PROJECT REPORT

On

Disease Detective

Submitted by

Shree Patel (IU2041050092)
Kavya Raval (IU2041050107)
Aashna Shah (IU2041050114)

In fulfillment for the award of the degree

Of

BACHELOR OF TECHNOLOGY

In

COMPUTER ENGINEERING

INSTITUTE OF TECHNOLOGY AND ENGINEERING INDUS UNIVERSITY CAMPUS, RANCHARDA, VIA-THALTEJ AHMEDABAD-382115, GUJARAT, INDIA,

WEB: www.indusuni.ac.in
MAY 2024

PROJECT REPORT

ON

Disease Detective

AT

In the partial fulfillment of the requirement
for the degree of
Bachelor of Technology
in
Computer Engineering

PREPARED BY

Shree Patel (IU2041050092)
Kavya Raval (IU2041050107)
Aashna Shah (IU2041050114)

UNDER GUIDANCE OF

Internal Guide

Professor Divyani Tirthyani,
Department of Computer Engineering,

I.T.E, Indus University, Ahmedabad

SUBMITTED TO

INSTITUTE OF TECHNOLOGY AND ENGINEERING
INDUS UNIVERSITY CAMPUS, RANCHARDA, VIA-THALTEJ
AHMEDABAD-382115, GUJARAT, INDIA,
WEB:
www.indusuni.ac.in
MAY 2024

CANDIDATE'S DECLARATION

I declare that final semester report entitled "**Disease Detective**" is my own work conducted under the supervision of the guide **Prof. Mrs. Divyani Tirthyani**.

I further declare that to the best of my knowledge, the report for B.Tech final semester does not contain part of the work which has been submitted for the award of B.Tech Degree either in this university or any other university without proper citation.

Candidate's Signature

Shree Patel (IU2041050092)

Guide: Mrs. Divyani Tirthyani
Professor,
Department of Computer Engineering,

Indus Institute of Technology and Engineering
INDUS UNIVERSITY– Ahmedabad,
State: Gujarat

CANDIDATE'S DECLARATION

I declare that final semester report entitled "**Disease Detective**" is my own work conducted under the supervision of the guide **Prof. Mrs. Divyani Tirthyani**.

I further declare that to the best of my knowledge, the report for B.Tech final semester does not contain part of the work which has been submitted for the award of B.Tech Degree either in this university or any other university without proper citation.

Candidate's Signature

Kavya Raval (IU2041050107)

Guide: Mrs. Divyani Tirthyani
Professor,
Department of Computer Engineering,

Indus Institute of Technology and Engineering
INDUS UNIVERSITY– Ahmedabad,
State: Gujarat

CANDIDATE'S DECLARATION

I declare that final semester report entitled "**Disease Detective**" is my own work conducted under the supervision of the guide **Prof. Mrs. Divyani Tirthyani**.

I further declare that to the best of my knowledge, the report for B.Tech final semester does not contain part of the work which has been submitted for the award of B.Tech Degree either in this university or any other university without proper citation.

Candidate's Signature

Aashna Shah (IU2041050114)

Guide: Mrs. Divyani Tirthyani
Professor,
Department of Computer Engineering,

Indus Institute of Technology and Engineering
INDUS UNIVERSITY– Ahmedabad,
State: Gujarat

INDUS INSTITUTE OF TECHNOLOGY AND ENGINEERING

COMPUTERENGINEERING

2020 -2021

CERTIFICATE

Date: <DATE>

This is to certify that the project work entitled "**Disease Detective**" has been carried out by **Shree Patel, Kavya Raval & Aashna Shah** under my guidance in partial fulfillment of degree of Bachelor of Technology in **COMPUTER ENGINEERING (Final Year)** of Indus University, Ahmedabad during the academic year 2023 - 2024.

Mrs. DIVYANI TIRTHYANI	Dr. SEEMA MAHAJAN
Professor,	Head of the Department,
Department of Computer Engineering,	Department of Computer Engineering,
I.T.E, Indus University	I.T.E, Indus University
Ahmedabad	Ahmedabad

ACKNOWLEDGEMENT

Towards the successful completion of my B. Tech in Computer Engineering final year project with the help of our mentor/guide Prof Divyani Tirthyani.

I am sincerely gratitude to Mrs. Divyani Tirthyani who providing me this opportunity and necessary support during the development of the project.

I have no other words to express my sincere thanks to all faculties of Indus University, Ahmedabad for their kind co-operation and able guidance. I would like to thank Mrs. Divyani Tirthyani, my internal project guide in college without whom the project could not be executed.

- Shree Patel,
Kavya Raval,
Aashna Shah

TABLE OF CONTENT

ABSTRACT

People are currently suffering from a variety of diseases. Many people are unsure if the symptoms they are experiencing are indicative of a certain disease, and hence they are unable to take the required safeguards. Anticipating the disease during prodromal stage lowers the likelihood of complications. People will not be able to visit a doctor every time they experience a symptom. It may sometimes become a serious ailment if not treated. A model is suggested that uses a variety of symptoms as input to predict the illness. For **disease prediction, the suggested method utilizes Decision trees, KNN, and Random forest classifiers.** The ultimate result will be the mode of all these machine learning models. Users will be given a graphical user interface (GUI) to choose their symptoms. The final result will be shown on the interface using all three machine learning techniques, and feature extraction will be done depending on their symptoms. **Four modules** make up the proposed methodology. Preprocessing will be done on the dataset in the first module. The **decision tree classifier** is used to generate a prediction model in the second module. **The Random forest** method is used for forecast the illness in the third module, and the **K's Nearest Neighbour(KNN)** technique is utilized in the fourth model, with the mode of the outputs from all the three models taken into account.

TABLES

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

- **PROJECT SUMMARY**
- **PROJECT PURPOSE**
- **PROJECT SCOPE**
- **GOALS AND OBJECTIVES**
- **TECHNOLOGIES AND LITERATURE OVERVIEW**

1.1 PROJECT SUMMARY

Utilizing machine learning algorithms, Disease Prediction with Symptoms analyzes symptoms provided by users to accurately forecast potential diseases. With user-friendly interface design and sophisticated algorithms, the system delivers personalized recommendations and comprehensive disease insights. This initiative fosters early disease identification, tailors healthcare provision, and bolsters public health awareness, thereby improving overall healthcare efficacy. The DISEASE DETECTIVE can classify the diseases into 41 different diseases.

1.2 PROJECT PURPOSE

The aim of this project is to create a DISEASE DETECTIVE App through Machine Learning, enabling precise forecasts of different patient ailments. By utilizing datasets sourced from Kaggle, the system will be trained and integrated into a web application. It aims to accelerate disease diagnosis in comparison to conventional lab techniques, achieving a success rate exceeding 94% for each model tested. The main goal is to rapidly determine whether a patient has a particular disease, achieving prediction accuracies of up to 94% for each condition.

1.3 PROJECT SCOPE

In the future, additional models could undergo training and deployment across different sectors, thereby improving efficacy through the inclusion of more symptoms in disease prediction. The project aims to create a web-based disease prediction application where users submit symptoms for analysis. Employing machine learning, the application accurately forecasts diseases, featuring symptom input, disease prediction, and tailored recommendations. Emphasis is placed on user-friendly interface, precise prediction algorithms, and smooth integration for efficient disease diagnosis.

1.4 GOALS AND OBJECTIVES

A system needs development to empower users in predicting diseases remotely, eliminating the need for physician visits. It involves recognizing diseases through patient symptoms, employing diverse Machine Learning techniques. Current methods lack a standardized approach for handling text and structured data, which the proposed framework will address. This system will encompass both structured and unstructured data, aiming to enhance prediction accuracy through Machine Learning advancements.

- Support healthcare professionals in their daily duties by aiding them in their essential role.
- Inform the community about prevalent diseases like diabetes, Fungal Infection, Allergy, GERD, Dengue, etc. focusing on their transmission, prevention methods, and government-implemented preventive measures.

In this Diseases Detector following technologies are being used:

1.Programming Language : Python

Python is a widely-used high-level programming language recognized for its simplicity and clear syntax, which prioritize readability. It is favored by developers across diverse fields such as web development, data analysis, artificial intelligence, and scientific computing. Python accommodates various programming approaches, including procedural, object-oriented, and functional programming. With its extensive standard library and broad array of third-party packages, Python proves adaptable and applicable to a broad spectrum of projects.

