**סיכום העבודה עד כה:**

התחלנו עם הרעיון לבנות אפליקציית רשת לניתוח נתונים פיננסיים. עד כה, התמקדנו בבניית השלד של אפליקציית Flask, ובפיתוח הפונקציונליות המרכזית של דף הבית:

1. **הקמת אפליקציית Flask בסיסית:**
   * יצרנו את app.py עם הגדרות ראשוניות, כולל טעינת FLASK\_SECRET\_KEY מקובץ secret.py.
   * הגדרנו Blueprints לניתוב חלקים שונים של האפליקציה (home\_bp, graphs\_bp, valuations\_bp).
   * יצרנו תבנית HTML ראשית (base\_layout.html) הכוללת תפריט צד ובר עליון עם טופס להזנת טיקר מניה. הטופס ממוקם כעת בפינה השמאלית העליונה.
   * יצרנו תבנית תוכן לדף הבית (content\_home.html) ותבניות placeholder לדפים אחרים.
2. **פונקציונליות דף הבית (הזנת טיקר והצגת גרפים):**
   * המשתמש יכול להזין טיקר בטופס.
   * הנתיב הראשי (/) ב-modules/routes/home.py מטפל בבקשות POST מהטופס ובבקשות GET להצגת הדף.
   * **אחזור נתונים (modules/price\_history.py):**
     + פונקציה get\_price\_history מורידה נתוני מחירים היסטוריים (OHLCV) באמצעות yfinance לפי תקופה ואינטרוול מבוקשים.
     + פונקציה get\_company\_name מאחזרת את שם החברה המלא באמצעות yfinance.
   * **יצירת גרפים (modules/chart\_creator.py):**
     + פונקציה create\_candlestick\_chart יוצרת גרפי נרות אינטראקטיביים באמצעות Plotly מהנתונים שהורדו.
     + הוספנו לגרפים קווי ממוצעים נעים ל-20, 50, 100, 150, ו-200 יחידות זמן (למשל, ימים).
   * **לוגיקה ב-Route של דף הבית:**
     + בעת שליחת טיקר, המערכת מורידה נתונים ומכינה JSON עבור שלושה גרפים ספציפיים:
       1. נתונים יומיים (3 שנים אחורה), הצגת גרף לשנתיים האחרונות.
       2. נתונים שבועיים (5 שנים אחורה).
       3. נתונים חודשיים (10 שנים אחורה).
     + ה-JSON של הגרפים מועבר ישירות לתבנית content\_home.html בעת רינדור הדף (לאחר POST), כדי להימנע מבעיית "session cookie too large".
     + נעשה שימוש במערכת flash של Flask להצגת הודעות שגיאה והתראות למשתמש.
   * **הצגה בתבנית (content\_home.html):**
     + התבנית מציגה את שם החברה, הטיקר, הודעות flash, ואת שלושת הגרפים באמצעות Plotly.js.
3. **בדיקות (pytest):**
   * יצרנו קובץ בדיקות בסיסי (tests/test\_routes.py) עם בדיקות לטעינת דפים ותקינות תוכן ראשוני.
   * פתרנו בעיות הקשורות להרצת pytest (שגיאות SyntaxError עם תווים שאינם ASCII, ובעיות איטיות בשלב איסוף הבדיקות, שזוהו כקשורות בעיקר לתוספים pytest-cov ו-pytest-typeguard).
   * תיקנו AssertionError בבדיקות על ידי הפיכת הבדיקה לגמישה יותר לשינויים קלים ב-HTML.
4. **פתרון בעיות:**
   * נפתרה שגיאת ModuleNotFoundError על ידי הבהרת הצורך להריץ את האפליקציה דרך app.py.
   * נפתרה שגיאת 405 Method Not Allowed על ידי וידוא שהנתיב הראשי מוגדר לקבל בקשות POST והטופס שולח אותן כראוי.
   * טופלה אזהרת "The 'session' cookie is too large" על ידי שינוי האסטרטגיה מהעברת JSON של גרפים דרך ה-session להעברה ישירה לתבנית.

השלב הבא שדיברנו עליו הוא הוספת אפשרות להסתיר/להציג את הממוצעים הנעים בגרפים.