

<i>Nom de naissance</i>	- PENA--BERMOND
<i>Nom d'usage</i>	- PENA--BERMOND
<i>Prénom</i>	- Thomas
<i>Adresse</i>	-

Titre professionnel visé

Développeur Web & Web Mobile

Modalité d'accès :

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

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 :

- 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.
- du **Dossier Professionnel** (DP) dans lequel le candidat a consigné les preuves de sa pratique professionnelle
- des résultats des évaluations passées en cours de formation lorsque le candidat évalué est issu d'un parcours de formation
- 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.



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

Sommaire

Exemples de pratique professionnelle

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

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

Développer la partie back-end d'une application web ou web mobile sécurisée	p.
▪ Mettre en place une base de donnée relationnelle	p.
▪ Développer des composants d'accès aux données SQL et NoSQL	p.
▪ Développer des composants métier côté serveur	p.
▪ Documenter le déploiement d'une application dynamique web ou web mobile	p.
Titres, diplômes, CQP, attestations de formation (<i>facultatif</i>)	p.
Déclaration sur l'honneur	p.
Documents illustrant la pratique professionnelle (<i>facultatif</i>)	p.
Annexes (<i>Si le RC le prévoit</i>)	p.

Exemples de pratique professionnelle

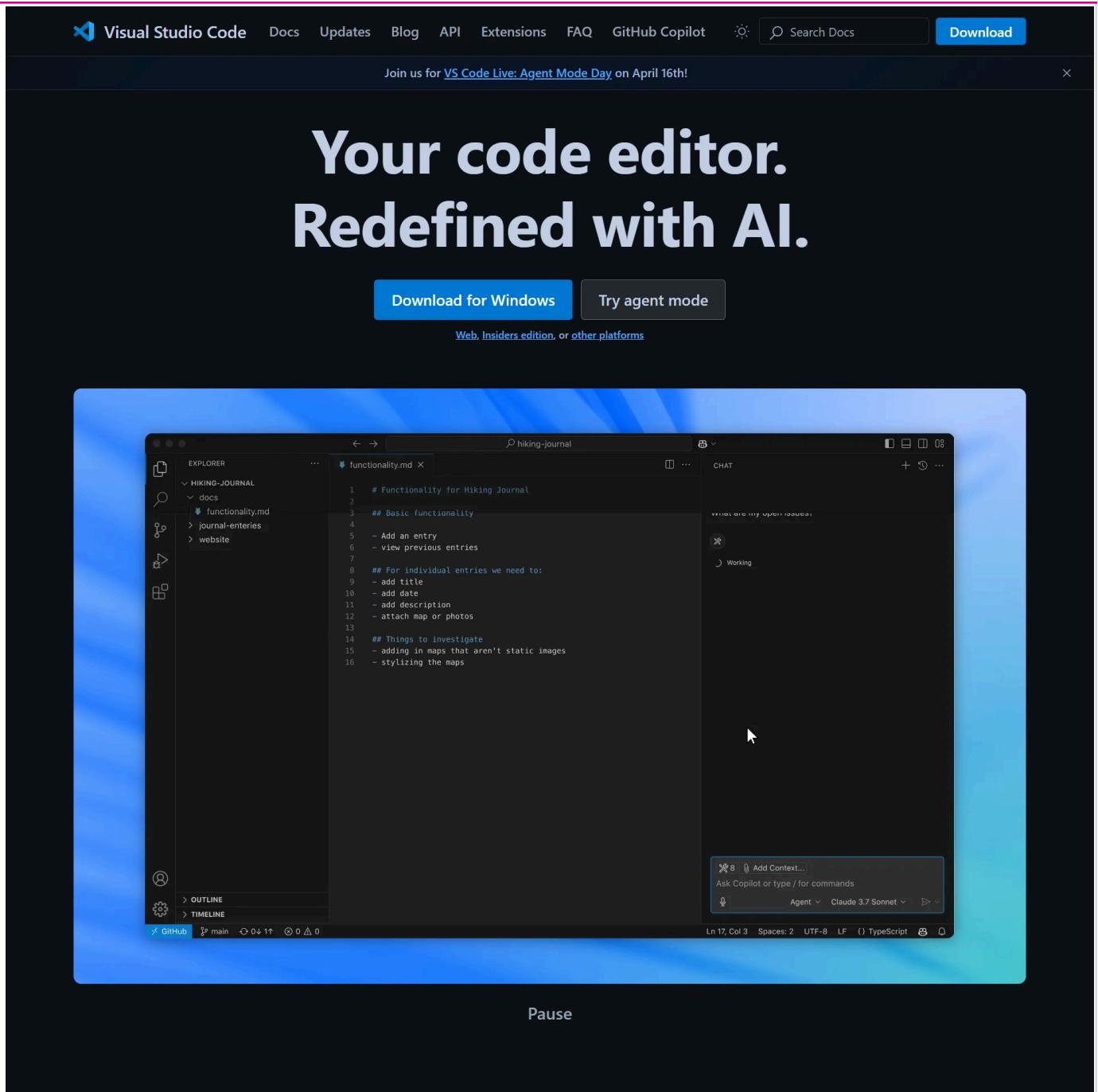
Activité-type

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 :

J'ai commencé par installer l'éditeur de code Visual Studio Code, que j'utilise comme environnement de développement principal..

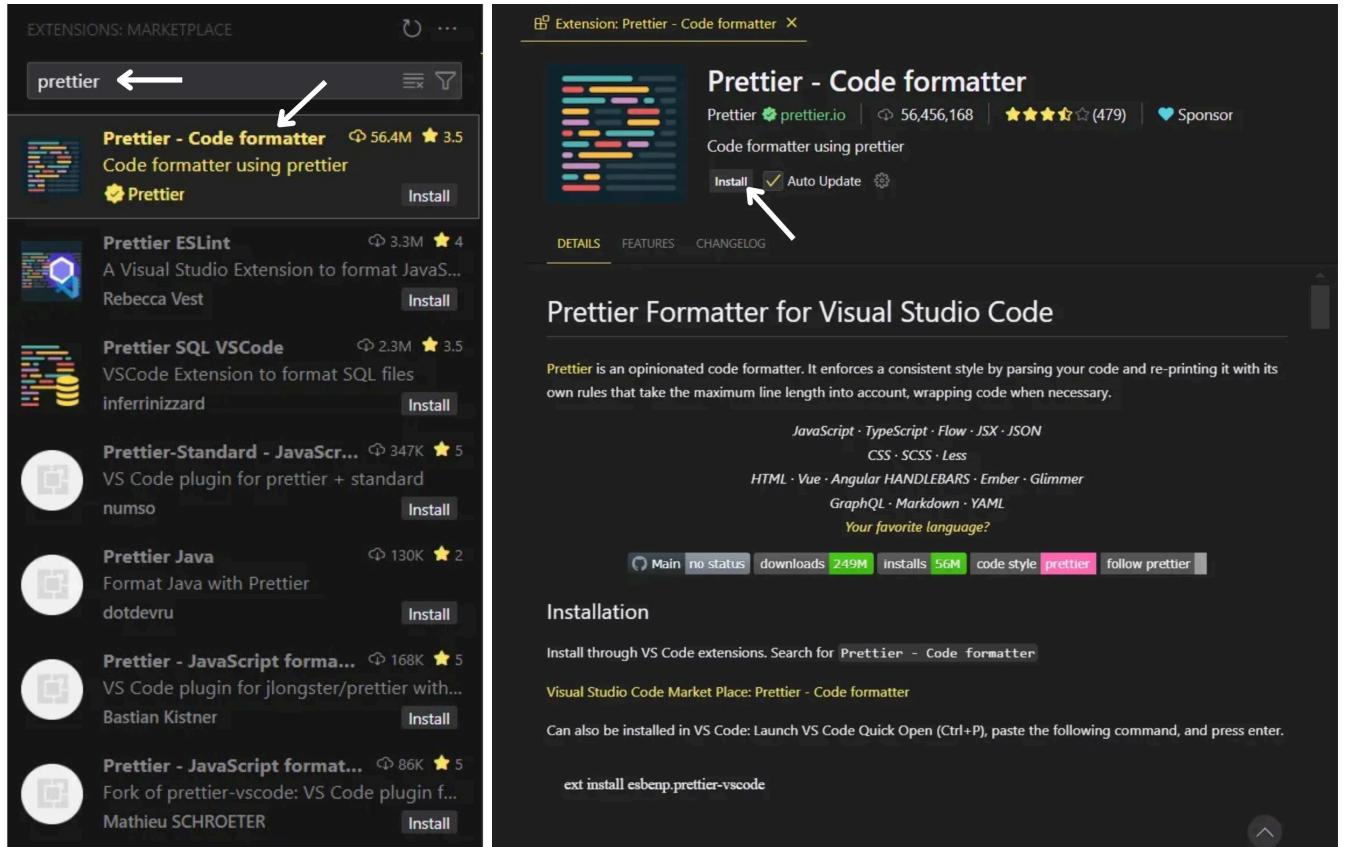


Cet outil me permet de travailler efficacement grâce à de nombreuses extensions telles que :

- GitLens améliore la gestion de Git en offrant des fonctionnalités avancées, comme l'historique détaillé des commits et la visualisation des auteurs des lignes de code. Cela facilite la collaboration et la compréhension de l'évolution du projet.
- ESLint analyse mon code JavaScript et détecte les erreurs ou les mauvaises pratiques. Il m'aide à écrire un code plus propre, plus fiable et conforme aux standards.
- PHP Intelephense fournit de l'autocomplétion et une intelligence de code pour PHP, facilitant l'écriture et la navigation dans le code avec des suggestions et des informations contextuelles.

- Prettier permet de formater automatiquement mon code selon des règles standard. Il améliore la lisibilité et garantit un style cohérent à chaque enregistrement de fichier.

Pour installer une extension, j'ouvre l'onglet dédié dans VSCode à l'aide du raccourci `Ctrl + Shift + X`. Dans la barre de recherche située en haut, je saisie le nom de l'extension souhaitée. Une fois trouvée, je la sélectionne, puis je clique sur "Installer".



Ensuite, j'ai installé Git depuis le site officiel. Une fois l'installation terminée, je vais pouvoir accéder au PowerShell de ma machine en appuyant sur Windows + R, en tapant powershell, puis en validant avec Entrée.



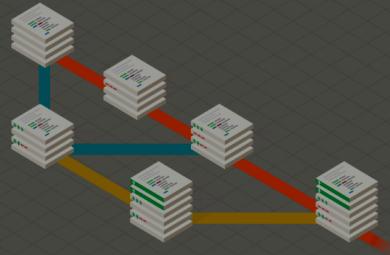
--local-branching-on-the-cheap

Type / to search entire site...



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.



Downloads

GUI clients and binary releases for all major platforms.



Community

Get involved! Bug reporting, mailing list, chat, development and more.



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



Latest source Release
2.49.0

[Release Notes \(2025-03-14\)](#)

[Download for Windows](#)

[Windows GUIs](#) [Tarballs](#)

[Mac Build](#) [Source Code](#)

Products providing Git hosting



radicle



Gitea



sourcehut



GitHub



Forgejo



Codeberg



GitLab



tangled.sh

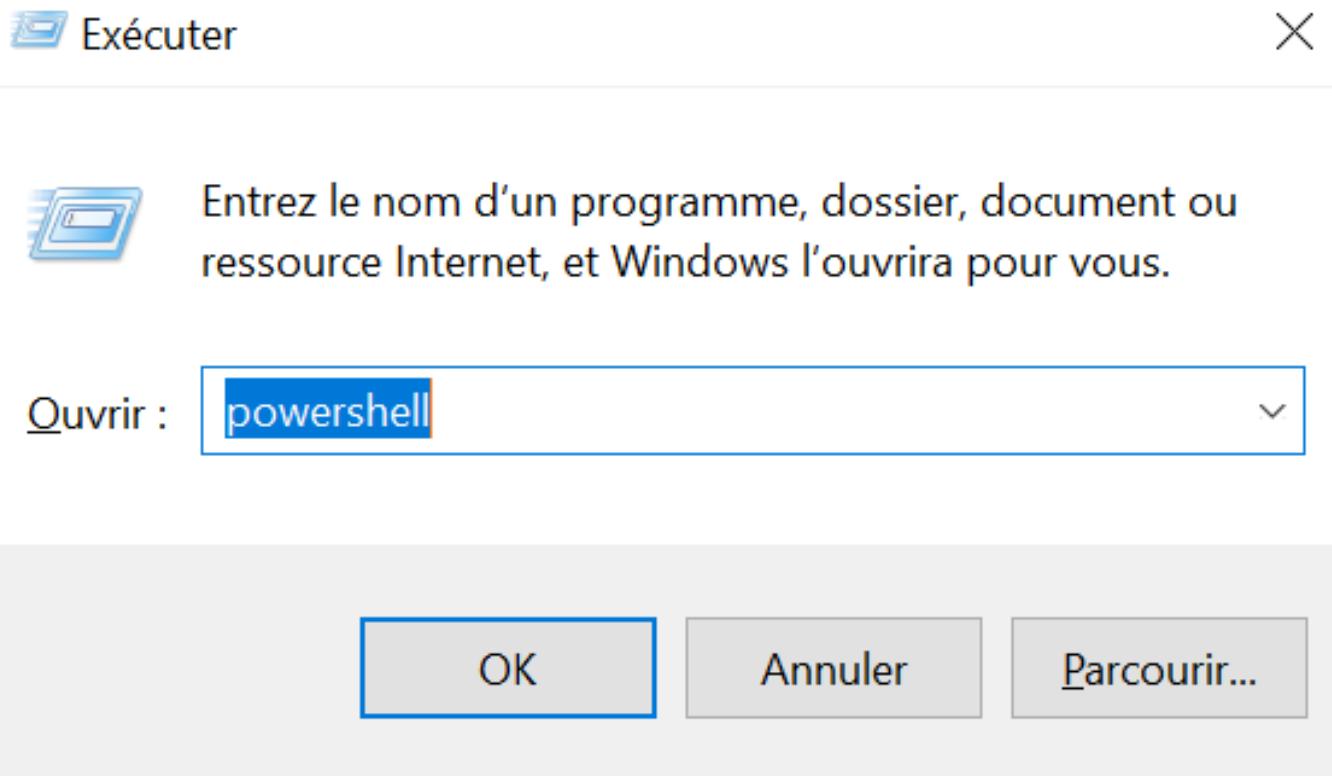


About this site

Patches, suggestions, and comments are welcome.

Git is a member of Software Freedom Conservancy





```
PS C:\Users\Utilisateur> git -v
git version 2.47.1.windows.2
```

Pour vérifier que **Git** est bien installé, je tape la commande `git -v`. Cela m'affiche la version de Git installée, par exemple `git version 2.47.1.windows.2`, ce qui confirme l'installation réussie.

Astuce : Pour plus de clarté, je peux utiliser la commande "`cls`" dans **PowerShell** pour effacer l'écran.

Ensuite, je configure mon identité Git en utilisant les commandes suivantes:

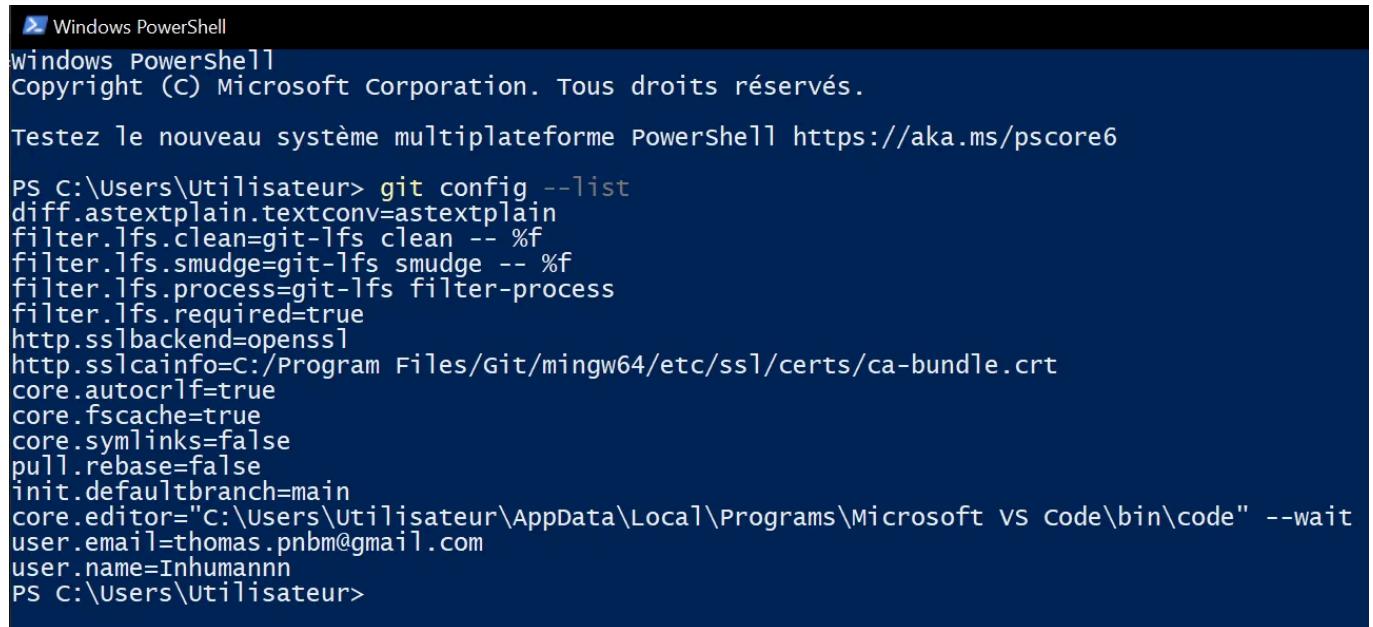
```
git config --global user.name "le Nom"
git config --global user.email "mon.email@example.com"
```

Je définis également Visual Studio Code comme éditeur par défaut pour Git avec cette commande :

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

Pour vérifier la configuration, je peux simplement exécuter la commande :

```
git config --list
```



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

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

PS C:\Users\utilisateur> git config --list
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="C:\Users\utilisateur\AppData\Local\Programs\Microsoft vs Code\bin\code" --wait
user.email=thomas.pnmb@gmail.com
user.name=Inhumann
PS C:\Users\Utilisateur>
```

Je crée un dossier de projet sur mon bureau, puis l'ouvre dans Visual Studio Code. Ensuite, je :

1. Créez un fichier `index.html` qui contiendra la structure de base de votre page.
2. Créez un dossier `css` qui servira à styliser vos pages HTML.
3. Dans le dossier `css` créer un fichier `style.css`
4. Créez un dossier `img` qui servira à mettre toutes vos images
5. Inclure un fichier fichier `.gitignore` pour exclure les fichiers inutiles du suivi Git.

Je me rends ensuite sur GitHub, où je crée un nouveau dépôt (repository) appelé "project". Je clique sur "New", puis rentre le nom du repository.



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner *



Inhumann

Repository name *

/

project

✓ project is available.

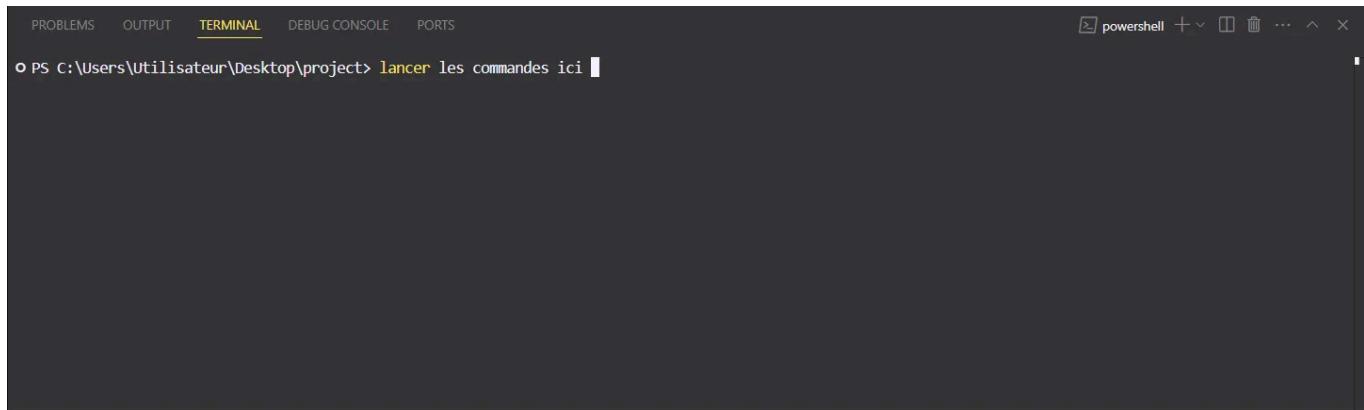
Cliquez sur « create repository »

Create repository

Une fois le dépôt créé, j'exécute les commandes suivantes dans le terminal de VSCode pour initialiser mon dépôt Git local et le connecter à GitHub :

