LAB – 2 Report -- Koushik Kumar Kamala

AWS Frontend:	ec2-3-213-4-33.compute-1.amazonaws.com
AWS Backend:	ec2-34-201-199-224.compute-1.amazonaws.com

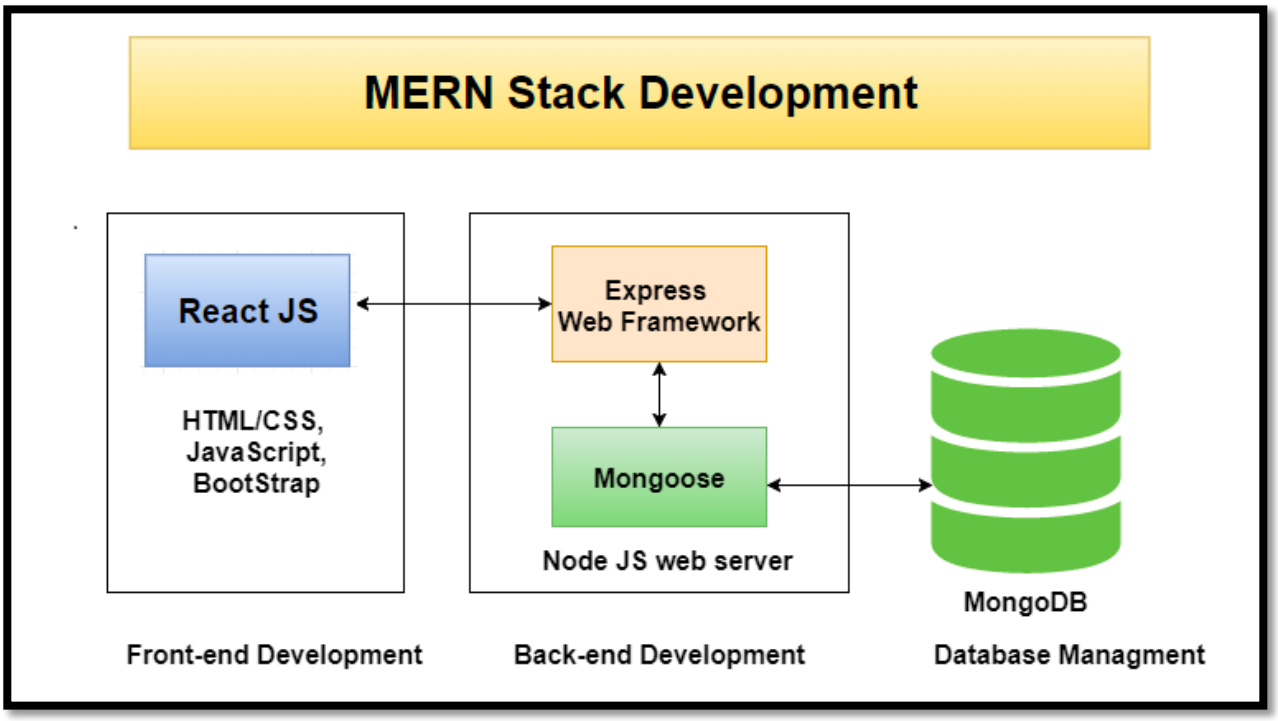
Canvas Application

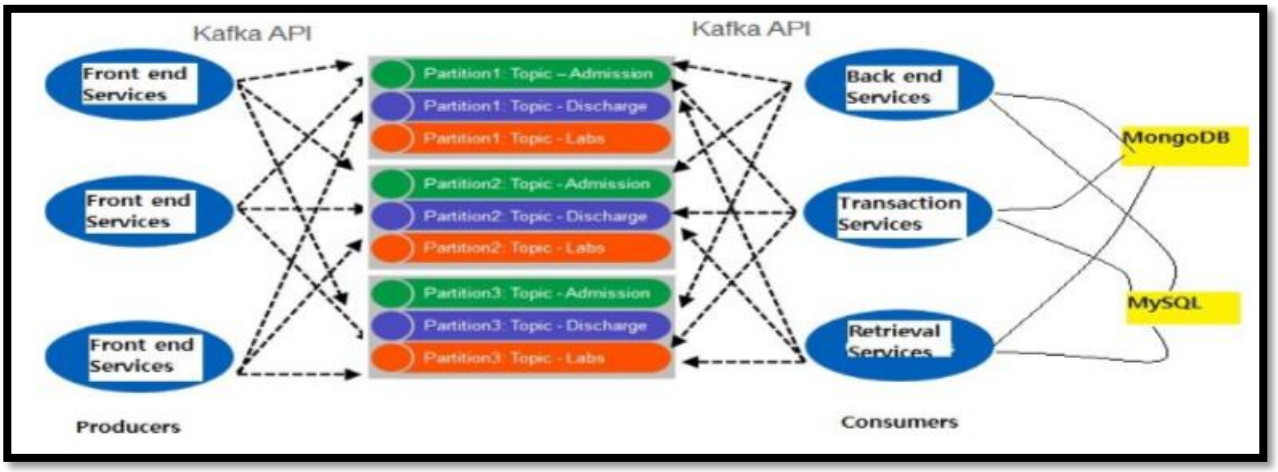
Introduction:

Goals: Design and implement the prototype of Canvas Application, using MERN stack i.e., MongoDB, Express.js, React and Node.js and Passport.js and JWT for authentication, Redux for state management, Kafka for messaging queues, provided this application is a student-faculty based app with features like registering for courses, dropping, requesting for permission number etc.

System Usecases: This application has two kinds of users mainly, i.e., Student and Professor. A student can search for the courses, register, drop, request for permission number, submit assignments, view their grades, contact professors, view class member, have a course dashboard, showing the courses, a side navigation bar to view calendar, files, courses and profile editing.

System Design:





Following is the operation flow,

1. Starting with a login page, build using JWT and passport, state being managed with ReactJS and redux. Once the user logs in, based on the type of the user, related dashboard is rendered. All the frond end code being build by react using npm modules.
2. As soon as a request is made, the same is sent to the backend, being build using node.js, where it uses the route module of the npm manager and routes the request based on the type of the request being made.
3. Kafka is the messaging queue, which communicates with the database and send the corresponding response back to the node which further send the response to the front end. And then using the axios calls' reponse, according the response is being reflected on to the browser
4. For this, we have used a MongoDB on cloud using mongodb atlas for storing the data where connection pooling is implemented automatically, and we have also used mongoose npm module for object data modeling. Once the application is built, the frontend is tested using enzyme and unit testing using Mocha

Functionality Flow:

Basically, two types of users, Professor and Student, where the professor has functionalities like create a course, drop students, grant permission, conduct online quizzes, grade them, communicate with the students.

Both the users have following functionalities,

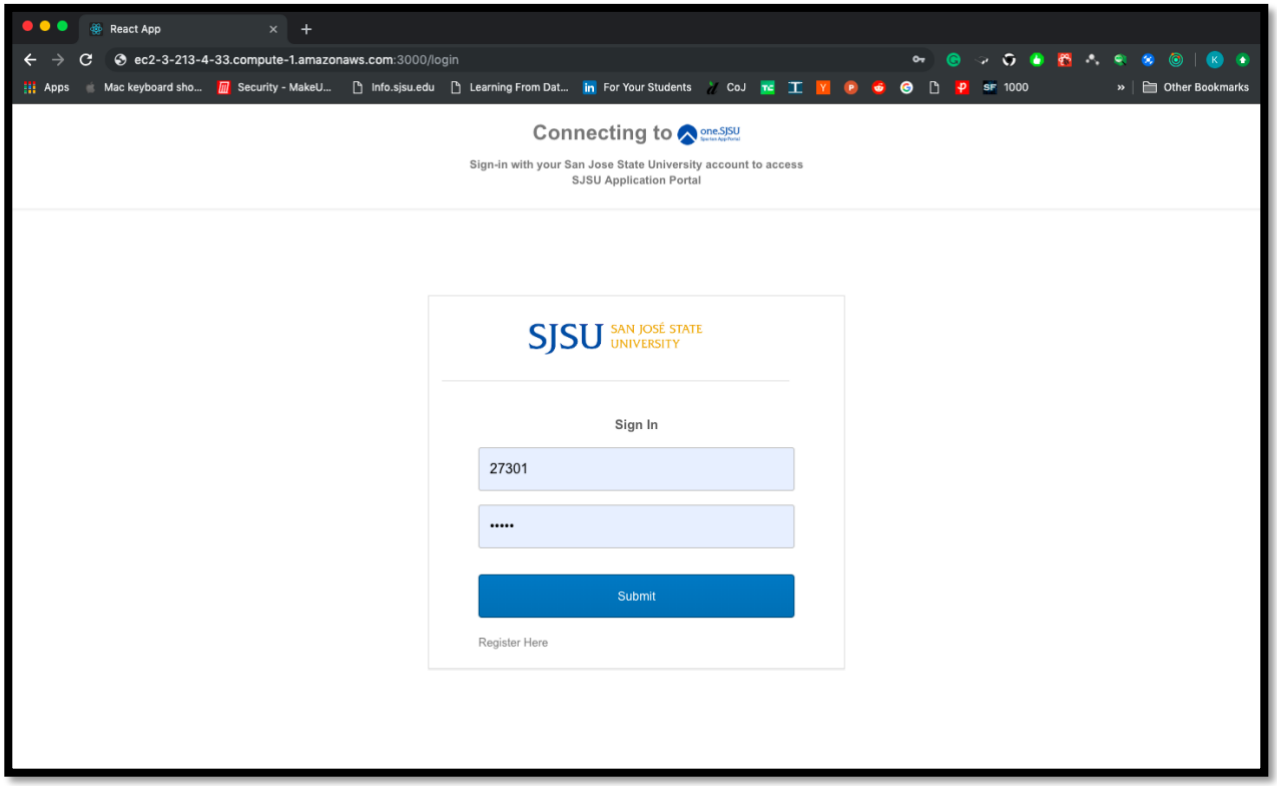
1. Signup new user
2. Signing in for existing users
3. Signout
4. Profile (Name, Contact, Email, Contact, About Me, City, Country, Company, School, Home towm, Languages, Gender)
5. Able to edit user profile data anytime
6. Faculty
    a. The Faculty able to create a course, with course specific details, required by the student and the course

b. Grant Permission number on request
c. Able to view all the courses created by the professor
d. Remove the student from a course.
e. Make an announcement

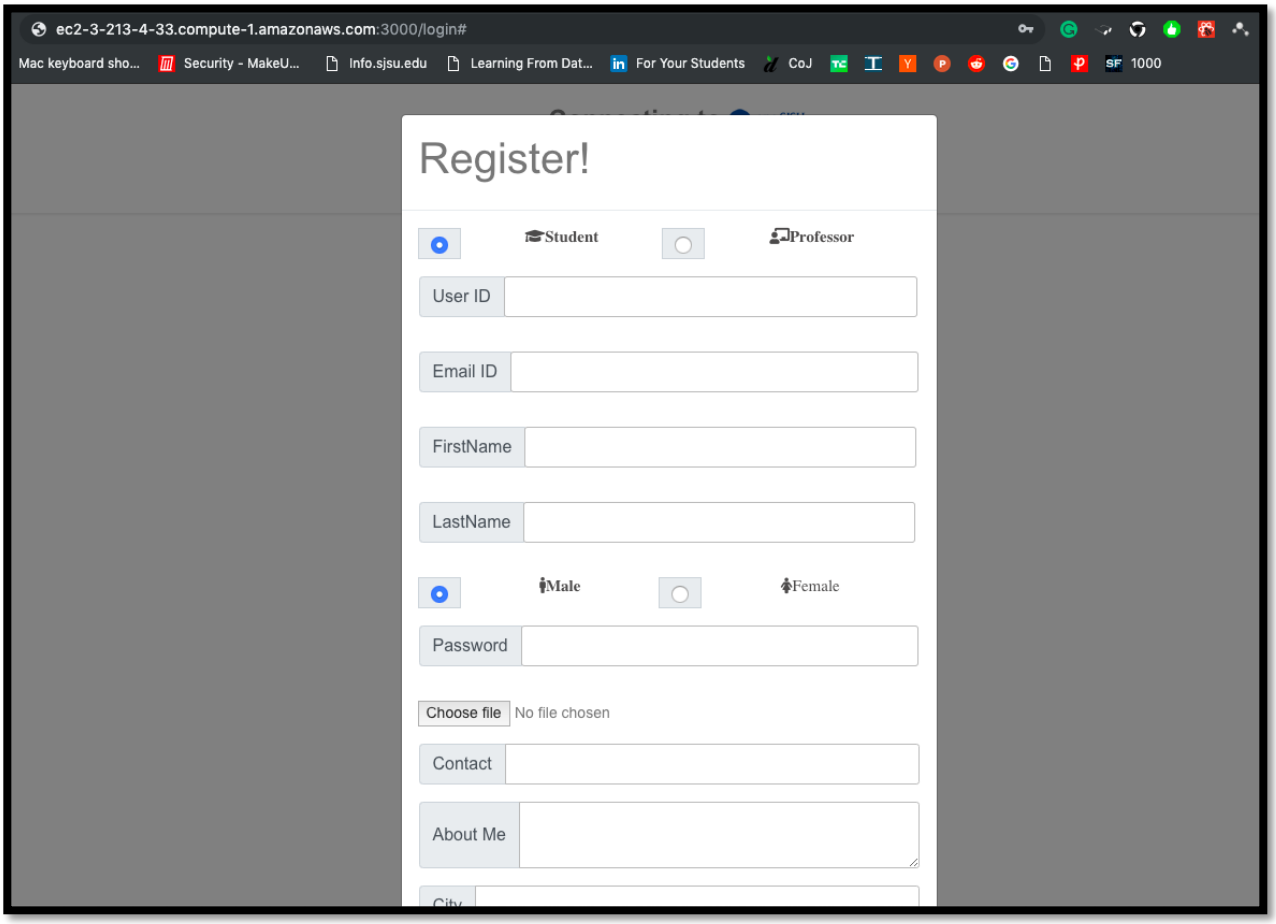
7. Student
    a. Student should be able to see all the courses in his dashboard
    b. Student able to submit the assignments, download the files.
8. Able to organize course cards on the dashboard
9. Students and faculty should be able to message with one another and display the messages under Inbox tab of sidebar.
10. Support the pagination for people search and course search.

Screenshots:

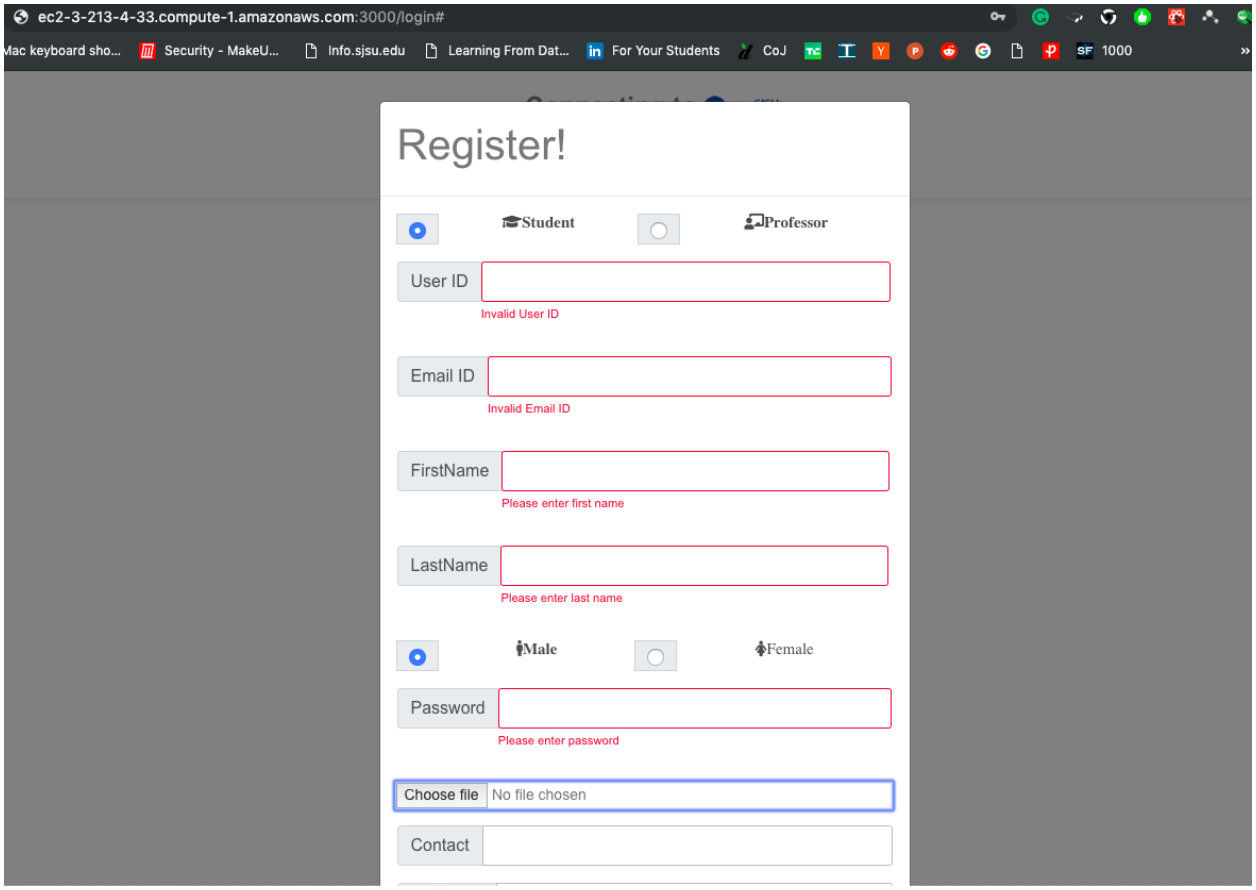
- Sign In



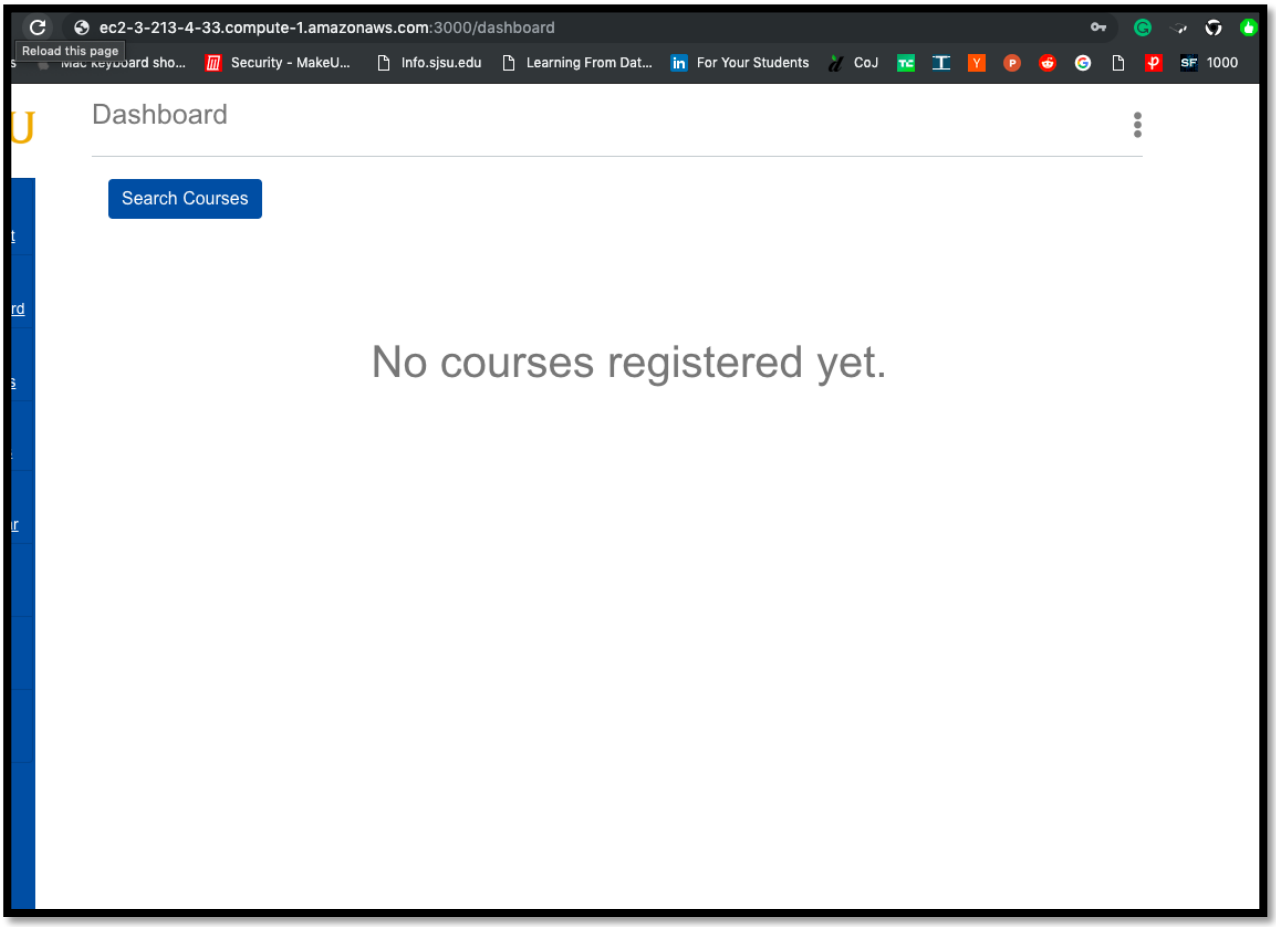
- Sign Up and validation



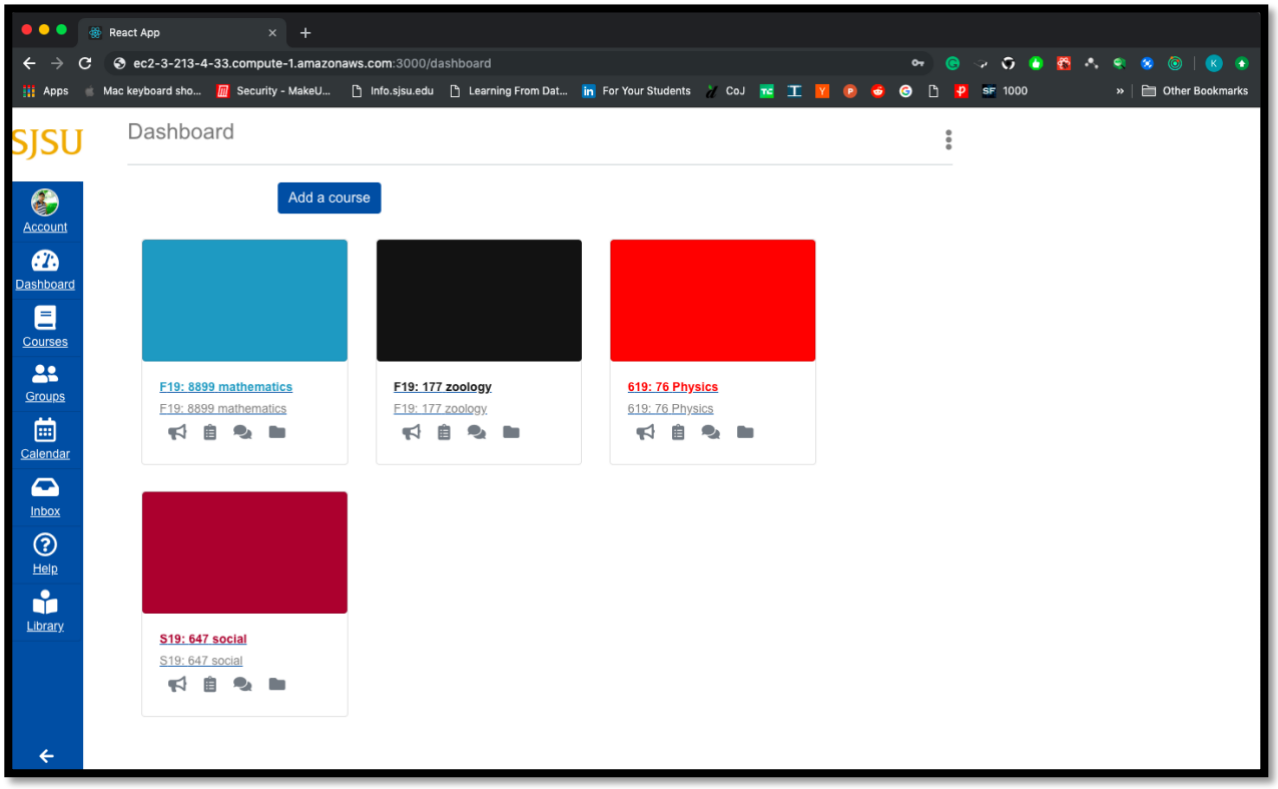
- Sign Up and validation



- No Courses message – and search courses button

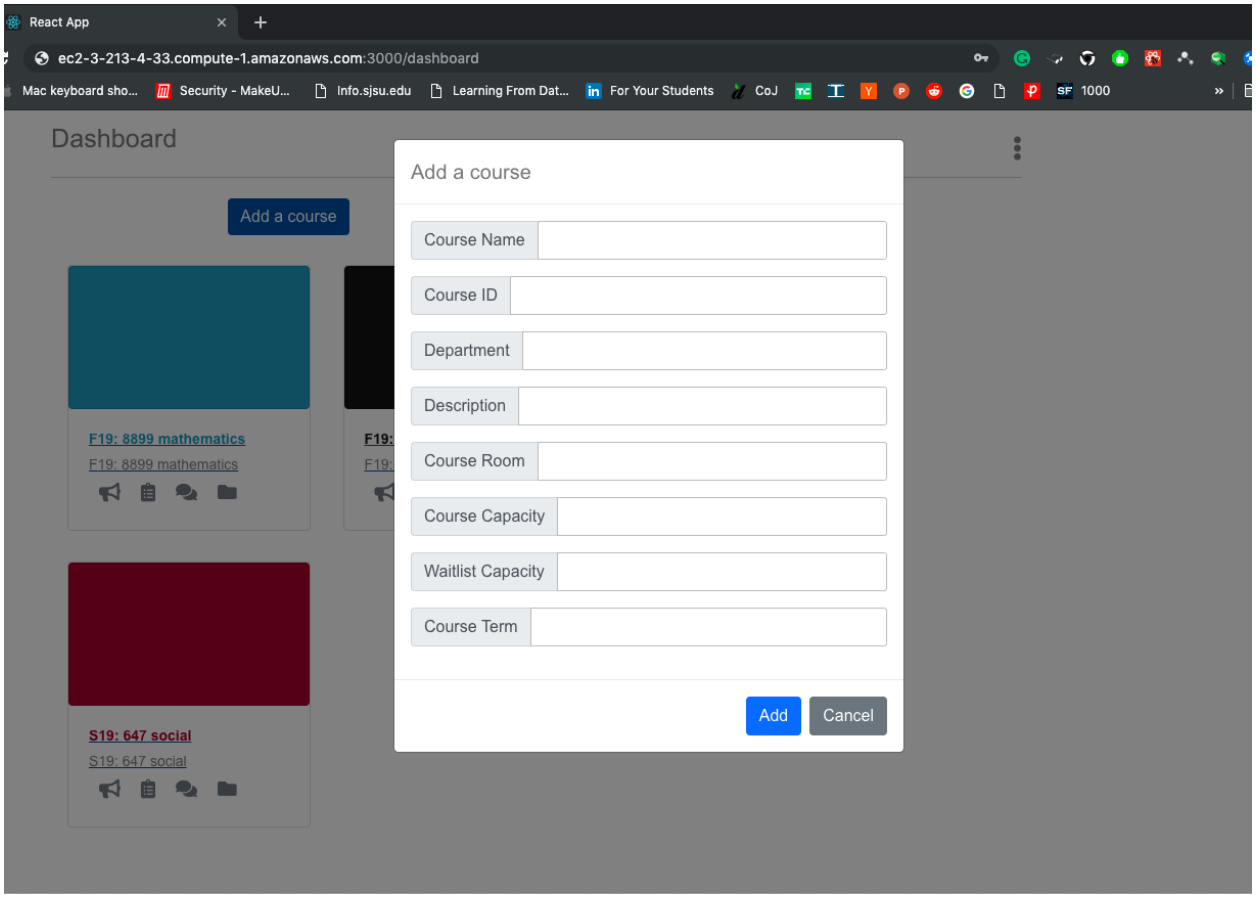


- Dashboard – Professor/ student viewing courses

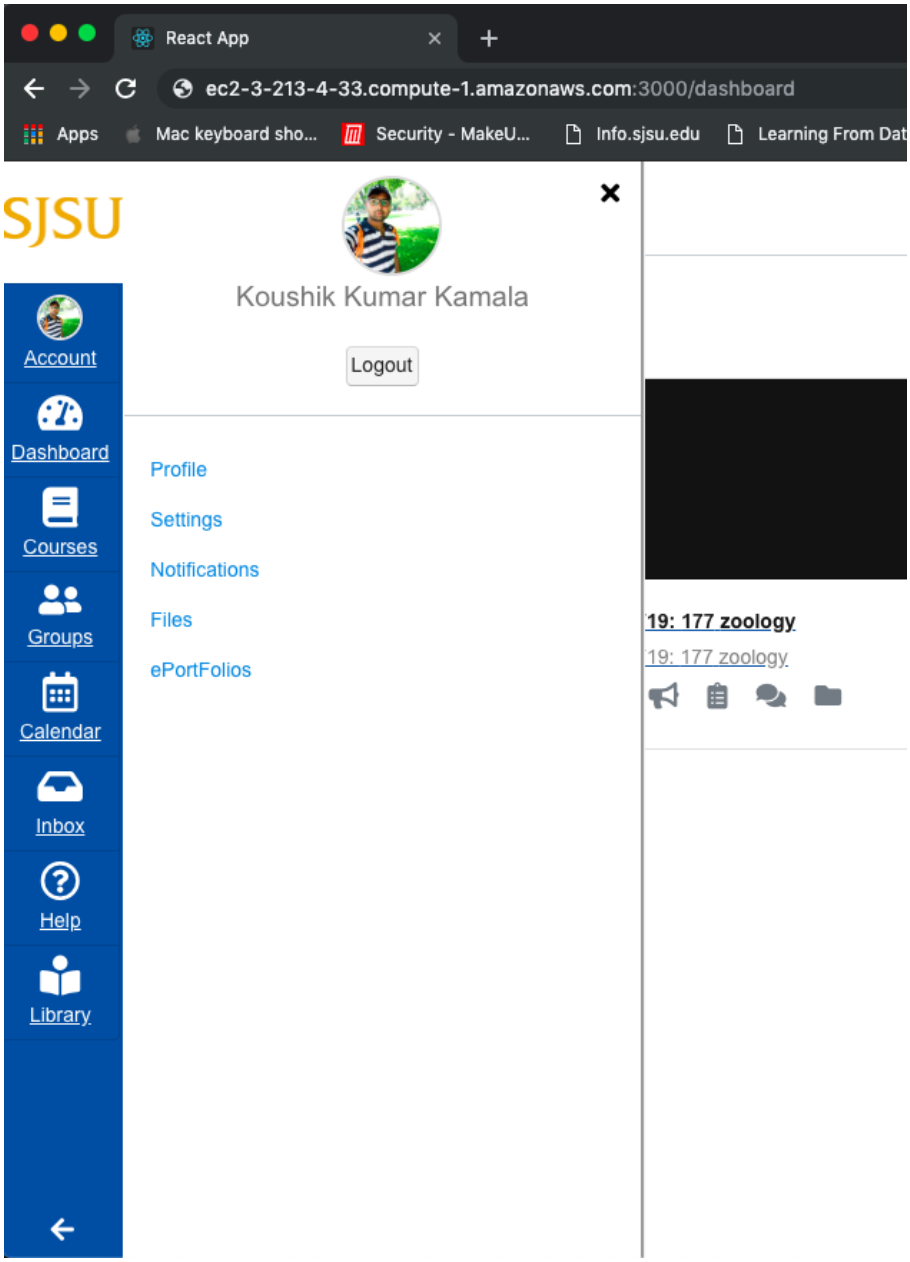


Add course button visible only to the Professor

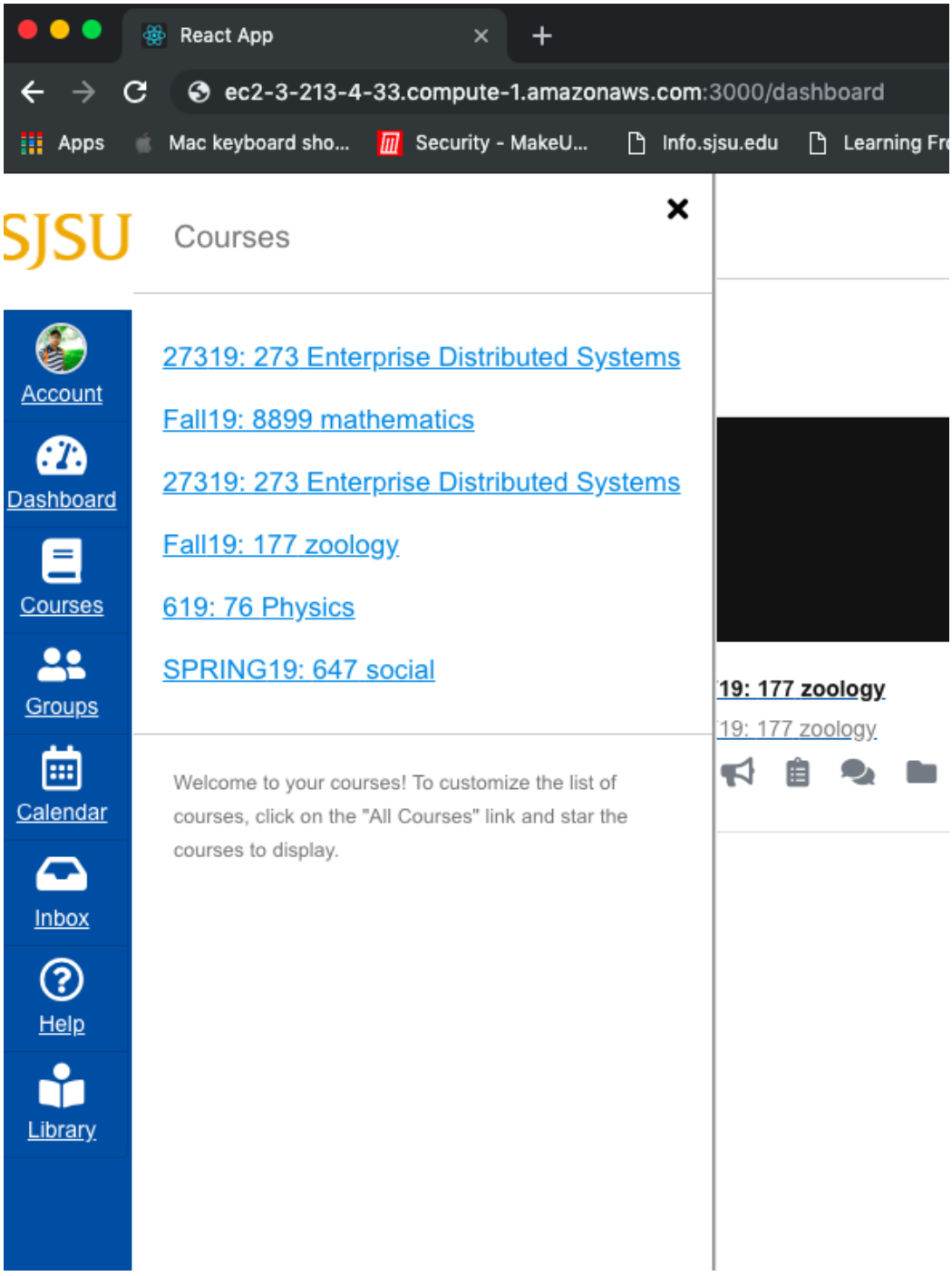
- Add a new course



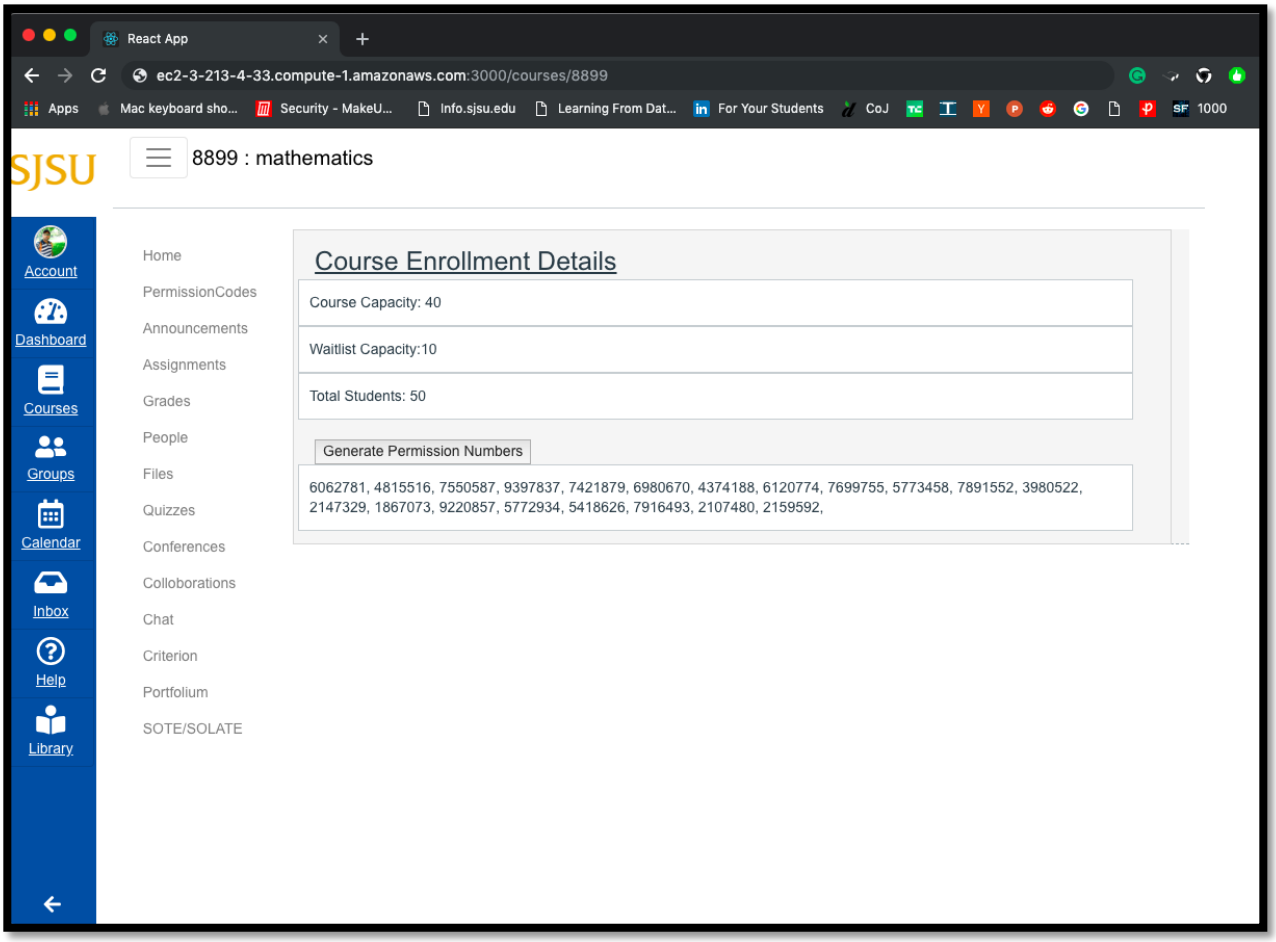
- Sidebar Navigation and Profile Tray



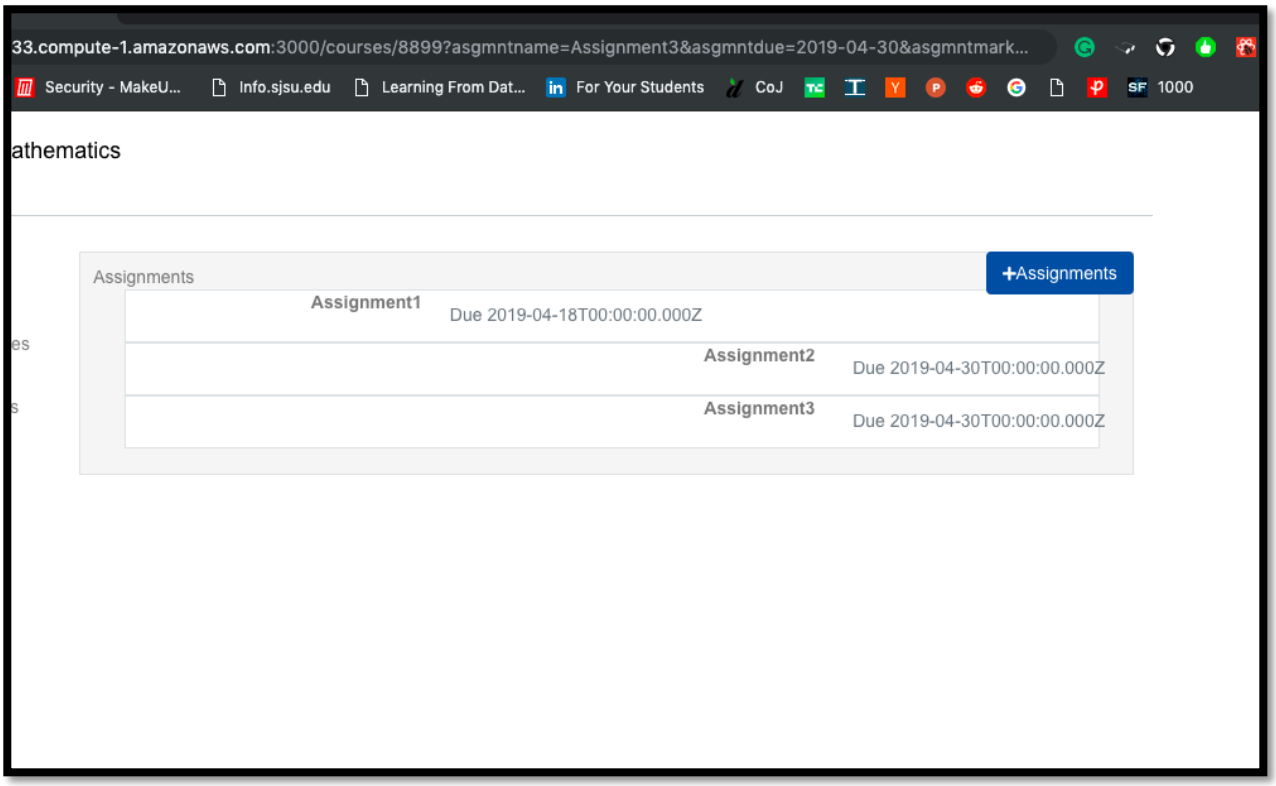
- Sidebar Navigation and Courses Tray



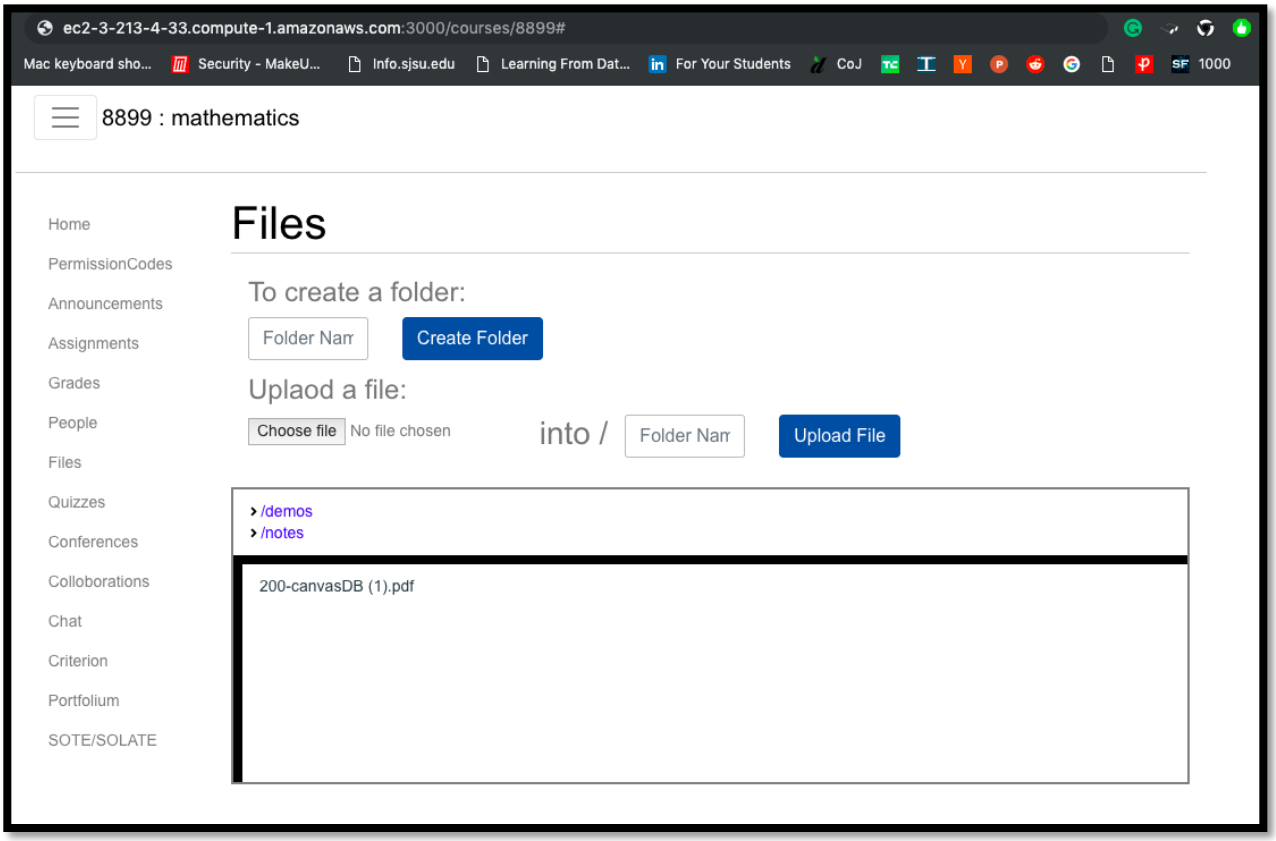
- Generate Permission Numbers by Professor



