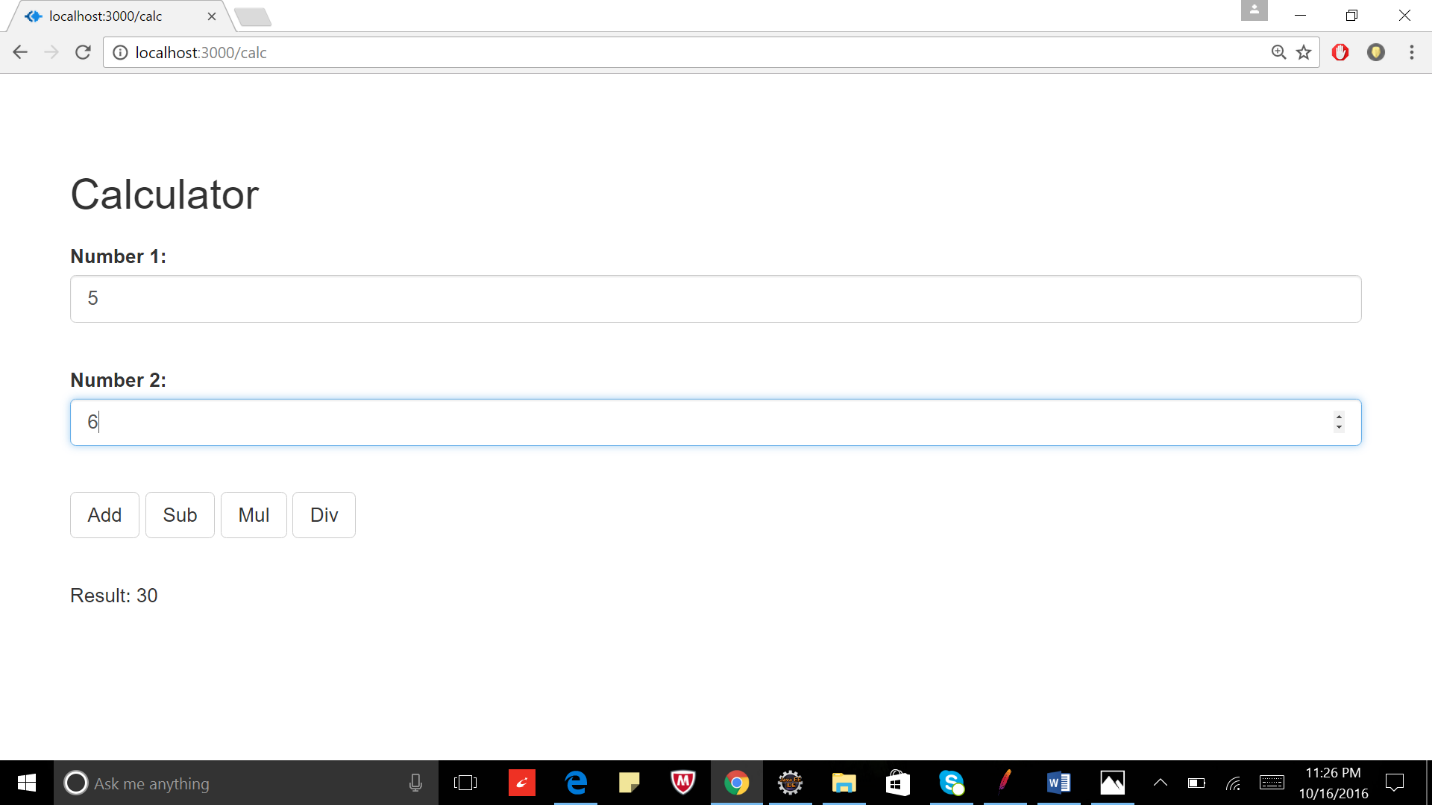
**CMPE 273 LAB 1**

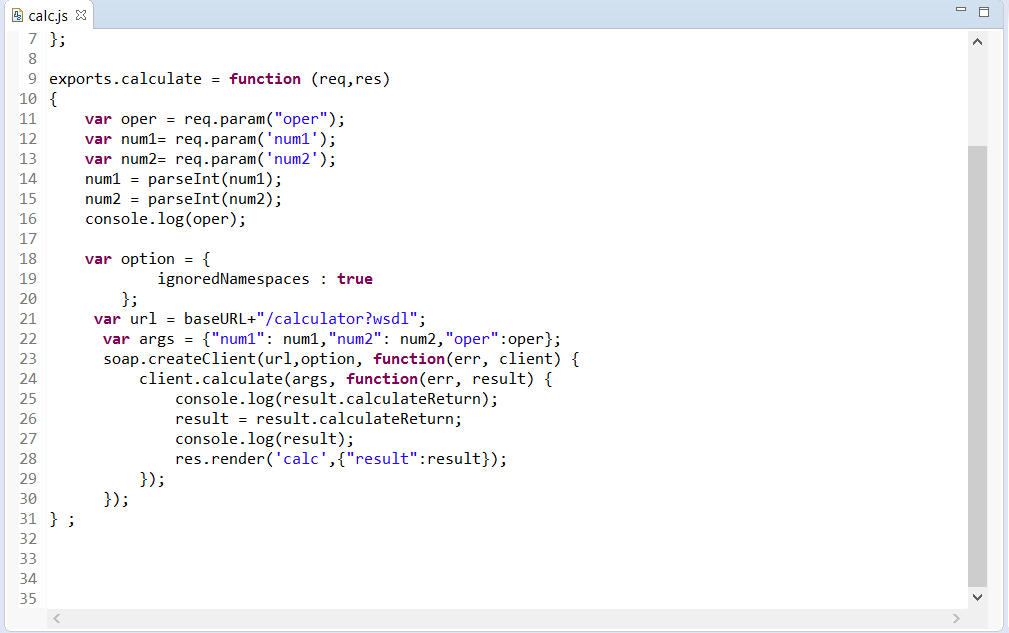
**Part 1- Calculator:**

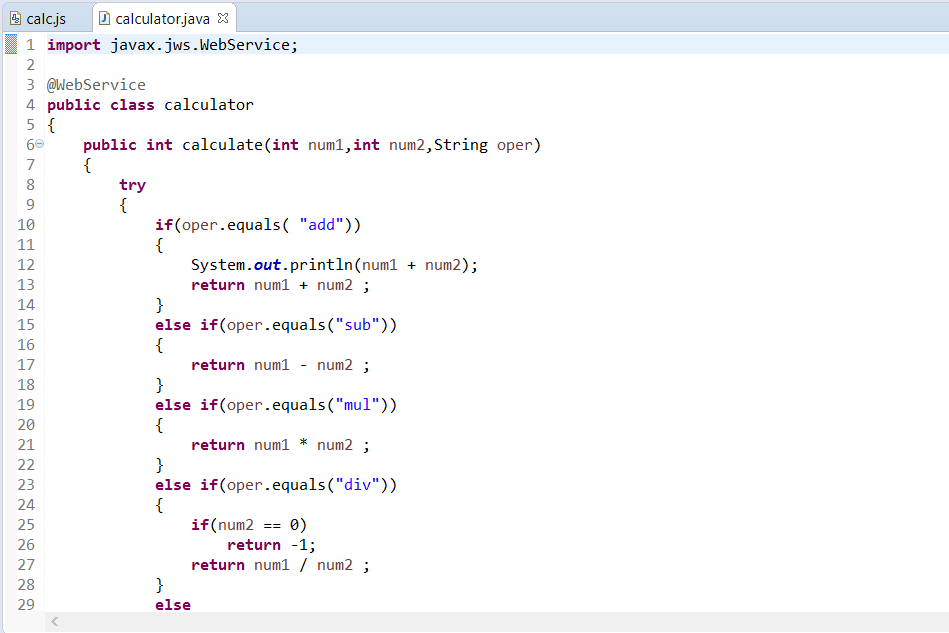
Introduction: To build a calculator with addition, subtraction, multiplication & division functionalities to demonstrate stateless web services.

System Design: The project is a Node express project as front end and SOAP web service as a backend both hosted on local host developed using eclipse IDE. It is tested using JMeter.

