

# Rapport de projet J2EE

Professeur en charge : Haddache Mohamed



Liens github : <https://github.com/Dnewe/ProjetJEE> (sans Spring Boot)

[Dnewe/jeeproject-springboot](https://github.com/Dnewe/jeeproject-springboot) (avec Spring Boot)

## **Sommaire :**

<b>1-Introduction.....</b>	<b>3</b>
<b>2-Conception.....</b>	<b>4</b>
<b>3-Implémentation.....</b>	<b>7</b>
<b>4-Spring Boot.....</b>	<b>24</b>
<b>5-Conclusion.....</b>	<b>25</b>

# 1-Introduction :

L'objectif de ce projet était de se familiariser avec toute l'architecture J2EE et de découvrir les frameworks les plus couramment utilisés dans le développement web en Java. Nous avons commencé par analyser les besoins des utilisateurs de l'application, en particulier ceux des administrateurs, qui nous semblaient les plus complexes à mettre en œuvre. Ensuite, comme demandé dans le sujet, nous avons configuré les outils nécessaires, tels que Tomcat et Hibernate.

Nous nous sommes ensuite mis d'accord sur l'architecture de notre base de données, qui comprend six tables : course (pour les cours), enrollment (pour les inscriptions aux cours), professor (pour les professeurs), result (pour les notes), student (pour les étudiants), user (pour les utilisateurs).

Nous avons décidé de séparer les tables user, student et professor, car les étudiants, les professeurs et les administrateurs ne partagent pas les mêmes données. Par exemple, l'administrateur ne possède pas de date de naissance, contrairement aux étudiants, et les professeurs n'en ont pas non plus. Afin de simplifier la gestion et la récupération des données, nous avons opté pour cette séparation dès le départ, ce qui s'est avéré très utile par la suite.

Après cela, nous avons entamé le développement du back-end, en respectant l'architecture MVC, suivi par la création du front-end.

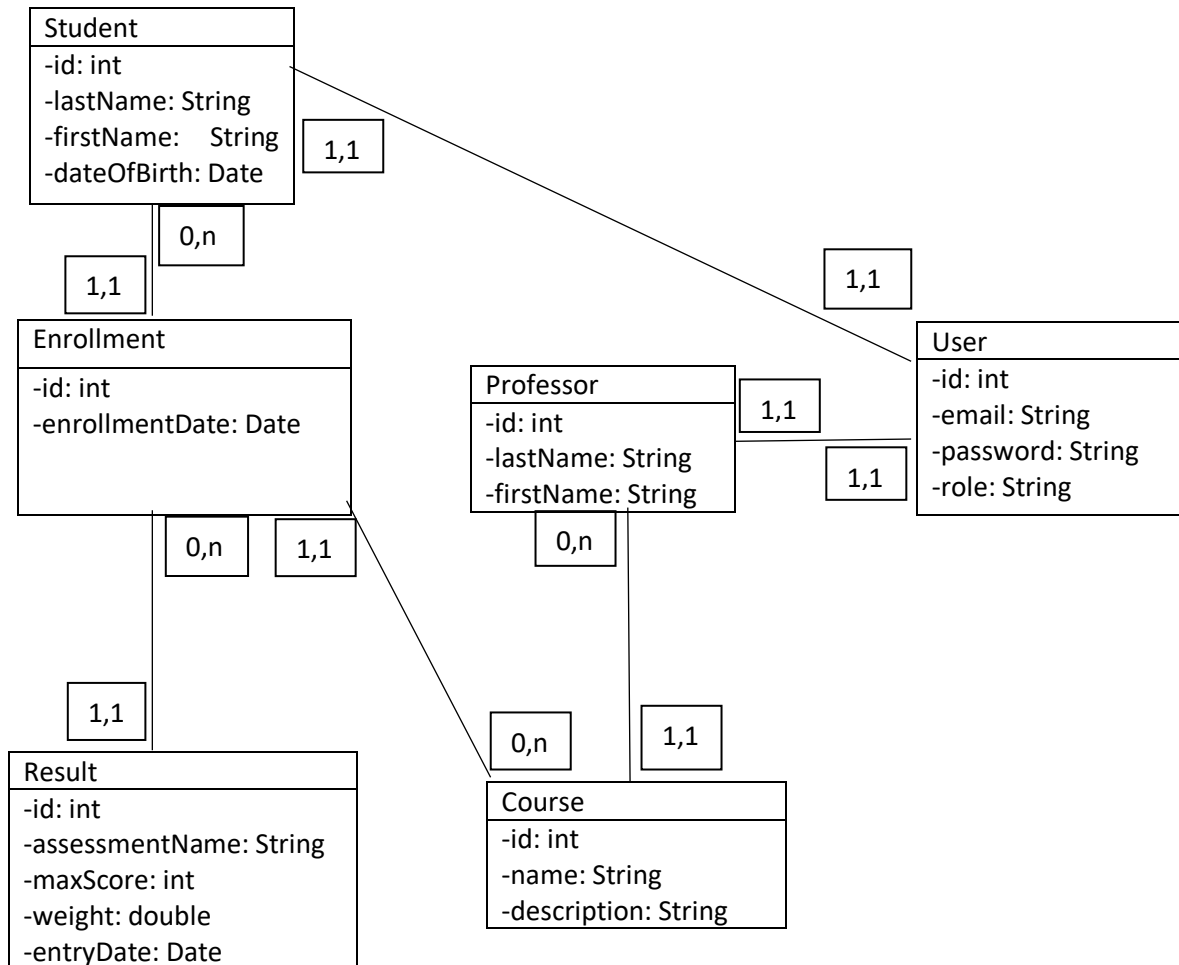
En ce qui concerne les ajouts de fonctionnalités, la génération de rapports sur les performances des étudiants en PDF nous a posé quelques difficultés. Nous nous sommes interrogés sur le contenu du rapport : devons-nous comparer les étudiants entre eux ou avec ceux des années précédentes ?

Finalement, nous avons décidé de comparer chaque étudiant avec sa promotion dans les matières auxquelles il est inscrit.

La mise en place des notifications par e-mail a pris du temps, mais nous n'avons rencontré aucun problème majeur lors de son implémentation.

## 2-Conception :

Pour la partie conception nous avons fait un diagramme de classe pour détailler et mieux comprendre le développement de notre application. Voici le diagramme effectué en début de projet :



## 1. Classe Student

Attributs :

- id: Identifiant unique de l'étudiant.
- lastName: Nom de famille de l'étudiant.
- firstName: Prénom de l'étudiant.
- dateOfBirth: Date de naissance de l'étudiant.

Relations :

- 1,1 vers User : Chaque étudiant est associé exactement à un utilisateur (relation 1-1 entre Student et User).
- 0,n vers Enrollment : Un étudiant peut être inscrit à plusieurs cours, mais n'est pas obligatoirement inscrit (relation 0-n).

## 2. Classe Enrollment

Attributs :

- id: Identifiant unique de l'inscription.
- enrollmentDate: Date d'inscription.

Relations :

- 1,1 vers Student : Chaque inscription est liée à un étudiant spécifique.
- 0,n vers Result : Une inscription peut contenir plusieurs résultats (évaluations).
- 1,1 vers Course : Chaque inscription est associée à un seul cours.

## 3. Classe Result

Attributs :

- id: Identifiant unique du résultat.
- assessmentName: Nom de l'évaluation (ex. : "Examen final").
- maxScore: Score maximum possible pour cette évaluation.
- weight: Poids de l'évaluation dans la note finale.
- entryDate: Date d'enregistrement du résultat.

Relations :

- 1,1 vers Enrollment : Chaque résultat est lié à une seule inscription.

#### 4. Classe Professor

Attributs :

- id: Identifiant unique du professeur.
- lastName: Nom de famille du professeur.
- firstName: Prénom du professeur.

Relations :

- 1,1 vers User : Chaque professeur correspond à un utilisateur unique.
- 0,n vers Course : Un professeur peut enseigner plusieurs cours, mais il peut aussi ne pas enseigner (relation 0-n).

