**TRƯỜNG ĐẠI HỌC AN GIANG**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BÁO CÁO ĐỒ ÁN**

**Chuyên đề Python (COS525)**

**XÂY DỰNG ỨNG DỤNG QUẢN LÝ NHÂN VIÊN VỚI**

**PYTHON, TKINTER VÀ MYSQL**

Giảng viên hướng dẫn:

Ths. Nguyễn Ngọc Minh

Sinh viên thực hiện:

DTH235696 - Bùi Thị Phương Mai - DH24TH2 - Nhóm 2 - Tổ 1

DTH235698 - Nguyễn Đỗ Minh - DH24TH2 - Nhóm 2 - Tổ 1

**AN GIANG, 11-2025**

**Thành viên và phân công công việc:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Họ tên** | **MSSV** | **Lớp** | **Nhiệm vụ** | **Tỉ lệ tham gia** |
| Bùi Thị Phương Mai | DTH235696 | DH24TH2 | Code, Word, SQL | 100% |
| Nguyễn Đỗ Minh | DTH235698 | DH24TH2 | Code, PPT, Word, SQL | 100% |

**Phân công công việc chi tiết**

- Bùi Thị Phương Mai

+ SQL: Tạo cơ sở dữ liệu

+ Word: Chụp giao diện, chụp code, mô tả đồ án.

+ Code và thiết kế giao diện các form: làm song song tất cả các form ( Login, Trang chủ Quản Lý, Trang chủ Nhân Viên, form Tuyến Du Lịch, form Khách Hàng, form Nhân viên, form Đặt Vé, form Chuyến Đi, form Doanh Thu) và các tác vụ ( thêm, xóa, sửa, lưu, hủy, thoát).

- Nguyễn Đỗ Minh:

+ SQL: Tạo cơ sở dữ liệu.

+ Word: Nội dung lý thuyết, chụp CSDL.

+ Code và thiết kế giao diện các form: làm song song tất cả các form ( Login, Trang chủ Quản Lý, Trang chủ Nhân Viên, form Tuyến Du Lịch, form Khách Hàng, form Nhân viên, form Đặt Vé, form Chuyến Đi, form Doanh Thu) và các tác vụ ( thêm, xóa, sửa, lưu, hủy, thoát).

**I. Đặt vấn đề**

- Du lịch là một trong những ngành kinh tế mũi nhọn của Việt Nam, đóng vai trò quan trọng trong việc quảng bá hình ảnh đất nước, con người và văn hóa Việt Nam ra thế giới. Trong những năm gần đây, cùng với sự phát triển mạnh mẽ của nền kinh tế và hội nhập quốc tế, ngành du lịch Việt Nam đã đạt được nhiều thành tựu đáng kể, thu hút hàng triệu lượt khách trong và ngoài nước mỗi năm.

- Tuy nhiên, bên cạnh những kết quả đạt được, công tác quản lý hoạt động du lịch vẫn còn tồn tại nhiều hạn chế như: thiếu sự đồng bộ giữa các cơ quan quản lý, công tác quy hoạch còn chồng chéo, tình trạng ô nhiễm môi trường và quá tải tại các điểm du lịch, cùng với việc ứng dụng công nghệ thông tin trong quản lý còn hạn chế.

- Chính vì vậy, việc nghiên cứu và xây dựng hệ thống quản lý du lịch hiệu quả, hiện đại, ứng dụng công nghệ thông tin là rất cần thiết. Điều này không chỉ giúp nâng cao hiệu quả quản lý, mà còn góp phần phát triển ngành du lịch theo hướng bền vững, chuyên nghiệp và hội nhập quốc tế

**II. Tổng quan và cơ sở lý thuyết**

****1. Đặt vấn đề****

- Trong quá trình phát triển du lịch, việc xây dựng và quản lý tuyến du lịch đóng vai trò quan trọng nhằm đảm bảo hoạt động tham quan diễn ra hợp lý, an toàn và hiệu quả. Tuy nhiên, nhiều doanh nghiệp lữ hành hiện gặp khó khăn trong việc tổ chức, điều hành, kiểm soát chất lượng và lưu trữ thông tin tuyến. Bài toán đặt ra là cần có cơ sở lý thuyết và phương pháp quản lý tuyến du lịch một cách khoa học để phục vụ cho công tác vận hành và hỗ trợ xây dựng hệ thống quản lý.

**2. Lịch sử giải quyết vấn đề**

- Quản lý tuyến du lịch đã được nghiên cứu và áp dụng từ lâu thông qua các mô hình quản lý tour truyền thống và quy trình điều hành của các doanh nghiệp lữ hành. Nhiều nghiên cứu tập trung vào việc tối ưu hóa lịch trình, tổ chức hướng dẫn viên, phân bổ chi phí và đánh giá dịch vụ. Tuy nhiên, trong bối cảnh công nghệ thông tin phát triển, việc áp dụng các phương pháp quản lý hiện đại, tự động hóa dữ liệu và tối ưu hóa vận hành tuyến vẫn còn nhiều hạn chế, đòi hỏi cần được tiếp tục nghiên cứu và hoàn thiện.

**3. Phạm vi của đề tài**

- Phạm vi nghiên cứu tập trung vào các nội dung chính liên quan đến quản lý tuyến du lịch, bao gồm: quản lý thông tin tuyến, quản lý điểm tham quan, quản lý chuyến đi, phân công nhân sự (đặc biệt là hướng dẫn viên), kiểm soát chất lượng dịch vụ, và theo dõi doanh thu – chi phí của từng tuyến. Đề tài không đi sâu vào phân tích chính sách du lịch cấp quốc gia hoặc nghiên cứu thị trường, mà chỉ tập trung vào quản lý nghiệp vụ của tuyến trong một doanh nghiệp lữ hành.

**4. Phương pháp nghiên cứu**

- Đề tài sử dụng các phương pháp nghiên cứu như:

+ Nghiên cứu tài liệu: thu thập lý thuyết về du lịch, quản lý tuyến, phương pháp điều hành tour.

+ Phân tích hệ thống: xác định quy trình quản lý tuyến, các thực thể dữ liệu và mối quan hệ giữa chúng.

+ Tổng hợp – đánh giá: đưa ra nhận xét và giải pháp cải thiện phù hợp.

**5. Trình bày tóm tắt cơ sở lý thuyết sử dụng**

- Cơ sở lý thuyết của quản lý tuyến du lịch bao gồm:

+ Lý thuyết về tuyến du lịch: khái niệm, cấu trúc tuyến, nguyên tắc xây dựng và khai thác tuyến.

+ Lý thuyết về quản lý tuyến: nội dung quản lý, tiêu chí đánh giá chất lượng, quy trình điều hành.

+ Quản lý dữ liệu trong du lịch: mô hình cơ sở dữ liệu, chuẩn hóa, khóa chính – khóa ngoại, ràng buộc dữ liệu.

+ Hệ thống thông tin quản lý (MIS): thu thập – xử lý – phân tích dữ liệu phục vụ điều hành tuyến.

+ Ứng dụng công nghệ thông tin trong du lịch: các phần mềm quản lý tour, hệ thống cơ sở dữ liệu, tự động hóa nghiệp vụ.

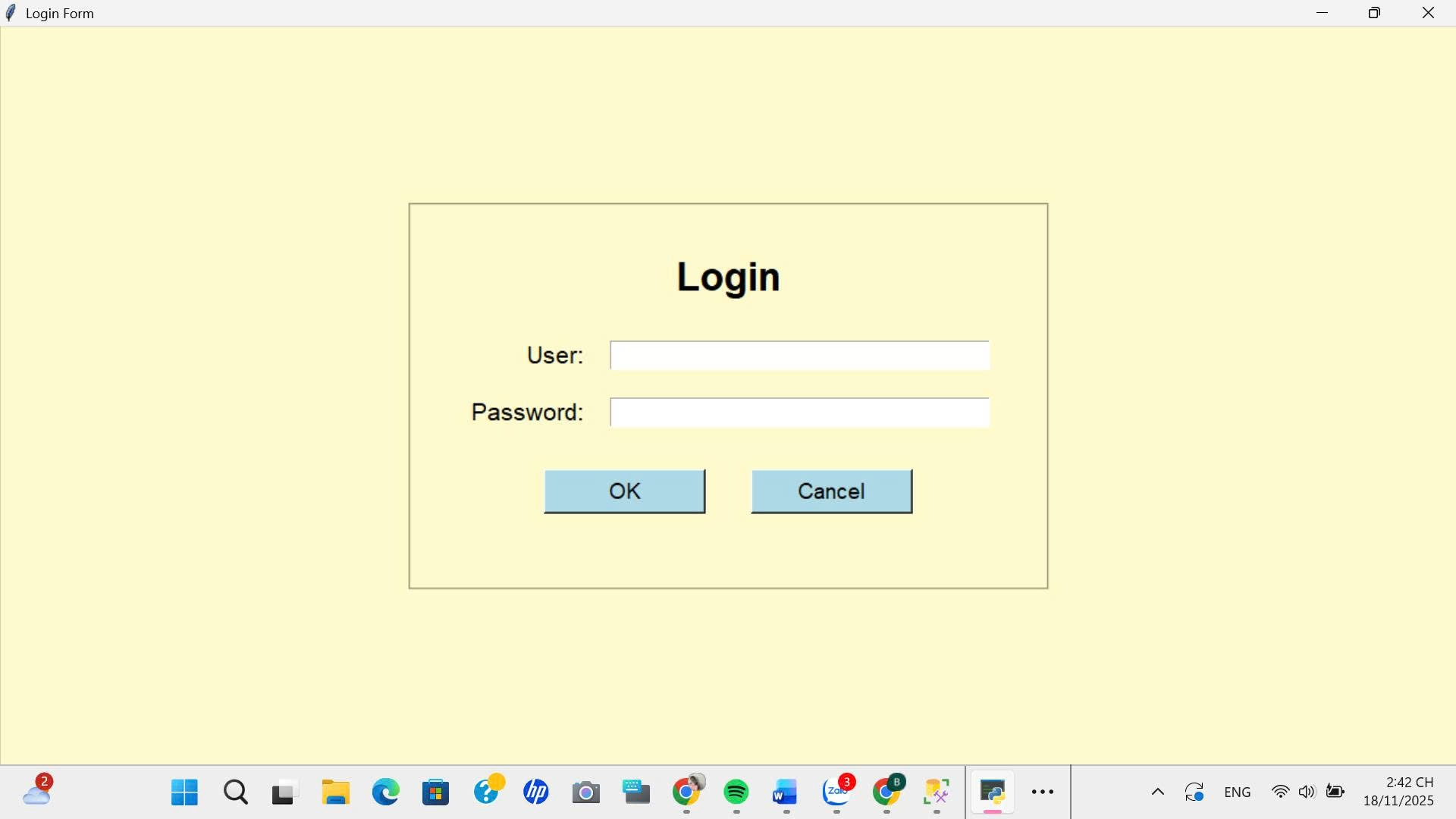
**III. Kết quả đạt được và Demo Code**

- Phân tích và xây dựng một ứng dụng bằng thư viện Tkinter trong Pyhthon có các chức năng cơ bản như : thêm, sửa, xóa, lưu … kết hợp với lưu trữ dữ liệu vào cơ sở dữ liệu SQL server.

- Vận dụng kỹ thuật lập trình ngôn ngữ Python để giải quyết các vấn đề trong thực tế về đề tài quản lý các tuyến du lịch.

