Contexte

Une PME souhaite automatiser la création d'études de marché pour ses produits en fonction des tendances et des avis clients disponibles en ligne. L'objectif de ce TP est de construire une mini application web avec **Flask** et un **LLM** (Ollama, GPT ou Gemini) qui génère automatiquement un rapport PDF synthétique.

Objectifs pédagogiques

- Comprendre comment intégrer un LLM via API ou local.
- Construire un backend Flask pour orchestrer l'analyse.
- Structurer automatiquement le texte généré par le LLM.
- Générer un PDF avec un rapport d'étude de marché exploitable.

Étape 1 : Préparer le projet Flask

1.1 Créer l'environnement virtuel

```
python -m venv venv
# Linux / Mac
source venv/bin/activate
# Windows
venv\Scripts\activate
```

1.2 Installer Flask et Requests

```
pip install flask requests reportlab
```

1.3 Créer la structure du projet

- app.py
- templates/
- static/

Étape 2 : Créer un endpoint pour analyser un produit

```
from flask import Flask, request, render_template
import requests

app = Flask(__name__)
```



```
5
6  def call_llm(prompt):
7      # Exemple avec Ollama local
8      response = requests.post(
9          "http://localhost:11434/api/generate",
10         json={"model": "llama3", "prompt": prompt}
11     )
12     return response.json()["response"]
13
14 @app.route('/')
15 def home():
16     return render_template('index.html')
17
18 @app.route('/analyse_market', methods=['POST'])
19 def analyse_market():
20     produit = request.form['produit']
21     secteur = request.form['secteur']
22     prompt = f"Fais une  tude  de march  synth tique pour {produit}
           dans le secteur {secteur}, inclut concurrents, tendances et
           points forts/faibles."
23     texte = call_llm(prompt)
24     return render_template('result.html', contenu=texte)
```

Étape 3 : Créer l'interface HTML simple

```
<!-- templates/index.html -->
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <title>Analyse de march </title>
</head>
<body>
    <h1>Analyse de march  automatique</h1>
    <form action="/analyse_market" method="post">
        Produit: <input type="text" name="produit"><br>
        Secteur: <input type="text" name="secteur"><br>
        <input type="submit" value="Analyser">
    </form>
</body>
</html>
```

Étape 4 : Génération du rapport PDF

```
from reportlab.lib.pagesizes import A4
from reportlab.pdfgen import canvas
```

```
def generate_pdf(content, filename="rapport.pdf"):
    c = canvas.Canvas(filename, pagesize=A4)
    c.setFont("Helvetica-Bold", 16)
    c.drawString(50, 800, " tude de march automatis e")
    c.setFont("Helvetica", 12)
    c.drawString(50, 780, "Contenu g n r par LLM:")
    y = 760
    for line in content.split('\n'):
        c.drawString(50, y, line)
        y -= 15
    c.save()
```

Étape 5 : Test complet

1. Lancer l'application Flask : `python app.py`
2. Accéder à `http://localhost:5000`
3. Tester différents produits et secteurs
4. Générer et télécharger le PDF

Challenge avancé : Étude de marché complète en autonomie

Dans cette partie, vous devez aller plus loin et construire **vous-même** des fonctionnalités supplémentaires pour rendre l'application plus proche d'une vraie solution business. Aucune solution complète n'est fournie : vous devez concevoir la logique, structurer vos prompts, et générer les livrables.

Objectifs du challenge

— Comparer plusieurs produits dans un même secteur et générer un tableau comparatif dans le PDF.

— Ajouter des graphiques simples (histogrammes, tendances, parts de marché simulées) à partir des données générées par le LLM.

— Permettre à l'utilisateur de sélectionner plusieurs produits à analyser en une seule requête.

— Ajouter la possibilité d'envoyer le rapport PDF par email à un destinataire choisi.

— Structurer le PDF avec des sections et sous-sections pour rendre le rapport professionnel.

Suggestions pour démarrer

— Réfléchissez à la façon de **structurer les prompts** pour que le LLM renvoie des informations exploitables (ex. JSON ou texte bien découpé).

— Explorez des librairies Python pour générer des tableaux et graphiques dans vos PDF (`reportlab`, `matplotlib`, etc.).

— Testez votre application étape par étape : d'abord le prompt et le LLM, ensuite la génération PDF, puis l'envoi par email.

— Documentez votre travail et soyez capable d'expliquer vos choix de conception.

Livrables attendus

— Code Flask complet avec endpoints et logique de génération PDF.

— Un PDF d'étude de marché généré automatiquement pour au moins deux produits différents.

— Optionnel : graphiques et comparaison détaillée dans le PDF.

— Une brève explication de votre approche et des prompts utilisés pour le LLM.

Note : Cette partie du TP est conçue pour que vous réfléchissiez de manière autonome et appliquiez les concepts appris précédemment. Vous pouvez vous inspirer des étapes précédentes, mais l'implémentation finale doit être votre propre travail.

