UNIVERSITÉ CAEN NORMANDIE

IUT GRAND OUEST NORMANDIE

BOURSETRACK

ANALYSE FINANCIÈRE ET ÉCONOMIQUE SIMPLIFIÉE

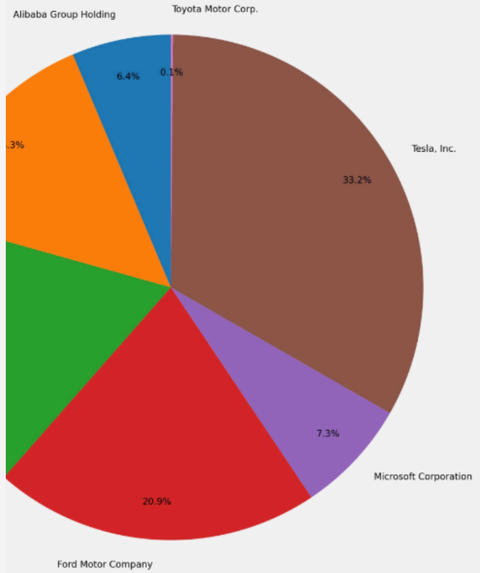
LESUEUR Romain
DIOP Sidy
SAINT-HUBERT Courteney

Année Universitaire : 2024-2025

Performances Boursières des Grandes Entreprises

Prix Moyen Ouverture	Prix Moyen Fermeture	Volatilité Moyenne	Volume Moyen	Tendance Globale	
225.34	225.55	4.17	49928544.75	0.21	
186.76	186.57	4.21	39976756.46	-0.19	
89.25	89.28	2.0	17744397.36	0.03	
11.13	11.13	0.28	58310526.3	0.0	
423.06	422.34	7.46	20470306.13	-0.72	
180.77	180.63	2.31	360169.53	-0.13	
246.1	245.81	11.25	92638267.38	-0.28	

Volume Total des Échanges par Entreprise



Actualités Financières

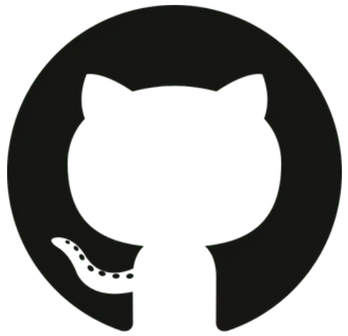
Entreprise	Date de l'article	Actualités
Apple Inc.	15 novembre 2024, 13:00	An anonymous reader quotes a report from Bloomberg: The Australian government plar online, the latest move by the center-left Labor administration to crack
Apple Inc.	06 novembre 2024, 12:26	Apple is still deciding on the displays for its expected lower-cost Apple Vision Pro, but is and also thinner than in the current headset.Inside of Apple Visio
Apple Inc.	13 novembre 2024, 11:58	The CEO of smart ring firm Oura has detailed the reasons there shouldn't be an Apple Ri possible Apple RingOh, just bring out a ring already. Apple Ring has
Alibaba Group Holding	15 novembre 2024, 09:54	(Bloomberg) -- Chinese tech conglomerate Alibaba Group Holding Ltd. is considering an soon...
Alibaba Group Holding	15 novembre 2024, 12:22	(Bloomberg) -- Alibaba Group Holding Ltd. reported solid growth in businesses including the ...
Alibaba Group Holding	18 novembre 2024, 02:55	(Bloomberg) -- Alibaba Group Holding Ltd. has hired banks to sell dollar and yuan bon shares...
Ford Motor Company	01 novembre 2024, 10:07	The Detroit automakers have made Election Day
Ford Motor Company	21 novembre 2024, 03:51	Ford lost $3.7 billion on its EV s
Ford Motor Company	18 novembre 2024, 23:58	Plus: Maxar Space Systems confirms employee info stolen in digital intrusion Ford Moto after attackers claimed to have stolen an internal database containing 4

