GOVERNMENT OF INDIA

MINISTRY OF SKILL DEVELOPMENT & ENTREPRENEURSHIP

DIRECTORATE GENERAL OF TRAINING

NATIONAL SKILL TRAINING INSTITUTE

NSTI Vidyanagar, Hyderabad - 500007

CERTIFICATE

This is to certify that following trainee have completed his project titled

"Simple Chat Application"

For IBM Program – IT, Networking and Cloud (Technical Diploma)

ROLL NO	NAME
ADIT19AU00581	DINESH MAMIDALA
ADIT19AU04084	KORAM DHASARATH

Mr. Satish P. Pise	Miss. Trupti Pawar	Mr. Gulab Chandra
IBM Faculty	TO / ADIT	AD / Section In-charge

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to several individuals and organizations for supporting me throughout my diploma study. First, I wish to express my sincere gratitude to my trainer, Mr. Satish Pise, for his enthusiasm, patience, insightful comments, helpful information, practical advice and unceasing ideas that have helped me tremendously at all times in my study and writing of this project report. His immense knowledge, profound experience and professional expertise in computer science has enabled me to complete this project successfully.

I also wish to express my sincere thanks to the National Skill Training Institute, Mumbai for accepting me into the diploma program. In addition, I am deeply indebted to the Ministry of Skill Development & Entrepreneurship and IBM for granting me the diploma course. This technical and financial support has enabled me to complete my diploma course studies successfully. Also, I am grateful the Miss. Trupti Pawar faculty of NSTI Mumbai for supporting me for course completion in the specific subject.

INDEX

CHAPTER 1

INTRODUCTION

1) Overview

The functionality of the chat application is to give the ability to chat with whoever is online on the application. The users and stakeholders will be a small group for now, The use cases will be what is available to the user, and the functional/nonfunctional requirements will be covered, as well as the milestones of the chat application.

Feature

Get one of the best chat apps loaded onto your phone, and you never have to worry about staying in touch. And staying in touch with friends and family is exactly what you'll want to do, as the coronavirus pandemic continues to keep us from being together in person.

There's no shortage of chat apps available for both Android and iPhones, but the best ones blend texting, voice and video chat. Some focus on the ability to play games with the folks on the other end of the line, while others make it easy to swap pictures. And some chat apps have even gotten into collaboration and productivity, helping people stay connected with colleagues when they're stuck working remotely.

- Best video chat apps let you see who you're talking to
- Find the best WhatsApp alternatives

The default chat apps on your phone — both Apple and Google make their own Messages app for iOS and Android, respectively — pack in a lot of features. But if you're looking for a different focus or some speciality capabilities, these are the best chat apps for taking your conversations in a different direction.

2) Advantages

Faster support.

Real-time text preview.

No waiting queues.

Non-intrusive

On-site.

3) Scope

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 java, but due to the lack of experience in java, 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. A database for the users registered will be developed and tested, a menu will be developed and tested, a client/server interface will be developed and tested, and GUI's will be developed and tested, for the users' benefits. When the chat application is near completion, more testing will be done in order to make it less buggy or more user friendly. Target Audience The target audience is any person who wants to use a chat application. Terms and Definitions GUI - Graphic User Interface

4) Future Work

With the knowledge I have gained by developing this **application**, I am confident that in the **future** I can make the **application** more effectively by adding these services. Increasing the effectiveness of the **application** by providing Voice **Chat**.

CHAPTER 2

SERVICES AND TOOLS REQUIRED

2.1 Services Used

2.1.1 Liberty Profile

IBM WebSphere Application Server V8.5 Liberty profile is a flexible and dynamic server profile of WAS which enables the WAS server to deploy only required custom features instead of deploying a big set of available JEE components.

2.2 Tools and Software's used

2.2.1 NodeJS

Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser.

2.2.2HTML

HTML is a programming language which was developed by Berners-Lee in 1980 but standardized in 1995. HTML stands for Hyper Text Markup Language. It is one of the most popular languages on the planet used for developing web pages and websites. HTML is used for the structure of web documents i.e., headlines, paragraphs and images etc. HTML is known as the basics of all programming language in the globe. HTML is highly used in other programming languages such as JavaScript and CSS for the its display attributes.

HTML is a language for describing web pages.

- HTML stands for Hyper Text Markup Language
- HTML is a markup language
- A markup language is a set of markup tags
- The tags describe document content
- HTML documents contain HTML tags and plain text
- HTML documents are also called web pages

HTML Tags:

HTML markup tags are usually called HTML tags.