```
git init  
git add *  
git commit -m "init: repo"  
git branch -M main  
git remote add origin https://github.com/votreNom/project.git  
git push -u origin main
```

Pour y accéder au terminal de VSCode je clique sur les logo en bas à gauche de votre VSCode et me rend dans l'onglet “TERMINAL”.

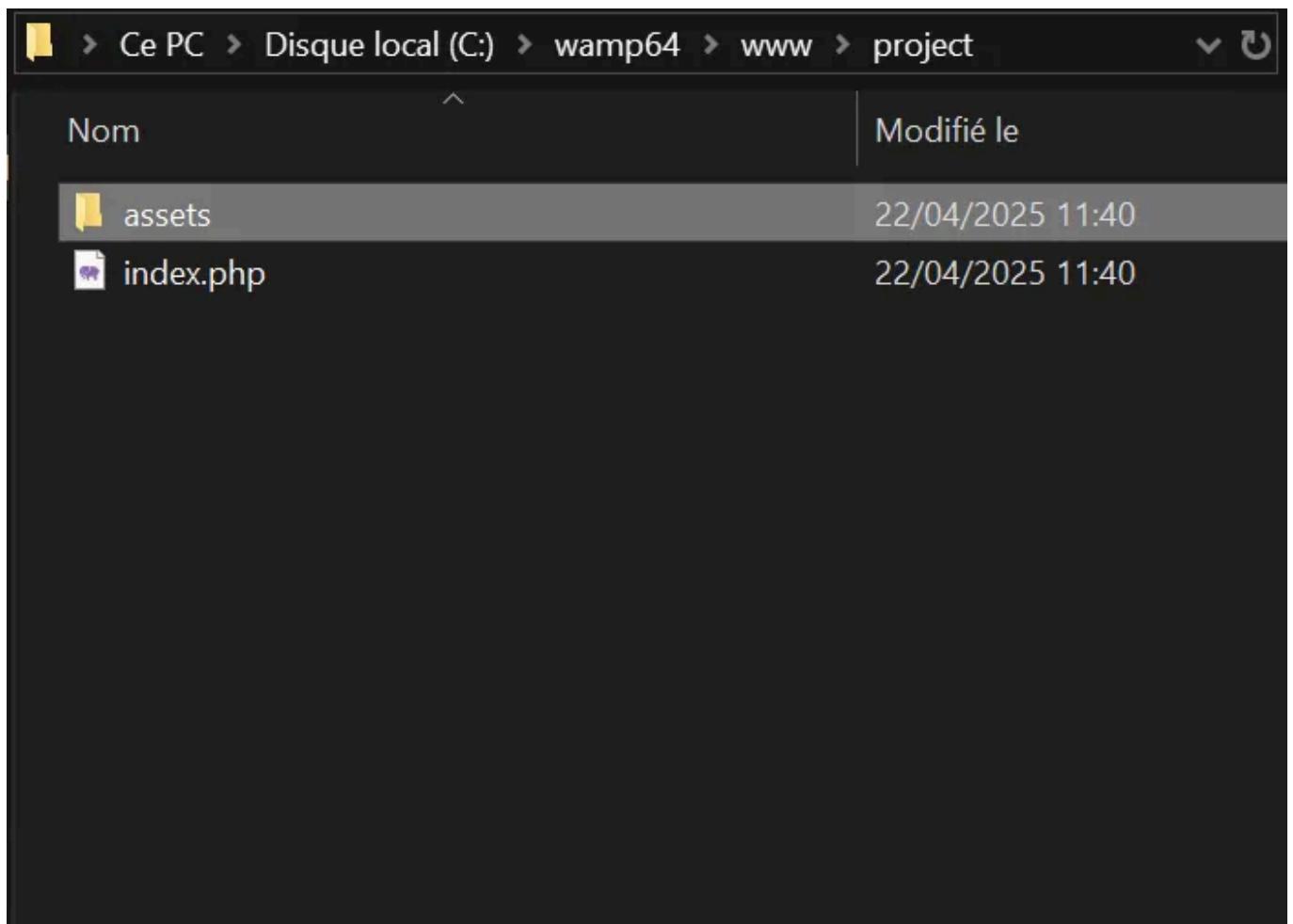


J'installe ensuite WampServer, un serveur local qui me permet de tester des fichiers PHP en local. Après avoir téléchargé et installé WampServer, je place mes fichiers dans le dossier `C:\wamp64\www`. Le chemin final pour mes fichiers sera donc `C:\wamp64\www\project`.



J'organise la structure du projet comme cela :

```
/project
└── index.php
└── /assets
```



Ensute, j'insére dans le fichier `index.php` via VSCode le code suivant :

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

Pour tester, je lance serveur avec WampServer, puis j'ouvre mon navigateur et allez à l'adresse suivante :
<http://localhost/project>.



Bonjour le monde !

Pour tester, je lance serveur avec WampServer, puis j'ouvre mon navigateur et allez à l'adresse suivante : <http://localhost/project>.

The screenshot shows the official Node.js website. On the left, there's a dark sidebar with the Node.js logo and navigation links like 'Apprendre', 'À propos', 'Téléchargement', 'Blog', 'Docs', 'Contribuer', and 'Certification'. The main content area has a dark background with a hexagonal pattern. It features a large heading 'Exécuter du JavaScript partout'. Below it, a paragraph explains what Node.js is: 'Node.js® est un environnement d'exécution JavaScript gratuit, open-source et multiplateforme qui permet aux développeurs de créer des serveurs, des applications web, des outils en ligne de commande et des scripts.' A button labeled 'Télécharger Node.js (LTS)' is visible. To the right, there's a code editor window titled 'JavaScript' containing a simple HTTP server example. 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's a 'Copier dans le Presse-papier' button. At the bottom of the page, there are links for 'Politique de Marque', 'Politique de confidentialité', 'Support de la version', 'Code de conduite', 'Politique de sécurité', and social media icons for GitHub, Slack, npm, Twitter, and LinkedIn.

Une fois installé, je vérifie que tout fonctionne avec les commandes suivantes dans PowerShell :

```
node -v
npm -v
```

```
Windows PowerShell

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

Ensuite, je crée un projet Node.js avec :

```
mkdir project (création du projet)
cd project (se déplace dans le dossier)
npm init -y (initialisation)
npm i express (installation de express)
```

Je crée le fichier server.js pour démarrer mon serveur local et tester le tout en lançant la commande suivante :

The screenshot shows a VS Code interface. On the left, the Explorer sidebar displays a project structure with 'PROJECT' expanded, showing 'backend' and 'frontend' folders. Inside 'backend', there are 'server.js', 'server.mjs', and 'index.html'. Inside 'frontend', there are 'css', 'img', 'gitignore', and 'index.html'. The main editor area shows the 'server.js' file with the following code:

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

At the bottom, the Terminal tab is active, showing the command 'cd backend' entered.

Cela permettra de lancer le serveur sur votre machine avec la commande suivante : `cd backend, node server`

```
PROBLEMS    OUTPUT    TERMINAL    DEBUG CONSOLE    PORTS

PS C:\Users\Utilisateur\Desktop\project> cd backend
○ PS C:\Users\Utilisateur\Desktop\project\backend> node server
  Listening on 127.0.0.1:3000
  
```

2. Précisez les moyens utilisés :

Pour installer et configurer mon environnement de travail de projet web ou web mobile, j'ai utilisé mes connaissances données lors des cours en formation.

3. Avec qui avez-vous travaillé ?

J'ai réalisé ce travail individuellement.

4. Contexte

Nom de l'entreprise, organisme ou association ► Centre de formation la plateforme à Brignole

Chantier, atelier, service Formation Développeur Web & Web Mobile

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

5. Informations complémentaires (facultatif)

Pour installer et configurer mon environnement de travail de projet web ou web mobile, j'ai utilisés :

- Le site de Visual Studio Code : <https://code.visualstudio.com/>
- Le site de git : <https://git-scm.com/>
- Le site github : <https://github.com/>
- Le site de wampserver : <https://www.wampserver.com/>
- Le site de node.js : <https://nodejs.org/en>

Activité-type

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 :

J'ai commencé par choisir le type d'application que je voulais réaliser. J'ai opté pour une application destinée à la gestion et au suivi de collections de plantes d'intérieur rare. Une fois ce choix fait, j'ai défini les fonctionnalités principales à travers des user stories.

Users Stories

User Story	Fonctionnalité	Ecran
US1	Ajouter une plante	Formulaire (nom, espèce, date d'acquisition et photo)
US2	Voir une fiche plante	Détails de la plante (historique d'arrosage, de fertilisation, notes de croissance)
US3	Corriger ou retirer des plantes	Modifier / Supprimer un plante
US4	Notifications d'entretiens	Notifications personnalisées
US5	Configurer des rappels	Configurer des rappels (fréquence et heure)
US6	Ajouter des photos pour l'évolution	ajouter une photo
US7	Filtrer par type / lumière / date	Menu filtre (type, lumière, date)
US8	Trier la collection	Menu tri (alphabétique, date d'acquisition ou dernier entretien)
US9	Historique global	Liste des derniers entretiens (arrosage, fertilisation, rempotage)

Cela m'a permis de me fixer des objectifs clairs et de mettre en place plusieurs idées pour l'application comme les fonctionnalités à implémenter, les écrans principaux à maquetter. Les informations essentielles à inclure, et quelques idées.

PlantApp : Destinée à la gestion et au suivi de collections de plantes d'intérieur rare

1. Fonctionnalités à implémenter

1. Gestion des fiches plantes
 - Création d'une nouvelle fiche (nom, espèce, date d'acquisition, photo)
 - Modification et suppression d'une fiche existante
2. Suivi de l'entretien
 - Enregistrement de chaque arrosage, fertilisation et rempotage
 - Historique détaillé pour chaque plante
 - Vue globale des actions d'entretien sur toutes les plantes
3. Rappels personnalisés
 - Notifications d'arrosage et de fertilisation
 - Configuration de la fréquence et de l'heure des rappels par plante
4. Documentation visuelle
 - Ajout régulier de photos d'évolution sur chaque fiche
 - Galerie chronologique
5. Organisation de la collection
 - Filtrage par type de plante, besoins en lumière, date d'acquisition...
 - Tri par ordre alphabétique, date d'acquisition, date de dernier entretien

3. Informations à inclure sur la fiche d'une plante

- Identité
 - Nom commun
 - Nom latin (espèce)
 - Date d'acquisition
 - Photo principale
- Santé & paramètres
 - Statut de santé (ex: bon / attention / critique)
 - Besoin en lumière (faible / modéré / fort)
- Entretien à venir
 - Prochain arrosage prévu (date & heure)
 - Prochaine fertilisation prévue
- Rappels
 - Fréquence d'arrosage (ex: tous les 5 jours)
 - Fréquence de fertilisation
- Historique d'entretien
 - Liste chronologique des arrosages (date, quantité)
 - Liste des fertilisations
 - Rempotages (date, type de substrat)
- Notes & observations
 - Textes libres sur croissance, problèmes, traitements...
- Galerie d'évolution
 - Photos datées, commentées

2. Écrans principaux à maquetter

1. Liste des plantes
 - Aperçu photo + nom + prochain entretien prévu
 - Boutons filtre / tri
2. Fiche détaillée d'une plante
 - Toutes les infos de la fiche + onglets Historique / Photos / Rappels
 - Boutons Éditer / Supprimer
3. Crédit / Édition de fiche
 - Formulaire (nom, espèce, date, photo)
 - Paramètres de rappel (fréquence & heure)
4. Historique global d'entretien
 - Timeline ou tableau de toutes les actions (toutes plantes confondues)
 - Filtres par type d'action ou par plante
5. Filtre & tri avancés
 - Sélection multi-critères
 - Aperçu dynamique des résultats
6. Galerie d'évolution photo
 - Miniatures datées, possibilité d'ajouter une photo
7. Réglages généraux (optionnel)
 - Notifications, fuseau horaire, thèmes...

Pour les vignettes « Recent Plants » sur la homepage :

- Espèce (nom latin)
Pour donner un peu plus de contexte botanique sans ouvrir la fiche complète.
- Date d'ajout
Par ex. "Ajoutée il y a 3 jours" (ou la date précise au survol) pour matérialiser la nouveauté.
- Prochain entretien prévu
Un petit pictogramme "🕒" avec la date ou "dans 2 jours" pour l'arrosage, voire "⚠️" si c'est déjà dû.
- État de santé rapide
Un badge (vert/jaune/rouge) ou une icône qui signale "Bon état", "Surveillez", "Besoin d'attention".
- Tag de besoin lumière
Par exemple un petit soleil ☀️/☀️/☀️ pour faible/modéré/plein soleil.
- Bouton d'accès rapide
Un mini-icône "▶" ou "Voir" qui mène directement à la fiche détaillée.

Ensuite, je suis passé à la partie design. Pour m'inspirer, j'ai consulté des applications similaires sur le Play Store, ce qui m'a aidé à définir le style visuel.



Jeux Applications

Livres Enfants



PlantNet Identification Plante

PlantNet

4,6★

256 k avis

10 M+

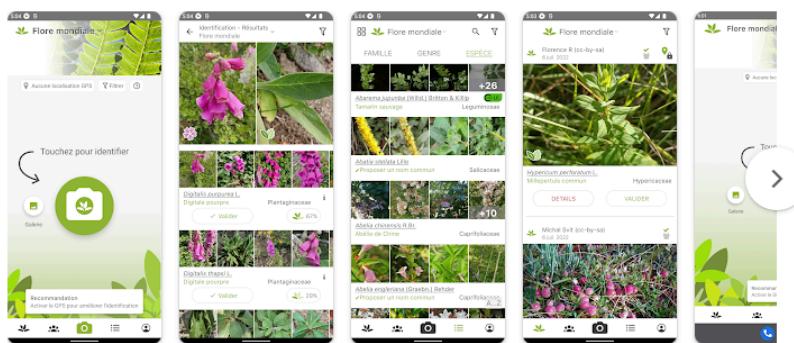
Téléchargements

3

PEGI 3

[Installer](#)[Partager](#)[Ajouter à la liste de souhaits](#)

Cette application est disponible pour tous vos appareils



Assistance de l'appli

Applications similaires →



Ebbn TreeEbb
Cybox
4,1★



Plantum:reconnaissance plante
AIBY Inc.
4,2★



Flora Incognita
Technische Universität Ilmenau
4,7★



Plant App - Identifiant Plante
ScaleUp
4,4★



Blossom - Reconnaissez plantes
Mosaic Srl
4,4★



iNaturalist
iNaturalist
4,0★

À propos de l'application →

Pl@ntNet est une application vous permettant d'identifier les plantes simplement en les photographiant avec votre smartphone. Très utile lorsqu'on n'a pas un botaniste sous la main ! Pl@ntNet, c'est aussi un grand projet de sciences participatives : toutes les plantes que vous photographiez sont collectées et analysées par des scientifiques du monde entier afin de mieux comprendre l'évolution de la biodiversité végétale et de mieux la préserver.

...

Date de mise à jour

14 août 2025

#4 meilleurs articles à 0 € – enseignement

Enseignement

Sécurité des données →

La sécurité, c'est d'abord comprendre comment les développeurs collectent et partagent vos données. Les pratiques concernant leur confidentialité et leur protection peuvent varier selon votre utilisation, votre région et votre âge. Le développeur a fourni ces informations et peut les modifier ultérieurement.

Cette appli peut partager ces types de données avec des tiers
Photos et vidéos

J'ai ainsi pu concevoir ma charte graphique de mon UI (User Interface), en choisissant les couleurs, les polices, les log, etc.

Charte graphique

Logo



Palette de couleurs



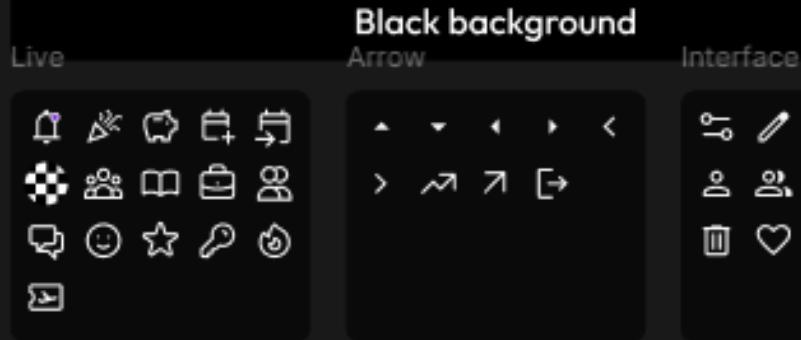
Polices

	Police	font-weight	font-size
Titre 1	Outfit	extra bold	36px
Titre 2	Outfit	semi bold	18px
Titre 3	Outfit	medium	16px
Texte	Outfit	regular	14px

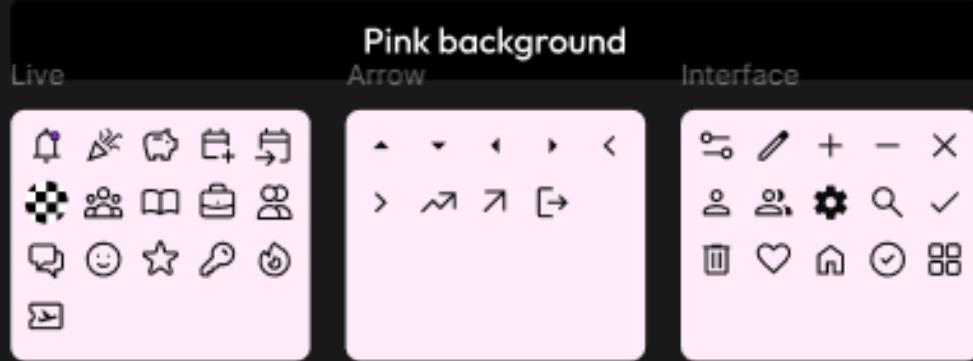
Frame 427318356

Icons

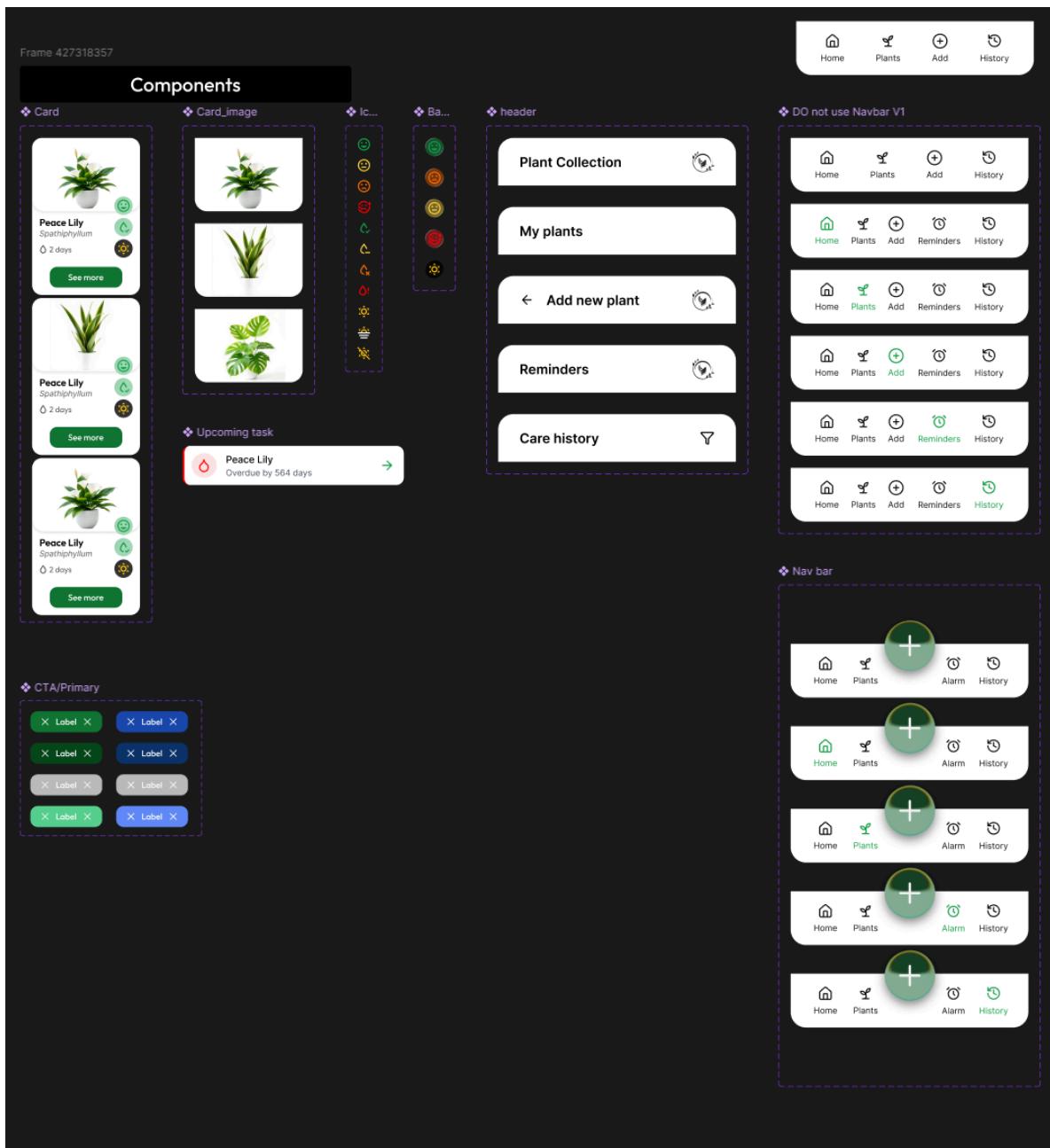
Frame 427318354



Frame 427318355



Après cette phase de réflexion, j'ai créé des composants réutilisables (comme le header, le footer, et d'autres éléments).