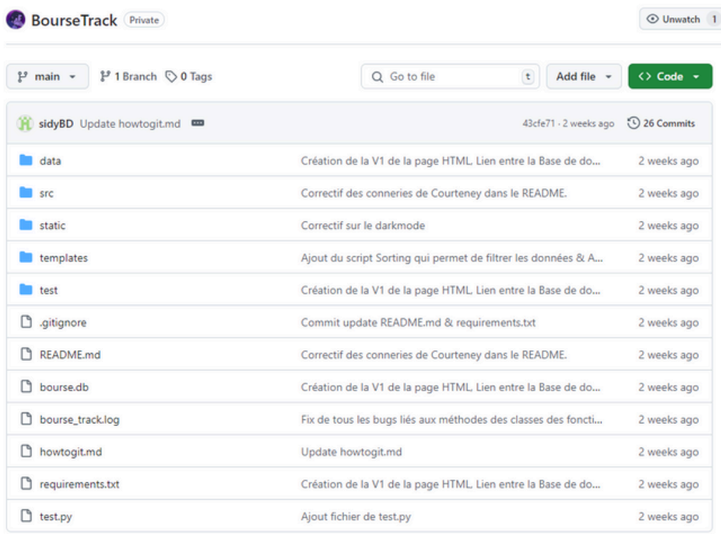
SOMMAIRE

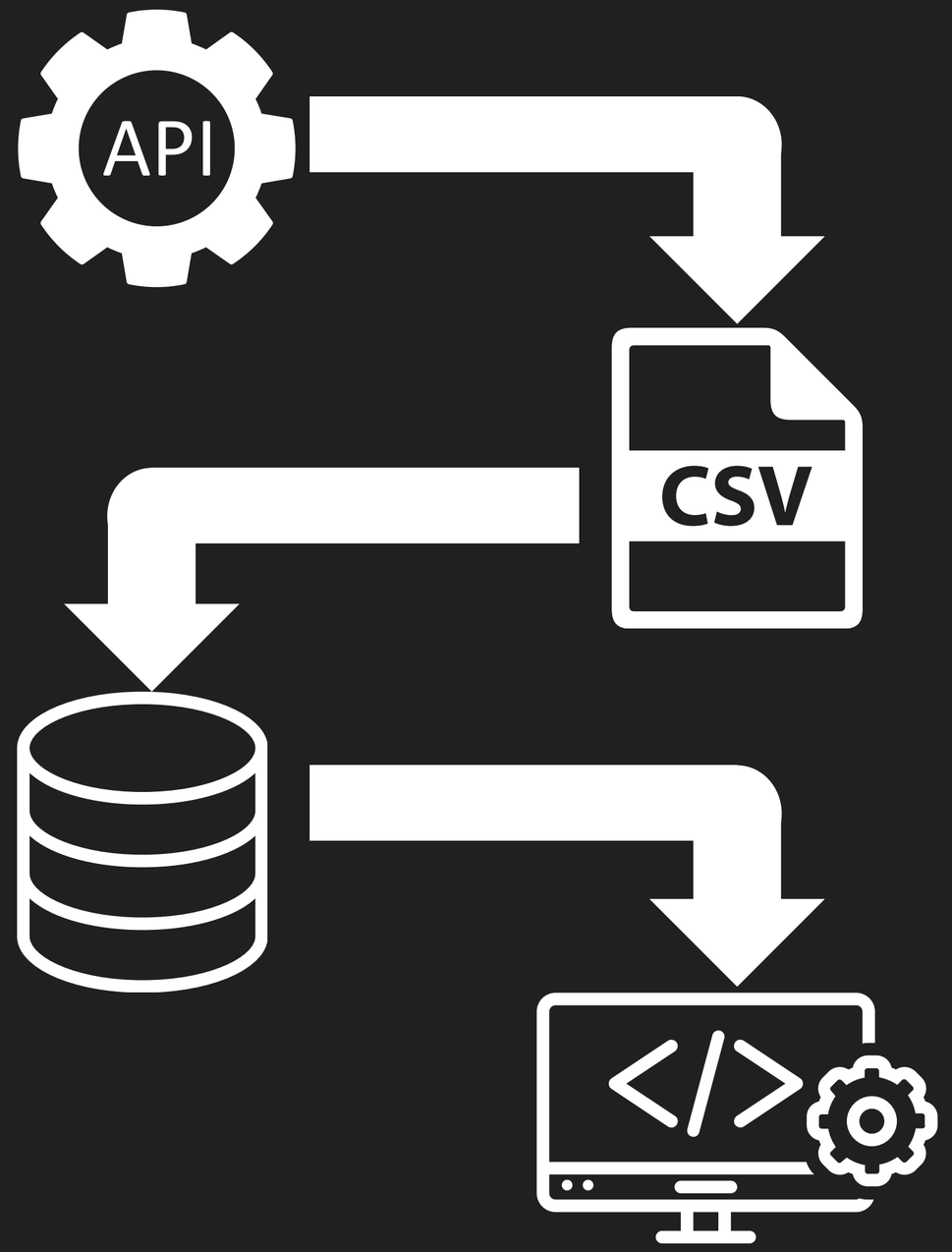
INTRODUCTION

OBJECTIF DU PROJET

Développer une page web simple et intuitive pour suivre les performances financières des grandes entreprises sur les 100 derniers jours ouvrés.







APIS ET COLLECTE DE DONNÉES

ALPHA VANTAGE

News API

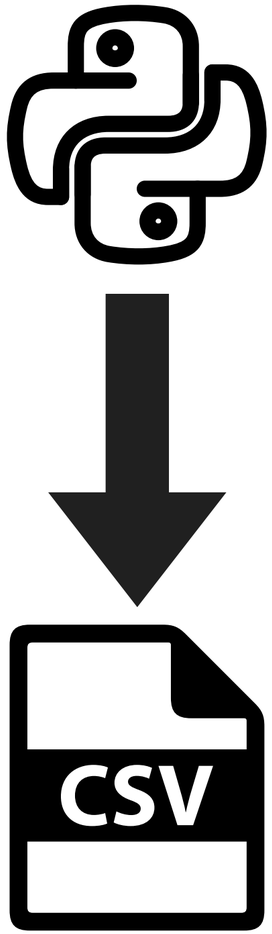
```python
def collecter_donnees_entreprises(self, entreprises):
    """
    Collecte les données financières et les actualités pour plusieurs entreprises.
    :param entreprises: Dictionnaire des entreprises à traiter.
    :return: Dictionnaire contenant les données collectées.
    """
    donnees_financieres = {}
    actualites = {}

    for symbole, info in entreprises.items():
        try:
            donnees_financieres[symbole] = self.get_stock_data(symbole)
            actualites[symbole] = self.get_news(info["nom"])
            print(f"Collecte réussie pour {info['nom']} ({symbole})")
        except Exception as e:
            print(f"Erreur pour {info['nom']} ({symbole}): {e}")

    return donnees_financieres, actualites
```

```python
def get_stock_data(self, symbol, max_retries=5):
    retry_count = 0
    url = "https://www.alphavantage.co/query"
    params = {
        "function": "TIME_SERIES_DAILY",
        "symbol": symbol,
        "apikey": self.alpha_vantage_key
    }
    while retry_count < max_retries:
        response = requests.get(url, params=params)
        if response.status_code == 200:
            data = response.json()
            if "Time Series (Daily)" in data:
                return self._formater_donnees_financieres(data["Time Series (Daily)"])
            elif "Note" in data:
                print(f"Rate limit atteint pour {symbol}. Attente...")
                time.sleep(60)
                retry_count += 1
            else:
                raise Exception(f"Erreur : données introuvables pour {symbol}.")
        elif response.status_code == 429:
            print(f"Rate limit HTTP 429 pour {symbol}. Réessai...")
            time.sleep(60)
            retry_count += 1
        else:
            raise Exception(f"Erreur API Alpha Vantage ({response.status_code}) pour {symbol}")
    raise Exception(f"Échec après {max_retries} tentatives pour {symbol}")
```

