



Projet d'Intégration des acquis de la première année Génie Informatique

Présenté par :

Les Elèves Ingénieurs de l'équipe XTRADH

Titre

Mise en place d'une application web pour la gestion du paiement des vacations et des heures supplémentaires à l'échelle de l'université Abdelmalek Essaâdi

Une Approche Basée des pratiques DevSecOps

Date de présentation : 15 Juin 2023

Membres du jury :

Pr. Mohamed EL HADDAD	ENSA de Tanger	Examinateur
Pr. Houda BELMOKADEM	ENSA de Tanger	Examinateuse
Pr. Jaber EL BOUHDIDI	ENSA de Tétouan	Examinateur
Pr. Mohamed GHAILANI	ENSA de Tanger	Encadrant/Animateur
Mr. Aboubaker IBRAHIM HASSAN	ENSA de Tanger	Etudiant Tuteur
Mr. Boudan Ahmad	ENSA de Tanger	Etudiant Tuteur

Avant-propos

Nom de l'équipe : XTRADH

Membre de l'équipe	Son rôle dans le projet
EL BAGHDADI NADA	Chef de projet
GLIOUAL YASSINE	Développeur Laravel
ECHBERAAL ANASS	Développeur Laravel
BOUKTAIB SOUFIAN	Développeur VueJS
EL HARRACHY ABDERRAHMANE	Administrateur système LINUX
ACHERNANE MOHAMED	Administrateur système LINUX
EL MOUMOUHI HOUDA	Développeur PL/PGSQL
EL HADDAD MOHAMED	Responsable de la sécurité applicative
AMINE TAHAA	Responsable des tests
MOHAMMED YASSINE EL AALOUCH	Responsable GIT/GITHUB

Remerciements

Je tiens à exprimer mes sincères remerciements à notre professeur encadrant, Mr. EL GHAILANI MOHAMED, pour son soutien inestimable tout au long de ce projet. Sa présence, son expertise et ses conseils éclairés ont grandement contribué à la réussite de ce travail. L'équipe est reconnaissante de l'opportunité qui nous a été donnée de bénéficier de ses connaissances et de sa guidance, qui ont joué un rôle déterminant dans la réalisation de ce projet.

J'adresse également mes remerciements à toute l'équipe pédagogique de l'École Nationale des Sciences Appliquées de Tanger (ENSAT) pour leur engagement et leur contribution à notre formation. Leur enseignement de qualité et leurs efforts ont été essentiels pour notre progression académique et le développement de nos compétences.

Je tiens à exprimer ma gratitude envers tous les membres de l'équipe, qui ont été des partenaires précieux tout au long de ce projet. Leur collaboration, leur support et leur esprit d'équipe ont été d'une importance capitale pour le succès de notre travail.

Enfin, je souhaite remercier tous ceux qui ont contribué de près ou de loin à la réalisation de ce projet. Votre appui, vos conseils et votre confiance nous ont permis de surmonter les défis rencontrés et d'atteindre les objectifs fixés.

Je suis reconnaissante envers notre encadrant et toutes les personnes qui ont contribué à cette expérience enrichissante. Votre soutien a été inestimable et on vous en est profondément reconnaissants.

Sommaire

Table des matières

AVANT-PROPOS	2
REMERCIEMENTS	3
1. CHAPITRE 1 :.....	6
1.1. METHODOLOGIE	6
1.2. JIRA.....	9
1.1. QU'EST-CE QUE JIRA SOFTWARE ?	10
1.2. QU'EST-CE QU'UN TICKET DANS JIRA ?.....	11
1.3. QU'EST-CE QU'UN WORKFLOW ?	11
1.4. COMMENT VISUALISER MON PROJET SUR JIRA ?	9
1.5. QU'EST-CE QU'UN BACKLOG ?	10
1.6. QU'EST-CE QU'UNE FEUILLE DE ROUTE ?	10
1.7. QU'EST-CE QU'UN TABLEAU DE BORD ?	11
1.8. PLANNING DE REALISATION :.....	14
2. CHAPITRE 2 :.....	15
2.1. LES FICHIERS DE MIGRATION :.....	16
2.2. FACTORIES :	21
2.3. MODELS & RELATIONS	26
2.4. CONTROLEURS	34
2.4.1. <i>Contrôleur Administrateur</i>	34
2.4.2. <i>Contrôleur Enseignant</i>	37
2.4.3. <i>Contrôleur Établissement</i>	42
2.4.4. <i>Contrôleur Grade</i>	42
2.4.5. <i>Contrôleur Intervention</i>	42
2.4.6. <i>Contrôleur Paiement</i>	45
2.4.7. <i>Contrôleur User</i>	49
2.4.8. <i>Contrôleur Forget</i>	56
2.5. GENERATION DES PDFs :	59
2.6. ROUTES	66
2.7. LIAISON AVEC FRONT-END	68
2.8. LES NOTIFICATIONS :.....	69
2.9. LES TESTS DE FONCTION :.....	71
3. CHAPITRE 3 :.....	72
3.1. INSTALLATION DE VUEJS :	73
3.2. NECESSITE DE L'OUTIL VUEX DANS LE PROJET :	79
3.2.1. <i>C'est quoi Vuex ?</i>	79
3.2.2. <i>But de vuex ?</i>	79
3.2.3. <i>Comment installer Vuex dans notre projet ?</i>	79
3.3. STRUCTURE DE NOTRE PROJET :	83
3.4. CONNEXION -MOT DE PASSE OUBLIE- RECUPERATION DE MOT DE PASSE.....	87
3.5. LES INTERFACES :	96
3.5.1. ADMIN D'ETABLISSEMENT	96
3.5.2. ADMIN D'UNIVERSITE :	111
3.5.3. DIRECTEUR D'ETABLISSEMENT	124
3.5.4. PRESIDENT D'UNIVERSITE.....	128
3.5.5. PROFESSEUR :	133

4. CHAPITRE 4 :	141
4.1. POSTFIX	142
1- <i>Etape1-Configuration de hostname :</i>	142
2- <i>Etape2- Installer Postfix</i>	143
3- <i>Etape3- Configuration Postfix</i>	144
4- <i>Etape4- Préparation Envoi</i>	146
5- <i>Etape5-Tester le serveur SMTP</i>	150
4.2. APACHE.....	151
4.2.1. <i>Etape1- Configuration Avant l'installation :</i>	151
4.2.2. <i>Etape2- Installer Appache2:</i>	151
4.2.3. <i>Etape3- Firewall Configuration :</i>	153
4.2.4. <i>Etape4- Creating Website:</i>	155
4.2.5. <i>Etape5- Installing SSL Certification</i>	158
4.2.6. <i>Etape6- Teste de Site Web...</i>	159
HTTP	159
4.2.7. <i>Etape7- Security de Site Web:</i>	159
4.3.1. ÉTAPE 1 : L'INSTALLATION DE "NGINX":.....	162
4.3.2. ÉTAPE 2 : GERER LE PROCESSUS "NGINX":.....	163
4.3.3. ÉTAPE 3 : FICHIER ET REPERTOIRE:.....	165
4.3.4. ÉTAPE 4 : CONFIGURATION DU SERVEUR:	168
PARTIE HTTP (PORT 80):	168
PARTIE HTTPS (PORT 443):.....	170
<i>ssl_certificate /home/mohamed/ssl-test/certificate.crt; ssl_certificate_key /home/mohamed/ssl-test/private.key ;</i>	170
4.4.1. QU'EST-CE QUE NGINX ?	174
4.4.2. AVANTAGES DE NGINX:	174
4.4.3. WHAT IS APACHE?	175
4.4.4. AVANTAGES D'APACHE :	175
4.4.5. PRINCIPALES DIFFÉRENCES ENTRE NGINX ET APACHE :	176
4.4.6. CONVERSION NGINX, POURQUOI ?	176
4.5. DNS	178
4.5.1. Qu'est-ce que c'est le DNS (Domain Name System) ?	178
4.5.2. Configuration DNS dans Notre Projet.....	179
4.6. DOCKER	188
4.6.1. Étape 1: Installation de "Docker"	188
4.6.2. Étape 2: Utilisation de la commande Docker	191
4.6.3. Étape 3: Travailler avec des images Docker	191
● EXECUTION D'UN CONTENEUR DOCKER.....	192
● GESTION DES CONTENEURS DOCKER	193
4.6.4. Étape 4: Dockerisation de notre projet.....	196
2. CONFIGURATION DU CONTENEUR LARAVEL API	203

5. CHAPITRE 5 :	233
5.1. PRESENTATION GIT ET GITHUB	234
5.1.1. GIT	234
5.1.2. GITHUB	236
5.1.3. CREATION ET LA SECURISATION DU PROJET	238
5.1.4. CLONAGE	240
5.1.5. LA SECURISATION DU PROJET A L'AIDE D'UNE CLE SSH (POUR REPOSITORY OWNER)	242
5.2. METHODOLOGIE DE TRAVAIL	245
5.2.1. LE FONCTIONNEMENT DE GIT.....	245
5.2.2. WORKFLOW	247
5.2.3. GITFLOW	248
5.2.4. PULL REQUEST	251
5.2.5. INTEGRATION CONTINUE.....	254
5.3. LES TACHES DU RESPONSABLE GIT & GITHUB DANS LE PROJET.	258
5.3.1. FORMATION, REFERENTIEL D'EQUIPE ET SCRIPTS SHELL POUR UNE COLLABORATION OPTIMISEE.	258
5.3.2. RESOLUTION EFFICACE DES CONFLITS.....	261
5.3.3. GESTION PROACTIVE DES PROBLEMES DE CODE SOURCE.....	263
6. CHAPITRE 6 :.....	265
6.1. LA PREPARATION DE L'ENVIRONNEMENT DE TRAVAIL	266
6.1.1. BASE DE DONNEES.....	266
6.1.2. LES TABLES	267
6.2. L'INSERTION DES DONNEES	270
6.2.1. REQUETES SQL.....	270
6.2.2. POSTMAN	270
6.3. LES DECLENCHEURS	271
6.3.1. LE DECLENCHEUR : "AJOUTERENSEIGNANTTRIGGER".....	273
6.3.2. LE DECLENCHEUR :"AJOUTPAIMENTTRIGGER"	274
6.3.3. TRIGGER 3 : "MODIFIERPAIMENTTRIGGER"	275
6.3.4. TRIGGER 4 : "VISA_TRIGGER"	276
6.4. TEST.....	276
6.4.1. FONCTIONS & PROCEDURES STOCKEES	277
6.5. DECLENCHEURS	286
6.5.1. TEST DE DECLENCHEUR 1 :	286
6.5.2. TEST DE DECLENCHEUR 2 :	289
6.5.3. TEST DE DECLENCHEUR 3 :	293
6.5.4. TEST DE DECLENCHEUR 4 :	296
6.6. CREATION DES UTILISATEURS :	297
6.6.1. UTILISATEUR 1 : ADMIN_UNIVERSITE	297
6.6.2. UTILISATEUR 2 : ADMIN_ETB	297
6.6.3. UTILISATEUR 3 : PRESIDENT_UNIVERSITE	298
6.6.4. UTILISATEUR 4 : DIRECTEUR_ENSAT	298
7. CHAPITRE 7 :.....	300

7.1. SECURISATION DE L'AUTHENTIFICATION	302
7.1.1. POURQUOI SANCTUM :	302
7.1.2. LES METHODES PRINCIPALES DU CONTROLEUR D'AUTHENTIFICATION	304
7.1.3. SECURISATION DES ROUTES QUI NECESSITE UNE AUTHENTIFICATION :	305
7.1.4. LA SECURISATION DES ROUTES PAR ROLES SPECIFIQUE :	306
7.1.5. LE HACHAGE DU MOT DE PASSE :	308
7.1.6. VALIDATION DES DONNES ENTRES COTE SERVEUR :	309
7.2. LES ATTAQUES XSS ET LES INJECTIONS SQL :.....	311
7.2.1. INJECTION SQL :	311
7.2.1.1. UTILISATION DE REQUETES PREPAREES (PREPARED STATEMENTS):	312
7.3. SECURITE DU CODE COTE CLIENT (FRONTEND) :	315
7.3.1. UTILISATION D'ATTRIBUTS NONCE	316
7.3.2. UTILISATION D'AXIOS.....	317
7.3.3. ABSENCE DE DONNEES SENSIBLES	317
7.3.4. CONTROLE DES ENTREES	317
7.3.5. CONTROLE D'ACCES RESTREIGNANT LES INTERFACES UTILISATEUR.....	318
8. CHAPITRE 8 :.....	320
8.1. PRESENTATION	321
8.2. PREPARATION DE L'ENVIRONNEMENT :	321
8.2.1. EMPLACEMENT :.....	321
8.2.2. CLASS TESTS ET MÉTHODES SETUP/TEARDOWN:	321
8.3. CATEGORIES DE TESTS :	322
8.3.1. TEST SIMPLE :	323
8.3.2. TEST REDUIT :	324
8.3.3. TEST AVANCE :	325
8.4. TESTS DU CONTROLEUR D'AUTHENTIFICATION :.....	327
8.4.1. ERREURS RENCONTREES :	327
8.4.2. RESULTATS DE L'EXECUTION DES TESTS :	330
8.5. TESTS DU CONTROLEUR USER :.....	330
8.5.1. ERREURS RENCONTREES :	330
8.5.2. RESULTATS DES TESTS APRES LA RESOLUTION :.....	334
8.6. TESTS DU CONTROLEUR D'INTERVENTIONS :.....	334
8.6.1. ERREURS RENCONTREES :	334
8.6.2. RESULTATS DES TESTS APRES LA RESOLUTION :.....	338
8.7. TEST DU CONTROLEUR D'ENSEIGNANT :.....	338
8.7.1. ERREURS RENCONTREES :	338
8.7.2. RESULTATS DES TESTS APRES LA RESOLUTION :	343
8.8. TEST DU CONTROLEUR D'ETABLISSEMENT :.....	343
8.8.1. ERREURS RENCONTREES :	343
8.8.2. RESULTATS DES TESTS APRES LA RESOLUTION :	345
8.9. TEST DU CONTROLEUR D'ADMINISTRATEUR :	345
8.9.1. ERREURS RENCONTREES :	345
8.10. TEST DU CONTROLEUR PAIEMENT :.....	347
8.10.1. ERREURS RENCONTREES :	347
8.10.2. RESULTATS DES TESTS APRES LA RESOLUTION :	350
8.11. TEST DU CONTROLEUR GRADES :.....	350
8.11.1. ERREURS RENCONTREES :	351
8.11.2. RESULTATS DE L'EXECUTION DES TESTS :	351
8.12. MENTIONS HONORABLES :	351
8.12.1. PREMIERE APPROCHE :.....	352
8.12.2. DEUXIEME APPROCHE :	354
8.13. BILAN DES TESTS BACKEND :	358
8.14. PRESENTATION DE PLAYWRIGHT :	360

8.14.1.	PREPARATION DE L'ENVIRONNEMENT :.....	360
8.14.2.	L'UTILISATION DU CODEGEN :.....	360
8.14.3.	OPTIONS SUPPLEMENTAIRES :	361
8.15.	TESTS DES INTERFACES	362
8.15.1.	TESTS DE L'INTERFACE D'AUTHENTIFICATION :	363
8.15.2.	TEST DES INTERFACES DU PROFESSEUR :.....	367
8.15.3.	TEST DES INTERFACES DU DIRECTEUR D'ETABLISSEMENT :.....	369
8.15.4.	TEST DES INTERFACES DU PRESIDENT UNIVERSITAIRE :.....	370
8.15.5.	TEST DES INTERFACES DE L'ADMINISTRATEUR D'ETABLISSEMENT :	373
8.15.6.	TEST DES INTERFACES DE L'ADMINISTRATEUR UNIVERSITAIRE :	374
8.16.	TEST DE RECUPERATION DU MOT DE PASSE :.....	376
8.17.	TESTS DE SECURITE :.....	379
8.17.1.	UTILISATION DE L'SQLMAP :.....	379
8.17.2.	UTILISATION D'OWASP ZAP :	384
8.17.3.	UTILISATION DE BURPSUITE :.....	386
8.18.	TESTS DES PERFORMANCES (APACHE JMETER):	388

INTRODUCTION

Le présent rapport final marque la conclusion d'un projet ambitieux de développement d'une application web visant à digitaliser la gestion des ressources humaines à l'université Abdelmalek Essaadi. Notre équipe, composée de dix étudiants passionnés, a travaillé en étroite collaboration sous la supervision de notre enseignant qui a agi en tant que client. Le principal objectif de ce projet était de mettre en pratique nos connaissances et compétences acquises en développant une application permettant d'automatiser le processus de paiement des vacations et des heures supplémentaires effectuées par les enseignants.

L'université Abdelmalek Essaadi accueille un grand nombre d'enseignants qui doivent assurer un certain nombre d'heures d'enseignement chaque année. Selon leur grade et leur statut (permanent ou vacataire), certaines de ces heures peuvent être considérées comme supplémentaires. Les enseignants permanents doivent dépasser une charge statutaire spécifique (240h pour les PA, 200h pour les PH et 190h pour les PES) pour que les heures supplémentaires soient prises en compte. Quant aux enseignants vacataires, toutes les heures effectuées sont considérées comme supplémentaires.

La gestion des heures supplémentaires et des vacations constitue un défi majeur pour l'université, en raison de la nécessité de calculer avec précision les rémunérations en fonction du grade de chaque enseignant. De plus, il est essentiel de mettre en place un système de suivi des plannings et des paiements pour assurer une gestion efficace et transparente. Le processus actuel, basé sur des méthodes manuelles et des documents papier, est source d'erreurs et de retards, ce qui nuit à la satisfaction des enseignants et à la bonne gestion des ressources financières de l'université.

A cet égard, notre équipe de dix étudiants s'est engagée à concevoir et à développer une application web répondant à ce besoin. Pour cela, nous avons opté pour une approche agile, notamment la méthodologie Scrum, afin de garantir une collaboration étroite avec notre client et de nous adapter rapidement aux changements éventuels.

En résumé, ce rapport se veut un compte rendu exhaustif du projet, il relate le travail accompli et s'articule autour de 8 chapitres :

- Le 1er chapitre définit le contexte général du projet.
- Le 2ème chapitre concerne Laravel.
- Le 3ème chapitre concerne le Front avec VueJS
- Le 4ème chapitre concerne l'administration système LINUX.
- Le 5ème chapitre concerne la gestion de la base de données.
- Le 6ème chapitre concerne la gestion du code source avec GIT/GITHUB.
- LE 7ème chapitre concerne la sécurité applicative.
- LE 8ème chapitre concerne les tests.

1. CHAPITRE 1 :

CONTEXTE GENERAL DU PROJET

Le présent chapitre a pour objectif d'introduire le contexte global du projet, en mettant en avant la méthodologie de travail adoptée, l'outil de gestion de projet utilisé et le plan d'action suivi pour la réalisation des objectifs fixés.

1.1. Méthodologie

La première étape à prendre en considération avant l'initialisation du projet, est le choix de la méthodologie de travail. Ce choix est important vu que cette dernière impacte l'organisation du projet, la gestion des changements, la collaboration, la communication, la valeur livrée et la satisfaction du client. Une méthodologie bien adaptée peut améliorer l'efficacité de l'équipe, accroître la qualité du produit final et favoriser la réussite du projet dans son ensemble.

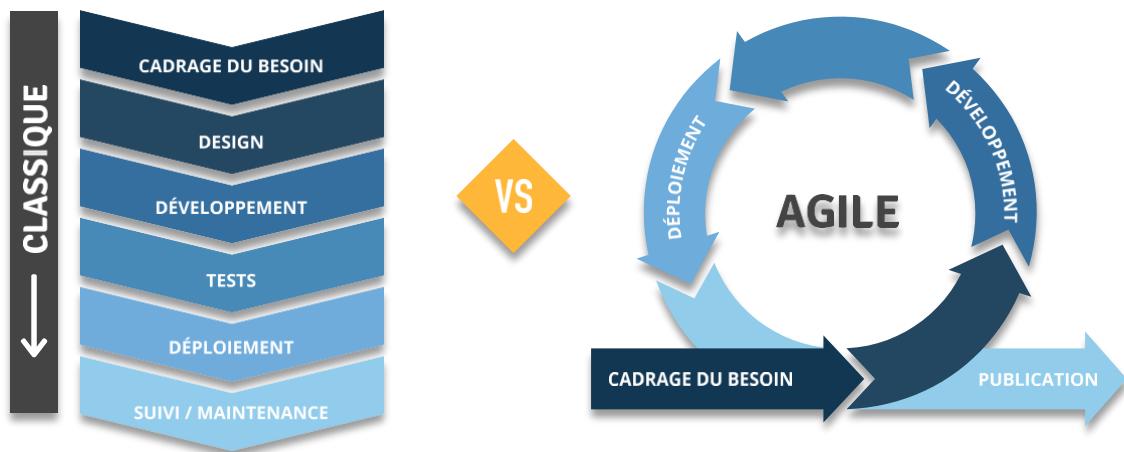
Il est donc essentiel d'examiner différentes méthodologies afin de prendre une décision éclairée. Voici une brève explication de quelques-unes des méthodologies de travail les plus couramment utilisées :

- **Méthode en cascade** : est une approche séquentielle et linéaire permettant de résoudre un problème et de délivrer un projet. Cette méthode est adaptée aux projets dont les livrables sont faciles à définir dès le départ. Les avantages de cette méthode sont qu'elle est facile à comprendre et à utiliser, elle permet de planifier et de suivre facilement les étapes du projet et elle est efficace pour les projets dont les exigences sont clairement définies. Cependant, cette méthode peut être inflexible et peu adaptée aux projets complexes ou à ceux dont les exigences évoluent rapidement. Elle ne permet pratiquement aucune marge de manœuvre pour les changements ou les révisions inattendus, car elle oblige les membres de l'équipe à ne passer à la tâche suivante qu'une fois la précédente achevée.

- **Méthode en Spirale** : Cette méthode est adaptée aux projets complexes où les risques et les incertitudes sont élevés. Elle permet une meilleure gestion des risques en identifiant et en atténuant les problèmes potentiels dès les premières phases du projet. Les boucles itératives de la méthodologie en spirale permettent de tester et de valider les idées, de recueillir des retours d'expérience et de prendre des décisions éclairées. Néanmoins, elle nécessite une bonne compréhension des risques et des contraintes du projet, ainsi qu'une planification et une coordination efficaces des itérations.

- **Méthodologie Agile** : est une démarche de développement de logiciels qui se concentre sur la collaboration entre les membres de l'équipe et la livraison rapide et continue de logiciels fonctionnels. Parmi ses avantages, elle permet une grande flexibilité et une grande réactivité aux changements, elle permet une meilleure gestion des risques, elle est efficace pour les projets dont les exigences ne sont pas clairement définies et elle favorise la collaboration entre les membres de l'équipe. Cependant, cette méthode peut être plus difficile à gérer pour les grandes équipes et les projets complexes et elle peut nécessiter une formation supplémentaire pour les membres de l'équipe pour se familiariser avec le langage qui l'accompagne.

En ce qui concerne notre projet, nous avons choisi d'adopter la méthodologie Agile pour plusieurs raisons fondamentales. Tout d'abord, l'Agilité favorise une approche itérative et incrémentielle, ce qui signifie que nous pouvons livrer des résultats concrets plus rapidement et obtenir des retours précoce de la part des parties prenantes. Cela nous permet d'ajuster nos objectifs et notre direction en fonction des besoins changeants du projet.



De plus, l'Agilité met l'accent sur la collaboration et la communication fréquentes avec les membres de l'équipe et les parties prenantes. Cette approche nous permet donc de construire un dialogue ouvert, d'identifier rapidement les problèmes et de trouver des solutions efficaces. En encourageant une culture de transparence et de responsabilité partagée, cette méthode renforce la cohésion de l'équipe et favorise un environnement de travail positif.

Enfin, l'Agilité intègre des pratiques de gestion des risques plus robustes, permettant une anticipation proactive des problèmes potentiels. En utilisant des itérations courtes et des évaluations régulières, nous pouvons identifier rapidement les risques et les contraintes, ce qui nous permet de les atténuer ou de les résoudre de manière opportune.

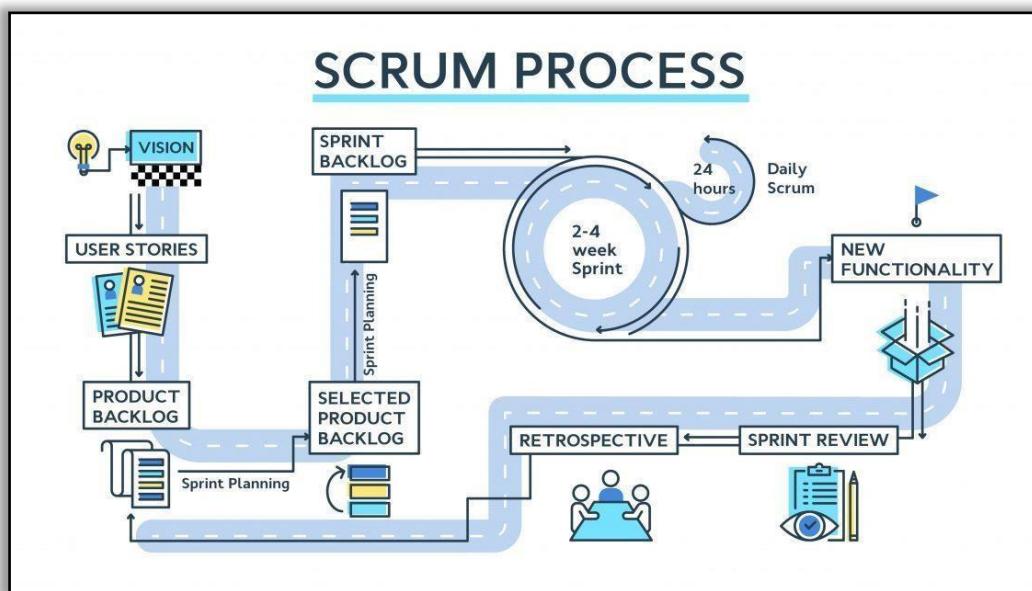
En combinant tous ces avantages, nous sommes convaincus que l'adoption de la méthodologie Agile dans notre projet nous permettra d'améliorer l'efficacité de l'équipe, d'accroître la qualité du produit final et de maximiser la satisfaction du client. L'Agilité offre la flexibilité nécessaire pour s'adapter aux changements, la transparence pour un suivi précis du projet, et une approche itérative pour une valeur livrée continue.

La question qui se pose ensuite est la suivante :

Quelle méthodologie Agile est la plus adéquate ?

En effet, il existe plusieurs méthodes agiles utilisées couramment pour la gestion de projets et le développement logiciel, notamment Kanban, Lean, Feature Driven Development (Le Développement Dirigé par les Fonctionnalités), Crystal ...etc. Cependant, parmi ces options, la méthodologie Scrum est la plus appropriée pour notre équipe d'étudiants de 10 membres, avec notre enseignant agissant en tant que client. Voici les raisons justifiant notre choix de Scrum :

- Scrum offre un cadre de travail itératif et incrémental qui convient parfaitement à notre projet. En le découplant en sprints, nous pourrons nous concentrer sur des objectifs spécifiques à court terme et livrer des fonctionnalités fonctionnelles à la fin de chaque sprint. Cela nous permettra d'obtenir des résultats concrets à intervalles réguliers, ce qui est crucial pour maintenir notre motivation et évaluer notre progression.



- La méthodologie Scrum favorise la collaboration et la communication au sein de l'équipe. Grâce à des réunions régulières comme les Daily Scrums, les revues de sprint et les rétrospectives, nous pourrons partager nos avancées, discuter des défis rencontrés et prendre des décisions collectives. Cela renforcera notre cohésion d'équipe et notre engagement envers le projet.
- En tant qu'étudiants, il est important d'être flexibles et de nous adapter aux changements et aux évolutions des exigences. Scrum offre une grande flexibilité pour gérer ces changements grâce à des itérations courtes et une planification adaptative. Cela nous permettra de prendre en compte les retours de notre enseignant-client, d'ajuster nos priorités et de garantir que le projet évolue en fonction de ses attentes.

- De plus, Scrum propose une structure de rôles bien définie, tels que le Scrum Master et le Product Owner, qui faciliteront l'organisation de notre travail, la répartition des responsabilités et la clarté des communications.
- Enfin, nous avons choisi Scrum car il s'agit d'une méthodologie agile largement adoptée dans l'industrie du développement logiciel. En acquérant de l'expérience pratique avec Scrum, nous pourrons développer des compétences et des connaissances précieuses pour notre future carrière professionnelle.

1.2. JIRA

En tant que chef de projet, je considère qu'il est de mon devoir de soutenir mon équipe, de la guider et de créer un environnement propice à son développement et à sa réussite. Voici quelques-unes de mes principales responsabilités envers mon équipe :

- Coordonner les membres de l'équipe dans l'ensemble du processus de développement de l'application web.
- Élaborer un plan de projet détaillé incluant des étapes spécifiques, des dates de livraison et des objectifs clairs pour chaque membre de l'équipe.
- Assigner des tâches à chaque membre de l'équipe, suivre l'avancement du projet, planifier les versions, et prioriser les bugs et les demandes de fonctionnalités.
- Assurer la communication entre les membres de l'équipe, ainsi qu'avec le client (professeur).
- Assurer que le cahier des charges et les délais soient respectés tout au long du projet.

Pour mener à terme mon devoir et exécuter les tâches mentionnées ci-dessus, il a été nécessaire de faire appel à l'outil JIRA, qui soutient parfaitement la méthodologie Scrum mentionnée précédemment pour la gestion de notre projet. Voici pourquoi le choix de Jira paraît le plus adéquat :

- Suivi des tâches et gestion des sprints : JIRA nous permet de créer des tâches, de les organiser en sprints et de les attribuer aux membres de l'équipe. Cela nous permet d'avoir une vue d'ensemble de toutes les tâches en cours, de suivre leur progression et d'assigner les responsabilités de manière claire et transparente.
- Tableaux scrum boards : JIRA propose des tableaux scrum boards personnalisables qui nous permettent de visualiser et de suivre visuellement l'avancement des tâches. Cela facilite la gestion des flux de travail, l'identification des goulots d'étranglement et la prise de décisions rapides pour maintenir le rythme du projet.

- Collaboration et communication : JIRA facilite la collaboration entre les membres de l'équipe en fournissant des fonctionnalités telles que les commentaires, les notifications et les discussions sur les tâches. Cela favorise la communication fluide et transparente, permettant à tous les membres de rester informés des mises à jour, des problèmes rencontrés et des éventuelles modifications des exigences.
- Suivi des problèmes et des retours : JIRA permet de signaler facilement les problèmes, les bugs et les retours d'utilisateur. Nous pouvons créer des tickets pour suivre ces problèmes, les attribuer aux membres de l'équipe et suivre leur résolution. Cela nous aide à assurer la qualité du produit final en corrigeant rapidement les problèmes identifiés.

1.1. Qu'est-ce que Jira Software ?

Jira Software est un outil de gestion de projet et de suivi des problèmes largement utilisé par les équipes de développement logiciel. Développé par la société Atlassian, Jira Software offre une plateforme centralisée pour la planification, le suivi et la collaboration tout au long du cycle de vie d'un projet logiciel.

L'objectif principal de Jira Software est d'améliorer l'efficacité et la productivité des équipes de développement en leur permettant de gérer les tâches, les problèmes et les projets de manière transparente.



1.2. Qu'est-ce qu'un ticket dans Jira ?

Un ticket est une tâche à effectuer, c'est un élément de travail. Un **ticket suit plusieurs étapes** : à faire, puis en cours, puis terminé (par exemple). On peut dire que le ticket passe dans chaque étape du workflow – flux de travail. Les tickets dans Jira Software peuvent être de plusieurs types. Dans un projet de développement logiciel, on retrouve les types de ticket suivants :

- * Une epic est considérée comme un grand objectif devant être simplifié et divisé en plusieurs tâches afin d'organiser le travail des équipes agiles – utiliser les epics dans Jira Software.
- * Une story représente une fonctionnalité à réaliser.
- * Une tâche est généralement une tâche technique à effectuer.
- * Un bug désigne un problème à corriger.



1.3. Qu'est-ce qu'un workflow ?

Un workflow, ou flux de travail, représente les étapes qu'un ticket doit suivre dans un projet. Il permet de structurer le processus de développement du produit. Au cours de notre projet, nous allons utiliser le workflow suivant : TO DO, IN PROGRESS, IN REVIEW, DONE, pour ainsi faciliter le suivi de l'avancement et l'achèvement de chaque ticket.

1.4. Comment visualiser mon projet sur Jira ?

Jira Software propose plusieurs modèles de projet pour aider les équipes à se lancer rapidement en utilisant des projets adaptés à leurs besoins spécifiques (par exemple une équipe RH ou de développement). Les 2 modèles les plus populaires sont : Scrum et Kanban.

Comme a été mentionné précédemment, au cours de notre projet, nous allons utiliser le modèle Scrum car il est le plus adéquat. Il est conçu pour les équipes qui travaillent à une cadence régulière ou qui travaillent sur de courtes périodes limitées appelées sprints. Le modèle Scrum permet également aux équipes d'organiser et de prioriser tout le travail sur le backlog et le tableau, tout en restant concentrées sur la vue d'ensemble via la feuille de route.

<p>Scrum</p> <p>Idéologie Apprendre de ses expériences et réfléchir aux réussites et aux échecs pour s'améliorer</p> <p>Cadence Sprints réguliers et à durée déterminée (à savoir, deux semaines)</p> <p>Fonctionnalités recommandées Analyses : utilisez les analyses pour optimiser la planification des sprints, améliorer la pertinence des stand-ups quotidiens et la vélocité des livraisons.</p>	<p>Backlog : le backlog contient toutes les tâches potentielles de votre équipe pour le projet.</p> <p>Sprint : un sprint désigne une brève période limitée dans le temps dont une équipe Scrum a besoin pour effectuer une quantité de travail donnée. Créez un sprint, définissez ses objectifs et ajoutez tous les tickets du backlog que votre équipe a l'intention de résoudre pendant ce sprint.</p>
--	--

1.5. Qu'est-ce qu'un Backlog ?

Un backlog produit est une liste hiérarchisée de tâches destinées à l'équipe de développement. Il est un des éléments fondamentaux de la méthodologie Scrum. Il s'agit de l'outil de travail principal du Product Owner qui se charge de recueillir les besoins auprès des parties prenantes et de les transformer en liste de fonctionnalités prêtes à être développées par l'équipe de développement.

C'est ici que la création des sprints, des epics et des tickets se fait.

The screenshot shows the Jira Backlog interface for the 'XtraDh' project. At the top, there are navigation tabs for 'Projets / XtraDh' and 'Backlog'. Below the tabs are search, filter, and version dropdowns, along with an 'Analyses' button. On the left, there's a sidebar for 'Epic' with a tree icon, a 'Create epic' button, and instructions to plan and prioritize work. The main area shows a 'Qu'est-ce qui doit être fait?' (What needs to be done?) section and a 'Backlog (0 ticket)' section with a note that it's empty. There are buttons to 'Create a sprint' and 'Create a ticket'.

1.6. Qu'est-ce qu'une feuille de route ?

Dans Jira, la feuille de route (ou roadmap en anglais) est une fonctionnalité qui permet de visualiser et de planifier les principales étapes d'un projet sur une ligne de temps. La feuille de route offre une vue d'ensemble stratégique du projet, ce qui facilite la communication, la coordination et la prise de décision.

Elle permet aussi de regrouper les différentes tâches, fonctionnalités ou versions en une vue chronologique. Elle est cruciale pour définir des objectifs à atteindre, de planifier les étapes clés du projet et de suivre l'avancement global.

La feuille de route est généralement utilisée pour définir une vision à long terme du projet et pour communiquer cette vision à toutes les parties prenantes.

PROJET XTRADH

Projets / XtraDh
Feuille de route

Envoyer un commentaire Partager Exporter ...

Catégorie d'état Epic Type

Sprints

Versions

XTRAD-5 acquisition des co... TERMINÉ(E)

XTRAD-43 Initialisation du ... TERMINÉ(E)

XTRAD-60 finalisation du projet

XTRAD-96 élimination d'err... TERMINÉ(E)

Tableau Sprint 2 MAI 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23

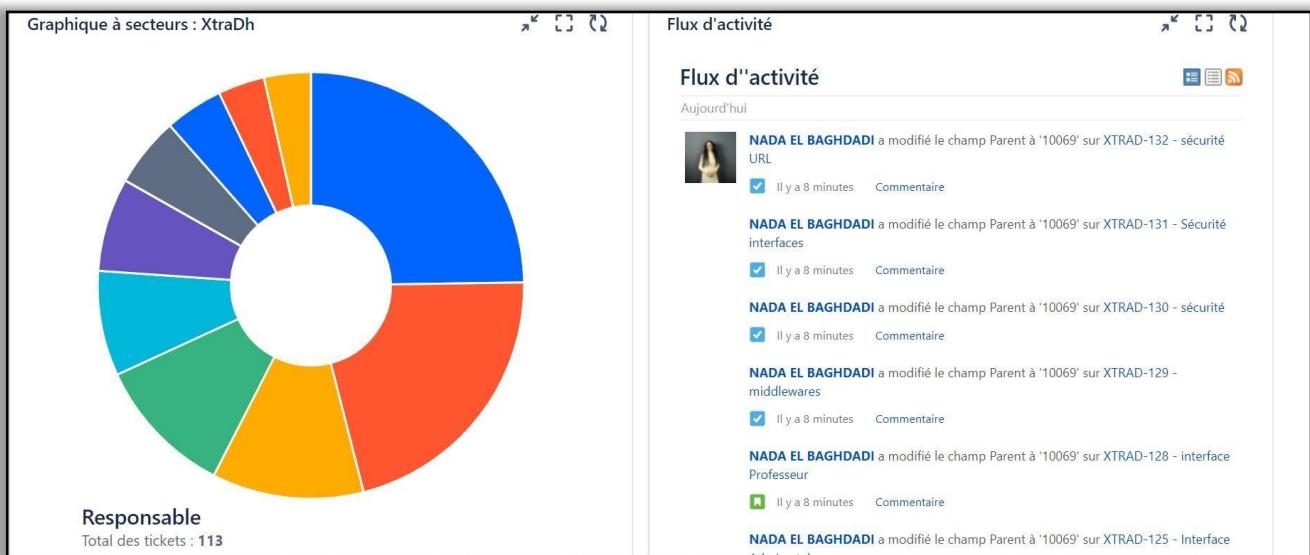
Tableau Sp MAI 22 23

1.7. Qu'est-ce qu'un tableau de bord ?

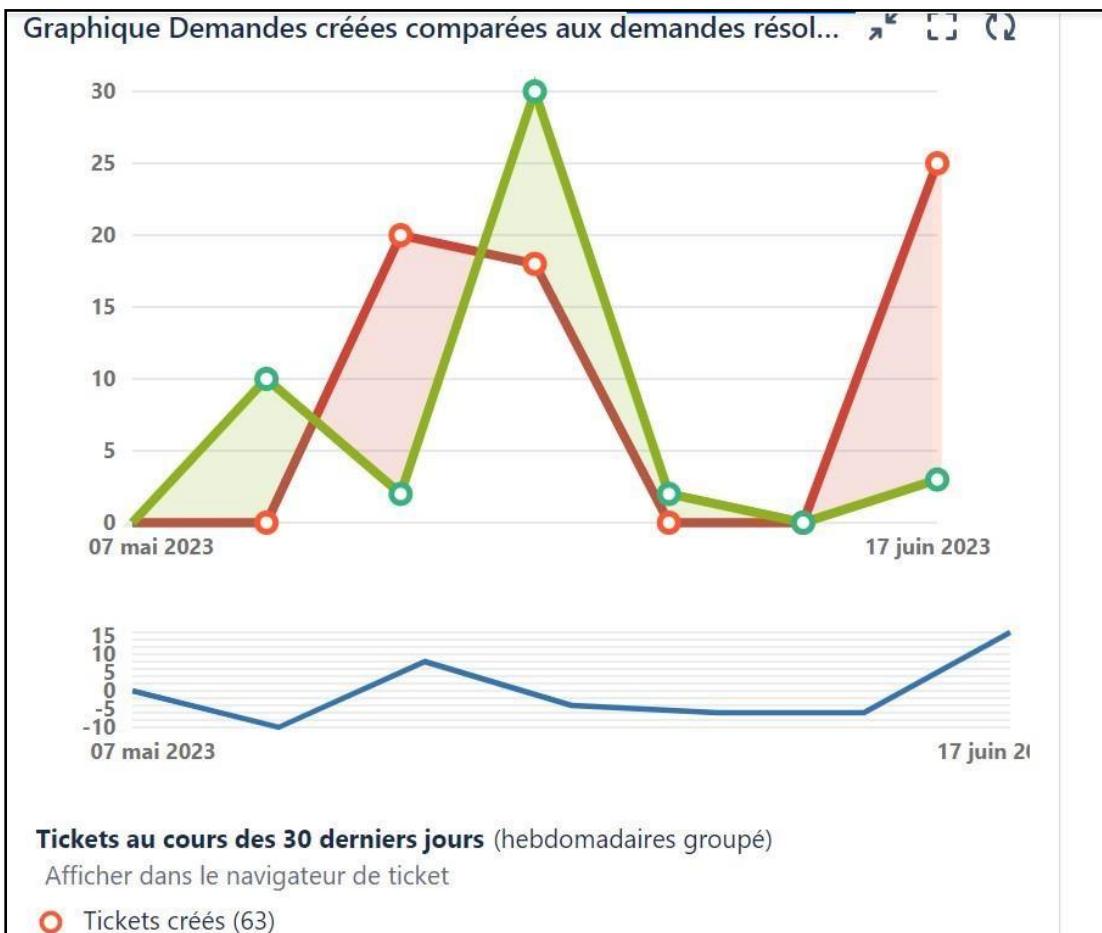
Il s'agit d'un espace centralisé où les utilisateurs peuvent suivre en temps réel les métriques et les statistiques importantes, et obtenir une vue d'ensemble de l'état du projet.

Les tableaux de bord dans Jira peuvent être configurés pour afficher différents types de gadgets. Un gadget est un élément graphique qui affiche des informations spécifiques, comme des tableaux, des listes ou des compteurs.

J'ai pris le soin de configurer un tableau de bord nommé « Our Dashboard » muni de plusieurs graphiques et statistiques qui nous seront utiles pour l'avancement et l'organisation de notre projet.

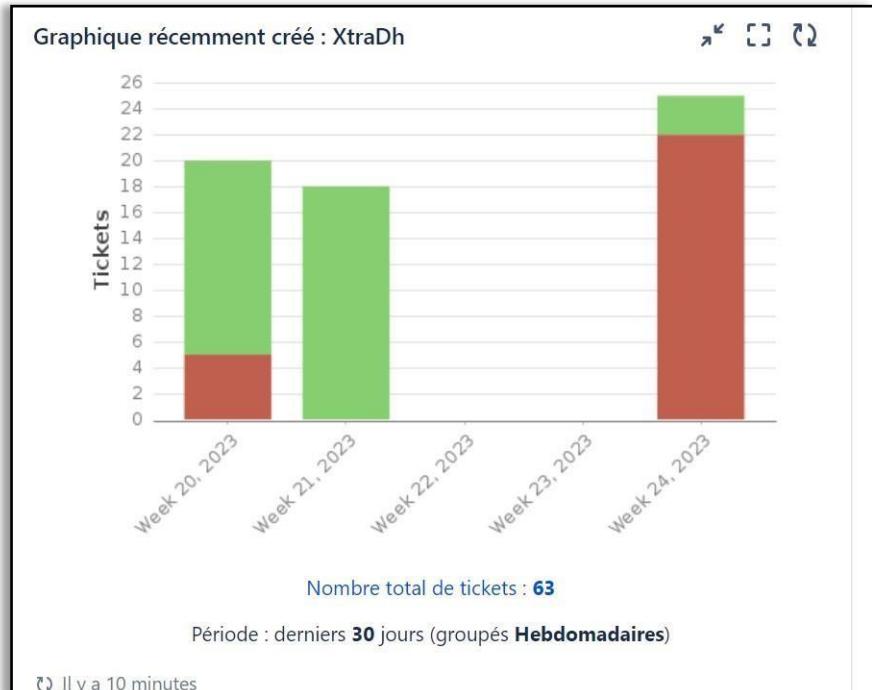


Ce graphique affiche chaque utilisateur et le nombre de tickets dont il a été responsable tout au long du projet.

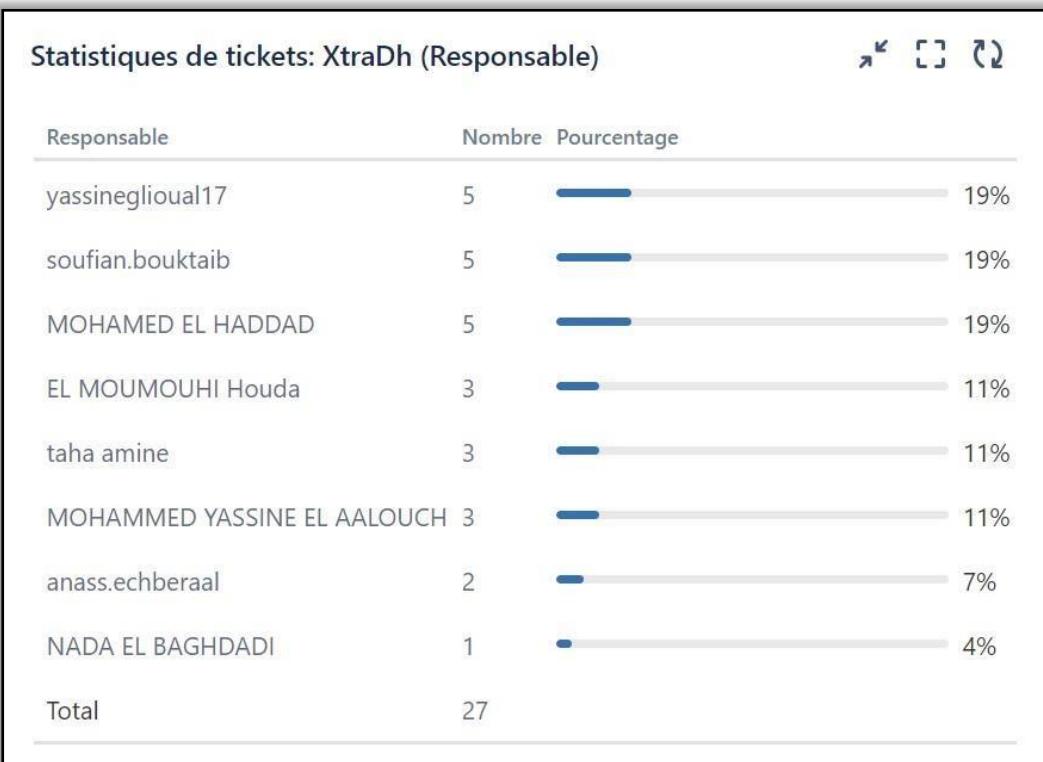


- Ce graphe permet de comparer entre les demandes résolues et les demandes créées au cours des 30 derniers jours.

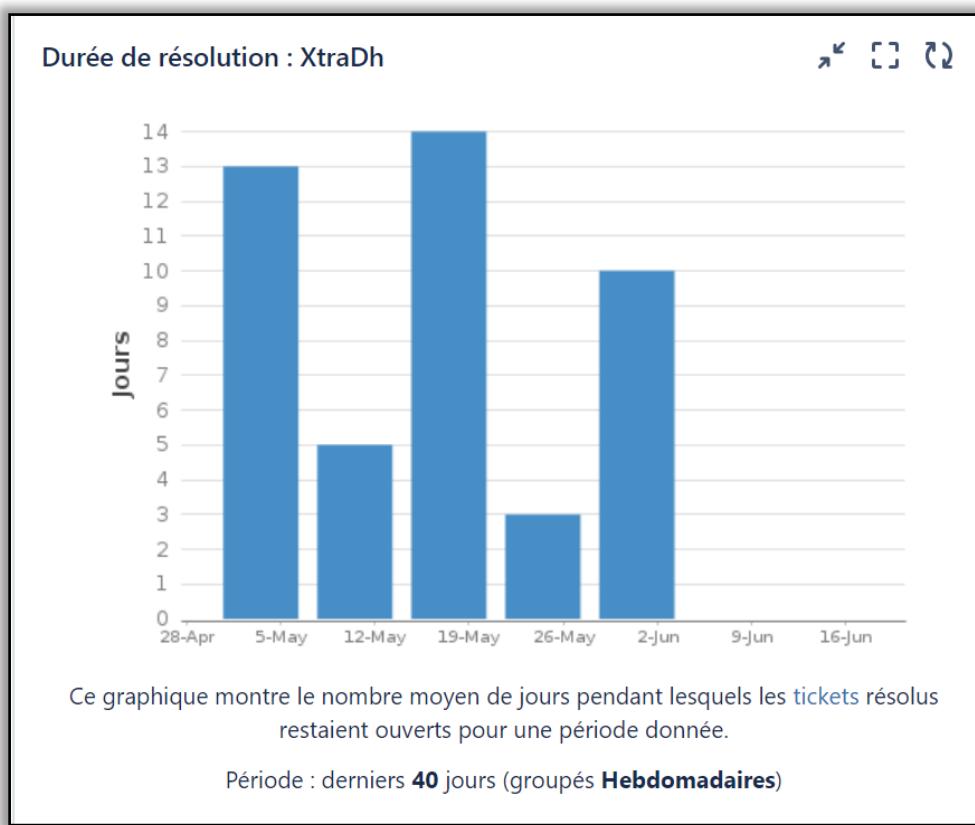
Ce graphique montre le nombre de tickets résolus (en vert) et non résolus (en rouge) groupés hebdomadairement pour une période de 30 jours.



PROJET XTRADH



Ces statistiques montrent le nombre de tickets assignés à chaque responsable et qui sont en cours de résolution.



PROJET XTRADH

The screenshot shows a project management interface with two main sections:

- Qui me sont assignés:** A table with columns: Epic Name, T Clé, Résumé, and Date d'échéance. One row is visible: "XTRAD-60 finalisation du projet" due on "13/juin/23".
- Tickets en cours:** Shows 1 ticket: "Il y a 31 secondes".

J'ai ajouté ces deux rubriques pour rappeler et faciliter l'accès aux tickets assignés ou tickets en cours propres à chaque membre de l'équipe.

1.8. PLANNING DE REALISATION :

Le planning de réalisation du projet est étroitement lié à notre outil central JIRA. Grâce à l'utilisation des sprints, des epics et d'autres fonctionnalités, nous avons pu gérer et planifier les différentes phases du projet. En effet, chaque sprint avait une durée prédéfinie et impliquait un ensemble spécifique de tâches à réaliser. Pour faciliter la planification et la gestion des différentes parties de l'application web, les epics ont également été utilisés.

La réalisation du projet s'est déroulée en quatre étapes essentielles :

- ✓ Sprint 1 : acquisition de connaissances.
- ✓ Sprint 2 : initialisation du projet.
- ✓ Sprint 3 : élimination d'erreurs.
- ✓ Sprint 4 : finalisation du projet.

The screenshot shows a JIRA sprint board titled "Tableau Sprint 4" with four columns:

- A FAIRE 19 TICKETS:**
 - GITHUB:
 - FINALISATION DU PROJET**
 - XTRAD-109 (checked)
 - MA
 - GITHUB:
 - FINALISATION DU PROJET**
 - XTRAD-110 (checked)
 - MA
 - vidéo
- EN COURS 1 TICKET:**
 - RWD:
 - FINALISATION DU PROJET**
 - XTRAD-70 (checked)
 - MA
- IN REVIEW 6 TICKETS:**
 - test:
 - FINALISATION DU PROJET**
 - XTRAD-67 (checked)
 - TA
 - test:
 - FINALISATION DU PROJET**
 - XTRAD-68 (checked)
 - TA
 - test
- FINI 1 TICKET:**
 - laravel:
 - FINALISATION**
 - XTRAD-111 (checked)

2. CHAPITRE 2 :

LARAVEL

Dans ce chapitre, nous nous concentrerons sur l'utilisation de Laravel, un framework de développement web, pour la réalisation de notre projet

2.1. Les fichiers de migration :

Les fichiers de migration sont utilisés pour définir et gérer les schémas de base de données. Ils permettent de définir les opérations de création, modification et suppression de tables, de colonnes, d'index, etc. dans une manière versionnée et contrôlée.

- Emplacement des fichiers : Les fichiers de migration se trouvent dans le répertoire *database/migrations* de notre projet XtraDH.
- Création d'un fichier de migration : On génère un nouveau fichier de migration en utilisant la commande artisan *make:migration*. Voici un exemple de la table établissement.



```
php artisan make:migration create_etablissements_table
```

- Structure d'un fichier de migration : Un fichier de migration contient deux méthodes principales : up et down. La méthode *up* définit les opérations à effectuer lors de l'exécution de la migration, tandis que la méthode *down* définit les opérations à effectuer lors du rollback (annulation) de la migration.



```
public function up()
{
    Schema::create('users', function (Blueprint $table) {
        $table->id('id_user');
        $table->string('email')->unique();
        $table->string('password');
        $table->string('type');
        $table->rememberToken();
        $table->timestamps();
    });
}

public function down()
{
    Schema::dropIfExists('users');
}
```

- Opérations courantes : Laravel fournit une variété de méthodes pour effectuer des opérations courantes dans les fichiers de migration. Par exemple, vous pouvez utiliser les méthodes *create*, *table*, *addColumn*, *dropColumn*, *renameColumn*, *dropTable*, etc. pour effectuer des modifications sur la structure de la base de données.
- Les fichiers de migration utilisent différents types de colonnes pour définir la structure des tables de la base de données à savoir : *id*, *string*, *float*, *integer*, *unsignedBigInteger*, *foreign*, ...

2.1.1.

Vue Générale

Les fichiers de migrations présentes dans notre projet XtraDH sont les suivant :

- 2023_05_09_102333_create_etablissements_table.php
- 2023_05_09_103340_create_grades_table.php
- 2023_05_09_105343_create_enseignants_table.php
- 2023_05_09_111323_create_administrateurs_table.php
- 2023_05_09_113128_create_interventions_table.php
- 2023_05_09_121358_create_paiements_table.php

2.1.2.

Table d'établissement



```
Schema::create('etablissements', function (Blueprint $table) {
    $table->id();
    $table->string('code');
    $table->string('Nom');
    $table->string('Telephone');
    $table->string('Fax');
    $table->string('ville');
    $table->integer('Nbr_enseignants')->default(0);
    $table->timestamps();
});
```

Le fichier de migration crée une table "etablissements" avec des colonnes pour stocker des informations telles que le code : Une clé primaire auto-incrémente de type entier, le nom, le numéro de téléphone, le numéro de fax, la ville et le nombre d'enseignants d'un établissement, avec une valeur par défaut de 0. La table inclut également des colonnes pour enregistrer les dates de création et de mise à jour des enregistrements.

2.1.3.

Table grade



```
Schema::create('grades', function (Blueprint $table) {
    $table->id('id_Grade');
    $table->string('designation');
    $table->integer('charge_statutaire');
    $table->float('Taux_horaire_Vocation');
    $table->timestamps();
});
```

Le fichier de migration crée une table "grades" avec des colonnes pour stocker l'id grade qui s'auto-incrémente de type entier, la désignation du grade, la charge statutaire et le taux horaire de vocation. La table inclut également des colonnes pour enregistrer les dates de création et de mise à jour des enregistrements.

2.1.4.

Table enseignants

```
● ● ●
Schema::create('enseignants', function (Blueprint $table) {
    $table->id();
    $table->string('PPR');
    $table->string('Nom');
    $table->string('prenom');
    $table->date('Date_Naissance');
    $table->unsignedBigInteger('Etablissement');
    $table->foreign('Etablissement')->references('id')->on('etablissements')->onDelete('set null');
    $table->unsignedBigInteger('id_Grade');
    $table->foreign('id_Grade')->references('id_Grade')->on('grades')->onDelete('set null');
    $table->unsignedBigInteger('id_user');
    $table->foreign('id_user')->references('id_user')->on('users')->onDelete('set null');
    $table->timestamps();
})
```

Le fichier de migration crée une table "enseignants" qui enregistre les informations des enseignants. Les colonnes incluent le PPR (Numéro d'identification) Paierie Principale des rémunérations de l'enseignant, son nom, prénom, date de naissance, ainsi que des clés étrangères vers d'autres tables telles que 'etablissements', 'grades' et 'users'. Ces clés étrangères sont utilisées pour établir des relations avec les enregistrements correspondants dans les tables référencées. La table 'enseignants' dispose également de colonnes pour enregistrer les dates de création et de mise à jour des enregistrements.

```
● ● ●
$table->foreign('Etablissement')->references('id')->on('etablissements')->onDelete('set null');
```

Ce genre de ligne crée une clé étrangère dans la table "enseignants" qui référence la colonne "id" de la table "etablissements". Cela établit une relation entre les deux tables, où la colonne "Etablissement" dans la table "enseignants" fait référence à la clé primaire "id" de la table "etablissements". L'option onDelete('set null') indique que si la valeur de la clé primaire de la table "etablissements" est supprimée, la valeur de la clé étrangère dans la table "enseignants" sera définie sur NULL. Cela permet de maintenir l'intégrité référentielle lorsque des enregistrements dans la table "etablissements" sont supprimés.

2.1.5.

Table administrateur

```
Schema::create('administrateurs', function (Blueprint $table) {
    $table->id();
    $table->string('PPR');
    $table->string('Nom');
    $table->string('prenom');
    $table->foreignId('Etablissement')->constrained('etablissements', 'id')->onDelete('set null');
    $table->unsignedBigInteger('id_user');
    $table->foreign('id_user')->references('id_user')->on('users')->onDelete('set null');
    $table->timestamps();
});
```

Le fichier de migration crée une table "administrateurs" pour enregistrer les informations des administrateurs. Les colonnes incluent le PPR (Numéro d'identification) de l'administrateur, son nom, prénom, ainsi que des clés étrangères vers d'autres tables telles que 'etablissements' et 'users'. La clé étrangère 'Etablissement' fait référence à la clé primaire 'id' de la table 'etablissements', et l'option 'onDelete('set null')' indique que si la valeur de la clé primaire de la table 'etablissements' est supprimée, la valeur de la clé étrangère dans la table 'administrateurs' sera définie sur NULL. De même, la clé étrangère 'id_user' fait référence à la clé primaire 'id_user' de la table 'users'. La table 'administrateurs' dispose également de colonnes pour enregistrer les dates de création et de mise à jour des enregistrements.

2.1.6.

Table intervention

```
Schema::create('interventions', function (Blueprint $table) {
    $table->id('id_intervention');
    $table->unsignedBigInteger('id_Intervenant');
    $table->foreign('id_Intervenant')->references('id')->on('enseignants')->onDelete('set null');
    $table->unsignedBigInteger('id_Etab');
    $table->foreign('id_Etab')->references('id')->on('etablissements')->onDelete('set null');
    $table->string('Initiale_Intervention');
    $table->string('Annee_univ',10);
    $table->string('Semestre');
    $table->date('Date_debut');
    $table->date('Date_fin');
    $table->integer('Nbr_heures');
    $table->integer('visa_etb')->default(0);
    $table->integer('visa_uea')->default(0);
    $table->timestamps();
});
```

La migration ci-dessus crée une table "interventions" qui enregistre les informations sur les interventions. Les colonnes incluent une clé primaire auto-incrémentée de type entier 'id_intervention'. Il y a également des clés étrangères 'id_Intervenant' et 'id_Etab' qui réfèrent respectivement les clés primaires de la table 'enseignants' et 'etablissements'. Ces clés étrangères permettent de lier les interventions à un intervenant spécifique et à un établissement spécifique.

La table comprend également des colonnes pour enregistrer l'intitulé de l'intervention, l'année universitaire, le semestre, les dates de début et de fin, le nombre d'heures, ainsi que des indicateurs ('visa_etb' et 'visa_ue') qui sont définis par défaut à 0. Enfin, la table enregistre les dates de création et de mise à jour des enregistrements.

2.1.7. *Table paiement*

```
Schema::create('paiements', function (Blueprint $table) {
    $table->id();
    $table->unsignedBigInteger('id_Intervenant');
    $table->foreign('id_Intervenant')->references('id')->on('enseignants')->onDelete('set null');
    $table->unsignedBigInteger('id_Etab');
    $table->foreign('id_Etab')->references('id')->on('etablissements')->onDelete('set null');
    $table->float('VH')->default(0);
    $table->float('Taux_H');
    $table->float('Brut')->storedAs(' "VH" * "Taux_H" ');
    $table->float('IR')->default(0.38);
    $table->float('NET')->storedAs(' "VH" * "Taux_H" * (1-"IR") ');
    $table->string('Annee_univ',10);
    $table->string('Semestre');
    $table->timestamps();
});
```

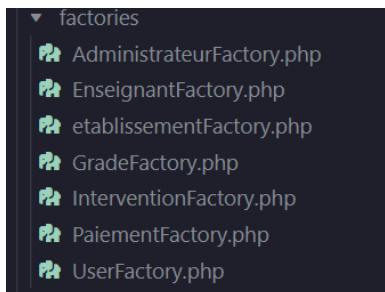
La migration ci-dessus crée une table "paiements".

Les colonnes incluent une clé primaire auto-incrémentée de type entier 'id'. Il y a également des clés étrangères 'id_Intervenant' et 'id_Etab' qui référencent respectivement les clés primaires de la table 'enseignants' et 'etablissements', permettant de lier les paiements à un intervenant correspondant et à un établissement qui lui appartient.

La table comprend également des colonnes pour enregistrer le nombre d'heures travaillées (Le volume horraire)('VH'), le taux horaire ('Taux_H'), le montant brut calculé par une formule déjà imposé, ainsi qu'un taux d'impôt sur le revenu par défaut ('IR') désigné en 0.38 par la suite, et un montant net calculé en déduisant l'impôt sur le revenu du montant brut ('NET'). Les colonnes 'Annee_univ' et 'Semestre' enregistrent respectivement l'année universitaire et le semestre du paiement.

2.2. Factories :

Les factories (usines) sont des classes utilisées pour générer des données factices (dummy data) lors de la création de modèles pour les tests et le développement. Elles fournissent un moyen pratique de créer des enregistrements fictifs dans la base de données, remplissant ainsi les tables avec des données réalistes pour simuler différentes situations. Elles permettent de définir les attributs des modèles de manière aléatoire, facilitant ainsi la création de jeux de données cohérents et variés pour tester les fonctionnalités de notre application XtraDH.



Elles sont accessibles depuis le chemin suivant :

database > factories

Les factories peuvent être utilisées conjointement avec les tests unitaires, les tests fonctionnels ou pour générer des données de démonstration.

2.2.1.

Structure de la classe

```

class AdministrateurFactory extends Factory
{
    protected $model = Model::class;

    public function definition()
    {
        //instanciation

        return [
        ];
    }
}

```

-La classe de factory hérite de la classe Factory fournie par Laravel.

-La propriété \$model est définie sur la classe du modèle correspondant, par exemple Model::class, où Model représente le nom de votre modèle associé à cette factory.

-La méthode definition() est responsable de la définition des attributs du modèle généré par la factory. À l'intérieur de cette méthode, vous pouvez instancier d'autres factories ou créer des instances de modèles liés si nécessaire.

-La méthode retourne un tableau associatif définissant les attributs du modèle généré. Chaque clé du tableau représente un attribut du modèle, et sa valeur peut être définie en utilisant les méthodes du générateur de fausses données (\$faker) ou d'autres valeurs prédéfinies.

2.2.2.

Administrateur Factory

```

protected $model = Administrateur::class;

public function definition()
{
    $user = User::factory()->create();
    $établissement = Etablissement::factory()->create();

    return [
        'PPR' => $this->faker->unique()->numberBetween(100000, 999999),
        'Nom' => $this->faker->lastName,
        'prenom' => $this->faker->firstName,
        'Etablissement' => $établissement->id,
        'id_user' => $user->id_user,
    ];
}

```

La classe *AdministrateurFactory* est une factory spécifique pour le modèle Administrateur. Elle définit les attributs de l'administrateur généré, tels que le PPR, le nom, le prénom, l'établissement et l'identifiant utilisateur.

Par exemple, La ligne '**Nom**' => **\$this->faker->lastName** indique que l'attribut 'Nom' de l'administrateur généré sera rempli avec une valeur générée aléatoirement à l'aide de la méthode **\$faker->lastName** du générateur de fausses données (**\$faker**)

Des instances factices des modèles User et Etablissement sont créées pour obtenir les valeurs nécessaires. Cette factory permet de générer facilement des instances d'administrateurs avec des attributs factices pour les tests ou le développement.

2.2.3.

Enseignant Factory

```

protected $model = Enseignant::class;

public function definition()
{
    $grade = Grade::factory()->create();
    $user = User::factory()->create();
    $etab = établissement::factory()->create();

    return [
        'PPR' => $this->faker->unique()->numberBetween(100000, 999999),
        'Nom' => $this->faker->lastName,
        'prenom' => $this->faker->firstName,
        'Date_Naissance' => $this->faker->dateTimeBetween('1960-01-01', '1994-12-31'),
        'Etablissement' => $etab->id,
        'id_Grade' => $grade->id_Grade,
        'id_user' => $user->id_user,
    ];
}

```

La méthode retourne un tableau associatif qui représente les attributs de l'enseignant générée. Par exemple, le PPR (Numéro d'identification) est généré de manière aléatoire à l'aide de la méthode unique()->numberBetween(100000, 999999) du générateur de fausses données (\$faker). Les autres attributs, tels que le nom, le prénom, la date de naissance entre deux dates, l'établissement, le grade et l'identifiant utilisateur, sont remplis avec les valeurs générées précédemment.

2.2.4. ***Établissement Factory***

```
public function definition()
{
    return [
        'Nom' => $this->faker->company,
        'code' => $this->faker->unique()->randomNumber,
        'Telephone' => $this->faker->phoneNumber,
        'Faxe' => $this->faker->phoneNumber,
        'ville' => $this->faker->city,
        'Nbr_enseignants' => $this->faker->numberBetween(1, 50),
    ];
}
```

2.2.5. ***Grade Factory***

```
public function definition()
{
    return [
        'designation' => $this->faker->randomElement(['Grade A', 'Grade B', 'Grade C']),
        'charge_statutaire' => $this->faker->numberBetween(10, 50),
        'Taux_horaire_Vocation' => $this->faker->numberBetween(10, 50),
    ];
}
```

Dans la factory de grade, la clé 'designation' est remplie avec une valeur aléatoire choisie parmi les options 'Grade A', 'Grade B' et 'Grade C'. Cela permet de générer différentes désignations de grade.

Les clés 'charge_statutaire' et 'Taux_horaire_Vocation' sont remplies avec des valeurs aléatoires générées par les méthodes \$faker->numberBetween(10, 50). Cela permet de générer des valeurs de charge statutaire et de taux horaire qui varient entre 10 et 50.

2.2.6.

Intervention Factory

```

protected $model = Intervention::class;

public function definition()
{
    $enseignant = Enseignant::factory()->create();
    $établissement = Etablissement::factory()->create();
    $date_debut = $this->faker->date();
    $date_fin = $this->faker->date($startDate = $date_debut);

    return [
        'id_Intervenant' => $enseignant->id,
        'id_Etab' => $établissement->id,
        'Intitule_Intervention' => $this->faker->sentence,
        'Annee_univ' => $this->faker->randomElement(['2022', '2023', '2021']),
        'Semestre' => $this->faker->randomElement(['Semestre 1', 'Semestre 2']),
        'Date_debut' => $date_debut,
        'Date_fin' => $date_fin,
        'Nbr_heures' => $this->faker->numberBetween(10, 50),
    ];
}

```

La classe InterventionFactory est une factory spécifique pour le modèle Intervention. La propriété \$model est définie sur la classe Intervention pour indiquer quel modèle est associé à cette factory.

La méthode definition() définit les attributs de l'intervention générée par la factory. À l'intérieur de cette méthode, des instances de la factory des modèles Enseignant et Etablissement sont créées pour obtenir des enseignants et des établissements factices.

Ensuite, des valeurs aléatoires sont générées pour les attributs tels que l'intitulé de l'intervention, l'année universitaire, le semestre, la date de début, la date de fin et le nombre d'heures. Par exemple, l'intitulé de l'intervention est généré à l'aide de la méthode \$faker->sentence, qui crée une phrase aléatoire. L'année universitaire et le semestre sont choisis aléatoirement parmi les options fournies. Les dates de début et de fin sont générées aléatoirement. Le nombre d'heures est généré avec une valeur aléatoire entre 10 et 50 à l'aide de la méthode \$faker->numberBetween(10, 50).

Ainsi, en utilisant cette factory, vous pouvez rapidement créer des instances d'interventions avec des attributs factices (fictifs) pour les tests ou les besoins de développement.

2.2.7.

Paiement Factory

```
protected $model = Paiement::class;

public function definition()
{
    $enseignant = Enseignant::factory()->create();
    $établissement = Etablissement::factory()->create();

    return [
        'id_Intervenant' => $enseignant->id,
        'id_Etab' => $établissement->id,
        'VH' => $this->faker->randomFloat(2, 10, 50),
        'Taux_H' => $this->faker->randomFloat(2, 10, 50),
        'IR' => $this->faker->randomFloat(2, 25, 45),
        'Annee_univ' => $this->faker->randomElement(['2022','2021','2023']),
        'Semestre' => $this->faker->randomElement(['Semestre 1', 'Semestre 2']),
    ];
}
```

De même, on désigne le modèle approprié puis via la fonction `definition` pour définir les attributs générés, et on renvoie un tableau associatif avec des valeurs automatiquement et aléatoirement nées via les différentes méthodes vues en-dessous.

2.3. Models & Relations

Les modèles sont des classes qui représentent des tables de la base de données. Ils sont utilisés pour interagir avec la base de données et effectuer des opérations CRUD (Create, Read, Update, Delete) sur les données.

Les modèles sont généralement situés dans le répertoire *app/Model*.

Ils peuvent également inclure des méthodes personnalisées pour encapsuler la logique métier spécifique au modèle.

Les modèles permettent de définir des relations entre les tables, ces relations facilitent la récupération de données associées à partir d'autres tables et simplifient les opérations de jointure. Laravel propose différents types de relations, tels que les relations "One-to-One" (un à un), "One-to-Many" (un à plusieurs), "Many-to-Many" (plusieurs à plusieurs), etc.

Les relations sont définies dans les modèles à l'aide de méthodes spécifiques telles que `belongsTo`, `hasMany`, `belongsToMany`...etc. Ces méthodes permettent de spécifier la relation entre les modèles et d'accéder aux données associées.

2.3.1. Structure du modèle – Modèle Administrateur

```

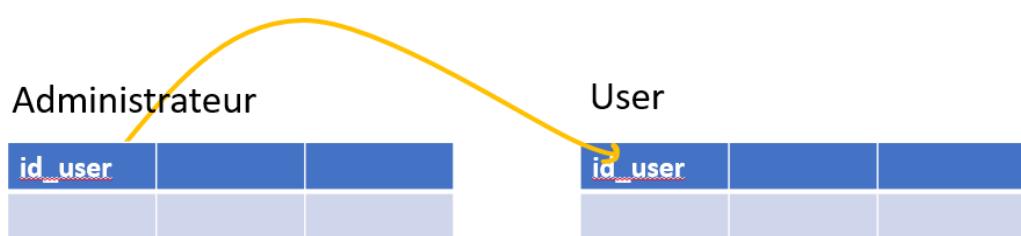
class administrateur extends Model (1)
{
    use HasFactory; <-- (2)
    protected $fillable = [
        'PPR',
        'Nom',
        'prenom',
        'Etablissement',
        'id_user',
    ];
}

public function user(){
    return $this->belongsTo(user::class,'id_user','id_user'); (4)
}

public function etablissement(){
    return $this->belongsTo(etablissement::class,'Etablissement','id'); (5)
}

```

- La classe administrateur hérite des fonctionnalités de base fournies par le modèle, telles que les opérations de base de données et les relations avec d'autres modèles.
 - **HasFactory** permet de créer facilement des instances de la classe administrateur en utilisant la méthode factory().
 - **\$fillable** spécifie les attributs du modèle administrateur qui peuvent être mass assignable, c'est-à-dire ceux qui peuvent être remplis en utilisant la méthode create() ou lors de la mise à jour en utilisant la méthode update() avec un tableau associatif de données. Les attributs mentionnés dans \$fillable sont ceux qui peuvent être affectés en masse et protègent les autres attributs de modification non autorisée.
 - La relation **belongsTo** indique que chaque ligne d'administrateur appartient à un seul utilisateur, le modèle de destination est (user::class) et les clés étrangères utilisées pour établir la relation sont : *id_user* dans le modèle administrateur et *id_user* dans le modèle user.



- De même pour établissement. On s'intéresse à avoir l'id de l'établissement dont l'administrateur dirige, dont les clés étrangères sont **Etablissement** et **id**.

2.3.2.

Modèle Enseignant

Le modèle définit les variables sur lesquelles on traite les fonctions, ainsi que les relations entre les tables.

```
public function etab_permanant(){
    return $this->belongsTo( related: etablissement::class, foreignKey: 'Etablissement', ownerKey: 'id' );
}
public function grade(){
{
    return $this->belongsTo( related: grade::class, foreignKey: 'id_Grade', ownerKey: 'id_Grade' );
}
public function paiement(){
    return $this->hasMany( related: paiement::class, foreignKey: 'id_Intervenant', localKey: 'id', 'enseignant' );
}
public function intervention(){
    return $this->belongsTo( related: intervention::class, foreignKey: 'id', ownerKey: 'id_Intervenant' );
}
no usages
public function interventionEtab(){
    return $this->belongsTo( related: etablissement::class, foreignKey: intervention::class, ownerKey: 'id_Intervenant', relation: 'id_Etab' );
}
public function user(){
    return $this->belongsTo( related: user::class, foreignKey: 'id', ownerKey: 'id_user', relation: 'enseignants' );
}
```

`belongsTo()` est appelée pour définir la relation "*appartient à*" entre le modèle actuel (Enseignant) et les autres modèles.

```
public function etab_permanant(){
    return $this-> belongsTo (etablissement::class,'Etablissement','id');
}
```

La méthode prend alors des arguments : la classe du modèle cible (`etablissement::class`) et la clé étrangère qui relie les deux modèles ('`Etablissement`').

Dans ce cas, la relation est définie en utilisant la colonne '`Etablissement`' dans la table de l'enseignant comme clé étrangère pour relier à la colonne '`id`' de la table `etablissement`.

Enseignant

....	Etablissement

Etablissement

id	

```

public function grade()
{
    return $this->belongsTo(grade::class,'id_Grade','id_Grade');
}

```

De la même manière, la relation est définie en utilisant la colonne 'id_Grade' dans la table de l'enseignant comme clé étrangère pour relier à la colonne 'id_Grade' de la table grade.

La méthode grade() renvoie la relation définie. Cela permet d'accéder au grade associé à un enseignant en utilisant la syntaxe \$enseignant->grade.

L'usage de belongsTo() est du à ce que chaque enseignant possède un et un seul grade, facilitant la récupération des données liées grâce aux méthodes with() ou load() .

```

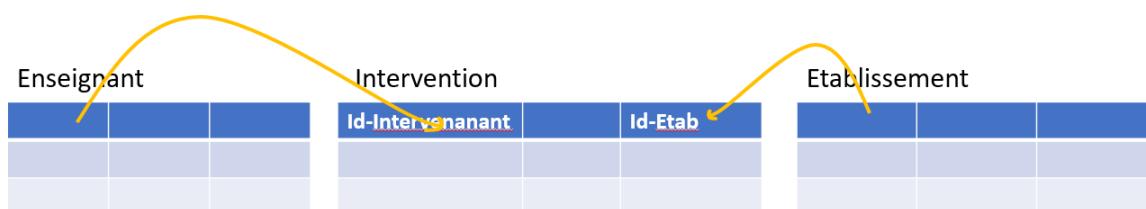
public function intervention(){
    return $this->belongsTo(intervention::class,'id','id_Intervenant');
}
public function interventionEtab(){
    return $this->belongsTo(etablissement::class,intervention::class,'id_Intervenant','id_Etab');
}
public function user(){
    return $this->belongsTo(user::class,'id','id_user','enseignants');
}

```

Le modèle actuel (Enseignant) *appartient* à un modèle intervention. La relation est définie en utilisant la colonne 'id' de la table de l'enseignant comme clé étrangère pour relier à la colonne 'id_Intervenant' de la table intervention. Cette fonction permet d'accéder à l'intervention associée à un enseignant par la syntaxe \$enseignant->intervention.

Le modèle actuel (Enseignant) *appartient* à un modèle etablissement dans le contexte d'une relation intervention. La fonction prend quatre arguments : la classe du modèle cible (etablissement::class), la classe du modèle intermédiaire (intervention::class), la clé étrangère du modèle actuel ('id_Intervenant'), et la clé étrangère du modèle cible ('id_Etab').

La relation est définie en utilisant la colonne 'id_Intervenant' de la table de l'enseignant comme clé étrangère pour relier à la colonne 'id_Etab' de la table etablissement.



public function user(): Cette fonction définit une relation "appartient à" (belongsTo) entre le modèle actuel (Enseignant) et le modèle (user). Elle indique que le modèle actuel appartient à un modèle user.

La relation est définie en utilisant la colonne 'id' de la table de l'enseignant comme clé étrangère pour relier à la colonne 'id_user' de la table user. Cette fonction permet d'accéder à l'utilisateur associé à un enseignant.

```
public function paiement(){
    return $this->hasMany(paiement::class,'id_Intervenant','id','enseignant');
}
```

hasMany() est appelée pour définir la relation "*a plusieurs*" entre le modèle actuel et le modèle paiement. C'est-à-dire qu'il est possible pour chaque enseignant avoir plusieurs paiements.

Elle prend trois arguments : la classe du modèle cible (paiement::class), la clé étrangère du modèle cible qui relie au modèle actuel ('id_Intervenant'), et la clé primaire du modèle actuel ('id'), le quatrième argument ne représente que le nom de la relation.

En bref, la relation est définie en utilisant la colonne 'id_Intervenant' de la table paiement comme clé étrangère pour relier à la colonne 'id' de la table de l'enseignant.

La fonction paiement() renvoie la relation définie. Cela permet d'accéder à toutes les instances du modèle paiements associés à un enseignant spécifique via \$enseignant->paiement.

2.3.3.

Modèle Établissement

```

protected $hidden = ['pivot'];
public function enseignant()
{
    return $this->hasMany(Enseignant::class,'Etablissement','id');
}
public function administrateur()
{
    return $this->hasMany/Administrateur::class,'Etablissement','id');
}
public function intervention()
{
    return $this->hasMany(Intervention::class,'id_Etab','id');
}
public function paiement()
{
    return $this->hasMany(Paiement::class,'id_Etab','id');
}

```

- La propriété `$hidden` est utilisée pour spécifier les attributs cachés lors de la conversion du modèle en tableau/json. Le seul attribut caché est pivot, qui est automatiquement généré par Laravel lors de l'utilisation de la relation "hasMany".
- **enseignant()** : Cette méthode définit une relation "hasMany" entre le modèle actuel et le modèle Enseignant. Elle indique que l'établissement a plusieurs enseignants associés. La clé étrangère utilisée pour la relation est Etablissement dans le modèle Enseignant et id dans le modèle actuel.
- **administrateur()** : Cette méthode définit une relation "hasMany" entre le modèle actuel et le modèle Administrateur. Elle indique que l'établissement a plusieurs administrateurs associés. La clé étrangère utilisée pour la relation est Etablissement dans le modèle Administrateur et id dans le modèle actuel.
- **intervention()** : Cette méthode définit une relation "hasMany" entre le modèle actuel et le modèle Intervention. Elle indique que l'établissement a plusieurs interventions associées. La clé étrangère utilisée pour la relation est id_Etab dans le modèle Intervention et id dans le modèle actuel.
- **paiement()** : Cette méthode définit une relation "hasMany" entre le modèle actuel et le modèle Paiement. Elle indique que l'établissement a plusieurs paiements associés. La clé étrangère utilisée pour la relation est id_Etab dans le modèle Paiement et id dans le modèle actuel.

2.3.4.

Modèle Grade

```
public function enseignant(){
    return $this->hasMany(Enseignant::class,'id_Grade','id_Grade','grades');
}
```

La méthode `enseignant()` établit une relation "un à plusieurs" entre le modèle actuel et le modèle `Enseignant`. Elle spécifie que le modèle `Enseignant` peut avoir plusieurs occurrences associées au modèle actuel. La clé étrangère utilisée est `id_Grade`. Cette méthode permet d'accéder facilement à la liste des occurrences liées au modèle `Enseignant` à partir du modèle actuel.

2.3.5.

Modèle Intervention

```
public function enseignant(){
    return $this->belongsTo(Enseignant::class,'id_Intervenant','id');
}
public function etablissement(){
    return $this->belongsTo(Etablissement::class,'id_Etab','id');
}
```

La méthode `enseignant()` dans ce modèle établit une relation "belongsTo" entre le modèle actuel et le modèle `Enseignant`. Cette relation indique que l'entité `Intervention` appartient à un seul enseignant.

La méthode `etablissement()` établit également une relation "belongsTo", mais cette fois entre le modèle actuel et le modèle `Etablissement`. Cette relation indique que l'entité actuelle appartient à un seul établissement.

2.3.6.

Modèle Paiement

```
public function enseignant(){
    return $this->belongsTo(Enseignant::class,'id_Intervenant','id');
}
public function etablissement(){
    return $this->belongsTo(Etablissement::class,'id_Etab','id');
}
public function grade(){
    return $this->belongsTo(Grade::class,'id_Grade','id');
}
```

La méthode enseignant() établit une relation "belongsTo" entre le modèle actuel et le modèle Enseignant. Elle indique que l'entité actuelle appartient à un seul enseignant.

La méthode etablissement() établit également une relation "belongsTo" entre le modèle actuel et le modèle Etablissement. Elle indique que l'entité actuelle appartient à un seul établissement.

La méthode grade() établit une relation "belongsTo" entre le modèle actuel et le modèle Grade. Elle indique que l'entité actuelle appartient à un seul grade.

2.3.7. *Modèle User*

```
protected $hidden = [
    'password',
    'remember_token',
];
protected $casts = [
    'email_verified_at' => 'datetime',
];

public function enseignant(){
    return $this->hasOne(Enseignant::class,'id_user','id_user');
}
public function administrateur(){
    return $this->hasOne(Administrateur::class,'id_user','id_user','users');
}
```

La propriété \$hidden spécifie les attributs du modèle qui ne doivent pas être inclus lors de la conversion en tableau ou en format JSON, tels que le mot de passe et le jeton de rappel.

La propriété \$casts permet de spécifier les types de données des attributs du modèle, par exemple, la conversion de la colonne 'email_verified_at' en objet DateTime pour une manipulation simplifiée des dates.

La méthode enseignant() établit une relation "hasOne" entre le modèle actuel et le modèle Enseignant, en utilisant la clé étrangère 'id_user'. Cela indique qu'un seul enseignant est associé à l'utilisateur.

La méthode administrateur() établit également une relation "hasOne" entre le modèle actuel et le modèle Administrateur, en utilisant la clé étrangère 'id_user'. Cela indique qu'un seul administrateur est associé à l'utilisateur.

2.4. Contrôleurs

2.4.1. Contrôleur Administrateur

2.4.1.1. *Store*

```
public function store(Request $request)
{
    $attributs = $request->validate([
        'PPR'=>'required',
        'Nom'=>'required',
        'prenom'=>'required',
        'Etablissement'=>'required',
        'id_user'=>'required'
    ]);
    $adm = administrateur::Create($attributs);
    return response()->json($adm);
}
```

Comme explique précédemment dans le contrôleur Enseignant, La fonction *store()* est utilisée pour créer un nouvel administrateur dans la base de données en utilisant les données fournies dans la requête HTTP. Tout d'abord, les attributs requis sont validés à l'aide de la méthode *validate()*, qui s'assure que les champs requis sont présents dans la requête. Ensuite, les attributs validés sont utilisés pour créer une nouvelle instance de modèle administrateur à l'aide de la méthode *Create()*. Enfin, la réponse renvoyée est le nouvel administrateur créé au format JSON.

2.4.1.2. *StoreETB*

```
public function storeETB(Request $request) {

    $validator = Validator::make($request->all(), [
        'PPR' => 'required',
        'Nom' => 'required|max:30',
        'prenom' => 'required|max:30',
        'Etablissement' => 'required',
        'id_user' => 'required',
    ]);

    if ($validator->fails()) {
        return response()->json(['errors' => $validator->errors()], 400);
    }

    $administrateur = Administrateur::create([
        'PPR' => $request->input('PPR'),
        'Nom' => $request->input('Nom'),
        'prenom' => $request->input('prenom'),
        'Etablissement' => $request["Etablissement"],
        'id_user' => $request["id_user"],
    ]);
    return $administrateur;
}
```

Tout d'abord, les champs de la requête sont validés à l'aide du validateur Validator::make() selon des règles de validation spécifiques. Si la validation réussit, un nouvel administrateur est créé en utilisant le modèle Administrateur::create() et les attributs extraits de la requête. Les attributs sont associés aux champs correspondants du modèle Administrateur. Enfin, la réponse renvoyée est le nouvel administrateur créé, au format JSON.

2.4.1.3.

DirecteurETB



```

public function directeurETB(){
    $etb = Administrateur::where('id_user',Auth::user()->id_user)->select('Etablissement')->first()->Etablissement;

    $users = DB::table('administrateurs')
        ->join('users', 'administrateurs.id_user', '=', 'users.id_user')
        ->join('etablissements', 'administrateurs.Etablissement', '=', 'etablissements.id')
        ->select('users.id_user','users.email','users.type',
            'administrateurs.Nom','administrateurs.prenom','administrateurs.PPR','etablissements.Nom as etab_Nom')
        ->where('administrateurs.Etablissement',$etb)
        ->where('users.type','directeur_etab')
        ->get();

    return response()->json($users) ;
}

```

Cette méthode permet d'afficher la liste des directeurs d'établissement qui appartiennent à l'établissement de l'administrateur actuel (etab_permanent).

Tout d'abord, on récupère l'ID de l'établissement à partir du modèle Administrateur en filtrant par l'ID de l'utilisateur actuel. Ensuite, on utilise cette valeur pour joindre les tables administrateurs, users et etablissements en utilisant les clés étrangères correspondantes.

On sélectionne les colonnes nécessaires dans la requête, telles que l'ID de l'utilisateur, l'e-mail, le type d'utilisateur, le nom et prénom de l'administrateur, le PPR (identifiant unique) et le nom de l'établissement.

Ensuite, on applique des conditions pour filtrer les résultats : on vérifie que l'établissement de l'administrateur correspond à l'établissement récupéré précédemment et que le type d'utilisateur est 'directeur_etab'.

Enfin, on retourne les résultats sous forme de réponse JSON.

2.4.1.4.

Show

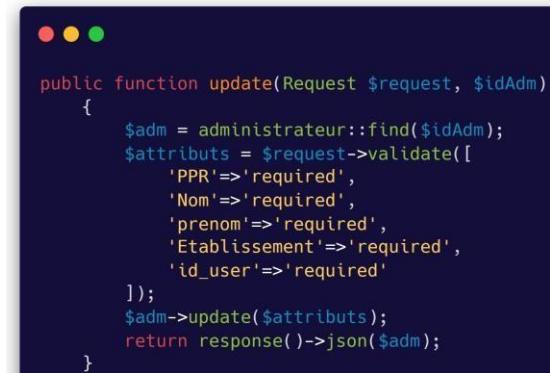


```
public function show($idAdm)
{
    $adm = administrateur::with(['user'])
        ->with(['Etablissement'])
        ->find($idAdm);
    return response()->json($adm);
}
```

La fonction `show()` récupère les détails d'un administrateur spécifique. Elle effectue une recherche dans la base de données en utilisant le modèle `administrateur` et la méthode `find()` pour trouver l'administrateur correspondant à cet identifiant. Ensuite, elle utilise la méthode `with()` pour charger les relations "user" et "Etablissement" liées à l'administrateur afin d'inclure ces informations supplémentaires dans la réponse, et les renvoie au format JSON.

2.4.1.5.

Update



```
public function update(Request $request, $idAdm)
{
    $adm = administrateur::find($idAdm);
    $attributs = $request->validate([
        'PPR'=>'required',
        'Nom'=>'required',
        'prenom'=>'required',
        'Etablissement'=>'required',
        'id_user'=>'required'
    ]);
    $adm->update($attributs);
    return response()->json($adm);
}
```

La fonction `update()` permet de modifier les informations d'un administrateur spécifique en utilisant son identifiant (`$idAdm`). Elle trouve d'abord l'administrateur correspondant dans la base de données avec la méthode `find()`. Ensuite, les attributs de la requête sont validés avec la méthode `validate()` pour s'assurer qu'ils sont présents et conformes aux contraintes. Si la validation réussit, les attributs sont utilisés pour mettre à jour l'administrateur avec la méthode `update()`. Finalement, la fonction renvoie une réponse JSON contenant les détails de l'administrateur mis à jour.

2.4.1.6.

Destroy

La fonction `destroy($id)` est utilisée pour supprimer un administrateur spécifique de la base de données en fonction de son ID. Elle recherche l'administrateur correspondant à l'ID fourni, puis appelle la méthode `delete()` sur l'instance de l'admin pour le supprimer de la base de données. La fonction renvoie `true` si la suppression est réussie et `false` sinon.

2.4.2.

Contrôleur Enseignant

```
class EnseignantController extends Controller{

    public function index(){}
    public function indexETB(){}
    public function store(Request $request){}
    public function storeETB(Request $request){}
    public function show($idens){}
    public function update(Request $request, $id){}
    public function destroy($id){}

}
```

Vue Ensemble

Le contrôleur enseignant est une classe qui hérite de la classe mère Controller contenant des traits. Ces traits fournissent des fonctionnalités communes pour les contrôleurs dans le framework, comme les méthodes authorize(), dispatch(), dispatchNow() et validate().

Ce contrôleur définit plusieurs méthodes :

2.4.2.1.

Index

```
public function index() {
    $enseignant = enseignant::with(['etab_permanant:id,Nom'])
        ->with(['grade'])
        ->with(['intervention'])
        ->with(['user'])
        ->with(['paiement'])
        ->get();
    return response()->json($enseignant);
}
```

La méthode récupère toutes les informations des enseignants, y compris les informations liées à d'autres modèles tels que etab_permanant, grade, intervention, user et paiement.

La méthode `with()` est utilisée pour charger les relations spécifiées définis dans le modèle

Ensuite, il renvoie les données des enseignants sous forme de réponse JSON.

2.4.2.2.

Store

```

public function store(Request $request){
    // Validation des champs
    $validator = Validator::make($request->all(), [
        'PPR' => 'required',
        'Nom' => 'required|max:30',
        'prenom' => 'required|max:30',
        'Date_Naissance' => 'date|required',
        'Etablissement' => 'required',
        'id_Grade' => 'required',
        'id_user' => 'required',
    ]);
    if ($validator->fails()) {
        return response()->json(['errors' => $validator->errors()], 400);
    }

    // Création de l'enseignant
    $enseignant = enseignant::create([
        'PPR' => $request->input('PPR'),
        'Nom' => $request->input('Nom'),
        'prenom' => $request->input('prenom'),
        'Date_Naissance' => date('Y-m-d', strtotime($request->input('Date_Naissance'))),
        'Etablissement' => $request->input('Etablissement'),
        'id_Grade' => $request->input('id_Grade'),
        'id_user' => $request->input('id_user'),
    ]);
    return $enseignant;
}

```

`$request` contenant les données issues des requête HTTP
(préalablement inséré par l'utilisateur généralement via un formulaire)

Un validateur est créé en utilisant la classe Validator de Laravel. Il définit les règles de validation pour tous les champs requis dans les données de la requête.

Si la validation réussit, un nouvel objet enseignant est créé en utilisant le modèle `enseignant::create()`.

Cette méthode statique du modèle enseignant permet de créer une nouvelle objet du modèle et de l'enregistrer dans la base de données avec les données saisi par l'utilisateur via la fonction

`$request->input('')`

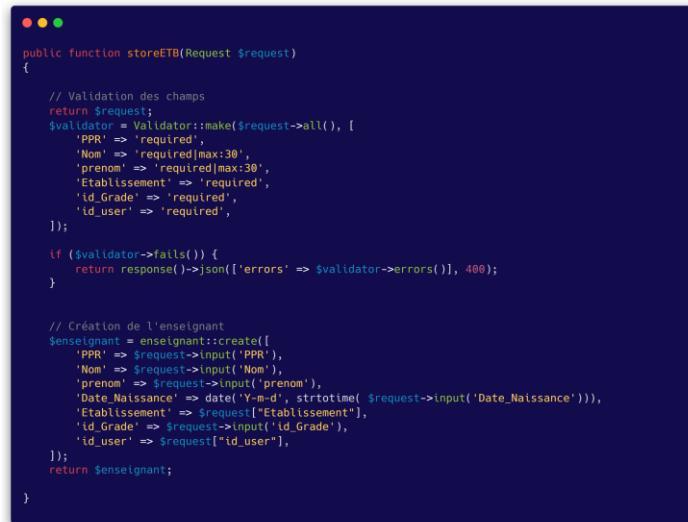
Si les données ne sont pas validées en utilisant la méthode `fails()` qui retourne true, un objet JSON contenant les erreurs de validation est retourné avec un code de statut HTTP 400 (Bad Request). Les erreurs de validation peuvent être obtenues en utilisant la méthode `errors()` de l'objet validateur.

L'objet enseignant nouvellement créé est renvoyé en tant que réponse en format JSON en utilisant la méthode `response()->json()`.

2.4.2.3.

StoreETB

Cette méthode pour stoker des enseignants dans établissement de admin (automatiquement)



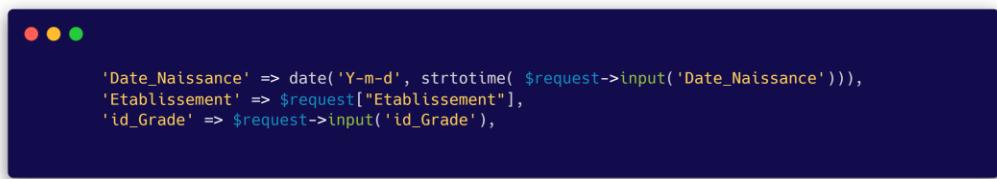
```

public function storeETB(Request $request)
{
    // Validation des champs
    return $request;
    $validator = Validator::make($request->all(), [
        'PPR' => 'required',
        'Nom' => 'required|max:30',
        'prenom' => 'required|max:30',
        'Etablissement' => 'required',
        'id_Grade' => 'required',
        'id_user' => 'required',
    ]);
    if ($validator->fails()) {
        return response()->json(['errors' => $validator->errors()], 400);
    }

    // Création de l'enseignant
    $enseignant = enseignant::create([
        'PPR' => $request->input('PPR'),
        'Nom' => $request->input('Nom'),
        'prenom' => $request->input('prenom'),
        'Date_Naissance' => date('Y-m-d', strtotime($request->input('Date_Naissance'))),
        'Etablissement' => $request->input('Etablissement'),
        'id_Grade' => $request->input('id_Grade'),
        'id_user' => $request->input('id_user'),
    ]);
    return $enseignant;
}

```

La même chose que pour la méthode Store, à l'exception que :



```

'Date_Naissance' => date('Y-m-d', strtotime($request->input('Date_Naissance'))),
'Etablissement' => $request["Etablissement"],
'id_Grade' => $request->input('id_Grade'),

```

La date est stockée en passant par la méthode intégré (date) qui est chargé de la conversion de n'importe quel type vers un type spécifique (le cas de Y-m-d).

Tout le champs établissement est passé pour le stockage.

2.4.2.4.

indexETB



```

public function indexETB(){
    $etb = Administrateur::where('id_user',Auth::user()->id_user)->select('Etablissement')->first()->Etablissement;
    $enseignant = Enseignant::where('Etablissement',$etb)->with(['etab_permanent:id,Nom'])
        ->with(['grade'])
        ->with(['intervention'])
        ->with(['user'])
    ->get();
    return response()->json($enseignant) ;
}

```

La méthode "indexETB" affiche de manière concise la liste des enseignants appartenant à un établissement spécifique. Elle récupère l'ID de l'établissement à partir de la table des administrateurs, puis interroge la table des enseignants pour récupérer les enseignants correspondants. Les informations détaillées telles que l'établissement permanent, le grade, les interventions et les informations utilisateur sont chargées grâce aux relations définies.

2.4.2.5.

Show

```

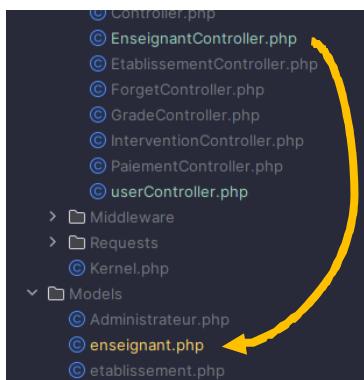
public function show($id_ens)
{
    $ens = enseignant::with(['user'])
        ->with(['grade'])
        ->find($id_ens);
    return response()->json($ens);
}

```

La fonction `show($id_ens)` reçoit l'ID de l'enseignant à afficher en tant que paramètre `$id_ens`, pour retourner les informations de l'enseignant correspondant en format JSON.

La méthode `with(['user'])` et `with(['grade'])` spécifie que la relation "user" et "grade" doit être préchargée avec l'enseignant.

Cela signifie que les données liées à l'utilisateur et au grade de l'enseignant seront incluses dans le résultat.



Le contrôleur appelle le modèle pour récupérer les informations nécessaires de la base de données correspondant à l'ID fourni

La méthode `find($id_ens)` recherche l'enseignant

2.4.2.6.

Update

Méthode pour mettre à jour les informations d'un enseignant spécifique en fonction de son ID, reçoit deux paramètres : un objet Request issu des requêtes HTTP qui contient les nouvelles données mit à jour, et l'ID de l'enseignant à mettre à jour.

```

public function update(Request $request, $id)
{
    $ens = enseignant::where('id',$id)->first();
    $attributs = $request->validate([
        'PPR'=>'',
        'Nom'=>'',
        'prenom'=>'',
        'Date_Naissance' =>'',
        'Etablissement'=>'',
        'id_Grade' => '',
        'id_user'=>''
    ]);
    $ens->update($attributs);
    return response()->json($ens);
}

```

La méthode `where('id', $id)` spécifie que la recherche doit se faire sur le champ 'id' de l'enseignant avec la valeur correspondant à \$id.

La méthode `first()` récupère le premier (un seul) enregistrement correspondant à la requête.

Puis la validation comme déjà expliqué auparavant.

La méthode `update()` applique les modifications sur l'enseignant dans la base de données.

