from flask import Flask, render\_template, request, redirect, session, url\_for

import pandas as pd

import os

import qrcode

from io import BytesIO

import base64

from datetime import datetime, timedelta

# Khởi tạo Flask app

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

# Đường dẫn cố định đến file Excel trên máy bạn

# Sau này phải thay đổi vị trí của ổ đĩa để trả kết quả

EXCEL\_PATH = r"C:\Users\Duy To\Inventory\_control\Khuon\_duc.xlsx"

@app.route("/")

def home():

    return redirect("/login")

# Đặt khóa bí mật để dùng session (bắt buộc)

app.secret\_key = "bat-ky-chuoi-bi-mat"

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

def login():

    error = None  # Biến lưu thông báo lỗi nếu có

    if request.method == "POST":

        # 🔍 Lấy mã nhân viên từ form

        ma\_nhan\_vien = request.form.get("employee\_code").strip()

        # 📄 Đọc danh sách nhân viên từ Excel

        df\_nv = pd.read\_excel("Danh\_sach\_nhan\_vien.xlsx", sheet\_name="Sheet1")

        # Kiểm tra cột "Number" tồn tại và mã hợp lệ

        if ma\_nhan\_vien in df\_nv["Number"].astype(str).values:

            # ✅ Nếu mã hợp lệ → lấy tên tương ứng

            ten = df\_nv.loc[df\_nv["Number"].astype(str) == ma\_nhan\_vien, "Tên nhân viên"].values[0]

            # 💾 Lưu mã và tên vào session để dùng toàn bộ phiên làm việc

            session["ma\_nhan\_vien"] = ma\_nhan\_vien

            session["ten\_nhan\_vien"] = ten

            # 👉 Chuyển sang trang dashboard (trang chính sau đăng nhập)

            return redirect(url\_for("dashboard"))

        else:

            # ❌ Mã không hợp lệ

            error = "Mã nhân viên không tồn tại!"

    # 🖼 Hiển thị form login (hoặc lỗi nếu có)

    return render\_template("login.html", error=error)

@app.route("/dashboard")

def dashboard():

    # 🔒 Kiểm tra nếu chưa đăng nhập, quay lại /login

    if "ma\_nhan\_vien" not in session:

        return redirect("/login")

    # ✅ Đã đăng nhập → truyền mã + tên nhân viên vào HTML

    return render\_template(

        "dashboard.html",

        ma\_nhan\_vien=session.get("ma\_nhan\_vien"),

        ten\_nhan\_vien=session.get("ten\_nhan\_vien")

    )

# Sửa lại route /borrow trong app.py

# Cập nhật route /borrow để hiển thị thông tin người mượn

@app.route("/borrow", methods=["GET", "POST"])

def muon\_hang():

    # 🔒 Kiểm tra nếu chưa đăng nhập → quay lại trang login

    if "ma\_nhan\_vien" not in session:

        return redirect("/login")

    thong\_tin = None

    thong\_bao = ""

    mau\_trang\_thai = ""

    icon = ""

    trang\_thai = ""

    tile = ""

    # 🔍 XỬ LÝ KHI NGƯỜI DÙNG TÌM KIẾM SẢN PHẨM

    if request.method == "GET" and request.args.get("code"):

        code = request.args.get("code")

        df = pd.read\_excel(EXCEL\_PATH, sheet\_name="Sheet1")

        if code in df["Code"].astype(str).values:

            # ✅ Nếu code tồn tại → lấy thông tin sản phẩm

            row = df[df["Code"].astype(str) == code].iloc[0].to\_dict()

            thong\_tin = row

            # 🎨 Xử lý hiển thị trạng thái

            if pd.notna(thong\_tin.get("Tình trạng")) and thong\_tin["Tình trạng"] == "Available":

                # ✅ Khả dụng - cho phép mượn

                mau\_trang\_thai = "green"

                icon = "✅"

                trang\_thai = "Khả dụng"

                tile = "100"

            elif pd.notna(thong\_tin.get("Tình trạng")) and thong\_tin["Tình trạng"] == "Unavailable":

                # ❌ Không khả dụng - kiểm tra người mượn

                mau\_trang\_thai = "red"

                icon = "❌"

                tile = "0"