APIS ET COLLECTE DE DONNÉES

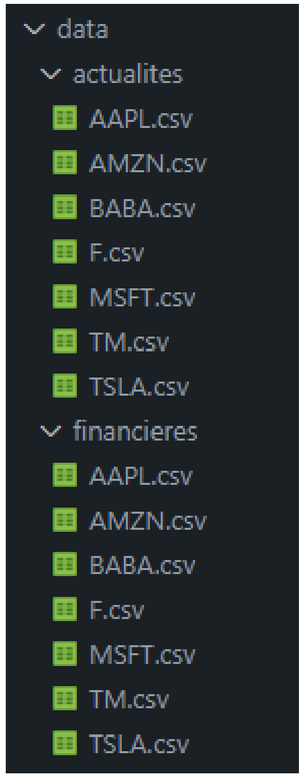



```python
from gestionnaire_api import GestionnaireAPI
from gestionnaire_fichier import GestionnaireFichier
from gestionnaire_logs import logger

NOMS_ENTREPRISES = {
    "TSLA": {"nom": "Tesla, Inc.", "secteur": "Automobile"},
    "F": {"nom": "Ford Motor Company", "secteur": "Automobile"},
    "TM": {"nom": "Toyota Motor Corp.", "secteur": "Automobile"},
    "AMZN": {"nom": "Amazon.com, Inc.", "secteur": "Commerce électronique"},
    "BABA": {"nom": "Alibaba Group Holding", "secteur": "Commerce électronique"},
    "AAPL": {"nom": "Apple Inc.", "secteur": "Technologie"},
    "MSFT": {"nom": "Microsoft Corporation", "secteur": "Technologie"}
}

def collecter_et_sauvegarder():
    api = GestionnaireAPI()
    try:
        donnees_financieres, actualites = api.collecter_donnees_entreprises(NOMS_ENTREPRISES)
        GestionnaireFichier.sauvegarder_donnees_entreprises(donnees_financieres, dossier="data/financieres")
        GestionnaireFichier.sauvegarder_donnees_entreprises(actualites, dossier="data/actualites")
        logger.info("Données financières et actualités sauvegardées dans les fichiers CSV.")
    except Exception as e:
        logger.error(f"Erreur lors de la collecte ou de la sauvegarde des données : {e}")


if __name__ == "__main__":
    collecter_et_sauvegarder()
```



BASE DE DONNÉES ET INTÉGRATION

SQLA

	id	symbole	secteur	date	open	high	low
	Search column...	Search column...	Search column...	Search column...	Search column...	Search column...	Search column...
1	1	AAPL	Technologie	2024-11-26	233.33	235.57	233.33
2	2	AAPL	Technologie	2024-11-25	231.46	233.245	229.74
3	3	AAPL	Technologie	2024-11-22	228.06	230.7199	228.06
4	4	AAPL	Technologie	2024-11-21	228.88	230.155	225.7103
5	5	AAPL	Technologie	2024-11-20	228.06	229.93	225.89
6	6	AAPL	Technologie	2024-11-19	226.98	230.16	226.66
7	7	AAPL	Technologie	2024-11-18	225.25	229.74	225.17
8	8	AAPL	Technologie	2024-11-15	226.4	226.92	224.27
9	9	AAPL	Technologie	2024-11-14	225.02	228.87	225
10	10	AAPL	Technologie	2024-11-13	224.01	226.65	222.76

```python
Base = declarative_base()

# Classe pour représenter une entreprise
class Entreprise(Base):
    __tablename__ = "entreprises"

    symbole = Column(String, primary_key=True)  # Identifiant unique pour l'entreprise
    nom = Column(String, nullable=False)
    secteur = Column(String, nullable=False)

    # Relation avec les données financières
    donnees_financieres = relationship("DonneeFinanciere", back_populates="entreprise")

    def __repr__(self):
        return f"<Entreprise(symbole={self.symbole}, nom={self.nom}, secteur={self.secteur})>"

# Classe pour représenter les données financières
class DonneeFinanciere(Base):
    __tablename__ = "donnees_financieres"

    id = Column(Integer, primary_key=True)
    symbole = Column(String, ForeignKey('entreprises.symbole'), nullable=False)
    secteur = Column(String, nullable=False)
    date = Column(Date, nullable=False)
    open = Column(Float, nullable=False)
    high = Column(Float, nullable=False)
    low = Column(Float, nullable=False)
    close = Column(Float, nullable=False)
    volume = Column(Integer, nullable=False)

    entreprise = relationship("Entreprise", back_populates="donnees_financieres")

    def __repr__(self):
        return f"<DonneeFinanciere(symbole={self.symbole}, date={self.date}, close={self.close})>"
```

