from flask import Flask, render\_template\_string, request, redirect, session, url\_for import pandas as pd import os from werkzeug.utils import secure\_filename app = Flask(\_\_name\_\_) app.secret\_key = "your\_secret\_key" EXCEL\_FILE = os.path.join(os.path.dirname(\_\_file\_\_), "base\_hor.xlsm") UPLOAD\_FOLDER = os.path.dirname(EXCEL\_FILE) ALLOWED\_EXTENSIONS = {'xls', 'xlsm'} app.config['UPLOAD\_FOLDER'] = UPLOAD\_FOLDER BASE\_HTML = ''' <!doctype html> <html lang="fa"> <head> <meta charset="UTF-8"> <title>صندوق قرض‌الحسنه</title> <link href="https://cdn.jsdelivr.net/gh/rastikerdar/vazir-font@v30.1.0/dist/font-face.css" rel="stylesheet"> <style> body { direction: rtl; font-family: 'Vazir', Tahoma, sans-serif; background-image: url('https://images.unsplash.com/photo-1617957742303-b9e5a986d718?auto=format&fit=crop&w=1400&q=80'); background-size: cover; background-attachment: fixed; color: #fff; padding: 0; margin: 0; } .container { background-color: rgba(0,0,0,0.7); margin: 40px auto; padding: 30px; max-width: 900px; border-radius: 15px; } table { border-collapse: collapse; width: 100%; margin-top: 20px; } th, td { border: 1px solid #ccc; padding: 10px; } th { background-color: #343a40; color: #f8f9fa; } tr:nth-child(even) { background-color: #2e2e2e; } tr:hover { background-color: #495057; } h2, h4 { color: #ffc107; } button, input[type="submit"] { background-color: #17a2b8; border: none; color: white; padding: 10px 18px; font-size: 14px; cursor: pointer; border-radius: 6px; } input[type="text"], input[type="password"], input[type="file"] { padding: 7px; margin: 6px 0; border: none; border-radius: 6px; width: 100%; } label { color: #f8f9fa; font-weight: bold; } a { text-decoration: none; color: #ffc107; } a:hover { color: #fff; } ul { line-height: 1.8; } @media (max-width: 768px) { table, thead, tbody, th, td, tr { display: block; width: 100%; } thead tr { display: none; } tr { margin-bottom: 15px; border-bottom: 2px solid #ccc; padding-bottom: 10px; } td { text-align: right; padding-right: 50%; position: relative; border: none !important; } td::before { content: attr(data-label); position: absolute; right: 10px; top: 8px; font-weight: bold; white-space: nowrap; color: #ffc107; } } </style> </head> <body> <div class="container"> {% if session.get('logged\_in') %} <div style="text-align:left;"><a href="{{ url\_for('logout') }}">🔒 خروج</a></div> {% endif %} {{ content|safe }} </div> </body> </html> ''' # 📌 توابع کمکی def read\_excel\_data(): try: df\_report = pd.read\_excel(EXCEL\_FILE, sheet\_name="report", header=6) df\_invoice = pd.read\_excel(EXCEL\_FILE, sheet\_name="invoice", header=1) return df\_report, df\_invoice except Exception as e: return str(e), None def format\_money(val): try: val = float(val) formatted = f"{val:,.0f} ریال" return to\_persian\_numbers(formatted) except: return "-" def to\_persian\_numbers(text): en\_to\_fa = str.maketrans('0123456789', '۰۱۲۳۴۵۶۷۸۹') return str(text).translate(en\_to\_fa) def allowed\_file(filename): return '.' in filename and filename.rsplit('.', 1)[1].lower() in ALLOWED\_EXTENSIONS LOGIN\_FORM = ''' <h2>ورود به سامانه</h2> <form method=post> <label>نام یا شماره حساب:</label> <input type=text name=username> <label>رمز:</label> <input type=password name=password><br> <input type=submit value="ورود"> </form> <hr> {{ result|safe }} ''' # 📍 روت‌ها @app.route("/", methods=["GET", "POST"]) def index(): if session.get("logged\_in"): if session.get("is\_admin"): return redirect(url\_for("admin\_panel")) else: return redirect(url\_for("user\_panel")) result = "" if request.method == "POST": username = request.form.get("username").strip() password = request.form.get("password").strip() if username == "admin" and password == "admin123": session["logged\_in"] = True session["is\_admin"] = True return redirect(url\_for("admin\_panel")) df\_report, df\_invoice = read\_excel\_data() if isinstance(df\_report, str): result = f"<b>خطا در خواندن فایل:</b> {df\_report}" else: for \_, row in df\_report.iterrows(): account = str(row.get("شماره حساب", "")).strip().replace(".0", "") password\_excel = str(row.get("رمز", "")).strip().replace(".0", "") name = str(row.get("نام و نام خانوادگي", "")).strip() if (username == account or username == name) and password == password\_excel: session["logged\_in"] = True session["is\_admin"] = False session["account"] = account session["name"] = name return redirect(url\_for("user\_panel")) result = "<p style='color:red;'>نام کاربری یا رمز عبور اشتباه است.