- HTML tags are keywords (tag names) surrounded by angle brackets like <html>

- HTML tags normally come in pairs like and
- The first tag in a pair is the start tag, the second tag is the end tag
- The end tag is written like the start tag, with a forward slash before the tag name
- Start and end tags are also called opening tags and closing tags

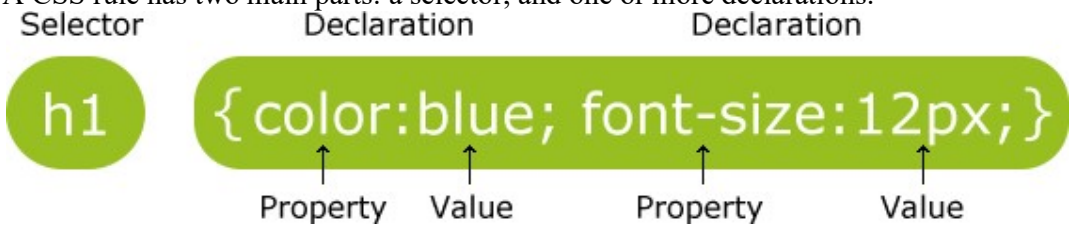
<Tag name>content</Tag name>

CSS:

- CSS stands for Cascading Style Sheets
- Styles define how to display HTML elements
- Styles were added to HTML 4.0 to solve a problem
- External Style Sheets can save a lot of work
- External Style Sheets are stored in CSS files

CSS Syntax:

A CSS rule has two main parts: a selector, and one or more declarations:



- The selector is normally the HTML element you want to style.
- Each declaration consists of a property and a value.
- The property is the style attribute you want to change. Each property has a value.

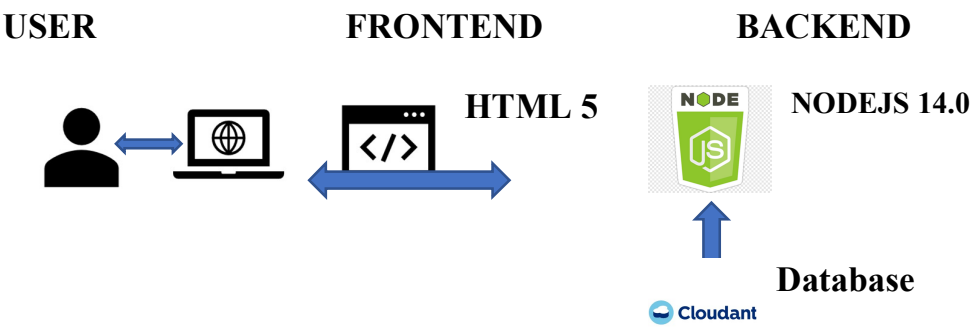
2.2.3CloudFoundry

Cloud Foundry is an open source, multi-cloud application platform as a service governed by the Cloud Foundry Foundation, a 501 organization. The software was originally developed by VMware, transferred to Pivotal Software, who then transferred the software to the Cloud Foundry Foundation upon its inception in 2015.

