



PROJET ETL **PRESENTATION**

Pipeline complet de traitement de données textuelles
Extract Transform Load

wail brimesse

1

PRÉSENTATION DU PROJET

2

GESTION DE PROJET

3

DÉTAILS TECHNIQUES

2

DIFFICULTÉS RENCONTRÉES

3

AMÉLIORATION FUTURE

SOMMAIRE



PRÉSENTATION DU PROJET

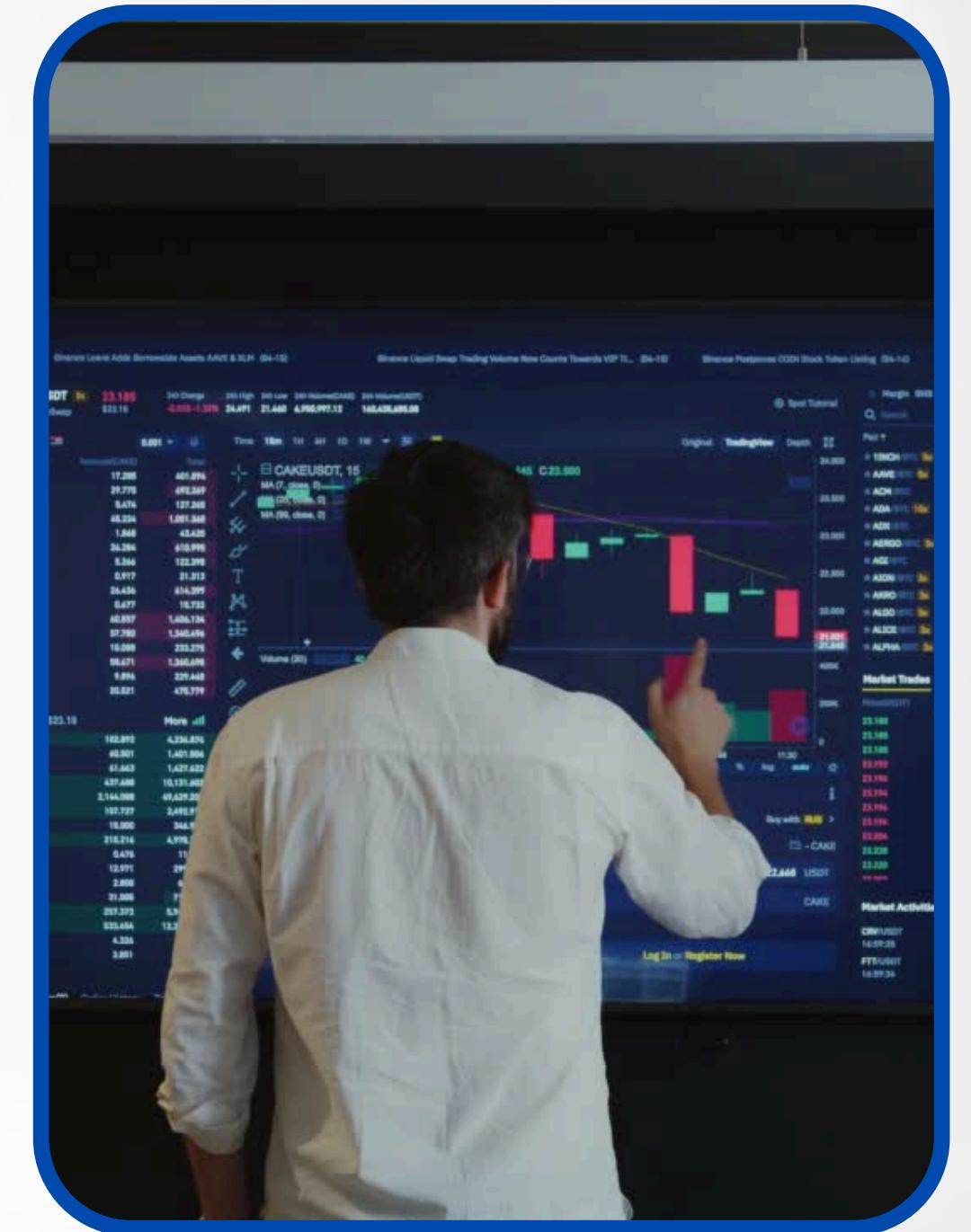
Objectif : créer un pipeline complet ETL de traitement de texte.

