

**פרויקט גמר ברשותות מחשבים ו인터넷 –  
83455-01**

מגיש: דניאל נגבי

ת.ז 326709144

תאריך הגשה סופי: 17.3.2024

מערכת הפעלה : Windows 11 Home, v. 23H2, 64bit

שפת תכנות : python v. 3.12.2 אך העבודה נעשתה גם מהמחשב הנייד  
האישי שלי בגרסה 3.10

## הסבר כללי על הפרויקט

אתה תתחיל בלהסביר בקצרה על הפרויקט.

בפרויקט זהה מימושי, כמו שבקשו, קוד שמאפשר את האפשרות לדבר עם יוזרים אחרים בתוך חדרי טקסט תוך כדי היישנות על סרבר. היו שלל דרישות, בקשות ודקויות בנוגע

לפרויקט אך ניתן לנו בחלק במקומות יד חופשיות ואמירות שניתנות למגוון פירושים.

אסביר על המימוש שלי במסמך זה ובנוסף אענה על שאלות החקלאות התיאורטי. כתבתי את הקוד בפייתון.

לאחר שמאתחלים את השרת ופותחים אותו, ומריצים את הקליענט רץ ומתחבר אל הסרבר (חיבור סטנדרטי בעזרת סוקטים בדיק כמו שראינו).

אתה נכנס ומוליך מופיעו תפריט הכנסה. בפרויקט זה עשייתו שני תפריטים מרכזיים, שיקראו "תפריט הכנסה" - שעליו אפרט מיד, והתפריט השני יקרא "התפריט הראשי" שעליו אסביר בהמשך. לאחר שקליענט מתחבר אל השרת, הוא מוצג עם תפריט הכנסה, התפריט הראשון. בתפריט הכנסה ישנן ארבע אפשרויות:

1. הרשמה – register :

להירשם אל הסרבר.

2. התחברות – login :

להתחבר לאתר אם אתה משתמש קיים.

3. יציאה – exit :

סגור את החיבור של המשתמש עם השרת.

4. להירשם בתור אדמין – register as admin :

להירשם אל הסרבר, אך אתה מקבל את הרול של אדמין. בהמשך אפרט על הרולים הקיימים.

את הבחירה עושים באמצעות הקשת המספר המתאים.

כאשר אתה נרשם, הן בתור יוזר רגיל והן בתור אדמין, אתה נדרש להזין שם משתמש וסיסמה שתרצה בשבייל המשתמש שאתה יוצר ולאחר מכן מכניס את הפרטים, נבדק אם המשתמש קיים כבר. אם כן מקבל הودעה מתאימה. אם לא, ההרשמה תבוצע בהצלחה והודעה מתאימה תתקבל. לאחר ההרשמה תחזור אל תפריט הכנסה.

שם המשתמש והסיסמה שאתה מכניס נרשמים אל תוךקובץ csv בשם users.csv שנוצר אם איןנו קיימים. אם קיימים אז נוספיםים שם המשתמש, הסיסמה (מודפנת בעזרת hash) והרול של היוזר. אפרט יותר בהרחבה על שמירת המידע בהמשך.

כasher אתה בוחר להתחבר (כולם 2 - login), אתה נדרש להכנס שם משתמש וסיסמה. כמובן שבדיקה נעשית אם היוזר אכן קיים (בודקים האם שם המשתמש והסיסמה המוצפנת מתאימות לזוג שקיים במאגר), אם המשתמש עם הסיסמא אכן קיים בדתאטייס, אז תיכנס בהצלחה. לאחר מכן תקבל הודעה מתאיימה שאחד מהפרטים אינם נכונים.

לאחר שנכנסת בהצלחה, תועבר אל התפריט הראשי. התפריט הראשי שונה עבור יוזר גיל ועבור אדמין.

#### אפשרויות בתפריט הראשי שקיימות לשני הולמים:

1. לראות את רשימת החדרים הקיימים שאלייהם אפשר להתחבר:  
אם תלחץ 1, אתה תועבר לרשימת החדרים הקיימים שאלייהם אפשר להתחבר,  
בכדי להיכנס לחדר שאתה רוצה, פשוט תקליד את השם שלו ואז תיכנס אל החדר.

2. לשנות סיסמא:  
אם תלחץ 2, אתה תוביל לבקשה שבקשת ממך להזין את הסיסמא החדשה שאתה  
תרצה להתאים אל המשתמש שלך, נשים לב שאנחנו כבר מחוברים למשתמש  
בשלב זה ולכן הסרבר יודע כבר מי מחובר, מה הפרטים שלו (שם משתמש,  
סיסמא, רול) ולכן אין בעיה להחליפם.

3. לראות את רשימת האנשים שהחוברים לחדרים:  
אם תלחץ 3, אתה תקבל רשימה של שמות משתמשים של אנשים שכעת מחוברים  
לחדר כלשהו, רק נשים לב שמיד לאחר שזה שולח את הרשימה, זה מיד שולח  
שוב את התפריט שתוכל לבחור אופציה אחרת.

כעת, אם היוזר שלך הוא עם רול של אדמין, יהיה לך שתי אפשרויות נוספות שאין  
מופעיות וכמובן לא ניתנות לבחירה עבור יוזר גיל:

4. יצירת חדר חדש:  
אם אתה אדמין, תופיע עבורך האפשרות ליצור חדר חדש. אם תבחר 4 בתפריט  
אתה תתקבש לרשום את השם של החדר החדש, החדר יוצר ותתקבל הודעה  
שהחדר נוצר בהצלחה.

5. מהיקת חדר קיים:  
אם אתה אדמין תוכל לבחור 5 (להקליד 5) ואז תתקבש לרשום את השם של  
החדר אותו תרצה למחוק, ביחד עם המהיקה ימחקו כל הקבצים הרלוונטיים  
שנוצרו עבור החדר. אפרט עליהם המשך.

