

Université Chouaib Doukkali
Faculté poly disciplinaire – SIDI BENNOUR
Filière : Business Intelligence et Big Data

Application WEB Gestion de Sclolarité

Réalisé par :

LAKOUAIRI Ibtissam

Année Universitaire : 2025-2026

Dédicace

Je dédie ce travail à mes chers parents, pour leur amour inconditionnel, leur soutien moral et financier, et leur encouragement constant tout au long de mon parcours académique.

À mes frères et soeurs, pour leur compréhension et leur présence réconfortante durant les moments difficiles.

À mes amis et collègues, pour leur aide précieuse et leur bonne humeur tout au long de cette expérience.

Enfin, à tous ceux qui ont contribué de près ou de loin à la réalisation de ce projet, avec toute ma gratitude.

Table des matières :

| | |
|---|----|
| Introduction générale : | 7 |
| CHAPITRE 1 : Contexte général et méthodologie de Développement du projet | 88 |
| Introduction | 9 |
| 1- Le processus de développement: | 9 |
| 2-1 Avantage: | 9 |
| 2-2 Limite: | 9 |
| 2-Le processus de développement en cascade: | 10 |
| Conclusion : | 10 |
| CHAPITRE 2 : Analyse et Conception | 11 |
| Introduction | 12 |
| 1-Cahier de charge | 12 |
| 1-2 Fonctionnalités de l'application : | 13 |
| B. Gestion des étudiants | 13 |
| C. Gestion des enseignants | 13 |
| D. Gestion des modules | 13 |
| E. Gestion des notes et contrôles | 14 |
| F. Relevé de notes | 14 |
| G. Interface utilisateur | 14 |
| H. Base de données structurée | 14 |
| I. Fiabilité et sécurité | 14 |
| 1-3 Besoins non fonctionnels : | 15 |
| 1-4 Contraintes techniques : | 15 |
| 2-Conception : | 16 |
| 2-1 Diagramme de cas d'utilisation: | 16 |
| □ Gérer les enseignants | 17 |
| □ Gérer les modules | 17 |
| □ Gérer les notes | 17 |
| □ Calculer les moyennes | 18 |

| | |
|--|-----------|
| □ Générer un relevé de notes | 18 |
| □ Affichage et consultation | 18 |
| ✓ Afficher la liste des étudiants..... | 18 |
| ✓ Afficher la liste des notes..... | 18 |
| CHAPITRE 3 : Les technologies utilisées | 20 |
| 1- Introduction : | 21 |
| 2- Langages de programmation utilisées : | 21 |
| 3- Les outils utilisés : | 24 |
| Conclusion : | 25 |
| CHAPITRE 4 : Présentation du System | 26 |
| Introduction : | 27 |
| 1- Page de connexion : | 27 |
| 2- Gestion des étudiants: | 27 |
| 3- Gestion des notes des étudiants : | 30 |
| 4- Gestion des modules: | 32 |
| 5- Gestion des étudiants: | 327 |
| 6- Gestion des enseignants: | 27 |
| 7- Génération de relvés de notes | 34 |
| ConclSION Générale | 34 |
| Webographie..... | 36 |

Liste des figures :

| | |
|---|----|
| Figure 1: Le processus de développement en cascade..... | 10 |
| Figure 2: Diagramme de cas d'utilisation de l' Admin | 16 |
| Figure 3: Diagramme de classe | 19 |
| Figure 4: Logo php..... | 21 |
| Figure 5: logo JavaScript..... | 21 |
| Figure 6: Logo HTML..... | 22 |
| Figure 7: Logo CSS..... | 22 |
| Figure 8: Logo SQL | 23 |
| Figure 9: Logo UML | 23 |
| Figure 10: Logo VS Code | 24 |
| Figure 11: Logo phpMyAdmin | 24 |
| Figure 12: Logo XAMPP | 25 |
| Figure 13: Page d'accueil..... | 27 |
| Figure 14: Liste des étudiants..... | 28 |
| Figure 15: Ajouter un nouveau étudiant..... | 28 |
| Figure 16: Modifier les informations d'un étudiant | 29 |
| Figure 17: Supprimer l'étudiante ibtissam..... | 29 |
| Figure 18: Gérer les notes d'un étudiant..... | 30 |
| Figure 19: Gérer les modules | 30 |
| Figure 20: Ajouter un module | 31 |
| Figure 21: Modifier un module | 31 |
| Figure 22: Liste des enseignants..... | 32 |
| Figure 23: Ajouter un étudiant | 32 |
| Figure 24: Modifier les informations d'un étudiant | 33 |
| Figure 25: Affichage de relevés de notes | 33 |
| Figure 26: Affichage de relevés de notes | 34 |
| Figure 27: Imprimer le relevé de notes d'un étudiant | 34 |

Tableau des acronymes :

| | |
|------|--------------------------------|
| CSS | Cascading Style Sheets |
| DCU | Diagramme de Cas d'Utilisation |
| JS | JavaScript |
| HTTP | HyperText Transfer Protocol |
| HTML | HyperText Markup Language |
| SQL | Structured Query Language |
| UML | Unified Modeling Language |

Introduction générale :

C'est avec un grand intérêt que je vous présente mon rapport d'un projet concret : **la conception et le développement d'une application web de gestion de scolarité**.réalisé dans le cadre de ma troisième année d'études en Licence Business Intelligence & BIG DATA à la Faculté Polydisciplinaire de Sidi Bennour.

Le premier chapitre expose le **contexte général du projet**, en mettant en lumière les enjeux et l'importance de la mise en place d'un tel système. Il permettra d'expliquer les motivations derrière ce projet ainsi que les objectifs à atteindre.

Le deuxième chapitre aborde **l'analyse et la conception**. Il analyse les fonctionnalités nécessaires pour répondre aux attentes des utilisateurs, Il détaillera l'architecture du système, les diagrammes UML, ainsi que la structuration des différentes fonctionnalités.

Le troisième chapitre présente les outils et technologies utilisés dans la réalisation de l'application. Il détaillera l'environnement de développement, les langages employés, ainsi que les différents outils logiciels qui ont permis de mettre en œuvre le projet.

Le troisième chapitre sera dédié à **la présentation du système**. Il décrira les principales fonctionnalités développées, l'interface utilisateur, ainsi que les tests réalisés pour assurer le bon fonctionnement de la plateforme.

CHAPITRE 1 :

Contexte général et méthodologie de Développement du projet

Ce paragraphe aborde le modèle de développement en cascade, ses avantages et ses limites et la méthodologie de développement du Projet.

Introduction

L'objectif principal de ce projet d'**application web de gestion de la scolarité**, permettant de faciliter l'organisation et la gestion des informations académiques de manière efficace et structurée.

Pour mener à bien ce projet, une **méthodologie de développement logiciel** a été adoptée afin d'assurer une progression claire et maîtrisée des différentes étapes de réalisation.

1- Le processus de développement:

Le modèle en cascade, ou « waterfall » en anglais, est une organisation des activités d'un projet sous forme de phases linéaires et séquentielles, où chaque phase correspond à une spécialisation des tâches et dépend des résultats de la phase précédente. Il comprend les phases d'exigences, de conception, de mise en œuvre et de mise en service. Le modèle en cascade est un cycle de vie de projet issu des industries manufacturières et du secteur de la construction, où une conception préalable est nécessaire, compte tenu des fortes contraintes matérielles et des coûts élevés afférents aux changements de la conception en cours de réalisation. Il est utilisé notamment dans les domaines de l'ingénierie et du développement de logiciels.

2-1 Avantage :

- Simple et facile à comprendre.
- Force la documentation : une phase ne peut se terminer avant qu'un document soit validé.
- Le test est inhérent à chaque phase.
- Les progrès sont tangibles (pour l'équipe de développement).
- Permet une **avancée linéaire et structurée**.

2-2 Limite :

- Non compréhensible par les clients.
- Le produit final c'est la première chose qui voit le client.
- Problèmes découverts en phase de validation.
- Ne marche que si les exigences sont stables et le problème connu.