                # Kiểm tra có thông tin người mượn không

                nguoi\_muon = thong\_tin.get("Người mượn", "")

                ngay\_lay\_hang = thong\_tin.get("Ngày lấy hàng", "")

                if pd.notna(nguoi\_muon) and str(nguoi\_muon).strip() and str(nguoi\_muon) != "nan":

                    # Có người mượn

                    if pd.notna(ngay\_lay\_hang) and str(ngay\_lay\_hang) != "nan":

                        trang\_thai = f"Sản phẩm không khả dụng - {nguoi\_muon} đã mượn hàng ngày {ngay\_lay\_hang}"

                    else:

                        trang\_thai = f"Sản phẩm không khả dụng - {nguoi\_muon} đã mượn hàng"

                else:

                    # Không có thông tin người mượn

                    trang\_thai = "Không có mẫu trong kho"

            else:

                # Trạng thái không xác định

                mau\_trang\_thai = "orange"

                icon = "⚠️"

                trang\_thai = "Trạng thái không xác định"

                tile = "50"

        else:

            # ❌ Không tìm thấy sản phẩm

            thong\_bao = "❌ Mã sản phẩm không tồn tại!"

    # 📤 Trả về trang HTML

    return render\_template("borrow.html",

                         thong\_tin=thong\_tin,

                         thong\_bao=thong\_bao,

                         mau\_trang\_thai=mau\_trang\_thai,

                         icon=icon,

                         trang\_thai=trang\_thai,

                         tile=tile)

# Thêm route xử lý xác nhận mượn hàng (từ form POST trong borrow.html)

# Phương án đơn giản hơn - chỉ dùng redirect (không có alert)

@app.route("/muon-xac-nhan", methods=["POST"])

def muon\_xac\_nhan():

    if "ma\_nhan\_vien" not in session:

        return redirect("/login")

    code = request.form.get("code")

    anh\_muon = request.files.get("anh\_muon")

    try:

        # Đọc file Excel sản phẩm và nhân viên

        df = pd.read\_excel(EXCEL\_PATH, sheet\_name="Sheet1")

        mask = df["Code"].astype(str) == str(code)

        df\_nv = pd.read\_excel("Danh\_sach\_nhan\_vien.xlsx", sheet\_name="Sheet1")

        ma\_nhan\_vien = session.get("ma\_nhan\_vien")

        ten\_nguoi\_muon = df\_nv.loc[df\_nv["Number"].astype(str) == ma\_nhan\_vien, "Tên nhân viên"].values[0]

        if mask.any():

            # Cập nhật thông tin

            df.loc[mask, "Tình trạng"] = "Unavailable"

            df.loc[mask, "Ngày lấy hàng"] = datetime.now().strftime("%Y-%m-%d")

            df.loc[mask, "Người mượn"] = ten\_nguoi\_muon

            # Tăng số lần sử dụng

            so\_lan\_cu = df.loc[mask, "Số lần đã sử dụng"].values[0]

            if pd.isna(so\_lan\_cu) or str(so\_lan\_cu) == "nan":

                df.loc[mask, "Số lần đã sử dụng"] = 1

            else:

                df.loc[mask, "Số lần đã sử dụng"] = int(so\_lan\_cu) + 1

            # Xử lý ảnh

            if anh\_muon and anh\_muon.filename:

                upload\_folder = "static/images"

                os.makedirs(upload\_folder, exist\_ok=True)

                filename = f"{code}\_{datetime.now().strftime('%Y%m%d\_%H%M%S')}.jpg"

                filepath = os.path.join(upload\_folder, filename)

                anh\_muon.save(filepath)

            # Lưu file

            df.to\_excel(EXCEL\_PATH, sheet\_name="Sheet1", index=False)

        return redirect("/dashboard")  # Đơn giản chỉ redirect về dashboard

    except Exception as e:

        print(f"Lỗi: {e}")  # In lỗi ra console

        return redirect("/borrow")

# THÊM VÀO APP.PY - TÍNH NĂNG MƯỢN HÀNG BẰNG QR

# Import thêm

import qrcode

from io import BytesIO

import base64

from datetime import datetime, timedelta

# Route tạo QR code cho mượn hàng

@app.route('/create\_borrow\_qr/<ma\_khuon>')

def create\_borrow\_qr(ma\_khuon):

    try:

        # Tìm thông tin khuôn đúc

        khuon\_info = df\_khuon\_duc[df\_khuon\_duc['Mã khuôn'] == ma\_khuon].iloc[0]