**A. THIẾT KẾ GIAO DIỆN**

**1. Giao diện đăng nhập**

****

Hình 1: giao diện khung khi đăng nhập

**2. Giao diện trang chủ**

****

Hình 2: Giao diện trang chủ quản lý

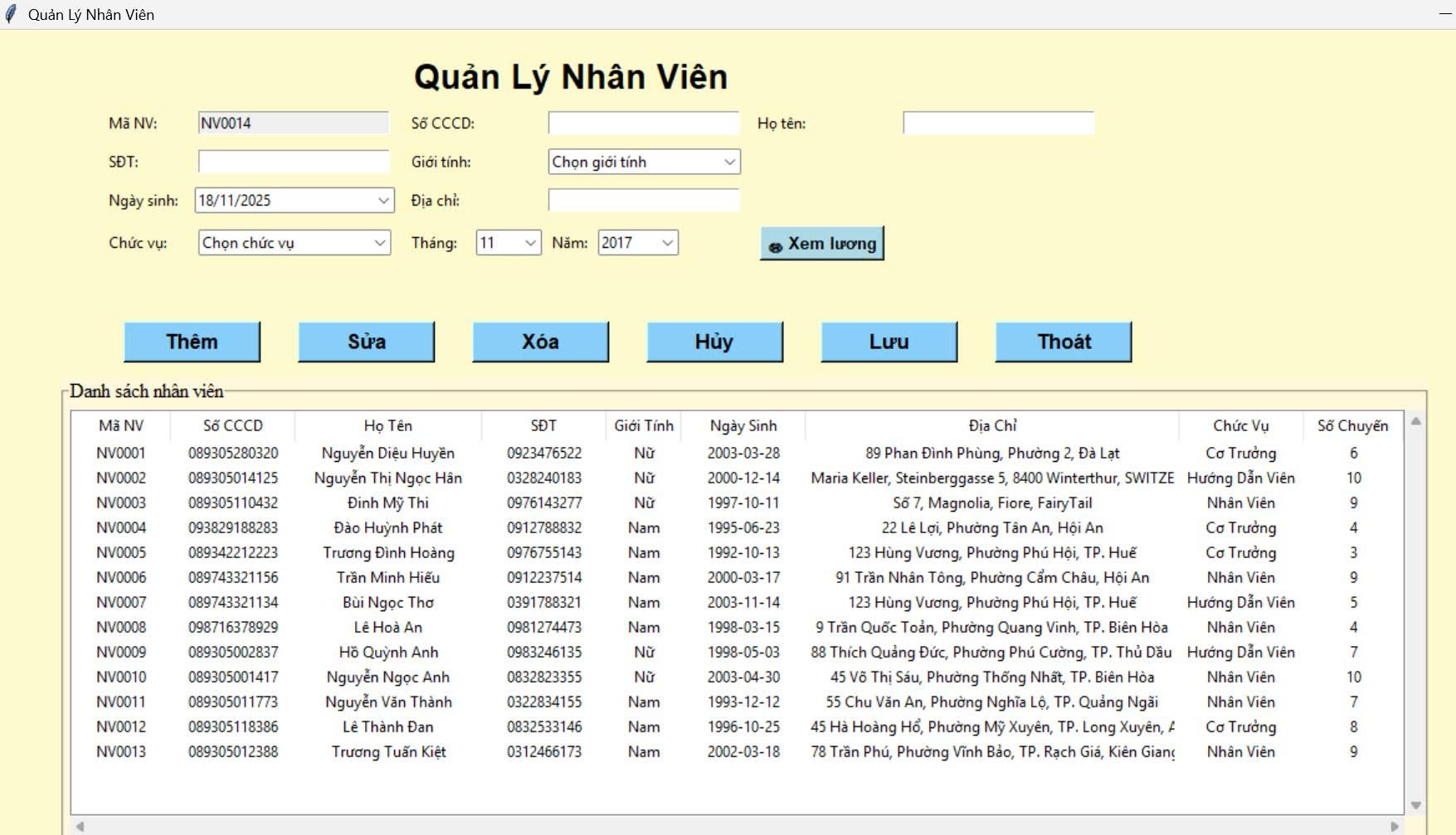
****

Hình 3: giao diện trang chủ nhân viên

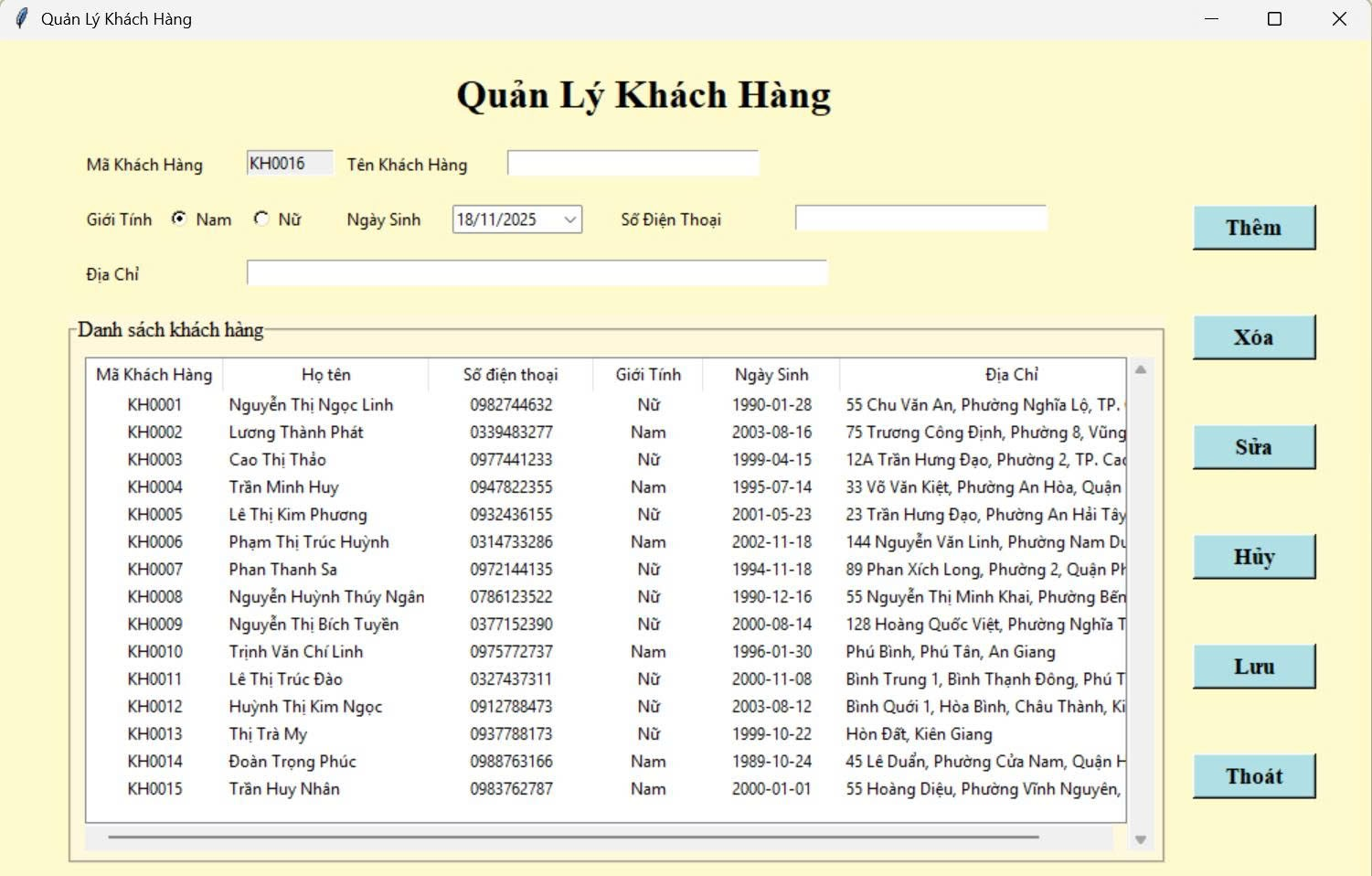
**3. Giao diện quản lý**

****

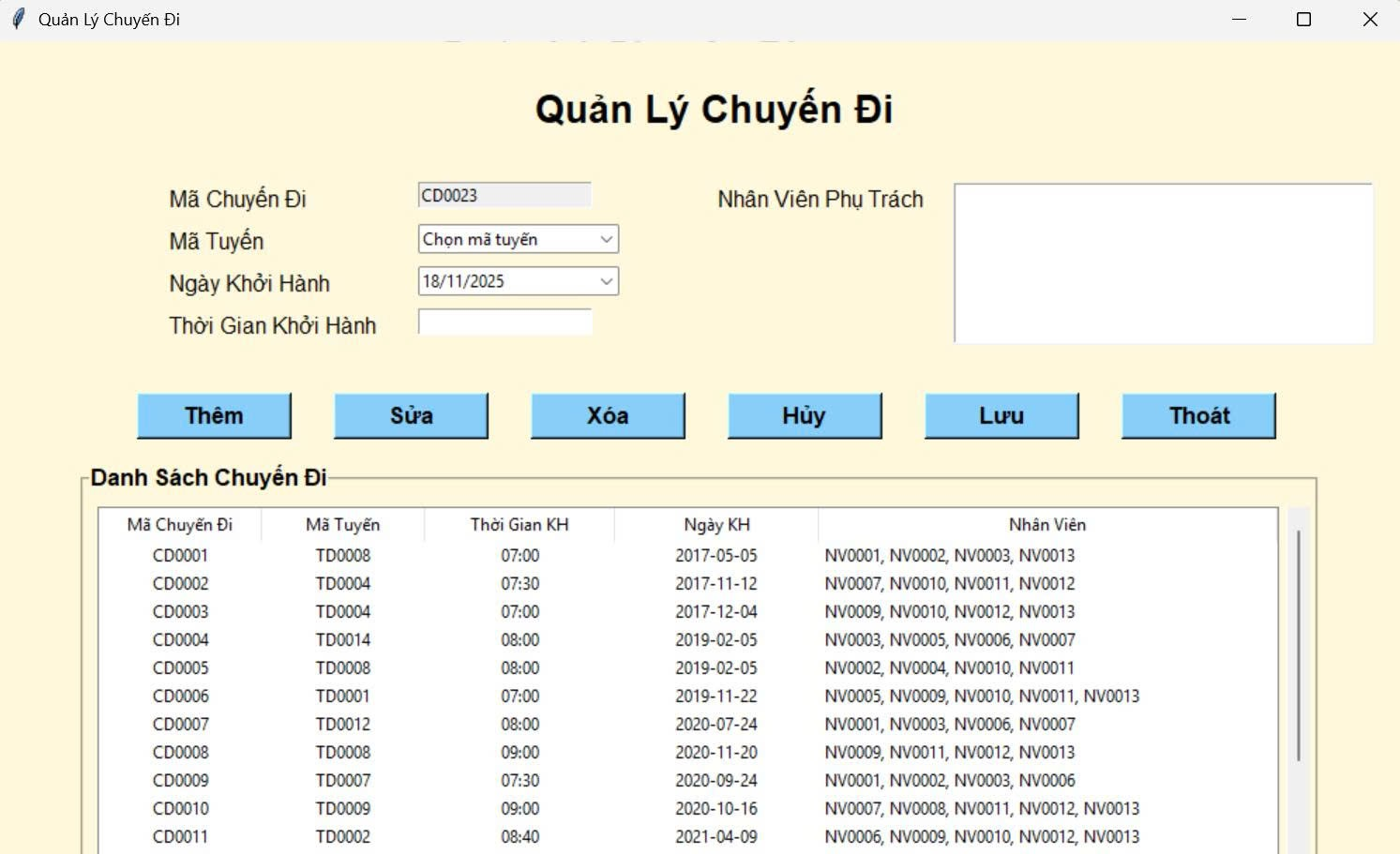
Hình 4: giao diện quản lý tuyến du lịch

****

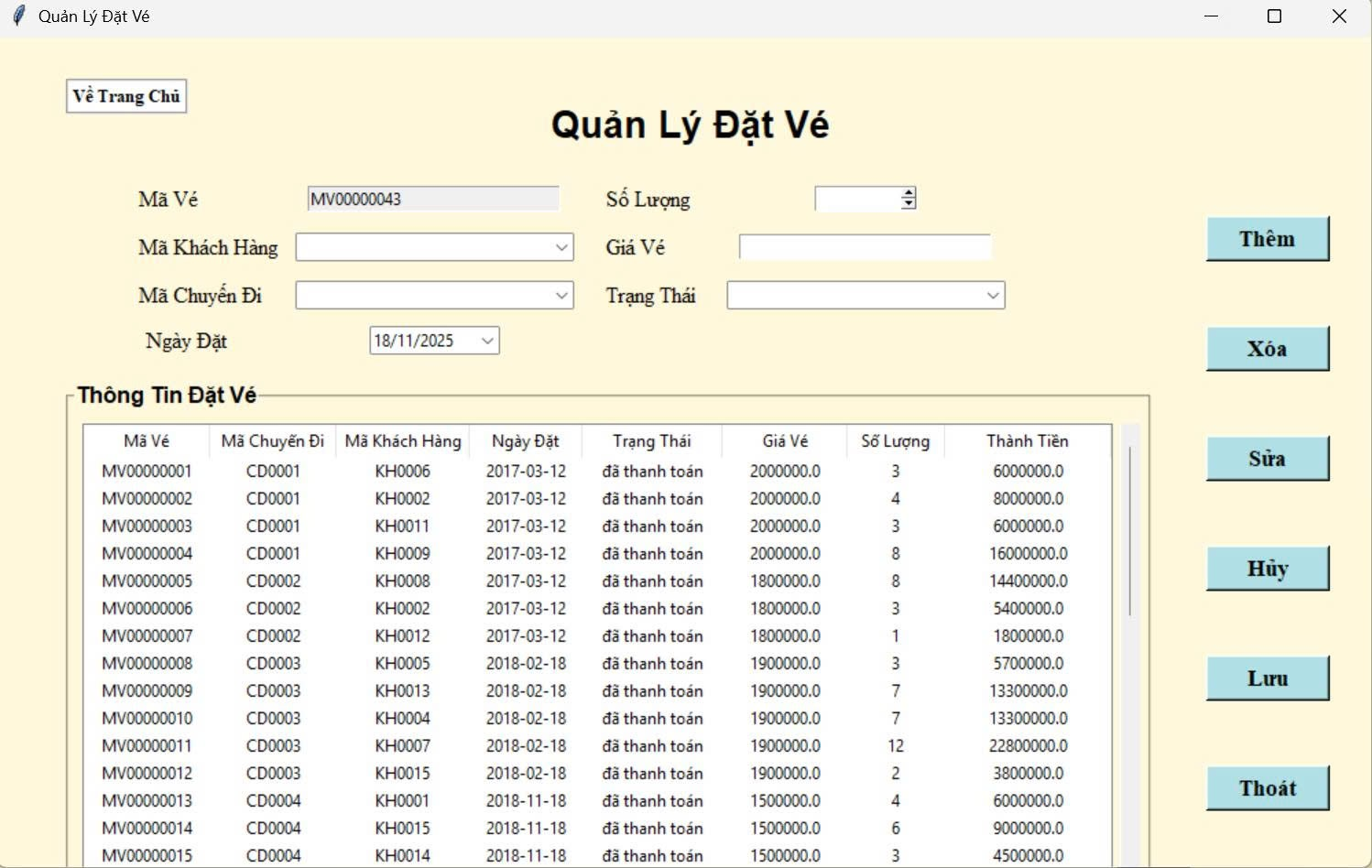
Hình 5: giao diện quản lý nhân viên

****

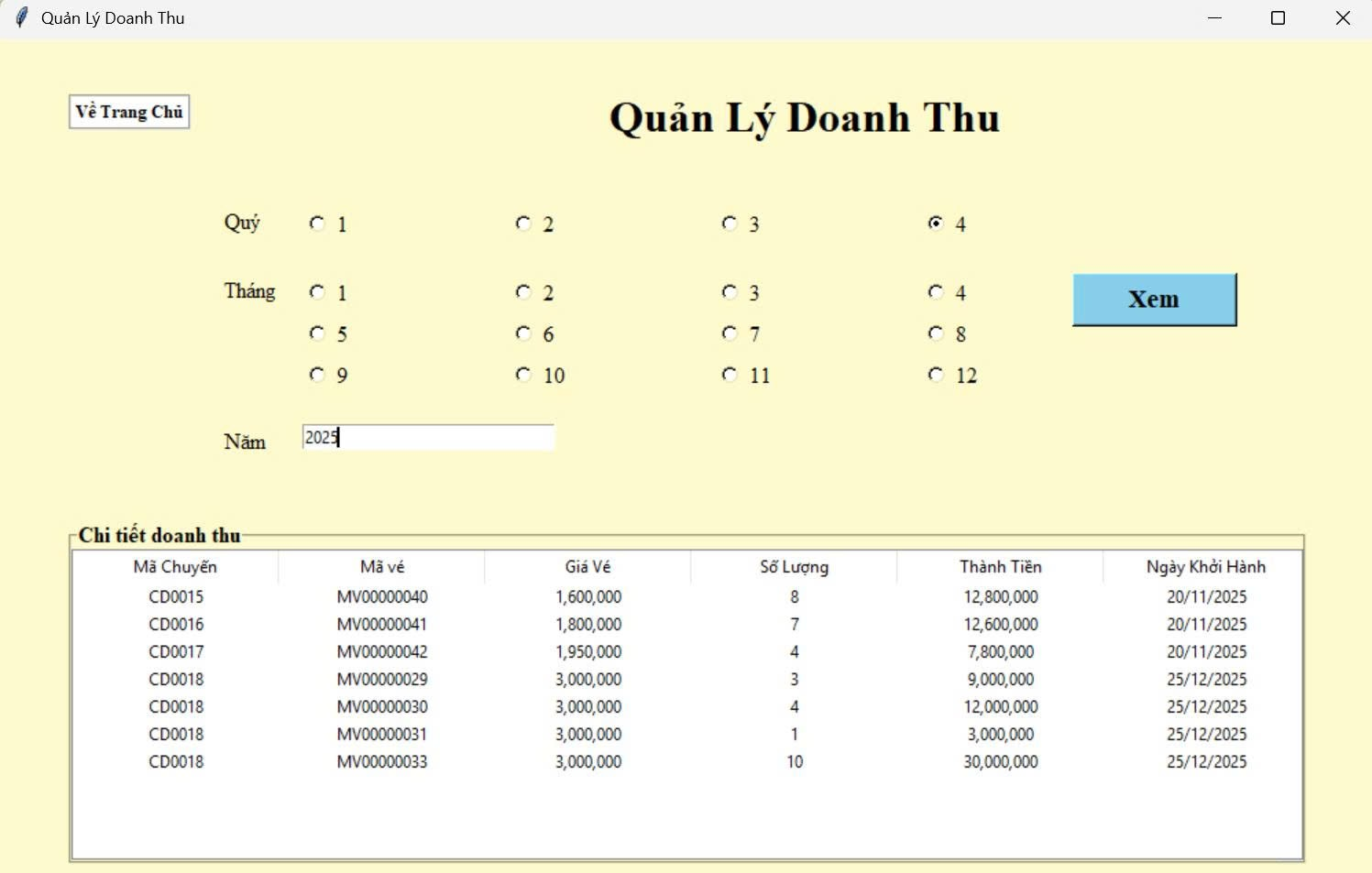
Hình 6: giao diện quản lý khách hàng

****

Hình 7: giao diện quản lý chuyến đi

****

Hình 8: giao diện quản lý đặt vé

****

Hình 9: giao diện quản lý doanh thu

**B. CÁC BƯỚC XÂY DỰNG ỨNG DỤNG**

Để xây dựng ứng dụng quản lý sản phẩm bằng Python, Tkinter và MySQL, cần cài đặt các thư viện cần thiết, thiết kế cơ sở dữ liệu MySQL để lưu trữ thông tin sản phẩm, sử dụng Tkinter để tạo giao diện người dùng, và kết hợp Python với thư viện mysql.connector để thực hiện các thao tác như thêm, xóa, sửa, và hiển thị danh sách Nhân Viên

