

TRƯỜNG ĐẠI HỌC AN GIANG
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN
Chuyên đề Python (COS525)

XÂY DỰNG ỨNG DỤNG QUẢN LÝ ĐIỂM SINH VIÊN
VỚI PYTHON, VÀ MYSQL

Giảng viên hướng dẫn: Ths. Nguyễn Ngọc Minh

Sinh viên thực hiện:

- DPM245455.Bùi Thanh Mừng.DH25PM.

NhómTH4_TỔTH2

-DPM245426.DươngTrọng

Hữu.DH25PM.NhómTH4_TỔTH2

AN GIANG,12-2025

1.1. Tính cấp thiết của đề tài

Trong môi trường giáo dục đại học, công tác quản lý điểm số là một trong những nhiệm vụ quan trọng và phức tạp nhất. Với số lượng sinh viên đông, nhiều môn học và các đầu điểm thành phần (chuyên cần, giữa kỳ, cuối kỳ), việc quản lý thủ công hoặc sử dụng Excel đơn thuần thường gặp nhiều rủi ro như: nhập liệu sai sót, khó khăn trong việc tính toán điểm trung bình, và tốn thời gian khi cần tra cứu lịch sử học tập của sinh viên.

Để giải quyết vấn đề này, việc xây dựng một phần mềm chuyên biệt giúp số hóa quy trình quản lý điểm là rất cần thiết. Đề tài "Xây dựng ứng dụng quản lý điểm sinh viên với Python, Tkinter và MySQL" được thực hiện nhằm mục đích tạo ra công cụ hỗ trợ giảng viên và giáo vụ quản lý điểm số chính xác, nhanh chóng và bảo mật.

1.2. Mục tiêu nghiên cứu

Đề tài hướng tới các mục tiêu cụ thể sau:

- Về kiến thức: Vận dụng ngôn ngữ Python và hệ quản trị MySQL để giải quyết bài toán thực tế.
- Về ứng dụng: Xây dựng phần mềm hoàn chỉnh với các chức năng: Quản lý hồ sơ sinh viên (Mã SV, Họ tên, Lớp, Ngành); Quản lý điểm các môn học; Tự động tính điểm tổng kết và xếp loại học lực; Tìm kiếm và thống kê kết quả học tập dễ dàng.

•

2: TỔNG QUAN VÀ CƠ SỞ LÝ THUYẾT

2.1. Đặt vấn đề và bối cảnh

Hiện nay, yêu cầu về độ chính xác và minh bạch trong điểm số tại các trường đại học ngày càng cao. Các hệ thống quản lý đào tạo lớn thường tốn kém và đôi khi quá phức tạp cho nhu cầu quản lý cục bộ của một giảng viên hay một bộ môn. Do đó, nhu cầu về một ứng dụng nhỏ gọn (Desktop Application) để quản lý điểm cá nhân hoặc nhóm lớp là rất thực tế.

2.2. Lịch sử giải quyết vấn đề

Trước đây, giảng viên thường lưu điểm vào sổ tay cá nhân hoặc các file Excel rời rạc. Khi cần tổng hợp điểm cuối kỳ hoặc báo cáo cho nhà trường, việc gom dữ liệu từ nhiều file rất dễ gây nhầm lẫn. Đề tài này đề xuất giải pháp lưu trữ tập trung vào cơ sở dữ liệu MySQL, giúp dữ liệu nhất quán và an toàn hơn so với lưu file rời.

2.3. Phạm vi của đề tài

- Đề tài tập trung xây dựng các chức năng quản lý cốt lõi:
- Đối tượng quản lý: Sinh viên (Thông tin cá nhân), Môn học, và Kết quả học tập (Điểm số).
- Chức năng: Thêm, Sửa, Xóa sinh viên và điểm số; Tự động tính điểm tổng kết môn học; Tìm kiếm điểm theo Mã sinh viên hoặc Tên

- Giới hạn: Chưa tích hợp chức năng đăng ký môn học trực tuyến hay cổng thông tin sinh viên.

