



GLA UNIVERSITY, MATHURA

## **WhatsApp Clone**

*A Project Report submitted in partial fulfillment of the requirements for the award of the degree of*

**Bachelor of Technology**

**In**

**Computer science and Engineering**

**by**

<b>Name of Candidate:</b>	<b>BEDPRAKASH</b>
<b>University Roll No:</b>	<b>201500189</b>
<b>Name of Candidate:</b>	<b>KABEER SAXENA</b>
<b>University Roll No:</b>	<b>201500321</b>
<b>Name of Candidate:</b>	<b>UJJWAL GUPTA</b>
<b>University Roll No:</b>	<b>201500751</b>
<b>Name of Candidate:</b>	<b>PIYUSH SINGH</b>
<b>University Roll No:</b>	<b>201500472</b>

Under the Guidance of  
Mr. Mayank Saxena  
(Technical trainer)

Department of training & Development

**Institute of Engineering & Technology**



Department of Computer Engineering and Applications  
GLA University, 17 km. Stone NH#2, Mathura-Delhi  
Road, Chaumuha, Mathura – 281406 U.P (India)

## **BONAFIDE CERTIFICATE**

I/we hereby declare that the work which is being presented in the B.Tech. Project “**WhatsApp clone**”, in partial fulfilment of the requirements for the award of the ***Bachelor of Technology*** in Computer Science and Engineering and submitted to the Department of Computer Engineering and Applications of GLA University, Mathura, is an authentic record of my own work carried under the supervision of **Mr. Mayank Saxena** who is **Technical Trainer** in GLA University.

The contents of this project report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree.

Sign\_\_Bedprakash\_\_\_\_\_

Name of Candidate: BEDPRAKASH

University Roll No.: **2015000189**

Sign\_\_Kabeer Saxena\_\_\_\_\_

Name of Candidate: KABEER SAXENA

University Roll No: **201500321**

Sign\_\_Ujjwal Gupta\_\_\_\_\_

Name of Candidate: UJJWAL GUPTA

University Roll No: **201500751**

Sign\_\_Piyush Singh\_\_\_\_\_

Name of Candidate: PIYUSH SINGH

University Roll No: **201500472**



## **Acknowledgement**

It gives us a great sense of pleasure to present the synopsis of the B.Tech mini project undertaken during B. Tech III Year. This project is going to be an acknowledgement to the inspiration, drive and technical assistance will be contributed to it by many individuals. We owe special debt of gratitude to Mr. Mandeep Singh, Technical Trainer, for providing us with an encouraging platform to develop this project, which thus helped us in shaping our abilities towards a constructive goal and for his constant support and guidance to our work. His sincerity, thoroughness and perseverance has been a constant source of inspiration for us. We believe that he will shower us with all his extensively experienced ideas and insightful comments at different stages of the project & also taught us about the latest industry-oriented technologies. We also do not like miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind guidance and co-operation

Sign\_\_Bedprakash\_\_\_\_\_

Name of Candidate: **BEDPRAKASH**

University Roll No.: **2015000189**

Sign\_\_Kabeer Saxena\_\_\_\_\_

Name of Candidate: **KABEER SAXENA**

University Roll No: **201500321**

Sign\_\_Ujjwal Gupta\_\_\_\_\_

Name of Candidate: **UJJWAL GUPTA**

University Roll No: **201500751**

Sign\_\_Piyush Singh\_\_\_\_\_

Name of Candidate: **PIYUSH SINGH**

University Roll No: **201500472**

## **ABSTRACT**

WhatsApp is one of the channels that are often used for communication. In the development of WhatsApp, it has made an application to facilitate users to access the technology on the computer desktop version.

Furthermore, web WhatsApp is also a cross-platform application that are used to access the WhatsApp messenger on desktop computers. It is a web-based application that synchronizes with WhatsApp application on mobile devices. To synchronize, users required to scan the QR code that generated when opening the application web WhatsApp. In line with mobile platforms, the rising of web-based applications have required a model for investigating criminal cases that utilizing WhatsApp on web application. The unavailability of model that used to investigate cases that utilize WhatsApp web could make it difficult for investigators. This paper gives a contribution to the preliminary model for investigating of crime cases related to the application web WhatsApp by studying the investigative process on the web browser application and network traffic

## **Contents**

Bonafide Certificate

Acknowledgement

Abstract

### **1. INTRODUCTION..**

1.1 Overview & Motivation

1.2 Objective

1.3 Contribution

1.4 Organization of Project Report

### **2. LITERATURE REVIEW..**

### **3. Tools And Technology...**

### **4.IMPLEMENTATION & RESULT..**

3.1 Psuedo code

3.2 Result

3.3 Analysis

### **5.CONCLUSION & FUTURE WORK..**

**Reference Work...**

## **INTRODUCTION**

WhatsApp web is a desktop version of WhatsApp Messenger launched on 21 January 2015 for android, blackberry, and window phone users only. But later the company launched it for iOS and Nokia users as well . Now you can also operate your WhatsApp Messenger on your computer and laptop. This is a new feature of WhatsApp which has been made to use WhatsApp Messenger on Browser. There is no need to download the app. Moreover, it allows the user to send and receive messages directly from their browser You can use WhatsApp web in the latest web browser and your mobile number should already be connected on your WhatsApp. It doesn't have any signup or sign in option. In short, it is a browser version of Messenger which allows you to connect through your laptop or desktop. If you are using WhatsApp from the laptop, then your mobile data will also be used. If you turn off the mobile data, then the connection of WhatsApp will also be closed. WhatsApp launched a web-friendly version of its platform in 2015. With the help of the WhatsApp web, users have got the option to use the messaging service on their desktop as well. In this web version, almost all the features found on the smartphone app have been given. Although video and voice calling is not available on WhatsApp web. Many features found on the web version make your experience better on the app. Recently, WhatsApp has released Dark Theme for WhatsApp web and desktop apps. Earlier these features were limited to mobile apps only. Now after the introduction of this feature, users will not have to use any trick to enjoy the dark mode.

