ONLINE CHATTING SYSTEM

A PROJECT REPORT

Submitted by

RAJ SANGAVI P	**(1513111)**
UDHAYAPRIYA M	**(1513147)**
VIGNESHWARAN R	**(1513156)**
VIJAY M	**(1513157)**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

K.S.R. COLLEGE OF ENGINEERING

(An Autonomous Institution & affiliated to Anna University, Chennai and Approved by AICTE)

TIRUCHENGODE - 637215

DECEMBER 2018

K.S.R. COLLEGE OF ENGINEERING

TIRUCHENGODE

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **“ONLINE CHATTING SYSTEM ”** is the bonafide work of **”RAJ SANGAVI R (1513111) , UDHAYAPRIYA M (1513147), VIGNESHWARAN R (1513156) , VIJAY M (1513157)”** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr. A. RAJIV KANNAN M.E., Ph.D.,	**Dr. C. ANAND M.E., Ph.D.,**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
Professor	**Associate Professor**
Department of CSE,	Department of CSE,
K.S.R. College of Engineering,	K.S.R. College of Engineering,
Tiruchengode-637215.	Tiruchengode-637215.

Submitted for the project viva – voce held on …………………..

Internal Examiner External Examiner

ACKNOWLEDGEMENT

We feel highly honored to extend our sincere gratitude to our beloved Founder cum Chairman **Lion Dr. K. S. RANGASAMY MJF.,** K.S.R Educational Institutions and our Chairman cum Managing Trustee **Mr. R. SRINIVASAN BBM., MISTE.,** for providing all facilities to complete this project work.

We would like to acknowledge the constant and kind support provided by our principal **Dr. P. SENTHILKUMAR M.E., Ph.D.(IITM),** who supported us in all the endeavors and been responsible for inculcating us all through our career.

We feel highly elated to thank our respectable Head of the Department **Dr. A. RAJIV KANNAN M.E., Ph.D., MISTE., MCSI.,** who guided us and was a pillar of support for the successful completion of the project.

We are thankful to our Project Coordinators **Dr. A. VISWANATHAN M.E., Ph.D.,** and **Mr. G. NAGARAJAN** of our department for their valuable suggestions and guidance to our project.

We are the most fortunate in having the opportunity to work under the guide **Dr. C. ANAND M.E., Ph.D.,** express our sincere thanks to him. This project has brought out the hidden talent within us.

It is a pleasure to express our gratefulness to our beloved parents for providing their support and confidence to us for the completion of the project and our heartfelt thanks to our entire department faculty members, beloved friends, directly and indirectly who helped us during the tenure of the project.

ABSTRACT

Chatting is a method of using technology to bring people have authenticate user details like username and password. If this both details is proper then he can get access into this online chat system. The technology has been available for years but the acceptance it was quit recent. Our project is an example of a chat server. It is made up of 2 applications the client applications, which runs on the user Android Device and server application, which runs on any Android Device on the network. To start chatting client should get connected to server where there can do private and group chat security measures were taken during the last one. The purpose of the chat application is to allow users be able to the chat with each other, like a normal chat application. The users will be able to chat with each other, most likely only from user to user, no group chatting will be developed, unless there is time to do so. The chat application will be written in PHP, but due to the lack of experience in PHP, while developing the application, practicing techniques with java and working on it as much as possible will help hone some java skills and be more ready to develop the application. For the scope of the project, the project will be tested as the program is being developed.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACRONYM	ABBREVIATION
AJAX	Asynchronous JavaScript and XML
CSS	Cascading Style Sheet
DFD	Data Flow Diagram
HTML	Hyper Text Markup Language
JS	Java Script
OS	Operating System
PHP	Hypertext Processor
RAM	Random Access Memory
SQL	Structured Query Language
UI	User Interface
URL	Uniform Resource Locator
UML	Unified Modeling Language

CHAPTER 1

INTRODUCTION

1.1 OBJECTIVE OF THE PROJECT

To gather information on approaches currently taken on the use of Chat in online to communicate with their friends. Broadcasting Chat Server Application is going to be a text communication software, it will be able to communicate between two computers using point to point communication. The limitation of Live Chat it is does not support audio conversations. To overcome the limitations we are concurrently working on developing better technologies. Companies would like to have a communication software wherein they communicate instantly within their organization.

User Registeration

User must be able to register for the application through a valid id. On installing the application user must be prompted to register their user. If user skips these steps application should close. The users ID will be the unique identifier of his / her account on chat application.

Adding new User ID

The application should detect all ID from the user servers. If any of the ID have user accounts with Chat Application, those ID must automatically be added to the users servers list on Chat Application. If any any of the user ID not yet registered on Chat Application, user should be provided with an invite option that send those ID a regular text message asking them to join Chat Application along with a link to the Chat Application.

Send Message

User should be able to send instant message to any ID on his/her Chat Application contact ID. User should be notified when message is successfully delivered to the recipient by displaying a tick sign next to the message sent.

Broadcast Message

User should be able to create group of contacts. User should be able to boardcast message to these groups.

Message Status

User must be able to get information on whether the message sent has been read by the intended recipient. If recipient reads the message 2 ticks must appear to the message read.

CHAPTER 2

SYSTEM STUDY

2.1 EXISTING SYSTEM

The main objective of the Online Chat Application is to manage the details of Chat Application,Online Chat,Smiles Chat,Users,Chat History. It manages all the information about Chat Application, Chat Profile, Chat History, Chat Application. The project is totally built at administrative end and thus only the administrator is guaranteed the access. The purpose of the project is to build an application program to reduce the manual work for managing the Chat Application, Online Chat, Chat Profile, Smiles Chat. It tracks all the details about the Smiles Chat,Users,Chat History.

2.1.1 DISADVANTAGES

- Avaliable in iPhone, Android and Windows Phone.
- send unlimited messages across the world using Application (Example: Whatsapp)
- send audio , video, messages and document files upto size 100Mb (Stored on phone storage)
- Download it on Appstores.
- In a group, the size limit of the group is <= 256 persons currently
- Storing/Retrieving the Chat - Backup may be messy and takes time.

2.2 PROPOSED SYSTEM

User the Webhost link through the browser. It can send audio , video ,messages and document files upto size more than 500Mb (stored on cloud server). It provides a group chat with a unlimit of members.Automatically Backup message it stored on cloud server. It can send more than 500 MB (Mega Byte).

2.2.1 ADVANTAGES

- Available in iPhone,Android Windows Phone,Desktop and Laptop.
- send unlimited messages across the world using Web Link (Example: gmail)
- send audio , video ,messages and document files upto size more than 500Mb (Stored on Cloud server)
- User the Webhost link through the browser.
- In a group, the size limit of the group is 600 persons currently
- Automatically Backup messages it stored on Cloud Server it takes more MB(Mega Byte)

CHAPTER 3

SYSTEM DESIGN AND DEVELOPMENT

3.1 SYSTEM ARCHITECTURE

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyze an existing system or model a new one. Like all the best diagrams and charts, a DFD can often visually “say” things that would be hard to explain in words, and they work for both technical and nontechnical audiences, from developer to CEO. That’s why DFDs remain so popular after all these years. While they work well for data flow software and systems, they are less applicable nowadays to visualizing interactive, real-time or database-oriented software or systems.

Data flow diagrams were popularized in the late 1970s, arising from the book Structured Design, by computing pioneers Ed Yourdon and Larry Constantine. They based it on the “data flow graph” computation models by David Martin and Gerald Estrin. The structured design concept took off in the software engineering field, and the DFD method took off with it. It became more popular in business circles, as it was applied to business analysis, than in academic circles.

