INTRODUCTION

1.1 OVERVIEW

The Thyroid gland is a vascular gland and one of the most important organs of the human body. This gland secretes two hormones which help in controlling the metabolism of the body. The two types of Thyroid disorders are Hyperthyroidism and Hypothyroidism. When this disorder occurs in the body, they release certain types of hormones into the body which imbalances the body's metabolism. A thyroid-related Blood test is used to detect this disease but it is often blurred and noise will be present. Data cleansing methods were used to make the data primitive enough for the analytics to show the risk of patients getting this disease. Machine Learning plays a very deciding role in disease prediction. Machine Learning algorithms, SVM - support vector machine, Random Forest Classifier, XGB Classifier and ANN - Artificial Neural Networks are used to predict the patient's risk of getting thyroid disease. The web app is created to get data from users to predict the type of disease.

1.2 PURPOSE

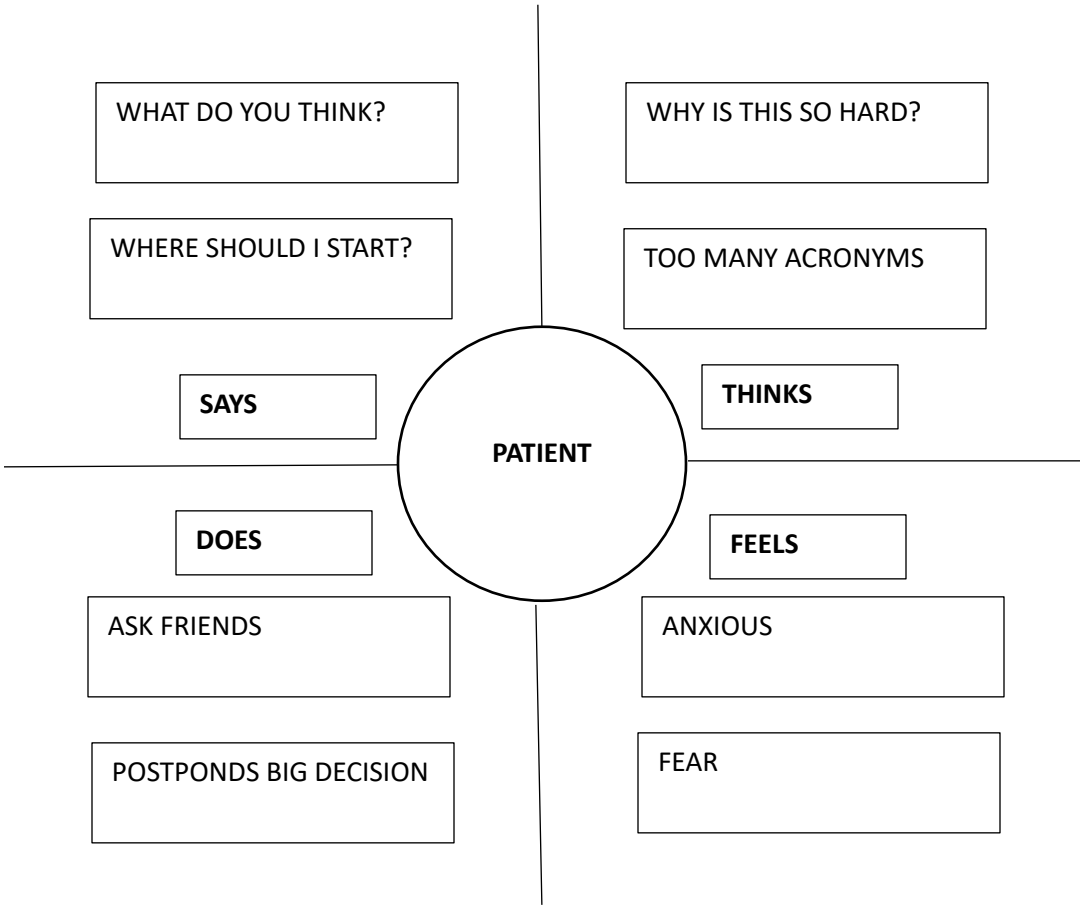
- The user interacts with the UI to enter the input.
- Entered input is analysed by the model which is integrated.
- Once the model analyses the input the prediction is showcased on the UI.

