PES UNIVERSITY,
BANGALORE
Department of Computer Science and Engineering

B.TECH.
(CSE)V
SEMESTER

UE20CS303 –SOFTWARE ENGINEERING
PROJECT REPORT

WeCONNKT
Real-Time Chat Application using SocketIO

SUBMITTED BY

- Y Srinivas: PES1UG20CS517
- Vrushank G: PES1UG20CS516
- Akash S: PES1UG20CS534
- Venkatesh BR : PES1UG20CS494

GitHub Link:
vrushank41/WeCONNKT: Project Title : Real-Time Chat Application using SocketIO (github.com)

TABLE OF CONTENTS

INTRODUCTION

Messaging is more widely and frequently used than even social media. Online chats have become an eminent part of both our daily life and business environment. Instant messaging is faster and more convenient than email and less stressful than a phone call, so there is no wonder why we use chat apps so often.

People around the world are used to chatting online about everything, including discussing the latest episode with a friend, organizing a team-building meeting, or asking your favourite coffee shop's chatbot about new flavours.

Numbers and statistics are also on the side of chat apps. The overwhelming majority of studies show that instant messaging is becoming more and more popular among users worldwide.
We will be using express to build the web server that Socket.IO will work with. Any other node-server-side framework or even node HTTP server can be used. Express JS makes it easy to define routes and other things. Socket.IO enables real-time bidirectional event-based communication. It works on browser, focusing equally on reliability and speed. Socket.IO is built on top of the WebSockets API (Client side) and Node.js. It is one of the most depended upon library on ***npm*** *(Node Package Manager) . The chat server's resources can include a REST API, a WebSocket server, MongoDB Atlas for media storage and cloud platform, etc.*

Software Requirement Specification

Functional Requirements :

1. User Registration

User must be able to register for the application through a valid phone number. On installing the application, user must be prompted to register their phone number. If user skips this step, application should close. This is of the highest priority as this will help us in knowing about the user and its detail will be stored in the database .

2. Adding New Contacts

The application should detect all contacts from the user's phone book.

3. Send Message

User should be able to send instant message . User should be notified when message is successfully delivered to the recipient .

4. Send Attachments

User should be able to send audio, video and images as attachments. Specifying of what file type can be attached or not should be made clear .

5. Broadcast Message

User should be able to create groups of contacts. User should be able to broadcast messages to these groups.

6. Message Status

User must be able to get information on whether the message sent has been read by the intended recipient.

The above are some of the basic functionalities of any Chat-Application and hence are of the highest priority in our " Real Time Chat Application Using SocketIO " .

Some other Features include :

1. ENCRYPTION & ONLINE PRIVACY

Starting from terabytes of data sent via your messenger, and ending with confidential data of the large corporations which are loaded, transferred and stored every single day thus, respect for privacy and adequate data protection are critical. Here the End-To-End encryption appears. The initial requirements of the app define Encryption & Online Privacy. Chat programs frequently change their cryptographic keys for further

protection. A true backdoor for hackers might be created by a messenger for a bank or cryptocurrency messenger.

2. CLOUD & DATA SYNCHRONIZATION

Cloud Services are hardly new features which enables storing your files in different places up-to-date. All other devices are updated automatically when you make a modification to a document on one device. As a result, conversation history and data supplied by users may be retrieved at any time and from any location.

3. BOTS

These are small special features of instant messaging software that are embedded in chats or public channels to perform a specific function.

What is its purpose?

- *Self-destructing statuses may be used to manage communities and blogs.*
- *Automated assistance with registration and other concerns.*
- *Custom created chat bots using the open-source frameworks, etc.*

4. LIGHT MODE AND DARK MODE

These are not only desirable features of good message passing system. Additionally, real improvements in battery life may be gained. Google claims that accessing YouTube in Dark Mode saves batteries by 14 percent when the screen brightness is set to 50 percent. With the screen brightness set to 100 percent, the saving jumps to 60 percent. The harsh bright light of a smartphone can cause discomfort when viewed in a dark room, and the muted aesthetic of a dark UI will not show up as well in a brightly lit area. Even yet, designing the user interface is a creative and aesthetic endeavor in and of itself.

5. DATA PERSISTENCE FOR OFFLINE USAGE

WiFi and mobile Internet eat your battery. Application with full support of offline mode is escalating at a higher rate. Creating a safe and secure environment that can be used by individuals of all ages and backgrounds is at the heart of the mission statement of these applications. Developing an application that will work effectively even in offline mode or adopt the offline capability will allow people to have smooth experience when the connection is low, slow, flickering or not working. TYour app's offline mode offers several benefits:

- *No roaming cost when you are traveling as all the files can be cached*
- *No monthly data usage on maps*
- *Quick loading time*

Due to time complexity, we will try our best to cover on all the features .

Other Nonfunctional Requirements

Performance Requirements

PR-1: RESPONSE TIME: The software system shall likely show no deterioration in accessing the required information about the user information and his/her previous records. Any major delay might be possible because of some incompatibility in the socket implementation which might take the software longer than usual time to respond.

PR-2: LOADNIG SPEED : The software has a fairly less probability of taking a unusually longer time to run . There might be some productivity tools or file issues or some problems related to the environment on which the system is running, which may lead to some lags and loading issues.

Safety Requirements

Real-time message synchronization between a user's apps, persistent chats, reliable push notifications for all apps.

Customer data sovereignty, data economy and minimal data usage, no storage of address books, anonymized user IDs (with secure algorithms), data is only processed and stored in the customer's country, data retention guidelines,

Direct integrations to third-party software and services, secured access under administrative control, automation of communication