2- Le processus de développement en cascade:

Ce modèle en cascade illustre une approche séquentielle du développement logiciel, où chaque étape dépend de la finalisation de la précédente. Il commence par la définition des **exigences**, suivie de l'**analyse**, puis de la **conception** du système. Ensuite, vient la phase de **codage**, puis celle des **tests**, et enfin la mise en **exploitation** du logiciel. Ce modèle est simple à comprendre et bien adapté aux projets dont les besoins sont bien définis dès le départ.

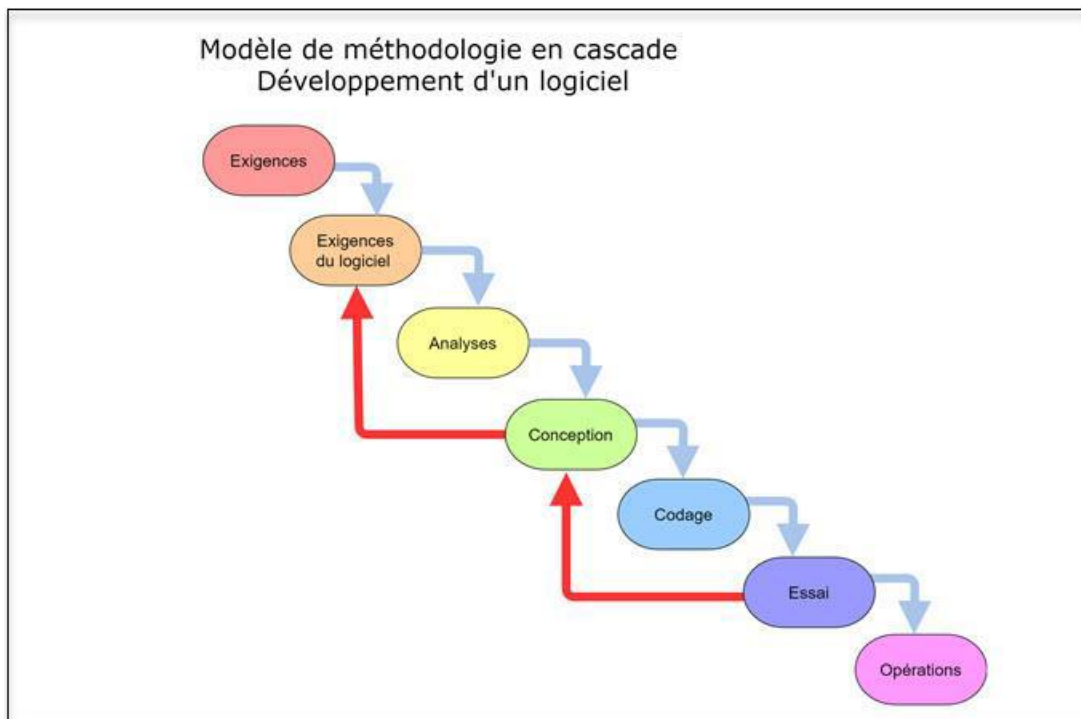


Figure 1 : Le processus de développement en cascade

Conclusion :

Au terme de cette partie, j'ai présenté le cadre général du projet et notamment le concept du processus du développement, le contexte du projet, ainsi que la conduite du projet.

CHAPITRE 2 :

Analyse et Conception

Dans ce chapitre, je présente une étude fonctionnelle susceptible de fournir une connaissance dit métier. La connaissance du métier est indispensable pour mener une bonne conception, et par la suite aborder la réalisation de palteforme.

Introduction

Ce chapitre est consacré à l'analyse détaillée des besoins fonctionnels et non fonctionnels du système, ainsi qu'à la conception globale de l'application web de gestion de scolarité.

L'objectif est d'identifier clairement les fonctionnalités attendues, de comprendre les interactions entre les différents utilisateurs et le système, puis de modéliser l'architecture générale à l'aide de diagrammes UML adaptés.

Cette étape constitue un fondement essentiel pour assurer une conception structurée, cohérente et conforme aux attentes des utilisateurs finaux.

1- Cahier de charge

1-1 Objectifs de l'application :

- **Centraliser toutes les informations scolaires**
Étudiants, enseignants, modules, notes — toutes les données sont stockées dans un système unique, sécurisé et structuré.
- **Automatiser les opérations de gestion**
Éviter la manipulation manuelle des fichiers Excel ou des documents papier.
- **Fiabiliser et sécuriser les données**
Réduire les erreurs humaines, éviter les duplications et garantir l'intégrité des données.
- **Faciliter le travail de l'administrateur**
Lui offrir une interface simple pour gérer toutes les opérations : ajout, modification, suppression, consultation.
- **Calculer automatiquement les moyennes et les mentions**
Automatiser les calculs pour éviter les erreurs.
- **Générer un relevé de notes clair et professionnel**
Affichage propre + option d'impression.
- **Offrir une interface intuitive et facile à utiliser**
Accessible même aux utilisateurs non techniques.

1-2 Fonctionnalités de l'application :

A. Authentification :

L'administrateur doivent se connecter à l'application à l'aide de leur identifiant et mot de passe pour accéder aux fonctionnalités. L'authentification est gérée via la **table users**, et des sessions PHP.

B. Gestion des étudiants

- Ajouter un étudiant
- Modifier les informations d'un étudiant
- Supprimer un étudiant
- Consulter la liste des étudiants
- Rechercher un étudiant
- Affecter des modules à un étudiant
- Voir les notes et la moyenne d'un étudiant

C. Gestion des enseignants

- Ajouter un enseignant
- Modifier les informations
- Supprimer
- Consulter la liste
- Associer un enseignant aux modules

D. Gestion des modules

- Ajouter un module
- Modifier
- Supprimer
- Consulter la liste des modules
- Définir les coefficients
- Associer un module à un enseignant

E. Gestion des notes et contrôles

- Ajouter une note (CC, examen)
- Modifier ou supprimer une note
- Calcul automatique :
 - ✓ moyenne du module
 - ✓ moyenne générale
 - ✓ mention (Ex. : Passable, Assez bien, Bien, Très bien)
- Affichage des notes même si une partie est absente (ex: pas encore saisi)

F. Relevé de notes

- Affichage automatique de tous les modules
- Calcul automatique des moyennes
- Mention générée automatiquement
- Mise en forme propre
- Bouton “Imprimer le relevé”

G. Interface utilisateur

- Interface claire, simple et intuitive
- Navigation fluide entre les pages
- Affichage propre des tables (étudiants, modules, notes...)

H. Base de données structurée

Tables utilisées

- étudiants
- enseignants
- modules
- notes
- utilisateurs (admin)

I. Fiabilité et sécurité

- Validation des données avant insertion
- Mot de passe crypté
- Accès administrateur sécurisé

1-3 Besoins non fonctionnels :

Rapidité : Notre plateforme doivent répondre aux besoins des utilisateurs dans le plus court délai.

Fiabilité : Bon fonctionnement de plateforme sans détection de défaillance.

L'extensibilité : L'application devront être extensibles, c'est-à-dire qu'il pourra y avoir une possibilité d'ajouter ou de modifier de nouvelles fonctionnalités.

La sécurité : les informations ne devront pas être accessibles à tout le monde.

La performance : le système doit réagir dans un délai précis, quel que soit l'action de l'utilisateur.

1-4 Contraintes techniques :

- L'application sera développée en utilisant un langage de programmation web (PHP, CSS, HTML , JavaScript).
- La base de données sera conçue à l'aide d'un système de gestion de bases de données relationnelles (MySQL).
- L'application devra être accessible via un navigateur web et être compatible avec les dernières versions des navigateurs web couramment utilisés (Google Chrome).

2- Conception :

2-1 Diagramme de cas d'utilisation:

Un diagramme de cas d'utilisation (DCU) est un type de diagramme UML qui représente les interactions entre un utilisateur (acteur) et un système. Il montre les différentes fonctionnalités offertes par le système sous forme de cas d'utilisation, ainsi que les relations entre eux. Ce diagramme est utilisé pour comprendre et définir les exigences fonctionnelles d'un projet.

