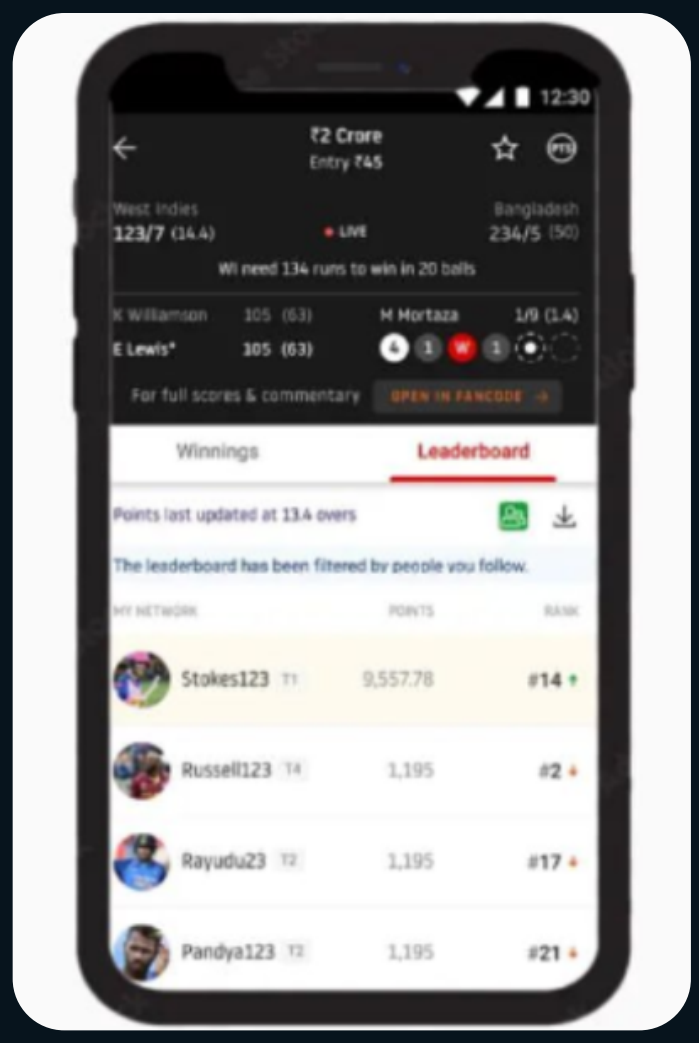
What is WEB SOCKET

Ankit Pangasa

What is Web Socket?