Finalement un retour du même enseignant en format JSON.

2.4.2.7.

Destroy

```

public function destroy($id)
{
    return enseignant::find($id)->delete();
}

```

La fonction `destroy($id)` est utilisée pour supprimer un enseignant spécifique de la base de données en fonction de son ID. Elle recherche l'enseignant correspondant à l'ID fourni, puis appelle la méthode `delete()` sur l'instance de l'enseignant pour le supprimer de la base de données. La fonction renvoie `true` si la suppression est réussie et `false` sinon.

La même logique se répète avec les méthodes de base dans le reste des contrôleur

On se focalisera pour la suite de cette partie sur les autres méthodes spécifiques

2.4.3.

Contrôleur Établissement

```

class EtablissementController extends Controller
{
    public function index() {}
    public function store(Request $request) {}
    public function show($idetab) {}
    public function update(Request $request, $idetab) {}
    public function destroy($idetab) {}
}

```

Méthodes basiques semblables à celles vues précédemment.

2.4.4.

Contrôleur Grade

```

class GradeController extends Controller
{
    public function index() {}
    public function store(Request $request) {}
    public function show($idgrade) {}
    public function update(Request $request, $idgrade) {}
    public function destroy($idgrade) {}
}

```

2.4.5.

Contrôleur Intervention

```

class InterventionController extends Controller
{
    public function index()
    public function interventionuaevalid()
    public function store(Request $request)
    public function storePPR(Request $request)
    public function show( $id)
    public function directeuretabintervall()
    public function directeuretabintervvalid()
    public function getprofIntervention()
    public function update(Request $request, $id)
    public function destroy($id)
    public function valideruae($id)
    public function valideretb($id)
    public function invalideruae($id)
    public function invalideretb($id)
}

```

2.4.5.1.

interventionuaevalid

```
public function interventionuaevalid()
{
    $interventions = DB::table('interventions')
        ->join('enseignants', 'interventions.id_Intervenant', '=', 'enseignants.id')
        ->join('etablissements', 'etablissements.id', '=', 'enseignants.Etablissement')
        ->where('interventions.visa_etb', 1)
        ->where('interventions.visa_uea', 1)
        ->select('id_intervention', 'Intitule Intervention',
            'Annee_univ', 'Semestre', 'Date_debut', 'Date_fin', 'Nbr_heures',
            'visa_etb', 'visa_uea', 'enseignants.Nom as prof_Nom', 'enseignants.prenom', 'etablissements.Nom as Nom_etb')
        ->get();
    return response()->json($interventions);
}
```

La méthode "interventionuaevalid" effectue une jointure entre les tables "interventions", "enseignants" et "etablissements" pour récupérer les informations pertinentes. Les interventions qui ont été approuvées à la fois par l'établissement et l'université sont sélectionnées à l'aide des conditions "visa_etb" et "visa_uea" égales à 1.

Les colonnes spécifiques telles que l'identifiant de l'intervention, l'intitulé, l'année universitaire, le semestre, les dates, le nombre d'heures, les statuts de visa, les noms des enseignants et le nom de l'établissement sont renvoyées sous forme de réponse JSON.

2.4.5.2.

directeuretabintervall – directeuretabintervalid

```
public function directeuretabintervall()
{
    $user = Auth::user();
    $etb = Administrateur::where('id_user', Auth::user()->id_user)->select('Etablissement')->first()->Etablissement;
    $intervention = DB::table('interventions')
        ->join('enseignants', 'interventions.id_Intervenant', '=', 'enseignants.id')
        ->join('etablissements', 'etablissements.id', '=', 'enseignants.Etablissement')
        ->where('etablissements.id', '$etb')
        ->select('id_intervention', 'Intitule Intervention', 'Annee_univ',
            'Semestre', 'Date_debut', 'Date_fin', 'Nbr_heures', 'visa_etb',
            'enseignants.Nom as prof_nom', 'enseignants.prenom')
        ->orderBy('id_intervention')
        ->get();
    return $intervention;
}
```

La méthode "directeuretabintervall" récupère de manière concise la liste des interventions liées à un établissement spécifique, qui est associé à l'utilisateur actuellement authentifié en tant que directeur. Elle effectue une jointure entre les tables "interventions", "enseignants" et "etablissements" pour récupérer les informations pertinentes. Les interventions sont filtrées en fonction de l'identifiant de l'établissement.

Les colonnes spécifiques telles que l'identifiant de l'intervention, l'intitulé, l'année universitaire, le semestre, les dates, le nombre d'heures, le statut de visa, le nom et le prénom des enseignants sont sélectionnées et renvoyées dans l'ordre croissant de l'identifiant de l'intervention. La liste des interventions est ensuite retournée.

La fonction *directeuretabintervalid* ajoute une couche conditionnelle, il s'agit des interventions validées par l'établissement où l'indicateur *visa_etb* = 1.

2.4.5.3.

Getprofintervention

```

public function getprofIntervention()
{
    $user = Auth::user();
    $intervention = DB::table('interventions')
        ->join('enseignants', 'interventions.id_Intervenant', '=', 'enseignants.id')
        ->join('etablissements', 'etablissements.id', '=', 'enseignants.Etablissement')
        ->where('enseignants.id_user', $user->id_user)
        ->where('interventions.visa_etb', 1)
        ->where('interventions.visa_uea', 1)
        ->select('Intitule_Intervention', 'Annee_univ', 'Semestre', 'Date_debut',
            'Date_fin', 'etablissements.Nom as etab', 'Nbr_heures')
        ->get();
    return $intervention;
}

```

La méthode "getprofIntervention" récupère la liste des interventions associées à un enseignant spécifique. Elle effectue une jointure entre les tables "interventions", "enseignants" et "etablissements" pour récupérer les informations pertinentes. Les interventions sont filtrées en fonction de l'identifiant de l'utilisateur actuellement authentifié en tant que professeur, et elles doivent également avoir le statut de visa approuvé à la fois par l'établissement et l'uae.

2.4.5.4.

valideruae – valideretb

```

public function valideruae($id)
{
    $intervention = Intervention::where('id_intervention', $id)->first();
    $intervention->visa_uea = 1 ;
    $intervention->update();
    return $intervention ;
}
public function valideretb($id)
{
    $intervention = Intervention::where('id_intervention', $id)->first();
    $intervention->visa_etb = 1 ;
    $intervention->update();
    return $intervention ;
}

```

Les deux méthodes valident respectivement l'indicateur de validation de l'intervention courante par l'université et par l'établissement, en cherchant l'intervention via son id passé en paramètre par la méthode where et on récupère une seule ligne du résultat, puis affectation à l'indicateur la valeur de 1 et retourner l'intervention en question.

2.4.5.5.

invalideruae – invalideretb

```

public function invalideruae($id)
{
    $intervention = Intervention::where('id_intervention',$id)->first();
    $intervention->visa_ue = 0 ;
    $intervention->update();
    return $intervention ;
}
public function invalideretb($id)
{

    $intervention = Intervention::where('id_intervention',$id)->first();
    $intervention->visa_etb = 0 ;
    $intervention->update();
    return $intervention ;
}

```

De la même logique que les deux méthodes précédentes, sauf que cette fois on désactive l'indicateur en mettant une valeur nulle.

2.4.6.

Contrôleur Paiement

Vue d'ensemble :

```

class PaiementController extends Controller
{
    public function index()
    public function store(Request $request)
    public function show( $id)
    public function update(Request $request,  $id)
    public function destroy( $id)
    public function postfix()
    public function generatePDFprof($id_paiement)
    public function consultpaiementetabdirecteur()
    public function paiementprof()
    public function historiquepdfpaie()
}

```

2.4.6.1. *postfix*

```

public function postfix(){
    $postfix = [];

    $mois = date('n');
    if($mois > 06){
        $avant = date("Y");
        $apres = date("Y") + 1;
    } else{
        $avant = date("Y") - 1;
        $apres = date("Y");
    }

    $date = $avant.'-'.$apres;

    //dd($date);
    $paiements = Paiement::select('id','id_Intervenant')->where('Annee_univ',$date)->get('id');

    foreach($paiements as $paie){
        $enseignant = enseignant::where('id',$paie->id_Intervenant)
            ->with(['user:id_user,email'])
            ->first();
        $email = $enseignant->user->email;
        $url = '/api/generate-pdf/'.$paie->id;
        $nom = $enseignant->Nom;
        $prenom = $enseignant->prenom;
        $user = ['nom'=>$nom,'prenom'=>$prenom,'email'=>$email,'url'=>$url];
        array_push($postfix,$user);
    }

    return response()->json($postfix);
}

```

Cela détermine l'année universitaire en fonction du mois courant. Si le mois est supérieur à 06 (juin)

Exemple : Septembre 2021 : Année 2021/2022

récupération tous les paiements ayant l'année universitaire correspondante.

Pour chaque paiement, elle recherche l'identifiant d'enseignant. Puis l'adresse e-mail de l'enseignant. Elle construit également une URL pour générer le fichier PDF du paiement. Les détails tels que le nom, le prénom, l'e-mail et l'URL sont ajoutés à un tableau associatif et poussés

2.4.6.2. *consultpaiementetabdirecteur*

```

public function consultpaiementetabdirecteur()
{
    $user = Auth::user();
    $etb = Administrateur::where('id_user',Auth::user()->id_user)->select('Etablissement')->first()->Etablissement;
    $mois = date('n');
    if($mois > 06){
        $avant = date("Y");
        $apres = date("Y")+1;
    }
    else{
        $avant = date("Y")-1;
        $apres = date("Y");
    }

    $date = $avant.'-'.$apres ;
    $paiement = DB::table('paiements')
        ->join('enseignants','paiements.id_Intervenant','=','enseignants.id')
        ->where('id_Etab',$etb)
        ->where('Annee_univ',$date)
        ->get();
    return $paiement;
}

```

Cette méthode permet à l'établissement de consulter pour l'année universitaire en cours.

Elle commence par récupérer l'utilisateur actuellement connecté en utilisant la fonction "Auth::user()". Ensuite, elle trouve l'ID de l'établissement associé à cet utilisateur dans la table "Administrateur". Elle récupère l'année universitaire en fonction du mois courant, de la même manière que précédemment.

Elle effectue par la suite une jointure entre la table "paiements" et la table "enseignants" en utilisant l'identifiant de l'intervenant. Elle filtre les paiements en fonction de l'ID de l'établissement et de l'année universitaire.

Enfin, elle récupère tous les enregistrements correspondants et les renvoie en tant que résultat.

2.4.6.3. *paiementprof*

```

public function paiementprof()
{
    $user = Auth::user();
    $ens = Enseignant::where('id_user',$user->id_user)->first();
    $mois = date('n');
    if($mois > 06){
        $avant = date("Y");
        $apres = date("Y")+1;
    }
    else{
        $avant = date("Y")-1;
        $apres = date("Y");
    }

    $date = $avant.'-'.$apres ;
    $paiement = DB::table('paiements')
        ->join('etablissements','id_Etab','=','etablissements.id')
        ->where('id_Intervenant',$ens->id)
        ->where('Annee_univ',$date)
        ->get();
    return $paiement;
}

```

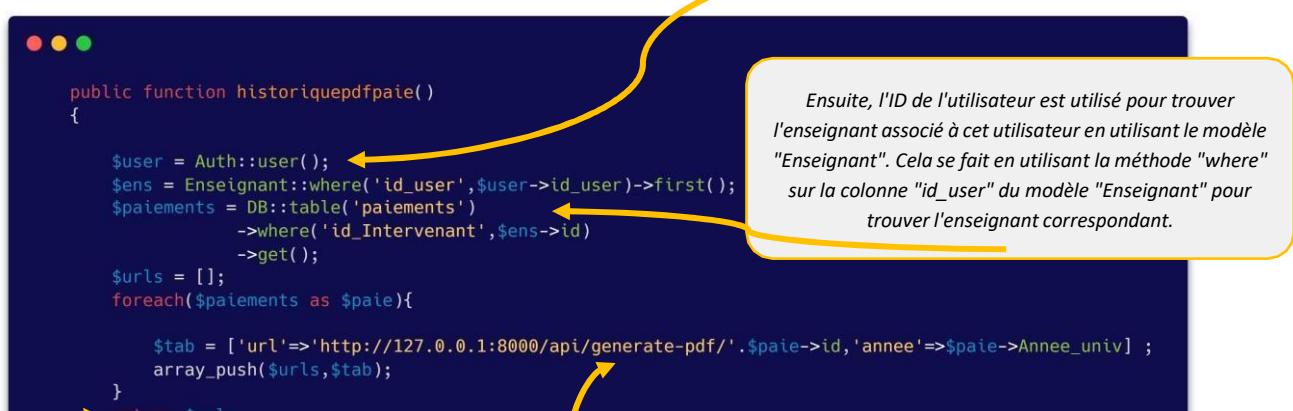
Cette méthode permet à un enseignant de consulter ses propres paiements pour l'année universitaire en cours.

La méthode effectue une jointure entre la table "paiements" et la table "etablissements" en utilisant l'identifiant de l'établissement. Elle filtre les paiements en fonction de l'ID de l'enseignant et de l'année universitaire.

2.4.6.4. ***Historiquepdfpaie***

La méthode "historiquepdfpaie" est utilisée pour générer l'historique des paiements d'un enseignant sous forme de fichiers PDF téléchargeables.

Tout d'abord, l'utilisateur actuellement connecté est récupéré en utilisant la fonction "Auth::user()". Cela permet d'obtenir les informations de l'utilisateur authentifié.



```

public function historiquepdfpaie()
{
    $user = Auth::user();
    $ens = Enseignant::where('id_user', $user->id_user)->first();
    $paiements = DB::table('paiements')
        ->where('id_Intervenant', $ens->id)
        ->get();
    $urls = [];
    foreach($paiements as $paie){
        $tab = [ 'url'=>'http://127.0.0.1:8000/api/generate-pdf/' . $paie->id, 'annee'=>$paie->Annee_univ] ;
        array_push($urls,$tab);
    }
    return $urls;
}

```

Ensuite, l'ID de l'utilisateur est utilisé pour trouver l'enseignant associé à cet utilisateur en utilisant le modèle "Enseignant". Cela se fait en utilisant la méthode "where" sur la colonne "id_user" du modèle "Enseignant" pour trouver l'enseignant correspondant.

Ensuite, la méthode itère sur chaque paiement dans la variable "\$paiements". Pour chaque paiement, elle construit une URL de téléchargement pour le fichier PDF correspondant en utilisant l'ID du paiement. L'URL est sous la forme "<http://127.0.0.1:8000/api/generate-pdf/{id}>" où "{id}" est l'ID du paiement. De plus, l'année universitaire du paiement est également associée à l'URL.

La méthode renvoie le tableau "\$urls", qui contient toutes les URLs des fichiers PDF d'historique de paiement pour l'enseignant spécifié.

2.4.7. Contrôleur User

Ce contrôleur est le contrôleur responsable de gérer les comptes des utilisateurs d'application et de contrôler aussi les autres contrôleurs Comme Celui d'Enseignant et Celui d'Administrateur.

2.4.7.1. Index

```
public function index()
{
    return response()->json(User::all());
```

Cette méthode est responsable d'apporter la liste des comptes d'utilisateurs et de retourner cette liste sous la forme JSON. Pour cela il fait appel à la méthode static de la classe user: all() qui retournera tout la liste des utilisateurs.

2.4.7.2. storeProfETB

```
public function storeProfEtb(Request $request)
{
    //cette méthode pour stocker des enseignants dans établissement de administrateur (automatiquement)
    $fields = $request->validate([
        'email' =>'required|email|unique:users,email',
        'type'=>'required'
    ]);
    // $fields['password']=Str::random(15);
    $fields['password'] = "1234";
    $user = User::create([
        'type' =>$fields['type'],
        'email' => $fields['email'],
        'password'=>bcrypt($fields['password'])
    ]);

    $email = $fields['email'];
    Mail::send('Mails.password',[ 'password'=>$fields['password']],function(Message $message)use($email){
        $message->to($email);
        $message->subject('Voici le mot de passe de votre compte hsup');
    });
    $token = $user->createToken('MyAppToken')->plainTextToken;
    $request['id_user'] = $user->id_user;
    $request['Etablissement'] = Administrateur::where('id_user',Auth::user()->id_user)->select('Etablissement')->first()->Etablissement;
    //create enseignant
    $ensctrl = new EnseignantController();
    $ensctrl = $ensctrl->storeETB($request);
    $response= [
        'user'=>$user,
        'token' =>$token,
        'enseignant'=>$ensctrl
    ];
    return response($response,202);
}
```

Cette méthode est responsable d'enregistrer des enseignants à la fois dans les deux tables: users et enseignant avec une particularité est que établissement de l'enseignant sera celle de administrateur qui va l'insérer.

Pour cela on valide les données de la requête avec la méthode validate de la classe Request (email: doit être obligatoire, de type email , et unique dans la table users || type : est

PROJET XTRADH

obligatoire). Ensuite le mot de passe de utilisateur sera généré aléatoirement à l'aide de la méthode statique random() de la classe Str.

Après on insère les données validées dans la table users à l'aide de la méthode statique create (), on donne à cette méthode email, le type ainsi que le mot de passe crypté à l'aide de la méthode bcrypt() (le choix de la méthode bcrypt est fait en collaboration avec le responsable de sécurité)

Etape suivant consiste à envoyer le mot de passe aléatoire généré par la méthode random () à l'email inséré par Administrateur à l'aide de la méthode send() de la classe Mail , le premier argument donnée à cette méthode est le fichier Template de email (qui se trouve dans ce cas dans le chemin views/Mails/password.blade.php), le second argument est le mot de passe transféré depuis ce contrôleur vers la Template password , le dernier argument est une call back fonction qui utilise comme argument, un argument de type Message \$message pour exécuter les 2 méthode to() et subject() :

La méthode to () : responsable de préciser à qui cet email sera envoyé, tandis que la méthode subject () est responsable de préciser l'objet d'email.

La prochaine étape est de créer un token qui sert à vérifier la validité d'authentification d'utilisateur dans notre application.

Étape suivante : est de pouvoir créer aussi enseignant dans la table enseignants pour cela il faut profiter de la méthode storeETB() développée dans le contrôleur enseignant. Et alors on créé une instance de type EnseignantController. Cette instance fait appel à la méthode storeETB() en fournissant comme argument la variable \$request modifié (la modification de \$request consiste à ajouter un champ d'établissement récupéré de l'utilisateur connecté à l'aide de: Administrateur::where('id_user',Auth::user()->id_user->select('Etablissement')->first()->Etablissement ,

L'autre modification de la variable \$request consiste à ajouter id_user d'utilisateur créé précédemment pour faire la liaison entre les deux tables enseignants et user.

Et finalement on retourne les différents comptes créés avec le token et un statut code 202 pour informer la partie Front end que l'opération a été faite avec succès.

2.4.7.3.

storeAdminETB

Cette méthode est exactement pareille à la méthode storeProfEtb() , le seule différence est que l'insertion se fait dans la table administrateurs.

```
● ● ●
public function storeAdmEtb(Request $request)
{
    //cette methode pour storer des administrateur etablissement dans etablissement de administrateur (automatiquement)

    $fields = $request->validate([
        'email' =>'required|email|unique:users,email',
        'type'=>'required'
    ]);
    // $fields['password']=Str::random(15);
    $fields['password'] = $request->password ;
    $user = User::create([
        'type' =>$fields['type'],
        'email' => $fields['email'],
        'password'=>bcrypt($fields['password'])
    ]);

    $email = $fields['email'];
    Mail::send('Mails.password ',['password'=>$fields['password']],function(Message $message)use($email){
        $message->to($email);
        $message->subject('Voici le mot de passe de votre compte hsup');
    });
    $token = $user->createToken('MyAppToken')->plainTextToken;
    $request['id_user'] = $user->id_user;
    $request['Etablissement'] = Administrateur::where('id_user',Auth::user()->id_user)->select('Etablissement')->first();
    //Etablissement;
    //create enseignant

    $admctrl = new AdministrateurController();
    $admctrl = $admctrl->storeETB($request);
    $response= [
        'user'=>$user,
        'token' =>$token,
        'admin'=>$admctrl
    ];
    return response($response,202);
}
```

2.4.7.4.

updateProf

```
● ● ●
public function updateprof(Request $request,$id){

    $user = User::where('id_user',$id)->first();
    // $user->update(['email'=>$request->email,'password'=>bcrypt($request->password)]);
    if($request->email)
        $user->email = $request->email ;
    if($request->password)
        $user->password = bcrypt($request->password);
    $user->save();

    //update enseignant
    $id = Enseignant::where('id_user',$user->id_user)->first();
    $ensctrl = new EnseignantController();
    return $ensctrl = $ensctrl->update($request,$id->id);
}
```

Cette méthode est responsable de la modification à la fois les données d'un professeur dans la table user et enseignant.

Pour cela on récupère l'utilisateur \$user à l'aide de la méthode statique where () qui effectue un recherche dans la table users et renvoi l'objet dont id_user = \$id, après on vérifier si un email ou mot de passe est donné pour le faire modifié à l'aide de la méthode save(),

PROJET XTRADH

Ensuite il faut changer les données de « enseignant » aussi.

Pour cela on effectue une recherche pour récupérer id enseignant concerné par l'opération de modification, ensuite on crée une instance de classe EnseignantController où on utilise la méthode update() en fournissant comme paramètre \$request et \$id : id de l'enseignant concerné , cette méthode retournera 1 en cas de succès

2.4.7.5. *updateAdm*

```
public function updateAdm(Request $req,$id){  
    // $id = $req->id;  
    $adm = user::where('id_user',$id)->first();  
    if($req->email)  
        $adm->email = $req->email ;  
    if($req->password)  
        $adm->password = bcrypt($req->password);  
    $adm->save();  
    $id=Administrateur::where('id_user',$adm->id_user)->first();  
    $adm_cntrol = new AdministrateurController();  
    return $adm_cntrol->update($req,$id->id);  
}
```

Cette méthode a la même logique que celle de updateProf.

2.4.7.6. *Show*

```
public function show($id )  
{  
    return response()->json(  
        user::with(['administrateur'])  
            ->with(['enseignant'])  
            ->find($id)  
    );  
}
```

Cette méthode est une méthode de test proposé par contrôleur de ressource, utilité de cette méthode est qu'elle permet d'afficher un utilisateur avec son compte soit administrateur soit enseignants.

2.4.7.7. *Update*

```
public function update(Request $request, $id)  
{  
    $fields = $request->validate([  
        'email' =>'email',  
        'password' =>'string',  
        'type'=>'string'  
    ]);  
  
    $user = user::find($id);  
    $user->update($fields);  
    // $token = $user->createToken('MyAppToken')->plainTextToken;  
    $response= [  
        'user'=>$user,  
        'token' =>Auth::user()->tokens()  
    ];  
    return response($response,202);  
}
```

PROJET XTRADH

Cette méthode est utilisée dans le test de modification des attributs du modèle user sur Postman.

La logique de cette méthode est qu'il valide les donné reçu dans la requête en suite partir de id donnée dans l'URL elle cherche la ligne dans la table user concerner ensuite ORM transforme cette ligne en un objet de type du model user ensuit un fait la modification convenable l'aide de la méthode update () est on retourne le nouvel utilisateur.

2.4.7.8. *Updateprof*

```
public function updateprof(Request $request,$id){

    $user = User::where('id_user',$id)->first();
    // $user->update(['email'=>$request->email,'password'=>bcrypt($request->password)]);
    if($request->email)
        $user->email = $request->email ;
    if($request->password)
        $user->password = bcrypt($request->password);
    $user->save();

    //update enseignant
    $id = Enseignant::where('id_user',$user->id_user)->first();
    $ensctrl = new EnseignantController();
    return $ensctrl = $ensctrl->update($request,$id->id);
}
```

Cette méthode permet de modifier à la fois un utilisateur dans la table user et un professeur dans la table enseignant. La logique de cette méthode est que on reçoit les données dans la requête et id de la personne concernée dans URL ensuite avec ce dernier on cherche le 'user' concerné puis on effectue un test pour voir si ce que \$request contient un champ email et password (utilité de ce test est de recrypté le mot de passe en cas de modification). Ensuite avec le id_user on cherche enseignant concerné on fait appel au constructeur de la classe EnseignantController pour créer un objet nommé ensctrl, avec ce dernier on fait appel à la méthode proposée par le Contrôleur de ressource update() pour modifier les champs affecté par la modification dans \$request.

2.4.7.9. *updateAdm*

```
public function updateAdm(Request $req,$id){
    // $id = $req->id;
    $adm = user::where('id_user',$id)->first();
    if($req->email)
        $adm->email = $req->email ;
    if($req->password)
        $adm->password = bcrypt($req->password);
    $adm->save();
    $id=Administrateur::where('id_user',$adm->id_user)->first();
    $adm_ctrl = new AdministrateurController();
    return $adm_ctrl->update($req,$id->id);
}
```

PROJET XTRADH

Cette méthode est exactement pareille à la méthode précédente sauf que la modification ce fait ici dans la table administrateurs.

2.4.7.10. *profileProfile*

```
● ● ●
public function profProfile()
{
    $etb = Enseignant::where('id_user',Auth::user()->id_user)->select('Etablissement')->first()->Etablissement;
    $user = Auth::user();
    $user = DB::table('enseignants')
        ->join('users', 'enseignants.id_user', '=', 'users.id_user')
        ->join('etablissements', 'enseignants.Etablissement', '=', 'etablissements.id')
        ->select('users.id_user','users.email','users.type',
    'enseignants.Nom','enseignants.prenom','enseignants.PPR','etablissements.Nom as etab_Nom')
        ->where('enseignants.Etablissement',$etb)
        ->where('users.id_user',$user->id_user)
        ->first();
    return response()->json($user) ;
}
```

Cette méthode permet de retourner les information concerné dun profile de type prof.

La logique de cette méthode est que on cherche on va effectuer commencer à construire des requête manuellement pour de raison de simplification, premièrement on va effectuer un jointure de la table enseignant avec les tables user et établissements ensuite un effectue un sélection des champs souhaité à l'aide de la méthode select puis on filtre les donné retourné par d'user et on retourne la première ligne retourné (normalement cette requête retournera une seule ligne mais si on utilise ->get() celle retournera un tableau avec un seul élément et on veut éviter d'écrire ->get[0] , et alors on utilisera le ->first() pour avoir que le premier élément retourné)

Et finalement on retournera les données trouvées

2.4.7.11. *adminProfil*

Cette méthode est exactement pareille que la méthode précédente à l'exception qu'elle effectue la recherche dans la table administrateurs.4

2.4.7.12. *ajouterinterventiontab*

```
● ● ●
public function ajoutinterventiontab(Request $request)
{
    $user = Auth::user();
    $etb = administrateur::where('id_user',$user->id_user)->select('Etablissement')->first();
    $request["id_etab"] = $etb->Etablissement;
    $ensctrl = new InterventionController();
    return $ensctrl = $ensctrl->storePPR($request);
}
```

Cette méthode permet d'ajouter une intervention avec récupération automatique d'établissement d'administrateur connecté. La logique de cette méthode est qu'elle injecté cet établissement récupéré dans la Requête \$request (qui au moment d'envoi ne contient pas établissement encore) ensuite ont déclenché un appel au constructeur de la classe InterventionController pour instancier ce \$ensctrl ensuite on utiliser la méthode storePPR () (cette méthode permet d'ajouter une intervention à un prof en lui fournir le PPR comme identifiant du professeur).

2.4.7.13. ***destroyprof***

```
public function destroyprof($id){
    $user = User::find($id);
    $ens = Enseignant::where('id_user',$user->id_user)->first()->delete();
    return $user->delete();
}
```

Cette méthode permet de supprimer un professeur automatiquement dans les 2 tables (user et enseignants). La logique de cette méthode est que on cherche utilisateur concerné par la suppression à l'aide de \$id (c'est un id_user) et on cherche enseignant qui a cet id dans le champ id_user on fait appel à la méthode ->delete () pour supprimer enseignant et utilisateur (attention ordre de suppressions très important ici)

2.4.7.14. ***destroyAdmin***

```
public function destroyAdmin($id){
    $user = User::find($id);
    $ens = Administrateur::where('id_user',$user->id_user)->first()->delete();
    return $user->delete();
}
```

Cette méthode est exactement pareille à la méthode précédente sauf que la suppression ici se fait dans la table administrateurs.

2.4.7.15. ***destroy***

```
public function destroy($id)
{
    return User::find($id)->delete();
}
```

Cette méthode permet de supprimer un compte utilisateur, cette méthode est utilisée des test de suppression d'utilisateur sur Postman, la logique de cette méthode est qu'elle cherche utilisateur concerné par la suppression à l'aide de id et elle fait appel à la méthode delete().

2.4.8. **Contrôleur Forget**

Vue Ensemble :

```
class ForgetController extends Controller
{
    public function forgot(ForgotPasswordRequest $request) {}
    public function reset(ResetRequest $request) {}

}
```

Ce contrôleur est le responsable de la procédure de récupération du mot de passe des utilisateurs en tout sécurité.

2.4.8.1. ***forgot***

```
public function forgot(ForgotPasswordRequest $request){
    $email = $request->input('email');
    if(User::where('email',$email)->doesntExist()){
        return response([
            'message'=>'L\'utilisateur n\'existe pas'
        ],400);
    }
    $token = Str::random(30);
    DB::table('password_resets')->insert([
        'email'=>$email,
        'token'=>$token
    ]);

    //send email
    Mail::send('Mails.forgot',[ 'token'=>$token],function(Message $message)use($email){
        $message->to($email);
        $message->subject('Réinitialiser votre mot de passe ');
    });

    try{
        return response([
            'message'=>'Consulter votre boîte mail'
        ]);
    }catch(\Exception $exception){
        return response([
            'message'=>$exception->getMessage()
        ],400);
    }
}
```

Cette méthode permet d'envoyer un email de récupération de mot de passe. La logique de cette méthode est : on cherche l'existence d'email récupéré dans la requête dans la base de

PROJET XTRADH

données s'il existe on génère un token aléatoire à l'aide de la méthode Str::random(30) qui générera un string de longueur 30 caractères. Ensuite on inscrit ce token et email trouvé dans la table password_resets

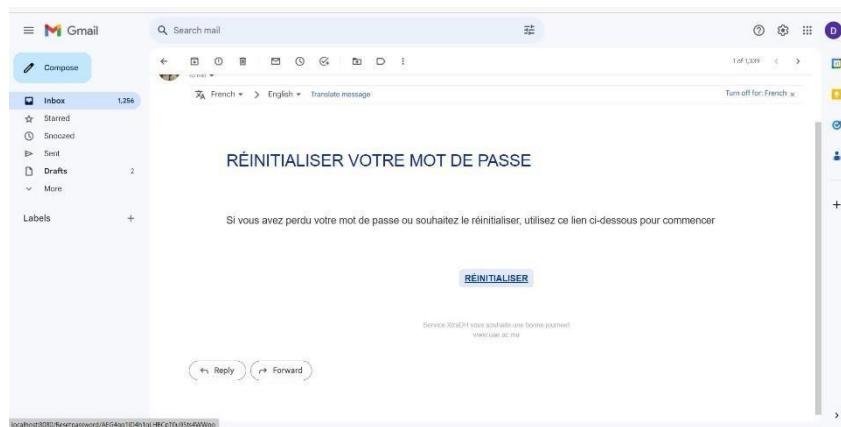
Cela va nous permettre de vérifier la validité du token lors de récupération du nouveau mot de passe

Ensuite on envoie un email de réinitialisation du mot de passe avec le token créé.

Si utilisateur n'existe pas

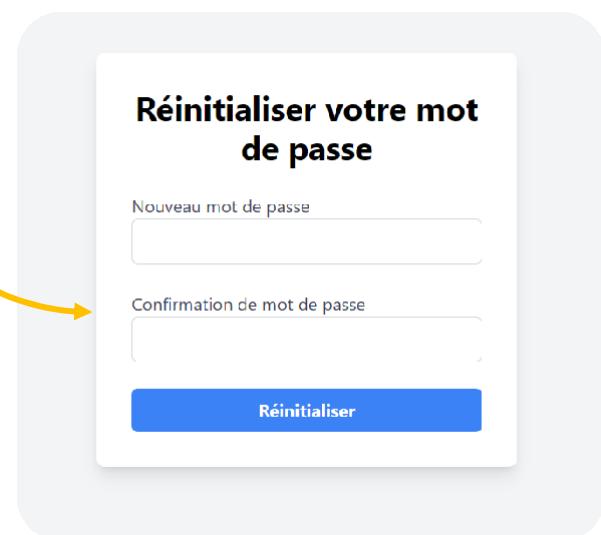


Si utilisateur existe

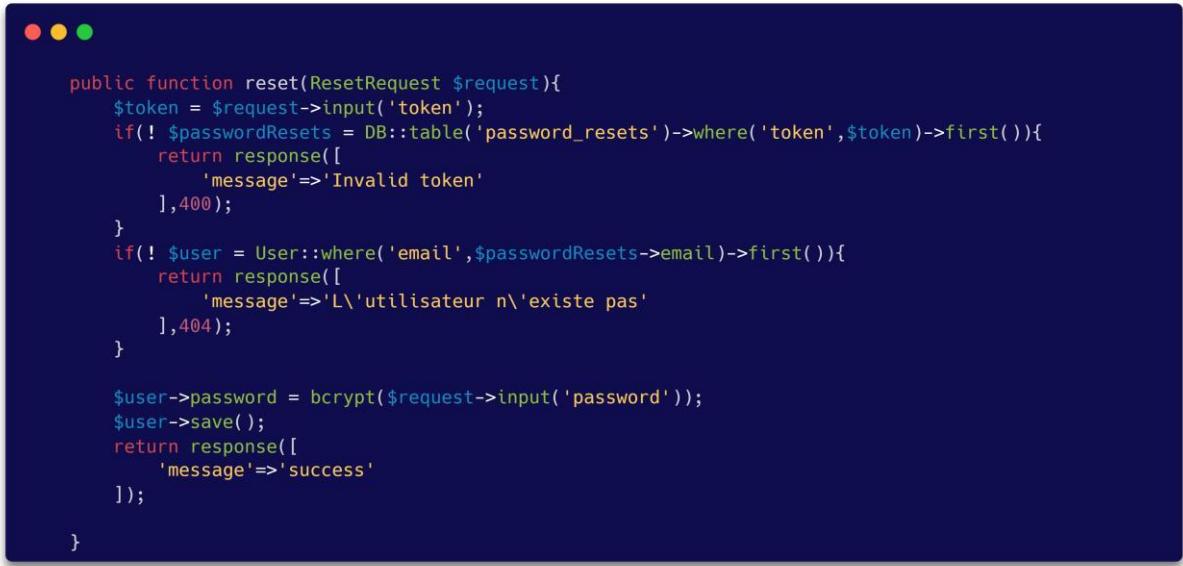


token

Après avoir cliqué dans ce lien il sera redirigé vers cette page :



2.4.8.2.

reset


```

public function reset(ResetRequest $request){
    $token = $request->input('token');
    if(! $passwordResets = DB::table('password_resets')->where('token',$token)->first()){
        return response([
            'message'=>'Invalid token'
        ],400);
    }
    if(! $user = User::where('email',$passwordResets->email)->first()){
        return response([
            'message'=>'L\'utilisateur n'existe pas'
        ],404);
    }

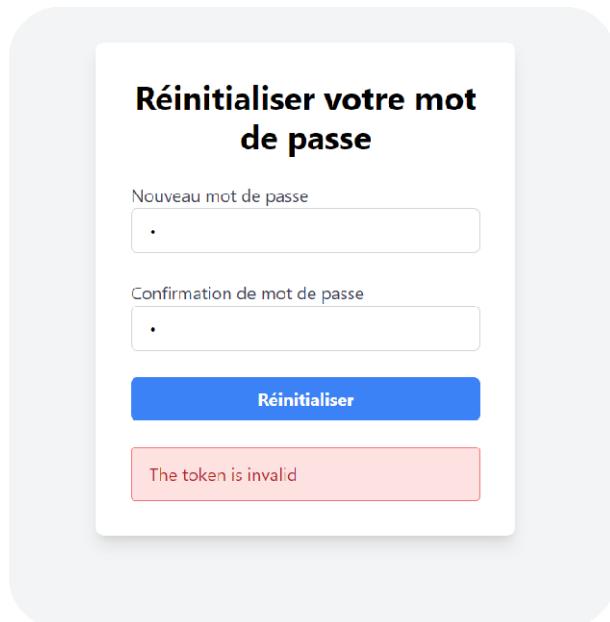
    $user->password = bcrypt($request->input('password'));
    $user->save();
    return response([
        'message'=>'success'
    ]);
}

```

Cette méthode permet de changer le mot de passe d'un utilisateur. La logique de cette méthode est qu'elle commence par la vérification du token récupéré de URL et le token qui existe dans la table

password_resets sont les même, si oui il cherche utilisateur concerné par le changement du mot de passe à l'aide d'email stocké déjà dans la cette table, puis elle effectue la modification du mot de passe après cryptage à l'aide de la méthode **bcrypt()**.

Si utilisateur essaie de jouer sur le Token :



2.5. Génération des PDFs :

UNIVERSITÉ ABDELMALEK ESSAADI

Fiche de Paiement
2023-05-25 09:18:43

À :
new name new prename

Status : Professeur Habilite
Établissement d'origine : Faculté de Medecine Tanger
Année universitaire : 2022/2023

ID	ETABLISSEMENT	VOLUME HORRAIRE	SALAIRE BRUTE	SALAIRE NET
1	Faculté de Medecine Tanger	50	MAD 20000	MAD 12400
2	Faculté des Sciences et Technique	50	MAD 20000	MAD 12400
Volume horraire			100 Heures	
Total			MAD 24800	

Detail des interventions

ID	INTERVENTION	ETABLISSEMENT	SEMESTRE	DÉBUT	FIN
3	Cours sur la littérature française	Faculté de Medecine Tanger	Semestre 2	2023-04-10	2023-05-20
7	fifa2028	Faculté des Sciences et Technique	semester2	1970-01-01	1970-01-01

Service XtraDH vous souhaite une bonne journee!
www.uae.ac.ma

Cahier de charge à respecter :

- ✓ La fiche de paie présente des informations liées à l'enseignant, son nom, son statuts, l'établissement d'origine dont il appartient, ainsi que les données relatives à son paiement.
- ✓ Vu que chaque paiement n'est pas lié directement à l'intervention correspondante, et que le cahier de charge oblige que l'enseignant reçoit son paiement associé à l'établissement dont il a effectué l'intervention, il est intéressant alors qu'on rassemble dans un tableau les informations suivantes :
 - L'établissement où il a enseigné.
 - Le volume horaire total dans la même établissement (Pour différentes interventions).
 - Son salaire brute (Sans aucune cotisation).
 - Son salaire net (Qu'il recevra dans son compte bancaire après les cotisations).
- ✓ Un volume horaire total (calculé à la base de toutes les interventions précédentes).
- ✓ Un total de paiement (calculé à la base de somme des salaires nets).

Il est aussi intéressant de savoir le détail pour chaque intervention, vu que le tableau précédent révèle des données attachées aux volumes horaires et à la monnaie en contrepartie, cela à cause de la liaison directe entre le salaire et l'établissement, et non pas le salaire et l'intervention.

- ✓ Chaque intervention est unique, à son propre id, l'établissement dont elle est effectuée, le semestre, ainsi qu'une date de début et une date de fin

Le code source :

La mise en page est purement travaillée avec HTML et CSS sans aucun Framework intégré, placée dans resources > views > PDF > pdf.blade.php.

Plusieurs directives sont utilisés pour faire le traitement nécessaire, à savoir @if, @foreach, Ainsi que {{ // data à afficher }} issue des modèles et les relations définis auparavant.

La route utilisée :

```
Route::get('/generate-pdf/{prof}', [PaiementController::class, 'generatePDFprof']);
```

UNIVERSITÉ

Fiche de Paiement

2023-05-25 09:18:43

```
<div class="date">    <?php
    $currentDateTime = date('Y-m-d H:i:s');
    echo $currentDateTime;
?>
```

La première information est celle de la date et l'heure, il s'agit de la date exacte du téléchargement de la fiche de paie suivant le modèle Y-m-d H :m :s en utilisant un Tag en PHP via la méthode date()

À :

new name new prename

Status : Professeur Habilite

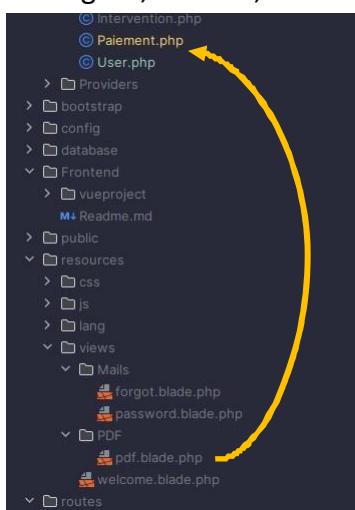
Établissement d'origine : Faculté de Medecine Tanger

Année universitaire : 2022/2023

```
<div class="title">{{$paiement->enseignant->Nom}} {{$paiement->enseignant->prenom}}</div>

<i>Status : </i>{{$grade->designation}}<br>
<i>Établissement d'origine : </i>{{$paiement->etablissement->Nom}}<br>
<i>Année universitaire : </i> {{$paiement->Annee_univ}}
```

Cela est possible grâce au jointure entre les modèles ainsi les relations déjà tissus via belongsTo,hasOne, HasMany..



La page fait appel aux modèle et ses relations

PROJET XTRADH

ID	ETABLISSEMENT	VOLUME HORRAIRE	SALAIRE BRUTE	SALAIRE NET
1	Faculté de Medecine Tanger	50	MAD 20000	MAD 12400
2	Faculté des Sciences et Technique	50	MAD 20000	MAD 12400

The diagram shows arrows pointing from the table rows to specific parts of a code snippet below.
 - An arrow points from the first row's 'ETABLISSEMENT' column to the variable `{{$item->établissement->Nom}}`.
 - Another arrow points from the first row's 'VOLUME HORRAIRE' column to the variable `{{$paiement->VH}}`.
 - A third arrow points from the first row's 'SALAIRE NET' column to the variable `{{$item->établissement->id}}`.

```

{{{$item->établissement->id}}}
{{{$item->établissement->Nom}}}
{{{$paiement->VH}}}
    
```

NB : \$item est une seule intervention (@foreach (\$intervent on \$item))

Formule de calcul des salaires selon son grade :

Brute : $MAD \{ (\$paiement->Taux_H * \$paiement->VH) \}$

Net : $MAD \{ (\$paiement->Taux_H * \$paiement->VH)*(1-\$paiement->IR) \}$

ID	ETABLISSEMENT	VOLUME HORRAIRE	SALAIRE BRUTE	SALAIRE NET
1	Faculté de Medecine Tanger	50	MAD 20000	MAD 12400
2	Faculté des Sciences et Technique	50	MAD 20000	MAD 12400
	Volume horaire			100 Heures
	Total			MAD 24800

The diagram shows annotations for the total volume and salary:
 - A yellow circle highlights the value '50' in the 'VOLUME HORRAIRE' column for both rows.
 - A green circle highlights the values 'MAD 12400' in the 'SALAIRE NET' column for both rows.
 - A green arrow points from the '100 Heures' label to the 'Total' row's 'SALAIRE NET' value 'MAD 24800'.
 - A green arrow points from the 'MAD 24800' value back up to the '100 Heures' label.

Le volume horaire et la somme ne sont que la somme des variables déjà présentes.

Pour ce faire, on a utilisé la directive @forach pour le calcul selon chaque intervention :

PROJET XTRADH

On a défini deux autres variables qui prennent des valeurs nulles au début, commençant de sommer tous les \$paiement->VH et \$net respectivement, pour toutes les interventions de l'enseignant en question.

```
<tfoot>
    <td colspan="3" class="name ">Volume horraire</td>
    <td colspan="3" class="number">@php echo $somme_heures; @endphp Heures</td>
    <td colspan="4" class="name "> Total </td>
    <td colspan="4" class="number"> MAD @php echo $somme_net; @endphp </td>
</tfoot>
```

Puis via un tag PHP, on affiche les valeurs finales.

Selon le cahier de charge. L'enseignant ne peut pas dépasser un paiement lié à un volume horaire total qui **dépasse les 200 heures**. C'est-à-dire il travaille tant qu'il veut, mais lors du paiement, le salaire maximum que l'université peut verser à son compte ne doit pas dépasser la limite de 200 heures correspondant son grade (le taux horaire).

C'est pour cela, un mécanisme de calcul est développé à la base de données pour arrêter tout versement illégal, ainsi que dans l'affichage.

```
@php
    if($somme_heures>200){
        $somme_net = 200 * $paiement->Taux_H *(1-$paiement->IR);
    }
@endphp
```

Ici, quel que soit la somme horaire calculé qui dépasse la limite. Le salaire net du professeur est limité aussi à 200 heures selon son grade assuré par la variable (\$paiement->Taux_H) toute en éliminant les cotisations de 38% de IR.

Un message de dépassement est aussi affiché pour informer l'enseignant.

ID	ETABLISSEMENT	VOLUME HORRAIRE	SALAIRE BRUTE	SALAIRE NET
1	Faculté de Medecine Tanger	300	MAD 90000	MAD 55800
	Volume horraire			300 Heures
	Plafond des heures			200 h
	Total			MAD 37200

Une fois le dépassement est détecté, un message est affiché grâce au :



```

@php
    if($somme_heures>200){
        @endphp
            <tr class="total" style="font-size:14px;color: #d31811;">
                <td colspan="4" class="name ">Plafond des heures</td>
                <td colspan="4" class="number">200 h</td>
            </tr>
        @endphp
    }
    @endphp

```

Le détail de chaque intervention présente les informations suivantes :

Detail des interventions

ID	INTERVENTION	ETABLISSEMENT	SEMESTRE	DÉBUT	FIN
3	Cours sur la littérature française	Faculté de Medecine Tanger	Semestre 2	2023-04-10	2023-05-20
7	fifa2028	Faculté des Sciences et Technique	semester2	1970-01-01	1970-01-01

Service XtraDH vous souhaite une bonne journee!
www.uae.ac.ma

Ils sont générés pour chaque intervention indépendante à partir de la base de données comme suit :



```

@foreach ($intervention as $item)
    <tr style="font-size: 11px; color: #101010" class="total">
        <td>{{$item->id_intervention}}</td>
        <td>{{$item->Intitule_Intervention}}</td>
        <td>{{$item->etablissement->Nom}}</td>
        <td>{{$item->Semestre}}</td>
        <td>{{$item->Date_debut}}</td>
        <td>{{$item->Date_fin}}</td>
    </tr>
@endforeach

```

Fonction generatePDF – PaiementController

```

public function generatePDFprof($id_paiement){

    $paiement = Paiement::where('id',$id_paiement)
        ->with(['enseignant:id,Nom,prenom,id_Grade'])
        ->with(['etablissement:id,Nom'])
        ->first();

    $grade = Grade::where('id_Grade',$paiement->enseignant->id_Grade)->first();
    $intervention = $paiement->enseignant->id ;

    $intervention = intervention::where('id_Intervenant',$intervention)
        ->where('visa_uea',1)
        ->where('visa_etb',1)
        ->with(['etablissement:id,Nom'])
        ->get()
        ;

    $data = [
        'paiement'=>$paiement,
        'intervention'=>$intervention,
        'grade'=>$grade
    ];
    $pdf = PDF::loadView( 'PDF.pdf', $data);
    return $pdf->download('itsolutionstuff.pdf');

}

```

La fonction generatePDFprof prend en paramètre l'ID d'un paiement (\$id_paiement).

\$paiement :

On récupère les informations du paiement en utilisant le modèle Paiement et la méthode where pour filtrer sur l'ID spécifié. La méthode with est utilisée pour charger les relations (enseignant et établissement) du paiement, puis ->first(), pour récupérer qu'un seul (la première trouvé)

\$grade :

On récupère les informations du grade de l'enseignant lié au paiement. Cela est fait en utilisant le modèle Grade et la méthode where pour trouver l'ID spécifique du grade.

\$intervention :

La variable \$in

Intervention est initialisée avec la clé primaire de l'enseignant du paiement.

On filtre le résultat qu'on souhaite obtenir sur le modèle Intervention par la méthode where() : Ces conditions doivent apporter sur :

- Le même ID de l'intervenant
- Ses interventions sont validés par l'établissement (`visa_etb = 1`) ainsi que par l'université (`visa_uea = 1`)

On s'intéresse aussi à avoir des infos de l'établissement où l'intervention est effectuée.

Les résultats sont récupérés avec la méthode `get()` et assignés à la variable `$intervention`.

\$data : (array)

Les données nécessaires pour générer le PDF sont rassemblées dans un tableau associatif `$data`, contenant les variables `$paientement`, `$intervention` et `$grade`.

\$pdf :

En utilisant la bibliothèque PDF, la fonction `loadView` est appelée pour charger la vue `pdf.blade.php` dans le dossier PDF avec les données `$data` et générer le contenu du PDF.

```

views
  ▶ Mails
  ▶ PDF
    ↳ pdf.blade.php ← Yellow arrow here
    ↳ welcome.blade.php
  ▶ routes
153   'grade' =>$grade
154   ];
155   $pdf = PDF::loadView('PDF.pdf', $data);
156   return $pdf->download('itsolutionstuff.pd
157   }
158
159 public function paiementprof($id_prof)

```

Celui-ci est téléchargeable en utilisant la méthode `download('nom_du_fichier')`.

2.6. Routes

Les routes web sont utilisées pour définir les points d'entrée de votre application accessible via un navigateur web. Elles gèrent les requêtes *HTTP GET, POST, PUT, DELETE*, etc... Les routes web sont généralement définies dans le fichier `routes/web.php`.

Les routes API sont utilisées pour créer des interfaces de programmation pour permettre à d'autres applications d'interagir avec votre application. Elles sont généralement définies dans le fichier `routes/api.php`.

PROJET XTRADH

```
/* Etablissement Routes */
Route::apiResource('Etablissement', EtablissementController::class);

/* Grade Routes */
Route::apiResource('Grade', GradeController::class);

/* Enseignant routes */
Route::apiResource('Enseignant', EnseignantController::class);

/* Administrateur routes */
Route::apiResource('Administrateur', AdministrateurController::class);

/* Intervention Routes */
Route::apiResource('Intervention', InterventionController::class);
Route::get('/valideruae',[InterventionController::class,'valideruae']);
Route::get('/invalideruae',[InterventionController::class,'invalideruae']);
Route::get('/valideretb',[InterventionController::class,'valideretb']);
Route::get('/invalideretb',[InterventionController::class,'invalideretb']);

/* Paiement Routes */
Route::apiResource('Paiement', PaiementController::class)
```

apiResource est une classe qui facilite la création d'une API RESTful permettant de définir rapidement les opérations de CRUD (Create, Read, Update, Delete) pour un modèle spécifique.

get est pour associer le déclanchement d'une méthode dans un contrôleur spécifique pour la route mentionnée

Voici la liste des routes de notre projet XtraDH

```
/* Paiement Routes */
Route::apiResource('paiement',PaiementController::class);
Route::get('/paiementprof',[PaiementController::class,'paiementprof'])->middleware("auth:sanctum");
Route::get('/historiquepdfpale',[PaiementController::class,'historiquepdfpale'])->middleware("auth:sanctum");
Route::get('/consultpaiementetabdirecteur',[PaiementController::class,'consultpaiementetabdirecteur'])->middleware("auth:sanctum");

/** AUTH ROUTE */
Route::post('/login',[AuthController::class,'login']);
Route::post('/forgot',[ForgotController::class,'forgot']);
Route::post('/reset',[ForgotController::class,'reset']);

/** GENERATE PDF */
Route::get('postfix', [PaiementController::class, 'postfix']);
Route::get('generate-pdf/{prof}', [PaiementController::class, 'generatePDFprof']);

/** Users Routes */
Route::apiResource('user',UserController::class);
Route::post('/storeprofetb',[UserController::class,'storeProfEtB'])->middleware("auth:sanctum");
Route::post('/storeadminetb',[UserController::class,'storeAdmEtB'])->middleware("auth:sanctum");
Route::patch('/updateprof',[UserController::class,'updateprof'])->middleware("auth:sanctum");
Route::patch('/updateadm/{idAdm}',[UserController::class,'updateAdm'])->middleware("auth:sanctum");
Route::post('/ajoutinterventiontab',[UserController::class,'ajoutInterventiontab'])->middleware("auth:sanctum");
Route::get('/adminprofile',[UserController::class,'adminProfile'])->middleware("auth:sanctum");
Route::get('/profprofile',[UserController::class,'profProfile'])->middleware("auth:sanctum");
Route::delete('/deleteadm/{id_user}',[UserController::class,'destroyAdmin']);

//protected routes
Route::group(['middleware'=>['auth:sanctum']], function () {
    Route::post('/logout',[AuthController::class,'logout']);
    Route::get('/user-profile',[AuthController::class,'userProfile']);
});

}); //protected for admin univ
Route::apiResource('établissement',EtablissementController::class);
/* Grade Routes */
Route::apiResource('grade', GradeController::class);

Route::apiResource('enseignant',EnseignantController::class);
/* Administrateur routes */

Route::apiResource('administrateur',AdministrateurController::class);
Route::get('/directeurtab',[AdministrateurController::class,'directeurETB'])->middleware('auth:sanctum');

/* Intervention Routes */
Route::apiResource('Intervention', InterventionController::class);
Route::get('/getintervention',[InterventionController::class,'getProfIntervention'])->middleware('auth:sanctum');
Route::get('/directeurtabinterval',[InterventionController::class,'directeurTabInterval'])->middleware('auth:sanctum');
Route::get('/directeurtabintervalvalid',[InterventionController::class,'directeurTabIntervalValid'])->middleware('auth:sanctum');
/);

/* Enseignant routes */
Route::apiResource('enseignant',EnseignantController::class);

/* Intervention Routes */
Route::apiResource('intervention', InterventionController::class);
Route::get('/interventionuaevalid',[InterventionController::class,'interventionuaevalid']);

//protected validation
//protected for directeur
Route::get('/valideretb/{id}',[InterventionController::class,'valideretb']);
Route::get('/invalideretb/{id}',[InterventionController::class,'invalideretb']);
/);

//protected for president
Route::get('/valideruae/{id}',[InterventionController::class,'valideruae']);
Route::get('/invalideruae/{id}',[InterventionController::class,'invalideruae']);
/);

//protected for prof
Route::apiResource('enseignant',EnseignantController::class);
Route::get('/profetab',[EnseignantController::class,'indexProf'])->middleware('auth:sanctum');
Route::delete('/deleteprof/{id_user}',[UserController::class,'destroyProf']);
/);
```

2.7. Liaison avec Front-End

La connexion du FRONT ET BACK : utilisation d'API :

```

actions: {
    async submit({dispatch},credentials){
        try {
            const response = await axios.post('/login',credentials)
            return dispatch('attempt',response.data.token)
        } catch (error) {
            console.log('failed')
        }
    },
    async attempt({commit,state}, token) {
        try {
            if(token){
                commit('setToken',token)
            }
            if(!state.token){
                return;
            }
            const response = await axios.get('/user-profile')
            commit('setUser', response.data)
        } catch (error) {
            commit('setUser',null)
            commit('setToken',null)
        }
    },
    logout({commit}){
        return axios.post('/logout').then(()=>{
            commit('setUser',null)
            commit('setToken',null)
        })
    }
}

```

Pour utiliser end-point “[/login](#)” il faut fournir avec la méthode post le login et le mot de passe, si authentification se fait avec succès utilisateur et un token serons retourner pour assurer la sécurité de notre API.

End point “[/user-profile](#)” permet de retourner utilisateur connecté à application pour vérifier Authentification à l'aide de VueX.

```

import store from '@/store'
import axios from 'axios';
store.subscribe((mutation) => {
    if(mutation.type === 'auth/setToken') {
        if(mutation.payload) {
            axios.defaults.headers.common['Authorization'] = `Bearer ${mutation.payload}`
            localStorage.setItem('token', mutation.payload);
        }
        else {
            axios.defaults.headers.common['Authorization'] = null
            localStorage.removeItem('token');
        }
    }
})

```

Pour utiliser tout end-point qui nécessite un middleware ”[auth : sanctum](#)” il est Obligatoire de fournir avec l'autorisation de type Bearer le token récupéré lors de l'authentification dans notre application pour des raison de sécurité.

- *Forgetpassword.vue*

```
async submitForm() {
  try {
    const response = await axios.post('/forgot',
    {
      email: this.email
    });
  }
}
```

Pour utiliser l'end-point "`/forgot`" il faut fournir avec une méthode de type post l'email d'utilisateur qui veux récupérer son mot de passe.

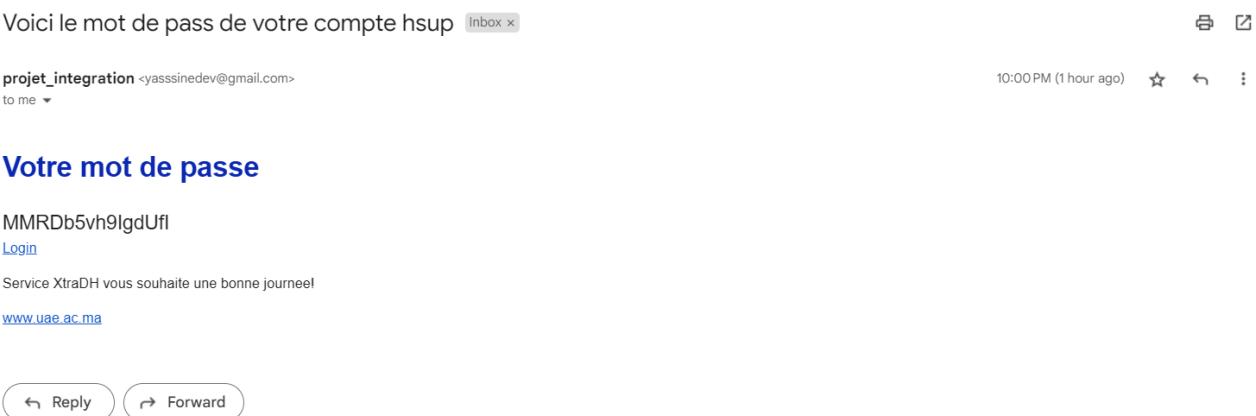
2.8. Les Notifications :

- *Envoyer le mot de passe par email aux utilisateurs :*

```
$fields['password']=Str::random(15);
$user = User::create([
  'type' =>$fields['type'],
  'email' => $fields['email'],
  'password'=>bcrypt($fields['password'])
]);

$email = $fields['email'];
Mail::send('Mails.password',[ 'password'=>$fields['password']],function(Message $message)use($email)
{
  $message->to($email);
  $message->subject('Voici le mot de passe de votre compte hsup');
});
```

Après la création des comptes des utilisateurs, ces derniers vont recevoir leur mot de passe par leur email

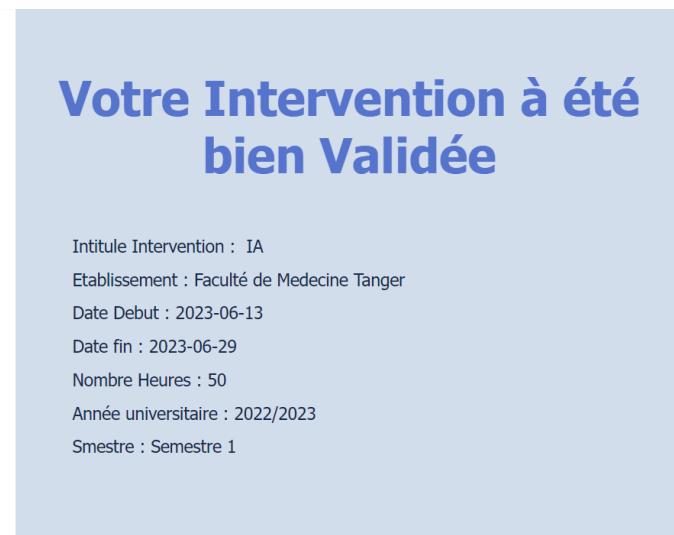


- ***Notification de validation d'intervention des enseignant***

```
$intervention->visa_uee = 1 ;
/intervention->update();
/intervention = DB::table('interventions')
    ->join('enseignants','id_Intervenant','=','enseignants.id')
    ->join('etablissements','interventions.id_Etab','=','etablissements.id')
    ->join('users','enseignants.id_user','=','users.id_user')
    ->where('id_intervention',$id)->first();

$email = $intervention->email;
Mail::send('Mails.valide',['intervention'=>$intervention],function(Message $message)use($email){
    $message->to($email);
    $message->subject('Validation de votre intervention');
});
```

- **Après La validation des interventions des enseignants par le président d'université, les enseignants vont recevoir un message pour informer que leur intervention est maintenant validée**



2.9. Les Tests de fonction :

Pour tester le fonctionnement de l'API, on a utilisé l'outil POSTMAN :



Postman est un outil populaire et puissant pour tester et développer des API. Il offre une interface conviviale qui permet aux développeurs de facilement envoyer des requêtes HTTP, de tester les réponses et d'explorer les fonctionnalités des API. Avec Postman, vous pouvez créer des collections de requêtes, organiser vos tests, automatiser des flux de travail et collaborer avec d'autres membres de l'équipe.

Méthode

END-POINT

Token

Format de Résultat

Status Code

Résultat

```

1
2   {
3     "id": 21,
4     "PPR": "65186",
5     "Nom": "SAIDI",
6     "prenom": "KASIM",
7     "Date_Naissance": "1994-05-03",
8     "Etablissement": 9,
9     "id_Grade": 9,
10    "id_user": 65,
11    "created_at": "2023-06-12T15:21:22.000000Z",
12    "updated_at": "2023-06-12T22:36:44.000000Z",
13    "etab_permanant": {
14      "id": 9,
15      "Nom": "Ecole national des science appliqués de Tanger"
16    }
}

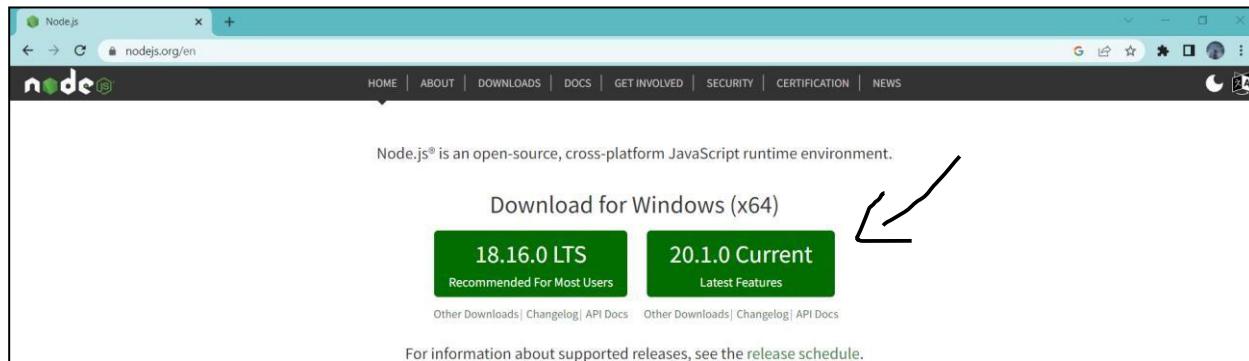
```

3. CHAPITRE 3 :

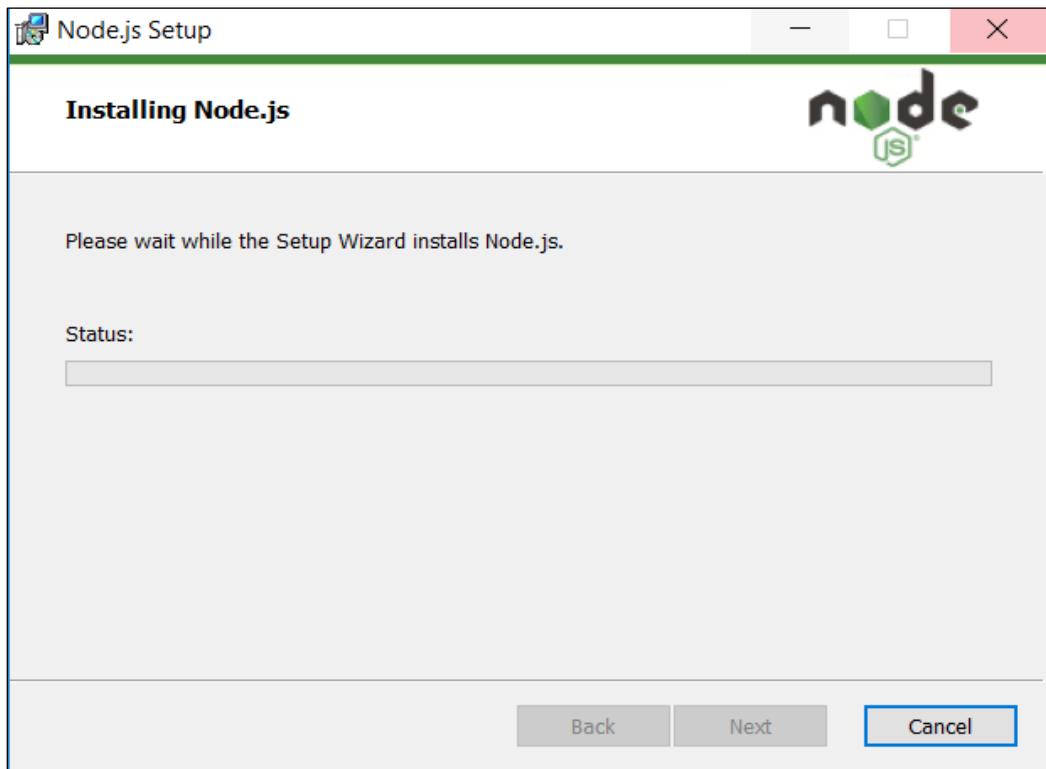
Développement Front-end avec Vue.js

Ce chapitre se concentre sur l'utilisation de Vue.js, un framework JavaScript progressif, pour le développement de la partie front-end de notre application web. Le Responsable Front-end nous fait explorer comment Vue.js facilite la création d'interfaces utilisateur réactives et interactives, en utilisant des composants modulaires et une approche basée sur les données.

3.1. Installation de vuejs :

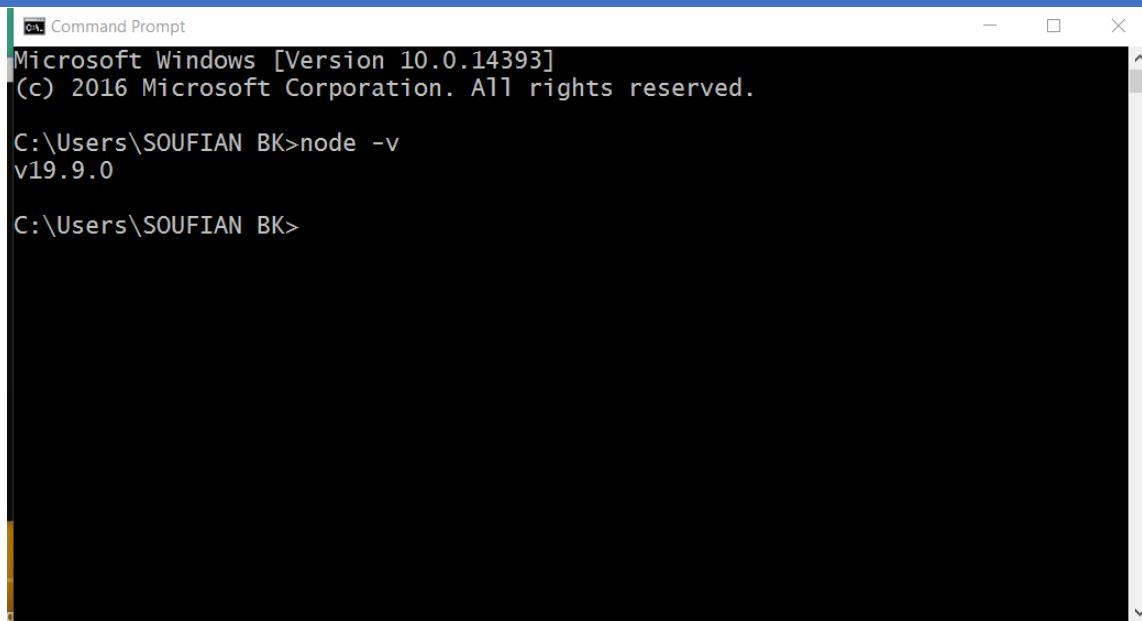


On installe la version signalée dans l'image.



Après l'installation, dans n'importe quel terminal (cmd, ou cmd du nodejs ou terminal de Vscode) On test si node est bien installé par la Commande **node -v**

PROJET XTRADH



```
cmd. Command Prompt
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\SOUFIAN BK>node -v
v19.9.0

C:\Users\SOUFIAN BK>
```

C'est bien installé.

Avant la création du projet vuejs en utilisant cmd :



```
cmd. Command Prompt
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\SOUFIAN BK>node -v
v19.9.0

C:\Users\SOUFIAN BK>npm install -g @vue/cli
[redacted] \ idealTree:npm: sill idealTree buildDeps
```

On utilise la commande `npm install -g @vue/cli`, pour installer vue CLI, qui permet de générer rapidement un nouveau projet.

Exécuter la commande `vue --version`, pour s'assurer que vue cli est installé.

Puis , on crée un dossier ou on va créer notre projet vuejs :

Le dossier que j'ai créé est nommé Tuts.

```
C:\Users\SOUFIAN BK>cd
C:\Users\SOUFIAN BK

C:\Users\SOUFIAN BK>cd SOUFIAN
The system cannot find the path specified.

C:\Users\SOUFIAN BK>cd Documents
C:\Users\SOUFIAN BK\Documents>cd ..
C:\Users\SOUFIAN BK>cd Documents
C:\Users\SOUFIAN BK\Documents>cd Tuts
The system cannot find the path specified.

C:\Users\SOUFIAN BK\Documents>cd Tuts
C:\Users\SOUFIAN BK\Documents\Tuts> vue create modal-project

Vue CLI v5.0.8
? Please pick a preset: (Use arrow keys)
> Default ([Vue 3] babel, eslint)
  Default ([Vue 2] babel, eslint)
  Manually select features
```

On peut choisir n'importe quelle configuration qu'on veut, cela dépend de notre choix, et de ce qu'on désire réaliser.

```
C:\Users\SOUFIAN BK\Documents>cd Tuts
C:\Users\SOUFIAN BK\Documents\Tuts> vue create modal-project

Vue CLI v5.0.8
? Please pick a preset: Manually select features
? Check the features needed for your project: (Press <space> to select,
<a> to toggle all, <i> to invert selection, and <enter> to proceed)
  (*) Babel
> ( ) TypeScript
  ( ) Progressive Web App (PWA) Support
  ( ) Router
  ( ) Vuex
  ( ) CSS Pre-processors
  (*) Linter / Formatter
  ( ) Unit Testing
  ( ) E2E Testing
```

Si on sélectionne **Manually select features**, on trouve cette liste où on va choisir les critères qu'on veut.

```
Vue CLI v5.0.8
? Please pick a preset: Manually select features
? Check the features needed for your project: Babel
? Choose a version of Vue.js that you want to start the project with
(Use arrow keys)
> 3.x
  2.x
```

```
c:\Users\SOUFIAN BK\Documents\Tuts> vue create modal-project

Vue CLI v5.0.8
? Please pick a preset: Manually select features
? Check the features needed for your project: Babel
? Choose a version of Vue.js that you want to start the project with 3.x

? Where do you prefer placing config for Babel, ESLint, etc.? In
dedicated config files
? Save this as a preset for future projects? No

Vue CLI v5.0.8

  □ Creating project in C:\Users\SOUFIAN BK\Documents\Tuts\modal-project.
  □□ Installing CLI plugins. This might take a while...
[██████████] - idealTree:modal-project: sill idealTree buildDep
```

L'installation est générée maintenant.

```
added 15 packages, and audited 852 packages in 15s

91 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
  □□ Invoking generators..
  □□ Installing additional dependencies..

added 15 packages, and audited 852 packages in 15s

91 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
  □ Running completion hooks...

  □□ Generating README.md...

  □□ Successfully created project modal-project.
  □□ Get started with the following commands:

$ cd modal-project
$ npm run serve

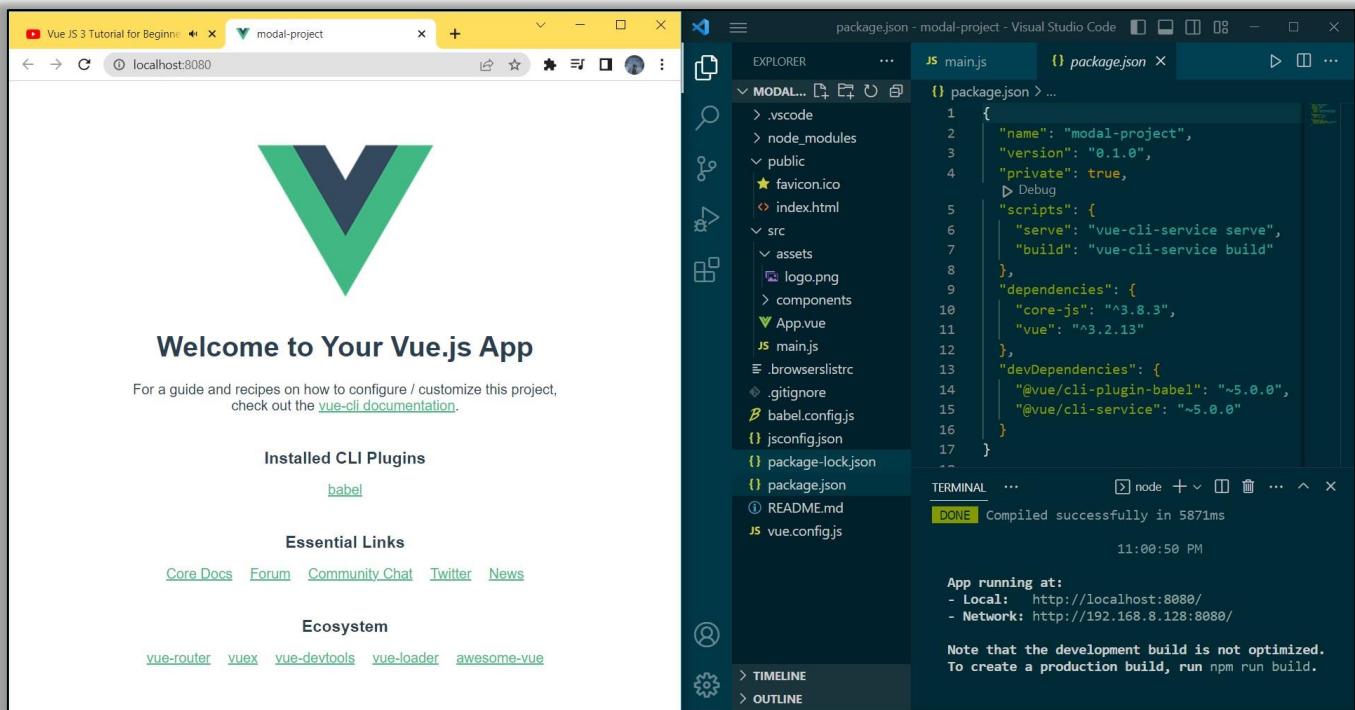
C:\Users\SOUFIAN BK\Documents\Tuts> █
```

Et donc le terminal donne les commandes à exécuter pour lancer le projet.(Utiliser **code .** pour lancer le projet dans vuejs)

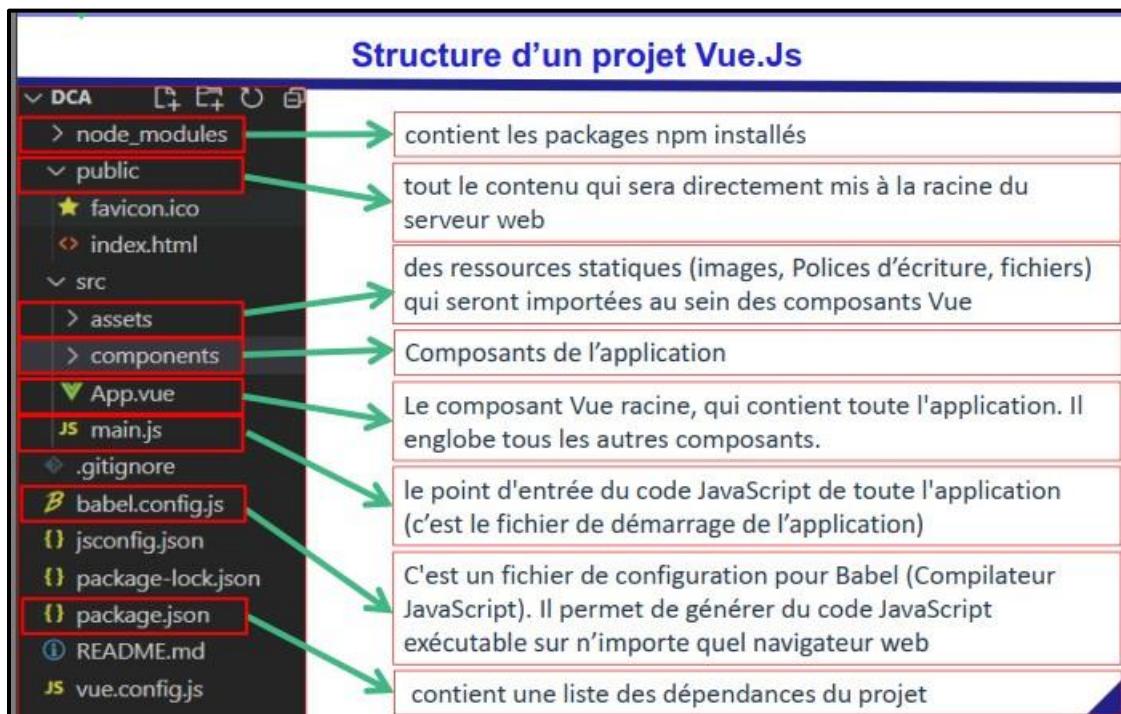
```
C:\Users\SOUFIAN BK\Documents\Tuts>cd modal-project
C:\Users\SOUFIAN BK\Documents\Tuts\modal-project>code .
C:\Users\SOUFIAN BK\Documents\Tuts\modal-project>
```

Et voici le résultat :

PROJET XTRADH



Voici la structure du projet par défaut :



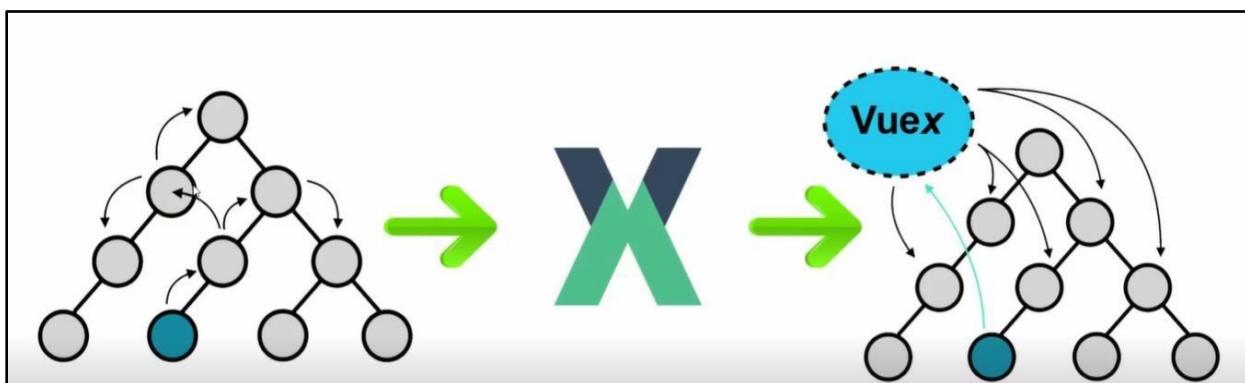
3.2. Nécessité de l'outil vuex dans le projet :

3.2.1. C'est quoi Vuex ?

Vuex est une bibliothèque d'état de gestion pour les applications Vue.js. Il s'agit d'un modèle d'architecture de gestion d'état inspiré du pattern Flux, largement utilisé dans le développement d'applications front-end. Vuex est spécialement conçue pour être utilisé avec Vue.js, bien qu'il puisse également être utilisé avec d'autres frameworks JavaScript.

3.2.2. But de vuex ?

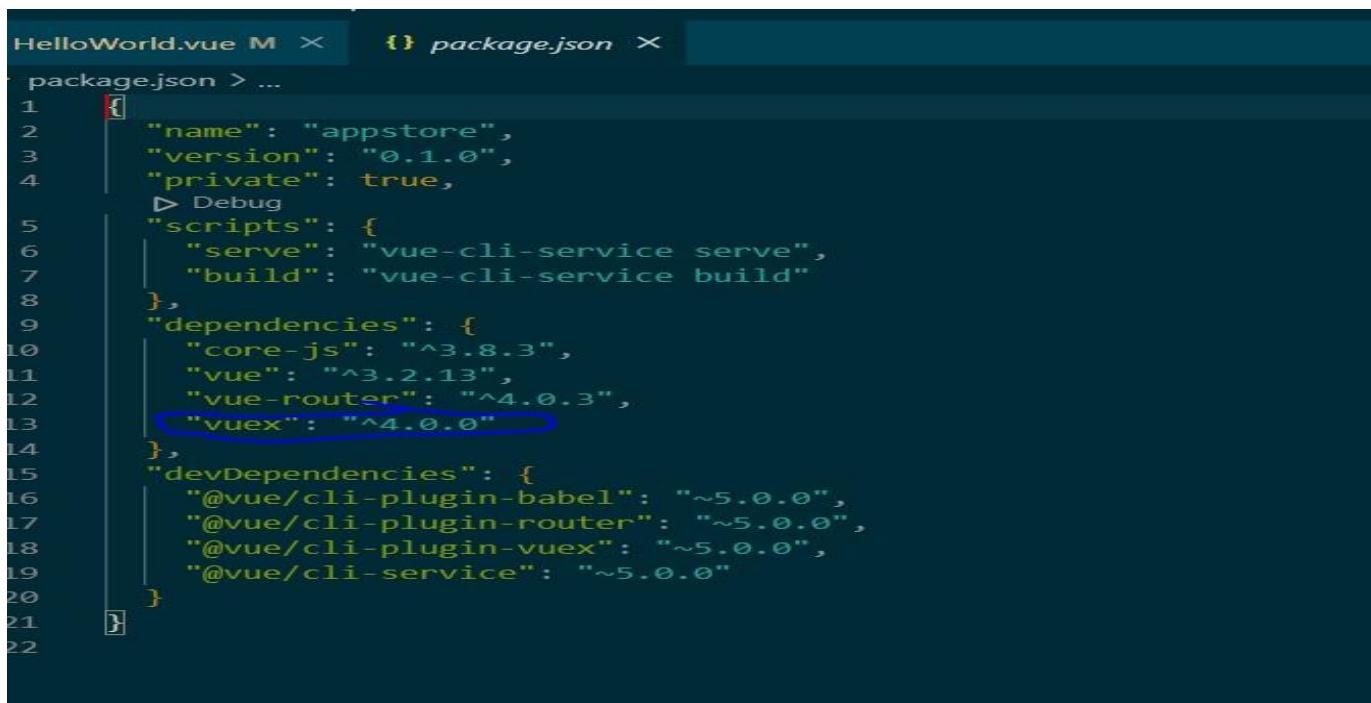
Le but principal de Vuex est de faciliter la gestion de l'état global de l'application, c'est-à-dire les données qui doivent être partagées entre différents composants de l'application. Lorsqu'une application devient complexe avec de nombreux composants interagissant les uns avec les autres, il peut devenir difficile de gérer et de synchroniser l'état à travers l'application.



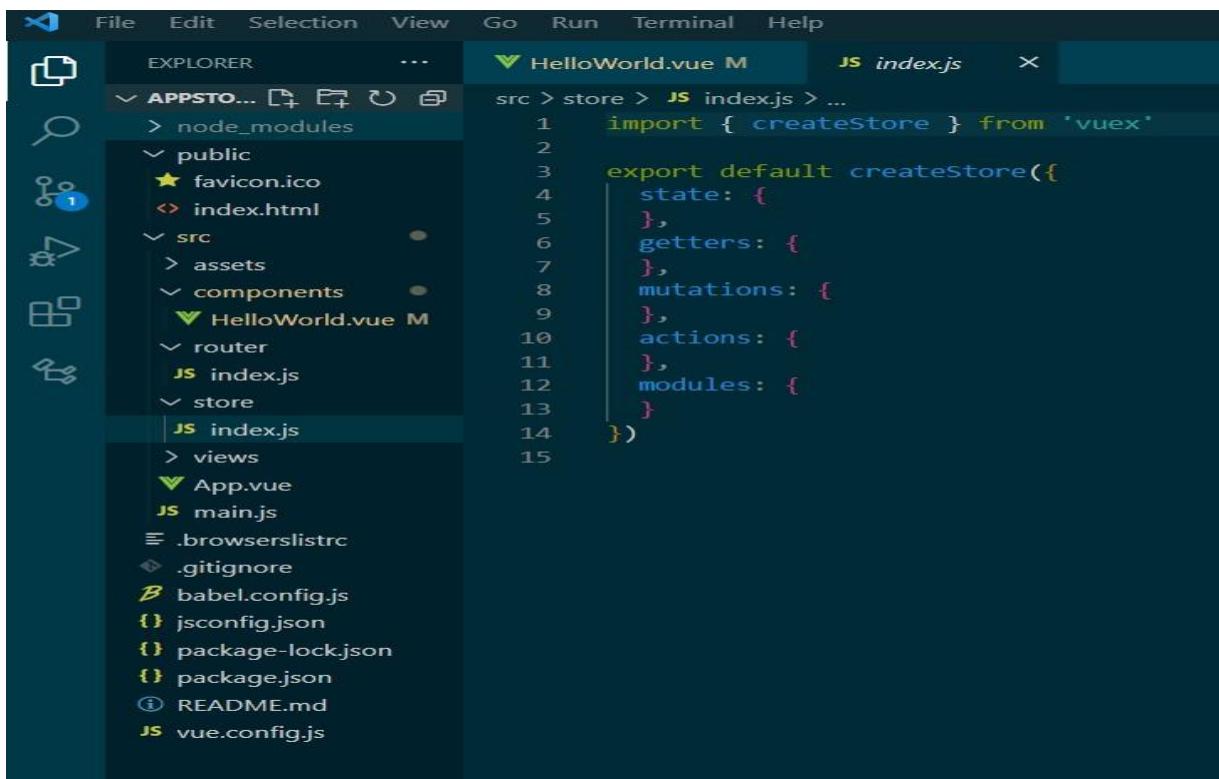
3.2.3. Comment installer Vuex dans notre projet ?

```
found 0 vulnerabilities
PS C:\Users\SOUFIAN BK\Documents\All of vue\prj-ofc> vue add vuex
>>
WARN | There are uncommitted changes in the current repository, it's recommended to commit or stash them first.
? Still proceed? Yes
📦 Installing @vue/cli-plugin-vuex...
```

PROJET XTRADH



```
package.json > ...
1  {
2    "name": "appstore",
3    "version": "0.1.0",
4    "private": true,
5    ▷ Debug
6    "scripts": {
7      "serve": "vue-cli-service serve",
8      "build": "vue-cli-service build"
9    },
10   "dependencies": {
11     "core-js": "^3.8.3",
12     "vue": "^3.2.13",
13     "vue-router": "^4.0.3",
14     "vuex": "4.0.0" // This line is highlighted with a blue box
15   },
16   "devDependencies": {
17     "@vue/cli-plugin-babel": "~5.0.0",
18     "@vue/cli-plugin-router": "~5.0.0",
19     "@vue/cli-plugin-vuex": "~5.0.0",
20     "@vue/cli-service": "~5.0.0"
21   }
22 }
```



The Explorer panel shows the project structure:

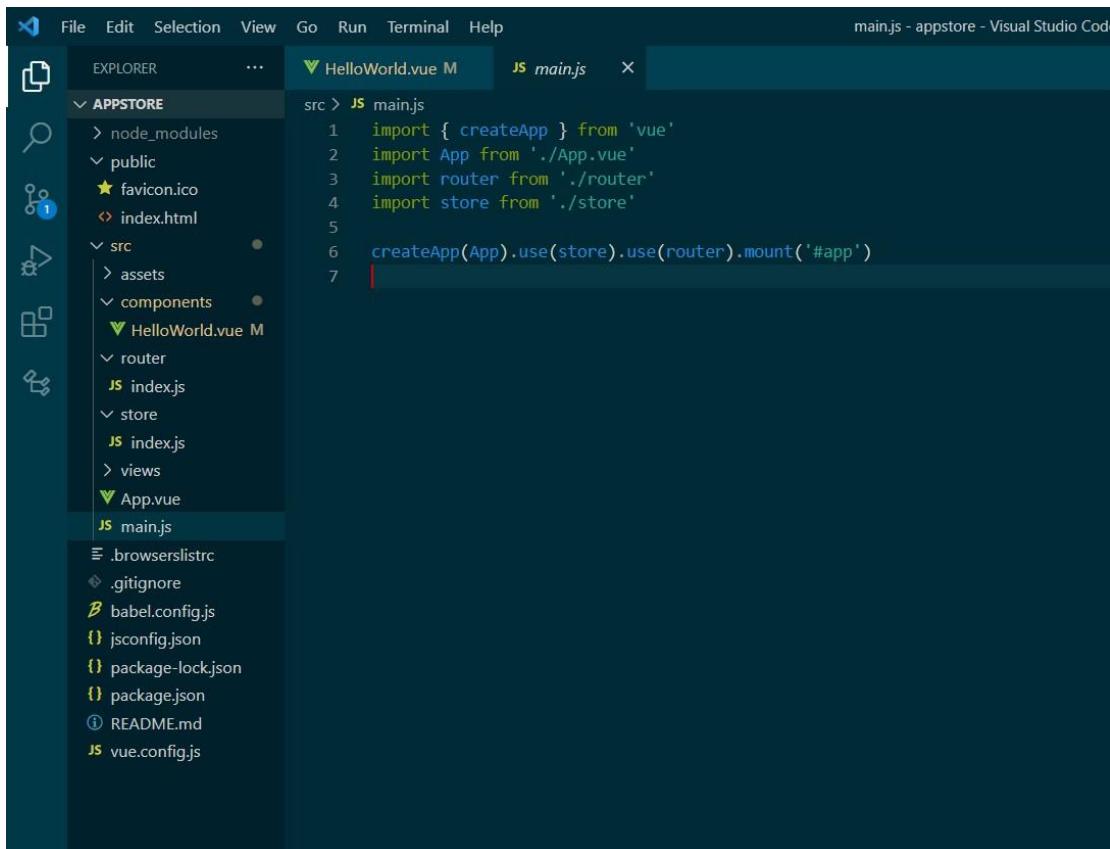
- APPSTO... (root)
- node_modules
- public
 - favicon.ico
 - index.html
- src
 - assets
 - components
 - HelloWorld.vue
 - router
 - index.js
 - store
 - index.js
 - views
- App.vue
- main.js
- .browserslistrc
- .gitignore
- babel.config.js
- jsconfig.json
- package-lock.json
- package.json
- README.md
- vue.config.js

The main editor panel shows the content of the index.js file in the store directory:

```
src > store > index.js > ...
1 import { createStore } from 'vuex'
2
3 export default createStore({
4   state: {
5     },
6     getters: {
7     },
8     mutations: {
9     },
10    actions: {
11    },
12    modules: {
13    }
14 })
```

PROJET XTRADH

Vuex est importé automatiquement dans **main.js**

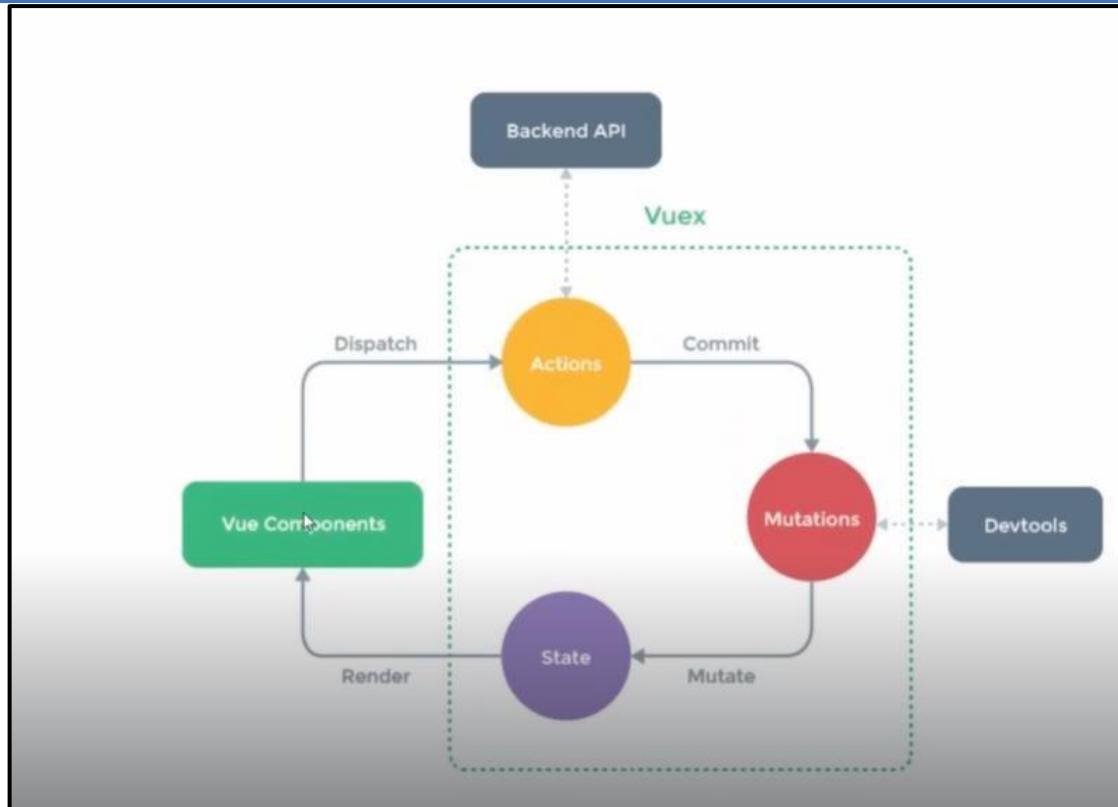


```
File Edit Selection View Go Run Terminal Help
EXPLORER APPSTORE ...
HelloWorld.vue M JS main.js X
src > JS main.js
1 import { createApp } from 'vue'
2 import App from './App.vue'
3 import router from './router'
4 import store from './store'
5
6 createApp(App).use(store).use(router).mount('#app')
7
```

The screenshot shows the Visual Studio Code interface with the 'main.js' file open in the center editor pane. The code imports 'createApp' from 'vue', 'App' from './App.vue', 'router' from './router', and 'store' from './store'. It then creates an app using these components and mounts it to the '#app' element. The left sidebar shows a tree view of the project structure under the 'APPSTORE' folder, including 'node_modules', 'public' (with 'favicon.ico' and 'index.html'), 'src' (with 'assets', 'components' containing 'HelloWorld.vue', 'router' with 'index.js', 'store' with 'index.js', 'views', 'App.vue', and 'main.js'), and various configuration files like '.browserslistrc', '.gitignore', 'babel.config.js', 'jsconfig.json', 'package-lock.json', 'package.json', 'README.md', and 'vue.config.js'. The status bar at the bottom right indicates 'main.js - appstore - Visual Studio Code'.

Dans le fichier store, on trouve le fichier index.js, qui contient 5 choses :

- State : qui sont les variables ou les données qu'on veut qu'elles soient partagées entre les composants.
- Mutations : sont des fonctions synchrones qui utilisées pour modifier l'état(state) dans le store
- Actions : Les actions sont similaires aux mutations, mais elles sont asynchrones. Elles sont responsables de l'exécution d'opérations asynchrones, telles que des requêtes API, puis de la validation des mutations pour modifier l'état.
- Getters : qui nous permettent de récupérer les states depuis le dossier store.
- Modules : où on signale les modules créés, la notion du modèle c'est lorsqu'on développe , un dossier dans store par exemple auth, contenant un fichier index.js lui-même contient la même structure state / mutations....



Dans notre projet on a utilisé vuex afin de pouvoir récupérer l'e-mail d'utilisateur depuis la page de connexion vers le dashboard qui conforme avec son type. Et pour centraliser les méthodes nécessaires pour la récupération des données d'utilisateur connecté.

3.3. Structure de notre projet :

Dans notre projet j'ai installé :

```
Vue CLI v3.0.8
? Please pick a preset: Manually select features
? Check the features needed for your project: (Press <space> to select,
<a> to toggle all, <i> to invert selection, and <enter> to proceed)
(*) Babel
>( ) TypeScript
  ( ) Progressive Web App (PWA) Support
  ( ) Router
  ( ) Vuex
  ( ) CSS Pre-processors
  (✓) Linter / Formatter
  ( ) Unit Testing
  ( ) E2E Testing
```

- *Router et Vuex en plus de babel. Pour la structure de projet adopté :*

📁 node_modules	05/23/23 1:57 PM	File folder	
📁 public	05/25/23 11:35 PM	File folder	
📁 src	05/23/23 1:57 PM	File folder	
📄 .browserslistrc	05/11/23 4:18 PM	BROWSERSLISTRC...	1 KB
⚙️ .gitignore	05/11/23 4:18 PM	Git Ignore Source ...	1 KB
JS babel.config	05/11/23 4:18 PM	JavaScript Source ...	1 KB
⌚ finish	05/24/23 3:42 PM	Shell Script	1 KB
📄 jsconfig	05/11/23 4:18 PM	JSON Source File	1 KB
📄 package	05/24/23 3:42 PM	JSON Source File	1 KB
📄 package-lock	05/24/23 3:42 PM	JSON Source File	409 KB
JS postcss.config	05/11/23 4:18 PM	JavaScript Source ...	1 KB
⬇️ README	05/11/23 4:18 PM	Markdown Source ...	1 KB
⌚ start	05/24/23 3:42 PM	Shell Script	1 KB
JS tailwind.config	05/11/23 4:18 PM	JavaScript Source ...	1 KB
JS vue.config	05/16/23 4:24 PM	JavaScript Source ...	1 KB

Je souligne qu'on a utilisé **TAILWIND** comme framework de css.

