

**TRƯỜNG ĐẠI HỌC AN GIANG**  
**KHOA CÔNG NGHỆ THÔNG TIN**

**BÁO CÁO ĐỒ ÁN**

**Chuyên đề Python (COS525)**

**XÂY DỰNG ỨNG DỤNG QUẢN LÝ CÁC  
TUYẾN DU LỊCH VỚI PYTHON, TKINTER VÀ  
MYSQL**

Giảng viên hướng dẫn: Ths. Nguyễn Ngọc Minh

Sinh viên thực hiện: <DPM245504>.<Nguyễn ThanhTrí>.<25PM.Python\_Nh4\_To3>

## 1. Đặt vấn đề

### 1.1 Tính cần thiết của đề tài

Trong thời đại công nghệ số, nhu cầu du lịch ngày càng phát triển mạnh mẽ. Các công ty lữ hành cần một hệ thống giúp quản lý các tuyến du lịch một cách khoa học, hạn chế sai sót trong khâu quản lý thủ công như: trùng lịch trình, cập nhật giá tour chậm, khó kiểm soát số lượng khách, dễ thất lạc thông tin như booking...

Tuy đã có nhiều ứng dụng quản lý du lịch, nhưng đa số chưa tối ưu cho các doanh nghiệp nhỏ hoặc cho mục đích học tập, nghiên cứu. Vì vậy, việc xây dựng một ứng dụng đơn giản, dễ sử dụng, hỗ trợ quản lý tuyến du lịch là rất cần thiết.

### 1.2 Mục tiêu nghiên cứu

- Xây dựng được ứng dụng quản lý hoàn chỉnh các tuyến du lịch: thêm, sửa, xóa, tìm kiếm.
- Quản lý được thông tin lịch trình, giá tour, thời gian diễn ra tour.
- Hỗ trợ quản lý khách hàng và đặt tour.
- Giao diện thân thiện, dễ sử dụng, hoạt động ổn định. Mục tiêu được thiết kế cụ thể, có tính khả thi, không quá rộng.

## 2. Tổng quan và cơ sở lý thuyết

### 2.1 Đặt vấn đề

Trong thực tế, việc quản lý tuyến du lịch thủ công thường gặp khó khăn như dữ liệu phân tán, dễ sai sót, khó thống kê. Đề tài đặt ra yêu cầu xây dựng một phần mềm giúp tự động hóa các nghiệp vụ quản lý tour.

### 2.2 Lịch sử giải quyết vấn đề

Hiện nay nhiều công ty sử dụng các hệ thống lớn như TravelERP, Amadeus... nhưng chúng khá phức tạp, chi phí cao, không phù hợp với mô hình vừa và nhỏ hoặc bài tập môn học. Các bài toán nhỏ trong trường học thường chỉ dừng ở mức quản lý thông tin đơn giản, chưa tích hợp đầy đủ chức năng về lịch trình, giá tour và quản lý khách đặt tour. Vì vậy, đề tài mang tính mới trong phạm vi học thuật, hướng đến việc xây dựng hệ thống nhỏ nhưng đầy đủ tính năng.

## 2.3 Phạm vi của đề tài

**Phạm vi nội dung:** Quản lý tuyến du lịch, lịch trình, giá tour, khách hàng, đơn đặt tour.

**Phạm vi người dùng:** Quản trị viên hoặc nhân viên điều hành tour.

**Giới hạn đề tài:** Không tập trung vào thanh toán trực tuyến hoặc bản đồ định vị.

## 2.4 Phương pháp nghiên cứu/hướng giải quyết vấn đề:

- Nghiên cứu lý thuyết về lập trình Python, xử lý file, cấu trúc dữ liệu (list, dict,...).
- Áp dụng mô hình CRUD (Create – Read – Update – Delete) để quản lý tuyến du lịch.
- Thiết kế chương trình theo dạng module hoặc hướng đối tượng (OOP) để dễ mở rộng.
- Thử nghiệm chương trình với nhiều bộ dữ liệu khác nhau để đảm bảo tính ổn định.

