**IF3A : Projet site internet**



Palleau Tom ◊ Metral Greg ◊ Laduguie Alexandra

**TABLE DES MATIERES**

Introduction 3

Objectifs 3

Travail réalisé 4

Diagramme entité association 4

Tableaux 5

Fonctionnalités effectivement implantées 6

Problèmes rencontrés 11

Conclusion 13

Introduction :

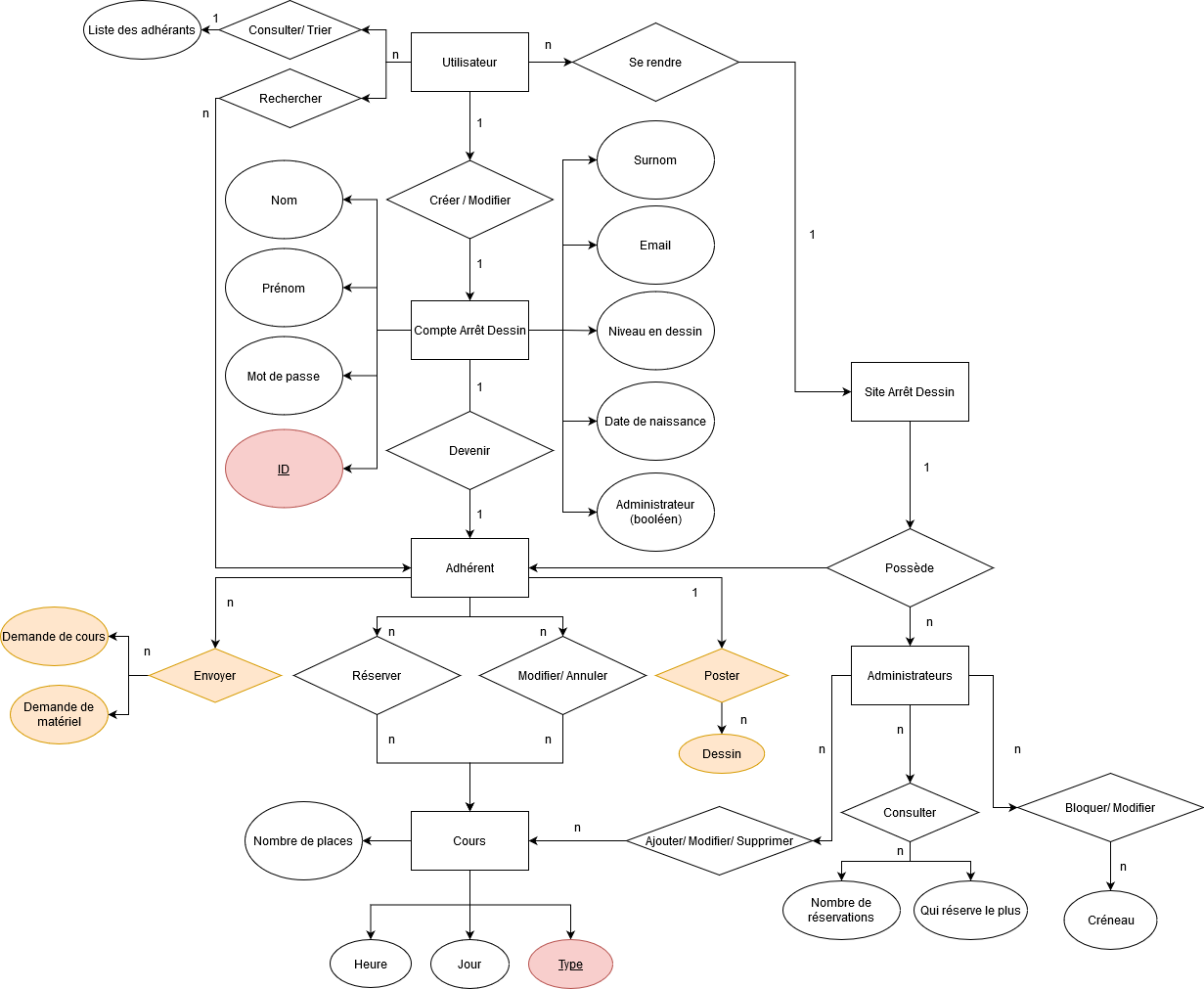
Le projet de fin de IF3A consiste à créer un site internet qui comporte plusieurs fonctionnalités. Face aux sujets proposés, nous avons décidé de créer un site pour le club Arrêt Dessin dont un des membres du projet en est la présidente. Notre sujet est assez proche de celui proposé qui portait sur le club de tennis ; il en comporte toutes les fonctions principales.