2. Machine Learning Libraries: Scikit-learn

Scikit-learn is a popular machine learning library in Python, widely used for building predictive models and performing data analysis tasks. It provides a wide range of tools for various machine learning tasks, including classification, regression, clustering, dimensionality reduction, and model evaluation. Scikit-learn is known for its ease of use, extensive documentation, and compatibility with other Python libraries such as NumPy, pandas, and matplotlib. It is widely used in both academia and industry for developing machine learning solutions.

3.Tkinter:

Tkinter serves as a common GUI (Graphical User Interface) toolkit for Python, enabling the development of desktop applications with visual interfaces. It offers a range of predefined components like buttons, labels, text entry fields, and menus, enabling developers to craft interactive applications. Tkinter's simplicity and integration with Python make it readily available for developers, requiring no additional installations. Consequently, it is extensively utilized across diverse domains, including education, business, and software development, to create desktop applications ranging from basic to moderately complex.

4. Trained Dataset: Kaggle

A Kaggle trained dataset is meticulously curated and provided by the Kaggle platform for training machine learning models. Kaggle, a renowned community for data scientists, offers datasets, competitions, and collaboration avenues. These datasets are preprocessed and cleaned, containing pertinent features and labels for various domains like healthcare and finance. Data scientists leverage them to refine machine learning techniques, fostering advancements in artificial intelligence.

CHAPTER 2

LITRATURESURVEY

- **INTRODUCTION TO SURVEY**
- **WHY SURVEY?**

2.1 INRODUCTION TO SURVEY

A study conducted by K.M. Al-Aidaroos, A.A. Bakar, and Z. Othman compared the performance of Nave Baeyes with LR, KStar, Decision Tree, Neural Network, and ZeroR classifiers across 15 real-world medical scenarios. Nave Baeyes demonstrated superior predictive accuracy, outperforming other algorithms in 8 out of 15 datasets. Darcy A. Davis et al. highlighted the inefficiency of globally treating chronic illness, prompting a study on disease risk forecasting using DISEASE DETECTIVE. DISEASE DETECTIVE predicts potential disease risks based on patient history and ICD-9-CM codes, while DISEASE DETECTIVE, an enhanced version, improves efficiency by integrating ensemble principles. These systems enable comprehensive disease prediction without requiring advanced knowledge, with ICAR offering early alerts for thousands of illnesses years in advance. The ICARE system facilitates exploration of various disease backgrounds, stimulating new discussions on early detection and prevention. Decision Tree displayed superior performance in experiments, with Bayesian classification showing comparable accuracy in certain instances. Shadab Adam Pattekari and Asma Parveen utilized the Decision Tree Algorithm for heart disease prediction by comparing patient data against predefined values. M.A.NisharaBanu and B. Gomathy analyzed medical issues using association rule mining, grouping, and clustering, utilizing decision trees to visualize potential outcomes based on factors such as age, sex, and lifestyle.

2.2 WHY SURVEY?

- The Provide foundation of knowledge on topic
- Identify areas of prior scholarship to prevent duplication and give credit to other researchers
- Identify inconstancies: gaps in research, conflicts in previous studies, open questions left from other research
- Identify need for additional research (justifying your research) Identify the relationship of works in context of its contribution to the topic and to other

works.

- Place your own research within the context of existing literature making a case for why further study is needed.

CHAPTER 3

PROJECT MANAGEMENT

- **PROJECT PLANNING & SCHEDULING**
- **RISK MANAGEMENT**

3.1 PROJECT PLANNING AND SCHEDULING

PROJECT PLANNING:

- The problem is decomposed into smaller problems, software managers use historical data (as well as personal experience and intuition) to determine estimates for each.
- Project planning is a method for stating how to complete a project within a certain period of time, usually with defined stages, and with designated resources.
- It begins by setting the scope of a project and eventually working through each level of dependent actions, tasks, checkpoints and deadlines.
- The final estimates are typically adjusted by taking project complexity and risk into account.
- The resulting work product is called project management plan.
- The objective of the software planning is providing the framework the enables the manager to make reasonable estimation of the resources, cost and schedule.
- Project planning involves estimating how much time, efforts, money, and resources will be required build a specific software system. So that project outcome can be bounded.

PROJECT SCHEDULING:

- Software project scheduling is an activity that distributes estimated efforts across the planned duration by allocating the effort to specific software engineering tasks.
- Proper Scheduling Requires:
    - All tasks appear in network.
    - Effort and timing are intelligently allocated to each task.
    - Interdependencies between tasks are properly indicated.
    - Resources are allocated for the work to be done.

3.2 PROJECT DEVELOPMENT APPROACH

The model was selected based on the project's requirements, which entail adjustments in certain stages to improve subsequent phases. This approach enables accurate estimation of time required for each major project phase.

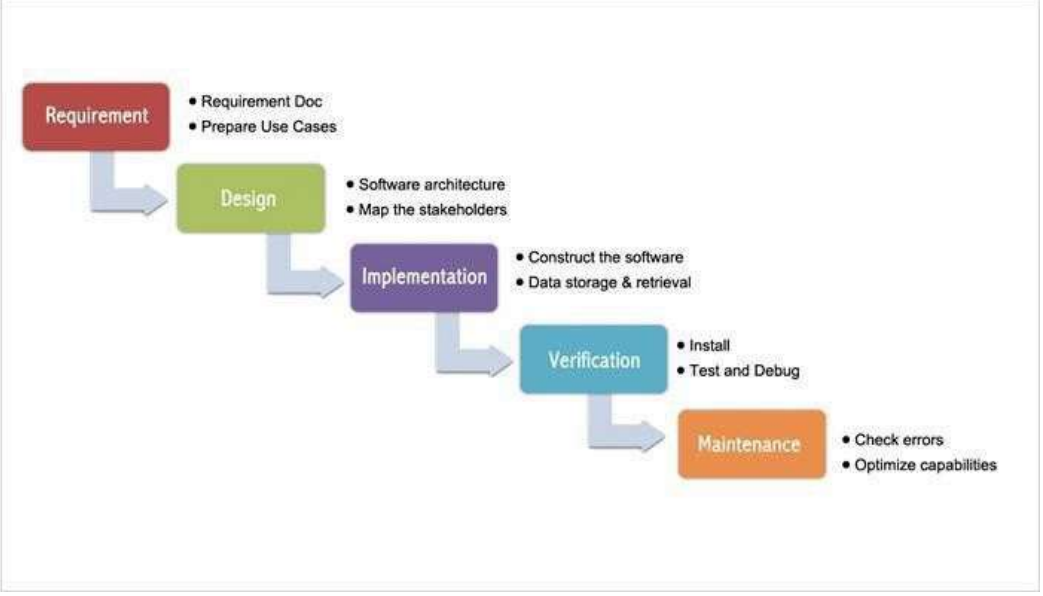

Fig 3.1.1 Waterfall Model

❖ Requirement Analysis

1) User Interface: Develop an intuitive platform facilitating easy symptom input.

2) Symptom Input: Establish a system enabling accurate symptom submission.

3) Data Collection: Compile comprehensive symptom and disease data from reputable sources.

4) Data Preprocessing: Refine collected data to ensure reliability and uniformity.

5) Machine Learning Models: Employ algorithms to analyze symptoms and forecast diseases.

6) Accuracy Assessment: Evaluate model precision using pertinent metrics and test datasets.

7) Integration: Fuse prediction models with the interface for instantaneous disease prognosis.

8) Personalized Recommendations: Provide tailored suggestions based on user profiles and predicted diseases.

❖ Design

In the design phase, key tasks include:

- Crafting an intuitive interface for symptom input
- Selecting suitable machine learning algorithms
- Defining system architecture for frontend, backend, and integration
- Ensuring smooth user experience and personalized features
- Planning for scalability to handle increased usage
- Implementing thorough testing for algorithm reliability and system robustness.

❖ Implementation

Based on the results of Requirement Analysis and Design phases, selection of appropriate technology was done and to achieve the planned system, actual coding was done in the Implementation Phase. In this phase, various software and installation files were downloaded to work with technologies.

