MULTI-DISEASE PREDICT SOFT-WEB APP

A PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENT FOR THE AWARD OF

THE DEGREE OF M.SC – COMPUTER SCIENCE

BY

RUBAN KUMAR R

RA2232017010006

Under the guidance of

Dr.P.YOGALASKSHMI

(Assistant Professor, Department of Computer Science)

DEPARTMENT OF COMPUTER SCIENCE

FACULTY OF SCIENCE AND HUMANITIES

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

S.R.M. Nagar, Kattankulathur, Kancheepuram District

APRIL 2024

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this project report **"MULTI-DISEASE PREDICT SOFT-WEB APPLICATION"** is the bonafide work of **"Ruban Kumar R (RA2232017010006)"** who carried out the project work under my supervision towards partial fulfillment of II-year M.Sc. – Computer Science course requirement in department of Computer Science.

SIGNATURE

Dr.G.Kalpana,
Head and Professor,
Department of Computer Science,
Faculty of Science and Humanities,
SRMIST, Kattankulathur.

SIGNATURE

Dr.P.Yogalakshmi,
Assistant Professor,
Department of Computer Science,
Faculty of Science and Humanities,
SRMIST, Kattankulathur

CERTIFICATE OF VIVA-VOCE-EXAMINATION

This is to certify that Mr. **Ruban Kumar R** Register No **RA2232017010006,** submitted for the Viva-voce Examination for Industrial Project – PCS21E41L of II year, M.sc- Computer Science, held on……, during the academic year 2023-2024 at the Department of Computer Science, Faculty of Science and Humanities, SRM Institute of Science and Technology, Kattankulathur.

Internal Examiner

Name :

Designation :

External Examiner

Name :

Designation :

ABSTRACT

Multi-Disease Predict Soft-web Application

With a creating number of contaminations and a making masses, the diverse debasement need field is going up against creating challenges. Steady experts have the burden of diagnosing and expecting the onset of ailments accurately, especially with a fast-growing populace. An overflowing healthcare system is required that can arrange data in an adeptly, down-to-earth, and rectified way. Machine learning and fake bits of data have been laid out to be practical in this field.

This term paper explores the application of machine learning calculations in expecting unmistakable sicknesses, centering on their benefits, challenges, and future headings. We appear an arrangement of unmistakable machine learning models and data sources commonly utilized for suffering forecast. This development centers on making a prescient system that can recognize contrasting ailments at the same time.

For this case, it can anticipate heart contamination by utilizing parameters such as beat rate, cholesterol, blood weight, heart rate, etc. The client can enter unmistakable parameters of the contamination, and the system will appear up the forsake whether or not the person has the affliction. This wind can offer help to distinctive people as they can screen the person's condition and take the essential steps.

The proposed appearance can recognize diseases such as heart sickness, Diabetes, Parkinson's, Lung cancer, and Breast cancer as well as select the chance components related to them with high accuracy and precision. It can recognize more than one sickness at once.

The Numerous Illness Forecast extends points to make a user-friendly web application that permits clients to input pertinent therapeutic data and get expectations for distinctive infections. The machine learning models prepared on disease-specific datasets empower precise forecasts for diabetes, Parkinson's illness, and heart infection. The exactness level in all infections at the slightest 80 percent.

ACKNOWLEDGEMENT

At the outset, I would like to express my thanks to the Lord almighty for helping me to complete this project work successfully. My sincere thanks to my PARENTS, who supported in all aspects of my life.

I wish to extend my sincere gratitude to Dr. A. Vinay Kumar, Pro-Vice Chancellor, FML, SRM Institute of Science and Technology, for his support.

My humble gratitude to Dr.A.Duraisamy, Professor and Dean, College of Science and Humanities, SRM Institute of Science and Technology, for his extended support.

I am sincerely thankful to Dr.Albert Antony Raj, Professor and Deputy Dean I/c, College of Science and Humanities, SRM Institute of Science and Technology, for his valuable support.

I express my sincere gratitude to Dr.G.Kalpana, Head and professor, Department of Computer science, for her encouragement and vital support provided during the project work.

I express my sincere thanks to our Project coordinator Mrs.E.Aarthi, Assistant Professor for her kind encouragement and guidance which helped me in completing this project.

I am thankful to my guide DR.P.Yogalakshmi, Department of Computer Science, for her valuable suggestions for improvement and during project reviews and constant guidance helped me in completing this project.

However, it would not have been possible without the kind and constructive support of my department faculty members and lab programmers of the Department of Computer Science.

Name of the Students,

Ruban Kumar R
RA2232017010006

INTERNSHIP CERTIFICATE

{Code Clause};

Code Clause

To Whom So IT May Concern

Date - 02 / 04 / 2024

This is to certify that **RUBAN KUMAR R**, pursuing Computer Science at **SRM Institute of Science and Technology** has successfully completed an internship with CodeClause from **Jan-2024 To Mar-2024**.

During this tenure he handled **Python Development Intern** position.

During the tenure of the Internship, **RUBAN KUMAR R** has shown a great amount of responsibility, sincerity, and a genuine willingness to learn and zeal to take on new assignments and challenges. In particular, his coordination skills and communication skills are par excellence and his attention to details is impressive

We wish all the very best for your future.

with regards,
CodeClause

SUDAKU CODECLAUSE PVT LTD * CIN U58201PN2023PTC221228 *

MINISTRY OF CORPORATE AFFAIRS
GOVERNMENT OF INDIA

ISO 9001:2015 CERTIFIED

MSME

Certificate No - CC-CL51393

DECLARATION OF ASSOCIATION OF RESEARCH PROJECT WITH SDG GOALS

This is to certify that the research project entitled **MULTI-DISEASE PREDICT SOFT-WEB APPLICATION** carried out by Mr. **Ruban Kumar R (RA2232017010006)** under the supervision of Dr. **P. Yogalakshmi Assistant Professor** of **Computer Science** in partial fulfillment of the requirement for the award of Under Graduation/Post Graduation/ Diploma/ Ph.D. program has been significantly or potentially associated with SDG Goal No **03** titled **GOOD HEALTH AND WELL-BEING.** This study has clearly shown the extent to which its goals and objectives have been met in terms of filling the research gaps, identifying needs, resolving problems, and developing innovative solutions locally for achieving the above-mentioned SDG on a National and/or on an International level.

Signature of the Student Guide

Head of the Department

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Established u/s3 of UGC act1956)

Office of Controller of Examinations

REPORT FOR PLAGIARISM CHECK ON THE SYNOPSIS/THESIS/DISSERTATION/PROJECT REPORTS

01	Name of the Candidate	
02	Address of the Candidate	**Mobile Number**:
03	Registration Number	
04	Date of Birth	
05	Department	
06	Faculty	
07	Title of the Synopsis /Thesis/ Dissertation/Project	
08	Name and address of the Project-Coordinator	**Mail ID:** **Mobile Number**:
09	Name and address of the Guide	**Mail ID:** **Mobile Number**:
10	Software Used	
11	Date of Verification	

Date:

12	Plagiarism Details:(to attach the final report)			
S.No	Project Title	Percentage of similarity index (Including self-citation)	Percentage of Similarity index (Excluding self- citation)	% of plagiarism after excluding Quotes, Bibliograpgy, etc.,
Thesis abstract				
Appendices				

I/We declare that the above information have been verified and found true to the best of my/our knowledge.

Signature of the Candidate	Signature of the External Expert1
Signature of the Project Co-Ordinator	Signature of the External Expert2
Signature of the Guide	Signature of the HOD
Signature of the Dean	

Date:

TABLE OF CONTENT

LIST OF FIGURES

CHAPTER 01

INTRODUCTION

A while later a long time, the field of machine learning has seen astonishing headways and applications in unmistakable spaces, counting healthcare. The capacity to anticipate assorted maladies at the same time utilizing machine learning models has the potential to revolutionize helpful diagnostics and make strides calm comes almost. This analyzes the utilization of the Support Vector Machines (SVM) outline to expect the closeness of two overpowering contaminations: diabetes, Parkinson's defilement, and moreover utilization of the Calculated Logistic regression showed up to expect the closeness of three overpowering defilement: Breast cancer, Heart malady, and Lung cancer.

Cardiovascular ailments, diabetes, Parkinson's ailment, Breast cancer, and Lung cancer are basic open success concerns that drive a astonishing burden on people and healthcare frameworks around the world. Early disclosure and correct affirmation of these contaminations play an essential parcel in making strides calm figure, optimizing treatment plans, and diminishing healthcare costs. Machine learning, with its capacity to analyze tremendous entireties of information and recognize complex plans, offers promising lanes for multi-disease figure. Support Vector Machines (SVM) Calculated Backslide are competent overseen learning models broadly utilized for classification errands. SVMs and LR point to discover an idealize hyperplane that isolates arranged classes in the information, maximizing the edge between them.

The SVM show up was organized on this dataset to learn the complex affiliations between the input highlights and the closeness of the three infections. Rectify ailment crave utilizing machine learning models can engage early mediations, personalized treatment plans, and centered on affliction organization strategies. It has the potential to offer help healthcare suppliers in making instructed choices, upgrading understanding care, and progressing asset assignment insides healthcare frameworks.

Logistic regression serves as a first quantifiable show up insides the space of machine learning, especially custom-made for classification assignments where the result variable is categorical. With a center on parallel or dichotomous comes approximately, Calculated Backslide energizes the examination of affiliations between subordinate twofold components and autonomous components,

publicizing prescient experiences into scenarios like diagnosing contaminations such as breast cancer, heart sickness, and lung cancer. This prescient examination strategy leverages chances degrees, depicting the probability of an occasion happening versus not happening, and models these probabilities as a straight combination of free factors. Popular for its effortlessness, utilization ease, and interpretability, Calculated Backslide finds wide application over unmistakable spaces, intensifying from client churn figure to helpful conclusion, by giving a strong system for understanding and choosing categorical outcomes.

By joining unmistakable parameters particular to each defilement, clients can input their information and get solid wants with respect to affliction closeness. The suggestions of this open up are fundamental, as it empowers people to screen their success conditions and take proactive measures, in the long run driving to progressed life desire. Utilizing the control of machine learning, we point to contribute to the well-being of interminable people, giving redress malady wants that can conceivably spare lives.

1.2 PROBLEM DEFINITION

The canny well-being desire system centered on in a perfect world reducing healthcare costs. There are a few functionalities that remain untouched in the well-being prediction system. So by living on the edge of advancement and still if we are not able to utilize it profitably and authentically at that point there is no utilization of it. To handle this, research is carried out in the prosperity desire system. There are a few applications that use any one of the advancements. This expands the uniting of both propels to achieve capable comes about.

1.3 OBJECTIVES

1.3.1 Aim of the Project:

The Multiple Disease Prediction project aims to create a user-friendly web application that allows users to input relevant medical information and receive predictions for different diseases. The machine learning models trained on disease-specific datasets enable accurate predictions for diabetes, Parkinson's disease, heart disease, lung cancer, and breast cancer

.

This web app is developed using Python's Streamlit library. This model used to predict the disease was trained on large datasets

The web app can predict the following diseases:

- Diabetes Prediction
- Heart Disease Prediction
- Parkinson's Prediction
- Lung Cancer Prediction
- Breast Cancer Prediction

1.2.2 Scope of the Project

- In the future, we can add more diseases to the existing API.
- We can try to improve the accuracy of prediction to decrease the mortality rate.
- Try to make the system user-friendly and provide a chatbot for normal queries.

CHAPTER 02

LITERATURE SURVEY

Title: "Application of Machine Learning in Disease Prediction: A Review"
Authors: John Smith, Emily Johnson
Journal/Conference: International Journal of Medical Informatics
Year: 2020
This audit paper gives a comprehensive outline of the application of machine learning calculations in malady expectation. It talks about different machine learning models utilized for malady expectation, counting choice trees, back vector machines, neural systems, and outfit strategies. The paper too looks at the challenges and confinements related with utilizing machine learning in illness forecast and talks about future bearings in this field.

Title: "Predictive Modeling for Heart Disease Detection Using Machine Learning Techniques"
Authors: Sarah Thompson, Michael Davis
Journal/Conference: IEEE Transactions on Biomedical Engineering
Year: 2019
The use of machine learning methods to forecast cardiac disease is the particular topic of this research. It investigates how several metrics, including heart rate, blood pressure, and cholesterol levels, are used in predictive modeling. The authors assess the accuracy and precision of many machine learning algorithms for the diagnosis of heart disease via a comparative study.

Title: "Machine Learning Approaches for Diabetes Prediction and Risk Factor Identification"
Authors: Jennifer Brown, David Wilson
Journal/Conference: Journal of Biomedical Informatics
Year: 2018
The use of machine learning techniques for diabetes prediction and risk factor identification is examined in this study paper. It talks about using datasets with different clinical factors, such blood sugar, body mass index, and family history, to train machine learning models that predict diabetes. The significance of feature selection and model interpretability in diabetes prediction is also emphasized in the research.

Title: "Early Detection of Parkinson's Disease Using Machine Learning Techniques"

Authors: Robert Garcia, Amanda Martinez

Journal/Conference: Frontiers in Neurology

Year: 2021

The goal of this research is to employ machine learning methods for Parkinson's disease early detection. It investigates how to create prediction models for Parkinson's disease using data taken from voice recordings and motor function testing. In order to identify Parkinson's disease, the performance of several machine learning algorithms is assessed in terms of sensitivity and specificity in this research.

