**LAB2**

Student ID:010694814

Name:Aparna Kulkarni

**Introduction:**

**Goal**: To learn MEAN stack i.e. nodejs,angularjs,RabbitMq and express framework imlementation of Facebook is done in this lab.Facebook application uses MongoDB to store the database and nodejs angularjs are used for front end implementation.Where as RabbitMQ is used for message queing.

**Purpose of system:**

**Facebook:**

* Creating a RESTful service.
* SignIn,SignUp,SignOut functionality
* Encrypting the password for security purpose
* Implementing Sessions for a user usign MongoDB
* Displaying About,Interests functions for a user
* User can see friendlist,send friend request,accept friend request
* User can create and delete Group,add and delete,view members to group
* Provide news feed functionality

**List of API's for Facebook:**

* Login:Helps to login into the system and maintains a session
* SignUp:User is signed up with the help of this function
* logout:User can logout and session will be deleted for the user
* UserProfile:Displays The main frontpage of facebook with news feed
* FriendList:Displays List of Friends
* Interests:Displays Interests like (Music,shows,Sports)
* ProfilePageNavigate:Displays the user profile page
* GroupPageNavigate:Displays the list of groups

**System Design:**

System design is the process of defining the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system.

In the current system I have used following systems:

NodeJS:Used as a an event-driven , non-blocking I/O model that makes applications lightweight and efficient.Its goal is to provide an easy way to build scalable network programs

AngularJS: It is framework for client-side [model–view–controller](https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller) (MVC) and [model–view–viewmodel](https://en.wikipedia.org/wiki/Model_View_ViewModel) (MVVM) architectures. It is a structural framework for dynamic web apps.It is focusing on creating single page applications .

MongoDB:It is a document oriented database.It is a NoSQL database.It is open source,schema free, horizontally scalable database.

RabbitMQ:It provides robust messaging for applications.It is easy to use and runs on all operating systems.

bcrypt:It is used for encryption of password so that it is secured.

Data design: It represents how data is stored within the system.

User interface design: It represents how user’s information flows to the system and back using the view provided by the User interface.

Process design: It defines how data moves through the system.

I have used Model-View-Ccontroller (MVC) architecture so that database logic,the business logic, presentation represents different decoupled layers.

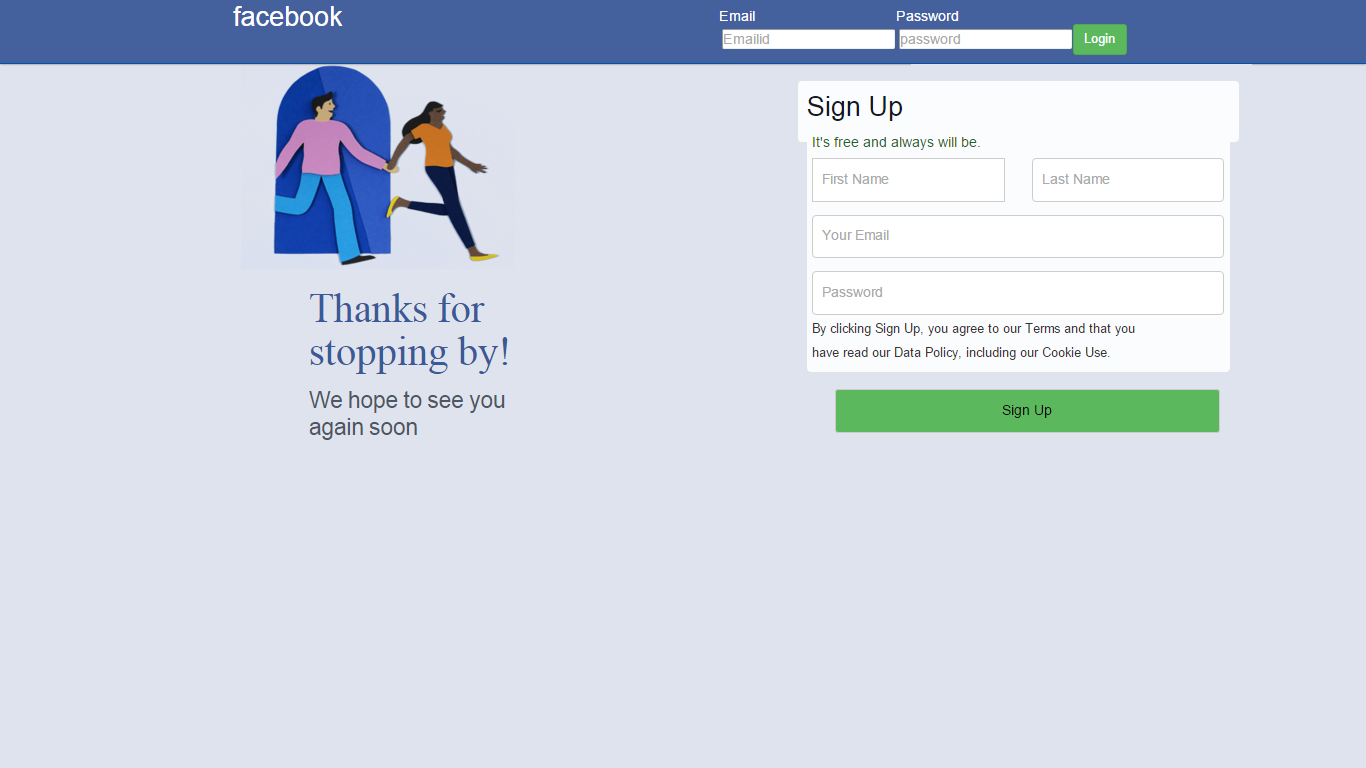


**Facebook Application:**

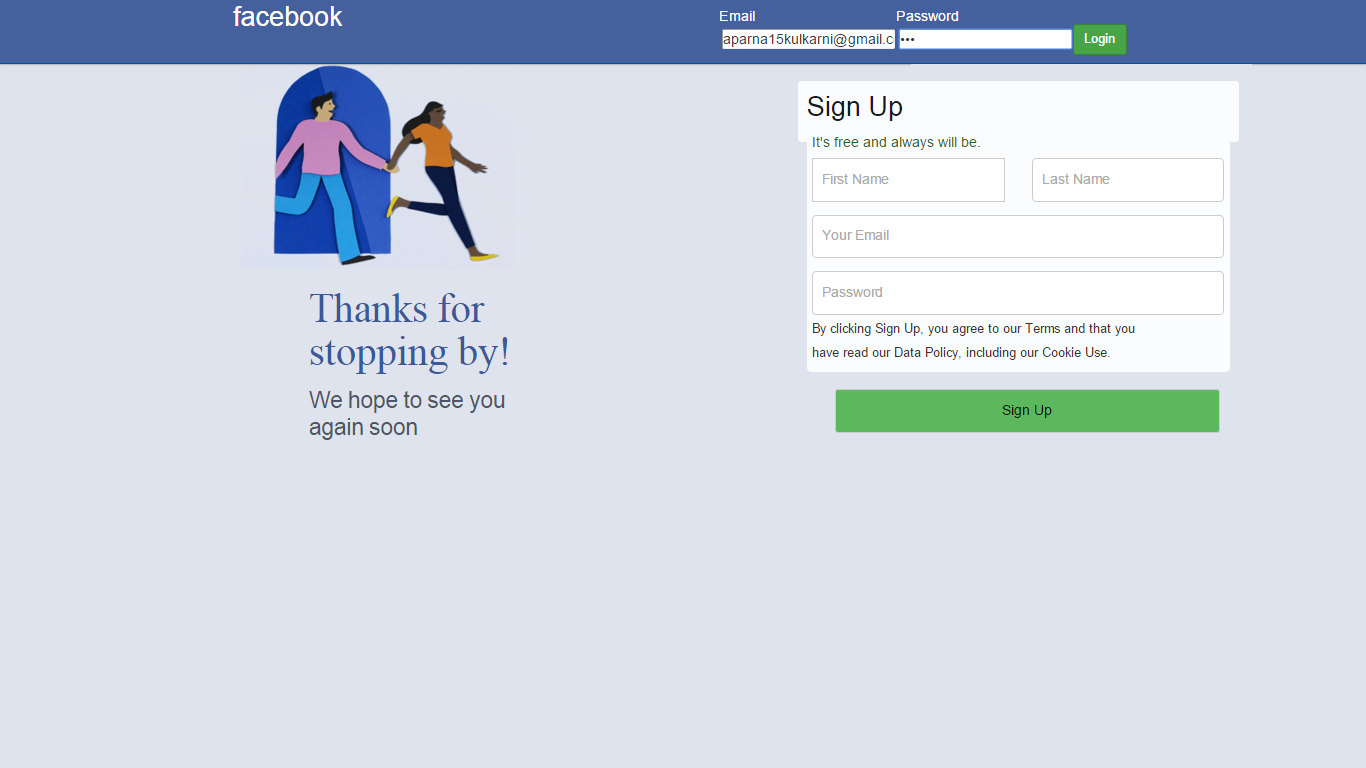
* When user hits the url signIn/signUp page will be displayed
* When user is not registered with the system he has to signUp first
* After signUp user can login using their own credentials
* When the user is successfully signed in he/she will see the home page which displays news feed and buttons to navigate to About page/Friends page/Groups page etc
* When user selects Profile page(About) he/she will be able to view all the details of the user like about user,interests of a user etc.
* Friends page displays the list of friends and friend requests if any to accept
* Groups page will display the list of groups of user
* User can create or delete a group and also add/delete/view group members.
* After Logout button click user session will be terminated and user will be logged out.

