Ci‑dessous, vous trouverez l’ensemble complet des fichiers de code (sans modification) ainsi qu’une présentation détaillée de l’application "Journal de Trading Web". Vous pourrez copier ce contenu dans un fichier Word pour l’enregistrer et le partager avec vos associés.

**Présentation Complète du Site "Journal de Trading Web"**

**Journal de Trading Web** est une application web complète développée avec Flask et SQLite, qui s’appuie sur Bootstrap et Animate.css pour offrir une interface moderne, responsive et animée. Le site a été conçu pour permettre aux traders de :

* **Gérer leurs journaux de trading :**  
  Chaque utilisateur peut créer plusieurs journaux. Pour chaque journal, l’utilisateur définit un nom, un capital initial et choisit une devise (EUR, USD, GBP, JPY). Le solde du journal se met à jour automatiquement en fonction des résultats des trades terminés.
* **Enregistrer et suivre leurs trades :**  
  Dans un journal, l’utilisateur peut enregistrer des trades en renseignant :
  + La date et l’heure de début.
  + La session (choix parmi Londres, New York, Tokyo, Sydney, Crypto 24/7).
  + L’instrument financier et la position (achat ou vente).
  + Le prix d’entrée, le lot et le ratio Risk/Reward.
  + Les time frames sont désormais sélectionnés séparément pour le "Lower Time Frame" et le "Higher Time Frame" (avec des options prédéfinies telles que M1, M5, M15, M30 pour le lower et H1, H4, D1, W1 pour le higher, ou l’option "Custom" pour une saisie manuelle).
  + Des commentaires et la possibilité de joindre une capture d’écran (qui est affichée cliquable pour l’ouvrir dans un nouvel onglet).
* **Analyser leurs performances :**  
  Le dashboard d’un journal affiche :
  + Le capital initial et le solde actuel (calculé automatiquement par l’addition du capital initial aux résultats de tous les trades terminés).
  + Des graphiques dynamiques (créés avec Chart.js) illustrant les gains par mois, le nombre de trades, une répartition du capital initial versus les gains/pertes et d’autres statistiques pertinentes.
  + Une liste des plateformes externes liées (MT4, MT5, TradingView).
* **Créer et partager des analyses :**  
  L’utilisateur peut créer des analyses en saisissant un titre, un contenu et en joignant une image. Une fois créées, ces analyses peuvent être partagées de deux façons :
  + **Partager avec la Communauté :** En entrant la valeur "all", l’analyse sera visible dans l’onglet Communauté.
  + **Partager directement :** En entrant un email spécifique, l’analyse apparaîtra dans l’onglet "Mes Partages" du destinataire.
  + Les destinataires peuvent consulter, commenter et interagir avec les analyses partagées.
* **Liaison de plateformes externes :**  
  Dans chaque journal, l’utilisateur peut lier des comptes de plateformes externes (MT4, MT5, TradingView) en renseignant un identifiant et des détails, ces informations étant affichées dans le dashboard.
* **Assistance et Conversations :**
  + Les utilisateurs peuvent envoyer des messages d’assistance via un formulaire. Ils consultent ensuite la liste de leurs messages et peuvent ouvrir une conversation pour échanger avec l’équipe d’administration.
  + Les administrateurs, via un onglet dédié, reçoivent tous les messages d’assistance, avec les coordonnées de l’expéditeur affichées (nom, prénom, email) et un lien pour modifier directement le compte de l’utilisateur si nécessaire. Ils peuvent répondre via une interface de conversation bidirectionnelle.
* **Publications Info (Type "Reel" ou "Statut") :**  
  Les administrateurs peuvent créer des publications d’information en saisissant un titre, un contenu et en joignant un média (vidéo, audio ou image). Ces publications sont affichées sous forme de cartes esthétiques avec le média en arrière-plan et un overlay de texte, offrant un rendu moderne similaire aux reels sur YouTube ou aux statuts WhatsApp.
* **Administration Avancée :**  
  Un espace administrateur permet de :
  + Consulter la liste complète des utilisateurs.
  + Modifier leurs informations (prénom, nom, pays, email, mot de passe).
  + Passer un compte en administrateur ou retirer ce statut.
  + Gérer les messages d’assistance et les conversations.
* **Esthétique et Design :**  
  L’interface est construite avec Bootstrap et Animate.css pour assurer un rendu moderne et attractif. Les images, notamment les captures d’écran des trades, sont affichées de manière cliquable pour permettre une visualisation en grand format.

**Fichiers de Code (Complet et Sans Modification)**

Vous trouverez ci-dessous l’intégralité de tous les fichiers de code créés pour le projet :

**1. app.py**

import os

from datetime import datetime

from flask import Flask, render\_template, request, redirect, url\_for, flash, session, send\_from\_directory

from flask\_sqlalchemy import SQLAlchemy

from werkzeug.security import generate\_password\_hash, check\_password\_hash

app = Flask(\_\_name\_\_)

app.config.from\_object('config.Config')

db = SQLAlchemy(app)

# Dossier pour l'upload de fichiers (captures, médias, etc.)

basedir = os.path.abspath(os.path.dirname(\_\_file\_\_))

UPLOAD\_FOLDER = os.path.join(basedir, 'static', 'uploads')

os.makedirs(UPLOAD\_FOLDER, exist\_ok=True)

app.config['UPLOAD\_FOLDER'] = UPLOAD\_FOLDER

#################################################

# Modèles de la base de données

#################################################

class User(db.Model):

\_\_tablename\_\_ = 'users'

id = db.Column(db.Integer, primary\_key=True)

prenom = db.Column(db.String(50), nullable=False)

nom = db.Column(db.String(50), nullable=False)

pays = db.Column(db.String(50))

email = db.Column(db.String(120), unique=True, nullable=False)

password = db.Column(db.String(200), nullable=False) # stocke le hash

is\_admin = db.Column(db.Boolean, default=False)

journals = db.relationship('Journal', backref='owner', lazy=True)

assistance\_messages = db.relationship('AssistanceMessage', backref='user', lazy=True)

class Journal(db.Model):

\_\_tablename\_\_ = 'journals'

id = db.Column(db.Integer, primary\_key=True)

nom = db.Column(db.String(100), nullable=False)

capital\_initial = db.Column(db.Float, nullable=False)

devise = db.Column(db.String(10), nullable=False)

user\_id = db.Column(db.Integer, db.ForeignKey('users.id'), nullable=False)

trades = db.relationship('Trade', backref='journal', lazy=True)

analyses = db.relationship('Analysis', backref='journal', lazy=True)

platforms = db.relationship('PlatformLink', backref='journal', lazy=True)

class Trade(db.Model):

\_\_tablename\_\_ = 'trades'

id = db.Column(db.Integer, primary\_key=True)

date\_debut = db.Column(db.DateTime, default=datetime.utcnow)

date\_fin = db.Column(db.DateTime, nullable=True)

session = db.Column(db.String(50), nullable=False) # Ex: Londres, New York, etc.

instrument = db.Column(db.String(50), nullable=False)

position = db.Column(db.String(10), nullable=False) # "achat" ou "vente"

prix\_entree = db.Column(db.Float, nullable=False)

prix\_sortie = db.Column(db.Float, nullable=True)

resultat = db.Column(db.Float, nullable=True)

pourcentage = db.Column(db.Float, nullable=True)

lower\_time\_frame = db.Column(db.String(10))

higher\_time\_frame = db.Column(db.String(10))

lot = db.Column(db.Float, nullable=False)

risk\_reward = db.Column(db.Float, nullable=False)

commentaires = db.Column(db.Text)

capture = db.Column(db.String(200))

statut = db.Column(db.String(20), default="EN\_COURS") # "EN\_COURS" ou "TERMINE"

journal\_id = db.Column(db.Integer, db.ForeignKey('journals.id'), nullable=False)

class Analysis(db.Model):

\_\_tablename\_\_ = 'analyses'

id = db.Column(db.Integer, primary\_key=True)

titre = db.Column(db.String(100), nullable=False)

contenu = db.Column(db.Text)

image = db.Column(db.String(200))

date\_creation = db.Column(db.String(20), default=lambda: datetime.now().strftime('%Y-%m-%d'))

journal\_id = db.Column(db.Integer, db.ForeignKey('journals.id'), nullable=False)

class PlatformLink(db.Model):

\_\_tablename\_\_ = 'platform\_links'

id = db.Column(db.Integer, primary\_key=True)

plateforme = db.Column(db.String(50))

identifiant = db.Column(db.String(100))

details = db.Column(db.String(200))

journal\_id = db.Column(db.Integer, db.ForeignKey('journals.id'), nullable=False)

class InfoPost(db.Model):

