



Master

Master 2 Biostatistiques

Promotion 2023-2024

Rapport technique site internet

ORUE Adrien

PASQUET Léa

PERPERE Maud

RUIZ Anne-Andrée

Février 2024

UE INF202

Étape 1 : Importation des données

La première étape a été de créer une base de données dans PhpMyAdmin et d'y importer les différentes tables (vides) à l'aide du fichier SQL fourni avec l'énoncé.

Ensuite, nous avons utilisé le fichier Accdb pour remplir les tables vides. Pour se faire, nous avons converti les fichiers Accdb en csv afin d'obtenir toutes les tables au format csv.

Puis, pour chaque table, nous avons converti toutes les dates dans un format accepté par PhpMyAdmin, nous avons encodé toutes les tables en UTF8. Nous avons aussi corrigé les petites erreurs commises lors de conversion en csv (espace, guillemets...).

Ensuite, nous avons souhaité importer les données de chaque table dans PhpMyAdmin qui ne contenait pour le moment, uniquement des tables vides. Nous avons eu beaucoup de mal à importer les données pour des raisons que nous ignorons encore. En effet, les données ne s'importent pas complètement, des lignes étaient ignorées alors que tout était correct dans les fichiers csv.

Nous avons finalement trouvé une solution : nous exportons les tables vides de PhpMyAdmin en csv, puis nous copions le contenu des tables csv (converties du fichiers Accdb) dans le fichier csv exporté. Enfin, nous importons les données dans PhpMyAdmin, à partir du nouveau fichier csv créé.

De plus, si quelques erreurs persistent, nous veillons à bien vérifier la structure des fichiers csv en les éditant dans NotePad ++, ce qui permet de vérifier les erreurs possibles d'importation et d'être vigilants sur les éléments qui pourraient être mal importés.

De cette manière, nous avons pu importer les données table par table sans erreur.

Enfin, nous avons conservé la structure présentée dans le sujet qui nous paraissait être la plus cohérente. Nous avons défini et vérifié les clés primaires et étrangères de la base de données.

Nous avons créé les liens entre les tables pour obtenir le schéma relationnel choisi.

Nous avons eu quelques difficultés à cette étape : en effet, nous n'avons pas réussi à créer tous les liens entre les tables. Certaines données manquantes ont été transformées en 0 à l'importation, ce qui entraîne des impossibilités de créer des liens car le 0 apparaît comme une modalité dans une table et est inconnue dans la table à lier. Au total, ce problème nous a empêché de créer les 13 relations suivantes :

- tab_suivi / tab_codage_suivi
- tab_suivi / tab_codage_adaptation
- tab_suivi / tab_patient
- tab_patient / tab_codage_csp

- tab_patient / tab_codage_ethnie
- tab_patient / tab_codage_latéralité
- tab_chirurgie / tab_patient
- tab_atcd_perso / tab_patient
- tab_contacto / tab_codage_nom_lentilles
- tab_chirurgie / tab_codage_chirurgie
- tab_atcd_famil / tab_codage_atcd_famil
- tab_atcd_famil / tab_codage_parente
- tab_atcd_perso / tab_codage_atcd

On obtient donc un schéma avec 7 relations (voir Annexe 1). Nous avons conscience de la difficulté rencontrée, mais cela ne nous empêchera pas de continuer le projet. Nous préférons donc passer du temps sur la compréhension et la structure du site.

Pour finir, nous avons exporté notre base de données au format SQL.

Le fichier SQL obtenu, permet d'obtenir directement et pour n'importe quel utilisateur, la base de données remplie lorsque nous exécutons dans PhpMyAdmin par exemple.

Afin de travailler ensemble mais à distance, nous avons choisi de travailler avec Github.

Etape 2 : Création de l'interface PHPMYSQL

Changement dans la base de données : Pour cette deuxième étape, nous avons rajouté une table dans la base de données relatives au personnel du service. En effet, nous avons rentré une table avec 4 professionnels de santé, qui permettra par la suite de se connecter sur le site pour avoir accès à des données des patients.

