**תיעוד צד שרת**

**הקוד:**

המודל: apprun

זהו מודל סטטי שנועד לצורך הרצת אפליקציית ה Flask עם אפשרות למתן פרמטרים ספציפיים לצרכי בדיקות.

מובא לידי שימוש בעיקר בהרצת development

port: נועד למתן פורט לשרת (8443 יבחר כדיפולטיבי מישום שהפורט הדיפלוטיבי 443 דורש הרשאות sudo)

host: כתובת IP להאזנה (localhost יבחר כדיפולטיבי)

המודל: wegiApp.py

מודל שמטרתו היא ליצור ולהריץ אפליקצית wegi של gunicon שמהווה מעטפת לאפליקצית flask הרגילה. משומש רק בהרצת המערכת ב production

הספרייה AppLibrary

ספריה שיוצרת אפליקציית Flask ומחברת אליה את כל הכתובות (URL) ומימושי הפונקציות שלהן.

המודל AppLibrary.urls

מכיל את כל כתובות ה URL שהשרת מאזין להם.

compile: מאזין לבקשות post ו get מצפה למידע טקסטי מסוג JSON מתוכו הוא מחלץ את הפקודה מן הערך script מבצע אותה, ומחזיר תוצאות ללקוח.

test: מאזין לבקשות post ו get ומחזיר תמיד את הטקסט "true" נועד לצורך בדיקת עמידות השרת ויכולת התחברות אליו.

הספריה AppLibrary.SafeEval

נועדה צורך קימפול פקודות Python בסביבת עבודה בטוחה ללא מתן גישה לביצוע פעולות לא רצויות או קריאה ליצירתם של פעולות לא רצויות.

האוביקט safeDict: אוביקט מסוג מילון, שמכיל שם פעולה כמפתח, וכתובת בזיכרון של המימוש עצמו. נועד לצרכי מתן גישה לפעולות ספציפיות בלבד, שכוללות את כל הפונקציות המתמטיות של מחלקת math, הפקודה של ערך מחולט (abs) וכמובן פעולת ההדפסה print

הפעולה get\_safe\_functions: פקודה שנועדה להדפסת כל הפונקציות המותרות לשימוש (קריאה על ידי בקשה מן המילון לערך safe\_fucntions)

הפעולה calcuateEval: פעולה שיוצרת סביבה הומוגנית שבה אין שום מודל או אפשרות לבניה של מודלים, וכן רק הפעולות המותרות. הפעולה מריצה את פקודת הפייטון הטקסטואלי ונותנת גם אופציה ליצירת ערכים על גבי הזיכרון (הנחה: בצד לקוח הגבילו את גודל הקלט) ומחזירה כטקסט את פלט הפקודה (אם קיים)

הקבצים cert.pem ו key.pem

קבצי קריפטוגרפיה שנועדו להצפין את התקשורת המתבצעת על ידי ה socket ונוצרו על ידי ספרית openssl וממושים בשכבת הבניה של השרת.

**הרצה:**

ישנם שלושה קבצי הרצה: שנים על גבי מערכת ubuntu ואחד על גבי window חשוב שהמערכת שאמורה לרוץ על גבי המוצר הסופי (production) יהיה במערכת ubuntu

כל אחד מקבצי ההרצה מתקין את הספריות הדרושות מקובץ ה AppLibrary/ requirements.txt ומריץ את האפליקציה על גבי שרת מתאים (Flask או Gunicorn) עם שימוש במודל app.

(הערה: המערכות כולם נבנו ונבדקו ב Python 3.6+ גם ב Windows וגם ב Ubuntu וחלק מהספריות אינם פועלות כראוי בגרסאות ישנות יותר של השפה כגון SSL בגרסה 3.5)