



Nom De l'entreprise : FEATWAY

# RAPPORT DE PÉRIODE DE FORMATION EN MILIEU PROFESSIONNEL

---

Développement d'une application web de  
gestion des bulletins.

---

*Nom D'établissement : Maryse Conde*

*Elaboré par : Abdelhanine Chaima*

*Période de formation : DU 18/11/2024 À 10/01/2025*

Année scolaire : 2024/2025

# Remerciement

Avant de commencer la présentation de ce rapport, j'en profite pour remercier du fond du cœur toute personne qui a contribué directement ou indirectement à la réalisation de cet ouvrage.

Je tiens à remercier Mr Abdelhanine Adel de m'avoir accueillie dans son entreprise durant ce stage. Il m'a permis de découvrir le monde des applications web. Je tiens également à remercier Monsieur Yahia Abdelwahed qui m'a aidé à faire ce choix d'application, et qui a été plus qu'un tuteur. Il m'a bien encadrée, m'a vraiment soutenue et m'a sincèrement orientée en me proposant à chaque fois de nouvelles suggestions. Ses encouragements constants, ses conseils pertinents et la bonne organisation du travail m'ont beaucoup facilité la tâche.

Mes remerciements sont également adressés à mes professeurs pour m'avoir donné la théorie et la pratique pertinentes et ciblées. Enfin, je voudrais remercier tous ceux qui m'ont aidé à préparer ce rapport de stage surtout ma famille.

# Table des matières

<b>Remerciement</b>	<b>I</b>
<b>Introduction</b>	<b>1</b>
<b>1 Présentation de l'entreprise</b>	<b>2</b>
1.1 Historique de la société . . . . .	2
1.2 Fiche d'identité de l'entreprise . . . . .	2
1.3 Organisation de l'entreprise . . . . .	3
1.4 Situation géographique . . . . .	3
1.5 Plan de l'entreprise / atelier de maintenance . . . . .	4
<b>2 Développement d'une activité</b>	<b>6</b>
2.1 Présentation du support d'activité . . . . .	6
2.1.1 STS : Spring Tool Suite . . . . .	6
2.1.2 Visual Studio Code . . . . .	6
2.1.3 Xampp . . . . .	6
2.1.4 Postman . . . . .	7
2.1.5 Spring boot . . . . .	7
2.2 Problématique . . . . .	7
2.3 Descriptif détaillé de l'activité . . . . .	7
2.3.1 Présentation . . . . .	7
2.3.2 Création du modèle de données . . . . .	8
2.3.3 Installation et configuration de Spring Boot : . . . . .	9
2.3.4 Developpement Backend : . . . . .	11
2.3.4.1 Creation des entités . . . . .	11

2.3.4.2	Création des contrôleurs . . . . .	13
2.3.4.3	Creation des Services . . . . .	15
2.3.4.4	Creation des Repository . . . . .	17
2.3.5	Developpement Frontdend : . . . . .	19
2.3.6	Interfaces Utilisateur . . . . .	23
<b>Conclusion</b>		<b>27</b>

# Table des figures

1.1	TOUR MAINE MONTPARNASSE . . . . .	3
1.2	Bureau . . . . .	4
1.3	Accueil . . . . .	5
1.4	salle de réunion . . . . .	5
2.1	démarrage de Xampp . . . . .	8
2.2	PHP MyAdmin . . . . .	8
2.3	Nouveau Projet Spring Boot . . . . .	9
2.4	Ajout de dependance . . . . .	10
2.5	Entité Utilisateur . . . . .	11
2.6	Entité Note . . . . .	12
2.7	Controller Note . . . . .	13
2.8	Controlleur Auth . . . . .	14
2.9	Service Note . . . . .	15
2.10	Service Utilisateur . . . . .	16
2.11	Repository Note . . . . .	17
2.12	Repository Utilisateur . . . . .	18
2.13	login.php . . . . .	19
2.14	prof_dashboard.php . . . . .	20
2.15	eleve_dashboard.php . . . . .	21
2.16	ajout_eleve.php . . . . .	22
2.17	Page de connection . . . . .	23
2.18	Tableau de Bord enseignant . . . . .	24

---

*TABLE DES FIGURES*

---

2.19 Tableau de Bord enseignant . . . . .	24
2.20 Tableau de bord élève . . . . .	25
2.21 ajout élève . . . . .	26

# Liste des tableaux

1.1 Fiche d'identité de l'entreprise . . . . .	2
--	---

# Introduction

Je m'appelle Abdelhanine Chaima. Je suis une étudiante dans l'établissement de Maryse Conde dans la classe TSNRISCA . Pour l'obtention de diplôme de Périodes de formation en milieu professionnel (PFMP), j'ai fait un stage du 18 Novembre 2024 jusqu'au 10 Janvier 2025 .

Ce stage présente un parcours très riche d'informations et d'expériences afin que je puisse utiliser mes compétences dans un cadre professionnel et bien encadré. J'ai fait mon stage au sein de l'entreprise FeatWay, située à l'adresse TOUR MAINE MONTPARNASSE, 14ème étage, Bureau 1408. Au cours duquel, je crée une application web de gestion des bulletins a l'aide de framework SpringBoot

# Présentation de l'entreprise

## 1.1 Historique de la société

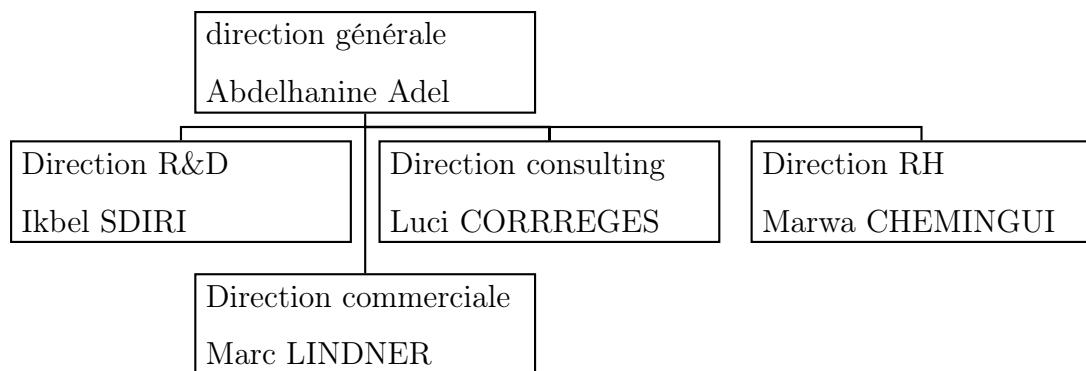
Société par actions simplifiées, est active depuis 4 ans. Elle a été créée par M. AbdelHanine Adel en 17/03/2020. Elle est spécialisée dans le secteur d'activité du conseil en systèmes et logiciels informatiques.

## 1.2 Fiche d'identité de l'entreprise

Forme juridique	Société par action simplifiées (SAS)
Capital	50 000 €
Chiffre d'affaires	4 Millions d'euros
Nombre de salariés	43
Secteur et domaines d'activités	technologie de l'informatique

**TABLE 1.1** – Fiche d'identité de l'entreprise

## 1.3 Organisation de l'entreprise



## 1.4 Situation géographique

TOUR MAINE MONTPARNASSE



FIGURE 1.1 – TOUR MAINE MONTPARNASSE

Featway se situe au 14ème étage , Bureau 1408

## 1.5 Plan de l'entreprise / atelier de maintenance



FIGURE 1.2 – Bureau



**FIGURE 1.3 – Accueil**



**FIGURE 1.4 – salle de réunion**

# Chapitre 2

## Développement d'une activité

### 2.1 Présentation du support d'activité

#### 2.1.1 STS : Spring Tool Suite

STS est un environnement de développement basé sur Eclipse qui est personnalisé pour le développement d'applications Spring. Il fournit un environnement prêt à l'emploi pour implémenter, déboguer, exécuter et déployer vos applications.

#### 2.1.2 Visual Studio Code

Visual Studio Code est un éditeur de code source léger, mais puissant, qui s'exécute sur votre bureau et est disponible pour Windows, macOS et Linux. Il est livré avec une prise en charge intégrée de JavaScript, TypeScript et Node.js et dispose d'un riche écosystème d'extensions pour d'autres langages et environnements d'exécution (tels que C++, C, Java, Python, PHP, Go, .NET).[10] Version utilisée : 1.86.0

#### 2.1.3 Xampp

XAMPP est un logiciel de développement web qui permet de créer et de gérer des sites internet localement sur son propre ordinateur. Ce logiciel, qui est disponible gratuitement, est largement utilisé par les développeurs pour développer leurs sites web et leurs applications.[2]

### 2.1.4 Postman

Postman est une plate-forme API pour la création et l'utilisation d'API. Postman simplifie chaque étape du cycle de vie des API et rationalise la collaboration afin que vous puissiez créer de meilleures API, plus rapidement.

### 2.1.5 Spring boot

Java Spring Framework (Spring Framework) est une infrastructure open source d'entreprise couramment utilisée qui permet de créer des applications autonomes de production qui fonctionnent sur la machine virtuelle Java (JVM).

## 2.2 Problématique

Comment concevoir une application web intuitive et efficace pour faciliter la gestion, le suivi et la consultation des bulletins scolaires, tout en répondant aux besoins des utilisateurs (enseignants et élèves) en termes de simplicité, d'accessibilité et de confidentialité des données ?