2.4. Phương pháp nghiên cứu

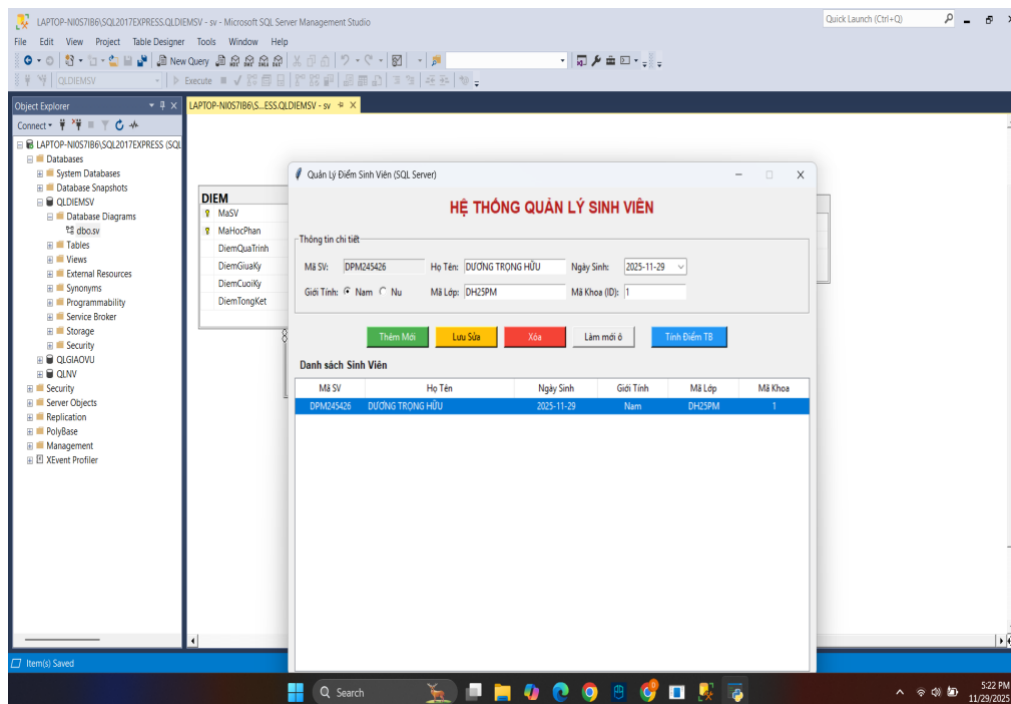
- Phương pháp nghiên cứu lý thuyết: Tìm hiểu về cấu trúc dữ liệu quan hệ (Relational Database) để thiết kế bảng Điểm và bảng Sinh viên sao cho tối ưu.
- Phương pháp nghiên cứu thực nghiệm: Sử dụng thư viện Tkinter để thiết kế giao diện nhập điểm dạng bảng và mysql-connector để truy vấn dữ liệu.

2.5. Cơ sở lý thuyết

- Ngôn ngữ Python: Được chọn lựa nhờ cú pháp đơn giản và khả năng xử lý tính toán số học mạnh mẽ, rất phù hợp cho việc xử lý điểm số và thống kê.
- Thư viện Tkinter: Được sử dụng để xây dựng giao diện người dùng (GUI), tạo ra các Form nhập điểm thân thiện và bảng hiển thị danh sách sinh viên trực quan.
- Hệ quản trị MySQL: Đóng vai trò là kho lưu trữ dữ liệu bền vững, đảm bảo an toàn dữ liệu sinh viên và điểm số.

3. KẾT QUẢ ĐẠT ĐƯỢC VÀ DEMO CODE

A. THIẾT KẾ GIAO DIỆN QUẢN LÝ NHÂN VIÊN



HỆ THỐNG QUẢN LÝ ĐIỂM SINH VIÊN

QUẢN LÝ HỒ SƠ SINH VIÊN QUẢN LÝ BẢNG ĐIỂM & KẾT QUẢ

Nhập liệu

Mã SV: Họ Tên: Ngày Sinh:

Giới Tính: Mã Lớp: Mã Khoa:

MaSV	HoTen	NgaySinh	GioiTinh	
DPM245426	DƯƠNG TRỌNG HỮU	2025-11-29	Nam	DH25PM
ẻw	eger	2025-11-29	Nam	1

HỆ THỐNG QUẢN LÝ ĐIỂM SINH VIÊN

QUẢN LÝ HỒ SƠ SINH VIÊN QUẢN LÝ BẢNG ĐIỂM & KẾT QUẢ

Tim tên SV:

Mã SV	Họ Tên	Môn Học	QT (20%)	GK (30%)	CK (50%)	T.Kết	Kết Quả
DPM245428	DƯƠNG TRỌNG HỮU	Tin học đại cương	5.00	5.00	5.00	5.00	ĐẦU

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

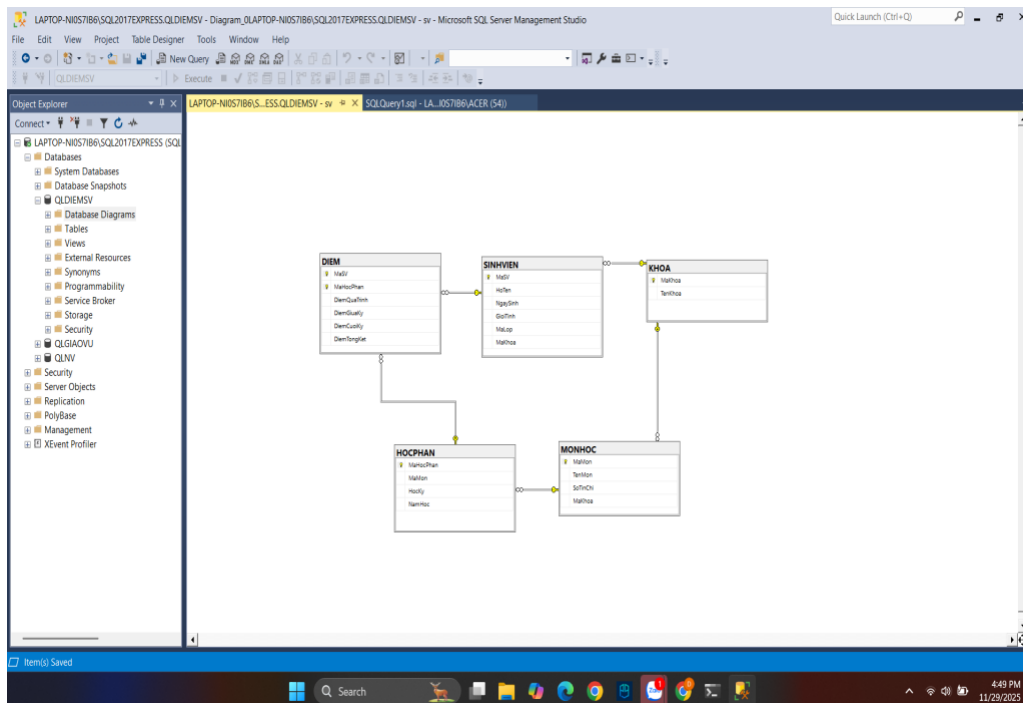
thực hiện các thao tác như thêm, xóa, sửa, và hiển thị danh sách Nhân Viên.

1. Cài đặt các thư viện cần thiết:

- Python: Đảm bảo bạn đã cài đặt Python trên máy tính 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

(**pip install mysql-connector-python**)

2. Thiết kế cơ sở dữ liệu MySQL



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

Để đáp ứng yêu cầu quản lý nhân viên, giao diện ứng dụng được xây dựng bằng thư viện **Tkinter** của Python, chia thành 3 khu vực chức năng chính để đảm bảo tính trực quan và dễ sử dụng:

a. Khu vực nhập liệu (Frame Thông tin):

- Sử dụng các widget Label và Entry để nhận thông tin: Mã số, Họ và tên lót, Tên.
- Sử dụng Radiobutton cho trường giới tính (Nam/Nữ).

- Sử dụng Combobox (thư viện ttk) để chọn Chức vụ từ danh sách có sẵn (Trưởng phòng, Kế toán, Lái xe,...).
- Sử dụng DateEntry (thư viện tkcalendar) để hỗ trợ người dùng chọn Ngày sinh từ lịch, đảm bảo định dạng dữ liệu ngày tháng chính xác.

b. Khu vực điều khiển (Frame Nút bấm): Hệ thống cung cấp các nút chức năng (Button) tương ứng với các thao tác nghiệp vụ :

