**LAB3**

**Introduction:**

**Goal**: To learn Web services using JAX-WS (Java API for XML Web Services) and Apache Axis and express framework imlementation of Calculator and Facebook is done in this lab.Calculator system performs arithmatic opeations addition,subtraction,multiplcation,division.Facebook application uses MySql to store the database and nodejs angularjs are used for front end implementation.Web services acts as a server.

**Purpose of system:**

**Calculator:**

1. Developing a calculator in order to perform basic operations like Addition, Subtraction, Multiplication and Division.
2. Taking care of different mathematical errors , exceptions that have a possibility of occurrence.
3. The calculator should give proper messages for answer due to every operation and error message for every mathematical error that can occur.

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

**1**.Calculate:Helps to perform all the arithmatic operations

**Facebook:**

a)The system should perform following Basic Users functionalities:

* System should allow Sign up for new user (at least first name, last name, Email and password)
* Sign in existing user.
* Sign out the logged in user.
* Sign Up should have first name, last name, Email and password.

b) In order to use the system, a user must sign in first to the system

c) Users account should provide basic details such as:

* About (information about the user): User overview, Work and education, contact info and life events.
* Friends list (Display user's friend list): Send friend request, accept friend request.
* Displaying user's interests like music, shows and sports.

c) Provide news feed functionality.

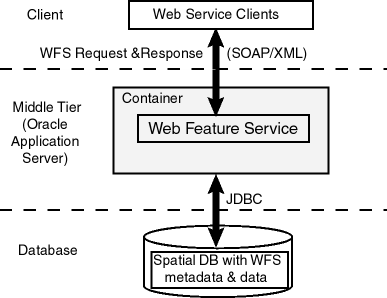
**List of API's for Calculator:**

* 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
* ProfilePageNavigate:Displays the user profile page
* profile: Navigates to profile page
* FriendList:Displays List of Friends
* Friendspage: Navigates to friends page.
* Interests:Displays Interests
* Interestspage :Navigates to interests page.

**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.

Here clients i.e express framework communicates with web services to perform the functions.Web services in turn get data or update data in database(mySql) using JDBC connectivity.



**Calculator:**

When user hits the url and find 2 textbox to enter the number and one dropdown list for selecting the operation to be performed between addition,subtraction,multiplication,division

* User have to give 2 inputs
* User should select one of 4 operations(Addition is set bydefault)
* As soon as the user hits the calculate button the result will be visible.

**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
* After Logout button click user session will be terminated and user will be logged out.

JS files on server:

facebookLogin.java : for validating login credentials on server,

facebookAbout.java: for fetching profile data of respective user from database ,

facebookFriend.java: for fetching user's friendlist from database,

facebookInterest.java: for fetching interests data of respective user from database ,

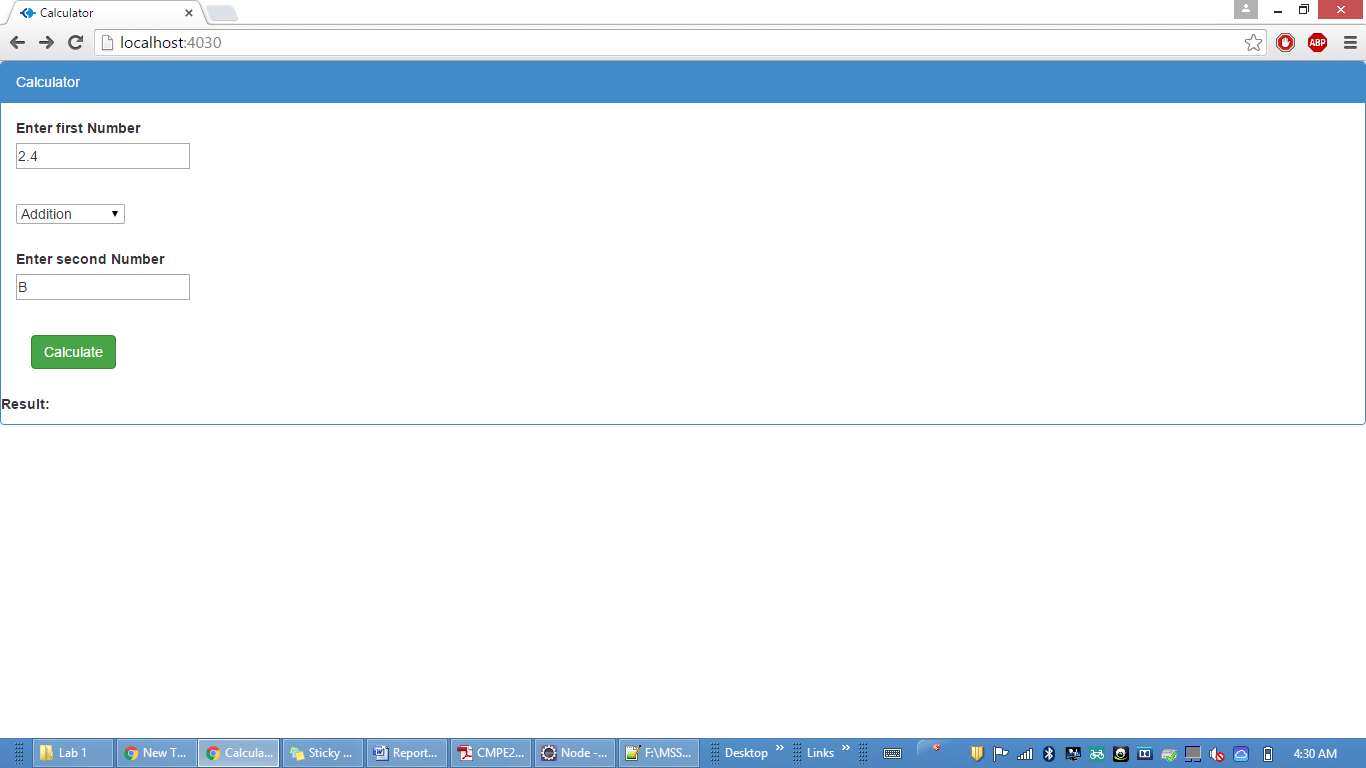
facebookFriend.java: for fetching pending friend requests for respective user from database and then accepting or deleting the requests ,

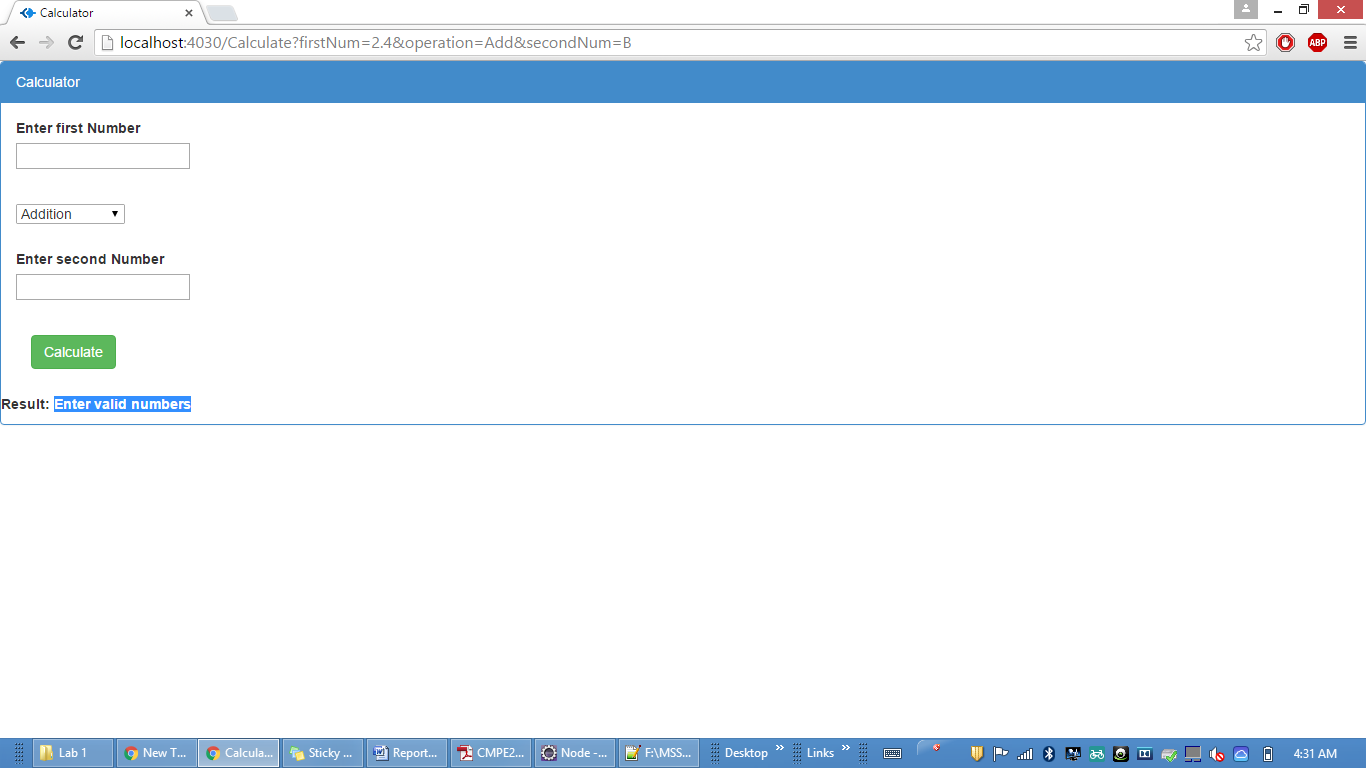
facebookServer.java : for inserting new user data into the database .

**Results:**

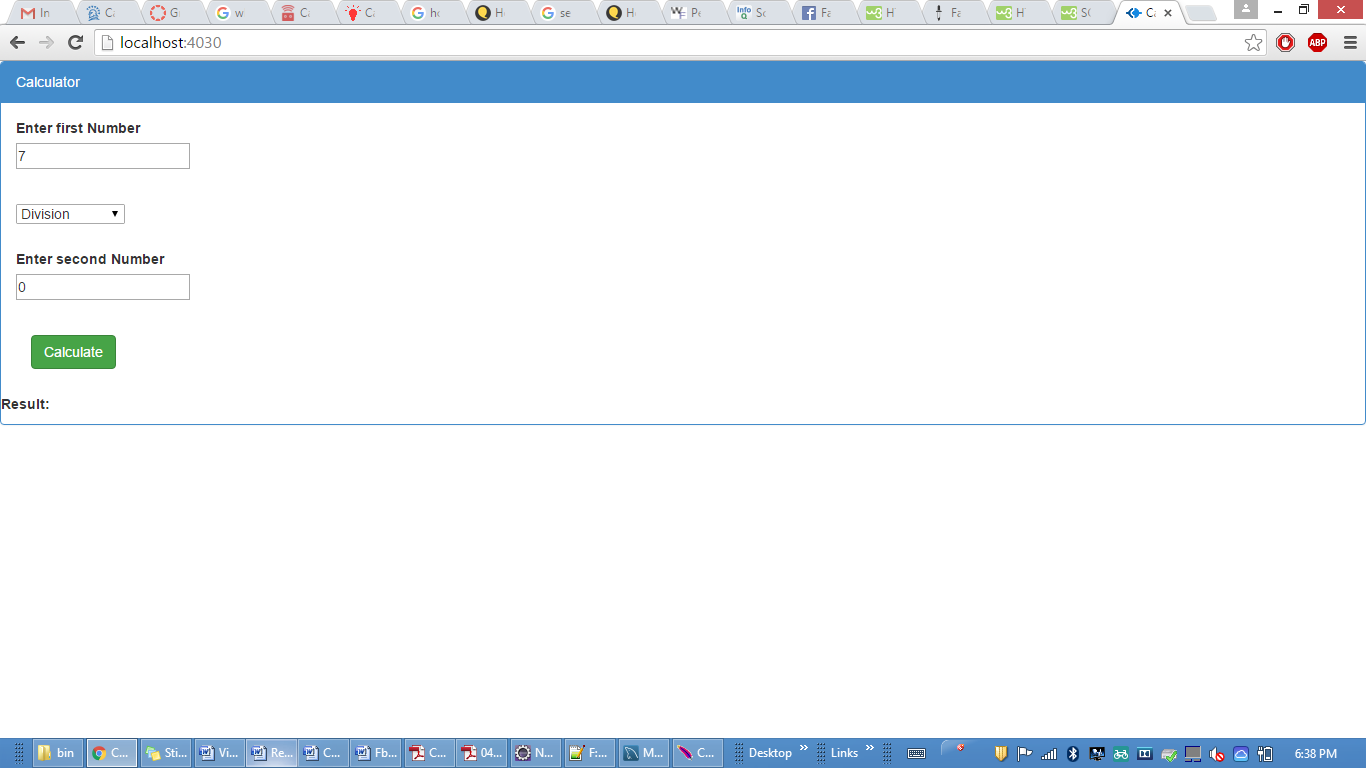
Result for wrong input(input 2:B)