Objectifs du site

Les utilisateurs peuvent s’inscrire sur le site ArrêtDessin.com : ils possèdent un compte créé en saisissent leurs informations personnelles (date de naissance, email, mot de passe, nom, prénom, surnom, niveau en dessin) qu’ils pourront modifier par la suite. En tant qu’adhérant, Ils pourront ensuite consulter les cours disponibles (jour par jour et heure par heure) et en réserver un ; un cours a un nombre limité de place et un utilisateur peut ajouter un invité qui ne fait pas partie du club. Une réservation peut être modifiée ou annulée. Le site aura un rôle « administrateur ». L’administrateur peut consulter le nombre de réservations réalisées, voir qui réserve le plus. Il peut aussi modifier/bloquer un créneau ou ajouter/modifier un cours. Tous les utilisateurs pourront consulter la liste des adhérents, rechercher un adhérent ou les trier. Les adhérents pourront envoyer des demandes pour un cours ou du matériel, et auront la possibilité de poster leurs dessins. Un utilisateur non connecté n’aura pas accès aux dessins.

Travail réalisé

Diagramme Entité-Association****

Les tableaux

Pour atteindre une base normalisée en 3NF, nous avons du procéder étape par étape.

Table « utilisateur » non normalisée: Elle possède des listes de valeur et n’est donc pas en 1NF

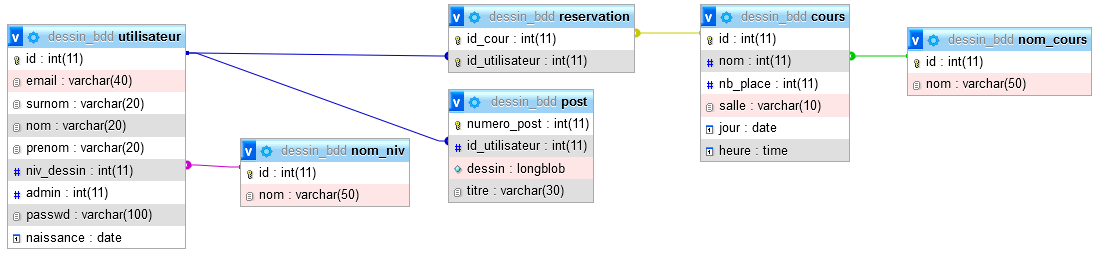
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Email | Surnom | Nom, Prénom | Niveau | Admin | Mdp | Date naissance | Cours |
| 1 | mail@mail.fr | JD | Jean, Dupont | Artiste | 1 | \*\*\*\* | 05/12/2001 | 10/11 p244 10h, 12/11 p145 14h |
| 2 | test@mail.fr | JM | Jean, Marc | Débutant | 0 | \*\*\*\* | 24/06/2002 | 09/11 p145 17h |

Table « utilisateur » : Elle est passée en 1NF mais toujours pas en 2NF ni 3NF

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Email | Surnom | Nom | Prénom | Niveau | Admin | Mdp | Date naissance | Cours (varchar) |
| 1 | mail@mail.fr | JD | Dupont | Jean | Artiste | 1 | \*\*\*\* | 05/12/2001 | 10/11 p244 10h |
| 1 | mail@mail.fr | JD | Dupont | Jean | Artiste | 1 | \*\*\*\* | 05/12/2001 | 12/11 p145 14h |
| 2 | test@mail.fr | JM | Marc | Jean | Débutant | 0 | \*\*\*\* | 24/06/2002 | 09/11 p145 17h |

La clé n’est pas composée de plusieurs colonnes, on peut donc directement transformer la base de données pour qu’elle soit en 3NF :

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Email | Surnom | Nom | Prénom | Niveau | Admin | Mdp | Date naissance |
| 1 | mail@mail.fr | JD | Dupont | Jean | Artiste | 1 | \*\*\*\* | 05/12/2001 |
| 2 | test@mail.fr | JM | Marc | Jean | Débutant | 0 | \*\*\*\* | 24/06/2002 |

La table « utilisateur » est ainsi en 3NF car chaque colonne dépend de la clé, et les colonnes non-clés comme par exemple « cours » ici est devenue indépendante. Nous avons ainsi rendu notre base de donnée « dessin\_bdd » en 3NF :