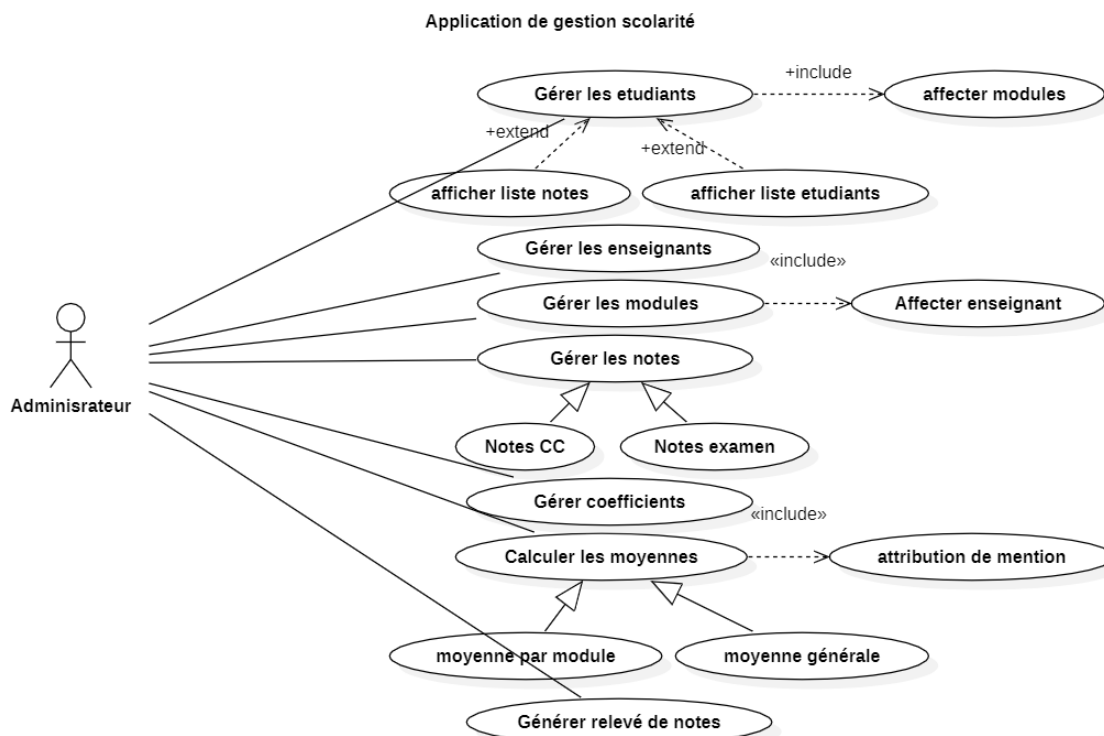


Figure 2: Diagramme de cas d'utilisation de l' Admin

Explication :

- **Gérer les étudiants**

- L'administrateur peut ajouter un nouvel étudiant en renseignant ses informations personnelles.
- Il peut modifier les informations d'un étudiant existant.
- Il peut supprimer un étudiant du système en cas de départ.
- L'administrateur peut consulter la liste complète des étudiants enregistrés.
- Il peut également **affecter des modules** à un étudiant, ce qui permet d'associer chaque étudiant aux matières qu'il doit suivre.

(Lien « include » avec affecter modules dans le diagramme.)

❖ Gérer les enseignants

- L'administrateur peut ajouter / supprimer un enseignant ainsi que ses informations (nom, spécialité...).
- Il peut modifier les informations d'un enseignant déjà existant.
- L'Admin peut consulter la liste de tous les enseignants.
- Il peut également **affecter un enseignant à un module**, afin de définir qui est responsable de chaque matière.
(Lien « include » visible dans le diagramme.)

❖ Gérer les modules

- L'administrateur peut créer un nouveau module (matière) en précisant son intitulé et son coefficient.
- Il peut modifier les modules existants.
- Il peut supprimer un module devenu inutile.
- L'administrateur peut consulter la liste complète des modules.
- Il peut également associer un module à un enseignant.
(Utilisation du cas « Affecter enseignant ».)

• Gérer les notes

- L'administrateur peut gérer les notes saisies pour chaque étudiant.
- Il peut ajouter ou modifier les notes de contrôle continu (Notes CC).
- Il peut ajouter ou modifier les notes d'examen final (Notes examen).
- Il peut gérer les coefficients des modules, influençant le calcul final.
- La gestion des notes est directement liée au calcul des moyennes (relation dans le diagramme).

❖ Calculer les moyennes

- Le système permet à l'administrateur de calculer automatiquement :
 - ✓ La **moyenne par module** (en fonction des notes CC, examen et coefficient),
 - ✓ La **moyenne générale** de l'étudiant.
- Le calcul inclut également l'**attribution de la mention** (Passable, Assez bien, Bien, Très bien...), selon la moyenne obtenue.
(Lien « include » avec la mention dans le diagramme.)

❖ Générer un relevé de notes

- L'administrateur peut générer un relevé de notes complet pour un étudiant.
- Le relevé contient : les modules, les notes CC et examen, les moyennes, le coefficient et la mention.
- Le relevé peut être imprimé sous forme de fichier PDF.
- Ce cas d'utilisation utilise les moyennes déjà calculées.

❖ Affichage et consultation

✓ Afficher la liste des étudiants

- L'administrateur peut visualiser tous les étudiants enregistrés dans le système.
- Cette fonctionnalité permet également d'accéder aux détails de chaque étudiant.

✓ Afficher la liste des notes

- L'Admin peut consulter les notes de tous les étudiants, module par module.
- Cette action étend (extend) la gestion des étudiants, car elle dépend de la sélection d'un étudiant.

2-2 Diagramme de classe :

Un **diagramme de classe** est un type de diagramme UML qui modélise la structure statique d'un système. Il représente les **classes**, leurs **attributs**, leurs **méthodes** et les **relations** entre elles. Ce diagramme est essentiel pour comprendre et concevoir l'architecture d'un logiciel, car il définit les objets et leurs interactions avant l'implémentation du code.