        # Tạo QR data với thông tin mượn hàng

        qr\_data = {

            'action': 'borrow',

            'ma\_khuon': ma\_khuon,

            'ten\_khuon': khuon\_info['Tên khuôn'],

            'timestamp': datetime.now().isoformat()

        }

        # Chuyển thành JSON string

        qr\_text = f"BORROW:{ma\_khuon}:{khuon\_info['Tên khuôn']}"

        # Tạo QR code

        qr = qrcode.QRCode(version=1, box\_size=10, border=5)

        qr.add\_data(qr\_text)

        qr.make(fit=True)

        # Tạo hình ảnh QR

        qr\_img = qr.make\_image(fill\_color="black", back\_color="white")

        # Convert sang base64 để hiển thị

        buffer = BytesIO()

        qr\_img.save(buffer, format='PNG')

        qr\_base64 = base64.b64encode(buffer.getvalue()).decode()

        return render\_template('qr\_borrow.html',

                             qr\_code=qr\_base64,

                             ma\_khuon=ma\_khuon,

                             ten\_khuon=khuon\_info['Tên khuôn'],

                             so\_luong=khuon\_info['Số lượng'])

    except Exception as e:

        return f"Lỗi tạo QR: {str(e)}"

# Route xử lý scan QR để mượn

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

def scan\_borrow():

    if request.method == 'POST':

        qr\_data = request.form.get('qr\_data')

        nguoi\_muon = request.form.get('nguoi\_muon')

        so\_luong\_muon = int(request.form.get('so\_luong\_muon', 1))

        try:

            # Parse QR data: "BORROW:MA001:Tên khuôn"

            parts = qr\_data.split(':')

            if parts[0] != 'BORROW':

                return "QR code không hợp lệ!"

            ma\_khuon = parts[1]

            # Kiểm tra khuôn tồn tại

            khuon\_idx = df\_khuon\_duc[df\_khuon\_duc['Mã khuôn'] == ma\_khuon].index

            if len(khuon\_idx) == 0:

                return "Không tìm thấy khuôn!"

            # Kiểm tra số lượng có đủ không

            current\_qty = df\_khuon\_duc.loc[khuon\_idx[0], 'Số lượng']

            if current\_qty < so\_luong\_muon:

                return f"Không đủ số lượng! Hiện có: {current\_qty}"

            # Cập nhật số lượng khuôn

            df\_khuon\_duc.loc[khuon\_idx[0], 'Số lượng'] -= so\_luong\_muon

            # Lưu lại file Excel

            df\_khuon\_duc.to\_excel('Khuon\_duc.xlsx', index=False)

            # Ghi log mượn hàng

            log\_muon = {

                'Thời gian': datetime.now().strftime("%Y-%m-%d %H:%M:%S"),

                'Mã khuôn': ma\_khuon,

                'Tên khuôn': parts[2],

                'Người mượn': nguoi\_muon,

                'Số lượng mượn': so\_luong\_muon,

                'Trạng thái': 'Đang mượn'

            }

            # Thêm vào file log (tạo mới nếu chưa có)

            try:

                df\_log = pd.read\_excel('Log\_muon\_hang.xlsx')

                df\_log = pd.concat([df\_log, pd.DataFrame([log\_muon])], ignore\_index=True)

            except:

                df\_log = pd.DataFrame([log\_muon])

            df\_log.to\_excel('Log\_muon\_hang.xlsx', index=False)

            return render\_template('borrow\_success.html',

                                 ma\_khuon=ma\_khuon,

                                 ten\_khuon=parts[2],

                                 nguoi\_muon=nguoi\_muon,

                                 so\_luong=so\_luong\_muon)

        except Exception as e:

            return f"Lỗi xử lý mượn hàng: {str(e)}"

    return render\_template('scan\_borrow.html')

# Route xem lịch sử mượn hàng

@app.route('/borrow\_history')

def borrow\_history():

    try:

        df\_log = pd.read\_excel('Log\_muon\_hang.xlsx')

        records = df\_log.to\_dict('records')

        return render\_template('borrow\_history.html', records=records)

    except:

        return render\_template('borrow\_history.html', records=[])