\_\_tablename\_\_ = 'info\_posts'

id = db.Column(db.Integer, primary\_key=True)

titre = db.Column(db.String(150), nullable=False)

contenu = db.Column(db.Text, nullable=False)

media = db.Column(db.String(200)) # Vidéo, audio ou image

date\_creation = db.Column(db.String(20), default=lambda: datetime.now().strftime('%Y-%m-%d'))

class AssistanceMessage(db.Model):

\_\_tablename\_\_ = 'assistance\_messages'

id = db.Column(db.Integer, primary\_key=True)

sujet = db.Column(db.String(150), nullable=False)

message = db.Column(db.Text, nullable=False)

date\_creation = db.Column(db.String(20), default=lambda: datetime.now().strftime('%Y-%m-%d'))

user\_id = db.Column(db.Integer, db.ForeignKey('users.id'), nullable=False)

replies = db.relationship('AssistanceReply', backref='assistance\_message', lazy=True)

class AssistanceReply(db.Model):

\_\_tablename\_\_ = 'assistance\_replies'

id = db.Column(db.Integer, primary\_key=True)

reply\_message = db.Column(db.Text, nullable=False)

date\_creation = db.Column(db.String(20), default=lambda: datetime.now().strftime('%Y-%m-%d'))

assistance\_id = db.Column(db.Integer, db.ForeignKey('assistance\_messages.id'), nullable=False)

sender = db.Column(db.String(50), default="admin") # "admin" ou "user"

class AnalysisShare(db.Model):

\_\_tablename\_\_ = 'analysis\_shares'

id = db.Column(db.Integer, primary\_key=True)

analysis\_id = db.Column(db.Integer, db.ForeignKey('analyses.id'), nullable=False)

shared\_by\_user\_id = db.Column(db.Integer, db.ForeignKey('users.id'), nullable=False)

shared\_with = db.Column(db.String(120), nullable=False) # "all" ou un email spécifique

date\_shared = db.Column(db.String(20), default=lambda: datetime.now().strftime('%Y-%m-%d'))

analysis = db.relationship('Analysis', backref='shares', lazy=True)

comments = db.relationship('AnalysisShareComment', backref='share', lazy=True)

class AnalysisShareComment(db.Model):

\_\_tablename\_\_ = 'analysis\_share\_comments'

id = db.Column(db.Integer, primary\_key=True)

share\_id = db.Column(db.Integer, db.ForeignKey('analysis\_shares.id'), nullable=False)

user\_id = db.Column(db.Integer, db.ForeignKey('users.id'), nullable=False)

comment = db.Column(db.Text, nullable=False)

date\_creation = db.Column(db.String(20), default=lambda: datetime.now().strftime('%Y-%m-%d'))

#################################################

# Routes et vues

#################################################

@app.route('/')

def index():

if 'user\_id' in session:

return redirect(url\_for('home'))

return redirect(url\_for('login'))

# 1. Authentification

@app.route('/register', methods=['GET', 'POST'])

def register():

if request.method == 'POST':

prenom = request.form['prenom']

nom = request.form['nom']

pays = request.form['pays']

email = request.form['email']

password = request.form['password']

if not email or "@" not in email:

flash("Format de l'email invalide.")

return redirect(url\_for('register'))

if len(password) < 8:

flash("Le mot de passe doit contenir au moins 8 caractères.")

return redirect(url\_for('register'))

if User.query.filter\_by(email=email).first():

flash("Cet email est déjà enregistré.")

return redirect(url\_for('register'))

new\_user = User(prenom=prenom, nom=nom, pays=pays, email=email,

password=generate\_password\_hash(password), is\_admin=False)

db.session.add(new\_user)

db.session.commit()

flash("Inscription réussie ! Vous pouvez maintenant vous connecter.")

return redirect(url\_for('login'))

return render\_template('register.html')

@app.route('/login', methods=['GET', 'POST'])

def login():

if request.method == 'POST':

email = request.form['email']

password = request.form['password']

user = User.query.filter\_by(email=email).first()

if user and check\_password\_hash(user.password, password):

session['user\_id'] = user.id

session['user\_name'] = user.prenom

session['is\_admin'] = user.is\_admin

flash("Connexion réussie.")

return redirect(url\_for('home'))

else:

flash("Email ou mot de passe incorrect.")

return redirect(url\_for('login'))

return render\_template('login.html')

@app.route('/logout')

def logout():

session.clear()

return redirect(url\_for('login'))

# 2. Gestion des journaux

@app.route('/home')

def home():

if 'user\_id' not in session:

return redirect(url\_for('login'))

journals = Journal.query.filter\_by(user\_id=session['user\_id']).all()

return render\_template('home.html', journals=journals)

@app.route('/create\_journal', methods=['GET', 'POST'])

def create\_journal():

if 'user\_id' not in session:

return redirect(url\_for('login'))

if request.method == 'POST':

nom = request.form['nom']

capital\_initial = request.form['capital\_initial']

devise = request.form['devise']

try:

capital\_initial = float(capital\_initial)

except ValueError:

flash("Veuillez saisir un capital initial valide.")

return redirect(url\_for('create\_journal'))

new\_journal = Journal(nom=nom, capital\_initial=capital\_initial,

devise=devise, user\_id=session['user\_id'])

db.session.add(new\_journal)

db.session.commit()

flash("Journal créé avec succès.")

return redirect(url\_for('home'))

return render\_template('create\_journal.html')

@app.route('/dashboard/<int:journal\_id>')

def dashboard(journal\_id):

if 'user\_id' not in session:

return redirect(url\_for('login'))

journal = Journal.query.filter\_by(id=journal\_id, user\_id=session['user\_id']).first()

if not journal:

flash("Journal introuvable ou non autorisé.")

return redirect(url\_for('home'))

solde = journal.capital\_initial + sum(trade.resultat or 0 for trade in journal.trades if trade.statut=="TERMINE")

monthly\_results = {}

monthly\_count = {}

for trade in journal.trades:

if trade.date\_fin:

month = trade.date\_fin.strftime('%b')

monthly\_results[month] = monthly\_results.get(month, 0) + (trade.resultat or 0)

monthly\_count[month] = monthly\_count.get(month, 0) + 1

mois = sorted(monthly\_results.keys())

gains = [monthly\_results[m] for m in mois]

counts = [monthly\_count[m] for m in mois]

stats = {

'solde': solde,

'mois': mois,

'gains\_per\_month': gains,

'trades\_count': counts

}

platforms = journal.platforms

return render\_template('dashboard.html', journal=journal, stats=stats, platforms=platforms)

# 3. Gestion des Trades

@app.route('/trades/<int:journal\_id>', methods=['GET', 'POST'])

def trades(journal\_id):

if 'user\_id' not in session:

return redirect(url\_for('login'))

journal = Journal.query.filter\_by(id=journal\_id, user\_id=session['user\_id']).first()

if not journal:

flash("Journal introuvable ou non autorisé.")

return redirect(url\_for('home'))

if request.method == 'POST':

date\_debut\_str = request.form.get('date\_debut')

heure\_debut\_str = request.form.get('heure\_debut')

session\_trade = request.form.get('session')

instrument = request.form.get('instrument')

position = request.form.get('position')

prix\_entree\_str = request.form.get('prix\_entree')

lot\_str = request.form.get('lot')

rr\_str = request.form.get('risk\_reward')

lower\_time\_frame = request.form.get('lower\_time\_frame')

custom\_lower = request.form.get('custom\_lower\_time\_frame', '')

if lower\_time\_frame == "Custom":

lower = custom\_lower

else:

lower = lower\_time\_frame

higher\_time\_frame = request.form.get('higher\_time\_frame')

custom\_higher = request.form.get('custom\_higher\_time\_frame', '')

if higher\_time\_frame == "Custom":

higher = custom\_higher

else:

higher = higher\_time\_frame

if not all([date\_debut\_str, heure\_debut\_str, session\_trade, instrument, position, prix\_entree\_str, lot\_str, rr\_str, lower, higher]):

flash("Veuillez remplir tous les champs obligatoires.")

return redirect(url\_for('trades', journal\_id=journal\_id))

try:

date\_debut = datetime.strptime(date\_debut\_str + ' ' + heure\_debut\_str, '%Y-%m-%d %H:%M')

prix\_entree = float(prix\_entree\_str)

lot = float(lot\_str)

risk\_reward = float(rr\_str)

except ValueError:

flash("Veuillez saisir des valeurs correctes pour la date, le prix, le lot ou le R/R.")

