Projet Archi-Web :

Find My night

Spécifications Techniques

Table des matières

[I) Contexte 3](#_Toc419710280)

[II) Choix Techniques 3](#_Toc419710281)

[III) Outils utilisés 4](#_Toc419710282)

[A) Bootstrap CSS 4](#_Toc419710283)

[B) PHP 5](#_Toc419710284)

# Contexte

Dans le cadre d’un projet scolaire, nous devons réaliser un site web exploitant un service web et une base de données open-data.

# Choix Techniques

Pour la réalisation de projet, les choix techniques que nous avons effectués sont les suivants :

Dans un premier temps, l’affichage sera réalisé en HTML & CSS, pour que cette tâche soit moins laborieuse et que le rendu soit plus professionnel nous avons décidé d’utiliser un framework CSS : Bootstrap.

Au niveau du traitement des données, il sera effectué en PHP, c’est un langage relativement standard quand il s’agit de la création de site web et il est assez simple d’utilisation et assez robuste pour la réalisation d’un tel projet.

Pour ce qui est de la communication entre les données et la google maps, le langage à utiliser est imposé par Google, puisque la map ne peut être paramétrée qu’en JavaScript.

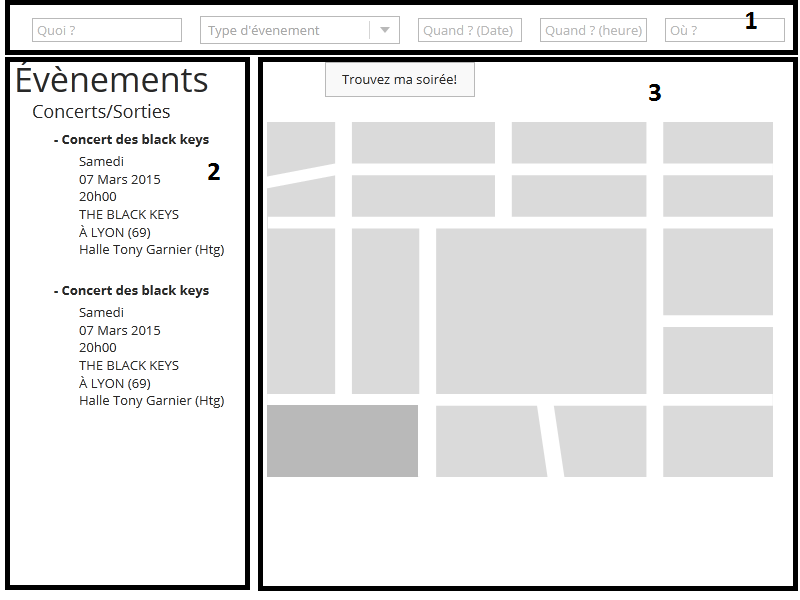
Nous utilisons également un service web REST(representational state transfer) puisque l’utilisation de ce type de web services est bien plus simple à mettre en place que les autres alternatives, notamment les services web SOAP.

Afin de faire fonctionner la map, il fallait également que notre projet soit hébergé et accessible en ligne, nous avons donc choisi de le mettre en ligne très tôt dans le développement et de le tester directement sur le serveur, puisqu’il nous était impossible de tester la map dans d’autre conditions.

# Outils utilisés

## Bootstrap CSS

Bootstrap est un Framework CSS très utilisé pour la création de site web aujourd’hui, principalement parce qu’il est très facile à mettre en place et qu’il permet d’avoir un site avec une apparence visuelle très sobre et propre sans avoir à y dépenser autant de ressources que pour la création entière d’un CSS.



Voici l’architecture utilisée pour la présentation, la zone de header(1) permet la recherche et est de type « navbar » (type bootstrap). Le reste de la page est contenu dans un canvas bootstrap : en effet, bootstrap considère une page comme un cadre de 12 colonnes, il est donc facile de dimensionner les différentes parties de la page en spécifiant le nombre de « colonnes » qu’elles doivent occuper. Dans notre page, la zone affichant le résultat de recherche(2) est d’une largeur de 4 colonnes, tandis que la zone qui contient la map occupe une largeur de 8 colonnes.

Un autre facteur nous ayant poussé à choisir bootstrap CSS est le fait qu’il implémente nativement le responsive design, qui est désormais indispensable à implémenter pour n’importe quel site web compte tenu de la diversité des terminaux pouvant y accéder.

## PHP

Nous avons choisi d’effectuer les traitements en PHP, car c’est un langage simple d’utilisation, puissant, et que la plupart des serveurs web sont capables de l’exécuter. En terme algorithmique, notre site se présente de la manière suivante :

Si l’utilisateur arrive simplement sur le site (affichage par défaut) nous lui affichons les différents concerts qui sont à venir à Lyon. On pourrait très facilement implémenter un système qui détecte la position de l’utilisateur et modifie la requête en fonction de cette dernière.

Si l’utilisateur a effectué une recherche, nous modifions les paramètres de la requête à l’API en fonction des paramètres rentrés par l’utilisateur. Les paramètres communiqués à l’API sont les suivants :

* API Key(App key) (obligatoire pour toutes les requêtes) : xTzS6N2Jxg2PT9ht
* Query (q): Détail de la recherche, mots clefs
* Category(category): Type d’évenement, choisi dans la dropdownlist.
* Lieu(l) : endroit où se déroule l’évenment : de préférence une ville
* Date(date) : Date à laquelle se déroule l’évènement, au format « YYYYMMDD »

Pour la requête par défaut, les paramètres utilisés sont :

« q=music&l=Lyon&date=This%20Week »

D’autres paramètres peuvent être utilisés pour que les résultats de la requête soient plus précis (notamment en termes de tri) mais il s’avère que les résultats fournis ne sont pas très pertinents pour le type de recherche que l’on effectue. Par exemple, si on trie par date de début, on obtient en premier des évènements qui sont réguliers et dont la durée a été spécifiée a plusieurs mois quand ils ont étés entrés dans la base open data, et ce n’est pas forcément ce type d’évènement que l’on recherche pour occuper une soirée.

Pour l’exemple : voici ce à quoi ressemble une requête dans son intégralité (requête par défaut ici):

« [http://api.eventful.com/rest/events/search?app\_key=xTzS6N2Jxg2PT9ht&q=music&l=Lyon&date=This%20Week](http://api.eventful.com/rest/events/search?app_key=xTzS6N2Jxg2PT9ht&q=music&l=Lyon&date=This%20Week )»

Une fois la que la requête a été construite, nous la traitons comme un « simpleXMLElement » ce qui nous permet de parcourir le résultat de manière très simple, on affiche donc les différents évènements retournés par la requête ainsi que leur emplacement sur la carte. Parfois, les lieux où se déroulent les évènements ne sont pas exactement spécifiés et la seule information de localisation que l’on a sur eux est la ville, ils sont donc placés sur la map au centre de la ville.