## 2.5 Trình bày tóm tắt cơ sở lý thuyết:

- Kiến thức về Python cơ bản: biến, kiểu dữ liệu, hàm, vòng lặp,...
- Kiến thức về Python nâng cao hơn: lập trình hướng đối tượng, xử lý file (txt, json, csv).
- Kiến thức về quản lý dữ liệu dạng danh sách (list) và từ điển (dict).
- Các nguyên tắc thiết kế phần mềm đơn giản: dễ dùng, dễ mở rộng, dễ bảo trì.

## 3. Kết quả đạt được và demo

### A. THIẾT KẾ GIAO DIỆN QUẢN LÝ CÁC TUYẾN DU LỊCH

Quản Lý Tuyến Du Lịch

## QUẢN LÝ CÁC TUYẾN DU LỊCH

**Thông tin tuyến du lịch**

Mã tuyến:  Tên tuyến:

Điểm đi:  Điểm đến:

Thời gian:  Giá tour:

**Danh sách các tuyến du lịch:**

Matuyen	Tentuyen	Diemdi	Diemden	Thoigian	Giatur
1	An Giang - Đà Lạt	An Giang	Đà Lạt	3 ngày 2 đêm	2500000
2	Hà Nội - Sapa	Hà Nội	Sapa	4 ngày 3 đêm	3500000
3	TP.HCM - Phú Quốc	TP.HCM	Phú Quốc	3 ngày 2 đêm	2800000
4	Cần Thơ - Đà Nẵng	Cần Thơ	Đà Nẵng	5 ngày 4 đêm	4200000
5	Hà Nội - Hạ Long	Hà Nội	Hạ Long	2 ngày 1 đêm	1800000

## B. CÁC BƯỚC XÂY DỰNG ỨNG DỤNG

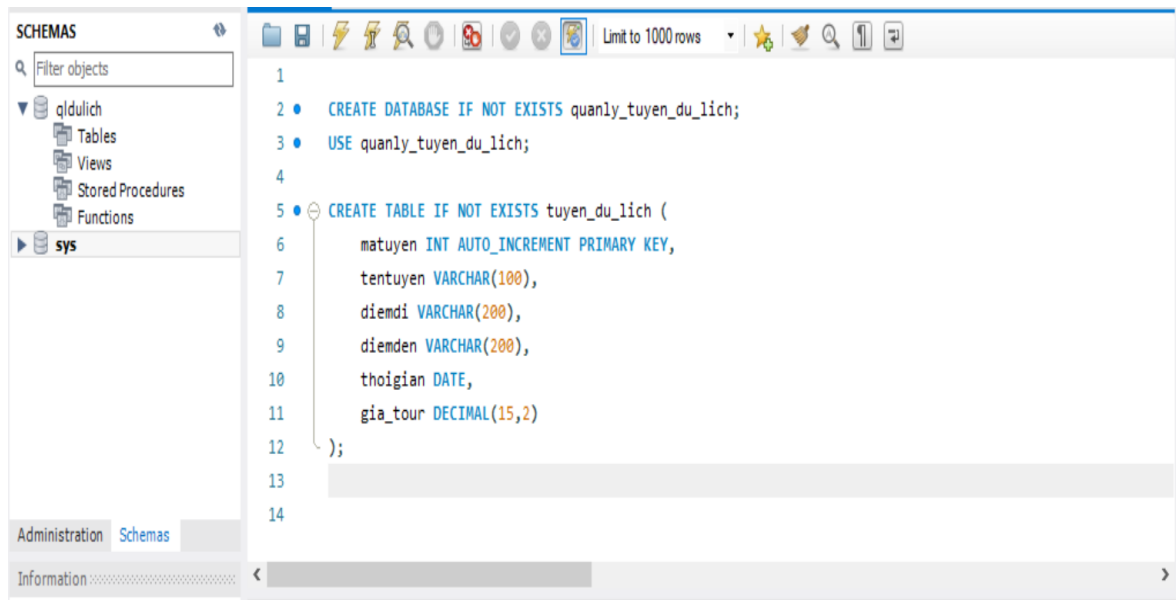
Để xây dựng ứng dụng quản lý các tuyến du lịch bằng Python, Tkinter và MySQL, cần cài đặt các thư viện cần thiết, thiết kế cơ sở dữ liệu MySQL để lưu trữ thông tin sản phẩm, sử dụng Tkinter để tạo giao diện người dùng, và kết hợp Python với thư viện mysql.connector để thực hiện thao tác như thêm, xóa,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