## **OVERVIEW**

In the existing system, each task is carried out manually and processing is also a tedious job. In previous system separately were maintaining time table details manually in pen and paper, which was time taking and costly. The travellers is not able to achieve its need in time and also the results may not accurate. Because of the manual maintenance there are number of difficulties and drawbacks exist in the system. Some of them are Increased transaction leads to increased source document and hence maintenance becomes difficult. If any admin, user entry is wrongly made then the maintenance becomes very difficult The proposed system is designed to be more efficient than the manual system. It invokes all base tasks that are now carried out manually, such as the forms transactions and reports which is added advantage. The proposed System is completely computer-based application. Thousands of records can searched and displayed without taking any significant time gives accurate information Simplifies the manual work It minimizes the documentation related work Provides up to date information Friendly Environment by providing warning messages. Product details can be provided n booking confirmation notification This module is mainly based on admin. All works are done under his control This module covers the details about the registration of users which they can be register by itself by adding data like name, password, email id and further details. After registration they can be sign in by their username and password

## **OBJECTIVE**

The objective of the Multitask web Project is to develop a system that automates the processes and activities of the college, hospital , shopping mall, entertainment, sports and healthy food related info and the purpose is to design a system using which one can perform all operation related to above .



## **CONTRIBUTION**

I also worked on this project with the same vision of ‘team work’ that leads to the completion of the project in the best possible way. I had an idea of what i supposed to do in order to contribute my bit towards the project completion. i tried to complete the tasks assigned to me in the given time limits and then took over the heading task. This also helped me in developing some team management skills needed for the project development.

i was continuously working on the coding stuff trying to optimize at each step to achieve maximum possible accuracy and at the same time some of us completing the documentation part as per the progress of the project.

So, it is a common experience that working on any project as a team helps in completing it at a better pace and i also get an opportunity to pick the best solution presented by me for any problem that increases the efficiency of the project.

## **ISSUES & CHALLENGES**

- Building and Maintaining the Credibility of Online Presence. ...
- Maintaining own Service Standard. ...
- Make Booking Procedure more Convenient. ...
- Providing Services as per Specific Requirements.

## **Organization of the project report:-**

### 1.) Literature Review:

Contains the research works done over the project.

### 2.) Proposed Work:-

The algorithm and the pseudo code used in the project.

### 3.) Implementation and Result analysis:-

The implementation of the actual code and analysing the outcomes of it.

### 4.) Conclusion:-

Conclusion of the report and preferring a classifier that might give best accuracy.

### 5.) References:-

The references taken for the completion of the project.

## **LITERATURE REVIEW**

Tourism activities can refer to an extension of a brand–consumer relationship. While growth in many industries is flat, worldwide tourism revenues continue to grow. A literature review is not only a crucial endeavour for any academic research, but also the foundation and inspiration for substantial, useful research. Among extant studies on tourism management (TM), few are on literature review while most are concerned with specific issues or countries/areas. This paper aims to draw up an integrated framework of TM. Little effort has been made to systematically examine the vast TM-related literature so as to facilitate better understandings of TM. To eliminate the gap among the extant studies and develop the TM trajectory, a content analysis was undertaken using keywords “TM” in 5 online electronic databases from 1990 to 2013. Based on 773 articles, we discovered the number of publications on TM has significantly increased since 2000 and a steady growth since 2008. While 773 articles are scattered across 196 journals, most appeared in 11 academic journals. We also categorize articles into 10 conceptual groups based on a proposed conceptual framework. The main contribution is to provide a conceptual framework incorporating keyword indexes to operationalize the coverage of TM.

ore recently, growth has been witnessed in the use of content analyses as a data analysis approach by scholars within the Hospitality and Tourism context (e.g., Adeyinka-Ojo, Nair & Khoo-Lattimore, 2014; Chang & Katrichis, 2016). There is an also a growing tendency to conduct content analyses based on user-generated photographs (Stepchenkova & Zhan, 2013), including content in the form of travelers' online reviews (Lu & Stepchenkova, 2015) A review of existing literature has revealed an increasing number of studies in which a content analysis is adopted as a data treatment technique within the Hospitality and Tourism context (e.g. Chang & Katrichis, 2016; Gros et al., 2013; Mura & PahlevanSharif, 2015; Prayag & Ryan, 2011), given the deeper understanding of the phenomenon being investigated provided to the user. Aside from content analyses, there are many other approaches available for tanalysis of qualitative data, including discourse analyses, analyses of online texts or narrative analyses, and text analyses (Pahlevan-Sharif et al., 2019) Another qualitative content analysis method is the literature review approach, which has recently become a popular technique among tourism scholars in this field (see Chang & Katrichis, 2016; Law et

al., 2009). This approach comprises four main stages, namely material collection, descriptive analysis, category selection and material evaluation (Mayring, 2003)

In recent years, tourism revenues have continued to increase worldwide because of the emergence of an ideology of entertainment and the increase in countries' national income. Tourism has become part of people's lives, both local and international (Chang & Katrichis, 2016).

There is also a set of factors that work as a dam for tourism development, such as economic growth, improving places for access, and political liberalization, given the role of technology in action (Tolkach2016)

Technology primarily works with the organization's three strategies, quality, and direction, human resource management, plus information technology. It solves its ability to communicate problems with customers, so organizations must take advantage of these innovative approaches to technology (Chang & Katrichis, 2016). Technology is bringing about a gradual revolution in the tourism industry

Tourism is the movement of people from their place of residence to a remote location for a temporary period so that you may consider tourism a social, cultural, and economic phenomenon (Chang & Katrichis, 2016). In other words, tourism can be defined as a commercial activity that depends mainly on extensive information and many human resources (Steinbauer & Werthner, 2016)

### **Fact Finding Techniques:**

We mainly used three fact finding techniques to find out for ourselves the correct information on the basis of which we will build the software. These fact finding techniques are extremely important because these are the facts on basis of which we can build the software that comprises of a friendly environment for the members work with. This is the reason why fact finding is an important activity grouped under the second phase “Requirement Analysis” of the Software Development life cycle.

