



**BIRZEIT UNIVERSITY**

**Faculty of Engineering and Technology Electrical  
and Computer Engineering Department**

**Linux Laboratory**

**ENCS3130**

---

**PROJECT #2 – Python**

---

Prepared by:

**Lana Musaffer 1210455**  
**Mayar Masalmeh 1211246**

Instructor: Dr. Ayman Hroub

Teaching assistant: Eng. Tarek Zaidan

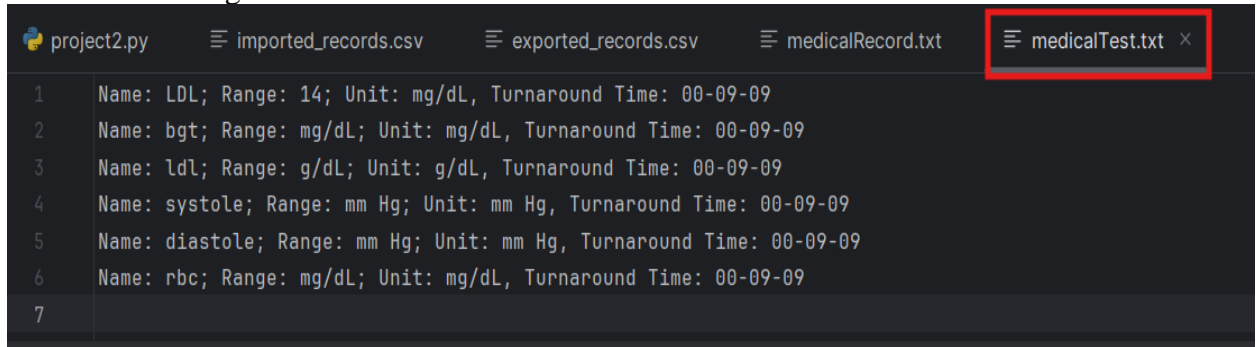
Section: 6

BIRZEIT

Date: 23<sup>rd</sup>.August.2024

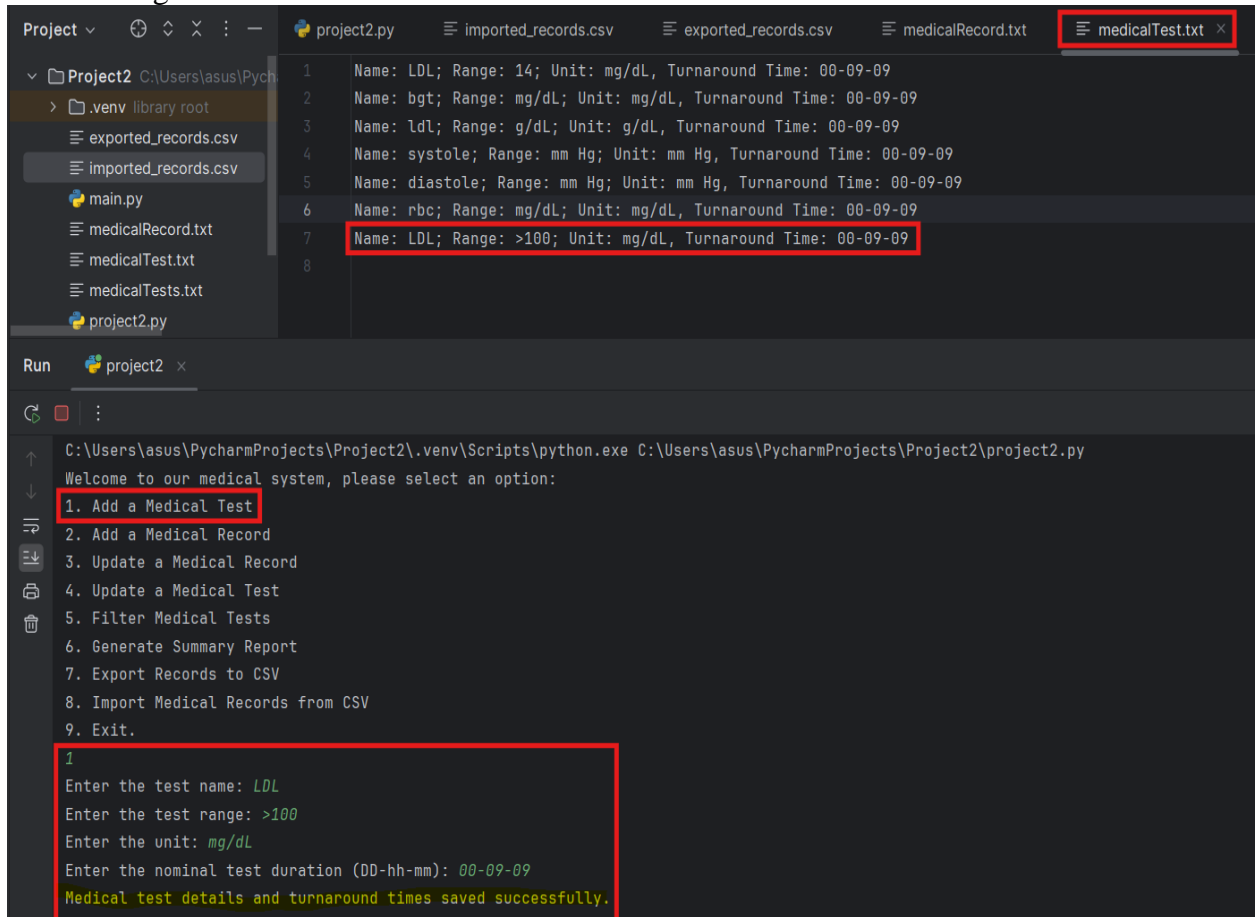
## Add new medical test

File before adding a new medical test:



```
project2.py  imported_records.csv  exported_records.csv  medicalRecord.txt  medicalTest.txt x
1  Name: LDL; Range: 14; Unit: mg/dL, Turnaround Time: 00-09-09
2  Name: bgt; Range: mg/dL; Unit: mg/dL, Turnaround Time: 00-09-09
3  Name: ldL; Range: g/dL; Unit: g/dL, Turnaround Time: 00-09-09
4  Name: systole; Range: mm Hg; Unit: mm Hg, Turnaround Time: 00-09-09
5  Name: diastole; Range: mm Hg; Unit: mm Hg, Turnaround Time: 00-09-09
6  Name: rbc; Range: mg/dL; Unit: mg/dL, Turnaround Time: 00-09-09
7
```

After adding a new medical test:



```
Project  v  project2.py  imported_records.csv  exported_records.csv  medicalRecord.txt  medicalTest.txt x
v  Project2  C:\Users\asus\PycharmProjects\Project2
  > .venv library root
    exported_records.csv
    imported_records.csv
    main.py
    medicalRecord.txt
    medicalTest.txt
    medicalTests.txt
    project2.py
1  Name: LDL; Range: 14; Unit: mg/dL, Turnaround Time: 00-09-09
2  Name: bgt; Range: mg/dL; Unit: mg/dL, Turnaround Time: 00-09-09
