

QuizMaster

Projet TPI du 02.05.2024 au 03.06.2024



Chef de projet : Helder Costa Lopes

Experts : Benjamin Wolf
Roger Malherbe

ETML – Alexandre Fernandes – CIN4B

Table des matières

1	Glossaire	4
2	Analyse préliminaire	7
2.1	Introduction	7
2.1.1	Titre	7
2.1.2	Description	7
2.1.3	Contexte	7
2.2	Objectifs	7
2.3	Planification initiale	8
3	Analyse / Conception	9
3.1	Concept	9
3.1.1	Méthodologie de gestion de projet : en cascade	9
3.1.2	Maquettes graphiques	10
3.1.3	Modèle Conceptuel des Données :	14
3.1.4	Modèle Logique des Données	15
3.2	Stratégie de test	16
3.3	Risques techniques	16
3.4	Planification	17
3.5	Dossier de conception	18
3.5.1	Matériel Hardware	18
3.5.2	Application / Logiciels / Outils	18
3.5.3	Site web	19
3.5.4	Suivi de projet	20
3.5.5	Base de données : MariaDB	20
3.5.6	Architecture du programme	23
3.5.7	Dépendance externe	24
3.5.8	Utilisation schématisée	24
4	Réalisation	25
4.1	Liste des fichiers : Répertoire	25
4.2	Liste des fichiers : Code source	26
4.3	Base de données : Comment fonctionnent les requêtes	27
4.4	Base de données : Authentification	28
4.4.1	Requêtes préparées utilisées	28
4.4.2	Connexion	28
4.4.3	Inscription	30
4.4.4	Déconnexion	32
4.5	Base de données : Affichage tous les quiz	33
4.5.1	Requêtes préparées utilisées	33
4.5.2	Affichage des quiz	33
4.6	Base de données : Affichage quiz spécifique	34
4.6.1	Requêtes préparées utilisées	34
4.6.2	Affichage du quiz	34
4.7	Base de données : Traitement du quiz	35
4.7.1	Requêtes préparées utilisées	35
4.7.2	Traitement	35
4.7.3	Affichage des réponses	36

4.8	Base de données : Création quiz.....	37
4.8.1	Requêtes préparées utilisées	37
4.8.2	Traitement et création du quiz	38
4.9	Base de données : Suppression quiz	40
4.9.1	Requêtes préparées utilisées	40
4.10	Base de données : Affichage classement.....	42
4.10.1	Requêtes préparées utilisées	42
4.10.2	Affichage du classement.....	42
4.11	Base de données : Gestion des utilisateurs.....	44
4.11.1	Requêtes préparées utilisées	44
4.11.2	Affichages de tous les utilisateurs.....	45
4.11.3	Modification.....	45
4.11.4	Suppression.....	46
4.12	Cookies.....	48
4.13	Base de données : résultat visuel (PHPMyAdmin)	48
4.14	Versions du projet.....	50
4.15	Description des tests effectués.....	51
4.16	Test unitaire.....	55
4.17	Erreurs restantes	56
4.18	Liste des documents fournis	56
5	Conclusions.....	57
5.1	Bilan des fonctionnalités demandées	57
5.2	Bilan personnel.....	58
5.3	Bilan de la planification	58
6	Annexes.....	59
6.1	Résumé du projet QuizMaster	59
6.2	Webographie	59
6.3	Table des illustrations	61
6.4	Dossier des tests	62
6.5	Archives du projet.....	62
6.6	Planification – Journal de travail	63
6.7	Journal de travail	70

1 Glossaire

A

Authentification

Processus de vérification de l'identité d'un utilisateur, souvent en comparant un mot de passe saisi avec un mot de passe stocké dans une base de données.

B

Bootstrap

Collection d'outils pour créer des sites et applications web, comprenant des codes HTML, CSS et des extensions JavaScript.

C

Connexion PDO

Utilisation de PHP Data Objects pour se connecter à une base de données, permettant une interface uniforme pour accéder à plusieurs types de bases de données.

CSS (Cascading Style Sheets)

Langage utilisé pour décrire la présentation des documents HTML ou XML.

Commit

Enregistrement d'un ensemble de modifications dans un système de contrôle de version comme Git, permettant de suivre et de gérer l'évolution du code source.

F

Flexbox

Module de mise en page CSS permettant un agencement plus flexible des éléments sur une page web.

G

GitHub

Plateforme d'hébergement de code source utilisant le système de contrôle de version Git.

J

jQuery

Bibliothèque JavaScript qui simplifie la manipulation des documents HTML, la gestion des événements, l'animation et les interactions Ajax.

M

MCD (Modèle Conceptuel des Données)

Représentation graphique de la structure d'une base de données, décrivant les entités et les relations entre elles.

MLD (Modèle Logique des Données)

Transformation du MCD en une structure logique adaptée à la mise en œuvre dans un SGBD (Système de Gestion de Base de Données).

MySQL

Système de gestion de base de données relationnelle open source.

P

PHP (Hypertext Preprocessor)

Langage de script utilisé principalement pour le développement web et la création de pages web dynamiques.

PHPMyAdmin

Application web pour gérer les bases de données MySQL et MariaDB.

R

Requêtes préparées

Requêtes SQL où les paramètres sont fournis séparément de la commande SQL, augmentant ainsi la sécurité en protégeant contre les injections SQL.

Responsive Design

Approche de la conception web qui permet à un site de s'adapter à différentes tailles d'écran et appareils.

S

SHA256

Algorithme de hachage cryptographique produisant un condensé de 256 bits, souvent utilisé pour sécuriser les mots de passe.

Suite Microsoft 365

Ensemble d'applications et de services de productivité basés sur le cloud, incluant Word, Excel, PowerPoint et d'autres outils.

T

TPI (Travail Pratique Individuel)

Projet individuel réalisé par un étudiant pour valider ses compétences et obtenir son diplôme.

V

Visual Studio Code

Editeur de code source développé par Microsoft, offrant des fonctionnalités comme le débogage, la complétion de code et le contrôle de version intégré.

2 Analyse préliminaire

2.1 Introduction

2.1.1 Titre

QuizMaster

2.1.2 Description

Ce projet a pour but de créer un système de quiz interactif en PHP. Les utilisateurs pourront faire différents quiz, répondre aux questions et voir leurs résultats tout de suite. L'application aura une interface utilisateur simple, une base de données MySQL pour garder les données des quiz et une authentification des utilisateurs pour distinguer les administrateurs des utilisateurs ordinaires.

Les fonctionnalités principales seront la création de quiz, le défilement des questions, le calcul du score et l'affichage du score.

On pourrait comparer cela à un « [Kahoot](https://kahoot.com/)¹ » pour l'ETML.

2.1.3 Contexte

Ce projet a été réalisé durant mon TPI. Celui-ci permettra de valider mon CFC. Le thème développement web et le chef de projet ont été choisis par moi-même puis le choix fut confirmé par le chef de projet Helder Manuel Costa Lopes.

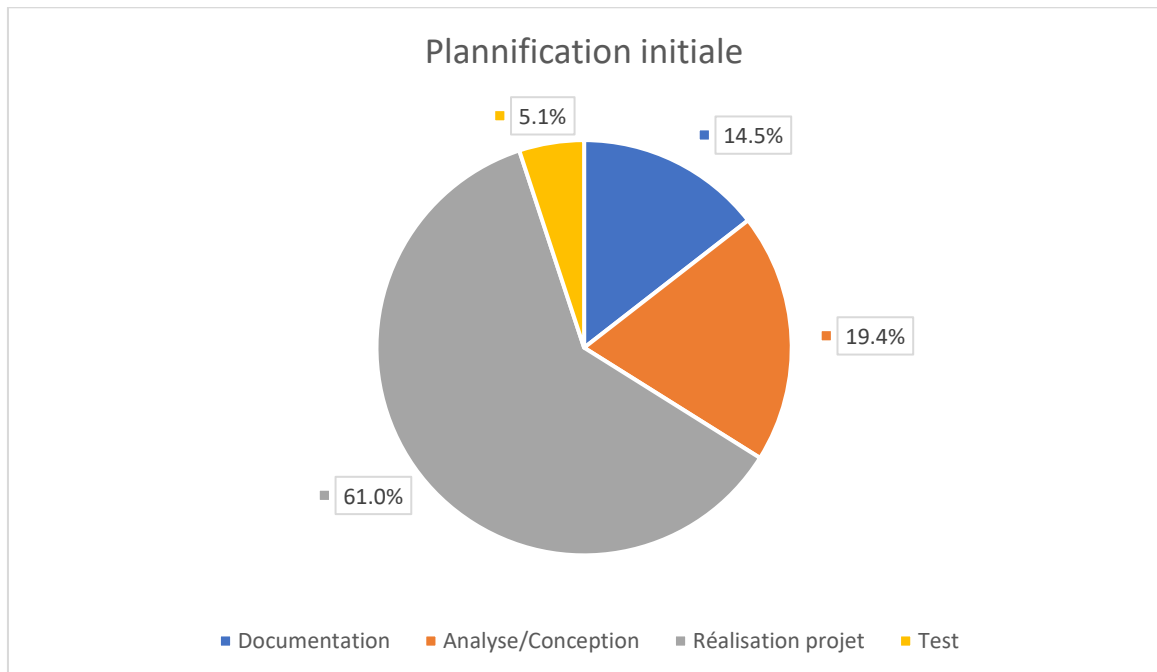
2.2 Objectifs

Ce projet a pour objectif de :

- La présence des différentes fonctionnalités tel que :
 - Gestion de la base de données MySQL
 - Création base de données
 - Création tables utilisateurs, questions, réponses, quiz
 - Insertion, récupération et mise à jour des données
 - Création de toutes les interfaces utilisateurs nécessaires
 - Intégration PHP-MYSQL
 - Connexion base de données
 - Exécution requêtes
 - Récupération et affichage des données
 - Authentification des utilisateurs (connexion, inscription, déconnexion)
 - Calcul du score des quizz et mise à jour du score
 - Affichage des scores à la fin du quiz avec les réponses correctes et fausses
 - Implémenter un classement des utilisateurs
 - Sécurité des données (hachage pour données sensibles)
 - Procédure de test
 - Design responsive
 - Gestion des erreurs
- Un schéma MCD/MLD et une sauvegarde régulière

¹ <https://kahoot.com/>

2.3 Planification initiale



Cette planification initiale permet de visionner de manière très large comment je pense devoir répartir mon temps pour la réalisation de ce projet.

Documentation :

- Ecriture du rapport et journal de travail

Analyse/Conception :

- Ecriture de la planification, réalisation de schéma

Réalisation projet :

- Mise en place des logiciels, réalisation du programme (base de données, backend, frontend)

Test :

- Réalisation de test du code

Annexe : [Planification](#)

3 Analyse / Conception

3.1 Concept

3.1.1 Méthodologie de gestion de projet : en cascade

Pour la réalisation de ce projet, la méthode de travail est la suivante : en cascade.

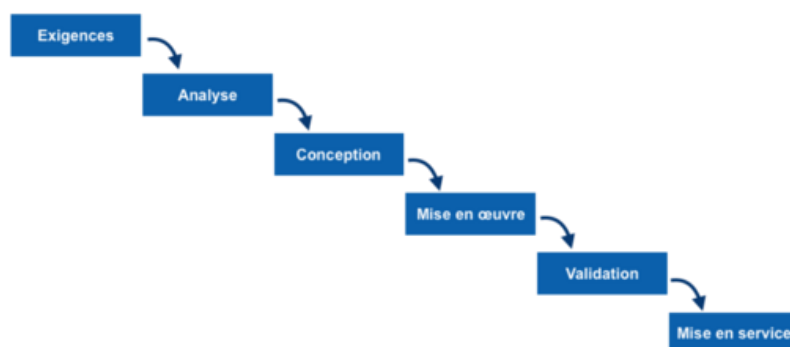


Figure 1 Modèle : en cascade

Exigences : Les exigences correspondent à la rencontre entre l'équipe de projet et le client pour comprendre les besoins et attentes concernant le produit final. Les informations permettent de définir le périmètre du projet et les bases du développement.

Dans notre cas, une rencontre avec le chef de projet et le premier expert a été effectuée.

Analyse : L'analyse permet de reprendre les informations obtenues lors de la phase d'exigences afin de les analyser et être sûr qu'elles soient claires, complètes et cohérentes. Des décisions peuvent déjà être prise sur cette étape concernant par exemple une planification initiale ou le choix des ressources nécessaire ou même des échéances concernant le projet.

Dans notre cas, cela a été effectué juste avant la signature du cahier des charges durant l'entrevue avec le premier expert.

Conception : La conception permet de réaliser une conception détaillée du projet, comme la conception d'une interface utilisateur, spécification des fonctionnalités, une planification détaillée.

Dans notre cas, cela a été effectué lors de la semaine suivant le début du projet, lors des deux premier jours une planification détaillée, suivant ces jours, des maquettes web et de base de données ont été effectuées.

Mise en œuvre : Lors de la mise en œuvre, l'équipe commence le code du projet en se basant sur les conceptions précédentes. Les fonctionnalités sont développées en utilisant dans la majorité des cas, des conventions.

Dans notre cas cela est effectué en utilisant les conventions ETML tout au fil du temps impartit pour le TPI.

Validation : La validation permet, une fois le développement terminé, de vérifier que le logiciel corresponde aux exigences du client. Cela est effectué à l'aide de test unitaires par exemple.

Dans notre cas, des tests sont effectués selon la stratégie de test cité plus bas, et la présentation de l'avancé au chef de projet

Mise en service : La mise en service correspond au déploiement dans son environnement. Une fois le développement aboutit, les tests validés.

Dans notre cas, cela correspond à la livraison du projet à la fin du TPI

3.1.2 Maquettes graphiques

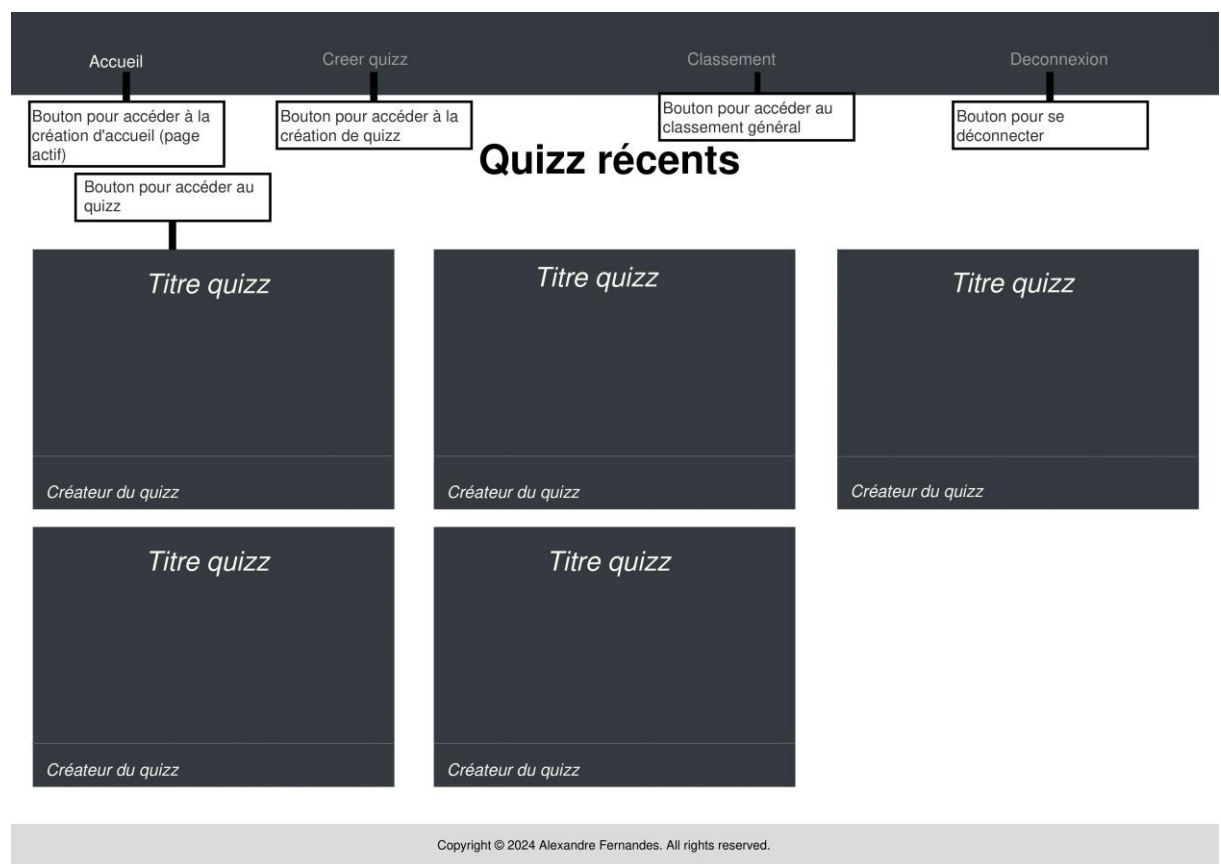
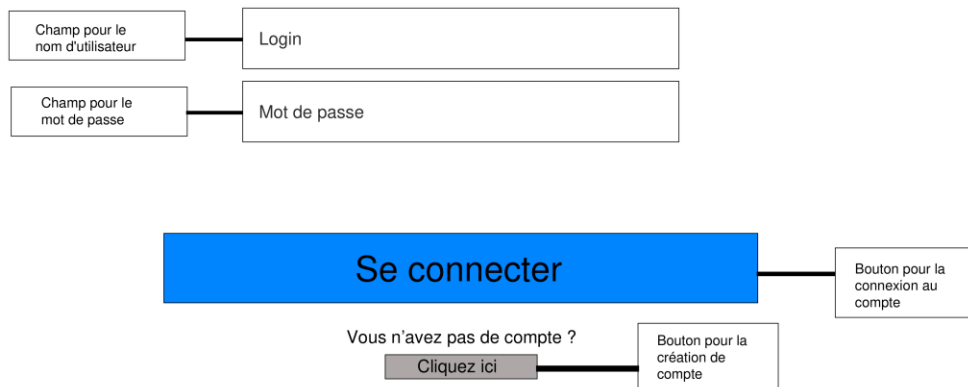


Figure 2 Maquette : Page d'accueil

Connexion



Maquette de la page de connexion. Elle comprend deux champs de saisie pour le nom d'utilisateur et le mot de passe, chacun avec une étiquette descriptive. En dessous se trouve un bouton bleu 'Se connecter' et un lien 'Vous n'avez pas de compte ? Cliquez ici' qui pointe vers un bouton 'Bouton pour la création de compte'.

Champ pour le nom d'utilisateur

Login

Champ pour le mot de passe

Mot de passe

Se connecter

Bouton pour la connexion au compte

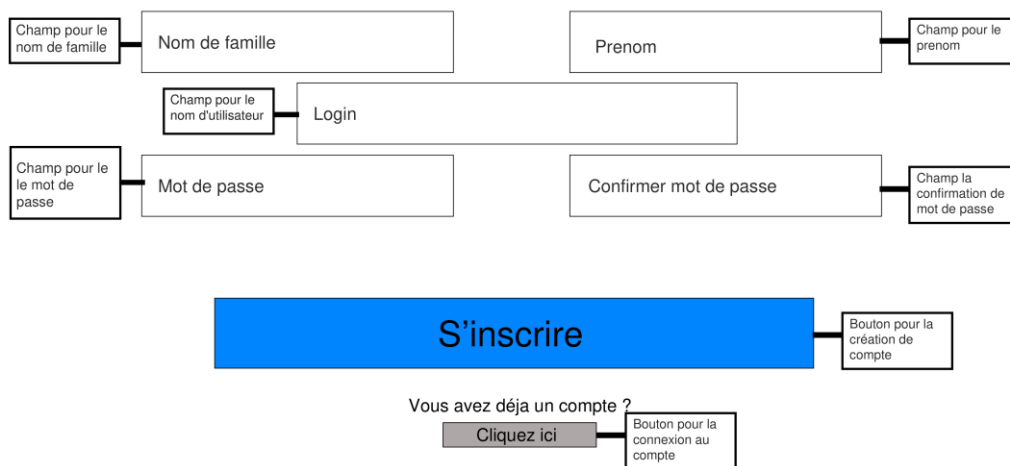
Vous n'avez pas de compte ? Cliquez ici

Bouton pour la création de compte

Copyright © 2024 Alexandre Fernandes. All rights reserved.

Figure 3 Maquette : Page de connexion

INSCRIPTION



Maquette de la page d'inscription. Elle contient des champs pour le nom de famille, le prénom, le login, le mot de passe et la confirmation du mot de passe, chacun avec une étiquette descriptive. En bas se trouve un bouton bleu 'S'inscrire' et un lien 'Vous avez déjà un compte ? Cliquez ici' qui pointe vers un bouton 'Bouton pour la connexion au compte'.

Champ pour le nom de famille

Nom de famille

Prénom

Champ pour le prénom

Champ pour le nom d'utilisateur

Login

Champ pour le mot de passe

Mot de passe

Confirmer mot de passe

Champ la confirmation de mot de passe

S'inscrire

Bouton pour la création de compte

Vous avez déjà un compte ? Cliquez ici

Bouton pour la connexion au compte

Copyright © 2024 Alexandre Fernandes. All rights reserved.

Figure 4 Maquette : Page d'inscription

Accueil	Creer quizz	Classement	Deconnexion
---------	-------------	------------	-------------

Menu navigation

Classement

1er Alexandre Fernandes	18 points
2. Kinan Jano	17 points
Fabrice Gelsomino	15 points
4. Yann Imperadori	12 points
5. Kevin Avdylaj	8 points
6. Antoine Roger	4 points
7. Yannick Pena Bustabad	1 points
Votre score : 18 points	Votre position : 1er

Copyright © 2024 Alexandre Fernandes. All rights reserved.

Figure 5 Maquette : Page de Classement

Accueil	Creer quizz	Classement	Deconnexion
---------	-------------	------------	-------------

Menu navigation

Creation quizz

Champ pour inscrire une question

1. Inscrivez une question

Champ pour inscrire une reponse

Inscrivez une reponse

Inscrivez une question

Inscrivez une reponse

Inscrivez une question

Inscrivez une reponse

Inscrivez une question

Inscrivez une reponse

...

Figure 6 Maquette : Page de Création Quizz

[Accueil](#) [Creer quizz](#) [Classement](#) [Deconnexion](#)

Menu de navigation

[Titre du Quizz]

1. Qui est le roi d'Espagne en 2024 ?

Felipe VI

Champ pour inscrire une réponse

2. Quel équipe à gagner l'Euro en 2016

France

3. Quel est la capitale de la Suisse ?

Inscrivez une réponse...

