Afin de commencer le projet et ma partie frontend je vais installer Node.js et npm car  **Node.js** est une plateforme qui permet d'exécuter JavaScript en dehors du navigateur.

 **npm** (Node Package Manager) est un gestionnaire de paquets pour installer les dépendances dans ton projet.

J’ai donc installé Node.js depuis le site officiel : <https://nodejs.org/fr>.

Pour vérifier qu’il était bien installé j’ai fais : node -v et npm -v afin de connaitre leur version.

Apres avoir installé j’ai taper la commande dans le CMD ‘npm install -g @vue/cli’ afin de créer un projet Vue.js avec une configuration prête à l'emploi, tout en te permettant de personnaliser certaines options selon mes besoin. «  Cette commande installe Vue CLI globalement sur ta machine, ce qui te permet de l’utiliser depuis n’importe quel répertoire »

« vue –version » pour connaitre la version téléchargé

« vue create interface\_camping » Cette commande crée un nouveau projet Vue.js dans un dossier appelé interface\_camping. Lors de l’exécution de cette commande, Vue CLI me demandera de choisir une préconfiguration pour mon projet. 3 Confuguration sont disponible : Default (VUE3 babel, eslint) ou(VUE3 babel, eslint) et pour finir une configratiuon manuel ou je choisi moi-même les configuration que je veux.

J’ai donc choisi manuellement avec les paramètres suivants : Babel, PWA support, Routeur, Linter/ Formatter. Et je pense utiliser dans le futur « Unit Testing et E2E Testing » Version 3.

Use History mode for router ? Yes

ESLint+ prettier / Lint on save / In dedicated config files

Pour le versionning j’ai installé git via le site <https://git-scm.com/downloads/win>. J’ai autorisé tt les interface de commande avec le path. Je suis allé sur git bash puis sur le dossier du projet et j’ai fait git init git init transforme un répertoire en un dépôt Git, permettant à Git de suivre les modifications des fichiers dans ce répertoire.

Cela crée un sous-répertoire caché nommé .git qui contient tous les fichiers nécessaires pour le dépôt Git.

J’ai configuré le fichier .gitignore qui est le fichier des fichier a ignoré pour le versionning.

Par la suite j’ai utilisé la commande git add . Cela ajoute les fichiers du projet qui ne sont pas dans le .gitignore au suivi de Git.

git commit -m "Premier commit avec .gitignore" Un commit permet de valider les modifications que tu viens d'apporter. Pour faire un commit avec un message

Pour Ajouter un dépôt distant j’ai utilisé la commande git « remote add origin https://github.com/leeeooooooooo/projet\_camping.git »

Et pour pousser le le commit fait sur le dépôt en ligne j’utilise git push -u origin master

$ git config --global user.name "Léo"

git config --global user.email [leoo.sicot@gmail.com](mailto:leoo.sicot@gmail.com)

ghp\_6cMt7ex21fCg7lxnJbbyjpGq7LW9Lq1sTxq9

J’ai voulu créer sur le github des branches pour chaque page que je créer. J’ai donc créer une branche page\_connexion\_token afin de mettre les nouvelles fonctionnalités.

Les commandes utilisés sont :

git checkout -b page-connexion-token pour créer la branche

git add src/views/connexion.vue pour sélectionner le fichier connexion.vue

git commit -m "Création de la page de connexion avec token" pour écrire un commentaire

git push origin page-connexion-token pour l’envoyer sur le dépôt

LES COMMANDES AFIN D UTILISER GIT :

git status : Vérifie l'état actuel du dépôt.

git add <fichier> : Ajoute un fichier spécifique.

git commit -m "message" : Crée un commit avec un message.

git log : Affiche l'historique des commits.

git checkout -b <branche> : Crée et passe à une nouvelle branche.

git checkout <branche> : Passe d'une branche à une autre.

git push origin <branche> : Pousse les changements vers GitHub.

git pull origin <branche> : Récupère les derniers changements depuis GitHub.

git merge <branche> : Fusionne une branche dans la branche actuelle.

git branch -d <branche> : Supprime une branche locale après fusion.

Problème de formatage j’installe donc npm install --save-dev --save-exact prettier.

Je créer le code dans .prettierrc et je fais la commande npx prettier --write src/views/connexion.vue

Problème afin de mettre mon code d’avant, axios pas installé : npm install axios

npx prettier --write . pour formater tout le fichier

git credential reject https://github.com

git credential approve https://github.com

Puis, exécute cette commande pour activer le stockage des identifiants :

bash

Copier

Modifier

git config --global credential.helper store

npm install crypto-js desinstaller

Prépare ton projet pour la production npm run build