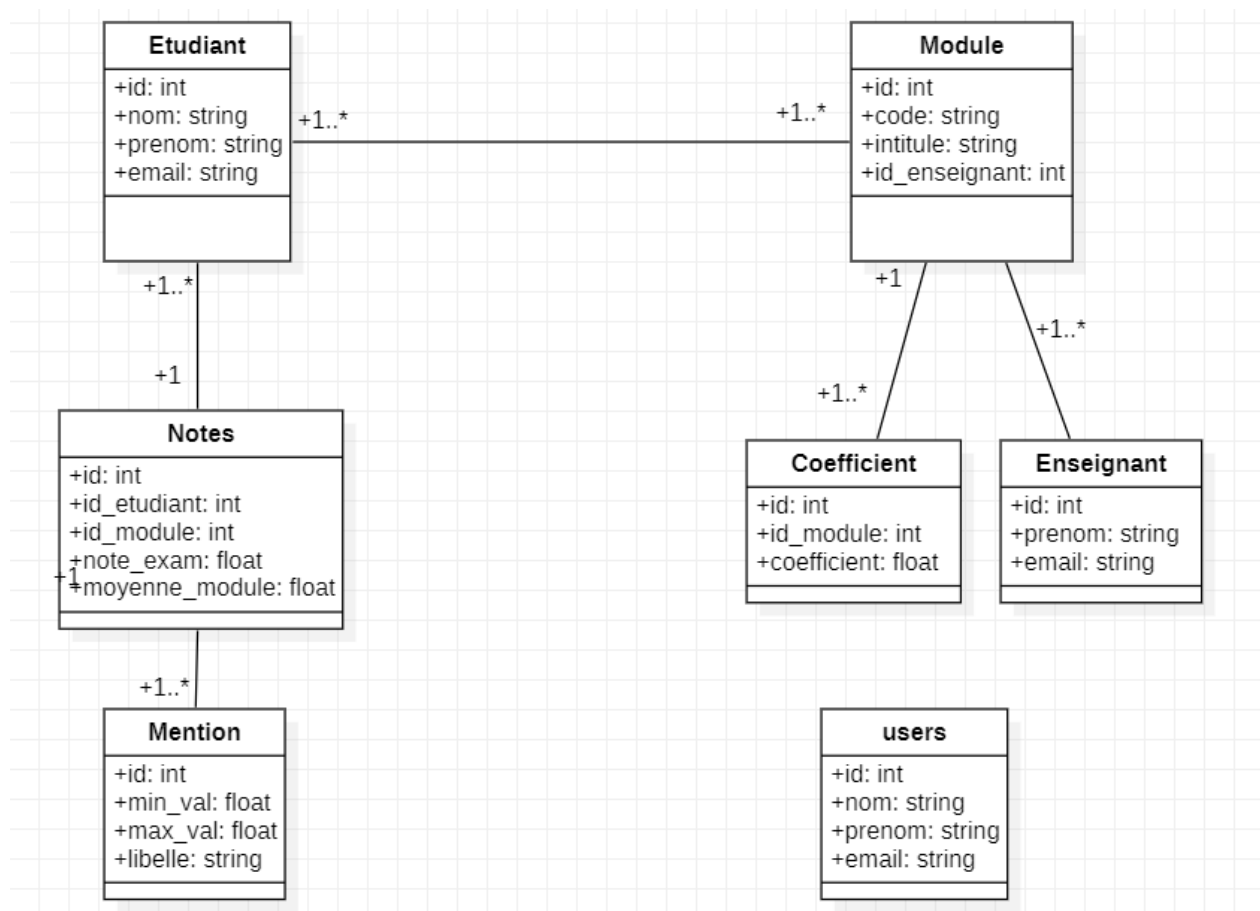


Figure 3: Diagramme de classe

Conclusion :

Dans ce chapitre, nous avons présenté les spécifications fonctionnelles et techniques du système de gestion de scolarité. L'analyse des besoins a permis de définir clairement les objectifs du projet, ses fonctionnalités principales ainsi que les contraintes associées. La phase de conception, quant à elle, a permis de modéliser le fonctionnement global du système à travers les diagrammes UML, illustrant les interactions entre l'administrateur et les différentes composantes de l'application.

CHAPITRE 3 :

Les technologies utilisées

Ce chapitre se propose de présenter, dans un premier temps, les langages de programmation utilisés pour le développement du site, et dans un second temps, les outils qui ont facilité le processus de création et de gestion de l'application web.

1- Introduction :

L'application web de gestion de scolarité utilise différentes technologies pour assurer son fonctionnement efficace. Les principales technologies utilisées sont Laravel, CSS, HTML, Blade, CSS et JavaScript.

2- Langages de programmation utilisées :

- **PHP :**

PHP (Hypertext Preprocessor) est un langage de programmation **côté serveur** principalement utilisé pour créer des sites web dynamiques et interactifs. Cela signifie que le code PHP s'exécute sur le serveur avant que la page ne soit envoyée au navigateur.



Figure 4: Logo php

- **Javascript JS :**

JavaScript est un langage de programmation côté client qui permet d'ajouter des fonctionnalités interactives et dynamiques aux pages web. Il est utilisé dans l'application de gestion des dépenses pour des tâches telles que la validation des formulaires, la manipulation des données en temps réel, l'interaction avec l'utilisateur, etc. JavaScript permet d'améliorer l'expérience utilisateur en rendant l'application plus réactive et conviviale.



Figure 5: logo JavaScript

❖ HTML (Hyper Text Markup Language)

HTML est un langage de balisage utilisé pour structurer le contenu des pages web. Il définit la structure des documents web en utilisant des éléments et des balises. Les balises HTML indiquent au navigateur comment afficher le texte, les images, les liens, les tableaux, et d'autres contenus sur une page web.



Figure 6: Logo HTML

❖ CSS (Cascading Style Sheets)

CSS est un langage de feuille de style utilisé pour décrire la présentation d'un document HTML. Il permet de définir les aspects visuels des éléments HTML, tels que les couleurs, les polices, les marges, les espacements et les mises en page. CSS aide à séparer le contenu de la présentation, rendant ainsi les pages web plus flexibles et plus faciles à maintenir.



Figure 7: Logo CSS

❖ SQL (Structured Query Language)

SQL est un langage standard utilisé pour interagir avec des bases de données relationnelles. Il permet de créer, manipuler et interroger des données stockées dans ces bases



Figure 8: Logo SQL

❖ UML :

UML (*Unified Modeling Language*) est un langage de modélisation standardisé utilisé pour visualiser, concevoir et documenter les systèmes logiciels. Il permet de représenter graphiquement la structure et le comportement d'un système à travers différents types de diagrammes (cas d'utilisation, classes, séquence, activité, etc.), facilitant ainsi la communication entre les développeurs, les analystes et les parties prenantes.



Figure 9: Logo UML

3- Les outils utilisés :

- **VS Code :**

Visual Studio Code (VS Code) est un **éditeur de code source gratuit et léger** développé par Microsoft. Il permet d'écrire et de gérer du code pour de nombreux langages (comme PHP, Python, JavaScript...), offre des **extensions**, la **coloration syntaxique**, le **debugging**, et facilite le développement grâce à son interface simple et personnalisable.

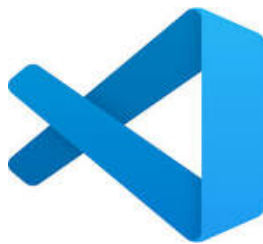


Figure 10: Logo VS Code

- **PhpMyAdmin**

phpMyAdmin est une application web open source écrite en PHP qui permet de gérer les bases de données MySQL et MariaDB via une interface graphique conviviale. Il est utilisé principalement pour effectuer des opérations de gestion des bases de données telles que la création, la modification et la suppression de bases de données, tables et enregistrements, ainsi que l'exécution de requêtes SQL.



Figure 11: Logo phpMyAdmin

❖ XAMPP

XAMPP est un ensemble de logiciels open source qui fournit une solution facile à installer pour configurer un serveur web local. Il inclut des composants essentiels pour le développement web, tels que le serveur web Apache, le système de gestion de base de données et des langages de programmation comme PHP et Perl. XAMPP est ce qui le rend idéal pour les développeurs qui souhaitent tester des applications web en local.



Figure 12: Logo XAMPP

Conclusion :

En utilisant ces technologies, L'application web de gestion de scolarité bénéficie d'un développement rapide, d'une interface utilisateur attrayante et réactive, ainsi que de fonctionnalités avancées pour gérer tous les informations scolaires.

CHAPITRE 4 :

Présentation du System

Dans ce chapitre, nous allons explorer les différentes interfaces de l'application de gestion de scolarité.

Introduction :

Dans ce chapitre, nous présentons le système développé, ses principales fonctionnalités et son interface utilisateur. Nous montrerons, à travers des captures d'écran, les différentes pages de l'application et expliquerons brièvement le rôle de chacune, afin de donner une vue d'ensemble du fonctionnement du système.

1 - Page de connexion :

Permet à l'administrateur de se connecter au système en utilisant son identifiant et mot de passe.

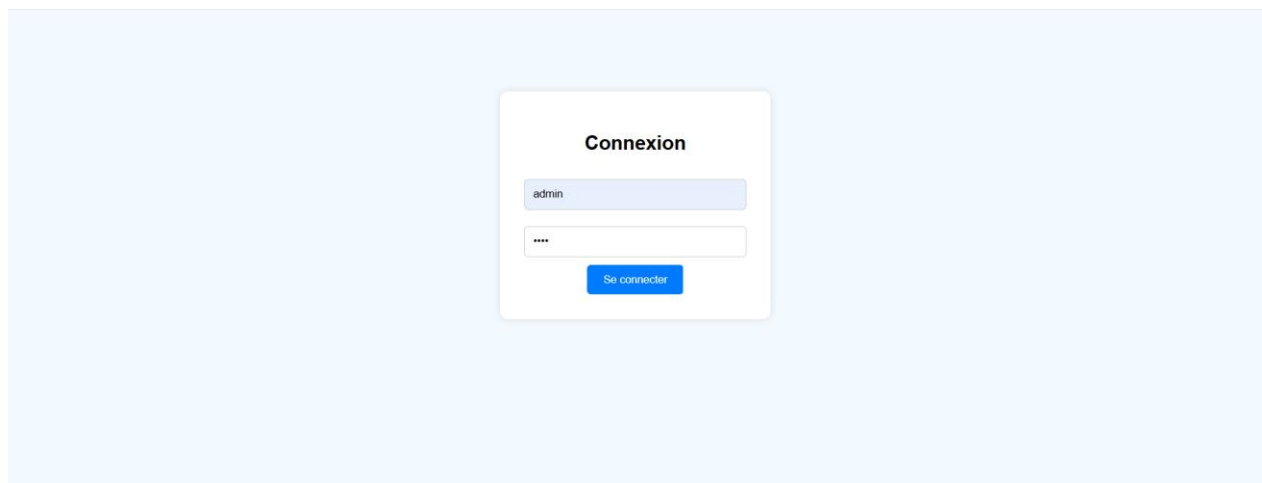


Figure 13: Page de connexion

2- Page Menu / Accueil :

Contient les liens ou boutons vers les différentes fonctionnalités du système pour gérer les étudiants, les modules, les enseignants, les notes, et les relevés etc...