3  Name: ldL; Range: g/dL; Unit: g/dL, Turnaround Time: 00-09-09
4  Name: systole; Range: mm Hg; Unit: mm Hg, Turnaround Time: 00-09-09
5  Name: diastole; Range: mm Hg; Unit: mm Hg, Turnaround Time: 00-09-09
6  Name: rbc; Range: mg/dL; Unit: mg/dL, Turnaround Time: 00-09-09
7  Name: LDL; Range: >100; Unit: mg/dL, Turnaround Time: 00-09-09
8

Run  project2 x
C:\Users\asus\PycharmProjects\Project2\.venv\Scripts\python.exe C:\Users\asus\PycharmProjects\Project2\project2.py
Welcome to our medical system, please select an option:
1. Add a Medical Test
2. Add a Medical Record
3. Update a Medical Record
4. Update a Medical Test
5. Filter Medical Tests
6. Generate Summary Report
7. Export Records to CSV
8. Import Medical Records from CSV
9. Exit.
1
Enter the test name: LDL
Enter the test range: >100
Enter the unit: mg/dL
Enter the nominal test duration (DD-hh-mm): 00-09-09
Medical test details and turnaround times saved successfully.
```

## Add new medical record

File before adding a new medical record:

	project2.py	imported_records.csv	exported_records.csv	medicalRecord.txt	medicalTest.txt
1	1300500: LdL, 2024-09-09 13:50, 14, g/dL, Completed, 2024-09-19 18:33				
2	1300500: RBC, 2024-03-01 05:20, 13.5, mg/dL, Completed, 2024-03-01 05:30				
3	1300511: LDL, 2024-03-02 07:30, 110, mg/dL, Pending				
4	1300520: systole, 2024-03-04 04:40, 150, mm Hg, Pending				
5	1111111: LDL, 2024-09-16 13:50, 80, mg/dL, Completed, 2024-09-18 04:01				
6	1111111: Hgb, 2024-08-09 13:50, 110, g/dL, Pending				
7					
8					

After adding a new medical record:

Project

Project2 C:\Users\asus\PyCharm\project2

.venv library root

exported\_records.csv

imported\_records.csv

main.py

medicalRecord.txt

1 1300500: LdL, 2024-09-09 13:50, 14, g/dL, Completed, 2024-09-19 18:33

2 1300500: RBC, 2024-03-01 05:20, 13.5, mg/dL, Completed, 2024-03-01 05:30

3 1300511: LDL, 2024-03-02 07:30, 110, mg/dL, Pending

4 1300520: systole, 2024-03-04 04:40, 150, mm Hg, Pending

5 1111111: LDL, 2024-09-16 13:50, 80, mg/dL, Completed, 2024-09-18 04:01

6 1111111: Hgb, 2024-08-09 13:50, 110, g/dL, Pending

7 1234567: LdL, 2024-09-09 13:44, 80, mg/dL, Completed, 2024-09-12 04:33

Run project2

8. Import Medical Records from CSV

9. Exit.

2

Enter patient ID (7 digits): 123456

Invalid ID! Please enter a 7-digit numeric ID:

Enter patient ID (7 digits): 123456k

Invalid ID! Please enter a 7-digit numeric ID:

Enter patient ID (7 digits): 12345\_4

Invalid ID! Please enter a 7-digit numeric ID:

Enter patient ID (7 digits): 1234567

Available test names:

LdL

Bgt

Systole

Diastole

Rbc

Enter the test name from the list above: LdL

Enter test date and time (YYYY-MM-DD hh:mm): 2024-09-09 13:44

Enter numeric result value: 80

Enter results unit: mm Hg

Invalid unit! The correct unit for LdL is mg/dL.

Enter results unit: mg/dL

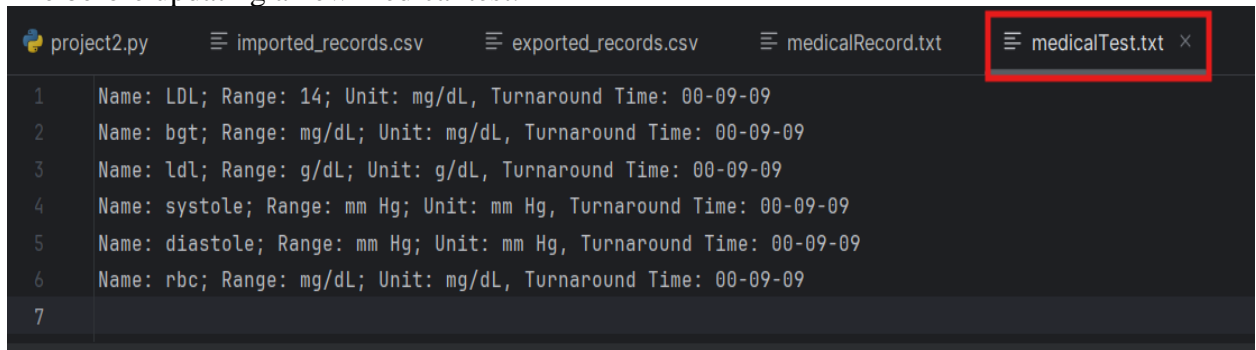
Enter status (Pending, Completed, or Reviewed): completed

Enter results date and time (YYYY-MM-DD hh:mm): 2024-09-12 04:33

Medical record added successfully.

