



**INSTITUT TECCART**

# **RAPPORT DE STAGE DE FIN D'ÉTUDES**

Présenté en vue de l'obtention du  
**Diplôme d'études collégiales (DEC)**  
**TECHNIQUE DE L'INFORMATIQUE**

**PAR**

**Ait Si Selmi Ouali**

**Développement d'une plateforme web de formation en ligne pour  
les agents de sécurité**

**Encadrant à l'institut : M. Safouen Bani**

**Année 2025**

# Remerciements

Je remercie chaleureusement M. Safouen Bani pour son accompagnement, ses conseils et sa disponibilité tout au long de ce stage. Merci également à l'équipe technique de l'Institut Teccart pour le soutien apporté à la mise en place de l'environnement et aux validations régulières. Enfin, merci à mes collègues et proches pour leurs encouragements.

# Résumé

Ce rapport présente la conception et la réalisation de **ACADEMIAE.CA**, une plateforme web de formation en ligne destinée aux agents de sécurité. Le projet a été conduit selon une approche agile et découpé en neuf sprints. La solution couvre l'inscription et l'authentification des utilisateurs, la gestion des cours et des examens, une messagerie interne, la remise de travaux, la rétroaction, un système de notifications et l'intégration d'un chatbot d'assistance. L'architecture repose sur React, Node.js/Express, MySQL, API REST, JWT, bcrypt. Les résultats démontrent la faisabilité technique et la pertinence fonctionnelle pour la formation continue des agents.

**Mots-clés :** E-learning, React, Node.js/Express, MySQL, API REST, JWT, bcrypt, Chatbot, Sécurité applicative.

# Abstract

This report describes the design and implementation of **ACADEMIAE.CA**, an online training platform for security agents. The project followed an agile methodology in nine sprints. The solution includes user registration and authentication, course and exam management, internal messaging, assignment submission, feedback, notifications, and a support chatbot. The architecture is based on React, Node.js/Express, MySQL, API REST, JWT, bcrypt. Results show the technical feasibility and functional relevance for continuous training.

**Keywords :** E-learning, React, Node.js/Express, MySQL, API REST, JWT, bcrypt, Chatbot, Application Security.

# Table des matières

<b>Remerciements</b>	<b>1</b>
<b>Résumé</b>	<b>2</b>
<b>Abstract</b>	<b>3</b>
<b>Tableau de bord des réalisations</b>	<b>6</b>
<b>Introduction générale</b>	<b>8</b>
<b>1 Présentation du cadre du projet</b>	<b>9</b>
1.1 Cadre du projet . . . . .	9
1.2 Présentation de l'établissement d'accueil . . . . .	9
1.3 Étude de l'existant . . . . .	9
1.4 Solution proposée . . . . .	9
1.5 Méthodologie adoptée . . . . .	10
<b>2 Spécification des besoins</b>	<b>11</b>
2.1 Identification des acteurs . . . . .	11
2.2 Besoins fonctionnels (extraits) . . . . .	11
2.3 Besoins non fonctionnels (extraits) . . . . .	11
2.4 Diagrammes UML (exemples) . . . . .	12
<b>3 Étude conceptuelle</b>	<b>13</b>
3.1 Architecture générale . . . . .	13
3.2 Modélisation (séquences – exemples) . . . . .	13
3.3 Base de données (tables principales) . . . . .	13
<b>4 Réalisation</b>	<b>15</b>
Sprint 1 – Préparation de l'environnement . . . . .	15
Sprint 2 – Authentification & demande d'inscription . . . . .	15
Sprint 3 – Espace Admin & début module cours . . . . .	15
Sprint 4 – Gestion des cours & notes . . . . .	15
Sprint 5 – Module examens . . . . .	16
Sprint 6 – Côté Agent . . . . .	16
Sprint 7 – Messagerie, remises, rétroaction . . . . .	16
Sprint 8 – Notifications, recherche, sécurité . . . . .	16
Sprint 9 – Chatbot, design, finalisation . . . . .	16
<b>Conclusion générale</b>	<b>17</b>

<b>Netographie</b>	<b>18</b>
<b>A Captures d'écran et description des fonctionnalités</b>	<b>19</b>

