**EASS שיעור 10- docker compose, Streamlit**

חלק שלישי של המטלה-

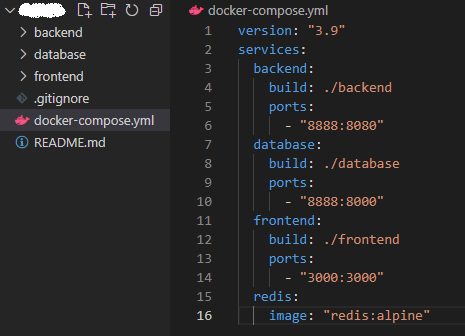
* Docker-compose
* סרטון קצר שמציג כיצד להריץ את האתר (docker-compose up וכו')
* לפחות 3 מיקרוסרוויסים (backend, frontend הם שני מיקרוסרוויסים שחייב לעשות)

Docker-compose:

מאפשר חיבור של כמה microservices באמצעות קובץ אחד מסוג .yml

נבצע את הבנייה של הimage באמצעות פקודה docker-compose build  
(במקום docker build שעשינו עד כה)

נבצע את יצירת הקונטיינר באמצעות פקודה docker-compose up  
(במקום docker run שעשינו עד כה)

עבור הפרוייקט שלנו, עלינו לבנות קובץ docker-compose.yml בצורה הבאה:  


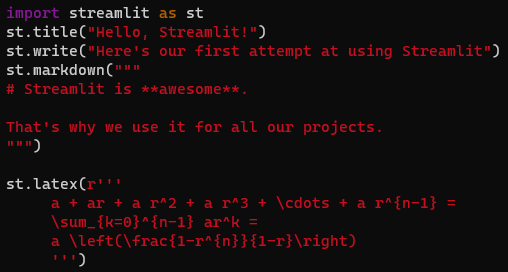
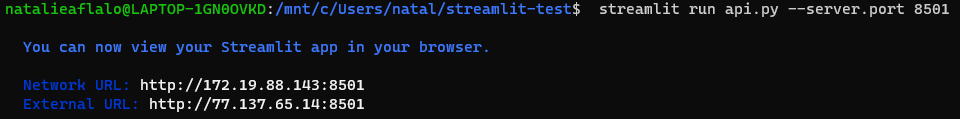
נשים לב למילים השמורות:

* Version- הגרסה של docker-composee
* service- הצגת המיקרוסרוויסים שנרצה לחבר באמצעות docker-compose
* build- מיקום התיקייה של המיקרוסרוויס
* ports- הפורטים שדרכם המיקרוסרוויס יעבוד. הפורט השמאלי הוא הפורט החיצוני (זה שנרשום בדפדפן לצד localhose:) והפורט הימני הוא הפורט הפנימי של הקונטיינר.
* Image- קריאה לimage שקיים בdocker hub ונרצה להשתמש בו (במקום ליצור image עם תיקייה נוספת שיש בה Dockerfile ושאר הרכיבים שבונים אותו). נרשום את שם ה-image והגרסה שלו כמו שקיים ב-docker hub. במידה ונשתמש בזה ניישם את השימוש בסרוויס זה בתוך אחד מקבצי .py בעזרת ספריות client של פייתון.
* Volumes- העתקת קובצים לוקליים לקונטיינר (מתאים לשימוש בסרוויס database). משמאל לנקודותיים הנתיב הלוקלי, מימין לנקודותיים הנתיב בקונטיינר. 

Streamlit:

ספרייה בפייתון שיוצרת UI (user interface).

שימוש: תחילה יש להריץ pip install streamlit.   
הערה: אם נבחר להשתמש ב-streamlit ב-frontend שלנו, נשים לב שכאשר ניצור את ה-dockerfile של ה-frontend נכתוב בקובץ requirements.txt את streamlit וכך פעולת ההתקנה תתבצע גם בקונטיינר.

ניצור תיקייה וקובץ  
  
דוגמה לתוכן של קובץ (מהמצגת שיעור)  
  
ונריץ את הפקודה  
  
כעת כאשר ניגש לקישור שיופיע על המסך נקבל את הדף הבא-  