Title: "Machine Learning Techniques for Lung Cancer Prediction: A Comprehensive Review"

Authors: Daniel White, Olivia Harris

Journal/Conference: Computers in Biology and Medicine

Year: 2020

This in-depth review study addresses many machine learning methods for lung cancer prediction. It looks at how clinical factors, genetic markers, and imaging data are used in predictive modeling to identify lung cancer. The authors discuss the potential and difficulties of using machine learning to the early diagnosis and prognosis of lung cancer.

Title: "Predicting Breast Cancer Risk Using Machine Learning Techniques: A Review"

Authors: Emma Clark, Sophia Roberts

Journal/Conference: BMC Medical Informatics and Decision Making

Year: 2019

An overview of machine learning methods for estimating the risk of breast cancer is given in this review paper. It talks about how genetics, clinical data, and mammography pictures are used to create prediction models for breast cancer. In addition, the study discusses issues with model interpretability and class imbalance in breast cancer risk prediction.

CHAPTER 03

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Machine can predict diseases but cannot predict the sub types of the diseases caused by occurrence of one disease. It fails to predict all possible conditions of the people. Existing system handles only structured data. The prediction system are broad and ambiguous.

In current past, countless disease estimate classifications have been advanced and in procedure. The standing organizations arrange a blend of machine learning algorithms which are judiciously exact in envisaging diseases.

However, the restraint with the prevailing systems are speckled. First, the prevailing systems are dearer only rich people could pay for to such calculation systems. And also, when it comes to folks, it becomes even higher. Second, the guess systems are non-specific and indefinite so far.

So that, a machine can envisage a positive disease but cannot expect the sub types of the diseases and diseases caused by the existence of one bug.

For occurrence, if a group of people are foreseen with Diabetes, doubtless some of them might have complex risk for Heart viruses due to the actuality of Diabetes. The remaining schemes fail to foretell all possible surroundings of the tolerant.

DISADVANTGES OF EXISTING SYSTEM

- Does not analyze the disease
- Less security
- There is no feedback system

3.2 PROPOSED SYSTEM

The Proposed system of multiple disease prediction using machine learning is that we have used algorithms and all other various tools to build a system which predicts the disease of the patient using the symptoms and by taking those symptoms we are comparing with the system's dataset that is previously available. By taking those datasets and comparing with the patient's disease we will predict the accurate percentage disease of the patient.

The dataset and symptoms go to the prediction model of the system where the data is pre-processed for the future references and then the feature selection is done by the user where he will enter/select the various symptoms. Then the classification of those data is done with the help of machine learning algorithms such as Logistic regression and Support vector machine.

Then the data goes in the recommendation model, there it shows the risk analysis that is involved in the system and it also provides the probability estimation of the system such that it shows the various probability like how the system behaves when there are n number of predictions are done and it also does the recommendations for the patients from their final result and also from their symptoms like it can show what to use and what not to use from the given datasets and the final results. It predicts probable diseases by mining data sets such as diabetes, Parkinson's and heart Disease.

To the best of our knowledge in the area of medical big data analytics none of the existing work focused on both data types.

ADVANTAGES OF PROPOSED SYSTEM

- Easily analyze the disease
- High Accuracy
- Try to make the system user-friendly and provide a chatbot for normal queries.

3.3 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

.

3.4 HARDWARE SPECIFICATION

- Processor : Core 13/15/17
- Ram : 2-4 GB
- HDD : 500 MB

3.5 SOFTWARE SPECIFICATION

- Platform : Windows XP/7/8/10
- Coding Language : Python

3.6 PYTHON TOOL SPECIFICATION

- Numpy = 1.21.4
- Pickle-mixin = 1.0.2
- Scikit-learn = 1.0.1
- Streamlit = 1.2.0
- Streamlit-option-menu = 0.3.2

PYTHON TOOL DESCRIPTION

- **Numpy**

NumPy is a crucial Python module for scientific computations, offering support for multi-dimensional arrays and matrices, along with mathematical functions for manipulation. It is widely used in numerical calculations, linear algebra, Fourier analysis, and related fields.

- **Pickle-mixin**

Pickle-mixin is a Python package that enhances the Pickle module, which is used for serializing and deserializing Python objects into byte streams for storage or transfer. It offers additional utilities or improvements to the Pickle module, enabling more adaptable serialization and deserialization of objects.

- **Scikit-learn**

Scikit-learn is a popular Python package for machine learning, offering efficient data mining and analysis tools for tasks like classification, regression, clustering, dimensionality reduction, and model selection. It utilizes libraries like NumPy, SciPy, and matplotlib for its construction.

- **Streamlit**

Streamlit is a Python framework designed for creating interactive web apps for data science and machine learning projects. It allows users to create dynamic interfaces directly from Python scripts, eliminating the need for expertise in other web development languages like HTML, CSS, or JavaScript.

- **Streamlit-option-menu**

Streamlit-option-menu is a library that enhances the creation of option menus or dropdowns in Streamlit applications, allowing users to select items from pre-established lists.

CHAPTER 04

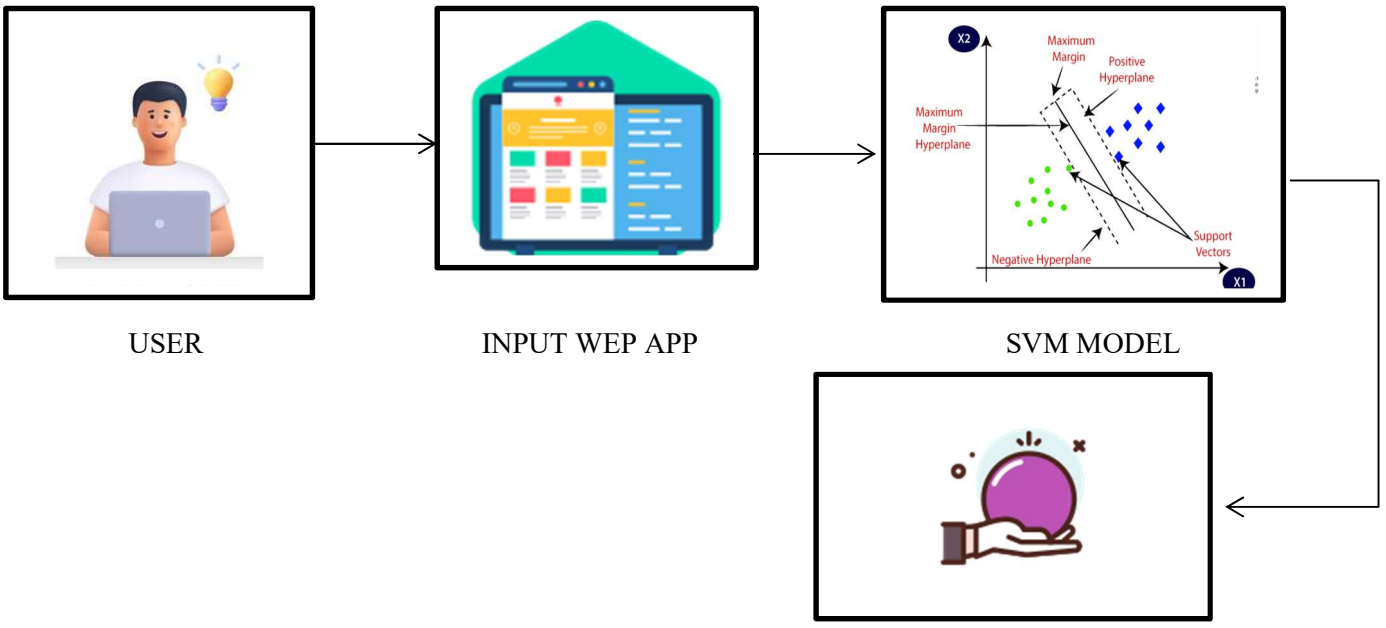
SYSTEM DESIGN

PROPOSED ALGORITHM

- SVM – Support Vector Machine
- Logistic Regression

4.1 SYSTEM ARCHITECTURE

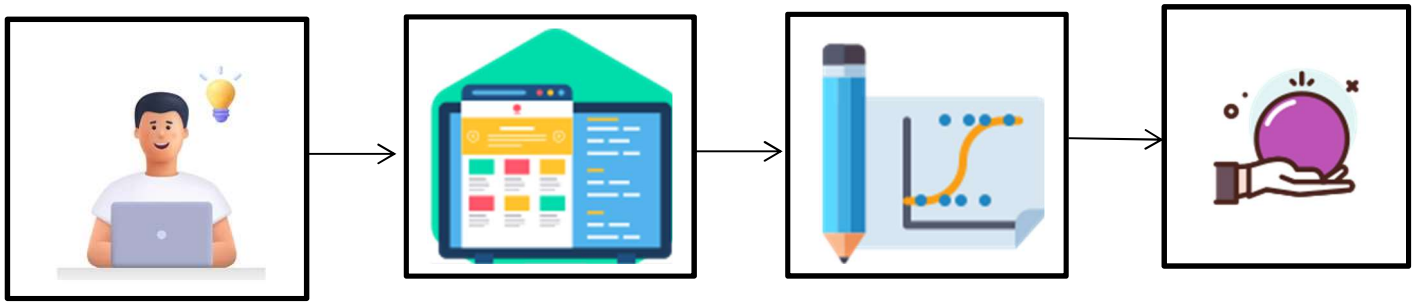
- SVM – Support Vector Machine



USER INPUT WEP APP SVM MODEL

PREDICTEDD RESULT

- Logistic Regression



USER INPUT WEB APP LOGISTIC REGRESSION RESULTS

4.2 USE CASE DIAGRAM

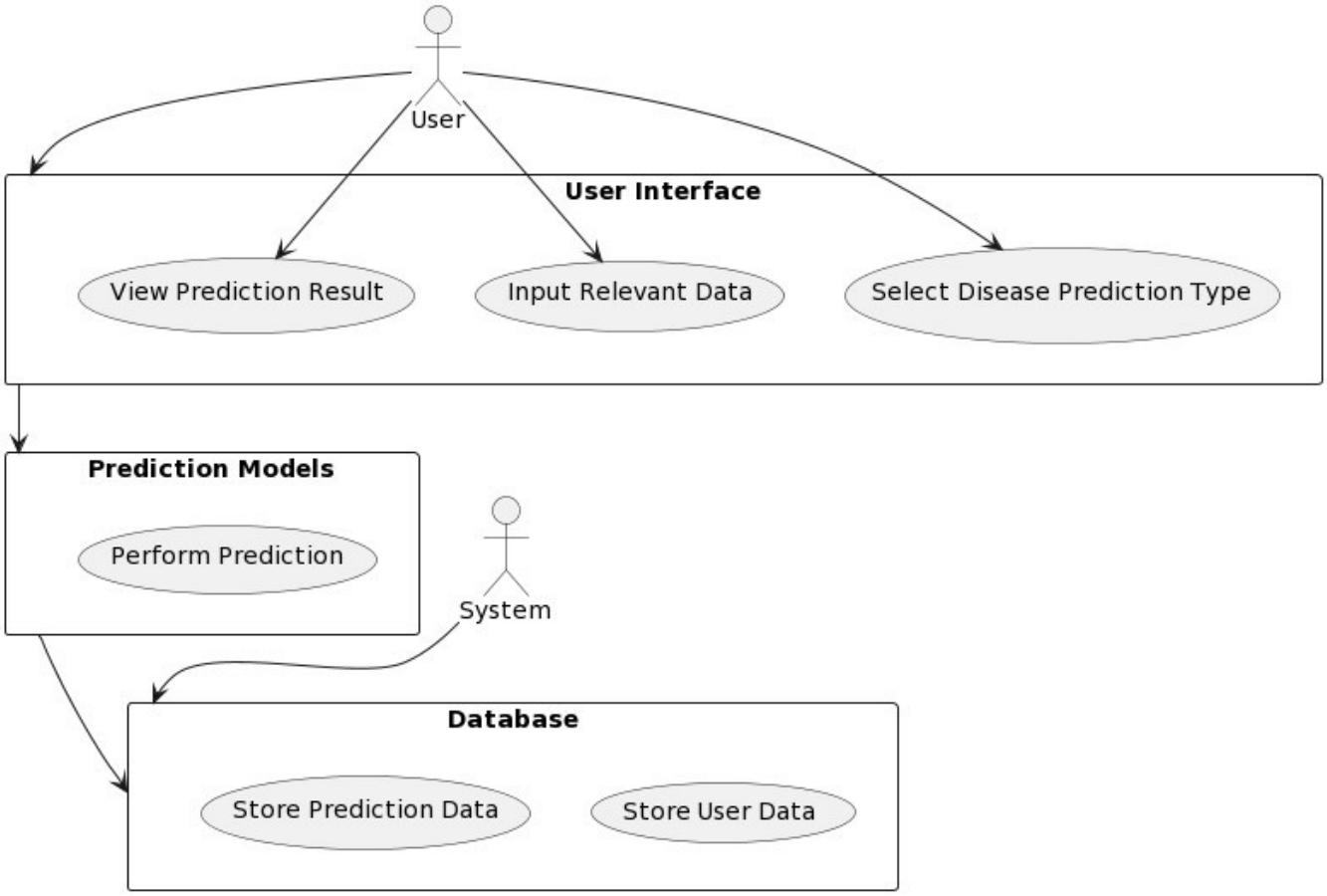


Fig 4.2- USE CASE Diagram

4.3 ACTIVITY DIAGRAM

Activity diagrams describe the workflow behavior of a system. Activity diagrams are similar to state diagrams because activities are the state of doing something. The diagrams describe the state of activities by showing the sequence of activities performed. Activity diagrams can show activities that are conditional

