🌱 Fertilizer Retailer Accounts

📊 Comprehensive Accounting & Inventory Management System

This project is a dedicated desktop accounting and inventory management application designed specifically for the unique needs of small to medium-sized fertilizer retailers in the agricultural sector.

The application focuses on providing a robust, local solution for managing inventory, tracking transactions, and generating essential financial reports.

1. Purpose Behind the Research

Project Goal

The primary purpose of this project was to develop and implement a tailored system that replaces manual ledgers with a digital, error-free solution for fertilizer retail businesses.

Research Focus

The research centered on exploring the effective integration of a **local database (e.g., SQLite)** to handle persistent, high-volume transactional data. The key objective was to design a system capable of:

- **Real-Time Stock Management:** Ensuring immediate and accurate inventory updates upon sales or purchases.
- **Automated Accounting:** Calculating profit/loss automatically by tracking unit costs against sales prices.
- **Report Generation:** Producing fast, reliable daily, monthly, and custom financial reports.

2. Core System Features

The Fertilizer Retailer Accounts system provides crucial functionality to manage the entire retail operation:

📦 Inventory Control

- **Real-time tracking** of products (Urea, DAP, NPK, etc.).
- Management of stock levels, initial unit costs, and selling prices.
- Prevents stockouts and helps optimize inventory levels.

💰 Transaction Logging

- **Seamless recording** of Sales Invoices and Purchase Receipts.
- Automated updates to stock levels and party balances upon transaction completion.

- Ensures detailed, auditable records for every movement of goods.

👥 Party Management

- Detailed database for both **Customers and Suppliers**.
- Tracks essential contact information.
- Monitors **outstanding balances and credit limits** for efficient collections and payments.

3. Data Structure & Data Creation Process

The application generates and manages its own internal data, stored in a local database.

Data Description

Variable/Table	Type	Description
products	Database Table	Stores product names, stock quantity, unit costs, and selling prices.
transactions	Database Table	Logs every sale/purchase with date, quantity, total amount, and Party ID.
parties	Database Table	Manages customer/supplier details and outstanding credit limits.
current_user	String	Tracks the currently logged-in user session for audit trails.

Data Creation Logic

All data is created programmatically:

1. **Adding Products:** The `add_product()` function initializes new inventory entries in the `products` table.
2. **Processing Transactions:** Functions like `process_sale()` or `process_purchase()` create new records in the `transactions` table while simultaneously running queries to update the `products` table stock levels for immediate consistency.
3. **Reporting:** SQL queries are run against the aggregated data to produce Profit & Loss statements and other required financial reports.

4. Key Performance Metrics

The application was optimized to deliver high performance in critical areas:

- ⚡ **Query Performance:** Optimized for generating real-time stock reports instantly, even with large datasets.
- ✅ **Inventory Accuracy:** Ensures precise stock level updates immediately following sales and purchase entries, eliminating manual calculation errors.

- ⏱️ **Ledger Balancing:** Drastically reduces the time required for end-of-day accounting and ledger reconciliation.
- 🛡️ **Data Integrity:** Robust transaction handling ensures data consistency and prevents loss during system operations.

5. Visual Interface & Retail Environment

Graphical User Interface (GUI)

The application utilizes a clean, intuitive GUI framework (e.g., Python Tkinter/PyQt) with dedicated screens:

- **Dashboard:** Provides an at-a-glance view of critical stock levels, daily sales summary, and cash balances.
- **Inventory Screen:** Allows for easy management, price adjustments, and stock checks.
- **Reports Tab:** A dedicated section for generating required daily, monthly, and custom financial reports.

Real-World Application

The system integrates seamlessly into the daily workflow of a fertilizer shop, managing the complexity of varying unit costs, bulk quantities (bags/tons), and credit transactions typical in the fertilizer trade, ensuring compliance and easy reporting.

6. How to Use

Follow these steps to install, run, and use the Fertilizer Retailer Accounts application:

1. Clone the Repository

```
git clone [https://github.com/YourGitHub/fertilizer_accounts.git](https://github.com/Yo
```

Or download the ZIP file and extract it.

2. Install Requirements

This project uses **Python**. Ensure you have Python 3.8 or above installed. Install required packages (if necessary, check `requirements.txt`):

```
pip install -r requirements.txt
```

3. Run the Application

Navigate to your project folder and run the main file:

```
python main_accounts_app.py
```

4. Basic Operation

- **Transactions:** Navigate to the **Transaction** tab to log new sales (invoices) or stock receipts (purchases).
- **Data Entry:** Ensure you enter the **Product Unit Cost** correctly to maintain accurate profit calculations.
- **Backup:** Regularly use the **Export Data** feature for secure backup.

7. Contact Information

If you have questions, suggestions, or want to collaborate, feel free to reach out:

- **Name**: Manas Solanki
- **Email:** manas.25bai10589@vitbhopal.ac.in
- **Contact Purpose:** Bug reports, feature requests (e.g., barcode support, GST integration), or general queries.

8. License

This project is released under the **MIT License**. This license allows you to freely use, modify, distribute, and share the code as long as proper credit is given to the original author.

You are free to:

- Use the code for personal, academic, or commercial business operations.
- Modify or expand the application (e.g., add new reports or features).
- Distribute your modified version.

You must:

- Include the original copyright notice.
- Include the MIT License in any redistributed version.