

## סיכום כללי:

אנחנו בונים מערכת של ניהול טיסות מכירת כרטיסים לנוסעים. המערכת יש שתי פנים - הפן של מנהל חברת התעופה והפן של הלוקו מנהל חברת התעופה יכול ליצור קווי תעופה חדשים - בתנאי שיש מספיק תקציב. נושא יכול לקנות כרטיס לקו מסוים, בתנאי שיש קו צזה.

התרגיל מבוסס על **persistence** - שימרת מידע מחוץ לאפליקציית פיתון כך שהוא מתקיים גם אחרי שהאפליקציה מפסיקת מסיים את ריצתה.

שימרת המידע תיעשה על 3 סוגי קבצים שונים:

1. קובץ טקסט רגיל
2. קובץ CSV
3. קובץ JSON

הפעולות שאתם נדרשים לתמוך בהן:

כמפורט:

1. הזרדות כמנהל על ידי סיפוק שם משתמש וסיסמה
2. קניית כניסה לנמל תעופה חדש (אם התקציב מאפשר את זה)
3. יצירת נתיב תעופה חדש (אפשר רק אם נקיתה כניסה לנקודת המוצא ולנקודת הקצה)

כמפורט:

1. קניית כרטיס טיסה לפי נקודת מוצא ונקודת קצה

חלק מהקבצים יספקו לכם, ואתם צריכים רק להכניס אותם לפרוייקט ול證וד את קיומם.

הנחיות כלליות:

1. הפרוייקט יוכל מוגש בREPO בגיטהאב
2. העבודה תיעשה עם ענף לכל שלב. יש לפתח ענפים שיוצאים מהענף של אותו שלב עבור המשימות הקטנות ועובדות MERGE עם PR לטור MAIN ככל העבודה על אותו שלב הסתיימה.
3. הענף של השלב עבור MERGE עם PR לטור MAIN ככל העבודה על אותו שלב הסתיימה.
4. יש לשמר על קוד נקי ומסודר
5. אין לMargin' לטור MAIN קוד שנמצא בתוך העורות, רווח' שורות מיותרים, וPRINT מיותר.

בכל שלב תתוארנו המשימות שאוטו שלב דרוש. אתם צריכים לפרק את המשימות, להבין מה לבדוק נדרש, וליצור ענף חדש לכל תת משימה (ולMargin' אותו חזרה לטור הענף של אותו שלב כשםמשימה הסת"ימה). לעיתים תתוארנו חלק מהfonkcioot שאטם נדרשים למש - אלו לא כל הפונkcioot הנדרשות. אתם נדרשים לקרוא היטב את הדרישות בכל פיצ'ר ולממש כל פונkcיה שצריך כדי שככל תוכנה וכל פעולה תתקיים.

## שלב א - תשתיות בסיסיות

### פיצ'ר יצירת קובץ תקציב ובדיקה קיומם הקובץ בתחילת הריצה:

1. יצירת קובץ טקסט שישמר את התקציב של חברת התעופה. אין צורך ליצור את הקובץ ולאכלס אותו במספר דרך פיתון - אפשר לעשות זאת דרך הטרמינל/AI. שם הקובץ: budget.txt.
2. כשמתחילה את האפליקציה יש לוודא בעזרת מודול OS שקיים קובץ התקציב. אם לא - יש לזרוק שגיאה ולעוזר את ריצת התוכנה.

חלק מהfonקציות ספציפיות לפיצ'ר זהה:

1. פונקציה שתוענת את קובץ budget ומחזירה את המספר ששמור בקובץ במספר
2. פונקציה שבודקת את קיומו של קובץ בשם budget.txt - מחזירה True או False בהתאם לתוצאה.

### פיצ'ר הזרחות כמנהל

1. יש ליצור קובץ בשם credentials.csv. יצירת הקובץ תיעשה דרך הAI/טרמינל, אין צורך ליצור ולאכלס את הקובץ דרך פיתון.
2. הקובץ הוא CSV שمدמה טבלה של שמות משתמש וסיסמות. השורה הראשונה היא: username,password
3. הכניסו ידנית בשורה השנייה Username של שם משתמש וסיסמה. הערכים אלו ימשכו להזרחות כמנהל לטבות פעולות המנהל. (אקסטרה - אם אין ערכים בהרצה ראשונה, בקשו מהמנהל להכניס שם משתמש וסיסמה, ושימרו את הסיסמה כHASH)
4. כשמתחילה את האפליקציה יש לוודא בעזרת מודול OS שקיים קובץ CREDENTIALS.csv. אם לא - יש לזרוק שגיאה ולעוזר את ריצת התוכנה.