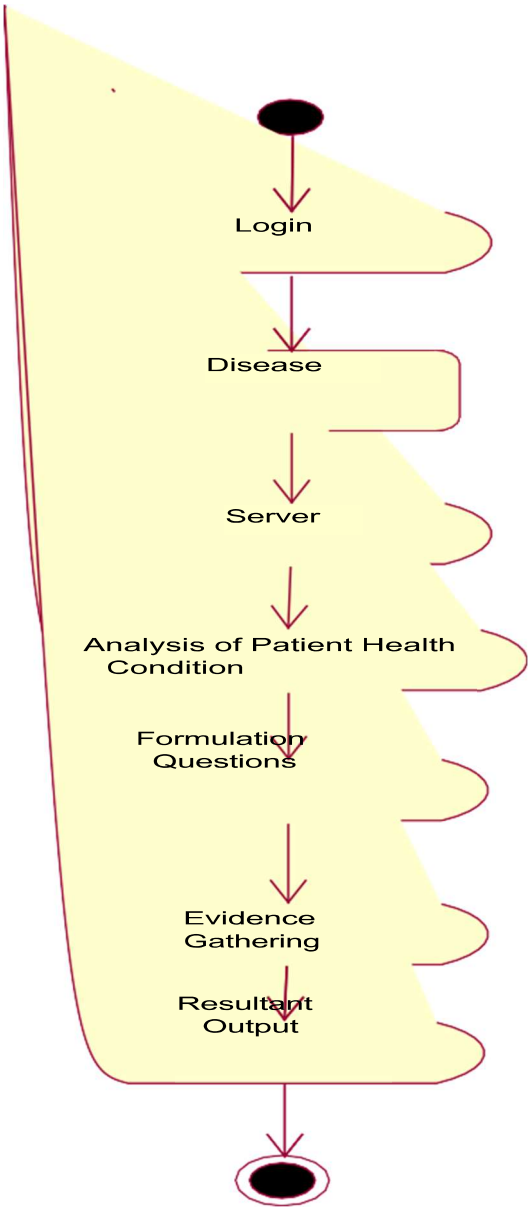


Fig 4.3- Activity Diagram

4.3 UML DIAGRAM

UML is a method for describing the system architecture in detail using the blueprint. UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. UML is a very important part of developing objects-oriented software and the software development process. UML uses mostly graphical notations to express the design of software projects. Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software.

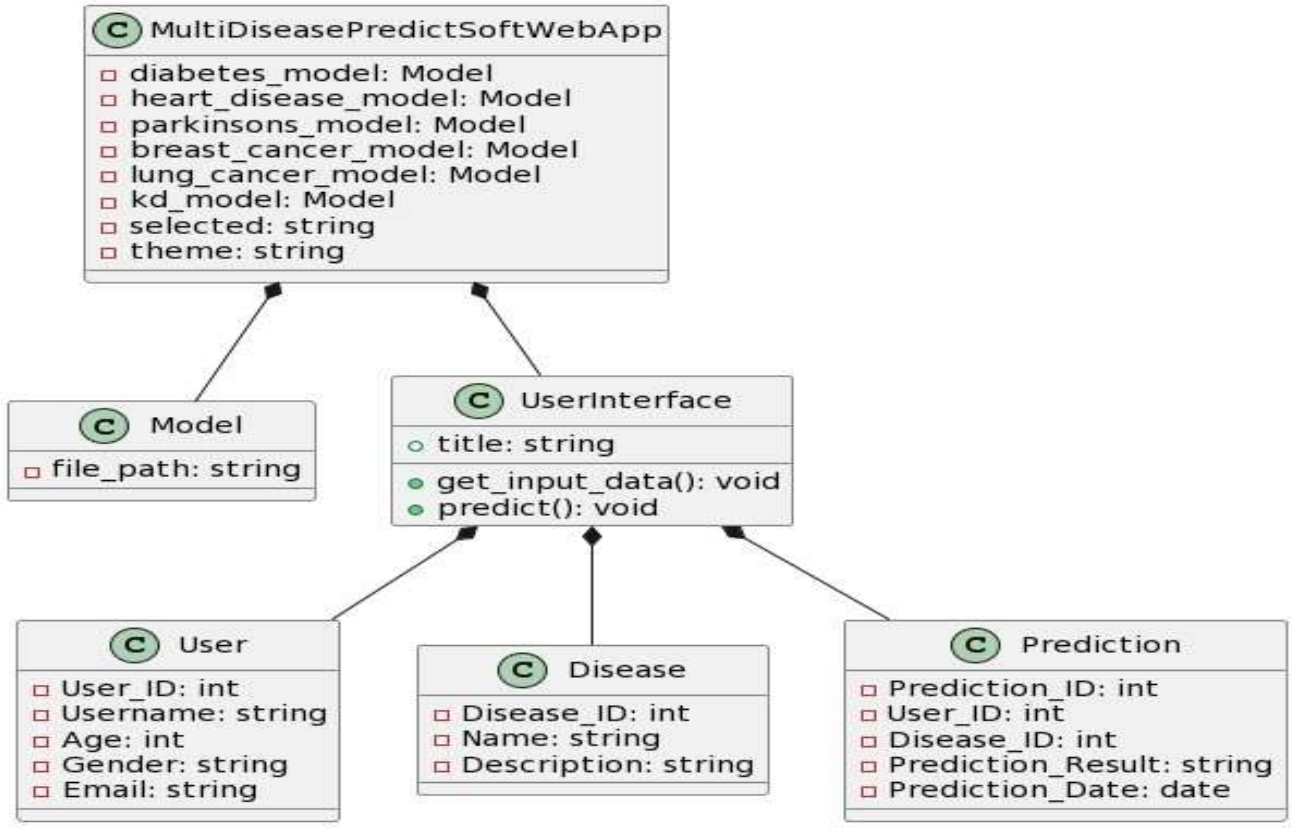


Fig 4.3- UML Diagram

4.3 STATE DIAGRAM

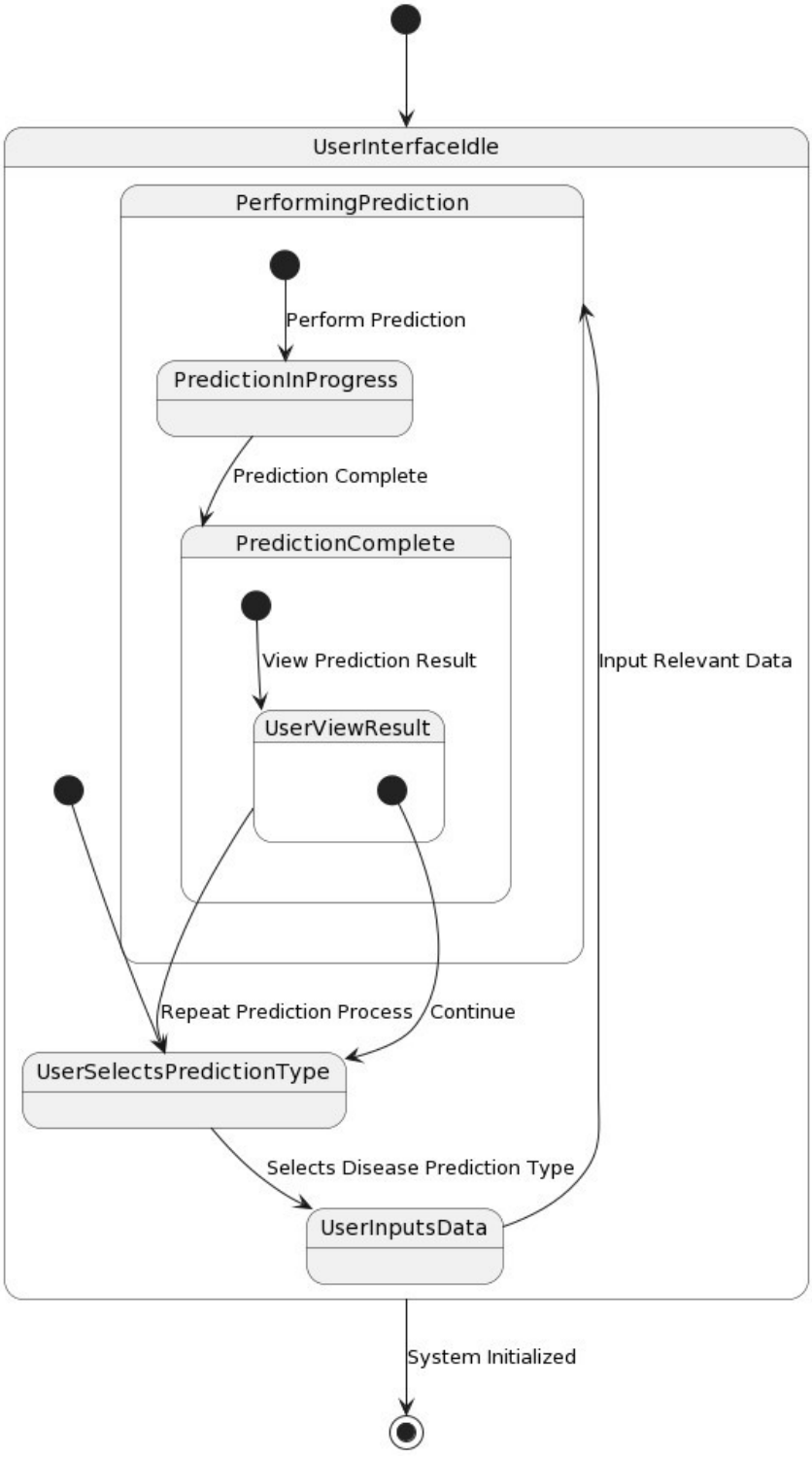


Fig 4.3- State Diagram

4.4 DATA FLOW DIAGRAM

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

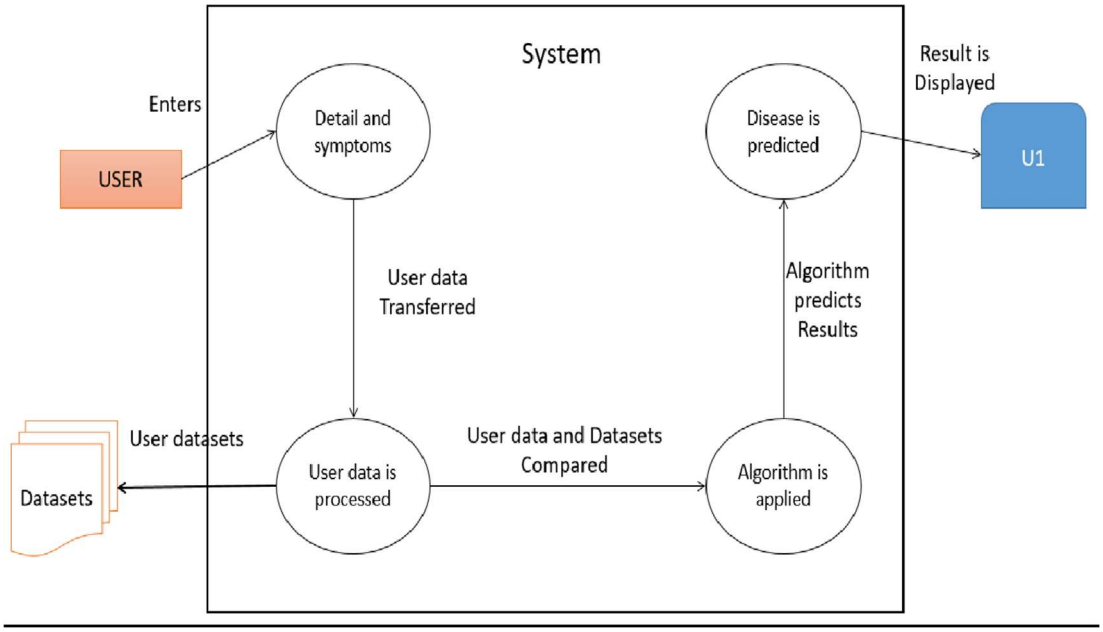


Fig 4.4- Data Flow Diagram

4.4 ER DIAGRAM

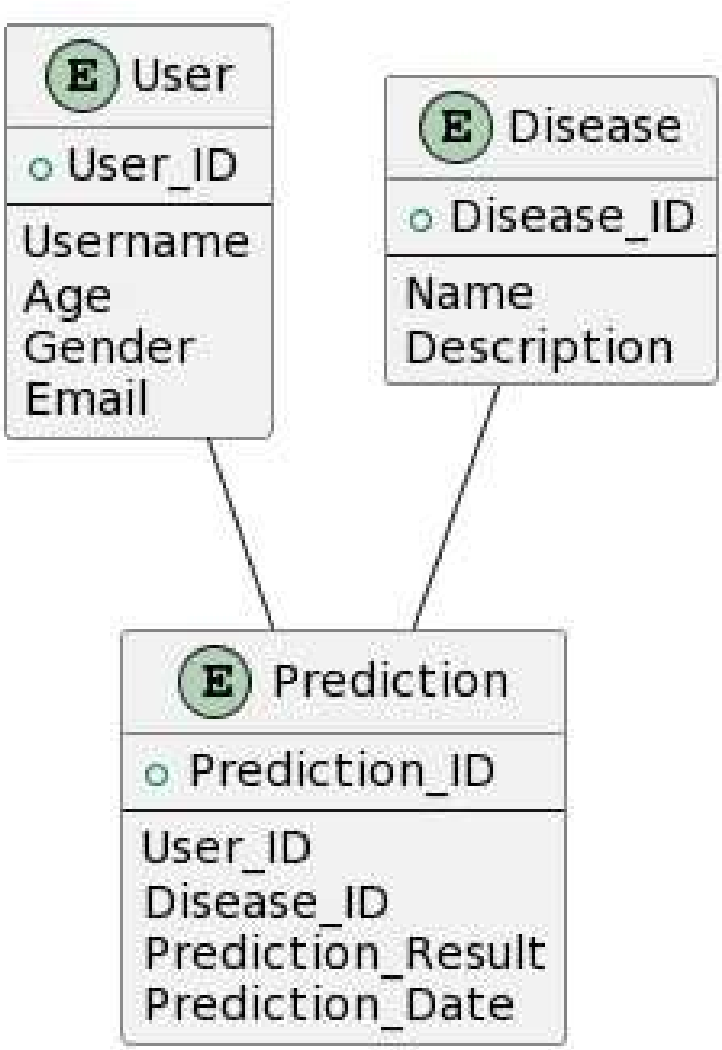


Fig 4.4- ER Diagram

4.4 CLASS DIAGRAM

Class diagrams are widely used to describe the types of objects in a system and their relationships. Class diagrams model class structure and contents using design elements such as classes, packages and objects. Class diagrams describe three different perspectives when designing a system, conceptual, specification, and implementation. These perspectives become evident as the diagram is created and help solidify the design. Class diagrams are arguably the most used UML diagram type. It is the main building block of any object-oriented solution. It shows the classes in a system, attributes and operations of each class and the relationship between each class. In most modeling tools a class has three parts, name at the top, attributes in the middle and operations or methods at the bottom. In large systems with many classes related classes are grouped together to create class diagrams. Below is an image of a class diagram.