## 2.3 Descriptif détaillé de l'activité

### 2.3.1 Présentation

Dans ce tutoriel, nous allons apprendre à développer une application web de suivi des bulletins scolaires en utilisant le framework Spring Boot. Cette application permettra de gérer les bulletins scolaires des étudiants, d'enregistrer les notes, de calculer les moyennes, et de générer des rapports de performances.

Grâce à ce projet, vous pourrez mettre en pratique plusieurs concepts liés au développement backend, tels que la gestion des bases de données, la création d'API REST, et l'authentification des utilisateurs.

- La gestion des utilisateurs ( enseignants, étudiants).
- L'enregistrement et la mise à jour des données des bulletins (notes, matières).
- La consultation des bulletins et la visualisation des performances.
- La génération automatique des moyennes et des rapports.

### 2.3.2 Création du modèle de données

Pour commencer, démarrez XAMPP, ce qui nous permettra d'accéder à phpMyAdmin afin de créer notre base de données.

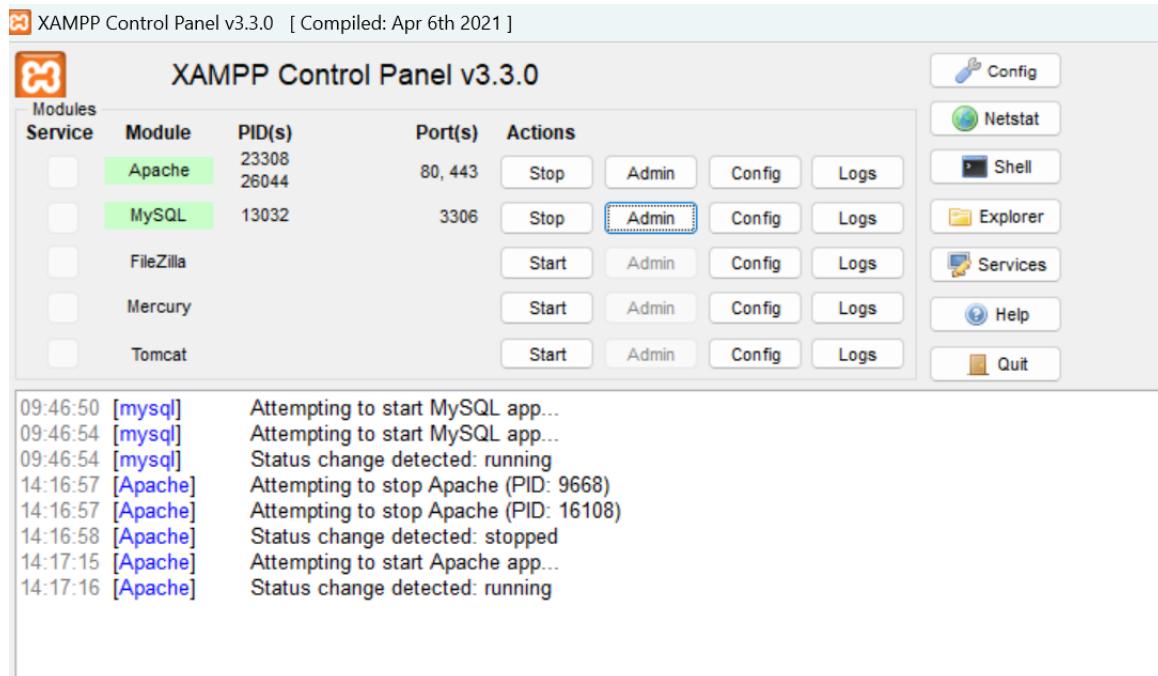


FIGURE 2.1 – démarrage de Xampp

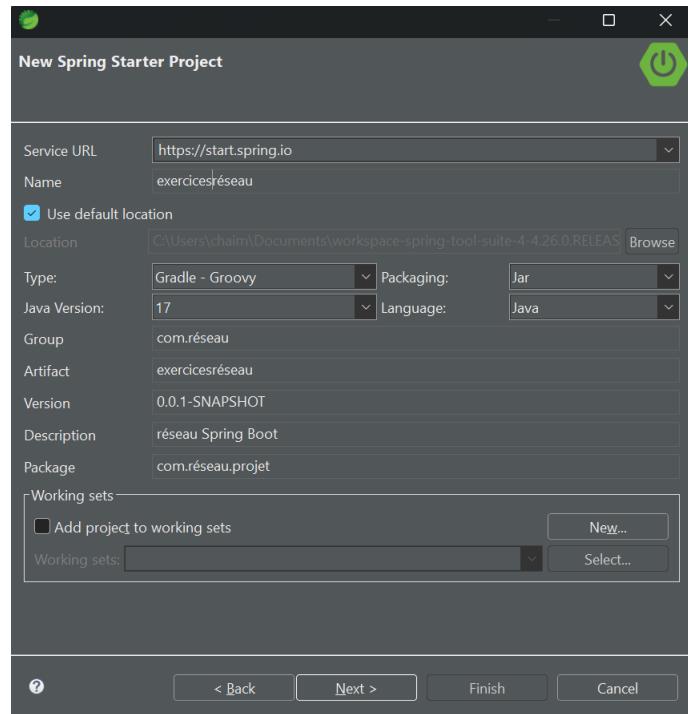
	id	email	password	role
<input type="checkbox"/>	1	y@gmail.com	Password	PROFESSEUR
<input type="checkbox"/>	2	chayma@gmail.com	pass	ELEVE
<input type="checkbox"/>	3	yaha@gmail.com	pass	ELEVE
<input type="checkbox"/>	6	sara@gmail.com	pass	ELEVE
<input type="checkbox"/>	7	zak@gmail.com	pass	ELEVE
<input type="checkbox"/>	8	adame@a.com	pass	ELEVE
<input type="checkbox"/>	9	sana@kk.com	pass	ELEVE
<input type="checkbox"/>	10	amine@gmail.com	pass	ELEVE
<input type="checkbox"/>	11	uu@mm.cc	azz	ELEVE
<input type="checkbox"/>	12	aaa	aaa	ELEVE

FIGURE 2.2 – PHP MyAdmin

Cette figure montre l'interface de phpMyAdmin, un outil de gestion de bases de données MySQL. Il permet aux utilisateurs de créer, visualiser, modifier et administrer des bases de données à travers une interface graphique conviviale.

### 2.3.3 Installation et configuration de Spring Boot :

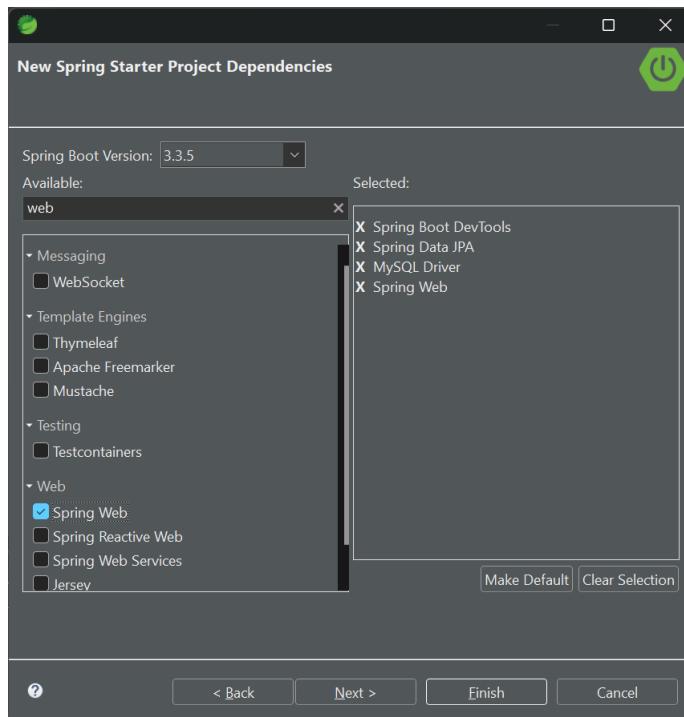
- Créez un nouveau projet Spring Boot



**FIGURE 2.3 – Nouveau Projet Spring Boot**

L'image présente l'étape initiale de création d'un projet Spring Boot à l'aide de Spring Tool Suite (STS). Cette étape est cruciale pour configurer notre projet backend avec les dépendances et les paramètres appropriés.