**Results:**

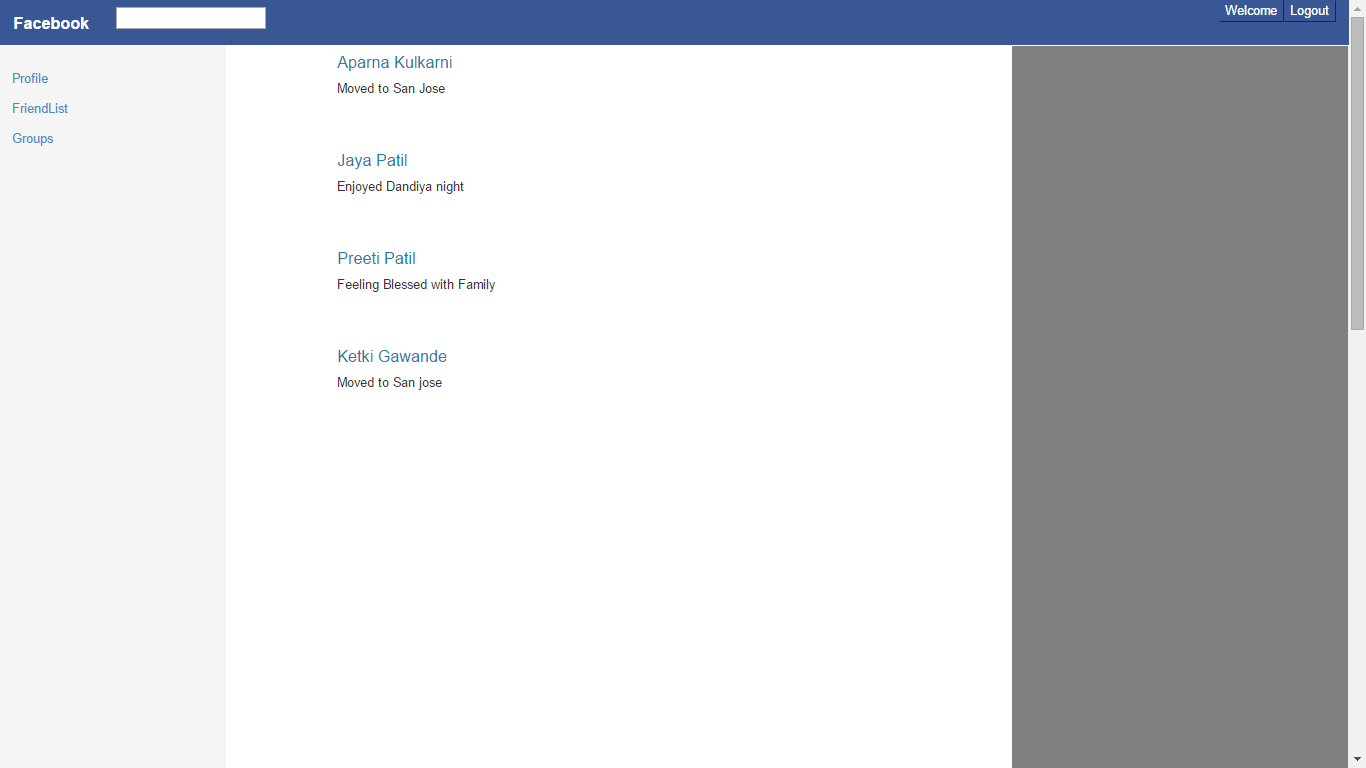
**Facebook**



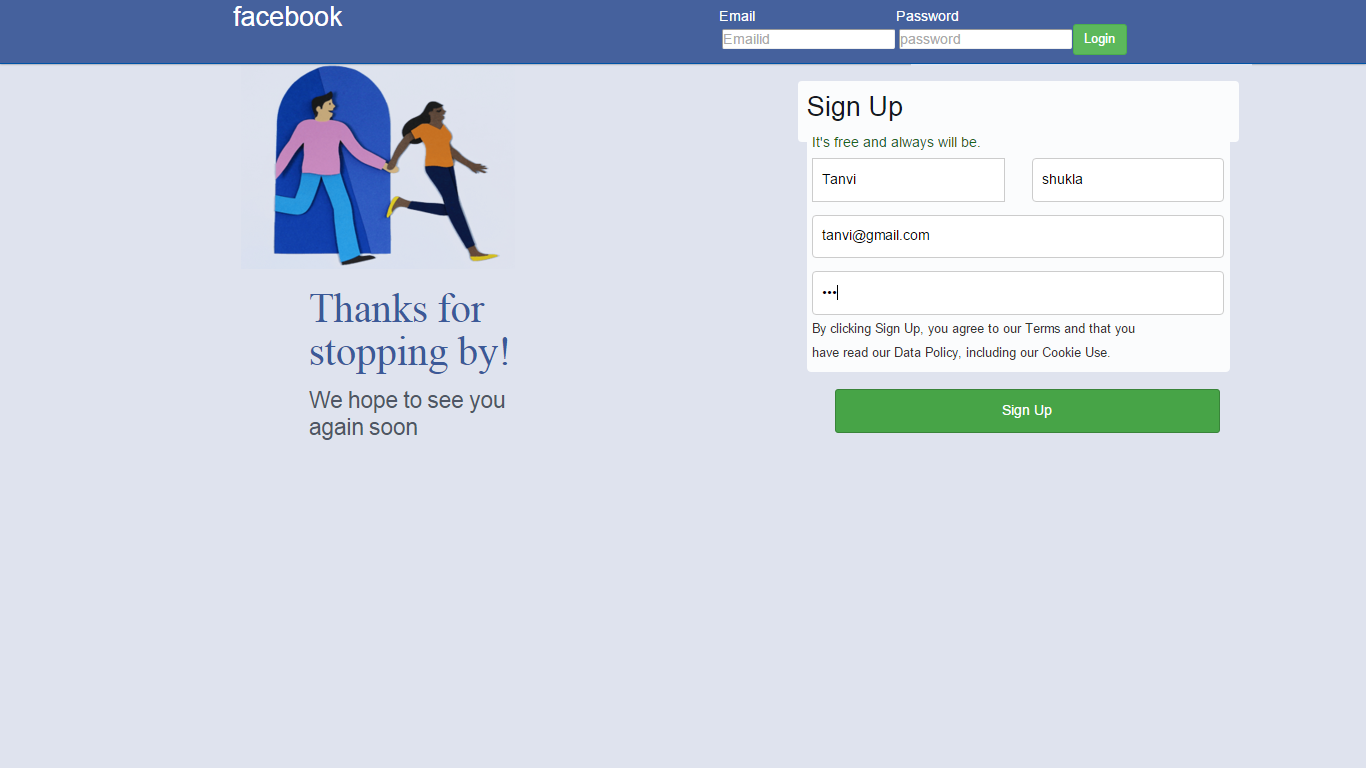
SignIn user

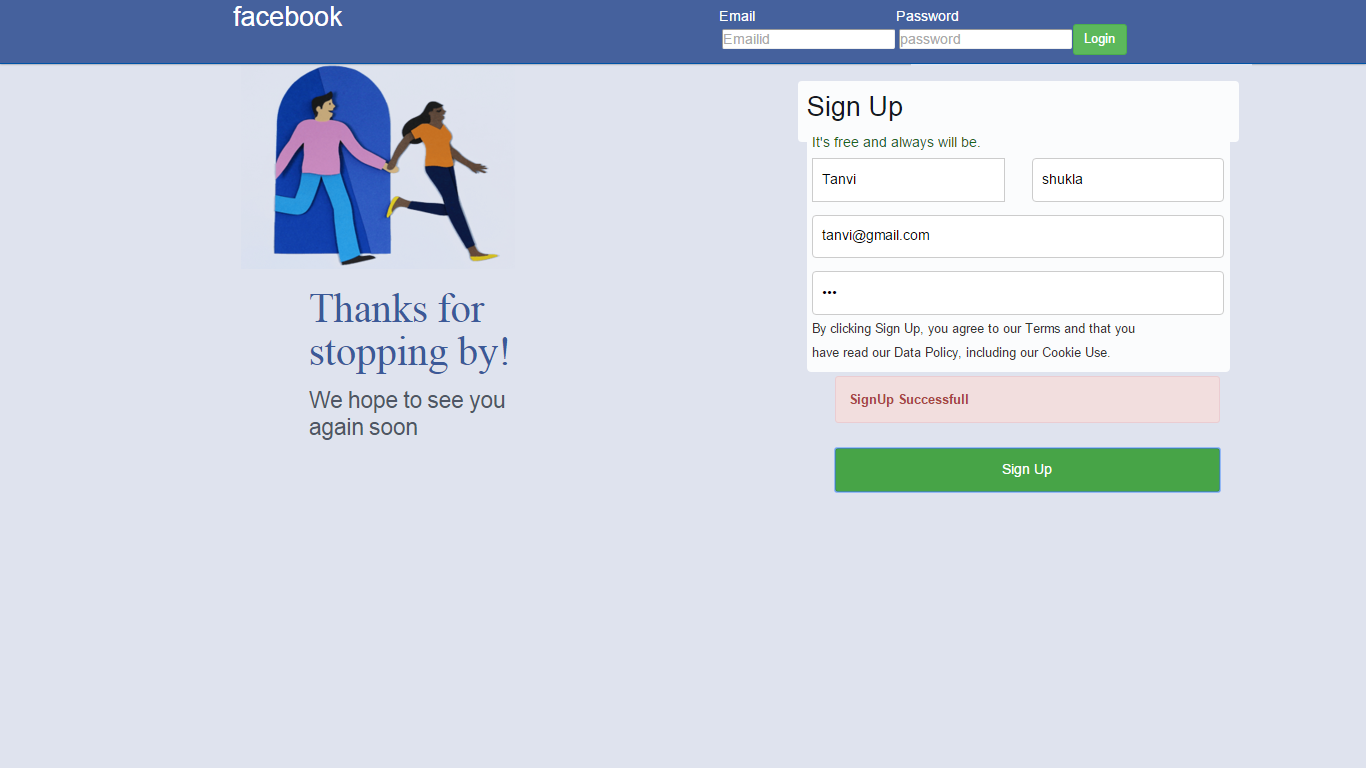


after signIn

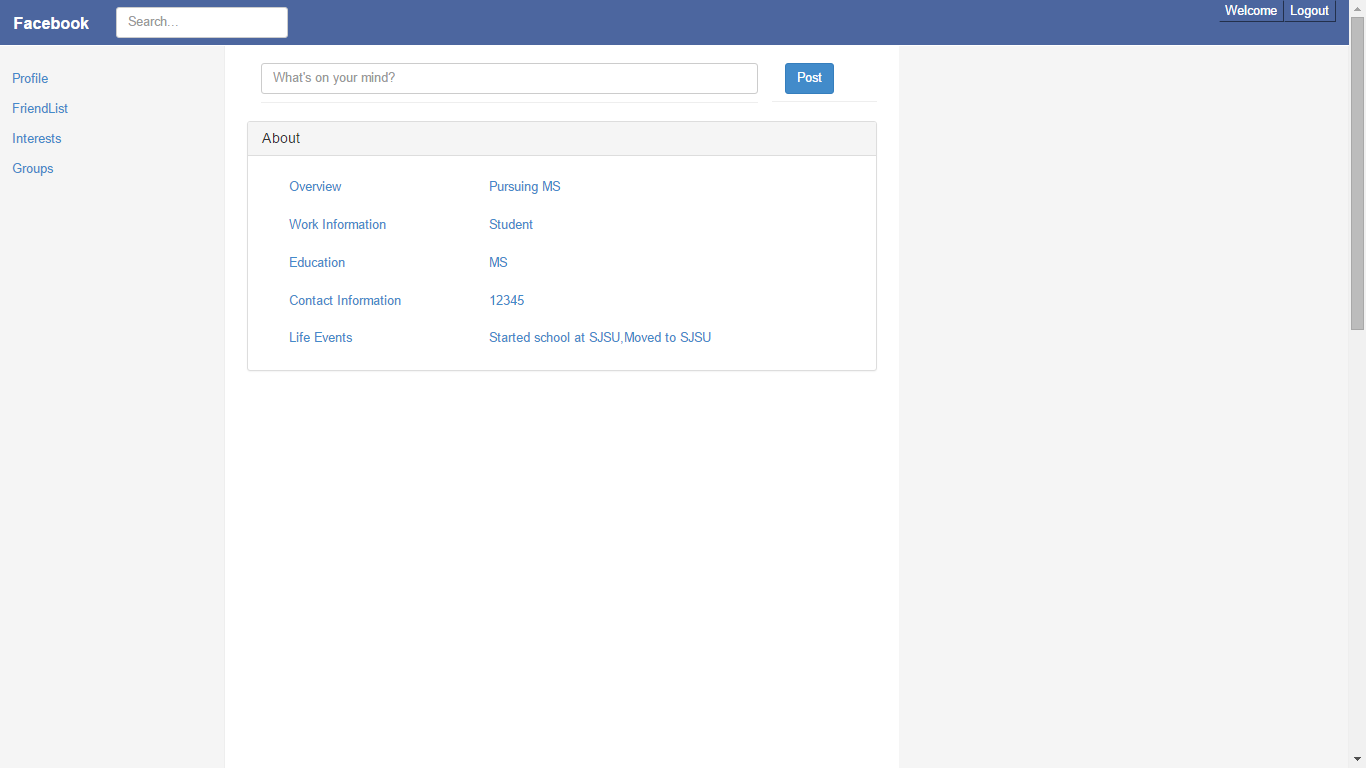


SignUp User

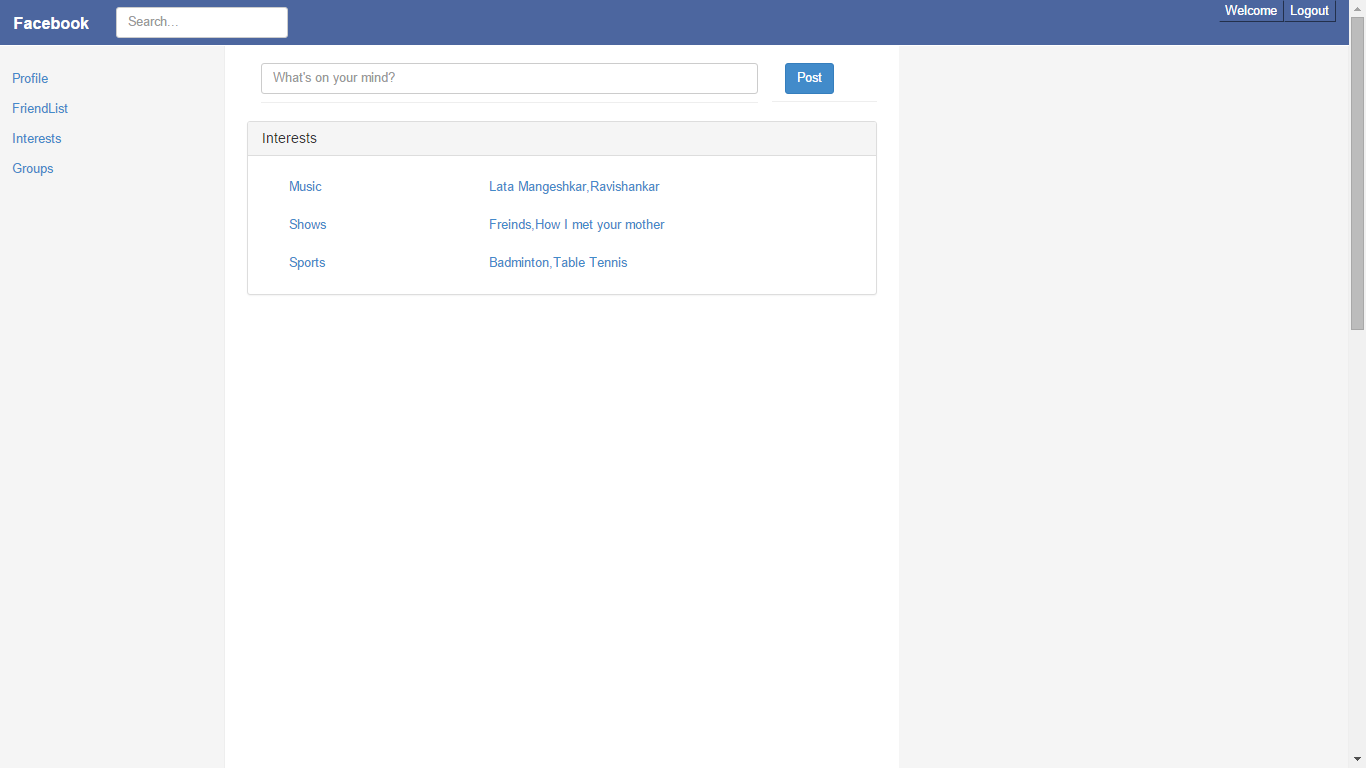