Copyright © 2024 Alexandre Fernandes. All rights reserved.

Figure 7 Maquette : Page de Quizz

2. Quel équipe à gagner l'Euro en 2016

France

0/1

3. Quel est la capitale de la Suisse ?

Berne

1/1

Total 2/3

Votre nouveau score de classement général est de 20 points (+2 points)

[Retour à la page d'accueil](#)

Copyright © 2024 Alexandre Fernandes. All rights reserved.

Figure 8 Maquette : Page Réponse de Quizz

3.1.3 Modèle Conceptuel des Données :

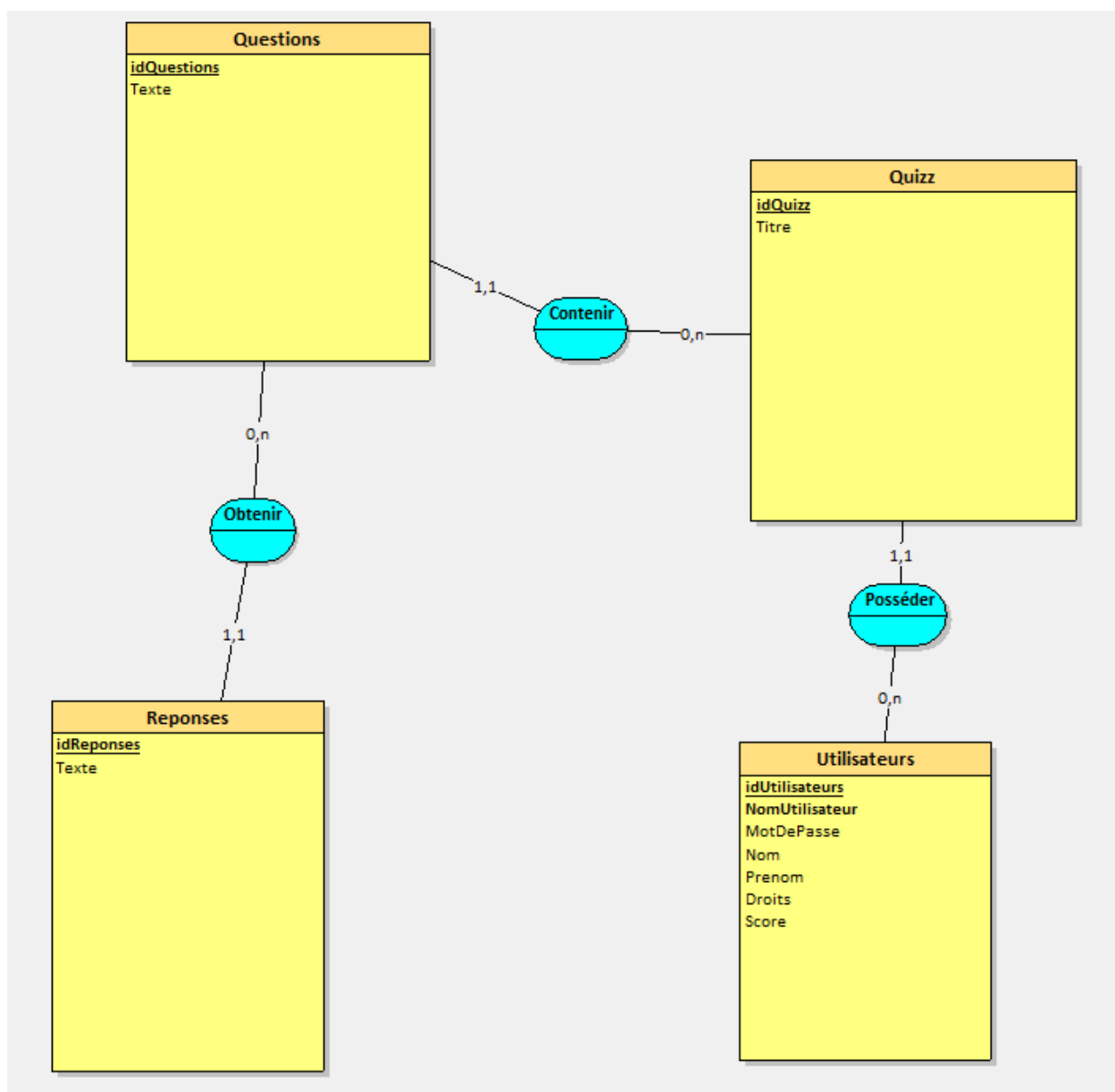


Figure 9 Modèle conceptuel des données

3.1.4 Modèle Logique des Données

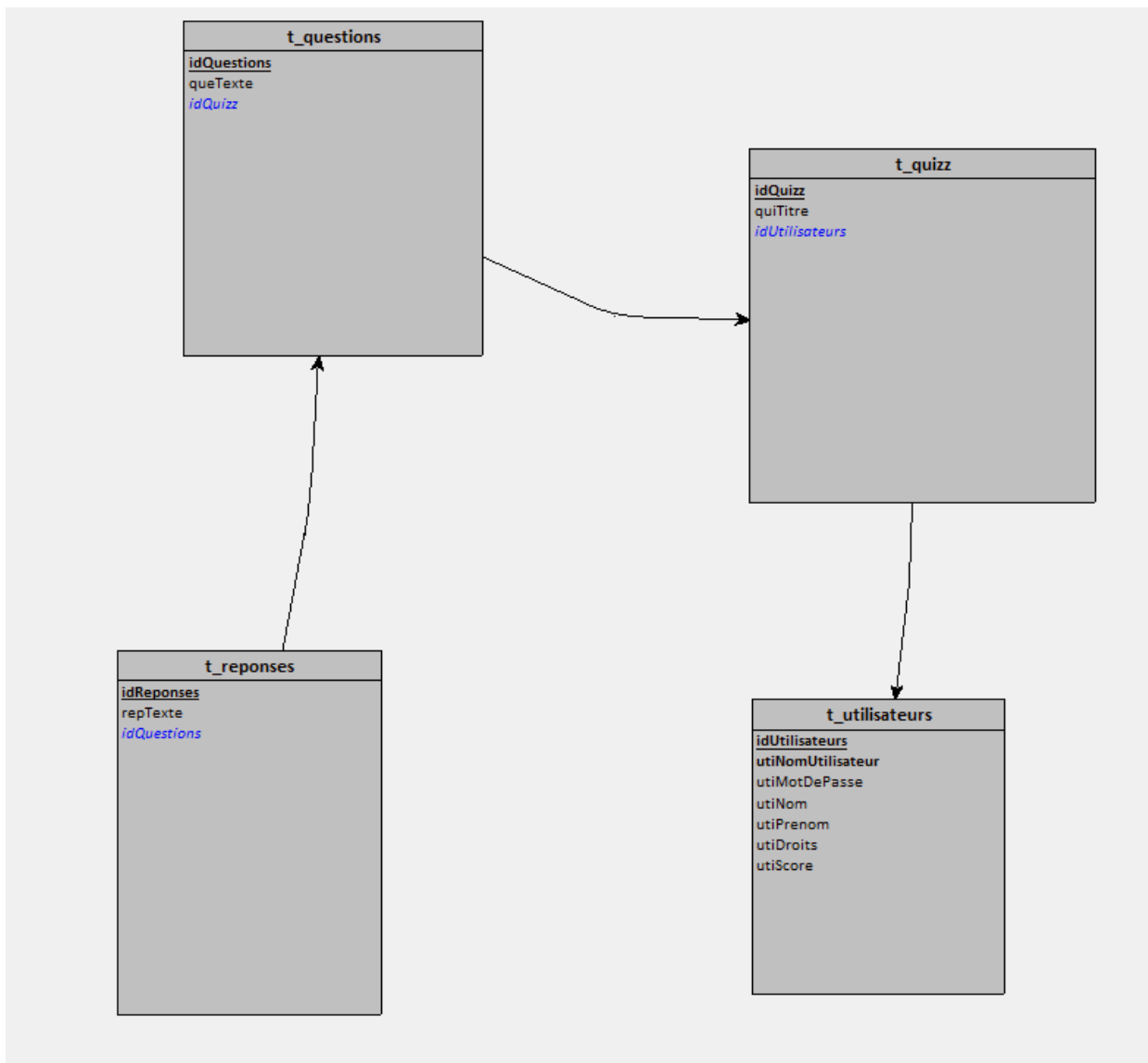


Figure 10 Modèle logique des données

3.2 Stratégie de test

- Test unitaires à l'aide de fonctions PHP et requêtes SQL
- Test d'une fonction sous diverses conditions, par exemple exécuter une requête avec différents paramètres possibles (numéro, lettre, etc...)
- Test « manuel » en utilisant l'application moi-même ou par une personne externe, en cherchant les potentiels situations sensibles (changer code html, rafraichir à un moment pas prévu, insérer des données pas attendues afin de voir si l'application tient la route)

3.3 Risques techniques

Durant ce projet je ne vois pas de potentiel problèmes que je risque de rencontrer au niveau du développement sur site.

Cependant ceci est le premier projet qui sera conçu pour être utilisé par un nombre conséquent d'utilisateurs simultanément, il est possible que certains problèmes inattendus surviennent lors des tests avec plusieurs utilisateurs.

3.4 Planification

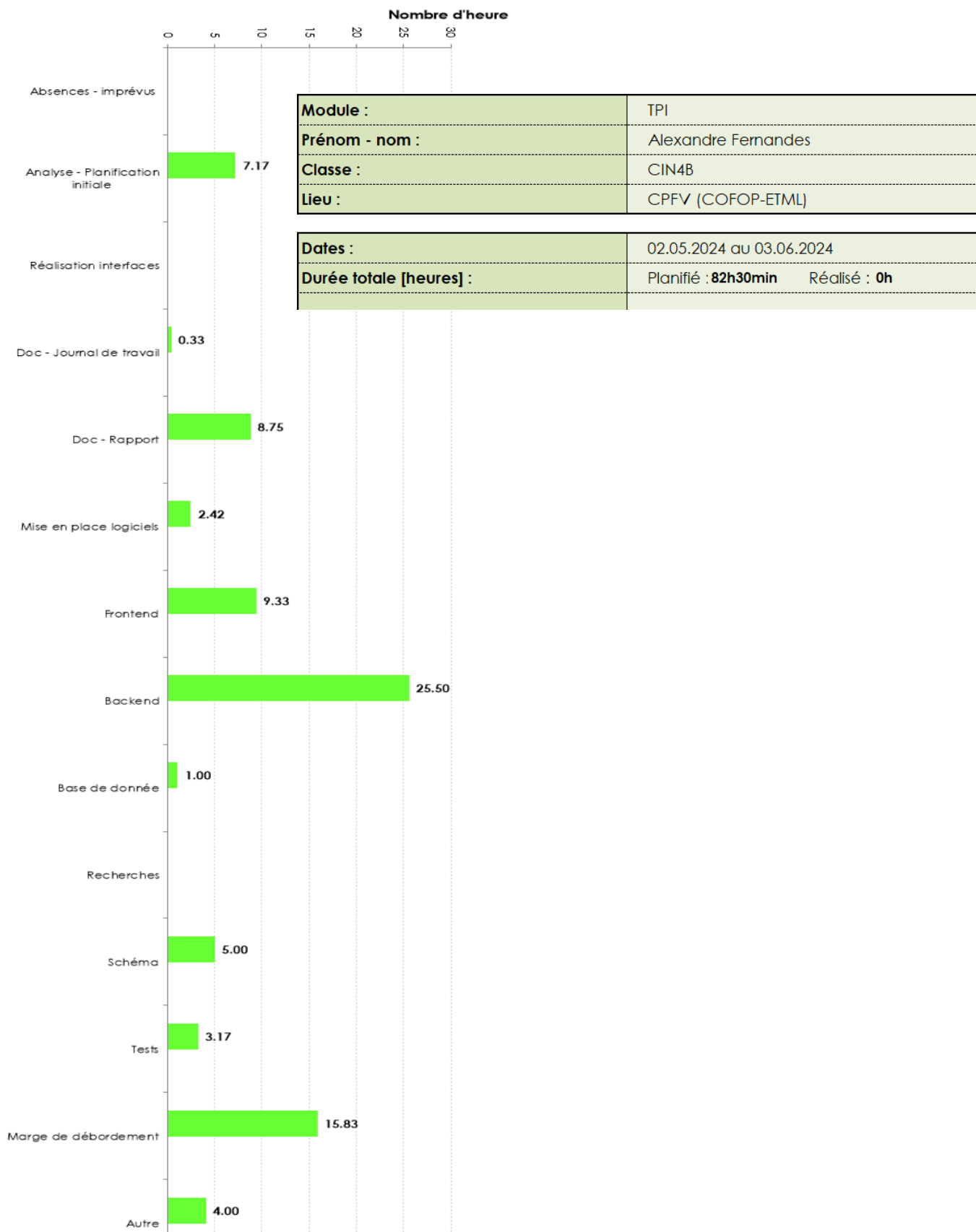


Figure 11 Graphique : Bilan de la planification initiale

3.5 Dossier de conception

3.5.1 Matériel Hardware

- Ordinateur ETML
 - Windows 10 Éducation 22H2 x64
 - 2.50 GHz Intel Core i7-11700 CPU
 - 32 GO RAM

3.5.2 Application / Logiciels / Outils

- PHPMyAdmin 6.0
 - PhpMyAdmin est une application Web de gestion pour les systèmes de gestion de base de données MySQL et MariaDB
 - Durant ce projet il a été utile pour simplifier la gestion de la base de données MariaDB
- PHP 8.3.4
 - PHP : Hypertext Preprocessor, est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur web, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale.
 - Un des codes utiliser durant mon projet, il a par exemple été utiliser pour la communication entre la page web et la base de données
- MariaDB server 11.3.2
 - MariaDB est un système de gestion de base de données.
 - Durant le projet il m'a permis de gérer ma base de données
- Visual Studio Code
 - Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS. Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, les snippets, la refactorisation du code et Git intégrer.
 - Il m'a été utile afin de programmer mon code
- jQuery 7.3.1
 - jQuery est une bibliothèque JavaScript libre et multiplateforme créée pour faciliter l'écriture de scripts côté client dans le code HTML des pages web
 - Un des langages de programmation utilisé dans mon code, il m'est utile afin de traiter des données et faire des retours d'erreurs
- GitHub Desktop 3.3.17
 - Société d'hébergement de développement
 - Me permet d'héberger mon projet et avoir un système de version
 - <https://github.com/Medellin05001/TPI-QuizMaster> <== Mon dépôt

- Bootstrap 4.5.2
 - Bootstrap est une collection d'outils utiles à la création du design de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option
 - M'est utile afin de gagner du temps en évitant de refaire des interfaces existantes et la mise en page
- Suite Microsoft 365
 - Suite de logiciels utiles à la réalisation du projet tel que la documentation et la communication
- Looping
 - Looping est un logiciel de modélisation conceptuelle de données entièrement gratuit et libre d'utilisation.
 - M'a été utile pour réaliser le MCD MLD
- Photopea
 - Photopea est un logiciel propriétaire de retouche et de traitement d'images matricielles et vectorielles
 - Ce logiciel en ligne m'a permis de modifier des images tel que mes maquettes web
- Figma
 - Figma est un éditeur de graphiques vectoriels et un outil de prototypage
 - Cet outil m'a été pratique lors de la réalisation des maquettes

3.5.3 Site web

- Design
 - Simple & Friendly afin de faciliter l'utilisation et les éléments inutiles
- Animation css/javascript
 - Minimaliste voire inexistants afin de ne pas impacter les performances web
- Priorité
 - Propreté du code la simplicité/optimisation afin qu'il puisse être repris facilement dans le futur
- Base de données et code
 - Un serveur MariaDB, PHP et PHPMyAdmin seront exécutés localement à l'aide de l'invite de commande pour faire tourner l'application web durant le TPI
- Sauvegardes
 - Un commit Git sera effectué chaque fin de journée de travail

3.5.4 Suivi de projet

Afin de garantir un suivi et une transmission du projet correct, diverses mesures ont été mises en place.

Les fichiers du programme en lui-même sera déposer dans un [répertoire git](#)² à chaque fin de journée de travail.

Chaque jour, je tiendrai un journal de travail qui contiendra également le planning initial dans un fichier Excel. Ce journal de travail détaillera toutes les tâches de la journée par tranche de 5 minutes et sera rempli en même temps que je réaliserai ces tâches. J'enverrai ce fichier par e-mail chaque fin d'après-midi aux experts et au chef de projet.

3.5.5 Base de données : MariaDB

- **Table t_utilisateurs :**
 - (INT 11) idUtilisateurs : Clé primaire, identifiant unique pour chaque utilisateur.
 - (VARCHAR 20) utiNomUtilisateur : Nom d'utilisateur, unique pour chaque utilisateur.
 - (VARCHAR 64) utiMotDePasse : Mot de passe de l'utilisateur (sécurisé par hachage).
 - (VARCHAR 50) utiNom : Nom de l'utilisateur.
 - (VARCHAR 35) utiPrenom : Prénom de l'utilisateur.
 - (VARCHAR 20) utiDroits : Les droits de l'utilisateur, par exemple, "utilisateur", "admin"
 - (INT 11) utiScore : Le score de l'utilisateur dans le contexte du quizz.
- **Table t_quizz :**
 - (INT 11) idQuizz : Clé primaire, identifiant unique pour chaque quizz.
 - (VARCHAR 50) quiTitre : Titre du quizz.
 - (INT 11) fkUtilisateurs : Clé étrangère faisant référence à l'ID de l'utilisateur qui a créé le quizz.

² <https://github.com/Medellin05001/TPI-QuizMaster>

- **Table t_questions :**
 - (INT 11) idQuestions : Clé primaire, identifiant unique pour chaque question.
 - (VARCHAR 255) queTexte : Texte de la question.
 - (INT 11) fkQuizz : Clé étrangère faisant référence à l'ID du quizz auquel cette question est associée.
- **Table t_reponses :**
 - (INT 11) idReponses : Clé primaire, identifiant unique pour chaque réponse.
 - (VARCHAR 255) repTexte : Texte de la réponse.
 - (INT 11) fkQuestions : Clé étrangère faisant référence à l'ID de la question à laquelle cette réponse est associée.
- **Requêtes :**
 - **Authentification** : Quand les données de connexion sont insérées, le code envoie une requête qui permet de vérifier si l'utilisateur existe, s'il existe on compare le mot de passe dans la base de donnée (haché) et le mot de passe inséré par l'utilisateur (haché en backend par le code) afin d'approuver la connexion s'ils correspondent sinon il refuse la connexion
 - **Classement** : Une requête obtient tous les joueurs en fonction de leurs nombres de points dans l'ordre croissants de leurs points en leur attribuant une position dans une variable temporaire lors de la requête. Si deux joueurs ont le même score, c'est le joueur qui a le compte le plus récent qui est pris en priorité (ID plus petit)
 - **Affichage des quiz** : Sur la page d'accueil, une requête est exécutée afin d'afficher tous les quiz de la base de données (du plus récent au moins récent)
 - **Récupération information quiz spécifique** : Lors de la sélection d'un quiz, le code envoie une requête qui permet d'obtenir toutes les informations du quiz en prenant compte de toutes les tables (t_quizz, t_question, t_reponse)
 - **Ajout score** : Quand un utilisateur finit un quiz, son score est calculé. S'il est supérieur à 0 le score s'additionne à celui actuel.

- **Création quiz** : Quand un administrateur crée un quiz, le code envoie 4 requêtes.
 - Création quiz (t_quizz)
 - Création question (t_question)
 - Création réponse (t_reponse)
 - Une requête sera chargée de récupérer l'id de l'élément qui vient d'être créé afin de la transmettre la clé étrangère
- **Suppression quiz** : Quand un administrateur supprime un quizz, le code envoie 3 requêtes.
 - Suppression réponses (t_reponses)
 - Suppression questions (t_question)
 - Suppression quiz (t_quizz)
 - Tout cela est englober dans un transactions afin de tout annuler s'il y a une erreur
- **Affichage des utilisateurs** : Quand un administrateur est sur la page de la liste des utilisateurs exécutée afin d'afficher tous les utilisateurs de la base de données. 2 boutons sont également présents (afin de modifier et supprimer l'utilisateur)
- **Modification utilisateur** : Lorsque l'utilisateur insère les nouvelles données de l'utilisateur, une requête est exécutée qui met à jour la base de données au niveau des données de l'utilisateurs
- **Suppression utilisateur** : Quand un administrateur supprime un utilisateur en appuyant sur le bouton prévu pour, 4 requêtes sont envoyées
 - Suppression réponses de l'utilisateur (t_reponses)
 - Suppression questions l'utilisateur (t_question)
 - Suppression quiz l'utilisateur (t_quizz)
 - Suppression de l'utilisateur (t_utilisateurs)
 - Tout cela est englober dans un transactions afin de tout annuler s'il y a une erreur

3.5.6 Architecture du programme

Durant ce projet j'ai réalisé une architecture spécifique qui consiste à séparer les pages visibles par l'utilisateur et les « pages systèmes ».

