

Name: **Mohammed Hamoud Ali**  
No: **202070363**

Dr. **Ahamed AL-Arashi**

To operate the Disease Diagnosis Tool code you provided, follow these steps:

### **Prerequisites**

#### **1. Python Installation:**

- Ensure you have Python installed on your computer. You can download it from [python.org](https://python.org).

#### **2. Tkinter:**

- Tkinter is included with most Python installations. You can check if it is available by running the following command in your Python environment:

```
python  
Copy
```

```
import tkinter
```

#### **3. Image Files:**

- Prepare the images corresponding to the diseases you want to diagnose. Name them as follows and place them in the specified directory:
  - measles.png
  - german\_measles.png
  - flu.png
  - common\_cold.png
  - mumps.png
  - chicken\_pox.png
  - whooping\_cough.png

- Ensure these files are located  
in `C:/Users/Qursan/Desktop/MohammedNoman_202070363_AI/`.

## Steps to Run the Code

### 1. Open a Text Editor or IDE:

- Use any text editor (like Notepad) or an Integrated Development Environment (IDE) like PyCharm, VSCode, or IDLE.

### 2. Copy the Code:

- Copy the entire code you provided into your text editor or IDE.

### 3. Save the File:

- Save the file with a `.py` extension, for example, `disease_diagnosis_tool.py`.

### 4. Run the Program:

- Open a terminal or command prompt.
- Navigate to the directory where your `.py` file is saved. For example:

```
bash
```

```
Copy
```

```
cd C:/path/to/your/script
```

- Run the script using Python:

```
bash
```

```
Copy
```

```
python disease_diagnosis_tool.py
```

### 5. Using the Application:

- Once the application window opens:
  - Enter the patient's name in the provided text box.
  - Click the "Start Diagnosis" button.
  - Answer the symptom-related questions that appear in pop-up dialogs.
  - After answering, the application will display the probable disease and show an image related to that disease (if the image file exists).

## Troubleshooting

- **Image Not Loading:** If you receive an error about the image not being found, double-check:

- The file names and paths.
- Ensure the images are in the correct format (.png) and located in the specified directory.
- **Python Errors:** If you encounter any errors while running the code, check the error messages for clues about what might be wrong. Common issues include:
  - Syntax errors in the code (ensure it was copied correctly).
  - Missing image files or incorrect paths.

## Summary

The code creates a graphical user interface (GUI) application that helps in diagnosing diseases based on patient symptoms. The user can input a patient's name and respond to symptom-related questions. Based on the responses, the application suggests a probable disease and displays a corresponding image.

## Code Breakdown

### 1. Imports:

- `tkinter`: The main library for creating the GUI.
- `messagebox`: Used to display pop-up messages (e.g., error messages).
- `filedialog`: Imported but not used in the current code. It could be used for file selection if needed.

### 2. Class Definition: `DiseaseDiagnosisApp`

- **`__init__` Method:** Initializes the GUI components.
  - Labels, entry fields, and buttons are created and placed in the window.
  - A placeholder for displaying images is set up.

### 3. Symptom Checking:

- **`symptom` Method:** Prompts the user with yes/no questions about specific symptoms for the patient. Returns the user's response.

### 4. Disease Hypotheses:

- **hypotheses Method:** Checks the symptoms against predefined conditions for various diseases (e.g., measles, flu). It returns True if the symptoms match, indicating a possible diagnosis.

#### 5. **Diagnosis Process:**

- **start\_diagnosis Method:**
  - Retrieves the patient's name from the input field.
  - Iterates through a list of diseases, checking if the patient's symptoms match any hypotheses.
  - Displays the result in a label and shows the corresponding image if a disease is diagnosed. If no diagnosis is found, it clears the image and shows an error message.

## 6. Image Display:

- **show\_photo Method:**
  - Constructs the path to the image file based on the diagnosed disease.
  - Attempts to load and display the image. If it fails (e.g., due to an incorrect path), it raises an error message.

## 7. Main Loop:

- The application is initiated in the if `__name__ == "__main__"`: block, which creates the main window and runs the Tkinter event loop.

## Explanation of Key Components

- **GUI Elements:** The app uses labels for text, an entry widget for user input, and buttons to trigger actions.
- **User Interaction:** The app interacts with users through message boxes that ask about symptoms, allowing for a dynamic diagnosis process.
- **Error Handling:** The `show_photo` method includes error handling to inform the user if an image cannot be loaded, which makes the application more robust.
- **Image Loading:** The application is designed to load images dynamically based on the diagnosed disease, enhancing user experience by providing visual context.

## Conclusion

The Disease Diagnosis Tool is a practical example of using Tkinter for creating user-friendly applications. It combines user input, conditional logic for diagnosis, and dynamic image display, making it an interactive tool for health-related inquiries.