## **Tools And Technology**

### **Front-end and Back-end: Front-end:**

HTML5, CSS, Javascript, Bootstrap (Framework)  
React.js, Express.js, Node.js, MongoDB, Socket.io

### **Hardware and Software Requirements:**

Minimum Hardware Requirements:

Processor: PIII 500MHZ or above

RAM: 128MB RAM Hard Disk: 100MB Free Hard  
disk space Monitor: Standard Color Monitor

### **Minimum Software Requirements:**

Operating System: Windows Family

Language: Microsoft Visual Studio  
2008

## **SYSTEM DESIGN & DEVELOPMENT**

➤ System design is the solution to the creation of a new system. This phase is composed of several systems. This phase focuses on the detailed implementation of the feasible system. It emphasis on translating design specifications to performance specification. System design has two phases of development logical and physical design.

➤ During logical design phase the analyst describes inputs (sources), out puts (destinations), databases (data sores) and procedures (data flows) all in a format that meats the uses requirements. The analyst also specifies the user needs and at a level that virtually determines the information flow into and out of the system and the dataresources. Here the logical design is done through data flow diagrams and database design.

➤ The physical design is followed by physical design or coding. Physical design produces the working system by defining the design specifications, which tell the programmers exactly what the candidate system must do. The programmers write the necessary programs that accept input from the user, perform necessary processing on accepted data through call and produce the required report on a hard copy or display iton the screen

## **Project Stages**

We can break down the project in the following stages:

1. Front-end
  - Sidebar Component
  - Chat Component
- 2 Back-end
  - Authentication
  - Database



## WORKING

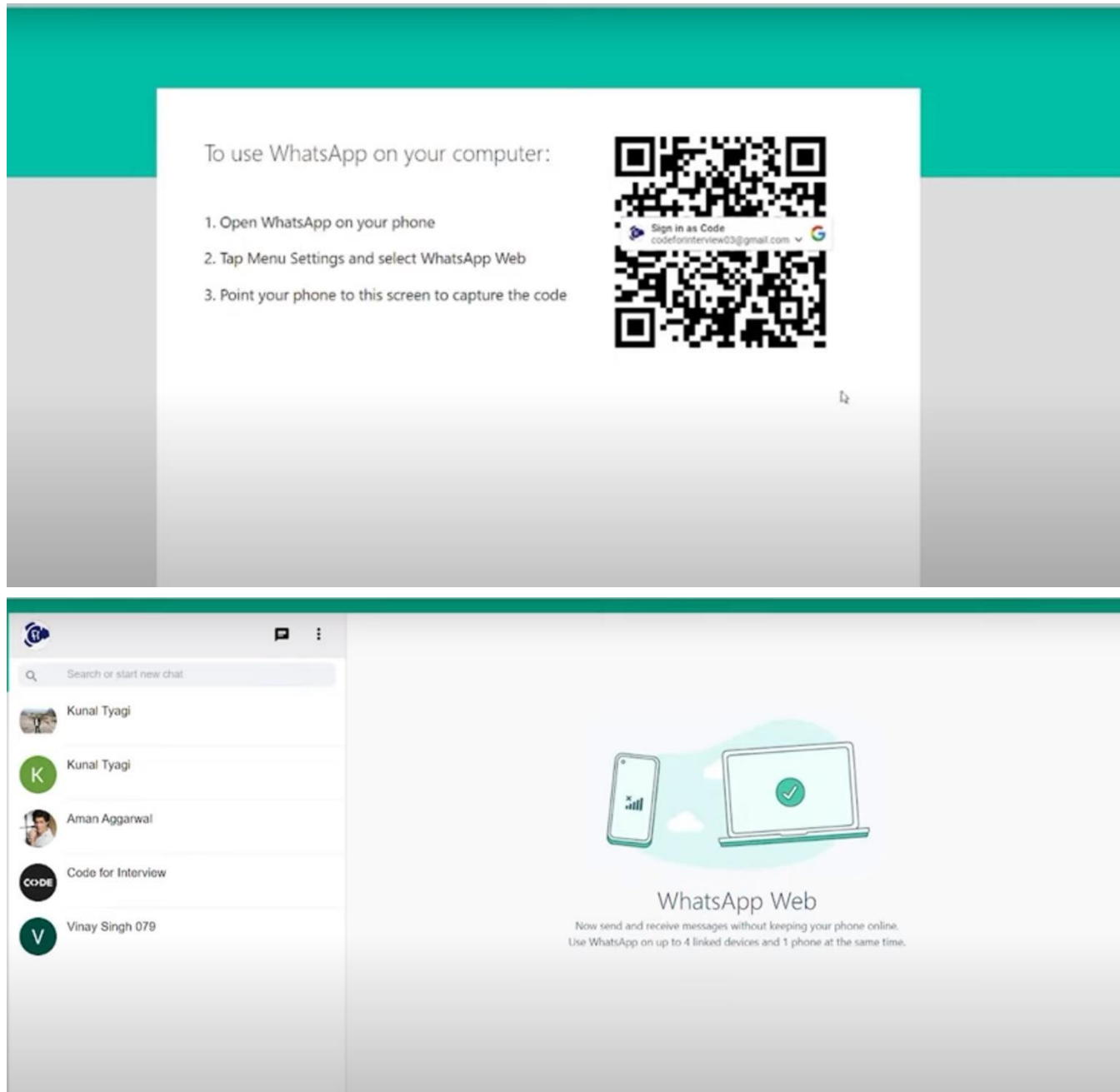
A working web based WhatsApp Clone with a responsive UI for one to one communication. Including Google Auth to verify users and avoid spam. This project will give you a hands on experience with the React library. Fullstack development currently has many different stacks and technologies to learn. It is very easy to get overwhelmed and get distracted. React is one of the most popularly used Frontend libraries used by companies such as Facebook, Pinterest, Uber, Instagram and many more. Beginner tutorials and mini projects are good for beginning but a full-fledged project can make you understand concepts with real-world applications. You will learn not just about UI designing but also integrating it with the backend. You will be learning about :

