Rapport de Projet

Gestion d'Utilisateurs - Application Fullstack avec Docker et CI/CD
Développé par : Majda Razzouk

1. Introduction

Ce rapport présente le développement d'une application web **fullstack** permettant la **gestion des utilisateurs**. Ce projet est conçu avec les technologies modernes du web : **React** pour le frontend, **Express.js** pour le backend, **MySQL** comme base de données, et **Docker** pour la conteneurisation. Il inclut également une **intégration continue et un déploiement continu (CI/CD)** via **GitHub Actions**.

Ce projet a été réalisé dans le cadre d'un apprentissage avancé du développement web, du DevOps et des bonnes pratiques de développement logiciel.

2. Objectifs du Projet

Le projet vise à créer une solution complète pour la gestion d'utilisateurs, avec les objectifs suivants :

- Concevoir une interface utilisateur moderne et responsive.
- Mettre en place un backend RESTful sécurisé et validé.
- Utiliser Docker pour assurer la portabilité de l'application.
- Mettre en œuvre un pipeline CI/CD pour automatiser les tests et le déploiement.
- Assurer la communication entre les différents services via Docker Compose.

3. Architecture Technique

L'application est divisée en plusieurs couches :

A. Backend (Express.js & Node.js)

- Framework : Express.js avec Node.js
- Base de données relationnelle : MySQL
- ORM : Sequelize
- Validation : Joi pour vérifier la structure des données entrantes
- Fonctionnalités :
    - CRUD complet des utilisateurs
    - Gestion des erreurs
    - API REST sécurisée et maintenable

B. Frontend (React.js)

- Bibliothèque : React 18
- Design : Material-UI
- Communication avec l'API : Axios
- Navigation : React Router
- Fonctionnalités :
    - Affichage des utilisateurs
    - Ajout / modification / suppression
    - Recherche, pagination, notifications toast

C. DevOps (Docker & GitHub Actions)

- Docker pour créer des conteneurs isolés (frontend, backend, base de données)
- Docker Compose pour orchestrer tous les services
- GitHub Actions pour :
    - Lancer les tests automatiquement
    - Construire les images Docker

- Déployer en cas de réussite des tests

4. Déploiement et Utilisation

Prérequis

- Docker ≥ 20.10
- Docker Compose ≥ 2.5
- Node.js ≥ 18

Installation

git clone https://github.com/majda/gestion-utilisateurs.git
cd gestion-utilisateurs
docker-compose up --build

Structure Docker

- **frontend : application React servie par Nginx**
- **backend : API Node.js/Express**
- **db : conteneur MySQL avec volume persistant**
- **nginx : serveur HTTP pour servir le frontend en production**

CI/CD

Un fichier `ci.yml` dans `.github/workflows` exécute automatiquement :

1. **L'installation des dépendances**
2. **Les tests unitaires (Mocha + Chai)**
3. **Le build des images Docker**

4. **Le déploiement si tous les tests réussissent**

5. Documentation API

Voici les principales routes RESTful disponibles :

Méthode	Endpoint	Description
GET	/api/users	Liste tous les utilisateurs
GET	/api/users/:id	Retourne un utilisateur spécifique
POST	/api/users	Crée un nouvel utilisateur
PUT	/api/users/:id	Modifie un utilisateur
DELETE	/api/users/:id	Supprime un utilisateur