**מטלת מבחן ליום אחד - OpenShift SQL Docker**

**הנחיות:**

**אופן עבודה:**

* **יש לעבוד ללא עזרי AI לסוגיהם, תלמיד שייתפס משתמש בכלי AI - מבחנו ייפסל ללא ערעור.**
* **ניתן ואף רצוי להיעזר בחומר שניתן בכיתה, ועבודות קודמות שלכם בקורס הנמצאות ב-GitHub.**
* **אין להעתיק קוד מהאינטרנט או מחברים.**
* **מותר להשתמש בחיפוש בגוגל.**

**שימוש ב - AI = גניבת דעת**

### **הגשה:**

* יצירת repository ציבורי מסודר ב-GitHub
* שיתוף הקוד ע"י הוספת קישור ל-repository בטופס הבא:[הגשת פרויקט](https://docs.google.com/spreadsheets/d/1zsRkPAFeabEfgRbwJkoEMTplcYnv3V8xJ51pXx2jqUY/edit?usp=sharing)

**זמן לביצוע:**

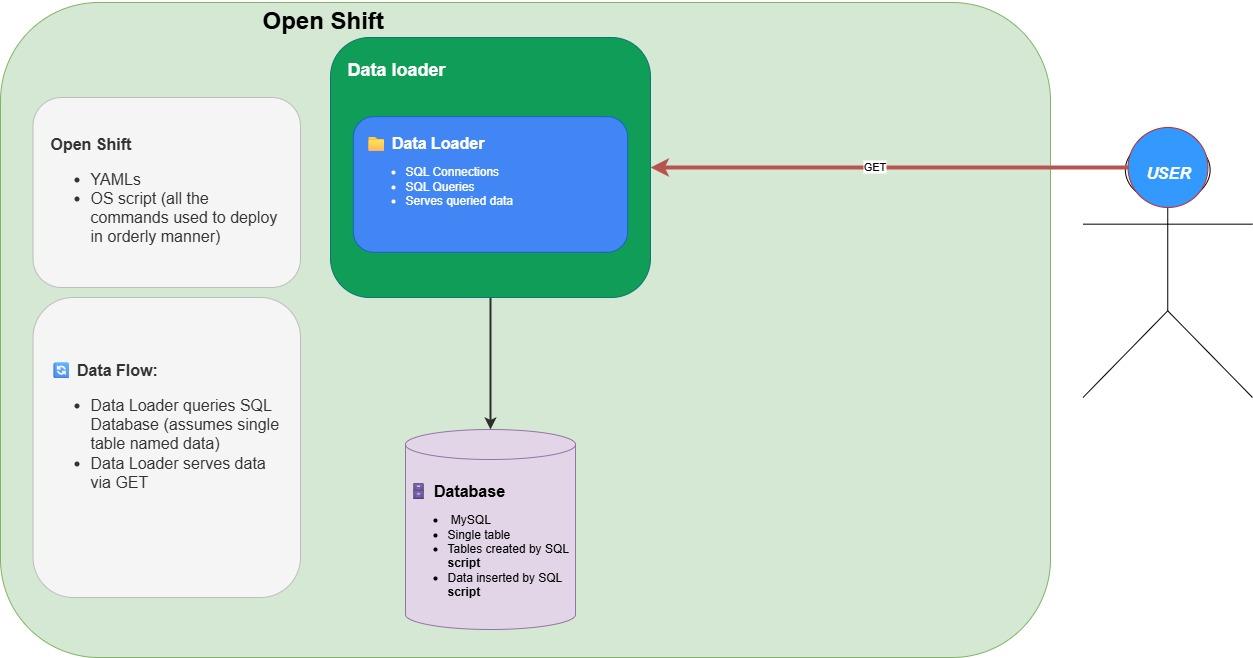
* משעה 10:30 ועד 16:40
* חובה עליכם להשתתף בשיעור תורה בשעה 13:00, ניתן לצאת להפסקות על פי לוז שיוגדר בכיתה, אולם מומלץ להתמקד במבחן.

שימו לב - מבחינת בדיקת המיני פרויקט - COMMITS לאחר השעה 16:40 לא יחושבו בבדיקה, יש לעמוד במסגרת הזמנים לציון מלא.

יש כמובן להתייחס לתכנון נכון, קריאות, עקרונות סוליד, תיעוד, מבנה קבצים, פונקציונליות ושימוש בגיט.

**המטרה:**

בניית סרוויס אשר מחזיר לבקשת GET את כל המידע שקיים בטבלה בודדת (הנקראת data).

