# app.py – v2.2 (2025-07-14)

# 모바일·데스크톱 레이아웃 + 1~30단원 일괄 실행 + 4단원 자동 포트 구성 지원

from \_\_future\_\_ import annotations

import importlib

import sys

from pathlib import Path

from types import ModuleType

from typing import Any, Dict, Optional

import streamlit as st

from modules.device\_detector import get\_device\_type

# ───────────────────────────────

# 0. 프로젝트 경로 설정

MODULE\_DIR = Path(\_\_file\_\_).parent / "modules"

if str(MODULE\_DIR) not in sys.path:

sys.path.append(str(MODULE\_DIR))

# ───────────────────────────────

# 1. 디바이스 감지

DEVICE = get\_device\_type() # 'mobile' | 'desktop'

# ───────────────────────────────

# 2. 페이지·CSS 기본 설정

st.set\_page\_config(

page\_title="My Stock Analyzer",

layout="wide",

initial\_sidebar\_state="expanded",

)

BASE\_FONT = "14px" if DEVICE == "mobile" else "16px"

st.markdown(

f"""

<style>

html, body, [class\*="css"] {{

font-family: 'Apple SD Gothic Neo', sans-serif;

font-size: {BASE\_FONT};

}}

.stButton button {{

padding: 0.6rem 1.2rem;

font-size: calc({BASE\_FONT} \* 0.95);

}}

.stSidebar .sidebar-content {{

background-color: #F8F9FA;

}}

</style>

""",

unsafe\_allow\_html=True,

)

# ───────────────────────────────

# 3. 세션 전역 초기화

if "user" not in st.session\_state:

st.session\_state.user = "나"

if "meta\_info" not in st.session\_state:

st.session\_state.meta\_info: Optional[Dict[str, Any]] = None

if "fundamental\_df" not in st.session\_state:

st.session\_state.fundamental\_df = None

if "market" not in st.session\_state:

st.session\_state.market = "KR" # 기본 국내

# 사이드바 – 사용자·시장 선택

st.sidebar.title("👤 사용자")

st.session\_state.user = st.sidebar.radio("포트폴리오 주체", ["나", "엄마"], index=0)

st.sidebar.markdown("---")

st.session\_state.market = st.sidebar.radio("시장 선택", ["국내 (KRX)", "해외 (NYSE/NASDAQ)"], index=0,

format\_func=lambda x: x.split()[0])

st.sidebar.markdown("---")

# ───────────────────────────────

# 4. 앱 헤더

device\_emoji = "📱" if DEVICE == "mobile" else "💻"

st.title(f"📊 My Stock Analyzer ({device\_emoji} {DEVICE})")

st.markdown(

"### 모든 단원을 한 번에 실행합니다. "

"모듈별 결과는 각 섹션(expander)에서 확인하세요."

)

st.write("---")

# ───────────────────────────────

# 5. 30개 모듈 이름

MODULE\_NAMES = [f"module\_{i:02d}" for i in range(1, 31)]

# ───────────────────────────────

# 6. 모듈 실행 헬퍼

def \_run\_module(mod: ModuleType, mod\_name: str, \*\*kwargs: Any) -> None:

"""

각 모듈 실행 진입점을 탐색해 호출.

우선순위: run() → main() → 특수 핸들러

"""

if hasattr(mod, "run") and callable(mod.run):

try:

mod.run(\*\*kwargs)

return

except TypeError:

mod.run()

return

if hasattr(mod, "main") and callable(mod.main):

mod.main(\*\*kwargs)

return

if mod\_name == "module\_01" and hasattr(mod, "analyze\_stock\_basic"):

\_module01\_ui(mod)

return

if mod\_name == "module\_03" and hasattr(mod, "StrategyAnalyzer"):

\_module03\_ui(mod)

return

if mod\_name == "module\_04" and hasattr(mod, "BacktestSimulator"):

\_module04\_ui(mod)

return

st.info("실행 가능한 진입점이 없습니다(run/main).")

# ───────────────────────────────

# 7-A. module\_01 전용 UI

def \_module01\_ui(mod: ModuleType) -> None:

user\_query = st.text\_input("종목명 또는 티커 입력", "삼성전자")

if st.button("🔍 기본 정보 분석", key="module01\_btn"):

with st.spinner("분석 중…"):

try:

fv = mod.analyze\_stock\_basic(user\_query)

st.session\_state.meta\_info = fv.as\_dict()

except Exception as e:

st.error(f"❌ 분석 실패: {e}")

return

st.success(f"✅ 완료 – {fv.as\_dict()['name']} ({fv.ticker})")

st.markdown("#### 🧠 초보자용 설명"); st.write(fv.as\_dict()["beginner\_summary"])

st.markdown("#### 📈 전문가용 요약")

st.code(fv.as\_dict()["expert\_summary"], language="markdown")

with st.expander("🔎 세부 데이터(JSON)", expanded=False):

st.json(fv.as\_dict(), expanded=False)