**1. Cài đặt các thư viện cần thiết**

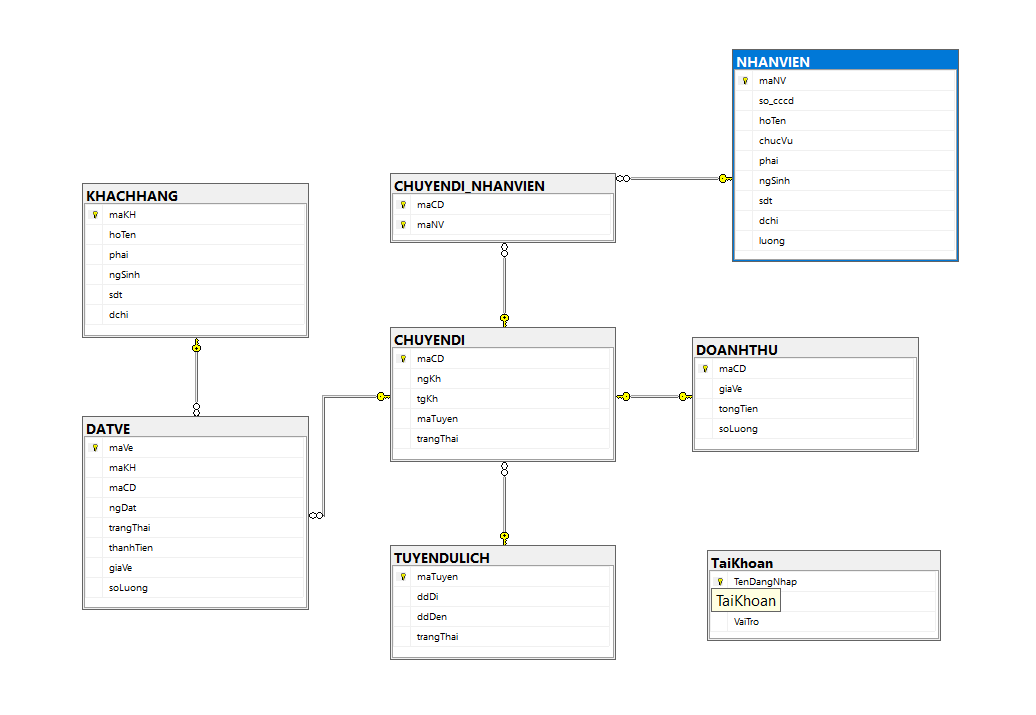
- Python: Đảm bảo cài đặt VsCode trên máy tính của mình.

- SQL server: cài đặt máy chủ SQL và tạo một cơ sở dữ liệu cho ứng dụng quản lý tuyến du lịch.

- mysql-connector-python: cài đtặ thư viện máy tính để kết nối Python với cơ sở dữ liệu SQL server.

**2. Thiết kế cơ sở dữ liệu SQL server**

- Tạo cơ sở dữ liệu trong phần mềm ứng dụng SQL SERVER để lưu trữu những thông tin từ các bảng: Tài Khoản, Nhân Viên, Khách Hàng, Chuyến Đi, Chuyến Đi – Nhân Viên, Tuyến Du Lịch, Đặt Vé và Doanh Thu.

****

Hình 10: bảng và khoá của cơ sở dữ liệu

**3. Xây dựng giao diện người dùng (GUI) bằng Tkinter**

- Sử dụng thư viện Tkinter để tạo các cửa sổ, trường nhập liệu, nút bấm và bảng hiển thị thông tin.

- Thiết kế giao diện cho phép người dùng thực hiện các chức năng:

+ Thêm

+ Xoá

+ Sửa

+ Lưu

+ Huỷ

+ Thoát

**4. Viết mã kết nối cơ sở dữ liệu**

- Sử dụng mysql.connector để thiết lập kết nối giữa Python và SQL server

- Viết các hàm:

+ Thêm: lấy thông tin dữ liệu từ các form Tkinter (nhân viên, khách hàng, tuyến du lịch…) để gắn vào treeview.

+ Xoá: xoá thông tin dữ liệu (nhân viên, khách hàng, tuyến du lịch…) khỏi treeview.

+ Sửa: cập nhật thông tin dữ liệu từ các form Tkinter (nhân viên, khách hàng, tuyến du lịch…) để lưu vào cơ sở dữ liệu SQL.

+ Huỷ: huỷ thông tin dữ liệu đang nhập vào treeview từ các form Tkinter (nhân viên, khách hàng, tuyến du lịch…).

+ Lưu: lấy thông tin dữ liệu từ các form Tkinter (nhân viên, khách hàng, tuyến du lịch…) để lưu vào cơ sở dữ liệu SQL.

+ Thoát: thoát khỏi form Tkinter hiện thời.

**5. Xây dựng giao diện Tkinter**

- Liên kết các nút bấm trên giao diện Tkinter với các hàm của Python để xử lí các tác vụ như: thêm, xoá, sửa, lưu… và cơ sở dữ liệu.

- Sử dụng các Widget như nhãn dán (label), nút (button), khung (frame), ô nhâp (entry)… để xây dựng form Tkinter.

**C. HƯỚNG DẪN CÀI ĐẶT & CODE ỨNG DỤNG**

- Ứng dụng có:

+ Giao diện xử lí (đăng nhập, tác vụ: thêm, xoá, sửa,…, xử lí tính toán và lưu dữ liệu vò cơ sở dữ liệu).

+ Bảng danh sách.

+ Dữ liệu được lưu vào cơ sở dữ liệu database SQL SERVER.

+ Giao diện được thiết kế như các hình minh hoạ trên.

**1. Cài đặt môi trường**

- Trước tiên cài đặt thư viện cần thiết:

|  |
| --- |
| pip install mysql-connector-python tkcalendar |

Giải thích:

+ mysql-connector-python: dùng để kết nối python và cơ sở dữ liệu SQL server.

+ tkcalendar: để chọn ngày, tháng năm bằng DataEntry.

**2. Chuẩn bị cơ sở dữ liệu SQL SERVER**

- Mở SQL SERVER tạo cơ sở dư liêu quản lý tuyến du lịch tương ứng gồm các bảng ( tài khoản, tuyến du lịch, nhân viên, khách hàng, chuyến đi, chuyến đi – nhân viên, đặt vé và doanh thu) cùng các ràng buộc về khá chính khoá ngoại.

|  |
| --- |
| CREATE DATABASE QuanLyTuyenDuLich;  GO  USE QuanLyTuyenDuLich;  GO  CREATE TABLE KHACHHANG (  maKH VARCHAR(6) PRIMARY KEY  CHECK (maKH LIKE 'KH[0-9][0-9][0-9][0-9]'),  hoTen NVARCHAR(50) NOT NULL,  phai NVARCHAR(3) CHECK (phai IN (N'nam', N'nữ')),  ngSinh DATE,  sdt VARCHAR(10) UNIQUE  constraint chk\_sdt\_KH check (sdt like '0[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),  dchi NVARCHAR(100)  );  CREATE TABLE NHANVIEN (  maNV VARCHAR(6) PRIMARY KEY  CHECK (maNV LIKE 'NV[0-9][0-9][0-9][0-9]'),  so\_cccd VARCHAR(12) UNIQUE,  hoTen NVARCHAR(50) NOT NULL,  chucVu NVARCHAR(20),  phai NVARCHAR(3) CHECK (phai IN (N'nam', N'nữ')),  ngSinh DATE,  sdt VARCHAR(10) UNIQUE  constraint chk\_sdt\_NV check (sdt like '0[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),  dchi NVARCHAR(100),  luong FLOAT  );  CREATE TABLE TUYENDULICH (  maTuyen VARCHAR(6) PRIMARY KEY  CHECK (maTuyen LIKE 'TD[0-9][0-9][0-9][0-9]'),  ddDi NVARCHAR(100) NOT NULL,  ddDen NVARCHAR(100) NOT NULL,  trangThai NVARCHAR(30) DEFAULT N'Hoạt động'  );  CREATE TABLE CHUYENDI (  maCD VARCHAR(6) PRIMARY KEY  CHECK (maCD LIKE 'CD[0-9][0-9][0-9][0-9]'),  ngKh DATE,  tgKh TIME,  maTuyen VARCHAR(6),  trangThai NVARCHAR(30) DEFAULT N'Hoạt động',  FOREIGN KEY (maTuyen) REFERENCES TUYENDULICH(maTuyen)  ON UPDATE CASCADE ON DELETE CASCADE  );  CREATE TABLE CHUYENDI\_NHANVIEN (  maCD VARCHAR(6),  maNV VARCHAR(6),  PRIMARY KEY (maCD, maNV),  FOREIGN KEY (maCD) REFERENCES CHUYENDI(maCD)  ON DELETE CASCADE,  FOREIGN KEY (maNV) REFERENCES NHANVIEN(maNV)  ON DELETE CASCADE  );  CREATE TABLE DATVE (  maVe VARCHAR(10) PRIMARY KEY  constraint chk\_maVe check (maVe like 'MV[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),  maKH VARCHAR(6),  maCD VARCHAR(6),  ngDat DATE,  trangThai NVARCHAR(50),  thanhTien FLOAT,  giaVe FLOAT,  soLuong INT,  FOREIGN KEY (maKH) REFERENCES KHACHHANG(maKH)  ON DELETE CASCADE,  FOREIGN KEY (maCD) REFERENCES CHUYENDI(maCD)  ON DELETE CASCADE  );  CREATE TABLE DOANHTHU (  maCD VARCHAR(6) PRIMARY KEY,  giaVe FLOAT,  tongTien FLOAT,  soLuong INT,  FOREIGN KEY (maCD) REFERENCES CHUYENDI(maCD)  ON DELETE CASCADE  );  CREATE TABLE TaiKhoan (  TenDangNhap NVARCHAR(50) PRIMARY KEY,  MatKhau NVARCHAR(50) NOT NULL,  VaiTro NVARCHAR(20) NOT NULL  );  INSERT INTO TaiKhoan VALUES  (N'quanly', N'toilaquanly', N'QuanLy'),  (N'nhanvien', N'emchaosep', N'NhanVien');  select \* from TUYENDULICH;  select \* from CHUYENDI;  select \* from CHUYENDI\_NHANVIEN;  select \* from DATVE;  select \* from KHACHHANG;  select \* from DOANHTHU;  select \* from NHANVIEN;  select \* from TaiKhoan |

**3. Code ứng dụng Tkinter và SQL server bằng ngôn ngữ Python**

**1. Code form đăng nhập**

|  |
| --- |
| import tkinter as tk  from tkinter import messagebox  import pyodbc  import subprocess  def login():      username = entry\_user.get()      password = entry\_pass.get()      conn = pyodbc.connect(              'DRIVER={SQL Server};'             'SERVER=ADMIN-PC;'             # 'SERVER=localhost\\SQLEXPRESS01;'              'DATABASE=QuanLyTuyenDuLich;'              'Trusted\_Connection=yes;'          )      cursor = conn.cursor()      cursor.execute("SELECT VaiTro FROM TaiKhoan WHERE TenDangNhap=? AND MatKhau=?", (username, password))      result = cursor.fetchone()        if result:              role = result[0]              if role == "QuanLy":                  subprocess.Popen(["python", "form\_trangChu\_QuanLy.py"])              elif role == "NhanVien":                  subprocess.Popen(["python", "form\_trangChu\_NhanVien.py"])      else:              messagebox.showerror("Lỗi", "Sai tên đăng nhập hoặc mật khẩu!")      conn.close()  def thoat():      root.destroy()  root = tk.Tk()  root.title("Login Form")  root.geometry("1000x650")  root.configure(bg="#FFFACD")  frame\_login = tk.Frame(root, bg="#FFFACD", padx=40, pady=40, relief="groove", bd=2)  frame\_login.place(relx=0.5, rely=0.5, anchor="center")  label\_title = tk.Label(frame\_login, text="Login", font=("Arial", 26, "bold"), bg="#FFFACD")  label\_title.grid(row=0, column=0, columnspan=2, pady=(0, 20))  label\_user = tk.Label(frame\_login, text="User:", font=("Arial", 16), bg="#FFFACD")  label\_user.grid(row=1, column=0, sticky="e", pady=10, padx=10)  entry\_user = tk.Entry(frame\_login, width=30, font=("Arial", 14))  entry\_user.grid(row=1, column=1, pady=10, padx=10)  label\_pass = tk.Label(frame\_login, text="Password:", font=("Arial", 16), bg="#FFFACD")  label\_pass.grid(row=2, column=0, sticky="e", pady=10, padx=10)  entry\_pass = tk.Entry(frame\_login, width=30, show="\*", font=("Arial", 14))  entry\_pass.grid(row=2, column=1, pady=10, padx=10)  def on\_enter(e):      e.widget["background"] = "#7EC8E3"  def on\_leave(e):      e.widget["background"] = "#ADD8E6"  button\_frame = tk.Frame(frame\_login, bg="#FFFACD")  button\_frame.grid(row=3, column=0, columnspan=2, pady=25)  button\_ok = tk.Button(button\_frame, text="OK", bg="#ADD8E6", width=12, height=1, font=("Arial", 14), command=login)  button\_ok.grid(row=0, column=0, padx=20)  button\_ok.bind("<Enter>", on\_enter)  button\_ok.bind("<Leave>", on\_leave)  button\_cancel = tk.Button(button\_frame, text="Cancel", bg="#ADD8E6", width=12, height=1, font=("Arial", 14), command=thoat)  button\_cancel.grid(row=0, column=1, padx=20)  button\_cancel.bind("<Enter>", on\_enter)  button\_cancel.bind("<Leave>", on\_leave)  entry\_user.focus()  root.mainloop() |