LEVEL 0 DFD

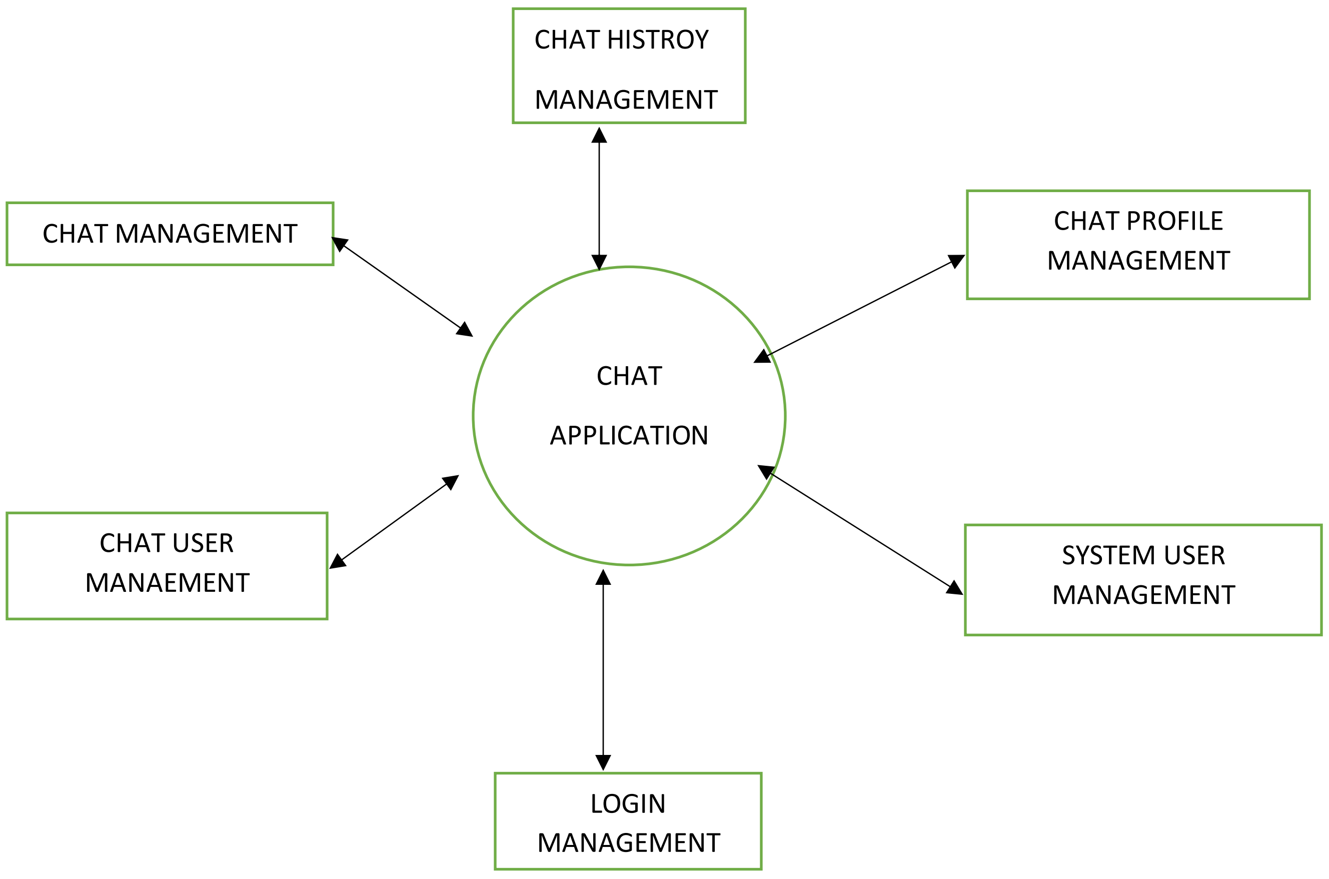


Figure 3.1 Level 0 DFD

LEVEL 1 DFD

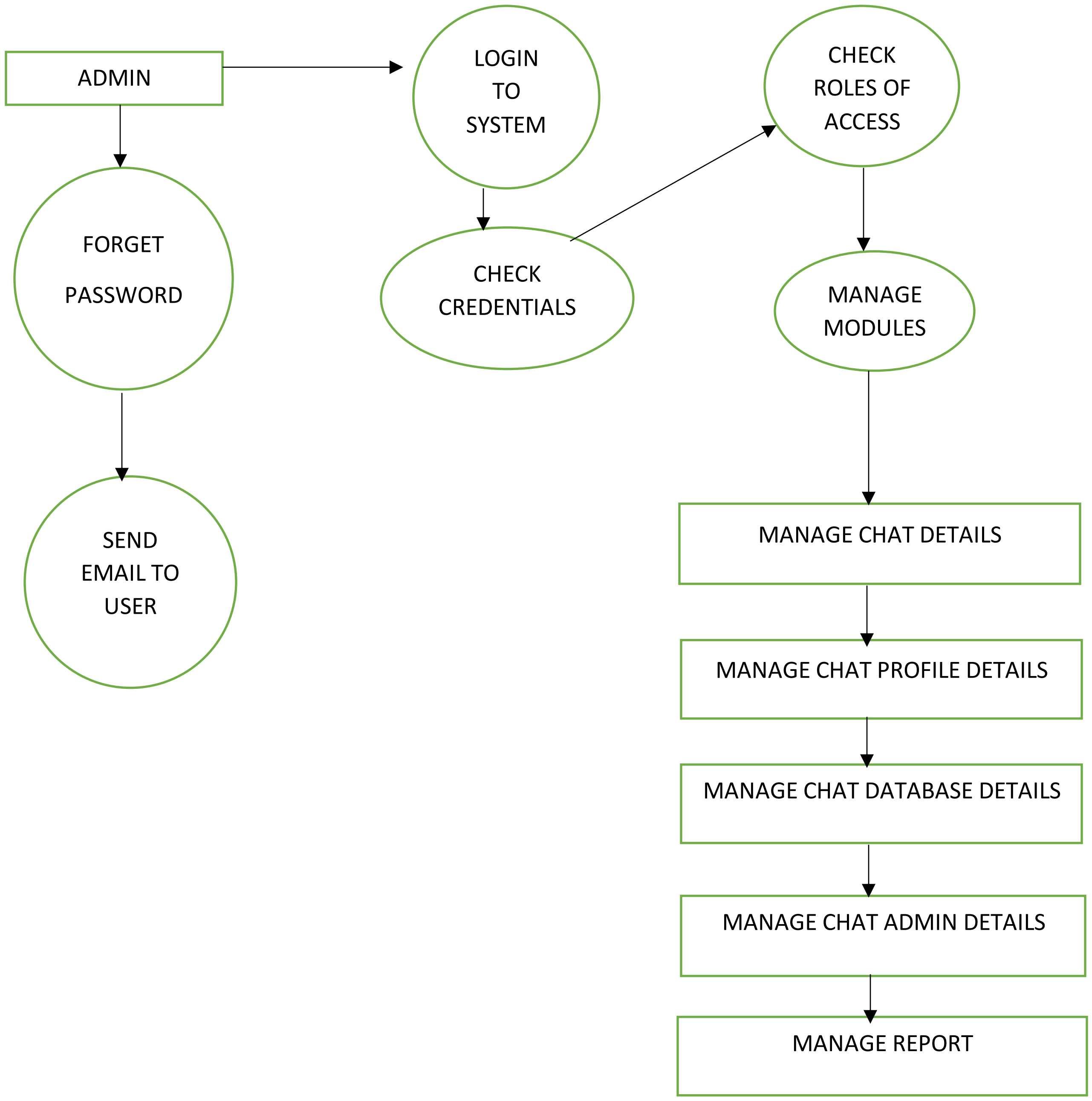


Figure 3.2 Level 1 DFD

3.2 UML DIAGRAM

The elements are like components which can be associated in different ways to make a complete UML picture, which is known as diagram. Thus, it is very important to understand the different diagrams to implement the knowledge in real-life systems.

Any complex system is best understood by making some kind of diagrams or pictures. These diagrams have a better impact on our understanding. If we look around, we will realize that the diagrams are not a new concept but it is used widely in different forms in different industries.

We prepare UML diagrams to understand the system in a better and simple way. A single diagram is not enough to cover all the aspects of the system. UML defines various kinds of diagrams to cover most of the aspects of a system.

USE CASE DIAGRAM

Use case diagram is a graph of actors, set of use cases enclosed by a system boundary, communication (participation) association between the actors and the cases and a generalization among the use cases.

Actors:

An actor represents a set of roles that user of a use case play when interacting with the use cases. Actor identified here is Staff, Student and Administrator.

Use case:

A use case is a description of a set of sequence of action that a system performs to yield result of value to an actor.

- The Login use case is to describe that, the user should choose his/her category whether he/she is user or administrator.
- The use case display the chat details and the offered with their instruction and also about the payment. .

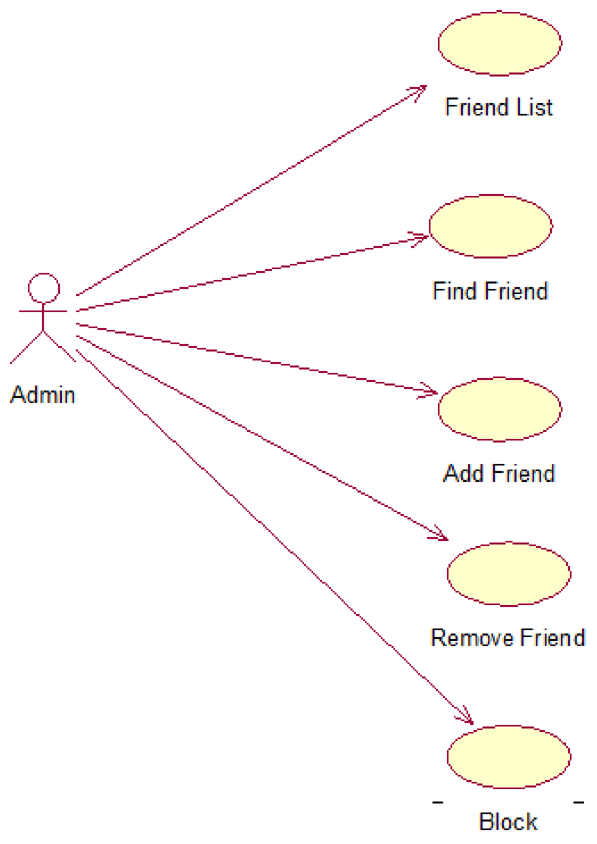


Figure 3.3 Use Case Diagram

SEQUENCE DIAGRAM

The user has to login and fill up the chat form for their othe users need and submit it. For each chat a reference number is given. The administrator views the details with that reference number and he updates the details with additional information.

