REPORT

**C/S프로그래밍**

**제목 : 14주차 과제**

**학과 : 컴퓨터소프트웨어과**

**학번 : 2017100301**

**이름 : 이상남**

**담당교수 : 김인태 교수님**

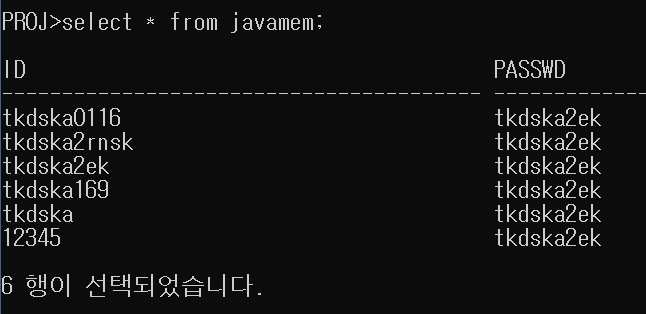
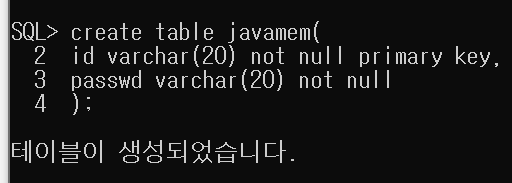
**제출일 : 2020.12.16**

