPanier)) {
88         const produitMisAJour : any = panier.find(
89             (produit : any) : boolean => produit.idCart === idPanier
90         );
91         // Vérification des quantités
92         if(parseInt(produitMisAJour.quantite) + parseInt(quantite) > 100 || parseInt(produitMisAJour.quantite) + parseInt(quantite) < 1){
93             showError()
94             return
95         }
96         produitMisAJour.quantite =
97             parseInt(produitMisAJour.quantite) + parseInt(quantite);
98         localStorage.setItem(key: "panier", JSON.stringify(panier));
99         showError(isError: false)
100        return;
101    }
102    // Sinon on ajoute le nouveau produit au panier
103    const nouveauProduit = {
104        idCart: idPanier,
105        id: productId,
106        taille: declinaison,
107        quantite,
108    };
109    panier.push(nouveauProduit);
110    localStorage.setItem(key: "panier", JSON.stringify(panier));
111    showError(isError: false)
112 }
```

Validation de l'ajout.



Message d'erreur.

DOSSIER PROFESSIONNEL (DP)

Acheter pour
11.99€

-5

30 x 20



Buy bird

Vous ne pouvez avoir qu'entre 1 et 100 unités dans votre [panier](#).

Le panier dans le localstorage.

```
panier [{"idCart":"def67890-0","id":"def67890","taille": "0", "quantite": "1"}, {"idCart": "def67890-0", "id": "def67890", "taille": "0", "quantite": "1"}, {"idCart": "def67890-1", "id": "def67890", "taille": "1", "quantite": 7}]
```

Je peux maintenant passer à la dernière page : la page panier. Pour cette page, aucun HTML ou CSS n'était fourni, uniquement un lien figma avec le design. J'ai reproduit le design en HTML/CSS classique. J'ai créé un espace d'affichage pour les produits avec le prix total de la commande et le nombre total de produits, ainsi qu'un formulaire de contact pour l'envoi des informations de livraisons. J'ai également préparé un template pour l'affichage des différents produits dans le panier.

Le design avec le panier vide.

DOSSIER PROFESSIONNEL (DP)

Votre panier

Votre panier est vide. Veuillez ajouter au moins un article à votre panier

Total de la commande

0 articles pour un montant de 0 €

Formulaire de commande

Prénom :

Nom :

Adresse :

Ville :

Email :

Passer commande

Le design avec des produits dans le panier.

DOSSIER PROFESSIONNEL (DP)

Votre panier



Sérénité Aquatique : Water

20 x 20

8.99€

Quantité :

[Supprimer](#)

Total de la commande

1 articles pour un montant de 8.99 €

Pour ce qui est du JavaScript, je vais présenter dans l'ordre logique d'utilisation des fonctions. Je commence par définir des variables, une variable "panier" et une variable "datas". Cette fois je définis "panier" en variable car je la modifie directement dans une des fonctions, et puis la variable "datas", je la défini en global pour qu'elle soit accessible dans toutes les fonctions directement.

```
243 // Déclaration des variables/constantes à portée globale
244
245 // Récupération du panier dans le local storage s'il existe, sinon création d'un panier vide
246 let panier : any = localStorage.getItem(key: "panier")
247   ? JSON.parse(localStorage.getItem(key: "panier"))
248   : [];
249
250 // On rend les datas accessible partout dans le document
251 let datas;
```

Ensuite, comme pour les autres pages, je prépare une fonction asynchrone pour récupérer les données de tous les produits. Récupérer tous les produits me permettra d'afficher les prix à partir des informations récupérées dans le panier du localstorage.

```
1 ▼ // Gestion des données et création de la page
2
3 // Récupération des données globales de tous les produits
4 ▼ async function getDatas(): Promise<any> {
5 ▼   try {
6     const req: Response = await fetch(input: "http://localhost:3000/api/products/");
7     return await req.json();
8   } catch (e) {
9     console.error(e);
10  }
11 }
```

Je fais une fonction "populateDatas" pour charger les données sur la page, comme pour les autres pages. Si mon panier est vide; ou qu'une erreur survient lors du chargement des données, j'affiche à l'utilisateur que le panier est vide.

DOSSIER PROFESSIONNEL (DP)

```
13 // Affichage des données du produit
14 async function populateDatas(): Promise<void> {
15   try {
16     datas = await getData();
17     if (panier.length === 0) {
18       document.querySelector(selectors: "#panier > div").insertAdjacentHTML(
19         position: "afterbegin",
20         string: `
21           <article>
22             <p>Votre panier est vide. Veuillez ajouter au moins un article à votre panier</p>
23           </article>
24         );
25       return;
26     }
27   } catch (e) {
28     console.error(e);
29     document.querySelector(selectors: "#panier > div").insertAdjacentHTML(
30       position: "afterbegin",
31       string: `
32         <article>
33           <p>Votre panier est vide. Veuillez ajouter au moins un article à votre panier</p>
34         </article>
35       );
36   }
37 }
```

Dans le cas où le panier contient au moins un élément, je réalise un “forEach” pour afficher chaque produit dans le panier. Je fais plusieurs déstructurations à la suite. Une première pour récupérer les données du produit dans le localstorage, donc l’ID unique du panier, l’ID unique du produit, la taille sélectionnée et la quantité de produit. Grâce à l’ID unique, je peux rechercher le produit parmi les données récupérées avec “getDatas”. Je fais ma seconde déstructuration pour récupérer le titre, l’image et les déclinaisons possibles du produit récupéré. Je stocke la taille choisie en utilisant la valeur de “taille” qui était dans mon localstorage, soit un entier, et je m’en sers comme clé pour sélectionner la bonne déclinaison. Finalement, je fais ma dernière déstructuration pour récupérer le prix dans la taille que j’ai choisie. Une fois toutes les données récupérées et stockées, je peux remplir mon template. Ensuite, je fais un “querySelector” sur ma div d’affichage des produits, et j’y insère le template rempli après le début de la div. Je sélectionne mon input, qui va me permettre de changer la quantité de produit voulu par l’utilisateur et de mettre automatiquement le localstorage. Je sélectionne également mon bouton de suppression pour pouvoir gérer la suppression d’un article. Enfin, je mets à jour le prix total et la nombre total de produit.

DOSSIER PROFESSIONNEL (DP)

```
27 panier.forEach((produit : any) : void => {
28     const { idCart, id, taille, quantite } = produit;
29     const produitRecherche : any = datas.find((prod : any) : boolean => prod._id === id);
30     const { titre, image, declinaisons } = produitRecherche;
31     const tailleChoisie : any = declinaisons[taille];
32     const { prix } = tailleChoisie;
33     const template = `<article id="${idCart}">
34         
35         <h2>${titre}</h2>
36         <p>${tailleChoisie.taille}</p>
37         <p>${prix}</p>
38         <div class="input-container">
39             <label for="quantity-${idCart}">Quantité : </label>
40             <input type="number" name="quantity-${idCart}" id="quantity-${idCart}" placeholder="${quantite}" value="${quantite}" minlength="1">
41         </div>
42         <button id="delete-${idCart}">Supprimer</button>
43     </article>
44     <p style="color:red;" id="erreur-quantite-${idCart}" class="erreur-quantite"></p>`;
45     document
46         .querySelector(selectors: "#panier > div")
47         .insertAdjacentHTML(position: "afterbegin", template);
48     const input : HTMLInputElement | null = document.querySelector(selectors: "input");
49     input.addEventListener(type: "change", (e : Event) : void => {
50         updateCart(idCart, input.value, input);
51     });
52
53     // Ecouteur d'évènement du bouton 'Supprimer'
54     const boutonDelete : Element | null = document.querySelector(selectors: `#delete-${idCart}`);
55     boutonDelete.addEventListener(type: "click", (e : Event) : void => {
56         supprimerArticle(idCart);
57     });
58 );
59 calcTotals();
```

Je vais maintenant présenter les fonctions appelées dans mon “populateDatas”. Je commence avec la fonction “updateCart”. Cette fonction prend en paramètres l’ID unique généré pour le panier, la quantité de produit, ainsi que l’input du produit. Je récupère le produit à modifier dans le panier grâce à l’ID envoyé lors de l’appel de la fonction. Je sélectionne ensuite le paragraphe d’affichage d’erreur avec un querySelector. Je vérifie ensuite si la quantité est bien comprise entre 1 et 100 grâce à un “if / else if / else”. Ici, SI la quantité est inférieure à 0 ou supérieure à 100, j’affiche un message d’erreur, et grâce à une ternaire, je limite la valeur affichée. Si la valeur entrée est supérieure à 100, j’affiche 100 quoi qu’il arrive, et si elle est inférieure à 0, j’affiche 0. Ensuite, SINON SI la valeur entrée est 0, j’appelle une fonction pour supprimer l’article du panier. SINON, les quantités sont donc valides, je modifie la valeur de “quantite” dans le localstorage et j’enregistre le panier mis à jour. Je m’assure également que le message d’erreur est bien vide. Enfin, je calcule le nouveau total.

```
72 // Gestion dynamique du panier et de l'affichage
73
74 // Mise à jour du panier
75 function updateCart(id : any, quantite : any, input : any) : void {
76     const produitAModifier : any = panier.find((produit : any) : boolean => produit.idCart === id);
77     const erreur : Element | null = document.querySelector(selectors: `#erreur-quantite-${id}`);
78     if (quantite < 0 || quantite > 100) {
79         erreur.textContent = "Vous devez avoir entre 1 et 100 articles";
80         input.value = quantite > 100 ? 100 : 0;
81     } else if (quantite == 0) {
82         supprimerArticle(id);
83     } else {
84         produitAModifier.quantite = quantite;
85         localStorage.setItem(key: "panier", JSON.stringify(panier));
86         erreur.textContent = "";
87     }
88     calcTotals();
89 }
```

DOSSIER PROFESSIONNEL (DP)

Je vais maintenant présenter la fonction “supprimerArticle”. La fonction prend en paramètre l’ID de l’article du panier. Je sélectionne l’`<article>` du produit à supprimer grâce à un `querySelector`. Je recherche ensuite le produit à supprimer dans le panier grâce à son ID. Je crée ensuite un élément HTML de modal grâce à “`createElement`” avec une balise `<dialog>`. La balise `<dialog>` est la plus adaptée pour l’utilisation de modal car plus simple à manipuler en JavaScript. Je définis ensuite le template de la modal avec une croix pour fermer la modale, un titre, un texte de demande de confirmation et enfin deux boutons pour valider ou annuler la suppression. J’insère ensuite ma modal au début du body. Enfin, j’affiche la modal grâce à “`showModal`”.

```
91 // Fonction de suppression d'élément
92 function supprimerArticle(id : any) : void {
93     const articleASupprimer : Element | null = document.querySelector(selectors: `#${id}`);
94     const produitASupprimer : any = panier.find((produit : any) : boolean => produit.idCart === id);
95     const modal : HTMLDialogElement = document.createElement(tagName: "dialog");
96     modal.id = "delete";
97     modal.innerHTML =
98         <i class="fa-solid fa-xmark"></i>
99         <p id="title-modale">Supprimer un article</p>
100        <p>Êtes-vous sûr de vouloir supprimer cet article ?</p>
101        <div>
102            <button id="yes">Oui</button>
103            <button id="no">Non</button>
104        </div>;
105        document.body.insertAdjacentElement(where: "afterbegin", modal);
106        modal.showModal();
```

J’ajoute un “`addEventListener`” sur mon icône de croix pour pouvoir fermer la modal en cliquant dessus. Ainsi, “`.close`” permet de fermer la modal et “`.remove`” supprimer l’élément du HTML. Je gère maintenant mes boutons “Oui” et “Non” pour valider ou non la suppression. Je les sélectionne avec un `querySelector` et ajoute un “`addEventListener`” à chacun. Pour le bouton “Non”, je ferme la modale et la supprime avec les méthodes “`close`” et “`remove`”, puis je mets à jour la valeur de l’input de quantité. Si sa valeur est inférieure ou égale à 0, je la mets à 1, sinon je n’y touche pas. Je recalcule les totaux une nouvelle fois. Pour le bouton “Oui”, je récupère l’index du produit à supprimer dans le panier, puis j’utilise la méthode “`splice`” pour retirer l’élément du panier. Je ferme la modal, je la supprime, je supprime l’`<article>` correspondant, je mets le panier à jour dans le localstorage et je termine en calculant les totaux une nouvelle fois.

DOSSIER PROFESSIONNEL (DP)

```
108 document.querySelector(selectors: "dialog > i").addEventListener(type: "click", () : void => {
109     modal.close();
110     modal.remove();
111     return;
112 });
113
114 // Ecouter si l'utilisateur confirme ou non vouloir supprimer l'article
115 const boutonOui : Element | null = document.querySelector(selectors: "#yes");
116 const boutonNon : Element | null = document.querySelector(selectors: "#no");
117
118 boutonNon.addEventListener(type: "click", (e : Event) : void => {
119     modal.close();
120     modal.remove();
121     document.querySelector(selectors: `#quantity-${id}`).value =
122         document.querySelector(selectors: `#quantity-${id}`).value <= 0
123         ? 1
124         : document.querySelector(selectors: `#quantity-${id}`).value;
125     calcTotals();
126 });
127
128 boutonOui.addEventListener(type: "click", (e : Event) : void => {
129     const index : any = panier.indexOf(produitASupprimer);
130     if (index != -1) {
131         panier.splice(index, 1);
132     }
133     modal.close();
134     modal.remove();
135     articleASupprimer.remove();
136     localStorage.setItem(key: "panier", JSON.stringify(panier));
137     calcTotals();
138 });
139 }
```

Je vais maintenant présenter la fonction “calcTotals”. Je sélectionne d'abord mon paragraphe d'affichage de total avec un querySelector. Je définis mes variables “prixTotal” et “articleTotal” à 0. Ensuite, pour chaque article dans le panier, je fais 2 déstructurations. La première me permet de récupérer l'ID de produit, la taille sélectionnée et la quantité stockés dans le localstorage. Grâce à l'ID, je récupère le produit dans “datas”. La seconde déstructuration me permet de récupérer les déclinaisons existantes du produit. Je récupère le prix correspondant à la taille sélectionnée. Ensuite, j'ajoute la quantité du produit à la quantité totale, ainsi que le prix x la quantité au prix total. J'ajoute une sécurité pour être sûr que si le panier est vide, et mes totaux sont bien à 0. Enfin, je mets à jour l'affichage des totaux.

DOSSIER PROFESSIONNEL (DP)

```
141 // Calcul du total d'article et du total de prix
142 function calcTotals() : void {
143     const totalArticles : Element | null = document.querySelector(selectors: "#total-article");
144     let prixTotal : number = 0;
145     let articleTotal : number = 0;
146     panier.forEach((article : any) : void => {
147         const { id, taille, quantite } = article;
148         const infoArticle : any = datas.find([art : any] : boolean => art._id === id); You, 4 mo
149         const { declinaisons } = infoArticle;
150         const prix : any = declinaisons[taille].prix;
151         articleTotal += parseInt(quantite);
152         prixTotal += parseInt(quantite) * prix;
153     });
154     if (panier == []) {
155         articleTotal = 0;
156         prixTotal = 0;
157     }
158     totalArticles.textContent = `${articleTotal} articles pour un montant de ${prixTotal.toFixed(
159         fractionDigits: 2
160     )} €`;
161 }
```

Maintenant que j'ai présenté les fonctions de gestion du panier, je vais présenter les différents éléments de gestion de l'envoi de commande et gestion du formulaire. Pour commencer, je récupère mon formulaire avec un "querySelector" et j'y attribue un "addEventListener" lors de la soumission du formulaire. J'utilise "e.preventDefault()" pour bloquer l'exécution par défaut de la soumission du formulaire (c'est-à-dire le rechargeement de la page). Ensuite je définis tous les regex qui me permettront de vérifier que le contenu des différents champs est cohérent avec les données attendues. Grâce à ces regex, j'ajoute un autre niveau de sécurité. Chaque regex est expliqué en détail sur la capture d'écran.

```
261 // Gestion du formulaire
262 const formulaire : HTMLFormElement | null = document.querySelector(selectors: "form");
263 formulaire.addEventListener(type: "submit", (e : submitEvent) : void => {
264     e.preventDefault();
265     const regexFirstName = /^[a-zA-ZÀ-ÿ-]{2,}$/; // Toutes les majuscules et minuscule, toutes les lettres avec accents, espaces et tirets. Minimum 2 caractères
266     const regexLastName = /^[a-zA-ZÀ-ÿ-]{2,}$/; // Toutes les majuscules et minuscule, toutes les lettres avec accents, espaces et tirets. Minimum 2 caractères
267     const regexAddress = /^[a-zA-Z0-9À-ÿ-]{10,}$/; // Toutes les majuscules et minuscules, toutes les lettres avec accents, tous les chiffres, les espaces, tirets et apostrophes. Minimum 10 caractères
268     const regexCity = /^[a-zA-ZÀ-ÿ-]{3,}$/; // Toutes les majuscules et minuscule, toutes les lettres avec accents, espaces et tirets. Minimum 3 caractères
269     const regexEmail = /^[a-zA-Z0-9._]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,10}$/; // Toutes les majuscules et minuscules, tous les chiffres, les points, les tirets et les underscores. Un arrobase obligatoire. Toutes les majuscules et minuscules, tous les chiffres, les points et les tirets. Un point obligatoire. Toutes les majuscules et minuscules, entre 2 et 10 caractères.
```

Ensuite, je définis une variable "isCorrect" avec pour valeur "true", qui me permettra de savoir s'il y a au moins une erreur dans les champs à remplir. Je vérifie le contenu de chacun des champs grâce à mes regex grâce à la méthode "test". Si le test n'est pas valide, je donne la valeur "false" à ma variable "isCorrect" et j'affiche un message d'erreur sous l'input en cause, sinon je retire le contenu textuel de cette zone d'erreur. Je reproduis les mêmes lignes (sauf la 270) pour tous les champs à remplir.