#### 5. Classe Course

Attributs:

- id : Identifiant unique du cours.
- name: Nom du cours.
- description: Description du cours.

Relations:

- 0,n vers Professor : Un cours peut être enseigné par un ou plusieurs professeurs, ou par aucun.
- 0,n vers Enrollment : Un cours peut avoir plusieurs inscriptions.

#### 6. Classe User

Attributs:

- id : Identifiant unique de l'utilisateur.
- email : Adresse email de l'utilisateur.
- password : Mot de passe de l'utilisateur.
- role : Rôle de l'utilisateur (Student, Professor, etc.).

Relations:

- 1,1 vers Student ou Professor : Chaque utilisateur peut être soit un étudiant, soit un professeur.

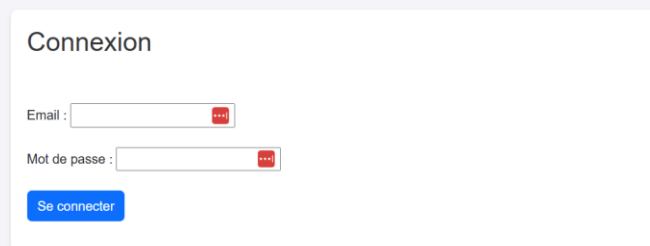
## 3-Implémentation :

Pour la partie implémentation on va détailler dans un premier temps les vues, ensuite nous allons expliquer le fonctionnement de l'application. Il y a un fichier style.css importé par toutes les pages. Le footer et le header sont présents sur toutes les pages sauf celle de connexion.

Le header a le logo de l'école et un bouton accueil (redirection au dashboard) à gauche, un bouton déconnexion et un bouton paramètres (settings.jsp) à droite.

Le footer est sobre, il y a les droits et le nom de l'école.

Capture d'écran de login.jsp :



The screenshot shows a login form titled "Connexion". It contains two input fields: "Email :" and "Mot de passe :". Each field has a red "x" icon on the right side, indicating a validation error. Below the fields is a blue button labeled "Se connecter".

Cette page permet aux utilisateurs de se connecter à l'application en saisissant leur adresse e-mail et leur mot de passe, qu'on récupérera via un formulaire de connexion. Cette page est le point d'entrée de l'application et joue un rôle crucial pour la sécurité, car elle authentifie les utilisateurs avant qu'ils puissent accéder aux fonctionnalités protégées. De plus il y a une gestion des messages d'erreurs.

Capture d'écran de settings.jsp :

Cette page permet à l'utilisateur de modifier son mot de passe.

Capture d'écran de adminDashboard.jsp :

Cette page est destinée à l'espace d'administration, où l'administrateur peut accéder aux différentes fonctionnalités de gestion des utilisateurs, professeurs, étudiants, et cours. Accès à ses fonctionnalités par des boutons. Cette page sert de point d'entrée pour l'administrateur, lui permettant de gérer les différentes entités liées à la gestion académique. Elle centralise toutes les actions administratives essentielles dans un seul tableau de bord.



Capture d'écran de users.jsp :

The screenshot shows the 'Gestion des Utilisateurs' (User Management) page. At the top, there is a header with the 'Cy-Tech Gestion' logo, an 'Accueil' button, and 'Paramètres' and 'Déconnexion' buttons. The main content area features a table with the following data:

#	Email	Rôle	Actions
1	admin@example.com	admin	<a href="#">Détails</a> <a href="#">Modifier</a> <a href="#">Supprimer</a>
2	professor@example.com	professor	<a href="#">Détails</a> <a href="#">Modifier</a> <a href="#">Supprimer</a>
3	student@example.com	student	<a href="#">Détails</a> <a href="#">Modifier</a> <a href="#">Supprimer</a>
4	ewenstvallier@gmail.com	student	<a href="#">Détails</a> <a href="#">Modifier</a> <a href="#">Supprimer</a>
13	professor2@example.com	professor	<a href="#">Détails</a> <a href="#">Modifier</a> <a href="#">Supprimer</a>
14	studen2t5@example.com	student	<a href="#">Détails</a> <a href="#">Modifier</a> <a href="#">Supprimer</a>

Below the table is a green button labeled 'Ajouter un Utilisateur'. At the bottom of the page, there is a footer with the text '© 2024 Cy-Tech. Tous droits réservés.'

Cette page permet aux administrateurs de gérer la liste des utilisateurs de l'application. Elle affiche un tableau contenant les utilisateurs et offre des actions pour afficher, modifier ou supprimer un utilisateur, mais aussi inscrire un nouvel utilisateur.

Capture d'écran de register.jsp :

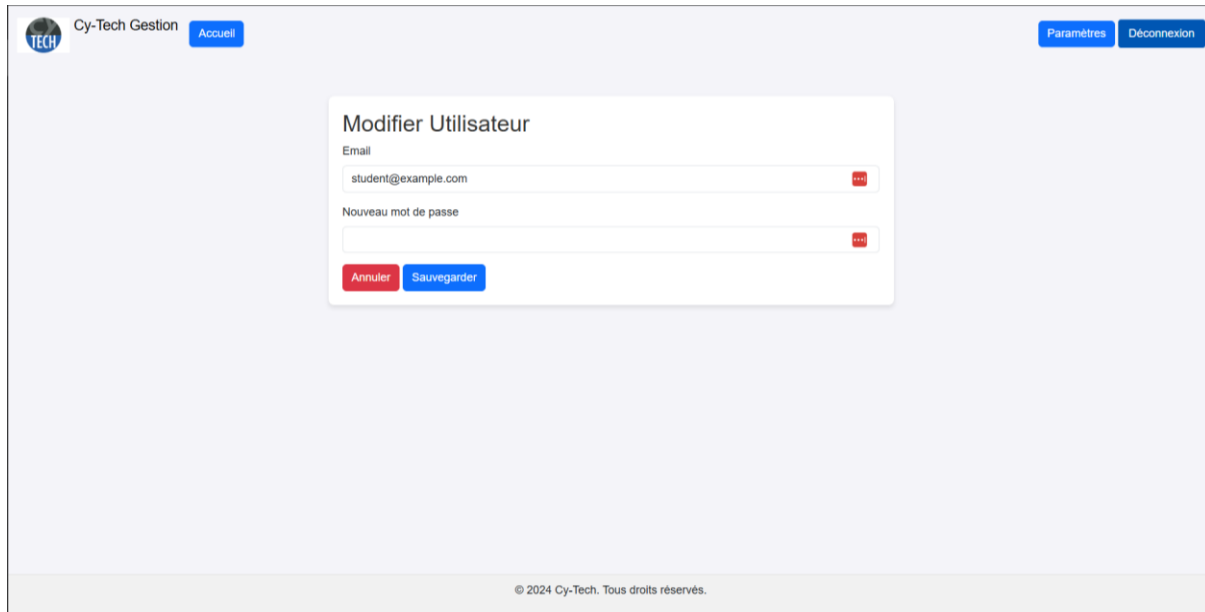
The screenshot shows the 'Inscription Utilisateur' (User Registration) page. At the top, there is a header with the 'Cy-Tech Gestion' logo, an 'Accueil' button, and 'Paramètres' and 'Déconnexion' buttons. The main content area features a form with the following fields:

- Email:
- Mot de passe:
- Rôle:
- Nom:
- Prénom:
- Date de naissance:

At the bottom of the form are two buttons: 'Annuler' and 'Enregistrer'. At the bottom of the page, there is a footer with the text '© 2024 Cy-Tech. Tous droits réservés.'

Cette page permet à un administrateur d'inscrire un utilisateur sur l'application en choisissant son rôle (administrateur, étudiant, ou enseignant). Selon le rôle sélectionné, des champs spécifiques s'affichent pour compléter l'inscription (formulaire dynamique). Cette page est essentielle pour l'inscription des utilisateurs dans l'application, en permettant de gérer différents types de rôles avec des informations spécifiques pour chaque profil (étudiant, enseignant).