בדומה לרשימת משתמשים, אני שמרתי בקובץ csv את הרשימה של החדרים הקיימים (וגם בדראקטורי, בשבייל האפשרות ל-threading ושהשכלה יהיה חדר לדבר בו). אם יוצר בוחר 1 בתפריט הרשימה זו תוצג לו.

אם בתפריט זה אתה תרשום exit/, החיבור ייסגר בין השרת ליוור. כאשר אתה בתוך חדר, כדי לצאת ולבוער חדר, ניתן לרשום שוב exit/ ובכך ניתן לעبور בחדרים כרצון המשתמש.

בכדי לקבל רשימה פקדות שימושיות (קיימות שלוש פקודות) ניתן לרשום בתוך חדר את הפקודה help/ ואז תקבל רשימה פקדות.

לבסוף על דרישות המטלה:

#### סרבר:

- בנה צ'אט סרבר שתומך במספר משתמשים עם אוטנטיקציה מימשתי כМОובן בעזרת threading וסוקטים, כל Thread הוא בעצם חיבור בין יוזר לחדר, ובכך כמה אנשים יכולים לדבר במקביל בליבר בתוך כל חדר. האוטנטיקציה נעשית כאמור אל מול קובץ csv שבוני (או נוצר) בפרויקט.

- קליניטים מוכחרים להכניס שם משתמש וסיסמה בכדי להתחבר : ממושך באמצעות אפליקציית login, הדרך היחידה להיכנס אל השרת היא להכניס שם משתמש וסיסמה שקיים במאגר.

- בנה רולים ליוורים:  
בניתי שני רולים בשרת, רול של משתמש רגיל (user – יוזר) ומנגדו יש רול של אדמין בשרת (admin). ההבדל בין הרולים הוא שאדמין יכול ליצור ולמחוק חדרים כראות עיניו, בעוד ששתי אפישויות אלה חסומות ליוור (היו תכניות להוסיף עוד פיצרים אך נגלו עקב בעיות זמן). כמו שאמרתי, בהתאם להרשמה אתה בוחר האם ליצור משתמש חדש בתחום אדמין, המידע נשמר בהתאם בcsv ששם ירשם מה הרול, ובעת ההתחברות לשרת עם שם המשתמש והסיסמה המתאימים אתה מקבל את התפריט הראשי עם האפשרויות שמתאימות לרול שלך.

- צור מערכת אחסון מידע עקבית לשמיירת כל הדטה בסרבר, כולל הודעות של צ'אטים, תציג שהמידע יישמר גם אם השרת סגור או מאוחחל: נתחיל בשמירה על פרטי יוזרים וקבוצות. כמו שכחתי לעיל, את שמות המשתמשים עם הסיסמות (המווצפנות) והרולים המתאימים לכל אחד שמרתי בקובץ csv, כל שורה מכילה את שלושת פרטיים אלו של משתמש קיימים. השרת יודע להוסיף יוזרים חדשים לקובץ וכמוון שהקובץ אינו נמחק או נסגר בעת סגירת השרת. אם הקובץ הראשוני אינו קיים השרת יודע ליצור חדש הנו לקבוצות והן לקובץ המשתמשים.

עבור שבירת ההודעות השתמשתי בקבצי log, לכל חדר שנוצר ונשלחת בו הודעה נוצר לוג משלו, בלוג נשמרות כל ההודעות שנשלחו בחדר הספציפי זהה, ברגע שהשרות צריך לטעון חיבור חדש בחדר (כלומר יוצר בחדר להיכנס אליו), השרת מעלה קודם כל את לוג ההודעות, ומדפיס הודעה מתאימה ששוחרר את כל ההיסטוריה ההודעות. הלוג כמובן אינו תלוי במצב השירות, ונשמר גם לאחר הפעלה חדשה של השירות. כך יוצר יכול להתחבר אל חדר ומיד לקבל את היסטוריית ההודעות בחדר זה.

- **בנה מערכת שתגביל את כמות ההודעות שיוורים יכולים לשולוח בזמן נתון ב כדי למנווע הספמה :**  
מיימשתי את דרישת זו בעזרת בניית דק deque שישמור לכל קליעינט את ההודעות ששולח ב30 שניות האחרונות, אם שלח יותר מ5 הודעות, יקבל הודעה json מתאימה שמודיעה לו עבר את רף ההודעות שאפשר לשולוח ב30 שניות.
- **פתח את הסרבר כך שיתמוך במספר חדרים צ'אט, ותאפשר ליוורים לעبور בין חדרים :**  
כמובן שבניתי את האפשרות ליצור חדרים חדשים ובכך יש תמייה נוספת מחדר אחד, ברגע למעבר של יוורים, כמו שציניתי לעיל, ניתן להשתמש בפקודה exit/ובכך לחזור לחריפיט הראשי ולבחור חדר שונה.

### cut גובר לקליעינט

#### קליעינט :

- **תוודא שהסרבר תומך באוטנטיקציה מובטחת עם שם משתמש וסיסמה :**  
תווך כמו שהסבירתי לעיל.
- **בנה פיצ'ר שמאפשר ליוורים להחליף סיסמה באופן בטוח :**  
הפיצ'ר הזה קיים, ובכך יוצר מזון סיסמה חדשה (לאחר שבוחר 2 בתחריט הראשי), הסיסמה מוצפנת בעזרת hash והסיסמה החדשה בקובץ csv מוחלפת בסיסמא החדש.
- **בנה פיצ'ר שמאפשר ליוורים לבקש ולצפות בכל היוורים המוחברים :**  
בתחריט הראשי, אם לוחצים 3, מקבלים רשימה של כל היוורים שכעת מחוברים לחדר מסוים. זה נעשה בכך שהחדרים שמורים גם ב dictionary שמתעדכנים כל הזמן שחדרים נוצרים ונמחקים, ובכך מועבר המידע.
- **tabena פיצ'ר שמאפשר להעלות ולהוריד קבצים בביטחון :**  
כאשר אתה בחדר, אתה תוכל להקליד את הפקודת upload - היוזר מתחילה העלה: התהיליך מתחילה מצד היוזר כאשר המשתמש upload\_file בוחר אפשרות להעלות קובי. פעללה זו מתואמת על ידי הפונקציה upload\_file (path) על functions.py של היוזר. המשמש צריך להזין את הנתיב (path) על המחשב שלו של הקובי שהוא מעוניין להעלות.