```
270     let isCorrect : boolean = true;
271     if (!regexFirstName.test(document.querySelector(selectors: "#first-name").value)) {
272         isCorrect = false;
273         document.querySelector(selectors: "#error-first-name").textContent =
274             "Erreur ! Veuillez entrer une valeur valide";
275     } else {
276         document.querySelector(selectors: "#error-first-name").textContent = "";
277     }
```

DOSSIER PROFESSIONNEL (DP)

Si “isCorrect” vaut “false”, je mets fin à la fonction avec un return. Sinon, je crée un objet “contact” avec les données du formulaire, une liste “products” dans laquelle je “push” tous les ID des produits présents dans la commande et enfin un objet “order” avec l’objet “contact” et la liste “products”. Pour “products”, je n’envoie que l’ID des produits conformément aux consignes, dans un projet réel j’envirrais la quantité et la déclinaison. Enfin, j’appelle la fonction “envoiCommande” avec l’objet “order” en argument.

```
306 if (!isCorrect) {  
307     return;  
308 }  
309 const contact = {  
310     firstName: document.querySelector(selectors: "#first-name").value,  
311     lastName: document.querySelector(selectors: "#last-name").value,  
312     address: document.querySelector(selectors: "#address").value,  
313     city: document.querySelector(selectors: "#city").value,  
314     email: document.querySelector(selectors: "#mail").value,  
315 };  
316 const products : any[] = [];  
317 panier.forEach((produit : any) : void => {  
    products.push(produit.id);  
});  
318  
319 const order = {  
    contact,  
    products,  
};  
320  
321     envoiCommande(order);  
322  
323 };  
324  
325 };  
326
```

Je vais maintenant expliquer la fonction “envoiCommande”. Elle prend comme paramètre “order” et est asynchrone. Pour commencer, je vérifie si le panier contient au moins un produit, dans le cas contraire, j’appelle la fonction “affichageErreur”. Si il n’y a pas d’erreur avec le panier, je fais un “try catch” pour la sécurité. Je fais une requête à la base de données, cette fois-ci j’utilise la méthode “POST” pour envoyer des informations. Je précise également dans ma requête que le contenu envoyé est une application ou un JSON. Enfin, j’intègre mon objet “order” dans le body de ma requête. J’attends ensuite la réponse du serveur, que je convertis en JSON pour pouvoir la manipuler. J’appelle enfin ma fonction “affichageNumeroCommande” avec le numéro de commande envoyé par le serveur.

DOSSIER PROFESSIONNEL (DP)

```
165 // Fonction d'envoi de commande
166 async function envoiCommande(order : any) : Promise<void> {
167   if (panier.length > 0) {
168     try {
169       const req : Response = await fetch(input: "http://localhost:3000/api/products/order", {
170         method: "POST",
171         headers: {
172           "Content-type": "application/JSON",
173         },
174         body: JSON.stringify(order),
175       });
176       const res : any = await req.json();
177       affichageNumeroCommande(res.orderId);
178     } catch (e) {
179       console.error(e);
180     }
181   } else {
182     affichageErreur();
183   }
184 }
```

Je vais présenter les fonctions “affichageErreur” puis “affichageNumeroCommande”. Pour la fonction “affichageErreur”, c'est assez proche de ma fonction de confirmation de suppression vue plus tôt. Je crée une modale, je l'insère dans le body de mon document, je définis le contenu, j'ajoute un “addEventListener” sur l'icône de croix pour fermer la modale. La différence ici est que j'utilise un “setTimeout” pour fermer automatiquement la modale après 3 secondes (3000 millisecondes).

```
210 function affichageErreur() : void {
211   const modal : HTMLDialogElement = document.createElement(tagName: "dialog");
212   document.body.insertAdjacentElement(where: "afterbegin", modal);
213   modal.innerHTML =
214     <i class="fa-solid fa-xmark"></i>
215     <p id="title-modale">Erreur</p>
216     <p>Votre panier est vide, ajoutez au moins un article pour pouvoir passer commande</p>
217   ;
218   document.querySelector(selectors: "dialog > i").addEventListener(type: "click", () : void => {
219     modal.close();
220     modal.remove();
221     return;
222   });
223   modal.showModal();
224   setTimeout(() : void => {
225     modal.close();
226     modal.remove();
227   }, timeout: 3000);
228 }
```

Pour la fonction “affichageNumeroCommande”, elle prend en paramètre le numéro de commande (“numeroDeCommande”). Elle fonctionne une nouvelle fois avec une modale. C'est le même fonctionnement que la précédente fonction, à la différence que cette fois ci, le “setTimeout” est réglé à 6 secondes. Il y a également un appel à la fonction “clearPage” lors de la fermeture de la modale.

DOSSIER PROFESSIONNEL (DP)

```
186 // Fonction d'affichage du numéro de commande
187 function affichageNumeroCommande(numeroDeCommande : any) : void {
188   const modal : HTMLDialogElement = document.createElement(tagName: "dialog");
189   document.body.insertAdjacentElement(where: "afterbegin", modal);
190   modal.innerHTML =
191     <i class="fa-solid fa-xmark"></i>
192     <p id="title-modal">Félicitations</p>
193     <p>La commande a été passée avec succès</p>
194     <p id="command-number">Votre numéro de commande est : ${numeroDeCommande}</p>
195   ;
196   document.querySelector(selectors: "dialog > i").addEventListener(type: "click", () : void => {
197     modal.close();
198     modal.remove();
199     clearPage();
200     return;
201   });
202   modal.showModal();
203   setTimeout(() : void => {
204     modal.close();
205     modal.remove();
206     clearPage();
207   }, timeout: 6000);
208 }
```

Je vais présenter la fonction “clearPage”. J'utilise la méthode “reset” pour vider tous les champs de mon formulaire. Je supprime ensuite tous les articles qui me permettaient d'afficher les produits du panier. J'utilise la méthode “clear” pour vider mon localStorage et ainsi m'assure qu'il ne reste aucun produit dans mon stockage. Je réinitialise le panier en liste vide et je rajoute un article à la place de mes articles de produit indiquant que le panier est vide. Enfin, j'appelle la fonction “calcTotals” pour remettre les totaux à 0.

```
230 // Fonction pour remettre la page à 0 et vider le panier
231 function clearPage() : void {
232   formulaire.reset();
233   const articles : NodeList<HTMLElement> = document.querySelectorAll(selectors: "article");
234   articles.forEach(article : HTMLElement) : void => {
235     article.remove();
236   };
237   localStorage.clear();
238   panier = [];
239   document.querySelector(selectors: "#panier > div").insertAdjacentHTML(
240     position: "afterbegin",
241     string: `
242       <article>
243         <p>Votre panier est vide. Veuillez ajouter au moins un article à votre panier</p>
244       </article>
245     `;
246   calcTotals();
247 }
```

J'ai également réalisé rapidement la version responsive mobile et tablette de la page panier à l'aide d'un media query en CSS. Ainsi, j'ai entièrement développé le front d'une boutique fictive avec gestion du panier et affichage dynamique des informations de produits et du panier sans recharge de page.

DOSSIER PROFESSIONNEL (DP)

2. Précisez les moyens utilisés :

Pour ce projet, j'ai travaillé avec HTML, CSS et JavaScript. J'ai suivi les indications fournies avec le projet. J'ai démarré le back du projet pour pouvoir en extraire les données à afficher. Pour ça, j'ai ouvert un terminal dans le back en faisant clic droit sur le dossier back > Ouvrir dans un terminal intégré puis j'ai tapé "npm install" pour installer les dépendances du back, puis "node server", j'ai pressé TAB pour auto-compléter le nom du document à démarrer puis j'ai appuyer sur entrée. Le terminal a bien indiqué que le back était disponible sur le port 3000.

3. Avec qui avez-vous travaillé ?

Pour ce projet, j'ai travaillé en totale autonomie.

4. Contexte

Nom de l'entreprise, organisme ou association - La plateforme

Chantier, atelier, service - Formation Développeur Web & Web Mobile

Période d'exercice - Du 04/06/2025 au 05/06/2025

5. Informations complémentaires (*facultatif*)

DOSSIER PROFESSIONNEL (DP)

Cliquez ici pour taper du texte.

Activité-type 1 Développer la partie front-end d'une application web ou web mobile sécurisée

Exemple n°4 - Développer la partie dynamique des interfaces utilisateur web ou web mobile

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

J'ai dû développer la partie front-end d'une application web de location de logements avec React et TypeScript. Le projet m'est fourni accompagné d'un lien [Figma](#) avec le design du site, ainsi qu'un fichier "logements.json" contenant les informations de tous les logements présents sur le site.

Je commence par initialiser le projet. J'ouvre un terminal, et je tape "npm create vite@latest" pour initialiser un nouveau projet React avec TypeScript. Je note "." dans "Project name" pour que mes fichiers soient créés directement dans le dossier où je me trouve. Si j'écris autre chose, un dossier sera créé avec le nom que j'ai renseigné. Ensuite, je sélectionne "React", je fais "entrée", je choisis "TypeScript" pour ce projet conformément aux consignes. Je choisis "No" pour "rolldown-vite" et enfin j'appuie sur "N" pour uniquement initialiser le projet. Si je fais "Y", j'initialise, installe et lance directement le projet.

DOSSIER PROFESSIONNEL (DP)

```
PS C:\Users\Utilisateur\Desktop\DWWM\React\test> npm create vite@latest
```

```
> npx
> create-vite

diamond Project name:
.
diamond Select a framework:
  React
diamond Select a variant:
  TypeScript
diamond Use rollup-vite (Experimental)?:
  No
diamond Install with npm and start now?:
  No
diamond Scaffolding project in C:\Users\Utilisateur\Desktop\DWWM\React\test...
  Done. Now run:

  npm install
  npm run dev
```

J'installe le projet avec "npm install" ou "npm i".

```
● PS C:\Users\Utilisateur\Desktop\DWWM\React\test> npm i
  added 235 packages, and audited 236 packages in 47s
  47 packages are looking for funding
    run `npm fund` for details
  found 0 vulnerabilities
```

Je vérifie que le projet est fonctionnel avec "npm run dev".

DOSSIER PROFESSIONNEL (DP)

```
PS C:\Users\Utilisateur\Desktop\DWMM\React\test> npm run dev

> test@0.0.0 dev
> vite

✓ Console Ninja extension is connected to Vite, see https://tinyurl.com/2vt8jxzw

VITE v7.1.7 ready in 1581 ms

→ Local: http://localhost:5173/
→ Network: use --host to expose
→ press h + enter to show help
```

Le projet fonctionne. J'utilise CTRL+C pour couper le projet et installer les dépendances qui me serviront pour le projet :

- react-i18next : pour les versions française et anglaise.
- lucide-react : pour les icônes.
- react-country-flag : pour afficher des drapeaux (pour le changement entre version française et anglaise)
- react-router-dom : pour gérer les routes de mes différentes pages
- sass : pour le style

J'installe tout en une fois avec "npm i" suivi de toutes les dépendances à installer.

```
PS C:\Users\Utilisateur\Desktop\DWMM\React\test> npm i react-i18next sass react-router-dom lucide-react react-country-flag

added 77 packages, and audited 267 packages in 27s

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

found 0 vulnerabilities
```

Une fois les dépendances installées, je prépare l'arborescence du projet. Je crée, dans le dossier "src", un dossier "components" qui contiendra les composants React, un dossier "data" qui contient le fichier "logements.json" fourni, un dossier "locales" qui me permettra de gérer les traductions en plusieurs langues, un dossier "pages" qui contiendra les pages du site et un dossier "styles" qui contiendra les fichiers SCSS. Dans le dossier "styles", je crée deux dossiers supplémentaires: un dossier "bases" qui contiendra les variables, animations et styles globaux ainsi qu'un dossier "components" qui contiendra les styles individuels de chaque composants et pages. Dans le dossier "locales", je crée deux dossiers supplémentaires, un dossier "en" et un dossier "fr". Chaque dossier contient un document "translation.json" qui contient tous les contenus du site dans la langue sélectionnée. Enfin, je crée un fichier "i18n.ts" pour gérer les traductions.

DOSSIER PROFESSIONNEL (DP)

```
▽ src
  > assets
  > components
  > data
  ▽ locales
    > en
    > fr
  > pages
  ▽ styles
    > bases
    > components
  ⚡ App.tsx
  TS i18n.ts
  ⚡ main.tsx
```

J'initialise ensuite un [repository GitHub](#) pour pouvoir conserver mon projet et ses différentes versions en ligne.

Je supprime les fichiers css “App.css” et “index.css”, ils ne seront pas utiles dans ce projet. Je crée dans mon dossier “styles/bases” un fichier “global.scss” qui contiendra tous les styles par défaut du projet. Je créerai un fichier SCSS pour chaque composant du projet.

Avant de commencer le développement du projet, j'analyse le Figma fourni pour en ressortir des variables de couleurs et de polices d'écriture, ainsi que des styles par défaut (par exemple : conteneurs de contenu, titrages, classes). Je crée donc un fichier “_variables.scss” et un fichier “_animations.scss”. Le underscore ici est important, il permet d'éviter que le fichier soit compilé. J'importerai ensuite les fichiers dans chaque autre fichier SCSS pour accéder aux variables/animation. Voici le fichier de variables.

DOSSIER PROFESSIONNEL (DP)

```
1 $font: "Inter", sans-serif;           You, last month • fix: improve UI ar
2 $logo-font: "Pacifico", cursive;
3 $white: #ffffff;
4 $dark-text: #1f2937;
5 $light-text: #4b5563;
6 $grey-text: #d1d5db;
7 $grey-background: #f9fafb;
8 $grey-border: #e5e7eb;
9 $dark: #1f2937;
10 $gradient: linear-gradient(116.22deg, #667eea 0%, #764ba2 100%);
11 $yellow: #ffd700;
12 $logo-pink: #ff6b6b;
13 $green: #4ecdc4;
14
```

Un exemple d'import et d'utilisation de variables dans “global.scss”.

```
3 @import "variables";
4
5 // Import fonts
6
7 @import url("https://fonts.go
8 @import url("https://fonts.go
9
10 // Uniformisation document
11
12 *
13   font-family: $font;
14   margin: 0;
15   padding: 0;
16   list-style: none;
17   text-decoration: none;
18   transition: all 0.3s ease;
19 }
```

Je remplace dans “main.tsx” l’import du fichier “index.css” par le nouveau fichier “global.scss”. Ainsi, les styles définis dans ce fichier seront accessibles globalement dans tout le projet. J’utilise “BrowserRouter” de la librairie “react-router-dom”. Sans cela, pas de navigation entre les pages du projet.

DOSSIER PROFESSIONNEL (DP)

```
1 import React from "react";
2 import ReactDOM from "react-dom/client";
3 import { BrowserRouter } from "react-router-dom";
4 import App from "./App";
5 import './styles/bases/global.scss';
6
7 ReactDOM.createRoot(document.getElementById("root")!).render(
8   <React.StrictMode>
9     <BrowserRouter>
10       <App />
11     </BrowserRouter>
12   </React.StrictMode>
13 );
14
```

Maintenant, je prépare la page d'accueil "Home.tsx" pour tester le bon fonctionnement des routes. Je crée une fonction fléchée "Home" qui renvoie un simple <h1> et j'exporte ma fonction pour la rendre accessible partout dans le projet.

```
1 const Home : () => Element = () => {
2   return(
3     <h1>Page d'accueil</h1>
4   )
5 }
6
7 export default Home;
```

Je vais maintenant supprimer le contenu par défaut du fichier "App.tsx" pour pouvoir définir les routes. J'inclu pour le moment uniquement la route pour la page d'accueil. Pour être sûr d'importer correctement mon élément "Home", à la ligne 7, j'écris "Home" puis j'appuie sur CTRL + Espace; cela me permet d'avoir la liste des modules disponibles contenant ce nom. Je sélectionne l'élément correspondant et j'appuie sur Entrée. L'import se fait alors automatiquement et j'évite ainsi toute erreur.

DOSSIER PROFESSIONNEL (DP)

```
1 import { Route, Routes } from "react-router-dom";
2 import Home from "./pages/Home";
3
4 function App() {
5   return (
6     <Routes>
7       <Route path="/" element={<Home />} />
8     </Routes>
9   );
10 }
11
12 export default App;
```

Je vais mettre en place la traduction entre anglais et français pour rendre le site accessible à l'international. Pour ça, je me rends dans mon fichier “i18n.ts” et j'y définit les ressources utilisées (soit les deux documents de traduction). J'initialise ensuite “i18n” avec mes ressources, et je choisis le français comme langue par défaut et l'anglais comme langue secondaire. J'importe le document dans mon fichier “main.tsx”.

DOSSIER PROFESSIONNEL (DP)

```
1 import i18n from "i18next";
2 import { initReactI18next } from "react-i18next";
3
4 import translationEN from "./locales/en/translation.json";
5 import translationFR from "./locales/fr/translation.json";
6
7 const resources = {
8   en: { translation: translationEN },
9   fr: { translation: translationFR },
10 };
11
12 i18n
13   .use(initReactI18next)
14   .init({
15     resources,
16     lng: "fr",
17     fallbackLng: "en",
18     interpolation: {
19       escapeValue: false,
20     },
21   });
22
23 export default i18n;
```

Maintenant que je sais que mes routes fonctionnent, que j'ai défini mes styles par défaut avec SCSS et que ma traduction est paramétrée, je peux commencer à intégrer le projet. Je commence par créer deux composants, "Header.tsx" et "Footer.tsx".

Pour le Header, je prépare un composant "NavBar". À long terme, avoir un composant NavBar à part entière me permettra de mettre à jour les liens de navigation du site plus facilement. Mon élément "NavBar" prend en paramètre une valeur booléenne "setClass" qui sera utile pour le responsive. Si "setClass" vaut "true", la classe "active" est attribuée au menu de navigation pour l'afficher, sinon elle reste cachée. Ensuite, j'importe et utilise "useTranslation". Cette fonction permettra au texte d'être affiché dans la bonne langue de manière dynamique. Enfin, tous mes textes (dans ce fichier comme dans tous les autres) seront notés sous le format {t("valeur.du.texte")}. Cette notation fonctionne comme pour du JSON. Par exemple ici, pour le premier "Link", l'application va récupérer la valeur correspondante à "nav > home" dans le fichier json correspondant à la langue choisie. Comme pour tous les composants et pages, je n'oublie pas d'exporter ma fonction.

DOSSIER PROFESSIONNEL (DP)

