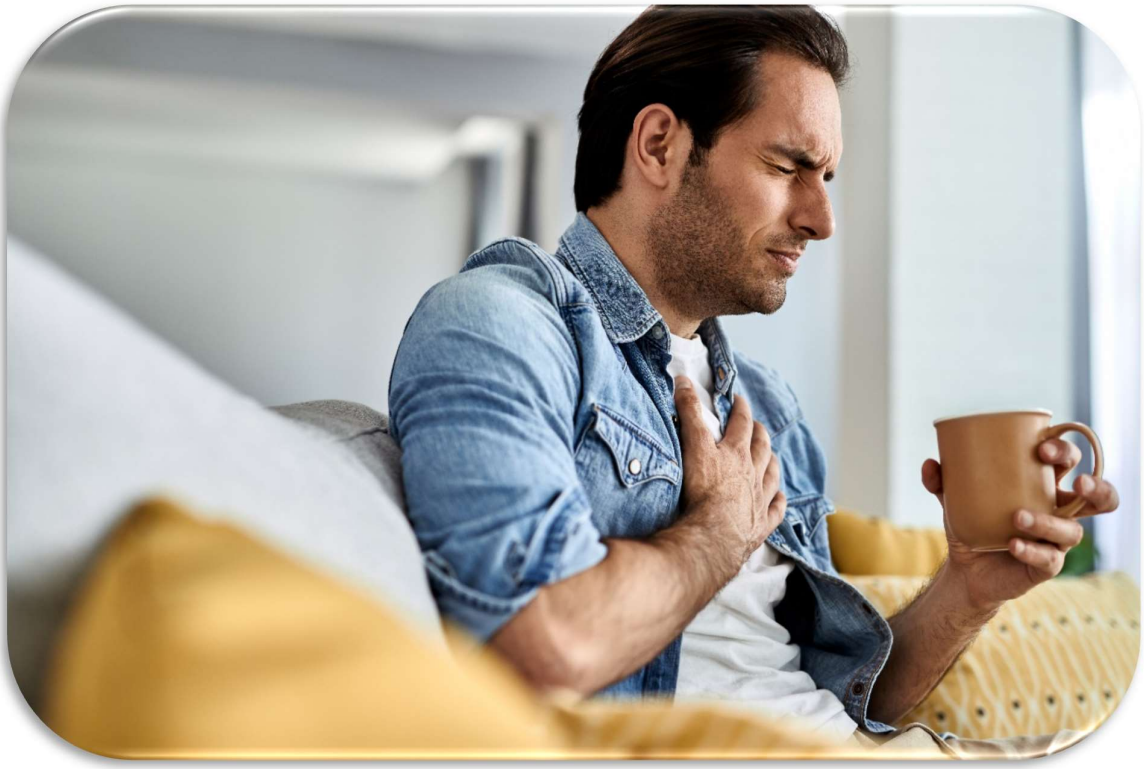
Hack for Health
(Hackathon From GeekForGeeks)



Project Title:

Quick Health Analyser

Contributor Name:

Vinay Kumar

Abhay Swarnakar

TABLE OF CONTENT

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.0 INTRODUCTION:-

Early illness identification is an important worldwide public health goal. Using machine learning (ML) approaches to assess consumer diagnostic data has emerged as a game-changing approach to early-stage illness identification. This paradigm focuses on three specific diseases: heart disease, diabetes, and Parkinson's disease. Machine learning algorithms, particularly classification models, play an important role in this process. These models are trained on various datasets containing diagnostic information from people. The input characteristics might comprise a variety of health indicators, genetic markers, lifestyle variables, and other pertinent data. Machine learning algorithms learn to discover patterns and develop connections with the early stages of heart disease, diabetes, and Parkinson's disease by analysing these complex datasets. The importance of early detection cannot be emphasized. Machine learning methods, particularly classification models, are critical for tackling the Quick Health Analyser Problem. These algorithms are trained on a variety of datasets, including genetic information, lifestyle factors, and health markers. Machine learning algorithms can detect small patterns and correlations in these complicated datasets, which may foretell the beginning of illness. Recognizing these early-stage indications is critical for creating prediction models that can warn healthcare workers to possible concerns. Ethical issues, data privacy, and model interpretability become critical when deploying these technologies in real-world healthcare settings. In short, using machine learning into early illness detection efforts represents a paradigm change in healthcare. We want to transform diagnostic capabilities, empower healthcare practitioners, and, eventually, improve the quality of life for people throughout the world by using the power of powerful algorithms.

CHAPTER 2

PROBLEM DEFINATION AND PROPOSED MODEL

2.1 PROBLEM DEFINITION: -

At its foundation, the Quick Health Analyser Problem is motivated by the understanding that standard healthcare practices, which frequently respond to symptoms as they emerge, have limits. Waiting for symptoms to appear might delay therapy beginning, lowering the effectiveness of therapies. The idea is to use modern technologies, notably machine learning, to filter through massive datasets containing a variety of health-related factors and indicators. Machine learning methods, particularly classification models, are critical for tackling the Quick Health Analyser Problem. These algorithms are trained on a variety of datasets, including genetic information, lifestyle factors, and health markers. Machine learning algorithms can detect small patterns and correlations in these complicated datasets, which may foretell the beginning of illness. Recognizing these early-stage indications is critical for creating prediction models that can warn healthcare workers to possible concerns. However, the Quick Health Analyser Problem is not without problems. Implementing machine learning-based diagnostic systems necessitates a multidisciplinary approach. Collaboration among healthcare specialists, data scientists, and technology experts is critical for navigating the intricacies of healthcare data and ensuring the ethical use of new technologies. Data privacy, model interpretability, and the ethical concerns of deploying predictive technology in healthcare settings all require careful study.

2.2 OBJECTIVE: -

The value of early detection cannot be emphasized. For illnesses such as heart disease, diabetes, and Parkinson's, early detection enables therapies that can slow or reverse disease development. Tailored treatment strategies can be implemented, thereby reducing problems and increasing overall patient outcomes. Furthermore, early detection improves the efficiency of healthcare systems by decreasing the demand on resources that would otherwise be used to manage advanced-stage disorders. Machine learning methods, particularly classification models, are critical for tackling the Quick Health Analyser Problem. These algorithms are trained on a variety of datasets, including genetic information, lifestyle factors, and health markers. Machine learning algorithms can detect small patterns and correlations in these complicated

datasets, which may foretell the beginning of illness. Recognizing these early-stage indications is critical for creating prediction models that can warn healthcare workers to possible concerns.

We are Work on Only Three Type of Diseases.

1. Heart Disease
2. Diabetes Disease
3. Parkinson's (Neurological Disease)

2.2.1 Heart Disease:

Heart disease, encompassing a range of conditions affecting the heart and blood vessels, stands as a formidable health challenge worldwide. It is imperative to delve into a comprehensive understanding of this complex ailment, considering its prevalence, risk factors, diagnostic methods, and the role of cutting-edge technologies, particularly machine learning, in its early detection and management.

