

03/04/2017

Projet Bases de données : Application de gestion d'un établissement scolaire

David HA, Charles ECOLLAN et Quentin
LECAILLE
Groupe A – Promo 2019 L'3



Sous la direction de M. RAMDANE-CHERIF

Énoncé

Ce programme de gestion d’une école devra être réalisé entièrement en Java ou tout autre langage en accord avec votre chargé de TP.

Vous êtes totalement libre des fonctions à implémenter.

Résumé

Ce projet consiste à réaliser un programme de gestion d’un établissement scolaire. L’utilisateur se connecte en précisant s’il est étudiant, professeur ou administrateur (membre de l’administration). Lorsque l’utilisateur se connecte, en fonction de son rôle dans l’établissement (les trois catégories données précédemment), il voit un panel avec toutes les fonctionnalités disponibles. Pour donner un exemple, l’administrateur peut créer un élève, un enseignant et un administrateur tandis que les autres non. Le projet est une application web faites en HTML/CSS pour la forme et en PHP/MySQL pour le fond et l’accès à la base de données (phpMyAdmin).

Toutes les fonctions du cahier des charges ont été implémentées. On peut, par exemple, créer une matière et lui assigner un coefficient, mais aussi assigner des notes à des élèves par rapport à une épreuve de passée.

L’application se trouve à l’adresse suivante : <http://www.charles-dev.com/> . Pour entrer dans l’application avec un identifiant administrateur :

- Identifiant : charles.ecollan@outlook.com
- Mot de passe : azerty (par défaut)

Elle peut être utilisée en local en important la base de données vide (il faut juste créer un identifiant admin par phpMyAdmin : `INSERT INTO `bdd`.`administrateur` (`nom_admin`, `prenom_admin`, `id_admin`, `email_admin`, `password`) VALUES ('Ecollan', 'Charles', '1', 'charles.ecollan@outlook.com', 'azerty');`

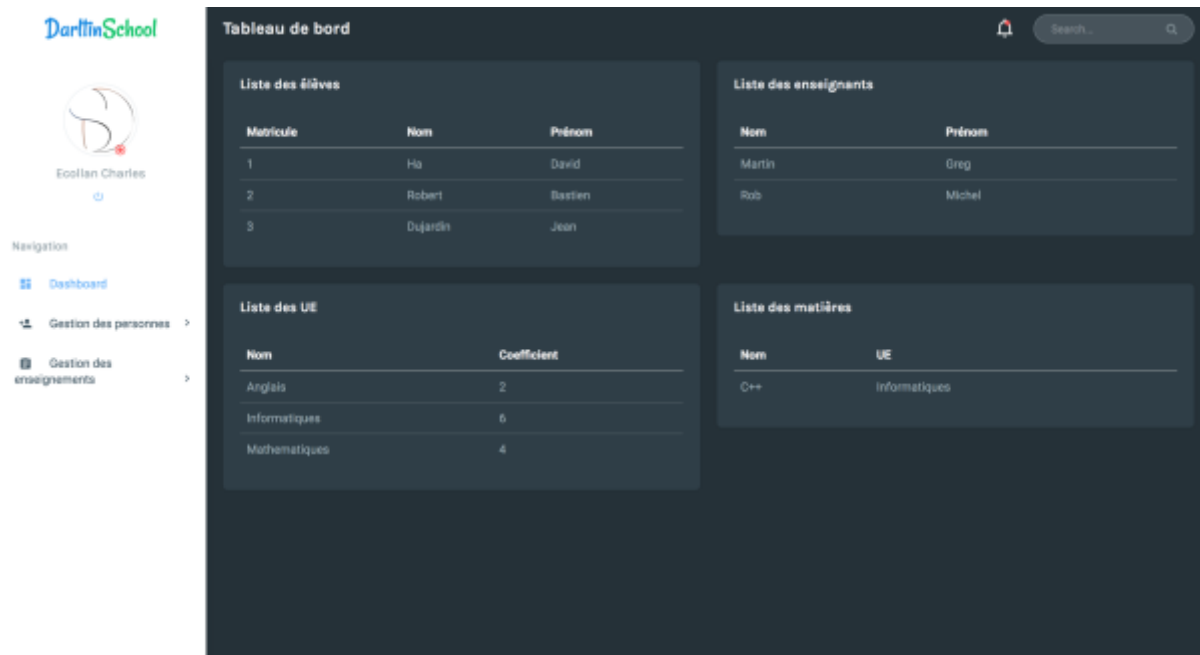


Tableau de bord

Liste des élèves

Matricule	Nom	Prénom
1	Ha	David
2	Robert	Bastien
3	Dujardin	Jean

Liste des enseignants

Nom	Prénom
Martin	Greg
Rob	Michel

Liste des UE

Nom	Coefficient
Anglais	2
Informatiques	6
Mathématiques	4

Liste des matières

Nom	UE
C++	Informatiques

Table de matière

Table des matières

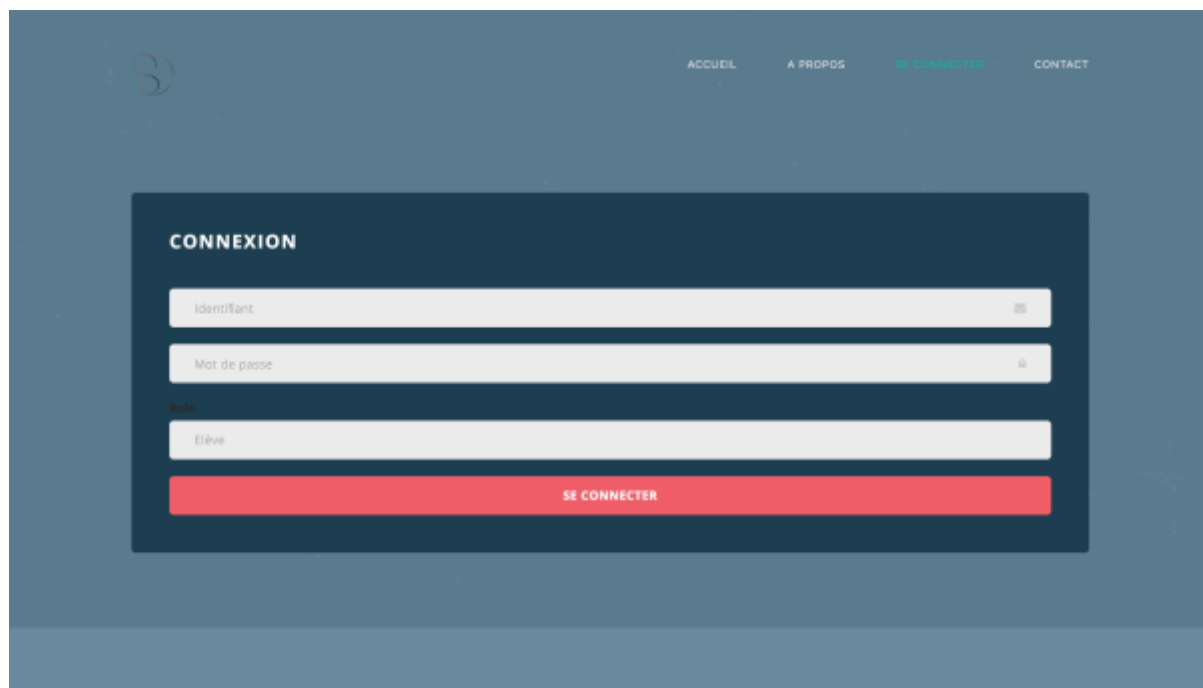
ÉNONCE	0
RESUME	2
INTRODUCTION	4
ANALYSE	5
CAHIER DES CHARGES	5
ANALYSE DE LA BASE DE DONNEES	5
<i>Dictionnaire de données</i>	5
<i>MCD</i>	7
<i>MLD</i>	7
<i>Script SQL</i>	7
LANGAGES	8
<i>HTML</i>	8
<i>PHP</i>	8
<i>MySQL</i>	8
<i>CSS</i>	9
<i>JavaScript</i>	9
DEVELOPPEMENT	10
PLAN DE L’APPLICATION	10
CONNEXION	12
BACK-END	12
ADMINISTRATEUR	13
<i>Gestion des personnes</i>	13
<i>Gérer les enseignements</i>	16
ÉLÈVES	18
CONCLUSION	20
ANNEXE 1 : SCRIPT SQL	21

Introduction

Ce projet a pour but de gérer un établissement scolaire (étudiant, enseignant et administrateur). Il a permis de faire appel à tout ce que l’on a appris en Bases de données (1), c’est-à-dire les requêtes SQL, et en Bases de données (2), c’est-à-dire les MCD et MLD.

Le but était donc de réaliser l’interface graphique ainsi qu’implémenter les différentes fonctionnalités définies dans le cahier des charges.

Il a fallu réaliser ce projet dans le cadre des séances de TP encadrés mais aussi à l’extérieur. Afin de réaliser ce travail à trois, nous avons utilisé *git* pour permettre de travailler chacun sur une partie et de l’envoyer directement par l’intermédiaire de ce logiciel. De plus, étant un logiciel de gestion de versions, si un problème apparaissait, on pouvait revenir sur une phase précédente sans soucis.