```
1 import { useTranslation } from "react-i18next";
2 import { Link } from "react-router-dom";
3 import "../styles/components/NavBar.scss";
4
5 You, 4 weeks ago | 1 author (You)
6 export interface NavBarProps {
7   setClass?: boolean;
8 }
9 const NavBar : ({ setClass }: NavBarProps) => Element = ({ setClass }: NavBarProps) => {
10   const { t } = useTranslation();
11   return (
12     <nav className={setClass ? "active" : ""}>
13       <ul>
14         <li>
15           <Link to="/">{t("nav.home")}</Link>
16         </li>
17         <li>
18           <Link to="#">{t("nav.about")}</Link>
19         </li>
20         <li>
21           <Link to="#">{t("nav.contact")}</Link>
22         </li>
23       </ul>
24     </nav>
25   );
26 }
27
28 export default NavBar;
```

Maintenant que le composant pour mon menu de navigation est terminé, je peux reprendre mon header. J'utilise un “useState” pour l'affichage du menu de navigation en responsive. Lorsque que l'icône du “burgermenu” est cliquée, une fonction est appelée pour inverser l'affichage du menu (si activé, se désactive et inversement). Si l'utilisateur scroll, le menu se ferme également. Le logo est importé de la bibliothèque “lucide-react” comme tous les autres logo du site. Les éléments <Link to= > permettent la navigation entre les différentes pages du site.

DOSSIER PROFESSIONNEL (DP)

```
1 import { Menu } from "lucide-react";
2 import { useState } from "react";
3 import { Link } from "react-router-dom";
4 import "../styles/components/Header.scss";
5 import NavBar from "./NavBar";
6
7 const Header : () => Element = () => {
8     const [isActive, setActive] = useState(false);
9     const showNav : () => void = () => {
10         setActive(!isActive);
11     };
12     document.addEventListener("scroll", () => {
13         setActive(false);
14     });
15     return (
16         <header>
17             <div className="container-1280">
18                 <Link to="/" id="logo">
19                     Kasa
20                 </Link>
21
22                 <NavBar setClass={isActive} />
23
24                 <Menu id="burgermenu" onClick={showNav} />
25             </div>
26         </header>
27     );
28 };
29
30 export default Header;
```

Pour le footer, j'ai principalement fait du HTML classique avec l'intégration des textes en multi-langues comme pour le header. Au clic sur un des boutons contenant les drapeaux, la fonction de changement de langue est appelée pour changer la langue avec la valeur correspondante à ce qui a été défini dans le fichier "i18n". Pour les drapeaux, j'ai utilisé "react-country-flag" qui demande de fournir le code ISO du pays sélectionné, de préciser que le logo doit être un SVG et le style de l'icône.

DOSSIER PROFESSIONNEL (DP)

```
16  const Footer : () => Element = () => {
17    const { t, i18n } = useTranslation();
18
19    const changeLanguage : (lng: string) => void = (lng: string) => {
20      i18n.changeLanguage(lng);
21    };
22
```

```
76    <div id="languages">
77      <button onClick={() => changeLanguage("fr")}>
78        <ReactCountryFlag
79          countryCode="FR"
80          svg
81          style={{
82            width: "30px",
83            height: "30px",
84          }}
85        />
86      </button>
87      <button onClick={() => changeLanguage("en")}>
88        <ReactCountryFlag
89          countryCode="GB"
90          svg
91          style={{
92            width: "30px",
93            height: "30px",
94          }}
95        />
96      </button>
```

Pour la page d'accueil, j'ai deux sections, la section Hero et la section de présentation de tous les logements. Pour chacune, je fais un composant individuel. Le Hero est du simple HTML stylisé, et pour la présentation des logements, je crée un composant "Card" pour styliser les cartes uniques de chaque logement automatiquement.

Tout d'abord, je définis les paramètres du composant et leur type. Typer les variables est obligatoire en TypeScript.

DOSSIER PROFESSIONNEL (DP)

```
1 import { MapPin, Star } from "lucide-react";
2 import { useTranslation } from "react-i18next";
3 import { Link } from "react-router-dom";
4 import "../styles/components/Card.scss";
5
6 You, 4 weeks ago | 1 author (You)
7 export interface CardProps {
8   id: string;
9   title: { fr: string; en: string };
10  location: string;
11  picture: string[];
12  tags: { fr: string; en: string }[];
13  rating: string;
14}
```

Pour les informations de logements, je ne passe pas par les fichiers de traduction présents dans le dossier “locales”; j’ai les différentes versions dans le document “logements.json”. Par exemple, pour le titre du logement, le fichier ressemble à l’image ci-contre :

```
4   "title": {
5     "fr": "Appartement cosy \u00e0 Marseille",
6     "en": "Cozy apartment in Marseille"
7   },
```

Je définis un tableau de 5 nombres, il me permet d’afficher correctement le nombre d’étoiles en accord avec la note du logement. Ensuite, je remplis mon HTML avec les informations récupérées lors de l’appel du module. Pour le choix de la langue, je précise qu’il faut appeler la langue active avec “i18n.language”. Ensuite, pour les tags du logement, j’utilise la méthode “map” qui fonctionne comme un “forEach” en JavaScript classique. Enfin, pour l’affichage de la note sous la forme d’étoiles, j’utilise la méthode “map” sur mon tableau défini plus tôt. Si la note est inférieure ou égale au numéro du tableau, l’étoile est remplie, sinon l’étoile reste vide. Pour l’affichage de la note sous forme numérique, j’ai décidé d’ajouter une valeur décimale, car plus réaliste selon moi. Cette note est récupérée dans le fichier, puis le nombre après la virgule est généré avec les méthodes “Maths.floor” et “Math.random” qui vont me permettre de générer aléatoirement un nombre décimal entre 0 et 1, le multiplier par neuf, et lui ajouter 1 avant de l’arrondir à l’entier inférieur. Ainsi, les notes seront plus réalistes. Enfin, si la note est de 5, aucun nombre décimal n’est généré car 5 est la note maximale.

DOSSIER PROFESSIONNEL (DP)

```
41 const Card : ({ id, title, location, picture, tags, rating }: CardProps) => {
42   const stars : number[] = [1, 2, 3, 4, 5];
43   const { i18n } = useTranslation();
44   return (
45     <article className="card">
46       <Link to={`/logement/${id}`}>
47         <img src={picture[0]} alt={title[i18n.language as "fr" | "en"]}>{" "}
48         /* On assure à Typescript que la clé est valide sinon erreur */
49       </div>
50       <h2>{title[i18n.language as "fr" | "en"]}</h2>
51       <p>
52         <MapPin size={14} />
53         {location}
54       </p>
55       <div>
56         {tags.map((tag : { fr: string; en: string }) => (
57           <p className="tag" key={tag[i18n.language as "fr" | "en"]}>
58             {tag[i18n.language as "fr" | "en"]}
59           </p>
60         )));
61       </div>
62       <p>
63         {stars.map((star : number) =>
64           star <= Number(rating) ? (
65             <Star key={star} className="filled" size={15} />
66           ) : (
67             <star key={star} size={15} />
68           )
69         );
70         ({rating}
71           {rating === "5" ? "" : `.${Math.floor(Math.random() * 9) + 1}`})
72         </p>
73       </div>
74     </Link>
75   </article>
```

Maintenant que mon module de carte est créé, je peux l'appeler dans mon composant et afficher tous les logements disponibles. Pour cela, j'importe toutes mes données de logements et j'utilise la méthode “map”. J'appelle pour chaque logement mon module “Card” avec toutes les informations requises. J'ajoute une valeur “key” pour que chaque carte ait un identifiant unique.

DOSSIER PROFESSIONNEL (DP)

```
1 import data from "../data/logements.json";
2 import "../styles/components/Accomodations.scss";
3 import Card from "./Card";
4
5 const Accomodations : () => Element = () => {
6   return (
7     <section id="accomodations">
8       <div className="container-1280">
9         {data.map((accomodation : { id: string; title: { fr: string; en: string } }) => (
10           <Card
11             key={accomodation.id}
12             id={accomodation.id}
13             title={accomodation.title}
14             location={accomodation.location}
15             picture={accomodation.pictures}
16             tags={accomodation.tags}
17             rating={accomodation.rating}
18           />
19         )));
20       </div>
21     </section>
22   );
23 };
24
25 export default Accomodations;
```

Ma page d'accueil terminée, je réalise maintenant ma page "Details" pour afficher un logement et ses informations précises. Pour cette page, j'ai besoin d'un composant "Carousel" pour afficher dynamiquement toutes mes images et permettre de les feuilleter. Pour cela, je défini en paramètre un tableau de chaîne de caractères. Ces chaînes de caractères correspondent au lien d'accès aux images. Initialement, le projet utilisait des liens d'API pour les images, mais suite à un crash de la plateforme d'images, j'ai décidé de télécharger des images placeholder dans le projet; le lien est donc le chemin d'accès à l'image. J'utilise un hook "useState" pour la gestion de l'index, pour me permettre de naviguer dynamiquement entre les images. Je définis deux fonctions, une pour afficher l'image suivante et une pour afficher l'image précédente, avec une sécurité pour que les images bouclent correctement (exemple : si l'utilisateur est à l'image 3/3 et clique sur suivant, l'image affichée sera l'image 1/3).

DOSSIER PROFESSIONNEL (DP)

```
19 const Carousel : ({ pictures }: CarouselProps) => Element = ({ pictures }: CarouselProps) => {
20   const [index, setIndex] = useState(0);
21   function nextPicture() {
22     if (index + 1 > pictures.length - 1) {
23       setIndex(0);
24     } else {
25       setIndex(index + 1);
26     }
27   }
28   function prevPicture() {
29     if (index - 1 < 0) {
30       setIndex(pictures.length - 1);
31     } else [
32       setIndex(index - 1);
33     ]
34   }
35   return (
36     <div id="carousel">
37       <img src={pictures[index]} alt={`Photo du logement, numéro ${index}`} />
38       <div id="buttons">
39         <button onClick={prevPicture}>
40           <ArrowLeft />
41         </button>
42         <button onClick={nextPicture}>
43           <ArrowRight />
44         </button>
45       </div>
46     </div>
47   );
48 }
```

Je peux maintenant m'occuper de la page “Details” en elle-même. Je commence par les imports; j'ai besoin de beaucoup d'icônes pour l'affichage des équipements du logement. J'importe également “useParams” pour pouvoir récupérer l'ID du logement à afficher.

DOSSIER PROFESSIONNEL (DP)

```
1 import {
2   AirVent,
3   ArrowDown,
4   ArrowUp,
5   BrickWallFire,
6   Cable,
7   Coffee,
8   CookingPot,
9   Droplets,
10  Flame,
11  FlameKindling,
12  Hamburger,
13  Heater,
14  MapPin,
15  SeparatorHorizontal,
16  Toilet,
17  ToyBrick,
18  Trees,
19  Tv,
20  Utensils,
21  WashingMachine,
22  WavesLadder,
23  Wifi,
24 } from "lucide-react";
25 import { useState } from "react";
26 import { useTranslation } from "react-i18next";
27 import { Link, useParams } from "react-router-dom";
```

Je prépare un objet “icons” qui permettra d’afficher l’icône correspondant au bon équipement lors du chargement de la page. Je récupère l’ID dans l’URL de la page et je l’utilise pour trouver le logement recherché grâce à la méthode “find”. J’initialise également l’affichage des menus déroulants de la description et des équipements.

DOSSIER PROFESSIONNEL (DP)

```
39 const Details: () => Element = () => {
40   const icons = {
41     Wifi: <Wifi />,
42     Cuisine: <Utensils />,
43     "T\u00e9l\u00e9viseur": <Tv />,
44     Climatisation: <AirVent />,
45     Chauffage: <Heater />,
46     "Cuisine \u00e9quip\u00e9e": <CookingPot />,
47     "Machine \u00e0 laver": <WashingMachine />,
48     Jardin: <Trees />,
49     Barbecue: <Hamburger />,
50     "Chemin\u00e9": <FlameKindling />,
51     "Po\u00eale en bois": <Flame />,
52     "Eau chaude": <Droplets />,
53     "Toilettes s\u00e8ches": <Toilet />,
54     "Machine \u00e0 caf\u00e9": <Coffee />,
55     Piscine: <WavesLadder />,
56     "Lave-vaisselle": <WashingMachine />,
57     Kitchenette: <CookingPot />,
58     Sauna: <BrickWallFire />,
59     "Cuisine moderne": <CookingPot />,
60     Ascenseur: <SeparatorHorizontal />,
61     "Jeux enfants": <ToyBrick />,
62     Terrasse: <Trees />,
63     "Electricit\u00e9": <Cable />,
64   };
65   const { id } = useParams();
66   const [isActiveDesc, setActiveDesc] = useState(false);
67   const [isActiveEquip, setActiveEquip] = useState(false);
68   const acco: { id: string; title: { fr: string; en: string } } = data.find(item => item.id === id);
69   const { t, i18n } = useTranslation();
```

J'utilise un affichage conditionnel selon si le logement est trouvé ou non. Dans le cas où le logement n'est pas trouvé, j'affiche un message d'erreur.

DOSSIER PROFESSIONNEL (DP)

```
150 } else {
151     document.title = "Logement introuvable";
152     return (
153         <>
154         <Header />
155
156         <main>
157             <section id="not-found">
158                 <h1>{t("details.error-accommodation")}</h1>
159                 <Link to="/">{t("details.list-link")}</Link>
160             </section>
161         </main>
162
163         <Footer />
164     </>
165 );
166 }
167 };
168
```

Si le logement est trouvé, je fais une déstructuration de données pour récupérer toutes les informations du logement. Je change le titre du document et j'initialise 2 fonctions pour afficher ou cacher la description et les équipements.

```
70 if (acco) {
71     const { title, location, pictures, tags, host, description, equipments } =
72         acco;
73     document.title = title[i18n.language as "fr" | "en"];
74
75     const showDesc : () => void = () => {
76         setActiveDesc(!isActiveDesc);
77     };
78
79     const showEquip : () => void = () => {
80         setActiveEquip(!isActiveEquip);
81     };
82 }
```

DOSSIER PROFESSIONNEL (DP)

Maintenant que tous les éléments sont initialisés, je peux construire la page. J'appelle d'abord l'élément "Carousel" créé plus tôt auquel j'envoie les images du logement. Ensuite, pour les tags j'utilise la méthode "map" comme pour les "Card" utilisées plus tôt.

```
83     return (
84       <>
85       <Header />
86
87       <main>
88         <section id="detail">
89           <Carousel pictures={pictures} />
90           <div id="infos">
91             <h1>{title[i18n.language as "fr" | "en"]}</h1>
92             <p>
93               <MapPin size={14} />
94               {location}
95             </p>
96             <div id="tags">
97               {tags.map((tag : { fr: string; en: string }) => (
98                 <p className="tag" key={tag[i18n.language as "fr" | "en"]}>
99                   {tag[i18n.language as "fr" | "en"]}
100                  </p>
101                )));
102             </div>
```

Je m'occupe maintenant de l'affichage de la description et des équipements. Le fonctionnement est le même pour les deux éléments. Lors du clique sur le bouton, la classe "active" est ajoutée à la div pour afficher le contenu sous le bouton, et l'icône de flèche change pour une meilleure clarté pour l'utilisateur. Pour la description, le contenu est un simple texte. Pour les équipements, j'utilise la méthode "map" sur l'objet "equipements" pour tous les afficher. Comme sur le figma, chaque équipement est accompagné d'une icône, j'utilise l'objet "icons" créé plus tôt pour afficher l'icône correspondante. Si je voulais optimiser cet affichage, je pourrais ajouter la balise de l'icône dans mon fichier "logements.json" et simplement importer toutes les icônes de "lucide-react" pour y accéder rapidement.

DOSSIER PROFESSIONNEL (DP)

```
110      <div id="description" className={isActiveDesc ? "active" : ""}>
111          <button onClick={showDesc}>
112              <h2>{t("details.description-title")}</h2>
113              {isActiveDesc ? (
114                  <ArrowUp size={16} />
115              ) : (
116                  <ArrowDown size={16} />
117              )}
118          </button>
119          <p>{description[i18n.language as "fr" | "en"]}</p>
120      </div>
121      <div id="equipments" className={isActiveEquip ? "active" : ""}>
122          <button onClick={showEquip}>
123              <h2>{t("details.equipments-title")}</h2>
124              {isActiveEquip ? (
125                  <ArrowUp size={16} />
126              ) : (
127                  <ArrowDown size={16} />
128              )}
129          </button>
130          <div>
131              {equipments.map((equipment : { fr: string; en: string }) => (
132                  <p key={equipment[i18n.language as "fr" | "en"]}>
133                      {
134                          icons[
135                              equipment["fr"] as keyof typeof icons
136                          ] /* Sécurité de typage (nécessaire en TS) */
137                      }
138                      {equipment[i18n.language as "fr" | "en"]}
139                  </p>
140              )));
141          </div>
142      </div>
```

Je crée maintenant une page pour gérer les erreurs 404. Pour ça, je renvoie un simple HTML avec un texte d'erreur.

DOSSIER PROFESSIONNEL (DP)

```
1 import { useTranslation } from "react-i18next";
2 import Footer from "../components/Footer";
3 import Header from "../components/Header";
4 import "../styles/components/NotFound.scss";
5 // You, last month • feat: add hero component
6 const NotFound : () => Element = () => {
7   const { t } = useTranslation();
8   document.title = t("error-404");
9   return (
10     <>
11       <Header />
12
13       <main>
14         <section id="not-found">
15           <h1>{t("error-404")}</h1>
16         </section>
17       </main>
18
19       <Footer />
20     </>
21   );
22 };
23
24 export default NotFound;
```

Maintenant que les pages sont toutes créées, je peux mettre à jour les routes d'accès. À la racine du site, on trouve la page d'accueil. On accède ensuite aux informations d'un logement par l'url "/logement/:id" où ":id" correspond à l'identifiant du logement. Enfin, toute URL autre que la racine ou "/logement" renverra à la page d'erreur 404.

DOSSIER PROFESSIONNEL (DP)

```
1 import { Route, Routes } from "react-router-dom";
2 import Details from "./pages/Details";
3 import Home from "./pages/Home";
4 import NotFound from "./pages/NotFound";
5
6 function App() {
7   return (
8     <Routes>
9       <Route path="/" element={<Home />} />
10      <Route path="/logement/:id" element={<Details />} />
11      <Route path="*" element={<NotFound />} />
12    </Routes>
13  );
14}
15
16 export default App;
```

You, last month • feat: add react-

Ainsi, je termine le projet. Pour une meilleure compréhension du projet pour moi-même ou bien d'autres potentiels contributeurs, je décide d'ajouter une documentation TypeDoc. J'installe TypeDoc sur mon projet. J'utilise TypeDoc car parfaitement adapté à des projets en TypeScript.

