1. Nous allons utiliser l’intégration de Git sur VS Code pour gérer notre projet avec Git. Le but est d'initialiser le repository Git en local, de créer le repository distant, et de faire des allers-retours entre le local et le distant pour avoir une copie de notre repository sur GitHub.
2. Création du dossier EcoRide dans htdocs
3. Ouverture du dossier dans vs code
4. Création du fichier readme.md dans vs.code
5. Le petit bohnomme en bas dans vs code est mon compte sur github puis l’icone source controle en haut (initialize pour soit initialiser un repository en local soit ou publish to github pour initialiser un repo en local et le push sur github
6. Cliquer sur le deuxième et choisir un repo public parce que le deuxième tout le monde peut y avoir accès mais pour le premier il faut avoir des droits avant d’accéder au code source
7. Github fait ensuite mon premier commit
8. Mon projet est désormais initialisé en local et branché à une remote sur github
9. Création de mon index.php

Création de mon dossier Router et à l’interieur le fichier route.js puis copier et coller le code :export default class Route {

constructor(url, title, pathHtml, pathJS = "") {

this.url = url;

this.title = title;

this.pathHtml = pathHtml;

this.pathJS = pathJS;

}

}

Ensuite je crée le fichier allRoutes.js dans le dossier route toujours et copier coller ce code:import Route from "/Route.js";

//Définir ici vos routes

export const allRoutes = [

new Route("/", "Accueil", "/pages/home.html"),];

//Le titre s'affiche comme ceci : Route.titre - websitename

export const websiteName = "EcoRide";

1. Création du dosssier pages et à l’intérieur le fichie home.php

Création du fichier router.js qui contiendra toute ma logique de routage puis copier coller :import Route from "./Route.js";

import { allRoutes, websiteName } from "./allRoutes.js";

// Création d'une route pour la page 404 (page introuvable)

const route404 = new Route("404", "Page introuvable", "/pages/404.html");

// Fonction pour récupérer la route correspondant à une URL donnée

const getRouteByUrl = (url) => {

let currentRoute = null;

// Parcours de toutes les routes pour trouver la correspondance

allRoutes.forEach((element) => {

if (element.url == url) {

currentRoute = element;

}

});

// Si aucune correspondance n'est trouvée, on retourne la route 404

if (currentRoute != null) {

return currentRoute;

} else {

return route404;

}

};

// Fonction pour charger le contenu de la page

const LoadContentPage = async () => {

const path = window.location.pathname;

// Récupération de l'URL actuelle

const actualRoute = getRouteByUrl(path);

// Récupération du contenu HTML de la route

const html = await fetch(actualRoute.pathHtml).then((data) => data.text());

// Ajout du contenu HTML à l'élément avec l'ID "main-page"

document.getElementById("main-page").innerHTML = html;

// Ajout du contenu JavaScript

if (actualRoute.pathJS != "") {

// Création d'une balise script

var scriptTag = document.createElement("script");

scriptTag.setAttribute("type", "text/javascript");

scriptTag.setAttribute("src", actualRoute.pathJS);

// Ajout de la balise script au corps du document

document.querySelector("body").appendChild(scriptTag);

}

// Changement du titre de la page

document.title = actualRoute.title + " - " + websiteName;

};

// Fonction pour gérer les événements de routage (clic sur les liens)

const routeEvent = (event) => {

event = event || window.event;

event.preventDefault();

// Mise à jour de l'URL dans l'historique du navigateur

window.history.pushState({}, "", event.target.href);

// Chargement du contenu de la nouvelle page

LoadContentPage();

};

// Gestion de l'événement de retour en arrière dans l'historique du navigateur

window.onpopstate = LoadContentPage;

// Assignation de la fonction routeEvent à la propriété route de la fenêtre

window.route = routeEvent;

// Chargement du contenu de la page au chargement initial

LoadContentPage();

1. Ensuite aller dans mon fichier form.html tout en bas pour taper <script type= ‘module’ src=’Router/router.js</script> afin de lier mon fichier de logique de routage
2. J’ai ensuite créer un fichier 404.htm dans le dossier page pour taper :

<div>

    <h1>404 - Page introuvable</h1>

    <h4>Oups..! Il semble que la page que vous recherchez n'existe pas ou a été déplacée.</h4>

    <p>Retour à la <a href="/">page d'accueil</a>.</p>

</div>

1. J’ai cliqué sur l’icone source controle puis choisi commit & sync car je suis sûr de moi et travaille seul
2. J’ai installé le npm de bootstrap : dans view->terminal taper npm puis npm install bootstrap
3. Puis le téléchargement de du dossier node modules et des fichiers package.lock et package.json ont été faits.
4. Ensuite création du fichier .gitignore puis dans ce fichier j’ai ecrit node\_modules afin que gitignore ignore mon dossier node\_modules