***Enterprise Chat System***

Design Document V1.2

Date: 11-06-2017

**Team Name:**

**Wireshark**

**Team Members :-**

# 1.Ginka Anusha

# 2.GurramKarthik

# 3.Hossen Saddam

# 4.Injeti Ravi Varma

# 5.Ivvala Venkata Sai Krishna Chaitanya

# 6.Kadamati Venkata Sai Sidhartha

# 7.Karumanchi Mahesh Chowdary

# 8.KondepatiDivya Naga Krishna

# 9.KovvuruNasirali

# 10.Sathi Santhosh Reddy

# 11.JanagamAnirudh

**Type of Document:** Design Document

**Version Number :**1.2

**Publication Date:** June 11th, 2017

**1.Preface**

In this document, we are providing information about how we are going to design the chat system such that it can sustain internet breaches, and the Description about each software we use.

**Release v1.2 on 11-06-2017**

* Made major revisions to the document
* Updated section3, section 4, section 5
* In section 3, information regarding Restful API’s are included, and detailed explanation of each test are provided
* In section 4, Database designing steps are changed.
* In section 5, Administrator activities and the corresponding tests are clearly explained.

**Release v1.1on 28-05-2017**

* Made major revisions to the document
* Updated section1,section 3
* Updated section 3 – Added suitable tests for front end module
* Updated section 4 – Added database structure and respective table structure
* Updated section 5 – Added suitable tests for backend module (user&admin)
* Few tests which are part of system requirements have been added

**Release v1.0 on 07-05-2017**

* **Initital release**

To explain this, we have divided this document into three Modules (Front end Module, Database Module, Backend Module).

The remainder of the document is organized as follows.

Section 2  Glossary and abbreviations,

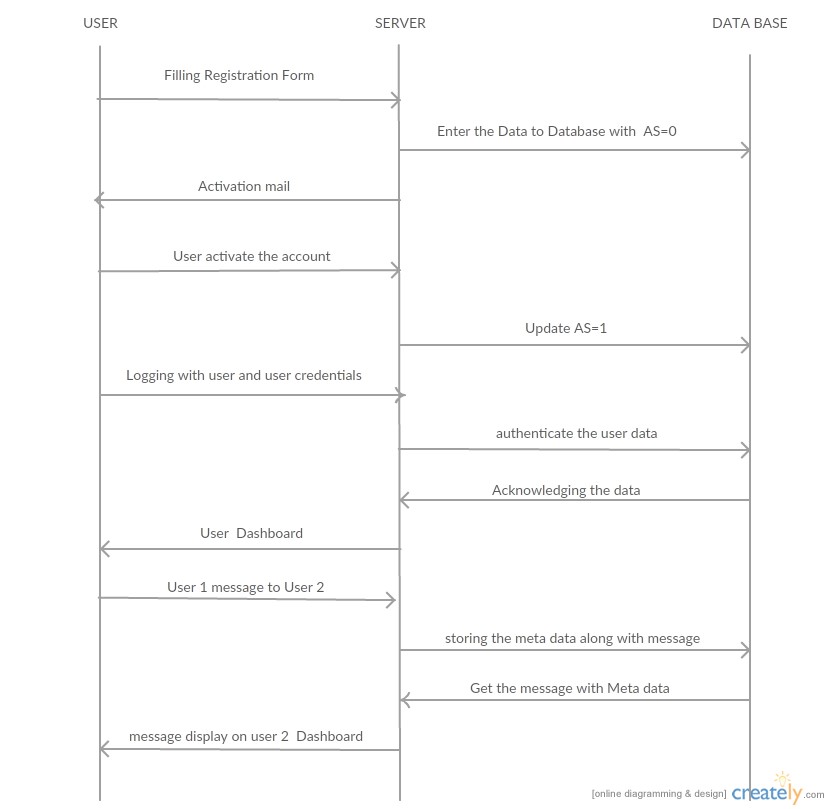
Section 3  Installing the XAMPP server

Section 4,5,6  Different Modules and their detailed design and unit test plan. Section 7  References

**2.Glossary and Abbreviations**

* **GUI:** Graphical User Interface.
* **API:** Application programming interface
* **SQL:** Standard Queuing Language.
* **IP ADDRESS:** Internal Protocol Address.
* **PHP:** Hypertext Pre-processor
* **HTML:** Hyper Text Mark up Language
* **JSON:** JavaScript Object Notation
* **REST:** Representational State Transfer
* **API**: Application program interface
* **MYSQL:** An open source relational database management system
* **PHP:** Hypertext Pre-processor.

# 3.Module1: Front end module

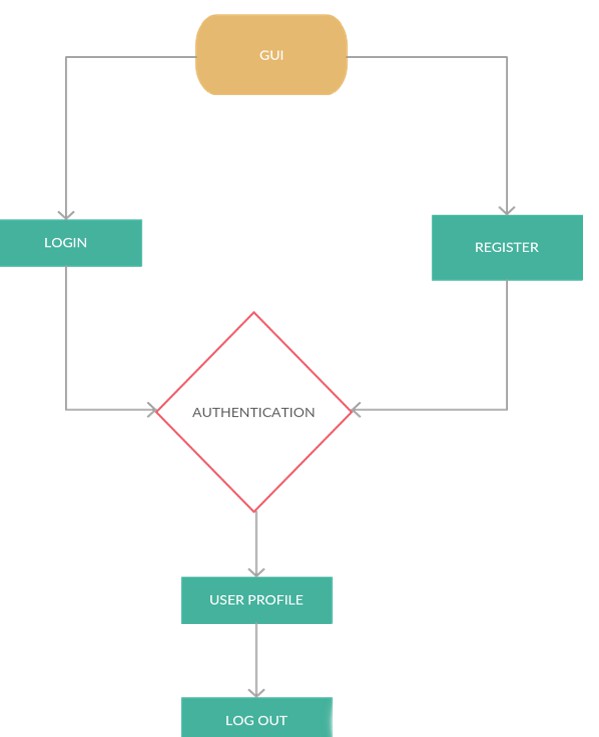


**Fig 1**: Front end Module

**RESTFUL API** is used between user to server communication along with **JSON** data encoding.

The fundamental module contains the following three elements

1. User
2. Server

 Front end – User functionality module

**3.1 User**

The front end of user interface modules consists of the GUI, login, register, authentication, user profile page, logout.

1.GUI provides a web based interface for a smooth user interaction.

2.When user visits the home page, he will be provided with an option to register for the first time to use the chat features.User provides necessary credentials and can register himself.

3.There will also be an option to login to the system after he has successfully registered.

4.After the user provides his login credentials the user credentials will be authenticated.If the details match the userwill be redirected to profile page where he can configure his account.

5.Logout option will also be provided apart from all these if the user wish to destroy his current session.

**Unit test plan**

The following tests can be performed by the front end of user Interface module.

**Test: F-T1**

**Purpose**: It is used to describe the GUI for the users.

**Requirements: UR1,UR2,SR1,SR2.**

**Environment**: In this project HTML, CSS & JAVASCRIPT are used at front end to display the GUI.

**Operation:**After integrating remaining parts of the project like Database, server etc we can see our GUI at a node by typing the file path of localhost in the browser i.e., <http://localhost/filepath.php>. (NOTE: This project is made for the local users who are connected to switch so by using internet and by following this link will not let you go to this application, it is exclusively created for the users who are connected to the LAN by means of a switch)And testing of signup and login of users is done, if tests (all tests) got passed then our product is ready to deliver.

**Expected result**: Web pages can be accessed by the user.

**Result:** Success

**Comment**: A dynamic webpage is displayed to user

**Test: F-T2**

**Purpose**: To provide user registration option through the webpage

**Requirements: UR1,SR1,SR2.**

**Environment**: XAMPP is the integrated frame work for PHP. It will have APACHE server and MYSQL database. In the project HTML, CSS &JAVASCRIPT is used for the display of the GUI. In the project, PHP version 5 is used and it is embedded with the HTML code. MYSQL database is used to store the users details and a friendly GUI has been included in the XAMPP frame work to access the database through the following link: https://localhost/phpmyadmin and PHP scripts will be stored in XAMPP/ht docs from the web browser. We can access the through the following link http://localhost/wireshark

**Operation:** It can be operated by typing the file path of localhost in the browser i.e., <http://localhost/filepath.php>.

**STEPS:**

**Step1:**First Signup an account by filling all fields under signup registration form and choose a password for our chat system then click on sign-up button, a mail is sent to your email address with a link, if you follow that link for the verification you can be a member in the database.

**Step2:** successfully signing up is one of the test.

**Step3:** Then login with the signed up detailed in the login field, if you successfully logged in, that means another test got passed.

**Expected result**: User must be able to provide respective credentials to register himself through the webpage.

**Result:** Success

**Comment**: User can enter respective credentials like first name, last name, email etc. to register himself through the webpage.

**Test: F-T4**

**Purpose**: Configure user account

**Requirements:** UR4,SR1,SR2

**Environment:**In this project HTML, CSS & JAVASCRIPT are used at front end to design user account. In this phase unit testing is done through HTML code, all the user desired operations can be done when it successfully integrated with Database.

**Operation:** After integrating this page to XAMPP frame work we can do all the testing as per our client requirements, It can be operated by typing the file path of localhost in the browser i.e., <http://localhost/filepath.php>)

**STEPS:**

**Step 1:** Firstly, in the chat window user will select his recipient to whom he want to send message and will type the message in the text box and hits the send button to send the message, and our text will be sent and shown in the above dashboard with feedback message with tick mark(if recipient sees the message) if not it will show cross mark means message is not shown along with this a time stamp is appeared to show how much back that message arrived to our inbox, user accessibility to make his own status as busy, active states is also checked (this Dashboard is appeared at top-right corner of our web page, by pressing button we can enable ) Such one of our test got finished, if you find any one of the above operation not happening Intime correctly then following tests should be done.

**Step1:** First test is considered on Restful API working, here the lag to load the page and synchronising the chat history are checked, if we find any delay in those portions then immediately corresponding code will be fixed, and will ensure working of API is good, this is one of our test.

**Step 2:** In the next test, the performance of the database is analysed. Meanwhile how data of participant is saved and the numbering sequence of the user is checked, if all the database operations seems fine then this test got passed.

**Step3:** GUI functionality about anchor tagging in our code are properly checked and if any irrelevant linking occurs then they need to be fixed, this is one of our test.

**Expected result:** The user must be able to configure his account and perform few operations like selecting other users, changing chat status (Available, offline, busy), logout etc.

**Result:** Success

**Test: F-T8**

**Purpose**: Send and receive binary files

**Requirements:** UR20,SR1,SR2

**Environment:**

In this project HTML, CSS & JAVASCRIPT are used at front end to design user account. In this phase all the user desired operations can be done when it successfully integrated with Database and Server, and for unit testing we chose XAMPP framework to test and verify our code.

**Operation:** It can be operated by typing the file path of localhost in the browser i.e., http://localhost/filepath.php

**Steps to send or receive the messages:**

**Step1:** Underneath message send button you can find attach a file button, by pressing that button it will roots us to the directories in our computer to choose our file choice and by pressing send button we can send the file this is one of the test. If you got any difficulty to perform this operation then we need to proceed to following steps to handle the bugs.

**Step 2:** Check the file size restrictions (In our project we have restricted the document size to 2MB, if your file is above 2MB document will not be sent and will appear an error message). Total code should be rechecked before proceeding further, this is one of the test.