חלק מהfonקציות נדרשות לפיצ'ר זהה:

1. פונקציה שבודקת את קיומו של קובץ בשם credentials.csv - מחזירה True או False בהתאם לתוצאה.
2. פונקציה שתוענת את הקובץ, ועובדת עליו שורה שורה. אפשר להשתמש במודול CSV או בכל כל אחר. הפונקציה מחזירה את תוכן הקובץ כמטריצה - כל שורה היא רשימה נפרדת.
3. פונקציה שמקבלת כפרמטר שם משתמש וסיסמה, ואת המטריצה של תוכן הקובץ CREDENTIALS.csv. הפונקציה עוברת על תוכן הקובץ ומחפשת את השם משתמש והסיסמה הנתונים. מחזירה TRUE או FALSE בהתאם לתוצאה.

## **שלב ב** - הרצה בסיסית, תפריט בסיסי, הזרחות

פיצ'ר תשתיות:

1. יש ליצורקובץ `uk.main`. הקובץ זה יהיה הקובץ היחיד שמריצים.
2. הפעולה הראשונה שתבוצע היא בדיקת קיומם של הקבצים `BUDGET` ו-`CREDENTIALS`. יש להשתמש בפונקציות שמומשו בשלב הקודם.
3. הקובץ יציג תפריט בסיסי. השאלה הראשונה של התפריט היא האם המשתמש הוא מנהל או נסען.
4. אם הוא מנהל - התפריט יבקש מהמשתמש שם משתמש וסיסמה.
5. יש לבדוק את תקינות שם המשתמש והסיסמה תוך שימוש בפונקציה שהוגדרה בשלב הקודם.

## **שלב ג - יצירת קווים תעופה חדשים כמנהל**

### **פיצ'ר טעינת בנק שדות תעופה:**

1. נתון לכם קובץ בשם csv.airports. בקובץ זה מפורטים בשורת CSV כל שדות התעופה האזמינים, יחד עם המחיר שיש לשם כדי להשתמש בשדה התעופה זהה חלק מקו טיסה. בנוסף, כתוב שם שדה התעופה כקוד בן 3 אותיות, והיבשת שבה שדה התעופה נמצא (רלוונטי לחישוב עלות טיסה, ראו בהמשך). עברו על הקובץ וודאו שאתם מבינים את המבנה שלו.
2. יש לוודא את קיומו של הקובץ של שדות התעופה בדמות לקבצי התקציב והסימאות - באותו שלב של האפליקציה, ולזרוק שגיאה באופן דומה אם הקובץ לא נמצא.

### **פיצ'ר יצירת קו חדש:**

1. יש להוציא קובץ טקסט חדש שישמר את הקווים הקיימים של חברת התעופה שלנו

a. שם הקובץ: json.available\_flights

b. יש לוודא את קיומו באותו שלב ובאותו אופן כמו שאר קבצי הטקסט.

c. מבנה הNJSON:

d. {

```
    "available_lines": [
        {
            "origin_airport": <airport code>,
            "destination_airport": <airport code>
        },
        {
            ...
        }
    ]
}
```

2. אחרי הזרה מוצלחת כמנהל, יש להציג בתפריט את האפשרות להוציא קו תעופה חדש.  
a. אם משתמש בוחר באפשרות זו, יש לבקש מהמשתמש לתת 3 אותיות שהן הקוד של שדה התעופה - אחד עבור נקודת המוצא של הקו ואחד עבור נקודת הקצה של הקו.

b. יש לטוען את קובץ שדות התעופה ולבודד שני צמיד 3 האותיות קיימים במאגר. אם אחד מהם לא קיים במאגר - יש לבקש קו אחר מהמשתמש במקום זה לאינו קיים.

3. אחרי שיש לנו 2 קודים של שדות תעופה קיימים, התוכנה צריכה לוודא שיש למנהל מספיק כסף בתקציב כדי כניסה לשני שדות התעופה האלה.

a. יש לטוען את קובץ התקציב כמספר. יש להשתמש בפונקציה משלב א'.

b. יש לטוען את סכום של דמי הכניסה של שני שדות התעופה מתוך קובץ שדות התעופה.

c. יש להשווות בין התקציב לבין הסכום - אם הסכום גדול מהתקציב, יש לכתוב הודעה שמשמעותה את הבעיה, לזרוק שגיאה ולחזור לתפריט המנהל.

d. אם התקציב גדול מהסכום - יש לבקש אימות סופי ל垦נית קו התעופה.

e. אחרי קבלת אימות סופי:

.i. יש לטוען את קובץ json.available\_lines

.ii. יש להוסיף את הקו החדש לתוך רשימת הקווים

.iii. יש לשמר את הקובץ מחדש - הפעם עם הקו החדש בתוכו.

