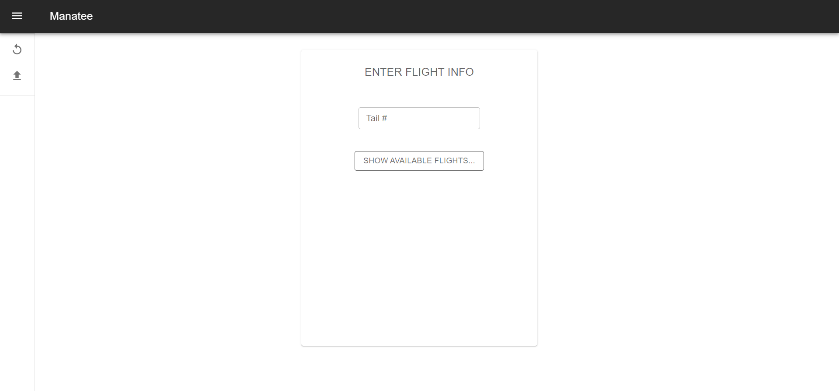
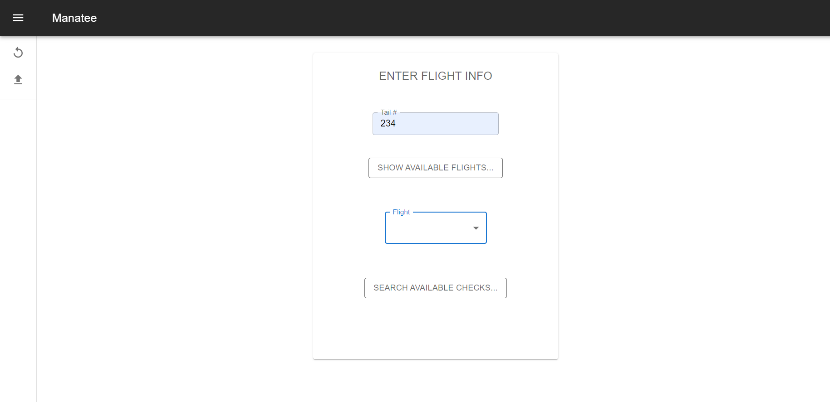
**מדריך לשימוש באתר- Manatee**

שימוש 1- תחקור טיסה אוטומטי

**שלב מס' 1: הכנסת פרטי הטיסה**

* יש להכניס מספר זנב של הטיסה שמעוניינים לתחקר
* לאחר מכן, ללחוץ על כפתור

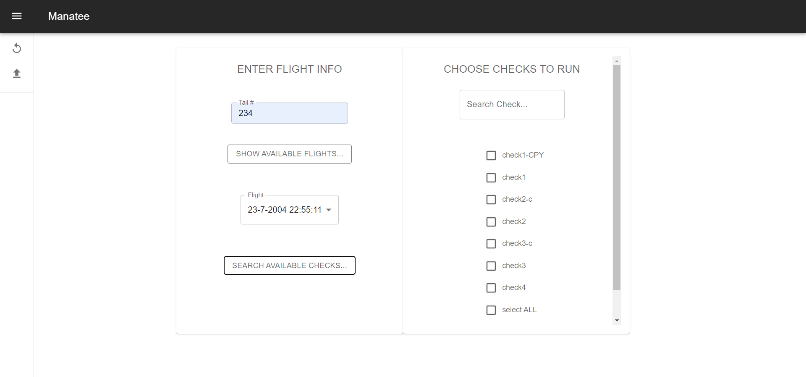
הSHOW AVAILABLE FLIGHTS… -

לאחר מכן, יופיע המסך הבא:

* יש לבחור את זמן החליבה של הטיסה שמעוניינים לתחקר
* לאחר מכן, ללחוץ על כפתור

הSEARCH AVAILABLE CHECKS… -

**שלב מס' 2: בחירת הבדיקות להרצה**

לאחר מכן, יופיע המסך הבא:

* יש לבחור את הבדיקות שמעוניינים להריץ
* לאחר מכן, ללחוץ על כפתור

הSEND CHECKS… -

**שלב מס' 3: תוצאות הבדיקות**

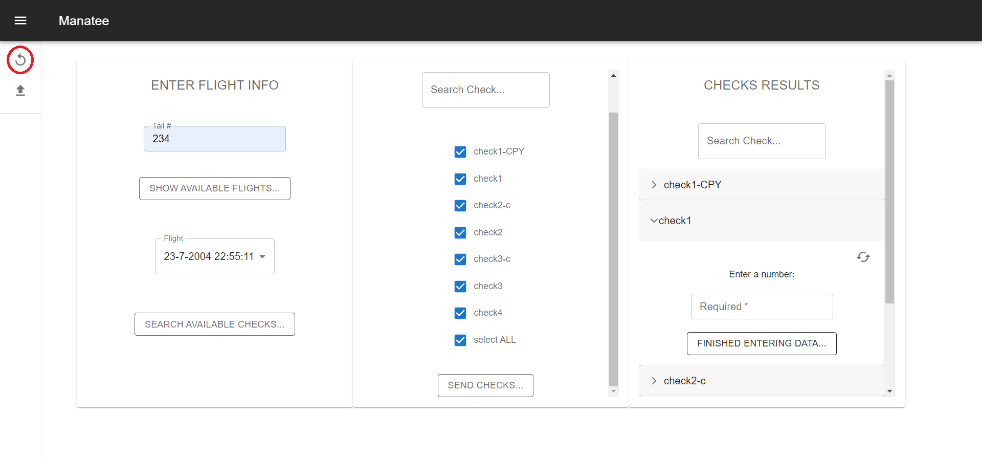
לאחר מכן, יופיע המסך הבא:

A screenshot of a computer

Description automatically generatedבמסך זה ניתן להסתכל על התוצאות של הבדיקות השונות שהורצו.

* יש לציין שבחלק מהבדיקות נדרש קלט מהמשתמש על מנת להשלים את ההרצה- בבדיקות אלה נדרש לעקוב אחרי ההוראות שמופיעות במסך הבדיקה.
* בכל הבדיקות מופיע כפתור אתחול- ניתן ללחוץ עליו על מנת להריץ את הבדיקה מההתחלה (רלוונטי בעיקר עבור בדיקות שדורשות קלט מהמשתמש, במקרה בו הוכנס קלט שגוי)

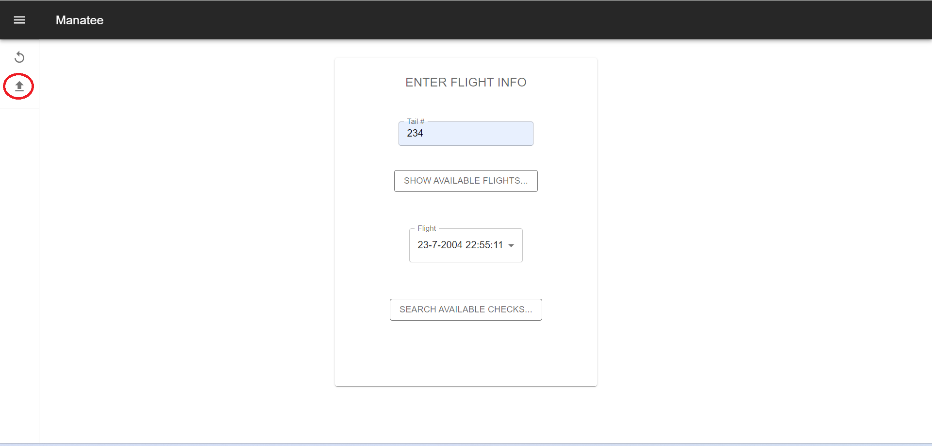
**פיצ'ר נוסף: טיסה חדשה**



ניתן ללחוץ על הכפתור הבא (המסומן באדום) על מנת לאתחל את מצב המערכת ולתחקר טיסה חדשה:

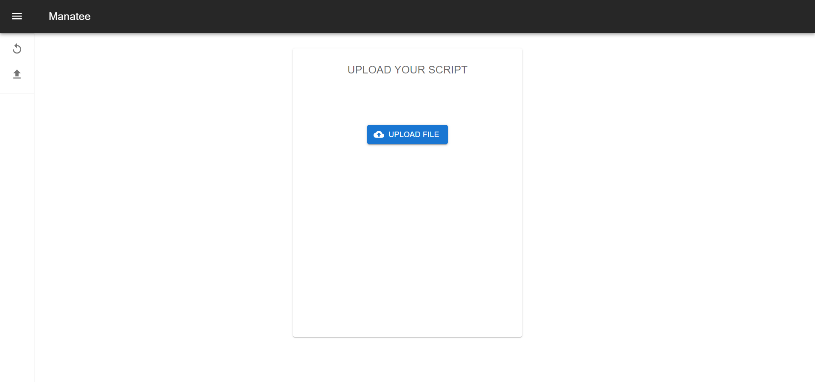
שימוש 2- העלאת בדיקה נוספת

**העלאת קובץ באתר:**



ניתן להעלות בדיקה חדשה על ידי לחיצה על הכפתור הבא (המסומן באדום):

לאחר מכן, יופיע המסך הבא:



במסך זה ניתן להעלות קובץ python (.py) שבו ממומשת בדיקה מסוימת- וכך מוסיפים בדיקות חדשות לאתר.

הסברים לגבי תוכן קבצי הבדיקות שמעלים נמצאים בחלק הבא של המסמך ("הסברים לגבי קבצי בדיקה")

הערה: יש להכניס סיסמת מנהל (makpitz) על מנת להעלות סקריפט בדיקה חדש. יש להכניס את הסיסמה כל 5 דק, לאחר פרק זמן זה הגישה תיחסם.

**הסברים לגבי קבצי בדיקה:**

* קובץ בדיקה מוכרח להכיל פונקציה עם החתימה הבאה:

def run\_check(tailNum, timestamp, socketio)

זוהי הפונקציה שתורץ בעת בחירת הבדיקה באתר.

הפרמטרים שלה:

tailNum- מספר הזנב של הטיסה

Timestamp- זמן החליבה של הטיסה

* בעזרת שני הפרמטרים הללו פונים ל-db

Socketio- פרמטר דרכו מתקשרים עם האתר (הסבר מפורט יותר בהמשך)

לקבצי בדיקה 2 אפשרויות להצגת דברים באתר עצמו:

1. הצגת טקסט: בעזרת הפונקציה הבאה:

request\_prompt(file\_name,request\_name,prompt,socketio)

הפרמטרים שלה:

File\_name- שם קובץ הפייתון

Request\_name- שם הבקשה- שם כלשהו לבחירתכם (פחות רלוונטי לבקשות להצגת טקסט)

Prompt- הטקסט שיוצג

Socketio- יש להעביר את ה- Socketioמחתימת הפונקציה run\_check

* (משתמשים בפונקציה זו בעיקר על מנת להציג את התוצאה הסופית של הבדיקה)

1. קלט מהמשתמש: בעזרת הפונקציה הבאה:

request\_input(file\_name, request\_name, content,socketio)

הפרמטרים שלה:

File\_name- שם קובץ הפייתון

Request\_name\*- שם הבקשה- שם כלשהו לבחירתכם (החשיבות שלו תוסבר בהמשך)

content- הטקסט שיוצג מעל המקום בו מכניסים קלט

Socketio- יש להעביר את ה- Socketioמחתימת הפונקציה run\_check

* הפונקציות הללו ממומשות בקובץ manager.py, ועל מנת לייבא אותן לקובץ שלכם יש להוסיף את השורות הבאות בראשו:

import sys

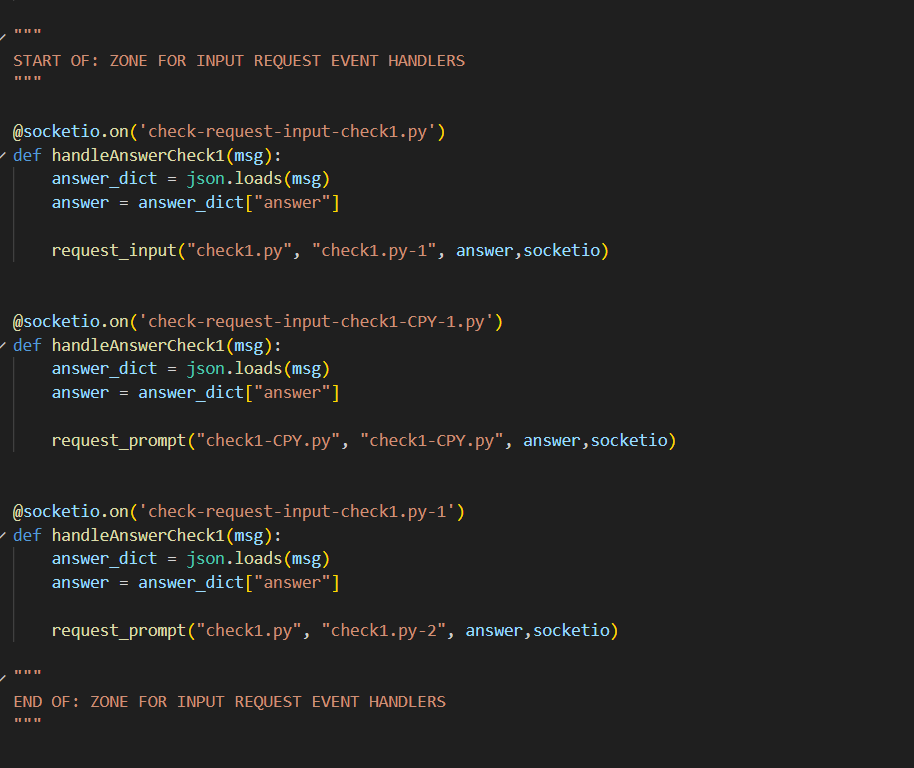
sys.path.append('../')

from manager import request\_prompt

from manager import request\_input

על מנת לקבל את הקלט מהמשתמש ולקודד את המשך הבדיקה, יש לפעול כך:

הקוד ייכתב בקובץ ששמו manager.py שנמצא בשרת, בחלק מיועד, שנראה כך:



כפי שניתן לראות- על כל פונקציה בחלק זה של הקובץ להתחיל בשורה:

@socketio.on('check-request-input-{ Request\_name }')

\*Request\_name- אותו פרמטר שהועבר לפונקציה request\_input

הפונקציה חייבת לקבל פרמטר יחיד, והשורות הבאות מחלצות את הקלט שהוכנס באתר, ואחריהן במשתנה answer יימצא הקלט שהועבר (string!):

answer\_dict = json.loads(msg)

answer = answer\_dict["answer"]

ניתן לאחר מכן לממש את המשך הבדיקה ולבקש בקשות לקלט/להציג טקסט כרצוי.

קובץ בדיקה במלואו לדוגמה:

בבדיקה זו מבקשים קלט מהמשתמש ומציגים את הקלט שהוא הכניס לאחר מכן כטקסט:

Check1-CPY.py:

import sys

sys.path.append('../')

from manager import request\_input

def run\_check(\_\_,\_,socketio):

    request\_input("check1-CPY.py", "check1-CPY-1.py", "Enter a number:", socketio)

manager.py:

@socketio.on('check-request-input-check1-CPY-1.py')

def handleAnswerCheck1(msg):

    answer\_dict = json.loads(msg)

    answer = answer\_dict["answer"]

    request\_prompt("check1-CPY.py", "check1-CPY.py", answer,socketio)

הבדיקה באתר:

Pressing ‘FINISHED ENTERING DATA…’

Inserting 12

1

2

3