# Tableau de bord des réalisations

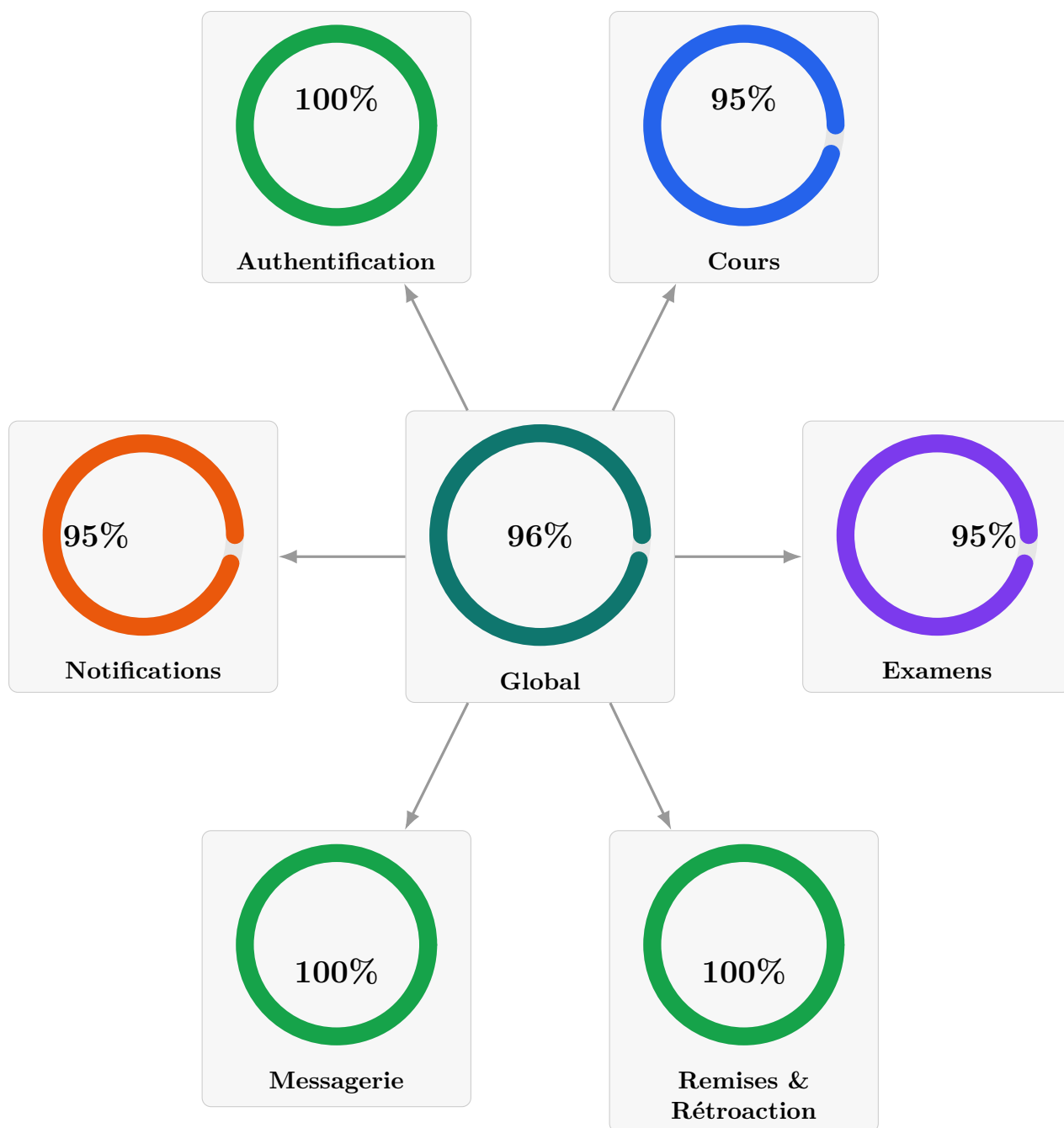


FIGURE 1 – Synthèse visuelle — progression globale et modules (exemple).

*Note* : adapte les pourcentages ci-dessus selon tes résultats réels.

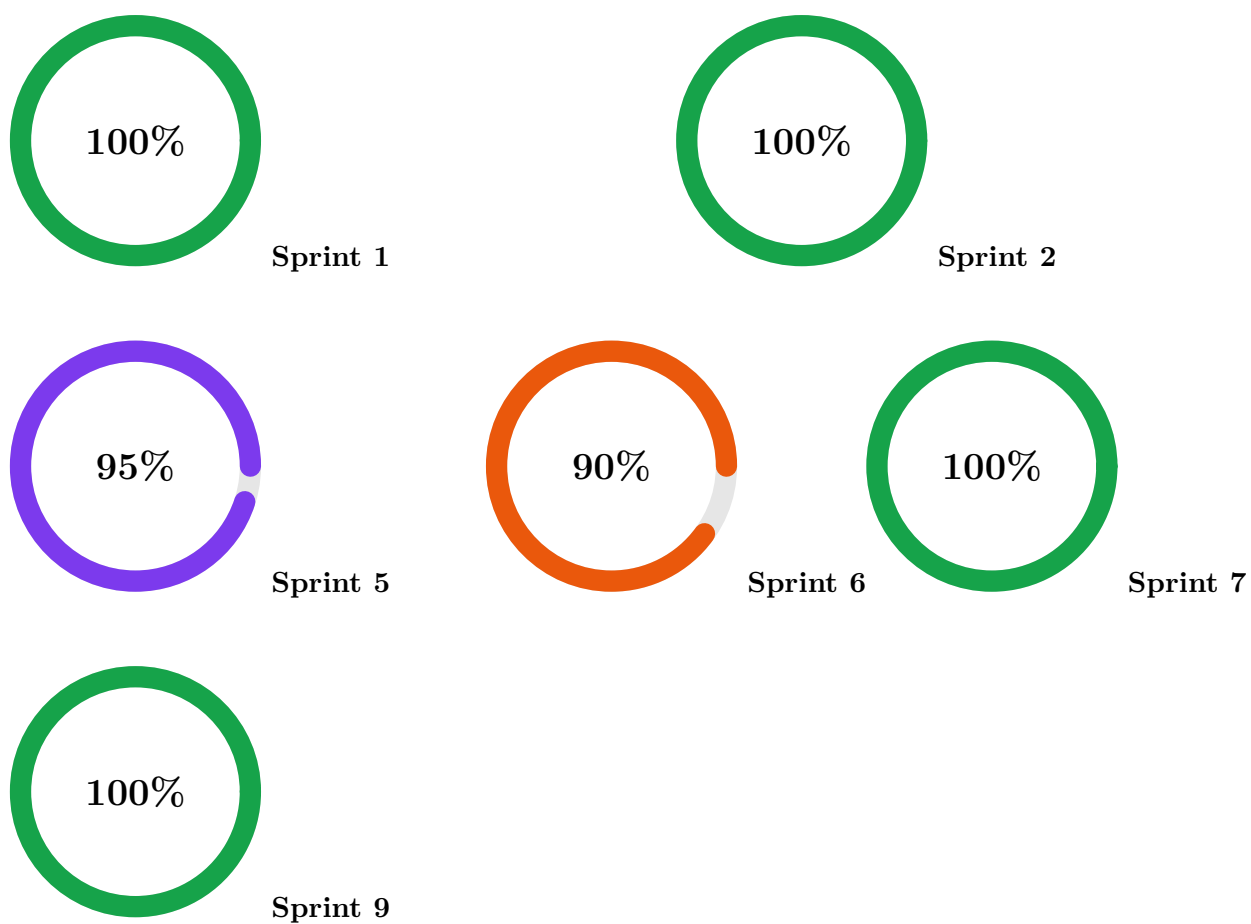


FIGURE 2 – Avancement par sprint (exemple, modifiable).

