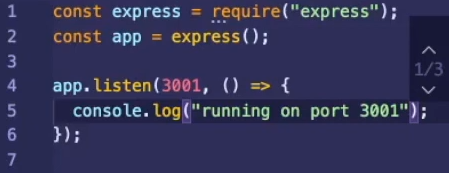
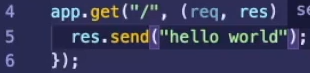
**Client-side :**

**Création de l’architecture front-end :**

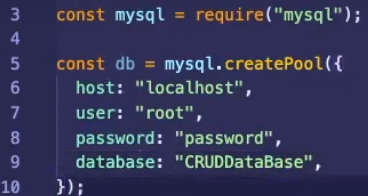
1. Créer deux dossiers : un client et un serveur (pour le frontend et le backend)
2. Dans le dossier client lancer la commande ;
   1. -npx create-react-app “nom\_du\_dossier”
3. Garder juste App.js App.css et index.js
4. Créer des inputs et un button pour submit les inputs dans la fonction App()
5. Import React, {useState, useEffect} from ‘react’;

**Server-side :**

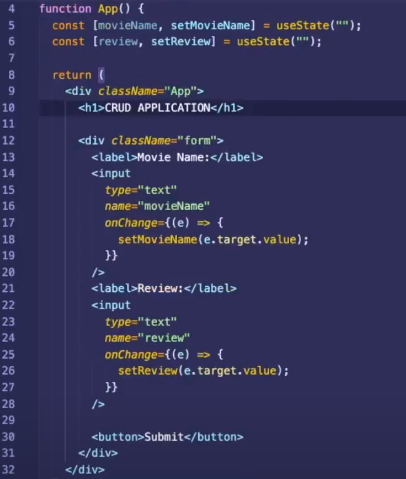
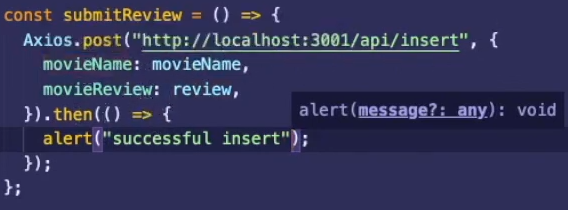
**Création du serveur :**

1. Dans le dossier serveur : -npm init(toutes les dépendances)
2. Créer un fichier index.js (fichier du serveur)
3. Installer express
4. Installer body-parser
5. Installer mysql
6. Installer nodemon
7. Dans index.js :
   1. 
8. Ajouter une route pour l’index :
   1. 
9. Ajouter 2 scripts dans package.json :
   1. 

**Création de la base de données MySQL :**

1. Peut s’utiliser avec MySQLWorkbench pour visualiser les tables
2. Créer une table dans la DB avec autant de colonnes que d’inputs dans App.js (coté client)
3. Pour accéder à la DB (paramètres par défaut):
   1. 
4. Modifier app.get() pour accéder à la BD en lui ajoutant ;
   1. -const sqlInsert = “COMMANDE SQL”
   2. -db.query(sqlInsert, (err, result) => )

**Client-side :**

1. Retourner dans App.js et créer un useState dans App();
   1. 
2. Dans le dossier ‘client’ ;
   1. -npm install axios
   2. Import Axios from ‘axios’ dans App.js
3. Dans App() ajouter ;
   1. -const submitReview = ()
   2. <button onClick={submitReview}>Submit</button> (modifier button déjà existant)
4. Dans l’index.js du server, ajouter une route pour l’API pour le POST;
   1. 
   2. Les “?” représentent des variable dans db.query() à déterminer plus tard
5. Dans App.js du côté client, modifier la fonction submitReview() ;
   1. 
   2. Le “then” est une promesse qui s’exécute une fois le code précédent exécuté

**Server-side :**

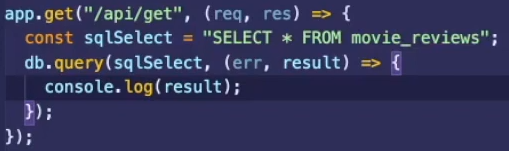
**Installation des services :**

1. Ajouter dans index.js const bodyParser = require(‘body-parser’);
2. Ajouter une ligne de code ;
   1. 
3. Dans le app.post rajouter des constantes qui vont traiter les variables envoyées par React ;
   1. 
4. Dans le dossier server ;
   1. -npm install cors (éviter les erreur)

b. -const cors = require(‘cors’);

* 1. -app.use(cors());
  2. -app.use(express.json()); (pour traiter le JSON du frontend)

**Afficher les données de la DB :**

1. Ajouter API pour recevoir des données app.get() ;
   1. 
   2. Affiche uniquement les résultats DB en console lorsqu’on accède à l’url spécifié
2. On peut modifier la fonction db.query() avec ;
   1. 
   2. Permet d’afficher sur la page, le JSON renvoyé par la fonction