DBMS Mini Project Report

Title: Online Shopping Management System (Using Java JDBC and MySQL)
Name: Siddharth Jagdale
Division: D10C
Roll Number: 21
Course: Database Management Systems (DBMS)
Semester: III
Guide: Kajal Jewani
Date: 08 | 10 | 2025

Abstract:

This project titled *“Online Shopping Management System”* demonstrates the use of **Java JDBC** for database connectivity and **Swing GUI** for front-end design. The system allows users to **view, add, update, delete, and search products**, as well as **add items to a shopping cart**.
The backend database is managed using **MySQL**, and data operations are performed through the **JDBC API**. The project serves as a foundation for understanding how desktop applications can interact with relational databases effectively.

Prerequisites:

Before running the project, ensure that the following hardware and software requirements are met:

Hardware Requirements:

- Minimum 4 GB RAM
- Intel i3 processor or higher
- At least 500 MB free disk space

Software Requirements:

- Operating System: Windows 10 or above
- Database: MySQL Server 8.0
- IDE: Visual Studio Code
- Build Tool: Apache Maven
- Programming Language: Java (JDK 17)

Tools & Versions Used

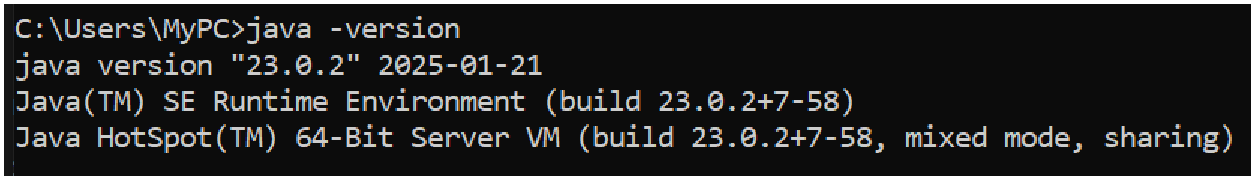
Tool / Library	Version	Purpose	
Java JDK	17.0.8	Programming language	
MySQL Server	8.0.33	Database backend	
MySQL Workbench	8.0.33	GUI tool for MySQL	
Apache Maven	3.9.6	Build automation and dependency management	
MySQL Connector/J	8.0.33	JDBC driver for MySQL	
Visual Studio Code	1.91+	Java IDE	
Windows PowerShell / CMD	—	For running Maven commands	

Environment Setup (Step-by-Step)

Step 1: Install JDK 17

Download and install JDK 17 from the official Oracle website.
Set up the environment variables (JAVA_HOME and add JDK bin to PATH).

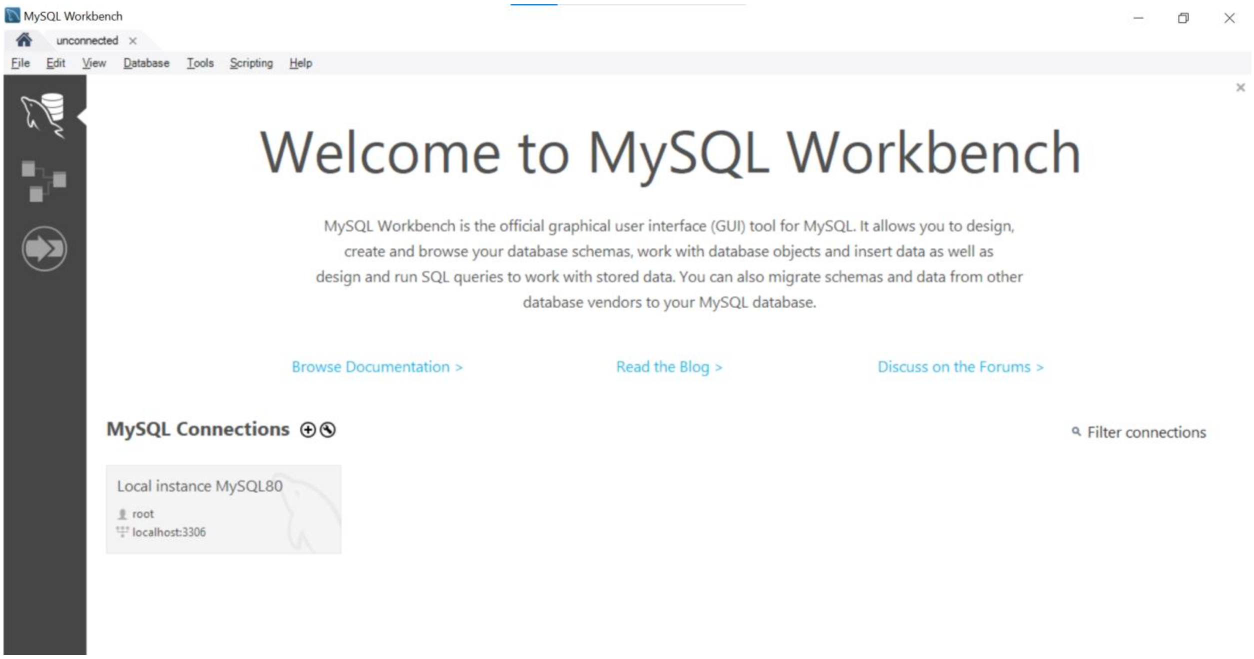
Command to verify installation:



```
C:\Users\MyPC>java -version
java version "23.0.2" 2025-01-21
Java(TM) SE Runtime Environment (build 23.0.2+7-58)
Java HotSpot(TM) 64-Bit Server VM (build 23.0.2+7-58, mixed mode, sharing)
```

Step 2: Install MySQL Server and Workbench

Download and install **MySQL Server** and **MySQL Workbench**.
During setup, create a root password (e.g., root) and note it down for later use.



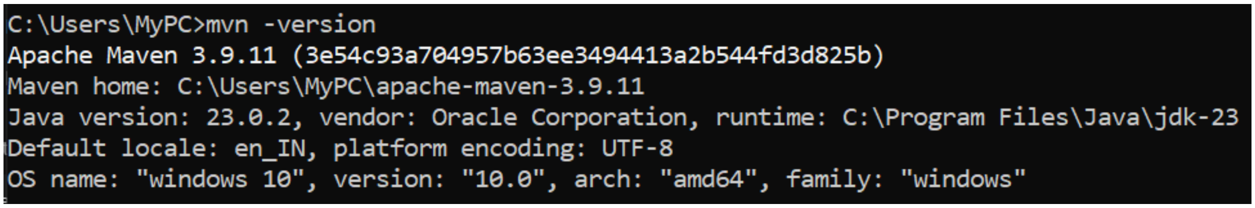
Welcome to MySQL Workbench

MySQL Workbench is the official graphical user interface (GUI) tool for MySQL. It allows you to design, create and browse your database schemas, work with database objects and insert data as well as design and run SQL queries to work with stored data. You can also migrate schemas and data from other database vendors to your MySQL database.

Step 3: Install Apache Maven

Download and extract Apache Maven, then set its bin folder in your PATH environment variable.

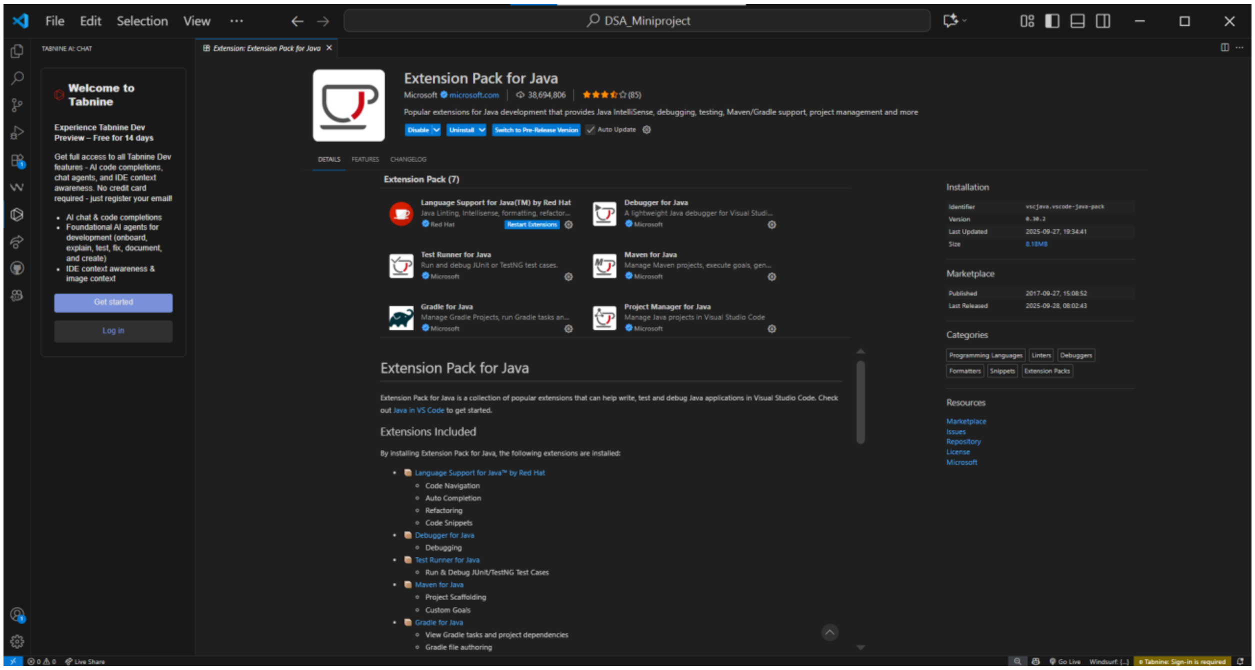
Command to verify installation:



```
C:\Users\MyPC>mvn -version
Apache Maven 3.9.11 (3e54c93a704957b63ee3494413a2b544fd3d825b)
Maven home: C:\Users\MyPC\apache-maven-3.9.11
Java version: 23.0.2, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk-23
Default locale: en_IN, platform encoding: UTF-8
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"
```

Step 4: Install Visual Studio Code

Download and install **VS Code** from https://code.visualstudio.com.
Install the **“Extension Pack for Java”** from the Extensions tab.