**Expected result:** Binary files must be sent and received from one user to other

**Result:** success.

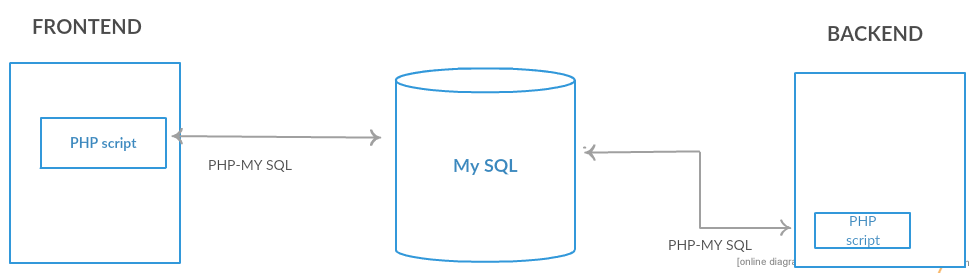
**Comments:** Functionality in progress

**4.Module 2- Database Module:**

This design describes the tables in the database and the relationship between the tables in the database and the database design.

In this project MySQL database was used.

**Design:**



**Fig 2**: database design

Database designing is the heart of any project, it is the enabler of any feature that we want to incorporate in our system, developers spent most of their time in database designing.

* In this project also lots of time is spent to make the database architecture.
* The main reason why we spent that much time is, as shown in the above figure both front end and backend operations require database services. Here, all the data is centrally stored and it automatically updates with the server.
* Admin functionalities are mainly dependent on the database ( delete an user, changing user details, adding users, activity of users etc are monitored using database of our system).

The structure of database can also be found below:

**Users:** In users table contains the users details such as id, username, email, password, ip, time and activation status.

Id - primary key

Email-unique key

**TABLE1: Online users:** In online users table contains the users online status this table consists of id, user id, ip, check in, status, checkout.

Id – primary key

User id – foreign key reference (users.id).

**TABLE2: Chat:** chat table consists of the chat details such as id, user1, user2, time, ip.

Id – primary key

User1, user2 – foreign key reference(users.id).

**Message:** message table consists of the messages of the users it consists of id, chat \_id, user \_id, message, type, size.

Id -primary key

User \_id - foreign key reference (users.id)

Chat \_id – foreign key reference (chat.id)

**TABLE 3: Broadcast Chat:** chat table consists of the chat details such as id, user1, user2, time, Ip.

Id – primary key

User1, user2 – foreign key reference(users.id).

**TABLE 4: Broadcast Message:** message table consists of the messages of the users it consists of id, chat \_id, user \_id, message, type, size.

Id -primary key

User \_id - foreign key reference (users.id)

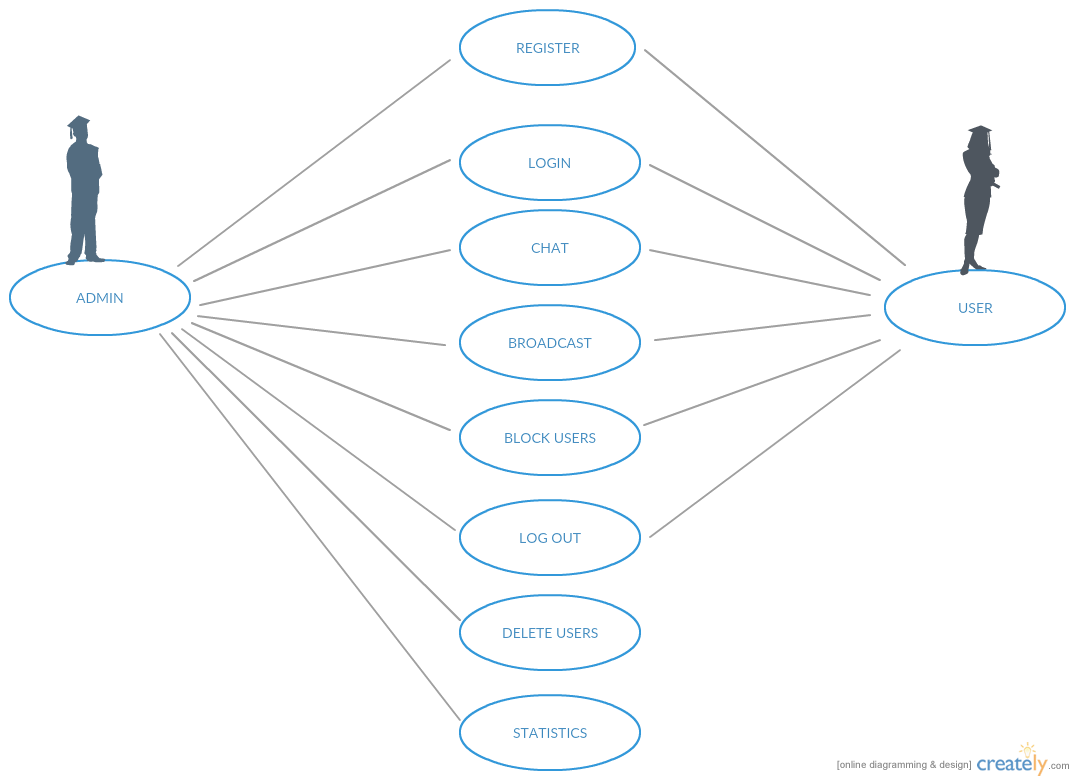
Chat \_id – foreign key reference (Broadcast chat.id) .

**5. Module 3: Backend Module**

5.1 Administrator (ADMIN)

Administrator is the employee who is appointed by the owner of the chat system license. Administrator have some rights (Authorisations) to change settings, create new accounts, access statistics of the chat system etc. But the supreme power is in the hands of owner (In our case our client is the owner).

Now let us see what are all the things that an administrator can do in the backend scripting. We have explained It in the following table.



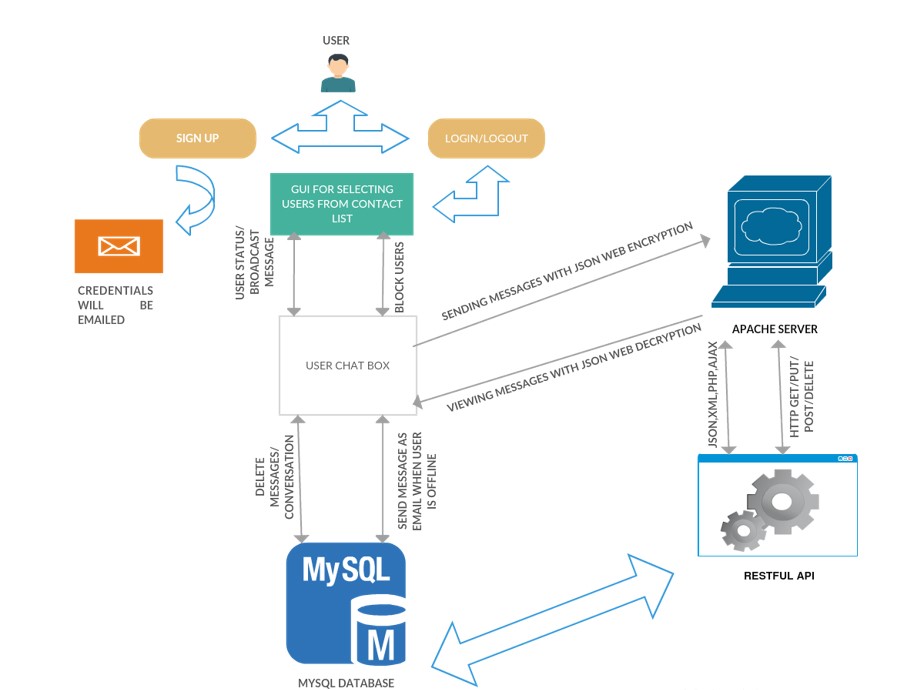
**Figure 3**: Use-case Module

|  |  |
| --- | --- |
| Work | Description |
| Admin Registration | Admin Registration is done by the owner. Once this phase is completed each Administrator is given by an Admin user name and password to get into the database or to serve the customers in other ways. |
| Admin Login | In this project, we don’t have the integrated link along with the user login (as **login as administrator)** but administrator is given by another link from which he can login and can do the corresponding changes in the system. |
| Chat | Administrator can chat with the users especially while troubleshooting or in emergency (especially cyber-attack to the website, post service denied information’s etc.). |
| Block other users | Administrator can block the users from the service. This action need some evidences about the user deeds (Misuse of services). |
| Delete users | Administrator can delete the users entirely from the system, this action too requires some evidences regarding misuse of the chat system. |
| User statistics | statistics and performance records of the system must be given to the administrator such they can think about the betterment of the system and do |
|  | estimates what are all the new things need to install in the system such they can serve all the customers without any interruptions. |
| Emergency Broadcast message | For the administrator We have provided an option to send an emergency message to all users at once in email in case of any failure of server or chat functionalities etc.  Requirements: SR1,SR2 |

**5.2 User Functionality:**

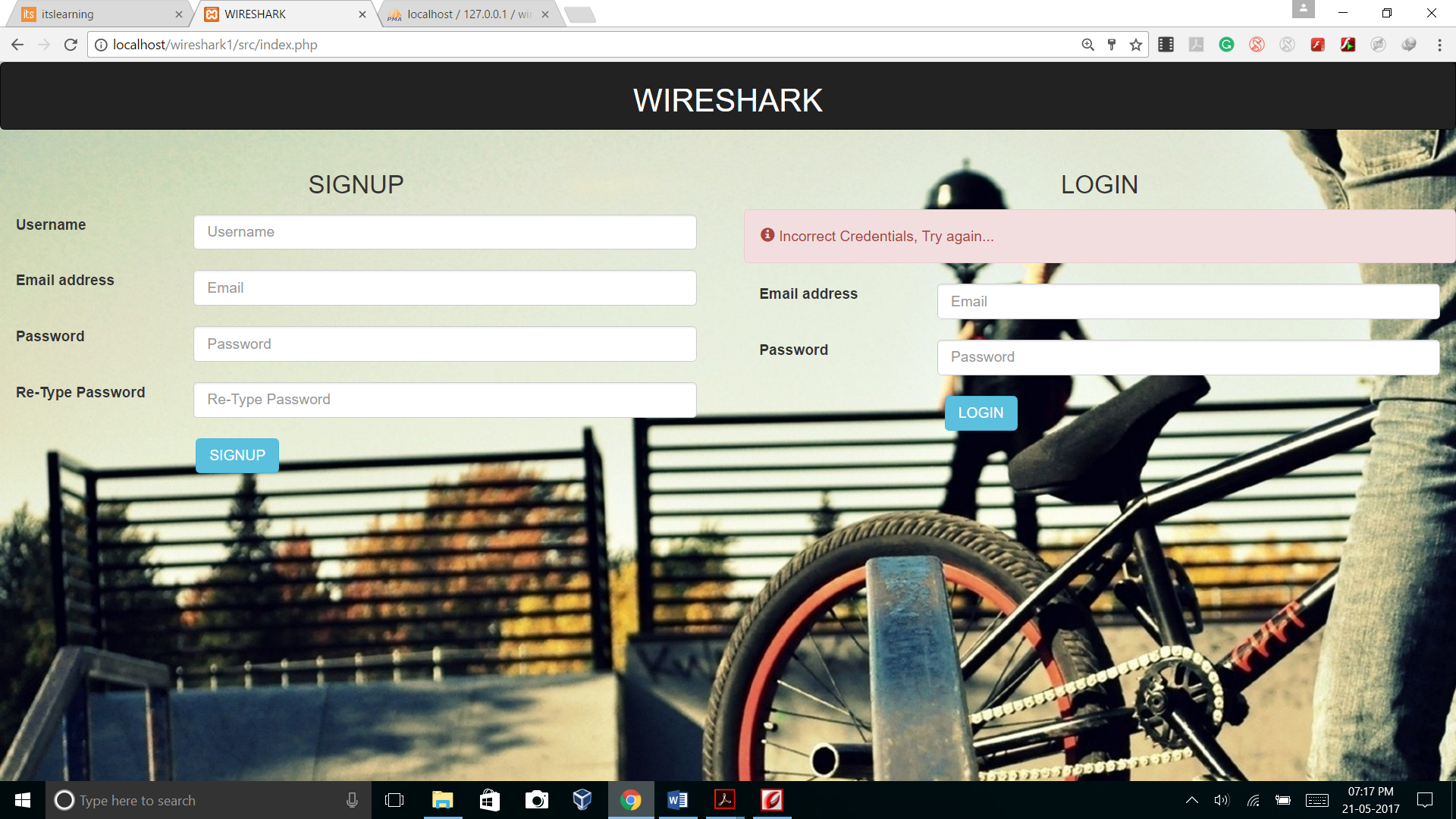
This module interacts with server and database using an API and also it needs to provide login and other functionalities to the users. The functionalities like sending and receiving messages (to view a user status (offline, online or busy))etc must be available.