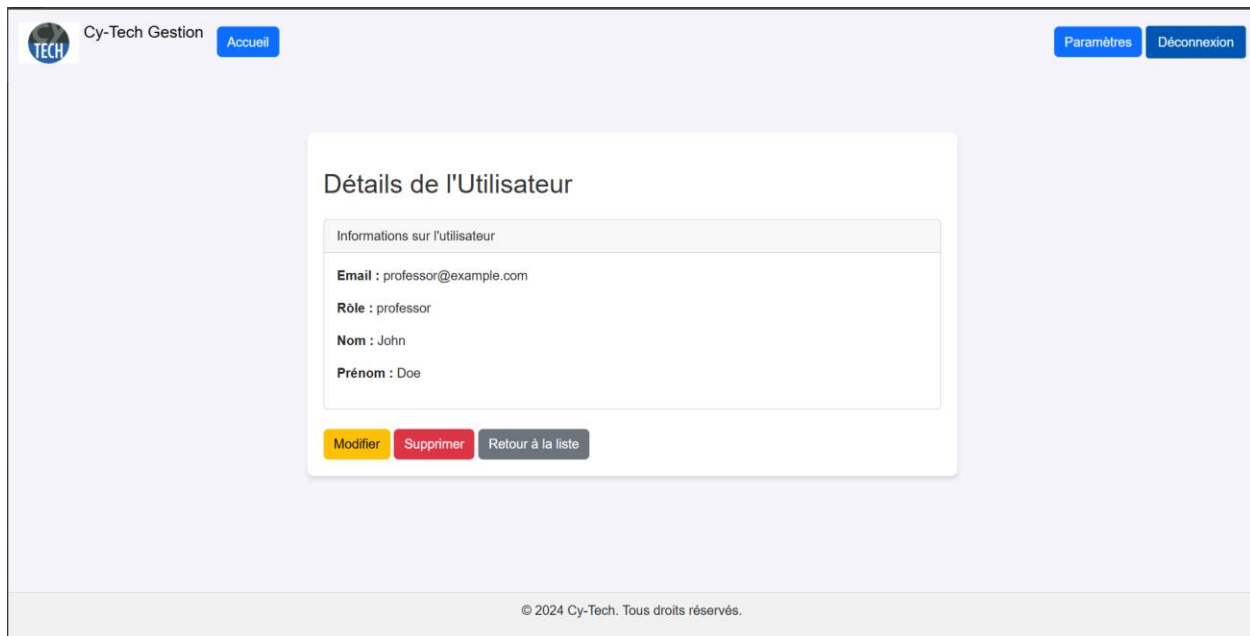
Capture d'écran de updateUser.jsp :



The screenshot displays a web application interface for 'Cy-Tech Gestion'. At the top left, there is a logo with the word 'TECH' and a navigation bar containing 'Accueil', 'Paramètres', and 'Déconnexion'. The main content area features a central form titled 'Modifier Utilisateur'. This form includes two input fields: 'Email' with the value 'student@example.com' and 'Nouveau mot de passe'. Each field has a red 'x' icon on the right side. Below the input fields are two buttons: 'Annuler' (red) and 'Sauvegarder' (blue). The footer of the page contains the copyright notice '© 2024 Cy-Tech. Tous droits réservés.'

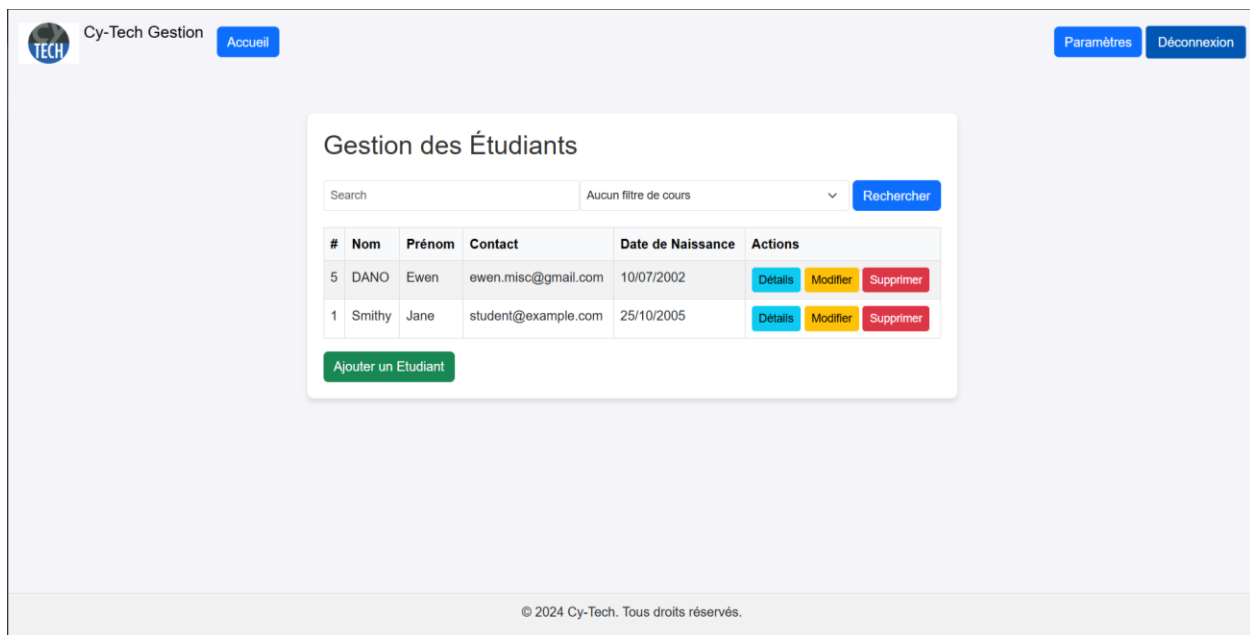
Cette page permet de modifier les informations d'un utilisateur existant dans l'application, notamment son email et son mot de passe.

Capture d'écran de userDetails.jsp :



Cette page affiche les informations détaillées d'un utilisateur spécifique (email, rôle, nom, prénom) et propose des actions pour modifier ou supprimer cet utilisateur.

Capture d'écran de students.jsp :



Cette page permet aux administrateurs de gérer la liste des étudiants de l'application. Elle affiche un tableau des étudiants et permet d'effectuer des actions telles que consulter, modifier ou supprimer un étudiant, mais également d'en ajouter un. Cette page permet ainsi une gestion complète des étudiants, de leur recherche à leur modification, en passant par la suppression, avec des filtres et des messages de retour pour améliorer l'expérience utilisateur.

Capture d'écran de studentDetails.jsp :

The screenshot shows the 'Détails de l'Étudiant' page. At the top, there's a header with 'Cy-Tech Gestion', an 'Accueil' button, and 'Paramètres' and 'Déconnexion' buttons. The main content area is titled 'Détails de l'Étudiant' and contains several sections:

- Informations sur l'étudiant:** A form displaying student details: Nom : DANO, Prénom : Ewen, Contact : ewen.misc@gmail.com, and Date de naissance : 10/07/2002.
- Cours inscrits:** A table showing enrolled courses.

Nom du cours	Professeur	Action
Introduction to Databases	Doe John	<button>Retirer</button>

- Inscrire à un Cours:** A section with a dropdown menu labeled 'Sélectionner un cours' and an 'Inscrire' button.
- Administratif:** A section with buttons for 'Télécharger Relevé de notes', 'Télécharger Rapport de Performance', 'Modifier', 'Supprimer', and 'Retour à la liste'.

At the bottom, there is a footer: '© 2024 Cy-Tech. Tous droits réservés.'

Cette page fournit une gestion complète des informations de l'étudiant, lui permettant à la fois de gérer les inscriptions aux cours, d'accéder à des documents administratifs, et de modifier ou supprimer les informations de l'étudiant selon les besoins.

Capture d'écran de updateStudent.jsp :

The screenshot shows the 'Modifier Étudiant' page. It features a form for updating student information:

- Nom:** A text input field containing 'DANO'.
- Prénom:** A text input field containing 'Ewen'.
- Date de naissance:** A date picker input field showing 'jj/mm/aaaa'.