The screenshot shows a web application interface with a dark blue background. At the top, there is a navigation bar with links: ACCUEIL, A PROPOS, SE CONNECTER, and CONTACT. The main content area features a dark blue box with the title 'CONNEXION'. Inside this box, there are three input fields: 'Identifiant', 'Mot de passe', and 'Rôle'. The 'Rôle' field has a dropdown menu with 'Elève' selected. Below the input fields is a red button labeled 'SE CONNECTER'.

Analyse

Cahier des charges

L’application doit permettre de gérer un établissement scolaire. Nous avons donc pris le modèle de l’EFREI où l’administration a accès à la totalité des fonctionnalités et entre les notes des élèves. L’application doit pouvoir gérer :

- Les étudiants
Consulter et modifier son dossier.
Consulter ses notes.
- Les administrateurs
Créer des élèves et des enseignants.
Créer des matières, des UE et des promotions.
Ajouter des notes à des élèves.
Créer des épreuves.
Assigner un UE à une promotion.
Assigner une matière à un professeur.

Analyse de la base de données

Dictionnaire de données

Sigle	Libellé	Type
Nom_contact	Nom	Varchar
Prenom_contact		Varchar
Num_rue_contact		Int
Nom_rue_contact		Varchar
Code_postal_contact		Int
Ville_contact		Varchar
Tel_contact		Int
Email_contact		Varchar
Id_contact		Int
Convoc_parent		Longtext
Id_document		Int
Annee_debut		Int
Annee_fin		Int
Matricule		Int
Nom		Varchar
Prenom		Varchar
Date_naissance		Date
Ville_naissance		Varchar
Pays_naissance		Varchar

Projet –Application de gestion d’un établissement scolaire

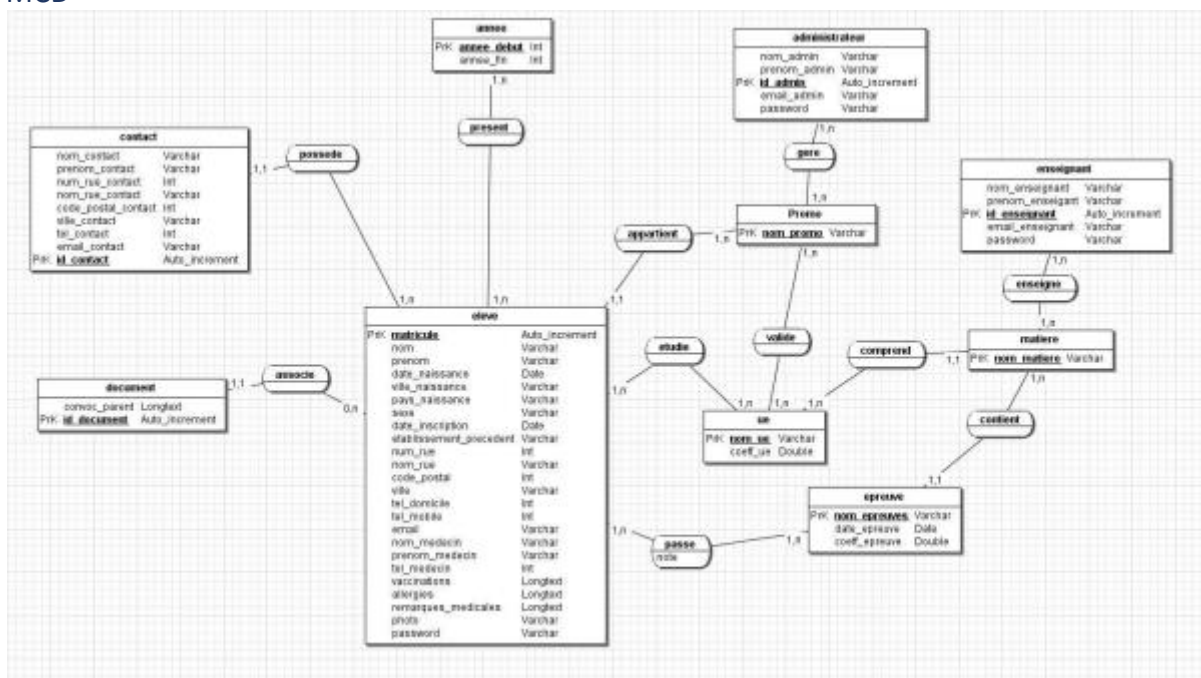
Bases de données

Sexe		Varchar
Date_inscription		Date
Etablissement_precedent		Varchar
Num_rue		Int
Nom_rue		Varchar
Code_postal		Int
Ville		Varchar
Tel_domicile		Int
Tel_mobile		Int
Email		Varchar
Nom_medecin		Varchar
Prenom_medecin		Varchar
Tel_medecin		Int
Vaccinations		Longtext
Allergies		Longtext
Remarques_medicales		Longtext
Photo		Varchar
Password		Varchar
Note		Int
Nom_admin		Varchar
Prenom_admin		Varchar
Id_admin		Int
Email_admin		Varchar
Password		Varchar
Nom_promo		Varchar
Nom_ue		Varchar
Coeff_ue		Double
Nom_epreuves		Varchar
Date_epreuve		Date
Coeff_epreuve		Double
Nom_matiere		Varchar
Nom_enseignant		Varchar
Prenom_enseignant		Varchar
Id_enseignant		Int
Email_enseignant		Varchar
Password		Varchar

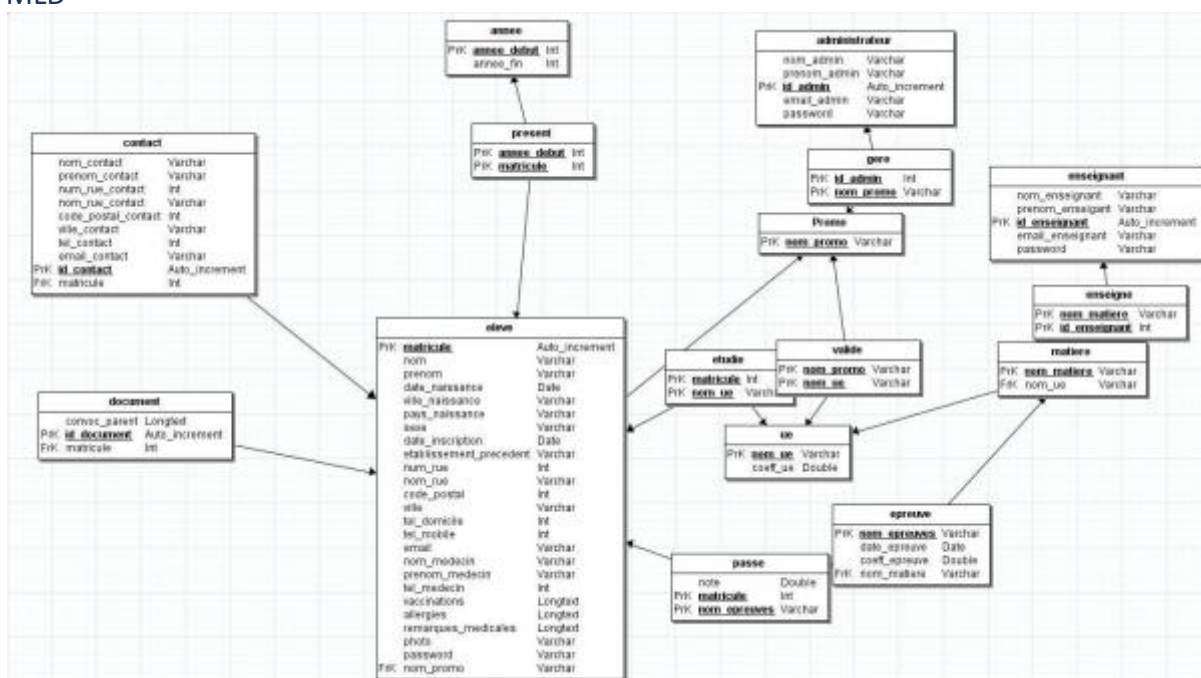
Projet – Application de gestion d'un établissement scolaire

Bases de données

MCD



MLD



Script SQL

Voir annexe 1

Langages

Ci-dessous, la liste des langages utilisés dans un tableau avec les principales utilisations afin d’avoir une vue plus générale.

Langage	Utilisation(s)
HTML	L’affichage des différentes pages du site web.
CSS	La présentation des différentes pages du site.
JavaScript	Contrôle des champs des formulaires.
PHP	Interaction avec le serveur. Permet le développement des pages dynamiques (relation avec la base de données)
MySQL	Interroger la base de données

HTML

Le HTML, qui signifie Hypertext Markup Language, est un langage de balisage pour l’écriture de l’hypertexte. Il permet de créer un site dit « statique ». On entend par page statique, non pas une page sans mouvement ou sans animations, mais une page visible telle qu’elle a été conçue. Ces pages peuvent présenter toute forme de contenu, animations flash, images, musiques ou vidéos, mais elles sont toujours présentées de la même façon. Elles ne changent pas et c’est en ce sens qu’elles sont statiques. Il faut différencier le statique du dynamique. Effectivement, une page dynamique est une page qui s’affiche en fonction des requêtes des utilisateurs, telles que les zones de saisies et les listes déroulantes.