Security Requirements

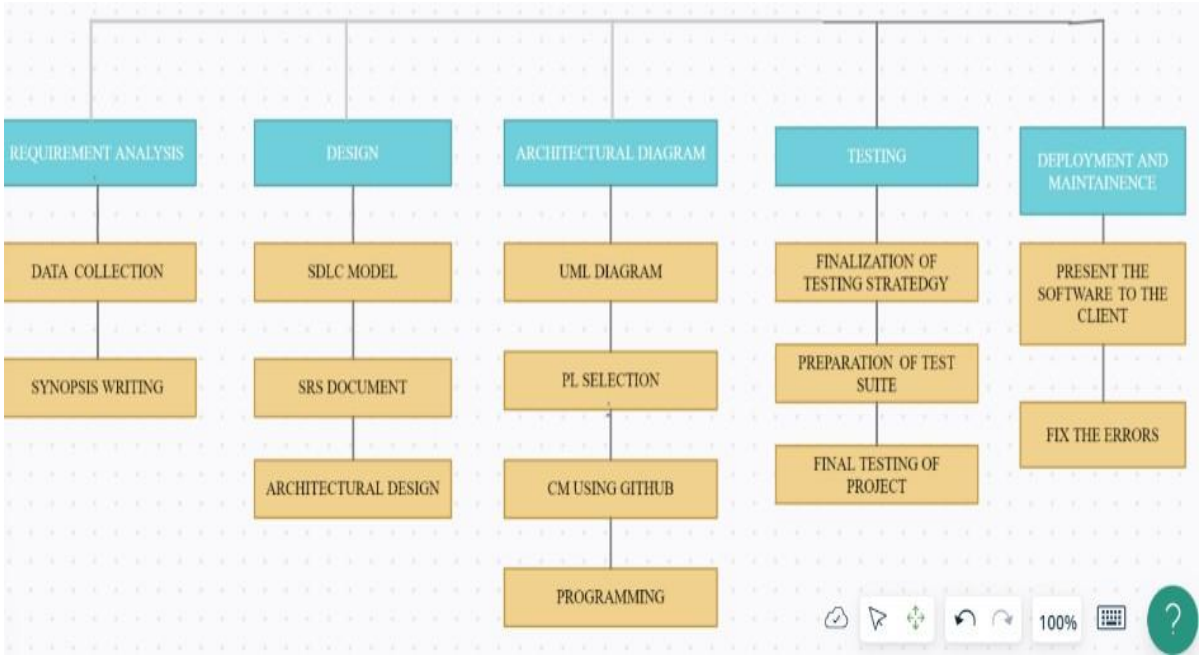
- ***End-to-end encryption**:-have a secure messaging via end-to-end encryption in communication, ensures the encryption of data that has been transferred during the communication. This is from the moment a message has been typed to the time it reaches the receiver, here no one else can be able to view the data, either the app maker or internet service providers or government organizations ensuring security.*
- ***Multi-mode communication**:-Getting bored by communicating with text messages, you can also share media. Whenever a platform allows the users to send across messages through multiple modes, it automatically permits the user to delete the messages if they wish to, enhancing the security.*
- ***Supportive to Multi-platform**:-You can be able to synchronize the messages across any platform like web app. However, it allows the user to store their private messages in any of their desired locations and also permits them to delete from other locations, if they are not feeling comfortable. Thus, this adds up a great impact to one's data privacy ensuring security.*

PROJECT PLAN

The work was equally distributed using the WBS and multiple sprints were conducted with frequent scrum meeting tracking and making sure the Gantt chart was followed to maintain the schedule .

Plan of work and product ownership

- *Developing back-end website applications and seeing through a project from conception to finished product-SRINIVAS Y*
- *Ensuring responsiveness of applications, working alongside graphic designers for web design features-VRUSHANK G*
- *Developing front-end website architecture and designing user interactions on web pages - VENAKTESH BR*
- *Specialize in developing and maintaining the server and the technical side of website development Creating servers and databases for functionality-AKASH S*



WBS DICTIONARY:

The WBS Dictionary contains all the details of the WBS which are necessary to successfully complete the project. Most importantly it contains a definition of each Work Package which can be thought of as a mini scope statement.

Level	WBS Code	Element Name	Definition
1	**1**	**WeConnkt**	**All work to implement our project.**
2	**1.1**	**Initiation**	**The work to initiate the project.**
3	1.1.1	Develop Project Charter	Develop the Project Charter.(consisting of objectives, detailed deliverables)
3	1.1.2	Submit Project Charter	Project Charter is delivered to the Project Sponsor.
3	1.1.3	Project Charter Signed/Approved	The Project Sponsor approves the Project Charter to move to the Planning Process.
2	**1.2**	**Planning**	**The work for the planning process for the project.**
3	1.2.1	Create Preliminary Scope Statement	Project Manager creates a Preliminary Scope Statement.
3	1.2.2	Project Team Meeting	The planning process is officially started with a project kickoff meeting .
3	1.2.3	Develop Project Plan	The Project team develops the project plan.
3	1.2.4	Submit Project Plan	Project team submits the project plan for approval.
2	**1.3**	**Execution**	**Work involved to execute the project.**
3	1.3.1	Verify & Validate User Requirements	The original user requirements is reviewed by the project manager and team, then validated with the users/stakeholders. This is where additional clarification may be needed.
3	1.3.2	Design System	The technical resources design the new widget management system.
3	1.3.3	Procure Hardware/Software	The procurement of all hardware, software and facility needs for the project.
3	1.3.4	Install Development System	Team installs a development system for testing and customizations of user interfaces.

2	**1.4**	**Control**	**The work involved for the control process of the project.**
3	1.4.1	Project Management	Overall project management for the project.
3	1.4.2	Risk Management	Risk management efforts as defined in the Risk Management Plan.
3	1.4.3	Update Project Management Plan	Project Manager updates the Project Management Plan as the project progresses.
2	**1.5**	**Closeout**	**The work to close-out the project.**
3	1.5.1	Audit Procurement	An audit of all hardware and software procured for the project, ensures that all procured products are accounted for and in the asset management system. Documents the lessons learned for the project.
3	1.5.2	Update Files/Records	All files and records are updated to reflect the widget management system.
3	1.5.3	Gain Formal Acceptance	The Project Sponsor formally accepts the project by signing the acceptance document included in the project plan.
3	1.5.4	Archive Files/Documents	All project related files and documents are formally archived.

