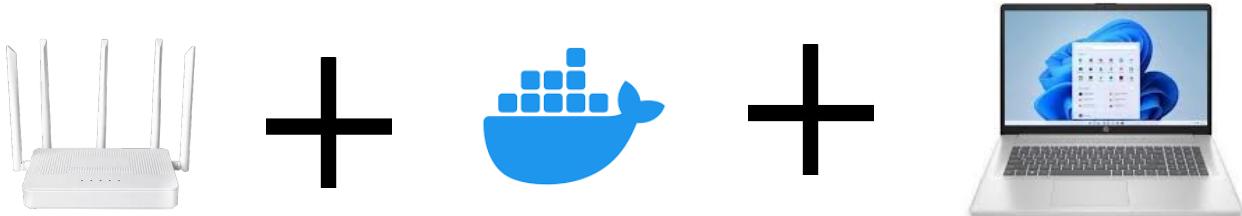


# Système local sécurisé pour examens pratiques



## Description générale :

Ce projet propose une solution **locale, sécurisée et anti-triche** pour la gestion des examens pratiques en informatique (PHP, Python, etc.). Les étudiants travaillent sur leurs **éditeurs de code habituels** (PyCharm, VSCode) tout en étant **supervisés en temps réel** par le professeur.

L'application est **déployée sur le poste du professeur via Docker** et crée un **réseau Wi-Fi local sécurisé**, sans accès Internet, pour les étudiants. Chaque étudiant se connecte via un navigateur pour :

- **Maintenir la session active,**
- **Télécharger le sujet de l'examen,**
- **Soumettre ses fichiers.**

Le professeur dispose d'un **tableau de bord en temps réel** qui permet :

- de voir les étudiants connectés avec leurs informations (nom, IP/MAC, statut),
- de recevoir les fichiers soumis en direct,
- de **déetecter les déconnexions, comportements suspects ou tentatives d'utiliser des outils d'IA,**
- d'ajouter ou retirer des étudiants de la session.

Cette solution garantit :

- **Un environnement d'examen sécurisé et isolé d'Internet**, empêchant la triche et l'usage d'outils d'IA,
- **La supervision en temps réel des étudiants**, avec alertes sur tout comportement suspect,
- **La gestion centralisée des examens et des soumissions,**
- **Un déploiement simple et portable grâce à Docker.**

## Backend / Serveur

**Objectif :** gérer les connexions des étudiants, WebSocket en temps réel, sauvegarde des fichiers, authentification.

- **Langages / Frameworks possibles :**

- **Node.js + Express + Socket.io** → idéal pour le temps réel (WebSocket) et facile à dockeriser.

## Base de données

- **SQLite** → simple, léger, suffisant pour un prototype et stockage local.
- **MongoDB** → si tu veux gérer des structures dynamiques (JSON, listes de fichiers, historique).

## Frontend / Interface web

- **Technologies :** HTML, CSS, JavaScript
- **Frameworks optionnels :**
  - **React** → composantisation, réactivité en temps réel
  - **Vue.js** → léger et facile à apprendre

## Temps réel / WebSocket

- **Socket.io (Node.js) ou WebSocket natif** → pour :
  - Suivi en direct des étudiants connectés
  - Alertes en cas de déconnexion ou comportement suspect
  - Notification des fichiers soumis

## Docker

- Conteneurisation complète pour :
  - Backend + frontend
  - Base de données
  - Configuration simple sur n'importe quel PC du professeur

Déploiement rapide et portable

Pas besoin d'installer manuellement toutes les dépendances

Tout reste local