The following are the requirements for a unit test plan

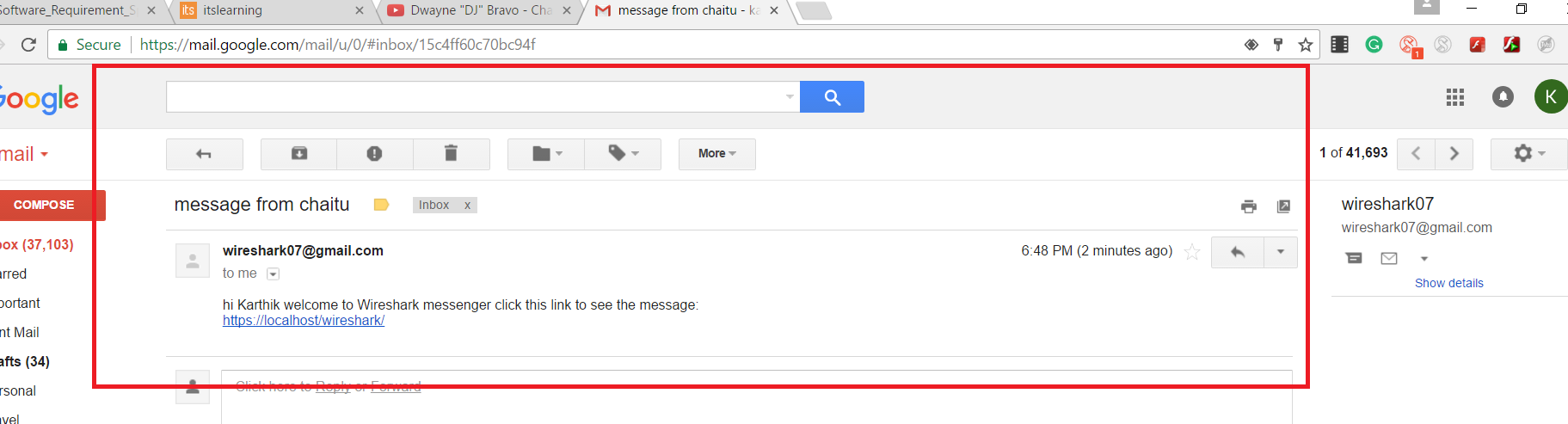


**Fig 4**: User functionality

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| UR1 | User authentication | SR1 and  SR2 | To install XAMPP server go to apache website and download XAMPP Package. Run the package and launch the XAMPP installer. XAMPP software must be installed from https://www.apachefr iends.org and necessary code in PHP must be written in an editor and saved. Make sure whether apache server and MYSQL database is running before accessing server which can be done by control the panel provided by software.  Required databases and tables are created and connected using scripts and are configured using  URL  http:///localhost  /phpmyadmin(NOTE: This project is made for the local users who are connected to switch so by using internet and by following this link will not let you go to this application) | When the user enters his login credentials in the login form, His credentials needed to be authenticated with the credentials stored in the database. |



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| UR2 | Email-  offline Users | SR1, SR2 | To install XAMPP server go to apache website and download XAMPP Package. Run the package and launch the XAMPP installer. XAMPP software must be installed from https://www.apachefr iends.org and necessary code in PHP must be written in an editor and saved. Make sure whether apache server and MYSQL database is running before accessing server which can be done by control the panel provided by software.  Required databases and tables are created and connected using scripts and are configured using  URL  http:///localhost  /phpmyadmin | Whenever a user sends a message to another user, the message must be deliverable through email if the user is in an offline state. |



**Test: B-T13**

**Purpose**: View all users registered/available users from database

**Requirements:** UR10, SR1, SR2

**Environment:** XAMPP software must be installed from https://www.apachefriends.org and necessary code in PHP must be written in an editor and saved. Make sure whether apache server and MYSQL the database is running before accessing server which can be done by control the panel provided by software. Required databases and tables are created and connected using scripts and are configured using URL <http:///localhost/phpmyadmin>(NOTE: This project is made for the local users who are connected to switch so by using internet and by following this link will not let you go to this application, it is exclusively created for the users who are connected to the LAN by means of a switch)

**Operation:** It can be operated by typing the file path of localhost in the browser i.e., http://localhost/filepath.php

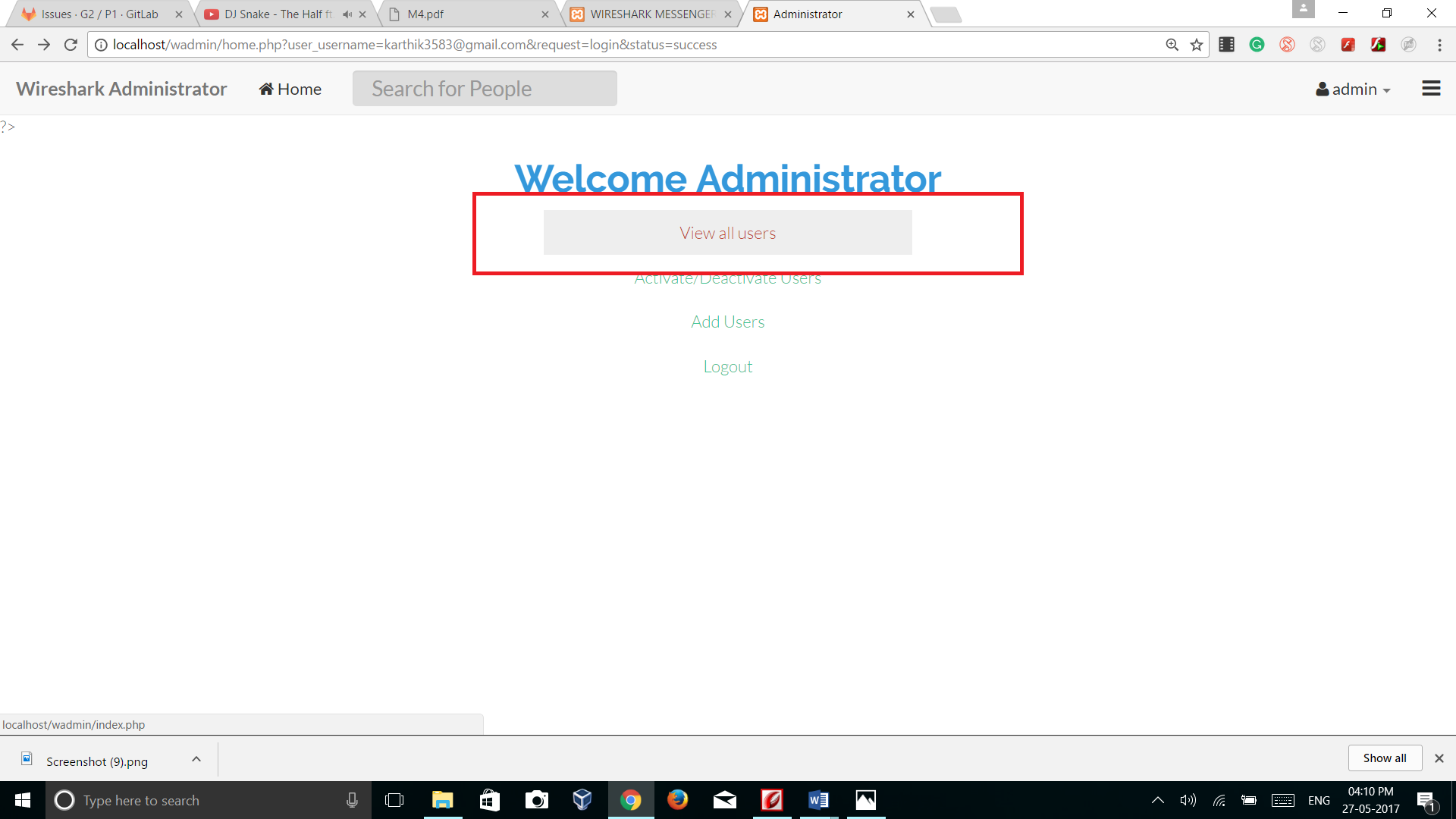
**Step 1:**Enter the login credentials of the admin and get access as you are a admin and you do have several more access to control the chat system. Here the test is weather you can login as admin or not.

