

SOMMAIRE

LISTE DES FIGURES	3
INTRODUCTION	4
CONCEPTION	5
REALISATION	7
CONCLUSION	12
WEBOGRAPHIE	13

LISTE DES FIGURES

FIGURE 1 : MCD	5
FIGURE 2 : MLD	6
FIGURE 3 : Les actions / Formulaire	7
FIGURE 4 : Les pages / Objectifs	8
FIGURE 5 : Les fonctions PHP / Objectifs	9
FIGURE 6 : Les fonctions Javascript / Objectifs	10
FIGURE 7 : Dossier config	11
FIGURE 8 : Exemple message automatique	11

INTRODUCTION

Le but général du ce projet est la création d'un espace où les étudiants, enseignants et fonctionnaires de l'Ecole Supérieur du Technologie d'Oujda peuvent avoir les informations du contact des autres d'une façon qui est simple et très efficace.

Tout ça en respectant la confidentialité des utilisateurs, puisque les informations ne sont pas données tous au n'importe qui.

Avec un coté administration qui possède toutes les accessibilités, pour bien gérer les procédures du site web.

CONCEPTION

- **Modèle Conceptuel du Données « MCD » :**

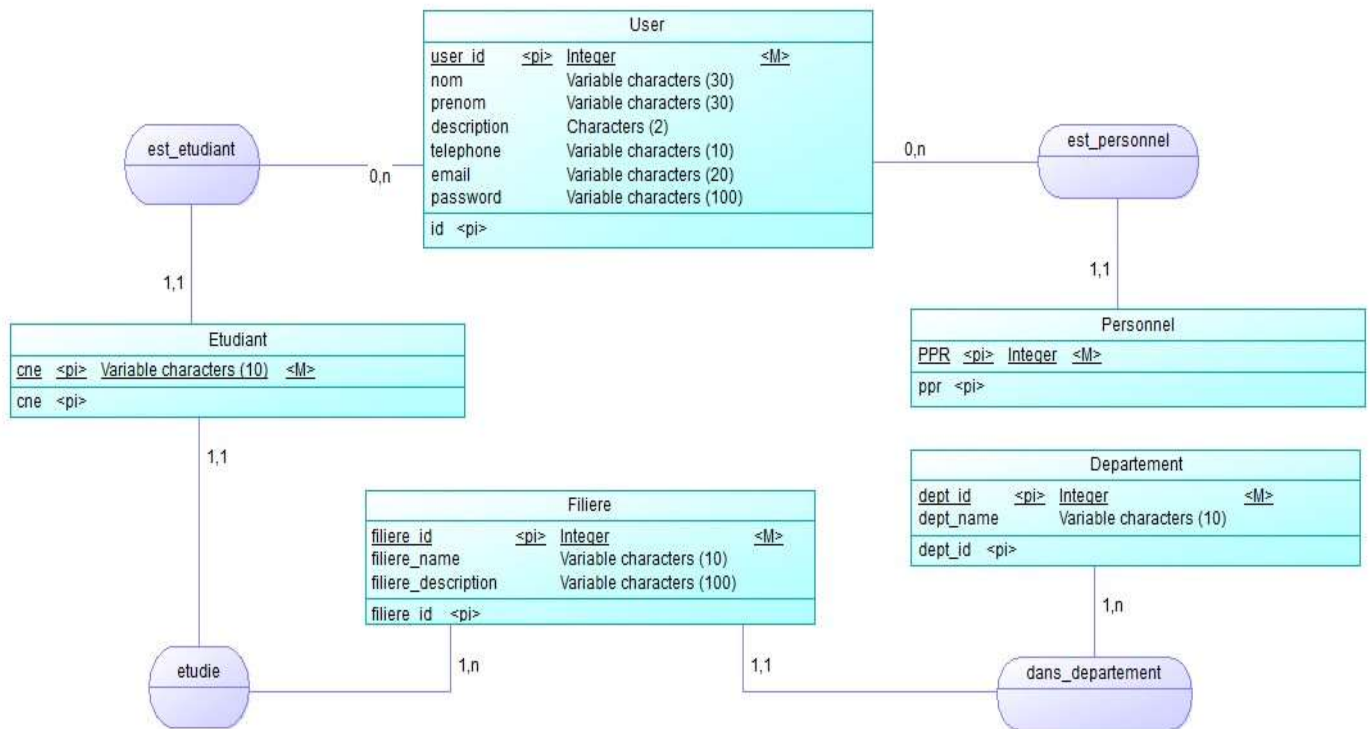


Figure 1 : Modèle Conceptuel du Données « MCD »

Ce modèle Conceptuel réalisé est composé de 5 entités qui ce sont : User, Etudiant, Personnel, Filière, Département.