- Extraction : scraping d'articles sur 8 sites.
- Transformation : analyse NLP (toxicité).
- Chargement : MongoDB + visualisations + Docker.

```
        else, message: 'Could not register user, user with same email already exists' );
    else, message: 'Could not register user, Error: ' + err.message);
}

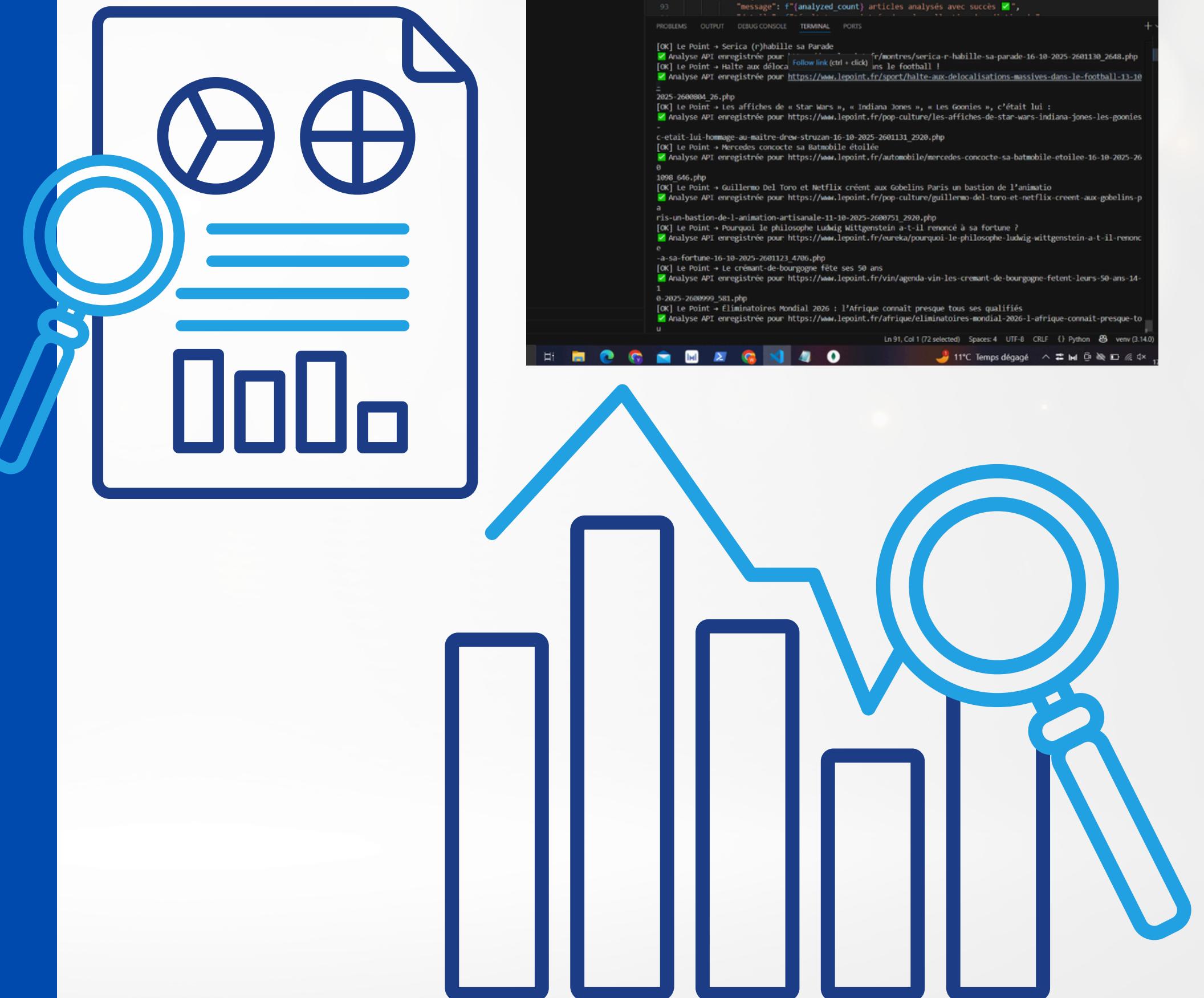
const sendEmail = (email, subject, html) => {
    const options = {
        method: 'POST',
        url: `https://api.sendgrid.com/v3/mail/send`,
        headers: {
            'Authorization': `Bearer ${process.env.SENDGRID_API_KEY}`,
            'Content-Type': 'application/json'
        },
        body: {
            'personalizations': [
                {
                    'to': [
                        { 'email': email }
                    ],
                    'subject': subject
                }
            ],
            'from': {
                'email': 'no-reply@baseurl.co.za'
            },
            'html': html
        }
    };
    request(options, (err, res, data) => {
        if (err) {
            console.error('Error sending email:', err);
            return;
        }
        console.log(`Email sent to ${email} with status ${res.statusCode}`);
    });
}

const generateConfirmationEmail = (user) => {
    const subject = 'Confirm Your Account';
    const html = `
        <p>Hello ${user.name},</p>
        <p>Please click the button below to confirm your account.</p>
        <p><a href="${host.baseurl}/api/accounts/confirm?token=${user.confirmation_token}">Confirm</a></p>
    `;
    sendEmail(user.email, subject, html);
}
```



PARTIE 1 — COLLECTE ET STOCKAGE

Scraping via BeautifulSoup et Selenium.
Chaque site a sa propre fonction dédiée
(balises différentes).
Nettoyage des doublons et insertion dans
MongoDB.
7 sites sur 8 scrappés avec succès.



