server.js

```
const express = require('express');
const path = require('path');
const http = require('http');
const socketio = require('socket.io');
const formatMessage = require('./utils/chatMessage');
const mongoClient = require('mongodb').MongoClient;

const dbname = 'chatApp';
const chatCollection = 'chats'; //collection to store all chats
const userCollection = 'onlineUsers'; //collection to maintain list of currently online users

const port = 5000;
const database = 'mongodb://localhost:27017/';
const app = express();

const server=http.createServer(app);
const io = socketio(server);

io.on('connection', (socket) => {
    console.log('New User Logged In with ID '+socket.id);

    //Collect message and insert into database
    socket.on('chatMessage', (data) =>{ //recieves message from client-end along with sender's and reciever's details
        var dataElement = formatMessage(data);
        mongoClient.connect(database, (err,db) => {
            if (err)
                throw err;
            else {
                var onlineUsers = db.db(dbname).collection(userCollection);
                var chat = db.db(dbname).collection(chatCollection);
                chat.insertOne(dataElement, (err,res) => { //inserts message to into the database
                    if(err) throw err;
                    socket.emit('message',dataElement); //emits message back to the user for display
                });
                onlineUsers.findOne({"name":data.toUser}, (err,res) => { //checks if the recipient of the message is on
                    if(err) throw err;
                    if(res!=null) //if the recipient is found online, the message is emmitted to him/her
                        socket.to(res.ID).emit('message',dataElement);
```

server.js

```
32                  if(err) throw err;
33                  socket.emit('message',dataElement); //emits message back to the user for display
34              });
35              onlineUsers.findOne({"name":data.toUser}, (err,res) => { //checks if the recipient of the message is on
36                  if(err) throw err;
37                  if(res!=null) //if the recipient is found online, the message is emmitted to him/her
38                      socket.to(res.ID).emit('message',dataElement);
39              });
40          }
41          db.close();
42      });
43
44  });
45
46  socket.on('userDetails',(data) => { //checks if a new user has logged in and recieves the established chat details
47      mongoClient.connect(database, (err,db) => {
48          if(err)
49              throw err;
50          else {
51              var onlineUser = { //forms JSON object for the user details
52                  "ID":socket.id,
53                  "name":data.fromUser
54              };
55              var currentCollection = db.db(dbname).collection(chatCollection);
56              var online = db.db(dbname).collection(userCollection);
57              online.insertOne(onlineUser,(err,res) =>{ //inserts the logged in user to the collection of online user
58                  if(err) throw err;
59                  console.log(onlineUser.name + " is online...");
60              });
61              currentCollection.find({ //finds the entire chat history between the two people
62                  "from" : { "$in": [data.fromUser, data.toUser] },
63                  "to" : { "$in": [data.fromUser, data.toUser] }
64              },{projection: {_id:0}}).toArray((err,res) => {
65                  if(err)
66                      throw err;
67                  else {
68                      //console.log(res);
```

server.js

```
            });
            currentCollection.find({ //finds the entire chat history between the two people
                "from" : { "$in": [data.fromUser, data.toUser] },
                "to" : { "$in": [data.fromUser, data.toUser] }
            },{projection: {_id:0}}).toArray((err,res) => {
                if(err)
                    throw err;
                else {
                    //console.log(res);
                    socket.emit('output',res); //emits the entire chat history to client
                }
            });
        }
        db.close();
    });
});
var userID = socket.id;
socket.on('disconnect', () => {
    mongoClient.connect(database, function(err, db) {
        if (err) throw err;
        var onlineUsers = db.db(dbname).collection(userCollection);
        var myquery = {"ID":userID};
        onlineUsers.deleteOne(myquery, function(err, res) { //if a user has disconnected, he/she is removed from th
          if (err) throw err;
          console.log("User " + userID + "went offline...");
          db.close();
        });
      });
});
});

app.use(express.static(path.join(__dirname,'front')));

server.listen(port, () => {
    console.log(`Chat Server listening to port ${port}...`);
});
```

server.js

```
                    throw err;
                else {
                    //console.log(res);
                    socket.emit('output',res); //emits the entire chat history to client
                }
            });
        }
        db.close();
    });
});
var userID = socket.id;
socket.on('disconnect', () => {
    mongoClient.connect(database, function(err, db) {
        if (err) throw err;
        var onlineUsers = db.db(dbname).collection(userCollection);
        var myquery = {"ID":userID};
        onlineUsers.deleteOne(myquery, function(err, res) { //if a user has disconnected, he/she is removed from th
          if (err) throw err;
          console.log("User " + userID + "went offline...");
          db.close();
        });
      });
});
});

app.use(express.static(path.join(__dirname,'front')));

server.listen(port, () => {
    console.log(`Chat Server listening to port ${port}...`);
});
```

package.json

package.json > ...

```
{
  "name": "chat-application",
  "version": "1.0.0",
  "description": "Creating a Chat Application using Node.js, Express, Socket.io, MongoDB and HTML & CSS",
  "main": "server.js",
  ▷ Debug
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "node server",
    "dev": "nodemon server"
  },
  "author": "Sumit Jaiswal",
  "license": "ISC",
  "dependencies": {
    "express": "^4.17.1",
    "moment": "^2.26.0",
    "mongodb": "^3.5.9",
    "socket.io": "^2.3.0"
  },
  "devDependencies": {
    "nodemon": "^2.0.4"
  }
}
```