Une fois ces composants prêts, j'ai commencé à construire l'interface de l'application : La page d'accueil, la page des plantes, la pages détails, l'ajout de plantes, les rappels, et l'historique. Par manque de temps, je n'ai pas pu terminer complètement le projet, cependant j'ai pris soin de structurer de manière claire mon projet ce qui me permettra de le poursuivre de manière cohérente par la suite.

PlantApp



Welcome back

you have 3 plants in your collection

 **3**
Total Plants

 **3**
Tasks

Upcoming Care Tasks

 **Peace Lily**
Overdue by 2 days 

 **Monstera Deliciosa**
Overdue by 1 day 

 **Aloe Vera**
Overdue by 365 days 

Recent Plants [View All](#)


Peace Lily
Spathiphyllum
⌚ 2 days 


[See more](#)


Peace Lily
Spathiphyllum
⌚ 2 days 


[See more](#)


Peace Lily
Spathiphyllum
⌚ 2 days 


[See more](#)

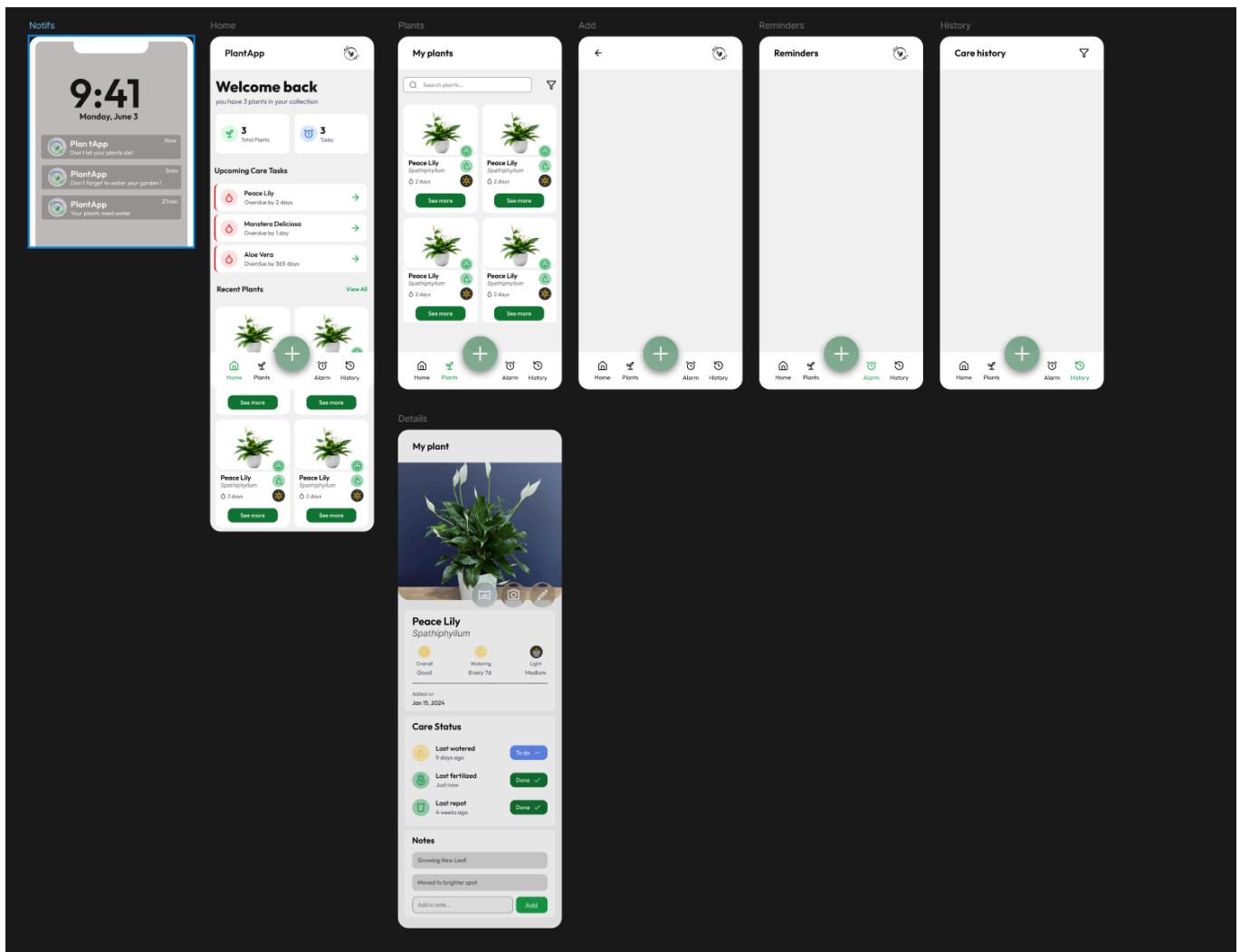

Peace Lily
Spathiphyllum
⌚ 2 days 


[See more](#)

Recent Plants [View All](#)

 Home  Plants   Alarm  History

 Home  Plants   Alarm  History



2. Précisez les moyens utilisés :

Pour maquetter des interfaces utilisateur web ou web mobile, j'ai utilisé mes connaissances données lors des cours en formation ainsi que des recherches internets (tutos youtube pour les composants réutilisables).

3. Avec qui avez-vous travaillé ?

J'ai réalisé ce travail individuellement.

4. Contexte

Nom de l'entreprise, organisme ou association	Centre de formation la plateforme à Brignole
--	--

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

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

5. Informations complémentaires (facultatif)

Pour maquetter des interfaces utilisateur web ou web mobile, j'ai utilisés :

- Le site de Figma : <https://www.figma.com/>
- Le site de Youtube : <https://www.youtube.com/>

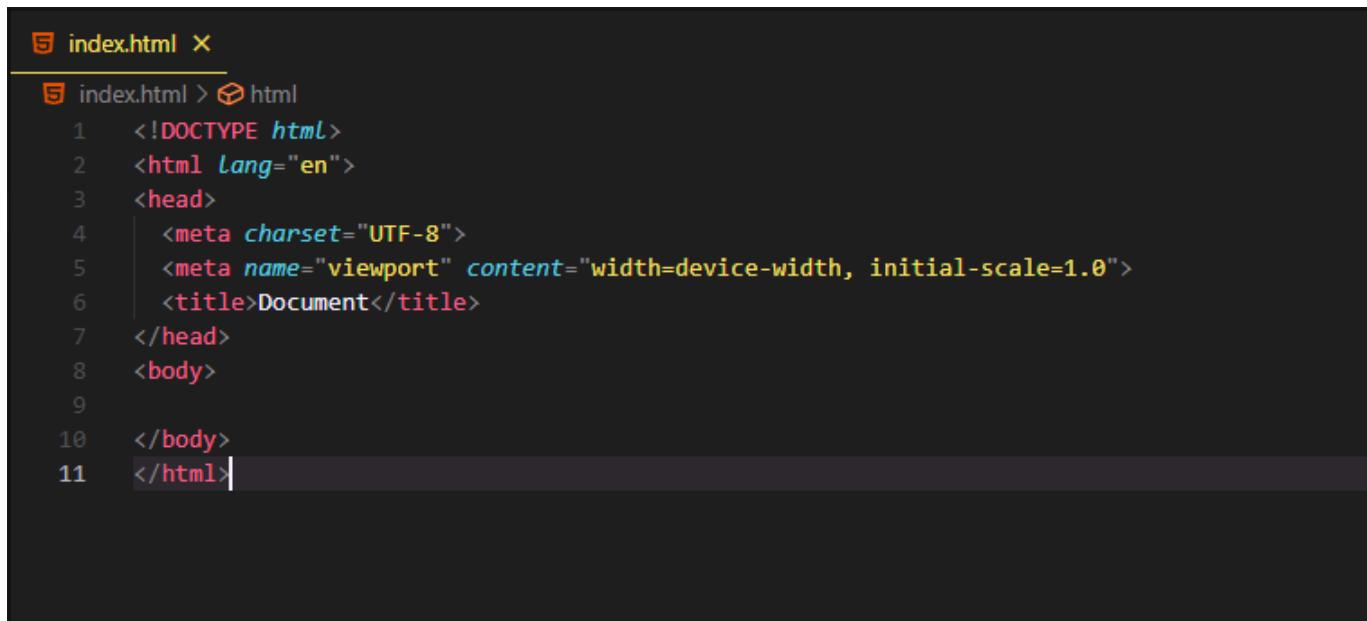
Développer la partie front-end d'une application web ou web mobile sécurisée

Activité-type

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 :

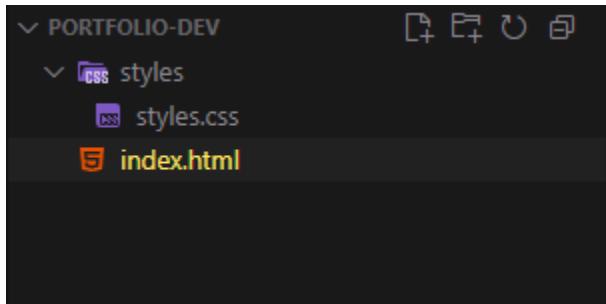
Pour commencer, j'ai créé la base de mon projet en réalisant un dossier nommé "portfolio.dev". A l'intérieur, j'ai ajouté un fichier index.html dans lequel j'ai saisi le raccourci "!" afin d'insérer automatiquement la structure de base du document (doctype, head, body, etc.).



The screenshot shows a code editor window with a dark theme. The file is named "index.html". The code is as follows:

```
index.html X
index.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |<meta charset="UTF-8">
5  |<meta name="viewport" content="width=device-width, initial-scale=1.0">
6  |<title>Document</title>
7  </head>
8  <body>
9
10 </body>
11 </html>
```

Ensuite, j'ai créé un dossier styles contenant mon fichier styles.css ce qui m'a permis de pouvoir initialiser mon dépôt sur Github en faisant mon premier commit.



Après cela, j'ai relié ma feuille de style à mon fichier html, ce qui m'a permis de commencer le développement structurel et visuel de mon portfolio. J'ai d'abord réalisé le header et le footer :

- Dans le header, j'ai ajouté le titre principal et le menu de navigation
- Dans le footer, j'ai opté pour quelque chose de simple comprenant un titre, un texte et un copyright.

```
index.html X
index.html > html > body > footer
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8" />
5      <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6      <link rel="stylesheet" href="styles/main.css" />
7      <title>Portfolio - Accueil</title>
8  </head>
9  <body>
10 <header>
11     <h1>Thomas.dev</h1>
12     <nav>
13         <ul>
14             <li><a href="index.html" class="active">Accueil</a></li>
15             <li><a href="about.html">À propos</a></li>
16             <li><a href="project.html">Projets</a></li>
17             <li><a href="contact.html">Contact</a></li>
18         </ul>
19         <i class="fa-solid fa-bars" id="bars"></i>
20     </nav>
21 </header>
22 > <main id="home"> ...
23 </main>
24 <footer>
25     <div>
26         <div>
27             <h2>Thomas.dev</h2>
28             <p>Développeur Web Front-End & Designer UI/UX</p>
29         </div>
30         <div>
31             <div class="social">
32                 <a href="#" target="_blank">
33                     <i class="fa-solid fa-basketball"></i>
34                 </a>
35                 <a href="#" target="_blank"><i class="fa-brands fa-github"></i></a>
36                 <a href="#" target="_blank">
37                     <i class="fa-brands fa-linkedin-in"></i>
38                 </a>
39             </div>
40             <p>© 2025 Thomas PENA. Tous droits réservés.</p>
41         </div>
42     </div>
43 </footer>
44 <script
45     src="https://kit.fontawesome.com/2bb90c57b5.js"
46     crossorigin="anonymous"
47 ></script>
48 </body>
49 </html>
50
```

```
styles.css  X
styles > styles.css > :root
1  @import url("https://fonts.googleapis.com/css2?family=Inter:ital,opsz,wght@0,14..32,100..900;1,14..32,100..900&family=Playfair+Display:ital,wght@0,400..900;1,400..900&display=swap");
2
3  *
4    margin: 0;
5    padding: 0;
6    box-sizing: border-box;
7    scroll-behavior: smooth;
8  }
9
10 :root {
11   --ft-title: "Playfair Display", serif;
12   --ft-main: "Inter", sans-serif;
13
14   --fz-logo: clamp(1.25rem, 4vw, 2rem);
15   --fz-img-presentation: clamp(4rem, 18vw, 15.5rem);
16   --fz-skill: clamp(2rem, 4vw, 2.25rem);
17
18   --fw-title: normal;
19   --fz-title: clamp(1.7rem, 5vw, 3.75rem);
20   --fz-title2: clamp(1.75rem, 4vw, 2.25rem);
21   --fz-title3: clamp(1.25rem, 3vw, 1.5rem);
22
23   --fz-main: clamp(0.775rem, 2vw, 1.125rem);
24   --fz-other: clamp(0.775rem, 1.5vw, 1rem);
25
26   --space-xs: 15px 15px;
27   --space-sm: 30px 15px;
28   --space-md: 50px 15px;
29   --space-lg: 75px 30px;
30
31   --p-course: clamp(0px, 5vw, 40px);
32
33   --clr-logo: □#2d3047;
34   --clr-main: □#000000;
35   --clr-second: □#ffffff;
36   --clr-third: □#d1d5db;
37   --clr-nav: □#333333;
38   --clr-link: □#419d78;
39   --clr-background-footer: □#2d3047;
40   --clr-background-section: □#f9fafb;
41   --clr-content: □#9ca3af;
42 }
43
44 h1,
45 h2,
46 h3 {
47   font-family: var(--ft-title);
48   font-weight: var(--fw-title);
49 }
50
51 h1 {
52   color: var(--clr-logo);
53   font-size: var(--fz-logo);
54 }
55
56 h2 {
57   font-size: var(--fz-title2);
58 }
59
60 h3 {
61   font-size: var(--fz-title3);
62 }
63
64 p,
65 a {
66   font-family: var(--ft-main);
67   font-size: var(--fz-main);
68 }
69
```

```
styles.css  X
styles > styles.css > :root
142
143 < header {
144   max-width: 1536px;
145   margin: auto;
146   display: flex;
147   justify-content: space-between;
148   align-items: center;
149   padding: var(--space-xs);
150
151 < ul {
152   list-style: none;
153   display: flex;
154   gap: 8vh;
155
156 < a {
157   color: var(--clr-logo);
158   transition: color 0.3s ease;
159 }
160
161 < a:hover {
162   color: var(--clr-link);
163   cursor: pointer;
164 }
165 }
166 }
167
168 < footer {
169   background-color: var(--clr-background-footer);
170   color: var(--clr-second);
171
172 < > div {
173   max-width: 1536px;
174   margin: auto;
175   display: flex;
176   justify-content: space-between;
177   align-items: center;
178   padding: var(--space-sm);
179
180 < div:nth-child(1) {
181   < h2 {
182     font-size: var(--fz-logo);
183     padding-bottom: 5px;
184   }
185
186 < < p {
187   font-size: var(--fz-other);
188   color: var(--clr-content);
189 }
190 < < a {
191   color: var(--clr-second);
192   font-size: var(--fz-other);
193 }
194 }
195 < < div:nth-child(2) {
196   < .social {
197     display: flex;
198     flex-direction: row-reverse;
199     padding-bottom: 12px;
200     gap: 4vh;
201     justify-content: center;
202   }
203
204 < < p {
205   font-size: var(--fz-other);
206   color: var(--clr-content);
207 }
208 }
209 }
210 }
```

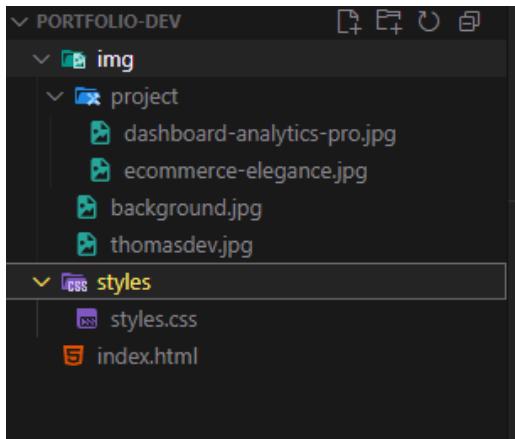
J'ai également intégré la bibliothèque Font Awesome via le script. Cela m'a permis d'ajouter des icônes à mon application.

```
<script
  src="https://kit.fontawesome.com/2bb90c57b5.js"
  crossorigin="anonymous"
></script>
```

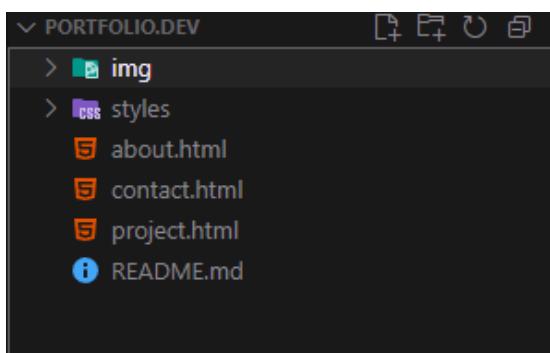
Une fois le header, et le footer mis en place, j'ai commencé à travailler sur ma partie principale. J'ai créé plusieurs sections : Présentation, A Propos, projet, contact.

```
<main id="home">
  <section id="presentation">
    <div>
      <div>
        <div id="res-tel">
          <div>
            <h2>Thomas Dev</h2>
            <p>Développeur Web Front-End & Designer UI/UX</p>
          </div>
          
        </div>
        <p>
          Je crée des expériences web élégantes, intuitives et performantes
          avec une attention particulière aux détails.
        </p>
        <a href="#" class="btn-primary">Voir mes projets</a>
      </div>
      <div>
        
      </div>
    </div>
  </section>
  <section id="about">
    <h2>À propos de moi</h2>
    <div>
      <div id="course">
        <h3>Mon parcours</h3>
        <p>
          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.
        </p>
        <p>
          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.
        </p>
        <p>
          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.
        </p>
        <a href="#" class="link">En savoir plus sur mon parcours </a>
      </div>
      <div>
        <h3>Mes compétences</h3>
        <div id="skills">
          <div>
            <i class="fa-brands fa-html5 skill"></i>
            <p>HTML5</p>
          </div>
          <div>
            <i class="fa-brands fa-css3 skill"></i>
          </div>
        </div>
      </div>
    </div>
  </section>
</main>
```

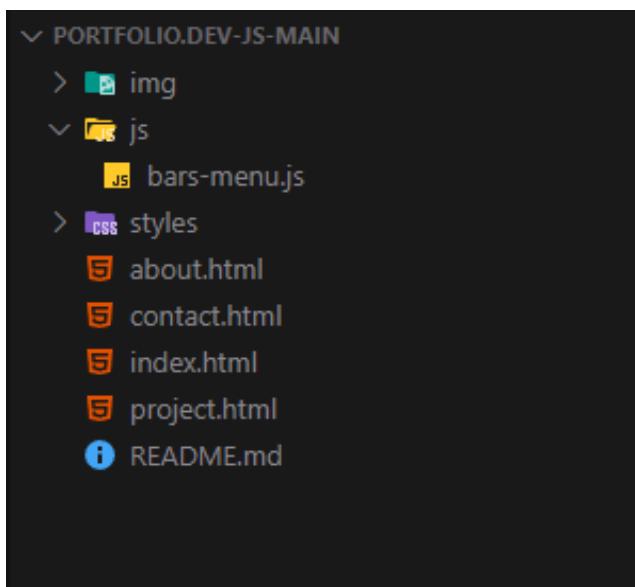
J'ai également créé un dossier img destiné à regrouper les images du site, telles que le fond d'écran et des images libres de droits.



