



Diplôme Universitaire de Technologie

Informatique

Tuniv

RAPPORT DE PROJET

Gaël Journet Jean-François Marcourt Nathan Ozimek

Promotion 2022/2023

REMERCIEMENTS

Nous adressons des remerciements à Lionel Buathier et Adrien Peytavie pour leur encadrement tout au long du projet, ainsi qu'à Émilien Nicolas et Benjamin Chazelle pour leur aide technique.

TABLE DES MATIERES

I.	Introduction.....	0
I.1	Objectifs et contexte du projet	0
I.2	Présentation de l'Equipe	0
II.	Réalisation du Projet	0
II.1	Présentation générale du projet	0
II.2	Architecture.....	3
II.3	Architecture routeur	4
II.4	Protection de la connexion utilisateur	4
III.	Résultats	5
IV.	Conclusion	7

I. INTRODUCTION

I.1 OBJECTIFS ET CONTEXTE DU PROJET

Le projet faisait suite à notre projet de SAÉ du premier semestre, Tuniv, qui visait à développer un site web de gestion de tournois de sport universitaire où des administrateurs pouvaient créer des tournois et y assigner des équipes pour que les arbres de tournois soient ensuite générés automatiquement. Des arbitres étaient assignés aux différents matchs et pouvaient entrer et mettre à jour les résultats en temps réel. Chaque match et chaque tournoi restait accessible via une base de données stockant tous les résultats des matchs actuels comme passés.

Notre projet du second semestre visait quant à lui à améliorer le projet que nous avons produit à l'issue du premier semestre, en se focalisant particulièrement sur la sécurité, l'ergonomie, la qualité logicielle et l'accessibilité, le tout en mettant en pratique les nouveaux savoirs et compétences acquis au cours de ce deuxième semestre.

Ce rapport détaillera la réalisation de ce projet, en présentant son architecture, les différents modules mis en place pour atteindre les objectifs d'amélioration fixés, ainsi que le résultat final, sans omettre de mentionner les améliorations possibles et les fonctionnalités prévues mais finalement non développées.

I.2 PRESENTATION DE L'EQUIPE

L'équipe est composée de Gaël Journet, Jean-François Marcourt et Nathan Ozimek.

Gaël Journet a travaillé principalement en tant que développeur Backend principalement en charge de l'amélioration de la sécurité et de l'ergonomie.

Jean-François Marcourt a travaillé en tant que Project-Leader et fut principalement en charge de l'ensemble des éléments visuels autour du projet (UI/UX/Présentations), ainsi que des tâches liées à l'optimisation de la sécurité logiciel.

Nathan Ozimek a travaillé en tant que développeur Backend principalement en charge de l'amélioration de la sécurité et de l'ergonomie.

Malgré des rôles bien définis, l'organisation mise en place autour du projet a permis à chacun des membres du groupe de contribuer à l'ensemble des tâches effectuées.

II. REALISATION DU PROJET

II.1 PRESENTATION GENERALE DU PROJET

Les améliorations apportées tout au long du projet relèvent de quatre axes principaux : la sécurité, l'ergonomie, la qualité logicielle et l'accessibilité.

En termes de sécurité, le changement le plus impactant a été le passage d'une architecture de site web « classique » à une architecture routeur, comme vu dans le module d'architecture logicielle : le serveur

est désormais lancé sur une page spécifique qui contient des routes définies manuellement vers chaque page. D'autres changements ont été mis en place, comme le chiffrement des mots de passe utilisateurs dans la base de données et des protections contre des attaques classiques comme les failles XSS ou le forçage des mots de passe, à l'aide notamment d'un système de captcha.