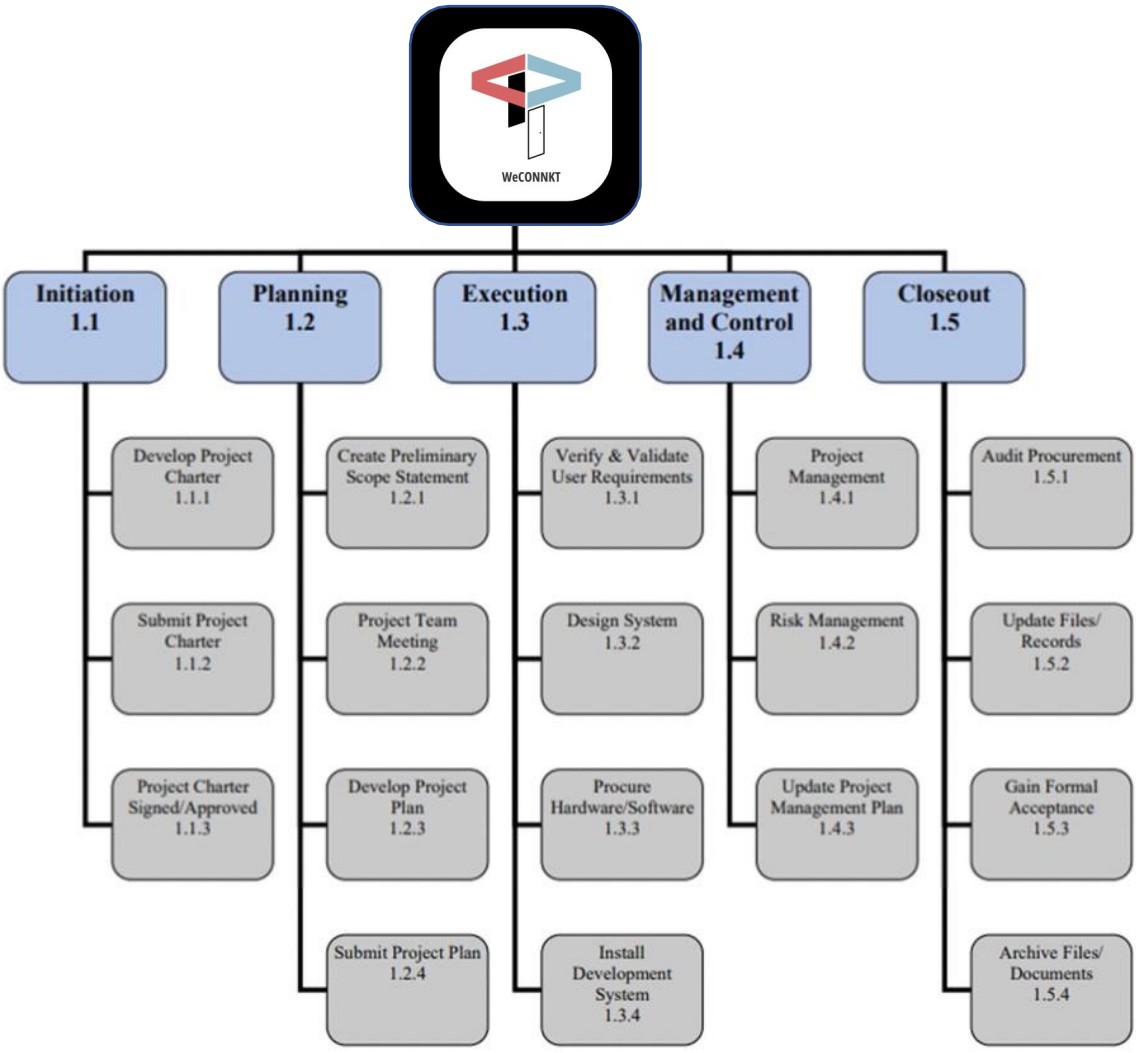
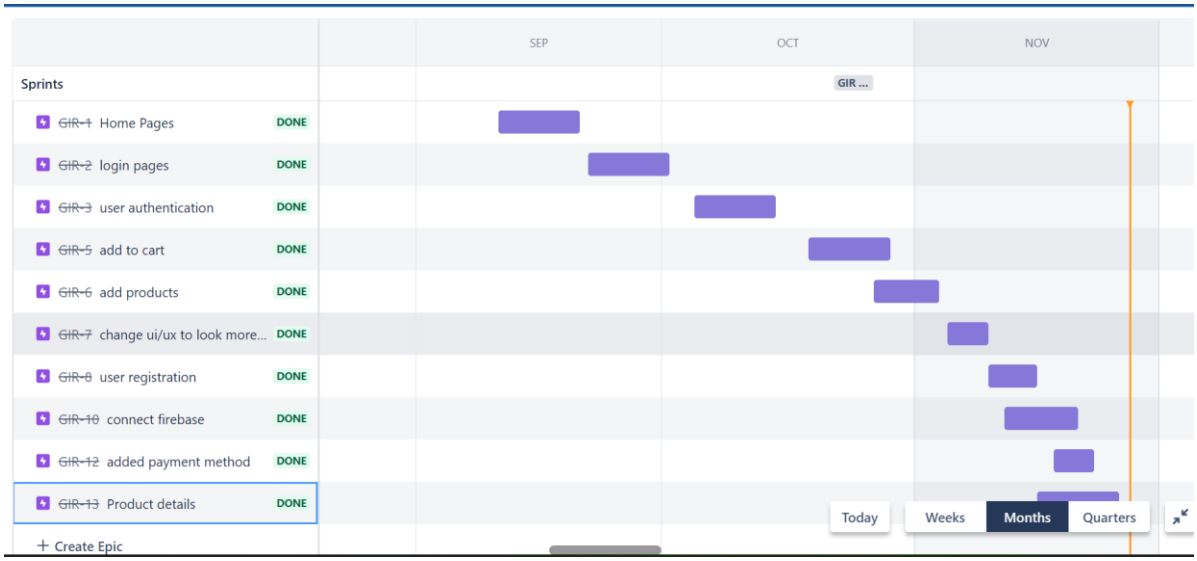


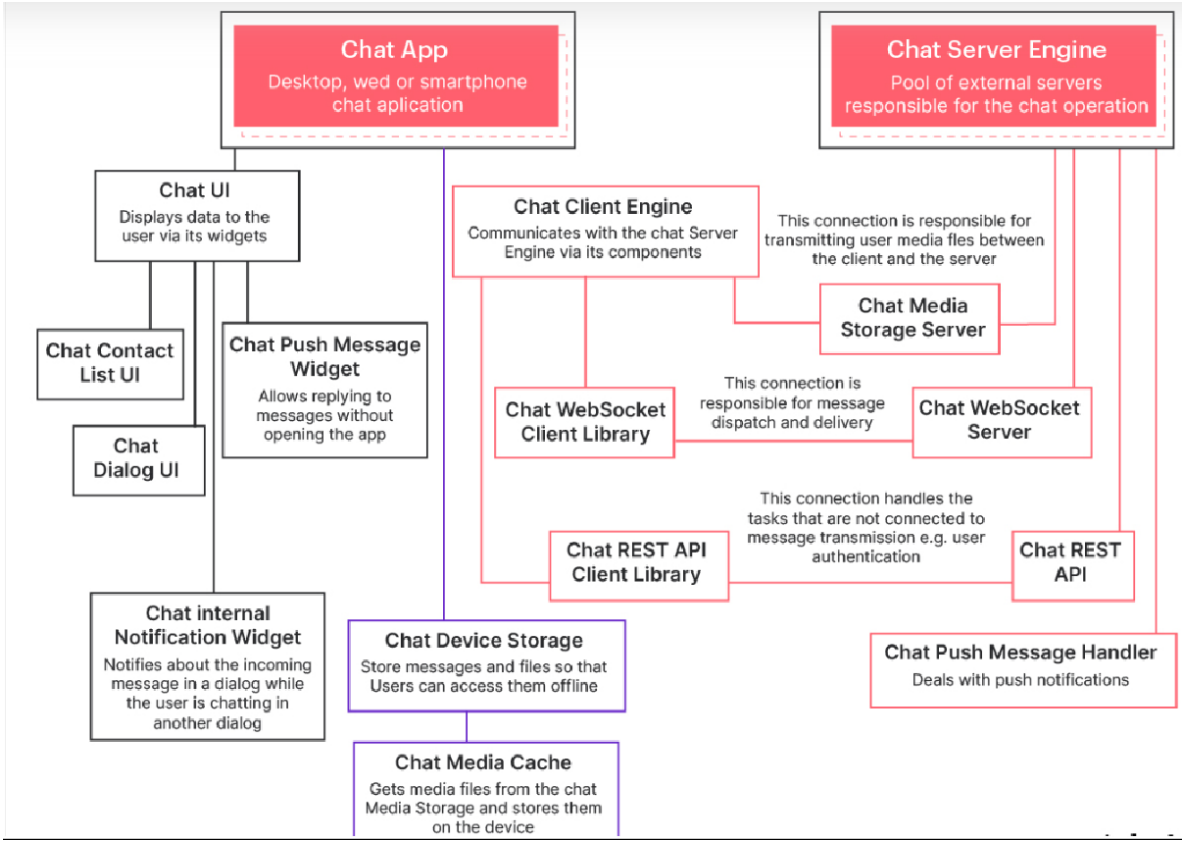
Figure2: Gantt chart



We have used tools like

- JIRA : tracking project development and Agile project management
- Selenium : Unit testing of the web application