Prevalence and Impact:

Heart disease, often used interchangeably with cardiovascular disease, remains a leading cause of morbidity and mortality globally. According to the World Health Organization (WHO), an estimated 17.9 million lives are claimed each year due to cardiovascular diseases, accounting for approximately 31% of all global deaths. The impact is not only measured in lives lost but also in the substantial economic burden it places on healthcare systems.

Risk Factors:

Understanding the risk factors associated with heart disease is paramount for both prevention and timely intervention. These factors can be broadly categorized into modifiable and non-modifiable. Non-modifiable risk factors include age, gender, and genetics, while modifiable factors encompass lifestyle choices like diet, physical activity, smoking, and excessive alcohol consumption. The interplay of these factors underscores the complexity of heart disease.

Diagnostic Approaches:

Accurate diagnosis forms the bedrock of effective heart disease management. Traditional diagnostic methods include electrocardiograms (ECG or EKG), echocardiograms, and stress tests. These techniques provide valuable insights into the heart's function and structure. Advanced imaging modalities such as cardiac MRI and CT angiography offer detailed anatomical information. Biomarkers like troponin and B-type natriuretic peptide (BNP) aid in detecting cardiac damage and heart failure, respectively.

Role of Machine Learning in Early Detection:

The integration of machine learning (ML) into healthcare, particularly for early disease detection, has emerged as a transformative approach. ML algorithms, fueled by vast datasets, excel in identifying subtle patterns indicative of pre-symptomatic stages. In the context of heart disease, these algorithms analyze diverse data points, including patient demographics, lifestyle factors, and medical history, to predict the likelihood of developing cardiovascular conditions. Early detection facilitated by ML not only enhances prognosis but also enables tailored intervention strategies.

Challenges in Implementation:

While ML holds immense promise, its integration into healthcare is not without challenges. Ethical considerations, data privacy concerns, and the need for transparent and interpretable models are critical aspects. Ensuring that the benefits of ML are accessible across diverse demographic groups while minimizing biases is an ongoing endeavor.

2.2.2 Diabetes Disease:

Diabetes, a chronic metabolic disorder characterized by elevated blood sugar levels, has emerged as a pervasive health challenge globally. This comprehensive exploration aims to shed light on the multifaceted nature of diabetes, encompassing its prevalence, risk factors, diagnostic approaches, and the evolving role of innovative technologies, particularly machine learning, in early detection and management.

Prevalence and Impact:

Diabetes has reached epidemic proportions, affecting millions of lives and posing a significant burden on healthcare systems. According to the International Diabetes Federation (IDF), approximately 537 million adults were living with diabetes in 2021, with projections indicating a rise to 643 million by 2030. The impact of diabetes extends beyond its immediate health implications, contributing to complications such as cardiovascular disease, kidney failure, and vision impairment.

Risk Factors:

Understanding the risk factors associated with diabetes is crucial for effective prevention and management. While genetics and family history play a role, lifestyle factors take center stage. Sedentary lifestyles, unhealthy dietary patterns, obesity, and age are key contributors. The intricate interplay between genetic predisposition and environmental factors underscores the complexity of diabetes.

Diagnostic Approaches:

Accurate and timely diagnosis forms the linchpin of diabetes management. Traditional diagnostic methods involve measuring fasting blood sugar levels, oral glucose tolerance tests, and glycated hemoglobin (HbA1c) tests. These tests provide insights into the body's ability to regulate glucose. Continuous glucose monitoring (CGM) systems offer real-time data, enhancing the precision of diabetes management.

Role of Machine Learning in Early Detection:

The integration of machine learning (ML) into diabetes care has ushered in a new era of early detection and personalized intervention. ML algorithms analyze diverse datasets, including patient demographics, lifestyle factors, and genetic markers, to identify patterns indicative of pre-diabetic states. This proactive approach allows for early intervention, lifestyle modifications, and personalized treatment plans tailored to individual needs.

Challenges in Implementation:

While ML holds promise in revolutionizing diabetes care, its implementation is not without challenges. Ethical considerations, data privacy, and the need for transparent and interpretable models are paramount. Ensuring accessibility and fairness in ML-driven diabetes solutions across diverse populations is an ongoing pursuit.

2.2.3 Parkinson's (Neurological Disease):

Parkinson's disease, a neurodegenerative disorder that primarily affects movement, has long been a subject of intensive research and medical scrutiny. This in-depth exploration aims to unravel the multifaceted nature of Parkinson's, delving into its clinical features, underlying mechanisms, diagnostic challenges, and the transformative role of machine learning in early detection and management.

Clinical Features and Impact:

Parkinson's disease manifests through a spectrum of clinical features, with motor symptoms like tremors, bradykinesia, and rigidity being hallmark indicators. Non-motor symptoms, including cognitive impairment, mood disorders, and autonomic dysfunction, contribute to the disease's pervasive impact on the quality of life. As a progressive condition, Parkinson's poses not only physical challenges but also places a substantial emotional and socioeconomic burden on individuals and their families.

Underlying Mechanisms and Pathophysiology:

The pathophysiology of Parkinson's is complex and involves the degeneration of dopaminergic neurons in the substantia nigra region of the brain. This leads to the disruption of

neurotransmitter balance, particularly dopamine, crucial for regulating movement. Accumulation of alpha-synuclein protein in the form of Lewy bodies further characterizes the disease. While these mechanisms provide insights, the exact etiology remains elusive, emphasizing the need for continued research.

Diagnostic Challenges:

Diagnosing Parkinson's disease poses significant challenges, particularly in its early stages. Clinical assessments, neuroimaging, and response to dopaminergic medications contribute to the diagnostic process. However, misdiagnosis and delayed identification are not uncommon, hindering timely intervention. The elusive nature of pre-symptomatic stages underscores the imperative for innovative diagnostic approaches.

Role of Machine Learning in Early Detection:

The advent of machine learning (ML) has brought a paradigm shift in Parkinson's disease detection. ML algorithms analyse extensive datasets, including clinical assessments, genetic markers, and even voice and gait patterns, to identify subtle patterns indicative of preclinical Parkinson's. These data-driven approaches hold promise for early and accurate detection, enabling interventions that may modify the course of the disease.

2.3 PROPOSED MODEL

Machine learning methods, particularly classification models, are critical for tackling the Quick Health Analyser Problem. These algorithms are trained on a variety of datasets, including genetic information, lifestyle factors, and health markers. Machine learning algorithms can detect small patterns and correlations in these complicated datasets, which may foretell the beginning of illness. Recognizing these early-stage indications is critical for creating prediction models that can warn healthcare workers to possible concerns. Developing a suggested model for early illness detection entails combining several components, including data gathering, feature selection, model building, and ethical concerns. The following is an outline of a potential paradigm for early illness detection.

2.3.1 Data Collection:

2.3.1.1 Diverse Dataset: Gather a comprehensive dataset that includes a range of health indicators, genetic markers, and lifestyle variables relevant to the targeted diseases (e.g., heart disease, diabetes, Parkinson's).

2.3.1.2 Quality Assurance: Ensure data quality by addressing issues such as missing values, outliers, and inconsistencies.

