Inventory Management System with MySQL (C++)

Welcome to the \*\*Inventory Management System\*\* project! This application integrates with a \*\*MySQL database\*\* to manage an inventory system where you can \*\*add, view, search, and delete products\*\*. Built using \*\*C++\*\*, it connects to MySQL for persistent data storage.

---

# 🛠️ Prerequisites

Before starting, make sure you have the following software installed:

## 1. \*\*MySQL Server\*\*

MySQL is required to store and manage your product data. - \*\*Download MySQL Server\*\*: [Download MySQL Server](https://dev.mysql.com/downloads/installer/)

## 2. \*\*C++ Compiler (MinGW or GCC)\*\*

To compile and run the C++ code, you'll need a C++ compiler. - \*\*Download MinGW\*\*: [Download MinGW](https://sourceforge.net/projects/mingw/) - \*\*Download GCC for Windows\*\*: [Download GCC](https://gcc.gnu.org/)

## 3. \*\*Visual Studio Code (VS Code)\*\*

Visual Studio Code is an IDE that makes writing and compiling C++ code easy. - \*\*Download VS Code\*\*: [Download VS Code](https://code.visualstudio.com/)

\*\*Necessary VS Code Extensions\*\*:

1. \*\*C/C++ Extension\*\*: By Microsoft for C++ support.

2. \*\*MySQL (optional)\*\*: For managing MySQL databases inside VS Code.

## 4. \*\*MySQL Connector for C++\*\*

You'll need the \*\*MySQL Connector/C++\*\* to connect C++ with MySQL. - \*\*Download MySQL Connector\*\*: [Download MySQL Connector](https://dev.mysql.com/downloads/connector/cpp/)

---

# 🚀 Step-by-Step Guide

## Step 1: Install MySQL

1. \*\*Download and Install MySQL Server\*\* from the [official MySQL website](https://dev.mysql.com/downloads/installer/).

2. \*\*Create the `inventory` Database\*\*:

```sql  
CREATE DATABASE inventory;  
```

\*\*Create the products Table\*\*:

```sql  
CREATE TABLE products (  
 id INT AUTO\_INCREMENT PRIMARY KEY,  
 name VARCHAR(255) NOT NULL,  
 quantity INT NOT NULL,  
 price DOUBLE NOT NULL  
);  
```

## Step 2: Install and Set Up MinGW or GCC

Download MinGW from the [MinGW SourceForge page](https://sourceforge.net/projects/mingw/).

Install it and add the `bin` directory to your system PATH so that `g++` is accessible from the command line.

## Step 3: Install Visual Studio Code (VS Code)

Download and Install VS Code from [here](https://code.visualstudio.com/).

Install the \*\*C/C++ Extension\*\* by Microsoft to provide support for C++ development.

## Step 4: Install MySQL Connector/C++

Download \*\*MySQL Connector for C++\*\* from the official MySQL website.

Make sure the include and lib directories are correctly set in your project for proper linking.

## Step 5: Configure VS Code for Building the Project

### 1. Include Path Configuration

To set up the MySQL Connector include path, modify the `c\_cpp\_properties.json` file in VS Code:

```json  
"includePath": [  
 "${workspaceFolder}/\*\*",  
 "C:/Program Files/MySQL/MySQL Server 8.0/include"  
];  
```

### 2. Library Path Configuration

Configure the library path in the `tasks.json` file in VS Code:

```json  
{  
 "version": "2.0.0",  
 "tasks": [  
 {  
 "label": "build",  
 "type": "shell",  
 "command": "g++",  
 "args": [  
 "-g",  
 "inventory.cpp",  
 "-o",  
 "inventory",  
 "-I\"C:/Program Files/MySQL/MySQL Server 8.0/include\"",  
 "-L\"C:/Program Files/MySQL/MySQL Server 8.0/lib\"",  
 "-lmysqlclient"  
 ],  
 "group": {  
 "kind": "build",  
 "isDefault": true  
 },  
 "problemMatcher": ["$gcc"],  
 "detail": "Generated task by VS Code"  
 }  
 ]  
}  
```

---

# 💡 Common Issues and Solutions

## Issue 1: `g++: 'mysql.h' Not Found`

\*\*Solution\*\*:

The error happens when the MySQL Connector include path is not properly set.

Ensure the include path is correctly added to `c\_cpp\_properties.json`.

Confirm that `mysql.h` exists in the `C:/Program Files/MySQL/MySQL Server 8.0/include` directory.

## Issue 2: Linking Error: `libmysqlclient Not Found`

\*\*Solution\*\*:

Ensure the library path is correctly set in the `tasks.json` file.

Verify that `libmysqlclient` exists in the lib directory (`C:/Program Files/MySQL/MySQL Server 8.0/lib`).

Add the correct path to the library in `tasks.json`.

## Issue 3: Table Does Not Exist

\*\*Solution\*\*:

The table must be created before interacting with the inventory.

Run the SQL script to create the inventory database and products table:

```sql  
CREATE DATABASE inventory;  
CREATE TABLE products (  
 id INT AUTO\_INCREMENT PRIMARY KEY,  
 name VARCHAR(255) NOT NULL,  
 quantity INT NOT NULL,  
 price DOUBLE NOT NULL  
);  
```

## Issue 4: Failed to Connect to MySQL Database

\*\*Solution\*\*:

Ensure MySQL is installed and running.

Verify your MySQL username and password are correct in the C++ code.

Make sure MySQL is running on port 3306.

Ensure you are using `localhost` or `127.0.0.1` as the hostname.

# 🏁 Running the Program

Once everything is set up, you can build and run the project:

1. Build the project by pressing \*\*Ctrl + Shift + B\*\* in VS Code.

2. Run the application in the terminal:

```bash  
./inventory  
```

The menu will appear with the following options:

1. \*\*Add Product\*\*: Add a new product to the database.

2. \*\*View Products\*\*: View all products in the inventory.

3. \*\*Search Product\*\*: Search for a product by ID.

4. \*\*Delete Product\*\*: Delete a product by ID.

5. \*\*Exit\*\*: Exit the program.

---

# 📄 License

This project is licensed under the MIT License - see the LICENSE file for details.