

Projet de raisonnement propositionnel

Edouard.H Théo.R.V

2022–2023

Résumé

Ce document constitue notre compte rendu du projet de web. Il traite de la production d'un site internet de réservation de chambre et de dortoir, conformément à l'énoncé de ce projet.

Le site que nous avons donc produit se nomme **Taup'Hotel** (Jeux de mots avec sa ressemblance avec **Top Hotel** qui signifie en anglais 'le meilleur hôtel'). Nous avons décidé de séparer la production de celui-ci en deux parties. C'est pour cela que nous commenceront par une présentation du 'front-end', puis du 'back-end'. Pour finir, nous aborderons les difficultés liées à la production de ce projet et des diverses améliorations de ce site.

Table des matières

I	Le front-end	3
II	Back-end	5
1	Justification du choix de production d'une API.	5
2	Gestion globale de l'API.	5
3	Gestion de compte	5
4	La transmission de donner	5
5	La gestion des images d'annonce	6
6	Le scroll 'infinie' de la page de recherche	6
III	Conclusion	7
IV	Remerciment	7

I Le front-end

Qui par définition selon le site linkweb, correspond à ‘l’ensemble des éléments visibles et accessibles directement sur un site web’. Pour ce site, cela constitue la production de pages php créant diverse page internet et de leurs mises en forme à l’aide des fichiers css. Il s’agit d’ailleurs de la première chose que nous avons produit pour ce site.

Pour la réalisation de ces pages, nous avons tous d’abord réaliser des maquettes nous permettant de mettre en formes toutes ces pages. Après cela nous avons donc réalisé les élément les plus récurrents du site. La production du haut de page (‘header’) et du bas de page (‘footer’) a donc été l’étape suivante de cette production.

Le ‘header’ contient la redirection vers la page de recherche. En effet, nous avons fait le choix de créer une page aillant pour bute la recherche d’annonce. Contrairement à ce qui était demandé, de l’incorporer dans la page d’accueille. Au contraire, nous avons créé une page d’accueille présentant le site à l’aide d’avis fictife.

La partie la plus intéressante de cette page d’accueille concerne la carousel. Un simple jeu sur des modulus et l’utilisation de la fonction **setInterval** (fonction qui permet de répéter l’appelle à une procédure spécifier dans un interval donner) permet sa production.

Le ‘header’ donne aussi accès au page de connection et d’inscription (dans le cas où l’utilisateur n’est pas connecter). Ces deux pages on recourt au même fichier de style css. Une spécificiter intéressante de ces pages concerne l’animation css qui se déclanche en cas d’erreur sur un des champs du formulaire. Celle-ci à recours au ‘keyframe’ élément presque obligatoire à la production d’animation en css. Nous avons aussi eu de petit problème en ce qui concerne cette page, le fond étant une image (libre bien évidemment) en résolution 4k, on peut alors observer un fond blanc le temps de chargement de cette image.

Dans le cas où l’utilisateur est connecté, le ‘header’ donne accès à la page de compte. Cette page à plusieurs version en fonction du type d’utilisateur qui accède à cette page. Dans le cas d’un utilisateur basique, il a la possibilité de changer les informations de son compte, mais aussi de noter les chambres et dortoirs que l’utilisateur a déjà visiter. Pour un propriétaire de chambre, il a accès à la modifications des informations de ces annonces, mais aussi à leurs suppression. Enfin pour un administrateur, l’accès à la suppression d’annonce et d’utilisateur est donné dans cette page. On notera par contre, l’impossibilité pour un administrateur de réserver, de détenir des chambres. Une telle condition peut être justifiée par une limitation pour les administrateurs à influencer les choix des utilisateurs vers des chambres qu’il détiendrait. Enfin cette page donne aussi accès au divers ‘sessions’ du compte, point qui sera développer plus profondément dans

la partie sur le ‘back-end’.

En ce qui concerne le ‘footer’, celui-ci donne accès à la page ‘post’, qui permet d’ajouter une annonce. En ce qui concerne ce processus d’ajout d’annonce, nous avons décidé de rendre possible une telle action seulement pour les personnes aillant reçu le grade adéquate. Grade qui est demandable à un administrateur à l’aide de la page de compte. Cette contrainte sur l’utilisateur pour l’ajout d’annonce (de chambre ou d’hotel) permettra de limiter le nombre d’annonces à administrer pour les administrateurs du site.

En ce qui concerne cette page (la page d’ajout d’annonce), la partie la plus intéressante réside dans la production d’une prévisualisation d’image que l’utilisateur rentre pour cette annonce. Une telle fonctionnalité permet pour l’utilisateur de ne pas envoyer de photo possiblement compromettante à notre serveur. En ce qui concerne l’implémentation de cette fonctionnalité, une simple script js permet ce rendu. De plus, pour cette page, on note la ressemblance avec la page des annonces et/ou hotel.

