

# Rapport d'Avancement du Projet Analyseur de Sentiments de Tweets

BOUAZZAOUI Ibtissam

18 avril 2025

## 1 Introduction

Ce projet consiste à développer une application web permettant d'analyser les sentiments exprimés dans des tweets en temps réel. L'objectif est de classer chaque tweet selon trois catégories : positif, négatif ou neutre, et d'afficher les résultats sur une interface React.

## 2 Fonctionnalités Principales

- Recherche de tweets en temps réel à l'aide de l'API Twitter.
- Analyse de sentiment avec la bibliothèque **Sentiment**.
- Sauvegarde des résultats dans MongoDB pour réutilisation.
- Interface utilisateur dynamique développée en React.

## 3 Environnement Technique

### 3.1 Frontend

L'interface a été réalisée avec **ReactJS**, et initialisée avec :

```
npx create-react-app sentiment-analyzer  
cd sentiment-analyzer  
npm start
```

### 3.2 Backend

Le serveur backend est basé sur **Express.js**, avec la configuration suivante :

```
npm install express cors dotenv twitter-api-v2 sentiment mongoose
```

Il se connecte à MongoDB et interagit avec l'API Twitter pour analyser et stocker les tweets.

## 4 Configuration de l'API Twitter

Les identifiants nécessaires ont été ajoutés au fichier `.env` :

```
TWITTER_API_KEY=...  
TWITTER_API_SECRET=...  
TWITTER_ACCESS_TOKEN=...  
TWITTER_ACCESS_TOKEN_SECRET=...  
MONGODB_URI=...
```

## 5 Récupération et Analyse des Tweets

Lorsqu'un utilisateur effectue une recherche, le backend :

1. Vérifie si des tweets correspondants sont déjà en cache dans MongoDB.
2. Sinon, interroge l'API Twitter.
3. Analyse les sentiments avec la bibliothèque `sentiment`.
4. Sauvegarde les tweets analysés.

Extrait de la logique côté backend :

```
const cachedTweets = await Tweet.find({  
  text: { $regex: query, $options: 'i' }  
});  
if (cachedTweets.length > 0) {  
  return res.json(cachedTweets.map(tweet => ({  
    tweetId: tweet.tweetId,  
    text: tweet.text,  
    sentiment: tweet.sentiment  
  })));  
}  
// Sinon, récupération depuis l'API Twitter
```

## 6 Frontend React

L'interface React permet :

- Une barre de recherche stylisée pour entrer un mot-clé.
- Un affichage dynamique des résultats.
- Des cartes colorées selon le sentiment (vert, rouge, gris).

