

# Projet M2 MIAGE

Développement d'une application Web cartographique  
avec MongoDB, Node.js et Leaflet

## Objectif général

L'objectif de ce projet est de développer une application Web cartographique permettant de visualiser, gérer et analyser des données géolocalisées stockées dans une base de données **MongoDB**, exposées via un backend **Node.js**, et affichées sur une carte interactive **Leaflet**.

Le projet doit être réalisé en binôme.

## 1. Contexte

Une collectivité territoriale souhaite une application simple permettant de :

- consulter des points d'intérêt géolocalisés ;
- ajouter de nouveaux points par interaction directe avec la carte ;
- éditer et supprimer des points existants ;
- effectuer des analyses statistiques simples ;
- visualiser ces données sur une carte Leaflet.

## 2. Fonctionnalités attendues

### 2.1. Backend Node.js / Express

Le backend doit exposer une API REST comprenant au minimum :

- GET `/api/items` — renvoie tous les points ;
- GET `/api/items/:id` — renvoie un point spécifique ;
- POST `/api/items` — ajoute un nouveau point ;
- PUT `/api/items/:id` — modifie un point ;

— DELETE /api/items/:id — supprime un point.

Exigences complémentaires :

- validation minimale des données ;
- gestion correcte des erreurs HTTP ;
- utilisation de express.json() ;

## 2.2. Base de données MongoDB

Chaque point doit être stocké sous la forme :

```
{  
    "address": {  
        "coord": {  
            "type": "Point",  
            "coordinates": [  
                -73.856077,  
                40.848447  
            ]  
        },  
    },  
    "borough": "Bronx",  
    "cuisine": "Bakery",  
    "grades": [  
        {  
            "date": {  
                "$date": "2014-03-03T00:00:00.000Z"  
            },  
            "grade": "A",  
            "score": 2  
        }  
    ],  
    "name": "Morris Park Bake Shop",  
    "restaurant_id": "30075445"  
}
```

## 2.3. Interface Web Leaflet

Fonctionnalités obligatoires :

- affichage des points sur une carte centrée sur une zone pertinente.

- utilisation de `circleMarker` avec style dynamique (couleur, taille, etc.).
- affichage de popups.
- bouton permettant d'ajouter un nouveau point via un formulaire.
- sélection d'une position sur la carte par clic.
- édition et suppression des points existants.
- filtres visuels (par cuisine, quartier, score, etc.).

Bonus possibles :

- clustering ;
- heatmap ;
- légende dynamique.

## 2.4. Analyses statistiques

Le projet doit comporter au moins **trois analyses**, par exemple :

- score moyen par type de cuisine.
- distribution des scores par type de cuisine.
- recherche de points proches (requête géospatiale).

Les résultats peuvent être affichés :

- sur la carte ;
- dans un tableau ;
- sous forme de graphiques (optionnel).

## 3. Livrables

- **Code complet** (frontend + backend + base MongoDB) ;
- **README** expliquant l'installation et l'architecture ;

## 4. Critères d'évaluation

### Technique (50%)

Qualité du code, structure, bonne utilisation de MongoDB, API REST, Leaflet.

### Interface et ergonomie (20%)

Lisibilité de la carte, interactions intuitives, cohérence visuelle.

## **Analyse des données (20%)**

Pertinence, rigueur, traitement géospatial.

## **Documentation (10%)**

Clarté du README