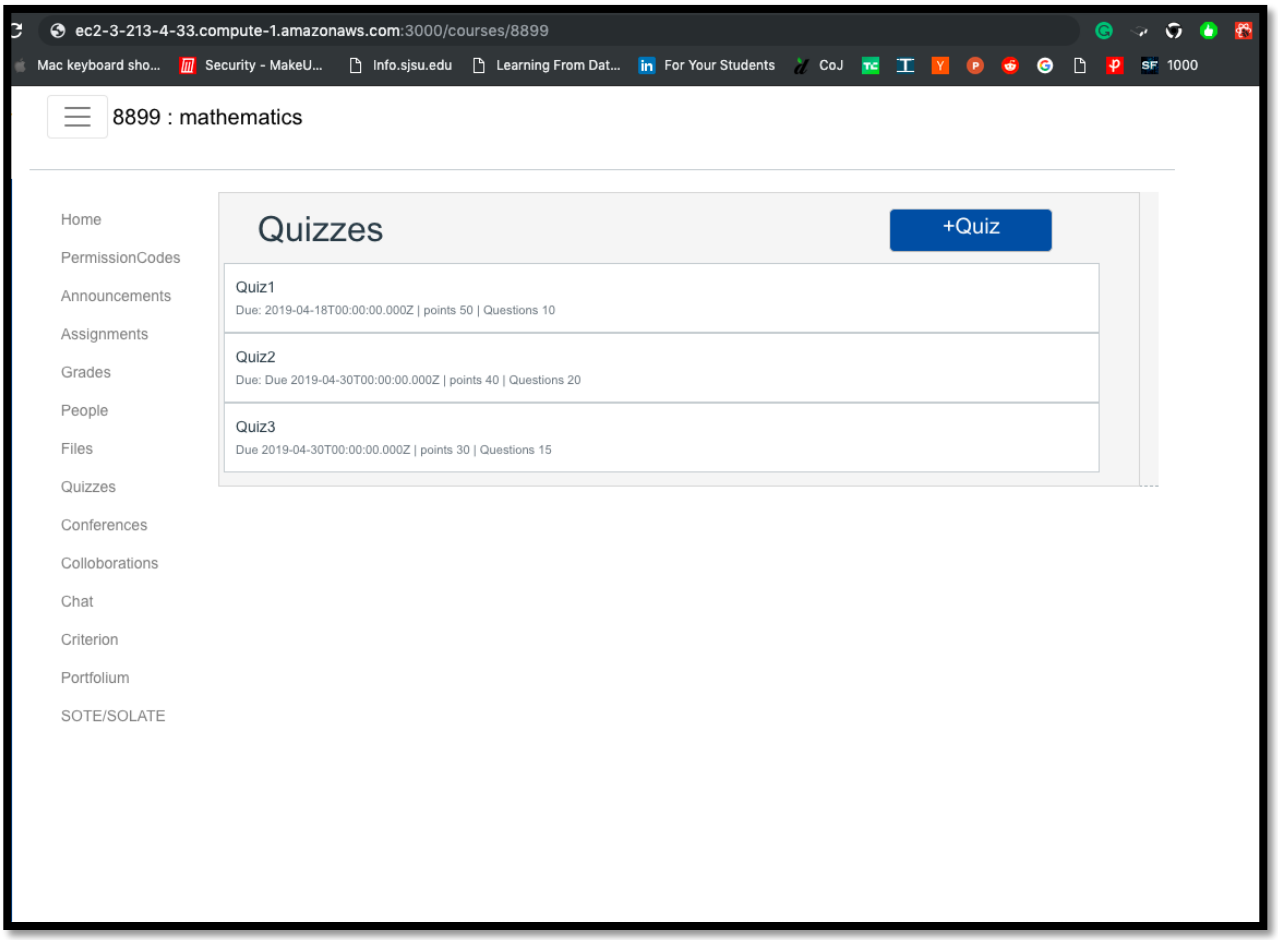
- Assignment View for a course



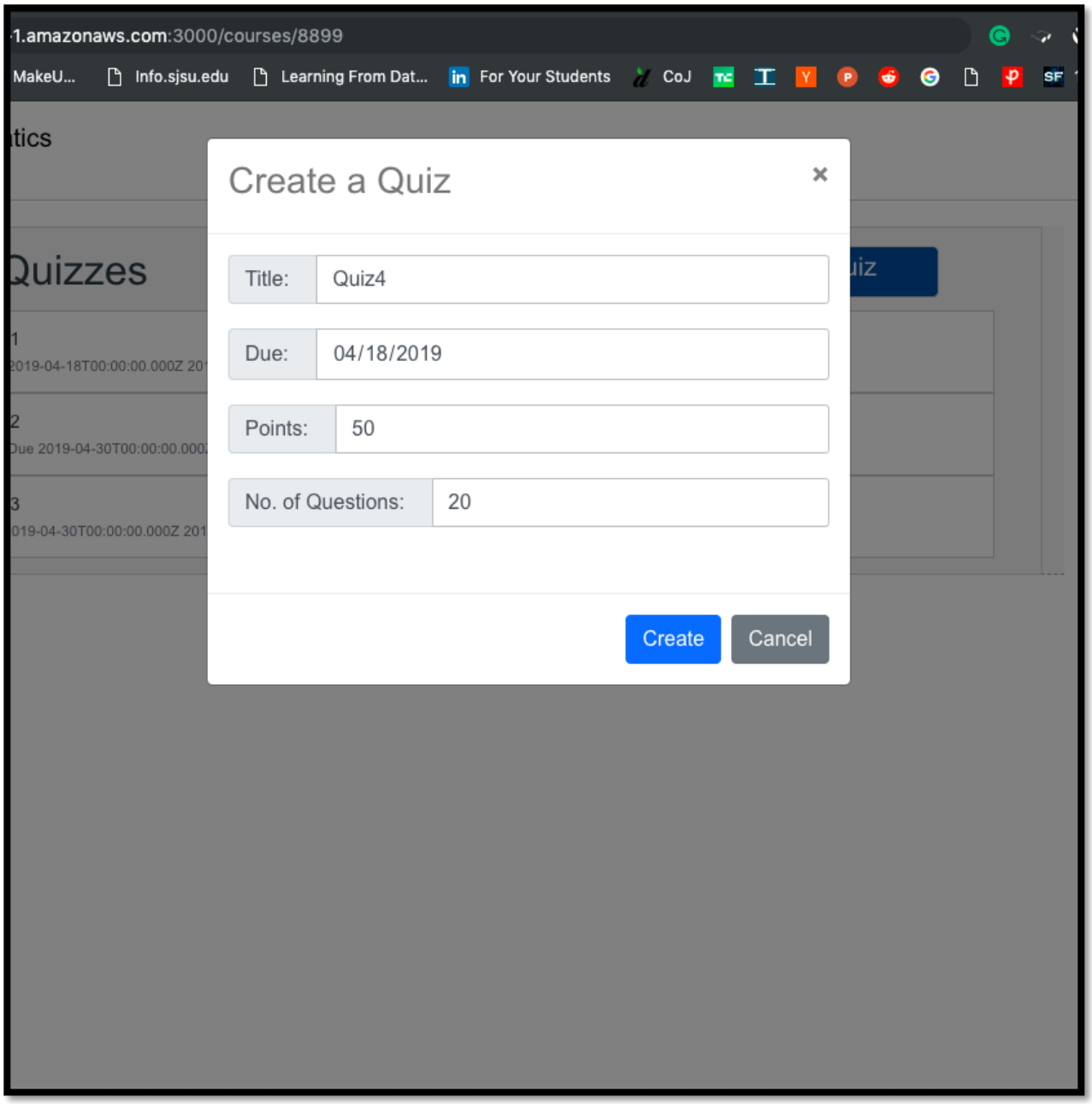
- Files Upload



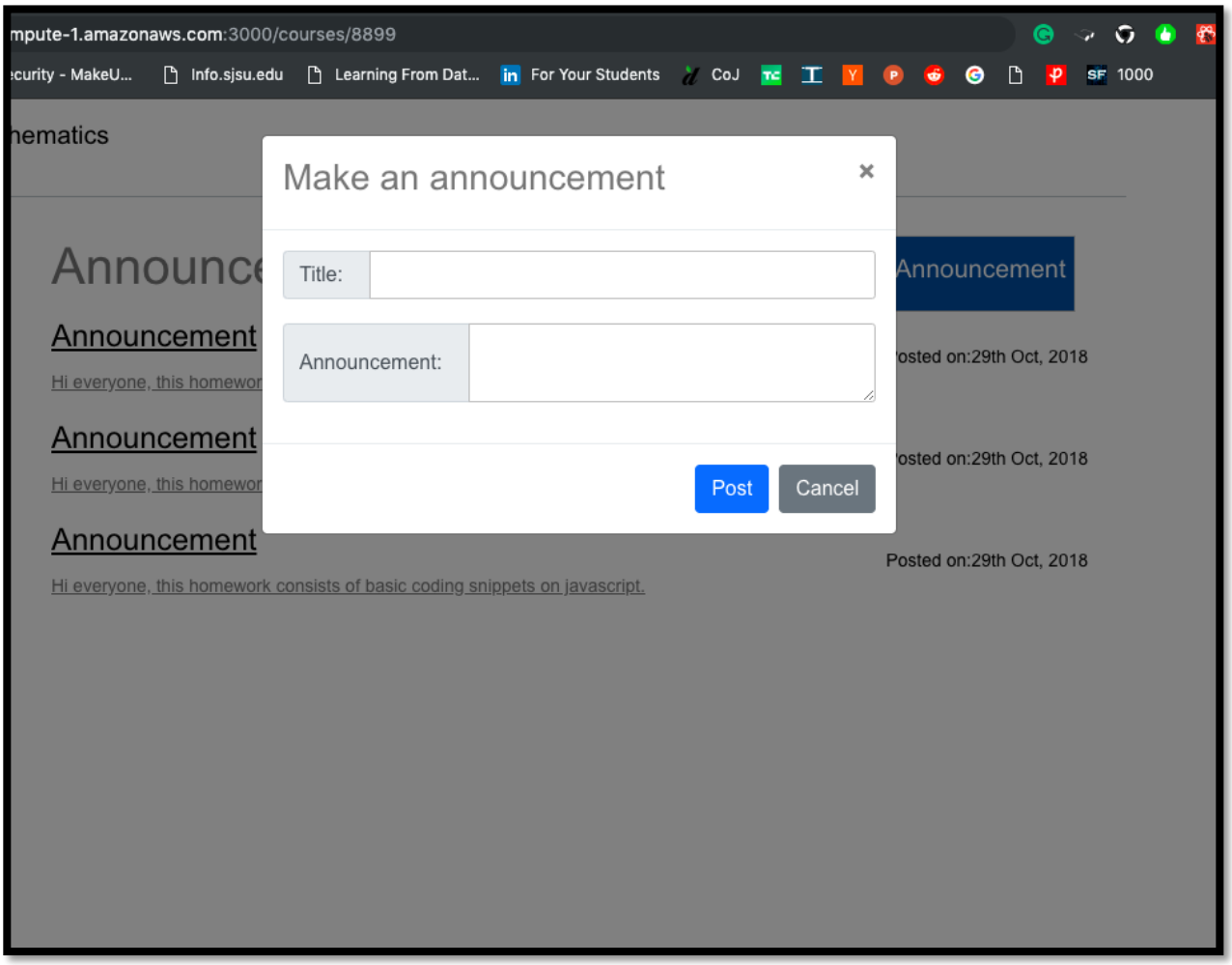
- Quizzes - list

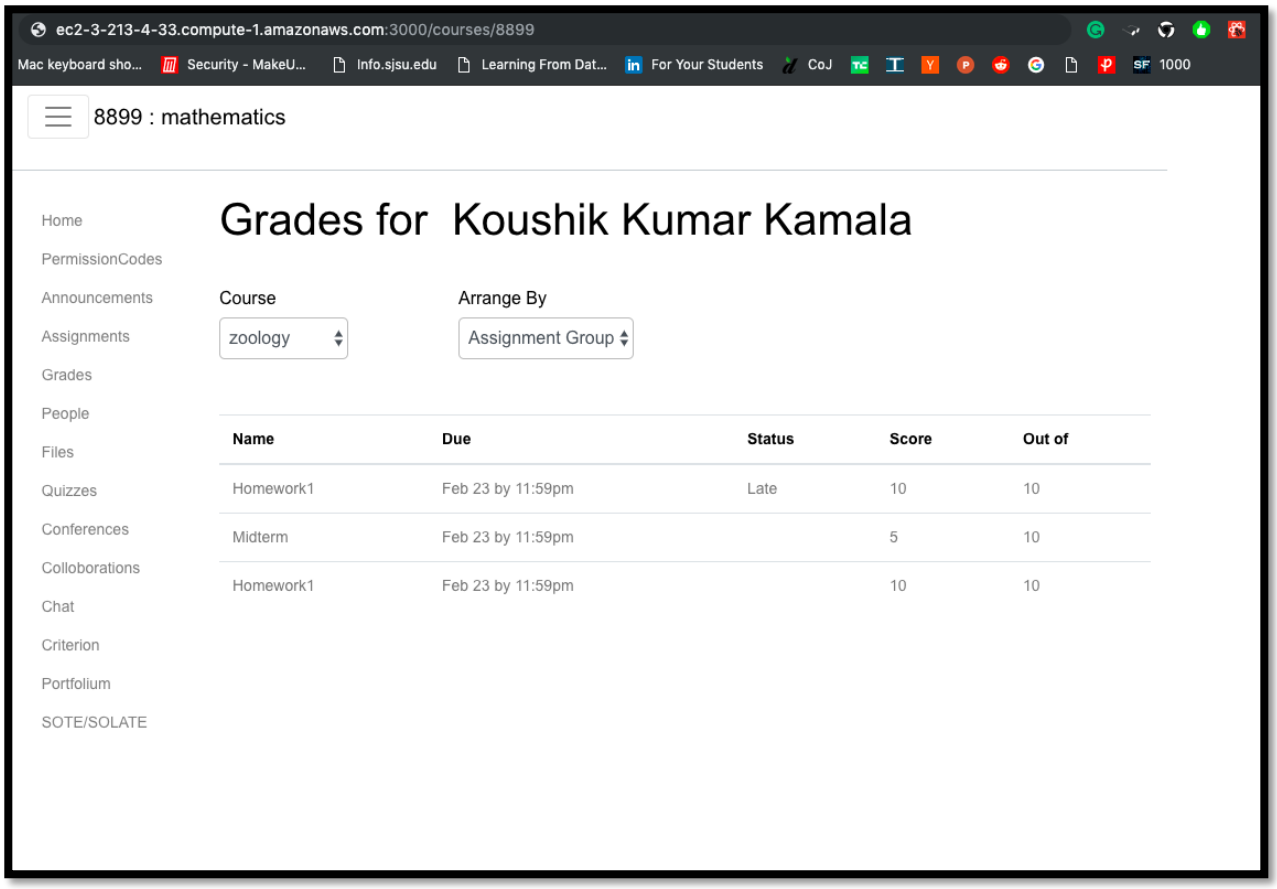


- Quizzes – Add a quiz

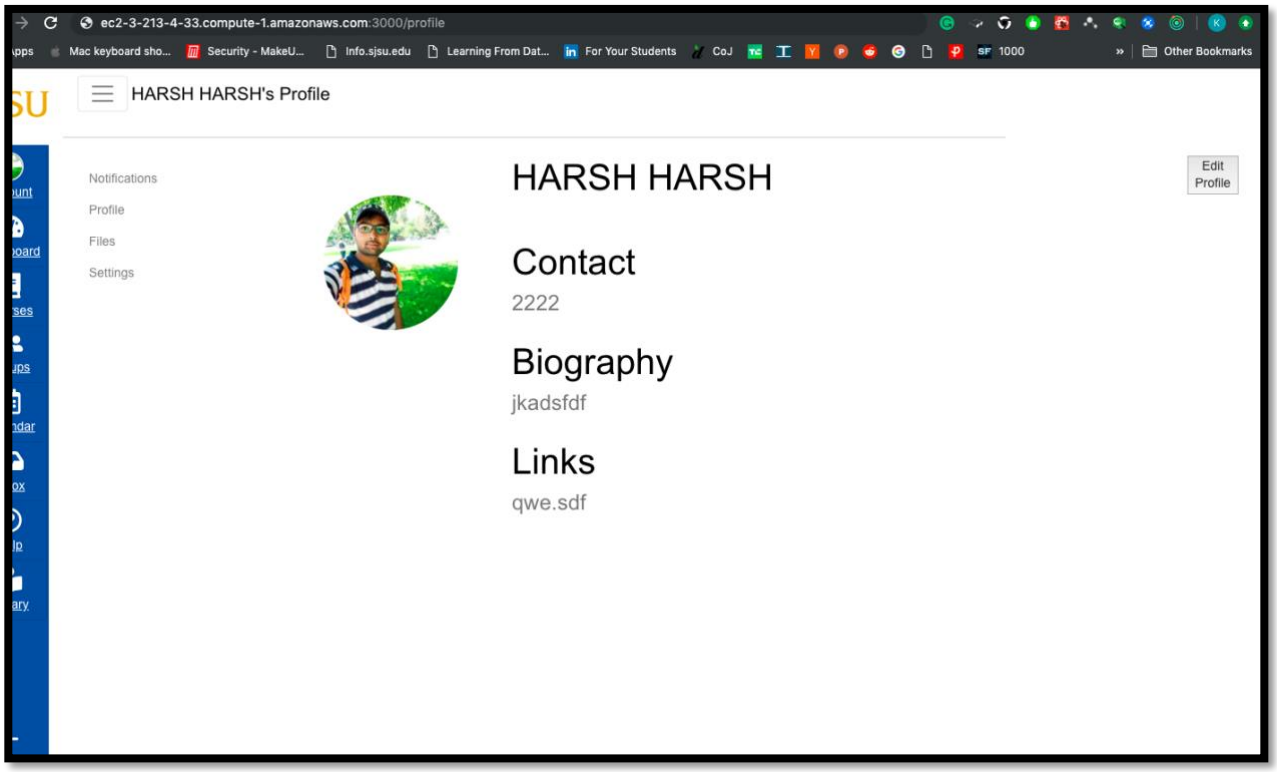


- Announcements – Add a announcement

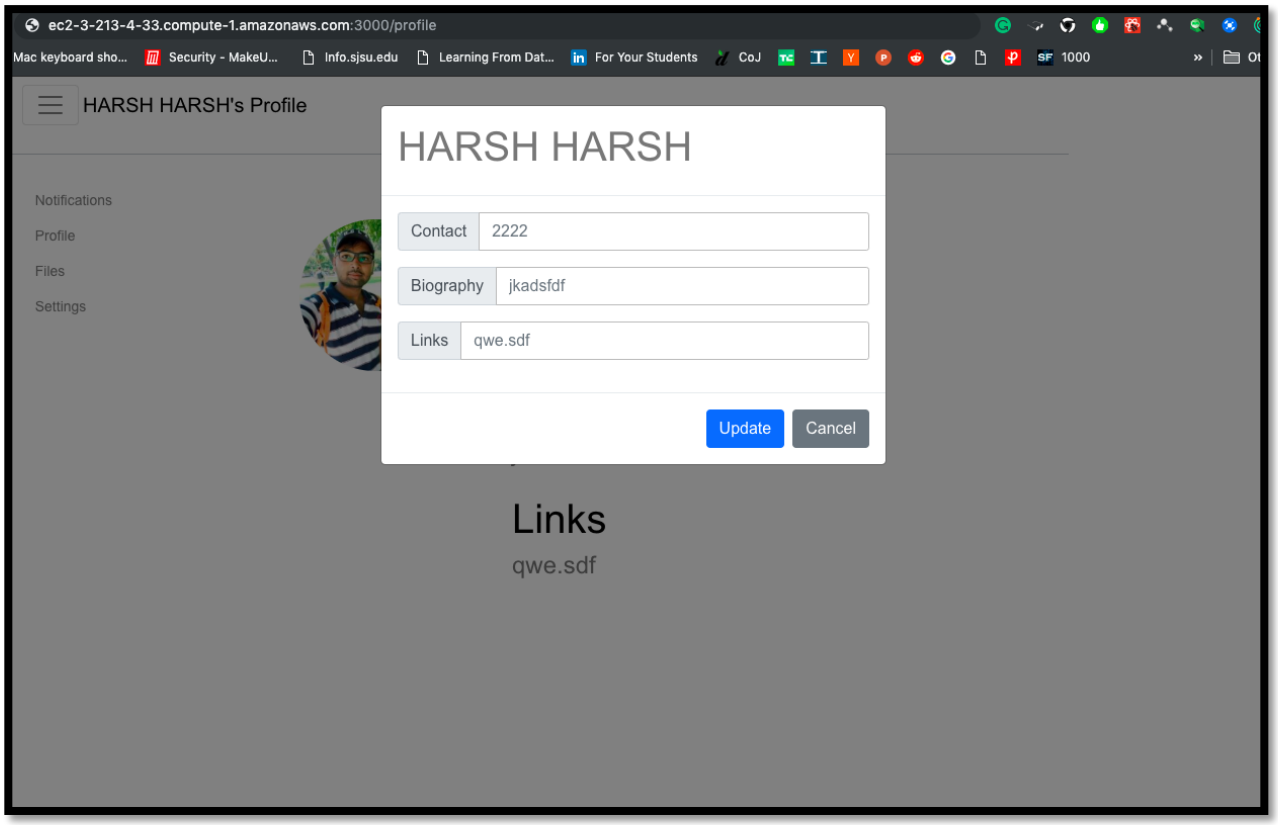



ec2-3-213-4-33.compute-1.amazonaws.com:3000/courses/8899
Mac keyboard sho...
Security - MakeU...
Info.sjsu.edu
Learning From Dat...
For Your Students
CoJ
1000
8899 : mathematics
Home
PermissionCodes
Announcements
Assignments
Grades
People
Files
Quizzes
Conferences
Colloborations
Chat
Criterion
Portfolium
SOTE/SOLATE
Grades for Koushik Kumar Kamala
Course
zoology
Arrange By
Assignment Group
Name
Due
Status
Score
Out of
Homework1
Feb 23 by 11:59pm
Late
10
10
Midterm
Feb 23 by 11:59pm
5
10
Homework1
Feb 23 by 11:59pm
10
10