**2. Code form trang chủ quản lý**

|  |
| --- |
| import tkinter as tk  import pyodbc  import subprocess  conn = pyodbc.connect(              'DRIVER={SQL Server};'             'SERVER=ADMIN-PC;'             # 'SERVER=localhost\\SQLEXPRESS01;'              'DATABASE=QuanLyTuyenDuLich;'              'Trusted\_Connection=yes;'          )  cursor = conn.cursor()  def open\_form\_KhachHang():      subprocess.Popen(["python", "form\_KhachHang.py"])  def open\_form\_NhanVien():      subprocess.Popen(["python", "form\_NhanVien.py"])  def open\_form\_ChuyenDi():      subprocess.Popen(["python", "form\_ChuyenDi.py"])  def open\_form\_TuyenDuLich():      subprocess.Popen(["python", "form\_TuyenDuLich.py"])  def open\_form\_DatVe():      subprocess.Popen(["python", "form\_DatVe.py"])  def open\_form\_DoanhThu():      subprocess.Popen(["python", "form\_DoanhThu.py"])  def thoat():      root.destroy()  root = tk.Tk()  root.title("Trang Chủ - Quản Lý")  root.geometry("1000x650")  root.configure(bg="#FFFACD")  frame = tk.Frame(root, bg="#FFFACD", padx=40, pady=40, relief="groove", bd=2)  frame.place(relx=0.5, rely=0.5, anchor="center")  tk.Label(frame, text="Trang Chủ", font=("Arial", 26, "bold"), bg="#FFFACD").grid(row=0, column=0, columnspan=2, pady=20)  tk.Button(frame, text="Quản Lý Khách Hàng", bg="#ADD8E6", font=("Arial", 14), width=20, height=2, command=open\_form\_KhachHang).grid(row=1, column=0, padx=30, pady=20)  tk.Button(frame, text="Quản Lý Nhân Viên", bg="#ADD8E6", font=("Arial", 14), width=20, height=2, command=open\_form\_NhanVien).grid(row=1, column=1, padx=30, pady=20)  tk.Button(frame, text="Quản Lý Chuyến Đi", bg="#ADD8E6", font=("Arial", 14), width=20, height=2, command= open\_form\_ChuyenDi).grid(row=2, column=0, padx=30, pady=20)  tk.Button(frame, text="Quản Lý Tuyến Du Lịch", bg="#ADD8E6", font=("Arial", 14), width=20, height=2, command=open\_form\_TuyenDuLich).grid(row=2, column=1, padx=30, pady=20)  tk.Button(frame, text="Quản Lý Đặt Vé", bg="#ADD8E6", font=("Arial", 14), width=20, height=2, command= open\_form\_DatVe).grid(row=3, column=0, padx=30, pady=20)  tk.Button(frame, text="Quản Lý Doanh Thu", bg="#ADD8E6", font=("Arial", 14), width=20, height=2, command= open\_form\_DoanhThu).grid(row=3, column=1, padx=30, pady=20)  tk.Button(root, text="Thoát", font=("Arial", 12), command=thoat).place(x=920, y=600)  root.mainloop() |

**3. Code form trang chủ nhân viên**

|  |
| --- |
| import tkinter as tk  import pyodbc  import subprocess  conn = pyodbc.connect(              'DRIVER={SQL Server};'             'SERVER=ADMIN-PC;'             # 'SERVER=localhost\\SQLEXPRESS01;'              'DATABASE=QuanLyTuyenDuLich;'              'Trusted\_Connection=yes;'          )  cursor = conn.cursor()  def open\_form\_KhachHang():      subprocess.Popen(["python", "form\_KhachHang.py"])  def open\_form\_ChuyenDi():      subprocess.Popen(["python", "form\_NhanVien.py"])  def open\_form\_DatVe():      subprocess.Popen(["python", "form\_DatVe.py"])  def open\_form\_TuyenDuLich():      subprocess.Popen(["python", "form\_TuyenDuLich.py"])  def thoat():      root.destroy()  root = tk.Tk()  root.title("Trang Chủ - Nhân Viên")  root.geometry("1000x650")  root.configure(bg="#FFFACD")  frame = tk.Frame(root, bg="#FFFACD", padx=40, pady=40, relief="groove", bd=2)  frame.place(relx=0.5, rely=0.5, anchor="center")  tk.Label(frame, text="Trang Chủ", font=("Arial", 26, "bold"), bg="#FFFACD").grid(row=0, column=0, columnspan=2, pady=20)  tk.Button(frame, text="Quản Lý Khách Hàng", bg="#ADD8E6", font=("Arial", 14), width=20, height=2, command=open\_form\_KhachHang).grid(row=1, column=0, padx=30, pady=20)  tk.Button(frame, text="Quản Lý Chuyến Đi", bg="#ADD8E6", font=("Arial", 14), width=20, height=2, command=open\_form\_ChuyenDi).grid(row=1, column=1, padx=30, pady=20)  tk.Button(frame, text="Quản Lý Đặt Vé", bg="#ADD8E6", font=("Arial", 14), width=20, height=2, command=open\_form\_DatVe).grid(row=2, column=0, padx=30, pady=20)  tk.Button(frame, text="Quản Lý Tuyến Du Lịch", bg="#ADD8E6", font=("Arial", 14), width=20, height=2, command=open\_form\_TuyenDuLich).grid(row=2, column=1, padx=30, pady=20)  tk.Button(root, text="Thoát", font=("Arial", 12), command=thoat).place(x=920, y=600)  root.mainloop() |

**4. Code form quản lý tuyến du lịch**

|  |
| --- |
| import tkinter as tk  from tkinter import ttk, messagebox  import pyodbc  conn = pyodbc.connect(      'DRIVER={SQL Server};'      #'SERVER=ADMIN-PC;'      'SERVER=localhost\\SQLEXPRESS01;'      'DATABASE=QuanLyTuyenDuLich;'      'Trusted\_Connection=yes;'  )  cursor = conn.cursor()  list\_di = ["Hà Nội", "Đà Nẵng", "Hồ Chí Minh"]  mien\_bac = ["Tràng An-Tam Cốc-Bái Đính-Hoa Lư", "Mộc Châu-Điện Biên-Sapa",              "Hà Giang-Cao Bằng", "Ninh Bình-Hạ Long-Sapa", "Vịnh Hạ Long-Hạ Long Park-Bãi Cháy-Yên Tử"]  mien\_trung = ["Quy Nhơn-Phú Yên", "Pleiku-Kontum- Măng Đen",                "Hội An-Bà Nà Hills-Núi Thần Tài", "Phan Thiết-Mũi Né", "Quảng Bình-Suối Moọc- Động Thiên Đường"]  mien\_nam = ["Tiền Giang-Bến Tre- An Giang-Cần Thơ", "Vũng Tàu", "Phú Quốc", "Tây Ninh"]  root = tk.Tk()  root.title("Quản Lý Tuyến Du Lịch")  root.geometry("1000x600")  root.configure(bg="#FFFACD")  frame = tk.LabelFrame(root, text="Danh sách tuyến du lịch", font=("Arial", 12, "bold"), bg="#FFFACD", width=850, height=280)  frame.place(x=70, y=270)  columns = ("maTuyen", "ddDi", "ddDen","tt")  tree = ttk.Treeview(frame, columns=columns, show="headings", height=12)  tree.heading("maTuyen", text="Mã Tuyến")  tree.heading("ddDi", text="Địa Điểm Đi")  tree.heading("ddDen", text="Địa Điểm Đến")  tree.heading("tt", text="Trạng Thái")  tree.column("maTuyen", width=150, anchor="center")  tree.column("ddDi", width=150, anchor="center")  tree.column("ddDen", width=300, anchor="center")  tree.column("tt", width=100, anchor ="center")  scrollbar = ttk.Scrollbar(frame, orient="vertical", command=tree.yview)  tree.configure(yscrollcommand=scrollbar.set)  tree.pack(side="left", fill="both", expand=True, padx=10, pady=10)  scrollbar.pack(side="right", fill="y")  def auto\_maTuyen():      ma\_list\_tree = [tree.item(item)['values'][0].strip() for item in tree.get\_children()]      cursor.execute("SELECT maTuyen FROM TUYENDULICH")      ma\_list\_sql = [row[0].strip() for row in cursor.fetchall()]      # Gộp danh sách      ma\_list = list(set(ma\_list\_tree + ma\_list\_sql))      if not ma\_list:          return "TD0001"        so\_list = sorted([int(ma[2:]) for ma in ma\_list if ma[2:].isdigit()])      next\_num = so\_list[-1] + 1      return f"TD{next\_num:04d}"  tk.Label(root, text="Quản Lý Tuyến Du Lịch", font=("Arial", 20, "bold"), bg="#FFFACD").place(x=330, y=40)  tk.Label(root, text="Mã Tuyến:", font=("Arial", 12), bg="#FFFACD").place(x=200, y=130)  entry\_matuyen = tk.Entry(root, width=50)  entry\_matuyen.place(x=350, y=130)  entry\_matuyen.config(state="readonly")  tk.Label(root, text="Địa Điểm Đi:", font=("Arial", 12), bg="#FFFACD").place(x=200, y=170)  entry\_ddDi = ttk.Combobox(root, width=48, values=list\_di, state="normal")  entry\_ddDi.place(x=350, y=170)  entry\_ddDi.set("")  tk.Label(root, text="Địa Điểm Đến:", font=("Arial", 12), bg="#FFFACD").place(x=200, y=210)  entry\_ddDen = ttk.Combobox(root, width=48, values=[], state="normal")  entry\_ddDen.place(x=350, y=210)  entry\_ddDen.set("")  def cap\_nhat\_ddDen(event=None):      di = entry\_ddDi.get()      if di == "Hà Nội":          den\_list = mien\_bac.copy()      elif di == "Đà Nẵng":          den\_list = mien\_trung.copy()      elif di == "Hồ Chí Minh":          den\_list = mien\_nam.copy()      else:          den\_list = []      cursor.execute("SELECT ddDen FROM TUYENDULICH WHERE ddDi=? AND trangThai=N'Hoạt động'", (di,))      csdl\_den = [row[0] for row in cursor.fetchall()]      tree\_den = [tree.item(item, "values")[2] for item in tree.get\_children() if tree.item(item, "values")[1] == di]      den\_list = [d for d in den\_list if d not in csdl\_den and d not in tree\_den]      entry\_ddDen['values'] = den\_list      entry\_ddDen.set("")  entry\_ddDi.bind("<<ComboboxSelected>>", cap\_nhat\_ddDen)  def load\_data():      for item in tree.get\_children():          tree.delete(item)      cursor.execute("SELECT maTuyen, ddDi, ddDen, trangThai FROM TUYENDULICH where trangThai=N'Hoạt động'")      rows = cursor.fetchall()      for row in rows:          tree.insert("", "end", values=(row[0].strip(), row[1].strip(), row[2].strip(), row[3].strip()))  def lam\_moi\_form():      entry\_ddDi.set("")      entry\_ddDen.set("")      entry\_matuyen.config(state='normal')      entry\_matuyen.delete(0, tk.END)      entry\_matuyen.insert(0, auto\_maTuyen())      entry\_matuyen.config(state='readonly')  def them():      ma = auto\_maTuyen()      di = entry\_ddDi.get()      den = entry\_ddDen.get()      if not di or not den:          messagebox.showwarning("Thiếu thông tin", "Vui lòng chọn địa điểm đi và đến")          return      tree.insert("", "end", values=(ma, di, den, "Hoạt động"))      lam\_moi\_form()  def xoa():      selected = tree.selection()      if not selected:          messagebox.showwarning("Chưa chọn", "Vui lòng chọn tuyến để xóa!")          return        confirm = messagebox.askyesno("Xác nhận", "Bạn có chắc muốn xóa tuyến này không?")      if confirm:          for item in selected:              ma = tree.item(item, "values")[0]              tree.delete(item)              cursor.execute("UPDATE TUYENDULICH SET trangThai=N'Đã xóa' WHERE maTuyen=?", (ma,))            conn.commit()          messagebox.showinfo("Thành công", "Đã xóa tuyến du lịch!")          lam\_moi\_form()          load\_data()  def sua():      selected = tree.selection()      if not selected:          messagebox.showwarning("Chưa chọn", "Vui lòng chọn tuyến để sửa!")          return      maTuyen = tree.item(selected[0])['values'][0].strip()      ddDi = entry\_ddDi.get().strip()      ddDen = entry\_ddDen.get().strip()      if not ddDi or not ddDen:          messagebox.showwarning("Thiếu dữ liệu", "Vui lòng chọn đầy đủ địa điểm đi và đến!")          return      tree.item(selected[0], values=(maTuyen, ddDi, ddDen, "Hoạt động"))      cursor.execute(          "UPDATE TUYENDULICH SET ddDi=?, ddDen=?, trangThai=N'Hoạt động' WHERE maTuyen=?",          (ddDi, ddDen, maTuyen)      )      conn.commit()      messagebox.showinfo("Thành công", "Đã cập nhật thông tin tuyến du lịch!")      lam\_moi\_form()  def luu():      try:          for item in tree.get\_children():              maTuyen, ddDi, ddDen, tt = tree.item(item, "values")              cursor.execute("""                  IF NOT EXISTS (SELECT 1 FROM TUYENDULICH WHERE maTuyen=?)                      INSERT INTO TUYENDULICH (maTuyen, ddDi, ddDen, trangThai)                      VALUES (?, ?, ?, N'Hoạt động')                  ELSE                      UPDATE TUYENDULICH                      SET ddDi=?, ddDen=?, trangThai=N'Hoạt động'                      WHERE maTuyen=?              """, (maTuyen, maTuyen, ddDi, ddDen, ddDi, ddDen, maTuyen))            conn.commit()          messagebox.showinfo("Thành công", "Đã lưu và đồng bộ dữ liệu với CSDL!")          load\_data()          lam\_moi\_form()      except Exception as e:          messagebox.showerror("Lỗi", f"Lưu dữ liệu thất bại!\n{e}")  def huy():      lam\_moi\_form()  def thoat():      root.destroy()  button\_style = {"font": ("Times New Roman", 13, "bold"), "bg": "#B0E0E6", "width": 8, "height": 1}  btn\_them = tk.Button(root, text="Thêm", \*\*button\_style, command=them)  btn\_them.place(x=870, y=120)  btn\_xoa = tk.Button(root, text="Xóa", \*\*button\_style, command=xoa)  btn\_xoa.place(x=870, y=200)  btn\_sua = tk.Button(root, text="Sửa", \*\*button\_style, command=sua)  btn\_sua.place(x=870, y=280)  btn\_huy = tk.Button(root, text="Hủy", \*\*button\_style, command=huy)  btn\_huy.place(x=870, y=360)  btn\_luu = tk.Button(root, text="Lưu", \*\*button\_style, command=luu)  btn\_luu.place(x=870, y=440)  btn\_thoat = tk.Button(root, text="Thoát", \*\*button\_style, command=thoat)  btn\_thoat.place(x = 870, y=520)  lam\_moi\_form()  load\_data()  root.mainloop() |

