# 

# ****Rapport de Projet de Fin de Formation****

### Développement Web Full Stack JS

# ****Développement d’une application****

# ****e-learning : MySmartCode****

### 

### Réalisé par :

**Ourari Mondher**

### Encadré par :

**M. Mehdi Ben Alaya  
Instructeur– GoMyCode**

### Année : 2024-2025

## Introduction générale

### Contexte et motivation

Dans un monde en constante digitalisation, l'accès à la formation continue est devenu essentiel.

L’e-learning offre une solution moderne permettant aux apprenants de se former à leur rythme,

selon leurs disponibilités. C’est dans ce contexte que nous avons développé une application web

de formation en ligne nommée **MySmartCode**.

### Objectif du projet

Concevoir et réaliser une application web de formation en ligne avec deux profils principaux d’utilisateurs :

**- Etudiant :** s'inscrire, se connecter, consulter /chercher des cours, se déconnecter.

**- Administrateur :** se connecter, ajouter/supprimer /chercher des cours, ajouter/supprimer /chercher

des étudiants, se déconnecter.

### Méthodologie

Nous avons adopté une méthodologie **Agile Scrum**, en organisant le travail par **sprints** successifs

de durée fixe, avec planification, développement, revues et rétrospectives.

### Organisation du rapport

Ce rapport est organisé comme suit :

- Chapitre 1- Analyse de l’existant et spécification des besoins

- Chapitre 2- Conception fonctionnelle et technique

- Chapitre 3- Gestion et réalisation du projet

- Conclusion et perspectives

## Chapitre 1 : Analyse de l’existant et spécification des besoins

### 1.1 Problématique

### De nombreuses plateformes de formation sont complexes, avec des interfaces surchargées

### et des fonctionnalités inutiles pour un utilisateur débutant. Il est donc utile de créer une solution minimale, et facile à utiliser.

### 1.2 Objectifs fonctionnels

**Rôle : Etudiant**

* Inscription
* Connexion
* Consultation des cours
* Chercher des cours
* Déconnexion

**Rôle : Administrateur**

* Connexion
* Ajouter / supprimer /chercher des cours
* Ajouter / supprimer /chercher des étudiants
* Déconnexion

### 1.3 Analyse concurrentielle

Des plateformes comme Moodle, Coursera ou OpenClassrooms sont riches mais trop lourdes.

MySmartCode opte pour une expérience utilisateur simplifiée, adaptée aux besoins fondamentaux

de formation.

### 1.4 Spécification des besoins

**Fonctionnels :**

* Gestion des étudiants
* Gestion des cours
* Interfaces adaptées à chaque rôle

**Non fonctionnels :**

* Simplicité d’utilisation
* Ergonomique
* Architecture claire (frontend/backend)

## Chapitre 2 : Conception fonctionnelle et technique

### 2.1 Architecture technique

L’application repose sur une architecture **MERN** (MongoDB, Express, React, Node.js) :

* **Frontend** : React (HTML, CSS, JS)
* **Backend** : Node.js avec Express
* **Base de données** : MongoDB via Mongoose