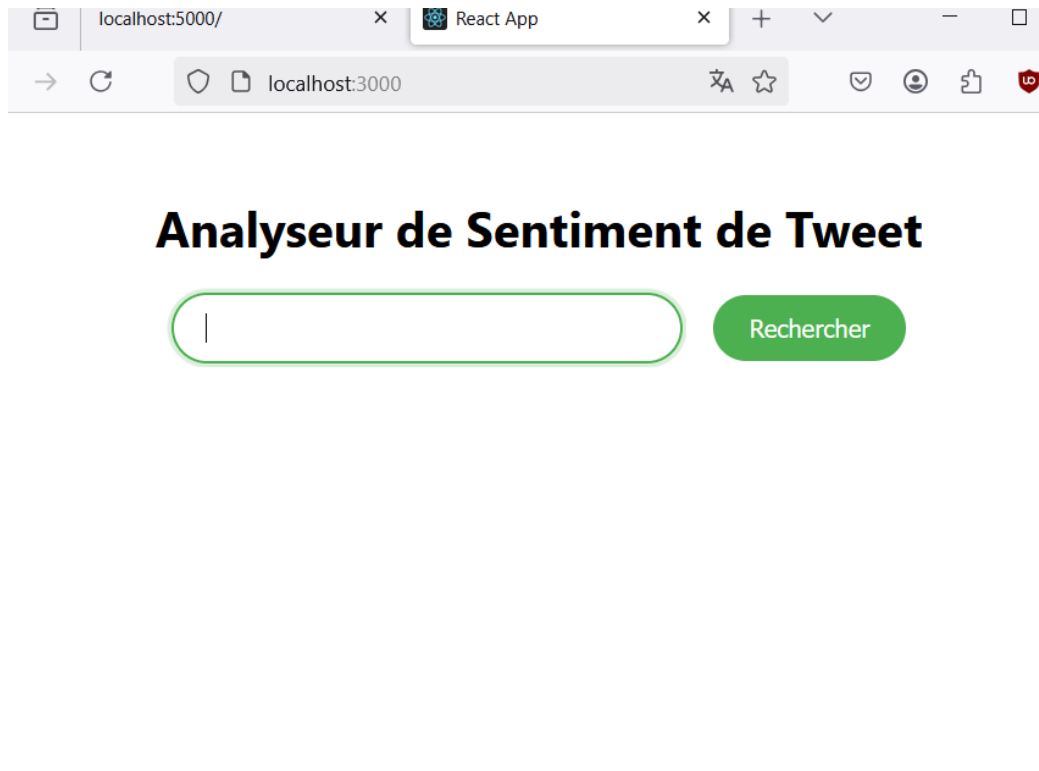


FIGURE 1 – Interface utilisateur React

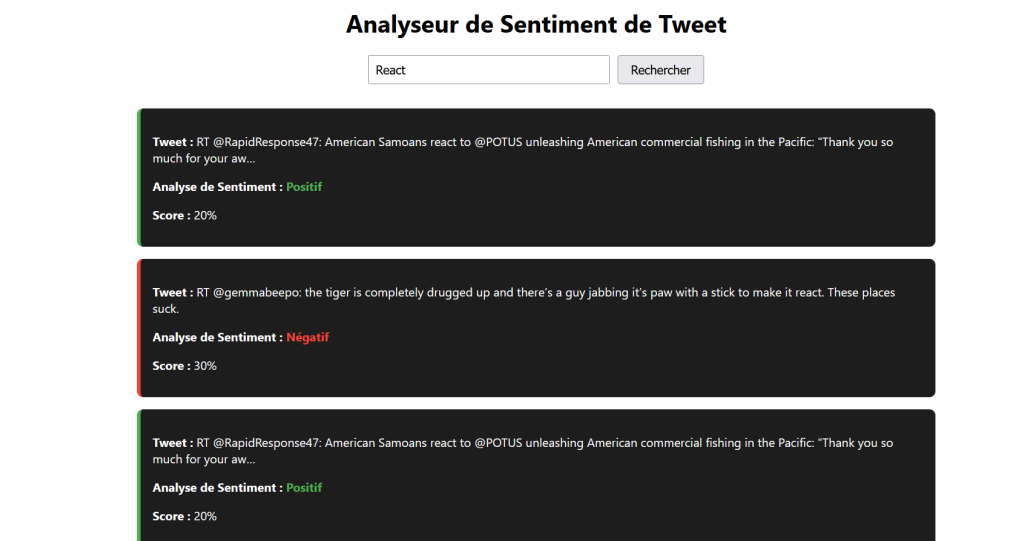


FIGURE 2 – Analyseur de sentiments

## 7 Backend Express

Le serveur est lancé via :

```
node server.js
```

Il utilise les routes suivantes :

- / : test du serveur
- /search/:query : recherche et analyse des tweets

## 8 Problèmes rencontrés

- **Limites de requêtes API (erreur 429) :** trop de requêtes en peu de temps provoquent une restriction temporaire.

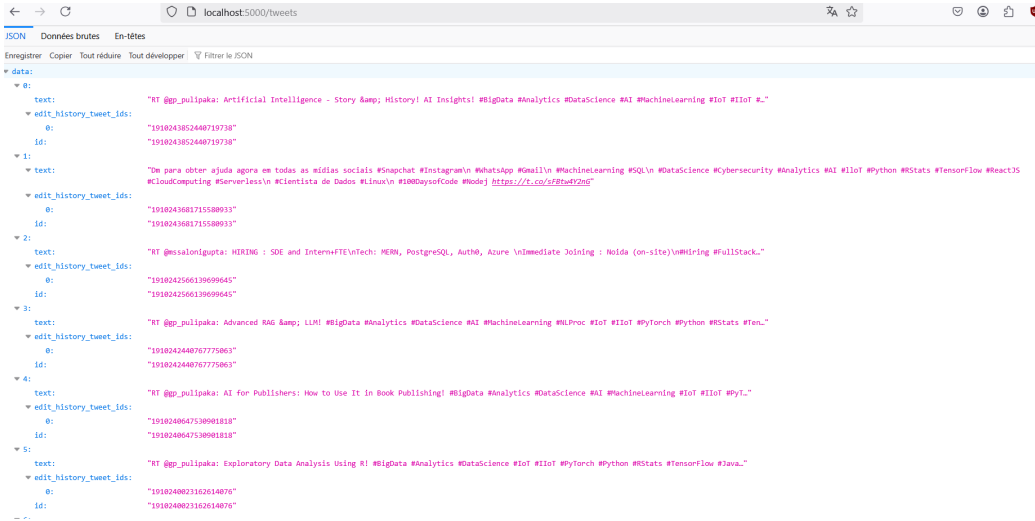


FIGURE 3 – Affichage JSON côté API

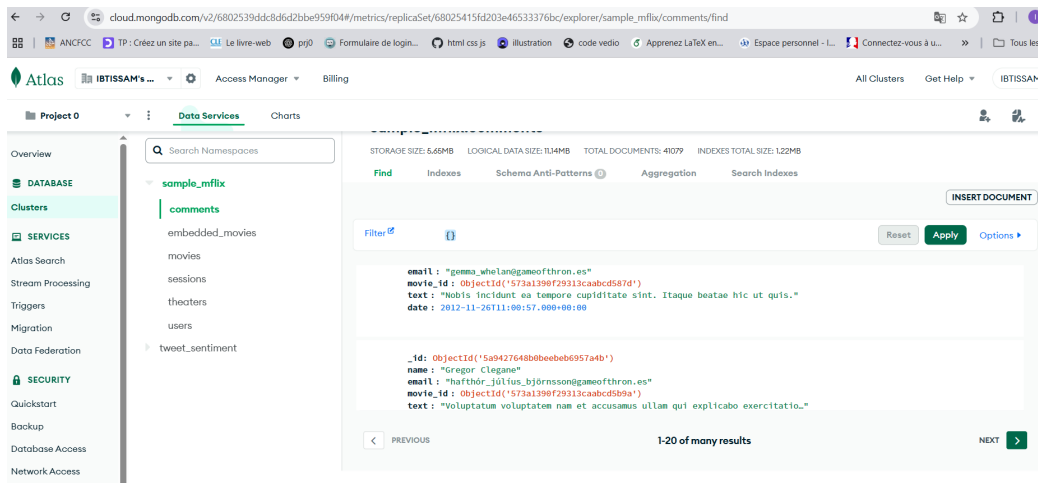


FIGURE 4 – enregistrer les données sur MongoDB