PHP

Le PHP, Hypertext Preprocessor, est un langage de programmation permettant de faire un site dynamique, c’est-à-dire en affichant des données, par exemple, venant d’une base de données. Cela permet de mettre un lien entre le serveur et le client. Par exemple, la page de connexion pour sécuriser l’accès à l’application est faite en PHP.

MySQL

MySQL, qui signifie My Structured Query Language, est un serveur de bases de données relationnelles SQL. Ici, on l’a couplé au PHP afin d’interroger, d’insérer et de mettre à jour la base de données.

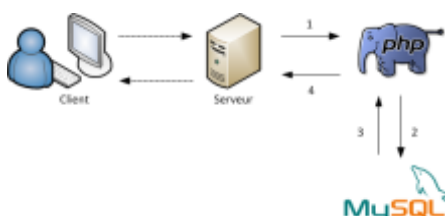


Figure 1 : Schéma récapitulatif HTML-PHP-MySQL

CSS

Le CSS, signifiant Cascading Style Sheets ou en français Feuilles de Style en Cascade, permet tout simplement de mettre en page les pages de l’application.

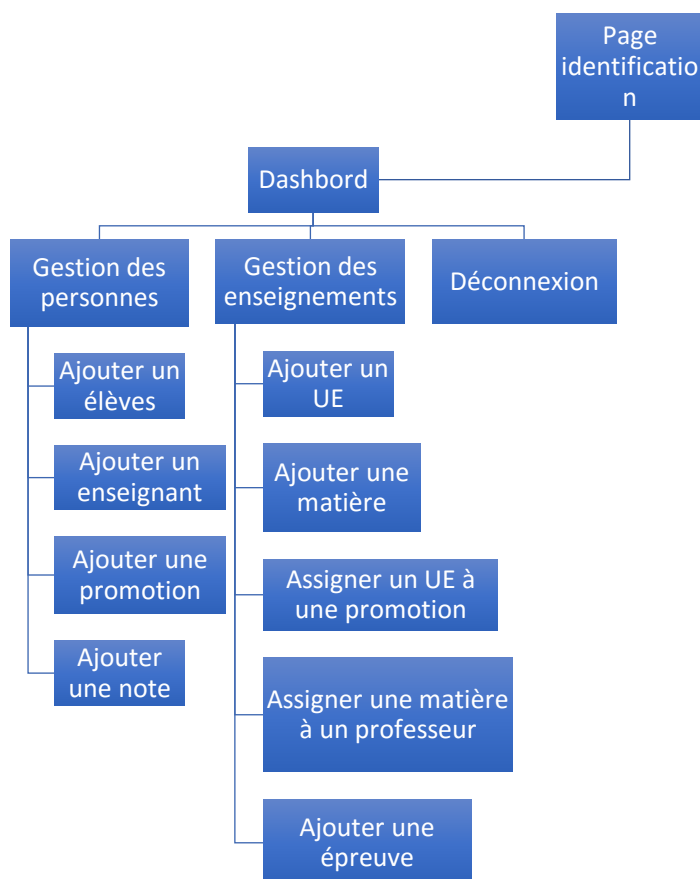
JavaScript

Le JavaScript est un langage de programmation permettant de faire des événements du côté du client.

Développement

Plan de l’application

Est établi un plan des différentes pages du site ci-dessous à créer pour se rendre compte du placement de telles et telles fonctions, mais aussi pour adopter une stratégie de conception claire et nette. Ce plan concerne la vue de l’administrateur.



Les grands points sont présentés ci-dessus. Au travers de ce diagramme, l'utilisateur va devoir s'authentifier avant de pouvoir accéder à l'application. Il se retrouve alors sur l'accueil et a le choix avec deux grandes possibilités distinctes :

- Gérer les personnes

Gérer les personnes, c'est-à-dire créer des étudiants et des enseignants. Mais aussi créer une promotion et ajouter une note à un élèves en fonction d'une épreuve donnée.

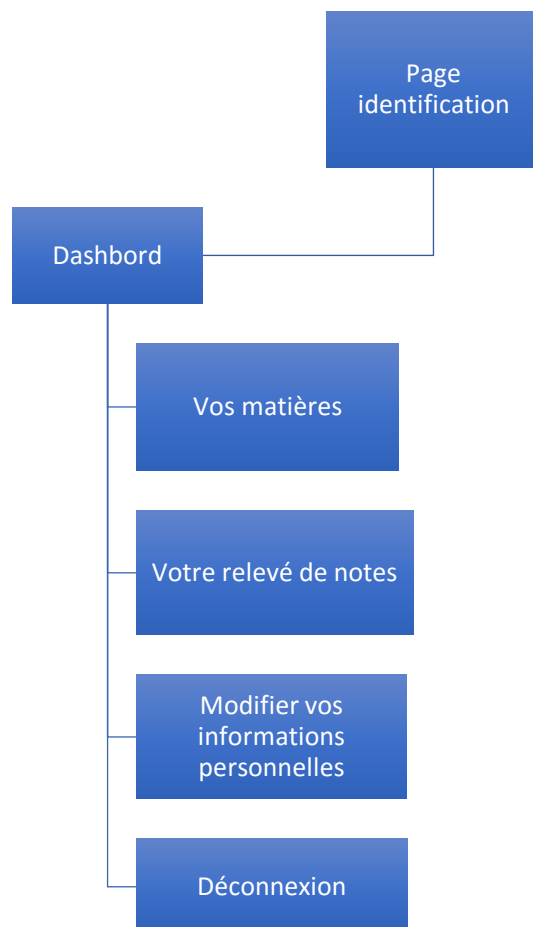
- Gérer les enseignements

Gérer les enseignements, c'est-à-dire créer un UE et une matière. Les assigner entre eux. Mais aussi assigner une matière à un professeur et ajouter une épreuve afin de pouvoir mettre des notes aux élèves.

- Se déconnecter

Quitter l'application.

Voyons maintenant un plan concernant la vue de l’étudiant.

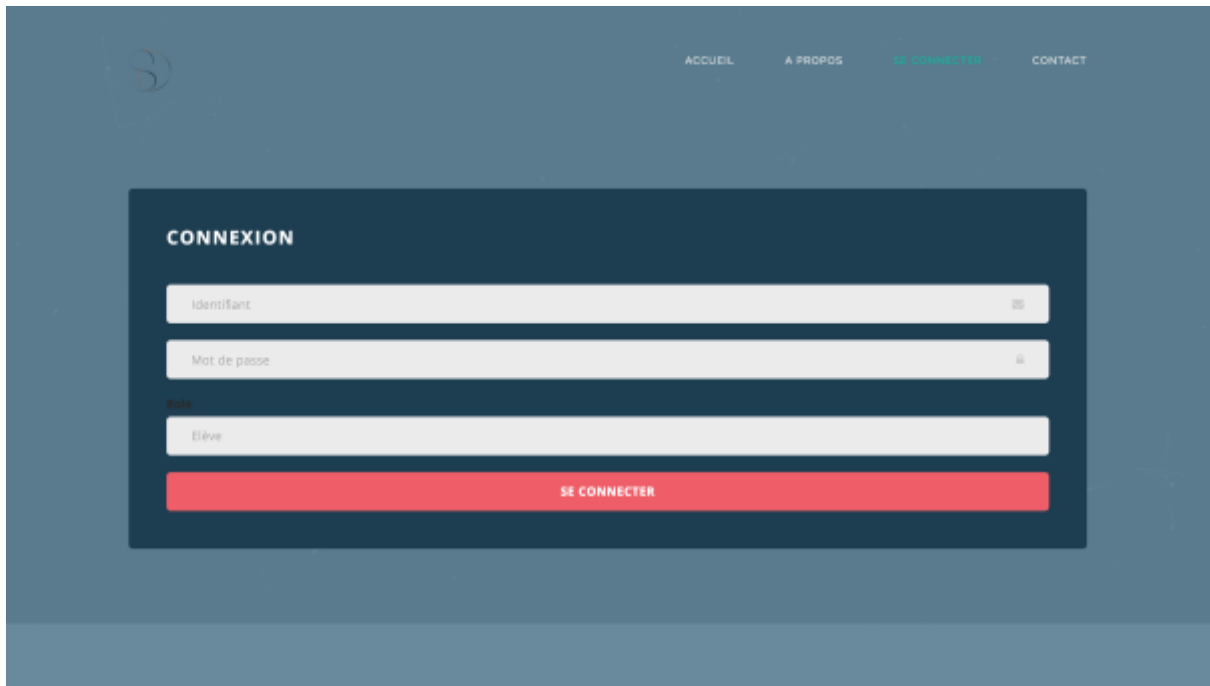


Les grands points sont présentés ci-dessus. Au travers de ce diagramme, l'utilisateur va devoir s'authentifier avant de pouvoir accéder à l'application. Il se retrouve alors sur l'accueil et a le choix avec quatre grandes possibilités distinctes :

- Vos matières
Accéder à toutes les matières que l'étudiant a.
- Votre relevé de notes
Permet de connaître les notes que l'étudiant a.
- Modifier vos informations personnelles
Modifier le dossier de l'élève (son dossier du coup, comme les informations personnelles afin de les mettre à jour).
- Se déconnecter
Quitter l'application.