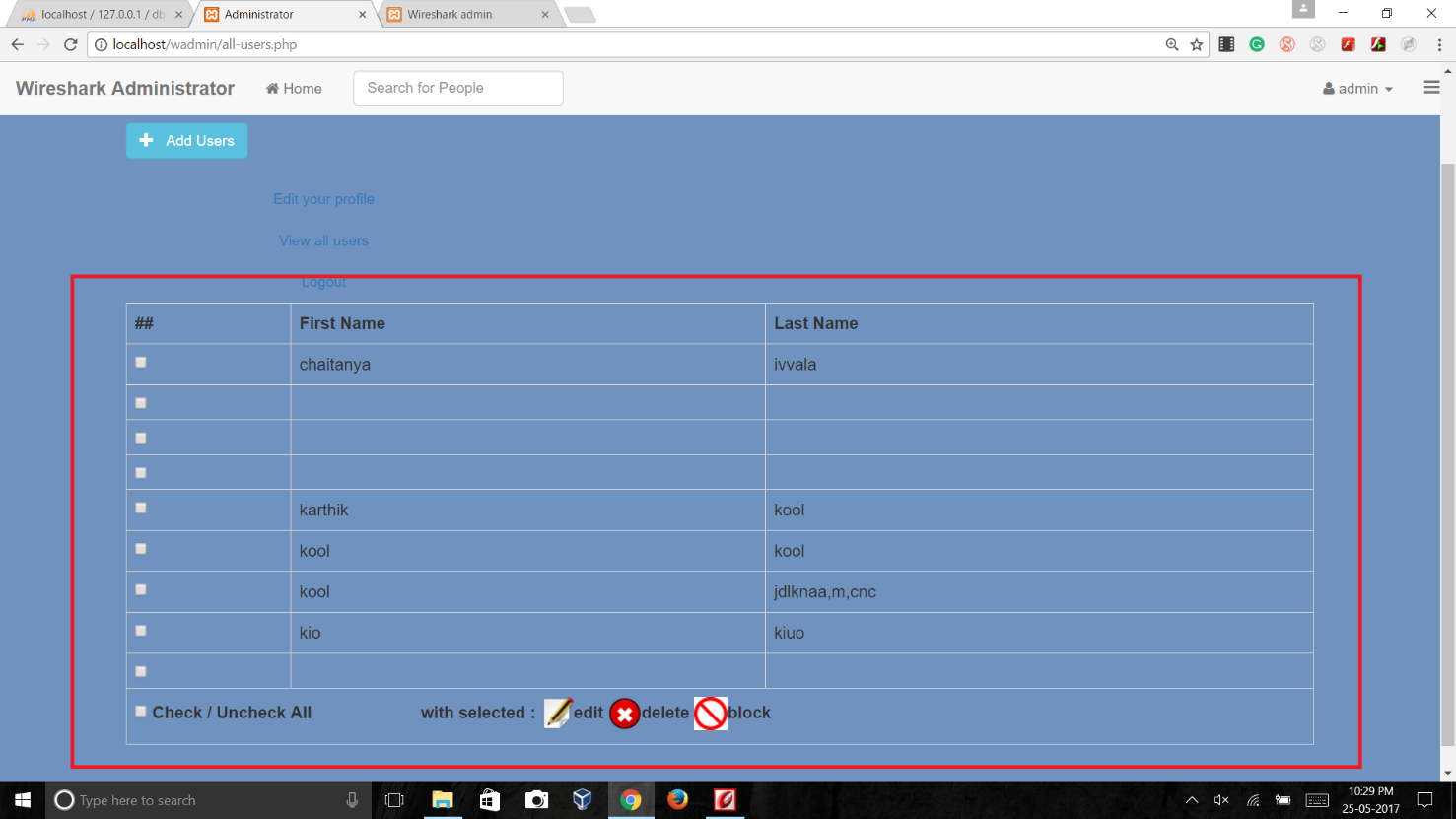
**Step 2:** If you are new to company and have posted as admin, then manager will send you a link to login as a admin and an exclusive password will be given to you, here testing part is link expiring when it confirms its recipient , this is one of the test.

**Expected result:** Admin must be able to view all users from database.

**Result:** Success

**Comments:** Admin can view all users successfully.





**Test: B-T14**

**Purpose**: To Add New Users directly from admin panel

**Requirements:** UR25,SR1,SR2

**Environment:** XAMPP software must be installed from https://www.apachefriends.org and necessary code in PHP must be written in an editor and saved. Make sure whether apache server and MYSQL the database is running before accessing server which can be done by control the panel provided by software. Required databases and tables are created and connected using scripts and are configured using URL <http:///localhost/phpmyadmin>(NOTE: This project is made for the local users who are connected to switch so by using internet and by following this link will not let you go to this application, it is exclusively created for the users who are connected to the LAN by means of a switch)

**Operation:** It can be operated by typing the file path of localhost in the browser i.e., http://localhost/filepath.php

Admin have the accessibility to add directly new members to the chat system from his panel, as shown in the following figure.

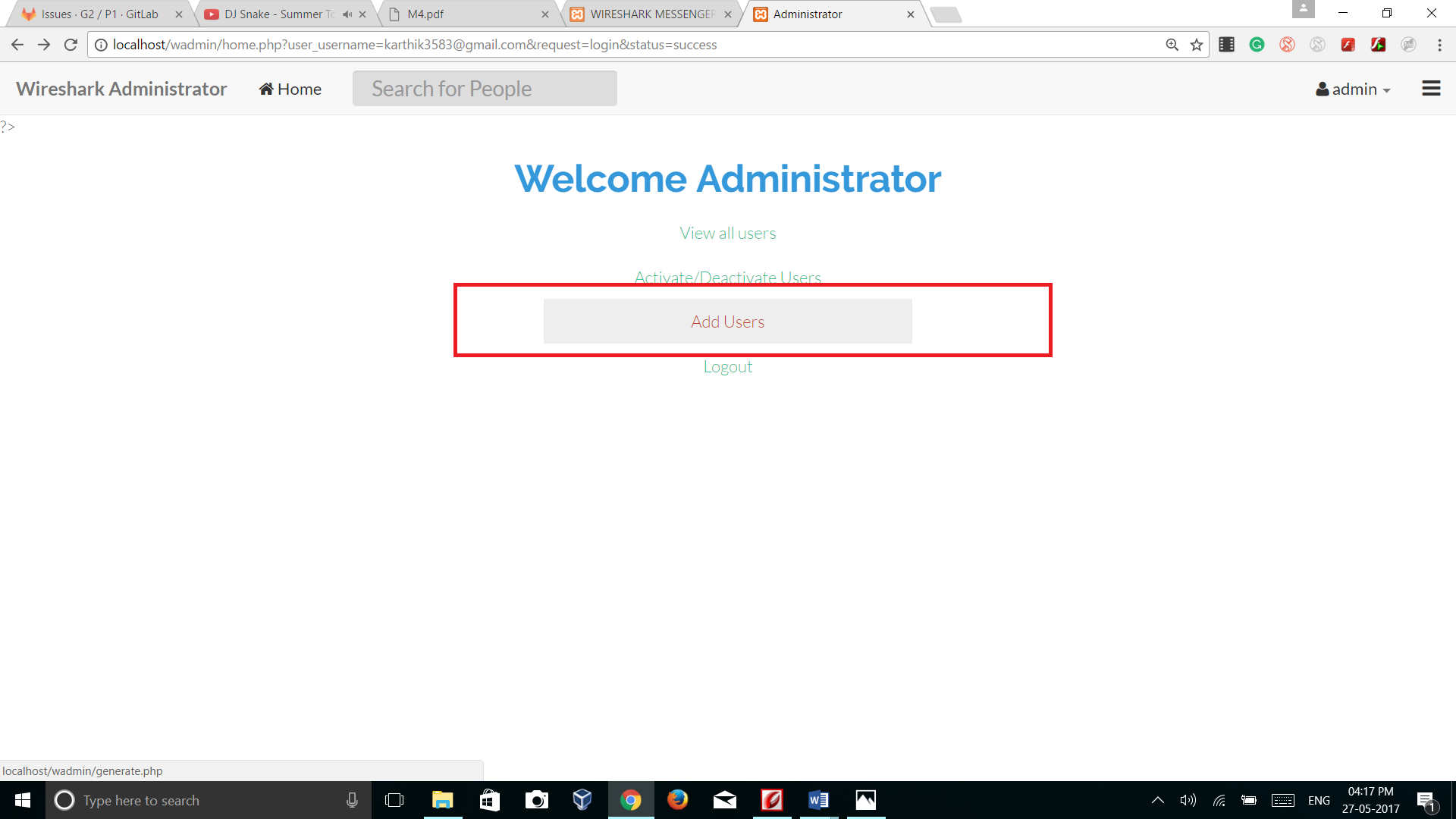
**Step 1:** at this phase the hyperlinking of buttons are checked perfectly, this is one of the test

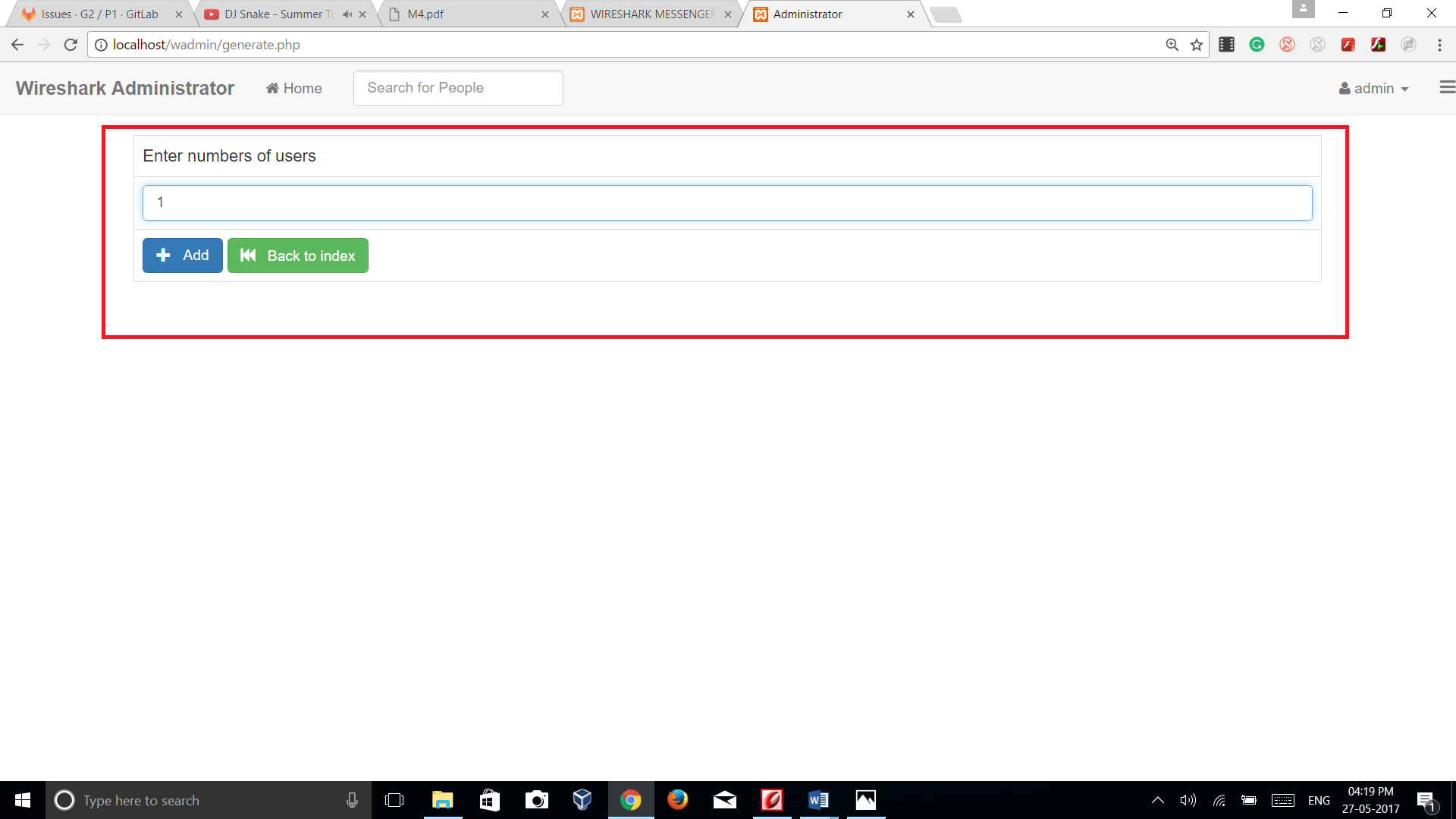
**Step 2:** In the next step it will ask for users first name and last name credentials, after filling them we need to check whether they are going to database correctly or not and need to check how server behaving for the changes , this is another test.