</p>" return render\_template\_string(BASE\_HTML, content=LOGIN\_FORM, result=result) @app.route("/user") def user\_panel(): if not session.get("logged\_in") or session.get("is\_admin"): return redirect(url\_for("index")) df\_report, \_ = read\_excel\_data() if isinstance(df\_report, str): return render\_template\_string(BASE\_HTML, content=f"<b>خطا:</b> {df\_report}") account = session.get("account") name = session.get("name") row = df\_report[df\_report["شماره حساب"].astype(str).str.replace(".0", "") == account].iloc[0] content = f"<h2>خوش آمدید {name}</h2><p><b>شماره حساب:</b> {account}</p><h4>اطلاعات مالی:</h4><ul>" financial\_data = row.iloc[6:] for col, val in financial\_data.items(): if pd.notna(val) and str(col).strip() != "" and not str(col).startswith("Unnamed") and "nan" not in str(col): content += f"<li>{col}: {format\_money(val)}</li>" content += "</ul><br><a href='/transactions'><button>مشاهده تراکنش‌ها</button></a>" return render\_template\_string(BASE\_HTML, content=content) @app.route("/transactions") def transactions(): if not session.get("logged\_in") or session.get("is\_admin"): return redirect(url\_for("index")) df\_report, df\_invoice = read\_excel\_data() if isinstance(df\_invoice, str): return render\_template\_string(BASE\_HTML, content=f"<b>خطا:</b> {df\_invoice}") account = session.get("account") name = session.get("name") user\_invoice = df\_invoice[df\_invoice["شماره حساب"] == int(account)] search\_query = request.args.get("search", "").strip() if search\_query: user\_invoice = user\_invoice[user\_invoice.astype(str).apply(lambda row: row.str.contains(search\_query, case=False, na=False)).any(axis=1)] important\_cols = ['تاریخ', 'برداشت', 'واریز', 'ماهیانه', 'قسط وام', 'وام', 'وام فوری', 'برگشت وام فوری', 'ش کارت'] content = f''' <h2>تراکنش‌های {name}</h2> <a href="/user">⬅️ بازگشت به داشبورد</a> <form method="get"> <input type="text" name="search" placeholder="جستجو..." value="{search\_query}"> <input type="submit" value="جستجو"> </form> ''' if not user\_invoice.empty: content += "<table><tr>" + "".join(f"<th>{col}</th>" for col in important\_cols) + "</tr>" for \_, row in user\_invoice.iterrows(): content += "<tr>" + "".join( f"<td data-label='{col}'>" + ( format\_money(row.get(col)) if col not in ['توضیحات', 'تاریخ'] and pd.notna(row.get(col)) else ( to\_persian\_numbers(row.get(col)) if pd.notna(row.get(col)) else "-" ) ) + "</td>" for col in important\_cols) + "</tr>" content += "</table>" else: content += "<p>تراکنشی یافت نشد.</p>" return render\_template\_string(BASE\_HTML, content=content) @app.route("/admin") def admin\_panel(): if not session.get("logged\_in") or not session.get("is\_admin"): return redirect(url\_for("index")) df\_report, \_ = read\_excel\_data() if isinstance(df\_report, str): return render\_template\_string(BASE\_HTML, content=f"<b>خطا:</b> {df\_report}") content = ''' <h2>پنل مدیر</h2> <form action="/upload" method="post" enctype="multipart/form-data"> <label>بارگذاری فایل Excel:</label> <input type="file" name="file"> <input type="submit" value="بارگذاری"> </form> <table><tr><th>نام</th><th>شماره حساب</th>''' financial\_columns = df\_report.columns[6:] for col in financial\_columns: if pd.notna(col) and str(col).strip() != "" and not str(col).startswith("Unnamed") and "nan" not in str(col): content += f"<th>{col}</th>" content += "</tr>" for \_, row in df\_report.iterrows(): content += f"<tr><td>{row.get('نام و نام خانوادگي')}</td><td>{row.get('شماره حساب')}</td>" for col in financial\_columns: if pd.notna(col) and str(col).strip() != "" and not str(col).startswith("Unnamed") and "nan" not in str(col): val = row.get(col) content += f"<td>{format\_money(val)}</td>" content += "</tr>" content += "</table>" return render\_template\_string(BASE\_HTML, content=content) @app.route("/upload", methods=["POST"]) def upload\_excel(): if not session.get("logged\_in") or not session.get("is\_admin"): return redirect(url\_for("index")) file = request.files.get("file") if not file or file.filename == '': return redirect(url\_for("admin\_panel")) if allowed\_file(file.filename): filename = "base\_hor.xlsm" filepath = os.path.join(app.config["UPLOAD\_FOLDER"], filename) file.save(filepath) return redirect(url\_for("admin\_panel")) @app.route("/logout") def logout(): session.clear() return redirect(url\_for("index")) if \_\_name\_\_ == "\_\_main\_\_": app.run(debug=True)

