

מגישים: 1. אושרי הלוי ת"ז 318830569 2. אוראל כהן ת"ז 214266819

### הסבר כללי על המערכת:

המערכת היא שרת צ'אט המאפשרת תקשורת בזמן אמת בין משתמשים שונים. המשתמשים יכולים להתחבר לשרת, להיכנס עם שם משתמש, לשלוח הודעות ולנהל שיחות פרטיות אחד עם השני.

### המערכת בנויה משני חלקים עיקריים:

1. Server: אחראי לניהול החיבורים, קבלת ההודעות מהלקוחות, ניהול השיחות ביניהם. השרת שומר את כל ההודעות שנשלחות בקובץ JSON.
2. Client: כל משתמש מתחבר לשרת ומבצע אינטראקציה עם המערכת דרך ממשק טקסטואלי. הלקוח יכול לשלוח בקשות לשיחות פרטיות, לשלוח הודעות למשתמשים אחרים ולסיים את השיחה.

### מבנה הקבצים:

- `chat\_Server.py` – הקובץ המנהל את השרת, כולל כל הפונקציות שמטפלות בחיבור עם הלקוחות.

- `chat\_Client.py` – הקובץ המנהל את הלקוח, מאפשר למשתמש לשלוח ולקבל הודעות דרך הממשק.

### מבנה הקוד:

#### 1. ChatServer.py:

השרת אחראי לניהול החיבורים וההודעות בין המשתמשים:

- `\_\_init\_\_(self, host, port)`: פונקציה זו מאתחלת את השרת. היא מקבלת את כתובת ה-IP והפורט של השרת. אם קובץ ה-JSON שמכיל את ההודעות לא קיים, היא תיצור אותו.

- `start(self)`: זו הפונקציה שמתחילה את השרת. היא מאזינה לחיבורים נכנסים, ומחלקת כל חיבור שנכנס לחוט נפרד.

- `handle\_client(self, client\_socket)`: הפונקציה הזו מטפלת בחיבור של כל לקוח. היא מקבלת את שם המשתמש, מטפלת בהודעות שנשלחות מהלקוח ומספקת תגובות מותאמות אישית.

- `log\_message(self, from\_user, to\_user, content)`: פונקציה זו שומרת את כל הודעה שנשלחת בקובץ JSON. כל הודעה שנשלחת נשמרת יחד עם המידע של מי שלח את ההודעה, למי היא נשלחה, התוכן ותאריך/שעה.

- `listen\_for\_commands(self)`: מאזינה לפקודות משרת המערכת.

- ``shutdown_server(self):` סוגרת את השרת בצורה מסודרת כאשר הפקודה ``shutdown/` מתקבלת.

2. `ChatClient.py`:

הלקוח מחובר לשרת ומאפשר למשתמש לשלוח ולקבל הודעות:

- ``__init__(self, host, port):` אתחול הלקוח, הגדרת כתובת ה-IP והפורט של השרת שאליו יתחבר.

- ``start(self):` פותחת חיבור עם השרת, שולחת את שם המשתמש לשרת ומחילה את החוט המתאים להאזנה להודעות נכנסות.

- ``command_loop(self):` פונקציה זו מאזינה לפקודות שהמשתמש שולח.

- ``receive_messages(self):` מקבלת הודעות מהשרת ומדפיסה אותן לממשק של המשתמש.

- ``send_message(self, message):` שולחת הודעה לשרת, מבצעת סדרה של המרה לקובץ JSON ושליחה בפקודת `socket`.

#### איך המערכת עובדת:

1. הפעלת השרת: השרת מאזין לחיבורים נכנסים מלקוחות ומתחיל להקשיב לפקודות.
2. הפעלת הלקוח: הלקוח מתחבר לשרת ומזין שם משתמש.
3. שליחת הודעות: הלקוח יכול לשלוח הודעות למשתמשים אחרים. השרת מנהל את כל ההודעות ומעביר אותן בין הלקוחות.
4. שיחות פרטיות: הלקוח יכול לשלוח הודעה למשתמש אחר. השרת בודק אם המשתמש קיים ומעביר את ההודעה.
5. סיום שיחה: ברגע שהלקוח רוצה לצאת, הוא יכול לשלוח את פקודת ``quit/` והחיבור ייסגר.