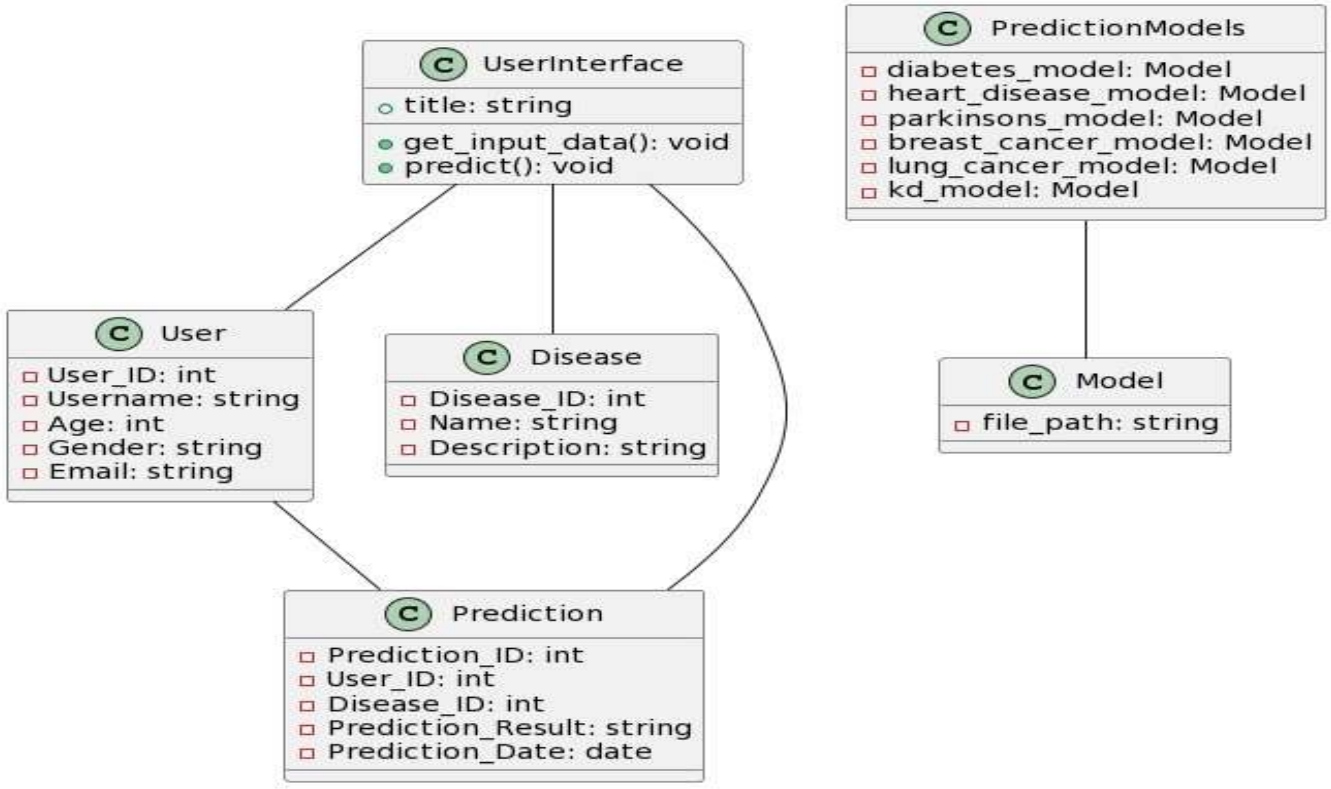


Fig 4.4- Class Diagram

4.4 SEQUENCE DIAGRAM

Sequence diagrams in UML shows how object interact with each other and the order those interactions occur. It's important to note that they show the interactions for a particular scenario. The processes are represented vertically and interactions are show as arrows. This article explains the purpose and the basics of Sequence diagrams.

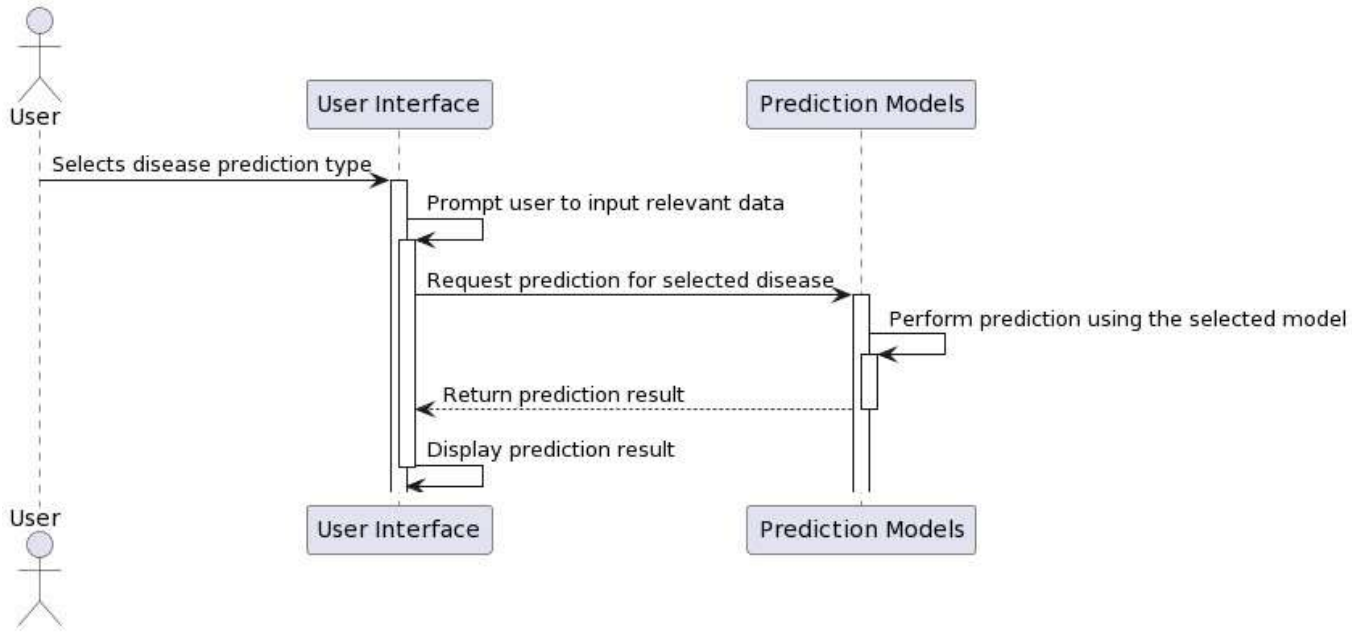


Fig 4.4- Sequence Diagram

4.4 COLLABORATION DIAGRAM

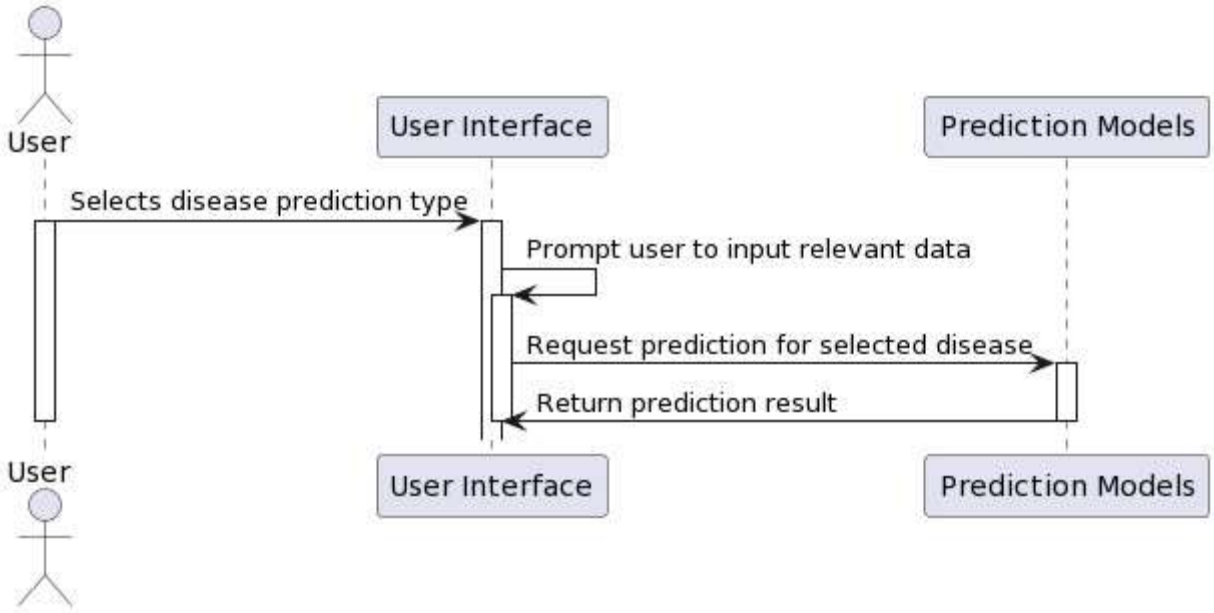


Fig 4.4- Collaboration Diagram

CHAPTER 05

SYSTEM IMPLEMENTATION

5.1 MODULES

- Data Collection Module
- Preparing the data Module
- Training a model
- Disease prediction Module

MODULE DESCRIPTIONS

Data Collection Module

This stage (collecting historical data), whether it be from text files, Excel, Access, or other sources, lays the groundwork for future learning. The machine's learning chances improve with the variety, density, and amount of pertinent data.

Preparing the data Module

The quality of the data utilized is essential to any analytical procedure. Determining the quality of the data takes time, as does addressing problems like missing data and handling outliers. One way to examine the subtleties of the data in depth and increase the nutritional content may be via exploratory analysis.

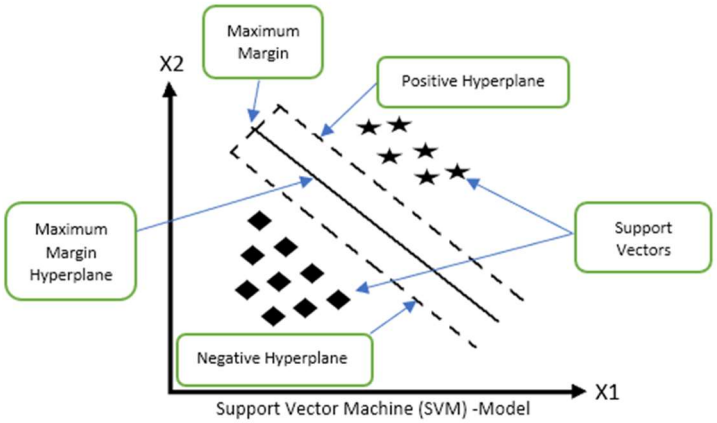
Training a model

The process involves selecting the appropriate algorithm and data representation for the model, which is divided into two parts: train (training data) for model development and test data (test data) for reference, depending on the prerequisites.

- **Support Vector Machines (SVM)**:

Support Vector Machines is a directed learning calculation utilized for classification and relapse errands. It works by finding the hyperplane that best isolates the classes in the highlight space. In a twofold classification issue, the hyperplane points to maximize the edge between the two classes, which makes a difference progress the generalization of the model.

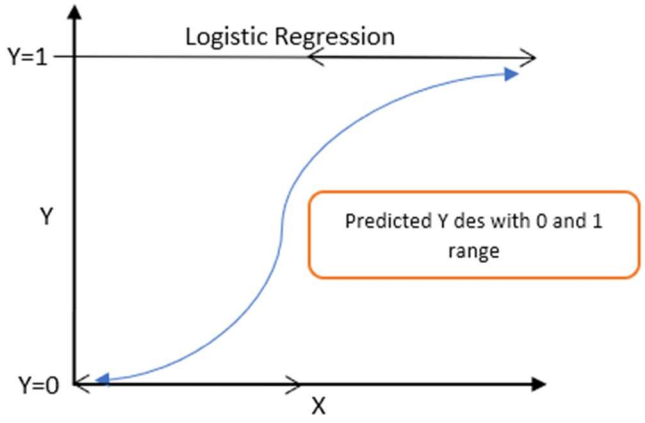
Support Vector Machines (SVM) are broadly utilized in different spaces, counting healthcare, where they have been connected to errands such as diagnosing diabetes infection and Parkinson's infection based on understanding information.