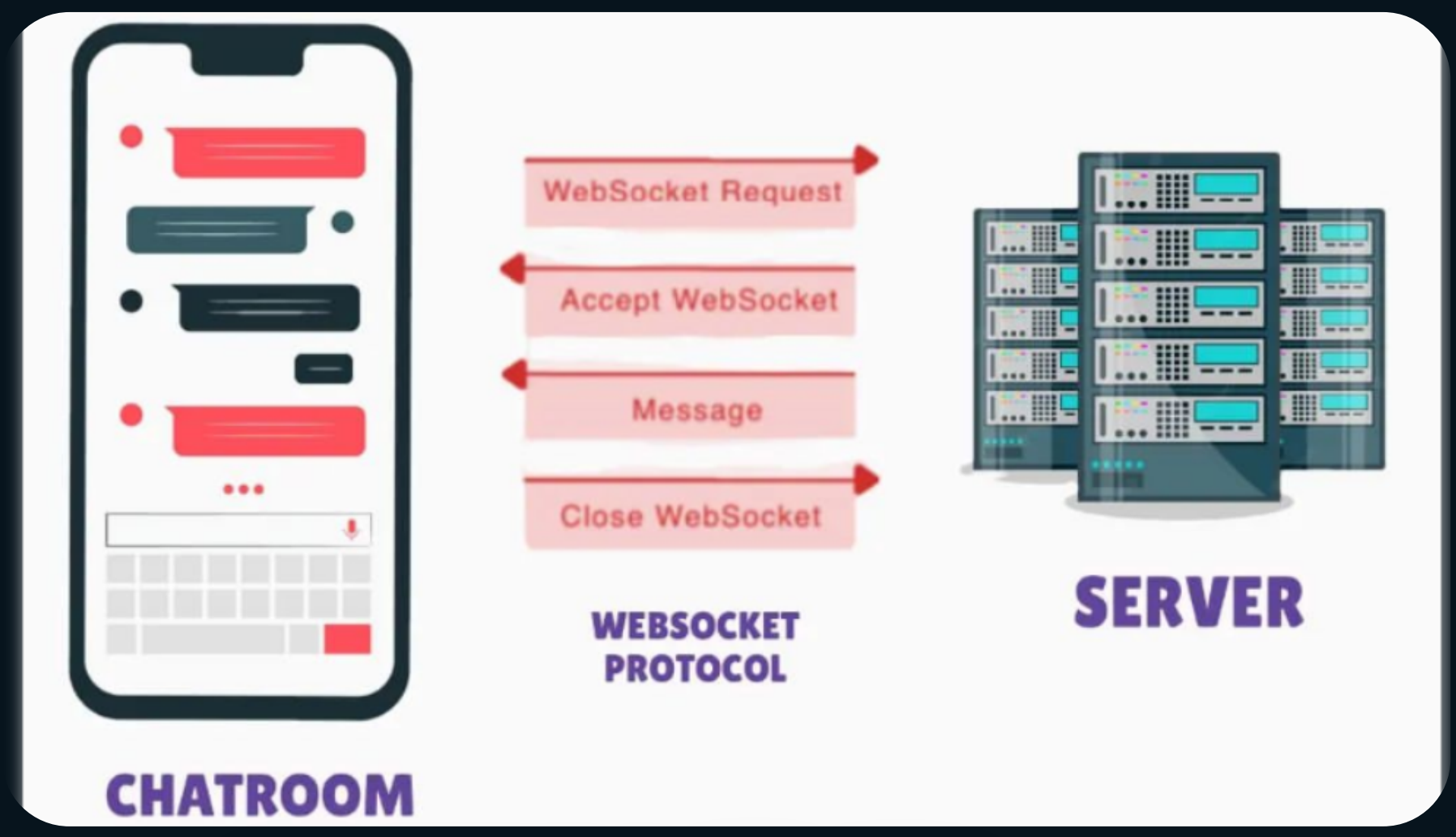
A WebSocket is a computer communication protocol that enables two-way communication between a client and a server over a single, long-lived connection.



SWIPE →

Ankit Pangasa

Now imagine if you are building a chat app using an HTTP so each time you had to send a request to the server for a new messages
But in the chat app, you receive a message instantly when someone sends you a message you don't have to request it eachtime
This is done by WebSocket

