

Xây dựng Ứng dụng Desktop Hệ Thống Quản Lý Đơn Hàng

Lập trình Python(5)



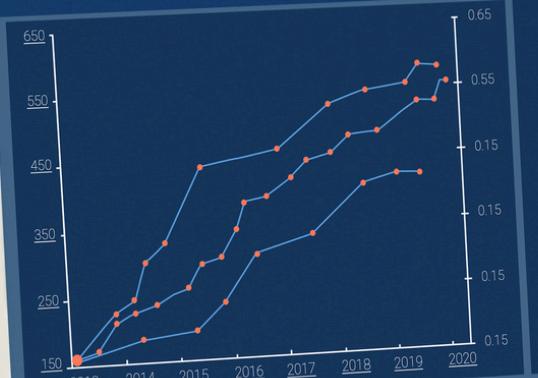
+84 336 440 523



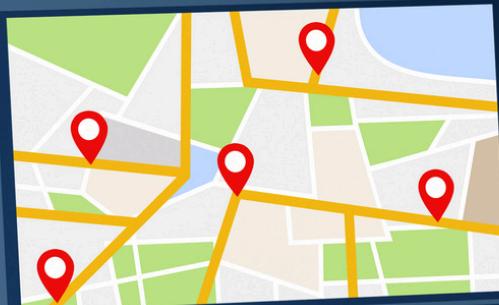
quantl.21it@vku.udn.vn

Dashboard

- Waiting for payment 589
- Paid 300
- Products awaiting delivery 150
- Products shipped 150



Warehouse Management



Transpo



Nội dung

- 01** **Tổng quan đề tài**
- 02** **Các Tính Năng Chính
của Ứng Dụng**
- 03** **Công Nghệ Sử Dụng**
- 04** **Kiến trúc hệ thống**
- 05** **Demo sản phẩm**
- 06** **Hướng Phát Triển
Tương Lai**
- 07** **Kết Luận**



Giới thiệu

Tổng quan đề tài

Hệ thống Quản lý Đơn hàng là một ứng dụng phần mềm được thiết kế để hỗ trợ quản lý hiệu quả các hoạt động kinh doanh, bao gồm quản lý người dùng, sản phẩm, đơn hàng và phân tích dữ liệu doanh thu.

Phạm vi ứng dụng:

- Phù hợp cho các cửa hàng bán lẻ, doanh nghiệp thương mại điện tử vừa và nhỏ.
- Hỗ trợ quản lý thông tin khách hàng, sản phẩm, đơn hàng và tạo báo cáo thống kê chi tiết.



CÁC TÍNH NĂNG CHÍNH



Quản lý người dùng

- Đăng ký, đăng nhập, phân quyền theo vai trò (Admin, Staff, Customer).
- Thêm, sửa, xóa và xuất dữ liệu người dùng ra Excel.



Quản lý sản phẩm

- Thêm, sửa, xóa sản phẩm với thông tin chi tiết (tên, mã, giá, tồn kho, hình ảnh).
- Lọc sản phẩm theo danh mục, trạng thái; hỗ trợ tải ảnh sản phẩm.



Quản lý đơn hàng

- Tạo, cập nhật trạng thái, hủy đơn hàng (New, Shipping, Completed, Canceled).
- Theo dõi đơn hàng qua bản đồ
- Gửi email thông báo tự động khi tạo/hủy/cập nhật đơn hàng.



Thống kê và báo cáo

- Biểu đồ doanh thu theo danh mục, tháng; phân bố trạng thái đơn hàng.
- Xuất thống kê ra Excel với các sheet chi tiết.



Gợi ý thông minh

- Gợi ý sản phẩm bán chạy theo danh mục.
- Hiển thị thông tin nhân viên bán hàng xuất sắc.