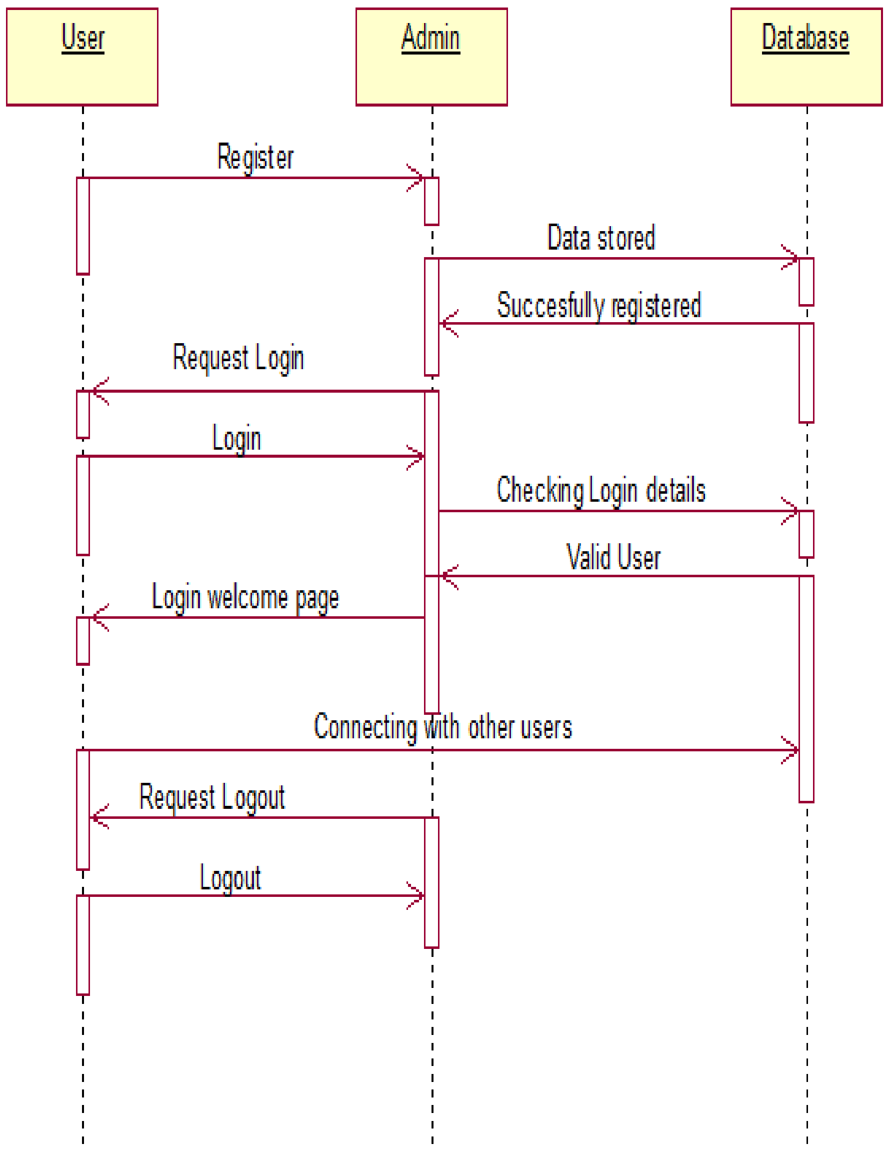


Figure 3.4 Sequence Diagram

ACTIVITY DIAGRAM

The activity diagram describes the sequence of activities with support for both conditional and parallel behavior.The activity diagram is used to describe the various activities taking place in an application.Here in our Chat Application,we have various activities strating from login.

After login,the user selection activity gets performed,where the user can be a user or admin If the user is a student,then they have to enter their password and those details are valid they can access the system.They can register the any user and get the reference number.

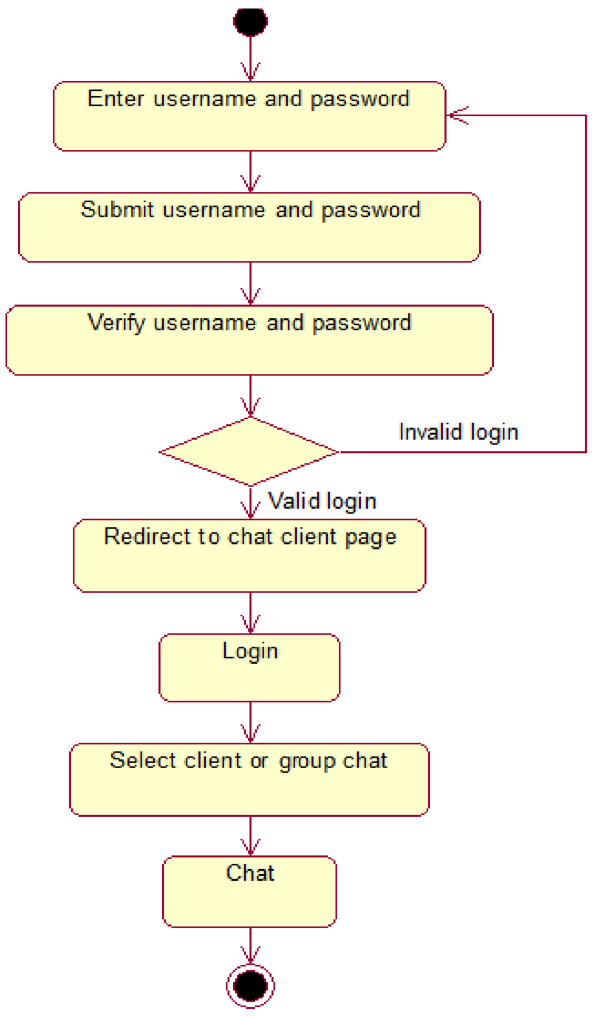

Enter username and password
Submit username and password
Verify username and password
Invalid login
Valid login
Redirect to chat client page
Login
Select client or group chat
Chat

Figure 3.5 Activity Diagram

CLASS DIAGRAM

The class diagram involves various classes used in project and their attributes.It also explains various class members details.

The four class in this project are,

1. Login detail
2. Admin detail
3. Register detail
4. Update detail

Each class has its own attributes and operations.

Login class-the attributes defined is user and admin

Admin class-It contains the attributes of the student and own details.

Register class-It contains the attributes of the student

Update class-the attributes of user can be added to the admin class

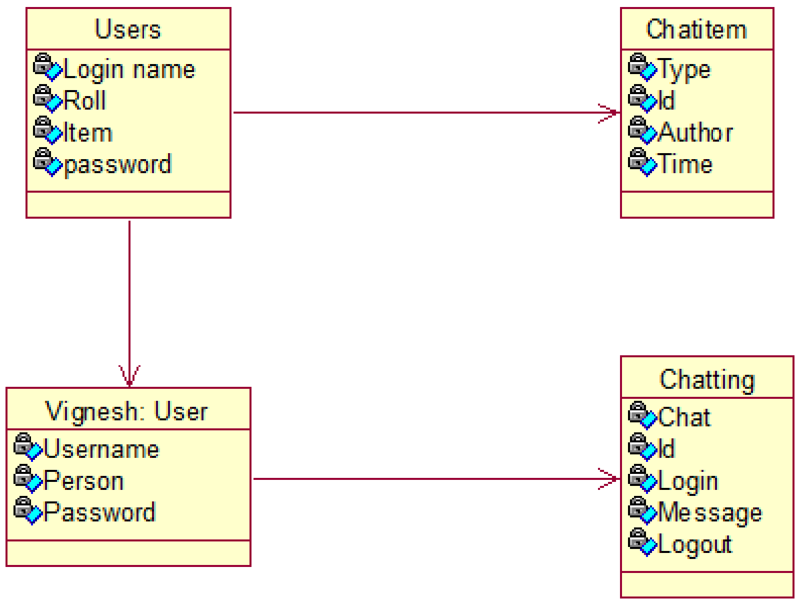


Figure 3.6 Class Diagram

CHAPTER 4

SYSTEM IMPLEMENTATION AND TESTING

4.1 SYSTEM REQUIREMENTS

Storing chat messages and retrieving those messages at a very fast rate is much needed for an chat app. Developers spend days to decide which DBMS to choose and how to design tables for storing messages. Sometimes they endup with difficult and bad designs. Most bad table designs require a lot of sorting. After doing a lot of research and smashing my brain I finally came up with a awesome way of storing and retrieving messages in SQL tables. My method doesn’t require any sorting. And retrieving is very fast even if you have only one server with good configuration.

DATABASE DESIGN

I will be using MySQL as my database management system. But this same design can me implemented using any SQL, No-SQL or any other kind of database management system. In this table we keep the total number of messages between two users. The identifier column will have two usernames separated by a colon. Therefore usernames cannot have colons. The identifier column is the primary key so it has unique values and the column is indexed.You need to construct the identifier while producing SQL statement. Format of the identifier is that, both the usernames are sorted in ascending order.

MESSAGES TABLE

In this table we store all the messages between all users. The identifier_message_number column has identifier appended with the message number between those two users. The identifier and message number is separated by a colon. Then we store the message in the message column.

INSERTING MESSAGES

While inserting a message you need to increment the total_messages value in total_messages column. After increment you need to add the message to the messages table. Here we are checking if row already exists or not. If exists then increment the value otherwise add a new row and make the total messages as 1.Now to insert the message into messages table we need to retrieve the total_messages count from total_messages table and then construct this SQL statement.

RETRIEVING MESSAGES

To retrieve last 4 messages between user1 and user2 we run this SQL query statement:Here we assumed there are total 83 messages for user1 and user2.Here identifier_message_number column is indexed therefore selection will be faster.

CONCLUSION

This table designs and formats makes insertion and selection very fast. This design makes it possible to run all queries in indexed columns. You can add more features and expand this tables.

4.1.1 Hardware Requirements

Processor	: Intel Core I7
RAM	: 4 GB
Hard disk	: 500 GB

4.1.2 SOFTWARE REQUIREMENTS

Operating System	: Windows OS,MacOS
Language	: PHP
Front End	: HTML, JavaScript
Platform	: Sublime Text 3
Web Servers	: Xampp Server
Backend	: MariaDB,Mysql
Browser	: Internal explorer/Mozilla Firefox/Chrome.

4.1.3 Software Description

HTML

It is used to design the web pages in our project.It describes the structure of a web page semantically and originally included cues for the appearance of the document.

CSS

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts.[3] This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

JAVA SCRIPT

JavaScript is a high-level,dynamic,un typed,and interpreted programming language.It has been standardized in the ECMA Script language specification.Alongside HTML and CSS,JavaScript is one of the three core technologies of World Wide Web content production.