Disease Prediction using Machine Learning is the system that is used to predict the diseases from the symptoms which are given by the patients or any user. The system processes the symptoms provided by the user as input and gives the output as the probability of the disease. The aim of the study was to predict chronic kidney disease using machine-learning techniques. The main objective is to predict the diseases from the given symptoms create and monitors a health profile of every individual patient.

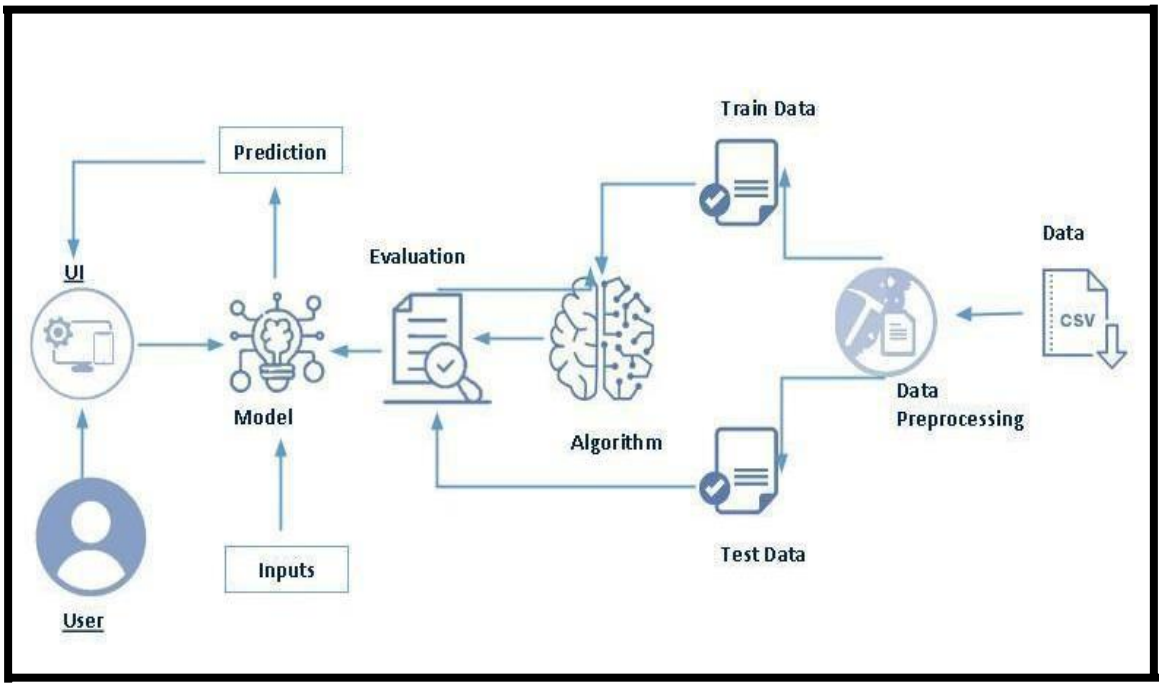
PROBLEM DEFINITION AND DESIGN THINKING

2.1 EMPATHY MAP

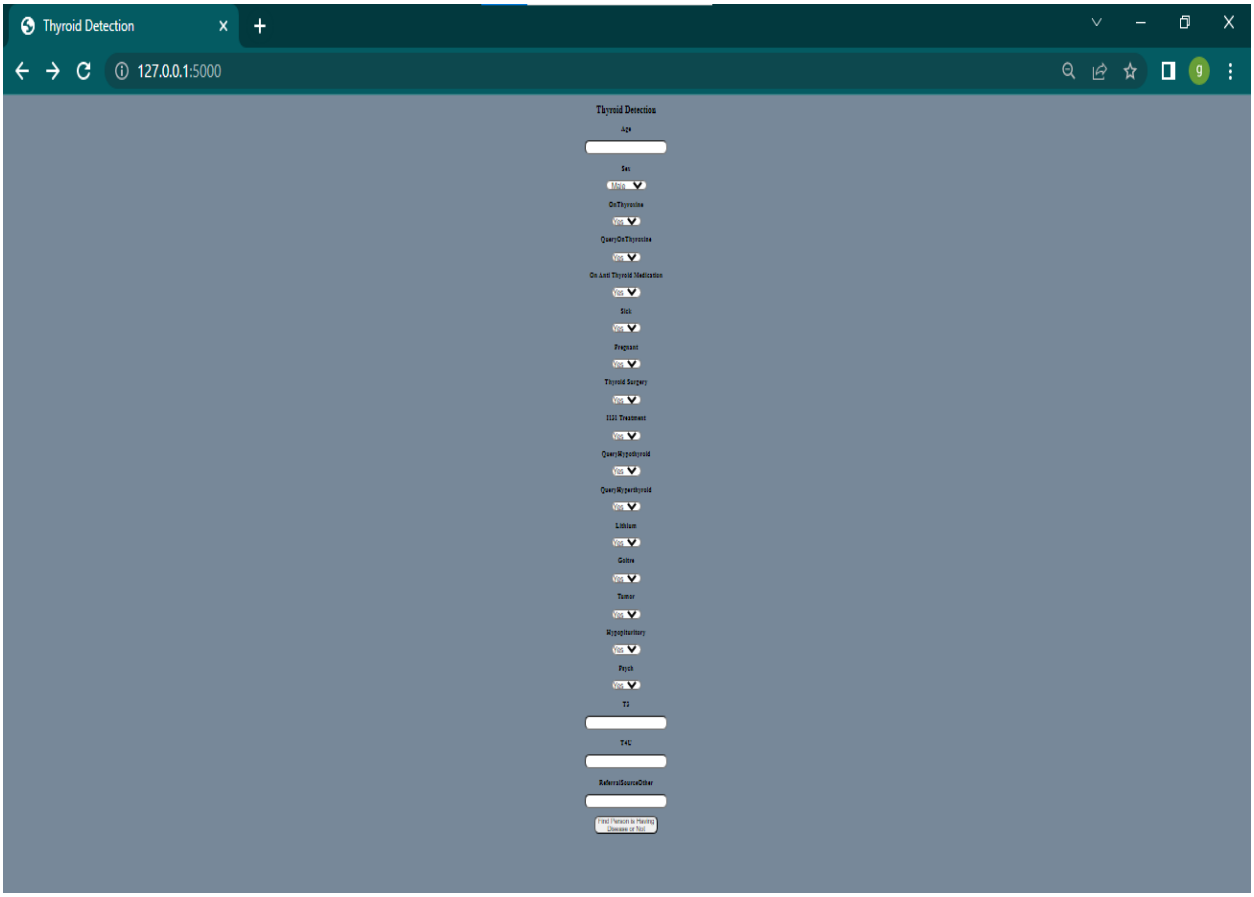
An empathy map is a square divided into four quadrants with the user or client in the middle. Each of the four quadrants comprises a category that helps us delve into the mind of the user. The four empathy map quadrants look at what the user says, thinks, feels, and does.



