

הסבר כללי על המערכת ומבנה הקוד

מערכת הצ'אט מבוססת פרוטוקול TCP ומיועדת לניהול תקשורת טקסטואלית בזמן אמת בין מספר משתמשים המחוברים לשרת מרכזי. המערכת מתוכננת באופן מודולרי כדי לאפשר קריאות, ניהול חיבורים ותיעוד של פעולות המשתמשים.

✓ רכיבי המערכת:

1. שרת (server.py):

- מנהל חיבורים רבים בו-זמנית (ריבוי תהליכים באמצעות Threads).
- מתווך בין הלקוחות, מפץ הודעות לכל המשתמשים.
- מטפל במקרים של ניתוק לא צפוי.
- מתעד את הפעולות (חיבור, ניתוק, ושליחת הודעות) בקובץ chat_log.txt.

2. לקוח (client.py):

- מאפשר למשתמש להתחבר לשרת באמצעות שם משתמש ייחודי.
- שולח הודעות לשרת ומציג את ההודעות המתקבלות.
- מודיע למשתמש אם השרת התנתק או אם לקוח אחר עזב את השיחה.

3. קובץ תיעוד (chat_log.txt):

- זמן החיבור והניתוק של כל משתמש.
- תוכן ההודעות שנשלחו במהלך השיחה.

✓ תכונות עיקריות:

1. תקשורת דו-כיוונית:

- משתמשים יכולים לשלוח הודעות טקסט בזמן אמת ולקבל תשובות מלקוחות אחרים דרך השרת.

2. תיעוד שיחה מלא:

- כל האירועים (חיבורים, ניתוקים, והודעות) נרשמים בקובץ לצורך מעקב.

3. ניהול חיבורים:

- תמיכה במספר חיבורים בו-זמנית תוך טיפול בשגיאות וניהול משאבים נכון.

4. טיפול בשגיאות:

- השרת מזהה מקרים של ניתוק לקוח או תקלה ברשת ומשדר הודעות מתאימות ללקוחות.

✓ מבנה הקוד:

1. פונקציות מרכזיות בשרת:

- `handle_client` – מנהלת כל לקוח בנפרד.
- `broadcast` - שולחת הודעה לכל הלקוחות המחוברים
- `log_chat_activity` - מתעד את הפעולות בקובץ התיעוד.

2. פונקציות מרכזיות בלקוח:

- `receive_messages` - מאזינה להודעות מהשרת.
- `send_messages` - שולחת הודעות לשרת.

✓ שימושים אפשריים:

- מערכות תקשורת פשוטות בין משתמשים.
- מערכות לניהול לוגים או בקרה של משתמשים מרובים.