PHP

PHP is a server-side scripting language designed primarily for web development but also used as a general-purpopse programming language.PHP code may be embedded into HTML or HTML5 markup,or it can be used in combination with various web template systems,web content management systems and web frameworks.

AJAX

AJAX is a new technique for creating better, faster, and more interactive web applications with the help of XML, HTML, CSS, and Java Script. Ajax uses XHTML for content, CSS for presentation, along with Document Object Model and JavaScript for dynamic content display.

JQUERY

Jquery is a fast,small,and feature-rich JavaScript library.It makes things like HTML document traversal and manipulation,event handling,animation,and Ajax much simpler with an easy-to-use API that works across a multitude of browsers.

MYSQL

MySQL is an open Source relational database management system. MySQL development project has made its source code available under the terms of the GNU General Public License,as well as under a variety proprietary agreements.

MARIADB

MariaDB is a free and open-source cross-platform document-oriented database program.Classified as a NoSQL database program,MariaDB use JSON-like documents with schemas.MariaDB is developed MariaDB and is free and opensource,published under a combination

4.2 MODULE DESCRIPTION

LOGIN FORM

A Login form is used to enter authentication credentials to access a restricted page or form. The login form contains a field for the *username* and another for the *password*. When the login form is submitted its underlying code checks that the credentials are authentic, giving the user can access the restricted page. If a user is not able to provide authentic credentials they will not be able to proceed past the login form.

Like the search form, a login form is basically a record form whose insert, update and delete properties have been disabled.

The Authentication Builder is the easiest and best way to create a login form, however, it is possible to build a login form manually based on the barebones record form provided under the Forms tab of the Toolbox.

Before a login form can be created you have to configure the project security settings under the Security and Security Groups tab of the Project Settings dialog window. These settings specify the database tables and fields that are used during the authentication process.

To expose the login form properties click on either of the glyphs under the Data tab of the Properties window or click on the name of the login form in the Project Explorer.

CHAT FORM

Online chat may refer to any kind of communication over the internet that offers a real-time transmission of text messages from sender to receiver. Chat messages are generally short in order to enable other participants to respond quickly. Thereby, a feeling similar to a spoken conversation is created, which distinguishes chatting from other text-based online communication forms such as internet forums and email. Online chat may address point to point communications as well as multicast communications from one sender to many receivers and voice and video chat, or may be a feature of a web conferencing service.

Online chat in a less stringent definition may be primarily any direct text-based or video-based (webcams), one-on-one chat or one-to-many group chat using tools such as instant messengers, Internet Really Chat talkers and possibly MUDs. The expression *online chat* comes from the word chat which means "informal conversation". Online chat includes web based applications that allow communication – often directly addressed, but anonymous between users in a multi-user environment. Web conferencing is a more specific online service, that is often sold as a service, hosted on a web server controlled by the vendor.

On the Internet, chatting is talking to other people who are using the Internet at the same time you are. Usually, this "talking" is the exchange of typed-in messages requiring one site as the repository for the messages and a group of users who take part from anywhere on the Internet. In some cases, a private chat can be arranged between two parties who meet initially in a group chat. Chats can be ongoing or scheduled for a particular time and duration. Most chats are focused on a particular topic of interest and some involve guest experts or famous people who "talk" to anyone joining the chat.

DATABASE FORM

Creation or deletion of databases in MariaDB requires privileges typically only given to root users or admins. Under these accounts, you have two options for creating a database − the mysqladmin binary and a PHP script.

The following example demonstrates the use of the mysqladmin binary in creating a database with the name Product the mysql_query function in creating a

MariaDB database. The function uses two parameters, one optional, and returns either a value of “true” when successful, or “false” when not.

MariaDB is a fork of the MySQL relational database management system. The original developers of MySQL created MariaDB after concerns raised by Oracle's acquisition of MySQL. This tutorial will provide a quick introduction to MariaDB, and aid you in achieving a high level of comfort with MariaDB programming and administration.

This tutorial targets novice developers and those new to MariaDB. It guides them in understanding basic through more advanced concepts in MariaDB. After completing this tutorial, your firm foundation in MariaDB and level of expertise will allow you to begin developing and easily build on your knowledge.

The tutorial assumes your familiarity with relational database management systems, querying languages, MySQL, and general programming. It also assumes familiarity with typical database operations in an application.

A database application exists separate from the main application and stores data collections. Every database employs one or multiple APIs for the creation, access, management, search, and replication of the data it contains.

4.3 SYSTEM TESTING

SOFTWARE TESTING

Testing objectives include

1. Testing is a process of executing a program with the intent of finding an error.
2. A good test case is one that has a high probability of finding an as yet undiscovered error.
3. A successful test is one that uncovers an as yet undiscovered error.

Testing should systematically uncover different classes of errors in a minimum amount of time and with a minimum amount of effort. A secondary hat it demonstrates that the software appears to be working as started in the specification, The data collected through testing can also provide an indication of the software reliability and quality. But, testing cannot show the absence of defect -- it can only show that software defects are present.

WHITE BOX TESTING

White box testing is a test case design method that uses the control structure of the procedural design to derive test cases. Test cases can be derived that

1. guarantee that all independent paths within a module have been exercised at least once,
2. exercise all logical decisions on their true and false sides,
3. execute all loops at their boundaries and within their operational bounds, and
4. exercise internal data structures to ensure their validity.

BLACK BOX TESTING

Black box testing attempts to derive sets of inputs that will fully exercise all the functional requirements of a system. It is not an alternative to white box testing type of testing attempts to find errors in the following categories:

1. incorrect or missing functions,
2. interface errors,
3. errors in data structures or external database access,
4. performance errors, and
5. Initialization and termination errors.

Tests are designed to answer the following

1. How is the function's validity tested?
2. What classes of input will make good test cases?
3. Is the system particularly sensitive to certain input values?
4. How are the boundaries of a data class isolated?
5. What data rates and data volume can the system tolerate?
6. What effect will specific combinations of data have on system operation?

White box testing should be performed early in the testing process, while black box testing tends to be applied during later stages. Test cases should be derived which reduce the number of additional test cases that must be designed to achieve reasonable testing, and tell us something about the presence or absence of classes oferrors, rather than an error associated only with the specific test at hand.

TEST PLAN

Testing starts with a test plan. The test plan identified all the testing related activities that need to be performed along with the schedule and guidelines for testing It describes the overall strategy for integration Testing is divided into phases and builds that address specific functional and behavioural characteristics of the software.

Each of these phase and sub phases delineates a broad functional category with in the software and can generally be related to a specific domain of the program builds [group of modules] are created to corresponding to each phases.

TESTING STRATEGIES

UNIT TESTING

The most 'micro' scale of testing, to test particular functions or code Typically done by the programmer and not by testers, as it requires tailed knowledge of the internal program design and code. Not always easily done unless the application has a well-designed architecture with tight code, may require developing test driver modules or test harnesses.

INTEGRATION TESTING

Continuous testing of an application as new functionality is added; requires that various aspects of an application's functionality be independent enough to work separately before all parts of the program are completed, or that test drivers be developed as needed; done by programmers or by testers. Testing of combined parts of an application to de Year in if they function together correctly. The 'parts' can be code modules, individual applications, client and server applications on a network, etc. This type of testing is especially relevant to client/server.

VALIDATION TESTING

Black-box type testing geared to functional requirements of an application; this type of testing should be done by testers. This doesn't mean that the programmers shouldn't check that their code works before releasing it (which of any stage of testing). Black box type testing that is based on overall requirements specifications, covers all combined parts of a system

ACCEPTANCE TESTING

Final testing based on specifications of the end-user or customer, or based on use by end-users/customers over some limited period of time.

OUTPUT TESTING

Similar to system testing; the 'macro' end of the test scale; involves testing of a complete application environment in a situation that mimics real-world use, such as interacting with a database, using network communications, or interacting with other hardware, applications, or systems if appropriate.

CHAPTER 5

CONCLUSION AND FUTURE ENCHANCEMENT

CONCLUSION

Web online services presents an effective means where by existing, perhaps loosely defined, system functionality can be adapted to operate in a web services paradigm. Through the use of Service Delegates, the details associated with directly interfacing with local system functionality are encapsulated and effectively isolated from reusable framework components. Semantic web presents design incorporates technologies such as inference engines, rule-based systems, web services and service-oriented architectures to provide the needed infrastructure to support meaningful interoperability among context-based systems. In order to facilitate the interoperability, has developed a set of technologies, standards, and interface protocols, for interoperability of data, information, and systems over the Web.The web service technology and standards are widely accepted and used by the for interoperability among grid systems.

