



Pro Forma

**LAZRAK NISSRINE- ABOUHAMOU AMEN - BELDILMI GUILLAUME-HAMMANI
ASSIA - DJERA FERIEL- KHENTICHE WISSAM
GROUPE: 1**

SOMMAIRE

I) Introduction

- En quoi consiste Pro Forma ?
- Comment est-il programmé ?

II) Conception

- Réalisation des chartes graphiques
- Conception BDD
- Diagramme USE CASE

III) Gestion de Projet

- Bases de notre projet
- Réunions et communication

IV) Développement du projet

- GitHub et Docker
- Front-End
- Back-End
- Lien entre le Front-End et le Back-End

V) Conclusion

- Présentation des pages web
- Contenu manquant
- Améliorations possibles

I) Introduction

En quoi consiste Pro Forma

Pro Forma ou Plateforme d'Education accessible en Ligne est le nom de notre projet de développement d'application Web. Ce projet de fin de 6^{ème} semestre de faculté d'informatique prend forme en groupe de 6 personnes. Le projet est open-source et complètement libre de droit.

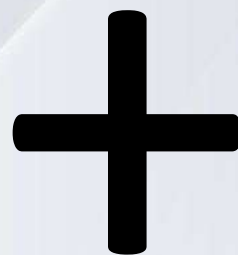
Comment est-il programmé ?

Pro Forma est codé de manière conventionnelle pour une application web :

- Pour la partie Front-End (partie visible par l'utilisateur) nous utilisons le trio **HTML**, **CSS** et **JS**.
- Pour la partie Back-End (partie invisible pour l'utilisateur) nous utilisons **PHP** en version 8.0.28
- Pour la base de données nous utilisons **MySQL** en version 8.0.32

Nous utilisons en plus de cela plusieurs autres technologies :

- **JQuery**, qui permet une utilisation plus facile de JavaScript dans quelques cas et...



II) Conception

Réalisation des chartes graphiques

Avant de faire la maquette finale, il était nécessaire de créer une charte graphique pour les deux thèmes. On a alors été élaborés des panels de couleurs avec leurs utilités retrouvables ici



Nous avons donc premièrement le thème clair composé de 6 couleurs. Sur ces 6 couleurs, les deux derniers forment les éléments les plus importants de la page. Le Platinum est le fond de la page et le noire le texte de la page. Nous avons ensuite deux couleurs le UT orange et l'orange carotte et correspondent aux boutons et à quelque div. Le couleur Timberwolf correspond à la couleur attribuée au bandeau, et l'orange claire correspond à la couleur d'accentuation des éléments en ayant besoin.

FFFFFF

White

737373

Dim gray

CC6600

Alloy orange

000000

Black

E6E6E6

Pastel

F89937

Sand orange

Puis nous avons le thème foncé, celui-ci est composé de 6 couleurs, nous retrouvons le fond de la page cette fois-ci dans un gris foncé et la couleur de texte en blanc. La différence se fait sur les couleurs d'accentuation. En effet, la couleur « Dim gray » est la couleur du bandeau et , la couleur « Alloy orange » des boutons.

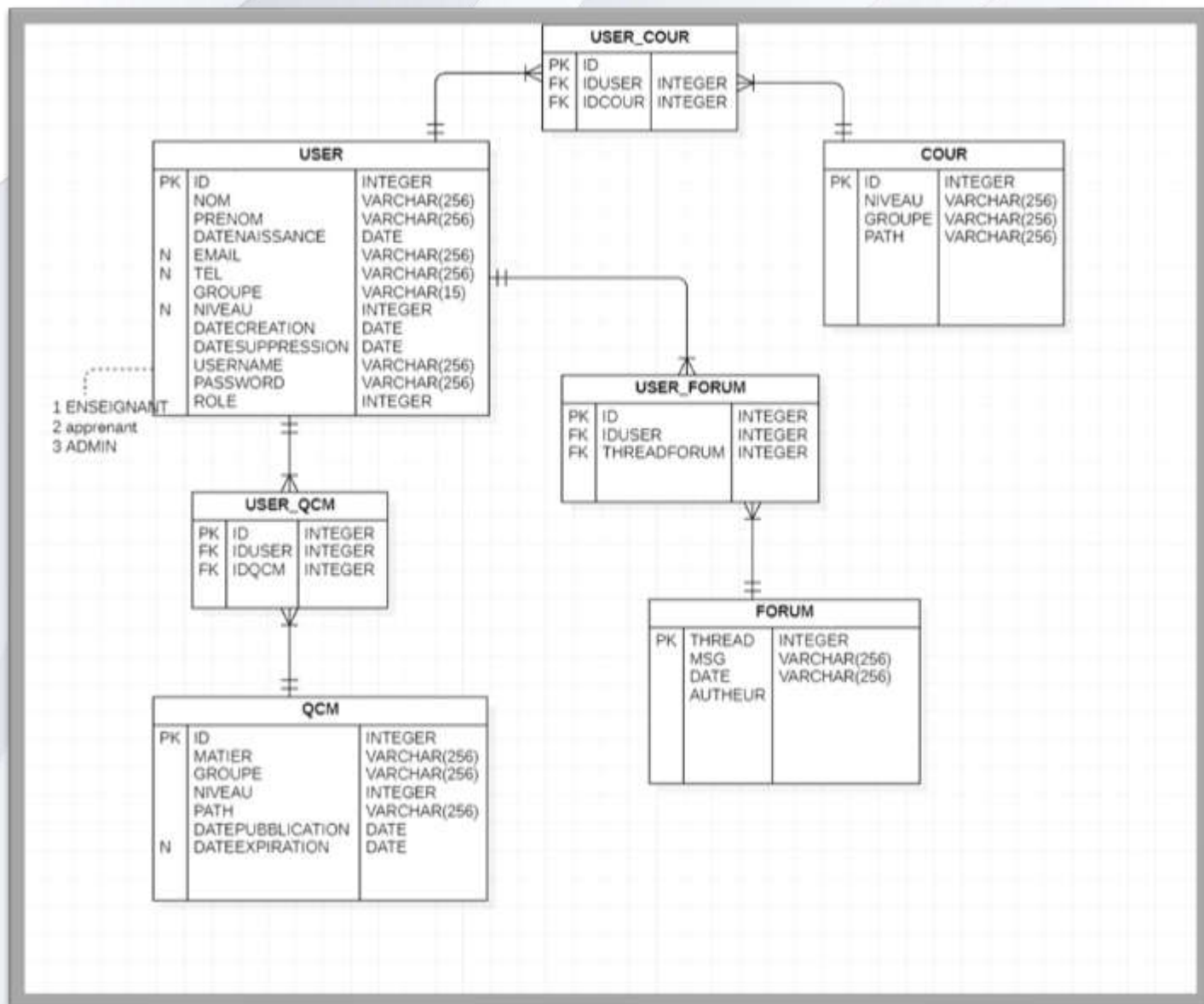
L'autre partie de cette charte graphique touche les polices. Nous avons choisi deux polices, je vais les laisser se présenter :

Bonjour, je suis [System-Ui](#) et je suis la police de tous les pages de Pro Forma, c'est-à-dire qui le site en base au OS il vas changer l'écriture du site.