1) Structure du site et fonctions d'usage du site :

La page d'accueil permet de présenter le service à la fois pour le personnel soignant, et tout public. Cette page présente le service en général puis sa spécialisation sur la pathologie du kératocône. Enfin, elle présente les différents engagements du service.

Ensuite, les professionnels de santé peuvent se connecter avec id et mot de passe afin de réaliser différentes actions sur les données des patients qui sont confidentielles. Cette partie permet par exemple au personnel de pouvoir ajouter ou supprimer un patient de la base de données lorsqu'il intègre ou quitte le service.

Concernant les patients, et le public en général, ils n'ont pas accès aux données concernant les patients mais ont accès à la page d'accueil et aussi à différentes données du service. Nous avons différencié une partie tout public, avec les données de service. Ces données apportent des informations sur les types de chirurgie réalisée dans le service ainsi que le personnel soignant qui travaille dans le service.

Concernant la structure du site, sur chaque page se trouve une barre de navigation et un footer. Ces éléments ont été créés par le groupe une seule fois comme un modèle, que nous incluons dans chaque page individuelle ou collective par la fonction "include".

Le footer contient diverses fonctionnalités renvoyant à des pages qui seront décrites dans section "pages communes".

2) Pages individuelles :

a) Page de connexion à l'espace personnel + page suivante d'accès aux données : Léa

Après avoir créé une table relative au personnel soignant du service, j'ai réalisé un formulaire avec deux champs de type texte pour entrer son nom d'utilisateur et son mot de passe, puis un bouton pour accéder à la page suivante. Ce formulaire (formulaire_co_equipe) est lié à un fichier php (connexion_equipe) qui est construit à partir de plusieurs éléments :

- Partie connexion à la base de donnée
- Partie requête : cette partie est constituée d'une requête SQL avec plusieurs paramètres.
- Partie résultat de la requête : cette partie donne le résultat de la requête, c'est ce que nous voyons à l'écran. Cette partie est différente selon ce que l'utilisateur écrit dans le formulaire. Le professionnel de santé peut accéder aux données des patients uniquement si le nom d'utilisateur et le mot de passe sont corrects. Sinon un message d'erreur est renvoyé (soit mot de passe incorrect ou utilisateur incorrect), puis un bouton nouvel essai, permet de retenter sa chance et de revenir au formulaire.

Enfin, une fois que le professionnel est parvenu à se connecter, il clique sur le lien "accéder aux données patients" qui mènent à une nouvelle page (donnees_patients_accueil). Cette page est centrale, nous y avons regroupé toutes les pages de chacun qui renvoient à des fonctionnalités différentes pour les professionnels de santé. J'ai réalisé une liste déroulante pour que l'utilisateur puisse choisir sa fonctionnalité.

Pour ma part, j'ai réalisé les pages permettant au personnel de supprimer un patient de la base de données informatique.

Pour rendre les formulaires plus jolis et respecter la charte graphique, j'ai utilisé des balises style, qui m'ont permis d'encadrer les formulaires, mettre des couleurs, modifier les

caractères, centrer le formulaire. Cette balise permet aussi d'intégrer du langage CSS. J'ai ajouté des commentaires dans les formulaires html pour comprendre à quoi correspondent les lignes de la balise style.

Limites :

- J'aurai pu ajouter le hachage du mot de passe qui apporte une sécurité supplémentaire.
- J'aurai pu ajouter la possibilité d'oublier son mot de passe avec une redirection pour le mettre à jour, pour se faire j'aurai utiliser une requête UPDATE.

b) Page "supprimer un patient de la base informatique" : Léa

Afin de supprimer un patient de la base, j'ai créé un formulaire (delete_patients_form) avec le nom, le prénom et le numéro de dossier du patient à supprimer. J'ai ajouté un bouton pour consulter le résultat. Ce formulaire est lié à un fichier php qui permet la connexion avec la base de données et la réalisation de requête sql (DELETE) qui permet de supprimer un patient de la base de données (delete_patient).

