

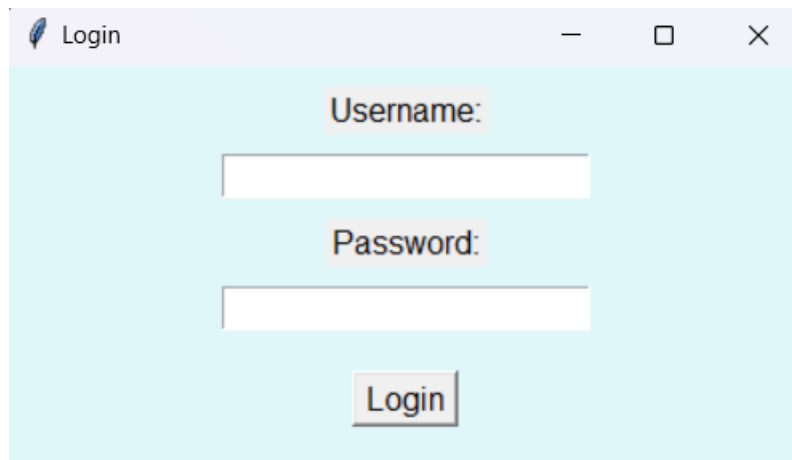
Bài Báo Cáo

Tạo Giao Diện Kết Nối Với SQL

Họ và Tên: Châu Gia Kiệt – MSSV: 2274802010449

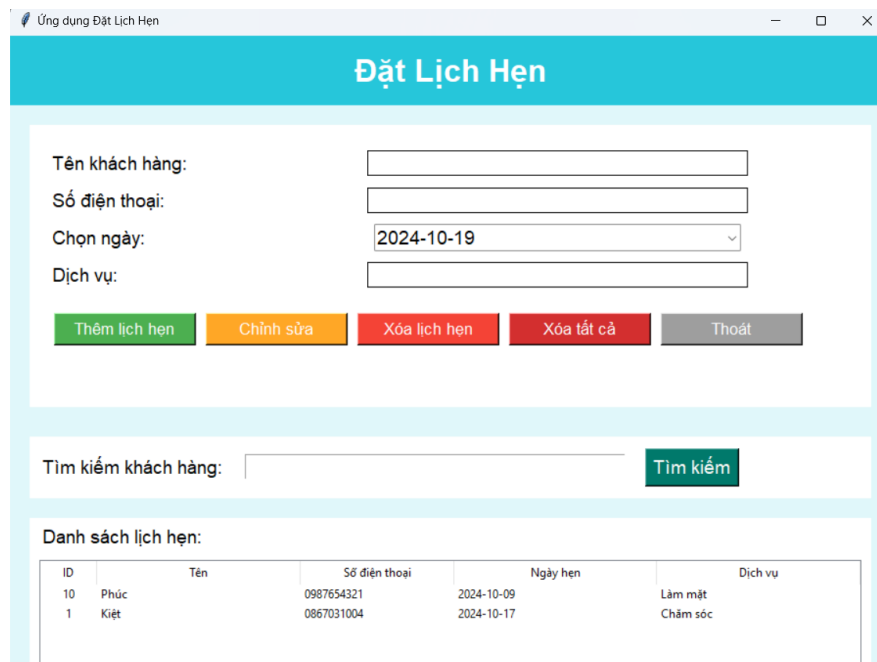
Phần 1. Giao Diện

1.1 Giao diện đăng nhập



A login form interface with a light blue background. It features a title bar with a feather icon and the text 'Login'. The form contains two input fields: 'Username:' and 'Password:'. Below the password field is a 'Login' button.

