**Multiple Disease Detection System**

A PROJECT REPORT

Submitted by

**Aarti Deokar**

In partial fulfilment for the award of the degree of

**M.Sc. Data Science and Artificial Intelligence – Part I**



**Department M.Sc. Data Science and Artificial Intelligence Ramniranjan Jhunjhunwala College of Arts, Science & Commerce**

Mumbai

(2022-2023)

**RAMNIRANJAN JHUNJHUNWALA COLLEGE OF ART’S, SCIENCE & COMMERCE (AUTONOMOUS), GHATKOPAR(W), MUMBAI-400 086**



# Certificate

This is to certify that the Project entitled **Multiple Disease Detection System** is bonafide work of **Aarti Deokar** bearing Seat No **730** submitted in partial fulfilment of the requirements for the award of Degree **Master of Science** in **Data Science & Artificial Intelligence.**

**Signature of Internal Guide Signature of Co-ordinator**

**College Seal and Date Signature Examiner**

**Index**

|  |  |  |
| --- | --- | --- |
| 1. | Abstract | 4 |
| 2. | Acknowledgement | 5 |
| 3. | Declaration | 6 |
| 4. | Introduction | 7 |
| 5. | Description | 9 |
| 6. | Problem System | 10 |
| 7. | Proposed System | 11 |
| 8. | Datasets | 11 |
| 9. | Deployment | 11 |
| 10. | Code | 12 |
| 11. | Output Images | 18 |
| 12. | References | 22 |

**Abstract**

With broad data development in biomedical and healthcare sectors, detailed analyzes of medical data support early detection of illness, patient care and community services. However, the quality of the study is lowered when the content of the medical data is incomplete. Also, various regions exhibit unique features of certain regional diseases. In this project, we streamline machine learning algorithms to effectively predict chronic disease outbreaks in populations with recurrent diseases. The diagnosis of diseases is a critical and central aspect of medicinal science. Doctors breakdown side effects in the human body more often than not to foresee diseases. In recent times, numerous research strategies have been used with a specific goal to make it more accurate. This system will help to predict the medical results efficiently. In this system, we will provide a user-friendly interface that can be used by the users to detect whether their medical test results are positive or normal.

**Acknowledgement**

Before we get into thick of things, I would like to add a few heartfelt words for the people who were part of **Multiple Disease Detection System** project in numerous ways, people who gave unending support right from the stage the project idea was conceived.

A project report is such a comprehensive coverage; it would not have been materialized without the help of many. The four things that go on to make a successful endeavour are dedication, hard work, patience and correct guidance. Able and timely guidance not only helps in making an effort fruitful but also transforms the whole process of learning and implementing into an enjoyable experience.

In particular, I would like to thank our **Mentor/Director Dr. (Mrs.) Usha Mukundan, R.J. College**. I would like to give a very special honor and respect to our teacher, **Prof. Mujtaba Shaikh** who took keen interest in checking the minute details of the project work and guided us throughout the same. A sincere quote of thanks to the non-teaching staff for providing us software their time. I appreciate outstanding co-operation by them, especially for the long Lab timings that we could receive.

**Declaration**

I hereby declare that the Project entitled, **“Multiple Disease Detection System”** done at R. J. COLLEGE, Ghatkopar(W), Mumbai, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, No one has submitted to any other University. The Project is done in partial fulfilment of the requirements for the award of degree of ***Master of Science Data Science and Artificial Intelligence*** to be submitted as mini project as part of our curriculum.

**Introduction**

In this digital world, data is an asset, and enormous data was generated in all the fields. Data in the healthcare industry consists of all the information related to patients. Here a general architecture has been proposed for predicting the disease in the healthcare industry. Many of the existing models are concentrating on one disease per analysis. Like one analysis for diabetes analysis, one for cancer analysis, one for heart diseases like that. There is no common system present that can analyze more than one disease at a time. Thus, we are concentrating on providing immediate and accurate disease predictions to the users about the symptoms they enter along with the disease predicted. So, we are proposing a system which used to predict multiple diseases by using streamlit. In this system, we are going to analyze Diabetes, Heart, and malaria disease analysis. Later many more diseases can be included In multiple disease prediction, it is possible to predict more than one disease at a time. So, the user doesn’t need to traverse different sites in order to predict the diseases. We are taking three diseases that are Liver, Diabetes, and Heart. As all the three diseases are correlated to each other. To implement multiple disease analyses we are going to use machine learning algorithms and Streamlit. When the user is accessing this API, the user has to send the parameters of the disease along with the disease name. Our Model will invoke the corresponding model and return the status of the patient. Our basic idea is to develop a system which will predict and give the details of the disease predicted along with its severity which as symptoms are given as input by the user. The system will compare the symptoms with the datasets provided in the database. If the symptom matches the datasets, then it should ask other relevant symptoms specifying the name of the symptom. If not, the symptom entered should be notified as the wrong symptom. After this a prompt will come up asking whether you want to still save the symptom in the database. If you click on yes, it will be saved in the database, if not it will go to the recycle bin. The main feature will be the machine learning, in which we will be using algorithms such as Naïve Bayes Algorithm, KNearest Algorithm, Decision Tree Algorithm, Random Forest Algorithm and Support Vector Machine, which will predict accurate disease and also, will find which algorithm gives a faster and efficient result by comparativelycomparing. The importance of this system analysis is that while analyzing the diseases all the parameters which cause the disease are included so it is possible to detect the disease efficiently and more accurately. The final model's behavior will be saved as a python pickle file.