DESIGN DIAGRAM



TEST CASES

Test Case ID	Name of Module	Test case description	Pre-conditions	Test Steps	Test data	Expected Results	Actual Result	Test Result
UT_01	Importing intializeApp getStorage,	Checking whether mongodb cloud server is connecting properly	All these libraries should be installed in the project path.	Execute the basic code for all these connection . and check whether they are	Console .log(err) No I/P test data, only code	All imports must be successful No errors must be generated stating	Successfully imported all modules	PASS
UT_02	Check for login succesfull	On click on button it should login the user with his/her credentials	Mongodb should be connected	Console the output of product	Username and password inputs	Successful Login	Logged with right details	PASS
UT_03	Check for user registered	User gives his/her details , He/She must be able to create an account in website.	Mongodb connected	User gives his/her details. This code cell is executed	Register using the asked details for future transactions	a)start the server b) signUp with user information	If user not present ,allow user to create account.	PASS
UT_04	Chat container	Test if code is able to generate the existing chat details	Log details of user and his/her chats	Login first to see the chats transaction details	Text inputs in the message box.	Successfully see all the prevailing details in order	Showing everything	PASS

UT_05	Set Avatar and emoji selection box	Check whether emoji box opening or not	First login and select the message container and opt for Emoji selection	Select emoji container and choose the one needed	Emoji in the selectbox option	Add the selected emoji to the message box container	Concatenates the emoji to the previous already written data in message container box	PASS
UT_06	Logout	Logs out of the current account	Logged out	Click the logout button on the interface	Emit event on logout button	Should logout of current account	Successful logout	PASS

CODING PRACTISES

THESE ARE THE CODING PRACTISES THAT WE HAVE FOLLOWED INOUR PROJECT IN ORDER TO MAKE IT EASY TO UNDERSTAND AND MAXIMIZE CODE PORTABILITY

- For loading fronend features we used vite+react and for backend integration in nodejs,expressjs along with mongodb cloud.
- Followed rules for usage of global variables.
- Reusing code via modularity.
- We have not used any "GOTO" statements.
- Code is written in such a way that it is easy for the testing teamto test and debug the program.
- Followed naming conventions LIKE CAMEL CASE whenevernecessary.
- Added comments for better reading and understanding.
- Logically grouped together codes related to a single

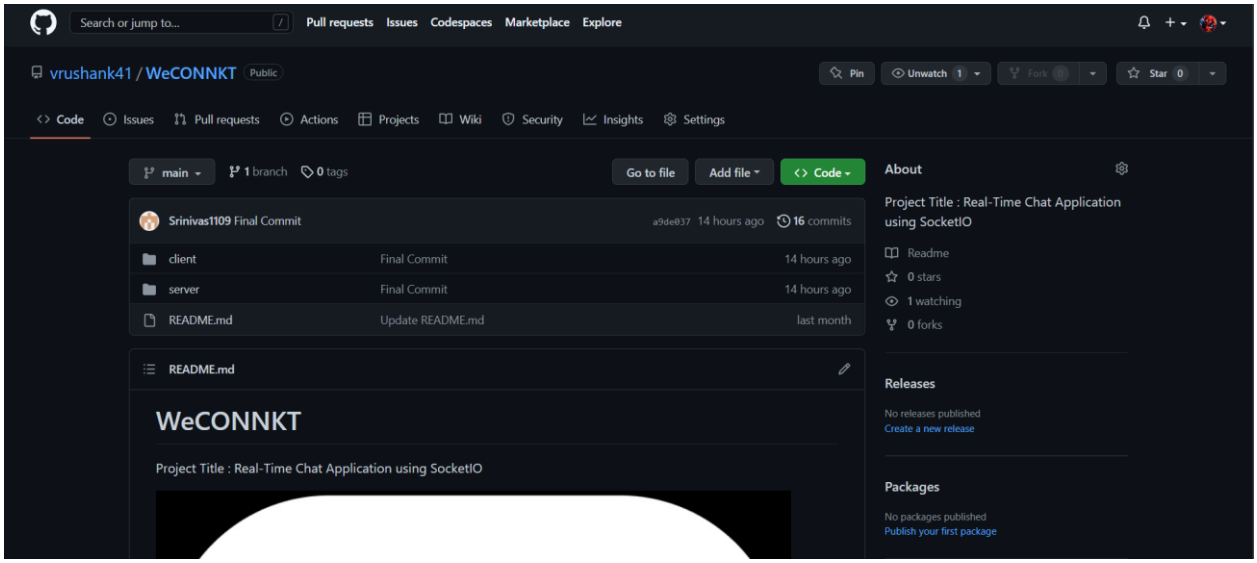
functioninto files.

- Only authorized users(who have signed up) can access our appliaction.