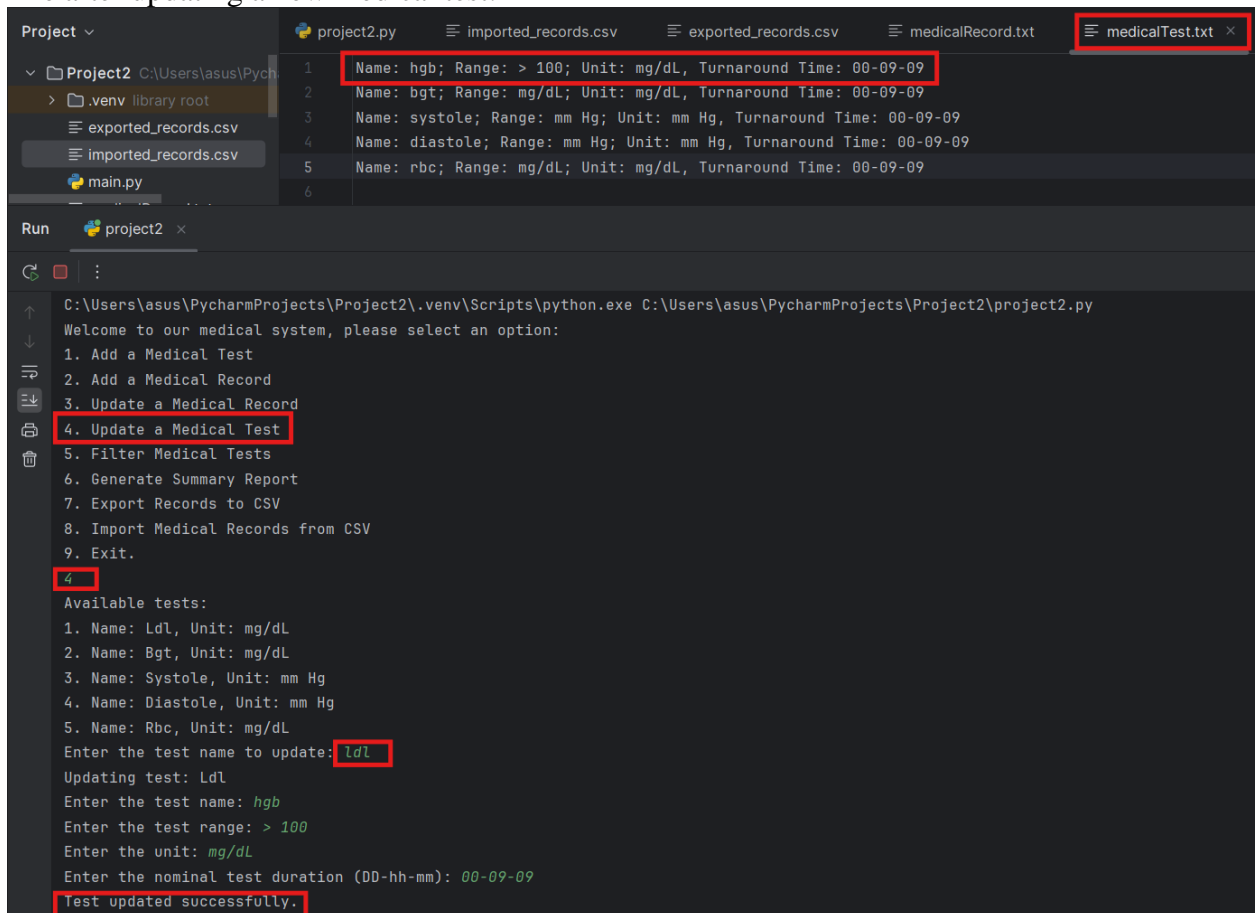
## Update medical tests

File before updating a new medical test:



```
project2.py  imported_records.csv  exported_records.csv  medicalRecord.txt  medicalTest.txt x
1  Name: LDL; Range: 14; Unit: mg/dL, Turnaround Time: 00-09-09
2  Name: bgt; Range: mg/dL; Unit: mg/dL, Turnaround Time: 00-09-09
3  Name: ldl; Range: g/dL; Unit: g/dL, Turnaround Time: 00-09-09
4  Name: systole; Range: mm Hg; Unit: mm Hg, Turnaround Time: 00-09-09
5  Name: diastole; Range: mm Hg; Unit: mm Hg, Turnaround Time: 00-09-09
6  Name: rbc; Range: mg/dL; Unit: mg/dL, Turnaround Time: 00-09-09
7
```

File after updating a new medical test:



```
Project  project2.py  imported_records.csv  exported_records.csv  medicalRecord.txt  medicalTest.txt x
Project2 C:\Users\asus\PycharmProjects\Project2
> .venv library root
  exported_records.csv
  imported_records.csv
  main.py
Run  project2 x
C:\Users\asus\PycharmProjects\Project2\.venv\Scripts\python.exe C:\Users\asus\PycharmProjects\Project2\project2.py
Welcome to our medical system, please select an option:
1. Add a Medical Test
2. Add a Medical Record
3. Update a Medical Record
4. Update a Medical Test
5. Filter Medical Tests
6. Generate Summary Report
7. Export Records to CSV
8. Import Medical Records from CSV
9. Exit.
4
Available tests:
1. Name: Ldl, Unit: mg/dL
2. Name: Bgt, Unit: mg/dL
3. Name: Systole, Unit: mm Hg
4. Name: Diastole, Unit: mm Hg
5. Name: Rbc, Unit: mg/dL
Enter the test name to update: Ldl
Updating test: Ldl
Enter the test name: hgb
Enter the test range: > 100
Enter the unit: mg/dL
Enter the nominal test duration (DD-hh-mm): 00-09-09
Test updated successfully.
```