2.3.1.3 Ethical Data Usage: Prioritize patient privacy and obtain explicit consent for data usage.

2.3.2 Feature Selection: -

2.3.2.1 Identification of Relevant Features: Employ feature selection techniques to identify the most informative variables contributing to disease prediction.

2.3.3 Model Development:

2.3.3.1 Machine Learning Algorithms: Select appropriate machine learning algorithms based on the nature of the data and the specific characteristics of the diseases.

2.3.3.2 Training and Validation: Split the dataset into training and validation sets to train the model and assess its performance.

2.3.3.3 Ensemble Methods: Explore ensemble methods to combine predictions from multiple models, enhancing overall accuracy.

2.3.4 Validation and Evaluation:

2.3.4.1 Cross-Validation: Implement cross-validation techniques to robustly evaluate the model's performance.

2.3.4.2 External Validation: Validate the model on external datasets to assess its generalizability.

2.3.6 Deployment:

2.5.6.1 User-Friendly Interface: Design an intuitive interface for healthcare practitioners to input data and interpret model outputs.

2.5.6.2 Integration with Healthcare Systems: Ensure seamless integration with existing healthcare systems for easy adoption.

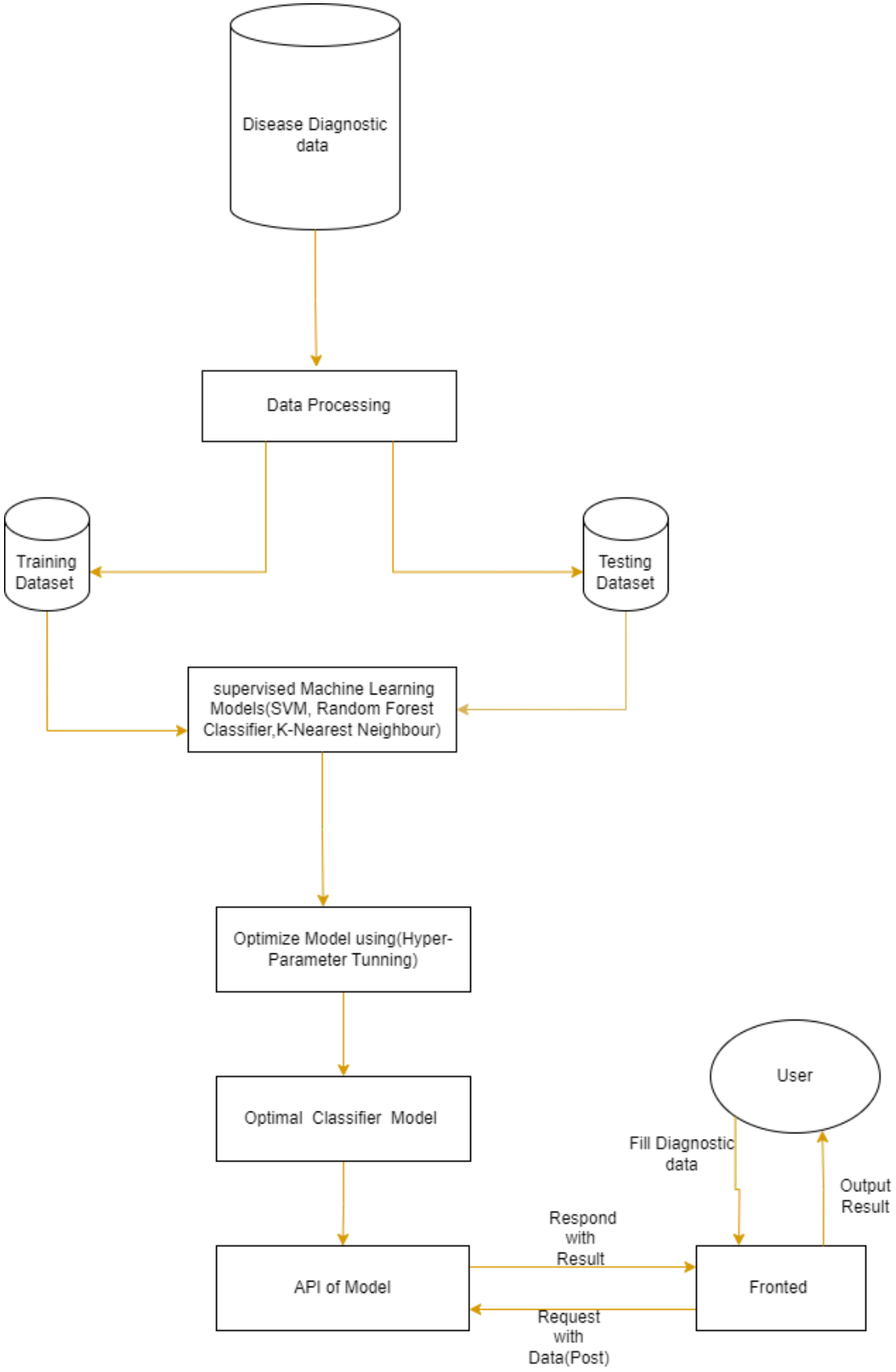


Fig 2.1. Proposed Model

CHAPTER 3

METHDOLOGY

3.1 Data Collection And Sources :

We use datasets for building models from Kaggle, UCI-datasets. Where link are there

3.1.1 Heart Diseases :

- https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset
- https://archive.ics.uci.edu/dataset/45/heart+disease

3.1.2 Diabetes Disease:

- binariks.com)https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database

3.1.3 Parkinson's Diseases :

- https://www.kaggle.com/datasets/naveenkumar20bps1137/parkinsons-disease-detection/data

3.2 Description Of Diseases Datasets:

1 Heart Diseases:

Variable Name	Role	Type	Demographic	Description	Units	Missing Values
age	Feature	Integer	Age	Age of Patient	years	no
sex	Feature	Categorical	Sex	Sex (male:1,Female:0)		no
cp	Feature	Categorical				no
trestbps	Feature	Integer		resting blood pressure (on admission to the hospital)	mm Hg	no
chol	Feature	Integer		serum cholestoral	mg/dl	no
fbs	Feature	Categorical		fasting blood sugar > 120 mg/dl		no
restecg	Feature	Categorical				no
thalach	Feature	Integer		maximum heart rate achieved		no

Variable Name	Role	Type	Demographic	Description	Units	Missing Values
exang	Feature	Categorical		exercise induced angina		no
oldpeak	Feature	Integer		ST depression induced by exercise relative to rest		no

Table 3.1 Data Dictionary of Heart Diseases

2 Diabetes Diseases:

Variable Name	Role	Type	Demographic	Description	Units	Missing Values
Pregnancies	Feature	Integer	No of Pregnancies		years	no
Glucose	Feature	Integer	Glucose	Sex (male:1,Female:0)		no
Blood Pressure	Feature	Integer		resting blood pressure (on admission to the hospital)	mm Hg	no
SkinThickness	Feature	Integer				no
DiabetesPedigreeFunction	Feature	Numeric		s	mg/dl	no
Insulin	Feature	Numeric		fasting blood sugar > 120 mg/dl		no
BMI	Feature	Integer				no
Age	Feature	Integer	Age		years	no
Status	Feature	Integer		Result (Yes =1,no 0)		no

