



*P.A INGHELBRECHT
Damien CLAEYMAN
Mickaël GRACA
Emrick DONADEI
Allan FERREIRA*

PROMO - 2021

Table des matières

Concept	2
Répondre à un besoin des consommateurs	2
Répondre à un besoin des commerces de proximité	2
Se démarquer pour créer un besoin	3
Étude du contexte et de la faisabilité du projet	3
Concurrence	3
Business Plan	4
Les fonctionnalités de Jäger	4
Consulter le site	4
Bénéficier des offres	5
Créer une offre	5
Géolocalisation	5
Travail d'équipe	6
Une organisation différente des anciens projets	6
Des supports de communication nouveaux	7
Stack technique	7
Front-End	7
Préparation du Front-End	7
Objectifs	7
Choix des frameworks	8
Charte graphique	8
Modifications apportées au cours du projet	8
Optimisation	8
Intégration	8
Dynamiser avec Javascript	9
Back-End	10
Choix des langages	10
Node.js	10
Langage de template: Nunjucks	11
Base de données NoSQL: MongoDB	11
Utilisation d'API	12
Infrastructure	14
Serveur en cloud	14
Systèmes Unix	14

Conteneurisation.....	15
Évolution chronologique du projet	15
Respect des deadlines	15
Apparition de nouveaux objectifs	16
Conclusion du projet	16

Concept

Répondre à un besoin des consommateurs

Afin de mener à bien ce projet transverse, nous sommes partis du constat que le besoin de faire des bonnes affaires, chez les consommateurs, se faisait grandissant. De plus, les consommateurs n'étaient pas prêts à parcourir des centaines de kilomètres pour bénéficier de ces offres.

Dans un souci d'identification de notre public-cible, il faut aussi comprendre que la majorité des salariés ne peuvent pas vraiment se déplacer de manière libre et sont contraints par leur travail, et n'ont donc pas le temps de pouvoir vraiment se lancer dans la recherche d'une promotion à une heure inhabituelle. Nous imaginons donc plus un public jeune en premier, possédant des horaires variables. (L'exemple d'un élève de l'EFREI !).

C'est pourquoi nous nous sommes penchés sur l'idée de créer une application ou un site web qui permettrait aux utilisateurs de bénéficier d'offres proche d'eux, à l'aide d'un système de géolocalisation.

Répondre à un besoin des commerces de proximité

Suite à nos multiples expériences personnelles, notamment lors de nos stages commerciaux, nous nous sommes retrouvé face à une problématique. Les boutiques ont toutes des périodes "creuses" durant lesquelles les boutiques voient l'affluence de clients très fortement réduite, leur faisant perdre du chiffre. Les frais salariaux et administratifs restent les mêmes selon chaque heure de la journée, alors que les rentrées non. Cela correspond souvent aux périodes moins fréquentées (matinées, jour de la semaine, etc...)

Nous avons donc cherché différents moyens de contrer ce problème inhérent à toutes les boutiques, et après concertation, nous nous sommes intéressés à un principe commercial utilisé fréquemment dans les bars, les **"Happy hours"**. Dans ce milieu qui est celui de la restauration, il est une problématique extrêmement classique de dépeuplement de sa clientèle en dehors des heures classiques. Mais on peut s'apercevoir que le phénomène se reproduit beaucoup dans les services de proximité.

Dans le milieu commerçant, notre public cible est celui de magasins "délaissés" géographiquement, et qui ont donc un besoin de relancer leur activité par un moyen non conventionnel. Pour imager cela, on peut prendre comme exemple deux commerces de proximité de l'EFREI, le supermarché Franprix, très exposé, et obtenant toujours des pics de rush à la fin d'un créneau horaire de cours, et le supermarché G20, étant à peine plus éloigné géographiquement, mais n'ayant pas un attrait suffisant pour attirer cet afflux potentiel d'acheteurs à bon pouvoir d'achat.

Se démarquer pour créer un besoin

Nous avons conscience de l'existence de milliers de services équivalents au nôtre, qui peut en premier temps n'apparaître que comme un simple annuaire de services. C'est pour cela que nous avons décidé de partir sur une stratégie du "jeu" sur un service qui n'est pas censé être fun à la base. Qui n'a pas eu vent du succès du jeu Pokémon Go, et du sentiment d'avoir réussi à attraper son Pokémon, et ce côté exclusif du fait que l'apparition de CE Pokémon soit limitée dans le temps.

Nous avons décidé de reproduire un sentiment équivalent chez l'utilisateur, en insufflant un côté "chasse au trésor" à notre application, en ne permettant pas à l'utilisateur de pouvoir prévoir dans le temps quelle nouvelle happy hours arrivera, et en laissant un sentiment d'exclusivité et d'intérêt à être le premier à y arriver. Ceci en incluant 2 outils très simples et ludiques :

- Affichage des annonces seulement si on se trouve dans un rayon proche de celle-ci
- Un temps limité pour pouvoir profiter de l'annonce
- Un nombre de places limitées, et une dégradation du pourcentage de promotion en fonction du nombre de personnes passées avant

C'est donc en partant de ce cahier des charges que nous avons commencé à engager une réflexion plus approfondie sur le contexte et sur l'existence d'éventuels concurrents, pour estimer quelle part de marché notre application pourrait emporter.

