

פרויקט גמר סמסטר א' מבוא לבינה מלאכותית

רון ולדמן, ת"ז 213002629

מבוא ומטרה

הפרויקט נבנה על בסיס flask, נכתב ב-python ורץ על גבי שרת מקומי בפורט ברירת המחדל 5000. מטרת הפרויקט היא ליצור api שיאפשר ליוצרי טריוויה גישה קלה יותר ליצירה ומחיקה של שאלות (שכן הם שמורים בJSON). בנוסף גם לאפשר גישה נוחה ואינטואיטיבית להרצה של שרת מקומי עם ניתובים רלוונטיים לצורך הרחבת הפרויקט.

מבנה

הפרויקט מורכב מקובץ `main.py` שמכיל בתוכו את כל הקוד והניתובים (routes) ומאכלס גם כן את קובץ ה `trivia_data.json` וה `index.html` בהם נעזר. ניתן וכדאי להריץ את השרת מ `server.bat`.

.idea	14/04/2024 9:36	File folder	
templates	13/04/2024 14:13	File folder	
venv	12/04/2024 22:11	File folder	
.gitattributes	12/04/2024 22:06	Text Document	1 KB
main.py	13/04/2024 20:08	Python File	4 KB
README.md	12/04/2024 22:06	MD File	1 KB
server.bat	14/04/2024 9:33	Windows Batch File	1 KB
trivia_data.json	13/04/2024 21:59	JSON File	6 KB

routes

ה `http routes` הם הניתובים של הפרויקט למתודות השונות בהם ניעזר למשימות אשר ניתנו והם:

[GET http://localhost:5000/trivia](http://localhost:5000/trivia) - for all trivia questions

[GET http://localhost:5000/trivia/categories](http://localhost:5000/trivia/categories) - to list all categories

[GET http://localhost:5000/trivia/420](http://localhost:5000/trivia/420) - for a specific trivia question

DELETE <http://localhost:5000/trivia/delete/id> - for a specific ID

POST <http://localhost:5000/trivia/add> - specify a unique ID in JSON (see example below)

ניתן כמו כן לראות ניתובים נוספים שהוספתי לנוחות המשתמש דרך
הדפדפן ב localhost:5000

דרך פעולה

לאחר שפתחנו PyCharm נוודא בterminal שמותקנות או מיובאות
הספריות הבאות, בדגש על flask. (בהתקנה הראשונית סביר להניח
שנצטרך לעדכן את הקוק שלנו באמצעות `python -m pip install --upgrade pip`)

```
Terminal Local x
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

(venv) PS C:\Github\Python\python-trivia-api> pip install flask
```

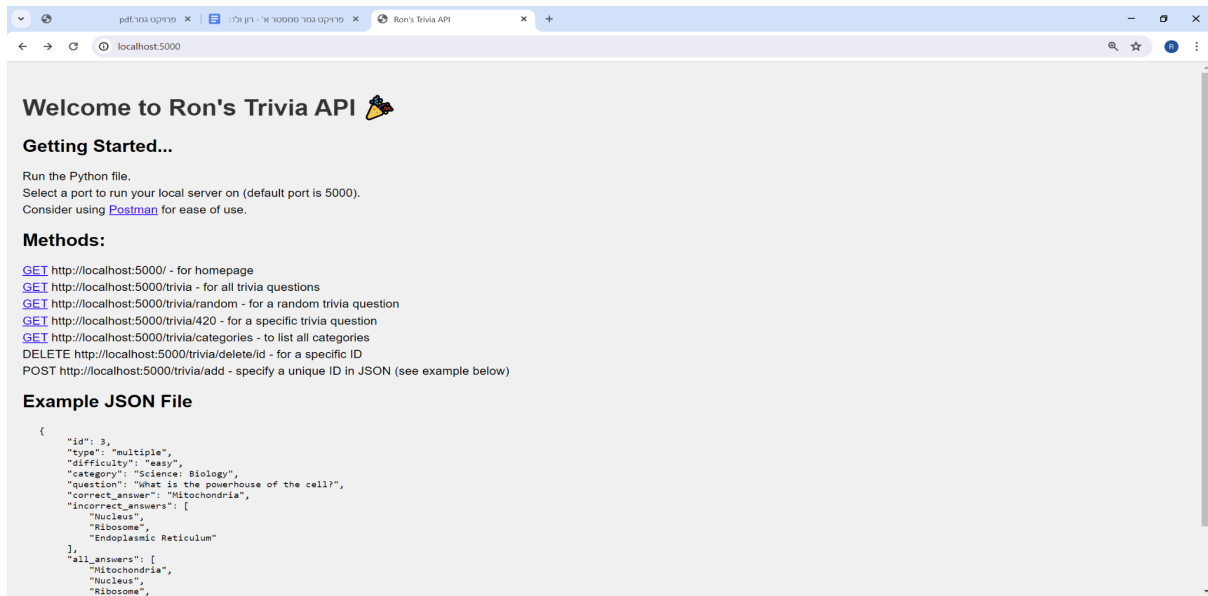
```
from flask import Flask, request, jsonify, render_template
import socket
import json
import os
import random
```

