**עבודת הגשה 2 - תקשורת ורשתות מחשבים**

מגיש - עמית יהושפט, 212359442

לפני שאתחיל לפרט של הקוד עצמו -

דוגמאות קלט ופלט: שמתי בתיקיה Test Pictures אפשר גם לראות את התהליך בהתחלה נתתי לקליינט לדבר רק עם השרת ולאחר מכן שיפרתי אותו שידע לדבר בין קליינטים אחרים - אפשר לראות את ההתקדמות אם נכנסים להיסטוריית הcommit בגיטהב.  
הוראות התקנה והרצה: ההוראות הברורות הפשוטות נמצאות בקובץ הReadme שנמצא בגיטהב וגם בקובץ.  
  
**מבנה הקוד מונה 2 קבצים:**

**קובץ ראשון, Server.py:**

זה הוא הקובץ המרכזי המחזיק את כל צד השרת, הוא פותח סוקט מסוג TCP ומצפה לקבל כתובת IP מסוג IPV4. הקובץ מונה 2 פונקציות עיקריות שאפרט עליהם כאן:

\*הפונקציה Run - הפונקציה שתכלס מריצה את השרת, הפונקציה מאזינה בכתובת ובפורט שהוגדרו ושומר את נתוני הלקוח באמצעות accept() שמחזיר את האייפי והסוקט שלו.  
בנוסף השתמשתי בThread המאפשר לנו לנהל מספר קליינטים בו זמנית, שם אנחנו קוראים לפונקציה handle\_client שעליה אפרט בהמשך.  
כמובן שברגע שהכל מסתיים (finally) השרת משתחרר בצורה מסודרת.

\*הפונקציה handle\_client - הפונקציה מבקשת מהלקוח להזין שם יחדוי, אם השם תפוס או ריק הוא מתבקש להזין שם חדש.  
הודעה מהלקוח מתקבלת באמצעות recv עם קידוד utf-8, אם ההודעה מתחילה ב@ אז השרת מפרק את ההודעה כדי להבין מי הלקוח שאליו היא מיועדת, אם הלקוח קיים אז ההודעה נשלחת אליו, ואם לא? אז השולח מקבל הודעה שהיעד אינו נמצא.  
אם הלקוח מתנתק אז הפונקציה מסירה אותו מרשימת הלקוחות וסוגרת את החיבור שלו.

**קובץ שני, clientTest.py:**

זה הוא קובץ הלקוח שמתחבר לשרת ומאפשר שליחת וקבלת הודעות, גם קובץ זה מחולק לשני פונקציות שאפרט:

\*הפונקציה run- הפונקציה אחראית על חיבור הלקוח לשרת והתחלת התקשורת, היא מבקשת קלט מהמשתמש ומאפשרת לו לשלוח הודעה בפורמט שציינתי קודם @שם\_יעד (תוכן).  
הפונקציה דואגת להעביר את ההודעות לשרת שמטפל בהן ומעביר אותן ליעד המתאים.

\*הפונקציה recieve\_messages- הפונקציה האחראית להאזין להודעות שמגיעות מהשרת, היא פועלת בלולאה אינסופית כדי להבטיח שכל הודעה שהשרת שולח תתקבל ותודפס על המסך מבלי לפספס אף אחת, במידה ויש תקלה או שהחיבור לשרת נקטע, הפונקציה סוגרת את החיבור בצורה מסודרת.

לסיכום:

בעבודה זו פיתחתי מערכת צ'אט מבוססת פרוטוקול TCP, תוך שימוש בשני קבצים נפרדים: קובץ לשרת וקובץ ללקוח. המערכת מאפשרת תקשורת דו-כיוונית בין לקוחות דרך השרת, כולל ניהול שמות ייחודיים ותיווך הודעות. במהלך תהליך הפיתוח למדתי רבות על עקרונות תקשורת ברשת וכתיבת קוד אפקטיבי ב-Python, נהניתי מאוד מתהליך כתיבת הקוד.