Dockerized Flask-Mongo App Project

פרטים לינקים והנחיות

מטרות התרגיל:

Web api

Docker

Docker compose

Mongodb

Flask

עבודה עם גיט

אבטחה (עבודה עם מפתח API)

<u>תאור הפרויקט:</u>

בפרויקט זה עליכם לפתח שירות המאפשר גישה תכנותית דרך האינטרנט, למאגר מידע של תמונות (פוסטרים של סרטים).

המאגר מבוסס mongoDB, בתוכו ישמרו קבצי הסרטים ומידע על הקבצים (שם סרט, קוד IMDB וכו')

התוכנה מבוססת flask.

על התוכנה ובסיס הנתונים להיות "עטופים בקונטיינרים"

על התוכנה לממש את האפשרויות הבאות כ :web API

1. חיפוש פוסטר במאגר (יחזיר קובץ תמונה, במידה ואין במאגר יועבר החיפוש ל (TMDB
2. מחיקת פוסטר מהמאגר.
3. עדכון פרטים על הפוסטר.

בנוסף התוכנה תאפשר את החיפוש גם בעזרת טופס HTML שיהיה נגיש בעזרת דפדפן.

1. תכנון מבנה הפרויקט
   a. קבצים, תיקיות וכו'
   b. שימו לב לmodules and packages
   c. שימו לב לעבודה על פרויקט בסביבה ווירטואלית ((venv
   d. פרמטרים (איזה פורטים, שמות של הוסטים, מפתחות וכו')

2. יצירת תשתית בגיט וגיט האב לשיתוף פעולה של הצוות.
   a. קראו על **git team workflow** תכננו את אופי העבודה בגיט כצוות
   https://hackandslash.blog/git-workflow-for-small-teams-and-single-repository/
   https://rogerdudler.github.io/git-guide/
   b. דאגו לסנכרן את פאיצ'ארם עם חשבון הגיטהאב שרלוונטי לפרויקט.

3. בניית אפליקציה פשוטה המשתמשת ב **API** של **TMDB** להוריד פוסטרים של תמונות לאחר חיפוש ושמירה בתיקיית קבצים.
   https://bin.re/blog/tutorial-download-posters-with-the-movie-database-api-in-python/
   עליכם לבנות את האפליקציה בחלוקה לקבצים ובאופן מודולרי.בהמשך הממשק הולך להשתנות (ממשק המשתמש והשכבה שבה שומרים את הנתונים).
   a. שימוש ב **web api** של **TMDB** (שימו לב להתייחסות למפתח והיכן שומרים אותו)
   https://sd18spring.github.io/notes/storing-api-keys
   https://www.freecodecamp.org/news/how-to-securely-store-api-keys-4ff3ea19ebda/
   b. מציאת קוד **IMDB** לסרט (מקל על החיפוש ב **TMDB**)
   https://www.geeksforgeeks.org/python-imdbpy-searching-a-movie/
4. תכנון ובניית מנגנון שמירה לקבצי סרטים במונגו.
   במודול נפרד (קובץ) יש לממש **CRUD** מלא

   a. קריאה של קובץ מה **DB**
   b. כתיבה של קובץ ל **DB** (רשות - לא בפונקציונאליות הסופית)
   c. עדכון ערך ברשומה
   d. מחיקה של רשומה

   (יש אתגר של שמירת קבצים בינאריים גדולים ב **DB**)
   https://pymongo.readthedocs.io/en/stable/examples/gridfs.html

   https://dev.to/thenishant/store-images-in-mongodb-via-python-2g73

   קריאת קבצים ממונגו

   https://psabhay.com/posts/mongodb/mongodb-gridfs-using-python/

5. שילוב של שני המודולים יחד - יצירת מודול המייצג את השילוב של שניהם - ממשק פשוט כשלב ביניים לפני יצירת ממשק ווב

6. שימוש בפלאסק של פייתון (פריימוורק ווב שנותן גם פרונט וגם בק) ליצירת ממשק וAPI עבור חיפוש פוסטר
https://ishmeet1995.medium.com/how-to-create-restful-crud-api-with-python-flask-mongodb-and-docker-8f6ccb73c5bc
יש ליצור בעזרת web api פעולות CRUD עבור מונגו

a. קריאה של קובץ מה DB
b. כתיבה של קובץ ל DB (רשות - לא בפונקציונאליות הסופית)
c. עדכון ערך ברשומה
d. מחיקה של רשומה

יש ליצור ממשק HTML פשוט לקלט חיפוש סרט (template render using flask)

7. לעטוף את מונגו בדוקר (שימוש באימג' מוכן)
(נמצא בלינק בסעיף 6)
8. לעטוף את האפליקציה בדוקר
https://zetcode.com/python/docker/
(מידע נוסף נמצא בלינק בסעיף 6)
9. לחבר בין כולם ולתת להם לנגן מומלץ לקרוא על docker compose לפני
(נמצא בלינק בסעיף 6)

10. יצירת קובץ user data שיתקין דוקר, ייקח את הגרסה האחרונה מהגיט, ייצור את האימג'ים וירים את הקונטניינרים. (שימו לב ל**מפתח**)
11. להרים EC2 ב sandbox עם ה user data ולראות שאכן האפליקציה עולה ועובדת.

עוד לינקים על דוקר :

https://realpython.com/dockerizing-flask-with-compose-and-machine-from-localhost-to-the-cloud/

https://realpython.com/twitter-sentiment-python-docker-elasticsearch-kibana/

https://realpython.com/python-versions-docker/

למתקדמים:

- לכתוב סקריפט המתחבר ל AWS בעזרת ה CLI מגדיר תשתית ומרים את ה EC2 עם user data באופן אוטומטי.
- https://www.learnaws.org/2020/12/16/aws-ec2-boto3-ultimate-guide/
- ליצור AMI מהEC2 עם הדוקרים ולנסות להרים את מעבדת אידון העומסים עם ה AMI שבנינו