**BTS INFORMATIQUE DE GESTION**

Option : Développeur d’applications

**EPREUVE E5**

**EPREUVE DE PRATIQUE DES TECHNIQUES INFORMATIQUES**

|  |
| --- |
| Nom et prénom du candidat :  VIGNERON Jean-Baptiste  N° candidat: 0313282587 |

Session 2011

|  |  |
| --- | --- |
| Académie de Lille | BTS Informatique de Gestion  Option DA |

**EPREUVE PRATIQUE DES TECHNIQUES INFORMATIQUES**

**Fiche de compétences**

Nom et Prénom du Candidat : VIGNERON Jean-Baptiste N°candidat : 0313282587

**Activité**

|  |
| --- |
| Titre de l’activité : MULTIWALL  Cadre de l’activé : Idée personnelle  Date de Lancement : Mars 2011 Fiche n°4 |

**Domaine de compétence principale**

**DA**

**PRODUIRE-DEVELOPPER**

**Compétences professionnelles mises en œuvre**

|  |
| --- |
| * **Compétence principale**   **C36: Développer dans le cadre d'une architecture client serveur** |

|  |
| --- |
| * **Autres compétences**   **C32 : Développer selon un algorithme procédural** |

**Liste des travaux**

|  |  |  |
| --- | --- | --- |
| Travaux effectués | Environnement matériel | Logiciels utilisés |
| Réalisation des images  Développement du logiciel | Core 2 Duo (2.1 Ghz)  3 Go de RAM  NVIDIA GeForce 8600M | Adobe Dreamwaver CS4  EasyPHP |

**Documents présentés**

|  |  |  |
| --- | --- | --- |
| Documents | Type | Observations |
|  |  |  |

**Configuration requise pour la présentation**

|  |  |
| --- | --- |
| Matériels | Intel Pentium 2 ou supérieur  Carte graphique 32 Mo DDR ou supérieur |
| Logiciels | Microsoft Windows 98 ou supérieur  Microsoft Visual Basic 6.0 |

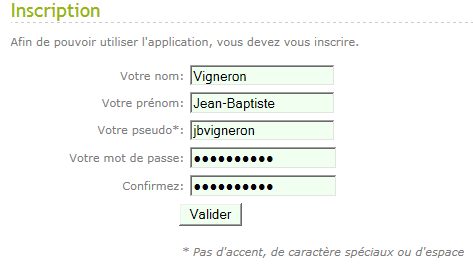
**MultiWall**

1. **Définition de l’activité**

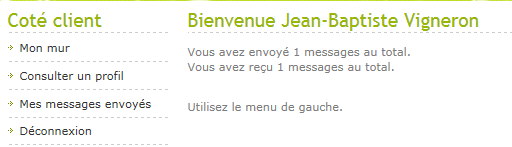
Ce projet est un système d’envoi de messages en reprenant le principe des murs Facebook ou Twitter.

1. **Spécifications générales :**

Afin d’utiliser l’application, il faut être enregistré. Si ce n’est pas le cas, le visiteur s’inscrit à l’aide d’un formulaire.



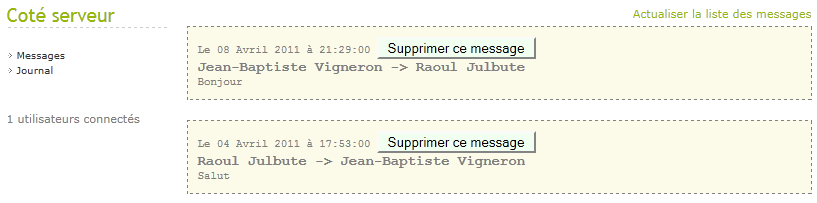
Une fois l’inscription terminée, le visiteur se connecte à l’aide de son pseudo et de son mot de passe.

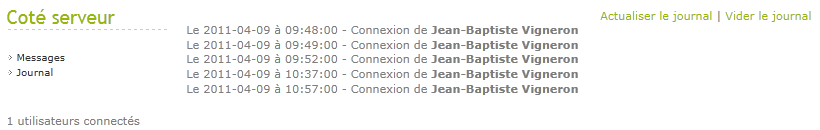


L’utilisateur peut ensuite consulter les sections suivantes :

* Son mur (ses messages reçus)
* Les profils des utilisateurs enregistrés
* Les messages qu’il a envoyé aux autres utilisateurs

Une partie « Administrateur » peut être lancée sur le serveur. Elle permet de regarder les messages échangés, le nombre de visites ainsi que le journal de connexion des utilisateurs.





