מה עושים היום:

ממשיכים עם pythonanywhere

נרים שם כמה services:

1. API (נקרא גם microservice) (מחזיר JSON)
2. שרת web \ אפליקציית web (מחזיר HTML)

API ראשון:

Notes

תרגיל כיתה (backend):

תעלו ל pythonanywhere API service שיש לו endpoint אחד:

/api/notes

שמחזיר JSON במבנה הבא (לשנות את התוכן של הפתקים):

[{"note": "note1"}, {"note": "note2"}, {"note": "note3"}]

מי שמסיים לשים ב chat את ה URL

דגשים:

1. שגיאות על CORS – להתקין pip install flask\_cors דרך console

של pythonanywhere

From flask\_cors import CORS

CORS(app)

1. להשתמש בספריית json של פייתון:

Json.dumps([])

תרגיל כיתה (frontend)

נכתוב אפליקציה SPA ב javascript שמציגה את ה Notes שפרסמנו ב API

תרגיל כיתה (frontend)

להוסיף לאפליקציה תיבת טקסט. המשתמש מדביק URL של API של Notes והאפליקציה מציגה את ה notes מה URL.

תרגיל הבא:

אפליקציה לא קשורה ל Notes

אפליקציה chat

המטרה: להוסיף גם SQLITE לסביבת הענן

קודם כל לפתח ב VSCODE לוקאלית:

Backend API service

* השירות מופעל ע"י URL:
* <https://tmanor.pythonanywhere.com/api/chat?message=hello>
* הוא מקבל message
* הוא יוצר יוזר רנדומלי ב session (אם עדיין לא קיים)
* הוא שומר את ה message וה user בטבלה ב sqlite
* מחזיר JSON עם רשימת ההודעות והמשתמשים:

[{"username": "user91", "message": "hi again"}, {"username": "user91", "message": "hello"}, {"username": "user27", "message": "hello"}, {"username": "user91", "message": "hello"}]

שלבים לפתרון:

1. לעשות route ב flask /api/chat
2. לחלץ ממנו את ה message, מתוך request.args
3. בודקים אם יש סשן:

From flask Import session, request, Flask

"username" not in session

1. אם לא קיים סשן, אז יוצרים סשן חדש:

Session["username"]=f"user{random.randrange(1,100)}"

1. יש לנו שם משתמש, יש לנו את ההודעה...אפשר לשמור ב DB
2. מכינים מראש טבלה עם 2 עמודות: שם משתמש והודעה ושם שומרים
3. שולפים את המידע מה DB ומציגים בתוך מבנה של רשימת מילונים