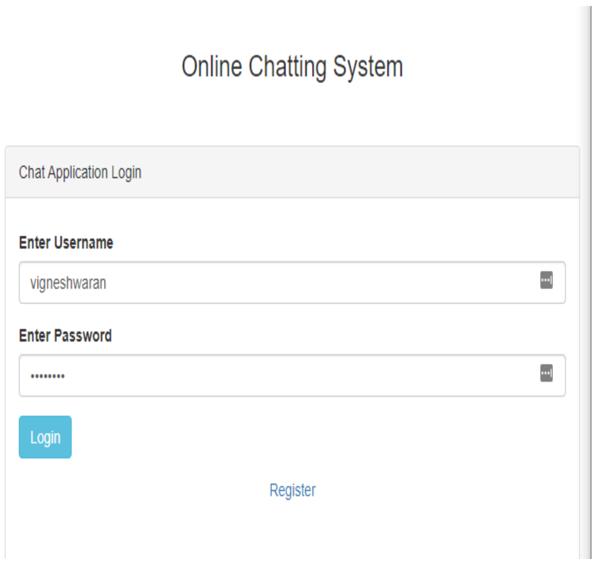
FUTURE ENCHANCEMENT

- Reduces expenses
- Increases sales
- Improve customer service and loyalty
- Faster problem resolution
- Customer convenience
- Competitive advantages
- Expand and market reach
- Reports and analytics

APPENDICES

SCREENSHOTS

Login Form



Online Chatting System

Chat Application Login

Enter Username

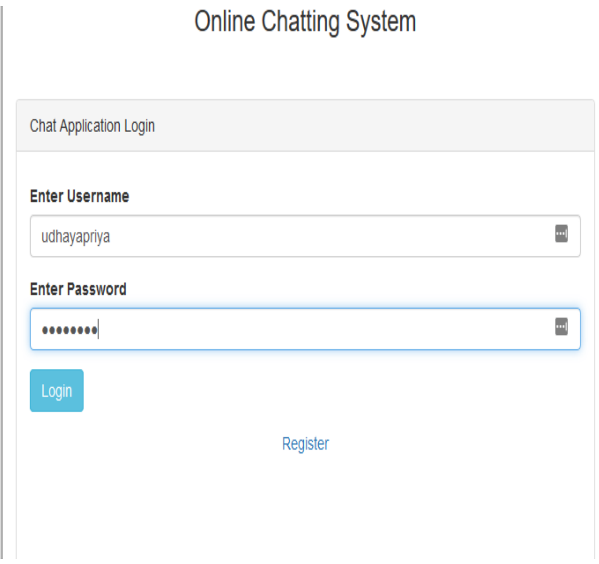
vigneshwaran

Enter Password

••••••••

Login

Register



Online Chatting System

Chat Application Login

Enter Username

udhayapriya

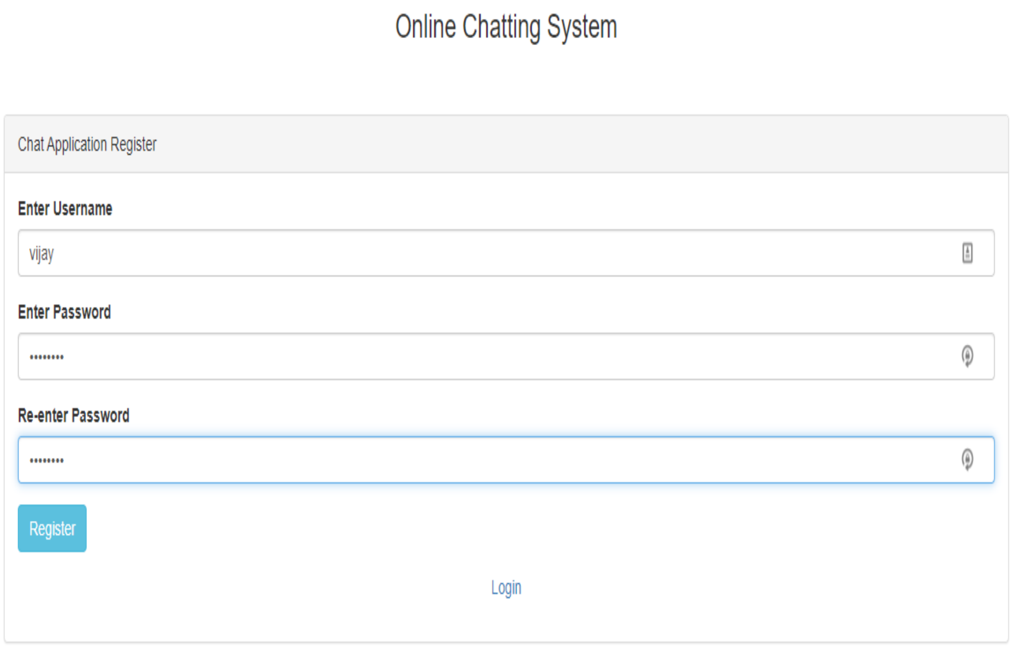
Enter Password

••••••••

Login

Register

Register Form



Online Chatting System

Chat Application Register

Enter Username

vijay

Enter Password

••••••••

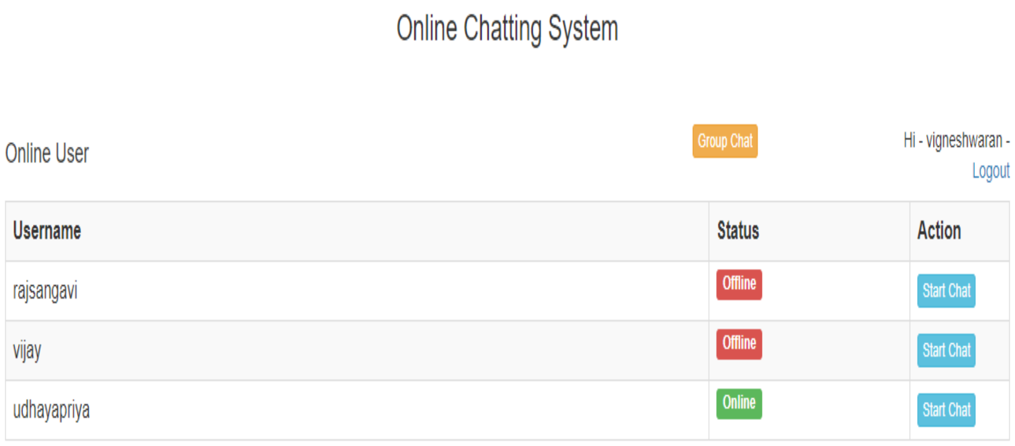
Re-enter Password

••••••••

Register

Login

User Form

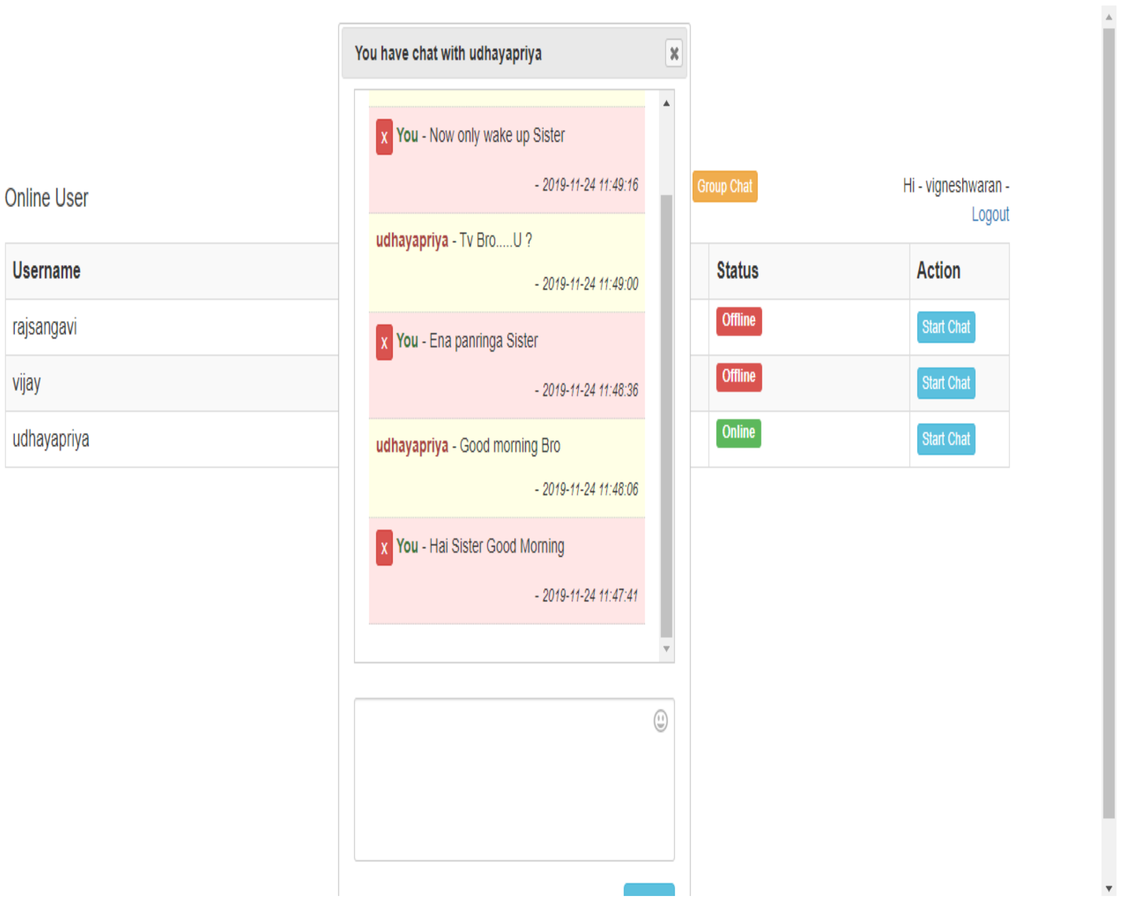

Online Chatting System

Online User

Hi - vigneshwaran - Logout

Username	Status	Action
rajsangavi	Offline	Start Chat
vijay	Offline	Start Chat
udhayapriya	Online	Start Chat

