

Le sommaire

- Page 1 : Le site
- Page 2 : La base de donnée
- Page 3 : La creation d'un compte, L'activation, Le changement de mot de passe
- Page 4 : Connexion, Recherche par mots, Recherche par critères
- Page 5 : Affichage
- Page 6 : Information détaillé d'un travail
- Page 7 : Ajout d'un travail
- Page 8 : Difficultés rencontrées

Remarque : il est conseillé de lire au moins la page 2 pour pouvoir comprendre le reste des pages

Le site

Le site que nous avons réalisé est un site de partage de document, on peut y déposer des TDs et TPs pour les partager avec ses camarades de classe ou même ses élèves. Le site est fait pour être compatible avec les écrans de petite taille des téléphones portables.

Un fichier sql créant les tables de la base de donnée du site est aussi à votre disposition (attention : certaines fonctionnalités du site requièrent d'être hébergé sur un serveur notamment à cause de la fonction php "mail()").

Pour le navigateur, il est conseillé d'utiliser les dernières versions de firefox, d'internet d'explorer ou de google chrome (qui ont été testé) pour visualiser le site, disponible à l'adresse :

<https://st-sites.com/projet>

Les options disponibles sur le site sont :

- *Création d'un compte*
- *Connexion et Déconnexion*
- *Modification du mot de passe*
- *Recherche par université/année/filière/matière/td/tp*
- *Recherche par mots*
- *Ajout d'un travail sur la plateforme avec images/pdfs/documents*
- *Visualisation des travaux*

Les options qu'il reste à implémenter :

- *Modification des informations du compte*
- *Modification des travaux*
- *Système de commentaire*

Les options implémentés on été sélectionné pour rendre le site accessible et fonctionnel. Ici nous vous présentons un résumé des options implémentés.

La base de donnée

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
prenom	varchar(30)	NO		NULL	
nom	varchar(30)	NO		NULL	
pseudo	varchar(30)	NO	UNI	NULL	
password	varchar(255)	NO		NULL	
email	varchar(255)	NO		NULL	
activation	tinyint(1)	NO		0	
anonyme	tinyint(1)	NO		0	
filiere	int(11)	NO	MUL	NULL	
université	int(11)	NO	MUL	NULL	
année	tinyint(4)	NO		NULL	
code_activation	varchar(255)	NO		NULL	

table des utilisateurs : users

Voici la table des utilisateurs appelée "users", "id" est la clé primaire de cette table mais "pseudo" reste unique (nous utilisons un identifiant pour faciliter certaines manipulations). Un utilisateur choisi un mot de passe ("password")

qui sera encodé lors du stockage, il choisi aussi un pseudo, un nom, un prenom et un email, le boolean "anonyme" permet de savoir si nous affichons le nom et prenom ou le pseudo, le boolean "activation" permet de savoir si le compte est activé, le code d'activation permet d'identifier l'utilisateur avec l'id lors d'un changement du mot de passe ou l'activation du compte (il sera modifié à chaque fois). Chaque utilisateur a une filière et une université qui sont représentées par des clés étrangères respectivement pour la table "filières" et "universités". Un utilisateur a aussi une année (d'étude) entre 1 et 8 inclus.

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
nom	varchar(255)	NO	UNI	NULL	

table des universités : universités

Les tables "universités", "filières" et "matières" sont les plus simples, elles ont une structure identique : constituées d'une clé primaire "id" (étant un entier avec auto_increment comme dans la table précédente) et d'un champ "nom" unique. Les entrées de ces tables doivent être saisies à la main par un admin à la différence des autres tables. La présence d'un id permet de rendre les clés étrangères reliées à ces tables moins lourdes à stocker.

