

מטרה: ליצור אפליקציה FULLSTACK שמנהלת תחקור של מאגר מידע של קורסים.

חלוקת האפליקציה:

קלינט: טרמינל, מציג תפריט של פעולות. כל פעולה שנבחרת מפעילה פונקציה משתמשת בחיבור DB.  
DB: חיבור לDB של MySQL, מרים בעזרת XAMPP, מתחברים בעזרת CONNECTOR.

התפריט צריך להיות בדילן כז:

Load CSV into DB .1

Search records by institution name .2

Search records by course name .3

Find most/least common course .4

Show course count per district .5

Free SQL query .6

Exit .7

הפרויקט משתמש ב-:

(PyMySQL או) mysql-connector-python •

• טבלה אחת בלבד: courses

• רק פקודות SQL גולמיות (raw SQL)

• **לא** ORM ולא מחלקות מודלים

• ארכיטקטורה נקייה (קבצים נפרדים לפי אחריות)

---

## מבנה תיירות ממליץ 📁

```
soldier_courses_explorer/  
    app.py           # תפריט + דפסה יפה של תוצאות  
    config.py        # פרטי חיבור למסד הנתונים  
    db/  
        connection.py # get_connection()
```

```
load_csv.py           # CSV-לוג יקרה לטעינתה
queries.py          # SQL-כל שאילתותה
data/
courses.csv         # שסוקט-CSV-קובץ ה-
```

קובץ ה-`courses.csv` מכיל כותרת עמודות בפורמט:

```
institution,city,address,course,district,telephone,email
```

דוגמה לשורה:

יאיר אסודי מקצועות המים הצלה, יסוד המעליה, ת.ד. 99 מיקוד  
50121, הצלה בבריכות שחיה סוג 1, חיפה  
והצפוני yairassodi@gmail.com, 6937152,

---

## שלב 1 — מסד נתונים וטבלה ב-MySQL

לפני הרצת פיתון, צרו מסד נתונים וטבלה:

```
CREATE DATABASE IF NOT EXISTS soldier_courses_db
CHARACTER SET utf8mb4
COLLATE utf8mb4_unicode_ci;

USE soldier_courses_db;

CREATE TABLE IF NOT EXISTS courses (
    id INT AUTO_INCREMENT PRIMARY KEY,
    institution VARCHAR(255) NOT NULL,           -- שם הגוף המכשיר
    city VARCHAR(255) NOT NULL,                   -- ישוב
    address VARCHAR(255),                         -- כתובת מלאה
    course VARCHAR(255) NOT NULL,                 -- שם הקורס
    district VARCHAR(255),                        -- מחוז (חיפה והצפון וכו')
    telephone VARCHAR(50),                         -- טלפון / שלוחה
    email VARCHAR(255)                           -- כתובת מייל
);
```

---

## שלב 2 — דרישות פונקציונליות לכל אופציה בתפריט

להלן מה שכל סעיף בתפריט חייב לעשות.

---

## Load CSV into DB 1

מטרה: להכניס את כל השורות מ-`courses.csv` לתוך הטבלה `courses`.

משימות:

- לקרוא את ה-CSV בעזרת המודול `csv` של פיתון.
  - לדלג על שורת הכותרת.
- פרטירית `INSERT` על כל שורה להריץ פקודת:

```
INSERT INTO courses (
    institution, city, address, course,
    district, telephone, email
)
VALUES (%s, %s, %s, %s, %s, %s, %s);
```

העזרו בפקודות שהופיעו במצגת.

---

## Search Records by Institution Name 2

קלט מהמשתמש: מילת חיפוש (או חלק שם הגוף/המוסד).

שאילתת דוגמה:

```
SELECT id, institution, city, course, district, telephone, email
FROM courses
WHERE institution LIKE CONCAT( '%', %s, '%' )
LIMIT 50;
```

פלט:

הדף רשימה של תוצאות, כולל לפחות:

`id` •

`institution` •

`city` •

course •

(לבחירהכם telephone / email •

---

### Search Records by Course Name3

קלט מהמשתמש: מילת חיפוש בשם הקורס (לדוגמה: "הצלה").