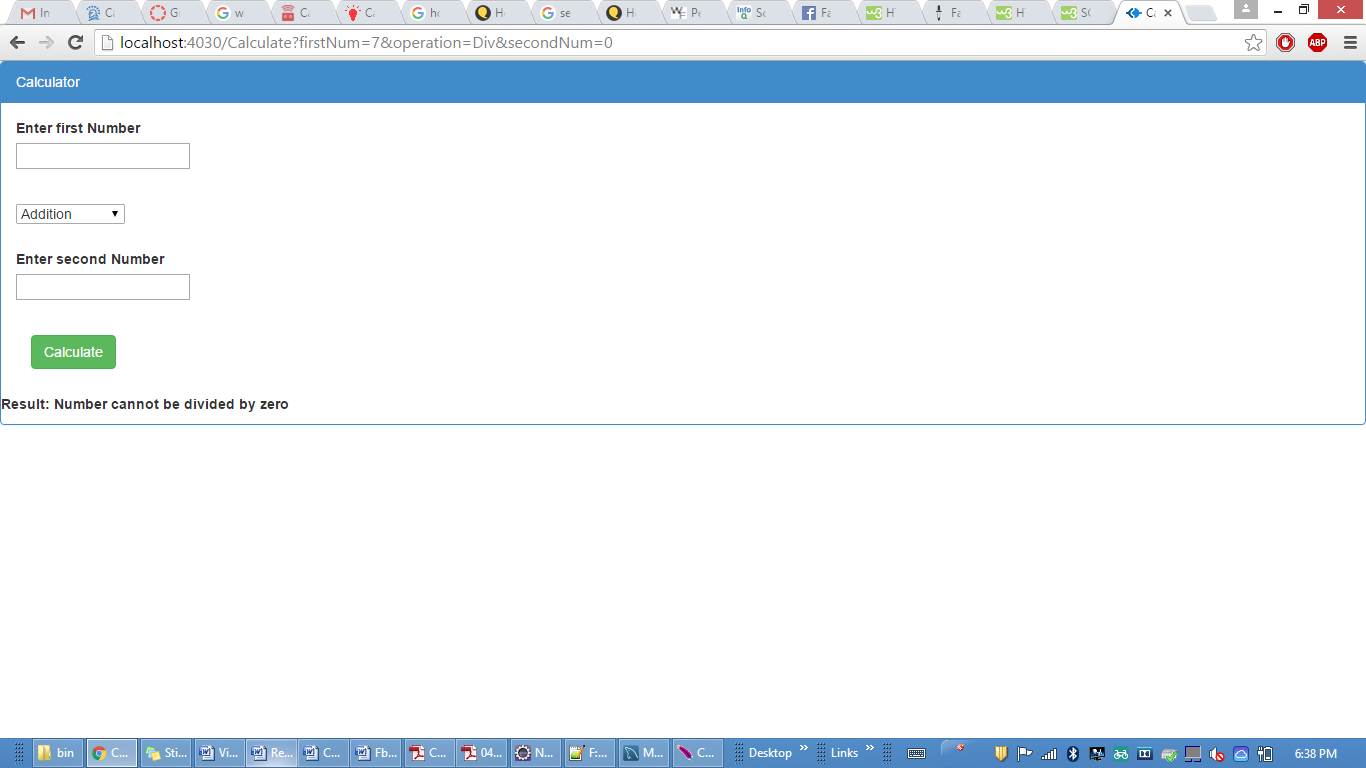
**Calulator:**





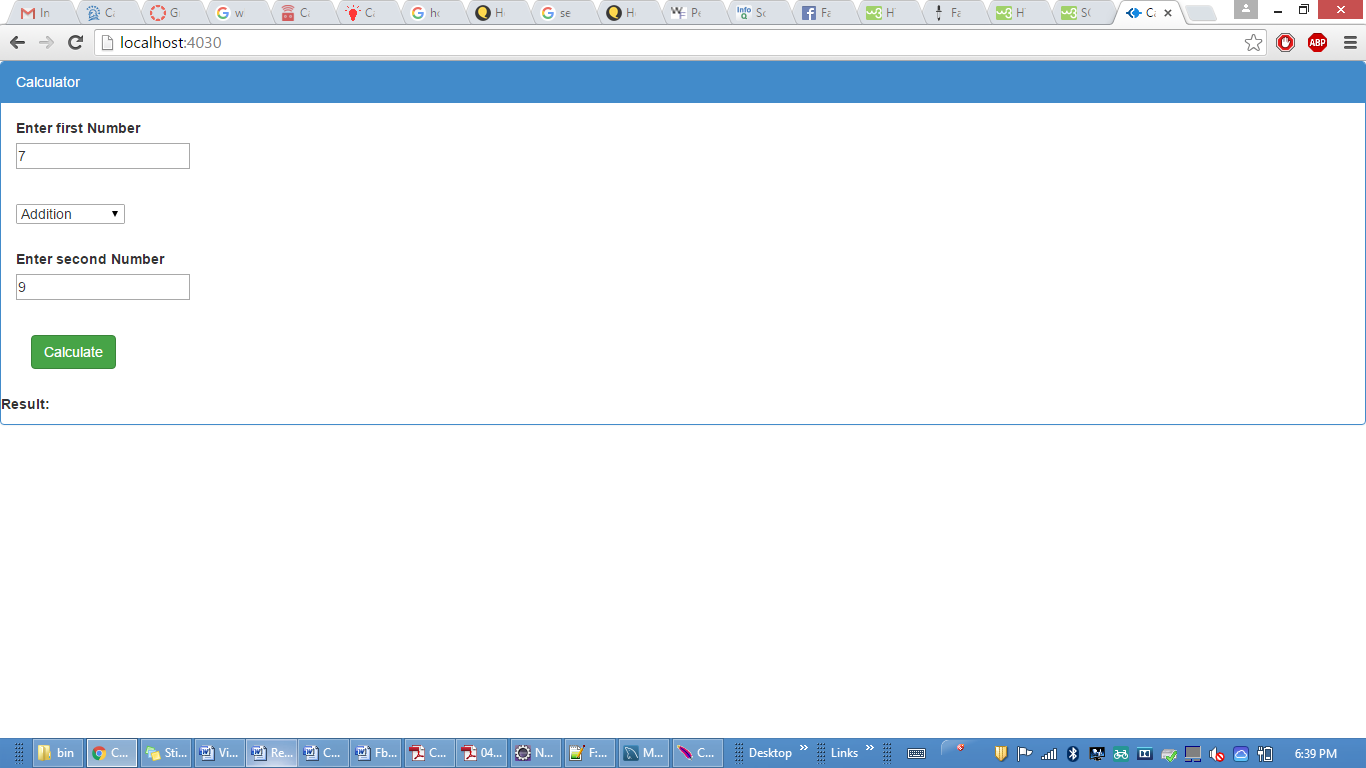
Division when second number is 0

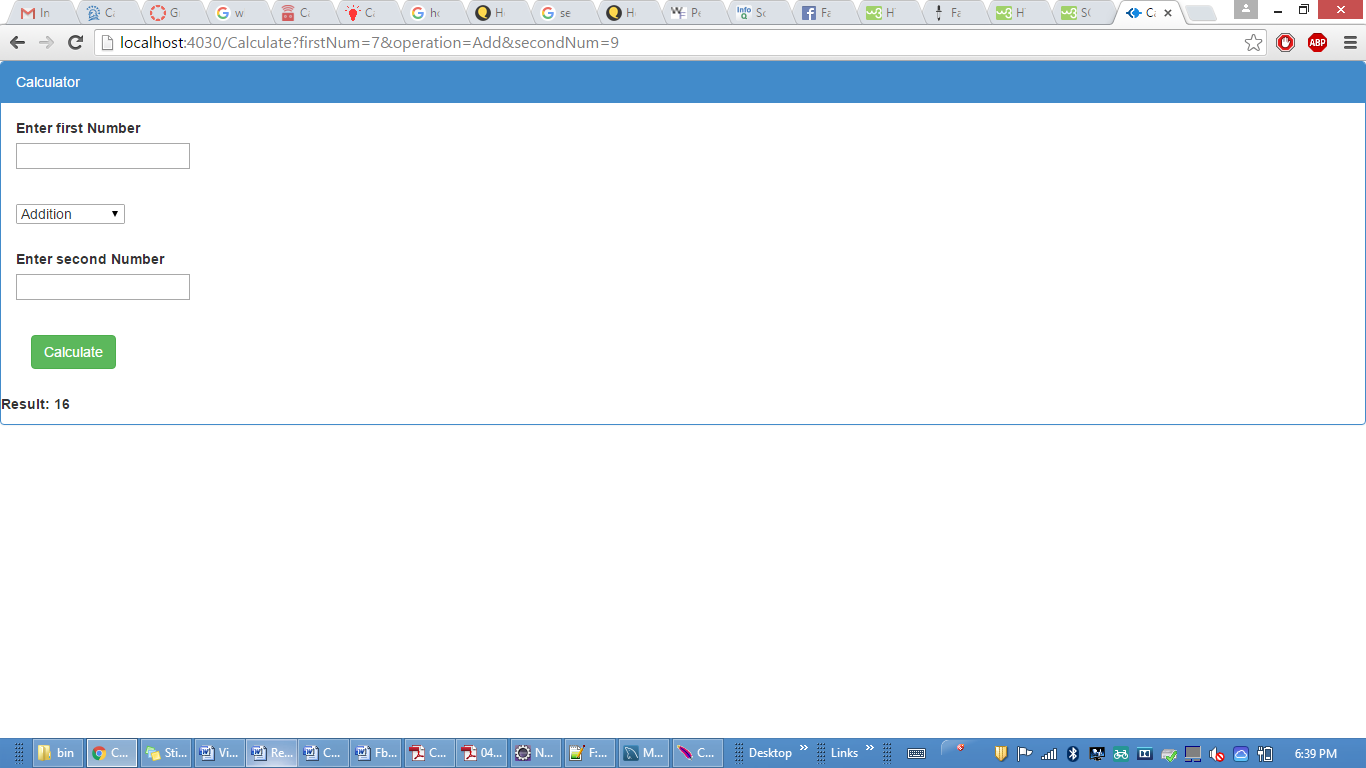




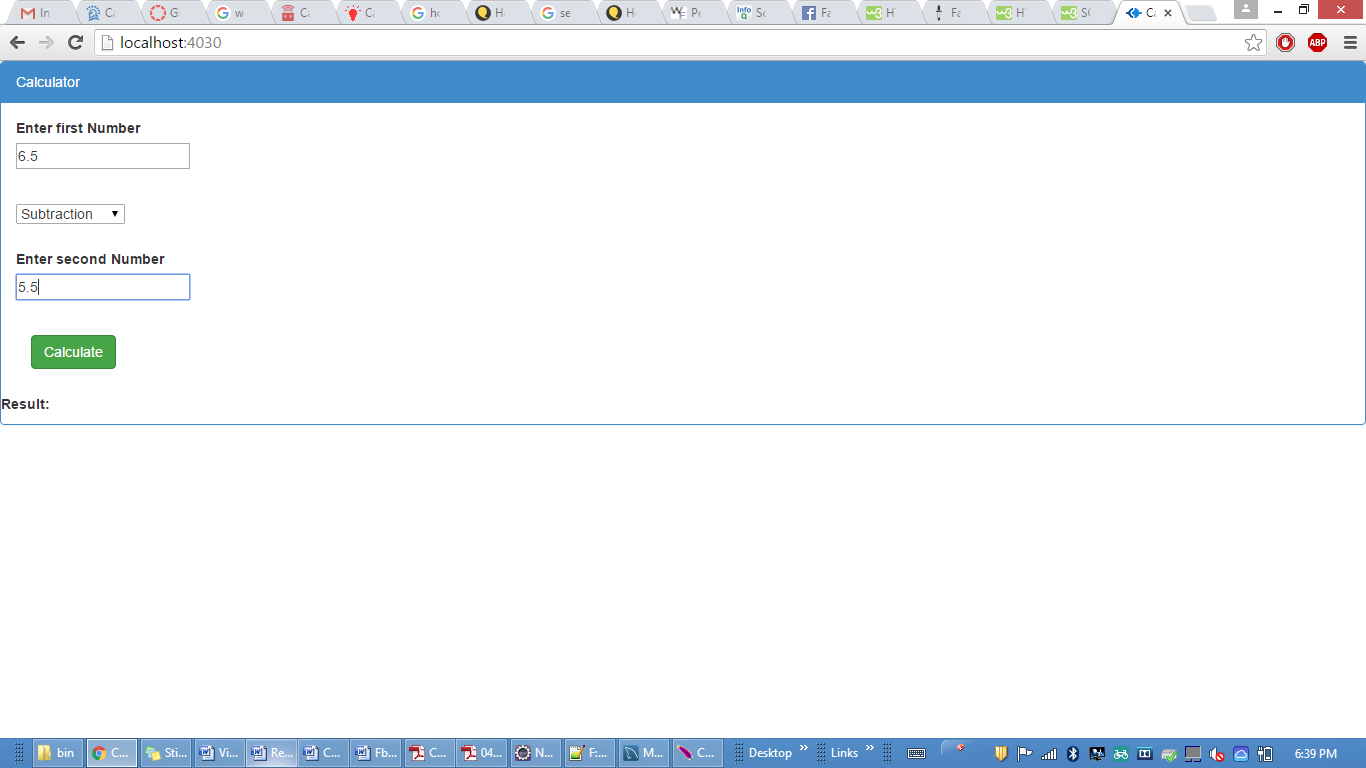
Correct Results