Results:

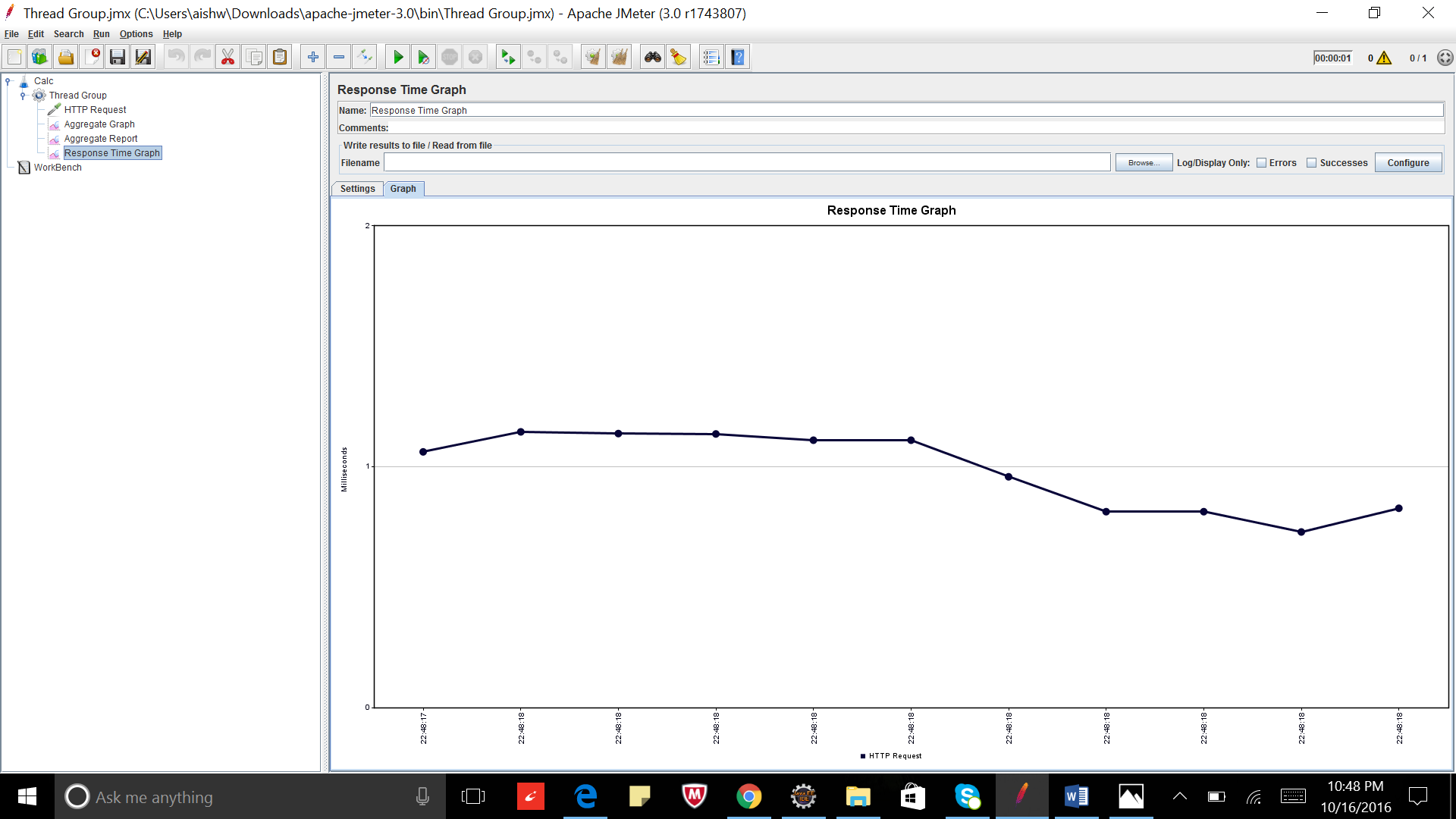




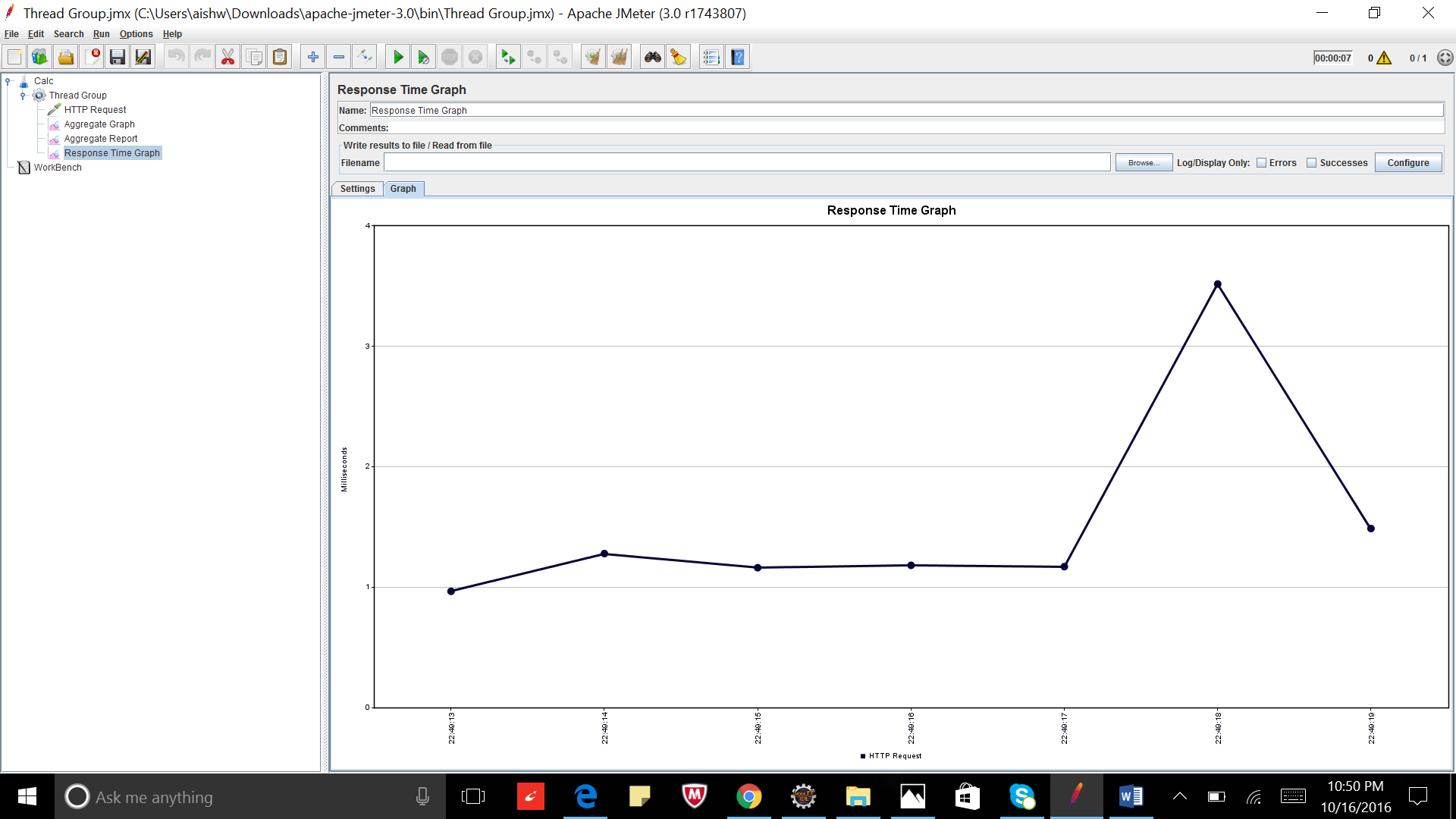


Performance Graph:

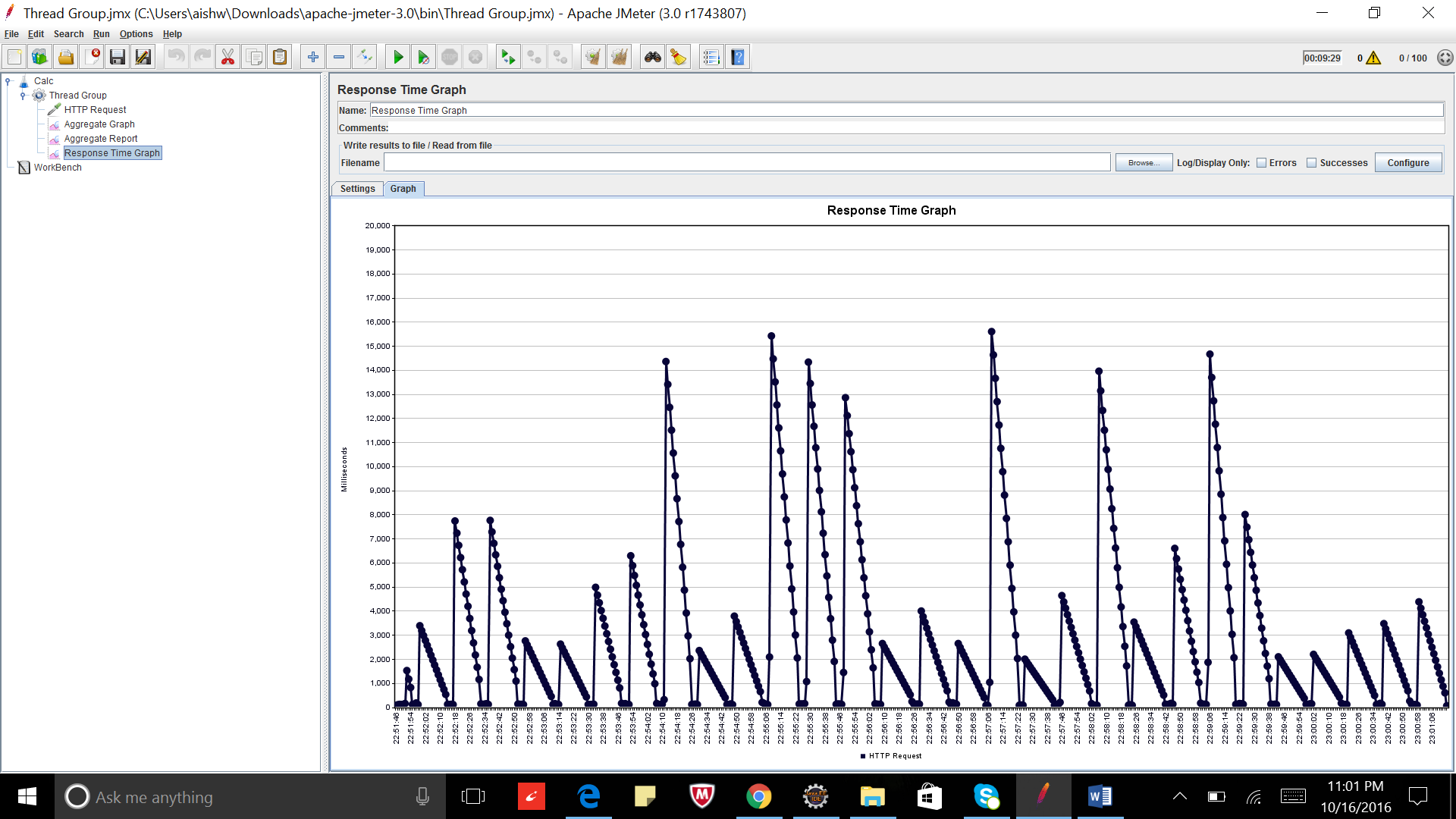
For 1000 calls:



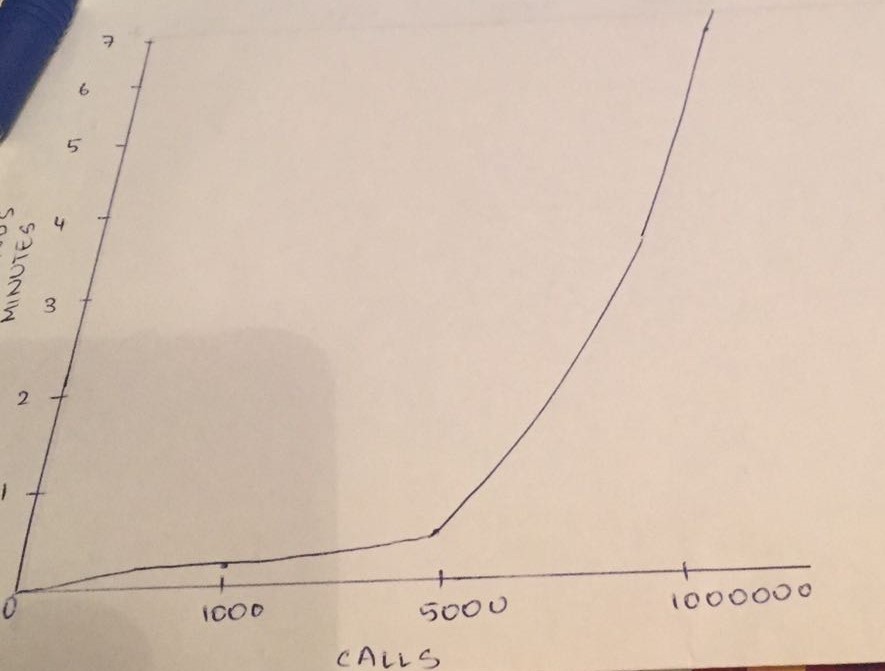
For 5000 calls:



For 1000 calls with 100 concurrent users:



**Average Times:**



**Observation**

The time required to implement random calls increased with the number of calls being made. When multiple users accessed the process concurrently, the implementation time increased exponentially.

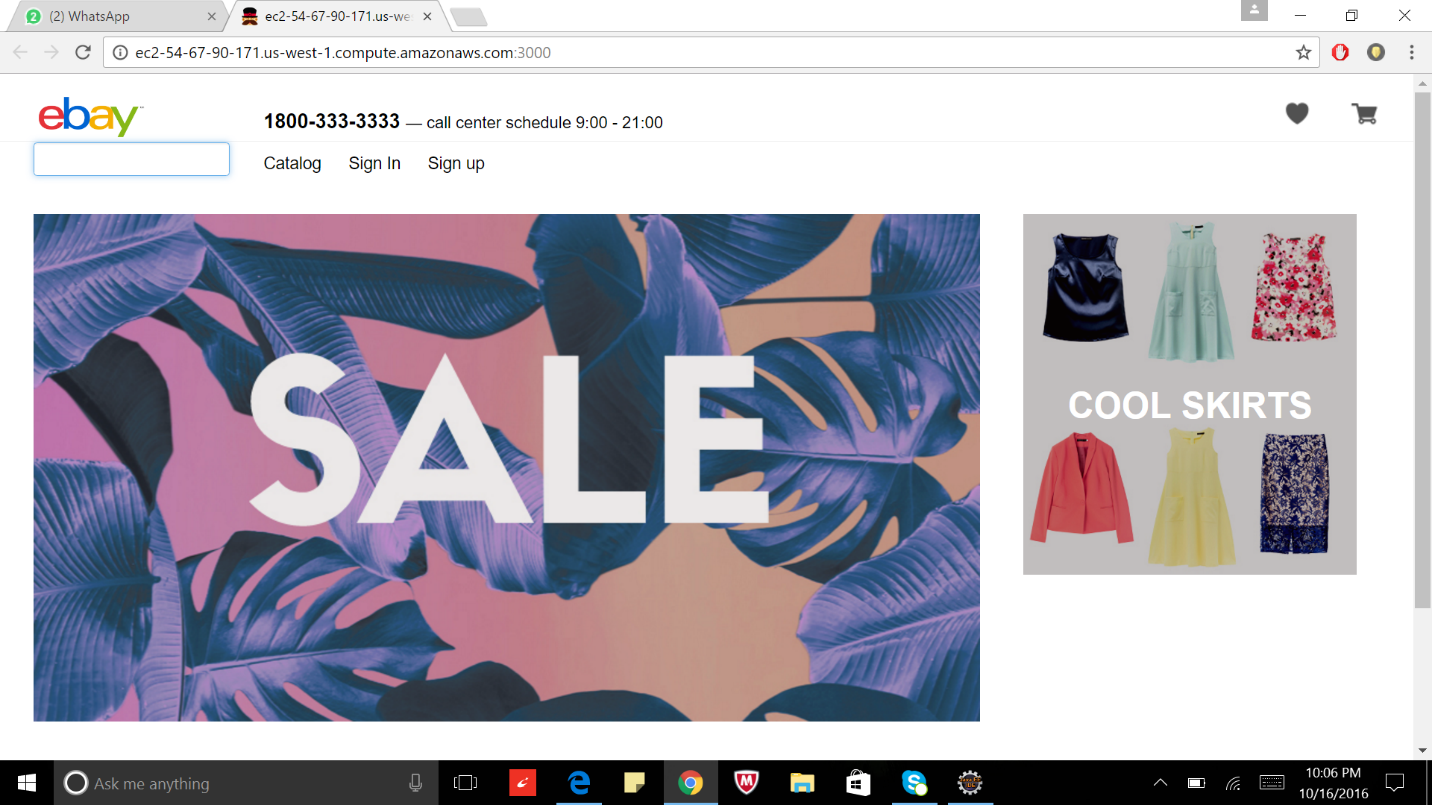
Comparing results to Lab1; when multiple user accessed the process concurrently; the SOAP web services were slower compare to RESTful web services as RESTful web services are stateless and no server side sessions are need to be maintained.

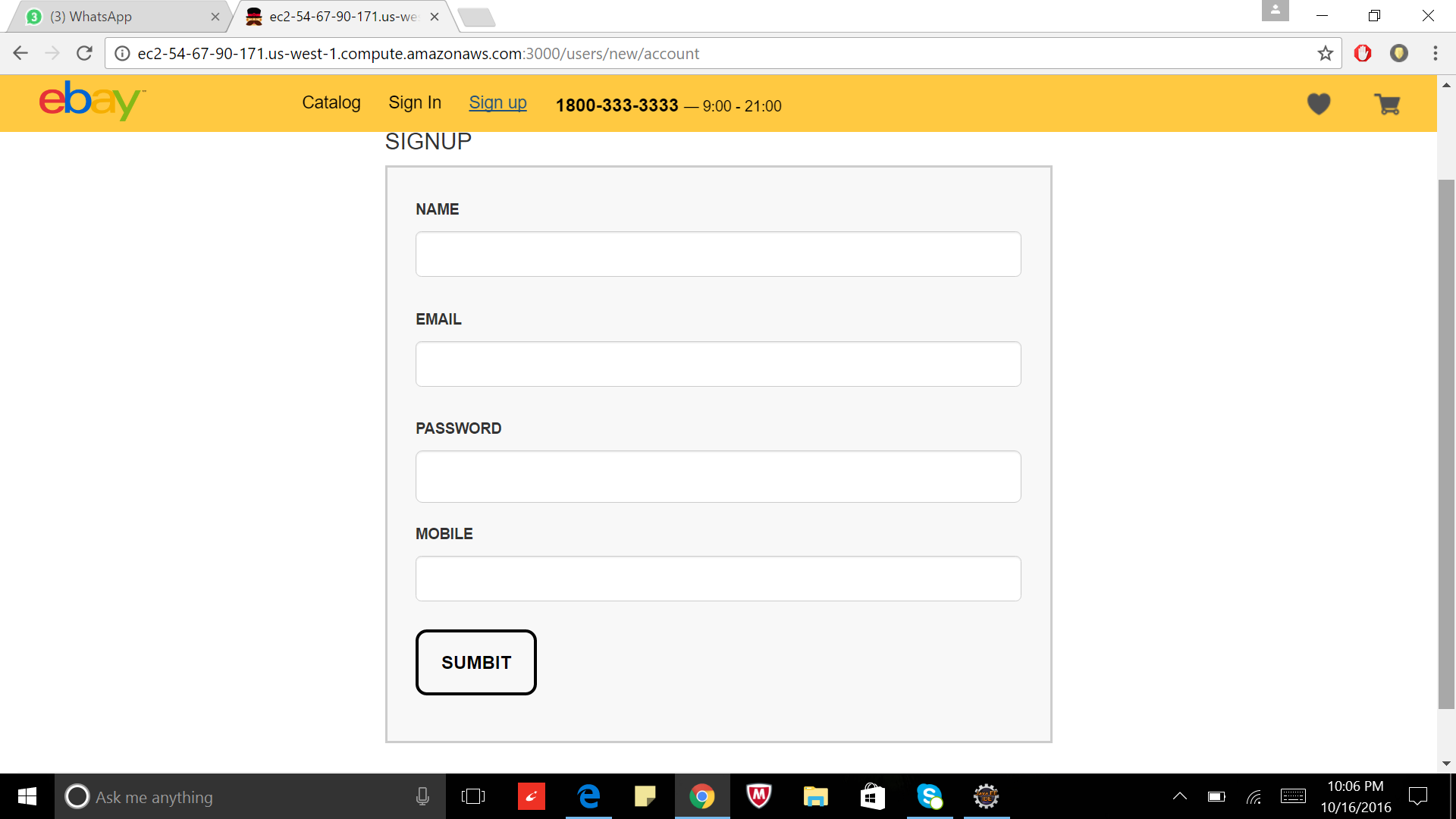
**Part 2 – Ebay Marketplace Application:**