Les entités Etudiant et Personnel sont ajouté pour bien distingués les informations qui sont juste pour les étudiants et les autres informations qui sont des enseignants et fonctionnaires.

On a 4 associations dans ce modèle, on a déjà cité « est_etudiant » et « est_personnel » sont juste pour distingués entre la fonction de l'utilisateur. La troisième association « etudie » est pour avoir la filière où cet étudiant etudie. Et dernièrement l'association « dans_departement » est pour avoir dans quel département appartient une filière.

- **Modèle Logique du Données « MLD » :**

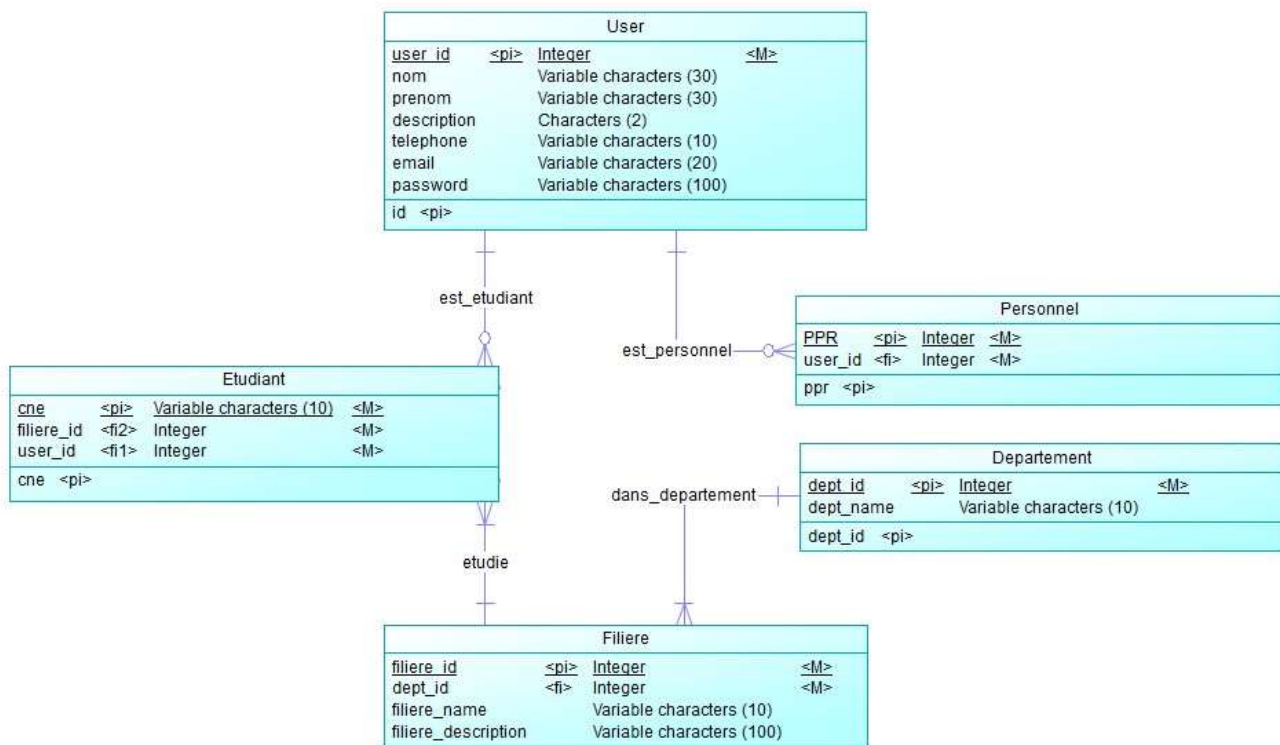


Figure 2 : Modèle Logique du Données « MLD »

Le modèle logique de données généré nous décrit exactement de quoi notre base de données est composée.