- Basics of React
  - Functional Components
  - UseState
  - UseEffect
  - PropTypes
- Context API
  - Material- UI
  - (Design)Firebase
  - Authentication (using Sign In With
  - Google)Mongo (DB)

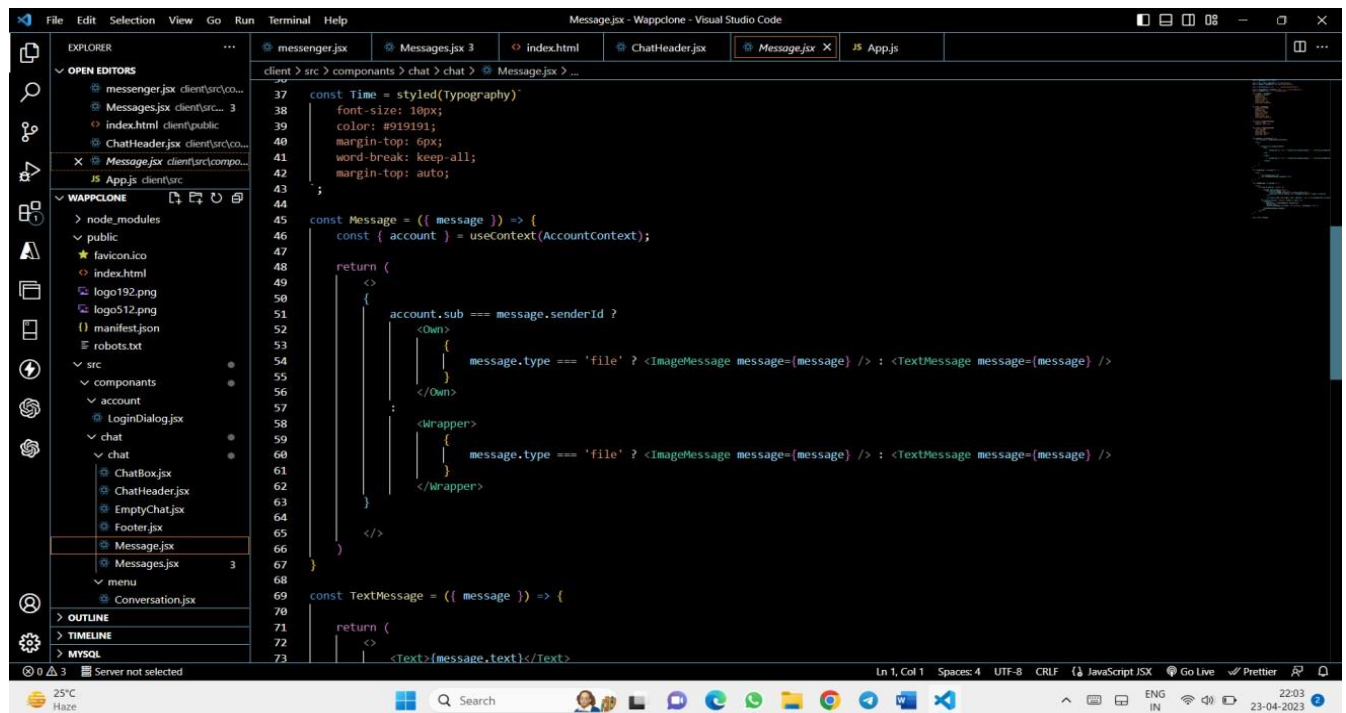
## IMPLEMENTATION

WhatsApp Clone provides an in depth view of implementation on how to create a full-stack, mobile, hybrid web application from scratch. This repository covers implementation of 3 major platform's mainly Android, IOS & Web using React Native & MERN stack.

