```
import streamlit as st
import tensorflow as tf
from PIL import Image
import numpy as np

# Load the trained model
@st.cache(allow_output_mutation=True)
def load_model():
    model = tf.keras.models.load_model('potato_leaf_disease_model.keras')
    # Path to the saved model
    return model

model = load_model()

# Class names for prediction (assuming two classes: healthy and diseased)
class_names = ['Healthy', 'Diseased']

# Title of the Streamlit app
st.title("Potato Leaf Disease Detection")

# Upload an image file
uploaded_file = st.file_uploader("Choose an image of a potato leaf...", type=["jpg", "jpeg", "png"])

if uploaded_file is not None:
    # Convert the file to an image and display it
    image = Image.open(uploaded_file)
    st.image(image, caption='Uploaded potato leaf image.', use_column_width=True)
    st.write("")
    st.write("Classifying...")

    # Preprocess the uploaded image to fit the model's input shape
    image = image.resize((150, 150))  # Resize to the size your model expects
    image = np.array(image) / 255.0  # Normalize the image
    image = np.expand_dims(image, axis=0)  # Add batch dimension

    # Make prediction
    predictions = model.predict(image)
    score = tf.nn.softmax(predictions[0])

    # Output the result with the class name and accuracy
    st.write(f"This leaf is **{class_names[np.argmax(score)]}** with a confidence of {100 * np.max(score

    # Display the possible treatment if the leaf is diseased
    if class_names[np.argmax(score)] == 'Diseased':
        st.write("### Possible Treatment")
        st.write("""- Remove infected leaves from the plant.
- Use fungicides like mancozeb or chlorothalonil.
- Ensure good air circulation around plants.
- Avoid watering plants from above; instead, water the base.
""")

# Instructions
st.write("Upload an image of a potato leaf to classify if it's healthy or diseased.")
```