On remarque la migration de quelques clés étrangères, par exemple, l'identifiant d'utilisateur de la table « User » vers les tables « Etudiant » et « Personnel », et aussi l'identifiant de la filière de la table « Filiere » vers la table « Etudiant », et finalement l'identifiant du département de la table « Departement » vers la table « Filiere ».

REALISATION

- Les actions « Dossier : Actions » / Formulaires :

Fichier de l'action	Objectif du formulaire	Page du formulaire
actionAdmin.php	Suppression d'un administrateur	Admin/Admins.php
actionUser.php	Unvérifier / Supprimer un utilisateur	Admin/Users.php
addAdmin.php	Ajouter un administrateur	Admin/Admins.php
addDept.php	Ajouter un département	Scolarite/School.php
addFiliere.php	Ajouter une filière	Scolarite/School.php
modifyInfo.php	Modification des informations personnelles	Home/Account.php
modifyPassword.php	Modification du mot de passe personnel	Home/Account.php
signin.php	Authentification d'utilisateur	Index.php
signup.php	Inscription d'un nouvel utilisateur	Index.php
updateDept.php	Modification du nom d'un département	Scolarite/School.php
updateFiliere.php	Modification des informations d'une filière	Scolarite/School.php
verifyUser.php	Vérification d'un utilisateur inscrit	Admin/verify-users.php

Figure 3 : Les actions / Formulaires

● **Les pages :**

Fichier de la page	Objectif de la page
Accessible pour tout le monde	
Index.php	La seule page qui est visible pour tout le monde, elle permet aux visiteurs l'authentification et aussi l'inscription
Accessible pour tous les utilisateurs authentifiés	
Home/Home.php	L'utilisateur est dirigé automatiquement vers cette page, elle permet aux utilisateurs la recherche des informations des autres selon sa fonction
Home/Account.php	La page des paramètres d'utilisateur, elle permet aux utilisateurs la modification des données personnelles ainsi que le mot de passe
Accessible pour les enseignants et les administrateurs seulement	
Scolarite/School.php	Cette page permet aux enseignants et administrateurs de visualiser tous les départements et les filières quelle appartient de même que la liste des étudiants. Ainsi elle permet aux administrateurs de modifier les informations des départements et aussi les filières quelle appartient.
Accessible pour les administrateurs seulement	
Admin/Admins.php	Cette page permet aux administrateurs de voir la liste des autres admins ainsi que l'ajout d'un nouveau
Admin/Verify-users.php	Cette page permet aux administrateurs de vérifier (accepter) les nouveaux utilisateurs inscrit, de même que le rejet de ses derniers.
Admin/Users.php	Cette page permet aux administrateurs la gestion des utilisateurs déjà acceptés dans l'annuaire.

Figure 4 : Les pages / Objectifs

● **Les fonctions PHP « Fichier : Config/Functions.php » :**

Nom du fonction	Objectif de la fonction
getConnection	Retourne une connexion PDO pour optimiser le code
addUser	Fonction qui permet l'ajout d'un nouvel utilisateur
getUserBy	La récupération d'un utilisateur par une valeur d'une colonne, par défaut c'est l'identifiant
getNotVerifiedUsers	La récupération des utilisateurs qui ne sont pas encore vérifiés
getVerifiedUsers	La récupération des utilisateurs qui sont déjà vérifiés
verifyUser	La vérification d'un utilisateur
unverifyUser	L'invérification d'un utilisateur déjà vérifié
rejectUser	Le rejet d'un utilisateur
updateUser	La modification des informations personnelles
updatePassword	La modification du mot de passe personnel
getStudentByCne	La récupération d'un étudiant par son code CNE
getStudentByEmail	La récupération d'un étudiant par son email
getAllStudents	La récupération de tous les étudiants
getStudentsInFiliere	La récupération de tous les étudiants d'une filière précis
getPersonnelByPPR	La récupération d'un personnel par son numéro PPR
getPersonnelByEmail	La récupération d'un personnel par son email
getAllAdmins	La récupération de tous les administrateurs
addAdmin	L'ajout d'un nouvel administrateur
deleteAdmin	La suppression d'un administrateur
getAllFiliere	La récupération de tous les filières