Mot_de_passe
\$2y\$10\$QyfpEgT5aLaZGF/jzhtGqugt6cuyIAAesw4XHfHlTD42bn9rf0f16
\$2y\$10\$qQvjgzST6PNXSrqYHsuTO4G6CIRWpTlFASgMentgdLnO/C0Uowe
\$2y\$10\$g/A1MLjeQ8hF3L6Pa.CnOOV7TDvJbSRvQva5pI2.ExB3WBA3bsQ9a
\$2y\$10\$Zfx.lK0BLJ6Bd7QVe8SPaeWY7JkePyRmDDAPjnvMd3Vo16XY.i96

En termes d'ergonomie, des changements ont été mis en place au niveau de la gestion des tournois, avec l'automatisation de l'avancement des tournois lorsque tous les matchs existants étaient clôturés, la possibilité pour un administrateur de modifier le score d'un match même après qu'il ait été clôturé ainsi qu'une révision graphique modeste.

En termes de qualité logicielle, le passage à l'architecture routeur mentionné précédemment a permis de centraliser certaines parties du code dans la page de routage, notamment la connexion à la base de données ; un repassage a également été effectué sur tous les fichiers afin de retirer les fichiers devenus inutiles, de passer du CSS au SCSS pour l'affichage et de faire un nettoyage complet du code afin de le rendre plus lisible et moins brouillon.

```

$res = $statement->fetch();
if ($res[0] == 1) { // On vérifie qu'il existe un utilisateur avec l'identifiant donné
    $statement = $pdo->prepare("SELECT Mot_de_passe FROM Utilisateurs WHERE Identifiant=:varLogin");
    $statement->execute(['varLogin' => "$login"]);
    $res = $statement->fetch();

    if (password_verify($password,$res[0])) { // Si oui, on vérifie que le mot de passe donné correspond à celui de l'utilisateur
        $statement = $pdo->prepare("SELECT Type_user FROM Utilisateurs WHERE Identifiant=:varLogin");
        $statement->execute(['varLogin' => "$login"]);

        // attribution du role
        $type = $statement->fetch()[0];
        if ($type == 0) {
            $_SESSION["loggedIn"] = true;
            $statement = $pdo->prepare("SELECT ID_User FROM Utilisateurs WHERE Identifiant=:varLogin AND Mot_de_passe=:varPassword");
            $statement->execute(['varLogin' => "$login", 'varPassword' => "$password"]);
            $res = $statement->fetch();
            $_SESSION["userId"] = $res[0];
            $_SESSION["type"] = "administrateur";
            header("Location: /index");
        }
    }
}

```

855fb0522 1 branch 0 tags Go to file Code

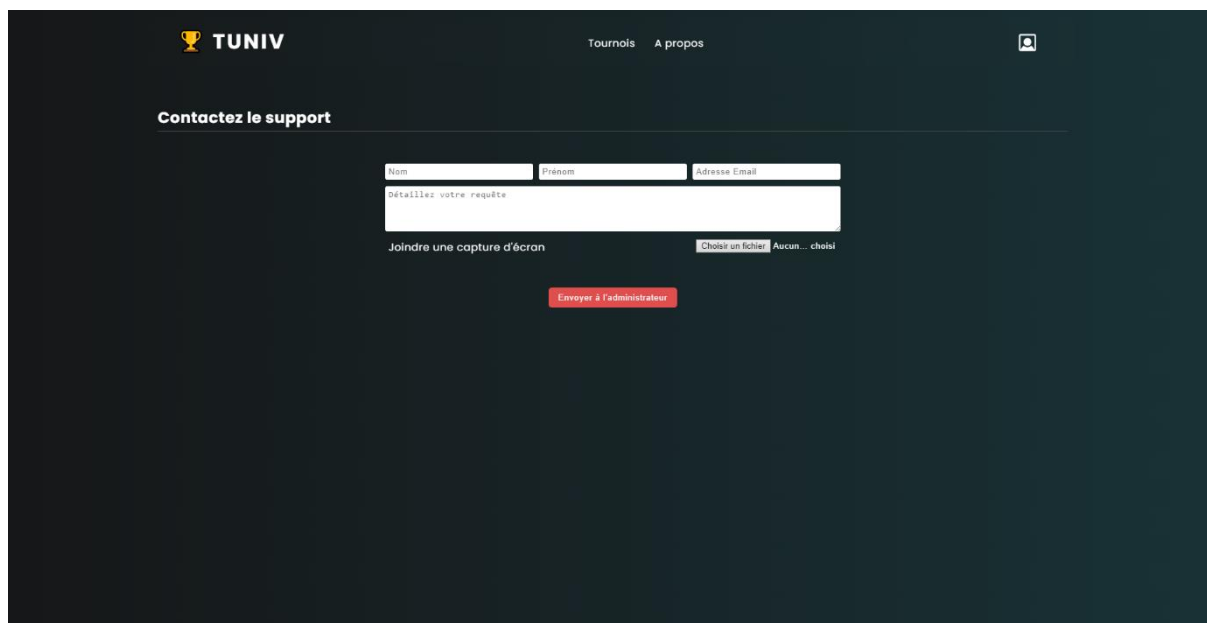
KaazDW mobile menu fixe 855fb05 on Jan 3 174 commits

