

2024 - TI 3 – Carte interactive

Objectifs

Création d'une carte interactive (carte utilisant OpenStreetMap) avec une interface d'administration privée accessible à un administrateur qui peut gérer, via un CRUD, les localisations à afficher sur la carte.

Compétences évaluées

Partie 1 - Le 29 et 30 avril 2024

git : création d'un fork, clone de se fork, création de branches, sauvegardes (commit), travail remis via un pull request github

Wamp : création d'un hôte virtuel vers le dossier public du test

PhpMyAdmin : création d'une base de données, et importation des deux fichiers SQL fournis (`utilisateurs.sql` et `localisations.sql`)

MySQL ou MariaDB : exécution de requêtes SELECT, INSERT, UPDATE et DELETE. Avec requêtes préparées si nécessaires et gestion des erreurs

PHP : respect du Modèle Vue Contrôleur (**MVC**) en PHP, mise en oeuvre d'un CRUD, utilisation des sessions pour un accès privé, connexion à la base de données via PDO

HTML / CSS : affichage de la carte et de la liste des localisations, responsive design et formulaire de connexion

JS : récupération des données et intégration dans la carte et la liste

Bootstrap : réalisation de l'interface d'administration

Partie 2 - Pour la présentation (début mai 2024)

FTP : Le travail est mis en ligne sur le serveur du stagiaire

SQL : Le travail est mis en ligne sur un serveur de base de données

Présentation Publique : Présenter son travail de manière cohérente

Marche à suivre pour démarrer le projet

- Créez un fork du dépôt TI3-2024 sur github.com puis clonez-le sur votre machine
- Créez un hôte virtuel dans WAMP sur le dossier **/TI3-2024/public/** nommé **TI3-2024**
- Créez une base de données (en MariaDB ou MySQL) dans phpMyAdmin nommée **`TI3_2024`**
- Créez un fichier **`config.php`** dans le dossier **`/TI3-2024/`** et ajoutez-y les informations de connexion à la base de données dans des **`constantes`** PHP (Il se trouve déjà dans le **`gitignore`** pour ne pas être commité)
- Dupliquez ensuite ce fichier en **`config.php.ini`** que l'on puisse installer votre site sur notre ordinateur pour la correction

Structure des fichiers à utiliser

```
.gitignore
README.md
config.php (à créer)
config.php.ini (à créer)
datas/
    data/utilisateurs.sql
    data/localisations.sql (Fourni par Pierre)
public/
    public/index.php (à modifier)
    public/js/
        public/js/...js (à créer)
    public/css/
        public/css/style.css (à créer)
    public/img/
        public/img/ (à créer)

view/
    view/private/...php (à créer)
    view/public/...php (à créer)
model/
    model/localisationsModel.php (à modifier)
    model/utilisateursModel.php (à modifier)

controller/
    controller/privateController.php (à modifier)
    controller/publicController.php (à modifier)
```

PHP

config.php :

contient les constantes de connexion à la Database "TI3_2024"

config.php.ini :

contient une copie des constantes de connexion à la Database "TI3_2024", il sera envoyé sur github

/public/index.php :

Contrôleur frontal, toutes les requêtes passent par lui.

Il crée/réutilise la **session**

Il charge les dépendances (**config.php**, les modèles)

il crée une instance de **PDO**

Il gère les conditions (liés à la session), et charge les contrôleurs

`**controller/publicController.php**` (si non connecté) ou `**controller/privateController.php**` (si connecté)

model/utilisateursModel.php :

Contient les fonctions de :

- connexion de l'administrateur avec vérification de son username et un password_verify de son mot de passe, avec création de session en cas de réussite.
- déconnexion de l'administrateur et destruction de la session

model/localisationsModel.php :

Contient les fonctions de :

- récupération de toutes les localisations
- chargement d'une localisation par son id
- mise à jour d'une localisation par son id
- création d'une nouvelle localisation
- suppression d'une localisation par son id

/view/... :

Affiche les pages HTML, en lien avec les js, css et images du dossier public

Permet l'affichage de toutes les données

Affiche les erreurs / messages si nécessaire.