Welcome to Tabnine

Experience Tabnine Dev Preview – Free for 14 days

Get full access to all Tabnine Dev features - AI code completions, chat agents, and IDE context awareness. No credit card required - just register your email!

- AI chat & code completions
- Foundational AI agents for development (onboard, explain, test, fix, document, and create)
- IDE context awareness & image context

Extension Pack for Java

Microsoft microsoft.com | 38,694,806 | ★★★★☆ (85)

Popular extensions for Java development that provides Java IntelliSense, debugging, testing, Maven/Gradle support, project management and more

Extension Pack (7)

Language Support for Java(TM) by Red Hat
Java Linting, Intellisense, formatting, refactor...
Red Hat

Debugger for Java
A lightweight Java debugger for Visual Studi...
Microsoft

Test Runner for Java
Run and debug JUnit or TestNG test cases.
Microsoft

Maven for Java
Manage Maven projects, execute goals, gen...
Microsoft

Gradle for Java
Manage Gradle Projects, run Gradle tasks an...
Microsoft

Project Manager for Java
Manage Java projects in Visual Studio Code
Microsoft

Extension Pack for Java

Extension Pack for Java is a collection of popular extensions that can help write, test and debug Java applications in Visual Studio Code. Check out Java in VS Code to get started.

Extensions Included

By installing Extension Pack for Java, the following extensions are installed:

- Language Support for Java™ by Red Hat
  - Code Navigation
  - Auto Completion
  - Refactoring
  - Code Snippets
- Debugger for Java
  - Debugging
- Test Runner for Java
  - Run & Debug JUnit/TestNG Test Cases
- Maven for Java
  - Project Scaffolding
  - Custom Goals
- Gradle for Java
  - View Gradle tasks and project dependencies
  - Gradle file authoring

Installation

Identifier	vscjava.vscode-java-pack
Version	0.30.2
Last Updated	2025-09-27, 19:34:41
Size	8.18MB

Marketplace

Published	2017-09-27, 15:08:52
Last Released	2025-09-28, 08:02:43

Categories

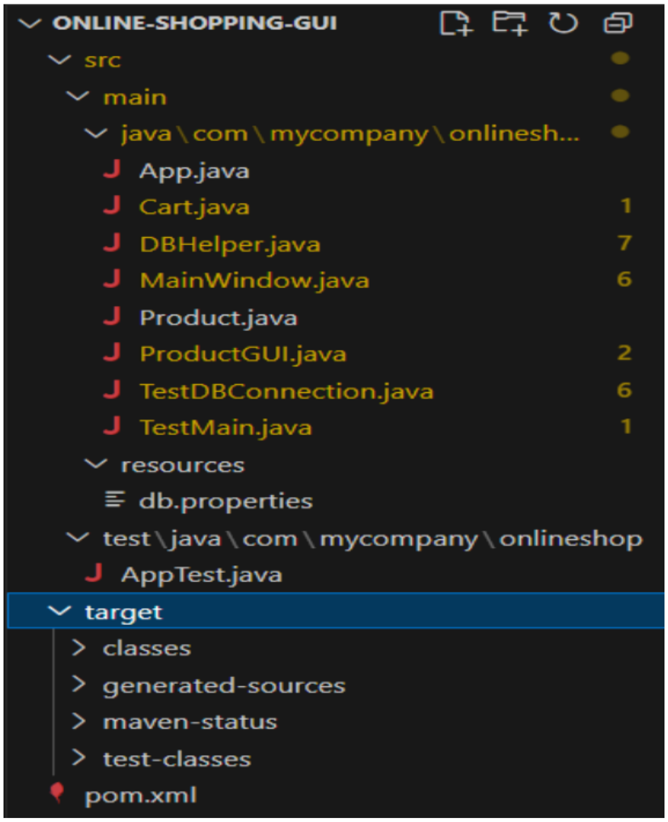
Programming Languages | Linters | Debuggers | Formatters | Snippets | Extension Packs

Resources

Marketplace
Issues
Repository
License
Microsoft

Step 5: Setup Project Folder

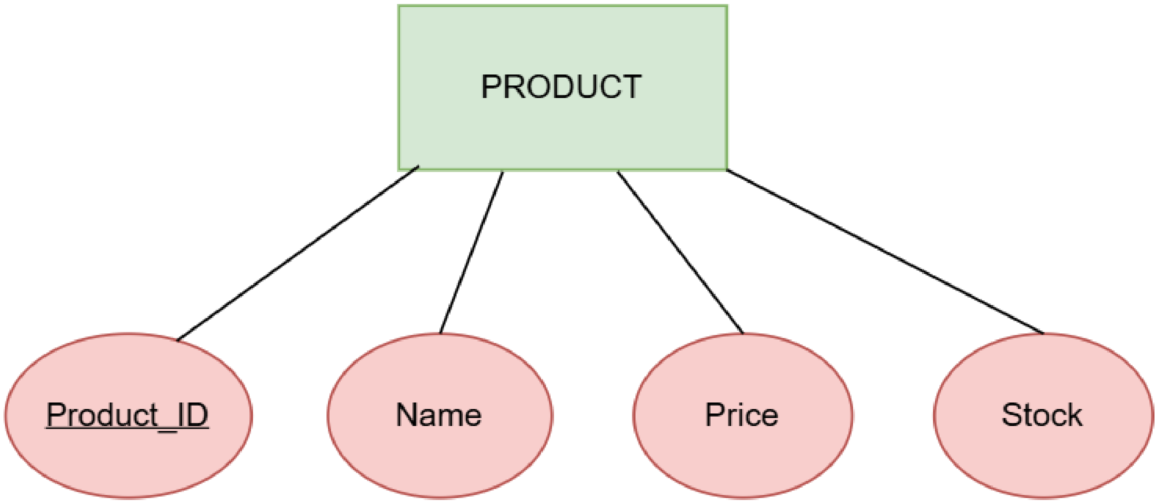
Create a new folder named **Online-Shopping-GUI** and open it in VS Code. Inside, maintain the following structure:



```
ONLINE-SHOPPING-GUI
  src
    main
      java\com\mycompany\onlinesh...
        App.java
        Cart.java                  1
        DBHelper.java              7
        MainWindow.java            6
        Product.java
        ProductGUI.java            2
        TestDBConnection.java      6
        TestMain.java              1
      resources
        db.properties
    test\java\com\mycompany\onlineshop
      AppTest.java
  target
    classes
    generated-sources
    maven-status
    test-classes
  pom.xml
```

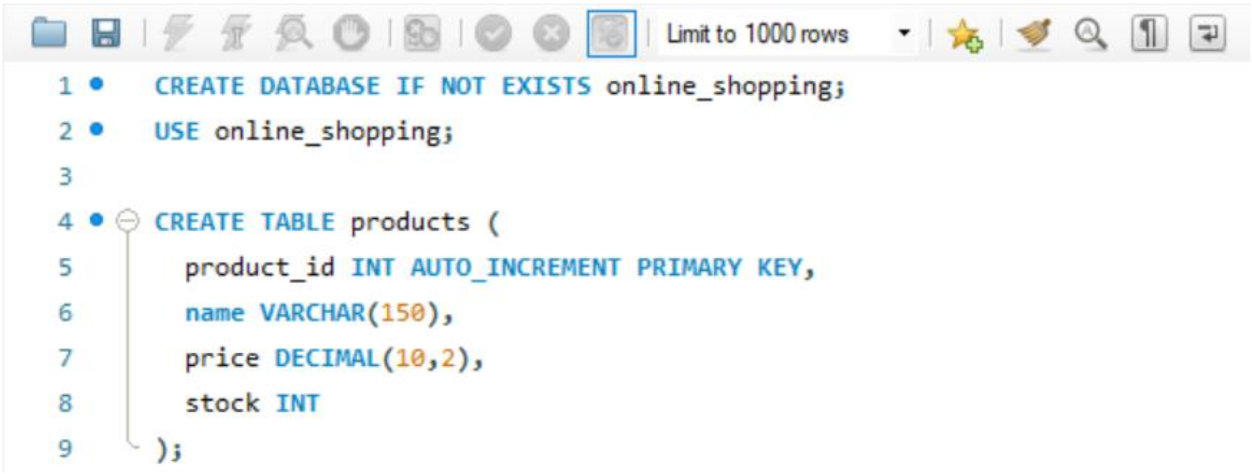
Database Design

ER Diagram



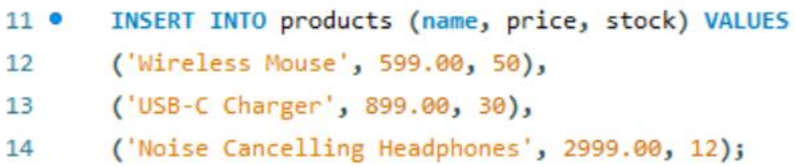
Schema Definition

Database Name: online_shopping
Table Name: products



```
CREATE DATABASE IF NOT EXISTS online_shopping;
USE online_shopping;

CREATE TABLE products (
  product_id INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(150),
  price DECIMAL(10,2),
  stock INT
);
```