Table 3.2 Data Dictionary Diabetes

3 Parkinson's Diseases

name - ASCII subject name and recording number

MDVP:Fo(Hz) - Average vocal fundamental frequency

MDVP:Fhi(Hz) - Maximum vocal fundamental frequency

MDVP:Flo(Hz) - Minimum vocal fundamental frequency

Five measures of variation in Frequency

MDVP:Jitter(%) - Percentage of cycle-to-cycle variability of the period duration

MDVP:Jitter(Abs) - Absolute value of cycle-to-cycle variability of the period duration

MDVP:RAP - Relative measure of the pitch disturbance

MDVP:PPQ - Pitch perturbation quotient

Jitter:DDP - Average absolute difference of differences between jitter cycles

Six measures of variation in amplitude

MDVP:Shimmer - Variations in the voice amplitdue

MDVP:Shimmer(dB) - Variations in the voice amplitdue in dB

Shimmer:APQ3 - Three point amplitude perturbation quotient measured against the average of the three amplitude

Shimmer:APQ5 - Five point amplitude perturbation quotient measured against the average of the three amplitude

MDVP:APQ - Amplitude perturbation quotient from MDVP

Shimmer:DDA - Average absolute difference between the amplitudes of consecutive periods

Two measures of ratio of noise to tonal components in the voice

NHR - Noise-to-harmonics Ratio and

HNR - Harmonics-to-noise Ratio

status - Health status of the subject (one) - Parkinson's, (zero) - healthy

Two nonlinear dynamical complexity measures

RPDE - Recurrence period density entropy

D2 - correlation dimension

DFA - Signal fractal scaling exponent

Three nonlinear measures of fundamental frequency variation

spread1 - discrete probability distribution of occurrence of relative semitone variations

spread2 - Three nonlinear measures of fundamental frequency variation

PPE - Entropy of the discrete probability distribution of occurrence of relative semitone variations

3.3 SYSTEM REQUIREMENTS

3.3.1 Operating System

- **Windows 7 or Later**

3.3.2 Hardware Requirement

- **CPU i3-6gen/Dual Core**
- **Ram 4/8Gb**
- **Storage 128GB min**

3.4 FRONTED

Language :

- HTML
- CSS
- JavaScript
- ReactJS

Platform:

- VS code
- Git
- GitHub

3.5 BACKEND

Language :

- Python

Library:

- FastAPI
- Pandas

- NumPy
- Scikit-learn
- Matplotlib
- Pickle
- Json

Platform:

- Anaconda
- Jupyter Notebook
- Git
- VS code

3.6 API(APPLICATION PROGRAMMING INTERFACE):

Unicorn is an ASGI(Asynchronous Server Gateway Interface) Server that's used to serve Python web applications, Particularly Those built using asynchronous frameworks like Fast API and Starlitte. ASGI is designed to handle asynchronous web applications. Which can perform multiple tasks concurrently, making them highly efficient for handling many simultaneous connections.

```
import pickle as pi
from sklearn.preprocessing import FunctionTransformer
import uvicorn
from pydantic import BaseModel
import json
import pandas as pd
from fastapi import FastAPI
from fastapi.middleware.cors import CORSMiddleware
def warn(*args, **kwargs):
    pass
import warnings
warnings.warn = warn
warnings.filterwarnings('ignore')
# Initializing the fast API server
app = FastAPI()

origins = [
    "http://localhost.tiangolo.com",
    "https://localhost.tiangolo.com",
    "http://localhost",
    "http://localhost:8080",
    "http://localhost:3000",
]


app.add_middleware(
    CORSMiddleware,
    allow_origins=origins,
    allow_credentials=True,
    allow_methods=["*"],
```

```
    allow_headers=["*"],
)
class model_input(BaseModel):
    # heart Diease
    age:int|None=None
    sex: int|None=None
    cp: int|None=None
    trestbps: int|None=None
    chol:int|None=None
    fbs:int|None=None
    restecg:int|None=None
    thalach:int|None=None
    exang: int|None=None
    oldpeak: float|None=None
    slope:int|None=None
    ca:int|None=None
    thal:int|None=None
    # for parkison
    MDVP_Fo_Hz:float|None=None
    MDVP_Fhi_Hz:float|None=None
    MDVP_Flo_Hz:float|None=None
    MDVP_jitter_percentage:float|None=None
    MDVP_Jitter_Abs:float|None=None
    MDVP_RAP:float|None=None
    MDVP_PPQ:float|None=None
    Jitter_DDP:float|None=None
    MDVP_Shimmer:float|None=None
    MDVP_Shimmer_dB:float|None=None
    Shimmer_APQ3:float|None=None
    Shimmer_APQ5:float|None=None
    MDVP_APQ:float|None=None
    Shimmer_DDA:float|None=None
    NHR:float|None=None
    HNR:float|None=None
    RPDE:float|None=None
    DFA:float|None=None
    spread1:float|None=None
    spread2:float|None=None
    D2:float|None=None
    PPE:float|None=None
    # for Diabeties
    preg :int|None=None
    plas  :int|None=None
    pres :int|None=None
    skin :int|None=None
    insu :int|None=None
    bmi:float|None=None
    pedi :float|None=None
    # height:float|None=None
    # weight:float|None=None

# Output label return
def output(n,for_use):
    if n==1:
        return "Positive {f}".format(f=for_use)
    else:
        return "Negative {f}".format(f=for_use)

def result_heart(age,sex,cp,trestbps,chol,fbs,restecg,thalach,exang,oldpeak,slope,ca,thal):
    colmn =["age","sex","cp","trestbps","chol","fbs","restecg","thalach","exang","oldpeak","slope","ca","thal"]
    data=[[age,sex,cp,trestbps,chol,fbs,restecg,thalach,exang,oldpeak,slope,ca,thal]]
    if len(data[0])%13==0 and  not(any(item is  None for item in data[0])):
        print(data)
```

```
        x=pd.DataFrame(data=data,columns=colmn)
        print(x)
        filename="Backend/models/HeartDIease.pickle"
        loaded_model=pi.load(open(filename,'rb'))
        result=output(loaded_model.predict(x),"heart")
        # result="check_1"
                return json.dumps(result)
            else:
                return json.dumps("Not Proper Value")

    #  function  Diabeteis

    def result_Diabeties(Pregnancies,Glucose,BloodPressure,SkinThickness,Insulin,BMI,DiabetesPedigreeFunction,Age):
            colmn =['Pregnancies','Glucose','BloodPressure','SkinThickness','Insulin','BMI','DiabetesPedigreeFunction','Age']
            data=[[Pregnancies,Glucose,BloodPressure,SkinThickness,Insulin,BMI,DiabetesPedigreeFunction,Age]]
            if len(data[0])%8==0 and not(any(item is  None for item in data[0])):
                print(data)
                x=pd.DataFrame(data=data,columns=colmn)
                print(x)
                filename="Backend/models/Diabetes.pickle"
                loaded_model=pi.load(open(filename,'rb'))
                result=output(loaded_model.predict(x),"Diabetes")
                # result="check_2"
                return json.dumps(result)
            else:
                value="Not Proper Value"
                return json.dumps(value)

    #   Parkison function
    def result_parkinson(MDVP_Fo_Hz,MDVP_Fhi_Hz,MDVP_Flo_Hz,MDVP_jitter_percentage,MDVP_Jitter_Abs,MDVP_RAP,MDVP_PPQ,Jitter_DDP,MDVP_Shimmer,MDVP_Shimmer_dB,Shimmer_APQ3,Shimmer_APQ5,MDVP_APQ,Shimmer_DDA,NHR,HNR,RPDE,DFA,spread1,spread2,D2,PPE):
            column =['MDVP_Fo_Hz','MDVP_Fhi_Hz','MDVP_Flo_Hz','MDVP_jitter_percentage','MDVP_Jitter_Abs','MDVP_RAP','MDVP_PPQ','Jitter_DDP','MDVP_Shimmer','MDVP_Shimmer_dB','Shimmer_APQ3','Shimmer_APQ5','MDVP_APQ','Shimmer_DDA','NHR','HNR','RPDE','DFA','spread1','spread2','D2','PPE']
            data=[[MDVP_Fo_Hz,MDVP_Fhi_Hz,MDVP_Flo_Hz,MDVP_jitter_percentage,MDVP_Jitter_Abs,MDVP_RAP,MDVP_PPQ,Jitter_DDP,MDVP_Shimmer,MDVP_Shimmer_dB,Shimmer_APQ3,Shimmer_APQ5,MDVP_APQ,Shimmer_DDA,NHR,HNR,RPDE,DFA,spread1,spread2,D2,PPE]]