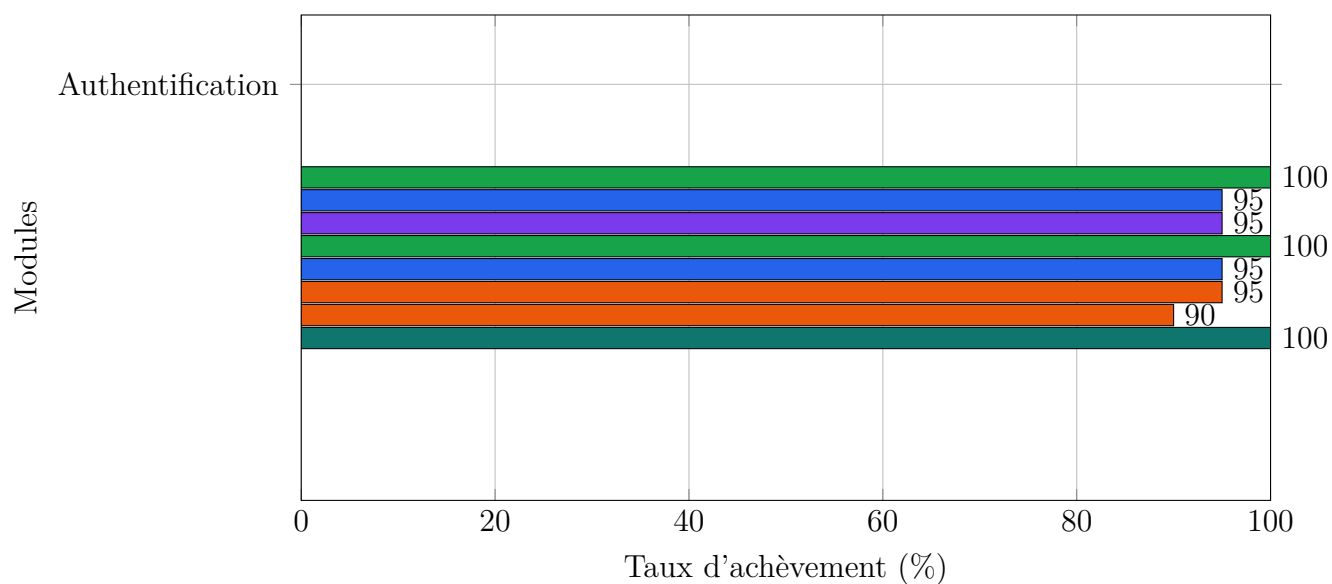


FIGURE 3 – Achèvement par module (exemple, modifiable).



# Introduction générale

La formation des agents de sécurité requiert une mise à jour régulière des connaissances et des pratiques. Les plateformes d'apprentissage en ligne (e-learning) répondent à ces besoins par l'accessibilité, la traçabilité et l'automatisation. Dans ce contexte, le projet **ACADEMIAE.CA** propose une solution web moderne permettant de gérer l'ensemble du cycle de formation (cours, examens, rétroaction, communication) au sein d'une même interface, avec une architecture fiable et sécurisée.

Ce rapport reprend la structure d'un rapport de fin d'études académique et détaille le cadre du projet, les besoins fonctionnels et non fonctionnels, l'étude conceptuelle (architecture et modèles), la réalisation (par sprints), puis conclut sur les acquis et pistes d'amélioration.

# Chapitre 1

## Présentation du cadre du projet

### 1.1 Cadre du projet

Le projet s'inscrit dans le cadre du stage de fin d'études du DEC *Technique de l'informatique* à l'Institut Teccart. L'objectif est de concevoir et développer **ACADEMIAE.CA**, une plateforme e-learning pour agents de sécurité, de la phase d'analyse jusqu'à une version fonctionnelle démontrable.

### 1.2 Présentation de l'établissement d'accueil

L'Institut Teccart est un établissement d'enseignement technique. Le stage est réalisé au sein du service technique (développement et support). Les échanges réguliers avec l'encadrant ont permis d'orienter les choix techniques et les priorités.

### 1.3 Étude de l'existant

Les processus actuels d'inscription, de suivi des cours, de gestion des examens et de communication sont souvent dispersés (emails, fichiers, formulaires manuels). Les limites observées :

- Multiplication des outils et absence de traçabilité centralisée ;
- Faible automatisation (validations, rétroaction, notifications) ;
- Accès hétérogène selon les profils, ergonomie non unifiée.

### 1.4 Solution proposée

**ACADEMIAE.CA** centralise :

- **Espace Admin** : gestion des agents, cours, examens, notes, rétroactions, suivi des soumissions ;
- **Espace Agent** : inscription/connexion, profil, liste de cours, tests, remise de fichiers, messagerie ;
- **Services transverses** : notifications, recherche, rôles et permissions, sécurité, chat-bot d'assistance.

## 1.5 Méthodologie adoptée

Une **démarche agile par sprints** a été retenue (neuf sprints). Chaque sprint comporte objectifs, tâches, résultats et validation. Les livrables incluent code source, schémas UML, scripts SQL et captures fonctionnelles.