Insert Sample Data



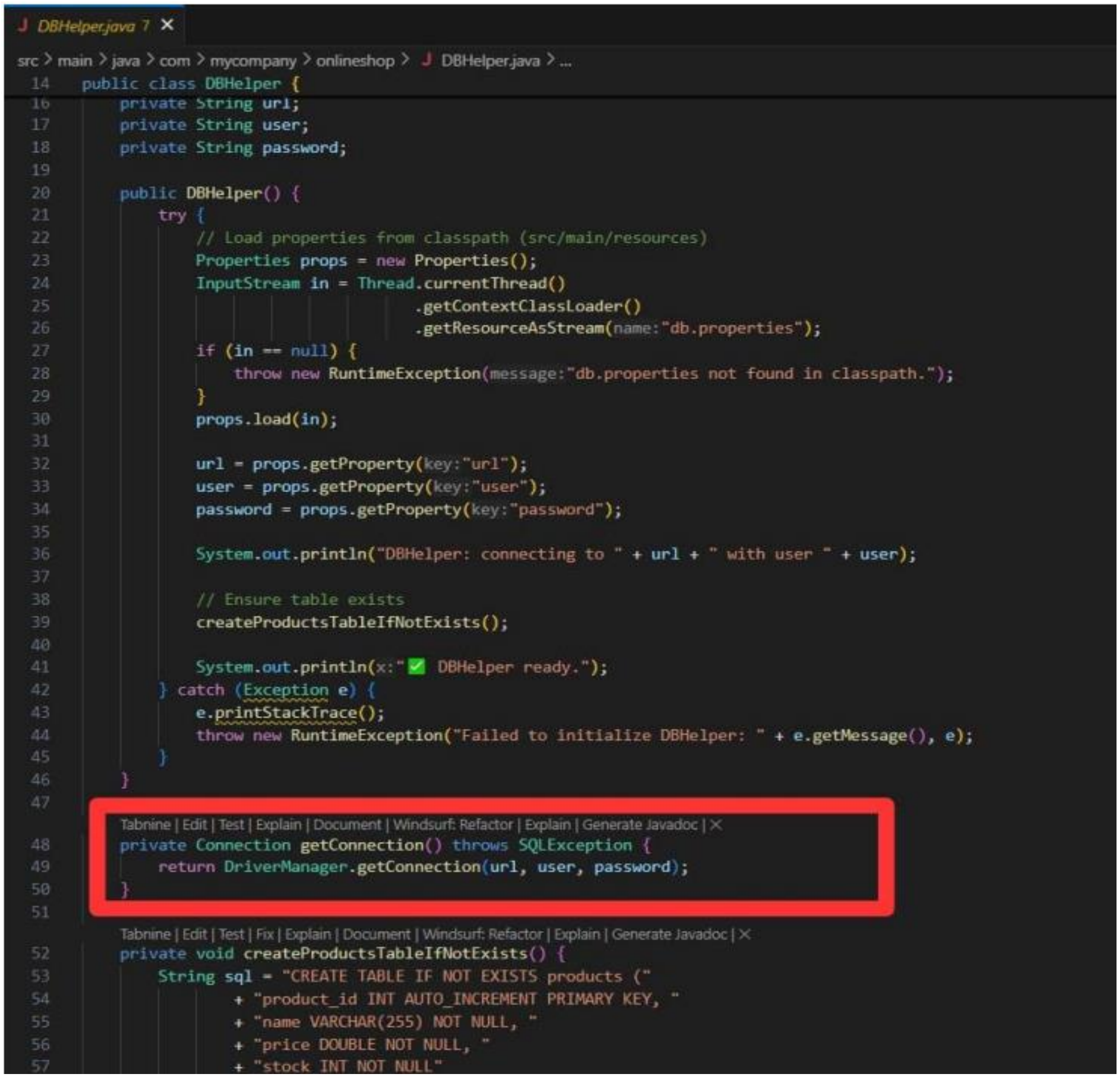
```sql
INSERT INTO products (name, price, stock) VALUES
('Wireless Mouse', 599.00, 50),
('USB-C Charger', 899.00, 30),
('Noise Cancelling Headphones', 2999.00, 12);
```

Implementation (Step-by-Step)

1. DBHelper.java

Handles database connection and CRUD operations.

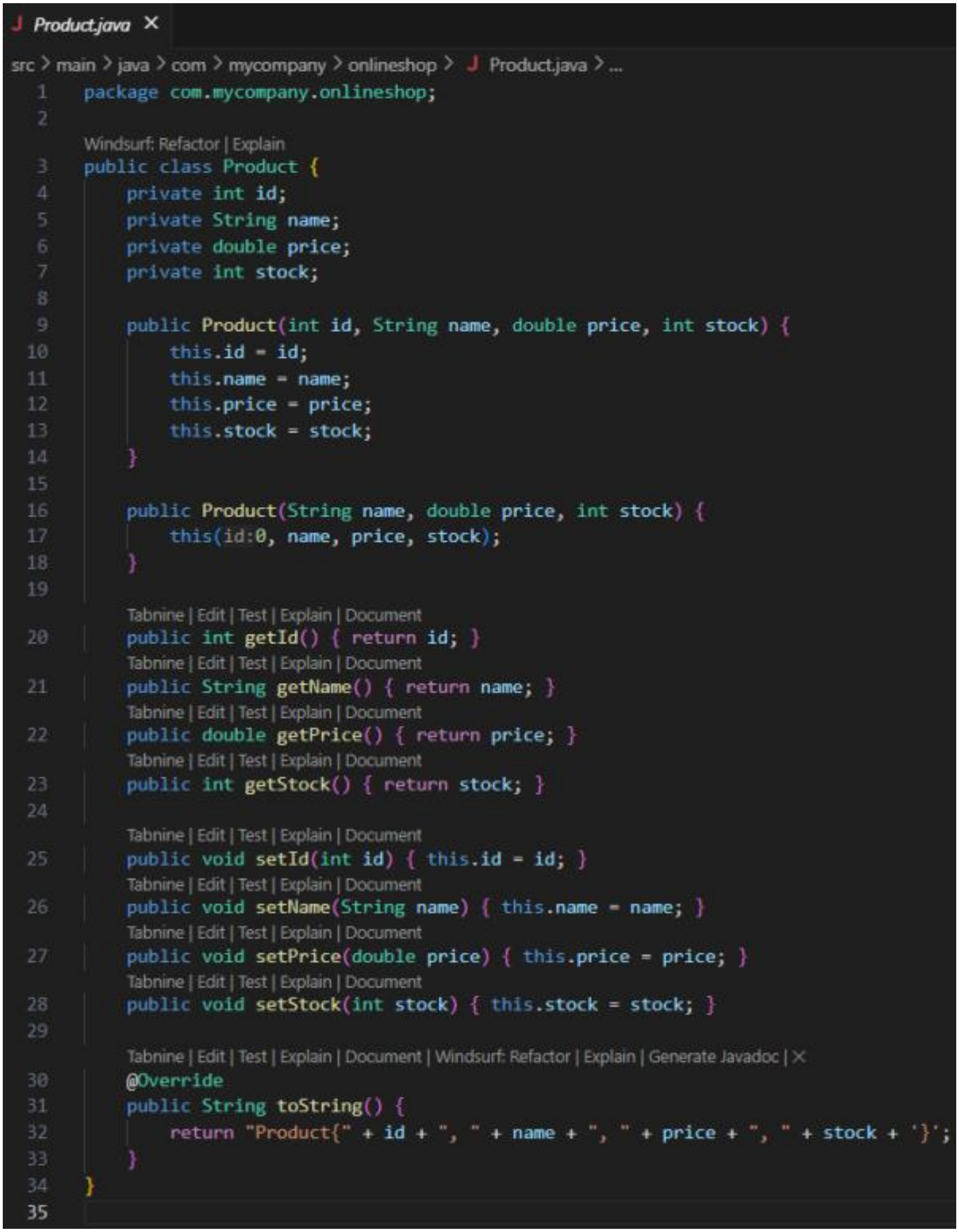
- Establishes JDBC connection using db.properties.
- Provides methods like getAllProducts(), addProduct(), updateProduct(), and deleteProduct().