            print(any(item is None for item in data[0]))
            if len(data[0])%22==0 and any(item is not None for item in data[0]):
                print(data)
                x=pd.DataFrame(data=data,columns=column)
                print(x)
                filename="Backend/models/Parkison.pickle"
                loaded_model=pi.load(open(filename,'rb'))
                result=output(loaded_model.predict(x),"Parkinson")
                # result="check_3"
                return json.dumps(result)
            else:
                return json.dumps("Not Proper Value")
    # Setting up the home route
    @app.get("/")
    def read_root():
            return {"Data": "Hack-for-Health\n Welcome to online Early Health Prediction "}

    # # Setting up the prediction route
    @app.post("/heart")
```

```
145.        async def prediction_heart(input_parameters: model_input):
146.            age=input_parameters.age
147.            sex=input_parameters.sex
148.            cp=input_parameters.cp
149.            trestbps=input_parameters.trestbps
150.            chol=input_parameters.chol
151.            fbs=input_parameters.fbs
152.            restecg=input_parameters.restecg
153.            thalach=input_parameters.thalach
154.            exang=input_parameters.exang
155.            oldpeak=input_parameters.oldpeak
156.            slope=input_parameters.slope
157.            ca=input_parameters.ca
158.            thal=input_parameters.thal
159.            return result_heart(age,sex,cp,trestbps,chol,fbs,restecg,thalach,e
    xang,oldpeak,slope,ca,thal)
160.
161.        @app.post("/parkinson")
162.        async def prediction_parkinson(input_parameters:model_input):
163.            MDVP_Fo_Hz=input_parameters.MDVP_Fo_Hz
164.            MDVP_Fhi_Hz=input_parameters.MDVP_Fhi_Hz
165.            MDVP_Flo_Hz=input_parameters.MDVP_Flo_Hz
166.            MDVP_jitter_percentage=input_parameters.MDVP_jitter_percentage
167.            MDVP_Jitter_Abs=input_parameters.MDVP_Jitter_Abs
168.            MDVP_RAP=input_parameters.MDVP_RAP
169.            MDVP_PPQ=input_parameters.MDVP_PPQ
170.            Jitter_DDP=input_parameters.Jitter_DDP
171.            MDVP_Shimmer=input_parameters.MDVP_Shimmer
172.            MDVP_Shimmer_dB=input_parameters.MDVP_Shimmer_dB
173.            Shimmer_APQ3=input_parameters.Shimmer_APQ3
174.            Shimmer_APQ5=input_parameters.Shimmer_APQ5
175.            MDVP_APQ=input_parameters.MDVP_APQ
176.            Shimmer_DDA=input_parameters.Shimmer_DDA
177.            NHR=input_parameters.NHR
178.            HNR=input_parameters.HNR
179.            RPDE=input_parameters.RPDE
180.            DFA=input_parameters.DFA
181.            spread1=input_parameters.spread1
182.            spread2=input_parameters.spread2
183.            D2=input_parameters.D2
184.            PPE=input_parameters.PPE
185.            return result_parkinson(MDVP_Fo_Hz,MDVP_Fhi_Hz,MDVP_Flo_Hz,MDVP_ji
    tter_percentage,MDVP_Jitter_Abs,MDVP_RAP,MDVP_PPQ,Jitter_DDP,MDVP_Shimmer,MDVP
    _Shimmer_dB,Shimmer_APQ3,Shimmer_APQ5,MDVP_APQ,Shimmer_DDA,NHR,HNR,RPDE,DFA,sp
    read1,spread2,D2,PPE)
186.
187.        @app.post("/Diabeties")
188.        async def prediction_diabeties(input_parameters:model_input):
189.            Pregnancies=input_parameters.preg
190.            Glucose=input_parameters.plas
191.            BloodPressure=input_parameters.pres
192.            SkinThickness=input_parameters.skin
193.            Insulin=input_parameters.insu
194.            BMI=input_parameters.bmi
195.            DiabetesPedigreeFunction=input_parameters.pedi
196.            Age=input_parameters.age
197.            return result_Diabeties(Pregnancies,Glucose,BloodPressure,SkinThic
    kness,Insulin,BMI,DiabetesPedigreeFunction,Age)
198.
199.        # configuring the server host and port
200.        if __name__=="__main__":
201.            uvicorn.run(app,"0.0.0.0","10000")
```

Run code : `uvicorn Backend.Heatlh:app -host 0.0.0.0 -port 10000`

```
gitpod /workspace/Hack-for-Health (main) $ uvicorn Backend.Health:app --host 0.0.0.0 --port 10000
INFO:     Started server process [1007]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     Uvicorn running on http://0.0.0.0:10000 (Press CTRL+C to quit)
```

Fig 3.1 Backend startup

Link: http://0.0.0.0:10000 in Fronted Where Post request Generate

And past on this place 'POST'

```
axios
  // .post('http://localhost:8080/prediction', params)
  // .post(`${process.env.API_URL}+/Diabeties`, params)

  // .post('process.env.API_URL/Diabeties', params)
  .post('https://10000-genratecode-hackforheal-evgm2dlr09z.ws-us107.gitpod.io/Diabeties', params)
  .then((res) => {
    const data = res.data
    // const parameters = JSON.stringify(params)
    // const msg = `Prediction: ${data.prediction}\nInterpretation: ${data.interpretation}\nParameters: ${parameters}`
    // alert(data)
    // reset()
    console.log(data)
    setResult(data)
  },)
```