PARTIE 2

API NLP ET PRÉDICTION

```
[OK] Le Point → Serica (r)habille sa Parade
✓ Analyse API enregistrée pour https://www.lepoint.fr/montres/serica-r-habille-sa-parade-16-10-2025-2601130_2648.php
[OK] Le Point → Halte aux délocalisations massives dans le football !
✓ Analyse API enregistrée pour https://www.lepoint.fr/sport/halte-aux-delocalisations-massives-dans-le-football-13-10-2025-2600804_26.php
[OK] Le Point → Les affiches de « Star Wars », « Indiana Jones », « Les Goonies », c'était lui :
✓ Analyse API enregistrée pour https://www.lepoint.fr/pop-culture/les-affiches-de-star-wars-indiana-jones-les-goonies-c-etais-lui-hommage-au-maitre-drew-struzan-16-10-2025-2601131_2920.php
[OK] Le Point → Mercedes concoste sa Batmobile étoilée
✓ Analyse API enregistrée pour https://www.lepoint.fr/automobile/mercedes-concopte-sa-batmobile-etoilee-16-10-2025-2601132_646.php
[OK] Le Point → Guillermo Del Toro et Netflix créent aux Gobelins Paris un bastion de l'animation
✓ Analyse API enregistrée pour https://www.lepoint.fr/pop-culture/guillermo-del-toro-et-netflix-creent-aux-gobelins-puis-un-bastion-de-l-animation-artisanale-11-10-2025-2600751_2920.php
[OK] Le Point → Pourquoi le philosophe Ludwig Wittgenstein a-t-il renoncé à sa fortune ?
✓ Analyse API en registrée pour https://www.lepoint.fr/eureka/pourquoi-le-philosophe-ludwig-wittgenstein-a-t-il-renonce-a-sa-fortune-16-10-2025-2601123_4706.php
[OK] Le Point → Le crémant-de-bourgogne
✓ Analyse API enregistrée pour https://www.lepoint.fr/boisson/le-cremant-de-bourgogne-16-10-2025-2600999_581.php
[OK] Le Point → Éliminatoires Mondial
✓ Analyse API enregistrée pour https://www.lepoint.fr/sport/eliminatoires-mondial-16-10-2025-2601133_1050.php
```

The screenshot shows the MongoDB Compass interface connected to a database named 'articles_db'. The 'predictions' collection is selected, displaying four documents. Each document represents a prediction for a news article, containing fields such as _id, text, label, score, polarity, timestamp, and url.

_id	text	label	score	polarity	timestamp	url
ObjectId('68f0f2fbf8263aa3c1ef36')	"Je déteste ce genre de comportements"	"neutre"	0	0	2025-10-16T13:28:27.924+00:00	
ObjectId('68f0f9bfd41c950050aec4f')	"Je déteste ce genre de comportements"	"toxique"	0.873	0	2025-10-16T13:57:19.619+00:00	
ObjectId('68f1195524d7d9c86b5156db')	"I hate this product so much"	"toxique"	0.9997256398200989	0	2025-10-16T16:12:05.518+00:00	"https://example.com/"
ObjectId('68f1195824d7d9c86b5156dc')	"I hate this product so much"	"toxique"	0.9997256398200989	0	2025-10-16T16:12:08.758+00:00	"https://example.com/"

FastAPI + modèle sentiment-analysis
(Hugging Face).

Endpoint /predict : texte → label (toxique / non toxique).

Stockage dans MongoDB avec score et date.

Difficulté : modèle en anglais

PARTIE 3

ANALYSE ET VISUALISATION

Objectif : analyser la toxicité par site.

Outils : Pandas, Matplotlib, Seaborn.

Graphiques : camembert + barres comparatives.