Après avoir finalisé la page d'accueil et assurer son adaptation responsive, j'ai réalisé les autres pages, tout en conservant la même identité visuelle afin de garantir une continuité graphique sur l'ensemble du site.



J'ai ensuite développé le menu mobile, pour lequel j'ai créé un dossier js contenant le fichier bars-menu.js qui sera utile uniquement pour la responsive mobile.

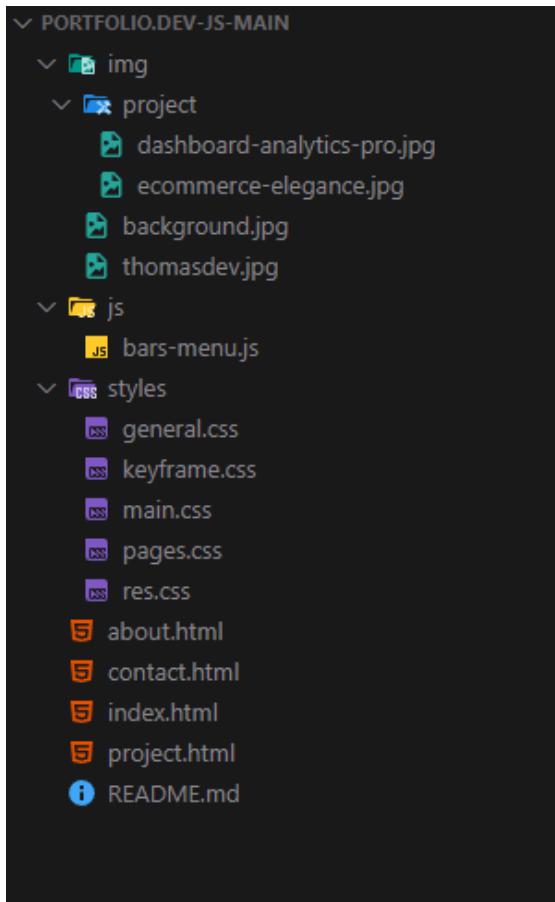


```

js bars-menu.js X
js > js bars-menu.js > ...
1  const barsMenu = document.querySelector("#bars");
2
3  barsMenu.addEventListener("click", () => {
4      const ulMenu = document.querySelector("#res-menu-ul");
5      ulMenu.style.display = ulMenu.style.display === "flex" ? "none" : "flex";
6      ulMenu.style.position = "absolute";
7      ulMenu.style.background = "white";
8      ulMenu.style.left = "clamp(15px, 2vw, 80px)";
9      ulMenu.style.top = "clamp(10px, 1vh, 40px)";
10     ulMenu.style.padding = "12px 0";
11     ulMenu.style.gap = "20px";
12
13     const navMenu = document.querySelector("#res-menu-nav");
14     navMenu.style.display = "flex";
15     navMenu.style.gap = "15px";
16  });
17

```

Une fois ce menu fonctionnel, j'ai ajouté plusieurs animations : un effet de clignotement sur le logo, ainsi qu'une animation d'apparition sur le header, ou sur la section principale. Enfin, pour une meilleure organisation de mes fichiers CSS, je me suis inspirée de la méthodologie Sass. J'ai ainsi créé un fichier principal main.css, dans lequel j'importe différentes fichiers : general, pages, responsives et keyframe. Cette structure me permet de garder un projet clair, évolutif, et bien organisé au fil du temps.



2. Précisez les moyens utilisés :

Pour installer et configurer mon environnement de travail de projet web ou web mobile, j'ai utilisé mes connaissances données lors des cours en formation.

3. Avec qui avez-vous travaillé ?

J'ai réalisé ce travail individuellement.

4. Contexte

Nom de l'entreprise, organisme ou association	Centre de formation la plateforme à Brignole
--	--

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

Période d'exercice	Du 22/05/2025	au 23/05/2025
---------------------------	----------------------	----------------------

5. Informations complémentaires (facultatif)

Suite à la réaliser des interfaces utilisateurs statiques web ou web mobile, vous trouverez :

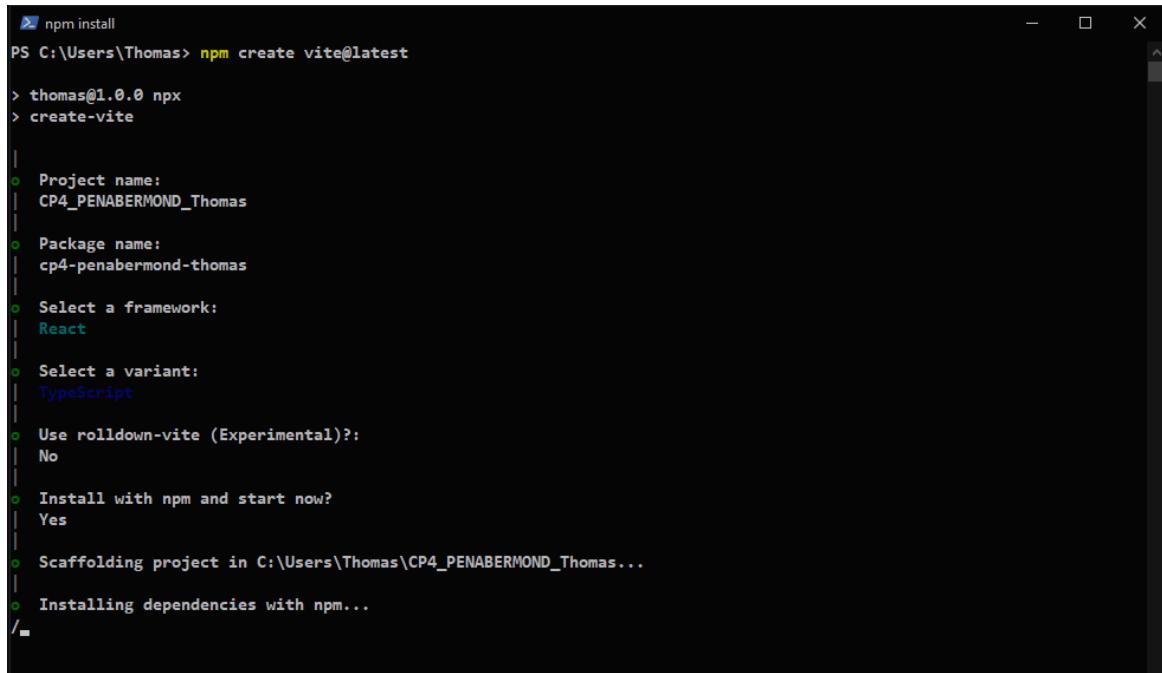
- Le lien du dépôt GitHub (24 commits): <https://github.com/Inhumannn/portfolio.dev.js>
- Le lien du site déployer (Vercel) : <https://portfolio-dev-two-puce.vercel.app/>

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 débuter, j'ai ouvert mon powershell pour créer mon projet React à l'aide de la commande `npm create vite@latest`. J'ai ensuite choisi le nom du projet `CP4_PENABERMOND_Thomas`, ainsi que le nom du package correspondant. J'ai opté pour le framework React avec TypeScript, puis j'ai répondu `non` à l'installation de Rollup-Vite et confirmé l'installation via `npm`, qui est le gestionnaire de paquets que j'utilise habituellement.



```
PS C:\Users\Thomas> npm create vite@latest
> thomas@1.0.0 npx
> create-vite

Project name: CP4_PENABERMOND_Thomas
Package name: cp4-penabermont-thomas
Select a framework: React
Select a variant: TypeScript
Use rollup-vite (Experimental)?:
Install with npm and start now?
Scaffolding project in C:\Users\Thomas\CP4_PENABERMOND_Thomas...
Installing dependencies with npm...
```

Une fois le projet créé, je me suis déplacé dans le dossier à l'aide de la commande

```
cd CP4_PENABERMOND_Thomas
```

J'ai ensuite installé les dépendances avec

```
npm i
```

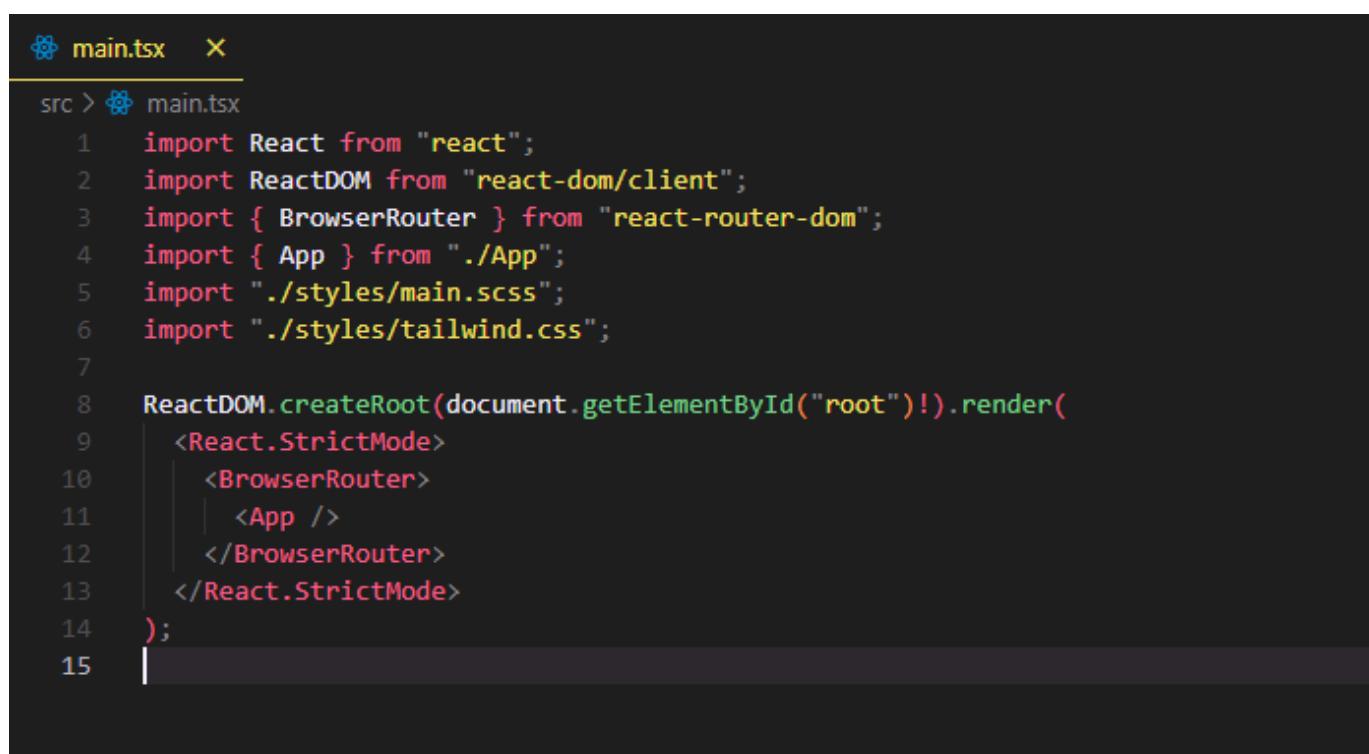
puis j'ai utilisé la commande

```
code .
```

qui me permet d'ouvrir vscode à partir du dossier dans lequel je suis situé. Une fois vscode lancé j'ai lancé le projet grâce à

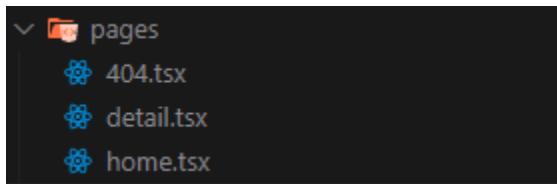
```
npm run dev
```

Après l'initialisation, j'ai procédé à la structuration de mon projet en supprimant les dossiers et fichiers inutiles (comme assets ou certains fichiers de style par défaut, etc.). J'ai ensuite modifié le fichier main.tsx afin d'y importer React Dom ainsi que les fichiers de styles nécessaires au bon fonctionnement de mon application.

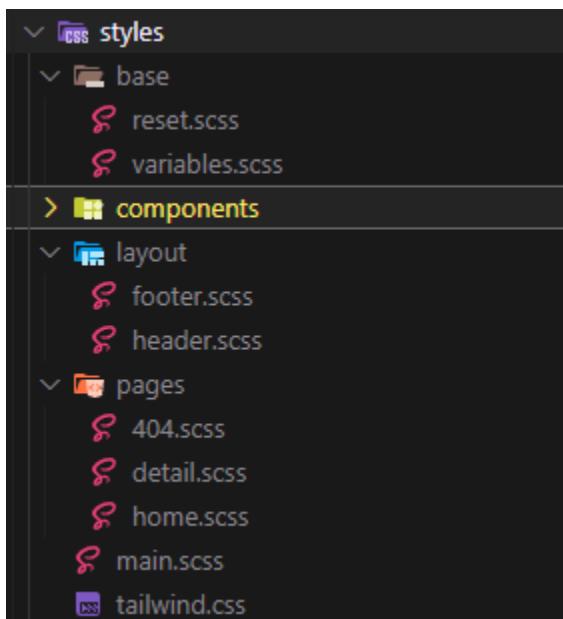


```
main.tsx
src > main.tsx
1 import React from "react";
2 import ReactDOM from "react-dom/client";
3 import { BrowserRouter } from "react-router-dom";
4 import { App } from "./App";
5 import "./styles/main.scss";
6 import "./styles/tailwind.css";
7
8 ReactDOM.createRoot(document.getElementById("root")!).render(
9   <React.StrictMode>
10     <BrowserRouter>
11       <App />
12     </BrowserRouter>
13   </React.StrictMode>
14 );
15 |
```

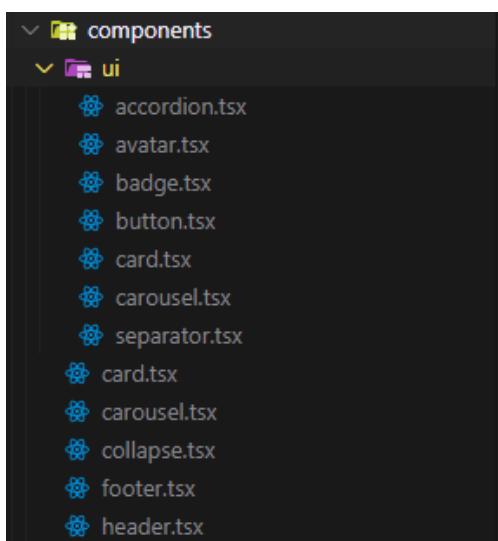
Dans app.tsx, j'ai mis en place le système de routage pour assurer la liaison entre les différentes pages. J'ai ensuite créé un dossier "pages",



contenant les pages principales de l'application, ainsi qu'un dossier styles destiné à regrouper mes fichiers de style. Pour la mise en forme, j'ai choisi d'utiliser Sass, en adoptant une structure claire et hiérarchisée.



J'ai également ajouté un fichier tailwind.css, car j'ai prévu d'intégrer la bibliothèque shadcn/ui pour bénéficier de composants préconstruits. J'ai donc créé un dossier components, destiné à accueillir tous les composants réutilisables, ainsi qu'un sous dossier "ui" pour ceux issus de shadcn/ui.



Une fois l'installation de shadcn/ui, je peux à présent attaquer le développement de mon header

```
header.tsx ×
src > components > header.tsx > Header
1 ✓ import { Button } from "@/components/ui/button";
2 import { Menu } from "lucide-react";
3 import { Link } from "react-router-dom";
4
5 ✓ export function Header() {
6   return (
7     <header>
8       <a href="/">
9         <h1 className="logo">Kasa</h1>
10        </a>
11       <nav>
12         <ul>
13           <li>
14             <Button variant="link" style={{ color: "#4b5563" }}>
15               <Link to="#">Accueil</Link>
16             </Button>
17           </li>
18           <li>
19             <Button variant="link" style={{ color: "#4b5563" }}>
20               <Link to="#">À propos</Link>
21             </Button>
22           </li>
23           <li>
24             <Button variant="link" style={{ color: "#4b5563" }}>
25               <Link to="#">Contact</Link>
26             </Button>
27           </li>
28         </ul>
29         <Menu className="menu" />
30       </nav>
31     </header>
32   );
33 }
34
```

et du footer

```
footer.tsx
src > components > footer.tsx > Footer
1  import { Separator } from "@/components/ui/separator";
2  import { Link } from "react-router-dom";
3  import { Button } from "./ui/button";
4
5  export function Footer() {
6    return (
7      <footer>
8        <div>
9          <div>
10            <h2>À propos de Kasa</h2>
11            <ul>
12              <li>
13                <Button
14                  variant="link"
15                  style={{ color: "#D1D5DB", padding: "12px 12px 12px 0px" }}>
16                  <Link to="#">Notre histoire</Link>
17                </Button>
18              </li>
19              <li>
20                <Button
21                  variant="link"
22                  style={{ color: "#D1D5DB", padding: "12px 12px 12px 0px" }}>
23                  <Link to="#">Equipe</Link>
24                </Button>
25              </li>
26              <li>
27                <Button
28                  variant="link"
29                  style={{ color: "#D1D5DB", padding: "12px 12px 12px 0px" }}>
30                  <Link to="#">Carrière</Link>
31                </Button>
32              </li>
33              <li>
34                <Button
35                  variant="link"
36                  style={{ color: "#D1D5DB", padding: "12px 12px 12px 0px" }}>
37                  <Link to="#">Presse</Link>
38                </Button>
39              </li>
40              <li>
41                <Button
42                  variant="link"
43                  style={{ color: "#D1D5DB", padding: "12px 12px 12px 0px" }}>
44                  <Link to="#">Support</Link>
45                </Button>
46              </li>
47              <div>
48                <h2>Support</h2>
49                <ul>
50                  <li>
51                    <Button
52                      variant="link"
53                      style={{ color: "#D1D5DB", padding: "12px 12px 12px 0px" }}>
54                      <Link to="#">Centre d'aide</Link>
55                    </Button>
56                  </li>
57                  <li>
58                    <Button
59                      variant="link"
60                      style={{ color: "#D1D5DB", padding: "12px 12px 12px 0px" }}>
61                      <Link to="#">Nous contacter</Link>
62                    </Button>
63                  </li>
64                  <li>
65                    <Button
66                      variant="link"
67                      style={{ color: "#D1D5DB", padding: "12px 12px 12px 0px" }}>
68                      <Link to="#">Signaler un problème</Link>
69                    </Button>
70                  </li>
71                </ul>
72              </div>
73            </ul>
74          </div>
75        </div>
76      </footer>
77    )
78  )
```

è

, que j'ai ensuite importés dans la page d'accueil.

```
home.tsx  X
src > pages > home.tsx > ...
1 import { Card } from "@/components/card";
2 import { Footer } from "@/components/footer";
3 import { Header } from "@/components/header";
4
5 export function Home() {
6   return (
7     <>
8       <Header />
9       <main id="home">
10         <section id="tagline">
11           <div>
12             <h2>Trouvez votre logement idéal</h2>
13             <p>
14               Découvrez des hébergements uniques et vivez comme un local <br />{" "}
15               partout dans le monde
16             </p>
17           </div>
18         </section>
19         <Card />
20       </main>
21       <Footer />
22     </>
23   );
24 }
25 |
```

