Performance Test Plan

### **1. Purpose**

The purpose of this test plan is to define the approach and scope of load testing for JSONPlaceholder using Apache JMeter, including executing GET, POST, PUT, PATCH, and DELETE requests to JSONPlaceholder.

## **2. Testing strategy**

### **2.1 Types of Testing**

1. **Load Testing -** checking the system performance under different load levels.
2. **Stress Testing -** determining the behavior of the system under extreme conditions.

### **2.2 Approaches to Testing**

Using the Apache JMeter tool to perform load tests.

### **3. Entry Criteria**

**Availability of the test environment**: The test environment must be ready to perform load testing. This includes:

* An installed and configured application server similar to the production environment.
* Stable internet connection to ensure continuous access to servers and services during testing.

**Installation of the necessary software:**

* Java Development Kit (JDK): Version 8 or higher. It is recommended to use the latest stable version to ensure compatibility and performance.
* Apache JMeter: Version 5.4.1 or higher. It is recommended to use the latest stable version to access all features and enhancements.

**Availability of test data:**

* Prepare the necessary test data to execute all planned load test scenarios.
* The data should be realistic and sufficient to simulate real-world use of the system.

### **4. Exit Criteria**

**Performing all scheduled load tests.**

**Identify and document all significant performance issues.**

## **5. Test tools**

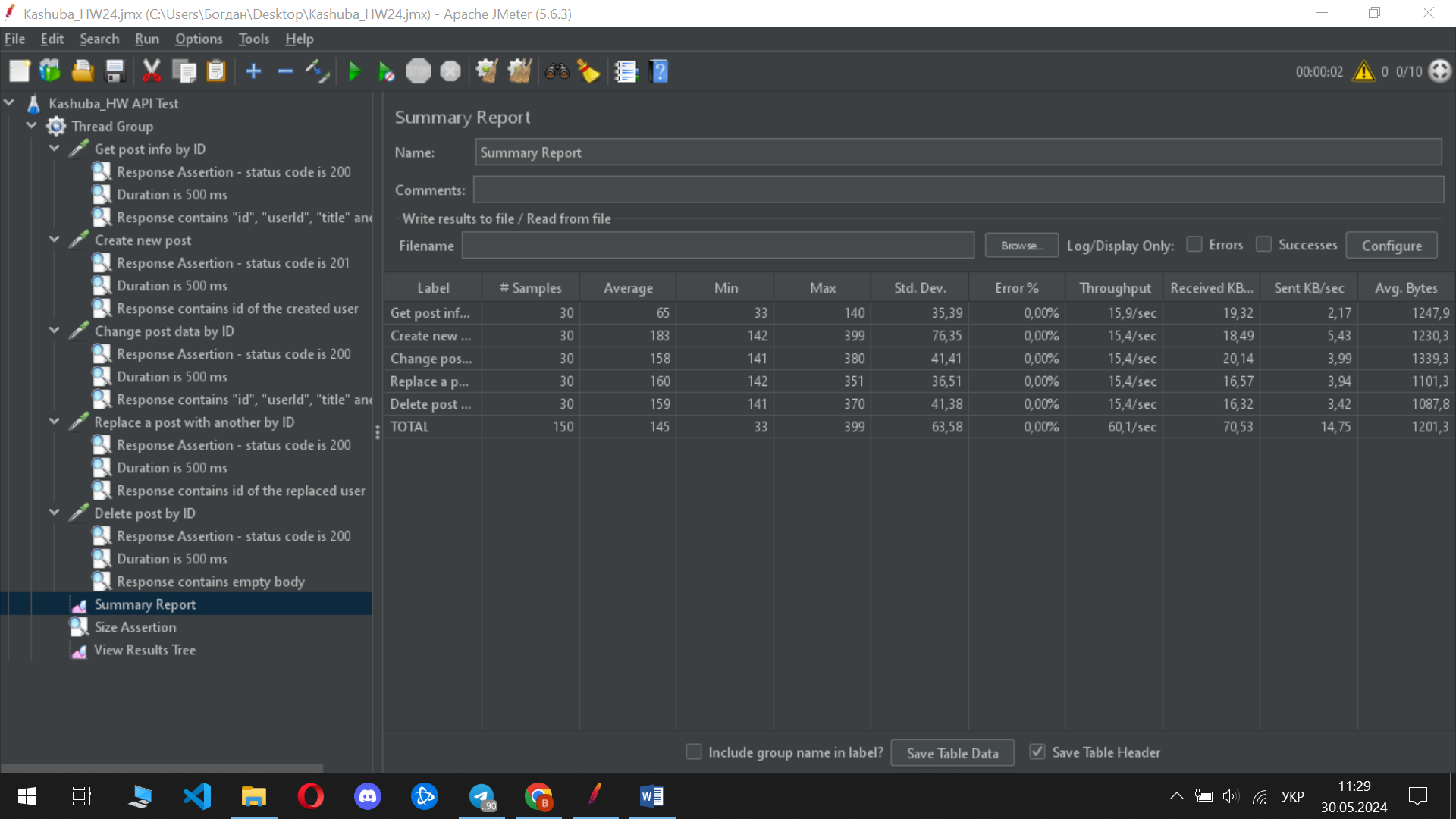
1. **Apache JMeter** to create and execute load tests.
2. **Google Docs** for writing documentation.
3. [**JSONPlaceholder**](https://jsonplaceholder.typicode.com/)as a free online REST API.

## **6. Load testing**

### **6.1 Description of the Test Scenario**

* **Number of users:** 10
* **Test duration:** 10 seconds per iteration, 3 iterations
* **Type of requests:** GET, POST, PUT, PATCH, DELETE to JSONPlaceholder

### **6.2 Results.**

* **Average response time: 145 ms**
* **Maximum response time: 399 ms**
* **Number of errors: 0**
* 

### **6.3 Conclusion**

Based on the conducted load testing, the system demonstrated excellent performance under the load of up to 10 simultaneous users. All requests (GET, POST, PUT, PATCH, DELETE) to the JSONPlaceholder were processed within the acceptable response time (average response time: 145 ms, maximum response time: 399 ms) and without errors, which indicates high system stability.

## 

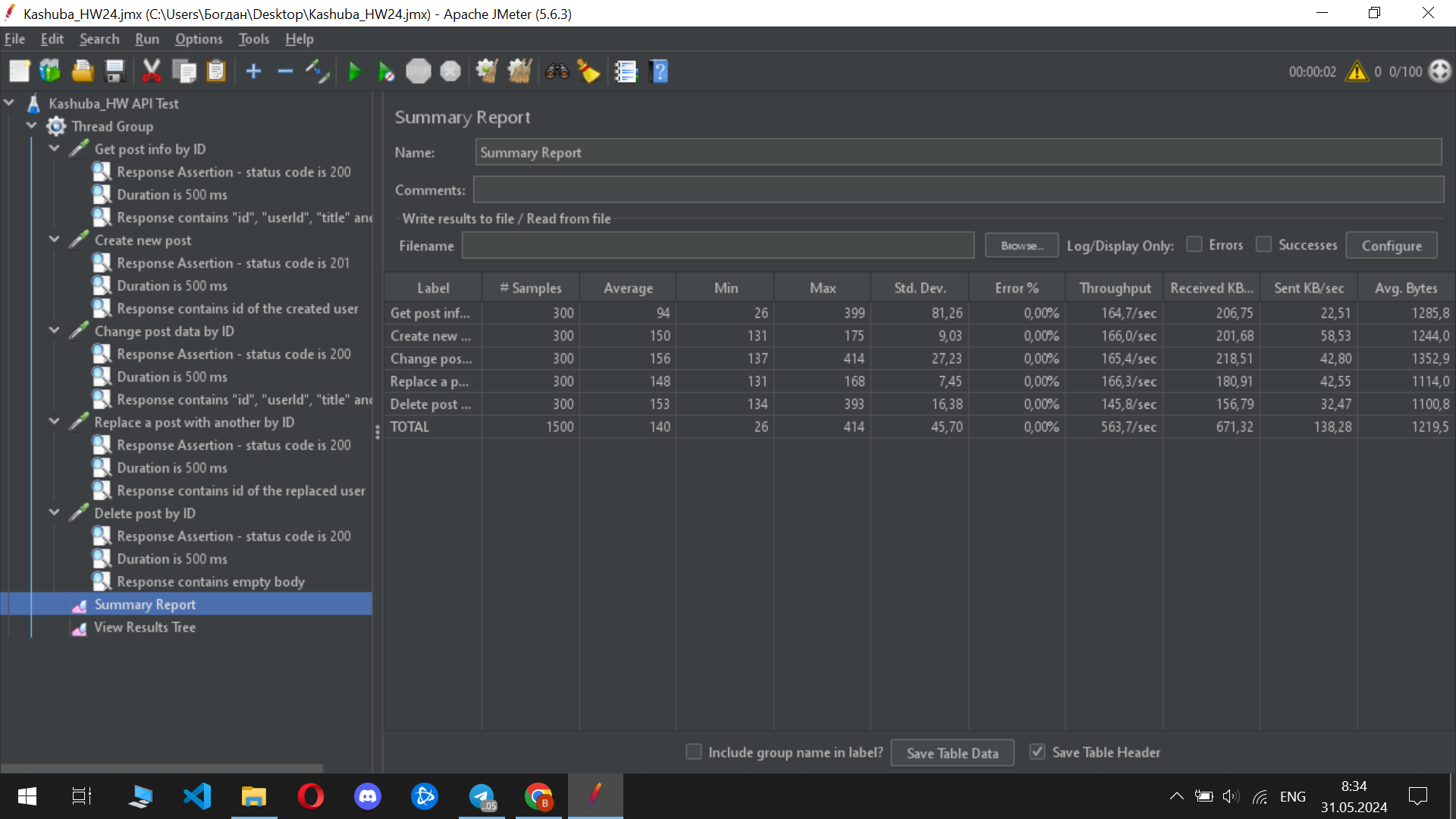
## **7. Stress testing**

### **7.1 Description of the Test Scenario**

* **Number of users:** 1000
* **Objectives:** Gradual increase from 100 to 1000 users

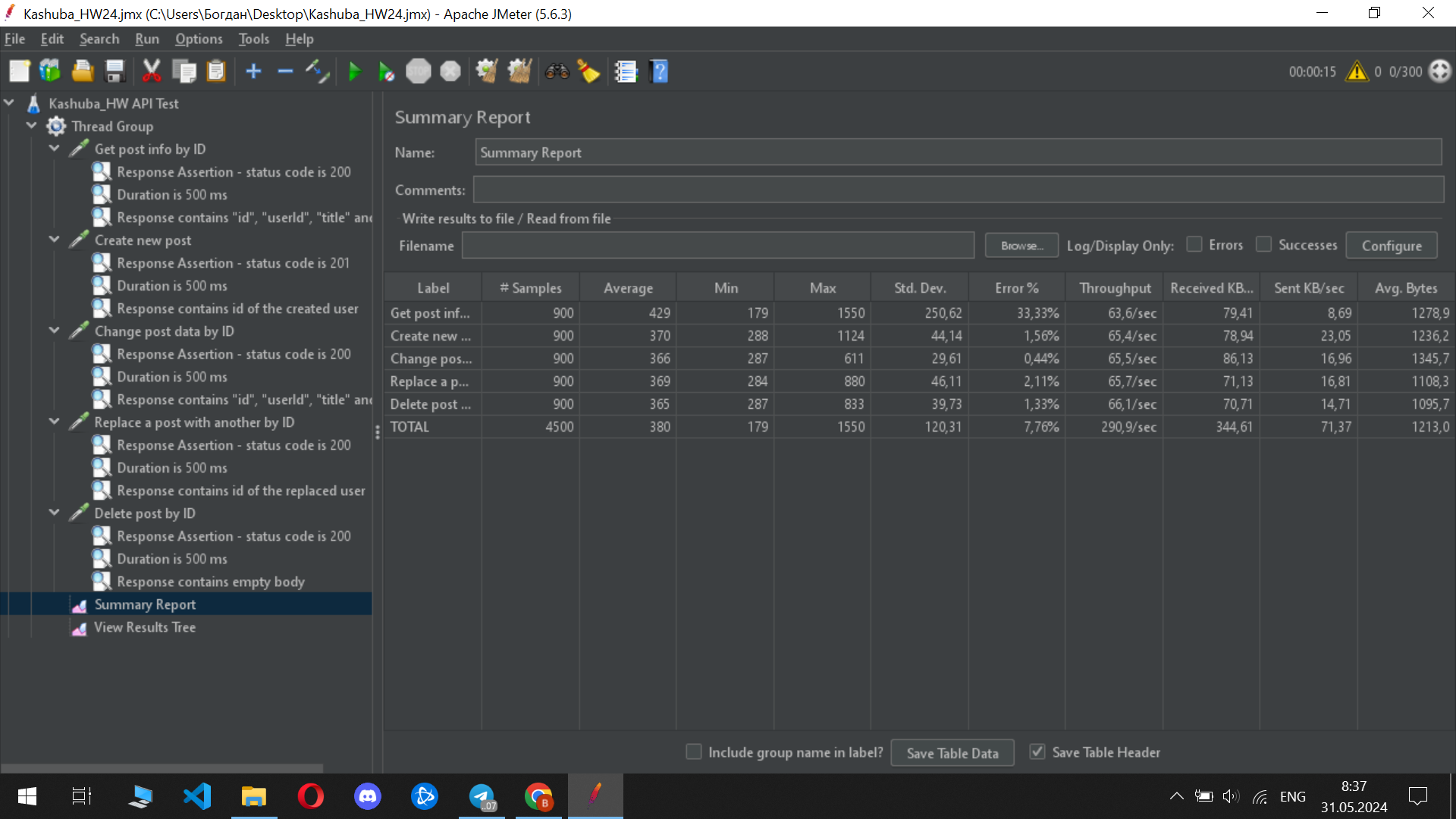
### **7.2 Results**

Load of 100 users

* **Average response time:** 140 ms
* **Maximum response time:** 414 ms
* **Percentage of errors:** 0%
* **Conclusion:** system works without errors under a load of 100 users
* 

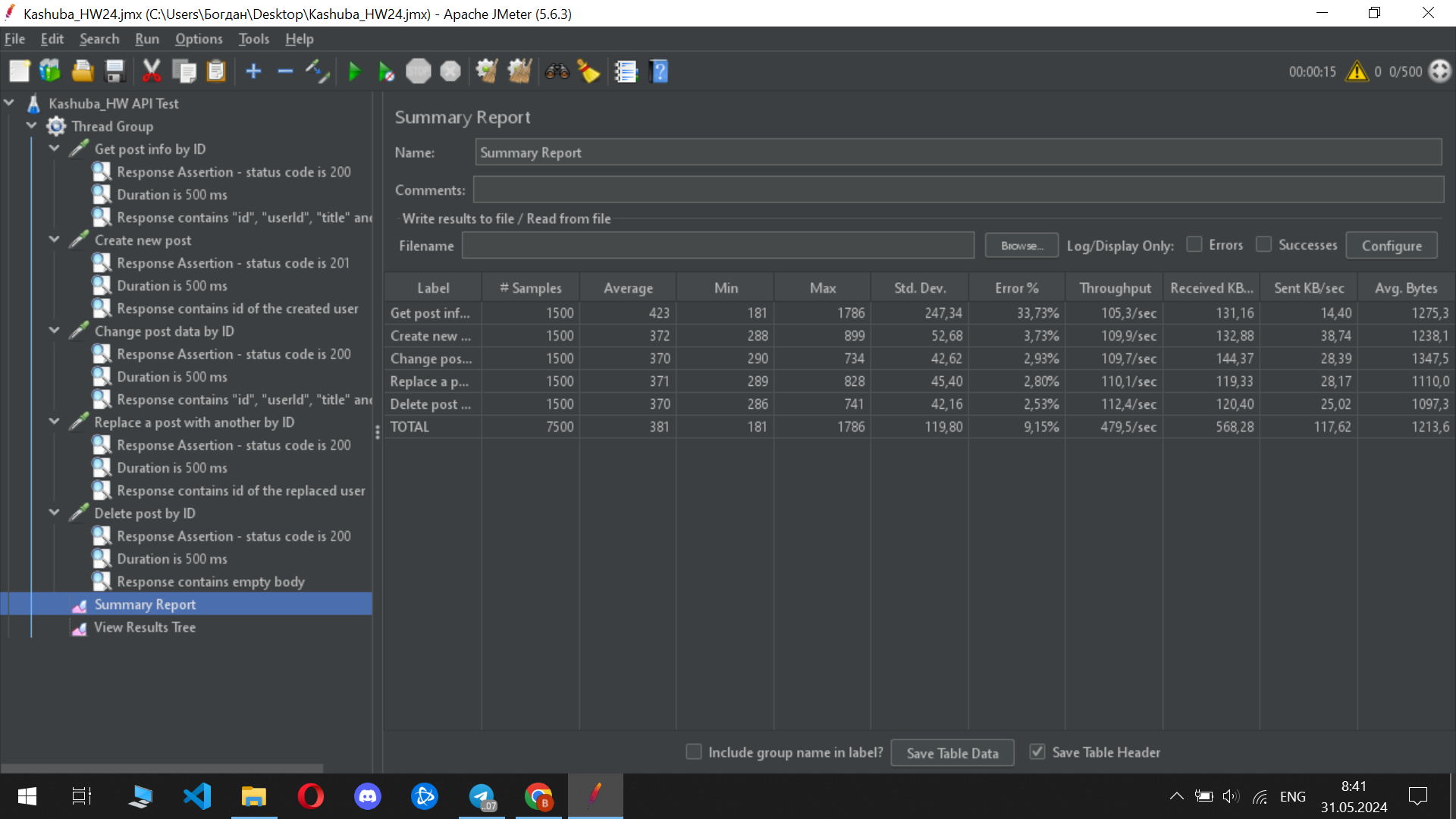
Load of 300 users

* **Average response time:** 380 ms
* **Maximum response time:** 1550 ms
* **Percentage of errors:** 7.78%
* **Conclusion:** The system starts to show delays, but remains stable.



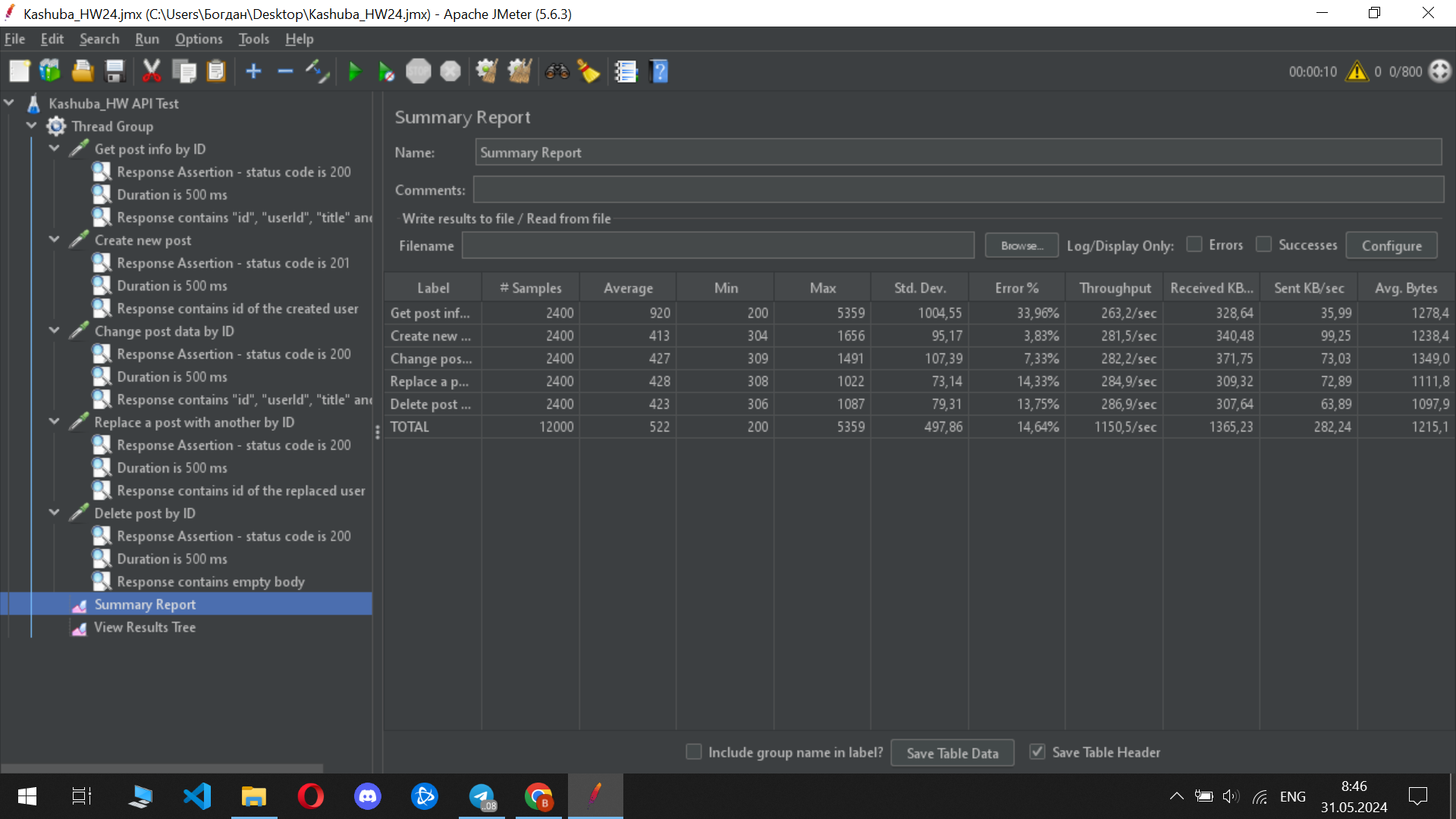
Load of 500 users

* **Average response time:** 381 ms
* **Maximum response time:** 1786 ms
* **Average error rate:** 9.15%
* **Conclusion:** The system shows a significant decrease in performance.

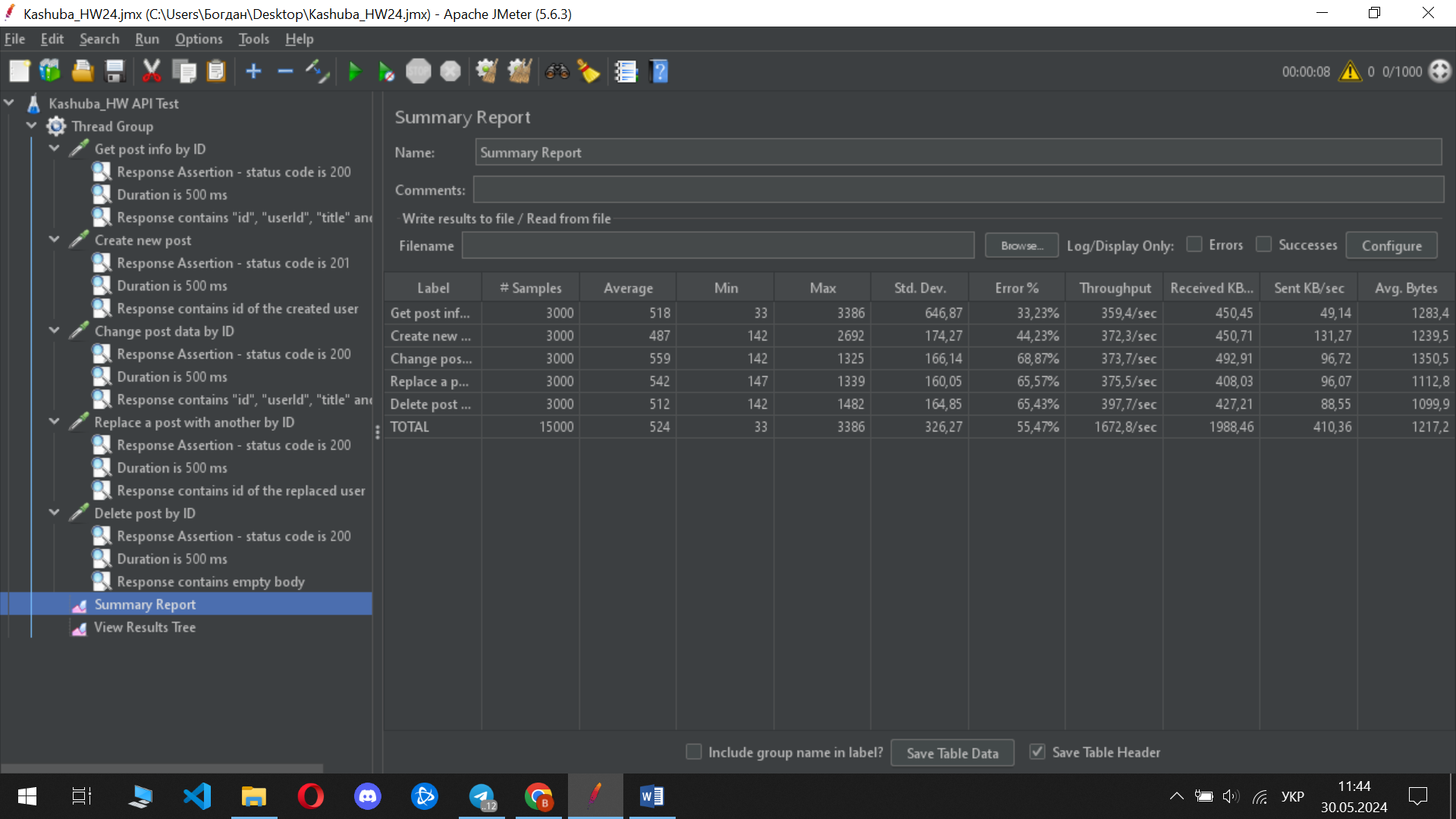


Load of 800 users

* **Average response time:** 522 ms
* **Maximum response time:** 5359 ms
* **Average error rate:** 14.64%
* **Conclusion:** The system shows a significant decrease in performance and a large number of errors.



Load of 1000 users

* **Average response time:** 524 ms
* **Maximum response time:** 3386 ms
* **Average error rate:** 55,47%
* **Conclusion:** system fails to withstand the load, critical performance degradation
* 

### 

### **7.3 Conclusion**

The system demonstrates stable and correct operation under a load of up to 300 simultaneous users. Starting with 500 users, performance significantly decreases and the number of errors increases. After reaching a load of 800 users, the system cannot cope with the load, which leads to a significant increase in response time and an unacceptably high percentage of errors. With a load of 1000 users, the system stops working correctly.