# Chapitre 2

## Spécification des besoins

### 2.1 Identification des acteurs

**Administrateur** gère utilisateurs, cours, examens, notes, rétroaction, notifications.

**Agent** s'inscrit/connexion, consulte cours et tests, dépose des fichiers, reçoit la rétroaction, communique.

**Formateur** (optionnel) crée/évalue contenus, fournit rétroaction (peut être assimilé au rôle Admin selon l'organisation).

### 2.2 Besoins fonctionnels (extraits)

#### Agent

- Authentification (inscription/connexion) ;
- Consultation des cours et tests ; remise de devoirs/examens (fichiers) ;
- Messagerie interne ; réception de rétroaction ; notifications.

#### Administrateur

- Gestion des comptes (agents) et des demandes d'inscription ;
- Gestion des cours, tests, barèmes ; suivi des soumissions ; attribution des notes et commentaires ;
- Supervision des messages, annonces, notifications ; export (rapports).

### 2.3 Besoins non fonctionnels (extraits)

- **Sécurité** : hashage (bcrypt), JWT, contrôle d'accès par rôles, expiration de session ;
- **Performance & UX** : interface réactive, pagination/recherche, responsive ;
- **Qualité** : journalisation des erreurs, validations côté client/serveur, tests de base ;
- **Évolutivité** : architecture modulaire (front React, API Express, BD MySQL).

## 2.4 Diagrammes UML (exemples)



FIGURE 2.1 – Diagramme de cas d'utilisation principal (Admin/Agent) — exemple.

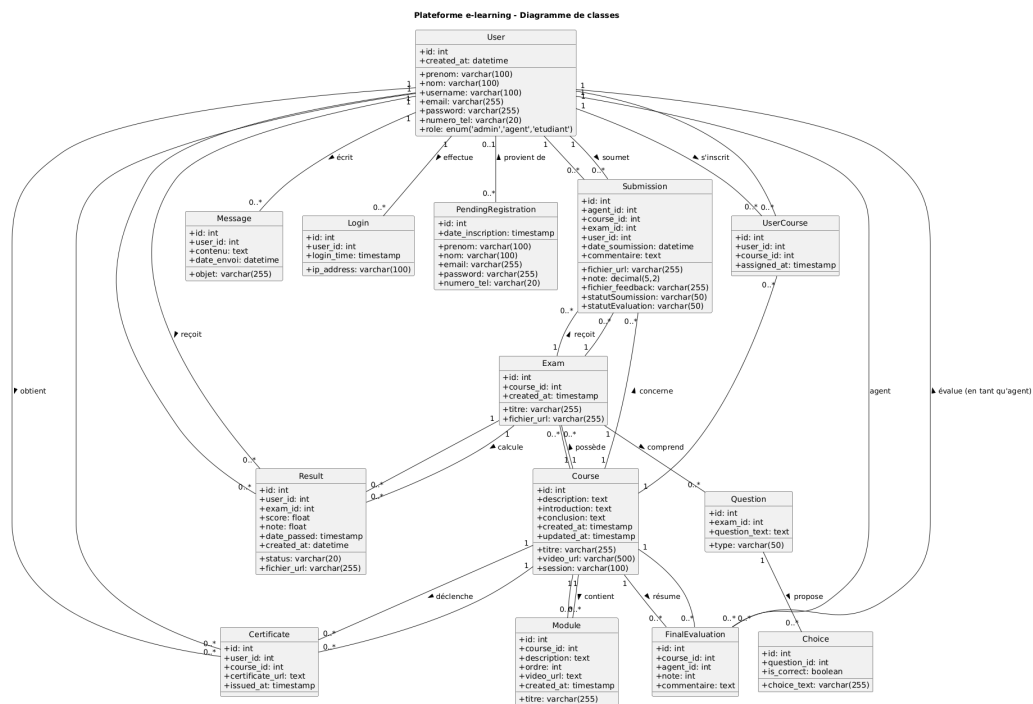


FIGURE 2.2 – Diagramme de classes (extrait) — exemple.

# Chapitre 3

## Étude conceptuelle

### 3.1 Architecture générale

Architecture **trois tiers** :

- **Front-end** : React (routes, pages, composants, appels API) ;
- **Back-end** : Node.js/Express (routes REST, contrôleurs, middleware auth, upload) ;
- **Base de données** : MySQL (tables users, courses, exams, submissions, messages, notifications...).

### 3.2 Modélisation (séquences – exemples)

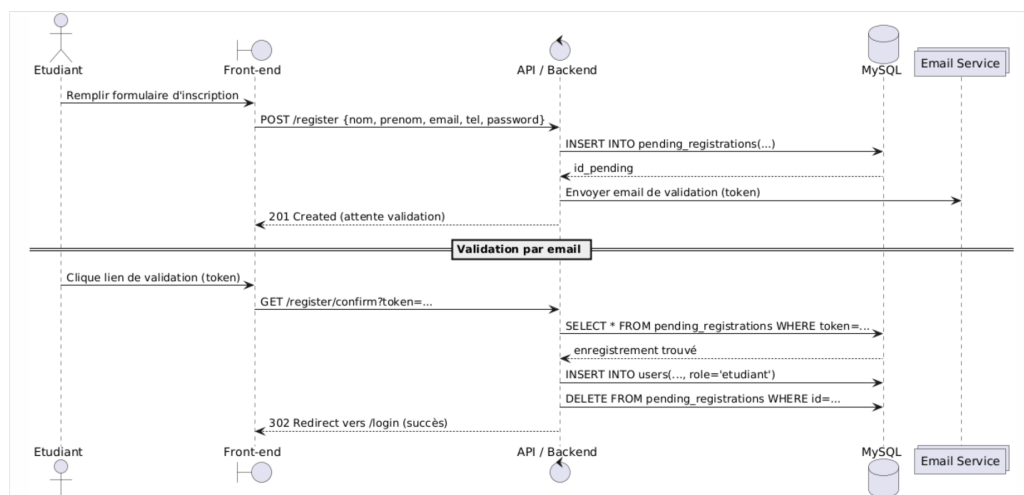


FIGURE 3.1 – Séquence : authentification (inscription/connexion) — exemple.