Chat Form


You have chat with udhayapriya

x You - Now only wake up Sister - 2019-11-24 11:49:16

udhayapriya - Tv Bro.....U ? - 2019-11-24 11:49:00

x You - Ena panringa Sister - 2019-11-24 11:48:36

udhayapriya - Good morning Bro - 2019-11-24 11:48:06

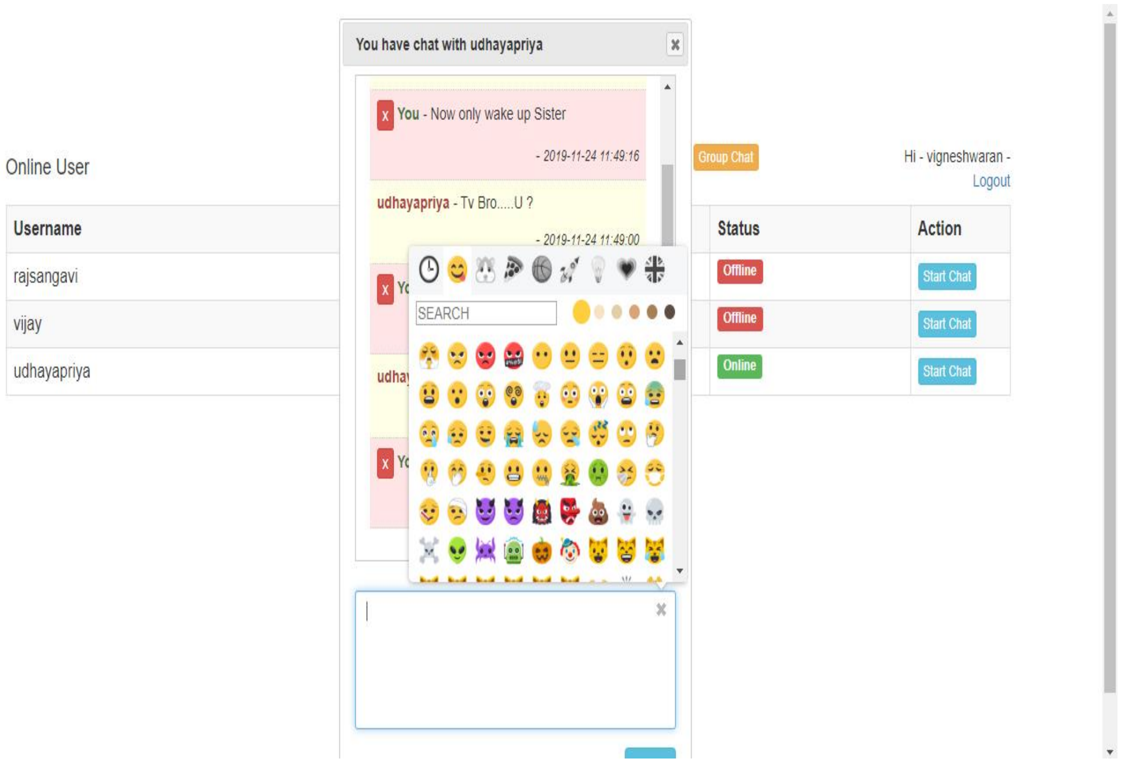
x You - Hai Sister Good Morning - 2019-11-24 11:47:41

Online User

Hi - vigneshwaran - Logout

Username	Status	Action
rajsangavi	Offline	Start Chat
vijay	Offline	Start Chat
udhayapriya	Online	Start Chat

Emoji


You have chat with udhayapriya

x You - Now only wake up Sister

- 2019-11-24 11:49:16

udhayapriya - Tv Bro.....U ?

- 2019-11-24 11:49:00

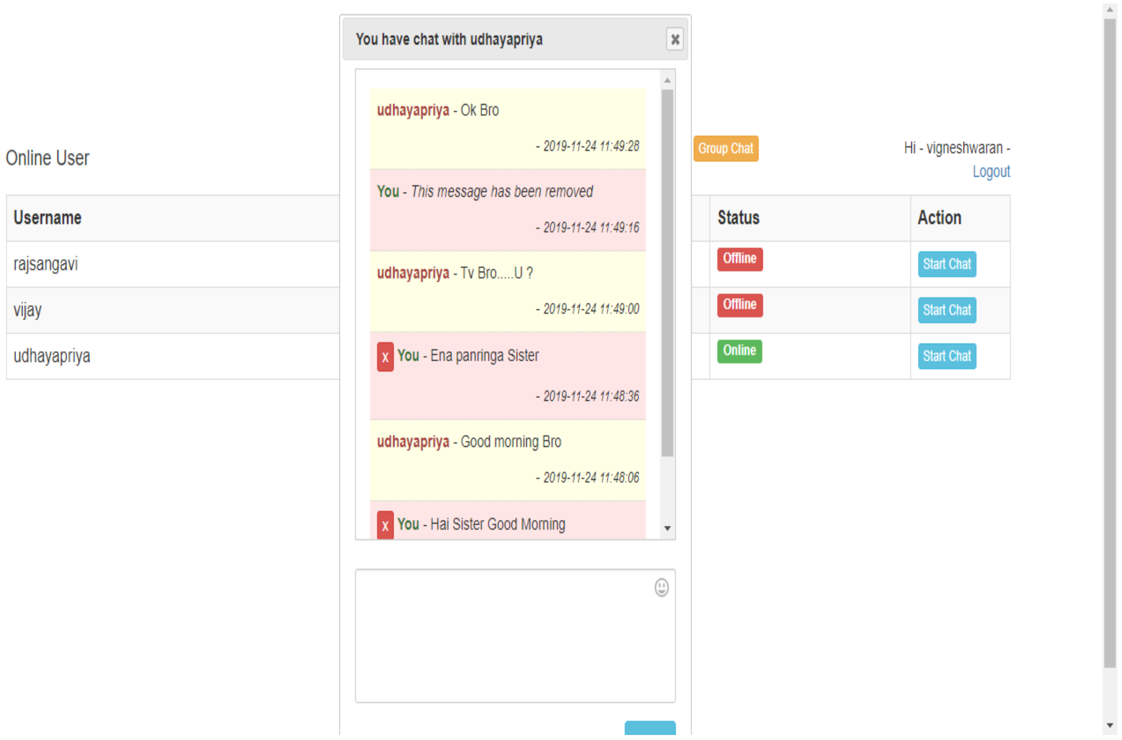
Online User

Group Chat

Hi - vigneshwaran - Logout

Username	Status	Action
rajsangavi	Offline	Start Chat
vijay	Offline	Start Chat
udhayapriya	Online	Start Chat

Delete Message


You have chat with udhayapriya

udhayapriya - Ok Bro

- 2019-11-24 11:49:28

You - *This message has been removed*

- 2019-11-24 11:49:16

udhayapriya - Tv Bro.....U ?

- 2019-11-24 11:49:00

x You - Ena panringa Sister

- 2019-11-24 11:48:36

udhayapriya - Good morning Bro

- 2019-11-24 11:48:06

x You - Hai Sister Good Morning

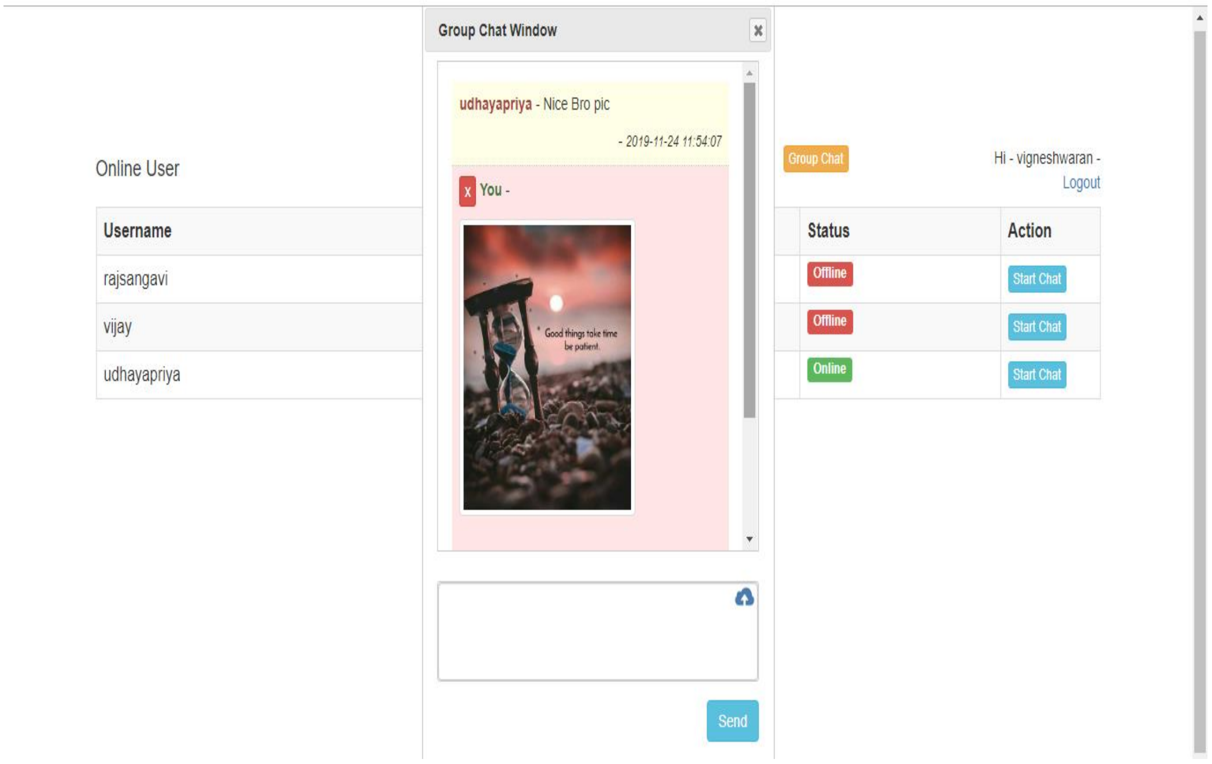
Online User

Group Chat

Hi - vigneshwaran - Logout

Username	Status	Action
rajsangavi	Offline	Start Chat
vijay	Offline	Start Chat
udhayapriya	Online	Start Chat