La table des travaux a pour clé primaire un identifiant "id". Chaque travail est représenté par un nom non unique, un type : TP=>1 ou TD=>0 représenté par le boolean "typ". Chaque travail a un auteur ("author" relié à "id" de la table "users"), une université, une filière et une matière (leurs champs étant reliées aux tables "universités", "filières" et "matières") et une année. Chaque travail peut aussi avoir un contenu ("text"), des images, des pdfs et des documents représentés par des booléens.

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
nom	varchar(30)	NO		NULL	
author	int(11)	NO	MUL	NULL	
typ	tinyint(1)	NO		0	
matiere	int(11)	NO	MUL	NULL	
université	int(11)	NO	MUL	NULL	
filiere	int(11)	NO	MUL	NULL	
pdf	tinyint(1)	NO		0	
image	tinyint(1)	NO		0	
document	tinyint(1)	NO		0	
text	longtext	YES		NULL	
année	tinyint(4)	NO		NULL	

table des travaux : travaux

La creation d'un compte

Creer un compte

Prenom :

Nom :

Pseudo :

Email :

Attention un email vous sera envoyé pour verifier votre compte

Mot de Passe :

Verification :

Université :

Paris Diderot

Filière :

Informatique

Année :

☐ Rester anonyme (identifier seulement par le pseudo)

Creer un compte

J'ai déjà un compte

Mot de passe oublié

Comme on peut le voir l'utilisateur choisi un prenom, un pseudo, un email, un mot de passe (on verifira qu'il est identique à Verification), une université présente dans la table "universités", de même pour la filière présente dans la table "filières", une année entre 1 et 8 inclus sera aussi à indiquer. Enfin on nous demandera si on veut rester anonyme.

Ainsi il ne reste plus que l'id, le code d'activation et l'activation à insérer dans la table.

L'id sera donné automatiquement, le code d'activation sera donnée "aléatoirement" grâce à "md5(microtime(TRUE)*100000)", un email sera envoyé à l'adresse email indiqué avec l'id et le code d'activation pour activer le compte, l'activation est donc initialisée à 0.

Le compte ne sera pas créé si le pseudo ou l'email et mot de passe sont déjà dans la table.

codeur : Dao Thauvin

code : creation.php => fonctions / nouveau.php => affichage

L'activation

L'activation se fait sur la page d'accueil avec le lien envoyé par mail sous la forme :

https://st-sites.com/projet/accueil.php?id='monid'&code_activation='moncode'

grâce au code d'activation et à l'id on retrouvera le compte qui aura alors un nouveau code d'activation et le booleen activation sera alors mis à 1. Un timesleep de 1 seconde à été ajouté à la fonction pour éviter une eventuelle attaque par brute force.

codeur : Dao Thauvin

code : validation.php

Le changement de mot de passe

Le principe est un peu le même que pour l'activation: avec le pseudo et l'email correspondant, on envoie un mail à l'adresse email avec l'id et le code d'activation du compte correspondant, le lien renvoie vers la page modif.php (sans les arguments valides dans l'url renvoie vers la page d'accueil) qui permet de modifier le mot de passe du compte correspondant, modifiant son code d'activation. Nous avons choisi de demander le pseudo pour simplifier le code (un email pouvant avoir plusieurs comptes).

Mot de passe oublié

Pseudo :

Email :

Attention un email vous sera envoyé pour debloquer votre compte

Envoyé

J'ai déjà un compte

Creer un compte

Changer de mot de passe

Mot de Passe :

Verification :

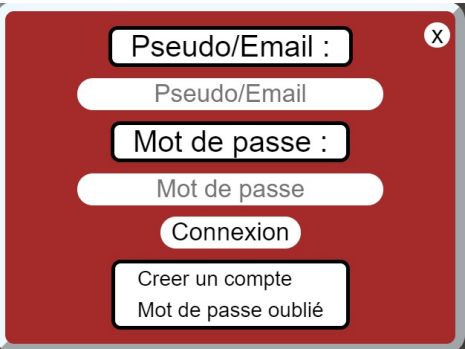
Valider

codeur : Dao Thauvin

code : retrouve.php => fonctions / modif.php => affichage

La connexion

La connexion est plutôt simple, on récupère les comptes ayant pour email ou pseudo l'information fournie et on vérifie si le mot de passe donné correspond à un des encodages des mots de passe récupérés dans la table users, si c'est le cas on met toutes les informations du compte dans une session. Une autre fonction "connecter()" vérifie si les informations de la session sont correctes par rapport à la base de donnée. Une redirection peut avoir lieu si nous sommes rentré sur une page non accessible sans connection (ex:add.php), "re" sera alors défini dans l'url.