Fig 3.2 Fronted/src/Diabetes.jsx

```
    // .post(`${process.env.API_URL}+/heart`, params)
    .post('https://10000-genratecode-hackforheal-evgm2dlr09z.ws-us107.gitpod.io/heart', params)
    .then((res) => {
      const data = res.data

      setResult(data)
      // reset()
    })
    .catch((error) => alert(`Error: ${error.message}`))
}
```

Fig 3.3 Fronted/src/HeartDiseases.jsx

```
    .post('https://10000-genratecode-hackforheal-evgm2dlr09z.ws-us107.gitpod.io/parkinson', params)
    .then((res) => {
      const data = res.data

      console.log(typeof(data))
      console.log(data)
      setResult(data)
      // reset()
    })
    .catch((error) => alert(`Error: ${error.message}`))
}
```

Fig 3.4 Fronted/src/NeruologicalDiseases.jsx

3.7 USER INTERFACE

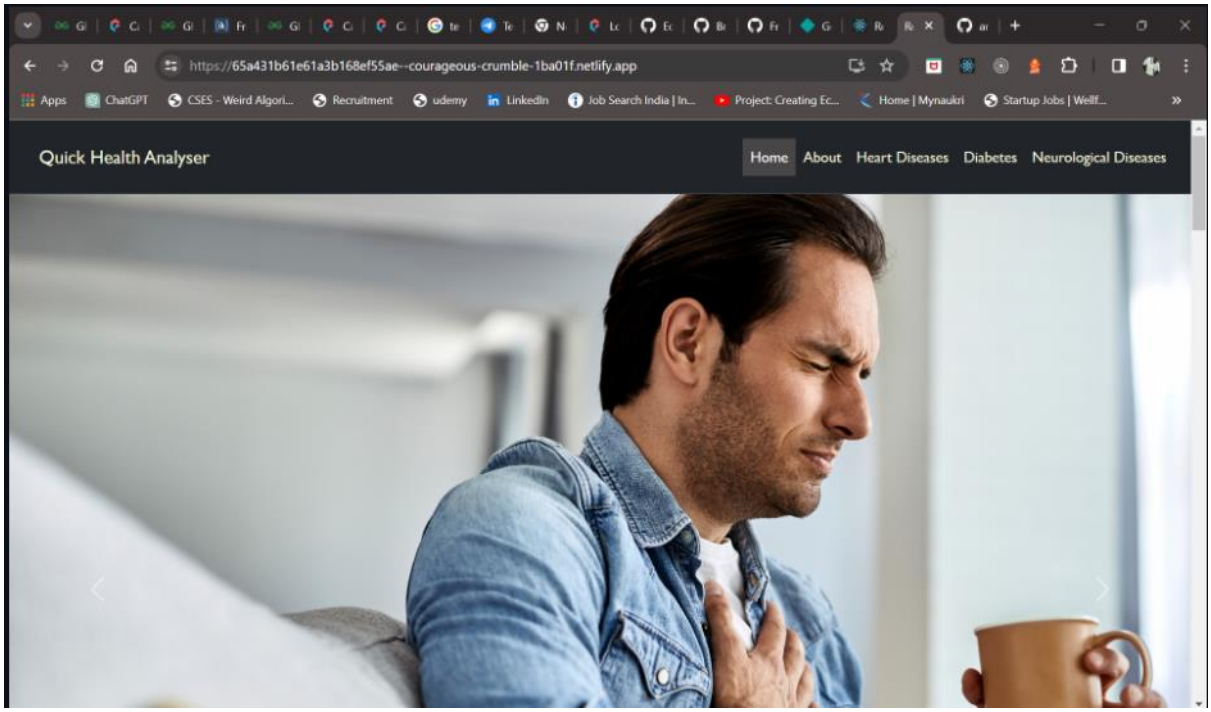


Fig 3.5 UI/Home Page

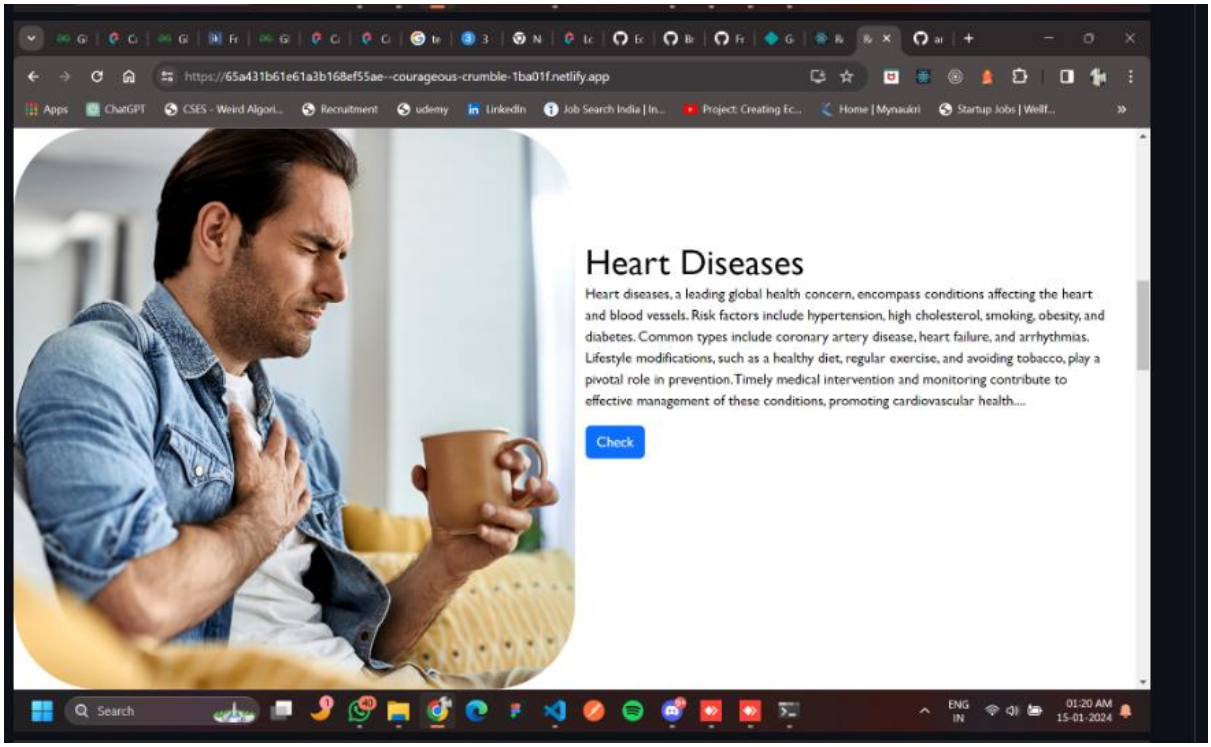


Fig 3.6 UI/Heart Diseases Home page

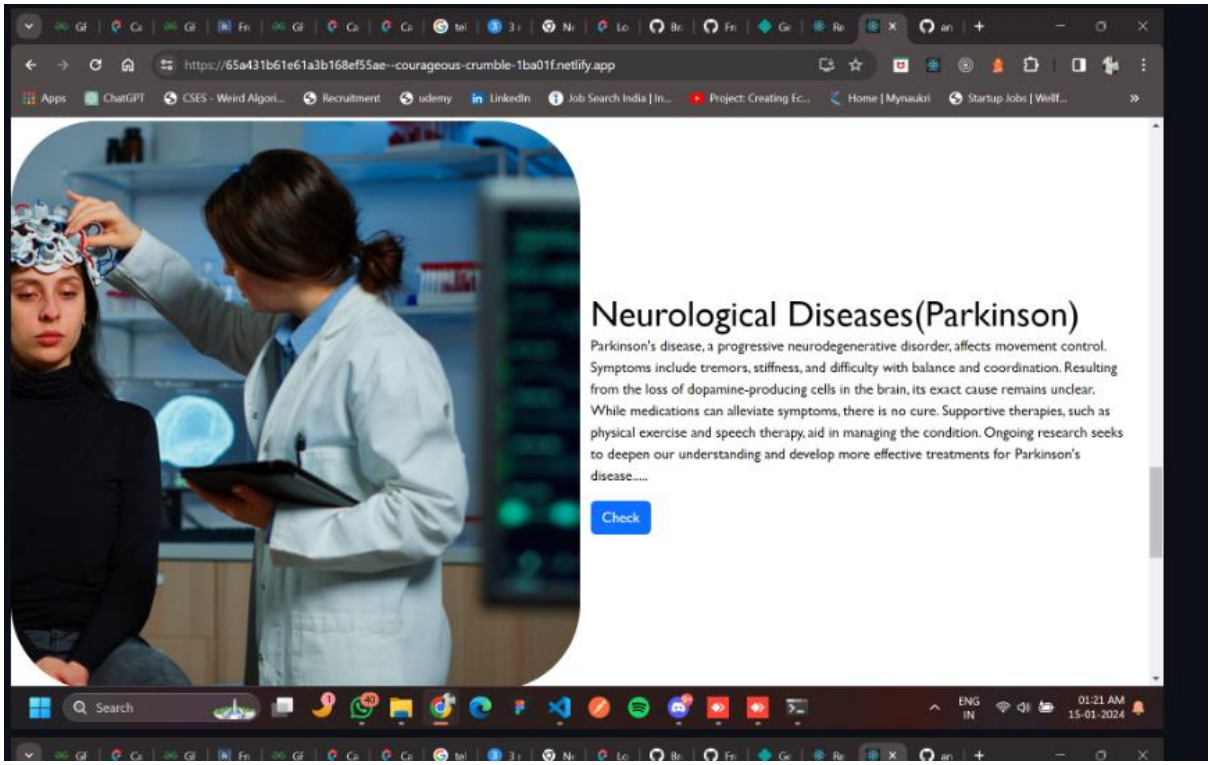


Fig 3.7 UI/Neurological Diseases(Parkinson)