assets	mobile menu fixe	3 months ago
config	changement hierarchie fichier + update responsive dashboard admin	3 months ago
modules	mobile menu fixe	3 months ago
pages	changement hierarchie fichier + update responsive dashboard admin	3 months ago
dbase.db	Update Responsive Header	4 months ago
README.md	Update README.md	4 months ago
app.js	mobile menu fixe	3 months ago
dbase.db	Update Responsive Header	4 months ago
index.php	Image par défaut annonce + 400 char contenu	4 months ago
information.md	header rework & more update	5 months ago

Natharagon Merge branch 'main' of https://github.com/KaazDW/A2S3-SAE-TUNIV... bbe46b0 5 minutes ago 388 commits

config	Merge branch 'main' of https://github.com/KaazDW/A2S3-SAE-TUNIV in...	2 hours ago
modules	page support + mot de passe oublié + correction style couleur backgrou...	13 hours ago
pages	page support + mot de passe oublié + correction style couleur backgrou...	13 hours ago
rendu	Documents	5 minutes ago
webroot	screen pour doc	6 minutes ago
.gitignore	Changement mot de passe	18 hours ago
README.md	Create README.md	3 months ago
app.php	page support + mot de passe oublié + correction style couleur backgrou...	13 hours ago
dbase.db	Update Responsive Header	4 months ago
index.php	visual finish	last month
oldREADME.md	Rename README.md to oldREADME.md	3 months ago
test.php	first part XSS fixing	2 months ago

Enfin, en termes d'accessibilité, des pages de contact et de support ont été mises en place pour permettre aux utilisateurs de contacter les administrateurs afin de faire des retours ou des requêtes.



The screenshot shows a dark-themed web page for TUNIV. At the top, there is a navigation bar with the TUNIV logo, the text 'Tournais A propos', and a user profile icon. Below this, a section titled 'Contactez le support' contains a form with three input fields for 'Nom', 'Prénom', and 'Adresse Email'. A larger text area below these fields is labeled 'Détaillez votre requête'. At the bottom of the form, there are two links: 'Joindre une capture d'écran' and 'Choisir un fichier' (with a sub-link 'Aucun... choisi'). A red button labeled 'Envoyer à l'administrateur' is positioned at the bottom right of the form area.

