

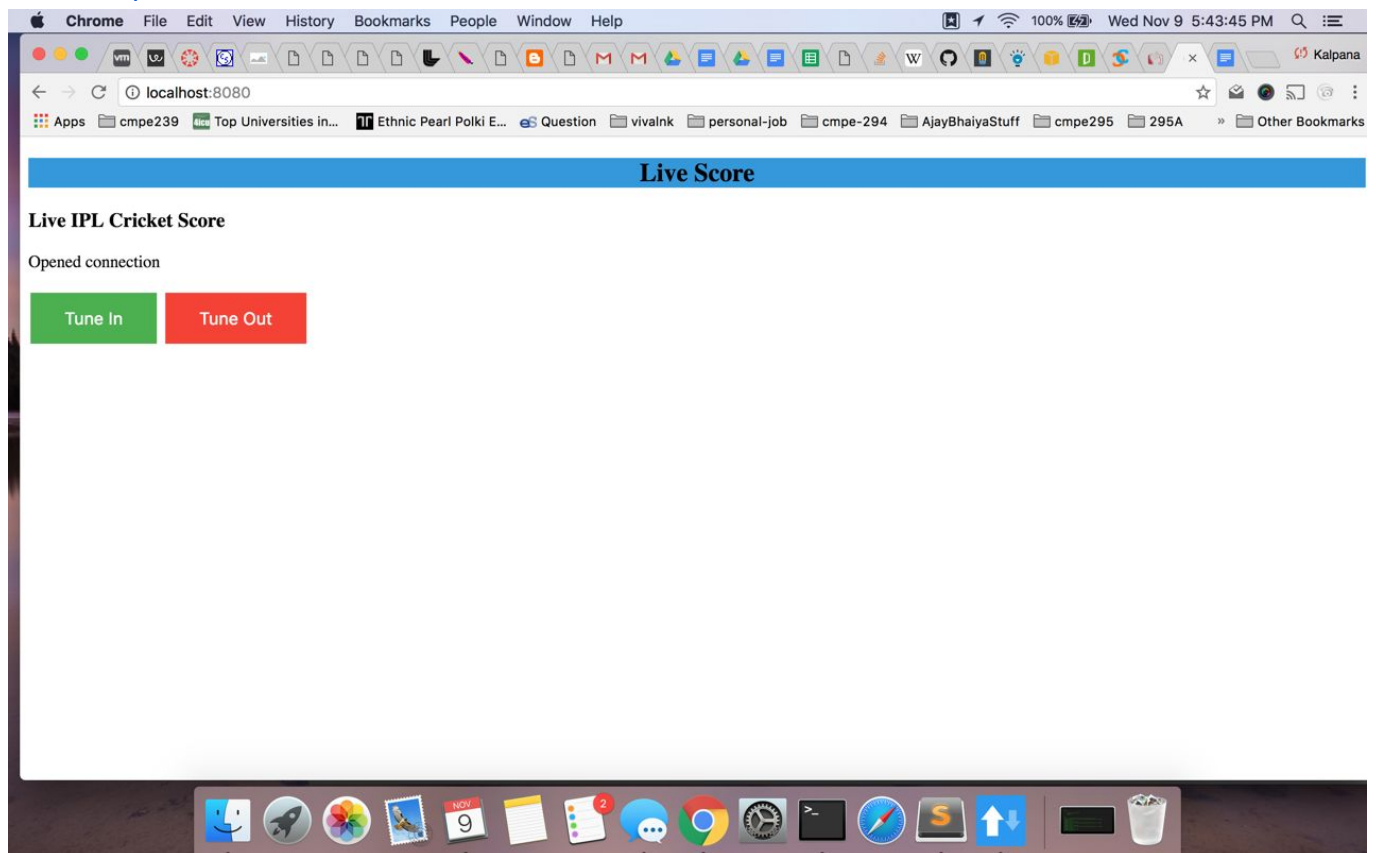
- Tested on google chrome
- Required nodejs libraries
 - Http
 - Express
 - Sse
 - Websocket
- Start server.js nodejs application using following command
 - node server.js

```

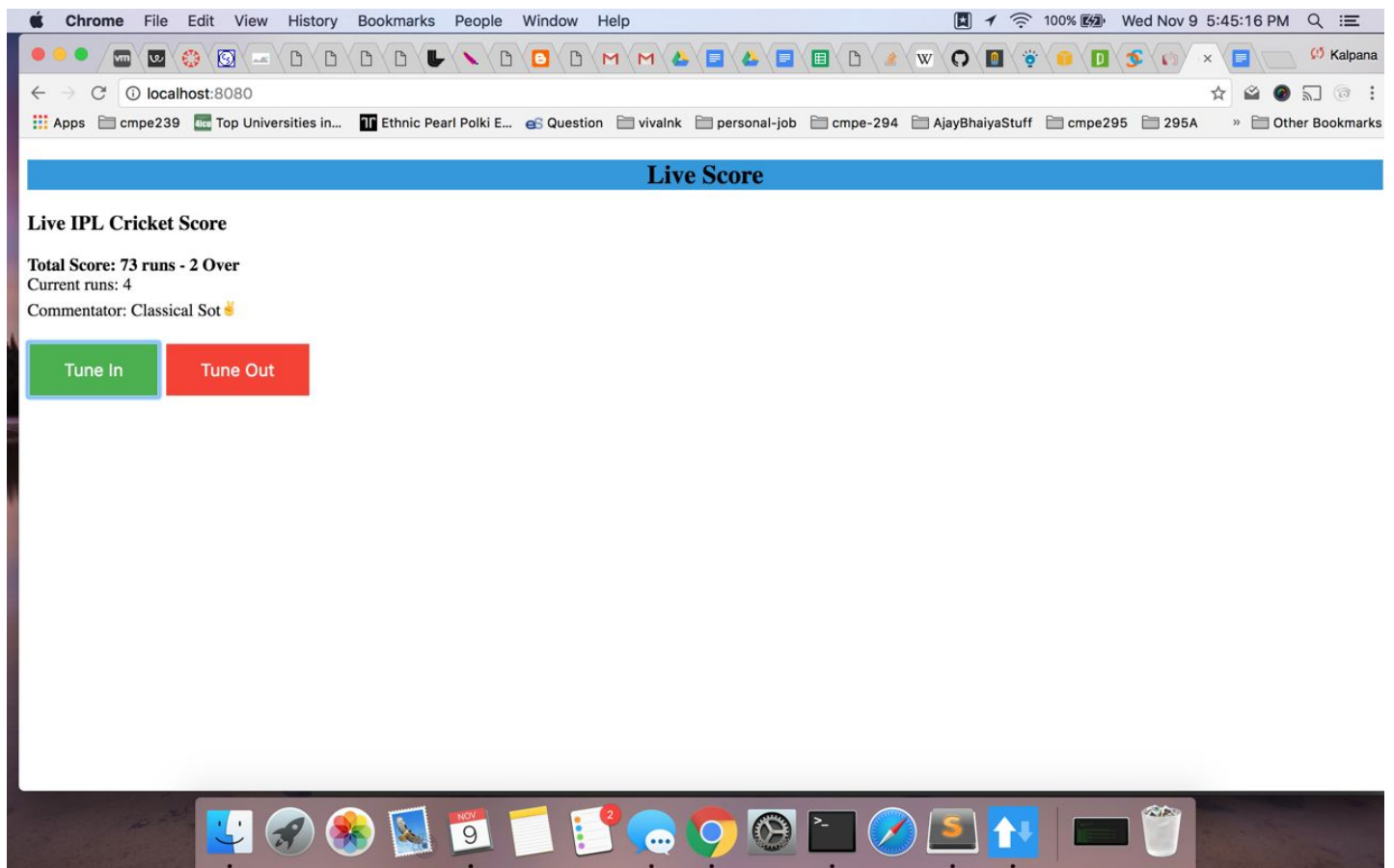
[kt:~/Documents/sjsu/websocket/server-sent-events-demo-master>
[kt:~/Documents/sjsu/websocket/server-sent-events-demo-master>
[kt:~/Documents/sjsu/websocket/server-sent-events-demo-master>
[kt:~/Documents/sjsu/websocket/server-sent-events-demo-master>
[kt:~/Documents/sjsu/websocket/server-sent-events-demo-master> node server.js
Opened connection
Sent: {"Status":"Connected"}
Sent: {"runs":2,"score":2,"over":1}
Sent: {"runs":1,"score":3,"over":1}
Sent: {"runs":0,"score":3,"over":1}
Sent: {"runs":0,"score":3,"over":1}
Sent: {"runs":6,"score":9,"over":1}
Sent: {"runs":2,"score":11,"over":1}
Sent: {"runs":0,"score":11,"over":1}
Sent: {"runs":2,"score":13,"over":1}
Sent: {"runs":0,"score":13,"over":1}
Sent: {"runs":0,"score":13,"over":1}
Sent: {"runs":6,"score":19,"over":1}