1.2 Giao diện chính

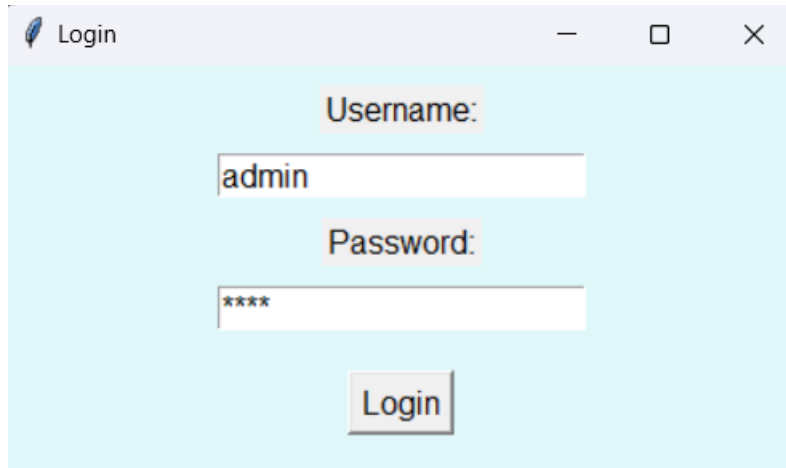


A main application interface titled 'Ứng dụng Đặt Lịch Hẹn' (Appointment Scheduling Application). The header is blue with the text 'Đặt Lịch Hẹn'. The main content area has a light blue background and contains several input fields: 'Tên khách hàng:', 'Số điện thoại:', 'Chọn ngày:' (with a dropdown menu showing '2024-10-19'), and 'Dịch vụ:'. Below these fields are five buttons: 'Thêm lịch hẹn' (green), 'Chỉnh sửa' (orange), 'Xóa lịch hẹn' (red), 'Xóa tất cả' (red), and 'Thoát' (grey). At the bottom, there is a search bar with the text 'Tìm kiếm khách hàng:' and a 'Tìm kiếm' button. Below the search bar is a table titled 'Danh sách lịch hẹn:' (Appointment List) with columns: ID, Tên, Số điện thoại, Ngày hẹn, and Dịch vụ. The table contains two rows of data.

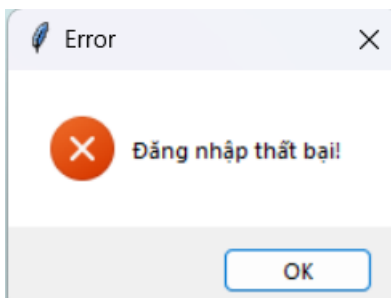
ID	Tên	Số điện thoại	Ngày hẹn	Dịch vụ
10	Phúc	0987654321	2024-10-09	Làm mặt
1	Kiệt	0867031004	2024-10-17	Chăm sóc

Phần 2. Chức Năng

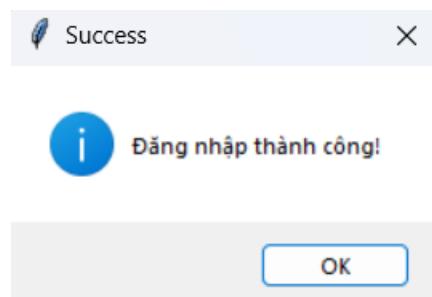
2.1 Chức năng đăng nhập

A screenshot of a web application's login window. The window has a title bar with a feather icon, the text "Login", and standard minimize, maximize, and close buttons. The main area has a light blue background. It contains two labels: "Username:" and "Password:". Below "Username:" is a text input field containing the text "admin". Below "Password:" is a password input field with four asterisks "****". At the bottom center is a "Login" button.

- Nếu đăng nhập sai thì hệ thống sẽ hiện thị thông báo rằng “Đăng nhập thất bại”.



- Nếu đăng nhập đúng thì hệ thống sẽ hiện hộp thoại thông báo rằng “Đăng nhập thành công”



2.2 Chức năng giao diện chính

- Sử dụng Tooltip để hiển thị người dùng biết những gì có trong giao diện.

Đây là phần tiêu đề của ứng dụng đặt lịch hẹn

Đặt Lịch Hẹn

Tên khách hàng:

Nhập tên khách hàng

Số điện thoại:

Nhập số điện thoại khách hàng

Chọn ngày:

Chọn ngày hẹn

Dịch vụ:

Nhập dịch vụ mà khách hàng yêu cầu

- Sử dụng DateEntry để tạo chọn ngày

Chọn ngày:

Dịch vụ:

Tìm kiếm khách hàng:

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
40	30	1	2	3	4	5	6
41	7	8	9	10	11	12	13
42	14	15	16	17	18	19	20
43	21	22	23	24	25	26	27
44	28	29	30	31	1	2	3
45	4	5	6	7	8	9	10

- Thêm lịch hẹn:** Khi điền đầy đủ thông, ấn chọn thêm lịch hẹn thì sẽ được lưu vào cơ sở dữ liệu.

Ứng dụng Đặt Lịch Hẹn

Đặt Lịch Hẹn

Tên khách hàng:

Số điện thoại:

Chọn ngày:

Dịch vụ: Chọn ngày hẹn

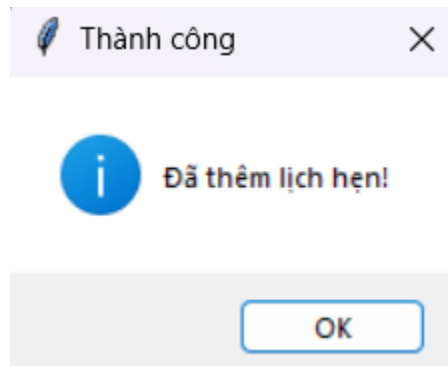
Thêm lịch hẹn **Chỉnh sửa** **Xóa lịch hẹn** **Xóa tất cả** **Thoát**

Tìm kiếm khách hàng: **Tìm kiếm**

Danh sách lịch hẹn:

ID	Tên	Số điện thoại	Ngày hẹn	Dịch vụ
10	Phúc	0987654321	2024-10-09	Làm mắt
1	Kiệt	0867031004	2024-10-17	Chăm sóc
12	Bình	0123456789	2024-10-23	Làm nail

Khi lưu thành công sẽ hiển thị hộp thông báo “Đã thêm lịch hẹn”.