codeur : Dao Thauvin

code : connection.php => fonctions / bases.php => connecter()

Recherche par mots



Il est possible de faire une recherche de travail par mots, c'est à dire par rapport aux mots que l'on saisi dans la barre de recherche. Les travaux correspondant (dont le nom contient au moins un des mots recherchés) seront affichés (affichage que nous verrons plus tard). Plus le nombre de mot correspondant est important, plus le travail sera mis en haut de la liste. La recherche par mot sera aussi mis en relation avec la recherche par critères que nous verrons juste après.

codeur : Dao Thauvin

code : options.php

Recherche par critères



La recherche peut aussi se faire par critères, on peut choisir de sélectionner seulement les TDs (les travaux dans la base de donnée où le champ "typ" vaut 0) ou seulement les TP (où "typ" vaut 1), nous avons choisi que l'effet de ne pas sélectionner TD et TP sera le même que de sélectionner les deux (il n'y aura donc pas de tri sur le type de travail).

On peut aussi sélectionner une filière ou/et une université parmi celle de la base, si "Tout" est sélectionné alors il n'y aura pas de tri dans ce domaine sinon on sélectionne les travaux avec la filière et/ou l'université correspondante.

Pour l'année, 0 indique qu'il n'y a pas de tri par année.

Les critères de sélection sont stockés dans des cookies pour pouvoir les réutilisés lors de la prochaine visite du site.

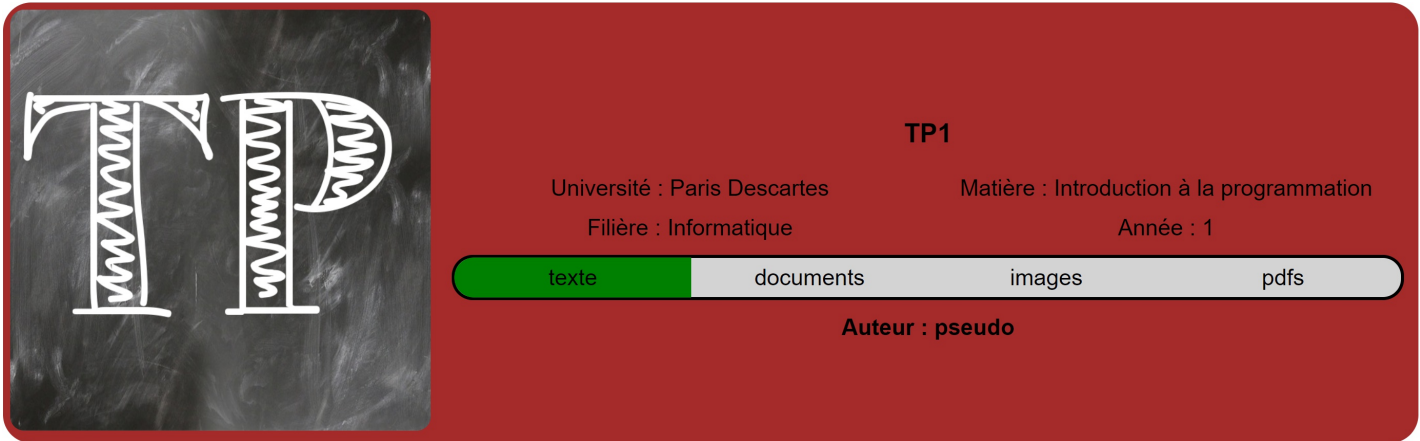
On peut combiner la recherche par critères et par mots. S'il n'y a pas de recherche par mot alors les travaux seront affichés par nom dans l'ordre alphabétique.