- Profile View



- Profile Edit

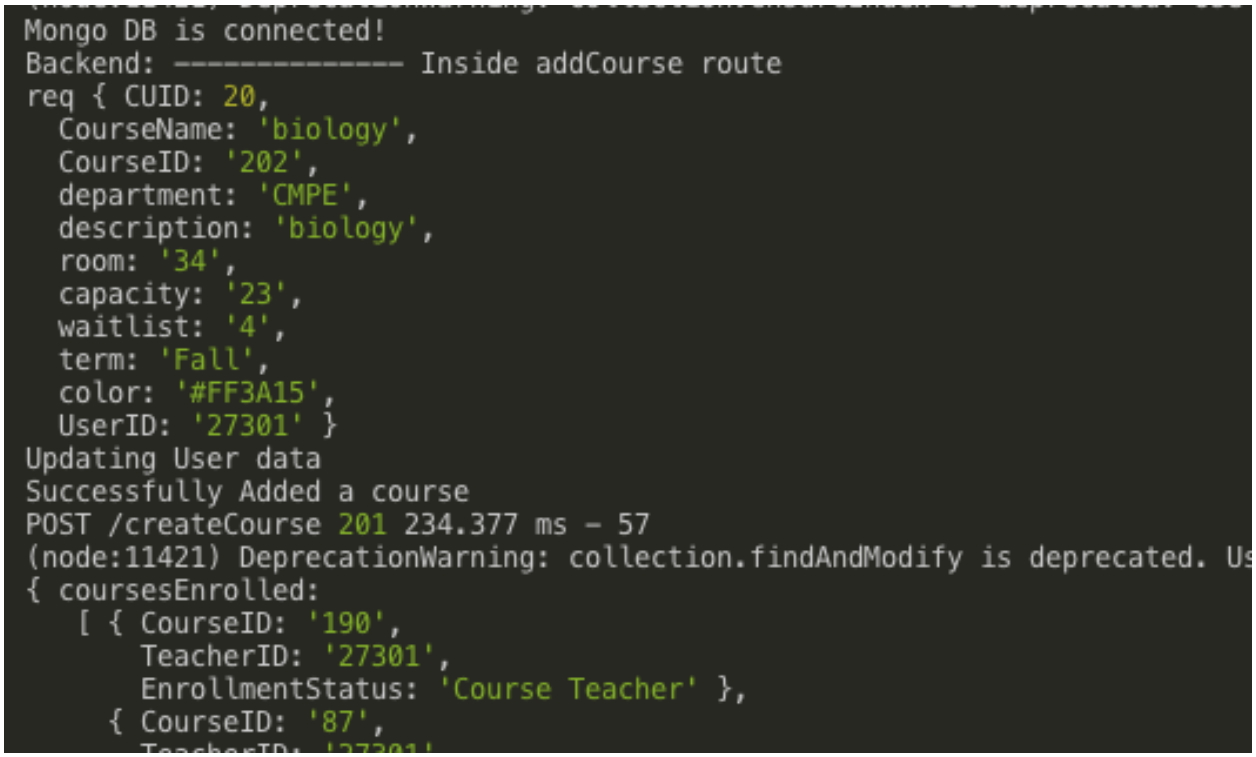


- Login – Backend console

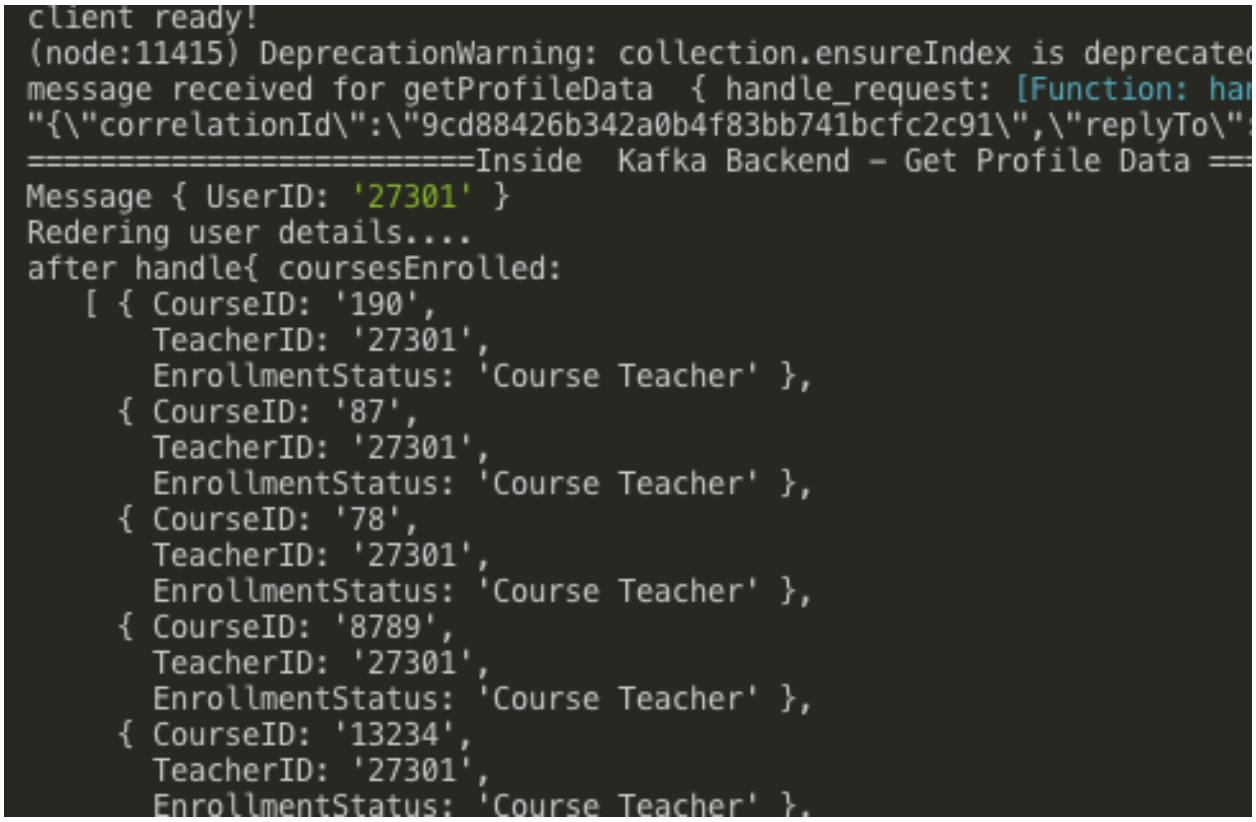
```
Mongo DB is connected!
BACKEND: ------- INSDIE LOGIN POST
Request Body:  { UserID: '27301', password: '27301' }
in make request
{ UserID: '27301', password: '27301' }
1
returning next
in response
in response1
true
========================Inside Backend - Get Courses ========================
Request Body:  { UserID: '27301', password: '27301' }
in make request
{ UserID: '27301', password: '27301' }
in response
in response1
true
client ready!
in response2
{ login: { '0': 70 } }
in response2
{ getCourses: { '0': 40 } }
msg received
response { coursesEnrolled:
   [ { CourseID: '190',
       TeacherID: '27301',
       EnrollmentStatus: 'Course Teacher' },
     { CourseID: '87',
       TeacherID: '27301',
       EnrollmentStatus: 'Course Teacher' },
     { CourseID: '78',
       TeacherID: '27301',
       EnrollmentStatus: 'Course Teacher' },
     { CourseID: '8789',
       TeacherID: '27301',
       EnrollmentStatus: 'Course Teacher' },
     { CourseID: '13234',
```

- Login – Backend console

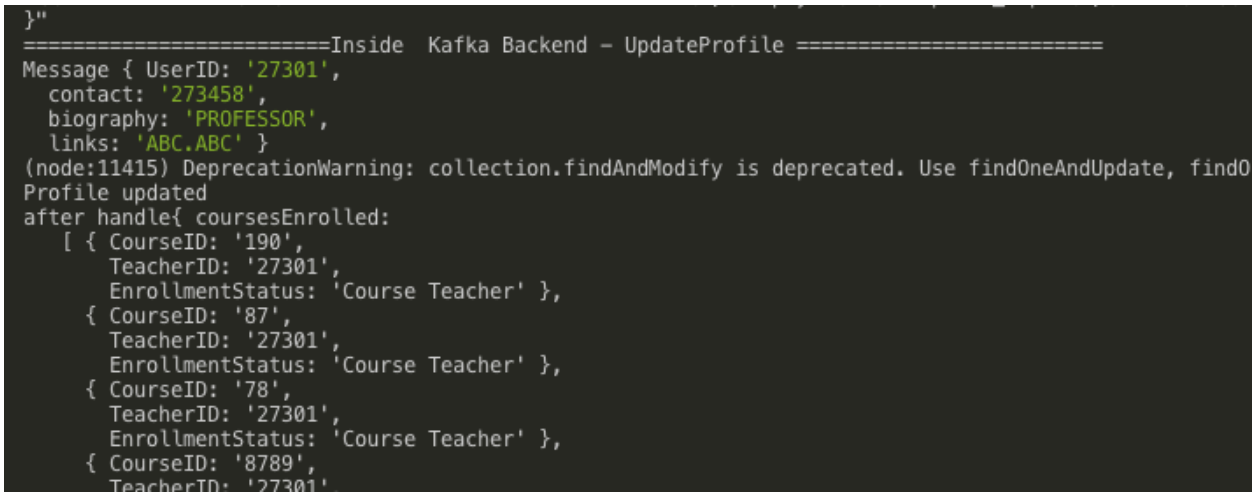
- Add a course – Backend console



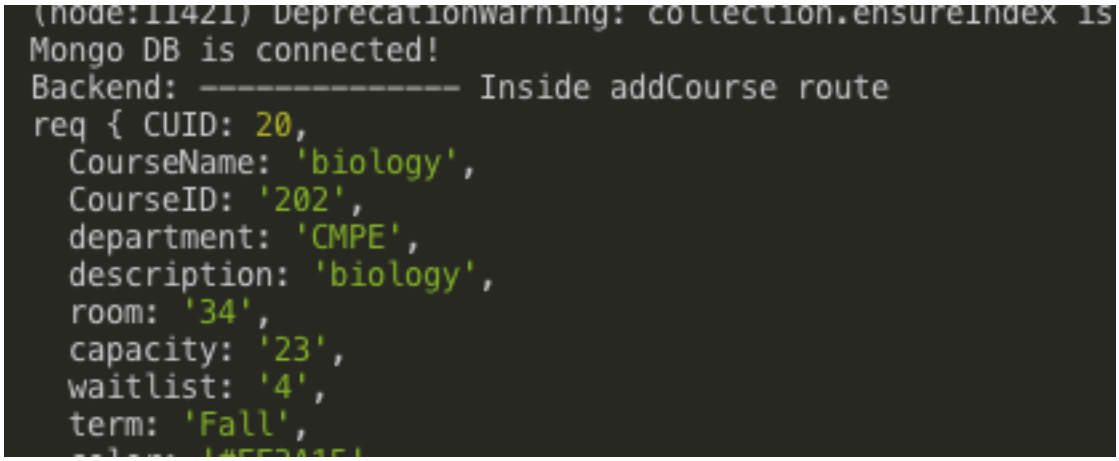
- Get Profile Data – Kafka console



- Update Profile Data – Kafka console



- Add a course – Kafka console



```
Successfully Added a course
POST /createCourse 201 234.377 ms - 57
(node:11421) DeprecationWarning: collection.findAndModify
{ coursesEnrolled:
   [ { CourseID: '190',
       TeacherID: '27301',
       EnrollmentStatus: 'Course Teacher' },
     { CourseID: '87',
       TeacherID: '27301',
       EnrollmentStatus: 'Course Teacher' },
     { CourseID: '78',
       TeacherID: '27301',
       EnrollmentStatus: 'Course Teacher' },
     { CourseID: '8789',
       TeacherID: '27301',
       EnrollmentStatus: 'Course Teacher' },
```