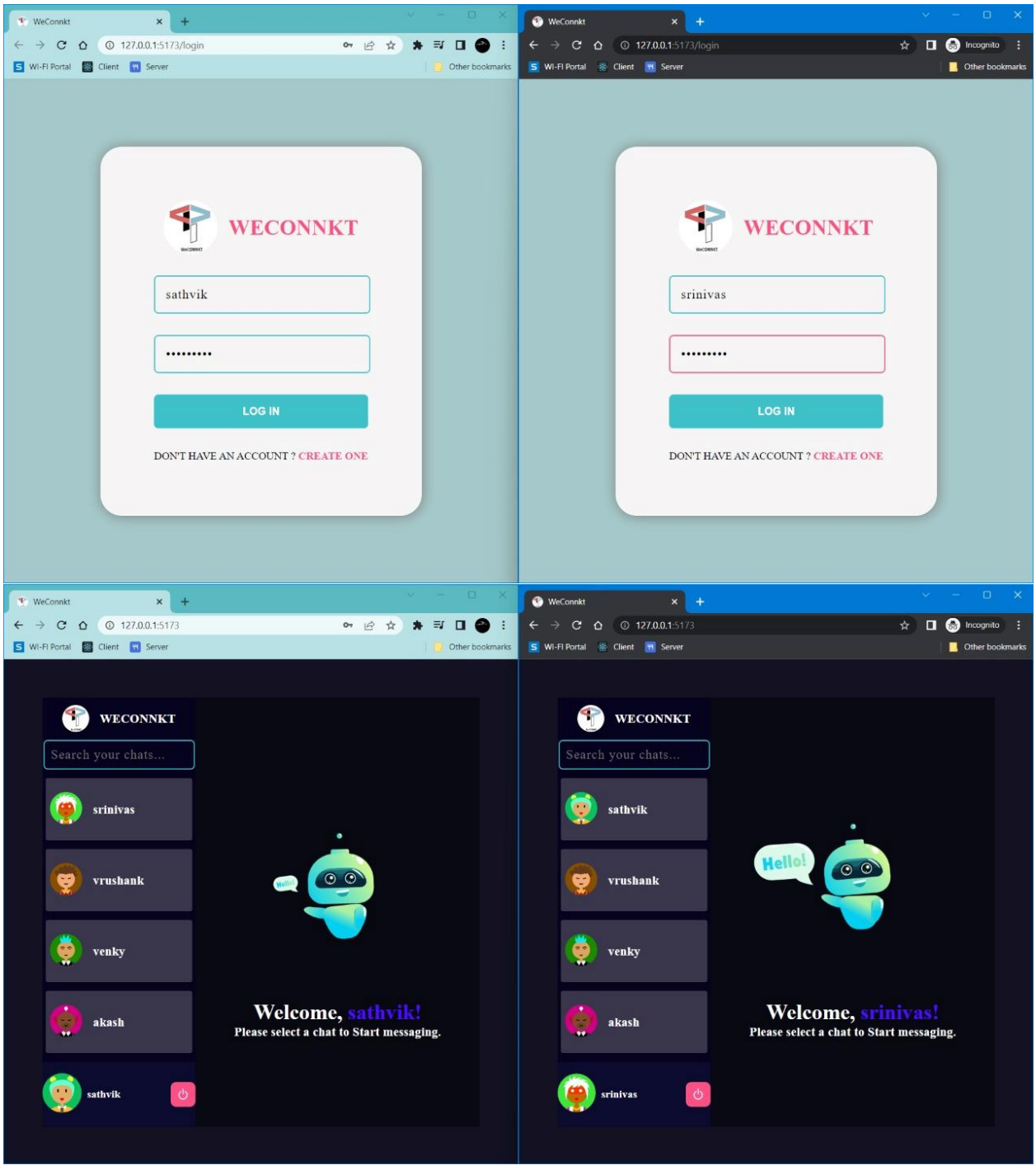
SCM

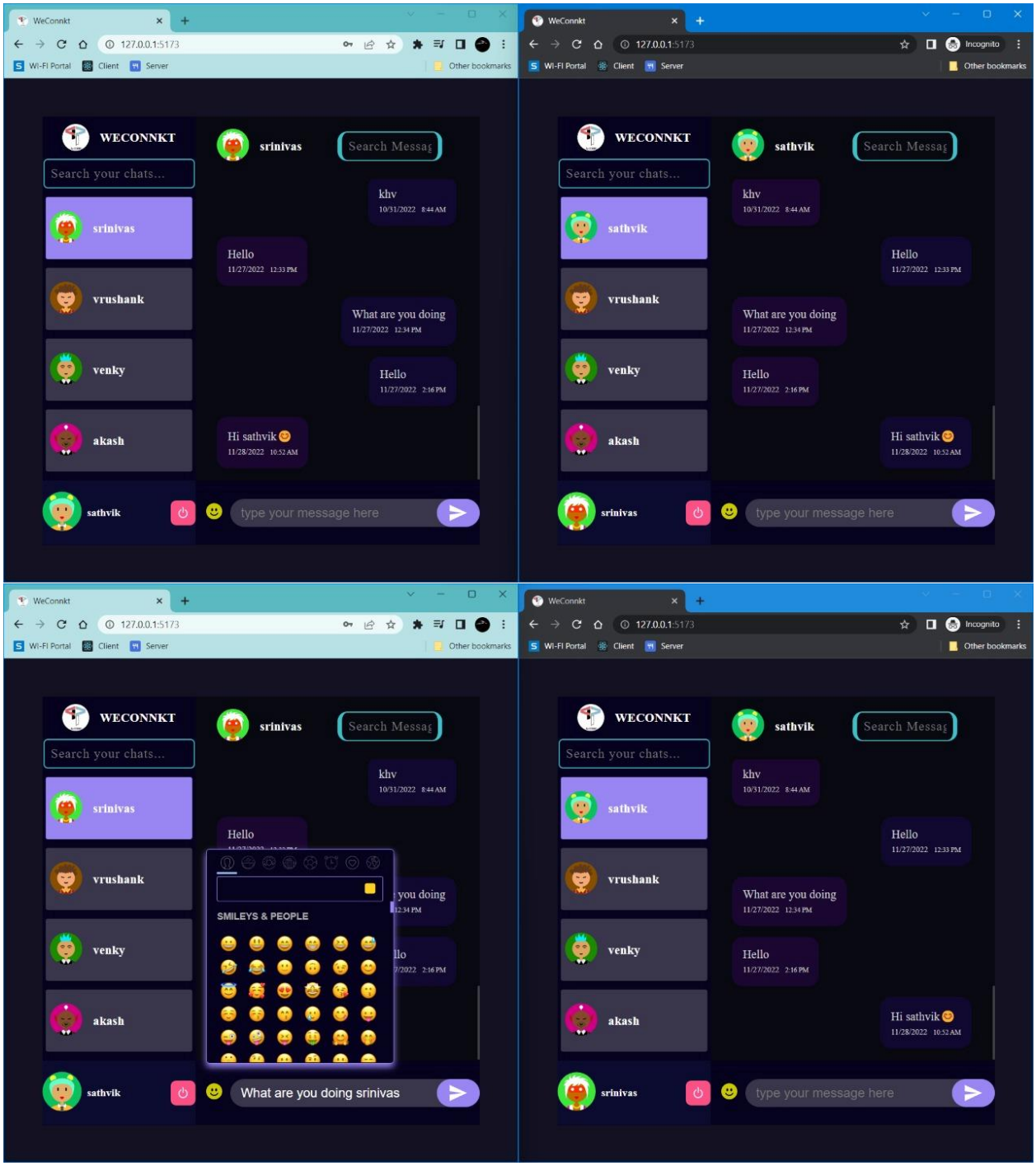
In our project, we used branching and versioning SCM methods.