II.2 ARCHITECTURE

Le site est structuré dans différentes pages web PHP, divisée en plusieurs dossiers :

- Le dossier 'webroot', qui contient la page de départ index et un sous-dossier « assets » dans lesquels se trouvent tous les fichiers nécessaires à l'affichage de chaque page (images, feuilles de style SCSS). La page index, quant à elle, renvoie directement à une page app, située à la racine du projet.
- La page app est le centre de l'architecture routeur du projet, car elle permet d'accéder à tous les autres fichiers. C'est également ici qu'est initialisée la connexion à notre base de données MariaDB, qui utilise la classe PDO de PHP (comme cette page renvoie à toutes les autres, elles ont donc toutes accès à la base de données).
- Les vues des autres pages sont stockées dans un dossier page et accessibles grâce aux routes mises en place dans le fichier routeur app. Quant aux traitements des informations liées aux formulaires et autres entrées utilisateurs, ils sont effectués dans des fichiers de configuration, stockés dans un dossier à part nommé config. Lorsqu'un traitement est requis, c'est le routeur qui renvoie vers la page de configuration correspondante, qui traite les données et modifie la base de données si besoin avant de renvoyer l'utilisateur vers la page d'origine (ou une page correspondante dans le cas de la création d'un nouveau tournoi par exemple).

Le choix a été fait de ne pas utiliser de framework web comme CakePHP car compte tenu de la taille du projet au début du deuxième semestre, refaire toute l'architecture en utilisant un framework nous aurait coûté plus de temps que nous n'en aurions gagné.

La sécurité étant le point le plus important de l'amélioration du projet, nous y avons consacré beaucoup de temps, en particulier sur la mise en place de l'architecture routeur (sur laquelle ont travaillé Gaël Journet et Nathan Ozimek) et de la protection de la connexion contre les attaques en force brute (sur laquelle ont travaillé Jean-François Marcourt et Nathan Ozimek).

Des améliorations quant à l'algorithme de génération des tournois (comme la possibilité de choisir le nombre de phases de poules, ou la prise en compte des résultats précédents de chaque équipe dans la génération des poules) n'ont finalement pas été mises en place par manque de temps et car la sécurisation du projet était notre priorité.

II.3 ARCHITECTURE ROUTEUR

L'architecture routeur a permis de grandement contribuer à la sécurité et qualité logicielle du projet. Lors du lancement du serveur, le site démarre à l'intérieur du dossier webroot, sur la page index.php qui inclut la page routeur app.php à l'aide d'une commande require_once. L'utilisation du require_once permet de s'assurer que les visiteurs passent par le routeur, et ne peuvent donc pas contourner l'architecture, tout comme le placement d'index.php à part dans le dossier webroot, d'où il n'est possible d'accéder à aucune page.

La page routeur index.php récupère toute URL entrée par l'utilisateur ou par un lien du site grâce à la variable superglobale \$_SERVER, en utilisant la fonction explode() afin de pouvoir utiliser de potentielles variables passées en GET.

L'URL est ensuite entrée dans un switch, qui redirige vers la page correspondante à l'aide d'un require_once. Le passage des variables passées en GET nous a d'abord posé problème, car nous ne voyions pas de façon de les faire transiter par le routeur malgré le fait qu'elles étaient cruciales au bon fonctionnement du site. Après avoir découvert et mis en place la fonction explode() pour récupérer l'URL tout en conservant ces variables, ce problème a été résolu.

Les pages de configuration associées à des formulaires, comme celles liées à la création de nouveaux tournois ou utilisateurs, n'ont pas de route spécifique. Lorsque le routeur détecte la route d'une page contenant un formulaire, il vérifie si la méthode est GET (signifiant que l'utilisateur accède simplement à la page) ou POST (signifiant que l'utilisateur a validé le formulaire et que la page s'est rechargée en méthode POST) et renvoie respectivement vers la page du formulaire ou sa page de configuration correspondante.

Les pages de configuration non associées à des formulaires ont leur propre route, car la méthode précédente ne peut pas être utilisée dans leur cas : afin de garantir la sécurité de ces pages sensibles, une vérification du statut de l'utilisateur est faite lors de l'accès à ces pages. Si l'utilisateur n'est pas un administrateur, la page n'effectue aucun traitement et le renvoie à la place immédiatement vers la page d'index.