```
● PS C:\Users\Utilisateur\Desktop\DWWM\React\CP4-Kasa> npm install --save-dev typedoc
up to date, audited 284 packages in 2s

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

found 0 vulnerabilities
```

Je rédige maintenant ma documentation sur chaque composant. Pour exemple, voilà la documentation du module "Card".

DOSSIER PROFESSIONNEL (DP)

```
15  /**
16   * Affichage des logements sous forme de carte cliquable
17   * @param CardProps - Contient tous les paramètres d'informations de logement
18   * @param CardProps.id - Identifiant du logement
19   * @param CardProps.title - Nom du logement
20   * @param CardProps.location - Localisation du logement
21   * @param CardProps.picture - Photos du logement
22   * @param CardProps.tags - Tag informatif du logement
23   * @param CardProps.rating - Note attribuée au logement
24   * @returns Une carte cliquable contenant les informations principales du logement
25   *
26   * @example
27   * // Utilisation de la carte pour afficher tous les logements
28   * {data.map((accomodation) => (
29     <Card
30       key={accomodation.id}
31       id={accomodation.id}
32       title={accomodation.title}
33       location={accomodation.location}
34       picture={accomodation.pictures}
35       tags={accomodation.tags}
36       rating={accomodation.rating}
37     />
38   ))}
39 */
```

Pour générer la documentation, il faut entrer la commande suivante dans le terminal. Ici, j'ai un warning car un lien de mon fichier "README.md" ne renvoie nulle part.

```
● PS C:\Users\Utilisateur\Desktop\DWMM\React\CP4-Kasa> npx typedoc --entryPointStrategy Expand src
./README.md:34:78 - [warning] The relative path CODE_OF_CONDUCT.md is not a file and will not be copied to the output directory
34  Les contributions sont les bienvenues ! Veuillez lire le [CODE_OF_CONDUCT.md](CODE_OF_CONDUCT.md) pour les directives de contribution
[info] html generated at ./docs
[warning] Found 0 errors and 1 warnings
❖ PS C:\Users\Utilisateur\Desktop\DWMM\React\CP4-Kasa>
```

Maintenant, un dossier "docs" a été généré. Il suffit de l'ouvrir, puis de faire clique droit sur le fichier "index.html" et de l'ouvrir avec Live Server pour y accéder. On arrive alors sur cette page.

DOSSIER PROFESSIONNEL (DP)

The screenshot shows a dark-themed web application interface. On the left, there's a sidebar with a tree view of components under 'kasa': 'App', 'components' (which further branches into 'Accomodations', 'Card', 'Carousel', 'Footer', and 'Header'). The main content area displays the 'Kasa' component. It has a title 'Kasa' and a description: 'Kasa est un site de location de logements en ligne. Il permet à l'utilisateur de choisir un logement parmi une liste et d'obtenir les informations spécifiques au logement. Ce projet a été réalisé dans le cadre de la formation Développeur Web & Web Mobile à La Plateforme.' Below this is a 'Table des matières' section with a list of items: 'Installation', 'Utilisation', 'Contribution', and 'Licence'. A 'Settings' section is also visible on the right.

Je peux, sur la gauche, naviguer entre les différents composants que je souhaite consulter pour obtenir toutes les informations renseignées.

Maintenant que mon projet est documenté, je peux le considérer comme étant terminé pour le moment. J'ai donc développé une application web front-end dynamique avec fonctions multilingues et documentation.

2. Précisez les moyens utilisés :

Pour ce projet, j'ai travaillé avec Vite/React et du langage TypeScript. Pour l'affichage disponible en plusieurs langues, j'ai utilisé la bibliothèque i18next pour React. Pour la documentation du projet, j'ai utilisé TypeDoc.

3. Avec qui avez-vous travaillé ?

DOSSIER PROFESSIONNEL (DP)

J'ai travaillé en totale autonomie sur ce projet.

4. Contexte

Nom de l'entreprise, organisme ou association - La plateforme

Chantier, atelier, service - Formation Développeur Web & Web Mobile

Période d'exercice - Du 26/08/2025 au 28/08/2025

5. Informations complémentaires (*facultatif*)

Cliquez ici pour taper du texte.

DOSSIER PROFESSIONNEL (DP)

Activité-type 2 Développer la partie back-end d'une application web ou web mobile sécurisée

Exemple n°5 - Mettre en place une base de données relationnelle

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Nous imaginons une commande fictive.

Je suis missionné pour construire la fondation d'une nouvelle plateforme en ligne de gestion d'événements. Cette application permettra à des utilisateurs de découvrir, de s'inscrire et de participer à divers événements tels que des conférences, des ateliers ou des rencontres professionnelles. Ma mission est de concevoir et implémenter la base de données relationnelle de l'application. Cette base stockera de manière structurée et fiable toutes les informations essentielles : les profils des utilisateurs, le catalogue des événements et le suivi des réservations.

Les user stories suivantes sont également fournies par le client :

1. Créer un compte utilisateur

- En tant que nouveau participant, je veux pouvoir créer un compte avec mon nom et une adresse email unique, afin de pouvoir m'inscrire à des événements.

2. Consulter les événements disponibles

- En tant que participant, je veux consulter la liste complète des événements (nom, description, date, lieu), afin de choisir celui qui m'intéresse.

3. S'inscrire à un événement

- En tant que participant, je veux pouvoir m'inscrire à un événement spécifique, afin de réserver ma place.

4. Consulter mes inscriptions personnelles

- En tant que participant, je veux retrouver la liste de tous les événements auxquels je suis déjà inscrit, afin de gérer mon agenda personnel.

5. Créer un nouvel événement

- En tant qu'organisateur, je veux pouvoir ajouter un nouvel événement à la plateforme, afin de le proposer à la communauté.

6. Lister les participants d'un événement

- En tant qu'organisateur, je veux obtenir la liste des noms et emails de tous les participants inscrits à mon événement, afin de pouvoir préparer leur accueil et la communication.

DOSSIER PROFESSIONNEL (DP)

7. Analyser la popularité des événements

- En tant qu'administrateur du site, je veux pouvoir compter le nombre de participants pour chaque événement, afin d'identifier les sujets les plus populaires et d'orienter la stratégie future.

8. Lister tous les utilisateurs enregistrés

- En tant qu'administrateur du site, je veux pouvoir afficher la liste complète de tous les utilisateurs, afin d'avoir une vue d'ensemble de la communauté grandissante.

9. Mettre à jour les informations d'un événement

- En tant qu'organisateur, je veux pouvoir modifier les détails d'un événement que j'ai créé (par exemple, changer le lieu ou corriger une faute dans la description), afin de m'assurer que les informations communiquées aux participants sont toujours exactes.

10. Annuler son inscription à un événement

- En tant que participant, je veux pouvoir annuler ma réservation à un événement, afin de libérer ma place si j'ai un empêchement.

11. Supprimer un événement annulé

- En tant qu'organisateur, je veux pouvoir supprimer complètement un événement de la plateforme, par exemple si celui-ci est annulé faute de participants.

À partir des user stories, j'en tire des informations qui me serviront à la conception de la base de données ainsi que la création des commandes pour son utilisation. Je les range dans un tableau pour mieux retrouver les informations:

Informations des user stories	Conception de la base de données	Commandes SQL
Compte utilisateur (participant) : Nom + Email unique + création de compte	X	X
Événements : Nom + Description + Date + Lieu + création d'événement	X	X
Inscription d'un utilisateur à un événement (table de jointure)	X	X
Commande SQL de tous les événements auxquels est inscrit un utilisateur donné		X
Organisateur (niveau de compte utilisateur) + création d'événements (fk_cree_par)	X	X
Commande SQL de tous les utilisateurs inscrits à un événement donné		X

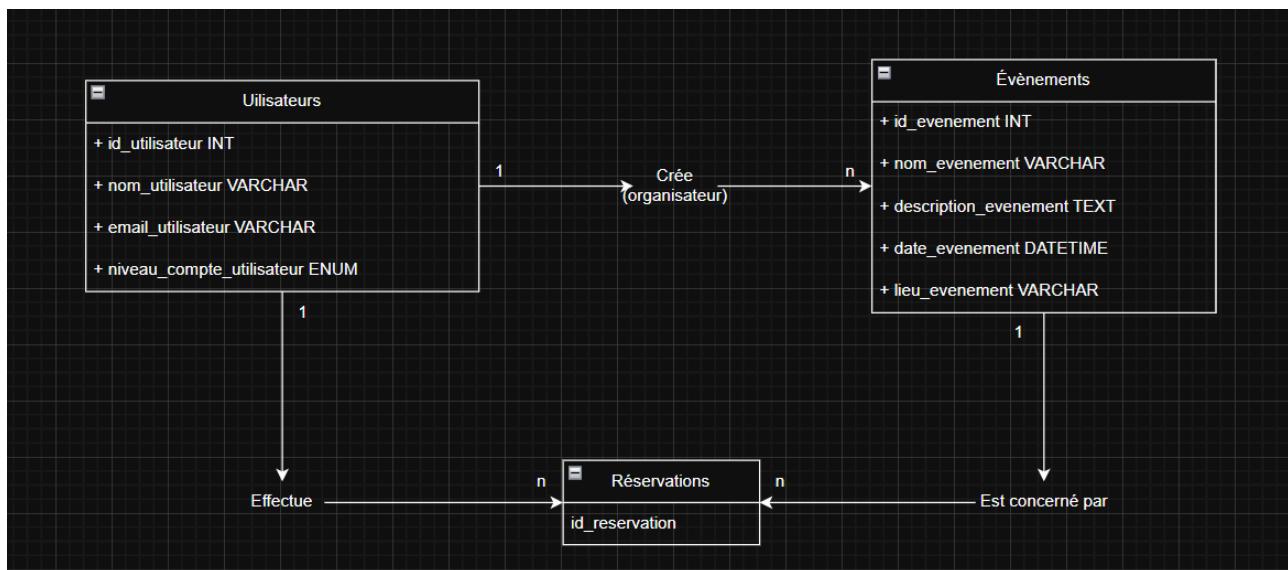
DOSSIER PROFESSIONNEL (DP)

Commande SQL pour compter les utilisateurs inscrits à chaque évènement pour classer par plus populaire		X
Administrateur (niveau de compte utilisateur) + commande SQL pour compter tous les utilisateurs	X	X
Commande SQL de mise à jour d'un événement donné		X
Commande SQL de suppression d'une réservation		X
Commande SQL de suppression d'un événement		X

Maintenant que j'ai préparé les informations avec lesquelles je vais travailler, je commence la conception de la base de données. Je choisis trois tables de ces informations :

- Table utilisateur : nom, email (unique), niveau de compte (Participant, Organisateur, Administrateur)
- Table événement : nom, description, date, lieu, créé par
- Table de réservation à un événement : utilisateur, événement

Je réalise donc un diagramme UML de la base de données afin d'avoir une représentation visuelle. Je définis le nom des tables, des champs ainsi que leur type.



J'ai pris la décision de faire une seule table "Utilisateurs" incluant les différents niveaux de compte plutôt que de faire une table "Participants", une table "Organisateurs" et une table "Administrateurs" car ils partagent tous les mêmes informations de compte (nom + email), les organisateurs et administrateurs peuvent également réaliser des réservations et enfin cela me permet d'alléger un peu la base.

DOSSIER PROFESSIONNEL (DP)

C'est pour cette raison que "niveau_compte_utilisateur" est en type "ENUM", car il ne sera possible que d'être un des trois niveaux de compte ('Participant', 'Organisateur', 'Administrateur').

Pour la table "Évènements", j'ai choisi de donner le type "TEXT" à "description_evenement" car il ne m'est pas possible de savoir à l'avance quelle taille feront les descriptions d'événements.

Avec cette conception, je remplis les besoins présents dans mon tableau de besoins réalisés à partir des user stories.

Maintenant que j'ai terminé la conception, je vais rédiger un document en ".sql" à exécuter pour créer la base de données et toutes les tables. Tout d'abord, je crée la base de données SI elle n'existe PAS déjà sur mon hébergeur. Je précise également qu'après la création, il faut utiliser cette base de données. Ainsi, toutes les commandes qui vont suivre seront exécutées au sein de la base.

```
1  -- Création de la base données
2  CREATE DATABASE IF NOT EXISTS gestion_evenements_db;
3  USE gestion_evenements_db;
```

Maintenant que la base est créée, je peux créer la table "utilisateurs" avec les informations que j'ai préparées dans la phase de conception. Pour plus de clarté, je rajoute "_utilisateur" à la fin de chaque nom de colonne, ainsi il sera plus simple de travailler les données. Je précise également quelles colonnes sont obligatoires à remplir lors d'un ajout de ligne; ici tous les champs sauf "id_utilisateur" car il est rempli automatiquement lors de la création d'une ligne. Enfin, pour cette table ainsi que pour toutes les autres, je précise qu'elles doivent être créées avec le moteur "INNODB". Sans ça, les clés étrangères ne seront pas visuellement fonctionnelles dans la base de données. J'ai mis la valeur par défaut 'Participant' pour "niveau_compte_utilisateur"; étant le rôle de base lors de la création d'un compte, il n'aura pas besoin d'être renseigné à chaque ajout d'utilisateur.

```
5  -- Création de la table utilisateurs
6  CREATE TABLE IF NOT EXISTS utilisateurs(
7    id_utilisateur INT PRIMARY KEY AUTO_INCREMENT,
8    nom_utilisateur VARCHAR(100) NOT NULL,
9    email_utilisateur VARCHAR(200) NOT NULL UNIQUE,
10   niveau_compte_utilisateur ENUM('Participant','Organisateur','Administrateur') NOT NULL DEFAULT 'Participant'
11 )ENGINE=INNODB;
```

Je crée maintenant la table "evenements". Les colonnes de cette table sont écrites de la même façon que la précédente. Sur cette table, j'ajoute une contrainte de clé étrangère sur la colonne "cree_par". Avec cette clé étrangère, je lie un utilisateur à la création d'un évènement par son id. Il sera donc très simple de retrouver l'organisateur de l'événement. Je précise qu'en cas de tentative de déletion de l'utilisateur organisateur, une erreur préviendra de cette déletion tant qu'il sera lié à au moins un événement. En cas de mise à jour des informations de l'utilisateur, l'événement est lui aussi mis à jour en conséquence. Cette clé étrangère pourrait par exemple permettre à des participants de rechercher tous les autres événements créés par cet organisateur s'ils le souhaitent. Une autre façon de nommer cette clé étrangère aurait pu être par exemple "fk_id_utilisateur" pour exprimer clairement la clé étrangère.

DOSSIER PROFESSIONNEL (DP)

```
13  -- Création de la table Évènements
14  CREATE TABLE IF NOT EXISTS evenements(
15    id_evenement INT PRIMARY KEY AUTO_INCREMENT,
16    nom_evenement VARCHAR(200) NOT NULL,
17    description_evenement TEXT NOT NULL,
18    date_evenement DATETIME NOT NULL,
19    lieu_evenement VARCHAR(100),
20    cree_par INT,
21    FOREIGN KEY (cree_par)
22      REFERENCES utilisateurs(id_utilisateur)
23      ON DELETE RESTRICT
24      ON UPDATE CASCADE
25  )ENGINE=INNODB;
```

Je crée maintenant la table de jointure entre les deux tables. La table de jointure permet de lier N utilisateurs à N événements. Par exemple, un utilisateur peut être inscrit à trois événements, et quatre utilisateurs peuvent être inscrits à un seul événement. J'inclue donc deux contraintes de clé étrangère pour lier les deux tables ensemble, avec un id unique pour la réservation. J'ai choisi de mettre les clés étrangères en cascade lors de la suppression soit d'un événement, soit d'un utilisateur, car la réservation n'a plus lieu d'être si un des deux est supprimé.

```
27  -- Création de la table Réservations
28  CREATE TABLE IF NOT EXISTS reservations(
29    id_reservation INT PRIMARY KEY AUTO_INCREMENT,
30    fk_id_utilisateur INT,
31    FOREIGN KEY (fk_id_utilisateur)
32      REFERENCES utilisateurs(id_utilisateur)
33      ON DELETE CASCADE
34      ON UPDATE CASCADE,
35    fk_id_evenement INT,
36    FOREIGN KEY (fk_id_evenement)
37      REFERENCES evenements(id_evenement)
38      ON DELETE CASCADE
39      ON UPDATE CASCADE
40  )ENGINE=INNODB;
```

DOSSIER PROFESSIONNEL (DP)

Pour tester le bon fonctionnement de la base de données, je crée des données tests (5 utilisateurs, 5 événements et 10 réservations).

5 utilisateurs :

- Petit | antoine.petit@mail.fr | Administrateur
- Gronier | ng93@hotmail.fr | Organisateur
- Prudenti | jean.p@gmail.com | Participant
- Baux | bbbasque@yahoo.fr | Participant
- Cuoq | val.c@outlook.fr | Organisateur

5 événements :

- Soirée cinéma | Marathon de la trilogie originale Star Wars, venez nombreux et costumés si vous le pouvez ! | 2025-08-30 20:00:00 | Grand Rex Paris | 5
- Session karting familiale | Venez en famille à cet après-midi karting ! Apportez des boissons et goûters et nous pourrons passer une bonne après midi chaleureuse et conviviale avec des courses endiablées ! | 2025-08-14 14:30:00 | Circuit de Karting d'Aix en Provence | 2
- Concert de Linkin Park | Vous êtes fan mais vous ne voulez pas aller seul à un concert ? Accompagnez nous ce weekend au concert de Linkin Park, vous serez bien entouré et encadré pour passer le meilleur moment possible ! | 2025-08-02 19:30:00 | Arène de Nîmes | 5
- Brunch vegan | Vous êtes vegan ou même vous souhaitez découvrir cette alimentation bio et éthique, venez vous joindre à nous pour un délicieux brunch sous le soleil du sud ! | 2025-08-19 11:00:00 | Parc du soleil Bayonne | 2
- Finale de coupe | Venez regarder la finale de la compétition de football du moment, apportez des chips, pizza et sodas et supportez l'équipe de votre choix dans une ambiance bon enfant ! | 2025-07-29 21:00:00 | Bar central Lyon | 5

10 réservations :

- Prudenti | Soirée cinéma
- Baux | Brunch vegan
- Cuoq | Session karting familiale
- Petit | Soirée cinéma
- Gronier | Finale de coupe
- Gronier | Concert de Linkin Park
- Baux | Session karting familiale
- Prudenti | Concert de Linkin Park
- Petit | Brunch vegan
- Baux | Finale de coupe

Je rédige maintenant mes commandes d'insertion pour enregistrer ces informations. Je commence avec les insertions d'utilisateurs. J'utilise "INSERT INTO [table](colonnes à remplir) VALUES(valeurs à enregistrer dans la colonne correspondante)" pour chacune des tables. Ici je renseigne les niveaux de compte de chaque utilisateur, même les 'Participant'. J'aurais pu ne pas remplir cette colonne pour ces utilisateurs étant déjà la valeur par défaut.

DOSSIER PROFESSIONNEL (DP)

```
42 -- Peupler la table utilisateurs
43 INSERT INTO utilisateurs(nom_utilisateur, email_utilisateur, niveau_compte_utilisateur) VALUES('Petit', 'antoine.petit@mail.fr',
44 'Administrateur');
44 INSERT INTO utilisateurs(nom_utilisateur, email_utilisateur, niveau_compte_utilisateur) VALUES('Gronier', 'ng93@hotmail.fr', 'Organisateur');
45 INSERT INTO utilisateurs(nom_utilisateur, email_utilisateur, niveau_compte_utilisateur) VALUES('Prudenti', 'jean.p@gmail.com', 'Participant');
46 INSERT INTO utilisateurs(nom_utilisateur, email_utilisateur, niveau_compte_utilisateur) VALUES('Baux', 'bbbasse@yahoo.fr', 'Participant');
47 INSERT INTO utilisateurs(nom_utilisateur, email_utilisateur, niveau_compte_utilisateur) VALUES('Cuoq', 'val.c@outlook.fr', 'Organisateur');
```

Je rédige de la même façon pour les insertions d'événements. La clé étrangère est, comme défini plus tôt, un simple entier. Une fois la ligne créée, cet entier renverra automatiquement à l'utilisateur à l'id correspondant.