1. **소스코드**

|  |
| --- |
| **server.java 서버페이지** |
| **package Server;**  **import java.awt.Color;**  **import java.io.BufferedReader;**  **import java.io.InputStream;**  **import java.io.InputStreamReader;**  **import java.io.OutputStream;**  **import java.io.PrintWriter;**  **import java.net.ServerSocket;**  **import java.net.Socket;**  **import java.util.ArrayList;**  **import java.util.concurrent.ExecutionException;**  **import javax.swing.JFrame;**  **import javax.swing.JScrollPane;**  **import javax.swing.JTextArea;**  **public class Server extends JFrame {**  **private JTextArea textArea;**  **private JScrollPane pane;**  **private ServerSocket soket;**  **private Socket s;**  **private ArrayList list;**  **public Server() {**  **super("채팅서버 v1.0.1");**  **setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);**  **textArea = new JTextArea();**  **pane = new JScrollPane(textArea);**  **textArea.setBackground(Color.LIGHT\_GRAY);**  **add(pane);**  **textArea.setText("채팅서버 시작됨!.\n");**  **setSize(400, 200);**  **setVisible(true);**  **try {**  **list = new ArrayList();**  **soket = new ServerSocket(5000);**  **while (true) {**  **s = soket.accept();**  **ChatThread t = new ChatThread();**  **list.add(t);**  **t.start();**  **}**  **} catch (Exception e) {**  **e.printStackTrace();**  **}**  **}**  **class ChatThread extends Thread {**  **BufferedReader br;**  **PrintWriter pw;**  **InputStream is;**  **String nickName;**  **public ChatThread() {**  **try {**  **is = s.getInputStream();**  **br = new BufferedReader(new InputStreamReader(is));**  **OutputStream os = s.getOutputStream();**  **pw = new PrintWriter(os, true);**  **} catch (Exception e) {**  **e.printStackTrace();**  **}**  **}**  **public void send(String str) {**  **pw.println(str);**  **}**  **public void run() {**  **try {**  **pw.println("로그인을 해주세요");**  **nickName = br.readLine();**  **broadcast(nickName + "님이 입장했습니다.");**  **while (true) {**  **String str = br.readLine();**  **broadcast("[" + nickName + "]" + str);**  **}**  **} catch (Exception e) {**  **e.printStackTrace();**  **}**  **}**  **private void broadcast(String str) {**  **// TODO Auto-generated method stub**  **for (int i = 0; i < list.size(); i++) {**  **ChatThread t = (ChatThread) list.get(i);**  **t.send(str);**  **}**  **}**  **}**  **public static void main(String[] args) {**  **// TODO Auto-generated method stub**  **new Server();**  **}**  **}** |
| **Client.java 클라이언트 페이지** |
| **package Server;**  **import java.awt.BorderLayout;**  **import java.awt.event.ActionEvent;**  **import java.awt.event.ActionListener;**  **import java.io.BufferedReader;**  **import java.io.InputStream;**  **import java.io.InputStreamReader;**  **import java.io.OutputStream;**  **import java.io.PrintWriter;**  **import java.net.Socket;**  **import javax.swing.JButton;**  **import javax.swing.JFrame;**  **import javax.swing.JLabel;**  **import javax.swing.JPanel;**  **import javax.swing.JScrollPane;**  **import javax.swing.JTextArea;**  **import javax.swing.JTextField;**  **public class Client extends JFrame implements ActionListener, Runnable {**  **private JTextArea textArea;**  **private JScrollPane pane;**  **private JTextField textField;**  **private JPanel panel = new JPanel();**  **private JLabel label1 = new JLabel("아이디");**  **private JLabel label2 = new JLabel("비밀번호");**  **private JTextField tid = new JTextField(10);**  **private JTextField tpw = new JTextField(10);**  **private JButton btn = new JButton("로그인");**  **private JButton btn2 = new JButton("회원가입");**  **Socket socket = null;**  **BufferedReader br;**  **PrintWriter pw;**  **OutputStream os;**  **int result;**  **public Client() {**  **// TODO Auto-generated constructor stub**  **setTitle("채팅 클라이언트 v.1.0.1");**  **setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);**  **panel.setSize(400, 100);**  **panel.add(label1);**  **panel.add(tid);**  **panel.add(label2);**  **panel.add(tpw);**  **panel.add(btn);**  **panel.add(btn2);**  **add(panel, BorderLayout.NORTH);**  **textArea = new JTextArea();**  **textArea.setEditable(false);**  **pane = new JScrollPane(textArea);**  **add(pane);**  **textField = new JTextField();**  **add(textField, BorderLayout.SOUTH);**  **tid.addActionListener(this);**  **tpw.addActionListener(this);**  **btn.addActionListener(this);**  **btn2.addActionListener(this);**  **textField.addActionListener(this);**  **setSize(500, 