- **Chỉnh sửa lịch hẹn:** Ấn vào người dùng muốn chỉnh sửa và nhấn chọn chỉnh sửa sẽ hiện ra giao diện chỉnh sửa người dùng.

Tên khách hàng:

Bình

Số điện thoại:

123456789

Chọn ngày:

2024-10-23

Dịch vụ:

Làm nail

Thêm lịch hẹn

Chỉnh sửa

Xóa lịch hẹn

Xóa tất cả

Thoát

Tìm kiếm khách hàng:

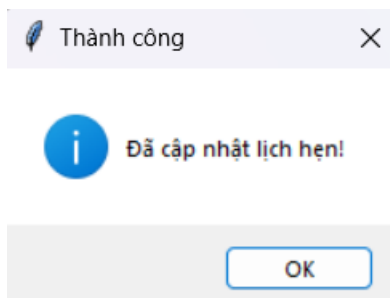
Tìm kiếm

Danh sách lịch hẹn:

ID	Tên	Số điện thoại	Ngày hẹn	Dịch vụ
10	Phúc	0987654321	2024-10-09	Làm mặt
1	Kiệt	0867031004	2024-10-17	Chăm sóc
12	Bình	0123456789	2024-10-23	Làm nail

Lưu thay đổi

Khi ấn chọn “Lưu thay đổi” sẽ hiện ra hộp thoại thông báo “Đã cập nhật lịch hẹn”.

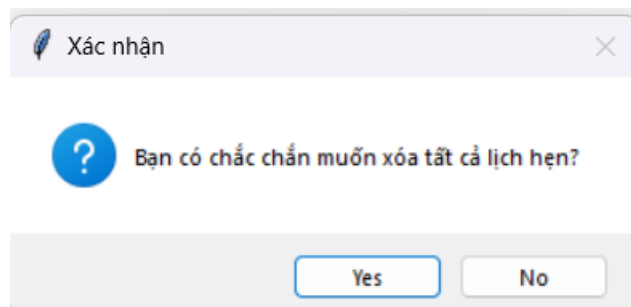


- **Xóa lịch hẹn:** Chọn vào người dùng muốn xóa lịch hẹn, ấn “Xóa lịch hẹn” hộp thoại hiển thị thông báo “Đã xóa lịch hẹn” xuất hiện.

The screenshot shows a top navigation bar with five buttons: "Thêm lịch hẹn" (green), "Chỉnh sửa" (orange), "Xóa lịch hẹn" (red, highlighted with a red box and an arrow), "Xóa tất cả" (dark red), and "Thoát" (grey). Below the buttons is a search section with the text "Tìm kiếm khách hàng:", a text input field, and a green "Tìm kiếm" button. A modal dialog titled "Thành công" (Success) is open, displaying an information icon, the text "Đã xóa lịch hẹn!" (Appointment deleted!), and an "OK" button. Below the dialog is a section titled "Danh sách lịch hẹn:" (Appointment list:) followed by a table.

ID	Tên	Số điện thoại	Ngày hẹn	Dịch vụ
10	Phúc	0987654321	2024-10-09	Làm mặt
1	Kiệt	0867031004	2024-10-17	Chăm sóc
12	Bình	123456789	2024-10-24	Làm nail

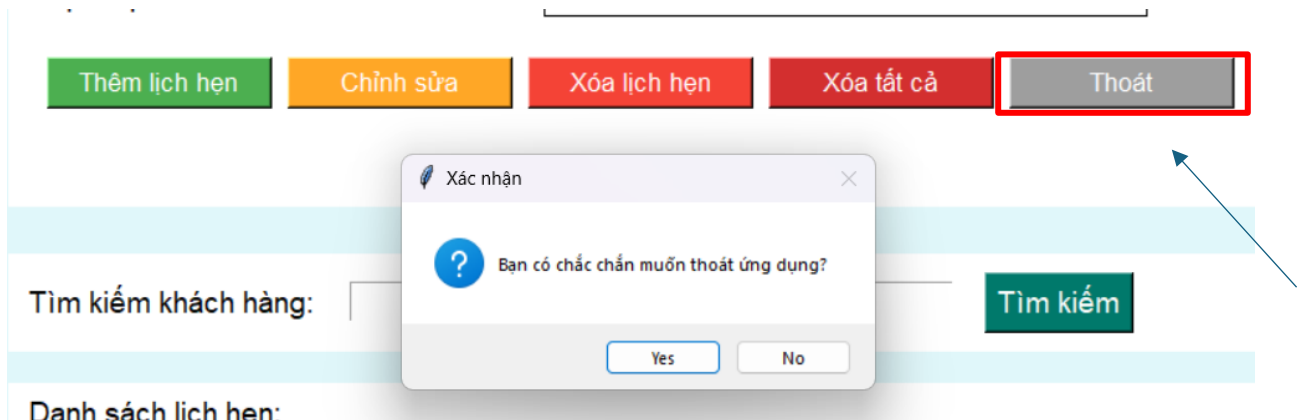
- **Xóa tất cả:** Khi nhấn xóa tất cả hộp thoại thông báo “Bạn có chắc chắn muốn xóa tất cả lịch hẹn?” xuất hiện. Nếu ấn “Yes” thì sẽ xóa tất cả, còn nếu “No” thì sẽ đóng hộp thoại.