J'ai également créé un composant Card, destiné à afficher les logements sur la page principale.

```
card.tsx  x
src > components > card.tsx > Card
1  import { Badge } from "@/components/ui/badge";
2  import { Star } from "lucide-react";
3  import { useEffect, useState } from "react";
4  import { Link } from "react-router-dom";
5
6  type Logement = {
7    id: string;
8    title: string;
9    location: string;
10   pictures: string;
11   tags: string[];
12   host: {
13     name: string;
14     picture: string;
15   };
16   rating: string;
17   description: string;
18   equipments: string[];
19 };
20
21 export function Card() {
22   const [logements, setLogements] = useState<Logement[]>([]);
23   useEffect(() => {
24     fetch("/data/logements.json")
25       .then((response) => response.json())
26       .then((data) => setLogements(data))
27       .catch((error) => console.error("Error fetching data:", error));
28   }, []);
29   return (
30     <section id="card-container">
31       <div>
32         {logements.map((logement) => (
33           <Link to={`/detail/${logement.id}`} key={logement.id}>
34             <article className="card">
35               <div>
36                 <img src={logement.pictures} alt={logement.title} />
37               </div>
38               <div>
39                 <h3>{logement.title}</h3>
40                 <p>{logement.location}</p>
41                 <div>
42                   {logement.tags.map((tag, i) => (
43                     <Badge className="tags" key={i}>
44                       {tag}
45                     </Badge>
46                   )));
47                 </div>
48                 <div>
49                   {[...Array(5)].map(_, j) => (
50                     // Tableau itéral -> [...Array(5)]
51                     <span key={j}>
52                       <Star
53                         size={17}
54                         fill={j < Number(logement.rating) ? "#facc15" : "none"}
55                         stroke="#facc15"
56                         className="transition-all"
57                         />
58                     </span>
59                   )));
60                   <span>{logement.rating}</span>
61                 </div>
62               </div>
63             </article>
64           </Link>
65         )));
66       </div>
67     </section>
68   );
69 }
70
```

Les données proviennent d'un fichier json situé dans le dossier “/data/logement.json”.

```
(ɔ) logements.json ×
```

```
public > data > (ɔ) logements.json > ...
```

```
1  [ 
2   { 
3     "id": "c1",
4     "title": "Appartement cosy à Marseille",
5     "location": "Marseille, France",
6     "pictures": "./img/img.jpg",
7     "tags": ["Vue mer", "Climatisée"],
8     "host": {
9       "name": "Claire Dupont",
10      "picture": "https://randomuser.me/api/portraits/women/10.jpg"
11    },
12    "rating": "4",
13    "description": "Un charmant appartement situé proche du Vieux-Port.",
14    "equipments": ["Wifi", "Cuisine", "Television", "Climatisation"]
15  },
16  { 
17    "id": "c2",
18    "title": "Studio moderne à Lyon",
19    "location": "Lyon, France",
20    "pictures": "./img/img.jpg",
21    "tags": ["Centre-ville", "Animaux autorisés"],
22    "host": {
23      "name": "Julien Martin",
24      "picture": "https://randomuser.me/api/portraits/men/11.jpg"
25    },
26    "rating": "3",
27    "description": "Studio lumineux et bien équipée pour un court séjour.",
28    "equipments": ["Wifi", "Chaussage", "Cuisine équipée"]
29  },
30  { 
31    "id": "c3",
32    "title": "Loft design à Bordeaux",
33    "location": "Bordeaux, France",
34    "pictures": "./img/img.jpg",
35    "tags": ["Design", "Spacieux"],
36    "host": {
37      "name": "Sophie Moreau",
38      "picture": "https://randomuser.me/api/portraits/women/12.jpg"
39    },
40    "rating": "5",
41    "description": "Loft spacieux avec une déco moderne et purifiante.",
42    "equipments": ["Machine à laver", "Wifi", "Television"]
43  },
44  { 
45    "id": "c4",
46    "title": "Maison de charme en Bretagne",
47    "location": "Quimper, France",
48    "pictures": "./img/img.jpg",
49    "tags": ["Campagne", "Famille"],
50    "host": {
51      "name": "Luc Bernard",
52      "picture": "https://randomuser.me/api/portraits/men/13.jpg"
53    },
54    "rating": "5",
55    "description": "Maison idéale pour des vacances en famille en pleine nature.",
56    "equipments": ["Jardin", "Cuisine", "Barbecue", "Cheminée"]
57  },
58  { 
59    "id": "c5",
60    "title": "Appartement avec terrasse à Nice",
61    "location": "Nice, France",
62    "pictures": "./img/img.jpg",
63    "tags": ["Terrasse", "Vue mer"]
```

Étant donné que le projet est réalisé en TypeScript, j'ai mis en place un typage strict des données, afin de préciser la nature de chaque élément (chaine de caractères, nombre, tableau, etc.).

```
type Logement = {
  id: string;
  title: string;
  location: string;
  pictures: string;
  tags: string[];
  host: {
    name: string;
    picture: string;
  };
  rating: string;
  description: string;
  equipments: string[];
};
```

J'ai utilisé un "useState" pour gérer la liste des logements,

```
const [logements, setLogements] = useState<Logement[]>([]);
```

et un "useEffect" pour effectuer un fetch vers le fichier json.

```
useEffect(() => {
  fetch("/data/logements.json")
    .then((response) => response.json())
    .then((data) => setLogements(data))
    .catch((error) => console.error("Error fetching data:", error));
}, []);
```

L'affichage des logements est ensuite généré dynamiquement à l'aide d'un .map().

```

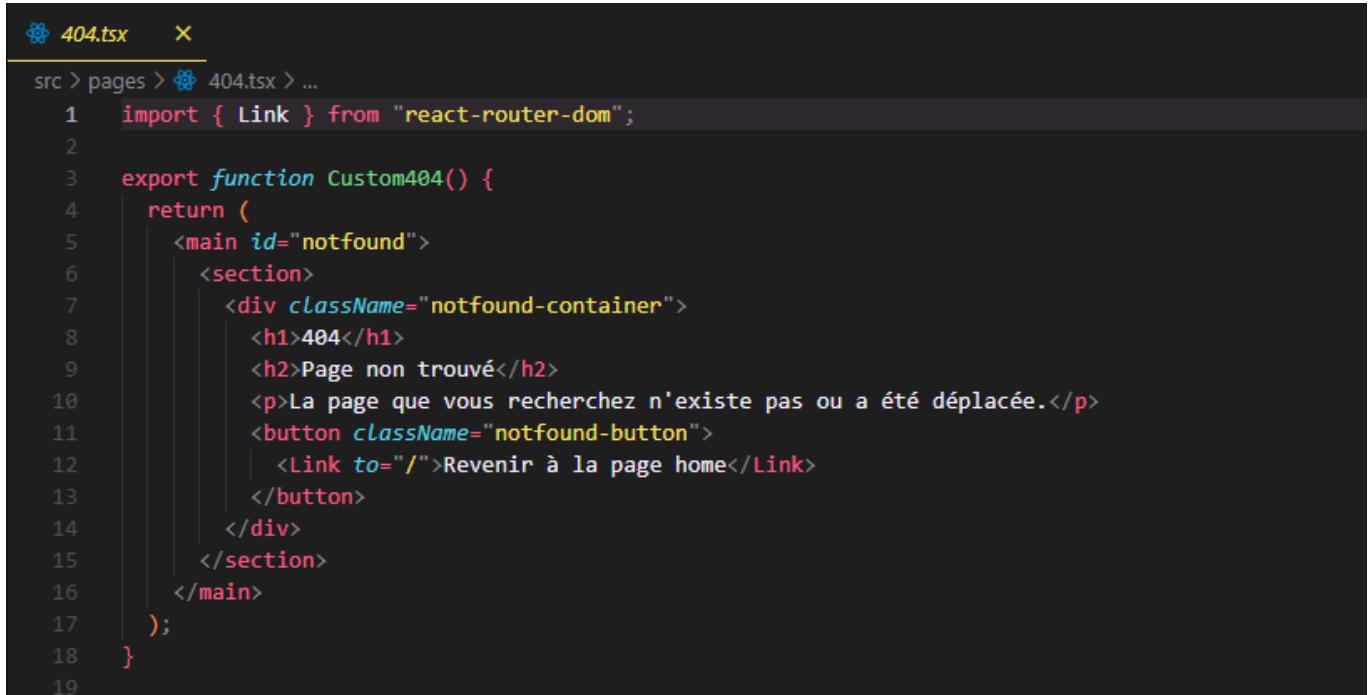
return (
  <section id="card-container">
    <div>
      {logements.map((Logement) => (
        <Link to={`/detail/${logement.id}`} key={logement.id}>
          <article className="card">
            <div>
              <img src={logement.pictures} alt={logement.title} />
            </div>
            <div>
              <h3>{logement.title}</h3>
              <p>{logement.location}</p>
              <div>
                {logement.tags.map((tag, i) => (
                  <Badge className="tags" key={i}>
                    {tag}
                  </Badge>
                )))
              </div>
              <div>
                {[...Array(5)].map(_, j) => (
                  // Tableau itératif -> [...Array(5)]
                  <span key={j}>
                    <Star
                      size={17}
                      fill={j < Number(logement.rating) ? "#facc15" : "none"}
                      stroke="#facc15"
                      className="transition-all"
                    />
                  </span>
                ))}
                <span>{logement.rating}</span>
              </div>
            </div>
          </article>
        </Link>
      )));
    </div>
  </section>
);

```

Pour chaque logement, la note est représentée par un système d'étoiles, affiché selon la valeur du rating présent dans le json.

```
<div>
  {[...Array(5)].map((_, j) => (
    // Tableau itéral -> [...Array(5)]
    <span key={j}>
      <Star
        size={17}
        fill={j < Number(logement.rating) ? "#facc15" : "none"}
        stroke="#facc15"
        className="transition-all"
      />
    </span>
  ))}
  <span>({logement.rating})</span>
</div>
***
```

Ensuite j'ai créé une page 404 (Not Found) simple, permettant de gérer les erreurs de navigation.



The screenshot shows a code editor window with a dark theme. The file is named '404.tsx' and is located in the 'src/pages' directory. The code defines a custom 404 component:

```
src > pages > 404.tsx > ...
1 import { Link } from "react-router-dom";
2
3 export function Custom404() {
4   return (
5     <main id="notfound">
6       <section>
7         <div className="notfound-container">
8           <h1>404</h1>
9           <h2>Page non trouvé</h2>
10          <p>La page que vous recherchez n'existe pas ou a été déplacée.</p>
11          <button className="notfound-button">
12            <Link to="/">Revenir à la page home</Link>
13          </button>
14        </div>
15      </section>
16    </main>
17  );
18}
19
```

Lorsque l'utilisateur clique sur une carte, il est redirigé vers la page de détails du logement correspondant. Cette page reprend la même logique que la page d'accueil, mais utilise l'identifiant (ID) pour récupérer et afficher les informations spécifiques au logement sélectionné.

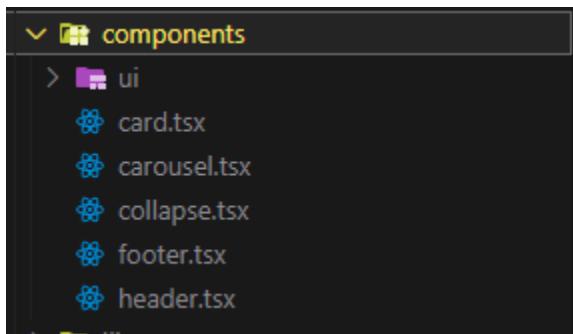
```
(detail.tsx) x
src > pages > detail.tsx > ...
11  type Logement = {
15    pictures: string[];
16    tags: string[];
17  < host: {
18    name: string;
19    picture: string;
20  };
21  rating: string;
22  description: string;
23  equipments: string[];
24};
25
26 < export function Detail() {
27   const { id } = useParams<{ id: string }>();
28   const [logement, setLogement] = useState<Logement | null>(null);
29
30 < useEffect(() => {
31   fetch("/data/logements.json")
32     .then((response) => response.json())
33   < then((data: Logement[]) => {
34     const found = data.find((l) => l.id === id);
35     setLogement(found || null);
36   })
37     .catch((error) => console.error("Error fetching data:", error));
38 }, [id]);
39
40 < if (!logement) {
41   return <Custom404 />;
42 }
43
44 < return (
45   <>
46     <Header />
47   <main id="detail">
48     <section>
49       <Carousels />
50       <div>
51         <h2>{logement.title}</h2>
52         <p>{logement.location}</p>
53       </div>
54       <div>
55         <{logement.tags.map((tag, index) => (
56           <Badge key={index} className="tags">
57             {tag}
58           </Badge>
59         )));
60       </div>
61       <div className="profil">
62         <Avatar>
63           <AvatarImage
64             src={logement.host.picture}
65             alt={logement.host.name}
66           />
67           <AvatarFallback>{logement.host.name}</AvatarFallback>
68         </Avatar>
69         <div className="profil-detail">
70           <h3>{logement.host.name}</h3>
71           <p>{logement.location}</p>
72         </div>
73       </div>
74       <div>
75         <Collapse />
76       </div>

```

Si l'id ne correspond à aucun logement, l'utilisateur est automatiquement redirigé vers page 404.

```
40  if (!logement) {  
41    return <Custom404 />;  
42  }  
43
```

Pour finir, grâce à l'intégration de shacn/ui, certains composants ont pu être importés directement sans nécessiter de développement supplémentaire, ce qui m'a permis de me concentrer sur la récupération et l'affichage des données des logements.



Page d'accueil

Kasa

Accueil À propos Contact

Trouvez votre logement idéal

Découvrez des hébergements uniques et vivez comme un local partout dans le monde

Appartement cosy à Marseille
Marseille, France
Vue mer, Climatisé
★★★★☆ (4)

Studio moderne à Lyon
Lyon, France
Centre-ville, Animaux autorisés
★★★★☆ (3)

Loft design à Bordeaux
Bordeaux, France
Design, Spacieux
★★★★★ (5)

Maison de charme en Bretagne
Quimper, France
Campagne, Famille
★★★★★ (5)

Appartement avec terrasse à Nice
Nice, France
Terrasse, Vue mer
★★★★☆ (4)

Cabane dans les bois
Annecy, France
Nature, Isolé
★★★★★ (5)

Appartement minimalist à Montpellier
Montpellier, France
Minimaliste, Calm
★★★★★ (5)

Riad traditionnel à Marrakech
Marrakech, Maroc
Exotique, Traditionnel
★★★★☆ (3)

Cabane sur pilotis à Arcachon
Arcachon, France
Original, Nature
★★★★☆ (4)

Appartement rétro à Toulouse
Toulouse, France
Vintage, Charmant
★★★★★ (5)

À propos de Kasa

Notre histoire
Équipe
Carrière
Presse

Support

Centre d'aide
Nous contacter
Signaler un problème
Page 404

Légal

Conditions d'utilisation
Politique de confidentialité
Mentions légales
Cookies

Page Not Found (404)

404

Page non trouvé

La page que vous recherchez n'existe pas
ou a été déplacée.

[Revenir à la page home](#)

Page détails

Kasa

Accueil À propos Contact

Logement Photo 1

Appartement cosy à Marseille

Marseille, France

Vue mer Climatisé



Claire Dupont

Marseille, France



Description

Équipements

À propos de Kasa

Notre histoire

Equipe

Carrière

Presse

Support

Centre d'aide

Nous contacter

Signaler un problème

Page 404

Légal

Conditions d'utilisation

Politique de confidentialité

Mentions légales

Cookies

© 2025 Thomas PENA--BERMOND. Tous droits réservés.

Description



Un charmant appartement situé proche du Vieux-Port.

Équipements

✓ Wifi

✓ Télévision

✓ Cuisine

✓ Climatisation

2. Précisez les moyens utilisés :

Pour développer la partie dynamique des interfaces utilisateur web ou web mobile, j'ai utilisé mes connaissances données lors des cours en formation et les documentations.

3. Avec qui avez-vous travaillé ?

J'ai réalisé ce travail individuellement.

4. Contexte

Nom de l'entreprise, organisme ou association	Centre de formation la plateforme à Brignole
Chantier, atelier, service	Formation Développeur Web & Web Mobile
Période d'exercice	Du 04/06/2025 au 05/06/2025

5. Informations complémentaires (facultatif)

Pour développer la partie dynamique des interfaces utilisateur web ou web mobile, j'ai utilisés :

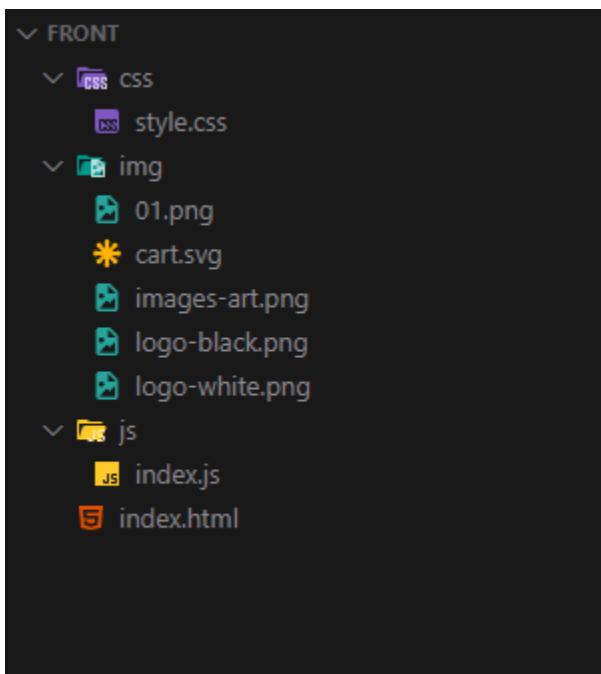
- Le site de shadn/ui
 - Installation : <https://ui.shadcn.com/docs/installation/vite>
 - Composant : <https://ui.shadcn.com/docs/components>
- Le site de tailwind :
 - Installation : <https://tailwindcss.com/docs/installation/using-vite>

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 :

Comme pour chacun de mes projets, j'ai commencé par créer un dossier dédié, que j'ai nommé "GeniArtHub". A l'intérieur, j'ai mis en place la structure de base du projet, composée des éléments suivants :



- Un fichier index.html
- Un dossier css contenant le fichier style.css
- Un dossier js avec le fichier index.js
- Un dossier img destiné aux images

Une fois les fichiers correctement reliés à mon index.html, j'ai pu débuter la phase de développement. J'ai commencé par construire le header et le footer,

```
index.html X
index.html > html
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       <link rel="stylesheet" href="css/style.css" />
7       <title>
8         GeniArtHub - Où la créativité rencontre l'intelligence artificielle
9       </title>
10      </head>
11      <body>
12        <header>
13          <div class="row">
14            <a href="index.html">
15              
16            </a>
17          </div>
18        </header>
19      <main>...
35
36        <footer>
37          <p>
38            <a href="https://www.errantecreation.com/">Errante Creation</a> © 2023 -
39            GeniArtHub
40          </p>
41        </footer>
42        <script src="js/index.js"></script>
43      </body>
44    </html>
45
```

puis j'ai travaillé sur la partie principale de la page, destinée à afficher le contenu principal.

