

הנדסת מערכות חלונות – פרוייקט סיום  
סמסטר א' תשפ"ה

כללי:

מערכות תוכנה מודרניות מעוצבות כמערכת מבוזזות, הכוללות מספר Tiers, מתממשקות לשירותי ענן וכן למודלים מסוג Generative AI המסוגלות לסייע במשימות כגון קשרי לקוחות, ניתוח נתונים או התייעצות לגבי מאפייני דומיין ספציפי וקבלת המלצות לדרכי פעולה אפשריות.

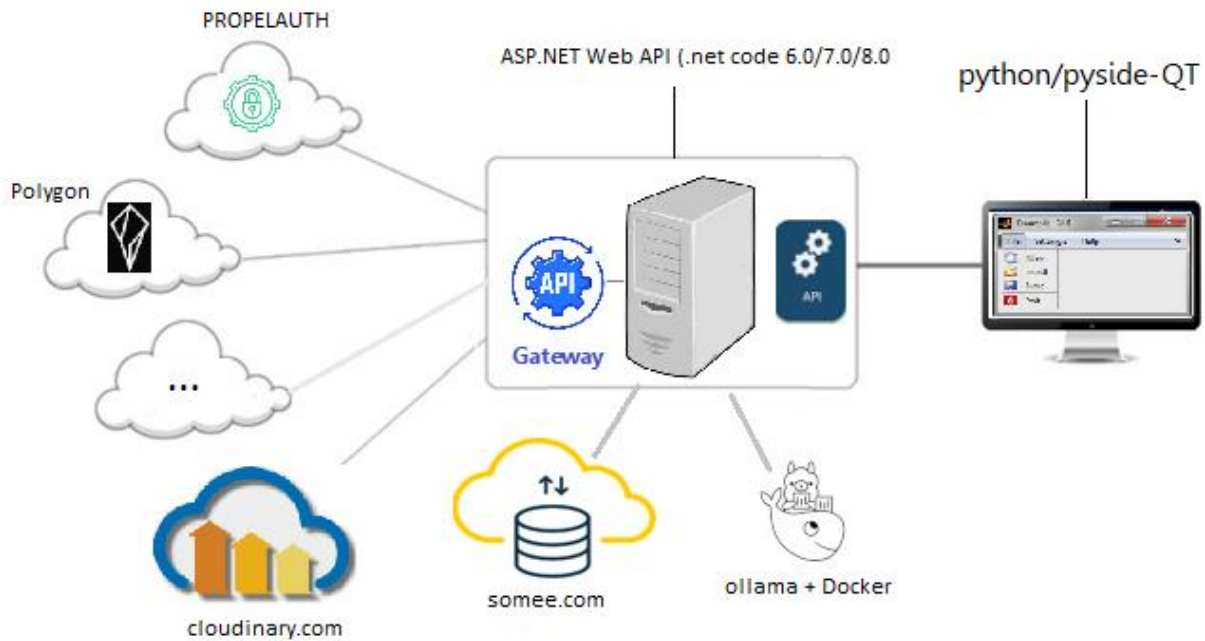
בפרויקט זה נתכנן ונפתח מערכת המאפשרת ניהול תיק מניות וכוללת תת מערכת תצוגה, שרת אפליקציה המשולב עם שרת Gateway המנהל תקשורת עם שירותי ענן וכן התממשקות למודל LLM המממש RAG (Retrieval Augmented Generation) ומשמש "כיועץ" אודות השקעות בבורסה.

דרישות פונקציונליות:

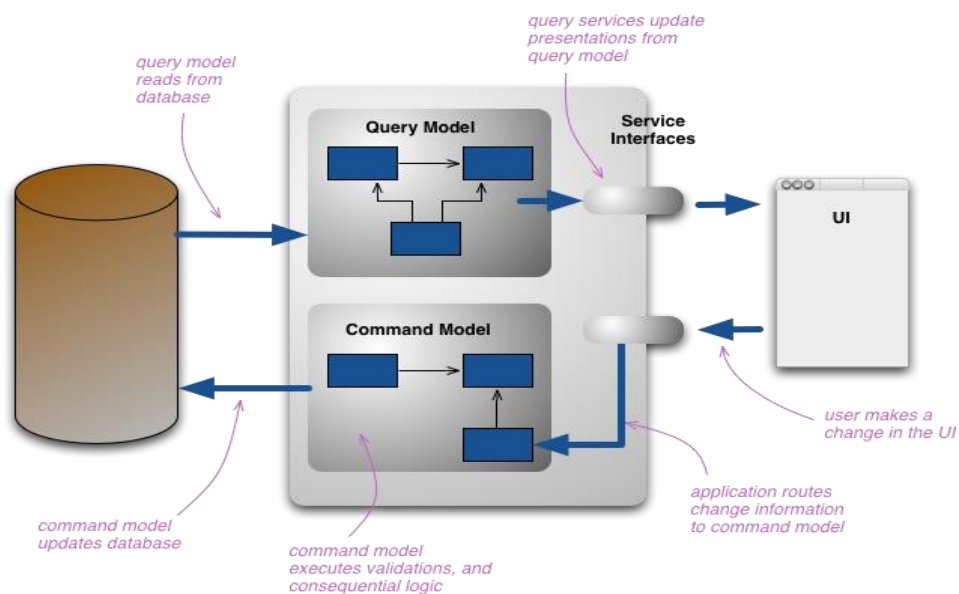
1. על המערכת לנהל משתמשים ולתמוך בביצוע אוטנטיקציה עם הכניסה אליה.
2. המערכת תתמוך בתהליכי משתמש הבאים:
  - 2.1. ביצוע הוראת קניה
  - 2.2. ביצוע הוראת מכירה
  - 2.3. הצגת היסטורית הוראות בטבלה או בגרף מתאריך לתאריך לגבי מניה אחת או יותר.
  - 2.4. הצגת סטטוס מניות כללי בתיק לרבות שער מניות נוכחי ושווי התיק.
  - 2.5. התייעצות עם סוכן AI אודות מושגים והמלצות בתחום הפיננסי.

