# **JMeter API Testing Report**

Created a test plan, and thread group starting from 10 users to onwards, and checking response each time.

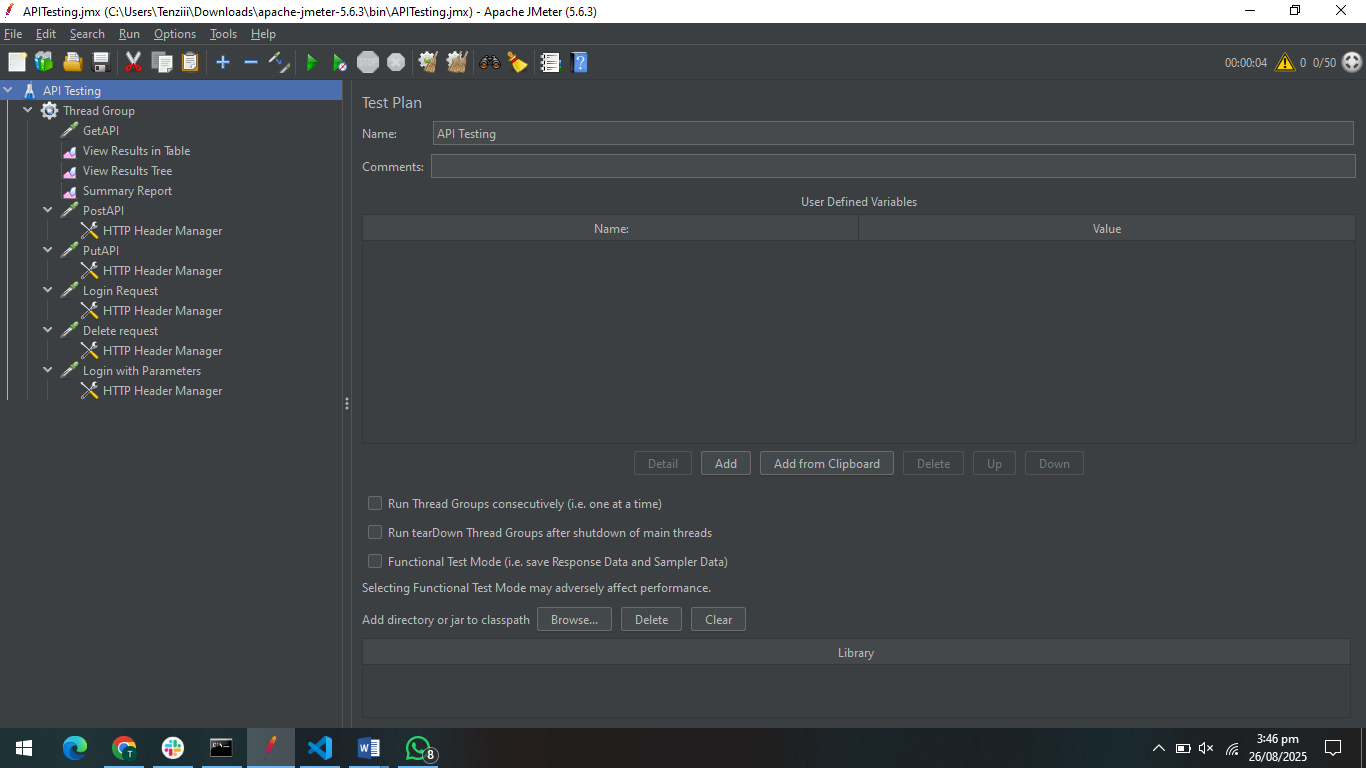
## **Test Setup**

* **Tools**: JMeter
* **Thread Group**: Started with **1 virtual user**, ramped up gradually to 10, 20, 30 … up to 50 users.
* **APIs Tested**: GET, POST, PUT, LOGIN, DELETE.
* **Metrics**: Response time, error percentage, system stability.

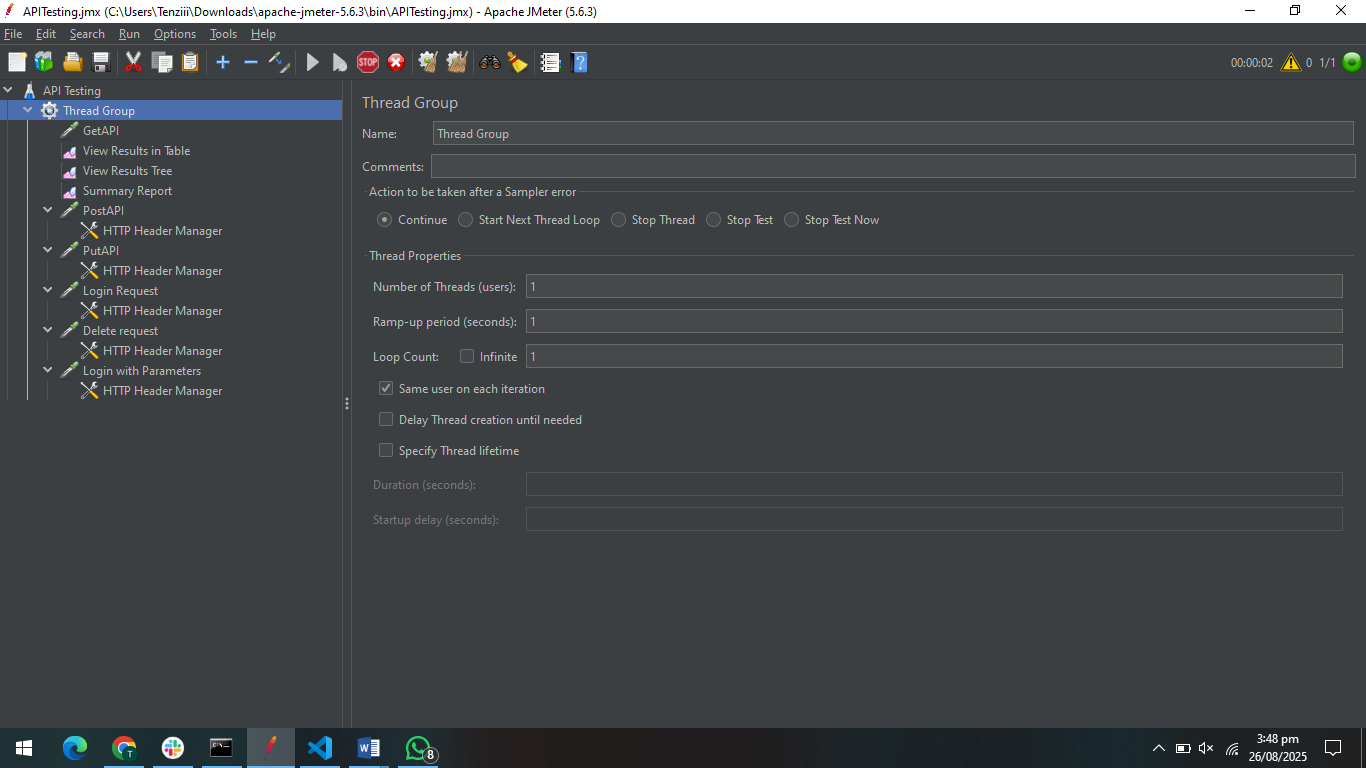
### **Observations**

* **Up to 10 users:** Response stable, low latency (~1204ms max for GET), throughput ~2.7/sec. Error rate around **33%** (slightly high but stable).
* **20-40 users:** Response time acceptable (max 514ms for POST), throughput ~26.9/sec. Error rate increases slightly to **41%**.
* **30-60 users:** Response time increases (max 839ms for GET), throughput ~48.8/sec. Error rate rises sharply to **88%**, indicating instability.
* **50- 100 users:** Response degradation observed (max 730ms for POST), throughput ~71.6/sec. Error rate fluctuates between **55%–90%**, showing system stress.
* **300 users (Ramp-up 1 sec):** High load with 600 requests, max response ~780ms (GET), throughput ~114.6/sec. Error rate very high (**67–95%**) across all APIs.
* **600 users (Ramp-up 1 min):**  Average response stable (~342ms), but max response 1552ms (GET). Error rate consistently high (**84–95%**) despite lower throughput (~9.6/sec).