500);**  **setVisible(true);**  **try {**  **socket = new Socket("127.0.0.1", 5000);**  **InputStream is = socket.getInputStream();**  **InputStreamReader isr = new InputStreamReader(is);**  **br = new BufferedReader(isr);**  **os = socket.getOutputStream();**  **pw = new PrintWriter(os, true);**  **} catch (Exception e) {**  **e.printStackTrace();**  **}**  **Thread t = new Thread(this);**  **t.start();**  **}**  **@Override**  **public void run() {**  **// TODO Auto-generated method stub**  **try {**  **String str = br.readLine();**  **textArea.append(str + "\n");**  **while (true) {**  **str = br.readLine();**  **textArea.append(str + "\n");**  **textArea.setCaretPosition(textArea.getDocument().getLength());**  **}**  **} catch (Exception e) {**  **// TODO: handle exception**  **e.printStackTrace();**  **}**  **}**  **@Override**  **public void actionPerformed(ActionEvent e) {**  **// TODO Auto-generated method stub**  **if (e.getSource() == btn) {**  **String id = tid.getText();**  **String passwd = tpw.getText();**  **DBconn d = new DBconn();**  **if (d.Login(id, passwd) == 1) {**  **pw.write(id);**  **result = 1;**  **String str = textField.getText();**  **pw.println(str);**  **textField.setText("");**  **}**  **}**  **if (e.getSource() == btn2) {**  **new Signup();**  **}**  **if ((e.getSource() == textField) && (result == 1)) {**  **String str = textField.getText();**  **pw.println(str);**  **textField.setText("");**  **}**  **}**  **public static void main(String[] args) {**  **// TODO Auto-generated method stub**  **new Client();**  **}**  **}** |
| **DBconn.java 디비 연결페이지** |
| **package Server;**  **import java.sql.Connection;**  **import java.sql.DriverManager;**  **import java.sql.SQLException;**  **import java.sql.PreparedStatement;**  **import java.sql.ResultSet;**  **public class DBconn {**  **public static Connection *dbConn*;**  **public String pw;**  **public String id;**  **public String pid;**  **public String getPw() {**  **return pw;**  **}**  **public void setPw(String pw) {**  **this.pw = pw;**  **}**  **public String getId() {**  **return id;**  **}**  **public void setId(String id) {**  **this.id = id;**  **}**  **public static Connection getConnection() {**  **Connection conn = null;**  **try {**  **String user = "proj";**  **String pw = "proj";**  **String url = "jdbc:oracle:thin:@localhost:1521:testdb";**  **Class.*forName*("oracle.jdbc.driver.OracleDriver");**  **conn = DriverManager.*getConnection*(url, user, pw);**  **System.*out*.println("Database에 연결되었습니다.\n");**  **} catch (ClassNotFoundException cnfe) {**  **System.*out*.println("DB 드라이버 로딩 실패 :" + cnfe.toString());**  **} catch (SQLException sqle) {**  **System.*out*.println("DB 접속실패 : " + sqle.toString());**  **} catch (Exception e) {**  **System.*out*.println("Unkonwn error");**  **e.printStackTrace();**  **}**  **return conn;**  **}**  **public int Login(String id, String pw) {**  **Connection conn = null; // DB연결된 상태(세션)을 담은 객체**  **PreparedStatement pstm = null; // SQL 문을 나타내는 객체**  **ResultSet rs = null; // 쿼리문을 날린것에 대한 반환값을 담을 객체**  **int result = 0;**  **try {**  **// SQL 문장을 만들고 만약 문장이 질의어(SELECT문)라면**  **// 그 결과를 담을 ResulSet 객체를 준비한 후 실행시킨다.**  **String quary = "SELECT id,passwd FROM javamem where id='" + id + "'";**  **conn = DBconn.*getConnection*();**  **pstm = conn.prepareStatement(quary);**  **rs = pstm.executeQuery();**  **while (rs.next()) {**  **String memid = rs.getString(1);**  **String mempw = rs.getString(2);**  **if (mempw.equals(pw)) {**  **result = 1;**  **pid = id;**  **} else if (!mempw.equals(pw)) {**  **result = 2;**  **} else**  **result = 0;**  **/\* String result = memid+" "+mempw; System.out.println(result); \*/**  **}**  **} catch (SQLException sqle) {**  **System.*out*.println("SELECT문에서 예외 발생");**  **sqle.printStackTrace();**  **} finally {**  **// DB 연결을 종료한다.**  **try {**  **if (rs != null) {**  **rs.close();**  **}**  **if (pstm != null) {**  **pstm.close();**  **}**  **if (conn != null) {**  **conn.close();**  **}**  **} catch (Exception e) {**  **throw new RuntimeException(e.getMessage());**  **}**  **}**  **return result;**  **}**  **}** |
| **Signup.java 회원가입 폼 페이지** |