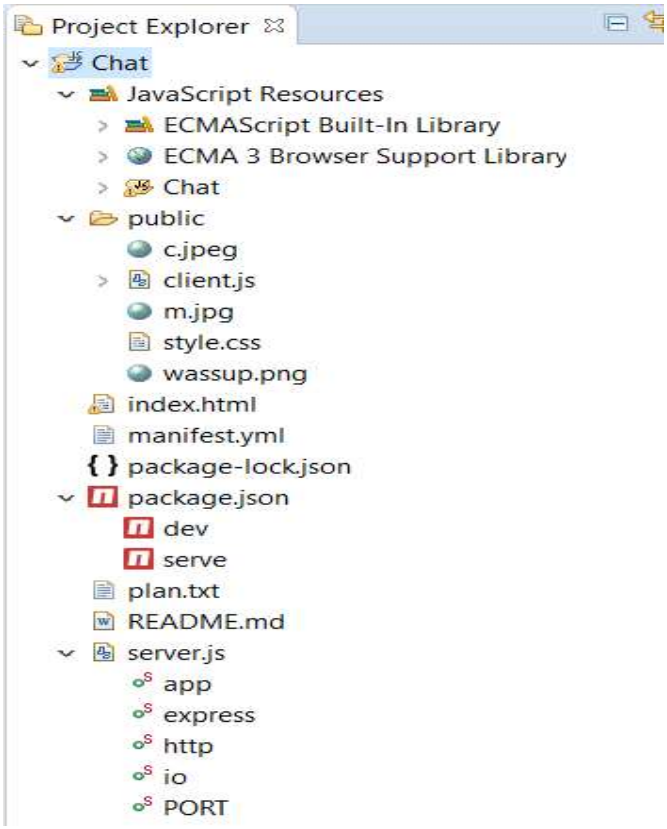
CHAPTER 3

PROJECT ARCHITECTURE

3.1 Architecture



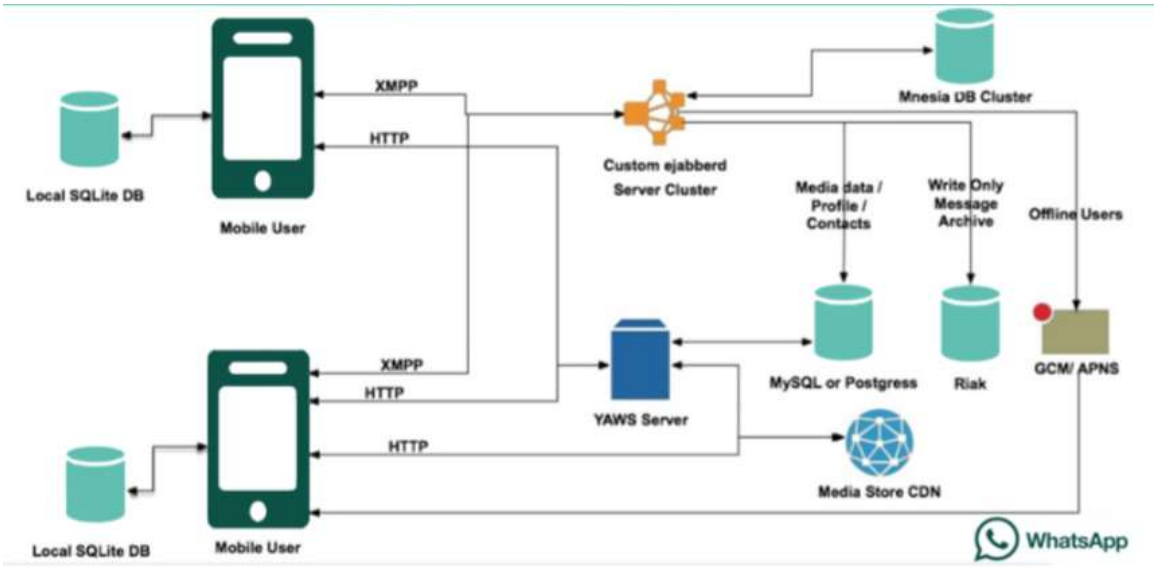
Architecture changes applied as per products used



CHAPTER 4

ARCHITECTURE BLOCKS DETAIL WORKING

4.1 Blocks



MVC stands for Model View Controller. It is an architectural design of an application. All these three components are essential for creating and working together for all web applications

Model: It is the basic part of the architectural design that controls the logic and behavior of the application, View: It displays the model´s data for the user in the application, Controller: It is the major part of the architecture design because if offers the interface between view and model

CHAPTER 5

SOURES CODE

INDEX.HTML

<!DOCTYPE html>

```
<html lang="en">
<head>
   <meta charset="UTF-8">
  <meta name="description" content="Free Web tutorials">
  <meta name="keywords" content="HTML, CSS, JavaScript">
  <meta name="author" content="John Doe">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Dinesh Mamidala</title>
   <link rel="stylesheet" href="/style.css">
</head>
<body>
<style>
body {
  background-image: url('m.jpg');
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-size: 100% 100%;
}
.message__area{
          background-image: url('c.jpg');
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-size: 100% 100%;
   padding: 20px;
```

```
    border-radius: 4px;
    margin-bottom: 40px;
    max-width: 300px;
    position: relative;
</style>
    <section class="chat__section">
        <div class="brand">
            <img height="40" src="/wassup.png" alt="">
            <h1>Mamidala's Messenger</h1>
        </div>
        <div style="background-image: url('c.jpg');" class="message__area"></div>
        <div class="FileUpload">
            <textarea id="textarea" cols="30" rows="1" placeholder="Write a message..."
type="file" id="file"></textarea>
        </div>
    </section>
    <script src="/socket.io/socket.io.js"></script>
    <script src="/client.js"></script>
</body>
</html>
```

SERVER.JS

```
const express = require('express')
const app = express()
```

```
const http = require('http').createServer(app)

const PORT = process.env.PORT || 3000

http.listen(PORT, () => {
    console.log(`Listening on port ${PORT}`)
})

app.use(express.static(__dirname + '/public'))

app.get('/', (req, res) => {
    res.sendFile(__dirname + '/index.html')
})

// Socket
const io = require('socket.io')(http)

io.on('connection', (socket) => {
    console.log('Connected...')
    socket.on('message', (msg) => {
        socket.broadcast.emit('message', msg)
    })
})
```

PUBLIC FLODER>

CLIENT.JS

```
const socket = io()

let name;

let textarea = document.querySelector('#textarea')

let messageArea = document.querySelector('.message__area')

do {
```

```
    name = prompt('Please enter your name: ')
} while(!name)
textarea.addEventListener('keyup', (e) => {
    if(e.key === 'Enter') {
        sendMessage(e.target.value)
    }
})
function sendMessage(message) {
    let msg = {
        user: name,
        message: message.trim()
    }
    // Append
    appendMessage(msg, 'outgoing')
    textarea.value = ''
    scrollToBottom()

    // Send to server
    socket.emit('message', msg)
}
function appendMessage(msg, type) {
    let mainDiv = document.createElement('div')
    let className = type
    mainDiv.classList.add(className, 'message')
```

```
    let markup = `
        <h4>${msg.user}</h4>
        <p>${msg.message}</p>
    mainDiv.innerHTML = markup
    messageArea.appendChild(mainDiv)
}
// Recieve messages
socket.on('message', (msg) => {
    appendMessage(msg, 'incoming')
    scrollToBottom()
})
function scrollToBottom() {
    messageArea.scrollTop = messageArea.scrollHeight
}
```