לאחר הרצת הפרויקט נתבקש להזין פורט עליו ירוץ השרת(ניתן להגדיר
מ65535-1). במידה ולא יוגדר פורט, השרת ירוץ על 5000.

```
Select a port for your Flask server:
Default server address is localhost:5000
* Serving Flask app 'main'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

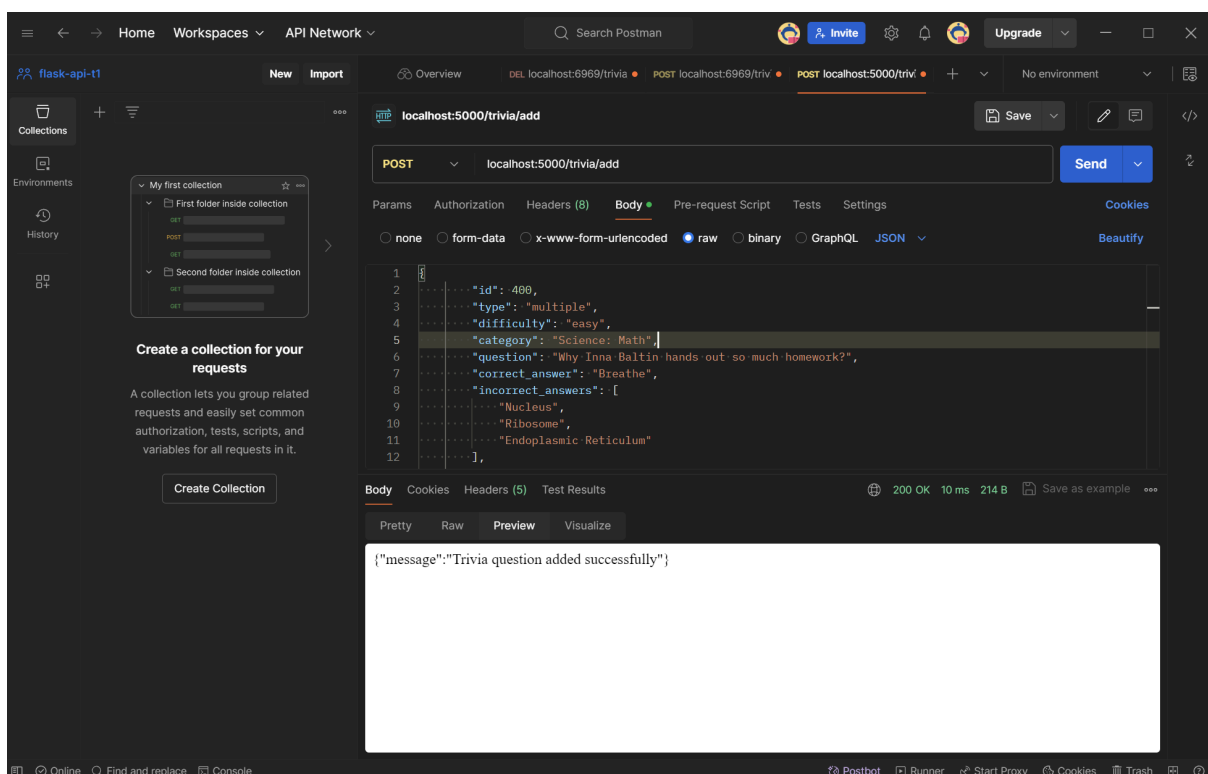
השרת ירוץ על localhost/127.0.0.1 נקודותיים הפורט שהגדרנו מעלה.
נוכל לבחור אם להיכנס לאתר ישירות או להיעזר בpostman(מומלץ
לdebugging).

כך נראה עמוד הבית (localhost:5000), בו נמצא לינקים לכלל חלקי הפרויקט. הוא נבנה בhtml פשוט לנוחות המשתמש. נוכל למצוא בו קטע json לדוגמא להדבקה במתודה בpostman.



שימוש בpostman להוספת טריוויה

נבחר במתודת POST ונרשום את הrouter המתאים לבקשה, קרי <http://localhost:5000/trivia/add>. הקוד יודע להתמודד עם דופליקטים בין אם בid או בתוכן ואינו מאפשר יצירת שאלה אם איננה מקורית. באופציה raw נדביק את הJSON לדוגמא ונקבל תשובה:



פעולה זו ואחרות (כגון DELETE) יעדכנו את קובץ ה JSON בזמן אמת.

```
{
  "id": 400,
  "type": "multiple",
  "difficulty": "easy",
  "category": "Science: Math",
  "question": "Why Inna Baltin hands out so much homework?",
  "correct_answer": "Breathe",
  "incorrect_answers": [
    "Nucleus",
    "Ribosome",
    "Endoplasmic Reticulum"
  ],
  "all_answers": [
    "Mitochondria",
    "Nucleus",
    "Ribosome",
    "Endoplasmic Reticulum"
  ]
}
```

טעינת JSON חיצוני בצורה גלובלית

הפרויקט משתמש ב JSON בקובץ נפרד וטוען (ושומר) אותו גלובלית לזכרון ברגע שיש צורך בכך דרך 2 פונקציות.