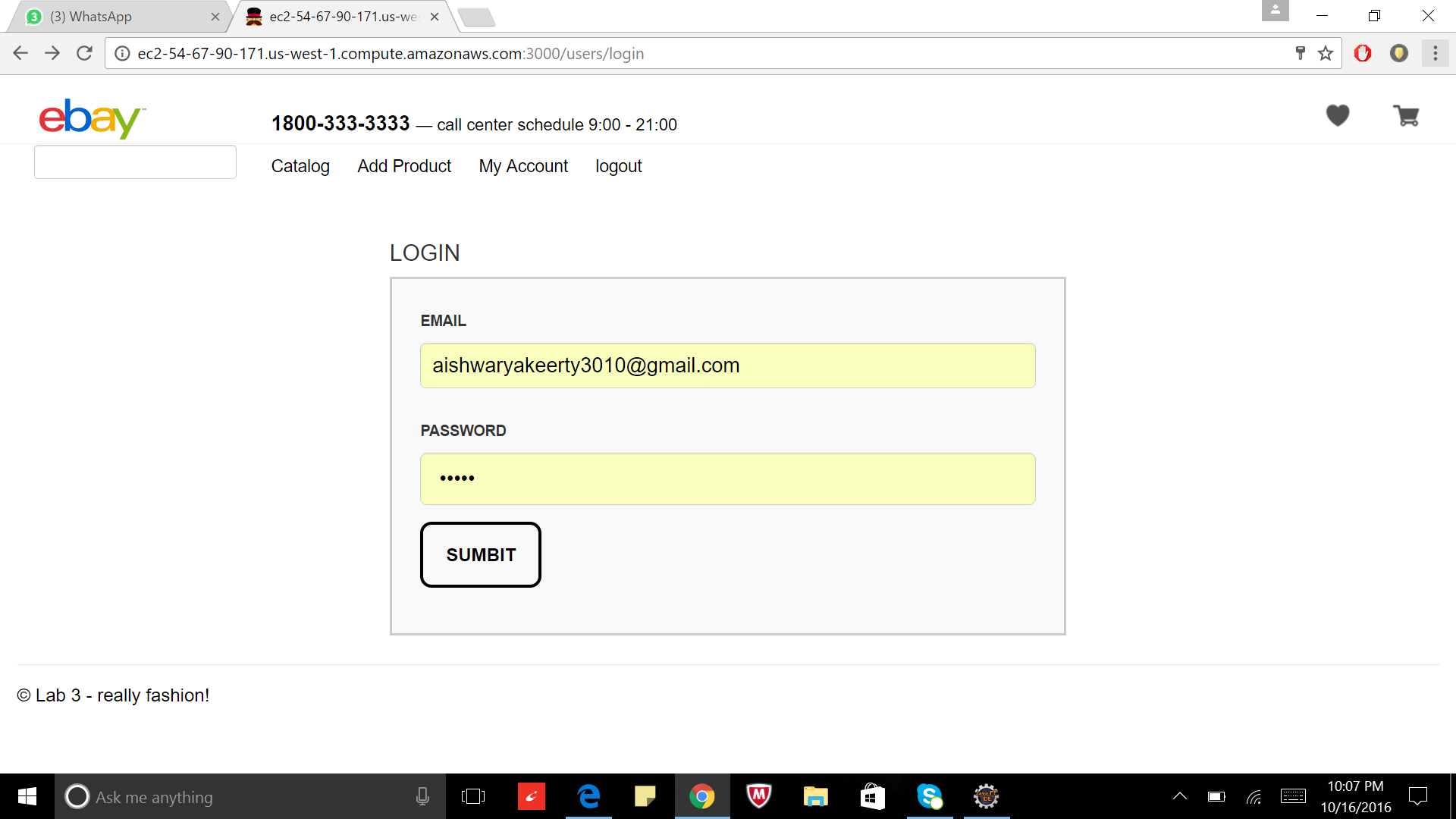
Introduction: To build a simple market place to demonstrate REST web services.

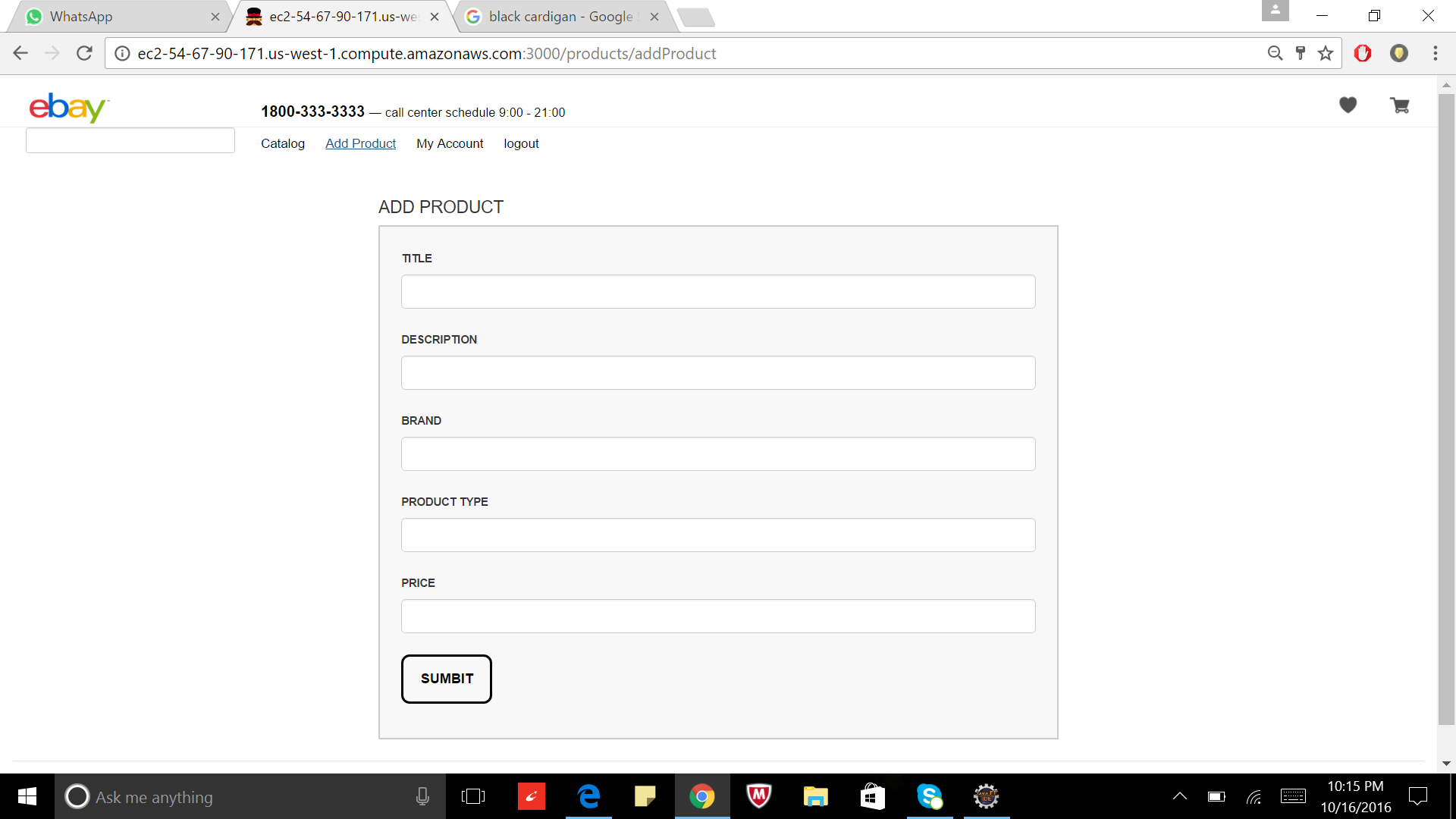
System Design: The client side is built using Nodejs, Angular js and HTML5 whereas the server side is built using Java SOAP web services. The database used is mysql and connection pooling is implemented. Project is tested using JMeter and Mocha.