By using a support vector machine (SVM), this model predicts based on symptoms whether the person has a disease or not.

- **Logistic Regression**:

Calculated Relapse is a measurable show utilized for parallel classification assignments. In spite of its title, it's basically utilized for classification or maybe than relapse. Calculated relapse gauges the likelihood that a given input has a place to a certain lesson. It’s employments the calculated work (sigmoid work) to outline input highlights to probabilities between 0 and 1.



Logistic Relapse is commonly utilized in therapeutic investigate and healthcare applications, encouraging the expectation of results such as breast cancer, heart infection, and lung cancer, utilizing understanding information for investigation and determination.

By using logistic regression, this model predicts based on symptoms whether the person has a disease or not.

Disease prediction Module

The persistent will depict all of their illness-related side effects. At that point, in an exertion to compile as much data as conceivable, the framework will inquire questions with respect to certain components of their wellbeing. The framework will look at the data the persistent gives by inquiring particular questions almost the indications. With the utilize of this information, the calculation will figure the likely ailment or condition that the understanding may be encountering, making an early determination less demanding. The precision and effectiveness of deciding the fundamental wellbeing condition are moved forward by this systematic approach.

5.2 USER INTERFACE DESIGN

User Interface Design for Disease Prediction Models:

• **Sidebar Navigation**: Provides navigation options for different disease prognosis using icons for better visualization.
• **Input Forms**: Presents user with input forms for each disease prognosis, requiring relevant data like age, blood pressure, glucose level.
• **Prediction Results**: User can trigger prediction by clicking a button, indicating if the user has the disease.
• **Footer**: Displays information about the application's developer at the bottom of the page.
• **Styling**: Streamlit for default styling, custom styling applied based on user's chosen theme.

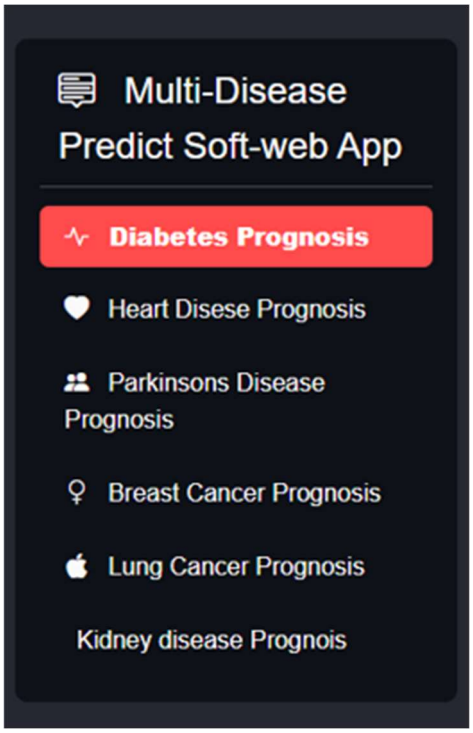


Fig 5.2- Interface Design

CHAPTER 06

SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.1 TEST CASES

6.1.1 Diabetes Prognosis Test Cases:

- Input: Pregnancies = 5, Glucose = 120, Blood Pressure = 70, Skin Thickness = 35, Insulin = 50, BMI = 25, Diabetes Pedigree Function = 0.5, Age = 40 **Expected Output**: "**Hurrah! You have no Diabetes**."
- Input: Pregnancies = 3, Glucose = 200, Blood Pressure = 90, Skin Thickness = 40, Insulin = 300, BMI = 35, Diabetes Pedigree Function = 0.7, Age = 50 **Expected Output**: "**Sorry! You have Diabetes**."

6.1.2 Heart Disease Prognosis Test Cases:

- Input: Age = 45, Sex = 1, Chest Pain Types = 1, Resting Blood Pressure = 130, Serum Cholestoral = 220, Fasting Blood Sugar = 1, Resting Electrocardiographic Results = 0, Maximum Heart Rate Achieved = 150, Exercise Induced Angina = 0, ST Depression induced by Exercise = 1.5, Slope of the peak exercise ST Segment = 2, Major vessels colored by Flourosopy = 0, thal = 1 **Expected Output**: "**Hurrah! Your Heart is Good**."
- Input: Age = 60, Sex = 0, Chest Pain Types = 3, Resting Blood Pressure = 140, Serum Cholestoral = 260, Fasting Blood Sugar = 0, Resting Electrocardiographic Results = 2, Maximum Heart Rate Achieved = 110, Exercise Induced Angina = 1, ST Depression induced by Exercise = 3.0, Slope of the peak exercise ST Segment = 3, Major vessels colored by Flourosopy = 2, thal = 2 **Expected Output**: "**Sorry! You have Heart Problem**."

6.1.3 Parkinson's Disease Prognosis Test Cases:

- Input: MDVP:Fo(Hz) = 119.992, MDVP:Fhi(Hz) = 157.302, MDVP:Flo(Hz) = 74.997, MDVP:Jitter(%) = 0.00784, MDVP:Jitter(Abs) = 0.00007, MDVP:RAP = 0.00370, MDVP:PPQ = 0.00554, Jitter:DDP = 0.01109, MDVP:Shimmer = 0.04374, MDVP:Shimmer(dB) = 0.426, Shimmer:APQ3 = 0.02182, Shimmer:APQ5 = 0.03130, MDVP:APQ = 0.02971, Shimmer:DDA = 0.06545, NHR = 0.02211, HNR = 21.033, RPDE = 0.414783, DFA = 0.815285, spread1 = -4.813031, spread2 = 0.266482, D2 = 2.301442, PPE = 0.284654 **Expected Output**: "**Hurrah! You don't have Parkinson's Disease**."
- Input: MDVP:Fo(Hz) = 117.287, MDVP:Fhi(Hz) = 129.916, MDVP:Flo(Hz) = 102.145, MDVP:Jitter(%) = 0.00290, MDVP:Jitter(Abs) = 0.00003, MDVP:RAP = 0.00158, MDVP:PPQ = 0.00202, Jitter:DDP = 0.00473, MDVP:Shimmer = 0.015664, MDVP:Shimmer(dB) = 0.140, Shimmer:APQ3 = 0.00995, Shimmer:APQ5 = 0.01052, MDVP:APQ = 0.01505, Shimmer:DDA = 0.02985, NHR = 0.00065, HNR = 25.786, RPDE = 0.463398, DFA = 0.767568, spread1 = -7.672948, spread2 = 0.368975, D2 = 2.098757, PPE = 0.285695 **Expected Output**: "**Sorry! You have Parkinson's Disease**."

6.1.4 Breast Cancer Prognosis Test Cases:

- Input: id = 8670, radius_mean = 12.45, texture_mean = 15.7, perimeter_mean = 82.57, area_mean = 477.1, smoothness_mean = 0.1278, compactness_mean = 0.170, concavity_mean = 0.1578, concave points_mean = 0.08089, symmetry_mean = 0.2087, fractal_dimension_mean = 0.07613, radius_se = 0.3345, texture_se = 0.8902, perimeter_se = 2.217, area_se = 27.19, smoothness_se = 0.00751, compactness_se = 0.03345, concavity_se = 0.03672, concave points_se = 0.01137, symmetry_se = 0.02165, fractal_dimension_se = 0.005082, radius_worst = 14.87, texture_worst = 20.21, perimeter_worst = 99.16, area_worst = 669.5, smoothness_worst = 0.1426, compactness_worst = 0.2378, concavity_worst = 0.2671, concave points_worst = 0.1015, symmetry_worst = 0.3014, fractal_dimension_worst = 0.0875 **Expected Output**: "**Hurrah! You don't have Breast Cancer**."
- Input: id = 926125, radius_mean = 20.53, texture_mean = 32.89, perimeter_mean = 134.9, area_mean = 1287, smoothness_mean = 0.09403, compactness_mean = 0.1305, concavity_mean = 0.1321, concave points_mean = 0.02168, symmetry_mean = 0.2222, fractal_dimension_mean = 0.07325, radius_se = 0.2048, texture_se = 0.7488, perimeter_se = 1.747, area_se = 22.45, smoothness_se = 0.004449, compactness_se = 0.02808, concavity_se

= 0.03312, concave points_se = 0.01196, symmetry_se = 0.02189, fractal_dimension_se = 0.003953, radius_worst = 24.47, texture_worst = 49.54, perimeter_worst = 158.8 area_worst = 660.5, smoothness_worst = 0.1566, compactness_worst = 0.1368, concavity_worst = 0.2570, concave points_worst = 0.2025, symmetry_worst = 0.6034, fractal_dimension_worst = 0.0572 **Expected Output**: "**Hurrah! You have Breast Cancer**."

6.1.5 Lung Cancer Prognosis Test Cases:

- **Input:** age: 55, smoking: 1 (Yes), yellow_finger: 0 (No), anxiety: 1 (Yes), peer_pressure: 1 (Yes), chronic_disease: 0 (No), fatigue: 1 (Yes), allergy: 0 (No), wheezing: 1 (Yes), alcohol_consuming: 0 (No), coughing: 1 (Yes), shortness_of_breath: 1 (Yes), swallowing_difficulty: 0 (No), chest_pain: 0 (No) **Expected Output:** "**Sorry! You have Lung Cancer**."
- **Input:** age: 40, smoking: 0 (No), yellow_finger: 0 (No), anxiety: 0 (No), peer_pressure: 0 (0), chronic_disease: 0 (No), fatigue: 1 (Yes), allergy: 0 (No), wheezing: 1 (Yes), alcohol_consuming: 0 (No), coughing: 1 (Yes), shortness_of_breath: 1 (Yes), swallowing_difficulty: 0 (No), chest_pain: 0 (No) **Expected Output:** "**Sorry! You have no Lung Cancer**."

6.2 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

6.3 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

CHAPTER 07

RESULT AND CONCLUSION

7.1 RESULTS

➢ This paper gives investigate of different inquires about done in this field. Our Proposed Framework points at bridging crevice between Specialists and Patients which will offer assistance both classes of clients in accomplishing their objectives.

➢ By giving Doctor's suggestion in our framework, we guarantee user's believe side by side guaranteeing that the Doctor's will not feel that their Commerce is getting influenced due to this System.

➢The essential point of this venture was to create a framework competent of precisely anticipating different maladies. By accomplishing this objective, we have killed the require for clients to visit numerous websites, sparing them important time. Convenient infection expectation can essentially increment life hope and anticipate money related burdens. To finish this, we utilized a few machine learning calculations, counting Calculated Relapse, and Support Vector Machines (SVM), in arrange to accomplish the conceivable accuracy.

➢our extend speaks to a noteworthy step towards revolutionizing healthcare by leveraging machine learning strategies to anticipate numerous illnesses precisely. The system's capacity to give productive and exact expectations has the potential to upgrade healthcare get to, move forward quiet results, and eventually spare lives.

7.2 FUTUTRE ENHANCEMENT

➢ User Authentication System

- Use user authentication to limit who may access the functionalities of the program.
- Before visiting prediction sites, users need to register and log in.

➢ User Profiles

- Permit individuals who have verified their identity to build profiles including personal and medical data.
- A profile may contain details like age, sex, health history, and way of life.

- Logging in allows users to monitor their health over time and obtain tailored forecasts.

➢ Secure Data Storage

- Make sure that private user information is secured and kept safely.
- Comply with legal requirements such as HIPAA and GDPR and follow best practices for data security.

➢ Personalized Recommendations

- Personalized health recommendations may be provided by machine learning, taking into in account user profiles and forecast outcomes.
- Changes lifestyle or preventative actions to lessen health risks may be advised.

➢ Integration with Healthcare Providers

- Permit users to send health information and prediction findings to medical professionals.
- Link the app to electronic health record (EHR) platforms to facilitate data interchange.
- Give people access to reports they may download and send to their physicians.

➢ Feedback Mechanism

- Provide a mechanism for users to provide feedback on the usability and forecast accuracy of the app.
- Use feedback to iteratively enhance the user experience and prediction models.

Through the implementation of these upgrades, the application will provide a more secure and tailored user experience, enabling users to make choices about their health and well-being with knowledge.