STYLE.CSS

```
@import url('https://fonts.googleapis.com/css2?family=Roboto&display=swap');
* {
    padding: 0;
    margin: 0;
    box-sizing: border-box;
}
body {
    background-image: url('m.jpeg');
    background-repeat: no-repeat;
    background-attachment: fixed;
    background-size: cover;
```

```
    display: flex;
    align-items: center;
    justify-content: center;
    min-height: 100vh;
    background: #F8F8F8;
    font-family: 'Roboto', sans-serif;
}
section.chat__section {
    width: 800px;
    max-width: 90%;
    background: #fff;
    box-shadow: 0 10px 15px -3px rgba(0, 0, 0, 0.1), 0 4px 6px -2px rgba(0, 0, 0, 0.05);
}
.brand {
    padding: 20px;
    background: #80ced6;
    display: flex;
    align-items: center;
}
.brand h1 {
    text-transform: uppercase;
    font-size: 20px;
    color: #444;
    margin-left: 10px;
```

```
}
.message__area{
    background-image: url('c.jpeg');
    background-repeat: no-repeat;
    background-attachment: fixed;
    background-size: cover;
    height: 500px;
    padding: 16px;
    display: flex;
    flex-direction: column;
    overflow-y: auto;
    padding-top: 40px;
}
textarea {
    width: 100%;
    border: none;
    padding: 20px;
    font-size: 16px;
    outline: none;
    background: #FBFBFB;
}
.message {
    padding: 20px;
    border-radius: 4px;
    margin-bottom: 40px;
```

```
    max-width: 300px;
    position: relative;
}
.incoming {
    background: #8F8BE8;
    color: #fff;
}
.outgoing {
    background: #e9eafd;
    color: #787986;
    margin-left: auto;
}
.message h4 {
    position: absolute;
    top: -20px;
    left: 0;
    color: #333;
    font-size: 14px;
}
```

PACKEGES.JSON

```
{
  "name": "wassup",
```

```
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "dev": "nodemon server",
    "serve": "node server"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "express": "^4.17.1",
    "socket.io": "^2.3.0"
  },
  "devDependencies": {
    "nodemon": "^2.0.3"
  }
}
```

CHAPTER 6

CONCLUSION

In several ways, instant messaging apps are the future of communications. This is due to the fact that the messaging market is continuously showing evolution. However, there is still a chance of building an app like WhatsApp, Telegram, and Facebook Messenger that will surely get a success like these game-changers. Apart from fierce competition, there is still space for quality messenger and chat apps that give users a new experience. So, if you might have any question like –

- How much does it cost to create a chatting app or social media app?
- What other unique features and functionalities I can include in chat and messenger apps?
- What is the importance of developing an MVP (Minimum Viable Product) of social networking apps?

Then, you can get in touch with us as we are a leading Android/iOS app development company, and have already developed over 3500 mobile applications in diverse mobile app categories.

In case, if you have any query or confusion regarding messenger/chat app development, just fill-up the below-given form and one of our sales representatives will revert you within in 16 hours with an optimal solution. The consultation is absolutely free of cost.

CHAPTER 7

REFERENCES

1) GitHub
2) Google
3) Lab manuals
4) YouTube
5) IBM COLUD

CHAPTER 7

SCREENSHOTS

1. Install NodeJS - https://nodejs.org/en/download/ as per your OS

2. Installing the stand-alone IBM Cloud CLI –

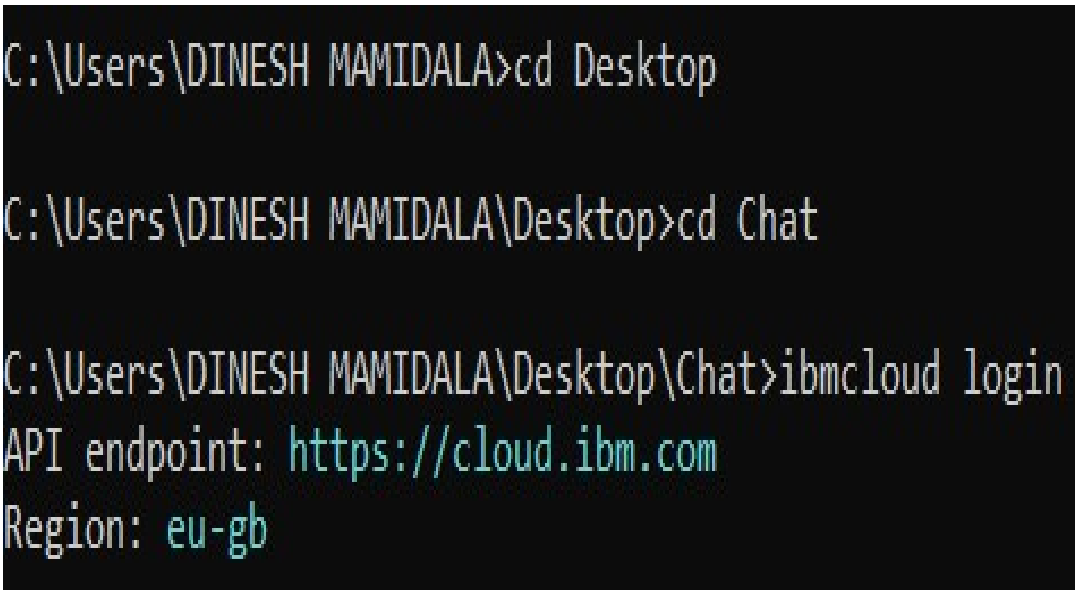
https://github.com/IBM-Cloud/ibm-cloudcli-release/releases/

