

ASSIGNMENT 4 –IMPORTANT SCREENSHOTS:

SonarQube Cloud Analysis (before):

The screenshot shows the SonarQube Cloud interface for the project 'FlightBookingApplication'. The 'Summary' tab is active, displaying various quality metrics. A message at the top indicates that the quality gate status will be updated after the next analysis. The metrics are as follows:

Metric	Value	Status
Security	0 Open Issues	Pass (A)
Reliability	0 Open Issues	Pass (A)
Maintainability	22 Open Issues	Pass (A)
Accepted Issues	0	Pass (A)
Coverage	20.1%	Fail (C)
Duplications	0.0%	Pass (A)
Security Hotspots	1	Pass (A)

The 'Coverage' metric is highlighted with a red circle, indicating a failure. The 'Define New Code' button is visible in the top right of the summary section.

SonarQube Cloud Analysis (After):

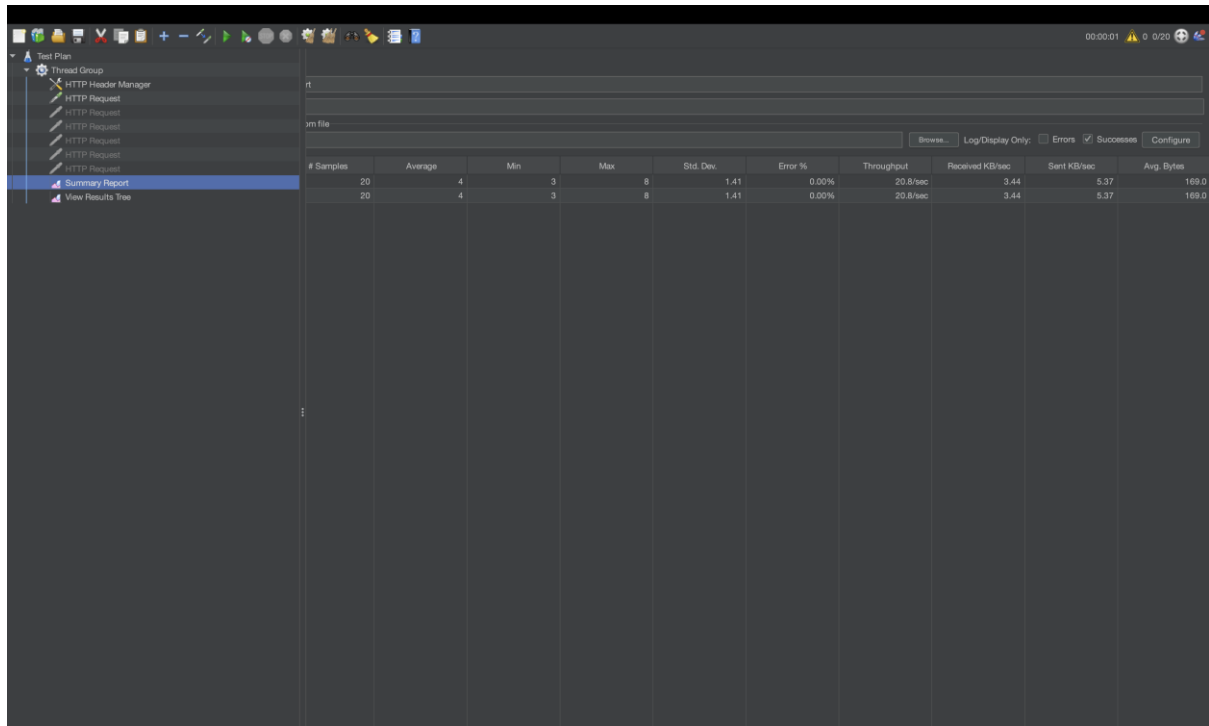
The screenshot shows the SonarQube Cloud interface for the project 'FlightBookingApplication' after a new code analysis. The 'Summary' tab is active, displaying the 'Main Branch Summary'. The quality gate status is 'Passed' (green checkmark). The metrics are as follows:

Metric	Value	Status
New Issues	0	Pass (A)
Accepted Issues	0	Pass (A)
Coverage	97.42%	Pass (A)
Duplications	0.0%	Pass (A)
Security Hotspots	0	Pass (A)

The 'Coverage' metric is highlighted with a green circle, indicating a pass. The 'New Code' tab is selected, showing the 'Overall Code' summary. The 'Quality Gate: Sonar way' is also indicated as passed.