# ───────────────────────────────

# 7-B. module\_03 전용 UI

def \_module03\_ui(mod: ModuleType) -> None:

from modules.module\_01 import analyze\_stock\_basic

from modules.module\_02 import analyze\_financials

from modules.module\_03 import StrategyAnalyzer

st.markdown("### 📘 3단원: 통합 전략 판단 시스템")

query = st.text\_input("종목명 또는 티커", "삼성전자", key="mod03\_query")

adv\_flag = st.toggle("고급 분석 (매크로·시나리오·리스크)", value=True)

if st.button("🔍 전략 분석 실행", key="mod03\_btn"):

with st.spinner("데이터 수집 및 분석 중..."):

fv = analyze\_stock\_basic(query)

fundamental\_df, \_ = analyze\_financials(fv.ticker)

analyzer = StrategyAnalyzer(

ticker=fv.ticker,

fundamental\_df=fundamental\_df,

meta\_info=fv.as\_dict(),

period="1y",

advanced=adv\_flag,

fred\_api\_key=st.secrets.get("FRED\_KEY"),

)

result = analyzer.run()

st.metric("📊 종합 전략 점수", f"{result['composite\_score']:.1f}")

st.markdown(f"\*\*추천 판단\*\*: {result['recommendation']}")

st.markdown(f"\*\*설명\*\*: {result['explanation']}")

with st.expander("📐 세부 점수", expanded=False):

st.json(result["scores"])

with st.expander("🧪 리스크 지표", expanded=False):

st.json(result["risk\_metrics"])

with st.expander("📈 시나리오 판단", expanded=False):

st.json(result["scenario"])

# ───────────────────────────────

# 7-C. module\_04 전용 UI (시장 선택 → 자동 포트 구성)

def \_module04\_ui(mod: ModuleType) -> None:

from modules.module\_04 import BacktestSimulator

market\_code = "KR" if st.session\_state.market == "국내 (KRX)" else "US"

st.markdown("### 📘 4단원: 포트 자동 구성 & 백테스트")

col1, col2 = st.columns(2)

with col1:

start = st.date\_input("시작 날짜", value=dt.date(2020, 1, 1), key="bt\_start")

with col2:

end = st.date\_input("종료 날짜", value=dt.date(2024, 6, 30), key="bt\_end")

top\_n = st.slider("종목 개수", 3, 15, 5, 1)

if st.button("🚀 백테스트 실행", key="bt\_run"):

with st.spinner("포트 구성 및 시뮬레이션 중…"):

sim = BacktestSimulator(

market=market\_code,

start=str(start),

end=str(end),

top\_n=top\_n,

rebal\_freq="M",

)

res = sim.run()

st.success("✅ 백테스트 완료")

st.subheader("선택 종목 & 비중")

st.write(res["weights"])

st.metric("CAGR", f"{res['metrics']['CAGR']\*100:.2f}%")

st.metric("Sharpe", f"{res['metrics']['Sharpe']:.2f}")

st.line\_chart(res["curve"])

# ───────────────────────────────

# 8. ‘전체 실행’ 버튼

if st.button("🚀 전체 단원 실행", key="run\_all"):

for idx, mod\_name in enumerate(MODULE\_NAMES, start=1):

with st.expander(f"📘 {idx:02d}단원 — {mod\_name}.py", expanded=False):

try:

mod = importlib.import\_module(f"modules.{mod\_name}")

except ModuleNotFoundError:

st.warning(f"`{mod\_name}.py` 파일이 없습니다. (건너뜀)")

continue

# ► module\_01

if mod\_name == "module\_01" and hasattr(mod, "analyze\_stock\_basic"):

\_module01\_ui(mod); continue

# ► module\_02

if mod\_name == "module\_02" and hasattr(mod, "analyze\_financials"):

if st.session\_state.meta\_info is None:

st.info("먼저 1단원에서 종목을 선택하세요."); continue

ticker = st.session\_state.meta\_info["ticker"]

with st.spinner("재무 데이터 분석 중…"):

try:

fundamental\_df, fin\_summary = mod.analyze\_financials(ticker)

st.session\_state.fundamental\_df = fundamental\_df

except Exception as e:

st.error(f"❌ 재무 분석 실패: {e}"); continue

st.success("✅ 2단원 완료"); st.json(fin\_summary, expanded=False); continue

# ► module\_03

if mod\_name == "module\_03" and hasattr(mod, "StrategyAnalyzer"):

if st.session\_state.meta\_info is None or st.session\_state.fundamental\_df is None:

st.info("1·2단원 결과가 필요합니다."); continue

from modules.module\_03 import StrategyAnalyzer

analyzer = StrategyAnalyzer(

ticker=st.session\_state.meta\_info["ticker"],

fundamental\_df=st.session\_state.fundamental\_df,

meta\_info=st.session\_state.meta\_info,

period="1y",

advanced=True,

fred\_api\_key=st.secrets.get("FRED\_KEY"),

)

with st.spinner("전략 판단 중…"):

try: res = analyzer.run()

except Exception as e:

st.error(f"❌ 3단원 실행 오류: {e}"); continue

st.metric("Composite Score", f"{res['composite\_score']:.1f}")

st.markdown(f"\*\*추천\*\*: {res['recommendation']}")

st.write(res["explanation"]); st.json(res["risk\_metrics"], expanded=False)

continue

# ► module\_04 (자동 포트 구성 & 백테스트)

if mod\_name == "module\_04" and hasattr(mod, "BacktestSimulator"):

\_module04\_ui(mod); continue

# ► 기타 모듈

try:

\_run\_module(

mod, mod\_name,

user=st.session\_state.user,

device=DEVICE,

market=st.session\_state.market,

meta\_info=st.session\_state.meta\_info,

fundamental\_df=st.session\_state.fundamental\_df,

)

except Exception as e:

st.error(f"❌ 실행 오류: {e}")

else:

st.info("왼쪽 위의 \*\*🚀 전체 단원 실행\*\* 버튼을 눌러 분석을 시작하세요.")