Documentation

What is Java GUI

GUI (Graphical User Interface) in Java is an easy-to-use visual experience builder for Java applications. It is mainly made of graphical components like buttons, labels, windows, etc. through which the user can interact with an application. GUI plays an important role to build easy interfaces for Java applications.

About this project

Student Registration Application is a Java GUI (Graphical User Interface) application which is connected to SQL Database that store the user information for Student Collage or School Registration, Only Administrators can access this application, this application is encrypted by the Authentication there can be multiple administrators in one app therefor more than one administrator can access this Registration Application form and it is a platform Independent and very flexible because This Application is built upon java Swing, Swing is very popular Package for GUI APIs Library.

It can be used by educational institutes or colleges to maintain the records of students easily. It also provides a less time-consuming process for viewing, adding, editing and deleting the marks of the students

Achieving this objective is difficult using a manual system as the information is scattered, can be redundant and collecting relevant information may be very time consuming. Student Registration System Project consists of two functional elements: an enhanced Student module for Registration, Edit Profile, View Test history, Change Password. The project provides facilities like online registration and profile creation of students thus reducing paperwork and automating the record generation process in an educational institution.

In details of this project

This project is CRUD Application means it can (Create, Read, Update, Delete) Elements on Database we can manipulate the database using Java GUI button on Click an event is generated (Action Event) that implements ActionListener Interface using adapter class we can directly on button actionPerfromed method

Like this:

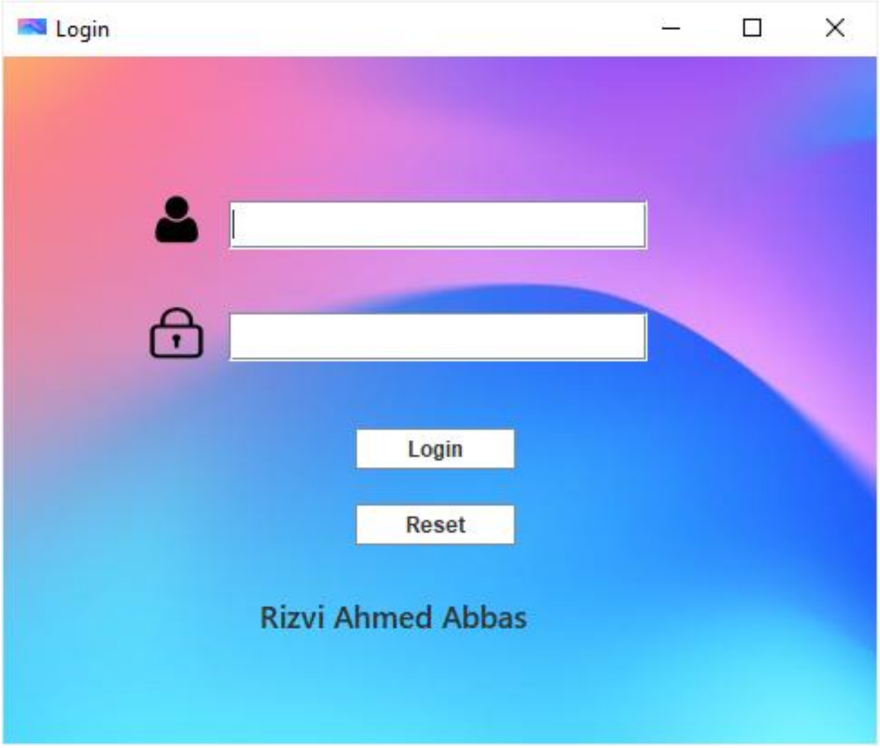
Syntax

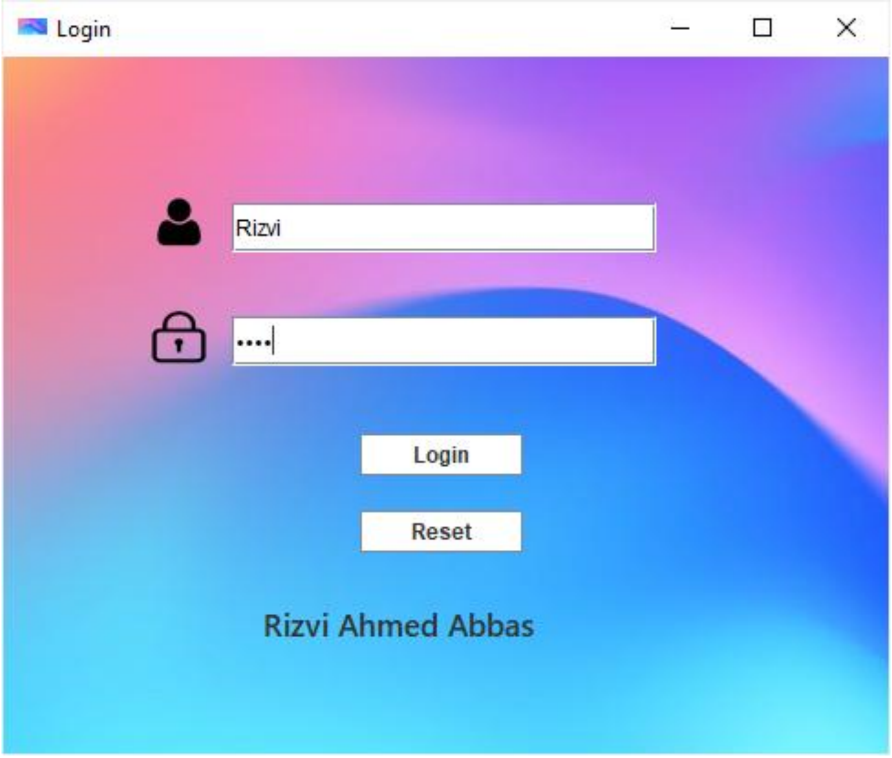
```
JButton UpdateButton = new JButton("Update");
UpdateButton.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent e) { }

)};
```

example

```
JButton UpdateButton = new JButton("Update");
UpdateButton.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent e) {


String StName,stENo,Stclass,stbatch,bid;


        StName = textName.getText();
        stENo = textEn.getText();
        Stclass = textClass.getText();
        stbatch = txtBatch.getText();
        bid = txtSearch.getText();


        try {
pst = con.prepareStatement("update student_regs set Name = ?,EnrollNo = ?,Class = ?,Batch = ?
where id = ?");
                        pst.setString(1, StName);
                        pst.setString(2, stENo);
                        pst.setString(3, Stclass);
                        pst.setString(4, stbatch);
                        pst.setString(5, bid);
                        pst.executeUpdate();
        JOptionPane.showMessageDialog(null, "Record Updated Successfully!");

                        table_Load();
                        textName.setText("");
                        textEn.setText("");
                        textClass.setText("");
                        txtBatch.setText("");
                        textName.requestFocus();
                                }

                catch(SQLException e1) {
                e1.printStackTrace();
                                }


                        }
                });
```

