**TP 2**

Un de vos amis vient de vous voir avec une idée extraordinaire, il vient avec ses amis de déterminer qu'un site de gestion des mises en chères sur les parties de football pourrait être extraordinaire et pourrait vous prendre que 5 à 10 minutes à faire. Il ne désire pas utiliser le fantasy game de la NFL car il pense que ce ne sera pas assez customizable.

Vous devez donc implanter un site de gestion des enchères sur les parties de footbal de la nfl. Sachant que ces demandes risquent d'être changeantes dans l'élaboration de votre projet, vous devrez l'implanter au moyen d'un MVC.

Le tout devra être utilisé en utilisant une base de donnée SQLite

**Téléchargement des futures parties et passées**

Après une brève recherche, votre ami vous suggère d'utiliser le site :

<http://espn.go.com/nfl/standings>

<http://espn.go.com/nfl/schedule>

Ils vous faudra obtenir la liste des équipes, les statistiques des équipes (PCT de gain), les résultats passés et les parties futures.

**PCT de gain**

Le pourcentage de gain est le nombre de victoire / nombre de parties.

**PF**

Le nombre de point pour (nombre de point marqués par l'équipe)

**PA**

Le nombre de point contre (nombre de point contre l'équipe)

**Gestion des utilisateurs**

**Aucun accès**

* Peut juste voir les parties futures et les résultats antérieurs. Il peut voir aussi voir le montant de gain s'il mise 10$ sur une partie.

**Utilisateurs**

* Acheter des jetons (paypal, carte de crédit) (à ne pas implémenter, seulement indiquer la possibilité)
* Leurs mises en cours
* Faire des nouvelles mises
* Voir leur gains antérieurs et leurs pertes

**Administrateurs**

* Gestion des utilisateurs
  + Inscription
    - 20 jetons donnés à la création du joueur
    - Le email est utilisé comme login
    - Les mots de passes sont encryptés dans la base de données.
  + Suppression
  + Modification de mot de passe
  + Modification du nombre de jeton

**Mises**

***Les mises antérieures ne sont pas influencées par les victoires dans le futur (une équipe qui gagne 3 fois après avoir fait la mise). Donc le gain potentiel est calculé et enregistré lors de l'entrée de la mise.***

* Les mises ne peuvent être appliquées que sur les parties à venir. N'importe quand avant que la partie commence.
* Les mises se font au moyen de jeton. Minimum de 10 jetons par partie afin de miser.
* Les mises peuvent être retirées et modifiées tant que la partie n'est pas commencée.

**Points totaux**

Une mise sur le nombre de points total qui sera inscrit dans une partie. Voici la façon de calculer le nombre de points estimés:

(PF\_A / Nbr\_Parties\_A + PF\_B / Nbr\_Parties\_B) / 2, arrondi à l'entier inférieur

\* Une réussite sur cette mise rapportera 150% de la mise.

**Différence de points**

Une mise sur la différence du résultat à la fin de la partie. Donc si l'équipe A marque 20 points et l'équipe B, 3 points, on a une différence de points de 17.

(|(PF\_A + PA\_B) - (PF\_B + PA\_A)|) / 4

* Le miseur peut choisir si le résultat final sera plus élevé que cette différence ou plus basse.
* Une réussite sur cette mise rapporte 150% la mise initiale.

**PCT de Gain**

* Le gain est calculé de la façon suivante:
  + À la base, les équipes ont 50% chaque de gagner la partie. Chaque équipe a un pourcentage de gain, l'équipe ayant le plus haut pourcentage est l'équipe qui devrait donc gagner.
  + La différence entre le pourcentage de gain de la meilleure équipe et de la moins bonne représente la différence de gain pour la partie en cours. On divise ce résultat par 2 et:
    - ajoute pour la meilleure équipe (GAIN\_A)
    - soustrait pour la moins bonne équipe (GAIN\_B)
  + Le calcul est effectué de la façon suivante: 200% moins le pourcentage calculé avant (A: 200% - GAIN\_A, B: 200% - GAIN\_B).
  + Donc par exemple, si A a 80% et B a 50%, la différence entre les deux est de 30%, donc A à (50% +30%/2) 65% de gagner et B (50% - 30%/2) 35%. Si on mise sur A, on pourrait gagner 135% (200% est la mise totale à laquelle on soustrait 65%), alors que pour B 165% (200% - 35% de la mise).