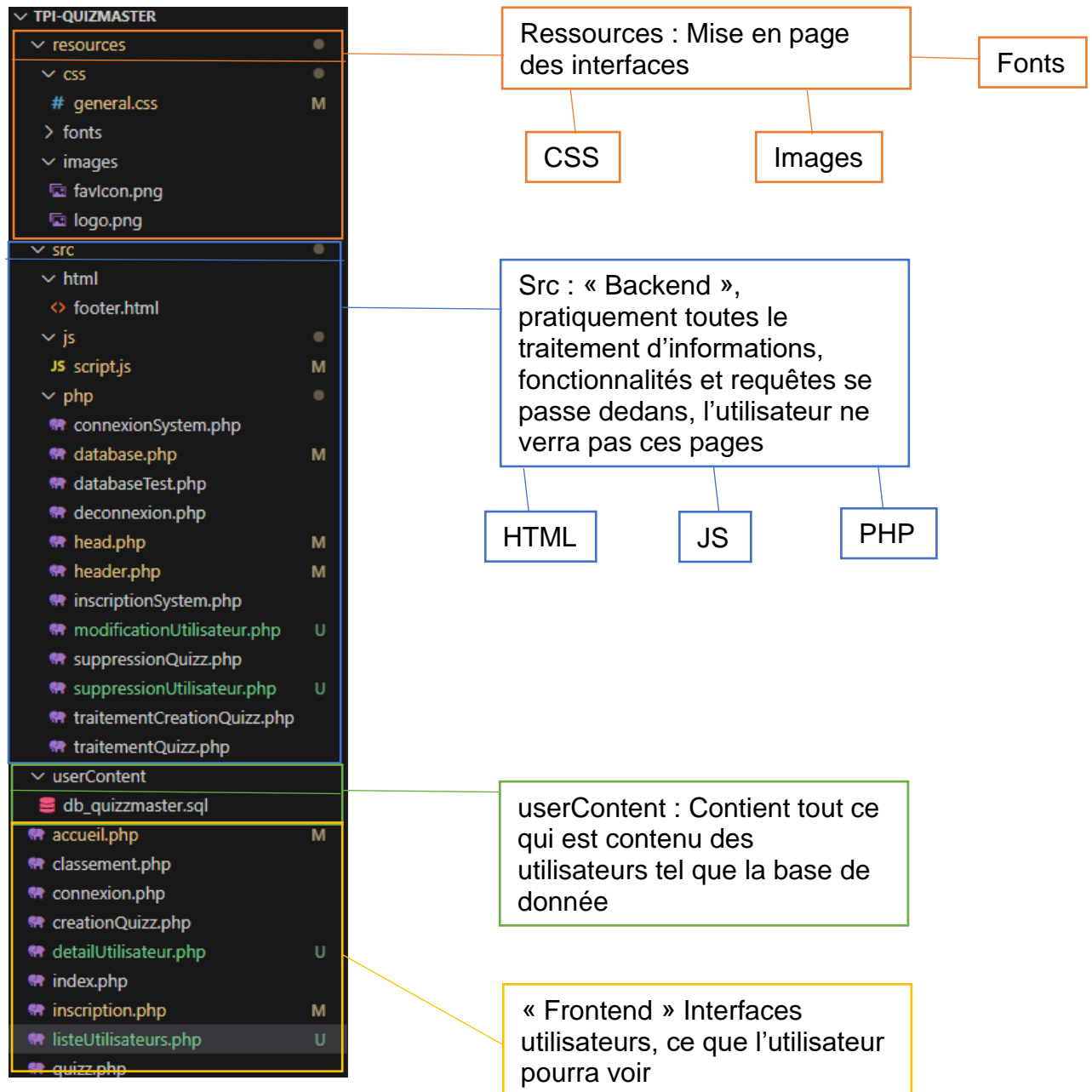


Figure 12 Architecture : programme entier

3.5.7 Dépendance externe

Durant ce projet diverses liaisons externes au code ont été utilisées, afin d'avoir une vue globale sur ceux-ci, ils sont listés ici :

- Bootstrap : Mise en page facilitée, utilisé dans toutes les interfaces utilisateurs, utilisation de bout de code HTML/CSS et javascript déjà disponible gratuitement sur internet afin d'accélérer la création d'interfaces
- Flexbox : Aide à la mise en page des éléments lors du CSS, comme pour l'alignement des éléments
- jQuery : Bibliothèque javascript qui permet d'utiliser des syntaxes simplifiées et des fonction déjà prêtes et optimisées

3.5.8 Utilisation schématisée

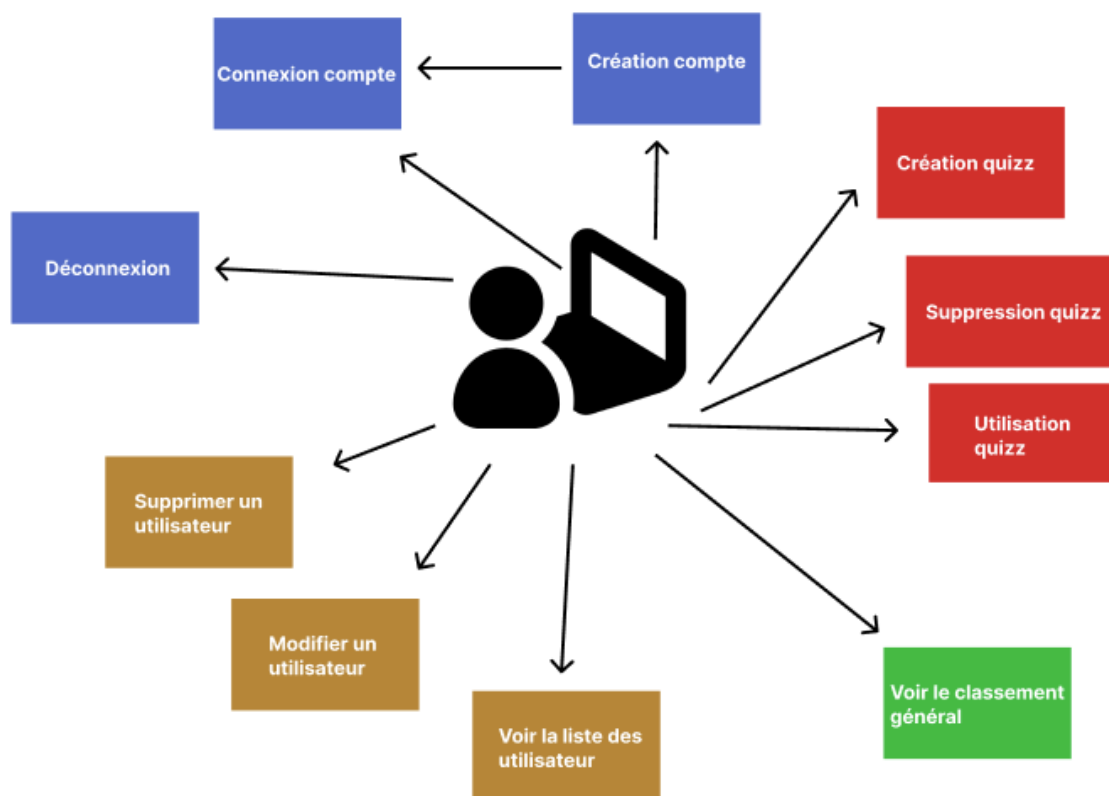


Figure 13 Maquette : Utilisation possibles

4 Réalisation

4.1 Liste des fichiers : Répertoire

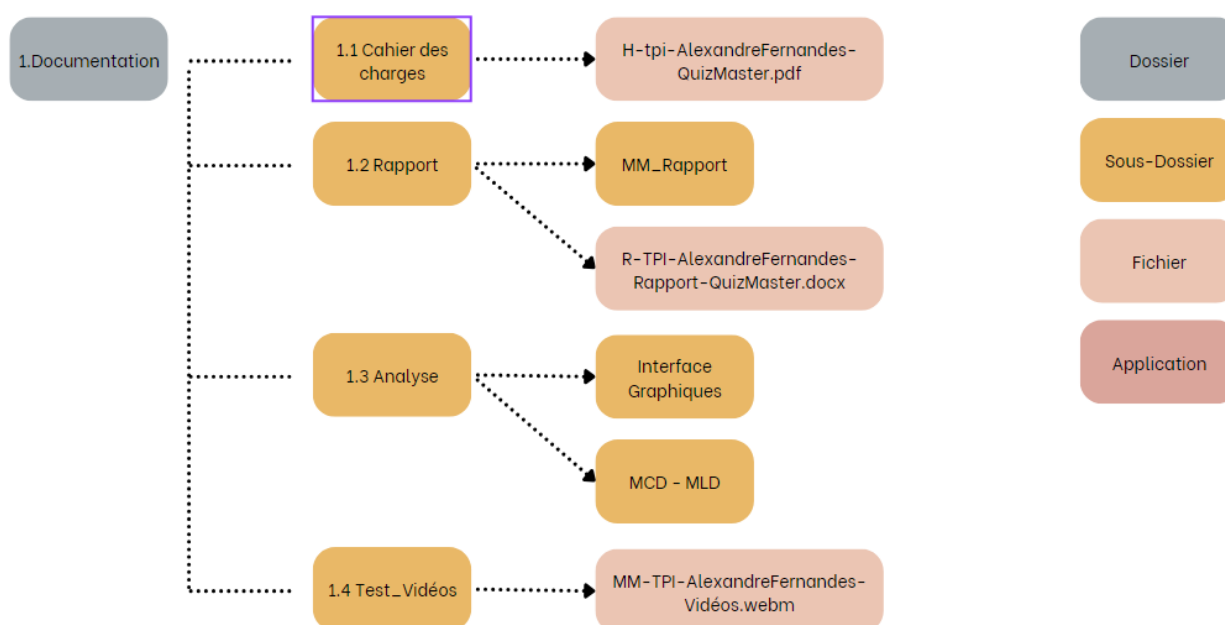


Figure 14 Répertoire : Projet (1)

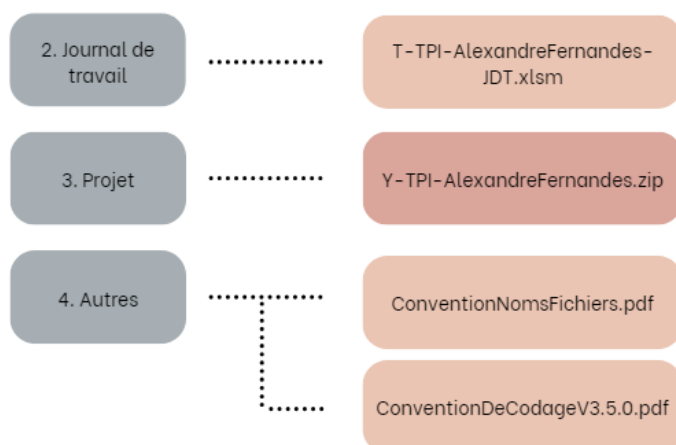


Figure 15 Répertoire : Projet (2)

4.2 Liste des fichiers : Code source

		« rouge » Dossier	« bleu » fichier
▼ TPI-QUIZMASTER			
▼ resources		« resource » Mise en page des interfaces	
▼ css		« css » Contient tous les fichiers css	
# general.css	M	css utilisé pour l'ensemble du site web	
> fonts		« fonts » Contient toutes les polices d'écritures (vide)	
▼ images		« images » Contient toutes les images/logo	
favicon.png		logo utilisé pour au niveau de l'onglet	
logo.png		logo utilisé pour le header	
▼ src		« src » traitement d'informations, fonctionnalités et requêtes	
▼ html		« html » Contient toutes les pages uniquement html du site	
<> footer.html		footer présent en bas de page	
▼ js		« js » Contient tous les fichier script du programme	
JS script.js	M	les différents scripts sont stockés dedans	
▼ php		« php » Contient tous les fichier php qui contribuent au traitement de données	
connexionSystem.php		Traitement des données et requêtes lors de la connexion	
database.php	M	Contient toutes les requêtes SQL du programme	
databaseTest.php		Tests Unitaires	
deconnexion.php		Système qui permet de déconnecter l'utilisateur	
head.php	M	Différentes liaisons, cookies, session start sont stockées ici	
header.php	M	La navBar est entièrement codée ici	
inscriptionSystem.php		Traitement des données et requêtes lors de la création d'un compte	
modificationUtilisateur.php	U	Code pour modifier un utilisateur	
suppressionQuizz.php		Code pour supprimer un quizz	
suppressionUtilisateur.php	U	Code pour supprimer un utilisateur	
traitementCreationQuizz.php		Traitement des données et requêtes lors de la création d'un quizz	
traitementQuizz.php		Traitement des données et requêtes lors de l'envoi d'un quizz par un utilisateur	
▼ userContent		« userContent » Données utilisateurs	
db_quizzmaster.sql		Base de données	
accueil.php	M	Interface utilisateur : Accueil	
classement.php		Interface utilisateur : Classement	
connexion.php		Interface utilisateur : Connexion	
creationQuizz.php		Interface utilisateur (admin) : Création quizz	
detailUtilisateur.php	U	Interface utilisateur (admin) : Information d'un utilisateur	
index.php		Index (routeur)	
inscription.php	M	Interface utilisateur : Inscription	
listeUtilisateurs.php	U	Interface utilisateur : Liste des utilisateurs	
quizz.php		Interface utilisateur : Utilisation quizz	

Figure 16 Répertoire : Programme

4.3 Base de données : Comment fonctionnent les requêtes

```
6 references | 0 overrides
public function __construct(){
    try{
        $this->connector = new PDO('mysql:host=localhost;dbname=db_quizzmaster', "root", "root");
    }catch(PDOException $err){
        die("Erreur de connexion");
    }
    $this->connector->query("SET NAMES UTF8");
}

4 references
private function querySimpleExecute($query){
    return $this->connector->query($query);
}

8 references
private function queryPrepareExecute($query, $binds){
    $req = $this->connector->prepare($query);
    for($i = 0;$i < count($binds); $i++){
        {
            $req->bindValue($binds[$i]['paramName'],$binds[$i]['paramValue'],$binds[$i]['paramType']);
        }
    }
    $req->execute();
    return $req;
}

10 references
private function formatData($req){
    return $req->fetchAll(PDO::FETCH_ASSOC);
}
```

Figure 17 SQL : Requêtes préparées

Afin de faire fonctionner les requêtes de façon sécuriser, une méthode a été mis en place.

`__construct ()` : Initialise la connexion à la base de données en PDO, permet de se connecter à une base de données MySQL. Elle est appelée à chaque nouvelle création d'instance de la classe, notamment dans le header.

`querySimpleExecute ($query)` : Permet de prendre une requête en paramètre, de l'exécuter, et de retourner son résultat obtenu. Elle est utilisée pour exécuter les requêtes sans paramètres

`QueryPrepareExecute ($query, $binds)` : Prend la requête SQL, la lie avec les paramètres puis l'exécute. Elle est utilisée pour exécuter des requêtes préparées avec des paramètres dynamiques, afin de protéger des injections SQL.

En préparant les requêtes, les requêtes seront traitées comme des valeurs littérales et non du code SQL

formatData (\$req) : Permet de formater les données sous formes de tableau, le code extrait les résultats sous forme de tableau associatif dans lequel les clés deviennent des noms de colonnes et les valeurs deviennent les données.
Cela me permet de manipuler les données plus facilement de l'exécution de la requête : \$utilisateur[0]['utiNom']

4.4 Base de données : Authentification

Les premières fonctions qui furent implémentées furent celles liées à l'authentification. Cela inclut la connexion, la création de compte (inscription) et la déconnexion.

4.4.1 Requêtes préparées utilisées

Requête qui permet de sélectionner un utilisateur dans la base de données, utile pour vérifier si le nom d'utilisateur est disponible et récupérer les informations d'un utilisateur spécifique.

```
public function connect($nomUtilisateur){  
    $query = ("SELECT * FROM t_utilisateurs WHERE utiNomUtilisateur = :nomUtilisateur");  
    $binds = [['paramName' => 'nomUtilisateur', 'paramValue' => $nomUtilisateur, 'paramType' => PDO::PARAM_STR]];  
    $statement = $this->queryPrepareExecute($query,$binds);  
    return $this->formatData($statement);  
}
```

Figure 18 SQL : Connexion utilisateur

Cette requête SQL permet créer un nouvel utilisateur avec les paramètres reçus.

```
public function addUser($login,$password,$nom,$prenom){  
    $query = "INSERT INTO `t_utilisateurs` (`utiNomUtilisateur`, `utiMotDePasse`, `utiNom`, `utiPrenom`) VALUES (:login,:password,:nom,:prenom)";  
    $binds = [  
        ["paramName" => "login", "paramValue" => $login, "paramType" => PDO::PARAM_STR],  
        ["paramName" => "password", "paramValue" => $password, "paramType" => PDO::PARAM_STR],  
        ["paramName" => "nom", "paramValue" => $nom, "paramType" => PDO::PARAM_STR],  
        ["paramName" => "prenom", "paramValue" => $prenom, "paramType" => PDO::PARAM_STR]  
    ];  
    $statement = $this->queryPrepareExecute($query,$binds);  
    return $this->formatData($statement);  
}
```

Figure 19 SQL : Création d'utilisateur

4.4.2 Connexion

L'interface utilisateur qui permet la connexion se passe sur la page « connexion.php », un formulaire est présent afin de réceptionner et envoyer les informations que l'utilisateur inscrit.

Les données sont envoyées sur la page « script.js », plus précisément dans la fonction « \$("#connectSubmit").click ».

Le rôle de cette fonction est de récupérer les informations et vérifier si elles correspondent aux deux restrictions imposées dans le code puis retourner à l'utilisateur le résultat :

- Input « nom d'utilisateur » vide → Veuillez entrer un nom d'uti...
- Input « mot de passe » vide → Veuillez entrer un mot de passe

```
/**
 * Traitement des données de connexion
 */
$("#connectSubmit").click(function() {
    var validationInput = 0;

    // Récupère le nom d'utilisateur introduit et vérifie s'il est vide
    var login = $("#nomUtilisateurInput").val();
    if (login === "") {
        $("#erreur_nomUtilisateur").html("Veuillez entrer un nom d'utilisateur !");
    } else {
        $("#erreur_nomUtilisateur").html("");
        validationInput++;
    }

    // Récupère le mot de passe introduit et vérifie s'il est vide
    var password = $("#motDePasseInput").val();
    if (password === "") {
        $("#erreur_motDePasse").html("Veuillez entrer un mot de passe !");
    } else {
        $("#erreur_motDePasse").html("");
        validationInput++;
    }
}
```

Figure 20 jQuery : Restriction Input connexion

Si toutes les conditions sont respectées les informations sont renvoyées en POST et les informations en paramètre sur la page « connexionSystem.php » à l'aide d'ajax. Ajax permet d'envoyer les informations mais également d'en recevoir afin que l'utilisateur reçoive un retour (connexion réussie ou ratée).

```
// Si toutes les données introduites sont correctes, il les renvoie en paramètres POST dans la page qui gère tout la suite de la connexion
if (validationInput === 2) {
    $.ajax({
        method: "POST",
        url: "src/php/connexionSystem.php",
        data: { username: login, password: password }
    })
    .done(function(response) {
        $("#solde").html(response);
    });
}
```

Figure 21 Ajax : Renvoi de page connexion en POST

ConnexionSystem.php récupère les informations envoyées en post (par script.js).

La page vérifie si les données sont bien présentes stock les données de la base de données concernant le nom d'utilisateur entré
Cela est réalisé par une requête SQL « connect »

Le mot de passe récupérer en POST sera hacher (sha256) afin de le vérifier avec celui de la variable qui contient les informations de la base de données

La connexion est approuvée ou refusée en fonction du résultat

```

<?php
// Insertion des cookies, démarrage d'une session php, et connexion à la base de donnée
ini_set('session.cookie_lifetime', 60 * 60 * 24 * 365);
ini_set('session.gc-maxlifetime', 60 * 60 * 24 * 365);
SESSION_START();
require_once("database.php");
$db = new Database();

// Récupération du nom d'utilisateur et mot de passe envoyés en paramètres depuis le script javascript
$username = $_POST['username'];
$password = $_POST['password'];

// Stocker les données qui permet la redirection à la page index
$indexRedirection = "><script>window.location.replace('../index.php');</script><?php";

// Vérifie si des données sont bien présentes
if (isset($username) && isset($password)){
    // Vérifie si l'utilisateur existe bien dans la base de données
    if($utilisateur = $db->connect($username)){
        $hashedPassword = hash('sha256', $password);

        // Vérifie si l'utilisateur et le mot de passe introduit correspondent
        if ($utilisateur[0]['utiMotDePasse'] == $hashedPassword){
            $_SESSION["login"] = $username;
            echo $indexRedirection;
        }else{
            echo "Le nom d'utilisateur et le mot de passe ne correspondent pas !";
        }
    } else {
        echo "L'utilisateur est introuvable !";
    }
}else{
    echo $indexRedirection;
}

```

Figure 22 PHP : Traitement des données de connexion

4.4.3 Inscription

L'utilisateur va pouvoir se créer un compte sur la page « inscription.php ». Comme pour la page connexion, l'utilisateur se retrouve avec un formulaire à remplir puis à envoyer afin de procéder à l'inscription.

Le processus est similaire à celui de la connexion mais avec plus de paramètres, les données sont premièrement envoyées sur la page « script.js » sur la fonction \$("#registerSubmit").click

Cette fonction va récupérer les informations et vérifie si elles sont conformes aux exigences du programme. La page retourne des erreurs dans le formulaire afin d'indiquer à l'utilisateur ses erreurs.

```

// Récupère le nom de famille introduit et vérifie s'il est correct
var nom = $("#nomInput").val();
if (nom == ""){
    $("#erreur_nom").html("Veuillez entrer un nom !");
} else {
    if(!nom.match(/^[a-zA-Z0-9-]{3,50}$/)){
        $("#erreur_nom").html("Le nom doit comporter au minimum 3 caractères et au maximum 50 (Lettres et '-' uniquement!)");
    }else{
        $("#erreur_nom").html("");
        validationInput++;
    }
}

// Récupère le prénom introduit et vérifie s'il est correct
var prenom = $("#prenomInput").val();
if (prenom == "") {
    $("#erreur_prenom").html("Veuillez entrer un prenom !");
} else {
    if(!prenom.match(/^[a-zA-Z0-9-]{3,35}$/)){
        $("#erreur_prenom").html("Le prenom doit comporter au minimum 3 caractères et au maximum 35 (Lettres et '-' uniquement) !");
    }else{
        $("#erreur_prenom").html("");
        validationInput++;
    }
}

```

Figure 23 jQuery: Restriction Input inscription

Liste des restrictions pour l'inscription :

- Nom de famille : Uniquement lettres, minimum 3 et maximum 50
- Prénom : Uniquement lettres, minimum 3 et maximum 35
- Nom d'utilisateur : Lettres et chiffres, minimum 3 et maximum 20
- Mot de passe : Uniquement lettres et chiffres, minimum 8 et maximum 64

Si toutes les conditions sont respectées les informations sont renvoyées en POST et les informations en paramètre sur la page « inscriptionSystem.php » à l'aide d'ajax. Ajax permet d'envoyer les informations mais également d'en recevoir afin que l'utilisateur reçoive un retour (connexion réussie ou ratée).

```
// Si toutes les données introduites sont correctes, il les renvoie en paramètres POST dans la page qui gère tout la suite de l'inscription
if (validationInput === 4) {
    $.ajax({
        method: "POST",
        url: "src/php/inscriptionSystem.php",
        data: { username: login, password: password, nom: nom, prenom: prenom }
    })
    .done(function(response) {
        $("#solde").html(response);
    });
}
```

Figure 24 Ajax : Renvoi de page inscription en POST

InscriptionSystem.php récupère les informations envoyées en post (par script.js)

La page vérifie si les données sont bien présentes. Le code vérifie si l'utilisateur est déjà présent dans la base de données (requête SQL « connect »).