היווזר שולח בקשה ה

עליה
: לאחר מכן, היווזר בונה הודעה הכוללת את סוג הפעולה ("upload\_file"), שם הקובץ והגודל שלו. מידע זה נשלח לשרת קובץ JSON. חלק זה של התהליך מנוהל על ידי הפונקציה `upload_file`, אשר משתמש ב-`messages.py` כדי לקובד את תוכן הקובץ ושלוח את הנתונים.

לאחר מכן השירות מתכוון לקבל את הקובץ: בתחום הקובץ `functions.py`, יש פונקציה מתאימה לטיפול בבקשת ה

עליה
. ברגע שהשרת מקבל את בקשה ה

עליה
, הוא מאשר אותה ומתכוון לקבל את נתוני הקובץ. השירות יודע לצפות לקובץ בגודל מסוים ולשמור אותו תחת השם שצוין, בזכות המידע שסופק בבקשת היווזר.

העברת הקובץ: היווזר מתחילה לשלוח את הקובץ בתים. זה מנוהל על ידי קריאה של הקובץ בחלקים קטנים (1024 בתים בכל פעם) ושליחת כל חלק דרך החיבור לSocket לשרת. לצד השירות, הנתונים נוכנסים אל השירות אל קובץ `files`, ומתקבל ונכתב לקובץ חדש בקובץ המצוין.

לבסוף, שהקובץ כולם הועבר בהצלחה, השירות שולח הודעה ליוזר שמאשרת את ה

עליה
 המוצלחת.

מעבר להזרה התהליך דומה: התהליך מתחילה כשהיווזר מבקש רשימת קבצים זמינים להורדה בעזרת הפקודה `download` /שאותה הוא כותב. לאחר מכן מוצגת לו רשימה של קבצים שהועלו לשרת, כולל זמינים להורדה. זה נעשה על ידי פונקציה ב-`functions.py` של היווזר, ששולחת בקשה לשרת המציגת את הרצון להוריד קובץ או לרשום קבצים. \*בקטע זה בקוד יש לי באג טניshi זמן ממושך לתיקן אך אין עובד באופן עקבי. אם יוזר מקליד `download` /ויזרים אחרים מדברים בחדר בזמן שהוא בוחר איזה קובץ להוריד, אז ההורדה מצילה אבל זה מוציא מהשרת את היווזר שהוריד. על שאר היווזרים אין שום השפעה.

לאחר קבלת בקשה לרשום קבצים, השירות, באמצעות ה-`functions.py` שלו, מאחזור את שמות כל הקבצים הזמינים בתקיינות חדר השיחה המצוין ושולח הודעה את הרשימה זו ליוזר. אם נבקש קובץ מסוים, השירות מתכוון לשלוח את הקובץ המבוקש על ידי פתיחתו וקריאת תוכנו בהכנה לשידור. השירות גם שולח עוד DATAה על הקובץ (למשל, גודל הקובץ) ליוזר כדי להקל על תהליך ההורדה.

যوزר شولח בקשה הורדת קובץ: אם התהליך התחיל ברישום קבצים, היווזר ישלח אז בקשה להוריד קובץ מסוים מהרשימה. הבקשה כוללת את שם הקובץ המבוקש.

אחרי קבלת בקשה הורדת הקובץ המסויימת, השרת מתחילה לשולח את הקובץ, בדומה לתחילה הعلاה אך בכיוון הפוך כמו שציינתי.. המחלקה FileTransfer ב- file\_transfer.py של השרת מנהלת את קריאת נתוני הקובץ ושליחתם ליוזר.

מצד היוזר, נתוני הקובץ הוכנסו מתקובלים ונכתבים לקובץ שלו. זה מנוהל על ידי המחלקה FileTransfer ב- file\_transfer.py של היוזר הפעם, שմבטייה שהנתונים מורכבים נכון לתסריט המקורי ונשמרים במקומם המקורי.

- לאחר שהקובץ כולם הועבר בהצלחה ונשמר, נשלחת הודעה בהתאם.
- תכנס אל הפרויקט הצפנה מקצת בין הודעה :
  - עקב בעיות זמן וקשיים טכניים, לא ביצעתו זאת. אציין שבר אמר בשעת קבלת שחוסר אימפלמנטציה של חלק זה תגרור הורדת נקודות מינורית.

סך הכל על דרישות הפרויקט. עבורו להסביר על הקבצים שיש לי ועל פונקציות מרכזיות ומטרתן:

**שרות:**

**קובץ - consts.py**

מגדיר קבועים כמו FORMAT, HOST, PORT שהשתמשו בהם ברחבי הפרויקט בשביל החיבור בין השרת ללקוחות, החיבור לsocket ולפורמט קידוד ההודעות.

**קובץ - auth.py**

: ממיר סיסמה פשוטה לגרסה מוצפנת לאחסון בטוח.  
hash\_password(password) : מוסיף משתמש חדש עם סיסמה מוצפנת register\_user(username, password, role)

וותפקיד לקובץ הנתונים של המשתמשים.

: בודק אם שם משתמש כבר קיים בקובץ הנתונים.  
user\_exists(username) : מעודכן את סיסמת המשתמש לסיסמא change\_password(username, password) מוצפנת חדשה.

**קובץ – chat\_room.py**

מחלקה ChatRoom: מנהלת פונקציונליות של חדר צ'אט כמו הוספה/הסרת לקוחות, שידור הודעות, ותיעוד. בעצם מגדרה את האובייקט של החדר ואת כל תכונותיו.

: מוסיף לקוח חדש לחדר הצ'אט.  
add\_client(client)

: מסיר לקוח מהדר הצ'אט.  
remove\_client(client)

: שולח הודעה לכל היוצרים בחדר.  
broadcast(message, sender)

## **קובץ - database\_controller.py**

מחלקה DatabaseController: מתחילה ומנוהלת את קבצי נתוני השירות, משתמשים וקבוצות.