- **Thêm:** Thu thập dữ liệu từ các ô nhập và lưu nhân viên mới vào CSDL.
- **Lưu:** Cập nhật thông tin nhân viên sau khi chỉnh sửa.
- **Sửa:** Lấy thông tin nhân viên đang chọn hiển thị lên ô nhập để chỉnh sửa.
- **Xóa:** Xóa nhân viên được chọn khỏi hệ thống.
- **Hủy (Reset):** Làm trống các ô nhập liệu để chuẩn bị nhập mới.
- **Tìm kiếm:** Cho phép lọc nhân viên theo Tên hoặc Mã số.
- **Thoát:** Đóng chương trình.

c. Khu vực hiển thị danh sách (Treeview):

- Sử dụng widget Treeview để hiển thị toàn bộ danh sách nhân viên dưới dạng bảng.
- Các cột hiển thị bao gồm: Mã số, Họ lót, Tên, Phái, Ngày sinh, Chức vụ.
- Tích hợp sự kiện "chọn dòng" để người dùng có thể click vào nhân viên và thực hiện thao tác Sửa hoặc Xóa.

C. CÀI ĐẶT & CODE ỨNG DỤNG

```

1  import tkinter as tk
2  from tkinter import ttk, messagebox
3  from tkcalendar import DateEntry
4  import pyodbc
5
6
7  # ===== 1. CẤU HÌNH KẾT NỐI =====
8  def connect_db():
9      try:
10         conn = pyodbc.connect(
11             "DRIVER={SQL Server};"
12             "SERVER=LAPTOP-NI0S7IB6\SQL2017EXPRESS;" # <--- HÃY SỬA TÊN SERVER CỦA BẠN Ở ĐÂY
13             "DATABASE=QLDIEMSV;"
14             "Trusted_Connection=yes;"
15         )
16         return conn
17     except Exception as e:
18         messagebox.showerror("Lỗi Kết Nối", f"Lỗi: {e}")
19         return None
20
21
22  # =====
23  # TAB 1: LOGIC QUẢN LÝ SINH VIÊN
24  # =====
25  def load_sv(tree):
26      for i in tree.get_children(): tree.delete(i)
27      conn = connect_db()
28      if conn:
29          cur = conn.cursor()
30          cur.execute("SELECT MaSV, HoTen, NgaySinh, GioiTinh, MaLop, MaKhoa FROM SINHVIEN")
31          for row in cur.fetchall(): tree.insert("", tk.END, values=list(row))
32          conn.close()

```

```

35 def them_sv(tree, e_ma, e_ten, e_ngay, e_gt, e_lop, e_khoa):
36     masv, hoten = e_ma.get(), e_ten.get()
37     ngaysinh = str(e_ngay.get_date())
38     makhoa = e_khoa.get()
39
40     if not masv or not hoten: return messagebox.showwarning("Lỗi", "Thiếu Mã SV hoặc Tên")
41
42     conn = connect_db()
43     if conn:
44         try:
45             cur = conn.cursor()
46             sql = "INSERT INTO SINHVIEN (MaSV, HoTen, NgaySinh, GioiTinh, MaLop, MaKhoa) VALUE
47             cur.execute(sql, (masv, hoten, ngaysinh, e_gt.get(), e_lop.get(), makhoa))
48             conn.commit()
49             messagebox.showinfo("Thành công", "Đã thêm SV mới")
50             load_sv(tree)
51         except Exception as e:
52             messagebox.showerror("Lỗi SQL", str(e))
53         finally:
54             conn.close()
55
56
57 def xoa_sv(tree):
58     sel = tree.selection()
59     if not sel: return
60     masv = tree.item(sel)["values"][0]
61     if messagebox.askyesno("Xóa", f"Xóa SV {masv} sẽ mất hết điểm của SV này. Tiếp tục?"):
62         conn = connect_db()
63         if conn:
64             cur = conn.cursor()
65             cur.execute("DELETE FROM DIEM WHERE MaSV=?", (masv,)) # Xóa điểm trước
66             cur.execute("DELETE FROM SINHVIEN WHERE MaSV=?", (masv,)) # Xóa SV sau
67             conn.commit()