## Update medical records

File before updating a new medical record:

	project2.py	imported_records.csv	exported_records.csv	medicalRecord.txt	medicalTest.txt
1	1300500: Ldl, 2024-09-09 13:50, 14, g/dL, Completed, 2024-09-19 18:33				
2	1300500: Rbc, 2024-03-01 05:20, 13.5, mg/dL, Completed, 2024-03-01 05:30				
3	1300511: LDL, 2024-03-02 07:30, 110, mg/dL, Pending				
4	1300520: systole, 2024-03-04 04:40, 150, mm Hg, Pending				
5	1111111: LDL, 2024-09-16 13:50, 80, mg/dL, Completed, 2024-09-18 04:01				
6	1111111: Hgb, 2024-08-09 13:50, 110, g/dL, Pending				
7					
8					

When the user selects option 3:

```
Welcome to our medical system, please select an option:
1. Add a Medical Test
2. Add a Medical Record
3. Update a Medical Record
4. Update a Medical Test
5. Filter Medical Tests
6. Generate Summary Report
7. Export Records to CSV
8. Import Medical Records from CSV
9. Exit.
3
Available records:
1. 1300500: Ldl, 2024-09-09 13:50, 14, g/dL, Completed, 2024-09-19 18:33
2. 1300500: Rbc, 2024-03-01 05:20, 13.5, mg/dL, Completed, 2024-03-01 05:30
3. 1300511: LDL, 2024-03-02 07:30, 110, mg/dL, Pending,
4. 1300520: systole, 2024-03-04 04:40, 150, mm Hg, Pending,
5. 1111111: LDL, 2024-09-16 13:50, 80, mg/dL, Completed, 2024-09-18 04:01
6. 1111111: Hgb, 2024-08-09 13:50, 110, g/dL, Pending,
Select the record number to update: 4
Select the record number to update: 4
Updating record for Patient ID: 1300520 and Test: systole
Enter patient ID (7 digits): 1450200
Available test names:
Ldl
Bgt
Systole
Diastole
Rbc
Enter the test name from the list above: Rbc
Enter test date and time (YYYY-MM-DD hh:mm): 2027-08-17 12:55
Enter numeric result value: 14.9
Enter results unit: mg/dL
Enter status (Pending, Completed, or Reviewed): Reviewed
Record updated successfully.
```

File after updating a new medical record:

	Project2.py	medicalTest.txt	medicalRecord.txt
1	1300500: Ldl, 2024-09-09 13:50, 14, g/dL, Completed, 2024-09-19 18:33		
2	1300500: Rbc, 2024-03-01 05:20, 13.5, mg/dL, Completed, 2024-03-01 05:30		
3	1300511: LDL, 2024-03-02 07:30, 110, mg/dL, Pending		
4	1300520: Rbc, 2027-08-17 12:55, 14.9, mg/dL, Reviewed		
5	1111111: LDL, 2024-09-16 13:50, 80, mg/dL, Completed, 2024-09-18 04:01		
6	1111111: Hgb, 2024-08-09 13:50, 110, g/dL, Pending		
7			

## Filter medical tests

Filter records with patients with id = 1300500:

The screenshot shows a Python IDE with a project named 'project2'. The file explorer on the left shows a directory structure with files like 'imported\_records.csv', 'exported\_records.csv', 'main.py', 'medicalRecord.txt', 'medicalTest.txt', and 'medicalTests.txt'. The main editor displays a CSV file with 10 rows of medical records. The first two rows are highlighted with a red box:

Patient ID	Test Name	Test Date	Result	Status	Results Date
1300500	LdL	2024-09-09 13:50	14	g/dL, Completed	2024-09-19 18:33
1300500	RBC	2024-03-01 05:20	13.5	mg/dL, Completed	2024-03-01 05:30

The terminal window shows the output of a program. It prompts the user to select an option from a list. Option 5, 'Filter Medical Tests', is selected. The program then prompts the user to filter by: 1. Patient ID, 2. Test Name, 3. Abnormal Tests, 4. Test Added Within a Specific Period, 5. Test Status. Option 1, 'Patient ID', is selected. The program then prompts the user to enter filter criteria number(s) separated by commas (e.g., 1,2). The user enters '1'. The program then prompts the user to enter Patient ID to filter. The user enters '1300500'. The program then displays the filtered records:

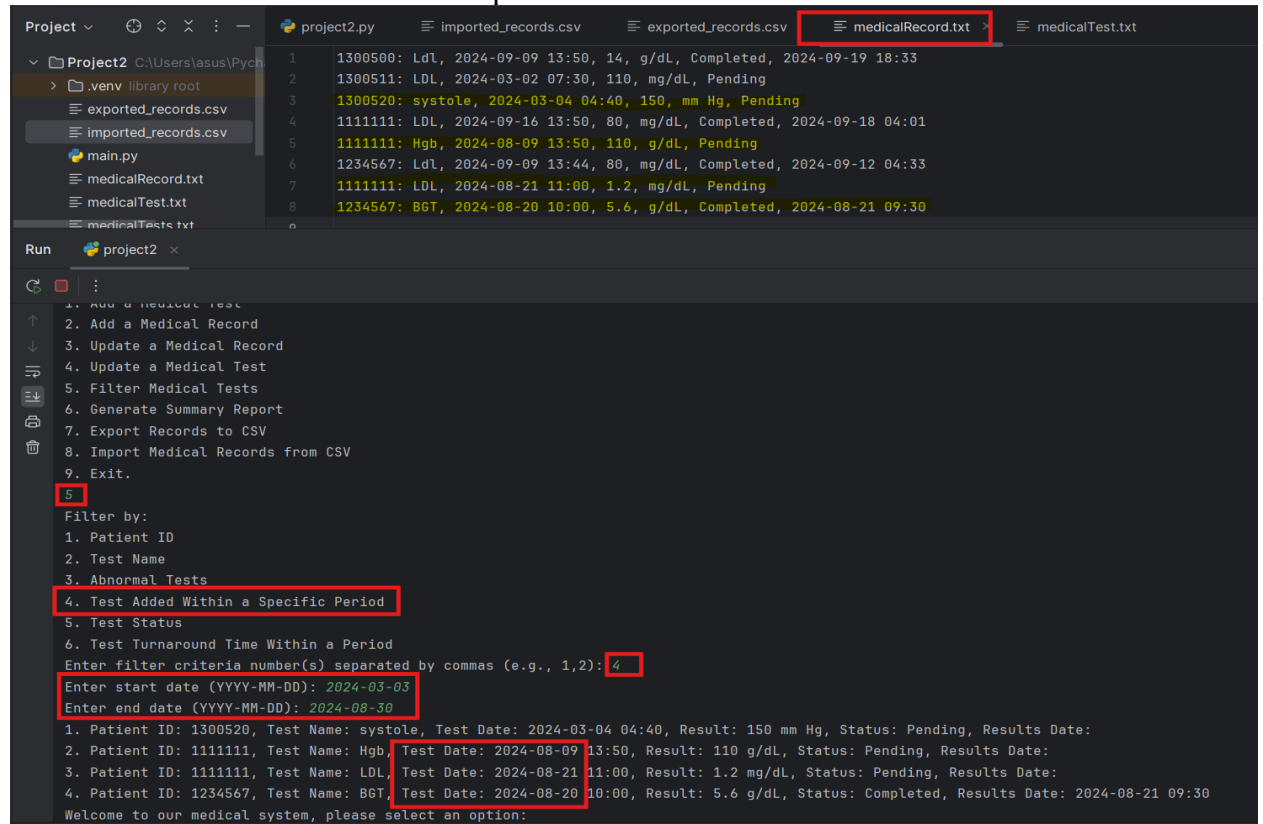
```
1. Patient ID: 1300500, Test Name: LdL, Test Date: 2024-09-09 13:50, Result: 14 g/dL, Status: Completed, Results Date: 2024-09-19 18:33
2. Patient ID: 1300500, Test Name: RBC, Test Date: 2024-03-01 05:20, Result: 13.5 mg/dL, Status: Completed, Results Date: 2024-03-01 05:30
```

Filter records with abnormal tests and completed status:

The screenshot shows the same Python IDE as before. The file explorer on the left shows the same directory structure. The main editor displays the same CSV file. The terminal window shows the output of the program. It prompts the user to select an option from a list. Option 5, 'Filter Medical Tests', is selected. The program then prompts the user to filter by: 1. Patient ID, 2. Test Name, 3. Abnormal Tests, 4. Test Added Within a Specific Period, 5. Test Status. Option 3, 'Abnormal Tests', is selected. The program then prompts the user to enter filter criteria number(s) separated by commas (e.g., 1,2). The user enters '3,5'. The program then prompts the user to enter Test Status to filter (Pending, Completed, or Reviewed). The user enters 'Completed'. The program then displays the filtered records:

```
1. Patient ID: 1300500, Test Name: LdL, Test Date: 2024-09-09 13:50, Result: 14 g/dL, Status: Completed, Results Date: 2024-09-19 18:33
2. Patient ID: 1111111, Test Name: LDL, Test Date: 2024-09-16 13:50, Result: 80 mg/dL, Status: Completed, Results Date: 2024-09-18 04:01
3. Patient ID: 1234567, Test Name: LdL, Test Date: 2024-09-09 13:44, Result: 80 mg/dL, Status: Completed, Results Date: 2024-09-12 04:33
4. Patient ID: 1234567, Test Name: BGT, Test Date: 2024-08-20 10:00, Result: 5.6 g/dL, Status: Completed, Results Date: 2024-08-21 09:30
```