| **package Server;**  **import java.awt.Component;**  **import java.awt.event.ActionEvent;**  **import java.awt.event.ActionListener;**  **import java.sql.Connection;**  **import java.sql.PreparedStatement;**  **import java.sql.ResultSet;**  **import java.sql.SQLException;**  **import javax.swing.JButton;**  **import javax.swing.JFrame;**  **import javax.swing.JLabel;**  **import javax.swing.JOptionPane;**  **import javax.swing.JPanel;**  **import javax.swing.JScrollPane;**  **import javax.swing.JTextArea;**  **import javax.swing.JTextField;**  **public class Signup extends JFrame implements ActionListener {**  **private int sign;**  **private JPanel pan = new JPanel();**  **private JLabel lab1 = new JLabel("아이디");**  **private JLabel lab2 = new JLabel("비밀번호");**  **private JTextField tid = new JTextField(10);**  **private JTextField tpw = new JTextField(10);**  **private JButton btn1 = new JButton("중복체크");**  **private JButton btn2 = new JButton("가입하기");**  **private Component frame;**  **public Signup() {**  **// TODO Auto-generated constructor stub**  **setTitle("회원가입");**  **setDefaultCloseOperation(JFrame.*DISPOSE\_ON\_CLOSE*);**  **pan.add(lab1);**  **pan.add(tid);**  **btn1.addActionListener(this);**  **pan.add(btn1);**  **pan.add(lab2);**  **pan.add(tpw);**  **btn2.addActionListener(this);**  **pan.add(btn2);**  **add(pan);**  **setSize(300, 150);**  **setVisible(true);**  **}**  ***@Override***  **public void actionPerformed(ActionEvent e) {**  **// TODO Auto-generated method stub**  **if (e.getSource() == btn1) {**  **// 중복체크 하기 해서 0이면 사용불가 1이면 사용가능**  **String id = tid.getText();**  **if (id.length() <= 0) {**  **JOptionPane.*showMessageDialog*(frame, "아이디를 입력해주세요.");**  **} else {**  **SignCheck s = new SignCheck();**  **sign = s.sign\_check(id);**  **}**  **} else if (e.getSource() == btn2) {**  **if (sign == 0) {**  **JOptionPane.*showMessageDialog*(frame, "중복체크를 해주세요.");**  **} else {**  **String pw = tpw.getText();**  **String id = tid.getText();**  **if (id.length() <= 0 || pw.length() <= 0) {**  **JOptionPane.*showMessageDialog*(frame, "아이디와 비밀번호를 입력해주세요.");**  **} else {**  **SignCheck s = new SignCheck();**  **s.SignUp(id, pw);**  **}**  **}**  **}**  **}**  **}** |
| **SignCheck.java 회원체크 or 회원가입 페이지** |
| **package Server;**  **import java.awt.Component;**  **import java.sql.Connection;**  **import java.sql.PreparedStatement;**  **import java.sql.ResultSet;**  **import java.sql.SQLException;**  **import javax.swing.JOptionPane;**  **public class SignCheck {**  **private Component frame;**  **public int sign\_check(String id) {**  **Connection conn = null; // DB연결된 상태(세션)을 담은 객체**  **PreparedStatement pstm = null; // SQL 문을 나타내는 객체**  **ResultSet rs = null; // 쿼리문을 날린것에 대한 반환값을 담을 객체**  **int result = 0;**  **try {**  **// SQL 문장을 만들고 만약 문장이 질의어(SELECT문)라면**  **// 그 결과를 담을 ResulSet 객체를 준비한 후 실행시킨다.**  **String quary = "SELECT id FROM javamem where id='" + id + "'";**  **conn = DBconn.getConnection();**  **pstm = conn.prepareStatement(quary);**  **rs = pstm.executeQuery();**  **if (rs.next()) {**  **result = 0;**  **JOptionPane.showMessageDialog(frame, "중복된 아이디 입니다.");**  **} else {**  **result = 1;**  **JOptionPane.showMessageDialog(frame, "사용가능 아이디입니다.");**  **}**  **} catch (SQLException sqle) {**  **System.out.println("SELECT문에서 예외 발생");**  **sqle.printStackTrace();**  **} finally {**  **// DB 연결을 종료한다.**  **try {**  **if (rs != null) {**  **rs.close();**  **}**  **if (pstm != null) {**  **pstm.close();**  **}**  **if (conn != null) {**  **conn.close();**  **}**  **} catch (Exception e) {**  **throw new RuntimeException(e.getMessage());**  **}**  **}**  **return result;**  **}**  **public void SignUp(String id, String pw) {**  **Connection conn = null; // DB연결된 상태(세션)을 담은 객체**  **PreparedStatement pstm = null; // SQL 문을 나타내는 객체**  **int rs; // 쿼리문을 날린것에 대한 반환값을 담을 객체**  **try {**  **// SQL 문장을 만들고 만약 문장이 질의어(SELECT문)라면**  **// 그 결과를 담을 ResulSet 객체를 준비한 후 실행시킨다.**  **String quary = "insert into javamem values(?,?)";**  **conn = DBconn.getConnection();**  **pstm = conn.prepareStatement(quary);**  **pstm.setString(1, id);**  **pstm.setString(2, pw);**  **rs = pstm.executeUpdate();**  **if (rs > 0) {**  **JOptionPane.showMessageDialog(frame, "회원가입 성공");**  **} else**  **JOptionPane.showMessageDialog(frame, "회원가입 실패");**  **} catch (SQLException sqle) {**  **System.out.println("SELECT문에서 예외 발생");**  **sqle.printStackTrace();**  **} finally {**  **// DB 연결을 종료한다.**  **try {**  **if (pstm != null) {**  **pstm.close();**  **}**  **if (conn != null) {**  **conn.close();**  **}**  **} catch (Exception e) {**  **throw new RuntimeException(e.getMessage());**  **}**  **}**  **}**  **}** |