DBHelper.java 7 ×
src > main > java > com > mycompany > onlineshop > DBHelper.java > ...
Tabnine | Edit | Test | Explain | Document | Windsurf: Refactor | Explain | Generate Javadoc | ×
Tabnine | Edit | Test | Fix | Explain | Document | Windsurf: Refactor | Explain | Generate Javadoc | ×

```java
public class DBHelper {
    private String url;
    private String user;
    private String password;

    public DBHelper() {
        try {
            // Load properties from classpath (src/main/resources)
            Properties props = new Properties();
            InputStream in = Thread.currentThread()
                                   .getContextClassLoader()
                                   .getResourceAsStream(name:"db.properties");
            if (in == null) {
                throw new RuntimeException(message:"db.properties not found in classpath.");
            }
            props.load(in);

            url = props.getProperty(key:"url");
            user = props.getProperty(key:"user");
            password = props.getProperty(key:"password");

            System.out.println("DBHelper: connecting to " + url + " with user " + user);

            // Ensure table exists
            createProductsTableIfNotExists();

            System.out.println(x:"✅ DBHelper ready.");
        } catch (Exception e) {
            e.printStackTrace();
            throw new RuntimeException("Failed to initialize DBHelper: " + e.getMessage(), e);
        }
    }

    private Connection getConnection() throws SQLException {
        return DriverManager.getConnection(url, user, password);
    }

    private void createProductsTableIfNotExists() {
        String sql = "CREATE TABLE IF NOT EXISTS products ("
                + "product_id INT AUTO_INCREMENT PRIMARY KEY, "
                + "name VARCHAR(255) NOT NULL, "
                + "price DOUBLE NOT NULL, "
                + "stock INT NOT NULL"
```

2. Product.java

Model class representing a single product entity (id, name, price, stock).

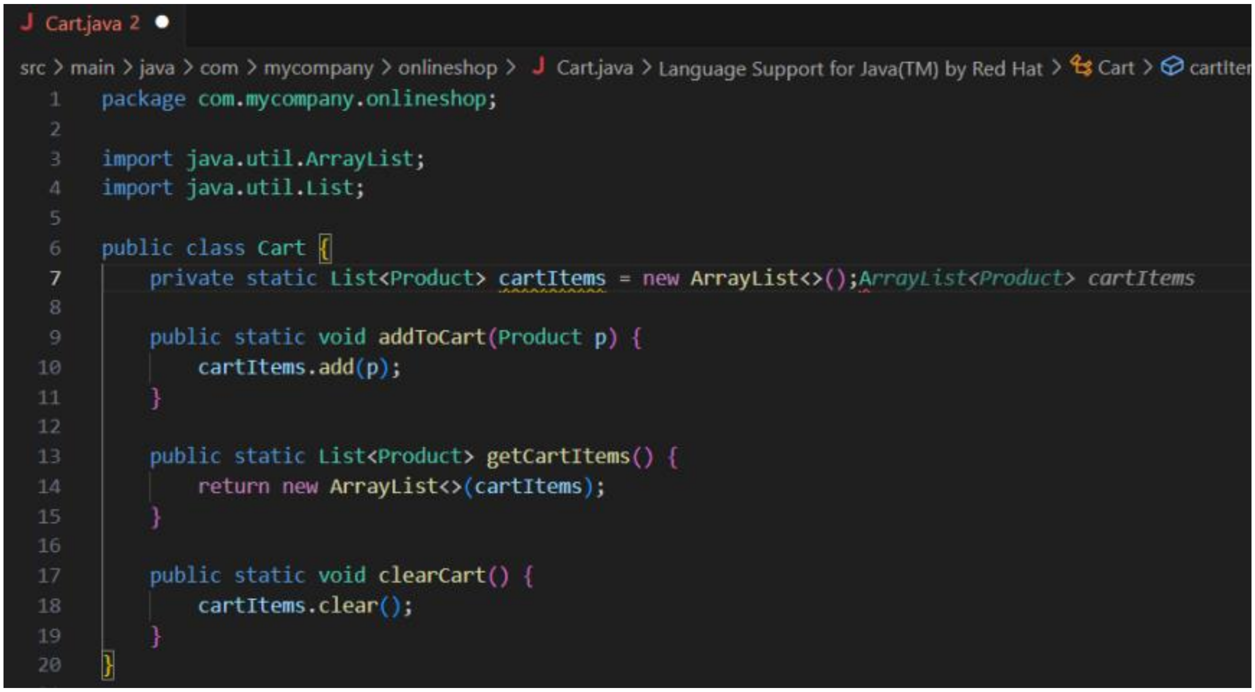

Product.java ×
src > main > java > com > mycompany > onlineshop > Product.java > ...
Windsurf: Refactor | Explain
Tabnine | Edit | Test | Explain | Document
Tabnine | Edit | Test | Explain | Document | Windsurf: Refactor | Explain | Generate Javadoc | ×

```java
package com.mycompany.onlineshop;

public class Product {
    private int id;
    private String name;
    private double price;
    private int stock;

    public Product(int id, String name, double price, int stock) {
        this.id = id;
        this.name = name;
        this.price = price;
        this.stock = stock;
    }

    public Product(String name, double price, int stock) {
        this(id:0, name, price, stock);
    }

    public int getId() { return id; }
    public String getName() { return name; }
    public double getPrice() { return price; }
    public int getStock() { return stock; }

    public void setId(int id) { this.id = id; }
    public void setName(String name) { this.name = name; }
    public void setPrice(double price) { this.price = price; }
    public void setStock(int stock) { this.stock = stock; }

    @Override
    public String toString() {
        return "Product{" + id + ", " + name + ", " + price + ", " + stock + '}';
    }
}

```

3. Cart.java

Handles cart operations:

- Add items
- View cart list
- Calculate total

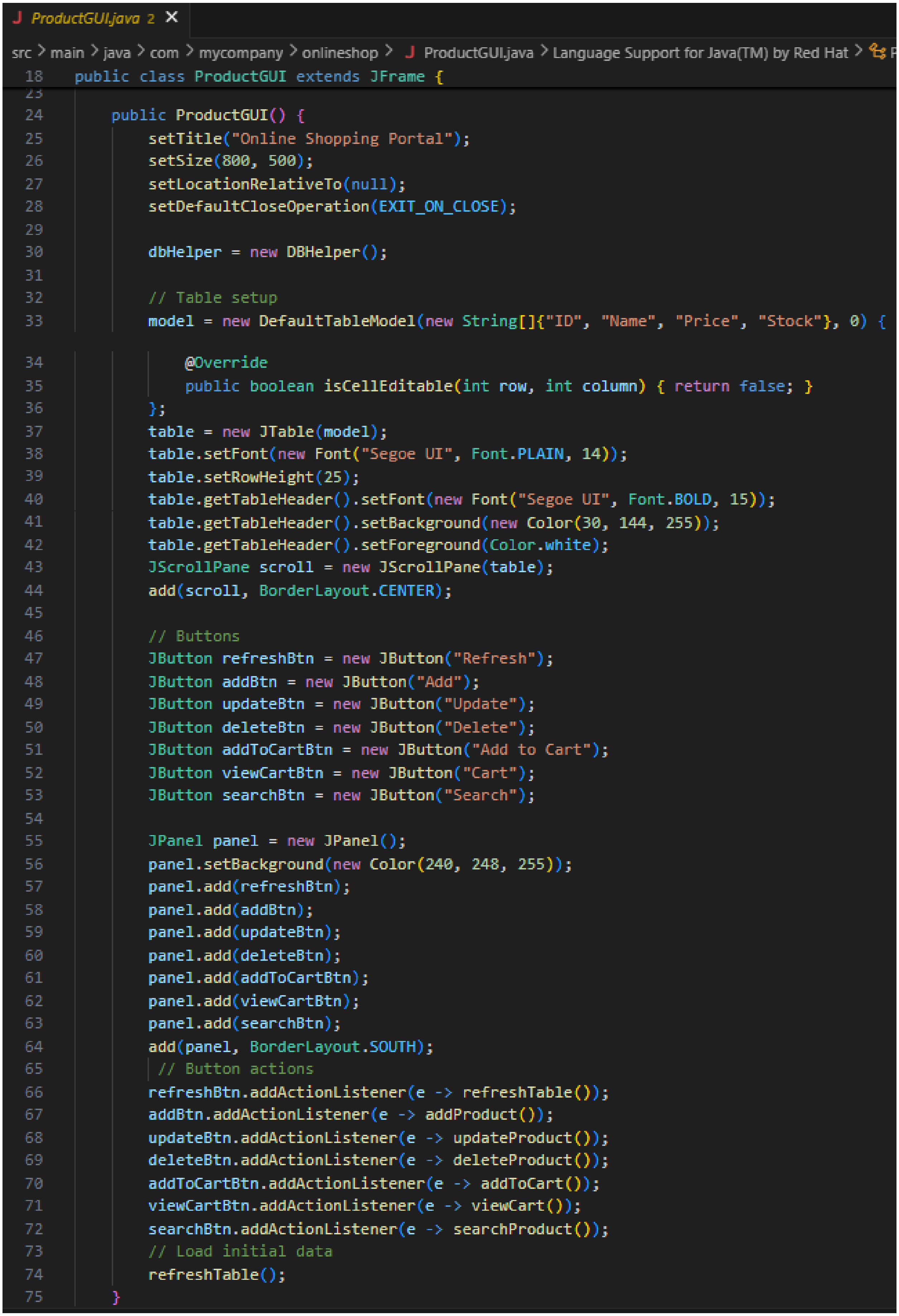

Cart.java 2
src > main > java > com > mycompany > onlineshop > Cart.java > Language Support for Java(TM) by Red Hat > Cart > cartIter

```java
package com.mycompany.onlineshop;

import java.util.ArrayList;
import java.util.List;

public class Cart {
    private static List<Product> cartItems = new ArrayList<>();ArrayList<Product> cartItems

    public static void addToCart(Product p) {
        cartItems.add(p);
    }

    public static List<Product> getCartItems() {
        return new ArrayList<>(cartItems);
    }

    public static void clearCart() {
        cartItems.clear();
    }
}
```

4. ProductGUI.java

Core GUI of the system built using Swing.

