Erez Babai- 313328874

Home Assignment 4

Users – table's attributes:

user_name varchar(255),

user_nickname varchar(255),

user_email varchar(255),

user_age int,

user_Gender varchar(10),

user_ID int primary key,

user_password varchar(255)

כמו שסיכמתי עם ארסני הסתפקתי בעיצוב בסיסי ופונקציונליות מלאה של מטלת הבית.

פעולות שעשיתי:

PART A:

- תחילה יצרתי בסיס נתונים חדש בMySQL
- מימשקתי את בסיס הנתונים עם הpycharm- הורדתי את חבילת MySQL שם.
- יצרתי Blueprint חדש של ש.ב 4
- יצרתי את הroute הבסיסי שלנו שמעבר לעובדה שהוא מפנה אותנו לדף הבית הוא מדפיס את כל הusers שנמצאים בטבלה בבסיס הנתונים שלנו.
- יצרתי form לinsert אשר דורש את כל השדות של הuser שלנו (כמו שמופיעים למעלה), בנוסף יש בדיקה שאין חזרה על פרטים חד חד ערכיים כמו אימייל, nickname ומספר ת.ז- בטבלה למעלה רואים בעצם איך מימשתי את זה בSQL אך ההגבלות על השדות שהם לא ת.ז והשמירה עליהם כחח"ע קורה בfrontend.
- יצרתי טופס עדכון שבו מכניסים ת.ז וניתן לשנות את מין הלקוח והסמא שלו, כמו כל שאר הפורמים גם פה קופצת התראה כאשר הפעולה בוצעה/ נכשלה.
- יצרתי טופס מחיקה, מכניסים ת.ז ומוחקים משתמש.
- כל לחיצה על הsubmit של הפורמים תחזיר אותנו לעמוד הבית עם רשימת היוזרים המעודכנת.

PART B:

- יצרתי את הקובץ הנ"ל וצירפתי אליו את הגדרת התכונות מהטבלה Users.
- יצרתי route חדש users/ אשר מציג את כל היוזרים שלנו בפורמט ג'ייסון – ניתן להגיע אליו גם מהnav.
- יצרתי route חדש outer_source/ אשר מציג דף html חדש ובו שתי תיבות חיפוש אחת frontend והשניה backend
- מימשתי את הפונקציה backend דרך assignment4.py. מנעתי מצב בו המערכת קורסת בגלל שמכניסים בה מספר ID לא קיים. על מנת לעשות את זה ייבאתי את תיקיית requests ל python שלנו.
- מימשתי את הפונקציה frontend דרך הjs.

Part C:

- יצרתי route חדש assignemt4/restapi_users.
- הגדרתי פונקציה שמעליה פעמיים קריאה לroute הזה כנלמד בשיעור, הקריאה הראשונה עם ערך דיפולטיבי משמע כשלא מכניסים (ואז מוצג משתמש שבחרתי) והאופציה השנייה היא כאשר כן מכניסים ID (דורש int) ואז זה מציג יוזר לפי ה ID הרצוי.
- כל יוזר שהמערכת תראה יהיה בפורמט JSON.
- במקרה של הכנס יוזר לא נכון יוצג המסך הבא:

```
{
  "ERROR": "We couldnt find the user"
}
```