## שלב ד - קניית כרטיס טיסה כנוע.

### פיצ'ר הצגת תפריט בסיסי כנוע

1. יש לעדכן את התפריט הראשי בבחירה הראשונה - ולאפשר בחירה כנוע, ולא רק כמנה.
2. אם המשתמש בוחר להמשיך כנוע, על התפריט להציג לו את כל הטיסות הזרומות.
  - a. יש לטען את קובץ json\_lines\_available\_lines
  - b. יש להציג بصورة מסודרת את כל הקווים - קו מוצא וקו יעד.
  - c. המשתמש צריך ללחוץ טישה.
  - d. ברגע שהוא בוחר בחירה תקינה, מתבצע חישוב מחיר הכרטיס.

### פיצ'ר חישוב מחיר הכרטיס

1. חברת תעופה חשובה לנו להישאר רוחחים. לכן נדרש חישוב מדויק של מחיר הכרטיס.
2. חישוב מחיר הכרטיס מחושב באופן הבא:
  - a. יש לקחת את מחיר הנסיעה לשדות התעופה של שדה המוצא ושדה היעד.
  - b. יש לחלק את הסכום ל400 - כמה הנוסעים בטיסה ממוצעת. זהו **מחיר הבסיס**.
  - c. יש להוסיף לחשבון את אורך הטיסה. לצורך כך, ניתן למסור נתון הקובץ continents\_pricing.csv
  - i. יש לוודא את קיום הקובץ זהה יחד עם שאר קבצי התקסט.
  - ii. יש להתenga אם קובץ זה חסר بصورة זהה להתנהגות של חסרים עם שאר קבצי התקסט.
  - d. יש לזיהות את היבשת של שדה המוצא:
  - i. יש לטען את קובץ airports.csv.
  - ii. יש לחפש לפי קוד שדה תעופה
  - iii. יש לשולף מהשורה את היבשת
  - e. לחזור על השלב הקודם עבור שדה היעד.
  - f. יש לטען את קובץ continents\_pricing.csv ולבדוק מה המחיר לפי שתי היבשות לפי יישוב מוצא –> יישוב יעדי
  - g. יש לקחת את הסכום הזה, לחלק אותו ל400 ולהוסיף אותו למחיר הבסיס. זהו **מחיר הכרטיס**.
3. יש להציג למשתמש את המחיר של הכרטיס ולבקש אימות סופי שהוא מעוניין לקנות את הכרטיס.
4. אם המשתמש מאשר:
  - a. יש להוסיף לקובץ התקציב של החברה את הסכום שהמשתמש שילם על הכרטיס
  - b. יש ליצור ID אקריא בין 8 אותיות המורכב מאותיות ומספרים - זהו ID הכרטיס.
  - c. אחרי העדכון של התקציב יש להודיע למשתמש כי הכרטיס נקנה בהצלחה, יש להציג את הID ולהזור לתפריט המשתמש הראשי.

בונוס:

1. הגבלת כמה כרטיסים פרטיסה - ברגע שנקנה כרטיס יש כרטיס אחד פחות
2. אפשרויות ביטול של כרטיס (עדכון התקציב בהתאם) עד 10 ימים מרגע הקניה.
3. אפשרויות של עיריכת הכרטיס.

## **שלב ב - תוספות**

### **פתרונות נוספים:**

1. יצירת "שות של טיסות כמנהל חברת התעופה. כל טיסה:
  - a. קיימת על קו קויים של שדה יעד ושדה מוצא
  - b. בעלת ID ייחודי
  - c. בעלת מספר נסיעים מסוים, שרק עד אליו ניתן לקבלות כרטיסים.
  - d. מחולקת למחולקות משנהות את מחיר הcarteis - אקונומי, עסקים, ראשונה.
2. לוגים:
  - a. כדי להבין מה קורה באפליקציה, יש ליצור לוגים מסודרים לפי הפורמט:  
`date | log level | message`
  - b. יש להכניס את הלוגים האלה לפני כל פעולה מורכבת אחרת