PROJET XTRADH

Dans le fichier src on a :

📁 assets	05/23/23 1:57 PM	File folder
📁 components	05/23/23 1:57 PM	File folder
📁 Interfaces	05/23/23 1:57 PM	File folder
📁 Pages	05/23/23 1:57 PM	File folder
📁 router	05/24/23 4:11 PM	File folder
📁 store	05/23/23 1:57 PM	File folder
📁 views	05/24/23 4:11 PM	File folder
📄 App	06/09/23 7:24 PM	VUE Source File 1 KB
📄 main	05/23/23 9:18 AM	JavaScript Source ... 1 KB

Où j'ai mis toutes les vues utilisateur dans Interfaces et Pages (page deconnexion, mot de passe oublié, récupération de mot de passe)

nents > XtraDh > Frontend > vueproject > src > assets			
Name	Date modified	Type	Size
📁 css	05/23/23 1:57 PM	File folder	
📁 images	05/27/23 11:05 PM	File folder	
🖼 logo	05/11/23 4:18 PM	PNG File	7 KB

- **Components :**

Name	Date modified	Type
📁 Dashboard	05/24/23 4:11 PM	File folder

Dashboard :

📁 etab	05/23/23 1:57 PM	File folder
📁 Icons	05/24/23 4:11 PM	File folder
📁 Intervention	05/30/23 11:23 PM	File folder
📁 Président	05/25/23 11:35 PM	File folder
📁 Prof	05/25/23 11:42 PM	File folder
📁 Univ	05/24/23 4:11 PM	File folder

- **Contenu des Interfaces :**

Name	Date modified	Type
Admin_Etab	05/25/23 11:42 PM	File folder
Admin_Univ	05/25/23 11:42 PM	File folder
Directeur_Etab	05/25/23 11:42 PM	File folder
President_Univ	05/25/23 11:42 PM	File folder
Users	06/09/23 8:24 PM	File folder

Contenu des Pages :

Forgetpassword	05/26/23 12:55 PM	VUE Source File	3 KB
Login	05/27/23 11:06 PM	VUE Source File	6 KB
Resetpassword	05/26/23 12:55 PM	VUE Source File	4 KB
Wait	05/26/23 12:55 PM	VUE Source File	3 KB

Contenu de store :

auth	05/23/23 1:57 PM	File folder	
index	05/20/23 10:26 AM	JavaScript Source ...	1 KB
subscriber	05/26/23 6:45 PM	JavaScript Source ...	1 KB

On va détailler au fur et à mesure le contenu de chaque fichier.

Pour l'installation de tailwind :

PROJET XTRADH

1. Create a new Vite project running the following commands in your terminal:

```
npm init vite my-project
cd my-project
```

2. Install Tailwind CSS:

```
npm install -D tailwindcss postcss autoprefixer
```

3. Generate the `tailwind.config.js` file by running the following command:

```
npx tailwindcss init -p
```

4. Configure the template paths inside the `tailwind.config.js` file:

```
module.exports = {
  content: [
    "./index.html",
    "./src/**/*.{vue,js,ts,jsx,tsx}",
  ],
  theme: {
    extend: {},
  },
  plugins: [],
}
```

5. Create a new `./src/index.css` file and add the Tailwind directives:

```
@tailwind base;
@tailwind components;
@tailwind utilities;
```

6. Import the newly created CSS file inside your `./src/main.js` file:

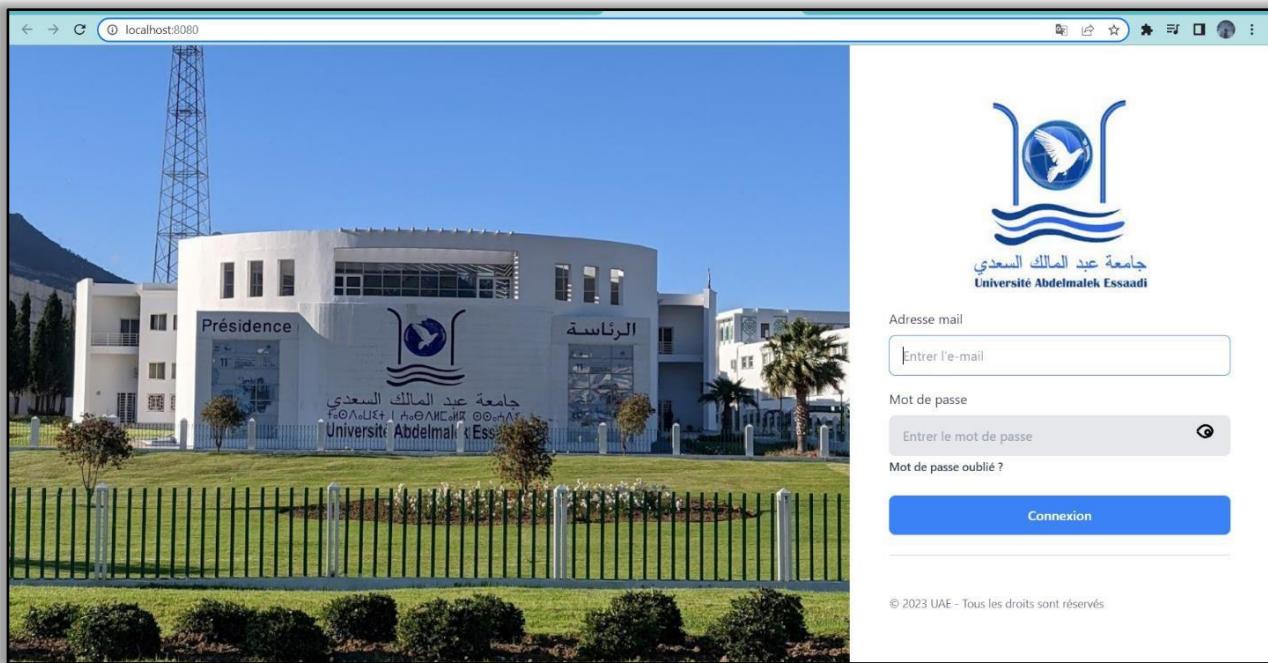
```
import { createApp } from 'vue'
import App from './App.vue'

// add this
import './index.css'

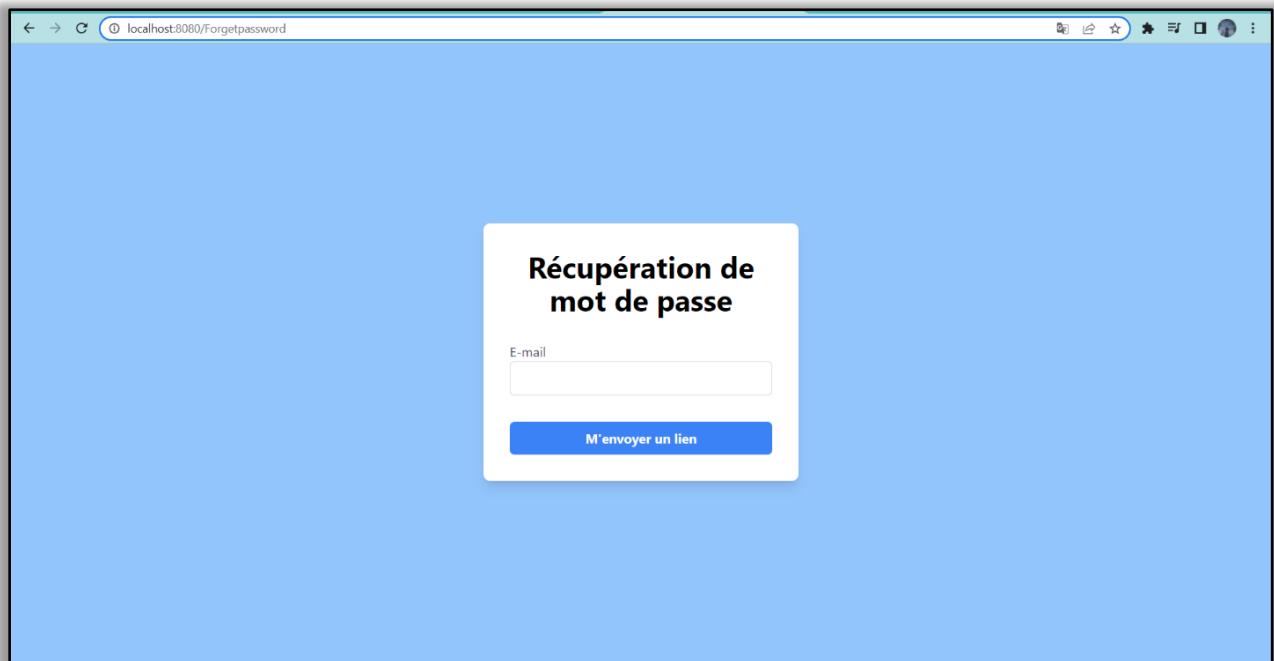
createApp(App).mount('#app')
```

3.4. Connexion -mot de passe oublié- Récupération de mot de passe

La page de Connexion :

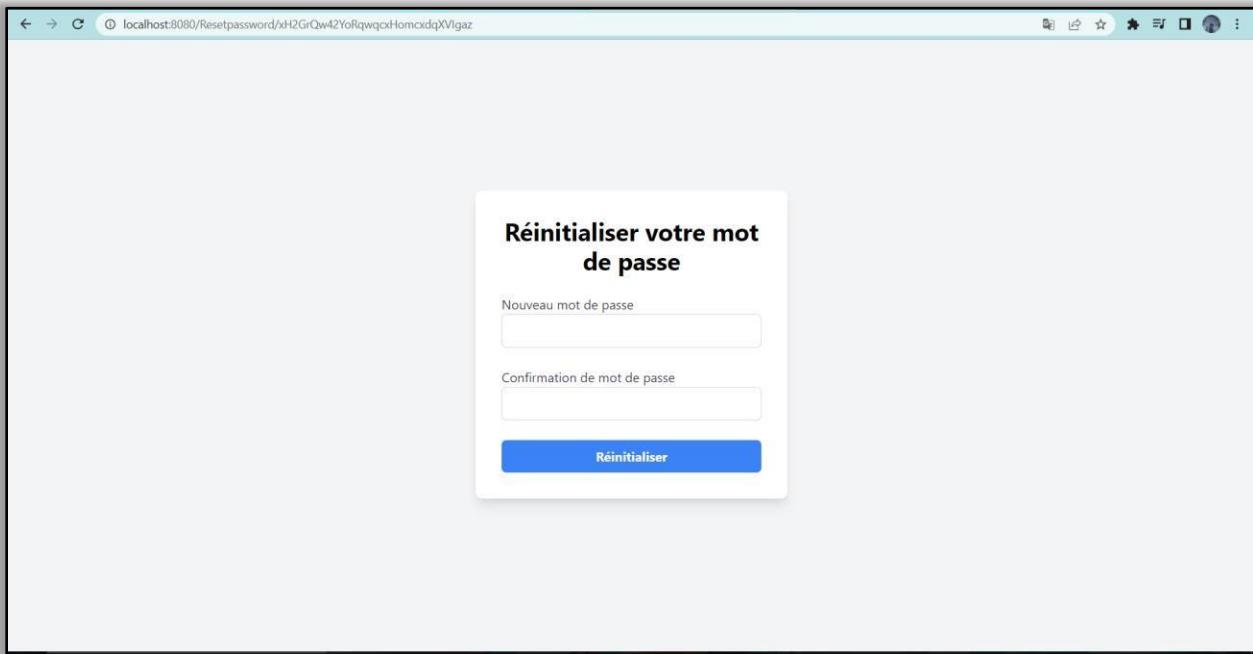


- *Où l'utilisateur peut récupérer son mot de passe au cas d'oubli.*



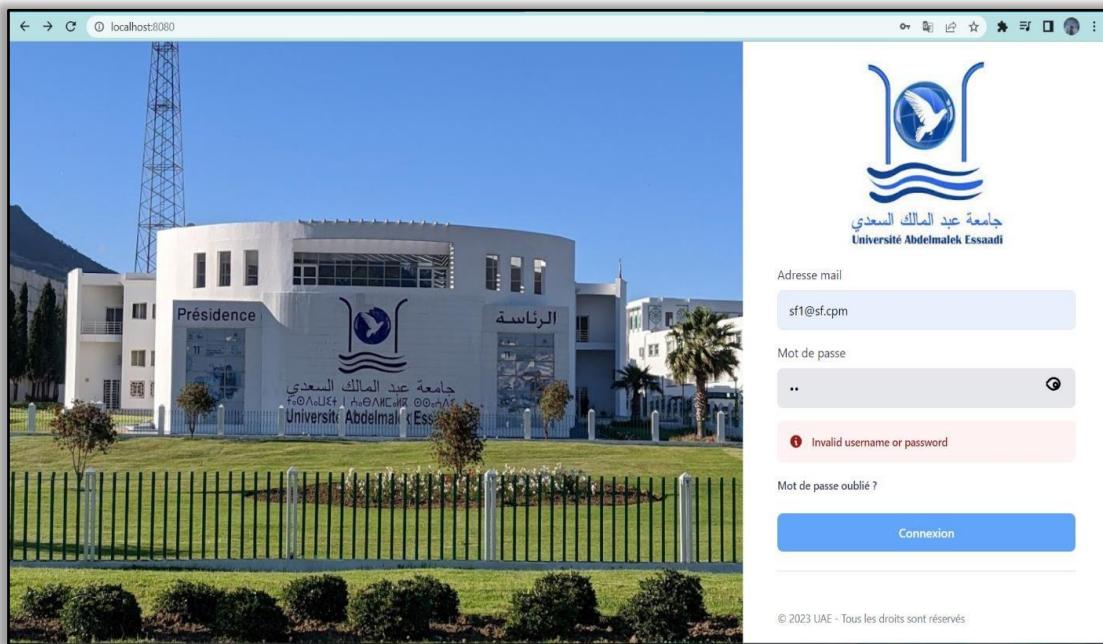
Il faut juste écrire l'adresse email, un email sera envoyé à l'adresse email donné.

- *Après avoir cliqué sur le lien envoyé via mail, l'utilisateur sera redirigé vers :*

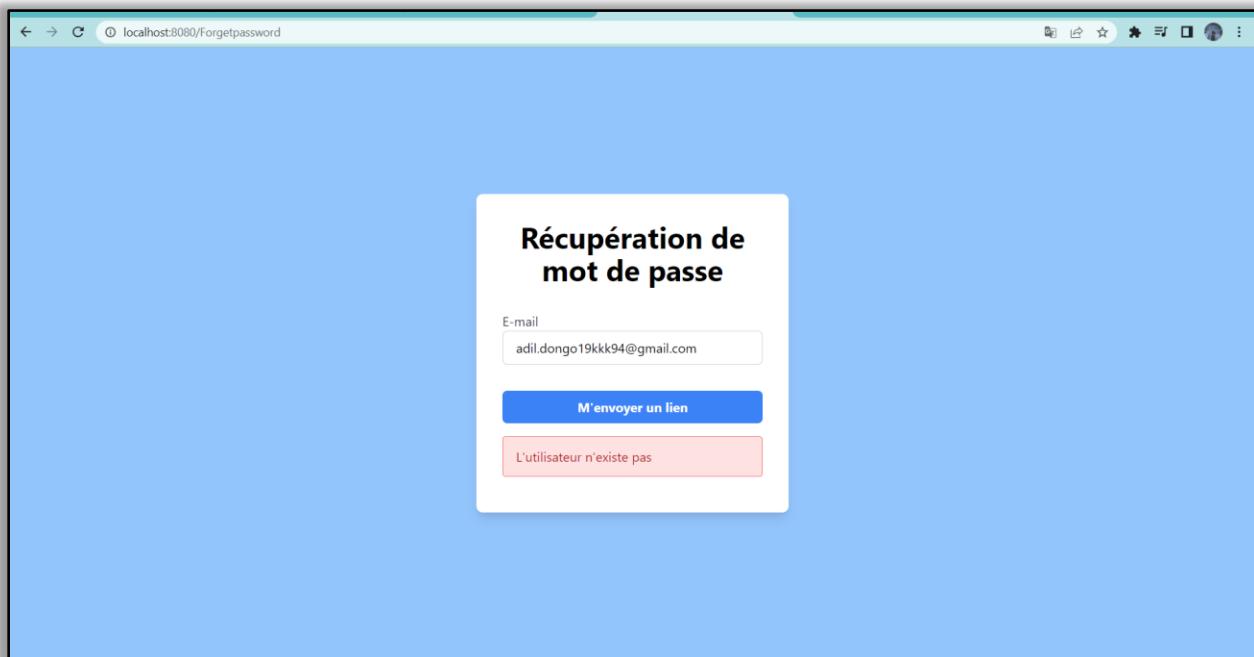


Ici il peut avoir un nouveau mot de passe.

PROJET XTRADH



souligner qu'en cas de saisie d'un email ou mot de passe erroné(login), ou en cas de saisie d'un email non existant dans la base de données(forgotpassword) ou une confirmation de mot de passe erronée (resetpassword), il y'aura apparition des messages informatifs suivants



Une erreur se génère aussi dans le cas de mot de passe et confirmation de mot de passe différents.

- *L'utilisation de vuex, et son rôle dans le login :*

Dans store/auth/index.js:

J'ai mis le token et l'user en state, pour ainsi pouvoir utiliser ces données dans des pages différentes.

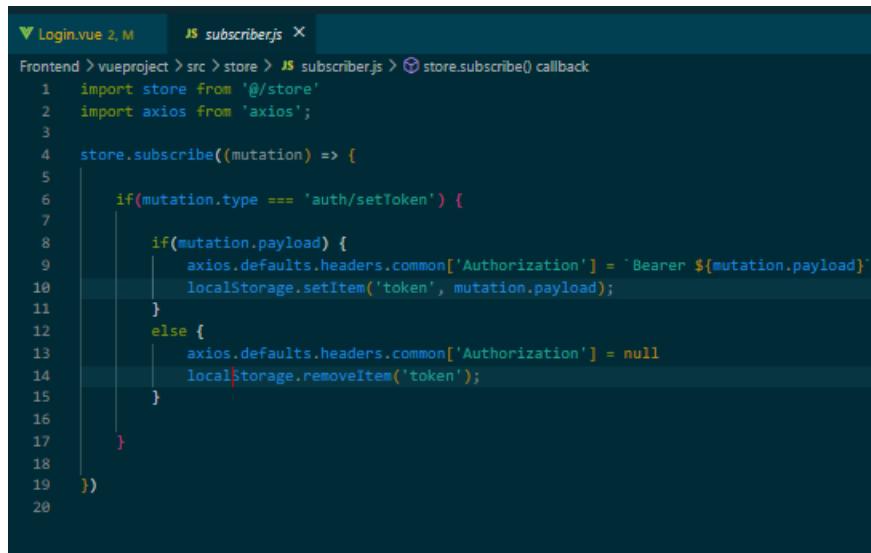
```

Login.vue 2. M   JS index.js  ×
frontend > vueproject > src > store > auth > JS index.js > ...
1  import axios from 'axios'
2
3  export default {
4      namespaced: true,
5      state: {
6          token: null,
7          user: null,
8      },
9      mutations: {
10         setToken(state, token) {
11             state.token = token
12         },
13         setUser(state, data) {
14             state.user = data
15         }
16     },
17     getters: {
18         authenticated(state) {
19             return state.token && state.user;
20         },
21         user(state) {
22             return state.user
23         }
24     },
25     actions: {
26
27         async submit({dispatch},credentials){
28             try {
29                 const response = await axios.post('/login',credentials)
30                 return dispatch('attempt',response.data.token)
31             } catch (error) {
32                 console.log('failed')
33             }
34         },
35
36         async attempt({commit,state}, token) {
37             try {
38                 if(token){
39                     commit('setToken',token )
40                 }
41                 if(!state.token){
42                     return;
43                 }
44                 const response = await axios.get('/user-profile')
45                 commit('setUser', response.data)
46             } catch (error) {
47                 commit('setUser',null )
48                 commit('setToken',null )
49             }
50         },
51
52         logout({commit}){
53             return axios.post('/logout').then(()=>{
54                 commit('setUser',null)
55                 commit('setToken',null)
56             })
57         }
58     }
59 }

```

PROJET XTRADH

Le fichier **store/subscriber.js** :



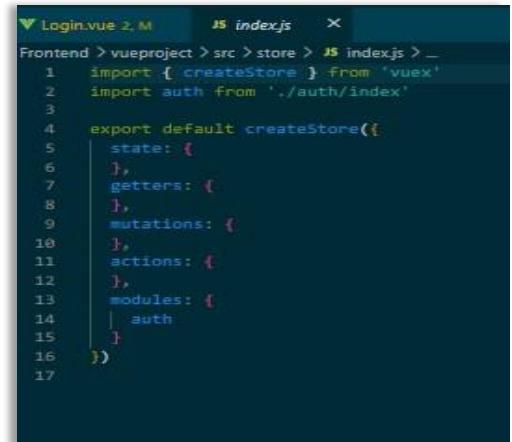
```
▼ Login.vue 2, M   JS subscriber.js ×
Frontend > vueproject > src > store > JS subscriber.js > store.subscribe() callback
1 import store from '@/store'
2 import axios from 'axios';
3
4 store.subscribe((mutation) => {
5
6     if(mutation.type === 'auth/setToken') {
7
8         if(mutation.payload) {
9             axios.defaults.headers.common['Authorization'] = `Bearer ${mutation.payload}`;
10            localStorage.setItem('token', mutation.payload);
11        } else {
12            axios.defaults.headers.common['Authorization'] = null;
13            localStorage.removeItem('token');
14        }
15    }
16
17 }
18
19 })
20
```

Son rôle est d'écouter la requête http entrante, et attribuer un header automatiquement, spécifiquement lorsque on attribue un token (setToken). Le token(Bearer) est stocké ensuite dans localStorage.

Headers sont responsables sur les requêtes http.

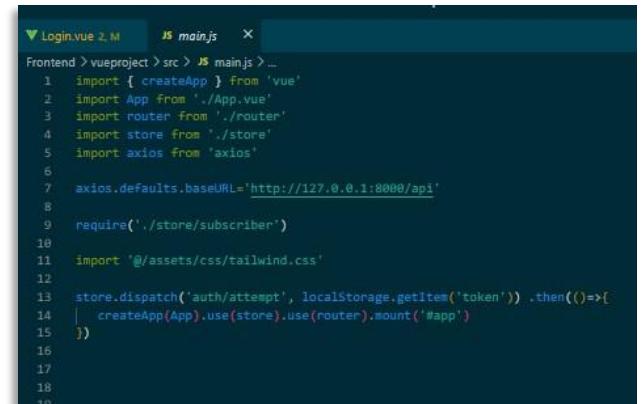
Le fichier **store/index.js** :

Ne contient que le nom du module auth importé.



```
▼ Login.vue 2, M   JS index.js ×
Frontend > vueproject > src > store > JS index.js > ...
1 import { createStore } from 'vuex'
2 import auth from './auth/index'
3
4 export default createStore({
5     state: {},
6     getters: {},
7     mutations: {},
8     actions: {},
9     modules: {
10         auth
11     }
12 })
13
14
15
16
17
```

A signaler que j'ai mis le default url dans **main.js**, afin de directement appeler le nom de l'api sans url(127...)



```
▼ Login.vue 2, M   JS main.js ×
Frontend > vueproject > src > JS main.js > ...
1 import { createApp } from 'vue'
2 import App from './App.vue'
3 import router from './router'
4 import store from './store'
5 import axios from 'axios'
6
7 axios.defaults.baseURL='http://127.0.0.1:8000/api'
8
9 require('@/assets/css/tailwind.css')
10
11 import '@/assets/css/tailwind.css'
12
13 store.dispatch('auth/attemp', localStorage.getItem('token')).then(()=>{
14     createApp(App).use(store).use(router).mount('#app')
15 })
16
17
18
19
```

PROJET XTRADH

Dans le login, la redirection est faite selon le type de l'user, en profitant de vuex. Ci-dessous est la partie script de la page :

```
<script>
import { mapGetters, mapActions } from 'vuex'
import axios from 'axios'
export default (await import('vue')).defineComponent({
  data() {
    return {
      user: {
        email:'',
        password:'',
      },
      error:'',
      showPassword: false,
    }
  },
  computed: {
    passwordFieldType() {
      return this.showPassword ? 'text' : 'password';
    },
    ...mapGetters([
      'authenticated':'auth/authenticated',
      'user':'auth/authenticated'
    ]),
  },
  methods: {
    ...mapActions([
      'submit':'auth/submit'
    ]),
    ...mapGetters([
      'authenticated':'auth/authenticated',
      'user':'auth/authenticated'
    ]),
    | togglePasswordVisibility() {
      this.showPassword = !this.showPassword;
    },
    async submitlogin() {
      try {
        this.error = '' // Clear the error message
        await this.$store.dispatch('auth/submit', this.user);
        if (this.$store.getters['auth/authenticated']) {
          const userType = this.$store.getters['auth/user'].type;
          switch (userType) {
            case 'prof':
              this.$router.push('Dash_users');
              break;
            case 'admin_univ':
              this.$router.push('Dash_au');
              break;
            case 'president_univ':
              this.$router.push('Dash_pu');
              break;
            case 'admin_etb':
              this.$router.push('Dash_ae');
              break;
            case 'directeur_etb':
              this.$router.push('Dash_de');
              break;
            default :
              this.$router.push('/')
          }
        } else {
          this.error = 'Invalid username or password';
        }
      } catch (error) {
        console.log('failed', error);
        this.error = 'An error occurred during login';
      }
    }
  }
})
```

mapGetters et mapActions sont utilisés, pour importer les méthodes nécessaires d'après store.

La page forgotpassword :

```

Frontend > vueproject > src > Pages > auth > ▼ Forgetpassword.vue > () script > default
  1  <template>
  2    <div class="flex justify-center items-center h-screen bg-blue-300">
  3      <div class="w-96 bg-white rounded-lg shadow-lg p-8">
  4        <h1 class="text-4xl font-bold mb-8 text-center">Forgot Password</h1>
  5        <form @submit.prevent="submitForm()" class="space-y-8">
  6          <div>
  7            <label for="email" class="block text-gray-700">Email</label>
  8            <input v-model="email" id="email" type="email" class="w-full border border-gray-300 focus:outline focus:outline-blue-200 rounded-md px-4 py-2" required>
  9          </div>
 10          <div class="flex justify-between items-center">
 11            <button @click="" type="submit" class="w-full bg-blue-500 text-white font-bold rounded-md py-2 px-4 hover:bg-blue-600">Email Me a link</button>
 12          </div>
 13        </form>
 14        <div v-if="isLoading" class="fixed top-0 left-0 right-0 bottom-0 flex items-center justify-center z-20">
 15          <div class="animate-spin rounded-full h-16 w-16 border-t-2 border-b-2 border-gray-900"></div>
 16        </div>
 17
 18        <div class="bg-red-100 border border-red-400 text-red-700 px-4 py-3 rounded relative" role="alert" v-show="error">
 19          <span class="block sm:inline" >{{error}}</span>
 20        </div>
 21      </div>
 22    </template>
 23
 24
 25  <script>
 26    import axios from 'axios';
 27    import Header from '@/components/Login/Header.vue'
 28    import Footer from '@/components/Login/Footer.vue'
 29    export default{await import('vue')).defineComponent({
 30      components:{Footer,Header},
 31      data() {
 32        return {
 33          email: '',
 34          isLoading:'',
 35          error:''
 36        };
 37      },
 38      methods: {
 39        async submitForm() {
 40          try {
 41            const response = await axios.post('/forgot', {
 42              email: this.email
 43            });
 44            const redirectTime = 5000; // Redirect after 5 seconds (5000 milliseconds)
 45            setTimeout(() => {
 46              this.$router.push('/Wait'); // Replace with your desired route path
 47            }, redirectTime);
 48            if (this.email !== '') {
 49              this.isLoading = true;
 50            }
 51          } catch (error) {
 52            this.error = error.response.data.message;
 53            console.log(error.response.data.message);
 54          }
 55        },
 56      }
 57    })
 58  </script>

```

Lorsque l'utilisateur saisit son adresse e-mail, le système vérifie si cette adresse est déjà présente dans la base de données. Si c'est le cas, un e-mail lui est envoyé et il est redirigé vers la page de connexion. En consultant cet e-mail, il trouvera un lien qui le redirigera vers l'endroit où il pourra modifier son mot de passe.

PROJET XTRADH

La page **wait.vue** :



L'utilisateur sera notifié par cette page, si l'adresse email existe dans la base de données.

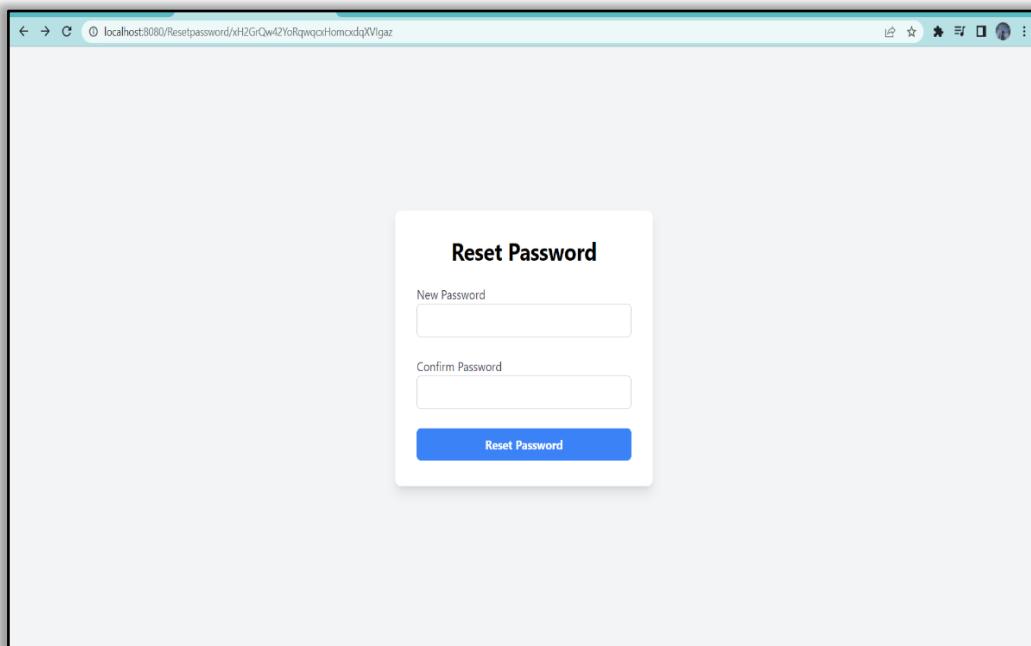
```
Frontend > vueproject > src > Pages > auth > ▼ Wait.vue > {} script > default > methods
  1  <template>
  2    <div class="flex justify-center items-center h-screen bg-blue-400">
  3      <div class="w-96 bg-white rounded-lg shadow-lg p-8">
  4        <h1 class="text-4xl font-bold mb-8 text-center">Wait Please an email will be sent, check your email </h1>
  5        <button @click="startRedirect" type="submit" class="w-full bg-blue-500 text-white font-bold rounded-md py-2 px-4 hover:bg-blue-600">Back to login</button>
  6        <div v-if="isLoading" class="fixed top-0 left-0 right-0 bottom-0 flex items-center justify-center z-20">
  7          <div class="animate-spin rounded-full h-16 w-16 border-t-2 border-b-2 border-gray-900"></div>
  8        </div>
  9      </div>
 10    </div>
 11  </template>
 12
 13 <script>
 14   import Header from '@/components/Login/Header.vue'
 15   import Footer from '@/components/Login/Footer.vue'
 16   import axios from 'axios';
 17   export default({await import('vue')).defineComponent({
 18     components:{Footer,Header},
 19     data() {
 20       return {
 21         email: '',
 22         isLoading: false,
 23       };
 24     },
 25     methods:{
 26       startRedirect() {
 27         const redirectTime = 2000; // Redirect after 5 seconds (5000 milliseconds)
 28         setTimeout(() => {
 29           this.$router.push('/'); // Replace with your desired route path
 30         }, redirectTime);
 31         this.isLoading=!this.isLoading
 32       },
 33     }
 34   })
 35
 36 </script>
 37
```

La page Resetpassword :

```

1 <template>
2   <div class="flex justify-center items-center h-screen bg-gray-100">
3     <div class="w-96 bg-white rounded-lg shadow-lg p-8">
4       <h1 class="text-3xl font-bold mb-6 text-center">Reset Password</h1>
5       <form @submit.prevent="submitForm" class="space-y-6">
6         <div>
7           <label for="password" class="block text-gray-700">New Password</label>
8           <input v-model="password" id="password" type="password" class="w-full border border-gray-300 focus:ring focus:ring-blue-200 rounded-md px-4 py-2">
9         </div>
10        <div>
11          <label for="confirmPassword" class="block text-gray-700">Confirm Password</label>
12          <input v-model="confirmPassword" id="confirmPassword" type="password" class="w-full border border-gray-300 focus:ring focus:ring-blue-200 rounded-md px-4 py-2">
13        </div>
14
15        <div class="flex justify-between items-center">
16          <button type="submit" class="w-full bg-blue-500 text-white font-bold rounded-md py-2 px-4 hover:bg-blue-600">Reset Password</button>
17        </div>
18
19        <div class="bg-red-100 border border-red-400 text-red-700 px-4 py-3 rounded relative" role="alert" v-show="error">
20          <span class="block sm:inline" >{{error}}</span>
21          <span class="absolute top-0 bottom-0 right-0 px-4 py-3">
22            </span>
23        </div>
24      </form>
25    </div>
26  </template>
27
28 <script>
29
30   import axios from 'axios';
31   import Header from '@/components/Login/Header.vue'
32   import Footer from '@/components/Login/Footer.vue'
33   import { createLogger } from 'vuex';
34   export default (await import('vue')).defineComponent({
35     components:{Header,Footer},
36
37     data() {
38       return {
39         email: '',
40         password: '',
41         confirmPassword: '',
42         error: ''
43       };
44     },
45     methods: {
46       async submitForm() {
47         try{
48           let response = await axios.post('/reset',{
49             password : this.password,
50             password_confirm : this.confirmPassword,
51             token: this.$route.params.token,
52           });
53           this.$router.push('/')
54         }catch(error){
55           //console.log(error.response.data.errors.password_confirm[0])
56           this.error = error.response.data.errors.password_confirm[0]
57         }
58       }
59     }
60   );
61 </script>

```



Si le mot de passe entré est bien correct (identique dans les deux champs) alors l'utilisateur sera redirigé vers le login où il pourra s'authentifier de façon normale avec son email, et le nouveau mot de passe.

3.5. Les interfaces :

3.5.1. Admin D'établissement

La page **Gestionp**, gestion des profs (ajouter, modifier supprimer) :

PPR	NOM	PRENOM	DATE DE NAISSANCE	ETABLISSEMENT	GRADE	TYPE	Ajouter
1000000	Lahmidi	Sara	1988-07-08	Faculté des Lettres et des Sciences Humaines	Professeur Agrégé	admin_etb	<button>Modifier</button> <button>Supprimer</button>
1234	benani	smires	2023-06-14	Faculté des Lettres et des Sciences Humaines	Professeur Habilite	prof	<button>Modifier</button> <button>Supprimer</button>
666666	Soufian	yassine	1964-05-27	Faculté des Lettres et des Sciences Humaines	Professeur Habilite	prof	<button>Modifier</button> <button>Supprimer</button>

```

<th class="px-4 py-3"> TYPE </th>
<th scope="col" class="py-3 px-2">
|   <div class="flex justify-center" >
|   |   <Create :test="test" />
|   </div>
</th>
</tr>

```

Ce composant contient :

```
<template>
  <div>
    <button @click="showModal = true" class="bg-green-500 hover:bg-blue-700 text-white font-bold py-3 px-2 rounded">
      Ajouter
    </button>

    <div class="modal fixed w-full h-full top-0 left-0 flex items-center justify-center" v-if="showModal">
      <div class="modal-overlay absolute w-full h-full bg-gray-900 opacity-50"></div>
```

- Un bouton qui inverse la valeur de variable showModal , du coup , si elle est vraie , la condition v-if= showModal est vraie , et donc le contenu s'affiche :

```
<div class="modal fixed w-full h-full top-0 left-0 flex items-center justify-center" v-if="showModal">
  <div class="modal-overlay absolute w-full h-full bg-gray-900 opacity-50"></div>

  <div class="modal-content bg-white rounded-lg p-6 max-w-2xl mx-auto">
    <span class="close absolute top-0 right-0 m-4 cursor-pointer" @click="closeModal">&times;</span>

    <div class="mb-4">
      <label for="PPR" class="block text-gray-700 font-bold mb-2">PPR:</label>
      <input type="text" id="PPR" v-model="formData.PPR" required class="border rounded w-full py-2 px-3">
    </div>

    <div class="mb-4">
      <label for="Nom" class="block text-gray-700 font-bold mb-2">Nom:</label>
      <input type="text" id="Nom" v-model="formData.Nom" required class="border rounded w-full py-2 px-3">
    </div>

    <div class="mb-4">
      <label for="#Prenom" class="block text-gray-700 font-bold mb-2">Prenom:</label>
      <input type="text" id="#Prenom" v-model="formData.Prenom" required class="border rounded w-full py-2 px-3">
    </div>

    <div class="mb-4">
      <label for="Email" class="block text-gray-700 font-bold mb-2">Email:</label>
      <input type="email" id="Email" v-model="formData.Email" required class="border rounded w-full py-2 px-3">
    </div>

    <div class="mb-4">
      <label for="DATE_NAISSANCE" class="block text-gray-700 font-bold mb-2">DATE_NAISSANCE:</label>
      <input type="date" id="DATE_NAISSANCE" v-model="formData.Date_Naissance" required class="border rounded w-full py-2 px-3">
    </div>

    <div class="mb-4">
      <label for="type" class="block text-gray-700 font-bold mb-2">Grade:</label>
      <select id="type" v-model="formData.grade" required class="border rounded w-full py-2 px-3">
        <option selected value="" disabled>Selectionner le grade</option>
        <option :value="grad.id_Grade" selected v-for="grad in grads" :key="grad.id_Grade">{{ grad.designation }}</option>
      </select>
    </div>
    <div class="mb-4" v-if="show">
      <label for="type" class="block text-gray-700 font-bold mb-2">Etablissement Origine:</label>
      <select id="type" v-model="formData.Etablissement" required class="border rounded w-full py-2 px-3">
        <option selected value="" disabled>Selectionner etablissement </option>
        <option v-for="etb in formData.etabs" :key="etb.id" :value="etb.id">{{ etb.Nom }}</option>
      </select>
    </div>
  </div>
```

Plus d'inputs, il y a deux select options responsables sur le grade et l'établissement.

- Puis il y'a :

```


<button class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded"
    @click="submitForm">
    Ajouter
  </button>
  <button class="bg-red-500 hover:bg-red-700 text-white font-bold py-2 px-4 rounded ml-2"
    @click="closeModal">
    Annuler
  </button>
  {{ errors }}
</div>


```

Un bouton qui permet la soumission de formulaire, et un autre qui permet de cacher le modal, en exécutant la fonction closeModal , qui inverse la valeur de showModal.

- Maintenant on va voir la partie script :

```

<script>
import axios from 'axios'
export default(await import('vue')).defineComponent({
  props : {
    test : {
      type : Number
    }
  },
  data(){
    return {
      showModal: false,
      formData:{
        PPR: '',
        Nom : '',
        Prenom : '',
        Date_Naissance: '',
        grade : '',
        email : '',
        type : 'prof',
        grads : '',
        errors : '',
        Etablissement : '',
        etabs : []
      },
      show : 0
    }
  }
})

```

Dans cette partie de code, on initialise les variables à utiliser dans le formulaire .

```

async created(){
  this.show = this.test
  console.log( this.formData.show )
  const grads = await axios.get('/grade');
  this.grads = grads.data ;
  const etbs = await axios.get(['etablissement']);
  // console.log(etbs.data)
  etbs.data.forEach(e => {
    if(e.Nom !='UAE'){
      this.formData.etabs.push(e)
    }
  });
}

```

- Ici on a un hook de cycle de vie, qui s'appelle created, elle est appelée avant le que le composant ne soit monté dans le DOM. On consomme ici l'Api grade et établissement leurs contenu s'affiche dans select option citée précédemment.

Modifier est une vue, sa route est définie dans le router.

```
<td class="px-4 py-3 text-ms font-semibold border" >{{ data.user.type }}</td>
<td class="px-4 py-3 text-ms font-semibold border" >{{ data.grade.designation }}</td>
<td class="px-4 py-3 text-ms font-semibold border" >{{ data.user.type }}</td>
<td class="border-solid border-2 border-gray-100 py-4 px-6 text-right">
  <div class="inline-flex">
    <div class="py-4 px-6 text-right">
      <div class="inline-flex">
        <a href="/Gestionp/Edit/${data.id}" class="bg-blue-500 hover:bg-blue-400 text-white font-bold py-2 px-4 mx-2 rounded-full">
          Editer
        </a>
        <button class="bg-red-500 hover:bg-blue-400 text-white font-bold py-2 px-4 rounded-full" @click="deleteprof(data)">
          Supprimer
        </button>
      </div>
    </div>
  </div>
</td>
```

Cette vue contient :

Un formulaire, où les données de l'utilisateur qu'on veut modifier vont s'afficher.

```
<template>
  <form @submit.prevent="submitForm" >

    <div class="relative z-0 w-full mb-6 group">
      <label for="PPR" class="block text-gray-700 font-bold mb-2">PPR:</label>
      <input type="text" id="PPR" v-model="formData.PPR" required class="border rounded w-full py-2 px-3">
    </div>

    <div class="relative z-0 w-full mb-6 group">
      <label for="Nom" class="block text-gray-700 font-bold mb-2">Nom:</label>
      <input type="text" id="Nom" v-model="formData.Nom" required class="border rounded w-full py-2 px-3">
    </div>

    <div class="relative z-0 w-full mb-6 group">
      <label for="Prenom" class="block text-gray-700 font-bold mb-2">Prenom:</label>
      <input type="text" id="Prenom" v-model="formData.Prenom" required class="border rounded w-full py-2 px-3">
    </div>

    <div class="relative z-0 w-full mb-6 group">
      <label for="Email" class="block text-gray-700 font-bold mb-2">Email:</label>
      <input type="Email" id="Email" v-model="formData.Email" required class="border rounded w-full py-2 px-3">
    </div>

    <div class="relative z-0 w-full mb-6 group">
      <label for="DATE_NAISSANCE" class="block text-gray-700 font-bold mb-2">Date_naissance:</label>
      <input type="date" id="DATE_NAISSANCE" v-model="formData.Date_Naissance" required class="border rounded w-full py-2 px-3">
    </div>

    <button type="submit" class="text-white bg-blue-700 hover:bg-blue-800 focus:ring-4 focus:outline-none focus:ring-blue-300 font-medium py-2 px-4 w-full rounded-lg">Enregistrer</button>
  </form>
</template>
```

Et contient : la partie script

Les données initialisées, à utiliser dans le formulaire :

```
<script>
import axios from 'axios';
import { createLogger } from 'vuex';

export default (await import('vue')).defineComponent({
  data() {
    return {
      formData: {
        id_user: '',
        PPR: '',
        Nom: '',
        Prenom: '',
        Date_Naissance: '',
        Email: '',
      },
      prev_email: ''
    };
  }
});
```

La méthode **submitForm**, envoi les données modifiées par l'utilisateur, et permet la redirection vers la page Gestionp :

```
methods: {
  async submitForm() {
    try {
      const response = await axios.patch('/updateprof/' + this.formData.id_user, {
        PPR: this.formData.PPR,
        Nom: this.formData.Nom,
        prenom: this.formData.Prenom,
        Date_Naissance: this.formData.Date_Naissance,
        id_user: this.formData.id_user,
        email: this.formData.Email,
      });
      console.log(response);

      this.$router.push('/Gestionp')

    } catch (error) {
      console.error(error);
      // Handle error
    }
  }
},
```

De plus :

Un hook de cycle de vie(mounted)

Permettant l'interaction avec l'api

Une fois le composant est monté dans le DOM

```
  },
  async mounted(){
    let id = this.$route.params.id
    this.formData.id_user=id
    console.log(this.$route.params.id)
    const response = await axios.get('/enseignant/'+this.formData.id_user)
    console.log(response.data)
    this.formData.PPR = response.data.PPR ;
    this.formData.Nom = response.data.Nom;
    this.formData.Prenom = response.data.prenom ;
    this.formData.Date_Naissance = response.data.Date_Naissance ;
    this.formData.Email = response.data.user.email ;
    this.prev_email = response.data.user.email ;
    this.formData.id_user = response.data.user.id_user;
  },
  cancelForm() {
  },
}
</script>
```

PROJET XTRADH

Et donc on charge les variables initialisées qui vont être affichées, par les valeurs de champ de l'utilisateur qu'on veut modifier, par l'id_user.

Et la page **Gestionp** principale :

```
<template>
<section class="container mx-auto p-6 font-mono">
  <h3 class="text-2xl font-serif text-left pb-4">Listes des profs</h3>
  <div class="w-full mb-8 overflow-hidden rounded-lg ">
    <div class="w-full overflow-x-auto overflow-y-auto h-[calc(100vh-200px)] scrollbar [& scrollbar-track-gray-100 ">
      <table class="w-full">
        <thead>
          <tr class="text-md font-semibold tracking-wide text-left [& text-gray-900 [& bg-gray-100 uppercase border-b [& border-gray-600">
            <th class="px-4 py-3">PPR</th>
            <th class="px-4 py-3">NOM</th>
            <th class="px-4 py-3">PRENOM</th>
            <th class="px-4 py-3">DATE DE NAISSANCE</th>
            <th class="px-4 py-3">ETABLISSEMENT</th>
            <th class="px-4 py-3">GRADE</th>
            <th class="px-4 py-3">TYPE</th>
            <th scope="col" class="py-3 px-2">
              <div class="flex justify-center" >
                <&Create :test="test" />
              </div>
            </th>
          </tr>
        </thead>
        <tbody class=" [& bg-white">
          <tr class=" [& text-gray-700" v-for="data in profs " :key="data.id">
            <td class="px-4 py-3 border">
              <div class="flex items-center text-sm">
                <div>
                  <p class="font-semibold [& text-black">{{ data.PPR }} </p>
                </div>
              </div>
            </td>
            <td class="px-4 py-3 text-ms font-semibold border">{{ data.Nom }}</td>
            <td class="px-4 py-3 text-ms font-semibold border">
              {{ data.prenom }}
            </td>
            <td class="px-4 py-3 text-ms font-semibold border">{{ data.Date_Naissance }}</td>
            <td class="px-4 py-3 text-ms font-semibold border">{{ data.etab_permanant.Nom }}</td>
            <td class="px-4 py-3 text-ms font-semibold border">{{ data.grade.designation }}</td>
            <td class="px-4 py-3 text-ms font-semibold border">{{ data.user.type }}</td>
            <td class="border-solid border-2 [& border-gray-100 py-4 px-6 text-right">
              <div class="inline-flex">
                <div class="py-4 px-6 text-right">
                  <div class="inline-flex">
                    <router-link :to="/Gestionp/Edit/{{data.id}}" class=" [& bg-blue-500 [& hover:bg-blue-400 [& text-white font-bold py-2 px-4 mx-2 rounded-full">
                      Editer
                    </router-link>
                    <button class=" [& bg-red-500 hover:bg-blue-400 [& text-white font-bold py-2 px-4 rounded-full " @click="deleteprof(data)">
                      Supprimer
                    </button>
                  </div>
                </div>
              </div>
            </td>
          </tr>
        </tbody>
      </table>
    </div>
  </div>
</section>
</template>
```

Partie script :

```

<script>
import axios from 'axios';
import Create from '@/components/Dashboard/Prof/Create.vue';

export default (await import('vue')).defineComponent({
  components: { Create },
  data() {
    return {
      profs: null,
      showModal: false,
      test : 0
    };
  },
  methods: {
    async deleteprof(id){
      console.log(id.user.id_user)
      const response = await axios.delete('/deleteprof/'+id.user.id_user)
      console.log(response)
      let index = this.profs.indexOf(id)
      this.profs.splice(index,1)
    }
  },
  async mounted() {
    try {
      await axios.get('/profetab').then(res => {
        console.log(res)
        this.profs=res.data
      })
      this.showModal = false;
    } catch (error) {
      console.error(error);
    }
  }
})
</script>

```

On retrouve aussi la méthode delete , qui permet de supprimer un enregistrement selon l'id de l'utilisateur. Et ici mounted est responsable de l'affichage des valeurs de champ d'utilisateur, une fois le composant est monté, Via l'API profetab.

A signaler, que dans La vue Gestion des profs, l'administrateur d'établissement peut gérer les professeurs travaillant dans le même établissement.

PROJET XTRADH

La page **Gestionde** : gestion des directeurs (ajouter , modifier supprimer) :

The screenshot shows a web application interface for managing directors. On the left, a sidebar titled "Admin D'établissement" contains links for "Gestion des directeurs", "Gestion des professeurs", "Gestion des Interventions", "Profil", and "Déconnexion". The main content area is titled "Listes des directeurs d'établissement". It displays a table with columns: PPR, NOM, PRENOM, EMAIL, ETABLISSEMENT, and TYPE. A green "Ajouter" button is at the top right of the table. Below the table, there is one row of data:

PPR	NOM	PRENOM	EMAIL	ETABLISSEMENT	TYPE
666670	Bouktaib	Soufiane	Soufianbouktaib1@gmail.com	Faculté des Lettres et des Sciences Humaines	directeur_etb

Below the table are two buttons: "Modifier" (blue) and "Supprimer" (red).

Même principe de modifier et ajouter cités précédemment, les noms d'API sont les seuls qui changent.

Maintenant on arrive à la page **Gestional** (Gestion des interventions), où l'administrateur peut attribuer une intervention à un professeur , la modifier ou cas où elle est pas validé par directeur établissement.

The screenshot shows a web application interface for managing interventions. On the left, a sidebar titled "Admin D'établissement" contains links for "Gestion des directeurs", "Gestion des professeurs", "Gestion des Interventions", "Profil", and "Déconnexion". The main content area is titled "Gestion des Interventions". It displays a table with columns: NOM, PRENOM, INTERVENTION, ETAB, ANNEE, SEMESTRE, NBR_HRS, DEBUT, FIN, and a green "ajouter" button. Below the table, there are two rows of data:

NOM	PRENOM	INTERVENTION	ETAB	ANNEE	SEMESTRE	NBR_HRS	DEBUT	FIN
Lahmidi	Sara	Séminaire sur la gestion des ressources humaines	Faculté des Lettres et des Sciences Humaines	2022/2023	Semestre 2	15	2023-03-05	2023-03-07
Soufian	yassin	javascript_js	Faculté des Lettres et des Sciences Humaines	2021/2023	Semestre 1	30	2023-06-22	2023-06-24

Below each row are two buttons: "Modifier" (blue) and "Supprimer" (red).

La partie template , contient :

```

template:
<section class="container w-full mx-auto p-6 font-mono">
  <h3 class="text-2xl font-serif text-left pb-4">Gestion des Interventions</h3>
  <div class="w-full mb-8 overflow-hidden rounded-lg">
    <div class="w-full overflow-x-auto overflow-y-auto h-[calc(100vh-200px)] scrollbar [&scrollbar-track-gray-200]">
      <table class="w-full">
        <thead>
          <tr class="text-sm tracking-wide text-left [&text-gray-900 [&bg-gray-100 uppercase border-b [&border-gray-600]]">
            <th class="px-2 py-3">Nom</th>
            <th class="px-2 py-3">Prénom</th>
            <th class="px-2 py-3">Intervention</th>
            <th class="px-2 py-3">Etab.</th>
            <th class="px-2 py-3">Année</th>
            <th class="px-2 py-3">Semestre</th>
            <th class="px-2 py-3">Nbr_heures</th>
            <th class="px-2 py-3">début</th>
            <th class="px-2 py-3">fin</th>
            <th scope="col" class="py-3 px-2">
              <div class="flex justify-center">
                <a href="#"><router-link to="/GestionAI/Create"></router-link>
                <button class=" [&bg-green-500 [&hover:bg-blue-700 [&text-white font-bold py-3 px-2 rounded]]">
                  ajouter
                </button>
              </div>
            </th>
          </tr>
        </thead>
        <tbody class=" [&bg-white]">
          <tr class=" [&text-gray-700]" v-for="data in interv " :key="data">
            <td class="px-2 py-3 text-sm border">{{ data.prof_nom }}</td>
            <td class="px-2 py-3 text-sm border">{{ data.prenom }}</td>
            <td class="px-2 py-3 border">
              <div class="flex items-center text-sm">
                <div>
                  <p class=" [&text-black]">{{ data.Intitule_Intervention }}</p>
                </div>
              </div>
            </td>
            <td class="px-2 py-3 text-sm border">{{ data.etab }}</td>
            <td class="px-2 py-3 text-sm border">{{ data.Annee_univ }}</td>
            <td class="px-2 py-3 text-sm border">{{ data.Semestre }}</td>
            <td class="px-2 py-3 text-sm border text-center">{{ data.Nbr_heures }}</td>
            <td class="px-2 py-3 text-sm border">{{ data.Date_debut }}</td>
            <td class="px-2 py-3 text-sm border">{{ data.Date_fin }}</td>

            <td class="border-solid border-2 [&border-gray-100 py-4 px-2 text-right">
              <div class="inline-flex">
                <div class="py-4 px-2 text-right">
                  <div class="inline-flex">
                    <a href="#"><router-link :to="`GestionAI/Edit/${data.id_intervention}`" class=" [&bg-blue-500 [&hover:bg-blue-400 [&text-white font-bold py-2 px-4 rounded-full]]">
                      Modifier
                    </router-link>
                    <button class=" [&bg-red-500 hover:bg-blue-400 [&text-white font-bold py-2 px-4 rounded-full]" @click="deleteinterv(data)">
                      Supprimer
                    </button>
                  </div>
                </div>
              </div>
            </td>
          </tr>
        </tbody>
      </table>
    </div>
  </div>

```

Ajouter et modifier sont ici des routes .

PROJET XTRADH

The screenshot shows a web application interface for managing interventions. On the left, a sidebar titled 'Admin D'établissement' contains links for 'Gestion des directeurs', 'Gestion des professeurs', '+ Gestion des Interventions' (which is expanded), and 'Profil'. At the bottom of the sidebar is a 'Déconnexion' button. The main content area has a title 'Modifier l'intervention' (Edit Intervention). It includes fields for 'Intitulé d'intervention' (Séminaire sur la gestion des ressources humaines), 'Nombre d'heures' (15), 'Année Scolaire' (2022/2023), 'Semestre' (Semestre 2), 'Faculté' (Faculté des Lettres et des Sciences Humaines), and date fields for 'Date Début' (03/05/2023) and 'Date Fin' (03/07/2023). A blue 'Soumettre' (Submit) button is at the bottom.

The screenshot shows a web application interface for adding a new intervention. The sidebar is identical to the previous screenshot. The main content area has a title 'Ajouter une intervention' (Add Intervention). It includes fields for 'PPR', 'Intitulé d'intervention', 'Nombre d'heures', 'Année Scolaire', 'Sélectionnez une semestre', and date fields for 'Date Début' (mm/dd/yyyy) and 'Date Fin' (mm/dd/yyyy). A blue 'Soumettre' (Submit) button is at the bottom.

Voici les deux routes.

Suite de Gestionde :

par axios , on récupère les données par get de l'api directeuretabintervall.

Et on supprime avec l'api intervention en utilisant la méthode delete.

```
<script>

import axios from 'axios';
export default [await import('vue')).defineComponent({
  components: {},
  data(){
    return {
      | | interv : []
    }
  },
  async mounted()
  {
    const response = await axios.get('/directeuretabintervall')
    this.interv = response.data
    console.log(this.interv)
  },
  methods:{
    async deleteinterv(data)
    {
      const response = axios.delete('/intervention/'+data.id_intervention).then(()=>{
        let index = this.interv.indexOf(data);
        console.log(index)
        this.interv.splice(index,1);
      });
      console.log(response)

    }
  }
}
</script>
```

On passe maintenant à la vue profile :

The screenshot shows a web application interface for managing profiles. On the left, there's a sidebar with a dark blue background and white text. It includes links for 'Gestion des directeurs', 'Gestion des professeurs', 'Gestion des Interventions', and 'Profil'. At the bottom of the sidebar is a 'Déconnexion' (Logout) button. The main content area has a light blue header with the title 'Modifier votre profil'. Below the header, there are five input fields: 'Nom:' (Abdo), 'Prénom:' (El harrachy), 'Email:' (abdotest2023@gmail.com), 'PPR:' (666669), and 'Établissement:' (Faculté des Lettres et des Sciences Humaines). A blue 'Modifier' (Update) button is located at the bottom of the form. In the top right corner of the main window, there's a user profile icon and the email 'abdotest2023@gmail.com' followed by an 'Admin' status indicator.

Les données de l'utilisateur connectés s'affiche automatiquement, il peut aussi les modifier, via un modal.

Voila la partie template :

PROJET XTRADH

```
<template>
<form @submit.prevent="submitForm" >

  <div class="relative z-0 w-full mb-6 group">
    <label for="nom" class="block text-gray-700 font-bold mb-2">Nom:</label>
    <input type="text" id="nom" v-model="formData.Nom" required class="border rounded w-full py-2 px-3" disabled>
  </div>

  <div class="relative z-0 w-full mb-6 group">
    <label for="prenom" class="block text-gray-700 font-bold mb-2">Prénom:</label>
    <input type="text" id="prenom" v-model="formData.prenom" required class="border rounded w-full py-2 px-3" disabled>
  </div>

  <div class="relative z-0 w-full mb-6 group">
    <label for="email" class="block text-gray-700 font-bold mb-2">Email:</label>
    <input type="text" id="email" v-model="formData.email" required class="border rounded w-full py-2 px-3" disabled>
  </div>

  <div class="relative z-0 w-full mb-6 group">
    <label for="PPR" class="block text-gray-700 font-bold mb-2">PPR:</label>
    <input type="text" id="PPR" v-model="formData.PPR" required class="border rounded w-full py-2 px-3" disabled>
  </div>

  <div class="relative z-0 w-full mb-6 group">
    <label for="etablissement" class="block text-gray-700 font-bold mb-2">Établissement:</label>
    <input type="text" id="etablissement" v-model="formData.etablissement" required class="border rounded w-full py-2 px-3" disabled>
  </div>

  <div class="py-4 px-6 text-left">
    <div class="inline-flex" v-if="formData.id">
      <Editprofile :user="formData"/>
    </div>
  </div>
</form>
</template>
```

Modifier ici est un composant, que l'on passe des données, afin de les afficher en cliquant sur l'utilisateur.

```
<script>
import axios from 'axios';
import Editprofile from '@/components/Dashboard/etab/Admin/Editprofile.vue'

export default (await import('vue')).defineComponent({
  components:{Editprofile},
  data() {
    return {
      formData: {
        id : '',
        PPR: '',
        Nom: '',
        prenom: '',
        email: '',
        etablissement:'',
      },
    }
  },
  async mounted (){
    const user = (await axios.get('/adminprofile')).data
    console.log(user)
    this.formData.PPR = user.PPR
    this.formData.id = user.id_user
    this.formData.email = user.email
    this.formData.Nom = user.Nom
    this.formData.prenom = user.prenom
    this.formData.etablissement = user.etab_Nom
  }
})
</script>
```

La partie script consomme l'API (adminprofile), pour afficher les données.

En cliquant sur modifier, ce modal s'affiche, avec les données de l'utilisateur, il peut fermer ce modal en cliquant sur annuler, ou bien il peut modifier . A souligner que le mot de passe va pas s'afficher .

La partie template de Modal modifiée :

The screenshot shows a web application interface. On the left, there's a sidebar with navigation items: 'Gestion des directeurs', 'Gestion des professeurs', 'Gestion des interventions', 'Profil', and 'Déconnexion'. The main content area has a title 'Modifier votre profil'. Inside, there's a modal dialog with fields for 'Nom' (Name), 'Prénom' (First Name), 'Email', and 'PPR' (Employee ID). The 'Nom' field contains 'Abdo', 'Prénom' contains 'El harrachy', 'Email' contains 'abdotest2023@gmail.com', and 'PPR' contains '666669'. At the bottom of the modal are two buttons: 'Modifier' (in blue) and 'Annuler' (in red).

```

<template>
  <div>
    <button @click="showModal = true" class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded">
      Modifier
    </button>

    <div class="modal fixed w-full h-full top-0 left-0 flex items-center justify-center" v-if="showModal">
      <div class="modal-overlay absolute w-full h-full bg-gray-900 opacity-50"></div>

      <div class="modal-content bg-white rounded-lg p-6 w-11/12 w-6xl mx-auto">
        <span class="close absolute top-0 right-0 m-4 cursor-pointer" @click="closeModal">&times;</span>

        <div class="mb-4">
          <label for="nom" class="block text-gray-700 font-bold mb-2">Nom:</label>
          <input type="text" v-model="formData.Nom" required class="border rounded w-full py-2 px-3">
        </div>

        <div class="mb-4">
          <label for="prenom" class="block text-gray-700 font-bold mb-2">Prénom:</label>
          <input type="text" v-model="formData.prenom" required class="border rounded w-full py-2 px-3">
        </div>

        <div class="mb-4">
          <label for="email" class="block text-gray-700 font-bold mb-2">Email:</label>
          <input type="text" v-model="formData.email" required class="border rounded w-full py-2 px-3">
        </div>

        <div class="mb-4">
          <label for="password" class="block text-gray-700 font-bold mb-2">Mot de passe:</label>
          <input type="password" v-model="formData.password" class="border rounded w-full py-2 px-3">
        </div>

        <div class="flex justify-end mb-4">
          <button class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded" @click="submitForm">
            Modifier
          </button>
          <button @click="closeModal" class="bg-red-500 hover:bg-red-700 text-white font-bold py-2 px-4 rounded mr-2">
            Annuler
          </button>
        </div>
      </div>
    </div>
  </div>

```

L'évènement **submitForm** permet l'envoi des données après modification.

La partie script :

```
<script>
import axios from 'axios';

export default (await import('vue')).defineComponent({
  props: {
    user: {
      type: Object
    }
  },
  data() {
    return {
      showModal: false,
      formData: {
        PPR : '',
        id : '',
        Nom:'',
        prenom : '',
        email: '',
        password : ''
      }
    };
  },
  watch: {
    user: {
      handler(newUser) {
        console.log(newUser)
        this.formData.PPR = newUser.PPR
        this.formData.Nom = newUser.Nom;
        this.formData.id = newUser.id;
        this.formData.prenom = newUser.prenom;
        this.formData.email = newUser.email;
      },
      immediate: true
    }
  },
  methods: {
    closeModal() {
      this.showModal = false;
    },
    async submitForm() {
      const response = await axios.patch('/updateadm/'+this.formData.id,{
        Nom : this.formData.Nom,
        prenom : this.formData.prenom,
        email : this.formData.email,
        password : this.formData.password,
        PPR : this.formData.PPR
      })
      window.location.reload();
    }
  }
})
```

Il contient les données initialisées afin de les utiliser, puis un watch qui définit des observateurs, qui surveillent les changements propriétés et exécutent des actions en conséquence.

Ici user définit l'observateur pour la propriété "user". Lorsque la valeur de "user" change, la fonction "handler" sera appelée.

Immediate indique que la fonction "handler" doit être exécutée immédiatement lors de la création du composant, même si la valeur de "user" n'a pas encore changé.

Puis il y a l'API updateadm, qui est définie avec la méthode patch, qui permet la modification partielle.

3.5.2. Admin D'université :

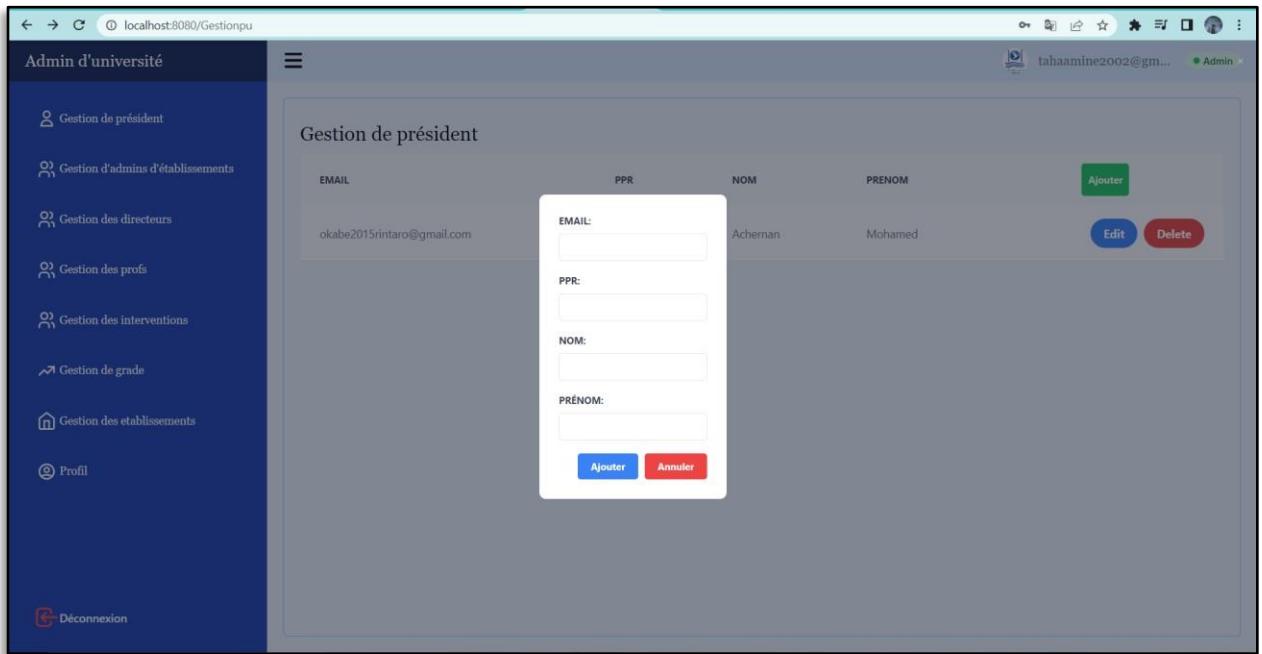
En plus de fonctionnalités semblables à celles de l'admin établissement, il a :

La gestion des présidents d'université : **Gestionpu**

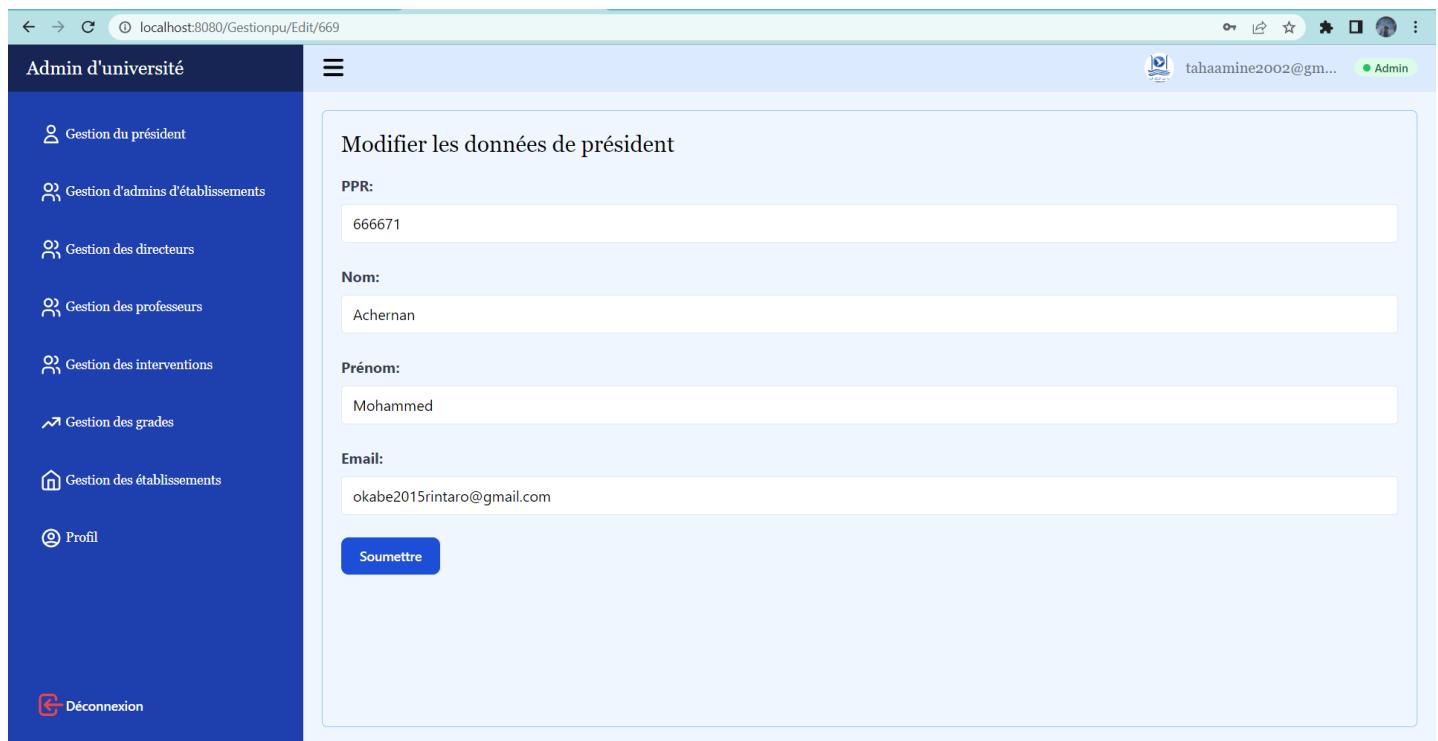
EMAIL	PPR	NOM	PRENOM	
okabe2015rintaro@gmail.com	666671	Achernan	Mohammed	Modifier Supprimer

PROJET XTRADH

Le modal ajouter :



Le bouton Modifier:



En cliquant sur soumettre les nouvelles données s'envoient à la base de données.

Le principe de fonctionnement est le même que celui cité précédemment, seuls les noms d'API sont modifiés.

PROJET XTRADH

Maintenant on passe à La gestion des admins d'établissement : **Gestionae**

EMAIL	PPR	NOM	PRENOM	ETABLISSEMENT	Ajouter
Zahiri.Hicham@uae.ac.ma	387801	Zahiri	Hicham	Faculté de Médecine Tanger	<button>Modifier</button> <button>Supprimer</button>
ZOUITA.ABDESALAM@uae.ac.ma	1051803	ZOUITA	ABDESALAM	Faculté des Sciences et Technique	<button>Modifier</button> <button>Supprimer</button>
abdoqftfpds@uae.ac.ma	52230	GUESSOUS	JENNATE	Faculté des Lettres et des Sciences Humaines	<button>Modifier</button> <button>Supprimer</button>
MASROUR.AICHA@uae.ac.ma	988977	MASROUR	AICHA	École Nationale de Commerce et de Gestion de Tanger	<button>Modifier</button> <button>Supprimer</button>
MHAMEDI.OMAR@uae.ac.ma	1124985	MHAMEDI	OMAR	École nationale des sciences appliquées de Tanger	<button>Modifier</button> <button>Supprimer</button>
abdotest2023@gmail.com	666669	Abdo	El Harrachy	Faculté des Lettres et des Sciences Humaines	<button>Modifier</button> <button>Supprimer</button>

La même chose ici Ajouter est un composant, et modifier nous redirige vers la vue où on peut éditer :

EMAIL	PPR	NOM	PRENOM	ETABLISSEMENT	Ajouter
Zahiri.Hicham@uae.ac.ma	387801	Zahiri	Hicham	Faculté de Médecine Tanger	<button>Modifier</button> <button>Supprimer</button>
ZOUITA.ABDESALAM@uae.ac.ma	1051803	ZOUITA	ABDESALAM	Faculté des Sciences et Technique	<button>Modifier</button> <button>Supprimer</button>
abdoqftfpds@uae.ac.ma	52230	GUESSOUS	JENNATE	Faculté des Lettres et des Sciences Humaines	<button>Modifier</button> <button>Supprimer</button>
MASROUR.AICHA@uae.ac.ma	988977	MASROUR	AICHA	École Nationale de Commerce et de Gestion de Tanger	<button>Modifier</button> <button>Supprimer</button>
MHAMEDI.OMAR@uae.ac.ma	1124985	MHAMEDI	OMAR	École nationale des sciences appliquées de Tanger	<button>Modifier</button> <button>Supprimer</button>
abdotest2023@gmail.com	666669	Abdo	El Harrachy	Faculté des Lettres et des Sciences Humaines	<button>Modifier</button> <button>Supprimer</button>

PROJET XTRADH

modifier nous renvoie vers :

A screenshot of a web application interface. On the left, a dark blue sidebar menu titled "Admin d'université" lists various administrative functions: Gestion du président, Gestion d'admins d'établissements, Gestion des directeurs, Gestion des professeurs, Gestion des interventions, Gestion des grades, Gestion des établissements, Profil, and Déconnexion. The main content area has a light blue header "Modifier les données de l'admin". It contains several input fields: "PPR:" with value "387801", "Nom:" with value "Zahiri", "Prénom:" with value "Hicham", "Email:" with value "Zahiri.Hicham@uae.ac.ma", and "Etablissement:" with a dropdown menu showing "Faculté de Médecine Tanger". A blue "Soumettre" button is at the bottom.

On passe maintenant à la vue gestion des grades : **Gestiongrade**

A screenshot of the "Gestiongrade" view. The left sidebar is identical to the previous screenshot. The main content area shows a table titled "Liste des grades" with three rows of data. The columns are labeled "DESIGNANTION", "CHARGE_STATUTAIRE", and "TAUX_HORAIRE_VACATION". Each row includes "Ajouter grade" (blue button), "Modifier" (blue button), and "Supprimer" (red button) actions. The data is as follows:

DESIGNANTION	CHARGE_STATUTAIRE	TAUX_HORAIRE_VACATION	
Professeur d'Enseignement Supérieur	190	500	Modifier Supprimer
Professeur Habilité	200	400	Modifier Supprimer
Professeur Agrégé	240	300	Modifier Supprimer

PROJET XTRADH

Voici le code template :

```
<template>
<div>
  <h3 class="text-2xl font-serif text-left py-2">Listes de grade</h3>

  <div class="w-full overflow-x-auto overflow-y-auto h-[calc(100vh-200px)] scrollbar [& scrollbar-track-gray-100">
    <table class="w-full text-sm text-left [& text-gray-500 [& dark:text-gray-400">
      <thead class="text-xs [& text-gray-700 uppercase [& bg-gray-50 ">
        <tr>
          <th scope="col" class="py-3 px-6">
            Designation
          </th>

          <th scope="col" class="py-3 px-6">
            Charge_statutaire
          </th>

          <th scope="col" class="py-3 px-6">
            Taux_horaire_Vocation
          </th>

          <th scope="col" class="py-3 pl-16">
            <div class="flex justify-center" >
              <creer/>
            </div>
          </th>
        </tr>
      </thead>
      <tbody>
        <tr v-for="data in Obj" :key="data.id" class=" [& bg-white border-b">
          <td class="py-4 px-6" >
            {{ data.designation }}
          </td>
          <td class="py-4 px-6" >
            {{ data.charge_statutaire}}
          </td>
          <td class="py-4 px-6" >
            {{ data.Taux_horaire_Vocation}}
          </td>
          <td class="py-4 px-6 text-right">
            <div class="inline-flex">
              <router-link :to="/Gestiongrade/Edit/${data.id}" class=" [& bg-blue-500 [& hover:bg-blue-400 [& text-white font-bold py-2 px-4 rounded-full mr-2">
                | Edit
              </router-link>
              <button class=" [& bg-red-500 hover:bg-blue-400 [& text-white font-bold py-2 px-4 rounded-full" @click="">
                | Delete
              </button>
            </div>
          </td>
        </tr>
      </tbody>
    </table>
  </div>
```

La partie script contient :

```
<script>
import creer from '@/components/Dashboard/grade/creer.vue'
//import Create from '@/components/Dashboard/Prof/Create.vue'
import axios from 'axios';
export default (await import('vue')).defineComponent({
  components: { creer },
  data(){
    return {
      Obj:[],
    }
  },
  async mounted() {
    try {
      const response=await axios.get('/grade')
      console.log(response.data)
      this.Obj=response.data
      this.showModal = false;
    } catch (error) {
      //console.error(error);
    }},
  methods:{
    togglemodal(){
      this.showmodal=!this.showmodal
    },
    async deletegrade(data){
      const response = await axios.delete('/grade/'+data.id_Grade)
      let index = this.Obj.indexOf(data);
      this.Obj.splice(index,1);
    }
  })
}

</script>

<style>
.scrollable {
  max-width: 100%;
  overflow-x: scroll;
}
</style>
```

PROJET XTRADH

La consommation de l'API grade par la méthode get.

```
<template>
  <div>
    <button @click="showModal = true" class="bg-green-500 hover:bg-blue-700 text-white font-bold py-3 px-2 rounded">
      Ajouter grade
    </button>

    <div class="modal fixed w-full h-full top-0 left-0 flex items-center justify-center" v-if="showModal">
      <div class="modal-overlay absolute w-full h-full bg-gray-900 opacity-50"></div>

      <div class="modal-content bg-white rounded-lg p-6 max-w-xl">
        <span class="close absolute top-0 right-0 m-4 cursor-pointer" @click="closeModal">&times;</span>

        <div class="mb-4">
          <label for="Designantion" class="block text-gray-700 font-bold mb-2">Designantion:</label>
          <input type="text" id="Designantion" v-model="formData.designation" required class="border rounded w-full py-2 px-3">
        </div>

        <div class="mb-4">
          <label for="Charge_statutaire" class="block text-gray-700 font-bold mb-2">Charge_statutaire:</label>
          <input type="number" id="Charge_statutaire" v-model="formData.charge_statutaire" required class="border rounded w-full py-2 px-3">
        </div>

        <div class="mb-4">
          <label for="Taux_horaire_Vocation" class="block text-gray-700 font-bold mb-2">Taux_horaire_Vocation:</label>
          <input type="number" id="Taux_horaire_Vocation" v-model="formData.Taux_horaire" required class="border rounded w-full py-2 px-3">
        </div>

        <div class="flex justify-end">
          <button type="submit" class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded" @click="submitForm">
            Ajouter
          </button>
          <button class="bg-red-500 hover:bg-red-700 text-white font-bold py-2 px-4 rounded ml-2" @click="closeModal">
            Annuler
          </button>
        </div>
      </div>
    </div>
  </template>
```

Le composant ajouter contient :

La partie template qui contient un modal.

La partie script contient :

```
2  <script>
3  import axios from 'axios';
4
5  export default (await import('vue')).defineComponent({
6    data() {
7      return {
8        showModal: false,
9        formData: {
10          designation : '',
11          charge_statutaire : '',
12          Taux_horaire : ''
13        }
14      };
15    },
16    methods: {
17      closeModal() {
18        this.showModal = false;
19      },
20      async submitForm() {
21        const response = await axios.post('/grade',{
22          designation : this.formData.designation,
23          charge_statutaire : this.formData.charge_statutaire,
24          Taux_horaire_Vocation : this.formData.Taux_horaire
25        })
26        window.location.reload();
27        this.closeModal();
28      }
29    }
30  });
31  </script>
32
33  <style>
34  .modal {
35    z-index: 9999;
36  }
37
38  .modal-overlay {
39    z-index: -1;
40  }
41
42  .close {
43    color: #aaa;
44    cursor: pointer;
45  }
46
47  .close:hover {
48    color: #000;
49  }
50  </style>
```

PROJET XTRADH

Il y'a consommation de L'api grade par la méthode post.

Pour modifier on a :

```
<template>
  <form @submit.prevent="submitForm" >

    <div class="relative z-0 w-full mb-6 group">
      <label for="Designation" class="block text-gray-700 font-bold mb-2">Designation:</label>
      <input type="text" id="Designation" v-model="formData.designation" required class="border rounded w-full py-2 px-3">
    </div>

    <div class="relative z-0 w-full mb-6 group">
      <label for="Charge_statutaire" class="block text-gray-700 font-bold mb-2">Charge_statutaire:</label>
      <input type="number" id="Charge_statutaire" v-model="formData.charge_statutaire" required class="border rounded w-full py-2 px-3">
    </div>

    <div class="relative z-0 w-full mb-6 group">
      <label for="Prenom" class="block text-gray-700 font-bold mb-2">Taux Horaire Vocation :</label>
      <input type="number" id="Prenom" v-model="formData.Taux_horaire" required class="border rounded w-full py-2 px-3">
    </div>

    <button type="submit" class="text-white bg-blue-700 hover:bg-blue-800 focus:ring-4 focus:outline-none focus:ring-blue-300 font-medium rounded-lg text-sm py-2 px-4 uppercase tracking-widest">Envoyer</button>
  </form>
</template>

<script>
```

La partie script contient :

```
</template>

<script>
import axios from 'axios';
import { createLogger } from 'vuex';

export default (await import('vue')).defineComponent({
  data() {
    return {
      formData: {
        id: '',
        designation: '',
        charge_statutaire: '',
        Taux_horaire: ''
      },
      prev_email: ''
    };
  }
});
```

L'initialisation des données qu'on va utiliser.

PROJET XTRADH

```
methods: {
  async submitForm() {
    try {
      const response = await axios.patch('/grade/'+ this.formData.id, {
        designation : this.formData.designation,
        charge_statutaire : this.formData.charge_statutaire,
        Taux_horaire_Vocation : this.formData.Taux_horaire
      });
      console.log(response);

      this.$router.push('/Gestiongrade')
    } catch (error) {
      console.error(error);
      // Handle error
    }
  },
},
```

Ici on consomme l'API grade par la méthode patch, après la modification on redirige l'utilisateur vers la vue gestiongrade.

```
},
async mounted(){
  let id = this.$route.params.id
  this.formData.id=id
  const response = await axios.get('/grade/'+id)
  console.log(response)
  this.formData.designation = response.data.designation
  this.formData.charge_statutaire = response.data.charge_statutaire
  this.formData.Taux_horaire = response.data.Taux_horaire_Vocation
},
cancelForm() {
},
}

</script>
```

Cette partie de code est responsable de l'affichage des données récupérées en cliquant sur modifier, puisque les données sont montées une fois un enregistrement est choisi.

PROJET XTRADH

On passe maintenant à la vue gestion des établissements :

CODE	NOM	TELEPHONE	FAX	VILLE	NBR_ENSEIGNANTS	
ABD001	Faculté des Sciences et Technique	0667893458	9876543210	AL HOCIEMA	50	<button>Modifier</button> <button>Supprimer</button>
ABD002	Faculté des Lettres et des Sciences Humaines	0663423458	9876543210	Tetouan	40	<button>Modifier</button> <button>Supprimer</button>
ABD003	École Nationale de Commerce et de Gestion de Tanger	0662348456	9876543210	Tanger	100	<button>Modifier</button> <button>Supprimer</button>
ABD004	École nationale des sciences appliquées de Tanger	0668123456	9876543210	Tanger	70	<button>Modifier</button> <button>Supprimer</button>
ABD005	Faculté de Medecine Tanger	0663456903	9876543210	Tanger	60	<button>Modifier</button> <button>Supprimer</button>

En cliquant sur ajouter :

CODE	NOM	VILLE	NBR_ENSEIGNANTS	
ABD001	Faculté des Sciences et Tech	AL HOCIEMA	50	<button>Modifier</button> <button>Supprimer</button>
ABD002	Faculté des Lettres et des Sciences Humaines	Tetouan	40	<button>Modifier</button> <button>Supprimer</button>
ABD003	École Nationale de Commerce et de Gestion de Tanger	Tanger	100	<button>Modifier</button> <button>Supprimer</button>
ABD004	École nationale des sciences appliquées de Tanger	Tanger	70	<button>Modifier</button> <button>Supprimer</button>
ABD005	Faculté de Medecine Tanger	Tanger	60	<button>Modifier</button> <button>Supprimer</button>

PROJET XTRADH

En cliquant sur modifier :

Admin d'université

Gestion du président

Gestion d'admins d'établissements

Gestion des directeurs

Gestion des professeurs

Gestion des interventions

Gestion des grades

Gestion des établissements

Profil

Déconnexion

Modifier l'établissement

code:

ABD001

Nom:

Faculté des Sciences et Technique

Téléphone:

0667893458

Fax:

9876543210

ville:

AL HOCIEMA

Nbr_enseignants:

50

Soumettre

Pour le fonctionnement, la même logique se répète, uniquement dans ce cas l'API s'appelle établissement

```
<script>
import creer from '@/components/Dashboard/etablissement/creer.vue'
import axios from 'axios'
export default (await import('vue')).defineComponent({
  components: {creer},
  data(){
    return {
      Obj:[],
    }
  },
  async mounted() {
    try {
      const response=await axios.get('/etablissement')
      console.log(response.data)
      this.Obj=response.data
      this.showModal = false;
    } catch (error) {
      console.error(error);
    },
  },
  methods:{
    togglemodal(){
      this.showmodal=!this.showmodal
    },
    async deleteetb(data){
      console.log(data)
      const response = await axios.delete('/etablissement/'+data.id);
      let index = this.Obj.indexOf(data)
      this.Obj.splice(index,1)
    }
  }
})
```

PROJET XTRADH

De même pour profil. Avec la consommation de l'API **adminprofile**.

The screenshot shows a web application interface for managing user profiles. On the left, there is a sidebar with a dark blue background containing various navigation items:

- Gestion du président
- Gestion d'admins d'établissements
- Gestion des directeurs
- Gestion des professeurs
- Gestion des interventions
- Gestion des grades
- Gestion des établissements
- Profil

At the bottom of the sidebar is a "Déconnexion" (Logout) button.

The main content area has a light blue background and contains the following form fields:

PPR: 666668

Nom: Amine

Prénom: Taha

Email: tahaamine2002@gmail.com

Établissement: Faculté des Sciences et Technique

A blue "Modifier" (Update) button is located at the bottom of the form.

3.5.3. Directeur d'établissement

La vue gestion des interventions :

INTITULE_INTERVENTION	ANNEE_UNIV	SEMESTRE	NBR_HEURES	DATE_DEBUT	DATE_FIN	ETAT	OPERATION
Séminaire sur la gestion des ressources humaines	2022/2023	Semestre 2	15	2023-03-05	2023-03-07	invalidé	<button>valider</button>
Conférence sur le Metaverse	2022/2023	Semestre 2	34	2023-01-29	2023-05-27	invalidé	<button>valider</button>
Conférence sur le Metaverse 2	2022/2023	Semestre 1	34	2022-11-29	2023-01-27	invalidé	<button>valider</button>

Le directeur ici peut valider ou invalider une intervention d'un professeur, qui travaille dans son même établissement. La partie template :

```
<template>
  <div>
    <h3 class="text-2xl font-serif text-left py-2">Gestion des Interventions</h3>

    <div class="w-full overflow-x-auto overflow-y-auto h-[calc(100vh-200px)] scrollbar [&] scrollbar-track-gray-100">
      <table class="w-full text-sm text-left [&] text-gray-500">
        <thead class="text-xs [&] text-gray-700 uppercase [&] bg-gray-50">
          <tr>
            <th scope="col" class="py-3 px-6">
              | Intitule_intervention
            </th>
            <th scope="col" class="py-3 px-6">
              | <div class="flex items-center">
                |   Annee_univ
              </div>
            </th>
            <th scope="col" class="py-3 px-6">
              | <div class="flex items-center">
                |   Semestre
              </div>
            </th>
            <th scope="col" class="py-3 px-6">
              | &nbsp; Nbr_heures
            </th>
            <th scope="col" class="py-3 px-6">
              | <div class="flex items-center">
                |   Date_debut
              </div>
            </th>
            <th scope="col" class="py-3 px-6">
              | <div class="flex items-center">
                |   Date_fin
              </div>
            </th>
            <th scope="col" class="py-3 px-6">
              | <div class="flex items-center">
                |   Etat
              </div>
            </th>
            <th scope="col" class="py-3 px-6">
              | Operation
            </th>
          </tr>
        </thead>
        <tbody>
          <tr>
            <td>Séminaire sur la gestion des ressources humaines</td>
            <td>2022/2023</td>
            <td>Semestre 2</td>
            <td>15</td>
            <td>2023-03-05</td>
            <td>2023-03-07</td>
            <td>invalidé</td>
            <td><button>valider</button></td>
          </tr>
          <tr>
            <td>Conférence sur le Metaverse</td>
            <td>2022/2023</td>
            <td>Semestre 2</td>
            <td>34</td>
            <td>2023-01-29</td>
            <td>2023-05-27</td>
            <td>invalidé</td>
            <td><button>valider</button></td>
          </tr>
          <tr>
            <td>Conférence sur le Metaverse 2</td>
            <td>2022/2023</td>
            <td>Semestre 1</td>
            <td>34</td>
            <td>2022-11-29</td>
            <td>2023-01-27</td>
            <td>invalidé</td>
            <td><button>valider</button></td>
          </tr>
        </tbody>
      </table>
    </div>
  </div>

```

PROJET XTRADH

Ici les données consommées via l'api s'affichent :

```
<tbody>
  <tr class="bg-white border-b" v-for="data in interv" :key="data">
    <th class="py-4 px-6 font-medium text-gray-900 whitespace nowrap">
      {{ data.Intitule_Intervention }}
    </th>
    <td class="py-4 px-6">
      {{ data.Annee_univ }}
    </td>
    <td class="py-4 px-6">
      {{ data.Semestre }}
    </td>
    <td class="py-4 px-6">
      {{ data.Nbr_heures }}
    </td>
    <td class="py-4 px-6">
      {{ data.Date_debut }}
    </td>
    <td class="py-4 px-6">
      {{ data.Date_fin }}
    </td>
    <td class="py-4 px-6">
      <span v-if="!data.visa_etb" class="bg-red-100 text-red-800 text-xs font-medium mr-2 px-2.5 py-0.5 rounded">invalidé</span>
      <span v-if="data.visa_etb" class="bg-green-100 text-green-800 text-xs font-medium mr-2 px-2.5 py-0.5 rounded">valide</span>
    </td>
    <td class="py-4 px-6">
      <div class="inline-flex">
        <button v-if="!data.visa_etb" class="bg-blue-500 hover:bg-blue-400 text-white font-bold py-2 px-4 rounded-full" @click="validerinterv(data)">
          valider
        </button>
        <button v-if="data.visa_etb" class="bg-red-500 hover:bg-blue-400 text-white font-bold py-2 px-4 rounded-full" @click="invaliderinterv(data)">
          invalider
        </button>
      </div>
    </td>
  </tr>
</tbody>
</table>
</div>
</div>
```

L'API utilisé c'est directueretabintervall :

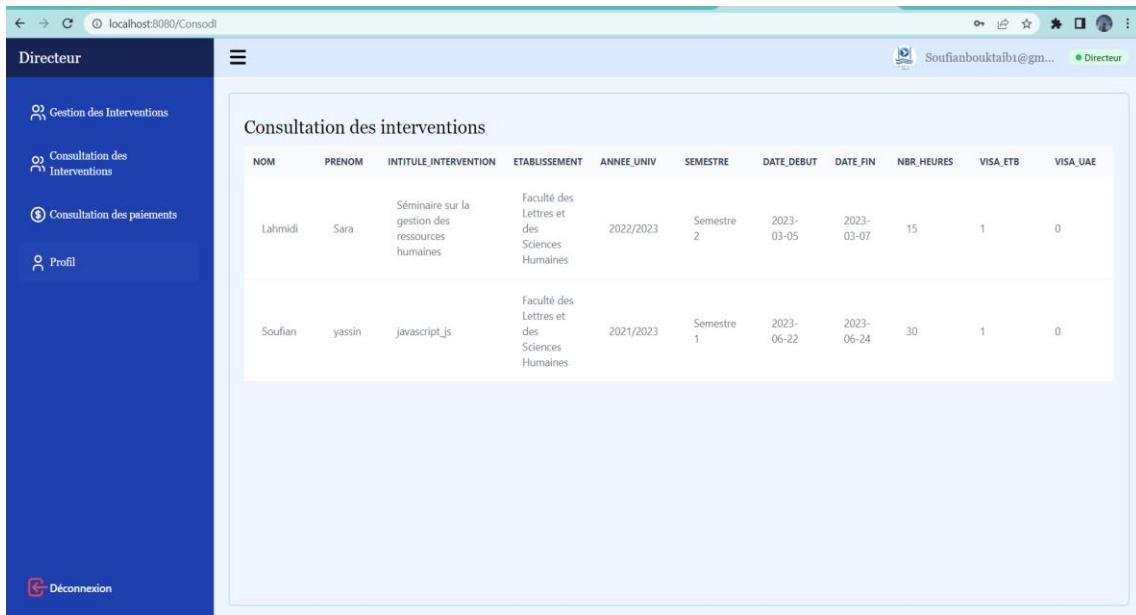
```
91
92  <script>
93  import axios from 'axios';
94  export default (await import('vue')).defineComponent({
95    components: {},
96    data(){
97      return {
98        |   interv : []
99
100
101
102      }
103    },
104    async mounted()
105    {
106      const response = await axios.get('/directueretabintervall')
107      this.interv = response.data
108    },
109    methods:[],
110
111    async validerinterv(e)
112    {
113      const response = await axios.get('/valideretb/'+e.id_intervention)
114      let index = this.interv.indexOf(e)
115      this.interv[index].visa_etb = 1 ;
116    },
117    async invaliderinterv(e)
118    {
119      const response = await axios.get('/invalideretb/'+e.id_intervention)
120      let index = this.interv.indexOf(e)
121      this.interv[index].visa_etb = 0 ;
122    },
123
124  }
```

PROJET XTRADH

Les deux méthodes utilisées pour valider et invalider l'intervention sont validerinterv et invaliderinterv , les deux API sont respectivement : valideretb et invalideretb , on passe le paramètre id_intervention pour spécifier cette action sur chaque enregistrement.

Maintenant on passe à la vue, Consultation des interventions :

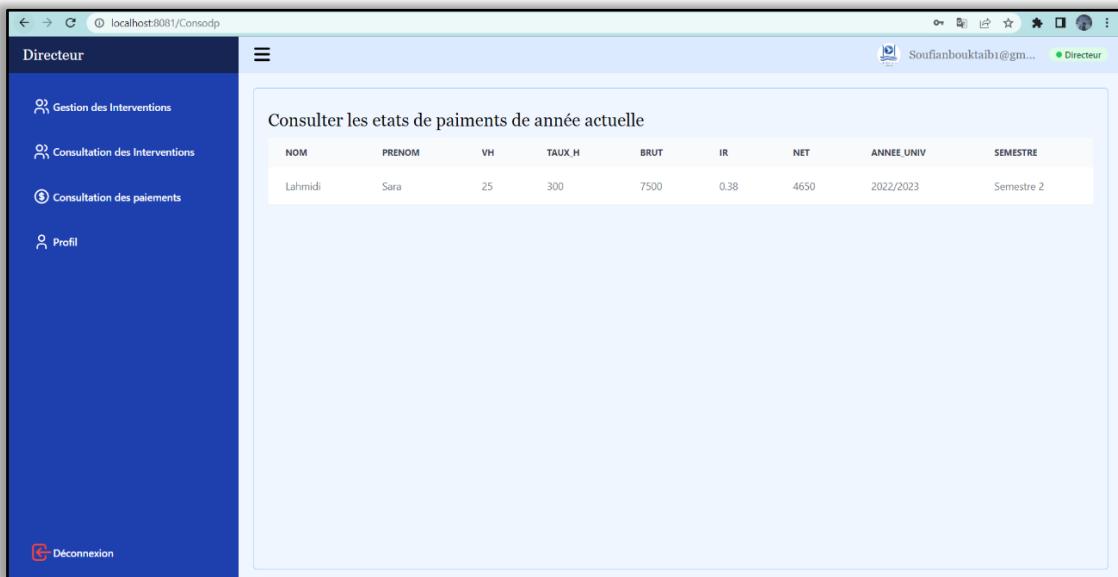
Où il y a que les interventions validées :



The screenshot shows a web application interface for a director. On the left, a sidebar menu lists 'Gestion des Interventions', 'Consultation des Interventions', 'Consultation des paiements', and 'Profil'. At the bottom is a 'Déconnexion' button. The main content area is titled 'Consultation des interventions' and displays a table of interventions. The table has columns: NOM, PRENOM, INTITULE_INTERVENTION, ETABLISSEMENT, ANNEE_UNIV, SEMESTRE, DATE_DEBUT, DATE_FIN, NBR_HEURES, VISA_ETB, and VISA_UAE. Two rows of data are shown:

NOM	PRENOM	INTITULE_INTERVENTION	ETABLISSEMENT	ANNEE_UNIV	SEMESTRE	DATE_DEBUT	DATE_FIN	NBR_HEURES	VISA_ETB	VISA_UAE
Lahmidi	Sara	Séminaire sur la gestion des ressources humaines	Faculté des Lettres et des Sciences Humaines	2022/2023	Semestre 2	2023-03-05	2023-03-07	15	1	0
Soufian	yassin	javascript_js	Faculté des Lettres et des Sciences Humaines	2021/2023	Semestre 1	2023-06-22	2023-06-24	30	1	0

Il y a la vue consultation des états de paiement :



The screenshot shows a web application interface for a director. The sidebar menu is identical to the previous screenshot. The main content area is titled 'Consulter les etats de paiements de l'année actuelle' and displays a table of payment records. The table has columns: NOM, PRENOM, VH, TAUX_H, BRUT, IR, NET, ANNEE_UNIV, and SEMESTRE. One row of data is shown:

NOM	PRENOM	VH	TAUX_H	BRUT	IR	NET	ANNEE_UNIV	SEMESTRE
Lahmidi	Sara	25	300	7500	0.38	4650	2022/2023	Semestre 2

PROJET XTRADH

Ces deux vues sont simples, elles utilisent l'API avec la méthode get :

```
4 import axios from 'axios';
5 export default (await import('vue')).defineComponent({
6   components: {},
7   data(){
8     return {
9       | | interv : []
10      }
11    },
12    async mounted(){
13      const response = await axios.get('/directeurtabintervvalid');
14      this.interv=response.data
15      console.log(response.data)
16    }
17  },);
18
19 </script>
20
```

```
<script>
import axios from 'axios';
export default (await import('vue')).defineComponent({
  data() {
    return {
      ints:''
    };
  },
  methods:{},
},async mounted(){
  const response = await axios.get('/consultpaiementetabdirecteur');
  this.ints=response.data
  console.log(response.data)
}
});</script>
```

Et finalement on retrouve la vue profile :

Et son principe est le même que celui cité précédemment.

PROJET XTRADH

The screenshot shows a web application interface for a 'Directeur' user. The left sidebar has a dark blue background with white icons and text:

- Gestion des Interventions
- Consultation des Interventions
- Consultation des paiements
- Profil

At the bottom of the sidebar is a 'Déconnexion' button.

The main content area has a light blue header 'Modifier votre profil'. It contains several input fields with placeholder text:

- Nom: Bouktaib
- Prénom: Soufiane
- Email: Soufianbouktaib1@gmail.com
- PPR: 666670
- Établissement: Faculté des Lettres et des Sciences Humaines

A blue 'Modifier' button is located at the bottom left of the form.