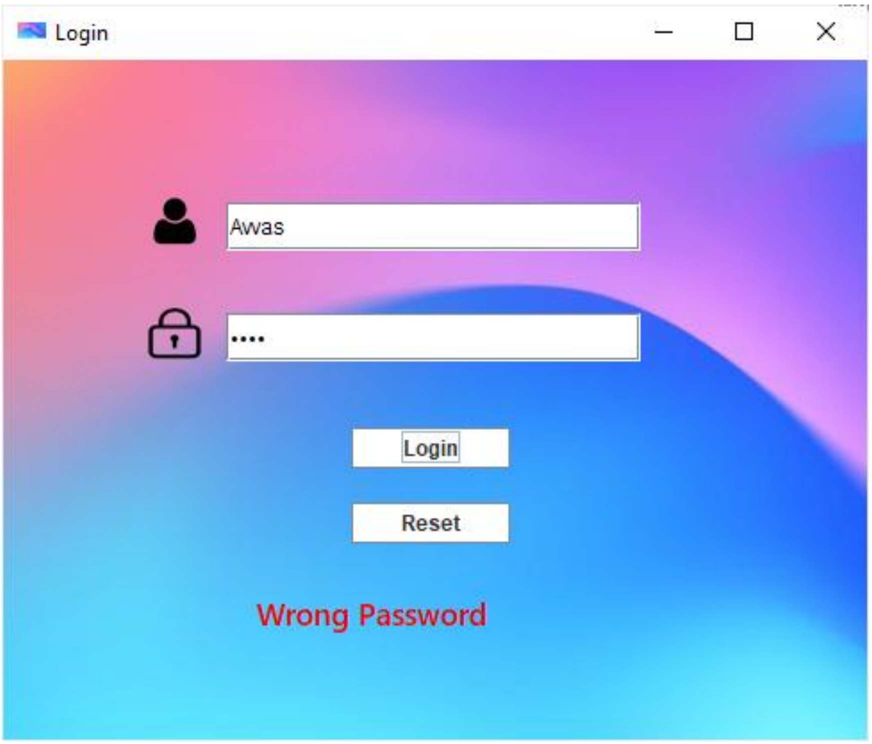
This application is composed of two applications (Login Frame & Student Registration Frame), when you run this program, you have to login in first and if you are not authorize person then you cannot login otherwise you can login to your Student Registration Application Form.