```



```

57 def xoa_sv(tree):
58     conn.close()
59     load_sv(tree)
60     messagebox.showinfo("Xong", "Đã xóa!")
61
62
63 # =====
64 # TAB 2: LOGIC QUẢN LÝ ĐIỂM
65 # =====
66 def load_diem(tree):
67     (variable) conn: Any | None tree.delete(i)
68     conn = connect_db()
69     if conn:
70         try:
71             cur = conn.cursor()
72             # Kỹ thuật JOIN bảng để lấy Tên Môn thay vì chỉ hiện ID
73             sql = """
74             SELECT
75             SV.MaSV, SV.HoTen,
76             MH.TenMon,
77             D.DiemQuaTrinh, D.DiemGiuaKy, D.DiemCuoiKy, D.DiemTongKet,
78             HP.MaHocPhan
79             FROM DIEM D
80             JOIN SINHVIEN SV ON D.MaSV = SV.MaSV
81             JOIN HOCPHAN HP ON D.MaHocPhan = HP.MaHocPhan
82             JOIN MONHOC MH ON HP.MaMon = MH.MaMon
83             """
84             cur.execute(sql)
85             for row in cur.fetchall():
86                 # Xử lý xếp loại đơn giản
87                 tong = row[6] if row[6] else 0
88                 ketqua = "ĐẠU" if tong >= 4.0 else "RỐT"
89

```

```

76 def load_diem(tree):
77     # Chèn vào bảng (Thêm cột Kết quả vào cuối)
78     data = list(row)
79     data.insert(7, ketqua) # Chèn chữ Đậu/Rốt trước cột Mã HP ẩn
80     tree.insert("", tk.END, values=data)
81 except Exception as e:
82     messagebox.showerror("Lỗi Tải Điểm", str(e))
83 finally:
84     conn.close()
85
86
87 def tim_kiem_diem(tree, keyword):
88     for i in tree.get_children(): tree.delete(i)
89     conn = connect_db()
90     if conn:
91         cur = conn.cursor()
92         sql = """
93         SELECT SV.MaSV, SV.HoTen, MH.TenMon, D.DiemQuaTrinh, D.DiemGiuaKy, D.DiemCuoiKy, D.DiemTongKet, HP.MaHocPhan
94         FROM DIEM D
95         JOIN SINHVIEN SV ON D.MaSV = SV.MaSV
96         JOIN HOCPHAN HP ON D.MaHocPhan = HP.MaHocPhan
97         JOIN MONHOC MH ON HP.MaMon = MH.MaMon
98         WHERE SV.HoTen LIKE ? OR SV.MaSV LIKE ?
99         """
100         cur.execute(sql, (f'%' + keyword + '%', f'%' + keyword + '%'))
101         for row in cur.fetchall():
102             tong = row[6] if row[6] else 0
103             ketqua = "ĐẠU" if tong >= 4.0 else "RỐT"
104             data = list(row)
105             data.insert(7, ketqua)
106             tree.insert("", tk.END, values=data)

```

```

110 def tim_kiem_diem(tree, keyword):
111     conn.close()
112
113
114 def popup_nhap_diem(parent_root, tree_diem):
115     win = tk.Toplevel(parent_root)
116     win.title("Nhập / Sửa Điểm")
117     win.geometry("400x500")
118
119     tk.Label(win, text="NHẬP ĐIỂM SINH VIÊN", font=("bold", 14), fg="blue").pack(pady=10)
120
121     # Form nhập
122     f = tk.Frame(win);
123     f.pack(pady=5)
124
125     tk.Label(f, text="Mã Sinh Viên:").grid(row=0, column=0, pady=5, sticky="e")
126     e_msv = tk.Entry(f);
127     e_msv.grid(row=0, column=1)
128
129     tk.Label(f, text="Mã Học Phần (ID):").grid(row=1, column=0, pady=5, sticky="e")
130     e_mhp = tk.Entry(f);
131     e_mhp.grid(row=1, column=1)
132     tk.Label(f, text="(Xem bảng HOCPHAN trong SQL)", font=("Arial", 7), fg="gray").grid(row=2, column=1)
133
134     tk.Label(f, text="Điểm Quá Trình (20%):").grid(row=3, column=0, pady=5, sticky="e")
135     e_qt = tk.Entry(f);
136     e_qt.grid(row=3, column=1)
137
138     tk.Label(f, text="Điểm Giữa Kỳ (30%):").grid(row=4, column=0, pady=5, sticky="e")
139     e_gk = tk.Entry(f);
140     e_gk.grid(row=4, column=1)
141
142     tk.Label(f, text="Điểm Cuối Kỳ (50%):").grid(row=5, column=0, pady=5, sticky="e")