Connexion

La page de connexion est à part. En effet, c’est elle qui va permettre à l’utilisateur de rentrer dans l’application. Il faut d’abord créer la vue de la page ainsi que le formulaire de connexion afin de pouvoir envoyer les données au contrôleur qui va lui-même appeler le modèle pour interroger la base de données. Si l’utilisateur existe et que le mot de passe rentré est correct, alors on redirige l’utilisateur vers la vue de l’accueil de l’application correspondant à son profil : étudiant, enseignant ou administrateur. Sinon, on affiche un message d’erreur.



Back-End

Nous avons maintenant un site dit statique, c’est-à-dire qui n’affiche que ce que l’on a écrit dans les différentes vues. Afin de rendre le site dynamique et de mettre un lien avec la base de données, nous allons passer à un site dynamique en utilisant trois choses :

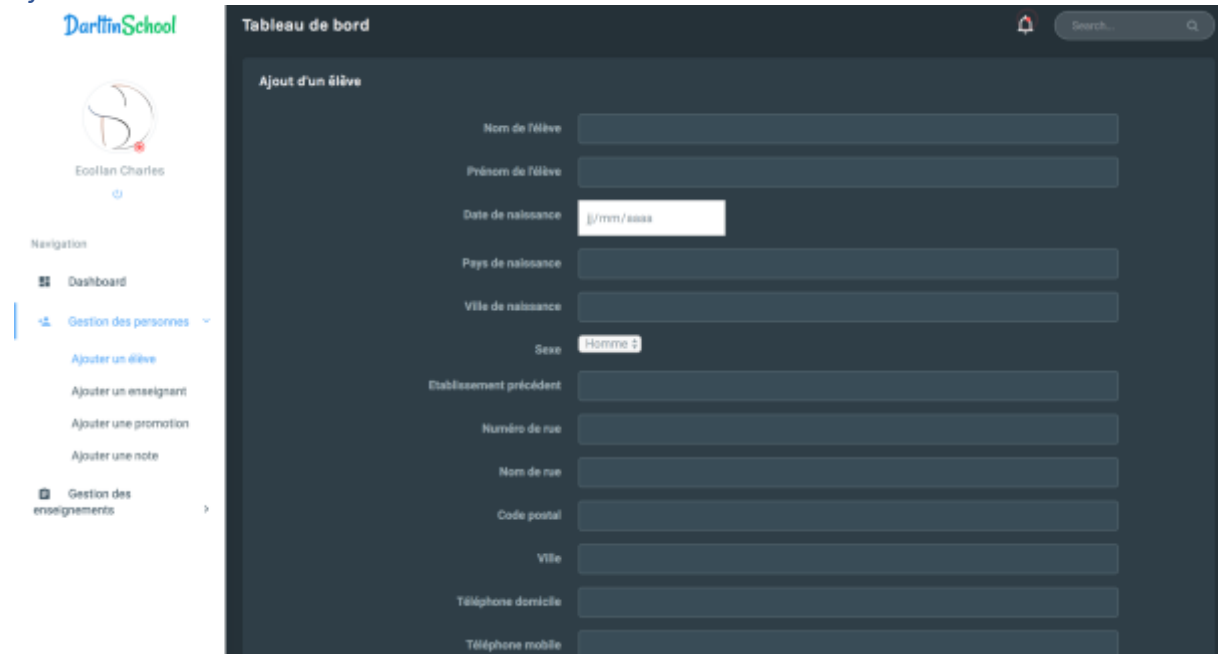
- Le serveur. Dans notre cas, l’application est hébergée sur un serveur chez OVH. Toutes les pages y sont stockées ainsi que la base de données.
- Le site web. On parle ici de la partie cachée (requêtes...)
- La base de données. Elle stocke toutes les personnes (administrateurs, enseignants et élèves), les notes, les matières...

Les langages utilisés sont donc le PHP et le MySQL.

Administrateur

Gestion des personnes

Ajouter une élève



The screenshot shows the 'Ajout d'un élève' (Add a student) form within the Darl'inSchool application. The interface is dark-themed. On the left, there is a sidebar with the 'Darl'inSchool' logo, a user profile for 'Ecollan Charles', and a navigation menu with options like 'Dashboard', 'Gestion des personnes', 'Ajouter un élève', 'Ajouter un enseignant', 'Ajouter une promotion', 'Ajouter une note', and 'Gestion des enseignements'. The main area is titled 'Tableau de bord' and contains the 'Ajout d'un élève' form. The form fields are as follows:

Field Label	Input Type
Nom de l'élève	Text
Prénom de l'élève	Text
Date de naissance	Text (with date format hint: dd/mm/yyyy)
Pays de naissance	Text
Ville de naissance	Text
Sexe	Dropdown (selected: Homme)
Établissement précédent	Text
Numéro de rue	Text
Nom de rue	Text
Code postal	Text
Ville	Text
Téléphone domicile	Text
Téléphone mobile	Text

Pour ajouter un élève, il faut remplir les informations suivantes :

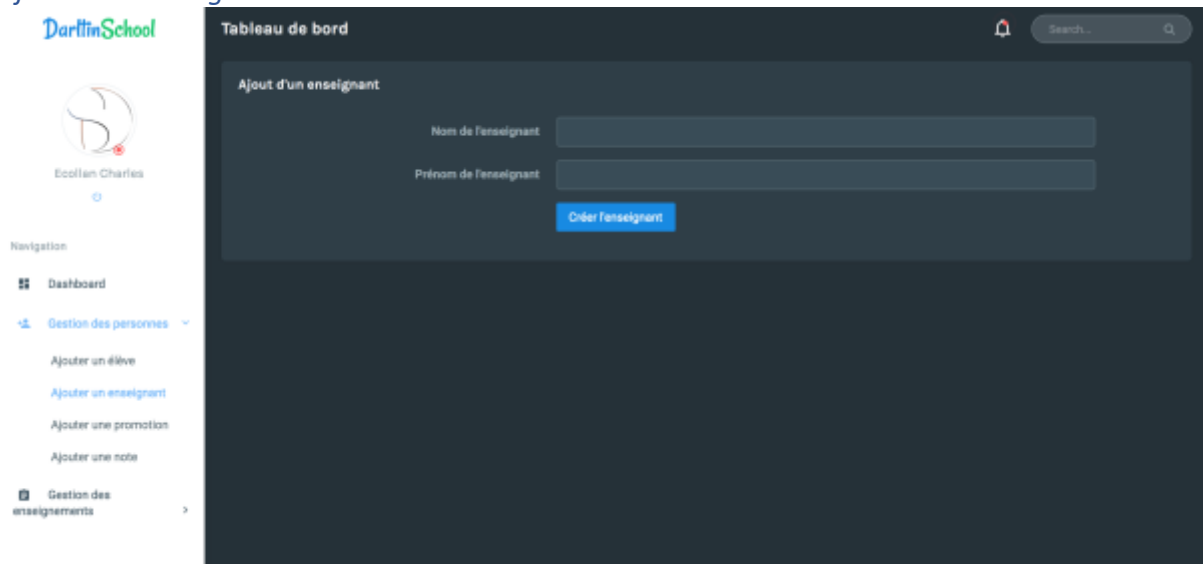
- Nom de l'élève
- Prénom de l'élève
- Date de naissance
- Pays de naissance
- Ville de naissance
- Sexe
- Établissement précédent
- Numéro de rue
- Nom de rue
- Code postal
- Ville
- Téléphone domicile
- Téléphone mobile
- Nom du médecin

Projet –Application de gestion d’un établissement scolaire
Bases de données

- Prénom du médecin
- Téléphone du médecin
- Allergie(s)
- Vaccination(s)
- Remarque(s) médicale(s)
- Promotion (choix menu si promotion créée)

L'email est automatiquement créé à la création de l'élèves : prenom.nom@darltin.com ainsi que le mot de passe : azerty.