**5. Code form quản lý nhân viên**

|  |
| --- |
| import tkinter as tk  from tkinter import ttk, messagebox  from tkcalendar import DateEntry  from datetime import date, datetime  import pyodbc  import re  conn = pyodbc.connect(      'DRIVER={SQL Server};'      'SERVER=localhost\\SQLEXPRESS01;'      # 'SERVER=ADMIN-PC;'      'DATABASE=QuanLyTuyenDuLich;'      'Trusted\_Connection=yes;'  )  cursor = conn.cursor()  root = tk.Tk()  root.title("Quản Lý Nhân Viên")  root.geometry("1000x650")  root.configure(bg="#FFFACD")  deleted\_items = []  reusable\_ids = []  def auto\_maNV():      ma\_list\_tree = [tree.item(item)['values'][0].strip() for item in tree.get\_children()]      cursor.execute("SELECT maNV FROM NHANVIEN")      ma\_list\_sql = [row[0].strip() for row in cursor.fetchall()]        # Gộp mã      ma\_list = list(set(ma\_list\_tree + ma\_list\_sql))        if not ma\_list:          return "NV0001"        so\_list = sorted([int(ma[2:]) for ma in ma\_list if ma[2:].isdigit()])      next\_num = so\_list[-1] + 1      return f"NV{next\_num:04d}"  def load\_nam():      # lấy năm từ danh sách chuyến đi      cursor.execute("SELECT DISTINCT YEAR(ngKh) FROM CHUYENDI ORDER BY YEAR(ngKh)")      nam\_list = [str(r[0]) for r in cursor.fetchall()]      combo\_nam['values'] = nam\_list      if nam\_list:          combo\_nam.set(nam\_list[0])      else:          combo\_nam.set(str(date.today().year))  def lam\_moi\_form():      tree.selection\_remove(tree.selection())      entry\_maNV.config(state='normal')      entry\_maNV.delete(0, tk.END)      entry\_maNV.insert(0, auto\_maNV())      entry\_maNV.config(state='readonly')      entry\_socccd.delete(0, tk.END)      entry\_hoTen.delete(0, tk.END)      entry\_sdt.delete(0, tk.END)      combo\_phai.set("Chọn giới tính")      date\_ngsinh.set\_date(date.today())      entry\_dchi.delete(0, tk.END)      combo\_chucvu.set("Chọn chức vụ")  # Ẩn cột lương  def toggle\_luong\_visibility(show=True):      if show:          tree["displaycolumns"] = ("maNV","so\_cccd","hoTen","sdt","phai","ngsinh","dchi","chucVu","soChuyen","luong")      else:          tree["displaycolumns"] = ("maNV","so\_cccd","hoTen","sdt","phai","ngsinh","dchi","chucVu","soChuyen")  tree\_frame = tk.LabelFrame(root, text="Danh sách nhân viên", font=("Times New Roman",12), bg="#fff8dc", width=900, height=400)  tree\_frame.place(x=50, y=280)  columns = ("maNV","so\_cccd","hoTen","sdt","phai","ngsinh","dchi","chucVu","soChuyen","luong")  tree = ttk.Treeview(tree\_frame, columns=columns, show="headings", height=15)  for col, text, width in zip(columns,                              ["Mã NV","Số CCCD","Họ Tên","SĐT","Giới Tính","Ngày Sinh","Địa Chỉ","Chức Vụ","Số Chuyến","Lương Thực Lãnh"],                              [80,100,150,100,60,100,150,100,80,120]):      tree.heading(col, text=text)      tree.column(col, width=width, anchor="center")  scrollbar\_v = ttk.Scrollbar(tree\_frame, orient="vertical", command=tree.yview) # thanh trượt dọc  scrollbar\_h = ttk.Scrollbar(tree\_frame, orient="horizontal", command=tree.xview) # thanh trươtcj ngang  tree.configure(yscrollcommand=scrollbar\_v.set, xscrollcommand=scrollbar\_h.set)  tree.grid(row=0, column=0, sticky="nsew", padx=(5,0), pady=(5,0))  scrollbar\_v.grid(row=0, column=1, sticky="ns", pady=(5,0))  scrollbar\_h.grid(row=1, column=0, sticky="ew", padx=(5,0))  tree\_frame.grid\_rowconfigure(0, weight=1)  tree\_frame.grid\_columnconfigure(0, weight=1)  toggle\_luong\_visibility(False)  def load\_data():      tree.delete(\*tree.get\_children())      cursor.execute("SELECT maNV, so\_cccd, hoTen, sdt, phai, ngSinh, dchi, chucVu FROM NHANVIEN")      nhanviens = cursor.fetchall()      cursor.execute("""          SELECT cn.maNV, COUNT(\*)          FROM CHUYENDI\_NHANVIEN cn          JOIN CHUYENDI c ON cn.maCD = c.maCD          GROUP BY cn.maNV      """)      chuyens = dict(cursor.fetchall())      for nv in nhanviens:          maNV, so\_cccd, hoTen, sdt, phai, ngSinh, dchi, chucVu = nv          soChuyen = chuyens.get(maNV, 0)          if chucVu == "Cơ Trưởng":              luong\_cb = 2000000          elif chucVu == "Hướng Dẫn Viên":              luong\_cb = 1800000          else:              luong\_cb = 1200000          luong\_thuc = luong\_cb \* soChuyen          tree.insert("", "end", values=(              str(maNV),              str(so\_cccd).zfill(12),              str(hoTen),              str(sdt).zfill(10),              str(phai),              str(ngSinh),              str(dchi),              str(chucVu),              str(soChuyen),              str(luong\_thuc)          ))  def xem\_luong():      thang\_str = combo\_thang.get()      nam\_str = combo\_nam.get()      if not thang\_str or not nam\_str:          messagebox.showwarning("Thiếu thông tin", "Vui lòng chọn tháng và năm!")          return      thang = int(thang\_str)      nam = int(nam\_str)      tree.delete(\*tree.get\_children())      cursor.execute("SELECT maNV, so\_cccd, hoTen, sdt, phai, ngSinh, dchi, chucVu FROM NHANVIEN")      nhanviens = cursor.fetchall()      cursor.execute("""          SELECT cn.maNV, COUNT(\*)          FROM CHUYENDI\_NHANVIEN cn          JOIN CHUYENDI c ON cn.maCD = c.maCD          WHERE MONTH(c.ngKh)=? AND YEAR(c.ngKh)=?          GROUP BY cn.maNV      """, (thang, nam))      chuyens = dict(cursor.fetchall())      for nv in nhanviens:          maNV, so\_cccd, hoTen, sdt, phai, ngSinh, dchi, chucVu = nv          soChuyen = chuyens.get(maNV,0)          if chucVu == "Cơ Trưởng":              luong\_cb = 2000000          elif chucVu == "Hướng Dẫn Viên":              luong\_cb = 1800000          else:              luong\_cb = 1200000          luong\_thuc = luong\_cb \* soChuyen          tree.insert("", "end", values=(              str(maNV),              str(so\_cccd).zfill(12),              str(hoTen),              str(sdt).zfill(10),              str(phai),              str(ngSinh),              str(dchi),              str(chucVu),              str(soChuyen),              str(luong\_thuc)          ))      toggle\_luong\_visibility(True)  def kiem\_tra\_cccd(so\_cccd: str) -> bool:      return bool(re.fullmatch(r"0\d{11}", so\_cccd))  def them():      maNV = auto\_maNV()      so\_cccd = entry\_socccd.get().strip()      hoTen = entry\_hoTen.get()      sdt = entry\_sdt.get().strip()      phai = combo\_phai.get()      ngSinh = date\_ngsinh.get\_date().strftime('%Y-%m-%d')      dchi = entry\_dchi.get()      chucVu = combo\_chucvu.get()      if not kiem\_tra\_cccd(so\_cccd):          messagebox.showerror("Lỗi","Số CCCD phải 12 số bắt đầu 0")          return      if not (hoTen and sdt and chucVu):          messagebox.showwarning("Thiếu dữ liệu","Nhập đầy đủ thông tin!")          return      # Kiểm tra trùng CCCD trên Treeview      for item in tree.get\_children():          cccd\_in\_tree = str(tree.item(item)['values'][1]).zfill(12)          if cccd\_in\_tree == so\_cccd:              messagebox.showerror("Lỗi","Số CCCD đã tồn tại!")              return      # Kiểm tra trùng SĐT trên Treeview      for item in tree.get\_children():          sdt\_in\_tree = str(tree.item(item)['values'][3]).zfill(10)          if sdt\_in\_tree == sdt:              messagebox.showerror("Lỗi","Số điện thoại đã tồn tại!")              return      tree.insert("", "end", values=(maNV, so\_cccd, hoTen, sdt, phai, ngSinh, dchi, chucVu,0,0))      lam\_moi\_form()  def xoa():      selected = tree.selection()      if not selected:          messagebox.showwarning("Chưa chọn","Chọn nhân viên để xóa!")          return      confirm = messagebox.askyesno("Xác nhận","Bạn có chắc muốn xóa?")      if confirm:          maNV = tree.item(selected[0])['values'][0]          deleted\_items.append(maNV)          reusable\_ids.append(maNV)          tree.delete(selected[0])          lam\_moi\_form()  def chon\_nhan\_vien(event):      selected = tree.selection()      if not selected:          return      item = tree.item(selected[0])['values']      if not item:          return      entry\_maNV.config(state='normal')      entry\_maNV.delete(0, tk.END)      entry\_maNV.insert(0, str(item[0]))      entry\_maNV.config(state='readonly')      entry\_socccd.delete(0, tk.END)      entry\_socccd.insert(0, str(item[1]).zfill(12))      entry\_hoTen.delete(0, tk.END)      entry\_hoTen.insert(0, str(item[2]))      entry\_sdt.delete(0, tk.END)      entry\_sdt.insert(0, str(item[3]).zfill(10))      combo\_phai.set(str(item[4]))      date\_ngsinh.set\_date(datetime.strptime(str(item[5]), '%Y-%m-%d').date())      entry\_dchi.delete(0, tk.END)      entry\_dchi.insert(0, str(item[6]))      combo\_chucvu.set(str(item[7]))  tree.bind("<<TreeviewSelect>>", chon\_nhan\_vien)  def sua():      selected = tree.selection()      if not selected:          messagebox.showwarning("Chưa chọn","Chọn nhân viên để sửa!")          return        maNV = tree.item(selected[0])['values'][0]      so\_cccd = entry\_socccd.get().strip().zfill(12)      sdt = entry\_sdt.get().strip().zfill(10)      # Kiểm tra CCCD hợp lệ và trùng      if not kiem\_tra\_cccd(so\_cccd):          messagebox.showerror("Lỗi","CCCD phải 12 số bắt đầu 0")          return      for item in tree.get\_children():          if tree.item(item)['values'][1] == so\_cccd and tree.item(item)['values'][0] != maNV:              messagebox.showerror("Lỗi","CCCD đã tồn tại!")              return      hoTen = entry\_hoTen.get()      sdt = entry\_sdt.get()      phai = combo\_phai.get()      ngSinh = date\_ngsinh.get\_date().strftime('%Y-%m-%d')      dchi = entry\_dchi.get()      chucVu = combo\_chucvu.get()      tree.item(selected[0], values=(maNV, so\_cccd, hoTen, sdt, phai, ngSinh, dchi, chucVu, 0, 0))      lam\_moi\_form()  def luu():      for item in tree.get\_children():          maNV, so\_cccd, hoTen, sdt, phai, ngSinh, dchi, chucVu, \_, \_ = tree.item(item,"values")          cursor.execute("SELECT 1 FROM NHANVIEN WHERE maNV=?", (maNV,))          if cursor.fetchone():              cursor.execute("""                  UPDATE NHANVIEN SET so\_cccd=?, hoTen=?, sdt=?, phai=?, ngSinh=?, dchi=?, chucVu=?                  WHERE maNV=?              """,(so\_cccd, hoTen, sdt, phai, ngSinh, dchi, chucVu, maNV))          else:              cursor.execute("""                  INSERT INTO NHANVIEN(maNV,so\_cccd,hoTen,sdt,phai,ngSinh,dchi,chucVu)                  VALUES(?,?,?,?,?,?,?,?)              """,(maNV, so\_cccd, hoTen, sdt, phai, ngSinh, dchi, chucVu))      for maNV in deleted\_items:          cursor.execute("DELETE FROM NHANVIEN WHERE maNV=?", (maNV,))      deleted\_items.clear()      conn.commit()      load\_data()      lam\_moi\_form()      messagebox.showinfo("Thành công","Đã lưu vào CSDL!")  def huy():      lam\_moi\_form()  def thoat():      root.destroy()  tk.Label(root, text="Quản Lý Nhân Viên", font=("Arial",20,"bold"), bg="#FFFACD").place(x=330,y=20)  form\_frame = tk.Frame(root,bg="#FFFACD")  form\_frame.place(x=80,y=60)  tk.Label(form\_frame,text="Mã NV:", bg="#FFFACD").grid(row=0,column=0,padx=5,pady=5,sticky="w")  entry\_maNV = tk.Entry(form\_frame,width=25)  entry\_maNV.grid(row=0,column=1,padx=5,pady=5)  tk.Label(form\_frame,text="Số CCCD:", bg="#FFFACD").grid(row=0,column=2,padx=5,pady=5,sticky="w")  entry\_socccd = tk.Entry(form\_frame,width=25)  entry\_socccd.grid(row=0,column=3,padx=5,pady=5)  tk.Label(form\_frame,text="Họ tên:", bg="#FFFACD").grid(row=0,column=4,padx=5,pady=5,sticky="w")  entry\_hoTen = tk.Entry(form\_frame,width=25)  entry\_hoTen.grid(row=0,column=5,padx=5,pady=5)  tk.Label(form\_frame,text="SĐT:", bg="#FFFACD").grid(row=1,column=0,padx=5,pady=5,sticky="w")  entry\_sdt = tk.Entry(form\_frame,width=25)  entry\_sdt.grid(row=1,column=1,padx=5,pady=5)  tk.Label(form\_frame,text="Giới tính:", bg="#FFFACD").grid(row=1,column=2,padx=5,pady=5,sticky="w")  combo\_phai = ttk.Combobox(form\_frame,width=22,state="readonly",values=["Nam","Nữ"])  combo\_phai.set("Chọn giới tính")  combo\_phai.grid(row=1,column=3,padx=5,pady=5)  tk.Label(form\_frame,text="Ngày sinh:", bg="#FFFACD").grid(row=2,column=0,padx=5,pady=5,sticky="w")  date\_ngsinh = DateEntry(form\_frame,width=23,date\_pattern="dd/mm/yyyy")  date\_ngsinh.grid(row=2,column=1,padx=5,pady=5)  tk.Label(form\_frame,text="Địa chỉ:", bg="#FFFACD").grid(row=2,column=2,padx=5,pady=5,sticky="w")  entry\_dchi = tk.Entry(form\_frame,width=25)  entry\_dchi.grid(row=2,column=3,padx=5,pady=5)  tk.Label(form\_frame,text="Chức vụ:", bg="#FFFACD").grid(row=3,column=0,padx=5,pady=5,sticky="w")  combo\_chucvu = ttk.Combobox(form\_frame,width=22,state="readonly",values=["Cơ Trưởng","Hướng Dẫn Viên","Nhân Viên"])  combo\_chucvu.set("Chọn chức vụ")  combo\_chucvu.grid(row=3,column=1,padx=5,pady=5)  tk.Label(form\_frame,text="Tháng:", bg="#FFFACD").grid(row=3,column=2,padx=5,pady=5,sticky="w")  combo\_thang = ttk.Combobox(form\_frame,width=5,state="readonly",values=[str(i) for i in range(1,13)])  combo\_thang.set(str(date.today().month))  combo\_thang.grid(row=3,column=2,padx=(60,0),sticky="w")  tk.Label(form\_frame,text="Năm:", bg="#FFFACD").grid(row=3,column=3,padx=5,pady=5,sticky="w")  combo\_nam = ttk.Combobox(form\_frame,width=7,state="readonly")  combo\_nam.grid(row=3,column=3,padx=(45,0),sticky="w")  btn\_xemluong = tk.Button(form\_frame,text="👁 Xem lương",bg="#ADD8E6",font=("Arial",10,"bold"),command=xem\_luong)  btn\_xemluong.grid(row=3,column=4,padx=10,pady=5)  btn\_them = tk.Button(root,text="Thêm",bg="#87cefa",font=("Arial",12,"bold"),width=10,command=them)  btn\_them.place(x=100,y=235)  btn\_sua = tk.Button(root,text="Sửa",bg="#87cefa",font=("Arial",12,"bold"),width=10,command=sua)  btn\_sua.place(x=240,y=235)  btn\_xoa = tk.Button(root,text="Xóa",bg="#87cefa",font=("Arial",12,"bold"),width=10,command=xoa)  btn\_xoa.place(x=380,y=235)  btn\_huy = tk.Button(root,text="Hủy",bg="#87cefa",font=("Arial",12,"bold"),width=10,command=huy)  btn\_huy.place(x=520,y=235)  btn\_luu = tk.Button(root,text="Lưu",bg="#87cefa",font=("Arial",12,"bold"),width=10,command=luu)  btn\_luu.place(x=660,y=235)  btn\_thoat = tk.Button(root,text="Thoát",bg="#87cefa",font=("Arial",12,"bold"),width=10,command=thoat)  btn\_thoat.place(x=800,y=235)  load\_nam()  load\_data()  lam\_moi\_form()  root.mainloop() |