**Description**

A lot of analysis over existing systems in the healthcare industry considered only one disease at a time. For example, one system is used to analyze diabetes, another is used to analyze diabetes retinopathy, and another system is used to predict heart disease. Maximum systems focus on a particular disease. When an organization wants to analyze their patient’s, health reports then they have to deploy many models. The approach in the existing system is useful to analyze only particular diseases. In multiple disease prediction systems, a user can analyze more than one disease on a single website. The user doesn’t need to traverse different places in order to predict whether he/she has a particular disease or not Main objective behind developing a system helps the doctors to cross verify their diagnosed results which gives promising solutions over existing death rates. By using our proposed work try to invent a unique platform and most promising solution for early diagnosis of multiple diseases. Existing work analysis accuracy is reduced when the quality of medical data is incomplete. Moreover, different regions exhibit unique characteristics of certain regional diseases, which may weaken the prediction of disease wrong. So, we are giving more accurate solutions by using machine learning and Convolutional neural networks to detect diseases and make predictions.

**Problem System**

Many of the existing machine learning models for health care analysis are concentrating on one disease per analysis. For example, first is for liver analysis, one for cancer analysis, one for lung diseases like that. If a user wants to predict more than one disease, he/she has to go through different sites. There is no common system where one analysis can perform more than one disease prediction. Some of the models have lower accuracy which can seriously affect patients’ health. When an organization wants to analyze their patient’s health reports, they have to deploy many models which in turn increases the cost as well as time. Some of the existing systems consider very few parameters which can yield false results.

**Proposed System**

In multiple disease prediction, it is possible to predict more than one disease at a time. So, the user doesn’t need to traverse different sites in order to predict the diseases. We are taking three diseases that are Liver, Diabetes, and Heart. As all the three diseases are correlated to each other. To implement multiple disease

**Dataset**

For this multiple disease detection system I have used six different datasets for Kaggle.

**Deployment**

This model is deployed using third party resource ‘Streamlit’ which helps the developers to deploy their applications or projects in a simple way.

**Code**

import joblib

import pickle

import streamlit as st

from streamlit\_option\_menu import option\_menu

# load models

cancer\_model=pickle.load(open('cancer\_model.pkl','rb'))

diabetes\_model=joblib.load(open('diabetes\_model.pkl','rb'))

heart\_model=joblib.load(open('heart\_model.pkl','rb'))

kidney\_model=joblib.load(open('kidney\_model.pkl','rb'))

liver\_model=joblib.load(open('liver\_model.pkl','rb'))

parkinsons\_model=joblib.load(open('parkinsons\_model.pkl','rb'))

# sidebar

with st.sidebar:

    selected=option\_menu('Multiple Disease Detection System',['Welcome',

        'Cancer','Diabetes','Heart','Kidney','Liver','Parkinsons'],

        icons=['book','bookmark-check','bi-app-indicator','heart','file-bar-graph','activity','bi-person'],

        default\_index=0)

if selected == 'Welcome':

    st.title('Welcome to Automatic Disease Detection Using Machine Learning')

    st.image('disease-diagnosis-using-machine-learning.png')

if selected=='Cancer':

    st.title('Breast Cancer Test')

    concave=st.number\_input('Concave',value=0.012,step=0.025)

    area=st.number\_input('Area',value=500.0,step=0.1)

    radius=st.number\_input('Radius',value=10.0,step=0.025)

    perimeter=st.number\_input('Perimeter',value=60.0,step=0.1)

    concavity=st.number\_input('Concavity',value=0.020470, step=0.025)

    result=''

    if st.button('Get result'):

        cancer\_result=cancer\_model.predict([[concave,area,radius,perimeter,concavity]])

        if cancer\_result[0]==1:

            result='You are likely to have cancer. Please see a doctor.'

        else:

            result='You do not have cancer.'

    st.success(result)

if selected=='Diabetes':

    st.title('Diabetes Test')

    pregnancies=st.number\_input('Number of pregnencies',min\_value=0,max\_value=10,step=1,value=1)

    glucose=st.number\_input('Glucose Level',min\_value=50,step=1,value=100)

    bp=st.number\_input('Current blood Pressure',min\_value=50,max\_value=130,step=1,value=70)

    bmi=st.number\_input('BMI',min\_value=1,max\_value=70,value=30,step=1)

    pedigree=st.number\_input('Diabetes Pedigree Function',min\_value=0.2,max\_value=2.5,step=0.1,value=0.5)

    age=st.number\_input('Age',min\_value=0,max\_value=100,step=1,value=18)

    result = ''

    if st.button('Get result'):

        diabetes\_result = diabetes\_model.predict([[pregnancies,glucose,bp,bmi,pedigree,age]])

        if diabetes\_result[0] == 1:

            result = 'You are likely to have diabetes. Please see a doctor.'

        else:

            result = 'You do not have diabetes.'

    st.success(result)

if selected=='Heart':

    st.title('Heart Disease Test')

    chest\_pain=st.selectbox('Chest Pain type',['Typical Angina','Atypical Angina', 'Non-Anginal Pain','Asymptomatic'])

    cp=int()

    if chest\_pain=='Typical Angina':

        cp=0

    elif chest\_pain=='Atypical Angina':

        cp=1

    elif chest\_pain=='Non-Anginal Pain':

        cp=2

    else:

        cp=3

    rest\_bp=st.number\_input('Resting Blood Pressure (in mm of Hg)',value=120,step=1)

    cholestrol=st.number\_input('Serum Cholestrol (in mg/dl)',value=200, step=5)

    blood\_sugar=st.radio('Is Fasting Blood Sugar <120 mg/dl',['Yes','No'])

    ecg=st.selectbox('Electrocardiograph Result',['Normal','Having ST-T wave Abnormality','Showing propbable or definate left Ventricular Hypertrophy'])

    max\_heart\_rate=st.number\_input('Maximum Heart Rate Achieved',value=150,step=1)

    exercise=st.selectbox('Exercise Induced Angina',['Yes','No'])

    result = ''

    if st.button('Get result'):

        heart\_result = heart\_model.predict([[cp,rest\_bp,cholestrol,1 if blood\_sugar=='Yes' else 0,0 if ecg=='Normal' else 1,max\_heart\_rate,1 if exercise=='Yes' else 0]])

        if heart\_result[0] == 1:

            result = 'You are likely to have a heart disease. Please see a doctor.'