Résultats enregistrés dans MongoDB
(toxicity_stats).

```

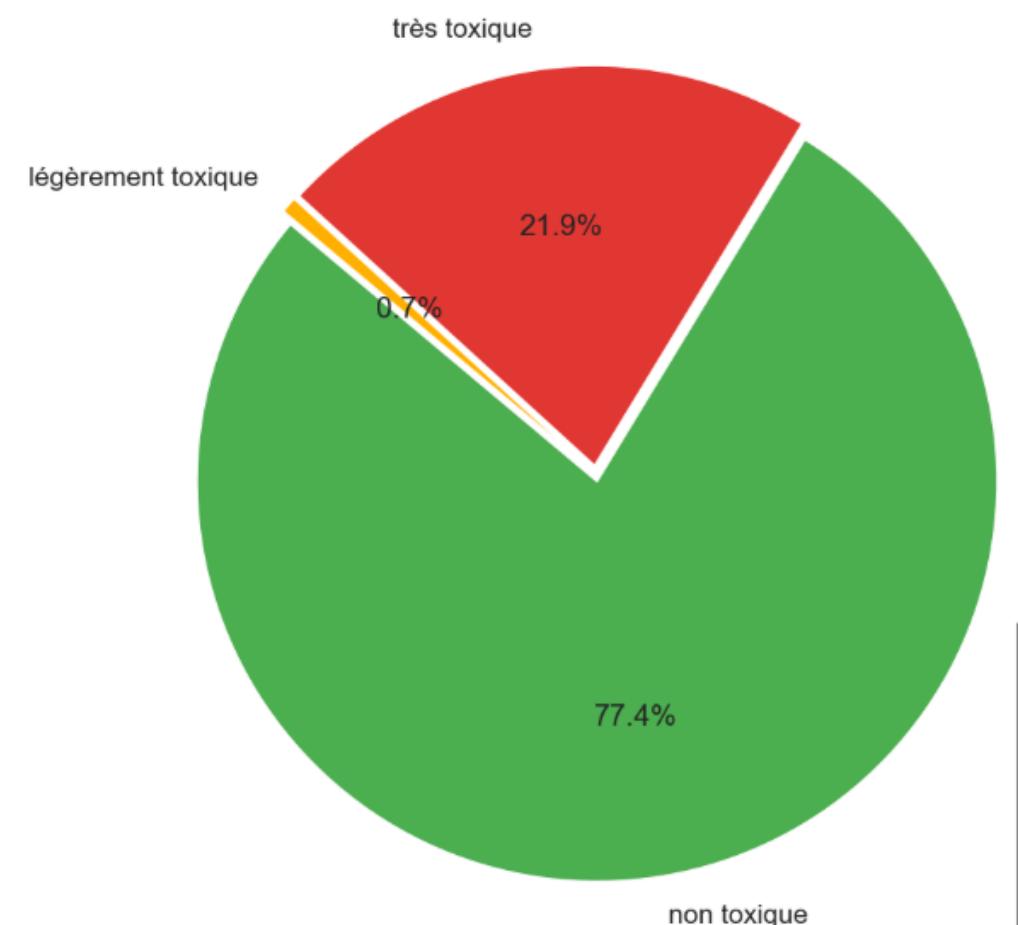
Windows PowerShell      Windows PowerShell      + 
mongo_express | basicAuth credentials are "admin:pass", it is recommended you change this in you
toxicity_api   | /usr/local/lib/python3.11/site-packages/transformers/utils/hub.py:127: FutureWar
deprecated and will be removed in v5 of Transformers. Use 'HF_HOME' instead.
toxicity_api   |     warnings.warn(
toxicity_api   |     Chargement du modèle anglais (sentiment-analysis)...
toxicity_api   |     No model was supplied, defaulted to distilbert/distilbert-base-uncased-finetuned
(tohttps://huggingface.co/distilbert/distilbert-base-uncased-finetuned-sst-2-english).
toxicity_api   |     Using a pipeline without specifying a model name and revision in production is n
toxicity_api   |     INFO:     Started server process [7]
toxicity_api   |     INFO:     Waiting for application startup.
toxicity_api   |     INFO:     Application startup complete.
toxicity_api   |     INFO:     Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
mongodb       | {"t":{"$date":"2025-10-17T00:09:52.812+00:00"},"s":"I", "c":"NETWORK", "id":22
ction accepted","attr":{"remote":"172.18.0.3:34262","uuid":"3e504bdb-0479-400d-a875-566e89f08a96"
5}}
mongodb       | {"t":{"$date":"2025-10-17T00:09:52.814+00:00"},"s":"I", "c":"NETWORK", "id":51
