

# Test execution report

## 1. פתיחה

מטרת הבדיקות הייתה לוודא שהמערכת מתפקדת באופן תקין בעת העלאת קבצים תקינים, ולזהות בצורה מדויקת בעיות כאשר מועלים קבצים פגומים או חריגים – כולל שדות חסרים, ערכים שגויים, או קבצים שלא עומדים בפורמט הנדרש. סיכום כללי של הביצוע

## 2. תיעוד תוצאות – דוח HTML

כחלק מתהליך האימות האוטומטי, הופק דוח HTML מלא המפרט את כל תוצאות ההרצה (כולל סטטוס לכל טסט, זמני ריצה והודעות Assertion מצ"ב קישור לדוח

[file:///Users/pelegoberlender/Desktop/school/home\\_assignment/report.html?sort=result](file:///Users/pelegoberlender/Desktop/school/home_assignment/report.html?sort=result)

## 3. גישת הבדיקה

כלל הבדיקות בוצעו מקומית דרך הטרמינל לאחר הפעלת השרת. הבדיקות כיסו קבצים תקינים, קבצים עם שגיאות צפויות, ותסריטים חריגים. כל בדיקה כללה אינטראקציה מלאה עם ממשק המשתמש, העלאת קובץ, המתנה לתוצאה, והשוואת הפלט לקובץ JSON המצופה.

## 4. סוגי הבדיקות שבוצעו

- I. Valid files – בדיקות אלו וידאו שהמערכת מזהה ומקבלת קבצים תקינים, ומציגה את הפלט באופן תואם לקובץ Invalid files - בדיקות אלו כללו העלאת קבצים לא תקינים (לדוגמה: מחירים לא תקינים, שם חסר) תוך ציפייה לראות את הודעות השגיאה הצפויות מהמערכת – גם כאן נעשה שימוש בקובץ expected\_results.json כנקודת השוואה.
- II. בדיקות שגיאה - בדיקות אלו כללו העלאת קבצים לא תקינים (לדוגמה: מחירים לא תקינים, שם חסר) תוך ציפייה לראות את הודעות השגיאה הצפויות מהמערכת – גם כאן נעשה שימוש בקובץ expected\_results.json כנקודת השוואה.
- III. Edge cases - בדיקות אלו נועדו לבדוק עמידות של המערכת בסיטואציות לא סטנדרטיות, במטרה לוודא שהיא מגיבה באלגנטיות גם כאשר מתקבלים קבצים בעייתיים במיוחד. מצ"ב טבלה המפרטת את מקרי הקצה, והצגת הפלט מהאתר והתוצאה הצפויה בקובץ json

	פלט מהאתר	תוצאה צפויה כפי שמוצפת בטופס expected
Empty file	<div>Choose File empty_file.csv Upload</div> <pre>{   "status": "error",   "message": "No columns to parse from file" }</pre>	
Missing header	<div>Choose File missing_headers.csv Upload</div> <pre>{   "status": "error",   "message": "'name'" }</pre>	<pre>"missing_headers": {   "total": 0,   "success_count": 0,   "error_count": 0,   "expected_errors": [     "Missing required column: name",     "Missing required column: price"   ] },</pre>
Missing cell	<pre>{   "status": "error",   "errors": [     "Missing name in row 2"   ] }</pre>	<pre>{   "empty_file": {     "total": 0,     "success_count": 0,     "error_count": 0,     "expected_errors": []   },</pre>
Non numeric price	<pre>{   "status": "error",   "errors": [     "Invalid price in row 0",     "Invalid price in row 1",     "Invalid price in row 2"   ] }</pre>	<pre>"non_numeric_price": {   "total": 2,   "success_count": 1,   "error_count": 1,   "expected_errors": [     "Invalid price in row 2"   ] },</pre>
Mixed validity	<pre>{   "status": "error",   "errors": [     "Missing name in row 1",     "Invalid price in row 4"   ] }</pre>	<pre>"mixed_validity": {   "total": 5,   "success_count": 3,   "error_count": 2,   "expected_errors": [     "Missing name in row 1",     "Invalid price in row 4"   ] },</pre>

Text in numric	<div>Choose File text_in_numeric.csv Upload</div> <pre>{   "status": "success",   "data": [     {       "id": 1,       "name": "Laptop",       "price": 999.99,       "category": "Electronics",       "stock": "50"     },     {       "id": 2,       "name": "Headphones",       "price": 500,       "category": "Electronics",       "stock": "hundred"     },     {       "id": 3,       "name": "Mouse",       "price": 29.99,       "category": "Electronics",       "stock": "200"     }   ] }</pre>	<pre>}, "text_in_numeric": {   "total": 3,   "success_count": 2,   "error_count": 1,   "expected_errors": [     "Invalid stock in row 1"   ] },</pre>
Non csv format		<pre>"file_not_found": {   "status": "error",   "success_count": 0,   "error_count": 1,   "expected_errors": ["No such file or directory"] }</pre>
File not found		<pre>"non_csv_format": {   "status": "error",   "success_count": 0,   "error_count": 1,   "expected_errors": ["Invalid file format. Please upload a CSV file."] }</pre>