- Ajoutez les dépendances nécessaires



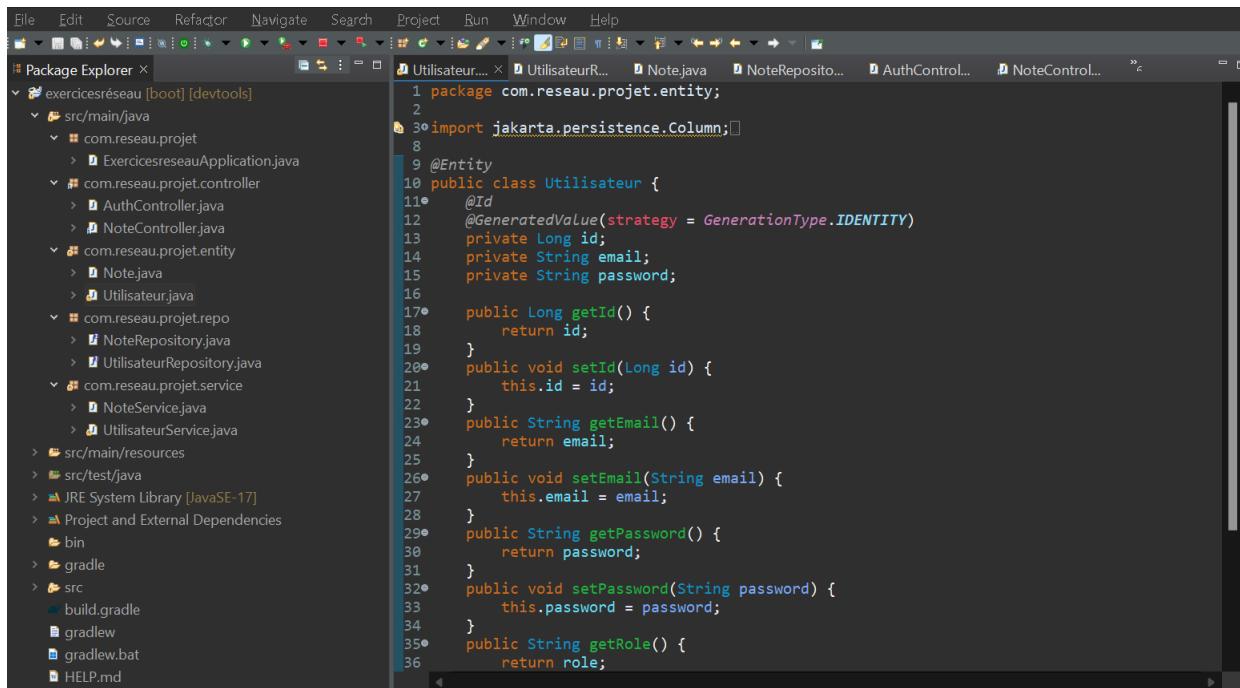
**FIGURE 2.4 – Ajout de dépendance**

Cette figure montre l'écran où les dépendances nécessaires, comme Spring Web, JPA et MySQL, sont ajoutées au notre projet Spring Boot. Ces dépendances nous permet de construire des applications web robustes et connectées à une base de données.

## 2.3.4 Développement Backend :

### 2.3.4.1 Création des entités

— Entité Utilisateur



The screenshot shows the Eclipse IDE interface with the 'Utilisateur....' file open in the code editor. The code defines a Java class 'Utilisateur' with annotations for JPA persistence. The code editor has syntax highlighting for Java and annotations.

```

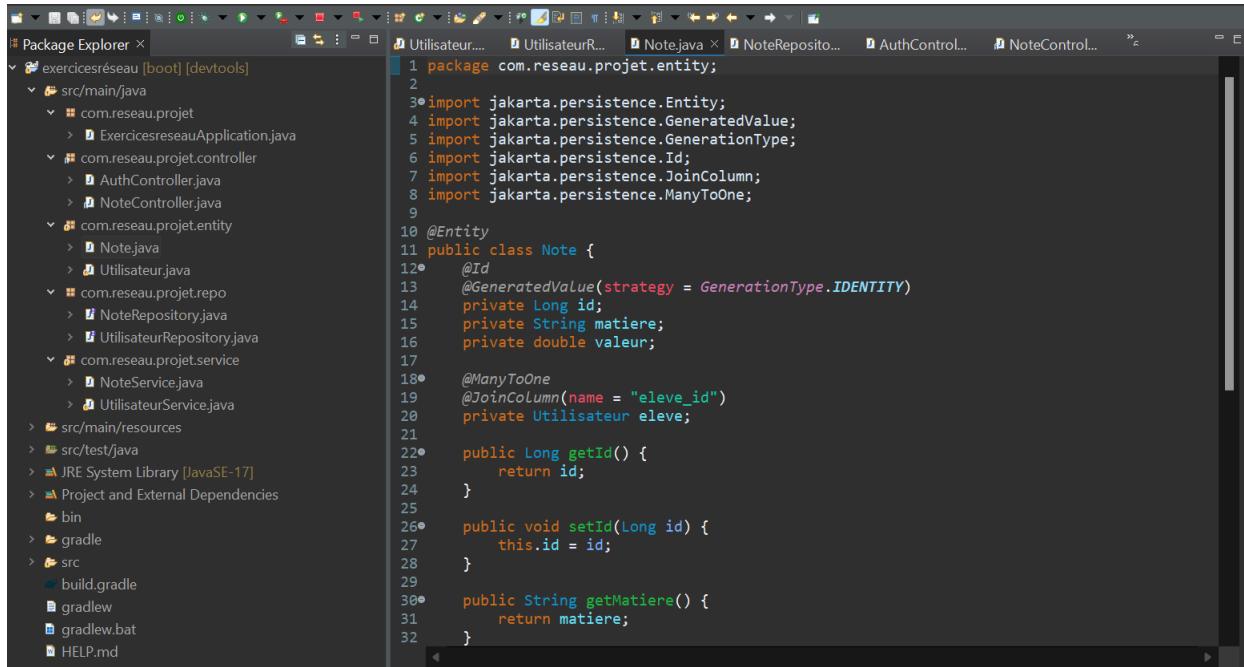
1 package com.reseau.projet.entity;
2
3 import jakarta.persistence.Column;
4
5 @Entity
6 public class Utilisateur {
7     @Id
8     @GeneratedValue(strategy = GenerationType.IDENTITY)
9     private Long id;
10    private String email;
11    private String password;
12
13    public Long getId() {
14        return id;
15    }
16
17    public void setId(Long id) {
18        this.id = id;
19    }
20
21    public String getEmail() {
22        return email;
23    }
24
25    public void setEmail(String email) {
26        this.email = email;
27    }
28
29    public String getPassword() {
30        return password;
31    }
32
33    public void setPassword(String password) {
34        this.password = password;
35    }
36    public String getRole() {
37        return role;
38    }
39}

```

FIGURE 2.5 – Entité Utilisateur

Cette figure représente la classe Utilisateur dans le projet Spring Boot. Elle définit la structure de l'entité utilisateur, incluant les attributs tels que l'identifiant, le nom, le prénom et l'adresse email. Chaque attribut est annoté avec des annotations JPA comme @Id et @GeneratedValue pour la gestion automatique des identifiants primaires.

## — Entité Note



```

1 package com.reseau.projet.entity;
2
3 import jakarta.persistence.Entity;
4 import jakarta.persistence.GeneratedValue;
5 import jakarta.persistence.GenerationType;
6 import jakarta.persistence.Id;
7 import jakarta.persistence.JoinColumn;
8 import jakarta.persistence.ManyToOne;
9
10 @Entity
11 public class Note {
12     @Id
13     @GeneratedValue(strategy = GenerationType.IDENTITY)
14     private Long id;
15     private String matiere;
16     private double valeur;
17
18     @ManyToOne
19     @JoinColumn(name = "eleve_id")
20     private Utilisateur eleve;
21
22     public Long getId() {
23         return id;
24     }
25
26     public void setId(Long id) {
27         this.id = id;
28     }
29
30     public String getMatiere() {
31         return matiere;
32     }
33 }

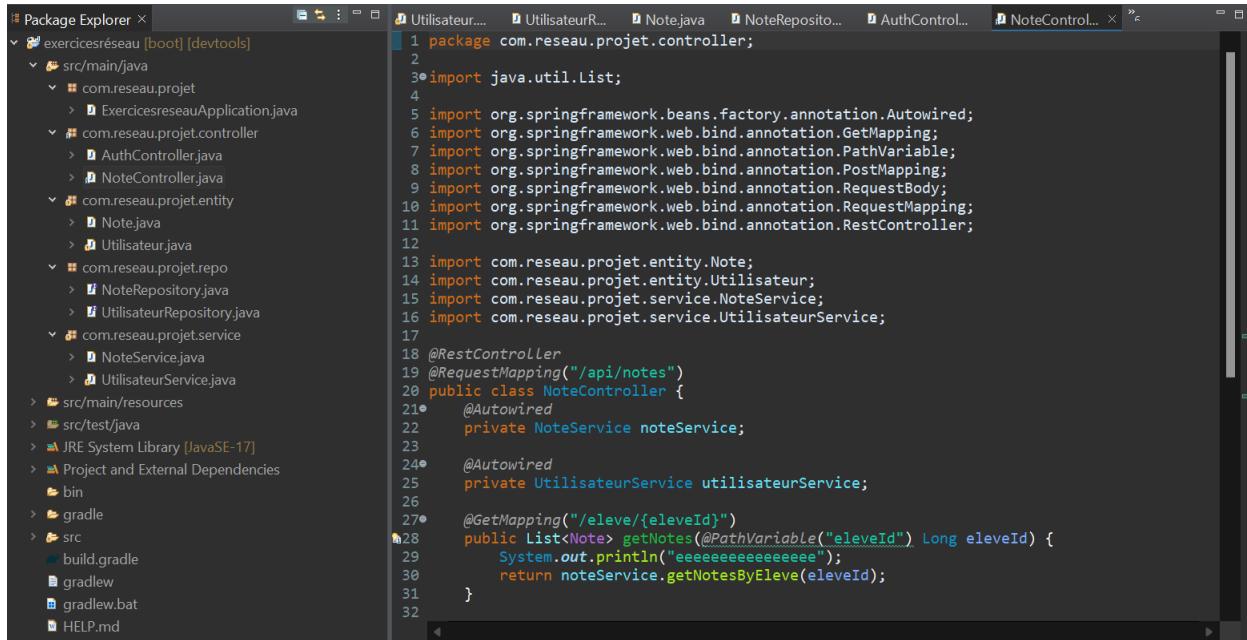
```

**FIGURE 2.6 – Entité Note**

L'image montre la classe Note, qui représente les notes attribuées aux élèves dans le système. Elle contient des champs tels que l'identifiant de la note, la matière et la valeur de la note. Ces entités sont ensuite utilisées pour créer la base de données à l'aide d'ORM (Object Relational Mapping).