        else:

            result = 'Your heart is healthy.'

    st.success(result)

if selected=='Kidney':

    st.title('Kidney Test')

    bp=st.number\_input('Blood Pressure',value=50,step=1)

    gravity=st.number\_input('Specific Gravity',value=1.000,step=0.025)

    albumin=st.number\_input('Albumin',value=1.0,step=0.5)

    sugar=st.number\_input('Blood Sugar Level',value=1,step=1,max\_value=5)

    rbc=st.radio('Red Blood Cells Count',['abnormal','normal'])

    pbc=st.radio('Pus Cell Count',['abnormal','normal'])

    pcclumps=st.radio('Pus Cell Clumps',['present','not present'])

    result = ''

    if st.button('Get result'):

        kidney\_result = kidney\_model.predict([[bp,gravity,albumin,sugar,1 if rbc=='abnormal' else 0,1 if pbc=='abnormal' else 1,1 if pcclumps=='present' else 0]])

        if kidney\_result[0] == 1:

            result = 'You are likely to have a kidney disease. Please see a doctor.'

        else:

            result = 'Your kidney is healthy.'

    st.success(result)

    st.header('')

if selected=='Liver':

    st.title('Liver Test')

    total\_bilirubin=st.number\_input('Total Bilirubin',value=0.4,step=0.1)

    direct\_bilirubin=st.number\_input('Direct Bilirubin',value=0.1,step=0.1)

    alkeline=st.number\_input('Alkaline Phosphotase',value=70,step=1)

    alamine=st.number\_input('Alamine Aminotransferase',value=10,step=1)

    protein=st.number\_input('Total Protein',value=3.0,step=0.1)

    albumin=st.number\_input('Albumin',value=2.7,step=0.1)

    agratio=st.number\_input('Albumin to Globulin ratio',value=3.9,step=0.1)

    result = ''

    if st.button('Get result'):

        liver\_result = liver\_model.predict([[total\_bilirubin, direct\_bilirubin, alkeline, alamine, protein, albumin, agratio]])

        if liver\_result[0] == 1:

            result = 'You are likely to have a liver disease. Please see a doctor.'

        else:

            result = 'Your liver is healthy.'

    st.success(result)

    st.header('symptoms of liver disease')

    st.markdown('Skin and eyes that appear yellowish (jaundice)')

    st.markdown('Abdominal pain and swelling.')

    st.markdown('Swelling in the legs and ankles.')

    st.markdown('Itchy skin.')

if selected=='Parkinsons':

    st.title("Parkinson's Disease Test")

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

    with col1:

        fo = st.number\_input('MDVP:Fo(Hz)',value=150.0,step=0.25)

    with col2:

        fhi = st.number\_input('MDVP:Fhi(Hz)',value=160.0,step=0.5)

    with col3:

        flo = st.number\_input('MDVP:Flo(Hz)',value=100.0,step=1.0)

    with col4:

        Jitter\_percent = st.number\_input('MDVP:Jitter(%)',value=0.005582,step=0.0005)

    with col5:

        Jitter\_Abs = st.number\_input('MDVP:Jitter(Abs)',value=0.000038,step=0.00001)

    with col1:

        RAP = st.number\_input('MDVP:RAP',value=0.002868,step=0.001)

    with col2:

        PPQ = st.number\_input('MDVP:PPQ',value=0.00315, step=0.001)

    with col3:

        DDP = st.number\_input('Jitter:DDP',value=0.008602,step=0.001)

    with col4:

        Shimmer = st.number\_input('MDVP:Shimmer',value=0.027968,step=0.0025)

    with col5:

        Shimmer\_dB = st.number\_input('MDVP:Shimmer(dB)',value=0.259400,step=0.05)

    with col1:

        APQ3 = st.number\_input('Shimmer:APQ3',value=0.01388,step=0.01)

    with col2:

        APQ5 = st.number\_input('Shimmer:APQ5',value=0.016510,step=0.01)

    with col3:

        APQ = st.number\_input('MDVP:APQ',value=0.023982,step=0.01)

    with col4:

        DDA = st.number\_input('Shimmer:DDA',value=0.041660,step=0.01)

    with col5:

        NHR = st.number\_input('NHR',value=0.011348,step=0.01)

    with col1:

        HNR = st.number\_input('HNR',value=22.14200,step=1.0)

    with col2:

        RPDE = st.number\_input('RPDE',value=0.516489,step=0.1)

    with col3:

        DFA = st.number\_input('DFA',value=0.743321,step=0.1)

    with col4:

        spread1 = st.number\_input('spread1',value=-5.706983,step=0.5)

    with col5:

        spread2 = st.number\_input('spread2',value=0.188637,step=0.1)

    with col1:

        D2 = st.number\_input('D2',value=2.170480,step=0.25)

    with col2:

        PPE = st.number\_input('PPE',value=0.210302,step=0.025)

    # code for Prediction

    result = ''

    # creating a button for Prediction

    if st.button("Get 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):

            result = "You are likely to have Parkinson's disease. Please see a doctor"

        else:

            result = "You do not have Parkinson's disease"

    st.success(result)

    st.header('What causes Parkinson’s disease?')