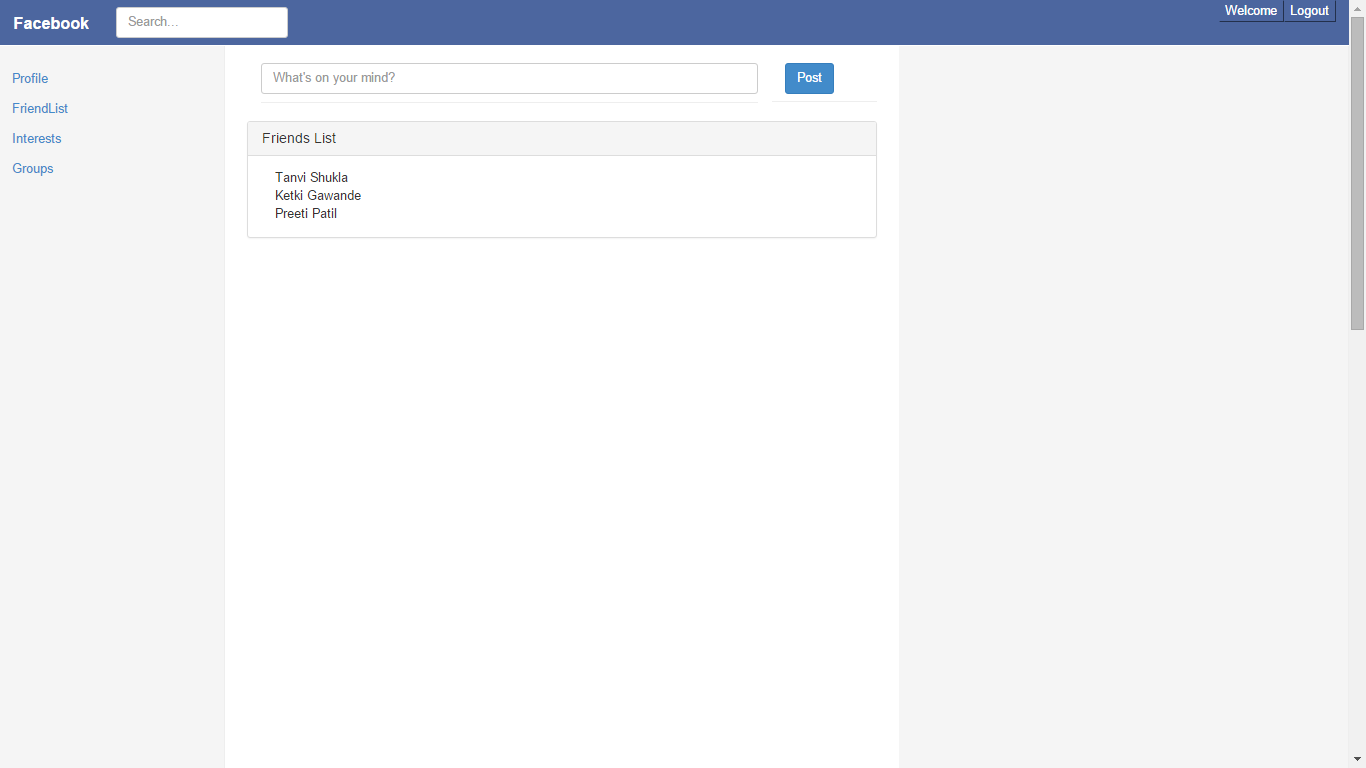
Profile page-About



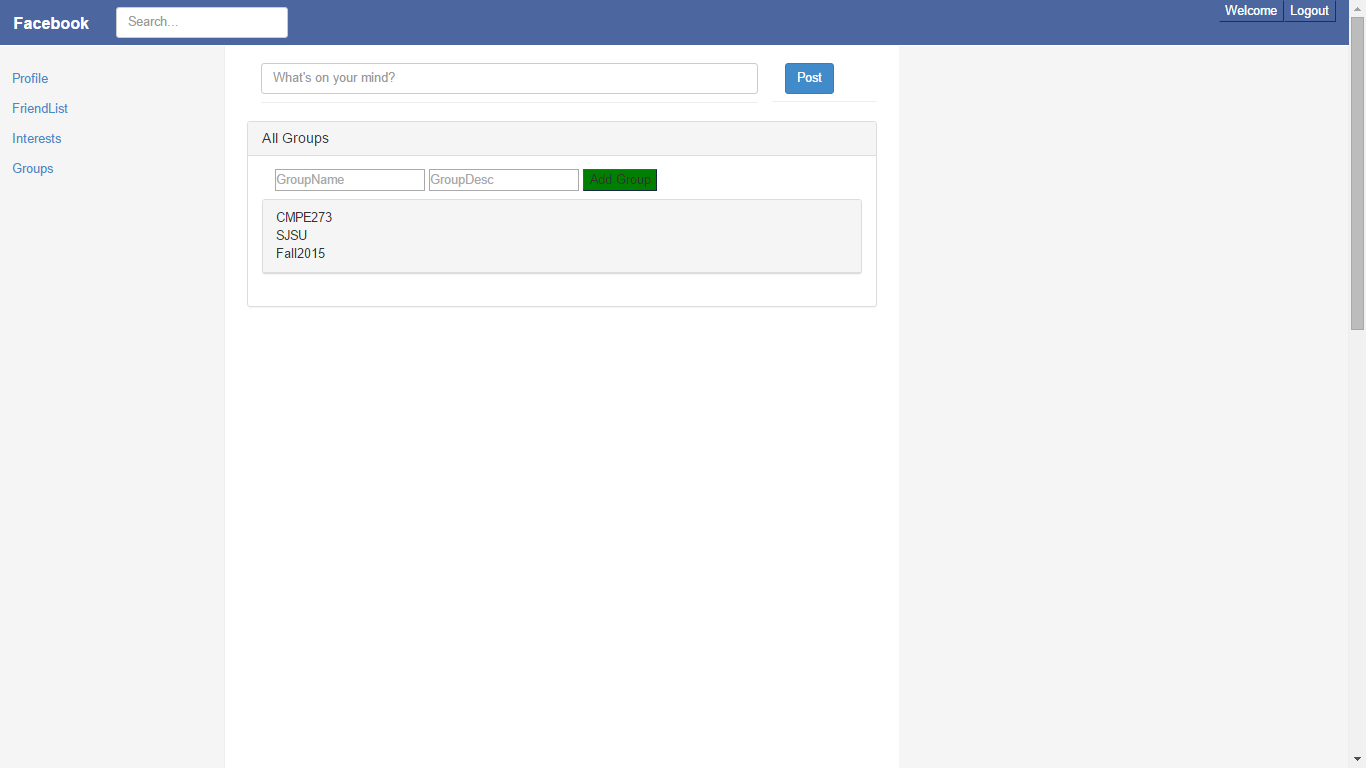
Interests



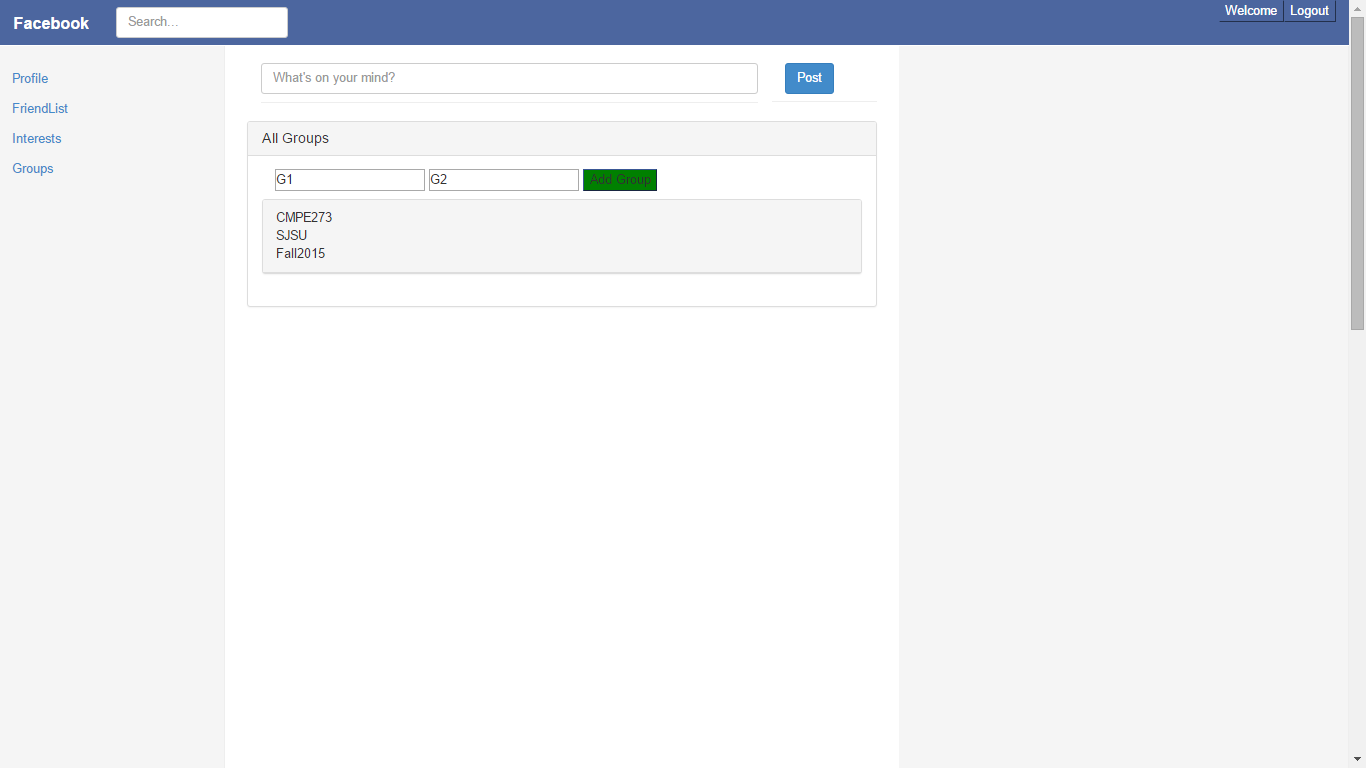
FriendList



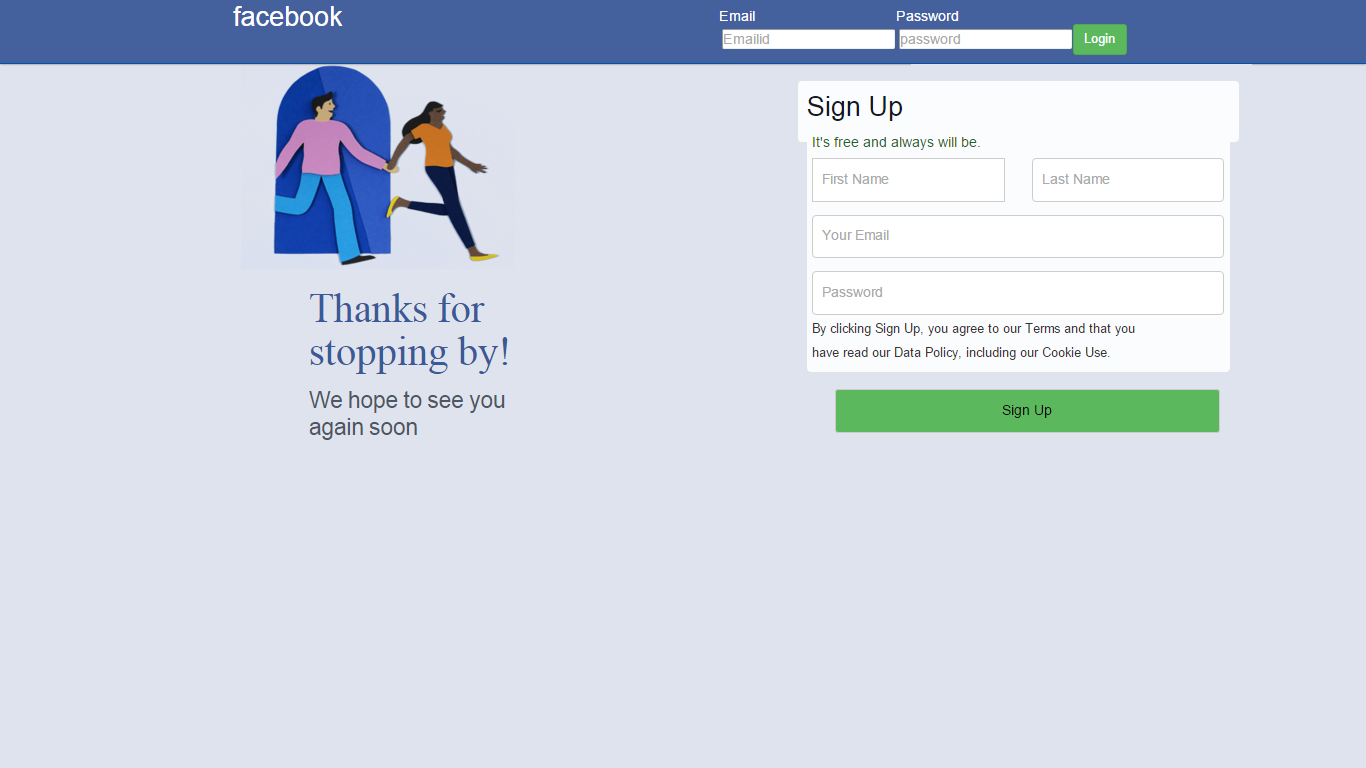
Groups