### 2.3.4.2 Cr éation des contrôleurs

#### — Controller Note



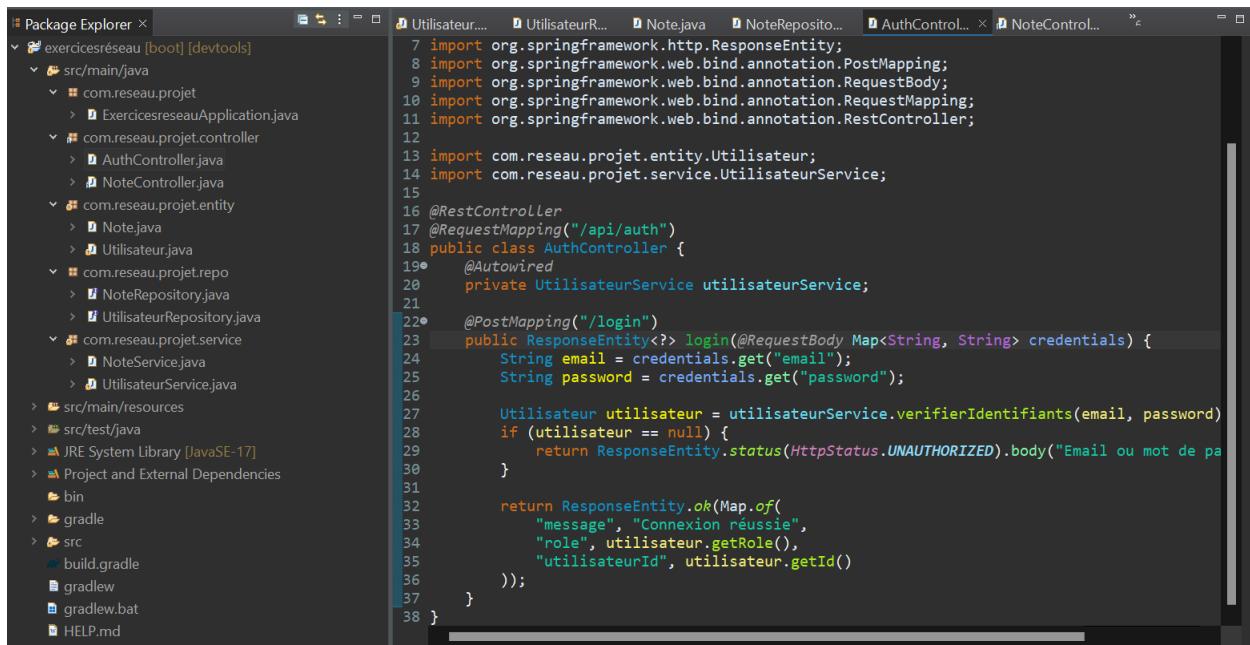
```

1 package com.reseau.projet.controller;
2
3 import java.util.List;
4
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.web.bind.annotation.GetMapping;
7 import org.springframework.web.bind.annotation.PathVariable;
8 import org.springframework.web.bind.annotation.PostMapping;
9 import org.springframework.web.bind.annotation.RequestBody;
10 import org.springframework.web.bind.annotation.RequestMapping;
11 import org.springframework.web.bind.annotation.RestController;
12
13 import com.reseau.projet.entity.Note;
14 import com.reseau.projet.entity.Utilisateur;
15 import com.reseau.projet.service.NoteService;
16 import com.reseau.projet.service.UtilisateurService;
17
18 @RestController
19 @RequestMapping("/api/notes")
20 public class NoteController {
21     @Autowired
22     private NoteService noteService;
23
24     @Autowired
25     private UtilisateurService utilisateurService;
26
27     @GetMapping("/eleve/{eleveId}")
28     public List<Note> getNotes(@PathVariable("eleveId") Long eleveId) {
29         System.out.println("eeeeeeeeeeeeeee");
30         return noteService.getNotesByEleve(eleveId);
31     }
32 }
```

FIGURE 2.7 – Controller Note

Cette figure illustre le contrôleur NoteController, qui expose plusieurs endpoints RESTful permettant d'interagir avec les notes. Les méthodes présentées permettent d'ajouter, de mettre à jour, de supprimer et de consulter les notes via des requêtes HTTP (GET, POST, PUT, DELETE).

### — Controller Auth



The screenshot shows the Eclipse IDE interface with the 'Package Explorer' view on the left and the 'AuthController.java' file in the 'Text Editor' view on the right. The 'Package Explorer' shows the project structure for 'exercicesréseau' with packages like 'com.reseau.projet' and 'com.reseau.projet.repo'. The 'Text Editor' displays the Java code for the 'AuthController' class, which contains a single endpoint for user login.

```

1 import org.springframework.http.ResponseEntity;
2 import org.springframework.web.bind.annotation.PostMapping;
3 import org.springframework.web.bind.annotation.RequestBody;
4 import org.springframework.web.bind.annotation.RequestMapping;
5 import org.springframework.web.bind.annotation.RestController;
6
7 import com.reseau.projet.entity.Utilisateur;
8 import com.reseau.projet.service.UtilisateurService;
9
10 @RestController
11 @RequestMapping("/api/auth")
12 public class AuthController {
13     @Autowired
14     private UtilisateurService utilisateurService;
15
16     @PostMapping("/login")
17     public ResponseEntity<?> login(@RequestBody Map<String, String> credentials) {
18         String email = credentials.get("email");
19         String password = credentials.get("password");
20
21         Utilisateur utilisateur = utilisateurService.verifierIdentifiants(email, password);
22         if (utilisateur == null) {
23             return ResponseEntity.status(HttpStatus.UNAUTHORIZED).body("Email ou mot de passe incorrect");
24         }
25
26         return ResponseEntity.ok(Map.of(
27             "message", "Connexion réussie",
28             "role", utilisateur.getRole(),
29             "utilisateurId", utilisateur.getId()
30         ));
31     }
32 }

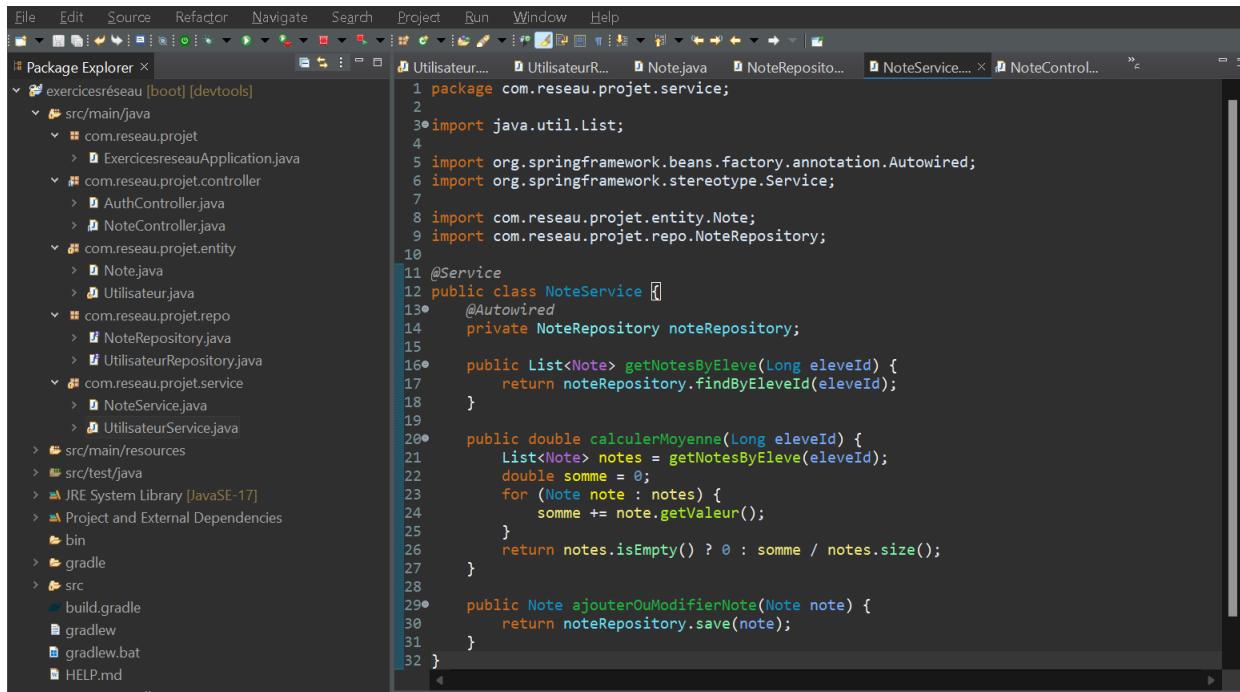
```

**FIGURE 2.8 – Contrôleur Auth**

L'image montre le contrôleur `AuthController`, qui gère l'authentification des utilisateurs. Il expose des endpoints permettant la connexion et la gestion des sessions utilisateur. Ce composant est essentiel pour sécuriser l'application et contrôler l'accès aux différentes fonctionnalités.

