Objectif : Créer une base de données et une api coté serveur avec node.js

Création du conteneur ubuntu

Créer un conteneur sur le serveur avec l'adresse IP : 10.20.222.1XX
Installer sur ce conteneur : mysql et créer une base de données : bddjs
apt update
apt install mysql-server

Créer la base de données

Créer une base de données 'bddjs'
Créer une table 'utilisateurs' avec les champs 'nom', 'email' et 'age'
Créer un utilisateur ciel avec le mot de passe ciel qui a tous les droits sur cette base.

```sql
CREATE DATABASE bddjs CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
USE bddjs;

CREATE TABLE utilisateurs (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nom VARCHAR(255),
    email VARCHAR(255),
    age INT
);

CREATE USER 'ciel'@'localhost' IDENTIFIED BY 'ciel';
GRANT ALL PRIVILEGES ON bddjs.* TO 'ciel'@'localhost';
FLUSH PRIVILEGES;
```

Créer une page web coté client

Créer une page index.html qui permet de saisir les champs à rentrer (Nom, email et age) et un bouton qui appellera un fichier javascript : script.js

```html
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Formulaire Utilisateur</title>
</head>
<body>
    <h2>Formulaire d'inscription</h2>
    <form id="userForm">
        <label for="nom">Nom:</label>
        <input type="text" id="nom" name="nom" required><br><br>

        <label for="email">Email:</label>
        <input type="email" id="email" name="email" required><br><br>

        <label for="age">Âge:</label>
        <input type="number" id="age" name="age" required><br><br>

        <button type="button" onclick="submitForm()">Soumettre</button>
    </form>
```

```
    <script src="script.js"></script>
</body>
</html>
```

Créer le script script.js qui permet d'envoyer les données vers le serveur nodejs en utilisant la fonction fetch.

```
function submitForm() {
    const nom = document.getElementById('nom').value;
    const email = document.getElementById('email').value;
    const age = document.getElementById('age').value;

    fetch('http://10.20.222.1XX:3000/api/utilisateurs', {
        method: 'POST',
        headers: {
            'Content-Type': 'application/json'
        },
        body: JSON.stringify({ nom, email, age })
    })
    .then(response => response.json())
    .then(data => {
        alert('Utilisateur ajouté avec succès!');
    })
    .catch(error => {
        console.error('Erreur:', error);
    });
}
```

Paramétrage du serveur

Créer un répertoire

mkdir projet_mysql

Installer nodejs et npm sur le serveur :

apt install nodejs npm

Entrer dans ce répertoire et initialiser le projet Node.js avec la commande suivante :

npm init -y

Installer les frameworks Express.js, mysql2 et body-parser.

npm install -g express mysql2 body-parser cors

Créer un nouveau fichier appelé app.js dans le dossier de projet.

```
const express = require('express');
const mysql = require('mysql2');
const bodyParser = require('body-parser');
const cors = require('cors'); // Importer le package CORS

const app = express();

app.use(cors()); // Activer CORS pour toutes les routes
app.use(bodyParser.json()); // Pour traiter les données JSON du corps des requêtes
```

```
// Configuration de la connexion à la base de données
const db = mysql.createConnection({
    host: 'localhost',
    user: 'ciel',
    password: 'ciel',
    database: 'bddjs'
});

db.connect((err) => {
    if (err) {
        console.error('Erreur de connexion à la base de donnees:', err);
    } else {
        console.log('Connecte a la base de donnees MySQL');
    }
});

// Route POST pour ajouter un utilisateur
app.post('/api/utilisateurs', (req, res) => {
    const { nom, email, age } = req.body;

    if (!nom || !email || !age) {
        return res.status(400).send({ error: 'Tous les champs sont obligatoires : nom, email,
age' });
    }

    const sql = 'INSERT INTO utilisateurs (nom, email, age) VALUES (?, ?, ?)';
    db.query(sql, [nom, email, age], (err, result) => {
        if (err) {
            console.error('Erreur lors de l\'ajout de l\'utilisateur:', err);
            res.status(500).send({ error: 'Erreur serveur' });
        } else {
            res.status(201).send({ message: 'Utilisateur ajouté avec succès!', id:
result.insertId });
        }
    });
});

// Route GET pour récupérer tous les utilisateurs
app.get('/api/utilisateurs', (req, res) => {
    const sql = 'SELECT * FROM utilisateurs';
    db.query(sql, (err, results) => {
        if (err) {
            console.error('Erreur lors de la recuperation des utilisateurs:', err);
            res.status(500).send({ error: 'Erreur serveur' });
        } else {
            res.json(results); // Renvoie les utilisateurs sous forme de JSON
        }
    });
});

// Démarrer le serveur
app.listen(3000, () => {
    console.log('Serveur démarré sur http://localhost:3000');
});
```

Exécuter l'application avec la commande suivante :

node app.js

Test de l'enregistrement

Lancer la page web et enregistrer une nouvelle donnée dans la table.
Vérifier sur la table si cela fonctionne.

Affichage de la base de données

Créer maintenant une API dans le script app.js qui permet de lire toute la table.

```
// API pour récupérer les utilisateurs
app.get('/api/utilisateurs', (req, res) => {
    const sql = 'SELECT * FROM utilisateurs';
    db.query(sql, (err, results) => {
        if (err) {
            console.error('Erreur lors de la récupération des utilisateurs :', err);
            res.status(500).send({ error: 'Erreur serveur' });
        } else {
            res.json(results); // Envoie les utilisateurs en format JSON
        }
    });
});
```

Créer une page web sur le client qui permet d'afficher toutes les lignes de la base de données.

```
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Liste des Utilisateurs</title>
</head>
<body>
    <h2>Liste des Utilisateurs</h2>
    <table border="1">
        <thead>
            <tr>
                <th>ID</th>
                <th>Nom</th>
                <th>Email</th>
                <th>Âge</th>
            </tr>
        </thead>
        <tbody id="userTableBody">

        </tbody>
    </table>

    <script>
        async function fetchUsers() {
            try {
                const response = await fetch('http://localhost:3000/api/utilisateurs');
                const users = await response.json();

                const tableBody = document.getElementById('userTableBody');
                tableBody.innerHTML = ''; // Efface le contenu actuel
```

```
                users.forEach(user => {
                    const row = document.createElement('tr');
                    row.innerHTML = `
                        <td>${user.id}</td>
                        <td>${user.nom}</td>
                        <td>${user.email}</td>
                        <td>${user.age}</td>
                    `;
                    tableBody.appendChild(row);
                });
            } catch (error) {
                console.error('Erreur lors de la récupération des utilisateurs :', error);
            }
        }

        // Charger les utilisateurs au chargement de la page
        window.onload = fetchUsers;
    </script>
</body>
</html>
```