package.json | package-lock.json

package-lock.json > ...

```
{
  "name": "chat-application",
  "version": "1.0.0",
  "lockfileVersion": 2,
  "requires": true,
  "packages": {
    "": {
      "name": "chat-application",
      "version": "1.0.0",
      "license": "ISC",
      "dependencies": {
        "express": "^4.17.1",
        "moment": "^2.26.0",
        "mongodb": "^3.5.9",
        "socket.io": "^2.3.0"
      },
      "devDependencies": {
        "nodemon": "^2.0.4"
      }
    },
    "node_modules/@sindresorhus/is": {
      "version": "0.14.0",
      "resolved": "https://registry.npmjs.org/@sindresorhus/is/-/is-0.14.0.tgz",
      "integrity": "sha512-9NET910DNaIPngYnLLPeg+Ogzqsi9uM4mSboU5y6p8S5DzMTVEsJZrawi+BoDNUVBa2DhJqQYUFvMDfgU062LQ==",
      "dev": true,
      "engines": {
        "node": ">=6"
      }
    },
    "node_modules/@szmarczak/http-timer": {
      "version": "1.1.2",
      "resolved": "https://registry.npmjs.org/@szmarczak/http-timer/-/http-timer-1.1.2.tgz",
      "integrity": "sha512-XIB2XbzHTN6ieIjfIMV9hlVcfPU26s2vafYWQcZHWXHOxiaRZYEDKEwdl129Zyg50+foYV2jCgtrqSA6qNuNSA==",
      "dev": true,
      "dependencies": {
        "defer-to-connect": "^1.0.1"
      },
      "engines": {
```

Ln 1, Col 1 Spaces: 2 UTF-8 LF { } JSON

38°C Sunny

package.json | package-lock.json

package-lock.json > ...

```
      "integrity": "sha512-XIB2XbzHTN6ieIjfIMV9hlVcfPU26s2vafYWQcZHWXHOxiaRZYEDKEwdl129Zyg50+foYV2jCgtrqSA6qNuNSA==",
      "dev": true,
      "dependencies": {
        "defer-to-connect": "^1.0.1"
      },
      "engines": {
        "node": ">=6"
      }
    },
    "node_modules/@types/color-name": {
      "version": "1.1.1",
      "resolved": "https://registry.npmjs.org/@types/color-name/-/color-name-1.1.1.tgz",
      "integrity": "sha512-rr+OQyAjxze7GgWrSaJwydHStIhHq2lvY3BOC2Mj7KnzI7XK0Uw1TOOdI9lDoajEbSWLiYgoo4f1R51erQfhPQ==",
      "dev": true
    },
    "node_modules/abbrev": {
      "version": "1.1.1",
      "resolved": "https://registry.npmjs.org/abbrev/-/abbrev-1.1.1.tgz",
      "integrity": "sha512-nne9/IiQ/hzIhY6pdDnbBtz7DjPTKrY00P/zvPSm5pOFkl6xuGrGnXn/VtTNNfNtAfZ9/1RtehkszU9qcTii0Q==",
      "dev": true
    },
    "node_modules/accepts": {
      "version": "1.3.7",
      "resolved": "https://registry.npmjs.org/accepts/-/accepts-1.3.7.tgz",
      "integrity": "sha512-Il80Qs2WjYlJIBNzNkK6KYqlVMTbZLXgHx2oT0pU/fjRHyEp+PEfEPY0R3WCwAGVOtauxh1hOxNgIf5bv7dQpA==",
      "dependencies": {
        "mime-types": "~2.1.24",
        "negotiator": "0.6.2"
      },
      "engines": {
        "node": ">= 0.6"
      }
    },
    "node_modules/after": {
      "version": "0.8.2",
      "resolved": "https://registry.npmjs.org/after/-/after-0.8.2.tgz",
      "integrity": "sha1-/ts5T58OAqqXaOcCvaI7UF+ufh8="
```

```
    },
    "node_modules/ansi-align": {
      "version": "3.0.0",
      "resolved": "https://registry.npmjs.org/ansi-align/-/ansi-align-3.0.0.tgz",
      "integrity": "sha512-ZpClVKqXN3RGBmKibdfWzqCY4lnjEuoNzU5T0oEFpfd/z5qJHVarukridD4juLO2FXMiwUQxr9WqQtaYa8XRYw==",
      "dev": true,
      "dependencies": {
        "string-width": "^3.0.0"
      }
    },
    "node_modules/ansi-align/node_modules/string-width": {
      "version": "3.1.0",
      "resolved": "https://registry.npmjs.org/string-width/-/string-width-3.1.0.tgz",
      "integrity": "sha512-vafcv6KjVZKSgz06oM/H6GDBrAtz8vdhQakGjFIvNrHA6y3HCF1CInLy+QLq8dTJPQ1b+KDUqDFctkdRW44e1w==",
      "dev": true,
      "dependencies": {
        "emoji-regex": "^7.0.1",
        "is-fullwidth-code-point": "^2.0.0",
        "strip-ansi": "^5.1.0"
      },
      "engines": {
        "node": ">=6"
      }
    },
    "node_modules/ansi-regex": {
      "version": "4.1.0",
      "resolved": "https://registry.npmjs.org/ansi-regex/-/ansi-regex-4.1.0.tgz",
      "integrity": "sha512-1apePfXM1UOSqw0o9IiFAovVz9M5S1Dg+4TrDwfMewQ6p/rmMueb7tWZjQ1rx4Loy1ArBggoqGpfqqdI4rondg==",
      "dev": true,
      "engines": {
        "node": ">=6"
      }
    },
    "node_modules/ansi-styles": {
      "version": "4.2.1",
      "resolved": "https://registry.npmjs.org/ansi-styles/-/ansi-styles-4.2.1.tgz",
      "integrity": "sha512-9VGjrMsG1vePxcSweQsN20KY/c4zN0h9fLjqAbwbPfahM3t+NL+M9HC8xeXG2I8pX5NoamTGNuomEUFI7fcUjA==",
      "dev": true,
```