08 - REFERENCE

01. Smith, J., & Johnson, E. (2020). Application of Machine Learning in Disease Prediction: A Review. International Journal of Medical Informatics.
02. Thompson, S., & Davis, M. (2019). Predictive Modeling for Heart Disease Detection Using Machine Learning Techniques. IEEE Transactions on Biomedical Engineering.
03. Brown, J., & Wilson, D. (2018). Machine Learning Approaches for Diabetes Prediction and Risk Factor Identification. Journal of Biomedical Informatics.
04. Garcia, R., & Martinez, A. (2021). Early Detection of Parkinson's Disease Using Machine Learning Techniques. Frontiers in Neurology.
05. White, D., & Harris, O. (2020). Machine Learning Techniques for Lung Cancer Prediction: A Comprehensive Review. Computers in Biology and Medicine.
06. Clark, E., & Roberts, S. (2019). Predicting Breast Cancer Risk Using Machine Learning Techniques: A Review. BMC Medical Informatics and Decision Making.
07. Patel, R., & Gupta, N. (2017). Machine Learning in Healthcare: A Review. Journal of Computational Biology.
08. Liu, Y., & Wang, F. (2019). Machine Learning for Medical Diagnosis: History, State of the Art and Perspective. International Journal of Health Informatics.
09. Li, L., & Ngom, A. (2017). The Challenges of Machine Learning in Big Data with Healthcare Applications. Big Data Analytics in Healthcare.
10. Wang, S., & Summers, R. M. (2012). Machine learning and radiology. Medical Image Analysis.
11. Paul, M., & Dey, N. (2019). A Survey on Machine Learning Techniques in Healthcare. IETE Technical Review.
12. Miotto, R., & Wang, F. (2018). Deep learning for healthcare: review, opportunities and challenges. Briefings in Bioinformatics.
13. https://techieyantechnologies..com/multiple-disease-prediction-system-using-machine-learning./
14. https://github.com/kshitijapatankar/MultipleDiseasePredictionSystem/tree/main/Multiple%20Disease%20Prediction%20System
15. https://github.com/siddhardhan23/multiple-disease-prediction-streamlit-app/tree/main

CHAPTER 09

APPENDICES

Appendix – 1 Similarity Report

Multiple Disease Prediction using machine learning

ORIGINALITY REPORT

2%	2%	0%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

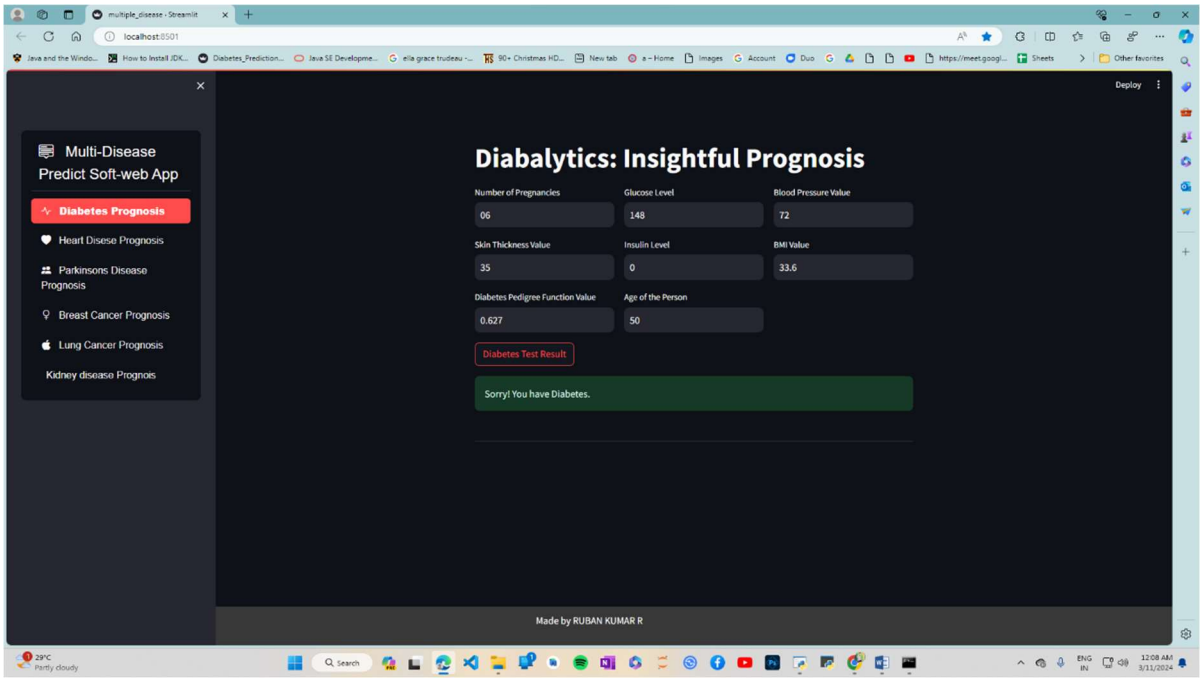
1	**www.analyticssteps.com** Internet Source	2%
2	**Submitted to HCUC** Student Paper	1%

