Application de gestion de la M2L – Documentation technique

Maison des Ligues

Sommaire :

I) Langages utilisés :

JavaScript :

- Créer les différents composants du site pour définir sa structure
- Communiquer avec l'API REST afin de récupérer et d'ajouter des informations dans la base de données du site

MySQL :

- Créer la base de données du site

NodeJs :

- Créer l'API REST et permettre la liaison entre le front et le back

II) Mise en place du projet :

A) Installation du front/back :

Pour l'installation du front end, nous importons dans chaque composant du projet les modules externes nécessaires.

```
1  import { useParams, useNavigate } from "react-router-dom";
2  import { useEffect, useState } from 'react';
3  import { useForm } from "react-hook-form";
4  import { Link } from 'react-router-dom'
5  import axios from 'axios';
```

- useParams et useNavigate permettent respectivement d'extraire les paramètre de l'URL dans un composant de l'application et de naviguer vers une autre URL

- useEffect et useState permettent d'exécuter du code en réponse à un changement d'état ou de propriété d'un composant et de déclarer et mettre à jour des variables d'état

- useForm facilite la gestion des formulaires en React

- Link permet de créer des liens de navigation au sein de l'application

- Axios va permettre d'effectuer les requêtes http directement à l'API

Il faut bien penser à installer tous ces modules au préalable via un terminal.

Pour l'installation du back end il faut commencer par la création de l'API qui communiquera entre l'application et la base de données :

```
const express = require('express') // la récupération d'express
const app = express() // variable utilisant la librairie express
let cors = require('cors')

require('dotenv').config()

// connexion à la bdd
const mariadb = require('mariadb');

const pool = mariadb.createPool({
    host: process.env.DB_HOST,
    database: process.env.DB_DTB,
    user: process.env.DB_USER,
    password: process.env.DB_PWD,
});

app.use(express.json())
app.use(cors())

//Début de l'api côté produit

app.get('/produit', async(req,res) => { // affichage des produits
    let conn;
    try{
        console.log("lancement de la connexion")
        conn = await pool.getConnection();
        console.log("lancement de la requete")
        const rows= await conn.query('SELECT * FROM produit');
        console.log(rows);
        res.status(200).json(rows)
    }
    catch(err){
        console.log(err);
    }
})
```

Dans l'exemple ci-dessus, on retrouve la constante « pool » qui permet à l'aide de variable de se connecter à la base de données. Les variables étant définit dans le fichier .env dans lequel se trouve les informations de connexion de la base.

On y retrouve aussi un exemple de requête (ici un get) qui sera plus tard utilisée dans les composants de l'application.

B) Liaison avec l'API :

Une fois le front end et le back end installés, il faut maintenant les liés afin que l'application fonctionne comme prévu.

Pour cela, nous allons utiliser le module axios, installé plus tôt dans la création du front end.

```
const recup = async () => {
  await axios.get(`http://localhost:8000/produit`)
      .then(res => {
          console.log(res)
          setProduits(res.data)
          setAffichage(true)
      })
```

Ce module va permettre d'envoyer une requête sur le serveur où l'on retrouve le back end.