LAB REPORT 01

CYBERSECURITY

RITIK TIWARI (B21CS098)

Link for the code files: https://github.com/testgithubtiwari/CyberSecurity-Labs/tree/main/lab01

Q1) Create a client server simple handshaking program using socket programming.

I have created two separate files for this question client. Dart and server. Dart where I have used the programming language is Dart because currently, I am working on some flutter project that is why I am very much familiar with this language and it is also in my practice.

Screenshot of the codes for both Client.dart and Server.dart file and the Output of the screenshot is shown below

Client.dart

```dart
import 'dart:io';

void startClient() async {
  Socket? socket;
  String host = '127.0.0.1';
  int port = 12346;

  try {
    socket = await Socket.connect(host, port);
    print('Connected to $host:$port');
  } catch (e) {
    print('Error: $e');
  } finally {
    socket?.close();
  }
}

Run | Debug
void main() {
  startClient();
}
```

Server.dart

```dart
import 'dart:io';

void startServer() async {
  ServerSocket? serverSocket;
  String host = '127.0.0.1';
  int port = 12346;

  try {
    serverSocket = await ServerSocket.bind(host, port);
    print('Server listening on $host:$port');
    await for (Socket clientSocket in serverSocket) {
      print(
          'Connection from ${clientSocket.remoteAddress}:${clientSocket.remotePort}');
      clientSocket.close();
    }
  } catch (e) {
    print('Error: $e');
  } finally {        You, 14 hours ago • lab01 added
    serverSocket?.close();
  }
}

Run | Debug
void main() {
  startServer();
}
```

Result

```
PS D:\cybersecurity\lab01\q1> dart .\server.dart
Server listening on 127.0.0.1:12346
Connection from InternetAddress('127.0.0.1', IPv4):57079
```

```
PS D:\cybersecurity\lab01\q1> dart .\client.dart
Connected to 127.0.0.1:12346
```

Q2) Create a socket program to connect to a predefined server (try https://iitj.ac.in/ use others too).

I have used my portfolio sever which is rt-portfolio.vercel.app . Screenshots of the code file and the Results are shown below.

Result

```
Connection established to rt-portfolio.vercel.app
```

Client.dart

```dart
import 'dart:io';

Future<Socket> connectToServer(String host, int port,
    {bool useSSL = false}) async {
  Socket clientSocket = await Socket.connect(host, port);

  if (useSSL) {
    SecurityContext context = SecurityContext.defaultContext;
    clientSocket = await SecureSocket.connect(host, port, context: context);
  }

  return clientSocket;
}

void sendRequest(Socket clientSocket, String request) {
  clientSocket.write(request);
}

void receiveResponse(Socket clientSocket, String serverHost) {
  print('Connection established to ${serverHost}');
  // clientSocket.listen(
  //   (List<int> data) {
  //     print(
  //       'Connection established to ${clientSocket.remoteAddress.host}:${clientSocket.remotePort}',
  //     );
  //     // String response = String.fromCharCodes(data);
  //     // print(response);
  //   },
  //   onDone: () {
  //     // Close the connection after receiving the response
  //     clientSocket.close();
  //   },
  //   onError: (error) {
  //     print('Error: $error');
  //   },
  // );
}

void closeConnection(Socket clientSocket) {
  // Close the connection
  clientSocket.close();
}

Run | Debug
void main() async {
  String serverHost = "rt-portfolio.vercel.app";
  int serverPort = 443; // HTTPS default port
  String httpRequest =        You, 14 hours ago • lab01 added
      "GET / HTTP/1.1\r\nHost: rt-portfolio.vercel.app\r\n\r\n";
  Socket clientSocket =
      await connectToServer(serverHost, serverPort, useSSL: true);
  sendRequest(clientSocket, httpRequest);
  receiveResponse(clientSocket, serverHost);
}
```

Q3) The server will simply echo whatever it receives back to the client.

Result

```
PS D:\cybersecurity\lab01\q3> dart .\server.dart
Server listening on 127.0.0.1:12346
Connection from InternetAddress('127.0.0.1', IPv4):57537
PS D:\cybersecurity\lab01\q3>
```

```
PS D:\cybersecurity\lab01\q3> dart .\client.dart
Connected to 127.0.0.1:12346
Enter the message from client to server: Please send what I am sent!
Server response: Please send what I am sent!
Connection closed
PS D:\cybersecurity\lab01\q3>
```

Client.dart

```dart
import 'dart:io';

void startClient() async {
  Socket clientSocket = await Socket.connect('127.0.0.1', 12346);
  print('Connected to 127.0.0.1:12346');

  stdout.write('Enter the message from client to server: ');
  String clientMessage = stdin.readLineSync()!;

  clientSocket.write(clientMessage);

  clientSocket.listen(
    (List<int> event) {
      String responseMessage = String.fromCharCodes(event);
      print('Server response: $responseMessage');
    },
    onDone: () {
      print('Connection closed');
      clientSocket.destroy();
    },
    onError: (error) {
      print('Error: $error');
      clientSocket.destroy();
    },
  );
}

void main() {
  startClient();
}
```

Server.dart

```dart
import 'dart:io';
import 'dart:convert';

void startServer() async {
  ServerSocket serverSocket = await ServerSocket.bind('127.0.0.1', 12346);
  print('Server listening on 127.0.0.1:12346');

  await for (Socket clientSocket in serverSocket) {
    print(
        'Connection from ${clientSocket.remoteAddress}:${clientSocket.remotePort}');

    List<int> data = await clientSocket.first;
    String clientMessage = utf8.decode(data);

    clientSocket.write(clientMessage);

    // Close the connection
    clientSocket.close();
    serverSocket.close();
  }
}

Run | Debug
void main() {
  startServer();
}
```

Q4) The client sends a text message in Lowercase to the server and the server returns it in Uppercase to the client.