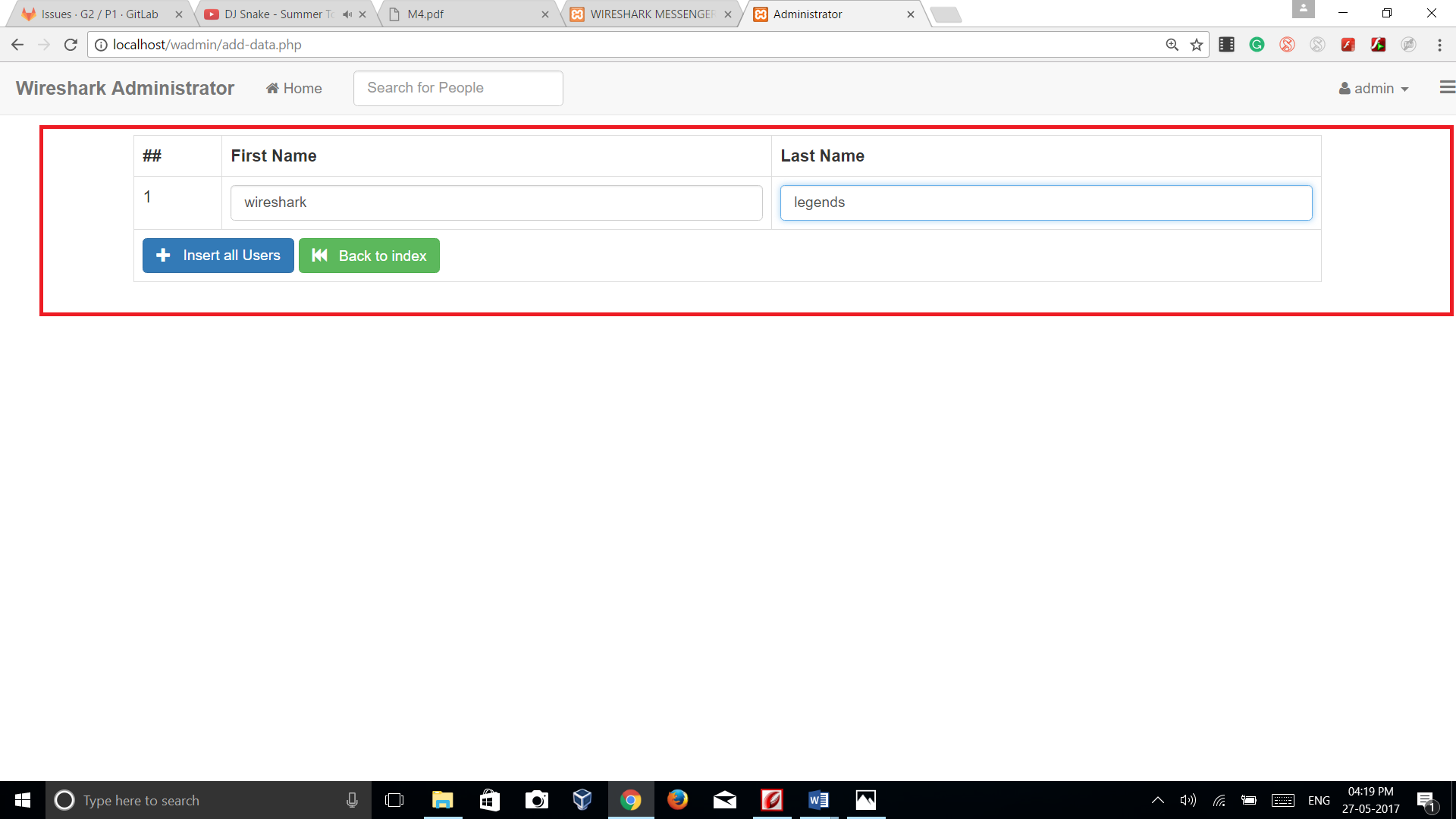
**Expected result:** Admin must be able to add new users from admin panel

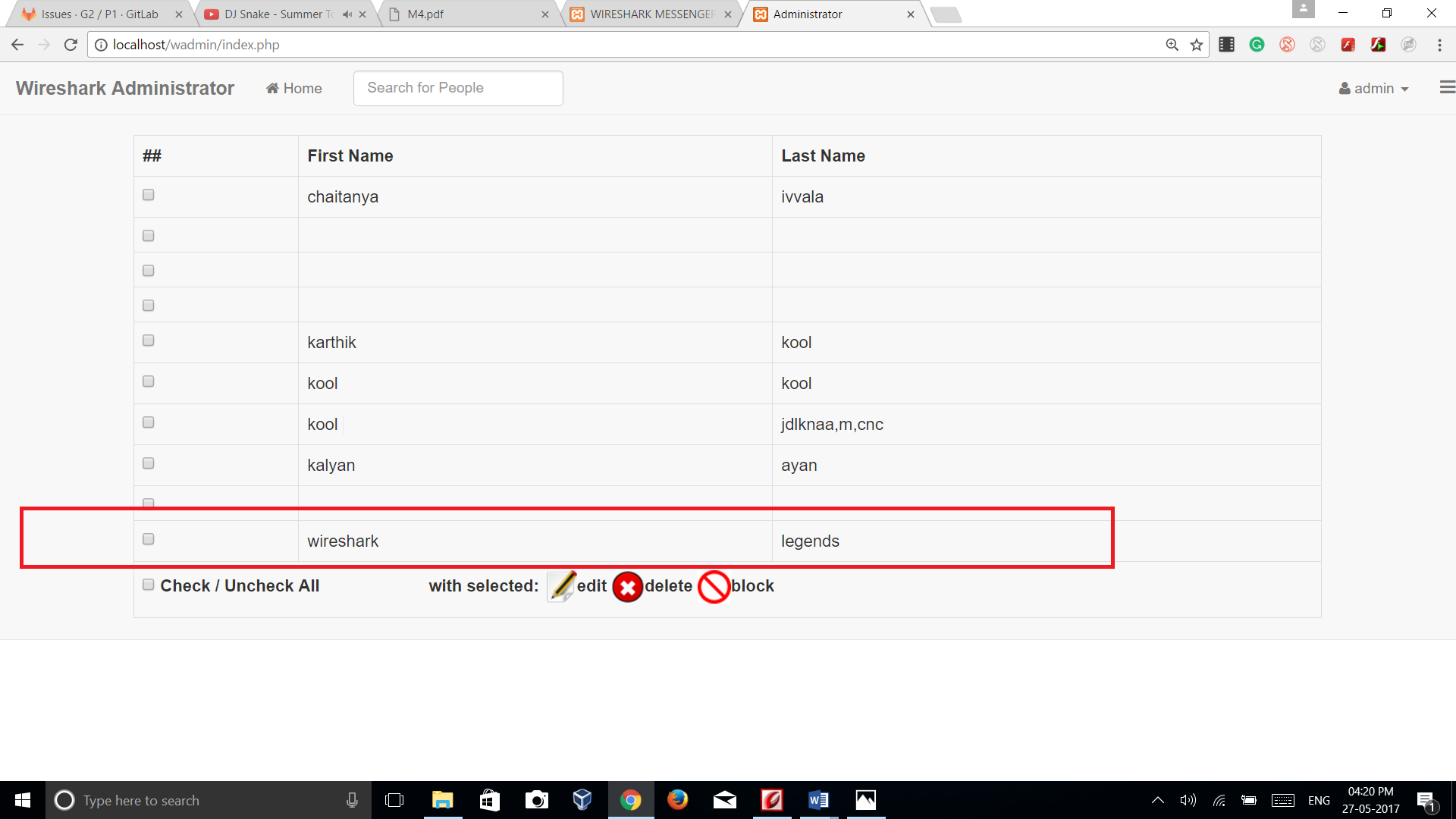
**Result:** Success

**Comments:** Admin can add new users directly from admin panel.









**Test: B-T15**

**Purpose**: To Edit users/users details

**Requirements:** UR11,SR1,SR2

**Environment:** XAMPP software must be installed from https://www.apachefriends.org and necessary code in PHP must be written in an editor and saved. Make sure whether apache server and MYSQL the database is running before accessing server which can be done by control the panel provided by software. Required databases and tables are created and connected using scripts and are configured using URL <http:///localhost/phpmyadmin>(NOTE: This project is made for the local users who are connected to switch so by using internet and by following this link will not let you go to this application, it is exclusively created for the users who are connected to the LAN by means of a switch)

**Operation:** It can be operated by typing the file path of localhost in the browser i.e., http://localhost/filepath.php

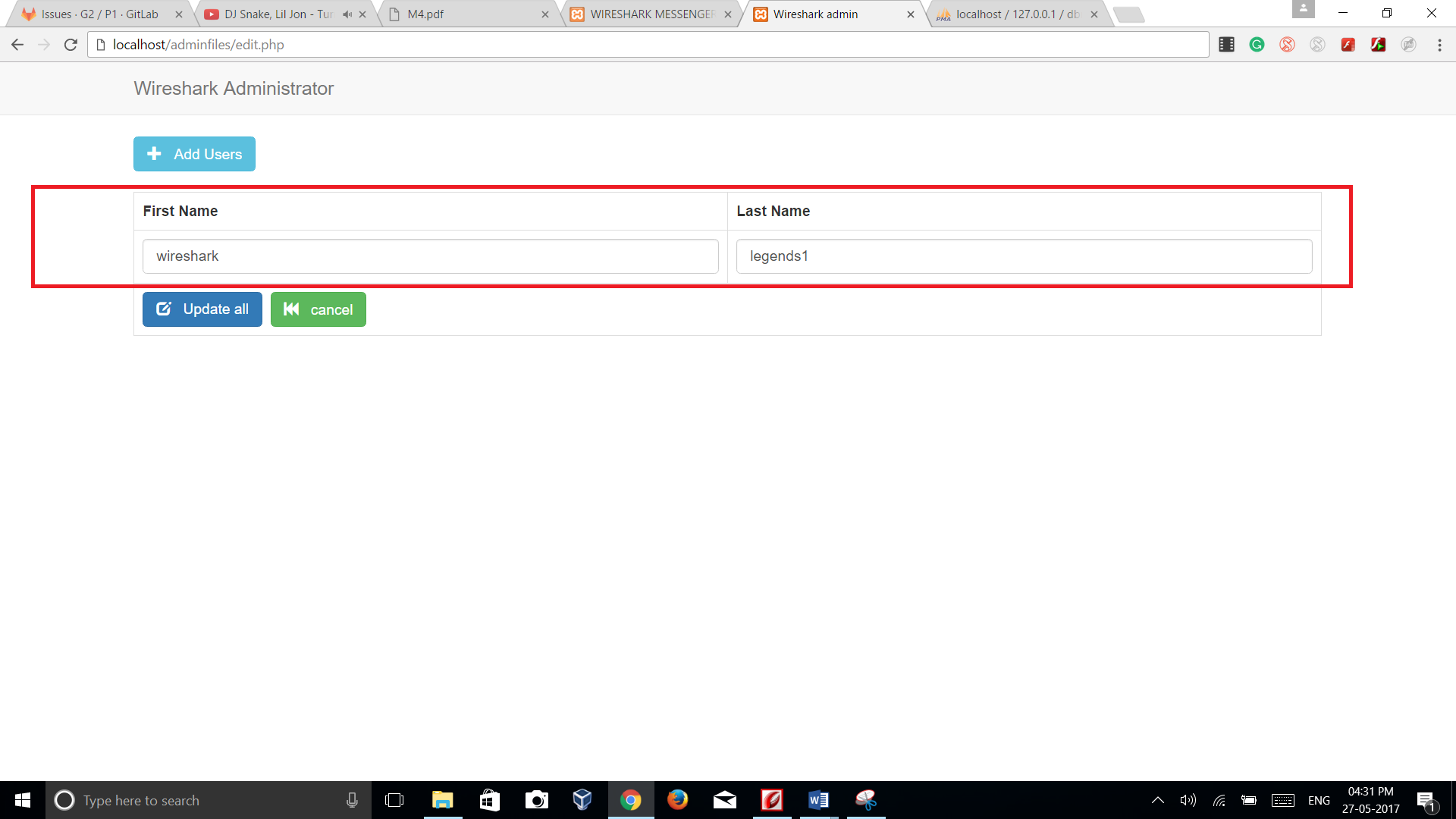
**Step 1:**To edit user details admin should be go to the database and need to search for serial number of the user and need to edit the user details, if all these things will not happen then you need to test for errors in the following manner.

