Inventory App

Project Overview:

In this project you will be making an app to track a store's inventory. The goal is to design and create the structure of an Inventory App which would allow a store to keep track of its inventory of products. The app will need to store information about price, quantity available, supplier, and a picture of the product. It will also need to allow the user to track sales and shipments and make it easy for the user to order more from the listed supplier.

Why this project?

In the most recent portion of the Diploma program, you learned about data storage in Android, using both SQLite tables and file storage on the device. These skills let you build apps which are critical to small businesses worldwide. By practicing these skills and building this app, you will have the foundation to build similar apps for any kind of business.

What will I learn?

This project is about combining various ideas and skills we've been practicing throughout the course. They include:

- Storing information in a SQLite database
- Integrating Android's file storage systems into that database]=
- Presenting information from files and SQLite databases to users
- Updating information based on user input.
- Creating intents to other apps using stored information.

Build Your Project:

Your project will be evaluated using the Inventory App project rubric shown below.

Additional Criteria

The intent of this project is to give you practice writing raw Java code using the necessary classes provided by the Android framework; therefore, the use of external libraries will not be permitted to complete this project.

Project Rubric:

Content Review

CRITERIA	MEETS SPECIFICATIONS
Overall Layout	The app contains a list of current products and a button to add a new product.
List item layout	Each ListItem displays the product name, current quantity, and price. Each list item also allows the user to track a sale of the iten
Detail layout	The detail layout for each item displays the remainder of th information stored in the database. The detail layout contains buttons to modify the current quantity either by tracking a sale or by receiving a shipmen The detail layout contains a button to order from the supplier. The detail view contains a button to delete the product record entirely.
Layout Best practices	The code adheres to all of the following best practices:

	● Text sizes are defined in sp ● Lengths are defined in dp ● Padding and margin is used appropriately, such tha the views are not crammed up against each other.
Default Textview	When there is no information to display in the database, the layout displays a TextView with instructions on how to populate the database.

Functionality

CRITERIA	MEETS SPECIFICATIONS
Runtime Errors	The code runs without errors
ListView Population	The listView populates with the current products stored in the table.
Add product button	The Add product button prompts the user for information about the product and a picture, each of which are then properly stored in the table.
Input validation	User input is validated. In particular, empty product information is not accepted.
Sale button	The sale button on each list item properly reduces the quantity available by one, unless that would result in a negative quantity.
Detail View intent	Clicking on the rest of each list item sends the user to the detail screen for the correct product.

Modify quantity buttons	The modify quantity buttons in the detail view properly increase and decrease the quantity available for the correc product. The student may also add input for how much to increase c decrease the quantity by.
Order Button	The ‘order more’ button sends an intent to either a phone app or an email app to contact the supplier using the information stored in the database.
Delete button	The delete button prompts the user for confirmation and, if confirmed, deletes the product record entirely and sends th user back to the main activity.
External Libraries and Packages	The intent of this project is to give you practice writing raw Java code using the necessary classes provided by the Android framework; therefore, the use of external libraries for core functionality will not be permitted to complete this project.

Code Readability

CRITERIA	**MEETS SPECIFICATIONS**
Formatting	The code is properly formatted i.e. there are no unnecessary blank lines; there are no unused variables or methods; there is no commented out code. The code also has proper indentation when defining variables and methods.
Naming conventions	All variables, methods, and resource IDs are descriptively named such that another developer reading the code can

	easily understand their function.

Prepare for Submission:

Clean Your Build

Before submitting, please follow the instructions for cleaning your project files. This removes some temporary files and greatly decreases the size of your project.

Clean Your Project Files

Final Submission Checklist

Before submitting your project for evaluation, we recommend that you check that each of the following is true:

1. Your app compiles and runs as expected.
2. You are proud of your app and its output.
3. You completed this project according to instructions.
4. You cleaned the project using the instructions above.
5. You checked your project against the rubric.