La recherche par critères se fait directement dans la fonction d'affichage "affi()".

codeur : Mohamed Diawara

code : options.php

Affichage



L'affichage des travaux se fait comme ci-dessus, on affiche d'abord s'il s'agit d'un TD ou d'un TP avec une image qu'on peut voir à gauche, ici il s'agit donc d'un TP ici.

On observe le nom du travail en haut à droite.

On récupère aussi les noms de l'université, de la matière et de la filière dans les tables "universités", "matières" et "filières" correspondant au numéro d'id des champs "matiere", "université" et "filiere".

On récupère de même le nom, le pseudo, le prénom et l'anonymat de l'utilisateur associé à l'id du champ "author".

Enfin on récupère le champ année mais aussi "text", "document", "image" et "pdf" dans la table "travail".

Si "text" est non null alors le fond du mot texte sera en vert (comme dans l'exemple si dessus).

De même si "document"/"image"/"pdf" est vrai (vaut 1) dans la table alors le fond du texte "documents"/"images"/"pdfs" sera en vert.

Voici la base de donnée pour l'exemple ci-dessous :

id	nom	author	typ	matiere	université	filiere	pdf	image	document	text	année
1	TP1	1	1	1	2	1	0	0	0	Le TP1 n'est pas disponible	1

La travail

id	nom
1	Introduction à la programmation

La matière

id	nom
2	Paris Descartes

L'université

id	nom
1	Informatique

La filière

id	prenom	nom	pseudo	password	email	activation	anonyme	filiere	université
1	prenom	nom	pseudo		test@gmail.com	0	1	2	2
2	1								

L'auteur du travail

codeur : Mohamed Diawara

code : options.php => fonctions / accueil.php => affichage

Information détaillé d'un travail

Après l'affichage des travaux, il est possible de selectionner un travail en particulier. C'est la page travail.php qui s'occupe de l'affichage, ainsi le travail affiché dépend de la valeur de "id" dans le tableau get (visible dans l'url). S'il n'est pas valide, la page redirige vers accueil.php. L'affichage se déroule comme ceci:

TP1

```
52 INSERT INTO 'filières' ('nom') VALUES
53 ('Math-Info');
54
55 -----
56
57 --
58 -- Structure de la table 'matières'
59 --
60
61 DROP TABLE IF EXISTS 'matières';
62 CREATE TABLE IF NOT EXISTS 'matières' (
63   'id' int(11) NOT NULL AUTO_INCREMENT,
64   'nom' varchar(255) NOT NULL UNIQUE,
65   PRIMARY KEY ('id')
66 ) ENGINE=MyISAM DEFAULT CHARSET=latin1;
67
68 INSERT INTO 'matières' ('nom') VALUES
69 ('Introduction à la programmation');
70 INSERT INTO 'matières' ('nom') VALUES
71 ('Concept Informatique');
72 INSERT INTO 'matières' ('nom') VALUES
73 ('Mathématique');
74 INSERT INTO 'matières' ('nom') VALUES
75 ('Internet et Outil');
76
77 -----
78
79 --
80 -- Structure de la table 'users'
81 --
82
83 DROP TABLE IF EXISTS 'users';
84 CREATE TABLE IF NOT EXISTS 'users' (
85   'id' int(11) NOT NULL AUTO_INCREMENT,
```

1

2

3

(1)(3).

- Comme ci-contre(1) le nom du travail sera d'abord affiché (ici TP1) puis s'il y a des images nous en affichons une, et des boutons rouges avec des numéros sont créés qui permettent de changer d'image visible (s'il n'y a qu'une image alors il n'y aura pas de bouton).

