

# Java CRUD System

## Developer Setup & Build Guide

*Step-by-step guide for setting up, building, and running the application*

**Java 11+****Maven****SQL Server****Swing GUI**

---

### About This Guide

This document provides a complete, step-by-step walkthrough for any developer receiving the Java CRUD System source code. It covers all prerequisites, environment setup, database configuration, building, and running the application.

Item	Detail
Application	Java Swing CRUD System
Database	Microsoft SQL Server (SampleDB)
Build Tool	Apache Maven
Java Version	JDK 11 or higher
Entry Point	gui.LoginFrame
Default Login	admin / 1234

## SECTION 1 — Prerequisites

Before building or running the project, make sure all of the following tools are installed on your machine.

### 1.1 Java Development Kit (JDK 11 or higher)

The application requires Java JDK 11+. Note that JRE alone is not sufficient — you need the full JDK to compile source files.

1. Open a terminal or PowerShell window.
2. Run the following command to check if Java is installed:

```
java -version
```

3. If installed correctly, you will see output similar to:

```
java version "11.0.x" 2024-xx-xx Java(TM) SE Runtime Environment (build 11.0.x+x) Java HotSpot(TM) 64-Bit Server VM (build 11.0.x+x, mixed mode)
```

4. If Java is not installed, download and install from:

- <https://adoptium.net/> (recommended — free, open source)
- <https://www.oracle.com/java/technologies/downloads/>

✔ **Required:** JDK 11 or higher. Anything below JDK 11 will NOT work.

## 1.2 Apache Maven

Maven is the build tool used to compile the project and download all dependencies (including the SQL Server JDBC driver).

5. Check if Maven is installed:

```
mvn -version
```

6. Expected output:

```
Apache Maven 3.x.x Maven home: C:\...\apache-maven-3.x.x Java version: 11.x.x
```

7. If not installed, download from: <https://maven.apache.org/download.cgi>

8. Extract the ZIP to a folder, e.g.: C:\Program Files\Apache\maven\apache-maven-3.9.6

9. Add Maven to your System PATH environment variable:

- Open: System Properties → Advanced → Environment Variables
- Under System Variables, add a new entry: M2\_HOME = C:\Program Files\Apache\maven\apache-maven-3.9.6
- Edit the Path variable and add: %M2\_HOME%\bin

10. Close and reopen your terminal, then run `mvn -version` to confirm.

□ **Note:** Always close and reopen PowerShell/CMD after changing environment variables — changes do not take effect in already-open terminals.

## 1.3 Microsoft SQL Server

The application connects to Microsoft SQL Server using JDBC. SQL Server Express (free) is sufficient.

- Download SQL Server Express: <https://www.microsoft.com/en-us/sql-server/sql-server-downloads>
- Download SQL Server Management Studio (SSMS): <https://learn.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms>

□ **Info:** SQL Server Express is free and handles all the features this application requires. SSMS is the recommended GUI tool for managing the database.

# SECTION 2 — Database Setup

The application expects a database called SampleDB with two tables: Users and Products. These should already exist if you received a pre-configured database. Follow the steps below to verify or create them.

## 2.1 Verify or Create the Database

---

Open SQL Server Management Studio (SSMS), connect to your server, and run the following SQL:

```
-- Check if database exists SELECT name FROM sys.databases WHERE name = 'SampleDB';
```

If the database does not exist, create it with the full schema:

```
CREATE DATABASE SampleDB; GO USE SampleDB; CREATE TABLE Users (      Id
INT IDENTITY PRIMARY KEY,      Username NVARCHAR(50) UNIQUE,      Password
NVARCHAR(255) ); CREATE TABLE Products (      Id      INT IDENTITY PRIMARY KEY,
Name NVARCHAR(100),      Price DECIMAL(10,2) ); -- Default admin user INSERT
INTO Users (Username, Password) VALUES ('admin', '1234');
```

## 2.2 Verify the Admin User

---

Confirm the default login account exists by running:

```
USE SampleDB; SELECT * FROM Users WHERE Username = 'admin';
```

You should see one row returned with Username = 'admin' and Password = '1234'.

## 2.3 Enable SQL Server Authentication

---

The application uses SQL Server Authentication (username + password), not Windows Authentication. You must ensure this mode is enabled:

11. In SSMS, right-click the server name in Object Explorer.
12. Select Properties.
13. Go to the Security page.
14. Under Server Authentication, select: SQL Server and Windows Authentication mode.
15. Click OK and restart SQL Server when prompted.

## 2.4 Enable TCP/IP and Start SQL Server Browser

---

The JDBC driver needs TCP/IP enabled and the SQL Server Browser service running to locate named instances.