3.5.4. Président D'Université

La vue consultation des intervention

The screenshot shows a web application interface for a 'Président' user. The left sidebar has a dark blue background with white icons and text:

- + Consultations des Interventions
- Gestion des Interventions
- Consultation des paiements
- Profil

At the bottom of the sidebar is a 'Déconnexion' button.

The main content area has a light blue header 'Consultation des Interventions'. It includes a filter section 'Filtrer par Etablissement' with a dropdown menu showing 'Tous les établissements'.

Below the filter is a table titled 'INTITULE_INTERVENTION' with the following columns: NOM, PRENOM, ETABLISSEMENT, ANNEE_UNIV, SEMESTRE, DATE_DEBUT, DATE_FIN, NBR_HEURES, VISA_UAE, and VISA_ETB.

INTITULE_INTERVENTION	NOM	PRENOM	ETABLISSEMENT	ANNEE_UNIV	SEMESTRE	DATE_DEBUT	DATE_FIN	NBR_HEURES	VISA_UAE	VISA_ETB
Conférence sur l'économie mondiale	Longo	Adil	Faculté des Sciences et Technique	2022/2023	Semestre 1	2022-12-01	2022-12-02	8	1	1
Formation sur les nouvelles technologies	Longo	Adil	Faculté des Sciences et Technique	2022/2023	Semestre 1	2022-10-10	2022-10-15	30	1	1
Cours sur la littérature française	Longo	Adil	Faculté des Sciences et Technique	2022/2023	Semestre 2	2023-04-10	2023-05-20	60	1	1
Séminaire de Digitalisation Web	Abdo	Elharrachyprof	Faculté de Médecine Tanger	2023/2024	Semestre 2	2023-12-05	2024-01-31	34	1	1
Atelier sur le service de messagerie postfix	Abdo	Elharrachyprof	Faculté de Médecine Tanger	2022/2023	Semestre 1	2022-12-05	2023-01-31	34	1	1

Le président d'université peut filtrer par établissement.

L'API consommé est :

```
<script>
import axios from 'axios';
export default (await import('vue')).defineComponent({
  components: {},
  data(){
    return {
      interv : [],
      etabs : [],
      filter : ''
    }
  },
  async mounted(){
    const response = await axios.get('/interventionuaevalid');
    const etbs = await axios.get('/établissement')
    this.etabs = etbs.data;
    this.interv=response.data
    console.log(response.data)
  },
  computed: {
    filteredData() {
      if (this.filter.length > 0 ) {
        return this.interv.filter((intr)=> intr.Nom_etb.toLowerCase().includes(this.filter.toLocaleLowerCase()) )
        //return this.cons; // Return all data if no filter is applied
      } else {
        return this.interv
      }
    },
  },
});
</script>
```

PROJET XTRADH

Passons maintenant à la vue Gestion des interventions :

The screenshot shows a web application interface for managing interventions. On the left, a sidebar menu for 'Président' includes options like 'Consultations des Interventions', 'Gestion des Interventions', 'Consultation des paiements', 'Profil', and 'Déconnexion'. The main content area is titled 'Gestion des Interventions' and displays a table of interventions. The columns are labeled: INTITULE_INTERVENTION, NOM, PRENOM, ETABLISSEMENT, ANNEE_UNIV, SEMESTRE, NBR_HEURES, DATE_DEBUT, DATE_FIN, ETAT, and OPERATION. There are six rows of data:

INTITULE_INTERVENTION	NOM	PRENOM	ETABLISSEMENT	ANNEE_UNIV	SEMESTRE	NBR_HEURES	DATE_DEBUT	DATE_FIN	ETAT	OPERATION
Conférence sur l'économie mondiale	Longo	Adil	Faculté des Sciences et Technique	2022/2023	Semestre 1	8	2022-12-01	2022-12-02	valide	<button>invalider</button>
Formation sur les nouvelles technologies	Longo	Adil	Faculté des Sciences et Technique	2022/2023	Semestre 1	30	2022-10-10	2022-10-15	valide	<button>invalider</button>
Cours sur la littérature française	Longo	Adil	Faculté des Sciences et Technique	2022/2023	Semestre 2	60	2023-04-10	2023-05-20	valide	<button>invalider</button>
Séminaire de Digitalisation Web	Abdo	Elharrachyprof	Faculté de Médecine Tanger	2023/2024	Semestre2	34	2023-12-05	2024-01-31	valide	<button>invalider</button>
Atelier sur le service de messagerie posftix	Abdo	Elharrachyprof	Faculté de Médecine Tanger	2022/2023	Semestre1	34	2022-12-05	2023-01-31	valide	<button>invalider</button>
Conférence sur le Metaverse	Amine	Taha	Faculté des Lettres et des Sciences Humaines	2022/2023	Semestre2	34	2023-01-29	2023-05-27	valide	<button>invalider</button>

Ici le président d'université, peut valider ou invalider, les interventions validées par directeur d'établissement.

L'API consommé est :

```

<script>
import axios from 'axios';
export default (await import('vue')).defineComponent({
  components: {},
  data(){
    return {
      interv : []
    }
  },
  async mounted()
  {
    const response = await axios.get('/intervention')
    this.interv = response.data
  },
  methods:{

    async validerinterv(e)
    {
      const response = await axios.get('/valideruae/'+e.id_intervention)
      console.log(response)
      let index = this.interv.indexOf(e)
      this.interv[index].visa_uee = 1 ;
    },
    async invaliderinterv(e)
    {
      const response = await axios.get('/invalideruae/'+e.id_intervention)
      let index = this.interv.indexOf(e)
      this.interv[index].visa_uee = 0 ;
    },
  }
})

```

PROJET XTRADH

Puis on a la vue consultation des paiements :

The screenshot shows a web application interface for 'Président'. On the left, there's a sidebar with navigation links: '+ Consultations des Interventions', 'Gestion des Interventions', 'Consultation des paiements' (which is currently selected), and 'Profil'. At the bottom of the sidebar is a 'Déconnexion' button. The main content area has a title 'Consulter les etats de paiements d'année actuelle'. Below it is a table with the following data:

NOM	PRENOM	ETABLISSEMENT	VH	TAUX_H	BRUT	IR	NET	ANNEE_UNIV	SEMESTRE
Hassani	Ahmed	Faculté des Sciences et Technique	50	400	20000	0.38	12400	2022/2023	Semestre 1
Lahmidi	Sara	Faculté des Lettres et des Sciences Humaines	25	300	7500	0.38	4650	2022/2023	Semestre 2
Mansouri	Amina	École Nationale de Commerce et de Gestion de Tanger	35	300	10500	0.38	6510	2022/2023	Semestre 1
Longo	Adil	Faculté de Médecine Tanger	150	500	75000	0.38	46500	2022/2023	Semestre 1
Longo	Adil	Faculté des Sciences et Technique	200	500	100000	0.38	62000	2022/2023	Semestre 2
Abdo	Elharrachyprof	École Nationale de Commerce et de Gestion de Tanger	34	650	22100	0.38	13702	2022/2023	Semestre 2
Longo	Adil	Faculté de Médecine Tanger	50	400	20000	0.38	12400	2022/2023	Semestre 1

L'api consommée est :

```
8   </template>
9
0   <script>
1   import axios from 'axios';
2   export default (await import('vue')).defineComponent({
3   data() {
4   |   return {
5   |   |   ints:''
6   |   };
7   },
8   methods:{},
9   },
0   async mounted(){
1   const response = await axios.get('/paiement');
2   this.ints=response.data
3   console.log(response.data)
4   }
5   });
6   </script>
7
```

PROJET XTRADH

Et finalement, La vue profile :

The screenshot shows a web application interface for managing user profiles. On the left, there is a sidebar with a blue background containing several menu items:

- Gestion du président
- Gestion d'admins d'établissements
- Gestion des directeurs
- Gestion des professeurs
- Gestion des interventions
- Gestion des grades
- Gestion des établissements
- Profil
- Déconnexion

The main content area has a light blue header with the text "Modifier votre profil". Below the header, there is a form with the following fields and their current values:

PPR:	666668
Nom:	Amine
Prénom:	Taha
Email:	tahaamine2002@gmail.com
Établissement:	Faculté des Sciences et Technique

At the bottom of the form is a blue button labeled "Modifier". In the top right corner of the main content area, there is a user profile icon, the email address "tahaamine2002@gm...", and a green circular status indicator with the text "Admin".

3.5.5. Professeur :

La vue Paiement :

The screenshot shows a web application interface for a professor. On the left, a sidebar menu includes 'Paiement' (selected), 'Interventions', 'Profil', and 'Déconnexion'. The main content area has a header 'Consulter votre procédure de paiement d'année en cours' and a button 'Anciennes fiches de paie'. Below is a table showing payment details:

ÉTABLISSEMENT	VOLUME HORRAIRE	TAUX_HORRAIRE	SALAIRE BRUT	IMPORT SUR REVENU	SALAIRE NET	ANNEE_UNIVIRSTITAIRE	SEMESTRE
École Nationale de Commerce et de Gestion de Tanger	34	650	22100	0.38	13702	2022/2023	Semestre2
École nationale des sciences appliquées de Tanger	34	650	22100	0.38	13702	2022/2023	Semestre2
Faculté de Médecine Tanger	34	200	6800	0.38	4216	2022/2023	Semestre1

Ici le professeur peut consulter l'état de paiement de l'état actuel, et peut voir l'historique de paiement via la route Anciennes fiches de paie , qui affiche :

The screenshot shows a web application interface for a professor. On the left, a sidebar menu includes 'Paiement' (selected), 'Interventions', 'Profil', and 'Déconnexion'. The main content area has a header 'Télécharger votre fiches de paie' and a table showing payment history:

ANNÉE UNIVERSITAIRE	ETAT	FICHE
2022/2023	Payée	Télécharger
2022/2023	Payée	Télécharger
2022/2023	Payée	Télécharger

Les APIS consommées sont :

En paiementp :

```

57   |         paiements: [
58   |         ]
59   |     };
60   | },
61   |     async mounted()
62   | {
63   |     const response = await axios.get('/paiementprof')
64   |     this.paiements = response.data
65   |     console.log(response.data)
66   |
67   | },
68   | });
69   |);

```

En ancienne fiche de paie

```

3  </div>
4  </template>
5  <script>
6  import axios from 'axios'
7  export default {await import('vue')).defineComponent({
8      async mounted()
9      {
10         const response = await axios.get('/historiquepdfpaie')
11         | this.Data = response.data ;
12     },
13     data() {
14         return {
15             | Data : []
16         };

```

```

<td class="px-6 py-4">
  <button class="!bg-gray-300 !hover:bg-gray-400 !text-gray-800 font-bold py-2 px-4 rounded inline-flex items-center">
    <svg class="fill-current w-4 h-4 mr-2" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 20 20"><path d="M13 8V2H7v6H218 8 8h-5zM0 18h20v2H0v-2z"/></svg>
    <a :href="item.url">Télécharger</a>
  </button>
</td>

```

Ces liens sont les liens de téléchargements qui s'affiche

Passons maintenant à la vue intervention :

The screenshot shows a web application interface for a professor. On the left, there is a sidebar with a dark blue background containing navigation links: 'Paiement' (with a credit card icon), 'Interventions' (with a clipboard icon), 'Profil' (with a person icon), and 'Déconnexion' (with a lock icon). The main content area has a light blue header with the title 'Consultation des interventions'. Below the header are two dropdown menus: 'Filtrer par année : Toutes les années' and 'Filtrer par semestre : Tous les semestres'. A table lists two interventions:

INTITULE_INTERVENTION	ETABLISSEMENT	ANNEE_UNIV	SEMESTRE	DATE_DEBUT	DATE_FIN	NBR_HEURES
Conférence sur le Metaverse	Faculté des Lettres et des Sciences Humaines	2022/2023	Semestre2	2023-01-29	2023-05-27	34
Conférence sur le Metaverse 2	Faculté des Lettres et des Sciences Humaines	2022/2023	Semestre1	2022-11-29	2023-01-27	34

Le professeur peut filtrer les interventions par semestre ou par années.

L'API consommées c'est :

```
4 |   },
5 | },
6 |
7 | async mounted(){
8 |   const response =await axios.get('/getintervention');
9 |   this.pfs=response.data
0 |   this.pfs.forEach(element => {
1 |     if (!this.years.includes(element.Annee_univ)) {
2 |       this.years.push(element.Annee_univ);
3 |     }
4 |   });
5 |
6 | }
```

Et finalement, la vue profile :

localhost:8080/Profileprof

Professeur

Paiement

Interventions

Profil

Déconnexion

aminetaha1964@gma... Prof

Modifier votre profil

PPR:

666666

Nom:

Soufian

Prénom:

yassin

Email:

aminetaha1964@gmail.com

Établissement:

Faculté des Lettres et des Sciences Humaines

Modifier

Qui fonctionne de même logique que les autres profiles.

Pour l'intégration d'API dans le site, j'ai travaillé en étroite collaboration avec les responsables du backend, ainsi de le responsable de git qui m'a aidé pour la partie responsive et style de site.

De plus, il a été nécessaire de coordonner avec le responsable de sécurité afin d'intégrer des mécanismes sécurisant les routes. Personne ne peut accéder à un dashboard sans passer par le login, et personne ne peut entrer à un dashboard s'il n'est pas du type dont l'accès est autorisé, et si un utilisateur non autorisé tente d'y accéder, une page l'informant de son absence de droit sera affichée.

4. CHAPITRE 4 :

LINUX

Dans ce chapitre, nous allons explorer la mise en place et la configuration de Linux, en nous concentrant sur la configuration des services indispensables tels que DNS, Apache2/NGNX, PostFix, Docker, etc.

4.1. POSTFIX



1- Etape1-Configuration de hostname :

D'abord on va commencer par avoir notre hostname :

```
abderrahmane@dns:~$ hostname
dns.hsup.uae.ma
abderrahmane@dns:~$ hostname -f
dns.hsup.uae.ma
abderrahmane@dns:~$ hostname --fqdn
dns.hsup.uae.ma
abderrahmane@dns:~$
```

ou bien on peut faire la commande suivant :

```
$ cd /etc/hostname
```

Aussi on va voir les autres hosts par la commande : cd /etc/hosts

GNU nano 6.2	/etc/hostname
dns.hsup.uae.ma	

2- Etape2- Installer Postfix

Dans cette étape, vous allez installer Postfix. Le moyen le plus rapide est d'installer le paquet mailutils qui regroupe Postfix avec quelques programmes supplémentaires que vous utiliserez pour tester l'envoi de courrier électronique.

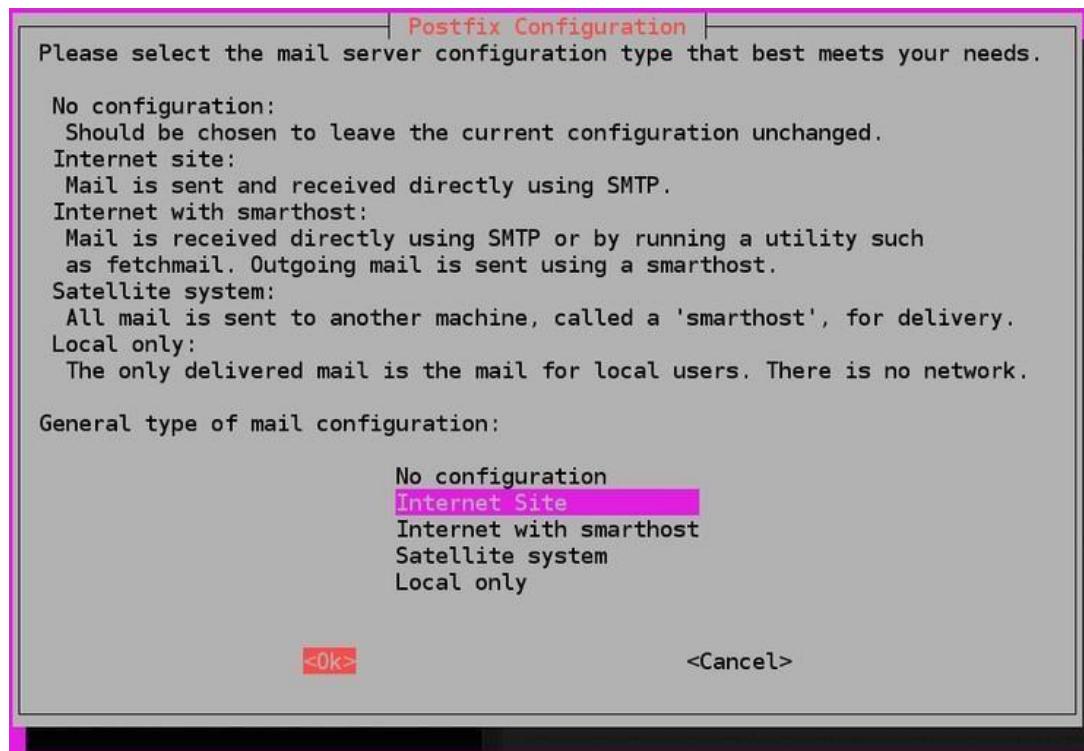
Tout d'abord, mettez à jour la base de données des paquets

```
$ sudo apt update
```

Ensuite, installez Postfix en exécutant la commande suivante :

```
$ sudo apt install mailutils
```

Vers la fin du processus d'installation, la fenêtre de configuration de Postfix vous sera présentée :

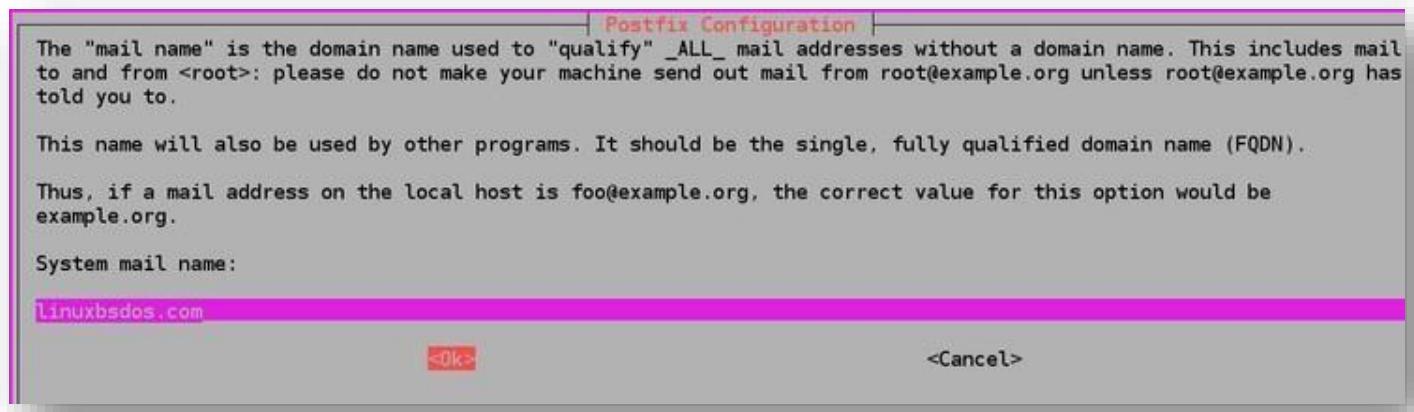


L'option par défaut est Site Internet. C'est l'option recommandée pour votre cas d'utilisation, donc appuyez sur TAB, puis ENTRÉE. Si vous ne voyez que le texte de la description, appuyez sur TAB pour sélectionner OK, puis sur ENTER.

s'affiche pas automatiquement, exécutez la commande suivante pour le démarrer :

```
$ sudo dpkg-reconfigure postfix
```

Après cela, vous obtiendrez une autre invite de configuration concernant le **nom de messagerie du système** :



Le **nom de messagerie du système** doit être le même que celui que vous avez attribué à votre serveur lors de sa création. Lorsque vous avez terminé, appuyez sur TAB, puis sur ENTRÉE.

Vous avez maintenant installé Postfix et vous êtes prêt à commencer à le configurer.

3- Etape3- Configuration Postfix

Au cours de cette étape, vous configurerez Postfix pour envoyer et recevoir des courriers électroniques uniquement à partir du serveur sur lequel il fonctionne, c'est-à-dire à partir de localhost.

Pour que cela arrive, Postfix doit être configuré pour écouter uniquement sur l'*interface de bouclage*, l'interface de réseau virtuel que le serveur utilise pour communiquer en interne. Pour effectuer les changements, vous devrez modifier le fichier de configuration principal de Postfix appelé main.cf, stocké sous etc/postfix.

Ouvrez-le pour l'éditer à l'aide de votre éditeur de texte préféré :

```
$ sudo nano /etc/postfix/main.cf
```

```

GNU nano 6.2                               /etc/postfix/main.cf
# See /usr/share/postfix/main.cf.dist for a commented, more complete version

# Debian specific: Specifying a file name will cause the first
# line of that file to be used as the name. The Debian default
# is /etc/mailname.
#myorigin = /etc/mailname

smtpd_banner = $myhostname ESMTP $mail_name (Ubuntu)
biff = no

# appending .domain is the MUA's job.
append_dot_mydomain = no

# Uncomment the next line to generate "delayed mail" warnings
#delay_warning_time = 4h

readme_directory = no

# See http://www.postfix.org/COMPATIBILITY_README.html -- default to 3.6 on
# fresh installs.
compatibility_level = 3.6

# TLS parameters
smtpd_tls_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
smtpd_tls_security_level=may

smtp_tls_CApth=/etc/ssl/certs
smtp_tls_security_level=may
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache

smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated defer_unauth_destination
myhostname = hsup.uae.ma
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = $myhostname, localhost, localhost.localdomain
relayhost = [smtp.gmail.com]:587
mynetworks = 127.0.0.0/8 [:ffff:127.0.0.0]/104 [:1]/128
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
inet_protocols = all

```

Ajouter les lignes suivantes :

```

# Enable SASL authentication
smtp_sasl_auth_enable = yes
smtp_sasl_security_options = noanonymous
smtp_sasl_password_maps = hash:/etc/postfix/sasl/sasl_passwd
smtp_tls_security_level = encrypt
smtp_tls_CAfile = /etc/ssl/certs/ca-certificates.crt

```

On va aussi ajouter ce code tout simplement pour personnaliser le format à envoyer

```

#template_email

message_content_type = text/html
html_directory = /usr/share/doc/postfix/html

```

Une autre directive que vous devrez modifier est mydestination, qui est utilisée pour spécifier la liste des domaines qui sont livrés via le transport de distribution de courrier local_transport. Par défaut, les valeurs sont similaires à celles-ci :

```
/etc/postfix/main.cf  
...  
mydestination = $myhostname, your_domain, localhost.com, , localhost  
...
```

Si votre domaine est en fait un sous-domaine, et que vous souhaitez que les messages électroniques aient l'air d'avoir été envoyés depuis le domaine principal, vous pouvez ajouter la ligne suivante à la fin de main.cf:

```
/etc/postfix/main.cf  
...  
masquerade_domains = your_main_domain
```

Le paramètre facultatif masquerade_domains spécifie pour quels domaines la partie desous-domaine sera supprimée dans l'adresse e-mail.

Lorsque vous avez terminé, enregistrez et fermez le fichier.

Note : Si vous hébergez plusieurs domaines sur un seul serveur, les autres domaines peuvent également être transmis à Postfix en utilisant la directive mydestination.

Ensuite, redémarrez Postfix en exécutant la commande suivante :

```
$ sudo systemctl restart postfix
```

4- Etape4- Préparation Envoi

D'abord on doit accéder à l'adresse mail qui va prendre en charge l'opération d'envoi des emails



- [Accueil](#)
- [Informations personnelles](#)
- [Données et confidentialité](#)
- [Sécurité](#)
- [Contacts et partage](#)
- [Paiements et abonnements](#)
- [À propos](#)



Bienvenue Abderrhamane test

Gérez vos informations, ainsi que la confidentialité et la sécurité de vos données pour profiter au mieux des services Google. [En savoir plus](#)

Confidentialité et personnalisation



Consultez les données de votre compte Google et sélectionnez les activités à enregistrer pour personnaliser votre expérience Google

[Gérer une donnée et une vie privée](#)

Des recommandations de sécurité sont disponibles



Consultez-les sur la page Check-up Sécurité

[Protéger votre compte](#)

Après cliquer sur validations en deux étape :

myaccount.google.com/u/1/security?hl=fr

Google Compte Rechercher dans le compte Google

Accueil Informations personnelles Données et confidentialité Sécurité Contacts et partage Paiements et abonnements À propos

Des recommandations de sécurité sont disponibles

Consultez-les sur la page Check-up Sécurité

[Protéger votre compte](#)

Activité récente liée à la sécurité de votre compte

Mot de passe d'application créé	05:47 - Maroc	>
Mot de passe d'application supprimé	05:47 - Maroc	>
Mot de passe d'application créé	29 avr. - Maroc	>

[Examiner l'activité liée à la sécurité \(19\)](#)

Comment vous connecter à Google

Assurez-vous que vous pouvez toujours accéder à votre compte Google en maintenant ces informations à jour

Validation en deux étapes	Activation : 24 avr.	>
Mot de passe	Dernière modification : 24 avr.	>
Téléphones pour la validation en deux étapes	0626-387972	>
Numéro de téléphone de récupération	0626-387972	>

Click

Apres on clique sur Mots de passe des applications

151

Mots de passe des applications

Les mots de passe d'application ne sont pas recommandés et sont inutiles dans la plupart des cas.
Pour sécuriser votre compte Google, utilisez Se connecter avec Google pour y associer des applis.



Après click sur « sélectionner une application » et choisis « autre »

← Mots de passe des applications

Les mots de passe d'application vous permettent de vous connecter à votre compte Google à partir d'applications sur des appareils non compatibles avec la validation en deux étapes. Comme vous ne devez saisir le mot de passe qu'une fois, vous n'avez pas besoin de le mémoriser. En savoir plus

← Mots de passe des applications

Les mots de passe d'application vous permettent de vous connecter à votre compte Google à partir d'applications sur des appareils non compatibles avec la validation en deux étapes. Comme vous ne devez saisir le mot de passe qu'une fois, vous n'avez pas besoin de le mémoriser. En savoir plus

Lorsque vous cliquez sur GÉNÉRER
il va donner un mot de passe

**Écrit par exemple mail
juste comme Nom**

Click

Maintenant on va régler l'émetteur :

```
$ sudo nano
```

```
GNU nano 6.2                               /etc/postfix/sasl/sasl_passwd
[smtp.gmail.com]:587 abdotest2023@gmail.com: [REDACTED]
```

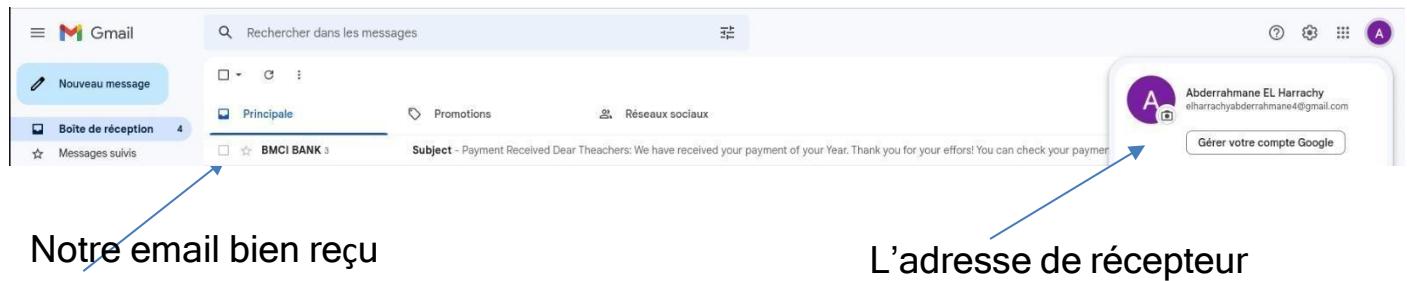
Port:587

Saisir votre mot de passe que vous obtenu dans l'étape précédent

5- Etape5-Tester le serveur SMTP

```
abderrahmane@dns:~$ sudo chown root:root /etc/postfix/sasl/sasl_passwd.db
abderrahmane@dns:~$ sudo chmod 600 /etc/postfix/sasl/sasl_passwd.db
abderrahmane@dns:~$ mail -a "Content-Type: text/html" -s "Subject" elharrachyabderrahmane4@gmail.com < /usr/share/doc/postfix/html/template.html
```

On vérifie dans notre boite mail



Subject : Boîte de réception

BMCI BANK <abdote2023@gmail.com>
À moi

anglais > français Traduire le message

Payment Received

Dear Teachers:

We have received your payment of your Year. Thank you for your efforts!

You can check your payment details by clicking on the link below:

[View Payment Details](#)

Thank you again for your payment. If you have any questions or concerns, please feel free to contact us.

Best regards,

The Payment Team

Lien de paiement de chaque prof



4.2. APACHE

4.2.1. Etape1- Configuration Avant l'installation :

Tout d'abord, mettez à jour la base de données des paquets

```
abderrahmane@dns:~$ sudo apt update && apt upgrade -y
```

Ensuite, modifions nos fichiers hôtes pour qu'ils correspondent au domaine que nous allons héberger.

```
abderrahmane@dns:~$ hostname  
dns.hsup.uae.ma  
abderrahmane@dns:~$ hostname --fqdn  
dns.hsup.uae.ma  
abderrahmane@dns:~$
```

4.2.2. Etape2- Installer Appache2:



Maintenant, nous pouvons installer Apache et configurer notre tout premier site Web. Reconnectez-vous à votre serveur via ssh cette fois en utilisant le nom d'utilisateur au lieu de root. Tapez ensuite la commande ci-dessous pour installer les paquets nécessaires, puis vérifiez s'il est en cours d'exécution.

```
abderrahmane@dns:~$ sudo apt install apache2 apache2-docs apache2-util
```

Maintenant pour avoir le statut d'Apache est ce que sa marche ou Non :

```
$ sudo systemctl status apache2
```

donne quelque chose comme cela :

```
abderrahmane@dns:~$ systemctl status apache2
● apache2.service - The Apache HTTP Server
  Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
  Active: active (running) since Wed 2023-05-17 08:25:31 +01; 6h ago
    Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 928 (apache2)
      Tasks: 55 (limit: 9277)
     Memory: 12.9M
        CPU: 1.951s
       CGroup: /system.slice/apache2.service
               └─ 928 /usr/sbin/apache2 -k start
                  ├─ 8519 /usr/sbin/apache2 -k start
                  ├─ 8520 /usr/sbin/apache2 -k start

May 17 08:25:31 dns.hsup.uae.ma systemd[1]: Starting The Apache HTTP Server...
May 17 08:25:31 dns.hsup.uae.ma systemd[1]: Started The Apache HTTP Server.
May 17 12:48:14 dns.hsup.uae.ma systemd[1]: Reloading The Apache HTTP Server...
May 17 12:48:14 dns.hsup.uae.ma systemd[1]: Reloaded The Apache HTTP Server.
May 17 12:52:20 dns.hsup.uae.ma systemd[1]: Reloading The Apache HTTP Server...
May 17 12:52:20 dns.hsup.uae.ma systemd[1]: Reloaded The Apache HTTP Server.
abderrahmane@dns:~$
```

À ce stade, vous pouvez voir s'il fonctionne sur un navigateur Web et vérifier si le domaine a complètement modifié si vous avez modifié des serveurs de noms. Copiez / collez simplement votre adresse IP dans la barre d'adresse du navigateur Web et vous devriez voir la page par défaut Apache2. Utilisez les commandes ci-dessous pour désactiver le site Web de default car nous créerons nos propres répertoires et configurations plus tard.

Chercher d'abord le fichier 000-default.conf

```
abderrahmane@dns:~$ cd /etc/apache2
abderrahmane@dns:/etc/apache2$ ls
apache2.conf      conf-enabled  magic          mods-enabled  sites-available
conf-available   envvars       mods-available ports.conf    sites-enabled
abderrahmane@dns:/etc/apache2$ cd sites-available
abderrahmane@dns:/etc/apache2/sites-available$ ls
000-default.conf  default-ssl.conf  hsup.conf
abderrahmane@dns:/etc/apache2/sites-available$ sudo a2dissite 000-default.conf
```

On redémarre Apache, et on voir leur statuts autre fois :

```
$ sudo systemctl reload
```

```
abderrahmane@dns:/etc/apache2/sites-available$ sudo systemctl reload apache2
abderrahmane@dns:/etc/apache2/sites-available$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
  Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
  Active: active (running) since Wed 2023-05-17 08:25:31 +01; 6h ago
    Docs: https://httpd.apache.org/docs/2.4/
   Process: 12560 ExecReload=/usr/sbin/apachectl graceful (code=exited, status=0/SUCCESS)
 Main PID: 928 (apache2)
    Tasks: 55 (limit: 9277)
   Memory: 10.5M
      CPU: 2.030s
     CGroup: /system.slice/apache2.service
             └─ 928 /usr/sbin/apache2 -k start
                  ├─12564 /usr/sbin/apache2 -k start
                  ├─12565 /usr/sbin/apache2 -k start

May 17 08:25:31 dns.hsup.uae.ma systemd[1]: Starting The Apache HTTP Server...
May 17 08:25:31 dns.hsup.uae.ma systemd[1]: Started The Apache HTTP Server.
May 17 12:48:14 dns.hsup.uae.ma systemd[1]: Reloading The Apache HTTP Server...
May 17 12:48:14 dns.hsup.uae.ma systemd[1]: Reloaded The Apache HTTP Server.
May 17 12:52:20 dns.hsup.uae.ma systemd[1]: Reloading The Apache HTTP Server...
May 17 12:52:20 dns.hsup.uae.ma systemd[1]: Reloaded The Apache HTTP Server.
May 17 14:53:07 dns.hsup.uae.ma systemd[1]: Reloading The Apache HTTP Server...
May 17 14:53:07 dns.hsup.uae.ma systemd[1]: Reloaded The Apache HTTP Server.
abderrahmane@dns:/etc/apache2/sites-available$
```

4.2.3. Etape3- Firewall Configuration :

Maintenant, nous devons activer et autoriser certaines applications avec notre pare-feu. L'application de pare-feu par défaut dans Ubuntu rend cela vraiment facile à faire. Pour voir une liste d'applications, exécutez simplement la commande ci-dessous.

```
$ sudo ufw app list
```

```
abderrahmane@dns:~$ sudo ufw app list
Available applications:
  Apache
  Apache Full
  Apache Secure
  Bind9
  CUPS
  OpenSSH
  Postfix
  Postfix SMTPS
  Postfix Submission
abderrahmane@dns:~$
```

Les applications que nous autoriserons sont Apache Full et OpenSSH

```
$ sudo ufw allow 'Apache Full'
```

```
$sudo ufw allow OpenSSH
```

```
abderrahmane@dns:~$ sudo ufw allow 'Apache Full'
Skipping adding existing rule
Skipping adding existing rule (v6)
abderrahmane@dns:~$ sudo ufw allow OpenSSH
Skipping adding existing rule
Skipping adding existing rule (v6)
abderrahmane@dns:~$ sudo ufw enable
Firewall is active and enabled on system startup
abderrahmane@dns:~$
```

De là, activons notre pare-feu et vérifions l'état.

```
$ sudo ufw status
```

```
abderrahmane@dns:~$ sudo ufw status
Status: active

To                         Action      From
--                         --          --
Apache                      ALLOW       Anywhere
Apache Full                 ALLOW       Anywhere
22/tcp                      ALLOW       Anywhere
OpenSSH                     ALLOW       Anywhere
20/tcp                      ALLOW       Anywhere
21/tcp                      ALLOW       Anywhere
4100:5100/tcp               ALLOW       Anywhere
990/tcp                     ALLOW       Anywhere
Postfix                     ALLOW       Anywhere
Bind9                       ALLOW       Anywhere
80                          ALLOW       Anywhere
443                         ALLOW       Anywhere
Apache (v6)                  ALLOW       Anywhere (v6)
Apache Full (v6)              ALLOW       Anywhere (v6)
22/tcp (v6)                  ALLOW       Anywhere (v6)
OpenSSH (v6)                  ALLOW       Anywhere (v6)
20/tcp (v6)                  ALLOW       Anywhere (v6)
21/tcp (v6)                  ALLOW       Anywhere (v6)
4100:5100/tcp (v6)           ALLOW       Anywhere (v6)
990/tcp (v6)                  ALLOW       Anywhere (v6)
Postfix (v6)                  ALLOW       Anywhere (v6)
Bind9 (v6)                   ALLOW       Anywhere (v6)
80 (v6)                      ALLOW       Anywhere (v6)
443 (v6)                     ALLOW       Anywhere (v6)

abderrahmane@dns:~$
```

4.2.4. Etape4- Creating Website:

La création de répertoires pour notre site Web est facile et se fera dans le répertoire

```
abderrahmane@dns:~$ cd /var/www
abderrahmane@dns:/var/www$ ls
hsup  hsup1
abderrahmane@dns:/var/www$
```

/var/www/html par défaut. Avec les commandes ci-dessous, remplacez 'example.com' par le domaine que vous utiliserez.

On va créer notre nouveau site Web

```
abderrahmane@dns:/var/www$ cd hsup
abderrahmane@dns:/var/www/hsup$ ls
css  favicon.ico  index.html  js
abderrahmane@dns:/var/www/hsup$
```

C'est le path de dossier :

```
abderrahmane@dns: $ cd ~/App
abderrahmane@dns:~/App$ ls
Linux-Server-docker-test
abderrahmane@dns:~/App$ cd Linux-Server-docker-test
abderrahmane@dns:~/App/Linux-Server-docker-test$ ls
api app docker-compose.yml README.md
abderrahmane@dns:~/App/Linux-Server-docker-test$ cd app
abderrahmane@dns:~/App/Linux-Server-docker-test/app$ ls
babel.config.js dist Dockerfile jsonconfig.json node_modules package.json package-lock.json postcss.config.js public README.md src tailwind.config.js vue.config.js
abderrahmane@dns:~/App/Linux-Server-docker-test/app$ cd dist
abderrahmane@dns:~/App/Linux-Server-docker-test/app/dist$ ls
css favicon.ico index.html js
abderrahmane@dns:~/App/Linux-Server-docker-test/app/dist$
```

On copie le fichier de deployment (dist) dans notre dossier (site Web).

```
abderrahmane@dns:~/App/Linux-Server-docker-test/app/dist$ ls
css favicon.ico index.html js
abderrahmane@dns:~/App/Linux-Server-docker-test/app/dist$ cd ..
abderrahmane@dns:~/App/Linux-Server-docker-test/app$ sudo cp dist /var/www/hsup
```

Fichier deployment Site Web

Une fois nos répertoires créés, nous allons revenir à l'emplacement des configurations de site et en créer un nouveau pour notre site Web.

```
$ cd /etc/apache2/sites-available
```

```
abderrahmane@dns:/etc/apache2/sites-available$ ls
000-default.conf default-ssl.conf hsup.conf
abderrahmane@dns:/etc/apache2/sites-available$ sudo nano hsup.conf
abderrahmane@dns:/etc/apache2/sites-available$
```

Dans cette configuration, vous voudrez copier / coller la configuration ci-dessous. Remplacer tout ce qui correspond à votre configuration.

```

GNU nano 6.2                                     hsup.conf
<VirtualHost *:80>
    ServerName hsup.uae.ma
    Redirect / https://hsup.uae.ma/               ← La redirection vers

<Directory /var/www/hsup>
    Options -Indexes +FollowSymLinks
    AllowOverride All
    Require all granted
</Directory>

</VirtualHost>

<VirtualHost *:443>
    ServerName hsup.uae.ma
    ServerAlias www.hsup.uae.ma
    DocumentRoot /var/www/hsup

    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/apache-selfsigned.crt
    SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key
</VirtualHost>

```

Concernant le certificat SSL

Maintenant, nous exécutons quelques commandes pour activer notre nouvelle configuration.

```

abderrahmane@dns:/etc/apache2/sites-available$ sudo a2ensite hsup
Site hsup already enabled
abderrahmane@dns:/etc/apache2/sites-available$ █

```

```

abderrahmane@dns:/etc/apache2/sites-available$ systemctl reload apache2
abderrahmane@dns:/etc/apache2/sites-available$ systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2023-05-17 08:25:31 +01; 6h ago
     Docs: https://httpd.apache.org/docs/2.4/
  Process: 13074 ExecReload=/usr/sbin/apachectl graceful (code=exited, status=0/SUCCESS)
 Main PID: 928 (apache2)
    Tasks: 55 (limit: 9277)
   Memory: 10.5M
      CPU: 2.149s
     CGroup: /system.slice/apache2.service
             └─ 928 /usr/sbin/apache2 -k start
               ├─13078 /usr/sbin/apache2 -k start
               ├─13079 /usr/sbin/apache2 -k start

May 17 08:25:31 dns.hsup.uae.ma systemd[1]: Starting The Apache HTTP Server...
May 17 08:25:31 dns.hsup.uae.ma systemd[1]: Started The Apache HTTP Server.
May 17 12:48:14 dns.hsup.uae.ma systemd[1]: Reloading The Apache HTTP Server...
May 17 12:48:14 dns.hsup.uae.ma systemd[1]: Reloaded The Apache HTTP Server.
May 17 12:52:20 dns.hsup.uae.ma systemd[1]: Reloading The Apache HTTP Server...
May 17 12:52:20 dns.hsup.uae.ma systemd[1]: Reloaded The Apache HTTP Server.
May 17 14:53:07 dns.hsup.uae.ma systemd[1]: Reloading The Apache HTTP Server...
May 17 14:53:07 dns.hsup.uae.ma systemd[1]: Reloaded The Apache HTTP Server.
May 17 15:06:59 dns.hsup.uae.ma systemd[1]: Reloading The Apache HTTP Server...
May 17 15:06:59 dns.hsup.uae.ma systemd[1]: Reloaded The Apache HTTP Server.
abderrahmane@dns:/etc/apache2/sites-available$
```

Maintenant, si vous voyagez sur votre site Web, vous devriez voir la page standard 'Index de /' signifiant jusqu'à présent, nous avons réussi ! De là, vous pouvez créer un fichier index.html pour confirmer à coup sûr. Voir l'exemple dans la vidéo de la façon de procéder.

4.2.5. Etape5- Installing SSL Certification

Un certificat SSL est un certificat numérique qui authentifie l'identité d'un site Web et permet une connexion cryptée. SSL signifie Secure Sockets Layer, un protocole de sécurité qui crée un lien crypté entre un serveur Web et un navigateur Web. Dans le Web moderne, c'est une exigence de base. Pour installer un certificat SSL, assurez-vous que votre domaine est correctement lié à votre nouveau serveur Apache et suivez les étapes ci-dessous.

```
$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt
```

```

abderrahmane@dns: $ sudo azenmod ssl
[sudo] password for abderrahmane:
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Module socache_shmcb already enabled
Module ssl already enabled
abderrahmane@dns: $ sudo systemctl restart apache2
abderrahmane@dns: $ sudo openssl req -x509 -nodes 365 -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt
...
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
Country Name (2 letter code) [AU]:MAROC
String too long, must be at most 2 bytes long
Country Name (2 letter code) [AU]:MR
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:hsup.uae.ma
Email Address []:abdoteest2023@gmail.com
Email Address []:abdoteest2023@gmail.com
...

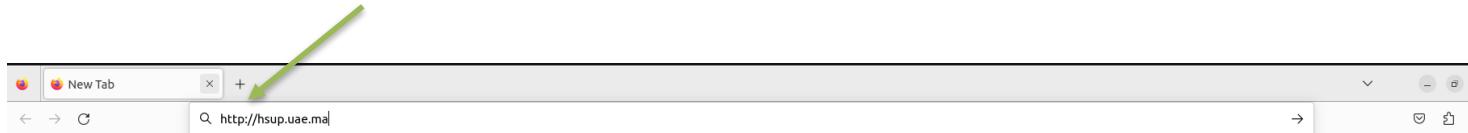
```

Les informations de certificat qui sera afficher sur certificat

Apres ca il faut remplir les informations .

4.2.6. Etape6- Teste de Site Web

HTTP



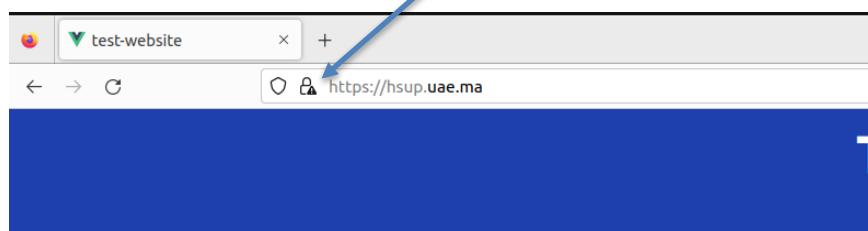
On Remarque il fait la direction directement vers https de plus il est en relation avec Vuejs.



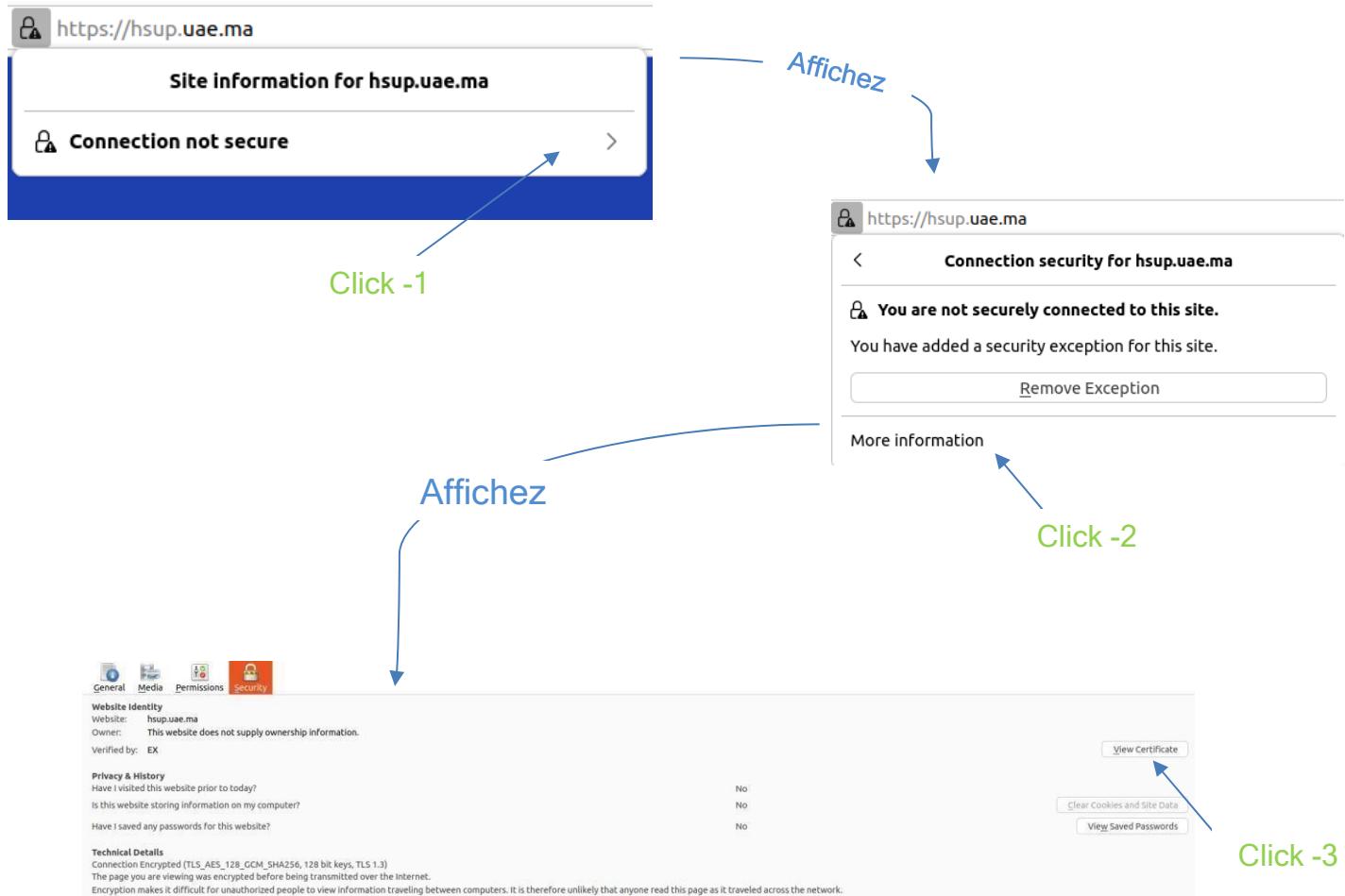
This is just a test website, Don't expect anything.

4.2.7. Etape7- Security de Site Web:

On Remarque que on un signe Serrure, Si on Click



Il va afficher l'icône suivant :



XTRADH

Certificate

hsup.uae.ma	
Subject Name	
Country	MR
State/Province	Some-State
Organization	Internet Widgits Pty Ltd
Common Name	hsup.uae.ma
Email Address	abdotest2023@gmail.com
Issuer Name	
Country	MR
State/Province	Some-State
Organization	Internet Widgits Pty Ltd
Common Name	hsup.uae.ma
Email Address	abdotest2023@gmail.com
Validity	
Not Before	Wed, 17 May 2023 14:15:25 GMT
Not After	Thu, 16 May 2024 14:15:25 GMT
Public Key Info	
Algorithm	RSA
Key Size	2048
Exponent	65537
Modulus	BC:4D:B5:A9:52:37:B6:14:70:10:43:6B:58:D9:8C:34:3D:E6:45:35:32:90:5A:E1...

Ci-dessus est notre certificat.

Cependant, si vous utilisez un certificat auto-signé, il n'est pas signé par une autorité de certification reconnue. Par conséquent, les navigateurs et les systèmes d'exploitation considéreront ce certificat comme non fiable par défaut. Lorsque vous accédez à un site web avec un certificat auto-signé, vous recevrez généralement un avertissement de sécurité indiquant que le certificat n'est pas valide.

Cela signifie que bien que votre serveur web ait un certificat SSL installé, il sera considéré comme non fiable par les navigateurs des utilisateurs, ce qui entraînera des avertissements de sécurité.

4.3. NGINX



4.3.1. Étape 1 : l'installation de "Nginx":

Nginx est disponible dans les référentiels par défaut d'Ubuntu, l'installation est donc assez simple.

Système de packaging apt dans cette session. Nous mettrions à jour le index de packages local ; afin que nous ayons accès aux listes de packages les plus récentes.

Ensuite, nous pouvons installer Nginx en utilisant la commande suivante



Bash Terminal (Ubuntu)

```
> sudo apt-get update  
> sudo apt-get install nginx
```

Après avoir accepté la procédure, apt-get installera Nginx et toutes les dépendances requises sur votre serveur.

4.3.2. Étape 2 : Gérer le processus "Nginx":

Maintenant que le serveur Web est opérationnel, nous pouvons utiliser certaines commandes de gestion de base pour contrôler l'état du serveur.

Pour arrêter votre serveur Web:



Bash Terminal (Ubuntu)

> sudo systemctl stop

```
sudo systemctl stop nginx
*le serveur est inactive
sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: inactive (dead) since Sun 2023-05-21 20:34:37 WEST; 2s ago
     Docs: man:nginx(8)
   Process: 3929 ExecStop=/sbin/start-stop-daemon --quiet --stop --retry QUIT/5 --pidfile /run/nginx.pid
 Main PID: 879 (code=exited, status=0/SUCCESS)
    CPU: 61ms

mai 21 19:55:01 mohamed-X541UJ systemd[1]: Starting A high performance web server and a reverse p>
mai 21 19:55:01 mohamed-X541UJ nginx[872]: nginx: [warn] the "ssl" directive is deprecated, use t>
mai 21 19:55:01 mohamed-X541UJ nginx[878]: nginx: [warn] the "ssl" directive is deprecated, use t>
mai 21 19:55:01 mohamed-X541UJ systemd[1]: Started A high performance web server and a reverse pr>
mai 21 20:34:37 mohamed-X541UJ systemd[1]: Stopping A high performance web server and a reverse p>
mai 21 20:34:37 mohamed-X541UJ systemd[1]: nginx.service: Deactivated successfully.
mai 21 20:34:37 mohamed-X541UJ systemd[1]: Stopped A high performance web server and a reverse pr>
```

1. Pour démarrer le serveur Web lorsqu'il est arrêté :



Bash Terminal (Ubuntu)

> sudo systemctl start

```

[sudo] systemctl start nginx
[sudo] systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
  Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
  Active: active (running) since Sun 2023-05-21 20:37:55 WEST; 2s ago
    Docs: man:nginx(8)
   Process: 3971 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0)
   Process: 3972 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0)
 Main PID: 3973 (nginx)
    Tasks: 5 (limit: 9307)
   Memory: 5.5M
      CPU: 40ms
     CGroup: /system.slice/nginx.service

```

*Le serveur est maintenant actif et en cours d'exécution

Si vous apportez simplement des modifications de configuration, NGINX peut souvent recharger sans interrompre les connexions. Pour ce faire, cette commande peut être utilisée



Bash Terminal (Ubuntu)

> sudo systemctl reload nginx // refrecher le serveur

Pour arrêter puis remettre le service



Bash Terminal (Ubuntu)

> sudo systemctl restart nginx // redémarrer le serveur

Par défaut, NGINX est configuré pour démarrer automatiquement au démarrage du serveur. Si ce n'est pas ce que vous voulez, vous pouvez désactiver ce comportement en tapant :



Bash Terminal (Ubuntu)

> sudo systemctl disable nginx // désactiver

Pour réactiver le service afin qu'il démarre au démarrage



```
> sudo systemctl enable nainx // ré-activer
```

Pour tester votre fichier de configuration, afin de détecter d'éventuelles erreurs de syntaxe ou de configuration avant de relancer le serveur



```
> sudo nginx -t // tester le fichier conf
```

```
sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

Dans ce cas "hsup-nginx" est le fichier de configuration utilisé pour configurer le serveur nginx pour nos sites Web et le test indique qu'il n'y a pas de syntaxe ni d'erreur de configuration

4.3.3.

Étape 3 : Fichier et Repertoire:

Contenu :

/var/www/html : le contenu Web réel, qui par défaut se compose uniquement de la page Nginx par défaut illustrée dans la capture d'écran, est diffusé à partir du répertoire /var/www/html.

Cela peut être modifié en modifiant les fichiers de configuration NGINX.



Configuration du serveur:

/etc/nginx : le répertoire de configuration de NGINX. Tous les fichiers de configuration Nginx résident ici.

```
total 64K
drwxr-xr-x 2 root root 4,0K nov. 10 2022 conf.d
-rw-r--r-- 1 root root 1,1K juil. 27 2022 fastcgi.conf
-rw-r--r-- 1 root root 1,1K juil. 27 2022 fastcgi_params
-rw-r--r-- 1 root root 2,8K juil. 27 2022 koi-utf
-rw-r--r-- 1 root root 2,2K juil. 27 2022 koi-win
-rw-r--r-- 1 root root 3,9K août 2 2022 mime.types
drwxr-xr-x 2 root root 4,0K nov. 10 2022 modules-available
drwxr-xr-x 2 root root 4,0K mai 18 23:09 modules-enabled
-rw-r--r-- 1 root root 1,5K juil. 27 2022 nginx.conf
-rw-r--r-- 1 root root 180 juil. 27 2022 proxy_params
-rw-r--r-- 1 root root 636 juil. 27 2022 scgi_params
drwxr-xr-x 2 root root 4,0K mai 18 23:09 sites-available
drwxr-xr-x 2 root root 4,0K mai 21 21:15 sites-enabled
drwxr-xr-x 2 root root 4,0K mai 18 23:09 snippets
-rw-r--r-- 1 root root 664 juil. 27 2022 uwsgi_params
-rw-r--r-- 1 root root 3,0K juil. 27 2022 win-utf
```

***Tous les fichiers reliés à la configuration du serveur NGINX résident dans cette répertoire**



/etc/nginx/nginx.conf : fichier de configuration principal de NGINX. Cela peut être modifié pour apporter des modifications à la configuration globale de NGINX.

```
user www-data;
worker_processes auto;
pid /run/nginx.pid;
include /etc/nginx/modules-enabled/*.conf;

events {
    worker_connections 768;
    # multi_accept on;
}

http {
    ##
    # Basic Settings
    ##

    sendfile on;
    tcp_nopush on;
    types_hash_max_size 2048;
    # server_tokens off;
}

# gzip_http_version 1.1;
# gzip_types text/plain text/css application/json application/xml application/xml+rss text/javascript;

##
## Virtual Host Configs
##
include /etc/nginx/conf.d/*.conf;
include /etc/nginx/sites-enabled/*;
```

The code block shows the main configuration file for NGINX. It includes basic settings like user, worker processes, and pid. The http block contains basic settings like sendfile and types_hash_max_size. The configuration is then included from the conf.d and sites-enabled directories.

Ce qui est très important, c'est de s'assurer que les fichiers de configuration appropriés tels que les fichiers .conf et les fichiers du dossier sites-enabled sont inclus comme indiqués dans la capture d'écran.

/etc/nginx/sites-available/ : le répertoire dans lequel les "blocs de serveur" parsites peuvent être stockés. NGINX n'utilisera pas les fichiers de configuration trouvés dans ce répertoire au moins qu'ils ne soient liés au répertoire sites-enabled (voir ci-dessous). En règle générale, toute la configuration du bloc de serveur est effectuée dans ce répertoire, puis activée en établissant un lien vers l'autre répertoire.

/etc/nginx/snippets : ce répertoire contient des fragments de configuration qui peuvent être inclus ailleurs dans la configuration NGINX. Les segments de configuration potentiellement reproductibles sont de bons candidats pour la refactorisation en extraits.

/etc/nginx/sites-enabled/ : le répertoire dans lequel les "blocs de serveur" activés par site sont stockés. En règle générale, ceux-ci sont créés en ayant un lien vers les fichiers de configuration trouvés dans le répertoire des sites disponibles.

```
ls -lai
total 12
820043 drwxr-xr-x 2 root root 4096 mai 21 21:15 .
819905 drwxr-xr-x 8 root root 4096 mai 20 23:37 ..
820104 lrwxrwxrwx 1 root root 34 mai 18 23:09 default -> /etc/nginx/sites-available/default
788990 -rw-r--r-- 1 root root 623 mai 21 21:15 hsup-nginx
```

*Le fichier de conf par défaut

*Le fichier de configuration de notre serveur

Journaux du serveur

/var/log/nginx/access.log : chaque demande envoyée à notre serveur Web est enregistrée dans ce fichier journal, au moins et jusqu'à ce que NGINX soit configuré pour faire autrement.

/var/log/nginx/error.log : chaque erreur NGINX sera enregistrée dans ce journal uniquement.

4.3.4. Étape 4 : Configuration du serveur:

Nous devons configurer NGINX pour servir notre site Web.

cd dans /etc/nginx/. C'est là que se trouvent les fichiers de configuration NGINX.

Les deux répertoires qui nous intéressent sont les sites disponibles et les sites activés.

sites-available : contient des fichiers de configuration individuels pour tous vos sites Web statiques possibles.

sites-enabled : des liens vers les fichiers de configuration que NGINX va évidemment lire et contenir.

Maintenant, **cd dans /etc/nginx/sites-enabled** et éditez le fichier par défaut en utilisant neo-vim ou nano selon votre choix comme suit :

Partie http (Port 80):

Dans ce cas, nous avons choisi de servir le site Web à l'adresse "**192.168.1.109**"

```
server {
    listen 80;
    listen 192.168.1.109:80;
    server_name www.hsup.uae.ma hsup.uae.ma;
    return 301 https://$server_name$request_uri;

    root /vue-app/dist;
    index index.html index.htm;

    location / {
        try_files $uri $uri/ /index.html;
    }
}
```

- **listen 80** : Le port 80 est le numéro de port attribué au protocole de communication Internet couramment utilisé, le protocole de transfert hypertexte(HTTP). Il s'agit du port réseau par défaut utilisé pour envoyer et recevoir des pages Web non chiffrées.
- **listen 192.168.1.109:80** : Le serveur doit écouter les requêtes qui passent par cette adresse IP "192.168.1.109" et elles doivent être servies par ce bloc de serveur.
- **server_name www.hsup.uae.ma hsup.uae.ma** : Le nom du serveur qui doit être configuré dans /etc/hosts afin que tout le trafic et les requêtes passant par "**192.168.1.109:80**" soient redirigés vers "www.hsup.uae.ma" et/ou "**hsup.uae.ma**"

XTRADH

```
#127.0.0.1      localhost
#127.0.1.1      mohamed-X541UJ
# 0.0.0.0          hsup.uae.ma
# 0.0.0.0          www.hsup.uae.ma

# Local Network Acess:
192.168.1.109  hsup.uae.ma
192.168.1.109  www.hsup.uae.ma

# 192.168.1.151 hsup.uae.ma
# 192.168.1.151 www.hsup.uae.ma

# The following lines are desirable for IPv6 capable hosts
::1      ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
/etc/hosts
```

1,1 All

*rediriger tous les requêtes et le traffic qui vient à travers "192.168.1.109" vers les adresses: "hsup.uae.ma" et "www.hsup.uae.ma"

root /app/dist : Le répertoire racine où l'on peut trouver le dossier de déploiement du projet vue.js.

renvoie 301 https://\$server_name\$request_uri ; : rediriger tout le trafic provenant du site http vers le site https

```
location / {
try_files $uri $uri/ /index.html;
}
```

Utilisez le fichier index.html et utilisez-le comme FallBackAddress vers d'autres routes et points de terminaison

Partie https (Port 443):

```
server {
    listen 443 ssl;
    listen 192.168.1.109:443;
    server_name www.hsup.uae.ma hsup.uae.ma;

    root /vue-app/dist;
    index index.html index.htm;

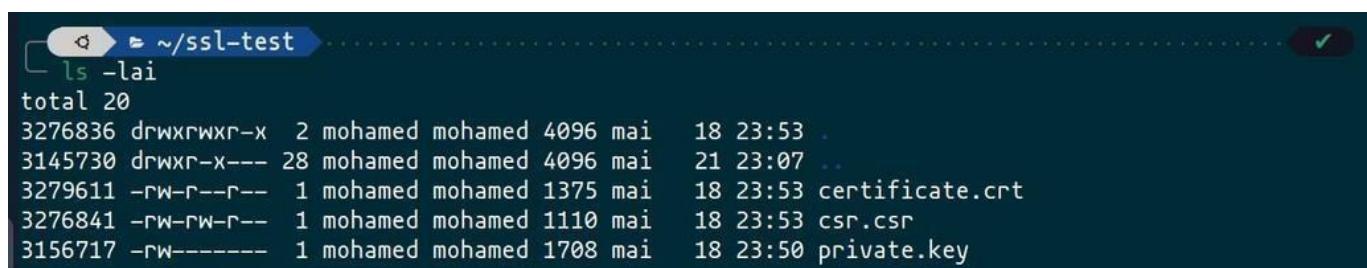
    ssl_certificate /home/mohamed/ssl-test/certificate.crt;
    ssl_certificate_key /home/mohamed/ssl-test/private.key;
    location / {
        try_files $uri $uri/ /index.html;
    }
}
```

listen 443 ssl: : Port 443. L'Internet Engineering Task Force (IETF) reconnaît le numéro de port TCP 443 comme protocole HTTPS par défaut. Il fournit un algorithme de cryptage pour l'échange d'informations entre les serveurs Web et les navigateurs. Le port HTTPS 443 fonctionne en sécurisant les paquets de trafic réseau avant que la transmission de données ne se produise.

SSL (Secure Sockets Layer) et son successeur, **TLS** (Transport Layer Security), sont des protocoles permettant d'établir des liens authentifiés et cryptés entre des ordinateurs en réseau.

ssl_certificate /home/mohamed/ssl-test/certificate.crt; **ssl_certificate_key** /home/mohamed/ssl-test/private.key ;

L'emplacement du fichier pour le certificat SSL auto-signé et la clé de certificat SSL. Ce certificat est auto-signé et indique au navigateur "**NON SÉCURISÉ**" lors de l'utilisation du site Web. Pour plus de détails, consultez la partie Configuration SSL



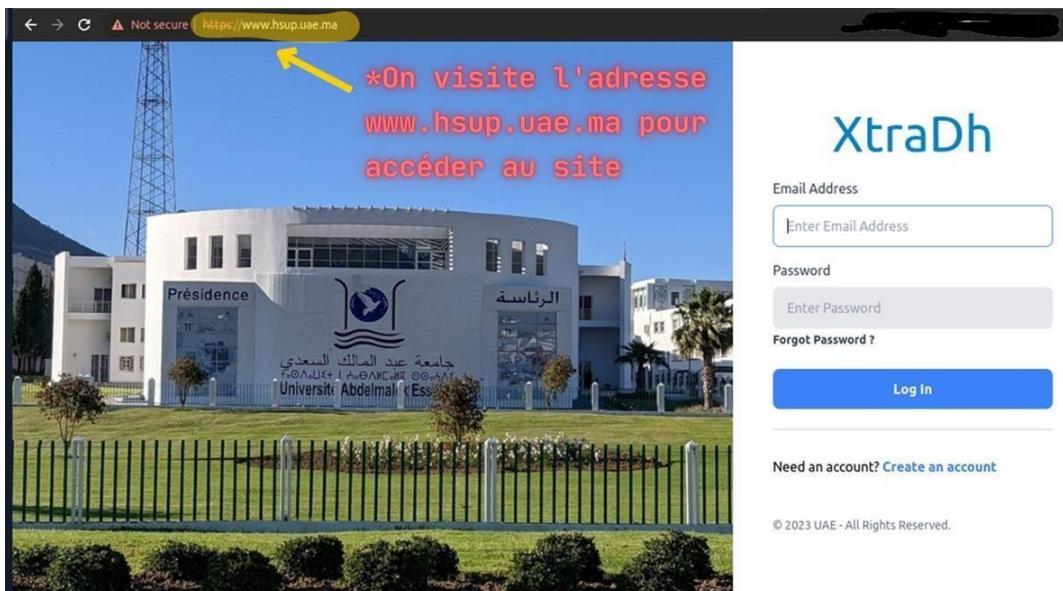
```
ls -lai
total 20
3276836 drwxrwxr-x  2 mohamed mohamed 4096 mai  18 23:53 .
3145730 drwxr-x--- 28 mohamed mohamed 4096 mai  21 23:07 ..
3279611 -rw-r--r--  1 mohamed mohamed 1375 mai  18 23:53 certificate.crt
3276841 -rw-rw-r--  1 mohamed mohamed 1110 mai  18 23:53 csr.csr
3156717 -rw-------  1 mohamed mohamed 1708 mai  18 23:50 private.key
```

Tous les autres paramètres sont les mêmes que la partie HTTP

Après avoir terminé la configuration, nous pouvons démarrer le serveur afin de tester les fonctionnalités que nous avons configurées précédemment

```
~> ~/ssl-test
[sudo] password for mohamed:
● nginx.service - A high performance web server and a reverse proxy server
  Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
  Active: active (running) since Sun 2023-05-21 20:37:55 WEST; 2h 31min ago
    Docs: man:nginx(8)
 Process: 3971 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0)
 Process: 3972 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0)
 Main PID: 3973 (nginx)
   Tasks: 5 (limit: 9307)
  Memory: 7.0M
     CPU: 72ms
    CGroup: /system.slice/nginx.service
            └─3973 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
              ├─3974 "nginx: worker process" " "
              ├─3975 "nginx: worker process" " "
              ├─3976 "nginx: worker process" " "
              ├─3977 "nginx: worker process" " "
              └─3978 "nginx: worker process" " "
```

On le test sur les navigateurs (Le site Web présenté dans ce rapport est différent de la partie apache2 car de nombreuses mises à jour ont été effectuées au cours de la période ultérieure.)



XTRADH

Nous utilisons l'onglet réseau du navigateur Chrome/Mozilla pour surveiller le trafic du site Web

The screenshot shows the Network tab in the Chrome DevTools. A request to `www.hsup.uae.ma` is selected. The Headers panel shows the following details:

- Request URL: `https://www.hsup.uae.ma/`
- Request Method: GET
- Status Code: 200 OK
- Remote Address: 192.168.1.109:443
- Referrer Policy: strict-origin-when-cross-origin

A yellow arrow points from the 'Remote Address' field to the text "On voit l'adresse et le port utilisé".

*On voit
l'adresse
et le
port
utilisé

Si nous essayons d'accéder à "`http://www.hsup.uae.ma`", le serveur lèvera un code d'état 301 et redirigera la requête vers l'adresse HTTPS

The screenshot shows the Network tab in the Chrome DevTools. A request to `http://www.hsup.uae.ma/` is selected. The Headers panel shows the following details:

- Request URL: `http://www.hsup.uae.ma/`
- Request Method: GET
- Status Code: 301 Moved Permanently
- Remote Address: 192.168.1.109:80
- Referrer Policy: strict-origin-when-cross-origin

A yellow arrow points from the 'Status Code' field to the text "rediriger vers HTTPS".

*rediriger
vers HTTPS

La même chose s'est adaptée si l'utilisateur a essayé d'accéder à l'adresse IP du serveur

XTRADH

The screenshot shows a browser developer tools Network tab. A request for '192.168.1.109' has been blocked and redirected to 'www.hsup.uae.ma'. The status code is 301 Moved Permanently. The response headers include Content-Length: 178, Content-Type: text/html, Date: Sun, 21 May 2023 21:57:51 GMT, Location: https://www.hsup.uae.ma/, and Server: nginx/1.18.0 (Ubuntu).

***rediriger vers HTTPS**

Email Address
Enter Email Address
Password
Enter Password
Forgot Password ?
Log In
Need an account? [Create an account](#)
© 2023 UAE - All Rights Reserved.

Nous remarquons que toutes les routes principales et secondaires fonctionnent également correctement

***le sous-route www.hsup.uae.ma/Gestionp fonctionne**

The screenshot shows a web application interface. On the left, a sidebar menu includes 'Gestion des profs' and 'Gestion des Interventions'. The main content area displays a table titled 'Listes des profs' with columns PPR, NOM, PRENOM, DATE_NAISSANCE, and ETABLISSEMENT. A green button labeled 'Ajouter prof' is visible. The URL in the browser bar is https://www.hsup.uae.ma/Gestionp.

***Une adresse aléatoire**

Welcome to your home

404 | Not found
BACK TO LOGIN

The screenshot shows a 404 Not Found error page. The header says 'Welcome to your home'. Below it, the error message '404 | Not found' is displayed, along with a 'BACK TO LOGIN' link. The URL in the browser bar is https://www.hsup.uae.ma/Gestqhqh.

4.4. APACHE VS NGINX



4.4.1. Qu'est-ce que NGINX ?

NGINX (également écrit Nginx) est un logiciel open source utilisé pour le service Web, le proxy inverse, la mise en cache, l'équilibrage de charge, le streaming multimédia, etc.

NGINX est entré en scène en 2004, lorsqu'il a été rendu public pour la première fois par le développeur russe Igor Sysoev. Lors de sa sortie, NGINX était principalement utilisé pour servir des fichiers statiques, mais aujourd'hui, il est devenu un serveur Web complet qui traite l'ensemble des tâches de serveur.

4.4.2. Avantages de NGINX :

Le serveur **NGINX** présente de nombreux avantages. Certains d'entre eux sont les suivants :

- **NGINX** est open source et son utilisation est gratuite.
- **NGINX** a un système de configuration simple, contrairement à Apache. L'utilisateur ne doit pas modifier plus de deux fichiers.
- **Les performances de NGINX** lorsqu'il s'agit de servir des fichiers statiques tels que pdf, zip, HTML, mp4, MPEG, Avi-, jpg, gif, png et autres sont excellentes.

• **NGINX** gère le trafic de n'importe quelle application Web sans aucune difficulté et fournit des guides de configuration pour la plupart des applications. Diverses applications telles que WordPress, python, ruby, Drupal, VBulletin, PhpBB et de nombreuses autres applications populaires sont servies par NGINX.

• **NGINX** est supérieur en matière de gestion des connexions simultanées, de temps de réponse et d'utilisation des ressources par rapport à d'autres serveurs Web tels qu'Apache ou Lighttpd.

• **NGINX** rend les sites Web plus rapides, leur permettant ainsi d'obtenir un classement plus élevé par Google, car ils ont déjà apporté diverses modifications à l'algorithme de classement de recherche et, récemment, ils ont également intégré la vitesse du site Web dans l'algorithme. Les sites Web plus rapides bénéficient d'une assistance maximale de Google. NGINX peut gérer plusieurs demandes de clients simultanément et efficacement avec un trafic important. Cela rend le site Web plus optimisé et plus rapide.

4.4.3. **What is Apache?**

• Apache (également connu sous le nom de Apache HTTP Server ou httpd) est un serveur Web multiplateforme open source qui fournit du contenu Web via Internet. Il a été conçu par Tim Berners Lee et publié en 1995. Apache est rapidement devenu populaire et est maintenant maintenu par l'Apache Software Foundation. Comme tous les serveurs Web, Apache héberge du contenu Web, tel que des pages HTML, des fichiers PHP, des fichiers audio et vidéo, et fournit le contenu aux utilisateurs lorsqu'ils visitent un site Web.

• Apache est le premier choix parmi les administrateurs de serveur en raison de sa simplicité architecturale, de sa flexibilité, de sa compatibilité de puissance et de sa prise en charge multiplateforme. Il est compatible avec presque tous les systèmes d'exploitation tels que Windows, UNIX, OSX, NetWare, etc. Mais il est couramment utilisé en combinaison avec Linux.

4.4.4. **Avantages d'Apache :**

• **Apache** est open source, ce qui le rend gratuit et permet aux développeurs de la communauté de l'étendre et de le modifier facilement.

• **Apache** est flexible grâce à la présence de modules dynamiques.

• **Apache** est compatible avec les systèmes d'exploitation de type Unix tels que Linux, BSD, etc. et MS Windows.

• **Apache** a une énorme communauté et un support facilement disponible en cas de problème.

- **Apache** a une délivrabilité optimale pour les fichiers statiques et est compatible avec n'importe quel langage de programmation.

4.4.5.

Principales différences entre NGINX et Apache :

• **NGINX** est un serveur Web open source hautes performances et un serveur proxy inverse, tandis qu'**Apache** est un serveur HTTP open source. **NGINX** étant un serveur proxy inverse, il se trouve derrière le pare-feu dans un réseau privé et dirige les demandes des clients vers le serveur principal approprié, assurant un flux fluide du trafic réseau entre les clients et les serveurs.

• La différence fondamentale entre les deux réside dans leur architecture. **NGINX** a une architecture à thread unique dans laquelle plusieurs requêtes client sont traitées dans un seul thread. D'autre part, **Apache** a une architecture multi-thread où un processus est créé pour une requête. L'architecture événementielle de **NGINX** garantit de meilleures performances même en cas de trafic intense.

• **NGINX** est bien équipé pour gérer le contenu statique. C'est 2,5 fois plus rapide qu'**Apache**. En ce qui concerne le contenu dynamique, **Apache** traite le contenu dynamique de manière native au sein du serveur Web lui-même, tandis que **NGINX** ne peut pas traiter le contenu dynamique en interne et dépend de processus externes pour l'exécution.

• Les serveurs **NGINX** et **Apache** sont tous deux excellents pour gérer la sécurité, mais la base de code **NGINX** est nettement plus petite, c'est donc un gros plus du point de vue de la sécurité.

• **NGINX** prend entièrement en charge presque tous les systèmes d'exploitation Unix, mais la prise en charge de Windows est partielle. D'autre part, **Apache** prend entièrement en charge les systèmes d'exploitation de type Unix et MS Windows.

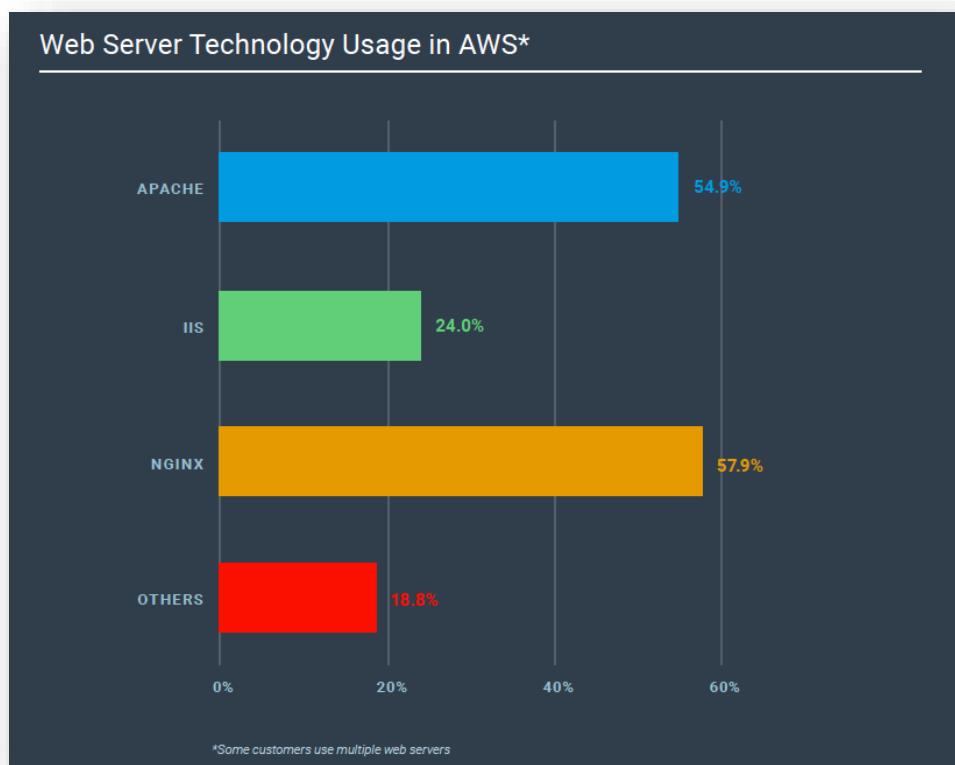
4.4.6.

Conversion NGINX, Pourquoi ?

Il est juste de dire que NGINX et Apache offrent des performances de qualité - ils sont flexibles, capables et puissants. Les deux ont leur place sur le marché actuel.

Mais NGINX est celui qui a le vent en poupe. Lorsqu'il était nouveau, Apache a capturé l'espace du serveur Web, mais il y a eu progressivement un changement. NGINX est léger et possède une architecture événementielle. Grâce à quoi il peut gérer plus efficacement un trafic Web important. De plus, il est plus rapide que le traitement de fichiers statiques. Ces

fonctionnalités ont plus de valeur aujourd'hui où le streaming et la consommation d'Internet sont à leur apogée. Par conséquent, les développeurs trouvent l'architecture NGINX plus adaptée au développement d'applications Web modernes. Cela est évident par le graphique ci-dessous montrant l'utilisation des serveurs Web sélectionnés pour les sites Web



4.5. DNS



4.5.1. Qu'est-ce que c'est le *DNS* (Domain Name System) ?

• Pour faciliter la recherche d'un site donné sur Internet, le système de noms de domaine (DNS) a été inventé. Le DNS permet d'associer un nom compréhensible, à une adresse IP. On associe donc une adresse logique, le nom de domaine, à une adresse physique l'adresse IP.

• Le nom de domaine et l'adresse IP sont uniques. Le DNS permet à votre message d'atteindre son destinataire et non quelqu'un d'autre possédant un nom de domaine similaire. Il vous permet également de taper "www.hsup.uae.ma" sans avoir à saisir une longue adresse IP et d'accéder au site web approprié.

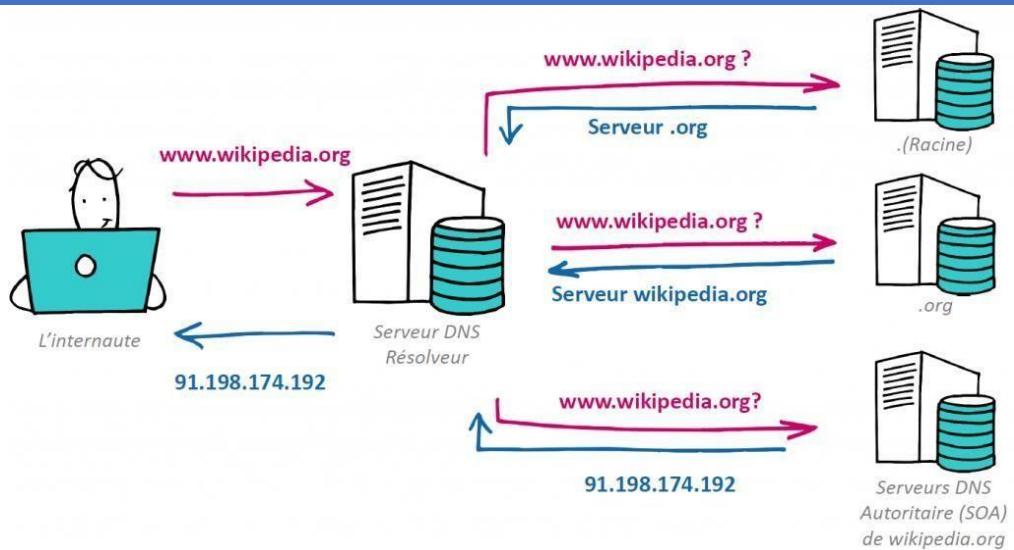
Résolution DNS

Lorsqu'un internaute a saisi une adresse dans son navigateur, c'est donc un serveur DNS qui traduit cette adresse humainement compréhensible, en une adresse IP, compréhensible par les ordinateurs et les réseaux. L'adresse www.hsup.uae.ma est ainsi traduite en 192.168.1.109.

On appelle cela la "**résolution DNS**".

Ce temps est d'autant plus faible que le serveur est performant : CPU (processeur), accès disque, et RAM (mémoire) doivent être correctement dimensionnés.

L'architecture logique du DNS est calquée sur la structure hiérarchique des noms de domaine.



Le DNS d'un niveau hiérarchique donné « délègue » au niveau inférieur le soin de traiter le sous-domaine suivant, jusqu'au dernier niveau qui, lui, connaît l'adresse IP correspondante au nom de domaine demandé.

4.5.2. Configuration DNS dans Notre Projet

Dans ce tutoriel, comment installer et configurer DNS (Bind9) sur le serveur Ubuntu 22.04. DNS est un service très important pour l'Internet local et public en raison de son utilisation pour le résolveur de noms comme la conversion IP en nom.

- **Etape1- Vérifier la version du serveur :**

```
$ lsb_release -a && ip r
$ apt update && apt upgrade -y
```

```
abderrahmane@dns:~$ lsb_release -a && ip r
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 22.04.2 LTS
Release:        22.04
Codename:       jammy
default via 192.168.1.1 dev wlp2s0 proto dhcp metric 600
169.254.0.0/16 dev wlp2s0 scope link metric 1000
192.168.1.0/24 dev wlp2s0 proto kernel scope link src 192.168.1.5 metric 600
abderrahmane@dns:~$ sudo apt update && apt upgrade -y
```

- *Etape2- Installation Configure DNS (Bind9) service :*

```
$ sudo apt install bind9
$ sudo apt-get install bind9utils bind9-doc
```

```
abderrahmane@dns:~$ sudo apt install bind9
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
bind9 is already the newest version (1:9.18.12-0ubuntu0.22.04.1).
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
abderrahmane@dns:~$ sudo apt-get install bind9utils bind9-doc
```

```
abderrahmane@dns:~$ sudo nano /etc/resolv.conf
```

```
GNU nano 6.2                               /etc/resolv.conf
# This is /run/systemd/resolve/resolv.conf managed by man:systemd-resolved(8).
# Do not edit.
#
# This file might be symlinked as /etc/resolv.conf. If you're looking at
# /etc/resolv.conf and seeing this text, you have followed the symlink.
#
# This is a dynamic resolv.conf file for connecting local clients directly to
# all known uplink DNS servers. This file lists all configured search domains.
#
# Third party programs should typically not access this file directly, but only
# through the symlink at /etc/resolv.conf. To manage man:resolv.conf(5) in a
# different way, replace this symlink by a static file or a different symlink.
#
# See man:systemd-resolved.service(8) for details about the supported modes of
# operation for /etc/resolv.conf.

nameserver 192.168.1.5
nameserver 8.8.8.8
search hsup.uae.ma
```

Après on exécute la commande suivante pour accéder au bind :

```
$ cd /etc/bind/
```

```
abderrahmane@dns:~$ cd /etc/bind/
abderrahmane@dns:/etc/bind$ ls
bind.keys  db.127  db.empty  named.conf          named.conf.local    rndc.conf  zones
db.0       db.255  db.local   named.conf.default-zones  named.conf.options  rndc.key   zones.rfc1918
abderrahmane@dns:/etc/bind$
```

On commence par les deux premiers

```
abderrahmane@dns:/etc/bind$ sudo nano named.conf.options
abderrahmane@dns:/etc/bind$ sudo nano named.conf.local
```

Le fichier named.conf.options apres fichier named.conf.local

```
GNU nano 6.2                                     named.conf.options
acl "Trusted" {
    192.168.1.5;      # ns1 - can be set to localhost
};

options {
    directory "/var/cache/bind";

    // If there is a firewall between you and nameservers you want
    // to talk to, you may need to fix the firewall to allow multiple
    // ports to talk. See http://www.kb.cert.org/vuls/id/800113

    // If your ISP provided one or more IP addresses for stable
    // nameservers, you probably want to use them as forwarders.
    // Uncomment the following block, and insert the addresses replacing
    // the all-0's placeholder.

    recursion yes;
    allow-recursion { trusted; };
    listen-on { 192.168.1.5; };
    allow-transfer { none; };

    forwarders {
        8.8.8.8;
        8.8.4.4;
    };

    // forwarders {
    //     0.0.0.0;
    // };

    //========================================================================
    // If BIND logs error messages about the root key being expired,
    // you will need to update your keys. See https://www.isc.org/bind-keys
    //========================================================================
    dnssec-validation auto;

    listen-on-v6 { any; };
};
```

Copier ce code
mais avec vous

```

GNU nano 6.2                                     named.conf.local *

// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";


zone "hsup.uae.ma" {
    type master;
    file "/etc/bind/zones/db.fwd.hsup.uae.ma"; # zone file path
    allow-transfer { 192.168.1.5; };           # ns/www private IP address
};

zone "1.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/zones/db.rev.hsup.uae.ma"; # 192.168.0.0/16 subnet
    allow-transfer { 191.168.1.5; };           # ns/www private IP address
};

```

.1 Zone direct .2 Zone inverse

Zone direct (1) :

Le code que vous avez fourni est une configuration simplifiée pour une zone DNS dans le serveur BIND. Il indique que le serveur DNS est responsable de la zone "hsup.uae.ma" et que les enregistrements DNS de cette zone sont stockés dans le fichier "/etc/bind/hsup". Le serveur répondra aux requêtes DNS pour ce domaine en fournissant les enregistrements appropriés.

Zone inverse (2) :

Ce code configure une zone DNS inverse pour le sous-réseau 192.168.1.5/16. Le serveur DNS BIND est défini en tant qu'autorité principale pour cette zone, et les enregistrements sont stockés dans le fichier "/etc/bind/zones/db.rev.hsup.uae.ma". Seule l'adresse IP 192.168.1.5 est autorisée à effectuer des transferts de zone.

Maintenant on va créer un fichier répertoire qui contient les zones :

```
abderrahmane@dns:/etc/bind$ mkdir zones
```

Dans la Zone direct :

```
abderrahmane@dns:/etc/bind/zones$ sudo nano db.fwd.hsup.uae.ma
```

Insérez ci-dessous tout le texte dans ce fichier (mais avec votre données).

```
GNU nano 6.2                                         db.fwd.hsup.uae.ma
;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     ns.hsup.uae.ma. root.ns.hsup.uae.ma. (
                      2           ; Serial
                      604800      ; Refresh
                      86400       ; Retry
                     2419200     ; Expire
                     604800 )    ; Negative Cache TTL
;
; name servers - NS records
      IN      NS      ns.hsup.uae.ma.

; name servers - A records
ns.hsup.uae.ma.        IN      A      192.168.1.5
www.hsup.uae.ma.       IN      A      192.168.1.5
;
```

Créez ensuite un fichier de zone inversée.

```
abderrahmane@dns:/etc/bind/zones$ sudo nano db.rev.hsup.uae.ma
```

Insérez ci-dessous tout le texte dans ce fichier.

```
GNU nano 6.2                                         db.rev.hsup.uae.ma
;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     ns.hsup.uae.ma. root.ns.hsup.uae.ma. (
                      5
                      604800
                      86400
                     2419200
                     604800 )

; name servers
      IN      NS      ns.hsup.uae.ma.

5       IN      PTR     ns.hsup.uae.ma.
5       IN      PTR     www.hsup.uae.ma.
```

Puis retour d'un répertoire

```
$ cd
```

~

Ensuite, affichez rndc.key info à l'aide de la commande cat.

```
abderrahmane@dns:/etc/bind/zones$ cd ..
abderrahmane@dns:/etc/bind$ sudo cat rndc.key
key "rndc-key" {
    algorithm hmac-sha256;
    secret "7Wn+ysaFOKe5USAagsQK4aYUPOFydeYrObo0tlMfpSY=";
};
```

Créez maintenant le fichier rndc.conf.

\$ sudo nano rndc.conf

Insérez ensuite ci-dessous tout le texte dans ce fichier.

```
GNU nano 6.2                                     rndc.conf
key "rndc-key" {
    algorithm hmac-sha256;
    secret "7Wn+ysaFOKe5USAagsQK4aYUPOFydeYrObo0tlMfpSY=";
};

options {
    default-key "rndc-key";
    default-server 127.0.0.1;
    default-port 953;
};
```

- Etape3- Autorisation d'installation et redémarrage du service bind9.**

Exécuter les commandes suivantes, il faut après redémarrage être Active.

```
abderrahmane@dns:/etc/bind$ sudo chown root:bind -R /etc/bind
abderrahmane@dns:/etc/bind$ sudo chown bind:bind -R /etc/bind/rndc.conf
abderrahmane@dns:/etc/bind$ sudo service bind9 restart
abderrahmane@dns:/etc/bind$ sudo service bind9 status
● named.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/named.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2023-05-17 15:35:35 +01; 8s ago
     Docs: man:named(8)
  Process: 14837 ExecStart=/usr/sbin/named $OPTIONS (code=exited, status=0/SUCCESS)
 Main PID: 14838 (named)
    Tasks: 10 (limit: 9277)
   Memory: 7.7M
      CPU: 147ms
     CGroup: /system.slice/named.service
             └─14838 /usr/sbin/named -u bind

May 17 15:35:35 dns.hsup.uae.ma named[14838]: network unreachable resolving './NS/IN': 2001:7fe::53#>
May 17 15:35:35 dns.hsup.uae.ma named[14838]: network unreachable resolving './NS/IN': 2001:7fd::1#53
May 17 15:35:35 dns.hsup.uae.ma named[14838]: network unreachable resolving './NS/IN': 2001:503:ba3e>
May 17 15:35:35 dns.hsup.uae.ma named[14838]: network unreachable resolving './NS/IN': 2001:dc3::35#>
May 17 15:35:35 dns.hsup.uae.ma named[14838]: zone 255.in-addr.arpa/IN: loaded serial 1
May 17 15:35:35 dns.hsup.uae.ma named[14838]: all zones loaded
May 17 15:35:35 dns.hsup.uae.ma named[14838]: running
May 17 15:35:35 dns.hsup.uae.ma systemd[1]: Started BIND Domain Name Server.
May 17 15:35:35 dns.hsup.uae.ma named[14838]: managed-keys-zone: Key 20326 for zone . is now trusted
May 17 15:35:35 dns.hsup.uae.ma named[14838]: resolver priming query complete: success
Lines 1-22/22 (END)
```

4.5.3. *Etape4- Tester service bind9.*

Maintenant, testez le service bind9 à partir du serveur.

```
abderrahmane@dns:/etc/bind$ cd
abderrahmane@dns:$ dig ns.hsup.uae.ma

; <>> DiG 9.18.12-0ubuntu0.22.04.1-Ubuntu <>> ns.hsup.uae.ma
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 1686
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 1232
;; COOKIE: e61f343a8f160ad5010000006464e67554de810532720c57 (good)
;; QUESTION SECTION:
;ns.hsup.uae.ma.           IN      A

;; ANSWER SECTION:
ns.hsup.uae.ma.      604800  IN      A      192.168.1.5

;; Query time: 0 msec
;; SERVER: 192.168.1.5#53(192.168.1.5) (UDP)
;; WHEN: Wed May 17 15:36:37 +01 2023
;; MSG SIZE rcvd: 87
```

```
abderrahmane@dns:$ dig www.hsup.uae.ma

; <>> DiG 9.18.12-0ubuntu0.22.04.1-Ubuntu <>> www.hsup.uae.ma
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 36316
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 1232
;; COOKIE: e2af827fc4faae3d010000006464e67a6f02cef76f7da193 (good)
;; QUESTION SECTION:
;www.hsup.uae.ma.        IN      A

;; ANSWER SECTION:
www.hsup.uae.ma.      604800  IN      A      192.168.1.5

;; Query time: 0 msec
;; SERVER: 192.168.1.5#53(192.168.1.5) (UDP)
;; WHEN: Wed May 17 15:36:42 +01 2023
;; MSG SIZE rcvd: 88
```

Vérifiez le résolveur PTR en utilisant la commande ci-dessous.

```
abderrahmane@dns:~$ dig -x 192.168.1.5
; <>> DiG 9.18.12-0ubuntu0.22.04.1-Ubuntu <>> -x 192.168.1.5
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 31497
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1
;;
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 1232
;; COOKIE: 62146f08afaa4f0b010000006464e6c03d4fe12600661596 (good)
;; QUESTION SECTION:
;5.1.168.192.in-addr.arpa.      IN      PTR
;
; ANSWER SECTION:
5.1.168.192.in-addr.arpa. 604800 IN      PTR      www.hsup.uae.ma.
5.1.168.192.in-addr.arpa. 604800 IN      PTR      ns.hsup.uae.ma.