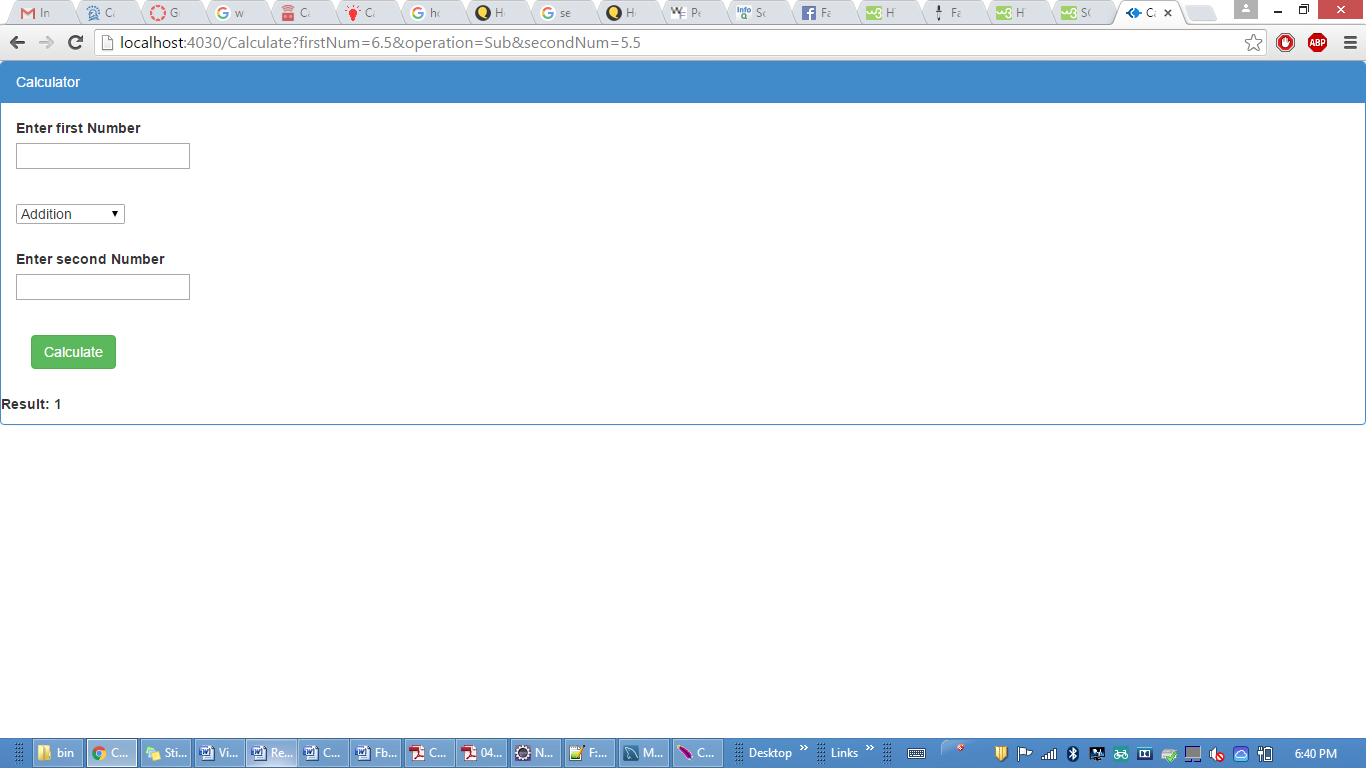
Addition



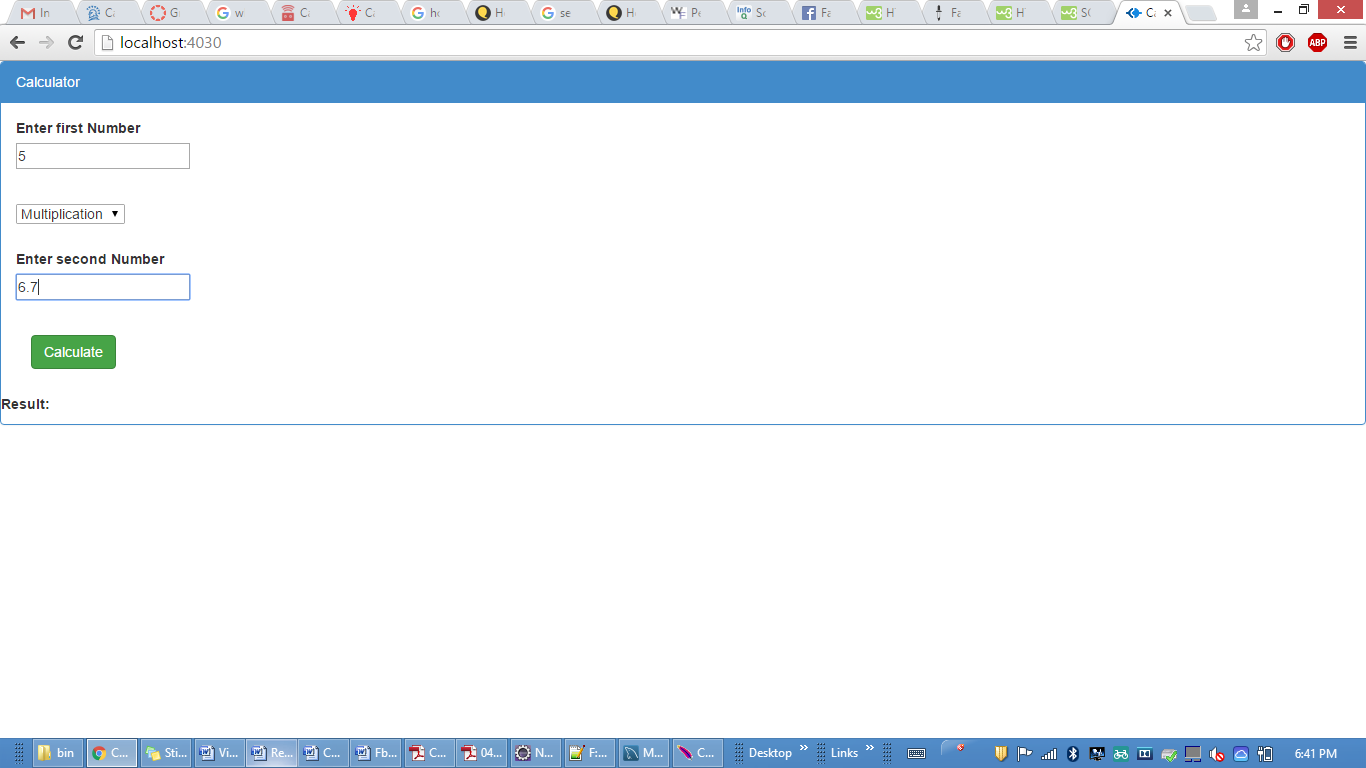


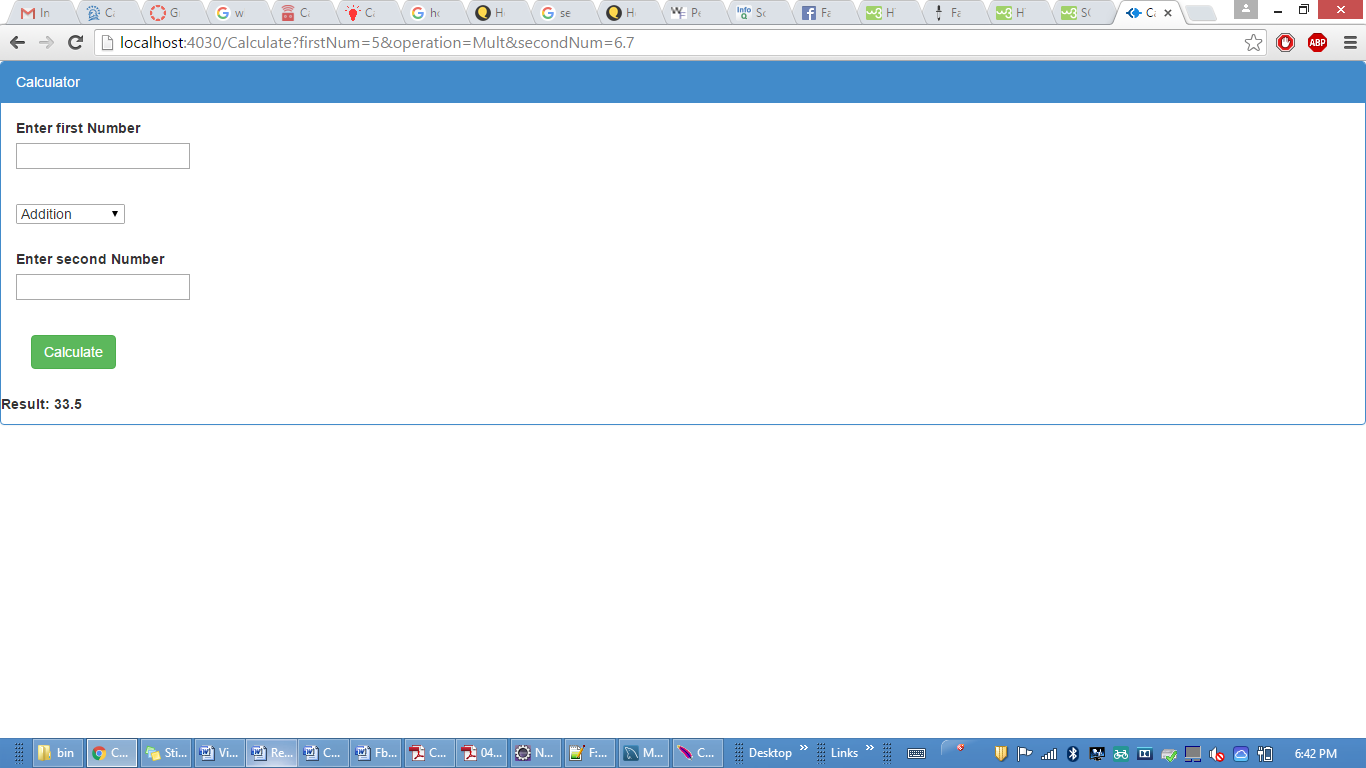
Subtraction



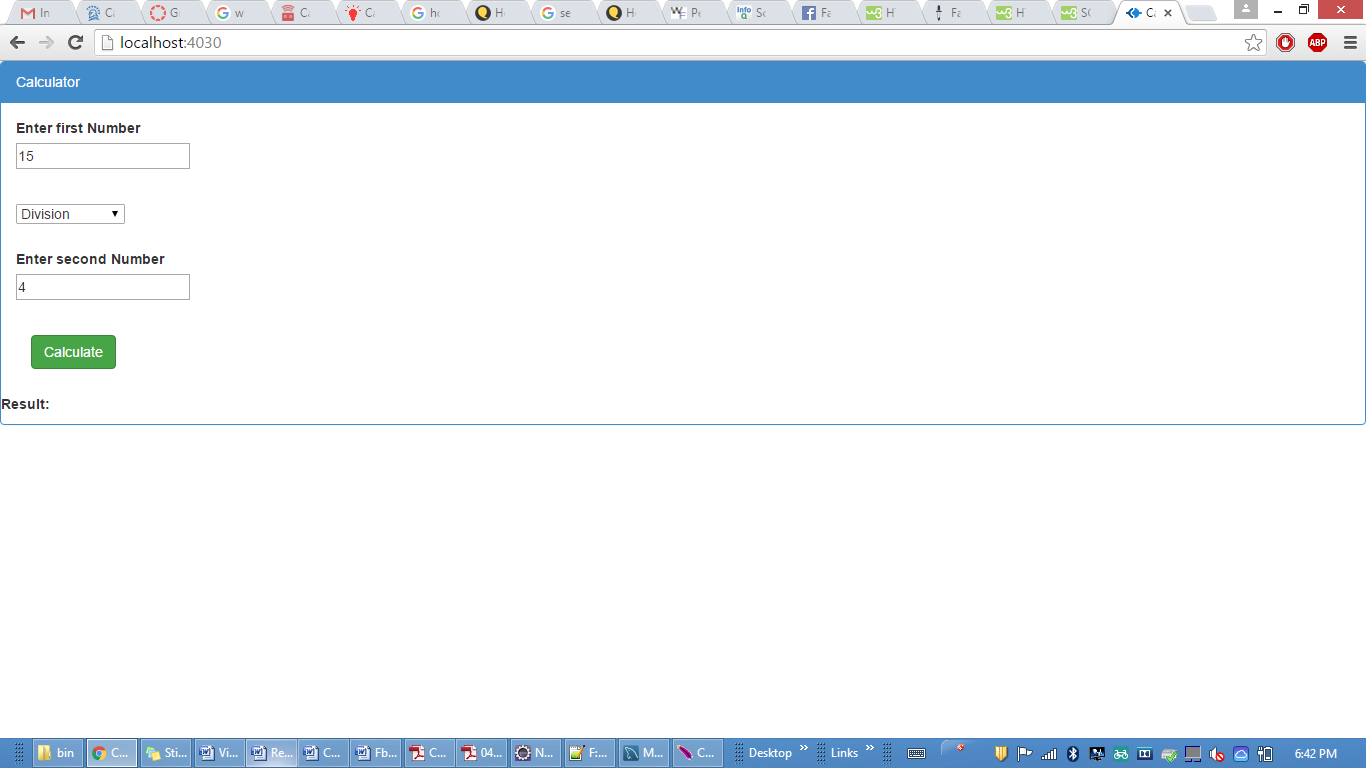


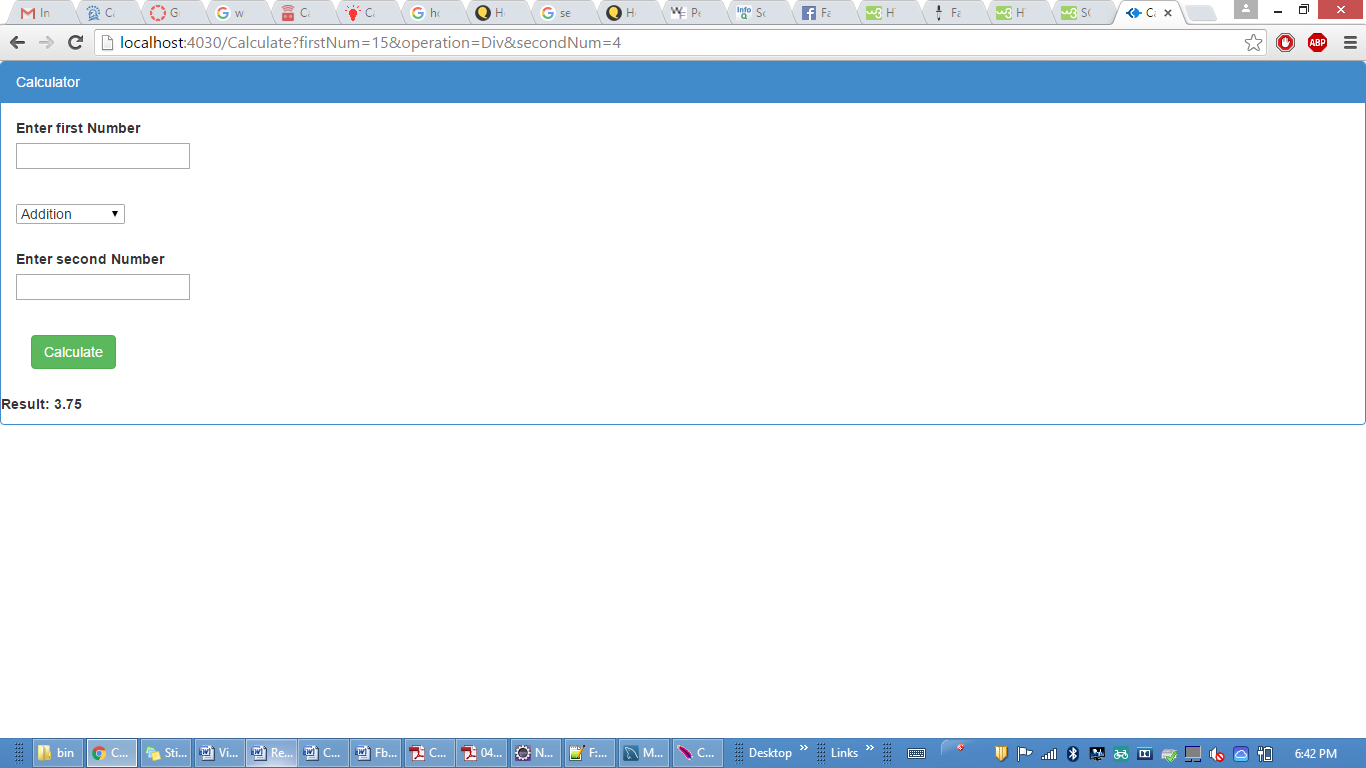
Multiplication