JMETER Output for 20 requests:

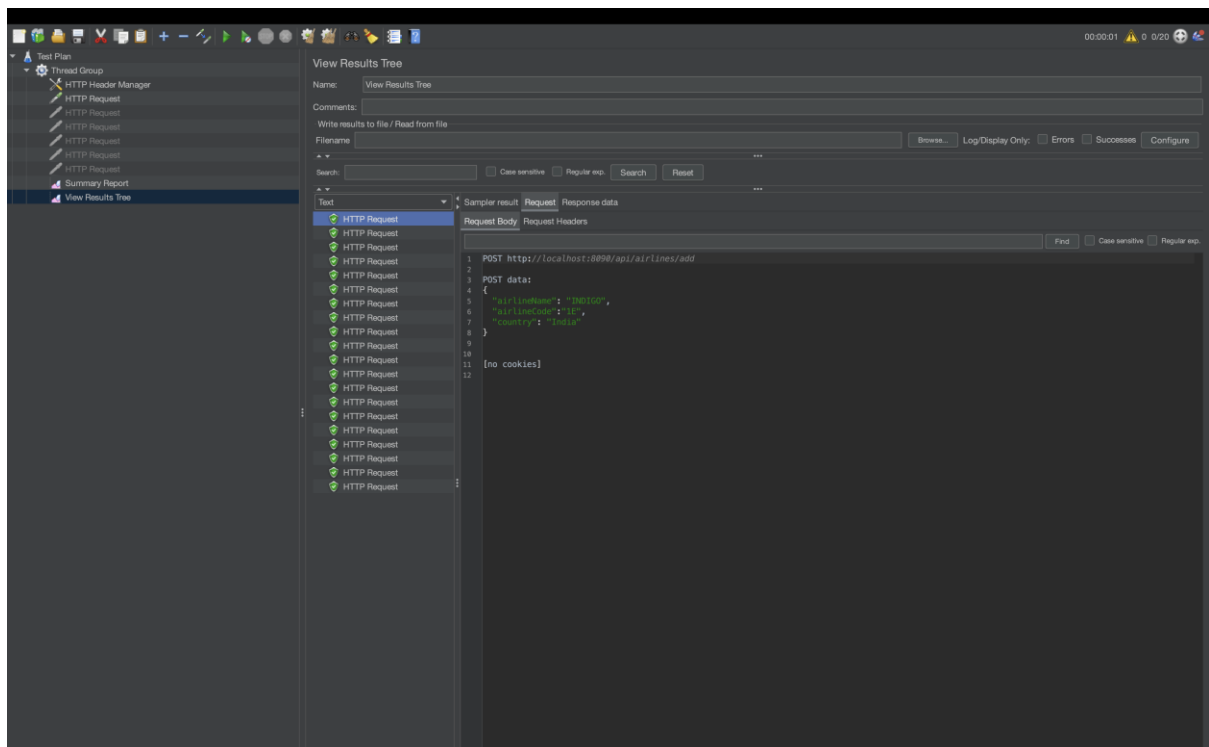
Summary-report:



The screenshot shows the JMeter Summary Report window. The left sidebar lists the test plan components: Thread Group, HTTP Header Manager, and 20 HTTP Request elements. The 'Summary Report' is selected. The main area displays a table with performance metrics for the 20 requests.

# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
20	4	3	8	1.41	0.00%	20.8/sec	3.44	5.37	169.0
20	4	3	8	1.41	0.00%	20.8/sec	3.44	5.37	169.0

View results tree:

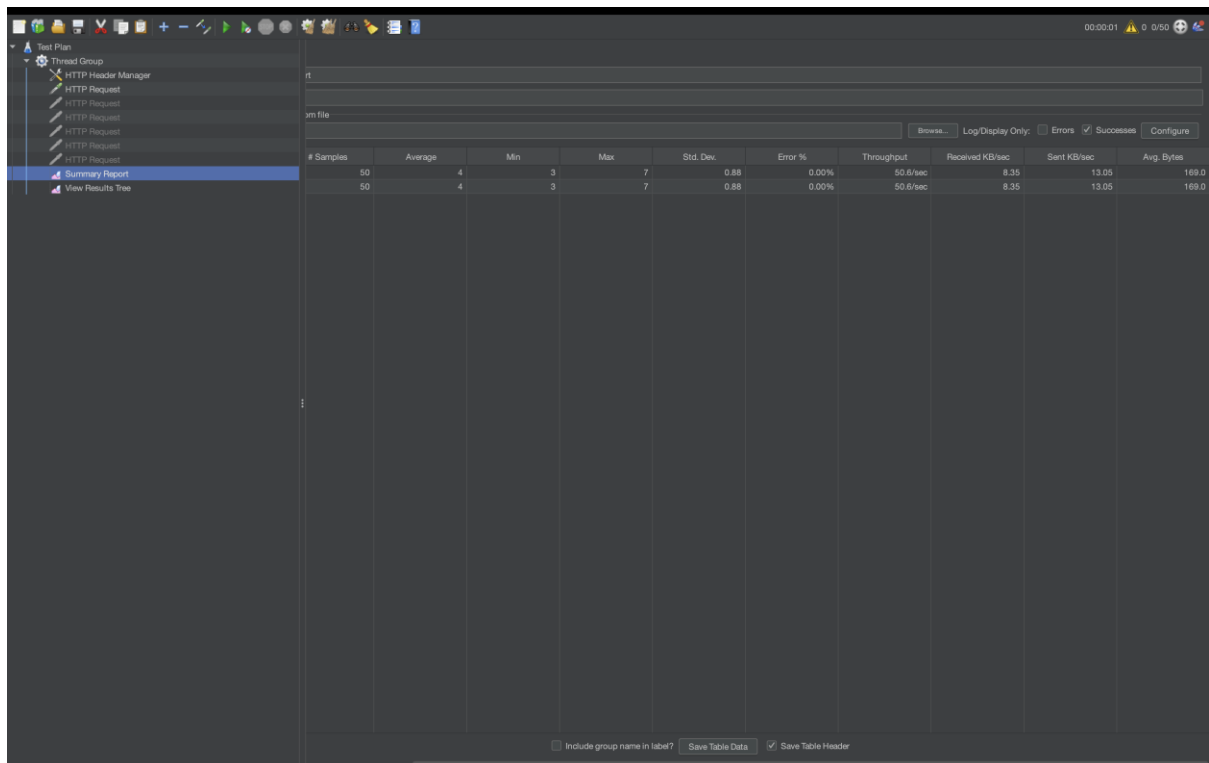


The screenshot shows the JMeter View Results Tree window. The left sidebar lists the test plan components, with 'View Results Tree' selected. The main area displays the details of the first HTTP request, including the request body and response data.