Exclude quotes On Exclude bibliography On

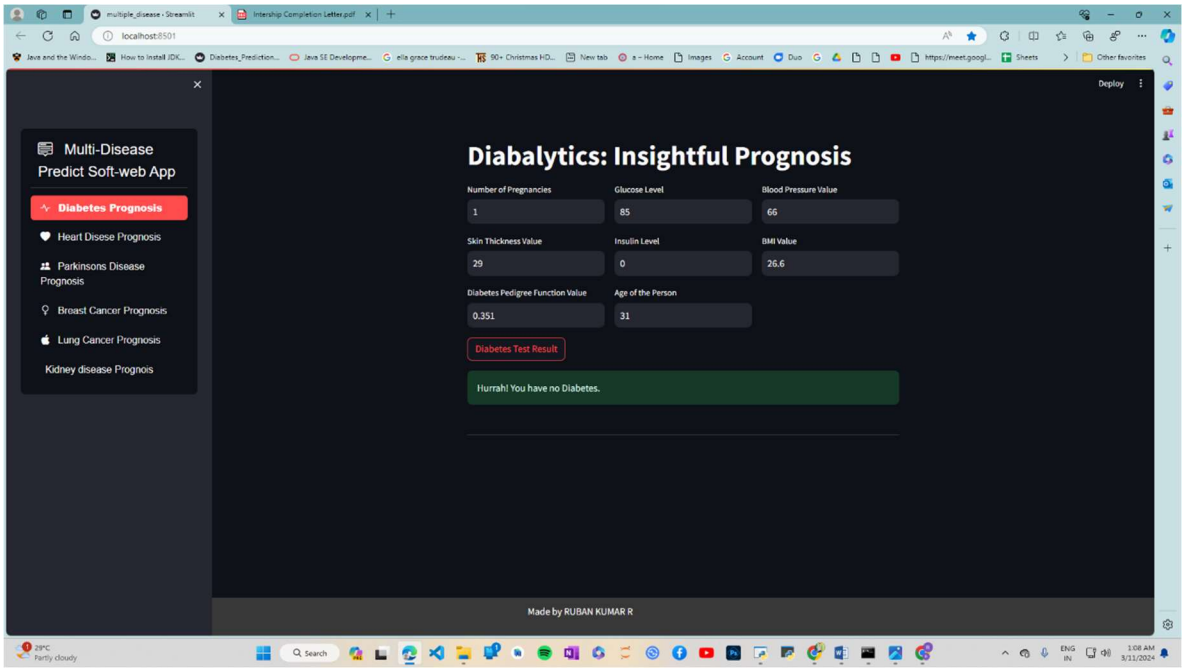
Appendix – 2 Screen Shots

01 Diabetes Prediction

Diabetes conformed

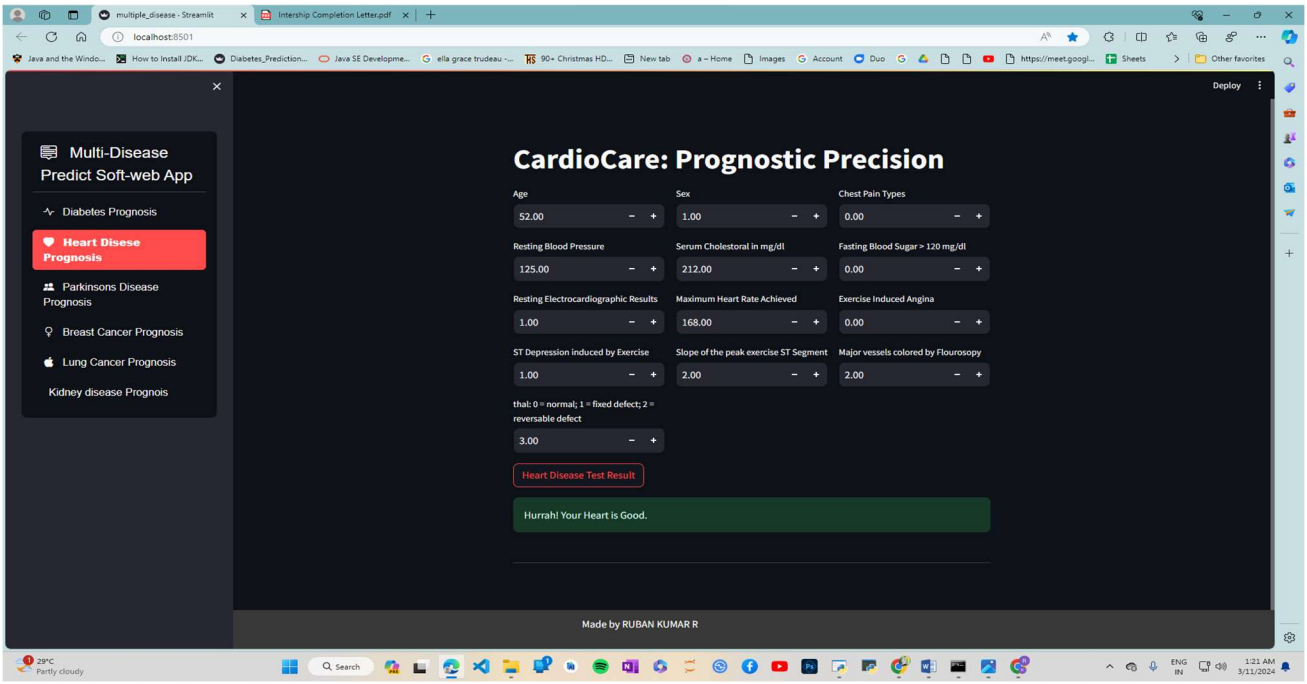


No Diabetes

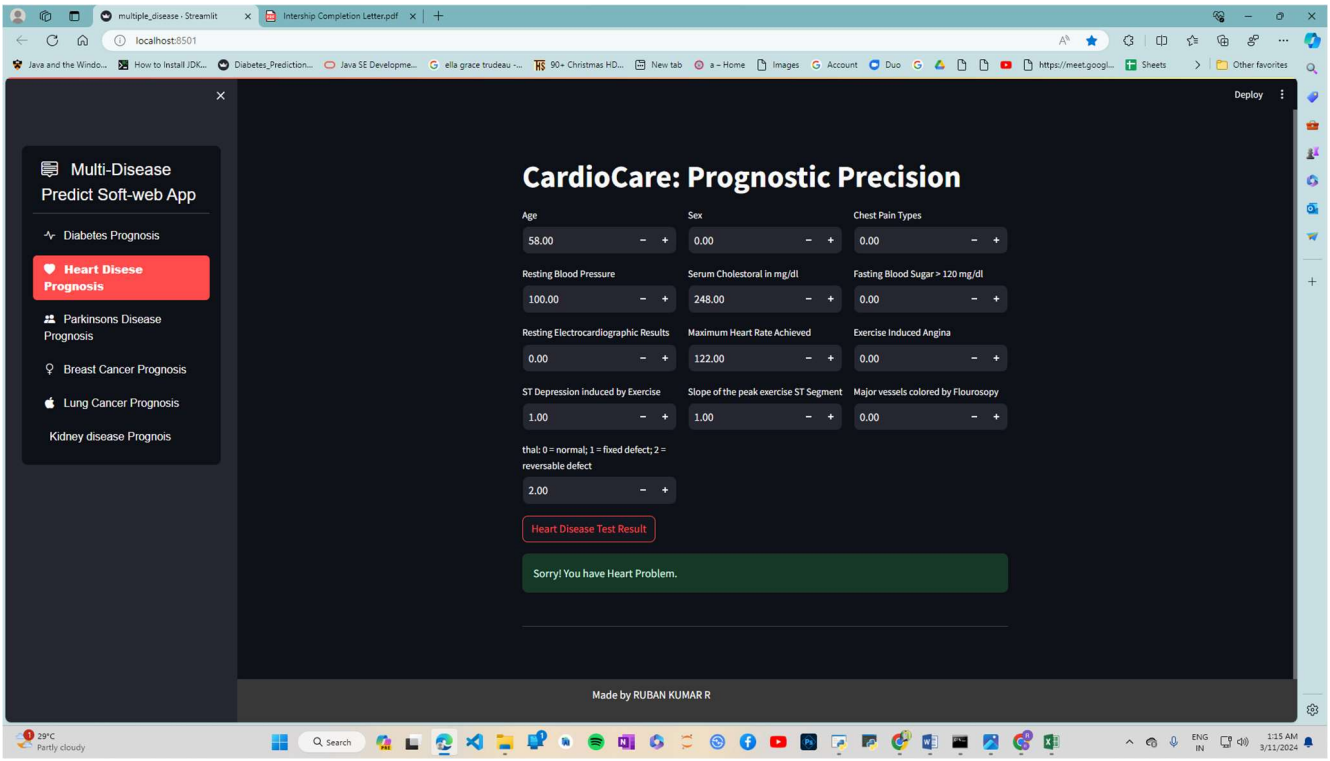


02 Heart Disease Prediction

No Disease in this heart

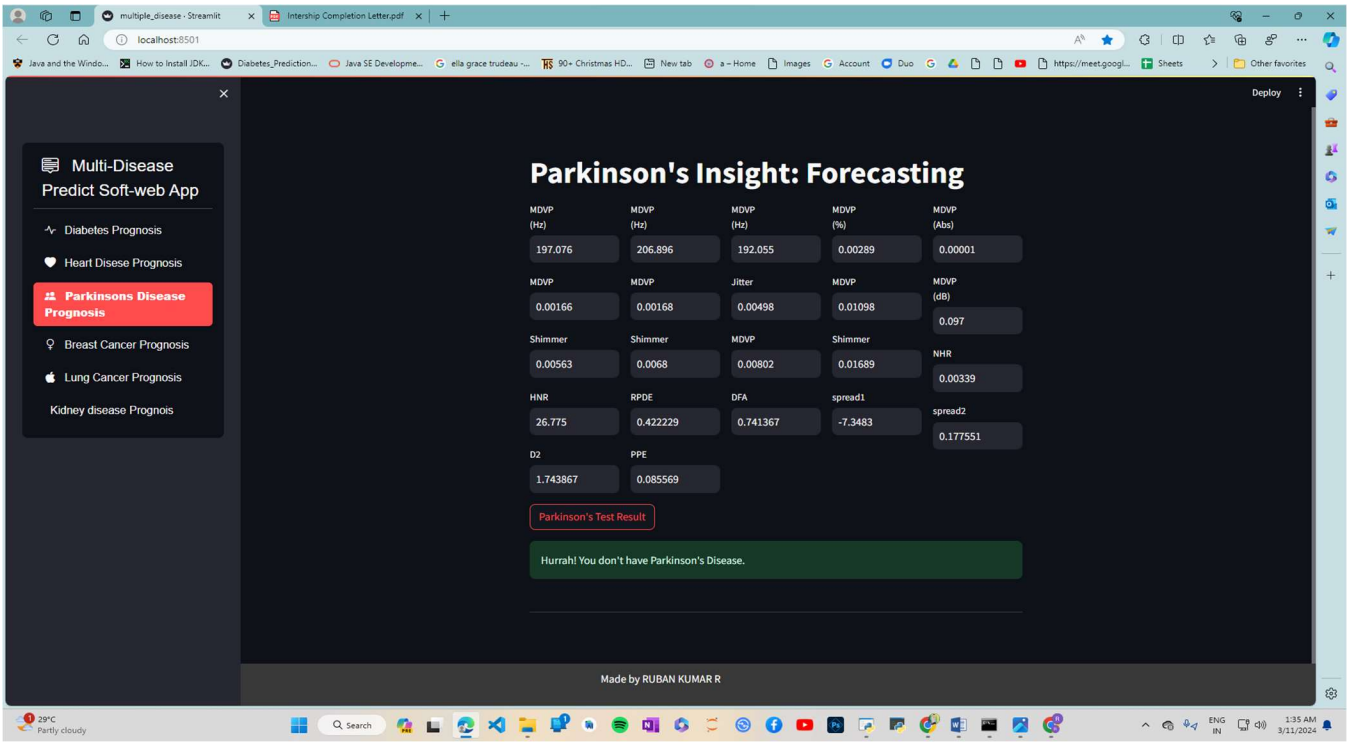


Disease in this heart

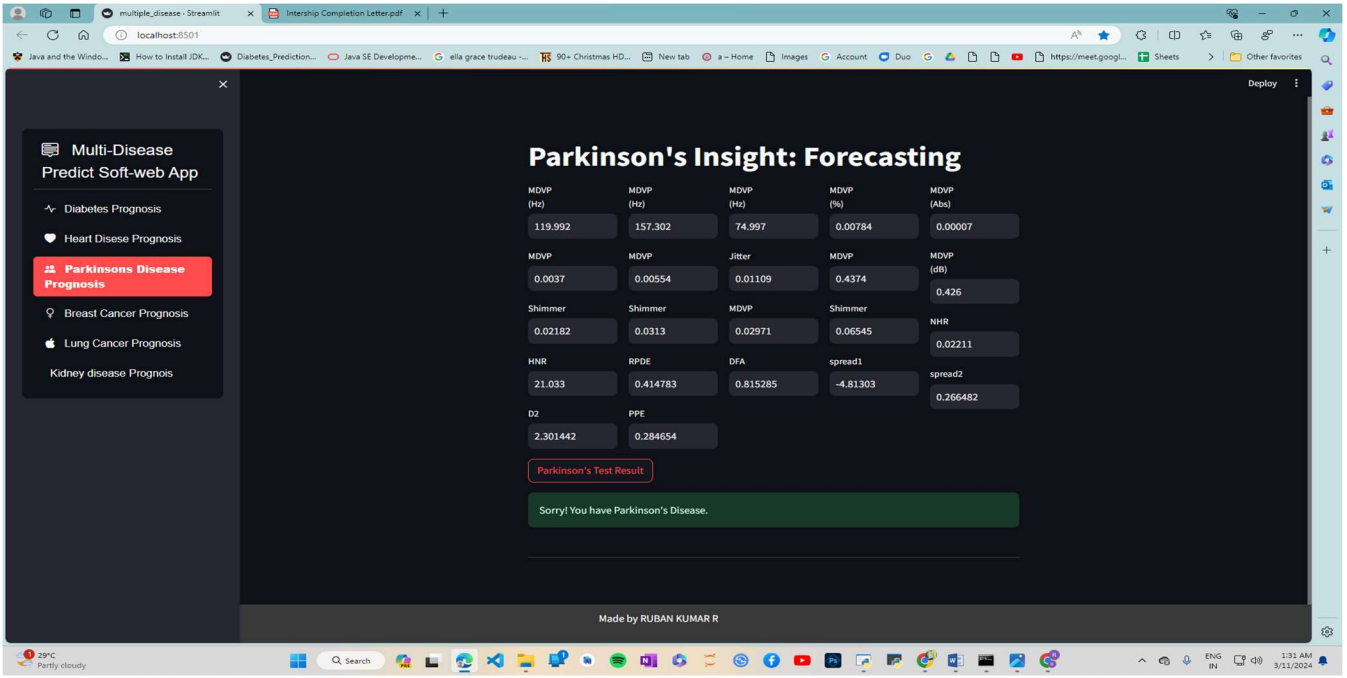


03 Parkinson's Disease Prediction

No Disease has found

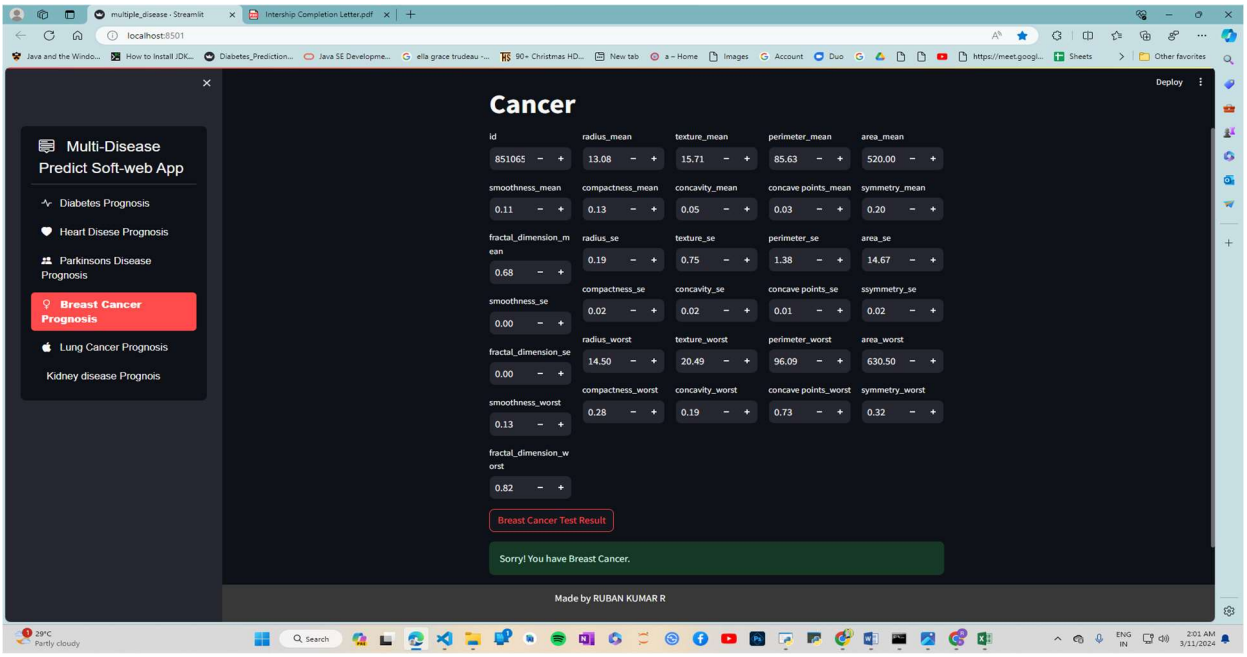


Disease has found

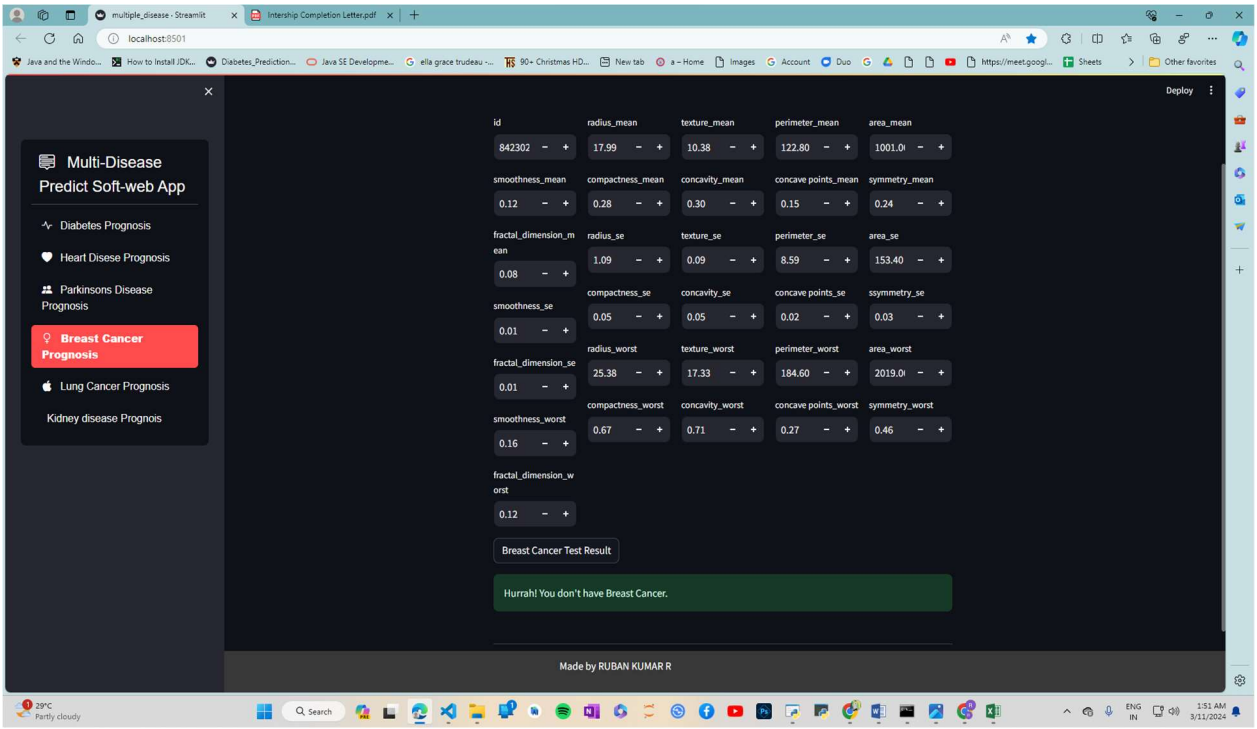


04 Breast Cancer Disease Prediction

Disease Founded

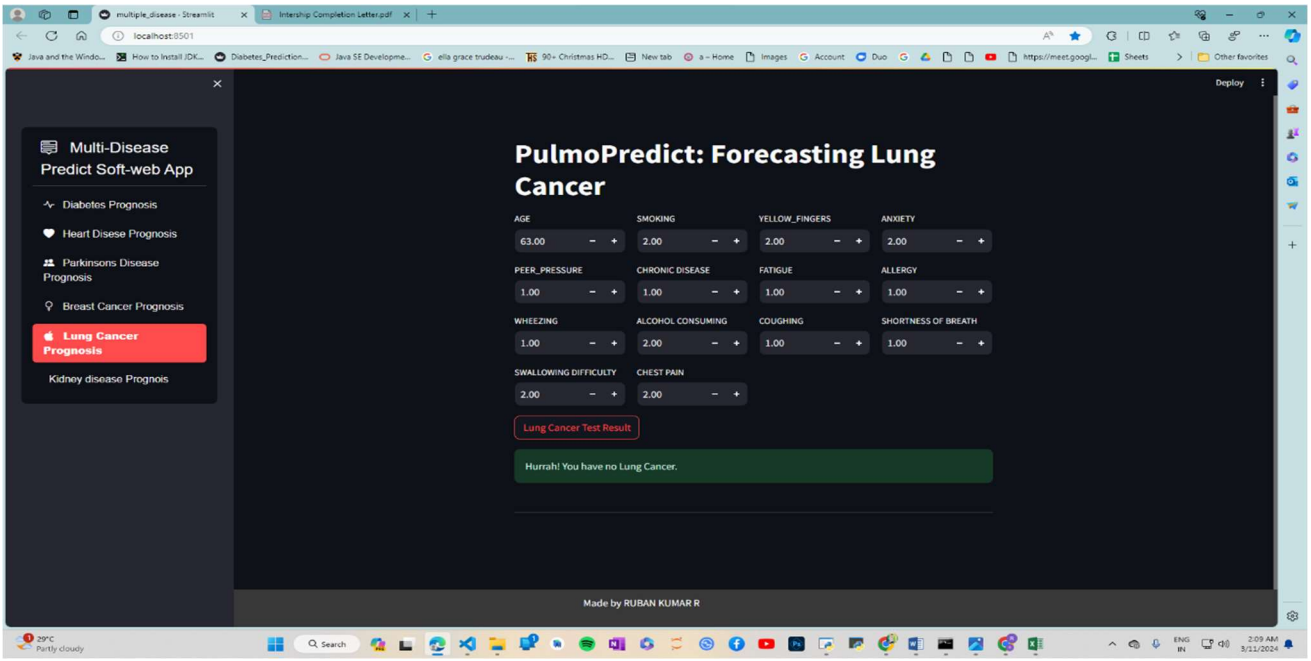


No Disease

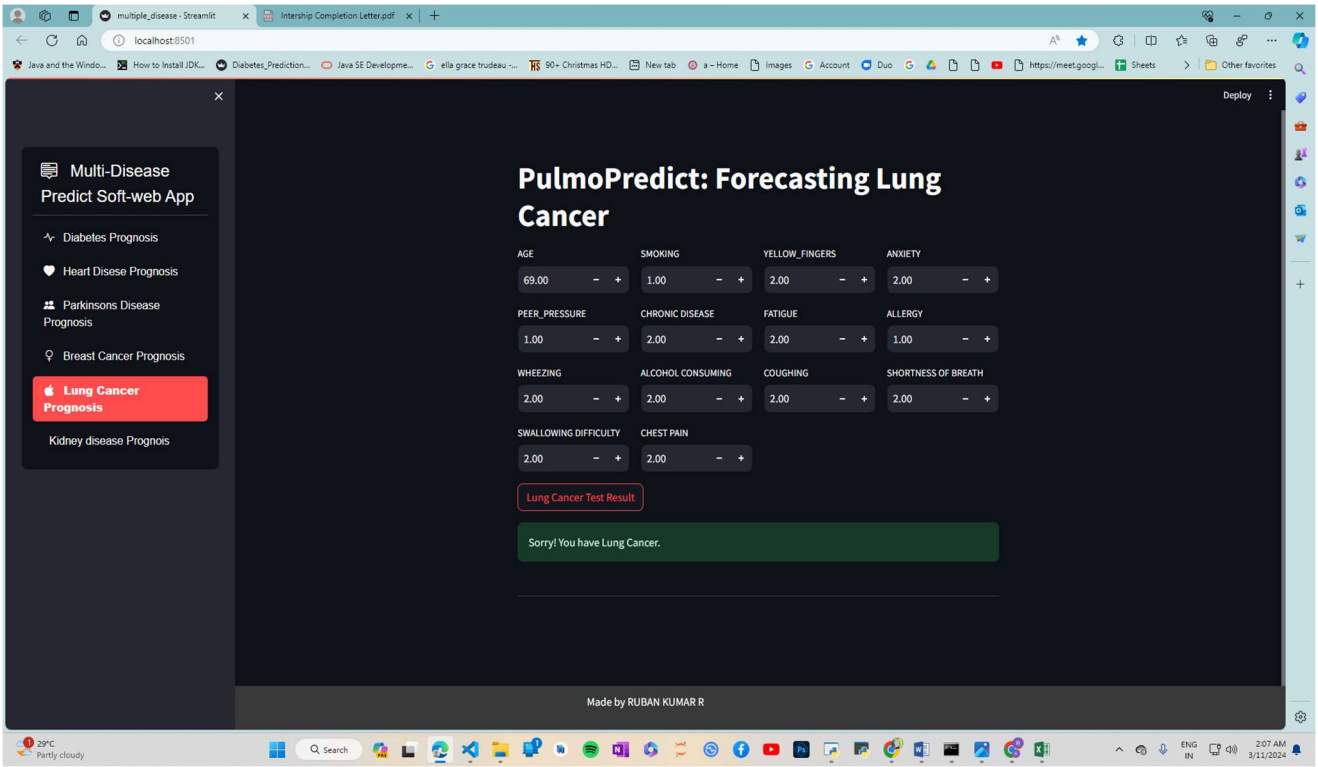


05 Lung Cancer Disease Prediction

Cancer has not found



Cancer has found



Appendix – 3 Sample Coding

```
# -*- coding: utf-8 -*-
"""
Created on Thu Nov  3 11:44:15 2022

@author: Sobhan
"""

import pickle
import streamlit as st
from streamlit_option_menu import option_menu


#loading the models

diabetes_model                                                                  =
pickle.load(open("F:\\project\\Project\\MDPS\Model\\diabetes_model.sav", "rb"))

heart_disease_model                                                             =
pickle.load(open("F:\\project\\Project\\MDPS\\Model\\heart_disease_model.sav",
"rb"))

parkinsons_model                                                                =
pickle.load(open("F:\\project\\Project\\MDPS\\Model\\parkinsons_model.sav", "rb"))

breast_cancer_model                                                             =
pickle.load(open("F:\\project\\Project\\MDPS\\Model\\breast_cancer_model.sav",
"rb"))

lung_cancer_model                                                               =
pickle.load(open("F:\\project\\Project\\MDPS\\Model\\lung_cancer_model.sav", "rb"))

kd_model    =    pickle.load(open("F:\\project\\Project\\MDPS\\Model\\kd_model.sav",
"rb"))

#sidebar for navigation

with st.sidebar:

    selected = option_menu("Multi-Disease Predict Soft-web App",

                           ["Diabetes Prognosis",
                            "Heart Disese Prognosis",
                            "Parkinsons Disease Prognosis",
```

```
                            "Breast Cancer Prognosis",
                            "Lung Cancer Prognosis",
                            "Kidney disease Prognois"],

                           icons = ["activity", "heart-fill", "people-fill",
                                    "gender-female", "apple","kidney"],

                           default_index = 0)









#Diabetes Prediction Page:

if(selected == "Diabetes Prognosis"):

    #page title
    st.title("Diabalytics: Insightful Prognosis")



# 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
    diabetes_diagnosis = " "

    # creating a button for Prediction

    if st.button("Diabetes Test Result"):
            diabetes_prediction  =  diabetes_model.predict([[Pregnancies,  Glucose,
BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, Age]])

        if (diabetes_prediction[0] == 0):
          diabetes_diagnosis = "Hurrah! You have no Diabetes."
        else:
          diabetes_diagnosis = "Sorry! You have Diabetes."

    st.success(diabetes_diagnosis)









#Heart Disease Prediction Page:

if(selected == "Heart Disese Prognosis"):

    #page title
    st.title("CardioCare: Prognostic Precision")



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

    with col1:
        age = st.number_input("Age")
```

```
    with col2:
        sex = st.number_input("Sex")

    with col3:
        cp = st.number_input("Chest Pain Types")

    with col1:
        trestbps = st.number_input("Resting Blood Pressure")

    with col2:
        chol = st.number_input("Serum Cholestoral in mg/dl")

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

    with col1:
        restecg = st.number_input("Resting Electrocardiographic Results")

    with col2:
        thalach = st.number_input("Maximum Heart Rate Achieved")

    with col3:
        exang = st.number_input("Exercise Induced Angina")

    with col1:
        oldpeak = st.number_input("ST Depression induced by Exercise")

    with col2:
        slope = st.number_input("Slope of the peak exercise ST Segment")

    with col3:
        ca = st.number_input("Major vessels colored by Flourosopy")

    with col1:
        thal = st.number_input("thal: 0 = normal; 1 = fixed defect; 2 = reversable
defect")




    # 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] == 0):
          heart_diagnosis = "Hurrah! Your Heart is Good."
        else:
          heart_diagnosis = "Sorry! You have Heart Problem."

    st.success(heart_diagnosis)







#Parkinsons Disease Prediction Page:

if(selected == "Parkinsons Disease Prognosis"):

    #page title
    st.title("Parkinson's Insight: Forecasting")



# getting the input data from the user

    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,PP
E]])

        if (parkinsons_prediction[0] == 0):
          parkinsons_diagnosis = "Hurrah! You don't have Parkinson's Disease."
        else:
          parkinsons_diagnosis = "Sorry! You have Parkinson's Disease."

    st.success(parkinsons_diagnosis)







#Breast Cancer Prediction Page:

if(selected == "Breast Cancer Prognosis"):

    #page title
    st.title("CancerGuard: Anticipating Breast Cancer")



# getting the input data from the user

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

    with col1:
        id = st.number_input("id")

    with col2:
        radius_mean = st.number_input("radius_mean")
```

```
    with col3:
        texture_mean = st.number_input("texture_mean")