### 2.3.4.3 Creation des Services

#### — Service Note



The screenshot shows a Java code editor within an IDE. The left pane displays the project structure under 'Package Explorer'. The 'src/main/java' package contains several sub-packages and files: 'com.reseau.projet' (with Application.java, AuthController.java, NoteController.java, Note.java, Utilisateur.java), 'com.reseau.projet.repo' (with NoteRepository.java, UtilisateurRepository.java), and 'com.reseau.projet.service' (with NoteService.java, UtilisateurService.java). The right pane shows the content of NoteService.java. The code defines a service class 'NoteService' that interacts with a 'NoteRepository' to perform operations like getting notes by student ID and calculating average grades.

```

1 package com.reseau.projet.service;
2
3 import java.util.List;
4
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.stereotype.Service;
7
8 import com.reseau.projet.entity.Note;
9 import com.reseau.projet.repo.NoteRepository;
10
11 @Service
12 public class NoteService {
13     @Autowired
14     private NoteRepository noteRepository;
15
16     public List<Note> getNotesByEleve(Long eleveId) {
17         return noteRepository.findById(eleveId);
18     }
19
20     public double calculerMoyenne(Long eleveId) {
21         List<Note> notes = getNotesByEleve(eleveId);
22         double somme = 0;
23         for (Note note : notes) {
24             somme += note.getValeur();
25         }
26         return notes.isEmpty() ? 0 : somme / notes.size();
27     }
28
29     public Note ajouterOuModifierNote(Note note) {
30         return noteRepository.save(note);
31     }
32 }

```

**FIGURE 2.9 – Service Note**

Cette figure représente la classe NoteService, qui implémente la logique métier liée à la gestion des notes. Le service encapsule les opérations sur la base de données et fournit des méthodes réutilisables pour les contrôleurs.

### — Service Utilisateur

```

File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer Utilisateur... UtilisateurR... Note.java NoteReposito... AuthControl... NoteControl...
exercicesréseau [boot] [devtools]
src/main/java
com.reseau.projet
ExercicesreseauApplication.java
com.reseau.projet.controller
AuthController.java
NoteController.java
com.reseau.projet.entity
Note.java
Utilisateur.java
com.reseau.projet.repo
NoteRepository.java
UtilisateurRepository.java
com.reseau.projet.service
NoteService.java
UtilisateurService.java
src/main/resources
src/test/java
JRE System Library [JavaSE-17]
Project and External Dependencies
bin
gradle
src
build.gradle
gradlew
gradlew.bat
HELP.md

23
24● @Autowired
25     private UtilisateurService utilisateurService;
26
27● @GetMapping("/eleve/{eleveId}")
28     public List<Note> getNotes(@PathVariable("eleveId") Long eleveId) {
29         System.out.println("eeeeeeeeeeeeeee");
30         return noteService.getNotesByEleve(eleveId);
31     }
32
33● @GetMapping("/eleve/{eleveId}/moyenne")
34     public double getMoyenne(@PathVariable("eleveId") Long eleveId) {
35         System.out.println("11111111");
36         return noteService.calculerMoyenne(eleveId);
37     }
38
39● @GetMapping("/eleves")
40     public List<Utilisateur> getEleves() {
41         return utilisateurService.getEleves();
42     }
43
44● @PostMapping
45     public Note ajouterOuModifierNote(@RequestBody Note note) {
46         return noteService.ajouterOuModifierNote(note);
47     }
48
49● @PostMapping("/user")
50     public Utilisateur ajouterOuModifierEleve(@RequestBody Utilisateur user) {
51         return utilisateurService.ajouterOuModifierEleve(user);
52     }
53 }
54

```

**FIGURE 2.10 – Service Utilisateur**

L'image montre la classe `UtilisateurService`, qui contient les méthodes métier pour gérer les utilisateurs. Ce service est responsable de l'ajout, de la modification, de la suppression et de la récupération des utilisateurs.

### 2.3.4.4 Creation des Repository

#### — Repository Note

```

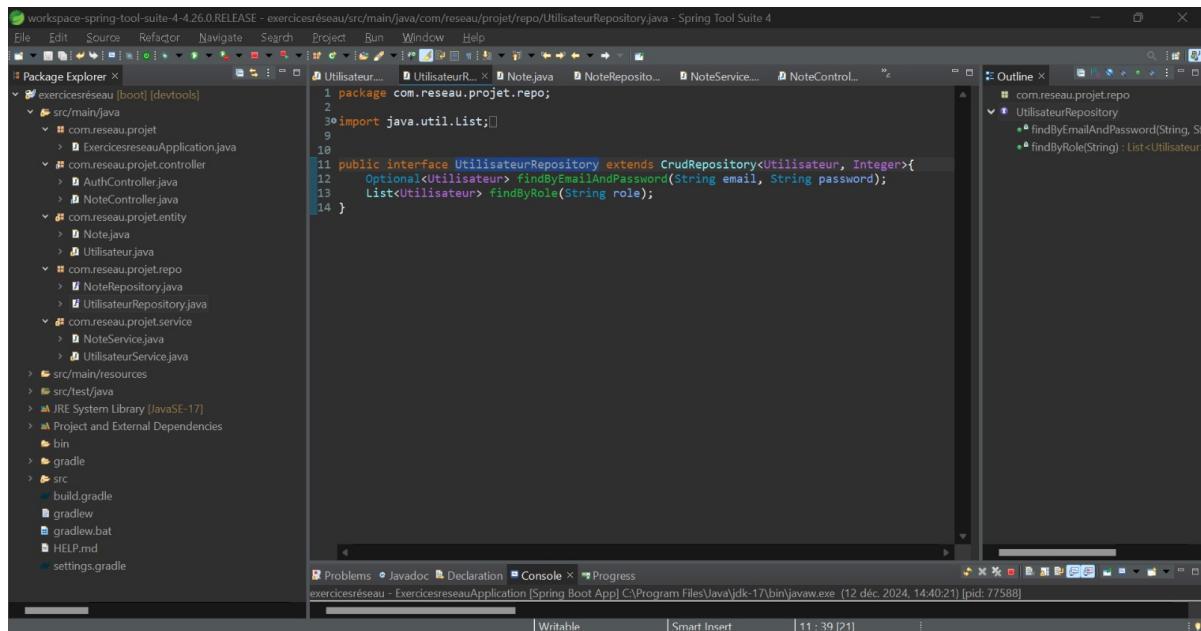
1 package com.reseau.projet.repo;
2
3 import java.util.List;
4
5 import org.springframework.data.repository.CrudRepository;
6
7 import com.reseau.projet.entity.Note;
8
9 public interface NoteRepository extends CrudRepository<Note, Long> {
10     List<Note> findByEleveId(Long eleveId);
11 }

```

**FIGURE 2.11** – Repository Note

Cette figure présente le repository NoteRepository, qui étend l’interface JpaRepository de Spring Data JPA. Cela permet de réaliser facilement les opérations CRUD (Create, Read, Update, Delete) sur l’entité Note sans écrire de code SQL explicite.

### — Repository Utilisateur



```
workspace-spring-tool-suite-4-4.26.0.RELEASE - exercicesréseau/src/main/java/com/reseau/projet/repo/UtilisateurRepository.java - Spring Tool Suite 4
File Edit Sources Refactor Navigate Search Project Run Window Help
Package Explorer X
exercicesréseau [boot] [devtools]
src/main/java
com.reseau.projet
ExercicesréseauApplication.java
com.reseau.projet.controller
AuthController.java
NoteController.java
com.reseau.projet.entity
Note.java
Utilisateur.java
com.reseau.projet.repo
NoteRepository.java
UtilisateurRepository.java
com.reseau.projet.service
NoteService.java
UtilisateurService.java
src/main/resources
src/test/java
JRE System Library [JavaSE-17]
Project and External Dependencies
bin
gradle
src
build.gradle
gradlew
gradlew.bat
HELP.md
settings.gradle
Utilisateur... UtilisateurR... Note.java NoteReposito... NoteService... NoteControl...
Utilisateur...
1 package com.reseau.projet.repo;
2
3 import java.util.List;
4
5
6 public interface UtilisateurRepository extends CrudRepository<Utilisateur, Integer>{
7     Optional<Utilisateur> findByEmailAndPassword(String email, String password);
8     List<Utilisateur> findByRole(String role);
9 }
10
11
12
13
14 }
```

