Practical - 8

Prerequisite: **Perform the programs given in the Lab. manual provided during the regular classroom sessions.**

1. Cerate appropriate Express.js/Socket.io code for the following web-application requirement:
   a. Create a server-side program that listens for incoming Socket.io connections.
   b. Using Socket.io, create a client-side program connects to the server.
   c. Upon receiving a connection request, the server sends a welcome message to client.
   d. Upon receiving a message from server, the client displays that message on command line.

2. Cerate appropriate Express.js/Socket.io code for the following web-application requirement:
   a. Create a server-side program that listens for incoming Socket.io connections.
   b. Using Socket.io, create a client-side HTML page that connects to the server.
   c. Upon receiving a connection request, the server sends a welcome message to client.
   d. Upon receiving a message from server, the client-side HTML page displays that message in browser.
   e. Upon receiving a message from server, the client-side HTML page displays that message in some HTML textarea element.

3. Cerate appropriate Express.js/Socket.io code for the following web-application requirement:
   a. Create a server-side program that listens for incoming Socket.io connections.
   b. Using Socket.io, create a client-side program that connects to the server.
   c. Upon receiving a connection request, the server sends a sample welcome.html page to client.
   d. Upon receiving a welcome.html file from server, the client should render the welcome.html in web browser.

4. Cerate appropriate Express.js/Socket.io code for the following web-application requirement:
   a. Create a server-side program that listens for incoming Socket.io connections.
   b. Using Socket.io, create a client-side HTML page that connects to the server.
   c. After connecting to the server, the client sends an integer (taken from user through an HTML input number element) to server.
   d. Upon receiving an integer from the client, the server computes factorial of it and sends the result back to the client.
   e. Upon receiving the computation factorial result from server, the client displays it in the browser.

5. Cerate appropriate Express.js/Socket.io code for the following web-application requirement:
   a. Create a server-side program that listens for incoming Socket.io connections.
   b. Using Socket.io, create a client-side HTML page that connects to the server.

c. After connecting to the server, the client sends a string (taken from user through an HTML input text element) to server.
d. Upon receiving a text from the client, the server checks whether it is palindrome or not. The server sends the decision back to the client.
e. Upon receiving the decision from server, the client displays it in the browser.
    i. Turn browser's background to green if the decision is 'Yes' i.e. string is palindrome, else turn the browser's background blue.
6. Create a basic upvote application as demonstrated during our classroom sessions.

Case study: Develop a basic peer-to-peer chat application functional in web browser.