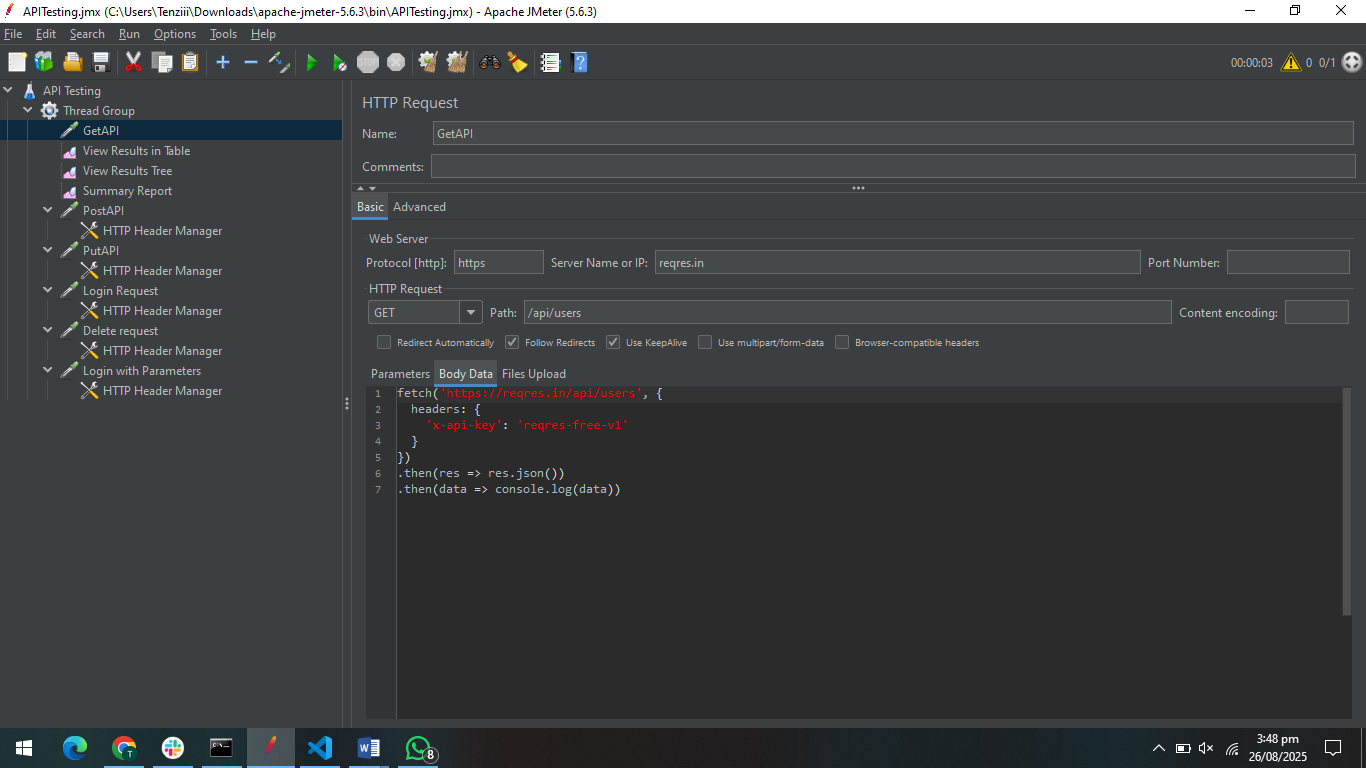
Test Plan



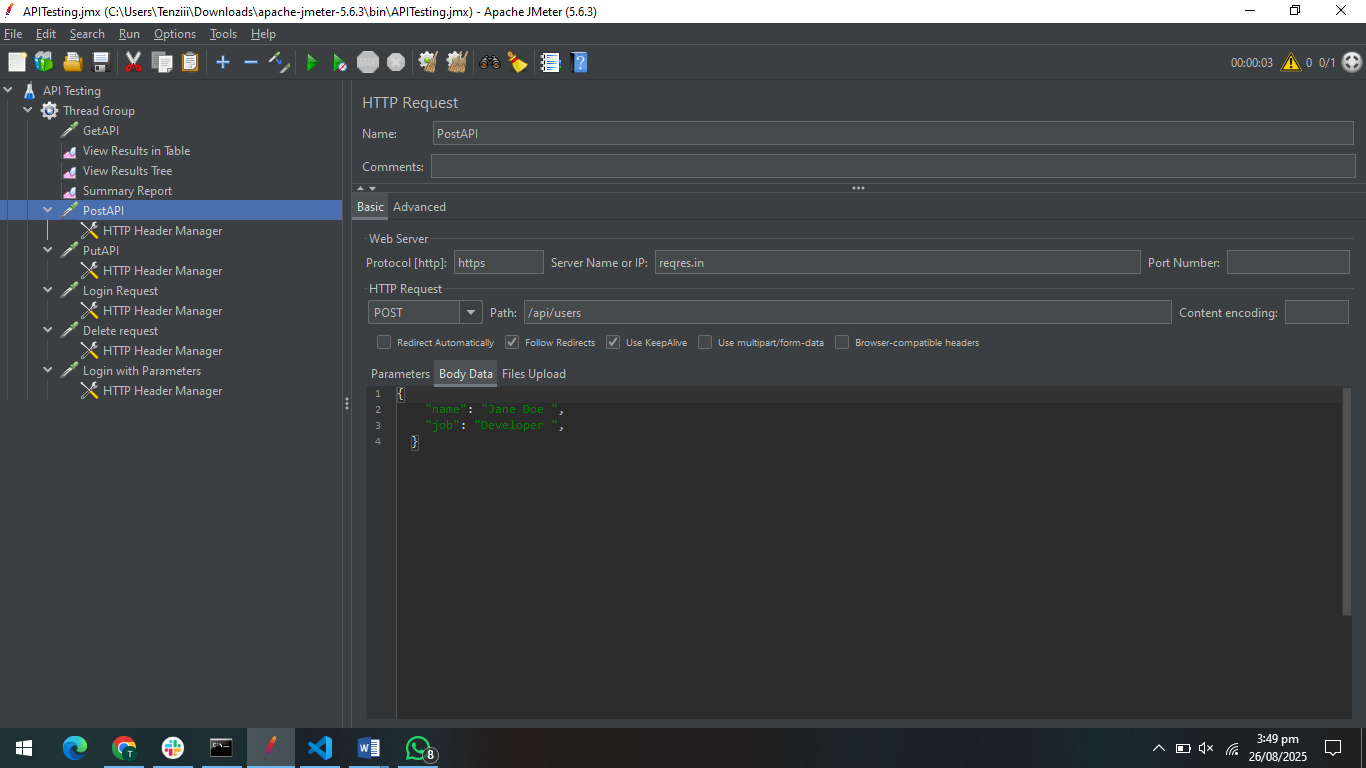
Thread Group



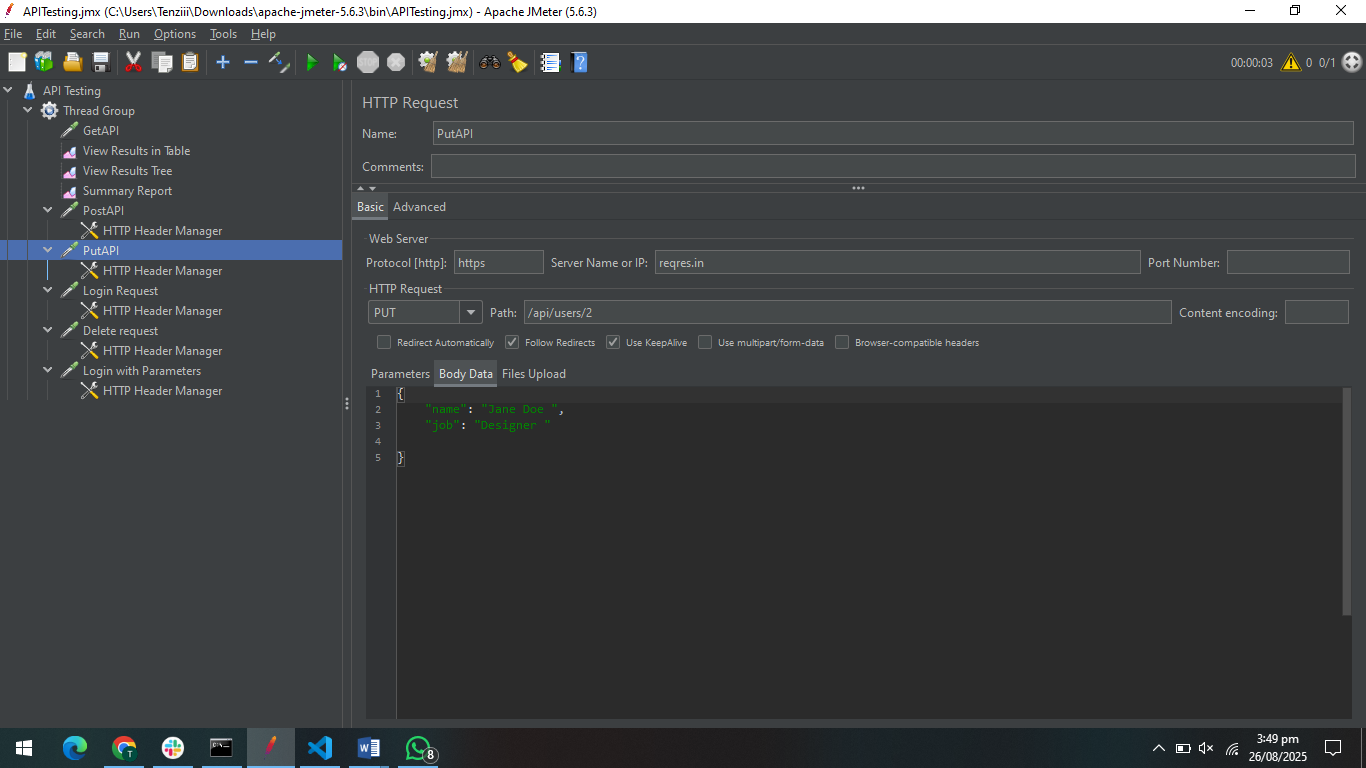
GETAPI



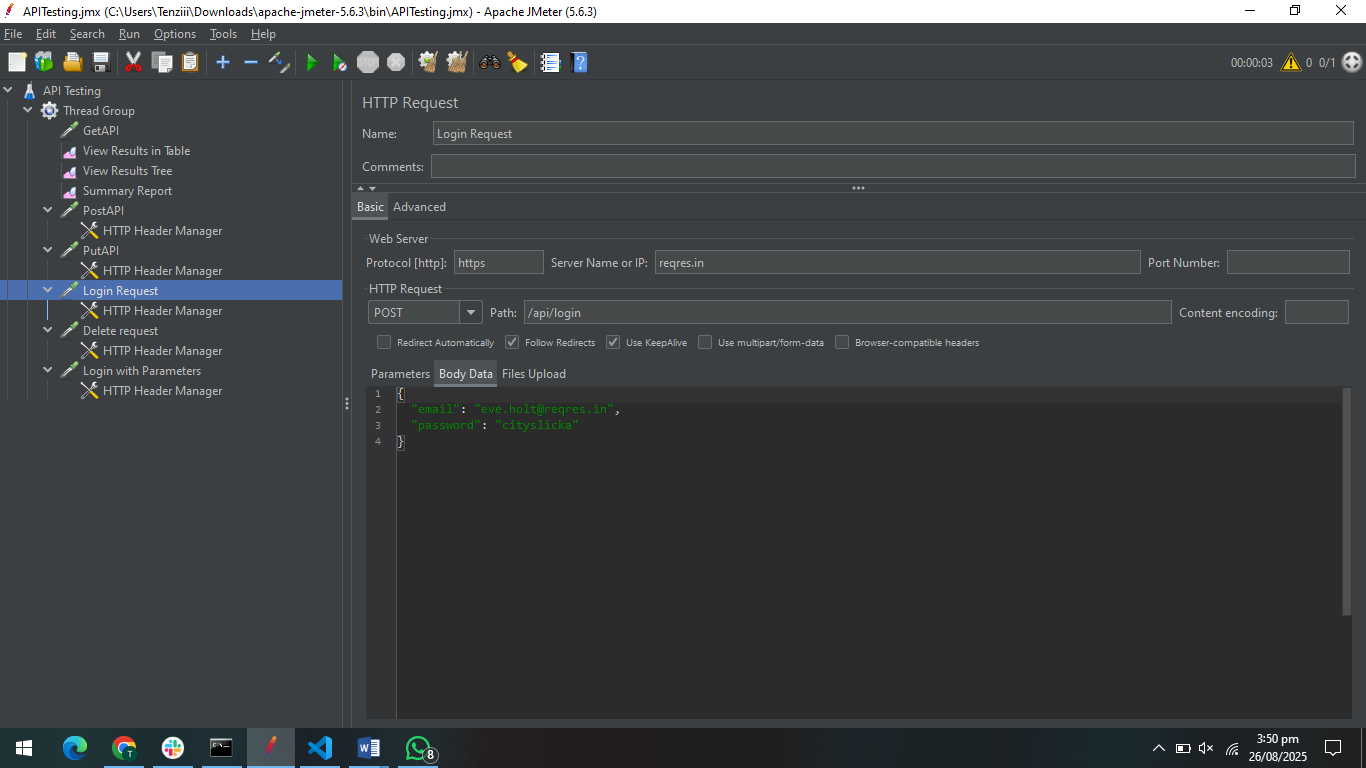
PostAPI



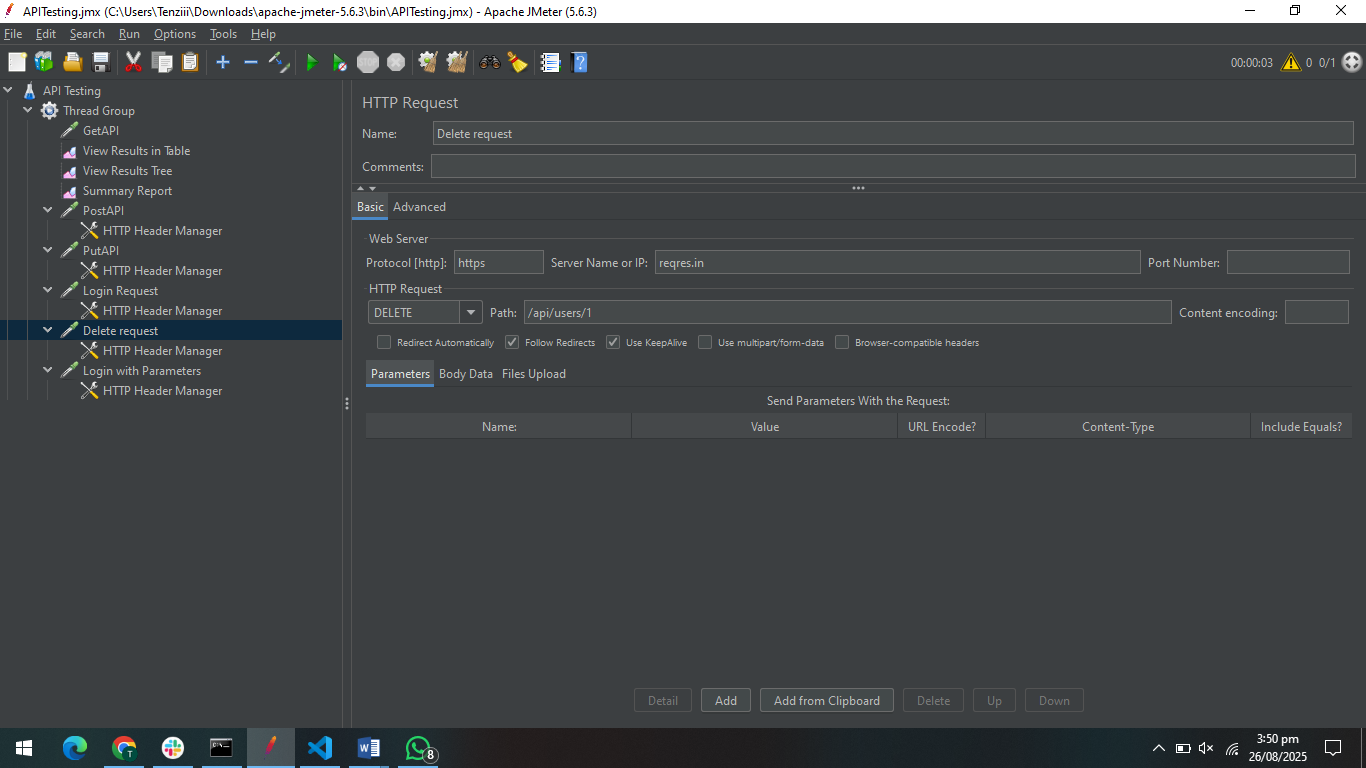
PutAPI



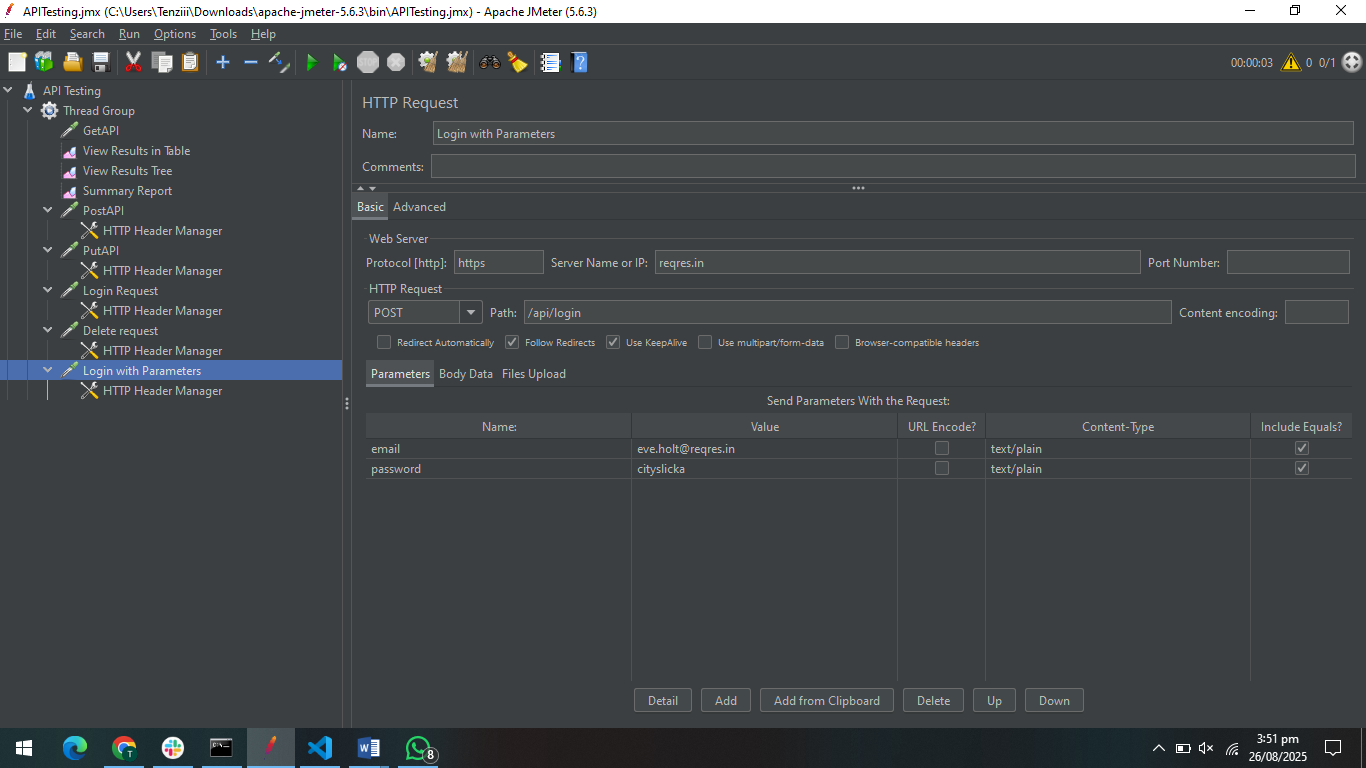