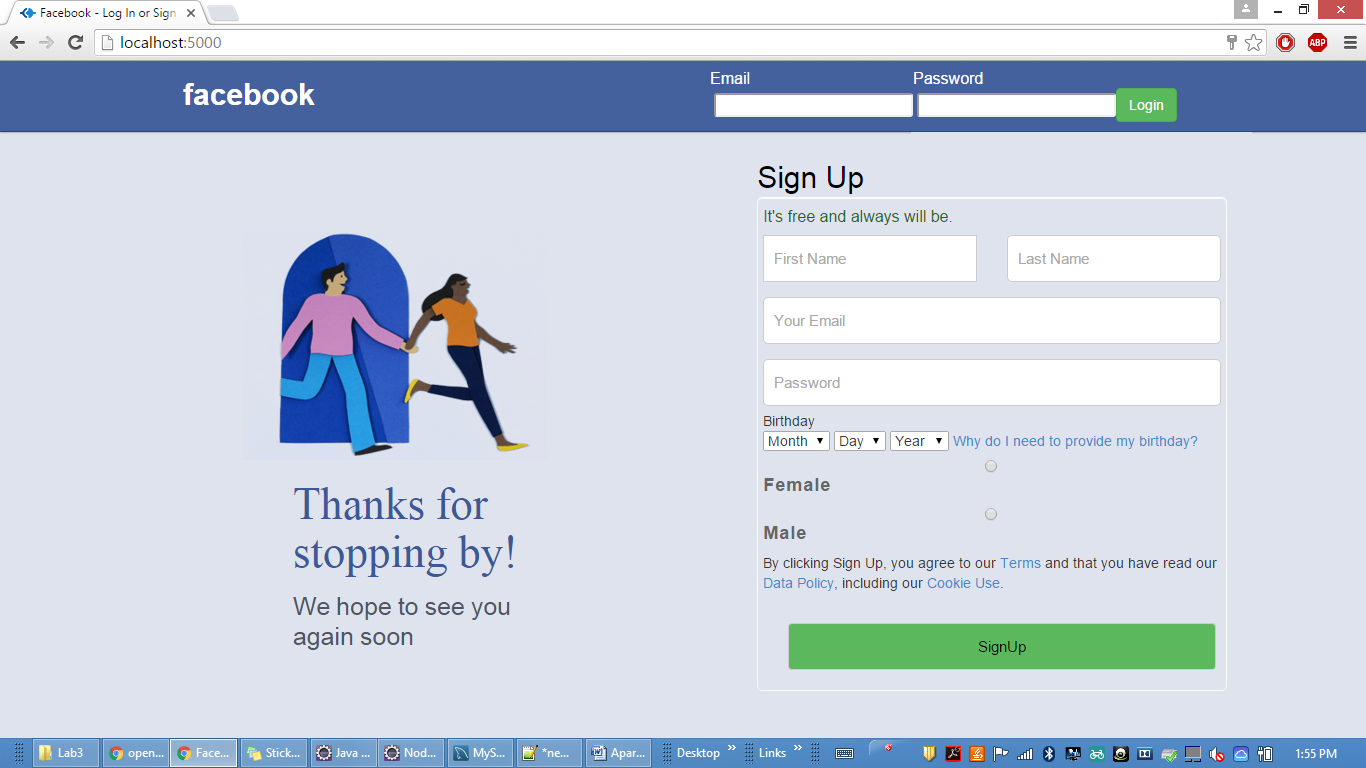
Division

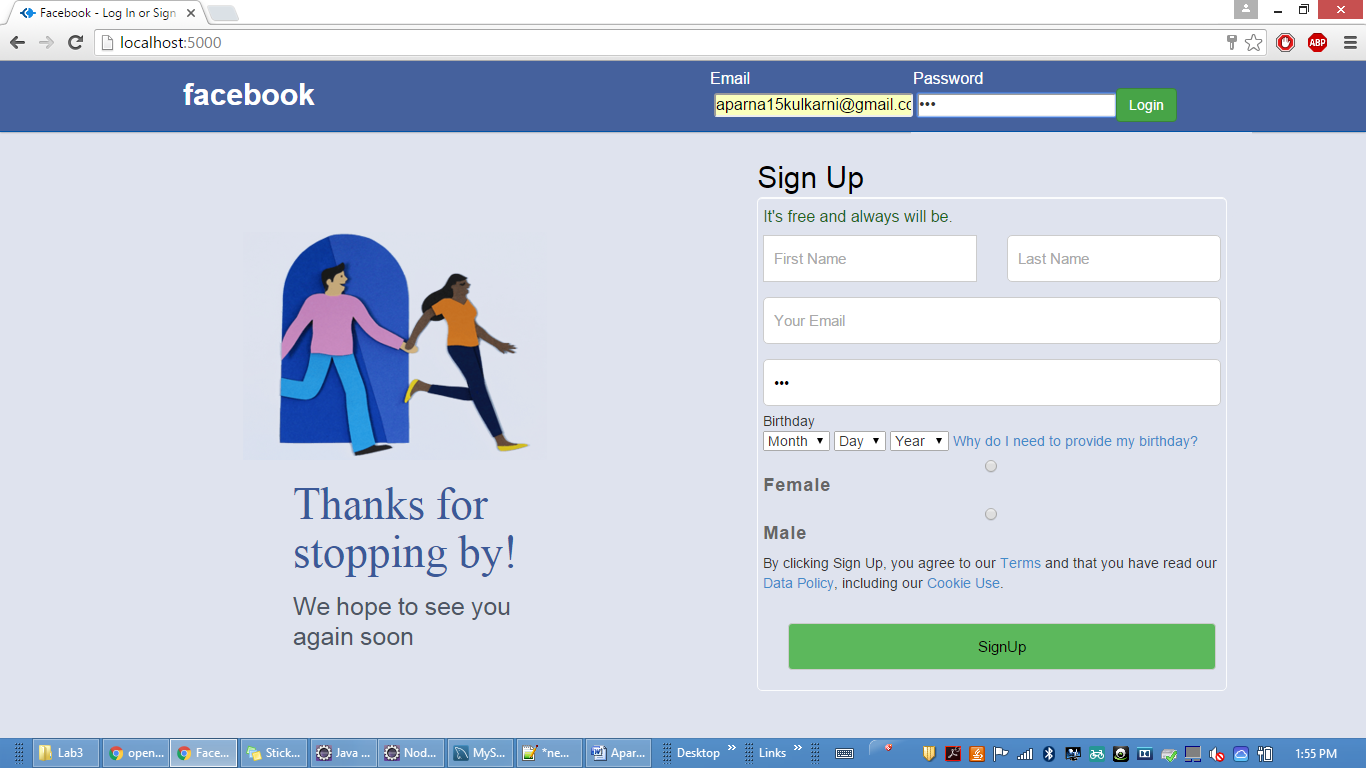




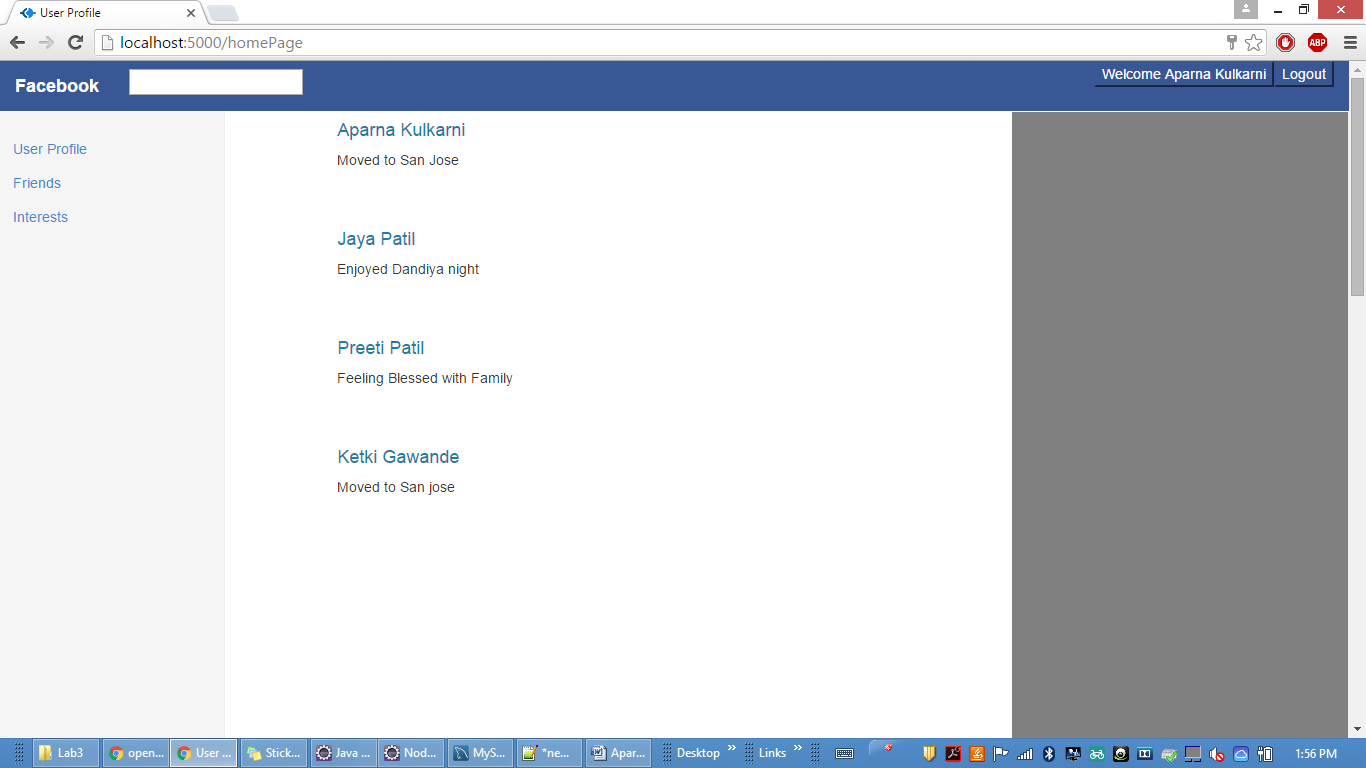
**Facebook**

SignIn user

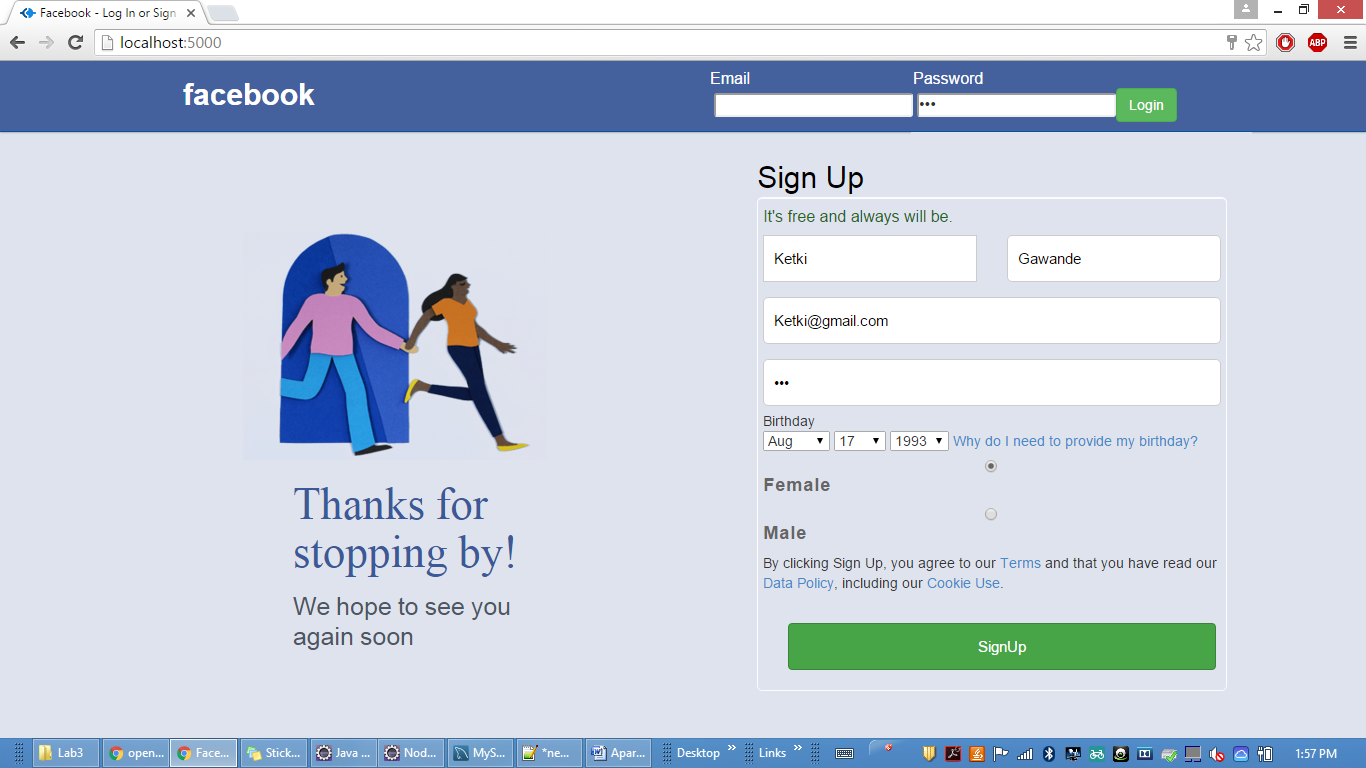




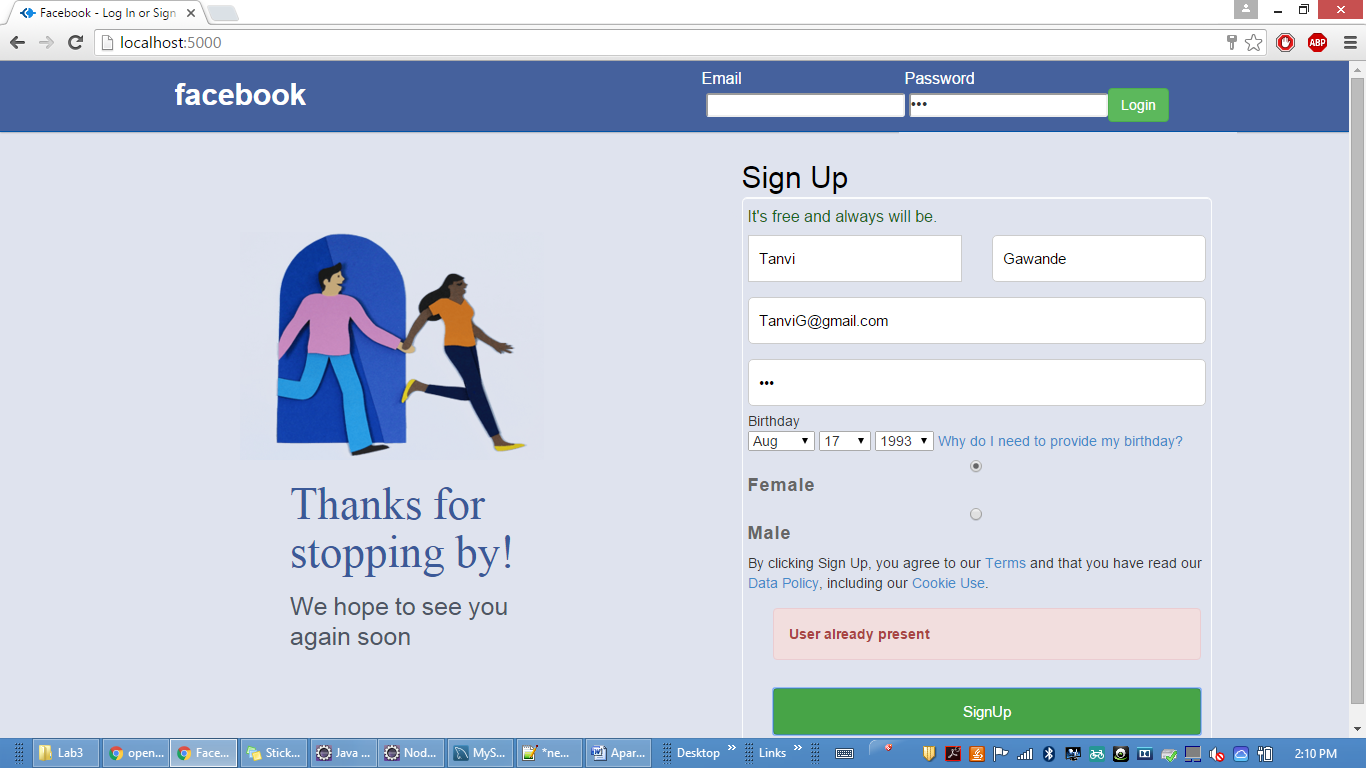
after signIn