    st.markdown('The most prominent signs and symptoms of Parkinson’s disease occur when nerve cells in the basal ganglia, an area of the brain that controls movement, become impaired and/or die. Normally, these nerve cells, or neurons, produce an important brain chemical known as dopamine. When the neurons die or become impaired, they produce less dopamine, which causes the movement problems associated with the disease. Scientists still do not know what causes the neurons to die.')

    st.header('Symptoms of Parkinson’s disease')

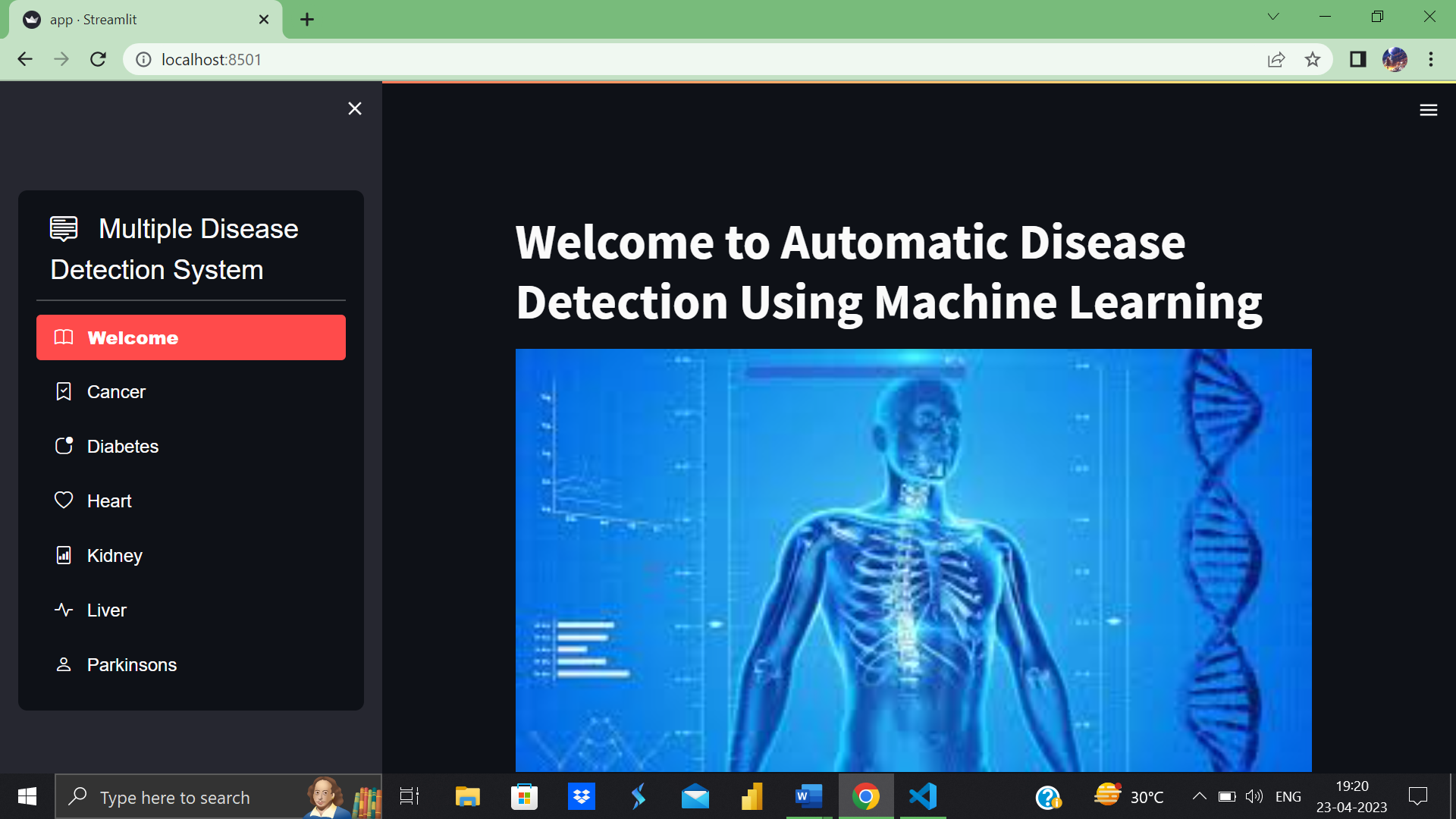
    st.markdown('Tremor in hands, arms, legs, jaw, or head')