Image manquante : `sequence_soumission.png`

FIGURE 3.2 – Séquence : remise d'un examen — exemple.

### 3.3 Base de données (tables principales)

Table	Champs (extraits)
users	id, firstname, lastname, email (unique), phone, role (admin/agent), password_hash, created_at
courses	id, title, description, created_by, created_at
exams	id, course_id, title, due_date, instructions
submissions	id, exam_id, user_id, filename, submitted_at, grade, feedback
messages	id, sender_id, recipient_id, subject, body, created_at, is_read
notifications	id, user_id, type, payload, created_at, is_seen

# Chapitre 4

## Réalisation

### **Sprint 1 – Préparation de l’environnement (26/05/2025–01/06/2025)**

**Objectifs** : installer outils, définir architecture, clarifier besoins.

**Tâches** : Installation Node.js/React, MySQL ; dépôt Git ; config env local ; réunion de cadrage ; rédaction UML ; doc initiale.

**Résultats** : Environnement prêt, outils fonctionnels, besoins clarifiés.

### **Sprint 2 – Authentification & demande d’inscription (02/06–06/06)**

**Objectifs** : pages connexion/inscription ; formulaire d’inscription agent ; liaison BD.

**Tâches** : Pages React (login/register) ; API `/auth` ; hashage bcrypt ; enregistrement MySQL ; tests utilisateurs fictifs.

**Résultats** : Authentification fonctionnelle, données persistées.

### **Sprint 3 – Espace Admin & début module cours (09/06–13/06)**

**Objectifs** : tableau de bord Admin ; liste agents ; demandes d’inscription ; amorce gestion cours.

**Tâches** : Pages Admin (liste, recherche) ; API agents/demandes (GET/PUT) ; structure page cours.

**Résultats** : Espace Admin partiel ; flux demandes OK ; base cours amorcée.

### **Sprint 4 – Gestion des cours & notes (17/06–21/06)**

**Objectifs** : poursuivre page cours ; créer page notes ; compléter BD ; retours superviseur.

**Tâches** : Affichage/chargement cours ; saisie/affichage notes ; nouvelles tables ; clés étrangères ; ajustements ergonomie.

**Résultats** : Notes (v1) fonctionnelles ; BD adaptée ; UI plus claire.



## Sprint 5 – Module examens (23/06–29/06)

**Objectifs** : finaliser page examens ; stabiliser ; améliorer CSS global.

**Tâches** : CRUD examens (front/back) ; tables associées ; routage `/api/examens` ; correction bugs ; uniformisation styles ; responsive.

**Résultats** : Examens opérationnels ; fiabilité accrue ; interface professionnelle.

## Sprint 6 – Côté Agent (30/06–06/07)

**Objectifs** : profil agent ; gestion des cours côté agent ; messagerie (réception/lecture).

**Tâches** : Page profil (édition) ; liste cours par session ; messagerie (boîte de réception, statut lu/non-lu) ; accueil agent repensé.

**Résultats** : Profil/cours/messagerie côté agent opérationnels ; UX améliorée.

## Sprint 7 – Messagerie complète & remises & rétroaction (07/07–13/07)

**Objectifs** : envoyer/répondre messages ; remise de fichiers ; page de rétroaction.

**Tâches** : Upload (Word/PDF) ; API soumissions + métadonnées ; page rétroaction (notes/commentaires/fichiers).

**Résultats** : Messagerie complète ; remises actives ; rétroaction consultable.

## Sprint 8 – Notifications, recherche, sécurité, gestion avancée (14/07–20/07)

**Objectifs** : notifications ; recherche globale ; renforcement sécurité ; gestion cours/tests (admin & agent).

**Tâches** : Notifications internes (polling simple) ; barres de recherche (cours/messages/examens) ; auto-logout inactivité ; vérif rôles ; CRUD cours (admin), vues détaillées (agent) ; suivi tests (admin) et remises (agent).

**Résultats** : Notifs/recherche déployées ; accès sécurisé ; flux cours/tests robustes.

## Sprint 9 – Chatbot IA, refonte design, finalisation (21/07–30/07)

**Objectifs** : chatbot d'assistance ; harmonisation UI ; stabilisation & doc.

**Tâches** : Intégration bouton flottant ; FAQ ; uniformisation composants ; responsive tablette/mobile ; correction bugs ; nettoyage code ; docs technique & utilisateur (captures).

**Résultats** : Chatbot utile ; interface moderne et cohérente ; projet finalisé et documenté.

# Conclusion générale

Le projet **ACADEMIAE.CA** a permis de concevoir et développer une plateforme e-learning adaptée aux besoins des agents de sécurité, en suivant une démarche agile et en s'appuyant sur React, Node.js/Express, MySQL, API REST, JWT, bcrypt. Les fonctionnalités livrées couvrent l'essentiel du parcours d'apprentissage et de suivi.

**Perspectives** : application mobile, tableaux de bord analytiques, intégration SSO, stockage objet (ex. S3) pour les soumissions, et tests automatisés plus complets (unitaires, intégration, E2E).