Dans le cas où un utilisateur entre une URL qui n'existe pas, le routeur renvoie à la page par défaut d'erreur 404.

II.4 PROTECTION DE LA CONNEXION UTILISATEUR

Dans l'optique d'une mise en production future il est nécessaire voir primordial de ne pas lésiner côté sécurité. Et quoi de plus sensible qu'un login sur un site web ?

Pour sécuriser le login nous avons deux possibilités :

- Développer nous même un algorithme permettant de vérifier que dans la même intervalle chronologique un utilisateur ne soumet pas une trop grande quantité de requêtes de

connexion, si cela arrivait notre algorithme devrait alors bloquer toutes les requêtes provenant de l'adresse IP de connexion de l'utilisateur.

- La deuxième option est de mettre en place une vérification par captcha sur le login, empêchant donc la soumission de toute requête ne provenant pas d'un réel utilisateur et ainsi prévenant toute tentative d'attaque en force brute.

L'option qui a été mise en application sur notre projet est celle du captcha, car elle nous permet d'atteindre le même objectif de protection tout en rentrant dans les délais restants.

Le captcha mis en place est le Turnstile de Cloudflare, car contrairement à celui de Google, celui de Cloudflare est rapide à passer en tant qu'utilisateur donc ne pénalise par l'ergonomie de l'application et est respectueux des RGPD.

La mise en place a été faite en suivant les documentations officielles et fonctionne de sorte à vérifier l'utilisateur avant que la requête de connexion ne soit envoyée au serveur, ce qui évite ainsi l'envoi de requêtes inutiles.

Pour résumer le fonctionnement de notre système de connexion, lorsqu'un utilisateur cherche à se connecter, il entre ses identifiants et attend que le captcha se valide, celui-ci validé il peut alors tester sa connexion en envoyant ses identifiants à notre fonction d'authentification, qui a pour objectif de comparer avec la base de données la véracité des informations de login envoyées. Selon la réponse de cette fonction, la connexion est initialisée avec les permissions et rôles associés au compte ou refusée, renvoyant alors l'utilisateur sur la page de connexion, où il est nécessaire d'attendre une nouvelle fois la vérification du captcha avant de pouvoir retenter de se connecter.

III. RESULTATS

Les points centraux de l'amélioration du projet énoncés lors de l'introduction ont tous été améliorés au cours du projet. L'application est bien plus sécurisée grâce à l'architecture routeur rendant les fichiers seulement accessibles selon ce qui a été défini lors du développement et aux mesures de défense contre de multiples formes d'attaques ayant été mises en place (captcha pour protéger la connexion contre les attaques en force brute, protection des formulaires contre les insertions de code malveillant, chiffrement des mots de passe à l'intérieur de la base de données).

En termes d'ergonomie, des tâches qui requéraient auparavant une intervention manuelle de l'administrateur sur le site ou dans la base de données (changement de phase d'un tournoi, modification du score d'un match verrouillé, changement d'un mot de passe) sont désormais automatisées et, dans le cas du changement de mot de passe, accessibles à tout le monde pour son propre compte, ce qui réduit la charge de travail imposée sur l'administrateur.

La qualité logicielle a été améliorée grâce à une révision de notre code afin d'adhérer aux standards PHP et de le rendre plus facilement maintenable, et le site est également plus accessible grâce à l'ajout d'une page de contact.