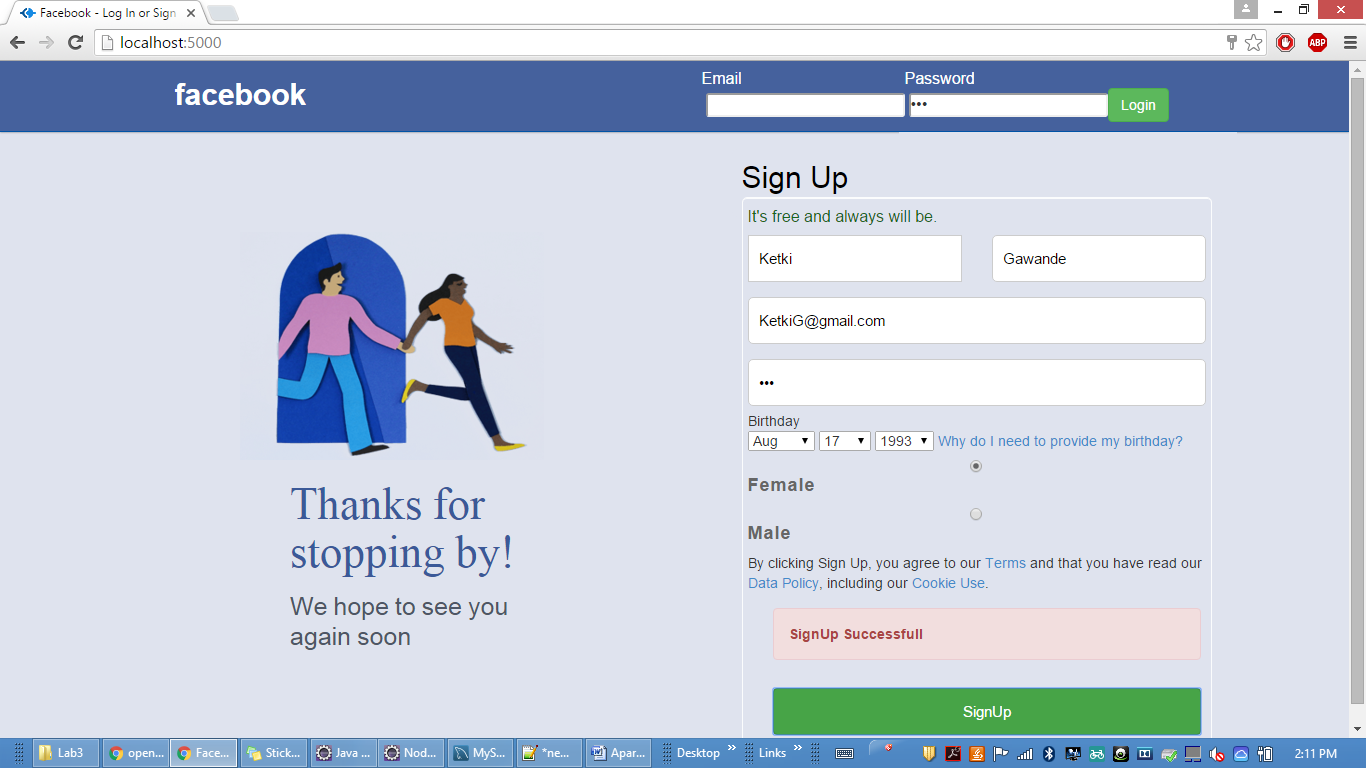
SignUp User



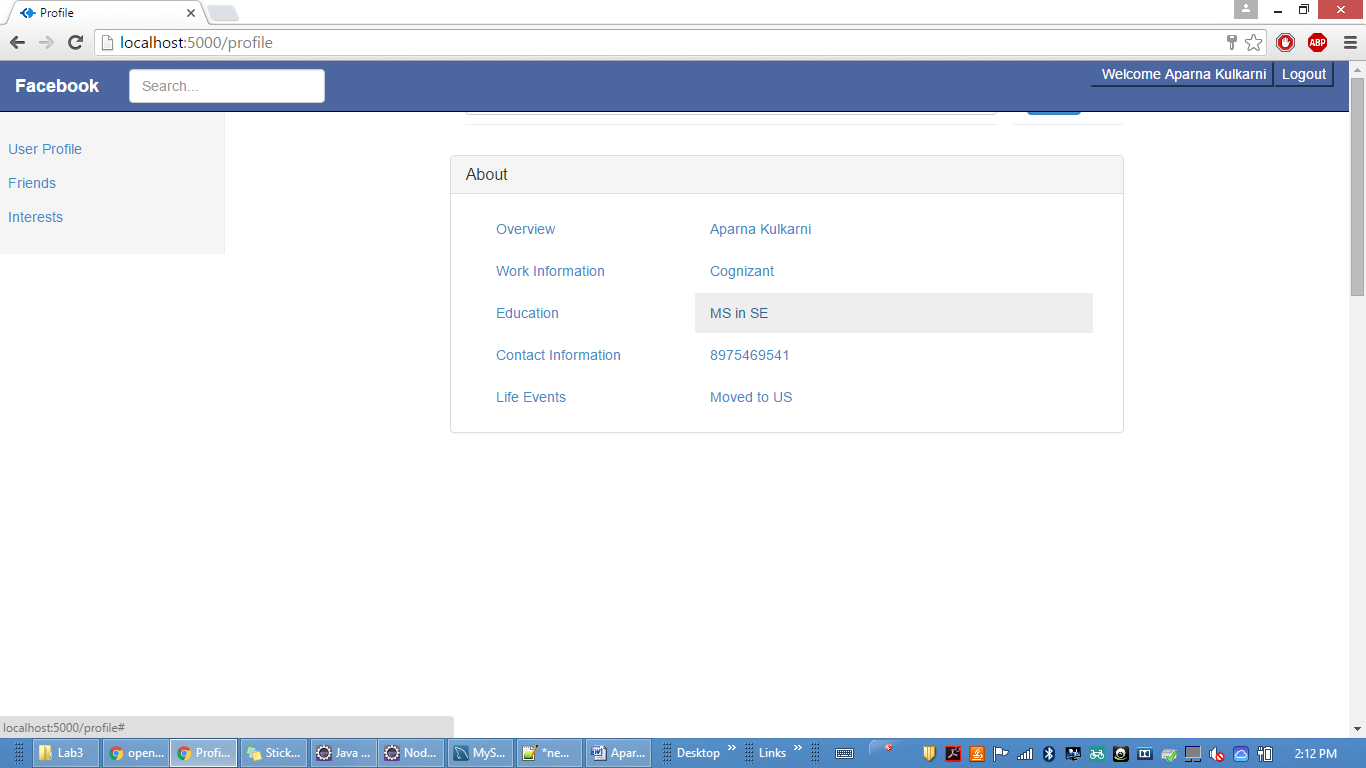
Error message is displayed if user already exists



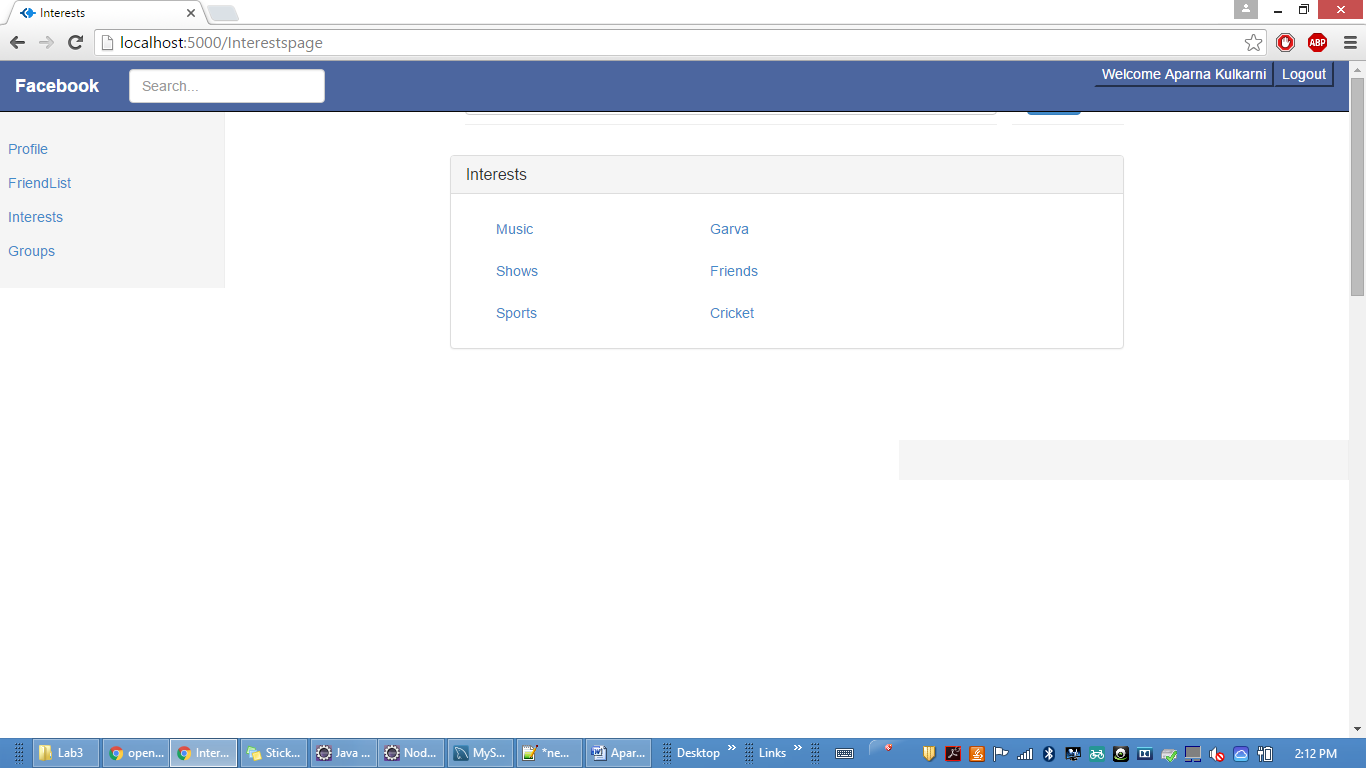
after successfull signup "success signup " message is displayed