Dans Windows on a comme font-family Segoe-UI ou dans Apple on a SF-PRO

Conception BDD

Après avoir prévu et conceptualisé l'intégralité de la partie visible de notre site, il est important de bien prévoir l'ensemble des données qui vont circuler et être stockée dans notre site web. Ainsi nous avons donc commencé par mettre en place un diagramme des différentes tables de la BDD. Voici la version originale :



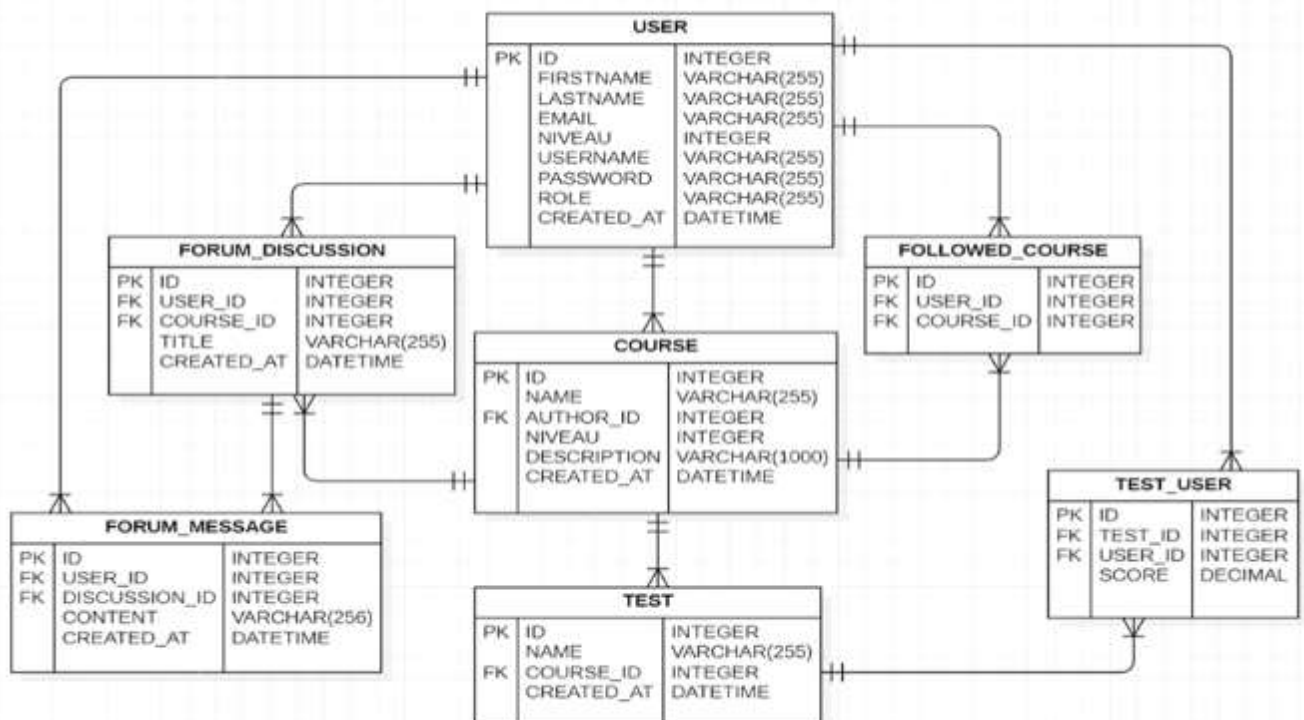
Puis à partir de celui-ci, nous avons créé le modèle relationnel suivant et avons réalisé le fichier SQL associé :

```

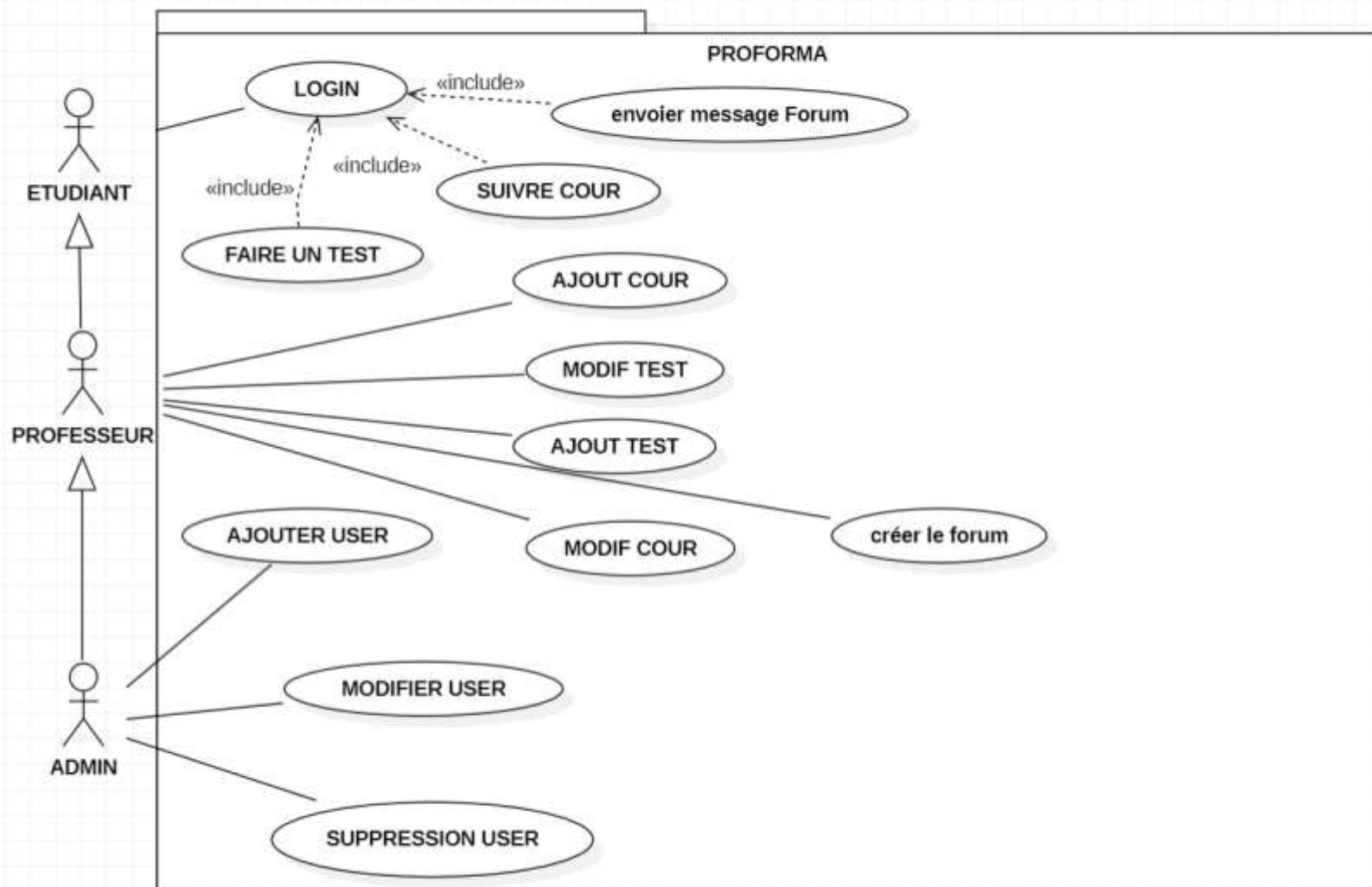
USER(ID,NOM,PRENOM,DATENAISSANCE,EMAIL,TEL,GROUPE,NIVEAU,DATECREATION,DATESUPP
RESSION,USERNAME,PASSWORD,ROLE)
COURSE (ID, NIVEAU,GROUPE,PATH)
USER_COUR (ID,#IDUSER,#IDCOUR)
USER_QCM (ID,#IDUSER,#IDQCM)
QCM (ID,MATIER,GROUPE,NIVEAU,PATH,DATEPUBLICATION,DATEEXPIRATION)
FORUM(THREAD,MSG,DATE)
USER_FORUM(ID,#IDUSER,#THREADFORUM)