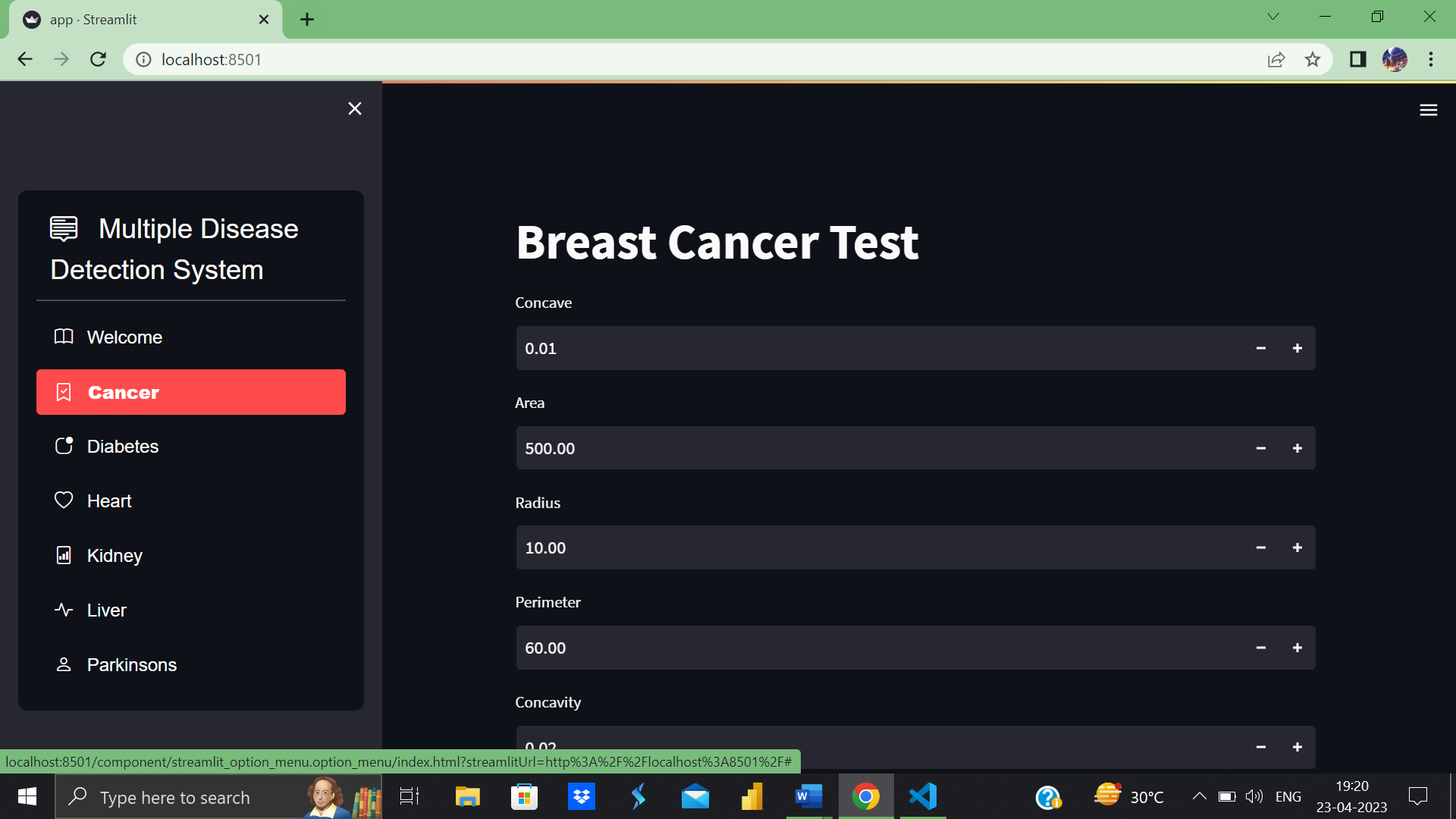
    st.markdown('Muscle stiffness, where muscle remains contracted for a long time')

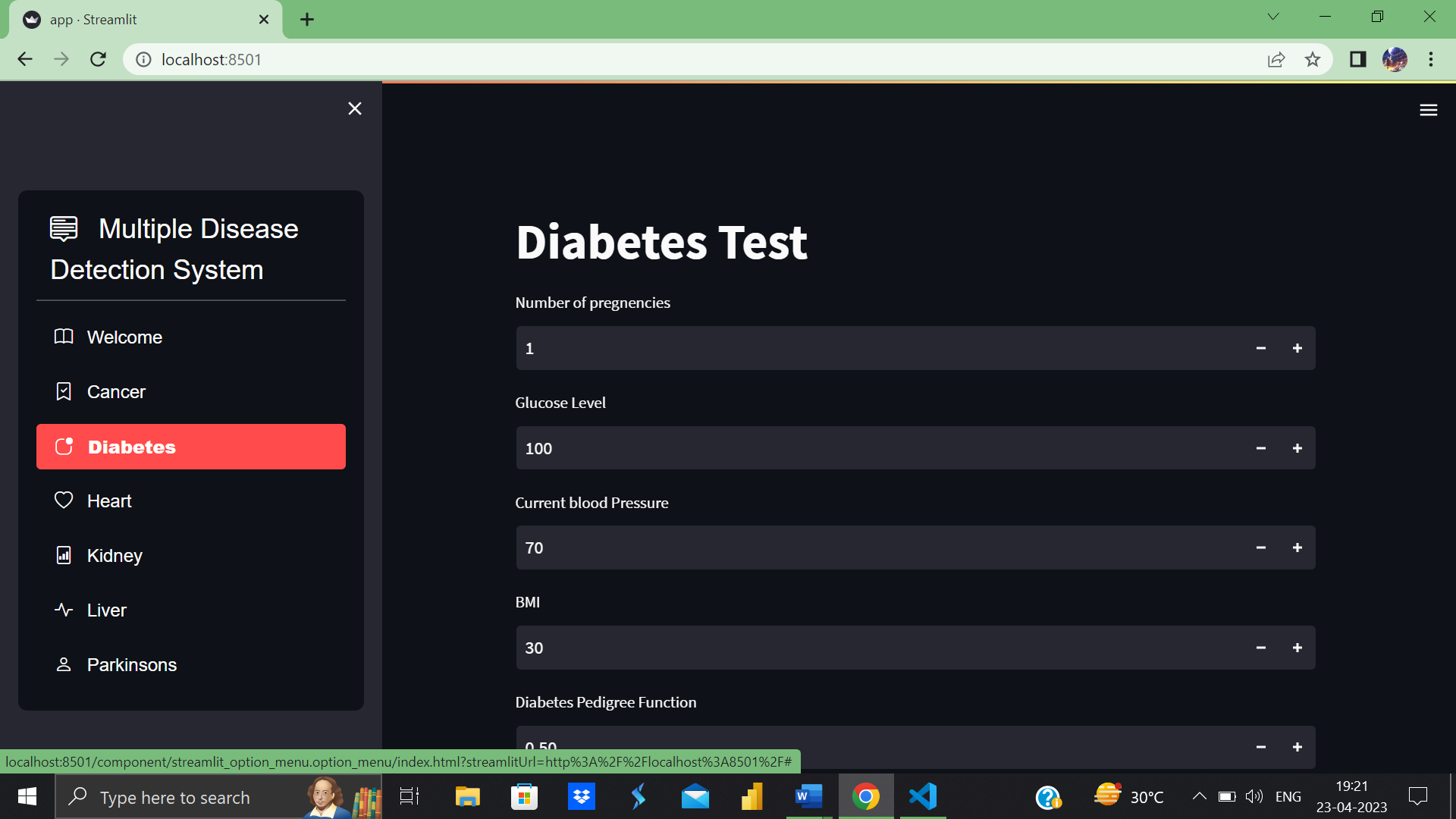
    st.markdown('Slowness of movement')