Khi ấn “Yes” sẽ xóa tất cả và hiển thị thông báo “Đã xóa tất cả lịch hẹn”.

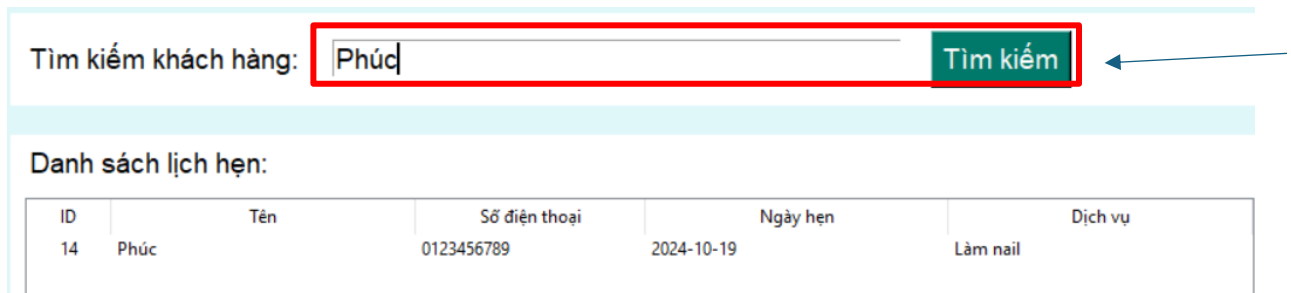
This screenshot is similar to the first one but shows the "Xóa tất cả" (Delete all) button highlighted with a red box and an arrow. The modal dialog is titled "Thành công" (Success) and displays "Đã xóa tất cả lịch hẹn!" (All appointments deleted!). The table below the dialog is empty, indicating that all appointments have been removed.

- **Thoát:** như chức năng của đăng xuất, sẽ hiển thị thông báo “Bạn có chắc chắn muốn thoát khỏi ứng dụng?”. Khi ấn “Yes” ứng dụng sẽ thoát.



Danh sách lịch hẹn:

- **Tìm kiếm:** nhập tên người dùng cần tìm vào ô tìm kiếm, sẽ xuất hiện tên người dùng cần tìm



Danh sách lịch hẹn:

ID	Tên	Số điện thoại	Ngày hẹn	Dịch vụ
14	Phúc	0123456789	2024-10-19	Làm nail

Phần 3. Mã Chương Trình

Link GitHub:

https://github.com/dustin314/Python_Programming/tree/main/Projects/BaiTap2

- Controllers.py

```
1 # Import các lớp và thư viện cần thiết
2 from models import Database # Import lớp Database từ models.py để thao tác với cơ sở dữ liệu
3 from views import LoginView, MainView # Import các lớp giao diện LoginView và MainView từ views.py
4 from tkinter import Tk # Import Tk từ thư viện tkinter để tạo cửa sổ ứng dụng
5
6 # Lớp Controller để điều khiển luồng dữ liệu giữa Model (Database) và View (LoginView, MainView)
7 class Controller:
8     def __init__(self):
9         # Khởi tạo đối tượng Database để kết nối và thao tác với cơ sở dữ liệu
10         self.database = Database()
11
12         # Khởi tạo cửa sổ chính của ứng dụng
13         self.root = Tk()
14
15         # Hiển thị giao diện đăng nhập đầu tiên
16         self.show_login_view()
17
18         # Bắt đầu vòng lặp chính của tkinter để ứng dụng có thể chạy và hiển thị
19         self.root.mainloop()
20
21 # Hàm xác thực người dùng
22 def verify_user(self, username, password):
23     # Gọi phương thức verify_user từ lớp Database để kiểm tra thông tin đăng nhập
24     return self.database.verify_user(username, password)
25
26 # Hàm hiển thị giao diện đăng nhập
27 def show_login_view(self):
28     # Khởi tạo LoginView và truyền vào đối tượng root và controller
29     LoginView(self.root, self)
30
31 # Hàm hiển thị giao diện chính của ứng dụng sau khi đăng nhập thành công
32 def show_main_view(self):
33     # Đặt lại cửa sổ root để mở một cửa sổ chính mới và hiển thị MainView
34     self.root = Tk()
35     MainView(self.root, self)
36
37 # Hàm thêm một lịch hẹn mới vào cơ sở dữ liệu
38 def add_appointment(self, name, phone, appointment_date, service):
39     # Gọi phương thức add_appointment từ lớp Database
40     self.database.add_appointment(name, phone, appointment_date, service)
41
42 # Hàm lấy danh sách tất cả lịch hẹn
43 def get_appointments(self):
44     # Gọi phương thức get_appointments từ lớp Database và trả về kết quả
45     return self.database.get_appointments()
46
47 # Hàm xóa một lịch hẹn dựa trên ID
48 def delete_appointment(self, appointment_id):
49     # Gọi phương thức delete_appointment từ lớp Database với ID của lịch hẹn cần xóa
50     self.database.delete_appointment(appointment_id)
51
52 # Các phương thức bổ sung cho Controller để xử lý các hành động từ giao diện có thể được thêm vào đây
53
54 # Đoạn mã này kiểm tra nếu tập tin được chạy trực tiếp, Controller sẽ được khởi chạy
55 if __name__ == "__main__":
56     Controller()
```