```

En fin de projet, nous nous sommes rendu compte régulièrement que notre conception de la BDD n'était pas bonne, nous l'avons donc régulièrement changé car nous avons manqué de rigueur (ou de recul) lors de la création initiale. Cette conception a été réalisée sur StarUML.



USE CASE :



Avant la modélisation de la base de données d'abord on a fait une modélisation avec le diagramme des cases d'utilisation en identifiant toutes les fonctionnalités.

D'abord on a identifié les acteurs qui vont interagir avec le site, ensuite on a cherché les fonctionnalités du site.

Une fois que les fonctionnalités étaient déterminées on a attribué chaque fonction à l'acteur correspondant.

III) Gestion de projet

Base de notre projet

La gestion de projet est la base de celui-ci dans le cadre d'un projet en collaboration. Et la base de la gestion de projet c'est la collaboration entre les différents membres/ressources de ce projet. En effet PETAL c'est tout d'abord des chiffres :

- **6** étudiants en License d'informatique
- **15** semaines de projet
- Une moyenne de **8h** de travail par semaine
- Des pics de travail à plus de **40h** par semaine

Planification :

La première étape de la gestion de notre projet est la planification, la planification implique la définition claire de nos objectifs du projet, la spécification des délais, et l'établissement d'un plan d'action pour atteindre ces objectifs. On a également identifié les ressources nécessaires pour réaliser ce projet, y compris les compétences de l'équipe, les outils de développement, les technologies, les serveurs.

Répartition des tâches :

Une fois qu'on établit un plan clair pour notre projet, la prochaine étape consistait à diviser le travail en tâches et à les attribuer à chaque membre de l'équipe en fonction de leurs compétences. La répartition des tâches était équitable pour assurer un travail efficace et garantir que toutes les parties prenantes sont satisfaites.

Backend : Abouhamou Amen et Lazrak Nissrine

Frontend : Beldilmi Guillaume- Hammani Assia - Djera Ferial- Khentiche Wissam

Communication

La communication est essentielle pour la réussite de tout projet, et cela est particulièrement vrai pour les projets de développement web impliquant une équipe de 6 personnes. On a établi des canaux de communication clairs et réguliers pour garantir une collaboration efficace. On a aussi organisé des réunions régulières pour discuter des progrès, des défis et des solutions possibles.

Suivi :

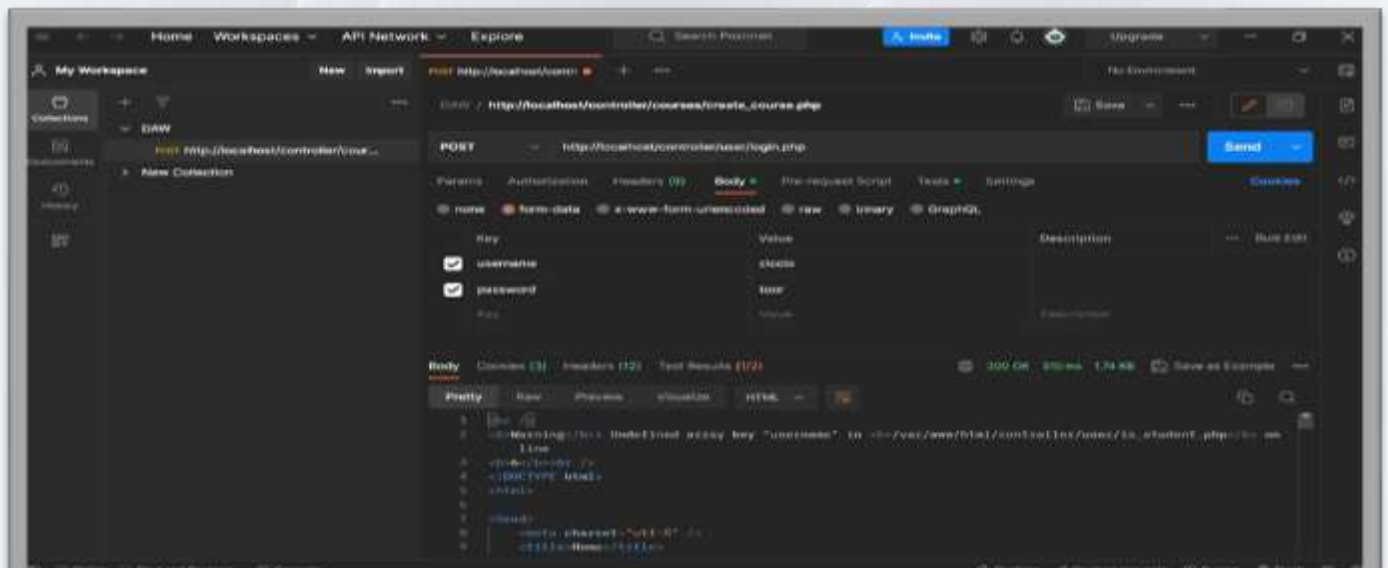
On a surveillé régulièrement l'avancement de notre projet pour s'assurer que les délais sont respectés et que les résultats sont conformes aux exigences initiales. Une fois que le travail a commencé, On a surveillé régulièrement l'avancement de notre projet pour s'assurer que les délais sont respectés et que les résultats sont conformes aux exigences initiales.