Group Chat



Group Chat Window

udhayapriya - Nice Bro pic

- 2019-11-24 11:54:07

x You -

Send

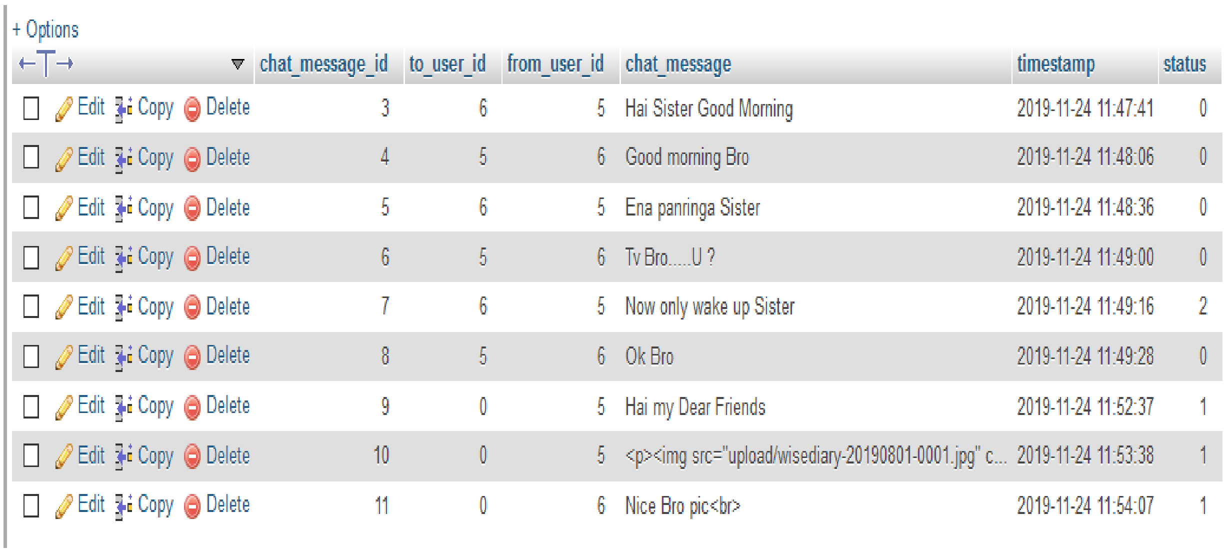
Group Chat

Hi - vigneshwaran -
Logout

Online User

Username	Status	Action
rajsangavi	Offline	Start Chat
vijay	Offline	Start Chat
udhayapriya	Online	Start Chat

Database Form



+ Options

	chat_message_id	to_user_id	from_user_id	chat_message	timestamp	status
Edit Copy Delete	3	6	5	Hai Sister Good Morning	2019-11-24 11:47:41	0
Edit Copy Delete	4	5	6	Good morning Bro	2019-11-24 11:48:06	0
Edit Copy Delete	5	6	5	Ena panringa Sister	2019-11-24 11:48:36	0
Edit Copy Delete	6	5	6	Tv Bro.....U ?	2019-11-24 11:49:00	0
Edit Copy Delete	7	6	5	Now only wake up Sister	2019-11-24 11:49:16	2
Edit Copy Delete	8	5	6	Ok Bro	2019-11-24 11:49:28	0
Edit Copy Delete	9	0	5	Hai my Dear Friends	2019-11-24 11:52:37	1
Edit Copy Delete	10	0	5	<p><img src="upload/wisediary-20190801-0001.jpg" c...	2019-11-24 11:53:38	1
Edit Copy Delete	11	0	6	Nice Bro pic 	2019-11-24 11:54:07	1

SOURCE CODE

DATABASE QUERIES

```
cd ../..

cd xampp

cd mysql

cd bin

mysql –u root –p –h localhost

create database chat;

use chat;

create table `chat_message` (

`chat_message_id` int(11) not null,

`to_user_id` int(11) not null,

`from_user_id` int(11) not null,

`chat_message` text not null,

`timestamp` timestamp not null default current_timestamp,

`status` int(1) not null

) engine=innodb default charset=latin1;

create table `login` (

`user_id` int(11) not null,

`username` varchar(255) not null,

`password` varchar(255) not null

) engine=innodb default charset=latin1;
```

```
insert into `login` (`user_id`,`username`,`password`)values

(1,’vigneshwaran’,’ password’),

(2,’rajsangavi’,’ password’),

(3,’udhayapriya’,’ password’),

(4,’vijay’,’ password’);

CREATE TABLE ‘login_details’ (
‘login_details id’  int (11) NOT NULL,
‘user_id’ int (11) NOT NULL,
‘last_activity’ timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
‘is_type’ enum('no',’yes') NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;


alter table `chat_message`

add primary key(`chat_message_id`);

alter table `login`

add primary key(`user_id`);

alter table `login_details`

add primary key(`login_details_id`);

alter table `chat_message`

modify `chat_message_id` int(11) not null auto_increment;

alter table `login`

modify `user_id` int(11) not null auto_increment, auto_increment=4;

alter table `login_details`

modify `login_details_id` int(11) not null auto_increment;
```

DATABASE_CONNECTION.PHP

```
<?php
//database_connection.php
$connect = new PDO("mysql:host=localhost;dbname=chat", "root", "");
date_default_timezone_set('Asia/Kolkata');
function fetch_user_last_activity($user_id, $connect)
{
 $query = "
 SELECT * FROM login_details
 WHERE user_id = '$user_id'
 ORDER BY last_activity DESC
 LIMIT 1
 ";
 $statement = $connect->prepare($query);
 $statement->execute();
 $result = $statement->fetchAll();
 foreach($result as $row)
 {
  return $row['last_activity'];
 }
}

function fetch_user_chat_history($from_user_id, $to_user_id, $connect)
{
 $query = "
 SELECT * FROM chat_message
 WHERE (from_user_id = '".$from_user_id."'
 AND to_user_id = '".$to_user_id."')
 OR (from_user_id = '".$to_user_id."'
 AND to_user_id = '".$from_user_id."')
 ORDER BY timestamp DESC
 ";
 $statement = $connect->prepare($query);
 $statement->execute();
 $result = $statement->fetchAll();
 $output = '<ul class="list-unstyled">';
 foreach($result as $row)
 {
  $user_name = '';
  if($row["from_user_id"] == $from_user_id)
  {
   $user_name = '<b class="text-success">You</b>';
  }
  else
  {
   $user_name = '<b class="text-danger">'.get_user_name($row['from_user_id'],
$connect).'</b>';
  }
  $output .= '
  <li style="border-bottom:1px dotted #ccc">
   <p>'.$user_name.' - '.$row["chat_message"].'
    <div align="right">
     - <small><em>'.$row['timestamp'].'</em></small>
    </div>
   </p>
  </li>
  ';
 }
 $output .= '</ul>';
 $query = "
 UPDATE chat_message
 SET status = '0'
 WHERE from_user_id = '".$to_user_id."'
 AND to_user_id = '".$from_user_id."'
 AND status = '1'
 ";
 $statement = $connect->prepare($query);
 $statement->execute();
 return $output;
}

function get_user_name($user_id, $connect)
{
 $query = "SELECT username FROM login WHERE user_id = '$user_id'";
 $statement = $connect->prepare($query);
 $statement->execute();
 $result = $statement->fetchAll();
 foreach($result as $row)
 {
  return $row['username'];
 }
}

function count_unseen_message($from_user_id, $to_user_id, $connect)
{
 $query = "
 SELECT * FROM chat_message
 WHERE from_user_id = '$from_user_id'
 AND to_user_id = '$to_user_id'
 AND status = '1'
 ";
 $statement = $connect->prepare($query);
 $statement->execute();
 $count = $statement->rowCount();
 $output = '';
 if($count > 0)
 {
  $output = '<span class="label label-success">'.$count.'</span>';
 }
 return $output;
}

function fetch_is_type_status($user_id, $connect)
{
 $query = "
 SELECT is_type FROM login_details
 WHERE user_id = '".$user_id."'
 ORDER BY last_activity DESC
 LIMIT 1
 ";
 $statement = $connect->prepare($query);
 $statement->execute();
 $result = $statement->fetchAll();
 $output = '';
 foreach($result as $row)
 {
  if($row["is_type"] == 'yes')
  {
   $output = ' - <small><em><span class="text-muted">Typing...</span></em></small>';
  }
 }
 return $output;
}
?>
```

LOGIN.PHP

```
<!--
//login.php
!-->

<?php

include('database_connection.php');

session_start();

$message = '';

if(isset($_SESSION['user_id']))
{
 header('location:index.php');
}

if(isset($_POST["login"]))
```

```
{
 $query = "
   SELECT * FROM login
    WHERE username = :username
 ";
 $statement = $connect->prepare($query);
 $statement->execute(
    array(
      ':username' => $_POST["username"]
     )
  );
  $count = $statement->rowCount();
  if($count > 0)
 {
  $result = $statement->fetchAll();
    foreach($result as $row)
    {
      if(password_verify($_POST["password"], $row["password"]))
      {
        $_SESSION['user_id'] = $row['user_id'];
        $_SESSION['username'] = $row['username'];
        $sub_query = "
        INSERT INTO login_details
        (user_id)
        VALUES ('".$row['user_id']."')
        ";
        $statement = $connect->prepare($sub_query);
        $statement->execute();
        $_SESSION['login_details_id'] = $connect->lastInsertId();
        header("location:index.php");
      }
      else
      {
       $message = "<label>Wrong Password</label>";
      }
    }
 }
 else
 {
  $message = "<label>Wrong Username</labe>";
 }
}

?>

<html>
    <head>
        <title>Chat Application using PHP Ajax Jquery</title>
  <link rel="stylesheet" href="//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">
        <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
  <script src="https://code.jquery.com/jquery-1.12.4.js"></script>
    <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
    </head>
    <body>
        <div class="container">
   <br />
```

```
    <h3 align="center">Chat Application using PHP Ajax Jquery</a></h3><br />
    <br />
    <div class="panel panel-default">
      <div class="panel-heading">Chat Application Login</div>
     <div class="panel-body">
      <form method="post">
       <p class="text-danger"><?php echo $message; ?></p>
       <div class="form-group">
        <label>Enter Username</label>
        <input type="text" name="username" class="form-control" required />
       </div>
       <div class="form-group">
        <label>Enter Password</label>
        <input type="password" name="password" class="form-control" required />
       </div>
       <div class="form-group">
        <input type="submit" name="login" class="btn btn-info" value="Login" />
       </div>
      </form>
     </div>
    </div>
   </div>
     </body>
</html>
```

