

# **UNIVERSITY OF MUMBAI**

A PROJECT REPORT ON

**“Talk”**

SUBMITTED BY

**Patel Arun Ramjanak**

Under the guidance of

**Prof. Sonia Dubey**

**Late Shri. Vishnu Waman Thakur Charitable Trust's**

**VIVA INSTITUTE OF TECHNOLOGY**

**Shirgaon, Virar(East)**

**2021-2022**

Late Shri. Vishnu Waman Thakur Charitable Trust's

# VIVA INSTITUTE OF TECHNOLOGY

Shirgaon, Virar(East)



## CERTIFICATE

This is to certify that

**Mr. Patel Arun Ramjanak**

Has satisfactorily completed the project entitled

**Talk**

Towards the partial fulfillment of the

**MASTER OF COMPUTER APPLICATION (MCA)**

As laid by University of Mumbai.

\_\_\_\_\_  
**Principal**

\_\_\_\_\_  
**External Examiner**

\_\_\_\_\_  
**Internal Guide**

## Declaration

We hereby declare that we have completed the project under the guidance of **Prof. Sonia Dubey**. We have designed the system and have done all the programming required.

It may require some modifications in the future as per the user's requirements. From practical implementation point of view, flexibility in the changes have incorporated in the package.

We assure that we can do any kind of modification suggested while practical implementation by modifying file design or the program code if necessary.

## Acknowledgement

First and foremost. we bow our head to the Almighty for being our light and His showers of blessing during this mini project. I extend our gratitude to **Dr. ARUN KUMAR**, Principal. Viva Institute of Technology, to provide all the resources to do this mini project work. we record our sincere thanks and respect to our Head of Department, Prof. Chandani Patel, Department of Master of Computer Application, Viva Institute of Technology, with her good attention and outstanding guidance in all the work of our project.

We are indebted to our project coordinator and project guide **Prof. Sonia Dubey**, Department of Master of Computer Application, Viva Institute of Technology for her guidance and important suggestions for all the ways to complete our project.

we extend our deepest gratitude to all the other staff for their support and all assistance during the study and course work of the project. Finally, we truly thank our parents and friends for giving us valuable and suggestion for the improvement of our project.

## **Abstract**

Communication is means for people to exchange messages. It has started since the beginning of human creation. Distance communication began as early as 1800 century with the introduction of television, telegraph and then telephony. Interesting enough, telephone communication is stands out as the fastest growing technology, from fixed line to mobile wireless, from voice call to data transfer. The emergence of computer network and telecommunication technologies bears the same objective that is to allow people to communicate.

Chatting may refer to the transmission of data over the internet that offers a real-time transmission of data from sender to receiver. Communication is essential for achieving managerial and organizational effectiveness, employees will not be able to aware of what their co-workers are doing, will not have any idea about what their goal are, and will not be able to assess their performance. In absence of channels of communication, supervisors will not be able to give instruction to their subordinates and management will not receive the information it requires to develop plans and take decision.

Chatting is a method of using the technology to bring people and ideas together despite of the geographical barriers. The technology has been available for years but the acceptance it was quite recent. Our project is an example of an android chat. It is made up of application, the client application which runs on the user mobile. To start the chatting our client should get connect to the server, where they can do group and private chatting.

# INDEX

Sr. No.	Contents	Page No.
1	<b>Introduction</b> <b>1.1 Introduction</b> 1.1.1 Problem definition 1.1.2 Objectives of Project 1.1.3 Scope of Project <b>1.2 Technical Details</b> 1.2.1 Overview of Front-End 1.2.2 Overview of Back-End	
2	<b>System Study and Planning</b> <b>2.1 System Study</b> 2.1.1 Existing System 2.1.2 Disadvantages of Existing system 2.1.3 Proposed System <b>2.2 System Planning and Schedule</b> 2.2.1 S/W development Model	
3	<b>System Design</b> <b>3.1 Software Requirement Specification (SRS)</b> 3.1.1 Introduction of SRS 3.1.2 Technology Requirements	

	3.1.2.1 Hardware to be used 3.1.2.2 Software/tools to be used <b>3.2 Detailed life Cycle of the Project</b> 3.2.1 Modules 3.2.2 Object Oriented Analysis & Design Diagrams 3.2.2.1 Use Case Diagram 3.2.2.2 Activity Diagram 3.2.2.3 Sequence Diagram 3.2.2.4 Flowchart/DFD/ER diagram 3.2.3 Database 3.2.3.1 Database Table 3.2.4 I/O Screen Layout	
4	<b>Testing</b> 4.1 Methodologies used for testing 4.2 Types of Testing (Whichever Used)	
5	<b>Conclusion</b>	
6	<b>Limitations</b>	
7	<b>Future Enchantments</b>	
8	<b>References</b>	

## Introduction

The purpose of this document is to present an overall description and listing of the functionality of The Testing Assistant for Regression Testing. It will explain the scope of the project as well as describe the system environment. This document will also include an easily traceable means by which the user can trace each functionality's brief description to its full description. Also included is a user interface specification whereby the user can demonstrate interface standards to be used in designing the system. Furthermore, considerations regarding non-functional requirements and system evolution are addressed.

This document is intended for users of the system including Manual Testing, User Interface Testing, and any individuals involved in testing a new system in development. This document will also be used as a reference for the developers of The Testing Assistant for Regression Testing

Teleconferencing or chatting, is a method of using technology to bring people and ideas "together" despite of the geographical barriers. The technology has been available for years but the acceptance of it was quite recent. This project is a web application which is hosted on the server computer of a network. The users are required to visit the website link on which the application is hosted in order to chat. Users can practice two kinds of chatting, public one (message is broadcasted to multiple connected users) and private one (between any 2 users only). There is also a facility to share images, and audio files. Typically, the user will first login or signup to the chat application. On successful login, the user reaches the main page. At this page, the user can select another user from recent chats list or from new users list for chatting. After selecting user, he will be displayed with their older conversation, the user can now type message in the input box or can attach files and send them.



## **Problem definition And Objectives of Project**

The chat application provides a platform for the clients/users to communicate with the other users and it gives a good user interface. Communication over a network is one field where this tool finds wide ranging application. Chat application establishes a connection between 2 or more systems connected over an intra-net or ad-hoc. This tool can be used for large scale communication and conferencing in an organization or campus of vast size, thus increasing the standard of co-operation. In addition, it converts the complex concept of sockets to a user-friendly environment. This application also provides facility to share files. This application can have further potentials, such as voice chatting and call facility options that can be worked upon later.

## **Scope of Project**

In this project, we can search any user and rooms which is available on database. In this Project, we can connect with any user and public rooms and start chatting. User can also send image and audio file in chat. And user can see image of another user. User can send emoji and delete their message in chat. And user can delete full chats.

## **Overview Of Frontend**

We are using ReactJS and JavaScript on the frontend. ReactJS use for creating single-page applications. That is the reason we are using ReactJS in the Frontend. we also use bootstrap and font-awesome in the frontend for creating beautiful UI. We are using a third-party emoji provide to add emoji in this chat application. In Frontend, we call Firestore display the user and rooms information. On the frontend side, we also manage security if a user is not login, then they will only able to view other users, rooms and chats and if they successfully login then they will be able to create room and chat with some on other. We are using firebase login system and internal google authentication to login and creating new users. We are using different – different firebase hook to using firebase properly. We are adding full page image view functionality and audio recording and audio player functionality.

## **Overview Of Backend**

Currently in this project we are not add any backend. But in future we also planning to add backend system using NodeJS. This backend server helps to video calling and audio calling and other feature like notification also.

## Existing System

There are many existing systems like WhatsApp, Instagram, Telegram and Facebook etc.

### Disadvantages of Existing system

Limitations the word itself says that there is a limit in the application. Every system has its boundaries and it cannot go beyond their limits are as follows: -

- ✚ In a group, the size limit of the group is equals to 256 persons currently (may be changed in future)
- ✚ In a group only the Group-Admin can add the persons/give admin title to other persons in the group and all the 256 can be the group admins.
- ✚ Storing/Retrieving the Chat-Backup may be messy and takes time.
- ✚ While changing to a new device using the same number, the existing chat stored in the old device cannot be retrieved in this new device (in-case of not having chat backup).
- ✚ Constant/Unwanted Messages like images and videos may eat all the space in the device (can be rectified by changing way of downloading images/videos/documents automatically or manually using the download preference option).
- ✚ It uses data as its fuel and will not work without data (offline) like MMS and Text Messaging System.
- ✚ Profile picture is visible to every person having your contact number and using this app, whether known by you or not.
- ✚ No logout option is available.

## Proposed System

We are going to develop an application to overcome the limitations of the current system are as follows: -

- ✚ Currently the present system holds the 256 members in the one group and we are proposing the system which can hold maximum 500 members in the one group.
- ✚ Logout option is available in the proposed system.
- ✚ To maintain the system reliability and maintain the system to hold the multiple user at the same time.
- ✚ User has the right to download the file automatically or manually as per his/her needs to save the storage and data.

## **Software Development model: - The Agile Modeling**

The rest of the SDLC models we've chosen fall under the umbrella of Agile. Nowadays, more than 70% of organizations employ this or that Agile approach in their IT projects. In general, at the heart of Agile are iterative development, intensive communication, and early customer feedback.

Each Agile iteration usually takes several weeks and delivers a complete working software version. The models of this group put more focus on delivering a functioning part of the application quickly. They pay less attention to detailed software documentation (detailed requirement specification, detailed architecture description), and more to software testing activities. This fosters quick development but considerably prolongs software transfer to the support team as well as makes its maintenance more complicated as more time is spent to find the problem when there's no detailed software description.

Agile is about working in close collaboration both across the team and with the customers. At the end of each iteration, stakeholders review the development progress and re-evaluate the priority of tasks for the future iteration to increase the return on investment (ROI) and ensure alignment with user needs and business goals.

Accordingly, frequent releases are characteristic to the Agile models. They also allow for continuous software improvement with easy fixes and changes, quick updates, and feature addition, and help to deliver applications that satisfy users' needs better. However, the lack of detailed planning and openness to changes make it difficult to accurately estimate budget, time and people required for the project.

In the Chat Website, we first developed the small module on the frontend side. After Integration, we test the module and if the system working fine then we take feedback on that system. And then we go on another module that is the reason we used the agile software developer model in our project

## **Technology Requirements**

- Client does not need any Technology Requirements to use this project.
- The developer has good knowledge of React - JS, HTML, CSS, JavaScript

## **Hardware requirement**

We are deploying this project on Firebase. Firebase is a cloud platform where anyone develops their projects for free. After deploying on Firebase no need for hardware anyone can access this project from anywhere in the world just using the internet.