### 1. Cài đặt các thư viện cần thiết:

- Python: Đảm bảo đã cài đặt Python trên máy của mình
- MySQL: Cài đặt máy chủ MySQL và tạo một cơ sở dữ liệu cho ứng dụng của bạn.
- mysql-connector-python: Cài đặt thư viện này để kết nối Python với cơ sở dữ liệu MySQL

## 2. Thiết kế dữ liệu MySQL workbench

- Tạo một bảng cơ sở dữ liệu MySQL để lưu trữ thông tin các tuyến du lịch như sau:



## 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 Nhân viên.

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

- + Thêm thông tin các tuyến du lịch mới.
- + Lưu thông tin các tuyến du lịch.
- + Sửa thông tin các tuyến du lịch.
- + Hủy thông tin các tuyến du lịch.
- + Xóa thông tin các tuyến du lịch.
- + Xem danh sách tất cả các tuyến du lịch..
- + Tìm kiếm các tuyến du lịch..

#### 4. Viết mã Python kết nối cơ sở dữ liệu:

- Sử dụng `mysql.connector` để thiết lập kết nối với cơ sở dữ liệu MySQL.
- Viết các hàm:
  - + Thêm tuyến du lịch: Lấy dữ liệu từ các trường nhập liệu trên giao diện và chèn vào bảng.
  - + Lưu thông tin tuyến du lịch
  - + Sửa thông tin tuyến du lịch: Sửa đổi thông tin du lịch trong cơ sở dữ liệu dựa trên matuyen
  - + Hủy thông tin tuyến du lịch.
  - + Xóa thông tin tuyến du lịch: Xóa tuyến du lịch khỏi cơ sở dữ liệu dựa trên matuyen
  - + Xem danh sách tất cả các tuyến du lịch: Truy vấn cơ sở dữ liệu để lấy tất cả tuyến du lịch và hiển thị chúng trong bảng trên giao diện.
  - + Tìm kiếm tuyến du lịch.

#### 5. Xây dựng giao diện Tkinter

- Liên kết các nút bấm trên giao diện Tkinter với các hàm Python tương ứng để xử lý dữ liệu và tương tác với cơ sở dữ liệu.
- Các Chức Năng:
  - + Thêm tuyến du lịch : Khi người dùng nhập tên tuyến, điểm đi, điểm đến và nhấn nút "Thêm", mã Python sẽ lấy các giá trị này và thêm một bản ghi mới vào bảng `tuyendulich` trong MySQL.
  - + Xem tuyến du lịch: Khi người dùng mở cửa sổ ứng dụng, mã Python sẽ truy vấn cơ sở dữ liệu, lấy tất cả thông tin tuyến du lịch hiển thị chúng dưới dạng một danh sách hoặc bảng trên giao diện Tkinter.

## C. HƯỚNG DẪN CÀI ĐẶT VÀ CODE ỨNG DỤNG

- Ứng dụng sẽ có:

- + Giao diện nhập thông tin nhân viên (Mã tuyến, Tên tuyến, Điểm đi, Điểm đến, Thời gian, Giá tour).
- + Bảng danh sách các tuyến du lịch.
- + Chức năng CRUD (Thêm, Sửa, Xóa, Lưu).
- + Dữ liệu lưu trực tiếp vào MySQL Database (không mất khi tắt ứng dụng).
- + Giao diện Tkinter thiết kế chuẩn như hình minh họa ở trên.