- Models.py

```
1 # Import các lớp và thư viện cần thiết
2 from models import Database # Import lớp Database từ models.py để thao tác với cơ sở dữ liệu
3 from views import LoginView, MainView # Import các lớp giao diện LoginView và MainView từ views.py
4 from tkinter import Tk # Import Tk từ thư viện tkinter để tạo cửa sổ ứng dụng
5
6 # Lớp Controller để điều khiển luồng dữ liệu giữa Model (Database) và View (LoginView, MainView)
7 class Controller:
8     def __init__(self):
9         # Khởi tạo đối tượng Database để kết nối và thao tác với cơ sở dữ liệu
10         self.database = Database()
11
12         # Khởi tạo cửa sổ chính của ứng dụng
13         self.root = Tk()
14
15         # Hiển thị giao diện đăng nhập đầu tiên
16         self.show_login_view()
17
18         # Bắt đầu vòng lặp chính của tkinter để ứng dụng có thể chạy và hiển thị
19         self.root.mainloop()
20
21 # Hàm xác thực người dùng
22 def verify_user(self, username, password):
23     # Gọi phương thức verify_user từ lớp Database để kiểm tra thông tin đăng nhập
24     return self.database.verify_user(username, password)
25
26 # Hàm hiển thị giao diện đăng nhập
27 def show_login_view(self):
28     # Khởi tạo LoginView và truyền vào đối tượng root và controller
29     LoginView(self.root, self)
30
31 # Hàm hiển thị giao diện chính của ứng dụng sau khi đăng nhập thành công
32 def show_main_view(self):
33     # Đặt lại cửa sổ root để mở một cửa sổ chính mới và hiển thị MainView
34     self.root = Tk()
35     MainView(self.root, self)
36
37 # Hàm thêm một lịch hẹn mới vào cơ sở dữ liệu
38 def add_appointment(self, name, phone, appointment_date, service):
39     # Gọi phương thức add_appointment từ lớp Database
40     self.database.add_appointment(name, phone, appointment_date, service)
41
42 # Hàm lấy danh sách tất cả lịch hẹn
43 def get_appointments(self):
44     # Gọi phương thức get_appointments từ lớp Database và trả về kết quả
45     return self.database.get_appointments()
46
47 # Hàm xóa một lịch hẹn dựa trên ID
48 def delete_appointment(self, appointment_id):
49     # Gọi phương thức delete_appointment từ lớp Database với ID của lịch hẹn cần xóa
50     self.database.delete_appointment(appointment_id)
51
52 # Các phương thức bổ sung cho Controller để xử lý các hành động từ giao diện có thể được thêm vào đây
53
54 # Đoạn mã này kiểm tra nếu tập tin được chạy trực tiếp, Controller sẽ được khởi chạy
55 if __name__ == "__main__":
56     Controller()
```