Login
Login
Reset
Rizvi Ahmed Abbas


Login
Login
Reset
Rizvi Ahmed Abbas

Rizvi

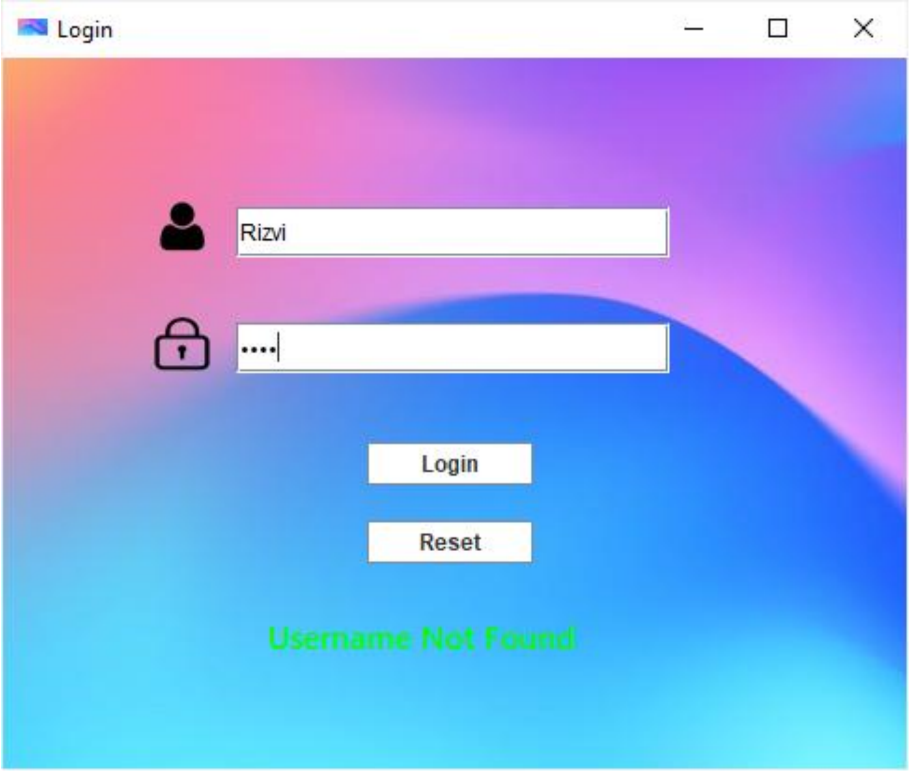
••••



Awas

••••

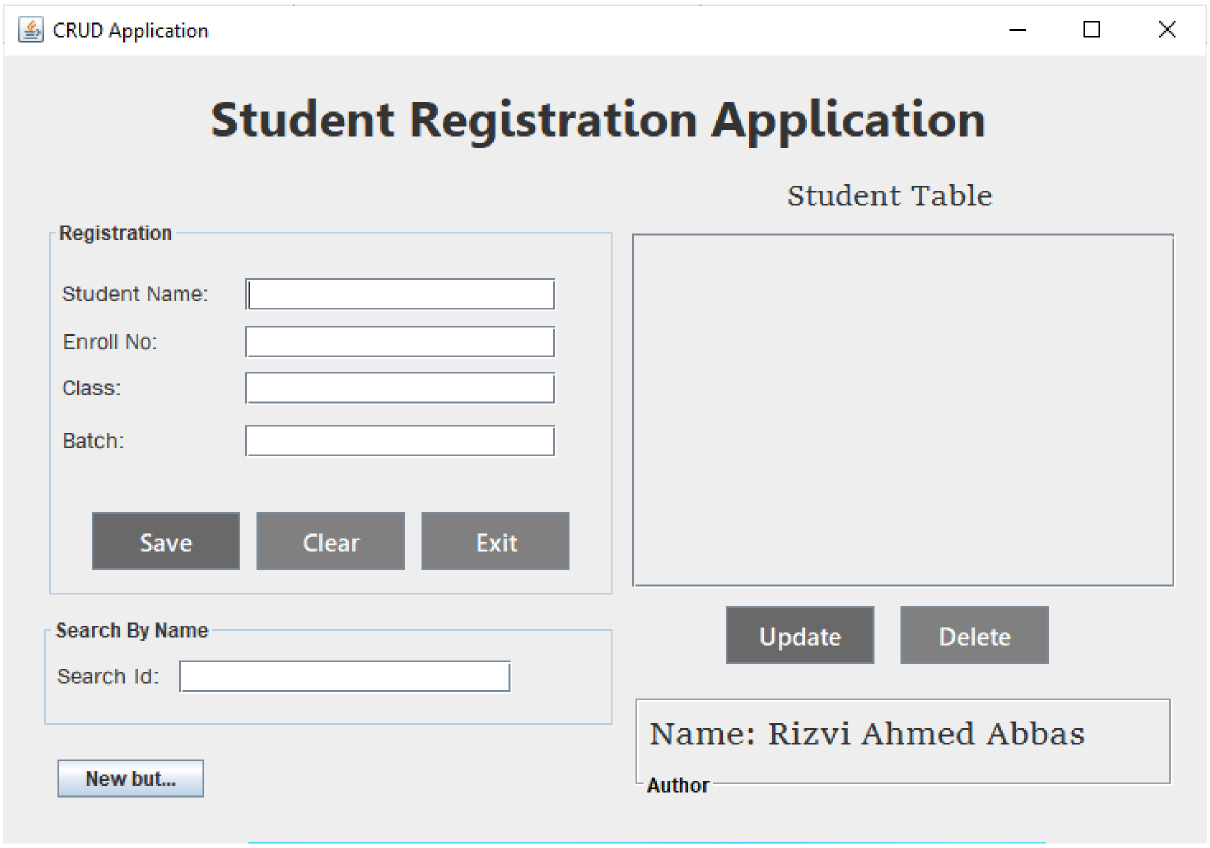
Wrong Password



Rizvi

••••

Username Not Found



CRUD Application

Student Registration Application

Student Table

Registration

Student Name:

Enroll No:

Class:

Batch:

Search By Name

Search Id:

Name: Rizvi Ahmed Abbas

Author

What is DBMS

Database management system. Connolly and Begg define database management system (DBMS) as a "software system that enables users to define, create, maintain and control access to the database". Examples of DBMS's include MySQL, PostgreSQL, Microsoft SQL Server, Oracle Database, and Microsoft Access.