J'ai créé une section "produits", dans laquelle j'ai d'abord réalisé la structure d'un produit type. Une fois celui-ci finalisé, je l'ai dupliqué plusieurs fois afin d'effectuer le travail de mise en page et de style.

```

<main>
  <section class="hero">
    <div class="row">
      <h1>Explorez l'AI-magination artistique</h1>
      <a href="#aiartshop">AI Art Shop</a>
    </div>
  </section>
  <section id="aiartshop" class="productlist">
    <div>
      
      <a href="cart.html"></a>
    </div>
    <section class="products"></section>
  </section>
</main>

```

Une fois la structure et le design terminés, je suis passé à la partie JavaScript. Je savais que j'allais devoir récupérer des données externes, j'ai donc prévu de faire un fetch. Avant cela, j'ai mis en place un fonction init(), qui a pour rôle d'initialiser l'ensemble des autres fonctions du script.

```

async function init() {
  // Initialisation de la page d'accueil
  const data = await getData();
  displayData(data);
}

init();

```

J'ai ensuite créé une fonction getData(), chargée de récupérer les données via un fetch. J'ai bien entendu géré les éventuelles erreurs grâce à un try / catch.

```

async function getData() {
  // Récupération des données depuis l'API
  try {
    const req = await fetch("http://localhost:3000/api/products/"); // URL de l'API
    return await req.json(); // Conversion de la réponse en JSON
  } catch (error) {
    throw new Error(error.message);
  }
}

```

Une fois les données récupérées, j'ai développé une fonction displayData(). Celle-ci utilise une boucle forEach() pour parcourir les éléments du backend (api) et les afficher dynamiquement dans la page à l'aide de innerHTML. (J'ai ainsi supprimer tous le contenu dans la section produit côté html, vu que je le gère depuis js)

```
function displayData(data) {
    // Affichage des données sur la page
    data.forEach(({ _id, titre, image }) => {
        // Pour chaque produit, on crée un article
        // Destructuration de l'objet
        const article = `
            <article>
                
                <a href="product.html?id=${_id}">${titre}</a>
            </article>`;

        document.querySelector(".products").innerHTML += article; // Ajout de l'article sur la page
    });
}
```

Après la page principale, je me suis attaqué à la page de détails du produit. J'ai reproduit une structure similaire, avec un fichier `product.html`, le fichier `style.css`, et un fichier [product.js](#).



product.html X

```
product.html > ...
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       <link rel="stylesheet" href="css/style.css" />
7       <title>TITRE DE L'OEUVRE - GeniArtHub</title>
8     </head>
9     <body class="page">
10    <header>
11      <div>
12        <a href="index.html">
13          
14        </a>
15        <a href="cart.html"></a>
16      </div>
17    </header>
18
19    <section class="detailoeuvre">
20      <article>
21        <figure>
22          
23        </figure>
24        <div>
25          <h1>Éclat Éthéré : Bird</h1>
26          <p>
27            Plongez dans l'univers mystique de 'Bird', une œuvre d'art
28            captivante qui transcende les limites de la réalité. Réalisée dans
29            le style éthéré et spectral, cette pièce évoque la présence d'un
30            oiseau envoûtant qui semble flotter dans l'au-delà.
31          </p>
32          <div class="price">
33            <p>Acheter pour</p>
34            <span class="showprice">35.25€</span>
35          </div>
36          <div class="declinaison">
37            <input
38              type="number"
39              name="quantity"
40              id="quantity"
41              placeholder="1"
42              value="1"
43              min="1"
44              max="100"
45            />
46            <select name="format" id="format"></select>
47          </div>
48          <a class="button-buy" href="#">Buy bird</a>
49        </div>
50      </article>
51
52      <aside>
53        <h2>Description de l'oeuvre : Éclat Éthéré : Bird</h2>
54      </aside>
55    </section>
56
57    <footer>
58      <p>
59        <a href="https://www.errantecreation.com/">Errante Creation</a> © 2023 -
60        GeniArtHub
61      </p>
62    </footer>
63    <script src="js/product.js"></script>
64  </body>
```

Dans cette page, les produits sont affichés en fonction de leur identifiant (ID) récupéré depuis l'url. J'ai repris la même logique que sur le index.js :

```
1  async function init() {
2      const data = await getData();
3      displayData(data);
4      getLocalStorage(data);
5  }
6  init();
7
8  async function getData() {
9      // Récupère les données de l'API
10     try {
11         const req = await fetch(
12             "http://localhost:3000/api/products/" +
13             new URLSearchParams(window.location.search).get("id") // Récupère l'ID du produit depuis l'URL
14             // window.Location.search permet de récupérer la partie de l'URL après le ?
15         );
16         return await req.json();
17     } catch (error) {
18         throw new Error(error.message);
19     }
20 }
21
```

```
product.js ×
js > product.js > displayData > article
22  function displayData({ titre, image, description, declinaisons, shorttitle }) {
27    const article = `
35      <div class="price">
36        <p>Acheter pour</p>
37        <span class="showprice">09.99€</span>
38      </div>
39      <div class="declinaison">
40        <input
41          type="number"
42          name="quantity"
43          id="quantity"
44          placeholder="1"
45          value="1"
46          min="1"
47          max="100"
48        />
49        <select name="format" id="format">
50          <option value="" disabled selected>Choisir un format</option>
51        </select>
52      </div>
53      <a class="button-buy" href="#">Buy ${shorttitle}</a>
54    </div>
55  </article>
56
57  <aside>
58    <h2>Description de l'œuvre :</h2>
59    <p>${description.replace(/\n/g, "<br />")}</p>
60  </aside>;
61 // description.replace(/\n/g, "<br />") sert à remplacer les sauts de ligne par des balises <br> pour l'affichage HTML
62 document.querySelector(".detailœuvre").innerHTML = article;
63
64 const select = document.querySelector("select");
65 const priceDisplay = document.querySelector(".showprice");
66
67 declinaisons.forEach(({declinaison, index}) => {
68   // Pour chaque déclinaison, on crée une option dans le select
69   const option = document.createElement("option");
70   option.value = index;
71   option.textContent = `Format : ${declinaisons.taille}`;
72   select.appendChild(option);
73 });
74
75 priceDisplay.textContent = `${declinaisons[0].prix}€`;
76 select.addEventListener("change", () => {
77   // Met à jour le prix affiché en fonction de la sélection
78   const index = select.value;
79   priceDisplay.textContent = `${declinaisons[index].prix}€`; // Affiche le prix de la déclinaison sélectionnée
80 });
81
82 const buyButton = document.querySelector(".button-buy");
83 buyButton.addEventListener("click", (e) => {
84   e.preventDefault();
85
86   const quantity = parseInt(document.querySelector("#quantity").value); // Récupère la quantité saisie par l'utilisateur
87   const index = select.value;
88
89   if (index === "" || declinaisons[index] === undefined) {
90     // Vérifie si un format a été sélectionné
91     alert("Choisissez un format valide avant d'acheter.");
92     return;
93   }
94
95   const { taille: format, prix: price } = declinaisons[index]; // Récupère le format et le prix de la déclinaison sélectionnée
96
97   if (quantity < 1) {
98     // Vérifie si la quantité est valide
99     alert("Vous devez commander au moins une oeuvre");
100    return;
101  }
102
103  if (quantity > 100) {
104    // Vérifie si la quantité ne dépasse pas 100
105    alert("Vous ne pouvez pas commander plus de 100 œuvres");
106    return;
107  }
108
109  alert(
110    // Affiche un message de confirmation
111    `Vous avez ajouté ${quantity} exemplaire(s) de ${shorttitle} (${format}) au produit pour un total de ${
112      price * quantity
113    }€
114  );
115});
```

```

function getLocalStorage({ _id, titre, image, shorttitle, declinaisons }) {
    // Récupère les données du localStorage et ajoute un article au panier
    const buttonBuy = document.querySelector(".button-buy");
    buttonBuy.addEventListener("click", (e) => [
        // Ajoute un écouteur d'événement au bouton d'achat
        e.preventDefault();

        const quantity = document.querySelector("#quantity").value;
        const index = document.querySelector("#format").value;

        if (index === "") {
            // si index est vide, on ne peut pas ajouter l'article
            return; // return rien
        }

        const format = declinaisons[index].taille;
        // declinaisons[index] permet de récupérer la déclinaison sélectionnée dans le select

        const article = {
            // Création de l'article à ajouter au panier
            _id,
            titre,
            shorttitle,
            image,
            format,
            quantity,
        };

        const products = JSON.parse(localStorage.getItem("products")) || [];
        // Récupère les produits localStorage ou initialise un tableau vide
        products.push(article); // Ajoute l'article au tableau des produits
        localStorage.setItem("products", JSON.stringify(products)); // Enregistre le tableau des produits
        localStorage
    ]);
}

```

- Une fonction init() pour initialisation
- Une fonction getData() pour récupérer les données selon l'ID
- Une fonction displayData() pour afficher les informations correspondantes
- Et j'ai rajouté une fonction getLocalStorage() pour gérer les interactions avec le panier

Cette dernière fonction définit un bouton sur lequel un écouteur d'événement est placé. Lors du click, la quantité et le format choisis sont récupérés, puis un article est créé et ajouté au panier, stocké dans le localStorage (le prix n'y est pas enregistré, pour des raisons de sécurité).

Ensuite, j'ai développé la page du panier (cart.html), sur le même principe.

cart.html

```
cart.html > ...
2   <html lang="fr">
9     <body class="page">
10    <header>
11      <div>
12        </div>
13      </header>
14
15      <main id="panier-container">
16        <section>
17          <h1>Mon Panier</h1>
18          <table>
19            <thead>
20              <tr>
21                <th>Nom</th>
22                <th>Format</th>
23                <th>Prix Unitaire</th>
24                <th>Quantité</th>
25                <th>Sous-total</th>
26                <th>Supprimer</th>
27              </tr>
28            </thead>
29            <tbody></tbody>
30          </table>
31          <div class="total">Total : 00€</div>
32        </section>
33        <section>
34          <h2>Formulaire de commande</h2>
35          <form class="form-group">
36            <div class="form-row">
37              <div>
38                <label for="prenom">Prénom: </label>
39                <input type="text" name="prenom" />
40              </div>
41              <div>
42                <label for="nom">Nom: </label>
43                <input type="text" name="nom" />
44              </div>
45              <div>
46                <label for="adresse">Adresse: </label>
47                <input type="text" name="adresse" />
48              </div>
49              <div>
50                <label for="mail">Email: </label>
51                <input type="text" name="mail" />
52              </div>
53            </div>
54          </form>
55        </section>
56        <div>
57          <a href="#" class="btn-checkout">Valider à la commande</a>
58        </div>
59      </main>
60
61      <footer>
62        <p>
63          <a href="https://www.errantecreation.com/">Errante Creation</a> © 2023 -
64          GeniArtHub
65        </p>
66      </footer>
67      <script src="js/order.js"></script>
68    </body>
69  </html>
```

Dans le fichier order.js, j'ai intégré une fonction permettant de récupérer les produits stockés dans le localStorage, ainsi de comparer les données et de les afficher à l'écran.

```
js order.js x
js > js order.js > mergeProducts > products.forEach() callback
1 let products = [] // Tableau pour stocker les produits du panier pour plus tard
2
3 async function init() {
4     const data = await getData(); // Récupère les données de l'API
5     products = await recoverLocalStorage(); // Récupère les produits du localStorage
6     compareData(data, products); // Compare les données de l'API avec celles du localStorage
7     deleteProduct(products); // Ajoute la fonctionnalité de suppression des produits du panier
8 }
9 init();
10
11 async function getData() {
12     try {
13         const req = await fetch("http://localhost:3000/api/products");
14         return await req.json();
15     } catch (error) {
16         console.error("Failed to fetch data:", error);
17     }
18 }
19
20 async function recoverLocalStorage() {
21     try {
22         const products = JSON.parse(localStorage.getItem("products")) || [];
23         return mergeProducts(products);
24     } catch (error) {
25         console.error("Failed to recover local storage:", error);
26     }
27 }
28
29 function mergeProducts(products) {
30     const mergeArray = [];
31     // On parcourt le tableau products pour fusionner les produits
32     // ayant le même id et le même format
33     products.forEach((product) => {
34         const merge = mergeArray.find(
35             // Vérifie si un produit avec le même identifiant (_id) et le même format existe déjà dans mergeArray
36             // (p) est un paramètre de la fonction fléchée qui représente chaque produit dans mergeArray
37             (p) => p._id === product._id && p.format === product.format
38         );
39         if (merge) {
40             merge.quantity = Number(merge.quantity) + Number(product.quantity);
41             // Number() sinon additionne une chaîne de caractères
42         } else {
43             // { ...product } crées une copie indépendante, ce qui permet de ne pas modifier l'original
44             mergeArray.push({ ...product });
45         }
46     });
47
48     return mergeArray;
49 }
```

```

50
51     function compareData(data, products) {
52         const tbody = document.querySelector("tbody");
53         tbody.innerHTML = ""; // Vide le tbody avant de le remplir
54         let hasProducts = false; // Indique s'il y a des produits à afficher
55         let totalGeneral = 0; // Initialise le total général à 0
56
57         products.forEach((product) => {
58             const items = data.find((item) => item._id === product._id);
59             if (items) {
60                 const declinaison = items.declinaisons.find(
61                     (d) => d.taille === product.format
62                 );
63                 if (declinaison) {
64                     hasProducts = true;
65                     const total =
66                         Math.round(declinaison.prix * product.quantity * 100) / 100; // arrondir à 2 décimales
67                     totalGeneral += total;
68                     const article = `
69                         <tr>
70                             <td>${product.titre}</td>
71                             <td>${product.format}</td>
72                             <td>${declinaison.prix}€</td>
73                             <td>${product.quantity}</td>
74                             <td>${total.toFixed(2)}€</td>
75                             <td><button class="delete-article" data-id="${
76                                 product._id
77                             }" data-format="${product.format}">X</button></td>
78                         </tr>
79                     `;
80                     tbody.insertAdjacentHTML("beforeend", article);
81                 }
82             }
83         });
84
85         if (!hasProducts) {
86             // Si aucun produit n'est trouvé, on affiche un message
87             tbody.innerHTML = "<tr><td colspan='6'>Panier vide</td></tr>";
88         }
89
90         const totalPrice = document.querySelector(".total");
91         if (totalPrice) {
92             // si totalPrice existe on affiche
93             totalPrice.textContent = `${totalGeneral.toFixed(2)}€`; // arrondir à 2 décimales
94         }
95     }
96

```

```
js order.js  X
js > js order.js > mergeProducts > products.forEach() callback
  97  function deleteProduct(products) {
  98    deleteButtons.forEach((button) => {
  99      button.addEventListener("click", () => {
100
101        for (let i = 0; i < products.length; i++) {
102          // Parcourt le tableau products pour trouver le produit à supprimer
103          if (products[i]._id === id && products[i].format === format) {
104            // Si le produit est trouvé, on le supprime
105            products[i].quantity -= 1;
106            // On décrémente la quantité du produit de 1
107            if (products[i].quantity <= 0) {
108              // Si la quantité est inférieure ou égale à 0, on supprime le produit du tableau
109              products.splice(i, 1); // Supprime le produit du tableau avec splice
110            }
111            // Met à jour le localStorage avec le tableau modifié
112            localStorage.setItem("products", JSON.stringify(products));
113            location.reload();
114            break;
115          }
116        }
117      });
118    });
119  }
120
121  });
122  });
123}
124
// orderValidation
125function secureForm(contact) {
126  const form = document.querySelector(".form-group");
127  const datas = new FormData(form); // Récupère les données du formulaire
128  let valid = true; // Variable pour vérifier la validité du formulaire
129
130  // Vérification des champs du formulaire si null ou trop courts
131  if (datas.get("prenom") === null || datas.get("prenom").length < 2) {
132    console.log("Le prénom doit contenir au moins 2 caractères");
133    valid = false;
134  }
135
136  if (datas.get("nom") === null || datas.get("nom").length < 2) {
137    console.log("Le nom doit contenir au moins 2 caractères");
138    valid = false;
139  }
140
141  if (datas.get("adresse") === null || datas.get("adresse").length < 10) {
142    console.log("L'adresse doit contenir au moins 10 caractères");
143    valid = false;
144  }
145
146  const email = datas.get("mail");
147  const emailRegex = /^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,6}$/; // Regex pour valider l'email
148  if (email === null || !emailRegex.test(email)) {
149    console.log("L'email n'est pas valide");
150    valid = false;
151  }
152
153
// si le formulaire est valide
154  if (valid) {
155    console.log("Le formulaire est valide.");
156
157    const form = document.querySelector(".form-group");
158    const datas = new FormData(form);
159
160    return {
161      // ...
162    };
163  }
164}
```

```
async function getOrderNumber(contact, products) {
  try {
    const req = await fetch("http://localhost:3000/api/products/order", {
      method: "POST",
      headers: {
        "Content-Type": "application/json",
      },
      body: JSON.stringify({ contact, products }),
    });

    const res = await req.json();
    console.log("Numéro de commande:", res.orderId);

    localStorage.setItem("orderId", res.orderId); // Stocke le numéro de commande dans le LocalStorage
    // localStorage.removeItem("products"); // Vide le panier
    // document.querySelector(".form-group").reset(); // Réinitialise le formulaire
    // window.Location.href = "confirmation.html"; // Redirige vers une page de confirmation
  } catch (error) {
    console.error(
      "Erreur lors de la récupération du numéro de commande:",
      error
    );
  }
}

function orderValidation(message, title = "") {
  // Crée un modal pour afficher le message de validation
  const modal = document.createElement("dialog");

  if (title) {
    // Si un titre est fourni, on l'affiche
    const h1 = document.createElement("h1");
    h1.textContent = title;
    modal.appendChild(h1);
  }
  if (message) {
    // Si un message est fourni, on l'affiche
    const p = document.createElement("p");
    p.textContent = message;
    modal.appendChild(p);
  }

  document.body.appendChild(modal);
  modal.showModal();

  setTimeout(() => {
    modal.close();
    modal.remove();
  }, 3000);
}
```

```

222 const btnvalid = document.querySelector(".btn-checkout");
223 btnvalid.addEventListener("click", async (e) => {
224   e.preventDefault();
225   const contact = secureForm();
226
227   // Si le formulaire n'est pas valide, on affiche un message d'erreur
228   if (!contact) {
229     console.log("");
230     orderValidation(
231       "Veuillez corriger les erreurs dans le formulaire.",
232       "Erreur de validation"
233     );
234     return;
235   }
236
237   // Si le formulaire est valide, on envoie la commande
238   try {
239     await getOrderNumber(contact, products);
240     orderValidation("Votre commande a bien été envoyée !", "Commande validée");
241   } catch (err) {
242     orderValidation(err, "Erreur de commande");
243     console.error(err);
244   }
245 });
246
247 // console err
248 // Le formulaire est valide.
249 //POST http://localhost:3000/api/products/order 400 (Bad Request)
250 //Numéro de commande: undefined
251

```

Pour finir, j'ai mis en place une vérification du formulaire : lors de la validation, une modale de confirmation s'affiche, contenant un message de remerciement ainsi que le numéro de commande généré.

2. Précisez les moyens utilisés :

Pour développer la partie dynamique des interfaces utilisateur web ou web mobile, j'ai utilisé mes connaissances données lors des cours en formation.

3. Avec qui avez-vous travaillé ?

J'ai réalisé ce travail individuellement.

4. Contexte

Nom de l'entreprise, organisme ou association Centre de formation la plateforme à Brignole

Chantier, atelier, service Formation Développeur Web & Web Mobile

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

5. Informations complémentaires (*facultatif*)

Activité-type 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 :

Pour débuter ce projet, j'ai établi un tableau récapitulatif des user stories accompagnées de leurs fonctionnalités et explications :

User Story	Fonctionnalité	Explication
US1	Créer un compte utilisateur	La requête créer un compter utilisateurs dans la table "users" et enregistre les valeurs telle que le nom, email, le mot de passe, la date de création du compte, ainsi que le rôle
US2	Voir tout les événements	La requête sélectionne le nom, la description, la date et le lieu des événements depuis la table "events"
US3	S'inscrire à un événement	La requête associe l'id de l'utilisateur (fk_id_user) ainsi que celui de l'événement (fk_id_event) et l'insère dans la table registration
US4	Voir mes inscription	
US5	Ajouter un événement	la requête ajoute un évènements avec les valeurs comme le nom, la description, la date, et le lieu dans la table "events"
US6	Liste des Participants d'un événement	
US7	Compter les participants par événement	La requête utilise l'id de l'évenement ainsi que count pour afficher le nombre total puis les regroupes
US8	Lister tous les utilisateurs	La requête sélectionne toutes valeurs avec "*" de la table "users"
US9	Modifier un événement	La requête mets à jour la description de l'évenement via la table "events"
US10	Annuler une inscription	La requête supprime la liaison entre l'id de l'utilisateur et de l'évenement de la table "registrations"
US11	Supprimer un événement	La requête supprime toutes les valeurs de l'évevents par la condition where

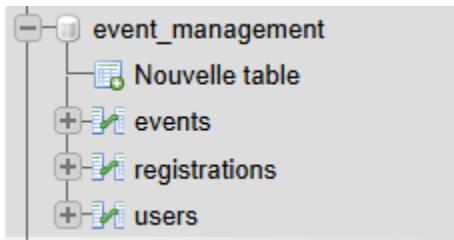
Après avoir défini les user stories, j'ai réfléchi à la conception de la base de données. J'ai utilisé WampServer pour la mise en place du projet, en créant un dossier nommé CP5_PENABERMOND_Thomas dans le répertoire www.

