

DESCRIPTION D'UNE RÉALISATION PROFESSIONNELLE		N° réalisation : 2
Nom, prénom : COTTET-EMARD Corentin		N° candidat :
Épreuve ponctuelle <input type="checkbox"/>	Contrôle en cours de formation <input checked="" type="checkbox"/>	Date : 18/022026 au 20/03/2026
Organisation support de la réalisation professionnelle : Projet pédagogique réalisé dans le cadre du BTS SIO SLAM à Mediaschool Toulouse, basé sur un contexte métier simulé d'une académie de formation nommée Training Médical Académie.		
Intitulé de la réalisation professionnelle :		
Période de réalisation : 11/2025 – 04/2026		Lieu : Mediaschool - TOULOUSE
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		
<u>Conditions de réalisation¹ (ressources fournies, résultats attendus) :</u>		
<p>Ce projet a été réalisé dans la continuité du projet 1, qui consistait en un backend Node.js / Express exposant une API REST sécurisée.</p> <p>L'objectif était de développer une application desktop permettant à un centre de formation de gérer ses activités de manière plus ergonomique qu'une interface web classique.</p> <p>L'application permet à différents utilisateurs (administrateur, enseignants, étudiants) de se connecter via un formulaire sécurisé. Une fois connecté, chaque utilisateur accède à un tableau de bord adapté à son rôle.</p> <p>L'administrateur peut gérer l'ensemble des données : utilisateurs, étudiants, enseignants, formations, sessions, inscriptions et notes. Il dispose également d'un tableau de bord avec des statistiques globales.</p> <p>Les enseignants peuvent consulter les formations et les sessions associées, ainsi que gérer les notes des étudiants.</p> <p>Les étudiants peuvent consulter leurs informations, leurs sessions et leurs résultats.</p> <p>L'application ne contient pas de base de données propre : elle communique avec le backend du projet 1 via une API REST sécurisée. Toutes les données sont échangées au format JSON.</p> <p>L'objectif principal est de proposer une interface desktop plus fluide, ergonomique et adaptée à un usage quotidien en entreprise.</p>		

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.

Description des ressources documentaires, matérielles et logicielles utilisées² :

Ressources documentaires :

- Documentation officielle JavaFX 21
- Documentation API REST (OpenAPI / Swagger)
- Documentation JSON (sérialisation / désérialisation)
- Documentation Java HttpClient
- Documentation Jackson ObjectMapper
- Guide d'architecture MVC (Model-View-Controller)
- Spécifications fonctionnelles du centre de formation

Ressources logicielles :

- Langage : Java
- Framework : JavaFX 21
- Backend : Node.js / Express (API REST du projet 1)
- Communication HTTP : Java HttpClient (java.net.http)
- Gestion JSON : Jackson ObjectMapper
- Base de données : MySQL / MariaDB (via backend uniquement)
- IDE : Visual Studio Code
- Outil de test : Postman
- Gestion de version : Git / GitHub

Ressources matérielles :

- Ordinateur personnel
- Environnement de développement local

Modalités d'accès aux productions³ et à leur documentation⁴ :

Le projet est accessible en local.

Accès au code source :

Base de données :

- Script SQL fourni avec le projet

Installation :

1. Cloner les deux projets (backend + Java FX)
2. Installer les dépendances backend :
 - npm install
3. Importer la base de données
4. Configurer le fichier config/db.js
5. Lancer le backend :
 - npm start ou npm run dev
6. Lancer l'application JavaFX depuis l'IDE

Accès :

- API REST : <https://localhost:3010/api>
- Application JavaFX : lancement local

Comptes de test :

- Administrateur :
login : admin@test.com
mot de passe : admin123
- Enseignant :
login : prof@test.com
mot de passe : prof123
- Étudiant :
login : etudiant@test.com
mot de passe : etudiant123

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.

**ANNEXE VII-1-B : Fiche descriptive de réalisation professionnelle
(verso, éventuellement pages suivantes)**

Épreuve E6 - Conception et développement d'applications (option SLAM)

Descriptif de la réalisation professionnelle, y compris les productions réalisées et schémas explicatifs

Présentation générale

Cette réalisation consiste en le développement d'une application desktop en JavaFX permettant de gérer les données d'un centre de formation.

Elle s'appuie sur un backend existant (Node.js / Express) qui expose une API REST sécurisée.

L'application fonctionne selon une architecture client-serveur.