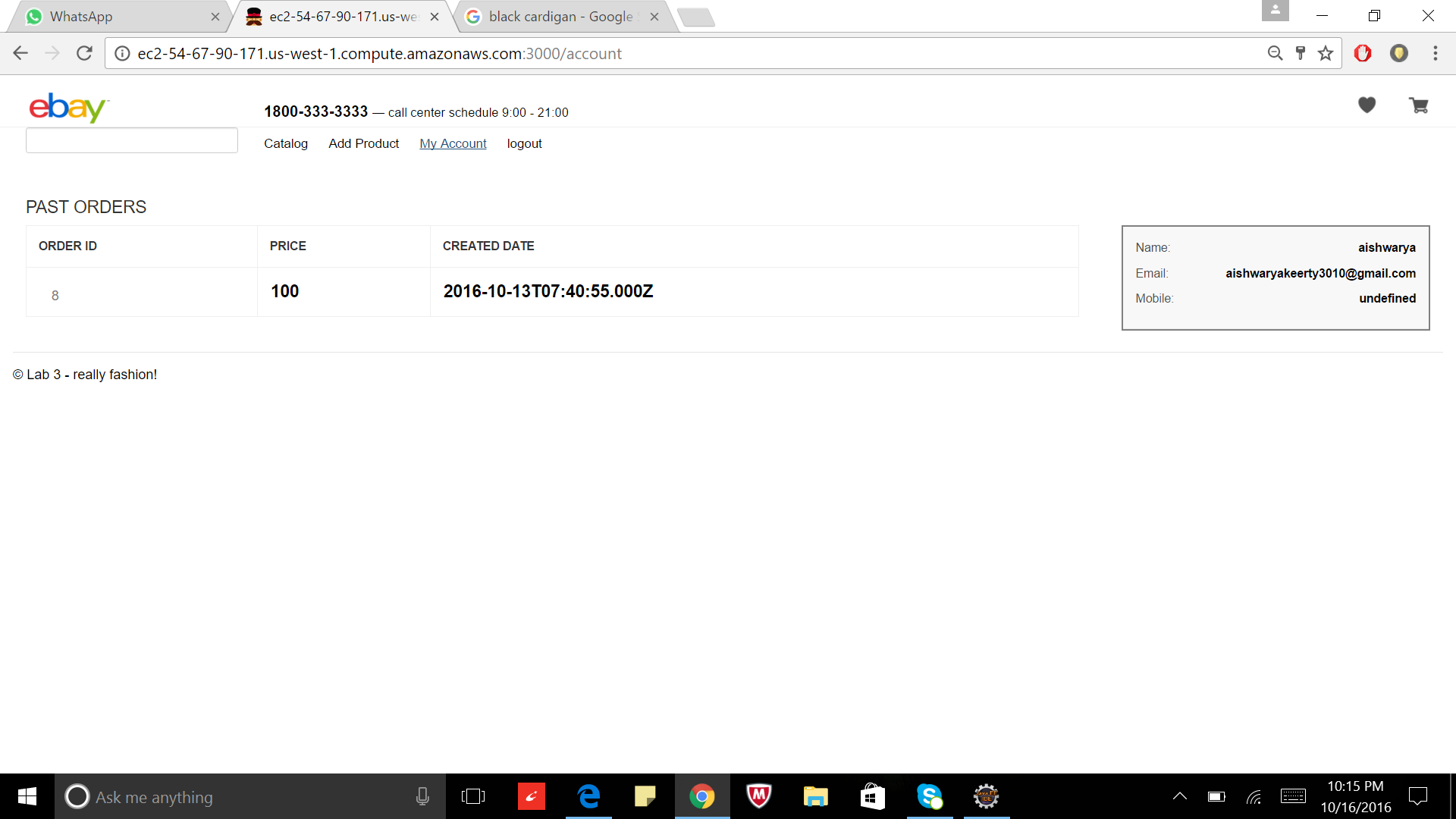
Results:

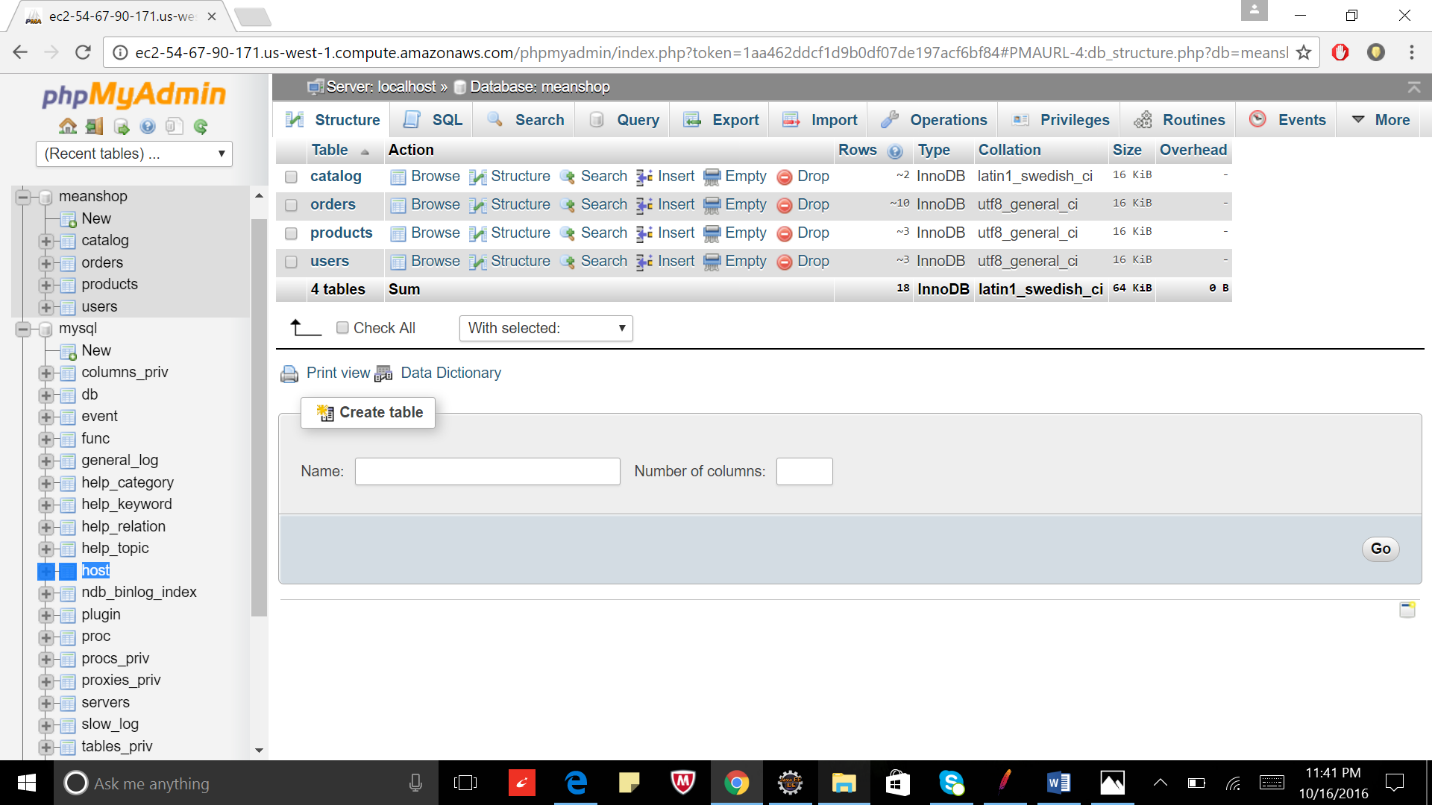


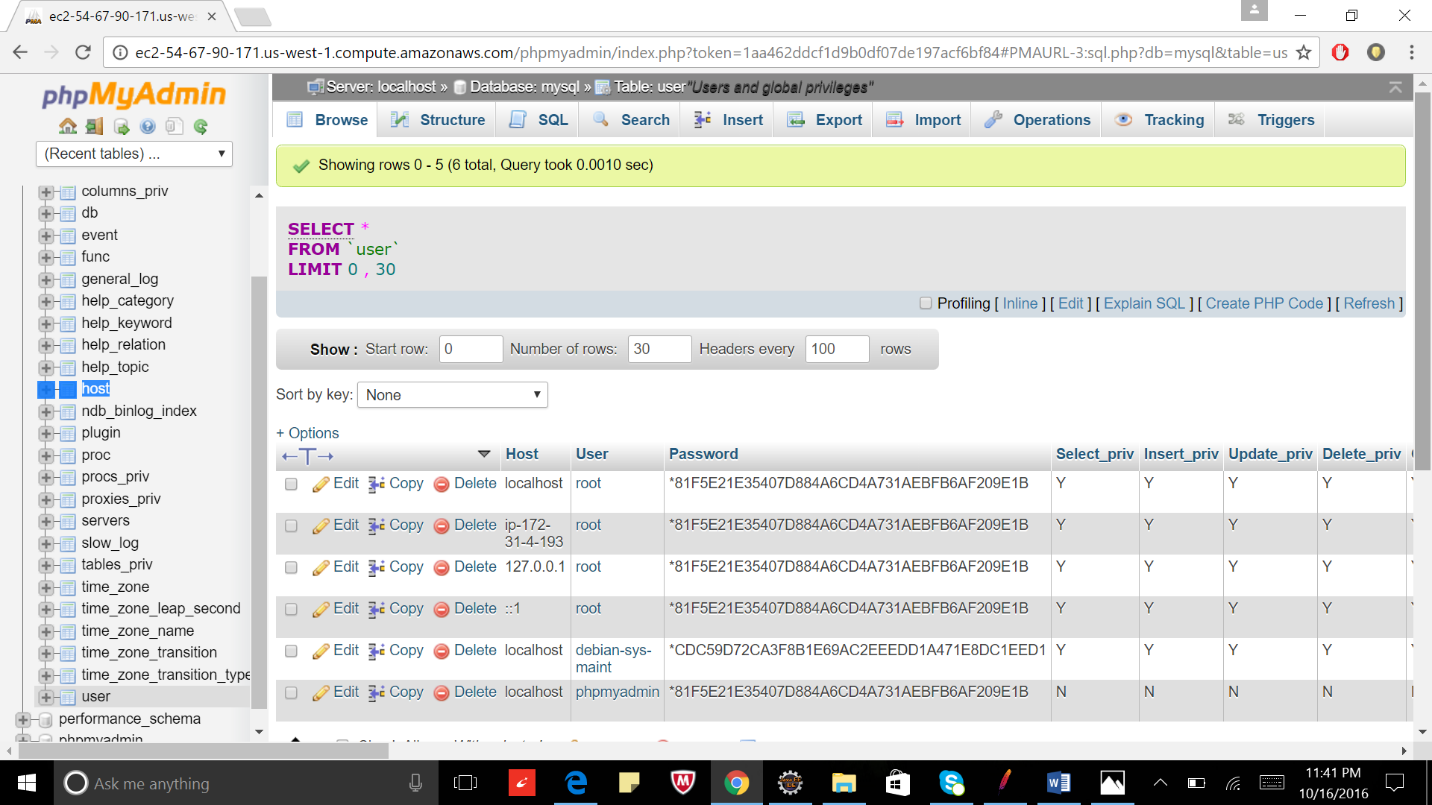


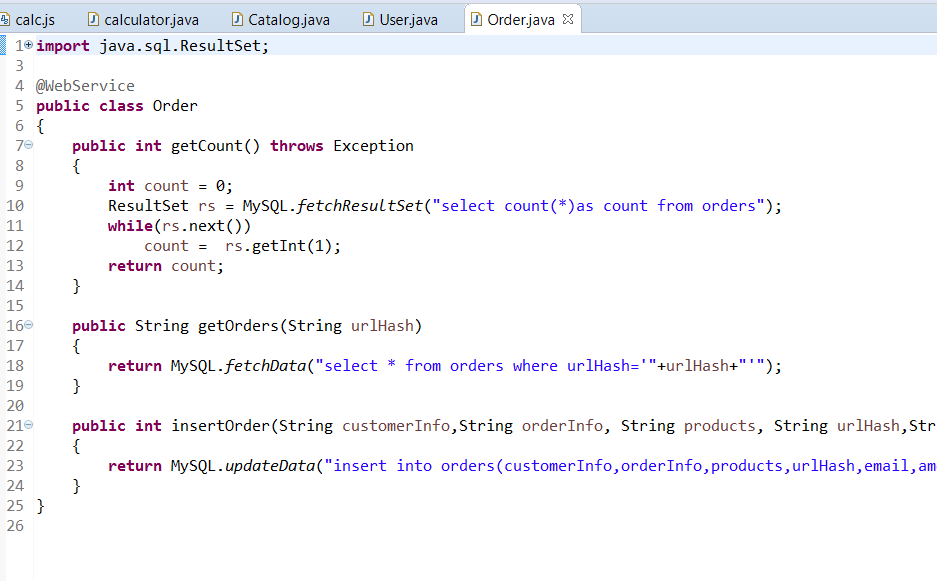






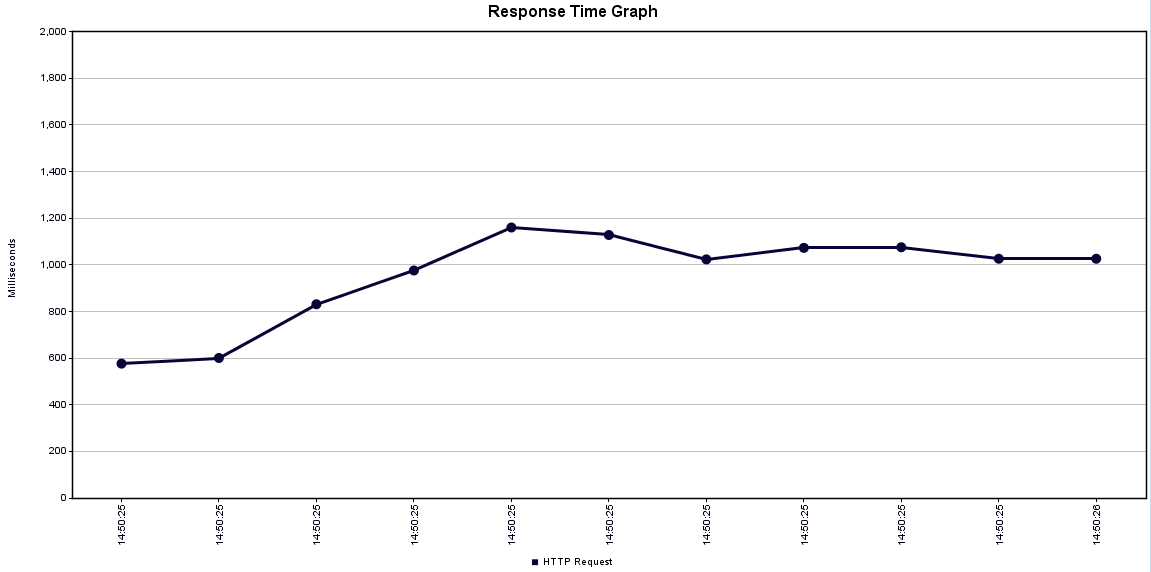




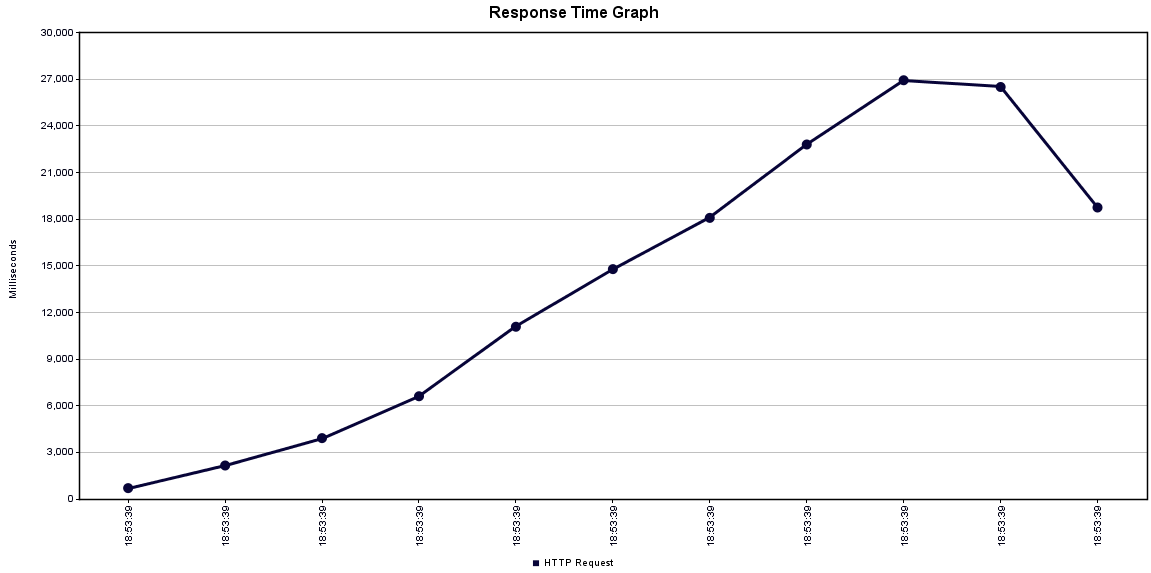


**Performance Graphs**

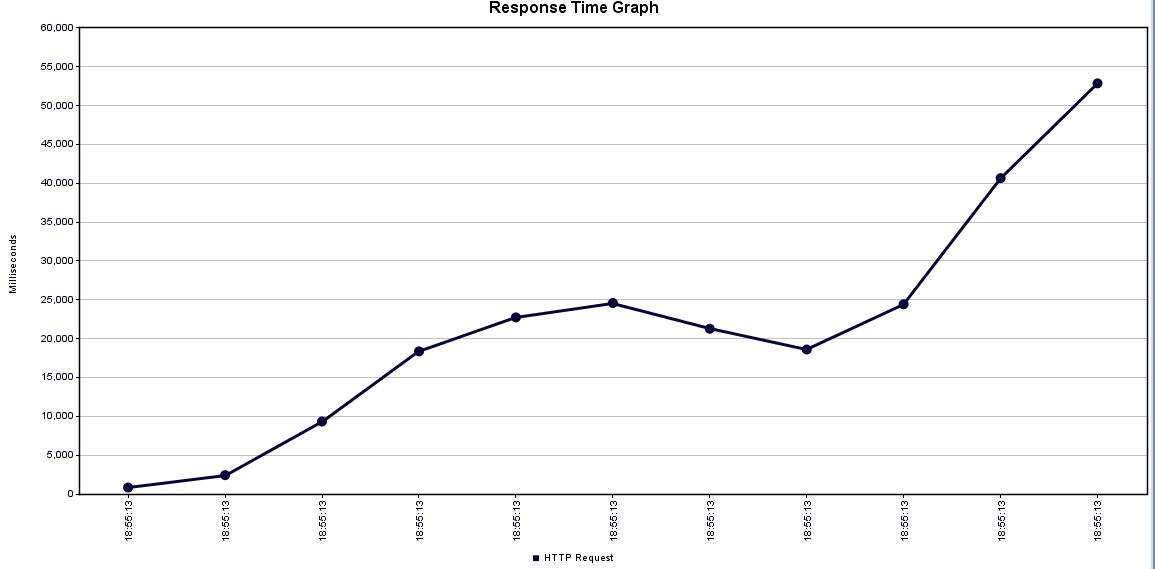
100 concurrent users without connection pool



