

Course Assessment Test

Course Title	Real time data streaming and monitoring with WebSocket and Socket.IO	Date	Nov 2025
Name		Dept	

- 1) This assessment test is to be given out before course commencement. Answers are to be filled in column entitled "Pre-Course Answer"
- 2) At the end of the course, the same assessment sheet is to be given out where answers are to be filled in column entitled "Post-Course Answer". Instructor will then share the answers and participants need to total the score in both "Pre" and "Post" columns through self-marking.
- 3) Assessment sheets will be collected for filling.

No	Question	Pre-Course Answer	Post-Course Answer
1	Which of the following are key features / characteristics of the WebSocket protocol? <ul style="list-style-type: none"> i. Primarily used for real-time data streaming between a server and a client ii. Protocol that supports full-duplex communication iii. The communication is persistent over a single TCP connection until explicitly closed iv. Requires a request to be sent from server and response received from client before any packet of data can be sent over the persistent connection <ul style="list-style-type: none"> a) Features i, ii and iii b) Features i, ii and iv c) Features i, iii and iv d) Features ii, iii and iv 		A
2	What are the two common HTTP headers in the request and response required to setup a WebSocket connection between client and server? <ul style="list-style-type: none"> a) connection: websocket, switch: immediate b) connection: upgrade, upgrade: websocket c) connection: websocket, mode: upgrade d) websocket: true, mode: switch 		B
3	What is the fallback protocol that Socket.IO uses in the event that a WebSocket connection cannot be established between the client and server? <ul style="list-style-type: none"> a) Redis / Kafka b) MQMT c) HTTP long polling d) HTTP queued messaging 		C
4	What are the key features of Socket.IO that make it attractive for use by developers to create real time streaming applications? <ul style="list-style-type: none"> i. Event-Based API - both client and server can emit and listen for events ii. Automatic reconnection - uses a heartbeat mechanism, which periodically checks 		D

Course Assessment Test

	<p>the status of the connection, and connects when it is broken</p> <p>iii. Broadcasting - from the server-side, send an event to all clients or to a subset of clients</p> <p>iv. Room and Namespaces - allow logical separation of communication channels that involve subset of connected users</p> <p>a) Features i, ii and iii b) Features i, ii and iv c) Features ii, iii and iv d) All the features</p>		
5	<p>What does this statement do on the server side:</p> <pre>socket.on('message', (msg) => { })</pre> <p>a) It listens for a message event emitted from the client b) It waits until a message queue event is triggered before sending the message to the client c) It listens for a connection attempt from the client to send the first message d) It listens for a specific message given by the <code>msg</code> parameter that it needs to intercept in an event handling sequence</p>		A
6	<p>What is the statement that a client needs to use to connect to a specific namespace <code>CoolSpace</code> on the server-side?</p> <p>a) <code>socket = namespace(CoolSpace);</code> b) <code>socket = connect(CoolSpace);</code> c) <code>socket = io(CoolSpace);</code> d) <code>socket = connect(namespace, CoolSpace);</code></p>		C
7	<p>What is the statement that the server uses to listen for an attempt from the client to connect to the namespace <code>CoolSpace</code>?</p> <p>a) <code>const nsp = listen.connection(CoolSpace);</code> b) <code>const nsp = io.of(CoolSpace);</code> c) <code>const nsp = namespace.connect(CoolSpace);</code> d) <code>const nsp = listen.nameSpace(CoolSpace);</code></p>		B
8	<p>What is the statement that the server uses to broadcast a message <code>msgContent</code> to a subset of connected users within a specific room <code>coolRoom</code>?</p> <p>a) <code>socket.emit('message', coolRoom, { msgContent });</code> b) <code>socket.emit(coolRoom, 'message', { msgContent });</code> c) <code>socket.emit('message', { msgContent }).to(coolRoom);</code> d) <code>socket.to(coolRoom).emit('message', { msgContent });</code></p>		D
9	<p>Imagine a socket <code>nsp</code> associated with a namespace that contains a subset of connected users in a particular application. How do we obtain this number of users?</p>		

Course Assessment Test

	<ul style="list-style-type: none">a) <code>nsp.sockets.users</code>b) <code>nsp.sockets.size</code>c) <code>nsp.connected.users</code>d) <code>nsp.namespace.size</code>		B
10	<p>What is the statement to be executed on the server-side to check on the number of users connected to a particular room <code>roomA</code> in a given namespace <code>myname</code>, where the socket <code>nsp</code> is associated with that namespace?</p> <ul style="list-style-type: none">a) <code>nsp.namespace(myname) .rooms (roomA)</code>b) <code>nsp.get(myname) .getRooms (roomA)</code>c) <code>nsp.adapter.rooms.get(roomA).size</code>d) <code>nsp.connection(myname) . get(roomA).size</code>		C
Total			