BASE DE DONNÉES ET INTÉGRATION

- Héritage:
  - Appel de la classe parente gestionnaire_bdd

```python
class InsertionBDD:
    def __init__(self):
        self.gestionnaire_bdd = GestionnaireBDD()  # Initialisation du gestionnaire de base de données
```

- Appel des fichiers CSV du dossier “financières” pour le traitement
- Vérification du contenu des fichiers
- Vérification de l’existence de l’entreprise dans le dictionnaire d’intérêt
- Vérification de l’existence de l’entreprise dans la table Entreprise
- Formatage du type des données
- Insertion des données dans leurs tables respectives

```python
    def traiter_et_inserer(self, dossier="data/financieres"):
        fichiers_csv = self.get_fichiers_csv(dossier)
        for fichier in fichiers_csv:
            if os.path.getsize(fichier) == 0:
                raise Exception(f"Erreur de format : le fichier {fichier} est vide.")

            symbole = os.path.basename(fichier).split(".")[0]
            entreprise = NOMS_ENTREPRISES.get(symbole)

            if not entreprise:
                raise Exception(f"Erreur : le symbole {symbole} n'est pas reconnu.")

            secteur = entreprise["secteur"]

            # Ajoute l'entreprise si elle n'existe pas déjà
            if not self.gestionnaire_bdd.entreprise_existe(symbole):
                self.gestionnaire_bdd.ajouter_entreprise(symbole, entreprise["nom"], secteur)

            donnees = GestionnaireFichier.charger_donnees_csv(fichier)
            if not donnees:
                raise Exception(f"Erreur de format : le fichier {fichier} est vide.")

            donnees_traites = self.formater_donnees_pour_insertion(donnees, secteur)
            self.gestionnaire_bdd.ajouter_donnees_financieres(donnees_traites, symbole)
```

- Création de l’objet de type InsertionBDD
  - Appel de la méthode traiter et inserer sur l’instance “insertion”

```python
if __name__ == "__main__":
    insertion = InsertionBDD()
    insertion.traiter_et_inserer(dossier="data/financieres")
```

INTERFACE WEB

```python
    # Filtrer les entreprises ayant une tendance >= 0
    entreprises_filtrees = [e for e in indicateurs if e["tendance_globale"] >= 0]

    # Lire les actualités pertinentes
    actualites = {}
    for entreprise in entreprises_filtrees:
        symbole = entreprise["symbole"]
        nom = entreprise["nom"]
        fichier_actualites = os.path.join(BASE_DIR, '../data/actualites', f'{symbole}.csv')
        if os.path.exists(fichier_actualites):
            with open(fichier_actualites, 'r', encoding='utf-8') as f:
                reader = csv.DictReader(f) #lire comme etant un fichier csv
                actualites[nom] = []
                count = 0
                for actualite in reader:
                    if count >= 3:
                        break
                    actualite["PublishedAt"] = datetime.strptime(actualite["PublishedAt"], "%Y-%m-%dT%H:%M:%SZ").strftime("%d %B %Y, %H:%M")
                    actualites[nom].append(actualite)
                    count += 1

    # Générer le HTML
    return render_template(
        'index.html',
        entreprises=entreprises_filtrees,
        actualites=actualites,
        indicateurs=indicateurs,
        graphique=graphique  # Ajouter le graphique au contexte
    )

```


```python
@app.route("/")
def index():
    """
    Route principale pour afficher les indicateurs financiers, le graphique et les actualités.
    """
    indicateurs = bdd.calculer_indicateurs_financiers()
    graphique_path = afficher_volume_total()  # Chemin relatif à static

    return generer_html(indicateurs, graphique=graphique_path)


if __name__ == "__main__":
    app.run(debug=True)
```