### 2.2 Arborescence du projet

MySmartCode/

├── backend/

│ ├── controllers/

│ ├── models/

│ ├── routes/

│ ├── uploads/

│ ├── middleware/

│ ├── server.js

│ └── .env

├── frontend/

│ ├── public/

│ │ └── index.html

│ ├── src/

│ │ ├── components/

│ │ ├── pages/

│ │ ├── App.js

│ │ ├── index.js

│ │ └── styles.css

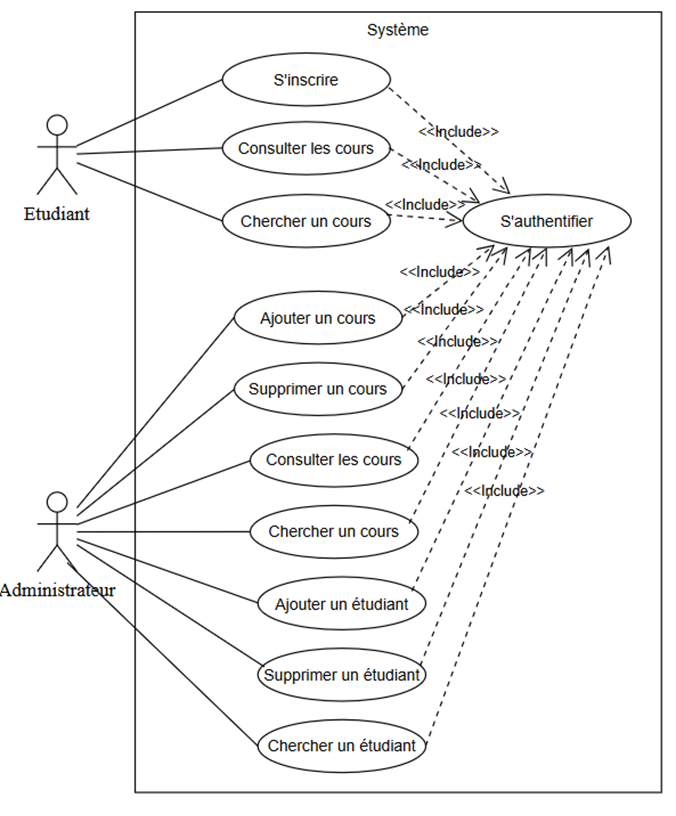
│ └── .env

└── README.md

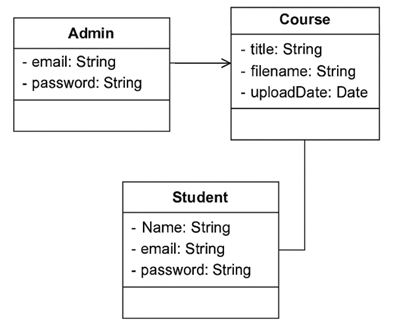
### 2.3 Modélisation des données

* **Admin** : email, password
* **Student** : name, email, password
* **Course** : title, fichier (HTML, PDF, Word, etc.))

### 2.4 Le diagramme de cas d’utilisation globale



### 2.5 Le diagramme de classe globale



## Chapitre 3 : Gestion et réalisation du projet

### 3.1 Rôles Scrum

* **Product Owner :** Mehdi Ben Alaya
* **Scrum Master :** Ourari Mondher
* **Equipe de développement :** Ourari Mondher

### 3.2 Backlog produit

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **User Story** | **Rôle** | **Priorité** |
| US1 | En tant qu'étudiant, je peux m'inscrire | Étudiant | Haute |
| US2 | En tant qu'étudiant, je peux me connecter | Étudiant | Haute |
| US3 | En tant qu'étudiant, je peux consulter les cours | Étudiant | Moyenne |
| US4 | En tant qu'étudiant, je peux rechercher un cours | Étudiant | Basse |
| US5 | En tant qu'étudiant, je peux me déconnecter | Étudiant | Haute |
| US6 | En tant qu'administrateur, je peux me connecter | Administrateur | Haute |
| US7 | En tant qu'administrateur, je peux ajouter un cours | Administrateur | Haute |
| US8 | En tant qu'administrateur, je peux supprimer un cours | Administrateur | Moyenne |
| US9 | En tant qu'administrateur, je peux consulter les cours | Administrateur | Moyenne |
| US10 | En tant qu'administrateur, je peux rechercher un cours | Administrateur | Basse |
| US11 | En tant qu'administrateur, je peux ajouter un étudiant | Administrateur | Moyenne |
| US12 | En tant qu'administrateur, je peux supprimer un étudiant | Administrateur | Moyenne |
| US13 | En tant qu'administrateur, je peux rechercher un étudiant | Administrateur | Moyenne |
| US14 | En tant qu'administrateur, je peux me déconnecter | Administrateur | Haute |

|  |  |
| --- | --- |
| **Priorité** | **Définition** |
| Haute | Fonction **indispensable** au bon fonctionnement de l’application. |
| Moyenne | Fonction **importante** mais l’application reste utilisable sans elle. |
| Basse | Fonction **complémentaire** ou d’amélioration. |

### 3.3 Sprints

|  |  |  |  |
| --- | --- | --- | --- |
| **Sprint** | **Durée** | **User Stories incluses** | **Détail par Sprint** |
| Sprint 1 | 1 semaine | US1, US2, US5, US6, US14 | Authentification sécurisée pour étudiant et admin : inscription, connexion, déconnexion, recherche d'étudiant |
| Sprint 2 | 1 semaine | US3, US4, US9, US10 | Consultation et recherche des cours pour étudiant/admin, ajout d’étudiants |
| Sprint 3 | 1 semaine | US7, US8, US11, US12, US13 | Ajout / suppression de cours, gestion complète  des étudiants et recherche avancée |

### ****Critères utilisés pour organiser les Sprints****

|  |  |
| --- | --- |
| **Critère** | **Explication** |
| **Priorité (Haute > Moyenne > Basse)** | On commence par les fonctionnalités **indispensables** (US1, US2, US3, US11...) pour que l’application fonctionne de base. |
| **Dépendances logiques** | Certaines fonctionnalités doivent être prêtes **avant d’en implémenter d'autres**. |
| **Rôle de l'utilisateur** | On regroupe certaines User Stories par **rôle** (étudiant vs admin),  mais sans être rigide, pour équilibrer la charge par sprint. |
| **Équilibrage du volume**  **de travail** | Chaque sprint est prévu pour durer **environ 1 semaine**.  On évite donc d’y mettre trop ou trop peu d’US. |