Result

```
PS D:\cybersecurity\lab01> cd .\q4\
PS D:\cybersecurity\lab01\q4> dart .\server.dart
Server listening on 127.0.0.1:12346
Connection from InternetAddress('127.0.0.1', IPv4):57635
PS D:\cybersecurity\lab01\q4>
```

```
PS D:\cybersecurity\lab01> cd .\q4\
PS D:\cybersecurity\lab01\q4> dart .\client.dart
Connected to 127.0.0.1:12346
Enter the message from client to server: I am lower case.
Server response: I AM LOWER CASE.
Connection closed
PS D:\cybersecurity\lab01\q4>
```

Client.dart

```dart
import 'dart:io';
// import 'dart:convert';

void startClient() async {
  Socket clientSocket = await Socket.connect('127.0.0.1', 12346);
  print('Connected to 127.0.0.1:12346');

  stdout.write('Enter the message from client to server: ');
  String clientMessage = stdin.readLineSync()!;

  clientSocket.write(clientMessage);

  clientSocket.listen(
    (List<int> event) {
      String responseMessage = String.fromCharCodes(event);
      print('Server response: $responseMessage');
    },
    onDone: () {
      print('Connection closed');
      clientSocket.destroy();
    },
    onError: (error) {
      print('Error: $error');
      clientSocket.destroy();
    },
  );
}

Run | Debug
void main() {
  startClient();
```

Server.dart

```dart
import 'dart:io';
import 'dart:convert';

void startServer() async {
  ServerSocket serverSocket = await ServerSocket.bind('127.0.0.1', 12346);
  print('Server listening on 127.0.0.1:12346');

  await for (Socket clientSocket in serverSocket) {
    print(
        'Connection from ${clientSocket.remoteAddress}:${clientSocket.remotePort}');
    List<int> data = await clientSocket.first;
    String clientMessage = utf8.decode(data);
    String messageUppercase = clientMessage.toUpperCase();
    clientSocket.write(messageUppercase);

    clientSocket.close();
    serverSocket.close();
  }
}

Run | Debug
void main() {
  startServer();
}
```

Q5) The client sends a text message to the server and the server returns it in reverse order to the client.

Result

```
PS D:\cybersecurity\lab01> cd .\q5\
PS D:\cybersecurity\lab01\q5> dart .\server.dart
Server listening on 127.0.0.1:12346
Connection from InternetAddress('127.0.0.1', IPv4):55566
PS D:\cybersecurity\lab01\q5>
```

```
PS D:\cybersecurity\lab01> cd .\q5\
PS D:\cybersecurity\lab01\q5> dart .\client.dart
Connected to 127.0.0.1:12346
Enter the message from client to server: please reverse me!
Server response: !em esrever esaelp
Connection closed
PS D:\cybersecurity\lab01\q5>
```

Client.dart

```dart
import 'dart:io';
// import 'dart:convert';

void startClient() async {
  Socket clientSocket = await Socket.connect('127.0.0.1', 12346);
  print('Connected to 127.0.0.1:12346');

  stdout.write('Enter the message from client to server: ');
  String clientMessage = stdin.readLineSync()!;

  clientSocket.write(clientMessage);

  clientSocket.listen(
    (List<int> event) {
      String responseMessage = String.fromCharCodes(event);
      print('Server response: $responseMessage');
    },
    onDone: () {
      print('Connection closed');
      clientSocket.destroy();
    },
    onError: (error) {
      print('Error: $error');
      clientSocket.destroy();
    },
  );
}

Run | Debug
void main() {
  startClient();
}
```

Server.dart

```dart
import 'dart:io';
import 'dart:convert';

void startServer() async {
  ServerSocket serverSocket = await ServerSocket.bind('127.0.0.1', 12346);
  print('Server listening on 127.0.0.1:12346');

  await for (Socket clientSocket in serverSocket) {
    print(
        'Connection from ${clientSocket.remoteAddress}:${clientSocket.remotePort
    List<int> data = await clientSocket.first;
    String clientMessage = utf8.decode(data);
    String messageUppercase = reverseString(clientMessage);
    clientSocket.write(messageUppercase);

    clientSocket.close();
    serverSocket.close();
  }
}

String reverseString(String clientMessage) {        You, 31 minutes ago • chnage
  List<String> characters = clientMessage.split('');
  List<String> reversedCharacters = characters.reversed.toList();
  return reversedCharacters.join('');
}

Run | Debug
void main() {
  startServer();
```