Login API



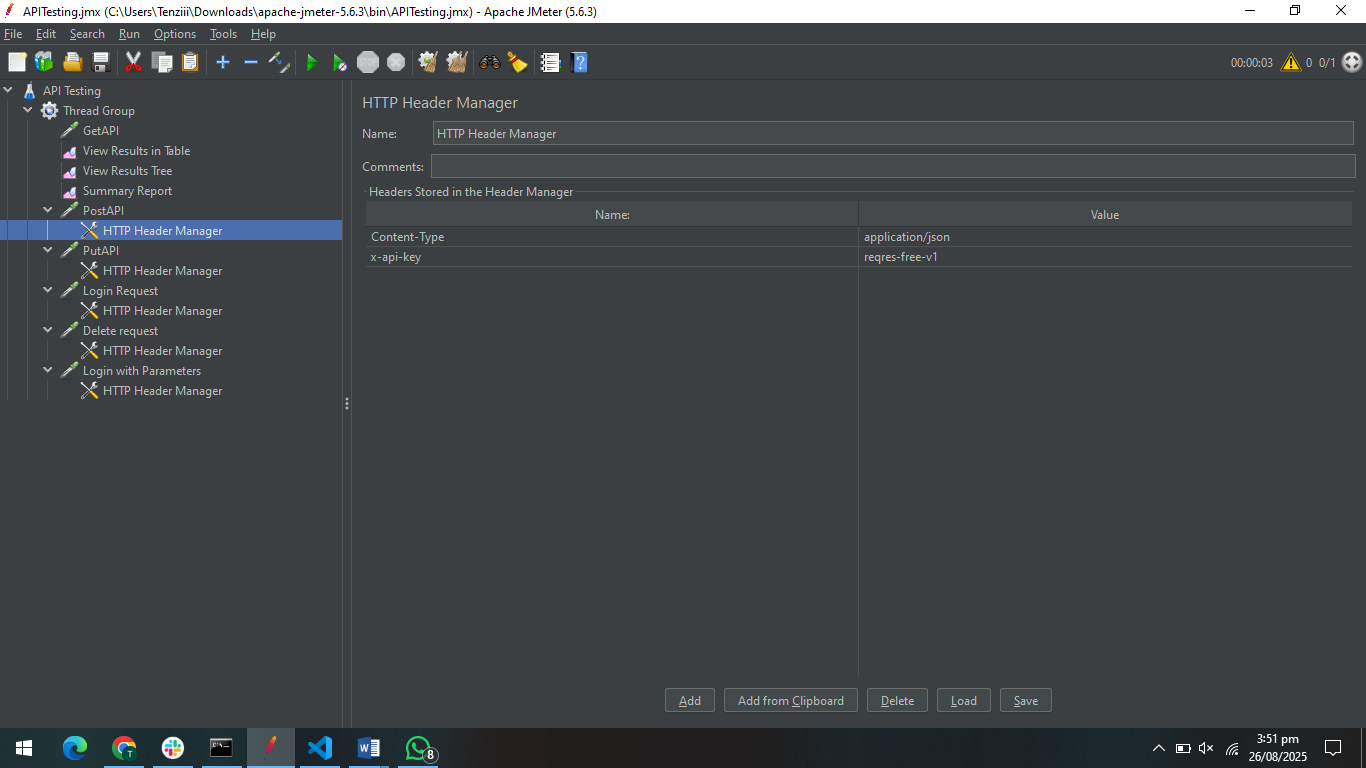
DeleteAPI



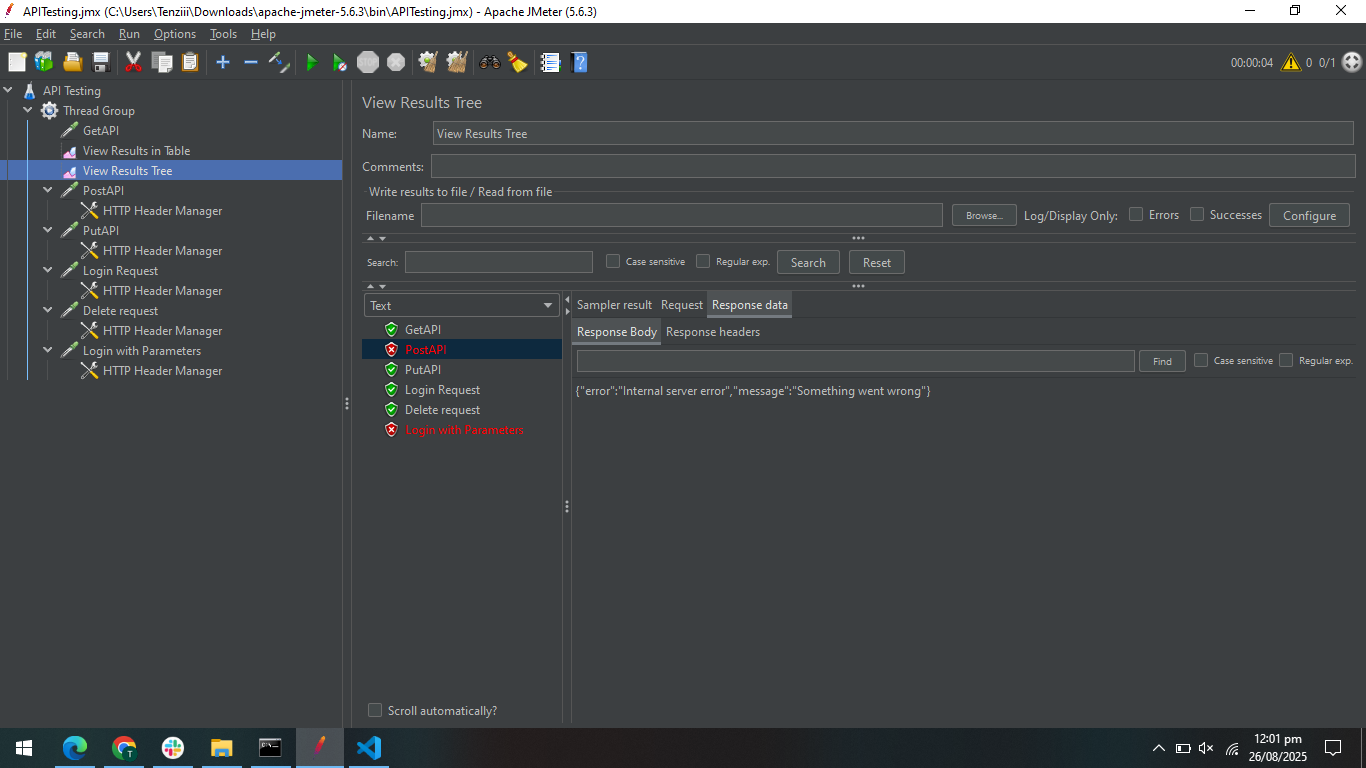
Login with paraemters

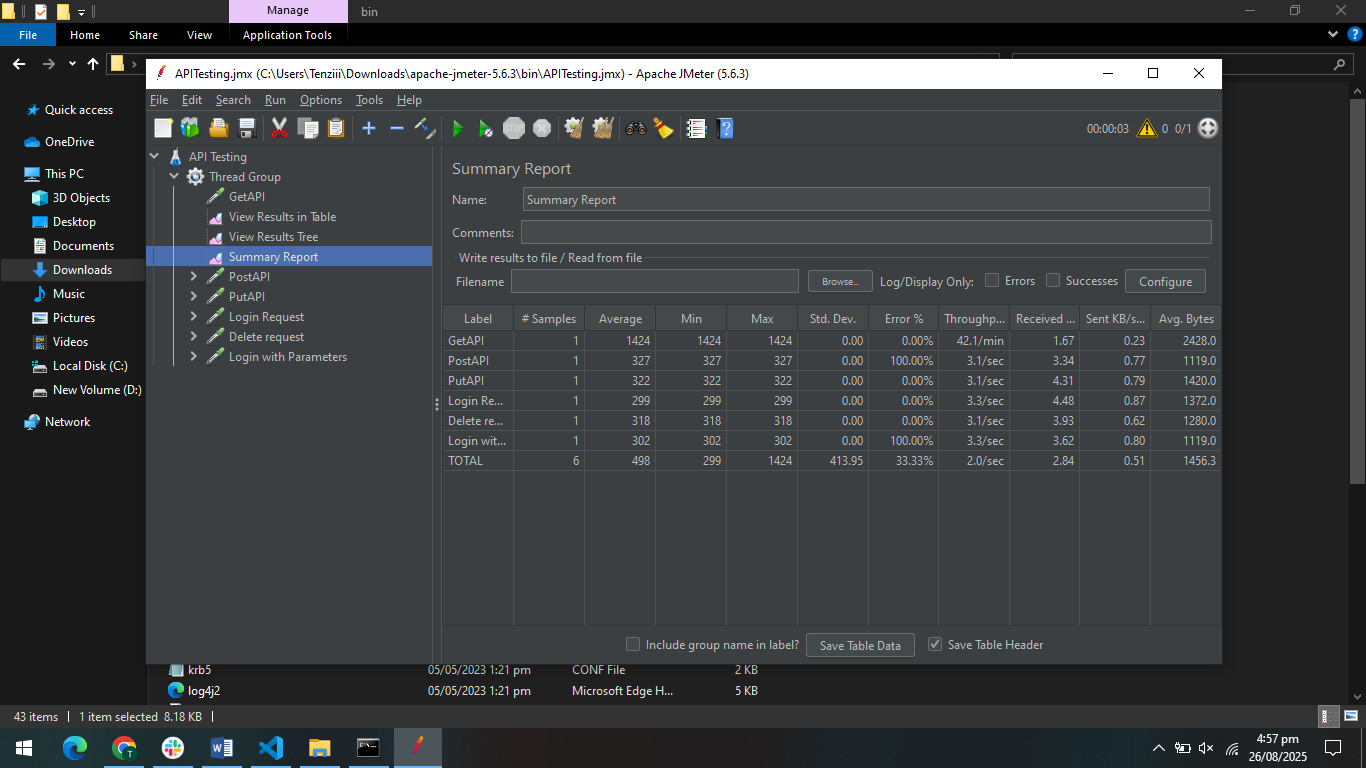


Header Manager



No of users: 10

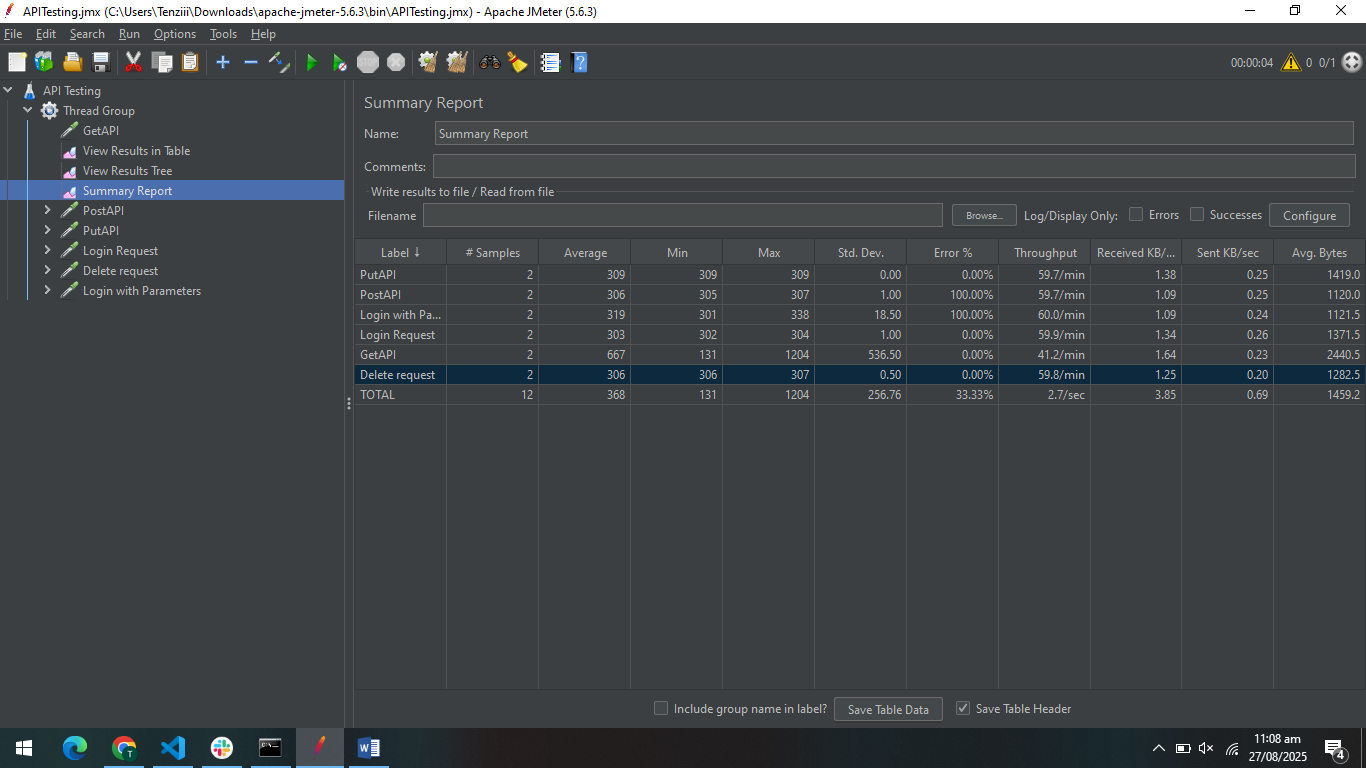




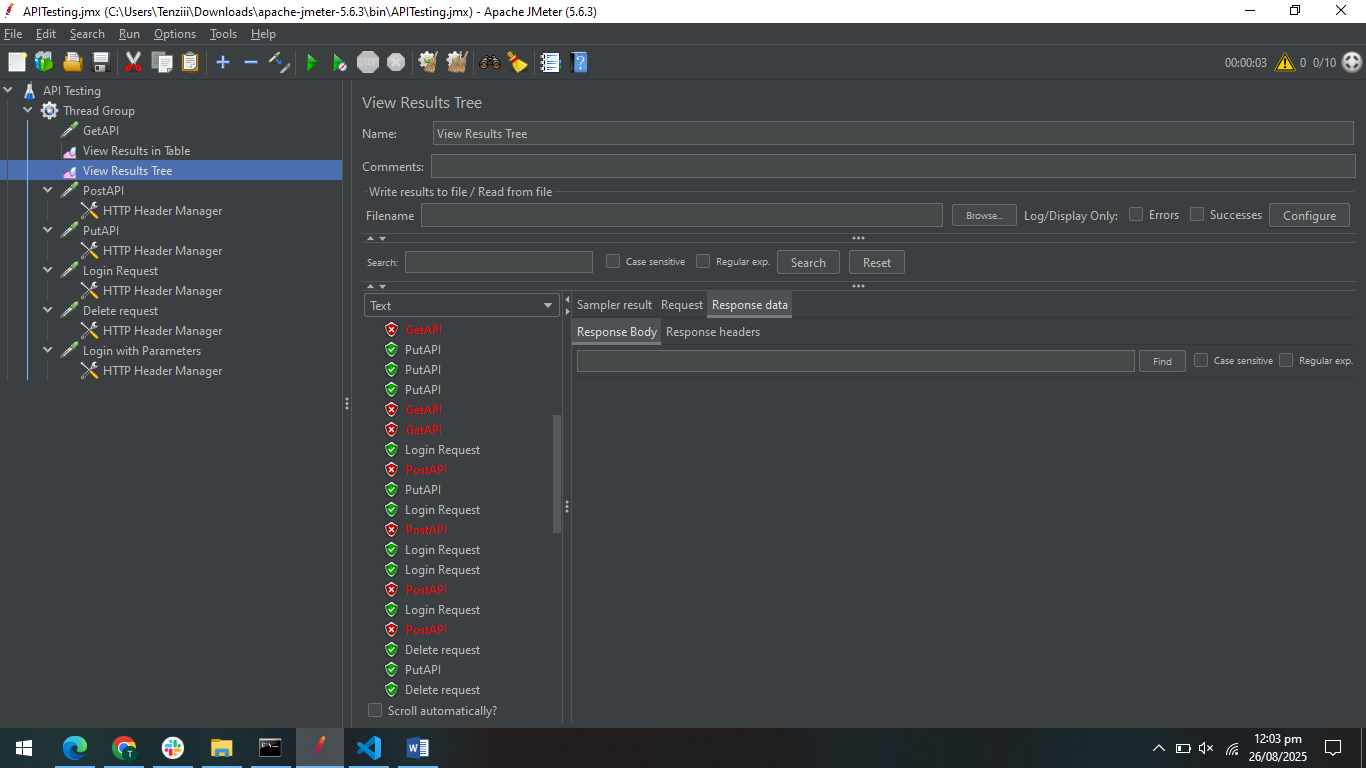