Database Management Systems (DBMS) are software systems used to store, retrieve, and run queries on data. A DBMS serves as an interface between an end-user and a database, allowing users to create, read, update, and delete data in the database.

Types of DBMS

1. Relational database.
2. Object oriented database.
3. Hierarchical database.
4. Network database.

What is RDBMS?

RDBMS stands for Relational Database Management System. RDBMS is the basis for SQL, and for all modern database systems such as MS SQL Server, IBM DB2, Oracle, MySQL, and Microsoft Access. The data in RDBMS is stored in database objects called tables.

Steps For Connectivity Between Java Program and Database

1. Import the Packages.
2. Load the drivers using the forName() method.
3. Register the drivers using DriverManager.
4. Establish a connection using the Connection class object.
5. Create a statement.
6. Execute the query.
7. Close the connections.

ODBC

ODBC stands for Open Database Connectivity A standard or open application programming interface (API) for accessing a database. ODBC provides a C interface for database access on Windows environment.

Open Database Connectivity is a standard application programming interface for accessing database management systems. The designers of ODBC aimed to make it independent of database systems and operating systems.

What is ODBC used for?

An ODBC driver uses the Open Database Connectivity (ODBC) interface by Microsoft that allows applications to access data in database management systems (DBMS) using SQL as a standard for accessing the data. ODBC permits maximum interoperability, which means a single application can access different DBMS.

JDBC

Java Database Connectivity is an application programming interface for the programming language Java, which defines how a client may access a database. It is a Java-based data access technology used for Java database connectivity. It is part of the Java Standard Edition platform, from Oracle Corporation.

What is JDBC used for?

Java™ database connectivity (JDBC) is the JavaSoft specification of a standard application programming interface (API) that allows Java programs to access database management systems. The JDBC API consists of a set of interfaces and classes written in the Java programming language.

The JDBC API is a Java API that can access any kind of tabular data, especially data stored in a relational database. JDBC helps you to write Java applications that manage these three programming activities: Connect to a data source, like a database. Send queries and update statements to the database.

How to connect to database using JBDC

```
Connection con;
PreparedStatement pst;
ResultSet rs;
String url = "com.mysql.cj.jdbc.Driver";



private void Connect() {
        {
                try
                {
Class.forName(url);
con = DriverManager.getConnection("jdbc:mysql://localhost/javacrud", "root", "");
                }
                catch (ClassNotFoundException ex)
                {

                }
                catch(SQLException ex)
                {

                }
        }
}

public void table_Load() {
        {
                try {
                        pst = con.prepareStatement("Select * from student_regs");
                        rs = pst.executeQuery();
                        table.setModel(DbUtils.resultSetToTableModel (rs));
                }
                catch(SQLException e) {
                        e.printStackTrace();
}
        }
}
```

What is a Prepare statement

What is PreparedStatement in Java?

A PreparedStatement is a pre-compiled SQL statement. It is a subinterface of Statement. Prepared Statement objects have some useful additional features than Statement objects. Instead of hard coding queries, PreparedStatement object provides a feature to execute a parameterized query.

The PreparedStatement interface is a subinterface of Statement. It is used to execute parameterized query.

```
String sql="insert into emp values(?,?,?)";
```

What is XAMPP server

The goal of XAMPP is to build an easy to install distribution for developers to get into the world of Apache. To make it convenient for developers, XAMPP is configured with all features turned on.

is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages

XAMPP is a software distribution which provides the Apache web server, MySQL database (actually MariaDB), Php and Perl (as command-line executables and Apache modules) all in one package. It is available for Windows, MAC and Linux systems. No configuration is necessary to integrate Php with MySQL