    with col4:
        perimeter_mean = st.number_input("perimeter_mean")

    with col5:
        area_mean = st.number_input("area_mean")

    with col1:
        smoothness_mean = st.number_input("smoothness_mean")

    with col2:
        compactness_mean = st.number_input("compactness_mean")

    with col3:
        concavity_mean= st.number_input("concavity_mean")

    with col4:
        concave_points_mean = st.number_input("concave points_mean")

    with col5:
        symmetry_mean = st.number_input("symmetry_mean")

    with col1:
        fractal_dimension_mean = st.number_input("fractal_dimension_mean")

    with col2:
       radius_se = st.number_input("radius_se")

    with col3:
        texture_se = st.number_input("texture_se")

    with col4:
        perimeter_se = st.number_input("perimeter_se")

    with col5:
        area_se = st.number_input("area_se")

    with col1:
       smoothness_se = st.number_input("smoothness_se")

    with col2:
       compactness_se = st.number_input("compactness_se")

    with col3:
```

```
        concavity_se = st.number_input("concavity_se")

    with col4:
       concave_points_se = st.number_input("concave points_se")

    with col5:
        symmetry_se = st.number_input("ssymmetry_se")

    with col1:
        fractal_dimension_se = st.number_input("fractal_dimension_se")

    with col2:
       radius_worst = st.number_input("radius_worst")

    with col3:
        texture_worst = st.number_input("texture_worst")

    with col4:
        perimeter_worst = st.number_input("perimeter_worst")

    with col5:
       area_worst = st.number_input("area_worst")

    with col1:
        smoothness_worst = st.number_input("smoothness_worst")

    with col2:
        compactness_worst = st.number_input("compactness_worst")

    with col3:
        concavity_worst = st.number_input("concavity_worst")

    with col4:
        concave_points_worst = st.number_input("concave points_worst")

    with col5:
        symmetry_worst = st.number_input("symmetry_worst")

    with col1:
        fractal_dimension_worst = st.number_input("fractal_dimension_worst")


    #code for Prediction
    breast_cancer_check = " "

    if st.button("Breast Cancer Test Result"):
```

```
         breast_cancer_prediction = breast_cancer_model.predict([[id, radius_mean,
texture_mean,   perimeter_mean,   area_mean,   smoothness_mean,   compactness_mean,
concavity_mean,   concave_points_mean,   symmetry_mean,   fractal_dimension_mean,
radius_se,  texture_se,  perimeter_se,  area_se,  smoothness_se,  compactness_se,
concavity_se, concave_points_se, symmetry_se, fractal_dimension_se, radius_worst,
texture_worst, perimeter_worst, area_worst, smoothness_worst, compactness_worst,
concavity_worst,              concave_points_worst,              symmetry_worst,
fractal_dimension_worst]])

        if (breast_cancer_prediction[0] == 0):
          breast_cancer_check = "Hurrah! You don't have Breast Cancer."
        else:
          breast_cancer_check = "Sorry! You have Breast Cancer."

    st.success(breast_cancer_check)




#Lung Cancer Prediction Page:

if(selected == "Lung Cancer Prognosis"):

    #page title
    st.title("PulmoPredict: Forecasting Lung Cancer")



# getting the input data from the user
    col1, col2, col3, col4 = st.columns(4)


    with col1:
        AGE = st.number_input("AGE")

    with col2:
        SMOKING = st.number_input("SMOKING")

    with col3:
        YELLOW_FINGERS = st.number_input("YELLOW_FINGERS")

    with col4:
        ANXIETY = st.number_input("ANXIETY")
```

```
    with col1:
        PEER_PRESSURE = st.number_input("PEER_PRESSURE")

    with col2:
        CHRONIC_DISEASE = st.number_input("CHRONIC DISEASE")

    with col3:
        FATIGUE = st.number_input("FATIGUE")

    with col4:
        ALLERGY = st.number_input("ALLERGY")

    with col1:
        WHEEZING = st.number_input("WHEEZING")

    with col2:
        ALCOHOL_CONSUMING = st.number_input("ALCOHOL CONSUMING")

    with col3:
        COUGHING = st.number_input("COUGHING")

    with col4:
        SHORTNESS_OF_BREATH = st.number_input("SHORTNESS OF BREATH")

    with col1:
        SWALLOWING_DIFFICULTY = st.number_input("SWALLOWING DIFFICULTY")

    with col2:
        CHEST_PAIN = st.number_input("CHEST PAIN")



# code for Prediction
    lung_cancer_result = " "

    # creating a button for Prediction

    if st.button("Lung Cancer Test Result"):
                lung_cancer_report   =   lung_cancer_model.predict([[AGE,   SMOKING,
YELLOW_FINGERS, ANXIETY, PEER_PRESSURE, CHRONIC_DISEASE, FATIGUE, ALLERGY, WHEEZING,
ALCOHOL_CONSUMING,     COUGHING,     SHORTNESS_OF_BREATH,     SWALLOWING_DIFFICULTY,
CHEST_PAIN]])

        if (lung_cancer_report[0] == 0):
          lung_cancer_result = "Hurrah! You have no Lung Cancer."
        else:
```

```
        lung_cancer_result = "Sorry! You have Lung Cancer."

    st.success(lung_cancer_result)









# Check Streamlit theme
theme = st.get_option("theme.primaryColor")

# Define footer style based on theme
if theme == "#FFFFFF":  # Light mode
    footer_bg_color = "#f1f1f1"
    text_color = "#000000"
else:  # Dark mode
    footer_bg_color = "#383838"
    text_color = "#FFFFFF"

# Footer HTML
footer_html = f"""
    <style>
    .footer {{
        position: fixed;
        bottom: 0;
        left: 0;
        width: 100%;
        background-color: {footer_bg_color};
        color: {text_color};
        text-align: center;
        padding: 10px 0;
    }}
    </style>
    <div class="footer">
        <p>Made by RUBAN KUMAR R</p>
    </div>
"""

# Render footer
st.markdown("---")
st.markdown(footer_html, unsafe_allow_html=True)
```