```
def load_json_data():
    with open(JSON_DATA_FILE, 'r') as file:
        return json.load(file)
```

2 usages Ron Valdman

```
def save_json_data(data):
    with open(JSON_DATA_FILE, 'w') as file:
        json.dump(data, file, indent=4)
```

load_json_data - טוען

save_json_data - שומר

לפניכם דוגמה לשימוש:

```
def is_question_duplicate(new_question):
    trivia_data = load_json_data()
    for question in trivia_data:
        if question['question'] == new_question['question']:
            return True
    return False
```

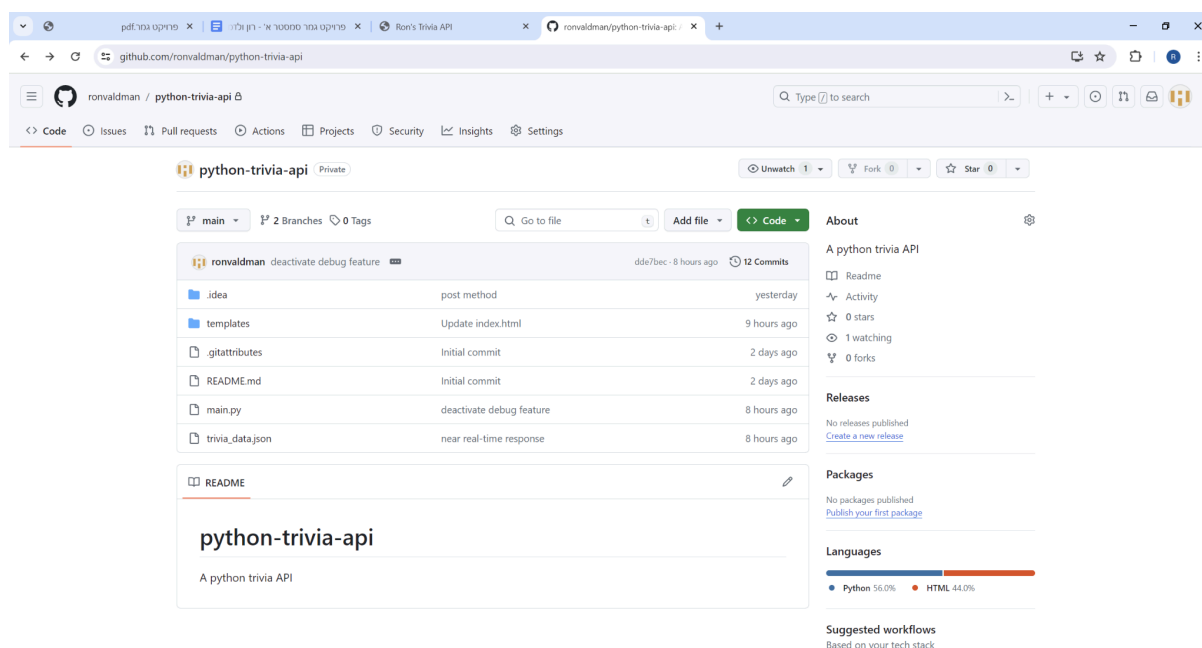
HTML

```

main.py  < index.html  trivia_data.json
2  <html lang="en">
3  <body>
4  </body>
5  </html>
6  </section>
7  <section class="section">
8  <h2>Methods:</h2>
9  <ul>
10 <li><a class="link" href="http://localhost:5000/">GET</a> http://localhost:5000/ - for homepage</li>
11 <li><a class="link" href="http://localhost:5000/trivia">GET</a> http://localhost:5000/trivia - for all trivia questions</li>
12 <li><a class="link" href="http://localhost:5000/trivia/random">GET</a> http://localhost:5000/trivia/random - for a random trivia question</li>
13 <li><a class="link" href="http://localhost:5000/trivia/420">GET</a> http://localhost:5000/trivia/420 - for a specific trivia question</li>
14 <li><a class="link" href="http://localhost:5000/trivia/categories">GET</a> http://localhost:5000/trivia/categories - to list all categories</li>
15 <li>DELETE http://localhost:5000/trivia/delete/id - for a specific ID</li>
16 <li>POST http://localhost:5000/trivia/add - specify a unique ID in JSON (see example below)</li>
17 </ul>
18 </section>
19 <section class="section">
20 <h2>Example JSON File</h2>
21 <pre class="example-json">
22 {
23   "id": 3,
24   "question": "What is the capital of France?",
25   "answer": "Paris"
26 }
27 </pre>
28 </section>
29 </body>
30 </html>

```

GITHUB INTEGRATION



סוף, תודה על הקריאה!