getFiliereById	La récupération d'une filière par son identifiant
getFiliereInDept	La récupération de toutes les filières qui appartient à un département
addFiliere	L'ajout d'une nouvelle filière
updateFiliere	La modification des informations d'une filière
getAllDepartements	La récupération de tous les départements
addDepartement	L'ajout d'un nouveau département
updateDepartement	La modification des informations d'un département

Figure 5 : Les fonctions PHP / Objectifs

● Les fonctions Javascript « Dossier : JS » :

Nom du fonction	Fichier	Objectif de la fonction
showSign	Main.js	L'affichage des formulaires de connexion / d'inscription
changeSignInfo	Main.js	L'affichage des champs du formulaire d'inscription selon le choix du fonction (Etudiant / Enseignant / Fonctionnaire)
verifyForm	Main.js	Vérification de la validité des données des formulaires d'authentification / d'inscription coté client avant la soumission au serveur
hideMessage	Main.js	Suppression des messages de succès / d'erreur après le click d'utilisateur
changeModifyForm	Main.js	Basculer entre le formulaire de modification des données personnelles et le formulaire de modification du mot de passe selon le choix
showModifyForm	Main.js	Basculer entre les boutons de changements des formulaires de modification des données / mot de passe
verifyModifyForm	Main.js	Vérification des données des formulaires du modification coté client avant la soumission au serveur
showUsers	Admin.js	L'affichage des utilisateurs selon la fonction à partir du champs « select »

deleteMessage	Admin.js	La suppression des messages d'erreurs / de succès après le click de l'utilisateur
showModal	Admin.js	L'affichage de la fenêtre flottante après le click sur le bouton approprié en utilisant les champs « data »
loadDeptContent	Admin.js	Le chargement des informations d'un département précis dans une fenêtre flottante afin de les modifier
loadFiliereContent	Admin.js	Le chargement des informations d'une filière précis dans une fenêtre flottante afin de les modifier
searchUsers	Search.js	La recherche des utilisateurs selon le choix de recherche (nom, prénom, email, téléphone) et retourne tous les résultats qui contient cette clé de recherche

Figure 6 : Les fonctions Javascript / Objectifs

● Les messages d'erreurs / succès :

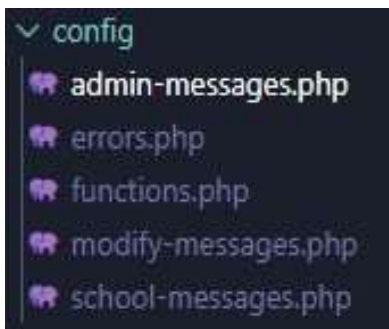


Figure 7 : Dossier Config

```

if(isset($_GET[sha1('phone-used')])) {
    $error = 'Telephone déjà utilisé par un autre utilisateur!';
    $newAdminError = 1;
}

if(isset($_GET[sha1('admin-deleted')])) {
    $succes = 'Administrateur supprimé avec succès!';
}

```

Figure 8 : Exemple des messages automatique

L'exemple au-dessus nous montre comment cette gestion des messages est effectué, donc pour chaque cas du message il correspond une clé qui hashé avec la fonction **sha1** pour des raisons de sécurité, et celui-ci est passée dans l'URL de la page.

CONCLUSION

Ce projet annuaireESTO ma permet de regrouper et développer plusieurs compétences dans le domaine du développement Web et ça en essayant de toucher tous les besoins fonctionnels possible dans le site web et aussi en respectant aussi les aspects de sécurité en termes d'accessibilité que ce soit entre les utilisateurs eux-mêmes et aussi entre les utilisateurs et le site web.

WEBOGRAPHIE

- <https://www.w3schools.com/>
- <https://www.flaticon.com/>
- <https://stackoverflow.com/>