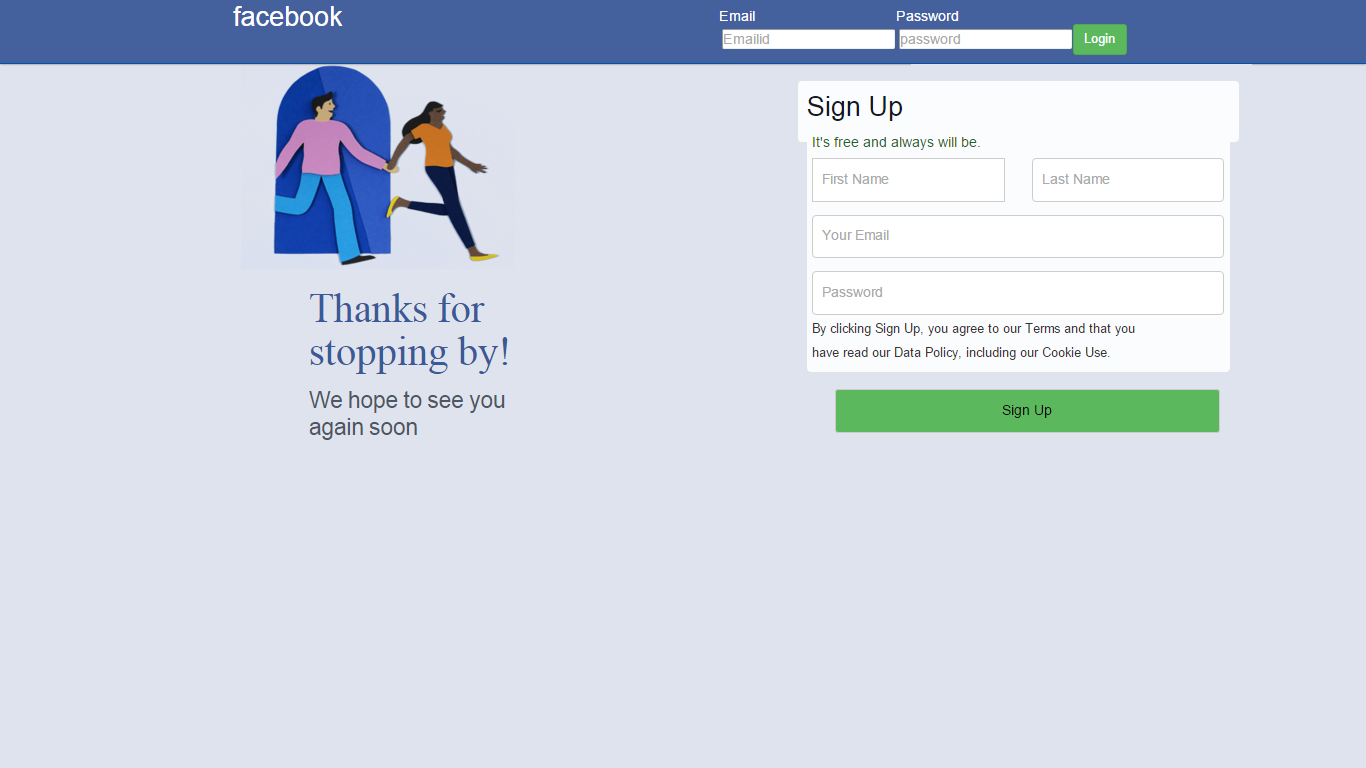
SignOut



After clicking logout

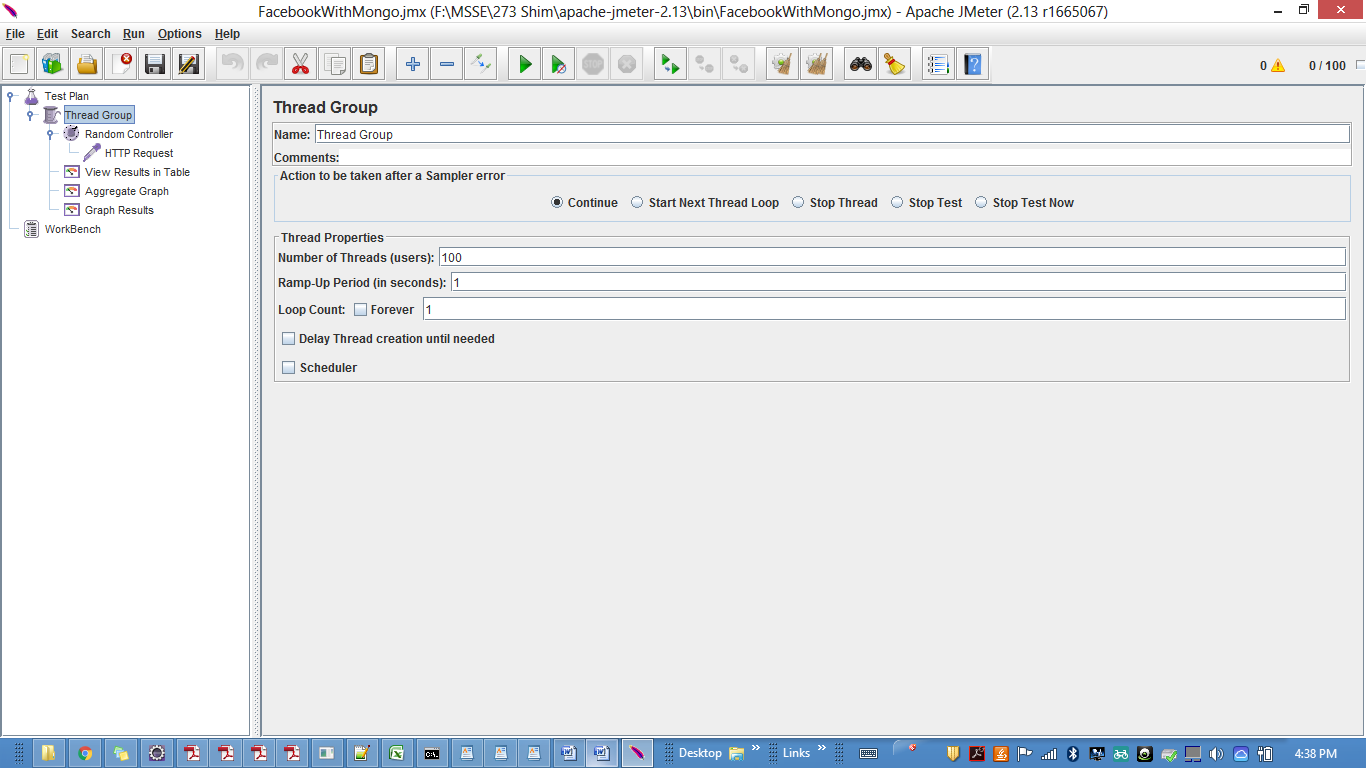


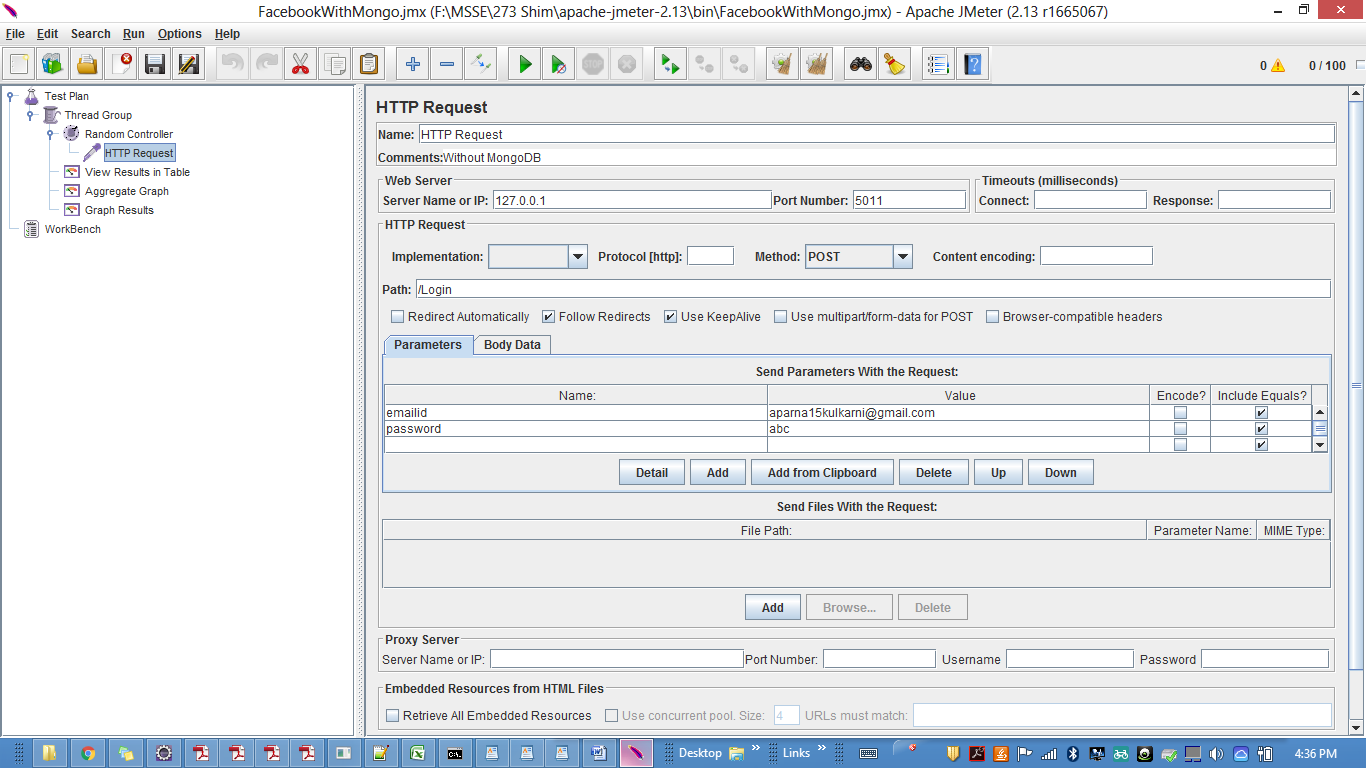
After clicking on back button of The first page of facebook since session is not maintained,it navigates to the same page.