16. Open SQL Server Configuration Manager (search in Start Menu).
17. Expand SQL Server Network Configuration.

18. Click Protocols for SQLEXPRESS (or your instance name).
19. Right-click TCP/IP and select Enable.
20. Go to SQL Server Services and right-click SQL Server Browser → Start.
21. Set SQL Server Browser Startup Type to Automatic.
22. Right-click your SQL Server instance → Restart.

❑ **Important:** If TCP/IP is not enabled, the application will receive a `SocketTimeoutException` on port 1434 and will not be able to connect — even if credentials are correct.

## SECTION 3 — Project Configuration

Before building, you must update the database connection settings to match your local SQL Server installation.

### 3.1 Find Your SQL Server Instance Name

Open SSMS and check the Server name field in the connection dialog. It will look like one of these:

What you see in SSMS	Your instance type	Connection string to use
DESKTOP-ABC\SQLEXPRESS	SQL Server Express	DESKTOP-ABC\SQLEXPRESS
localhost	Default instance	localhost:1433
MYPC\SQLSERVER2022	Named instance	MYPC\SQLSERVER2022
192.168.1.100	Remote server	192.168.1.100:1433

Alternatively, run this in PowerShell to list all SQL Server services on your machine:

```
Get-Service -Name 'MSSQL*' | Select-Object Name, Status
```

### 3.2 Edit DatabaseConfig.java

Open the following file in any text editor (Notepad, VS Code, etc.):

```
src\config\DatabaseConfig.java
```

Find and update the three lines shown below. Replace the placeholders with your actual values:

```
private static final String URL =  
"jdbc:sqlserver://YOUR_SERVER_NAME;databaseName=SampleDB;" +  
"encrypt=true;trustServerCertificate=true"; private static final String  
USERNAME = "sa"; // your SQL login private static final String PASSWORD  
= "YourPassword"; // your SQL password
```

Examples based on common setups:

Setup	URL value
SQLEXPRESS (named)	jdbc:sqlserver://ACER-NITROV15-F\\SQLEXPRESS;databaseName=SampleDB;encrypt=true;trustServerCertificate=true
Default instance	jdbc:sqlserver://localhost:1433;databaseName=SampleDB;encrypt=true;trustServerCertificate=true
Windows Auth	...same URL + integratedSecurity=true; leave USERNAME/PASSWORD empty

❏ **Java String Note:** In Java strings, backslashes must be doubled. Write `ACER\\SQLEXPRESS` in code even though Windows shows it as `ACER\SQLEXPRESS`.

## SECTION 4 — Building the Project

With the database configured, you are ready to build. Maven will automatically download the SQL Server JDBC driver and compile all source files.

### 4.1 Open a Terminal in the Project Folder

The easiest way on Windows is:

- 23. Open File Explorer and navigate to the JavaCRUDSystem folder.
- 24. Hold Shift and right-click on the folder.
- 25. Select Open PowerShell window here (or Open Command Prompt here).

Or navigate manually in PowerShell:

```
| cd C:\Users\YourName\Documents\JavaCRUDSystem
```

### 4.2 Run the Build Command

```
| mvn clean install
```

Maven will:

- Download all dependencies (requires internet on first run)
- Compile all Java source files
- Package everything into a single executable JAR file

A successful build ends with:

```
| [INFO] BUILD SUCCESS [INFO] Total time: xx.xxx s [INFO] Finished at: 2024-xx-xx
```

### 4.3 What Gets Created

After a successful build, the following file is created:

```
| target\JavaCRUDSystem-1.0-SNAPSHOT.jar
```

This is a self-contained JAR that includes the application and all dependencies. You only need this one file to run the application (plus a running SQL Server instance).

❑ **Build Troubleshooting:** If the build fails, the most common causes are: (1) No internet connection on first run, (2) Java version below 11, (3) JAVA\_HOME not set. Run `java -version` and `mvn -version` to verify both are installed correctly.

## SECTION 5 — Running the Application

### 5.1 Run the JAR File

From the project folder in PowerShell or CMD:

```
| java -jar target\JavaCRUDSystem-1.0-SNAPSHOT.jar
```

The login window will appear within a few seconds.

### 5.2 Login

Use the default credentials:

Field	Value
Username	admin
Password	1234

### 5.3 Application Features

After logging in, you will see two tabs in the main window:

Tab	What it manages	Operations available
Products	Product name and price	Add, Update, Delete, Search, Refresh
Users	System user accounts	Add, Update, Delete, Refresh

## SECTION 6 — Project Structure

Understanding the codebase layout will help developers navigate and modify the application.