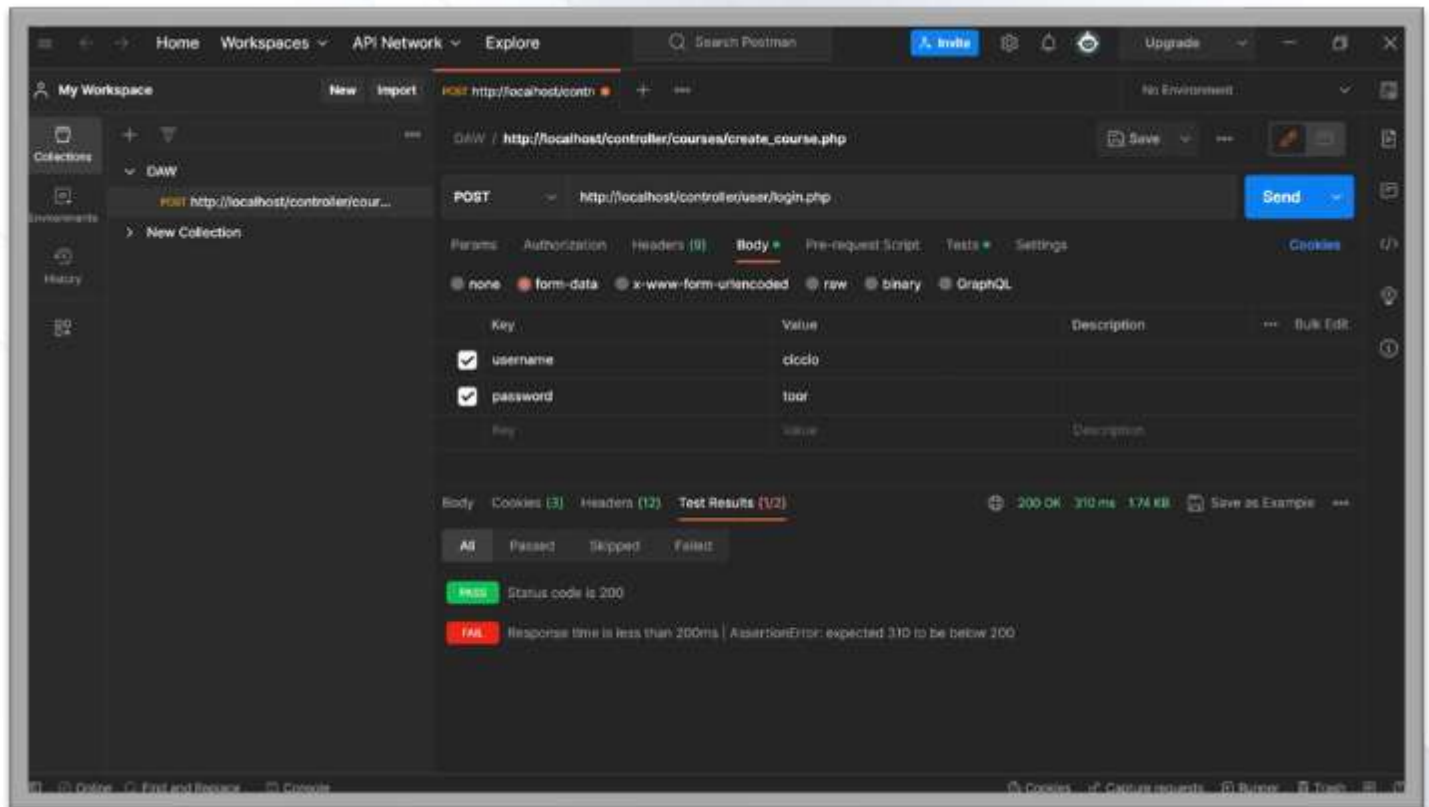
Tests et validation :

On a effectué des tests réguliers pour garantir la qualité du produit final. Implication des utilisateurs finaux pour obtenir leur feedback et apporter les modifications nécessaires. Les tests sont une étape importante du développement de sites web pour garantir la qualité du produit final. Les tests peuvent être effectués à différents stades du développement, du design aux fonctionnalités, et les résultats doivent être documentés et pour cela on a utilisé l'application POSTMAN



Postman est une application qui permet de tester les API (Interfaces de Programmation d'Applications) d'un site web. Pour cela on commence par créez un nouvel onglet, on choisit le type de requête HTTP qu'on souhaite envoyer (par exemple GET, POST, PUT, DELETE, etc.). Ensuite on doit entrez l'URL de notre site web dans la barre d'adresse de la requête et on joute un corps de requête dans la section "Body". Cliquer sur le bouton "Send" pour envoyer la requête et la réponse de l'API sera affichée dans la fenêtre "Response" de Postman.





C'est un exemple de comment elle se présente l'application POSTMAN, dans ce cas on a deux types de résultats :

En rouge ce qui signifie l'erreur Code HTTP 310 : La requête doit être redirigée de trop nombreuses fois, ou est victime d'une boucle de redirection.

En vert : statut code 200 veut dire la réussite de la requête HTTP

```

102 <?xml version="1.0"?>
103 <course100>
104   <name>dqqsdqsd</name>
105   <resource type="slide" path="/ciao/pptsqdf.ppt" />
106   <resource type="video" path="/mondo/ciao.mp4" />
107   <resource type="unkown" path="/hi/hello/coais.mp3" />
108   <prerequisite>pre1</prerequisite>
109   <prerequisite>pre2</prerequisite>
110   <prerequisite>pre3</prerequisite>
111 </course100>

```

Comme on peut voir grâce à l'application POSTMAN on peut tester si chaque fonction marche correctement et résoudre l'erreur avant de l'envoyer en production

Réunions et communications

La communication est la part la plus importante de la bien-réussite d'un projet collaboratif. Nous avons donc utilisé plusieurs outils facilitant le travail collectif.