return redirect(url\_for('trades', journal\_id=journal\_id))

capture\_filename = None

file = request.files.get('capture')

if file and file.filename:

timestamp = datetime.now().strftime('%Y%m%d%H%M%S')

capture\_filename = f"{timestamp}\_{file.filename}"

file.save(os.path.join(app.config['UPLOAD\_FOLDER'], capture\_filename))

new\_trade = Trade(

date\_debut=date\_debut,

session=session\_trade,

instrument=instrument,

position=position,

prix\_entree=prix\_entree,

lot=lot,

risk\_reward=risk\_reward,

lower\_time\_frame=lower,

higher\_time\_frame=higher,

commentaires=request.form.get('commentaires', ''),

capture=capture\_filename,

journal\_id=journal.id

)

date\_fin\_str = request.form.get('date\_fin')

heure\_fin\_str = request.form.get('heure\_fin')

prix\_sortie\_str = request.form.get('prix\_sortie')

if date\_fin\_str and heure\_fin\_str and prix\_sortie\_str:

try:

date\_fin = datetime.strptime(date\_fin\_str + ' ' + heure\_fin\_str, '%Y-%m-%d %H:%M')

prix\_sortie = float(prix\_sortie\_str)

new\_trade.date\_fin = date\_fin

new\_trade.prix\_sortie = prix\_sortie

if position.lower() == "achat":

new\_trade.resultat = (prix\_sortie - prix\_entree) \* lot

new\_trade.pourcentage = ((prix\_sortie - prix\_entree) / prix\_entree) \* 100

else:

new\_trade.resultat = (prix\_entree - prix\_sortie) \* lot

new\_trade.pourcentage = ((prix\_entree - prix\_sortie) / prix\_entree) \* 100

new\_trade.statut = "TERMINE"

except ValueError:

flash("Valeur incorrecte pour le prix de sortie ou la date de fin.")

db.session.add(new\_trade)

db.session.commit()

flash("Trade enregistré avec succès.")

return redirect(url\_for('trades', journal\_id=journal.id))

trades\_encours = Trade.query.filter\_by(journal\_id=journal.id, statut="EN\_COURS").all()

trades\_termine = Trade.query.filter(Trade.journal\_id == journal.id, Trade.statut == "TERMINE").all()

return render\_template('trades.html', journal=journal, trades\_encours=trades\_encours, trades\_termine=trades\_termine)

@app.route('/trade/<int:trade\_id>', methods=['GET', 'POST'])

def trade\_detail(trade\_id):

if 'user\_id' not in session:

return redirect(url\_for('login'))

trade = Trade.query.get(trade\_id)

if not trade:

flash("Trade introuvable.")

return redirect(url\_for('home'))

journal = Journal.query.get(trade.journal\_id)

if not journal or journal.owner.id != session['user\_id']:

flash("Accès non autorisé.")

return redirect(url\_for('home'))

if request.method == 'POST':

date\_fin\_str = request.form.get('date\_fin')

heure\_fin\_str = request.form.get('heure\_fin')

prix\_sortie\_str = request.form.get('prix\_sortie')

if date\_fin\_str and heure\_fin\_str and prix\_sortie\_str:

try:

date\_fin = datetime.strptime(date\_fin\_str + ' ' + heure\_fin\_str, '%Y-%m-%d %H:%M')

prix\_sortie = float(prix\_sortie\_str)

trade.date\_fin = date\_fin

trade.prix\_sortie = prix\_sortie

if trade.position.lower() == "achat":

trade.resultat = (prix\_sortie - trade.prix\_entree) \* trade.lot

trade.pourcentage = ((prix\_sortie - trade.prix\_entree) / trade.prix\_entree) \* 100

else:

trade.resultat = (trade.prix\_entree - prix\_sortie) \* trade.lot

trade.pourcentage = ((trade.prix\_entree - prix\_sortie) / trade.prix\_entree) \* 100

trade.statut = "TERMINE"

db.session.commit()

flash("Trade mis à jour et terminé.")

except ValueError:

flash("Valeur incorrecte pour le prix de sortie ou la date de fin.")

return redirect(url\_for('trade\_detail', trade\_id=trade.id))

return render\_template('trade\_detail.html', trade=trade)

# 4. Gestion des Analyses

@app.route('/analyses/<int:journal\_id>', methods=['GET', 'POST'])

def analyses(journal\_id):

if 'user\_id' not in session:

return redirect(url\_for('login'))

journal = Journal.query.filter\_by(id=journal\_id, user\_id=session['user\_id']).first()

if not journal:

flash("Journal introuvable ou non autorisé.")

return redirect(url\_for('home'))

if request.method == 'POST':

titre = request.form['titre']

contenu = request.form['contenu']

image\_filename = None

file = request.files.get('image')

if file and file.filename:

timestamp = datetime.now().strftime('%Y%m%d%H%M%S')

image\_filename = f"{timestamp}\_{file.filename}"

file.save(os.path.join(app.config['UPLOAD\_FOLDER'], image\_filename))

new\_analysis = Analysis(titre=titre, contenu=contenu, image=image\_filename, journal\_id=journal.id)

db.session.add(new\_analysis)

db.session.commit()

flash("Analyse ajoutée avec succès.")

return redirect(url\_for('analyses', journal\_id=journal\_id))

analyses\_list = Analysis.query.filter\_by(journal\_id=journal.id).all()

return render\_template('analyses.html', journal=journal, analyses=analyses\_list)

@app.route('/analysis/<int:analysis\_id>')

def analysis\_detail(analysis\_id):

if 'user\_id' not in session:

return redirect(url\_for('login'))

analysis = Analysis.query.get(analysis\_id)

if not analysis:

flash("Analyse introuvable.")

return redirect(url\_for('home'))

journal = Journal.query.get(analysis.journal\_id)

if not journal or journal.owner.id != session['user\_id']:

flash("Accès non autorisé.")

return redirect(url\_for('home'))

return render\_template('analysis\_detail.html', analysis=analysis)

# 5. Liaison de plateformes externes

@app.route('/link\_platform/<int:journal\_id>', methods=['GET', 'POST'])

def link\_platform(journal\_id):

if 'user\_id' not in session:

return redirect(url\_for('login'))

journal = Journal.query.filter\_by(id=journal\_id, user\_id=session['user\_id']).first()

if not journal:

flash("Journal introuvable ou non autorisé.")

return redirect(url\_for('home'))

if request.method == 'POST':

plateforme = request.form['plateforme']

identifiant = request.form['identifiant']

details = request.form['details']

new\_link = PlatformLink(plateforme=plateforme, identifiant=identifiant,

details=details, journal\_id=journal.id)

db.session.add(new\_link)

db.session.commit()

flash("Plateforme liée avec succès.")

return redirect(url\_for('link\_platform', journal\_id=journal.id))

links = PlatformLink.query.filter\_by(journal\_id=journal.id).all()

return render\_template('link\_platform.html', journal=journal, links=links)

# 6. Paramètres utilisateur

@app.route('/parametres', methods=['GET', 'POST'])

def parametres():

if 'user\_id' not in session:

return redirect(url\_for('login'))

user = User.query.get(session['user\_id'])

if not user:

flash("Utilisateur introuvable.")

return redirect(url\_for('login'))

if request.method == 'POST':

if 'new\_password' in request.form:

new\_password = request.form['new\_password']

if len(new\_password) < 8:

flash("Le mot de passe doit contenir au moins 8 caractères.")

return redirect(url\_for('parametres'))

user.password = generate\_password\_hash(new\_password)

db.session.commit()

flash("Mot de passe mis à jour avec succès.")

elif 'theme' in request.form:

session['theme'] = request.form['theme']

flash("Thème mis à jour.")

return render\_template('parametres.html', user=user)

# 7. Assistance utilisateur

@app.route('/assistance', methods=['GET', 'POST'])

def assistance():

if 'user\_id' not in session:

return redirect(url\_for('login'))

if request.method == 'POST':

sujet = request.form['sujet']

message = request.form['message']

new\_msg = AssistanceMessage(sujet=sujet, message=message, user\_id=session['user\_id'])

db.session.add(new\_msg)

db.session.commit()

flash("Votre message a été envoyé. Un administrateur vous répondra prochainement.")

return redirect(url\_for('assistance'))

user\_msgs = AssistanceMessage.query.filter\_by(user\_id=session['user\_id']).all()

return render\_template('my\_assistance.html', user\_msgs=user\_msgs)

