

## Projet (1-2 personnes)

Vous devez réaliser un outil de notation de contenu pour sites Web, grâce à JavaScript, jQuery, Ajax et PHP. Grâce à cet outil, vos « clients » (des développeurs Web) pourront ajouter facilement sur leurs sites un composant de notation, qui permet aux visiteurs d'attribuer une note de 1 à 5 à tout type de contenu (pages, articles, commentaires, etc.).

Les éléments à concevoir sont :

- Un script JavaScript qui permet d'ajouter sur n'importe quelle page Web (voir plus bas « Intégration de l'outil ») un composant de notation.
- Un script PHP qui récupère et mémorise les notations des joueurs.
- Une feuille de style CSS pour habiller votre composant.

### 1 Intégration de l'outil

L'intégration de l'outil doit être très simple. Une personne désirant ajouter ce système de notation à sa page Web devrait juste avoir à faire les choses suivantes :

1. Copier le dossier *notation* (que vous fournissez, et qui contient tous vos codes) à la racine de son site.
2. Ajouter une balise `<script src="/notation/script.js"></script>` dans l'en-tête de sa page HTML, qui appelle un script que vous aurez créé.
3. Ajouter une balise `<div name="article-XX" class="notation"/>` là où il souhaite insérer l'outil (attention, on peut en utiliser plusieurs, avec des noms différents).

C'est le script */notation/script.js* qui se charge d'ajouter votre feuille de style à la page, et d'habiller toutes les balises *div* ayant une classe *notation* sur la page.

**Packaging.** Votre outil doit se présenter sous une forme simple à intégrer. Tous les fichiers qui le composent doivent être présents dans un dossier *notation*, et il doit être accompagné d'un fichier *lisezmoi.txt* qui explique comment (1) l'installer et (2) l'intégrer à une page HTML existante.

## 2 Le composant de notation

### 2.1 Client

Le client est codé en JavaScript (essayez d'utiliser au maximum jQuery). Il se présente sous deux états, suivant si l'utilisateur a déjà voté ou non.

En interne, le composant se présente sous la forme d'une balise *div*, avec un attribut *name* (qui a pour valeur une chaîne de caractères qui identifie une instance du composant) et un attribut *class* (qui vaut obligatoirement « notation »).

```
<div name="article-XX" class="notation"/>
```

Votre script *script.js* aura pour mission d'habiller ce *div*.

À l'affichage, le client doit se présenter comme suit.

Si l'utilisateur n'a pas encore noté, il se présente sous forme d'un cadre contenant 5 étoiles grisées. L'utilisateur peut cliquer sur n'importe quelle étoile pour donner une note (1 pour la première étoile, 2 pour la seconde, etc.). La note choisie est envoyée au serveur, accompagnée d'un identifiant du

composant (pour que le serveur sache sur quoi porte la note).

Une fois que l'utilisateur a voté, il se présente toujours sous formes d'étoiles, mais cette fois en affichant la note moyenne (par exemple, si la note moyenne est 2, il affichera 2 étoiles normales et 3 étoiles grisées).

D'un point de vue ergonomique, il est important que l'utilisateur comprenne que, dans le premier état, il est invité à noter ; puis, dans le second état, il voit la note moyenne. Vous pouvez utiliser des messages statiques, des animations, etc.

Le client obtient auprès du serveur (par Ajax) sont état (déjà noté / pas encore noté), la note (au moment où l'utilisateur note), et la note moyenne (si déjà noté). L'adresse IP de l'utilisateur est utilisée pour déterminer si il a déjà voté.

## 2.2 Serveur

Le serveur est codé en PHP. Il doit fournir les fonctionnalités suivantes :

1. Dire si, pour un nom de composant donné, l'utilisateur a déjà voté.
2. Retourner la note moyenne.
3. Enregistrer des notes. À chaque fois que le client lui envoie une note et un nom (valeur de l'attribut *name* du *div* qui contient le composant), il mémorise :
  1. le nom (du composant),
  2. la note,
  3. l'adresse IP de l'utilisateur.

L'adresse IP de l'utilisateur est utilisée pour déterminer si il a déjà voté. En PHP, l'adresse IP de l'utilisateur est donnée par la variable réservée `$_SERVER['REMOTE_ADDR']`<sup>1</sup>.

Le serveur ne doit pas nécessairement utiliser une base de données (vous pouvez utiliser des fichiers), mais ce n'est pas interdit non plus (cela est neutre au niveau du barème). Si vous en utilisez une, n'oubliez pas d'indiquer comment l'installer dans le fichier *lisezmoi.txt* (voir « Packaging »).

## 3 Documentation

Une **documentation** brève accompagne cette fois encore votre projet.

Elle n'est pas nécessairement longue, mais vous permet de présenter ce que vous avez fait, les particularités de votre travail, de défendre (ou non) certains manques ou certains compromis...

Les éléments nécessaires à la mise en oeuvre sont également indiqués (éventuelle configuration, particularités...). Enfin, indiquez tout ce que vous jugerez utile de porter à ma connaissance.

## 4 Ressources

Vous trouverez des icônes d'étoile sous licence GPL à cette adresse :

[http://commons.wikimedia.org/wiki/File:Crystal\\_Clear\\_action\\_bookmark.png](http://commons.wikimedia.org/wiki/File:Crystal_Clear_action_bookmark.png)

[http://commons.wikimedia.org/wiki/File:Crystal\\_Clear\\_action\\_bookmark\\_Silver.png](http://commons.wikimedia.org/wiki/File:Crystal_Clear_action_bookmark_Silver.png)

(mais vous pouvez en utiliser d'autres !)

---

<sup>1</sup> <http://php.net/manual/fr/reserved.variables.server.php>

## **5 Évaluation**

Comme lors du projet du semestre 1, plusieurs points seront évalués (liste non exhaustive) :

- « Aspect global » du travail : design, mise en page, lisibilité, ergonomie, navigation ... Le graphisme pur, n'étant pas l'objet du cours, n'est pas un aspect primordial (il est possible de faire un site esthétique avec peu d'éléments graphiques).
- Clarté et qualité du code : facilité de lecture (indentation, clarté ...), respect des recommandations actuelles du W3C (je pourrais utiliser les validateurs proposés par le W3C pour vérifier la syntaxe de documents).
- Bonne utilisation du HTML5 et des feuilles de style, qualité du code (en accord avec les éléments exposés en cours).
- Organisation des répertoires et documents, stratégie de nommage des fichiers...
- Prise en main facile, aspect « professionnel », aspect « fini », rapidité probable des mises à jours futures.
- Fiabilité, portabilité.
- Qualité du code, fiabilité.
- ...