Si le nom d'utilisateur est disponible, le mot de passe introduit dans le formulaire est haché (sha256) puis un compte est créé avec toutes les données inscrites dans le formulaire à l'aide de la requête SQL « addUser »

```
<?php
// Insertion des cookies, démarrage d'une session php, et connexion à la base de donnée
ini_set('session.cookie_lifetime', 60 * 60 * 24 * 365);
ini_set('session.gc-maxlifetime', 60 * 60 * 24 * 365);
SESSION_START();
require_once("database.php");
$db = new Database();

// Récupération du nom d'utilisateur et mot de passe envoyés en paramètres depuis le script javascript
$username = $_POST['username'];
$password = $_POST['password'];
$nom = $_POST['nom'];
$prenom = $_POST['prenom'];

// Stocker les données qui permet la redirection à la page index
$indexRedirection = "<?><script>window.location.replace('../index.php');</script><?php";

// Stocker les données qui permet la redirection à la page connexion
$connexionSuccess = "<?><script>alert('La création du compte à été complétée avec succès');window.location.replace('../connexion.php');</script><?php";

// Vérifie si des données sont bien présentes
if (isset($username) && isset($password) && isset($nom) && isset($prenom)){
    // Vérifie le nom d'utilisateur introduit existe déjà dans la base de donnée
    if(!$db->connect($username)){
        $hashedPassword = hash('sha256', $password);

        // Création de l'utilisateur et redirection
        $db->addUser($username, $hashedPassword, $nom, $prenom);
        echo ($connexionSuccess);
    }else{
        echo "Ce nom d'utilisateur n'est plus disponible !";
    }
}else{
    echo $indexRedirection;
}
?>
```

Figure 25 PHP : Traitement des données d'inscription

4.4.4 Déconnexion

Afin d'effectuer la connexion, j'ai inclus dans le header un texte « Déconnexion » qui permet de se déconnecter en cliquant dessous tout simplement.

Quand l'utilisateur clique dessus il est renvoyé vers la page deconnexion.php qui détruit la session active tout simplement.

```
// Détruit la session actuel de l'utilisateur et le redirige vers la page index
session_start();
session_destroy();
header('Location: ../index.php');
exit;
```

Figure 26 PHP : déconnexion utilisateur

4.5 Base de données : Affichage tous les quiz

4.5.1 Requêtes préparées utilisées

Avec cette requête, on peut récupérer les informations du créateur du quiz et les informations du quiz

```
/**
 * Permet d'obtenir les informations de bases de tout les quizz de la base de donnée
 */
1 reference | 0 overrides
public function getAllQuizz(){
    $quizzes = $this->querySimpleExecute("SELECT * FROM t_quizz INNER JOIN t_utilisateurs ON t_quizz.fkUtilisateurs = t_utilisateurs.idUtilisateurs");
    return $this->formatData($quizzes);
}
```

Figure 27 SQL : obtenir tous les quiz

4.5.2 Affichage des quiz

Quand l'utilisateur est sur la page d'accueil, le code récupère tous les quizz afin de les stocker dans une variable

```
$quizzes = $db->getAllQuizz();
```

Figure 28 Code : stocker tous les quiz

Une boucle foreach va parcourir la variable afin d'afficher le titre du quiz et son créateur. Cette boucle est affichée sous forme de carte pour l'utilisateur.

```
// Affichage de tout les quizz, du plus récent au plus vieux
foreach($quizzes as $quizz){
    ?>
    <div class="col-lg-4 col-md-6 col-sm-12 mb-4">
        <?php echo"<a class='card-link' href='quizz.php?id=". $quizz['idQuizz'] ."'>"; ?>
        <div class="card bg-dark text-white">
            <div class="card-body text-center">
                <h5 class="card-title"><?php echo($quizz['quiTitre']); ?></h5>
            </div>
            <div class="card-footer">
                <p class="card-text card-text-small">Par : <?php echo($quizz['utiNom'])." ".$quizz['utiPrenom']; ?></p>
            </div>
        </div>
    </a>
</div>
<?php
}
```

Figure 29 PHP : Affichage de tous les quiz

4.6 Base de données : Affichage quiz spécifique

4.6.1 Requêtes préparées utilisées

Cette requête permet d'obtenir l'entièreté des informations d'un quiz spécifique (Question, Réponse, Titre du Quiz). Dans cette requête on relie toutes les tables.

```
/**
 * Permet d'obtenir toutes les informations d'un quizz en particulier
 * param $idquizz => id du quizz en question que l'on souhaite obtenir l'information
 */
2 references [0 overrides]
public function getQuizz($idQuizz){
    $query = ("SELECT t_questions.idQuestions, t_questions.queTexte, t_reponses.idReponses, t_reponses.repTexte, t_quizz.idQuizz, t_quizz.quiTitre, t_utilisateurs.idUtilisateurs,
    t_utilisateurs.utiNomUtilisateur, t_utilisateurs.utiNom, t_utilisateurs.utiPrenom, t_utilisateurs.utiDroits, t_utilisateurs.utiScore FROM t_questions JOIN t_reponses
    ON t_questions.idQuestions = t_reponses.fkQuestions JOIN t_quizz ON t_questions.fkQuizz = t_quizz.idQuizz JOIN t_utilisateurs
    ON t_quizz.fkUtilisateurs = t_utilisateurs.idUtilisateurs WHERE t_quizz.idQuizz = :idQuizz;");
    $binds = [
        ["paramName" => "idQuizz", "paramValue" => $idQuizz, "paramType" => PDO::PARAM_INT]
    ];
    $statement = $this->queryPrepareExecute($query,$binds);
    return $this->formatData($statement);
}
```

Figure 30 SQL : obtenir un quiz spécifique

4.6.2 Affichage du quiz

Avec cette ligne de code, on peut accéder obtenir un lien vers la page quizz comportant un paramètre dans le lien : l'id du quiz.

```
<?php echo "<a class='card-link' href='quizz.php?id=". $quizz["idQuizz"] ."'>"; ?>
```

Figure 31 PHP : accès à la page d'un quiz spécifique

Afin d'afficher les informations du quiz, on stocke tout dans une variable.

```
//Récupération des donnée du quizz
$quizz = $db->getQuizz($idQuizz);
```

Figure 32 PHP : stock les informations du quiz

Pour afficher toutes les questions, on utilise une boucle foreach qui va parcourir toutes les questions en les affichants.

```
<!-- Form quand l'utilisateur n'a pas soumis de réponses -->
<form id="quizz" action="src/php/traitementQuizz.php" method="post">
    <?php
    $counter = 1;
    foreach($quizz as $question){
        echo "<div class='question'>";
        echo "<label>" . $question['queTexte'] . "</label>";
        echo "<input type='text' name='reponse' . $counter . "'>";
        echo "</div><br>";
        $counter++;
    }
    ?>
    <!-- Ce label permet d'envoyer l'id du quizz en donnée dans le formulaire-->
    <input type="hidden" name="idQuizz" value="<?php echo $idQuizz; ?>">

    <input type="submit" value="Soumettre">
</form>
<?php
}
?>
</div>
```

Figure 33 PHP : Affichage du quiz

4.7 Base de données : Traitement du quiz

4.7.1 Requêtes préparées utilisées

Cette requête permet tout simplement de mettre à jour le score de l'utilisateur après l'envoi d'un nouveau quizz, afin d'ajouter ses nouveaux points.

```
/**
 * Permet d'ajouter un score à un utilisateur
 * param $score => score à additionner à son score actuel
 * param $idUtilisateur => id de l'utilisateur qui reçoit un nouveau score
 */
1 reference | 0 overrides
public function addScore($score, $idUtilisateur){
    $query = ("UPDATE `t_utilisateurs` SET `utiScore` = `utiScore` + :score WHERE idUtilisateurs = :idUtilisateur");
    $binds = [
        ["paramName" => "score", "paramValue" => $score, "paramType" => PDO::PARAM_INT],
        ["paramName" => "idUtilisateur", "paramValue" => $idUtilisateur, "paramType" => PDO::PARAM_INT]
    ];
    $statement = $this->queryPrepareExecute($query,$binds);
    return $this->formatData($statement);
}
```

Figure 34 SQL : update le score de l'utilisateur

4.7.2 Traitement

Quand un quiz est envoyé, le code va se charger de récupérer les informations et de les envoyer dans la page « traitementQuizz.php »

Cette page va récupérer les réponses que l'utilisateur a entré et les stocker dans une variable

Elle va également récupérer les réponses de la base de données (réponses correctes) afin de les stocker dans une seconde variable

Le code va se charger de comparer les deux tableaux, quand une réponse est correcte, il gagne un point. Quand une réponse est fausse, il perd un point.

Sachant qu'il ne peut pas avoir un score en dessous de 0 points.

```
// Récupération des réponses soumises par l'utilisateur
$reponsesUtilisateur = [];
foreach ($_POST as $key => $value) {
    if (strpos($key, 'reponse') === 0) {
        $reponsesUtilisateur[$key] = $value;
    }
}

// Récupération des réponses attendues depuis la base de données via la variable $quizz
$reponsesAttendues = [];
foreach ($quizz as $key => $question){
    $reponsesAttendues[$key] = $question['repTexte'];
}

// Comparaison des réponses de l'utilisateur avec les réponses attendues
$score = 0;
foreach ($reponsesAttendues as $key => $reponseAttendue) {
    if(isset($reponsesUtilisateur["reponse".$key+1])) {
        if(strtolower($reponsesUtilisateur["reponse".$key+1]) === strtolower($reponseAttendue)){
            $score++;
        }else{
            $score--;
        }
    }
}

// $score devient 0 si l'utilisateur a un score négatif
if($score < 0){
    $score = 0;
}
```

Figure 35 Code : traitement des réponses d'un quiz

4.7.3 Affichage des réponses

Pour afficher les réponses, le code stocke les données de « traitementQuizz.php » dans une variable session afin de pouvoir les réutiliser facilement dans la page de quizz.

```
// Ajout des données dans des variables sessions afin de les réutiliser dans d'autres pages
$_SESSION['score'] = $score;
$_SESSION['reponses_utilisateur'] = $reponsesUtilisateur;
header("Location: ../quizz.php?id=". $_POST['idQuizz'] . ".php");
exit;
```

Figure 36 PHP : stock résultat utilisateur

Pour résumer : quand l'utilisateur envoie une réponse, il est donc renvoyé vers « traitementQuizz.php » puis renvoyer instantanément vers le quizz où il était.

Cependant, la page quizz a été séparée en deux parties afin d'éviter d'éparpiller le code en recréant une nouvelle page. Une partie pour effectuer le quiz, et une partie pour afficher les réponses une fois le quiz envoyé

La question est affichée avec sa réponse, et la réponse de l'utilisateur également.

Le score est mis à jour dans la base de données au niveau de l'utilisateur

Un total est affiché en bas de page puis le nouveau score général de l'utilisateur.

```
// Vérification si l'utilisateur contient des réponse d'un quiz
if (isset($_SESSION['score']) && isset($_SESSION['reponses_utilisateur'])) {
    $reponsesUtilisateur = $_SESSION['reponses_utilisateur'];
    $score = $_SESSION['score'];
}

<form id="quizz">
    <?php
    $counter = 1;
    $NombreBonnesReponses = 0;
    // Affiche la question avec sa réponse puis la réponse inscrite par l'utilisateur
    foreach($quizz as $question){
        echo "<div class='question'>";
        echo "Question : <label>" . $question['queTexte'] . "</label>";
        echo " | Réponse correcte : <label>" . $question['repTexte'] . "</label>";

        // Vérifier si la réponse de l'utilisateur pour cette question existe
        if(isset($reponsesUtilisateur["reponse".$counter]) && str_replace(' ', '', $reponsesUtilisateur["reponse".$counter]) != "") {
            echo "<br><label> Votre réponse : " . $reponsesUtilisateur["reponse".$counter] . "</label>";
        } else {
            echo "<br><label>Aucune réponse</label>";
        }

        // Calcul du nombre de réponses juste sur le total de réponses
        if($question['repTexte'] === $reponsesUtilisateur["reponse".$counter]){
            $NombreBonnesReponses++;
        }
        echo "</div><br>";
        $counter++;
    }

    echo "Total : ".$NombreBonnesReponses."/".($counter-1). " <br>";
    if($score != 0){
        $db->addScore($score, $utilisateur[0]["idUtilisateurs"]);
    }
    echo "Votre nouveau score de classement général est de " . $utilisateur[0]["utiScore"]+$score . " points (+ " . $score . " points)";
}

</form>
```

Figure 37 PHP : Affichage réponses utilisateurs

4.8 Base de données : Création quiz

4.8.1 Requêtes préparées utilisées

Les 3 requêtes qui suivent permettent une nouvelle insertion dans la table quizz, réponses, questions afin de faire la création d'un nouveau quiz

Quiz :

```
/**
 * Permet de créer un quizz (uniquement pour la table t_quizz)
 * param $titre => Titre du quizz
 * param $idUser => id de l'utilisateur qui crée le quizz (clé étrangère)
 */
1 reference | 0 overrides
public function createQuiz($titre, $idUser){
    $query = ("INSERT INTO t_quizz (titre, fkUtilisateur) VALUES (:titre, :idUser)");
    $binds = [
        ["paramName" => "idUser", "paramValue" => $idUser, "paramType" => PDO::PARAM_INT],
        ["paramName" => "titre", "paramValue" => $titre, "paramType" => PDO::PARAM_STR]
    ];
    $statement = $this->queryPrepareExecute($query,$binds);
    return $this->formatData($statement);
}
```

Figure 38 SQL : insertion t_quizz

Question :

```
/**
 * Permet de créer un quizz (uniquement pour la table t_question)
 * param $texte => texte qui sera la question
 * param $idQuizz => id du quizz relia à cette question (clé étrangère)
 */
1 reference | 0 overrides
public function createQuestion($texte, $idQuizz){
    $query = ("INSERT INTO t_questions (questionTexte, fkQuizz) VALUES (:texte, :idQuizz)");
    $binds = [
        ["paramName" => "idQuizz", "paramValue" => $idQuizz, "paramType" => PDO::PARAM_INT],
        ["paramName" => "texte", "paramValue" => $texte, "paramType" => PDO::PARAM_STR]
    ];
    $statement = $this->queryPrepareExecute($query,$binds);
    return $this->formatData($statement);
}
```

Figure 39 SQL : insertion t_questions

Reponse :

```
/**
 * Permet de créer un quizz (uniquement pour la table t_reponse)
 * param $texte => texte de la réponse
 * param $idQuestion => id de la question relia à cette réponse (clé étrangère)
 */
1 reference | 0 overrides
public function createReponse($texte, $idQuestion){
    $query = ("INSERT INTO t_reponses (reponseTexte, fkQuestions) VALUES (:texte, :idQuestion)");
    $binds = [
        ["paramName" => "idQuestion", "paramValue" => $idQuestion, "paramType" => PDO::PARAM_INT],
        ["paramName" => "texte", "paramValue" => $texte, "paramType" => PDO::PARAM_STR]
    ];
    $statement = $this->queryPrepareExecute($query,$binds);
    return $this->formatData($statement);
}
```

Figure 40 SQL : insertion t_reponses

Cette requête permet d'obtenir le dernier ID créer dans la base de données afin de faire la création complète du quizz en faisant la gestion des clés étrangères.

```
/**
 * Permet d'obtenir l'id du dernier élément créer dans la base de donnée
 */
2 references | 0 overrides
public function getLastId(){
    $lastId = $this->querySimpleExecute("SELECT LAST_INSERT_ID() AS id;");
    return $this->formatData($lastId);
}
```

Figure 41 SQL : Obtenir dernier id créer dans la base de données

4.8.2 Traitement et création du quizz

Quand un administrateur souhaite ajouter un nouveau quizz au système il doit passer par la page « CreationQuizz.php ».

Un simple formulaire sera présent et enverra toutes les données à la page « traitementCreationQuizz.php »

La page de traitement va vérifier à chaque input de réponse/question :

- Si la question est associée à une réponse
- Si la réponse est associée à une question
- Si les deux inputs ne sont pas vides

```
// Boucle pour traiter toutes les entrées POST
foreach ($_POST as $key => $value) {
    // Vérifier si c'est une question
    if (strpos($key, 'question') === 0) {
        $questionNumber = substr($key, 8);

        // Vérifier si la question est vide, si oui -> question ajoutée | si non -> question supprimée du tableau
        if (empty(trim($value)) || !ctype_alnum($value)) {
            unset($questionsUtilisateur[$key]);
            unset($reponsesUtilisateurs['reponse' . $questionNumber]);
        } else {
            $questionsUtilisateur[$key] = $value;
        }
    }

    // Vérifier si c'est une réponse
    if (strpos($key, 'reponse') === 0) {
        $reponseNumber = substr($key, 7);

        // Vérifier si la réponse est vide, si oui -> question ajoutée | si non -> question supprimée du tableau
        if (!empty(trim($value)) || !ctype_alnum($value)) {
            $reponsesUtilisateurs[$key] = $value;
        } else {
            unset($questionsUtilisateur['question' . $reponseNumber]);
        }
    }
}

// Supprimer les réponses solitaires sans question associée
foreach ($reponsesUtilisateurs as $key => $value) {
    $reponseNumber = substr($key, 7);
    if (!isset($questionsUtilisateur['question' . $reponseNumber])) {
        unset($reponsesUtilisateurs[$key]);
    }
}
```

Figure 42 PHP : Traitement données création d'un quizz

Quand ce test est validé la question et la réponse sont ajoutés à un tableau séparément

Si le total de question/réponse est supérieur à 0, le quizz est créé

La création du quizz se compose de 3 parties distinctes :

- Insertion table t_quizz
 - Stock dans une variable l'id du nouvel élément
- Insertion table t_questions (boucle foreach pour insérer toutes les questions)
 - Stock dans une variable l'id du nouvel élément
- Insertions table t_reponses (boucle foreach pour insérer toutes les questions)
 - Stock dans une variable l'id du nouvel élément

```
// Vérifie s'il y a plus de 0 question/réponse
if(count($reponsesUtilisateurs) != 0){

    //Création quizz
    $db->createQuizz($nomQuizz,$idUtilisateur);
    $idQuizz = $db->getLastId();
    foreach($questionsUtilisateur as $key=>$value){

        //Création question
        $db->createQuestion($value,$idQuizz[0]['id']);
        $idQuestion = $db->getLastId();

        // Création reponse
        $key = str_replace('question','reponse',$key);
        $db->createReponse($reponsesUtilisateurs[$key], $idQuestion[0]['id']);
    }
}
```

Figure 43 PHP : Création d'un quizz

4.9 Base de données : Suppression quiz

4.9.1 Requêtes préparées utilisées

Afin de garantir une suppression complète, `beginTransaction()` permet de regrouper toutes les transactions afin de les considérer comme un groupe. Une fois toutes les opérations effectuées `$this->connector->commit()` sera effectué et donc la suppression sera effectuée.

Mais si un soucis est détecté dans une seule des transactions, `rollBack()` va permettre de tout annuler dans le groupe, comme si rien n'a été effectué depuis le début.

```
1 reference | 0 overrides
public function deleteQuizz($quizzId){
    try {
        $this->connector->beginTransaction();

        // Suppression reponses
        $query = "DELETE t_reponses FROM t_reponses
                JOIN t_questions ON t_reponses.fkQuestions = t_questions.idQuestions
                WHERE t_questions.fkQuizz = :quizzId";
        $binds = [ ["paramName" => "quizzId", "paramValue" => $quizzId, "paramType" => PDO::PARAM_INT]];
        $this->queryPrepareExecute($query, $binds);

        // Suppression questions
        $query = "DELETE FROM t_questions WHERE fkQuizz = :quizzId";
        $this->queryPrepareExecute($query, $binds);

        // Suppression Quizz
        $query = "DELETE FROM t_quizz WHERE idQuizz = :quizzId";
        $this->queryPrepareExecute($query, $binds);

        $this->connector->commit();
    } catch (Exception $e) {
        $this->connector->rollBack();
        throw $e;
    }
}
```

Figure 44 SQL : suppression quizz

Lorsque l'administrateur appuie sur le bouton « suppression quiz » d'un quizz, une fonction js est appelée. Cette fonction va permettre d'envoyer un message de confirmation à l'administrateur afin qu'il soit sûr de son action.

```
<?php if($utilisateur[0]['utiDroits'] == 'admin'){
    echo "<h6 class='card-subtitle'><a onclick='question(\".$quizz['idQuizz'].")' href='#'>Supprimer Quizz</a></h6>";
} ?>
```

Figure 45 PHP : renvoie vers suppression d'un quizz

```
function question(id){
    if (confirm('Etes-vous sûr de vouloir supprimer ce quizz ?')) {
        //Si oui
        window.location.replace("src/php/suppressionQuizz.php?id="+id+"");
    }
}
```

Figure 46 JS : message de confirmation de suppression

Quelques vérifications sont effectuées avant que le quizz ne soit supprimé :

- Si un id est présent dans l'URL
- Si l'utilisateur accédant à l'URL est connecté
- Si l'utilisateur accédant à l'URL est administrateur

```
//Si l'id est vide le site ne se charge pas
if(!isset($_GET['id']) || empty($_GET['id'])){
    header("Location: ../../index.php");
    exit;
}else{
    $idQuizz = $_GET['id'];
}

//Si l'utilisateur n'est pas connecté il est redirigé
if(!isset($_SESSION["login"])){
    header("Location: ../../index.php");
    exit;
}

// Si l'utilisateur n'est pas administrateur il est redirigé
if($_utilisateur[0]['utiDroits'] != 'admin'){
    header("Location: ../../index.php");
    exit;
}
```

Figure 47 PHP : Sécurité avant suppression quizz

Une dernière vérification est effectuée avant de supprimer le quizz, une requête va être effectuée afin de vérifier si le quizz existe, si oui il est supprimé, sinon il y a un message d'erreur

```
if($db->getQuizz($idQuizz)){
    echo "Quizz supprimer avec succès";
    $db->deleteQuizz($idQuizz);
    header("Location: ../../index.php");
    exit;
}else{
    ?><script>alert("Le quizz est introuvable");</script><?php
    header("Location: ../../index.php");
    exit;
}
```