etadata","attr":{"remote":"172.18.0.3:34262","client":"conn6","negotiatedCompressors":[],"doc":{"
4.13.0"},"os":{"type":"Linux","name":"linux","architecture":"x64","version":"E
v18.20.3, LE (unified)|Node.js v18.20.3, LE (unified)"}}
mongodb       | {"t":{"$date":"2025-10-17T00:10:40.658+00:00"},"s":"I", "c":"
iredTiger message","attr":{"message":{"ts_sec":1760659840,"ts_usec":657972,"th
ckpoint","category":WT_VERB_CHECKPOINT_PROGRESS,"category_id":6,"verbose_leve
nt snapshot min: 3, snapshot max: 3 snapshot count: 0, oldest timestamp: (0, 0
9"}}

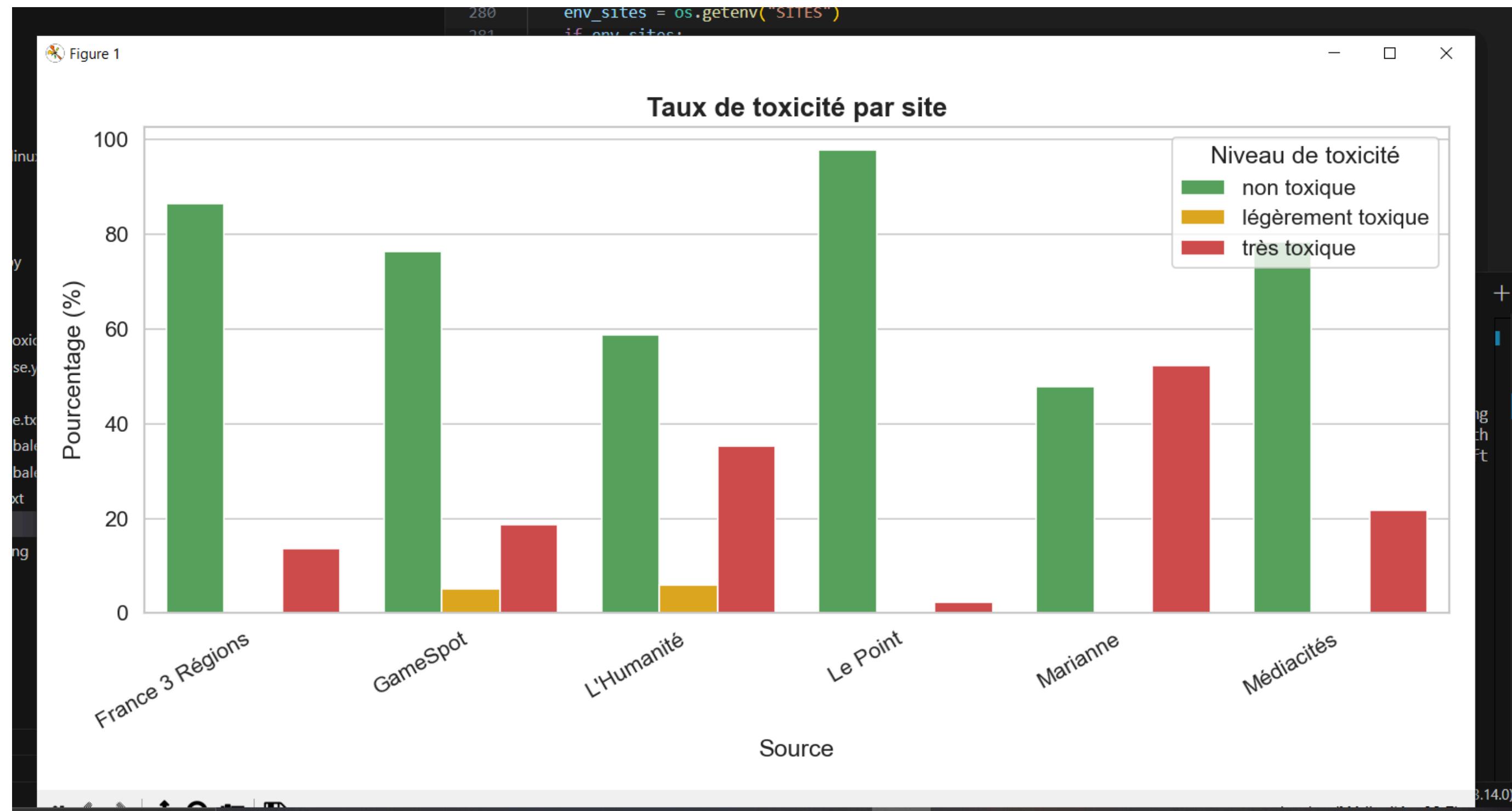

mongo_express | GET / 200 51.724 ms - 9260
mongo_express | GET /public/css/bootstrap-theme.min.css 200 14.609 ms - 23411
mongo_express | GET /public/css/style.css 200 11.405 ms - 1883
mongo_express | GET /public/img/mongo-express-logo.png 200 10.882 ms - 17847
mongo_express | GET /public/css/bootstrap.min.css 200 13.624 ms - 121457
mongo_express | GET /public/vendor-93f5fc3ae20e0dfd68cb.min.js 200 11.998 ms -
mongo_express | GET /public/index-56afe067afbbbde795be.min.js 200 8.394 ms -
mongo_express | GET /public/img/gears.gif 200 4.477 ms - 50281
mongo_express | GET /public/fonts/glyphicons-halflings-regular.woff2 200 1.52
mongo_express | GET /db/admin/ 200 33.373 ms - 8586
mongo_express | GET /public/css/bootstrap.min.css 304 2.300 ms - -