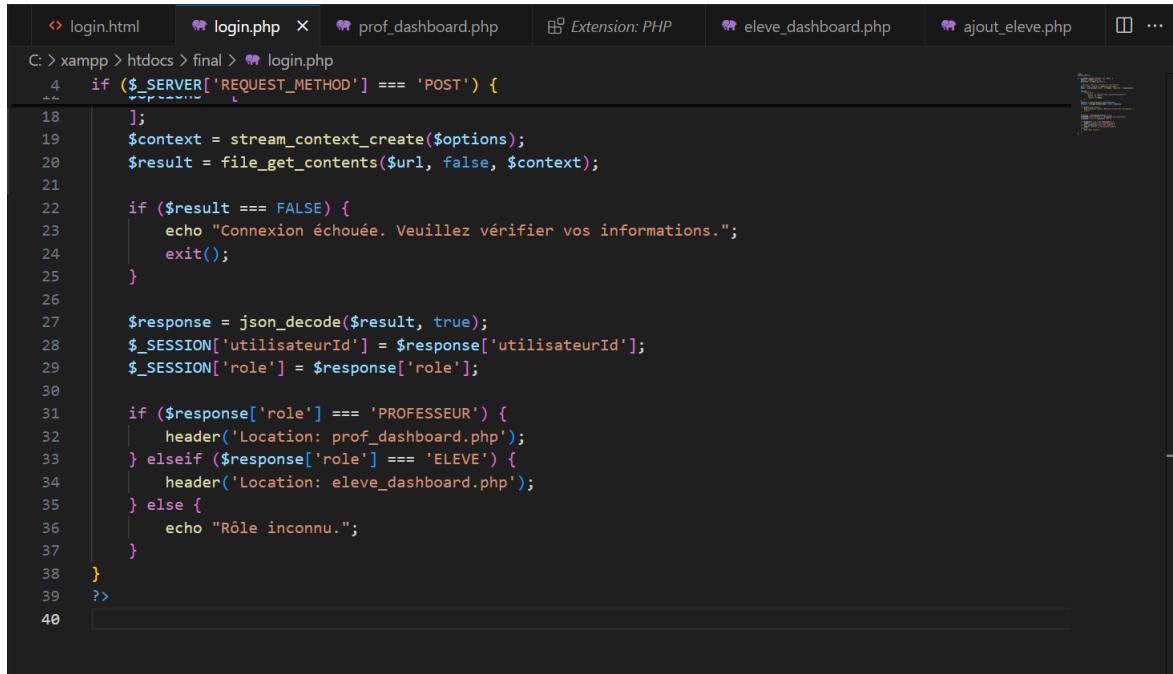
The screenshot shows the Spring Tool Suite 4 interface. The central area displays the code for the `UtilisateurRepository.java` file. The code defines an interface that extends the `CrudRepository` for the `Utilisateur` entity. It includes two custom methods: `findByEmailAndPassword` and `findByRole`. The `Outline` view on the right side shows the interface definition with its two methods. The `Package Explorer` view on the left lists the project structure, including packages like `com.reseau.projet`, `com.reseau.projet.repo`, and `com.reseau.projet.service`, along with various Java files and configuration files.

**FIGURE 2.12 – Repository Utilisateur**

L'image montre le repository UtilisateurRepository, qui permet de manipuler l'entité Utilisateur. Il hérite de l'interface JpaRepository et peut être étendu avec des requêtes personnalisées si nécessaire.

### 2.3.5 Développement Frontend :

- La page de Connection



```

C: > xampp > htdocs > final > login.php
Extension: PHP
ajout_eleve.php

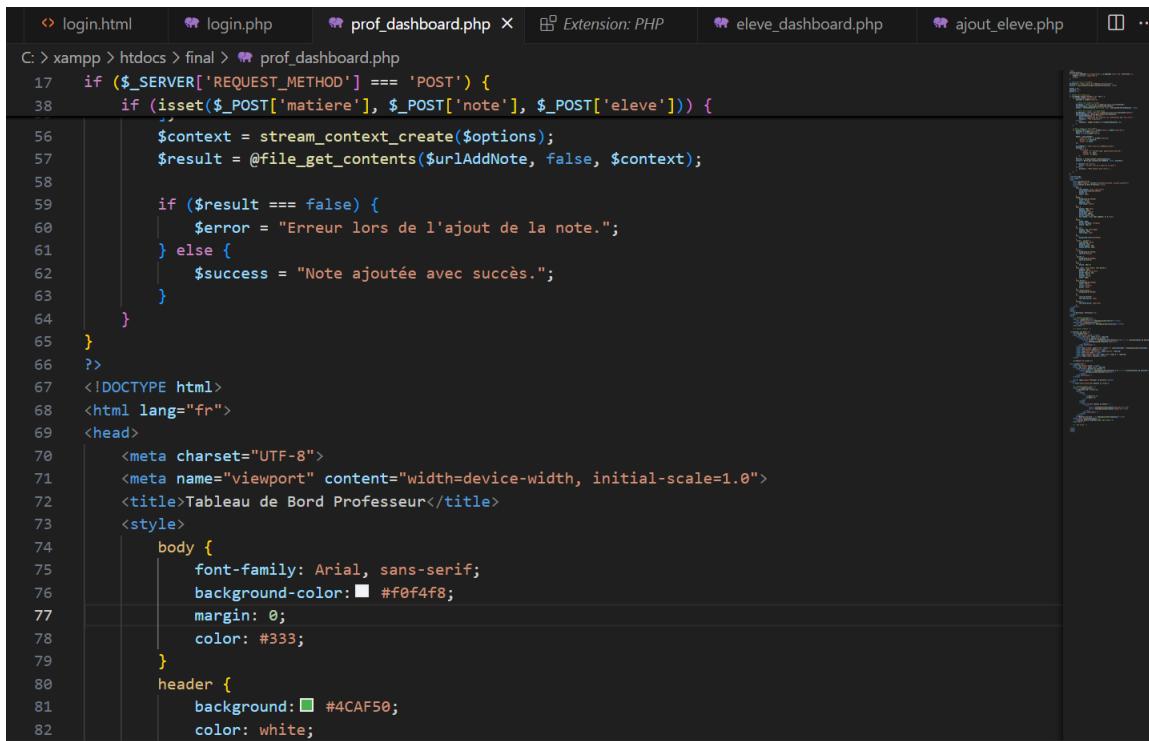
4 if ($_SERVER['REQUEST_METHOD'] === 'POST') {
5     $options = [
6         'http' => [
7             'method' => 'POST',
8             'header' => [
9                 'Content-Type' => 'application/json'
10            ],
11            'content' => json_encode($_POST)
12        ]
13    ];
14    $context = stream_context_create($options);
15    $result = file_get_contents($url, false, $context);
16
17    if ($result === FALSE) {
18        echo "Connexion échouée. Veuillez vérifier vos informations.";
19        exit();
20    }
21
22    $response = json_decode($result, true);
23    $_SESSION['utilisateurId'] = $response['utilisateurId'];
24    $_SESSION['role'] = $response['role'];
25
26    if ($response['role'] === 'PROFESSEUR') {
27        header('Location: prof_dashboard.php');
28    } elseif ($response['role'] === 'ELEVE') {
29        header('Location: eleve_dashboard.php');
30    } else {
31        echo "Rôle inconnu.";
32    }
33
34 }
35 ?>
36
37
38
39
40

```

**FIGURE 2.13 – login.php**

Cette figure présente le code source de la page login.php. Ce fichier PHP contient le formulaire de connexion ainsi que le traitement des données saisies. Il vérifie les identifiants fournis par l'utilisateur en les comparant à ceux enregistrés dans la base de données. Si les informations sont correctes, l'utilisateur est redirigé vers son tableau de bord ; sinon, un message d'erreur est affiché.

## — Tableau de Bord enseignant



```

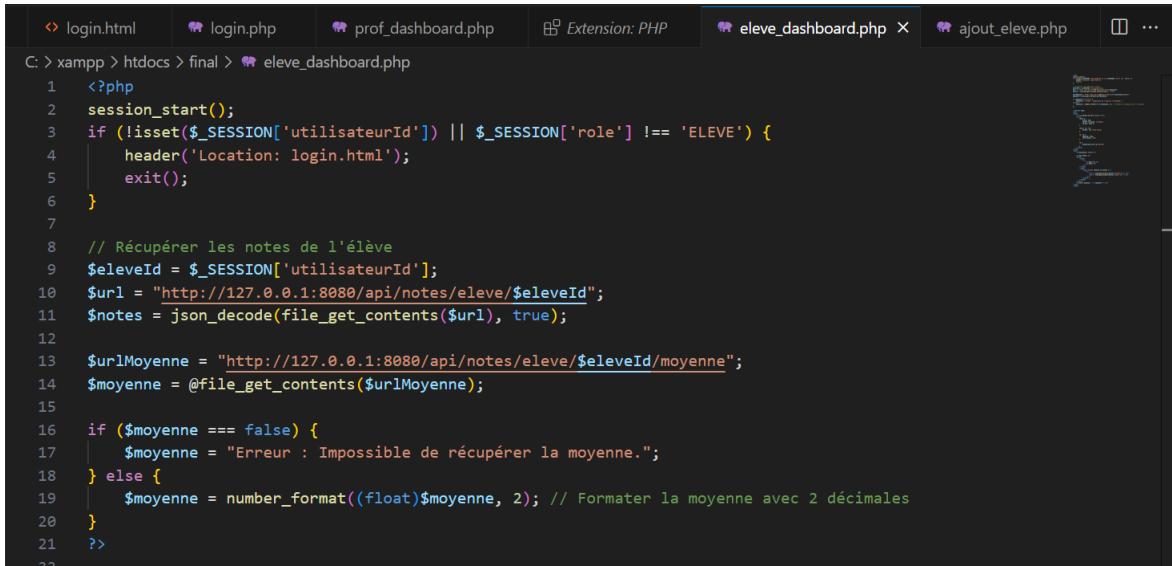
C: > xampp > htdocs > final > prof_dashboard.php
17  if ($_SERVER['REQUEST_METHOD'] === 'POST') {
38      if (isset($_POST['matiere'], $_POST['note'], $_POST['eleve'])) {
56          $context = stream_context_create($options);
57          $result = @file_get_contents($urlAddNote, false, $context);
58
59          if ($result === false) {
60              $error = "Erreur lors de l'ajout de la note.";
61          } else {
62              $success = "Note ajoutée avec succès.";
63          }
64      }
65  }
66 ?>
67 <!DOCTYPE html>
68 <html lang="fr">
69 <head>
70     <meta charset="UTF-8">
71     <meta name="viewport" content="width=device-width, initial-scale=1.0">
72     <title>Tableau de Bord Professeur</title>
73     <style>
74         body {
75             font-family: Arial, sans-serif;
76             background-color: #f0f4f8;
77             margin: 0;
78             color: #333;
79         }
80         header {
81             background: #4CAF50;
82             color: white;