Figure 48 PHP : Suppression Quizz

4.10 Base de données : Affichage classement

4.10.1 Requêtes préparées utilisées

Cette requête permet d'obtenir le classement de tous les joueurs, une variable est créée afin de déterminer sa position.

```
/**
 * Permet d'obtenir le classement des joueurs dans l'ordre
 */
1 reference | 0 overrides
public function getRanking(){
    // Exécute la première requête pour initialiser la variable @rank
    $this->querySimpleExecute("SET @rank = 0");

    $ranking = $this->querySimpleExecute("
        SELECT utiNomUtilisateur, utiNom, utiPrenom, utiScore, @rank := @rank + 1 AS position
        FROM t_utilisateurs
        ORDER BY utiScore DESC"
    );

    return $this->formatData($ranking);
}
```

Figure 49 SQL : Obtenir le classement général

4.10.2 Affichage du classement

Pour consulter le classement qui affiche les 7 joueurs ayant obtenus le plus de points, et visualiser sa propre position, l'utilisateur peut se rendre sur la page « classement.php »

Une simple variable contenant les utilisateurs dans l'ordre décroissant au niveau des points est créée puis est lue par une boucle foreach afin d'afficher toutes ses informations sous forme de tableau.

```
<?php
// Création du tableau
foreach ($ranking as $row) {
    // Stocke la position de l'utilisateur connecté dans une variable lorsque ses données seront scannées
    if ($utilisateur[0]['utiNomUtilisateur'] == $row['utiNomUtilisateur']){
        $positionUtilisateur = $row['position'];
    }
    // Permet l'affichage uniquement des 7 premiers joueurs
    if ($counter < 7){
        echo "<tr>";
        echo "<td>" . $row['position'] . ". " . $row['utiNom'] . "</td>";
        echo "<td>" . $row['utiPrenom'] . "</td>";
        echo "<td> | " . $row['utiScore'] . " Points" . "</td>";
        echo "</tr>";
        $counter++;
    }
}
```

Figure 50 PHP : Afficher les 7 premiers utilisateurs du classement

Un bout de code permet de tenir informer l'utilisateur de sa position et ses points

```
<tr>
    <!-- Ligne vide afin de séparer le score personnel et général-->
    <td></td>
    <td></td>
    <td class="invisibleTexte">.</td>
</tr>
<tr>
    <td>
        <?php
            echo("Votre score : " . $utilisateur[0]['utiScore']);
        ?>
    </td>
    <td>
    </td>
    <td>
        <?php echo("votre position : " . $positionUtilisateur); ?>
    </td>
</tr>
```

Figure 51 PHP : Affichage des points/position de l'utilisateur connecté

4.11 Base de données : Gestion des utilisateurs

4.11.1 Requêtes préparées utilisées

Cette requête permet d'obtenir toutes les informations de tous les utilisateurs

```
public function getAllUser(){
    $users = $this->querySimpleExecute("SELECT * FROM t_utilisateurs");
    return $this->formatData($users);
}
```

Figure 52 SQL : Obtenir tous les utilisateurs

Cette requête modifie les données de l'utilisateur en fonction de ce qu'il reçoit en paramètre

```
public function updateUser($username, $nom, $prenom, $score, $droit){
    $query = ("UPDATE `t_utilisateurs` SET `utiNom`=:nom, `utiPrenom`=:prenom, `utiDroits`=:droit, `utiScore`=:score WHERE `utiNomUtilisateur` = :username");
    $binds = [
        ["paramName" => "score", "paramValue" => $score, "paramType" => PDO::PARAM_INT],
        ["paramName" => "username", "paramValue" => $username, "paramType" => PDO::PARAM_STR],
        ["paramName" => "nom", "paramValue" => $nom, "paramType" => PDO::PARAM_STR],
        ["paramName" => "prenom", "paramValue" => $prenom, "paramType" => PDO::PARAM_STR],
        ["paramName" => "droit", "paramValue" => $droit, "paramType" => PDO::PARAM_STR]
    ];
    $statement = $this->queryPrepareExecute($query, $binds);
    return $this->formatData($statement);
}
```

Figure 53 SQL : Modifier un utilisateur

Cette requête l'utilisateur qu'il reçoit en paramètre

```
public function deleteUser($utilisateurId){
    try {
        // Démarrer la transaction
        $this->connector->beginTransaction();

        // Supprimer les réponses associées aux questions de l'utilisateur
        $query = "DELETE t_reponses FROM t_reponses
                JOIN t_questions ON t_reponses.fkQuestions = t_questions.idQuestions
                JOIN t_quizz ON t_questions.fkQuizz = t_quizz.idQuizz
                WHERE t_quizz.fkUtilisateurs = :utilisateurId";
        $binds = [ ['paramName' => 'utilisateurId', 'paramValue' => $utilisateurId, 'paramType' => PDO::PARAM_INT] ];
        $this->queryPrepareExecute($query, $binds);

        // Supprimer les questions associées aux quiz de l'utilisateur
        $query = "DELETE t_questions FROM t_questions
                JOIN t_quizz ON t_questions.fkQuizz = t_quizz.idQuizz
                WHERE t_quizz.fkUtilisateurs = :utilisateurId";
        $this->queryPrepareExecute($query, $binds);

        // Supprimer les quiz de l'utilisateur
        $query = "DELETE FROM t_quizz WHERE fkUtilisateurs = :utilisateurId";
        $this->queryPrepareExecute($query, $binds);

        // Supprimer l'utilisateur
        $query = "DELETE FROM t_utilisateurs WHERE idUtilisateurs = :utilisateurId";
        $this->queryPrepareExecute($query, $binds);

        // Valider la transaction
        $this->connector->commit();
    } catch (Exception $e) {
        // Annuler la transaction en cas d'erreur
        $this->connector->rollBack();
        throw $e;
    }
}
```

Figure 54 SQL : supprimer un utilisateur

4.11.2 Affichages de tous les utilisateurs

Pour afficher tous les utilisateurs sous forme de tableau, on récupère les informations et on les stocke dans une variable

```
$utilisateurs = $db->getAllUser();
```

Figure 55 PHP : Récupérer les infos des utilisateurs

Après les avoir stocker, on parcourt le tableau avec une boucle foreach afin de les afficher à l'écran de l'utilisateur

```
foreach($utilisateurs as $utilisateurSpecifique){
    echo "<tr>";
    echo "<th scope='row'>" . $utilisateurSpecifique['idUtilisateurs'] . "</th>";
    echo "<td>" . $utilisateurSpecifique['utiNom'] . "</td>";
    echo "<td>" . $utilisateurSpecifique['utiPrenom'] . "</td>";
    echo "<td>" . $utilisateurSpecifique['utiNomUtilisateur'] . "</td>";
    echo "<td>" . $utilisateurSpecifique['utiDroits'] . "</td>";
    echo "<td>" . $utilisateurSpecifique['utiScore'] . "</td>";
    echo "<td><a href='detailUtilisateur.php?id=".$utilisateurSpecifique['utiNomUtilisateur']."'><i class='bi bi-pen'></i></a>";
    if($utilisateurSpecifique['utiNomUtilisateur'] != $utilisateur[0]['utiNomUtilisateur']){
        echo "<a onclick='questionUtilisateur(\"".$utilisateurSpecifique['utiNomUtilisateur']."\")' href='#'><i class='bi bi-trash'></i></a>";
    }
    echo "</td>";
    echo "</tr>";
}
```

Figure 56 PHP : Afficher les infos des utilisateurs

4.11.3 Modification

Lors de l'affichage de utilisateurs on affiche un bouton sous forme d'icône de crayon qui renvoie à une page « detailUtilisateur.php » avec l'id de l'utilisateur en paramètre

Cette page récupère l'id de l'utilisateur et affiche les données de celui-ci sous forme de formulaire.

```
// Récupération des données de l'utilisateur
$utilisateurSpecifique = $db->connect($utilisateurSpecifique);
```

Figure 57 PHP : Récupération des données de l'utilisateur à modifier

Dans ce début de tableau on peut voir un exemple dans la manière que les données sont affichées, un input avec les données à l'intérieur afin de faciliter la modification

```
<div class="custom-width">
<h2 class="text-center">Detail -> <?php echo $utilisateurSpecifique[0]['utiNomUtilisateur'];> </h2>
<form action="#" method="POST">
    <div class="form-group">
        <label>Nom</label>
        <input type="text" class="form-control" id="nomInput" name="nom" value="<?php echo $utilisateurSpecifique[0]['utiNom'];>" />
        <div class="erreur"><p id="erreur_nom"></p></div>
    </div>
</form>
```

Figure 58 PHP : Formulaire de modification d'utilisateur

Lorsque les données sont complétées et le bouton submit est pressé, un code jQuery prend le relais en vérifiant les données de la même manière que l'inscription afin de vérifier si les données suivent les restrictions du site web.

Puis, sur la page « modificationUtilisateur.php » une requête updateUser est exécutée avec les nouvelles informations en paramètres, ce qui effectue la modification dans la base de données

```
// Vérifie si des données sont bien présentes
if (isset($score) && isset($nom) && isset($prenom) && isset($droit)){
    // Modification de l'utilisateur et redirection
    $db->updateUser($username, $nom, $prenom, $score, $droit);
    echo $redirection;
}else{
    echo $redirection;
}
```

Figure 59 PHP : Modification des données de l'utilisateur

4.11.4 Suppression

Lors de l'affichage de utilisateurs on affiche un bouton sous forme d'icône de poubelle crayon qui, lors du clique, un script javascript est appelé afin de demander une confirmation

```
function questionUtilisateur(login){
    if (confirm('Etes-vous sûr de vouloir supprimer cet utilisateur ('+login+') ?')) {
        //Si oui
        window.location.replace("src/php/suppressionUtilisateur.php?id="+login+"");
    }
}
```

Figure 60 JS : Confirmation de suppression d'utilisateur

Si la confirmation est effectuée, l'utilisateur est renvoyé vers la page suppressionUtilisateur avec l'id en paramètre URL.

La page récupère l'id passé en paramètre dans le lien afin de récupérer les informations de l'utilisateur et vérifier certaines sécurités.

```
//Si l'id est vide le site ne se charge pas
if(!isset($_GET['id']) || empty($_GET['id'])){
    header("Location: ../../index.php");
    exit;
}else{
    $NomUtilisateurUtilisateur = $_GET['id'];
}
```

Figure 61 PHP : Récupération de l'id dans l'URL

```
// Si l'utilisateur n'est pas administrateur il est redirigé
if($utilisateur[0]['utiDroits'] != 'admin'){
    header("Location: ../../index.php");
    exit;
}

// Si l'utilisateur tente de se supprimer il est redirigé
if($utilisateur[0]['utiNomUtilisateur'] == $NomUtilisateurUtilisateur){
    header("Location: ../../listeUtilisateurs.php");
    exit;
}

// Si l'utilisateur existe bien, il est supprimer
if($db->connect($NomUtilisateurUtilisateur)){
    $utilisateurSupprimer = $db->connect($NomUtilisateurUtilisateur);
    $db->deleteUser($utilisateurSupprimer[0]['idUtilisateurs']);
    echo "Utilisateur supprimer avec succès";
    header("Location: ../../listeUtilisateurs.php");
    exit;
}else{
    ?><script>alert("L'utilisateur est introuvable");</script><?php
    header("Location: ../../listeUtilisateurs.php");
    exit;
}
```

Figure 62 PHP : suppression de l'utilisateur

4.12 Cookies

Des cookies sont utilisés sur toutes les pages afin que l'utilisateur n'ait pas besoin de se reconnecter à chaque fois. La durée de ce cookie est d'une 365 jours.

```
<?php
//Insertion des cookies et démarrage d'une session php
ini_set('session.cookie_lifetime', 60 * 60 * 24 * 365);
ini_set('session.gc-maxlifetime', 60 * 60 * 24 * 365);
session_start();
?>
```

4.13 Base de données : résultat visuel (PHPMyAdmin)

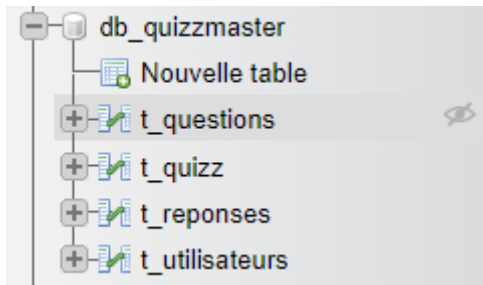


Figure 63 PhpMyAdmin : Arborescence

	idUtilisateurs	utiNomUtilisateur	utiMotDePasse	utiNom	utiPrenom	utiDroits	utiScore
<input type="checkbox"/> Éditer Copier Supprimer	8	fa1	b428fd6b33559ba36a54e17282ba3bbe4c86d1343f35dee76...	Fernandes	Alexandre	admin	33
<input type="checkbox"/> Éditer Copier Supprimer	9	jk1	b428fd6b33559ba36a54e17282ba3bbe4c86d1343f35dee76...	Jano	Kinan	utilisateur	20
<input type="checkbox"/> Éditer Copier Supprimer	10	gf1	b428fd6b33559ba36a54e17282ba3bbe4c86d1343f35dee76...	Gelsomino	Fabrice	utilisateur	15
<input type="checkbox"/> Éditer Copier Supprimer	11	iy1	b428fd6b33559ba36a54e17282ba3bbe4c86d1343f35dee76...	Imperadori	Yann	utilisateur	14
<input type="checkbox"/> Éditer Copier Supprimer	12	ak1	b428fd6b33559ba36a54e17282ba3bbe4c86d1343f35dee76...	Avdyaj	Kevin	utilisateur	0
<input type="checkbox"/> Éditer Copier Supprimer	13	ra1	b428fd6b33559ba36a54e17282ba3bbe4c86d1343f35dee76...	Roger	Antoine	utilisateur	0
<input type="checkbox"/> Éditer Copier Supprimer	14	pby1	b428fd6b33559ba36a54e17282ba3bbe4c86d1343f35dee76...	Pena-Bustabad	Yannick	utilisateur	0

Figure 64 PhpMyAdmin : Table utilisateur

	idQuizz	quiTitre	fkUtilisateurs
<input type="checkbox"/> Éditer Copier Supprimer	7	BMW	8
<input type="checkbox"/> Éditer Copier Supprimer	40	asd	8
<input type="checkbox"/> Éditer Copier Supprimer	41	test	8
<input type="checkbox"/> Éditer Copier Supprimer	42	as	8

Figure 65 PhpMyAdmin : Table quizz

























←T→	idQuestions	queTexte	fkQuizz
<input type="checkbox"/>  Éditer  Copier  Supprimer	15	En quelle année BMW a-t-elle été créée ?	7
<input type="checkbox"/>  Éditer  Copier  Supprimer	16	Qui est le fondateur de BMW ?	7
<input type="checkbox"/>  Éditer  Copier  Supprimer	21	Quel est le pays d'origine de BMW ?	7
<input type="checkbox"/>  Éditer  Copier  Supprimer	22	Quel est le slogan de BMW ?	7
<input type="checkbox"/>  Éditer  Copier  Supprimer	40	asd	40
<input type="checkbox"/>  Éditer  Copier  Supprimer	41	1	41
<input type="checkbox"/>  Éditer  Copier  Supprimer	42	1	41
<input type="checkbox"/>  Éditer  Copier  Supprimer	43	as	42

Figure 66 PhpMyAdmin : Table question

←T→	idReponses	repTexte	fkQuestions
<input type="checkbox"/>  Éditer  Copier  Supprimer	12	1916	15
<input type="checkbox"/>  Éditer  Copier  Supprimer	13	Karl Friedrich Rapp	16
<input type="checkbox"/>  Éditer  Copier  Supprimer	14	Allemagne	21
<input type="checkbox"/>  Éditer  Copier  Supprimer	15	Le plaisir de conduire	22
<input type="checkbox"/>  Éditer  Copier  Supprimer	21	asd	40
<input type="checkbox"/>  Éditer  Copier  Supprimer	22	2	41

Figure 67 PhpMyAdmin : Table réponse

4.14 Versions du projet

Toutes les versions du projet GitHub dans mon dépôt. Afin de consulter l'historique des versions depuis mon début il suffit de cliquer sur le bouton « Commits » de mon répertoire et de sélectionner celui voulu.

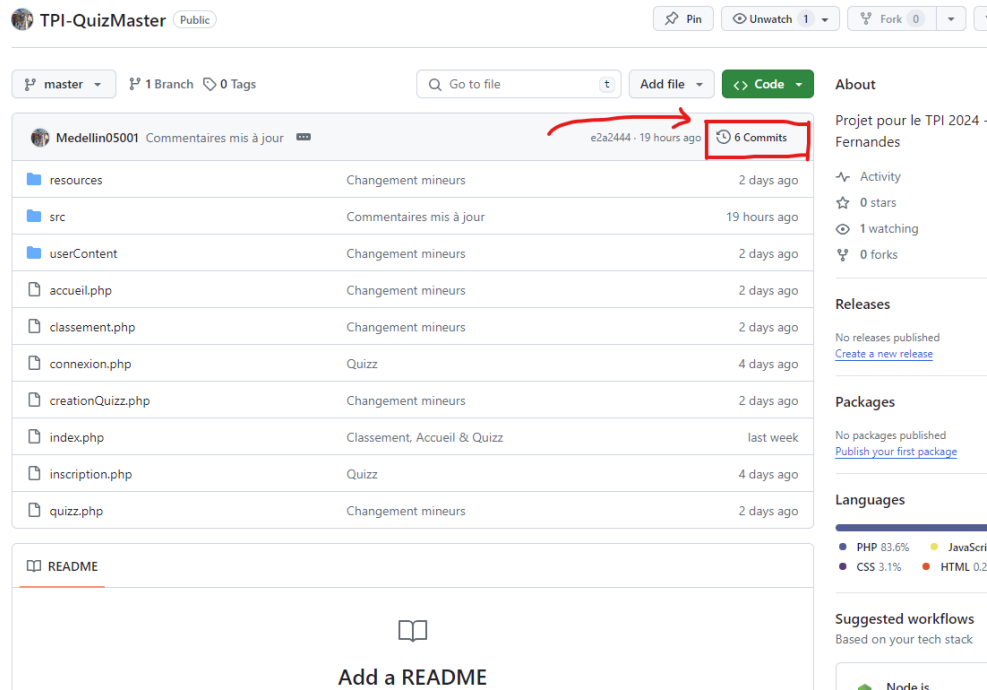


Figure 68 GitHub : Accès au commit



Figure 69 GitHub : Choix du commit

4.15 Description des tests effectués

Code couleur :

Non validé	Partiellement validé	Validé
-------------------	-----------------------------	---------------

Description catégorie :

Fonctionnalité	Retour	Résultat
Cette case décrira la fonctionnalité que l'on veut juger	Cette case démontrera ce qui est attendu de la fonctionnalité de manière résumé	Cette case contiendra la couleur indiquant le résultat du test

Fonctionnalité	Retour	Résultat
Page connexion <ul style="list-style-type: none"> Bouton connexion utilisateur 	Le bouton permet de rediriger l'utilisateur ayant inscrit les informations de compte correct sur la page d'accueil en étant connecté à son compte	
<ul style="list-style-type: none"> Bouton création compte 	Le bouton redirige correctement sur la page d'inscription afin que l'utilisateur puisse se créer un compte par la suite	
<ul style="list-style-type: none"> Retour d'erreur lors de la saisie d'information 	Quand l'utilisateur saisit une donnée pas accepté par le site, une erreur est retournée sous un petit texte rouge en dessous du champ en question. L'erreur retourne correctement ce qui ne convient pas	

Page création d'un compte <ul style="list-style-type: none"> Bouton création utilisateur 	Le bouton permet de rediriger l'utilisateur ayant inscrit des informations correctes sur la page de connexion, le compte voulu est créé également.	
<ul style="list-style-type: none"> Bouton connexion compte 	Le bouton redirige correctement sur la page	

	de connexion afin que l'utilisateur puisse se connecter par la suite	
<ul style="list-style-type: none"> Retour d'erreur lors de la saisie d'information 	Quand l'utilisateur saisit une donnée pas acceptée par le site, une erreur est retournée sous un petit texte rouge en dessous du champ en question. L'erreur retourne correctement ce qui ne convient pas	
Page d'accueil <ul style="list-style-type: none"> Affichage de tous les quizz (plus récents d'abord) 	Les quizz sont tous affichés sur la page d'accueil (plus récent au moins récent). Les informations tel que le titre et le nom/prénom du créateur sont bien affichés également	
<ul style="list-style-type: none"> Quizz cliquable 	Lors du clique sur un quizz, l'utilisateur est bien redirigé vers la page quizz.php?id=... Le lien contient bien l'id du quizz sur lequel l'utilisateur aurait cliqué	
<ul style="list-style-type: none"> Supprimer Quizz 	Lorsque l'utilisateur clique sur supprimer quizz, le quizz sélectionné est correctement supprimé	
Page Quizz (utilisation) <ul style="list-style-type: none"> Envoie du formulaire 	Le bouton soumettre permet d'envoyer les données saisies par l'utilisateur	
<ul style="list-style-type: none"> Affichage des réponses 	Le formulaire se transforme en « tableau de réponse » en affichant : La question la réponse correcte la réponse de l'utilisateur	
<ul style="list-style-type: none"> Calcul du score 	Le score de l'utilisateur est calculé de la manière suivante : réponse correcte = +1 réponse	

	incorrecte = -1 l'utilisateur ne peut pas avoir moins que 0.	
<ul style="list-style-type: none"> Affichage du résumer du score 	Un total est affiché (2/6) avec le nouveau score de classement de l'utilisateur	

Page création de quiz <ul style="list-style-type: none"> Bouton ajouter une question 	Le bouton ajouter de manière correct une paire d'input (question & réponse)	
<ul style="list-style-type: none"> Bouton création quiz 	Le bouton permet de créer un nouveau quizz à l'aide des informations ajoutées dans les différents champs, tout en prenant compte des inattendus qui ont été ajoutés	
<ul style="list-style-type: none"> Prise en compte des champs complets 	Une vérification est effectuée afin de supprimer toutes les paires incomplètes (question sans réponse associée et vice-versa) et les paires totalement vides.	
<ul style="list-style-type: none"> Création des quizz conformes aux règles 	Le code vérifie si le nom du quizz comporte maximum 50 caractères et contient uniquement lettre(accent)/chiffres. De même pour les questions/réponses mais avec maximum 255 caractères. Le quizz doit comporter minimum 1 question/réponse valable	

Page classement <ul style="list-style-type: none"> Affichage des 7 premiers utilisateurs 	Un tableau affichant les 7 meilleurs utilisateurs est affiché (Nom Prenom Points)	
<ul style="list-style-type: none"> Affichage score de l'utilisateur 	La page affiche le score de l'utilisateur et sa position par rapport aux autres joueurs	