○
○
○

Đối tượng người dùng

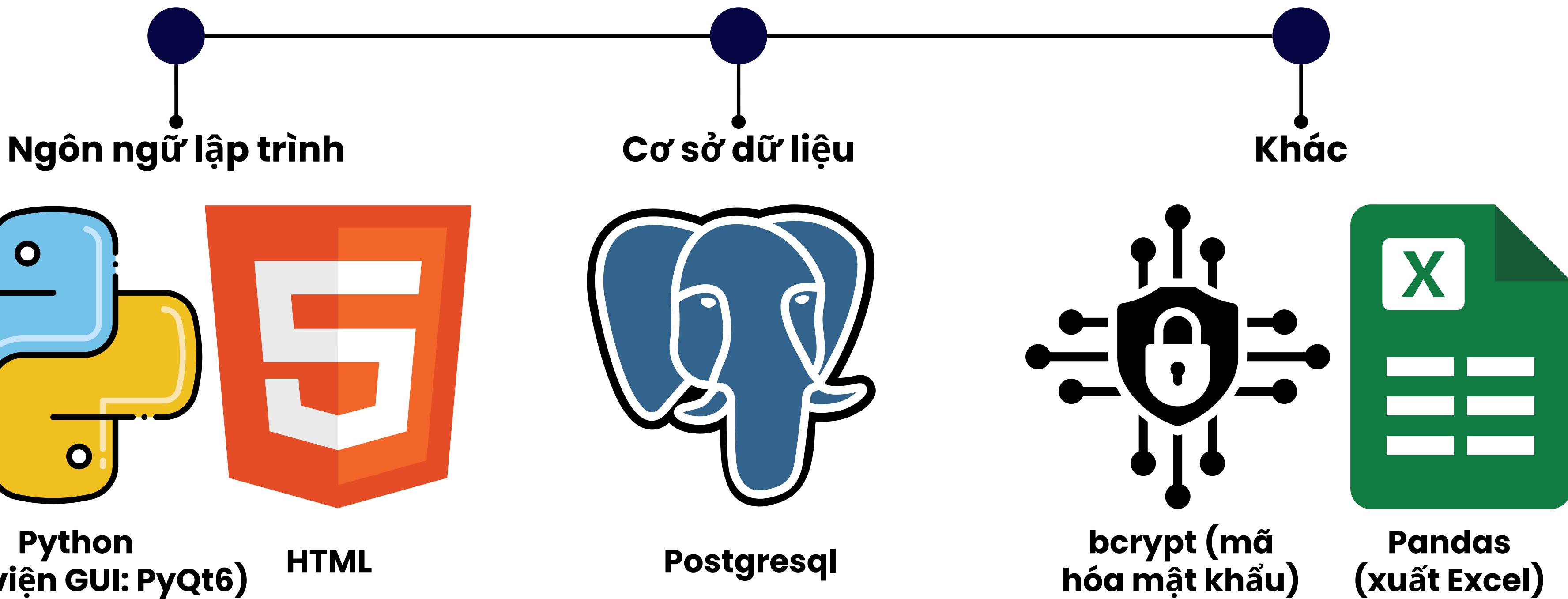
Admin: Quản lý toàn bộ hệ thống (người dùng, sản phẩm, đơn hàng, thống kê).



Staff: Quản lý sản phẩm, đơn hàng, xem thống kê cá nhân.

Customer: Đặt hàng, theo dõi đơn hàng, xem bản đồ giao hàng.