@app.route("/return", methods=["GET", "POST"])

def return\_item():

    thong\_bao = ""

    thong\_tin = None

    image\_path = None

    available\_positions = []

    if request.method == "POST":

        code = request.form.get("code", "").strip()

        df = pd.read\_excel(EXCEL\_PATH)

        if code not in df["Code"].astype(str).values:

            thong\_bao = "❌ Mã không tồn tại"

        else:

            row = df[df["Code"].astype(str) == code].iloc[0]

            if row["Tình trạng"] == "Available":

                thong\_bao = "❌ Sản phẩm đang tồn tại trong kho"

            else:

                thong\_tin = row.to\_dict()

                image\_path = row.get("Ảnh mượn", None)

                used\_pos = df[df["Vị trí"].notna()]["Vị trí"].tolist()

                all\_pos = [f"{i}{chr(c)}{j}" for i in range(1, 10) for c in range(ord("A"), ord("Z")+1) for j in range(1, 10)]

                available\_positions = [pos for pos in all\_pos if pos not in used\_pos]

    return render\_template("return.html", thong\_bao=thong\_bao, thong\_tin=thong\_tin, image\_path=image\_path, available\_positions=available\_positions)

@app.route("/return/confirm", methods=["POST"])

def return\_confirm():

    code = request.form.get("code")

    vi\_tri = request.form.get("vi\_tri").strip().upper()

    anh\_tra = request.files.get("anh\_tra")

    df = pd.read\_excel(EXCEL\_PATH)

    # Kiểm tra vị trí

    if vi\_tri in df["Vị trí"].astype(str).values:

        return "❌ Đã hết chỗ"  # Có thể render ra template lỗi tùy ý

    mask = df["Code"].astype(str) == code

    if not mask.any():

        return "❌ Mã không tồn tại"

    # Lưu ảnh

    folder = "static/images\_return"

    os.makedirs(folder, exist\_ok=True)

    img\_name = f"{code}\_{vi\_tri}.jpg"

    img\_path = os.path.join(folder, img\_name)

    anh\_tra.save(img\_path)

    # Cập nhật Excel

    now = datetime.now().strftime("%Y-%m-%d")

    df.loc[mask, "Tình trạng"] = "Available"

    df.loc[mask, "Vị trí"] = vi\_tri

    df.loc[mask, "Ngày trả"] = now

    df.loc[mask, "Người trả"] = session.get("ten\_nhan\_vien")  # Nếu dùng session login

    df.loc[mask, "Ảnh trả"] = img\_name

    df.to\_excel(EXCEL\_PATH, index=False)

    # Hiện chi tiết sản phẩm đã trả

    thong\_tin = df[mask].iloc[0].to\_dict()

    return render\_template("return\_success.html", thong\_tin=thong\_tin)

@app.route("/inventory")

def show\_inventory():

    # 🔒 Kiểm tra nếu chưa đăng nhập thì chuyển về trang login

    if "ma\_nhan\_vien" not in session:

        return redirect("/login")

    # 📄 Đọc dữ liệu từ Excel

    df = pd.read\_excel(EXCEL\_PATH, sheet\_name="Sheet1")

# Lấy thông tin phân trang từ URL (?page=1, 2, ...)

    page = int(request.args.get("page", 1))

    per\_page = 15

    start = (page - 1) \* per\_page

    end = start + per\_page

    # Cắt dữ liệu theo trang

    df\_page = df.iloc[start:end]

    # Chuyển thành list dict để truyền sang HTML

    data = df\_page.to\_dict(orient="records")

    columns = df.columns.tolist()

    # Tính tổng số trang

    total\_pages = (len(df) + per\_page - 1) // per\_page

    return render\_template(

        "inventory.html",

        data=data,

        columns=columns,

        current\_page=page,

        total\_pages=total\_pages

    )