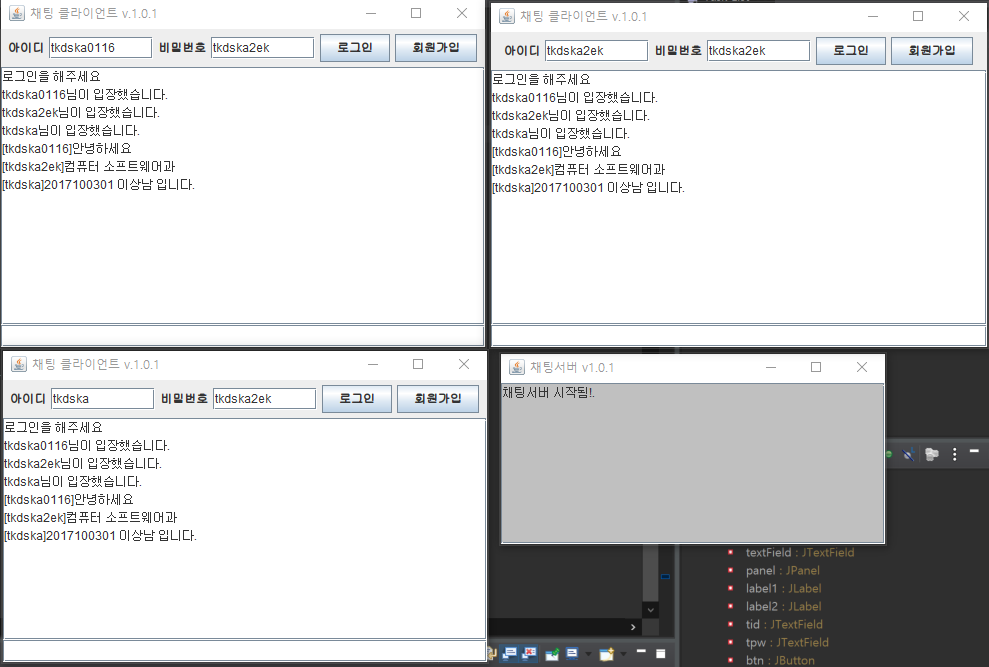
**코드는 보기 힘드실 것 같아 따로 파일로 첨부하겠습니다.**

1. **실행 화면**
2. **DB 설계 (오라클만있어서 오라클을 사용하였습니다.)**

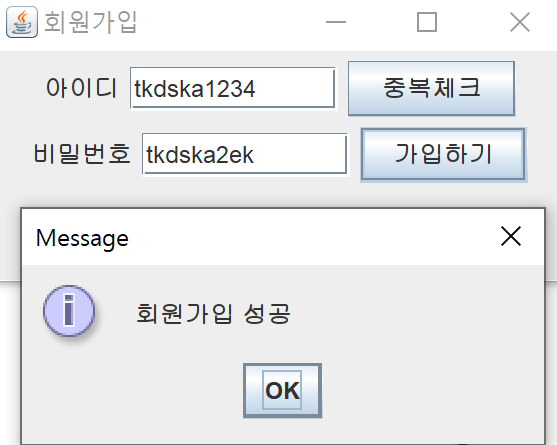
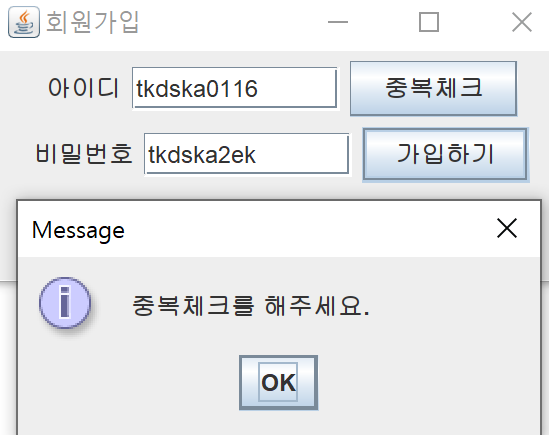
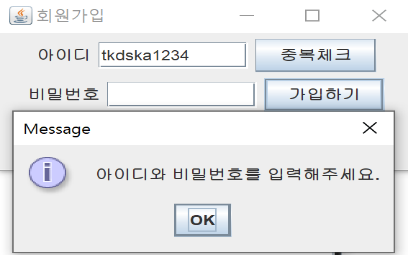
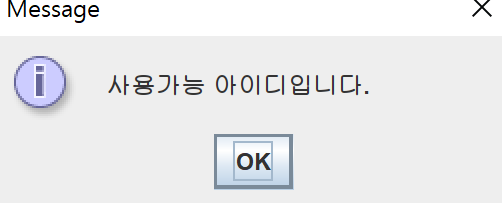
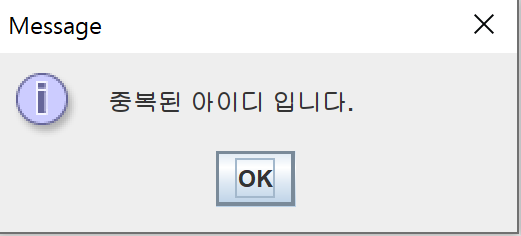
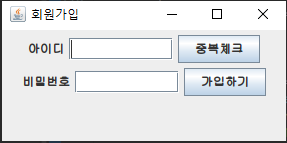
****