```

**FIGURE 2.14 – prof\_dashboard.php**

Cette figure montre le code source de la page prof\_dashboard.php, qui constitue le tableau de bord principal des enseignants. Ce fichier PHP affiche les notes des élèves et fournit des fonctionnalités de gestion des bulletins, comme l'ajout et la modification des notes. Il inclut également des contrôles d'accès pour s'assurer que seules les personnes autorisées (enseignants) peuvent y accéder.

## — Tableau de Bord élève



```

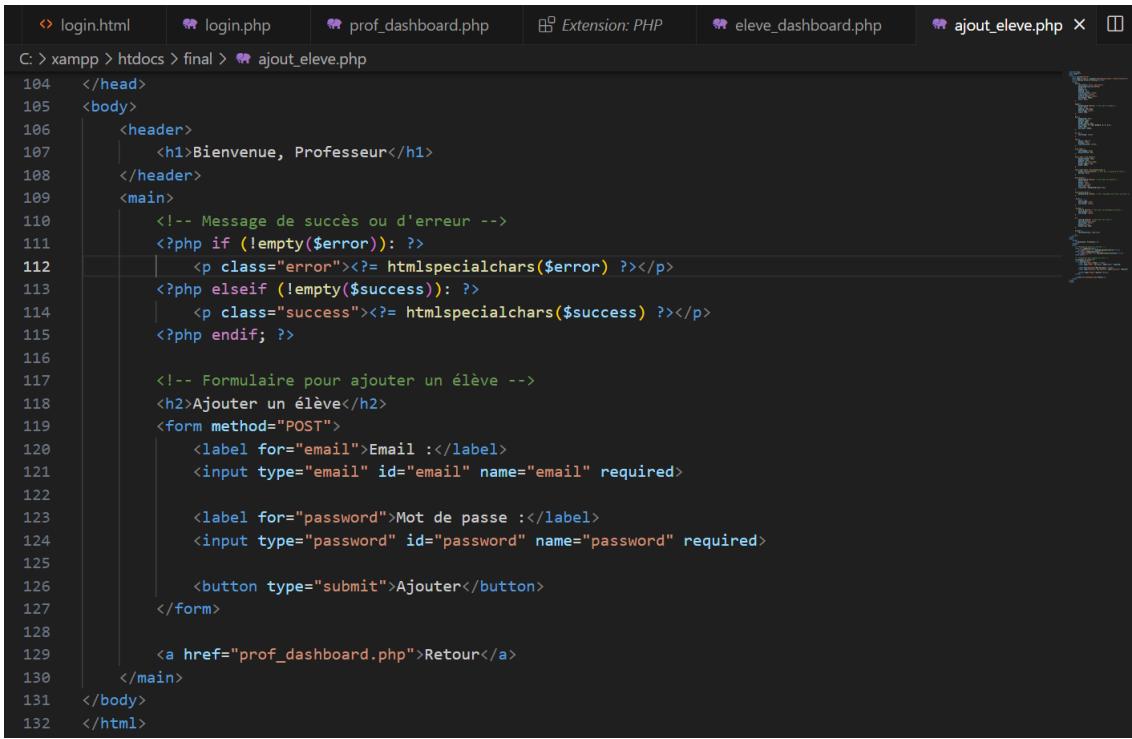
C: > xampp > htdocs > final > eleve_dashboard.php
1  <?php
2  session_start();
3  if (!isset($_SESSION['utilisateurId']) || $_SESSION['role'] !== 'ELEVE') {
4      header('Location: login.html');
5      exit();
6  }
7
8  // Récupérer les notes de l'élève
9  $eleveId = $_SESSION['utilisateurId'];
10 $url = "http://127.0.0.1:8080/api/notes/eleve/$eleveId";
11 $notes = json_decode(file_get_contents($url), true);
12
13 $urlMoyenne = "http://127.0.0.1:8080/api/notes/eleve/$eleveId/moyenne";
14 $moyenne = @file_get_contents($urlMoyenne);
15
16 if ($moyenne === false) {
17     $moyenne = "Erreur : Impossible de récupérer la moyenne.";
18 } else {
19     $moyenne = number_format((float)$moyenne, 2); // Formater la moyenne avec 2 décimales
20 }
21 ?>
22

```

**FIGURE 2.15 – eleve\_dashboard.php**

Cette figure présente le code source de la page eleve\_dashboard.php, qui est le tableau de bord destiné aux élèves. Ce script PHP affiche les bulletins de l'élève connecté, les détails des notes par matière et la moyenne générale. Il intègre des requêtes à la base de données pour récupérer dynamiquement les informations à afficher.

## — Page d'ajout d'élève



```

C: > xampp > htdocs > final > ajout_eleve.php
104   </head>
105   <body>
106     <header>
107       <h1>Bienvenue, Professeur</h1>
108     </header>
109     <main>
110       <!-- Message de succès ou d'erreur -->
111       <?php if (!empty($error)): ?>
112         <p class="error"><?= htmlspecialchars($error) ?></p>
113       <?php elseif (!empty($success)): ?>
114         <p class="success"><?= htmlspecialchars($success) ?></p>
115       <?php endif; ?>
116
117       <!-- Formulaire pour ajouter un élève -->
118       <h2>Ajouter un élève</h2>
119       <form method="POST">
120         <label for="email">Email :</label>
121         <input type="email" id="email" name="email" required>
122
123         <label for="password">Mot de passe :</label>
124         <input type="password" id="password" name="password" required>
125
126         <button type="submit">Ajouter</button>
127       </form>
128
129       <a href="prof_dashboard.php">Retour</a>
130     </main>
131   </body>
132 </html>

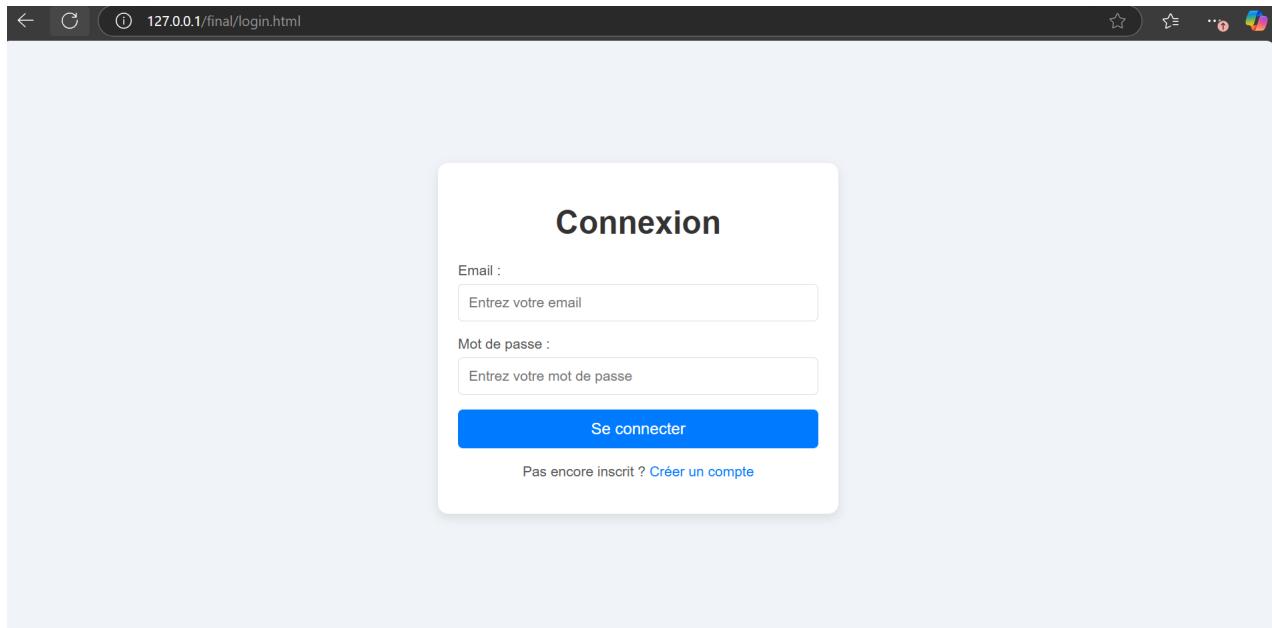
```

**FIGURE 2.16 – ajout\_eleve.php**

Cette figure illustre le code PHP de la page ajout\_eleve.php, qui permet l'ajout d'un nouvel élève dans le système. Ce fichier contient un formulaire pour saisir les informations de l'élève et traite les données saisies en les insérant dans la base de données via une requête SQL. Des contrôles de validation sont également inclus pour s'assurer que les champs obligatoires sont correctement remplis.