Étude du contexte et de la faisabilité du projet

Concurrence

Après un benchmark approfondi, se basant sur des recherches et sur notre propre connaissance du marché, nous avons identifié deux types de concurrence, pouvant faire de l'ombre à un éventuel projet dans ce domaine :

Les ventes privées / Les services d'achats en commun comme Groupon, ils permettent d'une certaine manière de s'assurer d'un certain afflux de consommateurs, tout en écoulant les stocks, il s'agit ici d'un concurrent direct. Même le côté ludique y est, n'hésitant pas à abuser des limitations dans le temps, et du côté exclusif et "privé" justement, que les consommateurs recherchent de plus en plus.

Les services d'enchère de services, comme le site internet "Loisirs enchères" qui a récemment explosé, il démocratise la vente de service comme celles d'un simple produit physique, qui en plus d'utiliser des ressorts comme celui des ventes privées, ajoute la compétition entre utilisateurs et la satisfaction de l'emporter sur l'autre.

Il n'est donc pas évident pour une jeune application de se faire une place dans un marché saturé, mais les utilisateurs cherchant de bonnes affaires n'hésitent pas souvent à faire l'effort, et la publicité pour un nouveau service peut se faire très rapidement par des sites comme **Dealabs** qui permet à des utilisateurs de recenser et de parler de bonnes affaires. Nous y ajoutons un côté ludique que nous avons beaucoup moins vu, car nous pensons que si l'utilisateur s'amuse, il retournera sur la plateforme.

Business Plan

Nous avons donc réfléchi à une mise en place d'un système viable pour pérenniser l'association sur le long terme, et pour cela nous y avons beaucoup réfléchi en même temps que notre stack technique. Effectivement, par besoin de scalabilité, car plus nous avons d'utilisateurs et de trafic, plus le besoin d'infrastructure se fait grand, nous l'avons résolu en optant pour un fournisseur de cloud computing leader sur le marché : **Amazon Web Services**, nous permettant non seulement d'expérimenter à loisir sans déboursier 1 seul euro, et de ne commencer à payer que si nous en avons vraiment besoin, pratique pour un projet web expérimental comme le nôtre !

Concernant le modèle de fonctionnement, nous sommes parti sur un modèle **Freemium**, proche de celui de Linked In, l'utilisateur est libre de s'inscrire et de profiter des services du produit, mais tout cela en acceptant que :

- Ses données ne lui appartiennent plus et peuvent être utilisées à des fins commerciales
- Tous les services de l'application ne lui sont pas immédiatement fournis, il ne peut y accéder qu'en souscrivant à un abonnement lui permettant d'accéder aux statistiques dont nous avons parlé précédemment ainsi qu'à des outils d'analyse de données simplifiés pour un usage immédiat.

Nous maîtrisons donc l'évolution de nos dépenses, et les faisons évoluer en même temps que l'évolution de nos recettes avec l'augmentation du trafic de la plateforme.


Les fonctionnalités de Jäger

Consulter le site

Tout d'abord, il faut avoir que Jäger est un site sécurisé et disponible sur tout type de périphérique. Vous pouvez dès maintenant consulter le site en libre accès sur www.jagerhours.fr, et cela gratuitement. Vous pouvez y consulter les différentes offres proposées, notamment : l'intitulé de l'offre, le temps restant avant la fin de ces dernières, le taux de réduction ainsi que la localisation. Il est également possible d'apprendre à mieux nous connaître et connaître le fonctionnement de Jäger dans la rubrique "Qui sommes-nous ?".

Franprix - Baisse des produits de repas

Paris

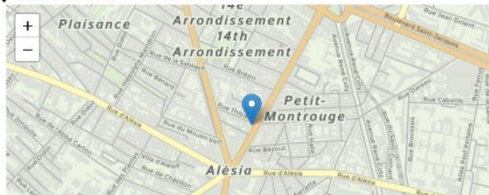
Type:  Accessibilité:   

Pars à la chasse:

Petite promotion ce midi seulement pour les clients de Jäger ! Profitez en !!

Localisation

 68 Avenue du Général Leclerc, Paris



30 % - 63
places

Vous l'avez raté !

00 : 00 : 00


Trouve d'autres Jagerhours


Bénéficier des offres


C'est bien beau de pouvoir consulter des offres, mais pouvoir en profiter c'est mieux ! Avant tout, pour profiter de ces offres, il vous faut vous connecter. Pour cela rien de plus simple, vous pouvez créer facilement et rapidement un compte Jäger en renseignant un identifiant, une adresse e-mail ainsi qu'un mot de passe et le tour est joué !


Vous trouvez ça trop long de créer un compte ? Vous avez peur de perdre votre mot de passe ? Pas de soucis, il est aussi possible de se connecter en deux clics à Jäger via votre compte Facebook, il vous suffit simplement d'autoriser à notre site d'accéder à quelques informations nous permettant d'associer votre compte Facebook à un compte Jäger.


Inscrivez-vous

 S'inscrire avec Facebook

 Identifiant

 Email

 Mot de passe

 Vérification du mot de passe

Inscription

Vous êtes un professionnel? [Inscription](#)

Créer une offre

Vous êtes un professionnel et vous souhaitez proposer une offre ? il vous suffit de créer un compte professionnel afin de renseigner les informations de votre commerce.

Ensuite, il ne vous reste plus qu'à éditer votre offre en indiquant son intitulé, sa description, votre géolocalisation, le nombre d'exemplaires ainsi que les dates de cette offre. Ca y est, votre offre est disponible et visible par les utilisateurs.