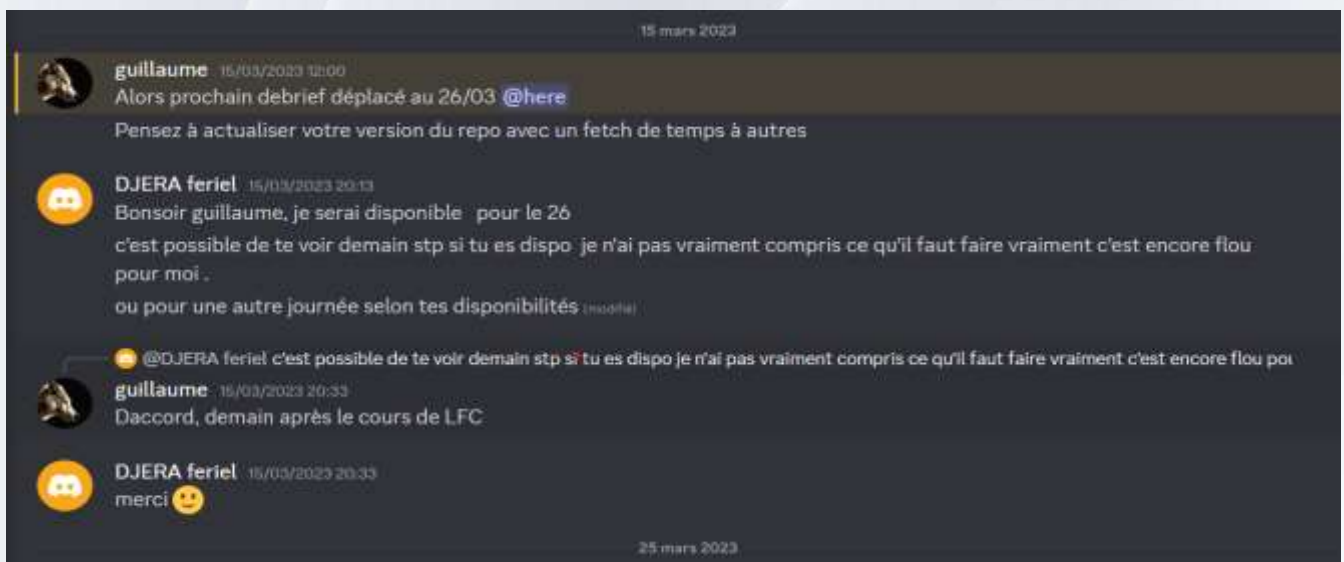


Le premier outil et le plus connu est GitHub. En effet celui-ci nous a permis de gérer le code et d'organiser le travail par équipe sans se soucier de fusionner les modifications de chacun.



Discord

La communication a également été assurée régulièrement via le serveur discord de notre projet. Chaque semaine une notification est envoyée à tout le monde pour voir l'avancement du travail.



La communication à distance est donc bien assurée et en plus de demander régulièrement à chaque membre du projet des nouveautés dans des canaux dédiés sur discord, des réunions régulières ont été mises en place pour assurer une communication de groupe constante et pour assurer la direction dans laquelle va le projet Pro Forma.



Voici donc la liste des réunions qui ont été effectuées avec leurs dates :

0^{ère} semaine → 01/02

1^{ème} semaine → 08/02

2^{ème} semaine → 15/02

4^{ème} semaine 21/02

5^{ème} semaine 08/03

6^{ème} semaine 15/03

7^{ème} semaine 26/03

9^{ème} semaine 30/03

11^{ème} semaine 11/04

14^{ème} semaine 24/04

15^{ème} semaine 2/05

IV) Développement du projet

GitHub et Docker

Avant même d'envisager le développement de PRO FORMA, il a fallu imaginer la gestion de notre base de données et surtout le type d'hébergement de notre site. En effet il n'était pas envisageable pour nous de travailler depuis un serveur distant. Nous avons donc choisi une solution assez spéciale.

Nous utilisons GitHub comme gestionnaire de code, nous voulons utiliser docker qui nous permet d'envoyer du code plus rapidement, de standardiser les opérations de notre site, de migrer aisément du code et de faire des économies en améliorant l'utilisation des ressources. Avec Docker, on obtient un objet unique qu'on peut exécuter n'importe où de manière fiable. Nous avons donc choisi de stocker le conteneur docker directement sur notre GitHub.



X



Arborescence du projet

Afin de respecter le modèle MVC, nous avons séparé le projet en trois dossiers principaux : `model`, `vue` et `controller`. Le premier contiendrait les fichiers relatifs aux données et à leur traitement (back-end), le second les fichiers relatifs à l'affichage (front-end) et le dernier les fichiers permettant de faire le lien entre les deux.

```
|-- controller
|   |-- color_theme
|   |-- courses
|   |-- forum
|   `-- user
|-- model
|   |-- data
|-- resources
|   |-- courses
|   |   `-- media
|   `-- tests
`-- vue
    |-- assets
    |   |-- css
    |   |-- img
    |   `-- js
    |-- index.php
    `-- pages
        |-- includes
        |   |-- account
        |   |-- course
        |   |   |-- student
        |   |   `-- teacher
        |   |-- forum
        |   |-- home
        |   `-- test
```

Concernant le front-end, sa mise en place s'est faite en plusieurs phases :

Tout d'abord, nous avons subdivisé chaque page en plusieurs éléments et séparés selon leur contexte. Nous avons créé un fichier `index.php` qui permet de charger dynamiquement le contenu de la page demandée située dans le dossier `pages` (si la page n'est pas trouvée, la page `404.php` sera chargée) ainsi que de charger le fichier `layout.php` qui contient la structure de base utilisée dans chaque page. À cela s'ajoute le dossier `includes` placé dans le dossier `pages` qui contient les fichiers `header.php` et `footer.php` qui sont respectivement chargés dans le `layout.php` ainsi qu'un dossier pour chaque page dont chaque fichier contient le contenu selon le contexte demandé (le fichier à charger est défini dans le fichier correspondant du dossier `page`). Ainsi l'organisation du contenu des pages est optimisée pour favoriser sa réutilisabilité et une gestion simplifiée des pages. Cette étape nous permet de diviser le contenu du site en plusieurs petits morceaux qui seront ensuite assemblés pour former la page finale.

Ensuite, nous avons commencé à remplir chacun de ces éléments. Leur petite taille et leur multiplicité nous a permis de diviser efficacement le travail entre les membres de l'équipe du front-end. Nous avons donc pu travailler en parallèle sur plusieurs éléments sans avoir à se soucier de la cohérence du site. À cette étape, nous avons pu remplir le contenu des pages ainsi que leur interaction avec l'utilisateur (boutons, formulaires, etc.).

Enfin, nous avons pu travailler sur l'aspect visuel du site. Nous avons donc pu ajouter des styles CSS aux éléments de chaque page afin de rendre le site plus agréable à l'œil.