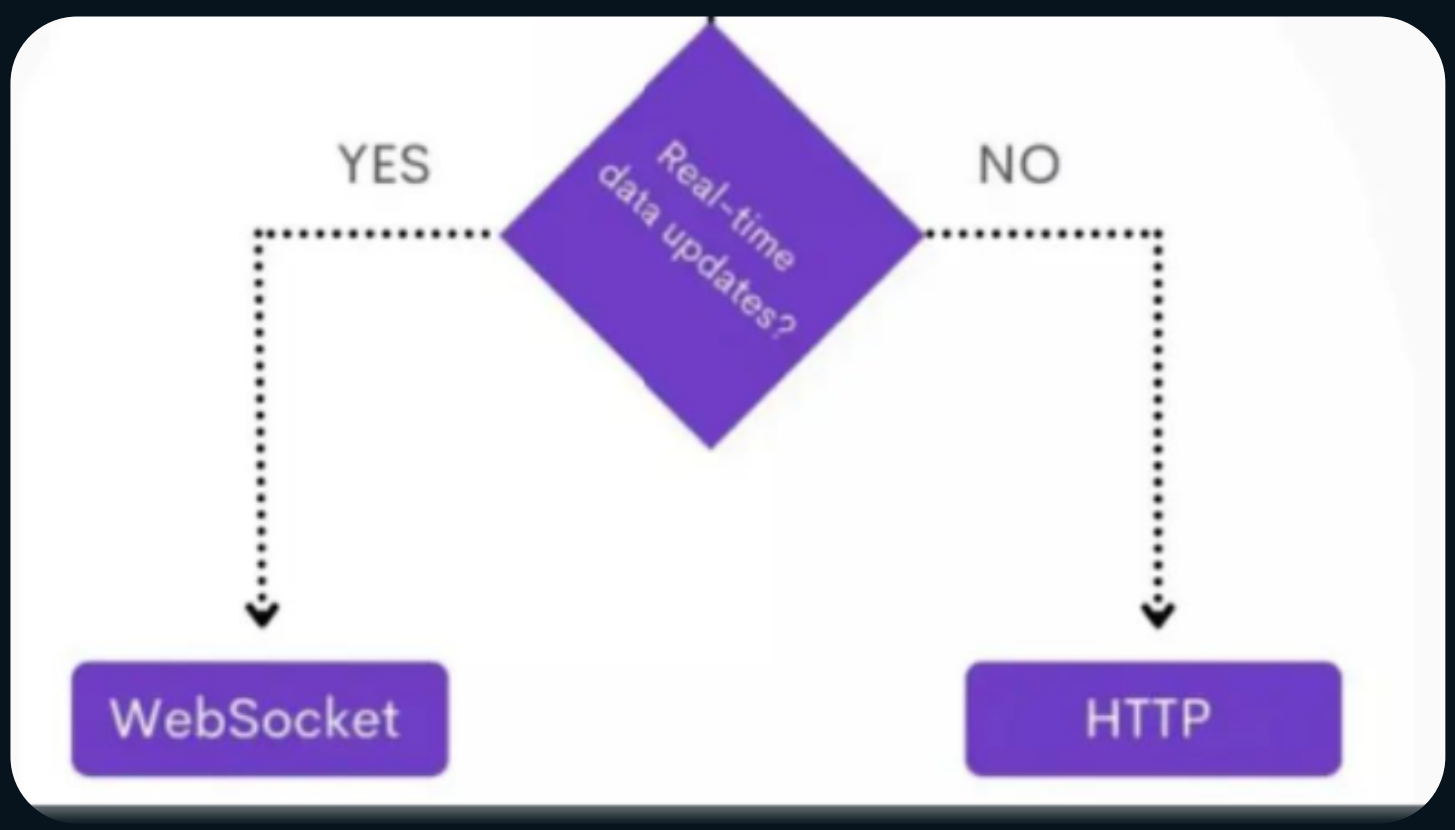


SWIPE →

Ankit Pangasa

WHEN TO USE WEBSOCKETS

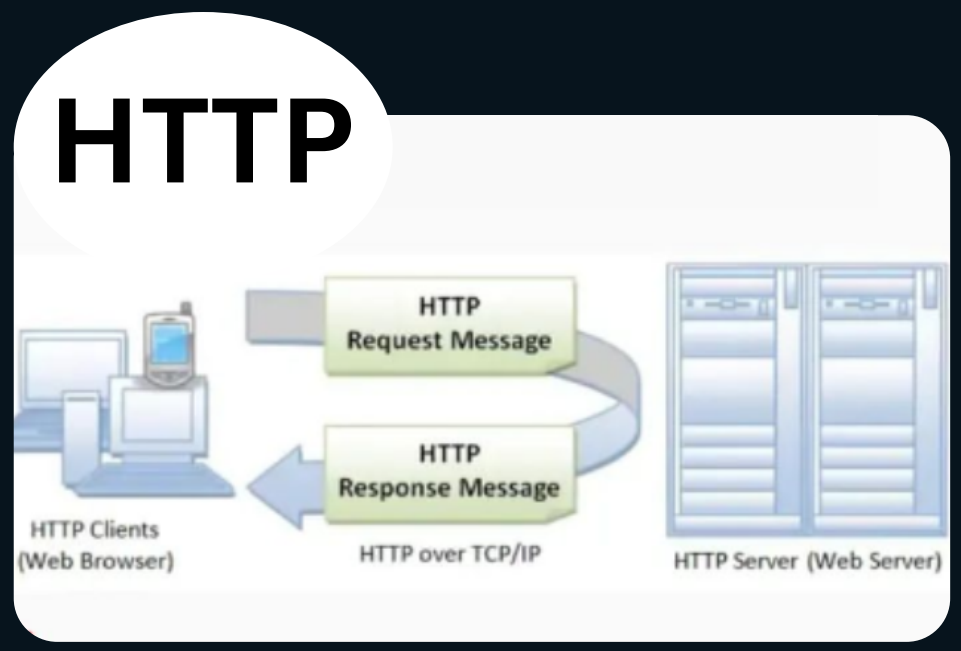
WebSockets are best suited for applications that require real-time, bidirectional communication between the client and server.