# 📦 ROUTE mới để cập nhật "Ngày nhập kho" theo mã hàng

#        @app.route("/nhap-ngay", methods=["GET", "POST"])

#       def nhap\_ngay\_nhap\_kho():

#            # Biến điều khiển hiển thị form nhập ngày

#           hien\_form = False

#           ma\_tim\_duoc = None

#           thong\_bao = ""

#

#           # 📁 Đường dẫn tới file Excel cố định

#           # Sau này phải thay đổi vị trí của ổ đĩa để trả kết quả

#            EXCEL\_PATH = r"C:\Users\Duy To\Inventory\_control\Khuon\_duc.xlsx"

#

#            # Khi có thao tác POST (nhấn nút)

#            if request.method == "POST":

#                action = request.form.get("action")  # Xác định nút bấm là "tim" hay "capnhat"

#                code = request.form.get("Code")  # Lấy mã hàng từ ô nhập

                # Đọc dữ liệu từ Excel

#                df = pd.read\_excel(EXCEL\_PATH, sheet\_name="Sheet1")

                # 🔍 Nếu nhấn "Tìm" mã hàng

#                if action == "tim":

#                   if code in df["Code"].astype(str).values:

#                      hien\_form = True  # Cho hiển thị form nhập ngày nếu tìm thấy

#                     ma\_tim\_duoc = code

#                else:

#                        thong\_bao = f"Không tìm thấy mã hàng: {code}"

#                # 💾 Nếu nhấn "Lưu" để cập nhật ngày

#                elif action == "capnhat":

#                    ngay\_nhap = request.form.get("ngay\_nhap")  # Lấy ngày nhập từ form

#                    # Tìm dòng có mã hàng trùng khớp

#                    mask = df["Code"].astype(str) == str(code)

#                    if mask.any():

#                        # ✅ Ghi đè cột "Ngày nhập kho" tại dòng tương ứng

#                        df.loc[mask, "Ngày nhập kho"] = ngay\_nhap

#                        # 💾 Ghi lại vào file Excel

#                        df.to\_excel(EXCEL\_PATH, sheet\_name="Sheet1", index=False)

#                        thong\_bao = f"✅ Đã cập nhật ngày nhập kho cho Code {code}"

#                    else:

#                        thong\_bao = f"Không tìm thấy mã hàng: {code}"

            # Render HTML, truyền biến điều khiển (form, mã, thông báo)

#            return render\_template("nhap\_ngay.html",

#                                hien\_form=hien\_form,

#                                ma\_tim\_duoc=ma\_tim\_duoc,

#                                thong\_bao=thong\_bao)

            #  THỂ HIỆN ngày nhập kho, lấy hàng, trả hàng theo mã code

@app.route("/update-ngay", methods=["POST"])

def update\_ngay\_bat\_ky():

    code = request.form.get("Code")

    cot = request.form.get("cot")  # Tên cột: "Ngày nhập kho", "Ngày lấy hàng", "Ngày trả hàng"

    ngay = request.form.get("ngay")

    df = pd.read\_excel(EXCEL\_PATH, sheet\_name="Sheet1")

    mask = df["Code"].astype(str) == str(code)

    if mask.any() and cot in df.columns:

        df.loc[mask, cot] = ngay

        df.to\_excel(EXCEL\_PATH, sheet\_name="Sheet1", index=False)

    return redirect("/inventory")  # Quay về trang danh sách hàng tồn kho

# Thể hiện việc thêm sản phẩm trên website

@app.route("/them-san-pham", methods=["POST"])

def them\_san\_pham():

    hinh\_dang = request.form.get("hinh\_dang")

    code = request.form.get("code")

    tuoi\_tho = request.form.get("tuoi\_tho")

    df = pd.read\_excel(EXCEL\_PATH, sheet\_name="Sheet1")

    # ✅ Kiểm tra trùng Code

    if code in df["Code"].astype(str).values:

        print(f"❌ Code {code} đã tồn tại.")

        return redirect("/inventory")  # Có thể xử lý lỗi tốt hơn sau

    # 📦 Tạo dòng mới (các trường còn lại để trống hoặc mặc định)

    new\_row = {

        "STT": len(df) + 1,

        "Hình dạng": hinh\_dang,

        "Code": code,

        "Ngày nhập kho": "",

        "Ngày lấy hàng": "",

        "Ngày trả hàng": "",

        "Số ngày trong kho": "",

        "Tuổi thọ": tuoi\_tho,

        "Số lần đã sử dụng": ""

    }

    df = pd.concat([df, pd.DataFrame([new\_row])], ignore\_index=True)

    df.to\_excel(EXCEL\_PATH, sheet\_name="Sheet1", index=False)

    return redirect("/")

# route đăng xuất

@app.route("/logout")

def logout():

    # 🔄 Xoá toàn bộ session hiện tại (xoá mã và tên nhân viên)

    session.clear()

    # 🔁 Chuyển về trang đăng nhập

    return redirect("/login")

# 🚀 Chạy Flask app

if \_\_name\_\_ == "\_\_main\_\_":

    app.run(debug=True, port=5000)