کد شما به‌خوبی ساختار یک سامانه‌ی صندوق قرض‌الحسنه با Flask در ادامه، ابتدا تقسیم‌بندی **کلی** ساختار پروژه را توضیح می‌دهم، سپس وارد **جزئیات هر بخش** می‌شوم:

**📦 بخش‌های کلی پروژه:**

1. **پیکربندی‌ها و تنظیمات اولیه**
2. **قالب پایه HTML و استایل**
3. **توابع کمکی (Helper Functions)**
4. **قالب فرم ورود (Login Form Template)**
5. **مسیرها (Routes) برای صفحات مختلف**
   * صفحه ورود
   * پنل کاربر
   * نمایش تراکنش‌ها
   * پنل مدیریت
   * بارگذاری فایل Excel
   * خروج

**🔍 جزئیات هر بخش:**

**1. 🎛 پیکربندی‌ها و تنظیمات اولیه**

python

CopyEdit

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

app.secret\_key = "your\_secret\_key"

EXCEL\_FILE = ...

UPLOAD\_FOLDER = ...

ALLOWED\_EXTENSIONS = ...

* تعریف اپ Flask
* تعیین کلید مخفی برای مدیریت سشن‌ها
* تعیین مسیر فایل اکسل و پوشه آپلود
* تعیین پسوندهای مجاز برای فایل‌ها

**2. 🎨 قالب پایه HTML و CSS (BASE\_HTML)**

python

CopyEdit

BASE\_HTML = '''...'''

* این بخش ساختار پایه صفحات HTML را مشخص می‌کند.
* فونت، رنگ، استایل‌ها و طراحی واکنش‌گرا دارد.
* همه صفحات با استفاده از این قالب ساخته می‌شوند.

**3. 🧰 توابع کمکی**

**📥 خواندن داده‌ها از Excel**

python

CopyEdit

def read\_excel\_data():

* داده‌های دو شیت report و invoice را می‌خواند.

**💵 فرمت کردن پول**

python

CopyEdit

def format\_money(val):

* عدد را با , جدا و به ریال تبدیل می‌کند.

**🔢 تبدیل اعداد انگلیسی به فارسی**

python

CopyEdit

def to\_persian\_numbers(text):

* تمام ارقام انگلیسی را به معادل فارسی تبدیل می‌کند.

**✔ بررسی پسوند فایل مجاز**

python

CopyEdit

def allowed\_file(filename):

* بررسی می‌کند که فایل Excel معتبر باشد.

**4. 🔐 قالب فرم ورود (LOGIN\_FORM)**

python

CopyEdit

LOGIN\_FORM = '''

<h2>ورود به سامانه</h2>

<form method=post>...

'''

* فرم ورود اعضا و مدیران
* شامل فیلد نام کاربری و رمز عبور

**5. 🌐 مسیرها (Routes)**

**🔸 / (صفحه ورود)**

python

CopyEdit

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

* در صورت ورود موفق:
  + اگر مدیر: می‌رود به /admin
  + اگر عضو: می‌رود به /user
* بررسی ورود با اطلاعات فایل Excel یا اکانت مدیر

**🔸 /user (پنل کاربر)**

python

CopyEdit

@app.route("/user")

* نمایش نام کاربر و شماره حساب
* نمایش اطلاعات مالی مربوط به هر ستون معتبر در شیت report

**🔸 /transactions (تراکنش‌ها)**

python

CopyEdit

@app.route("/transactions")

* فقط برای اعضا
* فیلتر کردن تراکنش‌ها بر اساس شماره حساب
* امکان جستجو با عبارت در تراکنش‌ها
* نمایش ستون‌های مهم مثل تاریخ، واریز، برداشت، وام، ماهیانه و غیره

**🔸 /admin (پنل مدیر)**

python

CopyEdit

@app.route("/admin")

* فقط برای مدیران
* نمایش جدول کامل اعضا و اطلاعات مالی آن‌ها
* فرم بارگذاری فایل اکسل جدید

**🔸 /upload (آپلود فایل اکسل)**

python

CopyEdit

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

* فقط برای مدیران
* ذخیره فایل base\_hor.xlsm جدید جایگزین فایل قبلی

**🔸 /logout (خروج)**

python

CopyEdit

@app.route("/logout")

* پاک کردن سشن و بازگشت به صفحه ورود