;; Query time: 0 msec
;; SERVER: 192.168.1.5#53(192.168.1.5) (UDP)
;; WHEN: Wed May 17 15:37:52 +01 2023
;; MSG SIZE  rcvd: 127
```

Vérifier le test de nom.

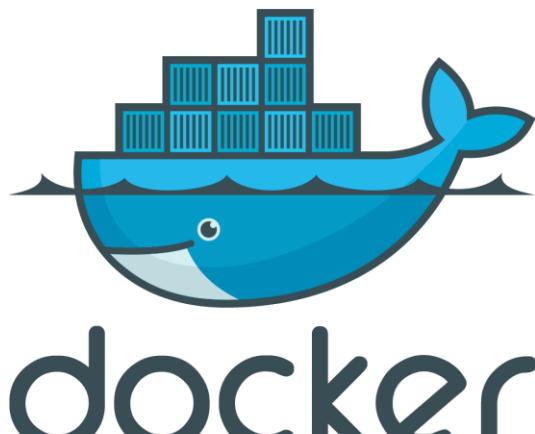
```
abderrahmane@dns:~$ nslookup www
Server:      192.168.1.5
Address:     192.168.1.5#53

Name:   www.hsup.uae.ma
Address: 192.168.1.5
```

Vérifier le test ip

```
abderrahmane@dns:~$ nslookup 192.168.1.5
5.1.168.192.in-addr.arpa      name = ns.hsup.uae.ma.
5.1.168.192.in-addr.arpa      name = www.hsup.uae.ma.
```

4.6. DOCKER



• Introduction

- ◆ Docker est une application qui simplifie le processus de gestion des processus d'application dans des conteneurs. Les conteneurs vous permettent d'exécuter vos applications dans des processus isolés des ressources. Ils sont similaires aux machines virtuelles, mais les conteneurs sont plus portables, plus respectueux des ressources et plus dépendants du système d'exploitation hôte.

4.6.1. Étape 1: Installation de "Docker"



Bash Terminal

```
> sudo apt update // mettre à jour les paquets  
> sudo apt install apt-transport-https ca-certificates curl software-properties-common
```

Ensuite, installez quelques packages prérequis qui permettent à apt d'utiliser des packages via HTTPS :

- Ajoutez ensuite la clé GPG du référentiel Docker officiel à notre système :



Bash Terminal (Ubuntu)

```
> curl -fsSL  
https://download.docker.com/linux/ubuntu/gpg | sudo  
gpg --dearmor -o /usr/share/keyrings/docker-archive-  
keyring.gpg
```

Ajoutez le dépôt Docker aux sources APT :



Bash Terminal (Ubuntu)

```
> echo "deb [arch=$(dpkg --print-architecture) signed-  
by=/usr/share/keyrings/docker-archive-keyring.gpg]  
https://download.docker.com/linux/ubuntu $(lsb_release -  
cs) stable" | sudo tee  
/etc/apt/sources.list.d/docker.list > /dev/null
```

- Nous mettons à jour à nouveau notre liste de packages existante pour que l'ajout soit reconnu :



Bash Terminal (Ubuntu)

```
> sudo apt update // mettre à jour les packets  
> apt-cache policy docker-ce // installer à partir du  
Repo Docker  
  
> sudo apt install docker-ce // Enfin, installez  
~48 ~  
Docker
```

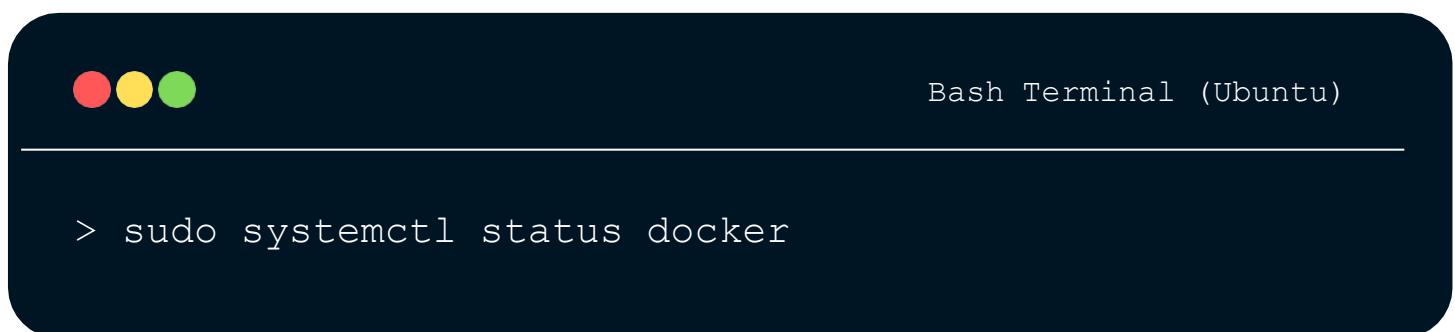


```

sudo apt install docker-ce
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3
  libaprutil1-ldap
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  containerd.io docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras
  docker-compose-plugin libslirp0 pigz slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-rootless-extras
  docker-compose-plugin libslirp0 pigz slirp4netns
0 upgraded, 9 newly installed, 0 to remove and 158 not upgraded.
Need to get 110 MB of archives.

```

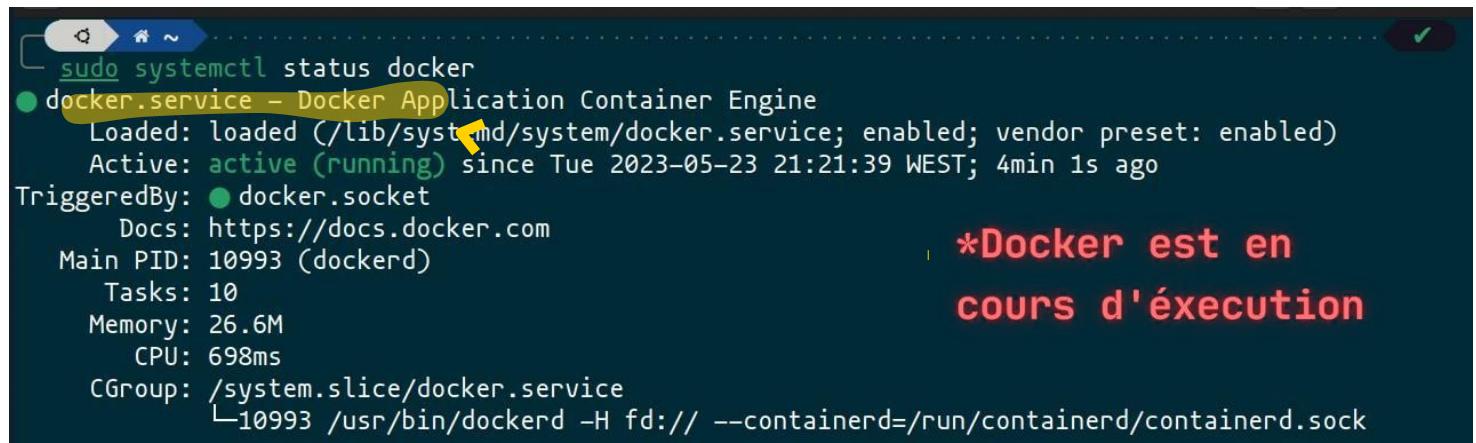
- ◆ Docker devrait maintenant être installé, le daemon démarré et le processus activé pour démarrer au démarrage.
On vérifie qu'il est en cours d'exécution :



```

> sudo systemctl status docker

```



```

sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2023-05-23 21:21:39 WEST; 4min 1s ago
     Tasks: 10
    Memory: 26.6M
      CPU: 698ms
     CGroup: /system.slice/docker.service
             └─10993 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

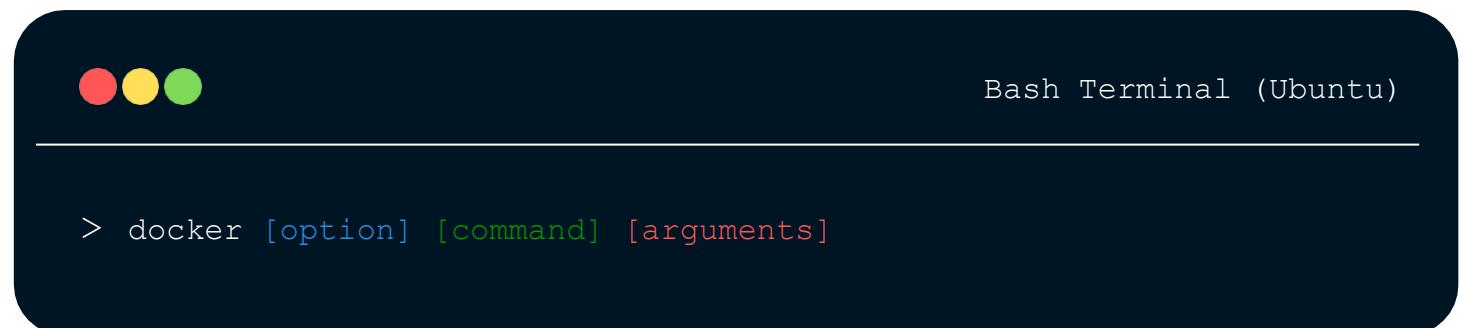
```

*Docker est en cours d'exécution

- ◆ L'installation de Docker nous donne désormais non seulement le service Docker (daemon), mais également l'utilitaire de ligne de commande Docker ou le client Docker.

4.6.2. Étape 2: Utilisation de la commande Docker

- Utiliser docker consiste à lui passer une chaîne d'options et de commandes suivies d'arguments. La syntaxe prend cette forme :



```

└── docker
Usage: docker [OPTIONS] COMMAND
A self-sufficient runtime for containers

Common Commands:
  run      Create and run a new container from an image
  exec    Execute a command in a running container
  ps       List containers
  build   Build an image from a Dockerfile
  pull    Download an image from a registry
  push    Upload an image to a registry
  images  List images
  login   Log in to a registry
  logout  Log out from a registry
  search  Search Docker Hub for images
  version Show the Docker version information
  info    Display system-wide information

```

4.6.3. Étape 3: Travailler avec des images Docker

- Les conteneurs Docker sont construits à partir d'images Docker. Par défaut, Docker extrait ces images de Docker Hub, un registre Docker géré par Docker, la société à l'origine du projet Docker. N'importe qui peut héberger ses images Docker sur Docker Hub, donc la plupart des applications et des distributions Linux dont vous aurez besoin auront des images hébergées là-bas.
- Pour vérifier si nous pouvons accéder et télécharger des images depuis Docker Hub, nous utilisons :



Bash Terminal

```
> docker run hello-
```

- La sortie indiquera que Docker fonctionne correctement :

```
docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
719385e32844: Pull complete
Digest: sha256:fc6cf906cbfa013e80938cdf0bb199fdbbb86d6e3e013783e5a766f50f5dbce0
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.
```

L'image `hello_world` :

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
606748c93abd	hello-world	"/hello"	52 minutes ago	Exited (0) 52 minutes ago		funny_wilson

- Exécution d'un conteneur Docker



Bash Terminal (Ubuntu)

```
> docker pull ubuntu:latest
```

```
> docker run -it ubuntu
```

- Par exemple, exécutons un conteneur en utilisant la dernière image d'Ubuntu. La combinaison des commutateurs `-i` et `-t` vous donne un accès Shell interactif au conteneur :

XTRADH

```
└─ docker pull ubuntu:latest
    latest: Pulling from library/ubuntu
    dbf6a9befcde: Pull complete
    Digest: sha256:dfd64a3b4296d8c9b62aa3309984f8620b98d87e47492599ee20739e8eb54fbf
    Status: Downloaded newer image for ubuntu:latest
    docker.io/library/ubuntu:latest
```

```
└─ docker run -it ubuntu
    root@77bb36931a86:/# ls
    bin  dev  home  lib32  libx32  mnt  proc  run  srv  tmp  var
    boot etc  lib   lib64  media  opt  root  sbin  sys  usr
    root@77bb36931a86:/#
```

• Gestion des conteneurs Docker

- ◆ Pour afficher tous les conteneurs - actifs et inactifs, exécutez docker ps avec -a :



Bash Terminal (Ubuntu)

```
> docker ps -a
```



Bash Terminal (Ubuntu)

```
> docker ps -l
```

```
└─ docker ps -a
    CONTAINER ID  IMAGE      COMMAND      CREATED      STATUS      PORTS      NAMES
    2e727b053eb2  ubuntu      "/bin/bash"  17 seconds ago  Up 16 seconds
    77bb36931a86  ubuntu      "/bin/bash"  7 minutes ago  Exited (0) 2 minutes ago
    ...
```

- Pour démarrer un conteneur arrêté, utilisez docker start, suivi de l'ID du conteneur ou du nom du conteneur. Commençons le conteneur basé sur Ubuntu avec l'ID 2e727b053eb2 :



Bash Terminal (Ubuntu)

```
> docker start 2e727b053eb2
```

Pour arrêter un conteneur en cours d'exécution, utilisez docker stop, suivie de l'ID ou du nom du conteneur. Cette fois, nous utiliserons le nom que Docker a attribué au conteneur, qui est silly_pare



Bash Terminal (Ubuntu)

```
> docker stop silly_pare
```

```
docker stop silly_pare
silly_pare
```

- Une fois que nous avons décidé que nous n'avions plus besoin d'un conteneur, nous pouvons le supprimer avec la commande docker rm, en utilisant à nouveau soit l'ID du conteneur, soit le nom.



Bash Terminal (Ubuntu)

```
> docker rm silly_pare
```

~ 53 ~

4.6.4. Étape 4: Dockerisation de notre projet

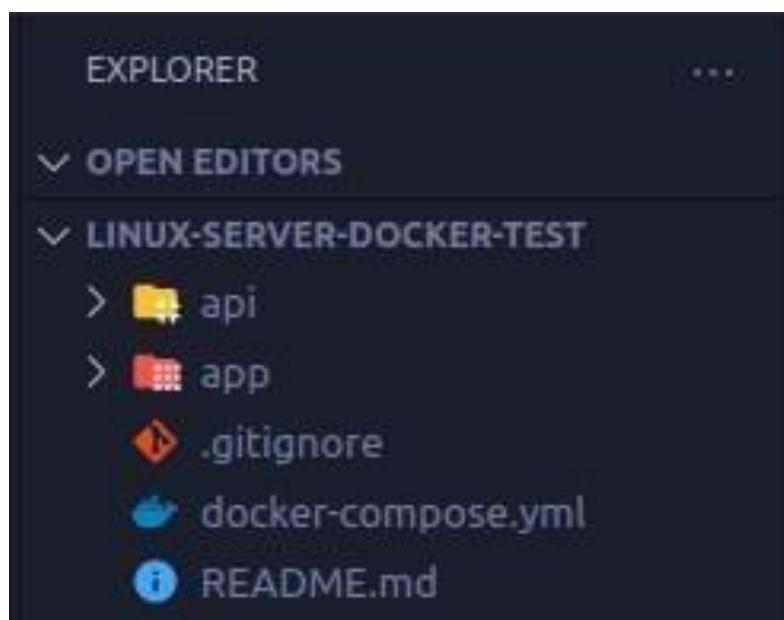
- ***Configuration du conteneur postgres***

-> Pourquoi utiliser Postgres avec docker pour le développement local?

Il existe de nombreuses bonnes raisons d'utiliser n'importe quelle base de données comme Postgres avec Docker pour le développement local, voici quelques bonnes raisons :

1. L'utilisation de plusieurs versions de PostgreSQL selon le projet ou tout autre besoin est très simple.
2. Généralement, avec docker, s'il s'exécute sur votre machine, il fonctionnera pour votre ami, sur un environnement intermédiaire et un environnement de production, étant donné que la compatibilité des versions est maintenue.
3. Lorsqu'un nouveau membre de l'équipe se joint, le nouveau membre peut démarrer en quelques heures, il ne faut pas des jours pour être productif.

- ◆ PostgreSQL with docker-compose



```

docker-compose.yml ✘
  docker-compose.yml
  31
  32 db:
  33   container_name: db
  34   image: postgres:13
  35   restart: always
  36   hostname: db
  37   ports:
  38     - "5432:5432"
  39
  40   environment:
  41     - POSTGRES_USER=postgres
  42     - POSTGRES_PASSWORD=postgres
  43     - POSTGRES_DB=postgres
  44     - POSTGRES_HOST=db
  45
  46   volumes:
  47     - pgdata:/var/lib/postgresql/data
  48
  49
  50 volumes:
  51   pgdata: {}

```

- ◆ container_name: db - le nom du conteneur.
- ◆ image: postgres:13 - Nous utilisons la version 13 de postgres. restart: always - toujours redémarrer le conteneur en cas d'erreur.hostname: db - hostname associé à la base de données
- ◆ ports: 5432:5432 - le port sur lequel la base de données écoutera les événements
- ◆ environment:
 - POSTGRES_USER=postgres
 - POSTGRES_PASSWORD=postgres
 - POSTGRES_DB=postgres
 - POSTGRES_HOST=db

Les paramètres requis pour connecter la base de données postgres, y compris le nom d'utilisateur, le mot de passe et le nom de la base de données.

- Pour démarrer la base de données PostgreSQL à l'aide de ce fichier docker-compose.yml, vous pouvez exécuter la commande suivante :



Bash Terminal (Ubuntu)

```
> docker-compose up -d
```



```
~/Downloads/Linux-Server-docker-test/test
docker-compose up -d
Creating network "test_default" with the default driver
Creating volume "test_pgdata" with default driver
Creating db ... done
```



```
~/Downloads/Linux-Server-docker-test/test
docker ps -l
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
4a0d413d0f34 postgres:13 "docker-entrypoint.s..." About a minute ago Up About a minute 0.0.0.0:5432->5432/tcp, :::5432->5432/tcp db
```

on se connecte à la base de données via la commande : docker exec -it db psql -U postgres

- Nous créons une table à des fins de test appelée "accounts"



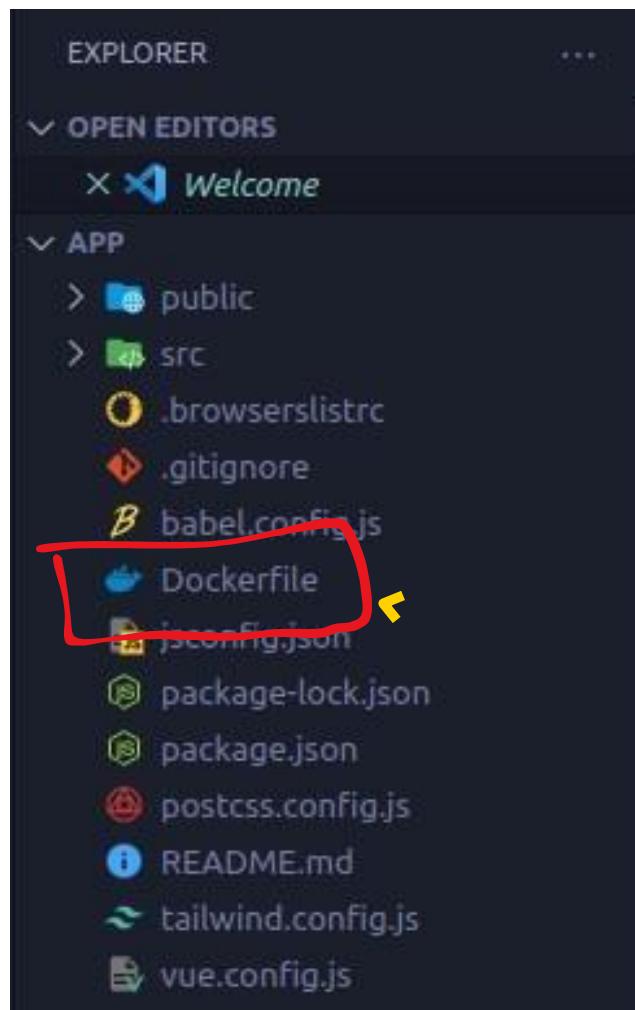
```
~/Downloads/Linux-Server-docker-test/test
docker exec -it db psql -U postgres
psql (13.11 (Debian 13.11-1.pgdg110+1))
Type "help" for help.

postgres=# CREATE TABLE accounts (
    user_id serial PRIMARY KEY,
    username VARCHAR ( 50 ) UNIQUE NOT NULL,
    password VARCHAR ( 50 ) NOT NULL,
    email VARCHAR ( 255 ) UNIQUE NOT NULL,
    created_on TIMESTAMP NOT NULL,
    last_login TIMESTAMP
);
CREATE TABLE
postgres=#
postgres=# \dt
      List of relations
 Schema |   Name    | Type  | Owner
-----+-----+-----+
 public | accounts | table | postgres
(1 row)

postgres=#
```

- **Configuration du conteneur vueJS**

- Commençons par créer un fichier Dockerfile dans le dossier racine de notre projet



Le fichier Dockerfile
de Configuration de
vueJS

- Le fichier Dockerfile:

```

Dockerfile X
Dockerfile > ...
1  FROM node:lts-alpine
2
3
4  RUN npm install -g http-server
5
6
7  WORKDIR /app
8
9
10 COPY package*.json ./
11
12
13 RUN npm install
14
15
16 COPY . .
17
18
19 RUN npm run build
20
21
22 CMD [ "npm", "start" ]
23

```

- ◆ FROM node:lts-alpine
 - Choisissez l'image sur laquelle Node est déjà installé
 - RUN npm install -g http-server
- ◆
 - installer un serveur http simple pour servir du contenu statique
 - WORKDIR /app
 - créer du dossier 'app' le répertoire de travail actuel
 - COPY package*.json ./
- ◆
 - copier à la fois 'package.json' et 'package-lock.json' (si disponible)
- ◆
- ◆ RUN npm install
 - installer les dépendances du projet
 - COPY . .
 - copier les fichiers et dossiers du projet dans le répertoire de travail actuel (c'est-à-dire le dossier 'app')
- ◆ RUN npm run build
 - créer une application pour la production avec minification
 - CMD ["npm", "start"]
- ◆
 - starting npm

```

1  docker-compose.yml ✘
2
3  docker-compose.yml
4    version: "3.9"
5
6  services:
7    web:
8      image: test_website
9      container_name: test_website
10     build:
11       context: app
12       dockerfile: ./Dockerfile
13     ports:
14       - 8080:8080
15     volumes:
16       - ./app:/app
17     command: npm run serve

```

container_name: test_website - le nom du conteneur. image: test_website - Nous créons une image de l'app.build:

- - context: app
- -dockerfile: ./Dockerfile

- L'emplacement du fichier Dockerfile de configuration et le context de l'app

- ports: 8080:8080 - le port sur lequel l'app vueJS écoutera les événements
- volumes:
 - ./app:/app

- Il monte un volume sur ./app sur la machine hôte.command:

- - npm run serve
- la commande pour lancer l'app vuejs sur l'address "0.0.0.0:8080"

Pour

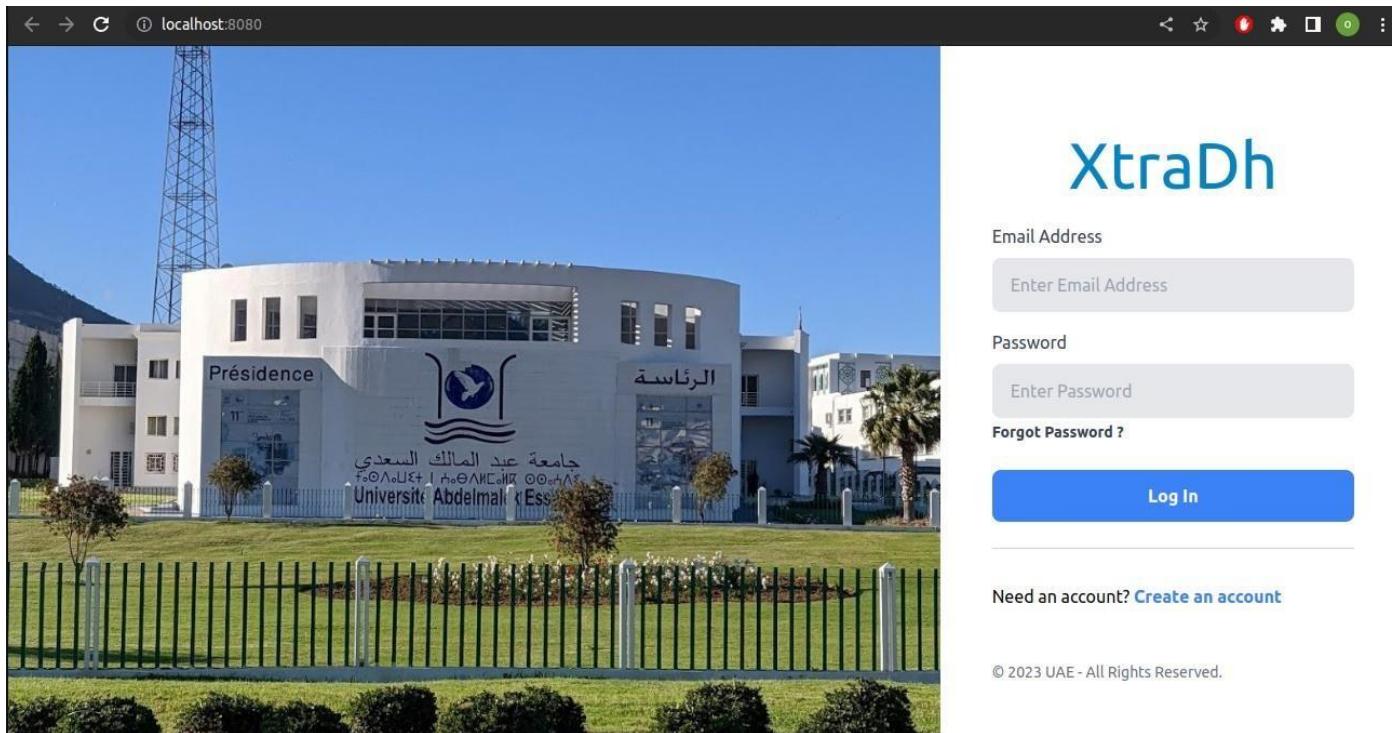
XTRADH

démarrer

```
↳ ~ ~/Downloads/Linux-Server-docker-test
↳ docker-compose up --build web
Creating network "linux-server-docker-test_default" with the default driver
Creating volume "linux-server-docker-test_pgdata" with default driver
Building web
[+] Building 8.6s (4/11)
=> [internal] load .dockerignore          0.2s
=> => transferring context: 2B           0.0s
=> [internal] load build definition from Dockerfile 0.2s
=> => transferring dockerfile: 210B       0.0s
=> [internal] load metadata for docker.io/library/node:lts-alpine 2.5s
=> [1/7] FROM docker.io/library/node:lts-alpine@sha256:1ccc70acda680aa4ba47f53e7c40b2d4d6892de74817128e0662d 5.8s
=> => resolve docker.io/library/node:lts-alpine@sha256:1ccc70acda680aa4ba47f53e7c40b2d4d6892de74817128e0662d 0.1s
=> => sha256:1ccc70acda680aa4ba47f53e7c40b2d4d6892de74817128e0662d32647dd7f4d 1.43kB / 1.43kB 0.0s
=> => sha256:44aaf1ccc80eaed6572a0f2ef7d6b5a2982d54481e4255480041ac92221e2f11 1.16kB / 1.16kB 0.0s
=> => sha256:6f44d13dd2586511d630d7708da1eaac09c05693d9e3e06f3f72206927d3201d 6.49kB / 6.49kB 0.0s
=> => sha256:f56be85fc22e46face30e2c3de3f7fe7c15f8fd7c4e5add29d7f64b87abdaa09 3.37MB / 3.37MB 3.5s
=> => sha256:931b0e865bc24d4d3f7032c4a9b98b4790ee7bb8dc0d2fd02e15e163d806034 1.05MB / 47.48MB 5.7s
=> => sha256:60542df8b663b3c876b9fd0ad8459ac628a50bd8684149e54c3dff2e44397f03 2.35MB / 2.35MB 5.6s
=> => extracting sha256:f56be85fc22e46face30e2c3de3f7fe7c15f8fd7c4e5add29d7f64b87abdaa09 0.1s
↳ docker ps -l
CONTAINER ID   IMAGE      COMMAND      CREATED     STATUS      PORTS     NAMES
ee1678910c35   test_website   "docker-entrypoint.s..."  9 minutes ago   Exited (127) 4 minutes ago
ebsite
```

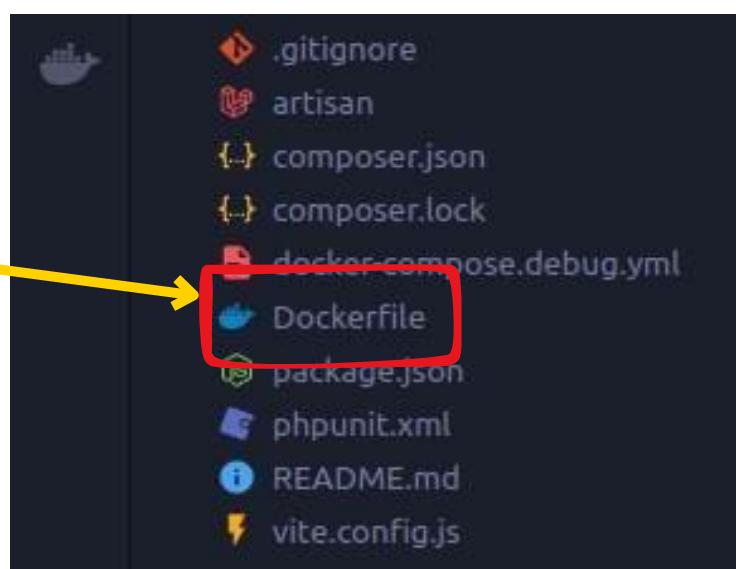
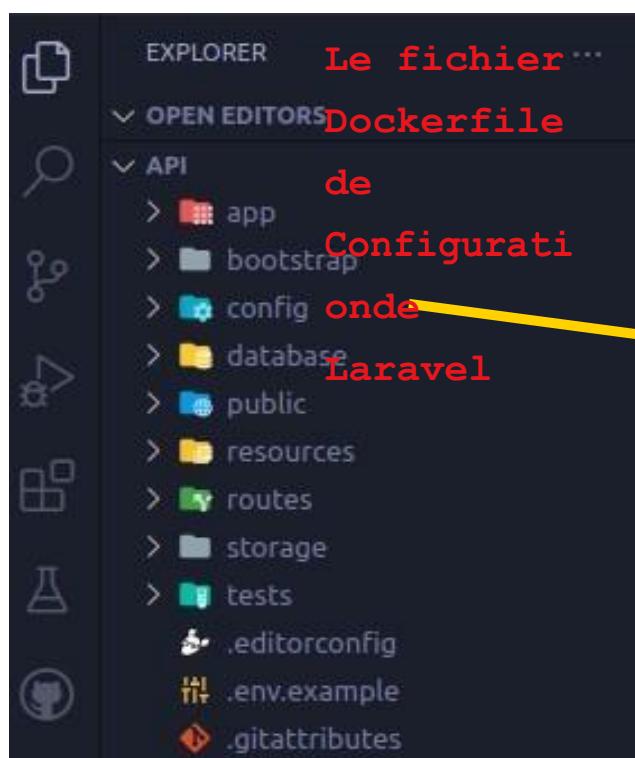
l'application VueJS à l'aide de ce fichier docker-compose.yml, vous pouvez exécuter la commande suivante :

On vérifie sur le navigateur: <http://localhost:8080/>



2. Configuration du conteneur Laravel API

- Commençons par créer un fichier Dockerfile dans le dossier racine de notre projet



- Le fichier Dockerfile:

```
Dockerfile X
Dockerfile > FROM
1  FROM php:8.1
2
3  RUN apt-get update && apt-get install -y \
4      libpq-dev \
5      && docker-php-ext-install pdo pdo_pgsql
6
7  RUN curl -sS https://getcomposer.org/installer | php -- --install-dir=/usr/local/bin --filename=composer
8
9  WORKDIR /var/www/html
10
11 COPY . /var/www/html
12
13 ENV COMPOSER_ALLOW_SUPERUSER=1
14
15 RUN composer install
16
17 RUN chown -R www-data:www-data /var/www/html/storage /var/www/html/bootstrap/cache
18
19 CMD php artisan migrate && php artisan serve --host=0.0.0.0 --port=8000
```

- FROM php:8.1
 - Nous utiliserons PHP version 8.1 comme image de base
 - RUN apt-get update && apt-get install -y \
 - libpq-dev \
 - && docker-php-ext-install pdo pdo_pgsql
 - installer les dépendances nécessaires pour se connecter à la base de données comme le pilote db etc...
- RUN curl -sS https://getcomposer.org/installer | php----- install-dir=/usr/local/bin --filename=composer
 - installer composer dans le conteneur pour exécuter l'application laravel
- WORKDIR /var/www/html
 - définir le répertoire /var/www/html comme une répertoire racine de l'application laravel.
- COPY . /var/www/html : copier les fichiers et dossiers du projet dans le répertoire du travaille.

- ◆ ENV


```
COMPOSER_ALLOW_SUPERUSER=1
  - accorder des autorisations au composer pour exécuter l'application laravel
```
- ◆ RUN composer install


```
- Finir l'installation de composer
```
- ◆ RUN chown -R www-data:www-data /var/www/html/storage


```
/var/www/html/bootstrap/cache
  - accorder les permissions d'exécuter les sous-dossiers du répertoire de l'application
```
- ◆ CMD php artisan migrate && php artisan serve --host=0.0.0.0 --port=8000


```
- migration des fichiers de configuration de la base de données avant de démarrer l'application sur l'hôte local
:8000
```
- ◆ Le fichier docker-compose.yml:

```
api:
  container_name: laravel_test_api
  image: laravel_test_api
  build:
    context: api
    dockerfile: ./Dockerfile

  ports:
    - "8000:8000"
  env_file:
    - api/.env
  volumes:
    - ./api:/var/www/html
  depends_on:
    - db
```

- ◆ container_name: laravel_test_api - le nom du conteneur. image: laravel_test_api - Nous créons une image de l'app.

- build:
 - context: api
 - dockerfile: ./Dockerfile
 - L'emplacement du fichier Dockerfile de configuration et le context de l'app
- ports: 8000:8000 - le port sur lequel l'app Laravel écoutera les événements
- volumes:
 - ./api:/var/www/html
 - Il monte un volume sur ./api sur la machine hôte.
- env_file:
 - api/.env
- L'emplacement du fichier .env qui contient les paramètres de connexion à la base de données.
- depends:
 - db
 - L'API du conteneur dépend de la fonction du conteneur de la base de données et ne démarrera que si le conteneur bd est opérationnel et fonctionne correctement.
- Pour démarrer l'application Laravel à l'aide de ce fichier docker-compose.yml, vous pouvez exécuter la commande suivante :



Bash Terminal (Ubuntu)

```
> docker-compose up --build api
```

```
docker ps
CONTAINER ID   IMAGE          COMMAND       CREATED      STATUS        PORTS
              NAMES
2e6cc8dcdf02  laravel_test_api  "docker-php-entrypoi..."  55 seconds ago  Up 54 seconds  0.0.0.0:8000->8000/tcp,
:::8000->8000/tcp    laravel_test_api
28b22e1df634  postgres:13     "docker-entrypoint.s..."  15 minutes ago  Up 5 minutes   0.0.0.0:5432->5432/tcp,
:::5432->5432/tcp    db
```

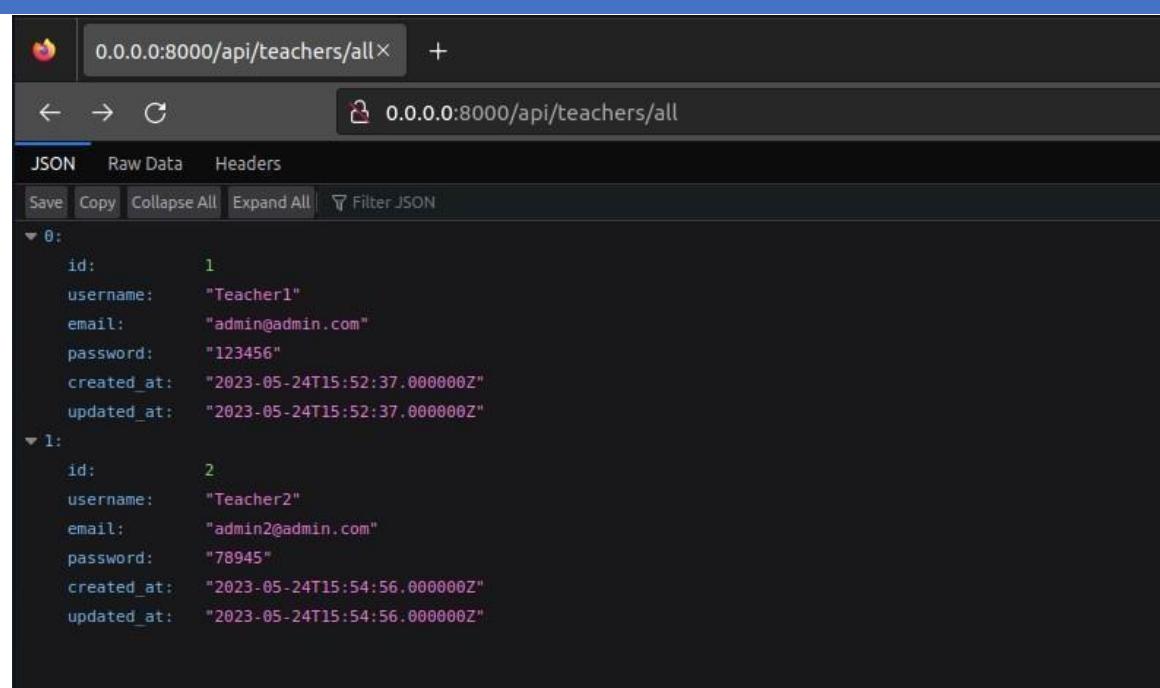
XTRADH

```
docker-compose up --build api
Building api
[+] Building 16.9s (4/11)
=> [internal] load build definition from Dockerfile          0.1s
=> => transferring dockerfile: 518B                         0.0s
=> [internal] load .dockerignore                            0.1s
=> => transferring context: 2B                           0.0s
=> [internal] load metadata for docker.io/library/php:8.1   2.6s
=> [1/7] FROM docker.io/library/php:8.1@sha256:1e2526b7fa73a1da7ebffa6773aee9a79c6e056031 14.1s
=> => resolve docker.io/library/php:8.1@sha256:1e2526b7fa73a1da7ebffa6773aee9a79c6e0560311 0.1s
=> => sha256:96844aaaa11691e3782ca2dceabd00e56cf7222c737b12806f93a0b50c26 2.20kB / 2.20kB 0.0s
=> => sha256:593ded604f117b0b52730cd26777b233477cae2d63a0f709e920d2818afdc 9.76kB / 9.76kB 0.0s
=> => sha256:662d8f2fcdb9ff5b9497538bdbb929df1cc5f1d3?1a15f5ea427ade8ed4dc08f 224B / 224B 0.3s
=> => sha256:1e2526b7fa73a1da7ebffa6773aee9a79c6e05603110b496bb8737651e332 1.86kB / 1.86kB 0.0s
=> => sha256:78fe0ef5ed7711aab5b14ba79dee1d8f750f1cb661645280d76336db601 8.39MB / 91.64MB 13.9s

=> => naming to docker.io/library/laravel_test_api           0.0s
db is up-to-date
Recreating laravel_test_api ... done
Attaching to laravel_test_api
laravel_test_api |     INFO  Preparing database.
laravel_test_api |     INFO  Creating migration table ..... 15ms DONE
laravel_test_api |     INFO  Running migrations.
laravel_test_api |     INFO  2014_10_12_000000_create_users_table ..... 28ms DONE
laravel_test_api |     INFO  2014_10_12_100000_create_password_reset_tokens_table ..... 27ms DONE
laravel_test_api |     INFO  2019_08_19_000000_create_failed_jobs_table ..... 43ms DONE
laravel_test_api |     INFO  2019_12_14_000001_create_personal_access_tokens_table ..... 45ms DONE
laravel_test_api |     INFO  2023_05_13_135305_create_teachers_table ..... 27ms DONE
laravel_test_api |
laravel_test_api |     INFO  Server running on [http://0.0.0.0:8000].
laravel_test_api |     Press Ctrl+C to stop the server
laravel_test_api |
```

- On vérifie sur le navigateur: <http://0.0.0.0:8000/>

XTRADH



The screenshot shows a browser window with the URL `0.0.0.0:8000/api/teachers/all`. The page displays a JSON response containing two teacher objects, indexed 0 and 1.

```
0:
  id:      1
  username: "Teacher1"
  email:   "admin@admin.com"
  password: "123456"
  created_at: "2023-05-24T15:52:37.000000Z"
  updated_at: "2023-05-24T15:52:37.000000Z"

1:
  id:      2
  username: "Teacher2"
  email:   "admin2@admin.com"
  password: "78945"
  created_at: "2023-05-24T15:54:56.000000Z"
  updated_at: "2023-05-24T15:54:56.000000Z"
```

4.6.4.1.

Configuration du conteneur NGINX

```
# Use Nginx as the server
FROM nginx:latest

# Remove the default Nginx configuration
RUN rm /etc/nginx/conf.d/default.conf

# Copy your custom Nginx configuration file
COPY nginx/myconfig.conf /etc/nginx/conf.d/

# Copy the built files from the previous stage
COPY --from=build-stage /app/dist /usr/share/nginx/html

# Expose port 80
EXPOSE 80

# Start Nginx
CMD ["nginx", "-g", "daemon off;"]
```

- ◆ FROM nginx:latest
 - extraire l'image de la dernière version de nginx à partir du hub docker
 - RUN rm /etc/nginx/conf.d/default.conf
- Supprimer le fichier de configuration par défaut
 - COPY nginx/myconfig.conf /etc/nginx/conf.d/
- ◆ Copiez notre fichier de configuration Nginx personnalisé
 - COPY --from=build-stage /app/dist /usr/share/nginx/html
- ◆ Copiez les fichiers de construction à partir de l'application vue
 - js
 - EXPOSE 80
- Utiliser le port 80
 - ◆ CMD ["nginx", "-g", "daemon off;"]
- lancer NGINX

- Le fichier docker-compose.yml:

```

14    nginx:
15      image: nginx:latest
16      ports:
17        - 80:80
18        - 443:443
19      volumes:
20        - ./nginx:/etc/nginx/conf.d
21        - ./ssl:/var/www/ssl
22      depends_on:
23        - app

```

- image: nginx:latest - la dernière image de nginxports:
 - 80:80
 - 443:443
 - Utiliser 80 pour HTTP et 443 pour HTTPS
- ./nginx:/etc/nginx/conf.d
- ./ssl:/var/www/ssl
- Copier le fichier de configuration et la certificat ssldepends_on:
 - app
- Le conteneur nginx dépend de l'application vuejs pour commencer.
- Pour démarrer l'applucation VueJS et NGINX à l'aide de ce fichier docker-compose.yml, vous pouvez exécuter la commande suivante :

```

q  ~/docker
└── docker-compose up -d
Starting docker_app_1 ... done
Starting docker_nginx_1 ... done
~ 66 ~

q  ~/docker
└── docker-compose up
Starting docker_app_1 ... done
docker_nginx_1 is up-to-date

```

On vérifie sur le navigateur : <https://hsup.uae.ma>

```
➜  ~ docker ps
CONTAINER ID   IMAGE      COMMAND   CREATED          STATUS          PORTS
0377e945752c   nginx:latest "/docker-entrypoint..."  55 seconds ago   Up 11 seconds   0.0.0.0:80
->80/tcp, ::80->80/tcp, 0.0.0.0:443->443/tcp, ::443->443/tcp   docker_nginx_1

```

XtraDh

Email Address

Password

[Forgot Password ?](#)

Log In

Need an account? [Create an account](#)

© 2023 UAE - All Rights Reserved.



Configuration Routeur

- Nous allons configurer le routeur (un routeur TP Link dans cecas) pour modifier les paramètres DNS afin de rendre notre nom de domaine "hsup.uae.ma" accessible à tous sur le réseau local
- **Etape1-Connecter à la page de configuration du routeur :**

L'address de la passerelle par défaut
192.168.1.1

- On entre les données de connexion pour accéder au dashboard principale de routeur

configuration du routeur :



<input type="radio"/> Disable DHCP Server	<input checked="" type="radio"/> Enable DHCP Server				
Start IP Address:	192.168.1.100				
End IP Address:	192.168.1.200				
Leased Time (hour):	24 (1~48)				
Static IP Lease List: (A maximum 32 entries can be configured)					
MAC Address	IP Address	Status	Enable/Disable	Edit	Remove
3c:a0:67:1f:fd:b8	192.168.1.109	Enabled	<input type="button" value="Disable"/>	<input type="button" value="Edit"/>	<input type="checkbox"/>
<input type="button" value="Add"/> <input type="button" value="Enable All"/> <input type="button" value="Select All"/> <input type="button" value="Remove"/>					
<input type="button" value="Save/Apply"/>					

***On ajoute l'adresse MAC du serveur (PC) 192.168.1.109**

***On l'assigne une adresse IP statique 192.168.1.109**

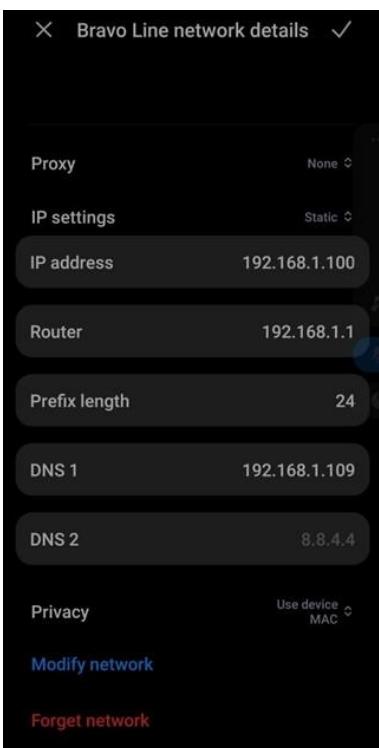
dans ce cas *Sauvegarder les changements

Maintenant, nous configurons les paramètres DNS dans le routeur pour qu'il fonctionne comme un serveur DNS pour le site Web



*modifier les paramètres de DNS

- Après ces changements, chaque appareil qui se connecte à ce réseau aura 192.168.1.109 comme DNS principal et pourra accéder au www.hsup.uae.ma

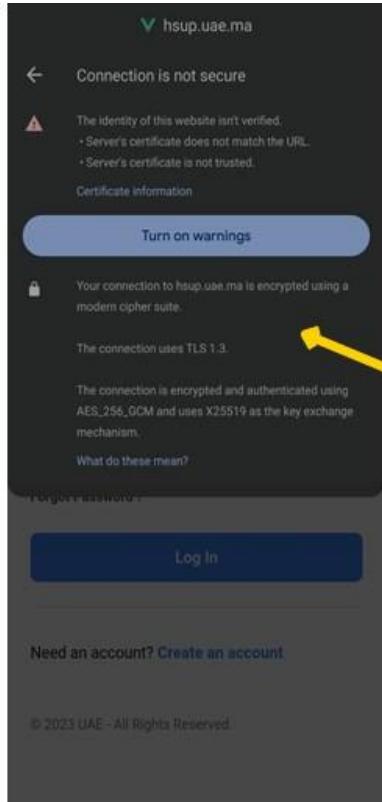


*DNS est bien configuré au niveau de réseau



*Le domaine
"hsup.uae.ma"
fonctionne
correctement

Fonctionnalités https et ssl/tls fonctionnent également sans problème



*Les détails de la certificat ssl/tls



4.6.5. **POSTFIX**

4.6.5.1. **Etape1 : Teste API**

Alors on Remarque que notre route pour obtenir Endpoint ca marche :

```
← → ⌂ ⓘ 127.0.0.1:8000/api/postfix
1 // 20230524164501
2 // http://127.0.0.1:8000/api/postfix
3
4 ▾ [
```

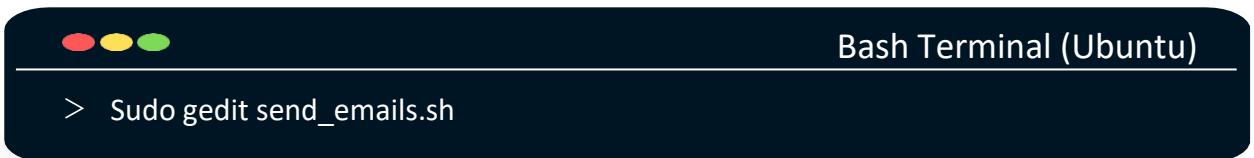
4.6.5.2. **Etape2 : Configuration Postfix :**

On a déjà fait la configuration de postfix pour envoyer l'email,mais juste a une personne precise et static,maintenant on a besoin d'envoyer a chaque prof leur email de notification concernant la fiche de paie.

D'abord on commence par la creation de fichier « **script shell** »

```
abderrahmane@dns:~$ touch send_emails.sh
```

On ouvre ce fichier utilidant la commande suivante :



```

Open ▾ | + send_emails.sh
~/
1 #!/bin/bash
2
3 # Récupérer les données de l'API
4 data=$(curl http://127.0.0.1:8000/api/postfix)
5
6 # Parcourir les données et envoyer les e-mails
7 for row in $(echo "$data" | jq -c '.[]'); do
8     email=$(echo "{$row}" | jq -r '.email')
9     url=$(echo "{$row}" | jq -r '.url')
10
11    # Construire le contenu de l'e-mail avec l'URL personnalisée
12    email_body=<html>
13    <head>
14        <style>
15            body {
16                font-family: Arial, sans-serif;
17                line-height: 1.5;
18            }
19
20            .container {
21                max-width: 600px;
22                margin: 0 auto;
23                padding: 20px;
24            }
25
26            .title {
27                font-size: 20px;
28                font-weight: bold;
29                margin-bottom: 10px;
30            }
31
32            .message {
33                margin-bottom: 20px;
34            }
35
36            .link {
37                display: inline-block;
38                padding: 10px 20px;
39                background-color: #007bff;
40                color: #ffffff;
41                text-decoration: none;
42                border-radius: 5px;
43            }
44        </style>
45    </head>
46    <body>
47        <div class=\"container\">
48            <div class=\"title\">Cher enseignant:</div>
49
50            <div class=\"message\">
51
52            <div class=\"message\">
53                Veuillez cliquer sur le lien suivant pour y avoir accès :
54            </div>
55
56            <div class=\"message\" >
57                <a class=\"link\" href="http://127.0.0.1:8000${url}" style="color: white;">Telecharger Fiche de Paie</a>
58            </div>
59
60            <div class=\"message\">
61                Cordialement,<br>
62                Votre Service XtraDH
63            </div>
64
65        </div>
66    </body>
67 </html>
68
69    # Envoyer l'e-mail en utilisant Postfix
70    echo -e "${email_body}" | mail -a "Content-Type: text/html" -s "Votre fiche de paie" "${email}"
71 done
72
73

```

4.6.5.3.

Etape3 : Teste sur ce API

```

41  {
42      "nom": "harrachy",
43      "prenom": "abdo",
44      "email": "abdotest2023@gmail.com",
45      "url": "/api/generate-pdf/7"
46  },
47  {
48      "nom": "Achernan",
49      "prenom": "Mohamed",
50      "email": "okabe2015rintaro@gmail.com",
51      "url": "/api/generate-pdf/8"
52  },
53  {
54      "nom": "Amine",
55      "prenom": "Taha",
56      "email": "tahaamine2002@gmail.com",
57      "url": "/api/generate-pdf/666"
58  }

```

4.6.5.4.

Etape4 : Exécution de script shell

```

abderrahmane@dns:~$ ./send_emails.sh
% Total    % Received % Xferd  Average Speed   Time     Time     Time  Current
                                         Dload  Upload   Total   Spent   Left  Speed
100  916    0  916    0      0   126      0  --::--:--  0:00:07  --::--:--  228
parse error: Unfinished string at EOF at line 2, column 0
parse error: Unfinished string at EOF at line 2, column 0
parse error: Invalid numeric literal at line 1, column 6
parse error: Invalid numeric literal at line 1, column 6
parse error: Unfinished string at EOF at line 2, column 0
parse error: Unfinished string at EOF at line 2, column 0
parse error: Invalid numeric literal at line 1, column 6
parse error: Invalid numeric literal at line 1, column 6
abderrahmane@dns:~$
```

4.6.5.5.

Etape5 : Vérification de la réception d'email :

The screenshot shows a Gmail inbox with the following details:

- Header:** mail.google.com/mail/u/1/?ogbl#inbox
- Navigation:** Nouveau message, Rechercher dans les messages, Boîte de réception (10), Principale, Promotions, Réseaux sociaux.
- Profile:** Abderrhamane test, abdotest2023@gmail.com, Gérer votre compte Google.
- Email Preview:** "Votre fiche de paie" - Cher enseignant: Cet e-mail vous a été envoyé afin de vous notifier que votre fiche de paie de l'an...
- Message Content:**
 - From: XtraDh <noreplyxtradh@gmail.com>
 - To: À moi
 - Date: 16:52 (il y a 4 minutes)
 - Subject: Votre fiche de paie
 - Body:

Cher enseignant:

Cet e-mail vous a été envoyé afin de vous notifier que votre fiche de paie de l'année est désormais prête à être téléchargée.

Veuillez cliquer sur le lien suivant pour y avoir accès :

Télécharger Fiche de Paie

Cordialement,
Votre Service XtraDh

Lorsque cliquera sur le bouton on Blue, la fiche de paie de ce prof ca sera installer automatiquement.

UNIVERSITÉ ABDELMALEK ESSAADI

Fiche de Paiement
2023-05-25 16:24:39

À :

harrachy abdo

Status : Professeur Agrége
Etablissement d'origine : École nationale des sciences appliquées de Tanger
Année universitaire : 2022/2023

ID	ETABLISSEMENT	VOLUME HORRAIRE	SALAIRE BRUTE	SALAIRE NET
	Volume horraire		0 Heures	
Total			MAD 0	

Detail des interventions

ID	INTERVENTION	ETABLISSEMENT	SEMESTRE	DÉBUT	FIN
	Service XtraDH vous souhaite une bonne journée! www.uae.ac.ma				

UNIVERSITÉ ABDELMALEK ESSAADI

Fiche de Paiement
2023-06-11 10:07:58

À :

Soufian yassingl

Status : Professeur Habilite
Etablissement d'origine : École nationale des sciences appliquées de Tanger
Année universitaire : 2022/2023

ID	ETABLISSEMENT	VOLUME HORRAIRE	SALAIRE BRUTE	SALAIRE NET
	Volume horraire		0 Heures	
Total			MAD 0	

Detail des interventions

ID	INTERVENTION	ETABLISSEMENT	SEMESTRE	DÉBUT	FIN
	Service XtraDH vous souhaite une bonne journée! www.uae.ac.ma				

On remarque que les emails sont envoyés au même temps

Et apres ca j'ai regler postfix pour envoyer de manière automatique à une date précise, car dans notre cahier de charge on est besoin d'envoyer l'email à la fin de l'année, on gère ca par la commande suivant :



Bash Terminal (Ubuntu)

```
> at 11:40 2023-06-11 <<< "./send_emails.sh"
```

4.6.5.6.

Etape6 : docker-compose/Project

ELAALOUCHE / XtraDh (Private)

Code Issues Pull requests Actions Projects Wiki Security Insights

Search branches...

Overview Yours Active Stale All branches New branch

Default branch

develop Updated now by MOHAMMED YASSINE EL AALOUCHE Default

Active branches

- feature/validate Updated 1 hour ago by MohamedElHaddad03
- feature/continue-interfaces Updated 2 hours ago by sofexbk
- feature/prof_feature Updated 4 hours ago by GliouYassine
- ELAALOUCHE-patch-4 Updated 5 hours ago by MOHAMMED YASSINE EL AALOUCHE
- release/0.2.0 Updated yesterday by ELAALOUCHE

#71 Merged #70 Merged #69 Open #66 Merged

New pull request

View more active branches >

D'abord on récupère le code a partir de GitHub, at précisément d'une branche qui s'appelle **release 0.2.0**

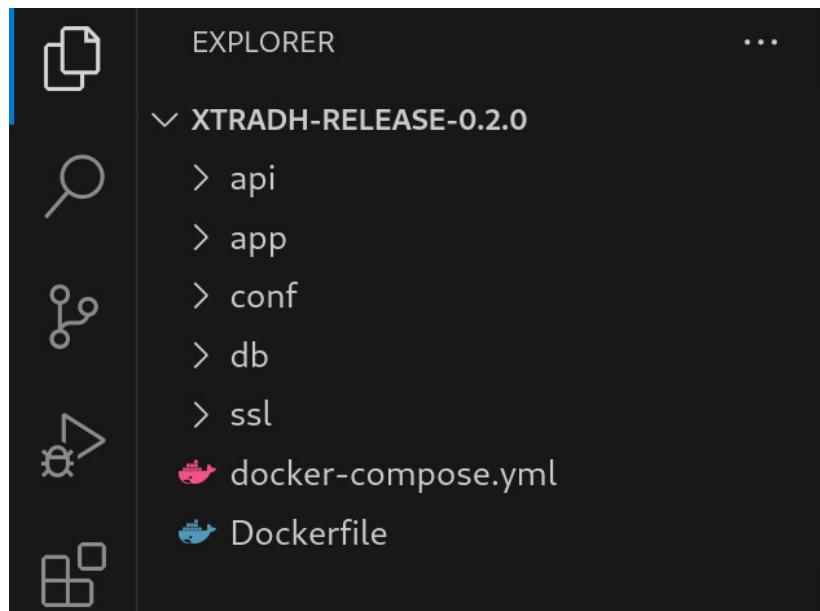
Car il contient la version la plus stable.

release/0.2.0 Updated yesterday by ELAALOUCHE

New pull request

View more active branches >

Après on transfère le code telle être de manière suivante :



Telle que :

Folder api: contient les dossiers et les fichiers de laravel juste on ajouter un fichier « Dockerfile »

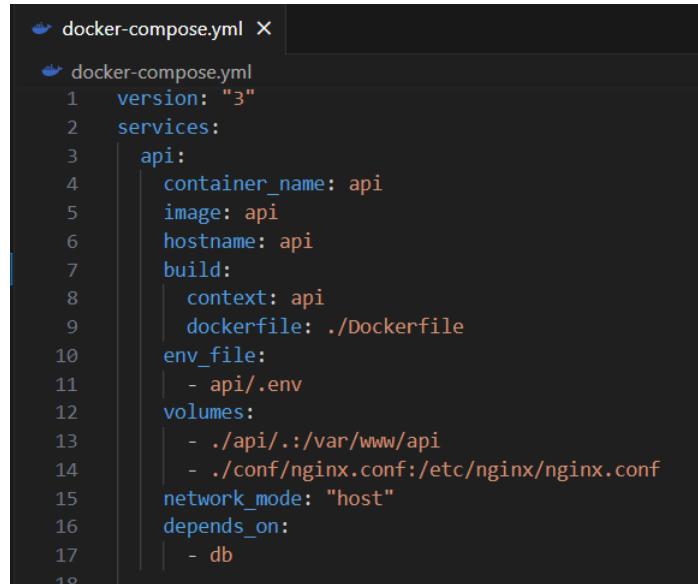
4.6.6. Laravel & API

```
Dockerfile X
api > Dockerfile
17  COPY . /var/www/api
18
19  ENV COMPOSER_ALLOW_SUPERUSER=1
20
21  RUN composer install --optimize-autoloader --no-dev
22
23  RUN php artisan key:generate
24
25  RUN chown -R www-data:www-data /var/www/api/storage /var/www/api/bootstrap/cache
26
27  CMD php artisan serve --host='127.0.0.1' --port=8000
```

Ce code prépare un environnement isolé et configuré spécifiquement pour exécuter l'API Laravel, en installant toutes les dépendances nécessaires et en configurant les paramètres de base. Cela permet de garantir que l'API Laravel fonctionne correctement dans le contexte du conteneur Docker.

4.6.7. Laravel & NGINX Aussi POSTGRES

On ouvre le fichier docker-compose.yml



```
docker-compose.yml ×
  ↳ docker-compose.yml
1   version: "3"
2   services:
3     api:
4       container_name: api
5       image: api
6       hostname: api
7       build:
8         context: api
9         dockerfile: ./Dockerfile
10      env_file:
11        - api/.env
12      volumes:
13        - ./api/:/var/www/api
14        - ./conf/nginx.conf:/etc/nginx/nginx.conf
15      network_mode: "host"
16      depends_on:
17        - db
18
```

Ce code configure plutôt un service appelé "api" dans un environnement de conteneur Docker à l'aide d'un fichier docker-compose.yaml. Ce service peut potentiellement exécuter une application qui utilise Laravel et Nginx

Apres on a sur le même fichier :

docker-compose.yml

```

19   app:
20     container_name: app
21     image: app
22     hostname: app
23     build:
24       context: .
25     volumes:
26       - ./api/:/var/www/api
27     network_mode: "host"
28     depends_on:
29       - db

```

Ce code configure un service "app" dans un environnement de conteneur Docker en spécifiant l'image, Il configure plutôt un service appelé "app" dans un environnement de conteneur Docker à l'aide d'un fichier docker-compose.yaml. Ce service peut potentiellement exécuter une application qui utilise Nginx et Vue.js

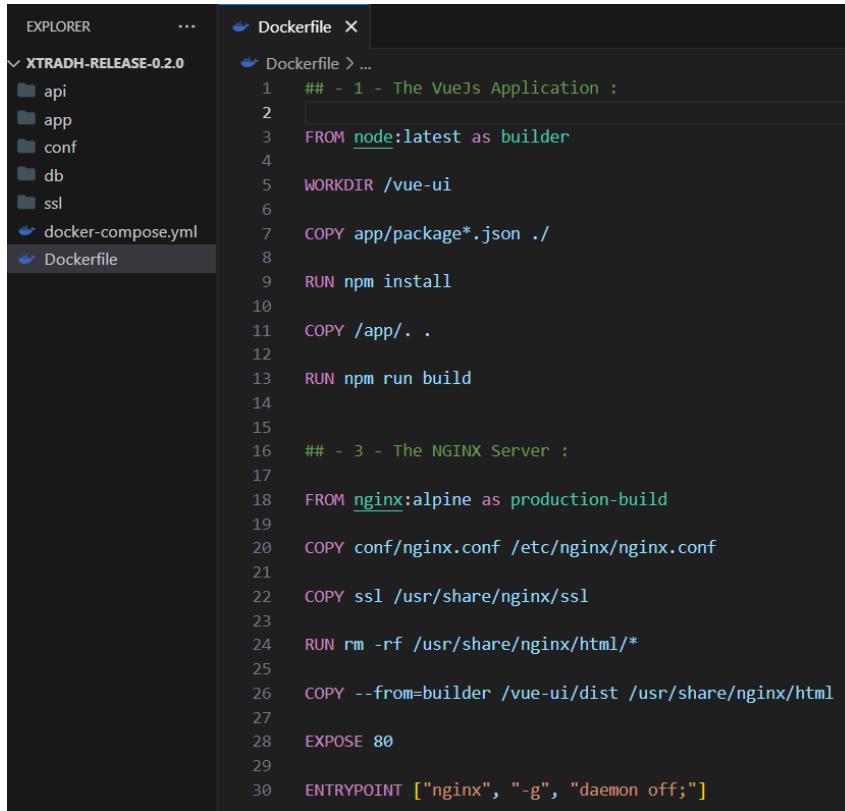
```

31   db:
32     container_name: db
33     image: postgres:13
34     hostname: db
35     restart: always
36     ports:
37       - 5432:5432
38     volumes:
39       - ./db/init.sql:/docker-entrypoint-initdb.d/0001-init.sql
40       - ./db/triggers.sql:/docker-entrypoint-initdb.d/0002-init.sql
41       - ./db/insertion.sql:/docker-entrypoint-initdb.d/0003-init.sql
42     environment:
43       - POSTGRES_USER=postgres
44       - POSTGRES_PASSWORD=postgres
45       - POSTGRES_DB=postgres
46       - POSTGRES_HOST=127.0.0.1
47

```

Ce code configure un service "db" dans un environnement de conteneur Docker en spécifiant l'image **PostgreSQL**, les ports à mapper, les volumes pour les scripts **SQL** d'initialisation et les variables d'environnement pour la configuration de la base de données. Cela permet de créer et d'exécuter une base de données PostgreSQL dans un conteneur Docker avec les paramètres spécifiés.

Après dans un fichier qui s'appelle Dockerfile qui se trouve dans la racine :



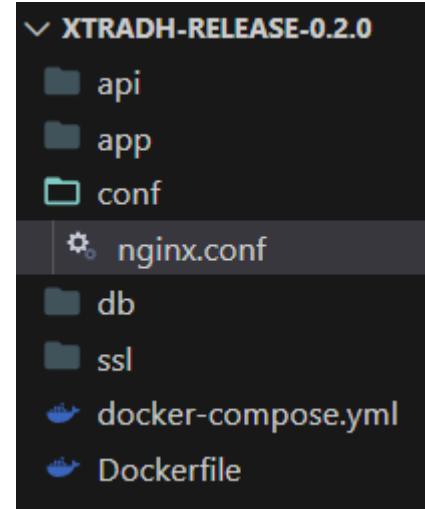
```

EXPLORER      ...
Dockerfile X
Dockerfile > ...
1 ## - 1 - The Vue.js Application :
2
3 FROM node:latest as builder
4
5 WORKDIR /vue-ui
6
7 COPY app/package*.json .
8
9 RUN npm install
10
11 COPY ./app/
12
13 RUN npm run build
14
15
16 ## - 3 - The NGINX Server :
17
18 FROM nginx:alpine as production-build
19
20 COPY conf/nginx.conf /etc/nginx/nginx.conf
21
22 COPY ssl /usr/share/nginx/ssl
23
24 RUN rm -rf /usr/share/nginx/html/*
25
26 COPY --from=builder /vue-ui/dist /usr/share/nginx/html
27
28 EXPOSE 80
29
30 ENTRYPOINT ["nginx", "-g", "daemon off;"]

```

ce code construit une application **Vue.js** à l'aide de l'image "node:latest" et génère une version optimisée de l'application. Ensuite, il configure un serveur **NGINX** en utilisant l'image "nginx:alpine" et copie les fichiers de l'application Vue.js dans le répertoire NGINX pour les servir. Le serveur NGINX écoute sur le port **80** et est exécuté en mode non démon pour gérer les requêtes entrantes vers l'application **Vue.js**.

Et finalement il reste un fichier qui s'appelle « nginx.conf » dans un folder appeler conf :



```

nginx.conf
conf > nginx.conf
43     server {
44         listen 192.168.1.136:443 ssl http2;
45         server_name hsup.uae.ma www.hsup.uae.ma;
46
47         access_log /var/log/nginx/access.log compression;
48         root /usr/share/nginx/html;
49         index index.html index.htm;
50
51         ssl on;
52         ssl_certificate /usr/share/nginx/ssl/certificate.crt;
53         ssl_certificate_key /usr/share/nginx//ssl/private.key;
54
55         location / {
56             if (!-e $request_filename) {
57                 rewrite ^(.*)$ /index.html break;
58             }
59         }
60         location ~* \.(?:jpg|jpeg|gif|png|ico|cur|gz|svg|svgz|mp4|ogg|ogv|webm|htc)$ {
61             expires 1M;
62             access_log off;
63             add_header 'Cache-Control' 'public';
64             add_header 'Access-Control-Allow-Origin' '*';
65             add_header 'Access-Control-Allow-Credentials' 'true';
66         }
67         location ~* \.(?:css|js)$ {
68             try_files $uri =404;
69             expires 1y;
70             access_log off;
71             add_header Cache-Control "public";
72         }
73         location ~ ^.+\.+\+$ {
74             try_files $uri =404;
75         }
76     }
77     server []
78         listen 8000;
79         index index.php index.htm index.html;
80
81         error_log /var/log/nginx/error.log;
82         access_log /var/log/nginx/access.log;
83
84         server_name hsup-backend.uae.ma www.hsup-backend.uae.ma;
85         root /var/www/api/public;
86
87         location / {
88             try_files $uri $uri/ /index.php?$query_string;
89         }
90

```

Ce code configure le serveur NGINX pour gérer les requêtes HTTP et HTTPS, rediriger les requêtes, gérer les fichiers statiques, et configurer la journalisation des accès. Les blocs de serveur commentés fournissent des exemples supplémentaires de configurations possibles pour une redirection spécifique et l'utilisation d'un proxy pour une API.

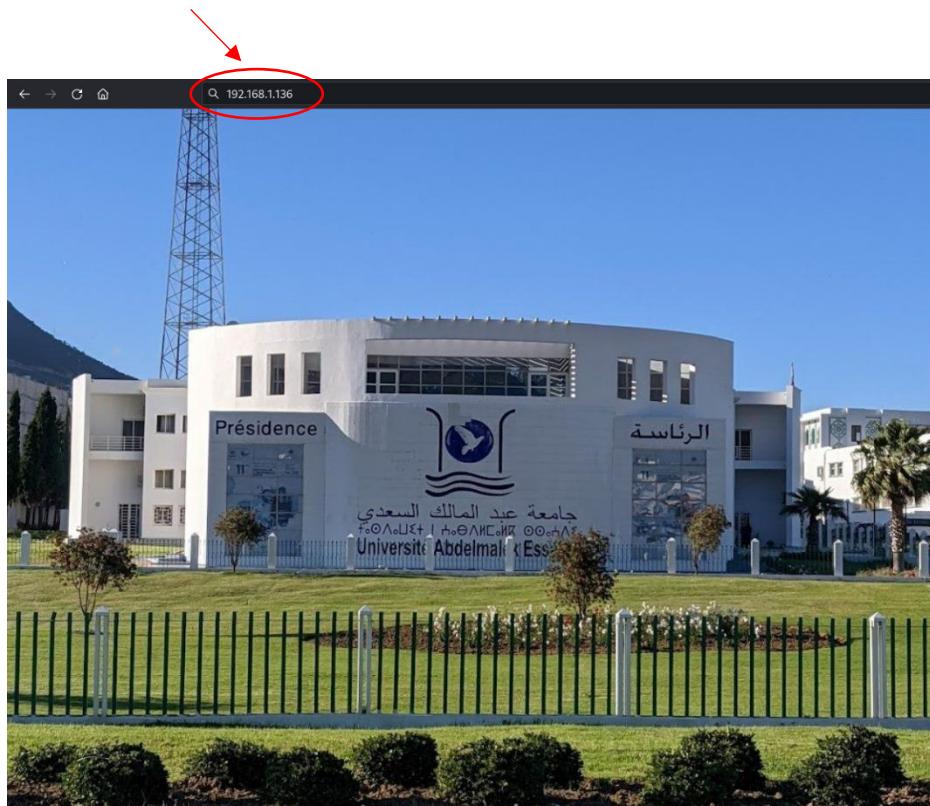
4.6.8. Tester

D'abord on fait la commande :

```
docker-compose up --build
```

Voici notre site :

La même adresse IP qui se trouve dans le fichier nginx.conf



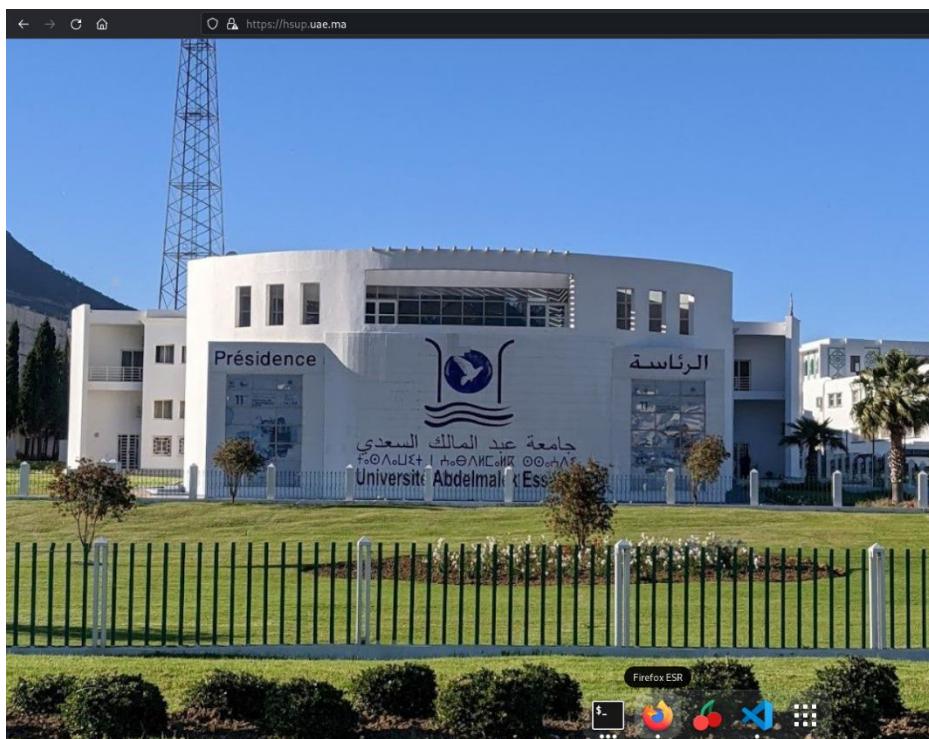
Entrer l'e-mail

Entrer le mot de passe

Connexion

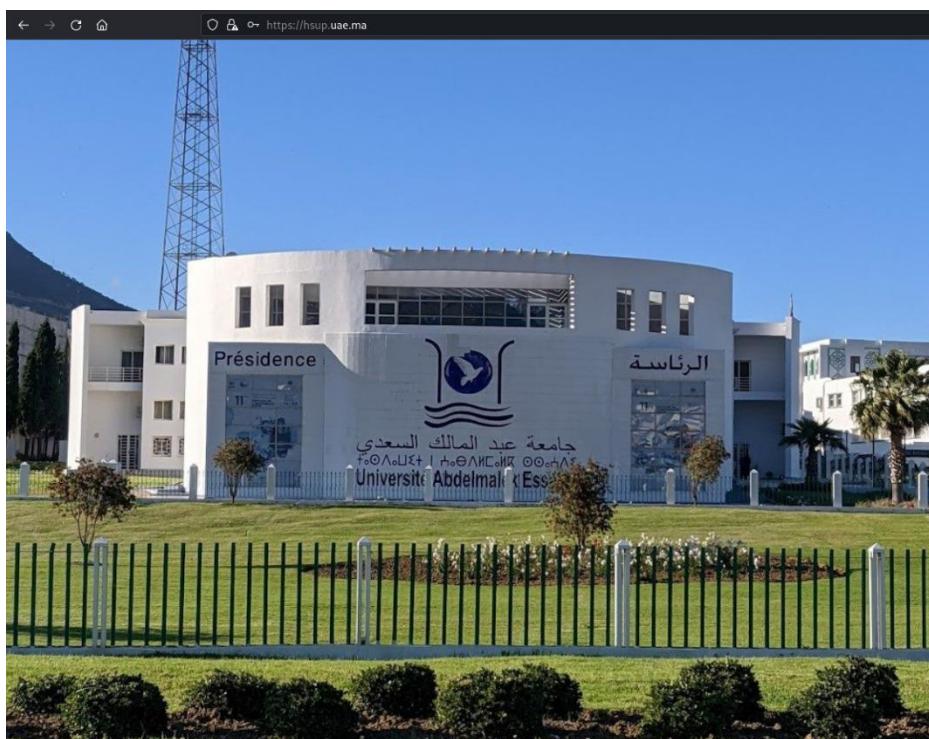
© 2023 UAE - Tous les droits sont réservés

Après click sur la bouton Entrer on remarque que le site est convertit en :
<https://hsup.uae.ma>



On teste juste une route et le reste dans la vidéo :

Les données de login il s'agit d'un prof :



On remarque que la route est bien passée et aussi on remarque une marque en vers qui signifie qu'il s'agit d'un **prof**:

The screenshot shows a web browser window with the URL <https://hsup.uae.ma/Profileprof>. The page is titled "Professeur". On the left sidebar, there are four menu items: "Paiement" (Payment), "Interventions" (Interventions), "Profil" (Profile), and "Déconnexion" (Logout). The main content area displays a form for updating a professor's profile. The fields are as follows:

- Nom: Soufian
- Prénom: yassingl
- Email: aminetaha1964@gmail.com
- PPR: 666666
- Établissement: Faculté des Lettres et des Sciences Humaines

A blue "Modifier" (Update) button is located at the bottom of the form.

5. CHAPITRE 5 :

GESTION DU CODE AVEC GIT/GITHUB

Dans ce chapitre, nous aborderons en détail notre utilisation de Git et GitHub pour la gestion du code source de notre projet. Nous présenterons les principes fondamentaux de Git, son rôle dans le contrôle de version et la collaboration au sein de l'équipe.

5.1. Présentation Git et GitHub

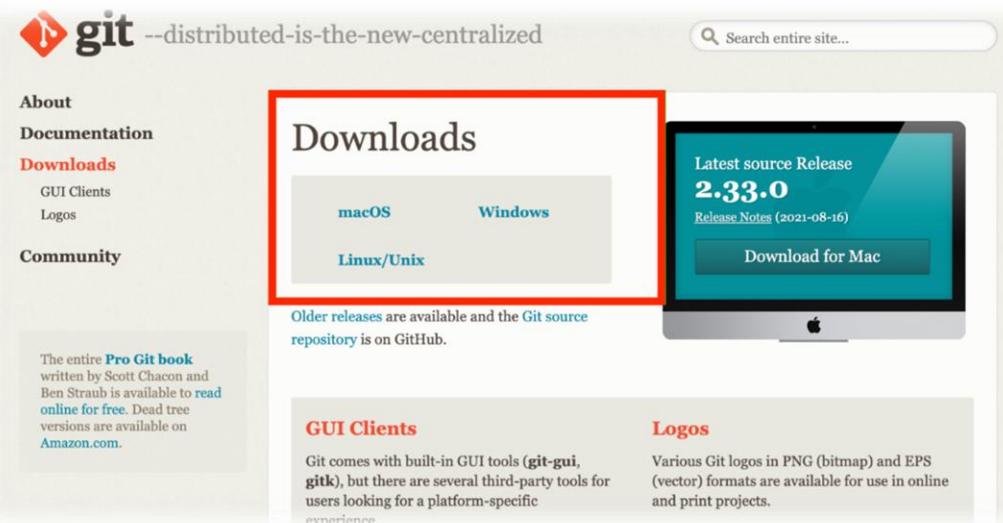
5.1.1. Git

5.1.1.1. Définition

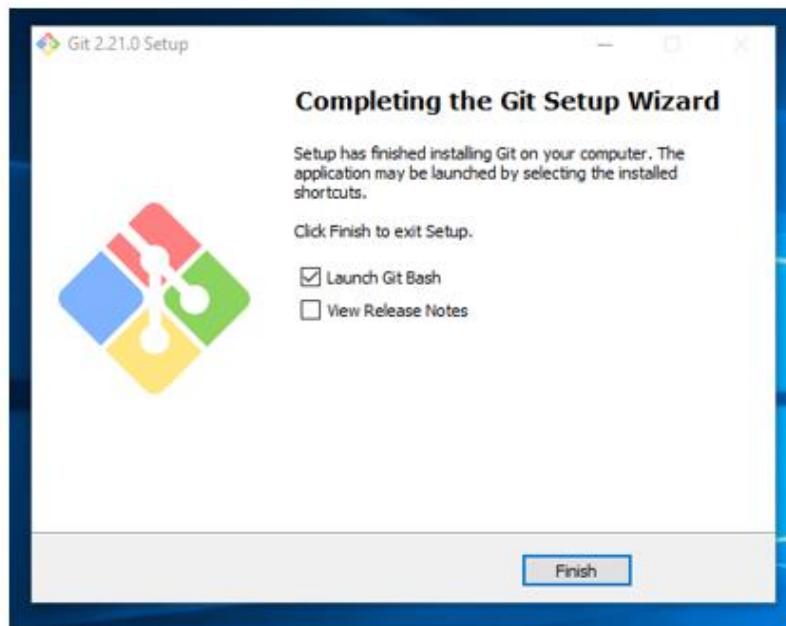
Git est un système de contrôle de version, largement utilisé pour la gestion du code source et la collaboration entre développeurs. Il permet de suivre les modifications apportées aux fichiers et d'enregistrer l'historique des différentes versions. Git est conçu pour être rapide, efficace et adapté aux projets de toutes tailles.

5.1.1.2. Téléchargez et configurer Git

- Téléchargez la version de Git qui correspond à votre installation : MacOs, Windows ou Linux/Unix



- Appuyez sur Suivant à chaque fenêtre puis sur Installer. Lors de l'installation, laissez toutes les options par défaut.



- Lancez Git Bash " Git Bash est l'interface permettant d'utiliser Git en ligne de commande".

5.1.1.3. Configuration locale

La première chose à faire est de configurer votre identité. Pour cela, vous allez entrer dans le monde des lignes de commande ! Commencez par renseigner votre nom et votre adresse e-mail. C'est une information importante car vous en aurez besoin pour toutes vos validations dans Git :

```
Lenovo@DESKTOP-R1ATFEG MINGW64 ~/Desktop
$ git config --global user.name "MOHAMMED YASSINE EL AALOUCH"

Lenovo@DESKTOP-R1ATFEG MINGW64 ~/Desktop
$ git config user.name
MOHAMMED YASSINE EL AALOUCH

Lenovo@DESKTOP-R1ATFEG MINGW64 ~/Desktop
$ git config --global email.name "elaalouch.dev@gmail.com"

Lenovo@DESKTOP-R1ATFEG MINGW64 ~/Desktop
$ git config email.name
elaalouch.dev@gmail.com
```

Grâce à l'option --global, vous n'aurez besoin de le faire qu'une fois.

5.1.2. GitHub

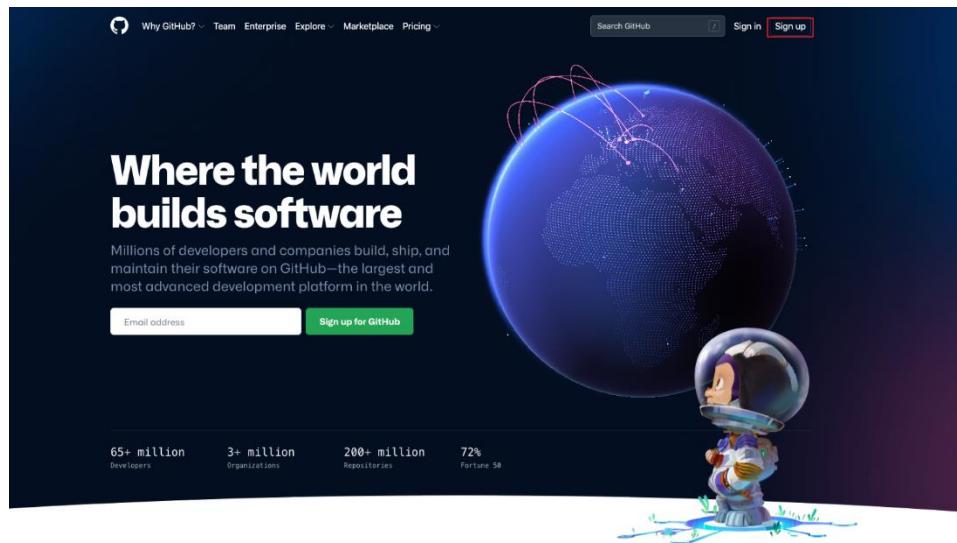
5.1.2.1. Définition

GitHub, quant à lui, est une plateforme web qui utilise Git pour héberger des projets et faciliter la collaboration entre développeurs. Il offre des fonctionnalités supplémentaires telles que le suivi des problèmes, la gestion des demandes de fusion (pull requests), les discussions, etc. GitHub permet aux développeurs de partager leur code source, de travailler ensemble sur des projets et de contribuer à des projets open source.

5.1.2.2. Créez un compte GitHub

- Sur la page d'accueil de GitHub, vous verrez un bouton "Sign up" (S'inscrire) en haut à droite de la page. Cliquez dessus.
- Vous serez redirigé vers le formulaire d'inscription. Remplissez les champs requis, à savoir :
 - Username (Nom d'utilisateur) : Choisissez un nom unique qui vous identifie sur GitHub.
 - Email address (Adresse e-mail) : Entrez votre adresse e-mail valide.
 - Password (Mot de passe) : Choisissez un mot de passe sécurisé pour votre compte.
 - Cochez la case "I agree to the GitHub Terms of Service" (J'accepte les conditions d'utilisation de GitHub) pour indiquer que vous acceptez les conditions d'utilisation.
 - Si vous souhaitez recevoir des mises à jour et des communications de GitHub, vous pouvez cocher la case "Send me occasional product updates, announcements, and offers" (Envoyez-moi occasionnellement des mises à jour produits, des annonces et des offres).
- Une fois que vous avez rempli tous les champs requis et coché les cases appropriées, cliquez sur le bouton vert "Create account" (Créer un compte).
- GitHub peut vous demander de résoudre un puzzle de vérification pour confirmer que vous êtes un utilisateur réel et non un robot. Suivez les instructions pour résoudre le puzzle.

- Après avoir résolu le puzzle, vous serez redirigé vers la page de bienvenue de GitHub où vous pourrez personnaliser votre profil et configurer vos préférences.



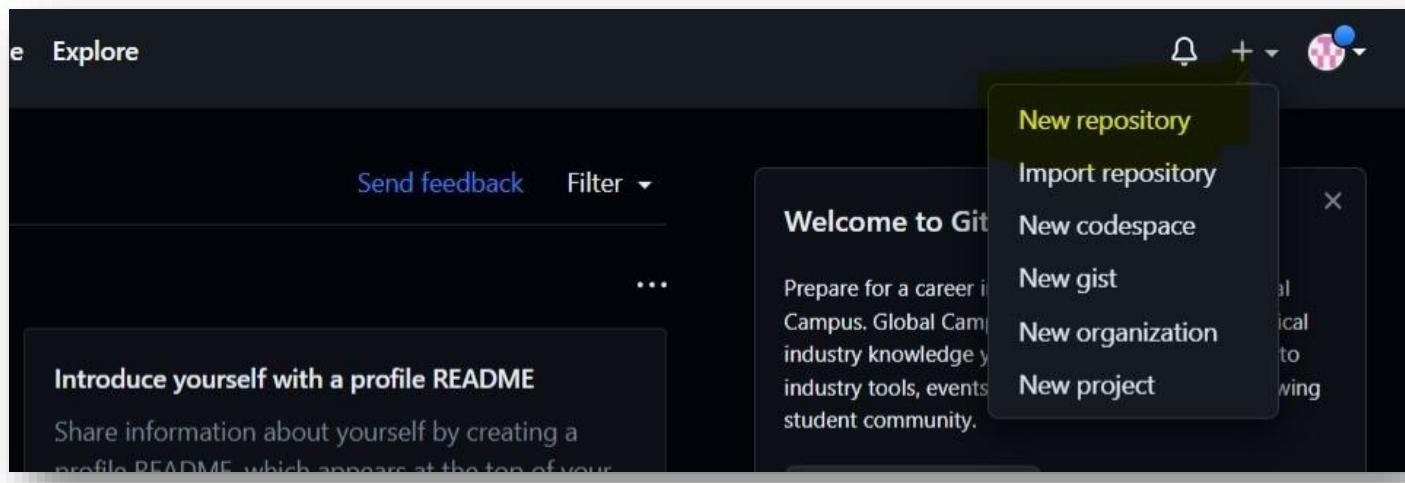
Cet outil a donc trois grandes fonctionnalités :

- Revenir à une version précédente de votre code en cas de problème.
- Suivre l'évolution de votre code étape par étape.
- Travailler à plusieurs sans risquer de supprimer les modifications des autres collaborateurs.

5.1.3. Crédit et la sécurisation du projet

5.1.3.1. Créez votre repository

Pour mettre votre projet sur GitHub, vous devez créer un repository (ou dépôt en français) dans lequel il pourra être installé.



Cliquez sur le "+" dans le coin supérieur droit, pour faire apparaître l'option New repository.

Choisissez un nom simple pour votre dépôt. Dans notre projet, nous utiliserons "XtraDh".

Puis, choisissez si vous souhaitez créer un dépôt public ou privé.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner *



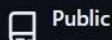
Repository name *

/ Projet_Integration

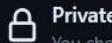
Projet_Integration is available.

Great repository names are short and memorable. Need inspiration? How about [fuzzy-carnival?](#)

Description (optional)



Anyone on the internet can see this repository. You choose who can commit.



You choose who can see and commit to this repository.

Initialize this repository with:

Add a README file

This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: [None](#) ▾

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

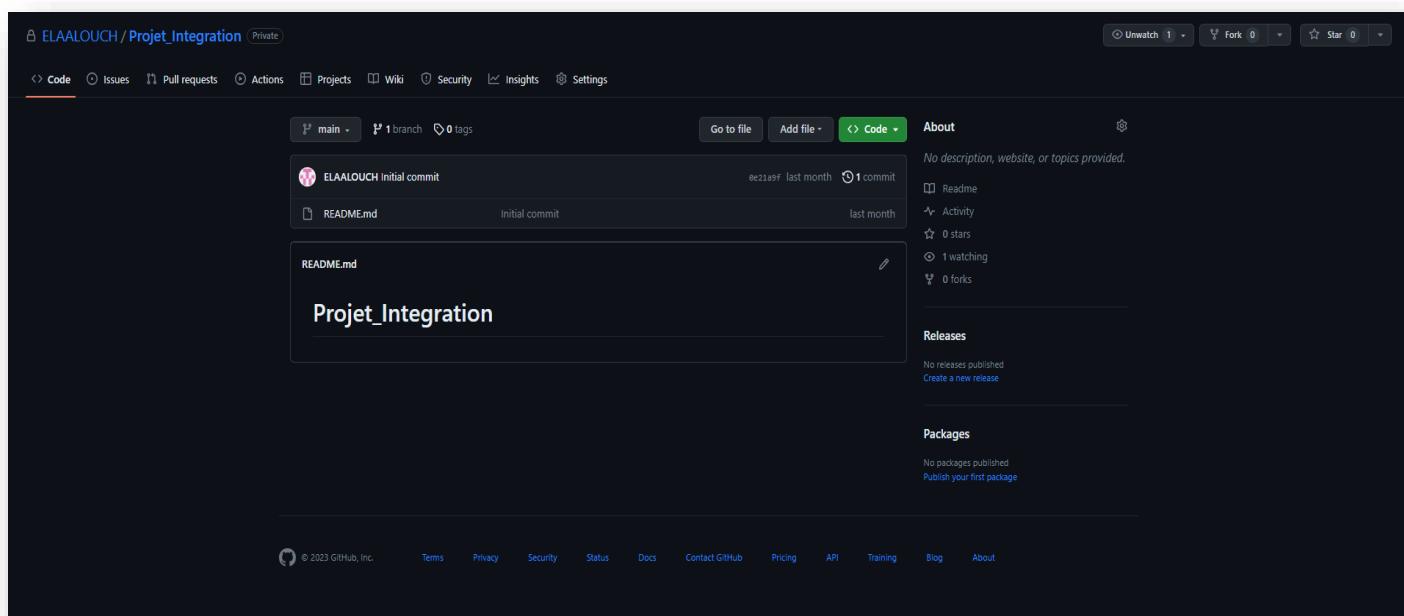
License: [None](#) ▾

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

-> README est un fichier qui indique les informations clés de votre projet : description, environnement à utiliser, dépendances possibles et droits d'auteurs. C'est un peu comme le mode d'emploi de votre projet.

-> gitignore est un fichier qui permet d'ignorer certains fichiers de votre projet Git.

Après la création du dépôt, vous serez redirigé vers l'espace du projet.

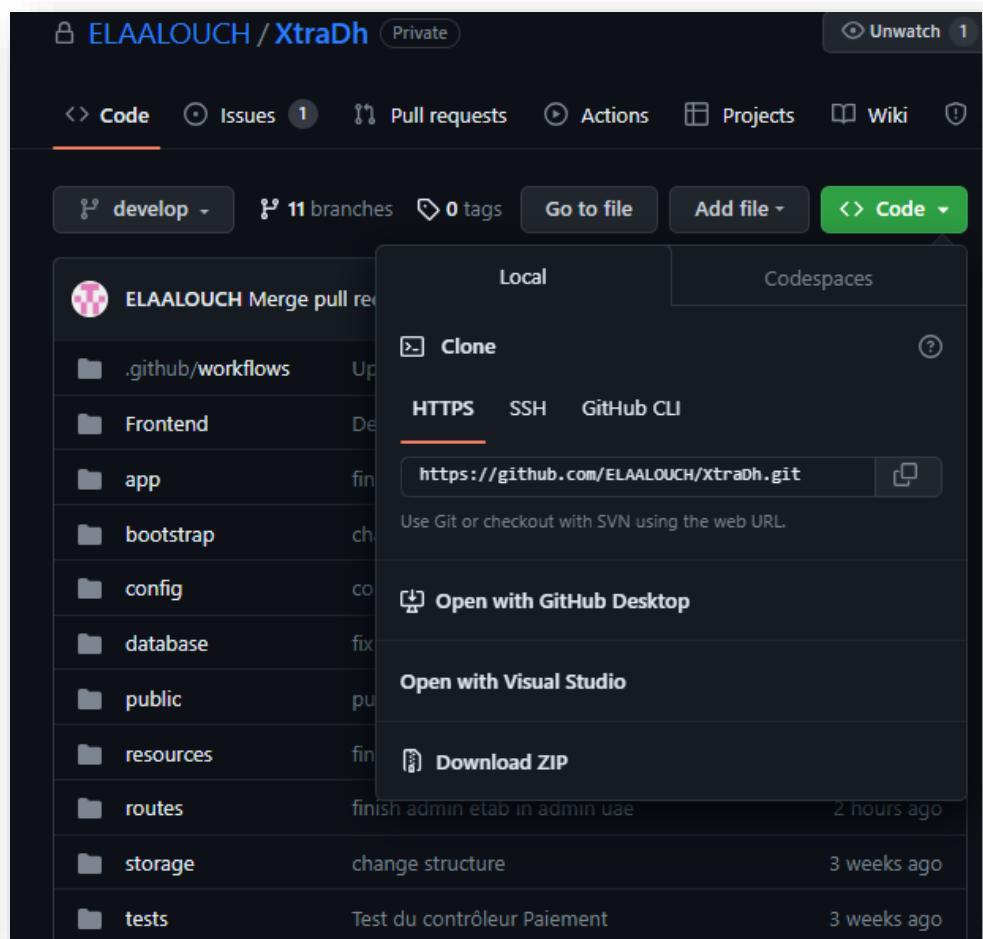


5.1.4. Clonage

Le clonage en GitHub et Git fait référence à la création d'une copie complète d'un dépôt Git distant sur votre machine locale. Cela vous permet de récupérer tout l'historique du projet, y compris toutes les branches et les commits, et de travailler sur le code localement.

Voici comment le processus de clonage fonctionne :

- Accédez à la page du dépôt GitHub que vous souhaitez cloner. Vous trouverez un bouton "Clone" vert sur la page du dépôt. Cliquez dessus pour afficher l'URL du dépôt.
- Copiez l'URL du dépôt.



- Ouvrez votre terminal (Git Bash).
- Utilisez la commande **git clone** suivie de l'URL du dépôt que vous avez copiée. Par exemple :

```
lenovo@DESKTOP-R1ATFEG MINGW64 ~/Desktop/Clone Rapport
$ git clone https://github.com/ELAALOUCH/XtraDh.git
Cloning into 'XtraDh'...
remote: Enumerating objects: 12662, done.
remote: Counting objects: 100% (3654/3654), done.
remote: Compressing objects: 100% (1467/1467), done.
Receiving objects: 22% (2843/12662), 5.70 MiB | 92.00 KiB/s
```

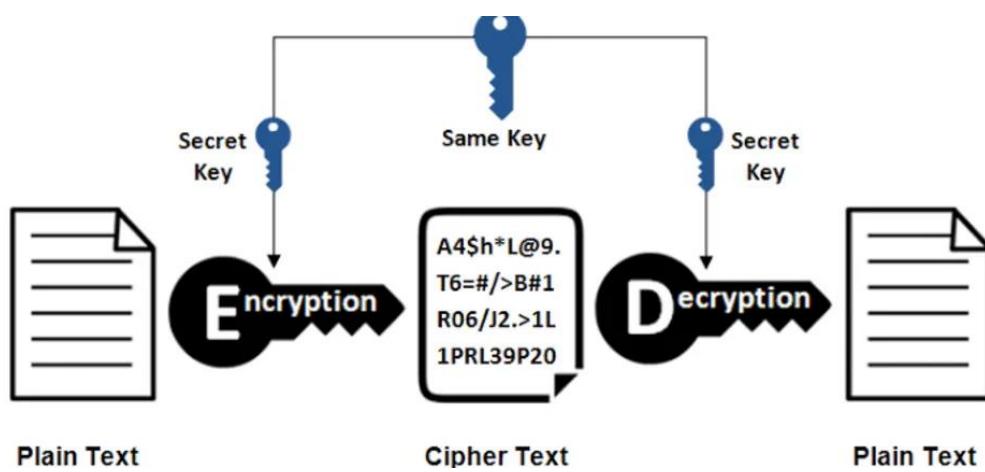
5.1.5. La sécurisation du projet à l'aide d'une clé SSH (Pour repository owner)

Une clé SSH (Secure Shell) est une paire de clés cryptographiques comprenant une clé privée et une clé publique. Dans le contexte de GitHub, une clé SSH est utilisée pour sécuriser les communications entre notre machine locale et les serveurs GitHub.

Lorsque vous générez une clé SSH, une clé privée est créée et stockée sur votre machine locale, tandis que la clé publique correspondante est ajoutée à votre compte GitHub. Lorsque vous effectuez des opérations telles que le clonage, le push ou le pull sur un dépôt GitHub, votre clé privée est utilisée pour vous authentifier auprès du serveur GitHub, confirmant ainsi que vous êtes bien le propriétaire autorisé du compte.

```
Lenovo@DESKTOP-R1ATFEG MINGW64 ~/Desktop/XtraDh09/XtraDh (develop)
$ git pull
Enter passphrase for key '/c/Users/Lenovo/.ssh/id_rsa': |
```

La clé privée doit être conservée en sécurité sur votre machine locale, car quiconque y a accès peut potentiellement accéder à vos dépôts et effectuer des opérations en votre nom. La clé publique, quant à elle, est accessible publiquement et peut être partagée avec les serveurs GitHub pour vous permettre d'accéder à vos dépôts de manière sécurisée.



En utilisant une clé SSH avec GitHub, vous évitez d'avoir à saisir votre nom

d'utilisateur et votre mot de passe à chaque interaction, ce qui facilite et sécurise votre expérience de travail avec des dépôts distants.

Pour générer une paire de clés SSH, vous pouvez suivre ces étapes :

- Exécutez la commande suivante pour générer une nouvelle paire de clés SSH :

```
lenovo@DESKTOP-R1ATFEG MINGW64 ~/Desktop
$ ssh-keygen -t rsa -C "MOHAMMED YASSINE"
```

- Vous serez invité à spécifier l'emplacement où vous souhaitez stocker la clé. Par défaut, elle sera enregistrée dans le répertoire .ssh de votre répertoire utilisateur. Vous pouvez appuyer sur "Entrée" pour accepter le chemin par défaut ou spécifier un autre emplacement.
- Ensuite, vous serez invité à fournir une phrase secrète (passphrase) pour protéger votre clé privée. Il s'agit d'une couche de sécurité supplémentaire, mais vous pouvez également appuyer sur "Entrée" pour laisser le champ vide si vous ne souhaitez pas utiliser de phrase secrète.