Loop Count 2

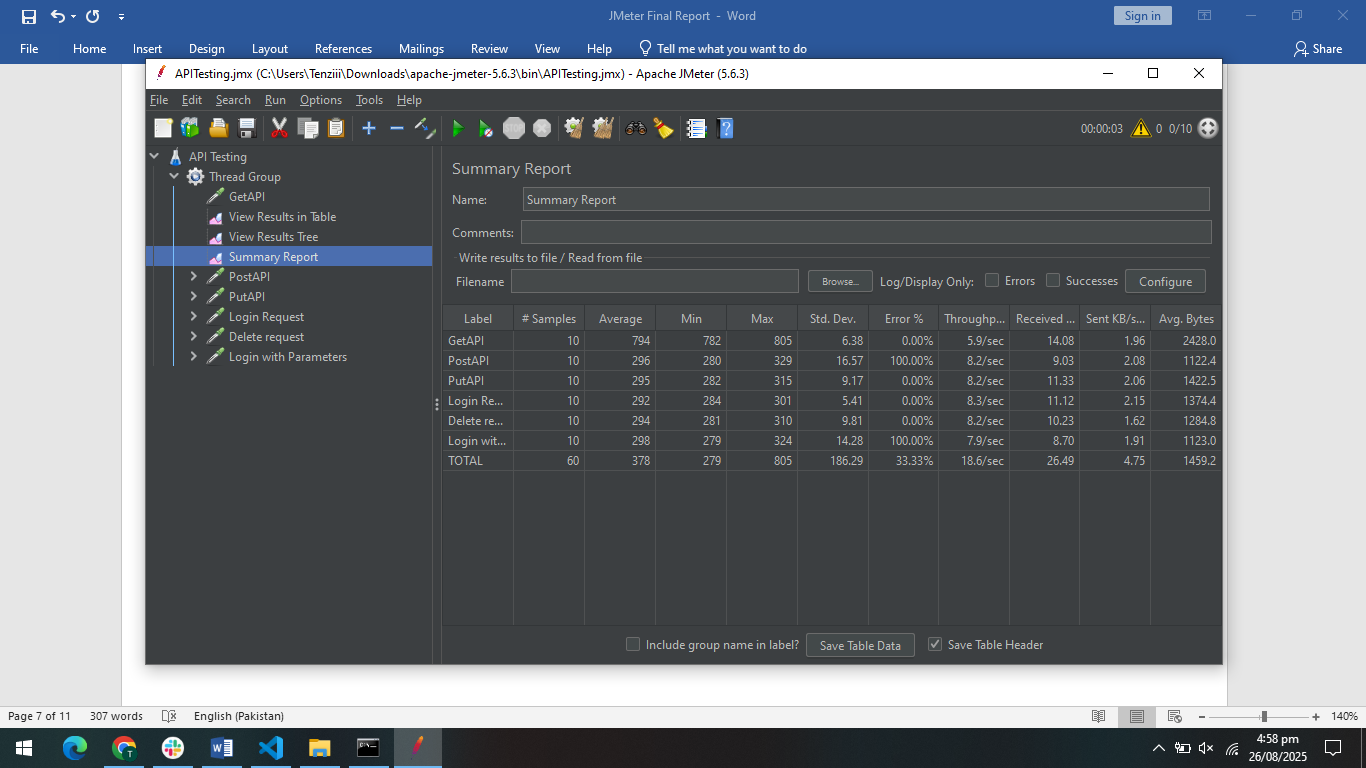
The test executed 1 requests across six APIs in ramp-up period 1 sec . While average response times remained within acceptable limits ,but highest response time was 1204ms for get API, Throughput remained steady at ~2.7/sec.

Error Percentage for overall process remains same ~33%, but for some API it increases



No of users: 10

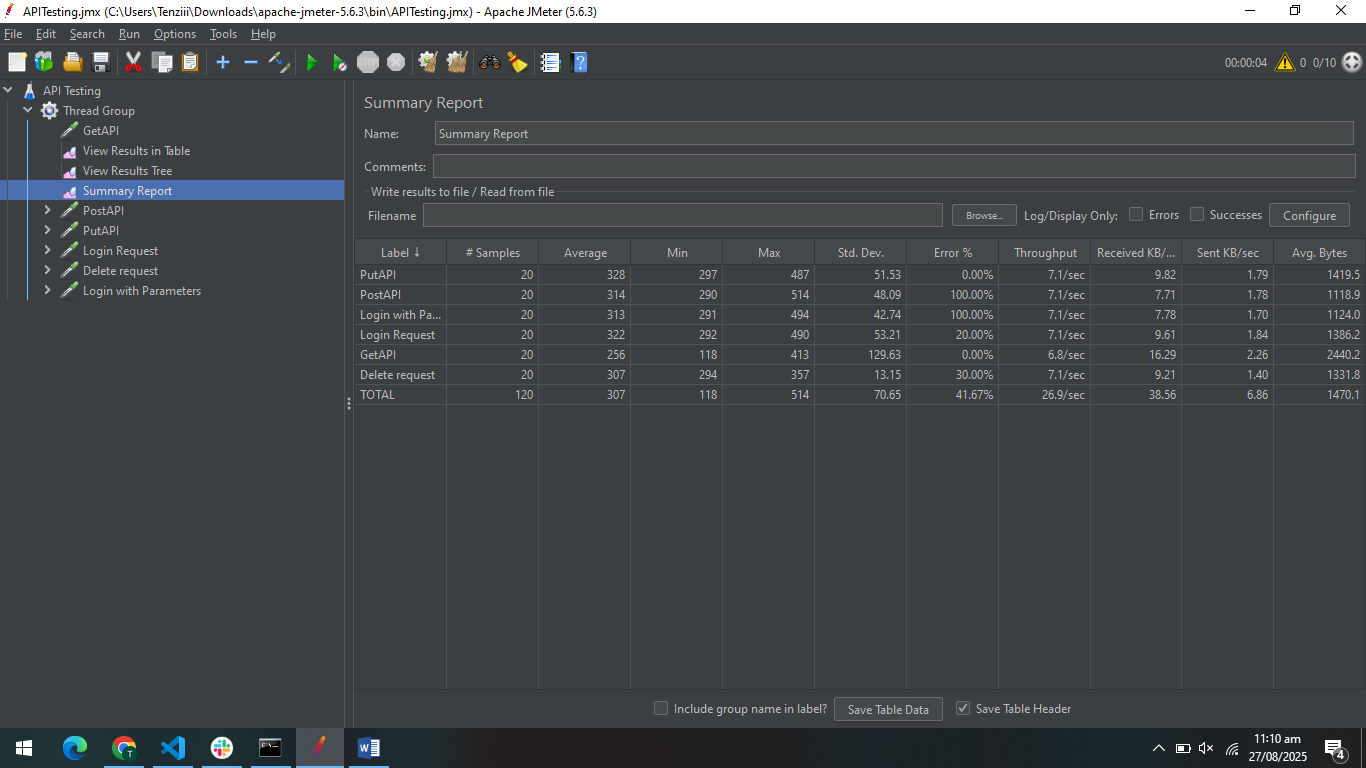




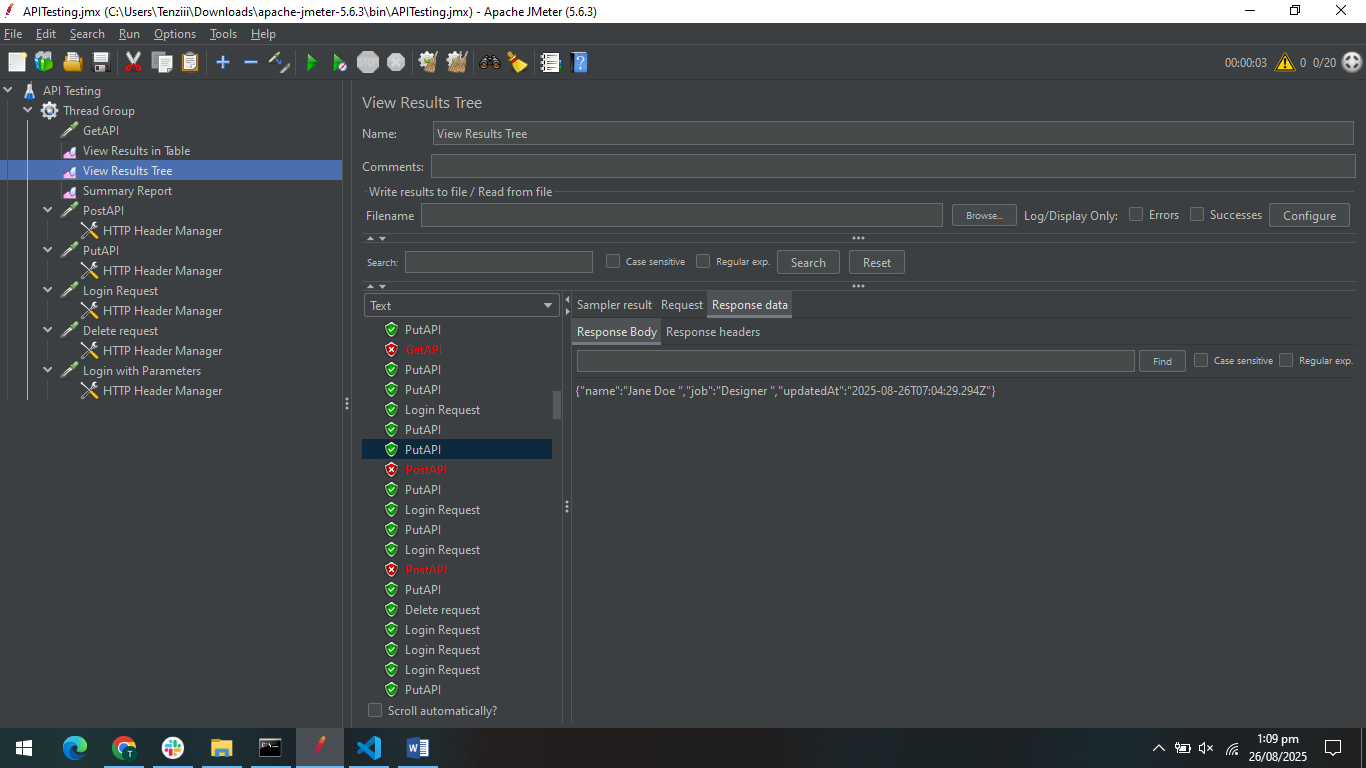
Loop Count 2

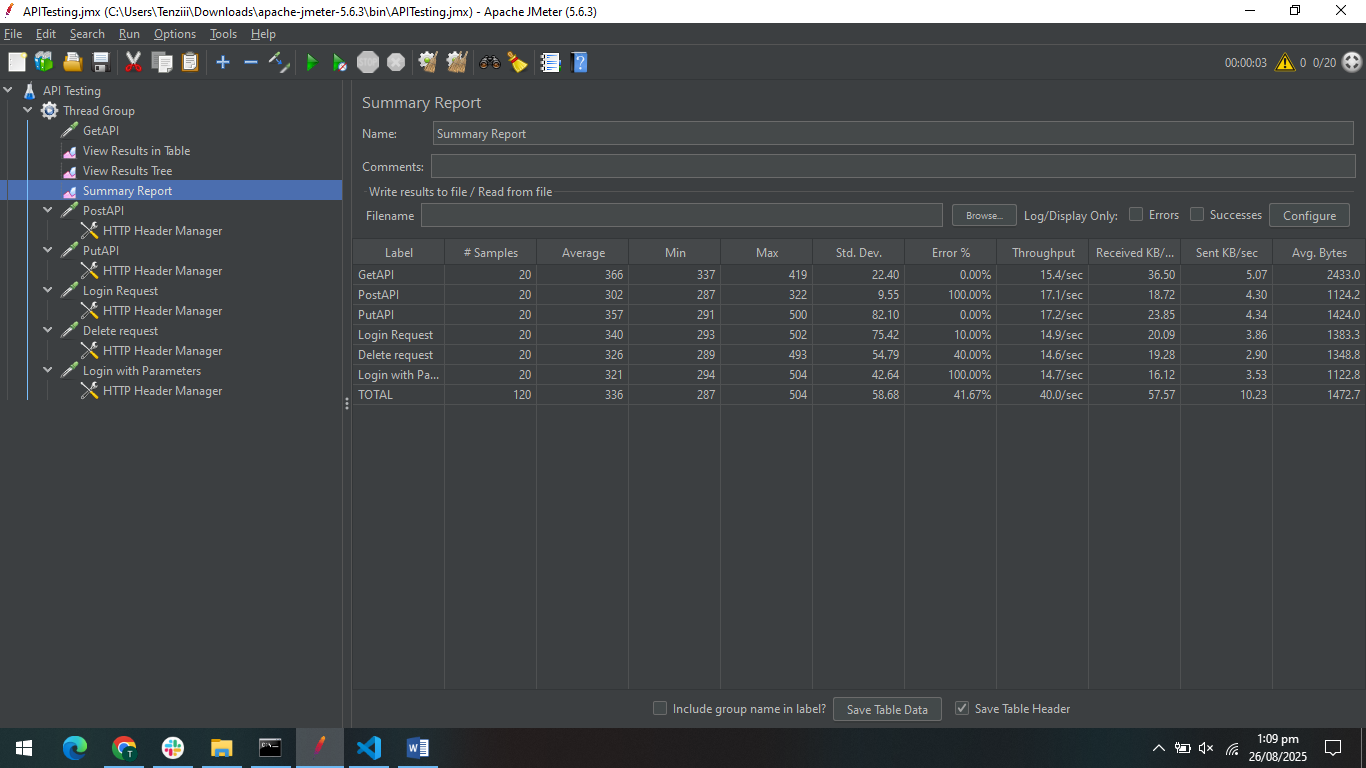
The test executed 20 requests across six APIs in ramp-up period 1 sec . While average response times remained slightly higher than acceptable limits ,but highest response time was 514ms for post API, Throughput remained steady at ~26.9/sec.

Error Percentage for overall process slightly change to ~41%, and for some API it increases slightly



No of users: 20

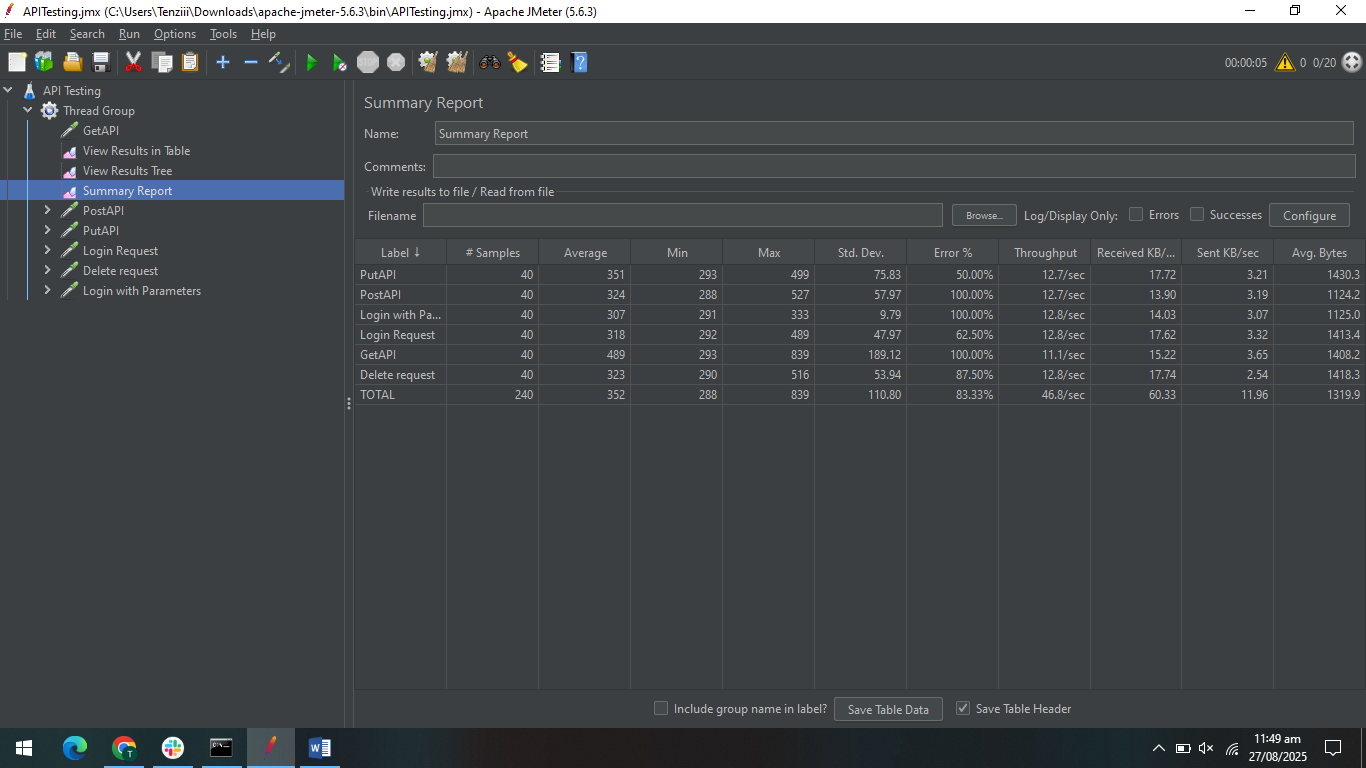




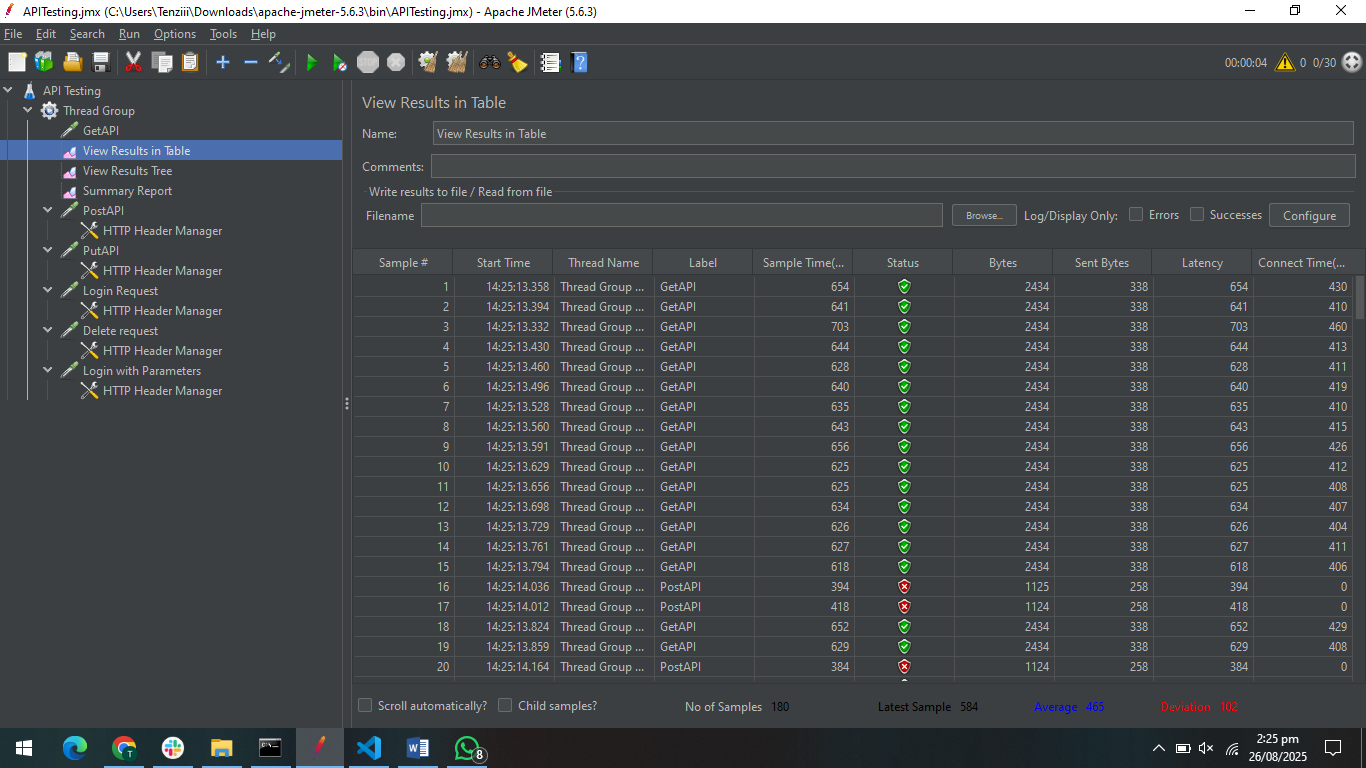
Loop Count 2

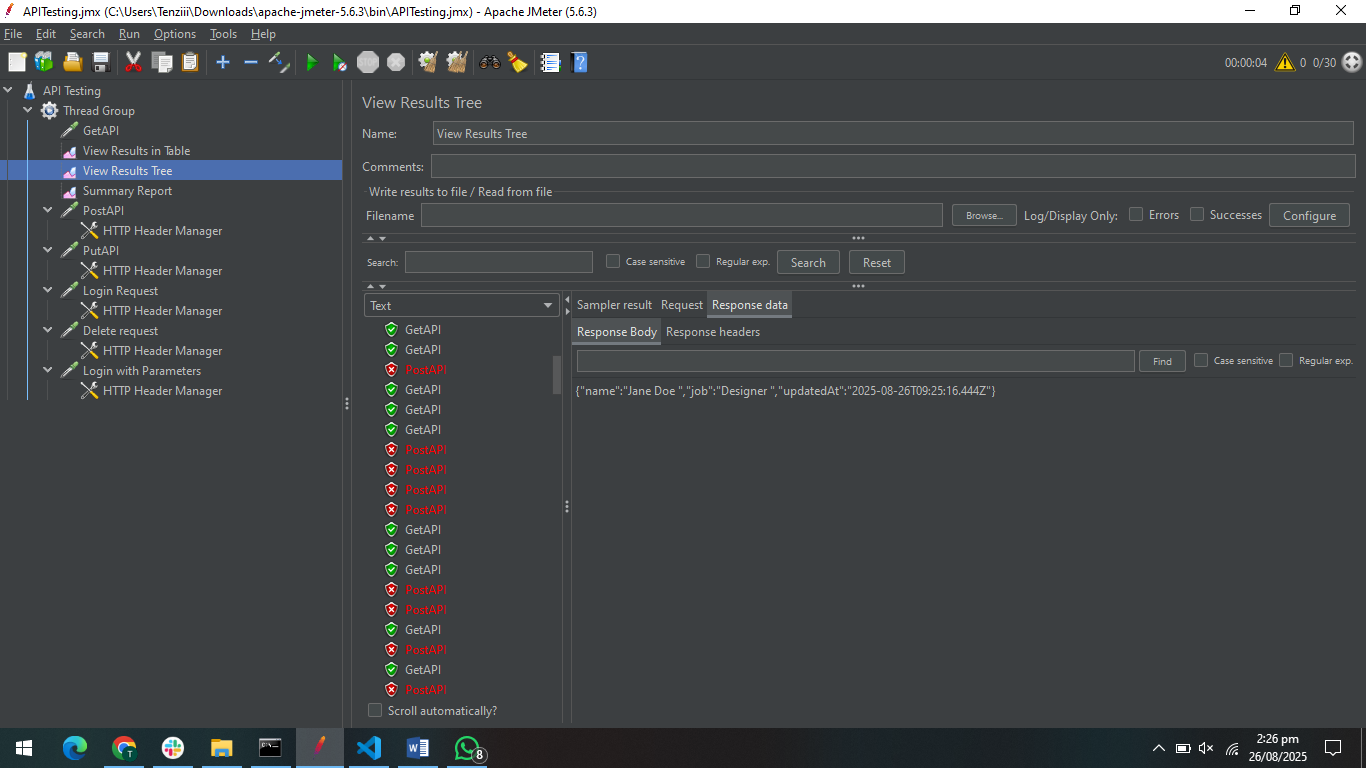
The test executed 40 requests across six APIs in ramp-up period 1 sec . While average response times remained slightly higher than acceptable limits ,but highest response time was 839ms for get API, Throughput remained steady at ~48.8/sec.