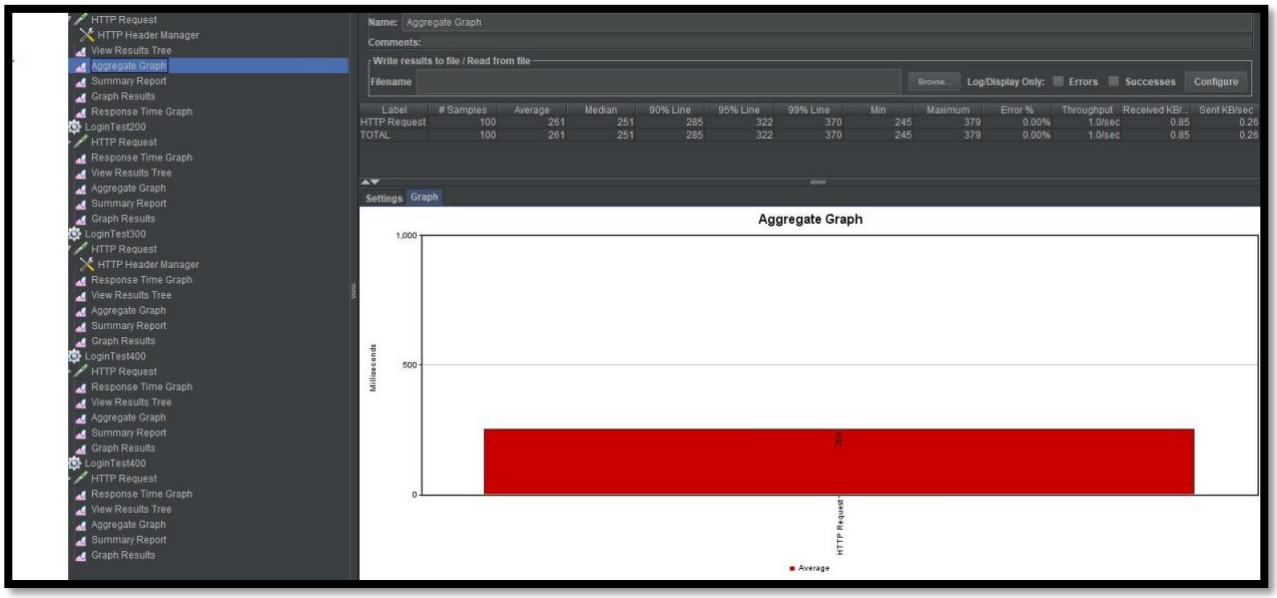
- Pagination

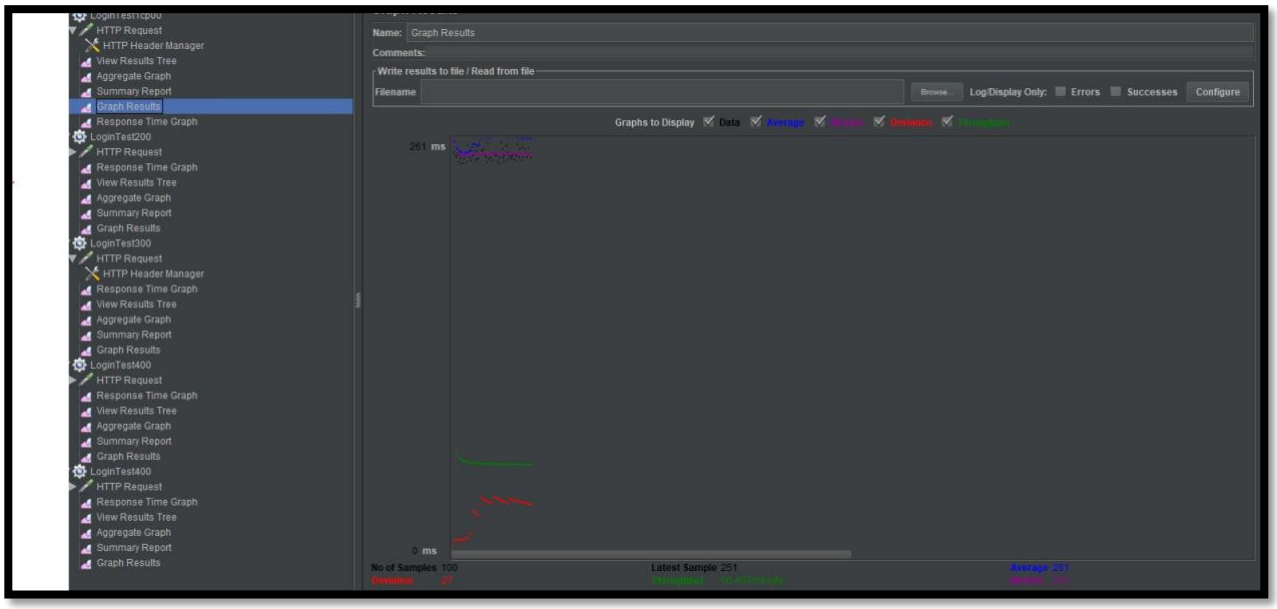

People
Student Name
Student Id
ranjith
443
vinay
444
Previous
Next

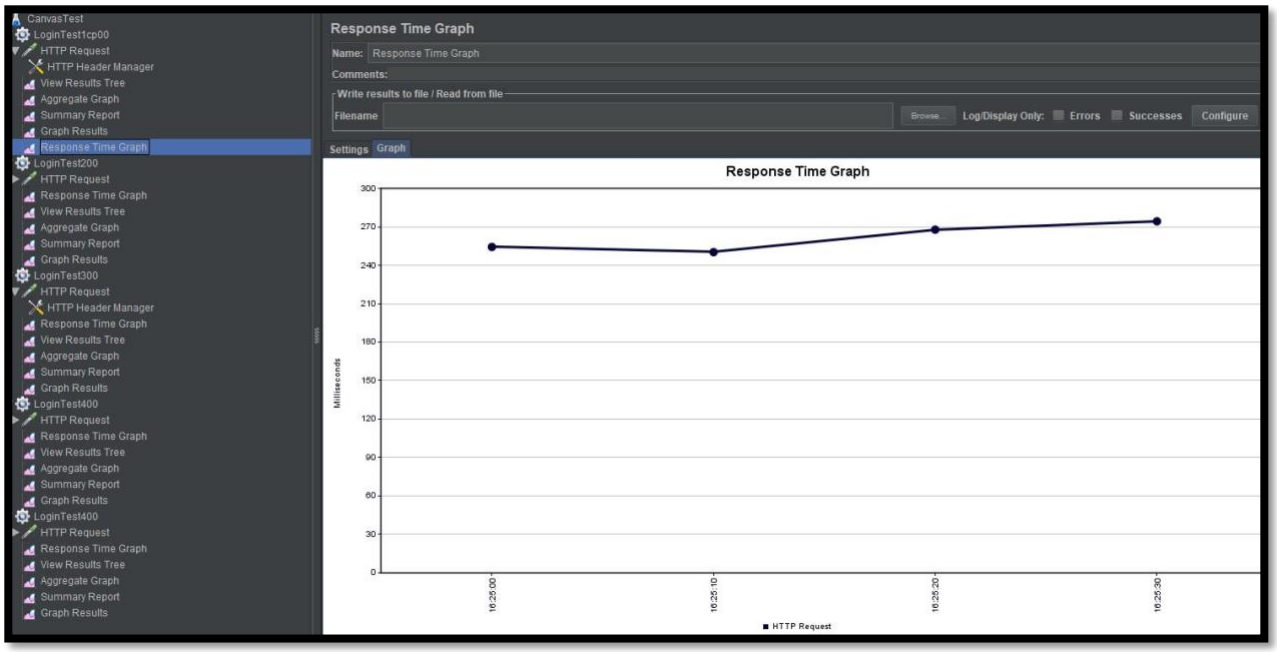

People
Student Name
Student Id
99
99
Previous
Next

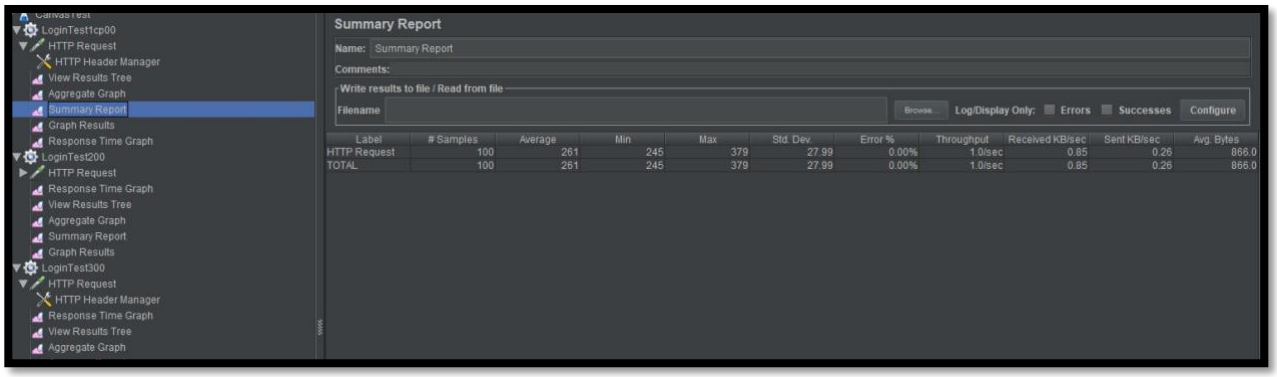
- JMeter

  For 100 calls,

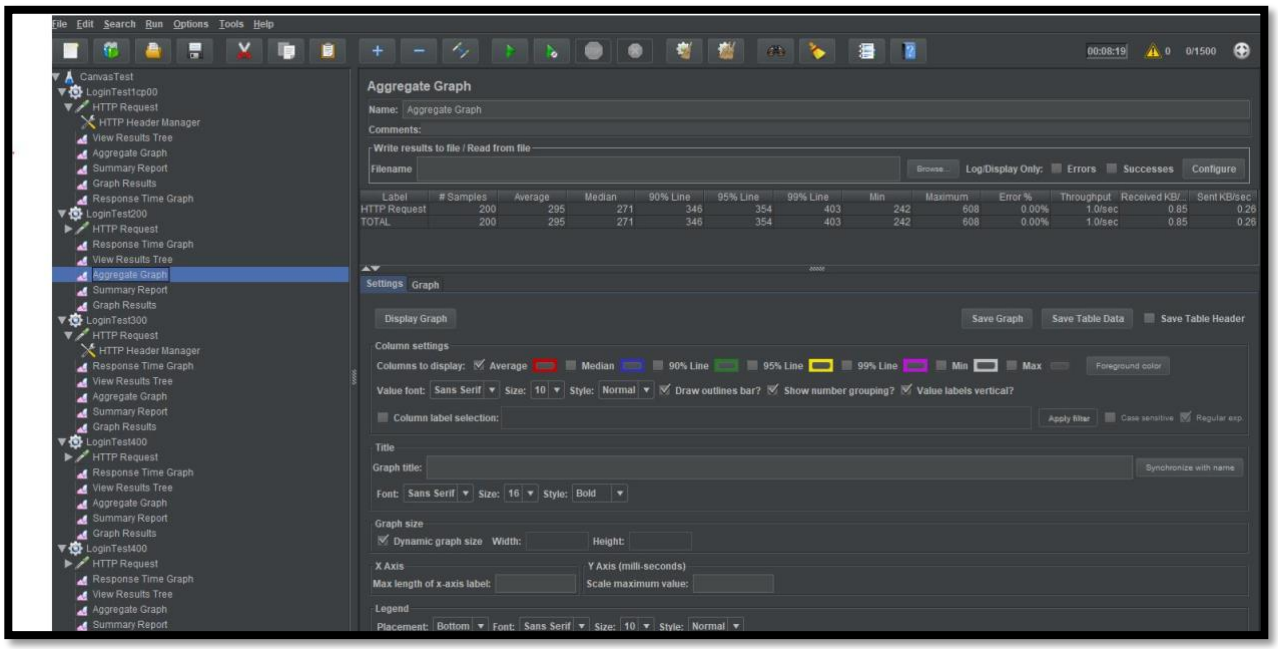


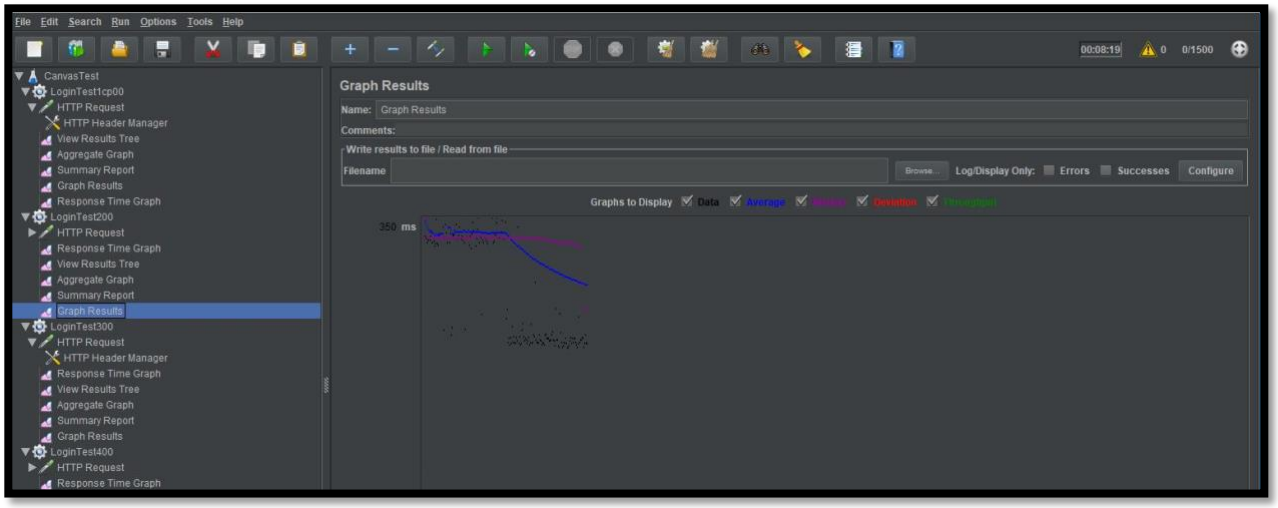


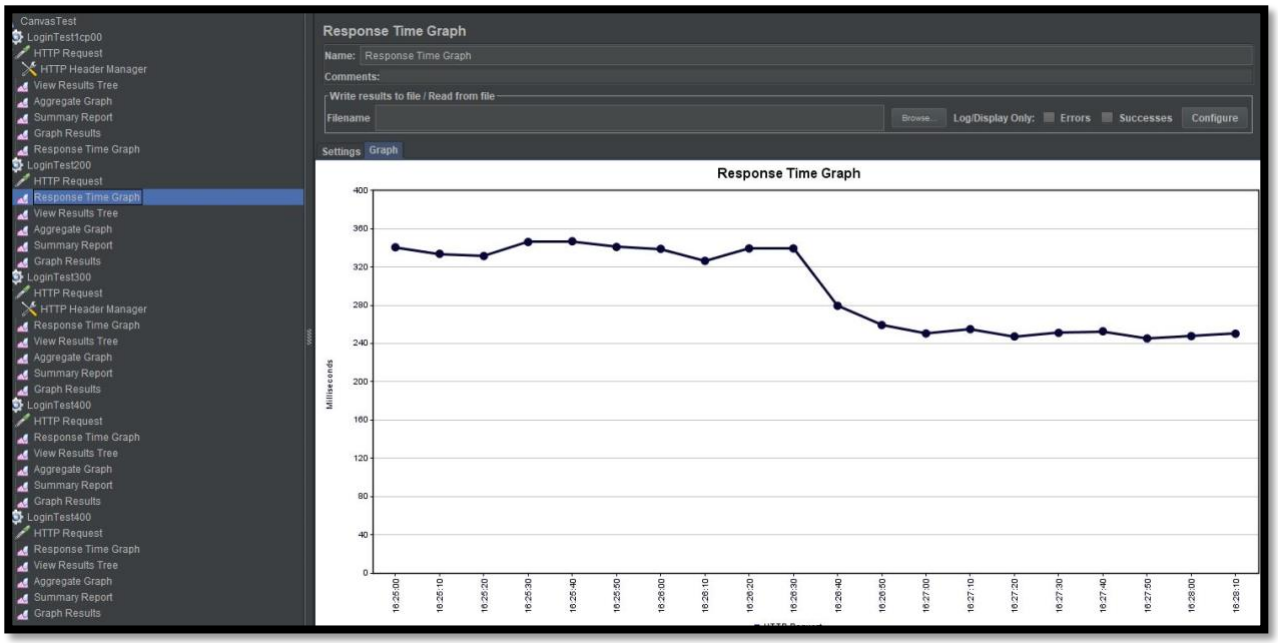

CanvasTest
LoginTest1cp00
HTTP Request
HTTP Header Manager
View Results Tree
Aggregate Graph
Summary Report
Graph Results
Response Time Graph
LoginTest200
HTTP Request
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
LoginTest300
HTTP Request
HTTP Header Manager
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
LoginTest400
HTTP Request
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
LoginTest400
HTTP Request
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
Response Time Graph
Name: Response Time Graph
Comments:
Write results to file / Read from file
Filename
Browse...
Log/Display Only: Errors Successes
Configure
Settings
Graph
Response Time Graph
Milliseconds
300
270
240
210
180
150
120
90
60
30
0
10:25:00
10:25:10
10:25:20
10:25:30
HTTP Request

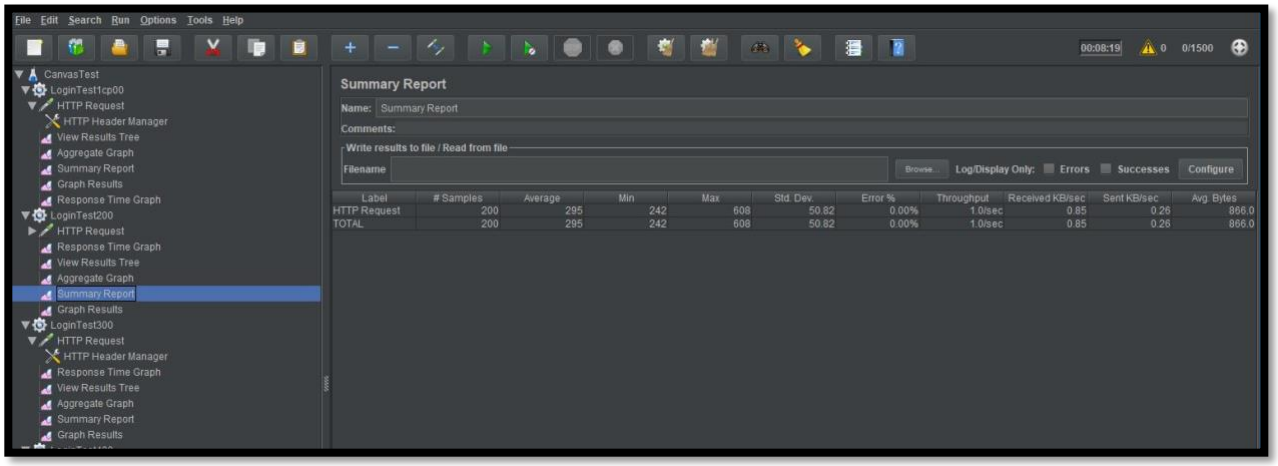

LoginTest1cp00
HTTP Request
HTTP Header Manager
View Results Tree
Aggregate Graph
Summary Report
Graph Results
Response Time Graph
LoginTest200
HTTP Request
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
LoginTest300
HTTP Request
HTTP Header Manager
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Name: Summary Report
Comments:
Write results to file / Read from file
Filename
Browse...
Log/Display Only: Errors Successes
Configure
Label # Samples Average Min Max Std. Dev. Error % Throughput Received KB/sec Sent KB/sec Avg. Bytes
HTTP Request 100 261 245 379 27.99 0.00% 1.0/sec 0.85 0.26 866.0
TOTAL 100 261 245 379 27.99 0.00% 1.0/sec 0.85 0.26 866.0

For 200 calls,

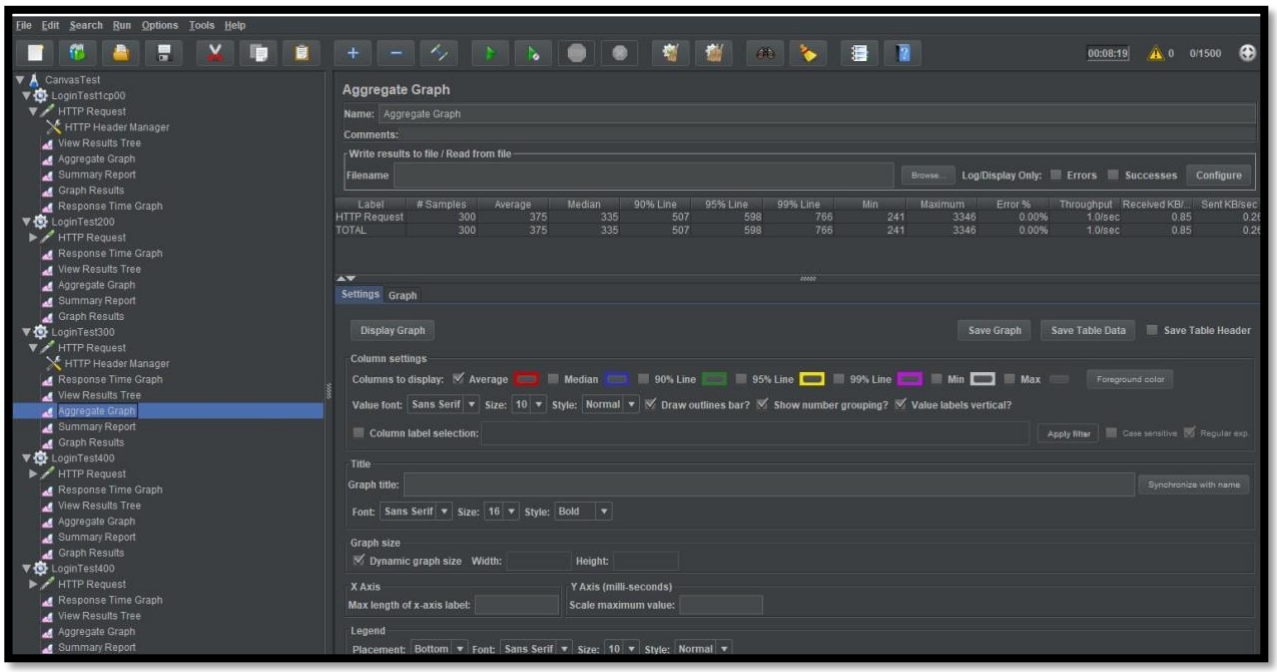


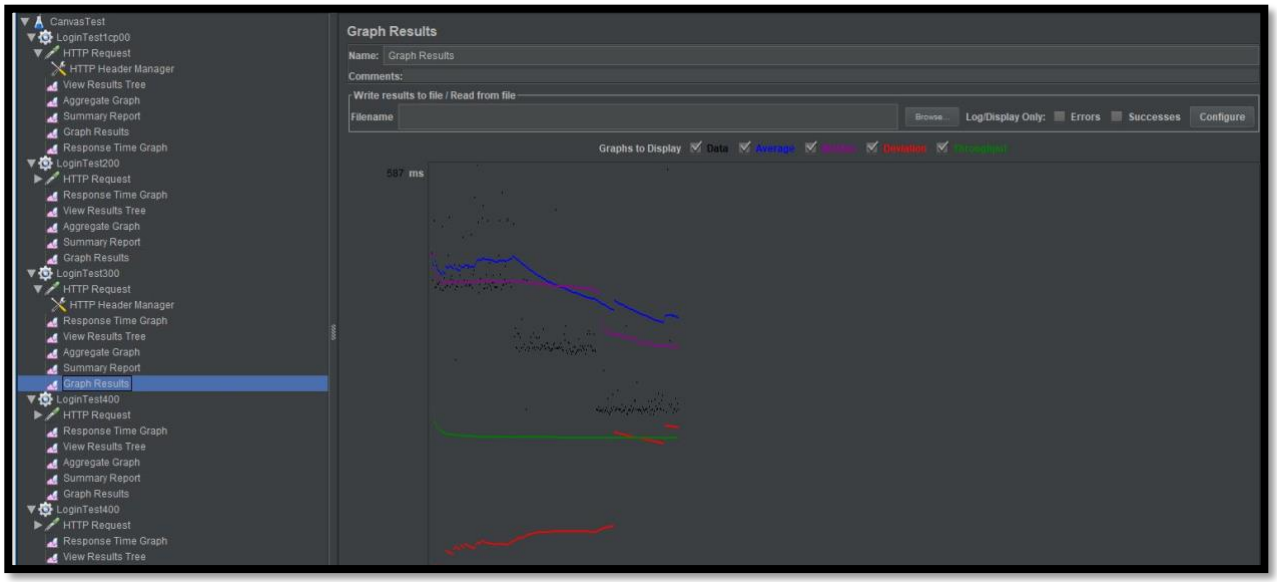


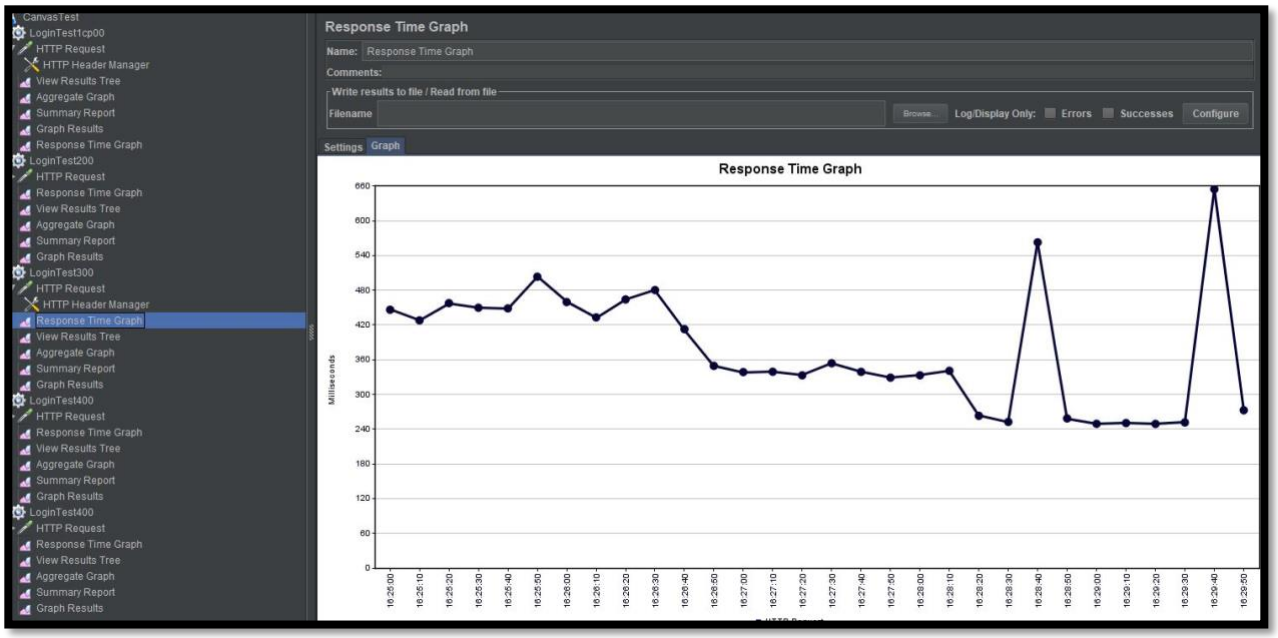

CanvasTest
LoginTest1cp00
HTTP Request
HTTP Header Manager
View Results Tree
Aggregate Graph
Summary Report
Graph Results
Response Time Graph
LoginTest200
HTTP Request
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
LoginTest300
HTTP Request
HTTP Header Manager
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
LoginTest400
HTTP Request
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
LoginTest400
HTTP Request
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
Response Time Graph
Name: Response Time Graph
Comments:
Write results to file / Read from file
Filename
Browse...
Log/Display Only:
Errors
Successes
Configure
Settings
Graph
Response Time Graph
Milliseconds
400
360
320
280
240
200
160
120
80
40
0
16:25:00
16:25:10
16:25:20
16:25:30
16:25:40
16:25:50
16:26:00
16:26:10
16:26:20
16:26:30
16:26:40
16:26:50
16:27:00
16:27:10
16:27:20
16:27:30
16:27:40
16:27:50
16:28:00
16:28:10

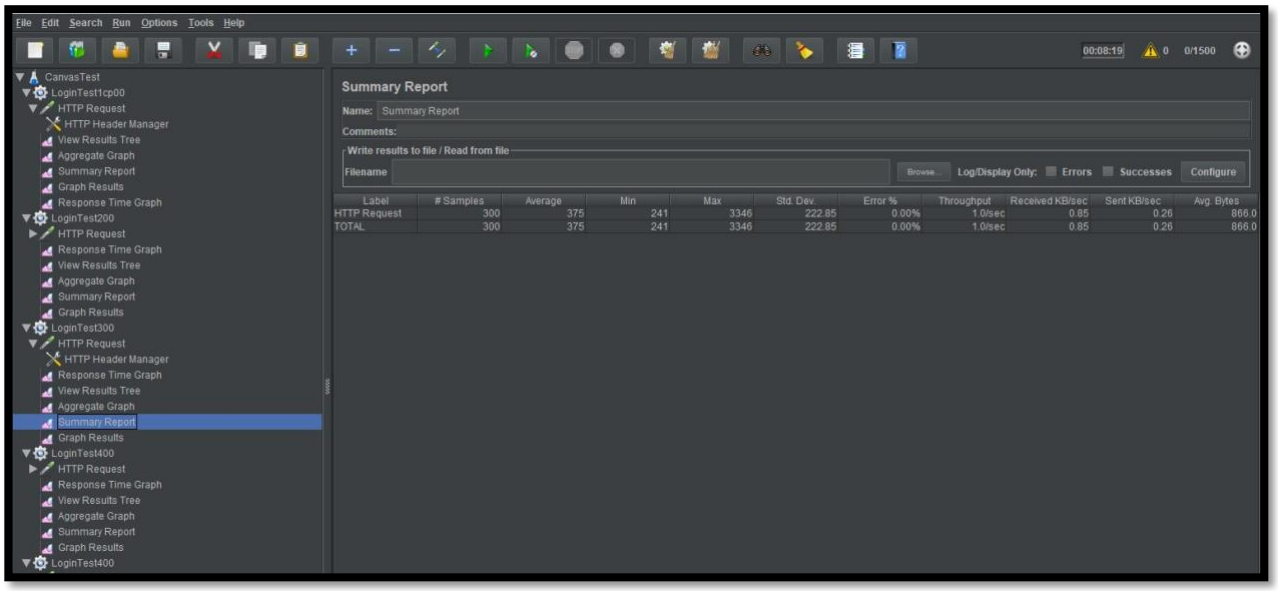

File
Edit
Search
Run
Options
Tools
Help
00:08:19
0
0/1500
CanvasTest
LoginTest1cp00
HTTP Request
HTTP Header Manager
View Results Tree
Aggregate Graph
Summary Report
Graph Results
Response Time Graph
LoginTest200
HTTP Request
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
LoginTest300
HTTP Request
HTTP Header Manager
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
Summary Report
Name: Summary Report
Comments:
Write results to file / Read from file
Filename
Browse...
Log/Display Only:
Errors
Successes
Configure
Label
Samples
Average
Min
Max
Std. Dev.
Error %
Throughput
Received KB/sec
Sent KB/sec
Avg. Bytes
HTTP Request
200
295
242
608
50.82
0.00%
1.0/sec
0.85
0.26
866.0
TOTAL
200
295
242
608
50.82
0.00%
1.0/sec
0.85
0.26
866.0

For 300 calls,

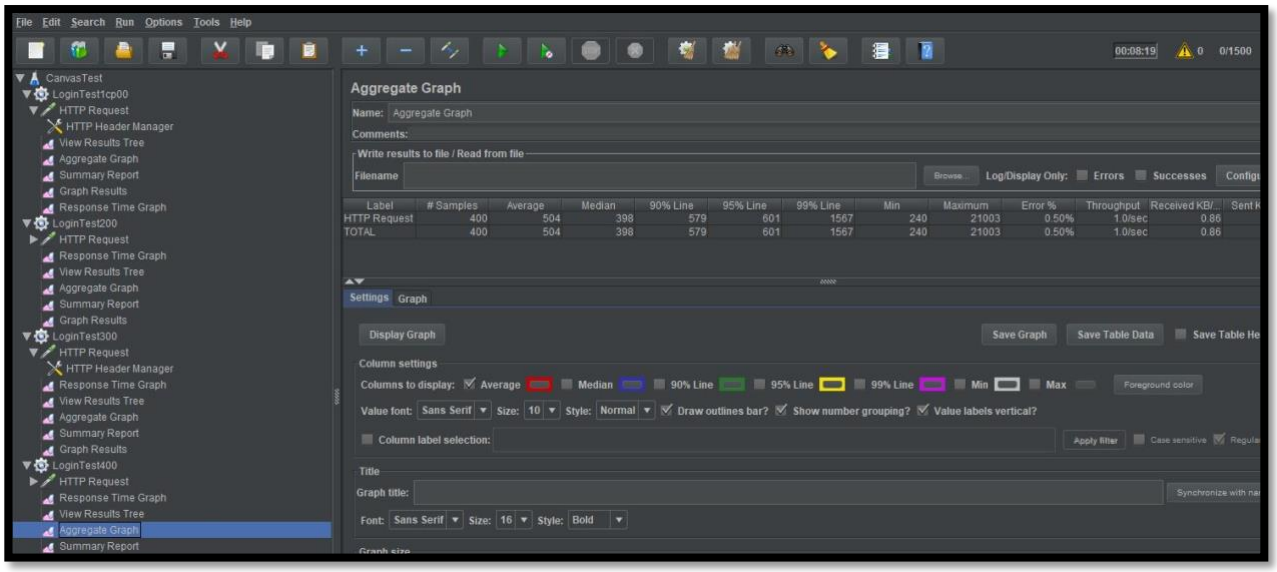


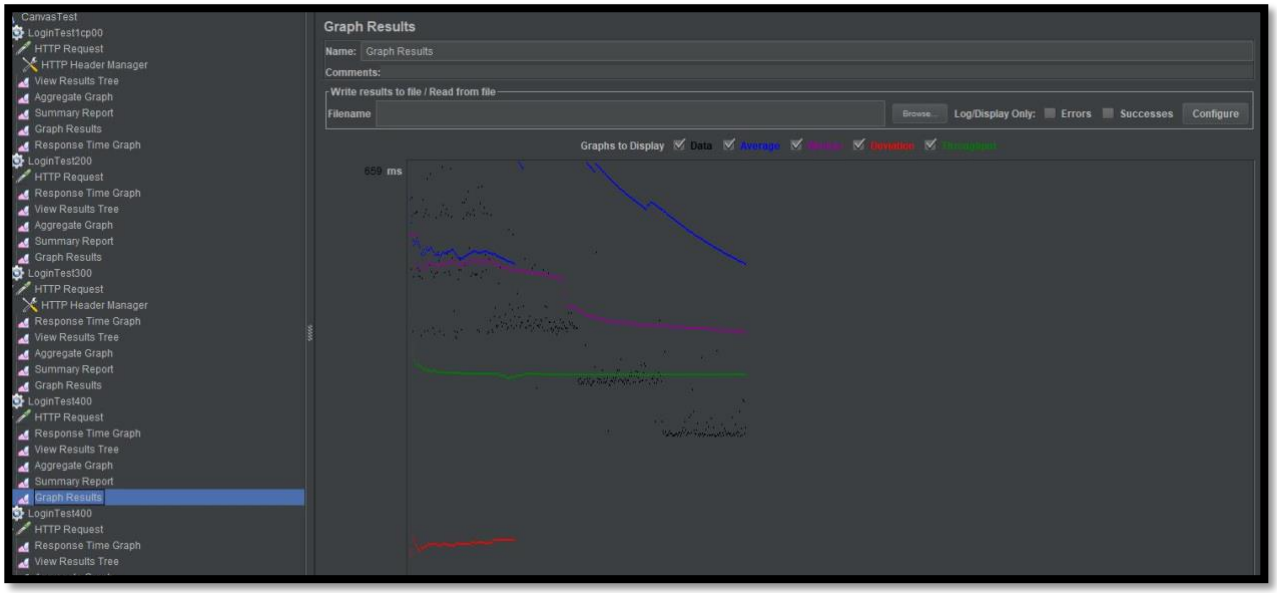


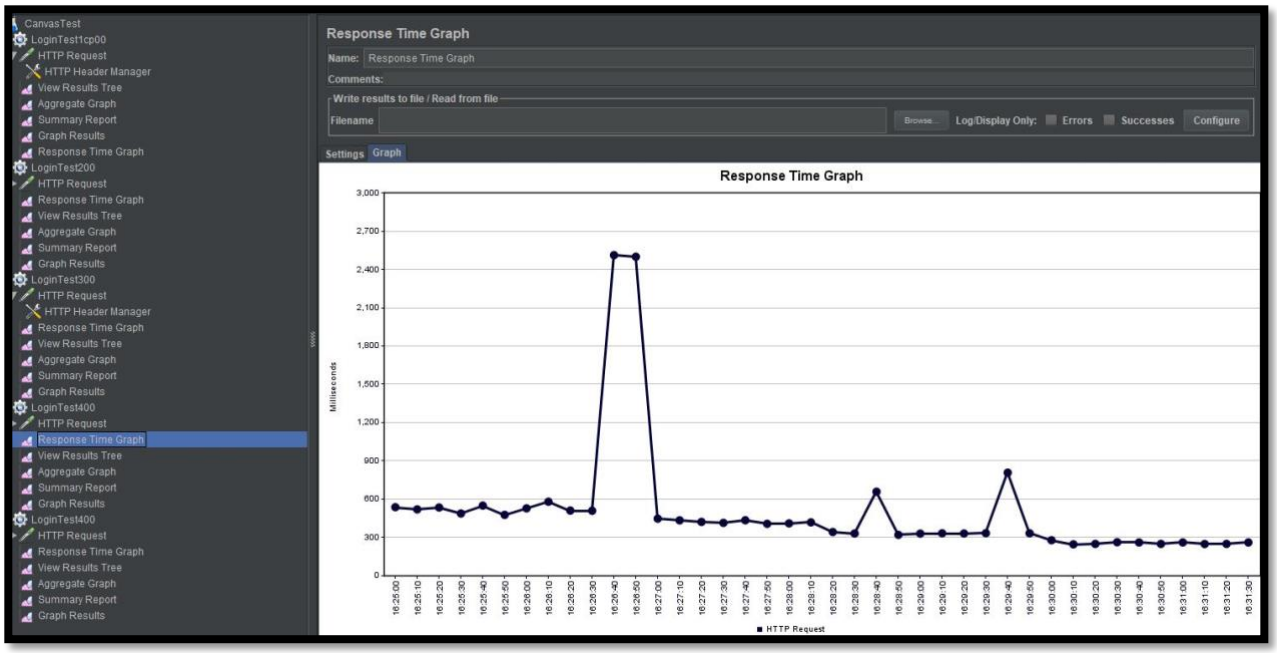

CanvasTest
LoginTest1cp00
HTTP Request
HTTP Header Manager
View Results Tree
Aggregate Graph
Summary Report
Graph Results
Response Time Graph
LoginTest200
HTTP Request
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
LoginTest300
HTTP Request
HTTP Header Manager
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
LoginTest400
HTTP Request
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
LoginTest400
HTTP Request
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
Response Time Graph
Name: Response Time Graph
Comments:
Write results to file / Read from file
Filename
Browse...
Log/Display Only: Errors Successes
Configure
Settings
Graph
Response Time Graph
Milliseconds
660
600
540
480
420
360
300
240
180
120
60
0