INTERFACE WEB


```html
<!-- Tableau des indicateurs financiers -->
<table border="1" id="tableau-indicateurs">
    <thead>
        <tr>
            <th data-column="nom" class="sortable">Entreprise<span class="sort-indicator"></span></th>
            <th data-column="prix_moyen_ouverture" class="sortable">Prix Moyen Ouverture<span class="sort-indicator"></span></th>
            <th data-column="prix_moyen_fermeture" class="sortable">Prix Moyen Fermeture<span class="sort-indicator"></span></th>
            <th data-column="volatilite_moyenne" class="sortable">Volatilité Moyenne<span class="sort-indicator"></span></th>
            <th data-column="volume_moyen" class="sortable">Volume Moyen<span class="sort-indicator"></span></th>
            <th data-column="tendance_globale" class="sortable">Tendance Globale<span class="sort-indicator"></span></th>
            <th data-column="rendement_moyen_journalier" class="sortable">Rendement Moyen Journalier<span class="sort-indicator"></span></th>
        </tr>
    </thead>
    <tbody>
        {% for indicateur in indicateurs %}
        <tr>
            <td>{{ indicateur.nom }}</td>
            <td>{{ indicateur.prix_moyen_ouverture | round(2) }}</td>
            <td>{{ indicateur.prix_moyen_fermeture | round(2) }}</td>
            <td>{{ indicateur.volatilite_moyenne | round(2) }}</td>
            <td>{{ indicateur.volume_moyen | round(2) }}</td>
            <td>{{ indicateur.tendance_globale | round(2) }}</td>
            <td>{{ indicateur.rendement_moyen_journalier | round(2) }}</td>
        </tr>
        {% endfor %}
    </tbody>
</table>
```

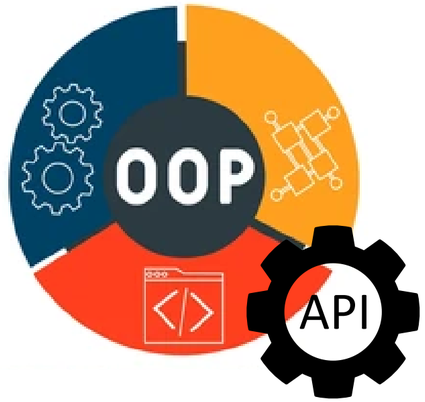
INTERFACE WEB

HTML

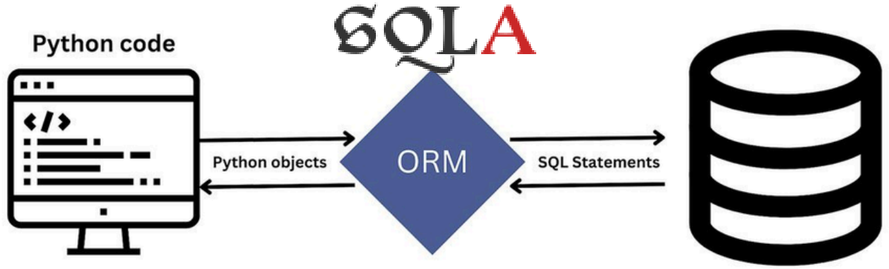
```
<table border="1" class="news-table">
    <thead>
        <tr>
            <th>Entreprise</th>
            <th>Date de l'article</th>
            <th>Actualités</th>
        </tr>
    </thead>
    <tbody>
        {% for entreprise in entreprises %}
            {% if entreprise.nom in actualites %}
                {% for actualite in actualites[entreprise.nom] %}
                    <tr>
                        <td>
                            <a href="{{ actualite.URL }}" target="_blank">{{ entreprise.nom }}</a>
                        </td>
                        <td>
                            <p><strong>{{ actualite.PublishedAt }}</strong></p>
                        </td>
                        <td>
                            <p>{{ actualite.Description }}</p>
                        </td>
                    </tr>
                {% endfor %}
            {% else %}
                <tr>
                    <td colspan="3">Aucune actualité disponible pour {{ entreprise.nom }}</td>
                </tr>
            {% endif %}
        {% endfor %}
    </tbody>
</table>
```