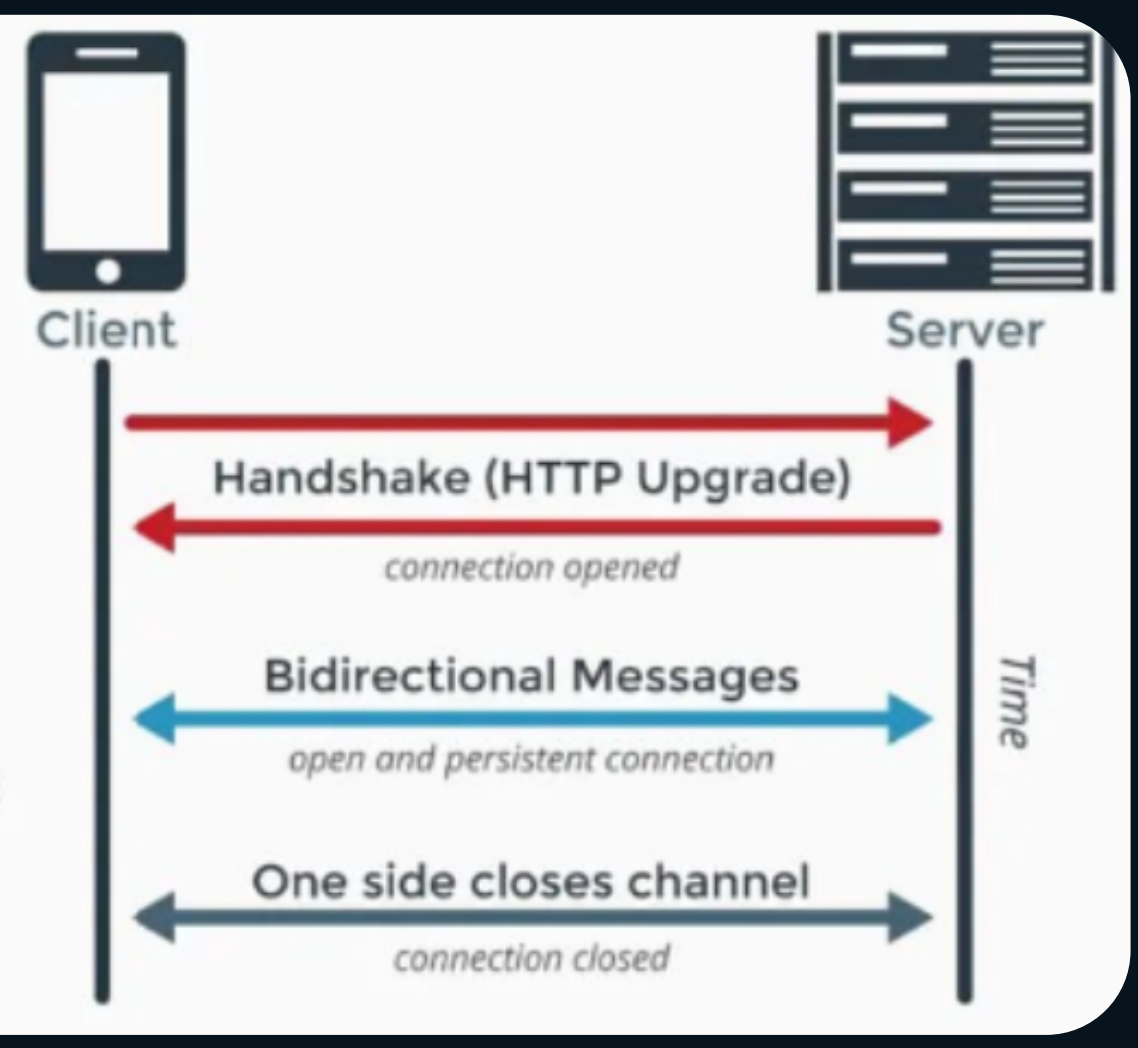


SWIPE →

Ankit Pangasa

HTTP (HyperText Transfer Protocol) is a request-response protocol used for transferring data over the web. It is primarily used to request and retrieve HTML pages, documents, images, and other resources from a server.