- Views.py

```

1 # Import các thư viện cần thiết từ Tkinter và tkcalendar
2 from tkinter import *
3 from tkinter import messagebox, ttk # Thư viện hỗ trợ hiển thị thông báo và bảng
4 from tkcalendar import DateEntry # Thư viện để chọn ngày
5
6 # Lớp Tooltip dùng để hiển thị gợi ý khi di chuột qua các widget
7 class Tooltip:
8     def __init__(self, widget, text):
9         self.widget = widget
10        self.text = text
11        self.tip_window = None
12        # Gắn các sự kiện để hiển thị và ẩn tooltip
13        self.widget.bind("<Enter>", self.show_tip)
14        self.widget.bind("<Leave>", self.hide_tip)
15
16    # Hàm hiển thị tooltip khi di chuột qua widget
17    def show_tip(self, event=None):
18        if self.tip_window or not self.text:
19            return
20        # Tính toán vị trí của tooltip
21        x, y, _ = self.widget.bbox("insert")
22        x += self.widget.winfo_rootx() + 25
23        y += self.widget.winfo_rooty() + 25
24        # Tạo cửa sổ tooltip
25        self.tip_window = tw = Toplevel(self.widget)
26        tw.wm_overrideredirect(True) # Loại bỏ viền của cửa sổ tooltip
27        tw.wm_geometry(f"{x}+{y}")
28        # Thiết lập nội dung và hiển thị tooltip
29        label = Label(tw, text=self.text, justify=LEFT,
30                    background="#ffffe0", relief=SOLID, borderwidth=1,
31                    font=("Arial", 10, "normal"))
32        label.pack(ipadx=1)
33
34    # Hàm ẩn tooltip khi rời chuột khỏi widget
35    def hide_tip(self, event=None):
36        if self.tip_window:
37            self.tip_window.destroy()
38            self.tip_window = None
39
40 # Lớp LoginView cho giao diện đăng nhập
41 class LoginView:
42     def __init__(self, root, controller):
43         self.root = root
44         self.controller = controller
45         self.setup_ui() # Gọi hàm thiết lập giao diện đăng nhập
46
47    # Thiết lập giao diện đăng nhập
48    def setup_ui(self):
49        self.root.title("Login")
50        self.root.geometry("400x200")
51        self.root.config(bg="#e0f7fa") # Thiết lập màu nền cho cửa sổ
52
53        # Tạo nhãn và ô nhập cho tên đăng nhập và mật khẩu
54        Label(self.root, text="Username:", font=("Arial", 12)).pack(pady=10)
55        self.username_entry = Entry(self.root, font=("Arial", 12))
56        self.username_entry.pack()
57
58        Label(self.root, text="Password:", font=("Arial", 12)).pack(pady=10)
59        self.password_entry = Entry(self.root, font=("Arial", 12), show="*")
60        self.password_entry.pack()
61
62        # Nút đăng nhập với sự kiện đăng nhập
63        login_btn = Button(self.root, text="Login", font=("Arial", 12), command=self.login)
64        login_btn.pack(pady=20)
65
66    # Hàm xử lý sự kiện đăng nhập
67    def login(self):
68        username = self.username_entry.get()
69        password = self.password_entry.get()
70        # Gọi hàm xác thực từ controller
71        if self.controller.verify_user(username, password):
72            messagebox.showinfo("Success", "Đăng nhập thành công!")
73            self.root.destroy() # Đóng cửa sổ đăng nhập
74            self.controller.show_main_view() # Hiển thị giao diện chính
75        else:
76            messagebox.showerror("Error", "Đăng nhập thất bại!") # Thông báo lỗi
77
78 # Lớp MainView cho giao diện chính của ứng dụng
79 class MainView:
80     def __init__(self, root, controller):
81         self.root = root
82         self.controller = controller
83         self.setup_ui() # Thiết lập giao diện chính
84
85    # Hàm thiết lập giao diện chính của ứng dụng
86    def setup_ui(self):
87        self.root.title("Ứng dụng Đặt Lịch Hẹn")
88        self.root.geometry("900x800")
89        self.root.config(bg="#e0f7fa") # Đặt màu nền
90
91        # Tạo tiêu đề cho ứng dụng
92        header_label = Label(self.root, text="Bật Lịch Hẹn", font=("Arial", 24, "bold"), bg="#26c6da", fg="white", pady=15)
93        header_label.pack(fill=X)
94        Tooltip(header_label, "Đây là phần tiêu đề của ứng dụng đặt lịch hẹn")
95
96        # Tạo khung chính chứa các trường nhập liệu
97        main_frame = Frame(self.root, bg="ffffff", padx=20, pady=20)
98        main_frame.pack(pady=20, padx=20, fill=BOTH, expand=True)