Ensuite, j'affiche un message pour confirmer que la suppression a bien été réalisée. Si ce n'est pas le cas, j'ai paramétré l'affichage d'un message d'erreur indiquant une mauvaise identification d'un patient avec un bouton pour revenir à la page d'identification du patient à supprimer.

Une fois de plus, j'ai essayé de rendre le formulaire joli pour l'utilisateur en l'encadrant et en coloriant le bouton d'accès, avec des balises styles.

Limite :

- J'aurai pu ajouter une étape intermédiaire de confirmation avec une question du type "êtes-vous sûr de vouloir supprimer ce patient", car il s'agit d'une action ayant une importance majeure.
- J'aurai pu ajouter des messages d'erreur différents selon si il s'agit d'une erreur de mot de passe, d'utilisateur ou de numéro de dossier.

c) Page qui ajoute un patient dans la base de données lorsque le médecin se connecte (formulaire d'ajout + requêtes) : Anne-Andrée

Cette page permet d'ajouter un patient à la base de données. Pour la trouver, il faut que le médecin aille se connecter puis qu'il choisisse "Ajouter un patient au service". La page suivante est constituée d'un formulaire ("ajout_pat_form.php") qui renseigne des informations sur le patient à ajouter comme son numéro de dossier, son nom et son prénom.

La page "ajout_pat_action.php" ajoute le patient à la base de données. Le numéro de dossier doit être unique pour chaque patient. Une limite de ce formulaire est qu'il ne vérifie pas que ce numéro n'est pas déjà présent dans la base. Une autre limite est qu'il n'y a pas de vérification pour voir si le patient n'est pas déjà présent dans la base également avant de le

rajouter. Pour cela, j'aurais pu créer un formulaire pour regarder si le patient était déjà dans la base ou non. Pour ajouter le patient, le médecin valide toutes ses informations puis il lui est proposé de revenir à la page une fois l'ajout terminé. Le lien "retour" permet de choisir une autre fonctionnalité une fois que le patient a bien été ajouté à la base.

d) Page qui donne le nombre de patients par type de chirurgie dans les données de service (formulaire de sélection de la chirurgie + affichage du nombre) : Anne-Andrée

La page qui donne le nombre de patients par type de chirurgie pratiqué est accessible à partir de l'onglet "Données de service" puis en sélectionnant le "nombre de patients par type de chirurgie" dans la liste déroulante. La personne sélectionne alors le type de chirurgie souhaité dans le formulaire ("sel_pat_chir.php") puis le nombre de patients s'affiche ("spc_action.php"). Un lien "retour" permet de revenir au choix d'option de la page qui renseigne les données du service.

Pour accéder au nombre de patients par type de chirurgie, j'ai lié les tables tab_codage_chirurgie, tab_chirurgie et tab_patient de la base de données. Pour relier la table tab_chirurgie à tab_patient, j'ai renommé le nom de la variable du numéro de dossier "N° dossier" en "num_dossier" dans la table tab_chirurgie. L'espace entre "N°" et "dossier" empêchait la liaison de se faire. Ceci m'a permis de récupérer tous les patients ayant subi la chirurgie puis la requête renvoie le nombre de patients.

e) Page de recherche d'antécédents personnels d'un patient : Maud

A partir de la page centrale "donnees_patients_accueil", j'ai créé un premier formulaire permettant de rechercher le nombre d'antécédents personnels et leur catégorie à partir du numéro de dossier renseigné dans un champ text.

En cliquant dans le menu déroulant sur "Chercher ou ajouter un antécédent à un patient", on accède à une page php "chercher_ajout_atcd.php" où se trouvent plusieurs formulaires. Le premier, nommé "Chercher les antécédents personnels d'un patient" permet d'entrer le numéro de dossier du patient et d'appuyer sur un bouton de type submit pour accéder au résultat de la requête contenue dans le fichier php lié "atcd_pers_pat.php". Ce fichier php se compose de plusieurs parties à la manière des précédents :

- La connexion à la base de données.
- La définition de la requête SQL permettant de récupérer le nombre d'antécédents et la catégorie, groupé par catégorie, pour un patient à partir de son numéro de dossier.
- L'exécution de la requête SQL permettant de récupérer le numéro de dossier renseigné dans le formulaire de la page précédente et de l'associer à la variable "num_user".