controller/publicController.php :

Gère les actions (via des variables GET ou POST) :

- chargement de la vue d'accueil publique (carte + liste)
- chargement des localisations et transformation en JSON pour la carte et la liste
- chargement d'un formulaire de connexion
- connexion effective de l'administrateur (suivi d'une redirection permettant de changer de contrôleur, donc de rôle)

controller/privateController.php :

Gère les actions (via des variables GET ou POST) :

- permet la déconnexion
- chargement des localisations et affichage dans un tableau (accueil de l'admin)
- chargement d'une localisation dans un formulaire pour modification
- modification réelle de la localisation
- chargement d'un formulaire pour insertion
- création réelle d'une nouvelle localisation
- suppression d'une localisation, avec confirmation (page ou popup)

Base de données

Concernant la base de données, elle contient :

- les données sur les utilisateurs ayant accès à l'interface d'administration
- les données de localisation des points à placer sur la carte et dans la liste

Chaque stagiaire a une base de données différente, les données sont fournies séparément.

Attention, les noms et le nombre de colonnes peuvent varier !

Il faut bien vérifier dans le code lors de la récupération des données et dans les requêtes SQL.

Pour la créer et la remplir, il faut :

1. démarrer **phpMyAdmin** dans WAMP
2. créer une **nouvelle base de données vide** appelée "**ti3_2024**"
3. importer le premier script SQL, avec les données utilisateurs, : **utilisateurs.sql**
4. importer le second script SQL, avec les données de localisation, : **localisations.sql**

Interface publique (HTML/CSS)

Cette section détaille les pages, en accès public, vues par n'importe quel visiteur.
Les autres pages, en accès privé, sont décrites dans les sections PHP et Bootstrap.

Page principale : Accueil

Elle est composée de :

- un titre, centré, intitulé “Carte interactive”
- un sous-titre, centré, indiquant le thème de la carte
- un **lien vers la page de connexion** à l'interface d'administration
- une première zone affichant une **carte OpenStreetMap** avec des marqueurs et des popups
- une seconde zone affichant la **liste détaillée** de tous les points représentés sur la carte

Pour la mise en forme de cette page, il faut obligatoirement utiliser du CSS classique (Flexbox, Grid,...) mais pas de Bootstrap !

En principe, la carte va afficher une zone dans laquelle on peut voir tous les marqueurs, c'est-à-dire que le zoom et le centre de la carte sont adaptés pour voir tous les marqueurs dont on a les coordonnées dans la base de données.

Pour des **écrans de largeur supérieure à 640px**, la zone “carte” et la zone “liste” sont affichées l'une à côté de l'autre, dans une répartition 50/50.

Voir exemple ci-dessous:



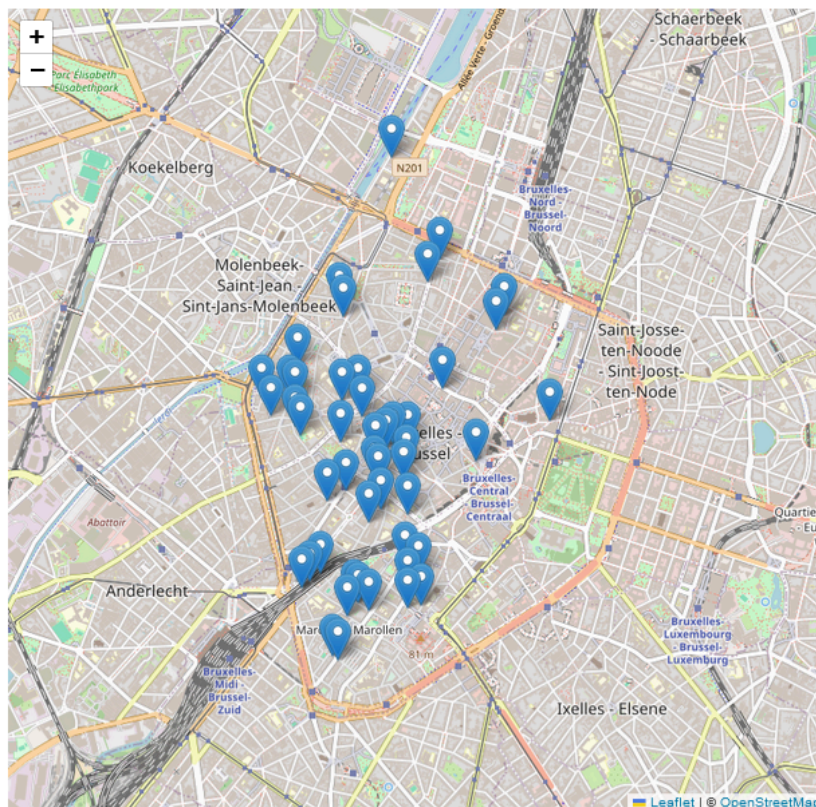
Pour des **écrans de largeur inférieure à 640px**, la zone “carte” et la zone “liste” sont affichées l'une au-dessus de l'autre, et occupant toute la largeur.