Text	Sampler result	Request	Response data
1 POST http://localhost:8080/api/airlines/add			
2			
3 POST data:			
4 {			
5 "airlineName": "INDIGO",			
6 "airlineCode": "IE",			
7 "country": "India"			
8 }			
9			
10			
11 [no cookies]			
12			

JMETER Output for 50 requests:

Summary-report:

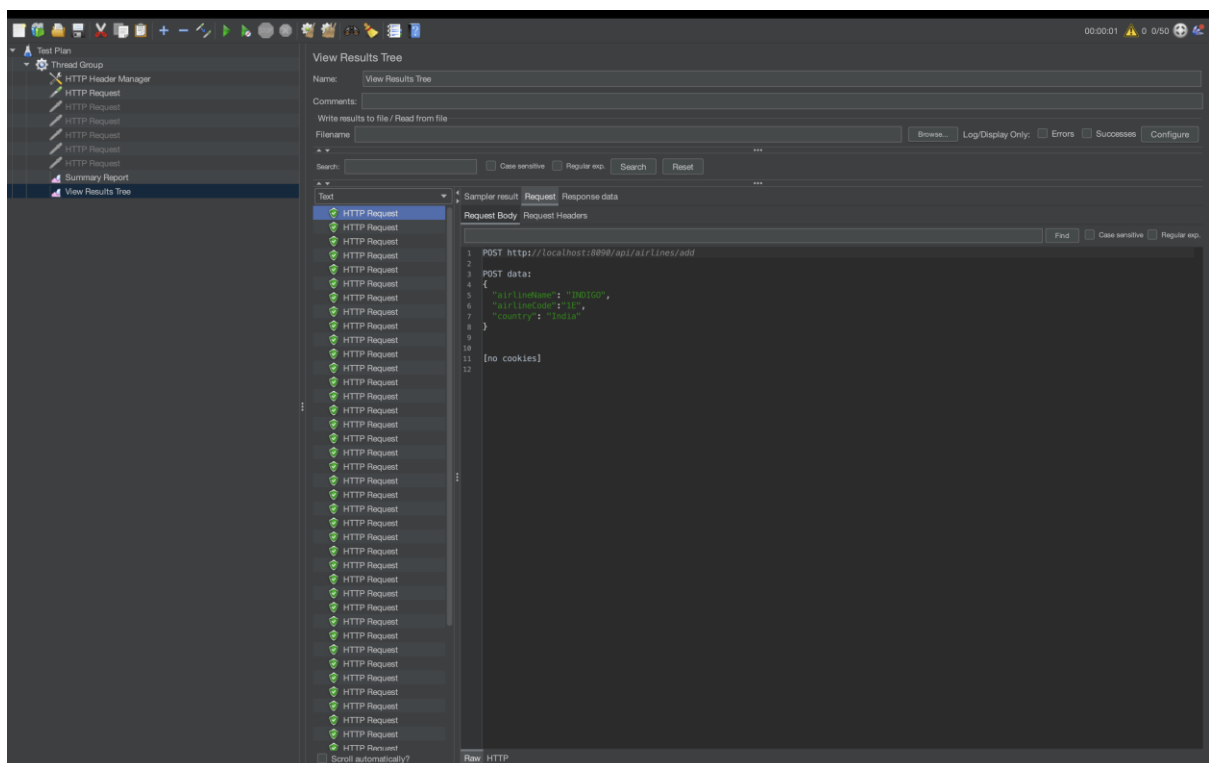


The screenshot shows the Apache JMeter Summary Report. The left sidebar lists the test plan components: Thread Group, HTTP Header Manager, and ten HTTP Request samplers. The 'Summary Report' is selected. The main area displays a table with performance metrics for the selected sampler.

# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
50	4	3	7	0.88	0.00%	50.6/sec	8.35	13.05	169.0
50	4	3	7	0.88	0.00%	50.6/sec	8.35	13.05	169.0

At the bottom, there are checkboxes for 'Include group name in label?' (unchecked), 'Save Table Data' (button), and 'Save Table Header' (checked).

View results tree:



The screenshot shows the Apache JMeter View Results Tree. The left sidebar shows the 'View Results Tree' selected. The main area displays the details of the first HTTP Request sampler.

Name: View Results Tree
Comments:
 Write results to file / Read from file
 Filename: ☐ Log/Display Only: ☐ Errors ☐ Successes

Search: ☐ Case sensitive ☐ Regular exp.

