Implement the Websocket Backend with the specific actions:

Installation:

* Setting up the project
* Create a new folder of the project in vs code
* Install the necessary packages of implememt the websocket backend
* Pip install websockets asyncio

Connecting to the websocket server in vs code:

* Import the necessary modules
* Create a new file name ‘websocket\_ser.py’
* Using the terminal run the server ‘python websocket\_ser.py’

Connecting to the websocket server to client in vs code:

* Create a new file in websocket common folder ‘test\_client.py’
* Using the split terminal option to run the server to client ‘python test\_client.py’

Available Actions and Response in json:

Echo the message in sstreaming format:

* send the each character of the received message back to the client one by one with a 0.1 – second delay
* Request send a text message
* Response streams each of message with 0.1 sec delay

{

  "type": "echo",

  "content": " The quick brown fox jumped over the lazy dog o"

}

Response:

Echo: T

Echo: h

Echo: e

Echo:

Echo: q

Echo: u

Echo: i

Echo: c

Echo: k

Echo:

Echo: b

Echo: r

Echo: o

Echo: w

Echo: n

Echo:

Echo: f

Echo: o

Echo: x

Echo:

Echo: j

Echo: u

Echo: m

Echo: p

Echo: e

Echo: d

Echo:

Echo: o

Echo: v

Echo: e

Echo: r

Echo:

Echo: t

Echo: h

Echo: e

Echo:

Echo: l

Echo: a

Echo: z

Echo: y

Echo:

Echo: d

Echo: o

Echo: g

Echo:

Echo: o

Echo:

Echo message in reverse in streaming format:

* send the each character of the received message in reverse order back to the client one by one with a 0.1 – second delay
* Request send a text message
* Response streams each of message with 0.1 sec delay

{

  "type": "echo",

  "content": " The quick brown fox jumped over the lazy dog o"

}

Response:

Reverse: o

Reverse:

Reverse: g

Reverse: o

Reverse: d

Reverse:

Reverse: y

Reverse: z

Reverse: a

Reverse: l

Reverse:

Reverse: e

Reverse: h

Reverse: t

Reverse:

Reverse: r

Reverse: e

Reverse: v

Reverse: o

Reverse:

Reverse: d

Reverse: e

Reverse: p

Reverse: m

Reverse: u

Reverse: j

Reverse:

Reverse: x

Reverse: o

Reverse: f

Reverse:

Reverse: n

Reverse: w

Reverse: o

Reverse: r

Reverse: b

Reverse:

Reverse: k

Reverse: c

Reverse: i

Reverse: u

Reverse: q

Reverse:

Reverse: e

Reverse: h

Reverse: T

Reverse:

Count the last character Repetitions:

* Counts the number of times the last character apper in the message (excluding the last character)

Request in json format:

{

  "type": "count",

  "content": " The quick brown fox jumped over the lazy dog o"

}

Response:

Testing Count:

Count: 4