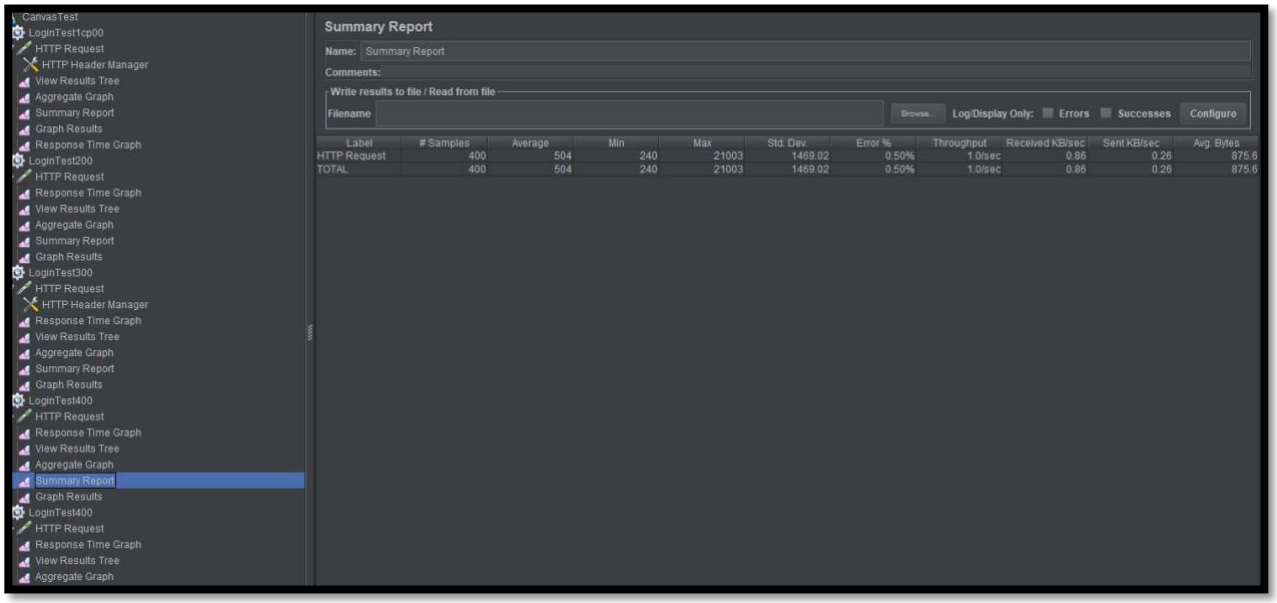

File Edit Search Run Options Tools Help
00:08:19
0
0/1500
CanvasTest
LoginTest1cp00
HTTP Request
HTTP Header Manager
View Results Tree
Aggregate Graph
Summary Report
Graph Results
Response Time Graph
LoginTest200
HTTP Request
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
LoginTest300
HTTP Request
HTTP Header Manager
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
LoginTest400
HTTP Request
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
LoginTest400
Summary Report
Name: Summary Report
Comments:
Write results to file / Read from file
Filename
Browse...
Log/Display Only: Errors Successes
Configure
Label # Samples Average Min Max Std. Dev. Error % Throughput Received KB/sec Sent KB/sec Avg. Bytes
HTTP Request 300 375 241 3346 222.85 0.00% 1.0/sec 0.85 0.26 866.0
TOTAL 300 375 241 3346 222.85 0.00% 1.0/sec 0.85 0.26 866.0

For 400 calls,

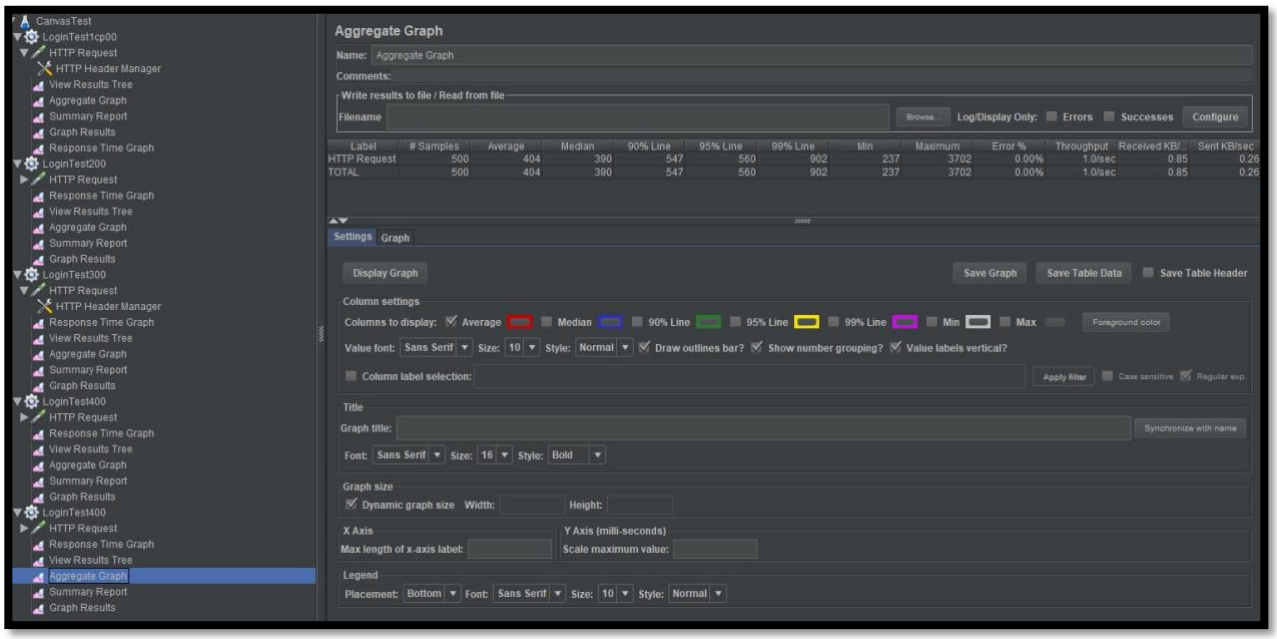


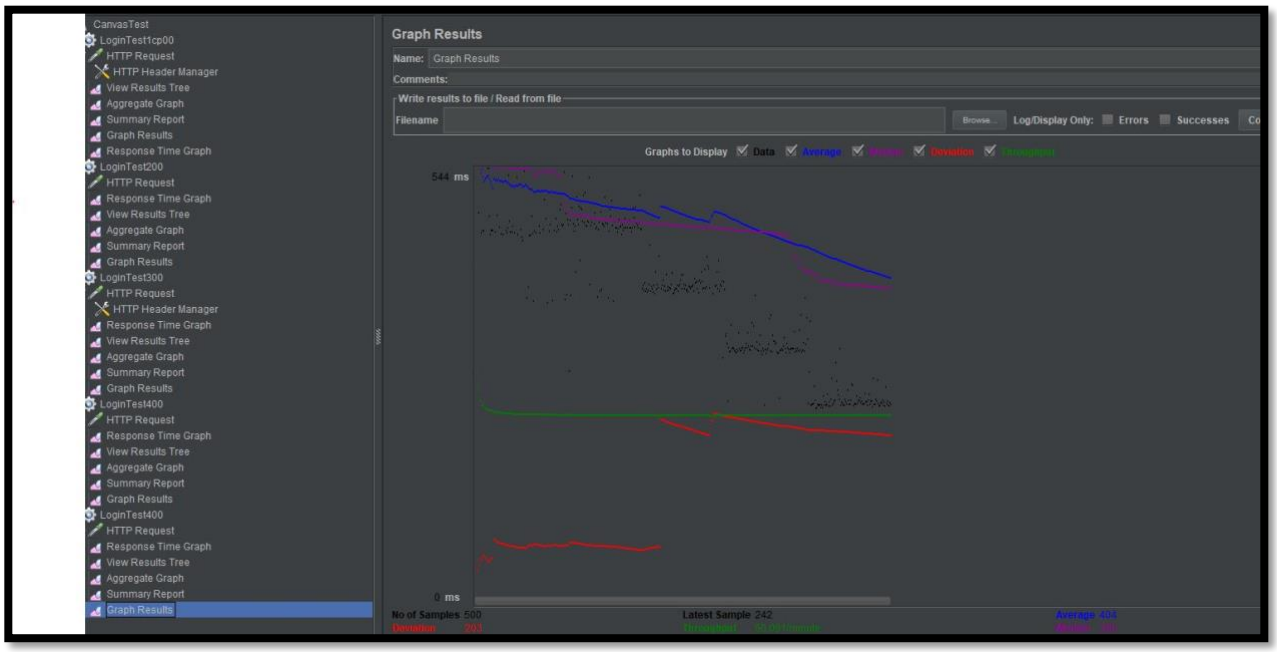


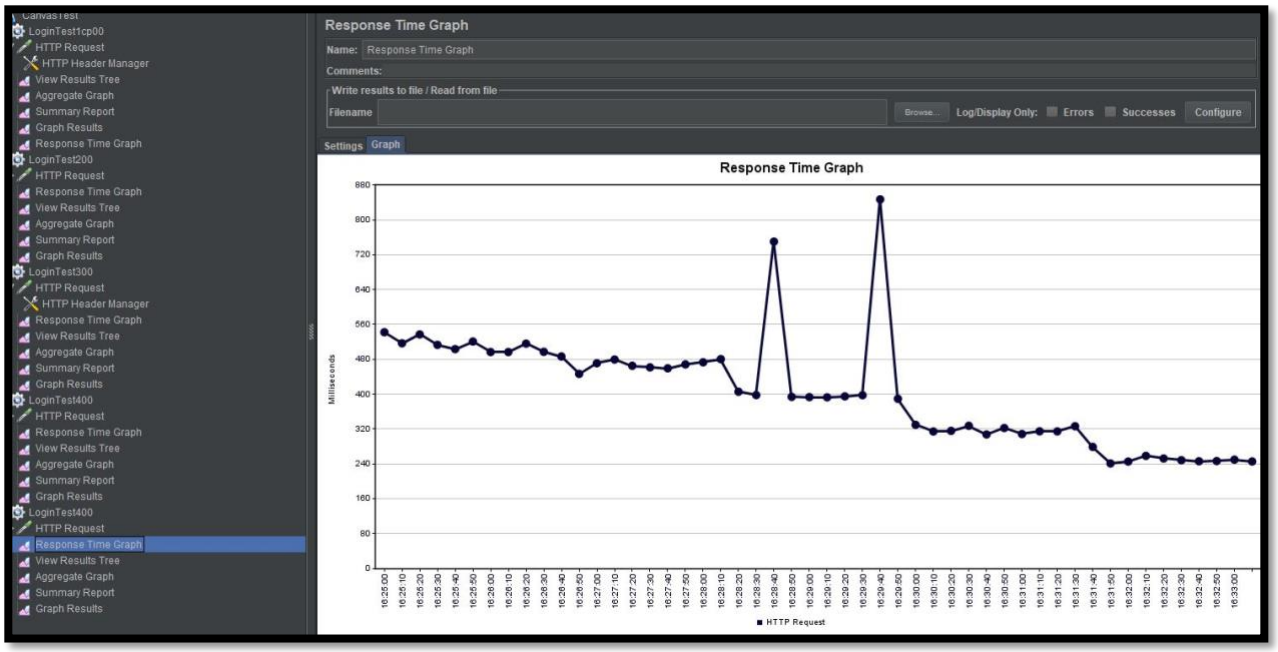

CanvasTest
LoginTest1cp00
HTTP Request
HTTP Header Manager
View Results Tree
Aggregate Graph
Summary Report
Graph Results
Response Time Graph
LoginTest200
HTTP Request
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
LoginTest300
HTTP Request
HTTP Header Manager
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
LoginTest400
HTTP Request
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
LoginTest400
HTTP Request
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
Response Time Graph
Name: Response Time Graph
Comments:
Write results to file / Read from file
Filename
Browse...
Log/Display Only: Errors Successes
Configure
Settings
Graph
Response Time Graph
Milliseconds
3,000
2,700
2,400
2,100
1,800
1,500
1,200
900
600
300
0
HTTP Request

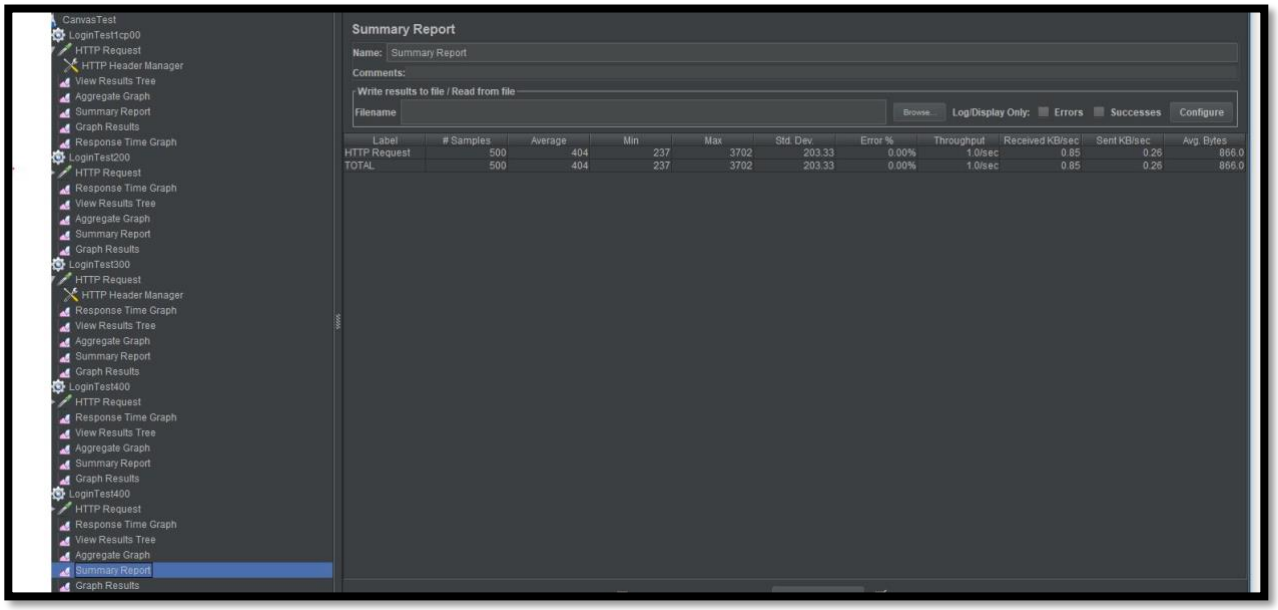

CanvasTest
LoginTest1cp00
HTTP Request
HTTP Header Manager
View Results Tree
Aggregate Graph
Summary Report
Graph Results
Response Time Graph
LoginTest200
HTTP Request
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
LoginTest300
HTTP Request
HTTP Header Manager
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
LoginTest400
HTTP Request
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
LoginTest400
HTTP Request
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Name: Summary Report
Comments:
Write results to file / Read from file
Filename
Browse...
Log/Display Only: Errors Successes
Configure
Label # Samples Average Min Max Std. Dev. Error % Throughput Received KB/sec Sent KB/sec Avg. Bytes
HTTP Request 400 504 240 21003 1469.02 0.50% 1.0/sec 0.86 0.26 875.6
TOTAL 400 504 240 21003 1469.02 0.50% 1.0/sec 0.86 0.26 875.6