Filter records with tests based on test period:



```
Project2 C:\Users\asus\PyCharm\project2.py imported_records.csv exported_records.csv medicalRecord.txt medicalTest.txt
> .venv library root
  exported_records.csv
  imported_records.csv
  main.py
  medicalRecord.txt
  medicalTest.txt
  medicalTests.txt

Run project2 x

1. Add a Medical Test
2. Add a Medical Record
3. Update a Medical Record
4. Update a Medical Test
5. Filter Medical Tests
6. Generate Summary Report
7. Export Records to CSV
8. Import Medical Records from CSV
9. Exit.
5
Filter by:
1. Patient ID
2. Test Name
3. Abnormal Tests
4. Test Added Within a Specific Period
5. Test Status
6. Test Turnaround Time Within a Period
Enter filter criteria number(s) separated by commas (e.g., 1,2): 4
Enter start date (YYYY-MM-DD): 2024-03-03
Enter end date (YYYY-MM-DD): 2024-08-30
1. Patient ID: 1300520, Test Name: systole, Test Date: 2024-03-04 04:40, Result: 150 mm Hg, Status: Pending, Results Date:
2. Patient ID: 1111111, Test Name: Hgb, Test Date: 2024-08-09 13:50, Result: 110 g/dL, Status: Pending, Results Date:
3. Patient ID: 1111111, Test Name: LDL, Test Date: 2024-08-21 11:00, Result: 1.2 mg/dL, Status: Pending, Results Date:
4. Patient ID: 1234567, Test Name: BGT, Test Date: 2024-08-20 10:00, Result: 5.6 g/dL, Status: Completed, Results Date: 2024-08-21 09:30
Welcome to our medical system, please select an option:
```

## Generate textual summary reports

The screenshot shows a Python IDE with a project named 'Project2'. The file explorer on the left lists files: `exported_records.csv`, `imported_records.csv`, `main.py`, `medicalRecord.txt`, `medicalTest.txt`, and `medicalTests.txt`. The editor displays a CSV file with 9 lines of medical data. The terminal window shows the execution of a script with the following steps:

```
7. Export records to CSV
8. Import Medical Records from CSV
9. Exit.
6
Filter by:
1. Patient ID
2. Test Name
3. Abnormal Tests
4. Test Added Within a Specific Period
5. Test Status
6. Test Turnaround Time Within a Period
Enter filter criteria number(s) separated by commas (e.g., 1,2): 2,3
Enter Test Name to filter: Ldl
1. Patient ID: 1300500, Test Name: Ldl, Test Date: 2024-09-09 13:50, Result: 14 g/dL, Status: Completed, Results Date: 2024-09-19 18:33
2. Patient ID: 1234567, Test Name: Ldl, Test Date: 2024-09-09 13:44, Result: 80 mg/dL, Status: Completed, Results Date: 2024-09-12 04:33

Summary Report:
Minimum Test Value: 14.00
Maximum Test Value: 80.00
Average Test Value: 47.00
Minimum Turnaround Time: 3769.00 minutes
Maximum Turnaround Time: 14683.00 minutes
Average Turnaround Time: 9226.00 minutes
Welcome to our medical system, please select an option:
```

The data in the CSV file is as follows:

Line	Patient ID	Test Name	Test Date	Result	Status	Results Date
1	1300500	Ldl	2024-09-09 13:50	14 g/dL	Completed	2024-09-19 18:33
2	1300511	LDL	2024-03-02 07:30	110 mg/dL	Pending	
3	1300520	systole	2024-03-04 04:40	150 mm Hg	Pending	
4	1111111	LDL	2024-09-16 13:50	80 mg/dL	Completed	2024-09-18 04:01
5	1111111	Hgb	2024-08-09 13:50	110 g/dL	Pending	
6	1234567	Ldl	2024-09-09 13:44	80 mg/dL	Completed	2024-09-12 04:33
7	1111111	LDL	2024-08-21 11:00	1.2 mg/dL	Pending	
8	1234567	BGT	2024-08-20 10:00	5.6 g/dL	Completed	2024-08-21 09:30
9						