```
49 -- Peupler la table evenements
50 INSERT INTO evenements(nom_evenement, description_evenement, date_evenement, lieu_evenement, cree_par) VALUES("Soirée cinéma", "Marathon de la
51 trilogie originale Star Wars, venez nombreux et costumés si vous le pouvez !", "2025-08-30 20:00:00", "Grand Rex Paris", 5);
51 INSERT INTO evenements(nom_evenement, description_evenement, date_evenement, lieu_evenement, cree_par) VALUES("Session karting familiale",
52 "Venez en famille à cette après midi karting ! Apportez des boissons et gouters et nous pourrons passer une bonne après midi chaleureuse et
53 convivial avec des courses endiablées !", "2025-08-14 14:30:00", "Circuit de Karting d'Aix en Provence", 2);
52 INSERT INTO evenements(nom_evenement, description_evenement, date_evenement, lieu_evenement, cree_par) VALUES("Concert de Linkin Park", "Vous
53 êtes fan mais vous ne voulez pas aller seul à un concert ? Accompagnez nous ce weekend au concert de Linkin Park, vous serez bien entouré et
54 encadré pour passer le meilleur moment possible !", "2025-08-02 19:30:00", "Arène de Nîmes", 5);
53 INSERT INTO evenements(nom_evenement, description_evenement, date_evenement, lieu_evenement, cree_par) VALUES("Brunch vegan", "Vous êtes vegan
54 ou même vous souhaitez découvrir cette alimentation bio et éthique, venez vous joindre à nous pour un délicieux brunch sous le soleil du sud
54 !", "2025-08-19 11:00:00", "Parc du soleil Bayonne", 2);
54 INSERT INTO evenements(nom_evenement, description_evenement, date_evenement, lieu_evenement, cree_par) VALUES("Finale de coupe", "Venez
regarder la finale de la compétition de football du moment, apportez des chips, pizza et sodas et supportez l'équipe de votre choix dans une
ambiance bon enfant !", "2025-07-29 21:00:00", "Bar central Lyon", 5);
```

Enfin je peuple les réservations. J'entre simplement les id des utilisateurs et événements liés.

```
56 -- Peupler la table reservations
57 INSERT INTO reservations(fk_id_utilisateur, fk_id_evenement) VALUES (3, 1);
58 INSERT INTO reservations(fk_id_utilisateur, fk_id_evenement) VALUES (4, 4);
59 INSERT INTO reservations(fk_id_utilisateur, fk_id_evenement) VALUES (5, 2);
60 INSERT INTO reservations(fk_id_utilisateur, fk_id_evenement) VALUES (1, 1);
61 INSERT INTO reservations(fk_id_utilisateur, fk_id_evenement) VALUES (2, 5);
62 INSERT INTO reservations(fk_id_utilisateur, fk_id_evenement) VALUES (2, 3);
63 INSERT INTO reservations(fk_id_utilisateur, fk_id_evenement) VALUES (4, 2);
64 INSERT INTO reservations(fk_id_utilisateur, fk_id_evenement) VALUES (3, 3);
65 INSERT INTO reservations(fk_id_utilisateur, fk_id_evenement) VALUES (1, 4);
66 INSERT INTO reservations(fk_id_utilisateur, fk_id_evenement) VALUES (4, 5);
```

Maintenant que la base de données est créée et peuplée, je vais pouvoir rédiger les requêtes pour valider les user stories renseignées plus tôt.

US1 : Commande simple d'insertion de données, on en a déjà utilisé pour peupler la base de données. Cette fois-ci je ne mets pas de valeur à "niveau_compte_utilisateur" car on crée un compte de participant et c'est la valeur par défaut.

```
1 -- User story 1 :
2 INSERT INTO utilisateurs (nom_utilisateur, email_utilisateur) VALUES ("Utilisateur", "utilisateur@gmail.com");
```

US2 : Commande de sélection permettant de récupérer les informations de tous les évènements. Cette commande permettra d'afficher tous les évènements disponibles à l'utilisateur pour qu'il puisse s'y inscrire.

DOSSIER PROFESSIONNEL (DP)

```
4  -- User story 2 :  
5  SELECT nom_evenement, description_evenement, date_evenement, lieu_evenement FROM evenements;
```

US3 : Commande d'insertion directement dans la table de jointure, permet de lier l'utilisateur et l'événement auquel il veut s'inscrire. La table de jointure nous permet d'inscrire plusieurs utilisateurs à un seul évènement et un seul utilisateur à plusieurs évènements.

```
7  -- User story 3 :  
8  INSERT INTO reservations (fk_id_utilisateur, fk_id_evenement) VALUES (6, 4);
```

US4 : Commande de sélection où on ne récupère que le nom de l'évènement. Je récupère tous les évènements qui ont été réservés par un utilisateur donné (dans l'exemple ici l'utilisateur qui correspond à l'id numéro 4) grâce à une jointure entre la table de réservations et celle d'événements. La jointure se fait grâce à la clé étrangère. J'utilise "as" pour donner un alias à chaque table pour ne pas avoir à ré-écrire leur nom complet à chaque fois.

```
10 -- User story 4 :  
11 SELECT e.nom_evenement FROM evenements as e INNER JOIN reservations as r ON r.fk_id_evenement = e.id_evenement WHERE r.fk_id_utilisateur = 4;
```

US5 : Commande d'insertion de donnée dans la table d'événements, permet d'ajouter l'événement et toutes ses informations à la liste.

```
13 -- User story 5 :  
14 INSERT INTO evenements (nom_evenement, description_evenement, date_evenement, lieu_evenement, cree_par) VALUES ("Pique nique en plein air", "Rejoignez nous avec vos produits maison et frais pour un charmant pique nique sous l'ombre d'un arbre !", "2025-07-29 12:00:00", "Square Victor Hugo", 5);
```

US6 : Même logique de commande que pour la US4 mais cette fois on récupère les informations utilisateurs liées à un événement donné (dans l'exemple on prend l'événement qui correspond à l'id numéro 4) grâce à une jointure entre la table de réservations et celle des utilisateurs.

```
16 -- User story 6 :  
17 SELECT u.nom_utilisateur, u.email_utilisateur FROM utilisateurs as u INNER JOIN reservations as r ON u.id_utilisateur = r.fk_id_utilisateur WHERE r.fk_id_evenement = 4
```

US7 : Sur cette commande, je sélectionne le nom des événements et le compte d'utilisateur inscrit au dit événement. Je lis ensuite la table de réservations à la table d'événements. La méthode "COUNT" compte toutes les lignes. Je demande de grouper par nom d'événement avec "GROUP BY" afin de les classer de l'événement ayant le plus d'inscriptions à celui qui en a le moins avec "ORDER BY".

```
19 -- User story 7 :  
20 SELECT e.nom_evenement, COUNT(*) FROM evenements as e INNER JOIN reservations as r ON e.id_evenement = r.fk_id_evenement GROUP BY e.nom_evenement ORDER BY COUNT(*) DESC
```

US8 : Sélection simple de tous les utilisateurs inscrits en affichant tous les éléments de la table utilisateurs.

```
22 -- User story 8 :  
23 SELECT * from utilisateurs;
```

US9 : Mise à jour de données avec une commande "UPDATE ... SET...". Dans l'exemple, on imagine que l'événement a changé sa programmation et l'organisateur met à jour les informations de description.

DOSSIER PROFESSIONNEL (DP)

```
25 -- User story 9 :  
26 UPDATE evenements SET description_evenement = "Marathon de la prélogie Star Wars, venez nombreux et costumés si vous le pouvez !" WHERE id_evenement = 1;
```

US10 : Ici on a deux possibilités. La première est de supprimer la réservation en prenant l'id de l'utilisateur qui annule sa réservation ainsi que l'id de l'événement dont il se retire. La deuxième est plus simple, on supprime simplement la réservation grâce à son id. Les deux fonctionnent et dépendent de la manière dont l'interface est programmée. La première permet de supprimer toutes les réservations d'un utilisateur à un évènement (en cas de duplicita) et la seconde permet de supprimer une seule ligne (si il n'y pas de duplicita).

```
28 -- User story 10 :  
29 DELETE FROM reservations WHERE fk_id_utilisateur = 3 AND fk_id_evenement = 1;  
30 | OU  
31 DELETE FROM reservations WHERE id_reservation = 1;
```

US11 : On supprime simplement l'événement grâce à une commande “DELETE FROM” en précisant bien l'événement à supprimer grâce à son id. Comme j'ai réglé la table de réservations pour une suppression en cascade, les réservations des utilisateurs sont automatiquement supprimées.

```
33 -- User story 11 :  
34 DELETE FROM evenements WHERE id_evenement = 1;
```

Ainsi, je remplis toutes les user stories fournies et complète le projet.

2. Précisez les moyens utilisés :

Pour la conception du schéma de la base de données, je me suis rendu sur <https://app.diagrams.net/> où j'ai utilisé les informations que j'ai récupérées dans les user stories. Pour réaliser et valider tous les tests de requêtes SQL, j'ai utilisé “PhpMyAdmin” au sein de WAMPserver, je me suis connecté avec le nom d'utilisateur “root” et pas de mot de passe, j'ai ensuite cliquer sur “importer” en haut de l'écran pour importer le fichier de base de données et l'implémenter en local. J'ai ensuite cliqué sur “SQL” au sein de la base de données et j'y ai réalisé tous les tests pour valider les user stories.

3. Avec qui avez-vous travaillé ?

DOSSIER PROFESSIONNEL (DP)

J'ai travaillé en totale autonomie sur ce projet.

4. Contexte

Nom de l'entreprise, organisme ou association - La plateforme

Chantier, atelier, service - Formation Développeur Web & Web Mobile

Période d'exercice - Du 28/07/2025 au 28/07/2025

5. Informations complémentaires (*facultatif*)

Cliquez ici pour taper du texte.

DOSSIER PROFESSIONNEL (DP)

Activité-type 2 Développer la partie back-end d'une application web ou web mobile sécurisée

Exemple n°6-7 - Développer des composants d'accès aux données SQL et NoSQL & Développer des composants métier côté serveur

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

On imagine une commande fictive.

Je suis missionné pour construire l'architecture backend d'une application dédiée à la gestion de compétitions e-sport, un domaine en pleine expansion.

Cette plateforme permettra à des joueurs passionnés de s'inscrire, de créer ou de rejoindre des équipes, et de participer à des tournois organisés autour de jeux vidéo populaires.

Les organisateurs pourront quant à eux publier des événements compétitifs, définir les règles des tournois et gérer les inscriptions des équipes.

Enfin, les administrateurs du système auront une vue d'ensemble sur les utilisateurs, les compétitions et les performances globales de la plateforme, leur permettant d'assurer le bon fonctionnement de l'application, de modérer les contenus et d'analyser les statistiques de participation.

Le système devra donc gérer plusieurs rôles avec des droits différenciés, garantir l'intégrité des données échangées, et permettre des opérations fiables sur les équipes, les utilisateurs et les tournois.

Des user stories étaient également fournies.

1. Création de compte

- En tant que nouveau participant, je veux pouvoir créer un compte avec email et mot de passe, afin de pouvoir rejoindre des équipes et participer à la plateforme.

2. Connexion

- En tant qu'utilisateur enregistré, je veux pouvoir me connecter avec mon email et mot de passe, afin d'accéder à mes fonctionnalités personnalisées.

3. Déconnexion

- En tant qu'utilisateur connecté, je veux pouvoir me déconnecter de mon compte, afin de sécuriser mon accès.

4. Modifier mon profil

- En tant qu'utilisateur, je veux pouvoir modifier mes informations personnelles, afin de garder mon profil à jour.

5. Créer une équipe

DOSSIER PROFESSIONNEL (DP)

- En tant que joueur, je veux pouvoir créer ma propre équipe, afin de participer à des compétitions avec mes coéquipiers.

6. Rejoindre une équipe

- En tant que joueur, je veux pouvoir rejoindre une équipe existante, afin de jouer en groupe.

7. Gérer les membres de mon équipe

- En tant que capitaine, je veux pouvoir ajouter ou retirer des joueurs de mon équipe, afin d'organiser efficacement les membres.

8. Créer un tournoi

- En tant qu'organisateur, je veux pouvoir créer un tournoi avec nom, jeu, date et règles, afin d'organiser des compétitions.

9. Modifier un tournoi

- En tant qu'organisateur, je veux pouvoir modifier les détails d'un tournoi que j'ai créé, afin de corriger ou mettre à jour les informations.

10. Supprimer un tournoi

- En tant qu'organisateur ou admin, je veux pouvoir supprimer un tournoi, afin de retirer un événement annulé ou terminé.

11. Incrire une équipe à un tournoi

- En tant que joueur/capitaine, je veux pouvoir inscrire mon équipe à un tournoi, afin de participer officiellement.

12. Lister les tournois ouverts

- En tant que joueur, je veux voir tous les tournois ouverts à l'inscription, afin de choisir où participer.

13. Voir les équipes inscrites à un tournoi

- En tant qu'organisateur, je veux voir la liste des équipes inscrites à mes tournois, afin de suivre la participation.

14. Supprimer une équipe (admin uniquement)

- En tant qu'administrateur, je veux pouvoir supprimer une équipe, afin de gérer les cas de non-respect des règles ou abus.

15. Voir les statistiques de participation

DOSSIER PROFESSIONNEL (DP)

- En tant qu'administrateur, je veux voir le nombre d'équipes inscrites à chaque tournoi, afin d'analyser la fréquentation.

16. Gérer les rôles utilisateurs

- En tant qu'administrateur, je veux attribuer ou modifier les rôles (joueur, capitaine, organisateur, admin), afin de contrôler les permissions.

17. Consulter le détail d'une équipe

- En tant que joueur ou organisateur, je veux consulter la composition et les informations d'une équipe, afin de mieux connaître mes adversaires ou coéquipiers.

18. Consulter mes inscriptions à des tournois

- En tant que joueur, je veux voir la liste des tournois auxquels mon équipe est inscrite, afin de gérer mon calendrier.

À partir de ces user stories, j'en ressors ces besoins :

- Page de création de compte (email + mot de passe)
- Page de connexion (email + mot de passe)
- Fonction de déconnexion (fermeture de session)
- Page de modification des informations (page paramètres de compte)
- Page de création d'équipe
- Page pour rejoindre une équipe existante
- Page de gestion d'équipe (ajout/suppression de joueurs)
- Page de création de tournoi
- Page de modification de tournoi
- Fonction de suppression de tournoi
- Page d'inscription d'équipe à un tournoi
- Page de listage des tournois existants et actifs
- Page d'équipes inscrites à un tournoi donné
- Fonction de suppression d'équipe (admin uniquement) (page liste équipe)
- Fonction de listage des tournois avec nombre d'équipe inscrite (admin uniquement) (page liste tournoi)
- Page de modification de rôles membre (admin uniquement) (page liste membres)
- Page d'information d'équipe (joueurs inscrits etc)
- Page liste des tournois auxquels une équipe donnée est inscrite. (page info team)

Un fichier m'est fourni contenant les requêtes SQL nécessaires à la création et au peuplement de la base de données.

Je lance WAMPserver, je me rend sur "PhPMyAdmin" et je me connecte avec le nom d'utilisateur "root" sans mot de passe. Je clique sur "Importer" en haut de l'écran et j'importe le fichier fourni.

Maintenant que la base de données est créée, je retourne à l'accueil de "PhPMyAdmin" et je clique sur "Comptes utilisateurs", je crée un nouvel utilisateur avec comme nom d'utilisateur "esportAdmin" et comme mot de passe "esport1234". Ce mot de passe est facile à retenir pour l'entièreté de la phase de test et développement en local, en cas de déploiement du projet, il devra être changé pour un mot de passe plus sécurisé.

DOSSIER PROFESSIONNEL (DP)

Une fois l'utilisateur créé, je clique dessus dans la liste des utilisateurs, en haut de l'écran je clique sur "Base de données", je trouve la base de données créée plus tôt, et j'ajoute des droits complets à l'utilisateur sur cette base de données.

Je peux maintenant me déconnecter du profil "root" et me connecter avec les informations que j'ai renseigné plus tôt. Grâce à ce compte utilisateur, je pourrais facilement voir le résultat de mes actions lors du développement du projet.

Je commence par réaliser l'architecture du projet. Le dossier "components" contient les éléments réutilisables soit header et footer, le dossier "config" gère la connexion à la base données et contient toutes les fonctions du projet. En rangeant les fonctions dans un seul fichier, je peux plus facilement réaliser la maintenance du projet et allèger les autres fichiers du projet. Je range ensuite toutes les actions CRUD dans des dossiers selon quels "tables" et quels éléments de ma base de données seront appelés. Par exemple, le dossier "tournaments" contient toutes les actions concernant l'affichage de plusieurs tournois, la création, la modification et la suppression d'un tournoi, tandis que le dossier "tournament" contient les actions d'affichage et d'inscription à un seul tournoi.

DOSSIER PROFESSIONNEL (DP)

```
✓ CP6-7
  ✓ components
    • footer.php
    • header.php
  ✓ config
    • db.php
    • fonctions.php
  > style
  ✓ teams
    • create.php
    • delete-player.php
    • delete-team.php
    • join.php
    • list.php
    • team.php
  ✓ tournament
    • list.php
    • register.php
  ✓ tournaments
    • create.php
    • delete.php
    • list.php
    • update.php
  ✓ users
    • disconnect.php
    • list.php
    • login.php
    • signup.php
    • update-user.php
    • update.php
```

DOSSIER PROFESSIONNEL (DP)

Je commence par mon fichier de connexion à la base de données. Je définis d'abord des variables pour pouvoir facilement modifier le chemin d'accès ou le mot de passe à utiliser par exemple. Je crée ensuite une variable "pdo", cette variable me servira à chaque requête envoyée à la base de données.