@app.route('/my\_conversation/<int:msg\_id>', methods=['GET', 'POST'])

def my\_conversation(msg\_id):

if 'user\_id' not in session:

return redirect(url\_for('login'))

msg = AssistanceMessage.query.get(msg\_id)

if not msg or msg.user\_id != session['user\_id']:

flash("Message introuvable ou non autorisé.")

return redirect(url\_for('assistance'))

if request.method == 'POST':

reply\_text = request.form['reply']

new\_reply = AssistanceReply(reply\_message=reply\_text, assistance\_id=msg.id, sender="user")

db.session.add(new\_reply)

db.session.commit()

flash("Réponse envoyée.")

return redirect(url\_for('my\_conversation', msg\_id=msg.id))

return render\_template('my\_conversation.html', msg=msg)

# 8. Admin – Assistance et conversation

@app.route('/admin\_assistance')

def admin\_assistance():

if 'user\_id' not in session or not session.get('is\_admin'):

flash("Accès refusé.")

return redirect(url\_for('login'))

messages = AssistanceMessage.query.order\_by(AssistanceMessage.date\_creation.desc()).all()

return render\_template('admin\_assistance.html', messages=messages)

@app.route('/conversation/<int:msg\_id>', methods=['GET', 'POST'])

def conversation\_detail(msg\_id):

if 'user\_id' not in session or not session.get('is\_admin'):

flash("Accès refusé.")

return redirect(url\_for('login'))

msg = AssistanceMessage.query.get(msg\_id)

if not msg:

flash("Message introuvable.")

return redirect(url\_for('admin\_assistance'))

if request.method == 'POST':

reply\_text = request.form['reply']

new\_reply = AssistanceReply(reply\_message=reply\_text, assistance\_id=msg.id, sender="admin")

db.session.add(new\_reply)

db.session.commit()

flash("Réponse envoyée.")

return redirect(url\_for('conversation\_detail', msg\_id=msg.id))

return render\_template('conversation\_detail.html', msg=msg)

# 9. Publications Info

@app.route('/info', methods=['GET', 'POST'])

def info():

if 'user\_id' not in session:

return redirect(url\_for('login'))

if session.get('is\_admin') and request.method == 'POST':

titre = request.form['titre']

contenu = request.form['contenu']

media\_filename = None

file = request.files.get('media')

if file and file.filename:

timestamp = datetime.now().strftime('%Y%m%d%H%M%S')

media\_filename = f"{timestamp}\_{file.filename}"

file.save(os.path.join(app.config['UPLOAD\_FOLDER'], media\_filename))

new\_info = InfoPost(titre=titre, contenu=contenu, media=media\_filename)

db.session.add(new\_info)

db.session.commit()

flash("Publication créée avec succès.")

return redirect(url\_for('info'))

posts = InfoPost.query.order\_by(InfoPost.id.desc()).all()

return render\_template('info.html', posts=posts)

# 10. Espace Administrateur – Gestion des utilisateurs

@app.route('/admin')

def admin():

if 'user\_id' not in session or not session.get('is\_admin'):

flash("Accès refusé.")

return redirect(url\_for('login'))

users = User.query.all()

return render\_template('admin.html', users=users)

@app.route('/admin/edit\_user/<int:user\_id>', methods=['GET', 'POST'])

def edit\_user(user\_id):

if 'user\_id' not in session or not session.get('is\_admin'):

flash("Accès refusé.")

return redirect(url\_for('login'))

user = User.query.get(user\_id)

if not user:

flash("Utilisateur introuvable.")

return redirect(url\_for('admin'))

if request.method == 'POST':

user.prenom = request.form['prenom']

user.nom = request.form['nom']

user.pays = request.form['pays']

user.email = request.form['email']

if 'new\_password' in request.form and request.form['new\_password']:

new\_pw = request.form['new\_password']

if len(new\_pw) < 8:

flash("Le mot de passe doit contenir au moins 8 caractères.")

return redirect(url\_for('edit\_user', user\_id=user.id))

user.password = generate\_password\_hash(new\_pw)

if 'is\_admin' in request.form:

user.is\_admin = True

else:

user.is\_admin = False

db.session.commit()

flash("Informations de l'utilisateur mises à jour.")

return redirect(url\_for('admin'))

return render\_template('edit\_user.html', user=user)

# 11. Partage d'analyses

@app.route('/share\_analysis/<int:analysis\_id>', methods=['GET', 'POST'])

def share\_analysis(analysis\_id):

if 'user\_id' not in session:

return redirect(url\_for('login'))

analysis = Analysis.query.get(analysis\_id)

if not analysis:

flash("Analyse introuvable.")

return redirect(url\_for('home'))

journal = Journal.query.get(analysis.journal\_id)

if not journal or journal.owner.id != session['user\_id']:

flash("Accès non autorisé.")

return redirect(url\_for('home'))

if request.method == 'POST':

shared\_with = request.form['shared\_with']

share = AnalysisShare(

analysis\_id=analysis.id,

shared\_by\_user\_id=session['user\_id'],

shared\_with=shared\_with

)

db.session.add(share)

db.session.commit()

flash("Analyse partagée avec succès.")

return redirect(url\_for('analysis\_detail', analysis\_id=analysis.id))

return render\_template('share\_analysis.html', analysis=analysis)

@app.route('/community')

def community():

if 'user\_id' not in session:

return redirect(url\_for('login'))

shares = AnalysisShare.query.filter\_by(shared\_with='all').all()

return render\_template('community.html', shares=shares)

@app.route('/my\_shares')

def my\_shares():

if 'user\_id' not in session:

return redirect(url\_for('login'))

user = User.query.get(session['user\_id'])

shares = AnalysisShare.query.filter(AnalysisShare.shared\_with != 'all',

AnalysisShare.shared\_with == user.email).all()

return render\_template('my\_shares.html', shares=shares)

@app.route('/share\_detail/<int:share\_id>', methods=['GET', 'POST'])

def share\_detail(share\_id):

if 'user\_id' not in session:

return redirect(url\_for('login'))

share = AnalysisShare.query.get(share\_id)

if not share:

flash("Partage introuvable.")

return redirect(url\_for('community'))

user = User.query.get(session['user\_id'])

if share.shared\_with != 'all' and share.shared\_with != user.email:

flash("Accès refusé.")

return redirect(url\_for('community'))

if request.method == 'POST':

comment\_text = request.form['comment']

new\_comment = AnalysisShareComment(

share\_id=share.id,

user\_id=user.id,

comment=comment\_text

)

db.session.add(new\_comment)

db.session.commit()

flash("Commentaire ajouté.")

return redirect(url\_for('share\_detail', share\_id=share.id))

return render\_template('share\_detail.html', share=share)

# 12. Service pour servir les fichiers uploadés

@app.route('/uploads/<filename>')

def uploaded\_file(filename):

return send\_from\_directory(app.config['UPLOAD\_FOLDER'], filename)

if \_\_name\_\_ == '\_\_main\_\_':

with app.app\_context():

db.create\_all()

app.run(debug=True)

**2. config.py**

class Config:

SECRET\_KEY = 'votre\_secret\_key'

SQLALCHEMY\_DATABASE\_URI = 'sqlite:///trading\_journal.db'

SQLALCHEMY\_TRACK\_MODIFICATIONS = False

**3. requirements.txt**

Flask==2.0.1

Flask-SQLAlchemy==2.5.1

Werkzeug==2.0.1

**4. static/css/style.css**

/\* Arrière-plan animé avec un dégradé dynamique \*/

body {

background: linear-gradient(-45deg, #ee7752, #e73c7e, #23a6d5, #23d5ab);

background-size: 400% 400%;

animation: gradientBG 15s ease infinite;

}

@keyframes gradientBG {

0% { background-position: 0% 50%; }

50% { background-position: 100% 50%; }

100% { background-position: 0% 50%; }

}

/\* Thème sombre \*/

.dark-theme {

filter: brightness(0.85);

}

/\* Animations & transitions \*/

.card, .list-group-item {

border-radius: 10px;

transition: transform 0.3s;

}

.card:hover, .list-group-item:hover {

transform: scale(1.02);

}

.btn {

border-radius: 5px;

transition: transform 0.2s, box-shadow 0.2s;

}

.btn:hover {

transform: scale(1.05);

box-shadow: 0 4px 8px rgba(0,0,0,0.2);