```

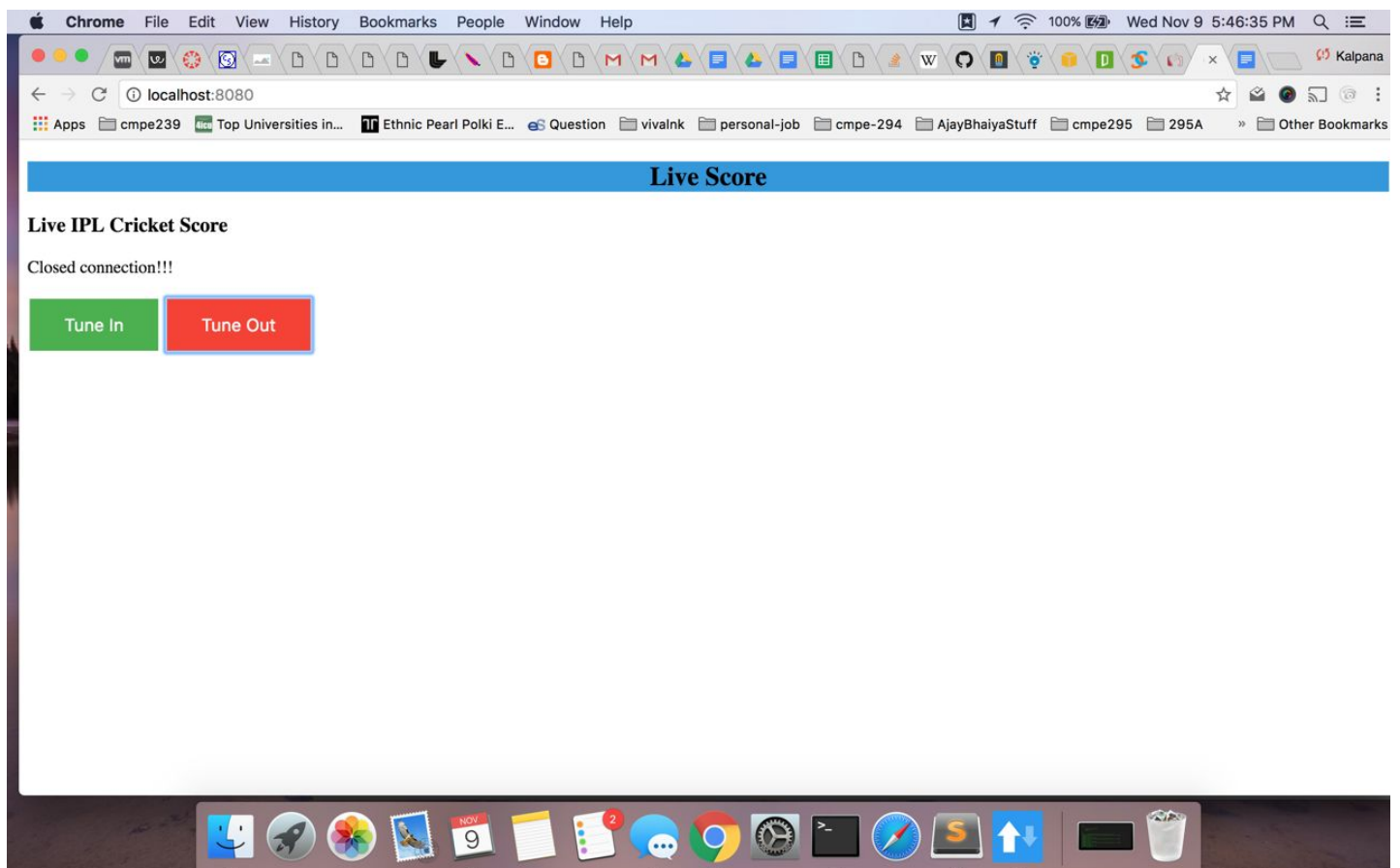
- Open client.html in chrome browser on 8080 port
 - <http://localhost:8080/>



- Server generates random numbers to client for scores initially on tuning to server.



- On tune out



- Tune in again