At the bottom of the form, there are two buttons: 'Annuler' (red) and 'Sauvegarder' (blue). The footer at the bottom reads: '© 2024 Cy-Tech. Tous droits réservés.'

Cette page permet de mettre à jour les informations personnelles de l'étudiant.

Capture d'écran de createTranscript.jsp :

Cette page offre une gestion efficace des relevés de notes des étudiants, permettant à un administrateur ou à un enseignant de générer un document officiel avec des résultats détaillés pour chaque cours, accompagné d'une appréciation. Les informations sont clairement organisées, et l'utilisateur peut facilement revenir à la page de détails de l'étudiant ou annuler l'opération.

Capture d'écran de professors.jsp :

The screenshot displays the 'Gestion des Professeurs' interface. At the top, the header includes the 'Cy-Tech Gestion' logo, the title 'Cy-Tech Gestion', and navigation buttons for 'Accueil', 'Paramètres', and 'Déconnexion'. The main content area is titled 'Gestion des Professeurs' and contains a table with the following data:

#	Nom	Prénom	Contact	Actions
1	Doe	John	professor@example.com	<a href="#">Détails</a> <a href="#">Modifier</a> <a href="#">Supprimer</a>
5	Dupont	Julie	professor2@example.com	<a href="#">Détails</a> <a href="#">Modifier</a> <a href="#">Supprimer</a>

Below the table is a green button labeled 'Ajouter un Professeur'. The footer of the page states '© 2024 Cy-Tech. Tous droits réservés.'

Cette page fournit une interface complète pour la gestion des professeurs dans l'application. L'administrateur peut afficher les informations de chaque professeur, les modifier, les supprimer et en ajouter de nouveaux. Le système de confirmation avant la suppression garantit que les utilisateurs ne suppriment pas accidentellement des données importantes.

Capture d'écran de professorDetails.jsp :

The screenshot shows the 'Détails du Professeur' page. At the top, there is a header with the 'Cy-Tech Gestion' logo, an 'Accueil' button, and 'Paramètres' and 'Déconnexion' buttons. The main content area contains a form titled 'Détails du Professeur' with two input fields: 'Nom' (John Doe) and 'Email' (professor@example.com). Below this is a section 'Cours Assignés' with a table showing one course: 'Introduction to Databases' with the description 'Learn the basics of SQL and database design.' and an 'Action' column with a 'Retirer' button. Underneath is a section 'Assigner un Nouveau Cours' with a dropdown menu 'Choisir un Cours' (Sélectionner un cours), an 'Assigner' button, and three buttons at the bottom: 'Modifier' (yellow), 'Supprimer' (red), and 'Retour à la liste des professeurs' (grey). The footer contains the copyright notice '© 2024 Cy-Tech. Tous droits réservés.'

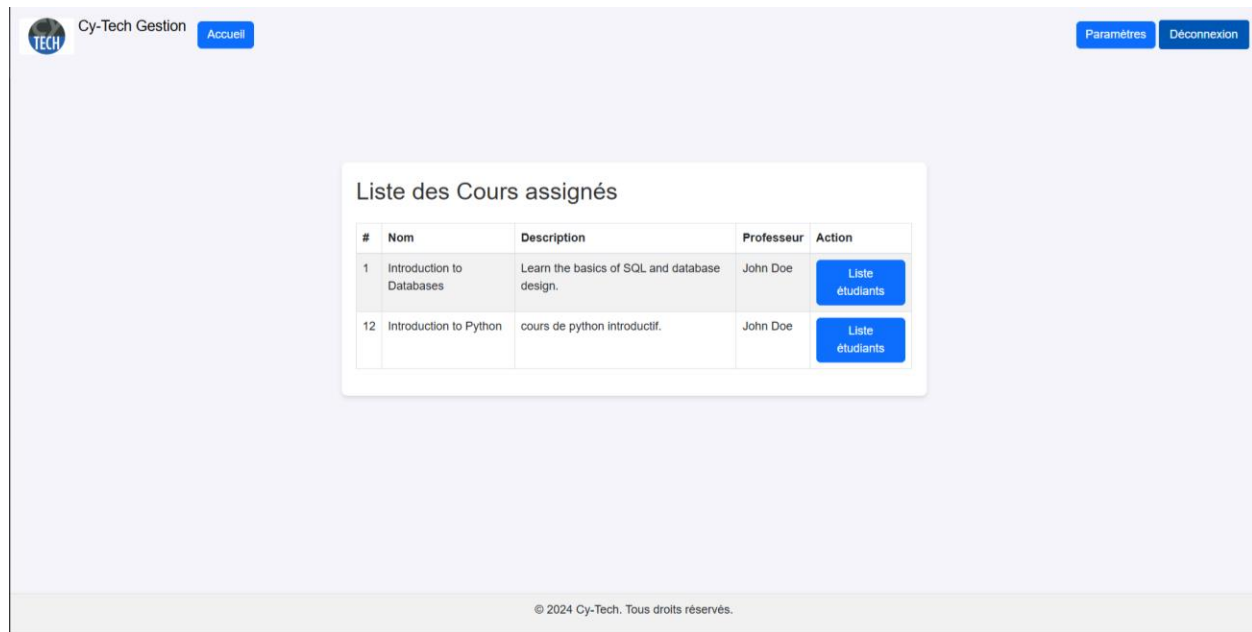
Cette page permet une gestion complète des informations relatives à un professeur : consulter ses détails, assigner ou retirer des cours, modifier ses informations, et éventuellement le supprimer. Elle est sécurisée par des confirmations avant des actions critiques.

Capture d'écran de updateProfessor.jsp :

The screenshot shows the 'Modifier Professeur' page. It features a form with two input fields: 'Nom' (Doe) and 'Prénom' (John). Below the fields are two buttons: 'Annuler' (red) and 'Sauvegarder' (blue). The page layout is consistent with the previous screenshot, including the header and footer.

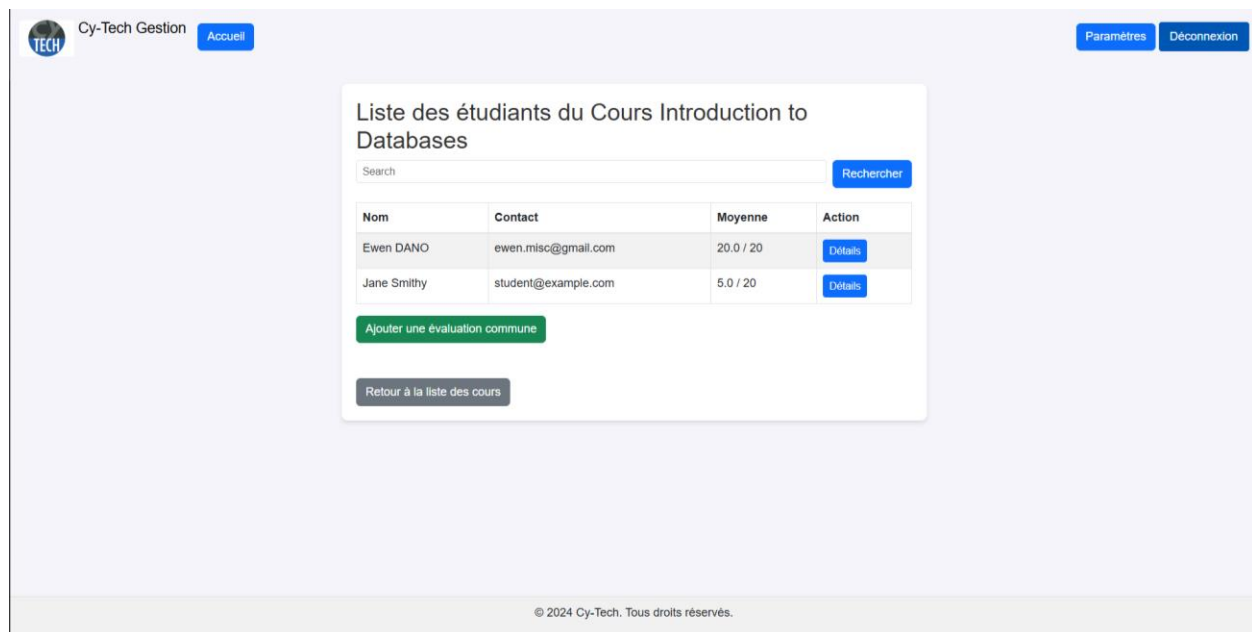
Cette page permet de modifier les informations d'un professeur

Capture d'écran de courses.jsp :



Cette page permet une gestion complète des cours, incluant la consultation, la modification, et la suppression des cours, ainsi que l'ajout de nouveaux cours.