Ajouter un enseignant



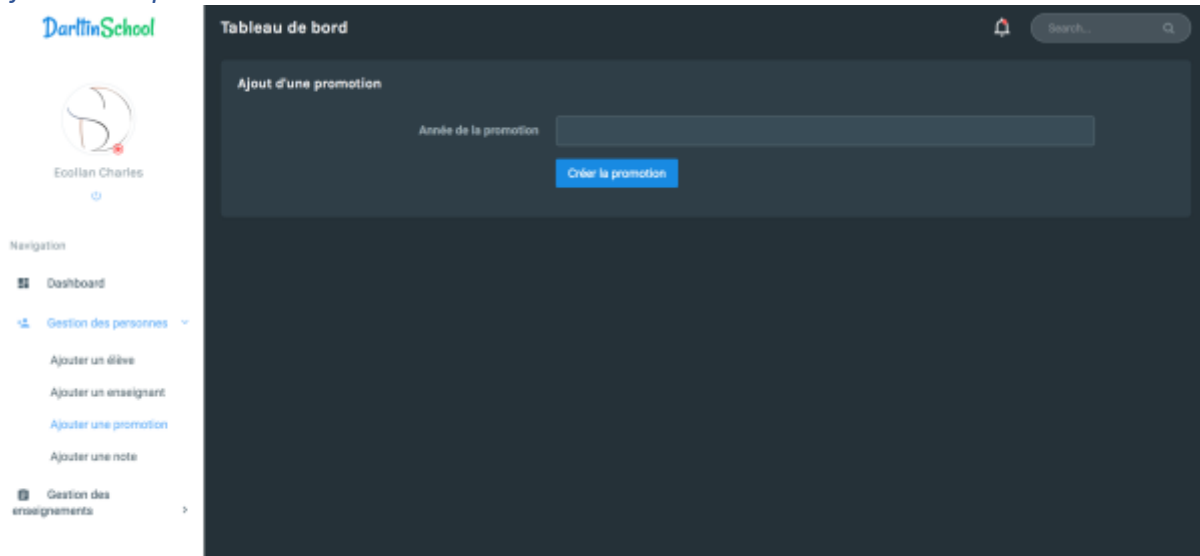
The screenshot shows the 'Tableau de bord' (Dashboard) of the DarltinSchool application. On the left is a navigation menu with 'Gestion des personnes' expanded, showing options like 'Ajouter un élève', 'Ajouter un enseignant', 'Ajouter une promotion', and 'Ajouter une note'. The main area displays a form titled 'Ajout d'un enseignant' with two input fields: 'Nom de l'enseignant' and 'Prénom de l'enseignant', followed by a blue 'Créer l'enseignant' button.

Pour ajouter un enseignant, il faut remplir les informations suivantes :

- Nom de l'enseignant
- Prénom de l'enseignant

L'email est automatiquement créé : prenom.nom@darltin.com ainsi que le mot de passe : azerty.

Ajouter une promotion

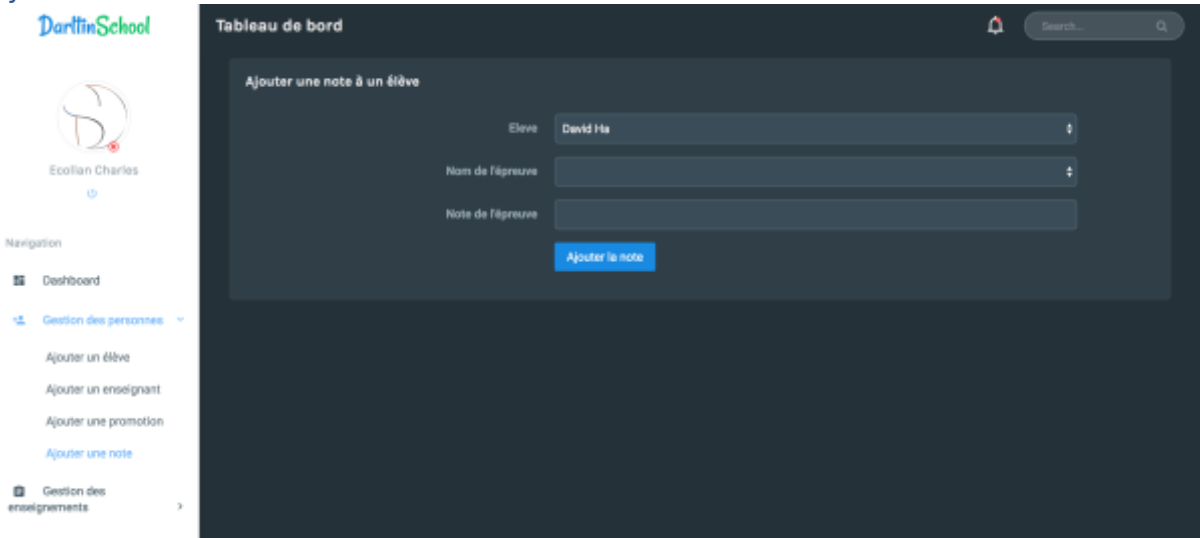


The screenshot shows the 'Ajout d'une promotion' form in the DartlinSchool application. The form is titled 'Ajout d'une promotion' and contains a single input field labeled 'Année de la promotion'. Below the input field is a blue button labeled 'Créer la promotion'. The left sidebar shows the user 'Ecollan Charles' and a navigation menu with options: 'Dashboard', 'Gestion des personnes' (expanded), 'Ajouter un élève', 'Ajouter un enseignant', 'Ajouter une promotion' (highlighted), and 'Ajouter une note'. The top right of the dashboard has a search bar and a notification bell icon.

Pour ajouter une promotion, il faut remplir les informations suivantes :

- Année de la promotion

Ajouter une note



The screenshot shows the 'Ajouter une note à un élève' form in the DartlinSchool application. The form is titled 'Ajouter une note à un élève' and contains three input fields: 'Eleve' (with 'David Ha' selected), 'Nom de l'épreuve', and 'Note de l'épreuve'. Below the input fields is a blue button labeled 'Ajouter la note'. The left sidebar shows the user 'Ecollan Charles' and a navigation menu with options: 'Dashboard', 'Gestion des personnes' (expanded), 'Ajouter un élève', 'Ajouter un enseignant', 'Ajouter une promotion', and 'Ajouter une note' (highlighted). The top right of the dashboard has a search bar and a notification bell icon.

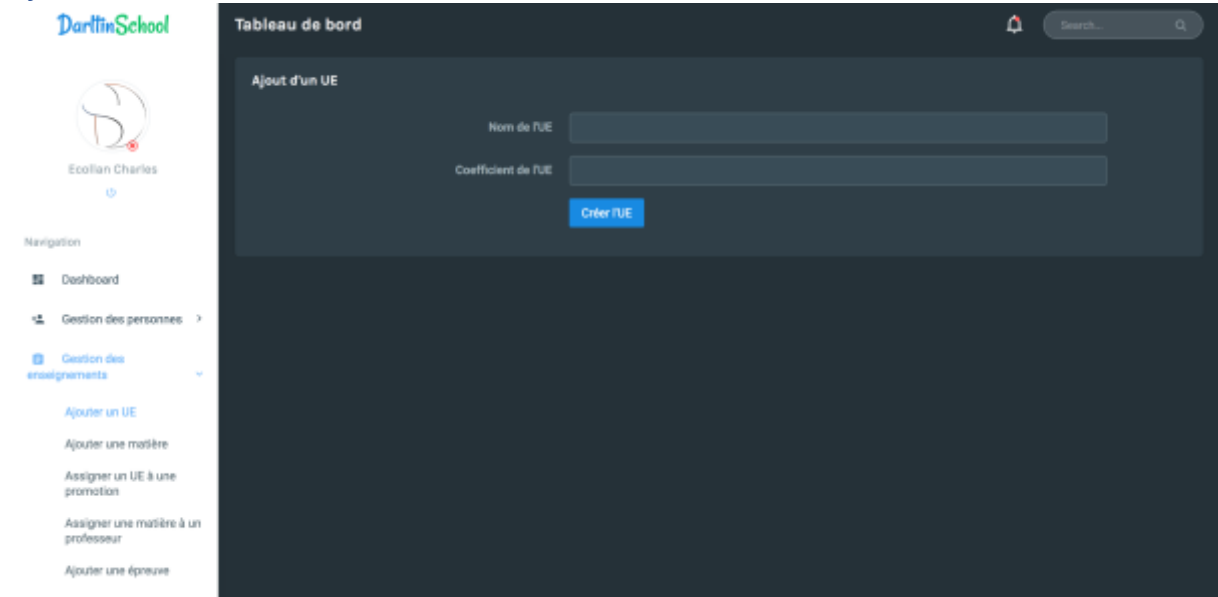
Pour ajouter une note, il faut remplir les informations suivantes :

- Nom de l'élève
- Nom de l'épreuve
- Note de l'épreuve

Pour ajouter une note à un élève, il faut auparavant ajouter une épreuve (expliqué un peu plus loin).

