***TP Noté – Data Visualisation***

***Sujet :*** TP noté pour tester vos connaissances

**Enoncé :**

Vous êtes prestataire en Data-Visualisation et vous êtes chargés de travailler sur un projet de réseau social qui a récemment vu le jour.

Il s’agit d’un réseau social appelé : Students Meeting

Chaque étudiant possède son profil avec son pseudo, son email et son avatar. (On simplifiera ici tous les champs qu’il faudrait pour décrire un profil car on n’en a pas besoin pour le TP).

Chaque étudiant peut prendre contact avec un autre étudiant via la messagerie.

Chaque étudiant peut poster un statut.

Chaque étudiant peut ajouter un autre étudiant dans ses contacts.

Chaque étudiant peut donner une « note » à l’avatar/photo d’un autre étudiant (même si il n’est pas dans ses amis).

Le but de ce TP est de réaliser une page de statistiques qui sera peut être intégré au site.

Cette page doit être personnelle pour chaque utilisateur lorsqu’il y accède via un onglet « Statistiques » que l’on ajoutera sur son profil. (Paramètre GET avec le user id dans l’URL de la page à réaliser).

(Il est possible de récupérer un paramètre GET javascript)

**Pour se faire, une liste de web services est mise à votre disposition :**

* liste\_amis\_user qui renvoie la liste des amis d’un utilisateur donné ou d’une liste avec la date de mise en relation, sexe et âge.
* notations\_user qui renvoie la liste des notes que l’avatar de l’utilisateur donné ou d’une liste a générées.
* messages\_user qui renvoie la liste des messages d’un utilisateur donné ou d’une liste (on n’a pas besoin du contenu du message ici, juste de l’émetteur et du destinataire, la base est simplifiée).
* statuts\_user qui renvoie la liste des statuts pour un ou plusieurs utilisateurs donnés.
* infos\_user qui renvoie les informations sur un ou plusieurs utilisateurs donnés.

Ces web-services sont là pour vous faciliter le travail et vous évitez de perdre du temps sur les requêtes.

Vous devez passer en paramètre GET de la requête l’id de l’utilisateur dont vous voulez récupérer les informations OU une liste d’ID utilisateurs pour un résultat groupé sous la forme id\_user2, id\_user2, ect….

Si vous passez « 0 » en id\_utilisateur, le webservice vous renverra le résultat pour TOUS les utilisateurs.

***Exemple :***

infos\_user.php ?user=0 chargera les informations sur TOUS les utilisateurs.

infos\_user.php ?user=1 chargera les informations sur l’utilisateur d’ID 1.

infos\_user.php ?user=1,5,8 chargera les informations sur les utilisateurs d’ID 1, 5 et 8.

Ceci dit, il faudra quand même être capable de traiter les tableaux de données et de faire des algorithmes et de la mise en forme pour obtenir les graphiques. (C’est beaucoup plus proche de la réalité comme exercice, même si je me suis arrangé pour que les données ne soient pas à parser et soit lisible).

Si mes webservices ne vous conviennent pas, vous pouvez créer les votre. Les requêtes prennent 5minutes montre en main à créer et trouver vu la complexité de la base.

**Voici maintenant la liste des graphiques à faire et les points associés (à vous de réaliser les graphiques que vous pouvez, soyez stratégique sur votre temps !) :**