### 1. Cài đặt môi trường

- Trước tiên cài đặt thư viện cần thiết:

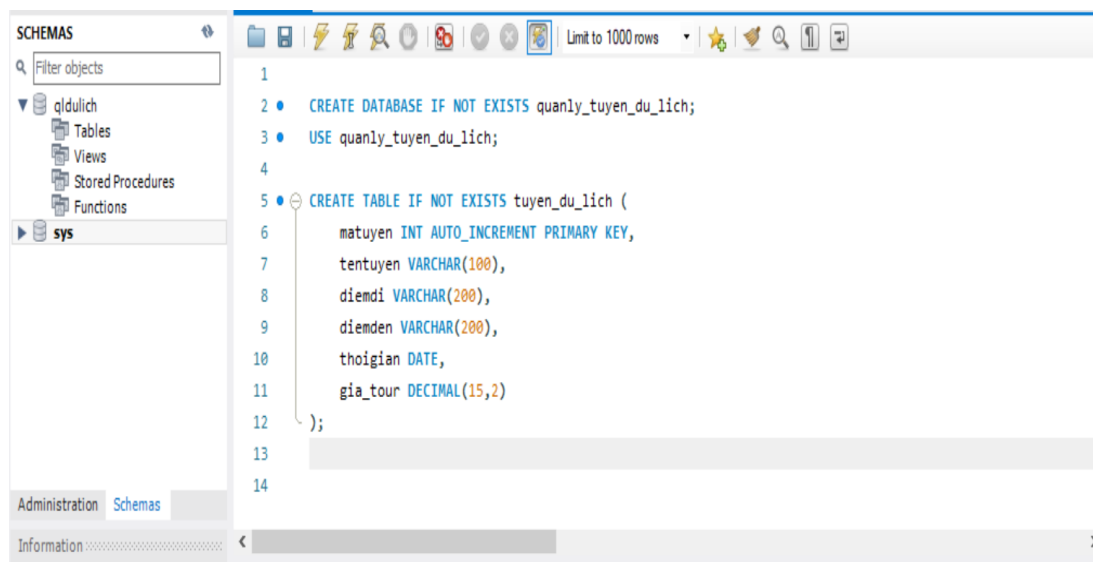
```
pip install mysql-connector-python tkcalendar
```

- Giải thích:

- + mysql-connector-python: dùng để kết nối Python với MySQL.
- + tkcalendar: để chọn ngày sinh bằng DateEntry

### 2. Chuẩn bị CSDL MySQL

- Mở MySQL (WorkBench, phpMyAdmin hoặc terminal) rồi tạo database và bảng :



### 3. Viết code ứng dụng Tkinter + MySQL

```
main.py  qltuyendl.py x
1  import tkinter as tk
2  from tkinter import ttk, messagebox
3  import mysql.connector
4
5
6  def connect_db(): 4 usages
7      return mysql.connector.connect(
8          host="localhost",
9          user="root",
10         password="123456",
11         database="qlnhanvien"
12     )
13
14
15  def center_window(win, w=700, h=500): 1 usage
16      ws = win.winfo_screenwidth()
17      hs = win.winfo_screenheight()
18      x = (ws // 2) - (w // 2)
19      y = (hs // 2) - (h // 2)
20      win.geometry(f'{w}x{h}+{x}+{y}')
```



```

22
23 def clear_input(): 4 usages
24     entry_matuyen.delete(first=0, tk.END)
25     entry_tentuyen.delete(first=0, tk.END)
26     entry_diemdi.delete(first=0, tk.END)
27     entry_thoigian.delete(first=0, tk.END)
28     entry_giatien.delete(first=0, tk.END)
29     cbb_phuongtien.set("")
30
31
32 def load_data(): 4 usages
33     for i in tree.get_children():
34         tree.delete(i)
35 > try: ...
42     except Exception as e:
43         messagebox.showerror(title="Lỗi CSDL", message=f"Không thể tải dữ liệu: {str(e)}\nVui lòng kiểm tra kết nối MySQL.")
44
45
46 def them_tuyendulich(): 1 usage
47     matuyen = entry_matuyen.get()
48     tentuyen = entry_tentuyen.get()
49     diemdi = entry_diemdi.get()
50     thoigian = entry_thoigian.get()
51     giatien = entry_giatien.get()
52     phuongtien = cbb_phuongtien.get()
53
54     if not all([matuyen, tentuyen, diemdi, thoigian, giatien, phuongtien]):
55         messagebox.showwarning(title="Thiếu dữ liệu", message="Vui lòng nhập đủ thông tin Tuyển Du lịch.")
56         return
57
58     try:
59         conn = connect_db()
60         cur = conn.cursor()
61         sql = "INSERT INTO tuyendulich VALUES (%s, %s, %s, %s, %s, %s)"
62         data = (matuyen, tentuyen, diemdi, thoigian, giatien, phuongtien)
63         cur.execute(sql, data)
64         conn.commit()
65         load_data()
66         clear_input()
67     except Exception as e:
68         messagebox.showerror(title="Lỗi Thêm", str(e))
69     finally:
70         conn.close()
71
72
73 def xoa_tuyendulich(): 1 usage
74     selected = tree.selection()
75     if not selected:

```

```

75         if not selected:
76             messagebox.showwarning( title: "Chưa chọn", message: "Hãy chọn tuyến du lịch để xóa.")
77             return
78
79         matuyen = tree.item(selected)["values"][0]
80         try:
81             conn = connect_db()
82             cur = conn.cursor()
83             cur.execute("DELETE FROM tuyendulich WHERE matuyen=%s", (matuyen,))
84             conn.commit()
85             load_data()
86         except Exception as e:
87             messagebox.showerror( title: "Lỗi Xóa", str(e))
88         finally:
89             conn.close()
90
91
92     def sua_tuyendulich(): 1 usage
93         selected = tree.selection()
94         if not selected:
95             messagebox.showwarning( title: "Chưa chọn", message: "Hãy chọn tuyến du lịch để sửa.")
96             return

```

```

97
98         values = tree.item(selected)["values"]
99
100         clear_input()
101         entry_matuyen.insert( index: 0, values[0])
102         entry_tentuyen.insert( index: 0, values[1])
103         entry_diemdi.insert( index: 0, values[2])
104         entry_thoigian.insert( index: 0, values[3])
105         entry_giatien.insert( index: 0, values[4])
106         cbb_phuongtien.set(values[5])
107
108
109     def luu_tuyendulich(): 1 usage
110         matuyen = entry_matuyen.get()
111         tentuyen = entry_tentuyen.get()
112         diemdi = entry_diemdi.get()
113         thoigian = entry_thoigian.get()
114         giatien = entry_giatien.get()
115         phuongtien = cbb_phuongtien.get()
116
117         if not all([matuyen, tentuyen, diemdi, thoigian, giatien, phuongtien]):
118             messagebox.showwarning( title: "Thiếu dữ liệu", message: "Vui lòng nhập đủ thông tin Tuyến Du lịch.")
119             return

```

```

121         try:
122             conn = connect_db()
123             cur = conn.cursor()
124             sql = """UPDATE tuyendulich SET tentuyen=%s, diemdi=%s, thoigian=%s, giatien=%s, phuongtien=%s
125                 WHERE matuyen=%s"""
126             data = (tentuyen, diemdi, thoigian, giatien, phuongtien, matuyen)
127             cur.execute(sql, data)
128             conn.commit()
129             load_data()
130             clear_input()
131         except Exception as e:
132             messagebox.showerror( title: "Lỗi Lưu", str(e))
133         finally:
134             conn.close()
135
136
137     root = tk.Tk()
138     root.title("Quản lý Tuyển Du lịch")
139     center_window(root, w: 700, h: 500)
140     root.resizable( width: False, height: False)
141
142     lbl_title = tk.Label(root, text="QUẢN LÝ TUYỂN DU LỊCH", font=("Arial", 18, "bold"))
143     lbl_title.pack(pady=10)