Ram - 4 GB

PROCESS - Intel core i3,

hard drive -250 GB

## **Software requirement**

IDEs - Visual Studio

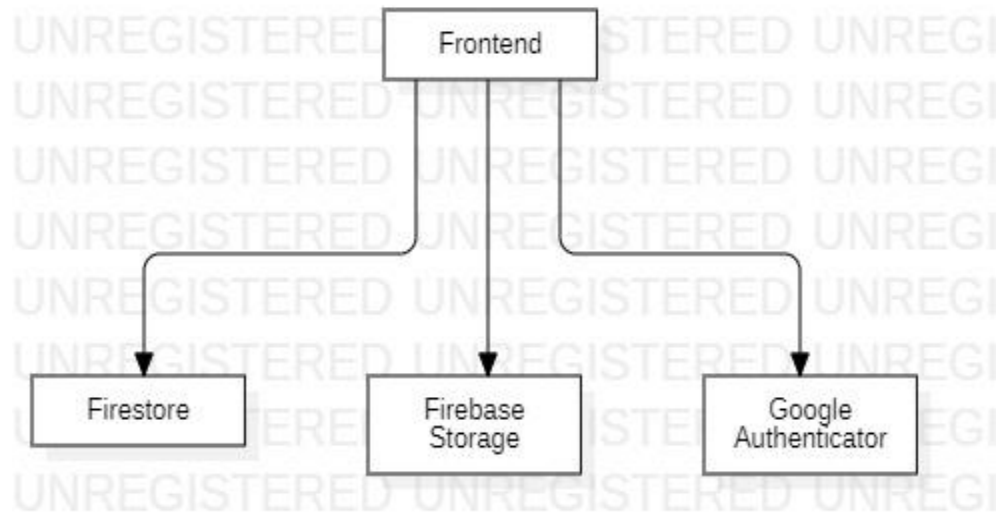
Database – Firestore

Development Tool - Node JS

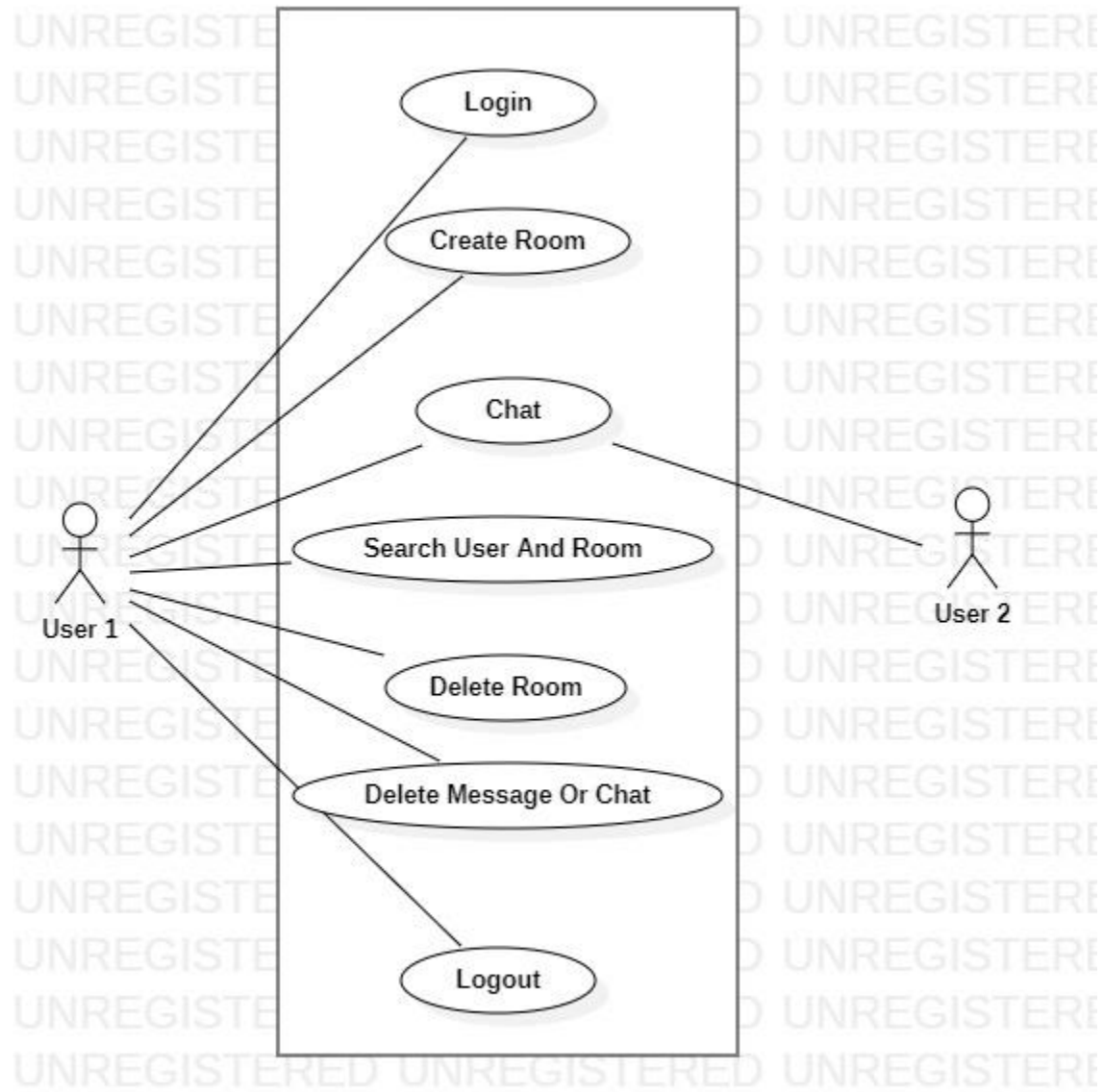
Programming Language - JavaScript

Framework – ReactJS

## Model Diagram

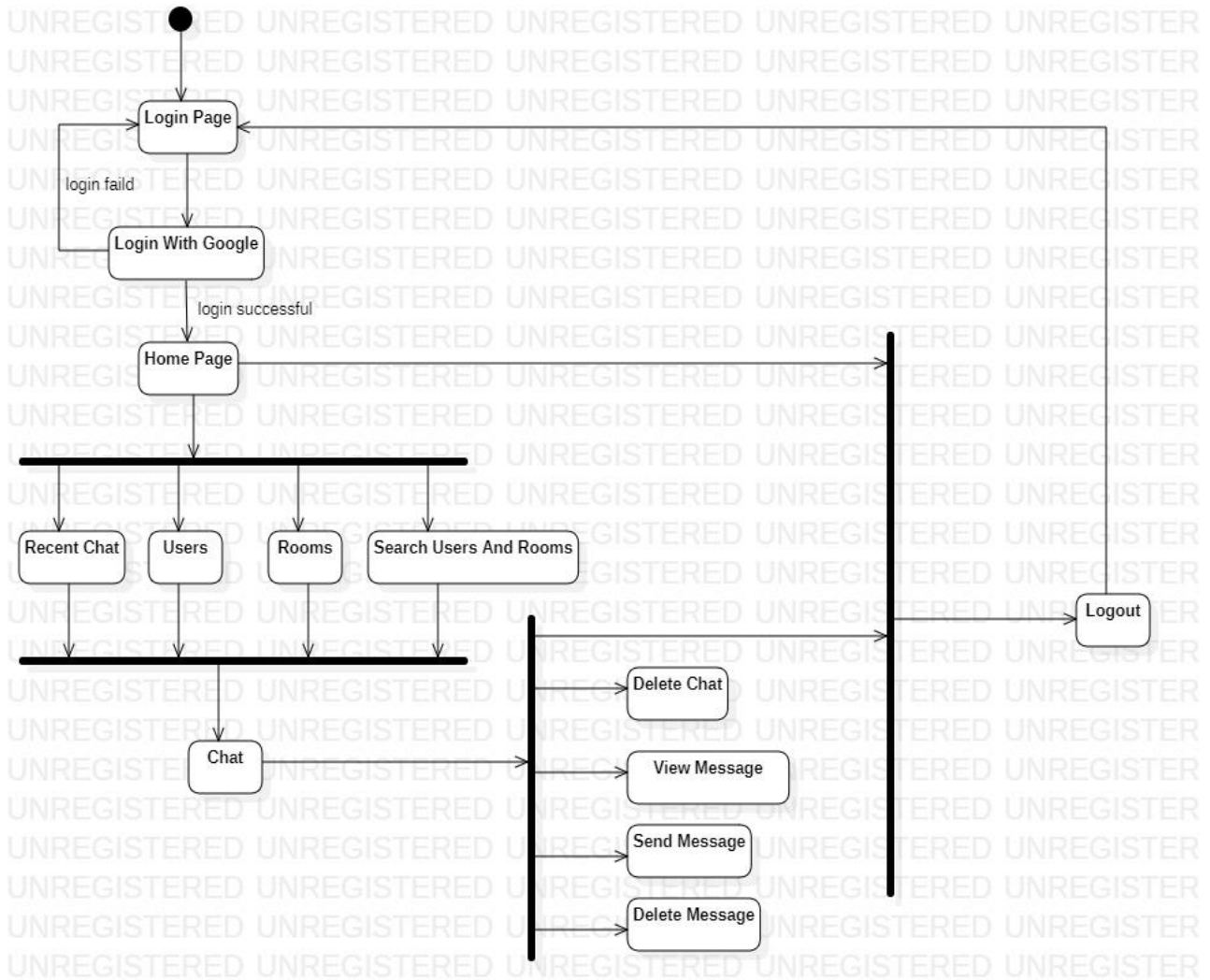


## Use Case Diagram

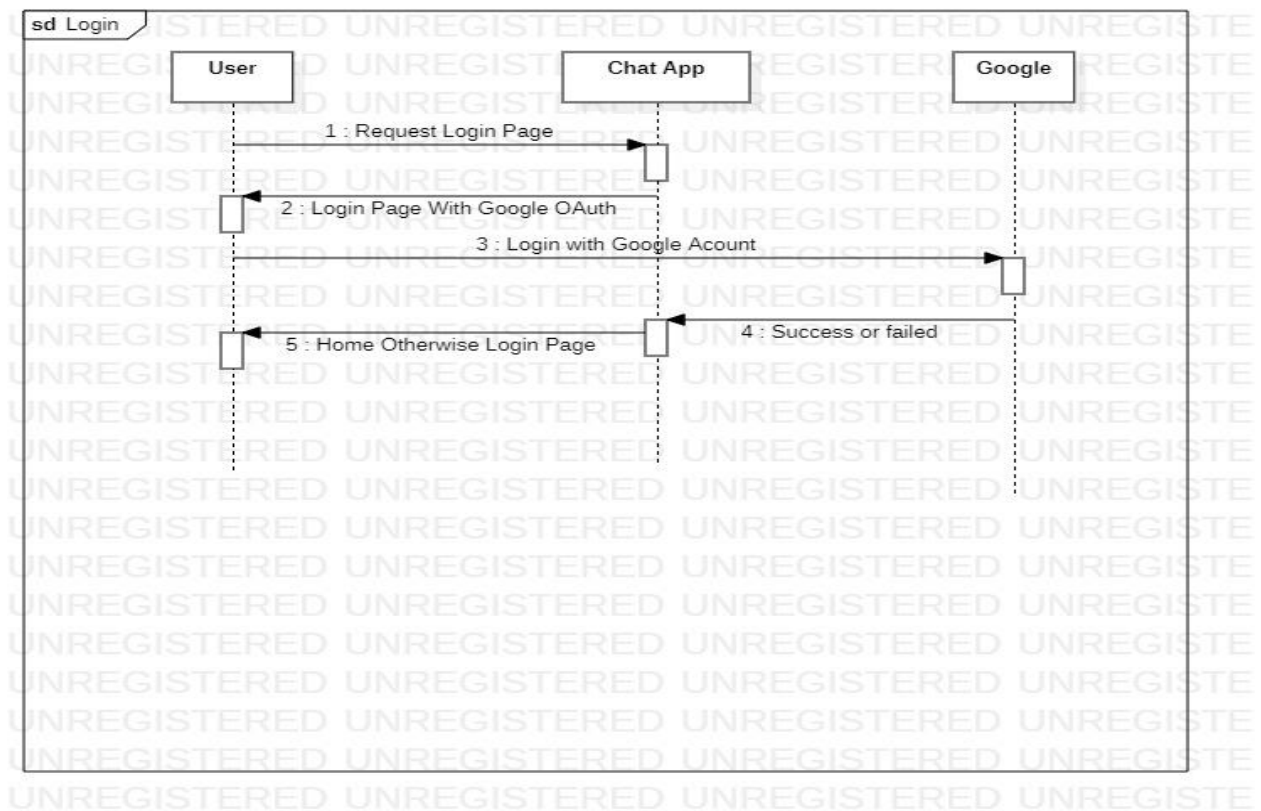
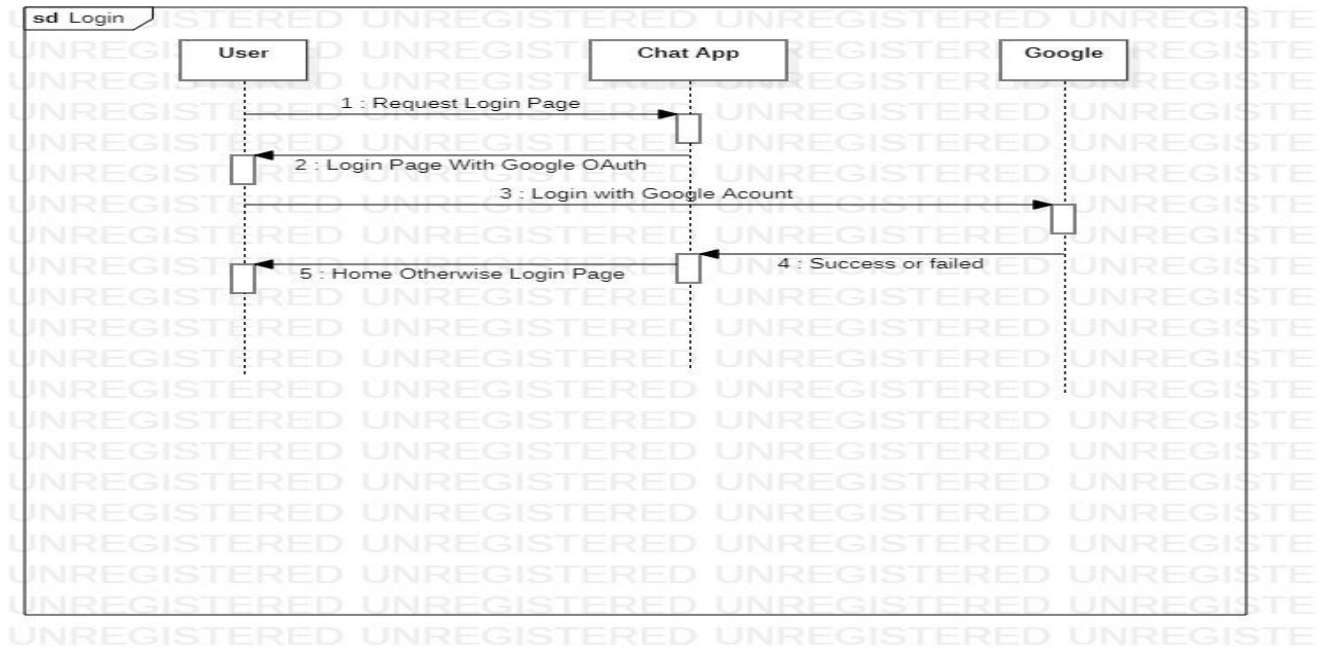