}

**5. templates/base.html**

<!DOCTYPE html>

<html lang="fr">

<head>

<meta charset="UTF-8">

<title>{% block title %}Trading Journal{% endblock %}</title>

<!-- Bootstrap CSS -->

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

<!-- Animate.css -->

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/animate.css/4.1.1/animate.min.css"/>

<!-- CSS personnalisé -->

<link rel="stylesheet" href="{{ url\_for('static', filename='css/style.css') }}">

</head>

{% if session.get('theme') == 'dark' %}

<body class="dark-theme">

{% else %}

<body>

{% endif %}

{% include 'navbar.html' %}

<div class="container mt-4">

{% with messages = get\_flashed\_messages() %}

{% if messages %}

{% for message in messages %}

<div class="alert alert-info">{{ message }}</div>

{% endfor %}

{% endif %}

{% endwith %}

{% block content %}{% endblock %}

</div>

<!-- Scripts Bootstrap -->

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>

<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"></script>

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

<!-- Chat.min.js via CDN si nécessaire -->

<script src="https://cdn.example.com/chat.min.js"></script>

{% block scripts %}{% endblock %}

</body>

</html>

**6. templates/navbar.html**

<nav class="navbar navbar-expand-lg navbar-light bg-light">

<a class="navbar-brand" href="{{ url\_for('home') }}">Trading Journal</a>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarNav">

<ul class="navbar-nav mr-auto">

{% if session.get('user\_id') %}

<li class="nav-item"><a class="nav-link" href="{{ url\_for('assistance') }}">Assistance</a></li>

<li class="nav-item"><a class="nav-link" href="{{ url\_for('info') }}">Info</a></li>

<li class="nav-item"><a class="nav-link" href="{{ url\_for('parametres') }}">Paramètres</a></li>

<li class="nav-item"><a class="nav-link" href="{{ url\_for('community') }}">Communauté</a></li>

<li class="nav-item"><a class="nav-link" href="{{ url\_for('my\_shares') }}">Mes Partages</a></li>

{% if session.get('is\_admin') %}

<li class="nav-item"><a class="nav-link" href="{{ url\_for('admin') }}">Admin</a></li>

<li class="nav-item"><a class="nav-link" href="{{ url\_for('admin\_assistance') }}">Assistance Admin</a></li>

{% endif %}

<li class="nav-item"><a class="nav-link" href="{{ url\_for('logout') }}">Déconnexion</a></li>

{% else %}

<li class="nav-item"><a class="nav-link" href="{{ url\_for('login') }}">Connexion</a></li>

<li class="nav-item"><a class="nav-link" href="{{ url\_for('register') }}">Inscription</a></li>

{% endif %}

</ul>

</div>

</nav>

**7. templates/register.html**

{% extends "base.html" %}

{% block title %}Inscription - Trading Journal{% endblock %}

{% block content %}

<h2>Inscription</h2>

<form method="POST">

<div class="form-group">

<label for="prenom">Prénom</label>

<input type="text" class="form-control" name="prenom" required>

</div>

<div class="form-group">

<label for="nom">Nom</label>

<input type="text" class="form-control" name="nom" required>

</div>

<div class="form-group">

<label for="pays">Pays</label>

<input type="text" class="form-control" name="pays" required>

</div>

<div class="form-group">

<label for="email">Email</label>

<input type="email" class="form-control" name="email" required>

</div>

<div class="form-group">

<label for="password">Mot de passe</label>

<input type="password" class="form-control" name="password" required>

</div>

<button type="submit" class="btn btn-primary">S'inscrire</button>

</form>

{% endblock %}

**8. templates/login.html**

{% extends "base.html" %}

{% block title %}Connexion - Trading Journal{% endblock %}

{% block content %}

<h2>Connexion</h2>

<form method="POST">

<div class="form-group">

<label for="email">Email</label>

<input type="email" class="form-control" name="email" required>

</div>

<div class="form-group">

<label for="password">Mot de passe</label>

<input type="password" class="form-control" name="password" required>

</div>

<button type="submit" class="btn btn-primary">Se connecter</button>

</form>

{% endblock %}

**9. templates/home.html**

{% extends "base.html" %}

{% block title %}Accueil - Trading Journal{% endblock %}

{% block content %}

<div class="d-flex justify-content-between align-items-center mb-4">

<h2>Bienvenue {{ session.get('user\_name') }} !</h2>

<a href="{{ url\_for('create\_journal') }}" class="btn btn-success">Créer un Journal</a>

</div>

<div class="row">

{% if journals %}

{% for journal in journals %}

<div class="col-md-4">

<div class="card card-journal mb-3">

<div class="card-body">

<h5 class="card-title">{{ journal.nom }}</h5>

<p class="card-text">Capital : {{ journal.capital\_initial }} {{ journal.devise }}</p>

<a href="{{ url\_for('dashboard', journal\_id=journal.id) }}" class="btn btn-primary btn-sm">Dashboard</a>

<a href="{{ url\_for('trades', journal\_id=journal.id) }}" class="btn btn-info btn-sm">Trades</a>

<a href="{{ url\_for('analyses', journal\_id=journal.id) }}" class="btn btn-warning btn-sm">Analyses</a>

</div>

</div>

</div>

{% endfor %}

{% else %}

<p>Aucun journal trouvé.</p>

{% endif %}

</div>

{% endblock %}

**10. templates/create\_journal.html**

{% extends "base.html" %}

{% block title %}Créer un Journal - Trading Journal{% endblock %}

{% block content %}

<h2>Créer un Journal</h2>

<form method="POST">

<div class="form-group">

<label for="nom">Nom du Journal</label>

<input type="text" class="form-control" name="nom" required>

</div>

<div class="form-group">

<label for="capital\_initial">Capital Initial</label>

<input type="number" step="0.01" class="form-control" name="capital\_initial" required>

</div>

<div class="form-group">

<label for="devise">Devise</label>

<select class="form-control" name="devise" required>

<option value="EUR">EUR</option>

<option value="USD">USD</option>

<option value="GBP">GBP</option>

<option value="JPY">JPY</option>

</select>

</div>

<button type="submit" class="btn btn-primary">Créer</button>

</form>

{% endblock %}

**11. templates/dashboard.html**

{% extends "base.html" %}

{% block title %}Dashboard - Trading Journal{% endblock %}

{% block content %}

<h2>Dashboard pour {{ journal.nom }}</h2>

<p>Capital Initial : {{ journal.capital\_initial }} {{ journal.devise }}</p>

<p><strong>Solde actuel :</strong> {{ stats.solde }} {{ journal.devise }}</p>

<div class="row">

<div class="col-md-6 mb-4">

<canvas id="chart1"></canvas>

</div>

<div class="col-md-6 mb-4">

<canvas id="chart2"></canvas>

</div>

</div>

<div class="row">

<div class="col-md-6 mb-4">

<canvas id="chart3"></canvas>

</div>

<div class="col-md-6 mb-4">

<canvas id="chart4"></canvas>

</div>

</div>

<h4>Plateformes Liées</h4>

{% if platforms %}

<ul class="list-group mb-4">

{% for plat in platforms %}

<li class="list-group-item">

<strong>{{ plat.plateforme }}</strong> - {{ plat.identifiant }} ({{ plat.details }})

</li>

{% endfor %}

</ul>

{% else %}

<p>Aucune plateforme liée.</p>

{% endif %}

<div class="mt-4">

<a href="{{ url\_for('trades', journal\_id=journal.id) }}" class="btn btn-info">Créer/Consulter Trades</a>

<a href="{{ url\_for('analyses', journal\_id=journal.id) }}" class="btn btn-warning">Créer/Consulter Analyses</a>

<a href="{{ url\_for('link\_platform', journal\_id=journal.id) }}" class="btn btn-secondary">Lier Plateformes</a>

</div>

{% endblock %}

{% block scripts %}

<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

<script>

var ctx1 = document.getElementById('chart1').getContext('2d');

var chart1 = new Chart(ctx1, {

type: 'bar',

data: {

labels: {{ stats.mois|tojson }},

datasets: [{

label: 'Gains (€)',

data: {{ stats.gains\_per\_month|tojson }},

backgroundColor: 'rgba(54, 162, 235, 0.6)'

}]

}

});

var ctx2 = document.getElementById('chart2').getContext('2d');

var chart2 = new Chart(ctx2, {

type: 'line',

data: {

labels: {{ stats.mois|tojson }},

datasets: [{

label: 'Nombre de Trades',

data: {{ stats.trades\_count|tojson }},

borderColor: 'rgba(255, 99, 132, 0.6)',

fill: false

}]