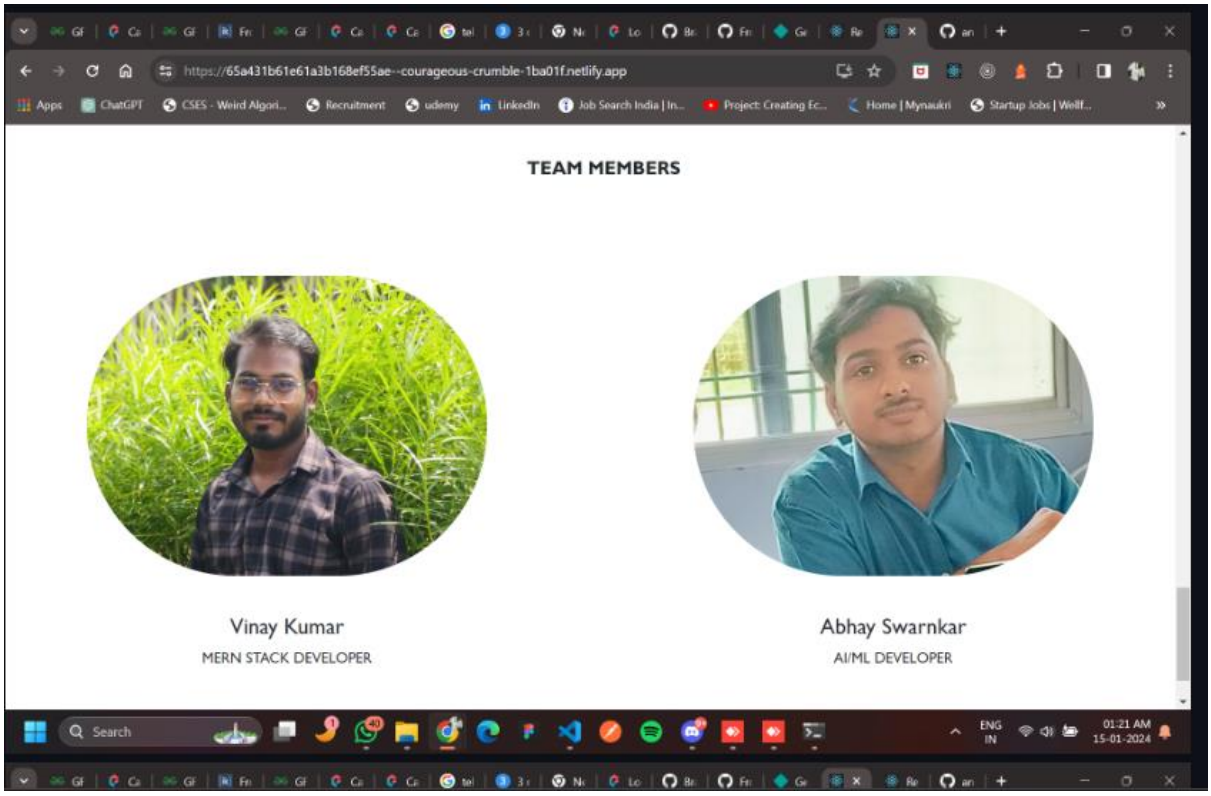


Fig 3.8 About Section

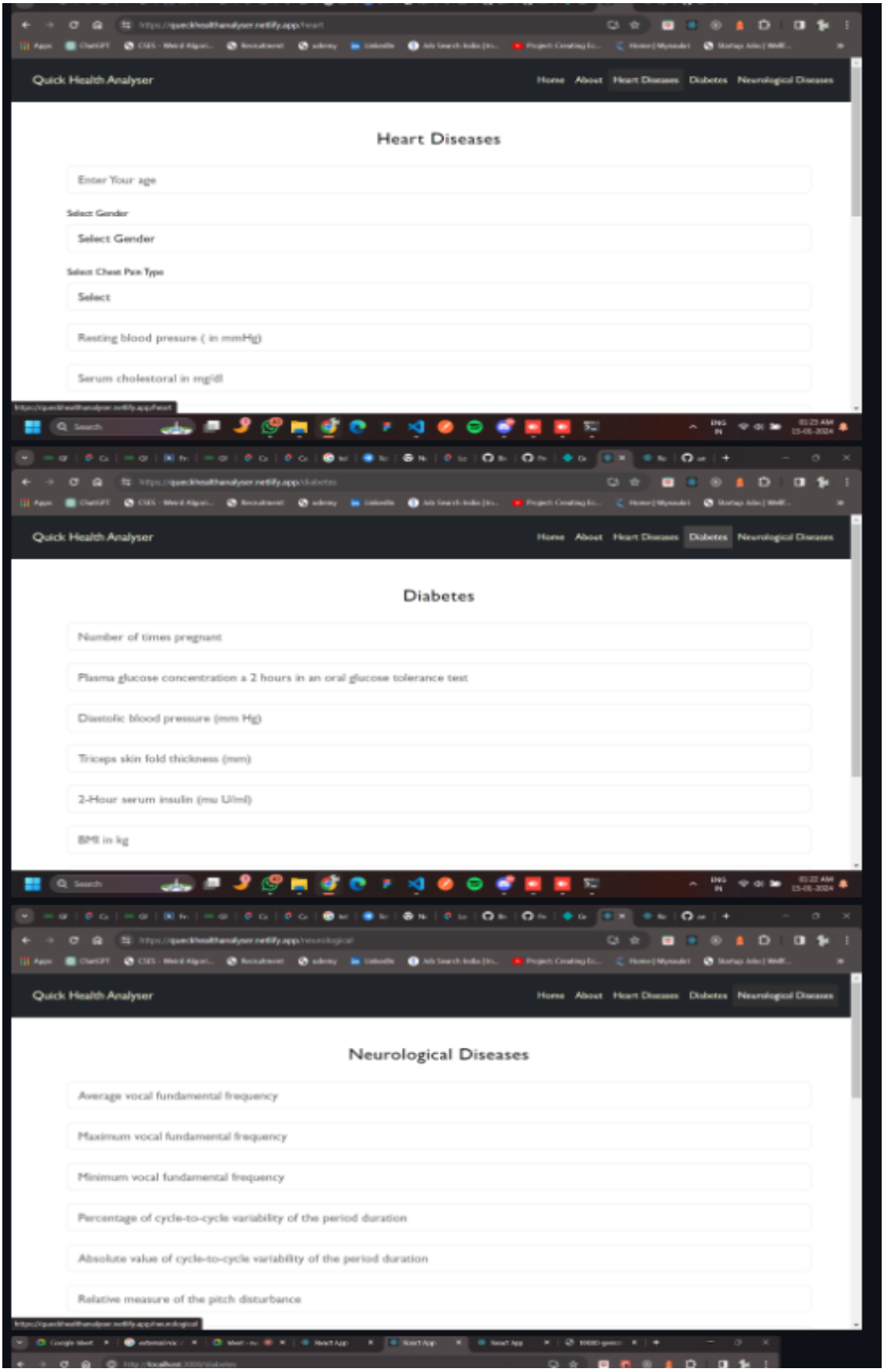


Fig 3.9 Diseases Input Section

3.8 RESULT

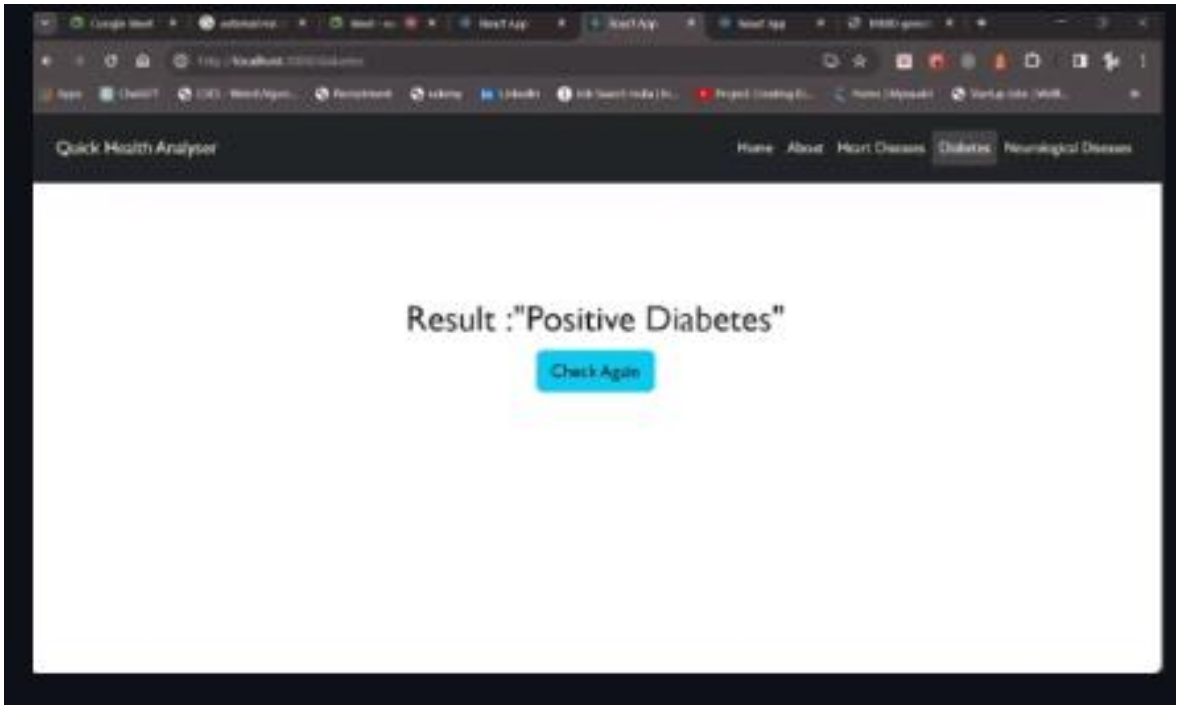


Fig 3.10 Output Show

CONCLUSION

The integration of machine learning into early illness detection epitomizes a paradigm shift in healthcare. It seeks to revolutionize diagnostic capabilities, empower healthcare practitioners, and ultimately elevate the quality of life on a global scale. As we navigate this transformative journey, a commitment to ethical practices and collaborative endeavors will be instrumental in realizing the full potential of machine learning in healthcare. using machine learning into early illness detection efforts represents a paradigm change in healthcare. We want to transform diagnostic capabilities, empower healthcare practitioners, and, eventually, improve the quality of life for people throughout the world by using the power of powerful algorithms In the context of machine learning applications for the Early Disease Detection Problem, notable achievements underscore the potential of these technologies. Achieving an 87% accuracy rate for heart diseases, 81% for diabetes, and an impressive 97% for Parkinson's demonstrates the efficacy of these models in real-world scenarios.

FUTURE SCOPE

- Integration of Advanced Technologies
- Expanding the Disease Portfolio
- Implementation of Telemedicine and Digital Health Platforms
- Add Prevacation and prevent method according diseases
- Patient Empowerment and Educations
- Innovative treatment Modalities:
- Integration of Image Technologies
- Blockchain for Data Security:

REFERENCE

- **AI/ML Algorithms for Early Disease Detection and Diagnosis**
- **(binariks.com)https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database**
- **https://www.kaggle.com/datasets/naveenkumar20bps1137/parkinsons-disease-detection/data**
- **https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset**
- **https://archive.ics.uci.edu/dataset/45/heart+disease**