HTML CODE

<!DOCTYPE html>

<html>

<head>

<title>Frontend</title>

</head>

<body>

<div id="data-display"></div>

<script>

const socket = new WebSocket("ws://localhost:8080");

socket.onmessage = (event) => {

const data = event.data;

displayData(data);

};

function displayData(data) {

const dataDisplay = document.getElementById("data-display");

dataDisplay.innerHTML += "<p>" + data + "</p>";

}

</script>

</body>

</html>

JAVA BACKEND CODE

import javax.crypto.Cipher;

import javax.crypto.KeyGenerator;

import javax.crypto.SecretKey;

import java.io.\*;

import java.net.ServerSocket;

import java.net.Socket;

import java.util.Base64;

public class BackendServer {

public static void main(String[] args) throws Exception {

// Generate encryption key

KeyGenerator keyGen = KeyGenerator.getInstance("AES");

keyGen.init(128);

SecretKey secretKey = keyGen.generateKey();

ServerSocket serverSocket = new ServerSocket(8080);

while (true) {

System.out.println("Waiting for a connection...");

Socket socket = serverSocket.accept();

InputStream in = socket.getInputStream();

OutputStream out = socket.getOutputStream();

// Wrap input and output streams with encryption

Cipher encryptCipher = Cipher.getInstance("AES");

encryptCipher.init(Cipher.ENCRYPT\_MODE, secretKey);

Cipher decryptCipher = Cipher.getInstance("AES");

decryptCipher.init(Cipher.DECRYPT\_MODE, secretKey);

// Handle incoming data stream

DataInputStream dis = new DataInputStream(in);

String encryptedData = dis.readUTF();

byte[] encryptedBytes = Base64.getDecoder().decode(encryptedData);

byte[] decryptedBytes = decryptCipher.doFinal(encryptedBytes);

String decryptedData = new String(decryptedBytes);

// Save data to time series database (not shown)

saveToTimeSeriesDB(decryptedData);

// Emit the saved data to the frontend (not shown)

emitToFrontend(decryptedData);

socket.close();

}

}

private static void saveToTimeSeriesDB(String data) {

// Implement saving data to your time series database

}

private static void emitToFrontend(String data) {

// Implement emitting data to the frontend

}

}