}

});

var ctx3 = document.getElementById('chart3').getContext('2d');

var chart3 = new Chart(ctx3, {

type: 'doughnut',

data: {

labels: ['Solde initial', 'Gains/Pertes'],

datasets: [{

data: [{{ journal.capital\_initial }}, {{ stats.solde - journal.capital\_initial }}],

backgroundColor: ['rgba(75, 192, 192, 0.6)', 'rgba(255, 205, 86, 0.6)']

}]

}

});

var ctx4 = document.getElementById('chart4').getContext('2d');

var chart4 = new Chart(ctx4, {

type: 'pie',

data: {

labels: {{ stats.mois|tojson }},

datasets: [{

data: {{ stats.gains\_per\_month|tojson }},

backgroundColor: ['rgba(54, 162, 235, 0.6)','rgba(255, 99, 132, 0.6)','rgba(255, 205, 86, 0.6)','rgba(75, 192, 192, 0.6)']

}]

}

});

</script>

{% endblock %}

**12. templates/trades.html**

{% extends "base.html" %}

{% block title %}Trades - Trading Journal{% endblock %}

{% block content %}

<h2>Trades pour {{ journal.nom }}</h2>

<form method="POST" enctype="multipart/form-data">

<h4>Ajouter un Trade</h4>

<div class="form-row">

<div class="form-group col-md-4">

<label for="date\_debut">Date de début</label>

<input type="date" class="form-control" name="date\_debut" required>

</div>

<div class="form-group col-md-4">

<label for="heure\_debut">Heure de début</label>

<input type="time" class="form-control" name="heure\_debut" required>

</div>

<div class="form-group col-md-4">

<label for="session">Session</label>

<select class="form-control" name="session" required>

<option value="Londres">Londres</option>

<option value="New York">New York</option>

<option value="Tokyo">Tokyo</option>

<option value="Sydney">Sydney</option>

<option value="Crypto 24/7">Crypto 24/7</option>

</select>

</div>

</div>

<div class="form-row">

<div class="form-group col-md-4">

<label for="instrument">Instrument</label>

<input type="text" class="form-control" name="instrument" required>

</div>

<div class="form-group col-md-4">

<label for="position">Position</label>

<select class="form-control" name="position" required>

<option value="achat">Achat</option>

<option value="vente">Vente</option>

</select>

</div>

<div class="form-group col-md-4">

<label for="prix\_entree">Prix d'entrée</label>

<input type="number" step="0.01" class="form-control" name="prix\_entree" required>

</div>

</div>

<div class="form-row">

<div class="form-group col-md-6">

<label for="lot">Lot</label>

<input type="number" step="0.01" class="form-control" name="lot" required>

</div>

<div class="form-group col-md-6">

<label for="risk\_reward">Risk/Reward</label>

<input type="number" step="0.01" class="form-control" name="risk\_reward" required>

</div>

</div>

<div class="form-row">

<div class="form-group col-md-6">

<label for="lower\_time\_frame">Lower Time Frame</label>

<select class="form-control" name="lower\_time\_frame" required>

<option value="M1">M1</option>

<option value="M5">M5</option>

<option value="M15">M15</option>

<option value="M30">M30</option>

<option value="Custom">Custom</option>

</select>

<input type="text" class="form-control mt-2" name="custom\_lower\_time\_frame" placeholder="Entrez votre valeur custom si 'Custom'">

</div>

<div class="form-group col-md-6">

<label for="higher\_time\_frame">Higher Time Frame</label>

<select class="form-control" name="higher\_time\_frame" required>

<option value="H1">H1</option>

<option value="H4">H4</option>

<option value="D1">D1</option>

<option value="W1">W1</option>

<option value="Custom">Custom</option>

</select>

<input type="text" class="form-control mt-2" name="custom\_higher\_time\_frame" placeholder="Entrez votre valeur custom si 'Custom'">

</div>

</div>

<div class="form-group">

<label for="commentaires">Commentaires</label>

<textarea class="form-control" name="commentaires"></textarea>

</div>

<div class="form-group">

<label for="capture">Capture d'écran</label>

<input type="file" class="form-control-file" name="capture">

</div>

<hr>

<h4>Terminer le Trade (optionnel)</h4>

<div class="form-row">

<div class="form-group col-md-4">

<label for="date\_fin">Date de fin</label>

<input type="date" class="form-control" name="date\_fin">

</div>

<div class="form-group col-md-4">

<label for="heure\_fin">Heure de fin</label>

<input type="time" class="form-control" name="heure\_fin">

</div>

<div class="form-group col-md-4">

<label for="prix\_sortie">Prix de sortie</label>

<input type="number" step="0.01" class="form-control" name="prix\_sortie">

</div>

</div>

<button type="submit" class="btn btn-primary">Enregistrer le Trade</button>

</form>

<hr>

<h3>Trades en cours</h3>

{% if trades\_encours %}

<ul class="list-group">

{% for trade in trades\_encours %}

<li class="list-group-item">

<a href="{{ url\_for('trade\_detail', trade\_id=trade.id) }}">Trade n°{{ trade.id }} - {{ trade.instrument }} ({{ trade.position }})</a>

</li>

{% endfor %}

</ul>

{% else %}

<p>Aucun trade en cours.</p>

{% endif %}

<h3>Trades terminés</h3>

{% if trades\_termine %}

<ul class="list-group">

{% for trade in trades\_termine %}

<li class="list-group-item">

<a href="{{ url\_for('trade\_detail', trade\_id=trade.id) }}">Trade n°{{ trade.id }} - {{ trade.instrument }} ({{ trade.position }})</a>

</li>

{% endfor %}

</ul>

{% else %}

<p>Aucun trade terminé.</p>

{% endif %}

{% endblock %}

**13. templates/trade\_detail.html**

{% extends "base.html" %}

{% block title %}Détail du Trade - Trading Journal{% endblock %}

{% block content %}

<h2>Détail du Trade n°{{ trade.id }}</h2>

<ul class="list-group">

<li class="list-group-item"><strong>Date de début :</strong> {{ trade.date\_debut.strftime('%Y-%m-%d %H:%M') }}</li>

<li class="list-group-item"><strong>Date de fin :</strong> {{ trade.date\_fin.strftime('%Y-%m-%d %H:%M') if trade.date\_fin else "En cours" }}</li>

<li class="list-group-item"><strong>Session :</strong> {{ trade.session }}</li>

<li class="list-group-item"><strong>Instrument :</strong> {{ trade.instrument }}</li>

<li class="list-group-item"><strong>Position :</strong> {{ trade.position }}</li>

<li class="list-group-item"><strong>Prix d'entrée :</strong> {{ trade.prix\_entree }}</li>

<li class="list-group-item"><strong>Prix de sortie :</strong> {{ trade.prix\_sortie if trade.prix\_sortie else "Non renseigné" }}</li>

<li class="list-group-item"><strong>Lot :</strong> {{ trade.lot }}</li>

<li class="list-group-item"><strong>Risk/Reward :</strong> {{ trade.risk\_reward }}</li>

<li class="list-group-item"><strong>Résultat :</strong> {{ trade.resultat if trade.resultat is not none else "Calcul en attente" }}</li>

<li class="list-group-item"><strong>Pourcentage :</strong> {{ trade.pourcentage if trade.pourcentage is not none else "Calcul en attente" }}%</li>

<li class="list-group-item"><strong>Commentaires :</strong> {{ trade.commentaires if trade.commentaires else "Aucun commentaire" }}</li>

{% if trade.capture %}

<li class="list-group-item">

<strong>Capture :</strong>

<a href="{{ url\_for('uploaded\_file', filename=trade.capture) }}" target="\_blank">

<img src="{{ url\_for('uploaded\_file', filename=trade.capture) }}" alt="Capture" style="max-width: 200px;">

</a>

</li>

{% endif %}

</ul>

<hr>

<h3>Mettre à jour le Trade (Terminer)</h3>

<form method="POST">

<div class="form-row">

<div class="form-group col-md-4">

<label for="date\_fin">Date de fin</label>

<input type="date" class="form-control" name="date\_fin" required>

</div>

<div class="form-group col-md-4">

<label for="heure\_fin">Heure de fin</label>

<input type="time" class="form-control" name="heure\_fin" required>

</div>

<div class="form-group col-md-4">

<label for="prix\_sortie">Prix de sortie</label>

<input type="number" step="0.01" class="form-control" name="prix\_sortie" required>