- L’affichage des résultats permettant de parcourir la base de données et d’afficher le résultat associé au numéro de dossier renseigné.
- Le footer de la page est identique à toutes les pages du site.

Une amélioration possible de cette partie aurait été d’afficher un message d’erreur et de renvoyer à la page “chercher_ajout_atcd.php” si le numéro de dossier entré ne correspond à aucun numéro de dossier enregistré dans la base. Aussi, on aurait pu donner 3 tentatives pour entrer un bon numéro de dossier et afficher un compteur retranscrivant le nombre de tentatives utilisées et restantes.

f) Page d’ajout d’un antécédent personnel à un patient : Maud

A partir de la page centrale “donnees_patients_accueil”, j’ai créé un deuxième formulaire permettant d’ajouter un antécédent personnel à un patient à partir du numéro de dossier renseigné dans un champ text.

En cliquant dans le menu déroulant sur “Chercher ou ajouter un antécédent à un patient”, on accède à une page php “chercher_ajout_atcd.php” où se trouvent plusieurs formulaires. Le deuxième, nommé “Ajouter un antécédent personnel à un patient” permet d’entrer plusieurs informations : le numéro de dossier du patient, la catégorie du nouvel antécédent, s’il y a un traitement actuel lié à cet antécédent et un traitement passé lié à cet antécédent et enfin d’appuyer sur un bouton de type submit pour accéder au résultat de la requête contenue dans le fichier php lié “ajout_atcd_pers.php”. Ce fichier php se compose de plusieurs parties à la manière des précédents :

- La connexion à la base de données.
- La définition de la requête SQL permettant d’insérer dans la table “tab_atcdperso” une ligne avec le numéro de dossier du patient, la catégorie de l’antécédent et la présence de traitement actuel ou passé.
- L’exécution de la requête SQL permettant de récupérer la valeur renseignée dans chaque champ du formulaire de la page précédente et de l’associer à la variable correspondante (num_user, cat_atcd, trt_act, trt_passe).
- La vérification de succès de l’ajout de l’antécédent qui affiche “Antécédent ajouté avec succès” si la requête a fonctionné ou “Erreur lors de l’ajout de l’antécédent.” si la requête a échoué.
- Le footer de la page identique à toutes les pages du site

Une amélioration possible de cette partie aurait été d’afficher un message d’erreur et de renvoyer à la page “chercher_ajout_atcd.php” si le numéro de dossier entré ne correspond à aucun numéro de dossier enregistré dans la base. Aussi, on aurait pu donner 3 tentatives pour entrer un bon numéro de dossier et afficher un compteur retranscrivant le nombre de

tentatives utilisées et restantes. On pourrait également afficher un message d'erreur si les valeurs renseignées dans les différents champs du formulaire ne correspondent pas à des valeurs possibles de ces champs dans la base de données.

g) Pages “Voir le nom et prénoms par fonction des employés” et “Ajouter une grossesse”: Adrien

Pour voir les employés par fonction, la page "fct_service.php" a été mise en place. Cette page permet aux utilisateurs de sélectionner une fonction dans une liste déroulante. Ensuite, en fonction de la fonction sélectionnée, la page affiche les noms et prénoms des employés qui correspondent à cette fonction. Cette page fonctionne en récupérant les fonctions uniques depuis la base de données et en les affichant dans une liste déroulante. Ensuite, lorsqu'une fonction est sélectionnée et que le formulaire est soumis, les noms et prénoms des employés correspondants sont récupérés à partir de la base de données et affichés sur la page. La présence de l'apostrophe dans “personnel d'entretien” rendait l'appel à la base de données impossible. Le problème a été endigué avec l'utilisation de `htmlspecialchars($row['fonction'], ENT_QUOTES)`.