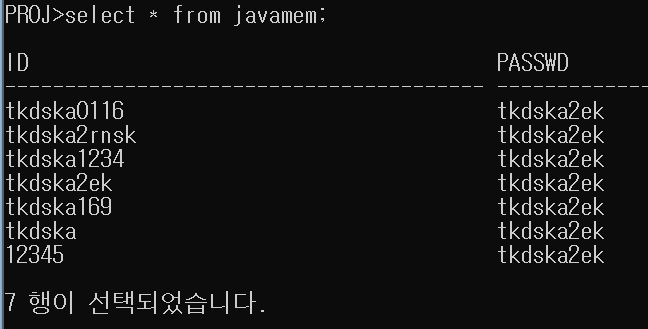
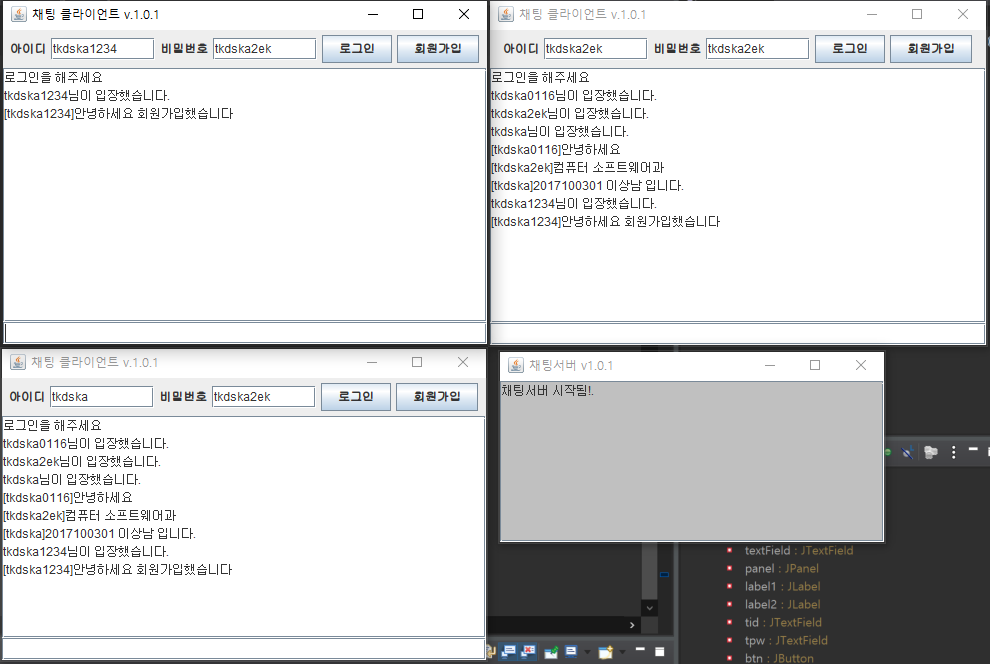
**테이블은 간단히 아이디와 비밀번호만 주었고, 아이디를 기본키로 설정하였습니다.**

1. **서버 클라이언트 로그인 or 채팅**

****

1. **회원가입창(중복체크 or 빈값체크 or 회원가입)  
   중복 체크 여부 와 빈칸이 있을경우 경고창을 띄우고, 회원가입이 성공하면 DB에 값을 넣습니다.**

****

1. **회원가입 후 결과  
   회원가입후 tkdska1234와 tkdska2ek 값이 테이블로 들어가고  
     
   바로 로그인해서 다른 유저들과 채팅이가능합니다.  
   **