**Script**

* Dans un monde idéal, le script serait exécuté à chaque jour par votre serveur (à l'intérieur d'un cron job). Mais nous ne sommes pas dans un monde idéal, vous devrez créer un bouton dans votre page web qui pourra faire appel au script de mise à jour. **exec** est la méthode qui devra être appelée afin d'exécuter votre script python pour lancer la mise à jour de la BD.
* Un seul script est nécessaire.
* La qualité de la présentation du script python (méthodes)

**Mises à jour**

* Le système ajuste les statistiques des équipes nécessaires aux calculs des mises.
* La mises à jour des parties effectue l'opération suivante:
  + Ajustement des jetons (si victoire)

**AJAX**

* Lors de la connexion utilisateur, une bulle au haut de l'écran lui indique les résultats des parties, indique les jetons gagnés et perdus par rapport aux mises effectuées :
  + L'équipe A a gagné contre l'équipe B lors de la partie du xx novembre 2015. Vous avez gagné votre mise + x jetons.
  + L'équipe A a perdu contre l'équipe B lors de la partie du xx novembre 2015. Vous avez perdu votre mise de x jetons.
  + L'équipe A et B ont réalisé x points dans la partie. Vous avez (gagné | perdu) votre mise de x jetons (et gagné x jetons|)
  + L'équipe A et B ont eu un différent de x points dans la partie. Vous avez (gagné | perdu) votre mise de x jetons (et gagné x jetons|)
* Cette bulle ne disparaît qu'à la condition que l'utilisateur appuie sur un X dans cette info-bulle.
* Toutes les opérations s'effectue dans la grande majorité par AJAX.

**API**

* Les résultats des API sont du json.
* Un API permet à des systèmes distants de voir vos résultats à la CRUD:

Voici les éléments à compléter

/API/Teams

Montre la liste de toutes les équipes

/APIGames/

Montres la liste de tous les matchs à venir

/API/Games/x

Montres toutes les mises de la partie x sans montrer les infos des utilisateurs. Comme dernière ligne, il faut afficher le total des mises et le total des gains potentiels.

**Sécurité**

* Vous devez valider à chaque page (bonne idée de le faire dans chaque contrôleur sauf pour home et API) si l'utilisateur est connecté.
* Pour tous les formulaires: vous devez générer une variable de session qui s'assurera du temps de complétion du formulaire. Faîtes ça à même le controller.php (le parent)

**Remarques importantes**

La base de donnée doit être initialisée par l'entremise de l'application. Les utilisateurs suivants doivent être créés lors de la création de la table.

**Miseurs**

joe@joe.com 123

**Admin**

michel@michel.com 123

**Pondération**

| **Sujet** | **Note** |
| --- | --- |
|  |  |
|  |  |
| **BD** |  |
| Connexion | /1 |
| Création | /1 |
| **Modèle MVC** |  |
| Vue | /5 |
| Modèle | /5 |
| Contrôleur | /5 |
| **API** |  |
| Affichage JSON | /5 |
| **Parseur** |  |
| Mises à jour des statistiques | /2 |
| Mises à jour des mises | /3 |
| **AJAX** |  |
| Affichage des M-A-J des mises | /3 |
| Appel du MVC et MAJ de l'interface | /2 |
| **Lisibilité du code** |  |
| PHP | /3 |
| Python | /2 |
| Respect des normes HTML | /2 |
| Respect du modèle MVC | /3 |
| **Total** | **/42** |

**Remise**

15 décembre 2015 par LÉA