Liste utilisateurs <ul style="list-style-type: none"> Affichage des utilisateurs 	Un tableau affiche les informations de tous les utilisateurs de la base de données. Cette page est uniquement accessible par les administrateurs	
<ul style="list-style-type: none"> Modification utilisateur 	Un bouton permet de modifier certaines données de l'utilisateur.	
<ul style="list-style-type: none"> Supprimer utilisateur 	Un bouton permet de supprimer un utilisateur sélectionner ainsi que tous les quizz qu'il a créés. L'administrateur ne peut pas se supprimer lui même	
Header <ul style="list-style-type: none"> Accéder aux différentes pages 	Lors du clique d'un élément sur le header, on est correctement redirigé vers la page en question (ex : classement)	
<ul style="list-style-type: none"> Information sur la page active 	Le texte de l'élément correspond à la page sur la quel nous somme devient blanc afin d'indiquer sur quel page l'utilisateur est (ex : Accueil)	
<ul style="list-style-type: none"> Déconnexion 	Le bouton texte déconnexion, déconnecte correctement l'utilisateur de sa session et le renvoie vers la page index	
Index <ul style="list-style-type: none"> Redirection de l'utilisateur 	La page redirige correctement l'utilisateur en fonction de son état (connecté ou non)	
Toutes les pages <ul style="list-style-type: none"> Responsive Design 	Toutes les pages sont adaptées pour accueillir mobile/tablette/ordinateur	

4.16 Test unitaire

Test Unitaire <ul style="list-style-type: none"> Get Ranking 	Le test unitaire créer 3 utilisateurs différents avec des score différents, le classement donné par le test unitaire est correct	
--	--	--

Ce test unitaire est séparé en 3 parties :

- Intervention manuelle de l'administrateur : afin que le test unitaire puisse être effectué, l'administrateur doit vider la table t_utilisateurs puis rendre public les fonctions querySimpleExecute et queryPrepareExecute du fichier database.php le temps du test unitaire
- Le code va insérer dans la base de données divers utilisateurs avec des score différents

```
private function setUpDatabase() {
    // Création d'utilisateurs fictifs
    $this->db->queryPrepareExecute(
        'INSERT INTO t_utilisateurs (utiNomUtilisateur, utiMotDePasse, utiNom, utiPrenom, utiScore) VALUES
        (:username1, :password1, :nom1, :prenom1, :score1),
        (:username2, :password2, :nom2, :prenom2, :score2),
        (:username3, :password3, :nom3, :prenom3, :score3)',
        [
            ['paramName' => 'username1', 'paramValue' => 'user1', 'paramType' => PDO::PARAM_STR],
            ['paramName' => 'password1', 'paramValue' => 'password1', 'paramType' => PDO::PARAM_STR],
            ['paramName' => 'nom1', 'paramValue' => 'Nom1', 'paramType' => PDO::PARAM_STR],
            ['paramName' => 'prenom1', 'paramValue' => 'Prenom1', 'paramType' => PDO::PARAM_STR],
            ['paramName' => 'score1', 'paramValue' => 300, 'paramType' => PDO::PARAM_INT],

            ['paramName' => 'username2', 'paramValue' => 'user2', 'paramType' => PDO::PARAM_STR],
            ['paramName' => 'password2', 'paramValue' => 'password2', 'paramType' => PDO::PARAM_STR],
            ['paramName' => 'nom2', 'paramValue' => 'Nom2', 'paramType' => PDO::PARAM_STR],
            ['paramName' => 'prenom2', 'paramValue' => 'Prenom2', 'paramType' => PDO::PARAM_STR],
            ['paramName' => 'score2', 'paramValue' => 200, 'paramType' => PDO::PARAM_INT],

            ['paramName' => 'username3', 'paramValue' => 'user3', 'paramType' => PDO::PARAM_STR],
            ['paramName' => 'password3', 'paramValue' => 'password3', 'paramType' => PDO::PARAM_STR],
            ['paramName' => 'nom3', 'paramValue' => 'Nom3', 'paramType' => PDO::PARAM_STR],
            ['paramName' => 'prenom3', 'paramValue' => 'Prenom3', 'paramType' => PDO::PARAM_STR],
            ['paramName' => 'score3', 'paramValue' => 100, 'paramType' => PDO::PARAM_INT],
        ]
    );
}
```

Figure 70 Test Unitaire : préparation test getRanking

- Le code va vérifier si la fonction getRanking a correctement sorti les utilisateurs dans l'ordre.

```
public function testGetRanking() {  
    $ranking = $this->db->getRanking();  
  
    $this->assertTrue(count($ranking) == 3, 'Le nombre de rangs doit être 3');  
    $this->assertTrue($ranking[0]['utiNomUtilisateur'] == 'user1', 'Le premier utilisateur doit être user1');  
    $this->assertTrue($ranking[0]['position'] == 1, 'Le rang du premier utilisateur doit être 1');  
    $this->assertTrue($ranking[1]['utiNomUtilisateur'] == 'user2', 'Le deuxième utilisateur doit être user2');  
    $this->assertTrue($ranking[1]['position'] == 2, 'Le rang du deuxième utilisateur doit être 2');  
    $this->assertTrue($ranking[2]['utiNomUtilisateur'] == 'user3', 'Le troisième utilisateur doit être user3');  
    $this->assertTrue($ranking[2]['position'] == 3, 'Le rang du troisième utilisateur doit être 3');  
  
    echo "All tests passed.\n";  
}  
  
7 references  
private function assertTrue($condition, $message = '') {  
    if (!$condition) {  
        echo "Test failed: " . $message . PHP_EOL;  
    } else {  
        echo "Test passed: " . $message . PHP_EOL;  
    }  
    echo "<br>";  
}
```

Figure 71 Test Unitaire: test getRanking

```
PS C:\Users\pz24gvr\Desktop\PHP\projets\TPI-QuizMaster\src\php> php databaseTest.php  
Test passed: Le nombre de rangs doit être 3  
<br>Test passed: Le premier utilisateur doit être user1  
<br>Test passed: Le rang du premier utilisateur doit être 1  
<br>Test passed: Le deuxième utilisateur doit être user2  
<br>Test passed: Le rang du deuxième utilisateur doit être 2  
<br>Test passed: Le troisième utilisateur doit être user3  
<br>Test passed: Le rang du troisième utilisateur doit être 3  
<br>All tests passed.
```

Figure 72 Test Unitaire : résultat

4.17 Erreurs restantes

À ma connaissance, le code ne comporte aucune erreur. Des tests ponctuels ont été réalisés lors de la mise en place de chaque fonctionnalité, ce qui m'a permis de maximiser les chances de détecter et corriger tout bug ou erreur éventuel.

4.18 Liste des documents fournis

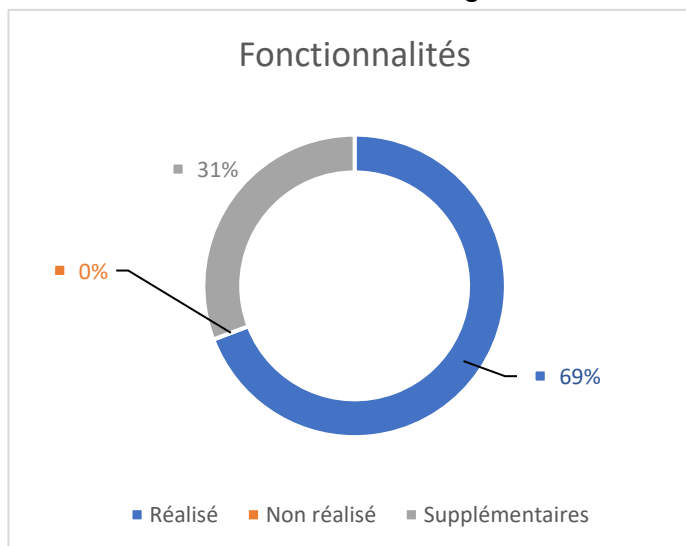
- Cahier des charges – 01.05.2024
- Planification initiale – 03.05.2024
- Journal de travail – 03.06.2024
- Maquettes de l'application – 03.06.2024
- Rapport du projet – 03.06.2024
- Abstract/résumé du projet – 03.06.2024
- Normes de codages ETML V.3.5.0 – 03.06.2024

- Code source de l'application – 03.06.2024
- Manuel de l'application - 03.06.2024

5 Conclusions

5.1 Bilan des fonctionnalités demandées

Toutes les fonctionnalités demandées dans le cahier des charges ont été implémentées et fonctionnent correctement. Cependant afin d'améliorer l'expérience utilisateur (administrateur) j'ai ajouté plusieurs fonctionnalités que je trouvais indispensable. Tel que la suppression de quizz et utilisateur, la modification d'information d'utilisateur et l'affichage de tous les utilisateurs. Ces fonctionnalités permettent grandement de faciliter la gestion du site et évite de devoir se perdre énormément de temps dans la base de données directement.



Si je devais améliorer cette application web, j'envisagerais de :

- Ajouter des types de questions
 - Exemple : choix entre plusieurs questions prédéfinies, choix multiple, etc...
- Modifications de quiz
 - Une fois qu'un quizz est créé il est actuellement impossible de le modifier, sauf si on le supprime et on le recrée

5.2 Bilan personnel

Durant ce projet, j'ai pu réaliser à quoi pourrait ressembler un projet venant d'un client.

J'ai rencontré des difficultés avec la liaison à jQuery et le responsive Design au niveau de la navbar. Faire face à ce problème a été une chance car j'ai pu apprendre à correctement structurer mon code. Les liens entre les différentes pages rentraient en conflit ce qui empêchait la page contenant mon jQuery et bootstrap de fonctionner.

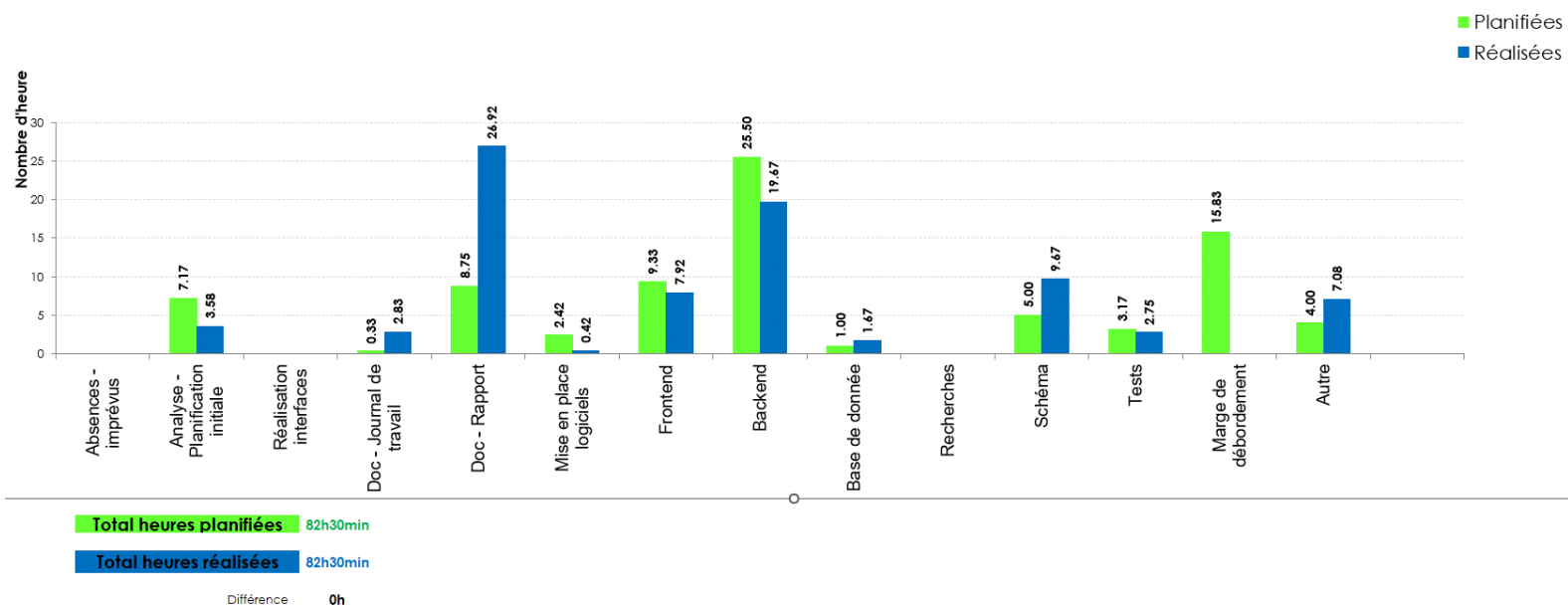
J'ai apprécié travailler sur ce projet, premièrement le développement web est le domaine que je préfère, cela a grandement continué à me motiver tout du long. Ensuite ce projet m'a permis de m'améliorer dans la façon dont je démarre un projet web au niveau de la structuration du code mais également d'apprendre à implémenter des sauvegardes Git dans mes projets.

En dernier point, le point négatif qui me concerne, ce serait la tenue du journal de travail. J'ai eu du mal à noter mes activités pendant que je les faisais ce qui, en les remplissant donc toutes en fin de journées, cela m'a empêché d'être précis. Cependant le projet m'a permis d'apprendre à le compléter de manière efficace et correcte.

5.3 Bilan de la planification

Durant le projet, je pense avoir relativement bien suivi la planification. Si la planification devait être refaire, j'aurais planifié plus de temps pour la documentation.

Comparaison entre les tâches planifiées et réalisées



6 Annexes

6.1 Résumé du projet QuizMaster

Situation de départ

Dans le cadre de ma formation d'informaticien à l'ETML visant l'obtention d'un CFC, j'ai développé une application web de gestion de quiz afin de valider les compétences acquises au cours de mes quatre années d'études. Ce projet consiste à créer une application web qui permet de consulter des quiz avec un système de classement en fonction des points obtenus. L'utilisateur pourra compléter un quizz et obtenir des points. Une interface administrateur permettra de gérer toutes les données.

Mise en œuvre

Pour mener à bien ce projet, j'ai adopté la méthodologie de travail dite « en cascade ». J'ai segmenté le projet en différentes étapes afin de le planifier de manière optimale : l'analyse du cahier des charges, la planification initiale, la conception de l'application web et des maquettes, le choix de MariaDB pour la base de données, la réalisation du MCD et du MLD, l'étude des besoins des utilisateurs, et enfin l'élaboration d'une stratégie de test. Pour le suivi de mon projet, j'ai utilisé un dépôt Git via l'application GitHub Desktop, me permettant de fournir aux experts des mises à jour régulières illustrant l'avancement de mon projet. J'ai également tenu un journal de travail à jour.

Pour finaliser, j'ai mené ce projet en utilisant Visual Studio Code connecté à GitHub pour la gestion des commits. Les tests ont été effectués directement dans mon navigateur web à l'aide de l'outil d'inspection d'éléments.

Résultat

L'application est entièrement opérationnelle ; toutes les fonctionnalités et les objectifs du cahier des charges ont été réalisés. Le design graphique de l'application web reste très basique et peut être amélioré. Pendant le développement, j'ai intégré un système de gestion des utilisateurs et des quiz, ajoutant ainsi une fonctionnalité significative.

Le répertoire Git :

<https://github.com/Medellin05001/TPI-QuizMaster>

6.2 Webographie

- <https://www.figma.com/> - Site internet utilisé pour la réalisation des maquettes graphiques.

- <https://canva.com> – Site internet utilisé pour la réalisation de logo graphique
- <https://photopea.com> – Site internet utilisé pour la réalisation de retouche graphique des maquettes
- <https://stackoverflow.com/> - Forum orienté développement de programme et code
- <https://www.developpez.com/> - Forum orienté développement de programme et code
- <https://getbootstrap.com/> - Page officiel Bootstrap, contient les informations nécessaires à son utilisation
- <https://chatgpt.com> – Intelligence artificiel
- <https://validator.w3.org/> - Permet d'évaluer la structure du code html

6.3 Table des illustrations

Figure 1 Modèle : en cascade	9
Figure 2 Maquette : Page d'accueil	10
Figure 3 Maquette : Page de connexion	11
Figure 4 Maquette : Page d'inscription	11
Figure 5 Maquette : Page de Classement	12
Figure 6 Maquette : Page de Création Quizz.....	12
Figure 7 Maquette : Page de Quizz	13
Figure 8 Maquette : Page Réponse de Quizz.....	13
Figure 9 Modèle conceptuel des données	14
Figure 10 Modèle logique des données.....	15
Figure 11 Graphique : Bilan de la planification initiale	17
Figure 12 Architecture : programme entier	23
Figure 13 Maquette : Utilisation possibles	24
Figure 14 Répertoire : Projet (1)	25
Figure 15 Répertoire : Projet (2)	25
Figure 16 Répertoire : Programme	26
Figure 17 SQL : Requêtes préparées.....	27
Figure 18 SQL : Connexion utilisateur.....	28
Figure 19 SQL : Création d'utilisateur.....	28
Figure 20 jQuery : Restriction Input connexion.....	29
Figure 21 Ajax : Renvoi de page connexion en POST	29
Figure 22 PHP : Traitement des données de connexion	30
Figure 23 jQuery: Restriction Input inscription	30
Figure 24 Ajax : Renvoi de page inscription en POST.....	31
Figure 25 PHP : Traitement des données d'inscription.....	32
Figure 26 PHP : déconnexion utilisateur	32
Figure 27 SQL : obtenir tous les quiz	33
Figure 28 Code : stocker tous les quiz	33
Figure 29 PHP : Affichage de tous les quiz	33
Figure 30 SQL : obtenir un quiz spécifique.....	34
Figure 31 PHP : accès à la page d'un quiz spécifique.....	34
Figure 32 PHP : stock les informations du quiz	34
Figure 33 PHP : Affichage du quiz.....	34
Figure 34 SQL : update le score de l'utilisateur	35
Figure 35 Code : traitement des réponses d'un quiz	35
Figure 36 PHP : stock résultat utilisateur.....	36
Figure 37 PHP : Affichage réponses utilisateurs	36
Figure 38 SQL : insertion t_quizz	37
Figure 39 SQL : insertion t_questions	37
Figure 40 SQL : insertion t_reponses	37
Figure 41 SQL : Obtenir dernier id créer dans la base de données.....	38
Figure 42 PHP : Traitement données création d'un quiz	38
Figure 43 PHP : Création d'un quizz	39
Figure 44 SQL : suppression quizz.....	40
Figure 45 PHP : renvoie vers suppression d'un quizz	40
Figure 46 JS : message de confirmation de suppression	40
Figure 47 PHP : Sécurité avant suppression quizz.....	41
Figure 48 PHP : Suppression Quizz	41

Figure 49 SQL : Obtenir le classement général.....	42
Figure 50 PHP : Afficher les 7 premiers utilisateurs du classement	42
Figure 51 PHP : Affichage des points/position de l'utilisateur connecté	43
Figure 52 SQL : Obtenir tous les utilisateurs	44
Figure 53 SQL : Modifier un utilisateur	44
Figure 54 SQL : supprimer un utilisateur	44
Figure 55 PHP : Récupérer les infos des utilisateurs	45
Figure 56 PHP : Afficher les infos des utilisateurs	45
Figure 57 PHP : Récupération des données de l'utilisateur à modifier	45
Figure 58 PHP : Formulaire de modification d'utilisateur	45
Figure 59 PHP : Modification des données de l'utilisateur.....	46
Figure 60 JS : Confirmation de suppression d'utilisateur.....	46
Figure 61 PHP : Récupération de l'id dans l'URL	46
Figure 62 PHP : suppression de l'utilisateur	47
Figure 63 PhpMyAdmin : Arborescence	48
Figure 64 PhpMyAdmin : Table utilisateur.....	48
Figure 65 PhpMyAdmin : Table quizz.....	48
Figure 66 PhpMyAdmin : Table question.....	49
Figure 67 PhpMyAdmin : Table réponse	49
Figure 68 GitHub : Accès au commit.....	50
Figure 69 GitHub : Choix du commit.....	50
Figure 70 Test Unitaire : préparation test getRanking	55
Figure 71 Test Unitaire: test getRanking	56
Figure 72 Test Unitaire : résultat	56
Figure 73 Manuel d'utilisation : inscription	84
Figure 74 Manuel d'utilisation : connexion.....	84
Figure 75 Manuel d'utilisation : header	85
Figure 76 Manuel d'utilisation : liste des quiz	85
Figure 77 Manuel d'utilisation : utilisation quiz.....	86
Figure 78 Manuel d'utilisation : réponse quiz.....	86
Figure 79 Manuel d'utilisation : création quiz.....	87
Figure 80 Manuel d'utilisation : consulter liste des utilisateurs	87
Figure 81 Manuel d'utilisation : Modification utilisateur.....	88
Figure 82 Manuel d'utilisation : Classement général	88

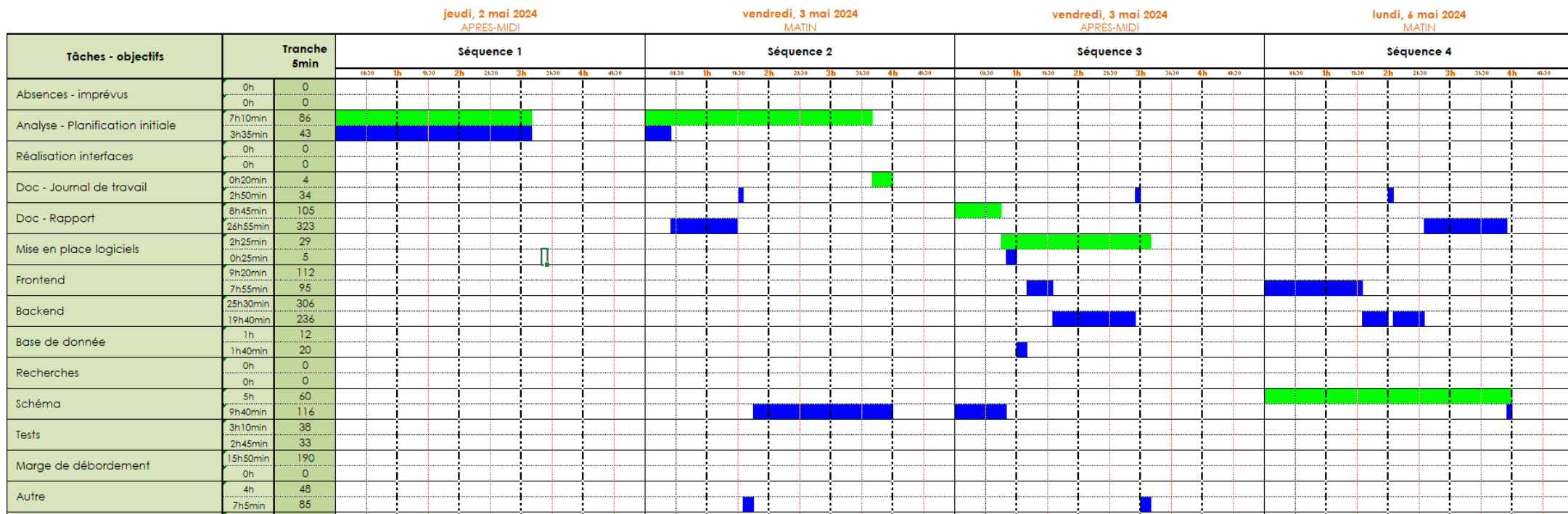
6.4 Dossier des tests

Dans le répertoire du projet, sous « 1. Documentation », vous pouvez accéder au sous-dossier intitulé « 1.4 Test_Vidéos ». Ce dernier contient des tests effectués sous forme de vidéos courtes ou de captures d'écran de l'application.