- Le backend gère la logique métier et la base de données
- L'application JavaFX agit comme client et consomme l'API

Les données sont échangées au format JSON

Architecture technique

L'application JavaFX est structurée selon une architecture MVC simplifiée :

- **Controllers** : gestion des interactions utilisateur
- **Models** : représentation des données
- **Services** : appels à l'API REST
- **Views (FXML)** : interface graphique

Données manipulées

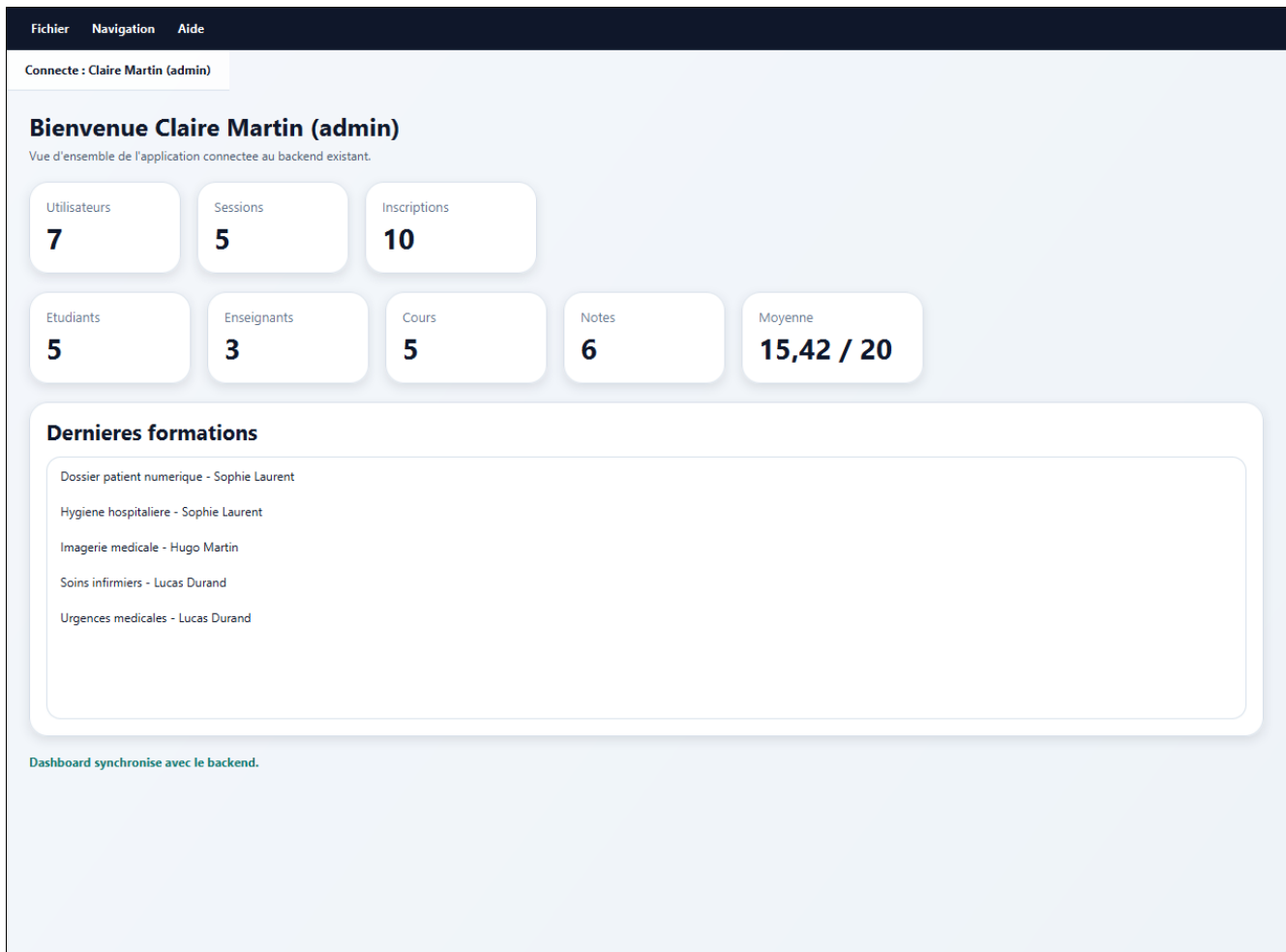
Les principales entités sont :

- **Utilisateurs**
- **Étudiants**
- **Enseignants**
- **Cours**
- **Sessions**
- **Inscriptions**
- **Notes**

Ces données sont récupérées via l'API et non stockées localement.

Fonctionnalités principales

- **Authentification** :
Connexion sécurisée avec token JWT
- **Gestion des données** :
CRUD complet sur étudiants, enseignants, cours, sessions, inscriptions et notes
- **Interface utilisateur** :
Tableaux, formulaires, navigation intuitive
- **Listes déroulantes** :
Sélection dynamique des entités liées pour éviter les erreurs
- **Tableau de bord** :
Statistiques globales récupérées depuis le backend



Cette capture présente le tableau de bord administrateur.

Il affiche des indicateurs globaux en temps réel récupérés depuis l'API (nombre d'utilisateurs, étudiants, enseignants, cours, sessions, inscriptions et notes).

Cela permet une vision synthétique de l'activité du centre de formation.

Résultat

L'application est fonctionnelle, connectée au backend et permet une gestion complète des données dans un environnement desktop.

Elle améliore l'ergonomie par rapport à une interface web et démontre la capacité à intégrer une API REST dans une application Java.