Voir exemple ci-dessous:

Carte interactive

Parcours BD à Bruxelles

Connexion à
l'administration



Liste des points

Cliquez sur un élément dans la liste ci-dessous pour le situer sur la carte

- Les Aventures de Tintin | Rue de l'Étuve 37 - 1000 Bruxelles | [Photo](#)
- Le passage | Rue du Marché au Charbon 19 - 1000 Bruxelles | [Photo](#)
- Bob et Bobette | Rue de Laeken 116 - 1000 Bruxelles | [Photo](#)
- Cori le Moussaillon | Rue des Fabriques 21 - 1000 Bruxelles | [Photo](#)
- Astérix le Gaulois | Rue de la Buanderie 33-35 - 1000 Bruxelles | [Photo](#)
- Le jeune Albert | Rue des Alexiens 49 - 1000 Bruxelles | [Photo](#)
- La Vache | Rue du Damier 23 - Sleep Well Youth Hostel - 1000 Bruxelles | [Photo](#)
- Corto Maltese | Quai des Péniches - 1000 Bruxelles | [Photo](#)
- Quick et Flupke | Rue Notre-Seigneur 19 - 1000 Bruxelles | [Photo](#)
- Spirou | Rue Notre Dames des Grâces 3 - 1000 Bruxelles | [Photo](#)
- Froud & Stouf | Bd Maurice Lemonnier 32 - 1000 Bruxelles | [Photo](#)
- In my area | Rue des Chartreux 39 - 1000 Bruxelles | [Photo](#)
- Jardin aux fleurs | Rue du Grand Serment - 1000 Bruxelles | [Photo](#)
- L'élève Ducobu | Rue des Six Jetons 25 - 1000 Bruxelles | [Photo](#)
- La voie lactée | Rue Sallaert - tunnel - 1000 Bruxelles | [Photo](#)
- Le petit gilet | Rue Basse - tunnel - 1000 Bruxelles | [Photo](#)
- Billy the cat | Rue d'Ophem 24 - 1000 Bruxelles | [Photo](#)
- Olivier Rameau | Rue du Chêne 9 - 1000 Bruxelles | [Photo](#)
- La patrouille des Castors | Rue Pieremans 47 - rue Blaes 200 - 1000 Bruxelles | [Photo](#)
- Passe-moi l'ciel | Rue des Minimes 93 - 1000 Bruxelles | [Photo](#)
- Yoko Tsumo | Rue Terre-Neuve 25 - 1000 Bruxelles | [Photo](#)
- Thorgal | Place Anneessens 2A - 1000 Bruxelles | [Photo](#)
- Léonard | Rue des Capucins 23A - 1000 Bruxelles | [Photo](#)
- Les Schtroumpfs | Carrefour de l'Europe 3 - Passage Hilton - 1000 Bruxelles | [Photo](#)
- Kinky & Cosy | Rue de la Gouttière 24 - 1000 Bruxelles | [Photo](#)
- Boule et Bill | Rue du Chevreuil 19 - 1000 Bruxelles | [Photo](#)
- XIII | Rue Philippe de Champagne 31 - 1000 Bruxelles | [Photo](#)
- Benoit Brisefer | Rue Haute 119 - 1000 Bruxelles | [Photo](#)
- Crocodiles | Angle rue du Canon et rue aux Choux - 1000 Bruxelles | [Photo](#)
- Les ancêtres bienveillants | Rue de la Fontaine - tunnel - 1000 Bruxelles | [Photo](#)
- L'ange de Sambre | Rue des Chartreux 21 - 1000 Bruxelles | [Photo](#)
- Les rêves de Nic | Rue des Fabriques 37 - 1000 Bruxelles | [Photo](#)
- Ric Hochet | Rue du Bon Secours 9 - 1000 Bruxelles | [Photo](#)

