

**HNO:2503A52L10**

## AI Assisted Coding (Lab Test-3)

### Task1:

Scenario: In the domain of Healthcare, a company is facing a challenge related to data structures with ai.

Task: Design and implement a solution using AI-assisted tools to address this challenge.

Include code, explanation of AI integration, and test results.

Deliverables: Source code, explanation, and output screenshots.

**Prompt:**

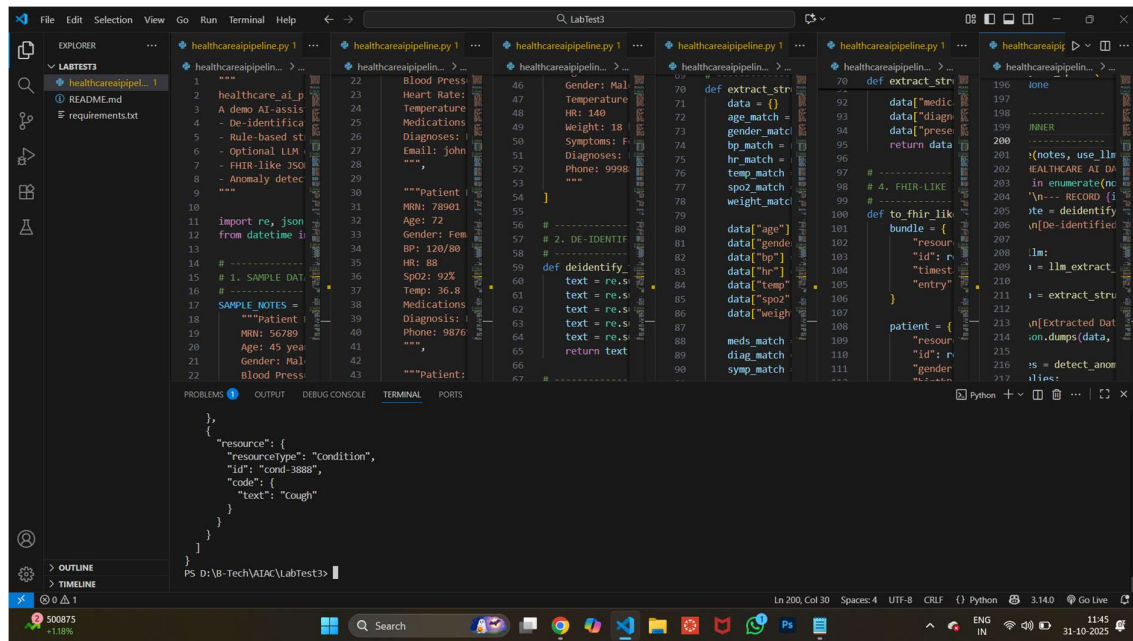
Generate a code and sample data to execute the task and related files and structure.