### **2.3.6 Interfaces Utilisateur**

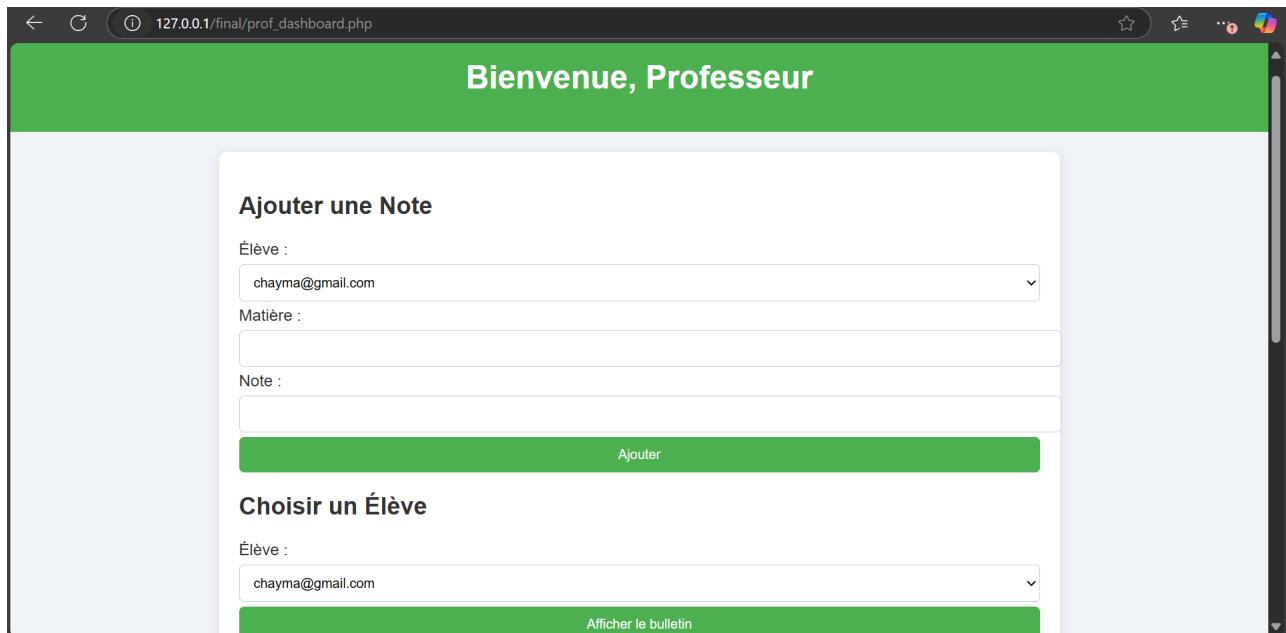
- La page de Connection



**FIGURE 2.17 – Page de connection**

Cette figure présente l’interface utilisateur de la page de connexion. Elle affiche un formulaire dans lequel les utilisateurs peuvent entrer leurs identifiants et leur mot de passe pour accéder à leur compte. Le design de cette page a été travaillé pour offrir une présentation claire et intuitive, avec des champs bien espacés et des instructions visibles pour guider l’utilisateur.

## — Tableau de Bord enseignant



The screenshot shows a web-based teacher dashboard. At the top, a green header bar displays the text "Bienvenue, Professeur". Below this, there are two main sections. The first section, titled "Ajouter une Note", contains fields for "Élève" (with a dropdown menu showing "chayma@gmail.com"), "Matière" (empty field), and "Note" (empty field). A green "Ajouter" button is at the bottom. The second section, titled "Choisir un Élève", contains a similar "Élève" dropdown menu (showing "chayma@gmail.com") and a green "Afficher le bulletin" button.

**FIGURE 2.18** – Tableau de Bord enseignant

The screenshot shows the "Bulletin de l'Élève" (Student Report Card) section of the dashboard. It features a green "Afficher le bulletin" button at the top. Below it, a green link "Ajouter un élève" is visible. The main area displays a table titled "Bulletin de l'Élève" with columns for "Matière" and "Note". The table data is as follows:

Matière	Note
RA	12
SVT	19
EN	13
MP	18
MP	18
SVT	18
INFO	13
SVT	17
SVT	17

Moyenne Générale : 16.11

**FIGURE 2.19** – Tableau de Bord enseignant

Cette figure montre le tableau de bord destiné aux enseignants. L'interface propose des options de gestion des élèves et de leurs notes, telles que la consultation des bulletins et l'ajout de nouvelles données.

— Tableau de Bord élève

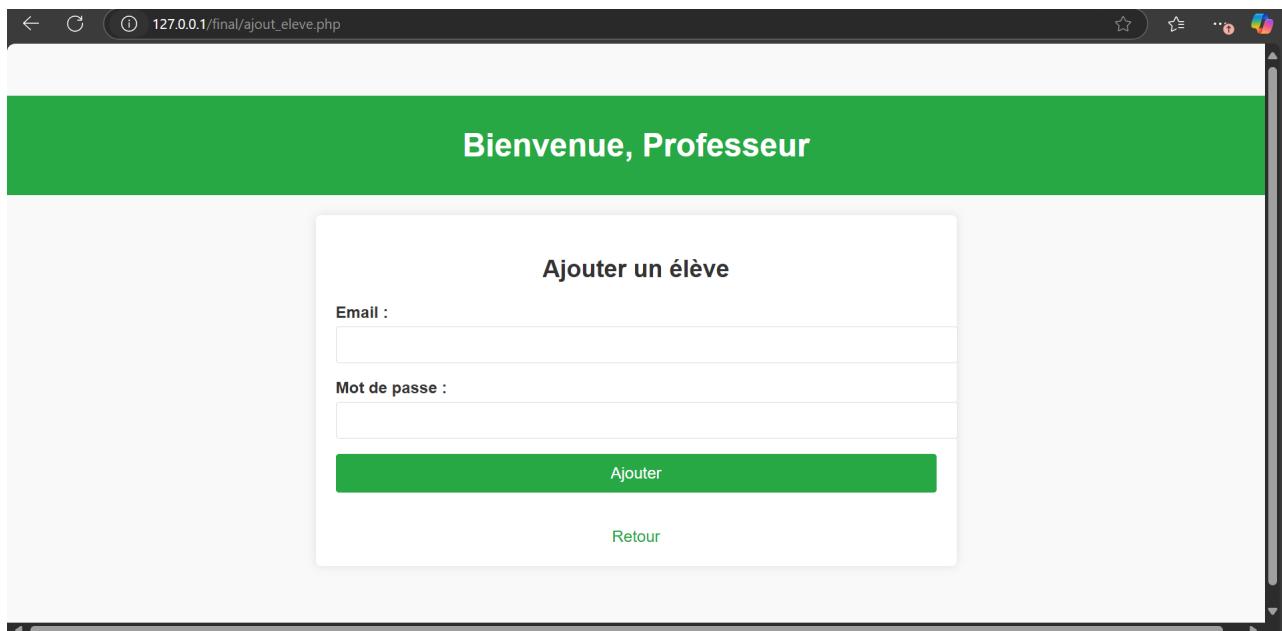
Bienvenue, Élève	
Vos Notes	
Matière	Note
RA	12
SVT	19
EN	13
MP	18
MP	18
SVT	18
INFO	13
SVT	17
SVT	17

**Votre moyenne : 16.11**

**FIGURE 2.20** – Tableau de bord élève

Cette figure illustre le tableau de bord conçu pour les élèves. L’interface permet aux élèves de consulter leurs résultats académiques sous forme de tableaux clairs et bien structurés. Les informations essentielles, comme les matières, les notes et les moyennes, sont présentées de manière lisible et ordonnée.

— Page d'ajout d'élève



**FIGURE 2.21** – ajout élève

Cette figure présente l'interface de la page d'ajout d'un élève. Le formulaire affiche des champs clairs permettant de saisir les informations nécessaires concernant l'élève, comme l'adresse email et le motdepasse . L'objectif de cette interface est de faciliter l'ajout rapide et efficace des élèves au système.

## Conclusion générale

Ce stage a été une bonne occasion d'approfondir mes connaissances pratiques, d'abord, pour m'adapter aux nouvelles technologies qui s'améliorent de jour en jour afin de persister dans un secteur qui s'éveille quotidiennement à de nouvelles innovations. Le projet m'a également donné la meilleure opportunité de la société de services informatiques, et connaître de près son fonctionnement interne et vis-à-vis des clients.

Il est utile de comprendre l'environnement pour commencer une carrière de développeur. J'ai appris que le domaine théorique et le domaine pratique sont complémentaires, et que, dans le monde du travail, il ne suffit pas seulement d'avoir des connaissances théoriques basiques, mais aussi, il faut surtout être sociable et coopératif et avoir un esprit d'équipe pour accomplir toutes les tâches. On peut en conclure que le stage est une opportunité très avantageuse pour le stagiaire, en dépit de sa durée très courte.