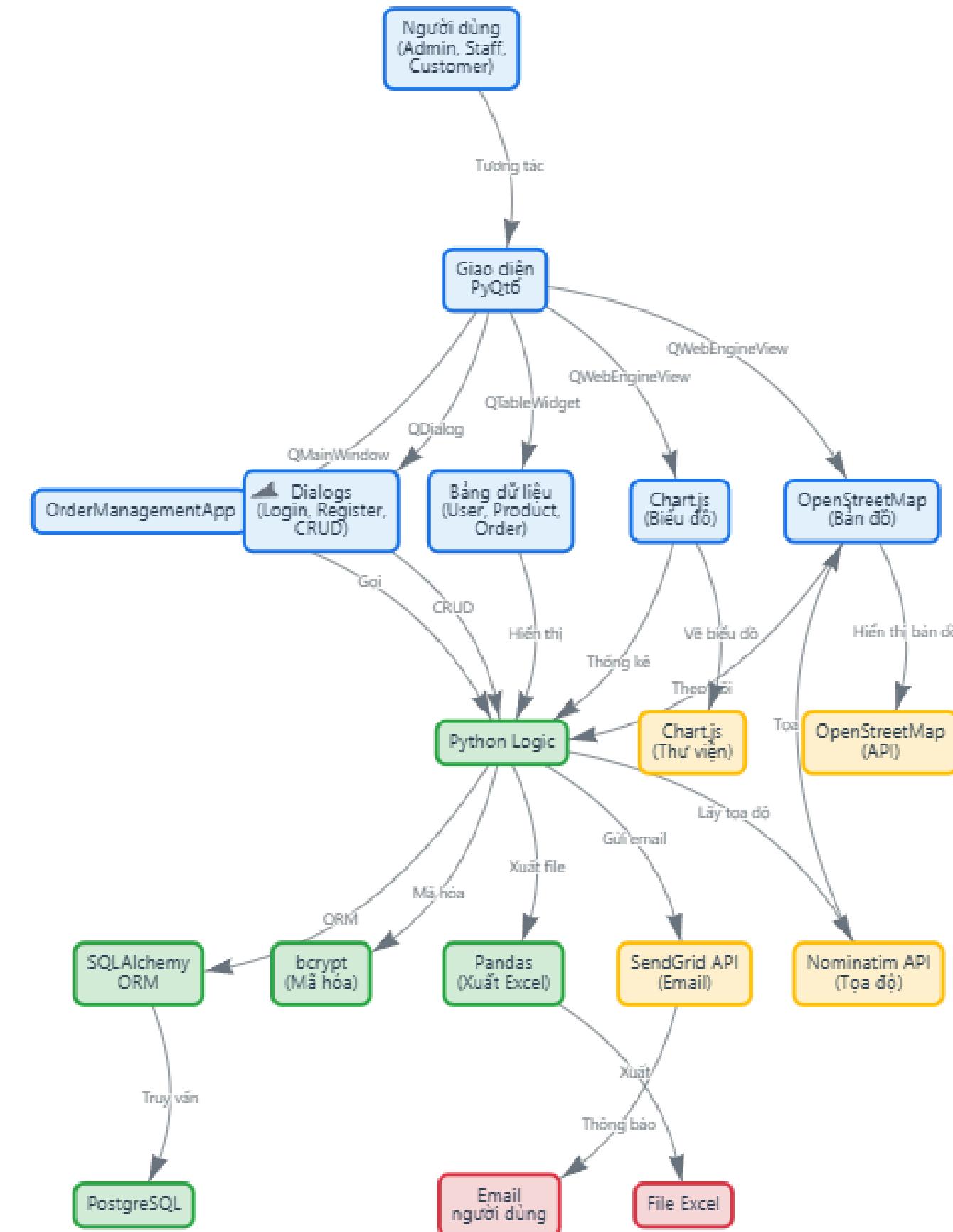
Công nghệ sử dụng



Công nghệ sử dụng (Tích hợp bên thứ ba)



Kiến trúc hệ thống



Demo sản phẩm

The screenshot shows a Windows desktop environment with a Python application running in a terminal window and a login dialog box.

Terminal Window:

```
PS D:\quan_ly_don_hang> cd order_management_gui
PS D:\quan_ly_don_hang\order_management_gui> python main.py
Bắt đầu tạo tài khoản admin mặc định...
Tài khoản admin đã tồn tại.
Kết thúc tạo tài khoản admin mặc định.
```

Login Dialog Box:

The dialog box is titled "Đăng nhập" (Login). It contains two input fields: "Tên đăng nhập:" (Username) with the value "luu" and "Mật khẩu:" (Password) with the placeholder "Nhập mật khẩu". Below the fields are two buttons: a blue "Đăng nhập" (Login) button and a green "Đăng ký" (Register) button.