Test: ☐ HTTP Request

Sampler result: ☐ Request ☐ Response data

Request Body: ☐ Request Headers ☐ Case sensitive ☐ Regular exp.

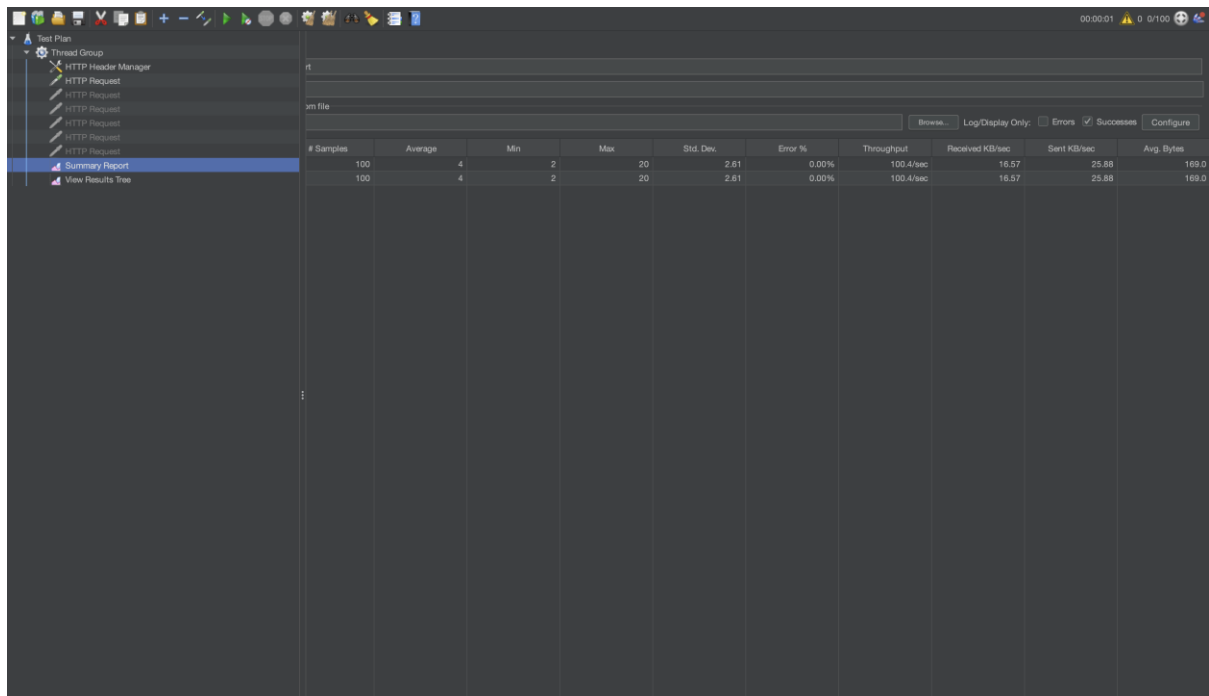
```

1 POST http://localhost:8090/api/airlines/add
2
3 POST data:
4 {
5   "airlineName": "Indigo",
6   "airlineCode": "IG",
7   "country": "India"
8 }
9
10 [no cookies]
11
12
  
