2021 Create Project Program Code

Code is spread across 4 files. The Python and Javascript files are the actual program code, but the HTML and CSS are just styling for the page. The path location of the files from the project directory is in the name of the file.

server.py

```
#!/usr/bin/env python3
# server.py

# 2021 CSP Create Project
# A realtime internet chat room using WebSockets handled with Flask SocketIO
# Built on a Flask Python server with a JavaScript client

# Libraries Flask and Flask SocketIO must be installed using PIP, the Python Package Manager
# This project is designed to be ran on a UNIX system, tested on MacOS 11 and Ubuntu 18.04 LTS

# Flask <https://flask.palletsprojects.com/en/1.1.x/>
# Flask SocketIO <https://flask-socketio.readthedocs.io/en/latest/>

from flask import Flask, render_template, url_for   # Flask handles hosting of webserver
                                                     # install with: pip install flask
from flask_socketio import SocketIO, emit            # Flask SocketIO handles WebSocket Interface
                                                     # install with: pip install flask_socketio
from datetime import datetime                        # Datetime handles dates and times
                                                     # preinstalled

# Setup Socket Enviorment with App
app = Flask(__name__)
socketio = SocketIO(app)


# Procedure makeSafe
# Parameter unsafeText is the text from the user
# Returns safeText, the safe unsafeText
# Iterates through string of text to search and replace characters
# <  : &lt;
# >  : &gt;
# \n : <br>

def makeSafe(unsafeText):
    safeText = str()

    # Sequencing and iteration (through string)
    for i in range(len(unsafeText)):
        char = unsafeText[i]

        # Selection
        if char == '<':
            safeText += '&lt;'
        elif char == '>':
            safeText += '&gt;'
        elif char == '\n':
            safeText += '<br>'
        else:
            safeText += char

    return safeText


# Instantiate list of messages
messages = list()

# Socket input event
# Message comes in from client (the parameter message)
# Parameter message is a dictionary containing username, message, and time

@socketio.event
def updateMessage(message):

    # Time is converted from UNIX timestamp epoch into human readable format
    message['time'] = datetime.utcfromtimestamp(int(message['time']) / 1000)
    message['time'] = message['time'].strftime('%m / %d<br>%Y<br>%H:%M:%S')

    # Calls makeSafe procedure to prevent users from rendering their own HTML
    message['username'] = makeSafe(message['username'])
    message['msg'] = makeSafe(message['msg'])

    # Appends message to list of messages
    messages.append(message)

    # List of messages is sent to all the clients to be updated
    emit('newMessage', messages, broadcast=True)


# On client connection send current messages list to client
# as they just joined and need to see the messages without waiting
# for someone to send

@socketio.event
def connected():

    # List of messages is sent to all the clients to be updated
    emit('newMessage', messages, broadcast=True)


# Using flask, serve index.html page to the client
@app.route('/')
def index():
    return render_template('index.html')


# ENTRY POINT
if __name__ == '__main__':
    # Run SocketIO App and Async subprocesses in debug mode
    socketio.run(app, debug=True)
```

static/index.js

```
/*
 * static/js/index.js
 *
 * 2021 CSP Create Project
 * A realtime internet chat room using WebSockets handled with Flask SocketIO
 * Built on a Flask Python server with a JavaScript client
 */

/*
 * Procedure scrollDown
 * Scrolls down the table of messages using a
 * fun little animation
 */

function scrollDown()
{
    $('table.msgtable').animate(
    {
        scrollTop: $('table.msgtable')[0].scrollHeight
    }, 500);
};


/*
* This procedure shows the list being used
*
* Procedure for displaying messages
* Takes paramter of message array from server
* Iterates through messages and renders them
* on the HTML page
*/

function displayMessages(servedMessages)
{
    // Get messages as array from server input
    var messages = Array.from(servedMessages);

    var table = '';
    // For each message in messages
    for (var i=0; i < messages.length; i++)
    {
        // Adds HTML for each message
        message = messages[i];
        table += `<tr><td class="username">${message.username}</td>`;
        table += `<td class="msg">${message.msg}</td>`;
        table += `<td class="time">${message.time}</td></tr>`;
    }

    table += '</table>';
    // Render table in document
    document.getElementById('messages').innerHTML = table;
}


var lastSent = 0; // The time since last sent message (UNIX timestamp)
const sendDelay = 3;    // Delay between sends in seconds

// On page load
$(document).ready(function()
{
    scrollDown();         // Scroll down messages on document load
    var socket = io();    // Inst Socket IO connection

    // On socket connection
    socket.on('connect',function()
    {
        // Calls for initial messages
        socket.emit('connected')

        /*
         * On form being submitted
         * Checks for valid username
         * Checks for valid message
         * Checks if user waited since last send
         * If all checks are correct, formats the
         * message dictionary and sends the socket
        */
        var form = $('form').on('submit',function(f)
        {
            f.preventDefault(); // Prevent default action
            var time = (new Date).getTime();    // Get the current time as UNIX timestamp
            if (time > lastSent + (sendDelay * 1000)) // Looks to see if its been longer than the delay since last send
            {
                var username = $('input.username').val();    // Sets username to username box
                var msg = document.getElementById('messageArea').value // Sets message to message box
                if (!msg.replace(/\s/g, '').length) // Check if there are all spaces etc.
                {
                    alert('Message does not contain any characters worth sending!');
                } else {
                    if (!username.replace(/\s/g, '').length)
                    {
                        alert('Must provide username!')
                    } else {
                        socket.emit('updateMessage',{
                            username: username,
                            msg: msg,
                            time: time
                        });

                        lastSent = time;
                        document.getElementById('msbox').value = '';
                        $('#msbox').val('');
                        $('input.message').val('').focus();
                    }
                }
            }
        });
    });

    /*
     * This action is performed when messages are
     * sent from the server. They are inputed as
     * the parameter serverdMessages
     */

    socket.on('newMessage', function(servedMessages)
    {
        // Display the messages on the HTML Page
        // Calls student made procedure
        displayMessages(servedMessages);

        // Scroll down to newest message
        // Calls student made procedure
        scrollDown();
    });
});
```