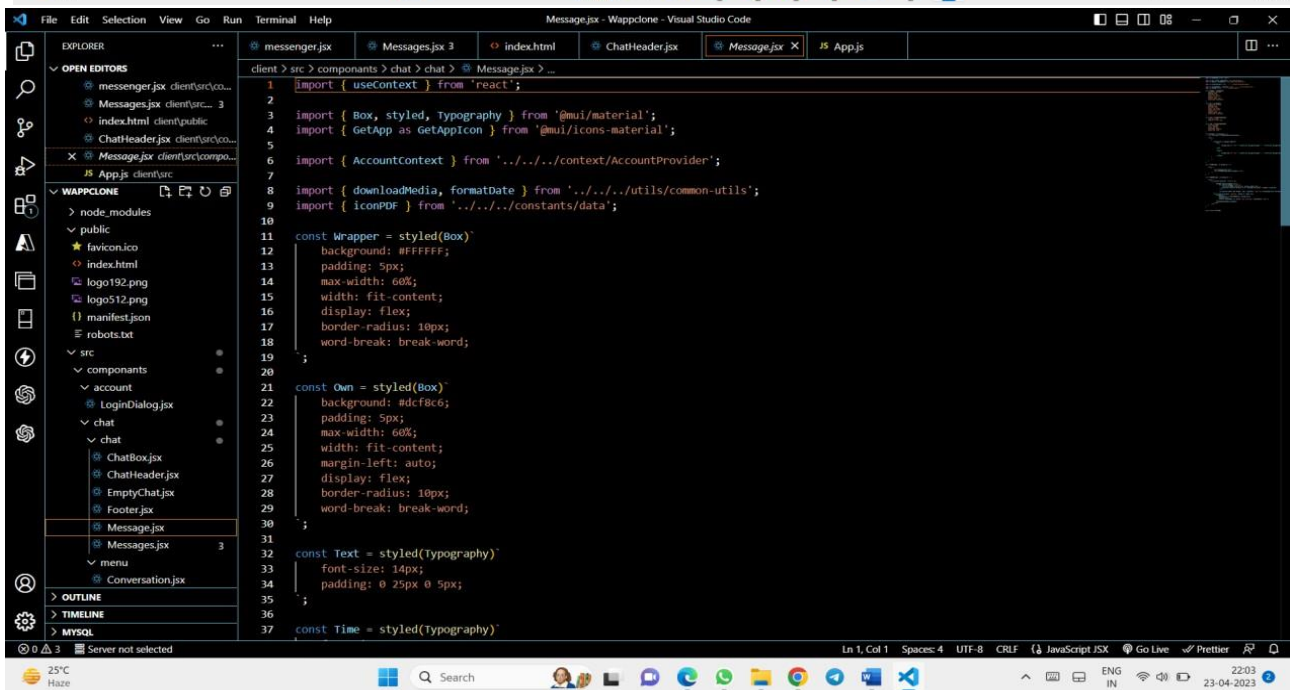
### HOME SCREEN



## Some Screenshots of source code



```
client > src > components > chat > chat > Message.jsx > ...
37 const Time = styled(Typography)`
38   font-size: 10px;
39   color: #919191;
40   margin-top: 6px;
41   word-break: keep-all;
42   margin-top: auto;
43 `;
44
45 const Message = ({ message }) => {
46   const { account } = useContext(AccountContext);
47
48   return (
49     <div>
50       {
51         account.sub === message.senderId ?
52         <Own>
53           {
54             message.type === 'file' ? <ImageMessage message={message} /> : <TextMessage message={message} />
55           }
56         </Own>
57       :
58       <Wrapper>
59         {
60           message.type === 'file' ? <ImageMessage message={message} /> : <TextMessage message={message} />
61         }
62       </Wrapper>
63     </div>
64   )
65 }
66
67
68
69 const TextMessage = ({ message }) => {
70   return (
71     <div>
72       <Text>{message.text}</Text>
73     </div>
74   )
75 }
```



```
client > src > components > chat > chat > Message.jsx > ...
1 import { useContext } from 'react';
2
3 import { Box, styled, Typography } from '@mui/material';
4 import { GetApp as GetAppIcon } from '@mui/icons-material';
5
6 import { AccountContext } from '../context/AccountProvider';
7
8 import { downloadMedia, formatDate } from '../utils/common-utils';
9 import { ICONPDF } from '../constants/data';
10
11 const Wrapper = styled(Box)`
12   background: #FFFFFF;
13   padding: 5px;
14   max-width: 60%;
15   width: fit-content;
16   display: flex;
17   border-radius: 10px;
18   word-break: break-word;
19 `;
20
21 const Own = styled(Box)`
22   background: #dcf8c6;
23   padding: 5px;
24   max-width: 60%;
25   width: fit-content;
26   margin-left: auto;
27   display: flex;
28   border-radius: 10px;
29   word-break: break-word;
30 `;
31
32 const Text = styled(Typography)`
33   font-size: 14px;
34   padding: 0 25px 0 5px;
35 `;
36
37 const Time = styled(Typography)`
```

```
1 import { useContext, useState, useEffect } from 'react';
2
3 import { Box } from '@mui/material';
4
5 import { UserContext } from '../../context/UserProvider';
6 import { AccountContext } from '../../context/AccountProvider';
7 import { getConversation } from '../../service/api';
8
9 //components
10 import ChatHeader from './ChatHeader';
11 import Messages from './Messages';
12
13 const ChatBox = () => {
14   const { person } = useContext(UserContext);
15   const { account } = useContext(AccountContext);
16
17   const [conversation, setConversation] = useState({});
18
19   useEffect(() => {
20     const getConversationDetails = async () => {
21       let data = await getConversation({ senderId: account.sub, receiverId: person.sub });
22       //setConversation(data);
23       const getConversationDetails = () => Promise<void>
24       getConversationDetails();
25     }, [person.sub]);
26
27     return (
28       <Box style={{height: '75%'}}>
29         <ChatHeader person={person} />
30         <Messages person={person} conversation={conversation} />
31       </Box>
32     )
33   }
34
35   export default ChatBox;
```

```
34 font-family: inherit;
35
36
37 const StyledDivider = styled(Divider)`
38   margin: 40px 0;
39   opacity: 0.4;
40 `;
41
42 const EmptyChat = () => {
43
44   return (
45     <Component>
46       <Container>
47         <Image src={emptyChatImage} alt="empty" />
48         <Title>WhatsApp Web</Title>
49         <Subtitle>Now send and receive messages without keeping your phone online.</Subtitle>
50         <Subtitle>Use WhatsApp on up to 4 linked devices and 1 phone at the same time.</Subtitle>
51       </Container>
52     </Component>
53   )
54
55 }
56
57 export default EmptyChat;
```