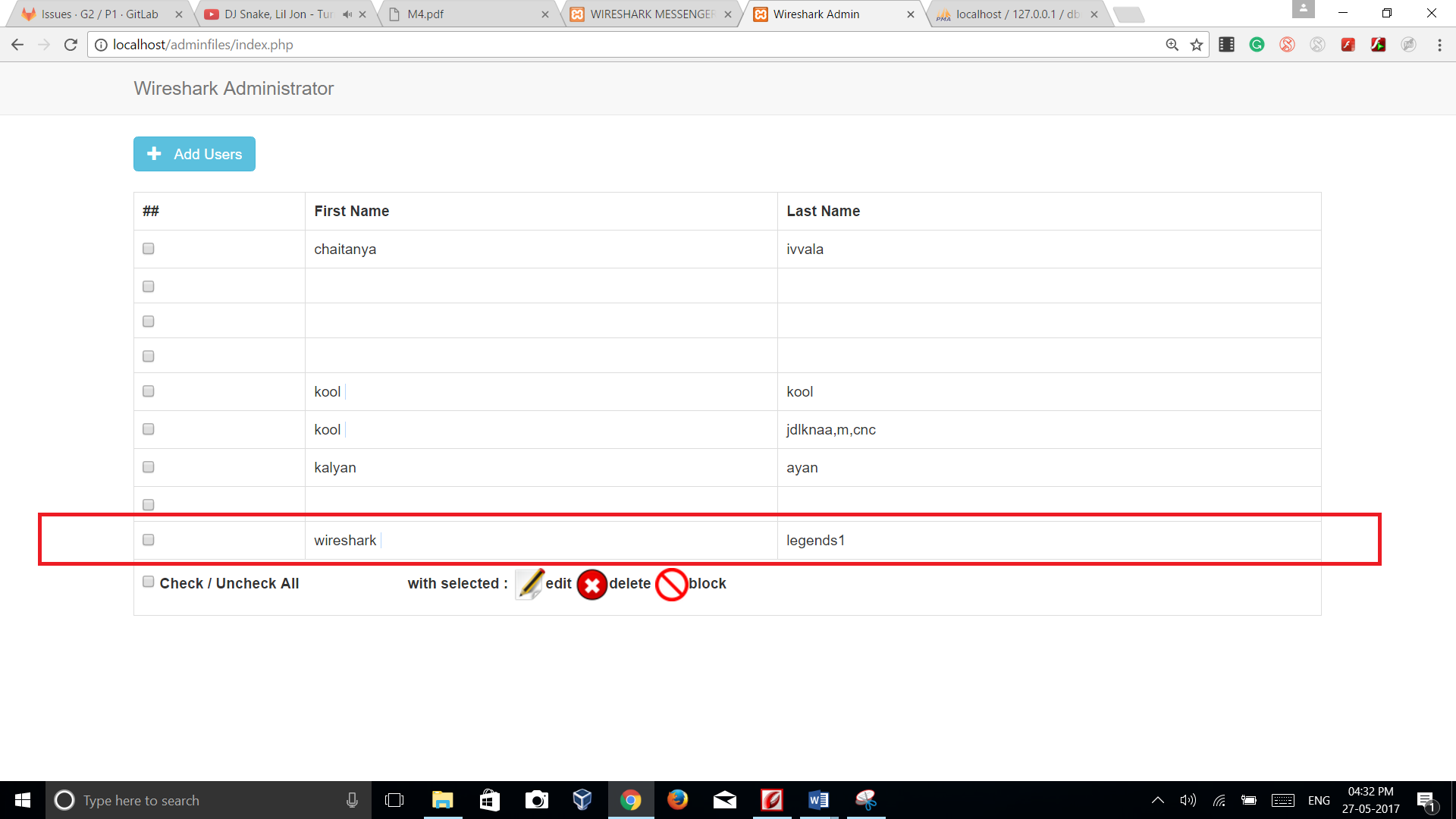
**Step 2:** After changing the name in the control panel, need to check our database whether the corresponding changes are made or not , if not then data base process id of the user must be changed.

**Expected result:** Admin must be able to edit user details

**Result:** Success

**Comments:** Admin can edit user details successfully





**Test: B-T16**

**Purpose**: To Delete users/users details

**Requirements:** UR11, SR1, SR2

**Environment:** XAMPP software must be installed from https://www.apachefriends.org and necessary code in PHP must be written in an editor and saved. Make sure whether apache server and MYSQL the database is running before accessing server which can be done by control the panel provided by software. Required databases and tables are created and connected using scripts and are configured using URL <http:///localhost/phpmyadmin>(NOTE: This project is made for the local users who are connected to switch so by using internet and by following this link will not let you go to this application, it is exclusively created for the users who are connected to the LAN by means of a switch)

**Operation:** It can be operated by typing the file path of localhost in the browser i.e., <http://localhost/filepath.php>

Steps to delete user/User details:

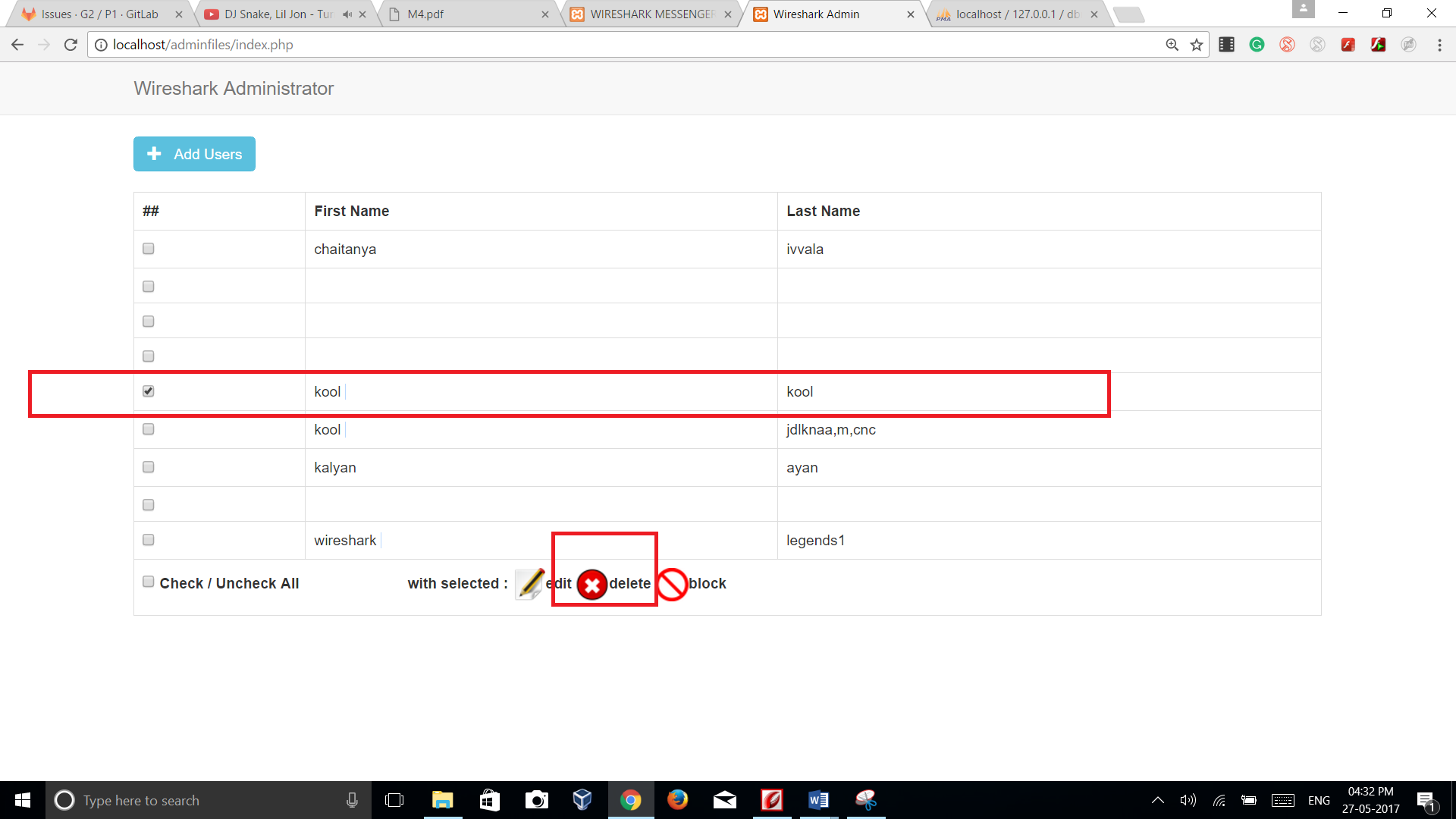
**Step 1:**To delete user details, admin should be go to the database and need to search for serial number of the user and need to verify deletion of the user, if all these things will not happen then you need to test for errors in the following manner.

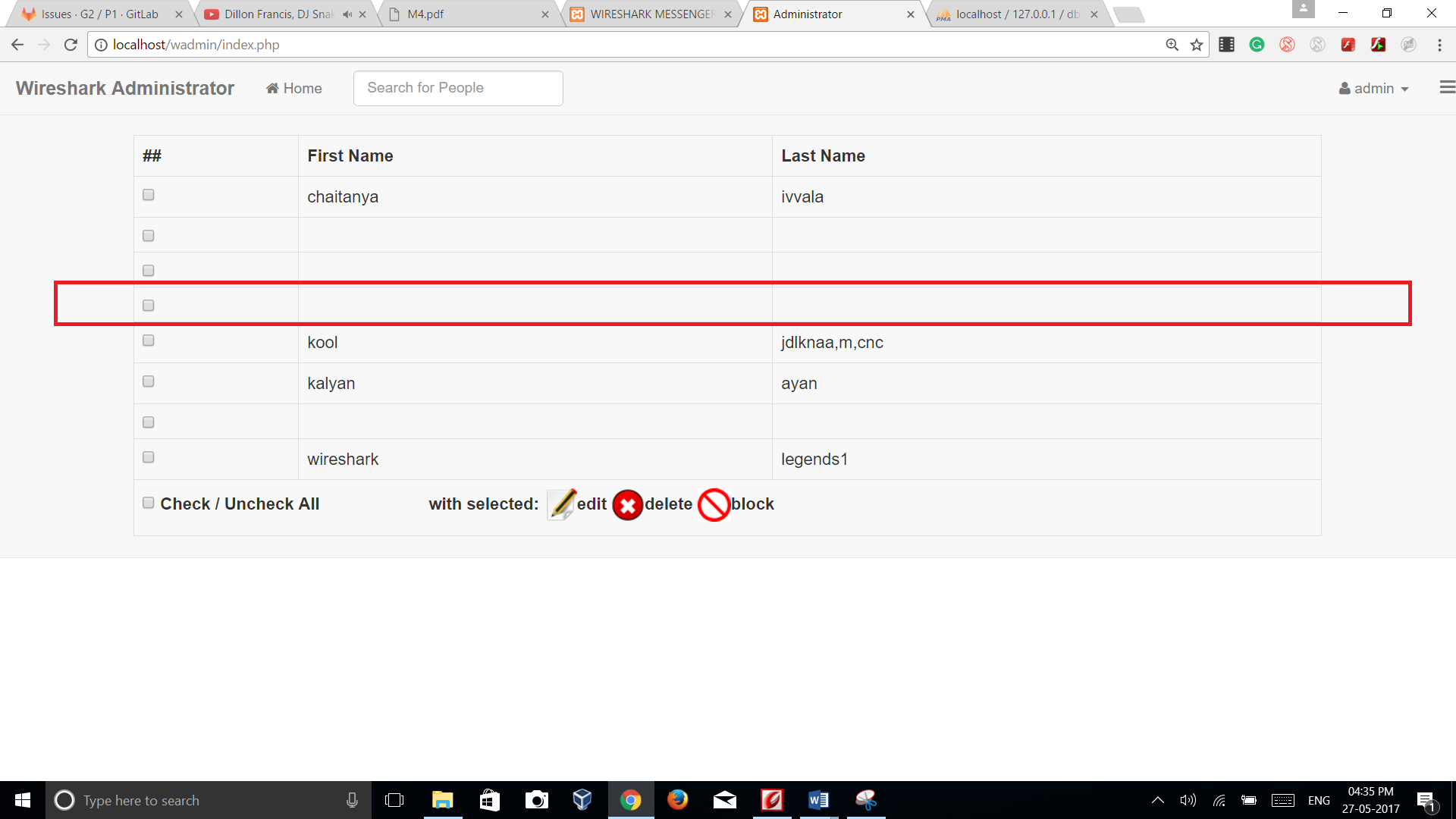
**Step 2:** After deleting a user in the control panel, need to check our database whether the corresponding changes are made or not , if not then data base process id of the user must be changed.