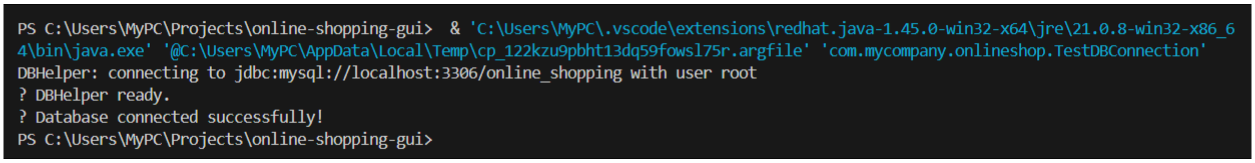
- Displays product table.
- Provides CRUD buttons (Add, Update, Delete, Refresh).
- Includes **Search**, **Add to Cart**, and **View Cart** options.


ProductGUI.java 2
src > main > java > com > mycompany > onlineshop > ProductGUI.java > Language Support for Java(TM) by Red Hat > P

```java
public class ProductGUI extends JFrame {

    public ProductGUI() {
        setTitle("Online Shopping Portal");
        setSize(800, 500);
        setLocationRelativeTo(null);
        setDefaultCloseOperation(EXIT_ON_CLOSE);

        dbHelper = new DBHelper();

        // Table setup
        model = new DefaultTableModel(new String[]{"ID", "Name", "Price", "Stock"}, 0) {
            @Override
            public boolean isCellEditable(int row, int column) { return false; }
        };
        table = new JTable(model);
        table.setFont(new Font("Segoe UI", Font.PLAIN, 14));
        table.setRowHeight(25);
        table.getTableHeader().setFont(new Font("Segoe UI", Font.BOLD, 15));
        table.getTableHeader().setBackground(new Color(30, 144, 255));
        table.getTableHeader().setForeground(Color.white);
        JScrollPane scroll = new JScrollPane(table);
        add(scroll, BorderLayout.CENTER);

        // Buttons
        JButton refreshBtn = new JButton("Refresh");
        JButton addBtn = new JButton("Add");
        JButton updateBtn = new JButton("Update");
        JButton deleteBtn = new JButton("Delete");
        JButton addToCartBtn = new JButton("Add to Cart");
        JButton viewCartBtn = new JButton("Cart");
        JButton searchBtn = new JButton("Search");

        JPanel panel = new JPanel();
        panel.setBackground(new Color(240, 248, 255));
        panel.add(refreshBtn);
        panel.add(addBtn);
        panel.add(updateBtn);
        panel.add(deleteBtn);
        panel.add(addToCartBtn);
        panel.add(viewCartBtn);
        panel.add(searchBtn);
        add(panel, BorderLayout.SOUTH);
         // Button actions
        refreshBtn.addActionListener(e -> refreshTable());
        addBtn.addActionListener(e -> addProduct());
        updateBtn.addActionListener(e -> updateProduct());
        deleteBtn.addActionListener(e -> deleteProduct());
        addToCartBtn.addActionListener(e -> addToCart());
        viewCartBtn.addActionListener(e -> viewCart());
        searchBtn.addActionListener(e -> searchProduct());
        // Load initial data
        refreshTable();
    }
```

5. TestDBConnection.java

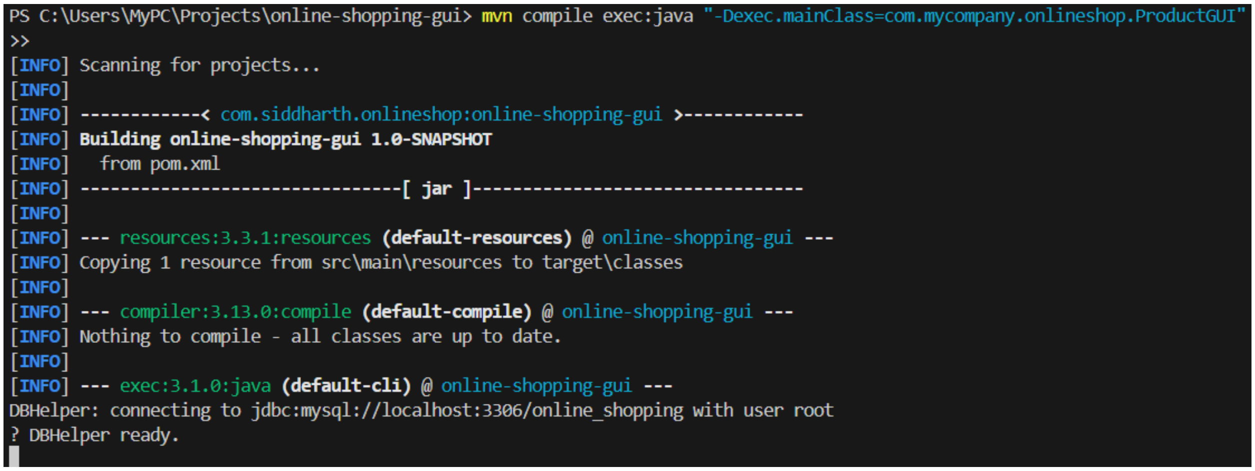
Verifies successful MySQL connectivity.



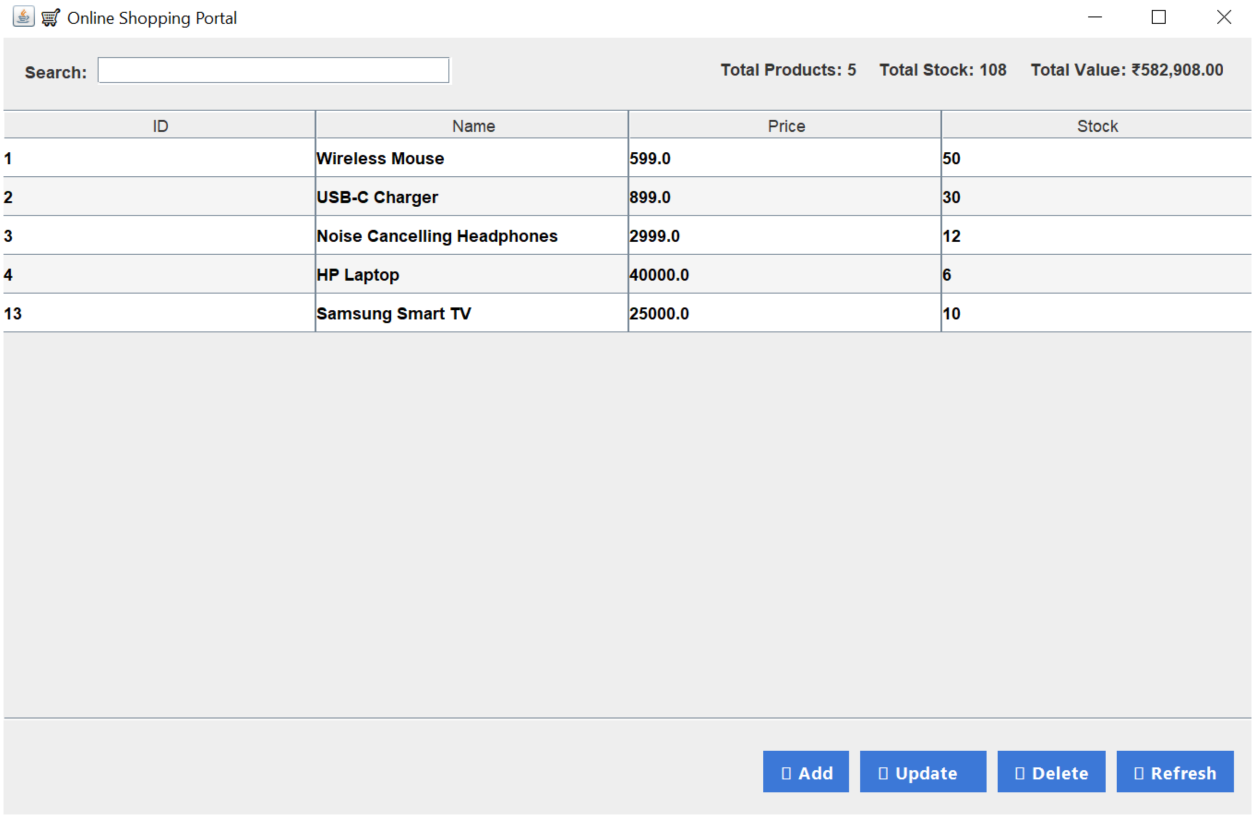
```
PS C:\Users\MyPC\Projects\online-shopping-gui>  & 'C:\Users\MyPC\.vscode\extensions\redhat.java-1.45.0-win32-x64\jre\21.0.8-win32-x86_6
4\bin\java.exe' '@C:\Users\MyPC\AppData\Local\Temp\cp_122kzu9pbht13dq59fowsl75r.argfile' 'com.mycompany.onlineshop.TestDBConnection'
DBHelper: connecting to jdbc:mysql://localhost:3306/online_shopping with user root
? DBHelper ready.
? Database connected successfully!
PS C:\Users\MyPC\Projects\online-shopping-gui>
```