Driver used in this project

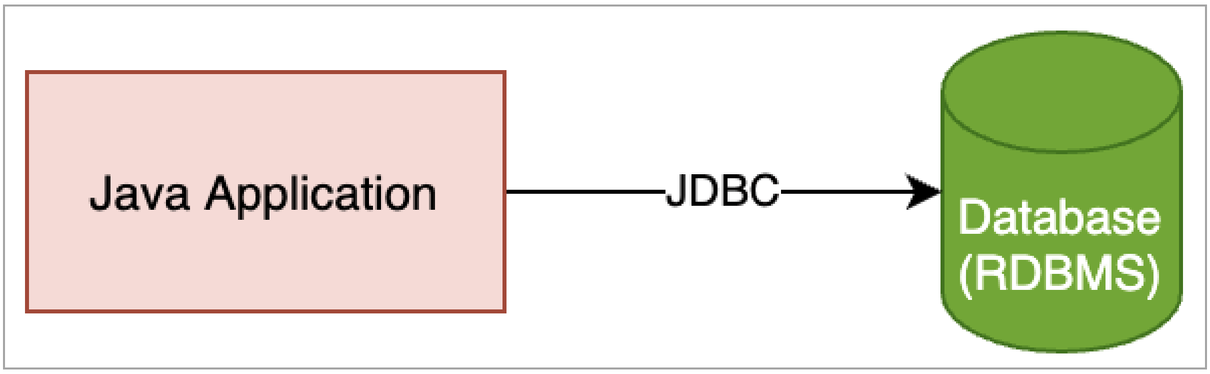
mysql-connector-java

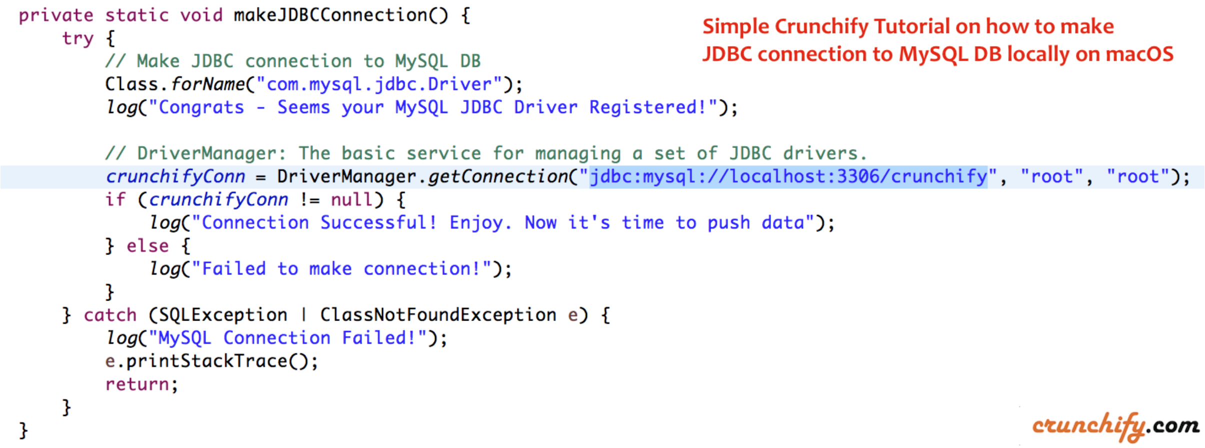
MySQL provides connectivity for Java client applications with MySQL Connector/J, a driver that implements the Java Database Connectivity (JDBC) API. The API is the industry standard for database-independent connectivity between the Java programming language and a wide range of – SQL databases, spreadsheets etc

mysql-connector-java-8.0.30

MySQL Connector/J is a JDBC Type 4 driver, implementing the JDBC 4.2 specification. The Type 4 designation means that the driver is a pure Java implementation of the MySQL protocol and does not rely on the MySQL client libraries

MySQL Connector/Python is a standardized database driver for Python platforms and development. Additionally, MySQL Connector/Python 8.0 supports the new X DevAPI for development with MySQL Server 8.0. Online Documentation: MySQL Connector/Python Installation Instructions. Documentation.