# Netographie

- Documentation React : <https://react.dev/>
- Documentation Express : <https://expressjs.com/>
- MySQL Reference : <https://dev.mysql.com/doc/>
- OWASP Cheat Sheets : <https://cheatsheetseries.owasp.org/>

# Annexe A

## Captures d'écran et description des fonctionnalités

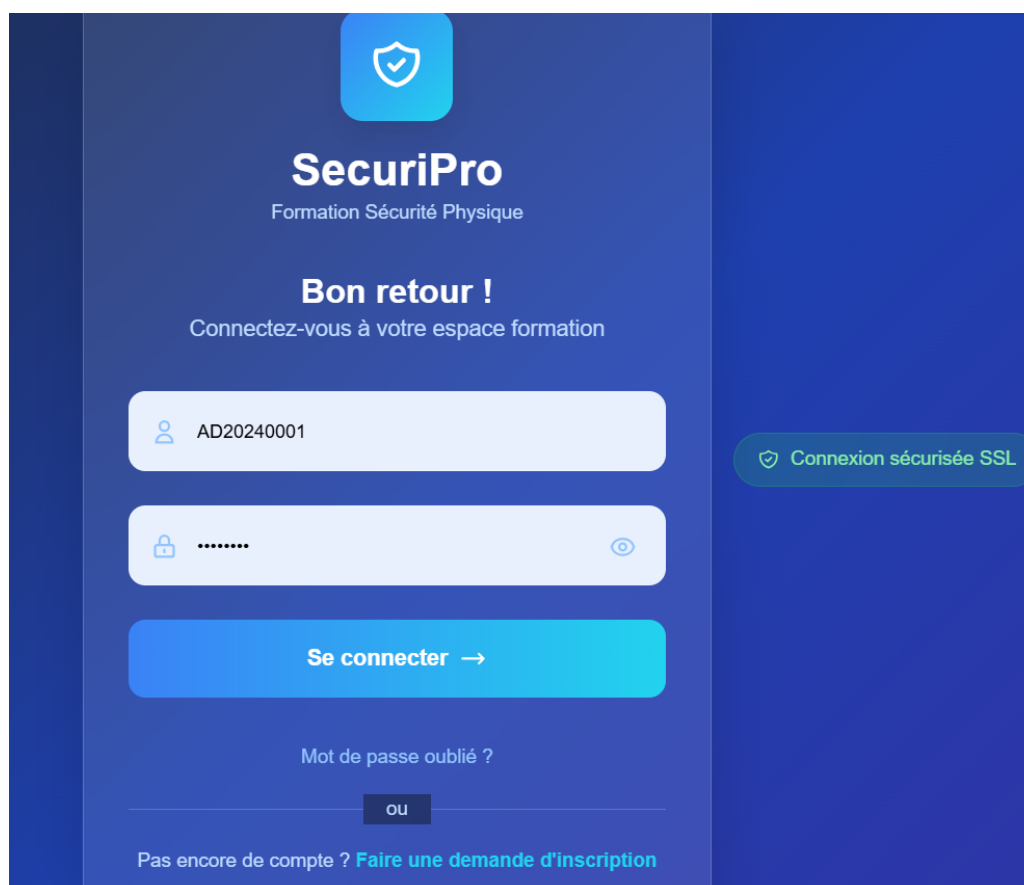


FIGURE A.1 – **Page de connexion** — Permet aux utilisateurs (agents, administrateurs) de se connecter à la plateforme avec leur identifiant et mot de passe. Inclut une validation des champs et un message d'erreur en cas d'identifiants incorrects.



FIGURE A.2 – **Tableau de bord administrateur** — Vue centralisée affichant les statistiques clés, l'état des cours, les évaluations en attente et les dernières connexions des utilisateurs.

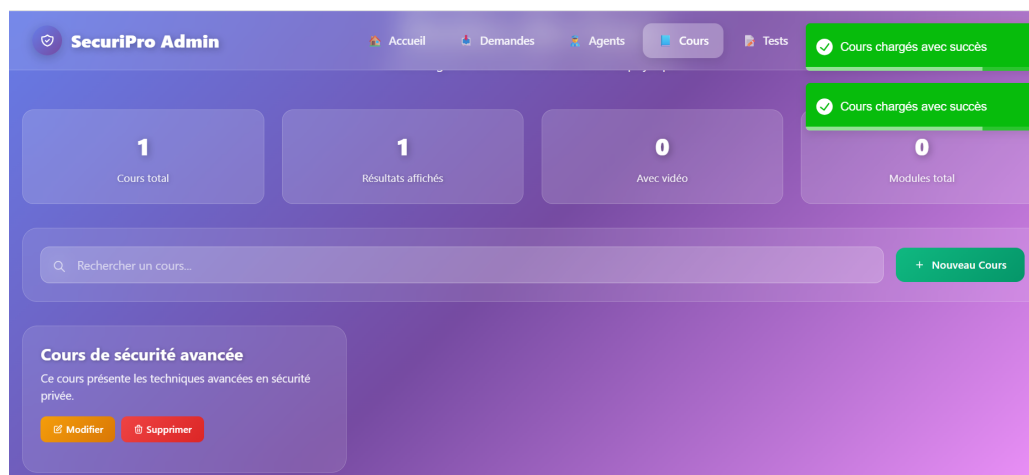


FIGURE A.3 – **Gestion des cours** — Interface pour créer, modifier, publier et supprimer des modules de formation. Permet aussi l'ajout de ressources téléchargeables et de vidéos.

