Developer Documentation: POS Lite

Overview

POS Lite is built to provide small businesses with a simple, offline-compatible point-of-sale solution. It features inventory management, sales processing, and reporting, using Python with SQLite for data persistence.

Project Structure

```
bash
CopyEdit
POSLite/
├── main.py          # Main application logic
├── pos_lite.db      # SQLite database (auto-generated)
├── README.md        # Documentation
├── requirements.txt# Dependencies (if needed)
└── dist/            # Generated executables (via PyInstaller)
```

Tech Stack

- **Programming Language**: Python
- **GUI Framework**: Tkinter
- **Database**: SQLite
- **Packaging Tool**: PyInstaller

Key Modules

1. **Database Operations**:
    - `initialize_db`: Sets up `inventory` and `sales` tables.
    - `save_product`: Adds new products to the database.
    - `update_inventory_product`: Updates product details in the database.
    - `save_sale`: Records completed sales.
2. **GUI**:
    - `POSApp`: Handles all user interactions and links to backend functions.
    - Tabs:
        - **Inventory**: Manage products.
        - **Sales**: Process transactions.
        - **Reports**: Generate and view sales data.

Setup

Dependencies

If using the source code, ensure the following:

- Python 3.8 or higher
- Required libraries:

```
bash
CopyEdit
pip install tkinter sqlite3
```

Running Locally

- Run the script:

```
bash
CopyEdit
python main.py
```

Creating an Executable

- Install PyInstaller:

```
bash
CopyEdit
pip install pyinstaller
```

- Generate the executable:

```
bash
CopyEdit
pyinstaller --onefile --windowed --name POSLite main.py
```

- The executable will appear in the `dist/` folder.

Testing

Unit Testing:

- Test database functions (e.g., `save_product`, `fetch_inventory`).
- Verify GUI actions using manual or automation testing tools.

Database Validation:

Ensure the `pos_lite.db` schema matches expectations:

```
sql
CopyEdit
PRAGMA table_info(inventory);
PRAGMA table_info(sales);
```

Contributing

1. Fork the repository.
2. Create a feature branch:

   ```
   bash
   CopyEdit
   git checkout -b feature-name
   ```

3. Commit changes and submit a pull request.