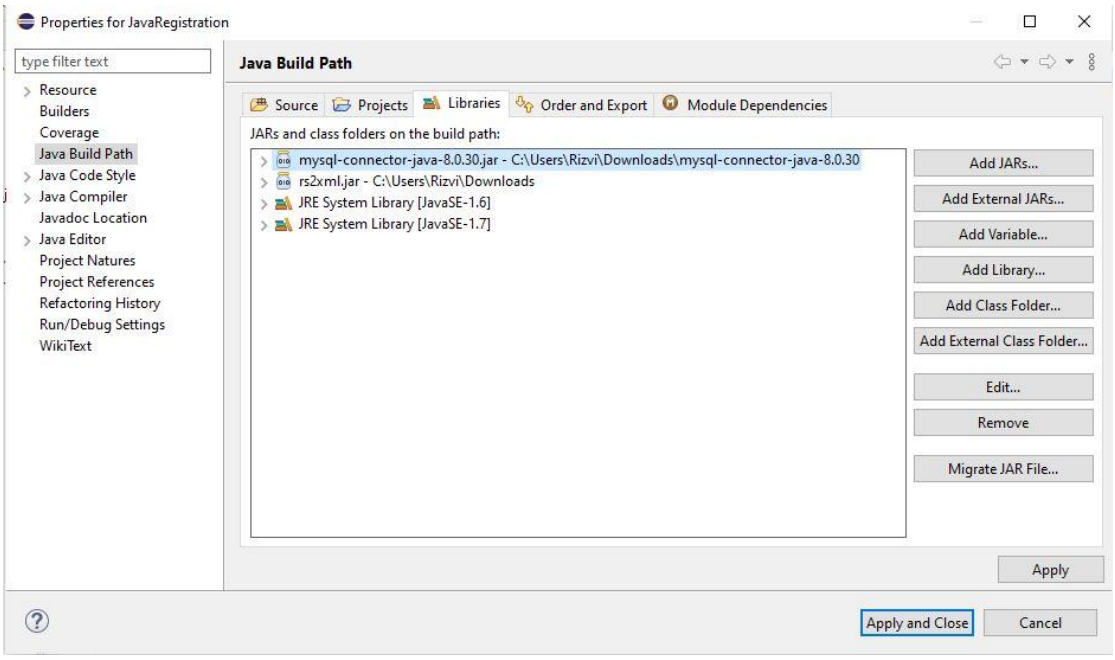




```
private static void makeJDBCConnection() {
    try {
        // Make JDBC connection to MySQL DB
        Class.forName("com.mysql.jdbc.Driver");
        log("Congrats - Seems your MySQL JDBC Driver Registered!");

        // DriverManager: The basic service for managing a set of JDBC drivers.
        crunchifyConn = DriverManager.getConnection("jdbc:mysql://localhost:3306/crunchify", "root", "root");
        if (crunchifyConn != null) {
            log("Connection Successful! Enjoy. Now it's time to push data");
        } else {
            log("Failed to make connection!");
        }
    } catch (SQLException | ClassNotFoundException e) {
        log("MySQL Connection Failed!");
        e.printStackTrace();
        return;
    }
}
```

Simple Crunchify Tutorial on how to make JDBC connection to MySQL DB locally on macOS

crunchify.com



JARs and class folders on the build path:

- mysql-connector-java-8.0.30.jar - C:\Users\Rizvi\Downloads\mysql-connector-java-8.0.30
- rs2xml.jar - C:\Users\Rizvi\Downloads
- JRE System Library [JavaSE-1.6]
- JRE System Library [JavaSE-1.7]

what is rs2xml

The rs2xml jar is used to display the data in a table format. So, once you create a project in Eclipse IDE, you have to import the rs2xml jar and JDBC connector JAR into the project. To do that, right-click on the project, choose Build Path -> Configure Build Path