```
# -*- coding: utf-8 -*-
"""
Created on Tue Oct 31 16:32:59 2023

@author:
lenovo
"""

# -*- coding:

import pickle
import streamlit as st
from
streamlit_option_menu import option_menu
import sklearn


# loading the saved
models

diabetes_model = pickle.load(open('C:/Users/lenovo/Downloads/saved
models/diabetes_model.sav', 'rb'))

heart_disease_model =
pickle.load(open('C:/Users/lenovo/Downloads/saved
models/heart_disease_model.sav','rb'))

parkinsons_model =
pickle.load(open('C:/Users/lenovo/Downloads/saved models/parkinsons_model.sav', 'rb'))



#
sidebar for navigation
with st.sidebar:

    selected = option_menu('Multiple Disease
Prediction System',

                          ['Diabetes
Prediction',
                           'Heart Disease Prediction',

 'Parkinsons Prediction'],
                          icons=['activity','heart','person'],

                      default_index=0)


# Diabetes Prediction Page
if (selected ==
'Diabetes Prediction'):

    # page title
    st.title('Diabetes Prediction using ML')



    # getting the input data from the user
    col1, col2, col3 = st.columns(3)


    with col1:
        Pregnancies = st.text_input('Number of Pregnancies')


with col2:
        Glucose = st.text_input('Glucose Level')

    with col3:

BloodPressure = st.text_input('Blood Pressure value')

    with col1:
```

```
SkinThickness = st.text_input('Skin Thickness value')

    with col2:
        Insulin =
st.text_input('Insulin Level')

    with col3:
        BMI = st.text_input('BMI
value')

    with col1:
        DiabetesPedigreeFunction = st.text_input('Diabetes
Pedigree Function value')

    with col2:
        Age = st.text_input('Age of the
Person')


    # code for Prediction
    diab_diagnosis = ''

    # creating
a button for Prediction

    if st.button('Diabetes Test Result'):

diab_prediction = diabetes_model.predict([[Pregnancies, Glucose, BloodPressure, SkinThickness,
Insulin, BMI, DiabetesPedigreeFunction, Age]])

        if (diab_prediction[0] ==
1):
          diab_diagnosis = 'The person is diabetic'
        else:

diab_diagnosis = 'The person is not diabetic'


st.success(diab_diagnosis)




# Heart Disease Prediction Page
if (selected == 'Heart
Disease Prediction'):

    # page title
    st.title('Heart Disease Prediction using
ML')

    col1, col2, col3 = st.columns(3)

    with col1:
        age =
st.text_input('Age')

    with col2:
        sex = st.text_input('Sex')


  with col3:
        cp = st.text_input('Chest Pain types')

    with col1:

trestbps = st.text_input('Resting Blood Pressure')

    with col2:
        chol =
st.text_input('Serum Cholestoral in mg/dl')

    with col3:
        fbs =
st.text_input('Fasting Blood Sugar > 120 mg/dl')

    with col1:
        restecg
= st.text_input('Resting Electrocardiographic results')

    with col2:

thalach = st.text_input('Maximum Heart Rate achieved')

    with col3:
        exang
= st.text_input('Exercise Induced Angina')

    with col1:
        oldpeak =
st.text_input('ST depression induced by exercise')

    with col2:
        slope =
st.text_input('Slope of the peak exercise ST segment')

    with col3:
        ca =
st.text_input('Major vessels colored by flourosopy')

    with col1:
        thal =
st.text_input('thal: 0 = normal; 1 = fixed defect; 2 = reversable defect')
    age =
float(age) if age else 0.0
    sex = float(sex) if sex else 0.0
    cp = float(cp) if cp else
0.0
    trestbps = float(trestbps) if trestbps else 0.0
    chol = float(chol) if chol else
0.0
    fbs = float(fbs) if fbs else 0.0
    restecg = float(restecg) if restecg else 0.0

 thalach = float(thalach) if thalach else 0.0
    exang = float(exang) if exang else 0.0

oldpeak = float(oldpeak) if oldpeak else 0.0
    slope = float(slope) if slope else 0.0

ca = float(ca) if ca else 0.0
    thal = float(thal) if thal else 0.0




    # code for Prediction
    heart_diagnosis = ''

    # creating a button for
Prediction

    if st.button('Heart Disease Test Result'):
        heart_prediction =
heart_disease_model.predict([[age, sex, cp, trestbps, chol, fbs,
restecg,thalach,exang,oldpeak,slope,ca,thal]])

        if
(heart_prediction[0] == 1):
          heart_diagnosis = 'The person is having heart disease'

       else:
          heart_diagnosis = 'The person does not have any heart disease'


    st.success(heart_diagnosis)
```

```
# Parkinson's Prediction Page
if
(selected == "Parkinsons Prediction"):

    # page title

st.title("Parkinson's Disease Prediction using ML")

    col1, col2, col3,
col4, col5 = st.columns(5)

    with col1:
        fo = st.text_input('MDVP:Fo(Hz)')


    with col2:
        fhi = st.text_input('MDVP:Fhi(Hz)')

    with
col3:
        flo = st.text_input('MDVP:Flo(Hz)')

    with col4:

Jitter_percent = st.text_input('MDVP:Jitter(%)')

    with col5:
        Jitter_Abs
= st.text_input('MDVP:Jitter(Abs)')

    with col1:
        RAP =
st.text_input('MDVP:RAP')

    with col2:
        PPQ = st.text_input('MDVP:PPQ')


    with col3:
        DDP = st.text_input('Jitter:DDP')

    with col4:

     Shimmer = st.text_input('MDVP:Shimmer')

    with col5:
        Shimmer_dB =
st.text_input('MDVP:Shimmer(dB)')

    with col1:
        APQ3 =
st.text_input('Shimmer:APQ3')

    with col2:
        APQ5 =
st.text_input('Shimmer:APQ5')

    with col3:
        APQ =
st.text_input('MDVP:APQ')

    with col4:
        DDA =
st.text_input('Shimmer:DDA')

    with col5:
        NHR = st.text_input('NHR')


    with col1:
        HNR = st.text_input('HNR')

    with col2:
```

```
RPDE = st.text_input('RPDE')

    with col3:
        DFA = st.text_input('DFA')


    with col4:
        spread1 = st.text_input('spread1')

    with col5:

    spread2 = st.text_input('spread2')

    with col1:
        D2 =
st.text_input('D2')

    with col2:
        PPE = st.text_input('PPE')




    # code for Prediction
    parkinsons_diagnosis = ''

    # creating a
button for Prediction
    if st.button("Parkinson's Test Result"):

parkinsons_prediction = parkinsons_model.predict([[fo, fhi, flo, Jitter_percent, Jitter_Abs,
RAP, PPQ,DDP,Shimmer,Shimmer_dB,APQ3,APQ5,APQ,DDA,NHR,HNR,RPDE,DFA,spread1,spread2,D2,PPE]])


        if (parkinsons_prediction[0] == 1):

parkinsons_diagnosis = "The person has Parkinson's disease"
        else:

parkinsons_diagnosis = "The person does not have Parkinson's disease"


st.success(parkinsons_diagnosis)
```