Nom de l'enseigne:

Nom de l'enseigne

Promotion:

%

Places disponibles:

Adresse:

Indiquez un lieu

Description de l'enseigne:

Description de l'enseigne

Début de la promotion:

jj/mm/aaaa --:--

Fin de la promotion:

jj/mm/aaaa --:--

Choisir une image

Browse

Géolocalisation

Pourquoi ne pas profiter d'offres autour de soi ? La géolocalisation vous permettra de voir les happy hours près de vous à n'importe quel moment. De plus, elle vous aidera avec un itinéraire à rejoindre le commerce qui possède la réduction si jamais vous ne le connaissez pas.

5

Travail d'équipe

Une organisation différente des anciens projets

Pour trancher avec nos anciens projets, nous avons opté pour une organisation très différente, et nous sommes passés dans une optique où tous les membres de l'équipe se sont mis à développer, permettant une plus grande compréhension du projet dans sa globalité par chaque personne.

Nous avons ensuite créé 2 équipes en un modèle très répandu dans le domaine du web développement. Une première équipe dite **front-end** s'occupant beaucoup plus de la partie graphique et visible de l'utilisateur, maniant les technologies d'affichage, pour permettre une expérience utilisateur la plus agréable possible. Étant une partie souvent plus légère techniquement, seulement 2 personnes se sont occupées de cette partie. Pour éviter de surcharger le github général, ils ont décidé de travailler sur un git personnel, et de rassembler de manière régulière leur travail pour se synchroniser avec l'autre équipe. Nous sommes d'ailleurs encore un peu déçu de ne pas avoir suffisamment exploité le concept de branches propres à Git, qui aurait pu être encore plus adaptés que la multiplication de repositories.

Lien du Github front-end:

<https://github.com/ClaeymanDamien/Front-End-Jager>

Les trois autres personnes ont donc décidé de s'atteler vers une grosse partie du travail, le **back-end**. En s'intéressant plus à la partie serveur, ainsi qu'à la mise en place de l'infrastructure pour permettre à la partie graphique d'être au maximum fonctionnel. Elle a été rendue difficile par le choix de technologies récentes et moins conventionnelles que le traditionnel LAMP sur une machine virtuelle, mais le temps d'adaptation en a valu la chandelle sur le long terme.

Lien du Github back-end:

<https://github.com/edonadei/jagerwebhappyhours>

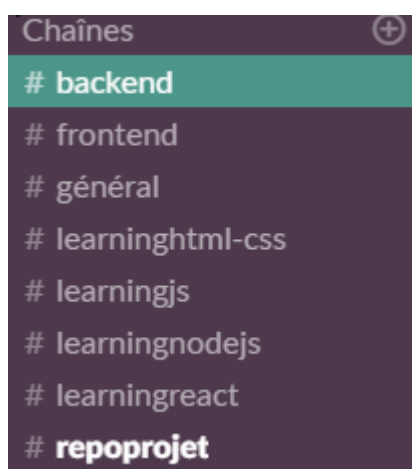
```
799 sudo git add -A
800 git status
801 git pull origin master
802 sudo git pull origin master
803 git pull origin master
804 git reset --hard
805 d mongo/
806 sudo chown ubuntu:ubuntu .
807 ll
808 git reset --hard
809 git pull origin master
810 cd app/uploads/
811 ll
812 sudo chown ubuntu:ubuntu .
813 cd ../../
814 git pull origin master
815 git reset --hard
816 git pull origin master
817 cd app/
818 cd public/
819 ll
820 cd uploads/
821 ll
822 cd ../../..
823 git reset --hard
824 git pull --rebase origin master
825 git status
826 cd mongo/
827 ll
828 cp data/ data2
829 cp data data2
830 mv data/ data2
831 ls
832 cd ../../
833 jagerwebapp/
834 cd jagerwebapp/
835 rm -r app/
836 ls
837 git pull --rebase origin master
838 git reset --hard
839 ls
840 git status
841 git pull --rebase origin master
842 cd app/
843 npm i
844 cd ..
845 lss
846 ls
847 git pull --rebase origin master
848 sudo docker-compose up -d web
849 sudo docker ps
```

Figure 1 - Exemple d'utilisation de Git pour debug un problème de merge

Des supports de communication nouveaux

Sans parler de l'utilisation classique des traditionnels messenger, discord et autres messageries vocales pour travailler à distance, nous avons donc continué à parfaire l'utilisation et la maîtrise de **Github**, nous avons d'ailleurs doucement transitionné de l'utilisation de la version Desktop, à celle de la ligne de commande, nous ouvrant plus de possibilité. (A droite, une historique d'un long coup de stress pour debug un problème de merge ...)

Nous avons aussi expérimenté comme à chaque fois un nouveau réseau professionnel, nous avons essayé de fonctionner avec Slack. Qui a été une expérience intéressante et réussie, car elle a permis de centraliser les discussions autour de sujets divers, sans perdre le fil, ainsi que de centraliser les ressources. (Ci-dessous nos channels de travail sur Slack)



Stack technique

Front-End

Le frontend n'est autre que la représentation visuelle de notre site. Par conséquent, c'est une étape clé dans la conception et réalisation d'un site web. En effet, il permet à la fois de mettre en valeur certaines informations, mais également de professionnaliser le rendu final, afin d'attirer un maximum de personnes et rendre l'utilisation ergonomique et agréable.

Préparation du Front-End

Dès le lancement du projet transverse, nous avons été amenés à réaliser quelques maquettes sur Photoshop, afin d'avoir une idée globale de ce qui pouvait potentiellement nous attendre en termes de travail. Néanmoins, nous n'avons pas réinventé la roue, et avons trouvé beaucoup d'inspirations sur d'autres sites internet, tel celui d'uber ou celui de Blablacar.