Capture d'écran de courseDetails.jsp :



Cette page affiche les détails d'un cours, ainsi que des options pour gérer les informations relatives à ce cours, telles que l'affectation de professeurs et l'inscription d'étudiants aux cours.

Capture d'écran de createCourse.jsp :

The screenshot shows the 'Créer Cours' form in the Cy-Tech Gestion application. The form is centered on a light purple background. At the top left, there is a logo with the word 'TECH' and the text 'Cy-Tech Gestion' next to it, followed by a blue 'Accueil' button. At the top right, there are two blue buttons: 'Paramètres' and 'Déconnexion'. The form itself has a title 'Créer Cours' and two input fields: 'Nom' and 'Description'. Below the input fields are two buttons: a red 'Annuler' button and a blue 'Créer' button. At the bottom of the page, there is a footer that reads '© 2024 Cy-Tech. Tous droits réservés.'

Cette page permet de créer un nouveau cours. En remplissant un formulaire.

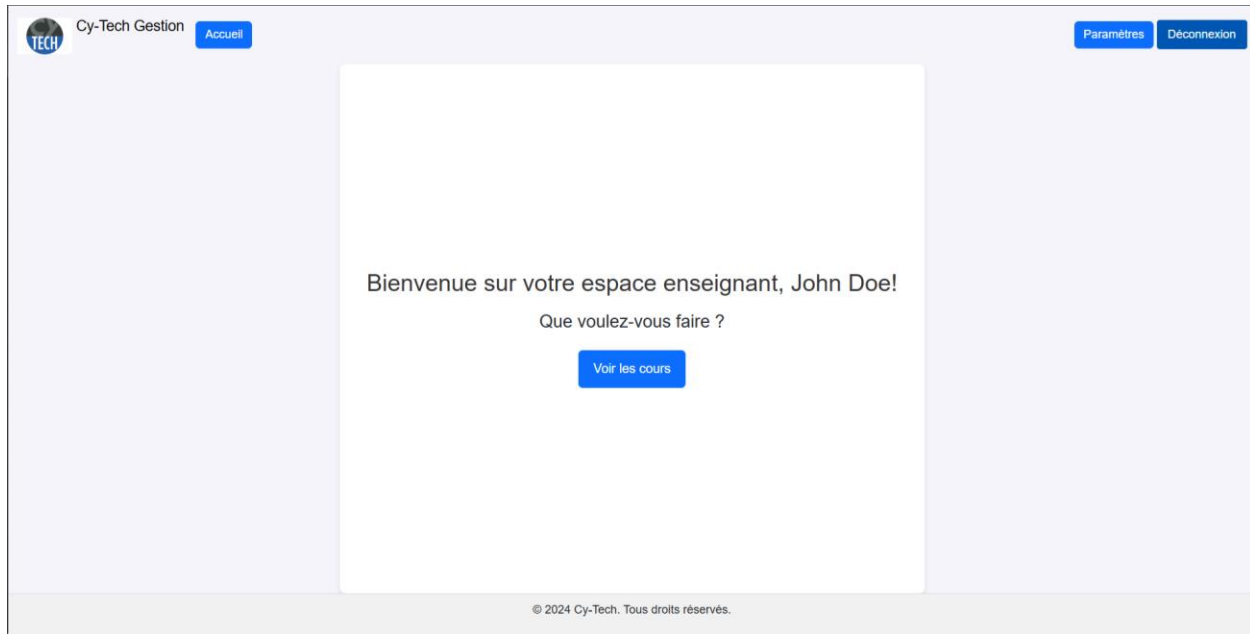
Capture d'écran de updateCourse.jsp :

The screenshot shows the 'Modifier Cours' form in the Cy-Tech Gestion application. The form is centered on a light purple background. At the top left, there is a logo with the word 'TECH' and the text 'Cy-Tech Gestion' next to it, followed by a blue 'Accueil' button. At the top right, there are two blue buttons: 'Paramètres' and 'Déconnexion'. The form itself has a title 'Modifier Cours' and two input fields: 'Nom' and 'Description'. The 'Nom' field is pre-filled with 'Introduction to Databases' and the 'Description' field is pre-filled with 'Learn the basics of SQL and database design.' Below the input fields are two buttons: a red 'Annuler' button and a blue 'Sauvegarder' button. At the bottom of the page, there is a footer that reads '© 2024 Cy-Tech. Tous droits réservés.'

Cette page permet à un utilisateur de modifier un cours existant. Le formulaire pré-remplit les champs avec les données actuelles et offre des options pour sauvegarder ou annuler les modifications.

Capture d'écran de professorDashboard.jsp :





Cette page est dédiée à l'espace personnel de l'enseignant, où il peut consulter les cours qui lui sont attribués. Cette page permet à l'enseignant de naviguer dans son espace et d'accéder à la gestion des cours qui lui sont associés.

Capture d'écran de gradeStudent.jsp :

The screenshot shows the 'Nouvelle notes' form. At the top left is the 'Cy-Tech Gestion' logo and an 'Accueil' button. At the top right are 'Paramètres' and 'Déconnexion' buttons. The main content area is a white card with the title 'Nouvelle notes pour Ewen DANO en Introduction to Databases'. Below the title is the section 'Information de l'évaluation'. This section contains four input fields: 'Nom de l'évaluation', 'Note', 'Note maximale', and 'Coefficient'. At the bottom of the card are two buttons: 'Annuler' (red) and 'Sauvegarder' (blue). At the bottom of the page is a footer with the text '© 2024 Cy-Tech. Tous droits réservés.'

Cette page est conçue pour ajouter une nouvelle note pour un étudiant dans un cours spécifique. Elle inclut des champs pour saisir le nom de l'évaluation, la note, la note maximale et le coefficient, avec des mécanismes pour valider les informations saisies et rediriger l'utilisateur après la soumission.

Capture d'écran de gradeStudents.jsp :

The screenshot shows a web application interface for 'Cy-Tech Gestion'. At the top, there is a header with the application name and a logo, and navigation buttons for 'Accueil', 'Paramètres', and 'Déconnexion'. The main content area is titled 'Nouvelle évaluation pour le cours Introduction to Databases' and 'Informations générales'. It contains several input fields: 'Nom de l'évaluation' (DS test), 'Note maximale' (100), and 'Coefficient' (5). Below these is a section titled 'Liste des étudiants' which contains a table with three columns: 'Prénom', 'Nom', and 'Note'. The table has two rows: one for 'Jane Smithy' with a note of '75', and another for 'Ewen DANO' with a placeholder 'Entrez la note'. At the bottom of the form are two buttons: 'Annuler' (red) and 'Sauvegarder' (blue). The footer of the page indicates '© 2024 Cy-Tech. Tous droits réservés.'

