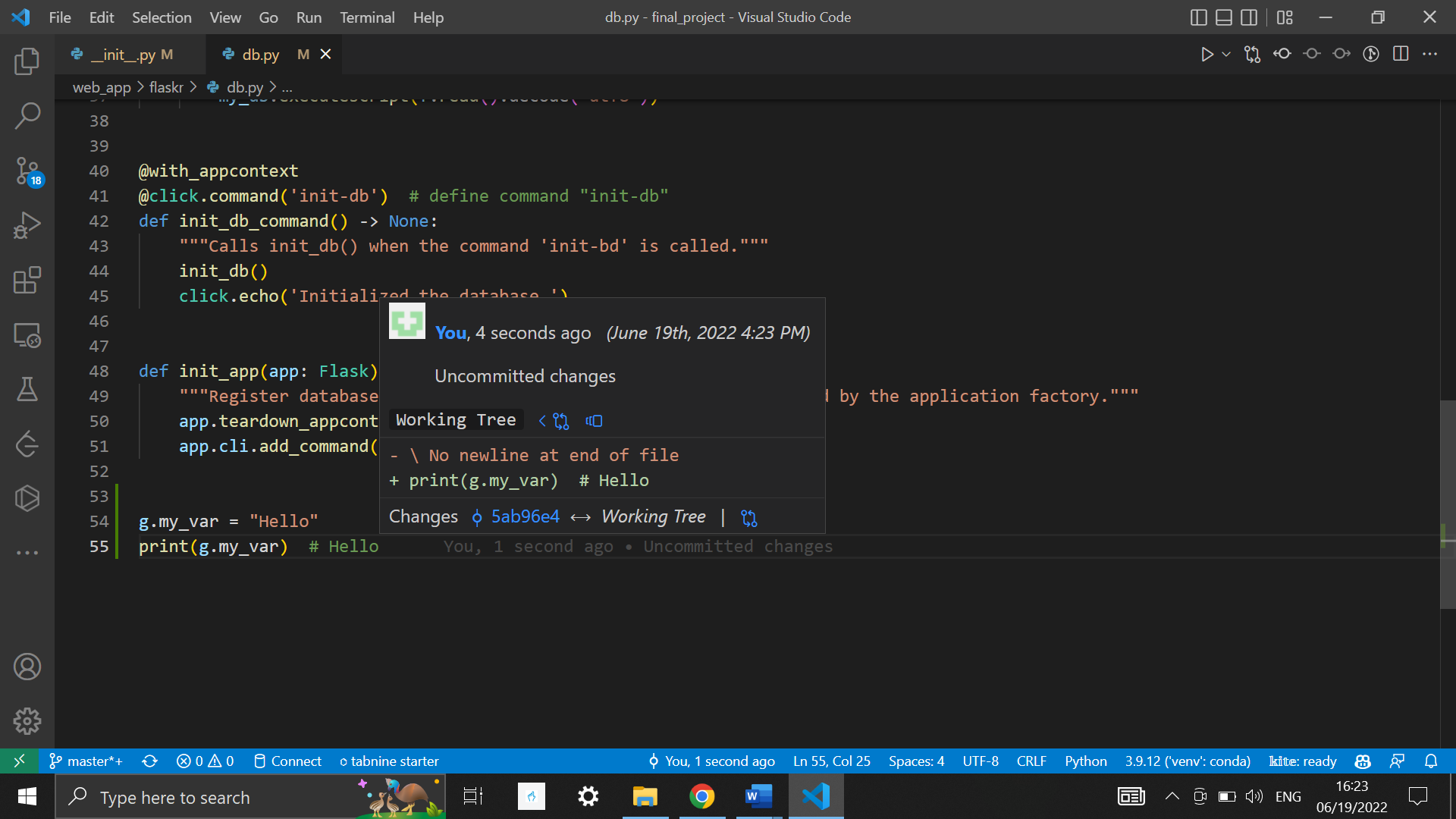
**החלק המחקרי**

**אפליקציית ווב**:

כל אפליקציה שמפותחת באמצעות Flask חייבת לכלול קובץ ששמו: "\_\_init\_\_.py" ובו פונקציה הנקראית "create\_app" בלבד. פונקציה זאת נקראת כשהשרת מתחיל להריץ את האפליקציה. הפונקציה מבצעת את הפעולות הבאות: יצירת אובייקט האפליקציה שיורש מהמחלקה Flask , הגדרת קונפיגורציה (למשל מיקום בסיס הנתונים), יצירת תקייה לבסיס הנתונים, יצירת בסיס הנותנים ושמירת blueprints.