6. TestMain.java

Used to launch and test the entire application.
Ensures all dependencies load correctly before running GUI.



```
PS C:\Users\MyPC\Projects\online-shopping-gui> mvn compile exec:java "-Dexec.mainClass=com.mycompany.onlineshop.ProductGUI"
>>
[INFO] Scanning for projects...
[INFO]
[INFO] -----------< com.siddharth.onlineshop:online-shopping-gui >------------
[INFO] Building online-shopping-gui 1.0-SNAPSHOT
[INFO]   from pom.xml
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ online-shopping-gui ---
[INFO] Copying 1 resource from src\main\resources to target\classes
[INFO]
[INFO] --- compiler:3.13.0:compile (default-compile) @ online-shopping-gui ---
[INFO] Nothing to compile - all classes are up to date.
[INFO]
[INFO] --- exec:3.1.0:java (default-cli) @ online-shopping-gui ---
DBHelper: connecting to jdbc:mysql://localhost:3306/online_shopping with user root
? DBHelper ready.
```

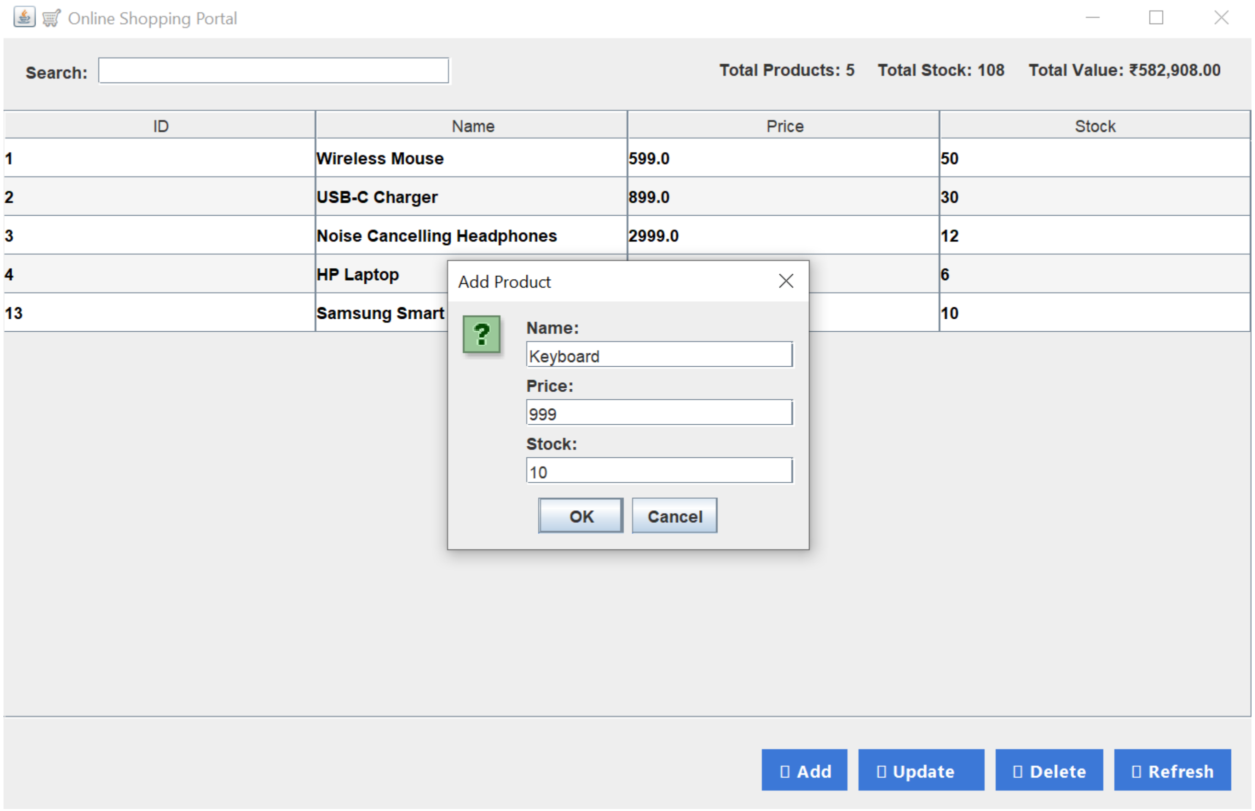


Total Products: 5 Total Stock: 108 Total Value: ₹582,908.00

ID	Name	Price	Stock
1	Wireless Mouse	599.0	50
2	USB-C Charger	899.0	30
3	Noise Cancelling Headphones	2999.0	12
4	HP Laptop	40000.0	6
13	Samsung Smart TV	25000.0	10

Testing & Results:

Test Case	Input	Expected Output	Result
Add Product	“Keyboard”, 999, 10	Product added successfully	✓
Update Product	Change price of “Mouse”	Product updated successfully	✓
Delete Product	Remove “USB-C Charger”	Product deleted	✓



Search:

Total Products: 5 Total Stock: 108 Total Value: ₹582,908.00

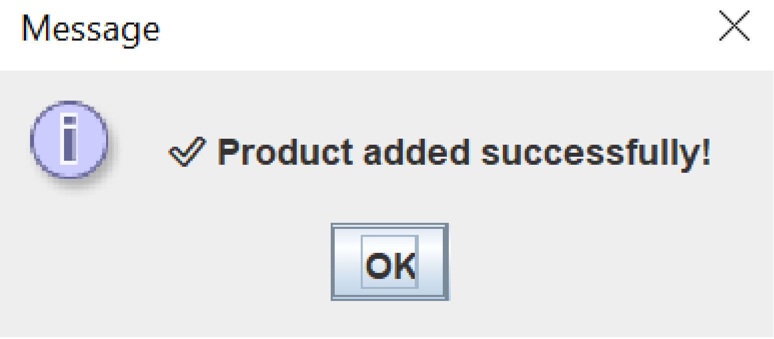
ID	Name	Price	Stock
1	Wireless Mouse	599.0	50
2	USB-C Charger	899.0	30
3	Noise Cancelling Headphones	2999.0	12
4	HP Laptop		6
13	Samsung Smart		10

Add Product

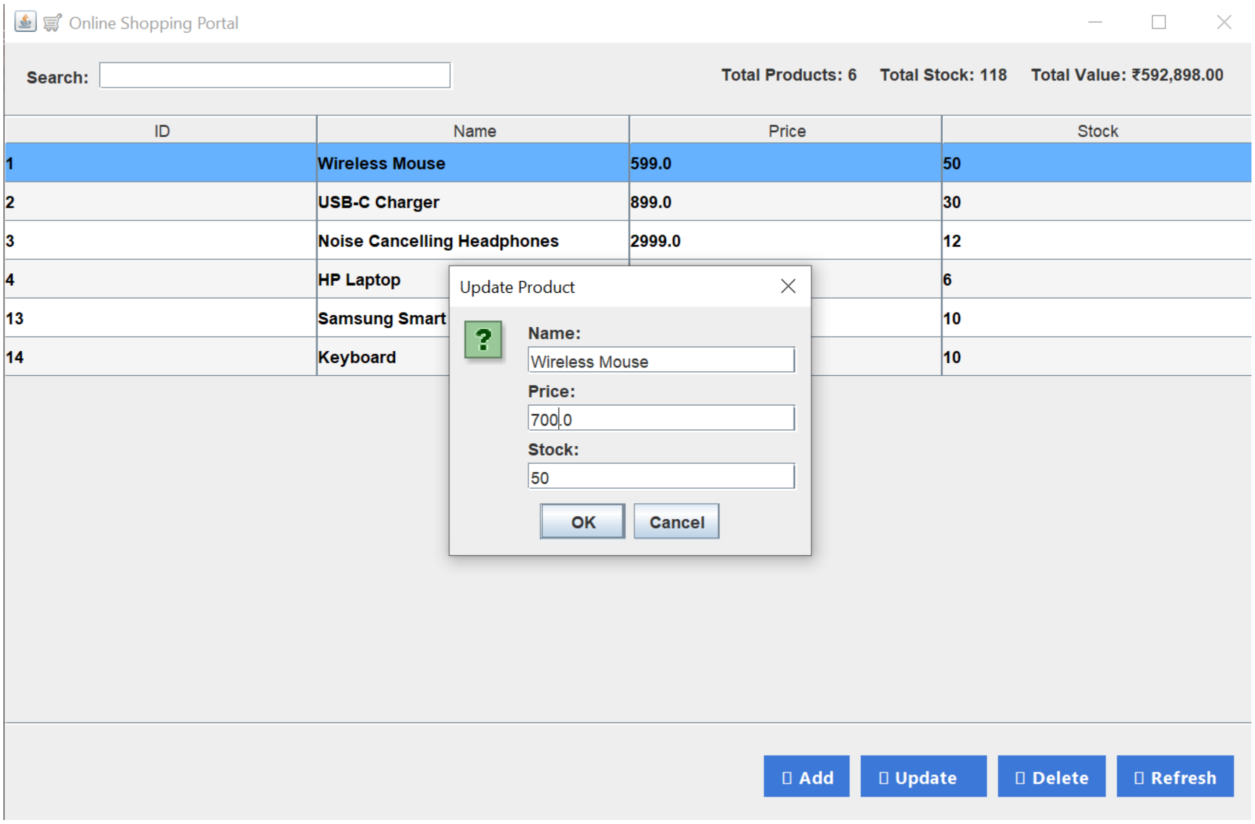
Name: Keyboard

Price: 999

Stock: 10



✓ Product added successfully!