Objectifs

Notre objectif premier fut de mettre en valeur un maximum d'informations utiles dans un espace assez restreint, en restant dans la sobriété avec la règle du flat design. Un site surchargé n'est en effet par la meilleure des solutions pour attirer le futur utilisateur. Faire bonne impression en affichant les termes clés nous à sembler la solution la plus judicieuse. C'est ce que l'on a fait sur la page d'accueil avec la mise en place des 4 logos rappelant le concept du site :

Choix des frameworks

Pour éviter de perdre trop de temps à créer nous-mêmes nos propres frameworks, ou pour se prendre les pieds dans un CSS que l'on ne maîtrise pas encore, nous avons décidé de partir sur le leader du framework CSS: **Bootstrap**, il nous a permis de très rapidement créer des interfaces efficaces, design et surtout responsive, nous permettant de plus rapidement nous synchroniser entre les deux équipes. Mais on peut vraiment s'améliorer à l'usage du framework, et on a vraiment senti l'amélioration entre les premières pages construites au début du projet, et celles qui ont été finalisées sur la fin, il y a donc eu une marge de progression énorme.

Charte graphique

De toutes nos inspirations, nous avons décidé de rapprocher notre charte graphique de celle d'uber en particulier, très sobre de noir et blanc.

Modifications apportées au cours du projet

Au départ, nous avons mis en place une barre de recherche afin que n'importe qu'elle personne puisse rechercher l'offre qui lui correspondait. Cependant, après quelques débats sur le sujet, nous avons décidé de l'enlever par simplicité. Effectivement, l'utilisateur ne doit pas forcément chercher ce dont il a besoins en marquant ce qu'il souhaite. Nous avons préféré lui faciliter la tâche en implémentant des offres par secteurs d'activités : Services, Alimentation, Activités et High-Tech. Ainsi, il pourra trouver son bonheur, en affichant uniquement les promotions en rapport avec la catégorie sélectionné.

Optimisation

Tout au long de ce projet, nous avons fait en sorte que notre site web s'affiche convenablement sur n'importe qu'elle type de plateforme. Un PC, un smartphone ou bien même une tablette adaptée à chacune des fonctionnalités présente sur Jäger, rendant alors la visite beaucoup plus agréable pour l'utilisateur. Cette optimisation n'a pas été facile à tous les coups, bien que Bootstrap nous ai fortement aidé. Votre site peut très bien être bien optimisé pour un PC, mais être complètement détérioré sur un smartphone. Ce fut donc important pour nous d'adapter ce site sur n'importe quelle plateforme, d'autant plus que l'utilisateur sera probablement sur mobile

Intégration

Depuis la L1 nous avons eu beaucoup de projets en groupe. L'un des premiers problèmes du travail en équipe qui se posait était la répartition des tâches entre les membres. Au fil des projets notre cohésion et notre travail à plusieurs a évolué, passant d'un niveau qu'on peut qualifier d'amateur à un niveau se rapprochant davantage du monde du travail. Pour ce projet nous avons une répartition en deux grosses entités le front end et le back end. Chacun son langage et sa manière de coder. Notre objectif était de simplifier la lecture de notre code et l'adapter à tous types d'informations qui pourraient être intégré à partir de notre base de données. Pour la lecture on a donc utilisé le traditionnel commentaire pour délimiter les éléments de la page. Puis nous avons fait des exemples pour montrer à quoi correspondaient chaque balises. Ensuite, pour la partie intégration des informations provenant de la base de données, il fallait réfléchir à comment aller s'afficher les éléments à partir d'une certaine taille ou bien si on le multipliait x fois. Cette façon de travailler a été essentielle pour ne pas faire perdre du temps inutilement à l'autre équipe et pour garder le design intacte et responsive dans n'importe quel cas de figure.

Dynamiser avec Javascript

Bien que bootstrap permet d'avoir un site avec un jolie rendu. Il est nécessaire d'avoir un peu de dynamisme pour le rendre plus attirant auprès des utilisateurs. Mais comme toutes les bonnes choses il faut les utiliser avec modération. Nous avons décidé d'utiliser très peu d'animations pour rester dans un design assez sobre et pas too much, en revanche nous avons opté pour un javascript qui permettait d'utiliser le site de manière plus ergonomique.

Pour commencer nous nous somme attaqué à l'inscription des professionnels. Etant donné qu'une entreprise doit enregistrer beaucoup plus d'informations qu'un simple particulier. Nous avons décidé de faire un formulaire en plusieurs étapes pour le rendre beaucoup moins lourd à utiliser. Toujours pour simplifier l'utilisation utilisateur nous avons créé un système de filtre très simple et ergonomique.

En second point il fallait rendre le site plus dynamique avec des animations et un peu plus ludiques, challengeantes pour les particuliers. Nous avons donc mis en place un système d'annonce qui apparaissent pour une durée maximale de 23h59 et pour pimenter le tout il a été implanté un minuteur qui défile lorsque l'on visualise une offre et qui le suit l'utilisateur sur l'intégralité de la page.

Pour finir, les photos ont une importance capitale pour les particuliers c'est souvent ce qu'il est décisif dans leur choix final. Il fallait trouver une solution pour valoriser les photos des annonces. On a donc trouvé le carrousel slick qui nous a permis de mettre en avant les photos tout en gardant l'utilisation très simple et responsive. Le dynamisme du site a donc été très important, quitte à avoir pris le temps de faire beaucoup de recherches pour n'en utiliser qu'une faible partie.