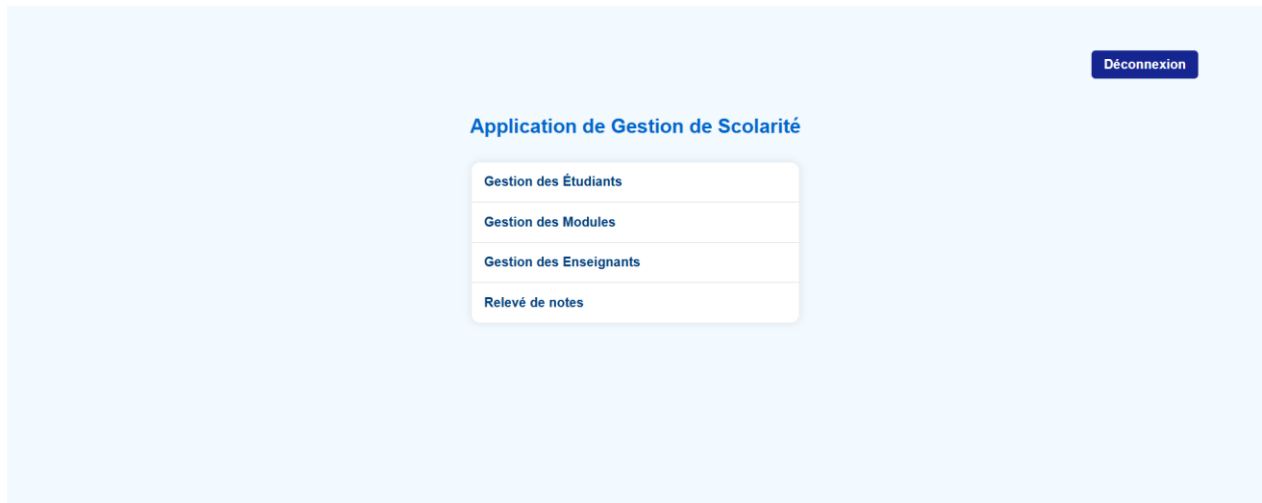
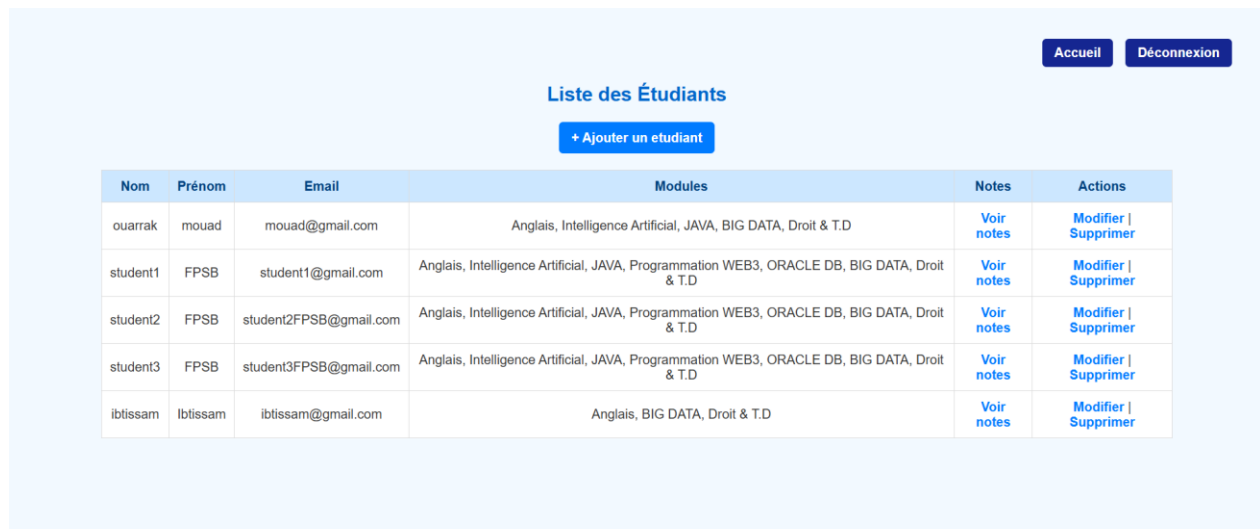


Figure 13: Page d'accueil

3- Gestion des étudiants:

Liste tous les étudiants enregistrés dans le système. L'administrateur a la possibilité d'ajouter, modifier supprimer et afficher un les informations de chaque étudiant (Nom, Prénom, Email, Modules, Notes...).

- **Liste des étudiants :**



| Nom | Prénom | Email | Modules | Notes | Actions |
|----------|----------|------------------------|--|----------------------------|--------------------------------------|
| ouarrak | mouad | mouad@gmail.com | Anglais, Intelligence Artificiel, JAVA, BIG DATA, Droit & T.D | Voir notes | Modifier Supprimer |
| student1 | FPSB | student1@gmail.com | Anglais, Intelligence Artificiel, JAVA, Programmation WEB3, ORACLE DB, BIG DATA, Droit & T.D | Voir notes | Modifier Supprimer |
| student2 | FPSB | student2FPSB@gmail.com | Anglais, Intelligence Artificiel, JAVA, Programmation WEB3, ORACLE DB, BIG DATA, Droit & T.D | Voir notes | Modifier Supprimer |
| student3 | FPSB | student3FPSB@gmail.com | Anglais, Intelligence Artificiel, JAVA, Programmation WEB3, ORACLE DB, BIG DATA, Droit & T.D | Voir notes | Modifier Supprimer |
| ibtissam | Ibtissam | ibtissam@gmail.com | Anglais, BIG DATA, Droit & T.D | Voir notes | Modifier Supprimer |