```
lenovo@DESKTOP-R1ATFEG MINGW64 ~/Desktop
$ ssh-keygen -t rsa -C "MOHAMMED YASSINE"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/lenovo/.ssh/id_rsa):
/c/Users/lenovo/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase): ← Yellow arrow pointing here
Enter same passphrase again:
```

- Le processus de génération des clés sera en cours, et une fois terminé, vous verrez les informations sur la clé générée, y compris l'emplacement de la clé publique et de la clé privée. Par exemple :

```

-----[RSA 3072]-----
E . |
o. . |
+ .... |
. . o. ==|
S . +.0 |
+ .o. . * |
o o.*.o +. |
. + + B++o*. |
+...+o*=B+o|

```



- Maintenant, vous avez généré une paire de clés SSH. La clé privée (id_rsa) est destinée à rester sur votre machine locale et doit être protégée. La clé publique (id_rsa.pub) est celle que vous allez ajouter à votre compte GitHub.
- Copiez le contenu de la clé publique (id_rsa.pub) en exécutant la commande suivante dans votre terminal :

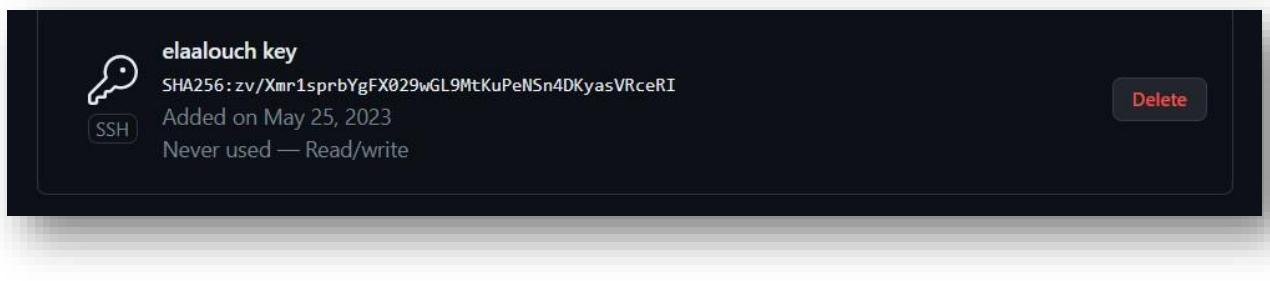
```

lenovo@DESKTOP-R1ATFEG MINGW64 ~/Desktop
$ cat /c/Users/Lenovo/.ssh/id_rsa.pub
sh-rsa AAAAB3NzaC1yc2EAAAQABAAQgQC+eLUXlnPam4LIhhY0Pvtu3Ph5BHucRbT9Y2OI+CQC
TuV0dv/T3NQC3VmmpJ0fNqbQY6EwJ1CFrxmX35ItqWLuvdPKGMPw1jmwHGqRjdQrEnfbY+1T7c+Kn8Lt
Wb8FKRGFxKaAj4U/X4hg5+4qAmmpH+EXFdHJfuMC4Xjm7hFU9n2zf68XYTa5KGUtkDbgAwNZJfepILc
IkVVRqGOF/+GgmQ9h/XhDjfVZxcRveurpesmsTucD+w194VZKnd7tQPqLx32f0vM4SToqAjaMGim8PYk
Fbd+nEHzRjXQ7/z8cM9jXA51xr19YMYebkrxYs03yYu14Zewc7HcKEAkw+8csOP58I4p8BnxODJhBcE
7bMRifxa+ei7lN1fNICQnApVp3qMt5MjR2u1/EkA+UGCa3wMYrQqQYCvoTRZk8uKwX3BdDYrcYoPFY
JOJAhYHuQrW03/56iYw/7mFlqFap7aw7vFxnjLrfUzAnbgZd2Jgo/jgHFkHb4rtLcrRCE0= MOHAMME
YASSINE

```

- Accédez à votre compte GitHub via votre navigateur, connectez-vous et accédez à votre page "Paramètres" (Settings).
- Dans le menu latéral, cliquez sur "SSH and GPG keys" (ou "Clés SSH" en français).
- Cliquez sur "New SSH key" (ou "Nouvelle clé SSH") pour ajouter une nouvelle clé.

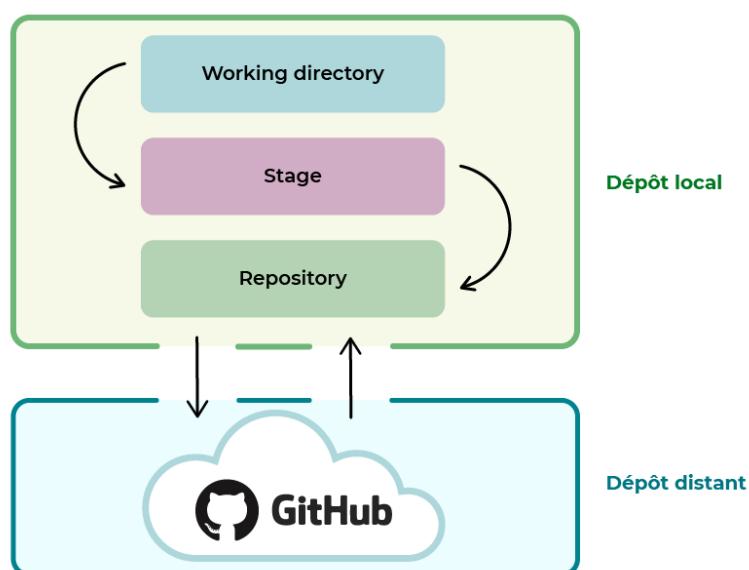
- Donnez un nom à votre clé (par exemple, "Ma clé SSH principale") et collez la clé publique que vous avez copiée précédemment dans le champ "Key" (ou "Clé").
- Cliquez sur "Add SSH key" (ou "Ajouter une clé SSH") pour enregistrer la clé publique sur votre compte GitHub.



- Une fois la clé publique ajoutée à votre compte GitHub, vous pourrez utiliser la clé privée correspondante sur votre machine locale pour vous authentifier et interagir avec les dépôts GitHub via SSH de manière sécurisée.

5.2. Méthodologie de Travail

5.2.1. Le fonctionnement de Git



Ce schéma illustre le fonctionnement de Git, comprenant trois zones qui forment le dépôt local, ainsi que le dépôt distant sur GitHub.

Examinons de plus près les différentes zones du dépôt local :

- Le répertoire de travail : Il s'agit du dossier du projet sur votre ordinateur. Lorsque nous avons initialisé le dépôt "XtraDh" dans la section précédente, il correspond à la zone bleue du schéma.
- L'espace de préparation ou index(stage) : Cette zone fait office d'intermédiaire entre le répertoire de travail et le référentiel (repository). Elle représente tous les fichiers modifiés que vous souhaitez inclure dans votre prochaine version du code.
- Le référentiel (repository) : C'est dans cette zone que les nouvelles versions d'un projet sont stockées lorsque vous les créez .

Ces trois zones sont présentes localement sur votre ordinateur.

- ✓ Initialiser votre dossier avec Git :

```
Lenovo@DESKTOP-R1ATFEG MINGW64 ~/Desktop/git
$ git init
Initialized empty Git repository in C:/Users/Lenovo/Desktop/git/.git/
Lenovo@DESKTOP-R1ATFEG MINGW64 ~/Desktop/git (master)
$
```

- ✓ Indexez vos fichiers avec la commande git add :

- Vous pouvez faire passer les fichiers vers l'index en utilisant la commande “git add” suivie du nom du fichier :

```
Lenovo@DESKTOP-R1ATFEG MINGW64 ~/Desktop/git (master)
$ touch index.html

Lenovo@DESKTOP-R1ATFEG MINGW64 ~/Desktop/git (master)
$ git add index.html
```

- ✓ Créez une nouvelle version avec la commande git commit :

- Maintenant que vos fichiers modifiés sont indexés, vous pouvez créer une version, c'est-à-dire archiver le projet en l'état. Pour ce faire, utilisez la commande "git commit" :

```
Lenovo@DESKTOP-R1ATFEG MINGW64 ~/Desktop/git (master)
$ touch index.html

Lenovo@DESKTOP-R1ATFEG MINGW64 ~/Desktop/git (master)
$ git add index.html

Lenovo@DESKTOP-R1ATFEG MINGW64 ~/Desktop/git (master)
$ git commit -m "first commit"
[master (root-commit) 1774f2d] first commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 index.html
```

5.2.2.

Workflow

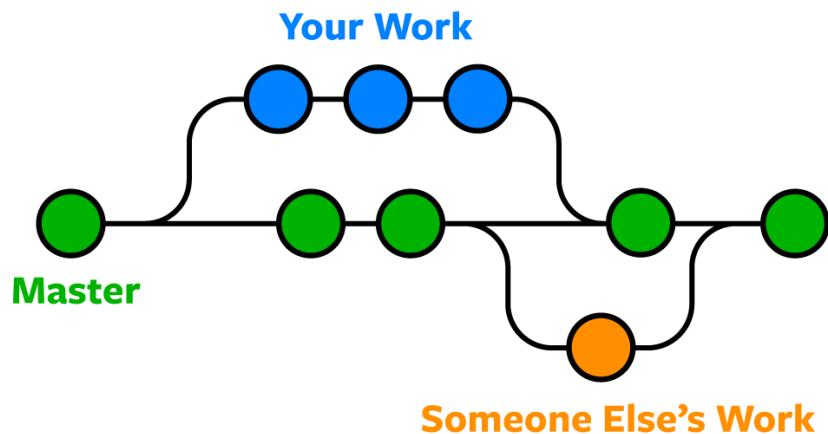
1. Qu'est-ce qu'un workflow ?

Il s'agit d'un ensemble d'instructions ou de recommandations qui guident l'utilisation efficace de Git et GitHub. En quelque sorte, c'est une approche adoptée par les collaborateurs pour orienter leurs actions.

Git lui-même ne préconise pas un workflow spécifique, car il vise à offrir la plus grande flexibilité possible. Ainsi, ce sont les collaborateurs qui ont le choix de déterminer le workflow à suivre. Ils ont deux options : créer un workflow spécifique pour chaque projet ou bien réutiliser des workflows existants.

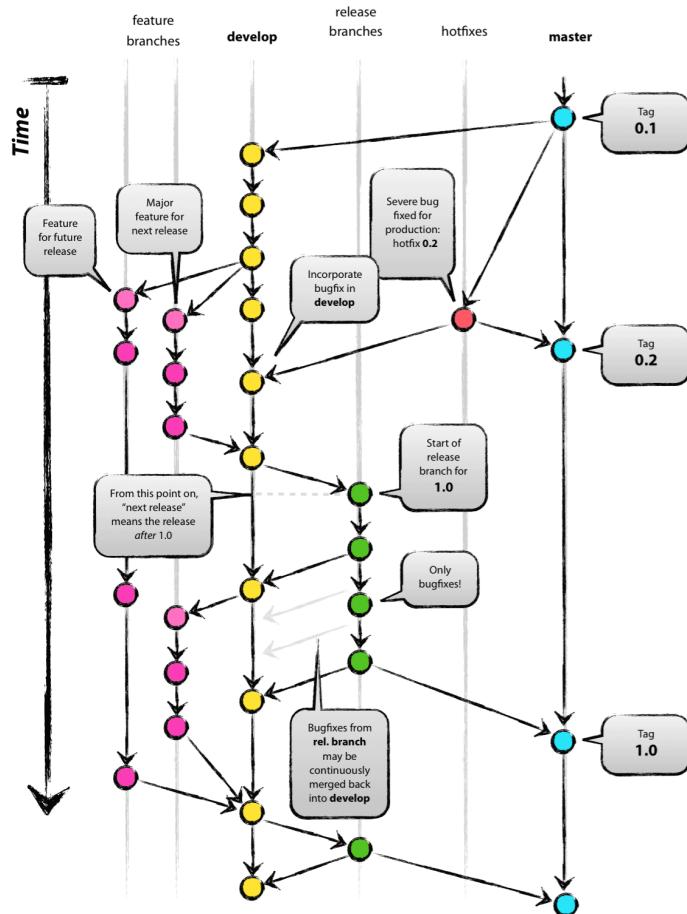
2. Les branches

En Git, les branches sont des références qui pointent vers un commit spécifique dans l'historique du dépôt. Elles sont utilisées pour travailler sur différentes fonctionnalités ou tâches de manière isolée, sans affecter la branche principale (généralement appelée branche "master" ou "main"). Les branches permettent de développer de manière parallèle et de fusionner ensuite les changements lorsque le travail est terminé.

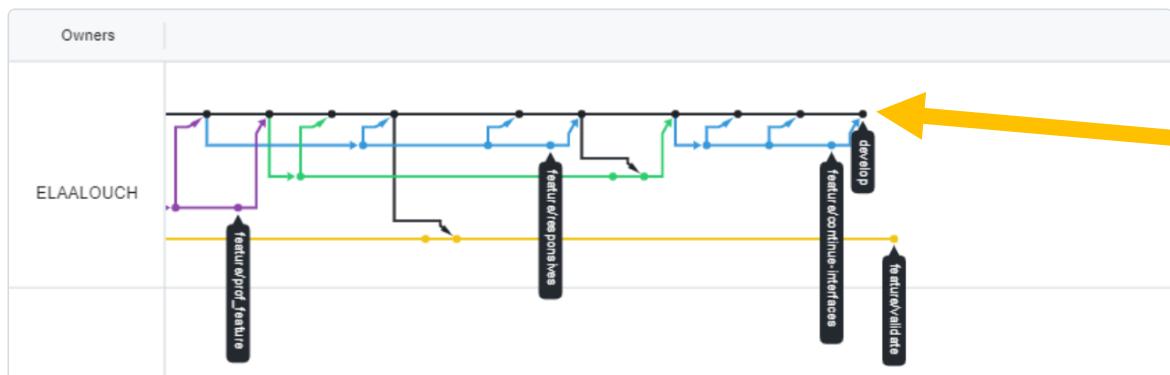


5.2.3. GitFlow

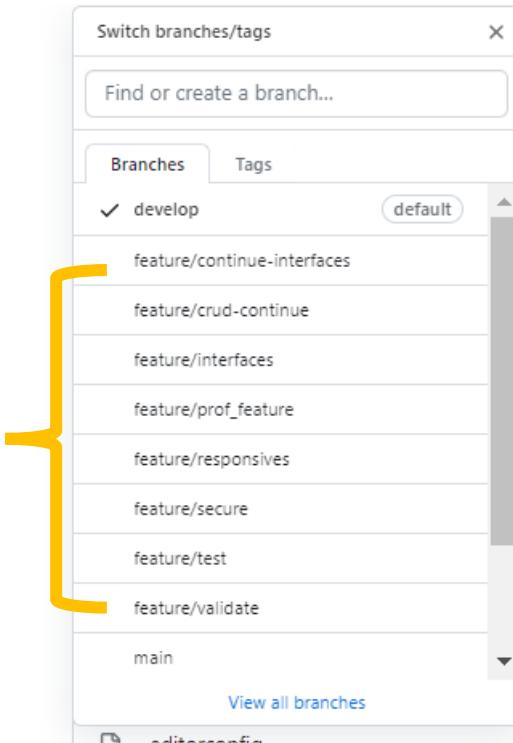
Gitflow est une amélioration du workflow de création de branches de fonctionnalités, introduit par Vincent Driessen en 2010 dans son article "[A successful Git branching model](#)" (un modèle réussi de gestion des branches avec Git). Son objectif principal est de clarifier les rôles des différentes branches et de préciser leurs interactions. Dans notre projet, nous avons décidé d'utiliser ce workflow.



- **Branche "master" (ou "main")** : La branche "master" est la branche principale du projet. Elle contient la version de production stable du code.
- **Branche "develop"** : La branche "develop" est la branche de développement principale. Elle contient les dernières fonctionnalités développées et est souvent considérée comme la branche de travail commune pour l'équipe. Les nouvelles fonctionnalités sont intégrées dans cette branche lorsqu'elles sont suffisamment stables.



- **Branches de fonctionnalités (feature branches)** : Pour chaque nouvelle fonctionnalité ou tâche, une branche de fonctionnalité est créée à partir de la branche "develop". Les développeurs travaillent sur leur propre branche de fonctionnalité isolée pour implémenter une fonctionnalité spécifique. Une fois la fonctionnalité terminée, elle est fusionnée dans la branche "develop".

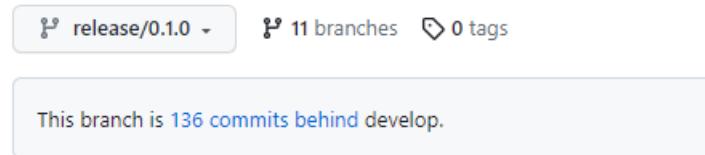


- Branche "release" : La branche "release" est créée lorsque vous préparez à publier une nouvelle version du logiciel. À partir de la branche "develop", une branche "release" est créée pour effectuer les derniers tests, corrections de bugs et préparations avant la publication. Une fois que tout est prêt pour la version, la branche "release" est fusionnée dans la branche "master" et dans la branche "develop". Un tag de version est généralement créé pour marquer la publication.

Le responsable Git peut passer la branche "release" à l'équipe de développement Docker et Postfix... afin qu'ils puissent travailler dessus.

Exemple :

J'ai créé la branche "release" de la version 0.1.0, qui est une branche plus stable que la branche "develop". J'ai partagé la branche avec l'équipe de Docker et Postfix afin qu'ils puissent tester les différentes versions dans un environnement plus fiable.

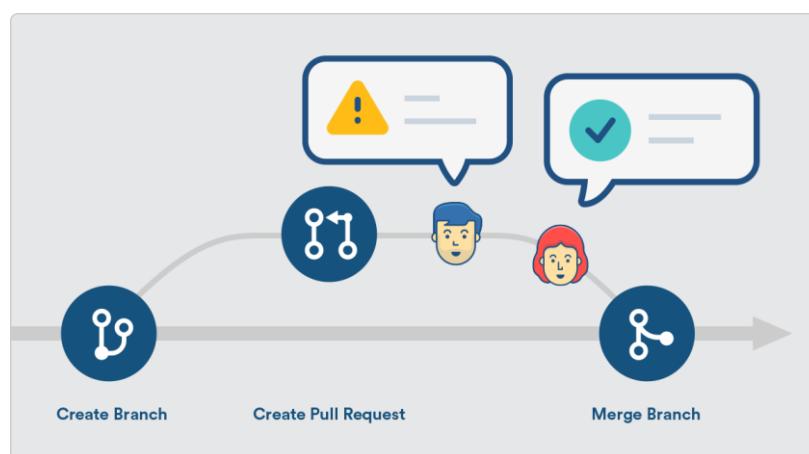


- Branche hotfix : Si un bug est découvert dans la version de production sur la branche "master", une branche de correctif (hotfix branch) est créée à partir de la branche "master". Les correctifs sont apportés sur cette branche, puis ils sont fusionnés à la fois dans la branche "master" et dans la branche "develop". Une fois les correctifs appliqués, la branche "hotfix" est fermée.

5.2.4.

Pull Request

Une Pull Request (PR) est un mécanisme utilisé dans les systèmes de contrôle de version, tels que Git, pour proposer des modifications apportées à un dépôt. Lorsqu'un développeur souhaite apporter des changements à un projet, il crée une branche avec ces modifications et crée une Pull Request pour demander que ces changements soient examinés et fusionnés dans une branche cible, telle que la branche "master" ou "develop".



Les pull requests offrent plusieurs avantages :

- **Révision et feedback** : Les autres contributeurs peuvent examiner vos modifications, apporter des commentaires, poser des questions ou proposer des

améliorations. Cela permet d'améliorer la qualité du code et de favoriser la collaboration.

- **Tests et intégration continue** : Les pull requests peuvent être soumises à des tests automatiques et à des systèmes d'intégration continue. Cela permet de s'assurer que les modifications ne cassent pas le code existant et qu'elles respectent les normes et les bonnes pratiques du projet.
- **Contrôle des modifications** : Les propriétaires du projet ont le contrôle final sur l'acceptation ou le rejet des pull requests. Ils peuvent examiner les modifications, vérifier leur conformité aux exigences du projet et décider de les fusionner ou non.

Pour inciter les développeurs à utiliser les pull requests, j'ai sécurisé les branches "main" et "develop". Ainsi, la seule possibilité qui leur reste est de passer par un pull request et de respecter les normes de Git et GitHub.

Branch protection rules

[Add rule](#)

main	Currently applies to 1 branch	Edit	Delete
develop	Currently applies to 1 branch	Edit	Delete

Protect matching branches

Require a pull request before merging
When enabled, all commits must be made to a non-protected branch and submitted via a pull request before they can be merged into a branch that matches this rule.

Require approvals
When enabled, pull requests targeting a matching branch require a number of approvals and no changes requested before they can be merged.
Required number of approvals before merging: 1 ▾

Dismiss stale pull request approvals when new commits are pushed
New reviewable commits pushed to a matching branch will dismiss pull request review approvals.

Require review from Code Owners
Require an approved review in pull requests including files with a designated code owner.

Require approval of the most recent reviewable push
Whether the most recent reviewable push must be approved by someone other than the person who pushed it.

Dans le projet, il s'est produit des cas où un développeur a travaillé sur la branche "develop" directement et a essayé de pousser ses modifications sans passer par un pull request, le résultat :

```
$ git push
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 4 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (8/8), 937 bytes | 312.00 KiB/s, done.
Total 8 (delta 5), reused 0 (delta 0)  back-reused 0
remote: Resolving deltas: 100% (5/5) Completed with 5 local objects.
remote: error: GH006: Protected branch update failed for refs/heads/develop.
remote: error: At least 1 approving review is required by reviewers with write access.
To https://github.com/ELAALOUCH/XtraDh.git
 ! [remote rejected] develop -> develop (protected branch hook declined)
error: failed to push some refs to 'https://github.com/ELAALOUCH/XtraDh.git'
```

Bref, les pull requests facilitent la révision, le contrôle, la collaboration et la qualité du code. Elles permettent également une intégration plus fluide des modifications, une meilleure gestion des conflits et une documentation complète des changements effectués dans le projet.

The screenshot shows a GitHub pull request interface. At the top, there's a list of commits:

- o ✖ [Duda II fait](#) 5094001
- o ✖ [Conflict](#) 57af695
- o ✖ [nonce validation contre xss](#) 7677040
- o ✖ [roles update](#) 46d577b

Below the commits, a message says: "Add more commits by pushing to the `feature/validate` branch on [ELAALOUCH/XtraDh](#)".

On the left, there's a "Review required" badge with a red circle and a minus sign, indicating that at least one review is needed. To its right is a link to "Add your review".

Further down, there are several status indicators:

- [Some checks haven't completed yet](#) 1 in progress check. A link to "Hide all checks" is to the right.
- [Laravel / laravel-tests \(pull_request\)](#) In progress — This check has started... A link to "Details" is to the right.
- ✖ [Merging is blocked](#) Merging can be performed automatically with 1 approving review.
- [Merge without waiting for requirements to be met \(bypass branch protections\)](#)

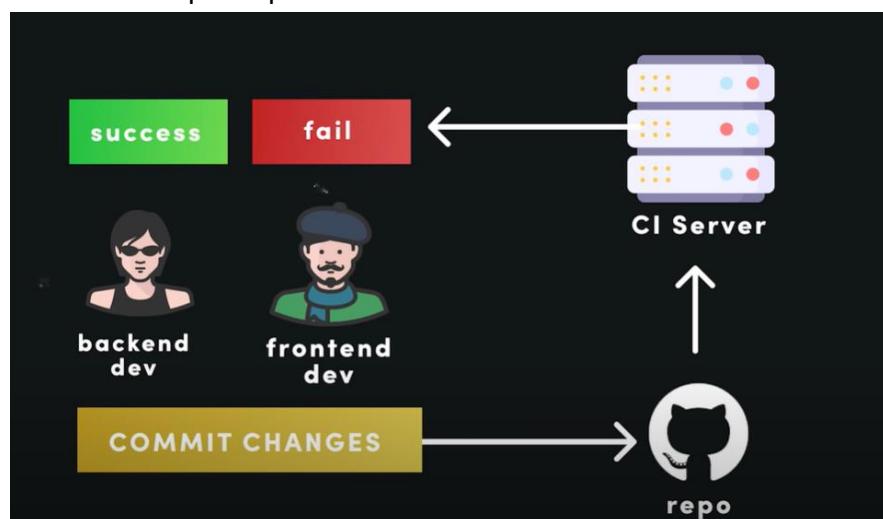
At the bottom, there's a "Merge pull request" button with a dropdown arrow, and a note: "You can also open this in GitHub Desktop or view command line instructions."

Quelques demandes de tirage (pull requests) de notre projet :

<input type="checkbox"/> Clear current search query, filters, and sorts		Author	Label	Projects	Milestones	Reviews	Assignee	Sort
<input type="checkbox"/>	0 Open ✓ 52 Closed							
<input type="checkbox"/>	↳ changement des titres ✓ #28 by sofexbk was merged yesterday • Review required							
<input type="checkbox"/>	↳ Modification en edit de admin univ ✓ #27 by sofexbk was merged yesterday • Review required							
<input type="checkbox"/>	↳ Feature/connect prof vue api ✓ #26 by GlioualYassine was merged yesterday • Review required					1		
<input type="checkbox"/>	↳ modification on editprofile ✓ #25 by sofexbk was merged yesterday • Review required							
<input type="checkbox"/>	↳ finalisation de connexion du front-back end (admin etb) ✓ #24 by GlioualYassine was merged yesterday • Review required					1		
<input type="checkbox"/>	↳ plus de changement ✓ #23 by sofexbk was merged 2 days ago • Review required							
<input type="checkbox"/>	↳ changes of some interfaces ✓ #22 by sofexbk was merged 2 days ago • Review required							
<input type="checkbox"/>	↳ Feature/connect prof vue api ✓ #21 by GlioualYassine was merged 2 days ago • Review required							
<input type="checkbox"/>	↳ modif en admin etab ✓ #20 by sofexbk was merged 4 days ago • Review required					1		
<input type="checkbox"/>	↳ Feature/continue interfaces ✓ #19 by sofexbk was merged 2 weeks ago • Review required							

5.2.5. Intégration continue

L'intégration continue (CI) est une pratique de développement logiciel qui vise à automatiser le processus de vérification et de validation du code source dès qu'une modification est apportée au dépôt de code. Dans le contexte de GitHub, l'intégration continue se réalise en utilisant des outils spécifiques comme les actions .



Lorsqu'un développeur soumet une modification de code à un dépôt GitHub, le

processus d'intégration continue est déclenché.

The screenshot shows a CI pipeline status page. At the top, it says "Triggered via pull request 3 weeks ago" and "Status: Failure". Below that, there's a section for "laravel.yml" with "on: pull_request". Underneath, a box indicates a failure in the "laravel-tests" step, which took 21s and ended with an exit code 2. Annotations show one error and one warning, with a note about deprecated Node.js actions and a link to "Show more".

Voici les principales étapes de ce processus :

- **Construction** : Le système CI récupère le code source depuis le dépôt et le compile, en créant un exécutable ou un package selon les spécifications du projet.
- **Analyse du code** : Des outils d'analyse statique du code peuvent être utilisés pour identifier les problèmes de qualité, les erreurs de syntaxe, les violations des bonnes pratiques de programmation, etc. Ces analyses permettent d'améliorer la lisibilité et la maintenabilité du code.
- **Rapports et notifications** : À la fin du processus d'intégration continue, des rapports sont générés pour informer les développeurs des résultats des tests, de l'état de la construction et de toute autre information pertinente. Des notifications peuvent être envoyées par e-mail, dans un système de messagerie ou dans des outils de gestion de projets .



Dans le cadre de notre projet, nous avons utilisé une action spécifique dans GitHub pour automatiser et exécuter un ensemble d'opérations essentielles.

```
Workflow file for this run
.github/workflows/laravel.yml at 3cd53dc

1  name: Laravel
2
3  on:
4    push:
5      branches: [ "develop","main" ]
6    pull_request:
7      branches: [ "develop","main" ]
8
9  jobs:
10    laravel-tests:
11
12      runs-on: ubuntu-latest
13
14      steps:
15        - uses: shivammathur/setup-php@15c43e89cdef867065b0213be354c2841860869e
16          with:
17            php-version: '8.0'
18        - uses: actions/checkout@v3
19        - name: Copy .env
20          run: php -r "$file_exists('.env') || copy('.env.example', '.env');"
21        - name: Install Dependencies
22          run: composer install -q --noansi --no-interaction --no-scripts --no-progress --prefer-dist
23        - name: Generate key
24          run: php artisan key:generate
25        - name: Directory Permissions
26          run: chmod -R 777 storage bootstrap/cache
27        - name: Create Database
28          run: |
29            mkdir -p database
30            touch database/database.sqlite
31
32
33
34
35
```

Cette action, nommée "laravel-tests", joue un rôle clé dans notre flux de travail de développement. L'action "laravel-tests" est responsable de l'exécution de tests pour notre application Laravel.

Elle est configurée pour s'exécuter sur une machine virtuelle Ubuntu (ubuntu-latest) afin de garantir une compatibilité optimale.

Le processus de l'action "laravel-tests" comprend plusieurs étapes qui décrivent les actions à effectuer.

Tout d'abord, nous utilisons l'action "shivammathur/**setup-php**" pour configurer la version spécifique de PHP requise pour notre application Laravel, en l'occurrence la version 8.0. Ensuite, nous utilisons l'action "actions/checkout" pour récupérer le code source de notre dépôt GitHub, permettant ainsi d'avoir une copie à jour du code pour les tests.

Une étape importante consiste à vérifier si le **fichier .env** est présent. Si ce n'est pas le cas, l'étape "Copy .env" copie le fichier **.env.example** en tant que .env, garantissant ainsi la disponibilité des configurations requises.

Ensuite, l'étape "**Install Dependencies**" utilise la commande composer install pour installer toutes les dépendances nécessaires à notre projet Laravel, en s'assurant que toutes les bibliothèques externes requises sont présentes.

L'étape suivante, "**Generate key**", génère une clé d'application Laravel en exécutant la commande php artisan key:generate, ce qui est essentiel pour le bon fonctionnement de l'application.

Pour s'assurer que les répertoires **storage** et **bootstrap/cache** ont les permissions appropriées, l'étape "**Directory Permissions**" leur attribue des permissions en lecture, écriture et exécution récursives (777), permettant ainsi un accès adéquat pour les opérations de stockage et de mise en cache.

L'étape "**Create Database**" crée un répertoire database et un fichier database.sqlite s'ils n'existent pas déjà.

-> Cette action automatisée effectue diverses tâches telles que la configuration de la version PHP, la récupération du code source, l'installation des dépendances, la génération de la clé d'application Laravel, l'attribution des permissions aux répertoires, et la création de la base de données SQLite . Elle facilite ainsi le processus de développement et de test de l'application Laravel en automatisant ces étapes essentielles.

5.3. Les tâches du responsable Git & GitHub dans le projet.

5.3.1. Formation, Référentiel d'équipe et Scripts Shell pour une collaboration optimisée.

- J'ai organisé une formation d'une durée de 2 heures pour expliquer à mes coéquipiers les notions de Git et GitHub, ainsi que les commandes de base associées.
- Après cette formation, j'ai préparé un référentiel dédié à l'équipe, qui contient toutes les commandes avancées de Git et Gitflow. Ce référentiel constitue une ressource précieuse pour l'équipe, facilitant ainsi la gestion du code source et la collaboration.

```

git push origin :<branche par exemple>
: sert à supprimer

commit + close issue

git commit -m "message , close <#id_issue>"

#GitFlow

initialiser GitFlow

git flow init

creation d'une nouvelle fonctionnalité

git flow feature start <feature name>

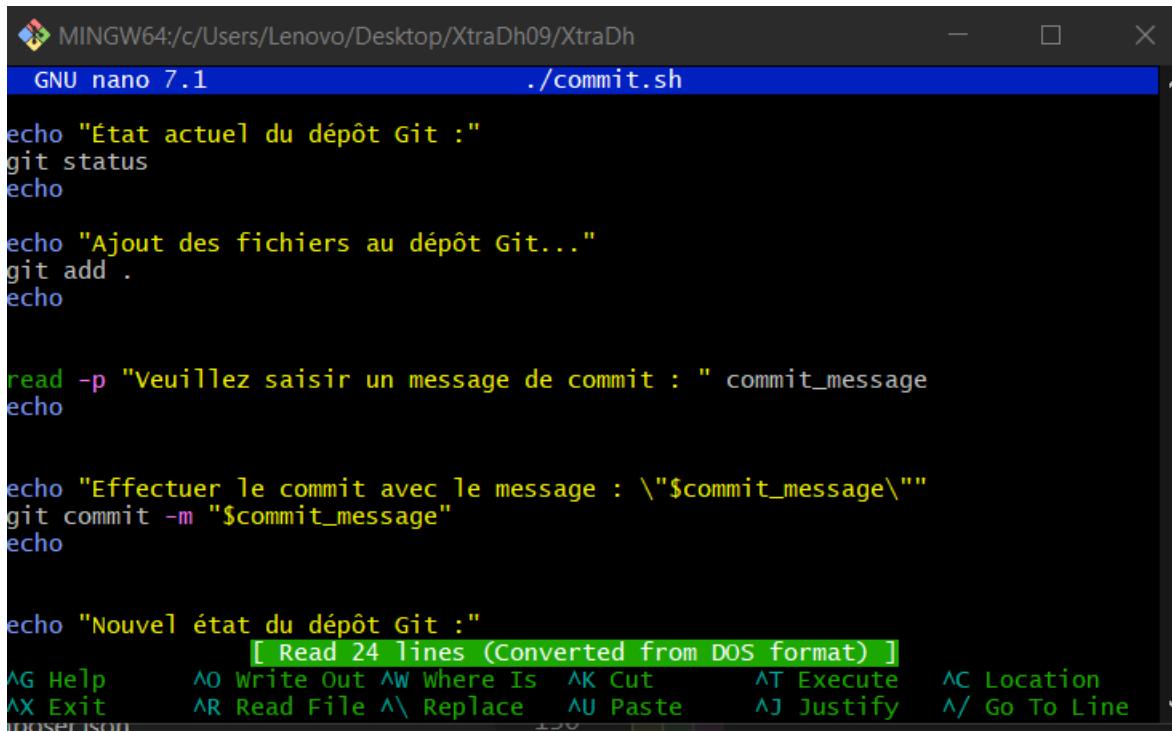
feature finish and merge to develop branch

git flow feature finish <feature name>

```

- Pour simplifier l'utilisation des commandes et résumer les commandes longues, j'ai également préparé des scripts shell pratiques. Ces scripts offrent une interface conviviale pour exécuter les commandes Git couramment utilisées, simplifiant ainsi le processus et améliorant l'efficacité de l'équipe lors des opérations de versionnage et de gestion de branches. Ils ont été conçus de manière à ce que les membres de l'équipe puissent facilement les utiliser, **même s'ils ne sont pas familiers avec les détails techniques de Git**.

Exemple :



```

MINGW64:/c/Users/Lenovo/Desktop/XtraDh09/XtraDh
GNU nano 7.1          ./commit.sh

echo "État actuel du dépôt Git :"
git status
echo

echo "Ajout des fichiers au dépôt Git..."
git add .
echo

read -p "Veuillez saisir un message de commit : " commit_message
echo

echo "Effectuer le commit avec le message : \"$commit_message\""
git commit -m "$commit_message"
echo

echo "Nouvel état du dépôt Git :"
[ Read 24 lines (Converted from DOS format) ]

```

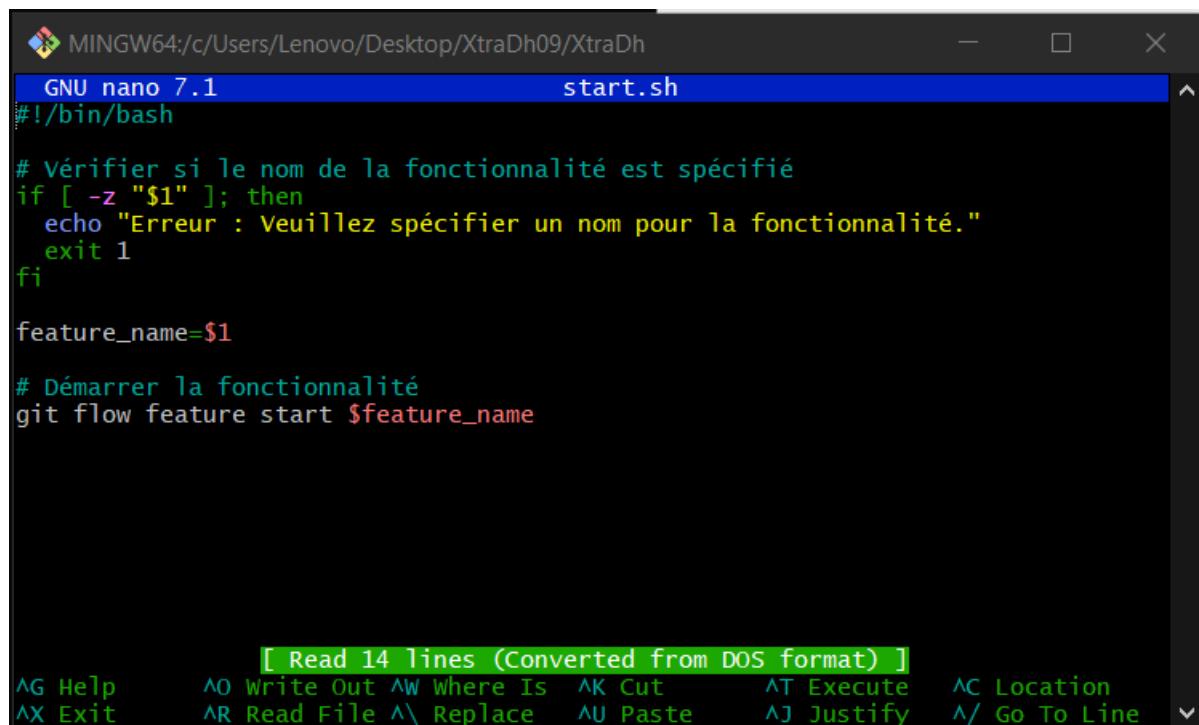
The terminal shows the script's execution. It starts with `git status` showing the initial state. Then it adds all modified files with `git add .`. It then prompts the user for a commit message using `read -p`. After the user provides the message, it performs the commit with `git commit -m`. Finally, it shows the updated repository state with `git status` again.

Ce script est conçu pour faciliter l'exécution de plusieurs commandes Git en une seule étape. Au lieu d'exécuter manuellement trois commandes distinctes, ce script permet d'utiliser une seule commande, à savoir ./commit.sh.

Le script effectue les actions suivantes :

- Affiche l'état actuel du dépôt Git en utilisant la commande git status.
- Ajoute tous les fichiers modifiés au dépôt Git en utilisant la commande git add ..
- Demande à l'utilisateur de saisir un message de commit.
- Effectue un commit avec le message saisi à l'aide de la commande git commit -m "\$commit_message".
- Affiche le nouvel état du dépôt Git.

Ainsi, l'utilisation de ce script permet de gagner du temps et de simplifier le processus de commit en regroupant plusieurs commandes en une seule. Cela rend le workflow plus efficace **et réduit les risques d'erreurs lors de l'exécution manuelle des commandes individuelles**. Pour l'utiliser, il suffit de lancer le script ./commit.sh, puis de suivre les instructions affichées à l'écran pour saisir un message de commit. Une fois le commit effectué, le script affichera le nouvel état du dépôt Git.



```

MINGW64:/c/Users/Lenovo/Desktop/XtraDh09/XtraDh
GNU nano 7.1          start.sh
#!/bin/bash

# Vérifier si le nom de la fonctionnalité est spécifié
if [ -z "$1" ]; then
    echo "Erreur : Veuillez spécifier un nom pour la fonctionnalité."
    exit 1
fi

feature_name=$1

# Démarrer la fonctionnalité
git flow feature start $feature_name

[ Read 14 Lines (Converted from DOS format) ]
^G Help      ^O Write Out ^W Where Is  ^K Cut      ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line

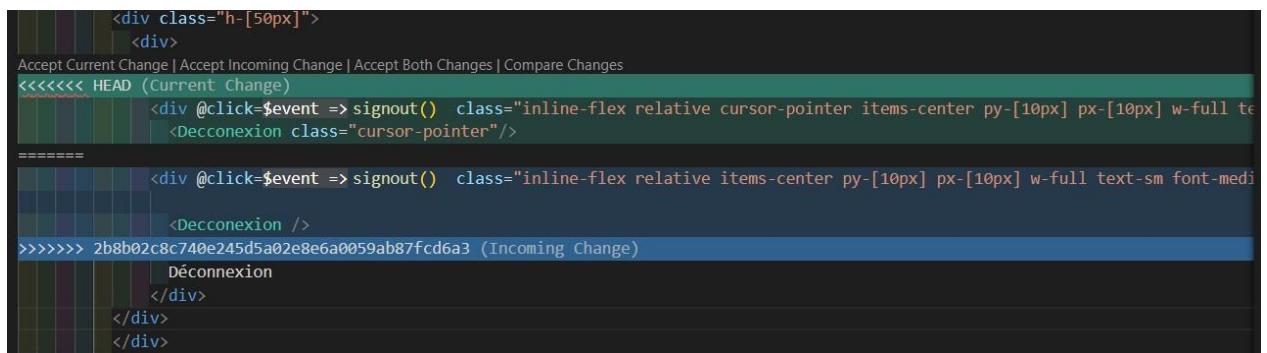
```

Ce script est conçu pour faciliter le démarrage d'une fonctionnalité avec Gitflow.

git flow feature start Myfeature → *./start.sh Myfeature*

5.3.2. Résolution efficace des conflits

En tant que responsable de Git et Github , j'ai joué un rôle clé dans la gestion des conflits rencontrés lors des demandes de fusion et des conflits de pull des développeurs . Ces conflits peuvent survenir lorsque plusieurs développeurs travaillent sur la même partie du code et tentent de fusionner leurs modifications dans la branche principale du projet(develop).



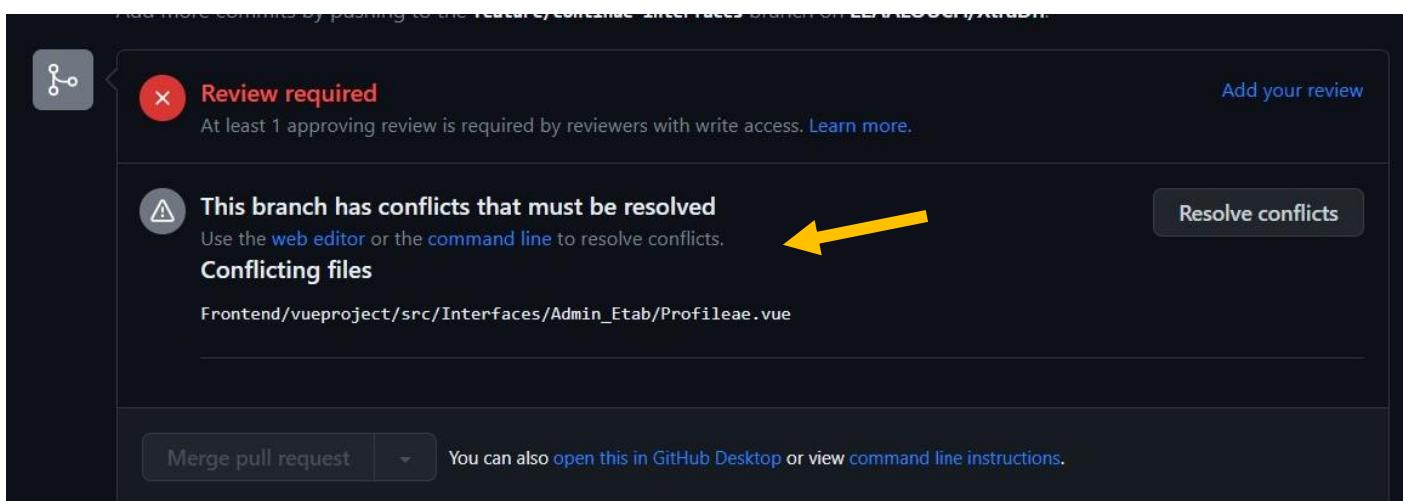
```

<div class="h-[50px]>
  <div>
    Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
    <<<<< HEAD (Current Change)
    <div @click=$event => signout() class="inline-flex relative cursor-pointer items-center py-[10px] px-[10px] w-full text-sm font-medium">
      <Decconexion class="cursor-pointer"/>
    </div>
    =====
    <div @click=$event => signout() class="inline-flex relative items-center py-[10px] px-[10px] w-full text-sm font-medium">
      <Deconexion />
    </div>
    >>>>> 2b8b02c8c740e245d5a02e8e6a0059ab87fcfd6a3 (Incoming Change)
    Déconnexion
    </div>
  </div>
</div>

```

Ma responsabilité principale était de comprendre la nature des conflits et de les résoudre de manière efficace. Cela impliquait **une connaissance de Laravel et Vue.js, les technologies utilisées dans le projet, ainsi qu'une compréhension des modifications apportées par les développeurs.**

Lorsqu'un conflit de pull request se produisait, j'examinais attentivement les différences entre les modifications apportées par les développeurs et la branche principale. Je comparais les lignes de code modifiées, les ajouts et les suppressions, et je cherchais des incohérences ou des conflits potentiels.



Ensuite, j'entreprendais les actions nécessaires pour résoudre ces conflits. Cela pouvait impliquer la modification manuelle du code, la fusion sélective des modifications ou **la communication avec les développeurs concernés** pour clarifier leurs intentions et parvenir à une solution appropriée.

La résolution des conflits de pull requests nécessitait une grande attention aux détails et une capacité à travailler de **manière collaborative**. Je devais m'assurer que les modifications apportées par les développeurs étaient cohérentes avec la vision globale du projet et qu'elles ne compromettaient pas l'intégrité du code existant.

```

1  <template>
2  <><<< feature/interfaces
3  <div class="flex flex-col md:flex-row h-screen items-center">
4
5  <div class="bg-blue-600 hidden lg:block w-full md:w-1/2 xl:w-2/3 h-screen">
6    
7  </div>
8
9  <div class="bg-white w-full md:max-w-md lg:max-w-full md:mx-auto md:mx-0 md:w-1/2 xl:w-1/3 h-screen px-6 lg:px-16 xl:px-12 flex items-center justify-center">
10   <div class="w-full h-100 ">
11
12   <div class="flex justify-center">
13     
14   </div>
15
16   <form @submit.prevent="submitLogin()" class="mt-6" >
17     <div>
18
19   <div class="flex flex-col md:flex-row h-screen items-center">
20     <div class="bg-blue-600 hidden lg:block w-full md:w-1/2 xl:w-2/3 h-screen">
21       
22     </div>
23
24     <div class="bg-white w-full md:max-w-md lg:max-w-full md:mx-auto md:mx-0 md:w-1/2 xl:w-1/3 h-screen px-6 lg:px-16 xl:px-12 flex items-center justify-cent
25       <div class="w-full h-100">
26         <h1 class="text-sky-600 text-6xl text-center">XtraDh</h1>
27
28         <form @submit.prevent="submitLogin()" class="mt-6">
29           <div>
30             >>>> develop
31             <label for="email" class="block text-gray-700">Email Address</label>
32             <input v-model="user.email" type="email" id="email" placeholder="Enter Email Address" class="w-full px-4 py-3 rounded-lg bg-gray-200 mt-2 border focu
33           </div>
34
35             <div class="mt-4">
36               <label for="password" class="block text-gray-700">Password</label>
37             <div class="relative">

```

En fin de compte, mon rôle consistait à garantir que les conflits de pull requests étaient résolus de manière satisfaisante, permettant ainsi une intégration harmonieuse des modifications dans le projet global. Cela contribuait à maintenir la qualité du code, à assurer la stabilité du projet et à favoriser une collaboration efficace entre les membres de l'équipe de développement.

5.3.3. Gestion proactive des problèmes de code source

En tant que responsable de la gestion du code source , j'ai été chargé de résoudre divers problèmes liés à la structure et à l'intégrité du code.

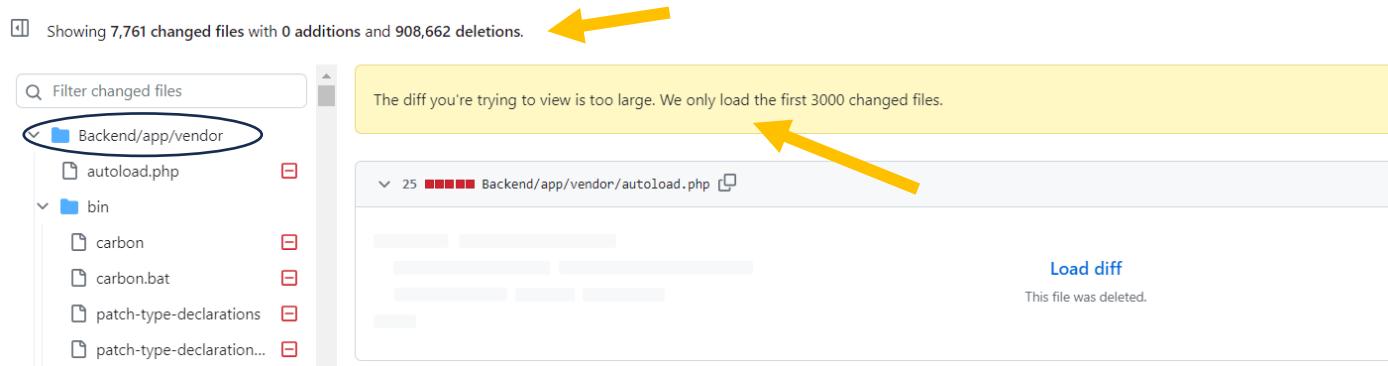
L'un des exemples notables auxquels j'ai été confronté était lorsque l'un des développeurs a modifié l'arborescence des fichiers "**vendor**", qui était initialement ignorée par Git à l'aide du fichier ".gitignore".

```

◆ .gitignore
1   /node_modules
2   /public/hot
3   /public/storage
4   /storage/*.key
5   /vendor
6   .env ←
7   .env.backup
8   .phpunit.result.cache
9   docker-compose.override.yml
10  Homestead.json
11  Homestead.yaml
12  npm-debug.log
13  yarn-error.log
14  /.idea
15  /.vscode
16

```

Ce changement dans l'arborescence des fichiers "vendor" a créé des problèmes d'hébergement, car les fichiers du répertoire "vendor" sont généralement des dépendances externes téléchargées à partir de sources tierces. Ils sont volumineux et souvent régulièrement mis à jour par les fournisseurs.



Après avoir identifié le problème, j'ai travaillé en collaboration avec le développeur concerné pour comprendre les raisons de ce changement. J'ai ensuite expliqué les meilleures pratiques de gestion des dépendances et les raisons pour lesquelles le répertoire "vendor" ne devrait pas être inclus dans le référentiel Git.

6. CHAPITRE 6 :

PGSQL

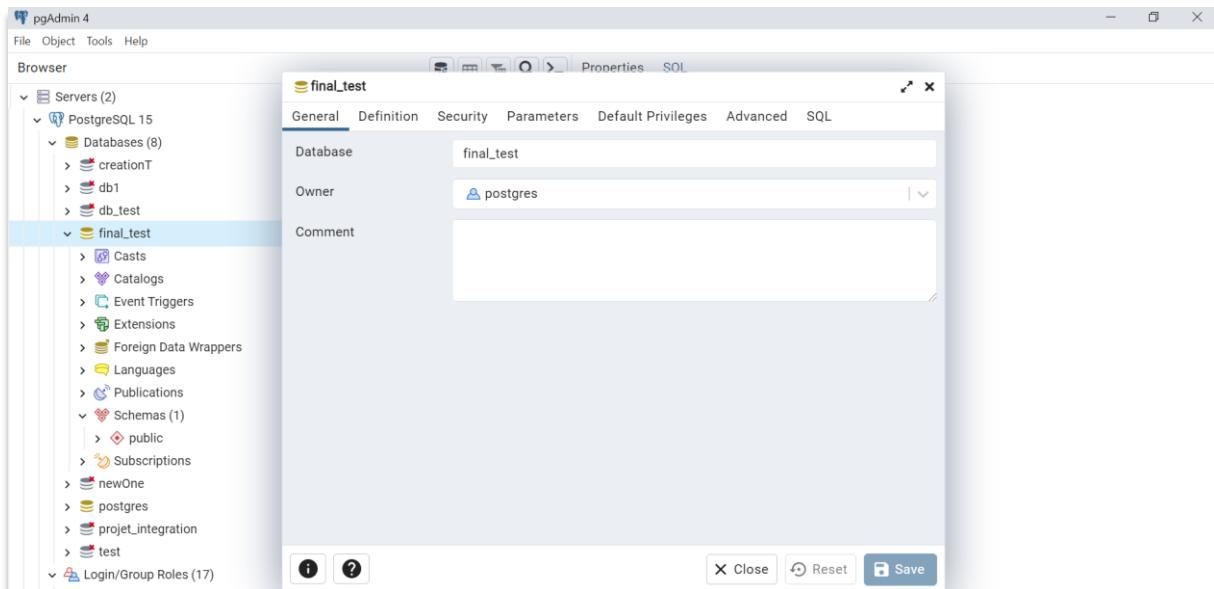
Le présent chapitre vise à concevoir et développer des procédures stockées, des fonctions et des déclencheurs en PGSQl pour gérer les données des vacataires, des plannings et des paiements. Il mettra également en œuvre des mécanismes de sécurité pour protéger les données stockées dans la base de données.

6.1. La préparation de l'environnement de travail

6.1.1. Base de Données

En utilisant l'interface graphique PgAdmin4 de PostgreSQL, nous avons créé une base de données afin de réaliser des tests de migration des tables, des fonctions, des procédures stockées et des déclencheurs.

Nous avons nommé notre base de données "final_test" par le superutilisateur "postgres".



La configuration du fichier ".env" de laravel :

Nous avons configuré les informations de la base de données via le fichier ".env" de notre projet.

```
DB_CONNECTION=pgsql
DB_HOST= localhost
DB_PORT=5432
DB_DATABASE=final_test
DB_USERNAME=postgres
DB_PASSWORD=houda
```

6.1.2. Les tables

Nous avons utilisé deux méthodes pour créer les tables dans notre projet. La première méthode consistait à utiliser les fichiers de migration. Ces fichiers nous ont permis de définir la structure des tables et les modifications à apporter à la base de données. En exécutant ces fichiers, les tables ont été créées automatiquement.

La deuxième méthode que nous avons utilisée était d'écrire un script SQL pour le rejoindre dans le Docker. Nous avons rédigé le script SQL en détaillant les instructions nécessaires pour créer les tables spécifiques à notre projet. En exécutant ce script dans le Docker, les tables ont été créées avec succès.

Exemple :

- Fichier de migration

```
public function up()
{
    Schema::create('users', function (Blueprint $table) {
        $table->id('id_user');
        $table->string('email')->unique();
        $table->string('password');
        $table->string('type');
        $table->rememberToken();
        $table->timestamps();
    });
}
```

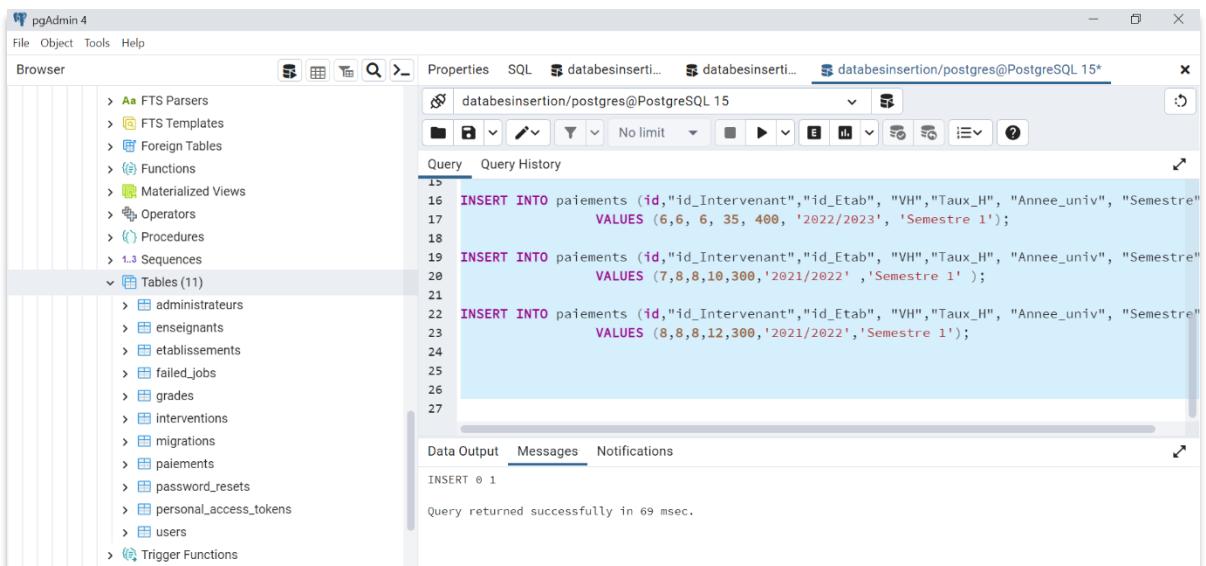
- Script SQL

```
CREATE TABLE IF NOT EXISTS public.users
(
    id_user BIGSERIAL NOT NULL,
    email character varying(191) COLLATE pg_catalog."default" NOT NULL,
    password character varying(191) COLLATE pg_catalog."default" NOT NULL,
    type character varying(191) COLLATE pg_catalog."default" NOT NULL,
    remember_token character varying(100) COLLATE pg_catalog."default",
    created_at timestamp(0) without time zone,
    updated_at timestamp(0) without time zone,
    CONSTRAINT users_pkey PRIMARY KEY (id_user),
    CONSTRAINT users_email_unique UNIQUE (email)
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public.users
OWNER to postgres;
```

- En PgAdmin4



Remarque :

Pour calculer les valeurs de "Salaire_Net" et "Salaire_Brut" dans la table "paiements", nous avons utilisé la fonctionnalité de colonne calculée de PostgreSQL. Cette fonctionnalité nous permet de définir une expression ou une formule pour calculer dynamiquement la valeur d'une colonne en fonction des autres colonnes de la même table.

Voici un exemple illustrant l'utilisation de colonnes calculées dans notre projet, en utilisant :

- Fichier de migration

```

Schema::create('paiements', function (Blueprint $table) {
    $table->id();
    $table->unsignedBigInteger('id_Intervenant');
    $table->foreign('id_Intervenant')->references('id')->on('enseignants')->onDelete('cascade');
    $table->unsignedBigInteger('id_Etab');
    $table->foreign('id_Etab')->references('id')->on('etablissements');
    $table->float('VH')->default(0);
    $table->float('Taux_H');
    $table->float('Brut')->nullable()->storedAs(' VH * Taux_H ');
    $table->float('IR')->default(0.38);
    $table->float('NET')->nullable()->storedAs(' VH * Taux_H * ( 1 - IR ) ');
    $table->string('Annee_univ',10);
    $table->string('Semestre');
    $table->timestamps();
});


```

- Le script SQL

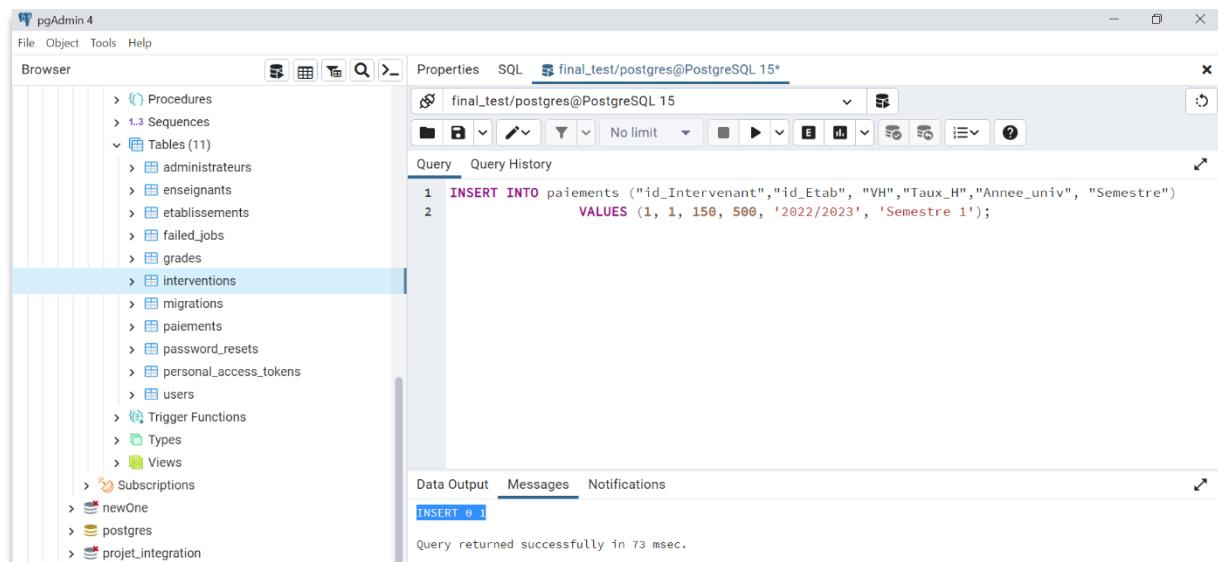
```

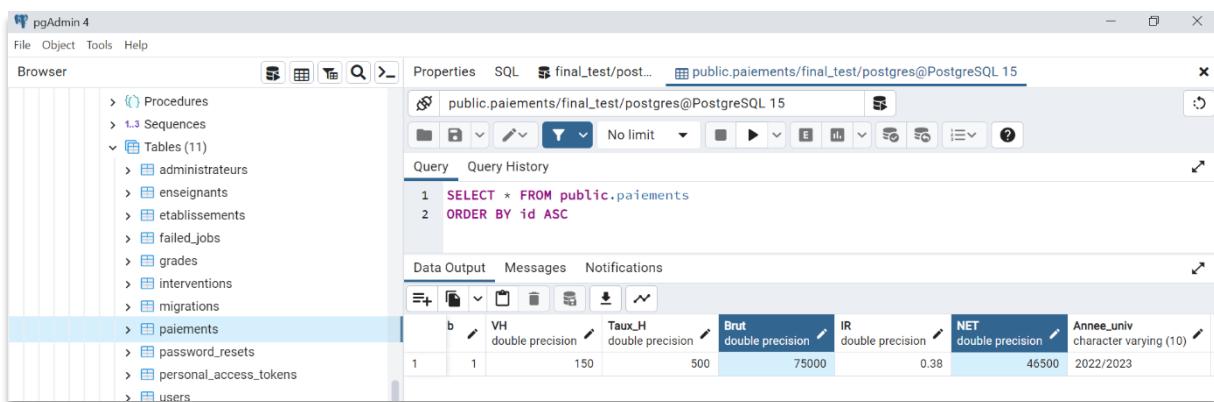
CREATE TABLE IF NOT EXISTS public.paiements
(
    id BIGSERIAL NOT NULL,
    "id_Intervenant" bigint NOT NULL,
    "id_Etab" bigint NOT NULL,
    "VH" double precision NOT NULL DEFAULT '0'::double precision,
    "Taux_H" double precision NOT NULL,
    "Brut" double precision GENERATED ALWAYS AS ((VH * Taux_H)) STORED,
    "IR" double precision NOT NULL DEFAULT '0.38'::double precision,
    "NET" double precision GENERATED ALWAYS AS ((VH * Taux_H) * ((1)::double precision - IR))) STORED,
    "Annee_univ" character varying(10) COLLATE pg_catalog."default" NOT NULL,
    "Semestre" character varying(191) COLLATE pg_catalog."default" NOT NULL,
    created_at timestamp(0) without time zone,
    updated_at timestamp(0) without time zone,
    CONSTRAINT paiements_pkey PRIMARY KEY (id),
    CONSTRAINT paiements_id_etab_foreign FOREIGN KEY ("id_Etab")
        REFERENCES public.etablissements (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT paiements_id_intervenant_foreign FOREIGN KEY ("id_Intervenant")
        REFERENCES public.enseignants (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE CASCADE
)
TABLESPACE pg_default;

ALTER TABLE IF EXISTS public.paiements
OWNER to postgres;
}

```

- Test d'insertion dans la table "paiements" :





6.2. L'insertion des données

Nous avons utilisé deux méthodes différentes pour insérer des données dans notre base de données.

6.2.1. Requêtes SQL

La première méthode consistait à utiliser des requêtes SQL de type INSERT. Nous avons rédigé des requêtes INSERT pour chaque table afin d'ajouter les enregistrements souhaités. En exécutant ces requêtes, les données ont été insérées dans la base de données.

```
INSERT INTO users (email, password, type) VALUES ('ELMOUMNI.Bouchta@uae.ac.ma', 'GKVK0q2ZGR', 'president_univ');
```

	id_user [PK] bigint	email character varying (191)	password character varying (191)	type character varying (191)	remember_token character varying (100)	created_at timestamp without time zone	updated_at timestamp with time zone
1	1	ELMOUMNI.Bouchta@uae.ac.ma	GKVK0q2ZGR	president_univ	[null]	[null]	[null]
2	2	ROUDIES.Anas@uae.ac.ma	pM88U1P02tK	admin_univ	[null]	[null]	[null]
3	3	Zahiri.Hicham@uae.ac.ma	AudBqv	admin_etb	[null]	[null]	[null]
4	4	ZOUITA.ABDESALAM@uae.ac.ma	7emyITNsq3D	admin_etb	[null]	[null]	[null]
5	5	GUESSOUS.JENNATE@uae.ac.ma	TmWxa1veFzXi	admin_etb	[null]	[null]	[null]
6	6	MASROUR.AICHA@uae.ac.ma	t1DLOn0zshsF	admin_etb	[null]	[null]	[null]
7	7	MHAMEDI.OMAR@uae.ac.ma	IgOfIWtg9	admin_etb	[null]	[null]	[null]
8	8	ABADAY.ABELAAZIZ@uae.ac.ma	JC471776	admin_etb	[null]	[null]	[null]
9	9	ABAHAZIM.LARABASS@uae.ac.ma	JF43247	admin_etb	[null]	[null]	[null]
10	10	ABBASSI.MOHAMEDALI@uae.ac.ma	SH148684	admin_etb	[null]	[null]	[null]
11	11	DRISSI.Asma@uae.ac.ma	uVjiVOGcw5g	directeur_etb	[null]	[null]	[null]
12	12	DERRAZ.Malika@uae.ac.ma	UZ0QSp	directeur_etb	[null]	[null]	[null]
13	13	BELHADJ.Youssef@uae.ac.ma	Dlx0taebL	directeur_etb	[null]	[null]	[null]
14	14	LAAROUSSI.Hamid@uae.ac.ma	3mETMmk	directeur_etb	[null]	[null]	[null]
15	15	MOHAMED.CHEINE_Sidi@uae.ac.ma	Gnfl3vXft1	directeur_etb	[null]	[null]	[null]

6.2.2. Postman

La deuxième méthode que nous avons utilisée était d'utiliser Postman. Postman est un outil de développement d'API qui nous a permis de tester et d'envoyer des requêtes HTTP à notre

application. Nous avons utilisé Postman pour envoyer des requêtes POST contenant les données à insérer dans la base de données. Nous avons également profité de sa fonctionnalité pour générer des mots de passe hachés. Postman nous a fourni des outils pour effectuer le hachage des mots de passe, ce qui nous a permis de stocker les mots de passe de manière sécurisée dans la base de données.

The screenshot shows the Postman application interface. At the top, there's a navigation bar with 'New', 'Import', 'Overview', 'POST add' (which is highlighted), 'User', 'GET New Request', and 'No Environment'. On the left, there's a sidebar with 'Collections' (selected), 'Environments', and 'History'. The main area shows a 'New Collection / add' section with a 'POST' method and URL 'http://127.0.0.1:8000/api/User'. Below this, under 'Body', the 'x-www-form-urlencoded' tab is selected, and the JSON body is defined as follows:

```

{
  "user": {
    "type": "admin_etb",
    "email": "STEHSRTHRTH.Houda@uae.ac.ma",
    "updated_at": "2023-06-11T08:17:21.000000Z",
    "created_at": "2023-06-11T08:17:21.000000Z",
    "id_user": 45
  },
  "token": "1|EjEEia5q9hIF0t7Ku78ix7R9HUu90vP0e8VtDZ36"
}

```

Below the body, the 'Response' tab is visible. The response status is '202 Accepted' with a time of '4.28 s' and a size of '799 B'. The response body is shown in JSON format:

```

{
  "user": {
    "type": "admin_etb",
    "email": "STEHSRTHRTH.Houda@uae.ac.ma",
    "updated_at": "2023-06-11T08:17:21.000000Z",
    "created_at": "2023-06-11T08:17:21.000000Z",
    "id_user": 45
  },
  "token": "1|EjEEia5q9hIF0t7Ku78ix7R9HUu90vP0e8VtDZ36"
}

```

At the bottom, a table shows the inserted data:

	44	45 STEHSRTHRTH.Houda@uae.ac.ma	\$2y\$10\$OL34/KCXS.6EyHh/G3zXqusEDjQ0UG.jMcV7A5tEc1KuquoXr1...	admin_etb
44	45	STEHSRTHRTH.Houda@uae.ac.ma	\$2y\$10\$OL34/KCXS.6EyHh/G3zXqusEDjQ0UG.jMcV7A5tEc1KuquoXr1...	admin_etb

6.3. Les déclencheurs

La syntaxe des triggers en PostgreSQL diffère de celle de PL/SQL. En PostgreSQL, pour définir un trigger, on crée d'abord une fonction spécifique - "Trigger functions" - qui contient le traitement à effectuer lorsque le trigger est déclenché. Ensuite, on associe cette fonction à un nom et à un événement particulier à l'aide d'un code séparé.

Pour créer un trigger en PostgreSQL peut être réalisée en suivant les étapes suivantes :

- **On définit la fonction de déclenchement (Trigger function) :**

Cette fonction contient le code qui sera exécuté lorsque le trigger est activé. On peut y inclure

des opérations de manipulation des données, des requêtes supplémentaires, des mises à jour, etc.

Par exemple :

```
● ● ●

CREATE OR REPLACE FUNCTION nom_de_la_fonction_de_trigger()
RETURNS TRIGGER AS $$ 
BEGIN
    RETURN NEW;
END;
$$ LANGUAGE plpgsql
FUNCTION AjoutEnseignantFunction();
```

Le "**RETURN NEW**" dans un trigger PostgreSQL permet de renvoyer la nouvelle ligne qui a déclenché le trigger.

- **On définit le trigger lui-même :**

Dans cette étape, on attribue un nom et un événement au trigger, puis on spécifie le moment où le trigger sera déclenché (avant, après, etc.) et la table concernée. On utilise la clause CREATE TRIGGER pour cela.

Par exemple :

```
● ● ●

CREATE TRIGGER nom_du_trigger
AFTER INSERT
ON nom_de_la_table
FOR EACH ROW
EXECUTE FUNCTION nom_de_la_fonction_de_trigger();
```

6.3.1. Le déclencheur : "AjouterEnseignantTrigger"

```

CREATE OR REPLACE FUNCTION AjoutEnseignantFunction()
RETURNS TRIGGER
AS
$$
BEGIN
    IF (TG_OP = 'INSERT') THEN
        UPDATE Etablissements
        SET "Nbr_enseignants" = "Nbr_enseignants" + 1
        WHERE id = NEW."Etablissement";

    ELSIF (TG_OP = 'UPDATE' )THEN
        IF ( NEW."Etablissement" <> OLD."Etablissement" ) THEN
            UPDATE Etablissements
            SET "Nbr_enseignants" = "Nbr_enseignants" + 1
            WHERE id = NEW."Etablissement";

            UPDATE Etablissements
            SET "Nbr_enseignants" = "Nbr_enseignants" - 1
            WHERE id = OLD."Etablissement";

        END IF;
    ELSE
        UPDATE Etablissements
        SET "Nbr_enseignants" = "Nbr_enseignants" - 1
        WHERE id = OLD."Etablissement";
    END IF;

    return NEW;

END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER AjoutEnseignantTrigger
AFTER INSERT OR UPDATE OF "Etablissement" OR DELETE
ON enseignants
FOR EACH ROW
EXECUTE FUNCTION AjoutEnseignantFunction();

```

Dans le cadre de notre projet, nous avons mis en place un déclencheur dans PostgreSQL pour gérer la mise à jour automatique de la colonne "Nbr_enseignants" dans les établissements de l'Université Abdmalek Saadi.

Ce déclencheur a été conçu pour répondre à plusieurs scénarios. Tout d'abord, lorsqu'un nouvel enseignant est inséré dans la base de données, le déclencheur est activé et met à jour la valeur de Nbr_enseignants de l'établissement correspondant en l'incrémentant de 1.

Ensuite, si un enseignant change d'établissement au sein de notre université, le déclencheur détecte cette modification et effectue les actions appropriées. Il diminue la valeur de Nbr_enseignants de l'établissement d'origine de 1, puis l'incrémente de 1 dans le nouvel établissement de destination.

Enfin, lorsque nous supprimons un enseignant d'un établissement, le déclencheur est également déclenché et met à jour la valeur de Nbr_enseignants de l'établissement concerné en la diminuant de 1.

6.3.2.

Le déclencheur :"AjoutPaimenttrigger"

```

CREATE OR REPLACE FUNCTION AjoutFunctiontrigger()
RETURNS TRIGGER
AS
$$
declare
    nbr_h double precision ;
    charge_S double precision;
    dif double precision;
    h_s double precision;
BEGIN

    charge_S = charge_S(NEW."id_Intervenant");
    nbr_h = Charge_Prof(NEW."id_Intervenant",NEW."id_Etab", NEW."Annee_univ");
    h_s = PaiementH(NEW."id_Intervenant");
    IF Type_Invervention(NEW."id_Intervenant",NEW."id_Etab") = 1 THEN --origine
        dif = nbr_h - charge_S;
        if dif > 0 then
            CALL Ajouter_Paiment(NEW."id_Intervenant",NEW."id_Etab",dif,charge_S,NEW."Annee_univ",
            NEW."Semestre");
        END IF ;
    ELSE -- vacation
        CALL Ajouter_Paiment(NEW."id_Intervenant",NEW."id_Etab",nbr_h,charge_S,NEW."Annee_univ",NEW."Semestre");
    END IF;

    RETURN NEW;
END; $$LANGUAGE plpgsql;
CREATE TRIGGER AjoutPaimenttrigger
AFTER UPDATE OF visa_uea ON interventions
FOR EACH ROW
WHEN (NEW.visa_uea = 1)
EXECUTE FUNCTION AjoutFunctiontrigger();

```

Ce déclencheur remplit automatiquement la table "paiements" en fonction de certaines conditions. Plus précisément, lorsque les valeurs "visa_etb" et "visa_uea" sont toutes les deux égales à 1 dans la table "interventions", ce qui signifie que l'intervention a été validée à la fois par le président de l'université et le directeur de l'établissement où l'enseignant effectue son intervention.

Pour réaliser cette tâche, nous avons utilisé plusieurs fonctions et procédures. Voici une description de chacune d'entre elles :

1. La fonction "charge_S" : permet de retourner la charge statutaire de l'enseignant en fonction de son identifiant (id).

2. La fonction "Charge_Prof" : retourne le nombre d'heures des interventions validées par cet enseignant dans l'établissement spécifié pour l'année en cours.

3. La fonction "PaiementH" : permet de retourner la valeur du taux horaire en fonction du grade de l'enseignant.

4. La fonction "Type_Intervention" : retourne 1 ou 2, où 1 indique que l'intervention a lieu dans l'établissement d'origine de l'enseignant et 2 indique que l'intervention a lieu dans un autre établissement. Cette fonction est utilisée pour déterminer comment traiter les heures supplémentaires dans la table "paiements".

5. La procédure "Ajouter_Paiement" : gère l'insertion ou la mise à jour des enregistrements dans la table "paiements" en fonction de différents cas. Les cas spécifiques sont traités dans la partie des " Fonctions & Procédures stockées ".

Pour une intervention dans l'établissement d'origine (valeur de retour égale à 1), nous vérifions si l'enseignant a dépassé sa charge statutaire dans cet établissement pour l'année en cours. Si c'est le cas, les heures supplémentaires (différence entre le nombre d'heures validées et la charge statutaire) doivent être insérées dans la table "paiement". Sinon, aucune action n'est requise.

Pour une intervention dans un autre établissement (valeur de retour égale à 2), toutes les heures sont considérées comme des heures supplémentaires et doivent être insérées dans la table "paiement".

6.3.3. Trigger 3 : "ModifierPaimenttrigger"

```

CREATE OR REPLACE FUNCTION Modifiertrigger()
RETURNS TRIGGER
AS
$$
declare
    nbr_h double precision ;
    charge_S double precision;
    res  double precision;

BEGIN
    nbr_h = Charge_Prof(OLD."id_Intervenant",OLD."id_Etab", NEW."Annee_univ");
    IF Type_Intervention(OLD."id_Intervenant",OLD."id_Etab") = 1 THEN --origine
        charge_S = charge_S(OLD."id_Intervenant");
        if nbr_h > charge_S then
            res = nbr_h - charge_S ;
            CALL Update_Paiment(OLD."id_Intervenant",OLD."id_Etab",res, NEW."Annee_univ");
        else
            CALL Delete_Paiment(OLD."id_Intervenant",OLD."id_Etab", NEW."Annee_univ" );
        END IF ;
        ELSE
            -- vacation
            if nbr_h > 0 then
                CALL Update_Paiment(OLD."id_Intervenant",OLD."id_Etab", nbr_h, NEW."Annee_univ");
            else
                CALL Delete_Paiment(OLD."id_Intervenant",OLD."id_Etab", NEW."Annee_univ");
            END IF ;
        END IF;
    RETURN NEW;
END; $$LANGUAGE plpgsql;

CREATE TRIGGER ModifierPaimenttrigger
AFTER UPDATE OF visa_uea OR DELETE ON interventions
FOR EACH ROW
    WHEN (OLD.visa_uea = 1)
    EXECUTE FUNCTION Modifiertrigger();

```

Ce déclencheur est également activé lorsque qu'un enseignant annule son intervention, c'est-à-dire lorsque les nouvelles valeurs de "visa_etb" et "visa_uea" sont toutes les deux égales à 0, ou lorsqu'une intervention est supprimée.

Pour gérer ces cas, nous avons utilisé deux fonctions déjà mentionnées dans la partie précédente, à savoir "Charge_Prof" et "Type_Intervention". Ces fonctions nous permettent d'obtenir des informations sur la charge de travail de l'enseignant et le type d'intervention.

De plus, nous avons créé deux procédures pour simplifier le code du déclencheur et le rendre

plus lisible :

1. La procédure "Update_Paiement" : est utilisée pour mettre à jour la nouvelle valeur du nombre d'heures de cet enseignant dans cet établissement pour l'année concernée par cette intervention. Cependant, cela se fait uniquement si l'enseignant a encore des heures supplémentaires dans cet établissement provenant d'autres interventions effectuées précédemment.

2. La procédure "Delete_Paiement" : est utilisée pour supprimer l'enregistrement dans la table "paiements" qui concerne une intervention dans un établissement où aucune heure supplémentaire n'existe en raison d'une suppression ou d'une annulation d'une intervention. Cette procédure permet de gérer la suppression des enregistrements inutiles dans la table "paiements".

6.3.4. Trigger 4 : "VisaTrigger"

```

CREATE OR REPLACE FUNCTION VisaTriggerFunction()
RETURNS TRIGGER
AS
$$
BEGIN
    update interventions
    set visa_uea = 0
    where "id_Intervenant" = OLD."id_Intervenant" and "id_Etab" = OLD."id_Etab" and "Annee_univ" =
OLD."Annee_univ";
    return NEW ;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER VisaTrigger
AFTER UPDATE of visa_etb
ON Interventions
FOR EACH ROW
WHEN (NEW.visa_etb = 0)
EXECUTE FUNCTION VisaTriggerFunction();

```

Nous avons mis en place un déclencheur qui automatise le traitement de l'annulation d'une intervention. Plus précisément, ce déclencheur met à jour la valeur de "visa_uea" à 0 lorsque la valeur de "visa_etb" est mise à jour à 0.

6.4. Test

Avant de procéder aux tests des déclencheurs, nous avons effectué des tests sur les fonctions et les procédures que nous avons préparées.

6.4.1. Fonctions & Procédures stockées

6.4.1.1. Type_Invervention

```
● ● ●
create or replace function Type_Invervention(id_en bigint ,id_et bigint)
  returns int
  language plpgsql
as
$$
declare
etb int ;
begin

  select "Etablissement" into etb
  from enseignants
  where id = id_en ;

  if etb <> id_et then
    return 2 ; --vacation --
  else
    return 1 ; --origine--
  end if;
end;
$$;
```

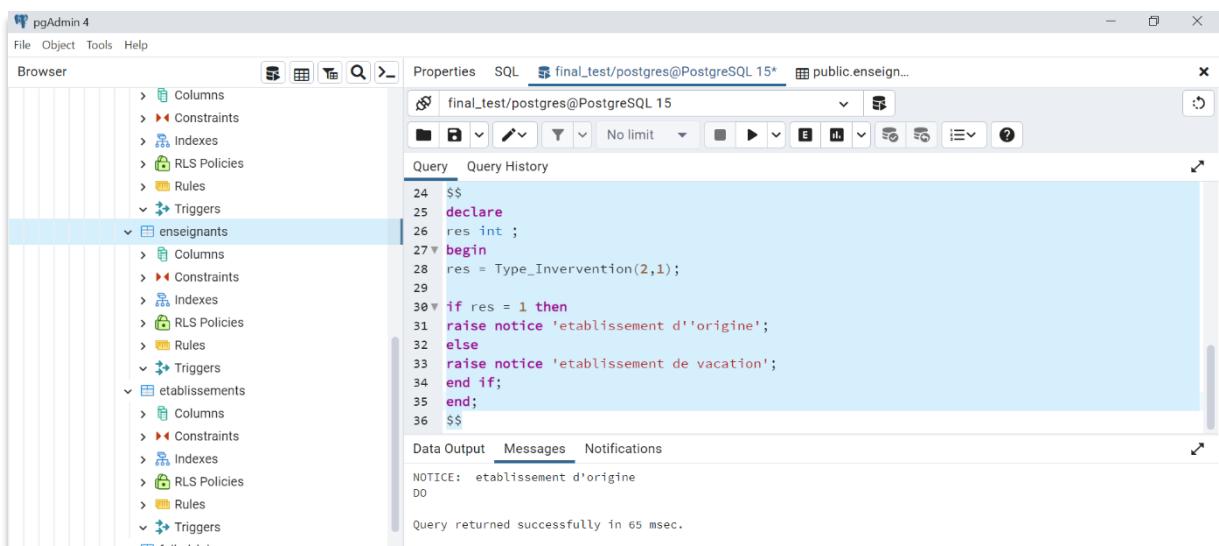
La fonction que nous avons développée prend en argument les identifiants de l'enseignant et de l'établissement afin de déterminer s'il s'agit de son établissement d'origine. Elle retourne la valeur 1 si c'est le cas, sinon elle retourne la valeur 2.

- Test de la fonction :

```
● ● ●
do
$$
declare
res int ;
begin
res = Type_Invervention(2,1);

if res = 1 then
raise notice 'établissement d''origine';
else
raise notice 'établissement de vacation';
end if;
end;
$$
```

En regardant l'exemple ci-dessus, nous pouvons voir que l'enseignant ayant l'ID 2 appartient à l'établissement 1. Alors La fonction a retourné "true", ce qui fournit l'affichage de "Établissement d'origine".



6.4.1.2.

Chercher_dejaP

```

create or replace function Chercher_dejaP( id_i bigint, etb bigint, an varchar)
    returns BOOLEAN
    language plpgsql
as
$$
declare
begin
PERFORM "id_Intervenant" from paiements
where "id_Intervenant" = id_i and "id_Etab" = etb and "Annee_univ" = an;

if not found then
return false ;
else
return true ;
end if ;

end;
$$;

```

La fonction "chercher_dejaP" que nous utilisons dans la procédure "Ajouter_Paiement" a pour but de vérifier si l'enseignant a déjà un paiement correspondant à une intervention effectuée dans le même établissement et la même année que celle fournie en argument. Si tel est le cas, la fonction renvoie "true", ce qui indique qu'un nouvel enregistrement doit être inséré dans la table "paiements". Sinon, la fonction renvoie "false", ce qui permet de mettre à jour la nouvelle valeur du volume horaire dans la table "paiements".

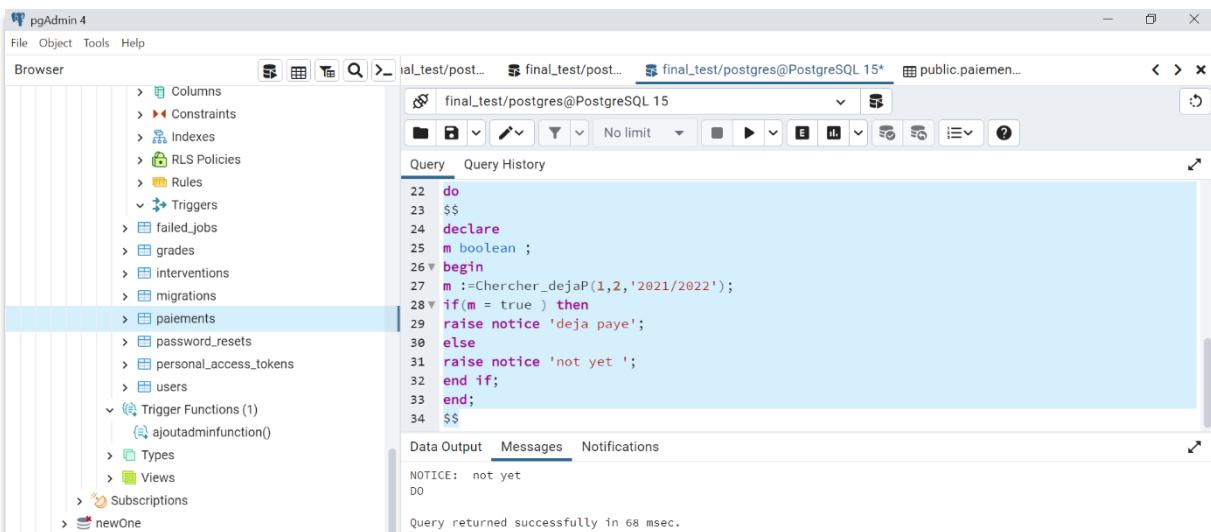
- Test de la fonction :

```

do
$$
declare
m boolean ;
begin
m :=Chercher_dejaP(1,2,'2021/2022');
if(m = true ) then
raise notice 'deja paye';
else
raise notice 'not yet ';
end if;
end;
$$

```

Étant donné que l'enseignant d'ID 1 n'a pas de paiement à l'établissement 2 pour l'année 2021/2022, cela suggère qu'il n'a pas encore effectué d'intervention dans cet établissement. Par conséquent, nous obtenons l'affichage du message "not yet".



6.4.1.3. Charge_Prof

```

create or replace function Charge_Prof(id bigint , etb bigint, an varchar)
  returns double precision
  language plpgsql
as
$$
declare
nbr_h int ;
begin

  select SUM("Nbr_heures") into nbr_h
  from interventions
  where "id_Intervenant" = id and "id_Etab" = etb and visa_uea = 1 and visa_etb = 1 and "Annee_univ" = an;
  return  nbr_h ;

end;
$$;

```

La fonction "Charge_Prof" que nous avons mentionnée précédemment a pour objectif de calculer le nombre d'heures des interventions validées, par le président de l'université et le

directeur de l'établissement, de l'enseignant dans l'établissement et l'année spécifiés dans les arguments de la fonction.

- Test de la fonction :

```
do
$$
declare
nbr_h int ;
begin
raise notice 'le nombre des heures est %',Charge_Prof(1,1,'2022/2023');
end;
$$
```

Si nous calculons le nombre d'heures dans la table "interventions" de l'enseignant d'ID 1 dans l'établissement 1, en tenant compte uniquement des interventions validées, nous obtenons un total de 38 heures.

	id_intervention	nom_univ	Semestre	Date_debut	Date_fin	Nbr_heures	visa_etb	visa_uea	created_at
1	1/22/2023		Semestre 1	2022-12-01	2022-12-02	8	1	1	[null]
2	1/22/2023		Semestre 1	2022-10-10	2022-10-15	30	1	1	[null]
3	1/22/2023		Semestre 2	2023-04-10	2023-05-20	60	0	0	[null]
4	1/22/2023		Semestre 1	2022-11-20	2022-11-25	25	0	0	[null]
5	1/22/2023		Semestre 2	2023-03-05	2023-03-07	15	0	0	[null]
6	1/22/2023		Semestre 1	2022-09-01	2022-09-30	20	0	0	[null]

Pour la ligne 5, il s'agit également d'une intervention de l'enseignant d'ID 1 dans l'établissement 1, mais elle n'a pas encore été validée.

```

14
15
16
17
18
19 do $$ 
20 declare
21 nbr_h int ;
22 begin
23 raise notice 'le nombre des heures est %',Charge_Prof(1,1,'2022/2023');
24 end;
25 $$
```

NOTICE: le nombre des heures est 38

DO

Query returned successfully in 61 msec.

6.4.1.4. Charge_S

```

create or replace function Charge_S(ens bigint)
  returns double precision
  language plpgsql
as
$$
declare
  ch_s double precision ;
begin

  select charge_statutaire into ch_s
  FROM grades G,enseignants E
  WHERE G."id_Grade" = E."id_Grade" and E.id =ens ;
return  ch_s ;
end;
$$;

```

La fonction "charge_S" permet de retourner la charge statutaire de l'enseignant en fonction de son identifiant (id).

- Test de la fonction :

```

do
$$
declare
  charge double precision;
begin
  charge = Charge_S(1);
  raise notice 'la charge statutaire est % ',charge;
end;
$$

```

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database schema with tables 'enseignants' and 'etablissements'. The main area shows a SQL query window with the following content:

```

do
$$
declare
  charge double precision;
begin
  charge = Charge_S(1);
  raise notice 'la charge statutaire est % ',charge;
end;
$$

```

The results pane at the bottom shows the output of the query:

NOTICE: la charge statutaire est 200

6.4.1.5. PaiementH

```

create or replace function PaiementH(ens bigint)
  returns double precision
  language plpgsql
  as
$$
declare
  h_s double precision ;
begin

  select "Taux_horaire_Vocation"
  into h_s
  FROM grades G,enseignants E
  WHERE G."id_Grade" = E."id_Grade" and E.id =ens ;
return  h_s ;
end;
$$;

```

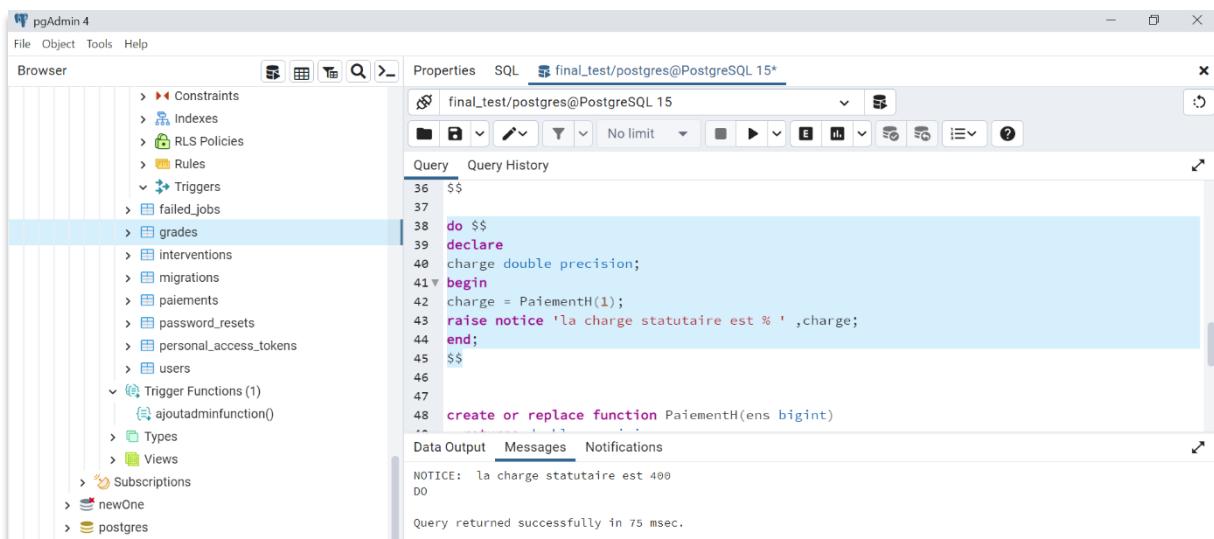
La fonction "PaiementH" permet de retourner la valeur du taux horaire en fonction du grade de l'enseignant.

- Test de la fonction :

```

do
$$
declare
charge double precision;
begin
charge = PaiementH(1);
raise notice 'la charge statutaire est % ',charge;
end;
$$

```



6.4.1.6.

Procédure : "Ajouter_Paiment"

```

create or replace procedure Ajouter_Paiment(id_i bigint, etb bigint , nbr_h double precision ,t_h double
precision ,a_n varchar, s varchar)
language plpgsql
as
$$
BEGIN
IF Chercher_dejaP(id_i,etb,a_n) = false THEN
    INSERT INTO paiements ("id_Intervenant","id_Etab","VH","Taux_H","Annee_univ", "Semestre")
        VALUES (id_i,etb,nbr_h,t_h,a_n,s);
else

    update paiements
    set "VH" = nbr_h
    where "id_Intervenant" = id_i and "id_Etab" = etb and "Annee_univ" = a_n;

end if ;
end;
$$

```

- Test de la procédure :

```

do
$$
declare
begin
call Ajouter_Paiment(5,1,6,500,'2022/2023','S2');
end;
$$

```

L'enseignant d'ID 5 effectuant une vacation à l'établissement 1 de 6 heures.

Avant d'appeler la procédure :

ID	[PK] bigint	id_Intervenant	id_Etab	VH	Taux_H	Brut	IR	NET
1	1	1	1	6	500	3000	0.38	
2	2	1	2	200	500	100000	0.38	
3	3	2	1	50	400	20000	0.38	
4	4	3	2	50	400	20000	0.38	
5	5	4	3	25	300	7500	0.38	
6	6	5	4	35	300	10500	0.38	

Après d'appeler la procédure :

Nous obtenons un nouvel enregistrement inséré dans la table "paiements" avec les informations correspondantes.

	id [PK] bigint	id_Intervenant bigint	id_Etab bigint	VH double precision	Taux_H double precision	Brut double precision	IR double precision	NET double
1	1		1	1	6	500	3000	0.38
2	2		1	2	200	500	100000	0.38
3	3		2	1	50	400	20000	0.38
4	4		3	2	50	400	20000	0.38
5	5		4	3	25	300	7500	0.38
6	6		5	4	35	300	10500	0.38
7	7		5	1	6	500	3000	0.38

6.4.1.7.

Procédure : "Update_Paiment"

```
Create or replace procedure Update_Paiment(id_i bigint, etb bigint , nbr_h double precision , an varchar)
language plpgsql
as
$$
BEGIN
update paiements
set "VH"= nbr_h
where "id_Intervenant" = id_i and "id_Etab" = etb and "Annee_univ" = an;
end;
$$
```

- Test de la procédure :

```
do
$$
begin
CALL Update_Paiment(5,4,20,'2022/2023');
end;
$$
```

Avant d'appeler la procédure :

Le VH était 35 heures.

	id [PK] bigint	id_Intervenant bigint	id_Etab bigint	VH double precision	Taux_H double precision	Brut double precision	IR double precision	NET double
1	4	3	2	50	400	20000	0.38	
2	5	4	3	25	300	7500	0.38	
3	6	5	4	35	300	10500	0.38	
4	9	1	1	90	200	18000	0.38	
5	10	2	1	85	200	17000	0.38	

Après d'appeler la procédure :

Le VH est 20 heures.

	id [PK] bigint	id_Intervenant bigint	id_Etab bigint	VH double precision	Taux_H double precision	Brut double precision	IR double precision	NET double
1	4	3	2	50	400	20000	0.38	
2	5	4	3	25	300	7500	0.38	
3	6	5	4	20	300	6000	0.38	
4	9	1	1	90	200	18000	0.38	
5	10	2	1	85	200	17000	0.38	

6.4.1.8.

Procédure : "Delete_Paiment"

```

Create or replace procedure Delete_Paiment(id_i bigint, etb bigint, an varchar )
language plpgsql
as
$$
BEGIN
delete from paiements where "id_Intervenant" = id_i and "id_Etab" = etb and "Annee_univ" = an;
end;
$$

```

- Test de la procédure :

```
do $$  
begin  
CALL Delete_Paiment(5,1,'2022/2023');  
end;  
$$
```

Avant d'appeler la procédure :

Nous avons 6 lignes dans la table.

The screenshot shows the pgAdmin 4 interface with the 'paitements' table selected in the browser pane. The table structure is as follows:

	id [PK] bigint	id_Interenant bigint	id_Etab bigint	VH double precision	Taux_H double precision	Brut double precision	IR double precision	NET double
1		4	3	2	50	400	20000	0.38
2		5	4	3	25	300	7500	0.38
3		6	5	4	35	300	10500	0.38
4		7	5	1	6	500	3000	0.38
5		9	1	1	90	200	18000	0.38
6		10	2	1	85	200	17000	0.38

Après d'appeler la procédure :

Nous obtenons 5 lignes.

The screenshot shows the pgAdmin 4 interface with the 'paitements' table selected in the browser pane. The table structure is as follows:

	id [PK] bigint	id_Interenant bigint	id_Etab bigint	VH double precision	Taux_H double precision	Brut double precision	IR double precision	NET double
1		4	3	2	50	400	20000	0.38
2		5	4	3	25	300	7500	0.38
3		6	5	4	35	300	10500	0.38
4		9	1	1	90	200	18000	0.38
5		10	2	1	85	200	17000	0.38

6.5. Déclencheurs

6.5.1. Test de déclencheur 1 :

La table "etablissements" avant d'exécuter le déclencheur.

XTRADH

pgAdmin 4

File Object Tools Help

Browser

public.établissements final_test/post... public.établissements final_test/post... public.établissements final_test/postgres@PostgreSQL 15 public.

Query Query History

```
1 SELECT * FROM public.établissements
2 ORDER BY id ASC
```

Data Output Messages Notifications

	Telephone	Faxe	ville	Nbr_enseignants	created_at
1	ion	05223 45678	0537890123	Tanger	30 [null]
2		05378 90123	AL HOCIEMA	25 [null]	
3	umaines	05244 56789	0537234567	Tetouan	35 [null]
4	iection	0539612345	0523398765	Tanger	50 [null]
5	es	0523398765	0535432109	Tanger	68 [null]
6	es	0537234567	0524465432	Tetouan	59 [null]
7	es	0524465432	0539678901	AL HOCIEMA	45 [null]
8		0535432109	0525543210	Laracheoee	20 [null]
9		0539979316	0537890117	Tetouan	300 [null]

Après, l'insertion dans la table "enseignants ", le nombre des enseignants de l'établissement 1 va augmenter de 70 à 71.

```
INSERT INTO enseignants ("PPR", "Nom", prenom, "Date_Naissance", "Etablissement", "id_Grade", id_user)
VALUES ('testPPR', 'testNOM', 'testPrenom', '1990-08-18', 1, 1, 25);
```

pgAdmin 4

File Object Tools Help

Browser

Properties SQL final_test/post... final_test/post... public.établissements final_test/postgres@PostgreSQL 15

Query Query History

```
1 SELECT * FROM public.établissements
2 ORDER BY id ASC
```

Data Output Messages Notifications

	Faxe	ville	Nbr_enseignants	created_at	updated_at
1	0537890123	Tanger	30 [null]	[null]	
2	0522654321	AL HOCIEMA	25 [null]	[null]	
3	0537234567	Tetouan	35 [null]	[null]	
4	0523398765	Tanger	50 [null]	[null]	
5	0535432109	Tanger	71 [null]	[null]	
6	0524465432	Tetouan	60 [null]	[null]	
7	0539678901	AL HOCIEMA	45 [null]	[null]	
8	0525543210	Laracheoee	20 [null]	[null]	
9	0537890117	Tetouan	300 [null]	[null]	

Dans ce cas, on change l'établissement de l'enseignant d'ID 22 de 1 à 3, ce qui possède une décrémentation du nombre des enseignants dans l'établissement 1 et une incrémentation dans l'établissement 3.

```
UPDATE enseignants
set "Etablissement"=3
where id=22;
```

XTRADH

```
1 SELECT * FROM public.etablissements
2 ORDER BY id ASC
```

The screenshot shows the pgAdmin 4 interface. The left sidebar is a tree view of database objects under 'Browser'. The main area has a 'Query' tab with the above SQL code and a 'Data Output' tab showing a table with 9 rows of data. The columns are: Telephone, Fax, ville, Nbr_enseignants, and created_at.

	Telephone	Fax	ville	Nbr_enseignants	created_at
1	05223 45678	0537890123	Tanger	30	[null]
2	05378 90123	0522654321	AL HOCEIMA	25	[null]
3	05244 56789	0537234567	Tetouan	36	[null]
4	0539612345	0523398765	Tanger	50	[null]
5	0523398765	0535432109	Tanger	68	[null]
6	0537234567	0524465432	Tetouan	59	[null]
7	0524465432	0539678901	AL HOCEIMA	45	[null]
8	0535432109	0525543210	Laracheoe	20	[null]
9	0539979316	0537890117	Tetouan	300	[null]

Dans ce cas, on supprime l'enseignant d'ID 15, ce qui possède une décrémentation du nombre des enseignants dans son établissement de 60 à 51.

```
delete from enseignants where id = 15;
```

The screenshot shows the pgAdmin 4 interface again. The browser pane is identical. The query editor pane still has the original SQL command. The data output table now has 8 rows, with the last row (id 15) removed.

	Telephone	Fax	ville	Nbr_enseignants	created_at
1	05223 45678	0537890123	Tanger	30	[null]
2	05378 90123	0522654321	AL HOCEIMA	25	[null]
3	05244 56789	0537234567	Tetouan	35	[null]
4	0539612345	0523398765	Tanger	50	[null]
5	0523398765	0535432109	Tanger	68	[null]
6	0537234567	0524465432	Tetouan	60	[null]
7	0524465432	0539678901	AL HOCEIMA	45	[null]
8	0535432109	0525543210	Laracheoe	20	[null]

XTRADH

	Telephone	Fax	ville	Nbr_enseignants	created_at
1	05223 45678	0537890123	Tanger	30	[null]
2	05378 90123	0522654321	AL HOCEIMA	25	[null]
3	05244 56789	0537234567	Tetouan	35	[null]
4	0539612345	0523398765	Tanger	50	[null]
5	0523398765	0535432109	Tanger	68	[null]
6	0537234567	0524465432	Tetouan	59	[null]
7	0524465432	0539678901	AL HOCEIMA	45	[null]
8	0535432109	0525543210	Laracheoe	20	[null]
9	0539979316	0537890117	Tetouan	300	[null]

6.5.2. Test de déclencheur 2 :

Test pour établissement de vacation :

Toutes les heures sont considérées comme des heures supplémentaires.