The task of designing the database and creating methods to retrieve the data from it was also done in this phase.

❖ Testing

As the name of the phase suggests, the task of testing the fully functioning system was done in this phase. It included many steps such as accessing all its pages on multiple devices with different resolution or screen sizes to test its Responsiveness. It also includes testing all the possible ways of storing, updating and retrieving the data from the database.

In this phase, we have also train the data that has been collected and upon which the machine learning algorithm has been implemented.

❖ Maintenance

In this stage, the well-functioning system is being looked after if it need any changes in its design or database planning. Such changes are usually required at certain intervals of time to ensure that the system works properly without causing any errors or data losses.

3.3 RISK MANAGEMENT

After establishing the context, the next step in the process of managing risk is to identify potential risks. Risks are about events that, when triggered, cause problems. Hence, risk identification can start with the source of problems, or with the problem itself.

In this project there can be following risks.

- The disease prediction app project entails risks such as data privacy, model accuracy, technical challenges, user acceptance, legal and ethical issues, scalability, regulatory compliance, data bias, dependency on external sources, and feedback incorporation. Mitigation strategies include planning, communication, testing, and compliance adherence throughout the project lifecycle.

3.3.1 RISK IDENTIFICATION:

Risk is an inevitable concept of software projects and it mainly concerns future happenings. We want to produce a well working system therefore we have to consider all the possible defects and unexpected conditions. First of all we have to define possible risks and develop a management style for each of them. After a deep investigation we identified the following possible risks of our project.

- The tasks are planned in the schedule may overflow.
- Some team members may not focus on the project deeply.
- A team member may not be suitable for group work.
- The motivation of team members may not be sufficient.
- The number of people in the project may not be enough for the project.

3.3.2 RISK ANALYSIS:

For the risk management we select the most catastrophic risks of this list and we developed special several strategies in order to handle or block these unexpected situations. According to our risk table the most effective risks are as shown below.

RISK	PROBABILITY	EFFECTS
Application components which should be reused contains defects limit their functionality.	Moderate	Serious
Change of requirements which require proposal of major	High	Serious
Scheduling Slippage	Moderate	Serious
The time required for the development	High	Tolerate
Power Failure	High	Tolerates
Inexperienced team members	High	Tolerate
The size software is underestimated.	High	Tolerate

Table 3.3.2 Risk Analysis

3.3.3 RISK PLANNING:

Risk planning is the process of identifying, prioritizing, and managing risk. Every project or initiative has **objectives**, that is, goals that it seeks to accomplish. These are often called Critical Success Factors (CSF).

Risk events threaten the successful completion of these critical success factors. Thus, risk planning involves identifying the most important risk events in advance, prioritizing them, and developing the appropriate risk response plans.

The output from the activity is a Risk Mitigation Plan that contains set if actions directed at minimizing the potential occurrence or impacts of risks on a project and Risk Contingency Plan to be activated. For low impact, low probability risks, a mitigation plan may not be prepared, rather these risk items will be monitored to ensure that they do not transpire or evolve risks.

3.3.4 RISK AVOIDANCE:

Risk avoidance is not performing any activity that may carry risk. A risk avoidance methodology attempts to minimize vulnerabilities which can pose a threat. Risk avoidance and mitigation can be achieved through policy and procedure, training and education and technology implementations.

Risk avoidance in the disease prediction app project involves proactively implementing measures to prevent potential risks. This includes ensuring robust security measures for protecting medical data, validating data thoroughly to ensure accuracy, employing skilled professionals and reliable technologies, prioritizing user feedback and testing, adhering to legal and ethical guidelines, designing for scalability and performance, conducting regular audits for compliance, validating external data sources, establishing effective feedback mechanisms, and continuously monitoring and reviewing risks.

RISK	PLAN
Requirement changes	Derive traceability information to access requirements.
Change	Impact and maximize information hiding.
Performance	Investigate Database which can effectively process.

CHAPTER 4

SYSTEM REQUIREMENTS

- **USER CHARACTERISTICS**
- **FUNCTIONAL/NON-FUNCTIONAL REQUIREMENTS**
- **SOFTWARE/HARDWARE REQUIREMENTS**

4.1 USER CHARACTERISTICS

There are mainly two different types of users interacting with this system. They are: End Users, Healthcare Providers and Researchers. Each of this user has some unique role and responsibilities in this system.

The following description indicates how each of the user interacts with the system:

- **End User**
- **Healthcare Providers**
- **Researchers**

- **End User**

  End users are individuals who use the application to input symptoms and obtain forecasts regarding possible illnesses.

- **Healthcare Providers**

  Healthcare providers refer to medical professionals who might employ the application as an adjunctive resource for diagnosing diseases and strategizing treatment plans.

- **Researchers**

  Researchers are individuals from academic or professional backgrounds who could utilize the application's data for epidemiological investigations, disease monitoring, or medical research endeavors.

4.1 CONSTRAINTS

4.1.1 Data Availability:

Data availability pertains to the accessibility of thorough and up-to-date symptom databases for utilization within the application. Restricted access to these databases can hinder the application's capacity to generate precise disease predictions based on symptoms. Insufficient data may result in challenges for the app to deliver dependable and exact forecasts, possibly resulting in inaccuracies when diagnosing medical conditions.

4.1.2 Accuracy Limitations:

Despite achieving high levels of prediction accuracy, accuracy limitations are constraints faced by the application. In medical scenarios, complexities in conditions and variations in symptom presentation can still lead to misdiagnoses. Despite the application's best efforts to accurately predict diseases, the inherent complexities and variations in medical conditions may result in incorrect predictions, potentially leading to misdiagnoses.

.

4.2 HARDWARE AND SOFTWARE REQUIREMENTS

Hardware and software requirements indicate the minimum required configuration to be present in the user devices to work effectively with the system. Devices that are not able to fulfil these requirements might not be able to work with the system and it may result in inaccessibility of few functionalities of the system on that particular user device.

4.2.1 *Software Requirements*

Operating System	Windows, Linux and MacOS
Web Browser	No need for any particular browse as it an application
Compatibility with Technologies	Jupyter notebook or google collab notebook, Visual Studio Code
Kaggle	It provide and online machine system which will help to train and test the downloaded dataset before connecting it to jupyter notebook

Table 4.3.1 Software Requirements

4.2.2 *Hardware Requirements*

Processor	Any Processor
RAM	4 GB or more
Graphics	Intel HD Graphics or higher

Table 4.3.2 Hardware Requirements

CHAPTER 5

SYSTEM ANALYSIS

- **STUDY OF CURRENT SYSTEM**
- **PROBLEMS IN CURRENT SYSTEM**
- **REQUIREMENTS OF NEW SYSTEM**
- **FEASIBILITY STUDY**

5.1 STUDY OF CURRENT SYSTEM

The conventional method of disease prediction typically involves healthcare professionals conducting manual diagnosis based on patient symptoms, medical history, and physical examinations. This process relies on the expertise and experience of the healthcare provider to recognize patterns and offer informed predictions about potential diseases. Additionally, diagnostic procedures like blood tests, imaging scans, and biopsies may be employed to confirm or exclude specific conditions.

5.2 PROBLEMS IN CURRENT SYSTEM

At business system used these days includes a lot of paper work and due to it there are increased chances of facing problems of data redundancy and data inefficiency

In this traditional approach, disease prediction heavily depends on the knowledge and discernment of medical professionals, which may vary due to individual expertise and experience. Although this method has been effective over time, it can be constrained by factors such as human error, differences in diagnosis among practitioners, and the time-consuming nature of manual assessments. Also, this system costs a lot of money.

5.3 REQUIREMENTS OF NEW SYSTEM

A new system which can overcome the drawbacks of currently used system is required to make the diseases prediction hassle-free and efficient. This proposed system aims to simplify this process by providing solutions to problems faced when operating with the current systems.

Automated systems have the capability to swiftly and effectively carry out routine diagnostic duties, lessening the workload on healthcare providers and lowering the chance of human mistakes. This streamlined approach to diagnostics can enhance overall efficiency and streamline the diagnostic workflow.

Technological innovations permit the remote surveillance of patients' health conditions and streamline telemedicine consultations. This enables the prompt identification of symptoms and timely intervention, especially in distant or underserved regions.