Ma sélection	
Promotion: 10%	×
Villejuif	×
Fast-food	×
Affiner la recherche	
Promotion:	
<input type="checkbox"/> Promotion: 5%	
<input checked="" type="checkbox"/> Promotion: 10%	
<input type="checkbox"/> Promotion: 20%	
<input type="checkbox"/> Promotion: 30%	
<input type="checkbox"/> Promotion: 40%	
Lieu:	
<input type="checkbox"/> Paris	
<input type="checkbox"/> Arcueil	
<input checked="" type="checkbox"/> Villejuif	

G20 - Supermarché - Venez plutôt chez nous !

Meudon

Type: Accessibilité: P

Pars à la chasse:

Des offres super intéressantes pour concurrencer notre Franprix !

Localisation

64 Rue de la République, Meudon



32 % - 18
places

Elle va bientôt s'enfuir !

3 : 10 : 24

Marquage de la cible

Back-End

Choix des langages

Node.js

Après avoir benchmarké les langages serveur, nous avons longuement hésité entre trois, **Node.js en javascript**, **Django avec du python** et **Ruby on rails avec ... du ruby**. Notre choix s'est porté vers Node.js car nous avions prévu d'apprendre le javascript de toute façon, il nous a donc semblé plus judicieux de choisir cette technologie. Seulement nous avons mal prévu l'introduction de **programmation asynchrone** qui était très différente de ce que nous avons rencontré jusqu'alors.

C'est effectivement sur ce point en particulier que l'apprentissage a été le plus long, car on rentre vraiment dans un monde très différent de ce qu'on l'a appris à l'Efrei, et ceci basé sur un langage encore fraîchement appris, nous avons donc d'abord fonctionné avec des schémas existants, avant de passer des heures sur Stackoverflow pour réussir à comprendre chaque fonction, décortiquer chaque objet pour vraiment réussir à de plus en plus s'appropriier le langage pour de plus augmenter notre panel de fonctionnalités.

Nous avons d'ailleurs utilisé le middleware **Express**, très pratique et puissant pour contrôler le reste des dépendances donc la liste peut être consultée dans notre fichier package.json.

```
router.post('/:id?', (req, res) => {
  new Promise((resolve, reject) => {
    if (req.params.id) {
      Event.findById(req.params.id).then(resolve, reject);
    }
    else {
      resolve (new Event())
    }
  }).then(events => {
    events.name = req.body.name;
    events.hour = req.body.hour;

    // Adresse
    events.street_number = req.body.street_number;
    events.route = req.body.route;
    events.city = req.body.locality;
    events.state = req.body.administrative_area_level_1;
    events.zip_code = req.body.postal_code;
    events.country = req.body.country;
    events.latitude = req.body.latitude;
    events.longitude = req.body.longitude;

    // Autres informations
    events.description = req.body.description;
    events.promonumber = req.body.promonumber;
    events.number_avaible = req.body.number_avaible;
    events.types = req.body.types;
    events.datedebut = req.body.datedebut;
    events.datefin = req.body.datefin;

    if (req.file) events.picture = req.file.filename;

    return events.save();
  }).then(() => {
    res.redirect('/');
  }, err => console.log(err));
});
```

Figure 2 - Exemple d'une fonction utilisée typique de l'utilisation de l'asynchrone

Langage de template: Nunjucks

Il ne manquait plus qu'un langage de template, permettant de coder en javascript serveur directement sur les pages HTML, nous nous étions d'abord intéressés à **Angular** et **React**, tous les deux extrêmement puissants, et pas seulement pour du templating. Mais pour éviter de nous surcharger de technologies à apprendre, nous sommes restés sur une petite dépendance rapide à prendre en main et à apprendre : **Nunjucks**.

```
<!-- Announce Box -->
{% for e in events %}
<a href="{{e.id}}" class="container-fluid pl-3 pr-3 announce_feed d-flex align-items-center height-175 text_decoration_none">
  <div class="col-12 p-0 d-flex">
    <div class="col-3 col-lg-4 p-0 height-150 overflow_hidden d-flex justify-content-center align-items-center">
      <div class="img-responsive">
        
      </div>
    </div>
    <div class="col-9 col-lg-8 p-0 height-150 d-block d-lg-flex flex-lg-row">
      <div class="col-12 col-lg-9 p-0 pl-3 pr-3 d-flex flex-column justify-content-lg-between">
        <div class="p-0">
          <h5 class="text-dark">{{e.name}}</h5>
        </div>
        <div class="p-0">
          <span class="text-secondary d-block">{{e.street_number}} {{e.route}}, {{e.zip_code}} {{e.city}}</span>
        </div>
        <div class="p-0">
          <span class="text-secondary">{{e.number_avaible}} places restantes</span>
        </div>
      </div>
      <div class="col-12 col-lg-3 p-0 pl-3 pr-3 pl-lg-0 pr-lg-0 d-lg-flex justify-content-lg-center align-items-lg-center">
        <span class="text-danger d-none d-lg-block display-4">{{e.promonumber}}%</span>
        <span class="text-danger d-block d-lg-none fonts_20">{{e.promonumber}}%</span>
      </div>
    </div>
  </div>
</a>
{% endfor %}
```

Figure 3 - un exemple d'utilisation et de génération d'affichage d'annonce, les mots clés entre double crochet {{variables}} permettant de récupérer et de manipuler les variables serveurs récupérées de MongoDB puis envoyées par Node.js.