```

```

144
145     frame_info = tk.Frame(root)
146     frame_info.pack(pady=5, padx=10, fill="x")
147
148     tk.Label(frame_info, text="Mã tuyển").grid(row=0, column=0, padx=5, pady=5, sticky="w")
149     entry_matuyen = tk.Entry(frame_info, width=10)
150     entry_matuyen.grid(row=0, column=1, padx=5, pady=5, sticky="w")
151     tk.Label(frame_info, text="Phương tiện").grid(row=0, column=2, padx=5, pady=5, sticky="w")
152     cbb_phuongtien = ttk.Combobox(frame_info, values=["Máy bay", "Tàu hỏa", "Ô tô", "Xe khách"], width=20)
153     cbb_phuongtien.grid(row=0, column=3, padx=5, pady=5, sticky="w")
154
155     tk.Label(frame_info, text="Tên tuyển").grid(row=1, column=0, padx=5, pady=5, sticky="w")
156     entry_tentuyen = tk.Entry(frame_info, width=25)
157     entry_tentuyen.grid(row=1, column=1, padx=5, pady=5, sticky="w")
158     tk.Label(frame_info, text="Điểm đi").grid(row=1, column=2, padx=5, pady=5, sticky="w")
159     entry_diemdi = tk.Entry(frame_info, width=15)
160     entry_diemdi.grid(row=1, column=3, padx=5, pady=5, sticky="w")
161
162     tk.Label(frame_info, text="Thời gian (ngày)").grid(row=2, column=0, padx=5, pady=5, sticky="w")
163     entry_thoigian = tk.Entry(frame_info, width=10)
164     entry_thoigian.grid(row=2, column=1, padx=5, pady=5, sticky="w")
165     tk.Label(frame_info, text="Giá tiền (VNĐ)").grid(row=2, column=2, padx=5, pady=5, sticky="w")
166     entry_giatien = tk.Entry(frame_info, width=15)
167     entry_giatien.grid(row=2, column=3, padx=5, pady=5, sticky="w")

```

```

68
69 lbl_ds = tk.Label(root, text="Danh sách tuyến du lịch", font=("Arial", 10, "bold"))
70 lbl_ds.pack(pady=5, anchor="w", padx=10)
71
72 columns = ("matuyen", "tentuyen", "diemdi", "thoigian", "giatien", "phuongtien")
73 tree = ttk.Treeview(root, columns=columns, show="headings", height=10)
74
75 tree.column("matuyen", width=60, anchor="center")
76 tree.heading("matuyen", text="Mã Tuyến")
77 tree.column("tentuyen", width=150)
78 tree.heading("tentuyen", text="Tên Tuyến")
79 tree.column("diemdi", width=100)
80 tree.heading("diemdi", text="Điểm Đi")
81 tree.column("thoigian", width=70, anchor="center")
82 tree.heading("thoigian", text="Thời Gian")
83 tree.column("giatien", width=100, anchor="center")
84 tree.heading("giatien", text="Giá Tiền")
85 tree.column("phuongtien", width=150)
86 tree.heading("phuongtien", text="Phương Tiện")
87
88 tree.pack(padx=10, pady=5, fill="both")
89

```

```

188 tree.pack(padx=10, pady=5, fill="both")
189
190 frame_btn = tk.Frame(root)
191 frame_btn.pack(pady=5)
192
193 tk.Button(frame_btn, text="Thêm", width=8, command=them_tuyendulich).grid(row=0, column=0, padx=5)
194 tk.Button(frame_btn, text="Lưu", width=8, command=luu_tuyendulich).grid(row=0, column=1, padx=5)
195 tk.Button(frame_btn, text="Sửa", width=8, command=sua_tuyendulich).grid(row=0, column=2, padx=5)
196 tk.Button(frame_btn, text="Hủy", width=8, command=clear_input).grid(row=0, column=3, padx=5)
197 tk.Button(frame_btn, text="Xóa", width=8, command=xoa_tuyendulich).grid(row=0, column=4, padx=5)
198 tk.Button(frame_btn, text="Thoát", width=8, command=root.quit).grid(row=0, column=5, padx=5)
199
200 load_data()
201 root.mainloop()

```