### Code and Output:

The screenshot displays a Windows desktop with a VS Code editor open. The editor has multiple tabs for a file named 'healthcareapi.pipeline.py'. The main window shows a Python script with the following content:

```

def extract_str:
    196     zone
    197
    198     data["medic
    199     data["diagn
    200     data["prese
    201     return data
    202
    203     #notes, use ille
    204     HEALTHCARE AT DP
    205     in enumerate(nc
    206     "\n-- RECORD i
    207     ste = identifi
    208     (De-identified
    209
    210     bundle = {
    211         "resourc
    212         "id": r
    213         "timest
    214         "entry"
    215     }
    216
    217     patient = {
    218         "resourc
    219         "id": r
    220         "gender
    221         "name": n
    222         "age": a
    223         "sex": g
    224         "bp": bp
    225         "hr": hr
    226         "spo2": spo2
    227         "temp": temp
    228         "weight": weight
    229         "meds": meds_match
    230         "diag": diag_match
    231         "symp": symp_match
    232
    233     return text
    234
    235 # 2. DE-IDENTIFY
    236
    237 def identify:
    238     text = re.s
    239     text = re.s
    240     text = re.s
    241     text = re.s
    242     return text
    243
    244 #
    245
    246 # 1. SAMPLE DATA
    247
    248 # SAMPLE NOTES =
    249
    250 """Patient
    251     MRN: 567890
    252     Age: 45 years
    253     Gender: Male
    254     BP: 120/80
    255     HR: 88
    256     SpO2: 92%
    257     Temp: 36.8 C
    258     Medications: Aspirin, Lisinopril
    259     Diagnosis: Hypertension, Hyperlipidemia
    260     Email: john.doe@hospital.com
    261
    262 """
    263
    264 # Patient:
    265
    266 Blood Press
    267
    268 Heart Rate
    269
    270 Temperature
    271
    272 Medications
    273
    274 Diagnoses: F
    275
    276 Email: john
    277
    278
    279
    280
    281
    282
    283
    284
    285
    286
    287
    288
    289
    290
    291
    292
    293
    294
    295
    296
    297
    298
    299
    300
    301
    302
    303
    304
    305
    306
    307
    308
    309
    310
    311
    312
    313
    314
    315
    316
    317
    318
    319
    320
    321
    322
    323
    324
    325
    326
    327
    328
    329
    330
    331
    332
    333
    334
    335
    336
    337
    338
    339
    340
    341
    342
    343
    344
    345
    346
    347
    348
    349
    350
    351
    352
    353
    354
    355
    356
    357
    358
    359
    360
    361
    362
    363
    364
    365
    366
    367
    368
    369
    370
    371
    372
    373
    374
    375
    376
    377
    378
    379
    380
    381
    382
    383
    384
    385
    386
    387
    388
    389
    390
    391
    392
    393
    394
    395
    396
    397
    398
    399
    400
    401
    402
    403
    404
    405
    406
    407
    408
    409
    410
    411
    412
    413
    414
    415
    416
    417
    418
    419
    420
    421
    422
    423
    424
    425
    426
    427
    428
    429
    430
    431
    432
    433
    434
    435
    436
    437
    438
    439
    440
    441
    442
    443
    444
    445
    446
    447
    448
    449
    450
    451
    452
    453
    454
    455
    456
    457
    458
    459
    460
    461
    462
    463
    464
    465
    466
    467
    468
    469
    470
    471
    472
    473
    474
    475
    476
    477
    478
    479
    480
    481
    482
    483
    484
    485
    486
    487
    488
    489
    490
    491
    492
    493
    494
    495
    496
    497
    498
    499
    500
    501
    502
    503
    504
    505
    506
    507
    508
    509
    510
    511
    512
    513
    514
    515
    516
    517
    518
    519
    520
    521
    522
    523
    524
    525
    526
    527
    528
    529
    530
    531
    532
    533
    534
    535
    536
    537
    538
    539
    540
    541
    542
    543
    544
    545
    546
    547
    548
    549
    550
    551
    552
    553
    554
    555
    556
    557
    558
    559
    560
    561
    562
    563
    564
    565
    566
    567
    568
    569
    570
    571
    572
    573
    574
    575
    576
    577
    578
    579
    580
    581
    582
    583
    584
    585
    586
    587
    588
    589
    590
    591
    592
    593
    594
    595
    596
    597
    598
    599
    600
    601
    602
    603
    604
    605
    606
    607
    608
    609
    610
    611
    612
    613
    614
    615
    616
    617
    618
    619
    620
    621
    622
    623
    624
    625
    626
    627
    628
    629
    630
    631
    632
    633
    634
    635
    636
    637
    638
    639
    640
    641
    642
    643
    644
    645
    646
    647
    648
    649
    650
    651
    652
    653
    654
    655
    656
    657
    658
    659
    660
    661
    662
    663
    664
    665
    666
    667
    668
    669
    670
    671
    672
    673
    674
    675
    676
    677
    678
    679
    680
    681
    682
    683
    684
    685
    686
    687
    688
    689
    690
    691
    692
    693
    694
    695
    696
    697
    698
    699
    700
    701
    702
    703
    704
    705
    706
    707
    708
    709
    710
    711
    712
    713
    714
    715
    716
    717
    718
    719
    720
    721
    722
    723
    724
    725
    726
    727
    728
    729
    730
    731
    732
    733
    734
    735
    736
    737
    738
    739
    740
    741
    742
    743
    744
    745
    746
    747
    748
    749
    750
    751
    752
    753
    754
    755
    756
    757
    758
    759
    760
    761
    762
    763
    764
    765
    766
    767
    768
    769
    770
    771
    772
    773
    774
    775
    776
    777
    778
    779
    780
    781
    782
    783
    784
    785
    786
    787
    788
    78
```



## Observation:

In operations made in the above code:

- Extracted raw health data
- Cleaned the data(missing values)
- Converted everything into FHIR like JSON format
- Added unique Id's and timestamps

## Task2:

Scenario: In the domain of Environmental Monitoring, a company is facing a challenge related to data structures with ai.

Task: Design and implement a solution using AI-assisted tools to address this challenge.