```

Figure 1

Répartition globale de la toxicité des articles





PARTIE 4

DÉPLOIEMENT AVEC

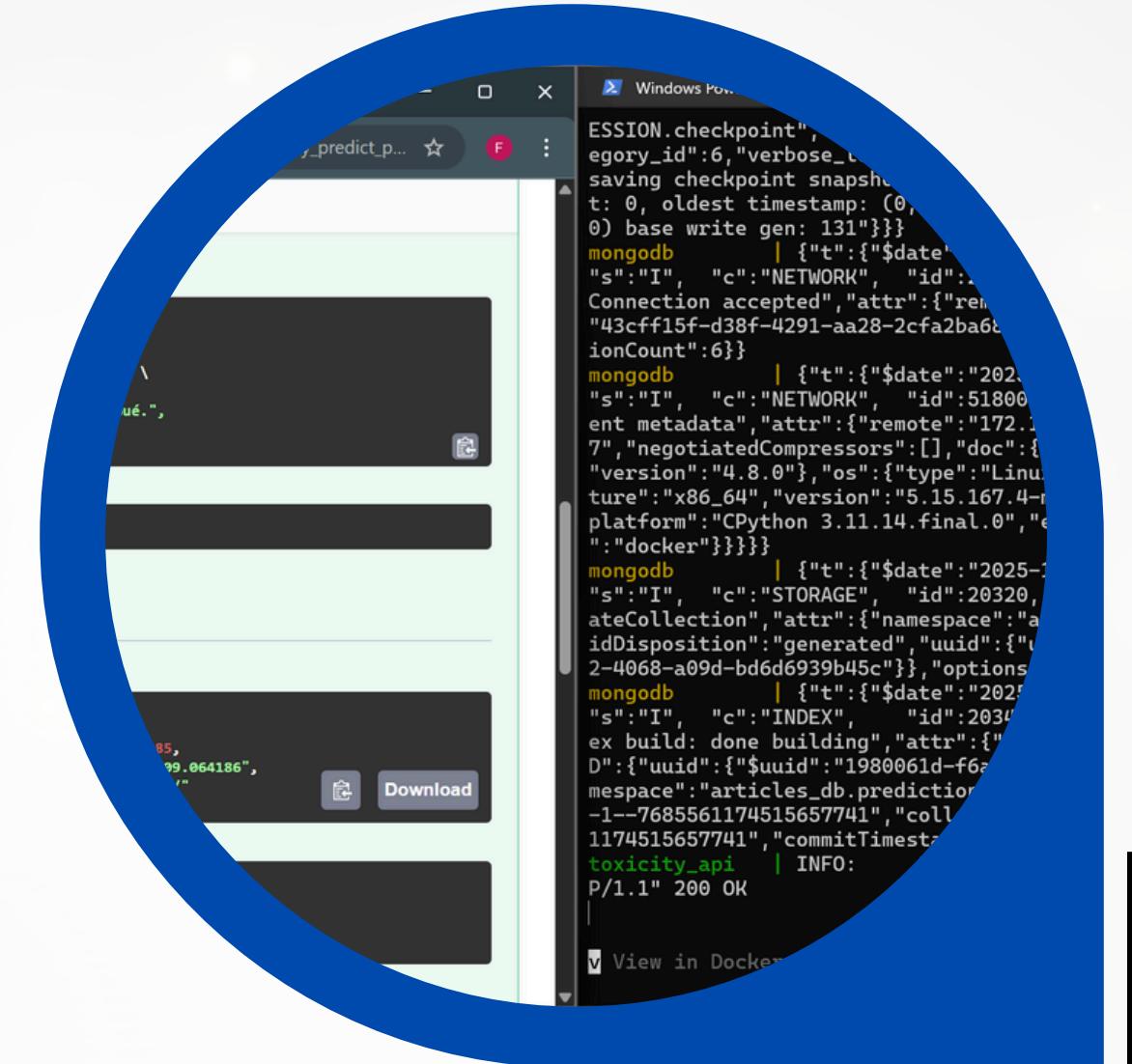
DOCKER

Conteneurisation de FastAPI, MongoDB et Mongo Express.

Fichiers : Dockerfile + docker-compose.yml.

Résolution des erreurs WSL et d'espace disque.

Résultat : pipeline complet exécutable en une commande.



The screenshot shows a browser window displaying API documentation for a toxicity prediction endpoint, followed by two terminal windows showing application logs.

API Documentation (localhost:8000/docs#default/predict_toxicity_predict_p...)

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:8000/predict' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "text": "Ce film est nul et mal jou\u00e9.",
    "url": "https://example.com/"
}'
```

Request URL

```
http://localhost:8000/predict
```

Server response

Code Details

200

Response body

```
{
  "label": "non toxique",
  "score": 0.7666643261909485,
  "date": "2025-10-17T00:25:09.064186",
  "url": "https://example.com/"
}
```

Response headers

```
content-length: 115
content-type: application/json
date: Fri, 17 Oct 2025 00:25:08 GMT
server: uvicorn
```

Logs (Windows PowerShell)