Figure 14: Liste des étudiants

- **Ajouter un étudiant :**

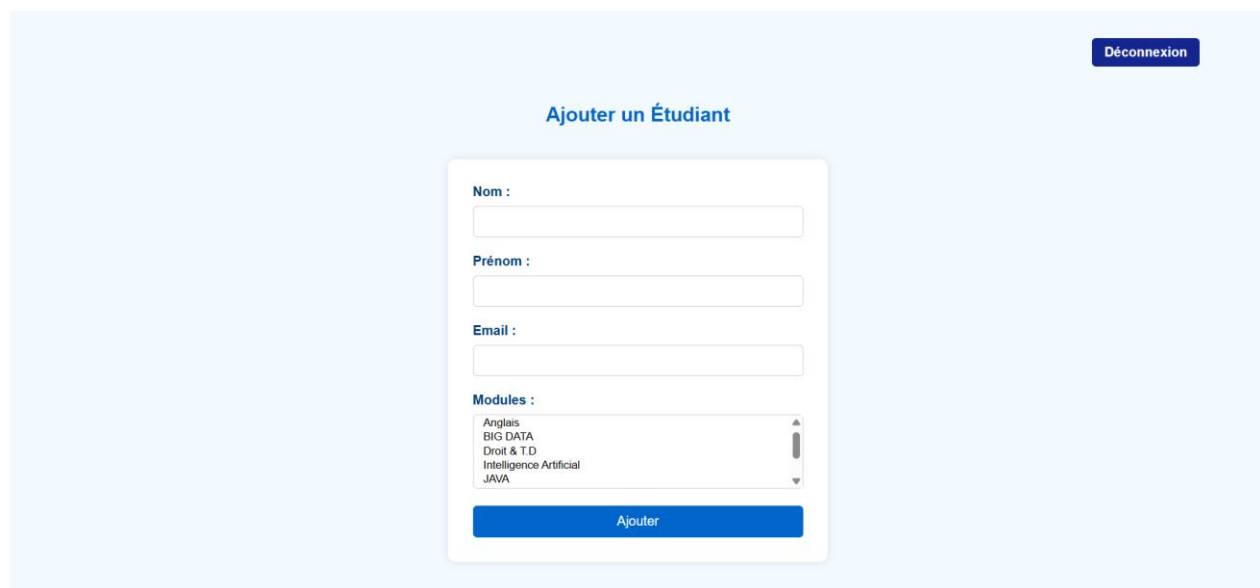


Figure 15: Ajouter un nouveau étudiant

- **Modifier un étudiant :**

Figure 16: Modifier les informations d'un étudiant

- **Supprimer l'étudiante ibtissam :**

| Nom | Prénom | Email | Modules | Notes | Actions |
|----------|--------|------------------------|--|----------------------------|--|
| ouarrak | mouad | mouad@gmail.com | Anglais, Intelligence Artificiel, JAVA, BIG DATA, Droit & T.D | Voir notes | Modifier Supprimer |
| student1 | FPSB | student1@gmail.com | Anglais, Intelligence Artificiel, JAVA, Programmation WEB3, ORACLE DB, BIG DATA, Droit & T.D | Voir notes | Modifier Supprimer |
| student2 | FPSB | student2FPSB@gmail.com | Anglais, Intelligence Artificiel, JAVA, Programmation WEB3, ORACLE DB, BIG DATA, Droit & T.D | Voir notes | Modifier Supprimer |
| student3 | FPSB | student3FPSB@gmail.com | Anglais, Intelligence Artificiel, JAVA, Programmation WEB3, ORACLE DB, BIG DATA, Droit & T.D | Voir notes | Modifier Supprimer |

Figure 17: Supprimer l'étudiante ibtissam

4- Gestion des notes d'un étudiant :

Cette section affiche les notes d'un étudiant, l'administrateur a la possibilité d'ajouter, modifier supprimer les notes de l'étudiant.

Notes de student1 FPSB

Notes enregistrées avec succès !

| Module | CC (40%) | Examen (60%) | Moyenne |
|-------------------------|----------|--------------|---------|
| Anglais | 15 | 16 | 15.00 |
| BIG DATA | 20 | 20 | 20.00 |
| Droit & T.D | 16 | 19 | 17.00 |
| Intelligence Artificiel | 18 | 19 | 18.00 |
| JAVA | 19 | 20 | 19.00 |
| ORACLE DB | 18 | 17.5 | 17.70 |
| Programmation WEB3 | 17 | 16 | 16.00 |

Enregistrer

Figure 18: Gerer les notes d'un etudiant

5- Gestion des modules :

Cette section affiche une liste de tous les modules, l'administrateur a la possibilité d'ajouter, modifier supprimer les modules.

- liste des modules :

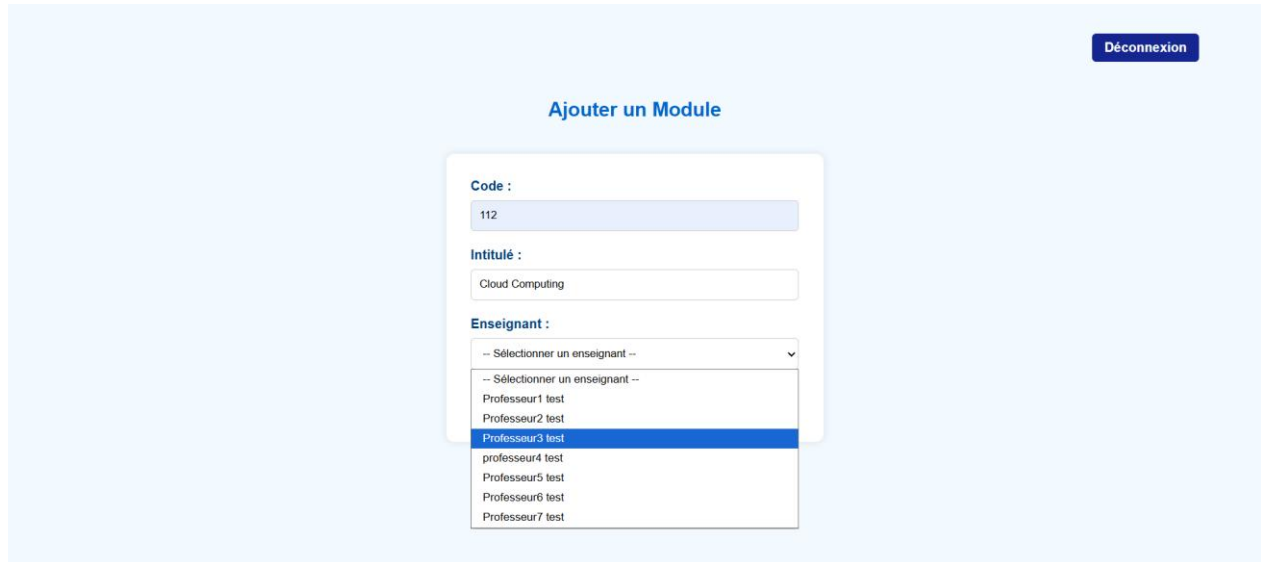
Liste des Modules

+ Ajouter un module

| Code | Intitulé | Enseignant | Actions |
|--------|-------------------------|-------------|--|
| 32 | Anglais | Professeur5 | Modifier Supprimer |
| 112 | Intelligence Artificiel | professeur4 | Modifier Supprimer |
| 134 | JAVA | Professeur6 | Modifier Supprimer |
| 972043 | Programmation WEB3 | Professeur5 | Modifier Supprimer |
| 972034 | ORACLE DB | Professeur6 | Modifier Supprimer |
| 1234 | BIG DATA | Professeur7 | Modifier Supprimer |
| 3211 | Droit & T.D | Professeur7 | Modifier Supprimer |

Figure 19: Gérer les modules

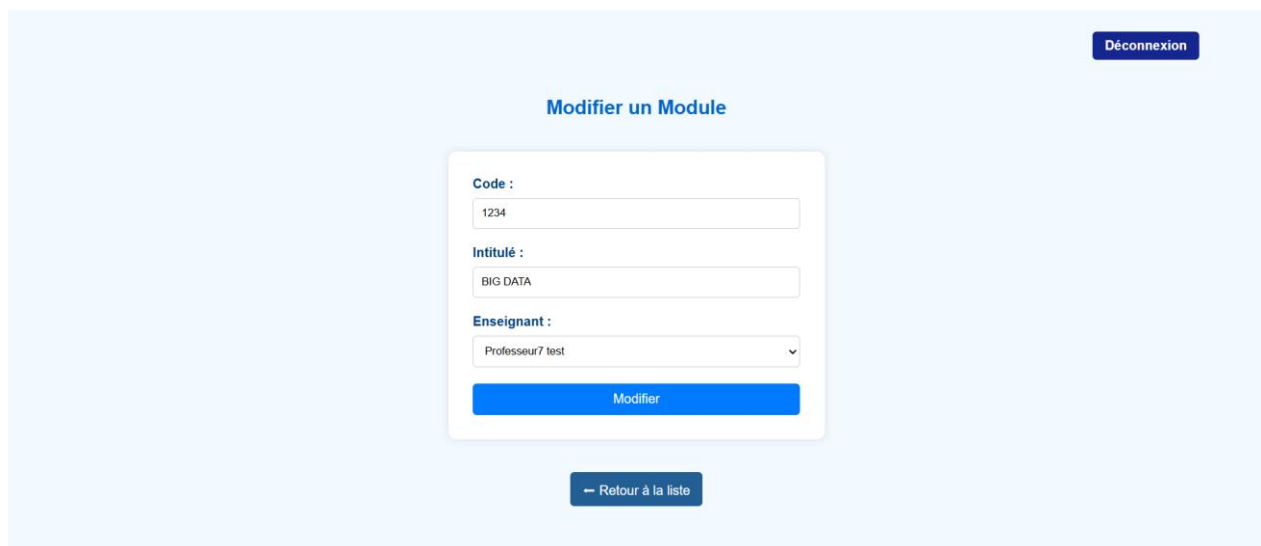
- **Ajouter un module :**



The screenshot shows a web interface for adding a module. In the top right corner, there is a blue button labeled "Déconnexion". The main heading is "Ajouter un Module". Below this, there is a form with three sections: "Code :" with a text input containing "112"; "Intitulé :" with a text input containing "Cloud Computing"; and "Enseignant :" with a dropdown menu. The dropdown menu is open, showing a list of options: "-- Sélectionner un enseignant --", "-- Sélectionner un enseignant --", "Professeur1 test", "Professeur2 test", "Professeur3 test" (which is highlighted in blue), "professeur4 test", "Professeur5 test", "Professeur6 test", and "Professeur7 test".