* Evolution du nombre d’amis au fil du mois. (2 points, JQplot Date Axis). (si vous faites cette question, choisissez un utilisateur intéressant avec beaucoup d’amis pour faire vos tests. Pour se faire n’hésitez pas à requêter sur la base pour trouver ce que vous cherchez)
* Evolution de la popularité (notation) au fil du mois (2 points, JQplot Date Axis, +1 point bonus si vous mettez la courbe de ce graphique sur le graphique au-dessus pour faire correspondre ces données)
* Pourcentage de messages envoyés à des amis et à des non amis (2 points, D3 Bar Chart OU JQPlot Bar Chart OU JQPlot Pie Chart).
* Pourcentage d’amis masculin et féminin (2 points, D3 Bar Chart OU JQPlot Bar Chart OU JQPlot Pie Chart).
* Popularité de votre profil pour un sexe donné par tranche de notation (pour chaque note possible, de 1 à 5) (2 points, JQPlot Pie Chart, + 2 points bonus si vous mettez un formulaire pour changer le sexe et recharger le graphique).
* Répartition des amis par tranche d’âge (18-21, 22-25, 26-29) et par sexe. (4 points, JQPlot Group Bar Chart.
* Faire un nuage de points qui montre la répartition de votre popularité (notation) auprès de vos amis par tranche d’âge (3 points, +2 point bonus si vous ajoutez un filtre pour le sexe sur votre graphique OU +1 point bonus si vous mettez un code couleur pour le sexe)

***BONUS :*** Faites apparaître sous forme de tableau (en D3.js, on génère un tableau HTML) le classement des amis les plus actifs de l’utilisateur (somme du nombre de statuts, messages et notations effectuées). (3 points + 1 point si vous faites figurer les avatars)

+2 points pour la présentation du rendu (page profil simplifiée de l’utilisateur avec les statistiques ?).

+2 point si la page est responsive bootstrap (mais peu de gens prendront le temps de le faire et c’est compliqué avec les graphiques)

+1 point pour la clarté du code.

+2 points si vous avez un graphique de chaque librairie (au moins un graphique JQPlot et un graphique D3.js).

***Quelques remarques :***

On a simplifié beaucoup de points dans la base actuelle.

Voici les éléments fonctionnels intéressants :

* Le site a été lancé le 01/06/2016 et votre étude se porte uniquement sur ce mois.
* Les notes vont de 1 à 5 pour une photo.
* Chaque photo est uniquement utilisée par un seul utilisateur (sinon elle est tout simplement dupliquée en BDD). De ce fait l’ID de la photo = l’ID de l’utilisateur
* Il existe un avatar par défaut « avatar-default.png » qui est attribué à un utilisateur lorsqu’il n’a pas choisi d’avatar.
* Les chemins d’image en base sont les noms de fichier, cela vous permettra de vous-même mettre votre chemin de dossier si vous voulez afficher les avatars.
* Il est fortement conseillé de montrer le pseudo et l’avatar du profil concerné par les statistiques pour une meilleure visibilité de rendu.
* Le sexe d’un utilisateur en base est 0 pour fille, 1 pour garçon. (car il est plus facile et rapide en terme de code de comparer des nombres que des chaînes).

***Initialisation du TP :***

* Récupérez l’archive du TP noté fourni par le prof et copiez la dans votre répertoire « www » de Wamp ou Xamp (ou autre, peu importe tant que vous ayez MySQL, Apache et PHP).
* Récupérez le dump de la BDD fourni également par le prof et mettez-le en local sur vos BDD.
* Récupérez d3.js et JQplot dans les anciens TP (ou sur MyGES)

N’oubliez pas JQuery !

Si vous avez le moindre problème d’initialisation, n’attendez pas 20 ans avant de me prévenir, chaque minute est précieuse pour augmenter vos points sur ce TP !

***Conseils pour avoir la meilleure note possible :***

* Bien vous inspirer des exemples des TP précédents.
* Amusez-vous à tester différentes configurations / options pour chaque Chart pour avoir des rendus qui vous paraissent beau et propre.
* Vous êtes notés sur la propreté de vos rendus aussi donc n’hésitez pas à y mettre du votre !
* Soignez aussi la présentation de votre code et la structure de votre rendu, j’en prendrais compte !

**Consignes de rendu :**

Chaque groupe doit m’envoyer à la fin du TP une archive contenant leur code source.

rspenato@gmail.com

Cette archive aura la forme :

E\_BUSINESS\_TP\_NOTE - NOM1 – NOM2 – EB1(2 ou 3, votre classe).

Pour rendre la notation la plus claire possible, je vous enverrai par mail votre barème détaillé en terme de points et les raisons qui ont fait baisser les points dans le cas où cela se produit.