```
update interventions
set visa_uea = 1
where id_intervention = 3;
```

Dans le cas où l'enseignant d'ID 3 (la ligne 2) a déjà une intervention validée dans le même établissement que celle mentionnée dans la ligne 1, lors de la validation de cette nouvelle intervention par le président, les 30 heures doivent être ajoutées aux 60 heures déjà présentes dans la table "paiements" pour cet enseignant et cet établissement.

Avant l'Update de visa_uea :

	id [PK] bigint	id_Intervenant	id_Etab	VH	Taux_H	Brut	IR	NET
1	2	1	2	200	500	100000	0.38	
2	3	2	1	50	400	20000	0.38	
3	4	3	2	50	400	20000	0.38	
4	5	4	3	25	300	7500	0.38	
5	6	5	4	35	300	10500	0.38	
6	7	5	1	6	500	3000	0.38	
7	9	1	1	60	200	12000	0.38	

pgAdmin 4

File Object Tools Help

Browser

- Columns
- Constraints
- Indexes
- RLS Policies
- Rules
- Triggers
- failed_jobs
- grades
- interventions
 - Columns
 - Constraints
 - Indexes
 - RLS Policies
 - Rules
 - Triggers (2)
 - ajoutpaimentrigger
 - intertrigger
- migrations
- paiements

Properties SQL final_test/post... public.paiemen... public.interventions/final_test/postgres@PostgreSQL 15

Query Query History

```
1 SELECT * FROM public.interventions
2 ORDER BY id_intervention ASC
```

Data Output Messages Notifications

Annee_univ	Semestre	Date_debut	Date_fin	Nbr_heures	visa_etb	visa_uee	created_a
2022/2023	Semestre 1	2022-12-01	2022-12-02	60	1	1	[null]
2022/2023	Semestre 1	2022-10-10	2022-10-15	30	1	0	[null]
2022/2023	Semestre 2	2023-04-10	2023-05-20	60	0	0	[null]
2022/2023	Semestre 1	2022-11-20	2022-11-25	25	0	0	[null]
2022/2023	Semestre 2	2023-03-05	2023-03-07	15	0	0	[null]
2022/2023	Semestre 1	2022-09-01	2022-09-30	20	0	0	[null]

Après l'Update de visa_uee :

pgAdmin 4

File Object Tools Help

Browser

- Columns
- Constraints
- Indexes
- RLS Policies
- Rules
- Triggers
- failed_jobs
- grades
- interventions
 - Columns
 - Constraints
 - Indexes
 - RLS Policies
 - Rules
 - Triggers (2)
 - ajoutpaimentrigger
 - intertrigger
- migrations
- paiements
- password_resets
- personal_access_tokens

Properties SQL final_test/post... public.paiemen... public.interventions/final_test/postgres@PostgreSQL 15

Query Query History

```
1 SELECT * FROM public.interventions
2 ORDER BY id_intervention ASC
```

Data Output Messages Notifications

Annee_univ	Semestre	Date_debut	Date_fin	Nbr_heures	visa_etb	visa_uee	created_a
2022/2023	Semestre 1	2022-12-01	2022-12-02	60	1	1	[null]
2022/2023	Semestre 1	2022-10-10	2022-10-15	30	1	1	[null]
2022/2023	Semestre 2	2023-04-10	2023-05-20	60	0	0	[null]
2022/2023	Semestre 1	2022-11-20	2022-11-25	25	0	0	[null]
2022/2023	Semestre 2	2023-03-05	2023-03-07	15	0	0	[null]
2022/2023	Semestre 1	2022-09-01	2022-09-30	20	0	0	[null]

Successfully run. Total query runtime: 274 msec. 6 rows affected.

pgAdmin 4

File Object Tools Help

Browser

- Columns
- Constraints
- Indexes
- RLS Policies
- Rules
- Triggers
- failed_jobs
- grades
- interventions
 - Columns
 - Constraints
 - Indexes
 - RLS Policies
 - Rules
 - Triggers (2)
 - ajoutpaimentrigger
 - intertrigger
- migrations
- paiements

Properties SQL final_test/post... public.paiements/final_test/postgres@PostgreSQL 15

Query Query History

```
1 SELECT * FROM public.paiements
2 ORDER BY id ASC
```

Data Output Messages Notifications

id	id_Intervenant	id_Etab	VH	Taux_H	Brut	IR	NET
1	2	1	2	200	500	100000	0.38
2	3	2	1	50	400	20000	0.38
3	4	3	2	50	400	20000	0.38
4	5	4	3	25	300	7500	0.38
5	6	5	4	35	300	10500	0.38
6	7	5	1	6	500	3000	0.38
7	9	1	1	90	200	18000	0.38

Test pour une établissement d'origine :

```
● ● ●
update interventions
set visa_uee = 1
where id_intervention = 4;
```

Puisque la charge statutaire de l'enseignant ayant l'intervention d'ID 4 (la ligne 3) est 200h et le nombre des heures de cet enseignant dans son établissement est inférieur à 200h, donc ces heures de cette intervention ne seront pas considérées comme des heures supplémentaires, d'où aucune insertion dans la table paiements.

Browser

- > Columns
- > Constraints
- > Indexes
- > RLS Policies
- > Rules
- < Triggers
- > failed_jobs
- > grades
- interventions**
 - > Columns
 - > Constraints
 - > Indexes
 - > RLS Policies
 - > Rules
 - < Triggers (2)
 - > ajoutpaiementtrigger
 - > intertrigger
- > migrations
- > paiements
- > password_resets

Properties **SQL** **final_test/postgres@PostgreSQL 15** **public.paiements/public.interventions/final_test/postgres@PostgreSQL 15**

Query **Query History**

```
1 SELECT * FROM public.interventions
2 ORDER BY id_intervention ASC
```

Data Output **Messages** **Notifications**

id	varying (10)	Semestre	Date_debut	Date_fin	Nbr_heures	visa_etb	visa_uee	created_at
1	23	Semestre 1	2022-12-01	2022-12-02	60	1	1	[null]
2	23	Semestre 1	2022-10-10	2022-10-15	30	1	1	[null]
3	23	Semestre 2	2023-04-10	2023-05-20	60	1	1	[null]
4	23	Semestre 1	2022-11-20	2022-11-25	25	0	0	[null]
5	23	Semestre 2	2023-03-05	2023-03-07	15	0	0	[null]
6	23	Semestre 1	2022-09-01	2022-09-30	20	0	0	[null]

Browser

- > Columns
- > Constraints
- > Indexes
- > RLS Policies
- > Rules
- < Triggers
- > failed_jobs
- > grades
- interventions**
 - > Columns
 - > Constraints
 - > Indexes
 - > RLS Policies
 - > Rules
 - < Triggers (2)
 - > ajoutpaiementtrigger
 - > intertrigger
- > migrations
- > paiements
- > password_resets

Properties **SQL** **final_test/postgres@PostgreSQL 15** **public.paiements/public.interventions/final_test/postgres@PostgreSQL 15**

Query **Query History**

```
1 SELECT * FROM public.paiements
2 ORDER BY id ASC
```

Data Output **Messages** **Notifications**

id	[PK] bigint	id_Intervenant	id_Etab	VH	Taux_H	Brut	IR	NET
1	3	2	1	50	400	20000	0.38	
2	4	3	2	50	400	20000	0.38	
3	5	4	3	25	300	7500	0.38	
4	6	5	4	35	300	10500	0.38	
5	7	5	1	6	500	3000	0.38	
6	9	1	1	90	200	18000	0.38	

Suite du même test :

```
● ● ●
update interventions
set visa_uee = 1
where id_intervention = 5;
```

Maintenant l'enseignant précédent ayant une nouvelle intervention d'ID 5 dans son établissement de 210 heures, alors la somme ces heures est 270 heures ce qui possède l'insertion de 70 heures dans la table paiements.

Avant l'Update de visa_uee :

	id_uee	Semestre	Date_debut	Date_fin	Nbr_heures	visa_etb	visa_uee	created_at
1	022/2023	Semestre 1	2022-12-01	2022-12-02	60	1	1	[null]
2	022/2023	Semestre 1	2022-10-10	2022-10-15	30	1	1	[null]
3	022/2023	Semestre 2	2023-04-10	2023-05-20	60	1	1	[null]
4	022/2023	Semestre 1	2022-11-20	2022-11-25	210	0	0	[null]
5	022/2023	Semestre 2	2023-03-05	2023-03-07	15	0	0	[null]
6	022/2023	Semestre 1	2022-09-01	2022-09-30	20	0	0	[null]

Après l'Update de visa_uee :

	id_uee	Semestre	Date_debut	Date_fin	Nbr_heures	visa_etb	visa_uee	created_at	up_tir
1		Semestre 1	2022-12-01	2022-12-02	60	1	1	[null]	[n]
2		Semestre 1	2022-10-10	2022-10-15	30	1	1	[null]	[n]
3		Semestre 2	2023-04-10	2023-05-20	60	1	1	[null]	[n]
4		Semestre 1	2022-11-20	2022-11-25	210	1	1	[null]	[n]
5		Semestre 2	2023-03-05	2023-03-07	15	0	0	[null]	[n]
6		Semestre 1	2022-09-01	2022-09-30	20	0	0	[null]	[n]

	id	id_Intervention	id_Etab	VH	Taux_H	Brut	IR	NET
1	4	3	2	50	400	20000	0.38	
2	5	4	3	25	300	7500	0.38	
3	6	5	4	35	300	10500	0.38	
4	7	5	1	6	500	3000	0.38	
5	9	1	1	90	200	18000	0.38	
6	10	2	1	70	200	14000	0.38	

6.5.3. Test de déclencheur 3 :

Pour les interventions de vacation :

```
update interventions
set visa_uea = 0
where id_intervention =2;
```

Nous avons annulé l'intervention d'ID est 2 (la ligne 1) ayant 60 heures comme des heures de vacation et puisque l'enseignant a déjà 90 heures du VH dans la table "paiements", alors la nouvelle valeur du VH sera 30.

Avant la mise à jour :

	id [PK] bigint	id_Intervenant bigint	id_Etab bigint	VH double precision	Taux_H double precision	Brut double precision	IR double precision	NET double
1	4	3	2	50	400	20000	0.38	
2	5	4	3	25	300	7500	0.38	
3	6	5	4	20	300	6000	0.38	
4	9	1	1	90	200	18000	0.38	
5	10	2	1	85	200	17000	0.38	

Après la mise à jour :

	id [PK] bigint	id_Intervenant bigint	id_Etab bigint	VH double precision	Taux_H double precision	Brut double precision	IR double precision	NET double
1	4	3	2	50	400	20000	0.38	
2	5	4	3	25	300	7500	0.38	
3	6	5	4	20	300	6000	0.38	
4	9	1	1	30	200	6000	0.38	
5	10	2	1	85	200	17000	0.38	

XTRADH

pgAdmin 4

Browser

Properties SQL final_test/post... public.interventions/final_test/postgres@PostgreSQL 15 public.paiements/final_test/postgres@PostgreSQL 15

Query Query History

```
1 SELECT * FROM public.interventions
2 ORDER BY id_intervention ASC
```

Data Output Messages Notifications

	ee_univ	Semestre	Date_debut	Date_fin	Nbr_heures	visa_etb	visa_uee	created_at
1	2/2023	Semestre 1	2022-12-01	2022-12-02	60	0	0	[null]
2	2/2023	Semestre 1	2022-10-10	2022-10-15	30	1	1	[null]
3	2/2023	Semestre 2	2023-04-10	2023-05-20	60	1	1	[null]
4	2/2023	Semestre 1	2022-11-20	2022-11-25	210	1	1	[null]
5	2/2023	Semestre 2	2023-03-05	2023-03-07	15	1	1	[null]
6	2/2023	Semestre 1	2022-09-01	2022-09-30	20	0	0	[null]

Suite de test :

```
update interventions
set visa_uee = 0
where id_intervention =3;
```

Dans ce cas, si on annule l'intervention d'ID 3 (la ligne 2), la procédure "Delete_Paiement" sera appelé puisqu'on n'a pas des heures supplémentaires dans cet établissement.

pgAdmin 4

Browser

Properties SQL final_test/post... public.interventions/final_test/postgres@PostgreSQL 15 public.paiements/final_test/postgres@PostgreSQL 15

Query Query History

```
1 SELECT * FROM public.paiements
2 ORDER BY id ASC
```

Data Output Messages Notifications

	id	id_Intervenant	id_Etab	VH	Taux_H	Brut	IR	NET
1	4	3	2	50	400	20000	0.38	
2	5	4	3	25	300	7500	0.38	
3	6	5	4	20	300	6000	0.38	
4	10	2	1	85	200	17000	0.38	

Pour les interventions dans les établissements d'origine :

```
update interventions
set visa_uee = 0
where id_intervention =6;
```

L'enseignant dispose toujours d'heures supplémentaires dans son établissement, ce qui se traduit par une mise à jour limitée du volume horaire (VH) dans la table des paiements, passant de 85 à 70.

Avant la mise à jour :

pgAdmin 4

Browser

Properties SQL public.interven... public.paiements/final_test/postgres@PostgreSQL 15

Query Query History

```
1 SELECT * FROM public.paiements
2 ORDER BY id ASC
```

Data Output Messages Notifications

	id [PK] bigint	id_Interenant bigint	id_Etab bigint	VH double precision	Taux_H double precision	Brut double precision	IR double precision	NET double
1	4	3	2	50	400	20000	0.38	
2	5	4	3	25	300	7500	0.38	
3	6	5	4	20	300	6000	0.38	
4	10	2	1	85	200	17000	0.38	

Après la mise à jour :

pgAdmin 4

Browser

Properties SQL public.interven... public.paiements/final_test/postgres@PostgreSQL 15

Query Query History

```
1 SELECT * FROM public.interventions
2 ORDER BY id_intervention ASC
```

Data Output Messages Notifications

	Date_debut date	Date_fin date	Nbr_heures integer	visa_etb integer	visa_uea integer	created_at timestamp without time zone	updated_at timestamp without time zone
1	2022-12-01	2022-12-02	60	0	0	[null]	[null]
2	2022-10-10	2022-10-15	30	0	0	[null]	[null]
3	2023-04-10	2023-05-20	60	1	1	[null]	[null]
4	2022-11-20	2022-11-25	210	1	1	[null]	[null]
5	2023-03-05	2023-03-07	15	0	0	[null]	[null]
6	2022-09-01	2022-09-30	20	0	0	[null]	[null]

pgAdmin 4

Browser

Properties SQL public.interven... public.paiements/final_test/postgres@PostgreSQL 15

Query Query History

```
1 SELECT * FROM public.paiements
2 ORDER BY id ASC
```

Data Output Messages Notifications

	id [PK] bigint	id_Interenant bigint	id_Etab bigint	VH double precision	Taux_H double precision	Brut double precision	IR double precision	NET double
1	4	3	2	50	400	20000	0.38	
2	5	4	3	25	300	7500	0.38	
3	6	5	4	20	300	6000	0.38	
4	10	2	1	70	200	14000	0.38	

Suite du test :

```
update interventions
set visa_uea = 0
where id_intervention =5;
```

À ce stade-là, si nous annulons encore une intervention de l'enseignant précédent dans son établissement, le déclencheur va exécuter la procédure de "Delete_Paiment" car les heures qui restent (60 heures), elles sont considérées comme des heures permanentes puisque la charge statutaire de cet enseignant est 200 heures.

Avant la mise à jour :

	Semestre	Date_debut	Date_fin	Nbr_heures	visa_etb	visa_uee	created_at	updated_at
1	Semestre 1	2022-12-01	2022-12-02	60	0	0	[null]	[null]
2	Semestre 1	2022-10-10	2022-10-15	30	0	0	[null]	[null]
3	Semestre 2	2023-04-10	2023-05-20	60	1	1	[null]	[null]
4	Semestre 1	2022-11-20	2022-11-25	210	0	0	[null]	[null]
5	Semestre 2	2023-03-05	2023-03-07	15	0	0	[null]	[null]
6	Semestre 1	2022-09-01	2022-09-30	20	0	0	[null]	[null]

Après la mise à jour :

	id	id_intervenant	id_Etab	VH	Taux_H	Brut	IR	NET
1	4	3	2	50	400	20000	0.38	
2	5	4	3	25	300	7500	0.38	
3	6	5	4	20	300	6000	0.38	

6.5.4. Test de déclencheur 4 :

Ce déclencheur met à jour la valeur de "visa_uee" à 0 lorsque la valeur de "visa_etb" est mise à jour à 0.

```

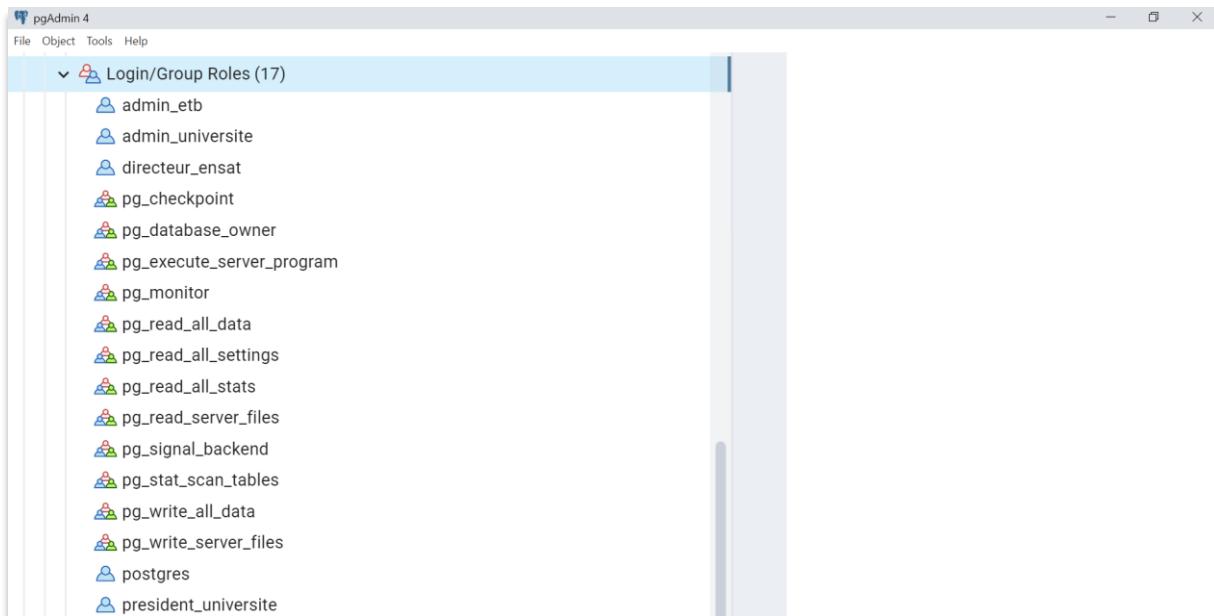
● ● ●

update interventions
set visa_etb = 0
where id_intervention = 5;

```

6.6. Création des utilisateurs :

À l'aide de psql (SQL Shell), Nous avons créé 4 utilisateurs, qui peuvent se connecter à la base de données.



6.6.1. Utilisateur 1 : admin_universite

Cet utilisateur peut lire (Select) et modifier (Insert, Update, Delete) les tables « users » et « administrateurs ».

- Script de la création :

```
CREATE USER admin_universite WITH PASSWORD 'admin';
GRANT SELECT, INSERT, UPDATE, DELETE ON users,administrateurs TO admin_universite;
GRANT USAGE, SELECT ON ALL SEQUENCES IN SCHEMA public TO admin_universite;
```

6.6.2. Utilisateur 2 : admin_etb

Cet utilisateur peut lire (Select) et modifier (Insert, Update, Delete) les tables « enseignants », « administrateurs » et « interventions ».

- Script de la création :

```
CREATE USER admin_etb WITH PASSWORD 'admin_etb';
GRANT SELECT, INSERT, UPDATE, DELETE ON users,administrateurs,interventions TO admin_etb;
GRANT USAGE, SELECT ON ALL SEQUENCES IN SCHEMA public TO admin_etb;
```

6.6.3. Utilisateur 3 : president_universite

Cet utilisateur peut lire (Select) et modifier (Insert, Update, Delete) la table « administrateurs » et que de lire les tables « paiements » et « enseignants » ainsi de lire et Update dans la table « interventions ».

- Script de la création :

```
CREATE USER president_universite WITH PASSWORD 'uae';
GRANT SELECT, INSERT, UPDATE, DELETE ON interventions TO president_universite;
GRANT SELECT ON paiements,enseignants TO president_universite;
GRANT SELECT,UPDATE ON interventions TO president_universite;
GRANT USAGE, SELECT ON ALL SEQUENCES IN SCHEMA public TO president_universite;
```

6.6.4. Utilisateur 4 : directeur_ensat

Cet utilisateur peut lire les tables « paiements » et « enseignants » ainsi de lire et Update dans la table « interventions ».

- Script de la création :

```
CREATE USER directeur_ensat WITH PASSWORD 'ensat';
GRANT SELECT ON paiements,enseignants TO directeur_ensat;
GRANT SELECT,UPDATE ON interventions TO directeur_ensat;
GRANT USAGE, SELECT ON ALL SEQUENCES IN SCHEMA public TO directeur_ensat;
```

- Test de connexion :

Nous avons connecté à la base de données comme l'utilisateur « directeur_ensat » pour faire les tests suivants :

```
Sélection SQL Shell (psql)
Server [localhost]:
Database [postgres]: final_test
Port [5432]:
Username [postgres]: directeur_ensat
Mot de passe pour l'utilisateur directeur_ensat :
psql (15.2)
Attention : l'encodage console (850) diffère de l'encodage Windows (1252).
Les caractères 8 bits peuvent ne pas fonctionner correctement.
Voir la section « Notes aux utilisateurs de Windows » de la page
référence de psql pour les détails.
Saisissez « help » pour l'aide.

final_test=> select * from grades;
ERREUR: droit refusé pour la table grades

final_test=> select visa_etb from interventions where id_intervention =1;
visa_etb
-----
(0 ligne)

final_test=> select visa_etb from interventions where id_intervention =2;
visa_etb
-----
      1
(1 ligne)

final_test=> update interventions set visa_etb = 0 where id_intervention = 2;
UPDATE 1
final_test=> select visa_etb from interventions where id_intervention =2;
visa_etb
-----
      0
(1 ligne)

final_test=> delete from interventions;
ERREUR: droit refusé pour la table interventions
final_test=>
```

7. CHAPITRE 7 :

SECURITE

L'objectif principal de ce chapitre est de mettre en œuvre des contrôles de sécurité appropriés pour réduire les risques liés aux vulnérabilités potentielles de l'application. Cela implique d'identifier les points d'accès sensibles, d'évaluer les risques associés, et de mettre en place des contrôles de sécurité adaptés pour prévenir les violations de sécurité et les accès non autorisés.

Lorsqu'il s'agit de gérer les heures supplémentaires des professeurs universitaires, la sécurité de l'application web revêt une importance primordiale. La sécurisation de cette application est cruciale pour préserver la confidentialité des données sensibles, protéger les informations financières, prévenir les attaques et violations de sécurité, ainsi que pour respecter les réglementations en vigueur.

Dans un contexte où les données personnelles et financières des professeurs sont manipulées, il est impératif de mettre en place des mesures de sécurité solides afin de garantir que seules les personnes autorisées aient accès à ces informations confidentielles. En sécurisant l'application, on protège les données personnelles des professeurs, telles que leurs noms, adresses, PPR, etc., contre toute utilisation abusive ou vol d'identité.

De plus, la gestion des heures supplémentaires implique des aspects financiers, tels que les montants des heures supplémentaires, les paiements et les calculs associés. La sécurisation de l'application permet de protéger ces informations financières contre tout accès non autorisé ou toute manipulation frauduleuse, garantissant ainsi l'intégrité des données et la confiance dans les transactions financières.

Les applications web sont souvent la cible d'attaques malveillantes, telles que les attaques par injection SQL et les attaques par cross-site scripting (XSS). Ces attaques peuvent compromettre la sécurité de l'application, entraîner des pertes de données ou des interruptions de service. En mettant en place des mesures de sécurité adaptées, comme la validation des entrées utilisateur, la protection contre les attaques connues et l'utilisation de pratiques de développement sécurisé, on peut réduire considérablement les risques d'attaques et de violations de sécurité.

Quelles sont les techniques qu'on va utiliser pour sécuriser l'application ?

7.1. Sécurisation de l'authentification

On a pris la décision d'utiliser le package Sanctum pour sécuriser l'authentification de notre application.



7.1.1. Pourquoi Sanctum :

Sanctum est un excellent choix pour les applications web et mobiles qui nécessitent une authentification simple et légère, il est tout à fait approprié de l'utiliser pour sécuriser l'authentification dans notre application.

Voici quelques raisons pour lesquelles on a préféré Sanctum dans le contexte de notre application :

- **Simplicité** : Sanctum est facile à mettre en œuvre et à utiliser. Il utilise les sessions et les cookies pour gérer l'authentification.
- **Contrôle d'accès** : Sanctum facilite la mise en place de mécanismes de contrôle d'accès basés sur les rôles et les autorisations. On peut gérer les autorisations des utilisateurs de manière granulaire en utilisant les fonctionnalités intégrées de Laravel.
- **Performance** : Sanctum est conçu pour être léger et optimisé en termes de performance. Il n'ajoute pas de surcharge significative à votre application et fonctionne bien même avec un grand nombre d'utilisateurs.

7.1.1.1. Le principe de fonctionnement de Sanctum :

Laravel Sanctum repose sur l'utilisation de sessions et de jetons d'authentification pour sécuriser l'authentification API dans une application Laravel. Lorsqu'un utilisateur se connecte à l'application, une session est créée, et un cookie de session est généré et stocké dans le

navigateur de l'utilisateur. Ce cookie de session permet d'authentifier l'utilisateur dans les requêtes ultérieures.

Laravel Sanctum intègre également une protection contre les attaques CSRF (Cross-Site Request Forgery) en générant automatiquement un jeton CSRF pour chaque utilisateur authentifié. Ce jeton est inclus dans les requêtes API pour vérifier l'origine de la demande et prévenir les attaques CSRF.

En parallèle, le rôle principale de Sanctum est de générer également des jetons d'authentification pour les requêtes API. Le jeton (token) d'authentification est généré lorsqu'un utilisateur se connecte à l'application et peut être utilisé pour effectuer des appels aux endpoints protégés de l'API.

Voici comment fonctionne le token d'authentification dans Sanctum :

- **Génération du token** : Lorsqu'un utilisateur se connecte avec succès à l'application, Sanctum génère un token d'authentification unique pour cet utilisateur.
- **Stockage du token** : Le token d'authentification est généralement stocké dans un cookie sécurisé dans le navigateur de l'utilisateur. Ce cookie est utilisé pour inclure le token dans les requêtes API ultérieures.
- **Envoyer du token dans les requêtes** : Lorsqu'un utilisateur authentifié effectue une requête vers une ressource protégée de l'API, le token d'authentification est automatiquement inclus dans l'en-tête de la requête (généralement dans l'en-tête Authorization) ou dans le corps de la requête, selon la configuration de Sanctum.
- **Validation du token** : Lorsque le serveur reçoit une requête avec un token d'authentification, il vérifie la validité du token. Si le token est valide, l'utilisateur est considéré comme authentifié et peut accéder aux ressources protégées de l'API. Sinon, une erreur d'authentification sera renvoyée.

Il est important de noter que les tokens d'authentification de Sanctum sont généralement des tokens à usage unique, ce qui signifie qu'ils peuvent être utilisés pour effectuer une seule requête. Cela renforce la sécurité en limitant la possibilité de réutilisation des tokens volés ou compromis.

7.1.1.2.

Comment installer Sanctum (s'il n'est pas déjà installé par défaut) ?

On utilise les commandes suivantes dans un terminal :

- ✓ composer require laravel/sanctum
- ✓ php artisan vendor:publish --provider="Laravel\Sanctum\SanctumServiceProvider"
- ✓ php artisan migrate.

Après l'installation de Sanctum on va créer un contrôleur d'authentification : authController
Ce contrôleur gère les fonctionnalités de connexion, de profil utilisateur et de déconnexion.

7.1.2. Les méthodes principales du contrôleur d'authentification

7.1.2.1. `login(Request \$request)` :

```

1  public function login(Request $request){
2      $fields = $request->validate([
3          'email' =>'required|email',
4          'password' =>'string|required'
5      ]);
6      //check email
7      $user = User::where('email',$fields['email'])->first();
8      //check password
9      if(!$user || !Hash::check($fields['password'],$user->password)){
10          return response([
11              'message' => 'Email or Password are incorrect'
12          ],404);
13      }
14      $token = $user->createToken('MyAppToken')->plainTextToken;
15      $response= [
16          'user'=>$user,
17          'token' =>$token
18      ];
19      return response($response,202);
20  }

```

Cette méthode gère le processus de connexion de l'utilisateur. Elle reçoit une demande (`\$request`) contenant les champs d'email et de mot de passe soumis par l'utilisateur. Les champs sont validés à l'aide de la méthode `validate()` pour s'assurer qu'ils sont présents et respectent les règles de validation spécifiées. Ensuite, elle vérifie si l'email existe dans la base de données et si le mot de passe correspond à celui enregistré pour cet utilisateur. Si les informations d'identification sont correctes, un jeton d'authentification est généré pour l'utilisateur à l'aide de la méthode `createToken()` et une réponse JSON contenant l'utilisateur et le jeton est renvoyée. Sinon, une réponse d'erreur 404 est renvoyée.

7.1.2.2. `userProfile()` :

```
● ● ●

1 public function userProfile(){
2     return response()->json(auth()->user());
3
4 }
```

Cette méthode renvoie les informations du profil de l'utilisateur authentifié. Elle utilise la fonction `auth()->user()` pour récupérer l'utilisateur actuellement authentifié et renvoie ses informations au format JSON.

7.1.2.3. `logout(Request \$request)` :

```
● ● ●

1 public function logout(Request $request)
2 {
3     $user = auth()->user();
4
5     if ($user) {
6         $user->tokens()->delete();
7         return [
8             'message' => 'Logged out'
9         ];
10    }
11
12
13 }
```

Cette méthode gère la déconnexion de l'utilisateur. Elle supprime tous les jetons d'authentification associés à l'utilisateur actuellement authentifié à l'aide de la méthode `tokens()->delete()`. Ensuite, elle renvoie un message indiquant que l'utilisateur s'est déconnecté avec succès.

7.1.3. Sécurisation des routes qui nécessite une authentification :

On utilise le middleware d'authentification :

```

● ● ●

1  class Authenticate extends Middleware
2  {
3      /**
4       * Get the path the user should be redirected to when they are not authenticated.
5       *
6       * @param  \Illuminate\Http\Request  $request
7       * @return string|null
8       */
9      protected function redirectTo($request)
10     {
11         if (! $request->expectsJson()) {
12             return route('login');
13         }
14     }
15 }

```

On prend à titre d'exemple les routes de la déconnection et de l'affichage de l'utilisateur courant (le fichier api.php) :

```

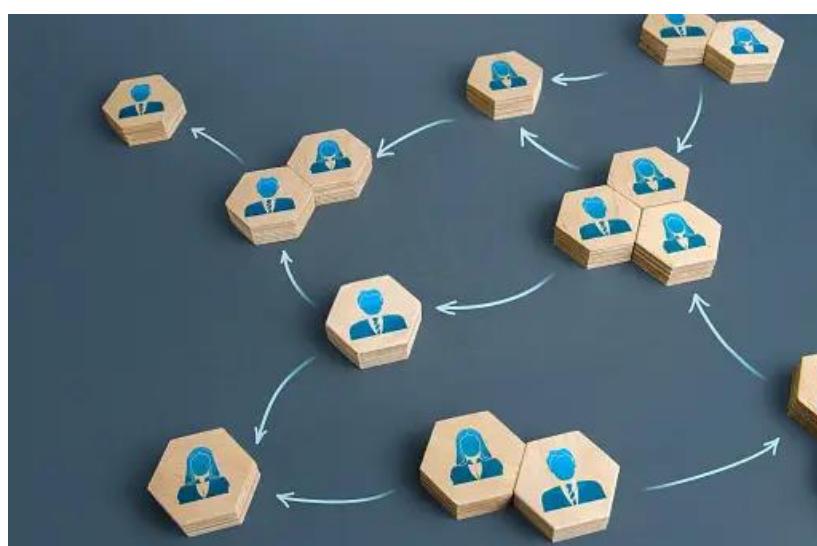
● ● ●

1  Route::group(['middleware'=>['auth:sanctum']], function () {
2      Route::delete('/logout',[AuthController::class,'logout']);
3      Route::get('/user-profile',[AuthController::class,'userProfile']);
4
5  });

```

7.1.4. La sécurisation des routes par rôles spécifique :

Dans notre application de gestion des heures supplémentaires des professeurs universitaires, nous avons mis en place cinq rôles distincts pour assurer un fonctionnement optimal du



système. Chaque rôle est associé à des responsabilités spécifiques et bénéficie de priviléges appropriés, garantissant ainsi une gestion efficace et sécurisée de l'application. Ces rôles sont :

Administrateur d'établissement, Administrateur d'université, Professeur, Président de l'université et Directeur de l'établissement.

On prend comme exemple le rôle d'un administrateur d'établissement :

Le but d'encapsuler ces routes est de limiter l'accès des utilisateurs chacun et ses priviléges.



```
1 Route::middleware(['auth:sanctum', 'role:admin_univ'])->group(function () {
2     // Routes protégées pour admin_univ
3     Route::get('/directeuretab', [AdministrateurController::class, 'directeurETB']);
4     Route::get('/listeAdminETBforadminuae', [AdministrateurController::class, 'listeAdminETBforadminuae']);
5     Route::get('/listepresidentuaeforadminuae', [AdministrateurController::class, 'listepresidentuaeforadminuae']);
6     Route::get('/listedirecteuretbforadminuae', [AdministrateurController::class, 'listedirecteuretbforadminuae']);
7 });
8 }
```

Cette sécurisation de route se fait par un middleware de gestion de route :



```
1 public function handle($request, Closure $next, string $role)
2 {
3     // Si $role n'est pas nul, convertissez-le en tableau
4     $roles = !is_null($role) ? (is_array($role) ? $role : explode('|', $role)) : [];
5
6     // Vérifiez si l'utilisateur authentifié a l'un des rôles spécifiés
7     if (in_array($request->user()->type, $roles)) {
8         return $next($request);
9     }
10
11    // Si l'utilisateur n'a pas le rôle requis, interrompez la requête avec une erreur 403 Forbidden
12    abort(403);
13 }
```

C'est un middleware qui s'assure qu'un utilisateur a le rôle spécifié avant de lui accorder l'accès à une certaine fonctionnalité ou à une route particulière. Voici une explication détaillée de son fonctionnement :

Le paramètre \$role représente le(s) rôle(s) requis pour la route. Il peut s'agir d'un seul rôle ou de plusieurs rôles séparés par un pipe (|).

Le middleware vérifie d'abord si \$role n'est pas nul. S'il n'est pas nul, il le convertit en tableau. Sinon, un tableau vide est utilisé.

Le code vérifie ensuite si le type de l'utilisateur authentifié (\$request->user()->type) correspond à l'un des rôles spécifiés (in_array(\$request->user()->type, \$roles)).

Si une correspondance est trouvée, le middleware permet à la requête de se poursuivre en appelant \$next(\$request).

Si l'utilisateur n'a pas le rôle requis, le middleware interrompt la requête avec une erreur 403 Forbidden.

Assurez-vous que la propriété type de l'utilisateur authentifié correspond aux noms des rôles pour que ce code fonctionne correctement. Vous devrez également vous assurer que l'utilisateur est authentifié avant que ce middleware soit exécuté.

En résumé, ce middleware s'assure que seuls les utilisateurs ayant le rôle spécifié ont l'autorisation d'accéder à la fonctionnalité ou à la route protégée. S'il y a une correspondance, la demande est transmise au prochain middleware ou à la route finale. Sinon, une exception

HTTP 403 est renvoyée pour indiquer un accès interdit.

7.1.5.

Le hachage du mot de passe :

Le hachage de mot de passe est essentiel pour la sécurité des systèmes informatiques. Il consiste à transformer un mot de passe en une chaîne de caractères illisible et unique, rendant ainsi les données confidentielles et protégeant la vie privée des utilisateurs. En utilisant des algorithmes de hachage, il est pratiquement impossible de reconstituer le mot de passe d'origine à partir du hachage. Cela empêche les attaquants d'accéder aux informations sensibles, même s'ils parviennent à obtenir les hachages de mots de passe. Le hachage de mot de passe protège également contre les attaques par force brute, car il est difficile et chronophage de retrouver les mots de passe d'origine à partir des hachages. En utilisant des sels aléatoires en conjonction avec le hachage, chaque hachage devient unique, renforçant ainsi la sécurité globale. De plus, le hachage de mot de passe est conforme aux normes de sécurité et aux réglementations de protection des données. En résumé, le hachage de mot de passe est essentiel pour protéger les données confidentielles, prévenir l'accès non autorisé et renforcer la sécurité des systèmes informatiques.



On a choisi la bibliothèque "bcrypt" parce qu'elle est un outil puissant et sécurisé pour le hachage de mots de passe. Elle offre une protection renforcée contre les attaques par force brute, les attaques par dictionnaire et par table arc-en-ciel, et est facile à utiliser dans un projet Laravel. En l'adoptant, vous pouvez renforcer la sécurité de vos mots de passe et protéger les informations sensibles de vos utilisateurs.



```

1  public function store(Request $request)
2      {
3
4          $fields = $request->validate([
5              'email' =>'required|email|unique:users,email',
6              'type'=>'required'
7          ]);
8
9          $fields['password'] = $request->password ;
10
11         $user = User::create([
12             'type' =>$fields['type'],
13             'email' => $fields['email'],
14             'password'= >bcrypt($fields['password'])
15         ]);

```

Voici quelques points importants à connaître sur la bibliothèque "bcrypt" :

- **Sécurité renforcée** : "bcrypt" utilise un algorithme de hachage de mot de passe adaptatif basé sur Blowfish. Cet algorithme est conçu pour être lent et coûteux en termes de ressources, ce qui rend les attaques par force brute beaucoup plus difficiles et lentes. Il ajoute également automatiquement un "sel" (salt) aléatoire aux mots de passe avant de les hacher, ce qui rend chaque hachage unique et difficile à prédire.
- **Résistance aux attaques par dictionnaire et par table arc-en-ciel** : Les attaques par dictionnaire et par table arc-en-ciel sont des techniques courantes utilisées par les attaquants pour essayer de deviner des mots de passe en utilisant des listes précalculées. En utilisant "bcrypt", ces attaques sont rendues plus difficiles car l'algorithme introduit un "sel" aléatoire et utilise un coût élevé de calcul pour le hachage.
- **Facilité d'utilisation** : La bibliothèque "bcrypt" est généralement facile à intégrer dans un projet Laravel. Elle offre des fonctions simples pour hacher et vérifier les mots de passe. Par exemple, la fonction `bcrypt()` permet de hacher un mot de passe, tandis que la fonction `password_verify()` permet de vérifier si un mot de passe correspond à un hachage donné.
- **Mise à jour automatique du hachage** : Avec "bcrypt", il n'est pas nécessaire de mettre à jour manuellement les hachages de mot de passe lorsqu'un nouvel algorithme ou une nouvelle version de l'algorithme est disponible. Lors de la vérification d'un mot de passe, "bcrypt" vérifiera automatiquement si le hachage utilisé correspond à l'algorithme actuel. Si ce n'est pas le cas, il effectuera une mise à jour transparente du hachage pour garantir la compatibilité.

7.1.6. Validation des données entrées côté serveur :

La validation des données entrées est une pratique importante dans le développement web,

car elle permet de s'assurer que les données fournies par l'utilisateur sont conformes aux règles spécifiées. Laravel offre une bibliothèque de validation robuste qui facilite grandement cette tâche. Voici quelques raisons pour lesquelles la validation des données est essentielle :

- **Sécurité** : La validation des données aide à prévenir les attaques potentielles telles que l'injection SQL, les attaques XSS (Cross-Site Scripting) et autres vulnérabilités de sécurité. En validant les données, vous pouvez vous assurer qu'elles sont propres et sûres avant de les traiter ou de les stocker dans votre application.
- **Cohérence des données** : La validation garantit que les données entrées par l'utilisateur sont cohérentes. On peut spécifier des règles de validation telles que les formats d'email, les longueurs minimales et maximales des champs, les valeurs numériques attendues, etc. Cela aide à maintenir l'intégrité des données et à éviter les erreurs ou les incohérences dans votre application.
- **Expérience utilisateur améliorée** : En validant les données côté serveur, on peut fournir des messages d'erreur clairs et explicites aux utilisateurs lorsqu'ils commettent des erreurs de saisie. Cela permet aux utilisateurs de comprendre rapidement les problèmes rencontrés et de corriger les erreurs. Une meilleure expérience utilisateur contribue à la convivialité et à la satisfaction des utilisateurs de l'application.
- **Prévention des erreurs** : La validation des données aide à prévenir les erreurs et les exceptions inattendues dans l'application. En spécifiant des règles de validation appropriées, on peut nous assurer que les données sont valides et qu'elles peuvent être utilisées en toute sécurité dans notre application. Cela permet de réduire les erreurs de traitement ultérieur des données et les éventuels dysfonctionnements de l'application.



```

1 $validator = Validator::make($request->all(), [
2     'PPR' => 'required',
3     'Nom' => 'required|max:30',
4     'prenom' => 'required|max:30',
5     'Date_Naissance' => 'date|required',
6     'Etablissement' => 'required',
7     'id_Grade' => 'required',
8     'id_user' => 'required',
9 ];
10
11 if ($validator->fails()) {
12     return response()->json(['errors' => $validator->errors()], 400);
13 }

```

Le code utilise la bibliothèque de validation de Laravel pour valider les données de la demande. Si des erreurs de validation sont détectées, une réponse JSON contenant les erreurs est renvoyée. Cela permet de garantir que les données respectent les règles spécifiées et d'éviter les erreurs potentielles dans l'application.

7.2. Les attaques XSS et les injections SQL :

On a parlé de ces vulnérabilités dans la partie de la validation des données.

7.2.1. Injection SQL :

L'injection SQL est une attaque de sécurité courante qui vise à exploiter les vulnérabilités des applications Web qui interagissent avec une base de données. Elle se produit lorsque des données non fiables ou non validées sont incluses dans une requête SQL, permettant à un attaquant d'exécuter des instructions SQL non autorisées.



Voici un exemple simple d'une injection SQL :

Supposons que nous ayons un formulaire de connexion qui demande à l'utilisateur de saisir son nom d'utilisateur et son mot de passe pour se connecter à un site web. La requête SQL associée pour vérifier les informations d'identification de l'utilisateur pourrait ressembler à ceci :

```
● ● ●
1 $query = "SELECT * FROM users WHERE username='$username' AND password='$password'";
```

Dans cet exemple, si l'utilisateur saisit dans le champ du nom d'utilisateur la valeur suivante : `` OR 1=1 -- ` , la requête SQL deviendra :

```
● ● ●
1 $query = "SELECT * FROM users WHERE username='` OR 1=1 --' AND password='$password'";
```

L'attaquant a inséré le code `` OR 1=1 -- ` , qui est une condition toujours vraie ($1=1$) et les deux tirets (--), qui commentent le reste de la requête. Cela permet à l'attaquant de contourner la protection. En conséquence, l'attaque d'injection SQL peut entraîner des conséquences graves telles que l'accès non autorisé à des données sensibles, la modification ou la suppression de données, ou même la compromission complète de la base de données.

Il est donc essentiel de mettre en place des mesures de sécurité appropriées, telles que l'utilisation de requêtes préparées, l'échappement des données utilisateur, la validation des

entrées et l'utilisation d'ORM (Object-Relational Mapping), pour prévenir les injections SQL et protéger les applications contre ces attaques.

On essaye d'éviter cette vulnérabilité par :

7.2.1.1. Utilisation de requêtes préparées (Prepared Statements):

Laravel utilise des requêtes préparées par défaut pour toutes les requêtes SQL. Les requêtes préparées permettent de séparer les instructions SQL des données fournies par l'utilisateur, ce qui empêche les attaquants d'injecter du code SQL malveillant. Les valeurs fournies par les utilisateurs sont traitées en tant que paramètres et ne sont pas directement concaténées dans la requête SQL, réduisant ainsi le risque d'injection SQL.



```

1 //check email
2         $user = User::where('email',$fields['email'])->first();
3 //check password
4 if(!$user || !Hash::check($fields['password'],$user->password)){
5     return response([
6         'message' => 'Email or Password are incorrect'
7     ],404);
8 }
```

La ligne \$user = User::where('email', \$fields['email'])->first(); qui prend en charge les requêtes préparées par défaut. Cela aide à protéger contre les injections SQL en liant les valeurs de manière sécurisée plutôt que d'incorporer directement les variables dans la requête.

Utilisation de la fonction Hash::check() : La condition if (!\$user || !Hash::check(\$fields['password'], \$user->password)) vérifie si l'utilisateur n'existe pas ou si le mot de passe fourni ne correspond pas au mot de passe haché stocké dans la base de données. Ces éléments suggèrent que le code est conçu pour résister aux attaques par injection SQL.

i. Utilisation de l'ORM Eloquent :

Laravel encourage l'utilisation de l'ORM Eloquent pour interagir avec la base de données. Eloquent utilise des requêtes préparées par défaut et effectue automatiquement l'échappement des valeurs fournies par les utilisateurs. Cela signifie qu'on peut construire des requêtes de manière sécurisée sans soucier des injections SQL.



```

1  public function index()
2      {
3
4          $adm = Administrateur::with('user')
5              ->with('Etablissement')
6              ->get();
7
8          return $adm;
9      }

```

7.2.1.2. Les attaques XSS

L'utilisation de la bibliothèque Laravel Sanctum qui permet de Sanctum qui peut contribuer à prévenir les attaques XSS (cross-site scripting). Sanctum protège les jetons utilisés pour l'authentification des utilisateurs en utilisant des mécanismes de sécurité tels que l'utilisation de cookies HTTP et la protection CSRF. Il valide également les requêtes entrantes pour s'assurer qu'elles sont légitimes et authentifiées, ce qui prévient les attaques XSS. De plus, il est important de valider et de filtrer correctement les données utilisateur pour éviter l'injection de scripts malveillants.

En plus de Sanctum on va utiliser un autre package de laravel qui s'appelle Spatie CSP :

Le package Laravel "Spatie CSP" (Content Security Policy) joue un rôle essentiel dans la sécurisation des applications web développées avec Laravel. Il permet d'implémenter facilement une politique de sécurité des contenus (Content Security Policy) pour réduire les risques d'injections de code malveillant tels que les attaques de type Cross-Site Scripting (XSS).

```

PS C:\Users\OMIN\Desktop\project\xtradh> composer require spatie/laravel-csp
Warning: Accessing repo.packagist.org over http which is an insecure protocol.
Info from http://repo.packagist.org: #StandWithUkraine
./composer.json has been created
Running composer update spatie/laravel-csp
Loading composer repositories with package information
Warning: Accessing repo.packagist.org over http which is an insecure protocol.
Updating dependencies
Lock file operations: 26 installs, 0 updates, 0 removals

```

La Content Security Policy est une couche de sécurité qui spécifie les sources légitimes à partir desquelles une application web peut charger des ressources externes. Cela inclut les scripts JavaScript, les feuilles de style CSS, les images, les polices, etc. En limitant les sources autorisées, la politique de sécurité des contenus renforce la protection contre les attaques XSS en empêchant l'exécution de code non autorisé provenant de sources malveillantes.

Le package "Spatie CSP" facilite la mise en place de la politique de sécurité des contenus dans Laravel. Il offre des fonctionnalités pratiques, notamment la définition aisée des directives CSP dans le fichier de configuration de l'application. Il permet également de générer des rapports automatiques en cas de violation de la politique CSP, offrant ainsi une visibilité sur les tentatives d'injections de code malveillant.

Grâce aux middlewares fournis par le package, il est possible d'appliquer la politique de sécurité des contenus aux routes spécifiques de l'application, permettant ainsi un contrôle plus granulaire. Cela assure une protection optimale contre les attaques XSS et renforce la sécurité de l'application Laravel.

```
● ● ●

1 Route::get('/api/get-nonce', function () {
2     $nonce = app(NonceGenerator::class)->generate();
3
4     return response()->json([
5         'nonce' => $nonce,
6     ]);
7 });


```

```
● ● ●

1 class Basic extends Policy
2 {
3     public function configure()
4     {
5         $this
6             ->addDirective(Directive::BASE, Keyword::SELF)
7             ->addDirective(Directive::CONNECT, Keyword::SELF)
8             ->addDirective(Directive::DEFAULT, Keyword::SELF)
9             ->addDirective(Directive::FORM_ACTION, Keyword::SELF)
10            ->addDirective(Directive::IMG, Keyword::SELF)
11            ->addDirective(Directive::MEDIA, Keyword::SELF)
12            ->addDirective(Directive::OBJECT, Keyword::NONE)
13            ->addDirective(Directive::SCRIPT, Keyword::SELF)
14            ->addDirective(Directive::STYLE, Keyword::SELF)
15            ->addNonceForDirective(Directive::SCRIPT)
16            ->addNonceForDirective(Directive::STYLE);
17     }
18 }
19


```

L'appel à la méthode `addNonceForDirective(Directive::STYLE)` dans le contexte de la politique de sécurité des contenus (Content Security Policy) permet d'ajouter un nonce (number used once) à la directive spécifiée.

Dans le cas spécifique de `Directive::STYLE`, cela signifie que chaque balise `<style>` incluse dans le contenu HTML de la page doit être accompagnée d'un attribut `nonce` contenant une valeur unique et aléatoire générée côté serveur. Cette valeur de nonce doit correspondre à la valeur spécifiée dans la politique CSP.

Lorsque le navigateur rencontre une balise `<style>` avec l'attribut `nonce`, il vérifie si la valeur du nonce correspond à celle spécifiée dans la politique CSP. Si c'est le cas, le style sera exécuté, sinon il sera bloqué. Cela permet de contrôler précisément les sources autorisées pour les

styles appliqués à la page et de prévenir les attaques d'injection de code malveillant dans les styles.

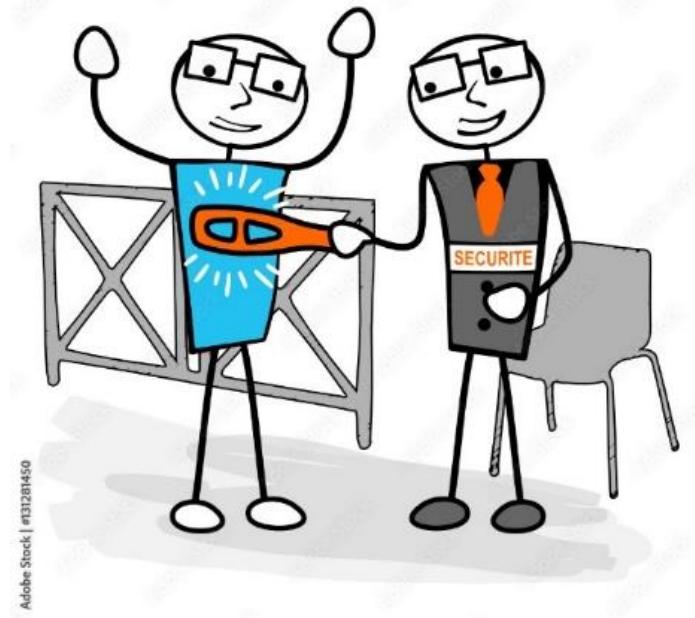
L'ajout d'un nonce pour la directive `Directive::STYLE` est une pratique de sécurité recommandée pour renforcer la protection contre les attaques XSS basées sur l'injection de styles malveillants. En utilisant des nonces, on s'assure que seuls les styles provenant du serveur sont exécutés, empêchant ainsi les styles externes non autorisés de modifier l'apparence de la page et de compromettre la sécurité de l'application.

7.3. Sécurité du code côté client (Frontend) :

En étroite collaboration avec le responsable du FrontEnd, nous avons travaillé ensemble pour mettre en place un système de sécurité solide dans notre application Vue.js. Pour renforcer la sécurité, nous avons utilisé des pratiques telles que l'utilisation d'attributs nonce pour protéger les ressources chargées, ainsi que la bibliothèque Axios pour gérer les requêtes HTTP en bénéficiant de ses fonctionnalités de sécurité intégrées. Nous avons veillé à ne pas inclure de données sensibles dans les réponses envoyées au client, minimisant ainsi les risques d'exposition d'informations confidentielles. De plus, nous avons mis en place des contrôles stricts sur les entrées utilisateur afin de prévenir les attaques d'injection de code et de garantir l'intégrité des données. Enfin, nous avons configuré les autorisations et les rôles utilisateur de manière à restreindre l'accès aux différentes interfaces, assurant ainsi que chaque utilisateur ne peut voir que les fonctionnalités qui lui sont autorisées. Grâce à ces mesures de sécurité, nous avons créé un environnement sûr où la confidentialité et l'intégrité des données sont préservées, et où les utilisateurs ne peuvent accéder qu'aux fonctionnalités qui leur sont attribuées.

7.3.1. Utilisation d'attributs nonce

On utilise l'attribut `nonce` lors de la création d'éléments `<script>` et `<style>`. Cela permet de renforcer la politique de sécurité des ressources en spécifiant un nombre aléatoire (nonce).



Cette pratique contribue à prévenir les attaques de type cross-site scripting (XSS) en limitant l'exécution de code malveillant provenant de sources non autorisées

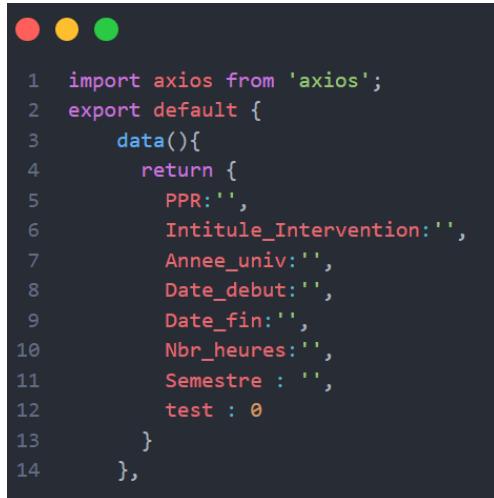
```

1  getNonce() {
2      axios.get('/api/get-nonce')
3          .then(response => {
4              const nonce = response.data.nonce;
5
6              const scriptElement = document.createElement('script');
7              scriptElement.setAttribute('nonce', nonce);
8              scriptElement.src = 'index.js';
9              document.head.appendChild(scriptElement);
10
11             const styleElement = document.createElement('style');
12             styleElement.setAttribute('nonce', nonce);
13             styleElement.innerHTML =
14                 `.my-style {
15                     color: red;
16                 }
17             `;
18             document.head.appendChild(styleElement);
19         })
20         .catch(error => {
21             console.error('Erreur lors de la récupération du nonce:', error);
22         });
23     },
24     created(){
25         this.getNonce();
26     }

```

7.3.2. Utilisation d'Axios

On fait appel à la bibliothèque Axios pour effectuer des requêtes HTTP. Axios est réputé pour sa sécurité et facilite la protection contre les attaques telles que l'injection de code malveillant. Cette approche témoigne d'une prise en compte des bonnes pratiques de sécurité lors des interactions avec des ressources externes.



```
1 import axios from 'axios';
2 export default {
3     data(){
4         return {
5             PPR:'',
6             Intitule_Intervention:'',
7             Annee_univ:'',
8             Date_debut:'',
9             Date_fin:'',
10            Nbr_heures:'',
11            Semestre : '',
12            test : 0
13        }
14    },
15}
```

7.3.3. Absence de données sensibles

Dans le code, aucune donnée sensible telle que des mots de passe ou des informations d'identification n'est stockée en clair. Cette bonne pratique renforce la sécurité globale de l'application en limitant les risques de compromission des informations sensibles.

7.3.4. Contrôle des entrées

On mentionne une méthode `handleSubmit()` qui implique une interaction avec l'utilisateur. Si cette méthode met en oeuvre une validation appropriée des entrées utilisateur avant de les traiter, cela renforce la sécurité en prévenant les attaques d'injection ou de manipulation de données.

```

1  async handlesubmit(){
2      const response = await axios.post('/ajoutinterventiontab',{
3          PPR:this.PPR,
4          Intitule_Intervention:this.Intitule_Intervention,
5          Annee_univ:this.Annee_univ,
6          Date_debut:this.Date_debut,
7          Date_fin:this.Date_fin,
8          Nbr_heures:this.Nbr_heures,
9          Semestre : this.Semestre
10         }).then(()=>{
11             this.PPR='',
12             this.Intitule_Intervention='',
13             this.Annee_univ='',
14             this.Date_debut='',
15             this.Date_fin='',
16             this.Nbr_heures='',
17             this.Semestre ''
18             this.test = 1 ;
19         })
20         console.log(response)
21     }
22 }
23 }
24

```

7.3.5. Contrôle d'accès restreignant les interfaces utilisateur

```

1  path: '/Dash_au',
2  name: 'Dash_au',
3  component:Dash_au,
4  beforeEnter: (to, from, next) => {
5
6      if(!store.getters['auth/authenticated']){
7          return next({ name: 'Login' })
8      }
9      const user=store.getters['auth/user'];
10     if(user.type!=='admin_univ'){
11         return next({name:'NotFound2'})
12     }
13     next()

```

Le code de navigation guard utilisé dans Vue Router pour restreindre l'accès à une route spécifique. Dans cet exemple, le guard vérifie si l'utilisateur est authentifié et si son type est "admin_univ" avant de lui permettre d'accéder à la route. Sinon, il redirige l'utilisateur vers une autre route (ceci est valable pour toutes autres types d'utilisateur).

Explication de code :

- Lorsque la route est accédée, la fonction `beforeEnter` est exécutée.

- Dans la fonction `beforeEnter`, elle vérifie si l'utilisateur est authentifié en utilisant `store.getters['auth/authenticated']` qui semble être une méthode pour vérifier si l'utilisateur est authentifié.
- Si l'utilisateur n'est pas authentifié, il est redirigé vers la route "Login" en utilisant `next({ name: 'Login' })`.
- Si l'utilisateur est authentifié, la fonction `beforeEnter` récupère le type d'utilisateur à partir du store en utilisant `store.getters['auth/user']`.
- Si le type d'utilisateur n'est pas "admin_univ", l'utilisateur est redirigé vers la route "NotFound2" en utilisant `next({ name: 'NotFound2' })`.
- Si l'utilisateur est authentifié et a le type "admin_univ", il est autorisé à accéder à la route en utilisant `next()`. Cela garantit que seuls les utilisateurs authentifiés avec le type "admin_univ" peuvent accéder à cette route spécifique, sinon ils seront redirigés vers une autre route.

En conclusion, la sécurisation d'une application web de gestion des heures supplémentaires des professeurs universitaires est essentielle pour protéger les données sensibles, prévenir les attaques, respecter les réglementations en matière de confidentialité et établir la confiance des utilisateurs. En investissant dans la sécurité de votre application, vous créez un environnement fiable où les professeurs universitaires peuvent gérer leurs heures supplémentaires en toute confiance et où leurs informations personnelles et financières sont protégées de manière adéquate.

8. CHAPITRE 8 :

TESTS

Ce présent chapitre, consacré aux tests joue un rôle essentiel dans la qualité et la fiabilité de l'application. Il vise à identifier et à corriger les erreurs, les problèmes de fonctionnalité et les défauts qui peuvent survenir pendant le développement. Pour cela, différentes techniques de tests sont utilisées, telles que les tests unitaires, les tests d'intégration, les tests de système, les tests de performance, etc.

8.1. Présentation :

Dans la partie suivante, nous allons nous plonger dans les tests du backend, en mettant l'accent sur les tests des contrôleurs. Les tests des contrôleurs sont d'une importance primordiale, car ils permettent de valider toutes les méthodes développées. Ils jouent un rôle fondamental dans la vérification du bon fonctionnement de notre application du point de vue du backend. Nous examinerons en détail chaque contrôleur et nous nous assurerons que toutes les fonctionnalités sont testées de manière rigoureuse. En réalisant ces tests approfondis, nous pourrons avoir confiance en la robustesse et la fiabilité de notre backend.

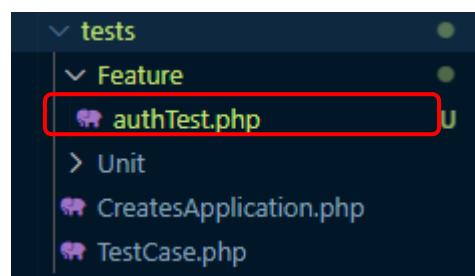
Dans cette section, nous allons effectuer les tests de notre backend en utilisant PHPUnit, un outil de test unitaire pour PHP largement reconnu dans la communauté de développement. PHPUnit nous offre une plateforme solide pour réaliser une variété de tests, y compris les tests des contrôleurs. En utilisant PHPUnit, nous pourrons effectuer des tests unitaires pour valider le bon fonctionnement des méthodes individuelles, ainsi que des tests d'intégration pour vérifier les interactions entre les différentes parties du backend. PHPUnit constitue un élément essentiel de notre suite de tests et nous permettra de garantir la qualité et la stabilité de notre backend.

8.2. Préparation de l'environnement :

8.2.1. Emplacement :

Pour des raisons d'homogénéité et de facilité de gestion, nous avons décidé d'opter pour l'utilisation du dossier "features" pour l'implémentation de tous nos tests, y compris les tests unitaires. Cette approche nous permet de maintenir une structure cohérente où tous les tests sont regroupés au même endroit.

Bien que les tests unitaires et les tests fonctionnels (feature tests) aient des objectifs légèrement différents, nous avons constaté qu'il n'y avait pas de différence de comportement significative lors de leur exécution. De plus, les tests sont déjà regroupés en classes distinctes, ce qui facilite leur organisation au sein du dossier "features".



8.2.2. Class Tests et méthodes setUp/tearDown:

```

1 class Testauthcontroller extends TestCase
2 {
3     use DatabaseMigrations;
4
5     public function setUp(): void
6     {
7         parent::setUp();
8         $this->artisan('migrate');
9         $type = "Enseignant";
10        $email = "testtest666@testing.com";
11        $password = "test";
12
13        $response = $this->post('/api/user', [
14            'type' => $type,
15            'email' => $email,
16            'password' => $password,
17            'password_confirmation' => $password,
18        ]);
19
20    }

```

La Fonction « **setUp** », nous permet de préparer l'environnement avec l'executon de chaque test, et donc avant chaque execusion y en a une migration qui s'effectue suivie d'une insertion des données qui seront utilisés par la suite dans les classes des tests, en général la méthode se déclenche avant le début du test.

La méthode « **tearDown** » et au contraire de la précédente, elle s'exécute après les tests, on l'utilisera généralement pour écraser le contenu de la base de données pour le bon fonctionnement de la méthode **setUp** dans les autres tests qui suivent.

```

1 public function tearDown(): void
2 {
3     $this->artisan('migrate:rollback');
4     parent::tearDown();
5 }

```

- ✓ On utilise aussi « **DatabaseMigrations** » pour forcer dans le cas d'échec la restitution de la base de données à son état initial.

8.3. Catégories de Tests :

Pour résumer la situation, on a choisi de catégoriser les tests en 3 subdivisions majeures :

- Les tests simples (insertion directe).
- Les tests réduits (utilisation des « factories »).
- Les tests avancés ou complexes (intermédiaires).

Les tests simples et réduits seront en général des tests unitaires, ou des tests de fonctionnalités qui sont basés sur l'utilisation de scripts minimisés, tandis que pour les tests avancés ça va comprendre des tests d'intégration qui nécessitent l'appel de différentes composantes et fonctionnalités.

Le fonctionnement des tests repose sur la vérification ***Logique*** des différentes assertions du type :

- AssertIfEquals
- AssertDatabaseHas
- AssertDatabaseMissing
- AssertStatus...

8.3.1. Test simple :

Comme l'indique son nom, il s'agit des tests les plus simples à faire et la majorité d'entre eux, font partie des tests unitaires :



```

1  public function testAuthLoginOffPassword()
2  {
3
4      $email = "testtest666@testing.com";
5      $password = "testfalse";
6
7      $response = $this->post('/api/login', [
8          'email' => $email,
9          'password' => $password,
10     ]);
11
12     $response->assertStatus(404);
13
14 }

```

- ✓ Comme illustré dans l'image ci-contre, il s'agit des tests auquel on spécifie l'ensemble des données qu'on va utiliser nous même avec précision

```
$email = "testtest666@testing.com";
$password = "testfalse";
```

```
$response = $this->post('/api/login', [
    'email' => $email,
    'password' => $password,
]);
```

- ✓ On envoie les données en utilisant la méthode post vers l' « end-point » qui va récupérer le résultat de cette requête.

- ✓ Et on vérifie si la réponse du serveur est bien celle qu'on attendait.

```
$response->assertStatus(404);
```

8.3.2. Test réduit :

Ce sont les tests qui se basent sur l'utilisation des données générées aléatoirement en utilisant des « factories » :

```
1 public function testStoreAdministrateur()
2 {
3
4     $admin=Administrateur::factory()->create();
5     $admin = $admin->toArray();
6     print_r($admin);
7     $response = $this->post('/api/administrateur', $admin);
8     $response->assertStatus(200);
9     $this->assertNotNull($response);
10    $response->assertStatus(200)
11    ->assertJsonStructure([
12        'PPR',
13        'Nom',
14        'prenom',
15        'Etablissement',
16        'id_user',
17    ]);
18    $this->assertDatabaseHas('administrateurs', $admin);
19
20 }
```

- ✓ On utilisera le modèle « Administrateur » pour en créer une instance au niveau de la base de données, pour s'assurer de la structure de l'objet on pourrait utiliser « print_r » pour l'afficher

```
$admin=Administrateur::factory()->create();
$admin = $admin->toArray();
print_r($admin);
```

- ✓ Pour le renvoi de données, il s'avère qu'il est plus simple que celui des méthodes

simples vu qu'on envoie directement un objet bien construit au niveau des méthodes de l'api.

```
$response = $this->post('/api/administrateur', $admin);
```

- ✓ On vérifie la réponse du serveur d'une façon qui est un peu similaire, en vérifiant la structure de l'objet renvoyé par la méthode ainsi que le statut du serveur.

```
$response->assertStatus(200);
$this->assertNotNull($response);
$response->assertStatus(200)
->assertJsonStructure([
    'PPR',
    'Nom',
    'prenom',
    'Etablissement',
    'id_user',
]);
$this->assertDatabaseHas('administrateurs', $admin);

}
```

8.3.3. Test avancé :

Ce genre de texte s'avère un peu plus compliqué que le reste, puisqu'il s'agit de l'intégration de plusieurs fonctionnalités ou méthodes développées au niveau de l'application et nécessitant une alternance entre l'utilisation d'une insertion directe ou l'on renvoie les données d'une façon explicite et précise.

En même temps l'utilisation des « factories » quelques parts, pour gagner en termes de longueur du code malgré sa complexité, il existe 2 approches différentes, cependant on va les traiter vers la fin (Voir : Section.12):

- ✓ Voici un petit exemple simple et conçis permettant de l'illustrer :

```

1  public function testUpdateProf(){
2      $user = User::factory()->create([
3          'email' => 'oldemail@example.com',
4          'password' => bcrypt('oldpassword'),
5          'type' => 'oldtype',
6      ]);
7      Sanctum::actingAs($user);
8      $prof=enseignant::factory()->create();
9      $id=$prof['id_user'];
10     $response = $this->patch('/api/updateprof/'.$id, [
11         'email' => 'newemail@example.com',
12         'password' => 'newpassword',
13         'type' => 'newtype',
14         'PPR' => '666666',
15         'Nom' => 'testcheck',
16         'prenom'=>'retestcheck',
17         'Date_Naissance' => "1980-01-01",
18         'Etablissement' => 1,
19         'id_Grade' => 1,
20     ]);
21     $response->assertStatus(200);
22     $this->assertDatabaseHas('enseignants', [
23         'id_user' => $id,
24         'PPR' => '666666',
25         'Nom' => 'testcheck',
26         'prenom'=>'retestcheck',
27         'Date_Naissance' => "1980-01-01",
28         'Etablissement' => 1,
29         'id_Grade' => 1,
30     ]);
31 }
32 }
```

- ✓ On utilise dans un premier temps le modèle « User » avec une précision de donnée pour plus d'exactitude, vu que les données seront reliées entre eux au niveau du test

```

$user = User::factory()->create([
    'email' => 'oldemail@example.com',
    'password' => bcrypt('oldpassword'),
    'type' => 'oldtype',
]);
```

```

Sanctum::actingAs($user);
$prof=enseignant::factory()->create();
$id=$prof['id_user'];
$response = $this->patch('/api/updateprof/'.$id, [
    'email' => 'newemail@example.com',
    'password' => 'newpassword',
    'type' => 'newtype',
    'PPR' => '666666',
    'Nom' => 'testcheck',
    'prenom'=>'retestcheck',
    'Date_Naissance' => "1980-01-01",
    'Etablissement' => 1,
    'id_Grade' => 1,
]);

```

- ✓ On utilise tout d'abord une authentification en utilisant la fonction « actingAs » du « Sanctum », qui nous permet de générer un token pour qu'on puisse accéder à cet « Endpoint ».

- ✓ Ensuite nous allons créer un objet de type « Enseignant » en réutilisant les « factories », et juste après on va essayer de lancer une requête afin qu'on puisse modifier dans cette situation l'objet de type « Enseignant » qu'on a créé au niveau de la base de données mais en spécifiant d'une façon explicite l'intégralité des données à modifier.

- ✓ Pour la vérification des réponses du serveur, c'est à peu près la même chose au niveau des 3 types de tests, on utilise des assertions pour s'assurer de la conformité des réponses à nos attentes !

```

$response->assertStatus(200);
$this->assertDatabaseHas('enseignants', [
    'id_user' => $id,
    'PPR' => '666666',
    'Nom' => 'testcheck',
    'prenom'=>'retestcheck',
    'Date_Naissance' => "1980-01-01",
    'Etablissement' => 1,
    'id_Grade' => 1,
]);
}

```

8.4. Tests du contrôleur d'Authentification :

8.4.1. Erreurs rencontrées :

En ce qui concerne le contrôleur d'« Authentification », Une seule erreur a été rencontrée, ainsi que les tests étaient effectués de deux façons différentes :

- ✓ Dans un premier lieu on a opté pour une vision dans le cas où les données

renseignées existent bien en base de données.

- ✓ Dans la seconde approche nous avons prévu, une insertion des données inexistantes au sein de la base de données pour vérifier l'interruption du processus de connexion.

Voici un extrait de deux scripts parmi tant d'autres permettant de vérifier les choses d'une telle façon :

- ✓ Utilisation de vraies données :

```

1 public function testAuthLoginOn()
2 {
3     $email = "testtest666@testing.com";
4     $password = "test";
5
6     $response = $this->post('/api/login', [
7         'email' => $email,
8         'password' => $password,
9     ]);
10
11     $response->assertStatus(202);
12
13     $response->assertJsonStructure([
14         'user' => [
15             'id_user',
16             'type',
17             'email',
18             'created_at',
19             'updated_at',
20         ],
21         'token',
22     ]);

```

```

1 public function testAuthLoginOffPassword()
2 {
3
4     $email = "testtest666@testing.com";
5     $password = "testfalse";
6
7     $response = $this->post('/api/login', [
8         'email' => $email,
9         'password' => $password,
10    ]);
11
12     $response->assertStatus(404);
13
14 }

```

- ✓ Utilisation de données inexistantes :

- ✓ La méthode « register » du contrôleur Authentification semble avoir une erreur au niveau des champs précisés pour la table “users”, puisque cette dernière ne contient pas de colonne avec ce nom.

```

1  public function testUpdateUser()
2  {
3      $user = User::factory()->create([
4          'email' => 'oldemail@example.com',
5          'password' => bcrypt('oldpassword'),
6          'type' => 'oldtype',
7      ]);
8      Sanctum::actingAs($user);
9      $userid=$user->id_user;
10
11     $response = $this->put('/api/user/'.$userid, [
12         'email' => 'newemail@example.com',
13         'password' => 'newpassword',
14         'type' => 'newtype',
15     ]);
16     $response->assertStatus(202);
17     $this->assertDatabaseHas('users', [
18         'id_user' => $userid,
19         'email' => 'newemail@example.com',
20         'type' => 'newtype',
21     ]);
22
23 }

```

```

FAIL Testauthcontroller
✗ auth registration

FAIL Tests\Feature\ExampleTest
✗ example
---

● Testauthcontroller > auth registration
Expected response status code [202] but received 404.
Failed asserting that 202 is identical to 404.

at C:\Users\Hp Victus\Documents\testprojetencore\XtraDh\Backend\tests\Unit\authTest.php:28
24 |         'password_confirmation' => $password,
25 |
26 |
27 |     // Vérifier le code de statut de la réponse
→ 28 |     $response->assertStatus(202);
29 |
30 |     // Vérifier si l'utilisateur a été créé dans la base de données
31 |     $this->assertDatabaseHas('users', [
32 |         'name' => $name,
33 |

1  C:\Users\Hp Victus\Documents\testprojetencore\XtraDh\Backend\vendor\phpunit\phpunit\phpunit:107
PHPUnit\TextUI\Command::main()

```

8.4.2. Résultats de l'exécution des tests :

- ❖ Auth Registration : méthode permettant de créer un utilisateur d'une façon simple.
- ❖ Auth Login on : méthode permettant la connexion (l'authentification) en utilisant des données correctes.
- ❖ Auth Login off password : méthode qui permet de vérifier que la connexion est interrompue lors de la saisie d'un faux mot de passe.
- ❖ Auth Login off email : méthode qui permet de vérifier que la connexion est interrompue lors de la saisie d'un fausse adresse mail.
- ❖ Logout : méthode qui permet de vérifier le bon fonctionnement de la méthode de déconnexion (destruction du token !)
- ✓ Résultat : 5/5 Tests ont été exécutés avec succès annonçant le bon fonctionnement des méthodes suivantes :

```
PS C:\Users\Hp Victus\Documents\pull11\xtradh> php artisan test
Warning: TTY mode is not supported on Windows platform.
2|ynvOrbvKq0z9Cxe7CYQMCXKh4ZZkrpjPPgk3Iuq
  PASS Testauthcontroller
    ✓ auth registration
    ✓ auth login on
    ✓ auth login off password
    ✓ auth login off email
    ✓ logout

  Tests: 5 passed
  Time: 1.31s
```

8.5. Tests du contrôleur User :

8.5.1. Erreurs rencontrées :

Lors de l'exécution des tests du contrôleur « User », trois erreurs ont été remontés :

- Méthode Show (1) du contrôleur User :

- ✓ Au niveau du contrôleur “User” que l'on remarque que cette méthode affiche tous les utilisateurs au lieu de se limiter seulement à l'affichage de l'utilisateur voulu.

- ❖ Méthode où l'on a rencontré le problème



```

1  public function show($id)
2  {
3      return response()->json(
4          user::find($id)
5              ->with(['administrateur'])
6              ->with(['enseignant'])
7              ->get());
8
9  }
10

```

```

[0] => Array
(
    [id_user] => 1
    [email] => testtest1@testing.com
    [type] => Enseignant
    [created_at] => 2023-05-21T23:04:47.000000Z
    [updated_at] => 2023-05-21T23:04:47.000000Z
    [administrateur] =>
        [
            [enseignant] =>
        ]
)

[1] => Array
(
    [id_user] => 2
    [email] => testtest2@testing.com
    [type] => haha
    [created_at] => 2023-05-21T23:04:47.000000Z
    [updated_at] => 2023-05-21T23:04:47.000000Z
    [administrateur] =>
        [
            [enseignant] =>
        ]
)

```

• **TestUserController > show user**
Failed asserting that an array has the key 'id_user'.

```

at C:\Users\Hp Victus\Documents\pull26\XtraDh\tests\Feature\userControllerTest.php:85
  81 |     $response->assertStatus(200);
  82 |
  83 |     $jsonResponse = $response->json();
  84 |     $response->assertStatus(200)
→ 85 |     ->assertJsonStructure([
  86 |         'id_user',
  87 |         'email',
  88 |         'type',
  89 |         'created_at',
  90 |
  91 |     )

```

1 C:\Users\Hp Victus\Documents\pull26\XtraDh\vendor\phpunit\phpunit\phpunit:107
PHPUnit\TextUI\Command::main()

- ❖ Le script suivant nous a permis de retrouver l'erreur :

```

1  public function testShowUser()
2      {
3          $id = 1;
4
5          $response = $this->get('api/user/' . $id);
6
7          $response->assertStatus(200);
8
9          $jsonResponse = $response->json();
10         $response->assertStatus(200)
11         ->assertJsonStructure([
12             'id_user',
13             'email',
14             'type',
15             'created_at',
16             'updated_at',
17         ]);
18     }

```

- ✓ Le code se base sur la vérification de l'objet retourné, s'il a la structure d'un objet de type « User » et non pas un tableau d'objets « User ».

- **Méthode Update du contrôleur User :**

D'après le code d'erreur rencontré, il s'avère que le problème est dû, au fait que la méthode « Auth()->user() », ne marche pas, un message d'erreur “Can't call currentAccessToken() on null object”.

```

FAIL TestUserController
✓ user store
✓ show user
✓ index user
✓ destroy user
✗ update user

---
• TestUserController > update user

```

```

1  public function update (Request $request, $id)
2      $fields = $request->validate([
3          'email' => 'required|email', 'password' => 'string|confirmed required', 'type' => 'required'
4      ]);
5      $user = User::find($id);
6      $user->update($fields);
7      // $token = $user->createToken ('MyAppToken')->plainTextToken;
8      $response= [
9          'user'=>$user,
10         'token' =>Auth::user()->currentAccessToken()
11     ];
12     return response($response, 202);
13 }
```

- ✓ Ci-joint le code ayant détecté l'erreur ci-dessus :

```

1  public function testUpdateUser()
2  {
3      $user = User::factory()->create([
4          'email' => 'oldemail@example.com',
5          'password' => bcrypt('oldpassword'),
6          'type' => 'oldtype',
7      ]);
8      Sanctum::actingAs($user);
9      $userid=$user->id_user;
10
11      $response = $this->put('/api/user/'.$userid, [
12          'email' => 'newemail@example.com',
13          'password' => 'newpassword',
14          'type' => 'newtype',
15      ]);
16      $response->assertStatus(202);
17      $this->assertDatabaseHas('users', [
18          'id_user' => $userid,
19          'email' => 'newemail@example.com',
20          'type' => 'newtype',
21      ]);
22
23 }
```

- ✓ Malgré qu'on s'est connecté en utilisant un token Sanctum, on arrivait pas à repérer l'utilisateur concerné :

8.5.2. Résultats des tests Après la résolution :

```

✓ index user
✓ destroy user
✓ update user
✓ store administrateur etb
✓ store enseignant etb
✓ update prof
✓ update admin
✓ destroy prof
✓ destroy admin
✓ ajout intervention

Tests: 12 passed
Time: 3.46s

PS C:\Users\Hp Victus\Documents\pull11\xtradh> 

```

- ✓ 12/12 Tests ont été exécutés avec succès.

8.6. Tests du contrôleur d'Interventions :

8.6.1. Erreurs rencontrées :

- ✓ Il s'avère d'après l'un des tests exécutés sur les méthodes du contrôleur "interventions" qu'il existe une petite erreur au niveau de la colonne "id_Intervention" de la base de données qui doit être entièrement minuscule au niveau des fichiers de migrations pour être avoir la même écriture que ceux déjà présents sur les fichiers du contrôleur/Model :

```

FAIL TestInterventioncontroller
✗ intervention store
✗ show intervention
✓ index intervention
✗ destroy intervention
✗ update intervention

```

```

● TestInterventioncontroller > intervention store
Expected response status code [200] but received 500.

The following exception occurred during the request:

PDOException: SQLSTATE[42703]: Undefined column: 7 ERREUR: la colonne « id_intervention » n'existe pas
LINE 1: ...1, $2, $3, $4, $5, $6, $7, $8, $9, $10) returning "id_interv...
^
HINT: Peut-être que vous souhaitez référencer la colonne « interventions.id_Intervention ». in C:\Users
Stack trace:

```

- ✓ Apparemment au niveau de la méthode show, on constate qu'elle affiche toutes les interventions et non pas seulement l'intervention spécifié par l' id.

```

● ● ●

1  public function show(Intervention $intervention)
2  {
3      return response()->json(
4          $intervention->with(['etablissement:id, Nom'])
5              ->with(['enseignant:id, PPR, Nom, prenom'])
6                  ->get()
7      );
8  }

```

```

[1] => Array
(
    [id_Intervention] => 2
    [id_Intervenant] => 2
    [id_Etab] => 1
    [Intitule_Intervention] => Test Intervention
    [Annee_univ] => 2022-2023
    [Semestre] => automne
    [Date_debut] => 2023-01-01
    [Date_fin] => 2023-02-28
    [Nbr_heures] => 40
    [visa_etb] => 0
    [visa_ue] => 0
    [created_at] => 2023-05-20T23:14:51.00000Z
    [updated_at] => 2023-05-20T23:14:51.00000Z
    [etablissement] => Array
        (
            [id] => 1
            [Nom] => Groupe scolaire L'élite
        )
)

```

- ❖ Le code qui a permis de remonter l'erreur est le suivant :

```

1  public function testShowIntervention()
2  {
3      $id = 1;
4
5      $response = $this->get('api/intervention/' . $id);
6
7      $response->assertStatus(200);
8
9      $jsonResponse = $response->json();
10     $response->assertStatus(200)
11     ->assertJsonStructure([
12         '*' => [
13             'id_Intervenant',
14             'id_Etab',
15             'Intitule_Intervention',
16             'Annee_univ',
17             'Semestre',
18             'Date_debut',
19             'Date_fin',
20             'Nbr_heures',]
21
22     ]);
23 }

```

- ✓ Au niveau des deux méthodes de validation des colonnes « Visa » et d'après l'erreur affichée lors de l'exécution des deux tests, on retrouve que le problème des deux méthodes est dû aux routes de ces derniers qui semblent être erronées.

```

• TestInterventioncontroller > invalider_ue
Expected response status code [200] but received 404.
Failed asserting that 200 is identical to 404.

at C:\Users\Hp Victus\Documents\pull26\XtraDh\tests\Feature\IntervControllerTest.php:294
290 |
291 |     {
292 |         $interv = Intervention::factory()->create();
293 |         $id=$interv['id_intervention'];
294 |         $response=$this->get('/api/valideruae/'.$id);
295 |         $response->assertStatus(200);
296 |         $this->assertDatabaseHas('interventions',[

297 |             'id_intervention'=>$id,
|             'visa_ue' => 1,

```

- **TestInterventioncontroller > valider uae**
 Expected response status code [200] but received 404.
 Failed asserting that 200 is identical to 404.

```
at C:\Users\Hp Victus\Documents\pull26\XtraDh\tests\Feature\IntervControllerTest.php:282
278     {
279         $interv = Intervention::factory()->create();
280         $id=$interv['id_intervention'];
281         $response=$this->get('/api/valideruae/'.$id);
→282         $response->assertStatus(200);
283         $this->assertDatabaseHas('interventions',[  

284             'id_intervention'=>$id,  

285             'visa_uae' => 1,
```

```
Route::get('/valideruae',[InterventionController::class,'valideruae']);
Route::get('/invalideruae',[InterventionController::class,'invalideruae']);
//});
```

❖ Les scripts ayant contribué à retrouver ces erreurs sont les suivants :

```
1 public function testValiderUae()
2 {
3     $interv = Intervention::factory()->create();
4     $id=$interv['id_intervention'];
5     $response=$this->get('/api/valideruae/'.$id);
6     $response->assertStatus(200);
7     $response->assertJson([
8         'id_intervention'=>$id,
9         'visa_uae' => 1,
10    ]);
11 }
```

```
1 public function testInvaliderUae()
2 {
3     $interv = Intervention::factory()->create();
4     $id=$interv['id_intervention'];
5     $response=$this->get('/api/valideruae/'.$id);
6     $response->assertStatus(200);
7     $response->assertJson([
8         'id_intervention'=>$id,
9         'visa_uae' => 1,
10    ]);
```

8.6.2. Résultats des tests après la résolution :

```

PASS TestInterventioncontroller
✓ intervention store
✓ show intervention
✓ index intervention
✓ destroy intervention
✓ update intervention
✓ valider uae
✓ invalider uae
✓ valider etb
✓ invalider etb
✓ directeur etab intervall
✓ directeur etab interv valid
✓ prof intervention

Tests: 12 passed
Time: 3.55s

PS C:\Users\Hp Victus\Documents\pull11\xtradh> []

```

- ✓ 12/12 Tests ont été exécutés avec succès.

8.7. Test du Contrôleur d'Enseignant :

8.7.1. Erreurs rencontrées :

- ✓ En se basant sur les résultats des tests, il s'avère qu'il existe un problème au niveau de la méthode show du contrôleur Enseignant, il s'agit de l'utilisation d'un objet de type Administrateur au lieu de l'utilisation d'un de la classe enseignant pour instancier un objet du même type et sur lequel la méthode sera utilisable.
- ✓ D'un autre côté, on remarque que la méthode show, au lieu d'afficher que l'enseignant spécifié à l'aide d'un "id", elle affiche tous les autres enseignants.
- ✓ La méthodes show, avant la modification, et après la première modification :

```
1 public function show($idAdm)
2 {
3     $adm = Enseignant::find($idAdm)
4         ->with(['user'])
5         ->with(['Etablissement'])
6         ->get();
7     return response()->json($adm);
8 }
```

```
1 public function show($idAdm)
2 {
3     $adm = administrateur::find($idAdm)
4         ->with(['user'])
5         ->with(['Etablissement'])
6         ->get();
7     return response()->json($adm);
8 }
```

```
FAIL TestEnseignantcontroller
✓ enseignant store
✗ show enseignant

...
• TestEnseignantcontroller > show enseignant
Expected response status code [200] but received 500.

The following exception occurred during the request:

Error: Class "App\Http\Controllers\administrateur" not found in C:\Users\Hp Victus\Documents
Stack trace:
#0 C:\Users\Hp Victus\Documents\pull21\XtraDh\Backend\vendor\laravel\framework\src\Illuminate
```

```
• TestEnseignantcontroller > show enseignant  
PHPUnit\Framework\InvalidArgumentException  
  
Argument #2 of PHPUnit\Framework\Assert::assertArrayHasKey() must be an array or ArrayAccess  
  
at C:\Users\Hp Victus\Documents\pull24lkher\XtraDh\tests\Feature\EnsgControllerTest.php:160  
156  
157     $jsonResponse = $response->json();  
158     print_r($jsonResponse);  
159     $response->assertStatus(200)  
→160     ->assertJsonStructure([  
161         '*' => [  
162             'PPR',  
163             'Nom',  
164             'prenom',
```

- ✓ Le script de test essaye de comparer la valeur de retour avec un objet de type Enseignant, cependant il retrouve un tableau d'Enseignant.
 - ✓ Le code ayant permis de récupérer l'erreur est le suivant :



```

1  public function testShowEnseignant()
2      {
3          $id = 1;
4
5          $response = $this->get('api/enseignant/' . $id);
6
7          $response->assertStatus(200);
8
9          $jsonResponse = $response->json();
10         print_r($jsonResponse);
11         $response->assertStatus(200)
12         ->assertJsonStructure([
13             'PPR',
14             'Nom',
15             'prenom',
16             'Date_Naissance',
17             'Etablissement',
18             'id_Grade',
19             'id_user',
20         ]);
21     }

```

- ✓ De même que sur les deux autres méthodes du contrôleur enseignant, au niveau de la méthode « Update » il s'agit de l'erreur due à l'utilisation de la classe administrateur au lieu de la classe Enseignant.

```

FAIL TestEnseignantcontroller
✓ enseignant store
✓ show enseignant
✓ index enseignant
✓ destroy enseignant
✗ update enseignant
```
• TestEnseignantcontroller > update enseignant
Expected response status code [200] but received 500.

The following exception occurred during the request:

Error: Class "App\Http\Controllers\administrateur" not found in C:\Users\Hp Victus\Document

```

```

1 public function update(Request $request, $idAdm)
2 {
3 $adm = administrateur::find($idAdm);
4 $attributs = $request->validate([
5 'PPR' => 'required',
6 'Nom' => 'required',
7 'prenom' => 'required',
8 'Etablissement' => 'required',
9 'id_user' => 'required'
10]);
11
12 $adm->update($attributs);
13 return response()->json($adm);
14 }

```

**Proposition :**

Ajout des autres champs ou des colonnes de la table administrateurs pour donner plus de liberté de modification et permettre la rectification en cas d'erreur par exemple.

- ✓ Le code ayant contribué au repérage de l'erreur est le suivant :

```

1 public function testUpdateEnseignant()
2 {
3 $id=1;
4 $modification = [
5 'PPR' => "123456789",
6 'Nom' => "testnom2",
7 'prenom' => "testprenom2",
8 'Date_Naissance' => "1989-01-01",
9 'Etablissement' => 1,
10 'id_Grade' => 1,
11 'id_user' => 1,
12];
13 $response = $this->put("/api/enseignant/{$id}", $modification);
14
15 $response->assertStatus(200);
16 // pour check si la table users de la bd contient une ligne avec les infos qu'on vient de modifier
17 $this->assertDatabaseHas('enseignants', [
18 'id' => $id,
19 'PPR' => '123456789'
20]);
21 // pour check si l'objet retourné est pareil à ce qu'on veut
22 $response->assertJson([
23 'PPR' => "123456789",
24 'Nom' => "testnom2",
25 'prenom' => "testprenom2",
26 'Date_Naissance' => "1989-01-01",
27 'Etablissement' => 1,
28 'id_Grade' => 1,
29 'id_user' => 1,
30]);
31 }

```

- ✓ Dernière erreur, cette fois-ci au niveau de la méthode Destroy du contrôleur

Enseignant, utilisation de la classe Administrateur au lieu de la classe Enseignant pour l'instanciation d'un objet.

```
● ● ●
1 public function destroy($idAdm)
2 {
3 return administrateur::find($idAdm)->delete();
4 }
```

```
FAIL TestEnseignantcontroller
✓ enseignant store
✓ show enseignant
✓ index enseignant
✗ destroy enseignant

• TestEnseignantcontroller > destroy enseignant
Expected response status code [200] but received 500.

The following exception occurred during the request:

Error: Class "App\Http\Controllers\administrateur" not found in C:\Users\Hp Victus\
```

- ✓ Le code qui a permis de retrouver le problème est le suivant :

```
● ● ●
1 public function destroy($idAdm)
2 {
3 return administrateur::find($idAdm)->delete();
4 }
5 public function tearDown(): void
6 {
7 $this->artisan('migrate:rollback');
8 parent::tearDown();
9 }
```

### 8.7.2. Résultats des tests après la résolution :

- ✓ 5/5 Tests ont été exécutés avec succès.

```

PASS TestEnseignantcontroller
✓ enseignant store
✓ show enseignant
✓ index enseignant
✓ destroy enseignant
✓ update enseignant

Tests: 5 passed
Time: 1.24s

PS C:\Users\Hp Victus\Documents\pull11\xtradh>

```

### 8.8. Test du contrôleur d'Etablissement :

#### 8.8.1. Erreurs rencontrées :

- ✓ Il s'avère qu'il existe un problème au niveau des méthodes du contrôleur établissement, l'erreur vient du fait qu'au niveau du model établissement, la colonne "nom" est mentionnée avec un "n" minuscule ce qui en contradiction avec la colonne présente au niveau des fichiers de migrations avec un "N" majuscule.

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Warning: TTY mode is not supported on Windows platform.

FAIL TestEtablissementcontroller
✖ etablissement store

• TestEtablissementcontroller > etablissement store
Expected response status code [200] but received 500.

The following exception occurred during the request:

PDOException: SQLSTATE[23502]: Not null violation: 7 ERREUR: une valeur NULL viole
DETAIL: La ligne en échec contient (2, la111, null, 0655967517, 418 643-3210, Casai
ndor\laravel\framework\src\Illuminate\Database\Connection.php:373

```

```

1 class etablissement extends Model
2 {
3 use HasFactory;
4
5 protected $fillable = [
6 'nom',
7 'code',
8 'Telephone',
9 'Fax',
10 'ville',
11 'Nbr_enseignants'
12];
13
14 protected $hidden = ['pivot'];
15
16 public function enseignant()
17 {
18 return $this->hasMany(enseignant::class, 'Etablissement', 'id');
19 }
20
21 public function administrateur()
22 {
23 return $this->hasMany(administrateur::class, 'Etablissement', 'id');
24 }
25
26 public function intervention()
27 {
28 return $this->hasMany(intervention::class, 'id_Etab', 'id');
29 }
30
31 public function paiement()
32 {
33 return $this->hasMany(paiement::class, 'id_Etab', 'id');
34 }

```

```

#66 {main}
Failed asserting that 200 is identical to 500.
52 |
→ 53 | $response->assertStatus(200);
54 |
55 | $this->assertDatabaseHas('etablissements', [
56 | 'code'=> $code,
57 |]);
1 C:\Users\Hp Victus\Documents\testprojetencore\XtraDh\Backend\vendor\phpunit\phpunit\phpunit:107
PHPUnit\TextUI\Command::main()

Tests: 1 failed
Time: 0.47s

```

```

FAIL TestEtablissementcontroller
✗ etablissement store

• TestEtablissementcontroller > etablissement store
Expected response status code [200] but received 500.

The following exception occurred during the request:

PDOException: SQLSTATE[23502]: Not null violation: 7 ERREUR: une valeur NULL v
DETALI: La ligne en échec contient (2, la111, null, 0655967517, 418 643-3210,
ndor\laravel\framework\src\Illuminate\Database\Connection.php:373

```

## 8.8.2. Résultats des tests après la résolution :

- ✓ 5/5 Tests ont été exécutés avec succès.

```

PASS TestEtablissementcontroller
✓ etablissement store
✓ show etablissement
✓ index etablissement
✓ destroy etablissement
✓ update etablissement

Tests: 5 passed
Time: 1.35s

PS C:\Users\Hp Victus\Documents\pull111\xtradh> []

```

## 8.9. Test du contrôleur d'Administrateur :

### 8.9.1. Erreurs rencontrées :

- ✓ A partir des résultats des tests sur le contrôleur Administrateur que la méthode show, au lieu d'afficher seulement l'administrateur qu'on lui a spécifié avec un "id", elle affiche tous les administrateurs.

```

lolArray
(
 [0] => Array
 (
 [id] => 1
 [PPR] => 456969
 [Nom] => Beer
 [prenom] => Meda
 [Etablissement] => 1
 [id_user] => 1
 [created_at] => 2023-05-23T00:12:52.000000Z
 [updated_at] => 2023-05-23T00:12:52.000000Z
 [user] => Array
 (
 [id_user] => 1
 [email] => bertha.ratke@example.net
 [type] => regular
 [created_at] => 2023-05-23T00:12:52.000000Z
 [updated_at] => 2023-05-23T00:12:52.000000Z
)

 [etablissement] => Array
 (
 [id] => 1
 [code] => 68166
 [Nom] => Pfannerstill, Morar and Ritchie
 [Telephone] => +12813017211
)
)
)

```



```

1 public function show($idAdm)
2 {
3 $adm = administrateur::find($idAdm)
4 ->with(['user'])
5 ->with(['Etablissement'])
6 ->get();
7 retu

```

- ✓ Le code qui était derrière le repérage de cette erreur est le suivant, il

essaye de vérifier que la valeur de retour est un objet et non pas un type d'objets :

```
1 public function testShowAdministrateur()
2 {
3 $id = 1;
4 $admin=Administrateur::factory()->count(5)->create();
5 $response = $this->get('api/administrateur/' . $id);
6
7 $response->assertStatus(200);
8
9 $jsonResponse =$response->json();
10 print_r($jsonResponse);
11
12 $this->assertNotNull($jsonResponse);
13 $response->assertStatus(200)
14 ->assertJsonStructure([
15 'PPR',
16 'Nom',
17 'prenom',
18 'id_user'
19]);
20 }
```

## 8.10. Test du contrôleur Paiement :

### 8.10.1. Erreurs rencontrées :

- ✓ D'après les résultats, on retrouve qu'un niveau de la méthode store,

utilisé dans la fonction Update Paiement, les 2 champs, "Brut" existe toujours , alors qu'on l'a passé en colonne calculable :

- ✓ Le code derrière la détection de cette erreur est le suivant :

```

1 public function store(Request $request)
2 {
3 $fields = $request->validate([
4 'id_Intervenant' => 'required|exists:enseignants,id',
5 'id_Etab' => 'required|exists:etablissements,id',
6 'VH' => 'required',
7 'Taux_H' => 'required',
8 'Annee_univ' => 'required',
9 'Semestre' => 'required',
10 'Brut' => 'required',
11 'IR' => 'required',
12]);
13
14 $paiement = new Paiement();
15 $paiement->fill($fields);
16 $paiement->Taux_H = $fields["Taux_H"];
17
18 return $paiement->save();
19 }
```

```

• TestPaiementcontroller > update paiement
Expected response status code [200] but received 302.

The following errors occurred during the request:

The brut field is required.

Failed asserting that 200 is identical to 302.

at C:\Users\Hp Victus\Documents\pull10\XtraDh\tests\Feature\PaiementControllerTest.php:109
```

```

1 public function testUpdatePaiement()
2 {
3 $userid=1;
4 $paiement0=Paiement::factory()->create();
5
6 $paiement1=Paiement::factory()->create();
7
8 $paiement1 = $paiement1->toArray();
9 $paiement0 = $paiement0->toArray();
10
11 $response = $this->put("/api/paiement/{$userid}", $paiement1);
12
13 $response->assertStatus(200);
14 $this->assertNotNull($response);
15 $this->assertDatabaseHas('paiements',$paiement1);
16 $this->assertDatabaseMissing('paiements', $paiement0);
17
18 }

```

- ✓ Lorsque le script avait lancé une tentative de modification du paiement, il a du rencontrer l'erreur mentionée juste avant.
- ✓ Deuxième erreur, cette fois-ci en se basant sur le résultat du test, on retrouve qu'au niveau des factories et plus précisement celle du modèle Paiement, elle contient toujours le champ "Brut", alors qu'on l'a passé avant en colonne calculable.