אם משאבים ותיקיות אלו אינן קיימות או נוצרים כאלו.

## **קובץ - file\_transfer.py**

מחלקה FileTransfer: מטפלת בהעלאה והורדה של קבצים בין קלינטים לשרת. ()upload\_file\_to\_client: שולח קובץ מהשרת לקלינט.

:download\_file\_from\_client(file\_size) מקבל קובץ מקלינט ושמור אותו על השירות.

## **קובץ - functions.py**

מכיל פונקציות שימושיות שונות לטיפול בבקשתות של קלינטים כמו רישום, התחברות, ייצור/מחיקת חדר צאט, וכו'.

:register(conn, request) מטפלת בבקשתות רישום משתמשים. :login(conn, request) מאמתת משתמשים שמנסים להתחבר.

:list\_logged\_users(conn) שולחת רשימה של משתמשים שהתחברו כעת לקלינט. :upload\_file(conn, room\_name), download\_file(conn, room\_name) פונקציות שמטפלות בבקשת העלה והורדת קבצים.

## **קובץ המיין – main.py**

מקים את השירות, מקשיב לחיבורים נכנים, ויוצר תהליכי (threads) לטיפול בכל חיבור של קלינט חדש.

## **קובץ - messages.py**

:send\_success(conn, data) שולח הודעה הצלחה עם נתונים אופציונליים לקלינט. :send\_failure(conn, error\_message) שולח הודעה כישלון עם הסבר השגיאה לקלינט.

## **קובץ - user\_client.py**

מחלקה UserClient: מייצגת קלינט המחבר לשרת עם תכונות כמו שם המשתמש והחיבור שלו.

בקلينט סיד:

קבצי הקלינט משלקים חלק מהפונקציונליות של הסרבר סיד אך מוקדים בטיפול בקלט מהמשתמש, הצגת התగובות מהשרת, וניהול פעולות מצד של הקלינט כמו התחלה העברות קבצים או שינוי סיסמה וכו'.

**הקבצים שלהם בקובץ – functions.py**  
המקבילים שלהם בקבצים של השירות אך מותאמים לפעולות בклиינט סיד.

### קובץ – functions.py

מכיל מגוון רחב של פונקציות ולוגיקות, לצד של הקלינט לאינטראקציה עם השירות, כמו שליחת בקשה התחברות, רישום, והודעות צ'אט, וניהול הערות/הורדות קבצים.

fonctions.py

```
join_chat_room(room_name)
leave_chat_room(room_name)
send_chat_message(message, room_name)
upload_file(file_path, room_name)
download_file(file_name, room_name)
list_chat_rooms()
list_room_users(room_name)
```

ניתן מיד להבין מה כל פונקציה עשו רק מלקראם את השם/חתימה שלה.

### קובץ הרצה – main.py

נקודות הכניסה של היוזר לתכנית ובעצם התחברות לשרת. את הקובץ זהה מריצים על מנת להתחבר.

הקובץ מתפל בחיבור התחלתי לשרת ובאינטראקטיה עם המשתמש דרך הטרמינל.  
אלו הקבצים והפונקציות המרכזיות בפרויקט שלי, עם פירוט למטרותיהם ועשיותן.

## חלק תיאורתי

בחלק השני של הפרויקט התבוננו לענות על מגוון שאלות תיאורטיות, להלן השאלות:

### 5.1 Security

1. How can you enhance the security of your application? Discuss potential vulnerabilities and propose solutions.
2. What are the risks associated with transmitting messages in plain text? How might you implement encryption to address this concern?

### 5.2 Scalability

1. How would you design your application to handle a large number of concurrent users? Discuss potential bottlenecks and scalability challenges.
2. What strategies could be employed to distribute the load and balance the server's resources effectively?

### 5.3 Reliability and Fault Tolerance

1. Describe how you would ensure the reliability of your application. What measures can be taken to handle server failures or unexpected crashes?
2. How might you implement message persistence to ensure that messages are not lost even if the server restarts?

### 5.4 User Authentication

1. Explain the importance of user authentication in your application. What methods could be used to authenticate users securely?
2. How would you handle user registration and password management to enhance the overall security of the system?

### 5.5 Concurrency and Synchronization

1. Discuss the challenges associated with handling multiple simultaneous connections in a server.
2. How might you implement thread safety and synchronization to prevent data corruption or race conditions?

### 5.6 Protocol Design

1. Explain the choice of communication protocol for your application (e.g., TCP, UDP). What factors influenced your decision?
2. Explain about the *socket* handshake – which protocol you used, what are the commands, are they *blocking* commands. How would you design the message format and protocol for communication between the client and server?

### 5.7 Message Ordering and Delivery

1. How would you address the issue of message ordering and delivery in your application? Discuss potential scenarios and solutions.
2. What mechanisms could be employed to ensure that messages are delivered in a timely and reliable manner?

### 5.8 Persistent Storage

1. Explain the importance of persistent storage in your server. How did you design and implement the storage on the server?
2. Discuss the trade-offs between different storage solutions in the context of your application.

לאחר שראינו את השאלות, להלן התשובות:

#### : Security 5.1

- שיפור האבטחה: ניתן לישם פרוטוקולי אבטחה נוספים כמו SSL/TLS לתקשורת מוצפנת. להשתמש באלגוריתמי hash יותר מתקדמים למשל כמו bcrypt לאחסון מידע מוצפן בטוח, או לעשות salt לhash ים כדי לשפר אבטחה. ניתן לבצע בדיקות אבטחה קבוצות.

- סיכון של הודעות בטקסט גלי: שידור הודעות בטקסט גלי גורם להן להיות חשופות לחטיפה (hijacking). כמו שרשמתי לעלה, בעקבות חוסר זמן ויכולת לא ביצעת הצפנה מסקה לenza להודעות, אז נקודות לשיפור אבטחה בפרויקט שלי מقلילות גם יישום של הצפנה מסקה לenza באמצעות פרוטוקולים כמו TLS לשידור ואלגוריתמים או כמו AES להצפנה הודעות כדי להבטיח סודיות ושלמות.