File / Folder	Purpose
src/config/DatabaseConfig.java	Database connection settings. Edit this to change server, username, or password.
src/model/User.java	User data model (Id, Username, Password).
src/model/Product.java	Product data model (Id, Name, Price).
src/dao/UserDAO.java	All SQL operations for the Users table (CRUD + login auth).
src/dao/ProductDAO.java	All SQL operations for the Products table (CRUD + search).
src/gui/LoginFrame.java	Login screen. This is the application entry point (contains main method).
src/gui/MainFrame.java	Main window after login. Hosts the tabbed interface and logout button.
src/gui/ProductPanel.java	UI panel for Product CRUD operations.
src/gui/UserPanel.java	UI panel for User CRUD operations.
pom.xml	Maven configuration. Defines the MSSQL JDBC dependency and build settings.
target/ (generated)	Build output folder. Contains the compiled classes and executable JAR.

## SECTION 7 — Troubleshooting

### 7.1 Connection Errors

Error Message	Fix
SocketTimeoutException on port 1434	Start SQL Server Browser service in services.msc. Enable TCP/IP in SQL Server Configuration Manager.
Login failed for user 'sa'	Enable SQL Server Authentication mode in SSMS Server Properties → Security. Verify your password is correct.
Cannot open database 'SampleDB'	Run the CREATE DATABASE script in Section 2. Make sure the database name matches exactly.
SQL Server JDBC Driver not found	Run mvn clean install to download dependencies. Check internet connection.

The server name could not be found	Double-check your server name in DatabaseConfig.java. Use the exact name shown in the SSMS connection dialog.
------------------------------------	---

## 7.2 Build Errors

Error	Fix
mvn is not recognized	Maven is not installed or not in PATH. See Section 1.2.
BUILD FAILURE — compiler error	Check Java version: java -version must show 11 or higher.
BUILD FAILURE — dependency error	Check internet connection. Run: mvn clean install -U
JAVA_HOME not set	Set JAVA_HOME environment variable to your JDK installation folder.
Invalid signature file digest (JNI error)	The pom.xml is missing the Shade plugin filter. Add META-INF/*.SF, *.DSA, *.RSA exclusions.

## 7.3 Runtime Errors

Error	Fix
Could not find or load main class gui.LoginFrame	Rebuild with mvn clean install. Make sure you are in the project root folder.
Login window does not appear	Check console output. Likely a database connection failure — fix DatabaseConfig.java first.
Invalid username or password	Verify the admin user exists: <code>SELECT * FROM Users WHERE Username = 'admin'</code> .
Button text not visible	This is a Windows Look & Feel issue. The paintComponent override in each button class fixes it. Rebuild with the latest source.

# SECTION 8 — Quick Command Reference

All commands assume PowerShell or CMD, run from the project root folder.

Task	Command
Check Java version	java -version
Check Maven version	mvn -version



Navigate to project	<code>cd C:\path\to\JavaCRUDSystem</code>
Build the project	<code>mvn clean install</code>
Run the application	<code>java -jar target\JavaCRUDSystem-1.0-SNAPSHOT.jar</code>
Clean build output	<code>mvn clean</code>
Force dependency update	<code>mvn clean install -U</code>
Check SQL Server services	<code>Get-Service -Name 'MSSQL*'   Select-Object Name, Status</code>
Start SQL Server Browser	<code>Start-Service SQLBrowser</code> (run as Admin)
Restart SQL Server Express	<code>Restart-Service 'MSSQL\$SQLEXPRESS'</code> (run as Admin)

## APPENDIX — Pre-Launch Checklist

Use this checklist every time you set up the project on a new machine:

<input type="checkbox"/>	Java JDK 11 or higher is installed ( <code>java -version</code> )
<input type="checkbox"/>	Apache Maven is installed ( <code>mvn -version</code> )
<input type="checkbox"/>	SQL Server is installed and running
<input type="checkbox"/>	TCP/IP protocol is enabled in SQL Server Configuration Manager
<input type="checkbox"/>	SQL Server Browser service is running and set to Automatic
<input type="checkbox"/>	SQL Server and Windows Authentication mode is enabled
<input type="checkbox"/>	SampleDB database exists with Users and Products tables
<input type="checkbox"/>	Admin user exists: Username = admin, Password = 1234
<input type="checkbox"/>	DatabaseConfig.java updated with correct server name
<input type="checkbox"/>	DatabaseConfig.java updated with correct USERNAME and PASSWORD
<input type="checkbox"/>	<code>mvn clean install</code> completes with BUILD SUCCESS
<input type="checkbox"/>	<code>java -jar target\JavaCRUDSystem-1.0-SNAPSHOT.jar</code> launches the login window
<input type="checkbox"/>	Login with admin / 1234 is successful

**Once all boxes above are checked, your Java CRUD System is fully configured and ready to use.**