Elle comporte 6 tables ; la principale étant « utilisateur ». Les tables « reservation » et « post » sont liés à la table « utilisateur » par l’élément « id » qui est une clé primaire de cette dernière. De plus « reservation » est une table de passage car elle ne comporte que des clés étrangères ; elle permet de relier la table « utilisateur » avec la table « cours » et permet ainsi de savoir quel utilisateur est inscrit à quel cours. La table « nom\_niv » est relié à « utilisateur » par l’élément « niv\_dessin » de cette dernière. Enfin, la table « nom\_cours » est reliée à « cours » par l’élément « nom » de cette dernière. Elle permet de dire de quel type est le cours, et ainsi d’avoir la table « cours » en 3NF. Toutes ces modifications permettent d’avoir la table dessin\_bdd en 3NF, c’est-à-dire qu’elles sont normalisées.

**Fonctionnalités effectivement implantées**

**Demandes au club**

**Objectif**

L’objectif de cette fonctionnalité était que les utilisateurs puissent envoyer des demandes au club pour emprunter du matériel, demander un cours précis, ou bien faire une autre demande quelconque. L’envoie d’une demande enverrait un mail au club avec la demande de la personne.

**Travail Réalisé**

Au centre de la fonctionnalité se trouve un formulaire html qui comporte les détailler la demande. Nous avons ensuite créé un nouveau fichier pour se charger de l’envoie des mails suite au remplissage du formulaire précédent :

Une image contenant texte

Description générée automatiquement

(Voir commentaires)

**Conclusio**Une image contenant texte

Description générée automatiquement**n**

Le formulaire marche mais les mails ne pourront s’envoyer qu’une fois que le site sera en ligne, parce qu’en local host le mail ne s’envoie pas. On a alors une réponse positive de la part du programme mais le mail n’est pas reçu.

**Recherche de cours**

**Objectif**

L’objectif de cette fonctionnalité était pour les utilisateurs inscrits de pouvoir réserver un cours ou bien annuler leur réservation à un cours. Les inscrits ayant le rôle administrateur pourront en plus de cela ajouter ou supprimer un cours. Nous voulions aussi implémenter une fonction de recherche pour trouver un cours facilement.

**Travail réalisé**

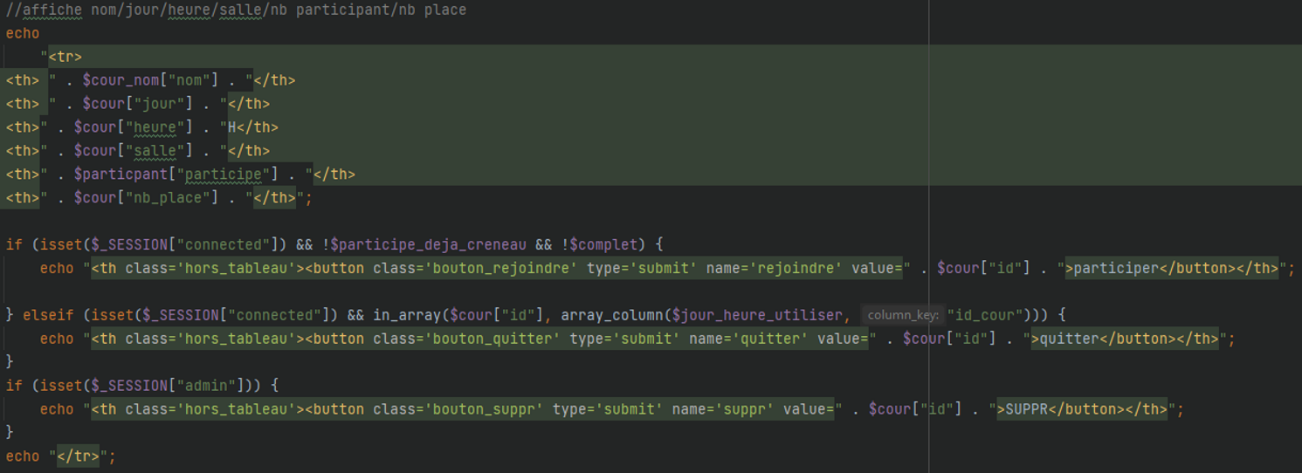
Comme précédemment on a tout d’abord créé un formulaire html qui demande de renseigner le type de cours, la date et l’heure de ce dernier. On a ensuite implémenté la fonction de recherche de cours selon les critères entrés par l’utilisateur.

