Oral Informatique et science de l’information

1 er slide

Bonjour, je vais vous présenter le site internet que j’ai développé avec mon camarade de classe : Tao Schreiner. Nous avons choisi de réaliser ce site internet après avoir réalisé une présentation sur l’Affaire Snowden. Nous avons compris l’importance des questions juridiques et éthiques liés à cette affaire. Ainsi nous voulions intégrer un projet avec de la cryptographie. De là nous avons décidé après une longue discussion de réaliser ce réseau social/messagerie instantanée cryptée. Sur cette image nous pouvons voir l’affichage du profil, nous pouvons y observer une image de profil modifiable, une photo de couverture, ainsi qu’un bouton pour éditer son profil. La page affichée est en Anglais grâce au système de traduction.

2ème slide :

Le but de ce site internet est assez simple, nous voulons fournir à nos utilisateurs le moyen de communiquer avec leurs amis. Pour cela nous avons choisis un système le plus intuitif possible afin que cela soit simple pour eux.

Le contrôle quant à lui passe pas plusieurs moyens, nous utilisons tout d’abord la souris afin de cliquer sur les boutons d’envois, de connexion ou encore sur n’importe quels onglets présents sur notre site.

Nous utilisons aussi le clavier afin de remplir tous les champs présents tels que le champs d’envois de message ou les champs de modification de profil.

Nous utilisons aussi le bouton entrer comme bouton d’envois de message.

3ème slide.

Dès le début de notre projet nous avons établi un cahier des charges très précis afin de ne plus revenir en arrière par la suite dans notre projet.

La fonction primaire correspond à la fonction communication c’est-à-dire qu’il s’agit du cœur de notre projet. L’id correspond à un pseudonyme qui servira pour la recherche mais aussi lors de l’envoie des messages afin de savoir à qui l’on parle. Il ne peut excéder 10 caractères. Quant aux messages ils ne peuvent excéder 150 caractères. Pour ce qui est de la sécurité nous utilisons une fonction symétrique afin de crypter et de décrypter le message avec une clé définie par nous-même. Ainsi les messages sont cryptés dans notre base de donnée.

Pour ce qui est du front end, nous avons développé une interface ainsi que le système pour s’inscrire et se connecter. Nous avons aussi développé le système de photo de profil modifiable, ainsi que le système et l’interface de traduction. Nous avons aussi donné la possibilité d’accompagner les messages d’emojis.

Pour finir nous allons parler du langage serveur nous avons choisis d’utiliser le PHP. Pour ce qui est de l’hébergement et du stockage nous avons utilisés un serveur du rectorat ainsi qu’une base de donnée appartenant au rectorat. Nous avons régulièrement effectué des sauvegardes de notre code sur Git hub.

4ème slide

Nous avons dès le départ, un planning de travail après avoir établis le cahier des charges que vous venez d’observer. Nous avons voulu répartir le travail de la manière la plus égale possible. Cependant il n’a pas été simple car Tao possédait des connaissances supplémentaires au miennes. J’ai dû ainsi beaucoup plus me former que lui. Pour autant nous avions tous les deux déjà réalisé un projet informatique. Ainsi nous avions déjà connaissances des outils que nous pouvons utiliser afin de faciliter le travail en groupe. Malgré quelques retards ponctuels de quelques jours, nous avons tenu ce planning prévisionnel.

5ème slide

Nous avons réalisé ce schéma afin d’expliquer les différentes possibilités que vous avez en tant qu’utilisateur de notre site internet. Nous avons montré déjà deux possibilités, soit l’utilisateur s’inscrit soit il se connecte s’il il s’est déjà inscrit. Par la suite il peut ajouter ses amis, grâce à leurs pseudonymes. Si votre ami accepte votre demande, vous devenez amis et vous pouvez discuter. Il est aussi possible de changer la langue si vous êtes plus à l’aise en Anglais. Pour finir vous pouvez changer vos informations que vous avez choisis lors de l’inscription. Toutes ces possibilités peuvent se faire par la suite de l’inscription ou de la connexion.

6ème slide

Nous avons utilisé une architecture Modèle-Vue-Contrôleur il s’agit d’un patron de conception (design pattern en anglais) très utilisé dans la réalisation des sites web. Ce patron de conception est une solution éprouvée et reconnue permettant de séparer l'affichage des informations, les actions de l'utilisateur et l'accès aux données.

Concrètement c’est à dire que lorsque l’utilisateur demande d’accéder à une page, il n’accède pas vraiment à la page en question il accède en réalité à un fichier maître qui s’occupe de le rediriger vers les différents modèles, vues et contrôleurs nécessaires à la page demandé.