VS Code IDE:

- EXPLORER:** Shows the project structure under "QUAN LY DON HANG".
- CODE:** Displays the "gui.py" file code:

```
1665     class OrderManagementApp(QMainWindow):
1666         def __init__(self):
1667             super().__init__()
1668             self.setWindowTitle("Quản lý đơn hàng")
1669             self.setWindowIcon(QIcon("icon/icon.png"))
1670             self.setCentralWidget(self.create_main_widget())
1671             self.create_menus()
1672             self.create_toolbars()
1673             self.create_status_bar()
1674             self.create_widgets()
1675             self.create_layouts()
1676             self.create_connections()
1677             self.load_settings()
1678             self.load_data()
1679
1680         def create_main_widget(self):
1681             main_widget = QWidget()
1682             main_widget.setLayout(self.main_layout)
1683             return main_widget
1684
1685         def create_menus(self):
1686             self.menu_bar = QMenuBar(self)
1687             self.menu_bar.setNativeMenuBar(False)
1688             self.menu_bar.addMenu(self.create_file_menu())
1689             self.menu_bar.addMenu(self.create_edit_menu())
1690             self.menu_bar.addMenu(self.create_view_menu())
1691             self.menu_bar.addMenu(self.create_terminal_menu())
1692             self.menu_bar.addMenu(self.create_help_menu())
1693             self.setMenuBar(self.menu_bar)
1694
1695         def create_file_menu(self):
1696             file_menu = QMenu("File", self)
1697             file_menu.addAction(self.create_new_action())
1698             file_menu.addAction(self.create_open_action())
1699             file_menu.addAction(self.create_save_action())
1700             file_menu.addAction(self.create_exit_action())
1701             return file_menu
1702
1703         def create_edit_menu(self):
1704             edit_menu = QMenu("Edit", self)
1705             edit_menu.addAction(self.create_cut_action())
1706             edit_menu.addAction(self.create_copy_action())
1707             edit_menu.addAction(self.create_paste_action())
1708             edit_menu.addAction(self.create_delete_action())
1709             edit_menu.addAction(self.create_select_all_action())
1710             return edit_menu
1711
1712         def create_view_menu(self):
1713             view_menu = QMenu("View", self)
1714             view_menu.addAction(self.create_toolbar_action())
1715             view_menu.addAction(self.create_statusbar_action())
1716             view_menu.addAction(self.create_font_size_action())
1717             return view_menu
1718
1719         def create_terminal_menu(self):
1720             terminal_menu = QMenu("Terminal", self)
1721             terminal_menu.addAction(self.create_terminal_action())
1722             return terminal_menu
1723
1724         def create_help_menu(self):
1725             help_menu = QMenu("Help", self)
1726             help_menu.addAction(self.create_about_action())
1727             return help_menu
1728
1729         def create_new_action(self):
1730             action = QAction("New", self)
1731             action.setShortcut("Ctrl+N")
1732             action.triggered.connect(self.create_new_file)
1733             return action
1734
1735         def create_open_action(self):
1736             action = QAction("Open", self)
1737             action.setShortcut("Ctrl+O")
1738             action.triggered.connect(self.open_file)
1739             return action
1740
1741         def create_save_action(self):
1742             action = QAction("Save", self)
1743             action.setShortcut("Ctrl+S")
1744             action.triggered.connect(self.save_file)
1745             return action
1746
1747         def create_exit_action(self):
1748             action = QAction("Exit", self)
1749             action.setShortcut("Ctrl+Q")
1750             action.triggered.connect(self.close)
1751             return action
1752
1753         def create_cut_action(self):
1754             action = QAction("Cut", self)
1755             action.setShortcut("Ctrl+X")
1756             action.triggered.connect(self.cut)
1757             return action
1758
1759         def create_copy_action(self):
1760             action = QAction("Copy", self)
1761             action.setShortcut("Ctrl+C")
1762             action.triggered.connect(self.copy)
1763             return action
1764
1765         def create_paste_action(self):
1766             action = QAction("Paste", self)
1767             action.setShortcut("Ctrl+V")
1768             action.triggered.connect(self.paste)
1769             return action
1770
1771         def create_delete_action(self):
1772             action = QAction("Delete", self)
1773             action.setShortcut("Delete")
1774             action.triggered.connect(self.delete)
1775             return action
1776
1777         def create_select_all_action(self):
1778             action = QAction("Select All", self)
1779             action.setShortcut("Ctrl+A")
1780             action.triggered.connect(self.select_all)
1781             return action
1782
1783         def create_toolbar_action(self):
1784             action = QAction("Toolbar", self)
1785             action.setCheckable(True)
1786             action.setChecked(self.toolbar.isVisible())
1787             action.triggered.connect(self.toggle_toolbar)
1788             return action
1789
1790         def create_statusbar_action(self):
1791             action = QAction("Status Bar", self)
1792             action.setCheckable(True)
1793             action.setChecked(self.statusBar.isVisible())
1794             action.triggered.connect(self.toggle_statusbar)
1795             return action
1796
1797         def create_font_size_action(self):
1798             action = QAction("Font Size", self)
1799             action.setCheckable(True)
1800             action.setChecked(self.font_size.isVisible())
1801             action.triggered.connect(self.toggle_font_size)
1802             return action
1803
1804         def create_terminal_action(self):
1805             action = QAction("Terminal", self)
1806             action.setCheckable(True)
1807             action.setChecked(self.terminal.isVisible())
1808             action.triggered.connect(self.toggle_terminal)
1809             return action
1810
1811         def create_about_action(self):
1812             action = QAction("About", self)
1813             action.triggered.connect(self.about)
1814             return action
1815
1816         def about(self):
1817             QMessageBox.about(self, "About", "This is a simple terminal application written in Python using PyQt5 framework. It includes basic file operations like New, Open, Save, Exit, and a terminal window for executing commands. The application is designed to be user-friendly and efficient for daily tasks. © 2023. All rights reserved.")
```
- PROBLEMS:** Shows the terminal output.
- OUTPUT:** Shows the terminal output.
- DEBUG CONSOLE:** Shows the terminal output.
- STATUS BAR:** Shows the current line (Ln 4111, Col 14), spaces (Spaces: 4), encoding (UTF-8), and file type (CRLF).
- SYSTEM TRAY:** Shows the date (28/05/2025), time (5:56 PM), weather (86°F, Partly sunny), and battery status.