5.4 FEASIBILITY STUDY

A Feasibility study aims to objectively and rationally uncover the strength and weakness of an existing business or proposed venture, opportunities and threats present in the environment, the resources required to carry through and ultimately the prospects for success.

In simple terms, the two criteria to judge feasibility are cost required and value to be attained. Feasibility study is an assessment of the practicality of proposed project.

The feasibility study is done to test weather the proposed system will fulfill the requirement of the organization and feasible under different criteria.

5.4.1 TECHNICAL FEASIBILITY

Technical feasibility determines whether the technology needed for the proposed system is available and how it can be integrated within the society. Technical evaluation must also assess whether the existing system can be upgraded to use the new technology and whether the organization has the expertise to use it.

The minimum memory requirement is 4GB of RAM while more than 4GB is better to have for better performance. As far as software is concerned, Python 3.x, sci-kit learn and Tkinter is installed.

5.4.2 ECONOMICAL FEASIBILITY

Economic feasibility determines whether the required project is capable of generating financial gains for an organization. It involves the cost incurred on the project development team, estimated cost software, cost of performing feasibility study, and so on. For this, it is essential to consider expenses made on purchases and activities required to carry out software development.

We have tried to complete this project with the minimum budget so that cost factor cannot influence the denial of installation of such a useful project.

5.4.3 OPERATIONAL FEASIBILITY

Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. The proposed software must have high operational feasibility.

CHAPTER 6

DETAIL DESCRIPTION

- **USER MODULE**

6.1 User Module:

User can be able to describe his/her problem by telling the symptoms that they are facing while trying to figure out the diseases. If they want to know basic description of the diseases they will be able to know that. If they want to know what precaution they can take so that the severity of the diseases does not increase.

CHAPTER 7

TESTING

- **BLACK-BOX TESTING**
- **WHITE-BOXTESTING**
- **TEST CASES**

7.1 Black- Box Testing:

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

7.2 White- Box Testing:

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

7.3 Test Cases:

Sr. No.	Test Case	Expected Output	Actual Output	Test Case Result
1	Input Validation Test Case: Enter invalid symptoms (e.g., non-alphabetic characters, empty input).	Error message prompting user to enter valid symptoms.	Error message displayed.	Passed
2	Accuracy Test Case: Enter symptoms known to be associated with a specific disease.	Prediction of the corresponding disease with a high probability.	Prediction matches the expected disease with a high probability.	Passed
3	Edge Case Test Case: Enter symptoms that could indicate multiple diseases.	Prediction of the most likely disease based on symptom severity and frequency.	Prediction of the most probable disease.	Passed
4	Performance Test Case: Input a large set of symptoms.	App provides prediction within a reasonable time frame.	Prediction is provided within an acceptable time frame.	Passed
5	Compatibility Test Case: Test the app on different devices and operating systems.	App functions correctly on various platforms.	App functions correctly on different devices and OS versions.	Passed
6	Regression Test Case: Test the app after making updates or modifications.	App retains its functionality and accuracy.	App functions correctly after updates.	Passed

Table 7.3 Test Case

Test Objectives:

There is strong evidence that effective requirement management leads to overall project cost savings.

The three primary reasons for this are:

Requirement errors typically cost well over 10 times more to repair than other errors. Requirements errors typically comprise over 40% of all errors in a software project. Small re-education in the number of requirement errors pays big dividends in avoided network costs and scheduled delay.
The testing procedure should care for all of these, as well as, in order to attain flawless, error free and efficient system too.

.

CHAPTER 8

SYSTEM DESIGN

- **CLASS DIAGRAM**
- **USE-CASEDIAGRAM**
- **ACTIVITY DIAGRAM**
- **DFD DIAGRAM**
- **ER DIAGRAM**
- **DATA DICTIONARY**