Select setup as per your OS

3. Download Code from https://github.com/DineshMamidala/CloudChat.io

4. Extract file in any location and rename folder Cloud Chat-master to Cloud Chat

5. Open folder Cloud Chat in command prompt



6. Login ibmcloud using username and password

```
Email> mamidaladinesh2266@gmail.com

Password>
Authenticating...
OK

Targeted account DINESH MAMIDALA's Account (81147bc788954370a87c71b779a09a16)


API endpoint:      https://cloud.ibm.com
Region:            eu-gb
User:              mamidaladinesh2266@gmail.com
Account:           DINESH MAMIDALA's Account (81147bc788954370a87c71b779a09a16)
Resource group:    No resource group targeted, use 'ibmcloud target -g RESOURCE_GROUP'
CF API endpoint:
Org:
Space:
```

7. Type ibmcloud target --cf

```
C:\Users\DINESH MAMIDALA\Desktop\Chat>ibmcloud target --cf
Targeted Cloud Foundry (https://api.eu-gb.cf.cloud.ibm.com)

Targeted org mamidaladinesh2266@gmail.com

Targeted space dev



API endpoint:      https://cloud.ibm.com
Region:            eu-gb
User:              mamidaladinesh2266@gmail.com
Account:           DINESH MAMIDALA's Account (81147bc788954370a87c71b779a09a16)
Resource group:    No resource group targeted, use 'ibmcloud target -g RESOURCE_GROUP'
CF API endpoint:   https://api.eu-gb.cf.cloud.ibm.com (API version: 2.164.0)
Org:               mamidaladinesh2266@gmail.com
Space:             dev
```

Type ibmcloud target -g Default

```
Targeted org mamidaladinesh2266@gmail.com

Targeted space dev



API endpoint:      https://cloud.ibm.com
Region:            eu-gb
User:              mamidaladinesh2266@gmail.com
Account:           DINESH MAMIDALA's Account (81147bc788954370a87c71b779a09a16)
Resource group:    No resource group targeted, use 'ibmcloud target -g RESOURCE_GROUP'
CF API endpoint:   https://api.eu-gb.cf.cloud.ibm.com (API version: 2.164.0)
Org:               mamidaladinesh2266@gmail.com
Space:             dev
```