EmptyChat.jsx - Wappclone - Visual Studio Code

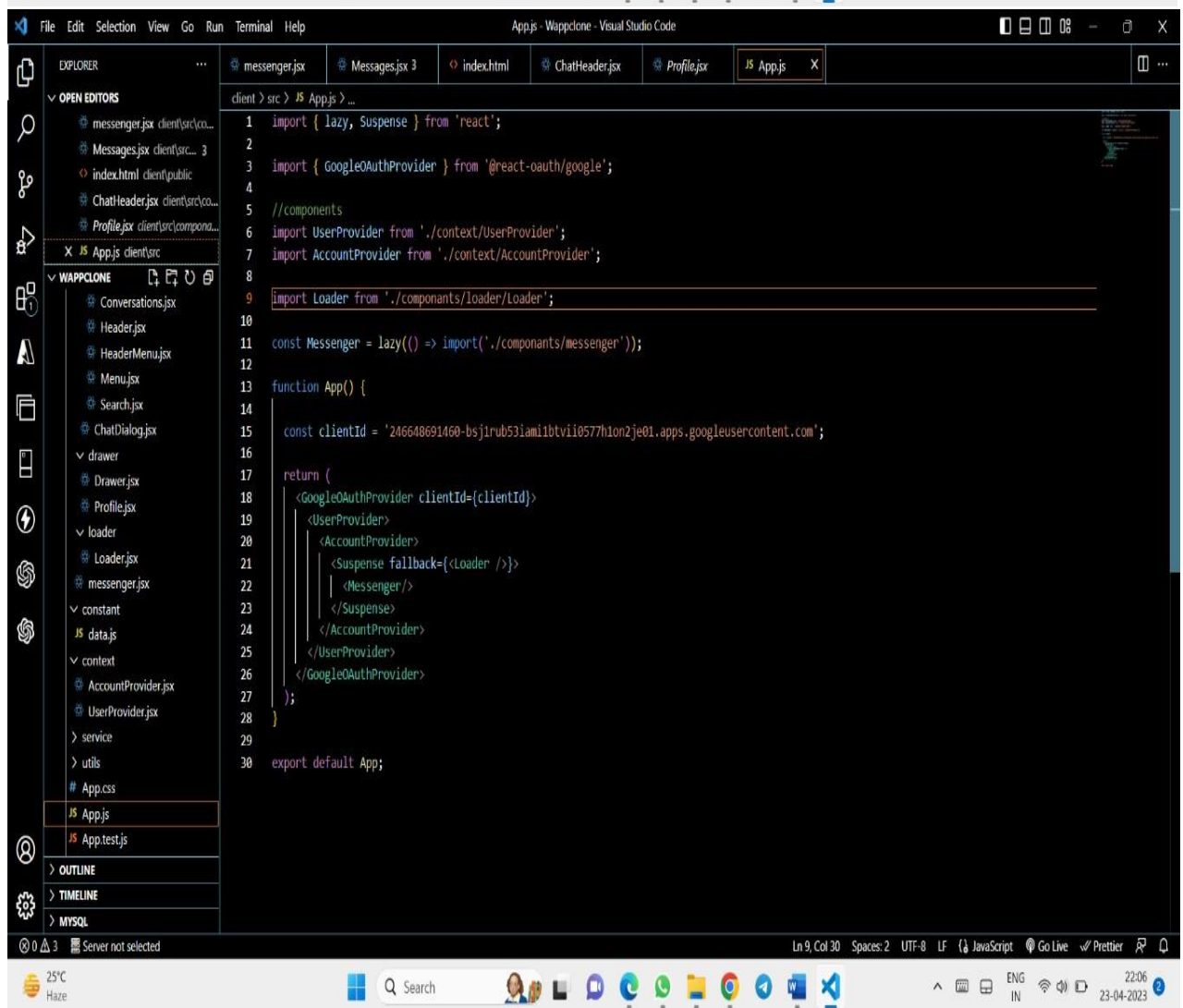
```
client > src > components > chat > chat > EmptyChat.jsx > | file
1
2 import { Box, styled, Typography, Divider } from '@mui/material';
3
4 import { emptyChatImage } from '../../constants/data';
5
6 const Component = styled(Box)`
7   background: #f8f9fa;
8   padding: 30px 0;
9   text-align: center;
10  height: 100%;
11 `;
12
13 const Container = styled(Box)`
14   padding: 0 200px;
15 `;
16
17 const Image = styled('img')`
18   margin-top: 100;
19   width: 400;
20 `;
21
22 const Title = styled(Typography)`
23   font-size: 32px;
24   font-family: inherit;
25   font-weight: 300;
26   color: #41525d;
27   margin-top: 25px 0 10px 0;
28 `;
29
30 const Subtitle = styled(Typography)`
31   font-size: 14px;
32   color: #667781;
33   font-weight: 400;
34   font-family: inherit;
35 `;
36
37 const StyledDivider = styled(Divider)`
```

ChatBox.jsx - Wappclone - Visual Studio Code

```
client > src > components > chat > chat > ChatBox.jsx > ...
1 import { useContext, useState, useEffect } from 'react';
2
3 import { Box } from '@mui/material';
4
5 import { UserContext } from '../../context/UserProvider';
6 import { AccountContext } from '../../context/AccountProvider';
7 import { getConversation } from '../../service/api';
8
9 //components
10 import ChatHeader from './ChatHeader';
11 import Messages from './Messages';
12
13 const ChatBox = () => {
14   const { person } = useContext(UserContext);
15   const { account } = useContext(AccountContext);
16
17   const [conversation, setConversation] = useState({});
18
19   useEffect(() => {
20     const getConversationDetails = async () => {
21       let data = await getConversation({ senderId: account.sub, receiverId: person.sub });
22       setConversation(data);
23     }
24     getConversationDetails();
25   }, [person.sub]);
26
27   return (
28     <Box style={{height: '75%'}}>
29       <ChatHeader person={person} />
30       <Messages person={person} conversation={conversation} />
31     </Box>
32   )
33 }
34
35 export default ChatBox;
```







```
client > src > components > chat > chat > Message.jsx > ...
37 const Time = styled(Typography)`
38   font-size: 10px;
39   color: #919191;
40   margin-top: 6px;
41   word-break: keep-all;
42   margin-top: auto;
43 `;
44
45 const Message = ({ message }) => {
46   const { account } = useContext(AccountContext);
47
48   return (
49     <div>
50       {
51         account.sub === message.senderId ?
52         <Own>
53           {
54             message.type === 'file' ? <ImageMessage message={message} /> : <TextMessage message={message} />
55           }
56         </Own>
57         :
58         <Wrapper>
59           {
60             message.type === 'file' ? <ImageMessage message={message} /> : <TextMessage message={message} />
61           }
62         </Wrapper>
63       }
64     </div>
65   )
66 }
67
68
69 const TextMessage = ({ message }) => {
70
71   return (
72     <div>
73       <Text>{message.text}</Text>
```

```
client > src > components > drawer > Profile.jsx > ...
1 import { useContext } from "react";
2 import { Box, styled, Typography } from "@mui/material"
3
4 import { AccountContext } from "../../context/AccountProvider";
5
6 const ImageContainer = styled(Box)`
7   display: flex;
8   justify-content: center;
9 `;
10
11 const Image = styled('img') ({
12   width: 200,
13   height: 200,
14   borderRadius: '50%',
15   padding: '25px 0'
16 });
17
18 const BoxWrapper = styled(Box)`
19   background: #FFFFFF;
20   padding: 12px 30px 2px;
21   box-shadow: 0 1px 3px rgba(0, 0, 0, 0.08);
22   & :first-child {
23     font-size: 13px;
24     color: #009688;
25     font-weight: 200;
26   };
27   & :last-child {
28     margin: 14px 0;
29     color: #4A4A4A;
30   }
31 `;
32
33 const DescriptionContainer = styled(Box)`
34   padding: 15px 28px 28px 30px;
35   & > p {
36     color: #8696a0;
37     font-size: 13px;
```



```

client > src > components > chat > ChatDialog.jsx > ...
32 |   borderRadius: 0,
33 |   boxShadow: 'none',
34 |   overflow: 'hidden'
35 | };
36 |
37 | const ChatDialog = () => {
38 |
39 |   const { person } = useContext(UserContext);
40 |
41 |   return (
42 |     <Dialog
43 |       open={true}
44 |       BackdropProps={{style: {backgroundColor: 'unset'}}}
45 |       PaperProps={{sx: dialogStyle }}
46 |       maxWidth={'md'}
47 |     >
48 |       <Component>
49 |         <LeftComponent>
50 |           <Menu/>
51 |         </LeftComponent>
52 |         <RightComponent>
53 |           {
54 |             Object.keys(person).length ? <ChatBox/> : <EmptyChat />
55 |           }
56 |         </RightComponent>
57 |       </Component>
58 |     </Dialog>
59 |   )
60 | }
61 |
62 | export default ChatDialog;