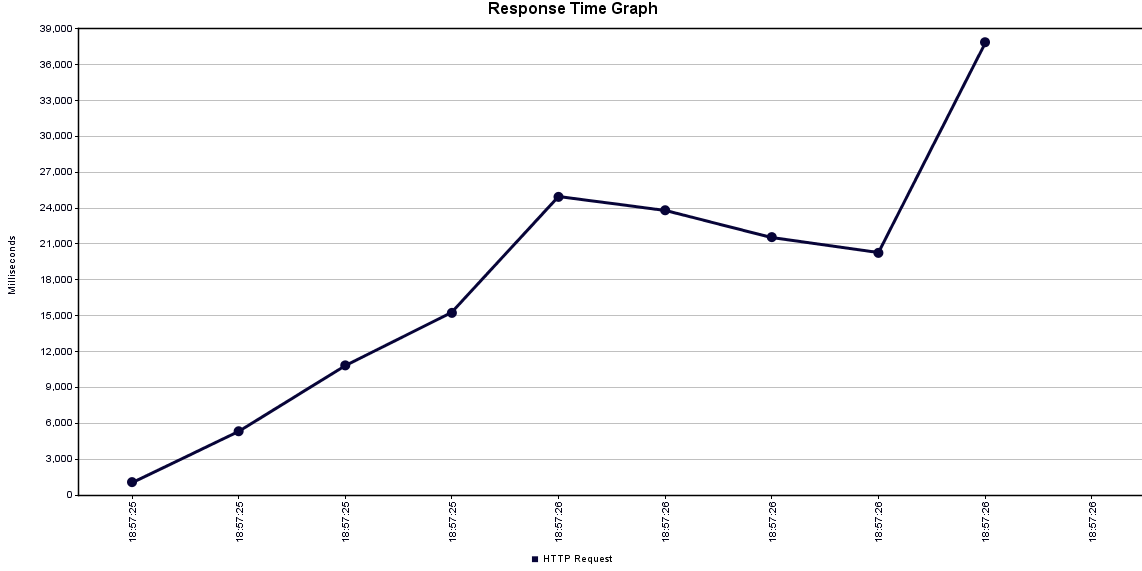
200 concurrent users without connection pool



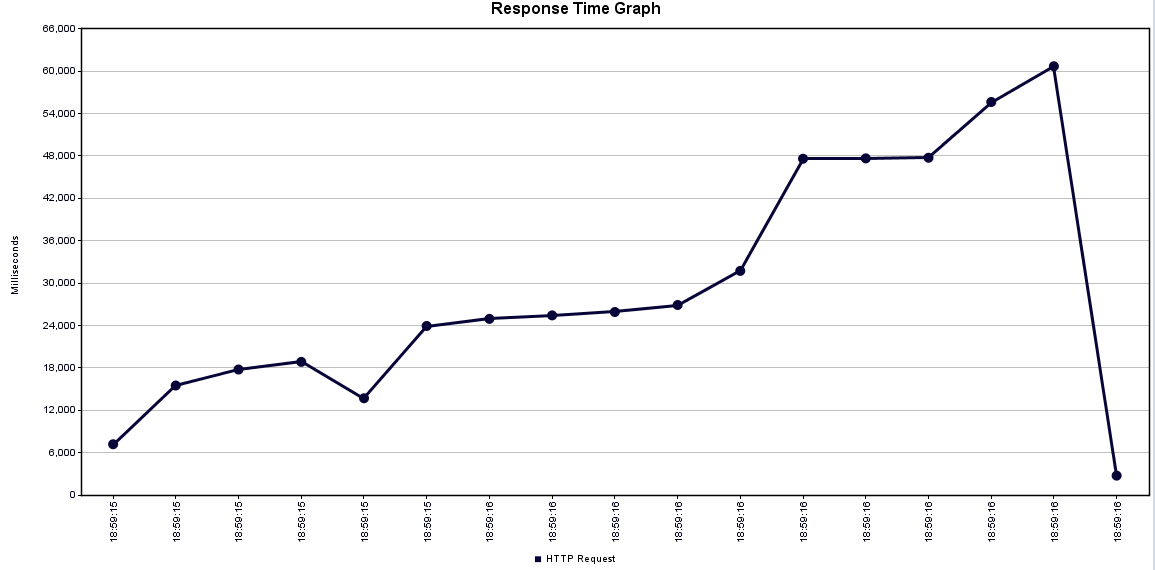
300 concurrent users without connection pool



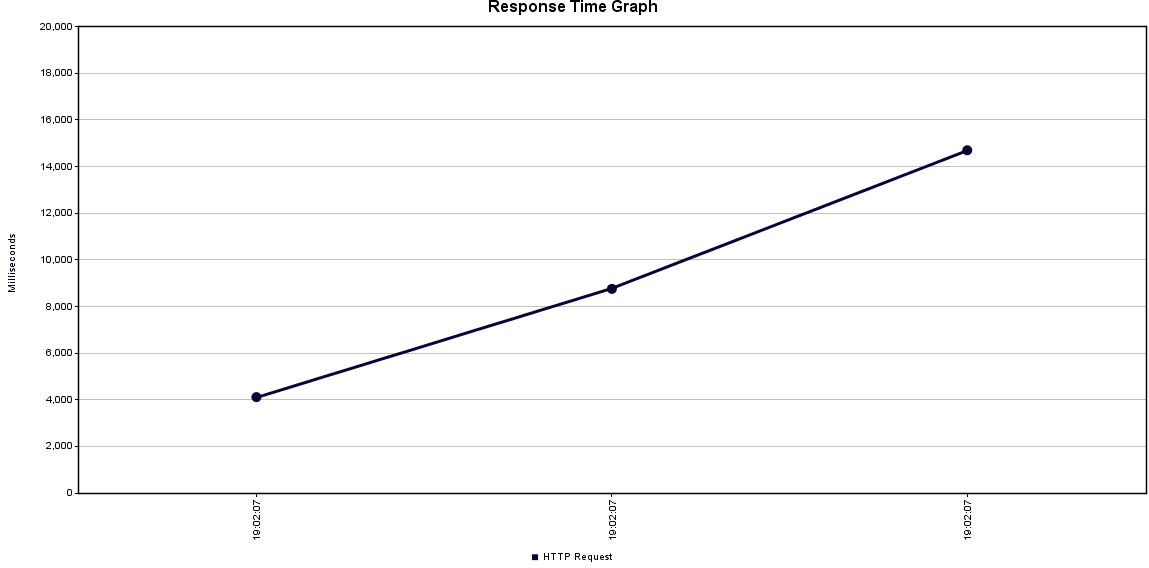
400 concurrent users without connection pool



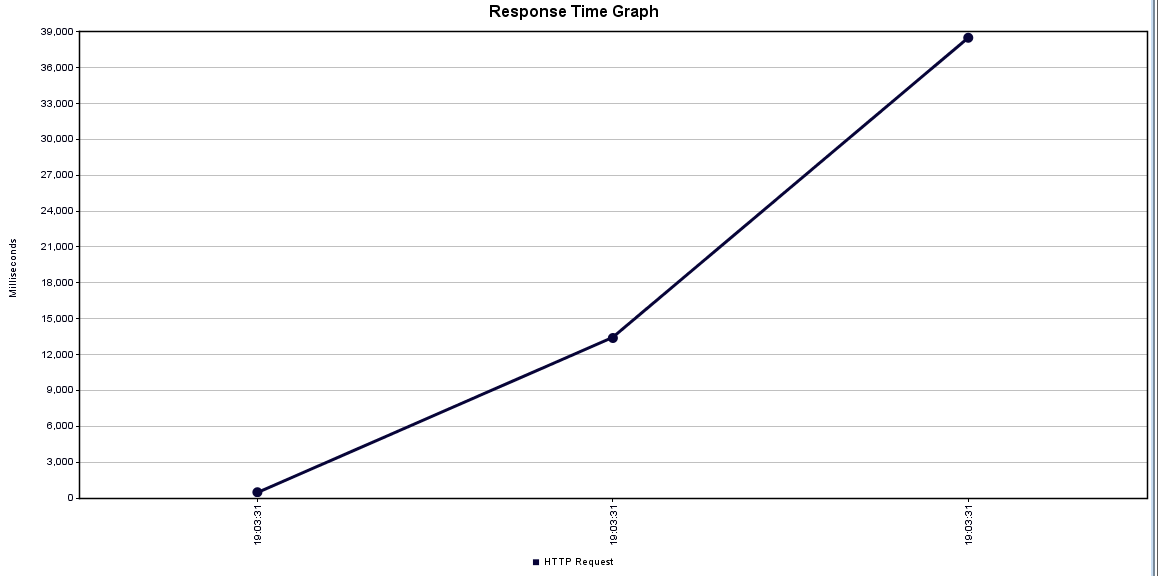
500 concurrent users without connection pool