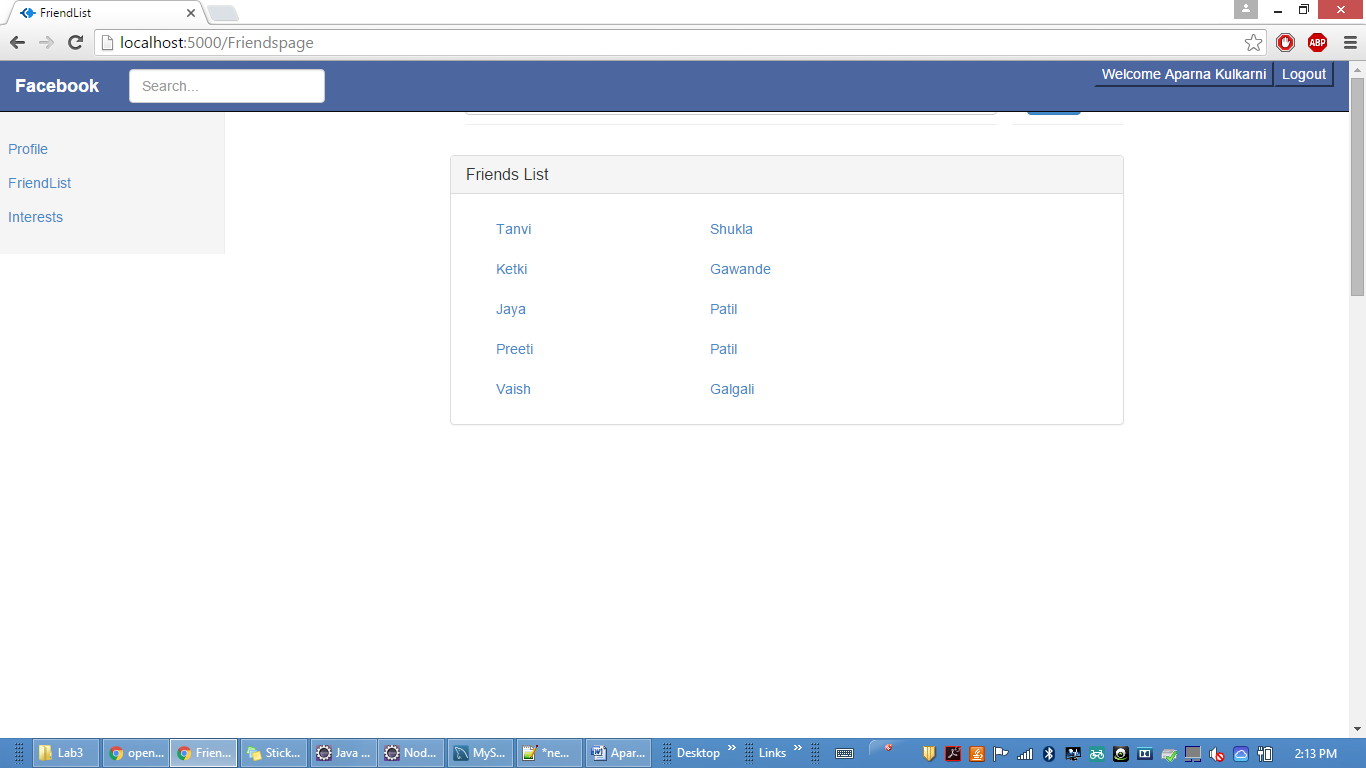
Profile page-About



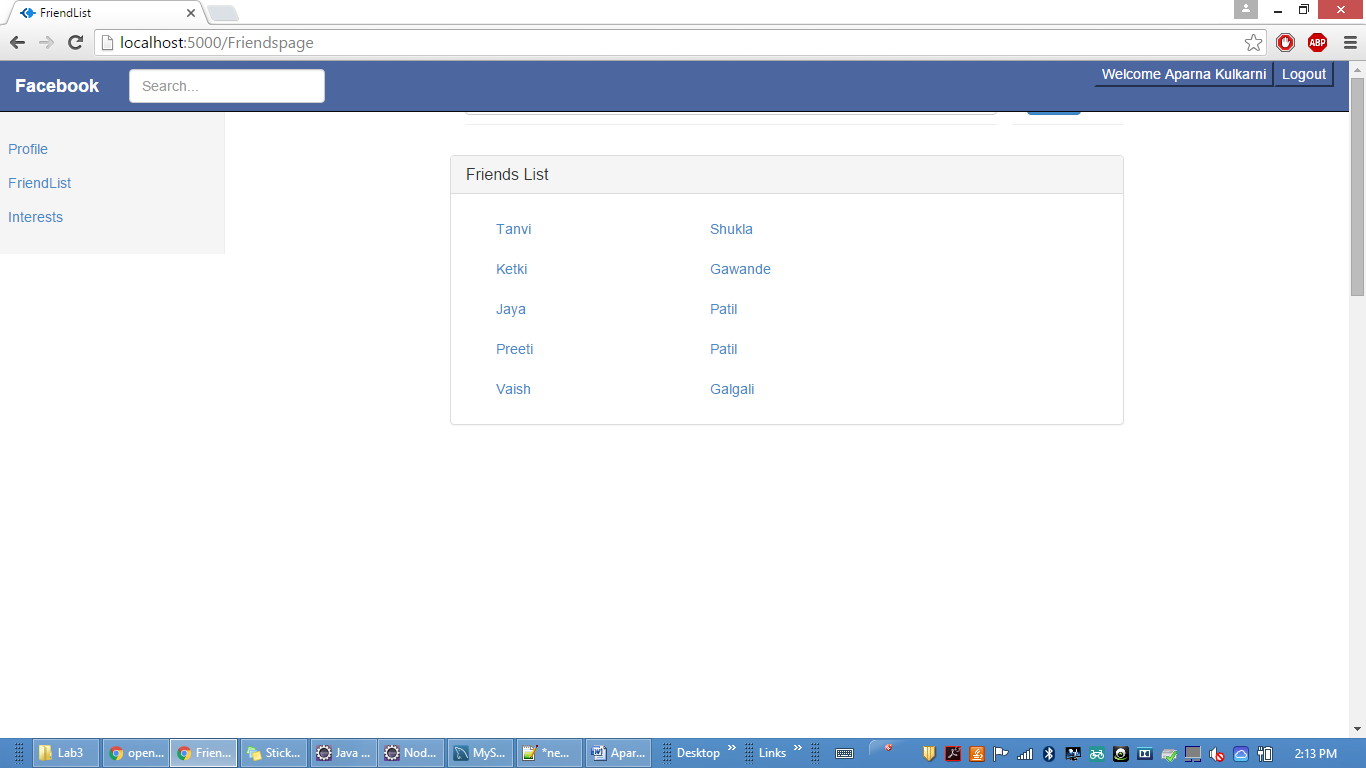
Interests



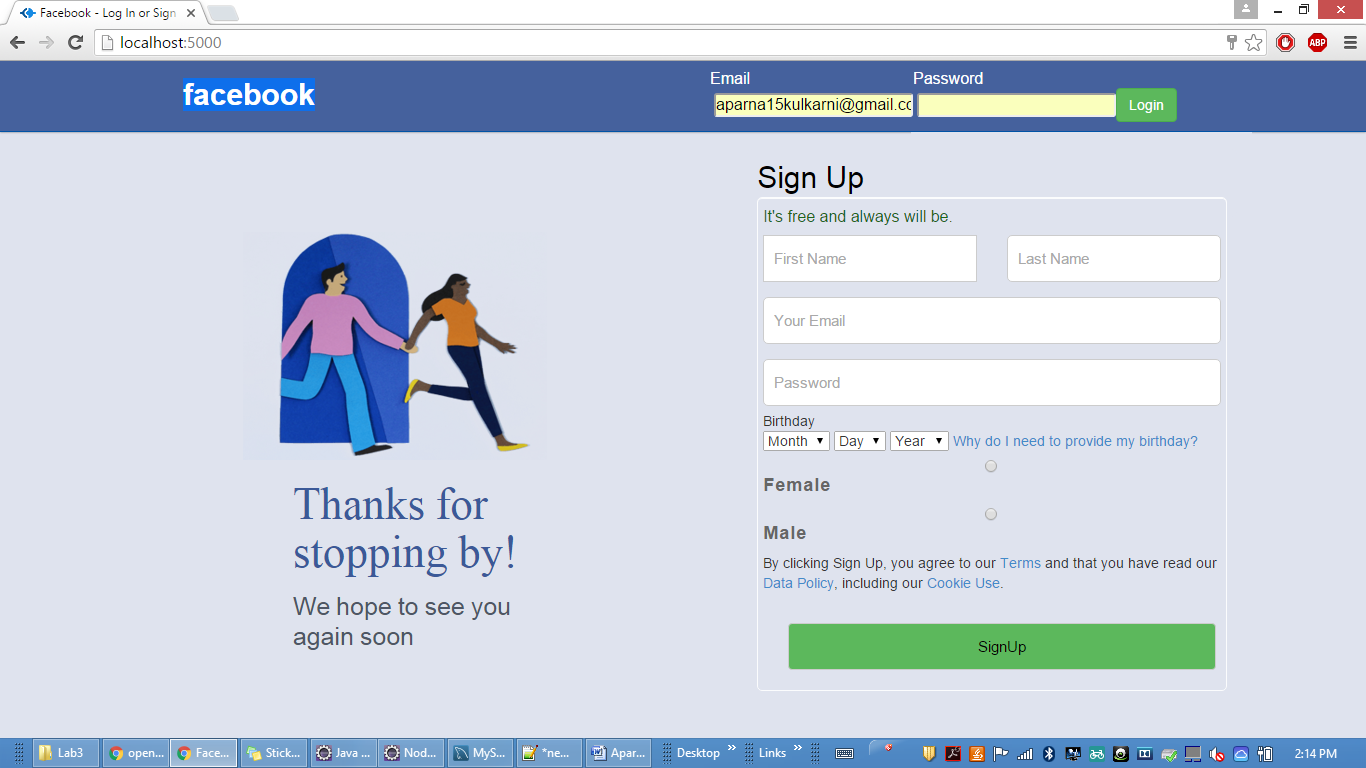
FriendList



SignOut click on logout button



clicking logout



**PERFORMANCES:**

Performance seems to have two meanings:

1) The speed at which a computer operates, either theoretically or by counting operations or instructions performed during a benchmark test. The benchmark test usually involves some combination of work that attempts to imitate the kinds of work the computer does during actual use. Sometimes performance is expressed for each of several different benchmarks.

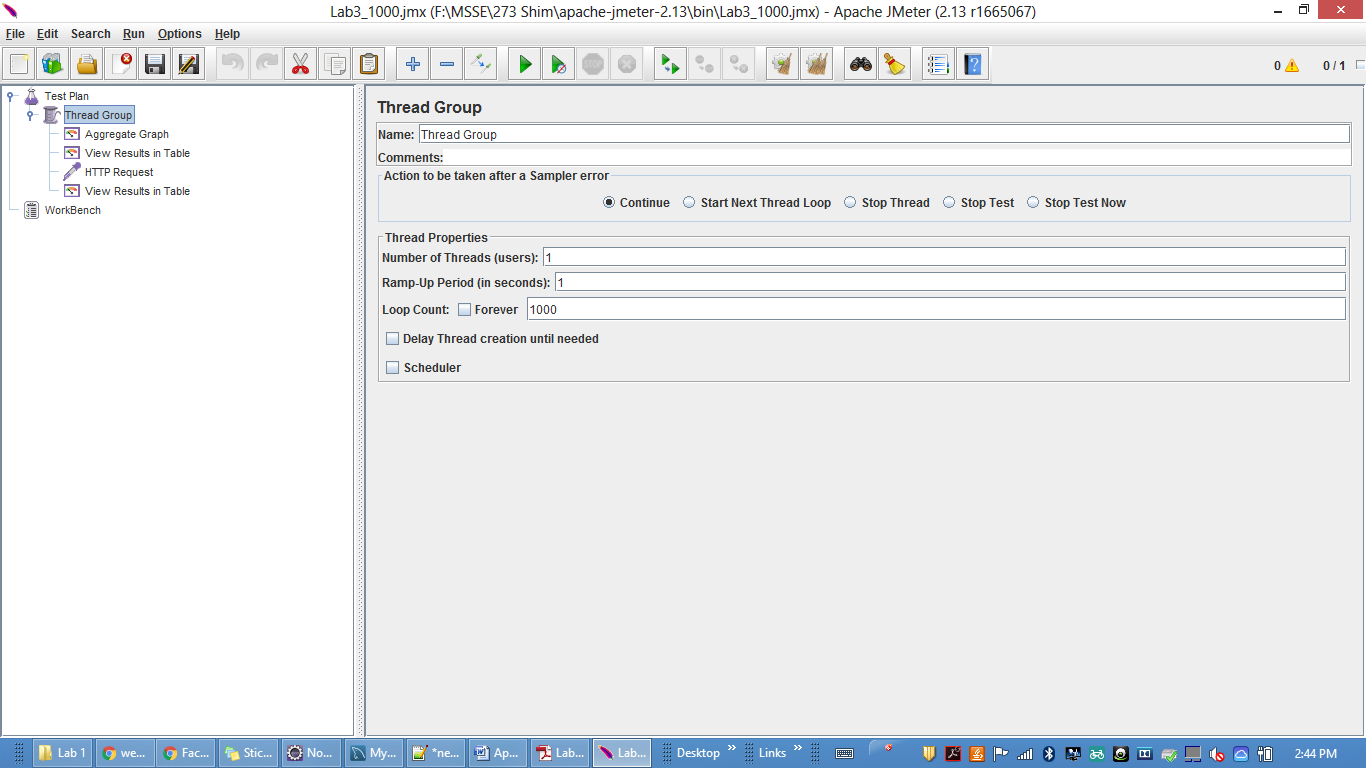
2) The total effectiveness of a computer system, including throughput , individual response time , and availability.

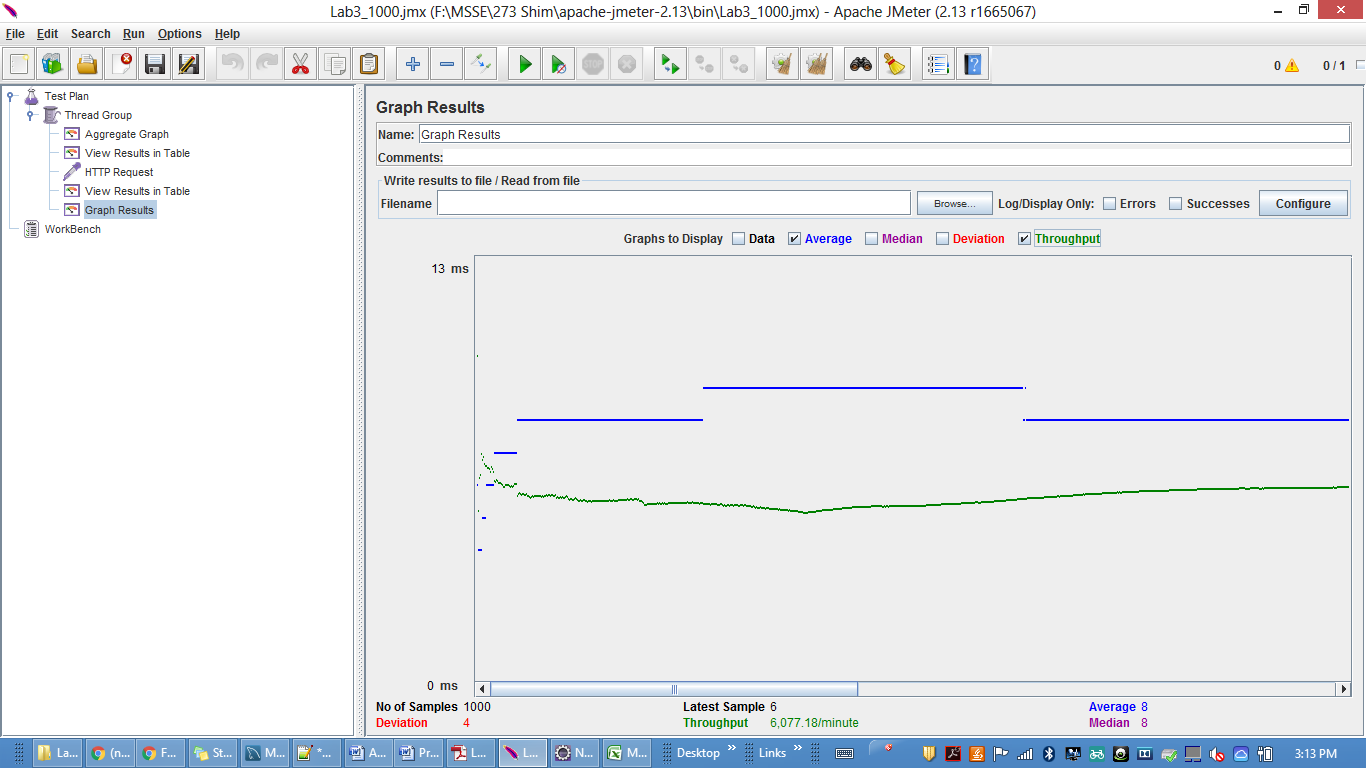
Performance testing for CALCULATOR is done by using JMeter.