**Expected result:** Admin must be able to delete users

**Result:** Success

**Comments:** Admin can delete user/user details successfully





**Test: B-T17**

**Purpose**: To block/deactivate users successfully

**Requirements:** UR12, SR1, SR2

**Environment:** XAMPP software must be installed from https://www.apachefriends.org and necessary code in PHP must be written in an editor and saved. Make sure whether apache server and MYSQL the database is running before accessing server which can be done by control the panel provided by software. Required databases and tables are created and connected using scripts and are configured using URL <http:///localhost/phpmyadmin>(NOTE: This project is made for the local users who are connected to switch so by using internet and by following this link will not let you go to this application, it is exclusively created for the users who are connected to the LAN by means of a switch)

**Operation:** It can be operated by typing the file path of localhost in the browser i.e., <http://localhost/filepath.php>

Steps to deactivate the user:

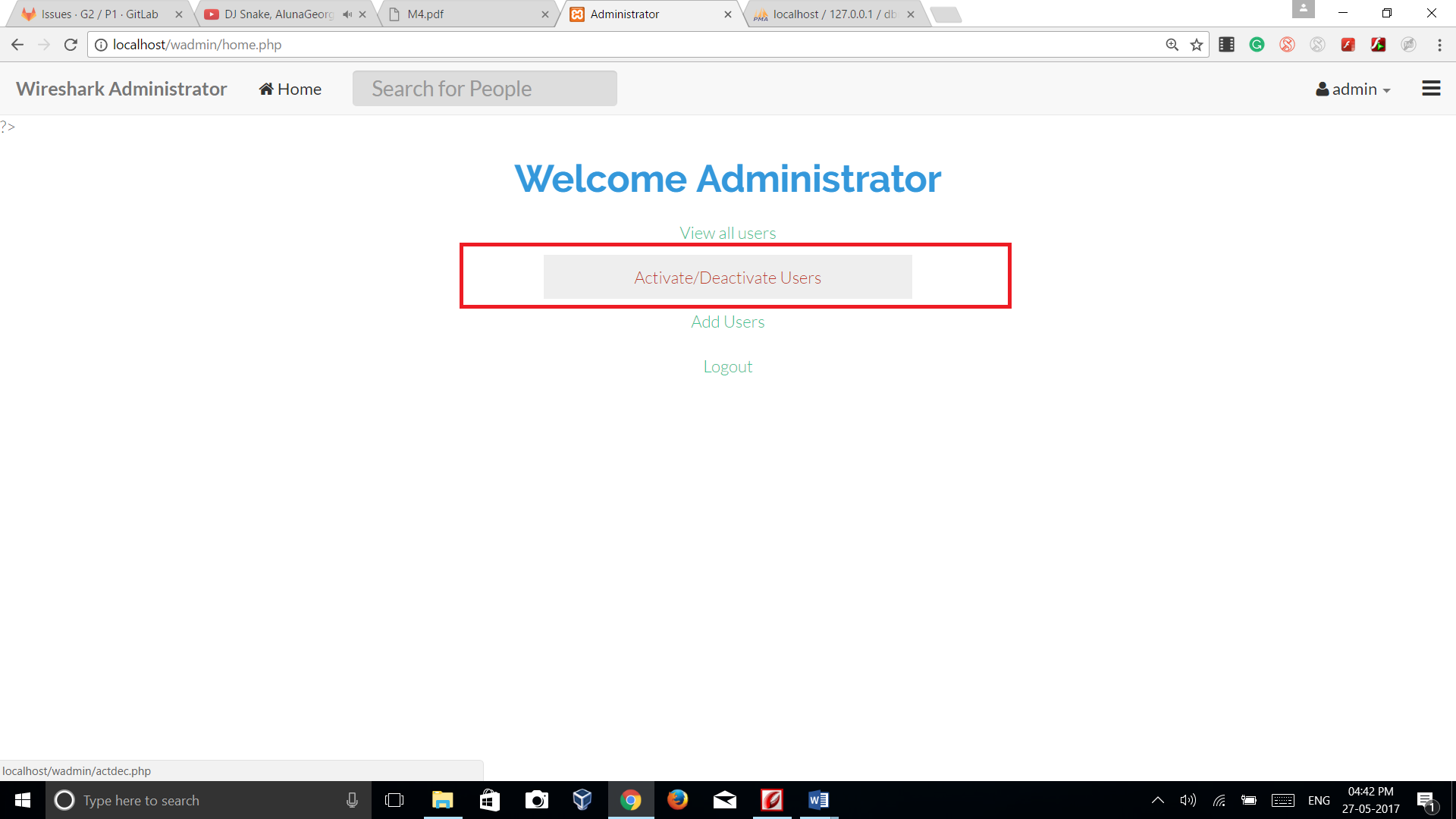
**Step 1:**To block an user, admin should be go to the database and need to search for serial number of the user and need to check blocking details of the user, if all these things will not happen then you need to test for errors in the following manner.

**Step 2:** After deleting the user in the control panel, admin need to check our database whether the corresponding changes are made or not , if not then data base process id of the user must be changed.

**Expected result:** Admin must be able to block/deactivate users.

**Result:** Success

**Comments:** Admin can block/deactivate users successfully.



**Test: B-T18**

**Purpose**: To unblock/activate users successfully

**Requirements:** UR14,SR1,SR2

**Environment:** XAMPP software must be installed from https://www.apachefriends.org and necessary code in PHP must be written in an editor and saved. Make sure whether apache server and MYSQL the database is running before accessing server which can be done by control the panel provided by software. Required databases and tables are created and connected using scripts and are configured using URL <http:///localhost/phpmyadmin>(NOTE: This project is made for the local users who are connected to switch so by using internet and by following this link will not let you go to this application, it is exclusively created for the users who are connected to the LAN by means of a switch)

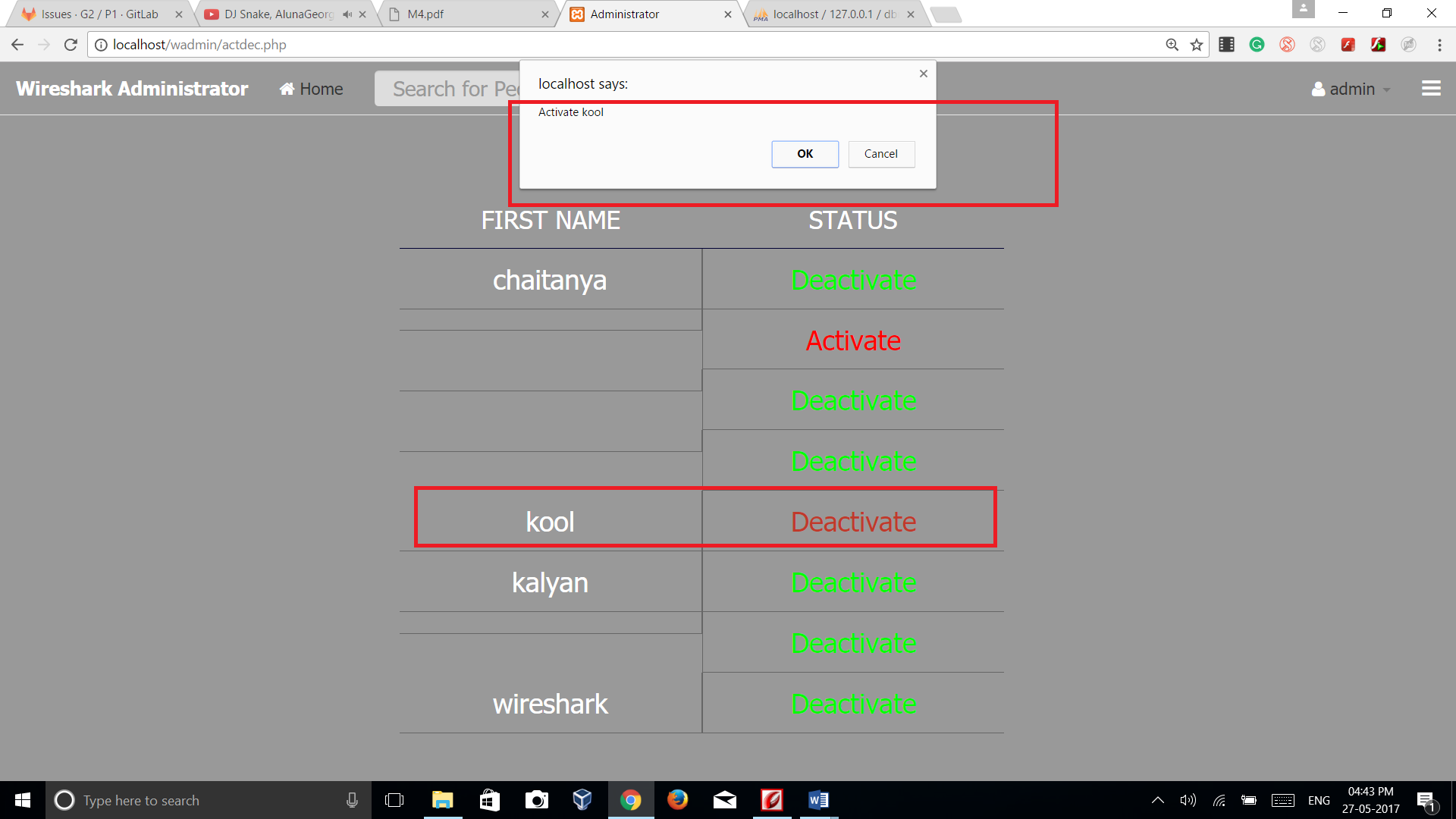
**Operation:** It can be operated by typing the file path of localhost in the browser i.e., <http://localhost/filepath.php>

* Admin have the accessibility to unblock an user
* **Testing:**After pressing to corresponding buttons, tester need to test the hyperlinking of that button to database, and server if the corresponding logic seems perfect then our test got passed.

