

Food Allergy Database Management System Code **(RA2111029010009)**

```
import tkinter as tk
from tkinter import messagebox
from tkinter import ttk

class UserManagementSystem(tk.Tk):
    def __init__(self):
        super().__init__()

        self.title("User Management System")
        self.geometry("400x200")
        self.configure(background='light blue')

        self.create_login_widgets()

    def create_login_widgets(self):
        self.login_frame = tk.Frame(self, bg='light blue')
        self.login_frame.pack(pady=20)

        tk.Label(self.login_frame, text="Login", font=("Arial", 14), bg='light
blue').grid(row=0, column=0, columnspan=2)

        tk.Label(self.login_frame, text="Username:", bg='light blue').grid(row=1,
column=0, padx=10, pady=5)

        self.username_entry = tk.Entry(self.login_frame)
        self.username_entry.grid(row=1, column=1)
```

```
tk.Label(self.login_frame, text="Password:", bg='light blue').grid(row=2,
column=0, padx=10, pady=5)
```

```
self.password_entry = tk.Entry(self.login_frame, show="*")
```

```
self.password_entry.grid(row=2, column=1)
```

```
login_button = tk.Button(self.login_frame, text="Login",
command=self.login)
```

```
login_button.grid(row=3, column=0, columnspan=2, pady=10)
```

```
register_button = tk.Button(self.login_frame, text="Register",
command=self.register)
```

```
register_button.grid(row=4, column=0, columnspan=2, pady=5)
```

```
def create_registration_window(self):
```

```
    registration_window = tk.Toplevel(self)
```

```
    registration_window.title("Registration")
```

```
    registration_window.geometry("300x200")
```

```
    registration_window.configure(background='light blue')
```

```
tk.Label(registration_window, text="Register", font=("Arial", 14),
bg='light blue').grid(row=0, column=0, columnspan=2)
```

```
tk.Label(registration_window, text="Username:", bg='light
blue').grid(row=1, column=0, padx=10, pady=5)
```

```
self.reg_username_entry = tk.Entry(registration_window)
```

```
self.reg_username_entry.grid(row=1, column=1)
```

```

tk.Label(registration_window, text="Password:", bg='light
blue').grid(row=2, column=0, padx=10, pady=5)

self.reg_password_entry = tk.Entry(registration_window, show="*")
self.reg_password_entry.grid(row=2, column=1)


tk.Label(registration_window, text="Confirm Password:", bg='light
blue').grid(row=3, column=0, padx=10, pady=5)

self.confirm_password_entry = tk.Entry(registration_window, show="*")
self.confirm_password_entry.grid(row=3, column=1)


register_button = tk.Button(registration_window, text="Register",
command=self.create_user)

register_button.grid(row=4, column=0, columnspan=2, pady=10)


def create_allergy_details_widgets(self):

    self.allergy_details_frame = tk.Frame(self, bg='light blue')
    self.allergy_details_frame.pack(pady=20)


    tk.Label(self.allergy_details_frame, text="Allergen Details Entry",
font=("Arial", 14), bg='light blue').grid(row=0, column=0, columnspan=2)


    tk.Label(self.allergy_details_frame, text="Allergen Name:", bg='light
blue').grid(row=1, column=0, padx=10, pady=5)

    self.allergen_name_entry = tk.Entry(self.allergy_details_frame)
    self.allergen_name_entry.grid(row=1, column=1)


    tk.Label(self.allergy_details_frame, text="Symptoms:", bg='light
blue').grid(row=2, column=0, padx=10, pady=5)

    self.symptoms_entry = tk.Entry(self.allergy_details_frame)