For 500 calls,



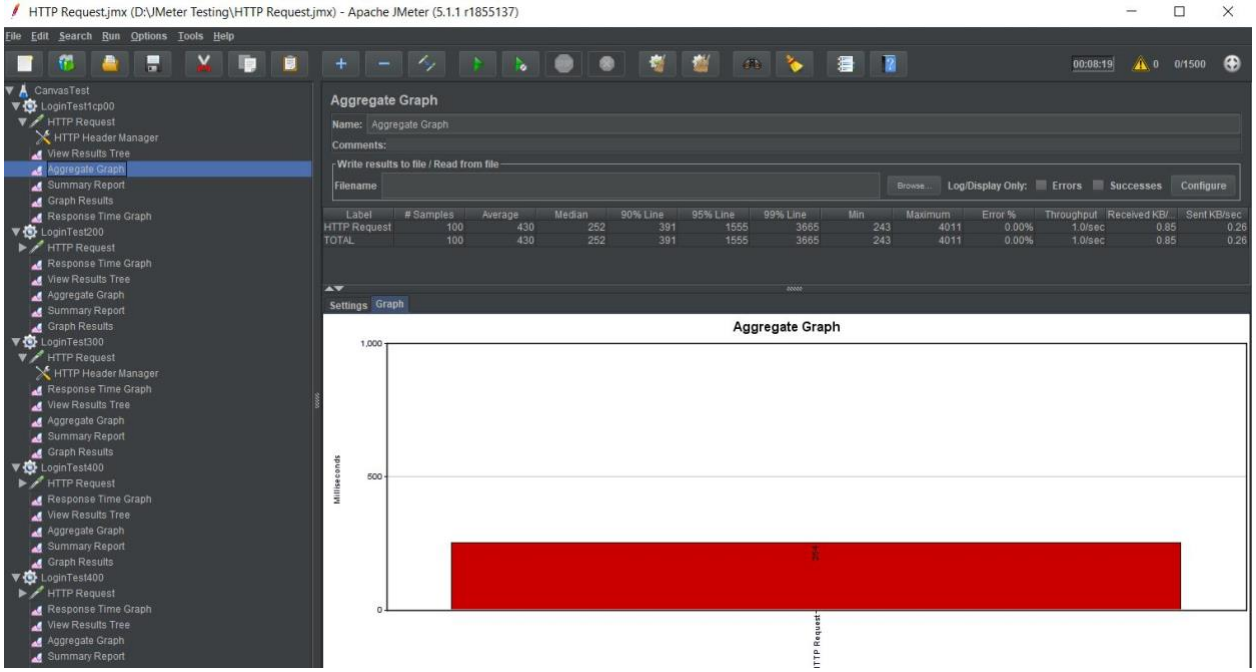


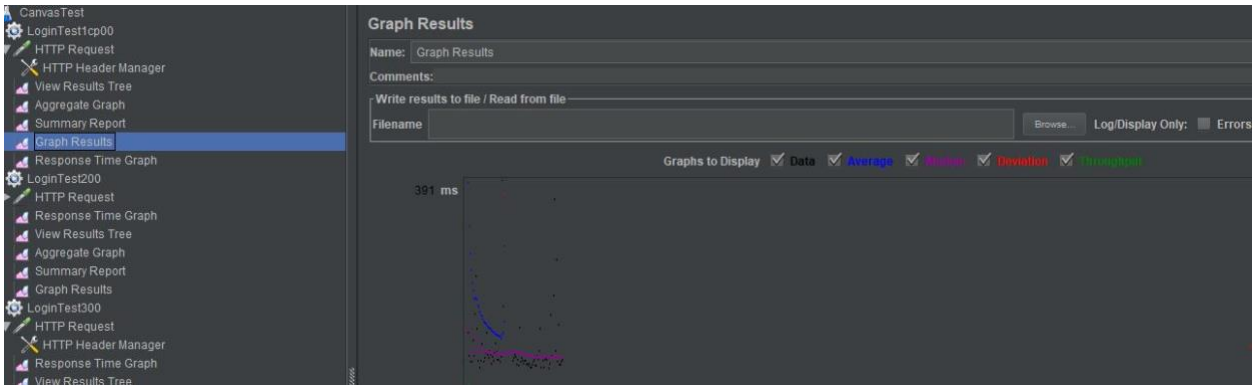


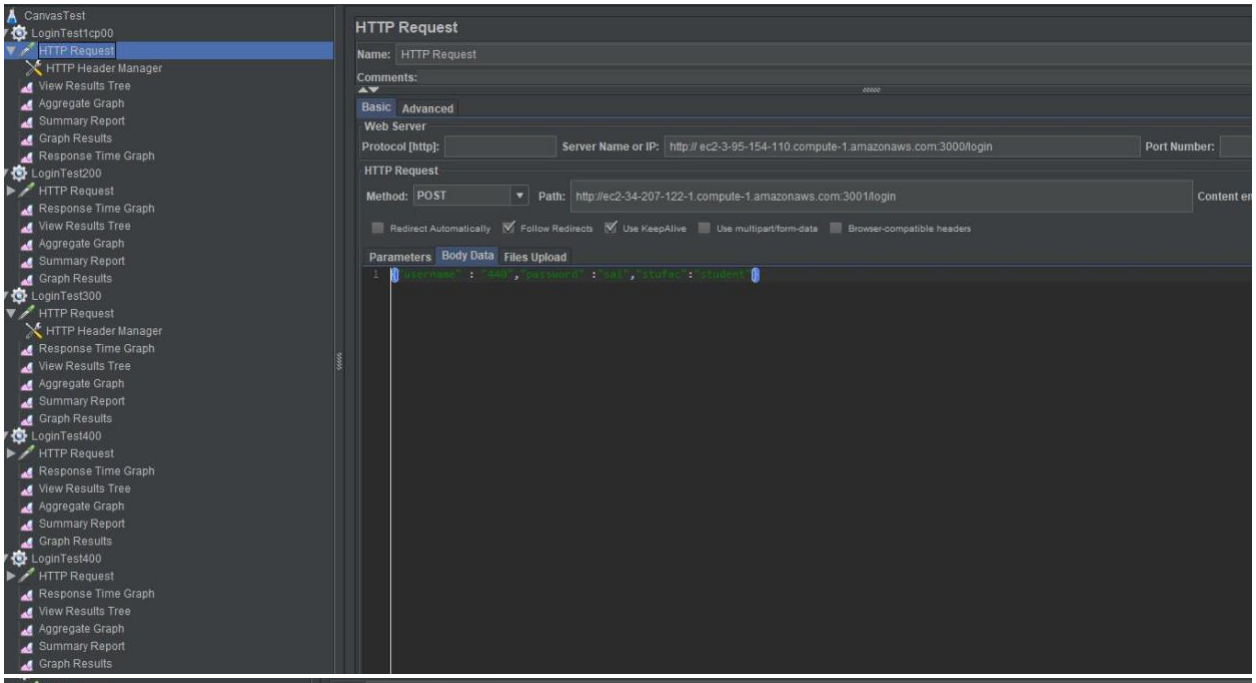


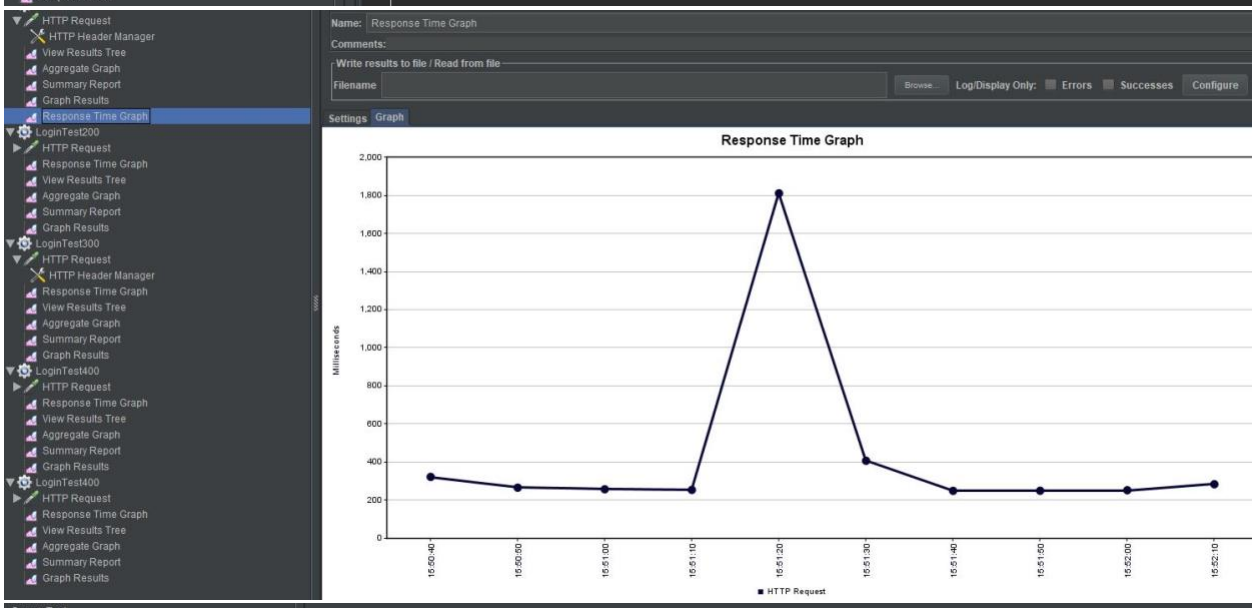
- <u>JMeter With Pooling</u>

  For 100 calls,


HTTP Request.jmx (D:\JMeter Testing\HTTP Request.jmx) - Apache JMeter (5.1.1 r1855137)
File Edit Search Run Options Tools Help
00:08:19
0
0/1500
CanvasTest
LoginTest1cp00
HTTP Request
HTTP Header Manager
View Results Tree
Aggregate Graph
Summary Report
Graph Results
Response Time Graph
LoginTest200
HTTP Request
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
LoginTest300
HTTP Request
HTTP Header Manager
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
LoginTest400
HTTP Request
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
LoginTest400
HTTP Request
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Aggregate Graph
Name: Aggregate Graph
Comments:
Write results to file / Read from file
Filename
Browse...
Log/Display Only:
Errors
Successes
Configure
Label # Samples Average Median 90% Line 95% Line 99% Line Min Maximum Error % Throughput Received KB/... Sent KB/sec
HTTP Request 100 430 252 391 1555 3665 243 4011 0.00% 1.0/sec 0.85 0.26
TOTAL 100 430 252 391 1555 3665 243 4011 0.00% 1.0/sec 0.85 0.26
Settings
Graph
Aggregate Graph
1,000
500
0
Milliseconds
HTTP Request

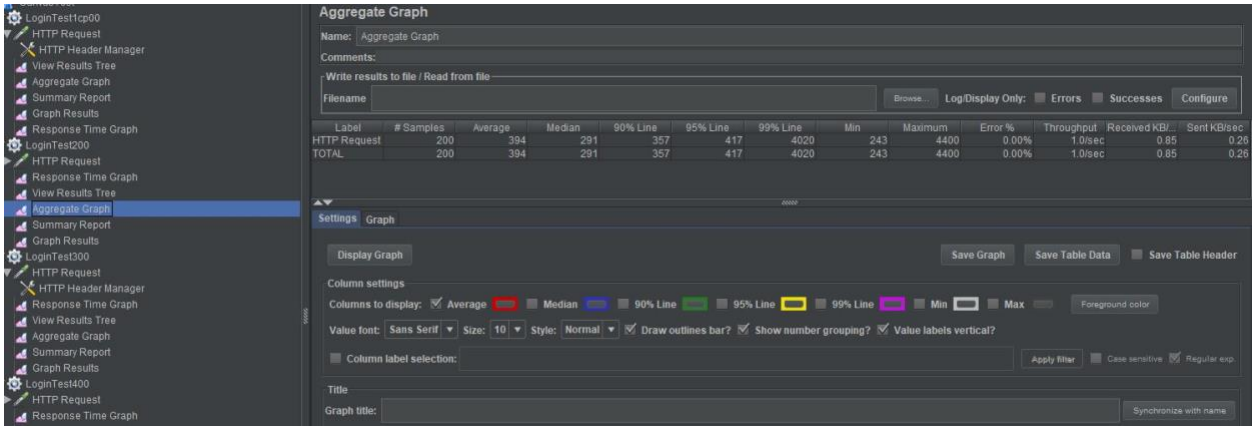

CanvasTest
LoginTest1cp00
HTTP Request
HTTP Header Manager
View Results Tree
Aggregate Graph
Summary Report
Graph Results
Response Time Graph
LoginTest200
HTTP Request
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
LoginTest300
HTTP Request
HTTP Header Manager
Response Time Graph
View Results Tree
Graph Results
Name: Graph Results
Comments:
Write results to file / Read from file
Filename
Browse...
Log/Display Only:
Errors
Graphs to Display
Data
Average
Deviation
Throughput
391 ms

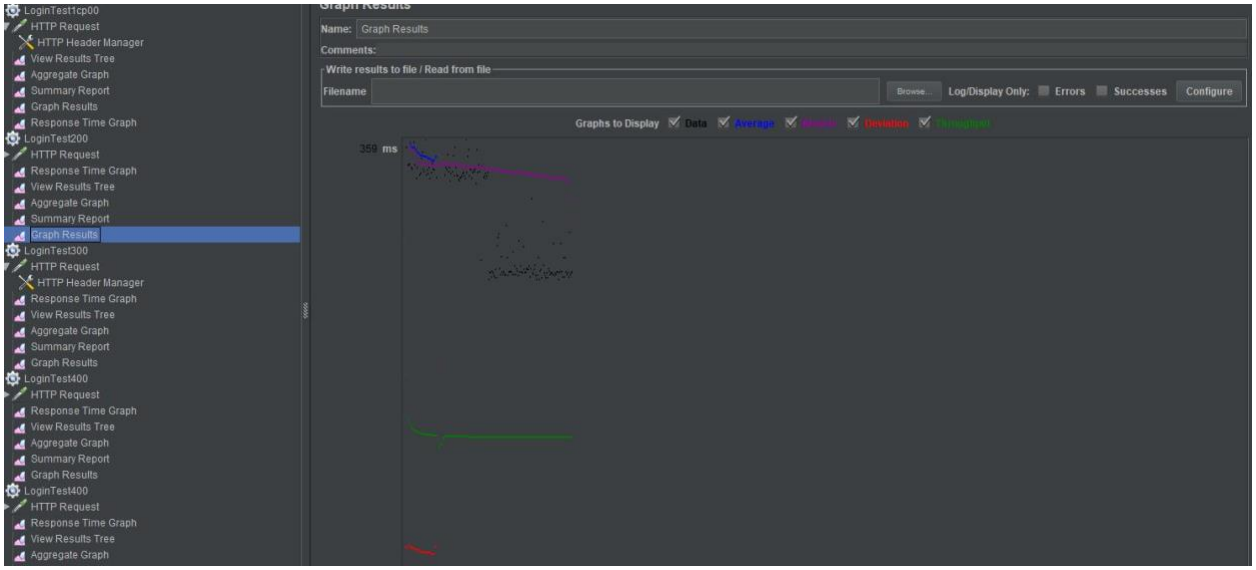

CanvasTest
LoginTest1cp00
HTTP Request
HTTP Header Manager
View Results Tree
Aggregate Graph
Summary Report
Graph Results
Response Time Graph
LoginTest200
HTTP Request
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
LoginTest300
HTTP Request
HTTP Header Manager
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
LoginTest400
HTTP Request
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
LoginTest400
HTTP Request
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
HTTP Request
Name: HTTP Request
Comments:
Basic
Advanced
Web Server
Protocol [http]:
Server Name or IP: http:// ec2-3-95-154-110.compute-1.amazonaws.com:3000/login
Port Number:
HTTP Request
Method: POST
Path: http://ec2-34-207-122-1.compute-1.amazonaws.com:3001/login
Content en
Redirect Automatically
Follow Redirects
Use KeepAlive
Use multipart/form-data
Browser-compatible headers
Parameters
Body Data
Files Upload
1 {"username" : "440","password" : sai","stufac": student"}

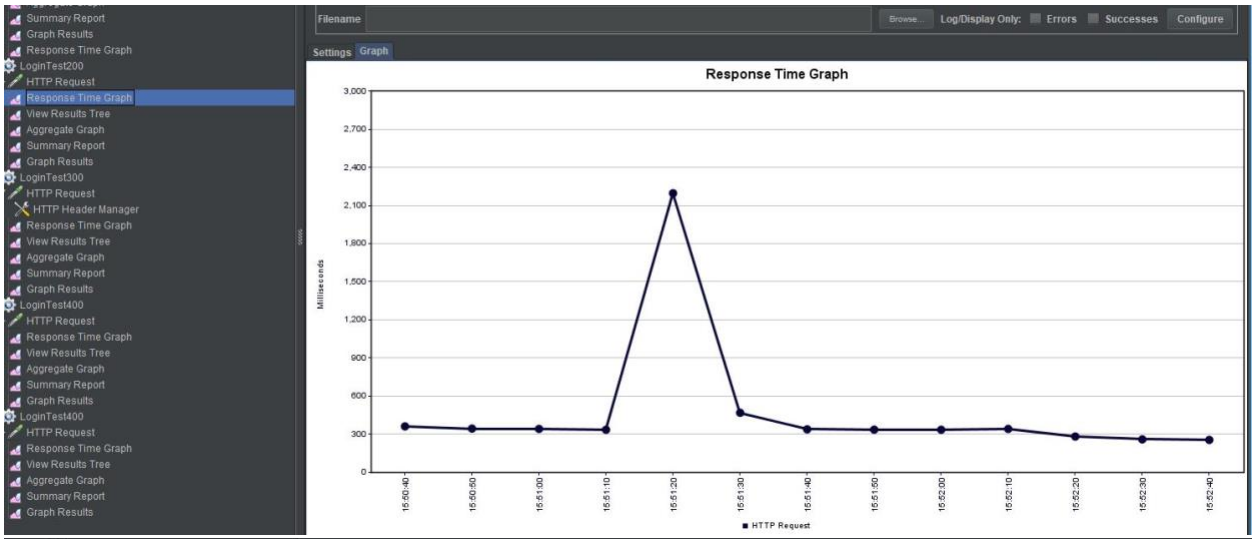

HTTP Request
HTTP Header Manager
View Results Tree
Aggregate Graph
Summary Report
Graph Results
Response Time Graph
LoginTest200
HTTP Request
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
LoginTest300
HTTP Request
HTTP Header Manager
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
LoginTest400
HTTP Request
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
LoginTest400
HTTP Request
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
Name: Response Time Graph
Comments:
Write results to file / Read from file
Filename
Browse...
Log/Display Only:
Errors
Successes
Configure
Settings
Graph
Response Time Graph
Milliseconds
2,000
1,800
1,600
1,400
1,200
1,000
800
600
400
200
0
15:60:40
15:60:50
15:61:00
15:61:10
15:61:20
15:61:30
15:61:40
15:61:50
15:62:00
15:62:10
HTTP Request

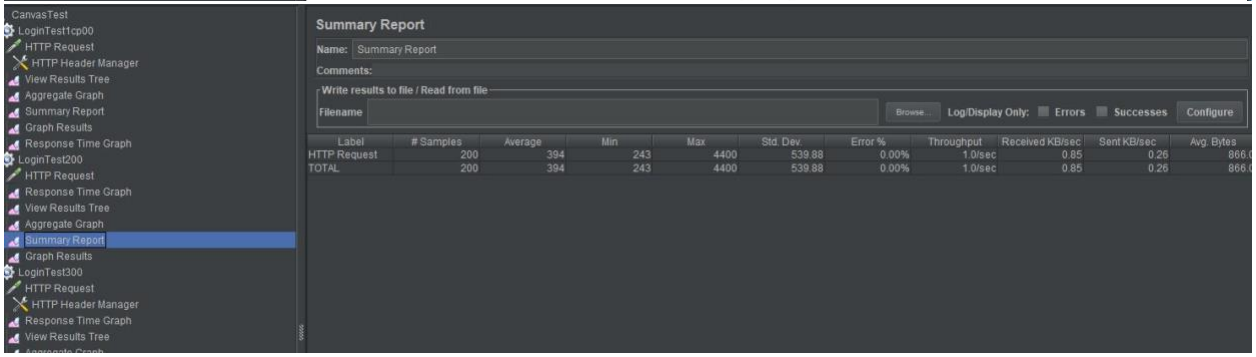

CanvasTest
LoginTest1cp00
HTTP Request
HTTP Header Manager
View Results Tree
Aggregate Graph
Summary Report
Graph Results
Response Time Graph
LoginTest200
HTTP Request
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
LoginTest300
HTTP Request
HTTP Request
Summary Report
Name: Summary Report
Comments:
Write results to file / Read from file
Filename
Browse...
Log/Display Only:
Errors
Successes
Configure
Label
Samples
Average
Min
Max
Std. Dev.
Error %
Throughput
Received KB/sec
Sent KB/sec
Avg. Bytes
HTTP Request
100
430
243
4011
638.10
0.00%
1.0/sec
0.85
0.26
866.0
TOTAL
100
430
243
4011
638.10
0.00%
1.0/sec
0.85
0.26
866.0