Online Shopping Portal

Search:

Total Products: 6 Total Stock: 118 Total Value: ₹592,898.00

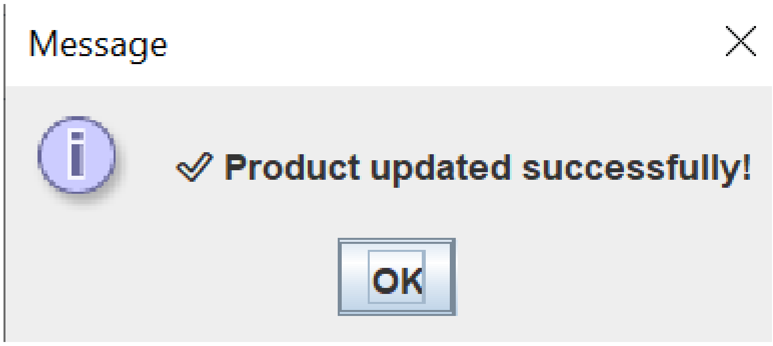
ID	Name	Price	Stock
1	Wireless Mouse	599.0	50
2	USB-C Charger	899.0	30
3	Noise Cancelling Headphones	2999.0	12
4	HP Laptop		6
13	Samsung Smart		10
14	Keyboard		10

Update Product

Name: Wireless Mouse

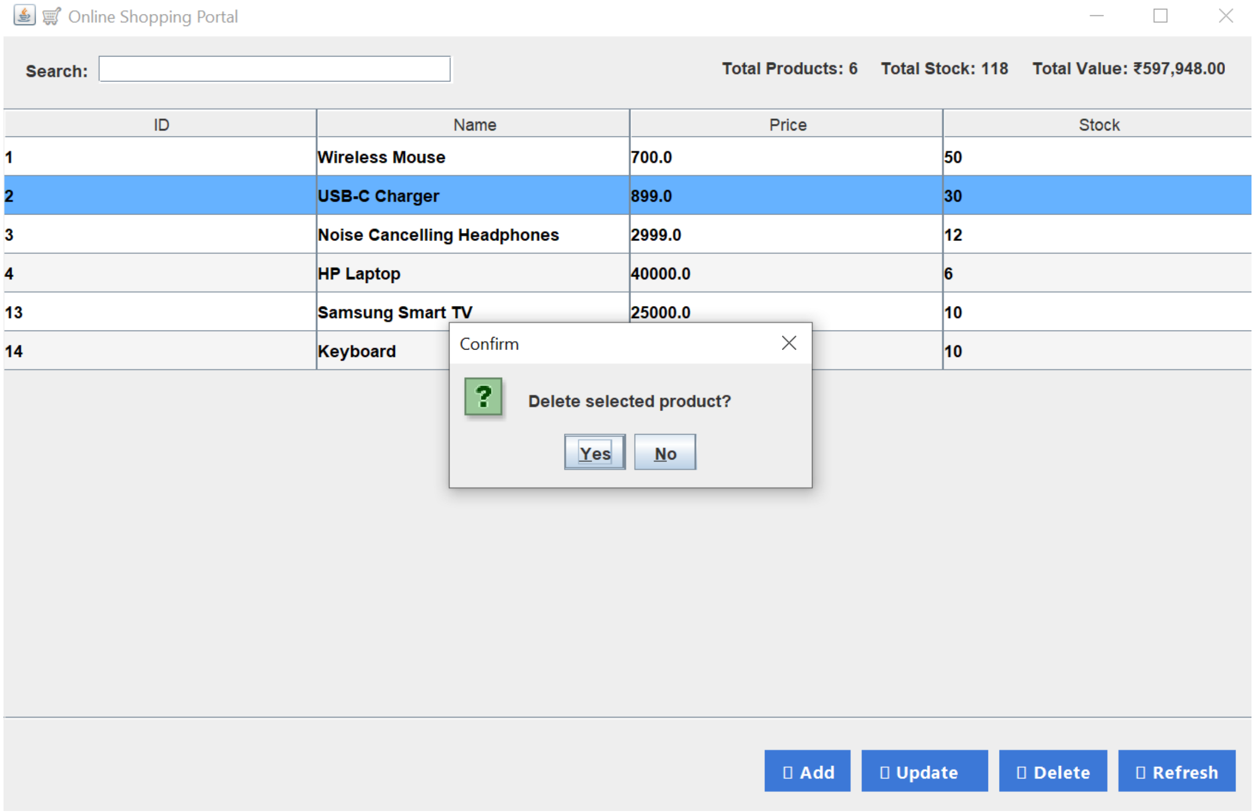
Price: 700.0

Stock: 50



Message

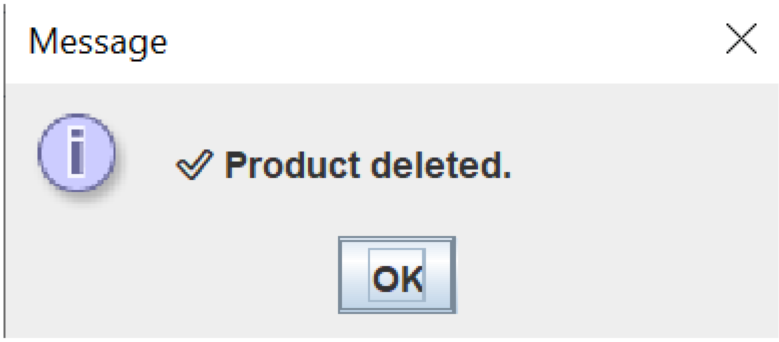
✓ Product updated successfully!



Total Products: 6   Total Stock: 118   Total Value: ₹597,948.00

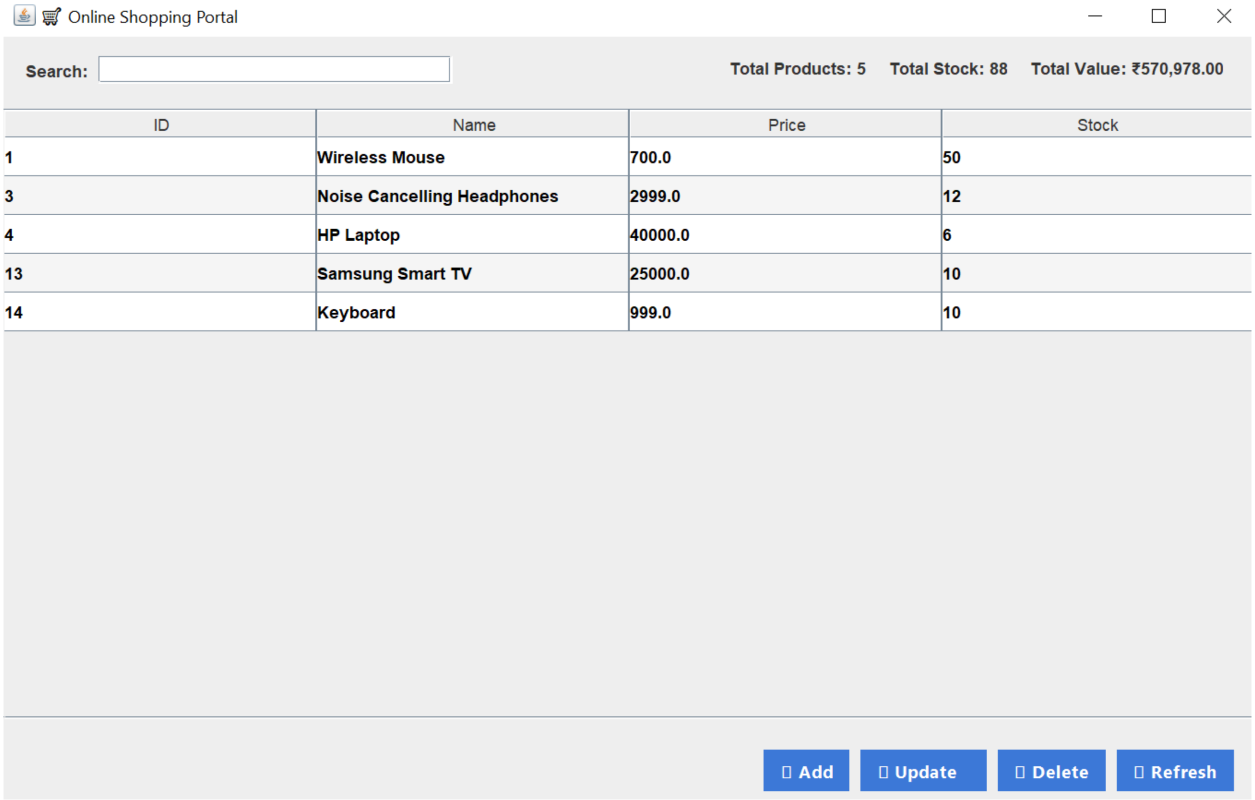
ID	Name	Price	Stock
1	Wireless Mouse	700.0	50
2	USB-C Charger	899.0	30
3	Noise Cancelling Headphones	2999.0	12
4	HP Laptop	40000.0	6
13	Samsung Smart TV	25000.0	10
14	Keyboard		10

Delete selected product?