#### :Scalability 5.2

- טיפול במספר רב של משתמשים מקבילים: יש לישם מודול קלט/פלט שאינו חוסם או מוגבל בקצב קלטים/הציג פלטים או להשתמש בתכנות אסינכרוני כדי לטפל ביעילות במספר גדול של משתמשים פעילים באותה עת. ניתן גם להשתמש באיזוני עומס כדי להפחיץ תנועה בין מופעים שונים של השרת.

- אסטרטגיות לפיזור וחלוקת עומס: ניתן להזדקק מערכת בסיס נתונים מבוזרת ואסטרטגיות כמו Horizontal Scalability כדי לפזר את העומס בצורה חכמת ויעילה. הטמעת מטמון או זיכרון קטן וזמן במילוי, כמו שלמדנו בקורסים שונים, יכולות לאחסן נתונים שניגשים אליהם לעיתים קרובות ויכולת גם להפחית עומס על השרת.

#### :Reliability and fault tolerance – 5.3

- הבטיח ווידוא אמינות: להשתמש בשיטות כמו Redundancy ו Clustering כדי להבטיח זמינות גבוהה בכל עת. ניתן גם להזדקק מערכת גיבוי קבוצה (כמו שקיים בפרויקט) ומנגנוני נסיגה ואקספושיונים/החרגות לטיפול בתקלות בשרת או בקריסות בלתי צפויות.

- שמירה על עקבות הודעות: לישם מערכת בסיס נתונים עמידה וגדולה עם תמייה transactions כדי להבטיח שמירת הודעות בעקבות. להשתמש בjournaling או logging הן גם אפשרויות טובות כדי להימנע מאובדן נתונים בעת תקלות וקריסות.

#### :authentication – 5.4

- חסיבות האותנטיקציה: חיונית לאבטחה ושמירה על פרטיות, האותנטיקציה מאמתת את זהות המשתמש ואת אמינותו. דרכיהם שנייתן להשתמש בהן כדי לשימוש אוטנטיקציה באופן מאובטח כוללות בין השאר יישום אוטנטיקציה רב-גורמית (MFA) להוספה שכבת

אבטחה נוספת מוסיפה מעבר לשם משתמש וסיסמה, למשל הכנסת מספר טלפון ובכך תישלח אל הטלפון הודעה ווידוא כניסה.

- רישום וניהול סיסמות: להשתמש בשיטה בטוחה כמו hashing שמייחדי בפרויקט שלבי בעת שמירת סיסמות היוצרים. שימוש ב sha 256 למשל. שוב לבצע salting. דרישות מורכבות לסיסמה (מינימום תווים, אותיות גדולות וקטנות, מספרים, תווים וסימונים ועוד) ומנגנוני איפוס סיסמה מאובטחים (מייחדי אחד בפרויקט) להגנה על חשבונות המשתמשים.

## 5.5 Concurrency and Synchronization :

- טיפול במספר חיבורים באותו זמן: חלק מן האתגרים של ניהול מספר חיבורים במקביל על ידי שימוש Concurreny models יעילים כמו ארכיטקטורה מבוססת מאירועים Skorers או שימושFrameworks שתומכות בעיבוד אסינכרוני.

- בטיחות ת'רדים וסנכרון ת'רדים: ניתן לישם בטיחות ואמינות ת'רדים באמצעות מנעולים ווידוא קטיעים קרייטיים בעזרת mutexים וsemaphores למיניהם. ניתן גם למש מגנוני סנכרון אחרים כדי למנוע מצבים של תחרות ו race conditions ולהבטיח עקביות נתוניות.

## 5.6 עיצוב פרוטוקול – Protocol design

- בחירת פרוטוקול תקשורת: כموבחן שבחرت לחששות בפרוטוקול תקשורת TCP. בחירתו אותו בעקבות אמינותו והבטחת סדר וארגון והבטחה שהמידע שנשלח יוודא שהגיע, תנאים אלה הכרחיים לישום צ'אט אמין, עקיי ואופרטיבי. בנוסף השיקולים שהחנתי כוללים רגישות נתוניות, צורך בתקשורת בזמן אמת ואמינות הרשות.

- TCP Design : Socket Handshake & Comm. (ACK,ACK- SYN,SYN) מודאות חיבור בטוח שהקלינט והסרבר מודעים לו. בעזרת שימוש ב JSONים בקוד שלי מבוצעת תקשורת ברוב המקרים.

## 5.7 סדר ומסירת הודעות – message ordering and delivery

- טיפול בסדר ומסירת הודעות: ניתן לחששות במספרים סידוריים או timestamps או כל מגנון אחר לוודא יחס סדר תקני בין ההודעות. בכדי להבטיח שההודעות עובדות בסדר שהן אמורות לעבור. בנוסף, אפשר לישם מגנוני אישור כדי לאשר קבלת הודעות וטיפול בהן (מייחדי חלקית בפרויקט)

- הבטחת מסירת הודעות בזמן ובצורה אמינה: אפשר לבצע, מעבר למנגנונים הפנימיים של TCP, פרוטוקולים ברמה היישומית כדי לנצל תורי הودעות ולהבטיח מסירה גם במקרה של כשלים זמינים בראשת או בשרת.