```

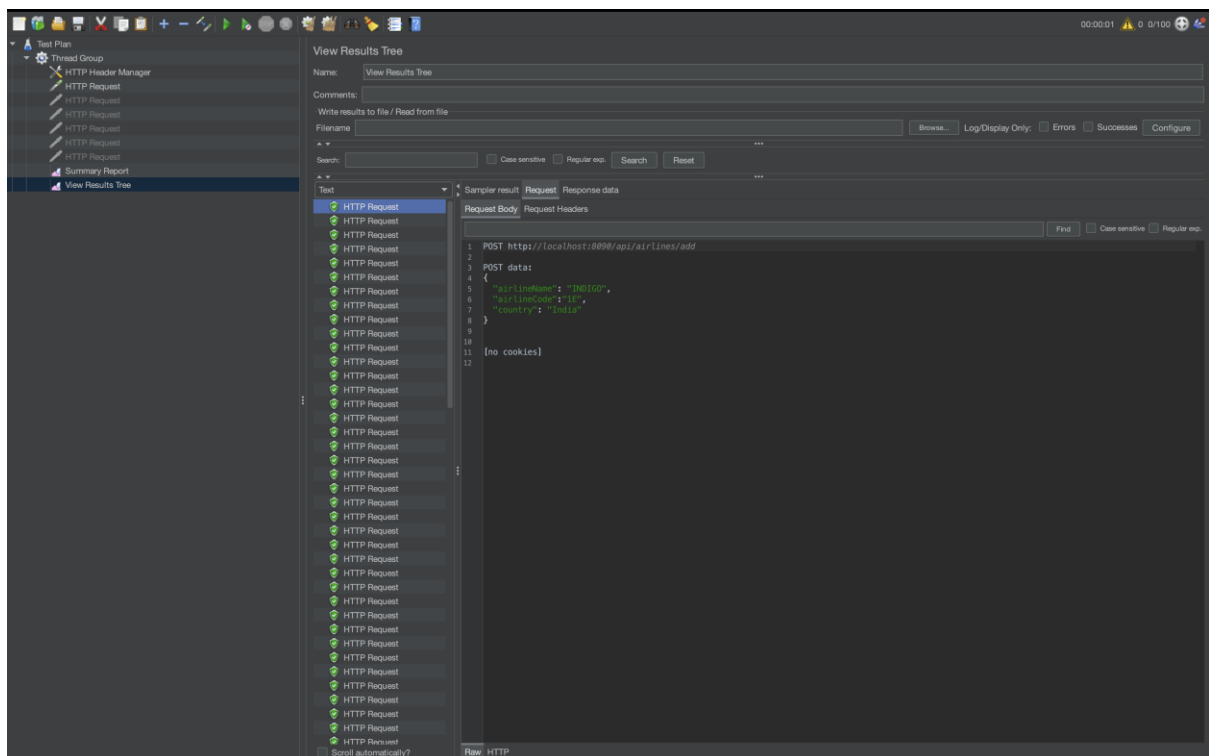
Raw: HTTP

JMETER Output for 100 requests:

Summary-report:



View results tree:



MongoDB Connected :

Compass

localhost:27017 flight_booking airlines +

localhost:27017 > flight_booking

Open MongoDB shell Create collection Refresh

Collection name	Properties	Storage size	Documents	Avg. document size	Indexes	Total index size
airlines	-	36.86 kB	2	125.00 B	1	36.86 kB
bookings	-	20.48 kB	1	243.00 B	1	20.48 kB
flights	-	32.77 kB	2	290.00 B	1	32.77 kB
passengers	-	20.48 kB	2	210.00 B	1	20.48 kB
payments	-	20.48 kB	1	180.00 B	1	20.48 kB

CONNECTIONS (1)

Search connections

- localhost:27017
 - admin
 - config
 - flight_booking
 - airlines
 - bookings
 - flights
 - passengers
 - payments
 - local
 - startup_log
 - reactive_db
 - tutorials
 - test
 - student

Working PostMan:

Joshika sree Polaku's Workspace

POST add new webmongo / add airline

POST http://localhost:8090/api/airlines/add

Body

```

1 {
2   "airlineName": "INDIGO",
3   "airlineCode": "6E",
4   "country": "India"
5 }
6

```

201 Created · 15 ms · 169 B

Body

```

1 {
2   "id": "6922f0a9f4239e0557a9301a",
3   "airlineName": "INDIGO",
4   "airlineCode": "6E",
5   "country": "India"
6 }

```

Working SwaggerUI:

