אוניברסיטת בן-גוריון בנגב
Ben-Gurion University of the Negev

הפקולטה למדעי ההנדסה
המחלקה להנדסת תוכנה

Faculty of Engineering Sciences
Dept. of software Engineering

Ben-Gurion University
of the Negev

"Trading System"

Version 4:
Analysis report of Load And Stress Testing

Team: 11B

Ziv Cohen Gvura

Eli Ben Shimol

Michael Daniarov

Nave Avraham

Chai-Shalev Hadad

Table of Contents

System Expectations:

The system is designed to accommodate various types of users, each with specific roles and expected frequencies of usage. The following five user types have been identified:

Guest: This type of user refers to visitors who are not registered in the system. It is estimated that approximately 55% of users will fall into this category.

Member: Members are users who have registered with the system. It is expected that around 45% of users will be registered members.

Store Creator: This user type represents owners who have created a store within the system. The estimated percentage of users falling into this category is about 10%.

Store Owner: Store owners are individuals who own a store within the system. It is estimated that approximately 5% of users will be store owners.

Store Manager: Store managers are responsible for managing a store within the system. The estimated percentage of users falling into this category is about 5%.

Main System Operations:

The system revolves around three primary operations, each with its own expected frequency of occurrence:

Purchases: This operation involves activities such as adding products to the cart, managing the cart, and making purchases. It is anticipated that purchases will constitute around 20% of the total system operations.

Product Search: Product search encompasses various types of searches within the system. It is expected that product searches will account for approximately 60% of the total operations performed within the system.

Store Management Operations: These operations involve tasks such as store openings, appointment inventory management, and other related activities. It is estimated that store management operations will comprise approximately 15% of the total system operations.

System Activity Summary:

Considering the projected system expectations and operations, the following key metrics have been identified:

Number of Users: It is expected that the system will accommodate an average of 1000 users per day.

User Registrations: The system is anticipated to receive approximately 20 new user registrations per day.

By analyzing these factors, we can gain insights into the system's load and stress capacity, ensuring it can handle the expected user base and activities without experiencing performance degradation or failures. Load and stress testing will be conducted to evaluate the system's performance under these projected conditions and validate its ability to meet the anticipated demands effectively.

Indexes:

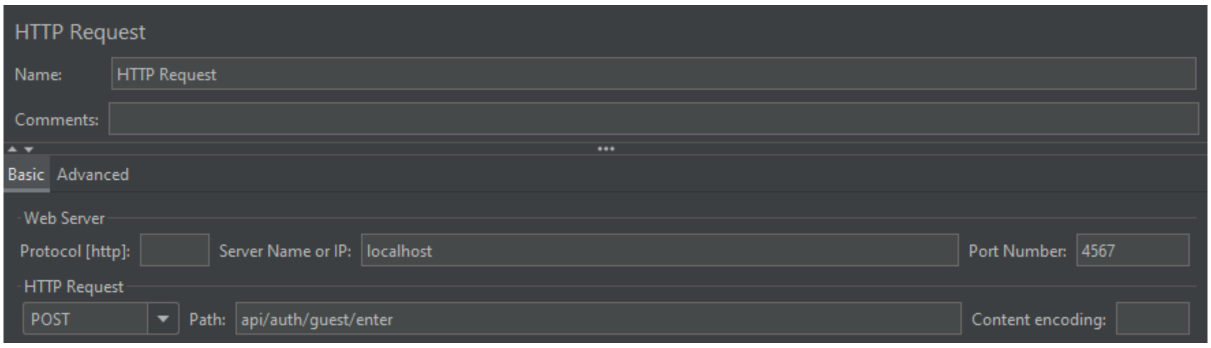
Index 1 – request at the same time:

Dealing with ____ requests at the same time while meeting a response time of no more than a second per request.

Scenario 1 – Guest enters to system:

Dealing with **2000** requests of enter guest to the system at the same time while meeting a response time of no more than a second per request.

HTTP Request:



HTTP Request
Name: HTTP Request
Comments:
Basic Advanced
Web Server
Protocol [http]: Server Name or IP: localhost Port Number: 4567
HTTP Request
POST Path: api/auth/guest/enter Content encoding:

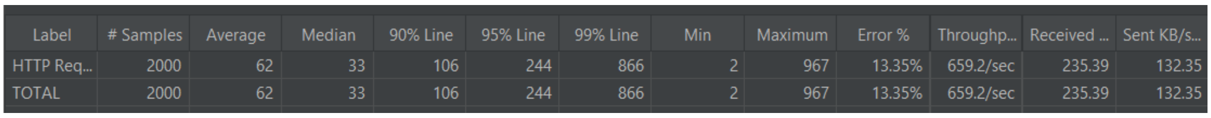
Result:

Success rate: 86.65%

Average time: 62 ms.

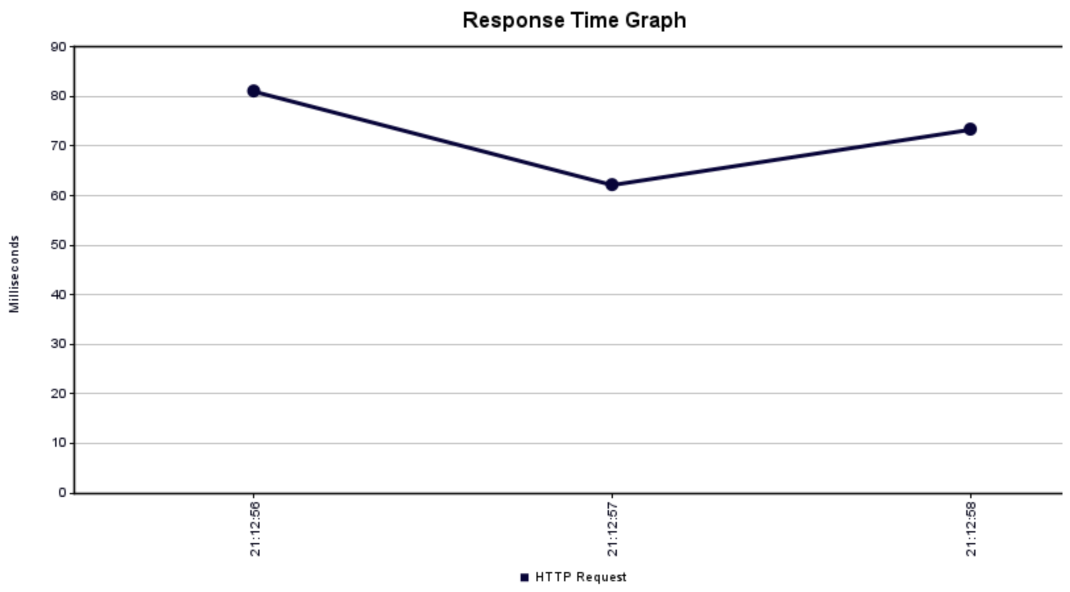
Max time: 976 ms.

Min time: 2 ms.



Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughp...	Received ...	Sent KB/s...
HTTP Req...	2000	62	33	106	244	866	2	967	13.35%	659.2/sec	235.39	132.35
TOTAL	2000	62	33	106	244	866	2	967	13.35%	659.2/sec	235.39	132.35

Response time graph:



<u>Conclusions:</u>

- The system effectively handles the entry of 2000 users simultaneously with high success rates.

Scenario 2- register to the system:

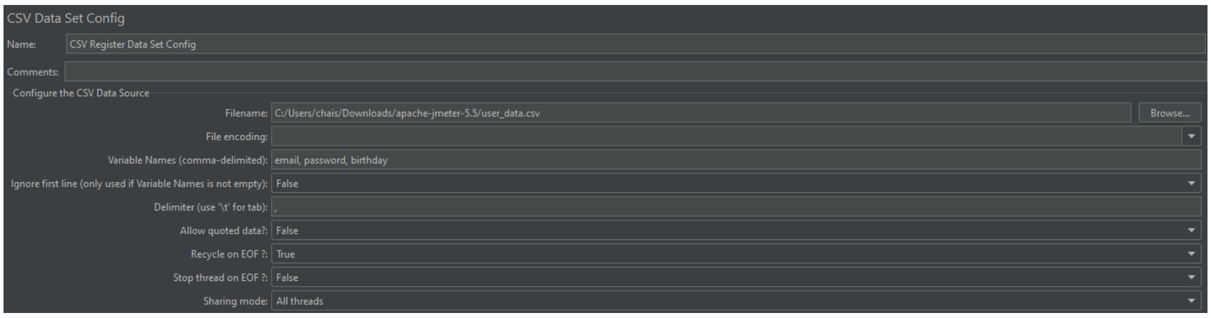
Dealing with **40** requests of register to the system at the same time while meeting a response time of no more than a second per request.

Process description:

We generated 10,000 valid records with a unique email address for each and prepared a csv file.

After that, using the JMeter tool, we loaded the csv file and verified the http request that will allow us to receive data from the csv file into the content of the request.

as follows:
CSV Data Set Config:



CSV Data Set Config

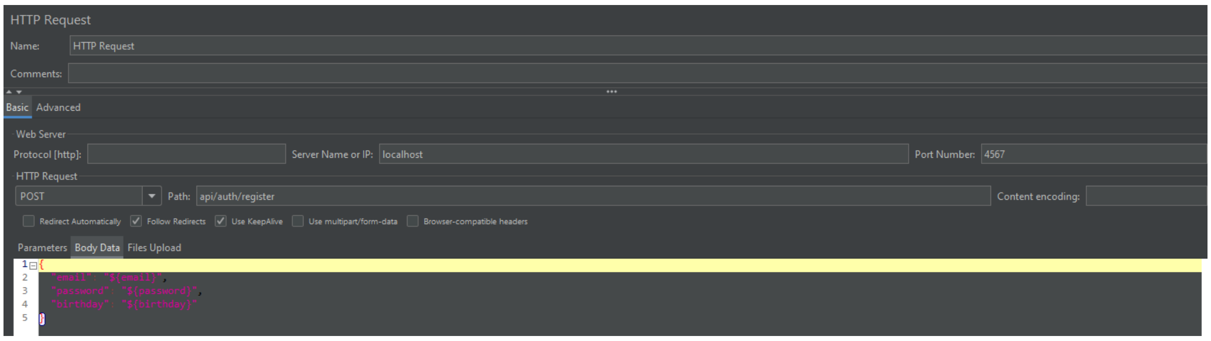
Name: CSV Register Data Set Config

Comments:

Configure the CSV Data Source

Filename:	C:/Users/chais/Downloads/apache-jmeter-5.5/user_data.csv
File encoding:	
Variable Names (comma-delimited):	email, password, birthday
Ignore first line (only used if Variable Names is not empty):	False
Delimiter (use '\t' for tab):	,
Allow quoted data?:	False
Recycle on EOF ?:	True
Stop thread on EOF ?:	False
Sharing mode:	All threads

HTTP Request:



HTTP Request

Name: HTTP Request

Comments:

Basic Advanced

Web Server

Protocol [http]: Server Name or IP: localhost Port Number: 4567

HTTP Request

POST Path: api/auth/register Content encoding:

Redirect Automatically [✓ Follow Redirects] [✓ Use KeepAlive] Use multipart/form-data Browser-compatible headers

Parameters Body Data Files Upload

```
{
  "email": "${email}",
  "password": "${password}",
  "birthday": "${birthday}"
}
```

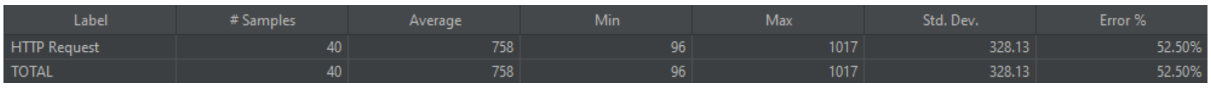
<u>Result:</u>

<u>Success rate:</u> 47.5%

<u>Average time:</u> 758 ms.

<u>Max time:</u> 1017 ms.

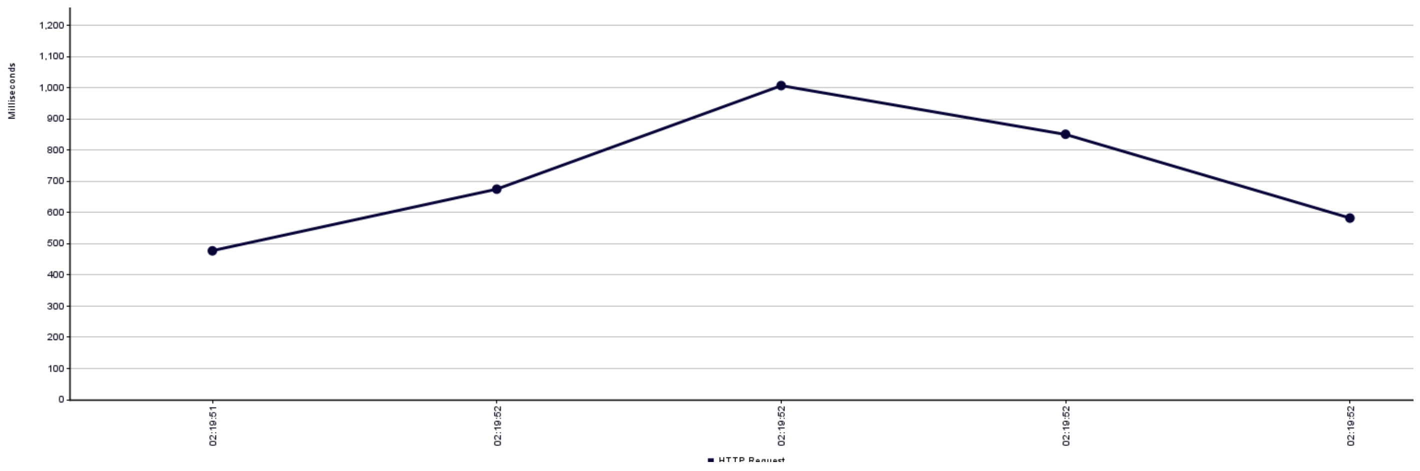
<u>Min time:</u> 96 ms.



Label	# Samples	Average	Min	Max	Std. Dev.	Error %
HTTP Request	40	758	96	1017	328.13	52.50%
TOTAL	40	758	96	1017	328.13	52.50%

* Please note that there is a max that is over a second, but that is because it is a total time, we limited the response time to one second.

Response time graph:



Conclusions:

- The system demonstrates efficient handling of simultaneous registration for 40 users, achieving satisfactory percentages that are neither exceptionally high nor low.
- This outcome is especially commendable considering our projected expectation of entries being two times lower per day, as opposed to per second.

Scenario 3 – Login to system:

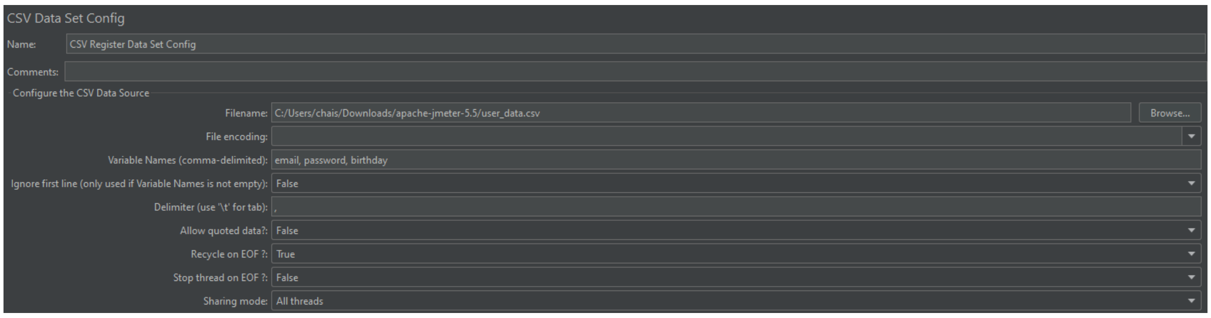
Dealing with **1750** requests of enter guest to the system at the same time while meeting a response time of no more than a second per request.

Process description:

We generated 10,000 valid records with a unique email address for each and prepared a csv file.

After that, using the JMeter tool, we loaded the csv file and verified the http request that will allow us to receive data from the csv file into the content of the request.

as follows:
CSV Data Set Config:



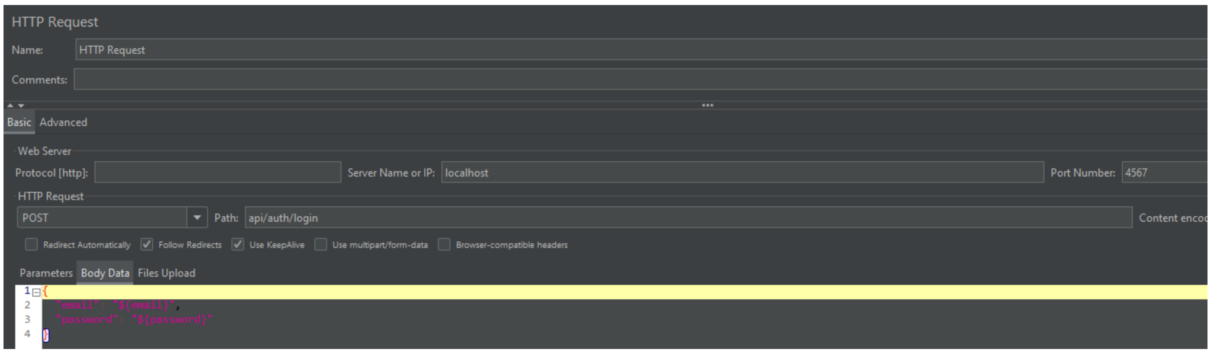
CSV Data Set Config

Name:	CSV Register Data Set Config
Comments:	

Configure the CSV Data Source

Filename:	C:/Users/chais/Downloads/apache-jmeter-5.5/user_data.csv
File encoding:	
Variable Names (comma-delimited):	email, password, birthday
Ignore first line (only used if Variable Names is not empty):	False
Delimiter (use '\t' for tab):	,
Allow quoted data?:	False
Recycle on EOF ?:	True
Stop thread on EOF ?:	False
Sharing mode:	All threads

HTTP Request:



HTTP Request

Name: HTTP Request

Comments:

Basic Advanced

Web Server

Protocol [http]: Server Name or IP: localhost Port Number: 4567

HTTP Request

POST Path: api/auth/login Content encod

Redirect Automatically [✓ Follow Redirects] [✓ Use KeepAlive] Use multipart/form-data Browser-compatible headers

Parameters Body Data Files Upload

```
{
  "email": "${email}",
  "password": "${password}"
}
```

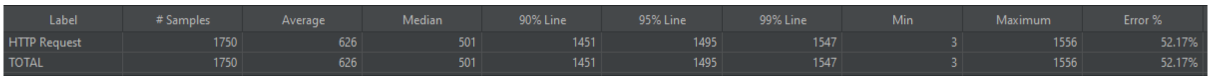
Result:

Success rate: 47.83%.

Average time: 626 ms.

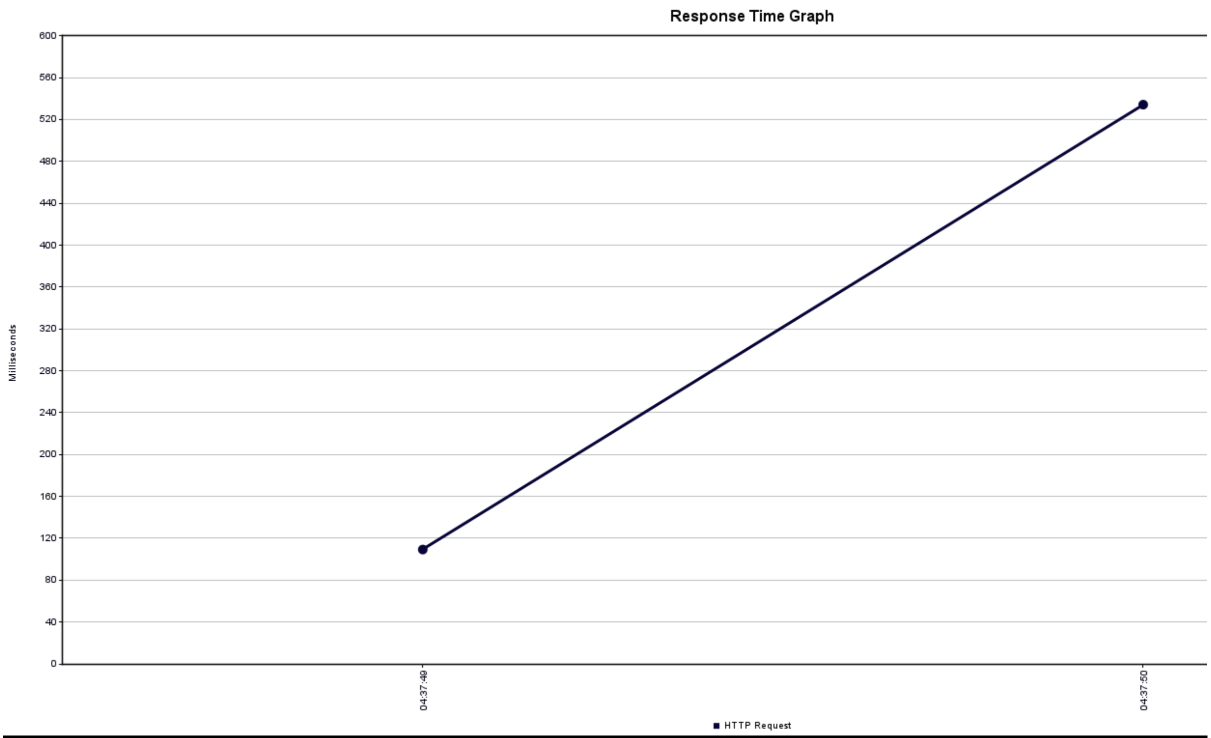
Max time: 1556 ms.

Min time: 3 ms.

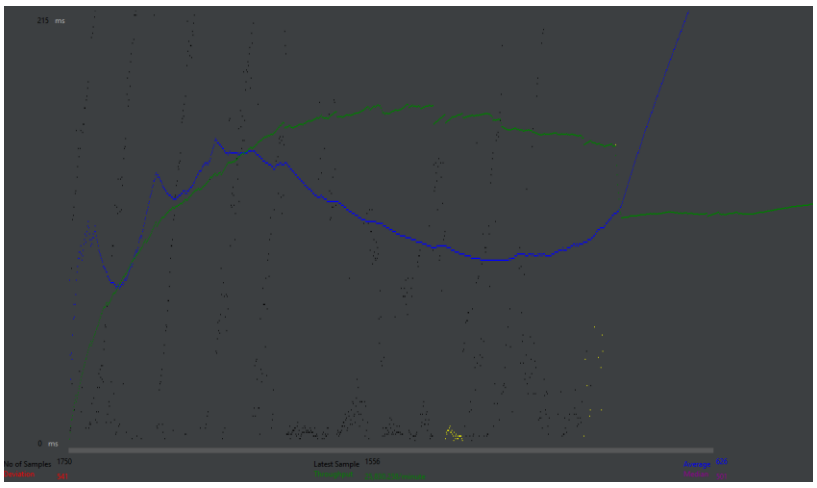


Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %
HTTP Request	1750	626	501	1451	1495	1547	3	1556	52.17%
TOTAL	1750	626	501	1451	1495	1547	3	1556	52.17%

Response time graph:



Graph Results:



Conclusions:

• The system demonstrates efficient handling of connecting to the system, while achieving satisfactory percentages that are neither high nor particularly low.

• This result is particularly commendable given our forecasted expectation that visits including both guests and users will be twice as low per day as opposed to per second.

Index 2 – Support for records:

Dealing with ____ requests at the same time while meeting a response time of no more than a second per request.

Scenario 1:

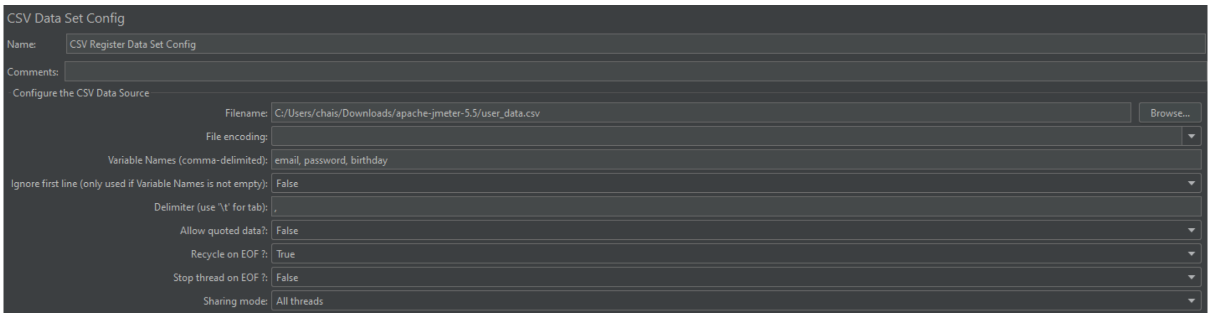
Support up to **5000** user records.

Process description:

We generated 10,000 valid records with a unique email address for each and prepared a csv file.

After that, using the JMeter tool, we loaded the csv file and verified the http request that will allow us to receive data from the csv file into the content of the request.

as follows:
CSV Data Set Config:



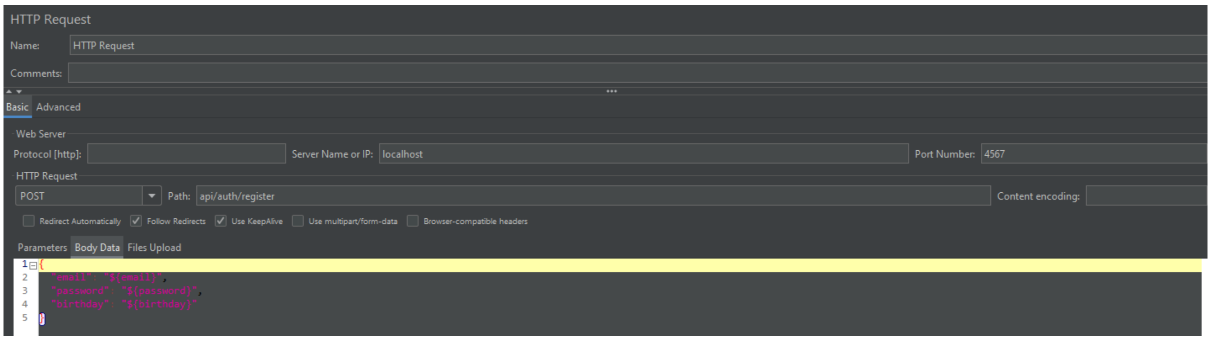
CSV Data Set Config

Name:	CSV Register Data Set Config
Comments:	

Configure the CSV Data Source

Filename:	C:/Users/chais/Downloads/apache-jmeter-5.5/user_data.csv
File encoding:	
Variable Names (comma-delimited):	email, password, birthday
Ignore first line (only used if Variable Names is not empty):	False
Delimiter (use '\t' for tab):	,
Allow quoted data?:	False
Recycle on EOF ?:	True
Stop thread on EOF ?:	False
Sharing mode:	All threads

HTTP Request:



HTTP Request

Name: HTTP Request

Comments:

Basic Advanced

Web Server

Protocol [http]: Server Name or IP: localhost Port Number: 4567

HTTP Request

POST Path: api/auth/register Content encoding:

[ ] Redirect Automatically [✓] Follow Redirects [✓] Use KeepAlive [ ] Use multipart/form-data [ ] Browser-compatible headers

Parameters **Body Data** Files Upload

```
{
  "email": "${email}",
  "password": "${password}",
  "birthday": "${birthday}"
}
```

Result:

<u>**Success rate:**</u> 100%

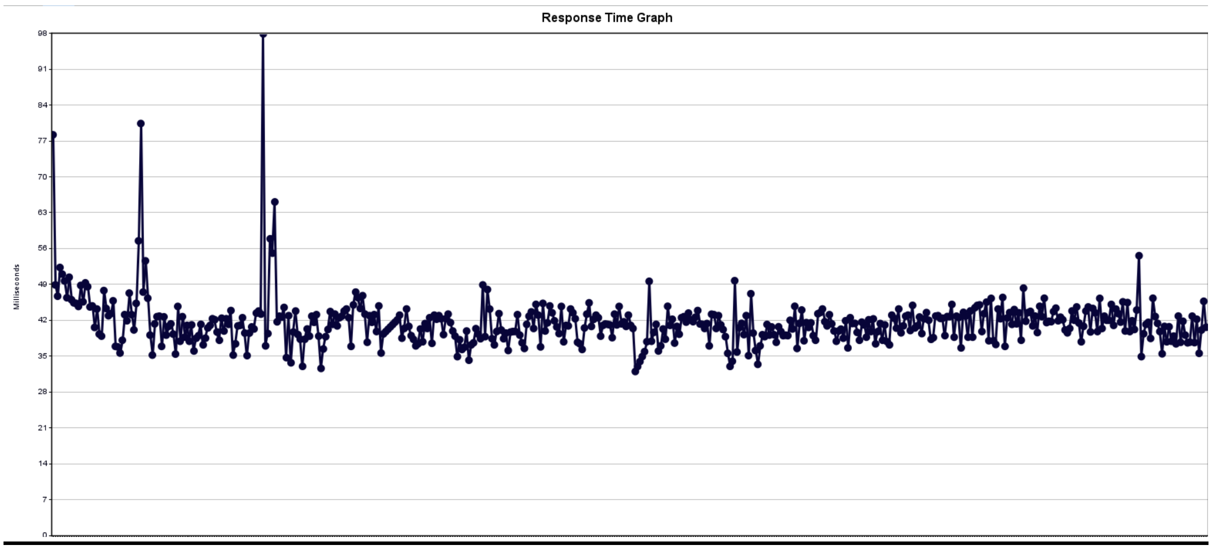
<u>**Average time:**</u> 41 ms.