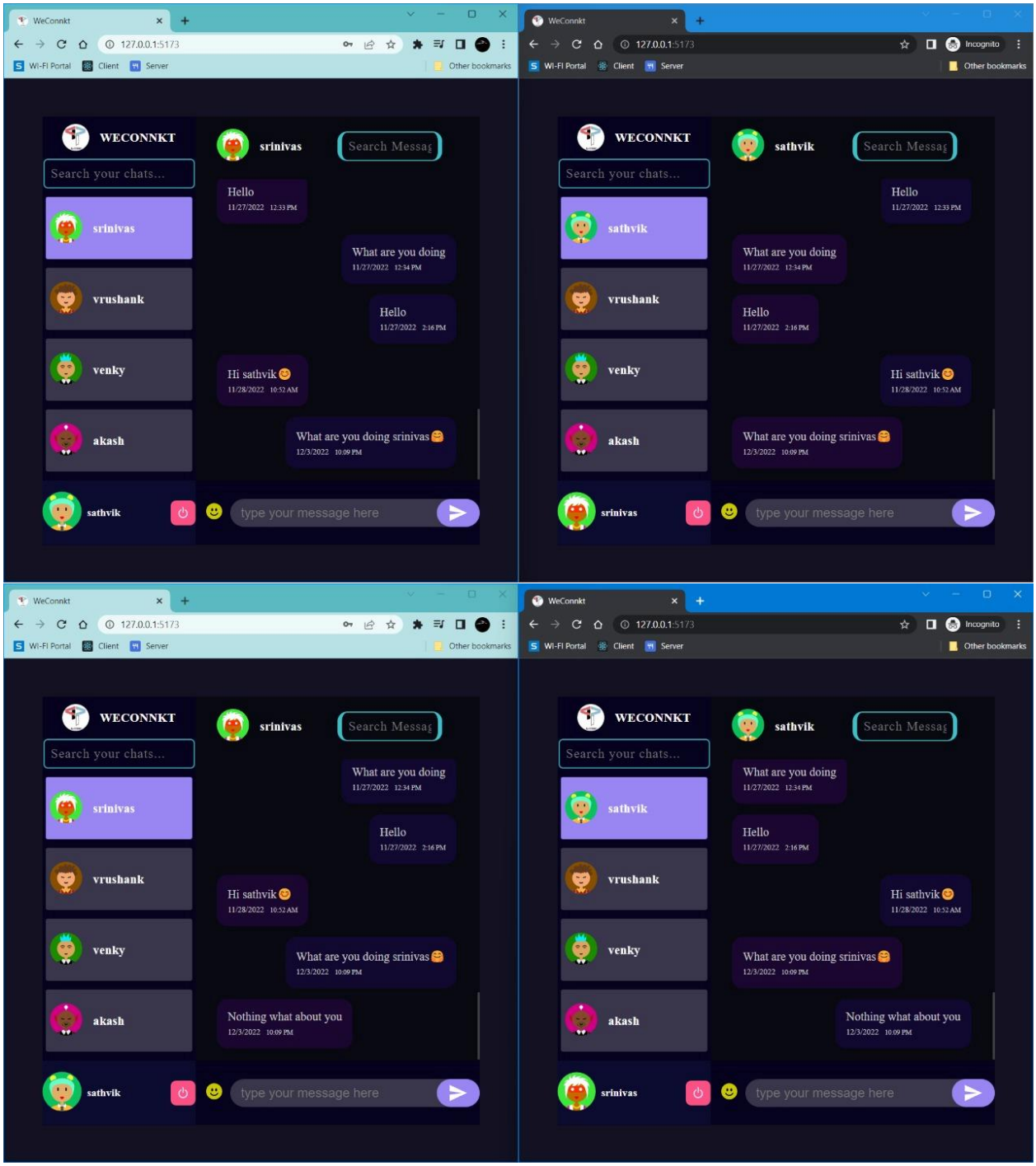
- After adding two to three features, the code was incrementally committed.
- Our project has a main branch initially containing modules for features of client and server.
- Following completion of each component including the code for the message container, emoji selection, server intergration along with mongodb , the branches were added one at a time and merged into the main branch by confirming the pull requests.
- Several commits were made within the main branch itself in order to increase application accuracy and various functionalities, such as improving the User Interface.