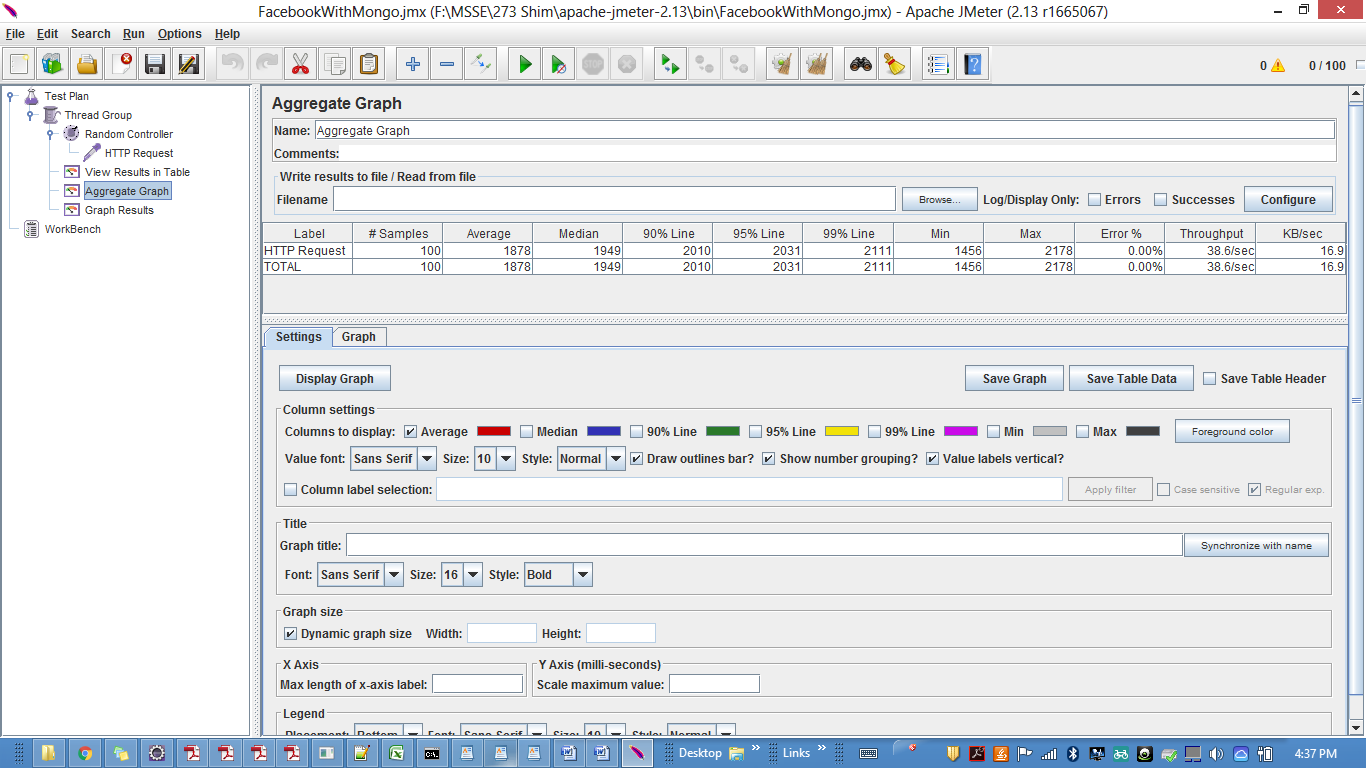


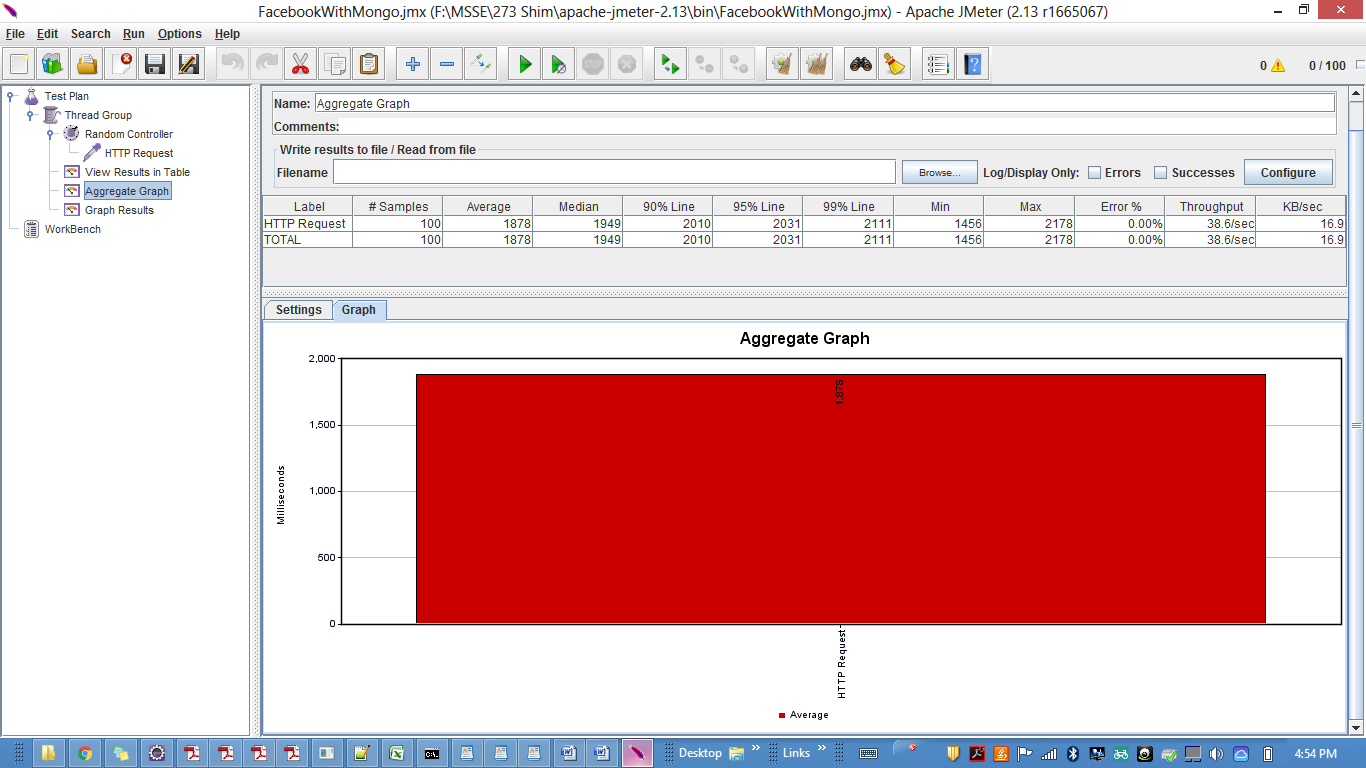
**Performances:**

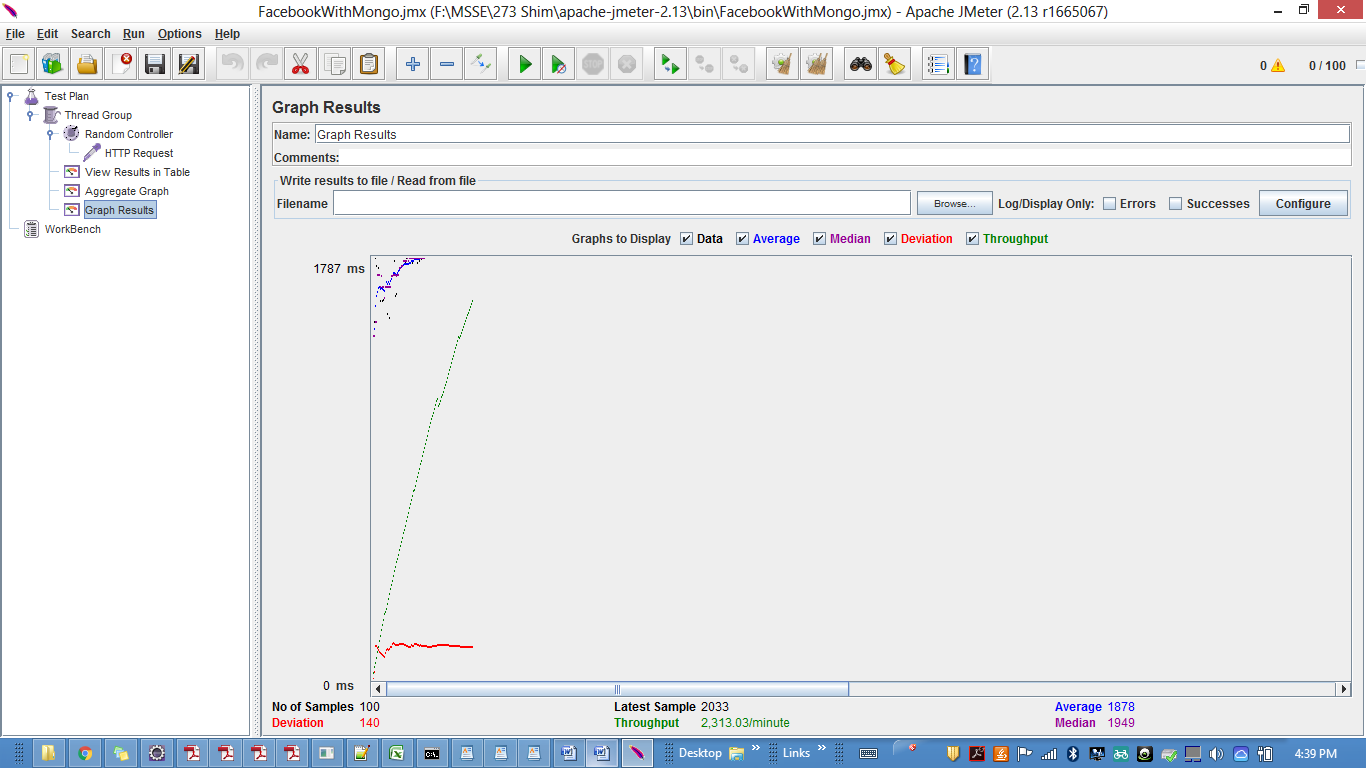
Performance of system:

100 users

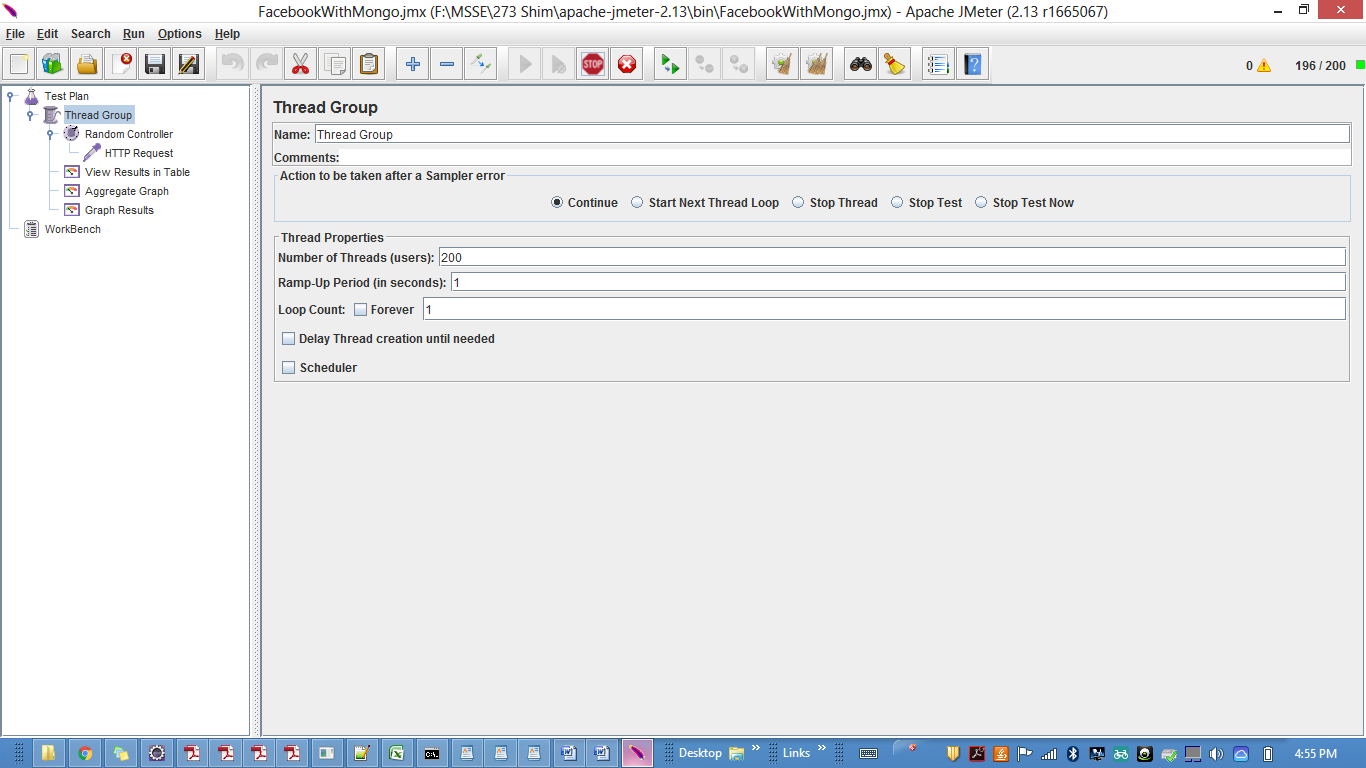


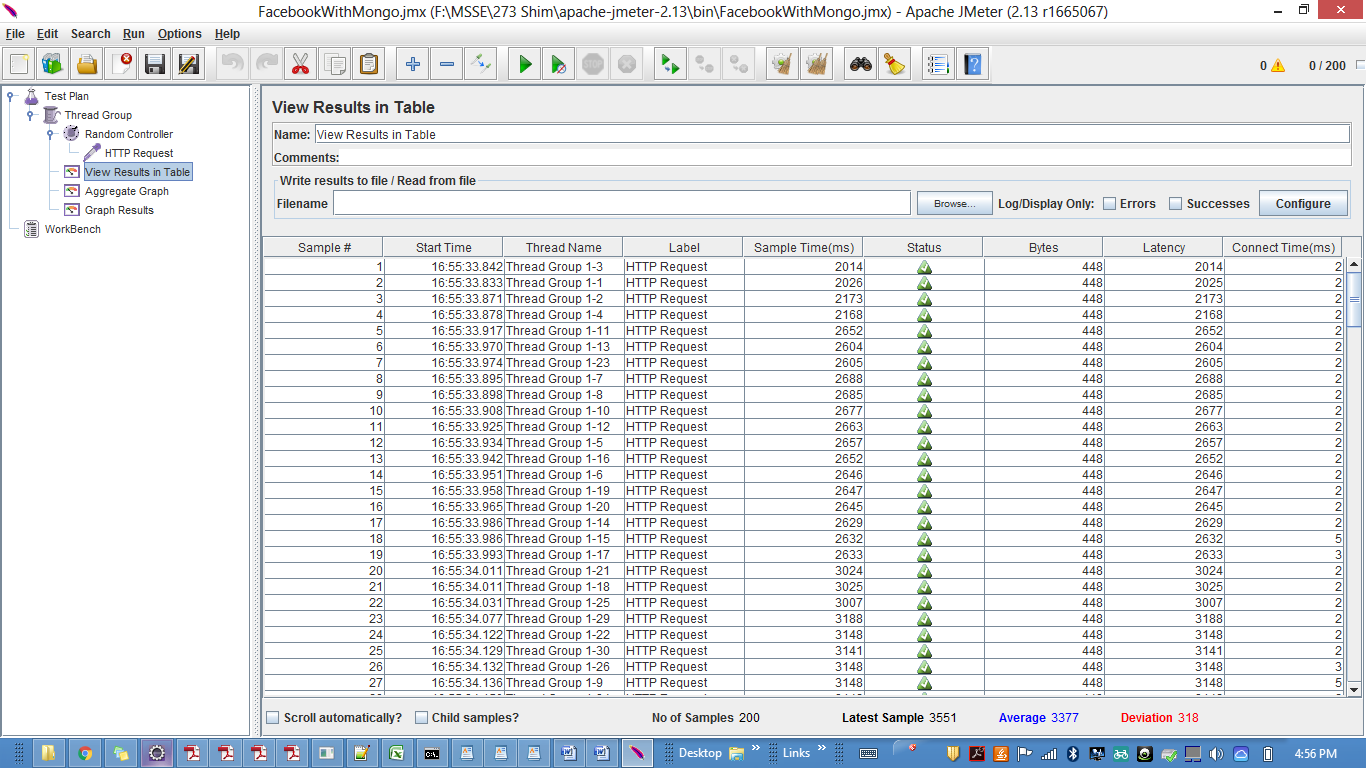


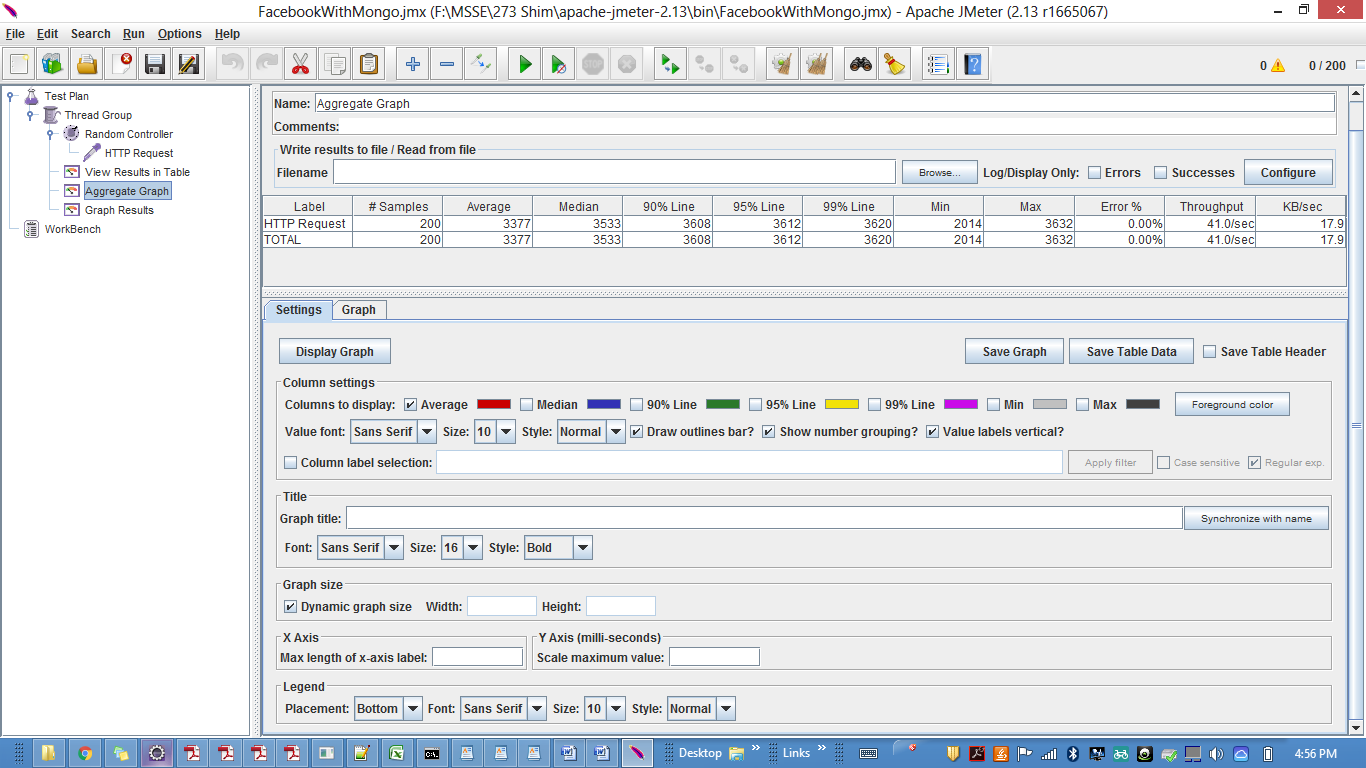


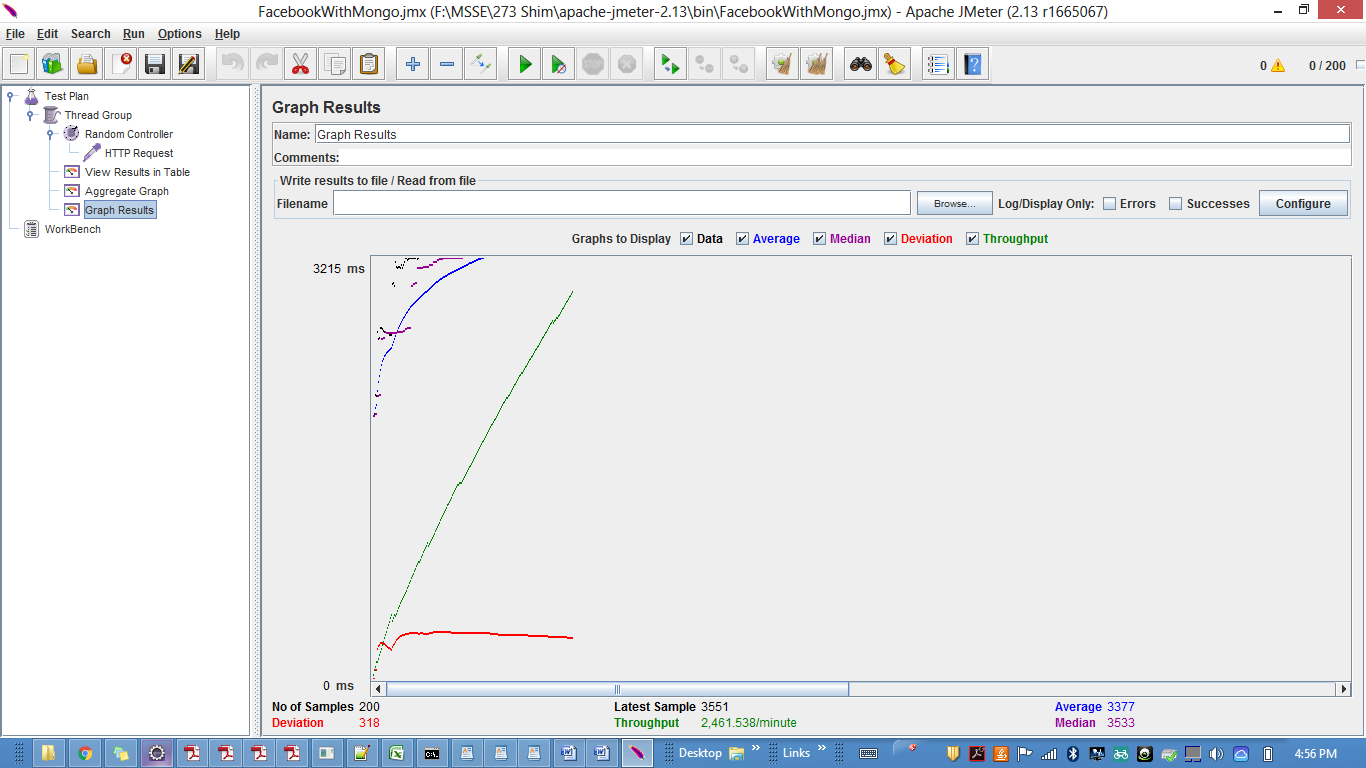


200 users

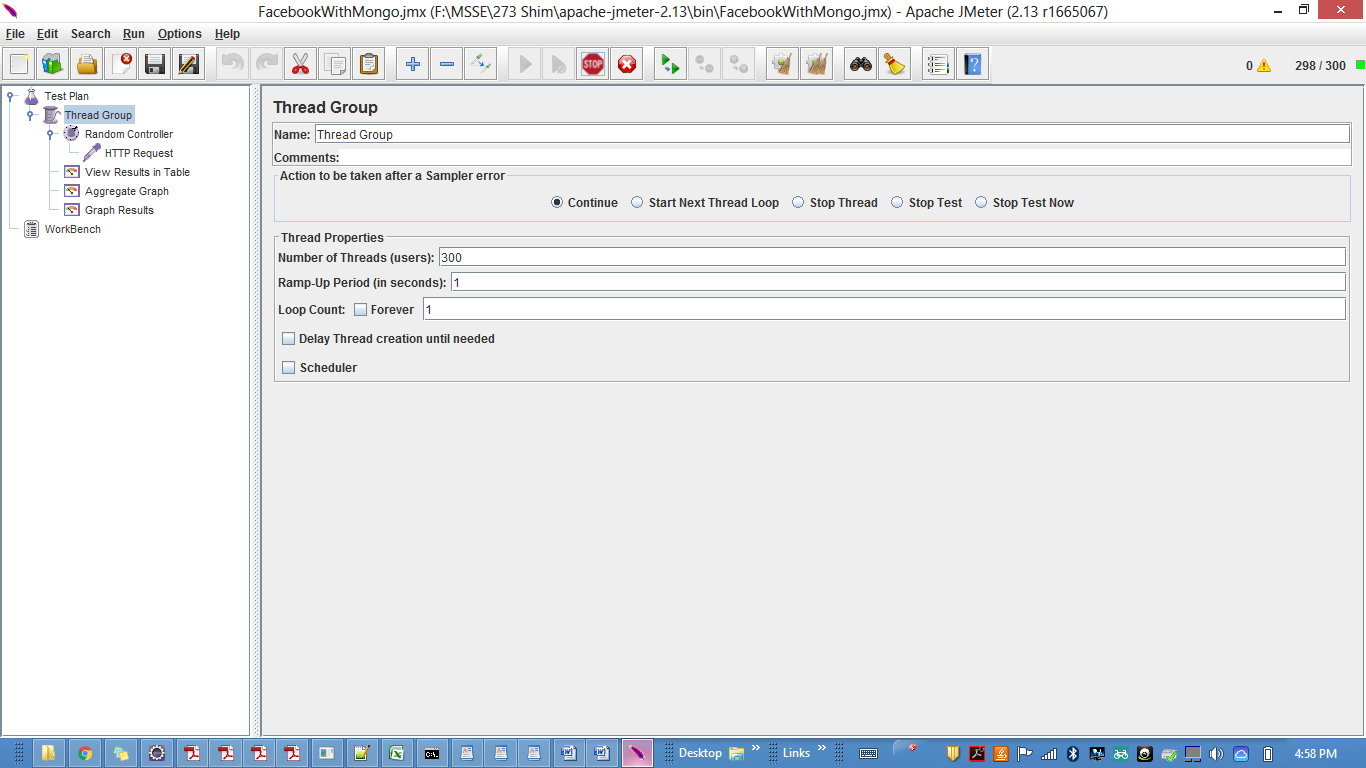


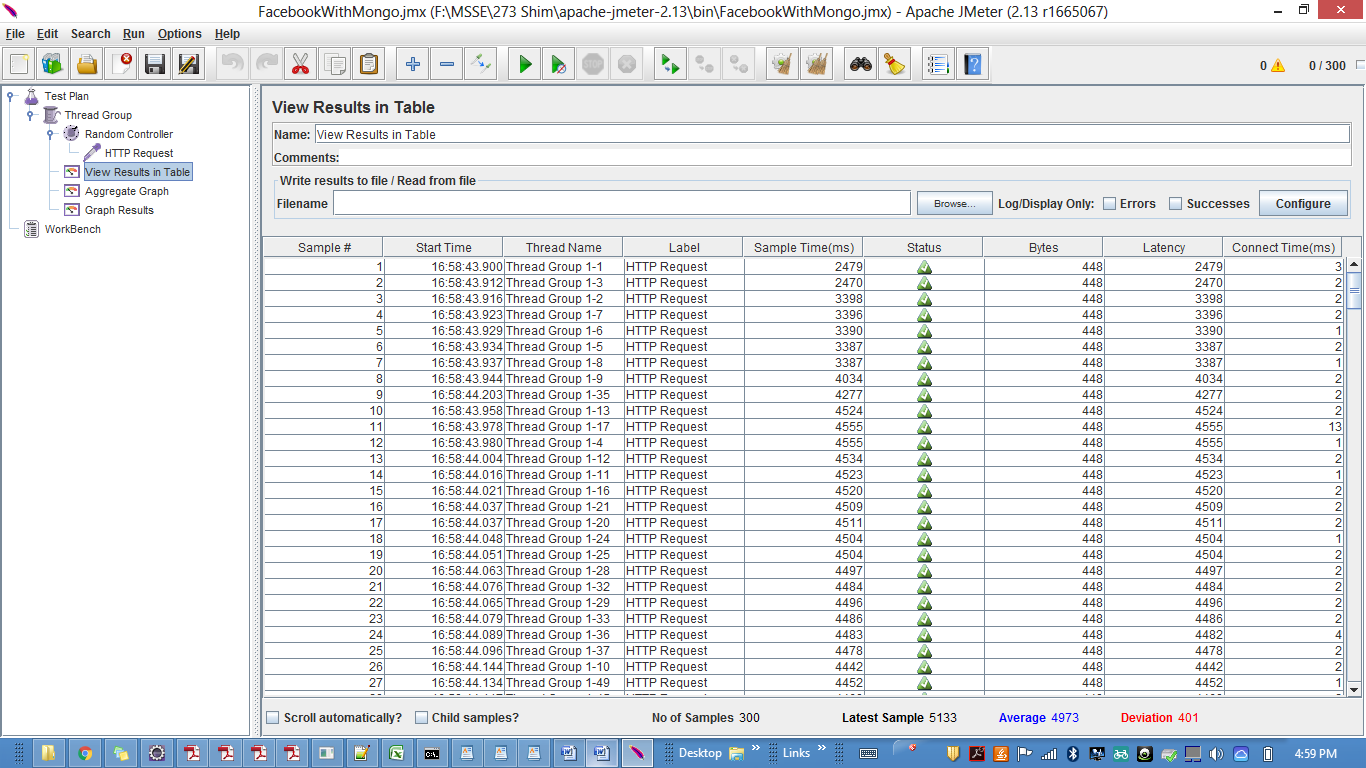


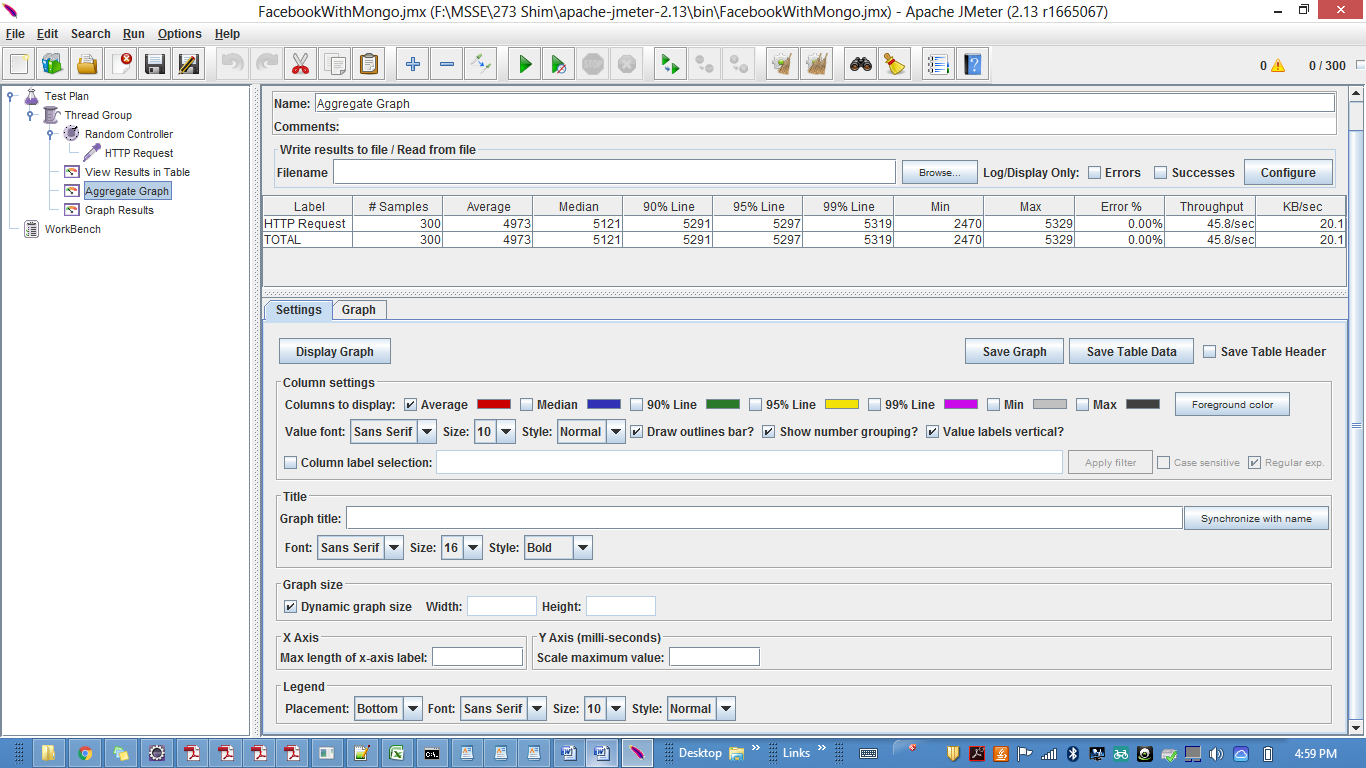


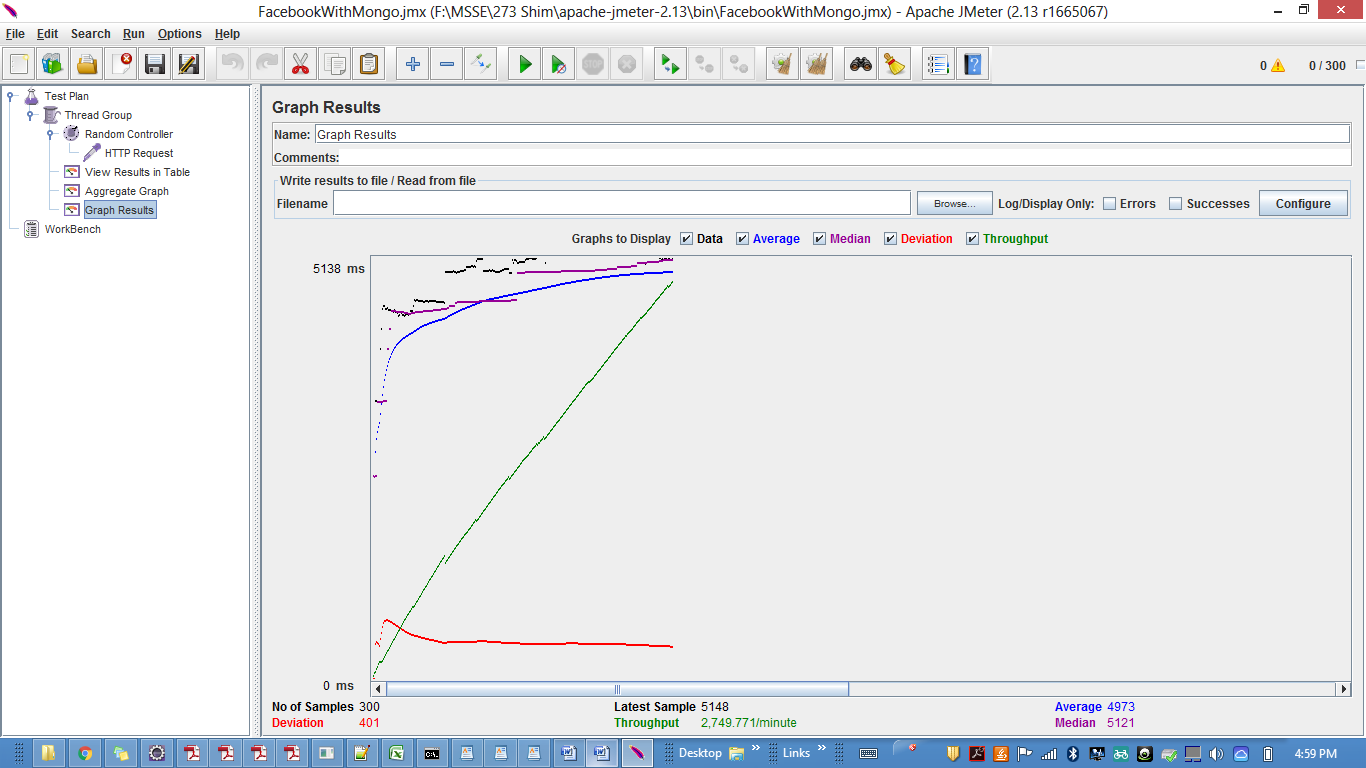


300 users

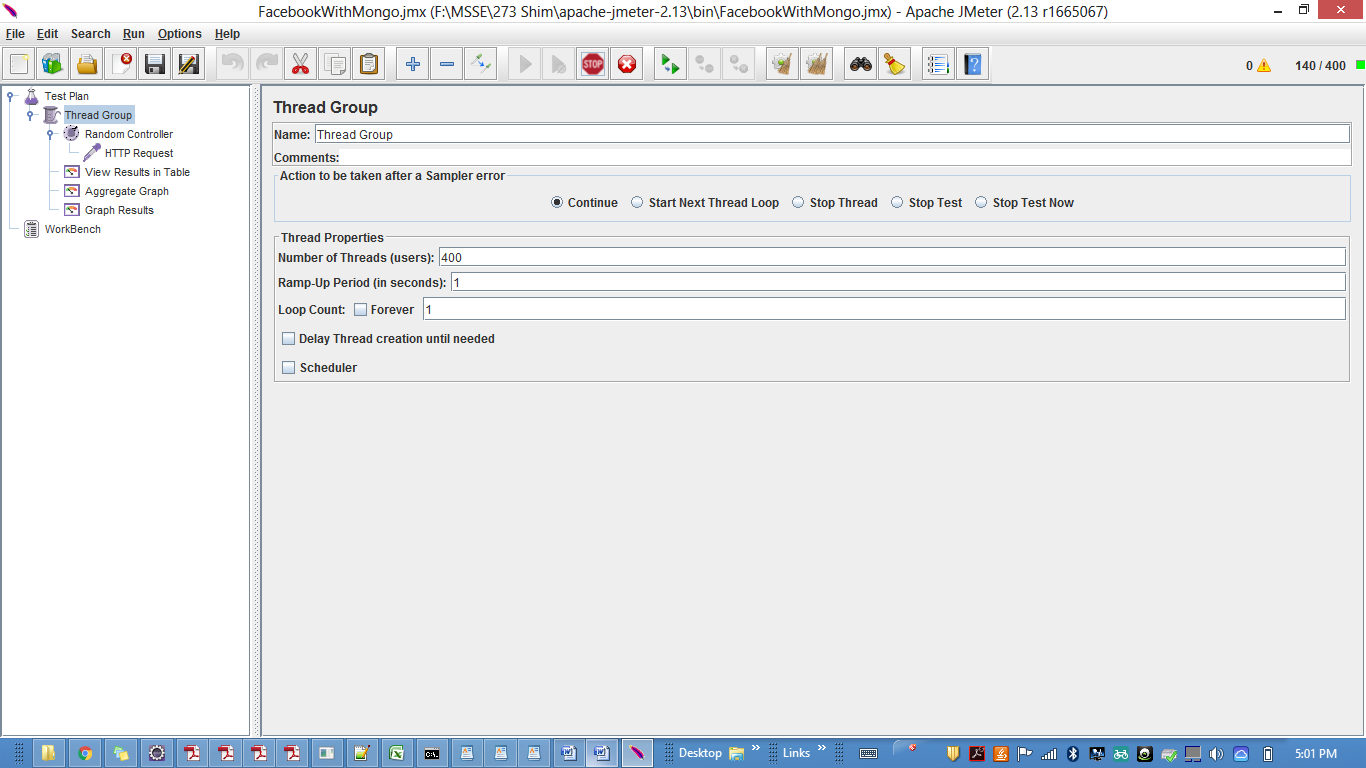


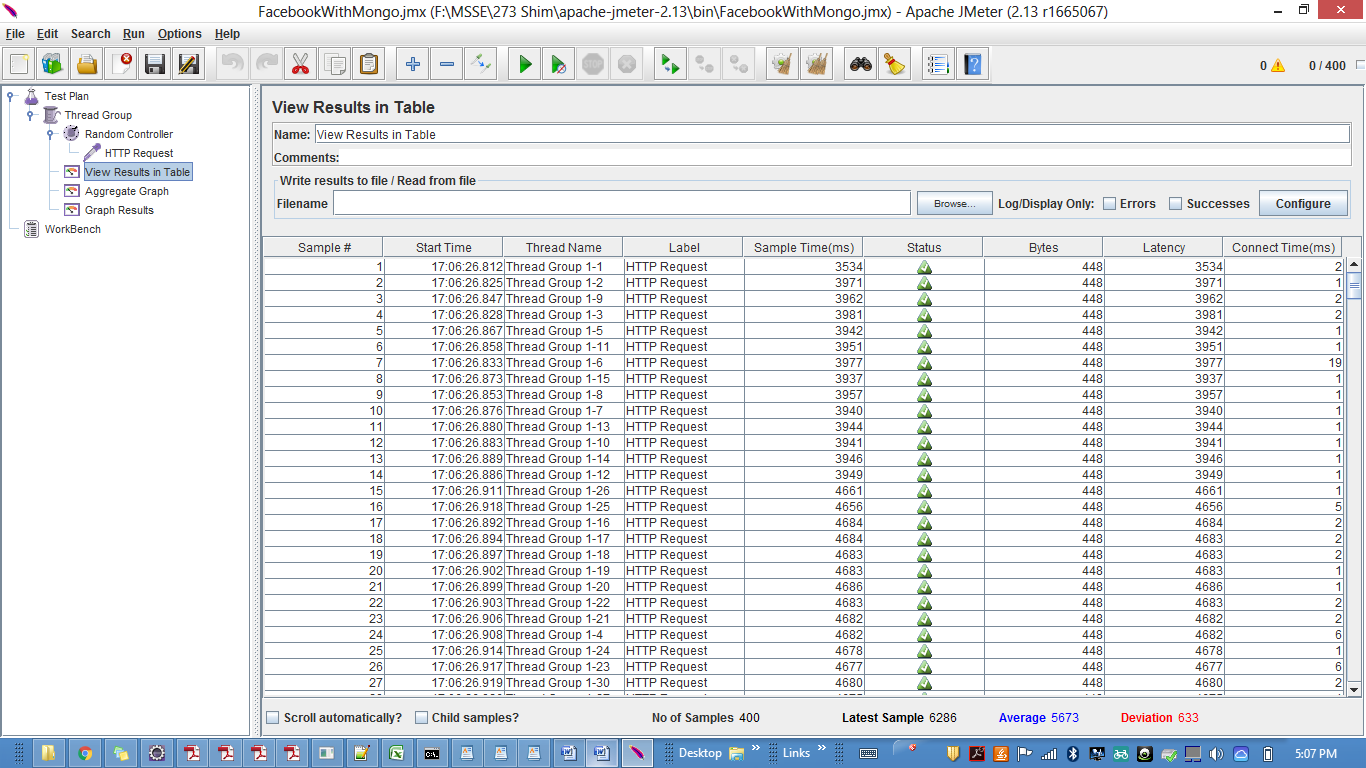


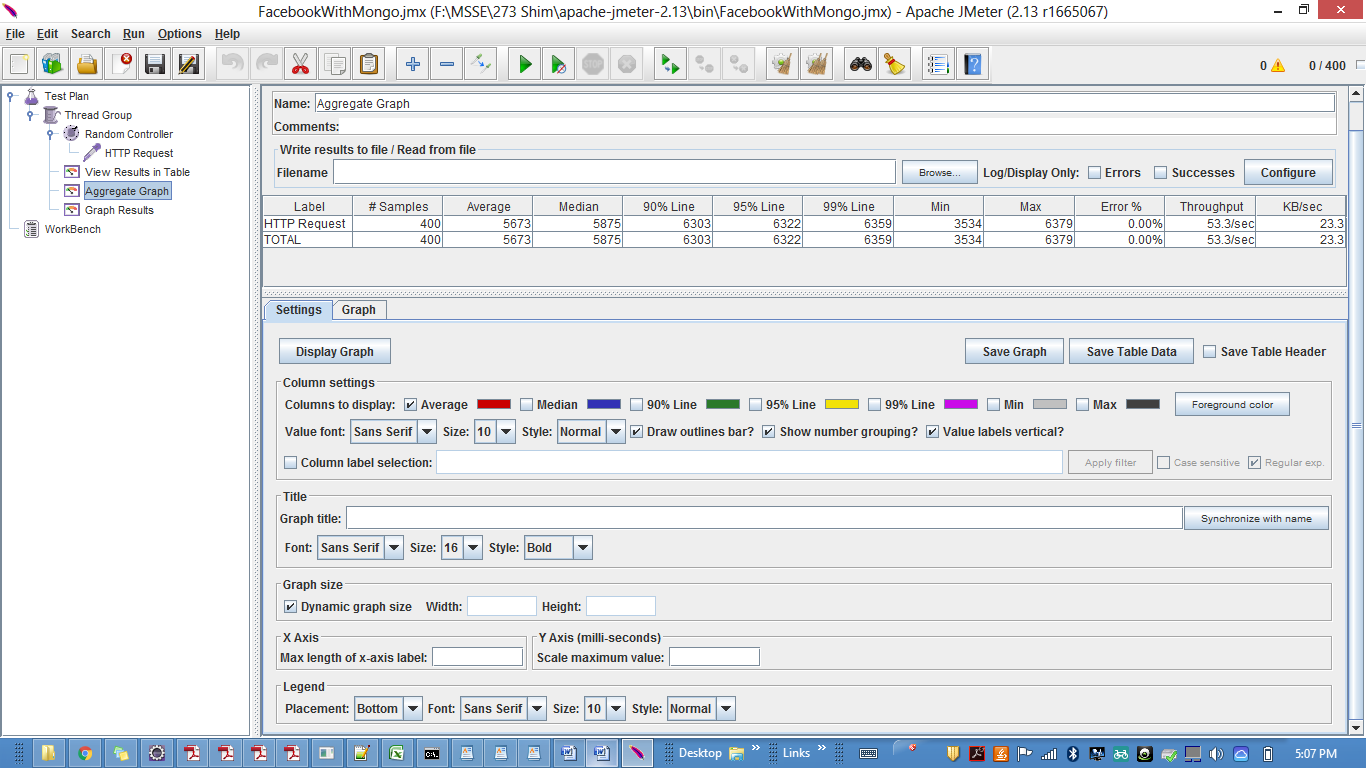


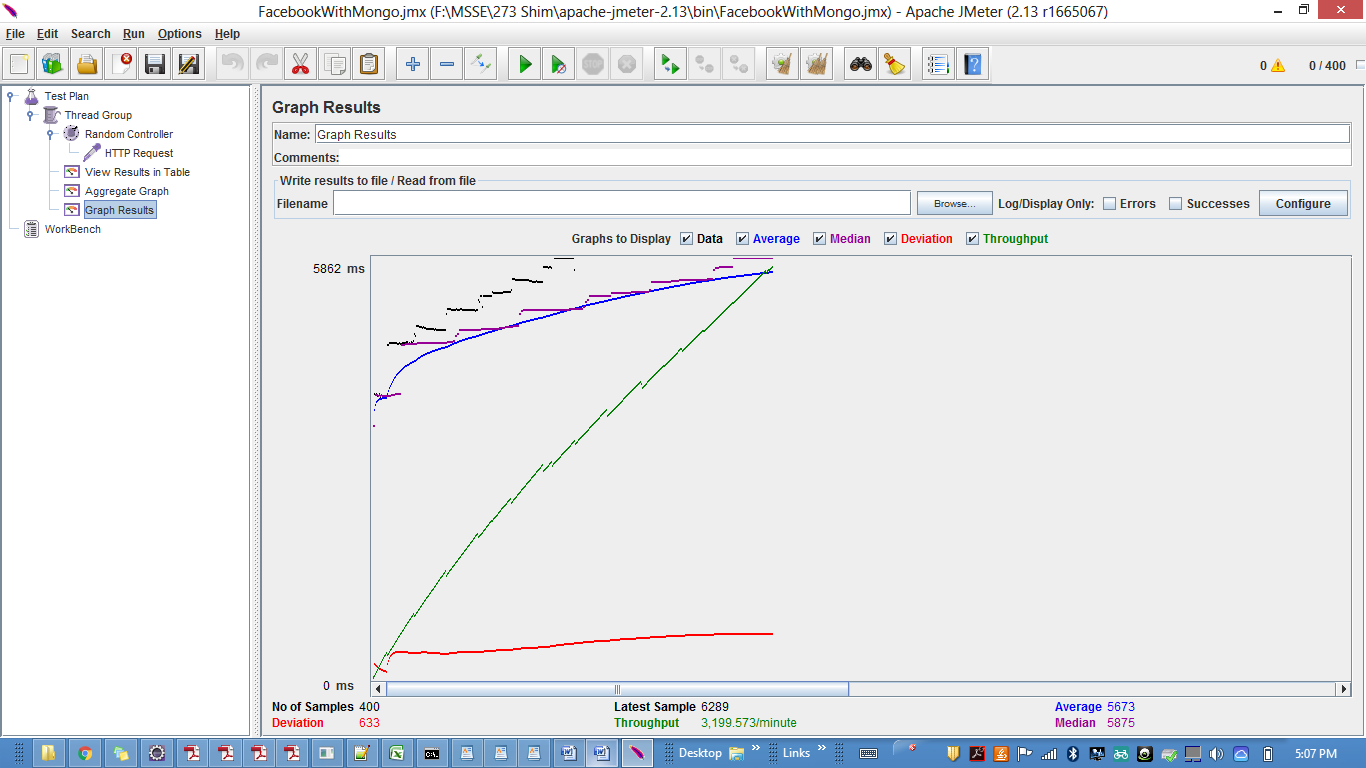


400 users

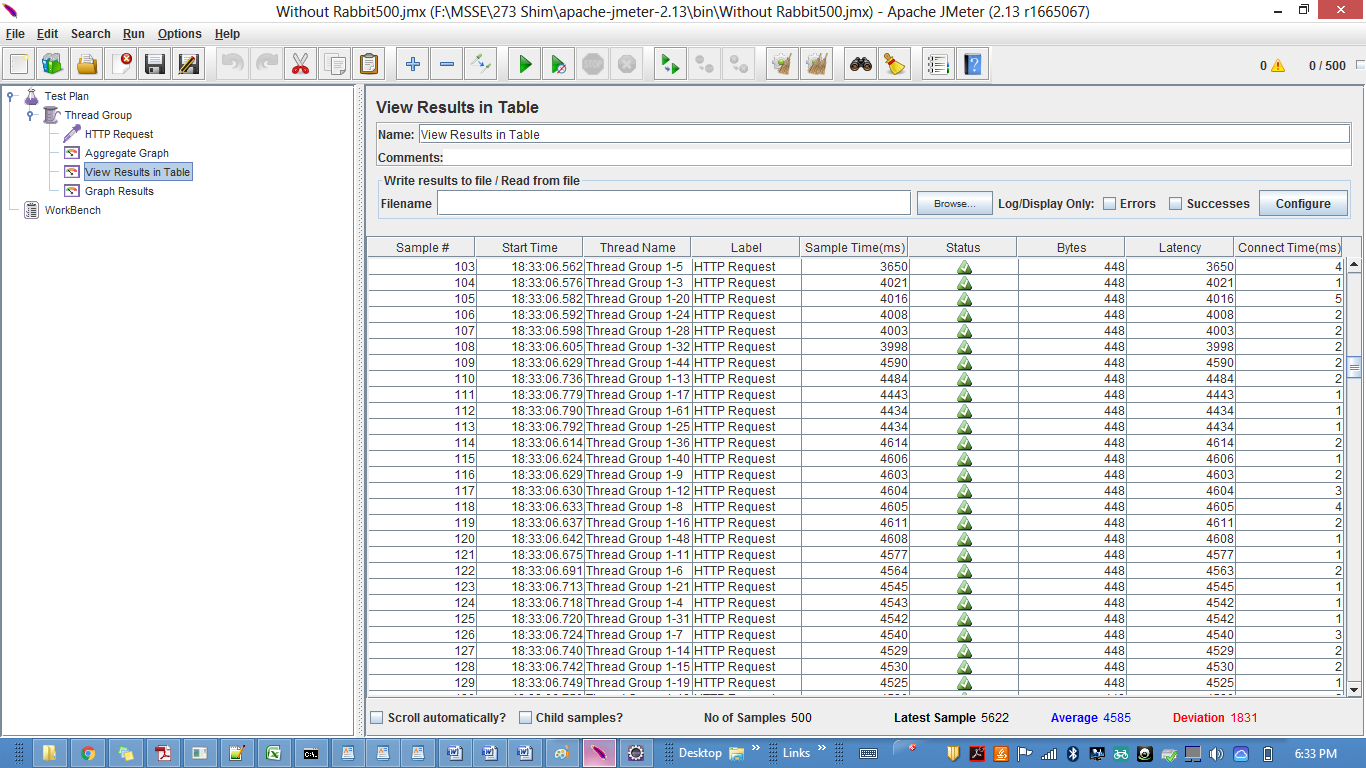


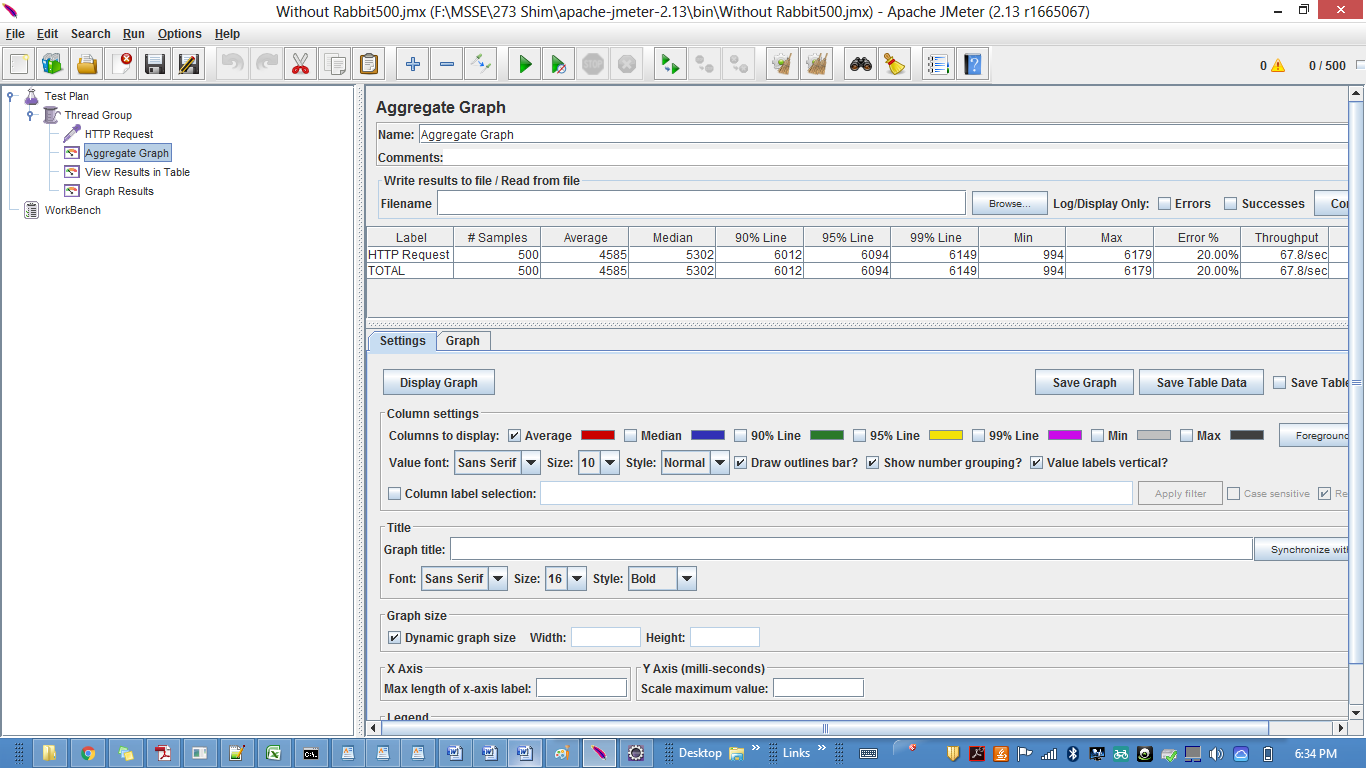


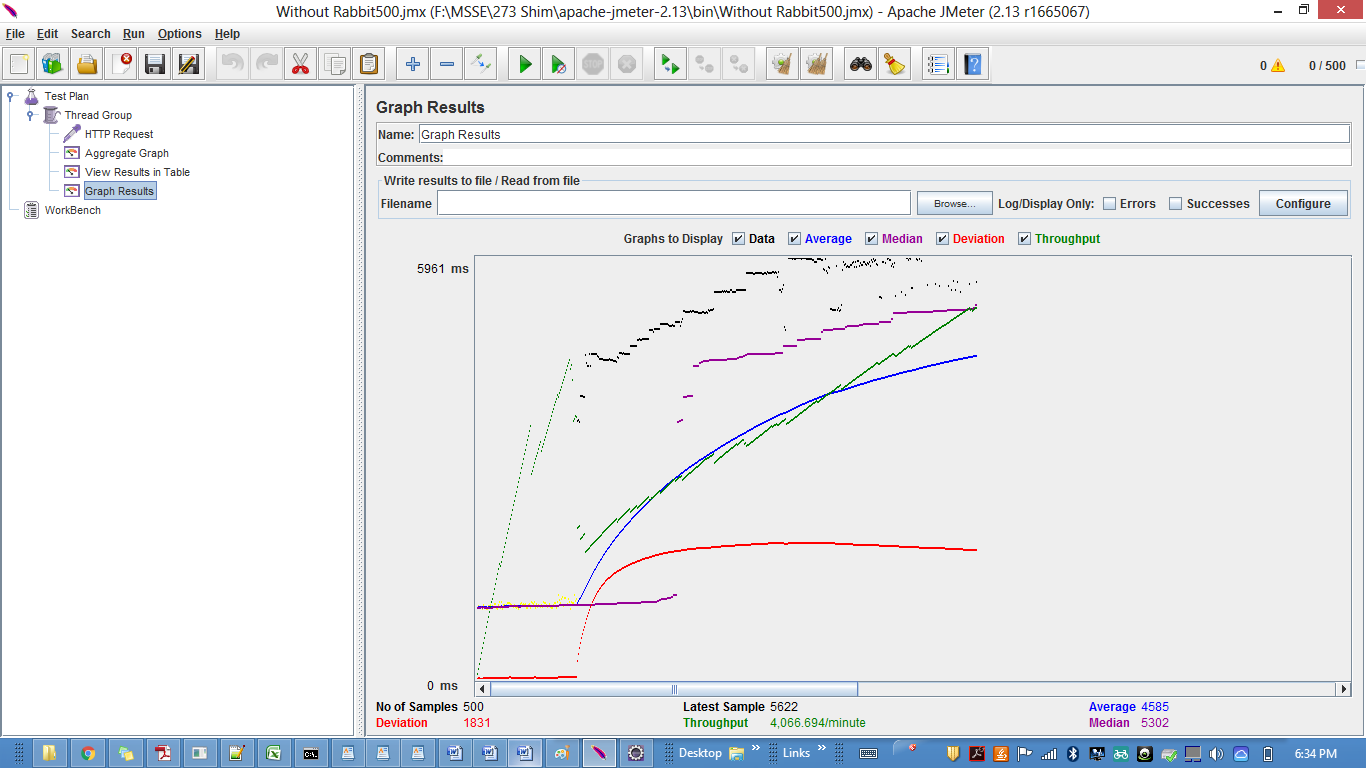




500 users







**Comparison:**

Comparison chart for Throughput of 100,200,300,400,500 concurrent users with RabbitMQ and Without RabbirMQ.

Using RabbitMQ the performance time of the system decreases and throughput increases.This happens because of the queue availability provded by the Rabbitmq.The system can be available to many users at a time hence increasing the throughput of the system.Where as when RabbitMQ was not used,throughput was less since the single node server was not able to process many users at a time.