Il nous a permis de faire le lien entre les variables serveur, les variables locales en javascript, et récupérer les informations de la base de données, et ceci avec une syntaxe simple directement dans le HTML.

Base de données NoSQL: MongoDB

Par choix, nous avons décidé de ne pas partir sur du **MySQL** classique, mais sur du **NoSQL** pour aller avec notre optique de scalabilité ainsi que pour apprendre une nouvelle technologie, nous avons donc utilisé **un serveur mongoDB**, enregistrant toutes nos entrées en json, donc sous forme d'objet javascript. Ce qui était extrêmement cohérent avec notre stack de technologies très basé sur le javascript.

De plus, pour nous permettre de simplifier nos requêtes, et créer de véritables modèles de base de données, nous avons utilisé le paquet **Mongoose** étant une surcouche de MongoDB pour vérifier la cohérence des données entrées par notre application. (Ci-dessous un aperçu proche d'un PHPMyadmin, généré par l'application studio 3T, qui nous a beaucoup simplifié le travail pour des opérations spécifiques sur nos données et à des tests)

_id	name	date	description	promonumber	_v	city	country	route	state	street_number
5a160b53b905...	Frangprx - Baiss...		Petite promotio...	30	7	Paris	France	Avenue du Gé...		68
5a160c01b905...	Fnac opéra - R...		Belle promotio...	45	5	Paris	France	Boulevard de l'...		64
5a160b2b23e45...	Le Bon Pain - ...		Tous les inven...	20	4	Cachan	France	Avenue Aristid...		49
5b005ecfeeeed...	G20 - Superna...		Des offres supe...	32	1	Meudon	France	Rue de la Répu...		64
5b0465663f61...	Salon de coiffu...		Un coiffeur typ...	50	1	Villejuif	France	Rue Georges L...		34
5b04684a1edb...	Franck Provost		Un salon d'exc...	20	1	Villejuif	France	Rue Jean Jaurès		61
5b0469501edb...	Cosmic Laser		Si vous arrivez ...	50	2	Thiais	France	Rue des Alouet...		3
5b046c669b98...	The Game		Préparez vous ...	45	2	Paris	France	Rue du Cardin...		51
5b0535b7da55...	EF Ride		EF Ride asso sk...	30	1	Villejuif	France			null
5b053715da55...	Brasserie des S...		Brasserie à deu...	20	0	Villejuif	France	Avenue de Stal...	Île-de-France	3
5b053889da55...	Maihak		Restaurant Indi...	15	1	Villejuif	France	Rue Jean Jaurès		102
5b0668581029...	EFREI PARIS		Ecole d'ingé	70	0	Villejuif	France		Île-de-France	null

Figure 4 - un aperçu proche d'un PHPMyadmin, généré par l'application studio 3T, qui nous a beaucoup simplifié le travail pour des opérations spécifiques sur nos données et à des tests

Utilisation d'API

Lors de la réalisation de ce projet, nous avons dû utiliser des API. Une API, Application Programming Interface, est une interface de programmation qui permet de rajouter un service à notre projet. Elle se compose de classes, de méthodes et de fonctions qui permettent d'améliorer l'utilisation du projet.

Dans notre cas, nous avons utilisé Google Autocomplete. Lorsque l'on rentre une adresse, Qui n'aimerait pas qu'une sélection se fasse automatique, en fonction de ce que l'on a déjà écrit, pour éviter de devoir taper toute l'adresse. C'est ce que propose cette API. De plus, pour le programmeur, il est donc facile de récupérer la longitude et la latitude, l'API gère et fournit ces deux données.

Adresse:

30 Avenue de la République, Villejuif, France

Street address	30	Avenue de la République
City	Villejuif	
State	Île-de-France	Zip code 94800
Country	France	
Latitude		
Longitude		

L'API de google à la capacité de faire la différence entre le numéro, le nom de la rue ou encore celui de la ville. Grâce à de grosses bases de données, elle est donc en capacité de donner des adresses complètes. Pour faire le lien avec notre projet et notre nécessité de faire une map, cette API, nous a beaucoup simplifier le travail car nous avons pu récupérer directement la longitude et la latitude. La longitude et la latitude n'ont d'utilité que sur une map. La seconde API est donc une carte, c'est Leaflet. Elle nous permet alors de permettre à l'utilisateur de savoir où est ce qu'il se situe par rapport au commerçant qui propose un happy hours. Il y a en plus de la map, une multitude de fonction

permettant par exemple, de rajouter des pins, géolocaliser la personne ou encore de créer un itinéraire.

D'un point de vue du code, par exemple la géolocalisation se code très rapidement : il suffit d'utiliser la commande suivante :

```
var location = e.latlng;
```

On récupère donc dans la variable location, les coordonnées géographiques de l'utilisateur. On va donc pouvoir ensuite faire apparaître, un itinéraire entre l'utilisateur et le magasin. Une fois de plus, c'est très simple avec Leaflet. Il suffit de faire appeler aux fonctions présentes avec l'API et de lui donner les bonnes valeurs.