1. **Spécifications détaillées :**

* **Un code exécutable sur le client**

Des scripts Javascript sont exécutés du côté client pour faire appel par exemple à AJAX ou pour effectuer des vérifications de saisies.

* **Un code exécutable sur le serveur**

Des scripts PHP sont exécutés sur le serveur, interprétés puis envoyés au client. Certains scripts servent par exemple à se connecter à la base.

Compte tenu que le Javascript peut être désactivé du coté client, il est important de tester les données coté serveur dans le but de renforcer la sécurité.

* **Un fichier sur le serveur**

Une base de données sous MySQL est mise en place sur le serveur afin de stocker les informations des utilisateurs, les messages, le journal de connexion ainsi que le nombre de connectés.

* **L’application doit fonctionner en multi-clients**

L’application est utilisable par plusieurs utilisateurs en même temps.

* **Un middleware**

La liaison entre le serveur et les clients s’établit à l’aide du protocole http, c’est-à-dire le port 80.

* **Respect d’une compétence secondaire :**

Développer selon un algorithme procédural (C32)

* + **Structure alternative (If Else)**

Cette application contient plusieurs structures alternatives dont l’une sert à contrôler si le message que l’utilisateur souhaite envoyer n’est pas vide.

**Coté client : js/profil.js**

|  |
| --- |
| if($('#message').val() == "")  {  alert("Entrez votre message");  }  else  {  …  } |

* + **Structure répétitive (For)**

Cette application contient plusieurs structures répétitives du côté client mais également du côté serveur. Un exemple figure juste en bas de la page.

* + **Structure de contrôle**

Une structure de contrôle composée d’une boucle ainsi que d’une condition permet d’interdire certains caractères dans le pseudo. Lorsque l’utilisateur rentre un caractère interdit, celui-ci est immédiatement effacé. La liste des caractères interdits est définie dans un tableau.

**Coté client : js/inscription.js**

|  |
| --- |
| var longueur = chaine.length;  var caracteres = new Array(' ', '&', 'é', '~', '\'', ……… ''!', '§');    for(var i = 0; i <= longueur; i++)  {  for(var j = 0; j <= caracteres.length; j++)  {  if(chaine.charAt(i) == caracteres[j])  {  var chaineSansEspace = chaine.substring(0, i);  chaineSansEspace += chaine.substring(i + 1, longueur);    chaine = chaineSansEspace;    i = 0;  j = 0;  }  }  } |

* + **Structure de données**

La structure de données est matérialisée par la base de données MySQL où sont stockées les données de l’application.

* + **Appels de procédure et de fonction avec passage de paramètres**

Une fonction existe contenant la structure de contrôle.

**Coté client : js/inscription.js**

|  |
| --- |
| function verifierPseudo(chaine) |

1. **Codage et test :**

J’ai développé ce site avec plusieurs technologies dont HTML, CSS, PHP, Javascript mais également AJAX. J’ai utilisé le logiciel Adobe Dreamweaver, permettant un affichage direct du résultat dans le logiciel contrairement à Notepad++, qui impose une vue du code source uniquement. De plus, Dreamweaver possède une fonction d’auto-complétion pour tous les langages, rendant le développement plus rapide et plus agréable. J’ai également utilisé le logiciel EasyPHP, contenant un serveur Apache, une base de données MySQL, un fichier de configuration PHP ainsi que l’interface de gestion PhpMyAdmin. De plus, j’ai utilisé un dernier logiciel qui est Win’Design, permettant la création de MCD, de le générer en MLD puis en script de création de tables pour MySQL.

J’ai développé dans un premier temps la partie Client puis ensuite la partie Serveur. J’ai également mis en place de l’AJAX pour l’affichage des messages à l’aide du framework jQuery, qui permet de développer du Javascript plus rapidement.

Afin de tester l’application, j’ai d’abord testé la connexion entre un client et le serveur ainsi qu’un second client. Celle-ci se fait à l’aider d’un navigateur Web. La connexion fut établie parfaitement. J’ai également mis hors ligne le serveur pour tester si les clients avaient une réponse mais ce ne fut pas le cas. J’ai ensuite effectué une batterie de tests standards. J’ai testé le formulaire d’inscription, les vérifications de saisie, l’envoi de messages ainsi que la consultation.

J’ai ensuite testé la partie Serveur afin de visualiser les messages échangés. J’ai terminé par le journal des connexions.

1. **Conclusion :**

En conclusion, ce projet m’a permis d’apprendre à utiliser le framework jQuery qui permet de simplifier le développement du code Javascript (coté Client). Il permet par exemple de simplifier les appels AJAX.