

Enoncé du Projet

On souhaite créer une application web de gestion de scolarité. L'application doit permettre aux étudiants, enseignants et administrateurs de gérer les informations académiques. Les fonctionnalités incluent la gestion des étudiants, des cours, des inscriptions, et des résultats.

1. Fonctionnalités Principales :

- **Gestion des Étudiants** : enregistrement et mise à jour des informations des étudiants (nom, prénom, date de naissance, contact), suppression d'étudiants, affichage de la liste des étudiants, consultation des détails d'un étudiant, recherche et filtrage des étudiants.
- **Gestion des Enseignants** : Enregistrement et mise à jour des informations des enseignants, consultation des détails d'un enseignant, affectation des enseignants aux cours.
- **Gestion des Cours** : création, modification, suppression de cours, affichage de la liste des cours, Attribution des cours aux enseignants et des étudiants.
- **Inscriptions** : Inscription d'étudiants aux cours, consultation des cours auxquels un étudiant est inscrit, affichage des cours inscrits pour chaque étudiant.
- **Saisie des Notes** : Interface pour que les enseignants saisissent les notes des étudiants.
- **Gestion des Résultats** : affichage des notes, calcul des moyennes, Consultation des résultats par étudiant et par cours, génération de relevés de notes
- **Authentification et Autorisation** : Gestion des rôles (étudiant, enseignant, administrateur).

2. Architecture du Projet

- **Modèle** : Classes Java représentant les entités (Étudiant, Enseignant, Cours, Inscription, Résultat).
- **Vue** : JSP pour l'affichage des données.
- **Contrôleur** : Servlets pour gérer les requêtes HTTP et la logique métier.

- **Base de Données** : Utilisation de Hibernate pour la gestion des données et un système de gestion de base de données (MySQL, PostgreSQL, ...).

3. Technologies Utilisées

- **Langages** : Java, HTML, CSS, JavaScript
- **Framework**: Spring Boot (pour le backend)
- **ORM** : Hibernate (pour la gestion des données)
- **Serveur Web** : Apache Tomcat
- **Base de Données** : MySQL ou PostgreSQL

4. Etapes de Développement et contraintes à respecter

- Analyse des Besoins : Identifier les besoins spécifiques des utilisateurs (administrateurs, enseignants, étudiants).
- Installer et configurer, ApacheTomcat, Hibernate et un système de gestion de base de données (MYSQL, PostgreSQL, ...)
- Créer la base de données les différentes tables nécessaires.
- Développer les classes Java pour Etudiant, Enseignant, Cours, ..., avec les relations appropriées.
- Créer des servlets pour gérer les opérations **CRUD** (Create, Read, Update, Delete), lecture, écriture et la mise à jour de la base de données.
- Développer des pages JSP nécessaires (les interfaces) pour chaque utilisateur (administrateur, Enseignant, Etudiant).
- Mettre en place une gestion des utilisateurs avec authentification pour sécuriser l'accès aux différentes fonctionnalités.
- Toutes les données saisies des interfaces doivent être vérifier par les servlets avant d'être sauvegarder sur la base de données.
- Ajouter une fonctionnalité pour générer des rapports sur les performances des étudiants.
- Mettre en place un système de notification par email pour informer les étudiants des changements dans leur inscription ou de la publication des notes.

5. Travail demandé

1. **La partie conception** : Dans cette partie, vous donnez le **MCD** (Modèle Conceptuel des Données) ou le diagramme de classes qui représente les différentes interactions entre les entités (classes) de l'application (Selon la méthode de conception utilisée).
2. **La partie implémentation** : vous créer une application Web dynamique qui doit :
 - Fournir toutes les fonctionnalités décrites dans la section 2.
 - Respecter tous les points les exigences donnés dans la section 4.
 - Utiliser les technologies vues en cours et cités aussi en niveau de la section 3
 - Respecter l'architecture MVC (Modèle, Vue, Contrôleur).
 - Assurer une cohérence de données pour toute manipulation de la base de données
3. Refaire l'application en utilisant **Spring Boot**

6. Exigences

- Le travail demandé sera réalisé par un groupe de **5 étudiants**.
- Chaque groupe doit être inscrit sur la liste qui sera mis sur teams et fournit un lien GitHub permettant l'accès à application (vous devez fournir un **lien privé e au début** et vous le rendez **public** à la date du dépôt final) avant le 08/11/2024 23h59. **Une pénalité est appliquée en cas du retard.**
- Chaque groupe d'étudiants doit rendre un rapport qui détaille la partie conception et qui explique le schéma de la partie implémentation (partie 2+partie SpringBoot) et faire une présentation orale entre **(10 à 15 min)**. Le rapport ne doit pas dépasser **15 pages**.
- Le dépôt final comporte le rapport et l'application que vous avez implémenté
- La date limite de dépôt est le **30/11/2024, 23h59**. Une **pénalité est appliquée en cas du retard**.