**🧴 MySkinKare – Console-Based Skincare Shopping App (C++)**

**MySkinKare** is a modular, console-based shopping application built in **C++**, designed to provide a simple yet functional e-commerce experience focused on skincare products. Developed with the assistance of **Claude Sonnet 3.5** and **GitHub Copilot** in **Visual Studio Code**, the application emphasizes clean architecture, user-friendly interaction, and essential shopping features.

**📦 Project Structure**

MySkinKare/

├── src

│ ├── main.cpp // Entry point and main menu logic

│ ├── database/

│ │ ├── Database.cpp // File-based data handling

│ │ └── Database.h

│ ├── auth/

│ │ ├── User.cpp // User registration and login

│ │ └── User.h

│ ├── products/

│ │ ├── Product.cpp // Product catalog and stock logic

│ │ └── Product.h

│ └── utils/

│ ├── FileHandler.cpp // Utility functions for file I/O

│ └── FileHandler.h

├── include/

│ └── headers.h // Centralized header includes

├── tests/

│ └── unit\_tests.cpp // Basic unit tests

├── CMakeLists.txt // Build configuration

└── README.md // Project documentation

**🔧 Key Features**

**👤 User Authentication**

* **Register/Login** system with input validation.
* Credentials stored in users.txt using file I/O.
* Prevents duplicate usernames and ensures secure access.

**🛍️ Product Catalog**

* Hardcoded skincare products with names, descriptions, prices, and stock.
* Displayed in a clean, console-friendly format.
* Easily extendable to support external product files.

**🛒 Shopping Cart**

* Add/remove/view products in the cart.
* Calculates total cost dynamically.
* Validates stock availability before adding items.

**📦 Stock Management**

* Tracks product inventory.
* Updates stock levels as users interact with the cart.

**💾 Persistent Data Storage**

* User data stored in users.txt.
* Product data currently hardcoded but structured for future file/database integration.

**🖥️ Console-Based Interface**

* Menu-driven navigation with clear prompts.
* Input validation throughout to ensure smooth user experience.

**🛠️ Development Tools**

* **Language**: C++
* **IDE**: Visual Studio Code
* **AI Assistant**: Claude Sonnet 3.5
* **Code Completion**: GitHub Copilot
* **Build System**: CMake

**✅ Platform Compatibility**

* Fully tested and **compiles/runs without issues on Windows**.
* Uses standard C++ libraries and file handling for cross-platform potential.

**🚀 Future Enhancements**

* Migrate product data to external files or databases (e.g., JSON, SQLite).
* Add skin-type-based product recommendations.
* Implement GUI using Qt or SFML.
* Add order history and checkout simulation.
* Enhance unit testing coverage.