Figure 20: Ajouter un module

- **Modifier un module :**



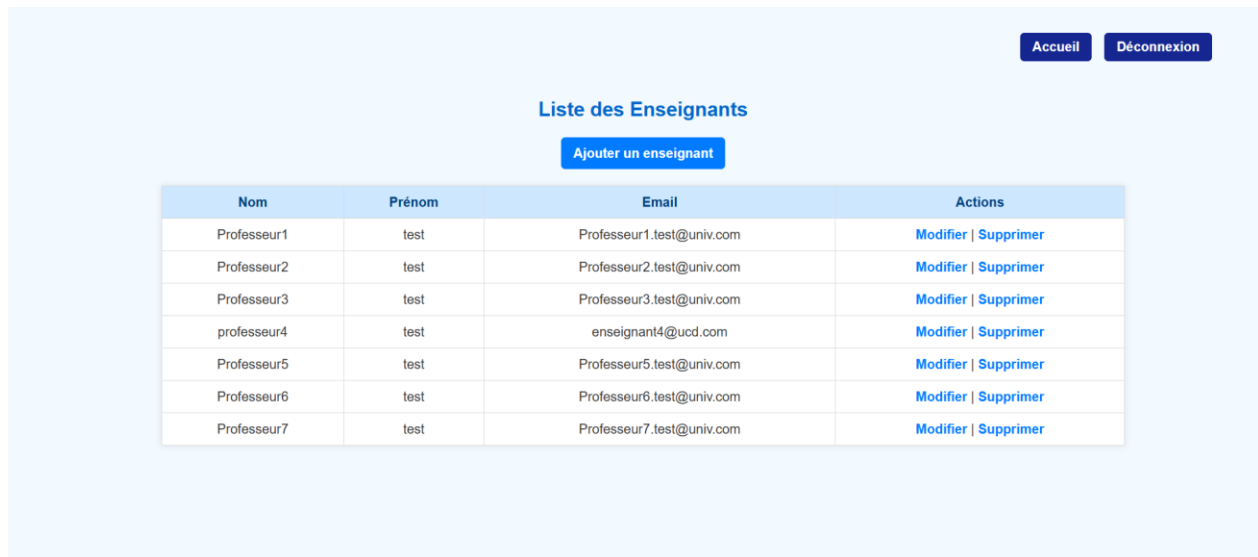
The screenshot shows a web interface for modifying a module. In the top right corner, there is a blue button labeled "Déconnexion". The main heading is "Modifier un Module". Below this, there is a form with three sections: "Code :" with a text input containing "1234"; "Intitulé :" with a text input containing "BIG DATA"; and "Enseignant :" with a dropdown menu showing "Professeur7 test". Below the form is a blue button labeled "Modifier". At the bottom center, there is a blue button labeled "← Retour à la liste".

Figure 21: Modifier un module

6- Gestion des modules :

Cette section affiche une liste de tous des enseignants, l'administrateur a la possibilité d'ajouter, modifier supprimer les enseignants.

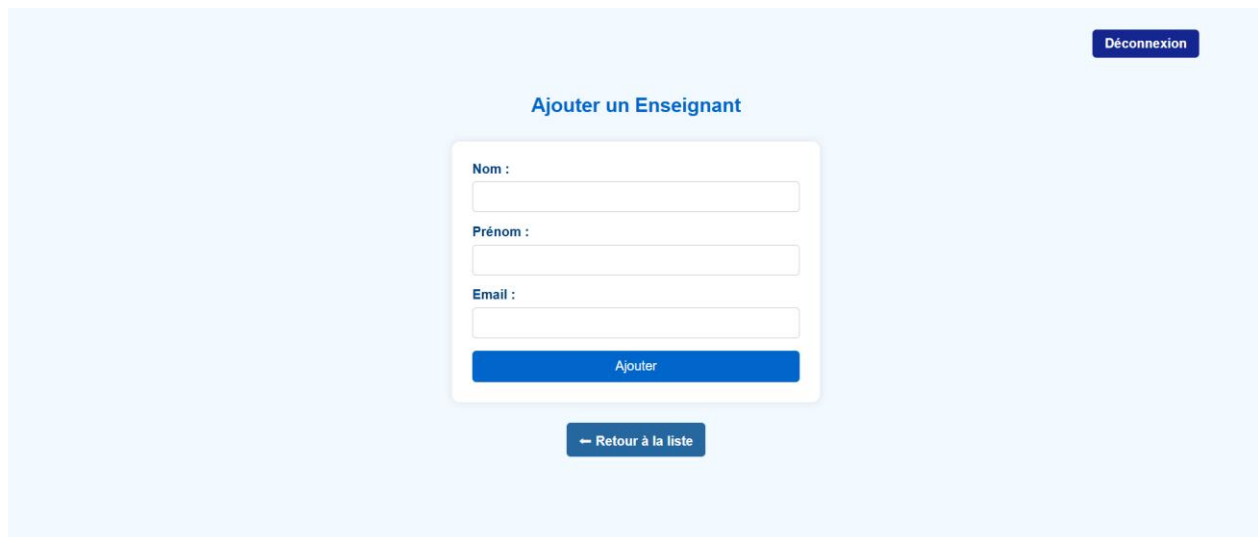
- **Liste des enseignants :**



| Nom | Prénom | Email | Actions |
|-------------|--------|---------------------------|--|
| Professeur1 | test | Professeur1.test@univ.com | Modifier Supprimer |
| Professeur2 | test | Professeur2.test@univ.com | Modifier Supprimer |
| Professeur3 | test | Professeur3.test@univ.com | Modifier Supprimer |
| professeur4 | test | enseignant4@ucd.com | Modifier Supprimer |
| Professeur5 | test | Professeur5.test@univ.com | Modifier Supprimer |
| Professeur6 | test | Professeur6.test@univ.com | Modifier Supprimer |
| Professeur7 | test | Professeur7.test@univ.com | Modifier Supprimer |

Figure 22: Liste des enseiagants

- **Ajouter un enseignant :**



Nom :

Prénom :