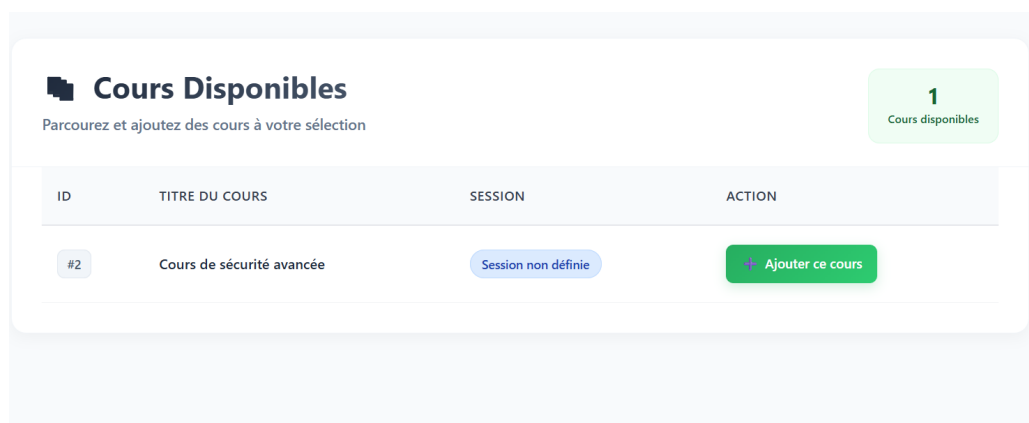


FIGURE A.4 – **Consultation des cours** — Les agents peuvent parcourir les modules, visualiser les leçons et télécharger les ressources associées.

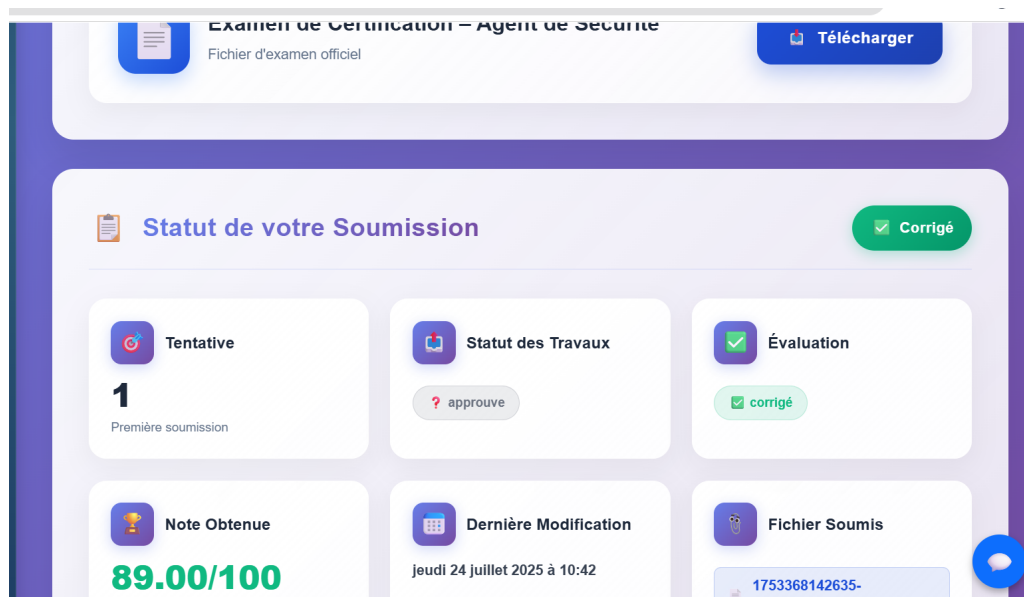


FIGURE A.5 – **Liste des tests** — Permet aux agents de voir les examens disponibles, de vérifier les dates limites et d'accéder aux consignes.



FIGURE A.6 – **Soumission d'examen** — Interface où l'agent peut envoyer ses fichiers de réponses et recevoir une confirmation immédiate de l'envoi.



FIGURE A.7 – **Rétroaction et notes** — Affiche la note obtenue, les commentaires du formateur et les fichiers de correction, permettant à l’agent d’améliorer ses compétences.

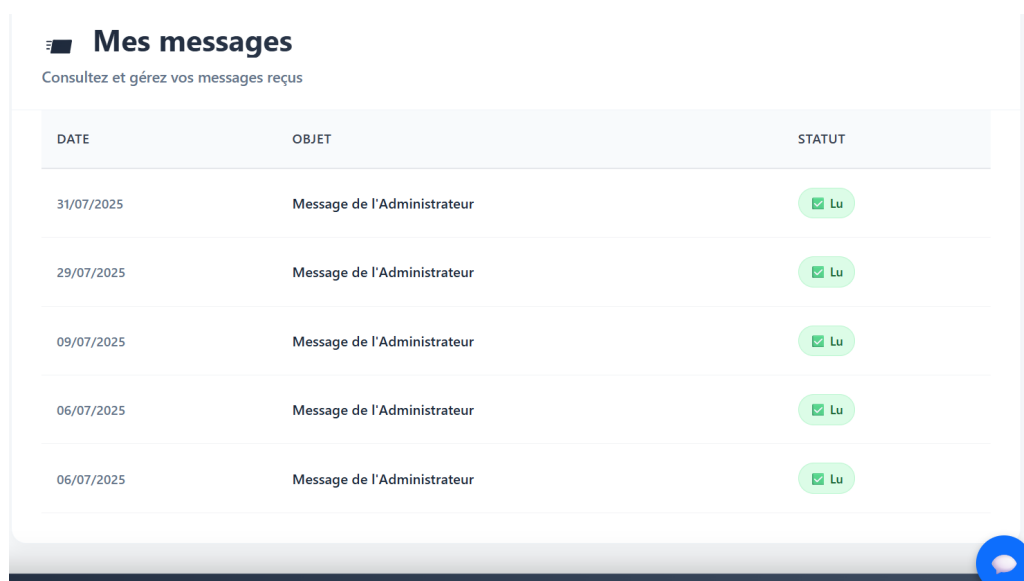


FIGURE A.8 – **Messagerie interne** — Système de communication intégré pour échanger entre agents et formateurs, avec historique des conversations.