```

```

client > src > components > chat > ChatDialog.jsx > ...
1 | import { useContext } from 'react';
2 | import { Dialog, styled, Box } from '@mui/material';
3 |
4 | import { UserContext } from '../../context/UserProvider';
5 |
6 | //components
7 | import Menu from './menu/Menu';
8 | import ChatBox from './chat/ChatBox';
9 | import EmptyChat from './chat/Emptychat';
10 |
11 | const Component = styled(Box)`
12 |   display: flex;
13 | `;
14 |
15 | const Leftcomponent = styled(Box)`
16 |   min-width: 450px;
17 | `;
18 |
19 | const RightComponent = styled(Box)`
20 |   width: 73%;
21 |   min-width: 300px;
22 |   height: 100%;
23 |   border-left: 1px solid rgba(0, 0, 0, 0.14);
24 | `;
25 |
26 | const dialogStyle = {
27 |   height: '95%',
28 |   width: '100%',
29 |   margin: '20px',
30 |   maxWidth: '100%',
31 |   maxHeight: '100%',
32 |   borderRadius: 0,
33 |   boxShadow: 'none',
34 |   overflow: 'hidden'
35 | };
36 |
37 | const ChatDialog = () => {

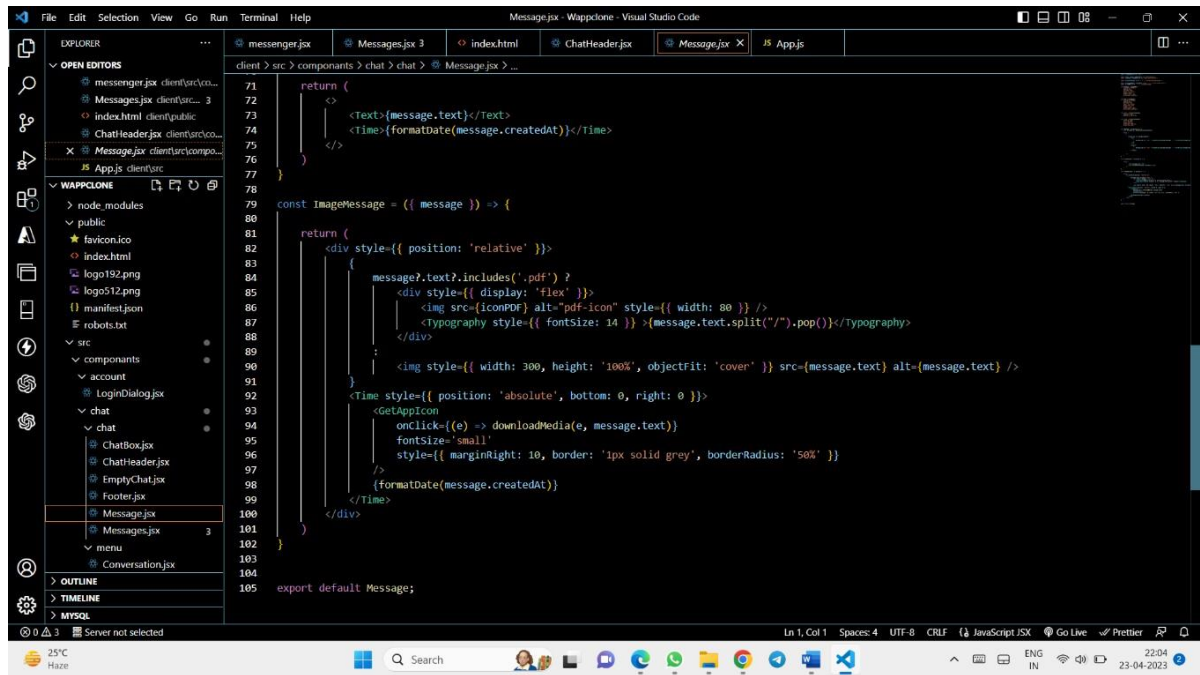
```

```
37
38 const Text = styled(Typography)`
39   display: block;
40   color: rgba(0, 0, 0, 0.6);
41   font-size: 14px;
42 `;
43
44 const Conversation = ({ user }) => {
45   const url = user.picture || emptyProfilePicture;
46
47   const { setPerson } = useContext(UserContext);
48   const { account, newMessageFlag } = useContext(AccountContext);
49
50   const [message, setMessage] = useState({});
51
52   useEffect(() => {
53     const getConversationMessage = async () => {
54       const data = await getConversation({ senderId: account.sub, receiverId: user.sub });
55       setMessage({ text: data?.message, timestamp: data?.updatedAt });
56     }
57     getConversationMessage();
58   }, [newMessageFlag]);
59
60   const getUser = async () => {
61     setPerson(user);
62     await setConversation({ senderId: account.sub, receiverId: user.sub });
63   }
64
65   return (
66     <Component onClick={() => getUser()}>
67       <Box>
68         <Image src={url} alt="display picture" />
69       </Box>
70       <Box style={{width: '100%'}}>
71         <Container>
72           <Typography>{user.name}</Typography>
73         </Container>
74       </Box>
75     </Component>
76   );
77 }
```

```
37
38 const Text = styled(Typography)`
39   display: block;
40   color: rgba(0, 0, 0, 0.6);
41   font-size: 14px;
42 `;
43
44 const Conversation = ({ user }) => {
45   const url = user.picture || emptyProfilePicture;
46
47   const { setPerson } = useContext(UserContext);
48   const { account, newMessageFlag } = useContext(AccountContext);
49
50   const [message, setMessage] = useState({});
51
52   useEffect(() => {
53     const getConversationMessage = async () => {
54       const data = await getConversation({ senderId: account.sub, receiverId: user.sub });
55       setMessage({ text: data?.message, timestamp: data?.updatedAt });
56     }
57     getConversationMessage();
58   }, [newMessageFlag]);
59
60   const getUser = async () => {
61     setPerson(user);
62     await setConversation({ senderId: account.sub, receiverId: user.sub });
63   }
64
65   return (
66     <Component onClick={() => getUser()}>
67       <Box>
68         <Image src={url} alt="display picture" />
69       </Box>
70       <Box style={{width: '100%'}}>
71         <Container>
72           <Typography>{user.name}</Typography>
73         </Container>
74       </Box>
75     </Component>
76   );
77 }
```

```
1 import { useContext, useEffect, useState } from 'react';
2
3 import { styled, Box, Typography } from "@mui/material";
4
5 import { UserContext } from '../../context/UserProvider';
6 import { AccountContext } from '../../context/AccountProvider';
7
8 import { setConversation, getConversation } from '../../service/api';
9 import { emptyProfilePicture } from '../../constants/data';
10 import { formatDate } from '../../utils/common-utils';
11
12 const Component = styled(Box)`
13   height: 45px;
14   display: flex;
15   padding: 13px 0;
16   cursor: pointer;
17 `;
18
19 const Image = styled('img') ({
20   width: 50,
21   height: 50,
22   objectFit: 'cover',
23   borderRadius: '50%',
24   padding: '0 14px'
25 });
26
27 const Container = styled(Box)`
28   display: flex;
29 `;
30
31 const Timestamp = styled(Typography)`
32   font-size: 12px;
33   margin-left: auto;
34   color: #00000099;
35   margin-right: 20px;
36 `;
37
```

```
1 import { useContext, useEffect, useState } from 'react';
2
3 import { styled, Box, Typography } from "@mui/material";
4
5 import { UserContext } from '../../context/UserProvider';
6 import { AccountContext } from '../../context/AccountProvider';
7
8 import { setConversation, getConversation } from '../../service/api';
9 import { emptyProfilePicture } from '../../constants/data';
10 import { formatDate } from '../../utils/common-utils';
11
12 const Component = styled(Box)`
13   height: 45px;
14   display: flex;
15   padding: 13px 0;
16   cursor: pointer;
17 `;
18
19 const Image = styled('img') ({
20   width: 50,
21   height: 50,
22   objectFit: 'cover',
23   borderRadius: '50%',
24   padding: '0 14px'
25 });
26
27 const Container = styled(Box)`
28   display: flex;
29 `;
30
31 const Timestamp = styled(Typography)`
32   font-size: 12px;
33   margin-left: auto;
34   color: #00000099;
35   margin-right: 20px;
36 `;
37
```



