## Name attributes of the http protocol that makes it difficult to use for real time systems

The http protocol works with request/response sets, as a request is sent to the server it will respond with data via a response. When the transaction is finished the connection is terminated and afterwards the client and the server don’t know anything about each other. This makes it difficult for a client and server to communicate in real time. So in order to get a real time system to work with HTTP you can use polling, long-polling, HTTP streaming or web sockets.

## What is WebSocketprotocol, how is it different from HTTP communication, what advantages has it over HTTP

The web socket protocol is an independent TCP-based protocol. The http protocol relies on polling(new connection every time a client request data) to establish connections between a client and a server. This puts a harder load on the server compared to web sockets, who only have to establish the initial connection.

## What is the advantage of using libraries like Socket.IO, Sock.JS, WS, over pure WebScoket libraries in the backend and standard APIs on frontend. Which problems do they solve

They simplify the usage of web sockets and more importantly provides failovers to other protocols in the event that web sockets are not supported on the browser or the server.

## What is Backend as a Service, Database as a Service, why would consider using Firebase in your projects

Backend as a service is where the backend is provided so the only code that needs to be written is client code.

Database as a service is where you can have a cloud database already setup – so you don’t need to create a server yourself to deploy your database to. Both these options are provided with Firebase so one of the main reasons to use Firebase is that it makes it a lot faster to create applications, as all you need to worry about is your client. It supports Oauth, communicates over SSL, and is therefore a “pretty” secure connection.

## Explain and demonstrate a “real time” system using either your: your own server and a library like Socket.IO or Firebase

Sockets\_Chat\_Server is a chat server created with the Socket.io library.

Firebase\_Chat\_Server is a chat client that uses Firebase to create a near identical chat system as Sockets\_Chat\_Server.

## Explain and demonstrate the difference between the simple chat system in your own WebSocket + Node.js backend vs. Firebase

The main difference between the two projects is that Firebase\_Chat\_Server doesn’t have a server. This is instead handled by Firebase where all messages and users that uses the system is also stored.

Sockets\_Chat\_Server doesn’t have a database connected. This means that if the server goes down previous messages will not be stored. Messages will be stored on the Firebase example.