Pour finir on peut évoquer le mode de fonctionnement d’ajout des annonces. Une annonce est soit un hotel, soit une chambre. Un hotel peut contenir plusieurs chambre et une chambre doit provenir d’un hotel. Enfin l’ajout d’une chambre à un hotel se fait lors de la création de la chambre. En effet, lors de cette création, l’utilisateur se voit forcer de lier la chambre qu’il vient de créer à un hotel.

Pour conclure cette section sur le ‘front-end’ le choix que nous avons fait de séparer le ‘front du back’ nous a permis une implémentation plus rapide de toutes ces pages php et de leurs styles. Les avantages pour le ‘back’ sont traités dans la deuxième section de ce compte rendu.

II Back-end

Dans cette partie seront évoquer la justification et production de l'API de notre site, mais aussi des liens permettant de lier le 'back au front'.

1 Justification du choix de production d'une API.

Le choix de la production d'une API, premièrement permet une séparation concrète entre le coter serveur et celui client. On comprend donc qui exécute quoi. Secondement, un autre 'front' pourrait être simplement implémenté. Il ne resterait alors qu'à reproduire les requêtes API semblables. On peut aussi envisager la possibilité de créer une application mobile ou encore un application desktop. Nous allons donc poursuivre cette section par une présentation des fonctionnalités de cette API.

2 Gestion globale de l'API.

Les requêtes faites à l'API sont toutes dans notre cas fait à l'aide de jquery de la commande **post**. Toutes requêtes sont traitées par l'API avec un module que nous avons élaboré **ratelimit**. Ce module permet de s'assurer qu'aucun utilisateur n'est malicieux, c'est-à-dire qu'il ne 'spam' pas notre API. Dans le cas de requêtes trop nombreuses pour un utilisateur, celui-ci va bloquer l'accès aux requêtes pour une durée de 5 minutes, valeur choisie arbitrairement.

3 Gestion de compte

La gestion des comptes prévue par notre API peut être vue comme très poussée. En effet, nous avons l'inscription et la connexion avec pour tous les champs concernés, une vérification de la validité de ces champs, une gestion des mots de passe en chiffré à l'aide de la fonction de hachage **sha512**. Mais nous avons surtout la gestion de session de compte. À la manière de grands réseaux sociaux comme tiktok ou même de moteur de recherche comme Google, notre site implémente cette fonctionnalité. Celle-ci correspond à la création d'une session à chaque nouvelle connexion. Une nouvelle connexion est définie comme étant la donnée d'une session n'existant pas encore. Le principe de ces sessions permet une grande amélioration de la sécurité des comptes.

4 La transmission de données

Comme évoqué plus tôt, la plupart de nos pages, effectue des requêtes pour accéder aux données qu'il les intéresse. Mais ce mode de fonctionnement ne permet pas la gestion de page comme celle de réservation de chambre. C'est pour cela que nous sommes passés par la fonction **urlParams**.

5 La gestion des images d'annonce

Les hotels et chambres étant obligatoirement représenté par une image, il nous a donc fallu trouver une méthode pour cette sauvegarde. Nous avons donc décidé d'utiliser l'API de imgure (site de partage de photo) pour la gestion de ces images. Cela permet de réduire l'espace nécessaire au serveur pour la gestion de ce site web en cas de déploiement. Cela pourrait aussi permettre à des utilisateurs de imgure d'être amené à réserver une chambre après l'avoir consulté sur le site.

6 Le scroll 'infinie' de la page de recherche

La page de recherche d'annonce pouvant contenir des dizaines de milliers d'annonces, une simple requête récupérant toutes ces annonces pourrait être très longues à recevoir et même à traiter. C'est pour cela que nous avons mis en place un scroll qui une fois arrivé au nombre d'annonces actuellement affiché re-éffectue des requêtes et donc affiche ces nouvelles annonces. Le 'ratelimit' de maximal de telles requêtes revient donc à une recherche maximale d'environ 1200 annonces toutes les 5 minutes, ce qui est plutôt raisonnable selon nous.

III Conclusion

Pour conclure, ce projet étant assez ouvert, nous avons donc pu innover par rapport au projet que nous avons produit l'année dernière notamment avec la production d'une API, ou encore le stockage des images sur imgur. En ce qui concerne les difficultés rencontrées, premièrement la production de ce compte rendu. N'ayant pas d'expérience dans ce domaine, nous ne savions pas de quoi parler ni même comment. Secondement, le temps, nous a fait défaut, plusieurs fonctionnalités et même pages n'ont pas pu être produites à cause de manque de temps. On peut notamment citer l'idée de rendre le site responsive (c'est-à-dire, produire des versions du 'front' en redimensionnant certaines parties du site pour le rendre plus accessible sur des appareils comme les téléphones). Il s'agit de la plus grande inconvenience de ce site, en effet, si l'utilisateur n'a pas une résolution générique de dimension 16 :9 alors une perte de lisibilité peut apparaître, cela rend même le site dans certains cas illisible.

IV Remerciement

Louis Dumontier, pour nous avoir produit la taupe qui nous sert de logo.