</div>

</div>

<button type="submit" class="btn btn-primary">Mettre à jour</button>

</form>

<a href="{{ url\_for('trades', journal\_id=trade.journal\_id) }}" class="btn btn-secondary mt-3">Retour aux Trades</a>

{% endblock %}

**14. templates/analyses.html**

{% extends "base.html" %}

{% block title %}Analyses - Trading Journal{% endblock %}

{% block content %}

<h2>Analyses pour {{ journal.nom }}</h2>

<form method="POST" enctype="multipart/form-data">

<h4>Ajouter une Analyse</h4>

<div class="form-group">

<label for="titre">Titre</label>

<input type="text" class="form-control" name="titre" required>

</div>

<div class="form-group">

<label for="contenu">Contenu</label>

<textarea class="form-control" name="contenu" rows="5" required></textarea>

</div>

<div class="form-group">

<label for="image">Image (facultatif)</label>

<input type="file" class="form-control-file" name="image">

</div>

<button type="submit" class="btn btn-primary">Ajouter l'Analyse</button>

</form>

<hr>

<h3>Liste des Analyses</h3>

{% if analyses %}

<ul class="list-group">

{% for analysis in analyses %}

<li class="list-group-item d-flex justify-content-between align-items-center">

<a href="{{ url\_for('analysis\_detail', analysis\_id=analysis.id) }}">{{ analysis.titre }}</a>

<a href="{{ url\_for('share\_analysis', analysis\_id=analysis.id) }}" class="btn btn-sm btn-success">Partager</a>

</li>

{% endfor %}

</ul>

{% else %}

<p>Aucune analyse enregistrée.</p>

{% endif %}

{% endblock %}

**15. templates/analysis\_detail.html**

{% extends "base.html" %}

{% block title %}Détail de l'Analyse - Trading Journal{% endblock %}

{% block content %}

<h2>{{ analysis.titre }}</h2>

<p>Date de création : {{ analysis.date\_creation }}</p>

<p>{{ analysis.contenu }}</p>

{% if analysis.image %}

<img src="{{ url\_for('uploaded\_file', filename=analysis.image) }}" alt="Image de l'analyse" class="img-fluid">

{% endif %}

<a href="{{ url\_for('analyses', journal\_id=analysis.journal\_id) }}" class="btn btn-secondary mt-3">Retour aux Analyses</a>

{% endblock %}

**16. templates/link\_platform.html**

{% extends "base.html" %}

{% block title %}Liaison de Plateformes - Trading Journal{% endblock %}

{% block content %}

<h2>Lier une plateforme pour {{ journal.nom }}</h2>

<form method="POST">

<div class="form-group">

<label for="plateforme">Plateforme</label>

<select class="form-control" name="plateforme" required>

<option value="MT4">MT4</option>

<option value="MT5">MT5</option>

<option value="TradingView">TradingView</option>

</select>

</div>

<div class="form-group">

<label for="identifiant">Identifiant</label>

<input type="text" class="form-control" name="identifiant" required>

</div>

<div class="form-group">

<label for="details">Détails</label>

<input type="text" class="form-control" name="details">

</div>

<button type="submit" class="btn btn-primary">Lier la plateforme</button>

</form>

<hr>

<h3>Plateformes liées</h3>

{% if links %}

<ul class="list-group">

{% for link in links %}

<li class="list-group-item">

{{ link.plateforme }} - {{ link.identifiant }} - {{ link.details }}

</li>

{% endfor %}

</ul>

{% else %}

<p>Aucune plateforme liée.</p>

{% endif %}

{% endblock %}

**17. templates/parametres.html**

{% extends "base.html" %}

{% block title %}Paramètres - Trading Journal{% endblock %}

{% block content %}

<h2>Paramètres de l'utilisateur</h2>

<div class="mb-4">

<h4>Modifier le mot de passe</h4>

<form method="POST">

<div class="form-group">

<label for="new\_password">Nouveau mot de passe</label>

<input type="password" class="form-control" name="new\_password" required>

</div>

<button type="submit" class="btn btn-primary">Mettre à jour le mot de passe</button>

</form>

</div>

<div>

<h4>Modifier le thème</h4>

<form method="POST">

<div class="form-group">

<label for="theme">Choisir le thème</label>

<select class="form-control" name="theme" required>

<option value="light" {% if session.get('theme', 'light') == 'light' %}selected{% endif %}>Clair</option>

<option value="dark" {% if session.get('theme') == 'dark' %}selected{% endif %}>Sombre</option>

</select>

</div>

<button type="submit" class="btn btn-primary">Mettre à jour le thème</button>

</form>

</div>

{% endblock %}

**18. templates/assistance.html**

{% extends "base.html" %}

{% block title %}Assistance - Trading Journal{% endblock %}

{% block content %}

<!-- Ce template redirige vers "my\_assistance.html" -->

{% endblock %}

**19. templates/my\_assistance.html**

{% extends "base.html" %}

{% block title %}Mes Messages d'Assistance - Trading Journal{% endblock %}

{% block content %}

<h2>Mes Messages d'Assistance</h2>

<form method="POST">

<div class="form-group">

<label for="sujet">Sujet</label>

<input type="text" class="form-control" name="sujet" required>

</div>

<div class="form-group">

<label for="message">Message</label>

<textarea class="form-control" name="message" rows="5" required></textarea>

</div>

<button type="submit" class="btn btn-primary">Envoyer</button>

</form>

<hr>

<h4>Messages Envoyés</h4>

{% if user\_msgs %}

<ul class="list-group">

{% for m in user\_msgs %}

<li class="list-group-item">

<strong>{{ m.sujet }}</strong> - {{ m.date\_creation }}

<a href="{{ url\_for('my\_conversation', msg\_id=m.id) }}" class="btn btn-sm btn-info float-right">Ouvrir</a>

</li>

{% endfor %}

</ul>

{% else %}

<p>Aucun message d'assistance.</p>

{% endif %}

{% endblock %}

**20. templates/my\_conversation.html**

{% extends "base.html" %}

{% block title %}Ma Conversation - Trading Journal{% endblock %}

{% block content %}

<h2>Conversation pour : {{ msg.sujet }}</h2>

<div class="card mb-3">

<div class="card-body">

<p><strong>Message initial :</strong> {{ msg.message }}</p>

<p><strong>Date :</strong> {{ msg.date\_creation }}</p>

</div>

</div>

{% if msg.replies %}

<h4>Réponses :</h4>

{% for reply in msg.replies %}

<div class="card mb-2">

<div class="card-body">

<p>{{ reply.reply\_message }}</p>

<small>Envoyé le {{ reply.date\_creation }} par {{ reply.sender }}</small>

</div>

</div>

{% endfor %}

{% endif %}

<form method="POST">

<div class="form-group">

<label for="reply">Votre réponse</label>

<textarea class="form-control" name="reply" rows="4" required></textarea>

</div>

<button type="submit" class="btn btn-primary">Envoyer la réponse</button>

</form>

<a href="{{ url\_for('assistance') }}" class="btn btn-secondary mt-3">Retour</a>

{% endblock %}

**21. templates/admin\_assistance.html**

{% extends "base.html" %}

{% block title %}Assistance Admin - Trading Journal{% endblock %}

{% block content %}

<h2>Messages d'Assistance</h2>

{% if messages %}

<ul class="list-group">

{% for msg in messages %}

<li class="list-group-item">

<strong>{{ msg.sujet }}</strong> – {{ msg.date\_creation }}<br>

<a href="{{ url\_for('conversation\_detail', msg\_id=msg.id) }}" class="btn btn-sm btn-primary mt-2">Voir/Répondre</a>

</li>

{% endfor %}

</ul>

{% else %}

<p>Aucun message d'assistance reçu.</p>

{% endif %}

{% endblock %}

**22. templates/conversation\_detail.html**

{% extends "base.html" %}

{% block title %}Conversation Assistance - Trading Journal{% endblock %}

{% block content %}

<h2>Conversation pour : {{ msg.sujet }}</h2>

<p>

<strong>Coordonnées de l'envoyeur :</strong>

{{ msg.user.prenom }} {{ msg.user.nom }} - {{ msg.user.email }}

<a href="{{ url\_for('edit\_user', user\_id=msg.user.id) }}" class="btn btn-sm btn-secondary">Modifier ce compte</a>