```

```

C:\Users\PC\Documents> DEMO_PYTHON.py popup_nhap_diem
134 def popup_nhap_diem(parent_root, tree_diem):
135     e_ck = tk.Entry(f);
136     e_ck.grid(row=5, column=1)
137     lbl_kq = tk.Label(win, text="...", font=("bold", 12), fg="red")
138     lbl_kq.pack(pady=10)
139
140     def xu_ly_luu():
141         msv, mhp = e_msv.get(), e_mhp.get()
142         try:
143             qt, gk, ck = float(e_qt.get()), float(e_gk.get()), float(e_ck.get())
144         except:
145             return messagebox.showerror("Lỗi", "Điểm phải là số!")
146
147         if not (0 <= qt <= 10 and 0 <= gk <= 10 and 0 <= ck <= 10):
148             return messagebox.showerror("Lỗi", "Điểm từ 0-10")
149
150         tong = round(qt * 0.2 + gk * 0.3 + ck * 0.5, 2)
151         lbl_kq.config(text=f"Tổng kết: {tong}")
152
153         conn = connect_db()
154         if conn:
155             try:
156                 cur = conn.cursor()
157
158                 cur.execute("SELECT * FROM DIEM WHERE MaSV=? AND MaHocPhan=?", (msv, mhp))
159                 if cur.fetchone():
160                     sql = "UPDATE DIEM SET DiemQuaTrinh=?, DiemGiuaky=?, DiemCuoiKy=?, DiemTongKet=? WHERE MaSV=? AND MaHocPhan=?"
161                     cur.execute(sql, (qt, gk, ck, tong, msv, mhp))
162                 else:
163                     sql = "INSERT INTO DIEM (MaSV, MaHocPhan, DiemQuaTrinh, DiemGiuaky, DiemCuoiKy, DiemTongKet) VALUES (?, ?, ?, ?, ?, ?)"
164                     cur.execute(sql, (msv, mhp, qt, gk, ck, tong))
165                 conn.commit()

```

```

194         messagebox.showinfo("OK", "Đã lưu điểm!")
195         load_diem(tree_diem) # Load lại bảng điểm ở giao diện chính
196         win.destroy()
197     except Exception as e:
198         messagebox.showerror("Lỗi SQL", f"Kiểm tra lại Mã SV hoặc Mã HP.\n{e}")
199     finally:
200         conn.close()
201
202     tk.Button(win, text="LƯU ĐIỂM", bg="blue", fg="white", font=("bold", 10), command=xu_ly_luu).pack(pady=5)
203
204
205 # =====
206 # GIAO DIỆN CHÍNH (MAIN WINDOW)
207 # =====
208 root = tk.Tk()
209 root.title("HỆ THỐNG QUẢN LÝ ĐIỂM SINH VIÊN")
210 root.geometry("900x600")
211
212 # --- TẠO TAB (NOTEBOOK) ---
213 notebook = ttk.Notebook(root)
214 notebook.pack(pady=10, expand=True, fill="both")
215
216 # Tạo 2 khung cho 2 tab
217 tab1 = tk.Frame(notebook)
218 tab2 = tk.Frame(notebook)
219
220 notebook.add(tab1, text="QUẢN LÝ HỒ SƠ SINH VIÊN ")
221 notebook.add(tab2, text="QUẢN LÝ BẢNG ĐIỂM & KẾT QUẢ ")
222
223 # -----