For 200 calls,

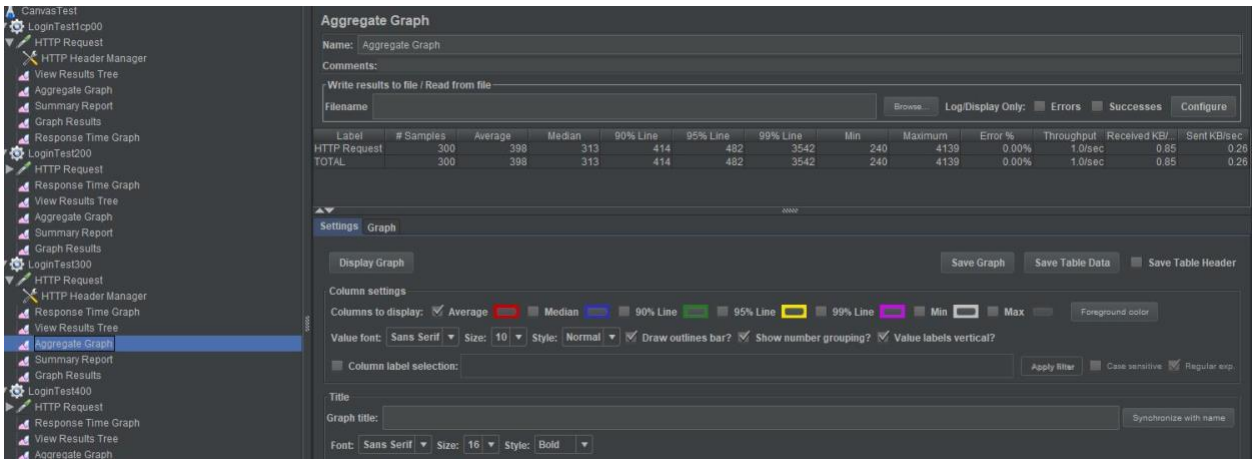


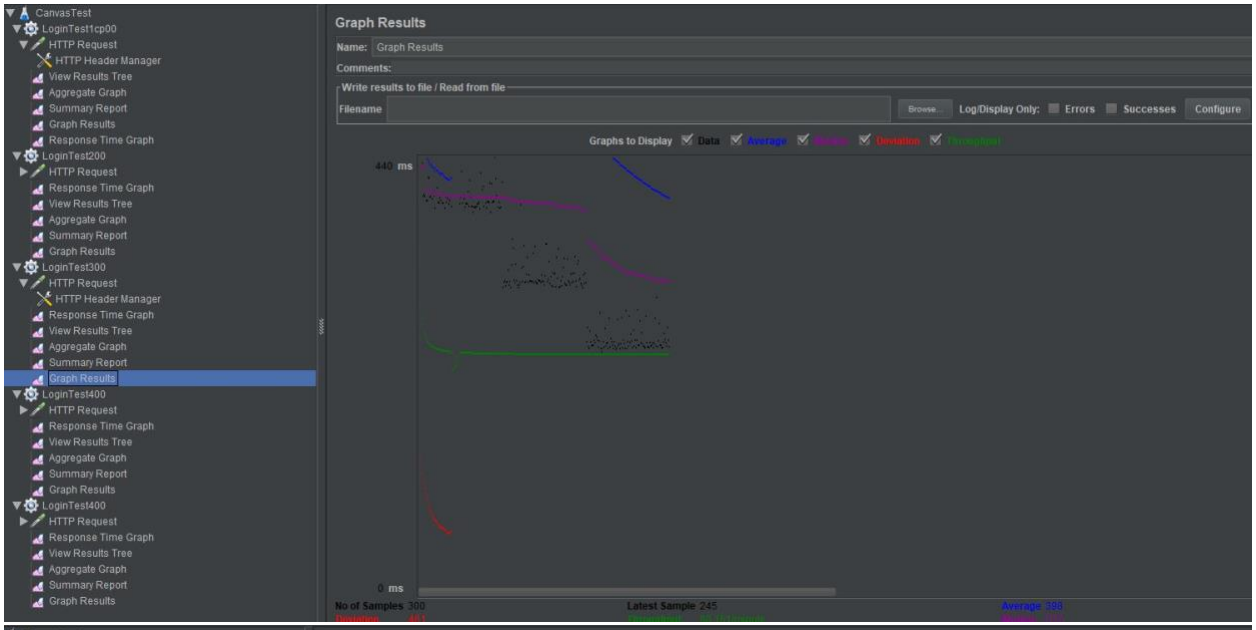


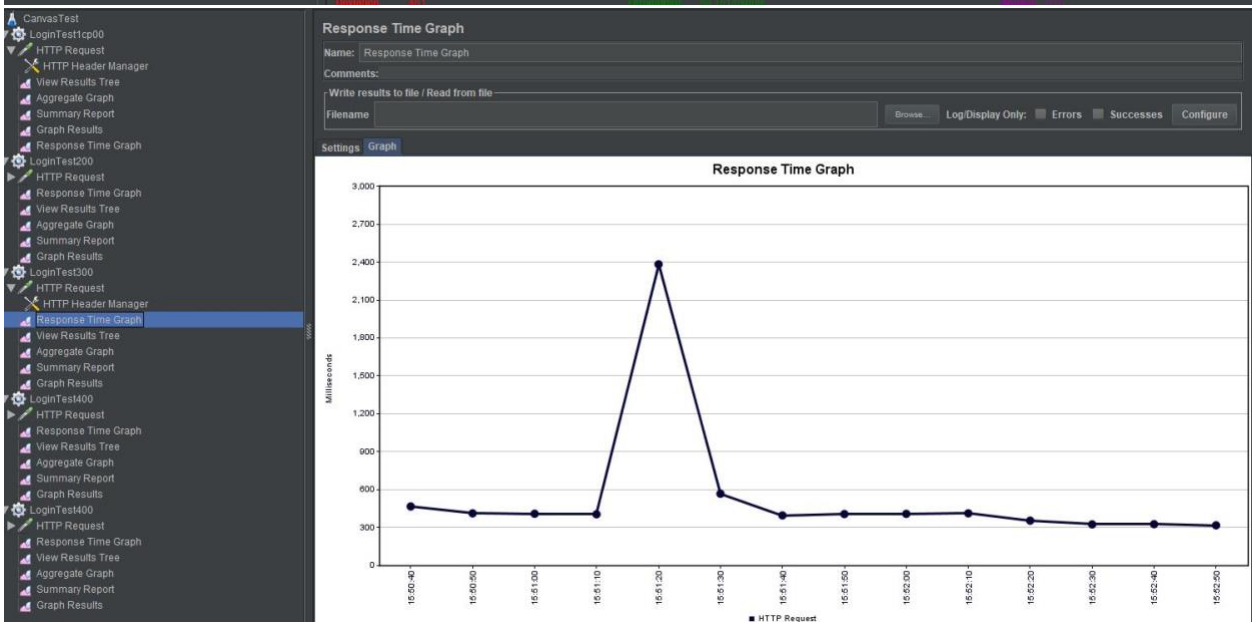


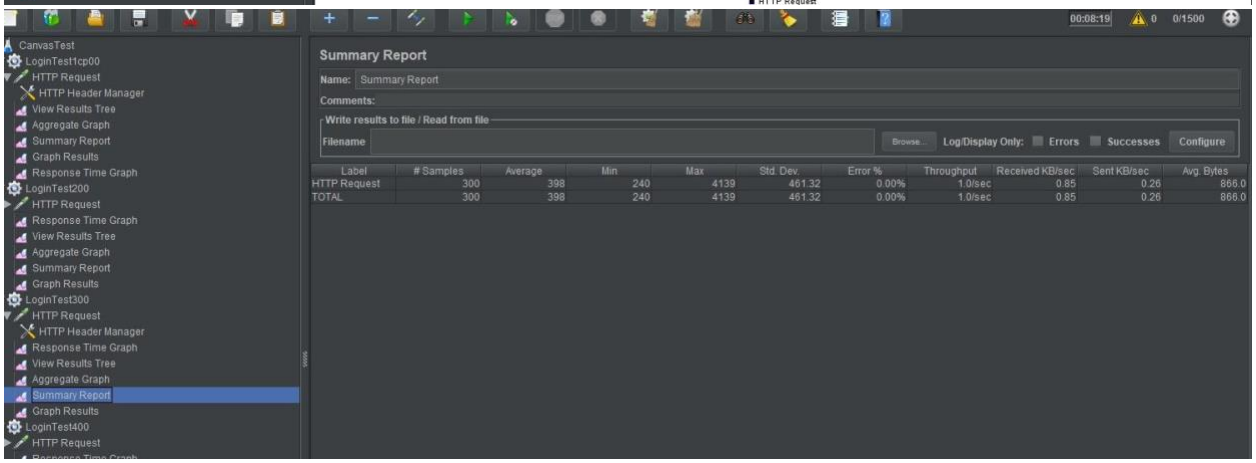


For 300 calls,

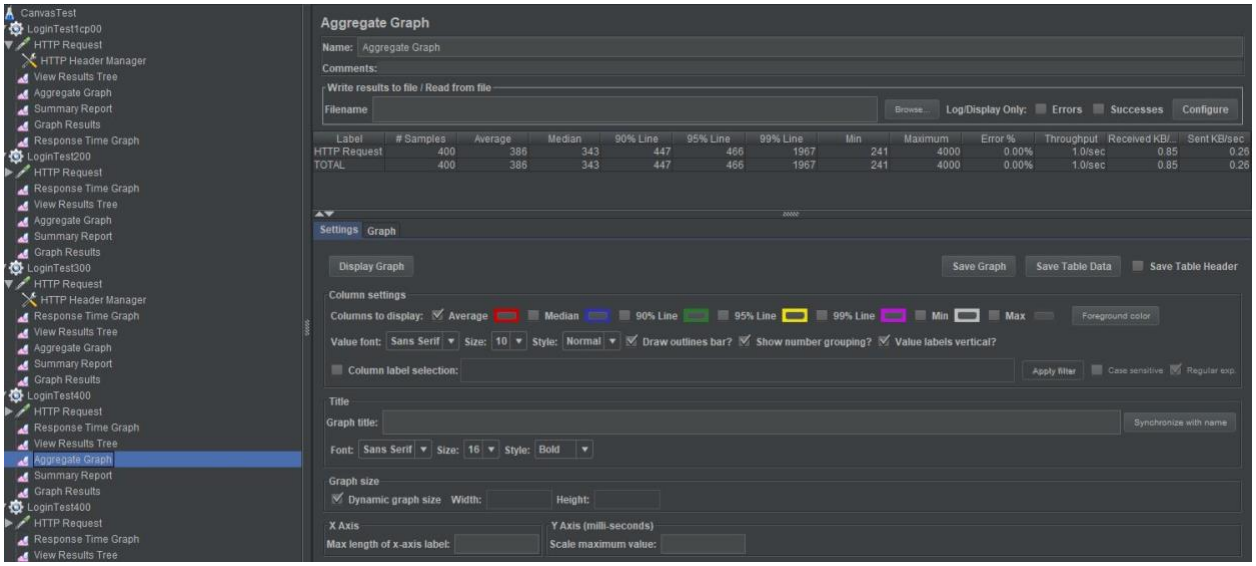


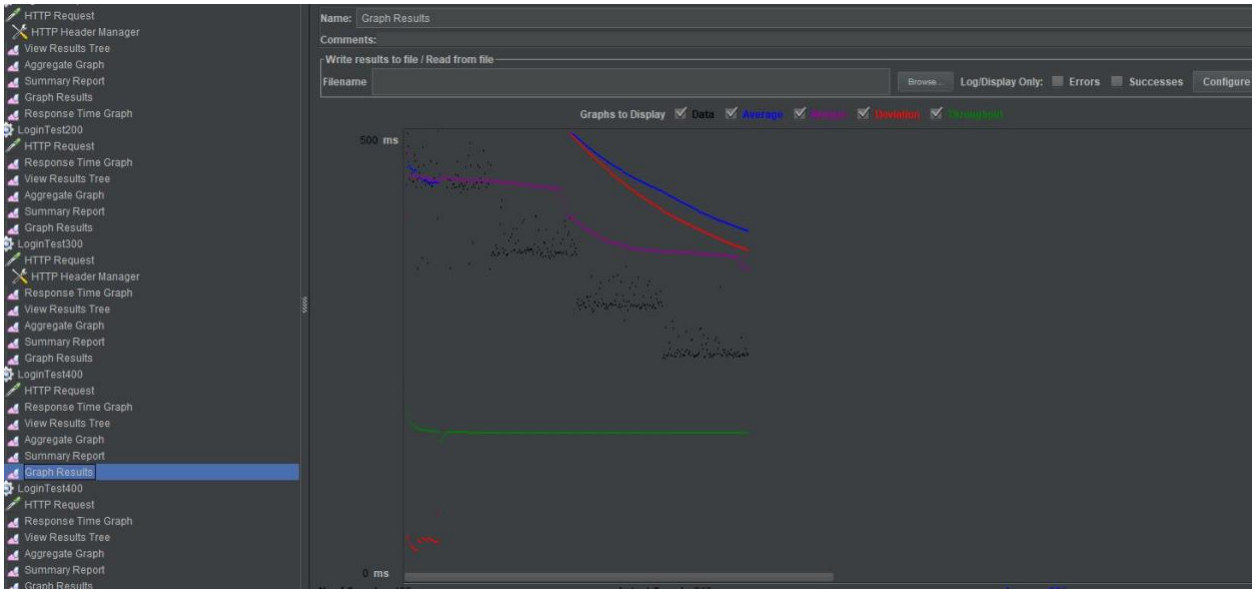

CanvasTest
LoginTest1cp00
HTTP Request
HTTP Header Manager
View Results Tree
Aggregate Graph
Summary Report
Graph Results
Response Time Graph
LoginTest200
HTTP Request
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
LoginTest300
HTTP Request
HTTP Header Manager
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
LoginTest400
HTTP Request
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
LoginTest400
HTTP Request
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
Graph Results
Name: Graph Results
Comments:
Write results to file / Read from file
Filename
Browse...
Log/Display Only: Errors Successes
Configure
Graphs to Display Data Average Median Deviation Throughput
440 ms
0 ms
No of Samples 300
Latest Sample 245
Average 398

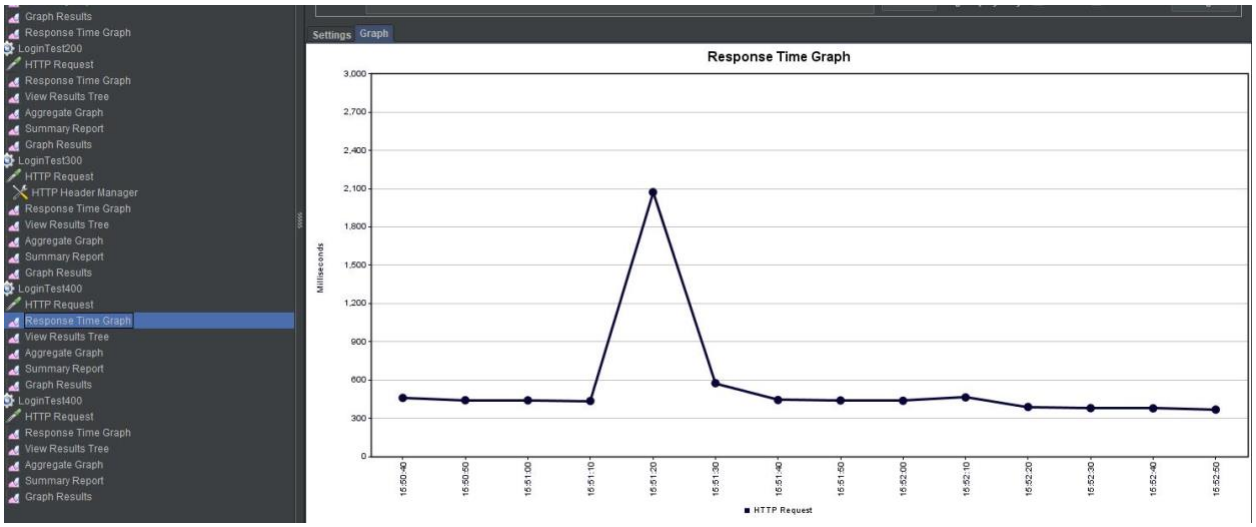

CanvasTest
LoginTest1cp00
HTTP Request
HTTP Header Manager
View Results Tree
Aggregate Graph
Summary Report
Graph Results
Response Time Graph
LoginTest200
HTTP Request
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
LoginTest300
HTTP Request
HTTP Header Manager
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
LoginTest400
HTTP Request
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
LoginTest400
HTTP Request
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
Response Time Graph
Name: Response Time Graph
Comments:
Write results to file / Read from file
Filename
Browse...
Log/Display Only: Errors Successes
Configure
Settings
Graph
Response Time Graph
Milliseconds
3,000
2,700
2,400
2,100
1,800
1,500
1,200
900
600
300
0
15:50:40
15:50:50
15:51:00
15:51:10
15:51:20
15:51:30
15:51:40
15:51:50
15:52:00
15:52:10
15:52:20
15:52:30
15:52:40
15:52:50
HTTP Request

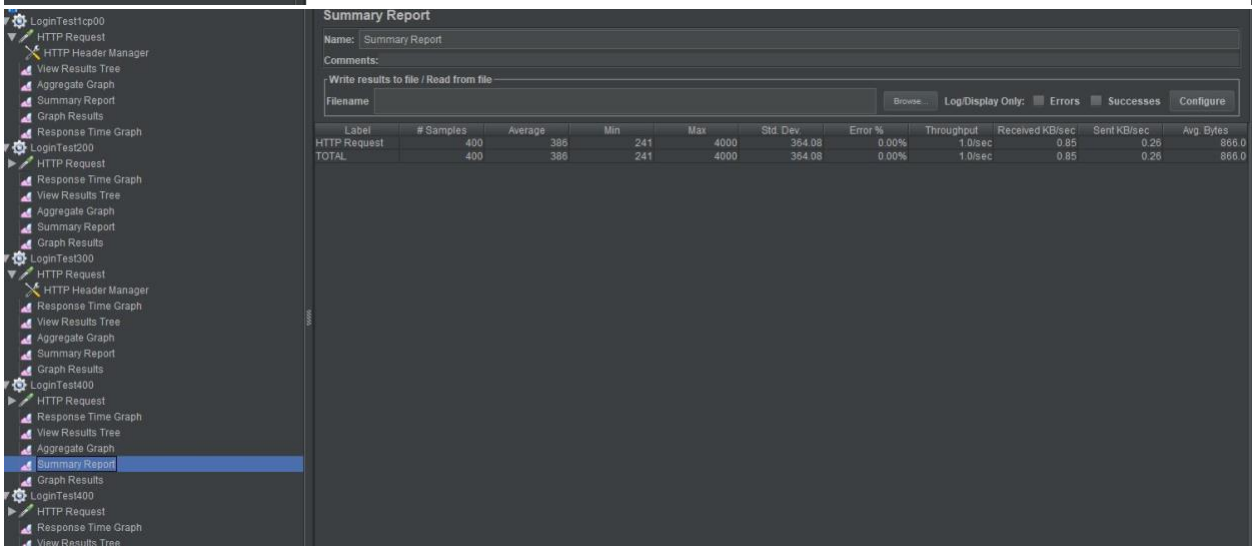

00:08:19
0
0/1500
CanvasTest
LoginTest1cp00
HTTP Request
HTTP Header Manager
View Results Tree
Aggregate Graph
Summary Report
Graph Results
Response Time Graph
LoginTest200
HTTP Request
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
LoginTest300
HTTP Request
HTTP Header Manager
Response Time Graph
View Results Tree
Aggregate Graph
Summary Report
Graph Results
LoginTest400
HTTP Request
Summary Report
Name: Summary Report
Comments:
Write results to file / Read from file
Filename
Browse...
Log/Display Only: Errors Successes
Configure
Label # Samples Average Min Max Std. Dev. Error % Throughput Received KB/sec Sent KB/sec Avg. Bytes
HTTP Request 300 398 240 4139 461.32 0.00% 1.0/sec 0.85 0.26 866.0
TOTAL 300 398 240 4139 461.32 0.00% 1.0/sec 0.85 0.26 866.0

For 400 calls,

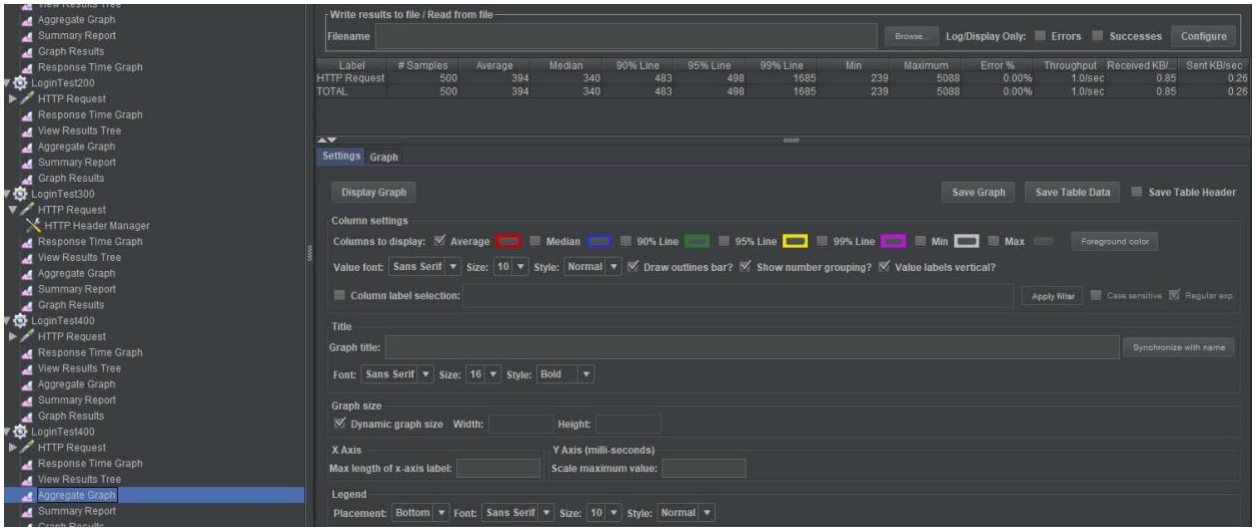


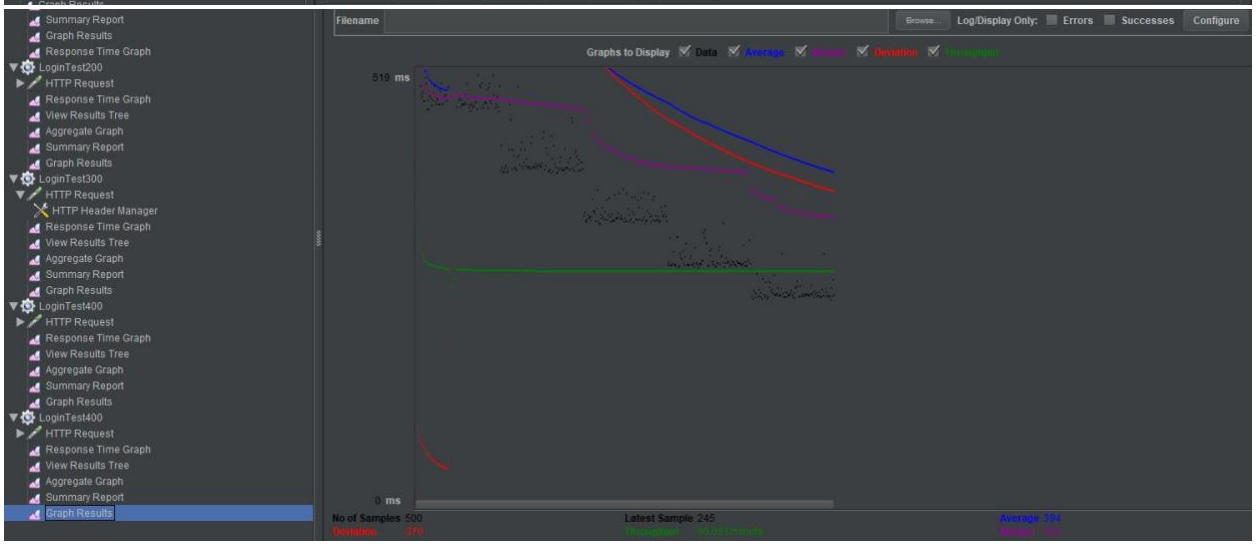




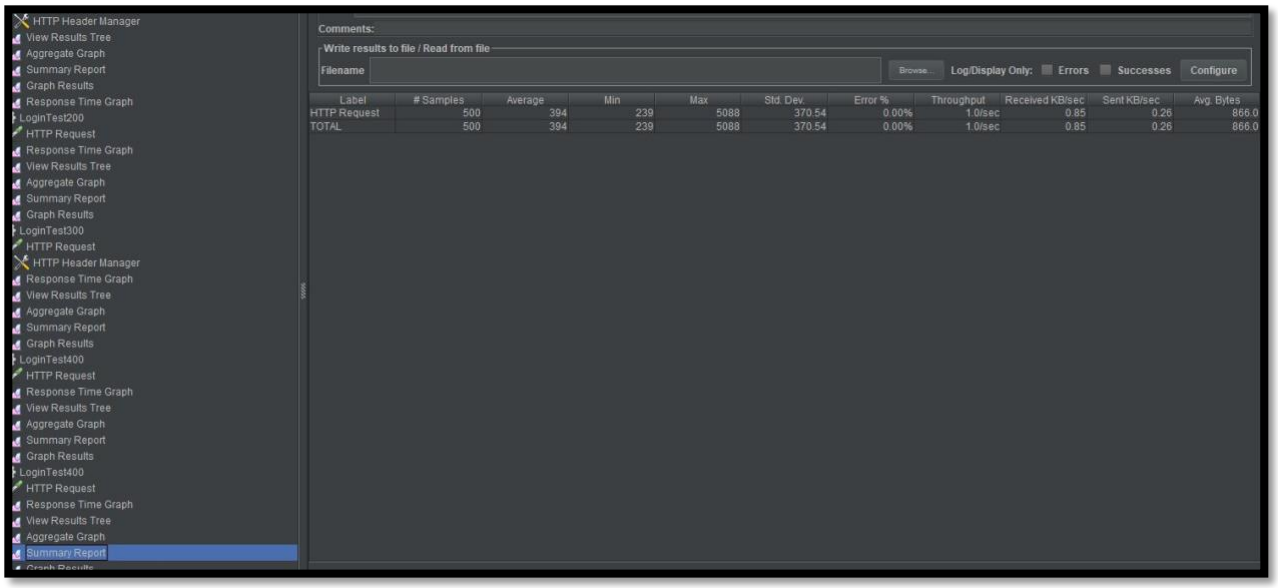


For 500 calls,


Write results to file / Read from file
Filename
Browse...
Log/Display Only: Errors Successes
Configure
Label # Samples Average Median 90% Line 95% Line 99% Line Min Maximum Error % Throughput Received KB/... Sent KB/sec
HTTP Request 500 394 340 483 498 1685 239 5088 0.00% 1.0/sec 0.85 0.26
TOTAL 500 394 340 483 498 1685 239 5088 0.00% 1.0/sec 0.85 0.26
Settings
Graph
Display Graph
Save Graph
Save Table Data
Save Table Header
Column settings
Columns to display: Average Median 90% Line 95% Line 99% Line Min Max
Foreground color
Value font: Sans Serif Size: 10 Style: Normal
Draw outlines bar? Show number grouping? Value labels vertical?
Column label selection:
Apply filter
Case sensitive
Regular exp.
Title
Graph title:
Synchronize with name
Font: Sans Serif Size: 16 Style: Bold
Graph size
Dynamic graph size Width: Height:
X Axis
Max length of x-axis label:
Y Axis (milli-seconds)
Scale maximum value:
Legend
Placement: Bottom Font: Sans Serif Size: 10 Style: Normal

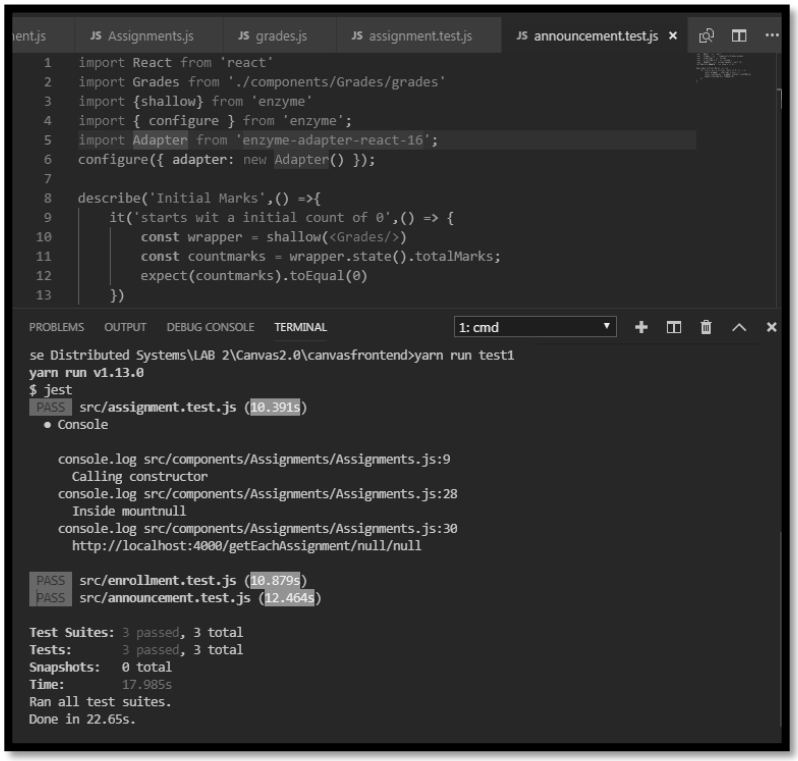

Filename
Browse...
Log/Display Only: Errors Successes
Configure
Graphs to Display Data Average Median Deviation Throughput
519 ms
0 ms
No of Samples 500
Latest Sample 245
Average 394
Deviation 370
Throughput 60.061/minute


Settings
Graph
Response Time Graph
Milliseconds
3,000
2,700
2,400
2,100
1,800
1,500
1,200
900
600
300
0
15:50:40
15:50:50
15:51:00
15:51:10
15:51:20
15:51:30
15:51:40
15:51:50
15:52:00
15:52:10
15:52:20
15:52:30
15:52:40
15:52:50
HTTP Request



- Mocha Testing

- Enzyme



- Question&Answers:

1. Kafka improves the response time drastically. If there are many users making a request onto a server like 1000 requests, the server cannot handle all the requests at a time. But, Kafka, A messaging queue with its low latency, high throughput does handle the incoming huge requests. MongoDB in cloud has automated load balancing by using sharding, but locally, it is quite a tough task and has a lot of complexity.
2. MySQL is the database model which consists of rows and columns and MongoDB consists of documents. Query of data is comparatively easy in MongoDB than in MySQL as it may have triggers and joins. But in MongoDB, it is very fast. When data is extensively large, mongoDB is used as it is horizontally scalable and can handle unstructured data and also can be accessed frequently. Whereas in MySQL is scalable vertically and can be used on small datasets.