Gérer les enseignements

Ajouter un UE

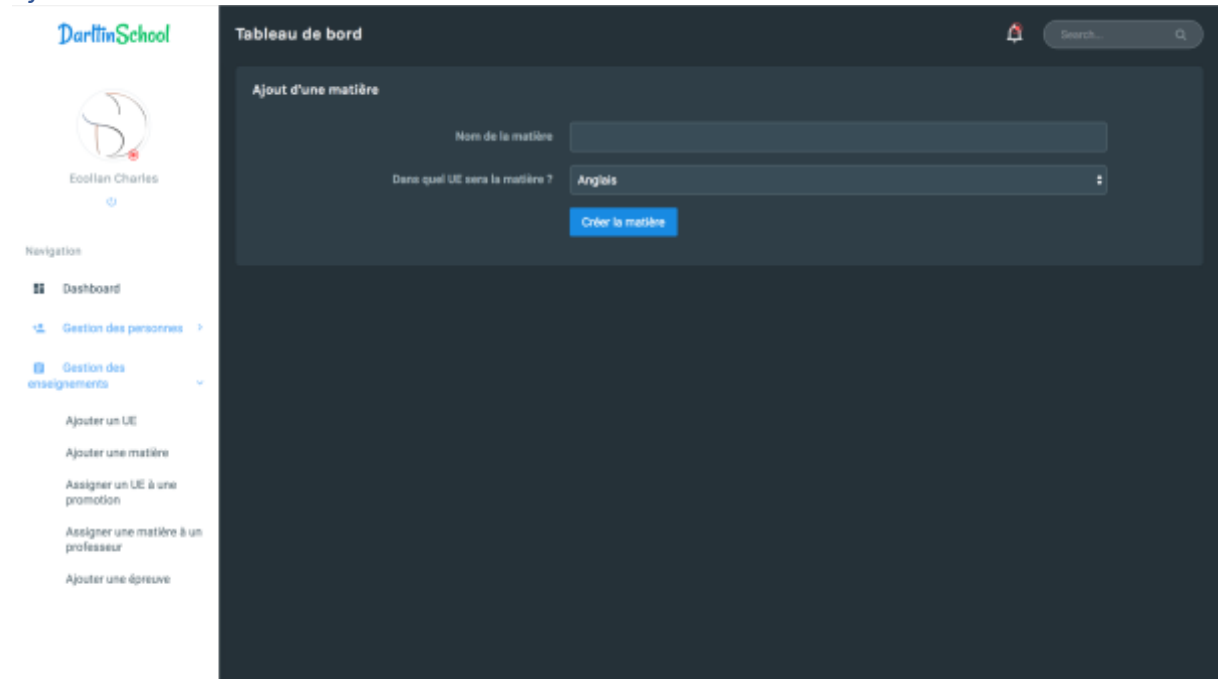


The screenshot shows the 'Ajout d'un UE' (Add a Unit) form in the DartfinSchool application. The form is titled 'Ajout d'un UE' and contains two input fields: 'Nom de l'UE' (Unit Name) and 'Coefficient de l'UE' (Unit Coefficient). Below these fields is a blue button labeled 'Créer l'UE'. The left sidebar shows the navigation menu with 'Gestion des enseignements' selected, and the main content area is titled 'Tableau de bord'.

Pour ajouter un UE, il faut remplir les informations suivantes :

- Nom de l'UE
- Coefficient de l'UE

Ajouter une matière



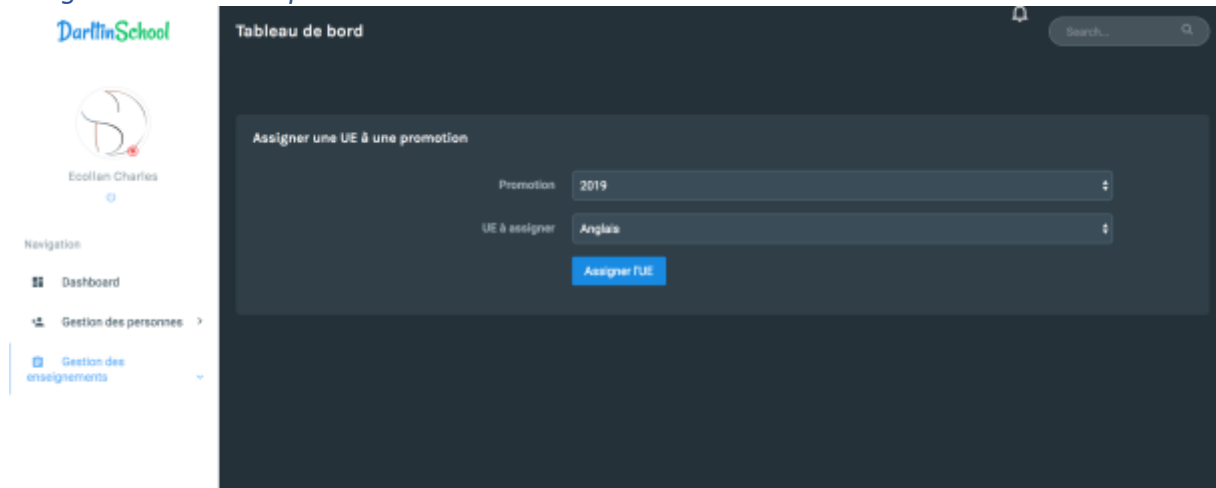
The screenshot shows the 'Ajout d'une matière' (Add a Subject) form in the DartfinSchool application. The form is titled 'Ajout d'une matière' and contains two input fields: 'Nom de la matière' (Subject Name) and 'Dans quel UE sera la matière ?' (In which Unit will the subject be?). Below these fields is a blue button labeled 'Créer la matière'. The left sidebar shows the navigation menu with 'Gestion des enseignements' selected, and the main content area is titled 'Tableau de bord'.

Après avoir ajouté une UE, pour ajouter une matière, il faut remplir les informations suivantes :

Projet –Application de gestion d’un établissement scolaire
 Bases de données

- Nom de la matière
- Choix de l’UE

Assigner un UE à une promotion

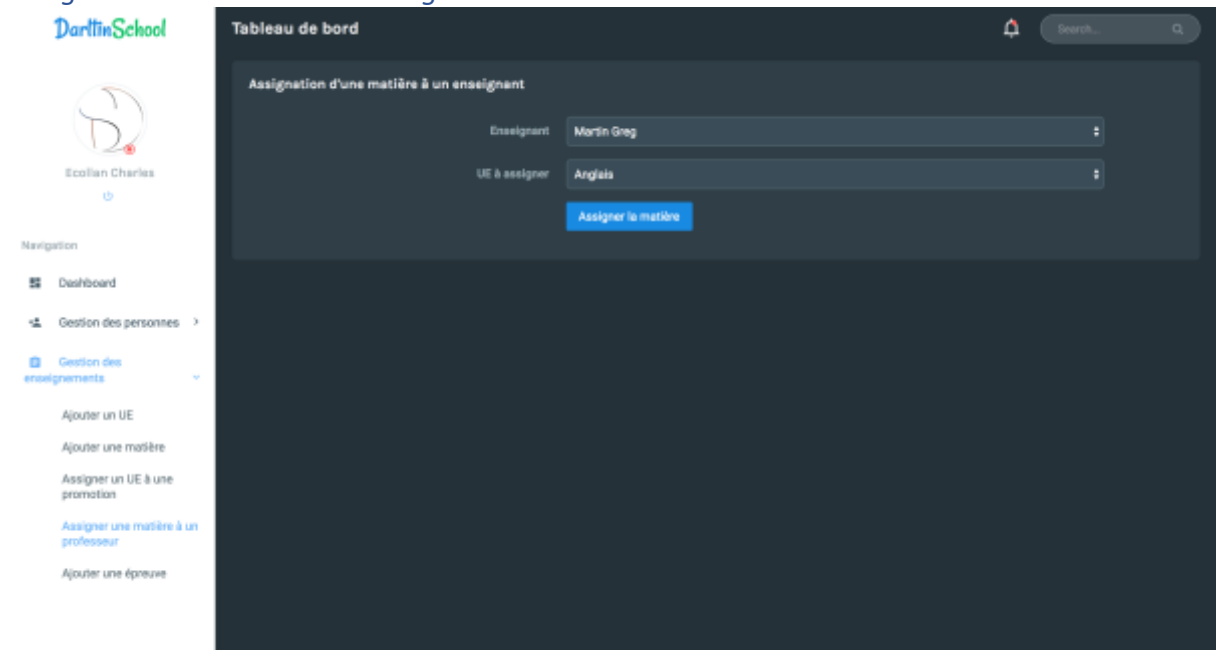


The screenshot shows the 'Tableau de bord' (Dashboard) of the DartlinSchool application. On the left is a navigation menu with the user profile 'Ecollan Charles' and options like 'Dashboard', 'Gestion des personnes', and 'Gestion des enseignements'. The main content area is titled 'Assigner une UE à une promotion'. It contains two dropdown menus: 'Promotion' with the value '2019' and 'UE à assigner' with the value 'Anglais'. Below these is a blue button labeled 'Assigner l'UE'.