Prénom	Nom	Note
Jane	Smithy	75
Ewen	DANO	Entrez la note

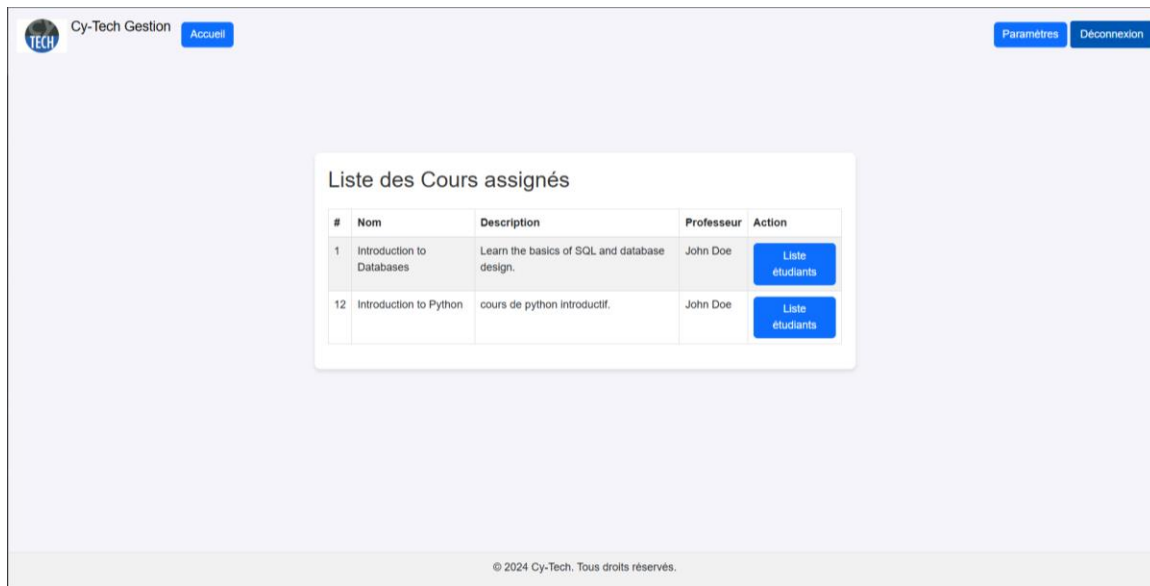
Cette page est conçue pour créer une nouvelle évaluation dans un cours spécifique et permettre l'entrée des notes pour plusieurs étudiants contrairement à la page gradeStudent.jsp. Elle inclut des champs pour saisir le nom de l'évaluation, la note maximale, le coefficient, et un tableau pour entrer les notes de chaque étudiant dans le cours. Le formulaire est validé avant l'envoi et permet de revenir à la page précédente ou d'enregistrer les informations.

Capture d'écran de updateResult.jsp :

The screenshot shows a web application interface for 'Cy-Tech Gestion'. At the top, there is a header with the application name and a logo, and navigation buttons for 'Accueil', 'Paramètres', and 'Déconnexion'. The main content area is titled 'Modifier Résultat'. It contains several input fields: 'Nom de l'évaluation' (TD2 test), 'Note' (13,5), 'Note maximale' (15), and 'Coefficient' (1,0). At the bottom of the form are two buttons: 'Annuler' (red) and 'Sauvegarder' (blue). The footer of the page indicates '© 2024 Cy-Tech. Tous droits réservés.'

Cette page permet à l'enseignant de modifier un résultat d'évaluation spécifique pour un étudiant dans un cours donné. Les champs pour saisir le nom de l'évaluation, la note obtenue, la note maximale, et le coefficient sont pré-remplis avec les valeurs existantes, et le formulaire envoie les modifications à un contrôleur.

Capture d'écran de courses.jsp :

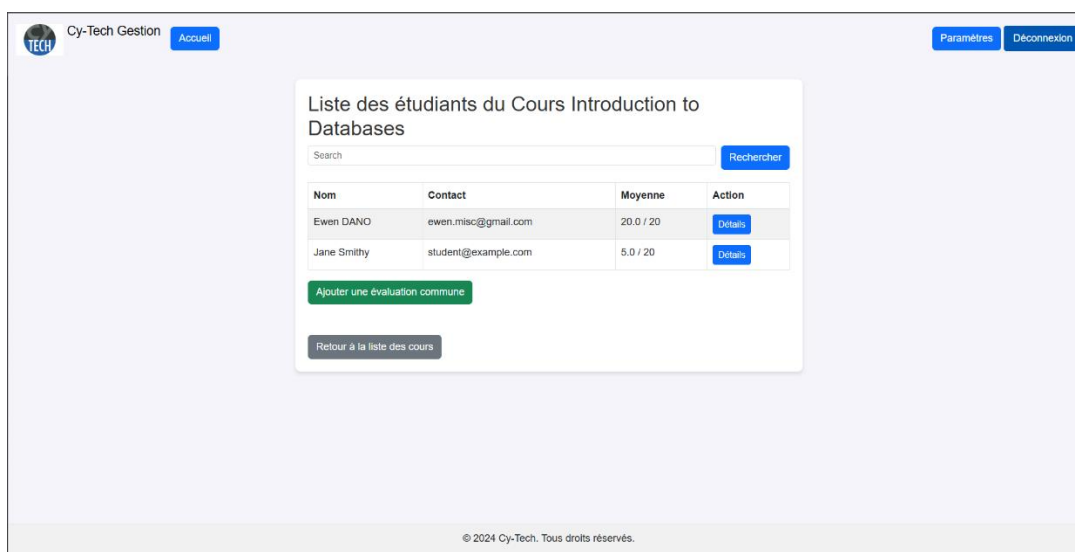


#	Nom	Description	Professeur	Action
1	Introduction to Databases	Learn the basics of SQL and database design.	John Doe	Liste étudiants
12	Introduction to Python	cours de python introductif.	John Doe	Liste étudiants

© 2024 Cy-Tech. Tous droits réservés.

Cette page permet d'afficher un tableau avec la liste des cours assignés à un professeur. Chaque cours est affiché avec son nom, sa description, le nom du professeur, et un lien vers students.jsp. Si aucun cours n'est trouvé, un message est affiché.

Capture d'écran de students.jsp :



Nom	Contact	Moyenne	Action
Ewen DANO	ewen.misc@gmail.com	20.0 / 20	Details
Jane Smithy	student@example.com	5.0 / 20	Details

Rechercher

Ajouter une évaluation commune

Retour à la liste des cours

© 2024 Cy-Tech. Tous droits réservés.

Cette page permet de gérer la liste des étudiants inscrits à un cours, en affichant leurs informations de contact et leur moyenne. Elle offre également des fonctionnalités de recherche, de consultation des résultats détaillés de chaque étudiant, d'ajout d'évaluations communes et de navigation vers la liste des cours.

Capture d'écran de studentDetails.jsp :

The screenshot displays the 'studentDetails.jsp' page. At the top, there's a header with 'Cy-Tech Gestion' and 'Accueil' buttons. On the right, there are 'Paramètres' and 'Déconnexion' buttons. The main content area is titled 'Notes de l'étudiant Ewen DANO pour le cours Introduction to Databases'. Below the title, a green box states 'Note enregistrée avec succès'. A section titled 'Informations sur l'étudiant' contains the following details: Nom : DANO, Prénom : Ewen, Contact : ewen.misc@gmail.com, and Date de naissance : 10/07/2002. Below this is a table of evaluations:

Evaluation	Date	Coefficient	Note	Action
DS1	02/12/2024	2.0	20.0 / 20	Modifier Supprimer
TD1	02/12/2024	0.5	4.0 / 5	Modifier Supprimer
TD2 test	02/12/2024	1.0	13.5 / 15	Modifier Supprimer
<b>Moyenne</b>			<b>18.86 / 20</b>	

Below the table, there is a green 'Ajouter une note' button and a grey 'Retour à la liste des étudiants' button.