```

```

self.symptoms_entry.grid(row=2, column=1)

submit_button = tk.Button(self.allergy_details_frame, text="Submit",
command=self.submit_allergy_details)

submit_button.grid(row=3, column=0, columnspan=2, pady=10)

def create_food_suggestions_window(self, keywords):
    food_suggestions_window = tk.Toplevel(self)
    food_suggestions_window.title("Food Suggestions")
    food_suggestions_window.geometry("600x400")
    food_suggestions_window.configure(background='light blue')

    # Create the table
    table_frame = tk.Frame(food_suggestions_window, bg='light blue')
    table_frame.pack(padx=20, pady=20)

    food_suggestions_table = ttk.Treeview(table_frame, columns=("Avoid",
"Substitute", "Additional Tips"), show="headings")

    food_suggestions_table.heading("Avoid", text="Avoid")
    food_suggestions_table.heading("Substitute", text="Substitute")
    food_suggestions_table.heading("Additional Tips", text="Additional Tips")
    food_suggestions_table.pack()

    # Insert data into the table based on keywords
    if "peanut" in keywords:
        food_suggestions_table.insert("", "end", values=("Peanuts, peanut
butter, peanut products", "Almond butter, sunflower seed butter, soy butter",
"Check for cross-contamination in processed foods and in facilities that handle
peanuts."))

```

```

    if "milk" in keywords:

        food_suggestions_table.insert("", "end", values=("Milk, cheese, yogurt",
"Almond milk, soy milk, coconut milk", "Check for hidden dairy in processed
foods and restaurant dishes."))

    if "egg" in keywords:

        food_suggestions_table.insert("", "end", values=("Eggs, mayonnaise,
baked goods", "Egg substitutes (e.g., tofu, banana, applesauce)", "Read labels
carefully and ask about ingredients when dining out."))

    if "shellfish" in keywords:

        food_suggestions_table.insert("", "end", values=("Shrimp, crab, lobster",
"Fish, poultry, tofu", "Avoid dishes that may contain hidden shellfish, such as
soups, sauces, and stir-fries."))

    if "wheat" in keywords:

        food_suggestions_table.insert("", "end", values=("Bread, pasta, cereal",
"Gluten-free bread, rice, quinoa", "Look for gluten-free alternatives and check
food labels for wheat-containing ingredients."))

    # Add more conditions for other allergen keywords


def register(self):

    self.create_registration_window()


def create_user(self):

    username = self.reg_username_entry.get()
    password = self.reg_password_entry.get()
    confirm_password = self.confirm_password_entry.get()

    if username and password and confirm_password:

        if password == confirm_password:

            # Here you would create the user in the database, for simplicity, just
            show a message

```

```

        messagebox.showinfo("Registration", "Registration successful!")

        # After successful registration, go back to the login page
        self.create_login_widgets()

    else:

        messagebox.showerror("Registration Error", "Passwords do not
match!")

    else:

        messagebox.showerror("Registration Error", "Please fill in all fields!")

def login(self):

    username = self.username_entry.get()
    password = self.password_entry.get()

    if username and password:

        # Show message dialog
        messagebox.showinfo("Login", "Login successful!")

        # Clear entry fields after login
        self.username_entry.delete(0, tk.END)
        self.password_entry.delete(0, tk.END)

        # Destroy login widgets and create allergen details entry widgets
        self.login_frame.destroy()
        self.create_allergy_details_widgets()

    else:

        messagebox.showerror("Login Error", "Please enter username and
password!")

```

```
def submit_allergy_details(self):
    allergen_name = self.allergen_name_entry.get()
    symptoms = self.symptoms_entry.get()
    print("Allergen Details Entry:")
    print("Allergen Name:", allergen_name)
    print("Symptoms:", symptoms)

    # Show message dialog
    messagebox.showinfo("Allergen Details Entry", "Allergen details
submitted successfully!")

    # Clear entry fields after submission
    self.allergen_name_entry.delete(0, tk.END)
    self.symptoms_entry.delete(0, tk.END)

    # Get keywords from allergen details and create food suggestions window
    keywords = allergen_name.lower().split() + symptoms.lower().split()
    self.create_food_suggestions_window(keywords)

if __name__ == "__main__":
    app = UserManagementSystem()
    app.mainloop()
```