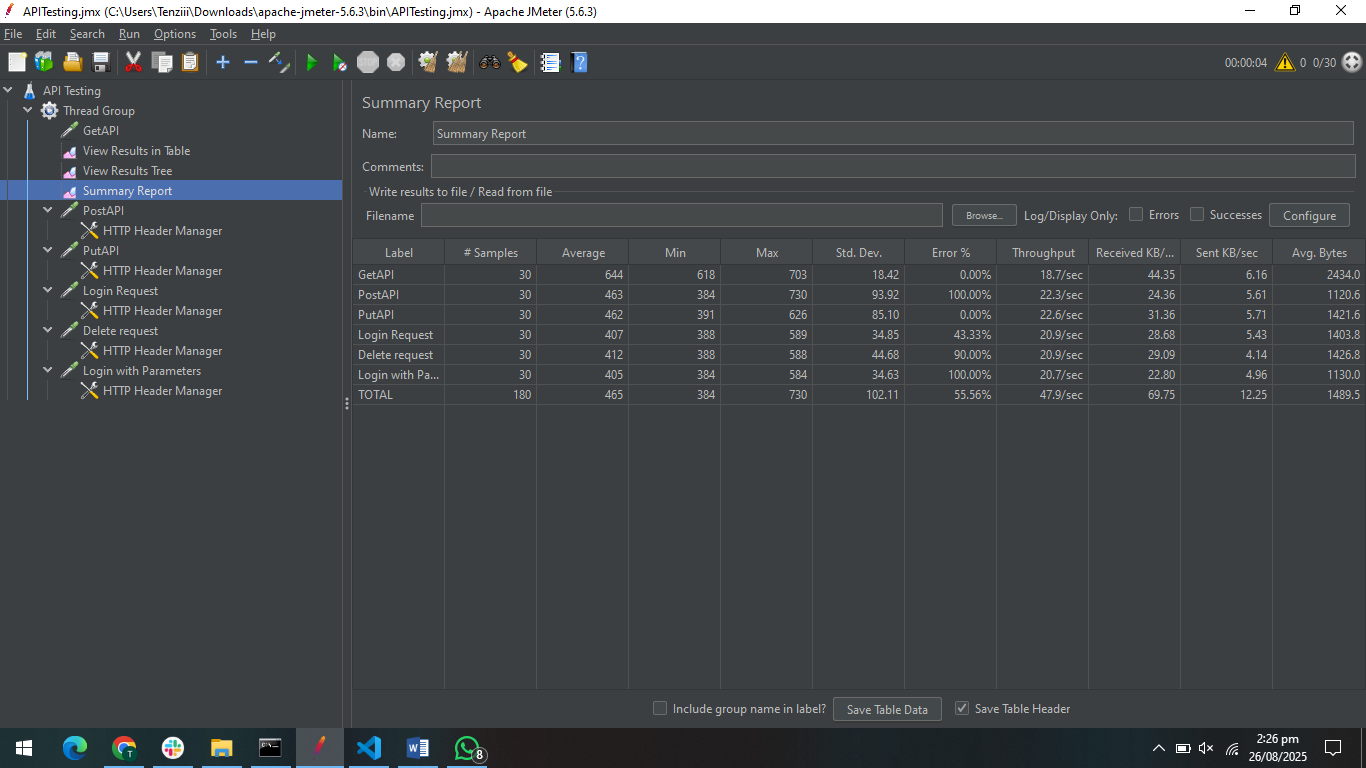
Error Percentage for overall process changes to ~88%, and for all API it increases gradually



No of users: 30



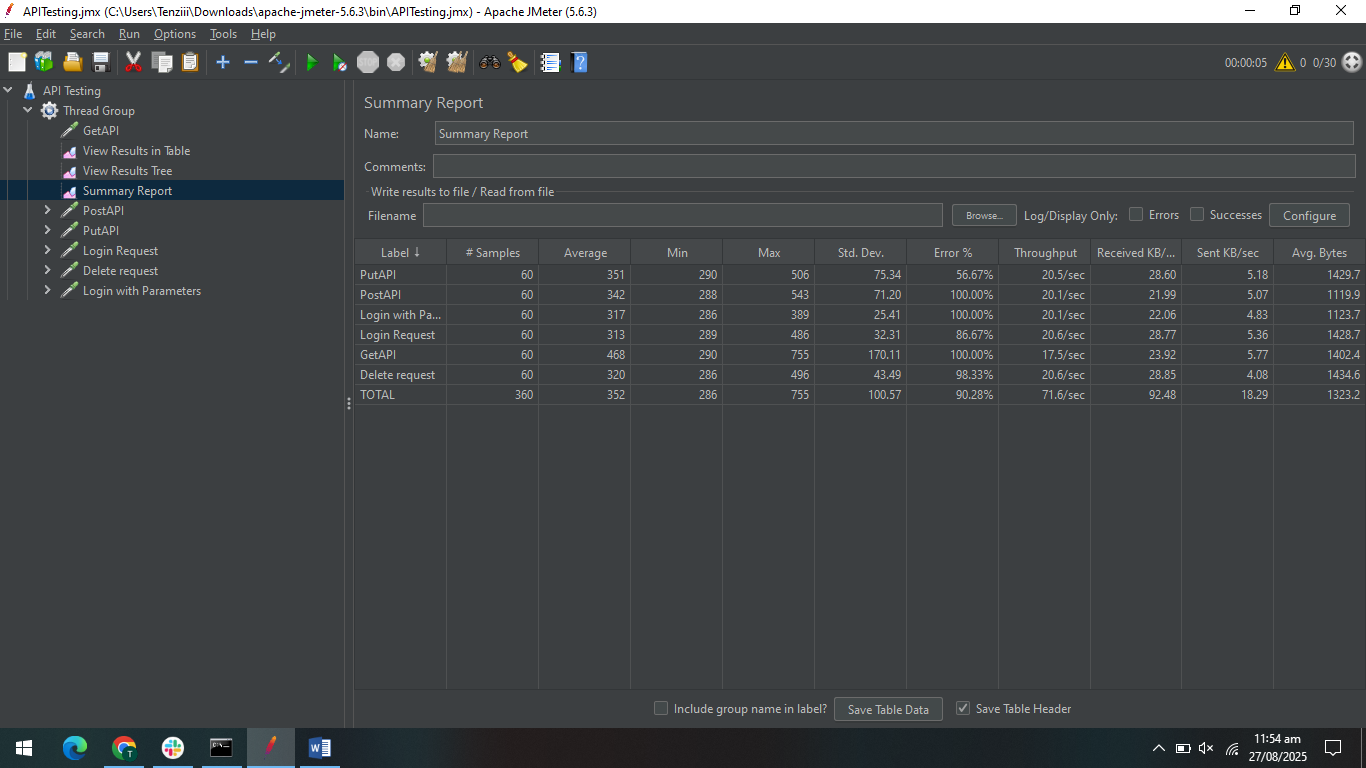




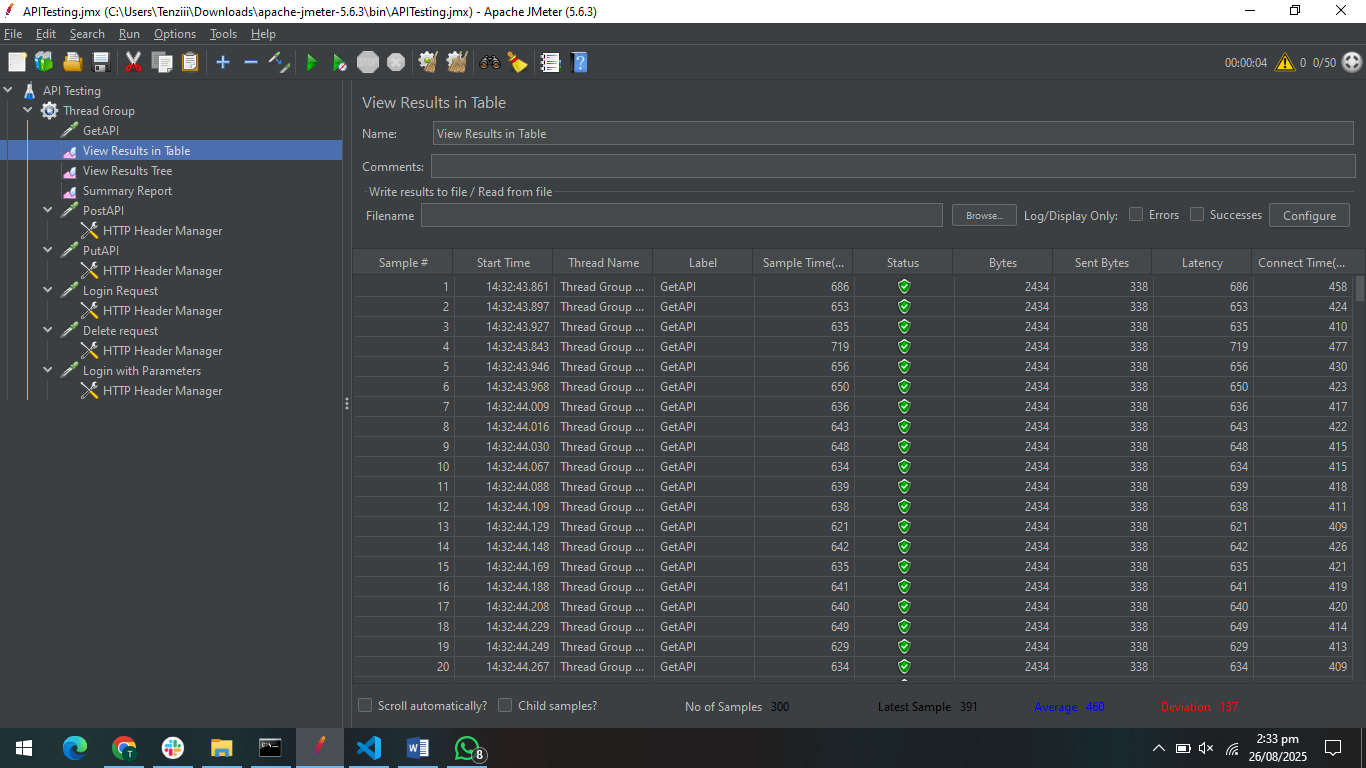
Loop Count 2

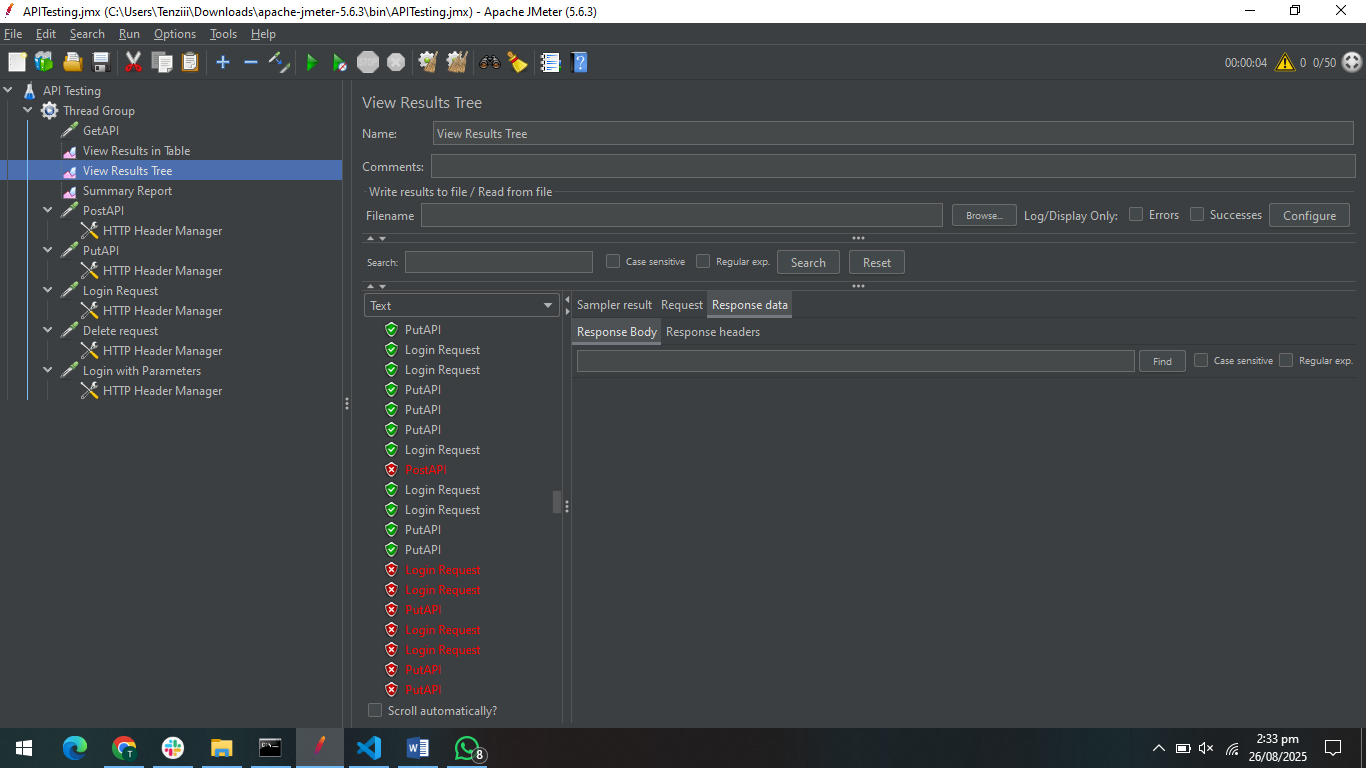
The test executed 60 requests across six APIs in ramp-up period 1 sec . While average response times remained slightly higher than acceptable limits ,but highest response time was 730ms for post API, Throughput remained steady at ~71.6/sec.