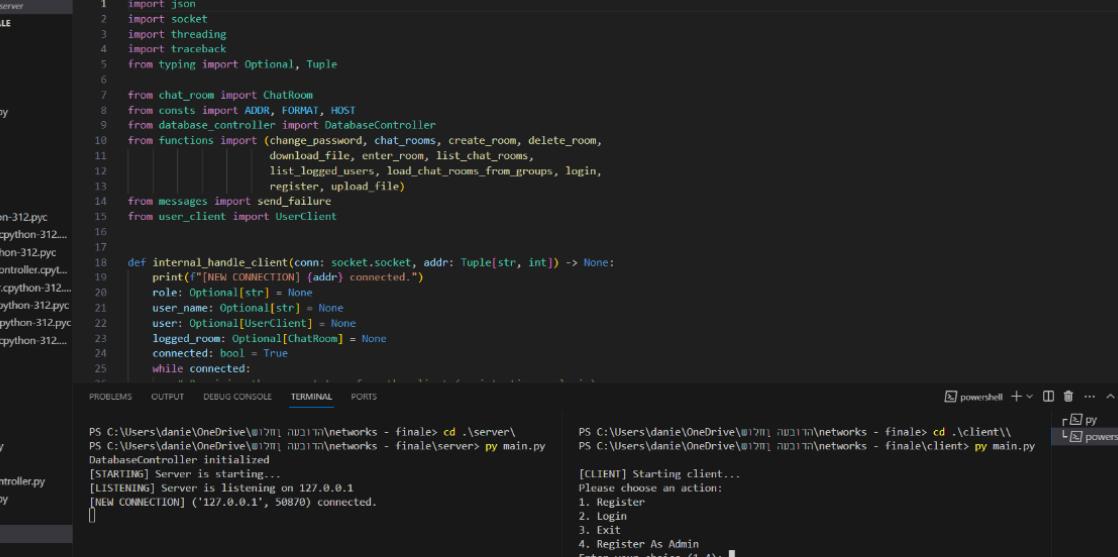
## 5.8 אחסון עקבי - : persistent storage

-חשיבות האחסון העקבי: נחוץ לשמירה על נתונים בין קרייסות, כיבויים והפעולות מחדש. תומך בתכונות כמו היסטוריות הודעות, הגדרות משתמש, ויום ני צ'אט (נשמרים בפורמטי .csv או ב log.csv. בהתאם אצל).

- פתרונות אחסון: הבחירה בין בסיסי נתונים יחסיים (SQL) לבין לא-יחסיים (NoSQL) תלויות במבנה הנתונים, דפוסי הגישה ודרישות ה scalability. צריך לשקול את הפתרונות בנוגע למורכבות, מהירות וscalability.

**מגון תМОנוות והרצות של הפרויקט:**

**הרצת הסרבר + תפריט הכנסה:**



The screenshot shows a terminal window with the following output:

```
PS C:\Users\danie\OneDrive\שולחן העבודה\networks - finale> cd ..\server\  
PS C:\Users\danie\OneDrive\שולחן העבודה\networks - finale> py main.py  
DatabaseController initialized  
[STARTING] Server is starting...  
[LISTENING] Server is listening on 127.0.0.1  
[NEW CONNECTION] ('127.0.0.1', 5080) connected.  
[]
```

Below the terminal, a message box displays:

[CLIENT] Starting client...  
Please choose an action:  
1. Register  
2. Login  
3. Exit  
4. Register As Admin  
Enter your choice (1-4):

ניסיונו התחברות כושל:

```
[NEW CONNECTION] ('127.0.0.1', 50870) connected.
Registering a new user. User name: ad Password: ad
-
Logging in user: ad
Logged in successfully, user role: admin
Creating a new chat room: newroom
[NEW CONNECTION] ('127.0.0.1', 50990) connected.
Logging in user: a
Registering a new user. User name: a Password: a.

Select an option: 1
Chat rooms:
newroom
Enter chat room name: newroom
Joined room: newroom
Replaying chat messages from room newroom

Replayed all messages
use /help to see available commands
> hi
> hello
> |
```

2. Login  
3. Exit  
4. Register As Admin

Enter your choice (1-4): 2

Enter username and password to login:

Enter username: a

Enter password: a

Login failed - Invalid username or password..

Please choose an action:

1. Register
2. Login
3. Exit

Col 1: group\_name Ln 3, Col 1 Spaces: 4 UTF-8 LF CSV

## הרשמה מוצלחת מתווספת לדאטאבייס:

The screenshot shows the VS Code interface with the terminal window active. The terminal output is as follows:

```
PS C:\Users\danie\OneDrive\שולחן העבודה\networks - finale> cd .\server\
PS C:\Users\danie\OneDrive\שולחן העבודה\networks - finale> py main.py
DatabaseController initialized
[STARTING] Server is starting...
[LISTENING] Server is listening on 127.0.0.1
[NEW CONNECTION] ('127.0.0.1', 50870) connected.
Registering a new user. User name: ad Password: ad
Enter your choice (1-4): 4
Registering a new user...
Please enter a username and password to register.
Username: ad
Password: ad
User ad registered successfully.
Please choose an action:
1. Register
2. Login
3. Exit
4. Register As Admin
Enter your choice (1-4):
```

## התחברות מוצלחת

The screenshot shows the VS Code interface with the terminal window active. The terminal output is as follows:

```
PS C:\Users\danie\OneDrive\שולחן העבודה\networks - finale> cd .\server\
PS C:\Users\danie\OneDrive\שולחן העבודה\networks - finale> py main.py
DatabaseController initialized
[STARTING] Server is starting...
[LISTENING] Server is listening on 127.0.0.1
[NEW CONNECTION] ('127.0.0.1', 50870) connected.
Registering a new user. User name: ad Password: ad
(CLIENT) Starting client...
Please choose an action:
1. Register
2. Login
3. Exit
4. Register As Admin
Enter your choice (1-4): 4
Registering a new user...
Please enter a username and password to register.
Username: ad
Password: ad
User ad registered successfully.
```

## התפריט הראשי

```
PS C:\Users\danie\OneDrive\שולחן העבודה\networks - finale> cd ..\server\  
PS C:\Users\danie\OneDrive\שולחן העבודה\networks - finale\server> py main.py  
DatabaseController initialized  
[STARTING] Server is starting...  
[LISTENING] Server is listening on 127.0.0.1  
[NEW CONNECTION] ('127.0.0.1', 50870) connected.  
Registering a new user. User name: ad Password: ad  
Logging in user: ad  
Logged in successfully, user role: admin  
Enter username: ad  
Enter password: ad  
Logged-in as admin successfully.  
1. Enter chat room  
2. Change password  
3. List logged in users  
4. Create a new chat room  
5. Delete a chat room  
/exit to exit  
Select an option: 1
```

לחיצה על 1 כשאין חדרים קיימים:

```
PS C:\Users\danie\OneDrive\שולחן העבודה\networks - finale> cd ..\server\  
PS C:\Users\danie\OneDrive\שולחן העבודה\networks - finale\server> py main.py  
DatabaseController initialized  
[STARTING] Server is starting...  
[LISTENING] Server is listening on 127.0.0.1  
[NEW CONNECTION] ('127.0.0.1', 50870) connected.  
Registering a new user. User name: ad Password: ad  
Logging in user: ad  
Logged in successfully, user role: admin  
/exit to exit  
Select an option: 1  
There are no available rooms
```

## שינוי סיסמה מוצלח (בוחרים 2 בתפריט הראשי ואו מקלידים את הסיסמה), מעודכו בדאטאביס :

The screenshot shows a terminal window with the following text output:

```

PS C:\Users\danie\OneDrive\שולחן העבודה\networks - finale> cd ..\server\
PS C:\Users\danie\OneDrive\שולחן העבודה\networks - finale> py main.py
DatabaseController initialized
[STARTING] Server is starting...
[LISTENING] Server is listening on 127.0.0.1
[NEW CONNECTION] ('127.0.0.1', 50870) connected.
Registering a new user. User name: ad Password: ad
Logging in user: ad
Logged in successfully, user role: admin
2. Change password
3. list logged in users
4. Create a new chat room
5. Delete a chat room
/exit to exit
Select an option: 2
Enter new password: a
Password changed successfully
1. Enter chat room
2. Change password

```

Col 1: username Ln 1, Col 1 Spaces: 4 UTF-8 CRLF CSV

**יצירת חדר חדש על ידי לחייה 4 בתפריט הראשי, החדר נוספף לקובץ:**

The screenshot shows a terminal window with the following text output:

```

PS C:\Users\danie\OneDrive\שולחן העבודה\networks - finale> cd ..\server\
PS C:\Users\danie\OneDrive\שולחן העבודה\networks - finale> py main.py
DatabaseController initialized
[STARTING] Server is starting...
[LISTENING] Server is listening on 127.0.0.1
[NEW CONNECTION] ('127.0.0.1', 50870) connected.
Registering a new user. User name: ad Password: ad
Logging in user: ad
Logged in successfully, user role: admin
Creating a new chat room: newroom
1. Enter chat room
2. Change password
3. List logged in users
4. Create a new chat room
5. Delete a chat room
/exit to exit
Select an option: 4
Enter new chat room name: newroom
Room newroom created successfully

```

Col 1: group\_name Ln 1, Col 1 Spaces: 4 UTF-8 LF CSV

החדר נוצר, להיזה שוב על 1 והחדר מופיע:

The screenshot shows a Windows desktop environment with the Visual Studio Code (VS Code) interface open. The terminal window displays the output of a Python application. The application starts by initializing a DatabaseController and listening on port 50870. It then registers a new user ('ad') with password 'ad'. Finally, it prompts the user to enter a chat room, listing 'newroom' as an option.

```
PS C:\Users\danie\OneDrive\שולחן העבודה\networks - finale> cd ..\server\  
PS C:\Users\danie\OneDrive\שולחן העבודה\networks - finale\server> py main.py  
DatabaseController initialized  
[STARTING] Server is starting...  
[LISTENING] Server is listening on 127.0.0.1, 50870  
[NEW CONNECTION] ('127.0.0.1', 50870) connected.  
Registering a new user. User name: ad Password: ad.  
  
Logging in user: ad  
Logged in successfully, user role: admin  
Creating a new chat room: newroom  
[]  
  
1. Enter chat room  
2. Change password  
3. List logged in users  
4. Create a new chat room  
5. Delete a chat room  
/exit to exit  
Select an option: 1  
Chat rooms:  
newroom  
Enter chat room name: 
```

## הכניסה לחדר ריק נראית כך:

The screenshot shows a terminal window with the following text output:

```
PS C:\Users\danie\OneDrive\שולחן העבודה\networks - finale> cd .\server\
PS C:\Users\danie\OneDrive\שולחן העבודה\networks - finale\server> py main.py
DatabaseController initialized
[STARTING] Server is starting...
[LISTENING] Server is listening on 127.0.0.1
[NEW CONNECTION] ('127.0.0.1', 50870) connected.
Registering a new user. User name: ad Password: ad.

Logging in user: ad
Logged in successfully, user role: admin
Creating a new chat room: newroom

Replaying all messages
use /help to see available commands
> 
```

הכנסת קלינט שני, ובדיקה יוזרים מחוברים לחדרים. Ad הינו הטרמינל האמצעי שנמצא בתוך החדר.

The screenshot shows a terminal window with the following text output:

```
PS C:\Users\danie\OneDrive\שולחן העבודה\networks - finale> cd .\server\
PS C:\Users\danie\OneDrive\שולחן העבודה\networks - finale\server> py main.py
DatabaseController initialized
[STARTING] Server is starting...
[LISTENING] Server is listening on 127.0.0.1
[NEW CONNECTION] ('127.0.0.1', 50990) connected.
Logging in user: a
Logged in successfully, user role: user

Replaying all messages
use /help to see available commands
> hi
> hello
> 
```

A red box highlights the user "ad" in the replays section of the terminal window.

