**OPTION 1: WEBPAGE PROJECT**  
  
**Project Description**

Pie Socket provides a simple WebSocket server that generates periodic data in the form of “Hello Worlds”, user connect messages and user chat messages. You task will be to write a program that monitors the data, processes it, and displays the result to the user on a web page in real time.

Pie Socket’s web socket sample page is available here: https://www.piesocket.com/websocket-tester.

Note: Please ensure the “channel” is set to channel 1. In their example the URL may be:

wss://demo.piesocket.com/v3/channel\_123?api\_key=VCXCEuvhGcBDP7Xhi-JJUDvR1e1D3eiVjgZ9VRiaV&notify\_self

Please change to:

wss://demo.piesocket.com/v3/channel\_1?api\_key=VCXCEuvhGcBDP7Xhi-JJUDvR1e1D3eiVjgZ9VRiaV&notify\_self

**Summary of Implementation**

* I created a simple React webpage which communicates with Pie Socket’s WebSocket server which is found at “wss://demo.piesocket.com/v3/channel\_1?api\_key=VCXCEuvhGcBDP7Xhi-JJUDvR1e1D3eiVjgZ9VRiaV&notify\_self”
* Standard node websocket library is used as the core library of this project
* “mathjs” library is used to calculate the “average” and “standard deviation” between messages
* I used “material-ui” library since it is very powerful library and easy to use for the JavaScript GUI applications
* “recharts” library is used to visualize the data. Run chart has been created and it displays 50 data samples at once.
* If the websocket connection is disconnected for some reason web app will try to reconnect after 5 seconds

**OBJECTIVES**

The following are specific things your program should accomplish.

1. Measuring “Hello World”’s Performance

I calculated followings statistics and display them Realtime on web page

1. Time from last messages (td)

I used following simple equation to calculate the time from last message.

Text

Description automatically generated

1. Total number of messages

Simple counter is used to calculate the total number of messages on each message event

1. Average time between messages

“mathjs” library is used to calculate the average time between messages. The latest 10 samples are taking to calculate the average calculation.

1. Standard deviation between messages

“mathjs” library is used to calculate the standard deviation between messages. The latest 10 samples are taking to calculate the standard deviation.

1. Tracking users
2. Messages

*Note: As of 2022-12-07 only “Hello Worlds” were being sent. So, the tracking users and messages are captured into the “Unknown Data” Tab of the web app.*

**How to test auto-reconnect functionality**

You can just disconnect the internet connection in your device. Then the running chart will be frozen and the statistics won’t be updating which means the webscoket connection has been disconnected.   
Now you can turn on the internet connection to your device and the chart will start running again.

**Further Implementation and Enhancement**

1. At the moment React code has the single App.js file which has the all the backend and front-end functions. It’s better to restructure the code and it will enhance the CI/CD process for the future releases.
2. Create the user interface more stylish

Graphical user interface, application

Description automatically generated