TÂCHE	DESCRIPTION	FINIE	NON-RÉSOLUE	MISE DE COTE
SÉCURITÉ/Q. LOGICIELLE : Architecture routeur	Mise en place d'une architecture routeur sur notre projet PHP conformément aux cours de PHP de Mr. Robin Frère afin de nettoyer notre architecture de fichiers et d'en limiter l'accès d'un point de vue sécurité.	X		
ERGONOMIE : Automatisation du changement de phase	Automatisation du passage à la phase suivante d'un tournoi lorsque le dernier match de la phase actuelle est verrouillé (avec génération des matchs de la prochaine phase ou verrouillage du tournoi s'il était en phase finale).	X		
ACCESSIBILITE : Mise en place d'une page de support	Mise en place d'une page de support accessible par n'importe quel visiteur du site afin de pouvoir prendre contact facilement avec l'administrateur (ex : signaler un bug, demande de contact ou effectuer une demande de réinitialisation de son mot de passe de compte pour oubli ou autre)	X		
RENDU : Déploiement de l'application sur un server web	Le déploiement de l'application sur un serveur web ne fonctionne pas totalement, nous n'arrivons pas à faire fonctionner les différents chemins afin de pouvoir inclure le CSS, SCSS ainsi que les différents modules (header, head, ...). Nous avons utilisé FileZilla et SSH afin de transférer notre projet sur le serveur, ainsi que apache2 et MariaDB.			X
SÉCURITÉ : Correction des failles XSS présentes sur le site	Sur notre application sont présents de nombreux formulaires demandant une entrée utilisateur : pour créer un tournoi, une équipe, actualiser les scores de matchs etc. Notre application n'utilisant aucun Framework spécifique la prévention des insertions de code malveillant au sein de ses formulaires n'est pas gérée automatiquement. Ainsi après documentation et tentative d'auto-attaque nous avons mis en application une protection sur l'ensemble des formulaires empêchant l'envoi de code malveillant au sein de la base de données, protégeant ainsi autant notre application que ses utilisateurs.	X		
SÉCURITÉ : Chiffrement des mots de passe	Chiffrement des mots de passe déjà présents dans la base de données et chiffrement du mot de passe d'un utilisateur lors de sa création afin qu'ils ne soient pas lisibles lorsqu'on consulte la base de données.	X		
SÉCURITÉ : Protection anti force brute sur la connexion	Mise en place d'un captcha permettant de sécuriser l'envoi des requêtes de connexion permettant aux utilisateurs de se connecter. Ce captcha permet ainsi d'empêcher toute tentative d'attaque en force brute de notre formulaire de connexion par la nécessité de celui-ci d'être valide avant l'envoi des informations de	X		

	connexion à l'algorithme de vérification de l'authentification.			
Q. LOGICIELLE : Nettoyage du code et adaptation de celui-ci aux normes PHP et RGAA	Code review sur l'ensemble du projet afin de retirer toutes les lignes inutiles, commenter les fonctions qui ne sont pas explicite, et adapter le contenu de notre code aux normes de développement de PHP et au RGAA (Référentiel Général d'Amélioration de l'Accessibilité).	X		
ERGONOMIE : Changement et réinitialisation des mots de passe	Possibilité pour l'administrateur de réinitialiser le mot de passe d'un utilisateur (par exemple suite à une demande faite depuis le formulaire de contact). Chaque utilisateur peut également modifier lui-même son mot de passe depuis son compte.	X		
ERGONOMIE : Mise en place d'une page d'erreur 404	Ajout d'une page d'erreur 404 vers laquelle sont redirigés les utilisateurs s'ils tentent d'accéder à une URL non prévue par le routeur.	X		
ERGONOMIE : Modification des scores d'un match verrouillé par l'administrateur	Un administrateur ne pouvait auparavant pas modifier les scores d'un match verrouillé, même en cas d'erreur : c'est désormais possible.	X		

Cependant, certains objectifs définis au début du projet n'ont pas été tenus : des améliorations concernant les algorithmes de génération des tournois, permettant de prendre en compte plus de paramètres pour que le système soit plus souple et adaptable à différents types de sport, n'ont pas été réalisées par manque de temps. Certains aspects, comme l'accessibilité, auraient également pu bénéficier de plus d'améliorations, comme avec un mode daltonien.