8.1 Class Diagram:

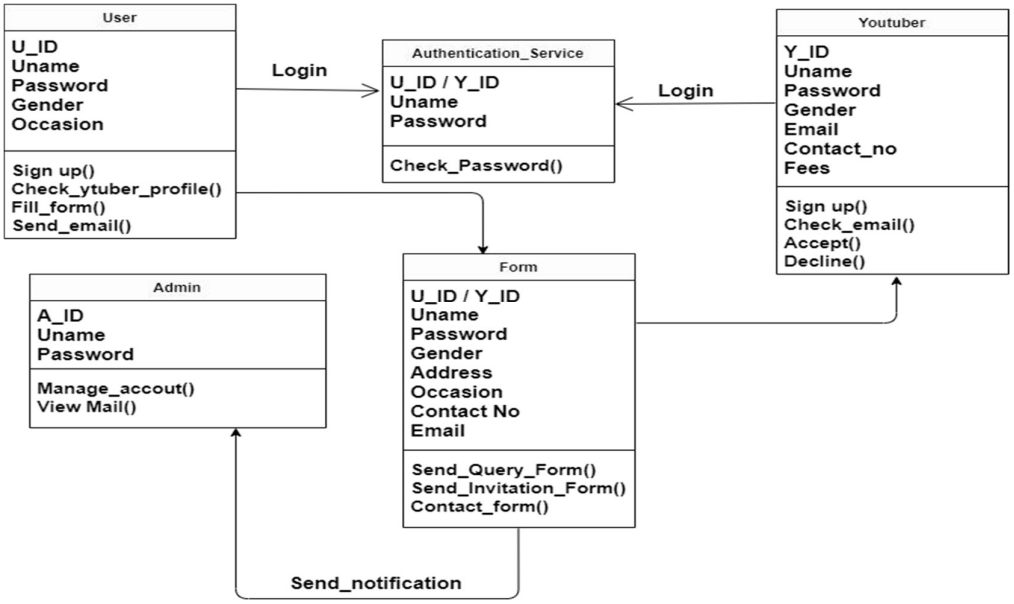

Fig 8.1 Class Diagram

8.2 Use Case Diagrams:

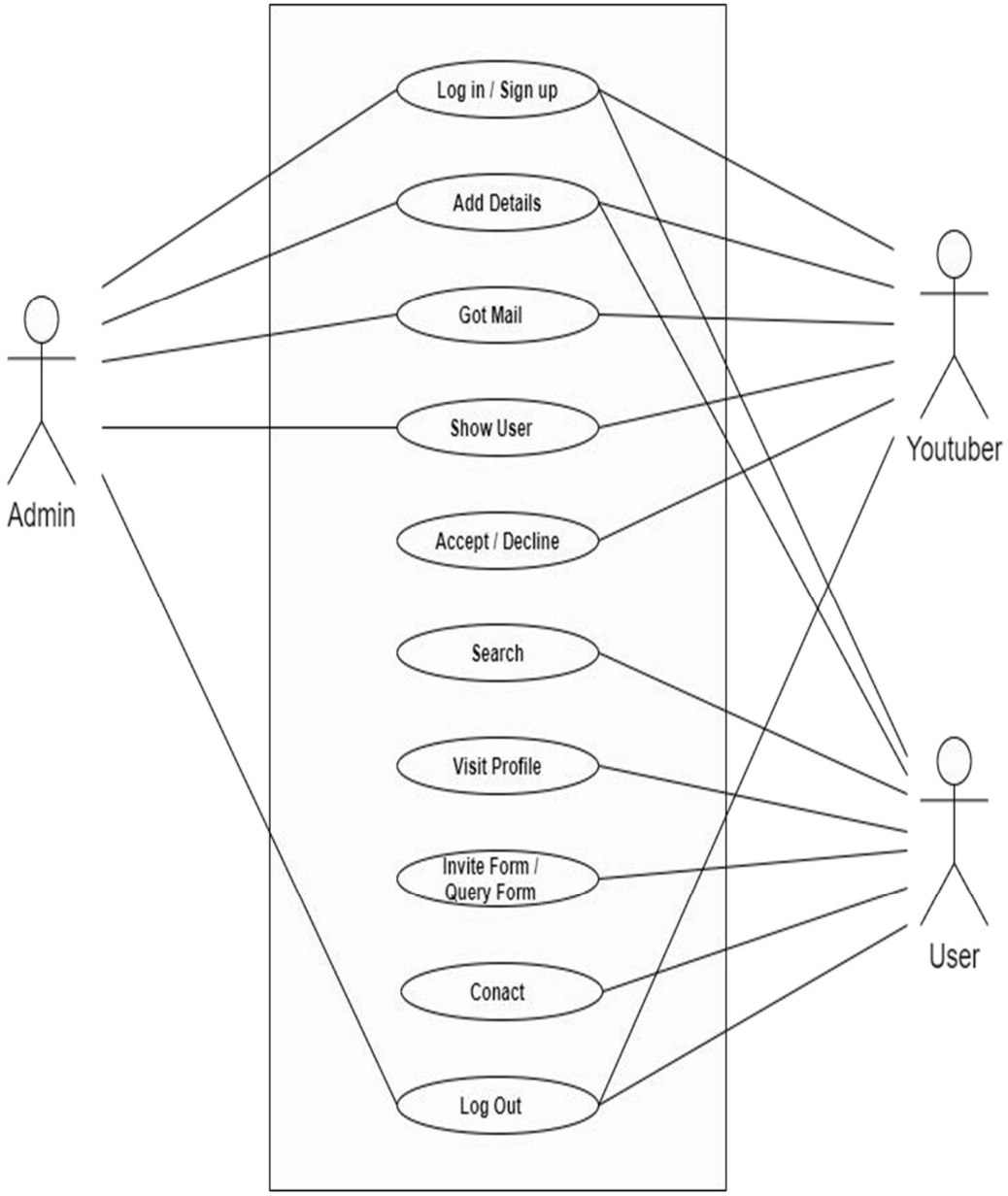

Fig 8.2 Use-Case Diagram

8.3 Activity Diagram:

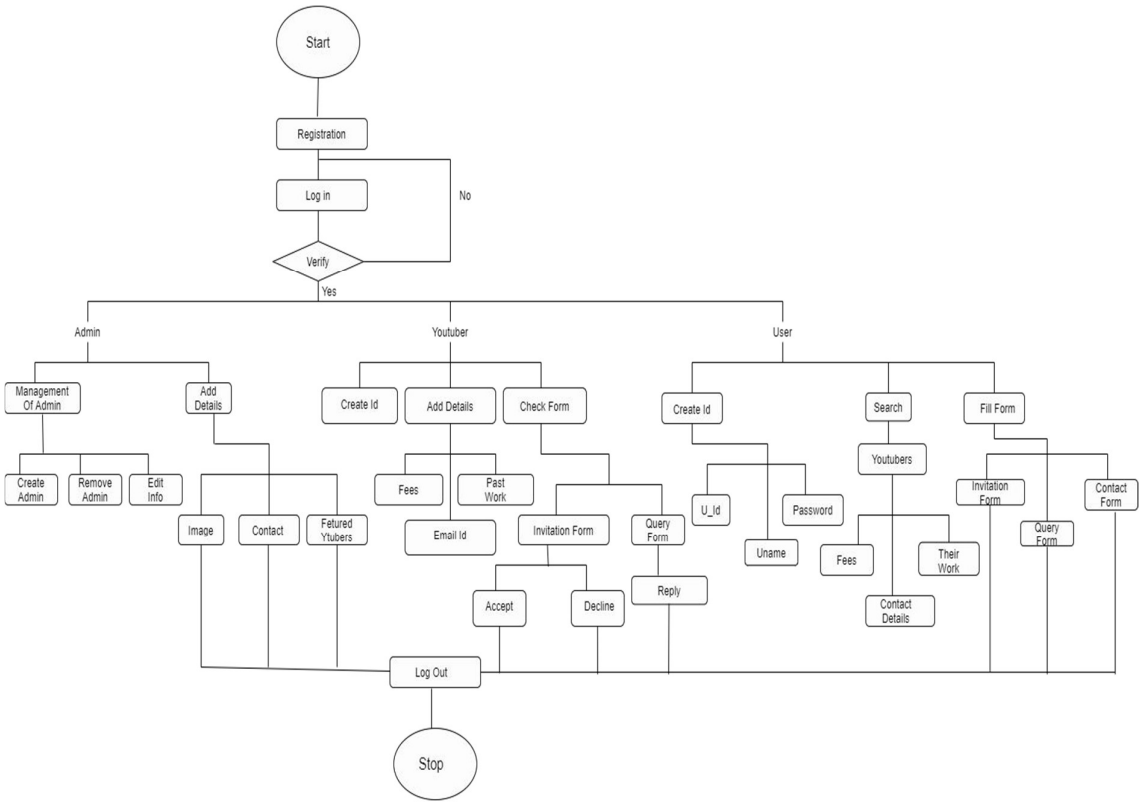

Fig 8.3 Activity Diagram

8.4 DFD Diagram:

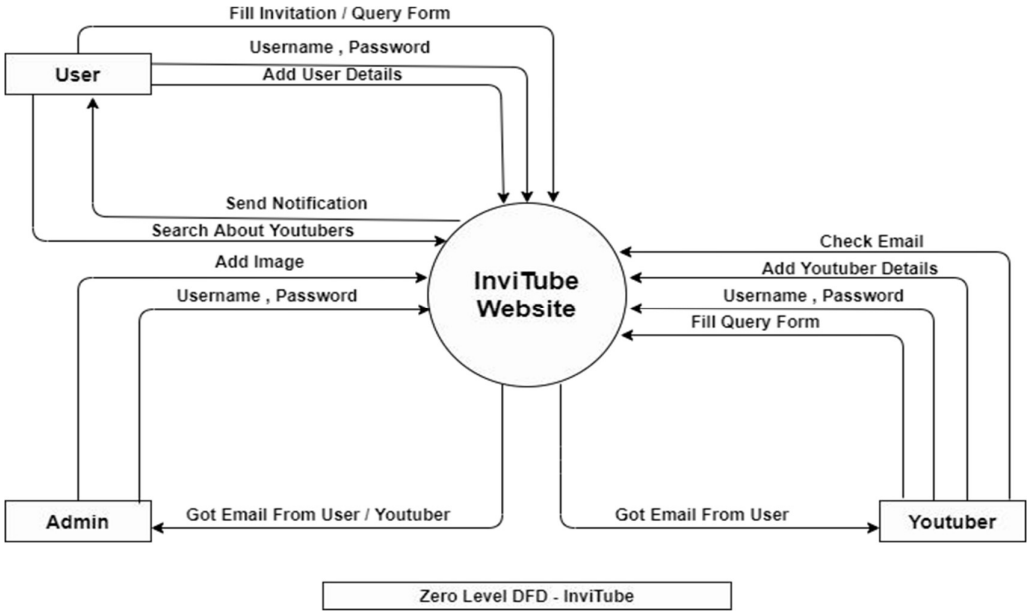

Fig 8.4.1 DFD Diagram(Level 0)

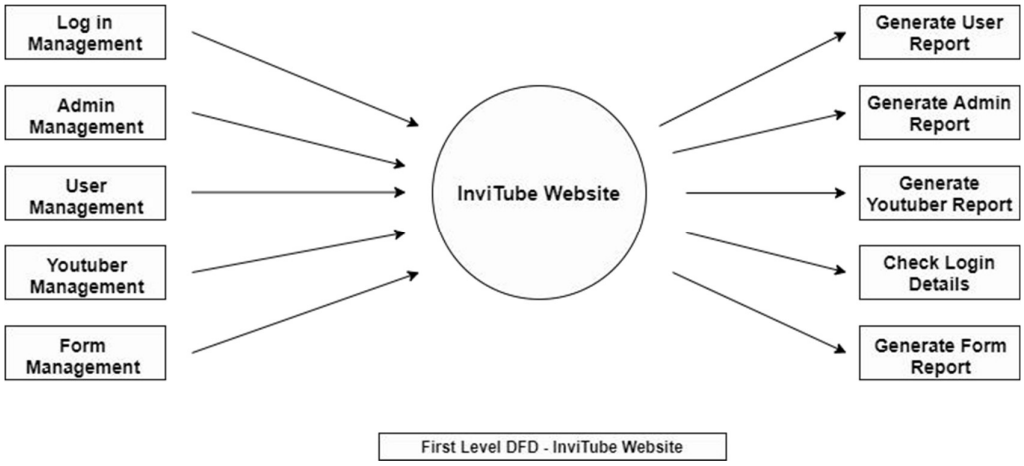

Fig 8.4.2 DFD diagram (Level 1)