```
1 <?php
2
3 $host = 'localhost';
4 $db = 'esports';
5 $username = 'esportAdmin';
6 $password = 'esport1234';
7 try {
8     $pdo = new PDO(
9         dsn: "mysql:host=$host;dbname=$db;charset=utf8",
10        username: $username,
11        password: $password,
12        options: [PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION]
13    );
14 } catch (PDOException $e) {
15     die('Erreur de connexion : ' . $e->getMessage());
16 }
17
```

Maintenant que la connexion à ma base de données est réalisée, je m'occupe des fonctions de compte utilisateur. Je crée les fonctions au fur et à mesure du développement du reste de l'application pour être sûr que ma fonction soit faite sur mesure pour l'action voulue.

J'utilise les sessions pour gérer l'état de connexion de l'utilisateur. Ici, si la variable de session "connected" est vide, l'utilisateur n'est alors pas connecté, je lui affiche donc un formulaire de création de compte. Si la variable de session n'est pas vide, l'utilisateur est alors déjà connecté et je lui affiche un message de redirection vers l'accueil. À la ligne 76, il y a une variable "error". Pour l'instant, cette variable n'affiche rien, mais elle servira à afficher une erreur si elle survient lors de l'envoi du formulaire.

DOSSIER PROFESSIONNEL (DP)

```
61 <body>
62     <?php include_once '../components/header.php'; ?>
63
64     <main>
65         <!-- Si utilisateur pas encore connecté -->
66         <?php if (empty($_SESSION['connected'])): ?>
67             <h1>Créer un compte</h1>
68             <form action="" method="post">
69                 <label for="username">Nom d'utilisateur</label>
70                 <input type="text" name="username" id="username" required>
71                 <label for="email">Email</label>
72                 <input type="email" name="email" id="email" required>
73                 <label for="password">Mot de passe</label>
74                 <input type="password" name="password" id="password" required>
75                 <input type="submit" name="submit" value="Créer un compte">
76                 <?= $error ?>
77             </form>
78         <?php endif;
79         // Si utilisateur déjà connecté
80         if (!empty($_SESSION['connected']) && $_SESSION['connected'] === true): ?>
81             <div style="text-align:center">
82                 <h1>Vous êtes connecté !</h1>
83                 <a href="../index.php">Revenir à l'accueil</a>
84             </div>
85         <?php endif; ?>
86     </main>
87
88     <?php include_once '../components/footer.php'; ?>
89 </body>
```

Pour l'envoi du formulaire, j'utilise plusieurs sécurités qui seront récurrentes sur la plupart des pages contenant des formulaires dans le projet. Ici, je démarre évidemment les sessions, puis j'initialise la variable "error" mentionnée plus tôt et j'initialise également une variable "isError" qui me permet de m'assurer qu'aucune erreur n'a lieu avant l'appel à la fonction de création de compte. Le formulaire est soumis avec une méthode "POST", je vérifierais donc chaque variable avec le préfixe "_POST". Je commence par vérifier l'envoi du formulaire avec la variable "submit", si elle est vide c'est que le formulaire n'a pas été envoyé, sinon c'est qu'il a bien été soumis. Je vérifie ensuite si les champs du formulaire ont tous été remplis, dans le cas contraire j'affiche une erreur demandant à l'utilisateur de bien remplir tous les champs. J'utilise ensuite "require_once" pour appeler mon fichier de fonctions une seule fois dans le fichier. J'utilise "htmlspecialchars" pour protéger la base de données contre des injections malveillantes et "trim" pour retirer de potentiels espaces au début et à la fin du contenu des champs. Ensuite, je valide le format de l'email avec "filter_var" et le filtre "FILTER_VALIDATE_EMAIL". Enfin, j'utilise des fonctions pour vérifier que l'adresse mail et le nom d'utilisateur n'existe pas déjà dans la base de données auquel cas j'affiche quel champ créé l'erreur.

DOSSIER PROFESSIONNEL (DP)

```
1 <?php
2 session_start();
3 $error = '';
4 $isError = false;
5 // Vérification que le formulaire a été envoyé
6 if (!empty($_POST['submit'])) {
7     // Vérification que les champs sont tous remplis
8     if (!empty($_POST['username']) && !empty($_POST['email']) && !empty($_POST['password'])) {
9         require_once '../config/fonctions.php';
10        $username = htmlspecialchars(string: trim(string: $_POST['username']));
11        $email = htmlspecialchars(string: trim(string: $_POST['email']));
12        // Vérification que l'adresse mail est valide
13        if (filter_var(value: $email, filter: FILTER_VALIDATE_EMAIL)) {
14            // Vérification de l'existence du compte
15            if (verifyExistingEmail(email: $email) || verifyExistingUsername(usernameUser: $username)) {
16                $isError = true;
17                $errorEmail = '';
18                $errorUsername = '';
19                if (verifyExistingEmail(email: $email)) {
20                    $errorEmail = 'L\'email est déjà pris. ';
21                }
22                if (verifyExistingUsername(usernameUser: $username)) {
23                    $errorUsername = 'Le nom d\'utilisateur est déjà pris. ';
24                }
25                $error = '<p style="color:red;">' . $errorEmail . $errorUsername . '</p>';
26            }
27        } else {
28            $isError = true;
29            $error = '<p style="color:red;">Cette adresse email est invalide !</p>';
30        }
31    } else {
32        $isError = true;
33        $error = '<p style="color:red;">Tous les champs doivent être remplis !</p>';
34    }
}
```

Je vais détailler ici les commandes “verifyExistingEmail” et “verifyExistingUsername”. La fonction “verifyExistingEmail” prend pour paramètre “email” et “id” qui prend une valeur par défaut de 0. Comme dans chaque fonction, j’appelle mon fichier “db.php” pour pouvoir accéder à la variable “pdo”. Je déclare une variable “stmt”, raccourci de “statement”, pour préparer ma requête SQL. J’utilise “SELECT” sur l’id dans la table “users” où l’adresse email vaut une variable “email” et id doit être différent d’une variable “id”. Maintenant que ma requête est prête, je l’exécute en y attribuant les variables reçues lors de l’appel de la fonction. Ici, je définis une variable qui va prendre pour valeur le nombre de lignes renvoyées par la requête, mais j’aurais pu directement renvoyer le nombre de ligne sans variable. Avec cette fonction, si la variable “existingUsers” vaut plus de 0, cela signifie qu’un compte existe déjà avec cette adresse email ailleurs que pour la valeur de l’id envoyée. Ce paramètre “id” sera utile pour la mise à jour des informations utilisateurs. Si l’utilisateur essayant d’utiliser cette adresse mail est déjà l’utilisateur la possédant, alors la fonction renverra 0 et permettra la sauvegarde des informations utilisateurs.

DOSSIER PROFESSIONNEL (DP)

```
18 // Vérification de l'existence du mail
 5 references
19 function verifyExistingEmail($email, $id = 0): mixed
20 {
21     require 'db.php';
22     $stmt = $pdo->prepare('SELECT id FROM users WHERE email = :email AND id != :id');
23     $stmt->execute([
24         "email" => $email,
25         "id" => $id
26     ]);
27     $existingUsers = $stmt->rowCount();
28     return $existingUsers;
29 }
```

Même fonctionnement pour la fonction “verifyExistingUsername”.

```
31 // Vérification de l'existence du nom d'utilisateur
 4 references
32 function verifyExistingUsername($usernameUser, $id = 0): mixed
33 {
34     require 'db.php';
35     $stmt = $pdo->prepare('SELECT id FROM users WHERE username = :username AND id != :id');
36     $stmt->execute([
37         "username" => $usernameUser,
38         "id" => $id
39     ]);
40     $existingUsers = $stmt->rowCount();
41     return $existingUsers;
42 }
```

Je reviens sur ma page de création de compte. Si aucune erreur n'est détectée, je fais la même manipulation avec “htmlspecialchars” que plus tôt sauf si le mot de passe. J'appelle donc ma fonction “createAccount” avec le nom d'utilisateur, l'adresse email et le mot de passe renseignés. Si la fonction renvoie au moins une ligne, alors la création de compte a fonctionné, sinon une erreur est affichée.

```
36 if ($isError === false) {
37     $username = htmlspecialchars(string: trim(string: $_POST['username']));
38     $email = htmlspecialchars(string: trim(string: $_POST['email']));
39     $password = trim(string: $_POST['password']);
40
41     $accountCreated = createAccount(usernameUser: $username, email: $email, password: $password);
42     if ($accountCreated >= 1) {
43         $error = '<p style="color:green">Votre compte a bien été créé</p>
44         <a href="/cp6-7/users/login.php">Vous connecter à votre compte</a>';
45     } else {
46         $error = '<p style="color:red;">Une erreur est survenue lors de la création de votre compte</p>';
47     }
48 }
49 }
```

DOSSIER PROFESSIONNEL (DP)

Pour la fonction “createAccount”, elle prend en paramètre un nom d’utilisateur, une adresse email et un mot de passe. Je commence par hacher le mot de passe avec la méthode “password_hash”. J’utilise l’algorithme de hachage par défaut, soit Bcrypt. Ensuite je prépare ma requête comme dans les précédentes fonctions. Ma requête insert un nouvel élément dans la table “users” avec les valeurs renseignées lors de l’exécution. J’exécute ma requête et renvoie le nombre de lignes renvoyées par la base de données.

```
3 // Création de compte
4 function createAccount($usernameUser, $email, $password): mixed
5 {
6     require 'db.php';
7     $passwordHash = password_hash(password: $password, algo: PASSWORD_DEFAULT);
8     $stmt = $pdo->prepare('INSERT INTO users (username, email, password_hash) VALUES (:username, :email, :password)');
9     $stmt->execute([
10         "username" => $usernameUser,
11         "email" => $email,
12         "password" => $passwordHash
13     ]);
14
15     return $stmt->rowCount();
16 }
```

Je valide donc la première user story.

Je fais donc la page de connexion utilisateur. Les sécurités de la page de connexion sont les mêmes que la page de création de compte à quelques exceptions près, par exemple la fonction “verifyExistingEmail” sert à vérifier l’existence du compte et renvoie une erreur si aucun compte n’est trouvé. La fonction “connectUser” prend en paramètres une adresse email et un mot de passe. Je prépare une requête qui sélectionne l’id et le mot de passe haché dans la table “users” où la colonne “email” correspond à l’email renseigné. J’exécute la requête et utilise la méthode “fetch” pour récupérer un seul élément. Ensuite j’utilise la méthode “password_verify” pour vérifier que le mot de passe renseigné correspond au mot de passe haché correspondant à l’adresse email renseignée. Si la vérification est réussie, je renvoie l’ID de l’utilisateur récupéré, sinon je renvoie “false”.

```
44 // Fonction de connexion
45 function connectUser($email, $password): mixed
46 {
47     require 'db.php';
48     $stmt = $pdo->prepare('SELECT id, password_hash FROM users WHERE email = :email');
49     $stmt->execute(["email" => $email]);
50     $user = $stmt->fetch();
51
52     if (password_verify(password: $password, hash: $user['password_hash'])) {
53         return $user['id'];
54     } else {
55         return false;
56     }
57 }
```

Pour la page de connexion, j’appelle donc la fonction “connectUser” avec l’adresse email et le mot de passe renseigné par l’utilisateur. Si la fonction renvoie “false”, un message d’erreur est affiché pour mot de passe incorrect, sinon, j’attribue la valeur “true” à la variable de session “connected” et l’ID renvoyé par la fonction à la variable de session “id_user”. Ces deux variables vont servir pour beaucoup d’affichage conditionnel sur l’application.

DOSSIER PROFESSIONNEL (DP)

```
28 if ($isError === false) {  
29     $email = htmlspecialchars(string: trim(string: $_POST['email']));  
30     $password = trim(string: $_POST['password']);  
31  
32     if (connectUser(email: $email, password: $password) === false) {  
33         $error = '<p style="color:red;">Mot de passe incorrect</p>';  
34     } else {  
35         $_SESSION['connected'] = true;  
36         $_SESSION['id_user'] = connectUser(email: $email, password: $password);  
37     }  
38 }  
39 }
```

Je valide donc la deuxième user story.

Je prépare maintenant la déconnexion utilisateur. Pour la déconnexion, c'est très simple, je démarre la session, puis je la détruis pour retirer toutes les variables actives, puis enfin je redirige l'utilisateur sur la page d'accueil.

```
1 <?php  
2 session_start();  
3 session_destroy();  
4 header(header: 'Location: /cp6-7/index.php');  
5
```

Je valide la troisième user story.

Pour la mise à jour utilisateur, je commence par vérifier si l'utilisateur est connecté. Si l'utilisateur est connecté, j'initialise toutes mes variables, messages d'erreur et je récupère les informations de l'utilisateur connecté avec la fonction "getUser" et l'ID enregistré en session.

DOSSIER PROFESSIONNEL (DP)

```
1 <?php
2 session_start();
3 if (isset($_SESSION['id_user'])) {
4     include '../config/fonctions.php';
5     $error = '';
6     $isError = false;
7     $userInfo = getUser(id: $_SESSION['id_user']);
8     $idUser = $_SESSION['id_user'];
9     $usernameUser = $userInfo['username'];
10    $messageUsername = '';
11    $emailUser = $userInfo['email'];
12    $messageEmail = '';
13    $messageNewPass = '';
14 }
```

La fonction "getUser" prend pour paramètre un ID. Je prépare une requête simple qui me permet de récupérer le nom d'utilisateur, l'adresse email et le rôle de l'utilisateur (participant, organisateur ou admin) où l'id correspond à l'ID renseigné. J'utilise ensuite la méthode "fetch" pour récupérer les informations et les renvoyer.

```
59 // Fonction pour récupérer les informations d'un utilisateur
60 // 10 references
61 function getUser($id): mixed
62 {
63     require 'db.php';
64     $stmt = $pdo->prepare('SELECT username, email, role FROM users WHERE id = :id');
65     $stmt->execute(["id" => $id]);
66     return $stmt->fetch();
```

Dans le cas où l'utilisateur n'est pas connecté, un message lui indique qu'il n'est pas connecté avec un lien qui renvoie à la page d'accueil. Si il est connecté, un formulaire est pré-rempli avec les informations de l'utilisateur.

DOSSIER PROFESSIONNEL (DP)

```
107 | <!-- si l'utilisateur est connecté -->
108 | <?php if (!empty($_SESSION['connected']) && $_SESSION['connected'] === true): ?>
109 | <h1>Profil de <?= $usernameUser ?></h1>
110 | <form action="" method="post">
111 |   <label for="username">Nom d'utilisateur</label>
112 |   <input type="text" name="username" id="username" value="<?= $usernameUser ?>">
113 |   <?= $messageUsername ?>
114 |   <label for="email">Email</label>
115 |   <input type="email" name="email" id="email" value="<?= $emailUser ?>">
116 |   <?= $messageEmail ?>
117 |   <label for="old-password">Mot de passe</label>
118 |   <input type="password" name="old-password" id="old-password">
119 |   <p style="margin-bottom: 20px; margin-top: -15px; font-size: 14px;">Veuillez entrer le mot de passe pour toute modification</p>
120 |   <label for="new-password">Nouveau mot de passe</label>
121 |   <input type="password" name="new-password" id="new-password">
122 |   <?= $messageNewPass ?>
123 |   <input type="submit" name="submit" value="Mettre à jour">
124 |   <?= $error ?>
125 | </form>
126 | <?php endif;
```

Ensuite, j'ai des sécurités similaires à la page de création de compte.

J'initialise ensuite une variable "accountUpdated" qui me permet de savoir si le compte de l'utilisateur a été mis à jour ou non. Ensuite, pour chaque information du compte, je vérifie que le champ n'est pas vide, et que l'information est bien à mettre à jour (si le nom d'utilisateur envoyé avec le formulaire est le même que le nom d'utilisateur récupéré dans la base de données, on ne lance pas la fonction de mise à jour). Si le champ n'est pas vide, j'appelle une fonction de mise à jour du champ. Si la fonction me renvoie au moins une ligne, alors la valeur a été mise à jour et je notifie l'utilisateur, puis j'ajoute 1 à ma variable "accountUpdated". Enfin, une fois toutes les fonctions appelées, si la variable "accountUpdated" est supérieure à 0, j'affiche à l'utilisateur que son profil a bien été mis à jour.