✓ Product deleted.

Final view after all the operations :



Total Products: 5 Total Stock: 88 Total Value: ₹570,978.00

ID	Name	Price	Stock
1	Wireless Mouse	700.0	50
3	Noise Cancelling Headphones	2999.0	12
4	HP Laptop	40000.0	6
13	Samsung Smart TV	25000.0	10
14	Keyboard	999.0	10

Troubleshooting / Common Errors :

Error	Cause	Solution	Example Terminal Output (Before Fix)	Example Terminal Output (After Fix)
Communications link failure	MySQL service not running	Start MySQL service before running the app	Communications link failure Cannot connect to database at localhost:3306	No error, program connects successfully to DB
Access denied for user 'root'@'localhost'	Wrong username/password in db.properties	Correct the username/password in db.properties	Access denied for user 'root'@'localhost' (using password: YES)	Program connects successfully to DB, GUI opens
ClassNotFoundException:	MySQL connector	Add MySQL connector	Exception in thread "main"	Program connects

com.mysql.cj.jdbc.Driver	not added	dependency in pom.xml or include jar in classpath	java.lang.ClassNotFoundException: com.mysql.cj.jdbc.Driver	successfully to DB, no ClassNotFoundException
GUI not opening	Wrong main class run	Run ProductGUI.java main class	Nothing appears, or Exception in thread "main"...	GUI opens showing product table and buttons

Project Demo Instructions

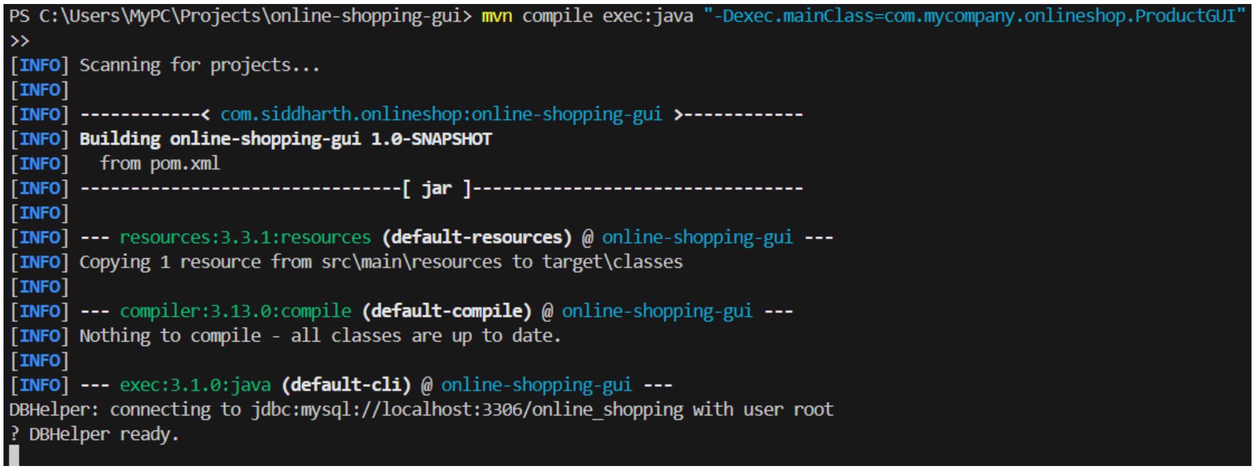
1. Start MySQL server and open Workbench.
2. Open VS Code → Load project folder.
3. Run these commands:

mvn clean compile

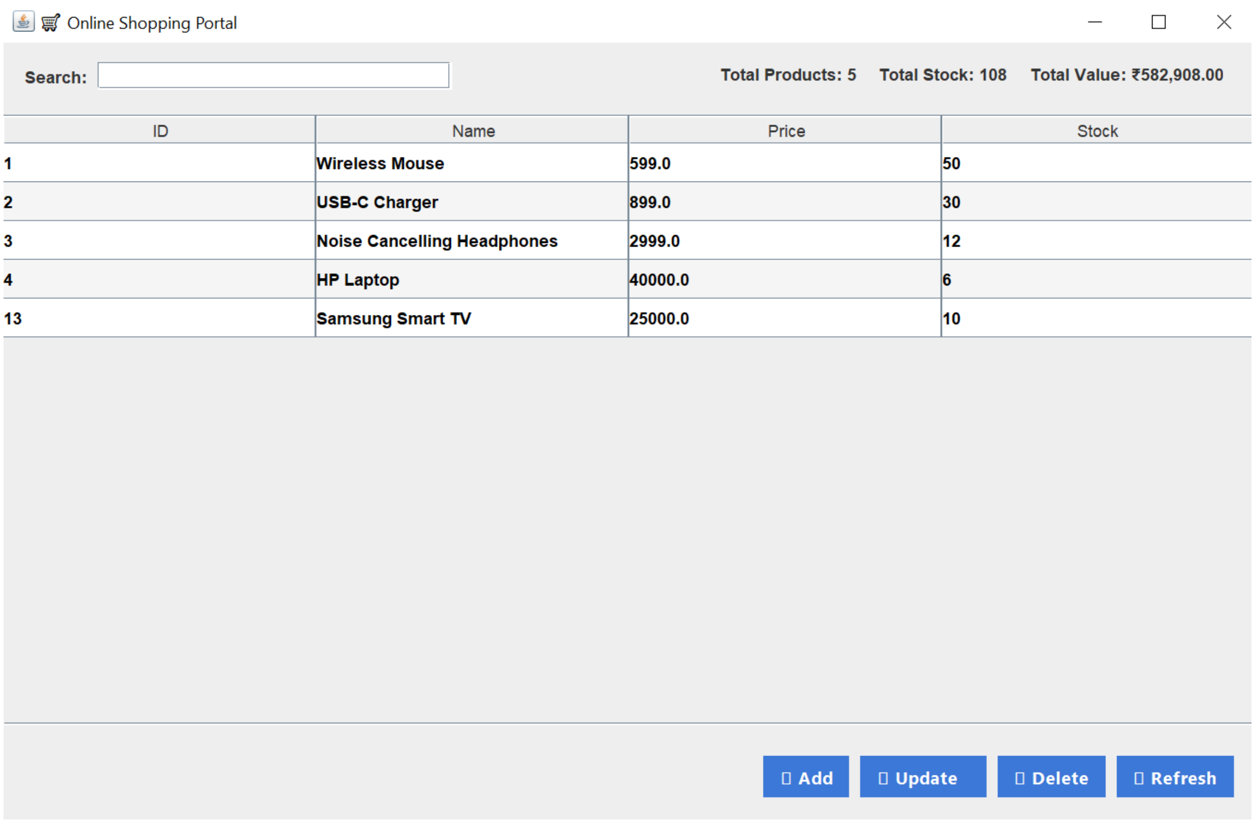
mvn exec:java -Dexec.mainClass=com.mycompany.onlineshop.ProductGUI

4. GUI window will open showing all products.

5. Test all CRUD and cart functionalities.



```
PS C:\Users\MyPC\Projects\online-shopping-gui> mvn compile exec:java "-Dexec.mainClass=com.mycompany.onlineshop.ProductGUI"
>>
[INFO] Scanning for projects...
[INFO]
[INFO] ------------< com.siddharth.onlineshop:online-shopping-gui >------------
[INFO] Building online-shopping-gui 1.0-SNAPSHOT
[INFO]   from pom.xml
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ online-shopping-gui ---
[INFO] Copying 1 resource from src\main\resources to target\classes
[INFO]
[INFO] --- compiler:3.13.0:compile (default-compile) @ online-shopping-gui ---
[INFO] Nothing to compile - all classes are up to date.
[INFO]
[INFO] --- exec:3.1.0:java (default-cli) @ online-shopping-gui ---
DBHelper: connecting to jdbc:mysql://localhost:3306/online_shopping with user root
? DBHelper ready.
```



Total Products: 5 Total Stock: 108 Total Value: ₹582,908.00

ID	Name	Price	Stock
1	Wireless Mouse	599.0	50
2	USB-C Charger	899.0	30
3	Noise Cancelling Headphones	2999.0	12
4	HP Laptop	40000.0	6
13	Samsung Smart TV	25000.0	10

Conclusion:

Through this project, I successfully implemented a **Java Swing-based GUI** that connects to a **MySQL database** using **JDBC**.
I learned how to perform CRUD operations, establish a stable database connection, and design a simple yet functional interface for product management.
This mini project enhanced my understanding of **database connectivity**, **front-end integration**, and **real-world Java application development**.