שאילתת דוגמה:

```
SELECT id, institution, city, course, district, telephone, email  
      FROM courses  
     ('%', WHERE course LIKE CONCAT('%', %s  
                                ;LIMIT 50
```

פלט:

הצגת רשימת קורסים מתאימים, כולל שם הקורס, המוסד, העיר והמחוז.

---

### Find Most/Least Common Course4

המטרה: למצוא איזה קורס מופיע הכי הרבה פעמים ואיזה הכי מעט פעמים בטבלה.

:Most common course

```
SELECT course, COUNT(*) AS num  
      FROM courses  
     GROUP BY course  
    ORDER BY num DESC  
   LIMIT 1;
```

:Least common course

```
SELECT course, COUNT(*) AS num  
      FROM courses  
     GROUP BY course  
    ORDER BY num ASC  
   LIMIT 1;
```

דרישות:

- להציג גם את שם הקורס וגם את מספר הפעמים (`num`).

- אופציוני (אתגר):

- להציג 5 TOP קורסים הכי נפוצים.

- להציג גם רשימה של מוסדות שפעילים את הקורס הנפוץ ביותר.

---

## Show Course Count per District 5

מטרה: לראות כמה קורסים יש בכל מחוז.

שאילתת לדוגמה:

```
SELECT district, COUNT(*) AS num_courses
FROM courses
GROUP BY district
ORDER BY num_courses DESC;
```

פלט:

- טבלה קטנה של: `district | num_courses`
- אפשר גם להציג רק המוחז עם הכי הרבה קורסים והמוחז עם הכי מעט.

אתגר קטן:

- אפשר להוסיף אופציה משנה:

- עם פרמטר מהמשתמש: "Show courses for specific district"

```
SELECT institution, city, course, telephone, email
FROM courses
WHERE district = %s;
```

---

## Free SQL Query 6

מטרה: לחת למשתמש להריץ כל שאילתת `SELECT` שירצה על הטבלה `courses`

משימות:

1. לקרוא מנ' המשמשழחרוזת SQL.
  2. לוודא שהיא מתחילה ב-`SELECT`, `UPDATE`, `DELETE`, `DROP` (לא לאפשר `SELECT`, וכו').  
`( "sql.strip().upper().startswith( "SELECT" )` בדיקה פשוטה:
  3. אם השאלה תקינה — להריץ אותה עם `cursor.execute(sql)` (ללא פרמטרים).
  4. להדפיס את כל השורות שחזרו.
  5. אם השאלה לא מתחילה ב-`SELECT` — להדפיס הודעה שגיאיה מתאימה.
- 

## הפרדת קוד (ארQUITקטורה נקייה)

`app.py`

- מכיל את לולאת התפריט:

```
while True:  
    print_menu()  
    choice = input("Choose option: ")  
    # קרייה לפונקציה המתאימה ב-queries / load_csv
```

- לא מבצע שאלות בעצמו – רק קורא לפונקציות ב-`db/queries.py` וב-`db/load_csv.py`.
- 

`db/connection.py`

```
import mysql.connector  
from config import DB_CONFIG  
  
def get_connection():  
    return mysql.connector.connect(  
        host=DB_CONFIG["host"],  
        user=DB_CONFIG["user"],  
        password=DB_CONFIG["password"],  
        database=DB_CONFIG["database"]  
)
```

---

## db/load\_csv.py

- פונקציה שמקבלת connection או יוצרת אחד עצמה.
  - קוראת את `.data/courses.csv`
  - מרים INSERT פרמטרי לכל שורה.
  - מדפסה כמה שורות נטענו בהצלחה.
- 

## db/queries.py

כל פונקציה מקבלת conn ופרמטרים, למשל:

```
def search_by_institution(conn, keyword):  
    sql = """  
        SELECT id, institution, city, course, district, telephone, email  
        FROM courses  
        WHERE institution LIKE CONCAT('%', %s, '%')  
        LIMIT 50;  
    """  
    cur = conn.cursor()  
    cur.execute(sql, (keyword,))  
    return cur.fetchall()
```

באופן דומה:

- `search_by_course(conn, keyword)`
  - `get_most_common_course(conn)`
  - `get_least_common_course(conn)`
  - `get_course_count_per_district(conn)`
  - `run_free_query(conn, sql)`
-