DFD Diagram:

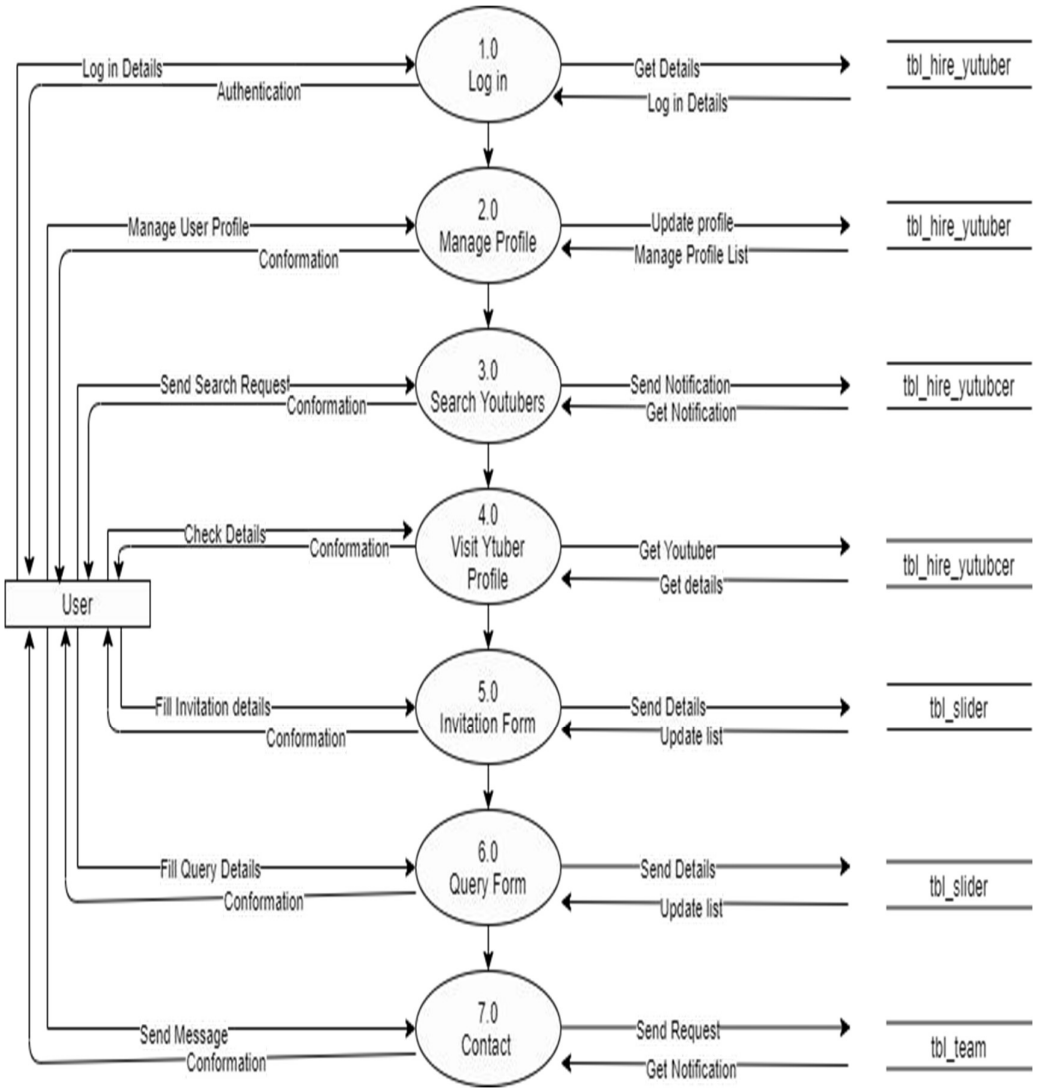

Fig 8.4.3 Context Diagram (Level 1)

8.5 ER Diagram:

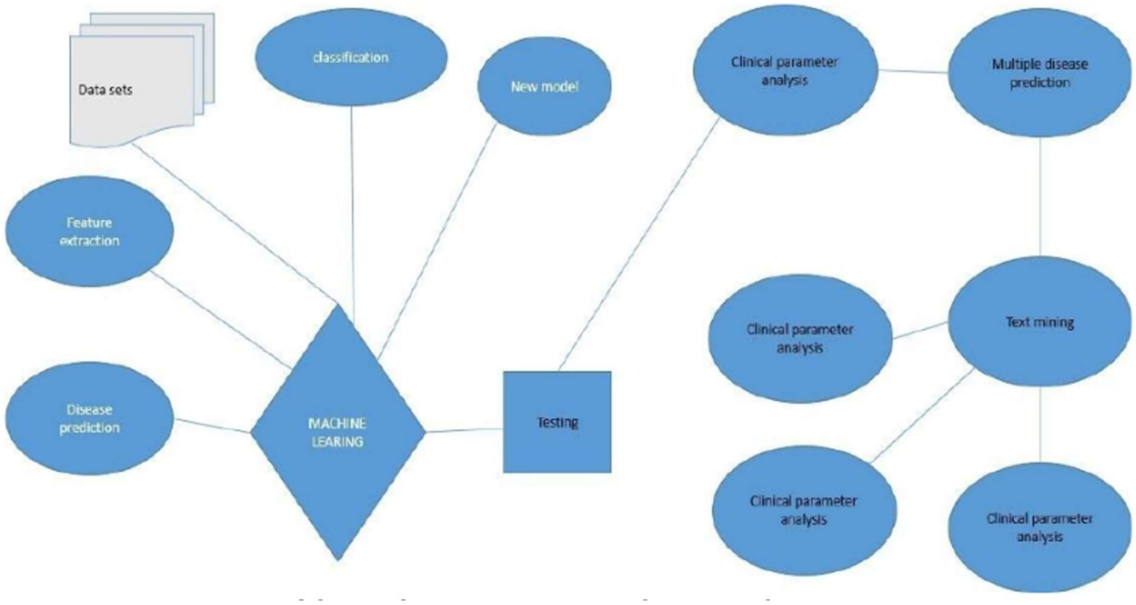

Fig 8.5 ER Diagram once the user has opened the app

8.7 Data Dictionary:

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4920 entries, 0 to 4919
Data columns (total 18 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Disease     4920 non-null   object
 1   Symptom_1   4920 non-null   int64
 2   Symptom_2   4920 non-null   int64
 3   Symptom_3   4920 non-null   int64
 4   Symptom_4   4920 non-null   int64
 5   Symptom_5   4920 non-null   int64
 6   Symptom_6   4920 non-null   int64
 7   Symptom_7   4920 non-null   int64
 8   Symptom_8   4920 non-null   int64
 9   Symptom_9   4920 non-null   int64
 10  Symptom_10  4920 non-null   int64
 11  Symptom_11  4920 non-null   int64
 12  Symptom_12  4920 non-null   int64
 13  Symptom_13  4920 non-null   int64
 14  Symptom_14  4920 non-null   int64
 15  Symptom_15  4920 non-null   int64
 16  Symptom_16  4920 non-null   int64
 17  Symptom_17  4920 non-null   int64
dtypes: int64(17), object(1)
memory usage: 692.0+ KB
```

Fig 8.7.1 data dictionary of symptom dataset

Data Dictionary:

```
data_sev.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 133 entries, 0 to 132
Data columns (total 2 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   Symptom  133 non-null    object
 1   weight   133 non-null    int64
dtypes: int64(1), object(1)
memory usage: 2.2+ KB
```

Fig 8.7.2 severity weightage

Data Dictionary:

```
dataset
```

	Disease	Symptom_1	Symptom_2	Symptom_3	Symptom_4	Symptom_5	Symptom_6	Symptom_7	Symptom_8	Symptom_9	Symptom_10
0	Fungal infection	itching	skin_rash	nodal_skin_eruptions	dischromic _patches	NaN	NaN	NaN	NaN	NaN	NaN
1	Fungal infection	skin_rash	nodal_skin_eruptions	dischromic _patches	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	Fungal infection	itching	nodal_skin_eruptions	dischromic _patches	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	Fungal infection	itching	skin_rash	dischromic _patches	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	Fungal infection	itching	skin_rash	nodal_skin_eruptions	NaN	NaN	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...
4915	(vertigo) Paroymsal Positional Vertigo	vomiting	headache	nausea	spinning_movements	loss_of_balance	unsteadiness	NaN	NaN	NaN	NaN
4916	Acne	skin_rash	pus_filled_pimples	blackheads	scurring	NaN	NaN	NaN	NaN	NaN	NaN
4917	Urinary tract infection	burning_micturition	bladder_discomfort	foul_smell_of urine	continuous_feel_of_urine	NaN	NaN	NaN	NaN	NaN	NaN
4918	Psoriasis	skin_rash	joint_pain	skin_peeling	silver_like_dusting	small_dents_in_nails	inflammatory_nails	NaN	NaN	NaN	NaN
4919	Impetigo	skin_rash	high_fever	blister	red_sore_around_nose	yellow_crust_ooze	NaN	NaN	NaN	NaN	NaN

4920 rows × 18 columns

Fig 8.7.3 symptom database for various diseases

```
dataset
```

	Disease	Symptom_1	Symptom_2	Symptom_3	Symptom_4	Symptom_5	Symptom_6	Symptom_7	Symptom_8	Symptom_9	Symptom_10	Symptom_11
0	Fungal_infection	1	3	4	6	0	0	0	0	0	0	0
1	Fungal_infection	3	4	6	0	0	0	0	0	0	0	0
2	Fungal_infection	1	4	6	0	0	0	0	0	0	0	0
3	Fungal_infection	1	3	6	0	0	0	0	0	0	0	0
4	Fungal_infection	1	3	4	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...
4915	(vertigo)_Paroymsal_Positional_Vertigo	5	3	5	6	4	4	0	0	0	0	0
4916	Acne	3	2	2	2	0	0	0	0	0	0	0
4917	Urinary_tract_infection	6	4	5	6	0	0	0	0	0	0	0
4918	Psoriasis	3	3	3	2	2	2	0	0	0	0	0
4919	Impetigo	3	7	4	2	3	0	0	0	0	0	0

4920 rows × 18 columns

Fig 8.7.4 symptom database for various diseases based on weightage given

CHAPTER 9

LIMITATIONAND FUTURE ENHANCEMENT

- **LIMITATIONS**
- **FUTURE ENHANCEMENT**

9.1 LIMITATIONS

The app may face limitations in terms of accuracy, as despite achieving high prediction rates, misdiagnoses may still occur due to the intricate nature of medical conditions and the variability in symptom manifestations. Additionally, its scope might be restricted, potentially leading to missed diagnoses or reduced predictive capabilities for less common ailments. Reliance on available symptom databases could hamper effectiveness in regions with limited medical data or less developed healthcare infrastructure. Furthermore, the accuracy of predictions heavily hinges on the precision and comprehensiveness of user-provided symptom data, which may fluctuate based on user understanding and adherence. Compliance with healthcare regulations and privacy laws may restrict functionality or impose constraints on data usage and storage. User engagement might be hindered by accessibility issues, interface design, and trust in the app's predictions. Continuous updates, maintenance, and enhancements may necessitate substantial resources, including time, expertise, and financial investment. Ethical considerations surrounding privacy, consent, and potential algorithmic biases may impede the app's acceptance and usage. The interpretability of prediction results may be limited due to the opaque nature of certain machine learning algorithms, making it difficult to discern the reasoning behind predictions. Lastly, the app may inadvertently widen health disparities by favoring certain demographics or neglecting the needs of marginalized populations.

9.2 FUTURE ENHANCEMENTS

Potential future improvements for the disease prediction app may encompass:

- Expanding symptom databases for heightened prediction precision.
- Adding more machine learning algorithms to bolster predictive abilities.
- Ensuring real-time data updates for up-to-date medical information.
- Developing a mobile app variant for enhanced accessibility.
- Including personalized health suggestions based on user data.
- Partnering with healthcare providers for telemedicine services and post-consultation support.
- Broadening the app's functionality for public health surveillance and outbreak prediction.
- Employing advanced data visualization methods for clearer prediction presentation.
- Incorporating multi-language support to accommodate diverse user bases.
- Integrating wearable devices or IoT sensors to gather supplementary health data for precise predictions and tailored advice.

CHAPTER 10

CONCLUSION

- **CONCLUSION**

10.1 CONCLUSION

This project endeavors to forecast diseases based on symptoms. It operates by receiving symptoms from an approver, typically a medical doctor, and generates disease predictions with an average accuracy probability of 95%. The GRAILS system was utilized to successfully integrate the disease predictor.

Additionally, a frontend website was developed to not only predict diseases but also provide precautionary measures and basic disease descriptions. Users can access information on various diseases, especially prevalent ones.

This application can be seamlessly integrated into any hospital website as needed.

CHAPTER 11

Appendices

- **Appendices**

11.1Appendices:

```
dataset = dataset.fillna(0) # put empty cell to 0
dataset = dataset.replace("foul_smell_of_urine" , 5)
dataset = dataset.replace("dischromic__patches" , 6)
dataset = dataset.replace("spotting__urination" , 6)
x = dataset.iloc[:, 1:]
y = dataset.iloc[:, 0]
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)
dtc = DecisionTreeClassifier(criterion='entropy', random_state=42)
dtc.fit(x_train, y_train)
return dtc
```

Fig 11.1.1 Decision Tree Training Dataset

