

Épreuve E5 - Conception et développement d'applications (option SLAM)

ANNEXE 7-1-B : Fiche descriptive de réalisation professionnelle (recto)

DESCRIPTION D'UNE RÉALISATION PROFESSIONNELLE		N° réalisation : 4
Nom, prénom : Juhasz Klaudia		N° candidat :
Épreuve ponctuelle <input type="checkbox"/>	Contrôle en cours de formation <input checked="" type="checkbox"/>	Date : .07 / 11 /2025
Organisation support de la réalisation professionnelle		
Intitulé de la réalisation professionnelle	MédiaStock – Application de gestion de matériel	
Période de réalisation : 03/10/2025 – 07/11/2025 Lieu : Nice		
Modalité : <input checked="" type="checkbox"/> Seul(e) <input type="checkbox"/> En équipe		
Compétences travaillées		
<input checked="" type="checkbox"/> Concevoir et développer une solution applicative <input checked="" type="checkbox"/> Assurer la maintenance corrective ou évolutive d'une solution applicative <input checked="" type="checkbox"/> Gérer les données		
Conditions de réalisation ¹ (ressources fournies, résultats attendus)		
Ressources fournies :	Résultats attendus :	
<ul style="list-style-type: none"> cahier des charges du projet MédiaStock, schéma Merise de la base de données, outils Docker et configurations préétablies (Dockerfile, docker-compose.yml), dossier backend du professeur (base de code à adapter), accès au serveur SISR mutualisé pour le déploiement.* 	<ul style="list-style-type: none"> application web fonctionnelle en local et sur serveur distant, gestion complète du matériel (CRUD + prêts/retours), génération et lecture de QR codes via smartphone , documentation technique et utilisateur complète, livraison du projet, déploiement et soutenance finale. 	
Description des ressources documentaires, matérielles et logicielles utilisées ²		
Ressources documentaires		
<ul style="list-style-type: none"> Cahier des charges du projet, Cours de développement web (PHP/MySQL/Bootstrap), Documentation officielle de PHP, PDO, Bootstrap, Docker et Flatpickr, Tutoriels sur QR code en JS ou en PHP * Exemples de code et dossiers "mediastock_backend" fournis par le professeur. 		
Ressources matérielles :		
<ul style="list-style-type: none"> Ordinateur personnel sous Windows 11, Smartphone Android pour les tests de scan QR code, Réseau local de l'école pour le déploiement serveur SISR. 		
Ressources logicielles :		
Outils de développement		
<ul style="list-style-type: none"> Git (v2.51.0) – gestion de version GitHub (v3.5.1) – hébergement du code source et collaboration VS Code (v1.105.0) – environnement de développement intégré (IDE) Figma – conception des maquettes Looping / DrawSQL (2025) – modélisation de la base de données 	Technologies Frontend <ul style="list-style-type: none"> HTML 5 – structure des pages web CSS 3 / Bootstrap 5.3.8 – mise en forme et design responsive JavaScript – interactions dynamiques côté client FontAwesome 7.0.1 – icônes graphiques Technologies Backend <ul style="list-style-type: none"> PHP 8.2 – langage côté serveur (procédural) MySQL 8.0 – gestion des données (SGBD) PhpMyAdmin 5.2.3 – interface d'administration MySQL Docker 28.1.1 – conteneurisation du projet (Apache, PHP, MySQL, PhpMyAdmin) 	
Modalités d'accès aux productions³ et à leur documentation⁴:		
<ul style="list-style-type: none"> Code source complet disponible sur GitHub : https://github.com/juklau/MediaStock/tree/klaudia/ Application accessible en local via Docker : Site : http://localhost:9080 / PhpMyAdmin : http://localhost:8082 Base de données : script « sql/init.sql » exécuté automatiquement lors du build. Documentation technique et utilisateur fournie en PDF (dans /docs). ** 		

¹ En référence aux *conditions de réalisation et ressources nécessaires* du bloc « Conception et développement d'applications » prévues dans le référentiel de certification du BTS SIO.

² Les réalisations professionnelles sont élaborées dans un environnement technologique conforme à l'annexe II.E du référentiel du BTS SIO.

³ Conformément au référentiel du BTS SIO « *Dans tous les cas, les candidats doivent se munir des outils et ressources techniques nécessaires au déroulement de l'épreuve. Ils sont seuls responsables de la disponibilité et de la mise en œuvre de ces outils et ressources. La circulaire nationale d'organisation précise les conditions matérielles de déroulement des interrogations et les pénalités à appliquer aux candidats qui ne se seraient pas munis des éléments nécessaires au déroulement de l'épreuve.* ». Les éléments peuvent être un identifiant, un mot de passe, une adresse réticulaire (URL) d'un espace de stockage et de la présentation de l'organisation du stockage.