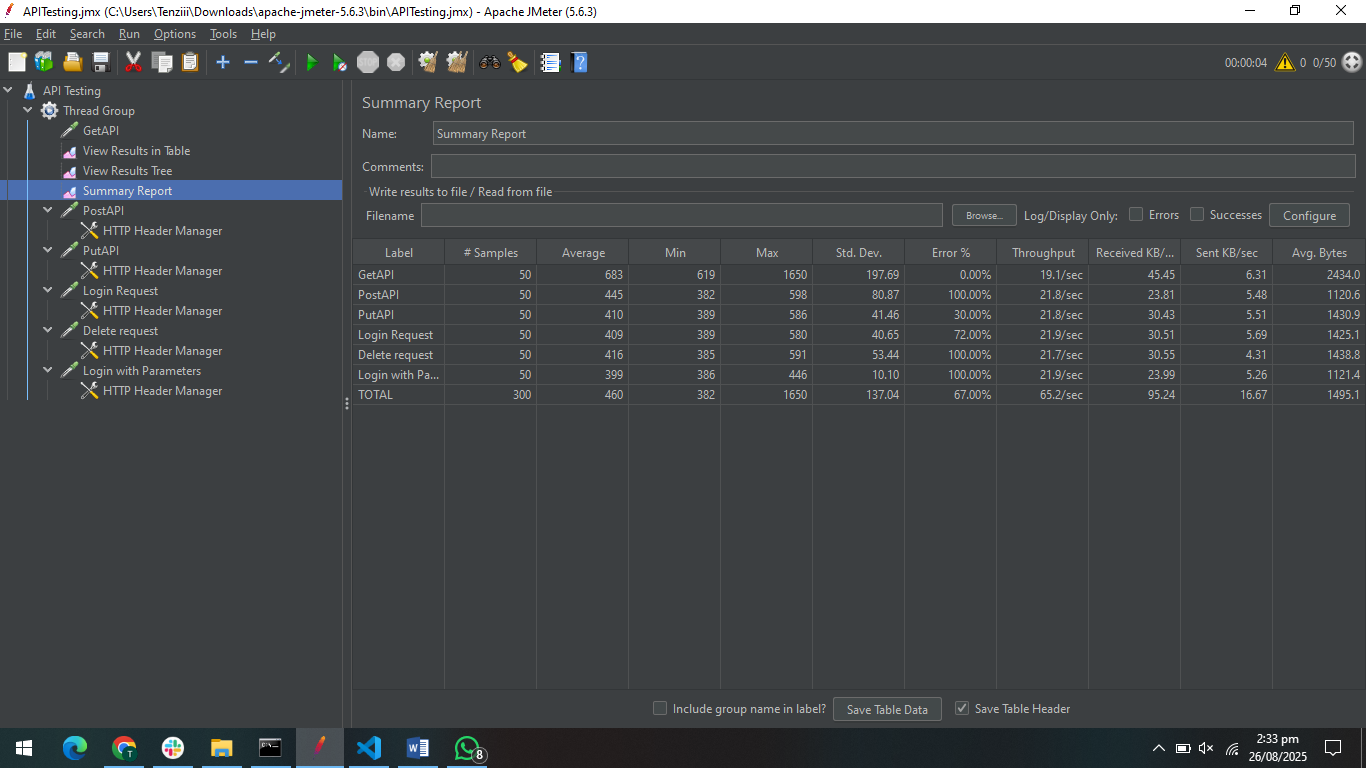
Error Percentage for overall process highly increase from ~55%, to ~90%, but for all API it increases



No of users: 50



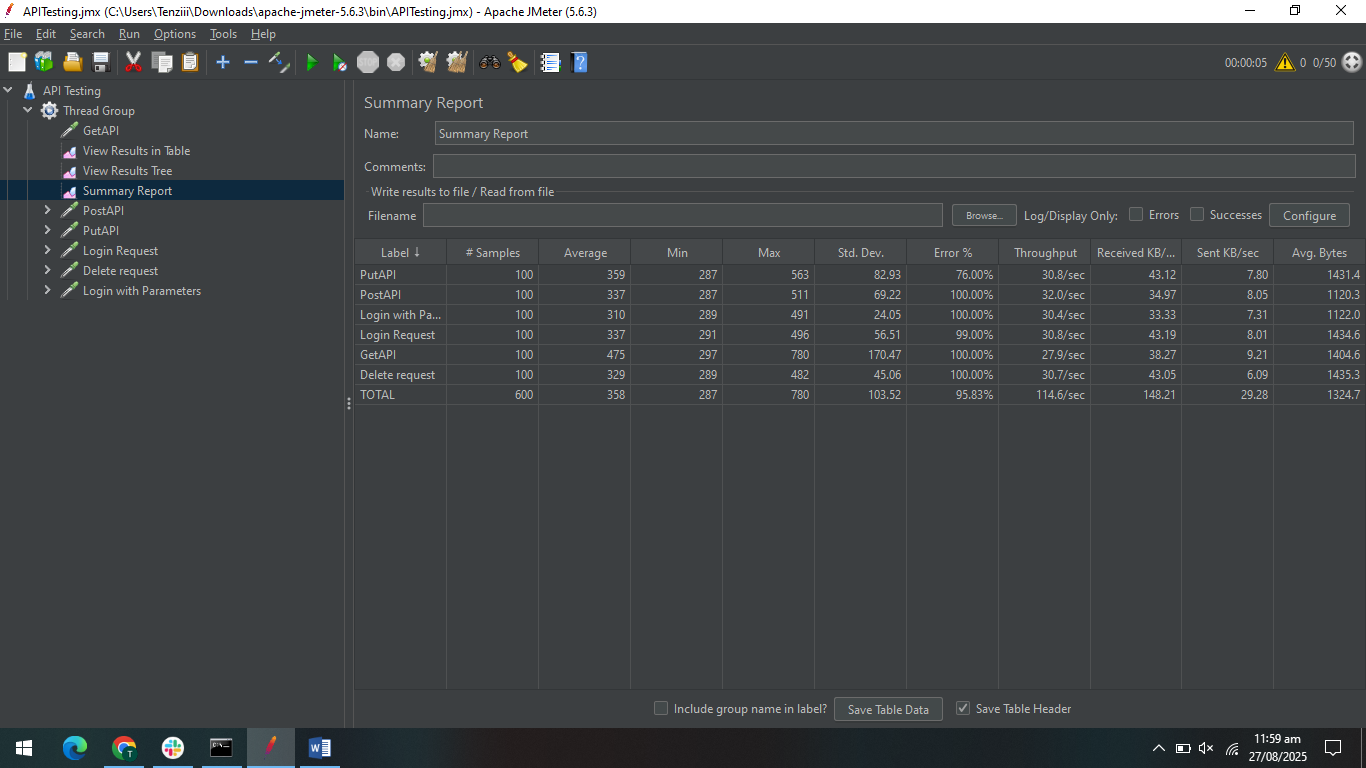




Loop Count 2

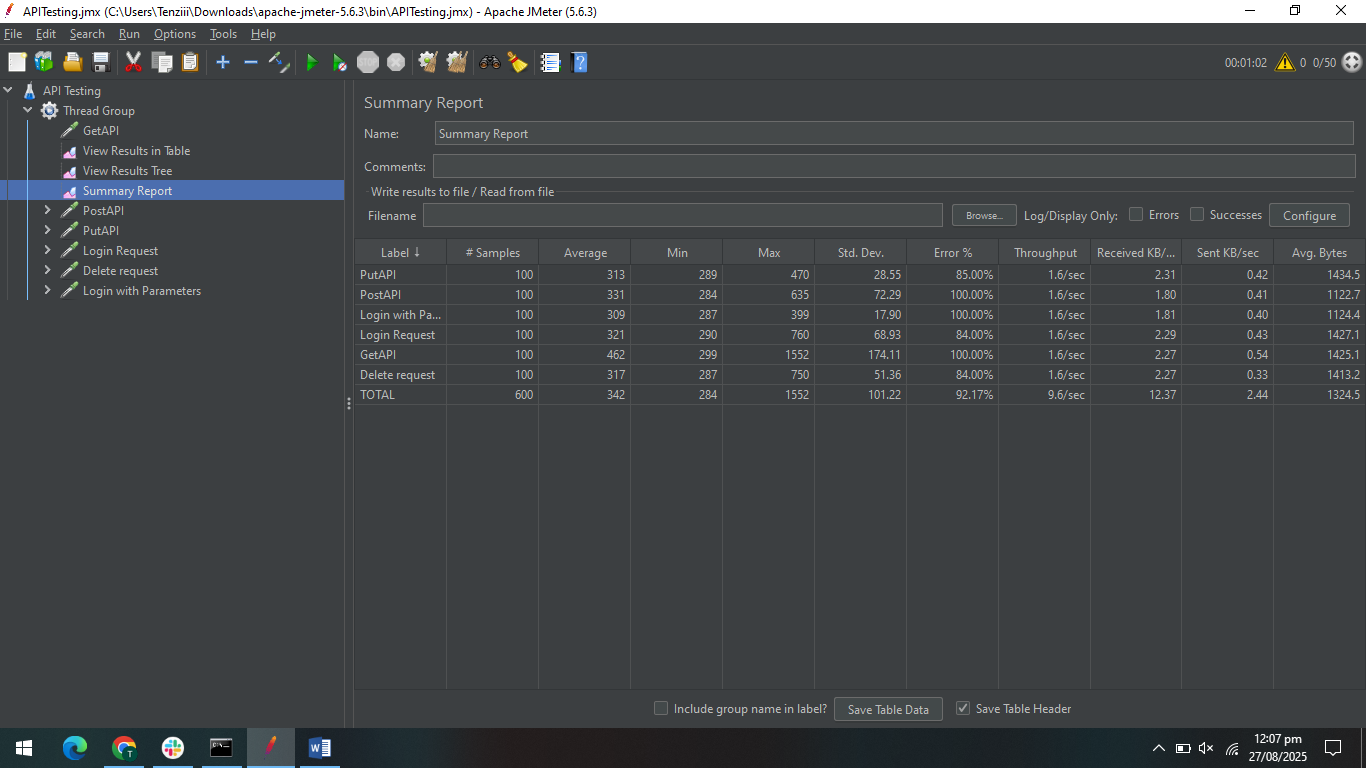
The test executed 300 requests across six APIs in ramp-up period 1 sec . While average response times remained slightly higher than acceptable limits , but highest response time was 780ms for get API, Throughput remained steady at ~114.6/sec, confirming the server can handle requests but requires debugging for error reduction.

Error Percentage for overall process gradually change from ~67% to ~95%, and for all API it increases to ~90%



Ramp-up Period (1 min)

The test executed 600 requests across six APIs in ramp-up period 1min . While average response times remained within acceptable limits (~342 ms) but highest response time was 1552ms for get API, the error rate was unusually high (84–100%), suggesting possible server-side/environmental misconfigurations rather than API logic failures. Throughput remained steady at ~9.6/sec, confirming the server can handle requests but requires debugging for error reduction.



### **Conclusion**

System handles up to **10–20 users smoothly** with acceptable response times. As load increases to **30+ users**, both response time and error percentage grow significantly. At **50+ users**, performance degradation is clear with high failure rates. Under **600 users**, the system shows severe instability (up to **95% errors**) even though throughput increases, indicating the server cannot handle high concurrency without optimization.