8. Type ibmcloud resource groups show all resource groups

9. Type ibmcloud resource groups –default List default group of currently targeted account

10. Deploying contents to ibm cloud Type ibmcloud cf push command

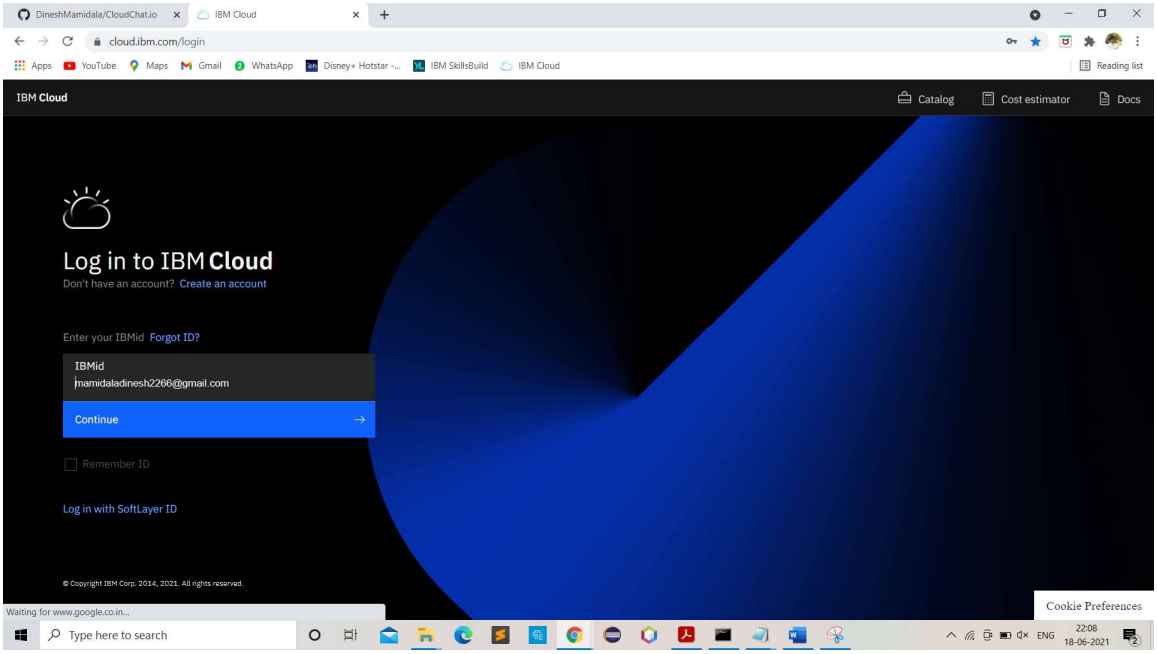
After command execution

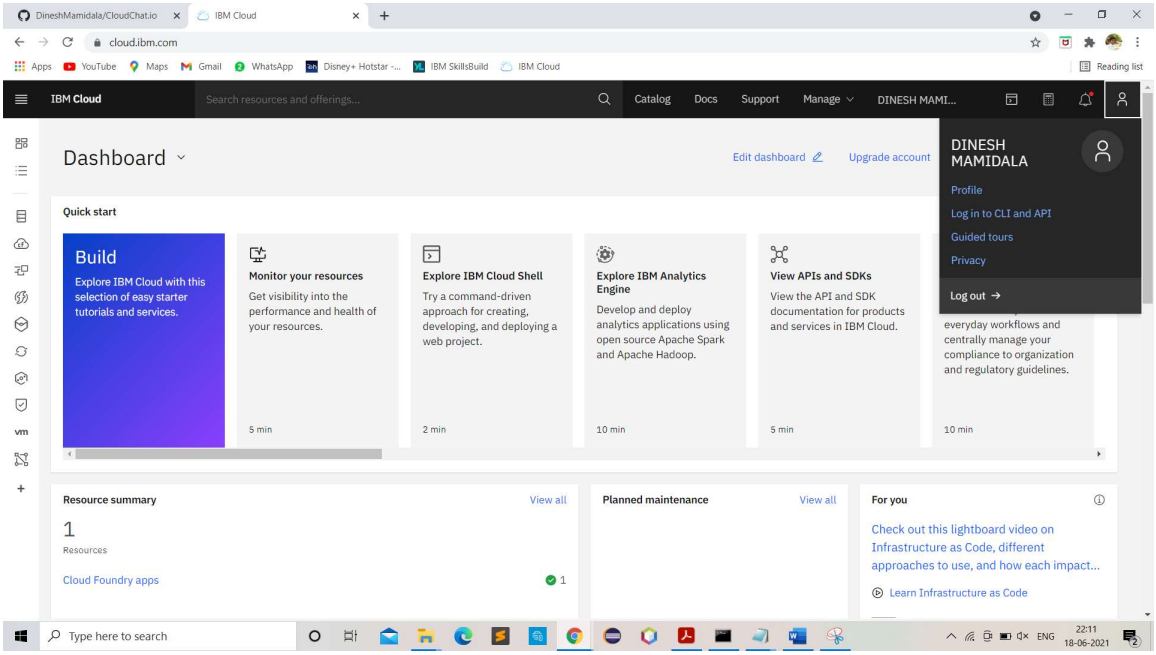
```
C:\Users\DINESH MAMIDALA\Desktop\Chat>ibmcloud cf push
Invoking 'cf push'...

Pushing from manifest to org mamidaladinesh2266@gmail.com / space dev as mamidaladinesh2266@gmail.com...
Using manifest file C:\Users\DINESH MAMIDALA\Desktop\Chat\manifest.yml
Getting app info...
Creating app with these attributes...
+ name:        PersonalChat
  path:        C:\Users\DINESH MAMIDALA\Desktop\Chat
+ instances:   1
+ memory:      256M
  routes:
+   personalchat-responsible-jaguar-yy.eu-gb.mybluemix.net

Creating app PersonalChat...
Mapping routes...
Comparing local files to remote cache...
Packaging files to upload...
Uploading files...
 232.71 KiB / 232.71 KiB [============================================================] 100.00% 12s
```