```

```

1  # Các trường nhập liệu: tên, số điện thoại, ngày hẹn, và dịch vụ
2  Label(main_frame, text="Tên khách hàng:", font=("Arial", 14), bg="#ffffff").grid(row=0, column=0, sticky=W, pady=5)
3  self.name_entry = Entry(main_frame, width=35, font=("Arial", 14), relief="solid", bd=1)
4  self.name_entry.grid(row=0, column=1, pady=5)
5  Tooltip(self.name_entry, "Nhập tên khách hàng")
6
7  Label(main_frame, text="Số điện thoại:", font=("Arial", 14), bg="#ffffff").grid(row=1, column=0, sticky=W, pady=5)
8  self.phone_entry = Entry(main_frame, width=35, font=("Arial", 14), relief="solid", bd=1)
9  self.phone_entry.grid(row=1, column=1, pady=5)
10 Tooltip(self.phone_entry, "Nhập số điện thoại khách hàng")
11
12 Label(main_frame, text="Chọn ngày:", font=("Arial", 14), bg="#ffffff").grid(row=2, column=0, sticky=W, pady=5)
13 self.date_entry = DateEntry(main_frame, width=32, font=("Arial", 14), date_pattern="yyyy-mm-dd", relief="solid", bd=1)
14 self.date_entry.grid(row=2, column=1, pady=5)
15 Tooltip(self.date_entry, "Chọn ngày hẹn")
16
17 Label(main_frame, text="Dịch vụ:", font=("Arial", 14), bg="#ffffff").grid(row=3, column=0, sticky=W, pady=5)
18 self.service_entry = Entry(main_frame, width=35, font=("Arial", 14), relief="solid", bd=1)
19 self.service_entry.grid(row=3, column=1, pady=5)
20 Tooltip(self.service_entry, "Nhập dịch vụ mà khách hàng yêu cầu")
21
22 # Khung chứa các nút chức năng
23 button_frame = Frame(main_frame, bg="#ffffff")
24 button_frame.grid(row=4, column=0, columnspan=2, pady=20)
25
26 # Các nút chức năng chính
27 button_width = 15
28 add_btn = Button(button_frame, text="Thêm lịch hẹn", font=("Arial", 13), command=self.add_appointment, bg="#4CAF50", fg="white", width=button_width)
29 add_btn.pack(side=LEFT, padx=5)
30 Tooltip(add_btn, "Nhấn vào đây để thêm lịch hẹn")
31
32 edit_btn = Button(button_frame, text="Chỉnh sửa", font=("Arial", 13), command=self.edit_appointment, bg="#FFA726", fg="white", width=button_width)
33 edit_btn.pack(side=LEFT, padx=5)
34 Tooltip(edit_btn, "Nhấn vào đây để chỉnh sửa lịch hẹn đã chọn")
35
36 delete_btn = Button(button_frame, text="Xóa lịch hẹn", font=("Arial", 13), command=self.delete_appointment, bg="#F44336", fg="white", width=button_width)
37 delete_btn.pack(side=LEFT, padx=5)
38 Tooltip(delete_btn, "Nhấn vào đây để xóa lịch hẹn đã chọn")
39
40 clear_all_btn = Button(button_frame, text="Xóa tất cả", font=("Arial", 13), command=self.clear_all_appointments, bg="#D32F2F", fg="white", width=button_width)
41 clear_all_btn.pack(side=LEFT, padx=5)
42 Tooltip(clear_all_btn, "Nhấn vào đây để xóa tất cả lịch hẹn")
43
44 exit_btn = Button(button_frame, text="Thoát", font=("Arial", 13), command=self.exit_app, bg="#9E9E9E", fg="white", width=button_width)
45 exit_btn.pack(side=LEFT, padx=5)
46 Tooltip(exit_btn, "Thoát ứng dụng")
47
48 # Khung tìm kiếm lịch hẹn
49 search_frame = Frame(self.root, bg="#ffffff")
50 search_frame.pack(pady=10, padx=20, fill=BOTH, expand=True)
51 search_label = Label(search_frame, text="Tìm kiếm khách hàng:", font=("Arial", 14), bg="#ffffff")
52 search_label.pack(side=LEFT, padx=10)
53 self.search_entry = Entry(search_frame, font=("Arial", 14), width=35)
54 self.search_entry.pack(side=LEFT, padx=10)
55 search_btn = Button(search_frame, text="Tìm kiếm", command=self.search_appointment, font=("Arial", 14), bg="#00796B", fg="white")
56 search_btn.pack(side=LEFT, padx=10)
57 Tooltip(self.search_entry, "Nhập tên khách hàng cần tìm")
58
59 # Khung và bảng Treeview để hiển thị danh sách lịch hẹn
60 tree_frame = Frame(self.root, bg="#ffffff")
61 tree_frame.pack(pady=10, padx=20, fill=BOTH, expand=True)
62
63 tree_label = Label(tree_frame, text="Danh sách lịch hẹn:", font=("Arial", 14), bg="#ffffff")
64 tree_label.pack(anchor=W, padx=10, pady=5)
65
66 # Cấu hình bảng Treeview hiển thị danh sách lịch hẹn
67 self.tree = ttk.Treeview(tree_frame, columns=("ID", "Tên", "Số điện thoại", "Ngày giờ hẹn", "Dịch vụ"), show="headings")
68 self.tree.heading("ID", text="ID")
69 self.tree.heading("Tên", text="Tên")
70 self.tree.heading("Số điện thoại", text="Số điện thoại")
71 self.tree.heading("Ngày giờ hẹn", text="Ngày hẹn")
72 self.tree.heading("Dịch vụ", text="Dịch vụ")