היווזר השני נכנס לחדר, הודיעו ששלחו נתונים גם לו (הרץ הנו שמורות בלוג, גם אם הסרבר נופל וكم הלוגים יישארו):

```

Select an option: 1
Chat rooms:
newroom
Enter chat room name: newroom
Joined room: newroom
Replaying chat messages from room newroom
Replayed all messages
use /help to see available commands
> hi
> hello
> []
> []

Select an option: 1
Chat rooms:
newroom
Enter chat room name: newroom
Joined room: newroom
Replaying chat messages from room newroom
ad:hi
ad:hello
Replayed all messages
use /help to see available commands
> []

```

הודעות נשלחות בין יוזרים בליב, ללא שום דילוי או תנאי מיוחד:

```

Chat rooms:
newroom
Enter chat room name: newroom
Joined room: newroom
Replaying chat messages from room newroom
Replayed all messages
use /help to see available commands
> hi
> hello
< a: hello
> []

Chat rooms:
newroom
Enter chat room name: newroom
Joined room: newroom
Replaying chat messages from room newroom
ad:hi
ad:hello
Replayed all messages
use /help to see available commands
> hello
> []

```

## ביצוע העלאה מוצלחת:

The screenshot shows a terminal window with the following text output:

```
Replayed all messages
use /help to see available commands
> hii
> hello
< a: hello
> /upload
Enter the path to upload: C:\Users\danie\OneDrive\/networks - finale\live messages.png
File sent over
File uploaded successfully
```

A red box highlights the command `/upload` and its output.

## ביצוע הורדה מוצלחת:

The screenshot shows a terminal window with the following text output:

```
Chat rooms:
newroom
Enter chat room name: newroom
Joined room: newroom
Replaying chat messages from room newroom
ad:hii
ad:hello
Replayed all messages
use /help to see available commands
> hello
> [/]

Chat rooms:
newroom
Enter chat room name: newroom
Joined room: newroom
Replaying chat messages from room newroom
ad:hii
ad:hello
Replayed all messages
use /help to see available commands
> hello
> [/]
```

A red box highlights the command `/download` and its output.

## הקובץ שהועלה לשרת אכן הגיע:

The screenshot shows a terminal window with the following logs:

```

Replayed all messages
use /help to see available commands
> hi
> hello
< a: hello
> /upload
Enter the path to upload: C:\Users\danie\OneDrive\שולחן העבודה\Networks - finale\live messages.png
File sent over
File uploaded successfully
> []

Replayed all messages
use /help to see available commands
> hello
> []

```

Chat rooms:

```

newroom
Enter chat room name: newroom
Joined room: newroom
Replaying chat messages from room newroom
ad:hi
ad:hello

```

Replayed all messages

```

use /help to see available commands
> hello
> []

```

Col 1: group\_name Ln 3, Col 1 Spaces: 4 UTF-8 LF CSV

יציאה מהדר בעזרת פקודה `/exit`

The screenshot shows a terminal window with the following logs:

```

Chat rooms:
newroom
Enter chat room name: newroom
Joined room: newroom
Replaying chat messages from room newroom
ad:hi
ad:hello

file_name: 'live messages.png'
File downloaded successfully
> hello
> /exit
stopped reading incoming messages from channel

1. Enter chat room
2. Change password
3. List logged in users
/exit to exit
Select an option: []

```

Col 1: group\_name Ln 3, Col 1 Spaces: 4 UTF-8 LF CSV

## כל ההודעות של החדרים נשמרים בלוגים משליהם:

The screenshot shows a terminal window with two tabs open: "chat\_room\_newroom.log" and "chat\_room\_newroom.log". The left pane displays the file structure of the application, including "groups.csv", "server\database\chat\_room\_newroom.log", and various Python files like "auth.py", "chat\_room.py", etc. The right pane shows the logs from the "chat\_room\_newroom.log" file.

```

chat_room_newroom.log
server > logs > chat_room_newroom.log
1 ad:hi
2 ad:hello
3 a:hello
4 a:.download
5 a:hello

```

The logs show interactions between users "ad" and "a". User "ad" sends "hi", "hello", ".download", and "hello". User "a" responds with "hello" and ".download".

פה כבר הוספה כמה חדרים, ניתן לראות שcols מופיעים ב database וגם מופיע בlijיב  
ברשימה החדרים הזמינים:

The screenshot shows a terminal window with two tabs open: "groups.csv" and "chat\_room\_newroom.log". The left pane displays the file structure, including "groups.csv" and "server\database\chat\_room\_newroom.log". The right pane shows the logs from the "chat\_room\_newroom.log" file.

```

[NEW CONNECTION] ('127.0.0.1', 50990) connected.
Logging in user: a
Registering a new user. User name: a Password: a.

Select an option: 4
Enter new chat room name: room3
Room room3 created successfully

1. Enter chat room
2. Change password
3. List logged in users
4. Create a new chat room
5. Delete a chat room
/exit to exit
Select an option: 1
Chat rooms:
new
room1
room3
Enter chat room name: 

```

The logs show a new user "a" connecting and creating a new chat room named "room3". The user then selects option 1 to see the list of available chat rooms, which includes "new", "room1", and "room3".

יוזר צמובן יכול להמשיך לדבר כרגע לאחר הורדת/העלאת הקובץ:

The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows the project structure under "OPEN EDITORS".
- Terminal:** Displays the command-line interaction between the user and the application. The user logs in as "ad", creates a new chat room "newroom", and uploads a file named "live messages.png". The application responds with file details and download links.
- Output:** Shows the file being uploaded over the network.
- Code Editor:** Shows the code for the "group.py" file, which handles file upload logic.
- Bottom Status Bar:** Shows the file size (198.33KB) and the date (17/03/2024).

מציאות הדר, כל הנתונים הקשורים אל ההדר נמחקים:

The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows the project structure under "OPEN EDITORS".
- Terminal:** Displays the command-line interaction where the user deletes a chat room named "newroom". The application responds with confirmation messages.
- Output:** Shows the file being deleted.
- Code Editor:** Shows the code for the "group.py" file, which handles room deletion logic.
- Bottom Status Bar:** Shows the file size (198.33KB) and the date (17/03/2024).

אלו רק חלק מהפתרונות בקורס וחלק מהפתרונות הקיימים. יש עוד כל מיני הודעות, ווידואים ופונקציונליות שלא הראתי בצילומי המסך אך הן מכוסות חלק ניכר מן הפרויקט.