```
52 | $accountUpdated = 0;
53 |
54 | // Mise à jour du nom d'utilisateur
55 | if (!empty($username) && $username != $usernameUser) {
56 |   $usernameUpdated = updateUsername(id: $idUser, newUsername: $username);
57 |   if ($usernameUpdated > 0) {
58 |     $messageUsername = '<p style="color:green; margin-top:-15px; margin-bottom:20px">Le nom d\'utilisateur a été mis à jour.</p>';
59 |     $accountUpdated += 1;
60 |   } else {
61 |     $messageUsername = '<p style="color:red; margin-top:-15px; margin-bottom:20px">Erreur de mise à jour du nom d\'utilisateur.</p>';
62 |   }
63 |
64 |
65 | // Mise à jour de l'email
66 | if (!empty($email) && $email != $emailUser) {
67 |   $emailUpdated = updateEmail(id: $idUser, newEmail: $email);
68 |   if ($emailUpdated > 0) {
69 |     $messageEmail = '<p style="color:green; margin-top:-15px; margin-bottom:20px">L\'email a été mis à jour.</p>';
70 |     $accountUpdated += 1;
71 |   } else {
72 |     $messageEmail = '<p style="color:red; margin-top:-15px; margin-bottom:20px">Erreur de mise à jour de l\'email.</p>';
73 |   }
74 |
75 |
76 | // Mise à jour du mot de passe
77 | if (!empty($password)) {
78 |   $passwordUpdated = updatePassword(id: $idUser, newPassword: $password);
79 |   if ($passwordUpdated > 0) {
80 |     $messageNewPass = '<p style="color:green; margin-top:-15px; margin-bottom:20px">Le mot de passe a été mis à jour.</p>';
81 |     $accountUpdated += 1;
82 |   } else {
83 |     $messageNewPass = '<p style="color:red; margin-top:-15px; margin-bottom:20px">Erreur de mise à jour du mot de passe.</p>';
84 |   }
85 |
86 |
87 | if ($accountUpdated >= 1) {
88 |   $error = '<p style="color:green">Votre compte a bien été mis à jour</p>';
```

DOSSIER PROFESSIONNEL (DP)

Les trois fonctions fonctionnent de la même manière à part la fonction de mise à jour du mot de passe qui a une ligne supplémentaire pour hacher le nouveau mot de passe. Chaque fonction prend en paramètre l'ID de l'utilisateur et la valeur à mettre à jour (nom d'utilisateur, adresse email, mot de passe).

```
68 // Fonction pour mettre à jour le pseudo utilisateur
 1 reference
69 function updateUsername($id, $newUsername): mixed
70 {
71     require 'db.php';
72     $stmt = $pdo->prepare('UPDATE users SET username = :username WHERE id = :id');
73     $stmt->execute([
74         "username" => $newUsername,
75         "id" => $id
76     ]);
77
78     return $stmt->rowCount();
79 }
```

Grâce à ces fonctions, je valide la 4ème user story.

Je peux donc passer à la 5ème user story, qui est la création d'une équipe. Pour la partie visuelle, je réalise un affichage conditionnel. Si l'utilisateur est connecté, j'affiche un formulaire demandant simplement le nom de l'équipe, avec un bouton de confirmation. Pour ce qui est du traitement des données, je vérifie simplement que le champ est rempli. J'utilise ensuite une fonction qui vérifie l'existence d'une autre équipe utilisant déjà ce nom.

```
117 // Fonction pour vérifier l'existence d'une équipe grâce au nom
 1 reference
118 function verifyExistingTeam($teamName): mixed{
119     require 'db.php';
120     $stmt = $pdo->prepare('SELECT * FROM teams WHERE name = :name');
121     $stmt->execute(["name" => $teamName]);
122
123     return $stmt->rowCount();
124 }
```

Si le nom d'équipe est disponible, j'appelle une fonction de création d'équipe prenant comme paramètres le nom de l'équipe et l'ID de l'utilisateur la créant. Cette fonction enregistre l'équipe, et récupère l'ID de l'équipe créée afin d'appeler une autre fonction pour attraper au créateur le rôle de capitaine.

DOSSIER PROFESSIONNEL (DP)

```
135 // Fonction pour créer une équipe et y ajouter le créateur en tant que capitaine  
1 reference  
136 function createTeam($teamName, $idUser): mixed{  
137     require 'db.php';  
138     $stmt = $pdo->prepare('INSERT INTO teams (name) VALUES (:name)');  
139     $stmt->execute(['name' => $teamName]);  
140  
141     if($stmt->rowCount() > 0){  
142         $stmt2 = $pdo->prepare('SELECT id FROM teams WHERE name = :name');  
143         $stmt2->execute(['name' => $teamName]);  
144         $team = $stmt2->fetch();  
145  
146         if(isset($team['id'])){  
147             addPlayerToTeam(idTeam: $team['id'], idUser: $idUser, role: 'captain');  
148         }  
149     }  
}
```

Je présenterai la fonction d'ajout d'un joueur à une équipe lors de la présentation de la prochaine user story. La 5ème user story est validée.

La 6ème user story demande de pouvoir rejoindre une équipe existante. Pour ça, j'ai besoin de deux éléments : une page listant toutes les équipes, et une page permettant de rejoindre une équipe choisie. Pour la première, j'utilise une fonction pour récupérer toutes les équipes créées.

```
108 // Fonction pour récupérer toutes les équipes créées  
1 reference  
109 function getAllTeams(): mixed  
110 {  
111     require 'db.php';  
112     $stmt = $pdo->prepare('SELECT id, name FROM teams');  
113     $stmt->execute();  
114     return $stmt->fetchAll();  
115 }
```

À partir de la liste d'équipe récupérées, j'utilise un `forEach` pour afficher toutes les équipes dans un tableau. Ce tableau comporte 2 colonnes pour un utilisateur classique. La première colonne comporte le nom de l'équipe, la seconde colonne contient un bouton permettant d'accéder à la page équipe et, si l'utilisateur est connecté ET ne fait pas partie de l'équipe, un bouton permettant de rejoindre l'équipe.

DOSSIER PROFESSIONNEL (DP)

```
$teams = getAllTeams();
foreach ($teams as $team):
?>
<tr>
  <td><?= $team['name'] ?></td>
  <td>
    <a href="team.php?id=<?= $team['id'] ?>">Voir l'équipe</a>
    <?php if(!empty($_SESSION['id_user']) && verifyPlayerInTeam(idTeam: $team['id'], idUser: $_SESSION['id_user']) === false): ?>
      |
      <a href="join.php?id=<?= $team['id'] ?>">Rejoindre l'équipe</a>
    <?php endif; ?>
  </td>
  <?php if(!empty($userInfo) && $userInfo['role'] === 'admin'): ?>
    <td><a href="delete-team.php?id=<?= $team['id'] ?>">Bannir l'équipe</a></td>
  <?php endif; ?>
</tr>
<?php endforeach; ?>
```

Pour cet affichage, je crée une fonction permettant de vérifier la présence du joueur dans l'équipe.

```
185 // Fonction pour vérifier la présence d'un joueur dans une équipe
  5 references
186 function verifyPlayerInTeam($idTeam, $idUser): mixed{
187   require 'db.php';
188   $stmt = $pdo->prepare('SELECT role_in_team FROM team_members WHERE user_id = :user_id AND team_id = :team_id');
189   $stmt->execute([
190     "user_id" => $idUser,
191     "team_id" => $idTeam
192   ]);
193
194   return $stmt->fetch();
195 }
```

Ensuite, si l'utilisateur ne fait pas partie de l'équipe, un lien est affiché lui permettant de rejoindre l'équipe. Sur la page pour rejoindre l'équipe, je vérifie si l'utilisateur est connecté, et je l'ajoute à l'équipe grâce à l'ID en variable GET dans l'URL. Pour ça, j'utilise la même fonction que pour ajouter le capitaine lors de la création de l'équipe vue plus tôt.

```
154 // Fonction pour ajouter un joueur à une équipe
  2 references
155 function addPlayerToTeam($idTeam, $idUser, $role = 'member'): mixed{
156   require 'db.php';
157   $stmt = $pdo->prepare('INSERT INTO team_members (user_id, team_id, role_in_team) VALUES (:user, :team, :role)');
158   $stmt->execute([
159     "user" => $idUser,
160     "team" => $idTeam,
161     "role" => $role
162   ]);
163
164   return $stmt->rowCount();
165 }
```

Je renvoie ensuite soit un message de validation, soit un message d'erreur selon le résultat de l'inscription de l'utilisateur. Je valide donc la user story 6.

La user story numéro 7 demande de pouvoir gérer les membres de l'équipe lorsque l'utilisateur est capitaine. Pour pouvoir gérer l'équipe, je crée une page d'équipe (page accessible par le bouton dans le tableau affichant toutes les équipes). Lors du chargement de la page, j'utilise plusieurs fonctions : une fonction pour récupérer les informations de l'équipe, une fonction pour récupérer tous les joueurs présents dans l'équipe et enfin une fonction permettant de vérifier si l'utilisateur fait partie de l'équipe ainsi que récupérer son rôle.

DOSSIER PROFESSIONNEL (DP)

```
167 // Fonction pour récupérer les informations d'une équipe
168 1reference
169 function getTeamInfo($idTeam): mixed{
170     require 'db.php';
171     $stmt = $pdo->prepare('SELECT name, created_at FROM teams WHERE id = :id');
172     $stmt->execute(['id' => $idTeam]);
173     return $stmt->fetch();
174 }
175
176 // Fonction pour récupérer tous les joueurs d'une équipe
177 1reference
178 function getAllPlayersFromTeam($idTeam): mixed{
179     require 'db.php';
180     $stmt = $pdo->prepare('SELECT u.id, u.username, tm.role_in_team FROM team_members as tm INNER JOIN users as u ON tm.user_id = u.id WHERE tm.team_id = :id ORDER BY tm.role_in_team DESC');
181     $stmt->execute(['id' => $idTeam]);
182     return $stmt->fetchAll();
183 }
184
185 // Fonction pour vérifier la présence d'un joueur dans une équipe
186 5 references
187 function verifyPlayerInTeam($idTeam, $idUser): mixed{
188     require 'db.php';
189     $stmt = $pdo->prepare('SELECT role_in_team FROM team_members WHERE user_id = :user_id AND team_id = :team_id');
190     $stmt->execute([
191         "user_id" => $idUser,
192         "team_id" => $idTeam
193     ]);
194     return $stmt->fetch();
195 }
```

Grâce à ces 3 fonctions, je peux gérer tout l'affichage de ma page. J'affiche les informations d'équipe en premier lieu ainsi que le capitaine de l'équipe en réalisant un `forEach` sur la liste des joueurs pour trouver le joueur possédant le rôle de capitaine. Ensuite, j'affiche les joueurs inscrits dans l'équipe dans un tableau contenant 2 colonnes : le nom du joueur ainsi que son rôle dans l'équipe. Si l'utilisateur connecté est capitaine de l'équipe, une troisième colonne s'affiche, nommée "action", permettant au capitaine d'exclure un joueur de rang "membre". Si le joueur affiché est un capitaine, le bouton "exclure" ne s'affiche pas. Ainsi, un capitaine ne peut pas être exclu de sa propre équipe, même par un autre capitaine. J'affiche également un second tableau, contenant les informations de tous les tournois auxquels l'équipe est inscrite ainsi que les informations des tournois. Cela permet un meilleur suivi pour les membres de l'équipe. Si le capitaine clique sur le bouton "exclure" dans le tableau des joueurs inscrits, il est envoyé sur une page permettant de gérer l'exclusion du joueur sélectionné. Une liste de vérification s'exécute au chargement de la page. Dans l'ordre : je vérifie que l'utilisateur est connecté; s'il est connecté, je vérifie que l'équipe existe bien dans la base de données et si elle existe bien, je vérifie que l'utilisateur fait partie de l'équipe et en est capitaine. Si il l'est, je vérifie que l'utilisateur à exclure est bien renseigné: si il est renseigné, je vérifie qu'il fait bien partie de l'équipe dont il faut l'exclure et qu'il n'en est pas capitaine, et enfin si toutes ces conditions sont remplies, j'appelle une fonction pour exclure le joueur.

DOSSIER PROFESSIONNEL (DP)

```
2 session_start();
3 $error = '';
4 // Vérification qu'un utilisateur est connecté
5 if(!empty($_SESSION['connected']) && $_SESSION['connected'] === true){
6     $idCaptain = $_SESSION['id_user'];
7     // Vérification que l'équipe est bien renseignée et existe
8     include_once '../config/fonctions.php';
9     if(!empty($_GET["id_team"]) && verifyExistingTeamById(idTeam: $_GET["id_team"]) > 0){
10        $idTeam = $_GET['id_team'];
11        // Vérification que l'utilisateur essayant d'exclure un joueur fait partie de l'équipe et est bien le capitaine
12        $captain = verifyPlayerInTeam(idTeam: $idTeam, idUser: $idCaptain);
13        if($captain != false && $captain['role_in_team'] === 'captain'){
14            // Vérification si le joueur à exclure a été renseigné
15            if(!empty($_GET['id_player'])) {
16                $idPlayer = $_GET['id_player'];
17                $playerToKick = verifyPlayerInTeam(idTeam: $idTeam, idUser: $idPlayer);
18                // Vérification si le joueur à exclure fait partie de l'équipe et n'est pas capitaine
19                if($playerToKick != false && $playerToKick['role_in_team'] != 'captain'){
20                    $isKicked = kickPlayer(idTeam: $idTeam, idPlayer: $idPlayer);
21                    // Vérification si le joueur a bien été exclu
22                    if($isKicked > 0){
23                        $error = '<p style="color:green;">Le joueur a bien été exclu</p>';
24                    } else {
25                        $error = '<p style="color:red;">Une erreur est survenue</p>';
26                    }
27                } else {
28                    $error = '<p style="color:red;">Le joueur ne fait pas partie de l'équipe ou en est un capitaine</p>';
29                }
30            } else {
31                $error = '<p style="color:red;">L\'identité du joueur à exclure n'a pas été renseignée</p>';
32            }
33        } else {
34            $error = '<p style="color:red;">Vous ne faites pas partie de cette équipe ou vous n'en êtes pas le capitaine</p>';
35        }
36    } else {
37        $error = '<p style="color:red;">L'équipe n'a pas été renseignée ou n'existe pas </p>';
38    }
39 } else {
40     $error = '<p style="color:red;">Vous n'êtes pas connecté.</p>';
41 }
```

Pour la fonction d'exclusion, j'utilise une simple commande "DELETE" sur ma base de données.

```
197 // Fonction pour exclure un joueur d'une équipe
198 // reference
199 function kickPlayer($idTeam, $idPlayer): mixed{
200     require 'db.php';
201     $stmt = $pdo->prepare('DELETE FROM team_members WHERE user_id = :user_id AND team_id = :team_id');
202     $stmt->execute([
203         "user_id" => $idPlayer,
204         "team_id" => $idTeam
205     ]);
206     return $stmt->rowCount();
207 }
```

Grâce à ces éléments, je valide la 7ème user story, la 17ème user story demandant de pouvoir consulter les détails d'une équipe, et la 18ème user story demandant de pouvoir consulter les tournois auxquels l'équipe a été inscrite.

La 8eme user story demande de pouvoir créer un tournoi si l'utilisateur est au moins un organisateur. Pour ça, je crée une page de création de tournoi. Pour afficher le formulaire de création, j'ajoute un affichage conditionnel. L'utilisateur doit être connecté, et il doit avoir un rôle utilisateur différent du rôle de membre (donc organisateur ou

DOSSIER PROFESSIONNEL (DP)

administrateur). Si il remplit les conditions, l'utilisateur peut renseigner le nom du tournoi, le jeu, une description du tournoi, et les dates de début et de fin du tournoi. Lorsque le formulaire est soumis, je vérifie que chaque champ est bien rempli. Ensuite, j'enregistre le contenu de tous les champs dans des variables, notamment les dates que j'enregistre au bon format pour pouvoir mieux les afficher. Une fois les champs récupérés, je vérifie qu'aucun tournoi n'existe avec le nom utilisé grâce à une fonction.

```
218 // Fonction pour vérifier l'existence d'un tournoi actif grâce au nom
219 function verifyExistingTournamentByName($name, $id = 0): mixed{
220     require 'db.php';
221     $stmt = $pdo->prepare('SELECT * FROM tournaments WHERE name = :name AND start_date >= CURRENT_TIMESTAMP AND id != :id');
222     $stmt->execute([
223         "name" => $name,
224         "id" => $id
225     ]);
226
227     return $stmt->fetch();
228 }
```

Ensuite, je vérifie que les dates de début et de fin du tournoi concordent. Les dates de début et de fin doivent être ultérieures à la date de création du tournoi, et la date de début doit être antérieure à la date de fin du tournoi.

```
27 // Vérification que les dates sont bien égales ou ultérieures à la date actuelle
28 if(strtotime(datetime: $startDate) < strtotime(datetime: date(format: 'Y-m-d')) || strtotime(datetime: $endDate) < strtotime(datetime: date('Y-m-d'))){
29     $isError = true;
30     $errorDate = '<p style="color:red; margin-top: -15px; margin-bottom:15px;">Veuillez choisir une date ultérieure à la date actuelle</p>';
31 }
32 // Vérification que la date de début est bien antérieure à la date de fin
33 if(!$isError != true && strtotime(datetime: $startDate) > strtotime(datetime: $endDate)) {
34     $isError = true;
35     $errorDate = '<p style="color:red; margin-top: -15px; margin-bottom:15px;">La date de fin ne peut pas être antérieure à la date de début.</p>';
36 }
```

Si aucune erreur n'est trouvée, j'utilise une fonction de création de tournoi ayant pour paramètres le nom, le jeu, la description, les dates de début et fin, et l'ID de l'organisateur du tournoi.