```



```

1  # Thiết lập chiều rộng và vị trí các cột
2  self.tree.column("ID", width=50, anchor=CENTER)
3  self.tree.column("Tên", width=200)
4  self.tree.column("Số điện thoại", width=150)
5  self.tree.column("Ngày giờ hẹn", width=200)
6  self.tree.column("Dịch vụ", width=200)
7
8  # Hiển thị bảng danh sách lịch hẹn
9  self.tree.pack(padx=10, pady=5, fill=BOTH, expand=True)
10 self.show_appointments() # Hiển thị dữ liệu từ cơ sở dữ liệu
11
12 # Hàm thêm lịch hẹn mới
13 def add_appointment(self):
14     name = self.name_entry.get()
15     phone = self.phone_entry.get()
16     appointment_date = self.date_entry.get_date()
17     service = self.service_entry.get()
18     if name == "" or phone == "" or service == "":
19         messagebox.showwarning("Cảnh báo", "Vui lòng nhập đầy đủ thông tin!")
20     else:
21         try:
22             self.controller.add_appointment(name, phone, appointment_date, service)
23             messagebox.showinfo("Thành công", "Đã thêm lịch hẹn!")
24             self.clear_entries()
25             self.show_appointments()
26         except Exception as e:
27             messagebox.showerror("Lỗi", f"Đã xảy ra lỗi: {e}")
28
29 # Hàm chỉnh sửa lịch hẹn đã chọn
30 def edit_appointment(self):
31     try:
32         selected_item = self.tree.selection()[0]
33         appointment_id = self.tree.item(selected_item)["values"][0]
34
35         # Đưa thông tin lịch hẹn đã chọn vào các trường nhập liệu để chỉnh sửa
36         self.name_entry.delete(0, END)
37         self.name_entry.insert(0, self.tree.item(selected_item)["values"][1])
38         self.phone_entry.delete(0, END)
39         self.phone_entry.insert(0, self.tree.item(selected_item)["values"][2])
40         self.date_entry.set_date(self.tree.item(selected_item)["values"][3])
41         self.service_entry.delete(0, END)
42         self.service_entry.insert(0, self.tree.item(selected_item)["values"][4])
43
44         # Nút để lưu các thay đổi
45         def save_edits():
46             appointment_date = self.date_entry.get_date()
47             self.controller.database.update_appointment(appointment_id, self.name_entry.get(), self.phone_entry.get(), appointment_date, self.service_entry.get())
48             messagebox.showinfo("Thành công", "Đã cập nhật lịch hẹn!")
49             self.clear_entries()
50             self.show_appointments()
51             save_btn.pack_forget() # Ẩn nút sau khi lưu
52
53         save_btn = Button(self.root, text="Lưu thay đổi", font=("Arial", 14), command=save_edits, bg="#2196F3", fg="white")
54         save_btn.pack(pady=10)
55     except IndexError:
56         messagebox.showwarning("Cảnh báo", "Vui lòng chọn lịch hẹn để chỉnh sửa!")
57
58 # Hàm xóa lịch hẹn đã chọn
59 def delete_appointment(self):
60     try:
61         selected_item = self.tree.selection()[0]
62         appointment_id = self.tree.item(selected_item)["values"][0]
63         self.controller.delete_appointment(appointment_id)
64         messagebox.showinfo("Thành công", "Đã xóa lịch hẹn!")
65         self.show_appointments()
66     except IndexError:
67         messagebox.showwarning("Cảnh báo", "Vui lòng chọn lịch hẹn để xóa!")
68
69 # Hàm xóa tất cả lịch hẹn sau khi người dùng xác nhận
70 def clear_all_appointments(self):
71     confirm = messagebox.askyesno("Xác nhận", "Bạn có chắc chắn muốn xóa tất cả lịch hẹn?")
72     if confirm:
73         self.controller.database.clear_all_appointments()
74         self.show_appointments()
75         messagebox.showinfo("Thành công", "Đã xóa tất cả lịch hẹn!")
76
77 # Hàm thoát ứng dụng với xác nhận
78 def exit_app(self):
79     confirm = messagebox.askyesno("Xác nhận", "Bạn có chắc chắn muốn thoát ứng dụng?")
80     if confirm:
81         self.root.destroy()
82
83 # Hàm hiển thị danh sách lịch hẹn từ cơ sở dữ liệu lên bảng Treeview
84 def show_appointments(self):
85     for row in self.tree.get_children():
86         self.tree.delete(row)
87     rows = self.controller.get_appointments()
88     for row in rows:
89         self.tree.insert("", "end", values=(row[0], row[1], row[2], row[3].strftime("%Y-%m-%d"), row[4]))
90
91 # Hàm tìm kiếm lịch hẹn theo tên khách hàng
92 def search_appointment(self):
93     query = self.search_entry.get()
94     for row in self.tree.get_children():
95         self.tree.delete(row)
96     rows = self.controller.database.search_appointments(query)
97     for row in rows:
98         self.tree.insert("", "end", values=(row[0], row[1], row[2], row[3].strftime("%Y-%m-%d"), row[4]))
99
100 # Hàm xóa nội dung của các trường nhập liệu
101 def clear_entries(self):
102     self.name_entry.delete(0, END)
103     self.phone_entry.delete(0, END)
104     self.service_entry.delete(0, END)

```