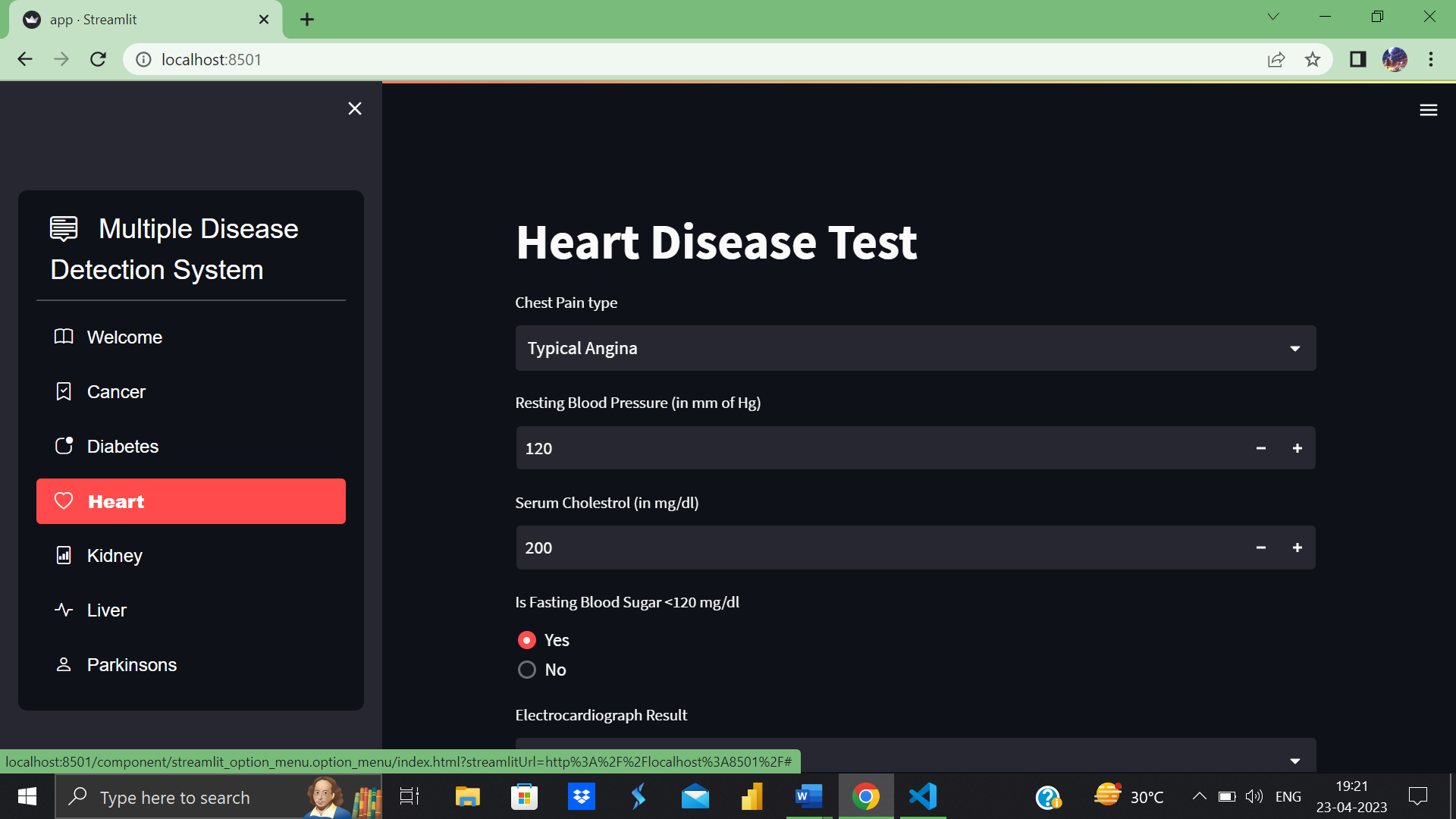
    st.markdown('Impaired balance and coordination, sometimes leading to falls')