Si l’utilisateur pose une condition alors on prépare la requête SQL qui peut sélectionner les cours correspondant dans la base de données

Une image contenant texte

Description générée automatiquement

Toutes les informations liées aux cours sont ensuite récupérées, et affichera le résultat dans un tableau html. Des boutons pour rejoindre ou quitter le cours seront aussi dans le tableau. Le bouton pour supprimer le cours ne sera visible qu’en session administrateur.

****

**Conclusion**

L’utilisateur peut rechercher facilement un cours par type, date ou heure, ou bien avec plusieurs de ces critères en même temps. Il peut aussi rechercher un cours en fonction des personnes qui s’y sont inscrites ou bien seulement voir les cours auquel il est lui-même inscrit.

Une image contenant texte

Description générée automatiquement

Il pourra ensuite s’y inscrire ou se désister. Les utilisateurs avec le rôle administrateur pourront aussi créer ou supprimer des cours.

Une image contenant texte, meubles, capture d’écran

Description générée automatiquement

**Inscription**

**Objectif**

Le but de cette fonctionnalité était de permettre aux utilisateurs de s’inscrire sur le site afin de pouvoir profiter des activités du club. Un utilisateur ne peut pas utiliser un email ou un surnom déjà utilisé par un autre utilisateur. Pour s’inscrire, un utilisateur devra donner quelques informations (Nom, prénom, surnom, email, niveau, date de naissance, un mot de passe).

**Travail réalisé**

Tout d’abord, un formulaire html permet aux utilisateurs d’enregistrer leurs informations, et d’ensuite les envoyer à la base de données. La page inscription renvoie alors à une deuxième page nommée inscription\_output, où le programme va vérifier si l’email et/ou le surnom est déjà présent dans la base de données. Mais le plus important est que cette page insert dans la table utilisateur le nom, prénom, surnom, email, niveau de dessin, date de naissance et le mot de passe associé (chiffré). Tout est alors stocké et peut être utilisé pour se connecter. La page renvoie ensuite a l'index si l'inscription est réussie

Une image contenant texte

Description générée automatiquement

Il pourra ensuite utiliser les informations rentrées pour se connecter.



**Conclusion**

L’utilisateur peut bien s’inscrire comme prévu. S’il utilise un surnom ou un mail déjà utilisé il n’est pas inscrit et un message d’erreur apparait en haut.

**Connexion**

**Objectif**

L’objectif ici était de pouvoir se connecter à une session unique à l’aide d’un email et d’un mot de passe. Le mot de passe doit correspondre à l’email associé pour démarrer sa session. Si l’email et le mot de passe ne correspondent pas, un message d’erreur s’affichera.

**Travail réalisé**

Un formulaire en html récupère dans un premier temps l’email et le mot de passe. Ensuite, le programme renvoie à la page connexion\_output. C’est sur cette page que le programme va vérifier que l’email et le mot de passe correspondent. S’ils correspondent, la session est ouverte et l’on est renvoyé a l’accueil. La page d’accueil change pour afficher « déconnexion » au lieu de « inscription » et « connexion ».



**Conclusion**

L’utilisateur peut se connecter a sa session et utiliser les fonctionnalités du site. L’utilisateur peut aussi être connecté en tant qu’administrateur, si la valeur de administrateur vaut 1 dans la base de données (elle ne peut pas être crée depuis le site).