6.5 Archives du projet

Toutes les archives du projet se retrouvent sur GitHub, plus précisément sur le dépôt suivant : <https://github.com/Medellin05001/TPI-QuizMaster>

6.6 Planification – Journal de travail



			lundi, 6 mai 2024 APRÈS-MIDI								mercredi, 8 mai 2024 MATIN								mercredi, 8 mai 2024 APRÈS-MIDI								lundi, 13 mai 2024 MATIN										
Tâches - objectifs	Tranche 5min		Séquence 5								Séquence 6								Séquence 7								Séquence 8										
			0h30	1h	1h30	2h	2h30	3h	3h30	4h	4h30	0h30	1h	1h30	2h	2h30	3h	3h30	4h	4h30	0h30	1h	1h30	2h	2h30	3h	3h30	4h	4h30	0h30	1h	1h30	2h	2h30	3h	3h30	4h
Absences - imprévus	0h	0																																			
	0h	0																																			
Analyse - Planification initiale	7h10min	86																																			
	3h35min	43																																			
Réalisation interfaces	0h	0																																			
	0h	0																																			
Doc - Journal de travail	0h20min	4																																			
	2h50min	34																																			
Doc - Rapport	8h45min	105																																			
	26h55min	323																																			
Mise en place logiciels	2h25min	29																																			
	0h25min	5																																			
Frontend	9h20min	112																																			
	7h55min	95																																			
Backend	25h30min	306																																			
	19h40min	236																																			
Base de donnée	1h	12																																			
	1h40min	20																																			
Recherches	0h	0																																			
	0h	0																																			
Schéma	5h	60																																			
	9h40min	116																																			
Tests	3h10min	38																																			
	2h45min	33																																			
Marge de débordement	15h50min	190																																			
	0h	0																																			
Autre	4h	48																																			
	7h5min	85																																			

Auteur : Alexandre Fernandes
Modifié par : Alexandre Fernandes
Version: 911 du 03.06.2024 09:21

			vendredi, 17 mai 2024 MATIN							vendredi, 17 mai 2024 APRÈS-MIDI							mercredi, 22 mai 2024 MATIN							mercredi, 22 mai 2024 APRÈS-MIDI						
Tâches - objectifs	Tranche 5min		Séquence 13							Séquence 14							Séquence 15							Séquence 16						
			08:30	1h	08:30	2h	21:30	3h	31:30	4h	41:30	08:30	1h	08:30	2h	21:30	3h	31:30	4h	41:30	08:30	1h	08:30	2h	21:30	3h	31:30	4h	41:30	08:30
Absences - imprévus	0h	0																												
	0h	0																												
Analyse - Planification initiale	7h10min	86																												
	3h35min	43																												
Réalisation interfaces	0h	0																												
	0h	0																												
Doc - Journal de travail	0h20min	4																												
	2h50min	34																												
Doc - Rapport	8h45min	105																												
	26h55min	323																												
Mise en place logiciels	2h25min	29																												
	0h25min	5																												
Frontend	9h20min	112																												
	7h55min	95																												
Backend	25h30min	306																												
	19h40min	236																												
Base de donnée	1h	12																												
	1h40min	20																												
Recherches	0h	0																												
	0h	0																												
Schéma	5h	60																												
	9h40min	116																												
Tests	3h10min	38																												
	2h45min	33																												
Marge de débordement	15h50min	190																												
	0h	0																												
Autre	4h	48																												
	7h5min	85																												

			jeudi, 23 mai 2024 APRES-MIDI								vendredi, 24 mai 2024 MATIN								vendredi, 24 mai 2024 APRES-MIDI								mercredi, 29 mai 2024 MATIN												
Tâches - objectifs	Tranche 5min		Séquence 17								Séquence 18								Séquence 19								Séquence 20												
			06:30	1h	06:30	2h	2h30	3h	3h30	4h	4h30	06:30	1h	06:30	2h	2h30	3h	3h30	4h	4h30	06:30	1h	06:30	2h	2h30	3h	3h30	4h	4h30	06:30	1h	06:30	2h	2h30	3h	3h30	4h	4h30	
Absences - imprévus	0h	0																																					
	0h	0																																					
Analyse - Planification initiale	7h10min	86																																					
	3h35min	43																																					
Réalisation interfaces	0h	0																																					
	0h	0																																					
Doc - Journal de travail	0h20min	4																																					
	2h50min	34																																					
Doc - Rapport	8h45min	105																																					
	26h55min	323																																					
Mise en place logiciels	2h25min	29																																					
	0h25min	5																																					
Frontend	9h20min	112																																					
	7h55min	95																																					
Backend	25h30min	306																																					
	19h40min	236																																					
Base de donnée	1h	12																																					
	1h40min	20																																					
Recherches	0h	0																																					
	0h	0																																					
Schéma	5h	60																																					
	9h40min	116																																					
Tests	3h10min	38																																					
	2h45min	33																																					
Marge de débordement	15h50min	190																																					
	0h	0																																					
Autre	4h	48																																					
	7h5min	85																																					

			mercredi, 29 mai 2024 APRÈS-MIDI								jeudi, 30 mai 2024 APRÈS-MIDI								vendredi, 31 mai 2024 MATIN								vendredi, 31 mai 2024 APRÈS-MIDI											
Tâches - objectifs	Tranche 5min		Séquence 21								Séquence 22								Séquence 23								Séquence 24											
			06:30	1h	06:30	2h	2h30	3h	3h30	4h	4h30	06:30	1h	06:30	2h	2h30	3h	3h30	4h	4h30	06:30	1h	06:30	2h	2h30	3h	3h30	4h	4h30	06:30	1h	06:30	2h	2h30	3h	3h30	4h	4h30
Absences - imprévus	0h	0																																				
	0h	0																																				
Analyse - Planification initiale	7h10min	86																																				
	3h35min	43																																				
Réalisation interfaces	0h	0																																				
	0h	0																																				
Doc - Journal de travail	0h20min	4																																				
	2h50min	34																																				
Doc - Rapport	8h45min	105																																				
	26h55min	323																																				
Mise en place logiciels	2h25min	29																																				
	0h25min	5																																				
Frontend	9h20min	112																																				
	7h55min	95																																				
Backend	25h30min	306																																				
	19h40min	236																																				
Base de donnée	1h	12																																				
	1h40min	20																																				
Recherches	0h	0																																				
	0h	0																																				
Schéma	5h	60																																				
	9h40min	116																																				
Tests	3h10min	38																																				
	2h45min	33																																				
Marge de débordement	15h50min	190																																				
	0h	0																																				
Autre	4h	48																																				
	7h5min	85																																				

lundi, 3 juin 2024
MATIN

Tâches - objectifs	Tranche 5min	Séquence 25							
		06:30	1h	1h30	2h	2h30	3h	3h30	4h
Absences - imprévus	0h 0h	0							
Analyse - Planification initiale	7h10min 3h35min	86 43							
Réalisation interfaces	0h 0h	0 0							
Doc - Journal de travail	0h20min 2h50min	4 34							
Doc - Rapport	8h45min 26h55min	105 323							
Mise en place logiciels	2h25min 0h25min	29 5							
Frontend	9h20min 7h55min	112 95							
Backend	25h30min 19h40min	306 236							
Base de donnée	1h 1h40min	12 20							
Recherches	0h 0h	0 0							
Schéma	5h 9h40min	60 116							
Tests	3h10min 2h45min	38 33							
Marge de débordement	15h50min 0h	190 0							
Autre	4h 7h5min	48 85							

6.7 Journal de travail

Séquence 1			Date: jeudi, 2 mai 2024	APRÈS-MIDI
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Analyse - Planification initiale	15	1h15min	Discussion avec Expert pour l'analyse du Cahier des charges et mise en place des règles de communications	
Analyse - Planification initiale	23	1h55min	Effectuer la planification du TPI	
Total tranche	38	3h10min		
Séquence 2			Date: vendredi, 3 mai 2024	MATIN
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Analyse - Planification initiale	5	0h25min	Finalisation de la planification du TPI	
Doc - Rapport	13	1h5min	Commencer le rapport de projet en inscrivant les données de base tel que la description, les objectifs, la planification	
Doc - Journal de travail	1	0h5min	Mise à jour du journal de travail	
Autre	2	0h10min	Envoie du JDT par mail aux experts et chef de projet	
Schéma	12	1h	Schema MCD / MLD de la base de donnée	
Schéma	15	1h15min	Commencer la maquette des différentes pages web	
Total tranche	48	4h		

Séquence 3			Date: vendredi, 3 mai 2024		APRÈS-MIDI
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Schéma	10	0h50min	Finalisation de la maquette des différentes page web		
Mise en place logiciels	2	0h10min	Arborescence et mise en place des différents fichiers et dossiers		
Base de donnée	2	0h10min	Création de la base de donnée sur PHPMyAdmin	dossier TPI-QuizzMatser	
Frontend	5	0h25min	Création du formulaire de connexion, inscription et d'une première version d'un header(html&CSS) (aide de ChatGPT)	connexion.php, inscription.php, header.php	
Backend	16	1h20min	Debut de la mise en place du système d'authentification des utilisateurs	connexionSystem.php, inscriptionSystem.php, deconnexion.php, header.php	
Doc - Journal de travail	1	0h5min	Mise à jour du journal de travail		
Autre	1	0h5min	Mise en forme du journal de travail et rapport en PDF afin de l'envoyer aux experts et chef de projet		
Autre	1	0h5min	Sauvegarde du début de projet sur GitHub		
Total tranche	38	3h10min			
Séquence 4			Date: lundi, 6 mai 2024		MATIN
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Frontend	19	1h35min	Utilisation de JQuery afin de gérer d'erreurs en cas d'insertions de données incorrectes lors de la connexion (ex: Veuillez entrer un nom d'utilisateur, veuillez entrez un mot de passe)	Aide de Kinan Jano pour JQuery Reprise de bout de code d'anciens projets	
Backend	5	0h25min	Adapter le code qui permet la connexion afin qu'il reçoive les données jquery correctement, et qu'il puisse les traiter correctement afin de retourner des messages d'erreur en cohérence à la situation (Ex : "Utilisateur introuvable, "Le nom d'utilisateur et le mot de passe ne correspondent pas")		
Doc - Journal de travail	1	0h5min	Mise à jour du journal de travail		
Backend	6	0h30min	Effectuer une gestion d'erreur d'insertion pour la page d'inscription (minimum de caractère, maximum, uniquement lettre acceptées, rien n'a été introduit, etc...)		
Doc - Rapport	16	1h20min	Mise à jour du rapport		
Schéma	1	0h5min	Correction du modèle mcd et mld en fonction des règles ETML		
Total tranche	48	4h			

Séquence 5			Date: lundi, 6 mai 2024	APRÈS-MIDI
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Backend	14	1h10min	Commenter le code selon normes ETML	
Backend	4	0h20min	Compléter la page index, qui permet de rediriger l'utilisateur en fonction des problèmes rencontrés	
Backend	4	0h20min	Rendre le active header dynamique	https://www.developpez.net/forums/d325175/php/langage/obt
Backend	2	0h10min	Récupérer les 7 meilleurs joueurs et les afficher dans la page classement sous forme de tableau	
Frontend	4	0h20min	Commencer la mise en page du tableau de classement, affichage du numéro de positionnement par rapport au classement devant les noms	
Total tranche	28	2h20min		
Séquence 6			Date: mercredi, 8 mai 2024	MATIN
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Frontend	2	0h10min	Valider le code html	https://validator.w3.org/
Frontend	6	0h30min	Continuer la mise en page du tableau de classement, réduire la taille du tableau et le centrer verticalement	
Backend	8	0h40min	Changer le fonctionnement du tableau, utiliser la façon "SET @rank = 0" et "@rank := @rank+1 AS position" afin d'attribuer des positions fictives des joueurs dans la base de donnée. Cela permet de stocker les position et de les manier comme souhaiter	
Autre	5	0h25min	Création d'un logo et favicon + Application sur le header et onglet	Utilisation de Canva pour la création du logo
Autre	5	0h25min	Présentation du l'avancée du projet au chef de projet	
Doc - Rapport	4	0h20min	Avancée sur le rapport de projet	10H45
Frontend	2	0h10min	Création du visuel de la page d'accueil	
Backend	6	0h30min	Faire une boucle pour afficher tous les quizz sous forme de cartes	
Total tranche	38	3h10min		

Séquence 7			Date: mercredi, 8 mai 2024		APRÈS-MIDI
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Frontend	4	0h20min	Affichage des cartes de manière propre à l'aide de CSS et bootstrap		
Backend	6	0h30min	Préparer les requêtes et afin de récupérer les données du quizz		
Frontend	4	0h20min	Création d'un formulaire à l'aide de bootstrap et chatGPT afin d'entrer les réponses et afficher les questions du quizz		
Backend	2	0h10min	Parcourir le quizz afin d'afficher les questions à l'aide d'une boucle foreach		
Doc - Rapport	2	0h10min	Mise à jour rapport		
Doc - Journal de travail	1	0h5min	Mise à jour JDT		
Backend	16	1h20min	Récupérer les données inscrites de l'utilisateur et traiter les données afin de les comparer avec les réponses de la base de donnée	Aide de chatGPT	
Autre	3	0h15min	Effectuer un commit Git, préparer le jdt et rapport de projet en pdf puis les envoyés par mail aux expert puis chef de projet		
Total tranche	38	3h10min			
Séquence 8			Date: lundi, 13 mai 2024		MATIN
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Backend	16	1h20min	Renvois des données de l'utilisateur sur la page quizz afin de faire une mise en forme des questions réponses, reponses inscrites et score général		
Base de donnée	2	0h10min	Régler problème utf-8 sur la base de donnée	https://openclassrooms.com/forum/sujet/phpmyadmin-	
Base de donnée	1	0h5min	Agmentation de la taille allouer à la base de donnée suite à une erreur d'espace insuffisant	https://fr.siteground.com/kb/correction-de-lerreur-taille-de-la-	
Base de donnée	12	1h	Problème phpMyAdmin, connectivité réseau perdue	https://www.youtube.com/watch?v=tSdD0A9gwHg	
Backend	3	0h15min	Correction de redirection sur différentes pages php		
Frontend	4	0h20min	Création de la page créer quizz		
Backend	7	0h35min	Faire en sorte que quand j'appuie sur un bouton, des lignes de questions se créent		
Backend	3	0h15min	Commencer la page de traitement de la création de quizz (récupérer les informations)		
Total tranche	48	4h			

Séquence 9			Date: lundi, 13 mai 2024		APRÈS-MIDI
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Backend	5	0h25min	Traitement des données afin de savoir ce qui est correctement inscrit ou non, une question est toujours associé à une réponse et inversement, seuls les chiffres et lettres sont acceptés	Aide de chatGPT	
Backend	17	1h25min	Effectuer la requête SQL qui permet de créer le quizz		
Backend	4	0h20min	Ajustement de quelques détails sur la page de réponse aux quizz		
Doc - Journal de travail	2	0h10min	Effectuer le journal de travail et un commit Git		
Total tranche	28	2h20min			
Séquence 10			Date: mercredi, 15 mai 2024		MATIN
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Frontend	4	0h20min	Commenter le code		
Frontend	1	0h5min	Modifier la fonction pour ajouter des nouvelles question lors de la création de quizz afin qu'il soit en jquery et soit ajouter au document.ready		
Doc - Rapport	10	0h50min	Rapport - Ajout de certaines information et mise en page du texte et images		
Schéma	17	1h25min	Amélioration des maquettes pour qu'elles soient plus lisibles		
Autre	6	0h30min	Visite du chef de projet		
Total tranche	38	3h10min			

Séquence 11			Date: mercredi, 15 mai 2024		APRÈS-MIDI
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Doc - Rapport	36	3h	Rapport - Refaire tout le point 2. Analyse / Conception et commencer la 3.1 Liste des fichiers : code source		
Doc - Journal de travail	2	0h10min	Effectuer le journal de travail et un commit Git		
Total tranche	38	3h10min			

Séquence 12			Date: jeudi, 16 mai 2024		APRÈS-MIDI
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Doc - Journal de travail	2	0h10min	Effectuer le journal de travail et un commit Git		
Doc - Rapport	36	3h	Rapport commencer sur la partie base de donnée (3.4 à 3.10)		
Total tranche	38	3h10min			

Séquence 13			Date: vendredi, 17 mai 2024	MATIN
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Doc - Rapport	47	3h55min	Rapport - Refaire toute la partie 1. Analyse préliminaire et terminer la partie base de donnée (3.4 à 3.10)	
Doc - Journal de travail	1	0h5min	Journal de travail	
Total tranche	48	4h		

Séquence 14			Date: vendredi, 17 mai 2024	APRÈS-MIDI
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Doc - Rapport	4	0h20min	Rapport - Inscire la partie "versions du projet"	
Autre	3	0h15min	Entrevue avec le deuxième expert	
Doc - Journal de travail	2	0h10min	Journal de travail, détailler les avancées du rapport (sur les 3 derniers jours)	
Doc - Rapport	17	1h25min	Inscire la description des tests effectués	
Backend	12	1h	Correction bug pour le calcul du score, correction pour pour la création de quiz, commencer à rednre responsive mon header	
Total tranche	38	3h10min		

Séquence 15			Date: mercredi, 22 mai 2024		MATIN
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Tests	14	1h10min	Création d'un test unitaire pour la fonction getRanking	Aide de ChatGPT	
Doc - Rapport	12	1h	Adaptation du rapport en fonction de nouveautés ajoutées (test unitaires)		
Doc - Rapport	4	0h20min	Réeffectuer la mise en page et réécrire certaines phrases		
Schéma	4	0h20min	Réalisation maquette pour rapport (2.8.8 Utilisation schématisée)		
Backend	4	0h20min	Bug lors de la création de quizz, certaines questions ne sont pas prises en compte		
Total tranche	38	3h10min			
Séquence 16			Date: mercredi, 22 mai 2024		APRÈS-MIDI
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Backend	3	0h15min	Correction du bug du quizz, le regex n'autorisait pas le caractère suivant " ? " et " " "		
Backend	1	0h5min	Modifier la réception de tout les quizz sur la page d'accueil afin qu'on aie d'abord les quizz récents		
Backend	12	1h	Fonction pour supprimer un Quizz		
Tests	2	0h10min	Test de création de divers quizz et suppression		
Doc - Rapport	4	0h20min	Modification du rapport en conséquences des nouveautés : suppression quizz		
Doc - Rapport	6	0h30min	Préparation du rapport afin de lister tout les titres à remplir		
Frontend	8	0h40min	Responsive Design		
Doc - Journal de travail	2	0h10min	Journal de travail et commit git		
Total tranche	38	3h10min			

Séquence 17			Date: jeudi, 23 mai 2024	APRÈS-MIDI
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Frontend	18	1h30min	Refaire toute la structure du code, séparation du header afin d'avoir une partie head et une header (navbar). De même pour le footer, afin qu'il contiennent les liaisons js	J'ai remarquer que 2 footer et header étaient présent dans mon code (f12, inspecter l'élément)
Tests	6	0h30min	Tester tout le site afin de voir si la nouvelle structure n'a pas impacter le programme	
Backend	2	0h10min	Nouvelles autorisations dans la création de quizz : le caractère , " sont maintenant autorisés	
Frontend	8	0h40min	Responsive Design : header qui se réduit et se met à droite lors de l'utilisation du téléphone	
Frontend	2	0h10min	Responsive Design : Page quizz/réponse quizz qui sont maintenant lisible proprement lors de l'utilisation du téléphone	
Doc - Journal de travail	2	0h10min	Commit Git et journal de travail	
Total tranche	38	3h10min		

Séquence 18			Date: vendredi, 24 mai 2024	MATIN
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Autre	3	0h15min	Visite du chef de projet	
Doc - Rapport	18	1h30min	Rapport : Finalisation de point réalisation et début de la conclusion	
Frontend	4	0h20min	Mettre en place une interface liste des utilisateurs uniquement accessibles par les administrateurs	
Backend	12	1h	Afficher les informations de tout les utilisateurs dans l'interface avec deux bouton cliquables (modifier, supprimer	
Backend	11	0h55min	Ajout de la fonctionnalité, supprimer utilisateur	
Total tranche	48	4h		

Séquence 19			Date: vendredi, 24 mai 2024		APRÈS-MIDI
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Backend	30	2h30min	Ajout de la fonctionnalité modifier utilisateur		
Doc - Rapport	3	0h15min	Rapport : Mise à jour du rapport en conséquences des nouveautés (affichage utilisateurs, modifications, suppression)		
Tests	3	0h15min	Test des nouvelles fonctionnalités		
Doc - Journal de travail	2	0h10min	Journal de travail et commit Git		
Total tranche	38	3h10min			

Séquence 20			Date: mercredi, 29 mai 2024		MATIN
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Autre	3	0h15min	Discussion avec le chef de projet		
Doc - Rapport	34	2h50min	Avancer sur la conclusion et commencer les annexes		
Doc - Journal de travail	1	0h5min	Compléter le journal de travail		
Total tranche	38	3h10min			

Séquence 21			Date: mercredi, 29 mai 2024		APRÈS-MIDI
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Schéma	36	3h	effectuer des maquettes pour le manuel d'utilisation		
Doc - Journal de travail	10	0h50min	Compléter le journal de travail et commit Git		
Total tranche	46	3h50min			
Séquence 22			Date: jeudi, 30 mai 2024		APRÈS-MIDI
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Schéma	9	0h45min	Finalisation des maquettes du manuel d'utilisation		
Autre	9	0h45min	Préparer le dossier d'installation et push sur git		
Backend	5	0h25min	Corriger de léger bugs sur le code		
Autre	10	0h50min	Réaliser des vidéos pour le dossier de test		
Doc - Rapport	4	0h20min	Mettre à jour le rapport avec la manuel d'utilisation, d'installation et dossier de test		
Doc - Journal de travail	1	0h5min	Mettre à jour le journal de travail		
Total tranche	38	3h10min			

Séquence 23			Date: vendredi, 31 mai 2024		MATIN
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Tests	8	0h40min	Création de différents quizz et utilisateurs afin de tester le programme dans une situation cohérente		
Mise en place logiciels	3	0h15min	Créer une arborescence pour contenir tout les fichiers du tpi		
Base de donnée	3	0h15min	Export de la base de donnée et corrections légères du code		
Autre	5	0h25min	Création d'un readMe sur GitHub et modification du manuel d'installation		
Schéma	3	0h15min	Réalisation d'une page de garde sur photopea		
Schéma	9	0h45min	Réalisation d'une maquette pour l'arborescence des fichiers du projet sur canva		
Doc - Rapport	17	1h25min	Rapport : Glossaire et ajout des différentes maquettes créées, ajout de chapitre dans la réalisation		
Doc - Journal de travail	1	0h5min	Compléter le journal de travail		
Total tranche	49	4h5min			
Séquence 24			Date: vendredi, 31 mai 2024		APRÈS-MIDI
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Doc - Rapport	6	0h30min	Corrections des erreurs d'orthographe et grammaire		
Doc - Rapport	6	0h30min	Vérification des points d'évaluations		
Doc - Rapport	12	1h	Relecture entière du JDT		
Doc - Rapport	12	1h	Ajout des screen du journal de travail		
Doc - Journal de travail	2	0h10min	Compléter journal de travail, commit Git, envoi aux experts le JDT et rapport		
Total tranche	38	3h10min			

Séquence 25			Date: lundi, 3 juin 2024	MATIN
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Autre	29	2h25min	Imprimer le rapport et l'envoyer, envoi des différents documents électroniquement, revue du rapport et projet et modifications si nécessaires, revue des critères d'évaluation pour le rapport	
Total tranche	29	2h25min		

6.8 Manuel d'Installation

Décompressez le dossier se trouvant dans « 3. Projet », ouvrez le fichier START pour démarrer les différents serveurs.