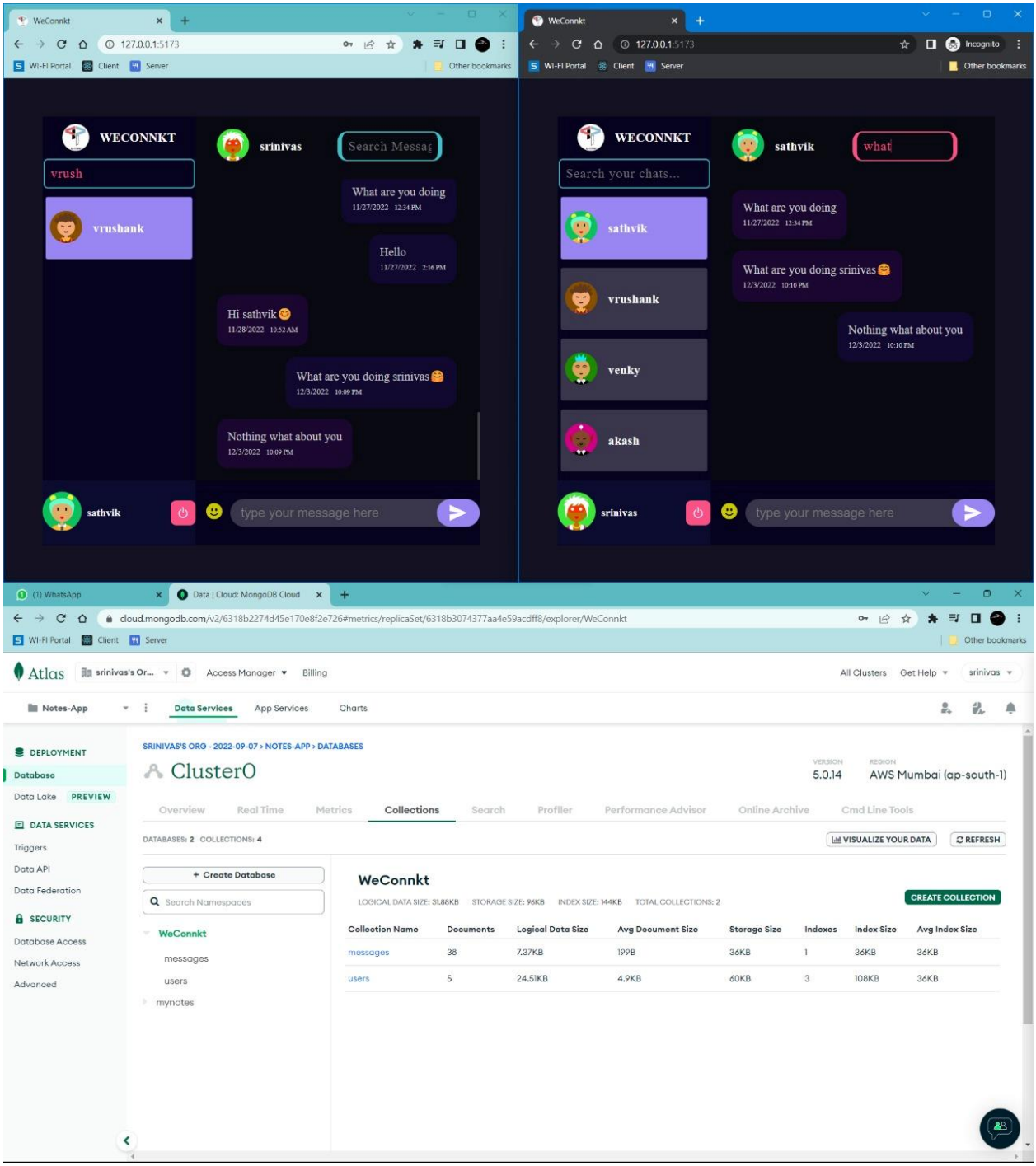


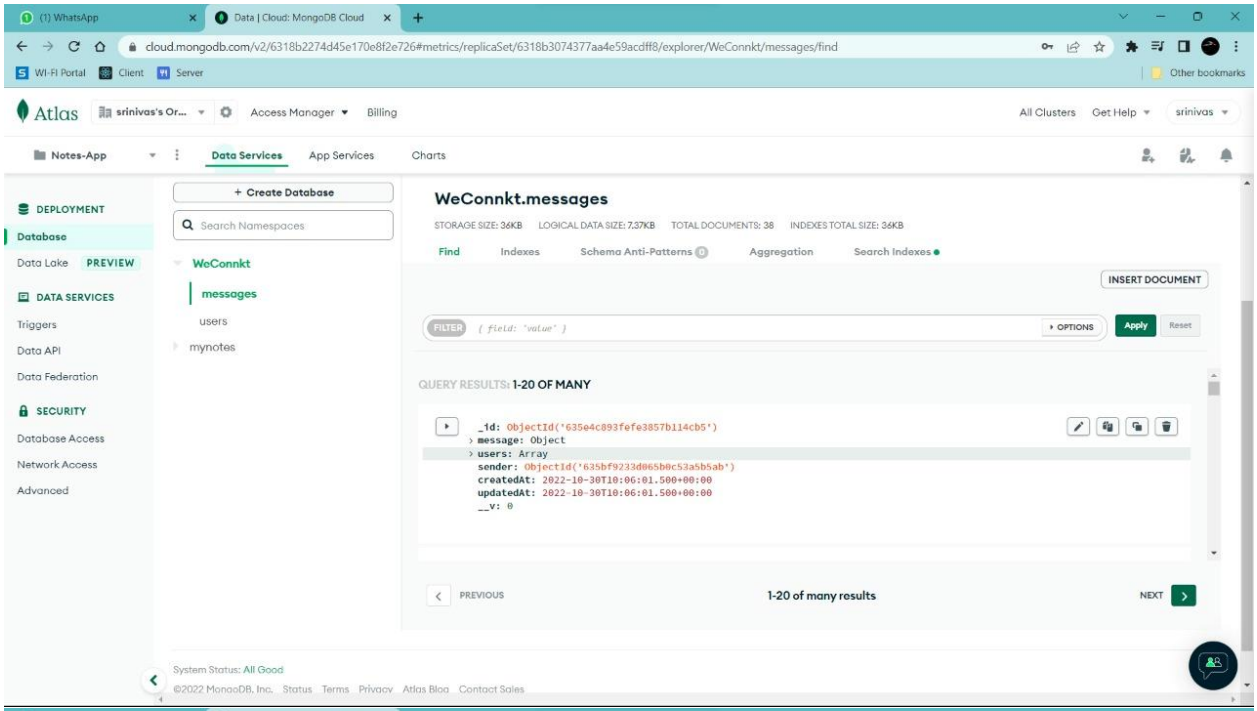
SCREENSHOTS OF THE OUTPUT

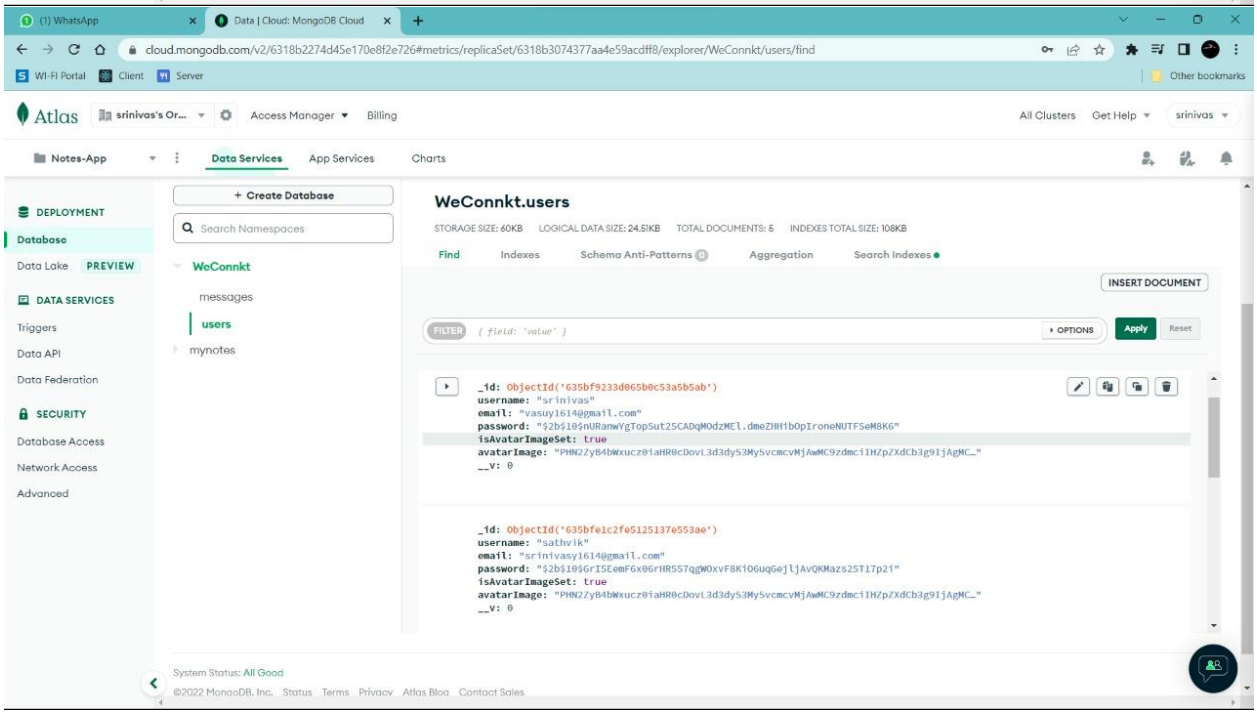

WeConnkt
127.0.0.1:5173/login
WECONNKT
sathvik
LOG IN
DON'T HAVE AN ACCOUNT ? CREATE ONE
WeConnkt
127.0.0.1:5173/login
Incognito
WECONNKT
srinivas
LOG IN
DON'T HAVE AN ACCOUNT ? CREATE ONE
WeConnkt
127.0.0.1:5173
WECONNKT
Search your chats...
srinivas
vrushank
venky
akash
sathvik
Welcome, sathvik!
Please select a chat to Start messaging.
WeConnkt
127.0.0.1:5173
Incognito
WECONNKT
Search your chats...
sathvik
vrushank
venky
akash
srinivas
Hello!
Welcome, srinivas!
Please select a chat to Start messaging.


WeConnkt
127.0.0.1:5173
Wi-Fi Portal
Client
Server
Other bookmarks
Incognito
WECONNKT
Search your chats...
srinivas
sathvik
vrushank
venky
akash
Search Messag
khv
10/31/2022 8:44 AM
Hello
11/27/2022 12:33 PM
What are you doing
11/27/2022 12:34 PM
Hello
11/27/2022 2:16 PM
Hi sathvik
11/28/2022 10:52 AM
type your message here
SMILEYS & PEOPLE
What are you doing srinivas


WeConnkt
127.0.0.1:5173
Wi-Fi Portal
Client
Server
Other bookmarks
Incognito
WECONNKT
Search your chats...
srinivas
vrushank
venky
akash
sathvik
Search Messag
Hello
11/27/2022 12:33 PM
What are you doing
11/27/2022 12:34 PM
Hello
11/27/2022 2:16 PM
Hi sathvik
11/28/2022 10:52 AM
What are you doing srinivas
12/3/2022 10:09 PM
Nothing what about you
12/3/2022 10:09 PM
type your message here