**Problèmes rencontrés**

**Poster des dessins**

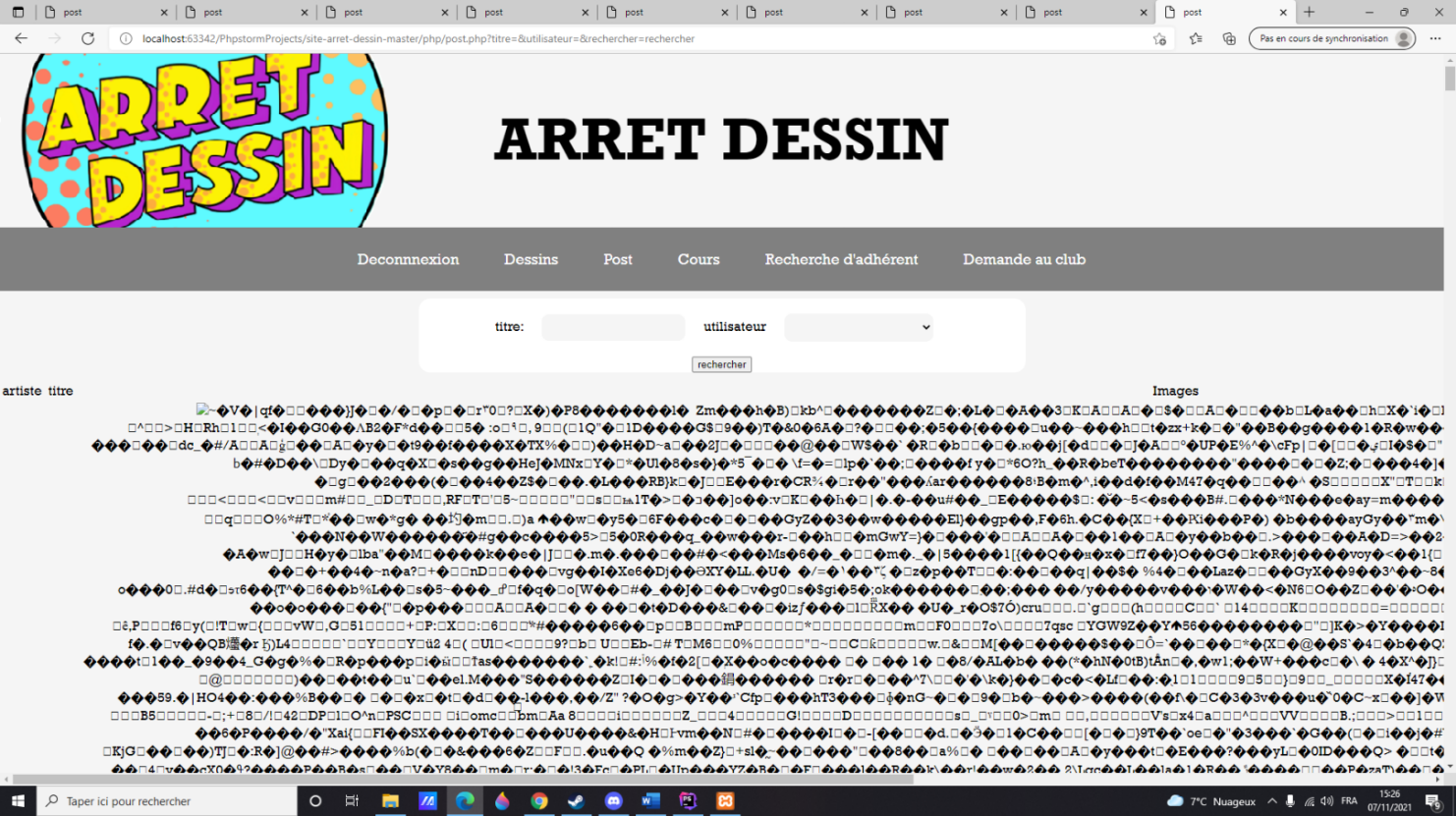
**Objectif**

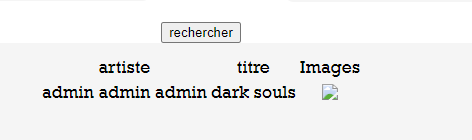
L’objectif de cette fonctionnalité était de permettre aux utilisateurs connectés d’envoyer des dessins et de consulter ceux des autres sur une page dédiée.

**Travail réalisé**

La fonctionnalité s’est révélée plus compliquée que prévue à mettre en place. Nous avons longtemps essayé de la faire marcher mais nous rencontré quelques problèmes :

* Affichage du code de l’image et non pas de l’image



* Affichage du titre et de l’auteur mais rien pour l’image
* Affichage de titre et de l’auteur mais l’adresse de l’image ne marche pas

**Conclusion**

Nous n’avons pas réussi à implémenter effectivement la fonctionnalité malgré nos nombreux essais.

**Autres problèmes rencontrés au cours du projet**

* Les formulaires étaient long à coder
* Il est devenu compliqué de se repérer lorsque l’on a commencé à avoir beaucoup de fichiers
* Nous avons eu du mal à faire en sorte que les données intègrent un tableau
* Nous avons aussi rencontré des problèmes avec le menu (éléments qui changent de position ou disparaissent) mais ils ont été réglés
* Le CSS a aussi causé quelques problèmes au début avec la mise en forme des menus et des formulaires

**Conclusion**

En l’espace de quelques semaines, nous avons atteint presque tous les objectifs que nous nous étions fixé pour le site Internet. Même si la fonctionnalité pour importer des images ne marche pas correctement, les autres fonctions marchent très bien et nous ont permis d’apprendre à coder en PHP, en HTML et aussi en CSS. Nous avons aussi pu nous entraîner à formuler des demandes dans des bases de données. Ce projet était donc bénéfique et nous avons beaucoup apprécier travailler dessus.