

## READ ME- Student Equipment Loan Management System

מערכת זו נועדה לניהול השאלות והחזרות של ציוד לסטודנטים במעבדה.  
היא כוללת ממשק משתמש ב-Tkinter, ושומרת נתונים במסדי נתונים מסוג SQLite.

### מבנה בסיסי

המערכת משתמשת בשני מסדי נתונים נפרדים:

- students.db: מידע על סטודנטים והשאלות.
- equipment.db: פרטי הציוד שנמצא במלאי.

באמצעות הפונקציה create\_db() — נוצרות הטבלאות במסדי הנתונים אם אינן קיימות.

ישנם שתי מסכים שונים שמציגים טבלאות :

1. ViewStudents שם ניתן לראות את כול הסטודנטים שלא החזירות את הציוד של המעבדה.

הטבלה מוצגת באופן הבא:

עמודה	תיאור
DateLoan	תאריך ההשאלה
DateAssumedReturn	תאריך ההחזרה הצפוי
StudentName	שם הסטודנט
StudentID	תעודת זהות
PhoneNum	מספר טלפון
Email	דוא"ל
ProjectTutor	מנחה פרויקט
GipNum	מספר GIP של הציוד
SerialNum	מספר סידורי של הציוד

הצגת הטבלה מתבצע באמצעות הפונקציה - view\_students() אשר מציג בטבלת Treeview את כלל הסטודנטים שהשאלו ציוד.

פונקציות:

- אפשרת להצגת הסטודנטים שעבר הזמן בו היו אמורים להחזיר אך לא החזירו- שימו לב להקפיד על הפורמט הבא בתאריכים על מנת לאפשר לפונקציה לפעול כדרוש: '%Y-%m-%d'
- : מופעל על ידי קראיה לפונקציה: def filter\_overdue\_students()
  - ביטול הפעולה על ידי לחיצה על פתור RESET – מופעל על ידי קראיה לפונקציה: def reset\_view():

2. ViewDatabase שם ניתן לראות את כול הציווד של המעבדה.

הטבלה מוצגת באופן הבא:

עמודה	תיאור
ID	מזהה
GipNum	מספר GIP
Type	סוג הציווד
Model	דגם
DateAdded	תאריך הוספה למערכת
SerialNum	מספר סידורי
Hdetails	פרטי חומרה
Edetails	פרטי תוכנה
DateUpdated	תאריך עדכון אחרון
State	מצב הציווד
Owner	בעלים
DateAssumedReturn	תאריך החזרה צפוי (אם בהשאלה)
Tnum	מזהה פנימי נוסף
Location	מיקום במעבדה
Notes	הערות כלליות
WikiLink	קישור לוויקי או מדריך
Returned	האם הוחזר (YES/NO)

במסך מתאפשרות הפונקציונליות הבאות:

1. צפייה ברשימת הציווד הקיים - `def view_database():`
2. הוספת פריט חדש למסד הנתונים - `def add_item():`
3. עריכת פריטים קיימים על ידי לחיצה על התא הרצוי - `def edit_item():`
4. מחיקת פריטים על ידי לחיצה על התא הרצוי - `def delete_item():`
5. ייצוא הנתונים לקובץ Excel - באמצעות `def export_to_xlsx():`
6. ייבוא נתונים מקובץ Excel - באמצעות `def import_from_xlsx():`
7. סינון ציווד לפי סוג הציווד - `def option_selected(option):`

הבדיקות שנעשות:

- בדיקת כפילויות - לא יתאפשר הוספה כפולה של ציווד (בודק על פי מספר GIP אם קיים ולאחר מכן לפי מספר סידורי)
- לא מאפשר עריכה של מזהה אם קיים או האם הוחזר

ישנם שתי מסכים נוספים, אחד נועד להלוואת ציוד והשני להחזרת הציוד

## מסך ההלוואה:

מאפשר השאלה של ציוד על ידי סטודנט לפי פרטי הזיהוי הבאים:

- שם הסטודנט

-תעודת זהות

-מספר טלפון

-דוא"ל

-מנחה פרויקט

על המשתמש להכניס או GIP Number או Serial Number של הפריט אותו הוא מעוניין להשאיל.

בנוסף ישנה אפשרות הוסיף תאריך החזרה צפוי ותאריך ההשאלה שימולא באופן אוטומטי לתאריך הנוכחי אם המשתמש לא מילא בעצמו.