```

```

224 # THIẾT KẾ TAB 1: SINH VIÊN
225 # -----
226 f1 = tk.LabelFrame(tab1, text="Nhập liệu")
227 f1.pack(fill="x", padx=10, pady=5)
228
229 tk.Label(f1, text="Mã SV:").grid(row=0, column=0);
230 e1_ma = tk.Entry(f1);
231 e1_ma.grid(row=0, column=1)
232 tk.Label(f1, text="Họ Tên:").grid(row=0, column=2);
233 e1_ten = tk.Entry(f1);
234 e1_ten.grid(row=0, column=3)
235 tk.Label(f1, text="Ngày Sinh:").grid(row=0, column=4);
236 e1_ngay = DateEntry(f1, date_pattern="yyyy-mm-dd");
237 e1_ngay.grid(row=0, column=5)
238 tk.Label(f1, text="Giới Tính:").grid(row=1, column=0);
239 e1_gt = ttk.Combobox(f1, values=["Nam", "Nu"], width=17);
240 e1_gt.grid(row=1, column=1);
241 e1_gt.set("Nam")
242 tk.Label(f1, text="Mã Lớp:").grid(row=1, column=2);
243 e1_lop = tk.Entry(f1);
244 e1_lop.grid(row=1, column=3)
245 tk.Label(f1, text="Mã Khoa:").grid(row=1, column=4);
246 e1_khoa = tk.Entry(f1);
247 e1_khoa.grid(row=1, column=5)
248
249 btn_f1 = tk.Frame(tab1);
250 btn_f1.pack(pady=5)
251 tk.Button(btn_f1, text="Thêm SV", bg="green", fg="white",
252           command=lambda: them_sv(tree1, e1_ma, e1_ten, e1_ngay, e1_gt, e1_lop, e1_khoa)).pack(side="left", padx=5)
253 tk.Button(btn_f1, text="Xóa SV", bg="red", fg="white", command=lambda: xoa_sv(tree1)).pack(side="left", padx=5)
254 tk.Button(btn_f1, text="Tải lại danh sách", command=lambda: load_sv(tree1)).pack(side="left", padx=5)

```

```

255     cols1 = ("MaSV", "HoTen", "NgaySinh", "GioiTinh", "MaLop", "Makhoa")
256     tree1 = ttk.Treeview(tab1, columns=cols1, show="headings", height=15)
257     for c in cols1: tree1.heading(c, text=c)
258     tree1.pack(fill="both", expand=True, padx=10, pady=5)
259     load_sv(tree1)
260
261     # -----
262     # TAB 2: BẢNG ĐIỂM
263     # -----
264     f2_top = tk.Frame(tab2);
265     f2_top.pack(fill="x", padx=10, pady=10)
266
267     tk.Button(f2_top, text="+ NHẬP ĐIỂM MỚI", bg="blue", fg="white", font=("bold", 10), height=2,
268             command=lambda: popup_nhap_diem(root, tree2)).pack(side="left", padx=5)
269
270     tk.Label(f2_top, text="Tìm tên SV:").pack(side="left", padx=(30, 5))
271     e2_tim = tk.Entry(f2_top)
272     e2_tim.pack(side="left")
273     tk.Button(f2_top, text="Tìm kiếm", command=lambda: tim_kiem_diem(tree2, e2_tim.get())).pack(side="left", padx=5)
274     tk.Button(f2_top, text="Hiện tất cả", command=lambda: load_diem(tree2)).pack(side="left", padx=5)
275
276     # Bảng điểm chi tiết
277     cols2 = ("MaSV", "HoTen", "TenMon", "DiemQT", "DiemGK", "DiemCK", "TongKet", "KetQua")
278     tree2 = ttk.Treeview(tab2, columns=cols2, show="headings", height=15)
279
280     tree2.heading("MaSV", text="Mã SV")
281     tree2.heading("HoTen", text="Họ Tên")
282     tree2.heading("TenMon", text="Môn Học")
283     tree2.heading("DiemQT", text="QT (20%)")
284     tree2.heading("DiemGK", text="GK (30%)")
285     tree2.heading("DiemCK", text="CK (50%)")

```

```

286     tree2.heading("DiemCK", text="CK (50%)")
287     tree2.heading("TongKet", text="T. Kết")
288     tree2.heading("KetQua", text="Kết Quả")
289
290     tree2.column("MaSV", width=80)
291     tree2.column("HoTen", width=150)
292     tree2.column("TenMon", width=150)
293     tree2.column("DiemQT", width=60, anchor="center")
294     tree2.column("DiemGK", width=60, anchor="center")
295     tree2.column("DiemCK", width=60, anchor="center")
296     tree2.column("TongKet", width=60, anchor="center")
297     tree2.column("KetQua", width=80, anchor="center")
298
299     tree2.pack(fill="both", expand=True, padx=10, pady=5)
300     load_diem(tree2)
301
302     root.mainloop()

```