IV. CONCLUSION

Dans l'ensemble, les objectifs définis en début de projet ont été atteints : il y a eu de vraies améliorations en termes de sécurité, d'ergonomie, d'accessibilité et de qualité logicielle, comme démontré précédemment. Certaines améliorations importantes, notamment celles liées aux tournois, n'ont pas été réalisées à cause d'un manque de temps qui découle probablement de problèmes d'organisations.

Bilans personnels :

Gaël Journet : Ce projet m'a permis de mettre en pratique beaucoup de compétences apprises au cours de mes 4 semestres au sein du BUT Informatique. Cela peut s'illustrer avec le langage de notre projet qui est PHP, la gestion d'une base de données, fournir un code propre et compréhensible pour mes collègues de travail. Mais aussi d'un point de vue de gestion du projet car cela demande une grande organisation par différents moyens tels que TRELLO ainsi que par la communication, la répartition des tâches et du travail d'équipe. Je pense que ce projet va me permettre de m'intégrer plus facilement au sein d'une équipe professionnelle comme lors de mon stage et mon alternance.

Jean-François Marcourt : Tout au long du développement du projet et au vu des rôles que j'ai eu l'occasion de jouer, j'ai sincèrement le sentiment d'avoir beaucoup progressé. En tant que Lead Développeur, j'ai appris à communiquer ma vision à mes camarades, à définir des objectifs pour respecter des délais, à répartir les tâches relativement aux préférences et/ou aptitudes de chacun, à motiver les troupes lors de périodes de relâchement ou encore à réagir lorsque de mauvaises décisions ont été prises. En tant que développeur, c'est principalement sur le plan de la qualité de code et de la sécurité que j'ai pu progresser, avoir une preuve concrète de l'utilité de coder proprement, de respecter les conventions ou encore de communiquer avec son équipe à ce propos. Le plan de la sécurité logicielle est sûrement le domaine dans lequel j'ai pu le plus apprendre ce semestre, en observant à quel point il est simple de s'introduire ou de nuire à une application non protégée, j'ai eu l'occasion d'effectuer beaucoup de tests sur notre propre projet qui malheureusement pour nous se sont retrouvés fonctionnels. C'est après ces observations que j'ai pu apprendre à mettre en place des moyens de s'en protéger, qui au fil du développement sont devenus des automatismes de réflexion lors du développement de projet.

Nathan Ozimek : Je pense que ce projet m'a permis d'utiliser plusieurs des compétences acquises au cours du BUT, notamment celles liées au langage PHP et aux liens faits entre une application/un site web et une base de données qui doit être en constante adaptation aux besoins de la plateforme utilisateur. Mes compétences de travail en groupe ont également été très utiles vis-à-vis de la répartition des tâches et du travail en équipe avec mes collègues (rester en dialogue avec l'équipe afin de savoir sur quelles parties du projet il me fallait travailler en priorité afin de permettre un meilleur avancement aux autres par exemple).

RESUME EN FRANÇAIS :

L'objectif du projet était l'amélioration du site web de gestion de tournois sportifs universitaires Tuniv développé au premier semestre concernant la sécurité, l'ergonomie, l'accessibilité et la qualité logicielle.

Ces objectifs ont dans l'ensemble été atteints, grâce à une protection de la connexion utilisateur et le passage à une architecture routeur pour la sécurité, l'automatisation et l'ajout de certaines fonctions pour l'ergonomie, et une révision du code dans son ensemble pour la qualité logicielle. Certains objectifs, notamment en termes d'accessibilité et d'ergonomie, n'ont pas été entièrement aboutis par manque de temps.

MOTS CLES :

Développement web, architecture routeur, captcha, PHP, chiffrement, base de données, sport, tournois

MATERIEL / LOGICIEL / METHODE UTILISE(E)(S) :

Ordinateurs de l'IUT et personnels, Visual Studio Code comme IDE, MySQLWorkbench sur les ordinateurs de l'IUT et une combinaison de Xampp/HeidiSQL sur l'ordinateur personnel, ProxMox pour l'hébergement