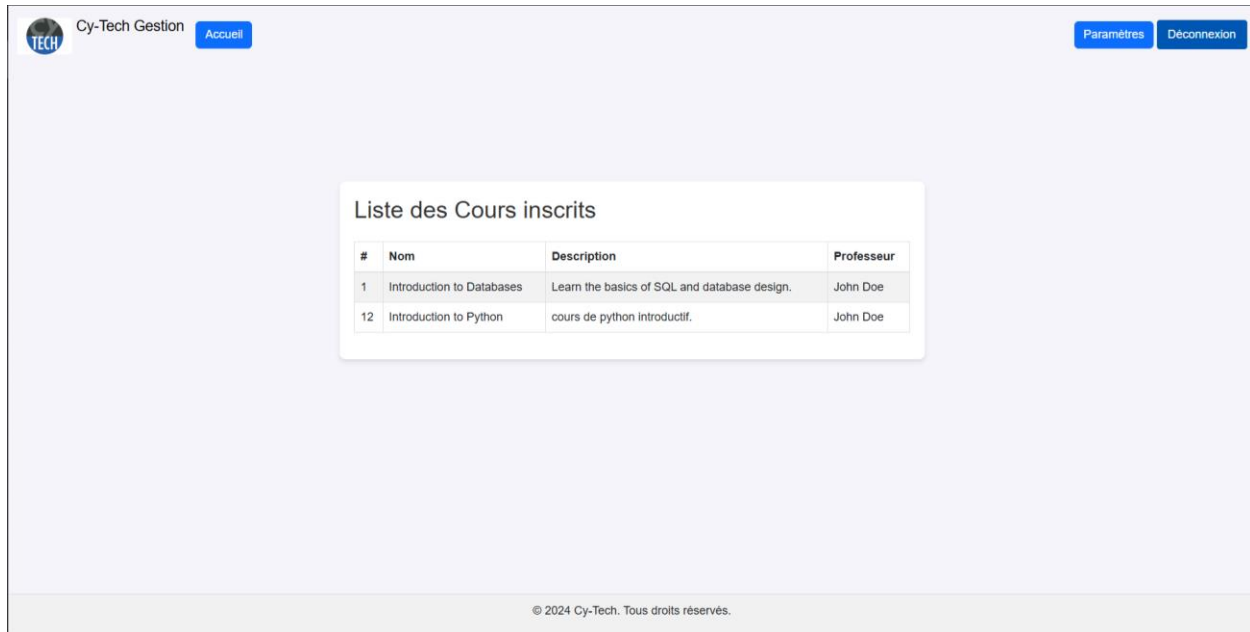
Cette page permet de gérer les évaluations d'un étudiant dans un cours, de modifier ou supprimer des évaluations existantes, d'ajouter de nouvelles évaluations, et d'afficher la moyenne de l'étudiant pour le cours.

Capture d'écran de studentDashboard.jsp :

The screenshot displays the 'studentDashboard.jsp' page. The header is identical to the previous page. The main content area features a white box with the text 'Bienvenue sur votre espace étudiant, Jane Smithy!' and 'Que voulez-vous faire ?'. Below this text are two blue buttons: 'Voir les cours' and 'Voir les notes'. At the bottom of the page, there is a footer that reads '© 2024 Cy-Tech. Tous droits réservés.'

Cette page est destinée à l'espace personnel de l'étudiant, où il peut consulter ses cours ainsi que ses résultats académiques. Cette page fournit à l'étudiant un accès facile à ses cours et à ses résultats, dans un environnement simple et fonctionnel, avec des informations personnalisées en fonction de son profil.

Capture d'écran de courses.jsp :

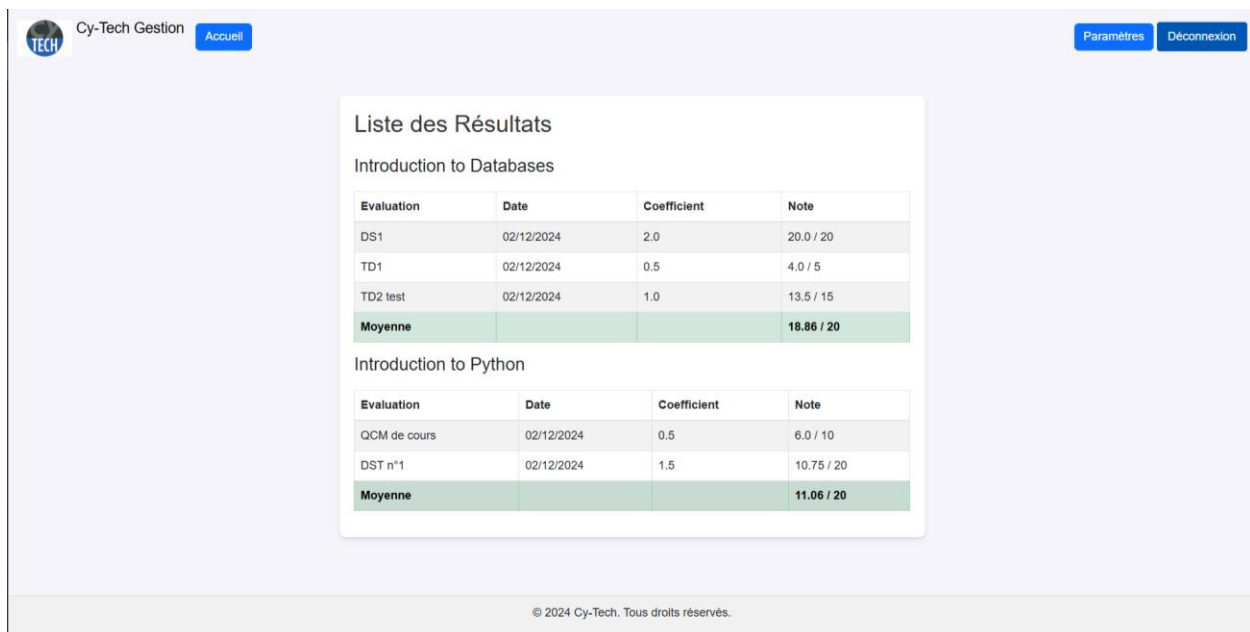


#	Nom	Description	Professeur
1	Introduction to Databases	Learn the basics of SQL and database design.	John Doe
12	Introduction to Python	cours de python introductif.	John Doe

© 2024 Cy-Tech. Tous droits réservés.

Cette page permet d'afficher une liste dynamique des cours auxquels l'étudiant est inscrit. Les cours sont affichés dans un tableau avec leurs détails, et des informations de gestion des erreurs sont également prises en compte.

Capture d'écran de results.jsp :