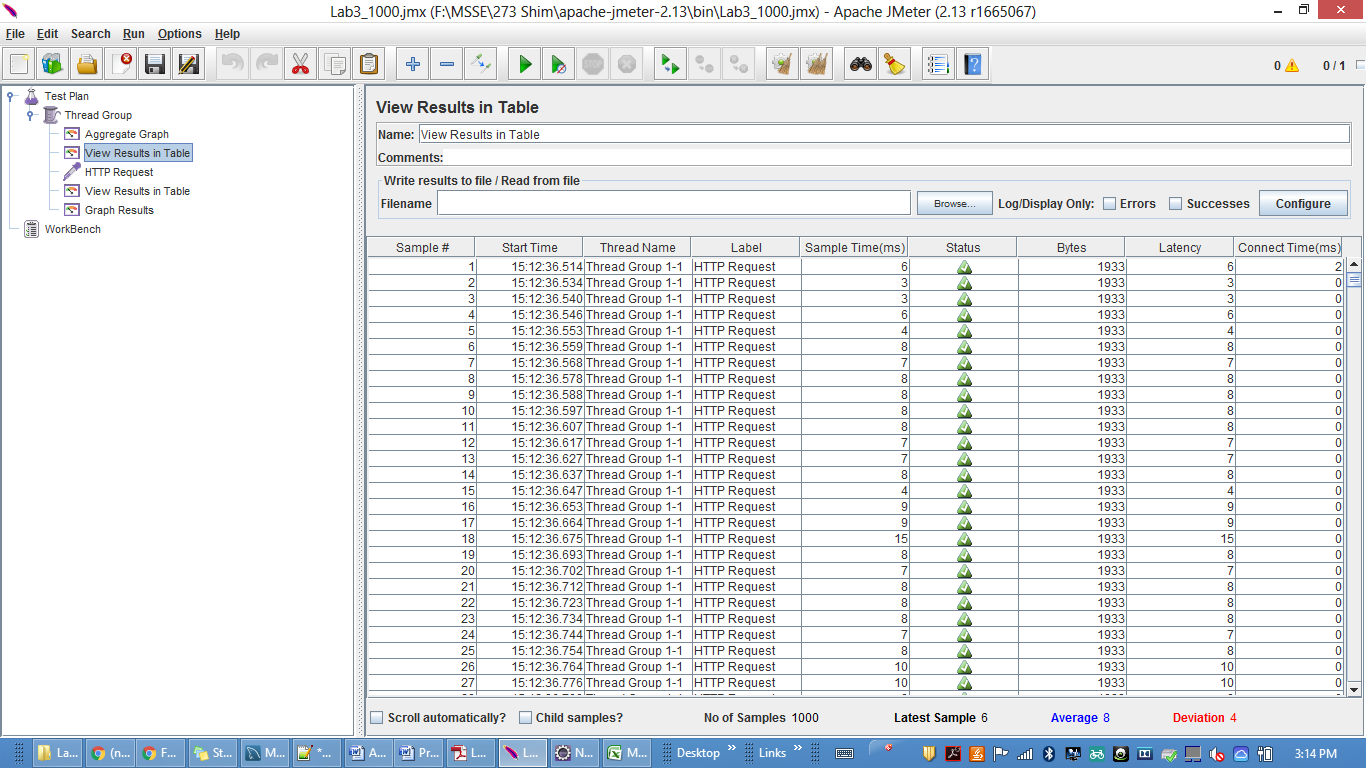
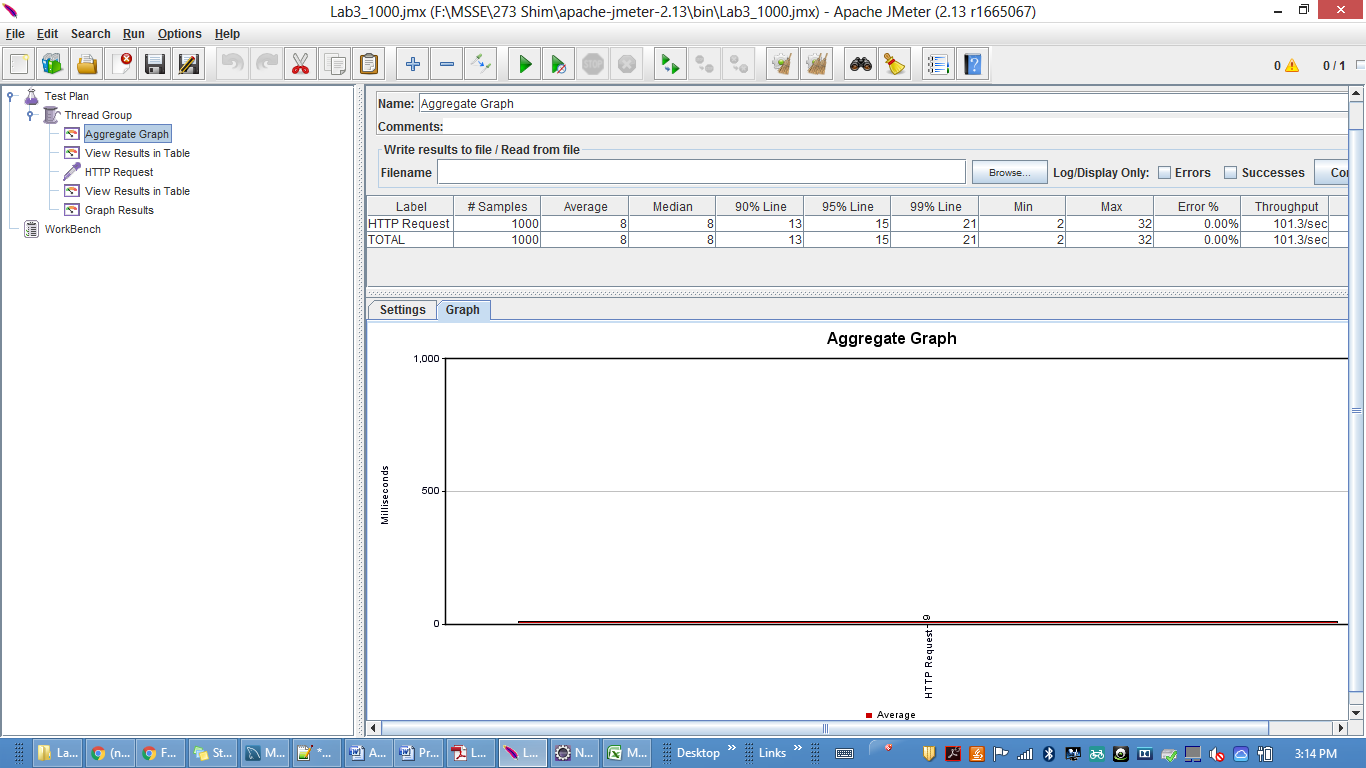
And a comparison chart is shown which shows performance difference in CALCULATOR developed using RESTful web services and JAVA web services using SOAP

**CALCULATOR:** using Random Controller in order to call all operations randomly.

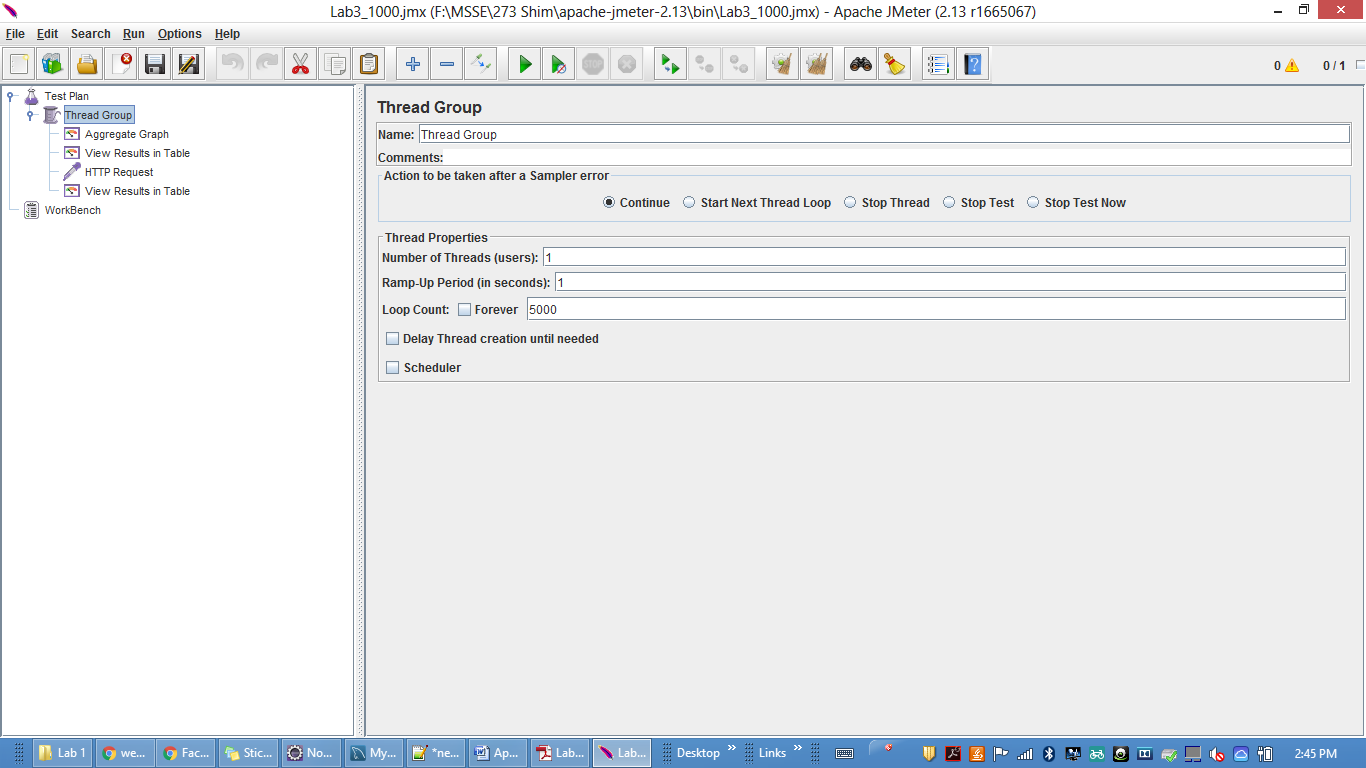
**Test 1 for 1000 randomly selected tasks:**

