DIANZENZA GLOIRE

Présentation Projet Backend

Dans le cadre d’un contexte d’une entreprise dont les employés ont perdu la motivation, cette entreprise (fictionnelle) a pris la décision de faire développer un site web permettant aux différents membres de communiquer entre eux via des posts pouvant contenir du texte et une image sous format gif.

Le but de ce projet et de développer les compétences en backend (notamment Express et Sequelize) et optionnellement frontend (Vuejs/React et Tailwind/Bootstrap). Pour cela, nous verrons les étapes de configuration, définition et enfin mise en place du code.

La partie backend, la plus importante, sera portée sur l’utilisation d’une API customisée qui permet d’interagir avec la base de données en partant du principe CRUD (Create, Read, Update, Delete), les 4 principales opérations sur une base de données et en se connectant avec localhost. Pour récupérer les tables sur Javascript, on utilisera Sequelize, un ORM (Object-relating mappin, un logiciel qui facilite l’interaction entre une application et une base de données relationnelle) utilisable facilement sur Javascript.

La partie frontend sera gérée avec Vuejs (que j’ai échoué à exploiter, j’aurais aimé utilisé à la place React). Ces deux bibliothèques utilisent le langage Javascript pour mieux interagir avec le backend. Contrairement aux fichiers HTML statiques, ils sont basés sur des composants donc ils gèrent les différents états et composants et mettent à jour les éléments qui changent sans impacter tout le site.

**Initialiser le projet**

Avant tout, la première chose à faire est d’ouvrir Visual Studio Code, l’un des nombreux logiciels d’édition de codes populaires. Une fois qu’on a ouvert l’application, on ouvre le dossier dans lequel on travaillera avant d’ouvrir le terminal pour initialiser en premier lieu le fichier package.json grâce à npm init. On devra ensuite rentrer des informations complémentaires telles que le nom du projet ou le code de test. Une fois que les fichiers sont créés, on commence le processus pour initialiser React ou Vuejs.

Que ce soit React ou Vue, le meilleur moyen de l’initialiser dans le dossier est d’utiliser npm ou yarn. Ce sont des module d’importation de packages. Ils sont très utiles pour faire importer n’importe quel package, comme express ou mysql.

Maintenant que le framework frontend est installé, on peut se reconcentrer sur le back-end.

Pour continuer l’initialisation, on commence par créer plusieurs fichiers JS : init, app at server. Init servira à importer les variables sequelize et DataTypes, indispensables pour communiquer avec la base de données. Ces deux variables proviennent du package Sequelize, qui gère l’interaction avec les bases de données auxquelles on est connectées. Dans init, on va également initialiser la connexion à la base de donnée voulue grâce à new Sequelize(avec notamment en paramètres hôte,bdd,mot de passe…).

Ensuite dans le fichier app, on importe les modules nécessaires (notamment express), ensuite on crée la variable app à partir de la fonction express. Cela permettra d’initialiser les routeurs et l’entête (tel que Access-Control-Allow-Origin…). On doit ensuite initialiser la fonction express.json() en l’utilisant comme paramètre de app.use(). Sa fonction est de gérer les données reçues par req.body pour le convertir en JSON et envoyer la réponse programmée dans le backend. Nous utiliserons aussi la fonction cors du module éponyme dans le but de gérer les autorisations CORS qui pourraient sinon bloquer l’appel de l’API.

Après nous pourrons importer la variable sequelize du fichier init pour l’utiliser plus bas. Ainsi nous créons la fonction asynchrone connect pour se connecter à la base de données inscrite avec le mot de passe enregistré. Si l’opération réussit, nous affichons un message justifiant le succès de la connexion. Dans ce cas, après nous utilisons la fonction sync pour synchroniser toutes les tables de la base de données en temps réel. Pour finir, on exporte la variable app pour l’utiliser dans server.

Dans le fichier server, on importe la variable app avant de configurer le package dotenv (permettant de gérer les variables dans le fichier .env) et d’initialiser un port (soit en utilisant process.env.port soit en définissant un port). Process.env.port permet d’utiliser le numéro de port enregistré dans le fichier .env. On importe le package http pour créer le server avec comme paramètre la variable app. On stocke cette fonction dans une variable server d’où on usera de la fonction on avec le paramètre listening pour démarrer une fonction lorsque le serveur est lu correctement. La fonction address de server retourne soit une chaîne de caractère qu’on utilisera comme pipe soit un object dans lequel cas on utilisera seulement le port. Lorsque tout cela fonctionnera, on demandera au serveur de lire le port indiqué. S’il est disponible, l’utilisateur sera immédiatement connecté à localhost :numéro de port.

**Créer une base de données**

Pour créer la base de données, j’ai utilisé le langage SQL que j’exploite avec l’application MySQL Workbench, qui contient une interface facile à comprendre. Je me connecte au localhost avec le port 3306 avec un mot de passe et j’accède à un menu où je peux voir toutes les bases de données qui existent dans l’appareil. Pour gérer les tables dans l’application, je dois écrire des codes SQL. Le premier consiste à créer la base de données : *CREATE DATABASE finalproject.* Attention : pour pouvoir interagir directement avec les tables, on rajoute un autre code : *USE finalproject*, sinon le code ne démarre pas.

Je créerai la table des utilisateurs, posts et commentaires. Pour cela j’utilise la commande *CREATE TABLE* pour cela ; chacune sera initialisée avec leurs propres attributs avec leurs paramètres : pour la table d’utilisateurs, en plus d’ajouter un id pour toutes les tables, on y intègrera l’adresse email, le mot de passe et admin qui indique si l’utilisateur est un administrateur ou non. L’id (concernant toutes les tables sans exception)