<u>**Max time:**</u> 622 ms.

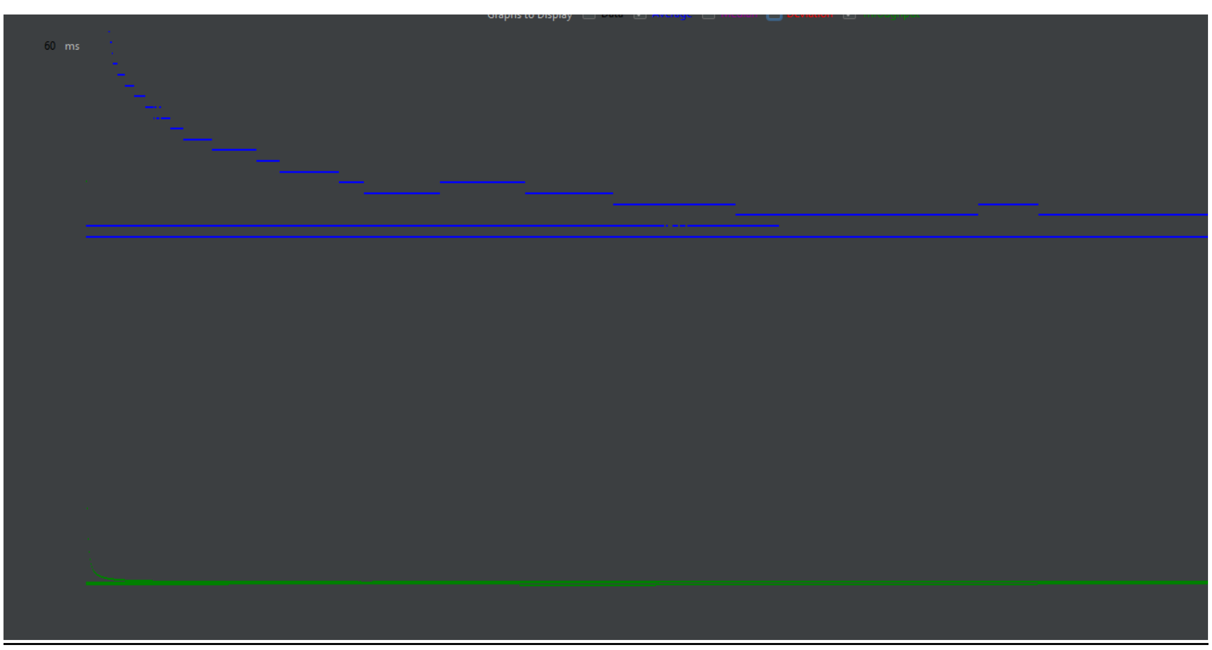
<u>**Min time:**</u> 23 ms.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %
HTTP Request	5000	41	23	622	12.88	0.00%
TOTAL	5000	41	23	622	12.88	0.00%

Response time graph:



Graph Results:



Conclusions:

- The number of registrations in our system is influenced by various factors such as user engagement, marketing efforts, and user demand, rather than being solely dependent on the data structure and memory allocation.
- However, a well-designed data structure with sufficient memory capacity can contribute to the system's scalability and efficient management of user data.
- Furthermore, it should be noted that the registration process within our system operates with notable efficiency, facilitating swift registration actions for users.

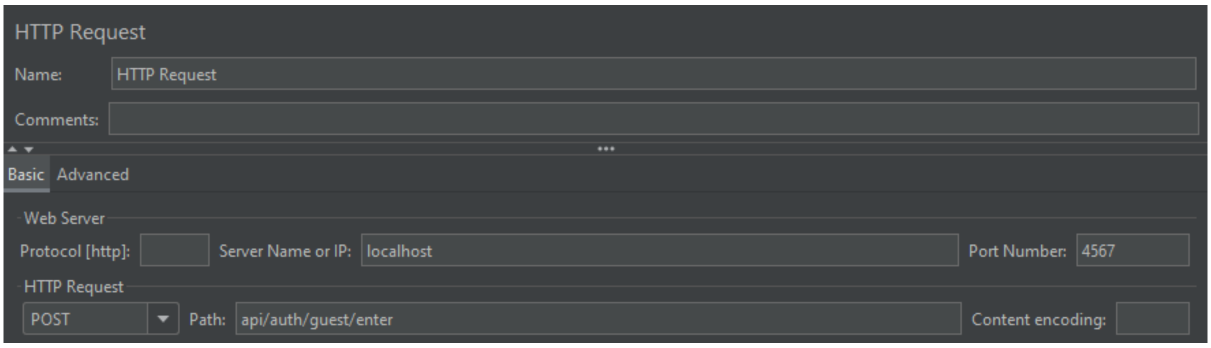
Index 3 – at any given moment:

Support for _____ visitors to the system at any given moment.