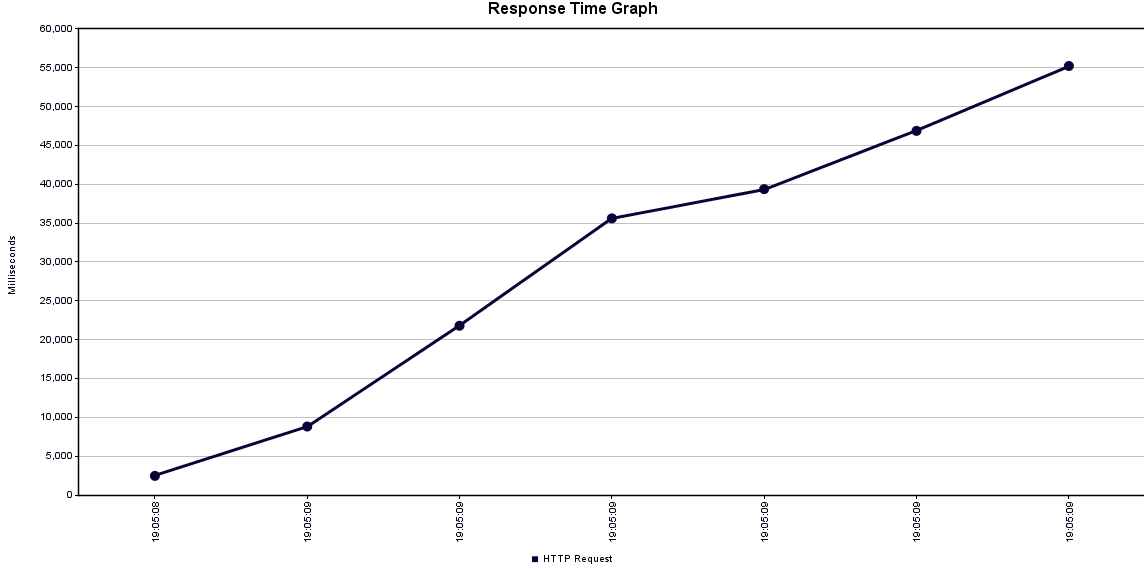
100 concurrent users with connection pool



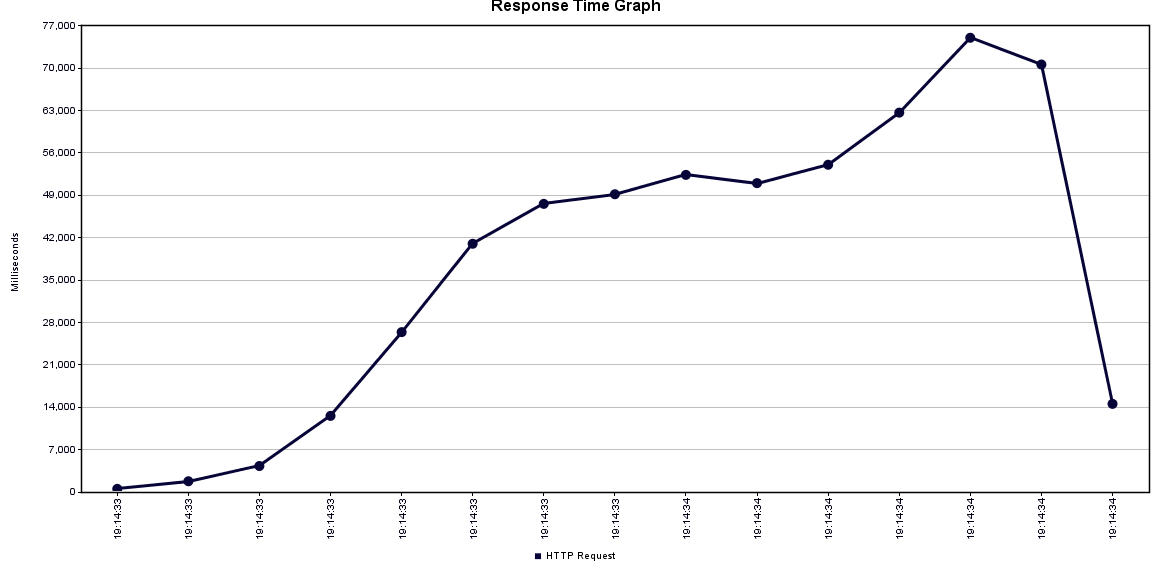
200 concurrent users with connection pool



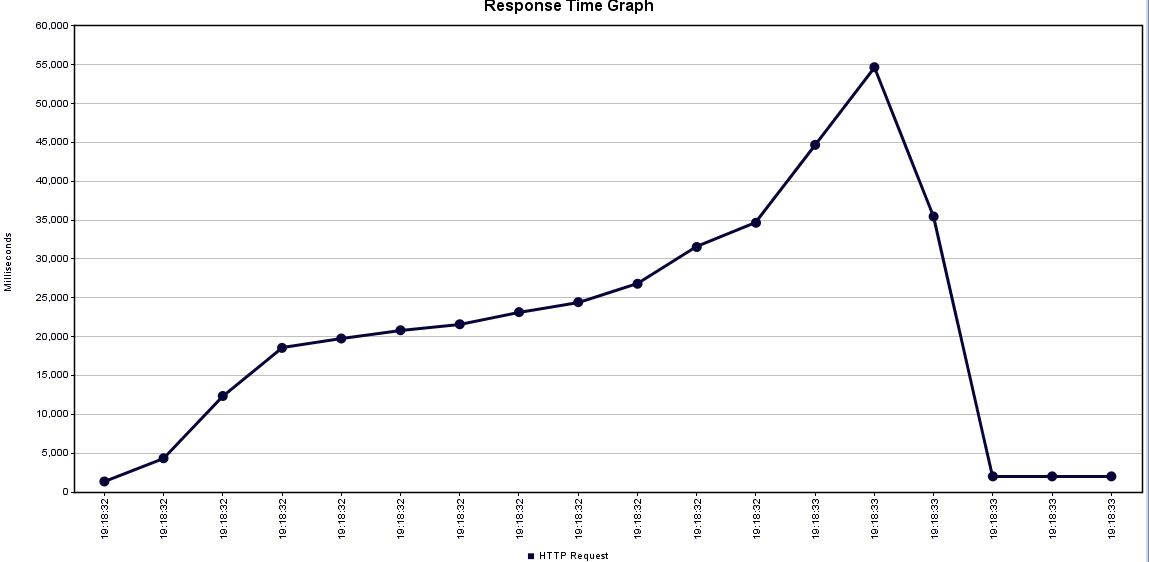
300 concurrent users with connection pool



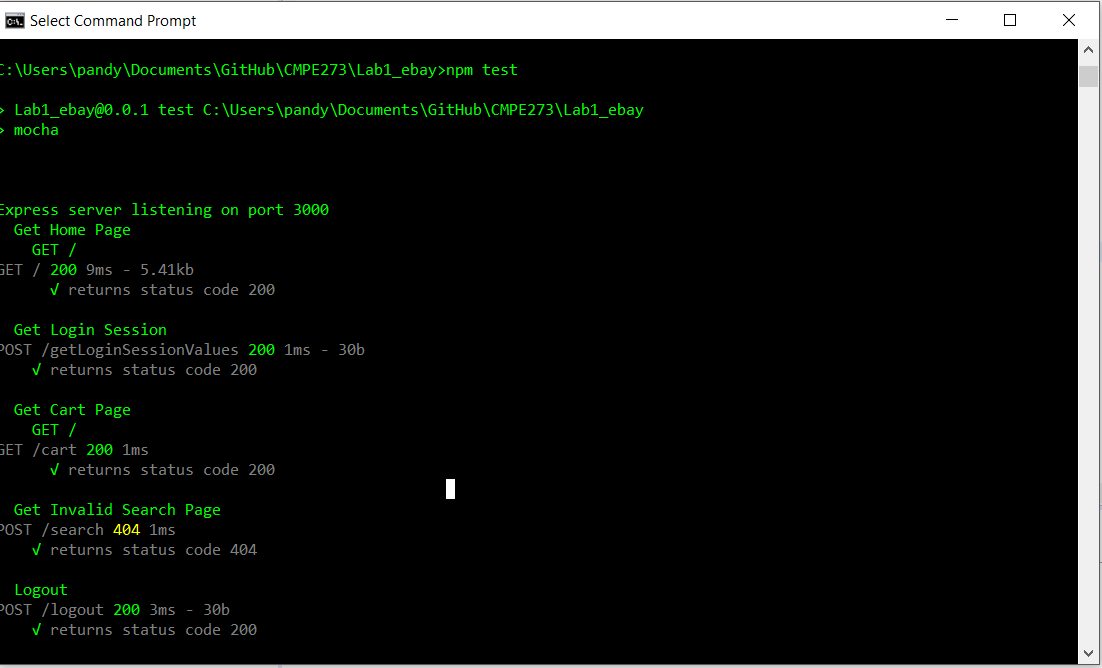
400 concurrent users with connection pool



500 concurrent users with connection pool



**Mocha Test**





**Comparison:**

Comparing results to Lab1; when multiple user accessed the process concurrently; the SOAP web services were slower compare to RESTful web services as RESTful web services are stateless and no server side sessions are need to be maintained.

git@github.com:Aishwaryakeerty/Lab3.git