HƯỚNG PHÁT TRIỂN

- Tăng Cường Tự Động Hóa
 - Tích hợp AI: Gợi ý sản phẩm thông minh dựa trên hành vi khách hàng.
 - Tự động hóa quy trình: Tự động cập nhật trạng thái đơn hàng và thông báo qua email/SMS.
 - Chatbot hỗ trợ: Hỗ trợ khách hàng 24/7, giải đáp thắc mắc và theo dõi đơn hàng.
- Mở Rộng Tính Năng
 - Hỗ trợ đa nền tảng: Phát triển ứng dụng di động (iOS/Android) với giao diện thân thiện.
 - Thanh toán trực tuyến: Tích hợp cổng thanh toán (Momo, ZaloPay, thẻ tín dụng).
- Cải Thiện Trải Nghiệm Người Dùng
 - Giao diện tối ưu: Thiết kế lại UI/UX hiện đại, hỗ trợ đa ngôn ngữ.
 - Báo cáo nâng cao: Thống kê chi tiết theo thời gian thực, xuất báo cáo PDF.
 - Cá nhân hóa: Tùy chỉnh giao diện và thông báo theo sở thích người dùng.
- Mở Rộng Hệ Sinh Thái
 - Tích hợp bên thứ ba: Kết nối với hệ thống kho, vận chuyển (GHN, GHTK).
 - API mở: Cho phép đối tác tích hợp với hệ thống quản lý đơn hàng.
 - Hỗ trợ thương mại điện tử: Liên kết với các sàn TMĐT (Shopee, Lazada).

KẾT LUẬN

- Hệ thống quản lý hiệu quả: Tích hợp quản lý người dùng, sản phẩm, đơn hàng và thống kê.
- Trải nghiệm người dùng tối ưu: Giao diện thân thiện, hỗ trợ đa vai trò (Admin, Staff, Customer).
- Tính năng hiện đại: Tự động hóa thông báo email, bản đồ theo dõi đơn hàng, xuất báo cáo Excel.



Cảm ơn thầy
đã lắng nghe



+84 336 440 523



quantl.21it@vku.udn.vn

Thank
You!!!