- Ensuite nous affichons le contenu texte (contenu dans le champ "text" de la table "travail") comme ci-dessous(2) dans un textarea qui ne sera pas modifiable.

- Enfin un lien de téléchargement des documents est affiché (s'il y a lieu) ainsi que les liens vers les pdfs, s'il y en a. Puis les informations sur le travail seront indiquées tel que la filière, l'université, l'année d'étude, la matière, son type et son auteur (voir

Contenu :

```
package examens;

public class CI {

    public static void main(String[] args) {
        System.out.println(wiboTerm(6));
        System.out.println(wibo(9));
    }
    public static int wiboTerm(int n) {
        if(n<=0) return 1;
        return recur(1,4,3,n);
    }
    public static int recur(int a,int b,int c,int n) {
        if(n==2) { return c; }
        if(n==1) { return b; }
        int e=a+b;
        int f=b+c;
        n-=2;
        return recur(c,e,f,n);
    }

    public static int wibo(int n) {
        if(n<=0) { return 1; }
        if(n==1) return 4;
        if(n==2) return 3;
        return wibo(n-2)+wibo(n-3);
    }

}
```

(2)

Télécharger les documents

Regarder le pdf numéro 1

Auteur : Dao THAUVIN

Filière : Informatique

Université : Paris Diderot

Type : TP

Année : 1

Matière : Internet et Outil

(3)

codeur : Mohamed Diawara
code : produit.php => fonctions / travail.php => affichage

6

L'ajout d'un travail

Pour accéder à la page d'ajout ci-dessus, il faut être connecté, sinon vous serez redirigé vers la page d'accueil.

Pour l'ajout d'un travail nous demandons :

- le nom du travail (champ "nom")
- le contenu (champ "text")
- s'il s'agit d'un TP ou d'un TD (champ "typ")
- La matière (champ "matiere")
- les Pdfs (champ "pdf", si présence de pdfs alors vaut 1)
- Les images (champ "image", si présence images alors vaut 1)
- Les documents (champ "document", si documents alors vaut 1)

Ainsi il ne reste que :

- L'id donné automatiquement (champ "id")
- L'auteur qui sera le compte connecté lors de l'ajout (champ "author")
- L'université qui sera celle de l'auteur (champ "université")
- La filière qui sera celle de l'auteur (champ "filiere")

Pour les documents, images et pdfs un nombre de document maximum et une taille maximum ont été imposés, de plus pour éviter des abus, les fichiers à double extension sont interdits (ce qui permet de vérifier plus facilement le type du fichier).

Les images et pdfs seront mis dans des répertoires (leurs noms sont les 6 premiers caractères de l'id + le nom du travail en remplaçant les caractères interdits par "-"), les noms des images et pdfs seront donnés "aléatoirement" grâce à un cryptage du nom du travail.

Pour les documents, ils seront compressés dans un dossier zip avec le même nom que les répertoires des images et pdfs.

codeur : Dao THAUVIN

code : ajout.php => fonctions / add.php => affichage

Difficultés rencontrées

Les difficultés que nous avons rencontré sont :

- *Trouver une base de donnée convenable à notre projet, notamment pour éviter la redondance dans les matières car il est possible qu'une matière se trouve dans plusieurs filières et universités mais nous devons aussi être capable de définir les matières d'une filière dans une université spécifiée (pour faire une recherche par matière de la filière de l'utilisateur), il a donc été difficile de trouver une base de donnée réunissant ces deux contraintes.*

- *Combiner la recherche par critères et par mots entre eux, nous avons fait séparément ces deux parties, il a été difficile de rendre le tout compatible.*

- *L'envoi de fichiers au serveur à été plutôt complexe car la sécurité sur ce point nécessite d'être très minutieux et nous espérons avoir éliminé le plus de faille possible.*

Nous espérons que ce petit résumé vous a plus et vous a été utile.

Mohamed Diawara - Dao Thauvin