בדיקות:

- המערכת לא מאפשרת הלוואה של ציוד ללא פרט מזהה 1 של סטודנט, פרט מזהה 1 של הציוד ( GIP Number או Serial Number ).
- המערכת לא תאפשר מהמשתמש להשאיל ציוד שאינו קיים במערכת.

הודעה מתאימה תופיע במידה ופעולה אסורה תתבצע.

הפעולות מתבצעות באמצעות הפונקציה `loan_item()` — מאמת את הנתונים, מוסיף את ההשאלה למסד הנתונים ומעדכן את מצב הציוד כ-"לא הוחזר".

בנוסף הפונקציה מרעננת את המסכים המציגים את הסטודנטים והציוד, ודואגת לנקות את השדות כדי שהשדות יהיו ריקים לקראת המילוי הבא על מנת לאפשר למשתמש נוחות במילוי רציף.

( הפונקציה הזו היא מעטפת ל `def add_loan(...):` שם מתבצעת עיקר הלוגיקה )

## החזרת ציוד:

`return_item(gipnum, serialnum)` — מעדכן את הציוד כמוחזר ומוחק את רשומת ההשאלה ממסד הנתונים.

גם במידה והציוד הוחזר נחזיר הודעת הצלחה כיוון שהציוד כבר קיים.

דרישות מערכת

Python 3.x

ספריות:

1. `tkinter` – ספריית GUI שמגיעה מובנית עם Python.
2. `sqlite3` – גישה למסד נתונים SQLite, מובנה ב־Python.
3. `datetime` – לטיפול בתאריכים ושעות.
4. `openpyxl` – קריאה וכתיבה של קובצי Excel (.xlsx). יש להתקין עם `pip`.

התקנה והרצה - רוב הספריות מובנות כברירת מחדל ב-Python חוץ מ openpyxl

איך ניצור קובץ exe מהאפליקציה? כתוב ב shell את הפקודות הבאות:

```
1. pip install pyinstaller
2. pyinstaller --onefile --windowed --hidden-import=openpyxl C:\Users\Public\...\main.py
   כאשר C:\Users\Public\...\main.py הוא ה path לקובץ py שלך
```

איך להפעיל?

השתמש בטאבים בממשק כדי לבצע השאלות, החזרות ולצפות ברשומות.

איך להוסיף אופצית סינון נוספת לעמוד view database שמציג את הציוד?

בהינתן ונרצה להוסיף את סוג ציוד בשם Tablet – נוסיף את השורה הבאה:

```
menu.add_command(label="Tablet", command=lambda: option_selected("Tablet"))
```

לאחר השורות האלו:

```
# Create a menu
menu = tk.Menu(view_db_tab, tearoff=0)
menu.add_command(label="Laptop", command=lambda: option_selected("Laptop"))
menu.add_command(label="Monitor", command=lambda: option_selected("Monitor"))
menu.add_command(label="PC", command=lambda: option_selected("PC"))
menu.add_command(label="Camera", command=lambda: option_selected("Camera"))
menu.add_command(label="Projector", command=lambda: option_selected("Projector"))
menu.add_command(label="Hard Disk", command=lambda: option_selected("Hard Disk"))
menu.add_command(label="Ptgray", command=lambda: option_selected("Ptgray"))
menu.add_command(label="Lens", command=lambda: option_selected("Lens"))
menu.add_command(label="Printer", command=lambda: option_selected("Printer"))
menu.add_command(label="Tripod", command=lambda: option_selected("Tripod"))
menu.add_command(label="Webcam", command=lambda: option_selected("Webcam"))
menu.add_command(label="Micro Ball head", command=lambda: option_selected("Micro Ball head"))
menu.add_command(label="Sensor", command=lambda: option_selected("Sensor"))
menu.add_command(label="MAC", command=lambda: option_selected("MAC"))
menu.add_command(label="Book", command=lambda: option_selected("Book"))
menu.add_command(label="Oculus", command=lambda: option_selected("Oculus"))
menu.add_command(label="Phone", command=lambda: option_selected("Phone"))
menu.add_command(label="Tablet", command=lambda: option_selected("Tablet"))
menu.add_command(label="Virtual reality", command=lambda: option_selected("Virtual reality"))
```

שימו לב: יש להקפיד על כתיבת סוג הציוד באופן מדויק.