This PC > Local Disk (C:) > wamp64 > www			
Name	Date modified	Type	Size
CP5_PENABERMOND_Thomas-main	7/29/2025 5:55 PM	File folder	

Depuis phpMyAdmin, j'ai créé une base de données nommée event_management, dédiée à la gestion d'événements.

```
CREATE DATABASE event_management;
USE event_management;
CREATE TABLE users (
    id_user INT PRIMARY KEY AUTO_INCREMENT,
    name_user VARCHAR(100) NOT NULL,
    email_user VARCHAR(191) NOT NULL UNIQUE,
    password_user VARCHAR(191) NOT NULL,
    registration_date_user TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    role_user ENUM('participant', 'organisateur', 'administrateur') NOT NULL
);
CREATE TABLE events (
    id_event INT PRIMARY KEY AUTO_INCREMENT,
    name_event VARCHAR(100) NOT NULL,
    description_event TEXT,
    date_event DATE NOT NULL,
    place_event VARCHAR(100) NOT NULL,
    fk_id_user INT,
    FOREIGN KEY (fk_id_user) REFERENCES users(id_user) ON DELETE CASCADE
);
CREATE TABLE registrations (
    id_registration INT PRIMARY KEY AUTO_INCREMENT,
    fk_id_user INT,
    fk_id_event INT,
    FOREIGN KEY (fk_id_user) REFERENCES users(id_user) ON DELETE CASCADE,
    FOREIGN KEY (fk_id_event) REFERENCES events(id_event) ON DELETE CASCADE,
    UNIQUE(fk_id_user, fk_id_event)
);
```

Cette base comporte trois tables principales :



- Users : contient les informations relatives aux utilisateurs (id, nom, email, mot de passe, date d'inscription, et rôle)
- Events : regroupe les informations sur les événements (id, nom, description, date, lieu, et clé étrangère reliant l'utilisateur organisateur)
- Registrations : fait la liaison entre les utilisateurs et les événements grâce à deux clés étrangères

Table	Action	Lignes	Type	Interclassement	Taille	Perte
events	★ Parcourir Structure Rechercher Insérer Vider Supprimer	10	MyISAM	utf8mb4_0900_ai_ci	4,4 kio	-
registrations	★ Parcourir Structure Rechercher Insérer Vider Supprimer	0	MyISAM	utf8mb4_0900_ai_ci	1,0 kio	-
users	★ Parcourir Structure Rechercher Insérer Vider Supprimer	20	MyISAM	utf8mb4_0900_ai_ci	9,6 kio	-
3 tables	Somme					

Une fois la base créée, je l'ai peuplée avec plusieurs utilisateurs et événements fictifs afin de pouvoir tester les différentes fonctionnalités de l'application.

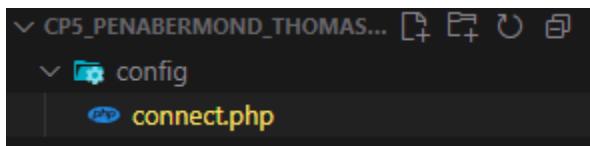
```

-- 5 users
-- 2 Participant
INSERT INTO users (name_user, email_user, password_user, registration_date_user, role_user) VALUES ('Mohamed', 'mohamed@mail.com', 'OG}cLJMa~ms/', '24/02/2025', 'participant');
INSERT INTO users (name_user, email_user, password_user, registration_date_user, role_user) VALUES ('Emma', 'emma@mail.com', 'Q^@HC=yYFzOs', '08/04/2025', 'participant');

-- 2 Organisateur
INSERT INTO users (name_user, email_user, password_user, registration_date_user, role_user) VALUES ('Marie', 'marie@mail.com', 'VykWc0z7~~]0', '12/07/2025', 'organisateur');
INSERT INTO users (name_user, email_user, password_user, registration_date_user, role_user) VALUES ('Santiago', 'santiago@mail.com', 't|mkCDSY<kz:', '29/10/2025', 'organisateur');
-- 1 Administrateur
INSERT INTO users (name_user, email_user, password_user, registration_date_user, role_user) VALUES ('Sophie', 'sophie@mail.com', 'K#,3x1nv/Zp_', '17/12/2025', 'administrateur');

-- 5 events
INSERT INTO events (name_event, description_event, date_event, place_event) VALUES ('South by South West', 'A major cultural and technological event held in Austin, Texas, combining music, film, and technology', '24/02/2025', 'Texas');
INSERT INTO events (name_event, description_event, date_event, place_event) VALUES ('Sundance Film Festival', 'A major independent film festival held in Park City, Utah, attracting filmmakers and actors from around the world', '08/04/2025', 'Utah');
INSERT INTO events (name_event, description_event, date_event, place_event) VALUES ('Wimbledon', 'The most prestigious tennis tournaments in the world, held in London, Great Britain', '12/07/2025', 'London');
INSERT INTO events (name_event, description_event, date_event, place_event) VALUES ('King's Day', 'A Dutch national holiday marked by parades, orange clothing, and festivities throughout the country.', '29/10/2025', 'Asahi');
INSERT INTO events (name_event, description_event, date_event, place_event) VALUES ('Stampede de Calgary', 'Western event celebrating the city's heritage with rodeos, concerts, and exhibitions', '17/12/2025', 'Calgary');
```

Ensuite, j'ai créé un dossier config pour la connexion à la base de données.



Cette connexion est importée dans les fichiers principaux grâce à une requête.

A screenshot of a code editor window. The title bar says 'connect.php X'. Below the title bar, it shows the path 'config > connect.php'. The main area of the editor contains the following PHP code:

```
1  <?php
2  $host = 'localhost';
3  $dbname = 'event_management';
4  $username = 'root';
5  $password = '';
6  <?>
7  <?>
8  <?>
9  <?>
10 <?>
11 <?>
12 <?>
13 <?>
14 <?>
15 <?>
16 <?>
```

The code defines variables for host, database name, username, and password. It then attempts to create a PDO object with the MySQL connection string and specified error mode. If an exception occurs, it outputs an error message using die().

Sur la page index.php, j'ai effectué une requête sql pour afficher le nom, la description, la date et le lieu des événements. Le rendu s'effectue via un foreach, en php, qui parcourt les résultats et les affiche directement grâce à la syntaxe abrégée.

```
index.php x
index.php
1 <?php
2 require_once('config/connect.php');
3
4 $stmt = $pdo->prepare("SELECT name_event, description_event, date_event, place_event FROM events");
5 $stmt->execute();
6 $events = $stmt->fetchAll(PDO::FETCH_ASSOC);
7 ?>
8 <!DOCTYPE html>
9 <html lang="en">
10 <head>
11   <meta charset="UTF-8">
12   <meta name="viewport" content="width=device-width, initial-scale=1.0">
13   <link rel="stylesheet" href="style.css">
14   <title>Accueil</title>
15 </head>
16 <body>
17   <header>
18     <a href="index.php"><h1>Events</h1></a>
19   </header>
20   <main id = "events">
21     <section>
22       <?php foreach($events as $event): ?>
23         <article>
24           <div>
25             <h2><?= ($event['name_event']) ?></h2>
26             <div class="badge">
27               <p><?= ($event['place_event']) ?></p>
28               <p><?= ($event['date_event']) ?></p>
29             </div>
30           </div>
31           <p><?= ($event['description_event']) ?></p>
32           <div>
33             <a href="signup.php">s'inscre</a>
34             <a href="delete.php">annuler</a>
35           </div>
36         </article>
37       <?php endforeach; ?>
38     </section>
39   </main>
40   <footer>by inhuman</footer>
41 </body>
42 </html>
```

J'ai ensuite conçu une page `signup.php` permettant aux utilisateurs de s'inscrire à un événement. Lorsqu'un formulaire est soumis, les champs (nom, email, mot de passe, rôle) sont vérifiés. Le mot de passe est ensuite haché, la date d'inscription enregistrée, puis les informations sont insérées dans la table `users`. Un `rowCount()` permet de vérifier si l'insertion s'est bien déroulée.

```
signup.php X
signup.php
1 <?php
2 require_once('config/connect.php');
3
4 if (!empty($_POST['register'])) {
5     $name = $_POST['name'];
6     $email = $_POST['email'];
7     $password = $_POST['password'];
8     $role = $_POST['role'];
9
10    if (empty($name) || empty($email) || empty($password) || empty($role)) {
11        exit("Attention : Tous les champs doivent être renseignés.");
12    }
13
14    $checkStmt = $pdo->prepare('SELECT COUNT(*) FROM users WHERE email_user = :email');
15    $checkStmt->execute(['email' => $email]);
16    if ($checkStmt->fetchColumn() > 0) {
17        exit("<p style='color: red;'>Cet email est déjà utilisé. Veuillez en choisir un autre.</p>");
18    }
19
20    $hashedPassword = password_hash($password, PASSWORD_DEFAULT);
21    $date = date('Y-m-d H:i:s');
22
23    $stmt = $pdo->prepare('INSERT INTO users (name_user, email_user, password_user, registration_date_user,
24                           role_user) VALUES (:name, :email, :password, :date, :role)');
25    $stmt->execute([
26        'name' => $name,
27        'email' => $email,
28        'password' => $hashedPassword,
29        'date' => $date,
30        'role' => $role
31    ]);
32
33    if ($stmt->rowCount() > 0) {
34        $lastId = $pdo->lastInsertId();
35        echo "<p style='color: green;'>Nouveau membre ajouté avec succès. ID inséré : $lastId</p>";
36        echo '<meta http-equiv="refresh" content="3;url=index.php">';
37        exit();
38    } else {
39        echo "Aucune ligne insérée";
40    }
41 ?>
42
43 <!DOCTYPE html>
44 <html lang="en">
45 <head>
46     <meta charset="UTF-8">
47     <meta name="viewport" content="width=device-width, initial-scale=1.0">
48     <link rel="stylesheet" href="style.css">
49     <title>Register</title>
50 </head>
51 <body>
52     <header>
53         <a href="index.php"><h1>Events</h1></a>
54     </header>
55     <main id="signup">
56         <h2>Formulaire</h2>
57         <form action="#" method="post">
58             <input type="text" name="name" id="name" placeholder=" Name" required>
59             <input type="email" name="email" id="email" placeholder=" Email" required>
60             <input type="password" name="password" id="password" placeholder=" Password" required>
61             <select name="role" id="role" required>
```

Enfin, un fichier delete.php permet à un utilisateur d'annuler son inscription. Le système vérifie la validité des informations saisies et supprime la ligne correspondante dans la table registrations si la correspondance est trouvée

```
delete.php X
delete.php
1  <?php
2  require_once('config/connect.php');
3
4  $message = '';
5  $step = 'form';
6
7  if ($_SERVER['REQUEST_METHOD'] === 'POST') {
8      $email = $_POST['email'] ?? '';
9      $password = $_POST['password'] ?? '';
10     $idEvent = $_POST['id_event'] ?? 0;
11
12     // Vérifie si les champs sont remplis
13     if (empty($email) || empty($password) || empty($idEvent)) {
14         $message = "Tous les champs sont obligatoires.";
15     } else {
16         // Vérifie l'email
17         $stmt = $pdo->prepare('SELECT id_user, password_user FROM users WHERE email_user = :email');
18         $stmt->execute(['email' => $email]);
19         $user = $stmt->fetch(PDO::FETCH_ASSOC);
20
21         if ($user && password_verify($password, $user['password_user'])) {
22             // Supprimer l'inscription à l'événement
23             $stmtDel = $pdo->prepare('DELETE FROM registrations WHERE fk_id_user = :id_user AND fk_id_event = :id_event');
24             $stmtDel->execute([
25                 'id_user' => $user['id_user'],
26                 'id_event' => $idEvent
27             ]);
28
29             if ($stmtDel->rowCount() > 0) {
30                 $message = "Inscription annulée avec succès.";
31                 $step = 'done';
32             } else {
33                 $message = "Aucune inscription trouvée à cet événement pour cet utilisateur.";
34             }
35         } else {
36             $message = "Identifiants incorrects.";
37         }
38     }
39 }
40 ?>
41
```

```
41  <!DOCTYPE html>
42  <html lang="fr">
43  <head>
44  |   <meta charset="UTF-8">
45  |   <title>Annuler une inscription</title>
46  </head>
47  <body>
48
49  </body>
50  </html>
51
52
53
54  <!DOCTYPE html>
55  <html lang="en">
56  <head>
57  |   <meta charset="UTF-8">
58  |   <meta name="viewport" content="width=device-width, initial-scale=1.0">
59  |   <link rel="stylesheet" href="style.css">
60  |   <title>Register</title>
61  </head>
62  <body>
63  |   <header>
64  |   |   <a href="index.php"><h1>Events</h1></a>
65  |   </header>
66  |   <main id="signup">
67  |   <h1>Annuler votre inscription</h1>
68  |   <?php if (!empty($message)): ?>
69  |   |   <p style="color: <?= $step === 'done' ? 'green' : 'red' ?>"><?= $message ?></p>
70  |   <?php endif; ?>
71
72  |   <?php if ($step === 'form'): ?>
73  |   <form method="post" action="cancel_registration.php">
74  |   |   <input type="email" name="email" id="email" placeholder=" Email" required>
75  |   |   <input type="password" name="password" id="password" placeholder=" Password" required>
76  |   |   <input type="number" name="id_event" required>
77  |   |   <button type="submit">Annuler mon inscription</button>
78  |   </form>
79  |   <?php endif; ?>
80
81  |   <p><a href="index.php">Retour à l'accueil</a></p>
82  </main>
83  <footer>by inhuman</footer>
84  </body>
85  </html>
```

2. Précisez les moyens utilisés :

Pour mettre en place une base de données relationnelle, j'ai utilisé mes connaissances données lors des cours en formation et les documentations.

3. Avec qui avez-vous travaillé ?

J'ai réalisé ce travail individuellement.

4. Contexte

Nom de l'entreprise, organisme ou association Centre de formation la plateforme à Brignole

Chantier, atelier, service Formation Développeur Web & Web Mobile

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

5. Informations complémentaires (*facultatif*)

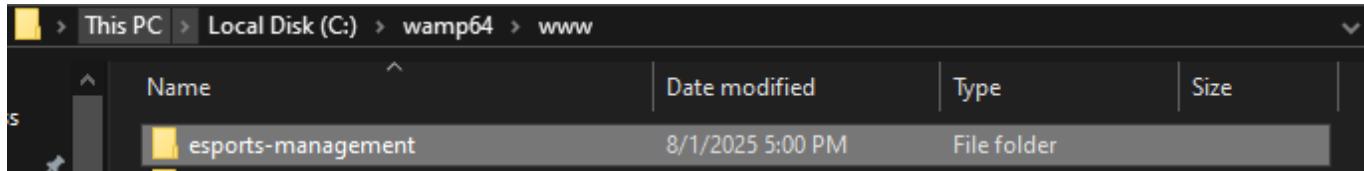
Activité-type

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 :

Pour commencer, comme pour tous mes projets en PHP, j'ai démarré WampServer puis je me suis rendu sur phpMyAdmin afin de créer ma base de données. Une fois cette étape réalisée, j'ai ouvert le dossier www et créé un dossier nommé esports_management, qui constitue la racine de mon projet.



A l'intérieur j'y ai ajouté un dossier config dans lequel j'ai mis en place la connexion à la base de données. (Cette fois j'ai créé un utilisateur pour la bdd, ce qui me permet d'ajouter une sécurité sur les bases de données.)

```
connect.php X
config > connect.php
1  <?php
2  // Paramètres pour la connexion à la base de données
3  $host = 'localhost'; // Hôte de la base de données
4  $dbname = 'esports'; // Nom de la base de données
5  $username = 'esportAdmin'; // Nom d'utilisateur pour se connecter à la base de données
6  $password = 'esport1234'; // Mot de passe de l'utilisateur
7  try {
8      // PDO pour se connecter à la base de données
9      $pdo = new PDO(
10          "mysql:host=$host;dbname=$dbname;charset=utf8",
11          $username,
12          $password,
13          [PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION]
14      );
15  } catch (PDOException $e) {
16      // Message d'erreur en cas d'échec au niveau de la connexion de la base de données
17      die("Erreur de connexion : ".$e->getMessage());
18  }
19 ?>
```

Une fois cette connexion conditionnelle, le projet était prêt à être développé. J'ai ensuite créé mon fichier principal index.php. Étant en PHP, j'ai pu mettre en place des composants réutilisables. J'ai donc créé un dossier components, contenant deux fichiers pour l'instant : header.php et footer.php, que j'ai inclus dans la page principale grâce à la fonction include().

```
index.php X
index.php
1 <!DOCTYPE html>
2 <html Lang="fr-FR">
3   <head>
4     <meta charset="UTF-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6     <link rel="stylesheet" href="styles/global.css" />
7     <link rel="shortcut icon" href="public/logo.png" type="image/x-icon">
8     <title>Esport - Accueil</title>
9   </head>
10  <!-- Inclus les composants dans la page principale -->
11  <body id="page-main">
12    <?php include 'components/header.php'; ?>
13    <main> ...
14    </main>
15    <?php include 'components/footer.php'; ?>
16  </body>
17  </html>
```

J'ai ensuite développé la partie authentification des utilisateurs. Pour cela, j'ai créé un dossier auth dans lequel se trouvent deux fichiers login.php et register.php. Dans register.php, j'ai conçu un formulaire d'inscription permettant à l'utilisateur de saisir son nom, son e-mail et son mot de passe. Lorsqu'il clique sur le bouton d'envoi, les données sont récupérées en méthode POST.

```

<!DOCTYPE html>
<html lang="fr-FR">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link rel="stylesheet" href="../styles/global.css" />
    <link rel="shortcut icon" href="../public/logo.png" type="image/x-icon" />
    <title>Esport Inscription</title>
  </head>
  <body id="page-register">
    <main>
      <h1>Formulaire</h1>
      <form action="#" method="post">
        <input type="text" name="name" id="name" placeholder=" Nom" required/>
        <input type="email" name="email" id="email" placeholder=" Email" required/>
        <input
          type="password"
          name="password"
          id="password"
          placeholder=" Mot de passe"
          required/>
        <input type="submit" name="register" value="Enregister" />
      </form>
    </main>
  </body>
</html>

```

Le script vérifie que tous les champs sont bien renseignés, sinon un message d'erreur s'affiche pour inviter l'utilisateur à compléter les champs manquants. Le mot de passe est ensuite haché avant d'être stocké dans la base de données, garantissant la sécurité des données utilisateur. Si l'insertion réussie, les variables (nom, email, mot de passe haché) sont enregistrées dans la session PHP, puis stockées en base. Pour confirmer le bon fonctionnement, j'ai utilisé la méthode rowCount().

```
register.php X
auth > register.php
1  <?php
2  require_once('../config/connect.php'); // Inclusion du fichier connect.php
3  session_start(); // Démarrer la session ou continue celle existante
4
5  if (!empty($_POST['register'])) { // Vérifie si le formulaire d'inscription a été soumis
6      // Récupère et arrange les données du formulaires
7      $name = htmlspecialchars(trim($_POST['name'])); // Protège des attaques XSS et supprime les espaces blancs
8      $email = htmlspecialchars(trim($_POST['email']));
9      $password = trim($_POST['password']);
10
11     // Vérifie que tout les champs sont remplis
12     if (empty($name) || empty($email) || empty($password)) {
13         exit("Attention : Tous les champs doivent être renseignés.");
14     }
15
16     // Sachage du mot de passe
17     $hashedPassword = password_hash($password, PASSWORD_DEFAULT);
18
19     // Stock les informations en sessions
20     $_SESSION['name'] = $name;
21     $_SESSION['email'] = $email;
22     $_SESSION['password'] = $hashedPassword;
23
24     // Insère un nouvelle utilisateur avec de nouvelle valeurs
25     $stmt = $pdo->prepare("INSERT INTO users (username, email, password_hash) VALUES (:name, :email, :password)");
26     $stmt->execute([
27         'name' => $name,
28         'email' => $email,
29         'password' => $hashedPassword
30     ]);
31
32     // Vérifie que la ligne a était insérer donc fonctionnelle
33     if ($stmt->rowCount() > 0) {
34         $lastId = $pdo->lastInsertId();
35         echo "<p style='color: green;'>Nouveau membre ajouté avec succès. ID inséré : $lastId</p>";
36         echo "<p>Bienvenu, ". $_SESSION['name']. " ! </p>";
37         echo "<meta http-equiv='refresh' content='3;url=../index.php'>";
38         exit();
39     } else {
40         echo "Aucune ligne insérée";
41     }
42 }
43 ?>
```