```

1 class PaiementFactory extends Factory
2 {
3 protected $model = Paiement::class;
4
5 public function definition()
6 {
7 $enseignant = Enseignant::factory()->create();
8 $établissement = Etablissement::factory()->create();
9
10 return [
11 'id_Intervenant' => $enseignant->id,
12 'id_Etab' => $établissement->id,
13 'VH' => $this->faker->randomFloat(2, 10, 50),
14 'Taux_H' => $this->faker->randomFloat(2, 10, 50),
15 'Brut' => $this->faker->randomFloat(2, 1000, 9000),
16 'IR' => $this->faker->randomFloat(2, 25, 45),
17 'Annee_univ' => $this->faker->randomElement(['2022', '2021', '2023']),
18 'Semestre' => $this->faker->randomElement(['Semestre 1', 'Semestre 2']),
19];
20 }
21 }

```

- `TestPaiementcontroller > update paiement`
- `Illuminate\Database\QueryException`

```
SQLSTATE[42601]: Syntax error: 7 ERREUR: ne peut pas insérer dans la colonne « Brut »
DETAIL: la colonne « Brut » est une colonne générée. (SQL: insert into "paiements" ("id_ Intervenant"
6721.28, 25.57, 2023, Semestre 2, 2023-06-10 02:06:26, 2023-06-10 02:06:26) returning "id")
```

```
1 C:\Users\Hp Victus\Documents\pull10\XtraDh\vendor\laravel\framework\src\Illuminate\Database\Connection.php:373
PDOException::("SQLSTATE[42601]: Syntax error: 7 ERREUR: ne peut pas insérer dans la colonne « Brut »
DETAIL: la colonne « Brut » est une colonne générée.")

2 C:\Users\Hp Victus\Documents\pull10\XtraDh\vendor\laravel\framework\src\Illuminate\Database\Connection.php:373
PDOStatement::execute()

Tests: 5 failed, 1 passed
Time: 1.57s
```

- ✓ Le code qui a permis de retracer l'erreur est le même que celui utilisé au niveau de la recherche d'erreur au niveau de la méthode « Update Paiement ».

### 8.10.2. Résultats des tests après la résolution :

```
1
PASS TestPaiementcontroller
✓ paiement store
✓ show user
✓ index paiement
✓ destroy paiement
✓ update paiement
✓ consult paiement etab directeur

Tests: 6 passed
Time: 1.68s
```

```
PS C:\Users\Hp Victus\Documents\pull11\xtradh> []
```

- ✓ 6/6 Tests ont été exécutées avec succès.

### 8.11. Test du contrôleur Grades :

### 8.11.1. Erreurs rencontrées :

- ✓ Aucune erreur n'a été rencontrée lors de la vérification et du repassage sur le code du contrôleur, puisque toutes les méthodes ont répondu positivement avec des valeurs de retours qui correspondent à nos attentes.

### 8.11.2. Résultats de l'exécution des tests :

```

PASS TestGradecontroller
✓ grade store
✓ show grade
✓ index grade
✓ destroy grade
✓ update grade

Tests: 5 passed
Time: 1.42s

PS C:\Users\Hp Victus\Documents\pull11\xtradh> []

```

### 8.12. Mentions honorables :

- ✓ Dans cette section dédiée aux mentions honorables, nous allons explorer un genre de scripts particulièrement fascinant et complexe. Ces scripts, dont nous allons discuter en détail, se distinguent par leurs caractéristiques uniques et leur importance puisqu'ils nous permettent d'accéder aux différentes parties du code de l'application.
- ✓ La particularité de ces scripts réside dans leur capacité à intégrer des morceaux de code qui ont le pouvoir de nous libérer de tâches fastidieuses. Ils permettent notamment d'automatiser les tests de manière complète, sans nécessiter d'intervention humaine. De plus, ces scripts parviennent à résoudre le problème inhérent à la nécessité de simuler une opération de connexion pour obtenir un « token » d'accès de base. En combinant ces fonctionnalités, ils offrent une solution efficace et automatisée pour faciliter les tests et les processus de connexion.

### 8.12.1. Première approche :

```

1 public function testUpdateAdmin(){
2 $user = User::factory()->create([
3 'email' => 'oldemail@example.com',
4 'password' => bcrypt('oldpassword'),
5 'type' => 'oldtype',
6]);
7 Sanctum::actingAs($user);
8 $admin=Administrateur::factory()->create();
9 $id=$admin['id_user'];
10 $response = $this->patch('/api/updateadm/'.$id, [
11 'id_user' => $id,
12 'email' => 'newemail@example.com',
13 'password' => 'newpassword',
14 'type' => 'newtype',
15 'PPR' => '666666',
16 'Nom' => 'testcheck',
17 'Etablissement' => 1,
18 'prenom'=>'retestcheck',
19]);
20 $response->assertStatus(200);
21 $this->assertDatabaseHas('administrateurs', [
22 'id_user' => $id,
23 'PPR' => '666666',
24 'Nom' => 'testcheck',
25 'prenom'=>'retestcheck',
26 'Etablissement' => 1,
27]);
28
29 }

```

- ✓ Prennons l'exemple de ce script, qui comme l'indique son nom est dédié au test d'une fonctionnalité développée en BackEnd qui donne le droit à un admin de modifier les informations d'un autre admin par exemple et qui regroupe plusieurs méthodes et fonctionnalités mineures.

```
$user = User::factory()->create([
 'email' => 'oldemail@example.com',
 'password' => bcrypt('oldpassword'),
 'type' => 'oldtype',
]);
```

✓ Ce morceau de code, son rôle est de déclarer ou de créer une instance d'un objet de type User, non pas seulement en Backend, mais au niveau de la base de données également.

✓ Il utilise dans un premier temps, les « factories » pour avoir une vraie instance, car si l'on ne le fait pas, faute de quoi, « User » crée sans « Token » d'accès et donc on recevra l'erreur de « Cannot call getAccessToken() » on NULL, c'est pour ça que malgré qu'on renseigne les données du façon explicite on ne pas se laisser passer l'utilisation du « User Factory » nécessaire pour la génération d'un « Token ».

```
Sanctum::actingAs($user);
$admin=Administrateur::factory()->create();
$id=$admin['id_user'];
$response = $this->patch('/api/updateadm/'.$id, [
 'id_user' => $id,
 'email' => 'newemail@example.com',
 'password' => 'newpassword',
 'type' => 'newtype',
 'PPR' => '666666',
 'Nom' => 'testcheck',
 'Etablissement' => 1,
 'prenom'=>'retestcheck',
]);
```

✓ Dans Cette partie, on commence par utiliser la méthode « Sanctum::actingAs(\$user) », cette méthode là nous permet de se détacher de la saisie des logins à chaque fois sur un utilitaire tiers comme « Postman » ou l'interface « XHR » de Mozilla firefox, pour qu'on puisse simuler l'authentification et qu'on recopie manuellement le « Token d'accès ».

✓ Juste après, on a essayé de créer un administrateur d'une façon classique, rien de spécial, sauf que le problème qu'on va rencontrer c'est que la création d'un administrateur en utilisant son modèle de base en tant que « Factory » va lui créer un « User » qui lui sera attaché implicitement, et le problème c'est qu'on ne pourra pas accéder aux champs de l'objet « User » auquel il est relié, pour remédier à ce problème, on va

intégrer la première partie mentionné ci-dessus, dans laquelle on avait explicité les données de l'utilisateur, pour se faire on va forcer l'id de cet utilisateur comme étant l'id de l'administrateur .

- ✓ Finalement, on utilise la méthode « ***patch*** » pour pouvoir envoyer une requête au serveur contenant les champs qu'on tente de modifier tout en renseignant les nouvelles valeurs.

```
$response->assertStatus(200);
$this->assertDatabaseHas('administrateurs', [
 'id_user' => $id,
 'PPR' => '666666',
 'Nom' => 'testcheck',
 'prenom'=>'retestcheck',
 'Etablissement' => 1,
]);
}
```

- ✓ Et pour clôturer le script, on procède à la vérification de la réponse du serveur en utilisant 2 assertions classiques :

- **assertStatus(200)** : pour vérifier que la requête s'est bien exécutée d'abord.
- **assertDatabaseHas...** : pour vérifier qu'au niveau de l'administrateurs modifié ( ayant le même \$id ) les champs mentionnées ont bien été modifiés et contiennent les nouvelles valeurs.

### 8.12.2. Deuxième approche :

- ✓ En raison de la longueur des scripts, de la deuxième approche nous allons le diviser en quelques parties :



```

1 public function testAjoutIntervention()
2 {
3 $code = "la123456";
4 $Nom = "Groupe scolaire L'éliteux";
5 $Telephone = "0655967517";
6 $Faxe = "418 643-3210";
7 $ville = "Casablanca";
8 $Nbr_enseignants = 50;
9
10 $response = $this->post('/api/etablissement', [
11 'code'=> $code,
12 'Nom'=> $Nom,
13 'Telephone'=> $Telephone,
14 'Faxe'=> $Faxe,
15 'ville'=> $ville,
16 'Nbr_enseignants'=> $Nbr_enseignants,
17]);
18
19 $designation = "Grade X";
20 $charge_statutaire = "50";
21 $Taux_horaire_Vocation = "100";
22
23 $response = $this->post('/api/grade', [
24 'designation'=> $designation,
25 'charge_statutaire'=> $charge_statutaire,
26 'Taux_horaire_Vocation'=> $Taux_horaire_Vocation,
27]);

```

- ✓ Première partie qui est simple et ressemble au début de l'autre approche, il s'agit de renseigner les données pour qu'on puisse avoir un contrôle total sur le comportement au niveau du script.
- ✓ Cependant est puisque cette fois ci il s'agit de la table intervention, celle-ci regroupe un nombre important de clés étrangères et donc va falloir pour chacun de ses champs qui est de ce type de refaire la même opération.

```

1 $type = "EnseignantX";
2 $email = "testtest2@testing.com";
3 $password = "test";
4 $response = $this->post('/api/user', [
5 'type' => $type,
6 'email' => $email,
7 'password' => $password,
8 'password_confirmation' => $password,
9]);
10 $PPR = "123456789";
11 $Nom = "testnom";
12 $Prenom = "testprenom";
13 $DateNaissance = "1989-01-01";
14 $Etablissement = 1;
15 $idGrade = 1;
16 $idUser = 2;
17
18 $response = $this->post('/api/enseignant', [
19 'PPR' => $PPR,
20 'Nom' => $Nom,
21 'prenom' => $Prenom,
22 'Date_Naissance' => $DateNaissance,
23 'Etablissement' => $Etablissement,
24 'id_Grade' => $idGrade,
25 'id_user' => $idUser,
26]);

```

- ✓ On continue toujours le remplissage et la saisie des données destinées à être utilisés.

```

1 $id_Intervenant =2;
2 $id_Etab = 1;
3 $Intitule_Intervention = 'Test Intervention';
4 $Annee_univ = '2022-2023';
5 $Semestre = 'automne';
6 $Date_debut = '2023-01-01';
7 $Date_fin = '2023-02-28';
8 $Nbr_heures = 40;
9
10 $type = "admin_etb";
11 $email = "testtest666@testing.com";
12 $password = "test";
13 $user = $this->post('/api/user', [
14 'type' => $type,
15 'email' => $email,
16 'password' => $password,
17 'password_confirmation' => $password,
18]);
19

```

```

1 $token=$user['token'];
2 $PPR = "123456";
3 $Nom = "tester";
4 $Prenom = "test";
5 $Etablissement = 1;
6 $idUser = 3;
7
8 $response = $this->post('/api/administrateur', [
9 'PPR' => $PPR,
10 'Nom' => $Nom,
11 'prenom' => $Prenom,
12 'Etablissement' => $Etablissement,
13 'id_user' => $idUser,
14]);

```

```

1 $PPR = "123456789";
2 $response = $this->withHeaders([
3 'Authorization' => 'Bearer ' . $token,
4]) >post('/api/ajoutinterventionetab', [
5 'PPR'=> $PPR,
6 'id_Intervenant' => $id_Intervenant,
7 'id_etab' => $id_Etab,
8 'Intitule Intervention' => $Intitule_Intervention,
9 'Annee_univ' => $Annee_univ,
10 'Semestre' => $Semestre,
11 'Date_debut' => $Date_debut,
12 'Date_fin' => $Date_fin,
13 'Nbr_heures' => $Nbr_heures
14]);

```

- ✓ Cependant ce dernier code n'est pas si simple que les autres, sa particularité, et comparé à la première approche, c'est qu'il envoie au niveau de l' « End-Point » **une Authorization Bearer ( autorisation du token )** attachée avec le point d'accès, ce qui permet entre temps de simuler le test dans des conditions réelles à 100%.



```

1 $response->assertStatus(200);
2
3 $this->assertDatabaseHas('interventions', [
4 "id_intervention" => 1,
5 "id_Intervenant"=> 1,
6 "id_Etab"=> 1,
7 "Intitule_Intervention"=> "Test Intervention",
8 "Annee_univ"=> "2022-2023",
9 "Semestre"=> "automne",
10 "Date_debut"=> "2023-01-01",
11 "Date_fin"=> "2023-02-28",
12 "Nbr_heures"=> 40,
13 "visa_etb"=> 0,
14 "visa_uee"=> 0,
15]);
16
17 $this->assertNotNull($response);
18 }
```

- ✓ Et Pour clôturer avec la deuxième approche on finit sur des assertions logique comme pour l'autre afin de vérifier si les réponses du serveur sont conformes à nos attentes.
- ✓ On remarque l'utilisation de « **assertNotNull** », chose à signaler c'est que malgré que parfois des tests ont pour même objectifs, cependant ils n'ont pas des valeurs de retour identiques, ce qui revient à vérifier la valeur de retour de la fonction de base en premier lieu avant de juste se lancer et utiliser n'importe qu'elle assertions.

### 8.13. Bilan des Tests Backend :

- ✓ Nous arrivons à la fin de cette partie dédiée aux tests. Après avoir exploré en détail les scripts, il est maintenant temps de faire un récapitulatif.
- ✓ Dans cette partie, nous passerons en revue les points saillants, les enseignements clés et les résultats obtenus. Préparez-vous à tirer des conclusions éclairantes et à découvrir les perspectives offertes par cette exploration approfondie des tests et des scripts associés ayant contribué énormément à purger le code de ses erreurs.

```

✓ show user
✓ index user
✓ destroy user
✓ update user
✓ store administrateur etb
✓ store enseignant etb
✓ update prof
✓ update admin
✓ destroy prof
✓ destroy admin
✓ ajout intervention

Tests: 70 passed
Time: 17.46s

PS C:\Users\Hp Victus\Documents\pullxd\XtraDh> []

```

```

)
..... 70 / 70 (100%) 3|9wP7hFfn3aZ44Q5QFTtiNJKHGzVzHNtyBWg9R1aO
Time: 00:17.125, Memory: 40.00 MB
OK (70 tests, 428 assertions)
PS C:\Users\Hp Victus\Documents\pullxd\XtraDh> []

```

En résumé :

- ✓ Les erreurs rencontrées, ont été remontés aux développeurs sur « Jira ».
- ✓ Plus de 70 Scripts différents et uniques.
- ✓ Plus de 70 méthodes et fonctionnalités testés.
- ✓ Plus de 420 assertions avec une moyenne de 6 assertions par test.
- ✓ Taux de réussite qui s'élève à 100 après l'élimination des erreurs !

## 8.14. Présentation de Playwright :

Playwright est une bibliothèque puissante et polyvalente conçue pour automatiser les tests et les interactions avec les navigateurs web. Il offre une solution complète pour la gestion des navigateurs, permettant aux développeurs de contrôler facilement Chrome, Firefox, Safari et d'autres navigateurs populaires à partir d'un seul et même environnement de programmation. Avec Playwright, on peut écrire des scripts de test ou des automatisations complexes qui simulent les interactions humaines avec les applications web, tels que le clic sur des boutons, la saisie de texte, la navigation, le remplissage de formulaires, et bien plus encore. Grâce à sa capacité à prendre en charge plusieurs navigateurs et plates-formes, Playwright est une option attrayante surtout pour la portabilité des tests ou de l'automatisation des applications web. Que ce soit pour tester la fonctionnalité d'un site web, effectuer des tâches répétitives ou encore réaliser des scénarios de test complexes, Playwright offre une approche efficace et facile à utiliser pour automatiser les interactions avec les navigateurs web.

### 8.14.1. Préparation de l'environnement :

Tout d'abord avant de se lancer dans la création des scripts et leurs exécutions, on doit tout d'abord préparer l'environnement et pour se faire on commence par la création d'un dossier dédié à Playwright uniquement, après ne reste plus qu'à se placer au niveau de ce dossier et d'exécuter la commande suivante :

Pour Installer Playwright au niveau du système :

```
npm install playwright
```

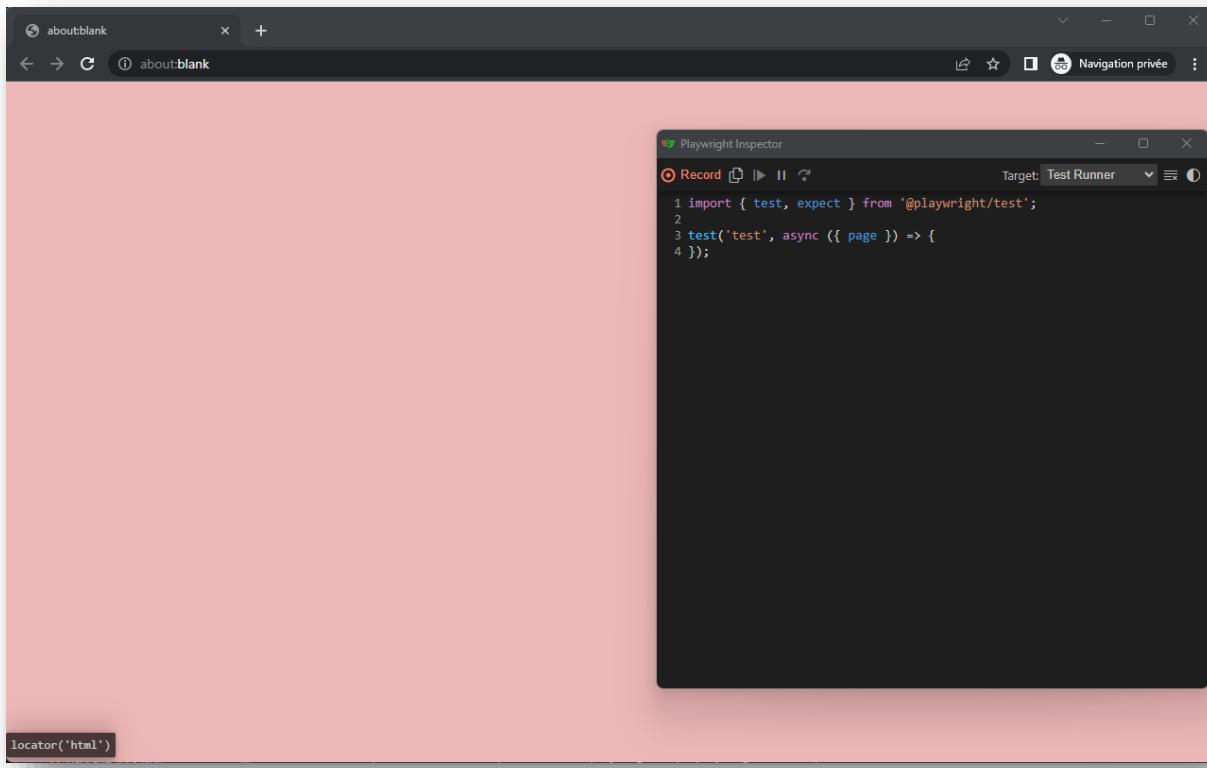
Pour installer Playwright au niveau du dossier de travail :

```
npm init playwright@latest
```

### 8.14.2. L'utilisation du codegen :

Le « **Codegen** » est une fonctionnalité de Playwright intégrée, qui permet d'accéder à partir d'un navigateur ( basé sur **chromium** ) à une page quelconque et de repérer ses éléments à partir des sélecteurs CSS/Texte, pour

le faire il suffit d'exécuter la commande suivante : `npx playwright codegen`  
Ça nous permet d'aboutir à des interfaces du type :



### 8.14.3. Options supplémentaires :

On peut Activer le « Trace Viewer » qui est une fonction parfaite, pour l'enregistrement d'une trace qui simule le fonctionnement du script en temps réels permettant d'avoir un aperçu de ses résultats et parfois même dans le cas d'échec de repérer l'action qui cause le problème, pour l'activer on remet au niveau du fichier config.js de Playwright le « Trace » en 'on' :

```

 1 /* Collect trace when
 2 trace: 'on',
 3 },
 4
 1 /* Collect trace when retrying
 2 trace: 'on-first-retry',
 3 },
 4

```

On peut même spécifier le navigateur/l'appareil sur lesquels on veut exécuter les tests :



```
1 projects: [
2 {
3 name: 'chromium',
4 use: { ...devices['Desktop Chrome'] },
5 },
6 {
7 name: 'firefox',
8 use: { ...devices['Desktop Firefox'] },
9 },
10 },
11]
```

Finalement pour exécuter un test sur

```
npx playwright test
```

Playwright il suffit de lancer la commande suivante :

## 8.15. Tests des interfaces

### 8.15.1. Tests de l'interface d'Authentification :

The screenshot shows a test results interface with a dark background. At the top, there's a header for 'TestAuth.spec.js'. Below it, a list of six tests is shown, each with a green checkmark indicating success. The tests are categorized by browser: chromium, firefox, and webkit. Each entry includes the test name, browser name in a colored button, the file path 'TestAuth.spec.js', and the execution time.

| Test Name                  | Browser  | File Path           | Time |
|----------------------------|----------|---------------------|------|
| AuthentificationCorrecte   | chromium | TestAuth.spec.js:3  | 5.8s |
| AuthentificationIncorrecte | chromium | TestAuth.spec.js:42 | 3.7s |
| AuthentificationCorrecte   | firefox  | TestAuth.spec.js:3  | 4.8s |
| AuthentificationIncorrecte | firefox  | TestAuth.spec.js:42 | 4.8s |
| AuthentificationCorrecte   | webkit   | TestAuth.spec.js:3  | 4.2s |
| AuthentificationIncorrecte | webkit   | TestAuth.spec.js:42 | 2.8s |

- ✓ Comme le montre l'image ci-contre les tests d'authentification ont été exécutés avec succès, il s'agit de deux tests différents :
  - Le premier est un test dans lequel les données saisies sont valides et donc il est censé nous rediriger vers le site web et repérer le bouton de déconnexion pour le quitter :

On vérifie à l'aide du Trace Viewer que le test a bien pu se connecter et qu'il arrive à repérer le bouton de déconnexion :

```
● ● ●
1 import { test, expect } from '@playwright/test';
2
3 test('AuthentificationCorrecte', async ({ page }) => {
4 await page.goto('http://localhost:8080/');
5 // on vérifie l'existence du formulaire d'ab
6 const elementCount = await page.locator('div').filter({ hasText: 'Email AddressPasswordInfo Forgot Password' });
7 await expect(elementCount).toBeGreaterThan(0);
8
9 const checkimg = await page.locator('img').first();
10 await expect(checkimg).toBeVisible();
11
12 const checkemail = await page.getText('Email Address');
13 await expect(checkemail).toBeVisible();
14 const checkpwd = await page.getText('Password', { exact: true });
15 await expect(checkpwd).toBeVisible();
16 const checkform = await page.getText('Email AddressPasswordInfo Forgot Password ? Log In');
17 await expect(checkform).toBeVisible();
18
19 const checkfoot = await page.getText('© 2023 UAE - All Rights Reserved.');
20 await expect(checkfoot).toBeVisible();
21
22 const checkem = await page.getByPlaceholder('Enter Email Address');
23 await expect(checkem).toBeVisible();
24 await page.getByPlaceholder('Enter Email Address').fill('tahaamine2002@gmail.com');
25
26 await page.getByPlaceholder('Enter Password').press('Tab');
27 await expect(checkfoot).toBeVisible();
28
29 await page.getByPlaceholder('Enter Password').fill('test');
30
31 const checklogin = await page.getByRole('button', { name: 'Log In' });
32 await expect(checklogin).toBeVisible();
33
34 await page.getByRole('button', { name: 'Log In' }).click();
35 // On check si la connexion est bien passé en faisant un check sur le bouton de déco s'il existe
36 const checkdeco = await page.getText('Déconnexion');
37 await expect(checkdeco).toBeVisible();
38 await page.getText('Déconnexion').click();
39
40 });

});
```

The screenshot shows a Playwright test run for 'AuthentificationCorrecte'. The timeline at the top indicates the sequence of actions: 'Before Hooks' (2.3s), 'page.goto' (0.77ms), 'locator.count' (2.7ms), 'expect.toBeGreaterThan' (1ms), 'expect.toBeVisible' (14ms), 'getByText' (7ms), 'expect.toBeVisible' (5ms), 'getByText' (4ms), 'expect.toBeVisible' (6ms), 'getByPlaceholder' (6ms), 'locator.fill' (28ms), 'expect.toBeVisible' (13ms), 'locator.click' (70ms), 'expect.toBeVisible' (2.95s), 'locator.click' (61ms), and 'After Hooks' (1ms). The browser window displays a list of administrators ('Listes des admins d'établissements') with columns: EMAIL, TYPE, PPR, NOM, PRENOM, ETABLISSEMENT, and actions (Edit, Delete). The 'Déconnexion' button is visible in the bottom-left corner of the browser window.

Le bouton de déconnexion est bien visible suite à la connexion au site et après avoir suivi l'enchainement du code.

```
1 test('AuthentificationCorrecte', async ({ page }) => {
2 await page.goto('http://localhost:8080/');
3 // on vérifie l'existence du formulaire d'ab
```

On commence par instancier une page et on utilise les différentes méthodes qu'offre les bibliothèques de Playwright, par exemple « `goto()` » qui nous permet de se rediriger vers une autre page en saisissant l'URL de la page qu'on veut atteindre.

```
1 const checkimg = await page.locator('img').first();
2 await expect(checkimg).toBeVisible();
```

On peut utiliser la méthode « `toBeVisible()` » par exemple pour pouvoir détecter la visibilité d'une composante, un teste ou même un sélecteur CSS.

La méthode « `fill()` », nous permet d'effectuer un remplissage d'un champ ou des différents

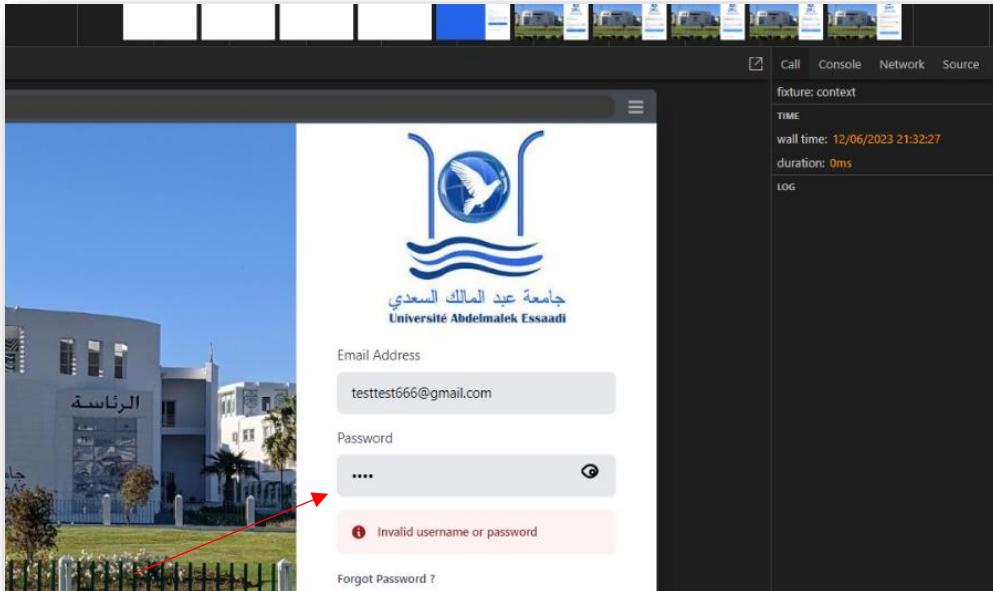
```
● ● ●
1 await page.getByPlaceholder('Enter Email Address').fill('tahaamine2002@gmail.com');
```

« inputs » d'un formulaire pour saisir des données par exemple afin de simuler une saisir de logins de connexion dans notre situation.

```
● ● ●
1 await page.getByPlaceholder('Enter Password').press('Tab');
```

On peut utiliser la méthode « `press()` » pour simuler d'une façon général l'utilisation des boutons du clavier.

- Le deuxième test d'authentification, concerne une authentification qui se caractérise avec une saisir de données erronées afin de détecter le bon fonctionnement de la gestion des erreurs et des alertes :



```
PS C:\Users\Hp Victus\Documents\pull10\XtraDh\playwright> npx playwright test tests/TestAuth.spec.js

Running 6 tests using 6 workers
6 passed (8.2s)

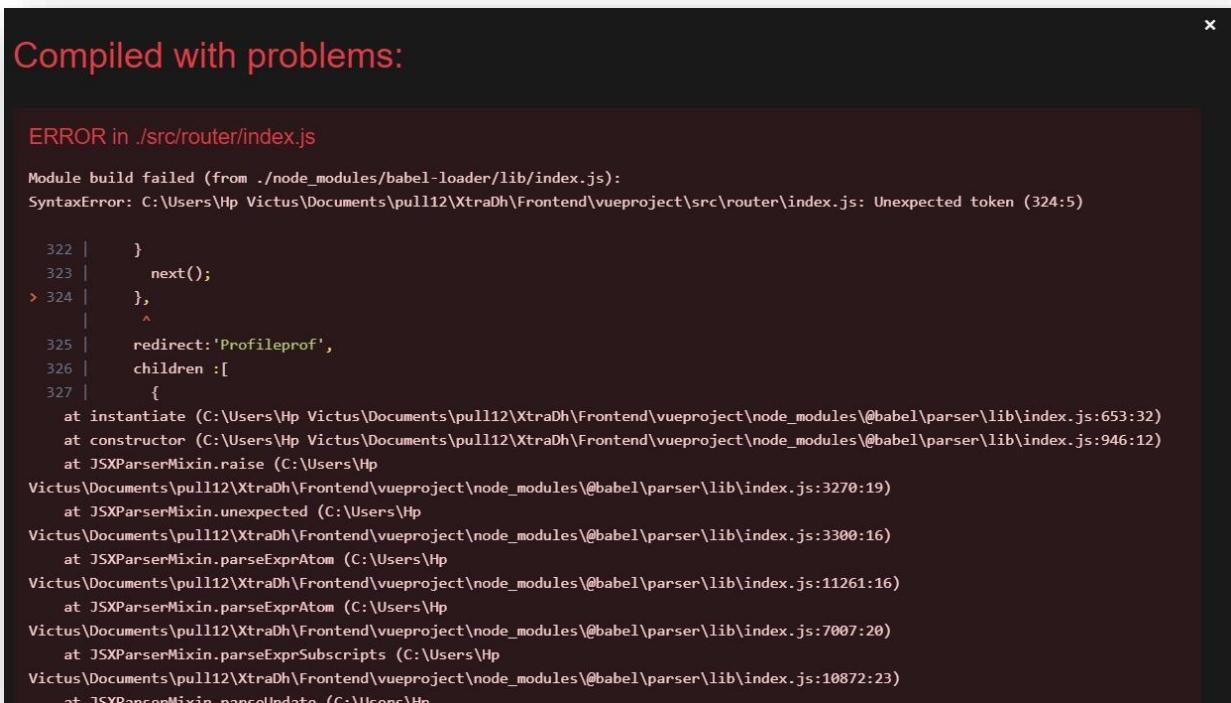
To open last HTML report run:

npx playwright show-report

○ PS C:\Users\Hp Victus\Documents\pull10\XtraDh\playwright> npx playwright show-report[]
```

### 8.15.2. Test des interfaces du professeur :

Lors de l'exécution des tests des interfaces au niveau de celle des professeurs, seulement une erreur a été rencontrée :



L'erreur indique qu'il existe un caractère de plus au niveau du routeur des interfaces frontales

Après la résolution des problèmes, le test s'est exécuté avec succès :

The screenshot displays the Playwright test runner interface. At the top, it shows a tree view of test results for 'Professeur.spec.js' under 'test chromium' and 'test firefox', both marked with green checkmarks. Below this is a detailed timeline of actions taken during the test, including navigating to a URL, filling input fields, and clicking buttons. To the right of the timeline is a screenshot of a web application titled 'Professeur' showing a list of interventions. The bottom right corner contains a log pane with test metadata: wall time (12/06/2023 23:49:02), duration (0ms), and a LOG section.

| Actions                                                                           | Metadata |
|-----------------------------------------------------------------------------------|----------|
| > Before Hooks                                                                    | 1.7s     |
| page.goto <a href="http://localhost:8080/interp">http://localhost:8080/interp</a> | 748ms    |
| locator.fill <code>getByPlaceholder("Nom")</code>                                 | 43ms     |
| locator.press <code>getByPlaceholder("Nom")</code>                                | 15ms     |
| locator.fill <code>getByPlaceholder("Prénom")</code>                              | 18ms     |
| locator.press <code>getByPlaceholder("Prénom")</code>                             | 18ms     |
| locator.click <code>getByText("Prof")</code>                                      | 1.4s     |
| locator.click <code>getByText("amin")</code>                                      | 44ms     |
| locator.click <code>getByRole("img")</code>                                       | 47ms     |
| locator.click <code>locator('div').fill("")</code>                                | 47ms     |
| locator.click <code>locator('#app').click()</code>                                | 262ms    |
| locator.click <code>locator('#app div').click()</code>                            | 44ms     |
| locator.click <code>locator('form div').click()</code>                            | 49ms     |
| locator.click <code>getByRole("button")</code>                                    | 48ms     |
| locator.click <code>getByText("Oui")</code>                                       | 43ms     |
| locator.click <code>locator('mb-4')</code>                                        | 46ms     |
| locator.fill <code>locator('mb-4 &gt; input')</code>                              | 13ms     |
| locator.click <code>getByRole("button")</code>                                    | 46ms     |
| locator.click <code>locator('mb-4 &gt; button')</code>                            | 47ms     |
| locator.click <code>getByRole("button")</code>                                    | 46ms     |
| page.screenshot                                                                   | 92ms     |
| expect.toMatchSnapshot                                                            | 124ms    |
| locator.click <code>getByRole("link")</code>                                      | 60ms     |
| locator.click <code>getByRole("cell")</code>                                      | 57ms     |
| locator.click <code>getByRole("cell")</code>                                      | 40ms     |
| locator.click <code>getByRole("cell")</code>                                      | 45ms     |

On se retrouve au niveau du « Trace Viewer » vers la dernière interface visitée au niveau du script avec la détection du bouton de déconnexion.

```
PS C:\Users\Hp Victus\Documents\pull12\XtraDh\playwright> npx playwright test tests/Professeur.spec.js
Running 2 tests using 2 workers
 2 passed (6.1s)

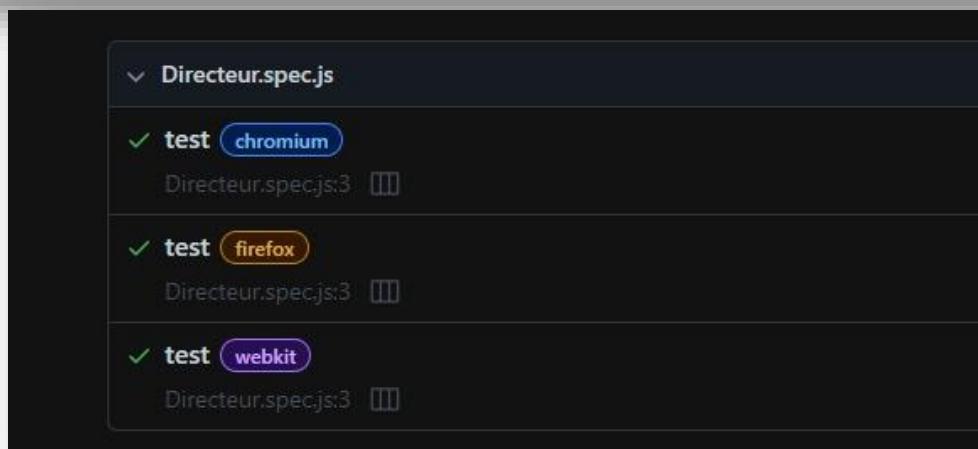
To open last HTML report run:
 npx playwright show-report
○ PS C:\Users\Hp Victus\Documents\pull12\XtraDh\playwright> []
```

### 8.15.3. Test des interfaces du directeur d'établissement :

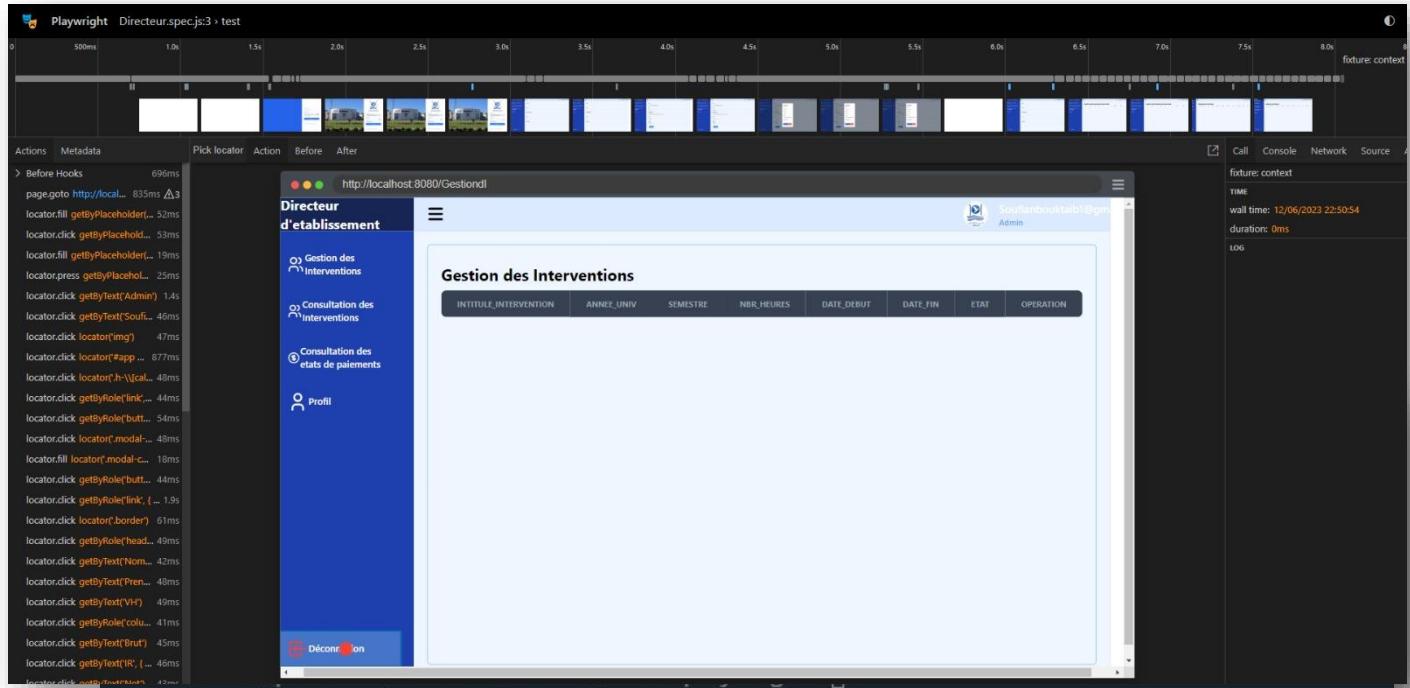
En se basant sur les résultats retournés par les scripts de tests utilisés sur les interfaces du directeur d'établissement il s'avère que ces dernières ne contiennent aucune erreur explicite, à part une erreur au niveau d'un élément qui ne s'affiche pas correctement : Après la résolution de l'erreur remonté, les tests s'exécutent avec succès :

```
=====
51 | await page.getByRole('columnheader', { name: 'Etat' }).click();
52 | await page.getByRole('columnheader', { name: 'Operation' }).click();
53 | await page.getByRole('button', { name: 'valider', exact: true }).click();
 ^
54 | await page.getByRole('row', { name: 'Séminaire sur la gestion des ressources humaines 2022/ click();
55 | await page.getByRole('heading', { name: 'Directeur d\'établissement' }).click();
56 | await page.getText('Déconnexion').click();

at C:\Users\Hp Victus\Downloads\XtraDh\XtraDh\playwright\tests\Directeur.spec.js:53:68
```

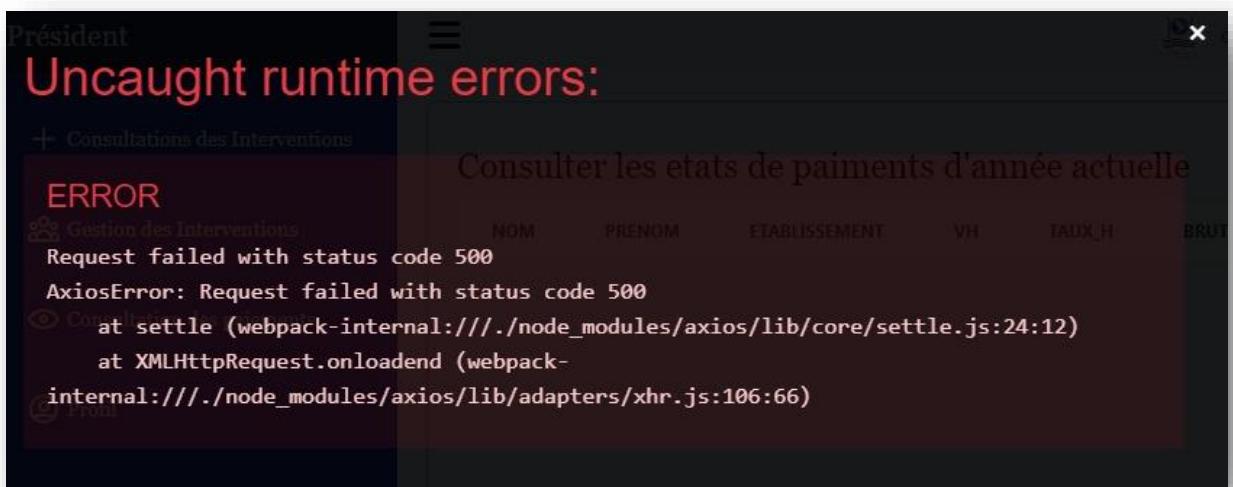


On aboutit vers la fin du test à la dernière interface indiqué au niveau du script en plus e la détection du bouton de déconnexion :



#### 8.15.4. Test des interfaces du président universitaire :

Une seule erreur seulement a été détectée au niveau des interfaces du président universitaire, il s'agit d'une erreur au niveau de l'interface de la gestion des paiements :



DevTools is now available in French! [Always match Chrome's language](#) [Switch DevTools to French](#) [Don't show again](#)

webpack-dev-server... at <DashPu onVnodeUnmounted=>fn<onVnodeUnmounted> ret=Ref< Proxy(0bject) > > at <RouterView> at <App>

Uncaught (in promise) [AxiosError](#) [ERR\\_BAD\\_RESPONSE](#)

```

 config: {transitional: {…}, adapter: Array(2), transformRequest: Array(1), transformResponse: Array(1), timeout: 0, …}
 message: "Request failed with status code 500"
 name: "AxiosError"
 request: XMLHttpRequest {onreadystatechange: null, readyState: 4, timeout: 0, withCredentials: false, upload: XMLHttpRequestUpload, …}
 response:
 config: {transitional: {…}, adapter: Array(2), transformRequest: Array(1), transformResponse: Array(1), timeout: 0, …}
 data:
 exception: "Illuminate\\Database\\QueryException"
 file: "C:\\Users\\Hp Victus\\Downloads\\XtraDh-4b2d9975868595f81fb750be9b2cc783349d8ad9\\XtraDh-4b2d9975868595f81fb750be9b2cc783349d8ad9\\vendor\\laravel\\framework\\src\\Illuminate\\Database\\Connection.php"
 line: 712
 message: "SQLSTATE[42702]: Ambiguous column: 7 ERREUR: la référence à la colonne « id » est ambiguë\nLINE 1: ...blissements\".\"id\" where \""
 trace: Array(50)
 0: {file: 'C:\\Users\\Hp Victus\\Downloads\\XtraDh-4b2d9975868595..\\framework\\src\\Illuminate\\Database\\Connection.php', line: 672, function: 'select'}
 1: {file: 'C:\\Users\\Hp Victus\\Downloads\\XtraDh-4b2d9975868595..\\framework\\src\\Illuminate\\Database\\Connection.php', line: 359, function: 'select'}
 2: {file: 'C:\\Users\\Hp Victus\\Downloads\\XtraDh-4b2d9975868595..framework\\src\\Illuminate\\Database\\Query\\Builder.php', line: 2413, function: 'select'}
 3: {file: 'C:\\Users\\Hp Victus\\Downloads\\XtraDh-4b2d9975868595..framework\\src\\Illuminate\\Database\\Query\\Builder.php', line: 2402, function: 'select'}
 4: {file: 'C:\\Users\\Hp Victus\\Downloads\\XtraDh-4b2d9975868595..framework\\src\\Illuminate\\Database\\Query\\Builder.php', line: 2936, function: 'select'}
 ...

```

En coopération avec l'un des développeurs Back-end, l'erreur a été retrouvé, il s'agit d'un accès à un élément d'un enregistrement directement sans passer par l'instance de l'objet qui contient l'enregistrement :

```

else{
 $avant = date("Y")-1;
 $apres = date("Y");
}

$date = $avant.'.'.$apres ;
$paiements = $paientement = DB::table('paiements')
 ->join('enseignants','paiements.id_Intervenant','=',$enseignants.id')
 ->join('etablissements','id_Etab','=',$etablissements.id')
 ->where('Annee_univ',$date)
 ->select('VH','Taux_H','Brut','Annee_univ','Semestre','IR','NET','enseignants.Nom','etablissements.Libelle','paiements.id')
 ->orderByDesc('id') ↓
->get();
return response()->json($paiements);
}

```

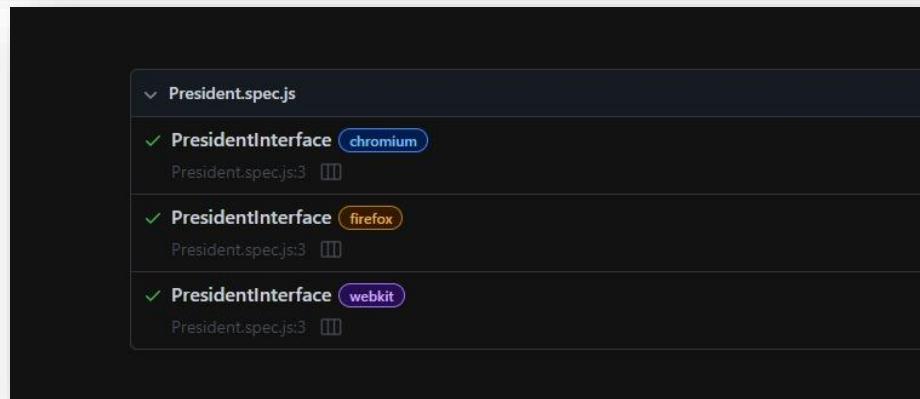
L'élément « id » de la clause « orderByDesc » doit devenir « paiements.id » pour qu'on puisse résoudre le problème.

Après la résolution du problème les tests des interfaces du président s'exécutent avec succès:

```
PS C:\Users\Hp Victus\Downloads\XtraDh\XtraDh\playwright> npx playwright test tests/President.spec.js
Running 3 tests using 3 workers
3 passed (23.2s)

To open last HTML report run:
npx playwright show-report

○ PS C:\Users\Hp Victus\Downloads\XtraDh\XtraDh\playwright> npx playwright show-report
```



The screenshot shows the Playwright test runner interface. At the top, it displays the test file "President.spec.js" and the specific test "PresidentInterface". Below this, a timeline shows the sequence of actions taken during the test. The left side of the interface lists "Actions" and "Metadata" for each step. The right side provides a detailed view of the test execution, including the browser used ("chromium"), the URL ("http://localhost:8080/Console"), and the specific interaction being recorded. A preview window shows a user interface with a table of data, and the bottom right corner shows the test's duration and log information.

### 8.15.5. Test des interfaces de l'administrateur d'établissement :

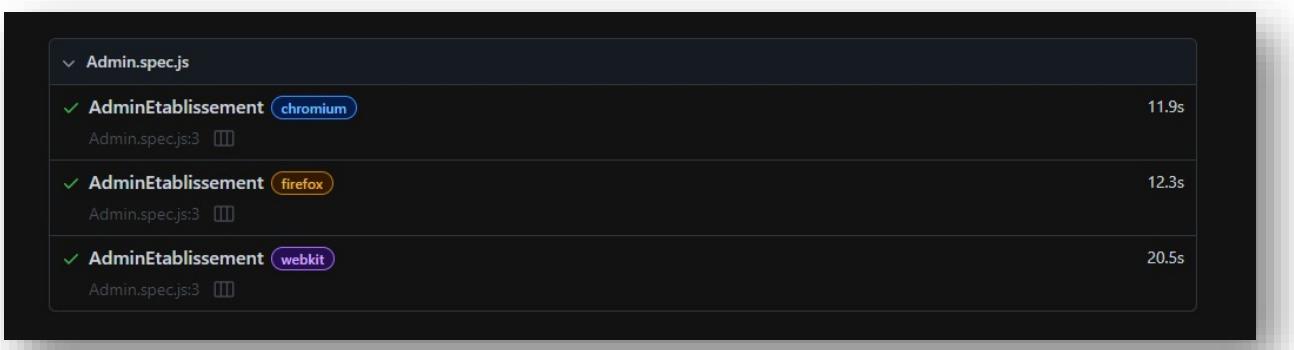
Un seule erreur seulement a été repérée suite à l'exécution des tests sur les interfaces du directeur d'établissement, il s'agit d'une erreur qui se déclenche à chaque fois qu'on essaye d'executer l'ajout d'un objet au niveau de la base de données :

Après la résolution du problème les test s'exécutent avec succès :

Finalement on se retrouve vers la dernière interface du script de test, tout en repérant

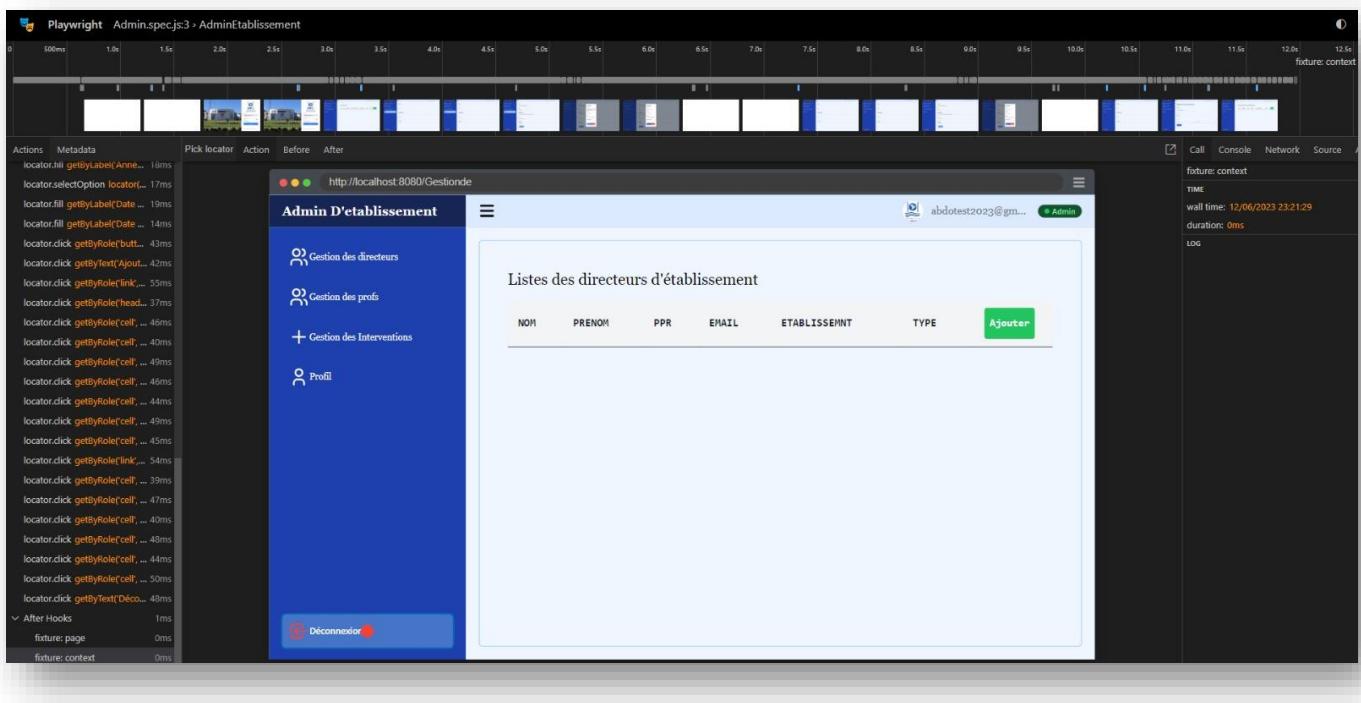
The screenshot shows a web application interface titled "Admin D'établissement". A modal dialog is open with the title "ERROR" and the message "Request failed with status code 403". Below the modal, a table titled "Listes des directeurs d'établissement" is visible, showing columns for PRENOM, PPR, EMAIL, ETABLISSEMENT, and TYPE. A green "Ajouter" button is at the bottom right of the table. The terminal window below the application shows the command "npx playwright test tests/Admin.spec.js" being run, followed by the output: "Running 3 tests using 3 workers" and "3 passed (21.5s)". It also shows the command "npx playwright show-report" and the resulting report summary table.

|   | Admin.spec.js               | AdminEtablissement chromium | 11.9s |
|---|-----------------------------|-----------------------------|-------|
| ✓ | AdminEtablissement chromium | Admin.spec.js:3             | 11.9s |
| ✓ | AdminEtablissement firefox  | Admin.spec.js:3             | 12.3s |
| ✓ | AdminEtablissement webkit   | Admin.spec.js:3             | 20.5s |



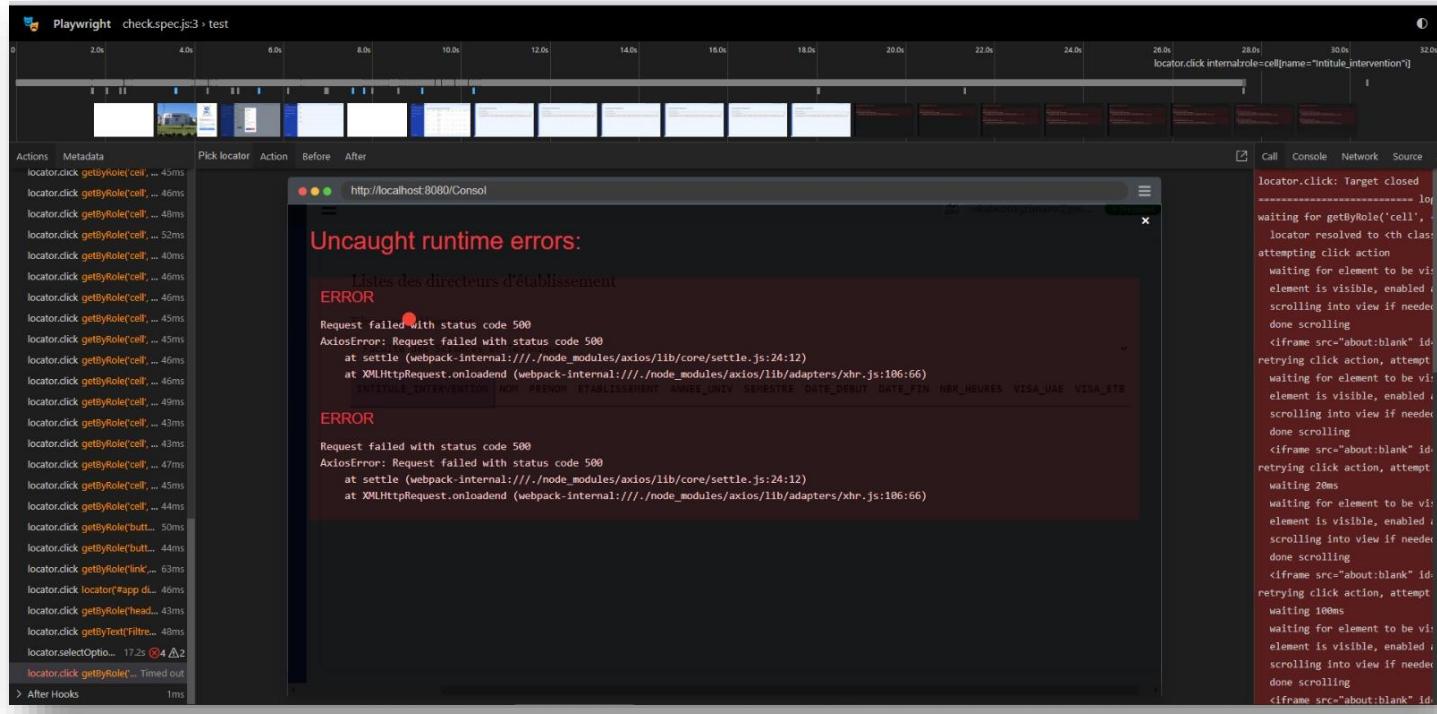
comme d'habitude le bouton de déconnexion :

### 8.15.6. Test des interfaces de l'administrateur universitaire :



Trois erreurs ont été repérées au niveau des interfaces de l'administrateur universitaire :





Le dernier problème semble venir du fait que la soumission du formulaire ne se fait pas d'une façon correcte, vu qu'on utilise « submitForm » au lieu de « submitForm » :

The screenshot shows a modal dialog box with the following HTML code:

```
<div class="flex justify-end">
 <button type="submit" class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded"
 @click="submitForm"
 Ajouter
 </button>
 <button class="bg-red-500 hover:bg-red-700 text-white font-bold py-2 px-4 rounded ml-2"
 @click="closeModal"
 Annuler
 </button>
</div>
```

Below the modal, a table row is visible with columns: Bouktaib, Soufiane, 666670, Soufianboukt, EMAIL:, Les Lettres et des Sciences Humaines, directeur\_etb, Edit, Delete. The modal contains input fields for EMAIL (empty), PPR (754191), NOM (empty), PRÉNOM (empty), and two buttons: Ajouter (blue) and Annuler (red).

Après la résolution des problèmes les test s'exécutent avec succès !

```
● PS C:\Users\Hp Victus\Documents\pull10\XtraDh\playwright> npx playwright test tests/AdminUniv.spec.js
Running 3 tests using 3 workers
3 passed (29.6s)

To open last HTML report run:
npx playwright show-report
○ PS C:\Users\Hp Victus\Documents\pull10\XtraDh\playwright> []
```

```
check.spec.js
✗ test chromium
check.spec.js:3 [30.0s]
✗ test firefox
check.spec.js:3 [30.0s]
```

Après l'exécution du test on vérifie qu'on est redirigés vers la page d'accueil.

## 8.16. Test de récupération du mot de passe :

```
AdminUniv.spec.js
✓ AdminUniversitaireInterface chromium
AdminUniv.spec.js:3 [26.1s]
✓ AdminUniversitaireInterface firefox
AdminUniv.spec.js:3 [26.3s]
✓ AdminUniversitaireInterface webkit
AdminUniv.spec.js:3 [25.2s]
```

Le test de récupération du Mot de passe s'est exécuté sans détecter d'erreurs

TestRecup.spec.js		
✓	RecupCorrecte	chromium
	TestRecup.spec.js:3	████
✓	RecupIncorrecte	chromium
	TestRecup.spec.js:55	████
✓	RecupCorrecte	firefox
	TestRecup.spec.js:3	████
✓	RecupIncorrecte	firefox
	TestRecup.spec.js:55	████
✓	RecupCorrecte	webkit
	TestRecup.spec.js:3	████
✓	RecupIncorrecte	webkit
	TestRecup.spec.js:55	████

● PS C:\Users\Hp Victus\Documents\pull10\XtraDh\playwright> npx playwright test tests/TestRecup.spec.js

Running 6 tests using 6 workers  
6 passed (32.6s)

To open last HTML report run:

npx playwright show-report

○ PS C:\Users\Hp Victus\Documents\pull10\XtraDh\playwright> []

On vérifie après l'exécution du test dans lequel on renseigne un e-mail qui est valide, qu'on

Playwright AdminUniv.spec.js:3 > AdminUniversitaireInterface

Actions Metadata Pick locator Action Before After

```
> Before Hooks 2.2s
page.goto http://localhost:804ms △3
locator.count(locator('div').filter... 31ms
expect.toBeGreaterThan 2ms
expect.toBeVisible getByRole('... 20ms
locator.click getByRole('link', { ... 60ms
expect.toBeVisible getByText('... 9ms
expect.toBeVisible getByRole('... 7ms
expect.toBeVisible getByText('... 4ms
expect.toBeVisible getByRole('... 6ms
expect.toBeVisible getByLabel('... 5ms
locator.fill getByLabel('Email') 26ms
locator.click getByRole('button', { ... 62ms
expect.toHaveURL locator('root') 8ms
page.waitForTimeout 10.0s
expect.toBeVisible getByRole('... 5ms
page.waitForTimeout 10.0s
expect.toBeVisible getByRole('... 5ms
expect.toBeVisible getByRole('... 4ms
locator.click getByRole('button', { ... 41ms
expect.toHaveURL locator('... 2.9s △3
< After Hooks 2ms
fixture:page 0ms
fixture:context 0ms
```

fixture: context

TIME  
wall time: 13/06/2023 11:34:30  
duration: 0ms  
LOG

est redirigés vers la page d'accueil :

Playwright TestRecup.spec.js:3 > RecupCorrecte

Actions Metadata Pick locator Action Before After

```
> Before Hooks 2.9s
page.goto http://localhost:804ms △3
locator.count(locator('div').filter... 39ms
expect.toBeGreaterThan 1ms
expect.toBeVisible getByRole('... 23ms
locator.click getByRole('link', { ... 78ms
expect.toBeVisible getByText('... 8ms
expect.toBeVisible getByRole('... 7ms
expect.toBeVisible getByText('... 7ms
expect.toBeVisible getByRole('... 8ms
expect.toBeVisible getByLabel('... 7ms
locator.fill getByLabel('Email') 27ms
locator.click getByRole('button', { ... 62ms
expect.toHaveURL locator('... 11ms
page.waitForTimeout 10.0s
expect.toBeVisible getByText('... 113ms
page.waitForTimeout 10.0s
expect.toBeVisible getByRole('... 8ms
expect.toBeVisible getByRole('button', { ... 6ms
locator.click getByRole('button', { ... 45ms
expect.toHaveURL locator('... 2.0s △3
< After Hooks 2ms
fixture:page 0ms
fixture:context 0ms
```

fixture: context

TIME  
wall time: 13/06/2023 11:29:32  
duration: 0ms  
LOG

Et dans le cas où l'on passe un e-mail qui invalide on s'assure qu'un message d'erreur est affiché :

The screenshot shows a Playwright test run titled "TestRecover.spec.js:55 > RecupIncorrecte". The timeline at the top shows the execution flow from 0 to 8.5 seconds. The left sidebar lists actions and metadata, including a "Before Hooks" section with several assertions and interactions with the "Forgot Password" page. The main area displays a screenshot of the "Forgot Password" page. A red box highlights the error message "User doesn't exist!" below the "Email Me a link" button.

## 8.17. Tests de sécurité :

Les tests de sécurité et de performances, sont le fruit d'un travail collaboratif qui se manifeste en une forte contribution de la part du responsable de sécurité Mohamed El Hadad.

### 8.17.1. Utilisation de l'sqlmap :

Tout d'abord on commence par installer « Sqlmap » sous linux, sans oublier « python » et « git » qui sont nécessaires pour l'exécution des commandes « sqlmap » :

```
taha@victus:~$ sudo apt install sqlmap
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
 python3-magic
Les NOUVEAUX paquets suivants seront installés :
 python3-magic sqlmap
```

```
taha@victus:~$ sudo apt install git
Lecture des listes de paquets... Fait
taha@victus:~$ git clone --depth 1 https://github.com/sqlmapproject/sqlmap.git sqlmap-dev
Clonage dans 'sqlmap-dev'...
remote: Enumerating objects: 730, done.
remote: Counting objects: 100% (730/730), done.
remote: Compressing objects: 100% (495/495), done.
```

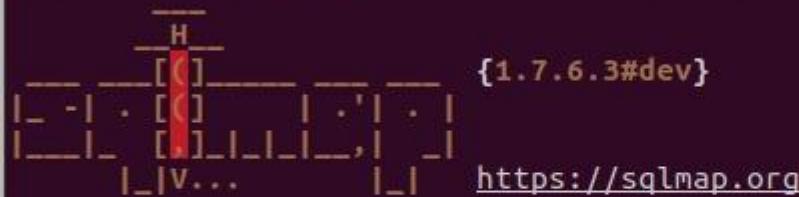
Cependant il en est nécessaire de cloner la dernière version de l'sqlmap disponible sur leur répertoire d'origine :

On vérifie que l'sqlmap est bien configuré :

On se procure tout d'abord d'une version du projet, préparée par les administrateurs linux, nous permettant de lancer le serveur d'une façon très simple, sachant qu'on est déjà sous linux et cela nous fait éviter pas mal de problèmes, notamment ceux des packages à installer et à re-configurer.

```
taha@victus:~/Bureau/XtraDh-release-0.2.0$ sudo docker compose up --build
[+] Building 0.7s (17/17) FINISHED
=> [app internal] load .dockerrcignore
=> => transferring context: 2B
=> [app internal] load build definition from Dockerfile
=> => transferring dockerfile: 479B
=> [app internal] load metadata for docker.io/library/nginx:alpine
=> [app internal] load metadata for docker.io/library/node:19
=> [app builder 1/6] FROM docker.io/library/node:19sha256:92f06fc13bcc09f1ddc51f6ebf1aa3d21a6532b74f076f224f188bc6b9317570
=> [app internal] load build context
=> => transferring context: 7.66kB
=> [app production-build 1/5] FROM docker.io/library/nginx:alpine@sha256:2e776a66a3556f001aba13431b26e448fe8acba277bf93d2ab1a785571a46d90
```

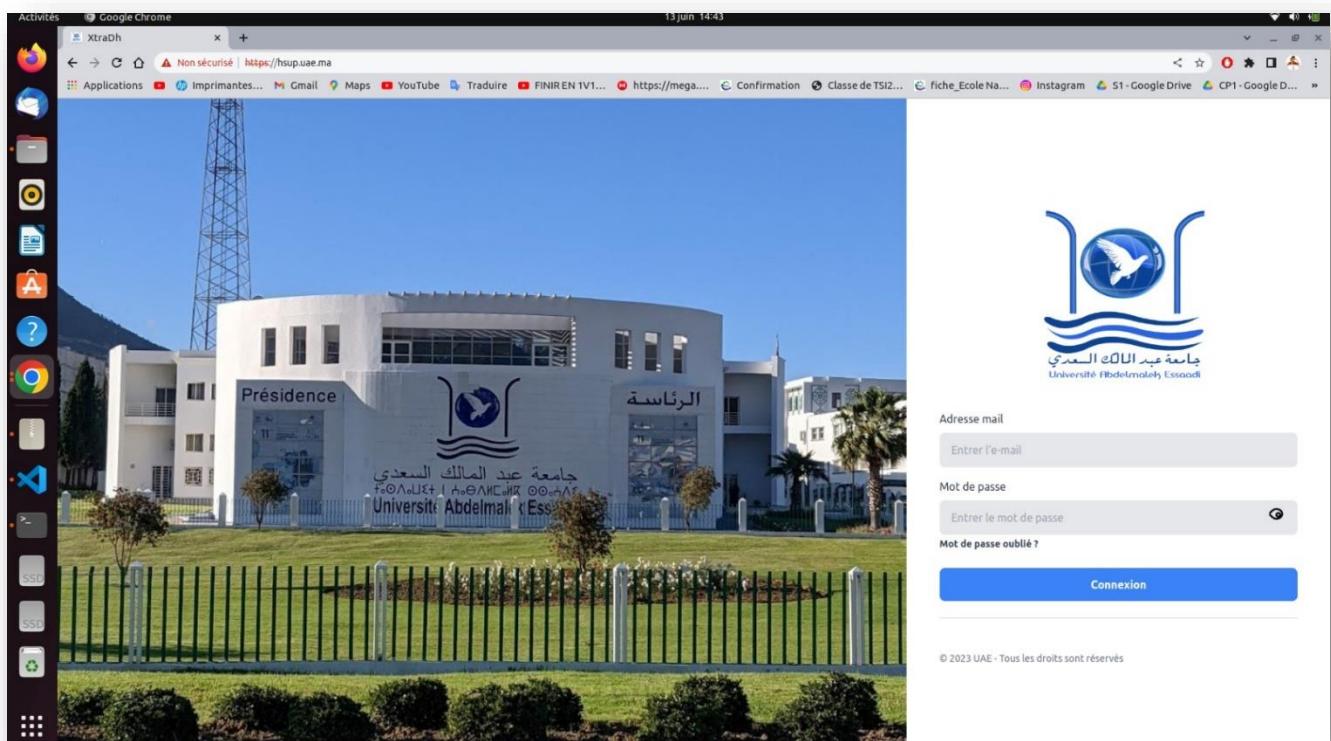
```
taha@victus:~$ cd sqlmap-dev
taha@victus:~/sqlmap-dev$ python3 sqlmap.py
```



Usage: python3 sqlmap.py [options]

sqlmap.py: error: missing a mandatory option (-d, -u, -l, -m, -r, -g  
rd, --shell, --update, --purge, --list-tampers or --dependencies). U

On vérifie avant de lancer la simulation des « SQL-injections » que le site s'affiche bien sur le



navigateur-web ainsi qu'il est accessible :

On exécute la commande suivante pour lancer la tentative d'injection sql :

```
● ● ●
1 python3 sqlmap.py -u 'https://hsup.uae.ma/' --data "email=EL
MOUMNI.Bouchta@uae.ac.ma&password=test" --random-agent

taha@victus:~/sqlmap-dev$ python3 sqlmap.py -u 'https://hsup.uae.ma/' --data "em
ail=ELMOUMNI.Bouchta@uae.ac.ma&password=test" --random-agent
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual
consent is illegal. It is the end user's responsibility to obey all applicable
local, state and federal laws. Developers assume no liability and are not responsible
for any misuse or damage caused by this program
[*] starting @ 14:44:13 /2023-06-13/
```

Il s'avère qu'à partir des résultats aperçus, le champ « email » de l'interface d'accueil du site est inpénétrable par les injections sql, il n'arrive même pas à détecter l'SGBD utilisé.

```
[14:44:14] [INFO] testing 'Boolean-based blind - Parameter replace (original val
ue)'
[14:44:14] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER B
Y or GROUP BY clause (EXTRACTVALUE)'
[14:44:14] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[14:44:14] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE o
r HAVING clause (IN)'
[14:44:14] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLT
ype)'
[14:44:14] [INFO] testing 'Generic inline queries'
[14:44:14] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[14:44:14] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)
'
[14:44:14] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - c
omment)'
[14:44:14] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[14:44:14] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
[14:44:14] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'
```

```
[14:44:13] [INFO] fetched random HTTP User-Agent header value 'Mozilla/5.0 (Windows; U; Windows NT 5.1; cs-CZ) AppleWebKit/525.28.3 (KHTML, like Gecko) Version/3.2.3 Safari/525.29' from file '/home/taha/sqlmap-dev/data/txt/user-agents.txt'
[14:44:13] [INFO] testing connection to the target URL
[14:44:13] [WARNING] the web server responded with an HTTP error code (405) which could interfere with the results of the tests
[14:44:13] [INFO] testing if the target URL content is stable
[14:44:13] [INFO] target URL content is stable
[14:44:13] [INFO] testing if POST parameter 'email' is dynamic
[14:44:13] [WARNING] POST parameter 'email' does not appear to be dynamic
[14:44:13] [WARNING] heuristic (basic) test shows that POST parameter 'email' might not be injectable
[14:44:13] [INFO] testing for SQL injection on POST parameter 'email'
[14:44:14] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
```

it is recommended to perform only basic UNION tests if there is not at least one other (potential) technique found. Do you want to reduce the number of requests ? [Y/n] y

```
[14:44:15] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[14:44:15] [WARNING] POST parameter 'email' does not seem to be injectable
[14:44:15] [INFO] testing if POST parameter 'password' is dynamic
[14:44:15] [WARNING] POST parameter 'password' does not appear to be dynamic
[14:44:15] [WARNING] heuristic (basic) test shows that POST parameter 'password' might not be injectable
[14:44:15] [INFO] testing for SQL injection on POST parameter 'password'
[14:44:15] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[14:44:15] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[14:44:15] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[14:44:15] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[14:44:15] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[14:44:15] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLTYPE)'
[14:44:15] [INFO] testing 'Generic inline queries'
[14:44:15] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[14:44:15] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
```

```
[14:44:15] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[14:44:15] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
[14:44:15] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'
[14:44:16] [INFO] testing 'Oracle AND time-based blind'
[14:44:16] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[14:44:16] [WARNING] POST parameter 'password' does not seem to be injectable
[14:44:16] [CRITICAL] all tested parameters do not appear to be injectable. Try to increase values for '--level'/'--risk' options if you wish to perform more tests. If you suspect that there is some kind of protection mechanism involved (e.g. WAF) maybe you could try to use option '--tamper' (e.g. '--tamper=space2comment')
[14:44:16] [WARNING] HTTP error codes detected during run:
405 (Method Not Allowed) - 146 times

[*] ending @ 14:44:16 /2023-06-13/
```

même chose pour le champ « password » de l'accueil du site, il s'avère qu'il est tout de même impénétrable.

Lors de l'exécution de cette commande on a intégré l'option `-random-agent`, qui nous permet d'utiliser des agents de requêtes aléatoires, ce qui augmente la probabilité de pénétration au niveau du site.

En général d'après les valeurs de retour, suite à l'exécution de la commande précédente, le site bloque toute tentative d'injection.

### 8.17.2. Utilisation d'OWASP ZAP :

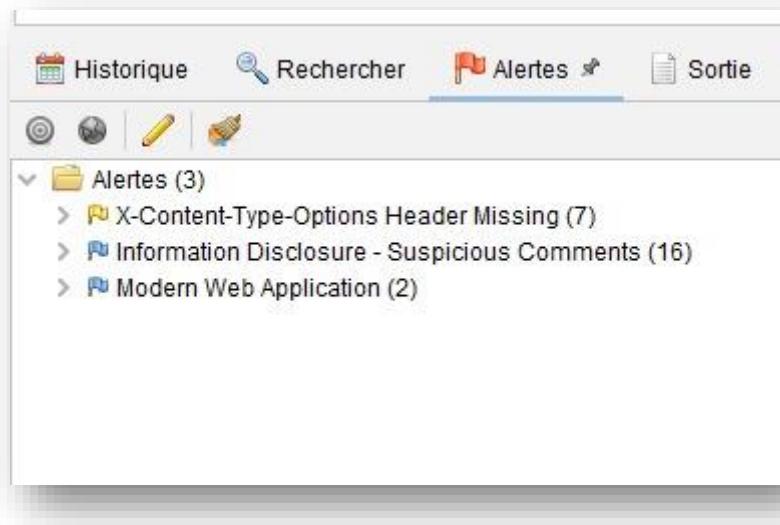
Owasp ZAP est un outil permettant de simuler les « injections sql », et les attaques « XSS » et d'autres failles/vulnérabilités disponibles au niveau du site.

Il suffit de lui fournir le point d'accès du site web afin qu'il lance une simulation d'attaques sur l'application-web concernée :

The screenshot shows the OWASP ZAP interface with the title bar "Session sans titre - OWASP ZAP 2.12.0". The main window displays an "Automated Scan" dialog box. Inside the dialog, the URL to attack is set to "http://192.168.1.177:8080". The "Attack" button is highlighted with a yellow arrow. Below the dialog, the main ZAP interface shows a list of network requests and responses. The requests table includes columns for Id, Timestamp de réq., Méthode, URL, Code, Raison, RTT, Taille de l'en-tête de rép., and Taille du corps de rép. The responses table shows a similar structure. At the bottom of the main interface, there are tabs for Historique, Rechercher, Alertes, Sortie, Robot d'indexation, AJAX Spider, WebSockets, and Scan adf!. The status bar at the bottom indicates "Scans en cours: 0 Num Requests: 440 New Alerts: 1 Exporter".

This screenshot shows the OWASP ZAP interface with the "Alertes" tab selected. The title bar still reads "Session sans titre - OWASP ZAP 2.12.0". The main window displays an "Automated Scan" dialog box, identical to the one in the previous screenshot. The "Alertes" tab is active, showing a list of detected vulnerabilities. A red arrow points from the "Attack" button in the previous screenshot to the "Alertes" tab here, indicating where to check for findings. The status bar at the bottom indicates "Scans en cours: 0 Num Requests: 440 New Alerts: 1 Exporter".

Les résultats du test de détection de vulnérabilité :



Aucune erreur d'un ordre grave n'a été détectée y compris ceux des injections sql et des attaques XSS, seulement des vulnérabilités relatives aux commentaires présents sur les fichiers des interfaces, ainsi que des erreurs liées au navigateurs d'anciennes générations (période pré 2014).

<ul style="list-style-type: none"> <li>▼ <b>Alertes (3)</b></li> <li>  &gt; <b>X-Content-Type-Options Header Missing (7)</b></li> <li>    <ul style="list-style-type: none"> <li><input type="checkbox"/> GET: http://192.168.1.177:8080</li> <li><input type="checkbox"/> GET: http://192.168.1.177:8080/</li> <li><input type="checkbox"/> GET: http://192.168.1.177:8080/favicon.ico</li> <li><input type="checkbox"/> GET: http://192.168.1.177:8080/img/AF1QipPd4KGT1xMp13</li> <li><input type="checkbox"/> GET: http://192.168.1.177:8080/img/logo.30871781.jpg</li> <li><input type="checkbox"/> GET: http://192.168.1.177:8080/js/app.js</li> <li><input type="checkbox"/> GET: http://192.168.1.177:8080/js/chunk-vendors.js</li> </ul> </li> <li>  &gt; <b>Information Disclosure - Suspicious Comments (16)</b></li> <li>  &gt; <b>Modern Web Application (2)</b></li> </ul>	<p>Attaquer:</p> <p>Preuve :</p> <p>Id CWE : 693</p> <p>Id WASC : 15</p> <p>Source: Passif (10021 - X-Content-Type-Options Header Missing)</p> <p>Input Vector:</p> <p>Déscription:</p> <p>The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'no-sniff' or 'nosniff'. This means that the browser may sniff the content type from the response body, which can lead to security issues if the content type does not match the declared type. Current (early 2014) and legacy browsers do not honor this header.</p> <p>Autre information:</p> <p>This issue still applies to error type pages (401, 403, 500, etc.) as those are also sniffed by the browser. At "High" threshold this scan rule will not alert on client or server error responses.</p> <p>Solution:</p> <p>Ensure that the application/web server sets the Content-Type header appropriately. If possible, ensure that the end user uses a standards-compliant and modern browser.</p>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Historique Rechercher Alertes Sortie Robot d'indexation AJAX Spider WebSockets Scan actif +

Confiance: Low  
Paramètre:  
Attaquer:  
Preuve : later  
Id CWE : 200  
Id WASC : 13  
Source: Passif (10027 - Information Disclosure - Suspicious Comments)  
Input Vector:  
Description:  
The response appears to contain suspicious comments which may help an attacker. Note: Matches made within script blocks.  
Autre information:  
The following pattern was used: \bLATER\b and was detected 2 times, the first in the element starting with: /\*\*\*\*\*\*/ suspicious comment/snippet.  
Solution:  
Remove all comments that return information that may help an attacker and fix any underlying problems they refer to.

Référence:

Alertes 0 1 2 Main Proxy: localhost:8080

Historique Rechercher Alertes Sortie Robot d'indexation AJAX Spider WebSockets Scan actif +

Confiance: Low  
Paramètre:  
Attaquer:  
Preuve : later  
Id CWE : 200  
Id WASC : 13  
Source: Passif (10027 - Information Disclosure - Suspicious Comments)  
Input Vector:  
Description:  
The response appears to contain suspicious comments which may help an attacker. Note: Matches made within script blocks.  
Autre information:  
The following pattern was used: \bLATER\b and was detected 2 times, the first in the element starting with: /\*\*\*\*\*\*/ suspicious comment/snippet.  
Solution:  
Remove all comments that return information that may help an attacker and fix any underlying problems they refer to.

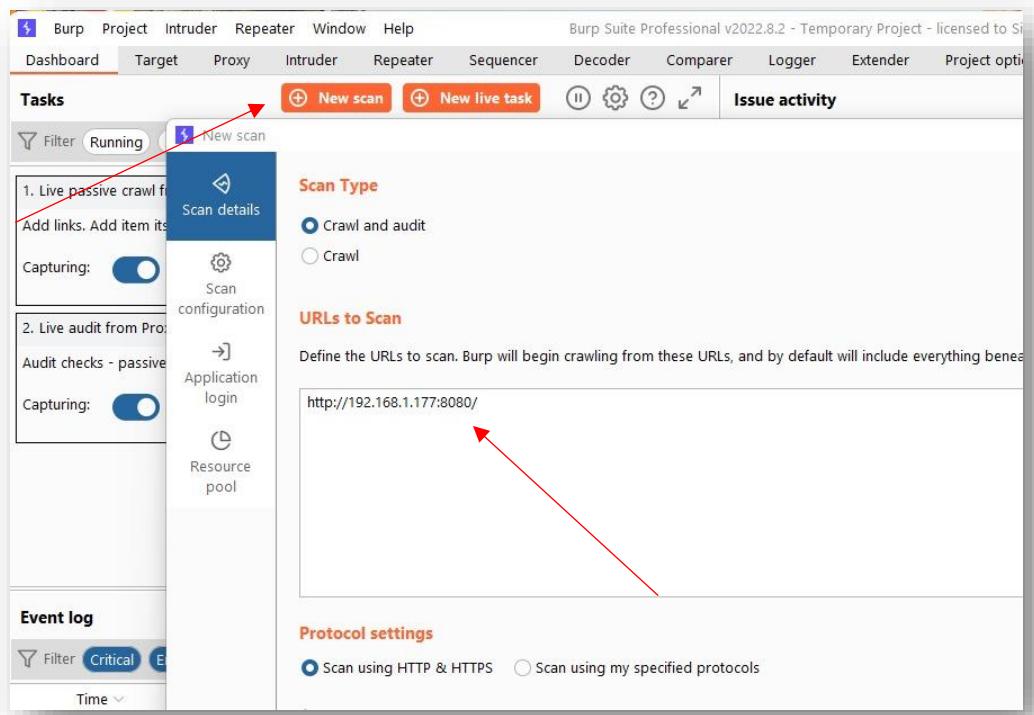
Référence:

Alertes 0 1 2 Main Proxy: localhost:8080 Session sans titre - OWASP ZAP 2.12.0

### 8.17.3. Utilisation de Burpsuite :

Finalement et pour le dernier outil utilisé pour s'assurer de la puissance des politiques de sécurité mises en place au niveau du projet, on se retrouve cette fois-ci avec Burpsuite, qui 'est un outil développé en java et qui nécessite une version java STL pour son fonctionnement, il permet comme les deux autres outils proposés juste avant, de repérer les vulnérabilités de l'application web y compris les injections sql et les attaques XSS.

On lui renseigne le l'URL du point d'accès du site web, puis on opte pour une analyse approfondie :



On ajuste les paramètres selon nos préférences, puis on exécute l'utilitaire de détection des erreurs :

Task	Description	Status	Issues
2. Live audit from Proxy (all traffic)	Audit checks - passive Capturing: On	0 requests (0 errors)	0 0 0 0
3. Crawl and audit of 192.168.1.177:8080	Crawl and Audit - Deep Finished.	1587 requests (4 errors) 6 locations crawled	1 2

Après l'exécution du test il s'avère que seulement des erreurs mineures concernant encore une fois les commentaires présents au niveau du code des interfaces ainsi que quelques certificats facultatifs.

## 8.18. Tests des performances (apache Jmeter):

Normalement laravel, limite le nombre de requêtes à 60 par minute, et donc une fois ce nombre de requête est dépassé, un message d'erreur 429 : « Too many requests » est généré

On Utilise dans cette partie l'outil « apache jmeter », qui permet de faire des tests de « load » ou de « performances » :

```
XtraDh > app > Http > Kernel.php
18 // \App\Http\Middleware\TrustHosts::class,
19 \App\Http\Middleware\TrustProxies::class,
20 \Fruitcake\Cors\HandleCors::class,
21 \App\Http\Middleware\PreventRequestsDuringMaintenance::class,
22 \Illuminate\Foundation\Http\Middleware\ValidatePostSize::class,
23 \App\Http\Middleware\TrimStrings::class,
24 \Illuminate\Foundation\Http\Middleware\ConvertEmptyStringsToNull::class,
25];
26
27 /**
28 * The application's route middleware groups.
29 *
30 * @var array<string, array<int, class-string|string>>
31 */
32 protected $middlewareGroups = [
33 'web' => [
34 \App\Http\Middleware\EncryptCookies::class,
35 \Illuminate\Cookie\Middleware\AddQueuedCookiesToResponse::class,
36 \Illuminate\Session\Middleware\StartSession::class,
37 // \Illuminate\Session\Middleware\AuthenticateSession::class,
38 \Illuminate\View\Middleware\ShareErrorsFromSession::class,
39 \App\Http\Middleware\VerifyCsrfToken::class,
40 \Illuminate\Routing\Middleware\SubstituteBindings::class,
41],
42
43 'api' => [
44 \Laravel\Sanctum\Http\Middleware\EnsureFrontendRequestsAreStateful::class,
45 'throttle:api',
46 \Illuminate\Routing\Middleware\SubstituteBindings::class,
47 // \Spatie\Csp\AddCspHeaders::class,
48],
49];
50];
```

http://apache-jmeter-3.0.0m1/view/results-tree.jmx - Apache JMeter (3.0)

Name: View Results Tree  
Comments:  
Write results to file / Read from file  
Filename:   
Search:  Case sensitive Regular exp. Search Reset

Sampler result Request Response data

Thread Name: Thread Group 1-61  
Sample Start: 2023-06-13 18:33:23 WEST  
Load time: 41376  
Connect Time: 1  
Latency: 41374  
Size in bytes: 7002  
Sent bytes: 255  
Headers size in bytes: 377  
Body size in bytes: 6625  
Sample Count: 1  
Error Count: 1  
Data type ("text"|"bin"|""): text  
Response code: 429  
Response message: Too Many Requests

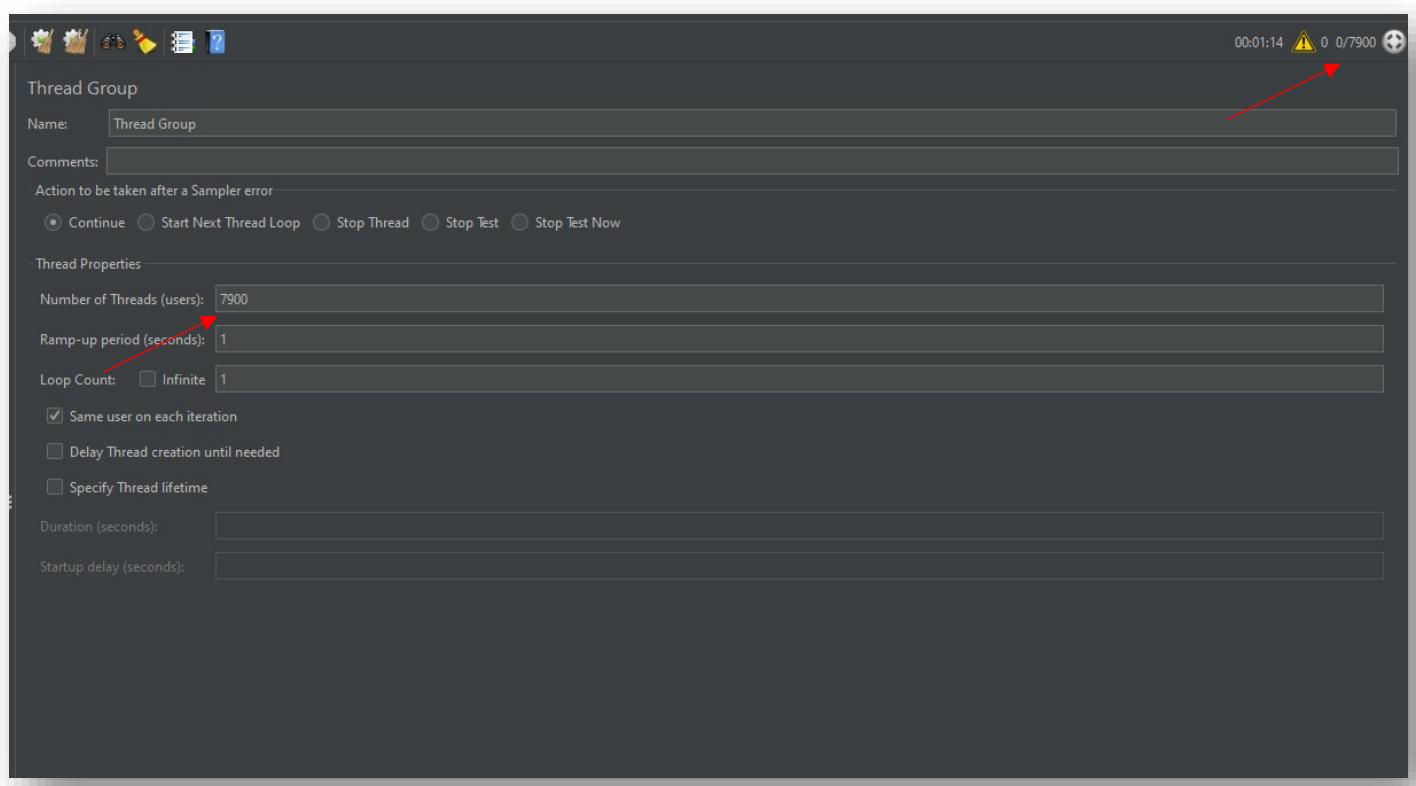
HTTPSampleResult fields:  
Content-Type: text/html; charset=UTF-8  
DataEncoding: UTF-8

Connexion  
Connexion  
**Connexion**  
Connexion  
Connexion  
Connexion  
Connexion

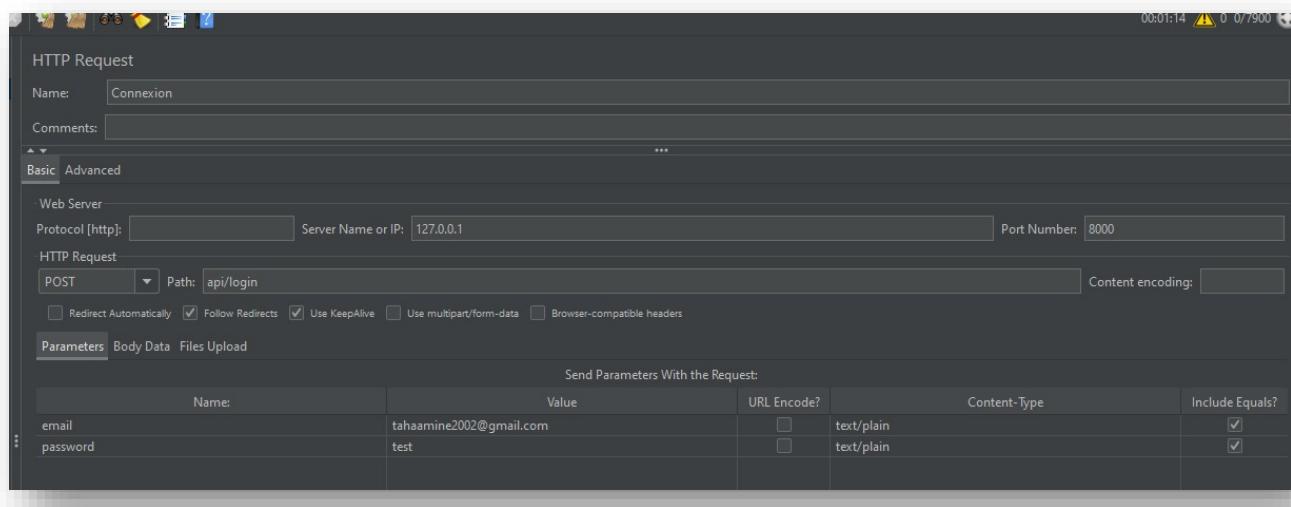
Sample Count: 1  
Error Count: 1  
Data type ("text"|"bin"|""): text  
Response code: 429  
Response message: Too Many Requests

En changeant la valeur du « Throttle » en 8000, le test peut recevoir jusqu'à 8000 requêtes sans erreur !

```
'api' => [
 \Laravel\Sanctum\Http\Middleware\EnsureFrontendRequestsAreStateful::class,
 'throttle:8000,1',
 \Illuminate\Routing\Middleware\SubstituteBindings::class,
 // \Spatie\Csp\AddCspHeaders::class,
],
;
```

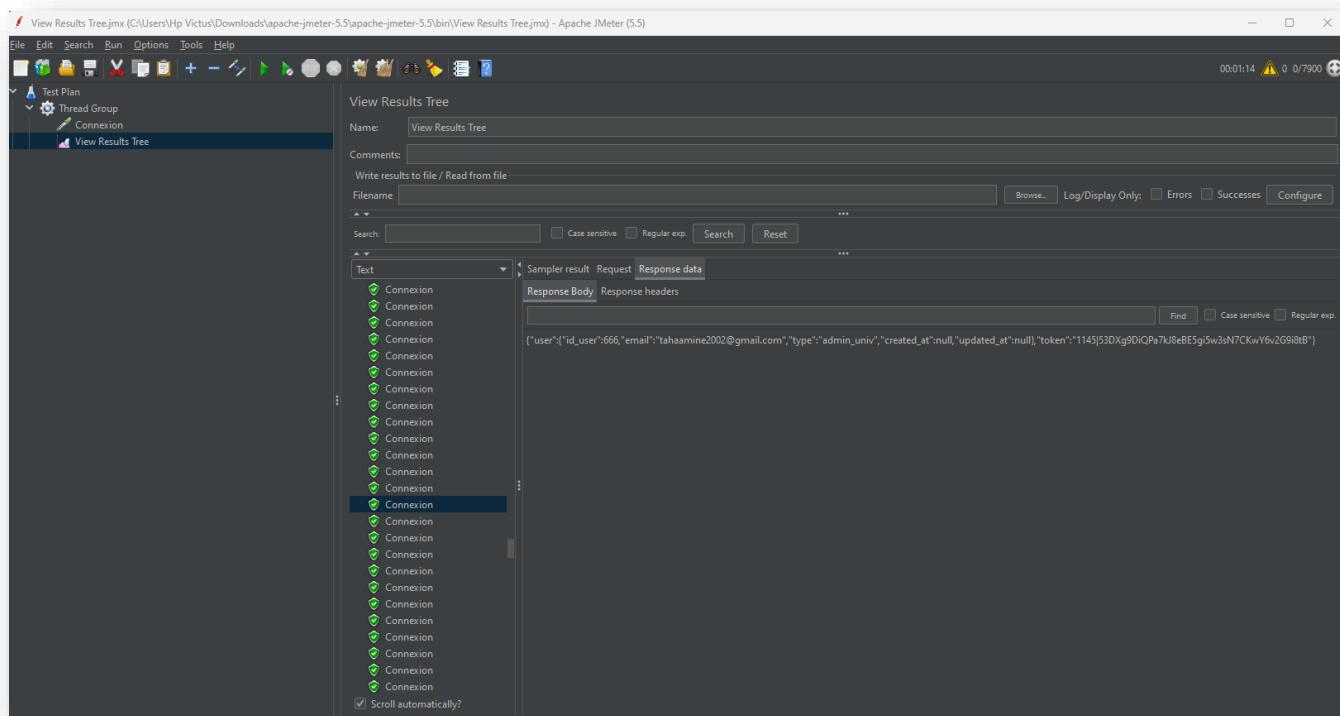


On configure Apache Jmeter pour qu'il puisse simuler un « load » de 7900 personnes environ, d'une façon à peu près simultanée



On configure les données d'accès pour simuler les performances en connexion en utilisant le chemin api/login.

Les requêtes ont été envoyés avec succès et bien plus que ça



## Conclusion

Dans cette conclusion, nous souhaitons récapituler les résultats obtenus et les objectifs que nous avons atteints tout au long de ce projet. Grâce à nos efforts et à notre collaboration, nous avons pu concevoir et développer une application web robuste et fonctionnelle pour automatiser le paiement des vacances et des heures supplémentaires des enseignants.

Ce projet n'a pas été sans défis. Nous avons été confrontés à des problèmes techniques et conceptuels, mais nous les avons surmontés grâce à notre détermination et à notre esprit d'équipe. Ces obstacles ont été des opportunités pour nous d'apprendre et de grandir, en nous permettant d'acquérir de nouvelles compétences techniques et de renforcer notre capacité à résoudre les problèmes de manière créative et efficace.

Cette expérience nous a permis d'acquérir de précieuses compétences techniques, telles que la maîtrise de PostgreSQL, Laravel, Vue.js et la gestion d'une architecture Docker. Sur le plan comportemental, nous avons développé notre capacité à travailler en équipe, à gérer les délais et à faire face aux défis rencontrés. Nous avons également retenu l'importance d'une planification rigoureuse, d'une communication claire et d'une gestion efficace des ressources. En plus des connaissances techniques que nous avons développées tout au long de ce projet, nous avons également eu l'opportunité d'échanger avec un 5<sup>ème</sup> année de l'Ecole Nationale des Sciences Appliquées. Nous avons eu la chance de bénéficier de ses conseils et de son partage d'expérience qui nous a permis de comprendre l'application concrète de nombreux outils largement utilisés dans les entreprises, tels que Jira, Gitlab, Bitbucket et Node.js.

Nous avons également été exposés à certaines pratiques moins répandues au Maroc, telles que l'utilisation de microservices dans les projets ou les équipes de développement, ainsi que les concepts de CI/CD (Continuous Integration/Continuous Deployment) pipelines. Ces nouvelles connaissances nous ont ouvert les yeux sur les avancées et les meilleures pratiques du monde professionnel, renforçant ainsi notre compréhension de l'industrie et élargissant notre horizon.

Grâce à ce projet, nous avons pu constater la pertinence et l'importance de rester constamment à jour avec les évolutions technologiques et les tendances du marché. Cela nous a motivés à poursuivre notre apprentissage et à continuer d'explorer de nouvelles technologies et de nouvelles méthodologies afin de rester compétitifs sur le marché du travail. En conclusion, ce projet nous a non seulement permis d'appliquer nos connaissances théoriques, mais il nous a également offert une opportunité précieuse de découvrir et d'incorporer des pratiques et des connaissances essentielles pour notre future carrière professionnelle.