⁴ Lien vers la documentation complète, précisant et décrivant, si cela n'a été fait au verso de la fiche, la réalisation professionnelle, par exemples service fourni par la réalisation, interfaces utilisateurs, description des classes ou de la base de données.

Épreuve E5 - Conception et développement d'applications (option SLAM)

ANNEXE 7-1-B : Fiche descriptive de réalisation professionnelle
(verso, éventuellement pages suivantes)**Descriptif de la réalisation professionnelle, y compris les productions réalisées et schémas explicatifs****Objectif du projet est de concevoir et déployer un outil de gestion d'inventaire permettant :**

- de gérer le matériel (ajout, suppression, modification, archivage),
- de gérer les prêts/retours avec suivi de statut,
- d'identifier chaque matériel par un QR code unique,
- d'interagir depuis un smartphone pour le scan, la restitution ou le prêt.

Description technique**Architecture :**

- Application PHP procédurale (backend), HTML/CSS/Bootstrap/JS (frontend).
- Base MySQL gérée par PDO (PHP Data Objects).
- Conteneurisation complète avec Docker Compose :
- Services : web, mysql, phpmyadmin.
- Persistance des données via volumes (mysql-data).
- Variables d'environnement centralisées dans .env., env.php

Fonctionnalités principales :

- CRUD complet sur le matériel.
- Génération automatique de QR codes à l'ajout.
- Scan via smartphone (redirection materiel.php?code=xx).
- Gestion des prêts et retours (dates, emprunteurs, statut).
- Archivage logique des emprunteurs et matériels.
- Sécurisation basique (login admin + session PHP).
- Interface responsive et claire avec Bootstrap

Backend (exemples) :

- config/database.php** : connexion PDO.
- src/** : contient les classes métier (ex. *Item*, *Pret*, etc.) avec leurs fonctions de requêtes vers MySQL.
- public/api/** : endpoints PHP (ex: *additem.php*, *updateitem.php*, *cloturepret.php*).
- sql/init.sql** : création et remplissage des tables (catégorie, item, formation, administrateur).

Frontend :

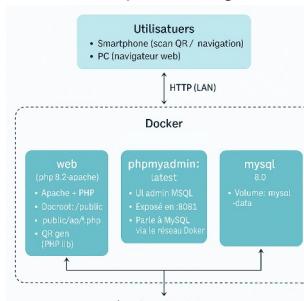
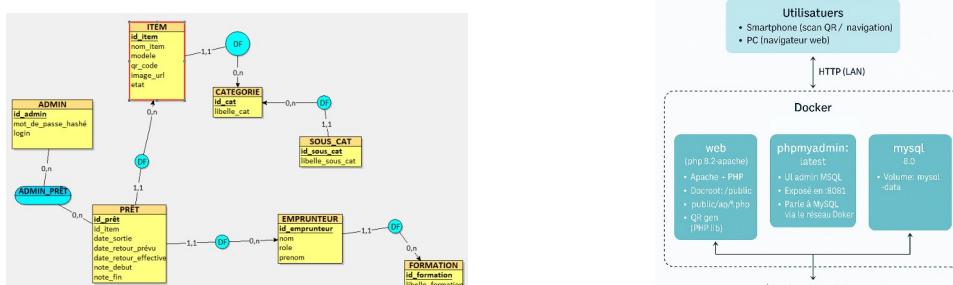
- Pages HTML dynamiques (*index.html*, *creation-pret.html*, *restitution.html*, *materiel.html*...).
- Scripts JS associés pour chaque page (gestion de formulaires, QR, modales, filtrage).
- Utilisation de Flatpickr pour les sélecteurs de dates.
- Utilisation de Bootstrap modals pour les validations.

Productions livrées

- Application web complète et testée (local + serveur).
- Scripts SQL et Docker entièrement configurés.
- Code source versionné sur GitHub.
- Documentation technique et guide utilisateur.
- Fiche descriptive de réalisation (présente).
- Captures d'écran, schéma MCD/MLD et architecture Docker. ***

Des schémas explicatifs accompagnent cette réalisation :

- un modèle conceptuel de données (MCD) illustrant les relations entre les entités ,
- un schéma d'architecture technique présentant les interactions entre les différents composants logiciels

**Représentation des flux typiques de l'application :****Consultation / Filtrage**

Client (smartphone/PC) → http://localhost:9080/frontend/index.html → appels fetch() → public/api/*.php → PDO → MySQL → réponse JSON → rendu Bootstrap.

Ajout matériel + QR

Formulaire → additem.php (retourne l'ID) → updateitem.php (qr_code = ID) → génération du QR côté client (QRCode.js) → téléchargement/impression.

Prêt / Retour

Création de prêt : addemprunteur.php (si besoin) → addpret.php → updateitem.php (mise à jour état)

Restitution : vérification prêt actif → cloturepret.php → updateitem.php (mise à jour de l'état) → confirmation.