Pour ce qui est de l'ajout d'une grossesse, cela se fait à travers deux fichiers : "ajout_naissance.php" pour le formulaire et "submit_birth.php" pour le traitement du formulaire. Le formulaire présenté dans "ajout_naissance.php" permet à l'utilisateur de saisir le numéro de dossier du patient ainsi que son année de naissance. Ensuite, une fois le formulaire soumis, les informations sont envoyées à "submit_birth.php" pour traitement. Si les données sont valides, une nouvelle entrée est ajoutée à la table "tab_grossesse" de la base de données. Ces deux fichiers permettent l'ajout d'une nouvelle grossesse dans la base de données en vérifiant les données saisies et en effectuant les opérations nécessaires. Les messages d'erreur ou de succès sont affichés à l'utilisateur en fonction du résultat du traitement.

Forces

1. JavaScript pour la redirection automatique : grandement optionnelle mais pertinente. Implémenter un compte à rebours qui redirige automatiquement l'utilisateur après un certain laps de temps rend le site plus proche d'un site “réel”. Cela offre une meilleure expérience utilisateur en guidant les utilisateurs vers la prochaine étape sans nécessiter d'action supplémentaire de leur part. La page de redirection est cependant à revoir et peu pertinente.
2. Contrôles de saisie : Les formulaires de saisie des données comportent des contrôles de saisie pour s'assurer que les données fournies sont valides. Par exemple, le numéro de dossier doit être un nombre à 6 chiffres et l'année de naissance ne peut pas dépasser l'année actuelle.

3. Messages d'erreur et de succès : affichés à l'utilisateur pour lui indiquer si une opération a réussi ou échoué. Cela aide les utilisateurs à comprendre le résultat de leurs actions et à corriger les erreurs éventuelles.

Faiblesses

1. Code : le code souffre de nombreux problèmes de lisibilité. Par exemple, pour me sortir de certains problèmes, j'ai écrit du code CSS à l'intérieur de ces pages afin de venir à bout de certaines erreurs dont la source du problème était inconnue.
2. Absence de validation côté serveur : bien que des contrôles de saisie soient effectués côté client à l'aide de JavaScript, il est important de noter qu'ils ne sont pas suffisants pour garantir l'intégrité des données. Une validation côté serveur est nécessaire pour s'assurer que les données fournies sont correctes et sécurisées avant de les enregistrer dans la base de données. L'absence de cette validation expose le système à des risques de sécurité tels que les attaques par injection SQL.
3. Manque de gestion des erreurs robuste : bien que des messages d'erreur soient affichés en cas de problème, leur gestion pourrait être améliorée pour fournir des informations plus précises sur la nature de l'erreur et des instructions sur la manière de la résoudre. Par exemple, au lieu de simplement afficher "Erreur lors de l'ajout de l'antécédent", il serait bénéfique d'indiquer la nature spécifique de l'erreur, comme un problème de connexion à la base de données ou une violation de contrainte d'intégrité.
4. Potentielles vulnérabilités de sécurité : l'implémentation actuelle pourrait présenter des vulnérabilités de sécurité, notamment des failles XSS (cross-site scripting) si les données saisies par les utilisateurs ne sont pas correctement échappées ou filtrées avant d'être affichées sur le site. Il est crucial de mettre en place des mesures de sécurité telles que l'échappement des données et la prévention de l'injection de code malveillant pour protéger le site contre les attaques.
5. Limitations dans la gestion des erreurs de saisie : bien que des contrôles de saisie soient effectués pour vérifier la validité des données saisies, ils ne couvrent peut-être pas tous les scénarios possibles d'erreurs de saisie.

3) Pages communes :

a) Accueil

Les espaces de la page d'accueil ont été créés avec des balises <div>. Nous avons ajouté une photo permettant de rendre plus attrayant le site. Nous y avons ajouté 3 boutons permettant la connexion à un profil professionnel de santé, un retour à l'accueil et un affichage des données du service qui ne sont pas confidentielles.

b) Equipe

Cette page présente les 4 personnes principales du personnel du service. Nous avons travaillé sur le visuel de cette page. Un bouton contact permet de renvoyer à la boîte mail de l'utilisateur du site pour envoyer un mail à la personne de l'équipe que l'on souhaite. Nous avons créé plusieurs divisions avec des cadres pour présenter les 4 personnes.