Pour importer les données, il suffit d'accéder à la page PhpmyAdmin (localhost :8080) (user : root, password : root)

Cliquez en haut à gauche sur « Nouvelle base de donnée » puis dans l'onglet importer choisissez le fichier .sql qui se trouve dans le dossier de code puis « PHP/Projets/TPI-QuizMaster/userContent »

L'adresse du projet d'après la configuration devrait être : « localhost :80 »

Login admin du projet :

- Utilisateur : Admin
- Mot de passe : Test1234

6.9 Manuel d'Utilisation

6.9.1 S'inscrire



Figure 73 Manuel d'utilisation : inscription

6.9.2 Se connecter

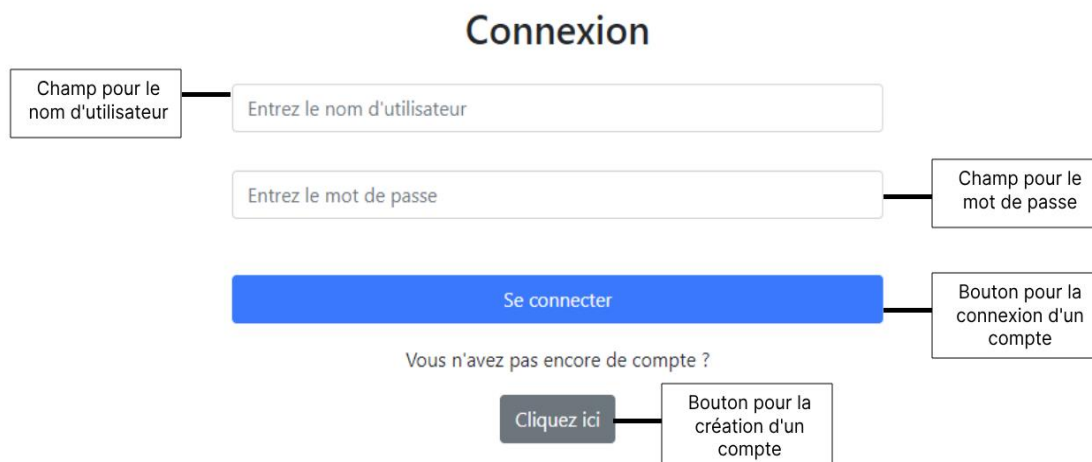


Figure 74 Manuel d'utilisation : connexion

6.9.3 Naviguer entre les différentes pages

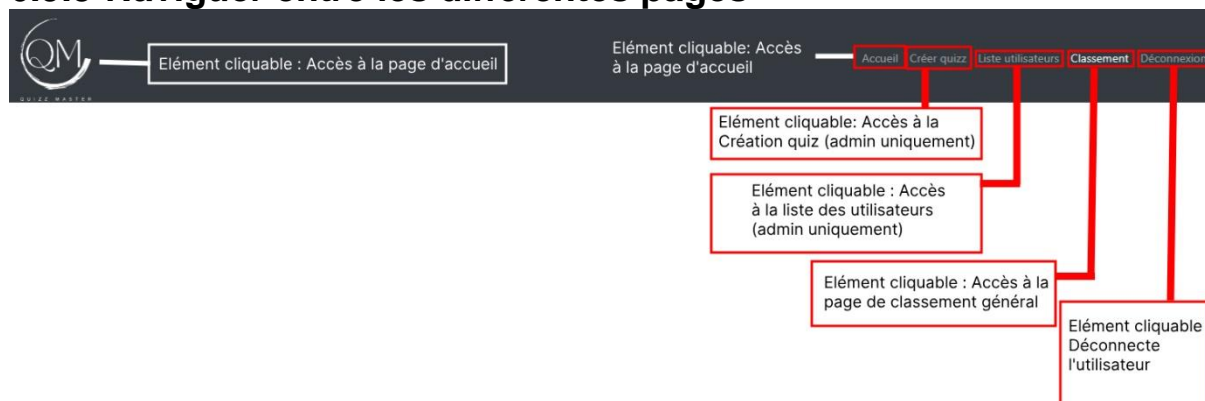


Figure 75 Manuel d'utilisation : header

6.9.4 Utilisation de quiz

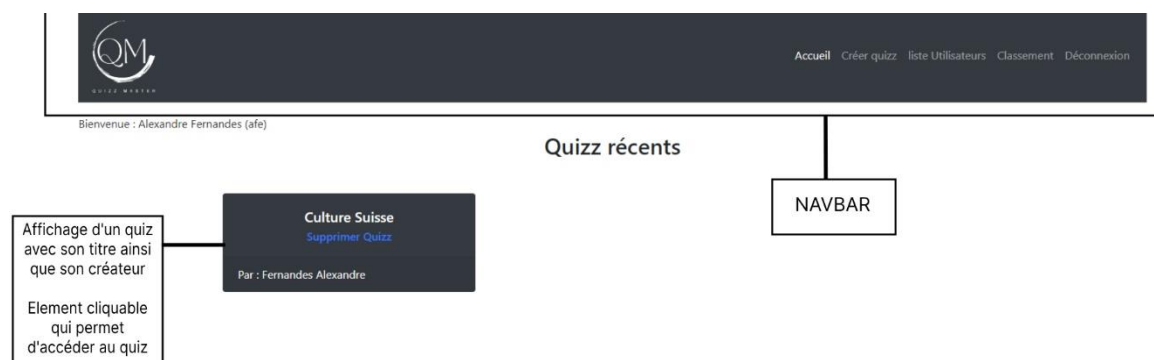
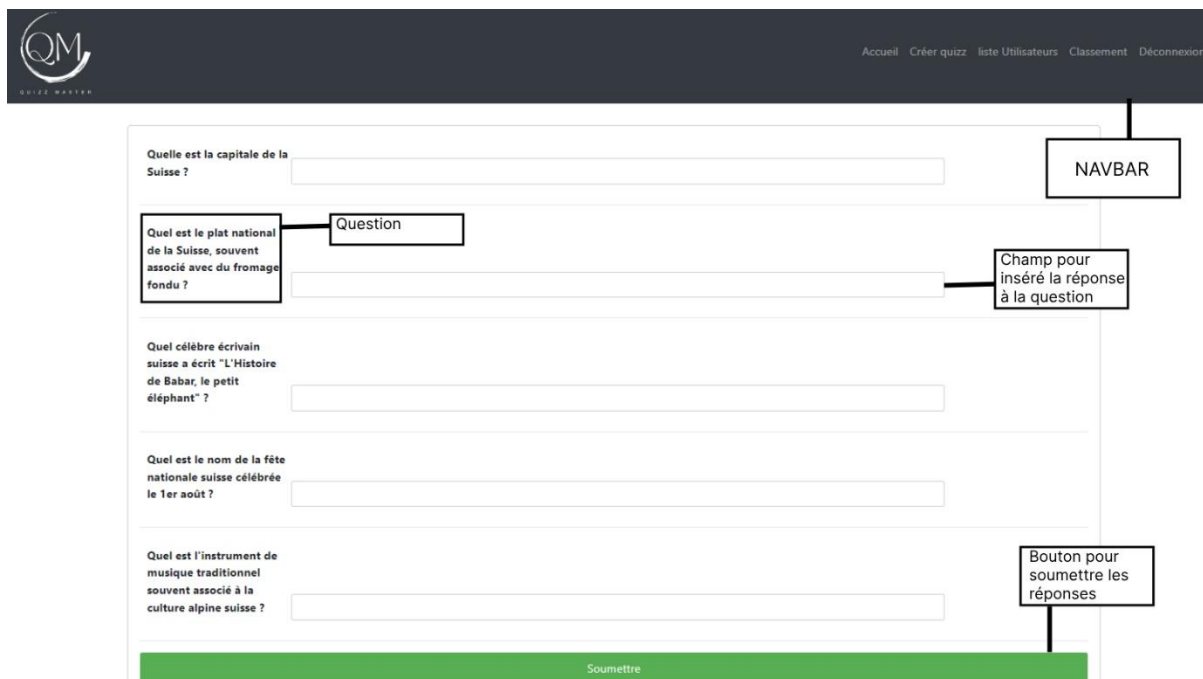


Figure 76 Manuel d'utilisation : liste des quiz



Quel est la capitale de la Suisse ?

Quel est le plat national de la Suisse, souvent associé avec du fromage fondu ?

Quel célèbre écrivain suisse a écrit "L'Histoire de Babar, le petit éléphant" ?

Quel est le nom de la fête nationale suisse célébrée le 1er août ?

Quel est l'instrument de musique traditionnel souvent associé à la culture alpine suisse ?

Soumettre

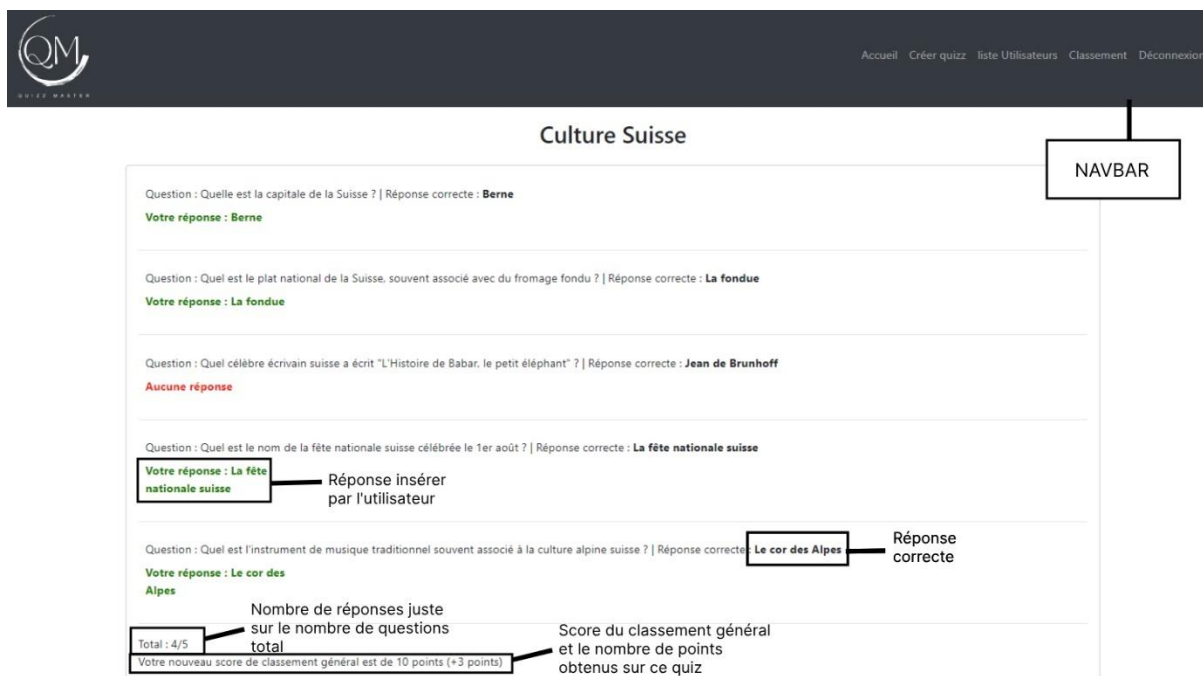
NAVBAR

Question

Champ pour insérer la réponse à la question

Bouton pour soumettre les réponses

Figure 77 Manuel d'utilisation : utilisation quiz



Culture Suisse

Question : Quelle est la capitale de la Suisse ? | Réponse correcte : **Berne**
Votre réponse : **Berne**

Question : Quel est le plat national de la Suisse, souvent associé avec du fromage fondu ? | Réponse correcte : **La fondue**
Votre réponse : **La fondue**

Question : Quel célèbre écrivain suisse a écrit "L'Histoire de Babar, le petit éléphant" ? | Réponse correcte : **Jean de Brunhoff**
Aucune réponse

Question : Quel est le nom de la fête nationale suisse célébrée le 1er août ? | Réponse correcte : **La fête nationale suisse**
Votre réponse : **La fête nationale suisse**

Question : Quel est l'instrument de musique traditionnel souvent associé à la culture alpine suisse ? | Réponse correcte : **Le cor des Alpes**
Votre réponse : **Le cor des Alpes**

Total : 4/5
Nombre de réponses juste sur le nombre de questions total
Score du classement général et le nombre de points obtenus sur ce quiz
Votre nouveau score de classement général est de 10 points (+3 points)

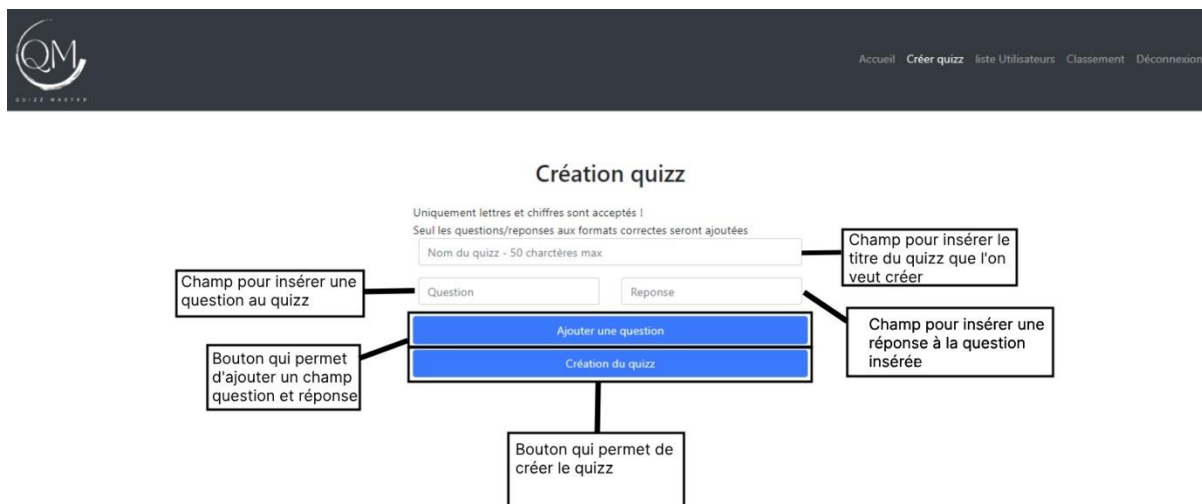
NAVBAR

Réponse insérer par l'utilisateur

Réponse correcte

Figure 78 Manuel d'utilisation : réponse quiz

6.9.5 Création d'un quizz (administrateur)



Création quizz

Uniquement lettres et chiffres sont acceptés |
Seul les questions/reponses aux formats correctes seront ajoutées

Nom du quizz - 50 caractères max

Question

Reponse

Ajouter une question

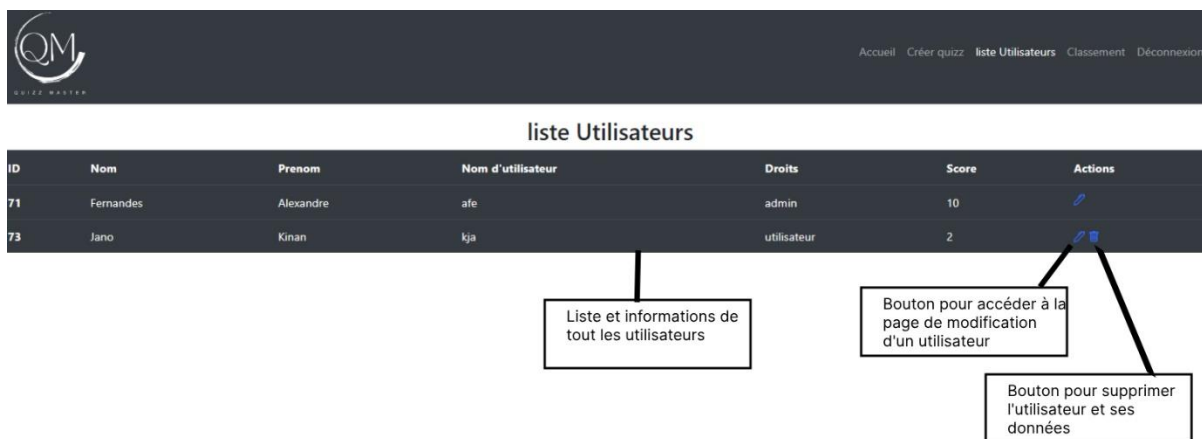
Création du quizz

Annotations :

- Champ pour insérer une question au quizz
- Bouton qui permet d'ajouter un champ question et réponse
- Champ pour insérer le titre du quizz que l'on veut créer
- Champ pour insérer une réponse à la question insérée
- Bouton qui permet de créer le quizz

Figure 79 Manuel d'utilisation : création quizz

6.9.6 Gestion des utilisateurs (administrateur)



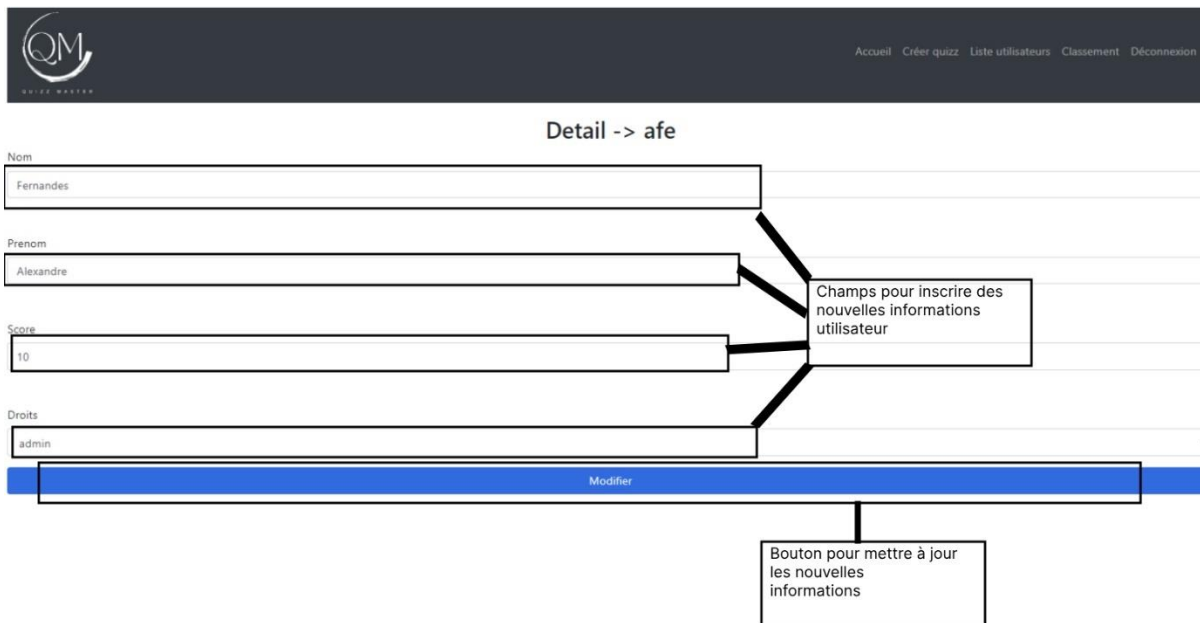
liste Utilisateurs

ID	Nom	Prenom	Nom d'utilisateur	Droits	Score	Actions
71	Fernandes	Alexandre	afe	admin	10	
73	Jano	Kinan	kja	utilisateur	2	

Annotations :

- Liste et informations de tout les utilisateurs
- Bouton pour accéder à la page de modification d'un utilisateur
- Bouton pour supprimer l'utilisateur et ses données

Figure 80 Manuel d'utilisation : consulter liste des utilisateurs



Detail -> afe

Nom
Fernandes

Prenom
Alexandre

Score
10

Droits
admin

Modifier

Champs pour inscrire des nouvelles informations utilisateur

Bouton pour mettre à jour les nouvelles informations

Figure 81 Manuel d'utilisation : Modification utilisateur

6.9.7 Consulter le classement général et son score personnel



Classement

Classement général.
7 premiers utilisateurs
sont affichés uniquement

1. Fernandes	Alexandre	10 Points
2. Jano	Kinan	2 Points
Votre score : 10		votre position : 1

Informations de l'utilisateur connecté

Figure 82 Manuel d'utilisation : Classement général