WeConnkt
127.0.0.1:5173
WECONNKT
srinivas
Search Messag
vrush
vrushank
What are you doing
11/27/2022 12:34 PM
Hello
11/27/2022 2:16 PM
Hi sathvik
11/28/2022 10:52 AM
What are you doing srinivas
12/3/2022 10:09 PM
Nothing what about you
12/3/2022 10:09 PM
sathvik
type your message here
WECONNKT
sathvik
what
Search your chats...
sathvik
What are you doing
11/27/2022 12:34 PM
vrushank
What are you doing srinivas
12/3/2022 10:10 PM
venky
Nothing what about you
12/3/2022 10:10 PM
akash
srinivas
type your message here
cloud.mongodb.com/v2/6318b2274d45e170e8f2e726#metrics/replicaSet/6318b3074377aa4e59acdff8/explorer/WeConnkt
Atlas
srinivas's Or...
Access Manager
Billing
All Clusters
Get Help
srinivas
Notes-App
Data Services
App Services
Charts
DEPLOYMENT
Database
Data Lake
PREVIEW
DATA SERVICES
Triggers
Data API
Data Federation
SECURITY
Database Access
Network Access
Advanced
SRINIVAS'S ORG - 2022-09-07 > NOTES-APP > DATABASES
Cluster0
VERSION
5.0.14
REGION
AWS Mumbai (ap-south-1)
Overview
Real Time
Metrics
Collections
Search
Profiler
Performance Advisor
Online Archive
Cmd Line Tools
DATABASES: 2 COLLECTIONS: 4
VISUALIZE YOUR DATA
REFRESH
+ Create Database
Search Namespaces
WeConnkt
messages
users
mynotes
WeConnkt
LOGICAL DATA SIZE: 31.88KB
STORAGE SIZE: 96KB
INDEX SIZE: 144KB
TOTAL COLLECTIONS: 2
CREATE COLLECTION
Collection Name
Documents
Logical Data Size
Avg Document Size
Storage Size
Indexes
Index Size
Avg Index Size
messages
38
7.37KB
199B
36KB
1
36KB
36KB
users
5
24.51KB
4.9KB
60KB
3
108KB
36KB


(1) WhatsApp
Data | Cloud: MongoDB Cloud
cloud.mongodb.com/v2/6318b2274d45e170e8f2e726#metrics/replicaSet/6318b3074377aa4e59acdff8/explorer/WeConnkt/messages/find
Atlas
srinivas's Or...
Access Manager
Billing
All Clusters
Get Help
srinivas
Notes-App
Data Services
App Services
Charts
DEPLOYMENT
Database
Data Lake PREVIEW
DATA SERVICES
Triggers
Data API
Data Federation
SECURITY
Database Access
Network Access
Advanced
+ Create Database
Search Namespaces
WeConnkt
messages
users
mynotes
WeConnkt.messages
STORAGE SIZE: 36KB
LOGICAL DATA SIZE: 7.37KB
TOTAL DOCUMENTS: 38
INDEXES TOTAL SIZE: 36KB
Find
Indexes
Schema Anti-Patterns
Aggregation
Search Indexes
INSERT DOCUMENT
FILTER
{ field: 'value' }
OPTIONS
Apply
Reset
QUERY RESULTS: 1-20 OF MANY
_id: ObjectId('635e4c893fefe3857b114cb5')
message: Object
users: Array
sender: ObjectId('635bf9233d065b0c53a5b5ab')
createdAt: 2022-10-30T10:06:01.500+00:00
updatedAt: 2022-10-30T10:06:01.500+00:00
__v: 0
PREVIOUS
1-20 of many results
NEXT
System Status: All Good
©2022 MongoDB, Inc. Status Terms Privacy Atlas Blog Contact Sales


(1) WhatsApp
Data | Cloud: MongoDB Cloud
cloud.mongodb.com/v2/6318b2274d45e170e8f2e726#metrics/replicaSet/6318b3074377aa4e59acdff8/explorer/WeConnkt/users/find
Atlas
srinivas's Or...
Access Manager
Billing
All Clusters
Get Help
srinivas
Notes-App
Data Services
App Services
Charts
DEPLOYMENT
Database
Data Lake PREVIEW
DATA SERVICES
Triggers
Data API
Data Federation
SECURITY
Database Access
Network Access
Advanced
+ Create Database
Search Namespaces
WeConnkt
messages
users
mynotes
WeConnkt.users
STORAGE SIZE: 60KB
LOGICAL DATA SIZE: 24.51KB
TOTAL DOCUMENTS: 5
INDEXES TOTAL SIZE: 108KB
Find
Indexes
Schema Anti-Patterns
Aggregation
Search Indexes
INSERT DOCUMENT
FILTER
{ field: 'value' }
OPTIONS
Apply
Reset
_id: ObjectId('635bf9233d065b0c53a5b5ab')
username: "srinivas"
email: "vasuy1614@gmail.com"
password: "$2b$10$nURamwYgTopSut25CADqMOdzMEl.dmeZHHibOpIroneNUTFSeM8K6"
isAvatarImageSet: true
avatarImage: "PHN2ZyB4bWxucz0iaHR0cDovL3d3dy53My5vcmcvMjAwMC9zdmciIHZpZXdCb3g9IjAgMC…"
__v: 0
_id: ObjectId('635bfe1c2fe5125137e553ae')
username: "sathvik"
email: "srinivasy1614@gmail.com"
password: "$2b$10$GrI5EomF6x06rHR557qgWOxvF8KiO6uqGejljAvQKMazs25T17p2i"
isAvatarImageSet: true
avatarImage: "PHN2ZyB4bWxucz0iaHR0cDovL3d3dy53My5vcmcvMjAwMC9zdmciIHZpZXdCb3g9IjAgMC…"
__v: 0
System Status: All Good
©2022 MongoDB, Inc. Status Terms Privacy Atlas Blog Contact Sales

Conclusion

We have used vite+reactJs for frontend and for backend integration in nodejs,expressjs along with mongodb cloud.
We also made the git repository for this project public.Our project is the live demonstration of the standard chat applications and it has most of thefeatures that the website has.

As a result, the product has been successfully developed in terms of extendability, portability, and maintainability and tested in order to meetall requirements .

Future Work

- *Increasing the effectiveness of the application by providing Voice/videocalls.*
- *When dealing with sensitive information like compliance requirements or personal user information, making sure that your real-time chat application has the proper security features in place is crucial for ensuring that the data of those using your app is protected.**End-to-EndEncryption** is a system of communication where particularly communicating users can read the messages.*
- *Chatbots (AI-based programs) are all over the messaging apps. Nevertheless, it is important to understand chatbots first. Chatbots are software agents that communicate and collaborate with human users viatext messaging by using natural voice to simplify tasks.*
- *To share files/documents*
- *Push Notifications : Instant messaging apps can't be completed withoutpush notifications that enable users to check the new messages. So, push notifications is a must-have feature, which serves direct communication between app user and messenger provider. Also, it informs users about new messages.*
- *Cloud Synchronization : This is a process of keeping files like images, docs, audios, and videos stored in different places. When the user makeschanges to the file on their device, then it will automatically be applied to all other instances of files.*