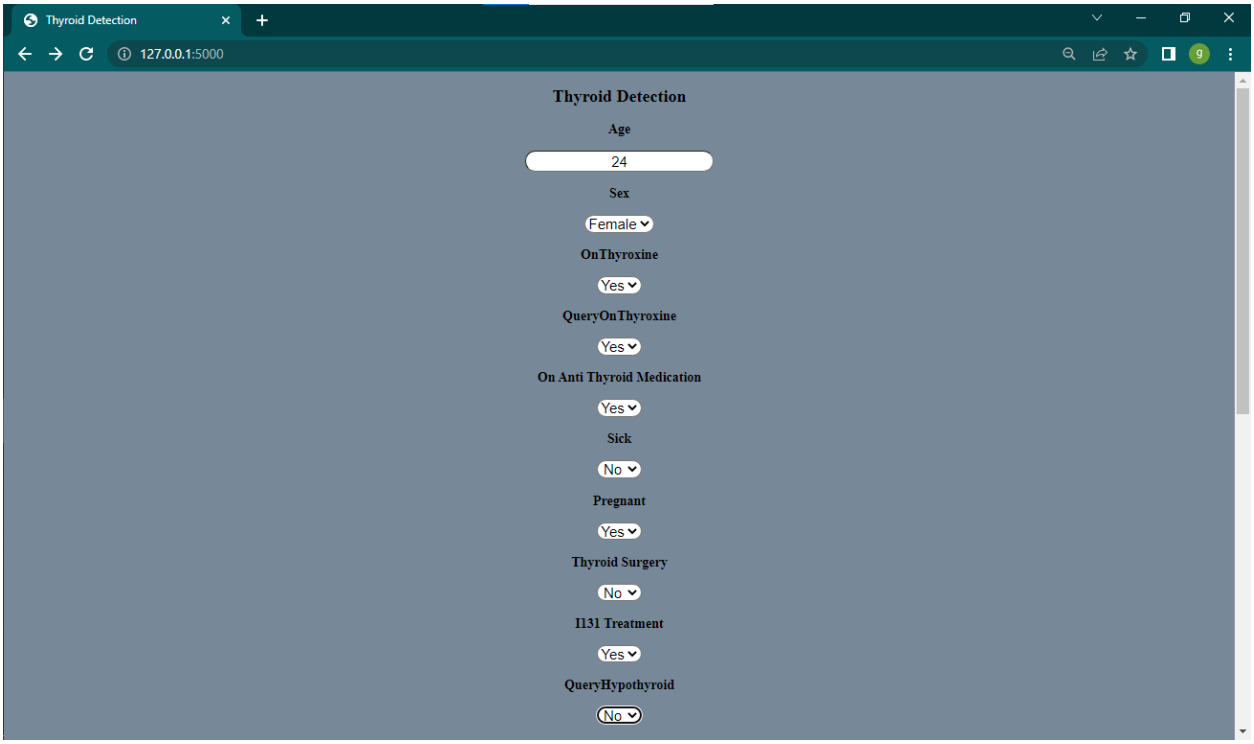
2.2 IDEATION MAP



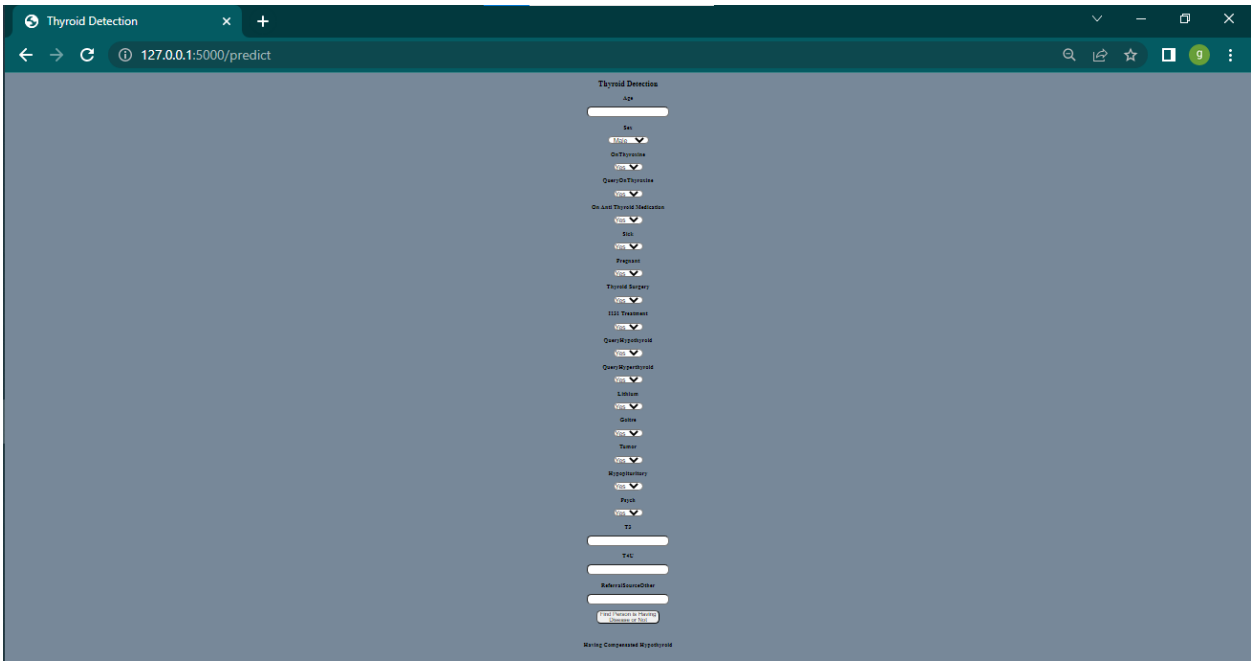
RESULT

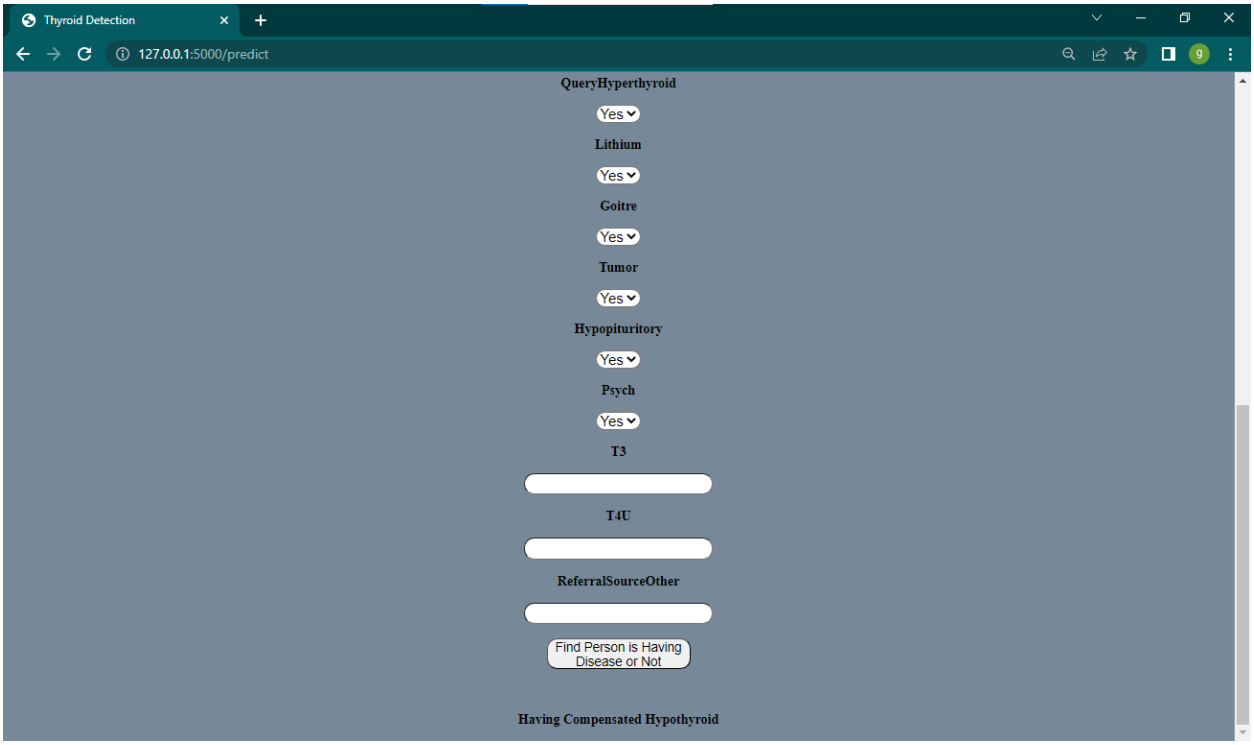


HOME PAGE


Thyroid Detection
127.0.0.1:5000
Thyroid Detection
Age
24
Sex
Female
OnThyroxine
Yes
QueryOnThyroxine
Yes
On Anti Thyroid Medication
Yes
Sick
No
Pregnant
Yes
Thyroid Surgery
No
I131 Treatment
Yes
QueryHypothyroid
No