```
mongo_express | basicAuth credentials are "admin:pass", it is recommended you change this in your config.js!
toxicity_api | /usr/local/lib/python3.11/site-packages/transformers/utils/hub.py:127: FutureWarning: Using 'TRANSFORMERS_CACHE' is deprecated and will be removed in v5 of Transformers. Use 'HF_HOME' instead.
toxicity_api |     warnings.warn(
toxicity_api |     Chargement du mod\u00e8le anglais (sentiment-analysis)...
toxicity_api |     No model was supplied, defaulted to distilbert/distilbert-base-uncased-finetuned-sst-2-english and revision af0f99b
(https://huggingface.co/distilbert/distilbert-base-uncased-finetuned-sst-2-english).
toxicity_api |     Using a pipeline without specifying a model name and revision in production is not recommended.
toxicity_api |     INFO:     Started server process [7]
toxicity_api |     INFO:     Waiting for application startup.
toxicity_api |     INFO:     Application startup complete.
toxicity_api |     INFO:     Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
mongodb | {"t": {"$date": "2025-10-17T00:09:52.812+00:00"}, "s": "I", "c": "NETWORK", "id": 22943, "ctx": "listener", "msg": "Connection accepted", "attr": {"remote": "172.18.0.3:34262", "uuid": "3e504bdb-0479-400d-a875-566e89f08a96", "connectionId": 6, "connectionCount": 5}}
mongodb | {"t": {"$date": "2025-10-17T00:09:52.814+00:00"}, "s": "I", "c": "NETWORK", "id": 51800, "ctx": "conn6", "msg": "client metadata", "attr": {"remote": "172.18.0.3:34262", "client": "conn6", "negotiatedCompressors": [], "doc": {"driver": {"name": "nodejs", "version": "4.13.0"}, "os": {"type": "Linux", "name": "linux", "architecture": "x64", "version": "5.15.167.4-microsoft-standard-WSL2"}, "platform": "Node.js v18.20.3, LE (unified)|Node.js v18.20.3, LE (unified)"}}
mongodb | {"t": {"$date": "2025-10-17T00:10:40.658+00:00"}, "s": "I", "c": "WTCHKPT", "id": 22430, "ctx": "Checkpointer", "msg": "WrittenTiger message", "attr": {"message": {"ts_sec": 1760659840, "ts_usec": 657972, "thread": "1:0x7fb048458640", "session_name": "WT_SESSION.checkpoint", "category": "WT_VERB_CHECKPOINT_PROGRESS"}, "category_id": 6, "verbose_level": "DEBUG", "verbose_level_id": 1, "msg": "saving checkpoint snapshot min: 3, snapshot max: 3 snapshot count: 0, oldest timestamp: (0, 0), meta checkpoint timestamp: (0, 0) base write gen: 89"}}
mongo_express | GET / 200 51.724 ms - 9260
mongo_express | GET /public/css/bootstrap-theme.min.css 200 14.609 ms - 23411
mongo_express | GET /public/css/style.css 200 11.405 ms - 1883
mongo_express | GET /public/img/mongo-express-logo.png 200 10.882 ms - 17847
mongo_express | GET /public/css/bootstrap.min.css 200 13.624 ms - 121457
mongo_express | GET /public/vendor-93f5fc3ae20e0dfdf68cb.min.js 200 11.998 ms - 131153
mongo_express | GET /public/index-56afe067afbbde795be.min.js 200 8.394 ms - 936
mongo_express | GET /public/img/gears.gif 200 4.477 ms - 50281
mongo_express | GET /public/fonts/glyphicons-halflings-regular.woff2 200 1.525 ms - 18028
mongo_express | GET /db/admin/ 200 33.373 ms - 8586
mongo_express | GET /public/css/bootstrap.min.css 304 2.300 ms - -
```

Logs (Windows PowerShell)