Les pages suivantes (contact, FAQ, mentions légales) ont principalement été créées en termes de structure par Adrien, qui a expliqué au reste du groupe sa façon de faire. Ensuite, le groupe entier a participé à l'aspect, au remplissage et aux précisions de ces pages. Il s'agit de pages supplémentaires par rapport au travail demandé, nous ne rentrerons donc pas dans les détails de création de ces pages.

c) Contact

Cette page permet de contacter le service. Il faut bien remplir tous les champs et un est nécessaire dans l'adresse mail sinon un warning est renvoyé à l'utilisateur.

d) FAQ

Cette rubrique est présente dans beaucoup de sites internet, elle permet de répondre aux questions les plus demandées par les patients. Nous avons utilisé le même principe qu'une liste déroulante pour donner la réponse à la question afin de rendre la page plus interactive et ludique pour l'utilisateur.

e) Mentions légales

Les mentions légales sur un site internet sont importantes pour plusieurs raisons, principalement liées à la transparence, à la conformité légale et à la protection des droits.

4) Charte graphique

Nous avons choisi un design simple et minimaliste, et nous sommes plutôt concentrés sur la saisie de données. Nous avons tenu à garder les mêmes tons de couleurs dans l'intégralité du site, avec du noir, du blanc et quelques nuances de bleu. En effet, le bleu est la couleur principale de notre logo, il nous a paru judicieux de la conserver pour quelques éléments du site comme les boutons d'accès à certaines fonctionnalités. Nous avons aussi conservé ces nuances de couleurs dans la page "équipe".

Nous avons choisi un logo simple autour du thème de la vision. Il est présent sur toutes les pages du site internet.

Concernant la typographie, nous avons utilisé Roboto sans serif.

Enfin, nous avons utilisé peu de couleurs et peu de contrastes pour garantir un design agréable pour la lecture de l'utilisateur.

5) Sécurité, flexibilité, architecture et règles

Sécurité :

Nous aurions pu hacher les mots de passe pour la connexion du professionnel de santé. Nous avons pensé à ajouter “htmlspecialchars” qui permet de s’assurer que le contenu ne contient pas de balises HTML ou des scripts. La fonction permet de remplacer ces caractères spéciaux par des entités HTML. Le fait que seuls les professionnels de santé puissent se connecter pour avoir accès aux données de santé des patients marque aussi une forme de sécurité.

Flexibilité, architecture et règles :

Nous avons fait en sorte d’utiliser des modèles, qui sont inclus dans toutes les pages (navbar et footer) pour éviter de toujours copier les codes. Pour se faire nous utilisons la fonction include(). Concernant l’ajout d’images, nous n’avons pas inclus les chemins entiers afin que d’autres personnes puissent facilement tourner le code.

6) Discussion : limites et améliorations

Concernant les atouts de notre site internet, il est connecté à la base de données donc interactive. Puis, chaque lien mène à des fonctionnalités utiles et choisies. Les formulaires de contact redirigent vers les boîtes mail. Enfin, la barre de navigation et le footer présents sur toutes les pages est un avantage du site pour garantir une harmonie et une facilité d’utilisation.

Concernant les limites, sachant que l’étape d’importation des données n’a pas abouti en termes de relation entre toutes les tables, lorsqu’on supprime un patient par exemple, ses données ne sont pas automatiquement supprimées de toutes les tables. Nous aurions aimé relier toutes les tables et utiliser le type “cascade” . Par ailleurs, au regard du temps et de la demande pour l’UE, notre site ne contient pas beaucoup de fonctionnalités. Nous aurions pu en ajouter d’autres, telles que l’ajout de matériel par patient, la mise à jour des données des patients, l’affichage des types de lentilles proposées dans le service, la modification du mot de passe en cas d’oubli...

Annexe 1 : Schéma relationnel (uniquement les titres des tables sont indiqués pour faciliter la lecture).