Après avoir rentré une promotion et un UE, pour assigner un UE à une promotion, il faut remplir les informations suivantes :

- Nom de la promotion
- UE à assigner

Assigner une matière à un enseignant



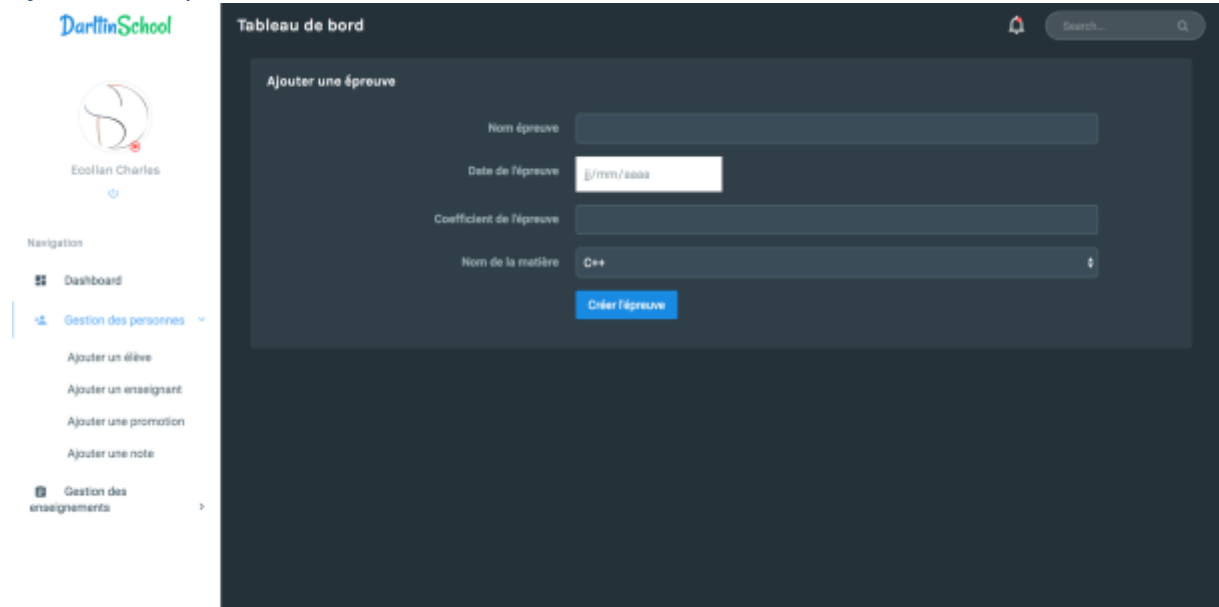
The screenshot shows the 'Tableau de bord' (Dashboard) of the DartlinSchool application. The navigation menu on the left is expanded, showing options like 'Ajouter un UE', 'Ajouter une matière', 'Assigner un UE à une promotion', 'Assigner une matière à un professeur', and 'Ajouter une épreuve'. The main content area is titled 'Assignment d'une matière à un enseignant'. It contains two dropdown menus: 'Enseignant' with the value 'Martin Greg' and 'UE à assigner' with the value 'Anglais'. Below these is a blue button labeled 'Assigner la matière'.

Après avoir rentré un enseignant et un UE, pour assigner une matière à un enseignant, il faut remplir les informations suivantes :

- Nom de l’enseignant

- UE à assigner

Ajouter une épreuve



The screenshot shows the 'Ajouter une épreuve' (Add exam) form within the DartfinSchool application. The form is titled 'Ajouter une épreuve' and is located on the 'Tableau de bord' (Dashboard) page. The form contains the following fields:

- Nom épreuve: A text input field.
- Date de l'épreuve: A date input field with a placeholder 'jj/mm/aaaa'.
- Coefficient de l'épreuve: A text input field.
- Nom de la matière: A dropdown menu with 'C++' selected.

Below the fields is a blue button labeled 'Créer l'épreuve'. The left sidebar shows the navigation menu with options like 'Dashboard', 'Gestion des personnes', 'Ajouter un élève', 'Ajouter un enseignant', 'Ajouter une promotion', 'Ajouter une note', and 'Gestion des enseignements'.

Pour ajouter une épreuve, il faut remplir les informations suivantes :

- Nom de l'épreuve (DE, TP...)
- Date
- Coefficient de l'épreuve
- Nom de la matière (préalablement créée)

Élèves

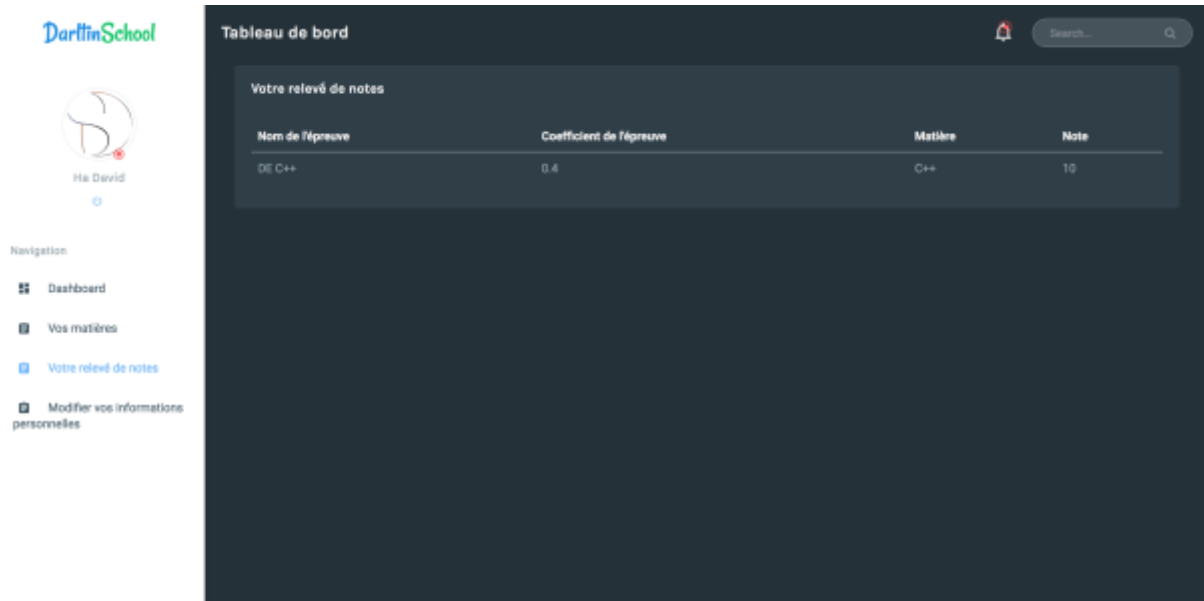
Vos matières

L'élève peut consulter toutes ses matières via cette vue.

Projet –Application de gestion d'un établissement scolaire

Bases de données

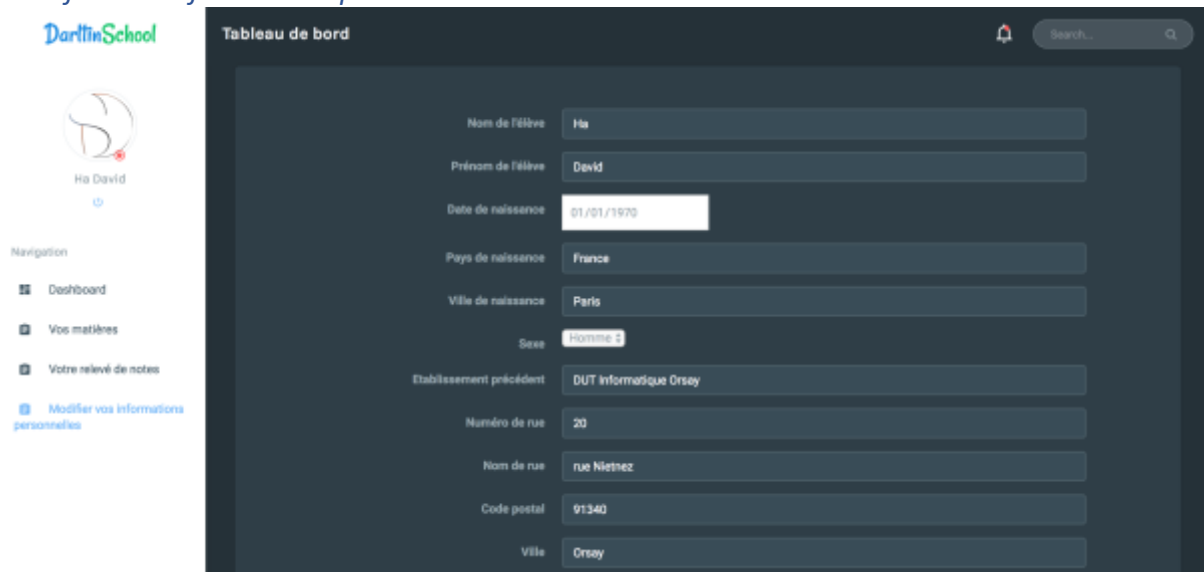
Votre relevé de notes



Nom de l'épreuve	Coefficient de l'épreuve	Matière	Note
DE C++	0.6	C++	10

L'élève peut consulter toutes ses notes des épreuves (DE, TP...).

Modifier vos informations personnelles



L'élève peut modifier ses informations personnelles.

Conclusion

L’application actuelle respecte bien celle imaginée dès le début dans le cahier des charges. Ce projet nous a permis de mettre en avant nos connaissances sur les bases de données ainsi que son intégration dans une application telle qu’un site internet.

Ce fut également une expérience très enrichissante dans le sens où on a appris à travailler en équipe en utilisant un système de versionning comme git.