INPUT PAGE


Thyroid Detection
127.0.0.1:5000/predict
Thyroid Detection
Age
Sex
OnThyroxine
QueryOnThyroxine
On Anti Thyroid Medication
Sick
Pregnant
Thyroid Surgery
I131 Treatment
QueryHypothyroid
QueryHyperthyroid
Lithium
Goitre
Tumor
HypopituitaryPsych
TSH
T3
T4U
ReferralSourceOther
Find Person is Having Disease or Not
Having Compensated Hypothyroid



OUTPUT (PREDICTION)

ADVANTAGES AND DISADVANTAGES

4.1 ADVANTAGES

- This is used to predict the kidney disease of a patient on the basis of dataset available.
- The system processes the symptoms provided by the user as input and gives the output as the probability of the disease.
- It will also recommend necessary precautionary measures required to treat the predicted disease.
- It saves time and money.
- Using predictive analytics in healthcare can improve the quality of healthcare, collect more clinical data for personalized treatment, and successfully diagnose the medical condition of individual patient.
- This could help to improve patient care and the safety and effectiveness of medical procedures.

4.2 DISADVANTAGES

- It will cause delays in providing the output. So, machine learning significantly depends on the data and its quality.
- The data that machines process remains huge in quantity and differs greatly.

- It requires massive and expensive resources and high-quality expertise to set up that quality of infrastructure. Trials runs are costly as they would cost in terms of time and expenses.
- It will have some degree of inaccuracy. For a high degree of accuracy, algorithms should be developed so that they give reliable results.

APPLICATIONS

6.1 APPLICATIONS

- Machine learning algorithm can be used in medical imaging (such as X-rays or MRI scans) using pattern recognition to look for patterns that indicate a particular disease. This could potentially help doctors make quicker, more accurate diagnoses.
- Can be used in health monitoring system.
- No need of expensive devices.
- Patients who are not willing to take test, can use this system.
- Can be used in daily checkup.

CONCLUSION

6.1 CONCLUSION

With the passage of time, diseases of the kidney are becoming increasingly widespread. These are solely going to get worse in the future, thanks to ongoing technological improvements. Despite the truth that people are becoming more health-conscious and enrolling in yoga and dancing classes, the sedentary lifestyle and facilities that are continuously being delivered and improved will proceed to be an issue. As a result, in this situation, our project will be rather recommended to society.

FUTURE SCOPE

7.1 FUTURE SCOPE

A future selection-based machine learning algorithm is proposed to predict three chronic diseases, namely, diabetes, heart attack, and cancer. Machine learning techniques, such as natural language processing (NLP) and image processing, help streamline data collection and convert data into a standard format. This can lead to enhancement in identifying clinical patterns and assist with better predictions. Machine learning with Quantum can improve the analysis of data and get more profound insights.