INDEX.PHP

```
<!--
//index.php
!-->

<?php

include('database_connection.php');

session_start();

if(!isset($_SESSION['user_id']))
{
 header("location:login.php");
}

?>

<html>
    <head>
        <title>Chat Application using PHP Ajax Jquery</title>
  <link rel="stylesheet" href="//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">
        <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
        <link rel="stylesheet"
href="https://cdn.rawgit.com/mervick/emojionearea/master/dist/emojionearea.min.css">
  <script src="https://code.jquery.com/jquery-1.12.4.js"></script>
    <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
    <script
src="https://cdn.rawgit.com/mervick/emojionearea/master/dist/emojionearea.min.js"></s
cript>
    </head>
    <body>
        <div class="container">
   <br />

   <h3 align="center">Chat Application using PHP Ajax Jquery</a></h3><br />
   <br />

   <div class="table-responsive">
    <h4 align="center">Online User</h4>
    <p align="right">Hi - <?php echo $_SESSION['username'];  ?> - <a
href="logout.php">Logout</a></p>
    <div id="user_details"></div>
    <div id="user_model_details"></div>
   </div>
  </div>
    </body>
</html>
<script>
$(document).ready(function(){

 fetch_user();

 setInterval(function(){
  update_last_activity();
  fetch_user();
  update_chat_history_data();
 }, 5000);

 function fetch_user()
 {
  $.ajax({
   url:"fetch_user.php",
   method:"POST",
   success:function(data){
    $('#user_details').html(data);
   }
  })
 }

 function update_last_activity()
 {
  $.ajax({
   url:"update_last_activity.php",
   success:function()
   {

   }
  })
 }

 function make_chat_dialog_box(to_user_id, to_user_name)
 {
  var modal_content = '<div id="user_dialog_'+to_user_id+'" class="user_dialog"
title="You have chat with '+to_user_name+'">';
```

```
  modal_content += '<div style="height:400px; border:1px solid #ccc; overflow-y: scroll;
margin-bottom:24px; padding:16px;" class="chat_history" data-
touserid="'+to_user_id+'" id="chat_history_'+to_user_id+'">';
  modal_content += fetch_user_chat_history(to_user_id);
  modal_content += '</div>';
  modal_content += '<div class="form-group">';
  modal_content += '<textarea name="chat_message_'+to_user_id+'"
id="chat_message_'+to_user_id+'" class="form-control chat_message"></textarea>';
  modal_content += '</div><div class="form-group" align="right">';
  modal_content+= '<button type="button" name="send_chat" id="'+to_user_id+'"
class="btn btn-info send_chat">Send</button></div></div>';
  $('#user_model_details').html(modal_content);
 }

 $(document).on('click', '.start_chat', function(){
  var to_user_id = $(this).data('touserid');
  var to_user_name = $(this).data('tousername');
  make_chat_dialog_box(to_user_id, to_user_name);
  $("#user_dialog_"+to_user_id).dialog({
   autoOpen:false,
   width:400
  });
  $('#user_dialog_'+to_user_id).dialog('open');
  $('#chat_message_'+to_user_id).emojioneArea({
   pickerPosition:"top",
   toneStyle: "bullet"
  });
 });
```

```
$(document).on('click', '.send_chat', function(){
 var to_user_id = $(this).attr('id');
 var chat_message = $('#chat_message_'+to_user_id).val();
 $.ajax({
  url:"insert_chat.php",
  method:"POST",
  data:{to_user_id:to_user_id, chat_message:chat_message},
  success:function(data)
  {
   //$('#chat_message_'+to_user_id).val('');
   var element = $('#chat_message_'+to_user_id).emojioneArea();
   element[0].emojioneArea.setText('');
   $('#chat_history_'+to_user_id).html(data);
  }
 })
});

function fetch_user_chat_history(to_user_id)
{
 $.ajax({
  url:"fetch_user_chat_history.php",
  method:"POST",
  data:{to_user_id:to_user_id},
  success:function(data){
   $('#chat_history_'+to_user_id).html(data);
  }
 })
}
```

```
function update_chat_history_data()
{
 $('.chat_history').each(function(){
  var to_user_id = $(this).data('touserid');
  fetch_user_chat_history(to_user_id);
 });
}

$(document).on('click', '.ui-button-icon', function(){
 $('.user_dialog').dialog('destroy').remove();
});

$(document).on('focus', '.chat_message', function(){
 var is_type = 'yes';
 $.ajax({
  url:"update_is_type_status.php",
  method:"POST",
  data:{is_type:is_type},
  success:function()
  {

  }
 })
});

$(document).on('blur', '.chat_message', function(){
 var is_type = 'no';
 $.ajax({
  url:"update_is_type_status.php",
   method:"POST",
   data:{is_type:is_type},
   success:function()
   {

   }
  })
 });

});
</script>
```

LOGOUT.PHP

```
<?php

//logout.php

session_start();

session_destroy();

header('location:login.php');

?>
```

FETCH USER.PHP

```
<?php

//fetch_user.php

include('database_connection.php');

session_start();

$query = "
SELECT * FROM login
WHERE user_id != '".$_SESSION['user_id']."'
";

$statement = $connect->prepare($query);

$statement->execute();

$result = $statement->fetchAll();

$output = '
<table class="table table-bordered table-striped">
 <tr>
  <th width="70%">Username</td>
  <th width="20%">Status</td>
  <th width="10%">Action</td>
 </tr>
';
```

```
foreach($result as $row)
{
 $status = '';
 $current_timestamp = strtotime(date("Y-m-d H:i:s") . '- 10 second');
 $current_timestamp = date('Y-m-d H:i:s', $current_timestamp);
 $user_last_activity = fetch_user_last_activity($row['user_id'], $connect);
 if($user_last_activity > $current_timestamp)
 {
  $status = '<span class="label label-success">Online</span>';
 }
 else
 {
  $status = '<span class="label label-danger">Offline</span>';
 }
 $output .= '
 <tr>
  <td>'.$row['username'].' '.count_unseen_message($row['user_id'],
$_SESSION['user_id'], $connect).' '.fetch_is_type_status($row['user_id'],
$connect).'</td>
  <td>'.$status.'</td>
  <td><button type="button" class="btn btn-info btn-xs start_chat" data-
touserid="'.$row['user_id'].'" data-tousername="'.$row['username'].'">Start
Chat</button></td>
 </tr>
 ';
}

$output .= '</table>';
```

```
echo $output;

?>
```

UPDATE LAST ACTIVITY.PHP

```
<?php

//update_last_activity.php

include('database_connection.php');

session_start();

$query = "
UPDATE login_details
SET last_activity = now()
WHERE login_details_id = '".$_SESSION["login_details_id"]."'
";

$statement = $connect->prepare($query);

$statement->execute();

?>
```

INSERT CHAT.PHP

```
<?php

//insert_chat.php

include('database_connection.php');

session_start();

$data = array(
 ':to_user_id'  => $_POST['to_user_id'],
 ':from_user_id'  => $_SESSION['user_id'],
 ':chat_message'  => $_POST['chat_message'],
 ':status'   => '1'
);

$query = "
INSERT INTO chat_message
(to_user_id, from_user_id, chat_message, status)
VALUES (:to_user_id, :from_user_id, :chat_message, :status)
";
$statement = $connect->prepare($query);
if($statement->execute($data))
{
 echo fetch_user_chat_history($_SESSION['user_id'], $_POST['to_user_id'], $connect);
}
?>
```

FETCH USER CHAT HISTORY.PHP

```
<?php

//fetch_user_chat_history.php

include('database_connection.php');

session_start();

echo fetch_user_chat_history($_SESSION['user_id'], $_POST['to_user_id'], $connect);

?>
```

UPDATE IS TYPE STATUS.PHP

```
<?php
//update_is_type_status.php
include('database_connection.php');
session_start();
$query = "
UPDATE login_details
SET is_type = '".$_POST["is_type"]."'
WHERE login_details_id = '".$_SESSION["login_details_id"]."'
";
$statement = $connect->prepare($query);
$statement->execute();
?>
```

REFERENCES

1. Luke welling, Laura Thomson, "PHP and MYSQL Web development", 3rd edition, Pearson Education.

2. Kevin Tat roe. Programming PHP", 3rd edition, Pearson Education.

3. Andy Harris, "PHP/MYSQL Programming for the absolute beginner" 3rd edition, Pearson Education.

4. G.K Gupta "Database Management System 2 edition, Tata McGraw-H:11 Education..

5. https://www.weblesson.info/2018/07/live-chat-system-in-php.jquery.html?
6.https://www.mylivechat.com