Annexe 1 : Script SQL

```
#-----
#   Script MySQL.
#-----

#-----
# Table: eleve
#-----

CREATE TABLE eleve(
    matricule      int (11) Auto_increment NOT NULL ,
    nom            Varchar (255) NOT NULL ,
    prenom         Varchar (255) NOT NULL ,
    date_naissance Date NOT NULL ,
    ville_naissance Varchar (255) NOT NULL ,
    pays_naissance Varchar (255) NOT NULL ,
    sexe           Varchar (255) NOT NULL ,
    date_inscription Date NOT NULL ,
    etablissement_precedent Varchar (255) NOT NULL ,
    num_rue        Int NOT NULL ,
    nom_rue        Varchar (255) NOT NULL ,
    code_postal    Int NOT NULL ,
    ville          Varchar (255) NOT NULL ,
    tel_domicile   Int NOT NULL ,
    tel_mobile     Int NOT NULL ,
    email          Varchar (25) NOT NULL ,
    nom_medecin    Varchar (255) NOT NULL ,
    prenom_medecin Varchar (255) NOT NULL ,
    tel_medecin    Int NOT NULL ,
    vaccinations   Longtext NOT NULL ,
    allergies      Longtext NOT NULL ,
    remarques_medicales Longtext NOT NULL ,
    photo          Varchar (255) NOT NULL ,
    password       Varchar (255) ,
    nom_promo      Varchar (255) ,
    PRIMARY KEY (matricule )
)ENGINE=InnoDB;

#-----
# Table: annee
#-----
```

```
CREATE TABLE annee(  
    annee_debut Int NOT NULL ,  
    annee_fin Int NOT NULL ,  
    PRIMARY KEY (annee_debut )  
)ENGINE=InnoDB;
```

```
#-----  
# Table: contact  
#-----
```

```
CREATE TABLE contact(  
    nom_contact Varchar (255) NOT NULL ,  
    prenom_contact Varchar (255) NOT NULL ,  
    num_rue_contact Int NOT NULL ,  
    nom_rue_contact Varchar (255) NOT NULL ,  
    code_postal_contact Int NOT NULL ,  
    ville_contact Varchar (255) NOT NULL ,  
    tel_contact Int NOT NULL ,  
    email_contact Varchar (255) NOT NULL ,  
    id_contact int (11) Auto_increment NOT NULL ,  
    matricule Int NOT NULL ,  
    PRIMARY KEY (id_contact )  
)ENGINE=InnoDB;
```

```
#-----  
# Table: document  
#-----
```

```
CREATE TABLE document(  
    convoc_parent Longtext NOT NULL ,  
    id_document int (11) Auto_increment NOT NULL ,  
    matricule Int NOT NULL ,  
    PRIMARY KEY (id_document )  
)ENGINE=InnoDB;
```

```
#-----  
# Table: Promo  
#-----
```

```
CREATE TABLE Promo(  
    nom_promo Varchar (255) NOT NULL ,
```

```
PRIMARY KEY (nom_promo )  
)ENGINE=InnoDB;
```

```
#-----  
# Table: matiere  
#-----
```

```
CREATE TABLE matiere(  
    nom_matiere Varchar (255) NOT NULL ,  
    nom_ue Varchar (255) NOT NULL ,  
    PRIMARY KEY (nom_matiere )  
)ENGINE=InnoDB;
```

```
#-----  
# Table: ue  
#-----
```

```
CREATE TABLE ue(  
    nom_ue Varchar (255) NOT NULL ,  
    coeff_ue Double ,  
    PRIMARY KEY (nom_ue )  
)ENGINE=InnoDB;
```

```
#-----  
# Table: enseignant  
#-----
```

```
CREATE TABLE enseignant(  
    nom_enseignant Varchar (255) NOT NULL ,  
    prenom_enseignant Varchar (255) NOT NULL ,  
    id_enseignant int (11) Auto_increment NOT NULL ,  
    email_enseignant Varchar (255) ,  
    password Varchar (255) ,  
    PRIMARY KEY (id_enseignant )  
)ENGINE=InnoDB;
```

```
#-----  
# Table: epreuve  
#-----
```



```
CREATE TABLE epreuve(  
    nom_epreuves Varchar (255) NOT NULL ,  
    date_epreuve Date ,  
    coeff_epreuve Double ,  
    nom_matiere Varchar (255) NOT NULL ,  
    PRIMARY KEY (nom_epreuves )  
)ENGINE=InnoDB;  
  
#-----  
# Table: administrateur  
#-----  
  
CREATE TABLE administrateur(  
    nom_admin Varchar (255) ,  
    prenom_admin Varchar (255) ,  
    id_admin int (11) Auto_increment NOT NULL ,  
    email_admin Varchar (255) ,  
    password Varchar (255) ,  
    PRIMARY KEY (id_admin )  
)ENGINE=InnoDB;  
  
#-----  
# Table: present  
#-----  
  
CREATE TABLE present(  
    annee_debut Int NOT NULL ,  
    matricule Int NOT NULL ,  
    PRIMARY KEY (annee_debut ,matricule )  
)ENGINE=InnoDB;  
  
#-----  
# Table: etudie  
#-----  
  
CREATE TABLE etudie(  
    matricule Int NOT NULL ,  
    nom_ue Varchar (255) NOT NULL ,  
    PRIMARY KEY (matricule ,nom_ue )  
)ENGINE=InnoDB;
```

#-----

Table: passe

#-----

```
CREATE TABLE passe(  
    note      Double ,  
    matricule  Int NOT NULL ,  
    nom_epreuves Varchar (255) NOT NULL ,  
    PRIMARY KEY (matricule ,nom_epreuves )  
)ENGINE=InnoDB;
```

#-----

Table: enseigne

#-----

```
CREATE TABLE enseigne(  
    nom_matiere  Varchar (255) NOT NULL ,  
    id_enseignant Int NOT NULL ,  
    PRIMARY KEY (nom_matiere ,id_enseignant )  
)ENGINE=InnoDB;
```

#-----

Table: valide

#-----

```
CREATE TABLE valide(  
    nom_promo Varchar (255) NOT NULL ,  
    nom_ue    Varchar (255) NOT NULL ,  
    PRIMARY KEY (nom_promo ,nom_ue )  
)ENGINE=InnoDB;
```

#-----

Table: gere

#-----

```
CREATE TABLE gere(  
    id_admin  Int NOT NULL ,  
    nom_promo Varchar (255) NOT NULL ,  
    PRIMARY KEY (id_admin ,nom_promo )  
)ENGINE=InnoDB;
```

```
ALTER TABLE eleve ADD CONSTRAINT FK_eleve_nom_promo FOREIGN KEY (nom_promo)
REFERENCES Promo(nom_promo);
ALTER TABLE contact ADD CONSTRAINT FK_contact_matricule FOREIGN KEY (matricule)
REFERENCES eleve(matricule);
ALTER TABLE document ADD CONSTRAINT FK_document_matricule FOREIGN KEY (matricule)
REFERENCES eleve(matricule);
ALTER TABLE matiere ADD CONSTRAINT FK_matiere_nom_ue FOREIGN KEY (nom_ue)
REFERENCES ue(nom_ue);
ALTER TABLE epreuve ADD CONSTRAINT FK_epreuve_nom_matiere FOREIGN KEY
(nom_matiere) REFERENCES matiere(nom_matiere);
ALTER TABLE present ADD CONSTRAINT FK_present_annee_debut FOREIGN KEY
(annee_debut) REFERENCES annee(annee_debut);
ALTER TABLE present ADD CONSTRAINT FK_present_matricule FOREIGN KEY (matricule)
REFERENCES eleve(matricule);
ALTER TABLE etudie ADD CONSTRAINT FK_etudie_matricule FOREIGN KEY (matricule)
REFERENCES eleve(matricule);
ALTER TABLE etudie ADD CONSTRAINT FK_etudie_nom_ue FOREIGN KEY (nom_ue)
REFERENCES ue(nom_ue);
ALTER TABLE passe ADD CONSTRAINT FK_passe_matricule FOREIGN KEY (matricule)
REFERENCES eleve(matricule);
ALTER TABLE passe ADD CONSTRAINT FK_passe_nom_epreuves FOREIGN KEY
(nom_epreuves) REFERENCES epreuve(nom_epreuves);
ALTER TABLE enseigne ADD CONSTRAINT FK_enseigne_nom_matiere FOREIGN KEY
(nom_matiere) REFERENCES matiere(nom_matiere);
ALTER TABLE enseigne ADD CONSTRAINT FK_enseigne_id_enseignant FOREIGN KEY
(id_enseignant) REFERENCES enseignant(id_enseignant);
ALTER TABLE valide ADD CONSTRAINT FK_valide_nom_promo FOREIGN KEY (nom_promo)
REFERENCES Promo(nom_promo);
ALTER TABLE valide ADD CONSTRAINT FK_valide_nom_ue FOREIGN KEY (nom_ue)
REFERENCES ue(nom_ue);
ALTER TABLE gere ADD CONSTRAINT FK_gere_id_admin FOREIGN KEY (id_admin)
REFERENCES administrateur(id_admin);
ALTER TABLE gere ADD CONSTRAINT FK_gere_nom_promo FOREIGN KEY (nom_promo)
REFERENCES Promo(nom_promo);
```

Annexe 2 : Code Projet

Voir dossier code source