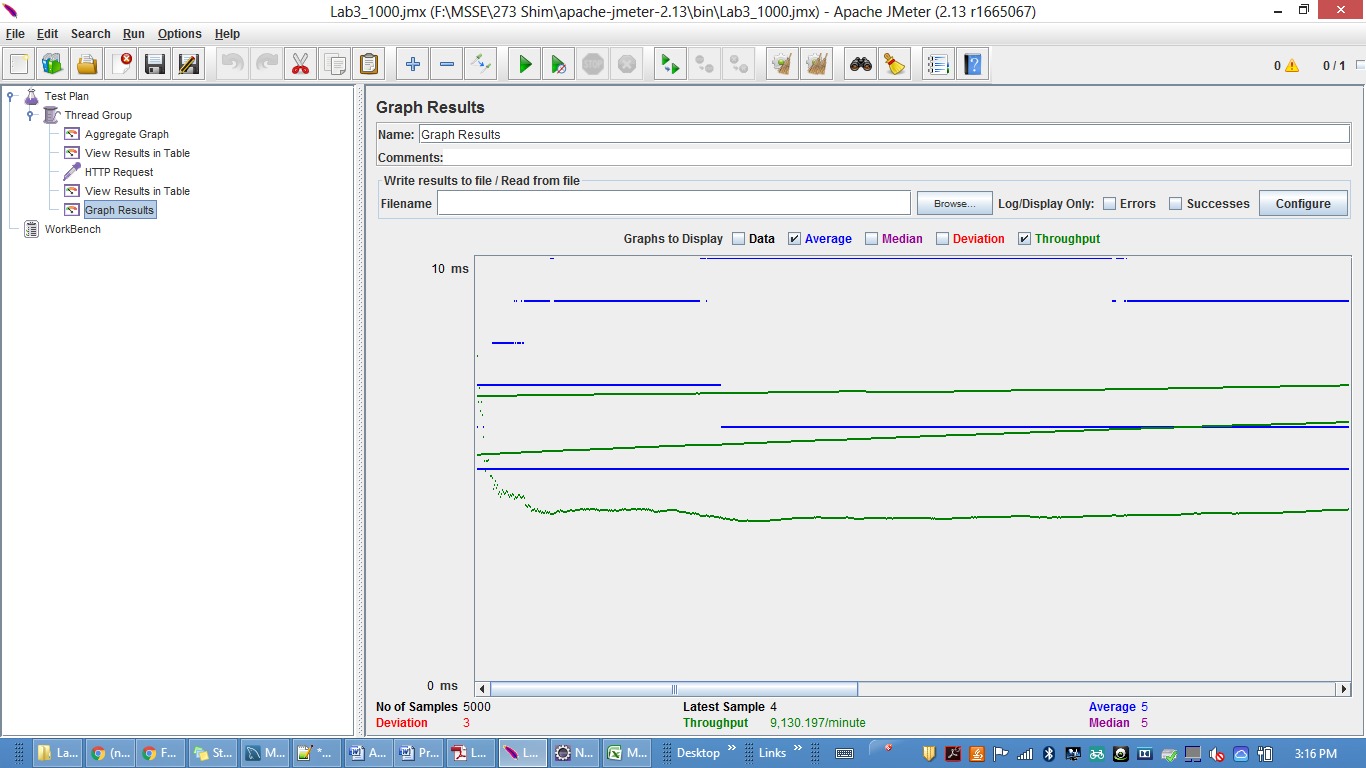
****

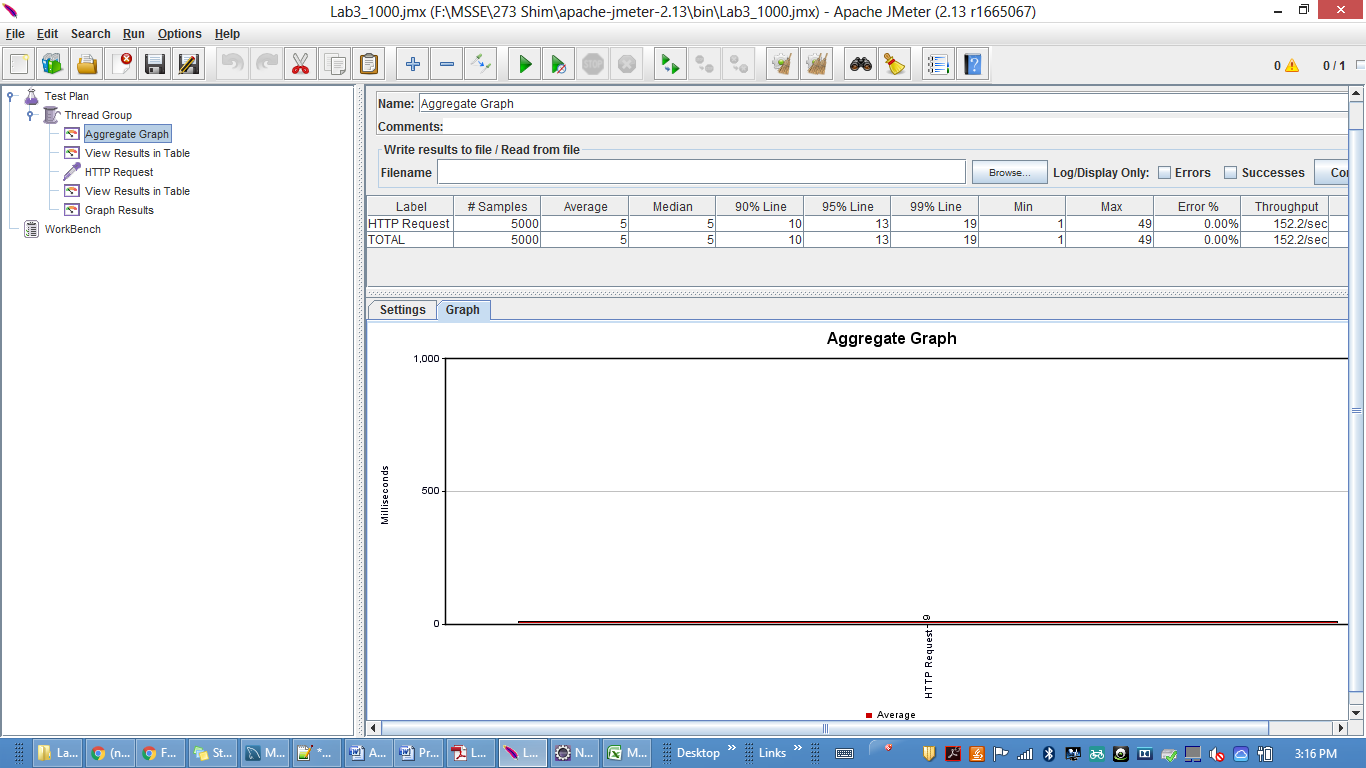
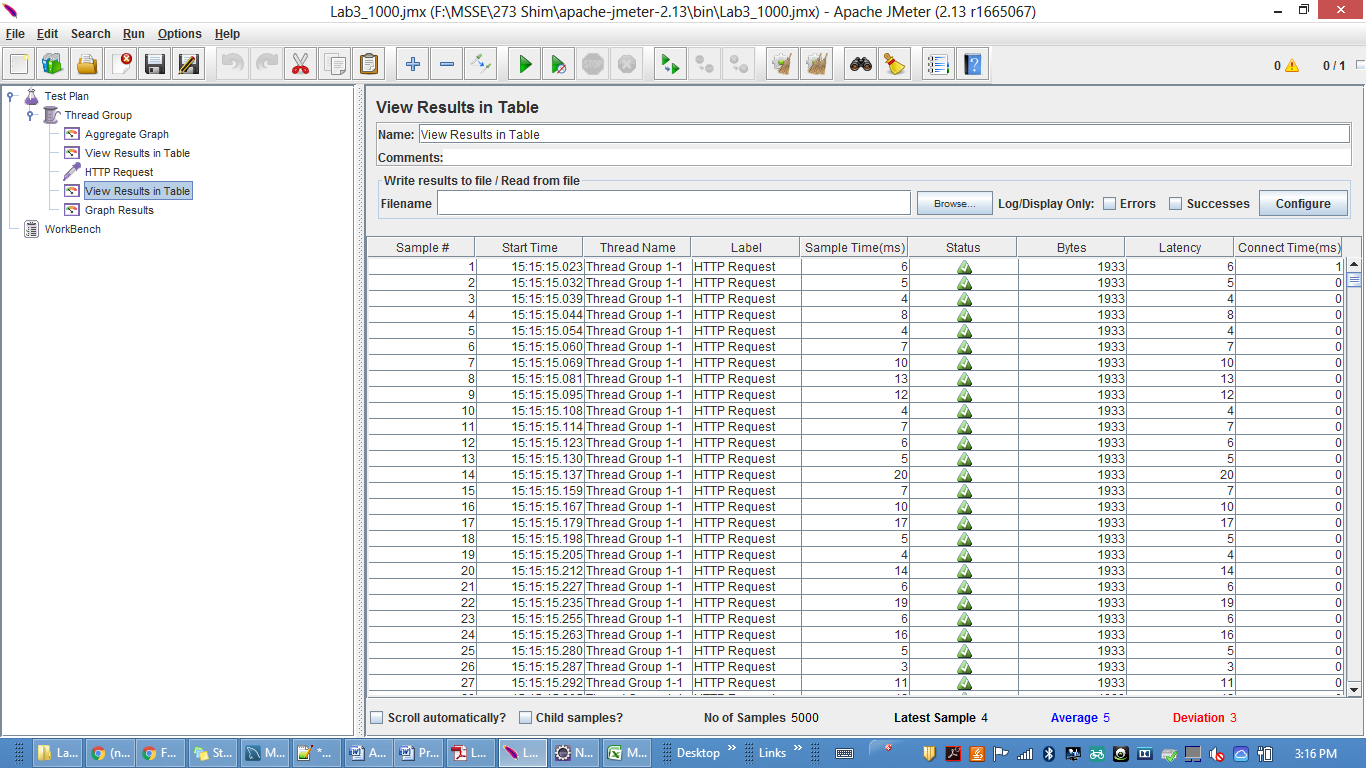
****

****

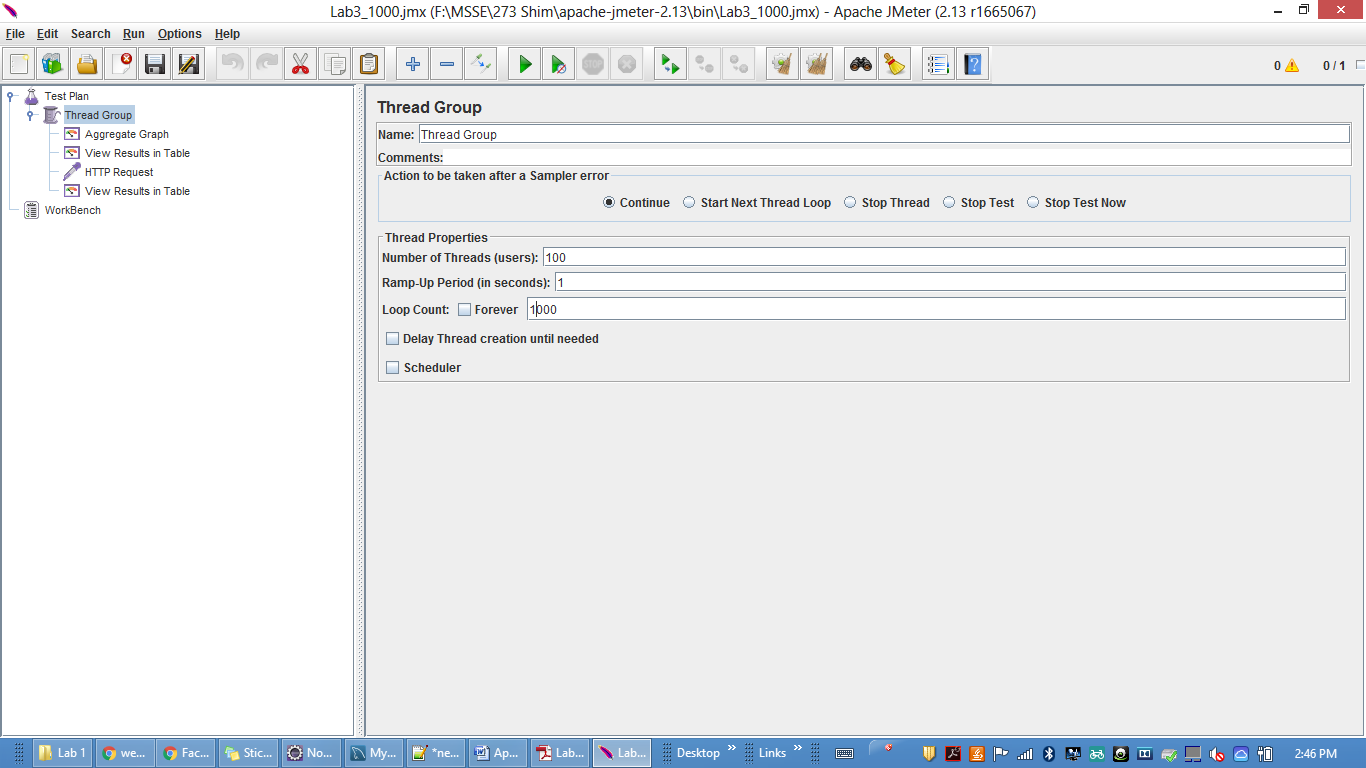
**Test 2 for 5000 randomly selected tasks:**

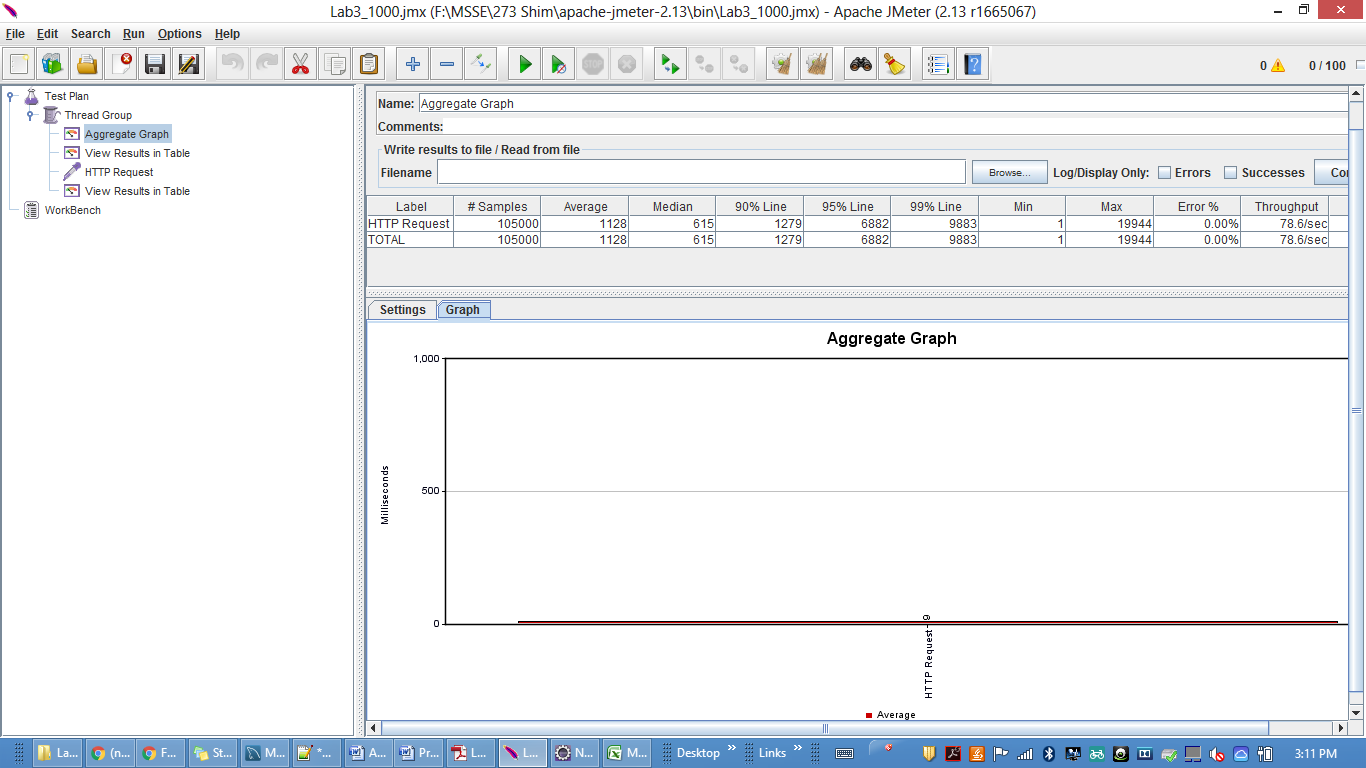
****

****

****

**Test 3 for 100 concurrent users 1000 randomly selected tasks:**

****

****

**GRAPH for CALCULATOR :**

|  |  |
| --- | --- |
| **No of Call** | **Average Time** |
| 1000 | 8 |
| 5000 | 5 |
| 100000 | 1128 |

**Comparison for the results with REST**

**Reasons are as follows:**

1. Load Performance:  REST has a bit better performance because it bears minimal overhead on top of HTTP. Usually SOAP brings with it a stack of different (generated) handlers and parsers. And it take time to load java files dynamically.

2. Network Performance : REST uses just JSON whereas SOAP uses WSDL .