**6. Code form quản lý khách hàng**

|  |
| --- |
| import tkinter as tk  from tkinter import ttk, messagebox  from tkcalendar import DateEntry  from datetime import date  import pyodbc  conn = pyodbc.connect(      'DRIVER={SQL Server};'     #'SERVER=ADMIN-PC;'      'SERVER=localhost\\SQLEXPRESS01;'      'DATABASE=QuanLyTuyenDuLich;'      'Trusted\_Connection=yes;'  )  cursor = conn.cursor()  root = tk.Tk()  root.title("Quản Lý Khách Hàng")  root.geometry("1000x650")  root.configure(bg="#FFFACD")  frame\_tree = tk.LabelFrame(root, text="Danh sách khách hàng", font=("Times New Roman", 12), bg="#fff8dc", width=800, height=400)  frame\_tree.place(x=50, y=200)  columns = ("maKh", "hoTen", "sdt", "phai", "ngsinh", "dchi")  tree = ttk.Treeview(frame\_tree, columns=columns, show="headings")  tree.heading("maKh", text="Mã Khách Hàng")  tree.heading("hoTen", text="Họ tên")  tree.heading("sdt", text="Số điện thoại")  tree.heading("phai", text="Giới Tính")  tree.heading("ngsinh", text="Ngày Sinh")  tree.heading("dchi", text="Địa Chỉ")  tree.column("maKh", width = 100, anchor="center")  tree.column("hoTen", width = 150)  tree.column("sdt", width = 120, anchor="center")  tree.column("phai", width = 80, anchor="center")  tree.column("ngsinh", width = 100, anchor="center")  tree.column("dchi", width = 250)  scrollbar\_v = ttk.Scrollbar(frame\_tree, orient="vertical", command=tree.yview) #dọc  scrollbar\_h = ttk.Scrollbar(frame\_tree, orient="horizontal", command=tree.xview) #ngang  tree.configure(yscrollcommand=scrollbar\_v.set, xscrollcommand=scrollbar\_h.set)  tree.place(x=10, y=10, width=760, height=340)  scrollbar\_v.place(x=770, y=10, width=20, height=360)  scrollbar\_h.place(x=10, y=350, width=750, height=20)  def load\_data():      for i in tree.get\_children():          tree.delete(i)      cursor.execute("SELECT maKH, hoTen, sdt, phai, ngsinh, dchi FROM KHACHHANG")      rows = cursor.fetchall()        for row in rows:          tree.insert("", "end", values=(row[0].strip(), row[1].strip(), row[2], row[3], row[4], row[5].strip()))  def auto\_maKh():      if reusable\_ids:          return reusable\_ids.pop(0)      ma\_list = [tree.item(item)['values'][0].strip() for item in tree.get\_children()]        if not ma\_list:          return "KH0001"        so\_list = sorted([int(ma[2:]) for ma in ma\_list if ma[2:].isdigit()])      next\_num = 1      for num in so\_list:          if num == next\_num:              next\_num += 1          elif num > next\_num:              break      return f"KH{next\_num:04d}"  def lam\_moi\_form():      entry\_maKH.config(state='normal')      entry\_maKH.delete(0, tk.END)      entry\_maKH.insert(0, auto\_maKh())      entry\_maKH.config(state='readonly')      entry\_ten.delete(0, tk.END)      entry\_sdt.delete(0, tk.END)      entry\_diachi.delete(0, tk.END)      gender.set("Nam")      date\_ngaysinh.set\_date(date.today())  def them():      maKH = auto\_maKh()      hoTen = entry\_ten.get().strip()      sdt = entry\_sdt.get()      phai = gender.get()      ngsinh = date\_ngaysinh.get\_date()      dchi = entry\_diachi.get().strip()      if not (hoTen or sdt):          messagebox.showwarning("Thiếu dữ liệu", "Vui lòng nhập đầy đủ thông tin!")          return      if not (sdt.isdigit() and len(sdt) == 10 and sdt.startswith("0")):          messagebox.showerror("Lỗi", "Số điện thoại phải bắt đầu bằng 0 và đủ 10 số!")          return      for item in tree.get\_children():          sdt\_in\_tree = str(tree.item(item)['values'][2]).zfill(10)          if sdt\_in\_tree == sdt:              messagebox.showerror("Lỗi","Số điện thoại đã tồn tại!")              return      tree.insert("", "end", values=(maKH, hoTen, sdt, phai, ngsinh, dchi))      messagebox.showinfo("Thành công", f"Đã thêm khách hàng {hoTen} với mã {maKH}")      lam\_moi\_form()  deleted\_items = []  # danh sách toàn cục  reusable\_ids = [] # cắt mã đã bị xoá  def xoa():      selected = tree.selection()      if not selected:          messagebox.showwarning("Chưa chọn", "Vui lòng chọn khách hàng để xóa!")          return        confirm = messagebox.askyesno("Xác nhận", "Bạn có chắc muốn xóa?")      if confirm:          maKH = tree.item(selected[0])['values'][0]          deleted\_items.append(maKH)          tree.delete(selected[0])          lam\_moi\_form()  def sua():      selected = tree.selection()      if not selected:          messagebox.showwarning("Chưa chọn", "Vui lòng chọn khách hàng để sửa!")          return      maKH = tree.item(selected[0])['values'][0].strip()      hoTen = entry\_ten.get().strip()      sdt = entry\_sdt.get().strip()      phai = gender.get()      ngsinh = date\_ngaysinh.get\_date()      dchi = entry\_diachi.get().strip()      messagebox.showinfo("Thành công","Đã cập nhật thông tin khách hàng")      tree.item(selected[0], values=(maKH, hoTen, sdt, phai, ngsinh, dchi))  def luu():      for item in tree.get\_children():          maKH, hoTen, sdt, phai, ngsinh, dchi = tree.item(item, "values")          if not (sdt.isdigit() and len(sdt) == 10 and sdt.startswith("0")):              messagebox.showwarning("Bỏ qua", f"SĐT {sdt} của {hoTen} không hợp lệ, không lưu vào CSDL!")              continue          cursor.execute("SELECT 1 FROM KHACHHANG WHERE maKH=?", (maKH,))          if cursor.fetchone():              cursor.execute(                  "UPDATE KHACHHANG SET hoTen=?, sdt=?, phai=?, ngsinh=?, dchi=? WHERE maKH=?",                  (hoTen, sdt, phai, ngsinh, dchi, maKH)              )          else:              cursor.execute(                  "INSERT INTO KHACHHANG (maKH, hoTen, sdt, phai, ngsinh, dchi) VALUES (?, ?, ?, ?, ?, ?)",                  (maKH, hoTen, sdt, phai, ngsinh, dchi)              )      for maKH in deleted\_items:          cursor.execute("DELETE FROM KHACHHANG WHERE maKH=?", (maKH,))      deleted\_items.clear()      conn.commit()      messagebox.showinfo("Thành công", "Đã lưu dữ liệu vào CSDL!")      load\_data()      lam\_moi\_form()  def hien\_thi\_chi\_tiet(event):      selected = tree.selection()      if selected:          ma, ten, sdt, phai\_val, ngsinh, dchi = tree.item(selected[0], "values")          entry\_maKH.config(state='normal')          entry\_maKH.delete(0, tk.END)          entry\_maKH.insert(0, ma)          entry\_maKH.config(state='readonly')          entry\_ten.delete(0, tk.END)          entry\_ten.insert(0, ten)          entry\_sdt.delete(0, tk.END)          entry\_sdt.insert(0, sdt)          gender.set(phai\_val)          if isinstance(ngsinh, str):              year, month, day = map(int, ngsinh.split('-'))              date\_ngaysinh.set\_date(date(year, month, day))          else:              date\_ngaysinh.set\_date(ngsinh)          entry\_diachi.delete(0, tk.END)          entry\_diachi.insert(0, dchi)  def huy():      lam\_moi\_form()  def thoat():      root.destroy()  tk.Label(root, text="Quản Lý Khách Hàng", bg="#FFFACD", fg="black", font=("Times New Roman", 22, "bold")).place(x=330, y=20)  frame\_input = tk.Frame(root, bg="#FFFACD", width=900, height=120)  frame\_input.place(x=50, y=70)  tk.Label(frame\_input, text="Mã Khách Hàng", bg="#FFFACD").place(x=10, y=10)  entry\_maKH = tk.Entry(frame\_input, width=10)  entry\_maKH.place(x=130, y=10)  entry\_maKH.config(state='readonly')  tk.Label(frame\_input, text="Tên Khách Hàng", bg="#FFFACD").place(x=200, y=10)  entry\_ten = tk.Entry(frame\_input, width=30)  entry\_ten.place(x=320, y=10)  tk.Label(frame\_input, text="Giới Tính", bg="#FFFACD").place(x=10, y=50)  gender = tk.StringVar(value="Nam")  tk.Radiobutton(frame\_input, text="Nam", variable=gender, value="Nam", bg="#FFFACD").place(x=70, y=48)  tk.Radiobutton(frame\_input, text="Nữ", variable=gender, value="Nữ", bg="#FFFACD").place(x=130, y=48)  tk.Label(frame\_input, text="Ngày Sinh", bg="#FFFACD").place(x=200, y=50)  date\_ngaysinh = DateEntry(frame\_input, width=12, background='darkblue', foreground='white', borderwidth=2, date\_pattern='dd/mm/yyyy')  date\_ngaysinh.place(x=280, y=50)  tk.Label(frame\_input, text="Số Điện Thoại", bg="#FFFACD").place(x=400, y=50)  entry\_sdt = tk.Entry(frame\_input, width=30)  entry\_sdt.place(x=530, y=50)  tk.Label(frame\_input, text="Địa Chỉ", bg="#FFFACD").place(x=10, y=90)  entry\_diachi = tk.Entry(frame\_input, width=70)  entry\_diachi.place(x=130, y=90)  tree.bind("<<TreeviewSelect>>", hien\_thi\_chi\_tiet)  button\_style = {"font": ("Times New Roman", 13, "bold"), "bg": "#B0E0E6", "width": 8, "height": 1}  tk.Button(root, text="Thêm", \*\*button\_style, command=them).place(x=870, y=120)  tk.Button(root, text="Xóa", \*\*button\_style, command=xoa).place(x=870, y=200)  tk.Button(root, text="Sửa", \*\*button\_style, command=sua).place(x=870, y=280)  tk.Button(root, text="Hủy", \*\*button\_style, command=huy).place(x=870, y=360)  tk.Button(root, text="Lưu", \*\*button\_style, command=luu).place(x=870, y=440)  tk.Button(root, text="Thoát", \*\*button\_style, command=thoat).place(x=870, y=520)  load\_data()  lam\_moi\_form()  root.mainloop() |