Pour la connexion (login.php), j'ai réutilisé le même principe, le script vérifie d'abord les champs du formulaire, puis compare les informations saisies à celles de la base de données. Si l'e-mail correspond et que le mot de passe est correct, les informations de l'utilisateur sont enregistrées dans la session, et il est redirigé vers la page principale (index.php) en tant qu'utilisateur connecté.

```
login.php X ...  
auth > login.php  
1  <?php  
2  require_once('../config/connect.php'); // Inclusion du fichier connect.php  
3  session_start(); // Démarrer la session ou continue celle existante  
4  $message = ""; // Initialise la variable pour l'utiliser par la suite grâce à sa portée globale  
5  
6  if (!empty($_POST['connexion'])) { // Vérifie si le formulaire d'inscription a été soumis  
7      $email = htmlspecialchars(trim($_POST['email'])); // Protège des attaques XSS et supprime les espaces blancs  
8      $password = trim($_POST['password']);  
9  
10     // Vérifie que tous les champs sont remplis  
11     if (empty($email) || empty($password)) {  
12         exit("Attention : Tous les champs doivent être renseignés.");  
13     };  
14  
15     // Réquiert l'adresse mail dans la bdd  
16     $stmt = $pdo->prepare("SELECT * FROM users WHERE email = :email");  
17     $stmt->execute([  
18         'email' => $email  
19     ]);  
20     $user = $stmt->fetch(); // retourne un objet json  
21  
22     // Vérification du mot de passe est correct  
23     if ($user && password_verify($password, $user['password_hash'])) {  
24         $_SESSION['name'] = $user['username']; // stocke le nom dans la session  
25         $_SESSION['email'] = $user['email']; // stocke l'email dans la session  
26         header('Location: /CP7_PENABERMOND_Thomas/index.php'); // Puis nous redirige sur la page principale  
27         exit(); // stop le script  
28     } else {  
29         $message = '<p style="color:red;">Mauvais identifiants</p>';  
30     }  
31 }  
32 ?>  
33 <!DOCTYPE html>  
34 <html lang="fr-FR">  
35     <head>  
36         <meta charset="UTF-8" />  
37         <meta name="viewport" content="width=device-width, initial-scale=1.0" />  
38         <link rel="stylesheet" href="../styles/global.css" />  
39         <link rel="shortcut icon" href="../public/logo.png" type="image/x-icon" />  
40         <title>Esport Inscription</title>  
41     </head>  
42     <body id="page-login">  
43         <main>  
44             <h1>Formulaire</h1>  
45             <form action="#" method="post">  
46                 <input type="email" name="email" id="email" placeholder="Email" required/>  
47                 <input  
48                     type="password"  
49                     name="password"  
50                     id="password"  
51                     placeholder="Password"  
52                     required/>  
53                 <input type="submit" name="connexion" value="Connexion" />  
54             </form>  
55             <?= $message ?>  
56         </main>  
57     </body>  
58 </html>
```

Une fois ce système mis en place, j'ai développé la partie affichage des équipes et des tournois. J'ai récupéré les informations stockées dans la base de données, puis les ai affichées dynamiquement à l'aide de boucles foreach et de la fonction echo. Ce qui me permet d'afficher les différents éléments sans avoir à dupliquer le code.

```
displayTeams.php X
teams > displayTeams.php
1  <?php
2  // Récupère les informations des équipes, membres et utilisateurs en faisant une jointure
3  $stmt = $pdo->prepare("SELECT teams.name, teams.created_at, users.username, team_members.role_in_team FROM
4  teams LEFT JOIN team_members ON teams.id = team_members.team_id LEFT JOIN users ON team_members.user_id =
5  users.id");
6  $stmt->execute();
7  $teams = $stmt->fetchAll(PDO::FETCH_ASSOC);
8  // Crédit d'un tableau pour ne pas avoir de duplications
9  $noDuplicates = [];
10 >
11 <section id="section-teams">
12   <!-- Utilisation d'une boucle pour afficher les résultats -->
13   <?php foreach($teams as $team):?>
14     <?php
15       if(in_array($team['name'], $noDuplicates)) continue;
16       $noDuplicates[] = $team['name'];
17     ?>
18     <article class="section-teams-card">
19       <!-- Nom de l'équipe -->
20       <h3><?= htmlspecialchars($team['name']) ?? 'Nom de la teams' ?></h3>
21       <div>
22         <!-- Nom du membre -->
23         <?php
24           foreach($team as $user):
25             if($team['name'] === $user['name']):
26               <div class="user" >
27                 <p><?= htmlspecialchars($user['username']) ?? 'utilisateur dans la teams' ?></p>
28                 <p><?= htmlspecialchars($user['role_in_team']) ?? 'role de l'utilisateur' ?></p>
29               </div>
30               <?php endif; ?>
31             <?php endforeach ?>
32           <!-- role de l'équipe -->
33           <p><?= htmlspecialchars($team['created_at']) ?? 'date de création de la teams' ?></p>
34         </article>
35         <?php endforeach; ?>
36     </section>
```

```
index.php X
index.php
1  <!DOCTYPE html>
2  <html lang="fr-FR">
3      <head>
4          <meta charset="UTF-8" />
5          <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6          <link rel="stylesheet" href="styles/global.css" />
7          <link rel="shortcut icon" href="public/logo.png" type="image/x-icon">
8          <title>Esport - Accueil</title>
9      </head>
10     <!-- Inclus les composants dans la page principale -->
11     <body id="page-main">
12         <?php include 'components/header.php'; ?>
13         <main>
14             <h2>Tournois</h2>
15             <?php include 'tournaments/displayTournaments.php' ?>
16             <h2>Equipe</h2>
17             <?php include 'teams/displayTeams.php' ?>
18         </main>
19         <?php include 'components/footer.php'; ?>
20     </body>
21 </html>
```

Ensuite, je me suis consacré à la gestion des profils utilisateurs. J'ai créé un dossier users contenant un fichier profil.php, où j'ai mis en place un formulaire permettant de modifier le nom, l'e-mail et le mot de passe de l'utilisateur, ou de supprimer complètement le compte. Pour chaque action, une vérification est faite afin de s'assurer que l'utilisateur est bien connecté. En cas de suppression, une requête SQL supprime le compte de la base de données à partir de l'adresse e-mail enregistrée, puis détruit la session avant de rediriger vers la page principale.

```
profils.php X
users > profils.php
1  <!DOCTYPE html>
2  <html lang="fr-FR">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <link rel="stylesheet" href="../styles/global.css">
7      <link rel="shortcut icon" href="../public/logo.png" type="image/x-icon">
8      <title>Esport Profil</title>
9  </head>
10 <body id="page-profil">
11     <?php include '../components/header.php'; ?>
12     <main>
13         <form action="#" method="post">
14             <div>
15                 <label for="name">Name : <?= htmlspecialchars($_SESSION['name']) ?></label>
16                 <input type="text" name="name" id="name" placeholder="Nouveau Name" />
17                 <input type="submit" name="changeName" value="Changer">
18             </div>
19             <div>
20                 <label for="email">Email : <?= htmlspecialchars($_SESSION['email']) ?></label>
21                 <input type="email" name="email" id="email" placeholder="Nouvelle Email" />
22                 <input type="submit" name="changeEmail" value="Changer">
23             </div>
24             <div>
25                 <label for="password">Mot de passe : </label>
26                 <input
27                     type="password"
28                     name="oldPassword"
29                     id="oldPassword"
30                     placeholder="Ancien password"
31                     />
32                 <input
33                     type="password"
34                     name="newPassword"
35                     id="newPassword"
36                     placeholder="Nouveaux Password"
37                     />
38                 <input type="submit" name="changePassword" value="Changer">
39             </div>
40             <input type="submit" name="delete" class="delete" value="Supprimer le compte">
41         </form>
42     </main>
43     <?php include '../components/footer.php'; ?>
44 </body>
45 </html>
```

Concernant la modification des informations, j'ai mis en place un système qui vérifie quel bouton à été cliqué. Pour chaque modification (nom, e-mail, ou mot de passe), le script vérifie que les champs sont remplis, puis effectue une requête update sur la base de données. La session est ensuite mise à jour avec les nouvelles informations. Pour le mot de passe, j'ai ajouté une sécurité supplémentaire, l'utilisateur doit saisir son ancien mot de passe ainsi que le nouveau. La modification ne s'effectue que si l'ancien mot de passe correspond à celui

enregistré en base.

profils.php

```
users > profils.php
46  <?php
47  require_once('../config/connect.php');
48 // vérifie que l'utilisateur est bien connecté
49 if(!empty($_SESSION['name']) || !empty($_SESSION['email'])){
50 // Puis si le formulaire a été envoyé
51 if(!empty($_POST['delete'])){
52 // supprime l'utilisateur de la base de données
53 $stmt = $pdo->prepare("DELETE FROM users WHERE email = :email");
54 $stmt->execute([
55     'email' => $_SESSION['email']
56 ]);
57 // Supprime les données ainsi que la session de l'utilisateur
58 session_unset();
59 session_destroy();
60 header('location: /CP7_PENABERMOND_Thomas/');
61 exit();
62 }
63 }
64 ?>
65 <?php
66 // Mise à jour des informations utilisateurs
67 if (!empty($_POST['changeName']) || !empty($_POST['changeEmail']) || !empty($_POST['changePassword'])){
68 // récupère l'email qui est en session
69 $email = $_SESSION['email'];
70 // Modifie le nom de l'utilisateur
71 if(!empty($_POST['changeName']) && !empty($_POST['name'])){
72 $stmt = $pdo->prepare('UPDATE users SET username = :name WHERE email = :email'); // mets à jour la bdd
73 $stmt->execute([
74     'name' => $_POST['name'],
75     'email' => $email
76 ]);
77 $_SESSION['name'] = $_POST['name']; // mets à jour la session
78 header('location: /CP7_PENABERMOND_Thomas/');
79 exit();
80 }
81 // Modifie l'email de l'utilisateur
82 if(!empty($_POST['changeEmail']) && !empty($_POST['email'])){
83 $stmt = $pdo->prepare('UPDATE users SET email = :newEmail WHERE email = :email'); // mets à jour la bdd
84 $stmt->execute([
85     'newEmail' => $_POST['email'],
86     'email' => $email
87 ]);
88 $_SESSION['email'] = $_POST['email']; // mets à jour la session
89 header('location: /CP7_PENABERMOND_Thomas/');
90 exit();
91 }
92 // Modifie le mdp de l'utilisateur
93 if(!empty($_POST['changePassword']) && !empty($_POST['oldPassword']) && !empty($_POST['newPassword'])){
94 $stmt = $pdo->prepare('SELECT password_hash FROM users WHERE email = :email');
95 $stmt->execute([
96     'email' => $email
97 ]);
98 $user = $stmt->fetch(PDO::FETCH_ASSOC);
99 // Vérifie l'ancien mot de passe puis hash le nouveau mdp
100 if($user && password_verify($_POST['oldPassword'], $user['password_hash'])){
101     $newHash = password_hash($_POST['newPassword'], PASSWORD_DEFAULT);
102     $stmt = $pdo->prepare('UPDATE users SET password_hash = :newHash WHERE email = :email');
103     $stmt->execute([
104         'newHash' => $newHash,
105         'email' => $email
106     ]);
107 // données stocker seulement en bdd donc pas de mise à jour de sessions
108     header('location: /CP7_PENABERMOND_Thomas/');
109     exit();
110 }
```

Malheureusement, je n'ai pas pu terminer le projet par manque de temps. Il me restait à développer la partie permettant aux utilisateurs de rejoindre des équipes ou de participer à des tournois. Le principe aurait été identique au reste du projet, basé sur le modèle CRUD (Create, Read, Update, Delete), afin de gérer efficacement les données et les interactions entre utilisateurs et tournois.

2. Précisez les moyens utilisés :

Pour développer des composants d'accès aux données SQL et NoSQL, j'ai utilisé mes connaissances données lors des cours en formation et les documentations.

3. Avec qui avez-vous travaillé ?

J'ai réalisé ce travail individuellement.

4. Contexte

Nom de l'entreprise, organisme ou association ► Centre de formation la plateforme à Brignole

Chantier, atelier, service Formation Développeur Web & Web Mobile

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

5. Informations complémentaires (facultatif)

Activité-type 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 :

Pour ce projet, j'ai choisi de mettre en ligne une application de type To-Do-List, composée d'une partie front-end (la partie client), et d'une partie back-end (la partie api). L'objectif était de rendre l'application accessible en ligne, avec un hébergement séparé pour le front et pour le back.

Debug your Render services in Claude Code and Cursor [Try Render MCP >](#)

Render Product Pricing Customers Blog Docs Changelog Company Contact Sign In Get Started

\$ git push

Your fastest path to production

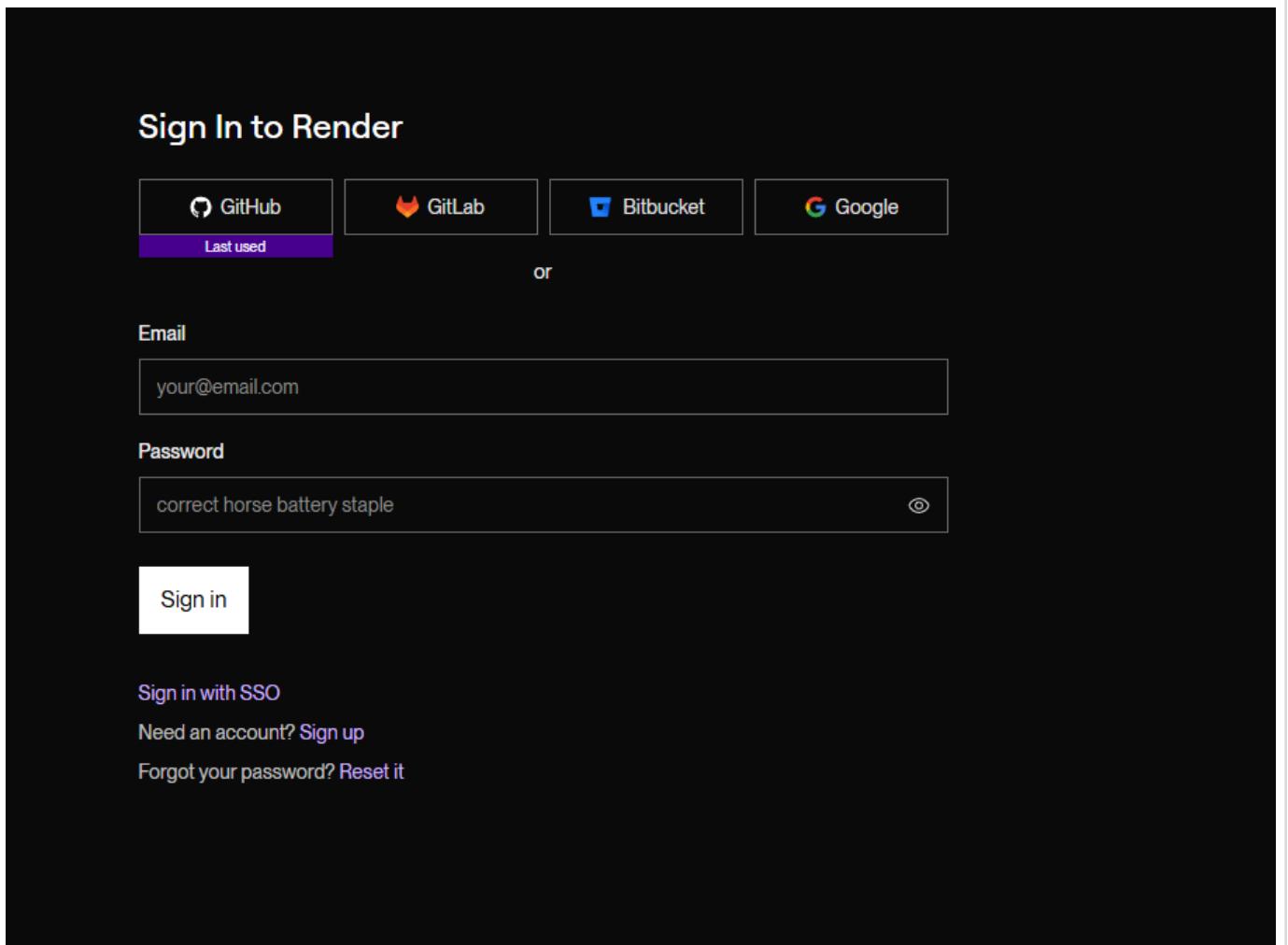
Build, deploy, and scale your apps with unparalleled ease – from your first user to your billionth

[Get started for free >](#) [Contact sales >](#)

Trusted by ambitious teams and product builders

R CBS Watershed twilio Alibaba.com Bridge a stripe company Base 44 Content Square Portola Warner Group Paradig

Pour le back, j'ai utilisé la plateforme Render, une solution simple et gratuite permettant d'héberger un serveur en ligne. Après m'être connecté sur le site render.com, j'ai relié mon compte GitHub à la plateforme afin d'importer directement le dépôt contenant mon code serveur.



J'ai ensuite configuré mon service en lui attribuant un nom, en sélectionnant la branche principale du projet (généralement "main" pour les nouveaux projets, et "master" pour les anciens) et en précisant les commandes nécessaires au lancement du serveur :

- Npm run build : pour la construction du projet
- Npm start : pour le démarrage du serveur

Une fois la configuration terminée, Render s'est chargé d'installer automatiquement les dépendances, de lancer la commande de build et de démarrer le serveur. Mon back-end était accessible en ligne via une URL publique, du type :



Pour le front-end, j'ai choisi d'utiliser Vercel, une autre plateforme gratuite spécialisée dans le déploiement d'interfaces web. Après m'être connecté sur vercel.com avec mon compte GitHub, j'ai importé le dépôt contenant la partie front-end du projet.

The screenshot shows the Vercel dashboard interface. At the top, there's a navigation bar with links for "Next.js [25] tickets →", "Find...", "Feedback", and notifications. A prominent button on the right says "Collaborate on a Pro Trial".

The main area features two main sections: "Import Git Repository" and "Clone Template".

Import Git Repository: This section shows a list of repositories from the user "Inhumann". The repos listed are:

- neurona-client · Oct 1
- Post-it-client · Sep 22
- Post-it-api · Sep 22
- Inhumann · Jun 10
- to-do-list-api · Sep 19

Below this list is a link: "Import Third-Party Git Repository →".

Clone Template: This section displays several template cards:

- Next.js Boilerplate:** Shows a preview of a Next.js application.
- AI Chatbot:** Shows a preview of an AI-powered chatbot application.
- Express.js on Vercel:** Shows a preview of an Express.js application.
- Vite + React Starter:** Shows a preview of a Vite + React application.

At the bottom of the "Clone Template" section is a link: "Browse All Templates →".

Vercel détecte automatiquement le framework utilisé, (React, Vue, etc.), ce qui facilite grandement la mise en ligne. Dans mon cas, il s'agissait d'un projet React. J'ai donc laissé la commande de build par défaut : `npm run build`

New Project

Importing from GitHub
Inhumannn/Post-it-client ⚡ main

Choose where you want to create the project and give it a name.

Vercel Team Project Name

inhuman-projects Hobby post-it-client

Framework Preset

Vite

Root Directory

./ Edit

Build and Output Settings

Environment Variables

Deploy

The screenshot shows the Vercel interface for creating a new project. At the top, it says "New Project" and "Importing from GitHub" with a link to "Inhumannn/Post-it-client". Below that, there are fields for "Vercel Team" (set to "inhuman-projects" with a "Hobby" plan) and "Project Name" ("post-it-client"). Under "Framework Preset", "Vite" is selected. The "Root Directory" is set to "./". There are two expandable sections: "Build and Output Settings" and "Environment Variables". At the bottom is a large "Deploy" button.

Quelques secondes après le déploiement, Vercel à généré une URL publique, du type :

todo-list-frontend.vercel.app

The screenshot shows a browser address bar with the URL "todo-list-frontend.vercel.app". To the right of the address bar are three icons: a monitor, a GitHub logo, and a Vercel logo.

me permettant d'accéder directement à l'application depuis n'importe quel navigateur.

Une fois les deux parties en ligne, j'ai relié le front-end au back-end en modifiant dans mon code le fetch qui appelait mon serveur en local, pour y insérer le lien du serveur hébergé sur render :

```
fetch("https://todo-list-api.onrender.com/tasks")
  .then(res => res.json())
  .then(data => console.log(data));
```

Enfin grâce à cette liaison, l'application pouvait désormais communiquer avec son serveur distant pour afficher, ajouter ou supprimer des tâches en ligne.

2. Précisez les moyens utilisés :

Pour documenter le déploiement d'une application dynamique web ou web mobile, j'ai utilisé mes connaissances données lors des cours en formation et les documentations.

3. Avec qui avez-vous travaillé ?

J'ai réalisé ce travail individuellement.

4. Contexte

Nom de l'entreprise, organisme ou

Centre de formation la plateforme à Brignole

association ▾

Chantier, atelier, service Formation Développeur Web & Web Mobile

Période d'exercice Du au

5. Informations complémentaires (*facultatif*)**Titres, diplômes, CQP, attestations de formation***(facultatif)*

Intitulé	Autorité ou organisme	Date
Diplôme nationale du brevet	Collège Marcel Pagnol à Pertuis	2021
Baccalauréat des Sciences et Technologies de l'Industrie et du Développement Durable	Lycée Vauvenargues à Aix-en-Provence	Juillet 2024

Déclaration sur l'honneur

Je soussigné(e) Thomas PENA-BERMOND ,

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 le
pour faire valoir ce que de droit.

Signature :

Documents illustrant la pratique professionnelle

(facultatif)

Annexes

(Si le RC le prévoit)