L'équipe Front-End est composée par Beldilmi G., Djera F., Hammani A. et Khentiche W. ; pour l'organisation des tâches à faire on a fait comme suivant :

Beldilmi Guillaume s'est occupée de cette tâche :

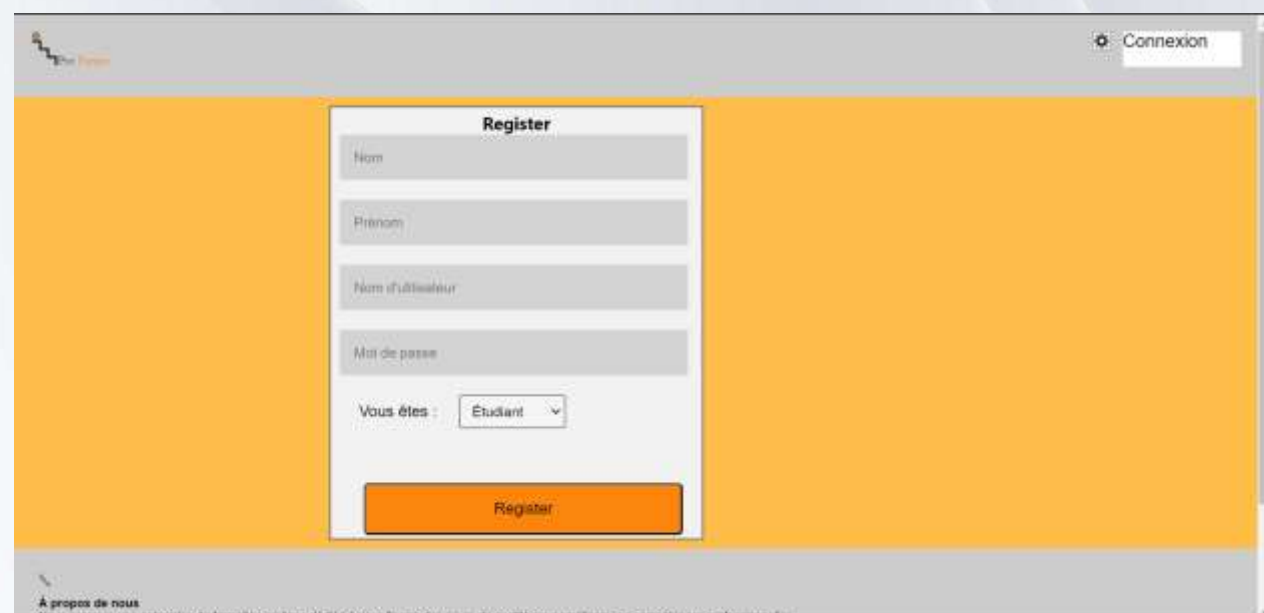
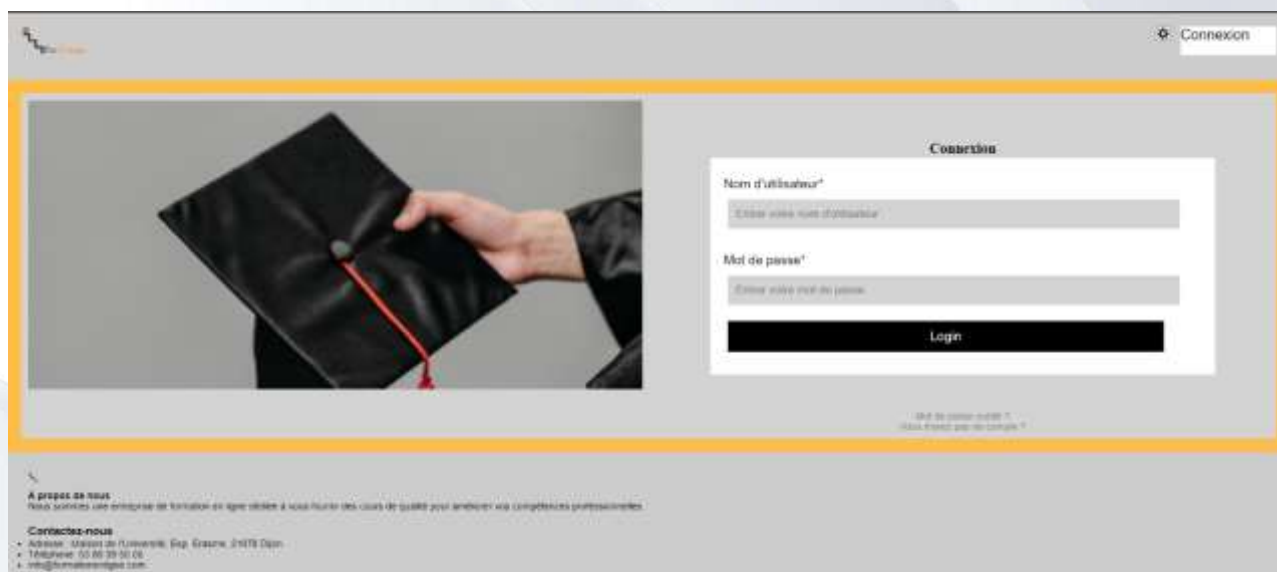
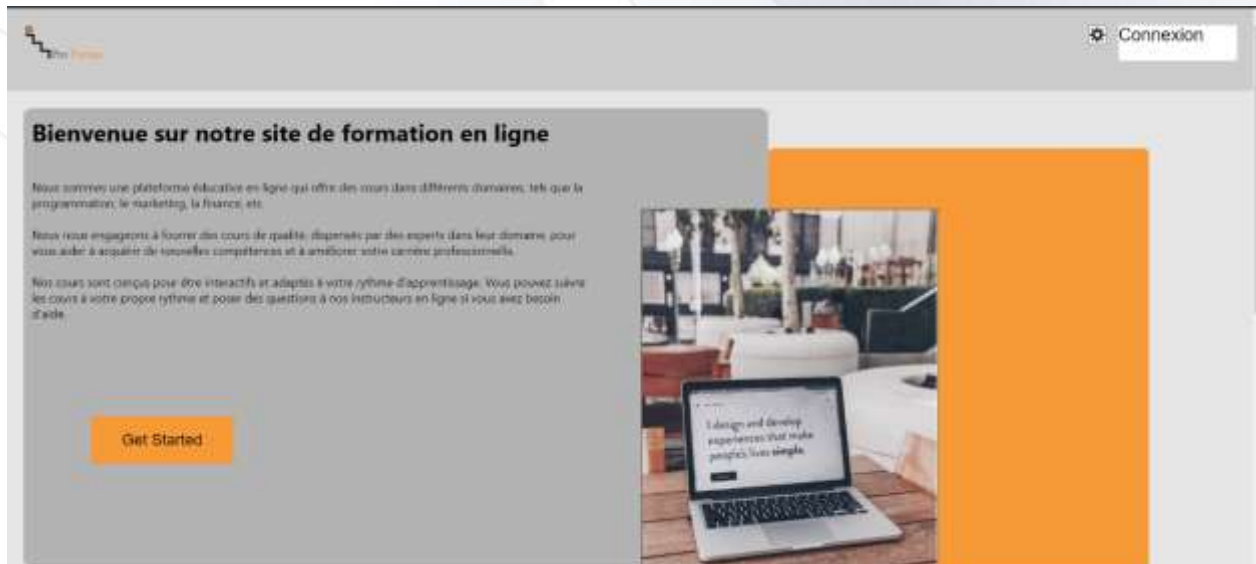
- Structure et supervision du front-end
- création des pages .php du site
- Création de multiples demandes de fonctions du contrôleur à l'équipe de back-end
- écriture de quelques scripts JS et quelque feuille de style CSS