DÉFIS RENCONTRÉS

ROMAIN

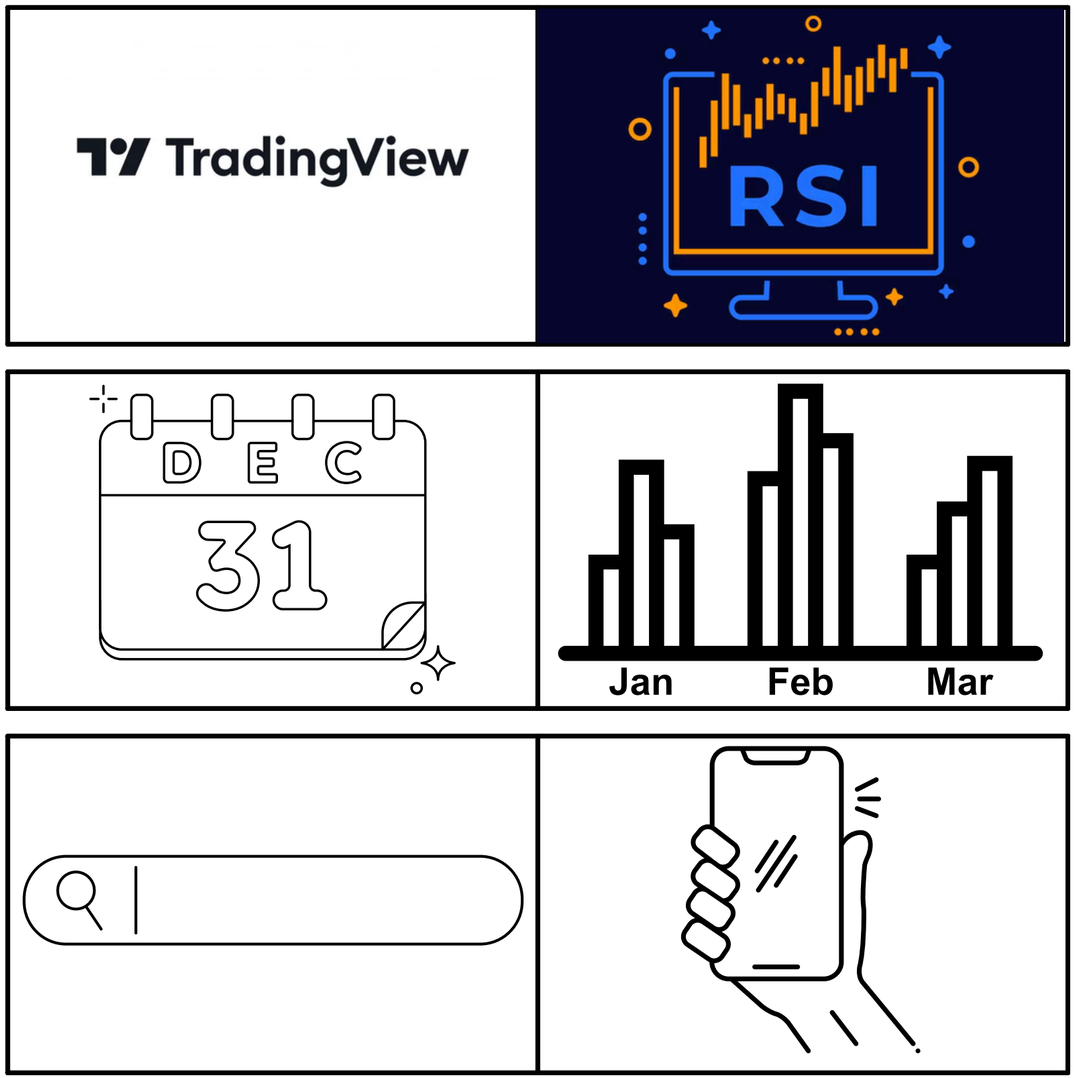


SIDY



COURTENEY

PERSPECTIVES D'AMÉLIORATIONS



MERCI.