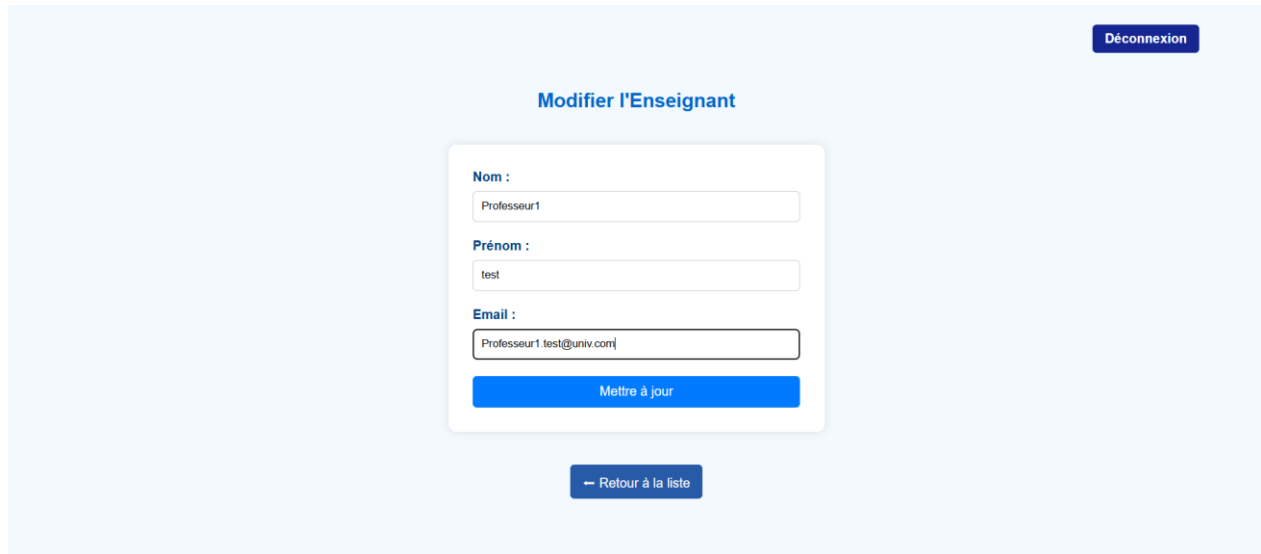
Email :

Ajouter

← Retour à la liste

Figure 23: Ajouter un etudiant

- **Modifier un enseignant :**

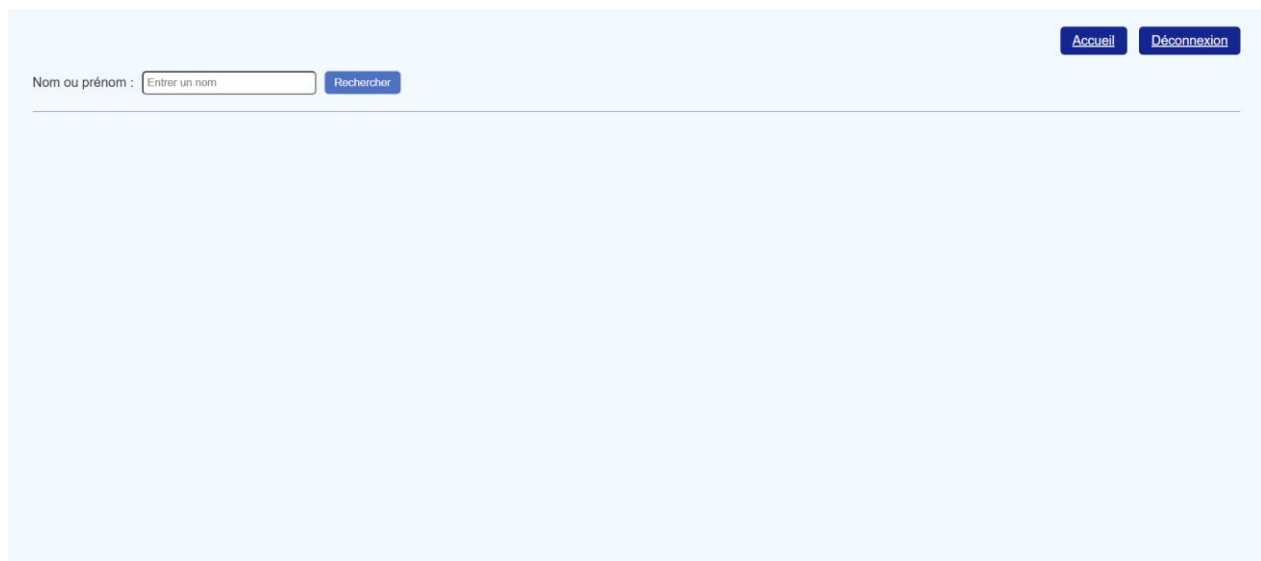


The screenshot shows a web interface for modifying a teacher's information. At the top right is a 'Déconnexion' button. The main heading is 'Modifier l'Enseignant'. Below it is a form with three input fields: 'Nom :' containing 'Professeur1', 'Prénom :' containing 'test', and 'Email :' containing 'Professeur1.test@univ.com'. A blue 'Mettre à jour' button is at the bottom of the form. Below the form is a '← Retour à la liste' button.

Figure 24: Modifier les information d'un etudiant

2- Génération de relevés de notes :

Cette section permet d'imprimer les relevés de notes des étudiants enregistré, il suffit d'écrire le nom de l'étudiant et le relevé de note va être afficher.



The screenshot shows a web interface for generating student grade reports. At the top right are 'Accueil' and 'Déconnexion' buttons. Below them is a search bar with the label 'Nom ou prénom :', an input field containing 'Entrez un nom', and a 'Rechercher' button. The rest of the page is empty.

Figure 25: Affichage de relves de notes

[Accueil](#)
[Déconnexion](#)

Nom ou prénom : [Rechercher](#)

Nom : student1
Prénom : FPSB
Email : student1@gmail.com



| Module | CC | Examen | Moyenne |
|-------------------------|----|--------|---------|
| Anglais | 15 | 16 | 15.00 |
| BIG DATA | 20 | 20 | 20.00 |
| Droit & T.D | 16 | 19 | 17.00 |
| Intelligence Artificiel | 18 | 19 | 18.00 |
| JAVA | 19 | 20 | 19.00 |
| ORACLE DB | 18 | 17.5 | 17.70 |
| Programmation WEB3 | 17 | 16 | 16.00 |

Moyenne générale : 17.53

Mention : Très bien

[Imprimer](#)

Figure 26: Affichage de relves de notes


Faculté Polydisciplinaire Sidi Bennour


Relevé de notes

Nom : student1
Prénom : FPSB
Email : student1@gmail.com

| Module | CC | Examen | Moyenne |
|-------------------------|----|--------|---------|
| Anglais | 15 | 16 | 15.00 |
| BIG DATA | 20 | 20 | 20.00 |
| Droit & T.D | 16 | 19 | 17.00 |
| Intelligence Artificiel | 18 | 19 | 18.00 |
| JAVA | 19 | 20 | 19.00 |
| ORACLE DB | 18 | 17.5 | 17.70 |
| Programmation WEB3 | 17 | 16 | 16.00 |

Moyenne générale : 17.53

Mention : Très bien

Imprimer

1 feuille de papier

Destination: [Microsoft Print to PDF](#)

Pages: [Toutes](#)

Mise en page: [Portrait](#)

Couleur: [Couleur](#)

[Plus de paramètres](#)

[Imprimer](#)
[Annuler](#)

Figure 27: Imprimer le relevés de note d'un étudiant

Conclusion générale :

Ce projet a permis de concevoir et de réaliser une application de gestion des étudiants simple et fonctionnelle. À travers ce travail, nous avons appliqué les connaissances acquises en programmation web, notamment l'utilisation de PHP, d'une base de données et d'un environnement de développement adapté.

Le système développé permet de gérer efficacement les étudiants à travers plusieurs fonctionnalités essentielles telles que l'authentification, l'ajout, la modification, la suppression et la consultation des informations. L'interface est claire et facile à utiliser, ce qui facilite la manipulation du système par l'utilisateur.

Ce projet a également contribué à renforcer nos compétences techniques et notre compréhension du fonctionnement des applications web. En perspective, le système pourrait être amélioré par l'ajout de nouvelles fonctionnalités comme la gestion des rôles utilisateurs, l'amélioration de la sécurité ou l'intégration de statistiques.

En conclusion, ce projet représente une expérience enrichissante qui a permis de mettre en pratique la théorie tout en développant une solution répondant aux besoins de gestion des étudiants.

Webographie :

- **Mozilla Developer Network (MDN) – Documentation HTML, CSS, JavaScript :**

<https://developer.mozilla.org/fr/>

- **Documentation framework php**

<https://laravel.com/docs/12.x>

- **Documentation de Bootstrap**

<https://getbootstrap.com/>

- **Scribd**

<https://fr.scribd.com>

- **Php my admin**

<https://www.phpmyadmin.net/>