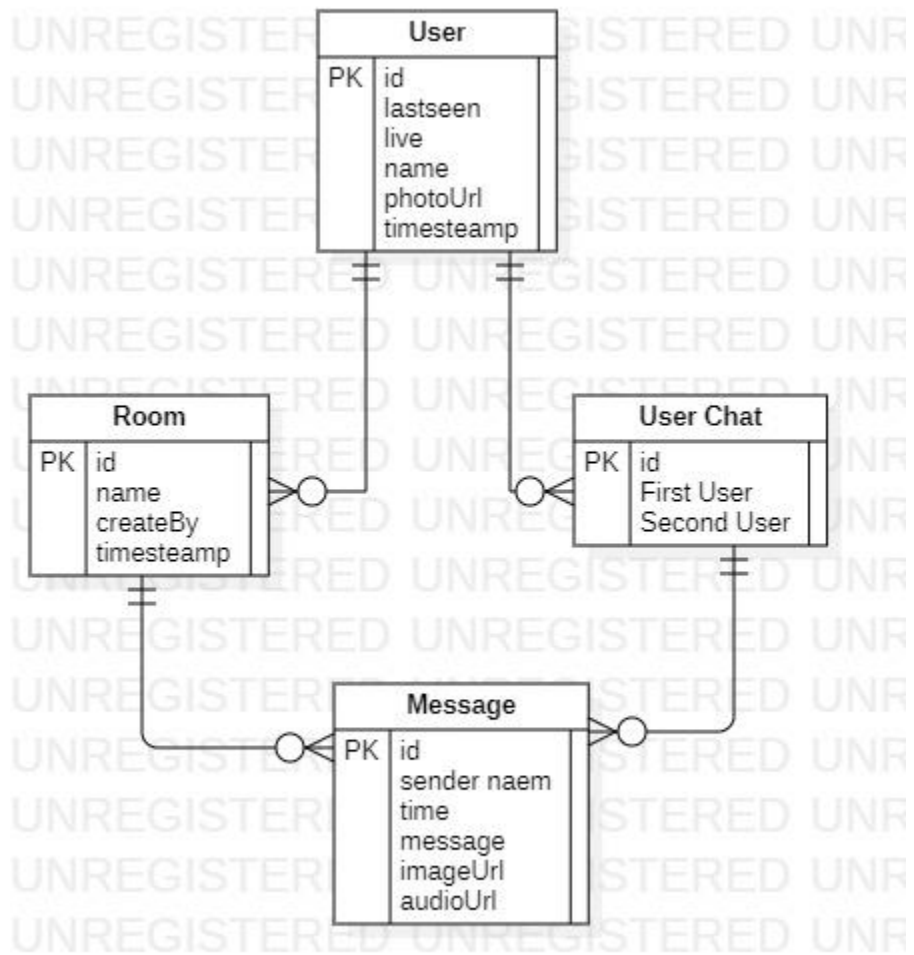
## Activity Diagram



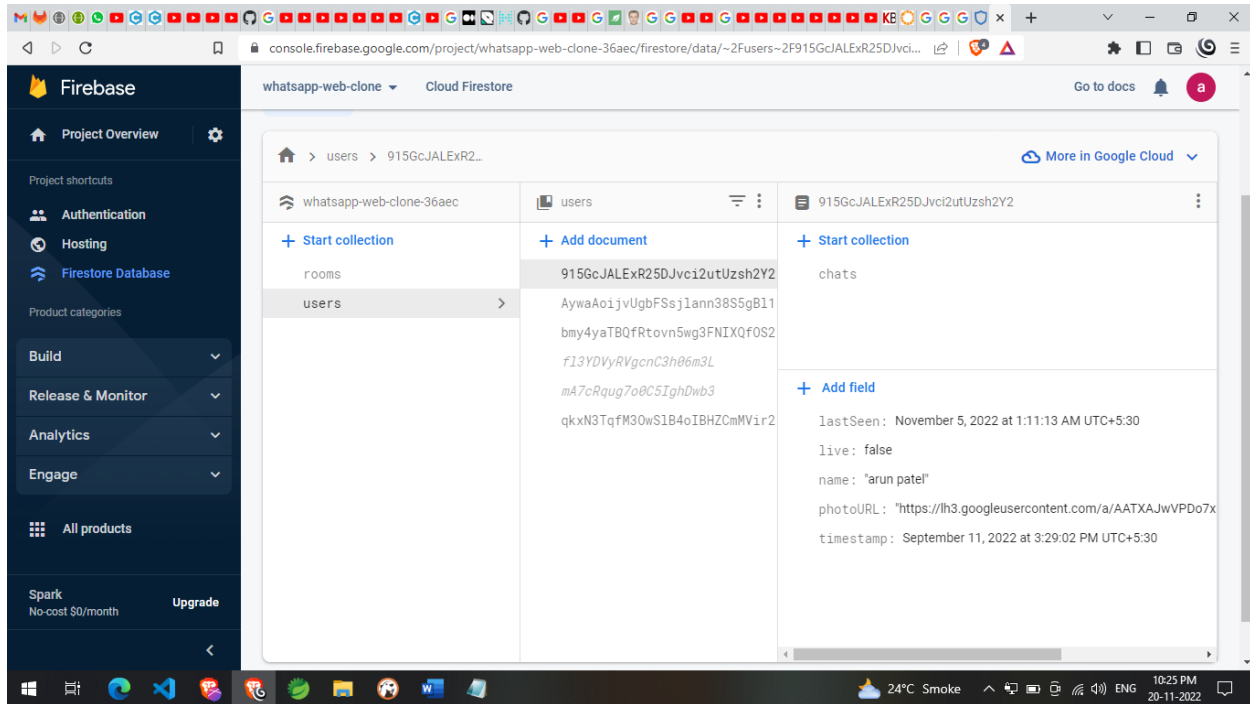
## Sequence Diagram



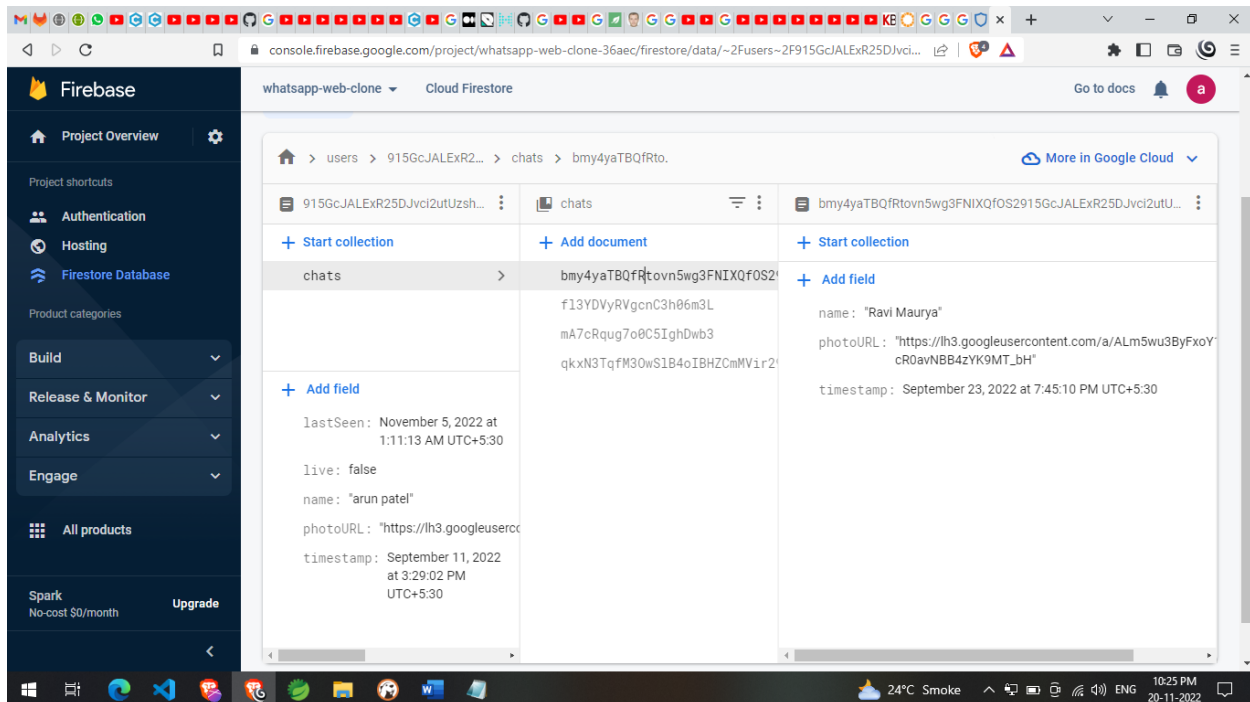
## Entity-Relationship Diagram



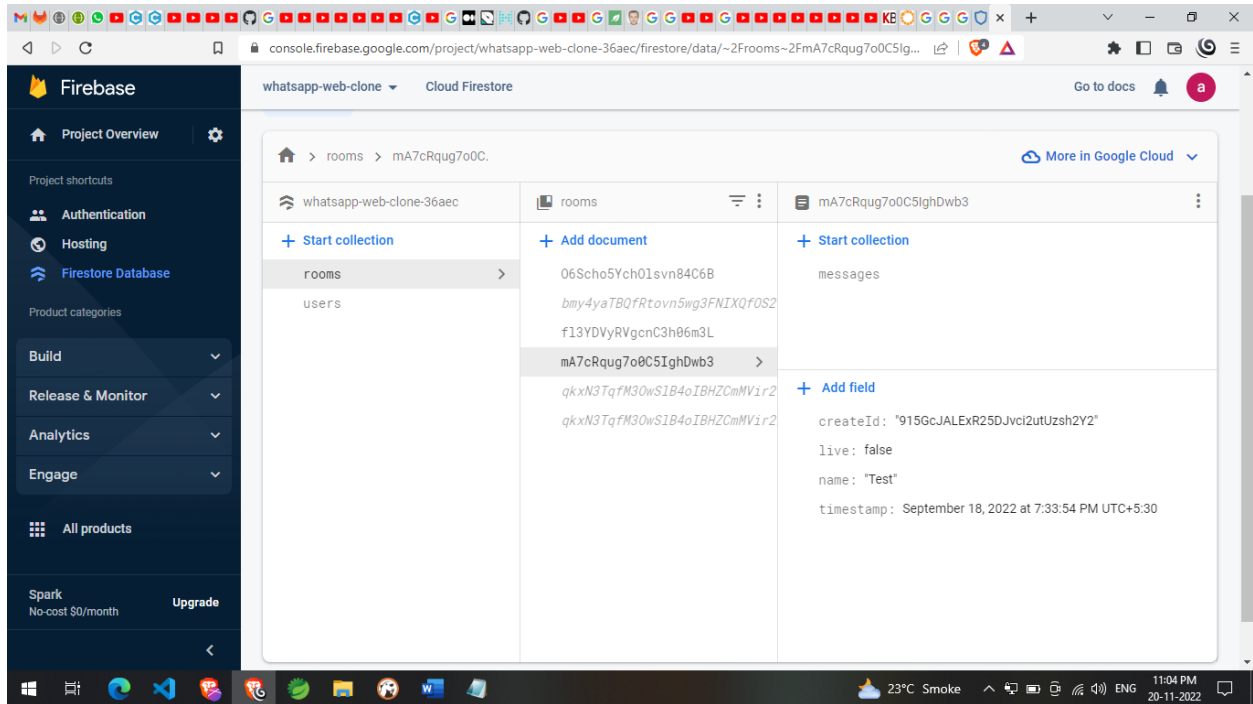
## User Collection:



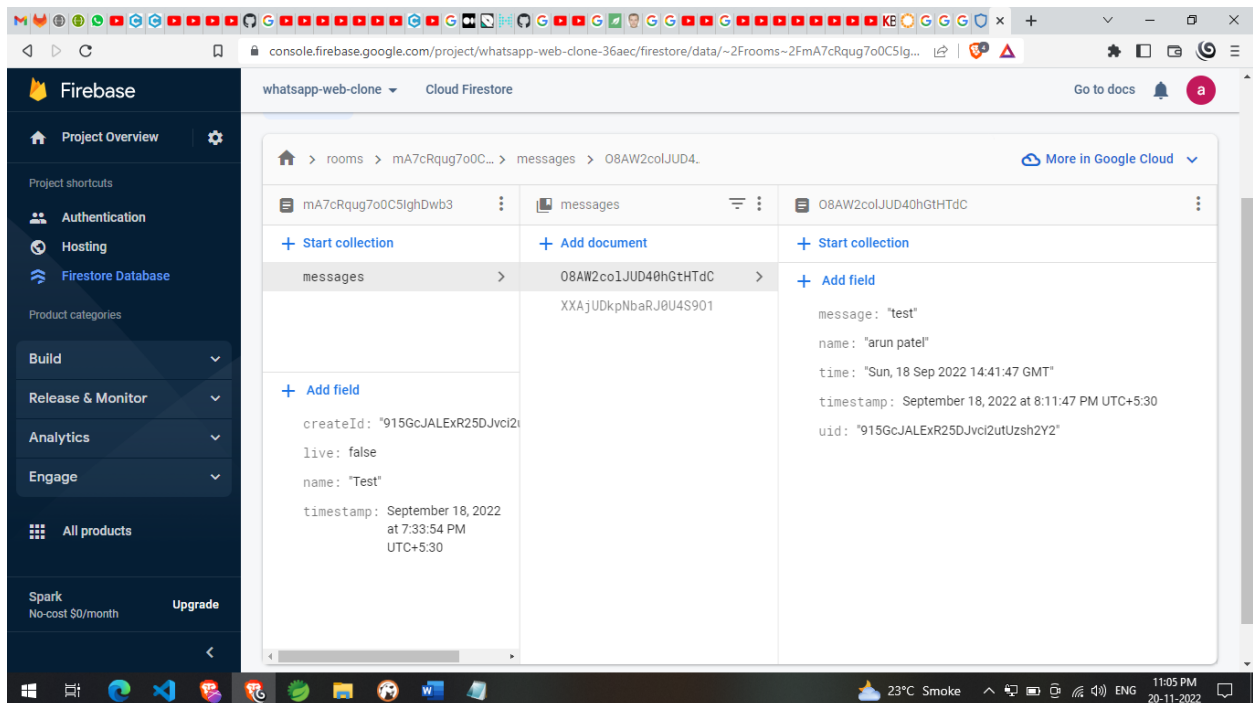
## User Chat List Collection:



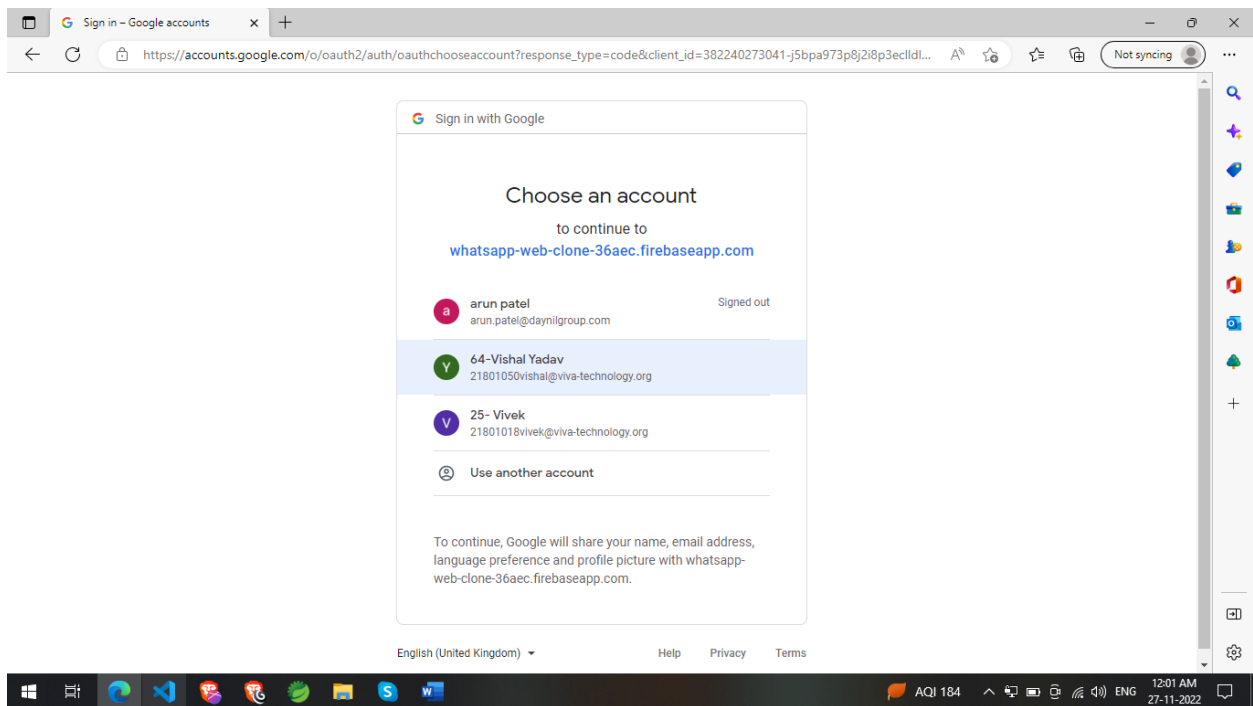
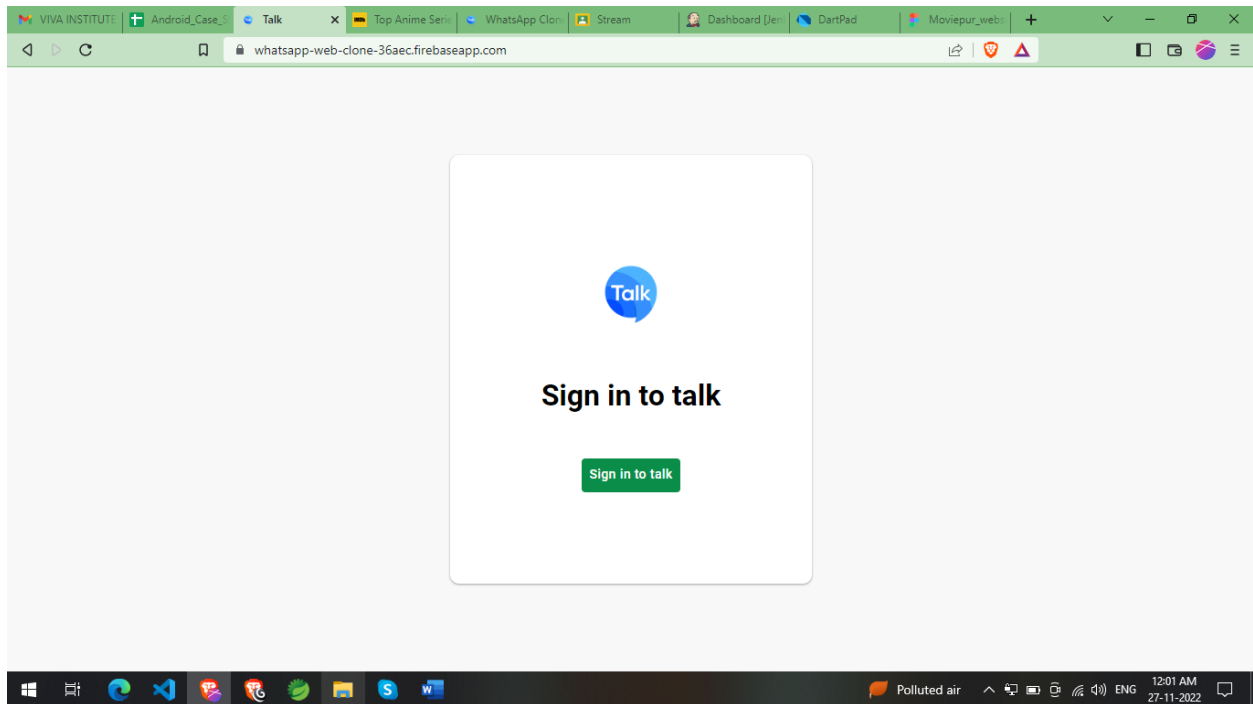
## Room Collection:



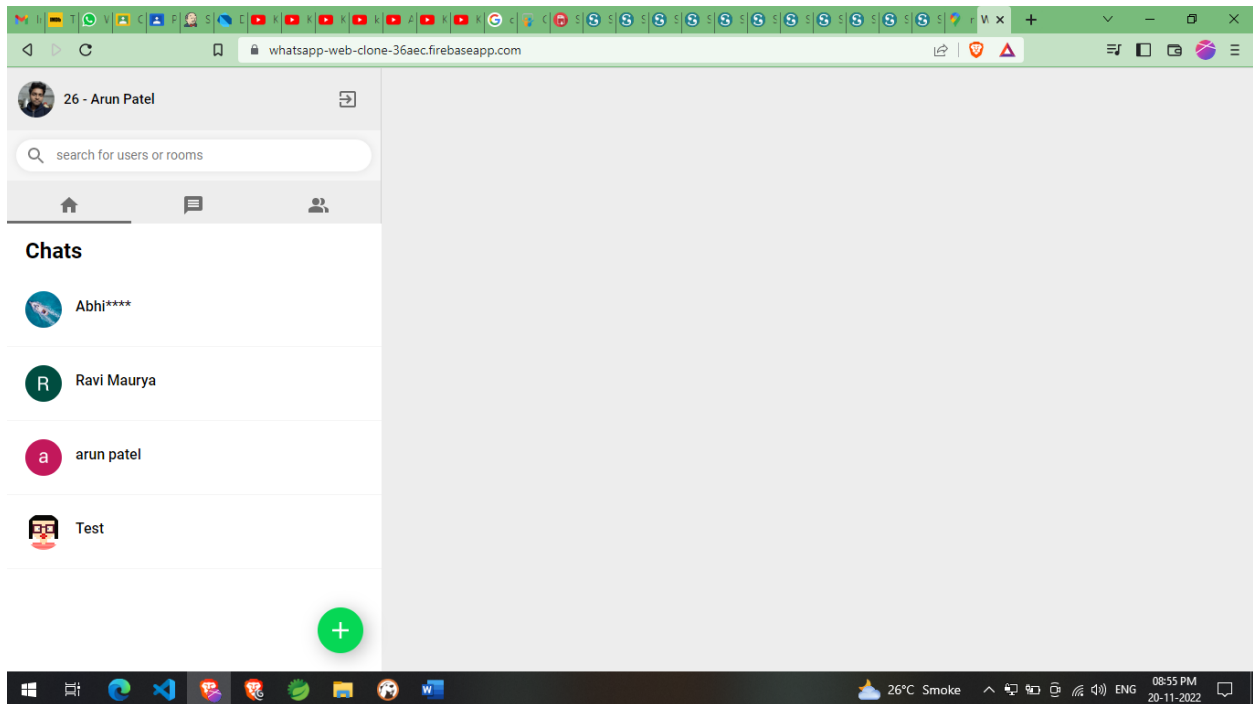
## Message Collection:



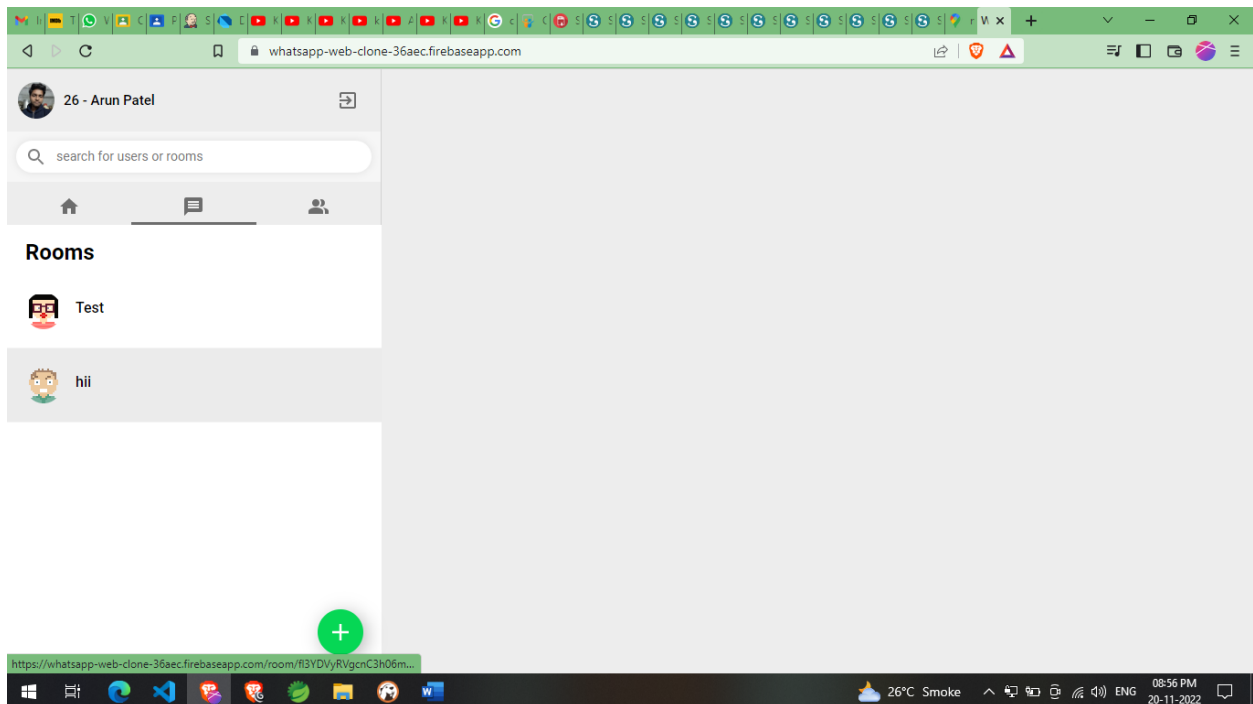
# Login



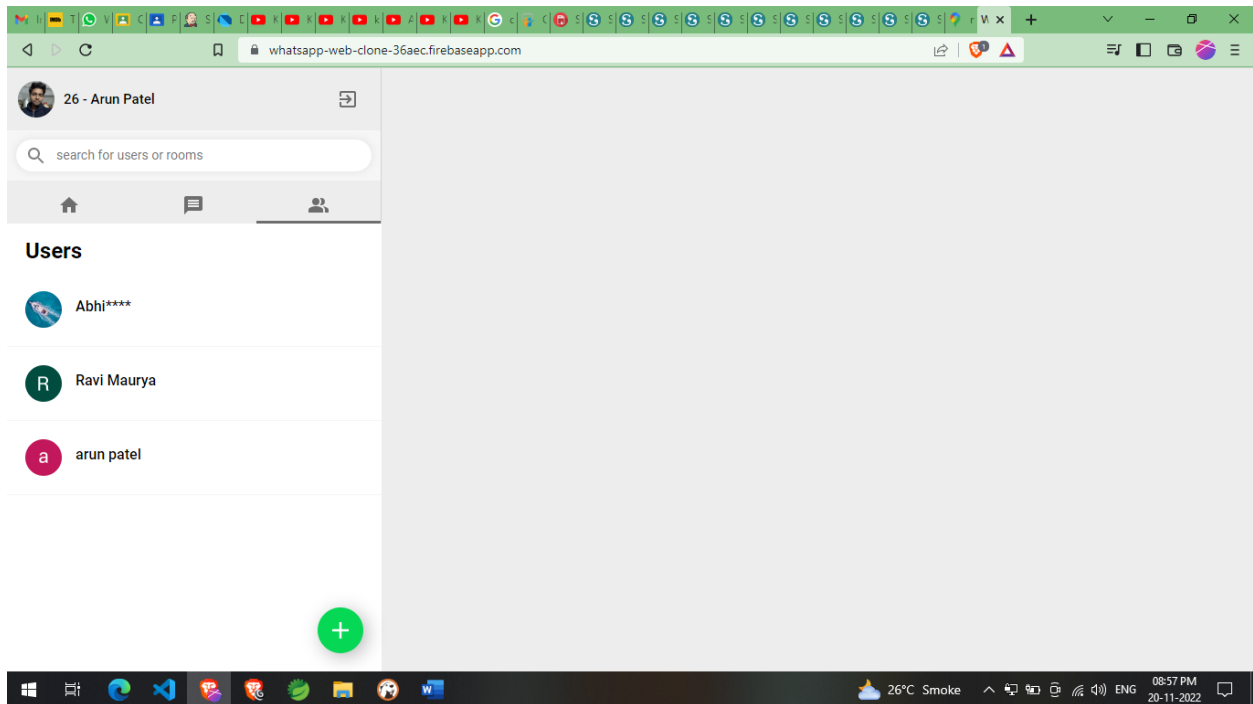
## HOME



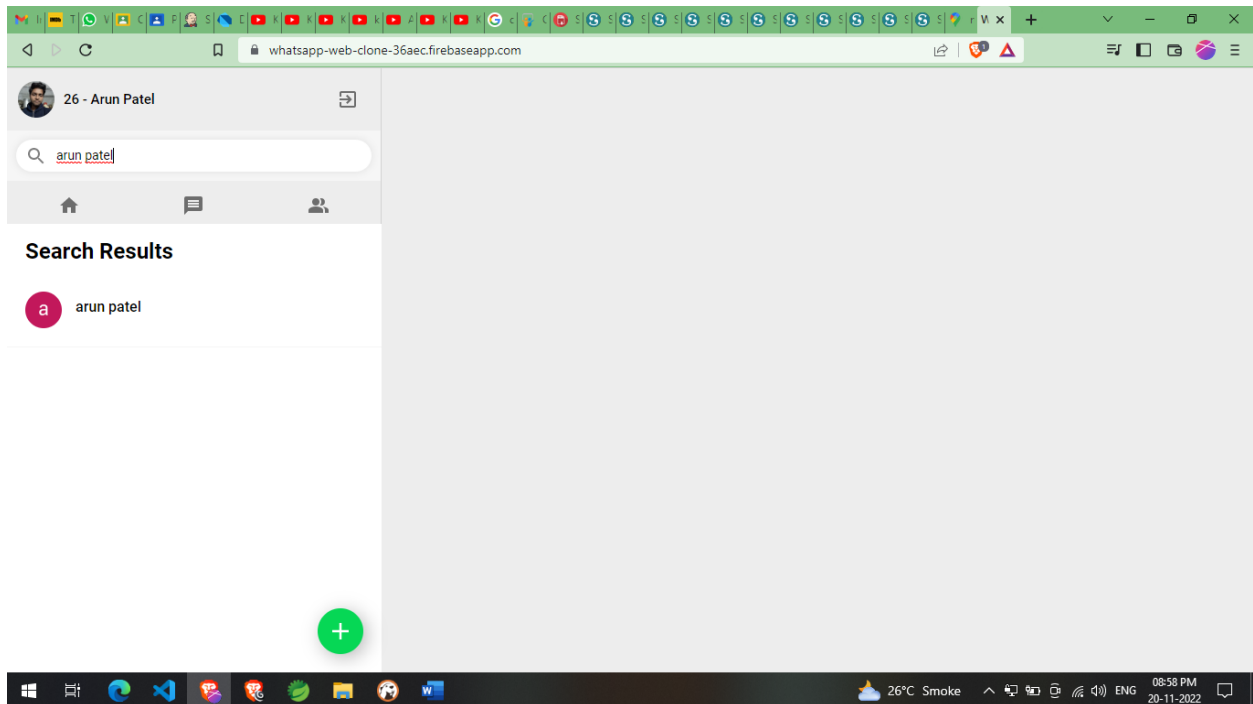
## Rooms



## Users

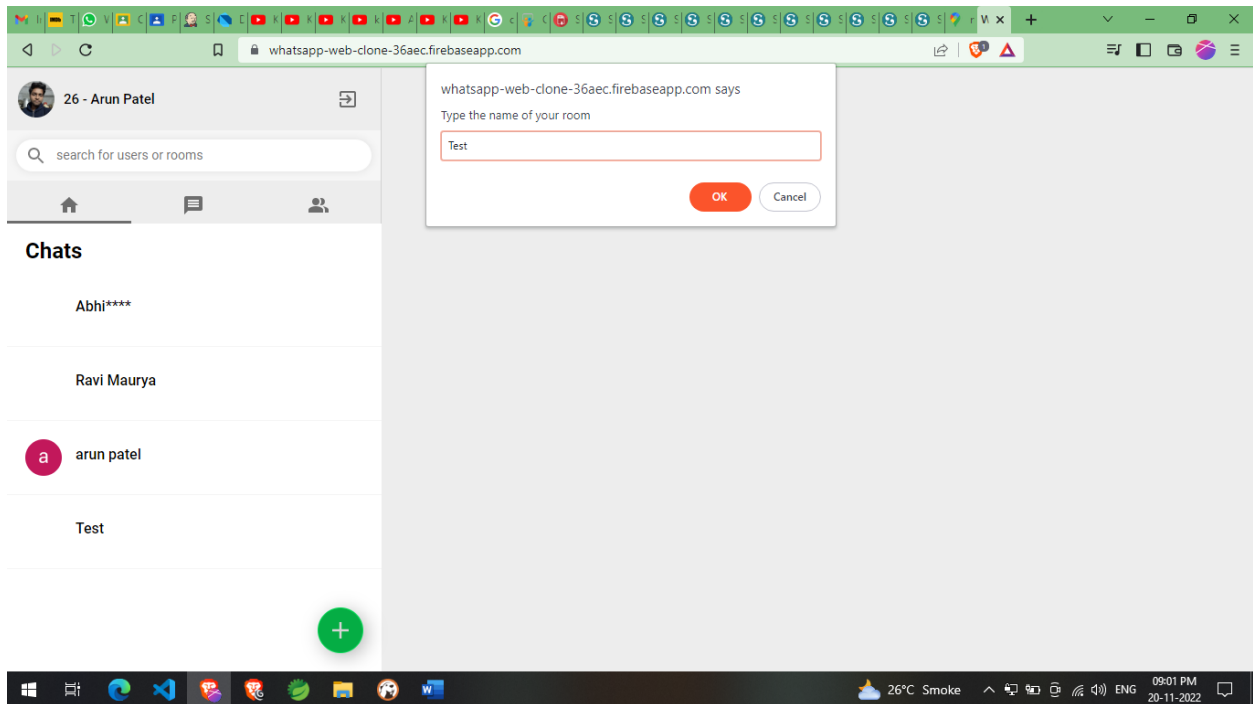


## Search

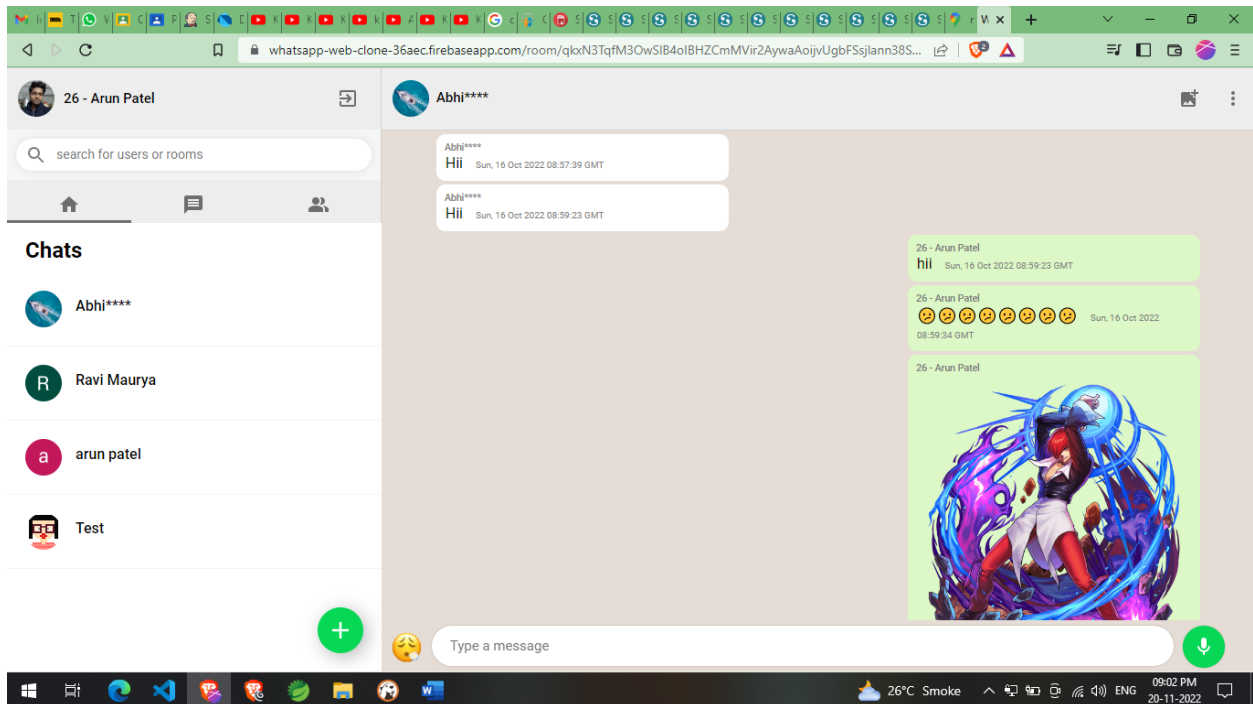




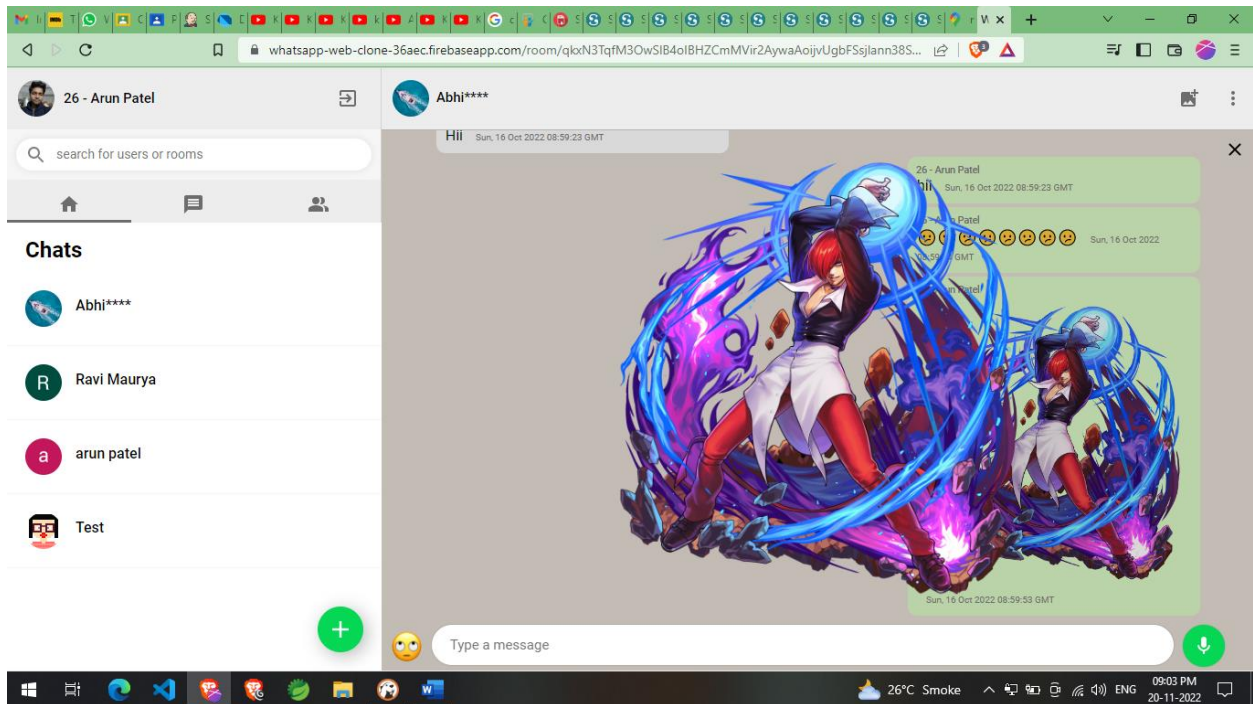
## Create New Room



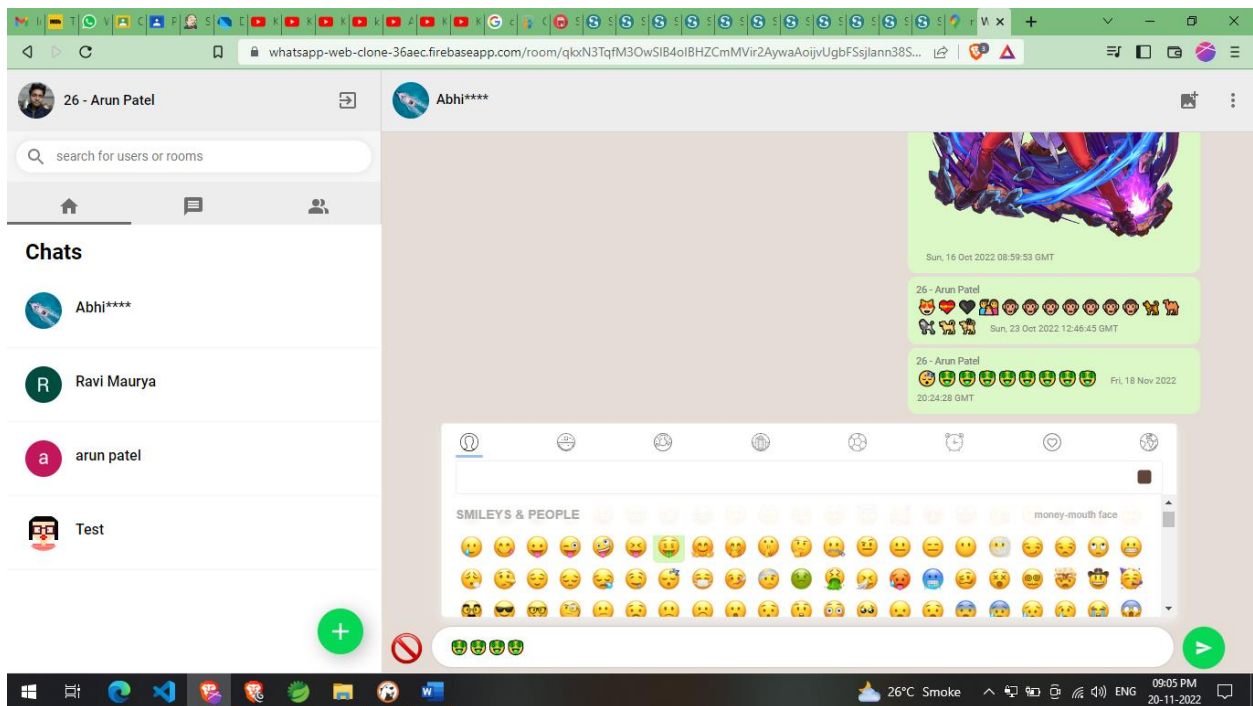
## Chat Room



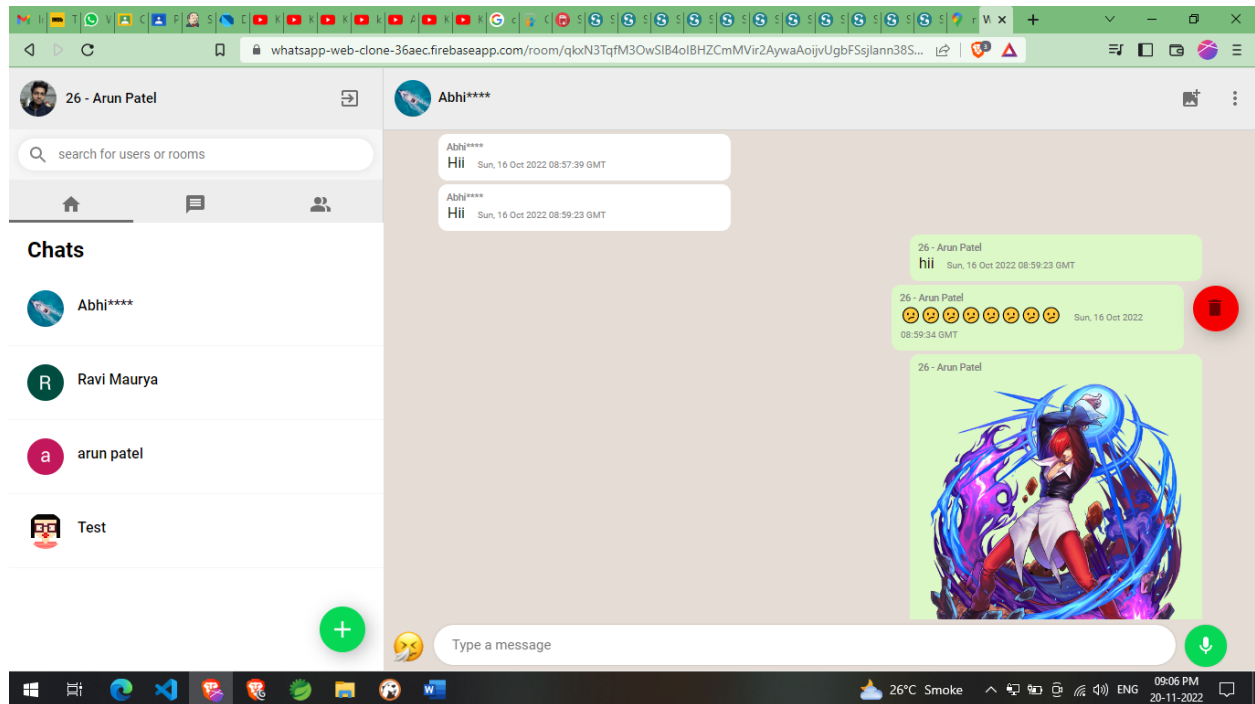
