

TP2 – Node.js, Express et MongoDB

Pondération : 30 points

Date de remise vendredi 23 mars

Objectif:

Développer une application Node.js/Express/MongoDB permettant de contrôler la gestion d'une liste de membres

Objectifs spécifiques

Intégrer les fonctionnalités suivantes:

1 - Un lien permettant de changer la langue d'affichage du site

Les éléments à traduire:

- Les choix du menu principale
- L'entête de la liste des membres
- Le contenu entier de la page d'accueil doit être traduite
- Les étiquettes de la page Clavardage («Chat»)

2 - Traitement de la liste des membres avec AJAX

- Ajouter un membre
- Modifier un membre
- Détruire un membre

3 - Une application de clavardage

- Inscription d'un utilisateur
- Affichage de la liste des utilisateurs
- Gestion des messages
- Gestion de la déconnexion

Le site permettra d'accéder les pages:

- Accueil
- Liste des membres
- Profil d'un membre
- Clavardage

Intégration

- Utiliser **LESS** pour le développement des feuilles de style
- Les **préfixes** seront ajoutés dans les fichiers CSS par «**autoPrefixe**» de **sublime**
- Un fichier «**normalize.css** » sera utilisé
- Le site devra s'adapter aux différentes plates-formes : téléphone mobile, tablette, portable et ordinateur de bureau.
- La mise en page du site sera entièrement contrôlé par des «**Flexbox**» et utilisera des **media queries**
- Développer un système de «**gabarit**» (template) en utilisant le langage de template **EJS**
- Le système de gabarit devra être modulaire et contiendra au minimum des composants pour :
 - Entete
 - Menu
 - Tableau
 - Pied de page
- Le dépôt final devra contenir un minimum de 10 commits

Remise

- Un dépôt «**veille-node-tp2**» sur github
- Démonstration en classe du fonctionnement de votre application

Barème de correction (sur 30 points)

- **Les fonctionnalités (10 points)**
- **L'utilisabilité (5 points)**
 - L'adaptabilité multi plates-formes
 - Qualité du design
 - Utilisabilité de l'interface
- **Organisation de la codification Node.js/Express/Socket.io (10 points)**
 - Commentaires, nom de variable, organisation des routes
 - Utilisation d'ES6 (let, const et =>)
 - Organisation des modules/fonctions
 - Bonne utilisation des fonctions socket.io
 - Organisation de la codification AJAX

- **Qualité de l'intégration (HTML/CSS/LESS/JS (3 points))**
 - Organisation de la codification LESS
 - Organisation des gabarits EJS
 - Organisation de la codification JavaScript client
- **Minimum de 10 «commit» (2 points)**
 - remise sur le dépôt veille-node-tp2