### 3.4 Détails par sprint (backlog + conception + réalisation)

### 3.4.1 ****Sprint 1 – Authentification****

#### 3.4.1.1 Backlog du Sprint 1

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **User Story** | **Rôle** | **Priorité** |
| US1 | En tant qu’étudiant, je peux m’inscrire | Étudiant | Haute |
| US2 | En tant qu’étudiant, je peux me connecter | Étudiant | Haute |
| US5 | En tant qu’étudiant, je peux me déconnecter | Étudiant | Haute |
| US6 | En tant qu’administrateur, je peux me connecter | Administrateur | Haute |
| US14 | En tant qu’administrateur, je peux me déconnecter | Administrateur | Haute |

#### 3.4.1.2 Analyse du Sprint 1

**Description textuelle** :

- L’étudiant s’inscrit avec son nom, email et mot de passe.

- L’étudiant et l’administrateur peuvent se connecter.

- L’étudiant et l’administrateur peuvent se déconnecter.

#### 3.4.1.3 Conception du Sprint 1

**Interfaces** :

|  |  |
| --- | --- |
|  |  |
| Inscription étudiant | Connexion étudiant |

|  |
| --- |
|  |
| Déconnexion étudiant |

### 3.4.2 ****Sprint 2 – Recherche et consultation de cours****

#### 3.4.2 .1 Backlog du Sprint 2

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **User Story** | **Rôle** | **Priorité** |
| US3 | En tant qu’étudiant, je peux consulter les cours | Étudiant | Moyenne |
| US4 | En tant qu’étudiant, je peux rechercher un cours | Étudiant | Basse |
| US9 | En tant qu’administrateur, je peux consulter les cours | Administrateur | Moyenne |
| US10 | En tant qu’administrateur, je peux rechercher un cours | Administrateur | Basse |

#### 3.4.2 .2 Analyse du Sprint 2

**Description textuelle** :

L’étudiant et l’administrateur peuvent saisir un mot-clé et affichage instantané les cours correspondants.

#### 3.4.2 .3 Conception du Sprint 2

**Interface** :

- Zone de recherche et liste de résultats

|  |
| --- |
|  |
| Recherche et consultation de cours par un étudiant |

### 3.4.3 ****Sprint 3 – Gestion des cours et étudiants****

#### 3.4.3 .1 Backlog du Sprint 3

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **User Story** | **Rôle** | **Priorité** |
| US7 | En tant qu’administrateur, je peux ajouter un cours | Administrateur | Haute |
| US8 | En tant qu’administrateur, je peux supprimer un cours | Administrateur | Moyenne |
| US11 | En tant qu’administrateur, je peux ajouter un étudiant | Administrateur | Moyenne |
| US12 | En tant qu’administrateur, je peux supprimer un étudiant | Administrateur | Moyenne |
| US13 | En tant qu’administrateur, je peux rechercher un étudiant | Administrateur | Moyenne |

#### 3.4.3 .2 Analyse du Sprint 3

**Description textuelle** :

- Ajout de cours : formulaire simple (titre du cours) puis cliquer sur "**choisir un fichier** "

enfin cliquer sur "**Téléverser**"**.**

- Ajout un étudiant: formulaire simple (nom, email et mot de passe) puis cliquer sur "**Ajouter**"**.**

- Suppression : bouton "**Supprimer**" sur chaque élément

- Recherche : filtre dynamique

#### 3.4.3 .3 Conception du Sprint 3

**Interface** :

|  |
| --- |
|  |
| administrateur |

## Conclusion générale et perspectives

**Bilan du projet**

Ce projet a permis de mettre en pratique toutes les compétences du développement web full-stack enseignées chez GoMyCode. L’application MySmartCode est fonctionnelle et répond aux besoins fixés.

### Apports personnels

* Maîtrise de la stack MERN (MongoDB, Express, React, Node.js)
* Création d’API REST sécurisées
* Expérience pratique avec la méthode Scrum
* Amélioration de l’organisation et gestion de projet

### Perspectives

* Adapter l’application pour le mobile (responsive)
* Ajout de fonctionnalités : commentaires, notifications, messagerie, etc.
* Intégrer des rôles avancés (ex : formateur)

## Annexes

* Code source : GitHub MySmartCode : <https://github.com/ourarimondher/MySmartCode>
* Guide d’installation local (Node, MongoDB, npm install)

## Références

* Documentation officielle : ReactJS, ExpressJS, MongoDB
* Tutoriels GoMyCode