**7. Code form quản lý chuyến đi**

|  |
| --- |
| import tkinter as tk  from tkinter import ttk, messagebox  from tkcalendar import DateEntry  import pyodbc  from datetime import date  import datetime  conn = pyodbc.connect(      'DRIVER={SQL Server};'     #'SERVER=ADMIN-PC;'      'SERVER=localhost\\SQLEXPRESS01;'      'DATABASE=QuanLyTuyenDuLich;'      'Trusted\_Connection=yes;'          )  cursor = conn.cursor()  root = tk.Tk()  root.title("Quản Lý Chuyến Đi")  root.geometry("1000x650")  root.config(bg="#fff8dc")  frame\_tree = tk.LabelFrame(root, text="Danh Sách Chuyến Đi", font=("Arial", 12, "bold"), bg="#fff8dc", width=880, height=350)  frame\_tree.place(x=60, y=300)  columns = ("maCD", "maTuyen", "tgKh", "ngKh", "nhanVien")  tree = ttk.Treeview(frame\_tree, columns=columns, show="headings", height=10)  for col, txt, w in zip(columns, ["Mã Chuyến Đi","Mã Tuyến","Thời Gian KH","Ngày KH","Nhân Viên"], [90,90,110,120,300]):      tree.heading(col, text=txt)      tree.column(col, width=w, anchor=tk.W if col=="nhanVien" else tk.CENTER)  tree.place(x=10, y=10, width=840, height=300)  scrollbar\_v = ttk.Scrollbar(frame\_tree, orient="vertical", command=tree.yview)  tree.configure(yscrollcommand=scrollbar\_v.set)  scrollbar\_v.place(x=855, y=10, height=300)  def auto\_maChuyen():      cursor.execute("SELECT maCD FROM CHUYENDI")      ma\_list = [row[0] for row in cursor.fetchall()]      if not ma\_list:          return "CD0001"        so\_list = sorted([int(ma[2:]) for ma in ma\_list if ma[2:].isdigit()])      next\_num = 1      for num in so\_list:          if num == next\_num:              next\_num += 1          elif num > next\_num:              break      return f"CD{next\_num:04d}"  def lay\_ds\_ma\_tuyen():      cursor.execute("SELECT maTuyen FROM TUYENDULICH WHERE trangThai = N'Hoạt động'")      return [row[0] for row in cursor.fetchall()]  def load\_data():      for item in tree.get\_children():          tree.delete(item)      cursor.execute("""      SELECT maCD, maTuyen, ngKh, tgKh, trangThai      FROM CHUYENDI      WHERE trangThai = N'Hoạt động'      ORDER BY maCD      """)      chuyen\_list = cursor.fetchall()      for chuyen in chuyen\_list:          maCD, maTuyen, ngKh, tgKh, trangThai = chuyen          cursor.execute("SELECT maNV FROM CHUYENDI\_NHANVIEN WHERE maCD=?", (maCD,))          nv\_list = [row[0].strip() for row in cursor.fetchall()]          nv\_str = ", ".join(nv\_list)            tgKh\_display = tgKh[:5] if tgKh else ""          ngKh\_display = ngKh.strftime("%d/%m/%Y") if isinstance(ngKh, datetime.date) else str(ngKh)            tree.insert("", "end", values=(maCD, maTuyen, tgKh\_display, ngKh\_display, nv\_str))  def load\_nhanvien\_ranh\_ghep(nv\_cua\_chuyen=None, ngKh\_value=None, current\_maCD=None):      lb\_nv.delete(0, tk.END)      if ngKh\_value is None:          ngKh\_value = date\_ngKh.get\_date()      ngKh\_str = ngKh\_value.strftime("%Y-%m-%d")      if nv\_cua\_chuyen is None:          nv\_cua\_chuyen = []      busy\_nv = set()      for item in tree.get\_children():          row = tree.item(item)["values"]          if len(row) < 5:              continue          maCD\_row, ngKh\_row, nhanVien\_row = row[0], row[3], row[4]          if ngKh\_row == ngKh\_str and maCD\_row != (current\_maCD or ""):              if nhanVien\_row:                  busy\_nv.update([nv.strip().upper() for nv in nhanVien\_row.split(",")])      selected\_nv = []      if current\_maCD:          cursor.execute("SELECT maNV FROM CHUYENDI\_NHANVIEN WHERE maCD=?", (current\_maCD,))          selected\_nv = [row[0].strip().upper() for row in cursor.fetchall()]      cursor.execute("SELECT maNV, hoTen, chucVu FROM NHANVIEN ORDER BY maNV")      for maNV, hoTen, chucVu in cursor.fetchall():          maNV\_clean = maNV.strip().upper()          if maNV\_clean not in busy\_nv or maNV\_clean in nv\_cua\_chuyen or maNV\_clean in selected\_nv:              display = f"{maNV\_clean} - {hoTen} - {chucVu}"              lb\_nv.insert(tk.END, display)              if maNV\_clean in nv\_cua\_chuyen or maNV\_clean in selected\_nv:                  lb\_nv.selection\_set(tk.END)  def lam\_moi\_form():      txt\_ma\_cd.config(state="normal")      txt\_ma\_cd.delete(0, tk.END)      txt\_ma\_cd.insert(0, auto\_maChuyen())      txt\_ma\_cd.config(state="readonly")      cb\_ma\_tuyen.set("Chọn mã tuyến")      txt\_tgkh.delete(0, tk.END)      date\_ngKh.set\_date(date.today())      load\_nhanvien\_ranh\_ghep()  def chon\_chuyen\_di(event=None):      sel = tree.selection()      if not sel: return      maCD, maTuyen, tgKh, ngKh, nhanVien = tree.item(sel[0])["values"]      txt\_ma\_cd.config(state="normal")      txt\_ma\_cd.delete(0, tk.END)      txt\_ma\_cd.insert(0, maCD)      txt\_ma\_cd.config(state="readonly")      cb\_ma\_tuyen.set(maTuyen)      txt\_tgkh.delete(0, tk.END)      txt\_tgkh.insert(0, tgKh)      try:          date\_ngKh.set\_date(datetime.datetime.strptime(ngKh, "%Y-%m-%d").date())      except:          try:              date\_ngKh.set\_date(datetime.datetime.strptime(ngKh, "%d/%m/%Y").date())          except:              date\_ngKh.set\_date(date.today())      nv\_cua\_chuyen = [nv.strip() for nv in nhanVien.split(",")] if nhanVien else []      load\_nhanvien\_ranh\_ghep(nv\_cua\_chuyen=nv\_cua\_chuyen, ngKh\_value=date\_ngKh.get\_date(), current\_maCD=maCD)  def them():      maCD = txt\_ma\_cd.get()      maTuyen = cb\_ma\_tuyen.get()      tgKh\_value = txt\_tgkh.get()      ngKh\_value = date\_ngKh.get\_date()      selected\_nv = [lb\_nv.get(i).split(" - ")[0].strip() for i in lb\_nv.curselection()]      if maTuyen=="Chọn mã tuyến" or not selected\_nv or not tgKh\_value:          messagebox.showwarning("Thiếu dữ liệu", "Vui lòng nhập đầy đủ dữ liệu")          return      try:          tgKh\_str = datetime.datetime.strptime(tgKh\_value, "%H:%M").strftime("%H:%M")      except:          messagebox.showerror("Lỗi", "Thời gian phải HH:MM")          return      cursor.execute("SELECT 1 FROM CHUYENDI WHERE maCD=?", (maCD,))      if cursor.fetchone():          messagebox.showerror("Lỗi", "Mã chuyến đi đã tồn tại")          return      ngKh\_str = ngKh\_value.strftime("%Y-%m-%d")      cursor.execute("INSERT INTO CHUYENDI(maCD, maTuyen, ngKh, tgKh, trangThai) VALUES (?, ?, ?, ?, N'Hoạt động')",                     (maCD, maTuyen, ngKh\_str, tgKh\_str))      for maNV in selected\_nv:          cursor.execute("INSERT INTO CHUYENDI\_NHANVIEN(maCD, maNV) VALUES (?, ?)", (maCD, maNV))      conn.commit()      messagebox.showinfo("Thành công", "Thêm chuyến đi thành công")      load\_data()      lam\_moi\_form()  def xoa():      sel = tree.selection()      if not sel:          messagebox.showwarning("Lỗi", "Chưa chọn chuyến đi")          return      for item in sel:          maCD = tree.item(item)["values"][0]          cursor.execute("UPDATE CHUYENDI SET trangThai=N'Ngưng hoạt động' WHERE maCD=?", (maCD,))      conn.commit()      messagebox.showinfo("Thành công", "Xóa chuyến đi thành công")      load\_data()      lam\_moi\_form()  def sua():      maCD = txt\_ma\_cd.get()      maTuyen = cb\_ma\_tuyen.get()      tgKh\_value = txt\_tgkh.get()      ngKh\_value = date\_ngKh.get\_date()      selected\_nv = [lb\_nv.get(i).split(" - ")[0].strip() for i in lb\_nv.curselection()]      if not maCD or maTuyen=="Chọn mã tuyến" or not tgKh\_value:          messagebox.showwarning("Thiếu dữ liệu", "Vui lòng nhập đầy đủ dữ liệu")          return      try:          tgKh\_str = datetime.datetime.strptime(tgKh\_value, "%H:%M").strftime("%H:%M")      except:          messagebox.showerror("Lỗi", "Thời gian phải HH:MM")          return      ngKh\_str = ngKh\_value.strftime("%Y-%m-%d")      cursor.execute("UPDATE CHUYENDI SET maTuyen=?, ngKh=?, tgKh=?, trangThai=N'Hoạt động' WHERE maCD=?",                     (maTuyen, ngKh\_str, tgKh\_str, maCD))      cursor.execute("DELETE FROM CHUYENDI\_NHANVIEN WHERE maCD=?", (maCD,))      for maNV in selected\_nv:          cursor.execute("INSERT INTO CHUYENDI\_NHANVIEN(maCD, maNV) VALUES (?, ?)", (maCD, maNV))      conn.commit()      messagebox.showinfo("Thành công", "Cập nhật chuyến đi thành công")      load\_data()      lam\_moi\_form()  def luu():      confirm = messagebox.askyesno("Xác nhận", "Bạn có chắc muốn lưu dữ liệu Treeview vào CSDL?")      if not confirm:          return      ds\_tree = []      chuyen\_di\_nv = {}      for item in tree.get\_children():          values = tree.item(item, "values")          if len(values) < 5:              continue          maCD, maTuyen, tgKh\_value, ngKh\_value, nhanVien\_str = values[:5]          ds\_tree.append(maCD)          nv\_list = []          if nhanVien\_str:              for nv in nhanVien\_str.split(","):                  nv\_list.append(nv.strip().split(" - ")[0])          chuyen\_di\_nv[maCD] = nv\_list      if ds\_tree:          placeholders = ",".join("?" for \_ in ds\_tree)          sql\_update\_off = f"""              UPDATE CHUYENDI              SET trangThai = N'Ngưng hoạt động'              WHERE maCD NOT IN ({placeholders})          """          cursor.execute(sql\_update\_off, ds\_tree)      else:          cursor.execute("UPDATE CHUYENDI SET trangThai = N'Ngưng hoạt động'")      for item in tree.get\_children():          maCD, maTuyen, tgKh\_value, ngKh\_value, \_ = tree.item(item, "values")[:5]          try:              ngKh\_date = datetime.datetime.strptime(str(ngKh\_value), "%Y-%m-%d").date()          except:              ngKh\_date = datetime.datetime.strptime(str(ngKh\_value), "%d/%m/%Y").date()          ngKh\_str = ngKh\_date.strftime("%Y-%m-%d")          tgKh\_str = tgKh\_value[:5] if tgKh\_value else ""          cursor.execute("SELECT COUNT(\*) FROM CHUYENDI WHERE maCD=?", (maCD,))          exists = cursor.fetchone()[0]          if exists:              cursor.execute("""                  UPDATE CHUYENDI                  SET maTuyen=?, ngKh=?, tgKh=?, trangThai=N'Hoạt động'                  WHERE maCD=?              """, (maTuyen, ngKh\_str, tgKh\_str, maCD))              cursor.execute("DELETE FROM CHUYENDI\_NHANVIEN WHERE maCD=?", (maCD,))          else:              cursor.execute("""                  INSERT INTO CHUYENDI(maCD, maTuyen, ngKh, tgKh, trangThai)                  VALUES (?, ?, ?, ?, N'Hoạt động')              """, (maCD, maTuyen, ngKh\_str, tgKh\_str))          for maNV in chuyen\_di\_nv.get(maCD, []):              cursor.execute(                  "INSERT INTO CHUYENDI\_NHANVIEN(maCD, maNV) VALUES (?, ?)",                  (maCD, maNV)              )      conn.commit()      messagebox.showinfo("Thành công", "Đã lưu dữ liệu!")  tk.Label(root, text="Quản Lý Chuyến Đi", font=("Arial", 20, "bold"), bg="#fff8dc").place(x=380, y=30)  tk.Label(root, text="Mã Chuyến Đi", font=("Arial", 12), bg="#fff8dc").place(x=120, y=100)  txt\_ma\_cd = tk.Entry(root, width=20)  txt\_ma\_cd.place(x=300, y=100)  txt\_ma\_cd.insert(0, auto\_maChuyen())  txt\_ma\_cd.config(state="readonly")  tk.Label(root, text="Mã Tuyến", font=("Arial", 12), bg="#fff8dc").place(x=120, y=130)  cb\_ma\_tuyen = ttk.Combobox(root, width=20, state="readonly", values=lay\_ds\_ma\_tuyen())  cb\_ma\_tuyen.place(x=300, y=130)  cb\_ma\_tuyen.set("Chọn mã tuyến")  tk.Label(root, text="Ngày Khởi Hành", font=("Arial", 12), bg="#fff8dc").place(x=120, y=160)  date\_ngKh = DateEntry(root, width=20, background='darkblue', foreground='white', borderwidth=2, date\_pattern='dd/mm/yyyy')  date\_ngKh.place(x=300, y=160)  date\_ngKh.set\_date(date.today())  tk.Label(root, text="Thời Gian Khởi Hành", font=("Arial", 12), bg="#fff8dc").place(x=120, y=190)  txt\_tgkh = tk.Entry(root, width=20)  txt\_tgkh.place(x=300, y=190)  tk.Label(root, text="Nhân Viên Phụ Trách", font=("Arial", 12), bg="#fff8dc").place(x=510, y=100)  lb\_nv = tk.Listbox(root, selectmode="multiple", height=7)  lb\_nv.place(x=680, y=100, width=300)  date\_ngKh.bind("<<DateEntrySelected>>", lambda e: load\_nhanvien\_ranh\_ghep(ngKh\_value=date\_ngKh.get\_date()))  tree.bind("<<TreeviewSelect>>", chon\_chuyen\_di)  btn\_them = tk.Button(root, text="Thêm", bg="#87cefa", font=("Arial", 12, "bold"), width=10, command=them)  btn\_them.place(x=100, y=250)  btn\_sua = tk.Button(root, text="Sửa", bg="#87cefa", font=("Arial", 12, "bold"), width=10, command=sua)  btn\_sua.place(x=240, y=250)  btn\_xoa = tk.Button(root, text="Xóa", bg="#87cefa", font=("Arial", 12, "bold"), width=10, command=xoa)  btn\_xoa.place(x=380, y=250)  btn\_huy = tk.Button(root, text="Hủy", bg="#87cefa", font=("Arial", 12, "bold"), width=10, command=lam\_moi\_form)  btn\_huy.place(x=520, y=250)  btn\_luu = tk.Button(root, text="Lưu", bg="#87cefa", font=("Arial", 12, "bold"), width=10, command=luu)  btn\_luu.place(x=660, y=250)  btn\_thoat = tk.Button(root, text="Thoát", bg="#87cefa", font=("Arial", 12, "bold"), width=10, command=root.destroy)  btn\_thoat.place(x=800, y=250)  load\_data()  root.mainloop() |