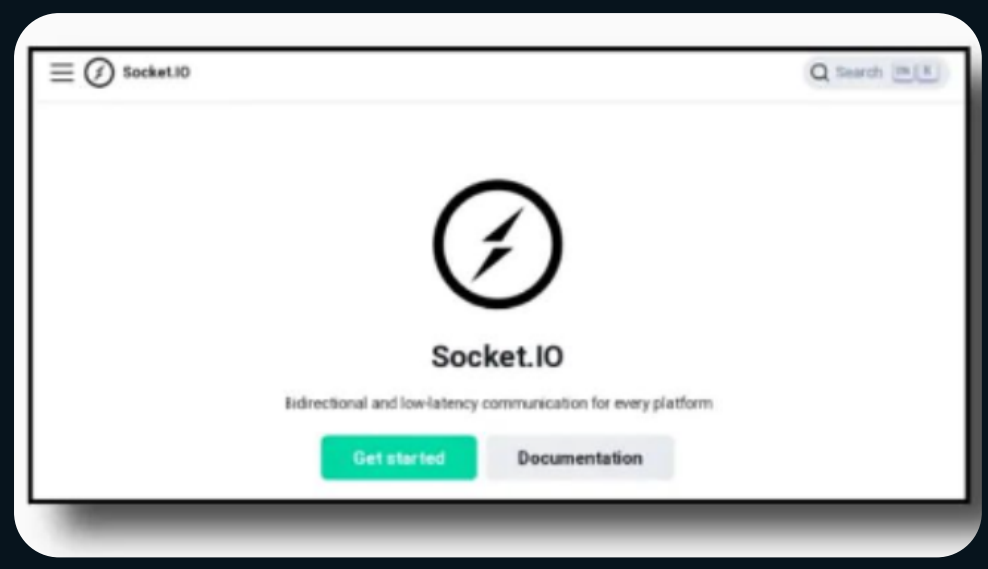
WebSocket, on the other hand, is a bidirectional communication protocol that allows real-time communication between the client and server.

SWIPE →

Ankit Pangasa

How are Websockets used in Projects?

There are many libraries that abstract Websocket in order to make them friendlier to developers. Possibly the most popular one is socket.io.



Node.je is used for building server-side event-driven apps using javascript

Express.js is a framework based on node.js

SWIPE →

Ankit Pangasa

Server Side

In a WebSocket application, the server side typically involves implementing a WebSocket server that can accept incoming connections from clients and handle the bidirectional communication between the client and server.

Server sending a message through the Websocket

```
// Receive socket connections
router.ws('/:userId/status', (ws, req) => {
  // The socket connected code would go here, not on an event handler

  // Message received event, handler receives raw message
  ws.on('message', (message) => {
    try {
      const status = JSON.parse(message);

      if (typeof status === 'boolean') {
        event.emit(
          'userStatusChange',
          { userId: req.params.userId, status },
        );
      }
    } catch (error) {
      // console.log(error);
    }
  });

  // Socket closed event, client disconnected
  ws.on('close', () => {
    event.emit(
      'userStatusChange',
      { userId: req.params.userId, status: false },
    );
  });
});
```

SWIPE →

Ankit Pangasa

Client Side

In a WebSocket application, the client side typically involves implementing a WebSocket client that can initiate the WebSocket connection with the server and handle the bidirectional communication between the client and server.

Client receiving a message through the Websocket

```
// Create socket with WebSockets API
const socket = new WebSocket(`${process.env.ROOT_SOCKET}/socket`);

// Message received handler, function receives raw message
socket.onmessage = (rawMessage) => {
  const message = JSON.parse(rawMessage.data);
  if (message.type === 'available call') {
    // Handle the event received, in this case using Redux
    dispatch(setAvailableRoom({ message.roomName, message.phoneNumber }));
  }
};
```

SWIPE →

Ankit Pangasa

IF YOU LIKE MYCONTENT

like comment save share

Ankit Pangasa