11. After successful command execution and content uploaded in ibmcloud

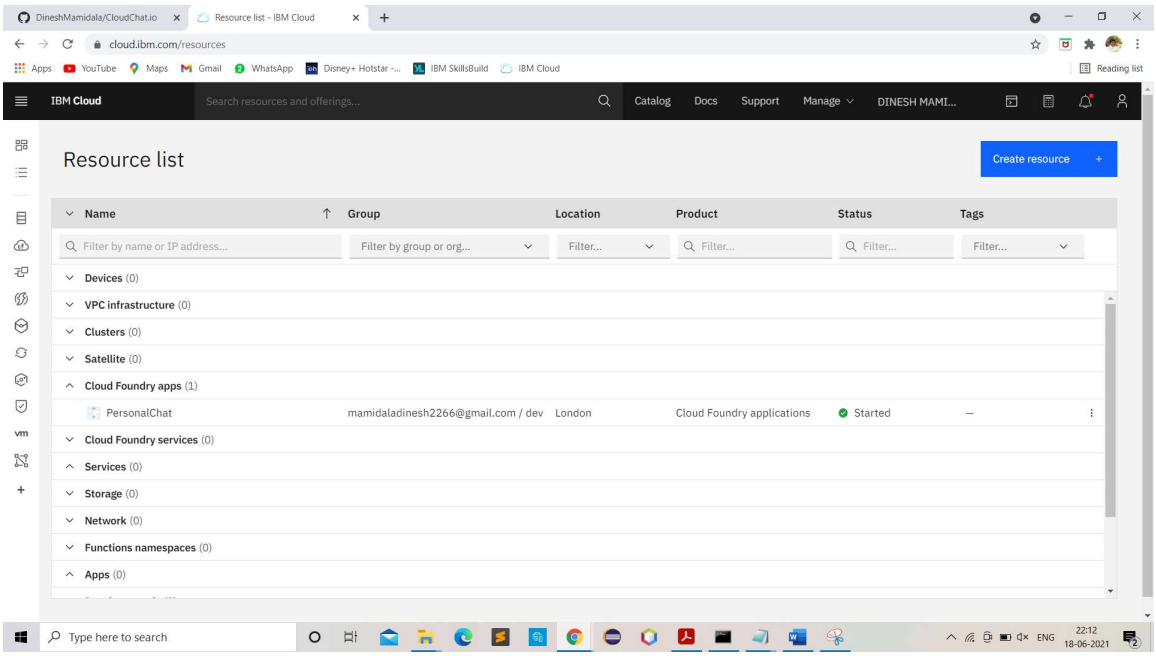
```
Waiting for app to start...

name:              PersonalChat
requested state:   started
routes:            personalchat-responsible-jaguar-yy.eu-gb.mybluemix.net
last uploaded:     Fri 18 Jun 21:28:15 IST 2021
stack:             cflinuxfs3
buildpacks:        nodejs

type:            web
instances:       1/1
memory usage:    256M
start command:   npm start
     state     since                  cpu    memory   disk     details
#0   running   2021-06-18T15:58:30Z   0.0%   0 of 0   0 of 0
```