  
בפרויקט קיימים הדברים הבאים:

* מסד נתונים **MySQL** רץ באמצעות OpenShift ( יש להשתמש ב image שנוריד מDockerHub )
* מימוש שכבת גישה לנתונים (Class DataLoader)
* הקמת Server ב **FastAPI -** שניגש ל MySQL ומחזיר את נתוני הטבלה בפניה ל-Endpoint ייעודי.
* הAPI נחשף החוצה בעזרת **Route** כך שאפשר לפתוח דפדפן ולראות את הנתונים שמגיעים מה- DB. (או באמצעות דרך בדיקה אחרת - סקריפט פייתון, CURL, postman וכו)

**דרישות מקדימות**

* חשבון Docker Hub בכדי למשוך ולהעלות image (צריך להתחבר לDockerHub דרך CLI).

**משימות:**

1. **יצירת פרויקט חדש ב - .OpenShift ניקוד - 10%**

* יש למחוק פרויקט קיים במידה ויש.
* יש ליצור פרויקט חדש ולעבור לעבוד בתוכו.
* יש לאגד את כל הפקודות שבונות את הפרויקט, מריצות את הפריסה, יוצרות אחסון וכו' , וכל פעילות אחרת, **בקובץ טקסט פשוט שייקרא commands.bat** (יש לשים בתיקיית scripts )
* יש ליצור קבצי YAML עבור כל צורך רלוונטי, אין להשתמש בממשק הגרפי.

1. **פריסת MySQL בפרויקט. ניקוד - 20%**

* יש להעביר ססמאות בעזרת משתני סביבה כפי שנלמד בתרגילי openshift.   
  בונוס למתקדמים - ניתן ליצור Secret עם סיסמאות הגישה ל - MySQL במקום משתני סביבה. (לא ניתן לקבל יותר מ 100)
* **יש ליצור קבצי YAML רלוונטיים**
* יש ליצור PVC לאחסון הנתונים של MySQL
* יש ליצור Deployment ו - Service עבור MySQL

1. **הקמת פרויקט FastAPI. ניקוד 20%**

* יש ליצור DAL, ולשלב אותו באופן נכון בתוך fastAPI
* EndPoind פשוט שמחזיר את המידע שקיים בטבלה data.

1. **הכנת Dockerfile ניקוד 5%**

* **ליצור Dockerfile** שאורז את האפליקציה לתמונה קלה לפריסה.

1. **Build & Push של הפרויקט (יש להשתמש בנקודה חמה hot spot ) ניקוד 5%**

* לבנות את התמונה המקומית.
* לדחוף את התמונה לDocker Hub

1. **פריסת ה-Backend ב -OpenShift וחיבור ל - MySQL ניקוד 20%**

* יש ליצור Deployment עבור אפליקציית FastAPI.
* יש להגדיר משתני סביבה לחיבור למסד הנתונים.
* יש ליצור Service פנימי לחשיפת האפליקציה בתוך הקלאסטר.
* יש לאגד את כל הפקודות שבונות את הפרויקט, מריצות את הפריסה, יוצרות אחסון וכו' , וכל פעילות אחרת, **בקובץ טקסט פשוט שייקרא commands.bat** (יש לשים בתיקיית scripts )
* יש ליצור קבצי YAML עבור כל צורך רלוונטי, אין להשתמש בממשק הגרפי.

1. **אתחול נתונים ב - MySQL ניקוד 10%**

* ליצור קובץ בשם **create\_data.sql** היוצר טבלה בשם data עם העמודות הבאות
  + ID שדה מספרי
  + first\_name
  + last\_name
* ליצור קובץ **insert\_data.sql** המכניס כחמש שורות נתונים מתאימים לבחירתכם.
* להעתיק את הקבצים לפוד
* להתחבר ל-MySQL מתוך הפוד, ולהריץ את הסקריפטים כך שהטבלה תיבנה והנתונים יוכנסו אליה. רמז: <https://dev.mysql.com/doc/refman/8.4/en/mysql.html>
* יש למקם את הקבצים בתיקיית scripts

1. **חשיפת השירות החוצה עם - Route ניקוד 10%**
   * ליצור Route ציבורי לשירות הFastAPI
   * לוודא שניתן לגשת אליו מהדפדפן ולקבל את הנתונים מהDB (או באמצעות דרך בדיקה אחרת)

מבנה תיקיות נדרש:.

data-loader/

├── services/

│ └── data\_loader/ # Data loading service

├── scripts/ # SQL scripts, OS scripts

├── infrastructure/

│ └── k8s/ # Kubernetes manifests (YAMLs)

├── Dockerfile

├── requirements.txt

└── README.md

This structure shows:

services/ - Contains the main data loading service

scripts/ - For SQL and OS scripts

infrastructure/ - Kubernetes deployment configurations

Standard project files at the root level