#### **Rôle Étudiant :**

* **Accès aux Informations Personnelles** : Accès à ses informations personnelles telles que le nom, prénom, date de naissance et contact.
* **Cours et Inscriptions** :
  + Consultation de la liste des cours disponibles.
  + Inscription aux cours souhaités.
  + Consultation de la liste des cours auxquels l'étudiant est inscrit.
* **Notes et Résultats** :
  + Consultation des notes et résultats pour chaque cours.
  + Calcul et affichage de la moyenne pour chaque étudiant.
  + Génération et consultation de relevés de notes.

#### **Rôle Enseignant**

* **Accès aux Informations Personnelles** : Accès à ses informations personnelles telles que nom, prénom, date de naissance et contact.
* **Cours Attribués** :
  + Consultation de la liste des cours qui lui sont attribués par l’administrateur.
* **Saisie des Notes** :
  + Interface pour saisir les notes des étudiants inscrits dans ses cours.
* **Résultats des Étudiants** :
  + Consultation des résultats pour chaque étudiant dans chaque cours attribué.
  + Calcul et affichage des moyennes pour les cours attribués.

#### **Rôle Administrateur**

* **Gestion des Étudiants** :
  + Ajout et mise à jour des informations des étudiants.
  + Suppression des étudiants de la base de données.
  + Affichage et consultation des détails des étudiants.
  + Recherche et filtrage des étudiants par divers critères.
  + Consultation des cours auxquels chaque étudiant est inscrit.
* **Gestion des Enseignants** :
  + Ajout et mise à jour des informations des enseignants.
  + Consultation des détails des enseignants.
  + Attribution des cours aux enseignants.
* **Gestion des Cours** :
  + Création, modification, et suppression des cours.
  + Affichage de la liste des cours.
  + Affectation des cours aux enseignants et inscription des étudiants.
* **Gestion des Rôles** :
  + Attribution des rôles pour gérer les accès (étudiant, enseignant, administrateur).

#### 

**Détails des technologies utilisées :**

* **Backend** : Java et Spring Boot pour le développement des fonctionnalités côté serveur.
* **Frontend** : HTML, CSS, et JavaScript pour l'interface utilisateur.
* **ORM** : Hibernate pour la gestion de la base de données.
* **Base de Données** : MySQL, gérée avec Hibernate.
* **Serveur** : Apache Tomcat pour déployer et exécuter l’application.

#### **Étapes de Développement**

1. **Développement des Entités et Logique CRUD** : Développer les classes Java pour les entités et les opérations CRUD (Create, Read, Update, Delete).
2. **Développement de l’Interface Utilisateur** : Création des pages JSP pour chaque rôle.
3. **Authentification et Autorisation** : Mise en place de la gestion des rôles et de la sécurité.
4. **Vérification et Validation des Données** : Validation des données côté serveur avant enregistrement.
5. **Rapports et Notifications** : (bonus)
   * Générer des rapports de performance des étudiants.
   * Mettre en place un système de notification par e-mail pour les mises à jour (ex : notes, inscriptions).

#### **Architecture MVC**

L'application doit être conçue selon le modèle MVC (Modèle-Vue-Contrôleur) pour assurer une séparation claire entre la logique métier, la gestion des données, et l'interface utilisateur.

* **Modèle** : Classes Java pour représenter les entités (ex. Étudiant, Enseignant, Cours, etc.).
* **Vue** : JSP pour le rendu de l’interface.
* **Contrôleur** : Servlets pour traiter les requêtes et la logique métier.