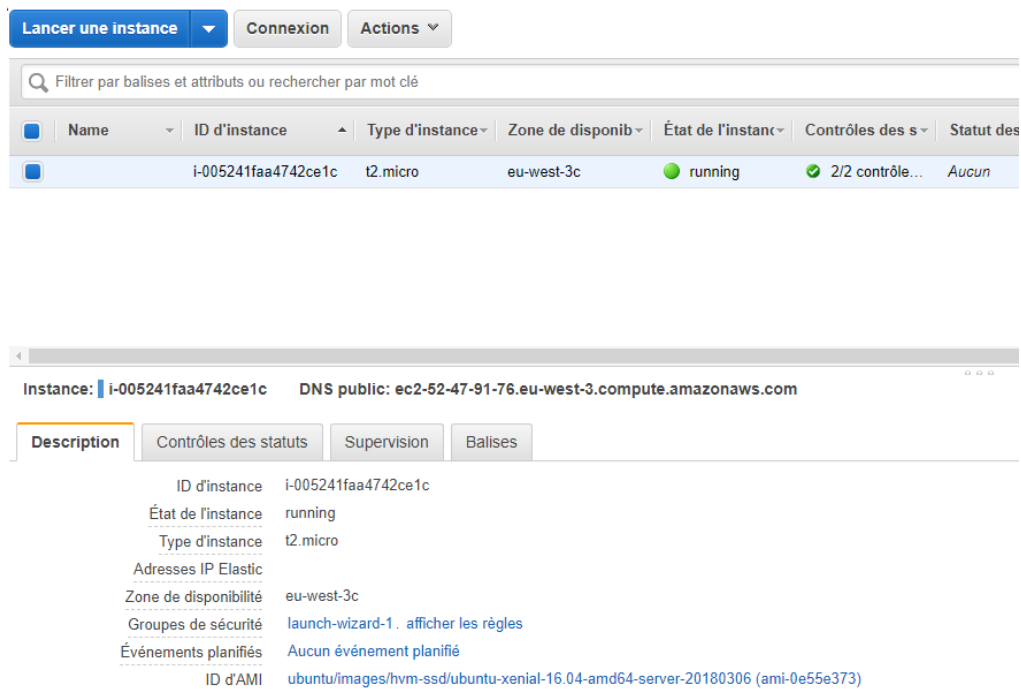
```
L.Routing.control({  
  waypoints: [  
    L.latLng(magasin),  
    L.latLng(location)  
  ],  
  language : 'fr'  
}).addTo(map);
```

Pour réaliser ce projet, nous avons préféré Leaflet à Google Maps car dans un premier, la map nécessite beaucoup moins de ressources pour fonctionner comparée à celle de Google Maps. De plus, elle propose toutes les fonctionnalités que nous avons eu besoin, affichage de la carte, la géolocalisation, faire apparaître des pins et enfin un itinéraire. Il n'y avait donc pas de réel besoin de passer sur celle de google, qui est, il faut le dire plus lourde mais plus performante.

Infrastructure

Serveur en cloud

Comme cité précédemment, nous avons utilisé le service **EC2 d'Amazon Web Services**, qui est un service d'instance virtuelle, accessible en SSH (Ci-dessous, un aperçu de l'interface de gestion d'instances virtuelles), cela nous a permis d'obtenir un serveur sans aucune difficulté, et à potentiel de pouvoir se scaler si augmentation du trafic. Et ceci en ne déboursant rien car l'on ne dépasse pas un certain nombre de requêtes. (Nombre de requêtes limité pour éviter tout abus grâce à des règles de facturation)



The screenshot displays the AWS Management Console interface for an EC2 instance. At the top, there are buttons for 'Lancer une instance', 'Connexion', and 'Actions'. Below these is a search bar and a table of instances. The table has columns for Name, ID d'instance, Type d'instance, Zone de disponibilité, État de l'instance, Contrôles des instances, and Statut des instances. One instance is listed with ID i-005241faa4742ce1c, type t2.micro, in the eu-west-3 zone, and is in a 'running' state. Below the table, there is a section for the selected instance, showing its details: ID d'instance, État de l'instance, Type d'instance, Adresses IP Elastic, Zone de disponibilité, Groupes de sécurité, Événements planifiés, and ID d'AMI.

Name	ID d'instance	Type d'instance	Zone de disponibilité	État de l'instance	Contrôles des instances	Statut des instances
	i-005241faa4742ce1c	t2.micro	eu-west-3	running	2/2 contrôle...	Aucun

Instance: i-005241faa4742ce1c DNS public: ec2-52-47-91-76.eu-west-3.compute.amazonaws.com

Description	Contrôles des instances	Supervision	Balises
ID d'instance	i-005241faa4742ce1c		
État de l'instance	running		
Type d'instance	t2.micro		
Adresses IP Elastic			
Zone de disponibilité	eu-west-3		
Groupes de sécurité	launch-wizard-1. afficher les règles		
Événements planifiés	Aucun événement planifié		
ID d'AMI	ubuntu/images/hvm-ssd/ubuntu-xenial-16.04-amd64-server-20180306 (ami-0e55e373)		

Systèmes Unix

Bien évidemment, comme beaucoup d'environnements de production, nous avons donc dû nous atteler à l'utilisation de systèmes unix sans interface graphique (Ici du **Debian** server) permettant d'optimiser l'utilisation des ressources et l'interopérabilité avec toutes nos technologies choisies précédemment.