**8. Code form quản lý đặt vé**

|  |
| --- |
| import tkinter as tk  from tkinter import ttk, messagebox  from tkcalendar import DateEntry  import pyodbc  from datetime import datetime, date  conn = pyodbc.connect(      'DRIVER={SQL Server};'     #'SERVER=ADMIN-PC;'      'SERVER=localhost\\SQLEXPRESS01;'      'DATABASE=QuanLyTuyenDuLich;'      'Trusted\_Connection=yes;'  )  cursor = conn.cursor()  root = tk.Tk()  root.title("Quản Lý Đặt Vé")  root.geometry("1000x650")  root.config(bg="#fff8dc")  frame\_tree = tk.LabelFrame(root, text="Thông Tin Đặt Vé", font=("Arial", 12, "bold"), bg="#fff8dc", width=790, height=400)  frame\_tree.place(x=50, y=250)  columns = ("maVe", "maCD", "maKH", "ngDat", "trangThai", "giaVe", "soLuong", "thanhTien")  tree = ttk.Treeview(frame\_tree, columns=columns, show="headings", height=10)  tree.heading("maVe", text="Mã Vé")  tree.heading("maCD", text="Mã Chuyến Đi")  tree.heading("maKH", text="Mã Khách Hàng")  tree.heading("ngDat", text="Ngày Đặt")  tree.heading("soLuong", text="Số Lượng")  tree.heading("giaVe", text="Giá Vé")  tree.heading("trangThai", text="Trạng Thái")  tree.heading("thanhTien", text="Thành Tiền")  tree.column("maVe", width=90, anchor="center")  tree.column("maCD", width=90, anchor="center")  tree.column("maKH", width=95, anchor="center")  tree.column("ngDat", width=80, anchor="center")  tree.column("soLuong", width=70, anchor="center")  tree.column("giaVe", width=90, anchor="center")  tree.column("trangThai", width=100, anchor="center")  tree.column("thanhTien", width=120, anchor="center")  scrollbar\_v = ttk.Scrollbar(frame\_tree, orient="vertical", command=tree.yview)  scrollbar\_h = ttk.Scrollbar(frame\_tree, orient="horizontal", command=tree.xview)  tree.configure(yscrollcommand=scrollbar\_v.set, xscrollcommand=scrollbar\_h.set)  tree.place(x=10, y=10, width=750, height=340)  scrollbar\_v.place(x=765, y=10, width=15, height=350)  scrollbar\_h.place(x=10, y=355, width=755, height=15)  def load\_data():      for i in tree.get\_children():          tree.delete(i)      cursor.execute("SELECT maVe, maCD, maKH, ngDat, trangThai, giaVe, soluong, thanhTien FROM DATVE")      rows = cursor.fetchall()      for row in rows:          tree.insert("", "end", values=(row[0].strip(), row[1].strip(), row[2].strip(), row[3], row[4], row[5], row[6], row[7]))  def auto\_maVe():      ma\_list\_tree = [tree.item(item)['values'][0].strip() for item in tree.get\_children()]      cursor.execute("SELECT maVe FROM DATVE")      ma\_list\_sql = [row[0].strip() for row in cursor.fetchall()]        # Gộp mã      ma\_list = list(set(ma\_list\_tree + ma\_list\_sql))        if not ma\_list:          return "MV00000001"        so\_list = sorted([int(ma[2:]) for ma in ma\_list if ma[2:].isdigit()])      next\_num = so\_list[-1] + 1      return f"MV{next\_num:08d}"  def lay\_danh\_sach\_ma\_khach\_hang():      cursor.execute("SELECT maKh FROM KHACHHANG")      data = cursor.fetchall()      ds\_ma = [row[0] for row in data]  # lấy cột maKh      return ds\_ma  def lay\_danh\_sach\_ma\_chuyen\_di():      cursor.execute("SELECT maCD FROM CHUYENDI")      data = cursor.fetchall()      ds\_cd = [row[0] for row in data]  # lấy cột maKh      return ds\_cd  def lam\_moi\_form():      entry\_maVe.config(state='normal')      entry\_maVe.delete(0, tk.END)      entry\_maVe.insert(0, auto\_maVe())      entry\_maVe.config(state='readonly')      entry\_maKH.set("")      entry\_maCD.set("")      date\_ngDat.set\_date(date.today())      entry\_trangThai.delete(0, tk.END)      entry\_giaVe.delete(0, tk.END)      spin\_soLuong.delete(0, tk.END)  def them():      maVe = auto\_maVe()      maKH = entry\_maKH.get().strip()      maCD = entry\_maCD.get().strip()      ngDat = date\_ngDat.get\_date().strftime('%Y-%m-%d')      trangThai = entry\_trangThai.get().strip()      giaVe = float(entry\_giaVe.get().strip())      soluong = int(spin\_soLuong.get().strip())      thanhTien = giaVe \* soluong      if not trangThai or not giaVe or not soluong:          messagebox.showwarning("Thiếu dữ liệu", "Vui lòng nhập đầy đủ thông tin!")          return      tree.insert("", "end", values=(maVe,  maCD, maKH, ngDat, trangThai, giaVe, soluong, thanhTien))      lam\_moi\_form()  def xoa():      selected = tree.selection()      if not selected:          messagebox.showwarning("Chưa chọn", "Vui lòng chọn vé để xóa!")          return      confirm = messagebox.askyesno("Xác nhận", "Bạn có chắc muốn xóa?")      if confirm:          tree.delete(selected[0])          lam\_moi\_form()    def sua():      selected = tree.selection()      if not selected:          messagebox.showwarning("Chưa chọn", "Vui lòng chọn khách hàng để sửa!")          return      maVe = tree.item(selected[0])['values'][0]      maCD = entry\_maCD.get().strip()      maKH = entry\_maKH.get().strip()      ngDat = date\_ngDat.get\_date().strftime('%Y-%m-%d')      trangThai = entry\_trangThai.get().strip()      giaVe = float(entry\_giaVe.get().strip())      soluong = int(spin\_soLuong.get().strip())      thanhTien = giaVe \* soluong      messagebox.showinfo("Thành công","Đã cập nhật thông tin khách hàng")      tree.item(selected[0], values=(maVe,  maCD, maKH, ngDat, trangThai, giaVe, soluong, thanhTien))    deleted\_items = []  # danh sách toàn cục  def luu():      for item in tree.get\_children():          maVe,  maCD, maKH, ngDat, trangThai, giaVe, soluong, thanhTien = tree.item(item, "values")          cursor.execute("SELECT 1 FROM DATVE WHERE maVe=?", (maVe,))          if cursor.fetchone():              cursor.execute(                  "UPDATE DATVE SET maCD=?, maKH=?, ngDat=?, trangThai=?, giaVe=?, soluong=?, thanhTien=? WHERE maVe=?",                  (maCD, maKH, ngDat, trangThai, giaVe, soluong, thanhTien, maVe)              )          else:              cursor.execute(                  "INSERT INTO DATVE (maVe, maCD, maKH, ngDat, trangThai, giaVe, soluong, thanhTien) VALUES (?, ?, ?, ?, ?, ?, ?, ?)",                  (maVe, maCD, maKH, ngDat, trangThai, giaVe, soluong, thanhTien)              )      for maKH in deleted\_items:          cursor.execute("DELETE FROM KHACHHANG WHERE maKH=?", (maKH,))      deleted\_items.clear()      conn.commit()      messagebox.showinfo("Thành công", "Đã lưu dữ liệu vào CSDL!")      load\_data()      lam\_moi\_form()  def huy():      lam\_moi\_form()  def hien\_thi\_chi\_tiet(event=None):      selected = tree.selection()      if selected:          maVe, maCD, maKh, ngDat, trangThai, giaVe, soluong, \_ = tree.item(selected[0], "values")          entry\_maVe.config(state='normal')          entry\_maVe.delete(0, tk.END)          entry\_maVe.insert(0, maVe)          entry\_maVe.config(state='readonly')          entry\_maKH.set(maKh)          entry\_maCD.set(maCD)          ngay = datetime.strptime(ngDat, '%Y-%m-%d').date()          date\_ngDat.set\_date(ngay)          entry\_trangThai.set(trangThai)          entry\_giaVe.delete(0, tk.END)          entry\_giaVe.insert(0, giaVe)          spin\_soLuong.delete(0, tk.END)          spin\_soLuong.insert(0, soluong)          giaVe = float(giaVe)          soluong = int(soluong)    def thoat():      root.destroy()  btn\_home = tk.Button(root, text="Về Trang Chủ", font=("Times New Roman", 10, "bold"), bg="white", relief="groove", command=quit)  btn\_home.place(x=50, y=30)  title\_label = tk.Label(root, text="Quản Lý Đặt Vé", font=("Arial", 20, "bold"), bg="#fff8dc")  title\_label.place(x=400, y=45)  frame\_tree = tk.Frame(root, bg="#fff8dc")  frame\_tree.place(x=100, y=100)  tk.Label(frame\_tree, text="Mã Vé", bg="#fff8dc", font=("Times New Roman", 12)).grid(row=0, column=0, sticky="w", pady=5)  entry\_maVe = tk.Entry(frame\_tree, width=30)  entry\_maVe.grid(row=0, column=1, padx=10, pady=5)  entry\_maVe.config(state='readonly')  tk.Label(frame\_tree, text="Số Lượng", font=("Times New Roman", 12), bg="#fff8dc").grid(row=0, column=2, sticky="w", padx=10, pady=5)  spin\_soLuong = tk.Spinbox(frame\_tree, from\_=1, to=100, width=10)  spin\_soLuong.grid(row=0, column=3, padx=10, pady=5, columnspan=2)  tk.Label(frame\_tree, text="Mã Chuyến Đi", bg="#fff8dc", font=("Times New Roman", 12)).grid(row=2, column=0, sticky="w", pady=5)  entry\_maCD = ttk.Combobox(frame\_tree, width=30, values=lay\_danh\_sach\_ma\_chuyen\_di(),state="readonly")  entry\_maCD.grid(row=2, column=1, padx=10, pady=5)  entry\_maCD.set("")  tk.Label(frame\_tree, text="Giá Vé", bg="#fff8dc", font=("Times New Roman", 12)).grid(row=1, column=2, sticky="w", padx=10)  entry\_giaVe = tk.Entry(frame\_tree, width=30)  entry\_giaVe.grid(row=1, column=3, padx=10, pady=5, columnspan=2)  tk.Label(frame\_tree, text="Mã Khách Hàng", bg="#fff8dc", font=("Times New Roman", 12)).grid(row=1, column=0, sticky="w", pady=5)  entry\_maKH = ttk.Combobox(frame\_tree, width=30, state='readonly', values=lay\_danh\_sach\_ma\_khach\_hang())  entry\_maKH.grid(row=1, column=1, padx=10, pady=5)  entry\_maKH.set("")  tk.Label(frame\_tree, text="Trạng Thái", bg="#fff8dc", font=("Times New Roman", 12)).grid(row=2, column=2, sticky="w", padx=10)  entry\_trangThai = ttk.Combobox(frame\_tree, width=30, values=["đã thanh toán","chưa thanh toán"], state='readonly')  entry\_trangThai.grid(row=2, column=3, padx=10, pady=5, columnspan=2)  entry\_trangThai.set("")  tk.Label(frame\_tree, text="Ngày Đặt", bg="#fff8dc", font=("Times New Roman", 12)).grid(row=3, column=0, sticky="w", padx=5)  date\_ngDat = DateEntry(frame\_tree, width=12, background='darkblue', foreground='white', borderwidth=2, date\_pattern='d/m/yyyy')  date\_ngDat.grid(row=3, column=1, padx=10, pady=5)  button\_style = {"font": ("Times New Roman", 13, "bold"), "bg": "#B0E0E6", "width": 8, "height": 1}  btn\_them = tk.Button(root, text="Thêm", \*\*button\_style, command=them)  btn\_them.place(x=880, y=130)  btn\_xoa = tk.Button(root, text="Xóa", \*\*button\_style, command=xoa)  btn\_xoa.place(x=880, y=210)  btn\_sua = tk.Button(root, text="Sửa", \*\*button\_style, command=sua)  btn\_sua.place(x=880, y=290)  btn\_huy = tk.Button(root, text="Hủy", \*\*button\_style, command=huy)  btn\_huy.place(x=880, y=370)  btn\_luu = tk.Button(root, text="Lưu", \*\*button\_style, command=luu)  btn\_luu.place(x=880, y=450)  btn\_thoat = tk.Button(root, text="Thoát", \*\*button\_style, command=thoat)  btn\_thoat.place(x=880, y=530)  tree.bind("<<TreeviewSelect>>", hien\_thi\_chi\_tiet)  load\_data()  lam\_moi\_form()  root.mainloop() |

**9. Code form quản lý doanh thu**

|  |
| --- |
| import tkinter as tk  from tkinter import ttk, messagebox  import pyodbc  from datetime import datetime  conn = pyodbc.connect(      'DRIVER={SQL Server};'     #'SERVER=ADMIN-PC;'      'SERVER=localhost\\SQLEXPRESS01;'      'DATABASE=QuanLyTuyenDuLich;'      'Trusted\_Connection=yes;'  )  cursor = conn.cursor()  root = tk.Tk()  root.title("Quản Lý Doanh Thu")  root.geometry("1000x650")  root.configure(bg="#FFFACD")  frame\_tree = tk.LabelFrame(root, text="Chi tiết doanh thu", font=("Times New Roman", 12, "bold"), bg="#FFFACD")  frame\_tree.place(x=50, y=350, width=900, height=250)  columns = ("maCD","maVe", "giaVe", "soLuong", "thanhTien", "ngKh")  tree = ttk.Treeview(frame\_tree, columns=columns, show="headings", height=10)  for col, txt in zip(columns, ["Mã Chuyến","Mã vé", "Giá Vé", "Số Lượng", "Thành Tiền", "Ngày Khởi Hành"]):      tree.heading(col, text=txt)      tree.column(col, anchor="center", width=150)  tree.pack(fill="both", expand=True)  scrollbar = ttk.Scrollbar(frame\_tree, orient="vertical", command=tree.yview)  tree.configure(yscroll=scrollbar.set)  scrollbar.pack(side="right", fill="y")  def chon\_quy():      month.set(0)  def chon\_thang():      quarter.set(0)  def xem\_doanh\_thu():      for row in tree.get\_children():          tree.delete(row)      try:          nam = int(entry\_year.get())      except:          messagebox.showwarning("Lỗi", "Vui lòng nhập năm hợp lệ!")          return      thang\_chon = month.get()      quy\_chon = quarter.get()      tong\_tien = 0      query = """          SELECT  DV.maCD,DV.maVe, DV.giaVe, DV.soLuong, (DV.giaVe \* DV.soLuong) AS thanhTien, C.ngKh          FROM DATVE DV          JOIN CHUYENDI C ON DV.maCD = C.maCD          WHERE DV.trangThai = N'đã thanh toán' AND YEAR(C.ngKh) = ?      """      params = [nam]      if quy\_chon != 0:          query += " AND DATEPART(QUARTER, C.ngKh) = ?"          params.append(quy\_chon)      elif thang\_chon != 0:          query += " AND MONTH(C.ngKh) = ?"          params.append(thang\_chon)      query += " ORDER BY C.ngKh, DV.maCD"      cursor.execute(query, params)      rows = cursor.fetchall()      for row in rows:          maCD,maVe, giaVe, soLuong, thanhTien, ngKh = row          if isinstance(ngKh, str):              try:                  ngKh\_dt = datetime.strptime(ngKh, "%Y-%m-%d").date()              except ValueError:                  ngKh\_dt = datetime.today().date()          else:              ngKh\_dt = ngKh          giaVe = float(giaVe)          thanhTien = float(thanhTien)          tree.insert(              "",              "end",              values=( maCD,maVe, f"{giaVe:,.0f}", soLuong, f"{thanhTien:,.0f}", ngKh\_dt.strftime("%d/%m/%Y"))          )          tong\_tien += thanhTien or 0      lbl\_tong.config(text=f"Tổng doanh thu: {tong\_tien:,.0f} VND")  def ve\_trang\_chu():      root.destroy()  tk.Button(root, text="Xem", font=("Times New Roman", 14, "bold"),bg="skyblue", width=10, command=xem\_doanh\_thu).place(x=780, y=170)  btn\_home = tk.Button(root, text="Về Trang Chủ", font=("Times New Roman", 10, "bold"),bg="white", relief="groove", command=ve\_trang\_chu)  btn\_home.place(x=50, y=40)  tk.Label(root, text="Quản Lý Doanh Thu", font=("Times New Roman", 24, "bold"), bg="#FFFACD").place(x=440, y=35)  tk.Label(root, text="Quý", font=("Times New Roman", 12), bg="#FFFACD").place(x=160, y=120)  quarter = tk.IntVar(value=0)  for i in range(1, 5):      tk.Radiobutton(root, text=str(i), variable=quarter, value=i, bg="#FFFACD",                     font=("Times New Roman", 12), command=chon\_quy).place(x=220+(i-1)\*150, y=120)  tk.Label(root, text="Tháng", font=("Times New Roman", 12), bg="#FFFACD").place(x=160, y=170)  month = tk.IntVar(value=0)  for i in range(12):      tk.Radiobutton(root, text=str(i+1), variable=month, value=i+1, bg="#FFFACD",                     font=("Times New Roman", 12), command=chon\_thang).place(x=220+(i%4)\*150, y=170+(i//4)\*30)  tk.Label(root, text="Năm", font=("Times New Roman", 12), bg="#FFFACD").place(x=160, y=280)  entry\_year = tk.Entry(root, width=30)  entry\_year.place(x=220, y=280)  lbl\_tong = tk.Label(root, text="Tổng doanh thu: 0 VND", font=("Times New Roman", 12, "bold"),bg="#FFFACD", fg="red")  lbl\_tong.place(x=50, y=610)  root.mainloop() |