12. Login ibm cloud website



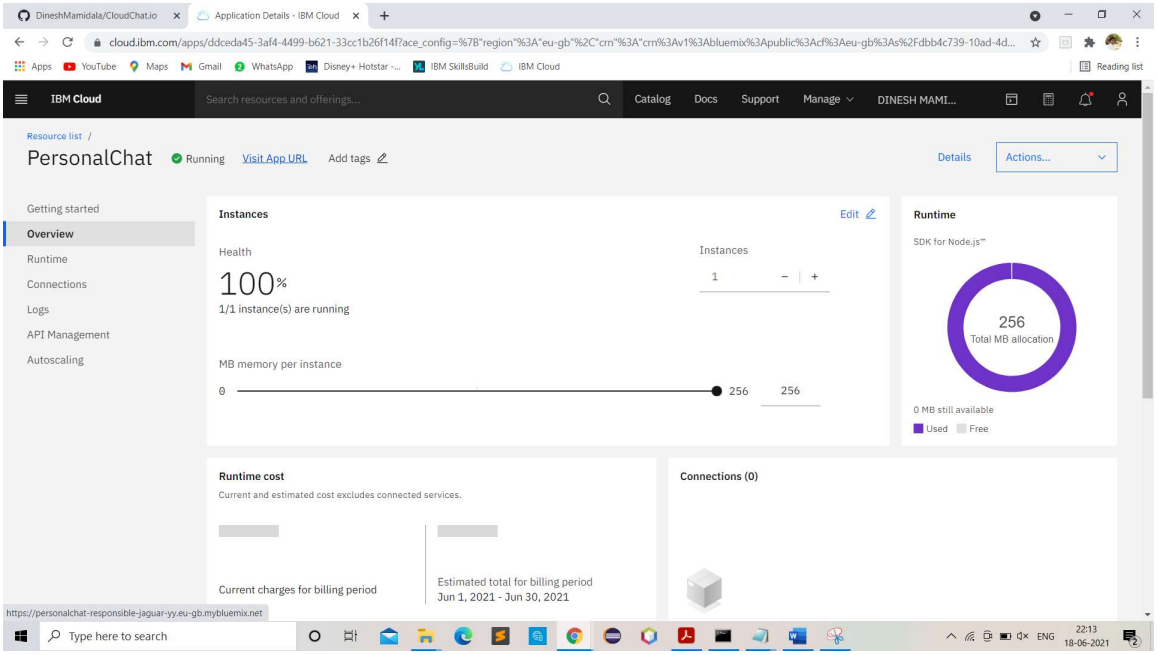


12.1. View resources list

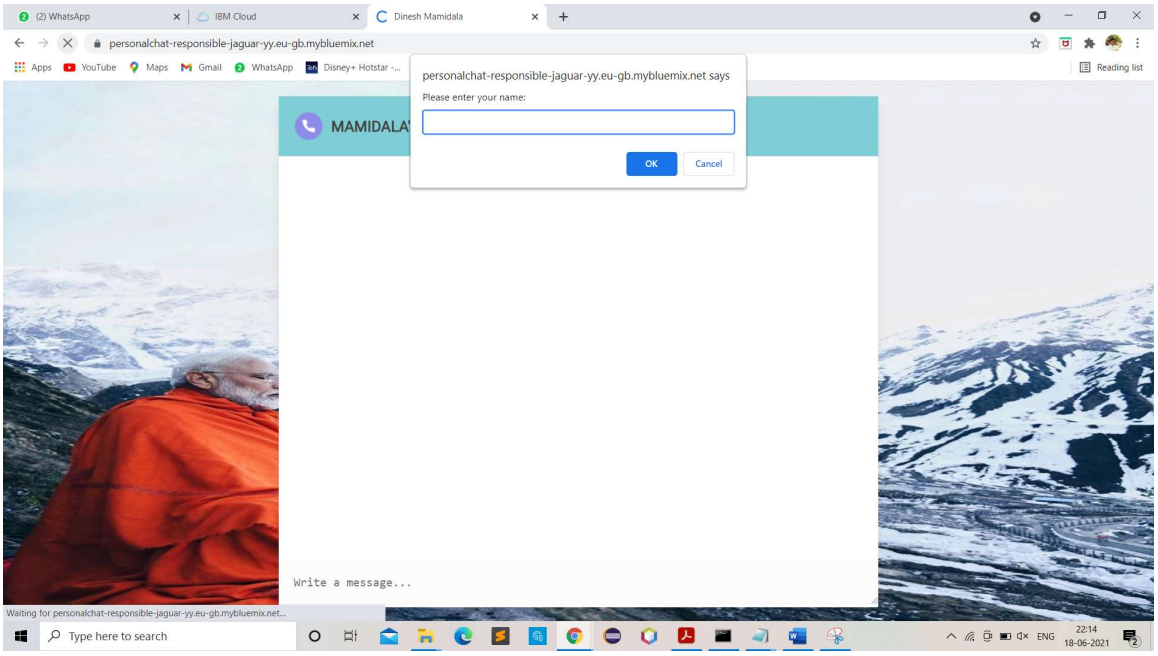


13. Cloud Foundry app running.

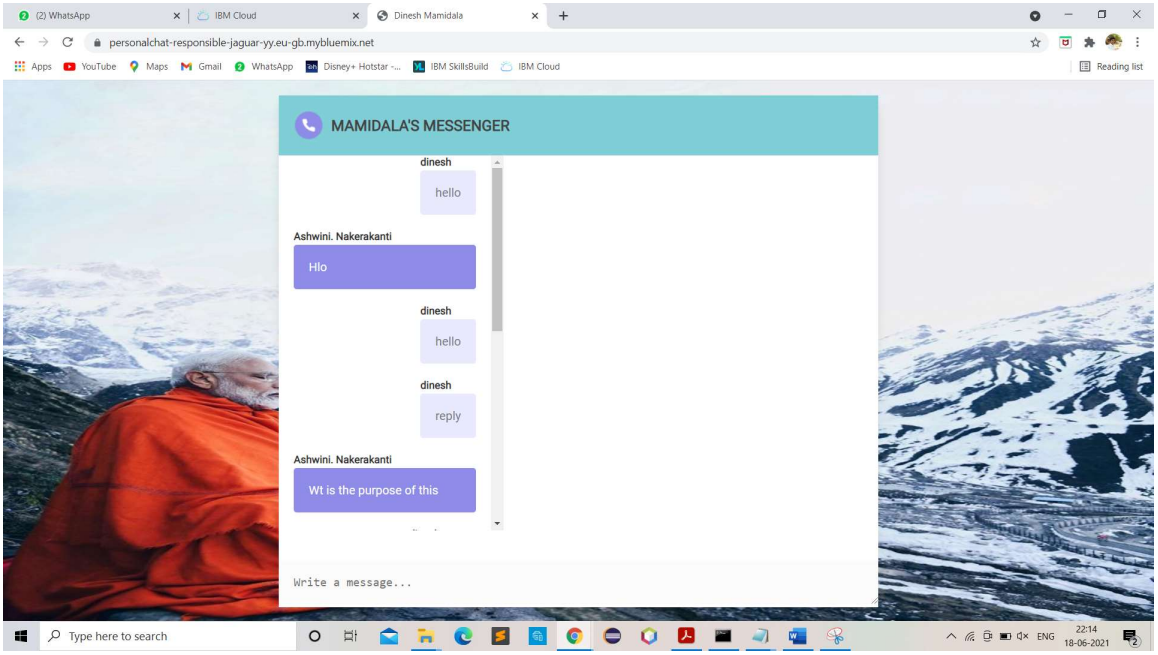
Click on CloudChat



14. Click on Visit App URL and view your application running successfully



15. Open app using URL in different locations or browser and test result.


(2) WhatsApp
IBM Cloud
Dinesh Mamidala
personalchat-responsible-jaguar-yy.eu-gb.mybluemix.net
Apps
YouTube
Maps
Gmail
WhatsApp
Disney+ Hotstar -...
IBM SkillsBuild
IBM Cloud
Reading list
MAMIDALA'S MESSENGER
dinesh
hello
Ashwini. Nakerakanti
Hlo
dinesh
hello
dinesh
reply
Ashwini. Nakerakanti
Wt is the purpose of this
Write a message...
Type here to search
ENG
22:14
18-06-2021