templates/index.html

```
<!DOCTYPE HTML>
<!--
Made for 2021 CSP Create Project
I am not responsible for the content
posted by users of the website
-->
<html>
    <head>
        <title>CSP ChatRoom</title>
                    <!--
                    External Scripts
                    JQuery: <https://jquery.com/>
                    SocketIO: <https://socket.io/> (client side JS, not Node)
                    -->
                    <link rel="stylesheet" href="{{ url_for('static', filename='index.css') }}"><!--index.css for page
stylesheet-->
                    <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.5.1/jquery.min.js"

integrity="sha512-bLT0Qm9VnAYZDflyKcBaQ2gg0hSYNQrJ8RilYldYQ1FxQYoCLtUjuuRuZo+fjqhx/qtq/1itJ0C2ejDxlt
ZVFg==" crossorigin="anonymous">
        </script><!--JQuery-->
                    <script src="https://cdnjs.cloudflare.com/ajax/libs/socket.io/3.0.4/socket.io.js"

integrity="sha512-aMGMvNYu8Ue4G+fHa359jcPb1u+ytAF+P2SCb+PxrjCdO3n3ZTxJ30zuH39rimUggmTwmh2u7wv
QsDTHESnmfQ==" crossorigin="anonymous">
        </script><!--SocketIO-->
                    <script src="{{ url_for('static', filename='index.js') }}"></script><!--index.js-->
```

```html
    </head>
    <!-- Body -->
    <body class=''>
        <div class='centered' style='width: 1000px;'> <!-- Send Box -->
            <form action='' method='POST' id='sendForm'> <!-- Form -->
            <table>
                <tr>
                    <td>Username: </td>
                    <td><input type='text' class='username' placeholder='Username' maxlength='50' id='userbox'>
                    </td>
                    <td><h1>Send Message!</h1></td>
                </tr>
                <tr>
                    <td>Message: </td>
                    <td><textarea maxlength="100"
                        name='message' class='message'rows='4' cols='50' name='message'
                        id='messageArea' form='sendForm' placeholder='Message...' id='msbox'></textarea>
                    </td>
                <td><input type='submit' value='Send'></td>
                </tr>
            </table>
            </form>
        </div>
        <!-- Message Box -->
        <div class='centered' style='width: 1000px; height: 580px;'>
            <table class='msgtable'  id='messages'> <!-- Table for messages to load -->
            </table>
                                                <p id="newmsg" style="margin-left:20px; margin-bottom:5px;"></p>
        </div>
    </body>
</html>
```

static/index.css

```css
/*
 * static/css/index.css
 *
 * Styles for templates/index.html
 * 2021 CSP Create Project
 * A realtime internet chat room using WebSockets handled with Flask SocketIO
 * Built on a Flask Python server with a JavaScript client
 */

body {
  background-color: #0a0a0c;
}
div {
  margin: 20px;
  padding: 0px;
  border: 2px #657786 solid;
  background-color: black;
}
```

```css
table {
  font-family:arial;
  border-collapse: collapse;
  margin: 20px 20px 20px 20px;
  padding: 0px;
}
.msgtable {
  width: 960px;
  overflow-y:scroll;
  height:85%;
  display:block;
}
td {
  text-align: left;
  padding: 8px;
  color: white;
}
.username {
  border-bottom:  2px solid #657786;
  overflow-wrap: anywhere;
  width: 20%;
}
.msg {
  border-bottom:  2px solid #657786;
  overflow-wrap: anywhere;
  width: 80%;
}
.time {
  border-bottom:  2px solid #657786;
  width: 30%;
}
h1 {
  font-family: arial;
  text-align:left;
  margin: 10px;
  padding: 0px;
  font-size: 220%;
}
p {
  color:white;
  font-size:120%;
  font-family: arial;
  text-align:left;
  margin: 10px;
  padding: 0px;
}
a {
  color:#1DA1F2;
  font-family:arial;
}
a:hover {
  color:#0d659c;
  font-size:100%;
  font-family:arial;
  text-align:left;
}
textarea {
  font-family:arial;
  margin: 0px;
  padding: 5px;
  border: 2px white solid;
  border-radius: 10px;
  resize: none;
}
input[type=text] {
  font-family:arial;
  margin: 0px;
  padding: 5px;
  border: 2px white solid;
  width: 365px;
  border-radius: 30px;
}
input[type=submit] {
  font-family:arial;
  font-size:130%;
  margin: 0px;
  padding: 10px;
  border-radius: 30px;
  background-color: #1DA1F2;
  color: white;
  border: 2px #1DA1F2 solid;
}
input[type=submit]:hover {
  background-color: #0d659c;
  border: 2px #0d659c solid;
  font-family: arial ;
}
```