## On Image Click



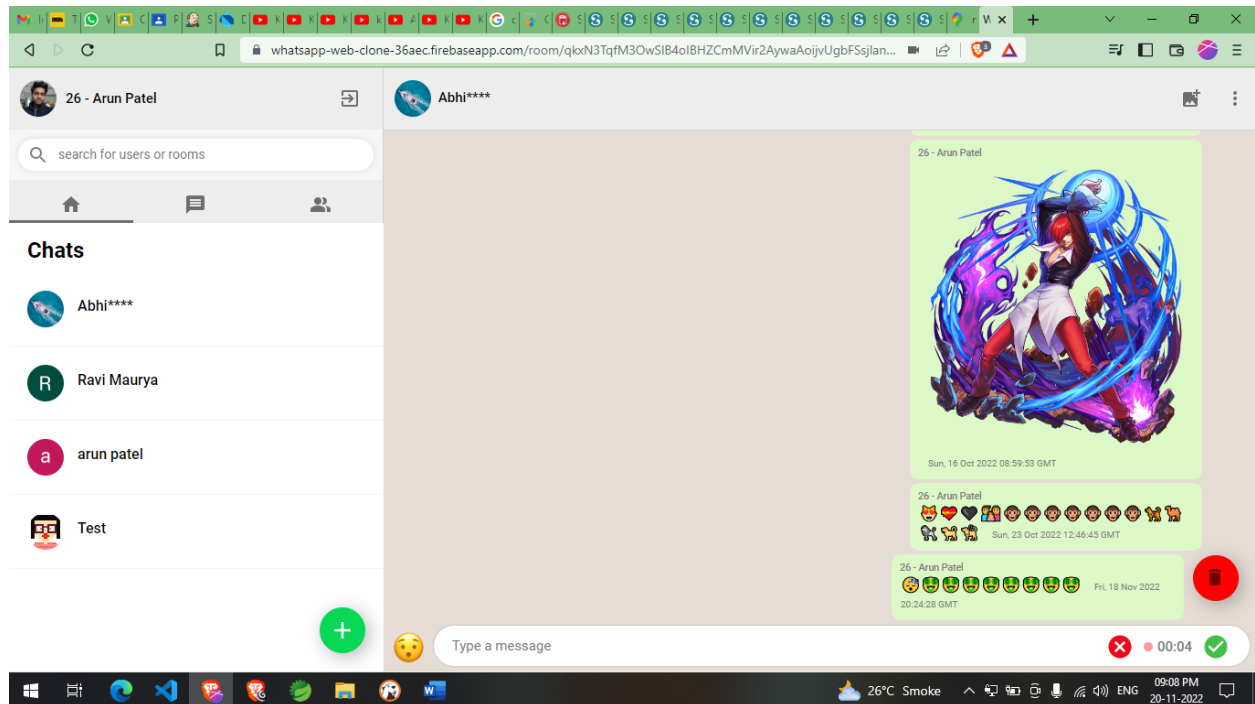
## Emoji



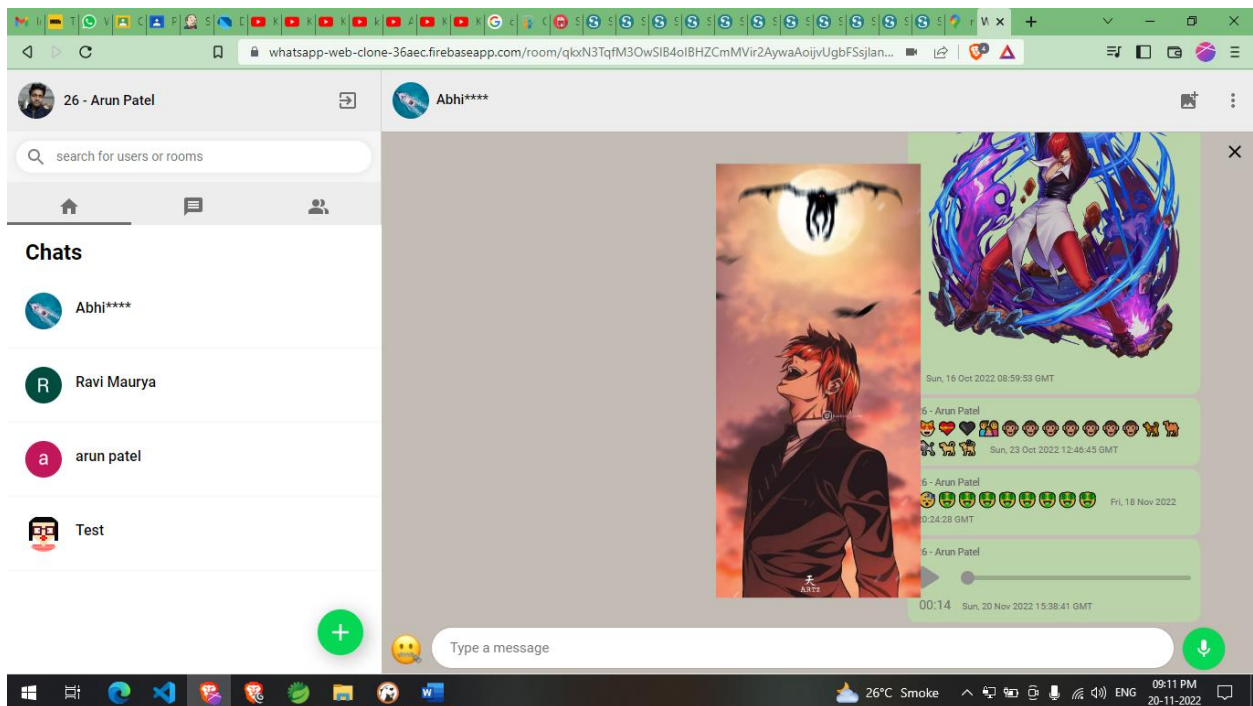
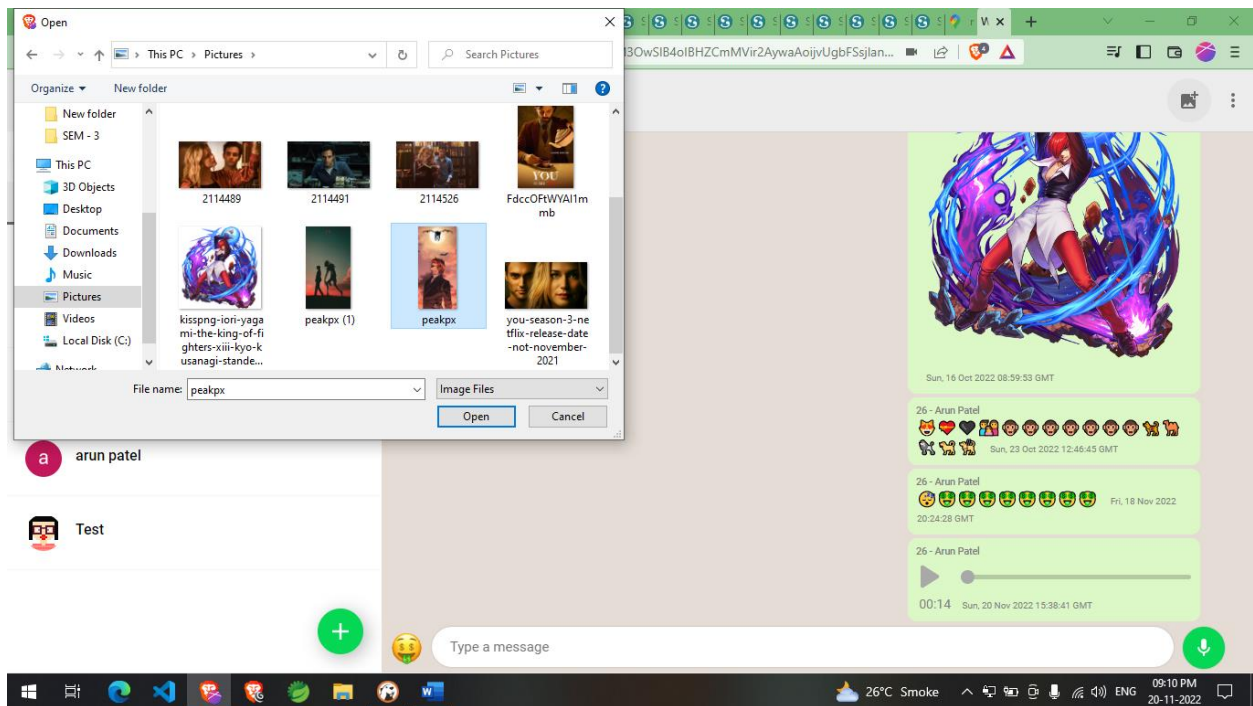
## Delete Message