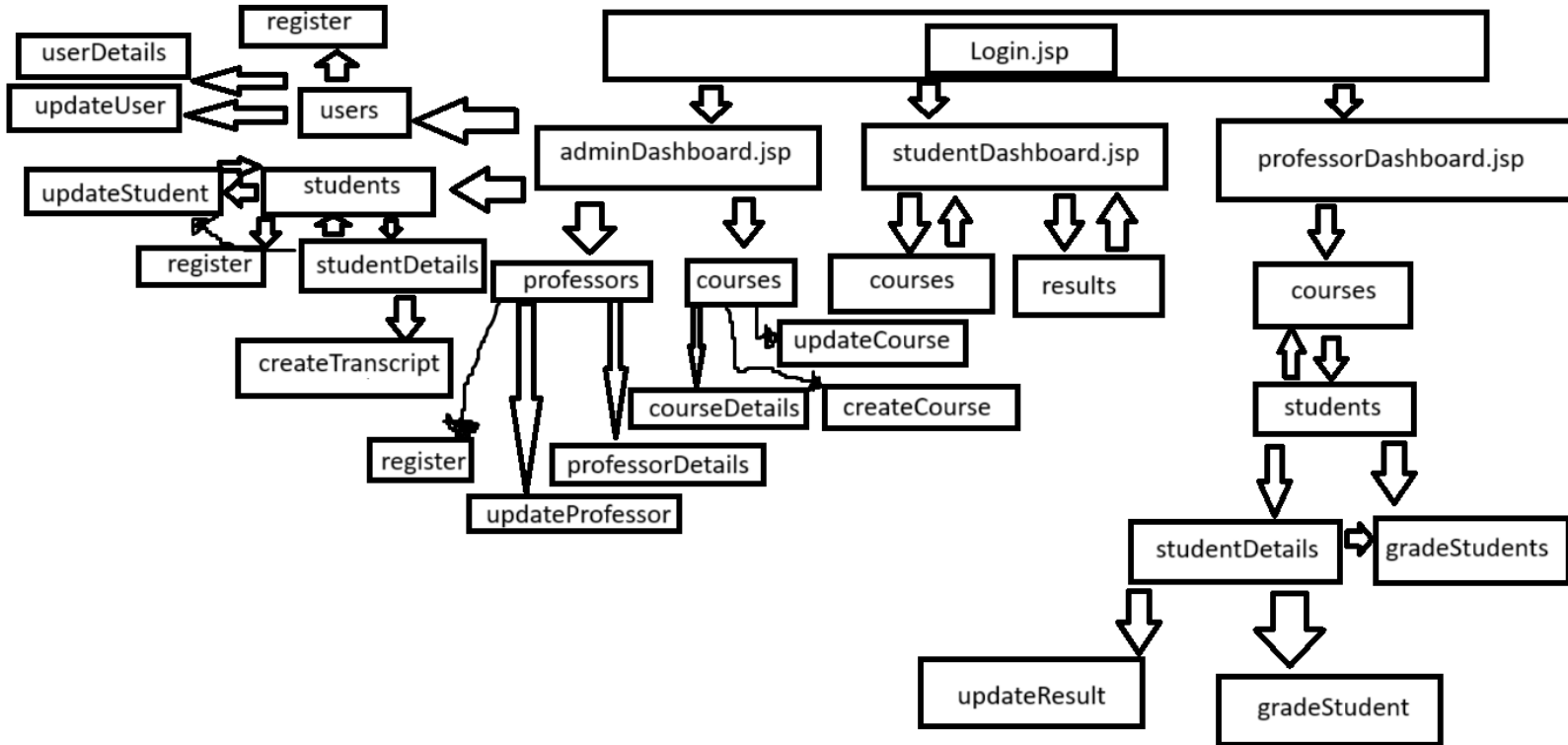
Evaluation	Date	Coefficient	Note
DS1	02/12/2024	2.0	20.0 / 20
TD1	02/12/2024	0.5	4.0 / 5
TD2 test	02/12/2024	1.0	13.5 / 15
<b>Moyenne</b>			<b>18.86 / 20</b>

Evaluation	Date	Coefficient	Note
QCM de cours	02/12/2024	0.5	6.0 / 10
DST n°1	02/12/2024	1.5	10.75 / 20
<b>Moyenne</b>			<b>11.06 / 20</b>

© 2024 Cy-Tech. Tous droits réservés.

Cette page affiche les résultats de l'étudiant trié par cours, mais également sa moyenne dans chaque matière et leurs coefficients.

Voici un schéma résumant les liens/chemins entre les vues :



Les flèches représentent la navigation sur l'application (action en appuyant sur un bouton), il ne faut pas oublier qu'il y a un bouton déconnexion sur le header donc on peut retourner à login.jsp via n'importe quel page. Le traitement des requêtes http (actions) est géré par les contrôleurs

### Services :

Les services servent de couche intermédiaire entre les DAO et la logique métier de l'application, on va synthétiser les plus « intéressants » :

ResultService : gère les résultats académiques des étudiants. Il facilite les opérations CRUD et permet de regrouper les résultats par cours ou par étudiant. Par exemple la méthode `getResultsByStudentIdGroupeByCourse` récupère les résultats d'un étudiant ainsi que leurs cours associés puis les regroupe par cours dans une structure de type `Map<Course, List<Result>>`.

Les services `PerformanceReportService` et `TranscriptService` permettent la génération des pdf, des relevés de notes et de rapport de performance.

### DAO :

Les DAO sont des ponts entre l'application et la base de données, ils permettent la connexion à la base

de données, de récupérer les informations, les supprimer, en ajouter ou les modifier. On utilise la session Hibernate pour exécuter les opérations sur la BDD. Chaque méthode des DAO ouvre une session Hibernate, exécute la requête, et ferme la session pour libérer les ressources.

### **Voici quelques classes intéressantes à analyser :**

EmailUtil.java est une classe utilitaire conçue pour gérer l'envoi d'e-mails via un serveur SMTP ainsi que pour valider les adresses e-mail dans une application. Cette classe est essentielle pour permettre la notification par mail des changements de cours ou d'inscriptions.

HibernateUtil.java permet l'initialisation de la SessionFactory en utilisant le fichier de configuration Hibernate qui contient les informations nécessaires à la connexion à la base de données. Elle possède un getter permettant d'accéder à l'instance de SessionFactory.

ServletUtil.java est une classe utilitaire qui regroupe des méthodes courantes pour simplifier la gestion des requêtes et des réponses http. Comme par exemple la méthode invalidAction, retournant une réponse http avec un code 400 et un message d'erreur ou encore la méthode forward qui transfère une requête vers une autre page.

### **Models :**

Les models sont comme dans le diagramme de classes, ils possèdent également leurs getters et setters pour faciliter leur manipulation.

### **Controller :**

Les contrôleurs ont pour rôle principal de gérer la logique métier de l'application et de contrôler l'interaction entre vues et les modèles. Donc ils reçoivent les requêtes http envoyés par les utilisateurs. En fonction de la requête le contrôleur invoque la méthode nécessaire pour exécuter la logique métier. Ils permettent aussi de vérifier la qualité des données récupérer avant de modifier la base de données, ou de connecter un utilisateur.

### **Système de notification par email :**

Système de notification par email pour informer les étudiants des changements dans leur inscription ou de la publication des notes. Cette fonctionnalité est gérée par la classe EmailUtil.java, le contrôleur EnrollmentController qui écrit le message puis appelle la méthode sendEmail lors d'un changement d'inscription. Dans le même principe le contrôleur StudentController écrit le message et appelle la fonction pour envoyer le mail lors d'une publication de note ou un changement dans l'inscription.

### **Téléchargement du relevé de note :**

La génération du relevé de note est gérée par le contrôleur TranscriptServlet qui en utilisant la bibliothèque itext, qui appelle la méthode generate du service TranscriptService qui génère le pdf et écrit les informations données par le contrôleur dedans.

### **Génération de rapports de performances :**

Même principe que précédemment mais avec le contrôleur PerformanceReportServlet et le service PerformanceReportService.

## 4-Spring Boot

L'implémentation de Spring Boot fut délicate, il y a eu beaucoup à réécrire. Premièrement reconfigurer les dépendances Spring Boot dans le fichier pom.xml. On a également dû créer une classe principale qui démarre le serveur Spring Boot. La partie délicate fut la réécriture de tout les contrôleurs et des services. Rectifier les chemins dans les contrôleurs et les vues. Nouvelles fonctionnalités pratique sont les repository, ils permettent de gérer les opérations classiques (CRUD) sans avoir à écrire de code SQL. Les repositories sont intégrés aux services dans l'architecture Spring Boot. Ils servent à découpler la logique métier de l'application de la logique d'accès aux données.



## 5-Conclusion

Pour conclure, ce projet a globalement été une réussite, malgré les défis rencontrés liés à notre effectif réduit, étant seulement deux dans le groupe. Cette contrainte nous a permis de développer une meilleure coordination et de maximiser nos efforts individuels pour atteindre les objectifs fixés.

Au cours de ce projet, nous avons mis en pratique des concepts fondamentaux de développement en J2EE, tels que l'architecture MVC, la gestion des bases de données avec Hibernate, et l'intégration des technologies front-end avec JSP et Servlet. Chaque étape, de l'analyse des besoins jusqu'au déploiement, nous a offert une perspective concrète sur les cycles de développement logiciel et les exigences d'un projet en milieu professionnel.

Malgré quelques difficultés techniques et organisationnelles, nous avons pu les surmonter grâce à une collaboration efficace et une approche méthodique. Ce projet nous a également permis de renforcer nos compétences en résolution de problèmes, en gestion de projet, et en développement d'applications robustes et performantes.

En somme, cette expérience a non seulement consolidé nos acquis techniques, mais elle nous a également préparés à relever des défis similaires dans un cadre professionnel, avec une meilleure autonomie et une meilleure capacité d'adaptation.