```
rfc = RandomForestClassifier(criterion='entropy', random_state=42)
rfc.fit(x_train, y_train)
rfc_y_pred = rfc.predict(x_test)
print("Random Forest Classifier accuracy score:", accuracy_score(rfc_y_pred, y_test))
```

Fig 11.1.2 Random forest Training Dataset and getting its accuracy

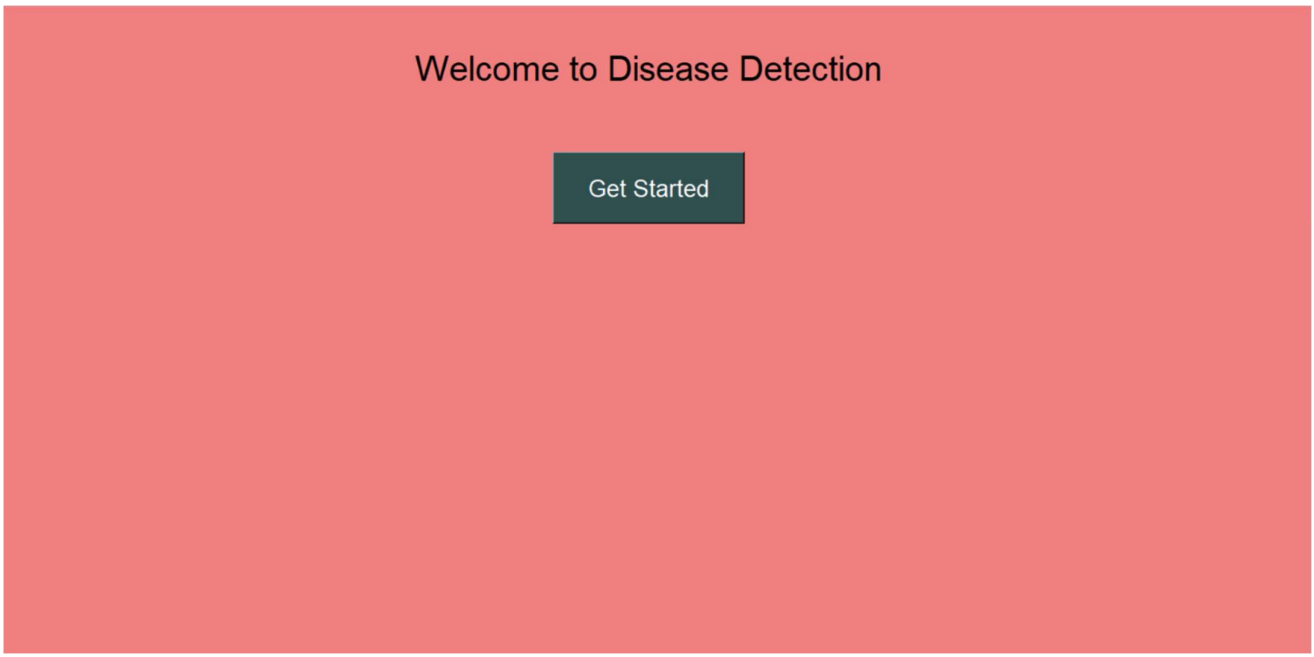

Fig 11.1.3 Home Page

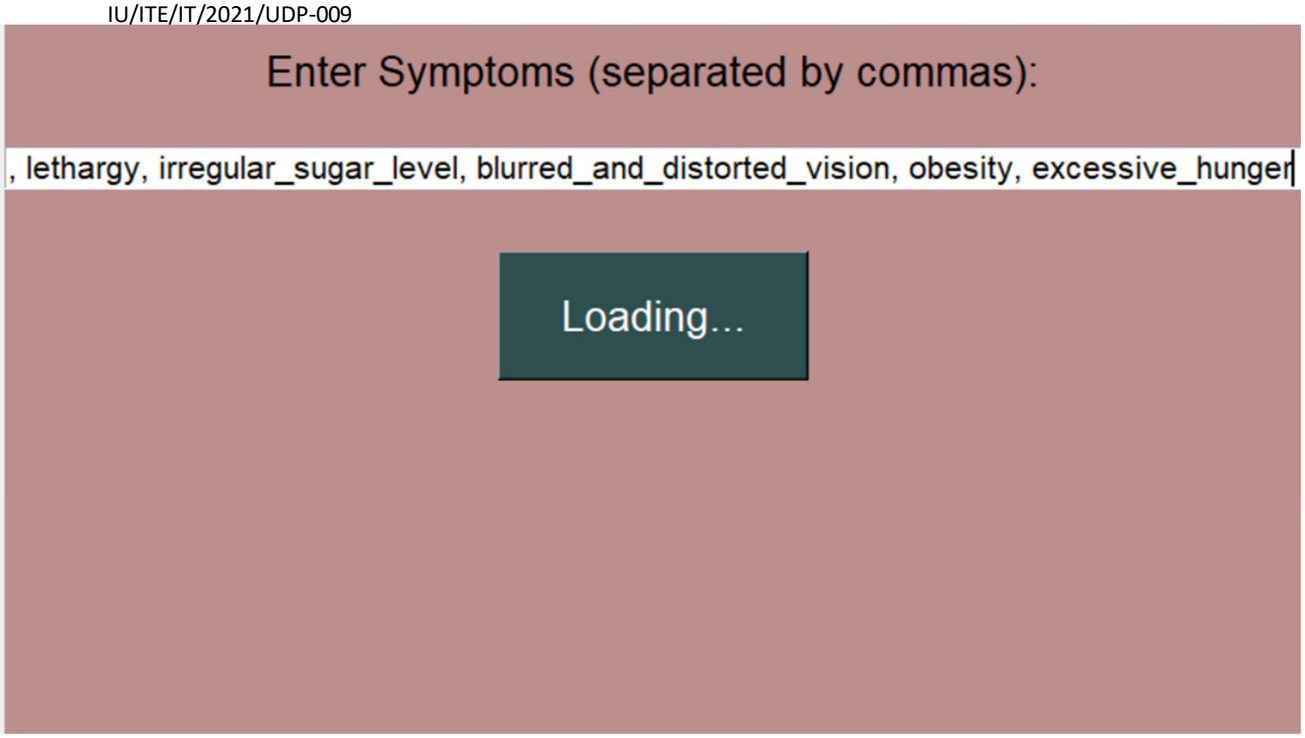

Fig 11.1.4 Loading Page

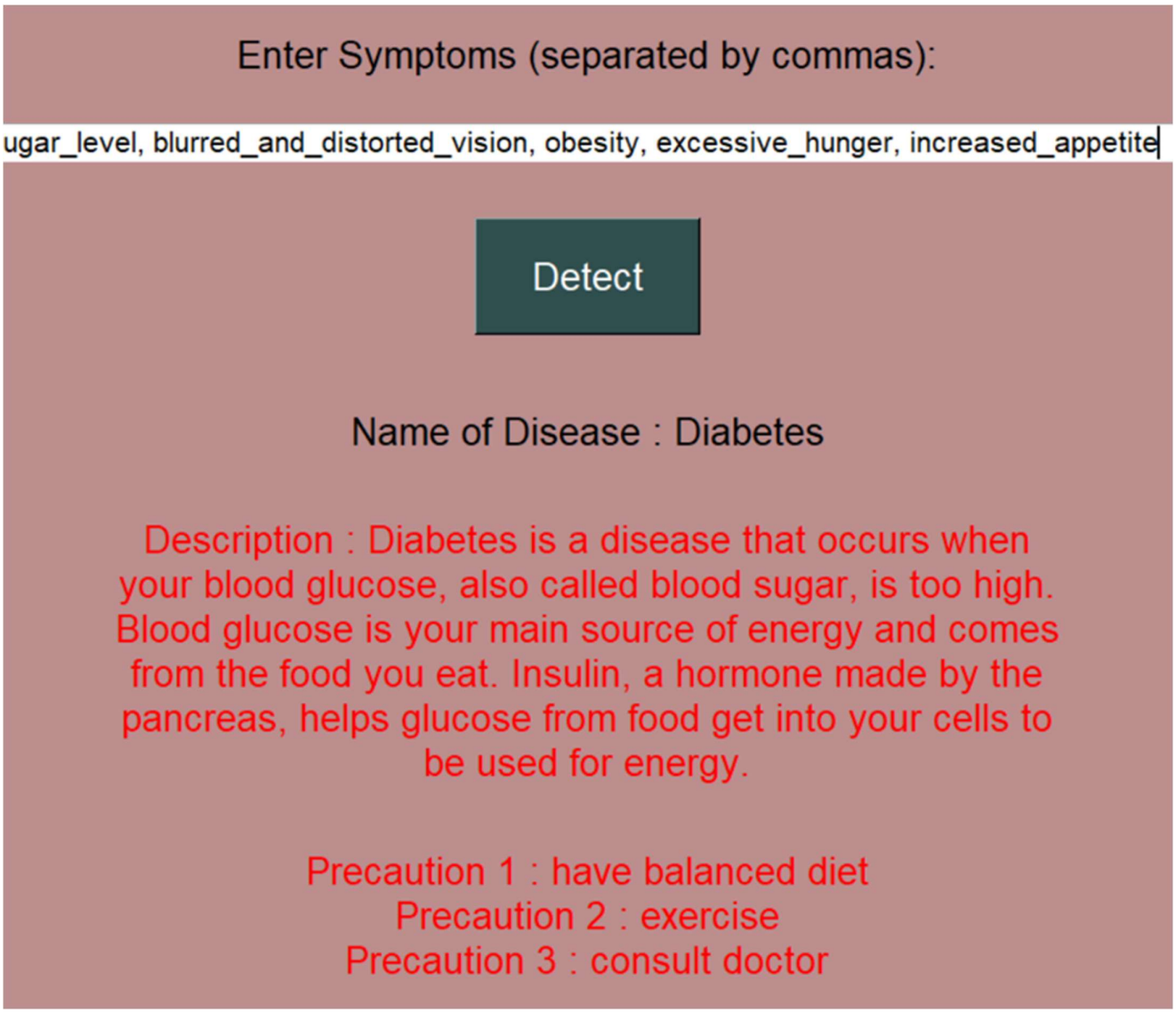

Fig 11.1.5 Result Page

BIBLIOGRAPHY

REFERENCES

https://sist.sathyabama.ac.in/sist_naac/documents/1.3.4/1822-b.e-cse-batchno-18.pdf

https://ijcrt.org/papers/IJCRT2105229.pdf

https://www.researchgate.net/publication/372859217_DISEASE_PREDICTION_WEB-APP_USING_MACHINE_LEARNING_A_SENIOR_YEAR_PROJECT_REPORT

https://www.javatpoint.com/python-tkinter

https://www.tutorialspoint.com/python/python_gui_programming.htm

https://www.geeksforgeeks.org/introduction-to-tkinter/