דרישות לא פונקציונליות:

1. תת מערכת תצוגה היא מסוג Desktop Application.
2. תת מערכת תצוגה תפותח באמצעות PySide 6/6.5 ותממש תבניות MVP ו-Microfrontends.
3. מומלץ שימוש בספריית [QTCharts](https://pypi.org/project/QTCharts/) ליצירת גרפים.
4. שירותי שרת האפליקציה יאורגנו באמצעות תבניות MVC/CQRS ותפותח באמצעות ASP.NET core Web API 6.0/8.0.
5. שמירת הנתונים תעשה בשירות ענן (מומלץ somee.com) ותממש גישת Event Sourcing.
6. הגישה לשירותים החיצוניים תאורגן באמצעות תבנית Gateway.
7. המערכת תקרא לשירות כדוגמת <https://polygon.io/> לצורך קבלת נתוני מניות.
8. המערכת תקרא לשירות <https://cloudinary.com/> בכדי לאחסן תמונות.
9. המערכת תשתמש בשרת Ollama הרץ במיכל Docker כדי לממש RAG ולהתייעץ אודות השקעות.
10. את הקוד יש לנהל במסגרת GitHub Repo (או תשתית דומה).
11. **אופציה:** השתמשו בשרות cloudinary לאחסון תמונות (ניתן גם לבצע טרנספורמציות מבוססות AI).
12. **אופציה:** ביצוע Deploy לשרת האפליקציה לספק שירותי ענן (כדוגמת somee.com).



תרשים 1: מרכיבי מערכת ניהול תיק מניות



תרשים 2: שימוש בתבנית CQRS לעיצוב שירותי שרת האפליקציה

#### משאבים:

- (1) אודות פיתוח GUI עם PySide: [PySide6 Tutorial: Building GUI Applications with Python | DataCamp](#)
- (2) תבניות ארכיטקטוניות מרכזיות למחשוב ענן: <https://code-b.dev/blog/best-cloud-native-architecture-patterns>
- (3) שימוש בספריית QtCharts ליצירת גרפים: [Python QT Charts | Creating And Customizing Charts | Graphs | Pyside | PyQt | Infrface Design #1 - YouTube](#)

#### הערות:

1. את הפרויקט יש להגיש בזוגות – הגשה בשלוש אפשרית אך מחייבת מימוש סעיפי אופציה.
2. נא להרשם להגשה כאן:

[https://docs.google.com/spreadsheets/d/1uypahC31RZlvGox\\_Svhn-OoIPKZv09vOtlMzuUtNMu4/edit?usp=sharing](https://docs.google.com/spreadsheets/d/1uypahC31RZlvGox_Svhn-OoIPKZv09vOtlMzuUtNMu4/edit?usp=sharing)

3. במסגרת ההגשה יישאלו שאלות הנוגעות לחומר שנלמד בקורס ודנו בו בהרצאות.
4. על התצוגה להיות מקצועית ואסתטית ככל האפשר, נא לחפש בגוגל רעיונות לעיצובים או להשתמש בסוכן AI לצורך כך. לתצוגה המחוזה *מקצועיות* יינתן משקל בהערכת הפרויקט הכללית.
5. אתם רשאים להשתמש בפרויקטים דומים ואף ב- ChatGPT, אך רק במסגרת פרויקט חדש שלכם אליו העברתם חלקים בצורה מיוחדת, על בסיס הארכיטקטורה והמודולים שאתם הגדרתם, ובכל מקרה עליכם להכיר באופן מלא תפקידו של כל מודול וחלק בקוד, בדגש על התכן \ עיצובו הלוגי.

בהצלחה. 