**Expected result:** Admin must be able to unblock/activate users.

**Result:** Success

**Comments:** Admin can unblock/activate users successfully.



**Test: B-T19**

**Purpose**: Able to view the records of the system or data base and user statistics

**Requirements:** UR24, SR2,SR1

**Environment:** XAMPP software must be installed from https://www.apachefriends.org and necessary code in PHP must be written in an editor and saved. Make sure whether apache server and MYSQL the database is running before accessing server which can be done by control the panel provided by software. Required databases and tables are created and connected using scripts and are configured using URL <http:///localhost/phpmyadmin>(NOTE: This project is made for the local users who are connected to switch so by using internet and by following this link will not let you go to this application, it is exclusively created for the users who are connected to the LAN by means of a switch)

**Operation:** It can be operated by typing the file path of localhost in the browser i.e., http://localhost/filepath.php

**Expected result:** Admin must be able to see the database of the system and the statistics between the users. Admin must also be able to view user statistics in a tabular/graphical view.

**Result:** Failed

**Comments: Functionality in progress**

**Test: B-T20**

**Purpose**: Encryption of web server

**Requirements:** UR18,SR1,SR2

**Environment:** XAMPP software must be installed from https://www.apachefriends.org and necessary code in PHP must be written in an editor and saved. Make sure whether apache server and MYSQL the database is running before accessing server which can be done by control the panel provided by software. Required databases and tables are created and connected using scripts and are configured using URL <http:///localhost/phpmyadmin>(NOTE: This project is made for the local users who are connected to switch so by using internet and by following this link will not let you go to this application, it is exclusively created for the users who are connected to the LAN by means of a switch)

**Operation:** It can be operated by typing the path of localhost in the browser i.e., <https://localhost/>

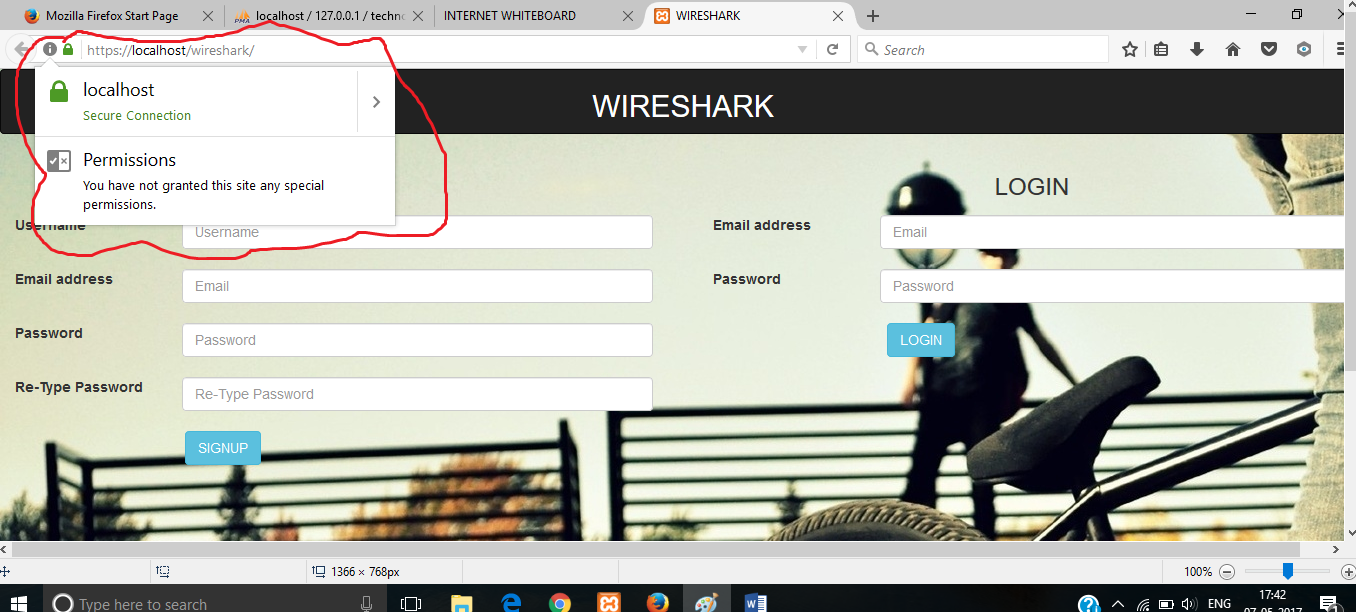
* By following to the above link we will be directed to the statistics page as shown in the following figure, where we can analyse each and every statistics like how many messages sent, online users , broadcast messages etc

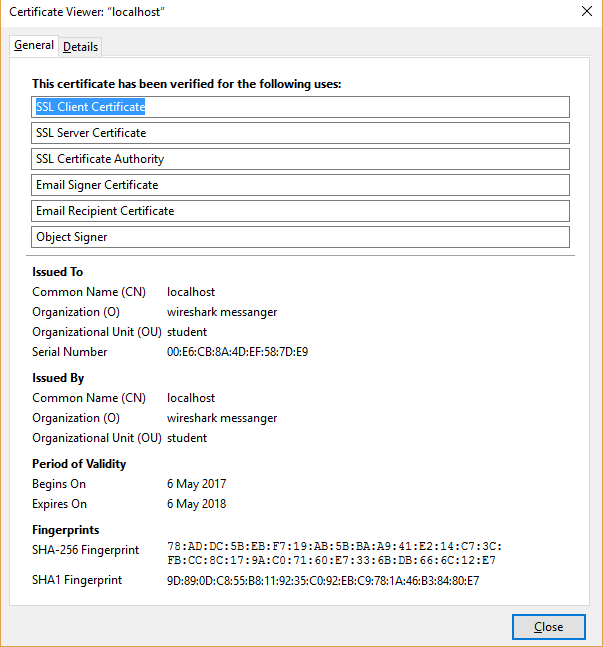
**Expected result:** The server must be setup with necessary certificates.

**Result:** Success

**Comments:** The server is set up with necessary certificates







**Test: B-T21**

**Purpose**: Testing RESTFUL API

**Requirements:** UR21,SR1,SR2,SR3

**Environment:** XAMPP software must be installed from https://www.apachefriends.org and necessary code in PHP must be written in an editor and saved. Make sure whether apache server and MYSQL the database is running before accessing server which can be done by control the panel provided by software. Required databases and tables are created and connected using scripts and are configured using URL <http:///localhost/phpmyadmin>(NOTE: This project is made for the local users who are connected to switch so by using internet and by following this link will not let you go to this application, it is exclusively created for the users who are connected to the LAN by means of a switch)

**Operation:** It can be operated by server run command ‘curl – I <http://localhost/example.php>’

* By using curl’s **PUT, GET, POST, DELETE** Commands, we can check the restful API

working.

**RESULT:** Success

**Comments:**

**Test: B-T22**

**Purpose**: Email validation

**Requirements:** UR23,SR1,SR2

**Environment:** XAMPP software must be installed from https://www.apachefriends.org and necessary code in PHP must be written in an editor and saved. Make sure whether apache server and MYSQL the database is running before accessing server which can be done by control the panel provided by software. Required databases and tables are created and connected using scripts and are configured using URL <http:///localhost/phpmyadmin>(NOTE: This project is made for the local users who are connected to switch so by using internet and by following this link will not let you go to this application, it is exclusively created for the users who are connected to the LAN by means of a switch)

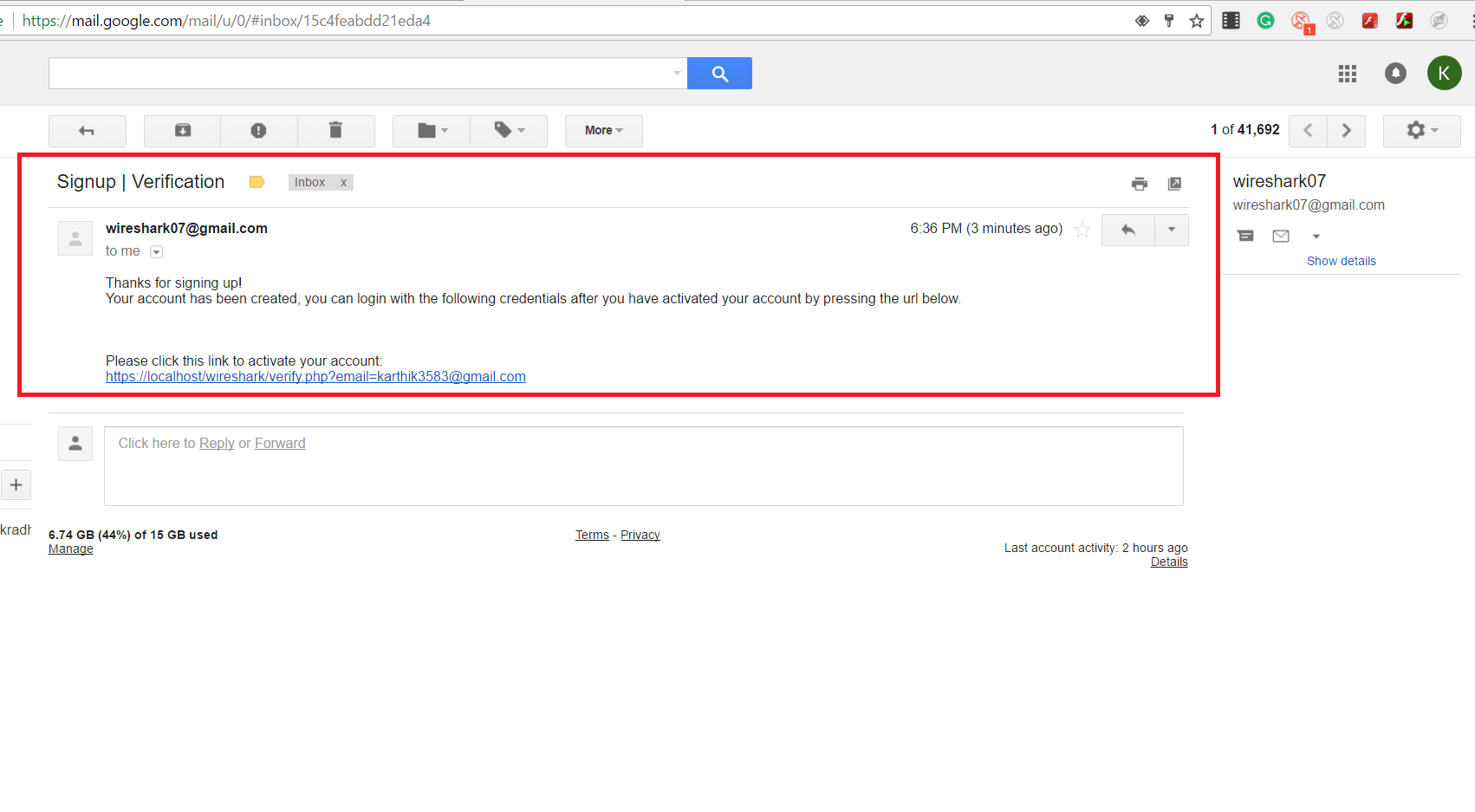
**Operation:** It can be operated by typing the path of registration page in the browser i.e., <https://localhost/index.php>

* During sign up process a verification mail will be sent and by following that link activation will takes place.

**Expected result:** User is validated through email

**Result**: Success

**Comments**: User needs to click on the activation link for succesfull activation of the account.



**References:**

[1]. Ian Sommerville. Software Engineering. 9th ed.