```
SESSION.checkpoint", "category": "WT_VERB_CHECKPOINT_PROGRESS", "category_id": 6, "verbose_level": "DEBUG", "verbose_level_id": 1, "msg": "saving checkpoint snapshot min: 5, snapshot max: 5 snapshot count: 0, oldest timestamp: (0, 0), meta checkpoint timestamp: (0, 0) base write gen: 131"}}
mongodb | {"t": {"$date": "2025-10-17T00:25:09.065+00:00"}, "s": "I", "c": "NETWORK", "id": 22943, "ctx": "listener", "msg": "Connection accepted", "attr": {"remote": "172.18.0.3:43306", "uuid": "43cff15f-d38f-4291-aa28-2cfa2ba68b87", "connectionId": 7, "connectionCount": 6}}
mongodb | {"t": {"$date": "2025-10-17T00:25:09.065+00:00"}, "s": "I", "c": "NETWORK", "id": 51800, "ctx": "conn7", "msg": "client metadata", "attr": {"remote": "172.18.0.3:43306", "client": "7", "negotiatedCompressors": [], "doc": {"driver": {"name": "Python", "version": "4.8.0"}, "os": {"type": "Linux", "name": "Ubuntu", "architecture": "x86_64", "version": "5.15.167.4-microsoft-standard"}, "platform": "CPython 3.11.14.final.0", "env": {"container": "docker"}}, "options": {}}}
mongodb | {"t": {"$date": "2025-10-17T00:25:09.065+00:00"}, "s": "I", "c": "STORAGE", "id": 20320, "ctx": "conn7", "msg": "createCollection", "attr": {"namespace": "articles_db.predictions", "idDisposition": "generated", "uuid": {"uuid": "2-4068-a09d-bd6d6939b45c"}}, "options": {}}
mongodb | {"t": {"$date": "2025-10-17T00:25:09.065+00:00"}, "s": "I", "c": "INDEX", "id": 20345, "ctx": "conn7", "msg": "ex build: done building", "attr": {"buildUUID": null, "index": {"name": "articles_db.predictions._id_"}, "indexId": "1-7685561174515657741", "collectionIdent": "collection1174515657741", "commitTimestamp": null}}
toxicity_api | INFO: 172.18.0.1:37088 - "POST P/1.1" 200 OK
```

PI de détection de toxicité - S | API de détection de toxicité - S | Home - Mongo Express

localhost:8081

Mongo Express

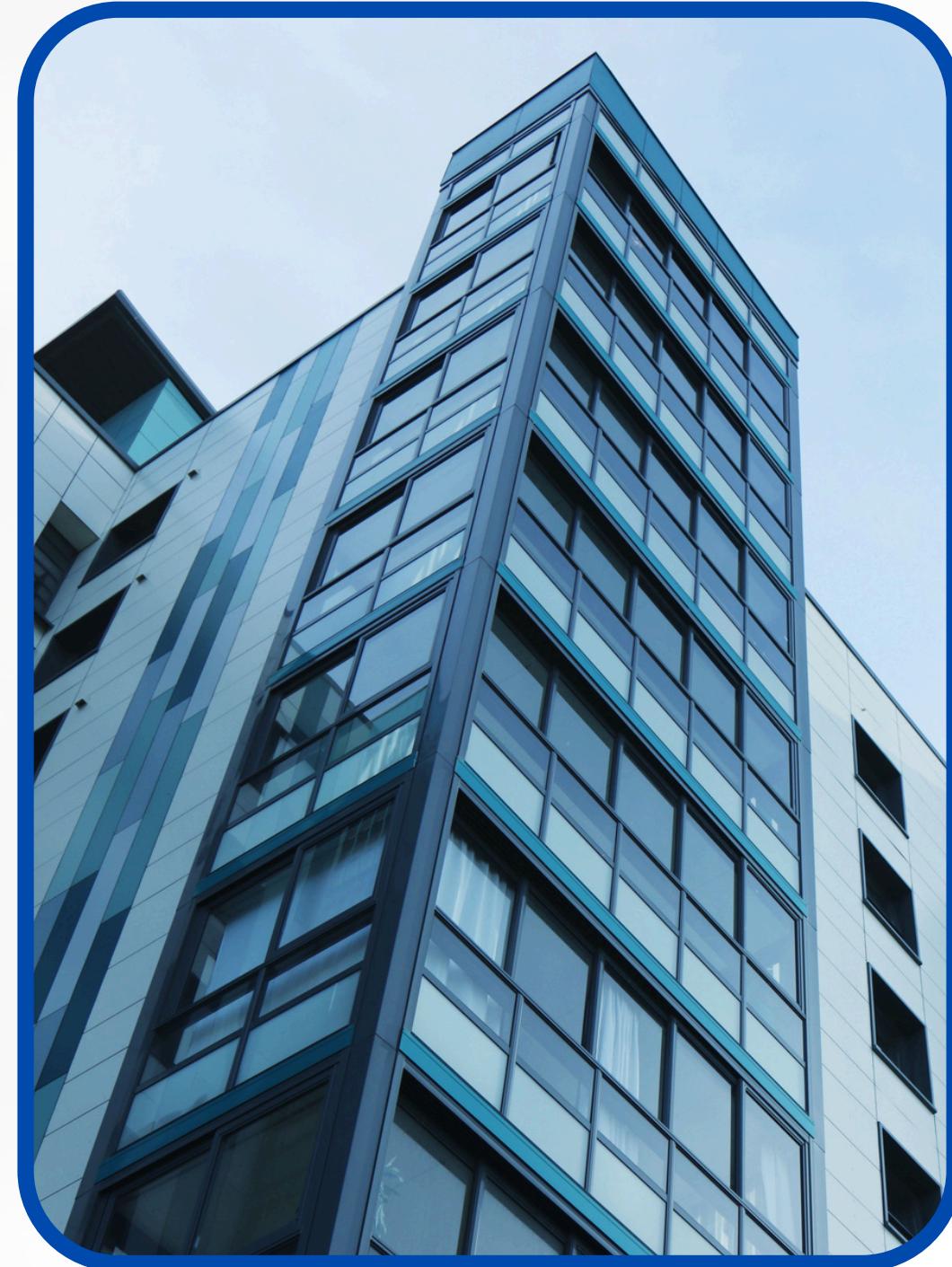
Databases

	Database Name	Create Database
<button>View</button>	admin	<button>Del</button>
<button>View</button>	config	<button>Del</button>
<button>View</button>	local	<button>Del</button>

Server Status

hostname	33a3234cc792	MongoDB Version	6.0.26
uptime	80 seconds	Node Version	18.20.3
server Time	Fri, 17 Oct 2025 00:10:59 GMT	V8 Version	10.2.154.26-node.37

CONCLUSION



Projet complet : scraping → NLP → analyse → Docker.
Approche ETL maîtrisée et documentée.
Apprentissage approfondi sur Docker et FastAPI.
« De la donnée brute à l'information utile.
»

REMERCIEMENTS

Merci pour l'attention portée à ce projet.
Ce test technique a été une véritable opportunité d'apprentissage

WAIL.BR



+33 638010145



wbrimesse@gmail.com