Khentiche Wissam et Djera Feriel se sont occupées des suivantes tâches :

- Pages d'administration
- Pages sur les cours (+ gestion du contenu XML)
- Partiellement la page d'accueil

Hammani Assia était chargée des suivantes tâches :

- Pages de la partie forum (presque entièrement)
- Conception des DTD pour les fichiers XML (cours et tests)
- Design de la page d'accueil, login et registration
- Création des logos



Back-End

Pour l'organisation de la partie Back-end contient 4 dossiers :

- 1/ color-theme pour lire les cookies et voir le theme qu'il va utilisé dans le css
- 2/ courses : ce fichier contient la gestion des cours et des QCMs entre la base de données et le front-end ainsi que leur creations et la conversion en xml
- 3/ forum : gestion des conversations et les messages
- 4/ user : gestion de l'utilisateur : création suppression et modification, et son statut c'est à dire est ce qu'il est etudiant, professeur ou administrateur et les fonctions de login et logout pour se connecter au site et se deconnecter d'une maniere securisé

Grace à l'organisation du modèle MVC toutes les fonctions du gestion des données ne vont pas passés directement par la base des données mais ils vont faire des appels a des fonctions secondaires dans le dossier model et ses fonctions secondaire s'occupe du fait que le model va faire appel a la base de données pour se connecter a la base de données et appliquer les requetes

L'équipe Back-End est constituée par Abouhamou A. et Lazrak N. , pour l'organisation des taches on a fait comme suivant :

Abouhamou Amen dans un premier temps avec la collaboration de Beldilmi Guillaume a mis on place toute la structure de la base de données, et selon les requis du côté front-end et back-end on a adaptée la base. Abouhamou et Lazrak se sont occupés par la mise en place de l'environnement Docker avec l'orchestration des différents containers en utilisant un docker-compose file ainsi que de l'écriture des scripts du cotée Backend et Model avec la collaboration.

Ensuite ils ont écrit les scripts qui vont convertir les données reçues par un form de création, d'une QCM ou cours, en XML après les avoir testé avec l'application POSTMAN, grâce à la collaboration de Hammani Assia on a testé les résultats avec l'interaction à une page PHP et le résultat obtenue était le même de celui obtenue sur POSTMAN.

Ensuite dans la partie model ont créés 2 triggers dans la base de données, le premier interdit la création d'un nouvel utilisateur avec le droit d'administrateur car dans la base on a un seul utilisateur et l'autre trigger a la fonction de crypter le mot passe dans la base.

Lien entre le Front-End et le Back-End

Afin de lier le front-end et le back-end, nous avons créés de multiples fichiers PHP au sein du dossier `controller`. Chacun de ces fichiers contient une fonction éponyme permettant d'effectuer le traitement désiré.

De cette façon, lorsque l'équipe du front-end avait besoin d'une fonctionnalité spécifique, elle n'avait qu'à créer un nouveau fichier dans le dossier `controller` en décrivant le traitement à réaliser par le back-end. Le fichier serait complété par la suite par l'équipe en charge du back-end en ajoutant les appels au modèle pour la base de données.

Une fois que les scripts du contrôleur qui étaient demandés par le front sont prêts, le backend exécute plusieurs tests avec POSTMAN et un fois qu'ils passent tout le test avec succès, c'est alors que on les envoie en production pour tester la typologie de réponse qu'en reçoit avec l'interaction back et front.

La grande partie des fois ont reçu des réponses positives et on a le même résultat qu'on obtient en testant sur POSTMAN, mais il y a quelquefois que le résultat obtenu contient quelque imperfection, l'équipe le communique au backend et en travaillent ensemble vont résoudre le problème qui la grande partie des cas est une partie d'incompatibilité ou des noms de variables ou que la méthode utilisée a besoin d'une petite modification.

V) Conclusion

Maintenant que nous avons expliqué les grandes lignes de Pro Forma. Nous pouvons donc lancer celui-ci. Donc les différentes étapes permettant de mettre en place le site et de l'utiliser c'est :

- 1) Téléchargez le site avec Docker : <https://github.com/gbeldilmi/daw.git>
- 2) Récupérez l'un des deux scripts de remplissage de données :
<https://github.com/gbeldilmi/daw/blob/d5f66636e2f04082ef9d9eb236e3c59e5c8e7b20/scriptData.sh>











« Pro Forma données de table.sql » → Jeu de données complet possédant plusieurs utilisateurs et de nombreuses données

| Administrateur | Prénom | Nom | Username | MotDePasse |
|----------------|---------|---------|----------|------------|
| OUI | ciccio | ciccio | ciccio | toor |
| NON | charles | charles | CharleT | toor |
| NON | charles | Charles | charle | toor |
| NON | charles | charles | charles | toor |

« Pro Forma base » → Jeu de données uniquement composé d'un administrateur possédant les identifiants suivants :

- Nom : ciccio
- Prenom : ciccio
- Username : ciccio
- MotDePasse : toor

- 3) Lancez docker compose up dans le dossier et tapez « localhost/vue/ » dans un navigateur web.

| <input type="checkbox"/> | Name | Image | Status | Port(s) |
|--------------------------|--|--|---------------|---|
| <input type="checkbox"/> |  daw | | Running (3/3) | |
| <input type="checkbox"/> |  php-apache adc949d64c88  | daw-php-apache-environment | Running | 80:80  |
| <input type="checkbox"/> |  phpmyadmin-1 adb00d43770  | phpmyadmin/phpmyadmin | Running | 8080:80  |
| <input type="checkbox"/> |  db 4c0c8e521a81  | mysql | Running | 3308:3306  |

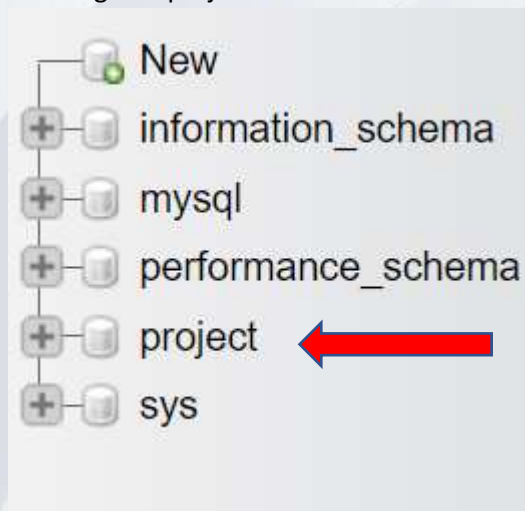
Voici les étapes à suivre si dans l'installation du projet l'importation des données grâce au script ne marche pas :

1) pour accéder à la base de données on utilise l'interface de phpMyAdmin en tapant « localhost :8080 » dans un navigateur web.