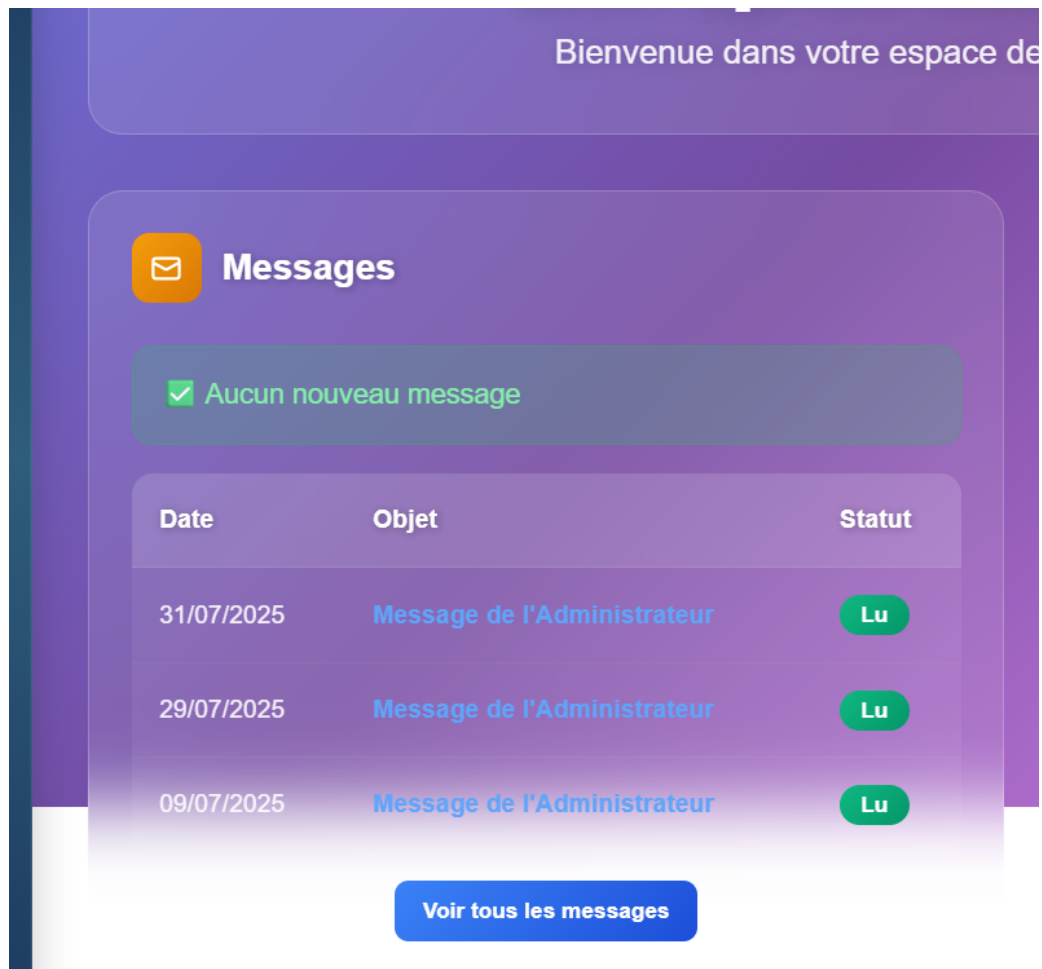


FIGURE A.9 – **Notifications en temps réel** — Alertes pour informer des nouveaux examens, messages, rétroactions ou mises à jour de cours.

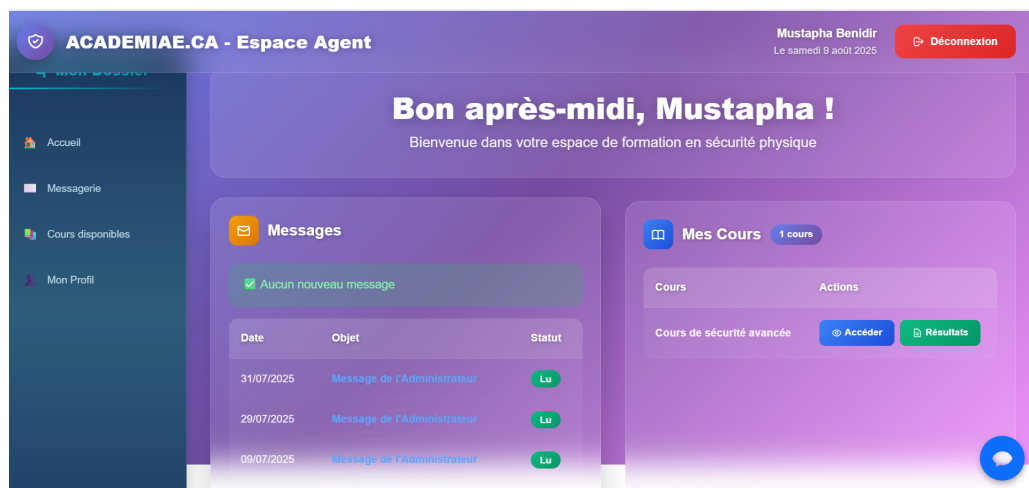


FIGURE A.10 – **Tableau de bord des réalisations** — Vue synthétique de l'avancement de toutes les tâches avec indicateurs colorés, cercles de progression et graphiques.