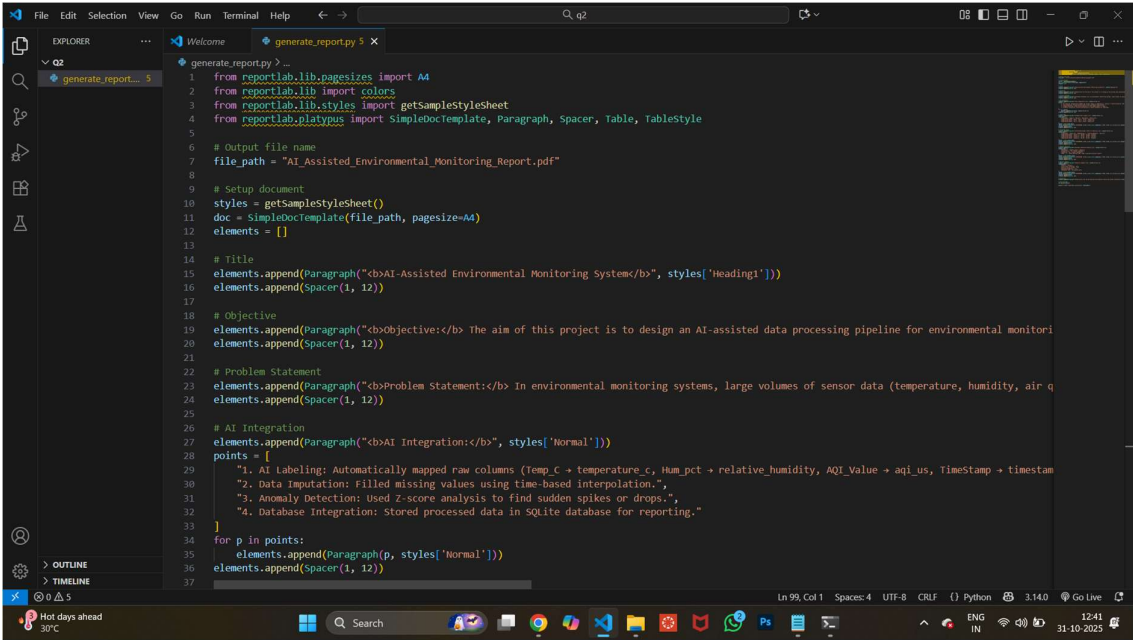
Include code, explanation of AI integration, and test results.

Deliverables: Source code, explanation, and output screenshots.

## Prompt:

Create a CSV file containing some data related to the task and explain the code and summary of the code and give me the structure and file names.

Code:



```
1 from reportlab.lib.pagesizes import A4
2 from reportlab.lib import colors
3 from reportlab.lib.styles import getSampleStyleSheet
4 from reportlab.platypus import SimpleDocTemplate, Paragraph, Spacer, Table, TableStyle
5
6 # Output file name
7 file_path = "AI_Assisted_Environmental_Monitoring_Report.pdf"
8
9 # Setup document
10 styles = getSampleStyleSheet()
11 doc = SimpleDocTemplate(file_path, pagesize=A4)
12 elements = []
13
14 # Title
15 elements.append(Paragraph("<b>AI-Assisted Environmental Monitoring System</b>", styles['Heading1']))
16 elements.append(Spacer(1, 12))
17
18 # Objective
19 elements.append(Paragraph("<b>Objective:</b> The aim of this project is to design an AI-assisted data processing pipeline for environmental monitoring systems.")
20 elements.append(Spacer(1, 12))
21
22 # Problem Statement
23 elements.append(Paragraph("<b>Problem Statement:</b> In environmental monitoring systems, large volumes of sensor data (temperature, humidity, air quality, etc.) are collected over time. The challenge is to process this data efficiently, identify anomalies, and generate actionable insights. This report outlines the design of an AI-assisted pipeline to address these challenges.")
24 elements.append(Spacer(1, 12))
25
26 # AI Integration
27 elements.append(Paragraph("<b>AI Integration:</b>", styles['Normal']))
28 points = [
29     "1. AI Labeling: Automatically mapped raw columns (temp_c + temperature_c, hum_pct + relative_humidity, aqi_value + aqi_us, timestamp + timestamp).",
30     "2. Data Imputation: Filled missing values using time-based interpolation.",
31     "3. Anomaly Detection: Used Z-score analysis to find sudden spikes or drops.",
32     "4. Database Integration: Stored processed data in SQLite database for reporting."
33 ]
34 for p in points:
35     elements.append(Paragraph(p, styles['Normal']))
36 elements.append(Spacer(1, 12))
37
```

Output:

Sample Data (Input)

TimeStamp	Temp_C	Hum_pct	AQI_Value	sensor_id
2025-10-01 00:00	20.0	45.2	29.8	sensor_1
2025-10-01 01:00	20.6	49.1	31.0	sensor_1
2025-10-01 02:00	21.4	49.4	25.9	sensor_1

Processed Output (After AI Pipeline)

timestamp	sensor_id	temperature_c	relative_humidity	aqi_us
2025-10-01 00:00	sensor_1	20.00	45.25	29.82
2025-10-01 01:00	sensor_1	20.62	49.09	31.00
2025-10-01 02:00	sensor_1	21.45	49.46	25.99
2025-10-01 03:00	sensor_1	24.29	49.57	35.06

**Observation/Conclusion:**

The AI-assisted environmental monitoring system successfully handled unstructured sensor data by automatically labeling columns, cleaning missing values, and detecting anomalies.

This solution improves data reliability, supports real-time decision-making, and enhances environmental awareness using Artificial Intelligence.