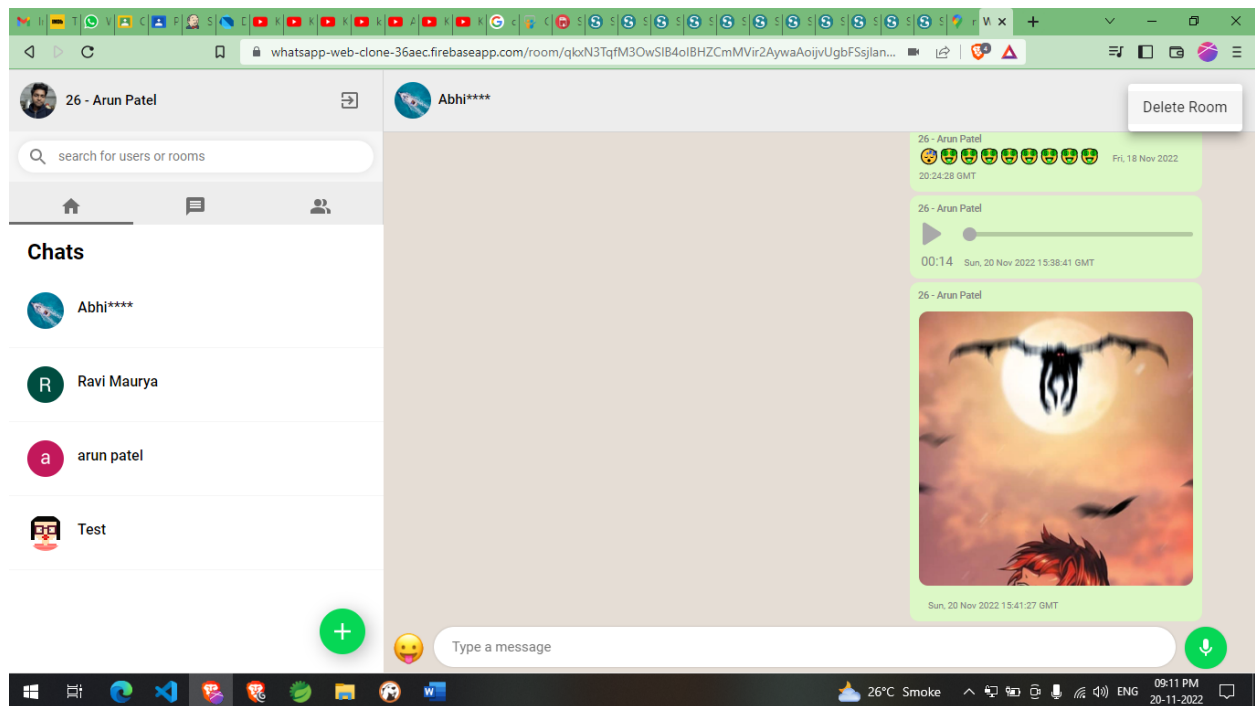
## Audio Recorder



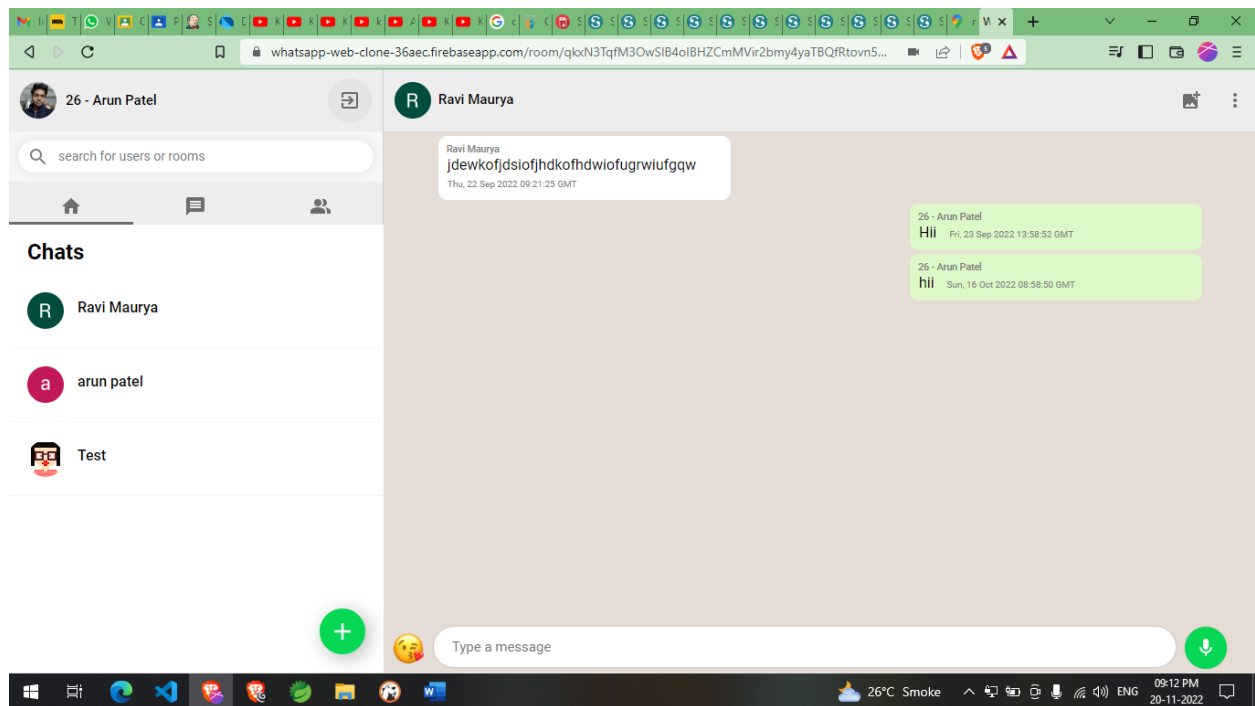
## Upload Image



## Delete Room



## Logout



# PROJECT TESTING

## Software Testing Strategies:

Testing is a set of activities that can be planned in advanced and conducted systematically. A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high-level tests that validate major system functions against customer requirements

There are Four types of testing strategies:

1. User Interface Testing
2. Integration Test
3. Performance Test
4. Manual Testing

## User Interface Testing:

User Interface testing, also known as UI testing or GUI testing, tests the application's visual elements to validate proper functionality and expected performance. It ensures that UI functions and application components have no defects, such as icons, radio buttons, text boxes, toolbars, color, fonts, checkboxes, windows, and menus. The primary aspects of UI testing are functionality, performance, usability, compliance, and visual design of the software application to make sure the application works successfully and satisfactorily.



## **Integration Testing:**

Integration testing is a systematic technique or construction the program structure while at the same time conducting tests to uncover errors associated with interfacing. Scope of testing summarizes the specific functional, performance, and internal design characteristics that are to be tested. It employs top-down testing and bottom-up testing methods for this case.

## **Performance Testing:**

Timing for both read and update transactions should be gathered to determine whether system functions are being performed in an acceptable timeframe.

## **Manual Testing:**

Manual testing is a software testing process in which test cases are executed manually without using any automated tool. All test cases executed by the tester manually according to the end user's perspective. It ensures whether the application is working, as mentioned in the requirement document or not. Test cases are planned and implemented to complete almost 100 percent of the software application. Test case reports are also generated manually.

Manual testing is essential because one of the software testing fundamentals is "100% automation is not possible."

## Conclusion

With Firebase Firestore database, we can build a simple chat website without any single line of server-side code. The main objective of the project is to develop a Secure Chat Website. The entire project has been developed and deployed as per the requirements specification; it is found to be bug-free as per the testing standards that are implemented. Any specification-untraced errors will be concentrated in the coming versions, which are planned to be developed in near future. It's developed in ReactJS, JavaScript, html and CSS and we testing the project. And project is bug-free and completely working fine. The complete system is tested and working well. This design is easy to understand and any new modules can be incorporated easily.



## **Limitation**

### **Non-Encrypted Message:**

In this exists system we are Storing the message in plain text. And it is created security issue.

### **Voice And Video chat:**

In this exists system we are support only text-based chat. we need to impalements voice and video-based chat also.

### **Video and File Sharing:**

In this exists system we are allowed only send image and voice message and we are planning to give more feature like video, gif and other document option to shearing.

## **Future Enchantments**

There is always a room for improvements in any software package, however good and efficient it may be done. But the most important thing should be flexible to accept further modification. Right now, we are just dealing with text and voice communication and Image sharing. In future, this application may be extended to include features such as:

### **Encrypted Message:**

In future we are stored message in encrypted form.

### **Voice chat:**

this will enhance the application to a higher level where communication will be possible via voice calling as in telephone.

### **Video chat:**

this will further enhance the feature of calling into video communication.

### **Video and File Sharing:**

this will enhance the application with ability to share video, ZIP or RAR Or Other files.

### **Screen share and remote access:**

this will enhance the application user can share screens and queries can be resolved from remote computers.

## References

- [www.google.com](http://www.google.com)
- [www.developer.mozilla.org](http://www.developer.mozilla.org)
- [www.w3schools](http://www.w3schools)
- [www.nodejs.org](http://www.nodejs.org)
- [www.firebase.com](http://www.firebase.com)
- [stackoverflow.com](http://stackoverflow.com)
- [fontawesome.io](http://fontawesome.io)
- [reactjs.org](http://reactjs.org)
- [getbootstrap.com](http://getbootstrap.com)
- [www.npmjs.com](http://www.npmjs.com)
- [github.com](http://github.com)