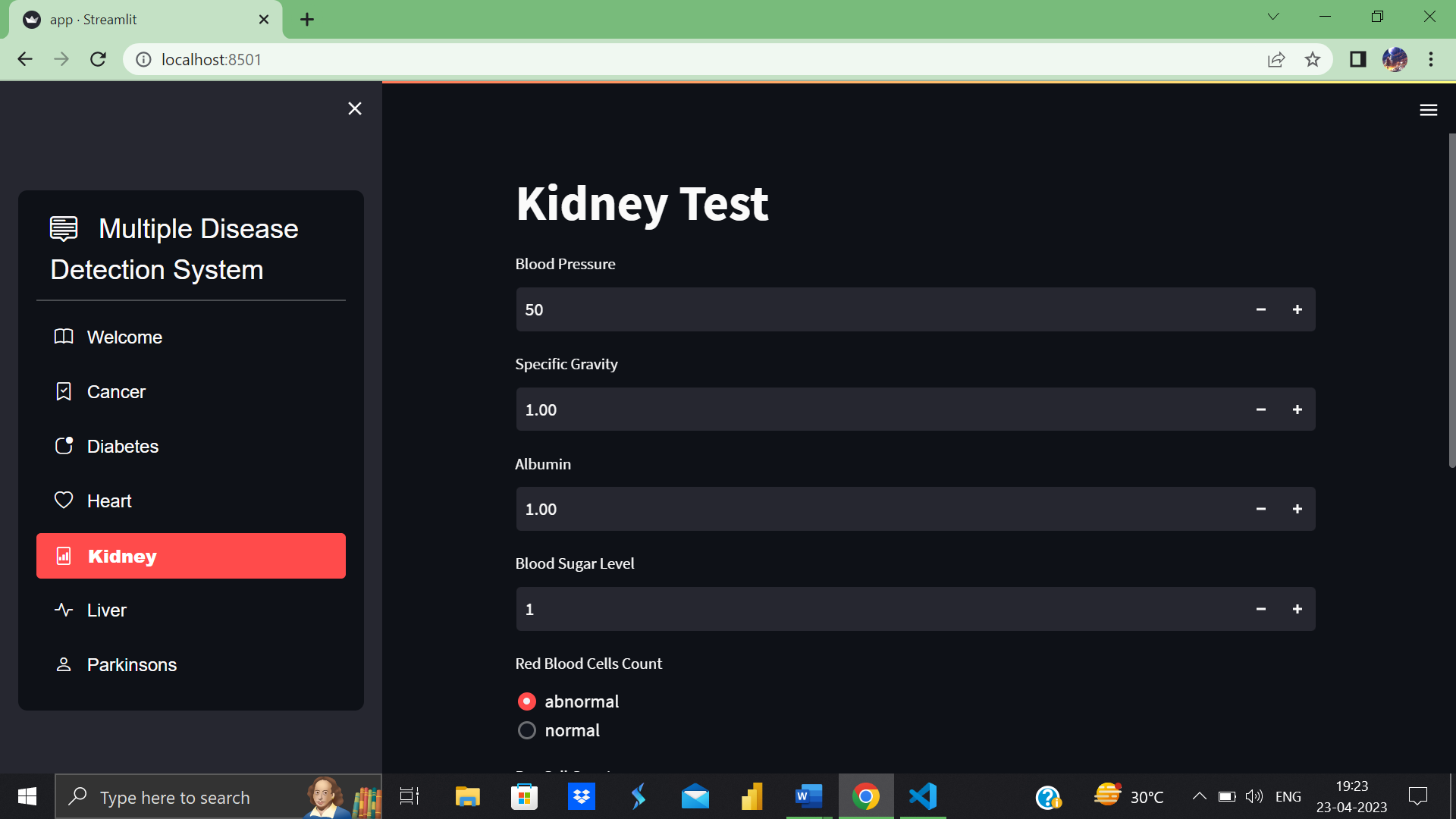
**Output Images**

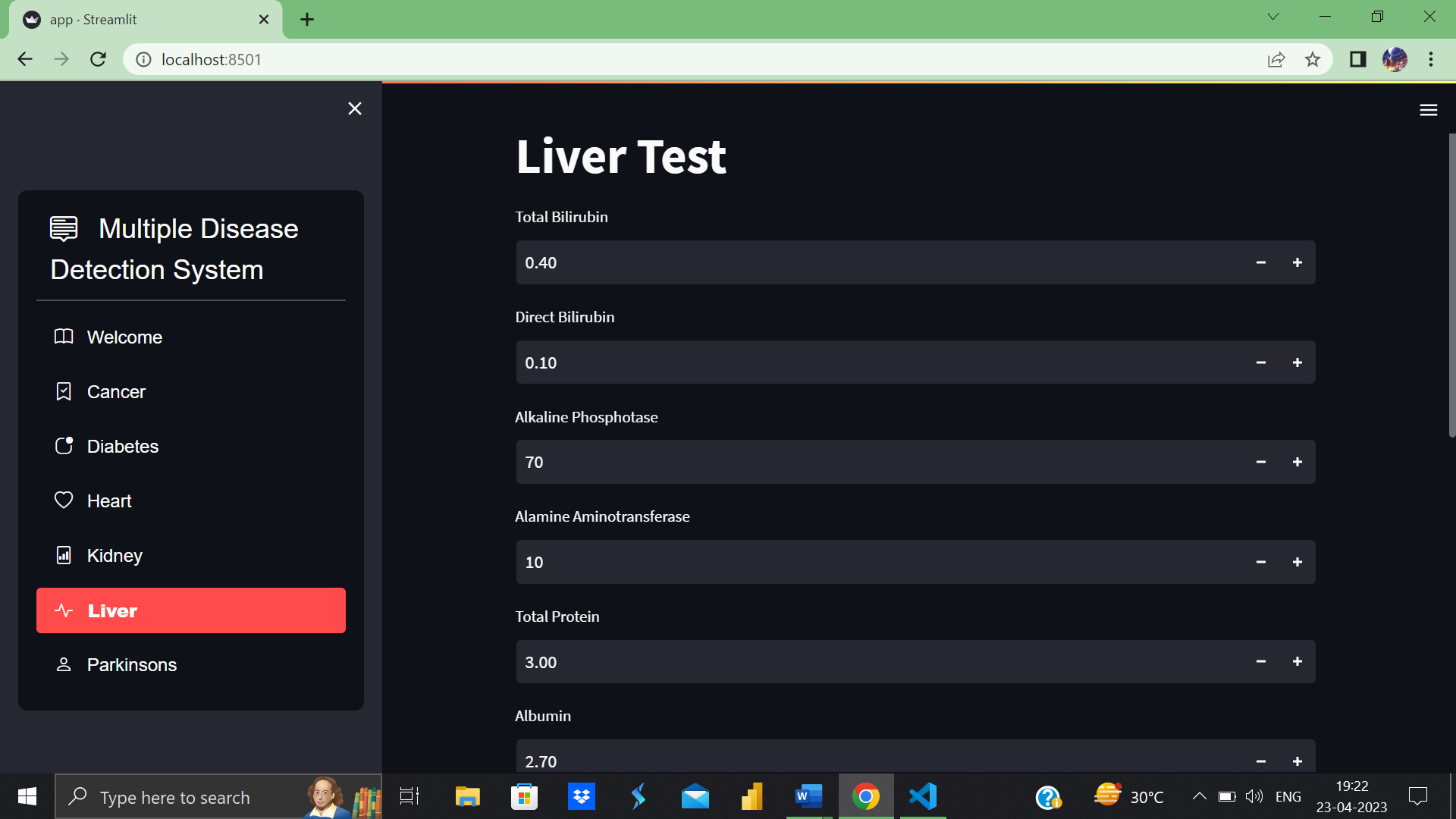


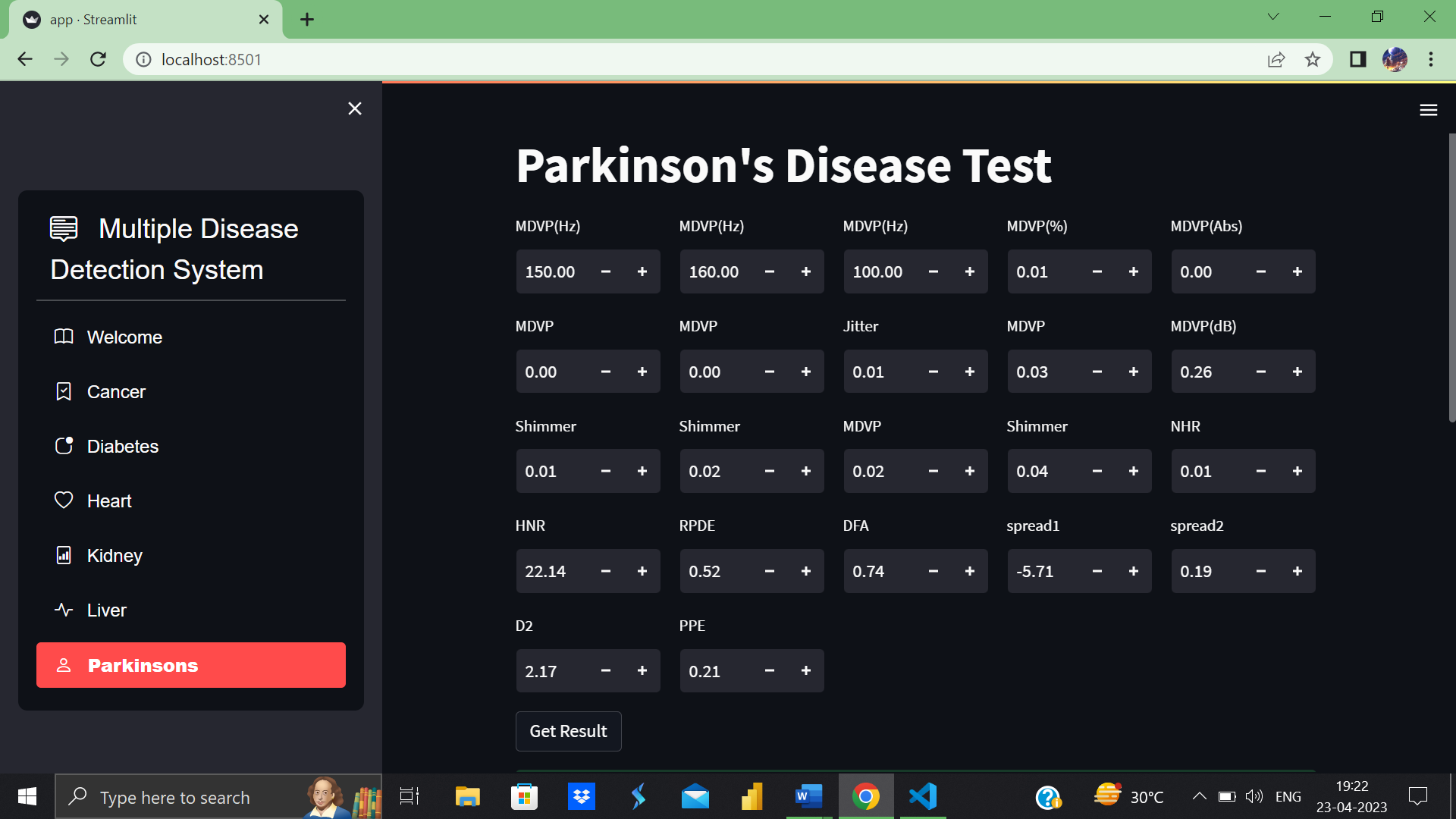












**REFERENCES**

[1] Priyanka Sonar, Prof. K. Jaya Malini,” DIABETES PREDICTION USING DIFFERENT MACHINE LEARNING APPROACHES”, 2019 IEEE ,3rd International Conference on Computing Methodologies and Communication (ICCMC)

[2] Archana Singh, Rakesh Kumar, “Heart Disease Prediction Using Machine Learning Algorithms”, 2020 IEEE, International Conference on Electrical and Electronics Engineering (ICE3)

[3] A. Sivasangari, Baddigam Jaya Krishna Reddy, Annamareddy Kiran, P. Ajitha,” Diagnosis of Liver Disease using Machine Learning Models” 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)