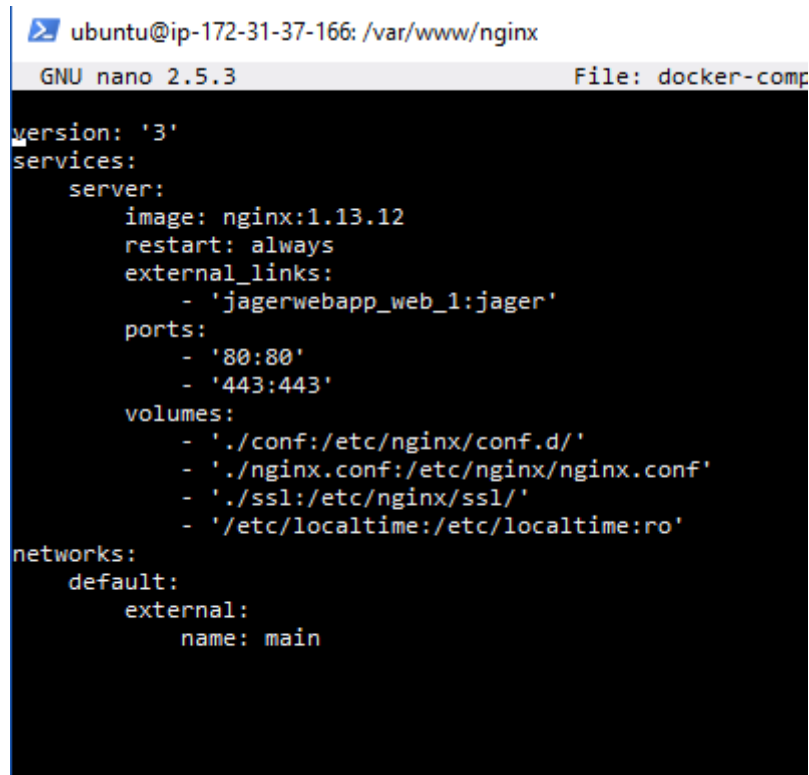
Ce qui nous a demandé d'apprendre ou de réapprendre quelques petites notions que nous avons à ce sujet. Mais à force de l'utiliser, nous avons vraiment réussi à comprendre l'intérêt de celui, nous faisant vraiment gagner du temps sur la mise en place de nos conteneurs d'applications.

```
915 sudo docker-compose down
916 git pull origin master
917 sudo docker-compose up -d web
```

Figure 5 - En 3 lignes le lancement de notre serveur mongo, ainsi que notre serveur Node.js

Conteneurisation

Et pour obtenir un tel niveau de rapidité pour l'exécution de nos machines, il a fallu apprendre à utiliser **Docker**, comprendre le fonctionnement des containers, et saisir tout son intérêt couplé à l'utilisation des fichiers de script et de création de containers d'instance, les **docker-compose** ainsi que les **Dockerfile** qui construisent l'application avec une suite d'instructions écrites dans un format proche de l'UML appris cette année, le YAML.

A screenshot of a terminal window with a dark background. The title bar shows 'ubuntu@ip-172-31-37-166: /var/www/nginx'. The terminal window title is 'GNU nano 2.5.3' and the file being edited is 'File: docker-comp'. The content of the file is a Dockerfile written in YAML format. It defines a container named 'server' using the 'nginx:1.13.12' image, with 'restart: always' and 'external_links' pointing to 'jagerwebapp_web_1:jager'. It maps ports '80:80' and '443:443'. It also defines volumes for configuration files and SSL certificates. Finally, it sets the container to join the 'main' network.

```
version: '3'
services:
  server:
    image: nginx:1.13.12
    restart: always
    external_links:
      - 'jagerwebapp_web_1:jager'
    ports:
      - '80:80'
      - '443:443'
    volumes:
      - './conf:/etc/nginx/conf.d/'
      - './nginx.conf:/etc/nginx/nginx.conf'
      - './ssl:/etc/nginx/ssl/'
      - '/etc/localtime:/etc/localtime:ro'
networks:
  default:
    external:
      name: main
```

Figure 6 - Un exemple de dockerfile pour la construction de notre conteneur NGINX qui gère toutes les requêtes envoyées sur notre machine

Évolution chronologique du projet

Respect des deadlines

Concernant nos deadlines, nous nous étions imposé le cahier des charges suivant :

- Phase d'apprentissage : (Mois de Février)
- Phase de prototypage : (Mois de mars)
- Phase de production : (Mois d'avril)

Bien évidemment, il est difficile de prévoir à 100% comment se passera le déroulement des deadlines, surtout dans un milieu scolaire, mais malgré quelques retards, nous avons réussi à rendre un prototype fonctionnel en temps et en heures.

Nous avons surtout sous-estimé la partie apprentissage, qui nous a vraiment pris beaucoup de temps à tous, et qui nous a mis en retard pour le reste, cependant, nous tendons à de plus en plus respecter des agendas que l'on s'auto-impose, ce qui fut extrêmement positif pour l'organisation en général, ainsi que pour le moral des troupes.

Apparition de nouveaux objectifs

L'intérêt de prototyper rapidement dans le cycle de développement nous a permis de nous rendre compte de maladroites, ou de bonnes idées sur le papier qui se sont avérées mauvaises par la suite. De ce fait, nous avons très souvent reconsidéré les objectifs, la manière de présenter l'information, il nous manque donc encore une certaine méthodologie de management pour gérer un milieu changeant constamment, mais nous lorgnons de plus en plus sur les méthodes agiles, et les cycles de développement continus.

Conclusion du projet

Nous sommes très satisfaits de ce projet car nous y avons appris énormément, en sortant des sentiers battus des cours de l'EFREI, nous nous sommes confrontés à beaucoup d'autres parties dans le développement dont on ne soupçonnait pas forcément l'existence (Charte graphique, business plan ou bien l'infrastructure). Ce qui nous a vraiment grandi et nous a rendu opérationnel pour de nombreux autres projets futurs encore plus ambitieux.