```
230 // Fonction de création de tournoi
231 function createTournament($name, $game, $description, $start, $end, $userId): mixed{
232     require 'db.php';
233     $stmt = $pdo->prepare('INSERT INTO tournaments(name, game, description, start_date, end_date, organizer_id) VALUES (:tournament_name , :game , :tournament_description , :starting , :ending , :id)');
234     $stmt->execute([
235         "tournament_name" => $name,
236         "game" => $game,
237         "tournament_description" => $description,
238         "starting" => $start,
239         "ending" => $end,
240         "id" => $userId
241     ]);
242
243     return $stmt->rowCount();
244 }
```

Je valide donc la user story numéro 8.

La user story numéro 9 demande de pouvoir modifier un tournoi déjà existant. Pour ça, je crée un affichage similaire à celui de la liste d'équipe, modifiant le bouton "Rejoindre l'équipe" pour un bouton de modification de tournoi si l'utilisateur connecté en est le créateur. L'affichage des tournois contient toutes les informations de tournoi enregistrées lors de sa création à l'exception de l'identité de l'organisateur. J'utilise un affichage conditionnel pour le nombre de tournois affichés. Si l'utilisateur connecté est un administrateur, tous les tournois de la base de données sont affichés, sinon, uniquement les tournois actifs (soit les tournois dont la date de début est ultérieure à la date actuelle) sont affichés. Lorsque tous les tournois sont affichés, un compte des équipes inscrites est affiché à l'administrateur, permettant de voir les tournois les plus populaires.

DOSSIER PROFESSIONNEL (DP)

```
331 // Fonction pour récupérer tous les tournois et compter les équipes inscrites
332 1 référence
333 function getAllTournaments(): mixed{
334     require 'db.php';
335     $stmt = $pdo->prepare('SELECT t.id as id, t.name as name, t.game as game, t.description as description, t.start_date as start_date, t.end_date as end_date, t.organizer_id as organizer_id,
336     COUNT(*) as inscrits FROM registrations as r INNER JOIN tournaments as t ON r.tournament_id = t.id GROUP BY t.name ORDER BY COUNT(*) DESC;');
337     $stmt->execute();
338 }
```

```
209 // Fonction pour récupérer tous les tournois actifs
210 1 référence
211 function getAllActiveTournaments(): mixed{
212     require 'db.php';
213     $stmt = $pdo->prepare('SELECT * FROM tournaments WHERE start_date >= CURRENT_TIMESTAMP;');
214     $stmt->execute();
215
216     return $stmt->fetchAll();
217 }
```

En cliquant sur le bouton de modification de tournoi, on obtient les mêmes sécurités et les mêmes champs que lors de la création du tournoi, la seule sécurité supplémentaire est la vérification que l'utilisateur connecté est bien l'organisateur du tournoi à modifier. Lors de la soumission du formulaire, les mêmes vérifications que lors de la création du tournoi sont exécutées, puis une fonction de mise à jour est appelée avec les mêmes paramètres que lors de la création, à l'exception que l'ID fourni ici est celui du tournoi à modifier et non de l'organisateur du tournoi, car cette information restera inchangée.

```
255 // Fonction pour mettre à jour un tournoi
256 1 référence
257 function updateTournament($id, $name, $game, $description, $start, $end): mixed{
258     require 'db.php';
259     $stmt = $pdo->prepare('UPDATE tournaments SET name = :name, game = :game, description = :description, start_date = :start, end_date = :end WHERE id = :id');
260     $stmt->execute([
261         "name" => $name,
262         "game" => $game,
263         "description" => $description,
264         "start" => $start,
265         "end" => $end,
266         "id" => $id
267     ]);
268
269     return $stmt->rowCount();
270 }
```

Je valide donc la 9eme user story ainsi que la 12eme user story demandant de lister les tournois ouverts et la 15eme user story demandant de voir les statistiques d'équipes inscrites à un tournoi.

La 10eme user story demande de supprimer un tournoi. Pour la suppression de tournoi, j'ajoute un bouton de suppression sous le formulaire de modification de tournoi. Ce bouton redirige sur une page de suppression contenant des sécurités s'assurant que l'utilisateur est connecté, qu'il est bien un organisateur et qu'il est bien l'organisateur du tournoi à supprimer. J'utilise ensuite une fonction de suppression de tournoi.

```
271 // Fonction pour supprimer un tournoi
272 1 référence
273 function deleteTournament($id): mixed{
274     require 'db.php';
275     $stmt = $pdo->prepare('DELETE FROM tournaments WHERE id = :id');
276     $stmt->execute(["id" => $id]);
277
278     return $stmt->rowCount();
279 }
```

DOSSIER PROFESSIONNEL (DP)

Ainsi, je valide la 10eme user story.

La 11eme user story demande de permettre d'inscrire une équipe à un tournoi. Pour cela, je crée une page d'informations de tournoi accessible depuis la liste des tournois. Cette page affiche les informations du tournoi (nom, jeu, description et dates) ainsi qu'un tableau contenant toutes les équipes inscrites ainsi que le nom du capitaine de l'équipe inscrite pour faciliter la communication entre les équipes lors du tournoi. Sous ce tableau, un bouton "Rejoindre le tournoi" est disponible. En cliquant sur ce bouton, si l'utilisateur est bien connecté, il peut voir un menu déroulant contenant toutes les équipes dont il est capitaine, ainsi qu'un bouton pour valider l'inscription. Lors de la validation, je vérifie que l'utilisateur est bien capitaine de l'équipe qu'il essaie d'inscrire. Je vérifie également si l'équipe est déjà inscrite ou non à ce tournoi grâce à une fonction.

```
289 // Fonction pour vérifier si une équipe est inscrite à un tournoi
290 1 référence
291 function verifyTeamInTournament($idTeam, $idTournament): mixed{
292     require 'db.php';
293     $stmt = $pdo->prepare('SELECT id FROM registrations WHERE team_id = :team_id AND tournament_id = :tournament_id');
294     $stmt->execute([
295         "team_id" => $idTeam,
296         "tournament_id" => $idTournament
297     ]);
298     return $stmt->fetchAll();
299 }
```

Si ces conditions sont bien remplies, j'utilise une fonction pour inscrire l'équipe au tournoi.

```
301 // Fonction pour inscrire une équipe
302 1 référence
303 function registerTeamInTournament($idTeam, $idTournament): mixed{
304     require 'db.php';
305     $stmt = $pdo->prepare('INSERT INTO registrations (team_id, tournament_id) VALUES (:team_id, :tournament_id)');
306     $stmt->execute([
307         "team_id" => $idTeam,
308         "tournament_id" => $idTournament
309     ]);
310     return $stmt->rowCount();
311 }
```

Je valide ainsi la user story numéro 11 et numéro 13 demandant de voir les équipes inscrites à un tournoi.

Pour la user story numéro 14, un administrateur doit pouvoir supprimer une équipe existante. Pour cela, sur la page de la liste des équipes, j'ajoute une colonne affichée uniquement à un utilisateur connecté ayant le rôle d'administrateur, contenant une action de bannissement d'équipe. En cliquant sur ce bouton, l'administrateur est redirigé sur une page de confirmation. Lors du chargement, je vérifie que l'utilisateur est connecté, que l'équipe à supprimée existe bien, que l'utilisateur connecté est bien un administrateur, et enfin j'utilise une fonction de suppression d'équipe.

DOSSIER PROFESSIONNEL (DP)

```
322 // Fonction pour supprimer une équipe
1 reference
323 function deleteTeam($id): mixed{
324     require 'db.php';
325     $stmt = $pdo->prepare('DELETE FROM teams WHERE id = :id');
326     $stmt->execute(['id' => $id]);
327
328     return $stmt->rowCount();
329 }
```

Je valide donc la user story numéro 14.

Pour la user story numéro 16, il est demandé à ce que l'administrateur puisse modifier les rôles des utilisateurs, par exemple pour donner les droits de création de tournoi à un utilisateur n'ayant que le rôle de membre. Pour cela, j'ajoute un affichage conditionnel dans le header permettant d'accéder à une liste de tous les utilisateurs non administrateurs existants. Pour ça je crée une fonction récupérant tous les utilisateurs correspondants à cette description.

```
340 // Fonction pour récupérer tous les membres non administrateurs
1 reference
341 function getAllNonAdminMembers(): mixed{
342     require 'db.php';
343     $stmt = $pdo->prepare('SELECT id, username, email, role FROM users WHERE role != "admin"');
344     $stmt->execute();
345
346     return $stmt->fetchAll();
347 }
```

J'affiche ainsi tous les utilisateurs dans un tableau, avec leur nom, adresse email, rôle et un lien de modification de rôle. En cliquant sur le bouton de modification, l'administrateur est dirigé sur une page contenant un menu déroulant avec les différents rôles disponibles (membre, organisateur, administrateur) et un bouton de validation. Lors de la validation, je vérifie que le rôle attribué correspond bien aux rôles disponibles pour éviter toute modification par l'utilisateur, je vérifie que l'utilisateur connecté est bien un administrateur, je vérifie ensuite que l'utilisateur à modifier existe bien et qu'il n'est pas administrateur lui-même. Une fois toutes ces vérifications faites, j'utilise une fonction pour mettre à jour le rôle utilisateur.

```
349 // Fonction pour mettre à jour un rôle utilisateur
1 reference
350 function updateUserRole($id, $role): mixed {
351     require 'db.php';
352     $stmt = $pdo->prepare('UPDATE users SET role = :role WHERE id = :id');
353     $stmt->execute([
354         'role' => $role,
355         'id' => $id
356     ]);
357
358     return $stmt->rowCount();
359 }
```

DOSSIER PROFESSIONNEL (DP)

Je valide donc la user story numéro 16.

Ainsi, je valide la totalité des user stories renseignées. Le projet étant conséquent et à réaliser dans une période courte, je n'ai réalisé qu'un style très basique, raison pour laquelle je ne me suis pas attardé dessus lors de mes explications et présentations de fonctionnalités.

Le projet est entièrement fonctionnel et conforme aux requêtes du client.

2. Précisez les moyens utilisés :

Pour ce projet, j'ai hébergé la base de données en local grâce à WAMP server, et je l'ai configuré et rempli en SQL sur PHPmyAdmin. L'entièreté du projet a été réalisée sans framework, avec du PHP simple. Pour le style, j'ai utilisé du CSS pour permettre de bien visualiser le travail que je réalisais. Le projet a été versionné et publié sur Github grâce à des commits réguliers.

3. Avec qui avez-vous travaillé ?

Pour ce projet, j'ai travaillé en totale autonomie.

4. Contexte

Nom de l'entreprise, organisme ou association ▶ La plateforme

Chantier, atelier, service ▶ Formation Développeur Web & Web Mobile

DOSSIER PROFESSIONNEL (DP)

Période d'exercice | Du 30/07/2025 au 01/08/2025

5. Informations complémentaires (*facultatif*)

Cliquez ici pour taper du texte.

Activité-type 2 Développer la partie front-end d'une application web ou web mobile sécurisée

Exemple n°8 Documenter le déploiement d'une application dynamique web ou web mobile

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Tout d'abord, je dois choisir un projet existant à documenter. Je trouve donc un projet, déjà sur Github, de gestionnaire d'habitudes. Ce projet est divisé en deux repository, un pour le front et un pour le back. Je vais donc commencer par créer un repository principal contenant les deux sous repository, puis documenter le déploiement du back et enfin celui du front.

Afin de rendre la documentation utilisable à quiconque souhaite utiliser l'application, je crée un fichier README.md dans mon dossier. Il va me servir à rassembler toutes les informations utiles et pratiques du projet, comme la version de React utilisée pour le front ou de Node et Express pour le back, les stacks techniques, les fonctionnalités et toutes les instructions d'installation et de déploiement. Ici, je vais m'intéresser rapidement à l'installation, et plus en détail au déploiement.

Pour l'installation du back, j'indique que npm, Node.js et MongoDB sont requis ainsi que leurs versions minimales. J'indique ensuite comment configurer MongoDB en local ou bien avec Atlas (en ligne) pour l'utiliser, puis j'indique comment cloner le repository principal du projet, comment se déplacer dans le dossier du back, et comment l'installer grâce à git et npm.

DOSSIER PROFESSIONNEL (DP)

Prérequis

- Node.js >= 22.14.0
- npm >= 11.6.0
- MongoDB (local ou MongoDB Atlas)

Configuration MongoDB

1. MongoDB locale

- Installer MongoDB : <https://www.mongodb.com/docs/manual/installation/>
- Démarrer le serveur : `mongodb`
- URL de connexion : `mongodb://localhost:27017/habits`

2. MongoDB Atlas (optionnel)

- Créer un compte Atlas : <https://www.mongodb.com/cloud/atlas>
- Créer un cluster et une base de données `habits`
- Récupérer l'URI de connexion

Installation

```
# Cloner le projet  
git clone --recurse-submodules https://github.com/AntoinePetit/habits  
cd habits/back  
  
# Installer les dépendances  
npm install
```

Je fourni la configuration des variables d'environnement ainsi que le démarrage du projet en local.

Configuration

Créer un fichier `.env` ou modifier le fichier `.env.example` en `.env`.

```
PORT=3000  
MONGO_URI=<votre_uri_mongodb>
```

Utilisation en local

```
node .\app.js
```

Par défaut, l'API tourne sur <http://localhost:3000>

DOSSIER PROFESSIONNEL (DP)

Une fois l'installation en local expliquée, j'explique à l'utilisateur comment déployer son back-end en ligne à partir d'un fork du repository sur Render. J'explique donc d'abord comment réaliser un fork du back.

Déploiement de l'application

Faire un fork

- Se rendre sur le [repository](#)
- Cliquer sur "Fork"
- Nommer le fork
- Cliquer sur "Create fork"

Une fois le fork réalisé, j'explique chaque étape de l'hébergement du back-end sur Render avec précision, afin que l'utilisateur puisse facilement le reproduire.

Hébergement

- Se rendre sur [Render](#)
- Se connecter en utilisant un compte GitHub
- Cliquer sur "Add new"
- Choisir "Web Service"
- Choisir le repository correspondant
- Choisir les paramètres suivants :
 - Language : Node
 - Region : Frankfurt (EU Central) -- Si vous voulez être héberger en Europe
 - Build Command : `npm install`
 - Start Command : `node ./app.js`
 - Instance Type : Free
 - Environment Variables > Add from .env > Copier ici votre fichier .env créé plus tôt > Add variables
- Cliquer sur "Deploy Web Service"

Lors de mon propre déploiement, j'ai remarqué un problème potentiel avec Render, qui est que le serveur sera mis en veille lorsqu'il n'est pas utilisé pendant un temps. J'indique donc à l'utilisateur comment pallier ce problème.

DOSSIER PROFESSIONNEL (DP)

Render mettra votre serveur back "en veille" si il n'est pas utilisé après un temps assez court. Pour éviter de devoir attendre le redémarrage du serveur à chaque utilisation, voici les étapes à suivre :

- Se rendre sur [UptimeRobot](#)
- Créer un compte/se connecter
- Ajouter un moniteur
- Choisir "Ping Monitoring"
- Copier le lien de votre back
- Régler sur 5 minutes
- Cliquer sur "Create monitor"

Ainsi, au travers de cette documentation, je fourni à l'utilisateur toutes les clés pour héberger lui-même le back-end du projet, pour une utilisation personnelle.

Je peux donc passer à la documentation du front-end. Pour l'installation, je fourni une nouvelle fois le prérequis npm, ainsi que les commandes git pour cloner le repository principal et comment naviguer dans le dossier du front pour l'installer et le démarrer.

⚙ Installation

Prérequis

- npm

1. Cloner le repository

```
git clone --recurse-submodules https://github.com/AntooinePetit/habits
cd habits/front
```



2. Installer les dépendances

```
npm install
```



3. Lancer le projet

```
npm run dev
```



L'application sera disponible sur <http://localhost:5173/>

Pour le déploiement en ligne, j'explique comment réaliser un hébergement sur Vercel pour le front-end. Comme pour le back, je fournis les informations pour le fork du front.

DOSSIER PROFESSIONNEL (DP)

Déploiement

Faire un fork du projet

- Se rendre sur le [repository](#)
- Cliquer sur "Fork"
- Nommer le fork
- Cliquer sur "Create fork"

Je donne ensuite les informations pour héberger le front-end sur Vercel, incluant également la configuration des variables d'environnement pour la connexion au back-end.

Déploiement sur Vercel

- Se rendre sur [Vercel](#)
- Créer un compte/Se connecter avec GitHub
- Cliquer sur "Add New" > "Project"
- Sélectionner le repository correspondant
- Nommer le projet si besoin
- Cliquer sur "Deploy"
- Cliquer sur "Continue to Dashboard"
- Rendez-vous sur "Environment Variables", remplacer le lien local par le lien d'accès à votre back (lien du site déployé). **⚠️** Attention à ne PAS mettre de `/` à la fin de votre lien d'accès ! **⚠️**
- Pour accéder à l'application, cliquer sur "Visit"

Une fois chaque élément expliqué indépendamment, je rédige un 3ème fichier README.md dans le repository principal du projet. Dans ce README, je précise le prérequis global de npm, je fournis un lien renvoyant à chaque documentation de déploiement pour le front et le back, ainsi qu'une license d'utilisation MIT.

DOSSIER PROFESSIONNEL (DP)

⚙ Installation

Prérequis

- npm

1. Cloner le repository

```
git clone --recurse-submodules https://github.com/AntoinePetit/habits
```



2. Installation/Déploiement du back

- Suivez les instructions d'installation et/ou de déploiement, et configuration du [back](#)

3. Installation/Déploiement du front

- Suivez les instructions d'installation et/ou de déploiement, et configuration du [front](#)

Licence

Ce projet est sous licence MIT. Voir le fichier [LICENSE](#) pour plus de détails.

Ainsi, tous les éléments nécessaires à la documentation du développement du front-end et du back-end sont transmis à l'utilisateur, rendant l'application accessible à tous facilement.

2. Précisez les moyens utilisés :

Pour réaliser la documentation, j'ai d'abord repris un projet existant, j'ai entièrement déployé le front-end et le back-end, notant chaque étape pour être sûr d'expliquer correctement le déploiement. Je me suis ensuite renseigné sur comment réaliser un README.md lisible, cohérent et complet, et j'ai rédigé chaque README.md pour une utilisation la plus simple possible.

DOSSIER PROFESSIONNEL (DP)

3. Avec qui avez-vous travaillé ?

Pour ce projet, j'ai travaillé en totale autonomie.

4. Contexte

Nom de l'entreprise, organisme ou association - La plateforme

Chantier, atelier, service - Formation Développeur Web & Web Mobile
Période d'exercice - Du 01/10/2025 au 01/10/2025

5. Informations complémentaires (*facultatif*)

Cliquez ici pour taper du texte.

DOSSIER PROFESSIONNEL (DP)

Titres, diplômes, CQP, attestations de formation

(*facultatif*)

Intitulé	Autorité ou organisme	Date
Baccalauréat	Lycée René Cassin Arpajon	16/06/2017
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.

Cliquez ici pour taper du texte.

DOSSIER PROFESSIONNEL (DP)

Cliquez ici pour choisir une date ▾

Déclaration sur l'honneur

Je soussigné(e) Antoine Petit, déclare sur l'honneur que les renseignements fournis dans ce dossier sont exacts et que je suis l'auteur(e) des réalisations jointes.

Fait à Egly

le 06/02/2026

pour faire valoir ce que de droit.



Signature :

DOSSIER PROFESSIONNEL (DP)

Documents illustrant la pratique professionnelle

(facultatif)

Intitulé

DOSSIER PROFESSIONNEL (DP)

ANNEXES

(Si le RC le prévoit)