## Export medical records to a comma separated file

```
8. Import Medical Records from CSV
9. Exit.
6
Filter by:
1. Patient ID
2. Test Name
3. Abnormal Tests
4. Test Added Within a Specific Period
5. Test Status
6. Test Turnaround Time Within a Period
Enter filter criteria number(s) separated by commas (e.g., 1,2): 2,3
Enter Test Name to filter: LdL
1. Patient ID: 1300500, Test Name: LdL, Test Date: 2024-09-09 13:50, Result: 14 g/dL, Status: Completed, Results Date: 2024-09-19 18:33
```

```
Summary Report:
Minimum Test Value: 14.00
Maximum Test Value: 14.00
Average Test Value: 14.00
Minimum Turnaround Time: 14683.00 minutes
Maximum Turnaround Time: 14683.00 minutes
Average Turnaround Time: 14683.00 minutes
Welcome to our medical system, please select an option:
1. Add a Medical Test
2. Add a Medical Record
3. Update a Medical Record
4. Update a Medical Test
5. Filter Medical Tests
6. Generate Summary Report
7. Export Records to CSV
```

Welcome to our medical system, please select an option:

1. Add a Medical Test
2. Add a Medical Record
3. Update a Medical Record
4. Update a Medical Test
5. Filter Medical Tests
6. Generate Summary Report
7. Export Records to CSV
8. Import Medical Records from CSV
9. Exit.

7

Enter filename for export: *ExportedRecords.csv*

Records successfully exported to ExportedRecords.csv

Project2.py    ExportedRecords.csv ×    medicalTest.txt    importedRcord.csv    medicalRecord.txt    ⋮

1	Patient ID,Test Name,Test Date,Result,Unit,Status,Results Date	✓
2	1300500,LdL,2024-09-09 13:50,14,g/dL,Completed,2024-09-19 18:33	
3		



## Import medical records from a comma separated file

Contents in imported\_records.csv file:

	project2.py	imported_records.csv	exported_records.csv	medicalRecord.txt	medicalTest.txt
1	Patient ID,Test Name,Test Date,Result,Unit,Status,Results Date				
2	1111111,LDL,2024-08-21 11:00,1.2,mg/dL,Pending				
3	1234567,BGT,2024-08-20 10:00,5.6,g/dL,Completed,2024-08-21 09:30				
4					

medicalRecord.txt file before:

	project2.py	imported_records.csv	exported_records.csv	medicalRecord.txt	medicalTest.txt
1	1300500: LdL, 2024-09-09 13:50, 14, g/dL, Completed, 2024-09-19 18:33				
2	1300500: RBC, 2024-03-01 05:20, 13.5, mg/dL, Completed, 2024-03-01 05:30				
3	1300511: LDL, 2024-03-02 07:30, 110, mg/dL, Pending				
4	1300520: systole, 2024-03-04 04:40, 150, mm Hg, Pending				
5	1111111: LDL, 2024-09-16 13:50, 80, mg/dL, Completed, 2024-09-18 04:01				
6	1111111: Hgb, 2024-08-09 13:50, 110, g/dL, Pending				
7	1234567: LdL, 2024-09-09 13:44, 80, mg/dL, Completed, 2024-09-12 04:33				
8					

medicalRecord.txt file after:

Project ▾

- Project2 C:\Users\asus\PyCh...
- venv library root
  - exported\_records.csv
  - imported\_records.csv
  - main.py
  - medicalRecord.txt
  - medicalTest.txt
  - medicalTests.txt
  - project2.py
- External Libraries

	project2.py	imported_records.csv	exported_records.csv	medicalRecord.txt	medicalTest.txt
1	1300500: LdL, 2024-09-09 13:50, 14, g/dL, Completed, 2024-09-19 18:33				
2	1300500: RBC, 2024-03-01 05:20, 13.5, mg/dL, Completed, 2024-03-01 05:30				
3	1300511: LDL, 2024-03-02 07:30, 110, mg/dL, Pending				
4	1300520: systole, 2024-03-04 04:40, 150, mm Hg, Pending				
5	1111111: LDL, 2024-09-16 13:50, 80, mg/dL, Completed, 2024-09-18 04:01				
6	1111111: Hgb, 2024-08-09 13:50, 110, g/dL, Pending				
7	1234567: LdL, 2024-09-09 13:44, 80, mg/dL, Completed, 2024-09-12 04:33				
8	1111111: LDL, 2024-08-21 11:00, 1.2, mg/dL, Pending				
9	1234567: BGT, 2024-08-20 10:00, 5.6, g/dL, Completed, 2024-08-21 09:30				
10					

Run project2 ×

1. Add a Medical Test

2. Add a Medical Record

3. Update a Medical Record

4. Update a Medical Test

5. Filter Medical Tests

6. Generate Summary Report

7. Export Records to CSV

8. Import Medical Records from CSV

9. Exit.

8

Enter filename to import records from: imported\_records.csv

Records successfully imported from imported\_records.csv.

welcome to our medical system, please select an option.