## **TESTING**

Testing is very vital for any system to be successfully implemented. The common view is that it is performed to prove that there are no errors in a program. Therefore the most useful and practical approach is with the explicit intention of finding the errors. The system is tested experimentally to ensure that the software does not fail. The system is run according to its specifications and in the way the user expects. Following testing practices are used. The system will process as normal input preparation of test-sample data.

## **STRATEGIES FOR TESTING**

### **Unit Testing**

Each and every module was intensively tested to check for errors and defects. All possible mistakes were rectified. Manually code is tested like logical errors. Once the manual checking is over the compilation has been done. Syntactical error if any has to be corrected. After the clean compilation of the program, some dummy data as per specifications has been used for testing of that module to see if it works as specified

### **Integration Testing**

Integration testing uncovers errors that arise when modules are integrated to build the overall system. The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together (called assemblages) All the unit tested modules were integrated & the errors that occurred were removed and the overall program structure was build as specified by the design.

### **System Testing**

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic. System testing is used to detect defects both within the "interassemblages" and also within the system as a whole.

## **CONCLUSION**

It was great opportunity for us as a student to learn and understand various aspects associated with project development. I did undergo from various phases of project development life cycle like analysis, design, coding, implementation, and testing. The preceding material is a sincere effort from my side to create the

**“WhatsApp Clone”** project. I got the idea about the ups and downs taking place during the project development. I analysed the problems and solved those problems that were faced in my project. The project shows the flow of each and every transaction which is being carried out by the desired user successfully thus giving him the desired result

## **FUTURE WORK**

The project has a very vast scope in future. The project can be implemented on intranet in future. Project can be updated in near future as and when requirement for the same arises, as it is very flexible in terms of expansion. With the proposed software of database Space Manager ready and fully functional the client is now able to manage and hence run the entire work in a much better, accurate and error free manner.

The following are the future scope for the project.

- 1) Should be added payment gateway
- 2) Can be added inventory management system Can be added multiple branches.
- 3) Can be added multi lingual to this application
- 4) And many features can be added this project to make it more robust.



## **REFERENCES**

- Create react app
- npm install
- React components, JSX, props for beginners
- Material-UI • Understanding React hooks
- How to use useState and useEffect with example
- React hooks API - useState, useEffect Websites:
- <https://reactjs.org/>
- [www.google.com](http://www.google.com)
- <https://www.npmjs.com/package/@react-oauth/google>
- <https://mui.com/>
- [www.projectdeveloper.com](http://www.projectdeveloper.com)

### **Faculty Guidelines:**

Mr. Mayank saxena (Technical Trainer in GLA University)

### **GitHub Repository link:**

<https://github.com/PiyushSingh1010/WhatsappCLONE>



**THANK YOU**