2) Connectez-vous à la base avec les identifiants par défaut : **root root**.

3) Cliquez dans la barre latérale sur l'onglet « projet ».



4) en suite clique sur SQL



5) Une fois terminé vous pouvez lancer l'app

Présentation des pages web

Cours créés :

php

dsfsdfsdf

[Voir le cours](#)

phplnit

dsfsdfsdf

[Voir le cours](#)

phplnit

dsfsdfsdf

[Voir le cours](#)

phplnit

dsfsdfsdf

[Voir le cours](#)

phplnit

dsfsdfsdf

[Voir le cours](#)

phplnit

dsfsdfsdf

[Voir le cours](#)

phplnit

dsfsdfsdf

[Voir le cours](#)

HTML

kjfhkefhzfcbkfjevclzjhick

[Voir le cours](#)

Toute modification apportée sera appliquée à l'ensemble de votre cours.

Titre Cours: Description: Niveau:

Ressource du cours

ressource N°1. /ciao/pptsqdf.ppt

ressource N°2. /mondo/ciao.mp4

ressource N°3. /hi/hello/coais.mp3

Prérequis

prérequis N°1. pre1

prérequis N°2. pre2

prérequis N°3. pre3

Associé ici une nouvelle ressource a votre cours

Ajouter le chemin vers la ressource: Aucun fichier choisi Type :

Associé ici un nouveau prérequis a ce cours

Ajouter :

Voici un exemple de la liste de cours et en cliquant sur un de cet cour il nous serait affiche le cours avec les prérequis, les ressources et la possibilité d'ajouter une nouvelle prérequis ou fichier

Difficulté rencontrée

Lors d'un projet de développement d'une application web impliquant 6 personnes, il est possible de rencontrer plusieurs difficultés, voici quelques difficultés que notre groupe a rencontrées beaucoup plus dans la partie Front-End :

1. Communication insuffisante : Avec autant de personnes impliquées dans le projet, il était difficile de maintenir une communication claire et régulière entre tous les membres de l'équipe. Cela peut entraîner des malentendus, des retards de tâches et un manque de coordination entre les différentes parties du projet.
2. Mauvaise répartition des tâches : Il est important de s'assurer que chaque membre de l'équipe a un rôle clair et défini dans le projet, avec des tâches qui leur conviennent. Et cela n'a pas été fait correctement, certaines tâches peuvent être négligées ou plusieurs personnes peuvent travailler sur la même tâche, ce qui entraîne un gaspillage de temps et de ressources.
3. Problèmes de qualité du code : On n'a pas tous les mêmes pratiques de développement, le code était quelquefois difficile à maintenir et à faire évoluer. Cela a entraîné des problèmes, des erreurs de fonctionnement et des difficultés à ajouter de nouvelles fonctionnalités à l'application.

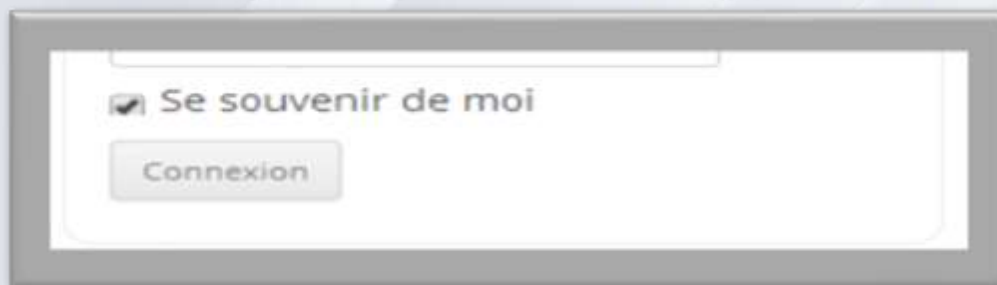
Contenu manquant

Nous avons terminé tous les éléments demandés pour notre site web et toutes les fonctionnalités attendues dans le sujet marché bien. Cependant, si nous avions eu plus de temps, nous aurions pu aller plus en profondeur et améliorer encore plus notre site.

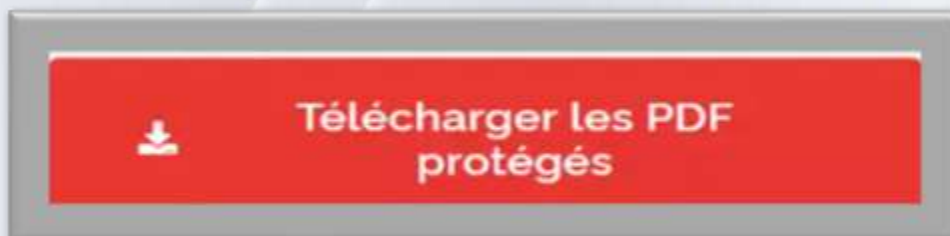
Améliorations possibles

Plusieurs sont les fonctionnalités supplémentaires qu'on peut ajouter à notre site web, en partie développées ou images et conceptualisées qui n'ont pas pu être mises en place comme :

- . Un lien « mot de passe oublié » a été envisagé sur la conception et sur les maquettes. Or un problème s'est posé à nous. Nous voulions envoyer un mail à l'utilisateur avec le mot de passe dedans mais on avait du mal à implémenter
- . Une case « se souvenir de moi » comme ça la prochaine fois qu'un utilisateur veut se connecter il n'aura pas besoin de rentrer ses informations



- . Ajouter une photo de profil à chaque utilisateur que ça soit étudiant, professeur ou administrateur et la stocker dans la base de données
- . La possibilité de télécharger un cours sous forme PDF ou Word



- . Un lien dans chaque cours qui permet à n'importe quel étudiant de poser des questions concernant ce cours au professeur

Le rapport a été fait par : LAZRAK Nissrine et ABOUHAMOU Amen