**Question & Answers:**

**1.** How to achieve horizontal scalability with single RabbitMQ Server? Elaborate on the results of throughput with and without using RabbitMQ. If you find any increase/decrease in the throughput, explain the reason for the same.

**Answer:** RabbitMQ is AMQP messaging broker.It provides clustering and high availability features.

horizontal scalability is mainly only for very large scale applications, with lots of data. As RabbitMQ provides the queing facility system can access multiple user requests.The throughput was increased as compared to the system without RabbitMQ.The reason behind it was that the system gets a lot of user requests at a time and because of availability of RabbitMQ queues the server can access those requests fast and hence increase in the throughput is obtained.

**2.** Explain the strategy used in implementing Sessions in this Lab. Compare your Sessions strategy with default express Sessions. Describe which Strategy is better.

**Answer:** In this lab I implemented the session using MongoDB.It stores session in a default"sessions collection" And after every logout function call the session is destroyed.Here even after server crashes the user will be stored in the session as a database(MongoDB) and hence request sent from client can be accessed by another server.

Client sessions are meant for debugging and development and does not store the session on the database.Also node session does not scale past a single process.Hence overall Mongo Sessions are better to use so that even aftre server failure it can be recognized using the database.

**3.** Describe the type of encryption algorithm used to store passwords in this lab and the reason to use it.

**Answer:** I have used bcrypt encryption algorithm.The password is encrypted with Bcrypt.genSalt function.Sync It is a key derivation function for passwords. bcrypt is an adaptive function: over time, the iteration count can be increased to make it slower, so it remains resistant to [brute-force search](https://en.wikipedia.org/wiki/Brute-force_search) attacks even with increasing computation power. The bcrypt function is the default password hash algorithm many systems including some [Linux distributions](https://en.wikipedia.org/wiki/Linux_distribution) such as [SUSE Linux](https://en.wikipedia.org/wiki/SUSE_Linux). The bcrypt algorithm depends heavily on its "Eksblowfish" key setup algorithm.It is not a symmetric encryption algorithm.