Javascript : interactions avec la carte

Pour gérer la carte **OpenStreetMap**, on utilise la bibliothèque **Leaflet.js**
(<https://leafletjs.com/examples.html>)

A l'ouverture de la page principale,

- on indique que le conteneur de la carte est placé dans la zone "carte" (balise <div>)
- on définit une coordonnée centrale (latitude,longitude) et un zoom
- on choisit un fond de carte (par exemple: le fond par défaut de OpenStreetMap)
- et on doit afficher la carte

Ensuite, on charge les données des marqueurs et des popups en appelant index.php avec une variable GET, prévue pour interroger la base de données, récupérer toutes les données et les renvoyer sous format JSON.

Puis, on extrait dans ces données JSON, pour chacun des points, les coordonnées géographiques permettant de placer un marqueur sur la carte.

Les autres informations (nom, adresse, URL, images,...) sont utilisées pour associer un popup qui s'ouvrira quand on cliquera sur le marqueur correspondant.

Voir plus de détails sur la fiche concernant le thème

Connaissant la position de tous les marqueurs à afficher, on est capable de déterminer la zone limite qui permet de les afficher tous au mieux sur la carte et d'adapter le zoom en conséquence (méthode fitBounds).

En utilisant les mêmes données récupérées en JSON, on peut créer une liste reprenant tous les points et les afficher sous forme de texte.

Voir plus de détails sur la fiche concernant le thème

Enfin, si on clique sur un élément de la liste, on doit montrer sur la carte le marqueur correspondant et ouvrir son popup.

Si on modifie des informations dans la base de données via l'interface d'administration, les modifications doivent être répercutées dans la carte et dans la liste.

Bootstrap :

Créez une interface d'administration pour une application web. L'interface doit inclure les éléments suivants :

1. Page de Connexion :

- Une page permettant aux utilisateurs de se connecter à l'application.
- Utilisez un formulaire pour saisir les informations d'identification (nom d'utilisateur et mot de passe).
- Ajoutez un bouton de connexion.

2. Page d'Accueil (Tableau) :

- Une page affichant la liste de tous les points.
- Utilisez un tableau pour présenter les données.
- Incluez des colonnes pour le nom du point, sa description et d'autres informations pertinentes.

3. Page de Création d'un Nouveau Point :

- Une page avec un formulaire permettant de créer un nouveau point.
- Le formulaire doit inclure des champs pour le nom, la description et d'autres détails du point.

4. Page de Modification d'un Point Existant :

- Une page avec un formulaire permettant de modifier un point existant.
- Affichez les informations actuelles du point dans le formulaire.
- Permettez à l'utilisateur de mettre à jour les détails du point.

5. Page de Confirmation pour Supprimer un Point :

- Une page de confirmation s'affiche lorsque l'utilisateur souhaite supprimer un point.
- Affichez le nom du point et demandez à l'utilisateur de confirmer la suppression.

Consignes

- Utilisez Bootstrap pour la mise en page et le style de l'interface.
- Assurez-vous que l'interface est réactive et s'adapte bien aux différentes tailles d'écran.

N'hésitez pas à personnaliser davantage l'interface en ajoutant des fonctionnalités supplémentaires ou en modifiant le design selon vos besoins. Bonne création !