

# CSE385 Final Project Documentation

## Installation Instructions:

1. I used Java 11 in the 'Eclipse IDE for Enterprise Java and Web Developers', hence I strongly suggest using this particular IDE.
2. Add the submitted .sql files to your MySQL Workbench. Name your database, '**inventory**' as the Java program recognizes the inventory database for all the GUI operations.

3. For establishing a successful connection between the Java program and the Inventory Database in your MySQL workbench, you will need to add your MySQL Workbench username and password to line 45. It looks like this:

Line 44 // Connection for MySQL Database

Line 45 `private static final String JDBC_URL =  
"jdbc:mysql://localhost:3306/inventory?user=root&password=1234";`

So instead of user='root' and password='1234'

Add your username and password. For example, if your username is 'manas' and your password is '5678'. The change should look like:

Line 44 // Connection for MySQL Database

Line 45 `private static final String JDBC_URL =  
"jdbc:mysql://localhost:3306/inventory?user=manas&password=5678";`

This change is absolutely necessary as without this change the Java program will **NOT** be able to access the inventory database in your MySQL Workbench. It requires your current username and password to do so. **IT IS ESSENTIAL FOR THE APPLICATION TO WORK.**

4. You will need to add an external JAR file to ensure a successful connection between the Java program and the SQL Database.

The name of the JAR file is: **mysql-connector-j-8.2.0.jar**

Access it from this link: <https://dev.mysql.com/downloads/connector/j/5.1.html>

Go to the website and select '**Platform Independent**' as your **selected operating system**. Then you download this file - Platform Independent (Architecture Independent), ZIP Archive. After you download it add it to your project's referenced libraries.

5. This is how you can add the JAR file to your Eclipse Java Project:

- Go to your project name, 'Inventory Management System' in my case.
- Right Click it and select 'Build Path', then select 'Configure Build Path'

- Following that select 'Classpath' under Libraries.
- Then select the option 'Add External JARS' from the right side of the screen.
- Following that select the downloaded **mysql-connector-j-8.2.0.jar** file.
- Then Apply and Close it.
- Following this, you will see a 'Referenced Libraries' tab in your Java project folder. The **mysql-connector-j-8.2.0.jar** file should be present there and that should allow you to successfully and securely connect to the inventory database in your MySQL Workbench.

### GUI Usage Instructions:

I have submitted an Inventory Database Management System and I am explaining how it works. Here are the instructions:

The application has the ability to add and remove products along with updating the product stock information in the database. It also finds products, their supplier's information, and the customer's information. For processing the added and existing information, it can display the top 5 products that are dangerously low in stock.

- For adding the product information:

1. Fill out all the fields with the details of the product in the Product Information Section.
2. Then, click on the 'Add Product' button under the 'Actions' Section.
3. The product details will appear under the Inventory Screen and will be successfully added to the products table in the inventory database.

- For removing a product from the database:

1. Click 'Remove Product' button
2. Then a dialogue box will pop up asking: Enter the product name to remove
3. Enter the product name and click ok
4. If the product exists, it will be successfully be removed from the database.

- Update Stock:

1. Click 'Update Stock' button
2. Then a dialogue box will pop up asking: Enter the product name to update
3. Following that, it will ask you to update all the required product information. Once you submit ok, it will update the information in the database.

\*\* If the quantity of the product goes below the reordered threshold, the system will give an low stock alert and add the product details in the lowstockalerts table in the database.

- Find Product/Supplier's/Customer's Info
  1. Click the button and it will ask you to enter the product/supplier/customer name
  2. Following that, press ok and it will give you all the detailed information regarding the product/supplier/customer.
  
- For the "Display Top 5 Low Stock Products"
  1. All the products have a quantity which is in-store. If the quantity goes lower than the reorder threshold. A low stock alert is sent and the product details are indeed in the lowstockalerts table in the database. The Display Top 5 Low Stock Products button displays the top five products that are dangerously low in stock from the updated list of products.

### Resources Used (Works Cited):

- I. Datasets & Data Gathering Resource - Used the data-generating application, Mockaroo. Used it to generate my products.csv, customers.csv, and suppliers.csv files. Link - <https://www.mockaroo.com/>
  
- II. External Libraries - I needed to add the **mysql-connector-j-8.2.0.jar** as one of my referenced libraries to effectively connect my Java program to my inventory database in MySQL Workbench. I downloaded the Active JAR file from <https://dev.mysql.com/downloads/connector/j/5.1.html>