המשתנה הגלובלי g הוא מוגדר כשהמשתמש נכנס לאפליקציה. הוא ריק כל עוד לא מכניסים אליו שום דבר. אפשר לשמור בו משתנים לפי שם ולגשת בו לכל משתנה לפי שם:



בקובץ db.py נמצאות הפונקציות שאחראיות על ניהול בסיס הנתונים:

הפונקציה get\_db() בודקת אם קיים חיבור לבסיס הנתונים (אובייקט מהמחלקה sqlite3.Connection) במשתנה הגלובלי ואם לא, יוצרת אחד כזה ושומרת אותו ב g.my\_db ולאחר מכן (בלי קשר לתנאי הראשון) מחזירה את g.my\_db . חשוב: בפייתון משתנים מוחזרים ב referenceככה שכל פונקציה שקוראת ל get\_db מקבלת גישת עריכה למסד הנתונים ויכולה לשנות אותו באמצעות פקודות SQL כמו DELETE, INSERT .

הפונקציה init\_db() מקבלת חיבור למסד הנתונים ומפעילה את פקודות ה SQL בקובץ schema.sql. הקובץ schema.sql מכיל את הפקודות הבאות:

אם קיימות טבלאות בשמות: completion, model, user מחק אותן.

צור את הטבלאות הבאות (כל הטבלאות בבסיס הנתונים):

user (המאחסנת משתמשים) עם העמודות:

id - מספר סידורי: שלם של המשתמש שהוא המפתח הראשי של הטבלה.

username – שם משתמש: טקסט ומיוחד.

password – סיסמה מוצפנת: טקסט (הסיסמה מוצפנת לפני שהיא נכנסת לבסיס הנתונים).

model המאחסנת מודלי למידה עמוקה עם העמודות:

Id - מספר סידורי: שלם של המודל שהוא המפתח הראשי של הטבלה.

User\_id – מספר סידורי של המשתמש שהעלה את המודל לאתר (כל משתמש יכול להעלות מספר מודלים לאתר וככה לאפשר לכל המשתמשים להשתמש בהם) בין מודל למשתמש יש קשר רבים לרבים.

model\_name – שם הקובץ של המודל: טקסט.

Created – הזמן בו נוצר המודל. ברירת המחדל היא הזמן בו המודל נכנס לבסיס הנתונים.

בנוסף, לכל אחד מהיפר הפרמטרים של המודל יש עמודה בטבלה זו. שמות היפר-הפרמטרים הם: set\_size, batch\_size, learning\_rate, num\_layers, d\_model, dff, num\_heads, dropout\_rate.

completions – השלמות. כל משתמש באתר יכול לבחור מודל מהמודלים הקיימים ולהכניס טקסט ולראות איך המודל משלים את הטקסט. העמודות בטבלה הן:

User\_id - מספר משתמש, model\_id – מספר מודל, created – הזמן בו נוצרה ההשלמה, prompt – הטקסט שהמשתמש הכניס למודל ו answer – הטקסט שהמודל פלט.

הקובץ auth.py מכיל את ה blueprint auth שכולל את הפונקציה login ו register שמציגות את עמודי הכניסה והרשמה לאתר. כל אחת מהפונקציות קולטת את שם המשתמש והסיסמה מעמוד ה HTML. אם המשתמש נרשם בהצלחה הוא מועבר לעמוד ההתחברות ואם הוא התחבר בהצלחה הוא מועבר לעמוד completion.index.

הפונקציה logout מנקה את הסשין ומעבירה את המשתמש לעמוד completion.index.

הפונקציות login, logout, register מופעלות (ללא פרמטרים) כשהמשתמש נכנס לקישור auth/login, auth/logout, auth/register בהתאמה.

בנוסף, הקובץ מכיל את הגדרת הקשטן (decerator) login\_required כשקוראים לפונקציה המקושתת בו, הוא בודק שיש משתמש במשתנה הגולבאלי (g) ואם לא, הוא מעביר את המשתמש לעמוד ההרשמה. הוא מקשט את הפונקציות של העמודים הדורשים התחברות למערכת.

הקובץ מכיל גם את הפונקציה load\_logged\_in\_user שנקראת באופן אוטומטי כשמשתמש מגיע לעמודים register, login או logout הבודקת אם המשתמש שמור בסשין ואם כן שומרת או במשתנה הגלובאלי.

הקובץ model.py מכיל את ה blueprint model

הפונקציה view\_all טוענת את העמוד בו המשתמש רואה את כל המודלים שהועלו לאתר על כה. הפונקציה upload טוענת את העמוד בו המתמש מעלה מודלים לאתר.

הפונקציות allowed\_file, file\_errors, model\_errors הן פונקציות עזר ל upload.

הקובץ creation.py מכיל את ה blueprint creation.

הפונקציה create טוענת את העמוד בו המתמש מעלה מ