7ème slide

J’ai pour moi-même codé plusieurs systèmes mais aujourd’hui je vous propose d’étudier le système de notifications qui est comme tous notre code découpé en plusieurs parties.

Commençons par étudier les contrôleurs, on vérifie d’abord deux conditions, que le membre n’est pas dans la partie administrateur, on vérifie aussi que le membre est connecté.

Ces deux conditions sont essentiel car si le membre est dans la partie administrateur alors.. . Tandis que si il n’est pas connecté il ne voit pas la partie notifications que est réservé à un profil.

On demande dans ces cas-là, deux informations au model. Tout d’abord le nombre de notifications, et par la suite on récupère les notifications.

On demande ces deux informations afin de les afficher par la suite, nous observerons dans les vues pourquoi nous avons besoins de ces deux informations.

8 ème slide

Nous allons maintenant étudier les modèles, la fonction notification count récupère le nombre de notifications dans la base de donnée que l’utilisateur n’a pas encore vue.

La fonction get notificiation Récupère toutes les notifications ou l’id du membre et celui de l’utilisateur et les organise de manière décroissante. Cette fonction montre aussi celui qui envoi et celui qui reçoit la notification. En retour elle renvoi les notifications.

La table de notification est aussi utilisée dans les fonctions acceptant une requête d’ami et celle annulant une requête d’ami, dans les deux cas elle les supprime de la table notification

10ème slide

Nous observons mainteant les vues c’est-à-dire ce qui va gérer l’affichage des informations.

Tout d’abord il faut observer que le code s’exécute dans l’unique cas où le membre est connecté, autrement dit s’il possède un id. La liste à puce se place à droite sur la barre de navigation. La balise <li> représente un élément de la liste à puce déclaré dans la balise <ul> précédemment. A l’intérieur d’une balise < a> pour les liens, nous utilisons Font-awesome afin d’avoir l’icône d’une cloche symbole des notifications sur internet.

Par la suite on regarde si l’utilisateur à des notifications. Dans le cas où l’utilisateur a des notifications, alors on affiche le nombre à côté de la cloche.

Ensuite nous appelons la classe « dropdown » afin de créer un menu déroulant. Nous avons ensuite une condition avec deux possibilités, soit l’utilisateur n’a pas de notifications dans ce cas-là on affiche une chaine de caractère expliquant que l’utilisateur n’a aucune notification. Dans le cas où l’utilisateur a des notifications, on crée un lien cliquable grâce à la balise <a> renvoyant vers le profil de la personne ayant envoyé la notification. Nous créons aussi le visuel en affichant le nom de la personne ayant envoyé la notification.

11 ème slide

Vous pouvez voir dans ce tableau les différentes parties qui ne vienne pas de nous. Il faut cependant différencier certaines choses. Nous avons utilisé du code source pour le design de profile et celui du bloc de discussion.

Ce qui es complètement différent avec les bibliothèques JavaScript c’est que ce sont des fichiers déjà codé qui met à dispositions des méthodes et des propriétés pour l’ajout de fonctionnalité dynamiques. Handlebars.js est un moteur de Template qui sert à séparer l’interface graphique du reste du code. Nous avons aussi utilisé list.js qui est une bibliothèque JavaScript permettant de simplifier les recherches, catégories et filtres de recherches sur les barres de recherche.

Nous avons aussi utilisé un plugin JQuery se nommant timeago.js nous permettant de gérer le temps s’affichant lors de l’envoi de message et d’ajout d’amis.

Nous avons aussi utilisé Bootstrap, il s’agit d’un Framework CSS permettant de créer le design tout en gérant automatiquement le responsive design, c’est-à-dire l’adaptation sur les différents écrans.

Pour finir avons utilisé aussi Font-Awesome, il s’agit d’une bibliothèque créant les logos de manière vectorielle. Cette bibliothèque nous a permis d’avoir des icônes.

12ème slide

Pour finir nous observons maintenant, les problèmes subsistant dans notre code. Il reste quelques problèmes aux niveaux du responsive design, notre site internet est entièrement responsive cependant à certains endroits, le responsive design n’est pas assez intuitif.

Nous avons un problème que nous pouvons résoudre car, nous n’avons pas le droit au niveau du serveur du rectorat. L’équipe informatique du rectorat de Paris nous interdit le téléchargement vers le serveur des photos de profil.

Pour finit au niveau de la cryptographie, nous avons en effet utilisé un système de cryptographie assez simple, car la cryptographie est un domaine extrêmement compliqué. Nous devions crypter mais aussi décrypter le message envoyé, ainsi il n’était pas simple car nous étions face à des problèmes mathématiques.