APPENDIX

8.1 SOURCE CODE

```
from flask import Flask, render_template, request
import pickle
from sklearn.preprocessing import StandardScaler
import logging
import os
app = Flask(__name__)


c_handler = logging.StreamHandler()
c_handler.setLevel(logging.INFO)
c_format = logging.Formatter('%(name)s - %(levelname)s - %(message)s')
c_handler.setFormatter(c_format)
logging.basicConfig(filename='test.log', filemode='w+', format='%(asctime)s %(message)s')
lg = logging.getLogger()
lg.addHandler(c_handler)
open(os.getcwd() + 'test.log', 'a')


model = pickle.load(open('thyroid.pkl', 'rb'))
@app.route('/',methods=['GET'])
def Home():
    try:
        return render_template('index.html')
```

```
    except Exception as e:
        lg.error(e)
        return render_template('error.html', message="Check logs for more info")


standard_to = StandardScaler()
@app.route("/predict", methods=['POST'])
def predict():
    try:
        if request.method == 'POST':
            Age=request.form['Age']
            Sex=request.form['Sex']
            if(Sex=='Male'):
                Sex=1
            else:
                Sex=0
            OnThyroxine=request.form['OnThyroxine']
            if(OnThyroxine=='Yes'):
                OnThyroxine=1
            else:
                OnThyroxine=0
            QueryOnThyroxine=request.form['QueryOnThyroxine']
            if(QueryOnThyroxine=='Yes'):
                QueryOnThyroxine=1
            else:
                QueryOnThyroxine=0
```

```
OnAntiThyroidMedication=request.form['OnAntiThyroidMedication']
if(OnAntiThyroidMedication=='Yes'):
    OnAntiThyroidMedication=1
else:
    OnAntiThyroidMedication=0
Sick=request.form['Sick']
if(Sick=='Yes'):
    Sick=1
else:
    Sick=0
Pregnant=request.form['Pregnant']
if(Pregnant=='Yes'):
    Pregnant=1
else:
    Pregnant=0
ThyroidSurgery=request.form['ThyroidSurgery']
if(ThyroidSurgery=='Yes'):
    ThyroidSurgery=1
else:
    ThyroidSurgery=0
I131_Treatment=request.form['I131']
if(I131_Treatment=='Yes'):
    I131_Treatment=1
else:
    I131_Treatment=0
```

```
QueryHypothyroid=request.form['QueryHypothyroid']
if(QueryHypothyroid=='Yes'):
    QueryHypothyroid=1
else:
    QueryHypothyroid=0
QueryHyperthyroid=request.form['QueryHyperthyroid']
if(QueryHyperthyroid=='Yes'):
    QueryHyperthyroid=1
else:
    QueryHyperthyroid=0
Lithium=request.form['Lithium']
if(Lithium=='Yes'):
    Lithium=1
else:
    Lithium=0
Goitre=request.form['Goitre']
if(Goitre=='Yes'):
    Goitre=1
else:
    Goitre=0
Tumor=request.form['Tumor']
if(Tumor=='Yes'):
    Tumor=1
else:
    Tumor=0
```

```
        Hypopituritory=request.form['Hypopituritory']
        if(Hypopituritory=='Yes'):
            Hypopituritory=1
        else:
            Hypopituritory=0
        Psych=request.form['Psych']
        if(Psych=='Yes'):
            Psych=1
        else:
            Psych=0
        T3=float(request.form['T3'])
        T4U=float(request.form['T4U'])
        ReferralSourceOther=float(request.form['ReferralSourceOther'])

prediction=model.predict([[Age,Sex,OnThyroxine,QueryOnThyroxine,OnAntiThyroidMedication,Sick,Pregnant,ThyroidSurgery,I131_Treatment,QueryHypothyroid,QueryHyperthyroid,Lithium,Goitre,Tumor,Hypopituritory,Psych,T3,T4U,ReferralSourceOther]])
        print(prediction)
        output=round(prediction[0],2)
        print(output)
        #lg.info(Age + "-" + Sex + "-" + OnThyroxine + "-" + QueryOnThyroxine + "-" + OnAntiThyroidMedication + "-" + Sick + "-" + Pregnant + "-" + ThyroidSurgery + "-" + I131_Treatment + "-" + QueryHypothyroid + "-" + QueryHyperthyroid + "-" + Lithium + "-" + Goitre + "-" + Tumor + "-" + Hypopituritory + "-" + Psych + "-" + T3 + "-" + T4U + "-" + ReferralSourceOther)
        if output==0:
```

```
            return render_template('index.html',prediction_texts="Not having Thyroid")
        elif output==1:
            return render_template('index.html',prediction_texts="Having Compensated Hypothyroid")
        elif output==2:
            return render_template('index.html',prediction_texts="Having Primary Hypothyroid")
        else:
            return render_template('index.html',prediction_texts="Having Secondary Hypothyroid")
    else:
        return render_template('index.html')
  except Exception as e:
    lg.error(e)
    print("Exception Raise!! Kindly Check the conditions for Inputs Entered",e)

if __name__=="__main__":
  app.run(debug=True)
```