</p>

<div class="card mb-3">

<div class="card-body">

<p><strong>Message initial :</strong> {{ msg.message }}</p>

<p><strong>Date :</strong> {{ msg.date\_creation }}</p>

</div>

</div>

{% if msg.replies %}

<h4>Réponses :</h4>

{% for reply in msg.replies %}

<div class="card mb-2">

<div class="card-body">

<p>{{ reply.reply\_message }}</p>

<small>Envoyé le {{ reply.date\_creation }} par {{ reply.sender }}</small>

</div>

</div>

{% endfor %}

{% endif %}

<form method="POST">

<div class="form-group">

<label for="reply">Votre réponse (Admin)</label>

<textarea class="form-control" name="reply" rows="4" required></textarea>

</div>

<button type="submit" class="btn btn-primary">Envoyer la réponse</button>

</form>

<a href="{{ url\_for('admin\_assistance') }}" class="btn btn-secondary mt-3">Retour</a>

{% endblock %}

**23. templates/info.html**

{% extends "base.html" %}

{% block title %}Info - Trading Journal{% endblock %}

{% block content %}

<h2>Actualités</h2>

{% if session.get('is\_admin') %}

<h4>Créer une Publication</h4>

<form method="POST" enctype="multipart/form-data">

<div class="form-group">

<label for="titre">Titre</label>

<input type="text" class="form-control" name="titre" required>

</div>

<div class="form-group">

<label for="contenu">Contenu</label>

<textarea class="form-control" name="contenu" rows="5" required></textarea>

</div>

<div class="form-group">

<label for="media">Fichier Média (audio/vidéo/image, facultatif)</label>

<input type="file" class="form-control-file" name="media">

</div>

<button type="submit" class="btn btn-primary">Publier</button>

</form>

<hr>

{% endif %}

<div class="d-flex flex-wrap">

{% if posts %}

{% for post in posts %}

<div class="card m-2" style="width: 18rem; position: relative; background: url('{{ url\_for('uploaded\_file', filename=post.media) }}') no-repeat center center; background-size: cover;">

<div class="card-body" style="background: rgba(0,0,0,0.5); color: white;">

<h5 class="card-title">{{ post.titre }}</h5>

<p class="card-text">{{ post.contenu }}</p>

<small>{{ post.date\_creation }}</small>

</div>

</div>

{% endfor %}

{% else %}

<p>Aucune publication disponible.</p>

{% endif %}

</div>

{% endblock %}

**24. templates/admin.html**

{% extends "base.html" %}

{% block title %}Espace Administrateur - Trading Journal{% endblock %}

{% block content %}

<h2>Gestion des Utilisateurs</h2>

<table class="table table-bordered">

<thead>

<tr>

<th>ID</th>

<th>Nom</th>

<th>Email</th>

<th>Rôle</th>

<th>Action</th>

</tr>

</thead>

<tbody>

{% for user in users %}

<tr>

<td>{{ user.id }}</td>

<td>{{ user.prenom }} {{ user.nom }}</td>

<td>{{ user.email }}</td>

<td>{% if user.is\_admin %}Administrateur{% else %}Utilisateur{% endif %}</td>

<td>

<a href="{{ url\_for('edit\_user', user\_id=user.id) }}" class="btn btn-sm btn-primary">Modifier</a>

</td>

</tr>

{% endfor %}

</tbody>

</table>

{% endblock %}

**25. templates/edit\_user.html**

{% extends "base.html" %}

{% block title %}Modifier Utilisateur - Trading Journal{% endblock %}

{% block content %}

<h2>Modifier les informations de l'utilisateur</h2>

<form method="POST">

<div class="form-group">

<label for="prenom">Prénom</label>

<input type="text" class="form-control" name="prenom" value="{{ user.prenom }}" required>

</div>

<div class="form-group">

<label for="nom">Nom</label>

<input type="text" class="form-control" name="nom" value="{{ user.nom }}" required>

</div>

<div class="form-group">

<label for="pays">Pays</label>

<input type="text" class="form-control" name="pays" value="{{ user.pays }}">

</div>

<div class="form-group">

<label for="email">Email</label>

<input type="email" class="form-control" name="email" value="{{ user.email }}" required>

</div>

<div class="form-group">

<label for="new\_password">Nouveau mot de passe (laisser vide si inchangé)</label>

<input type="password" class="form-control" name="new\_password">

</div>

<div class="form-check">

<input class="form-check-input" type="checkbox" name="is\_admin" id="is\_admin"

{% if user.is\_admin %}checked{% endif %}>

<label class="form-check-label" for="is\_admin">

Rendre ce compte administrateur

</label>

</div>

<button type="submit" class="btn btn-primary mt-3">Enregistrer les modifications</button>

</form>

<a href="{{ url\_for('admin') }}" class="btn btn-secondary mt-3">Retour</a>

{% endblock %}

**26. templates/share\_analysis.html**

{% extends "base.html" %}

{% block title %}Partager une Analyse - Trading Journal{% endblock %}

{% block content %}

<h2>Partager l'Analyse : {{ analysis.titre }}</h2>

<form method="POST">

<div class="form-group">

<label for="shared\_with">Partager avec :</label>

<input type="text" class="form-control" name="shared\_with" placeholder="Tapez un email ou 'all' pour la communauté" required>

</div>

<button type="submit" class="btn btn-primary">Partager</button>

</form>

{% endblock %}

**27. templates/community.html**

{% extends "base.html" %}

{% block title %}Communauté - Trading Journal{% endblock %}

{% block content %}

<h2>Analyses Partagées avec la Communauté</h2>

{% if shares %}

<ul class="list-group">

{% for s in shares %}

<li class="list-group-item">

<strong>{{ s.analysis.titre }}</strong> (partagé le {{ s.date\_shared }} par l'utilisateur ID {{ s.shared\_by\_user\_id }})

<a href="{{ url\_for('share\_detail', share\_id=s.id) }}" class="btn btn-sm btn-info float-right">Voir</a>

</li>

{% endfor %}

</ul>

{% else %}

<p>Aucun partage disponible.</p>

{% endif %}

{% endblock %}

**28. templates/my\_shares.html**

{% extends "base.html" %}

{% block title %}Mes Partages - Trading Journal{% endblock %}

{% block content %}

<h2>Mes Partages Reçus</h2>

{% if shares %}

<ul class="list-group">

{% for s in shares %}

<li class="list-group-item">

<strong>{{ s.analysis.titre }}</strong> (partagé le {{ s.date\_shared }} par l'utilisateur ID {{ s.shared\_by\_user\_id }})

<a href="{{ url\_for('share\_detail', share\_id=s.id) }}" class="btn btn-sm btn-info float-right">Voir</a>

</li>

{% endfor %}

</ul>

{% else %}

<p>Aucun partage reçu.</p>

{% endif %}

{% endblock %}

**29. templates/share\_detail.html**

{% extends "base.html" %}

{% block title %}Détail du Partage - Trading Journal{% endblock %}

{% block content %}

<h2>Analyse Partagée</h2>

<div class="card mb-3">

<div class="card-body">

<h5>{{ share.analysis.titre }}</h5>

<p>{{ share.analysis.contenu }}</p>

{% if share.analysis.image %}

<img src="{{ url\_for('uploaded\_file', filename=share.analysis.image) }}" alt="Image de l'analyse" class="img-fluid">

{% endif %}

<p><small>Partagé le {{ share.date\_shared }}</small></p>

</div>

</div>

<h4>Commentaires</h4>

{% if share.comments %}

{% for c in share.comments %}

<div class="card mb-2">

<div class="card-body">

<p>{{ c.comment }}</p>

<small>Envoyé par l'utilisateur ID {{ c.user\_id }} le {{ c.date\_creation }}</small>

</div>

</div>

{% endfor %}

{% else %}

<p>Aucun commentaire.</p>

{% endif %}

<hr>

<form method="POST">

<div class="form-group">

<label for="comment">Ajouter un commentaire</label>

<textarea class="form-control" name="comment" rows="3" required></textarea>

</div>

<button type="submit" class="btn btn-primary">Envoyer</button>

</form>

<a href="{{ url\_for('community') }}" class="btn btn-secondary mt-3">Retour</a>

{% endblock %}

**Conclusion**

Cette documentation contient :

* La présentation complète du fonctionnement et des fonctionnalités du site "Journal de Trading Web" (création et gestion de journaux, enregistrement des trades, analyses, dashboard, assistance, publications info, partage d’analyses, et administration avancée).
* L’intégralité de tous les fichiers de code (app.py, config.py, requirements.txt, le CSS, et tous les templates) sans aucune modification supplémentaire.

Vous pouvez copier ce contenu dans un document Word pour le conserver, le présenter à vos associés et disposer d’une référence complète pour votre projet.

N’hésitez pas à me solliciter pour tout complément d’information ou ajustement.