Scenario 1 – Guest enter to system:

Dealing with **2000** requests of enter guest to the system at the same time while meeting a response time of no more than a second per request.

HTTP Request:



HTTP Request

Name: HTTP Request

Comments:

Basic Advanced

Web Server

Protocol [http]: Server Name or IP: localhost Port Number: 4567

HTTP Request

POST Path: api/auth/guest/enter Content encoding:

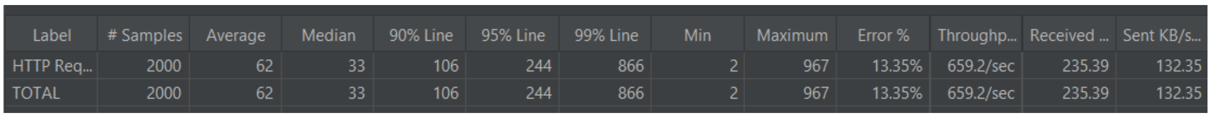
Result:

Success rate: 86.65%

Average time: 62 ms.

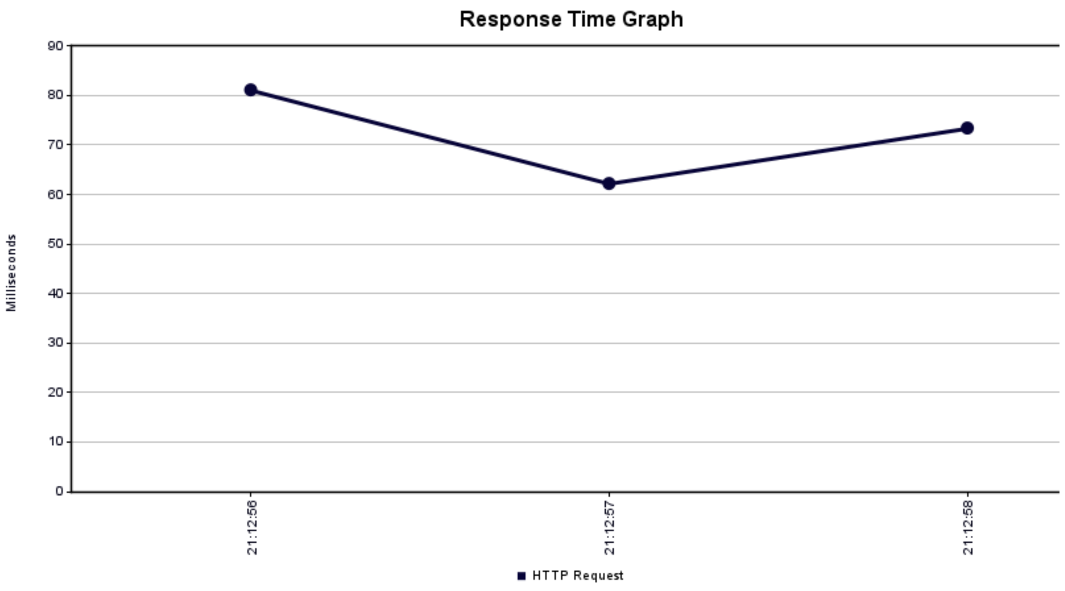
Max time: 976 ms.

Min time: 2 ms.



Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughp...	Received ...	Sent KB/s...
HTTP Req...	2000	62	33	106	244	866	2	967	13.35%	659.2/sec	235.39	132.35
TOTAL	2000	62	33	106	244	866	2	967	13.35%	659.2/sec	235.39	132.35

Response time graph:



Summary of Findings:

During the testing phase, we conducted three types of tests to assess the performance and functionality of our system across various operations. Due to time constraints and the anticipation of system changes, we were unable to perform an extensive number of tests as initially planned. However, based on the conducted tests, it is evident that the system performs exceptionally well, even surpassing expectations in extreme scenarios.

Key Observations:

- **Performance in Extreme Scenarios:** The system demonstrated robust performance in extreme scenarios, exceeding our expectations. It successfully handled demanding operations with efficiency and stability.
- **Limited Test Coverage:** Due to time limitations, the number of tests performed was restricted. However, the conducted tests provided valuable insights into the system's performance and identified areas for improvement.

Improvement Goals:

Based on the results and observations from the testing phase, our improvement goals are as follows:

- **Enhanced Scalability:** We aim to enable increased access and wider support for requests from the server in a second. This will ensure the system can handle a higher volume of concurrent requests efficiently.
- **Improved Response Time:** We will focus on optimizing the system's response time to the client. By enhancing performance, we aim to reduce latency and provide a more seamless user experience.
- **Expanded Functional Support:** We plan to extend the system's functionality to accommodate a broader range of requests and operations. This will enable users to leverage a wider set of features and capabilities.

Conclusion:

Although the number of tests conducted was limited, the system showcased exceptional performance, particularly in extreme scenarios. Moving forward, our focus will be on improving scalability, response time, and expanding functional support to provide an even more reliable and efficient system for our users.

We remain committed to refining and optimizing the system based on the lessons learned during the testing phase. By addressing the identified improvement goals, we are confident in delivering an enhanced user experience and meeting the evolving demands of our clients.