

# **Implementation of AI-Powered Medical Diagnosis System**

A Project Report

submitted in partial fulfillment of the requirements

of

AICTE Internship on AI: Transformative Learning  
with

TechSaksham – A joint CSR initiative of Microsoft & SAP

by

**Rupak Sarkar**

**rupaksarkar1102@gmail.com**

Under the Guidance of

**Saomya Chaudhury**

## ACKNOWLEDGEMENT

---

Bringing this AI-powered medical diagnostic system to life has been a challenging yet rewarding journey, and I am deeply grateful to those who have supported me along the way.

First and foremost, I would like to express my sincere gratitude to my instructor, **Saomya Chaudhury**, for their invaluable guidance, encouragement, and expertise throughout this project. Their insights and mentorship have been instrumental in shaping my understanding and execution of this system.

I also want to extend my appreciation to the healthcare professionals and researchers whose work provided the foundation for this project. Their contributions to the medical and technological fields have been an inspiration, and their insights have helped me develop a more effective and reliable system.

A special thank you to my family and friends for their unwavering support and encouragement during this journey. Your belief in me kept me motivated through every challenge and breakthrough.

Lastly, to the users and medical professionals who provided valuable feedback—thank you. Your insights have helped refine this system to ensure it truly makes a difference in healthcare.

This project has been a testament to my passion for technology and its potential to improve lives. I am proud of what I have accomplished and excited for the future possibilities it holds.

Thank you!

## ABSTRACT

---

AI is changing the way we approach medical diagnostics, making it faster, more accurate, and more accessible. This project focuses on building an AI-powered system that can analyze medical data—like patient records, images, and lab results—to help doctors make better decisions. By using machine learning and deep learning, the system can quickly detect patterns and provide diagnostic predictions, supporting healthcare professionals in their work.

The goal is to improve early disease detection, reduce errors, and streamline the diagnostic process. Developing this system involved data processing, training AI models, and testing accuracy to ensure reliability. Ethical concerns, patient data privacy, and ease of use were also major considerations in designing a system that fits seamlessly into real medical settings.

The results show that AI has the potential to revolutionize medical diagnostics by enhancing human expertise and improving patient outcomes. While challenges like data bias and regulatory requirements still exist, this project highlights how AI can be a game-changer in healthcare, making diagnostics smarter and more efficient.



## TABLE OF CONTENT

---

<b>Abstract</b>	.....	I
<b>Chapter 1.</b>	<b>Introduction .....</b>	<b>1</b>
1.1	Problem Statement .....	1
1.2	Motivation .....	1
1.3	Objectives.....	2
1.4.	Scope of the Project .....	2
<b>Chapter 2.</b>	<b>Literature Survey .....</b>	<b>4</b>
<b>Chapter 3.</b>	<b>Proposed Methodology .....</b>	<b>8</b>
<b>Chapter 4.</b>	<b>Implementation and Results .....</b>	<b>11</b>
<b>Chapter 5.</b>	<b>Discussion and Conclusion .....</b>	<b>20</b>
<b>References.....</b>		<b>23</b>



## LIST OF FIGURES

Figure No.	Figure Caption	Page No.
<b>Figure 1</b>	Importing Dependencies	<b>11</b>
<b>Figure 2</b>	Data Collection and Processing	<b>11</b>
<b>Figure 3</b>	Getting info about datasheet	<b>11</b>
<b>Figure 4</b>	Statistical Measure and Distribution	<b>12</b>
<b>Figure 5</b>	Splitting the Features and Targets	<b>12</b>
<b>Figure 6</b>	Train and Test Split	<b>12</b>
<b>Figure 7</b>	Model Training – Logistic Regression	<b>13</b>
<b>Figure 8</b>	Model Evaluation and Accuracy Score	<b>13</b>
<b>Figure 9</b>	Building Predictive System	<b>13</b>
<b>Figure 10</b>	Saving Trained Model	<b>13</b>
<b>Figure 11</b>	Library Imports	<b>14</b>
<b>Figure 12</b>	Webpage Configuration, Hiding Streamlit UI, Adding BG Image	<b>14</b>
<b>Figure 13</b>	Loading Saved Models.	<b>14</b>
<b>Figure 14</b>	Dropdown Menu Creation	<b>15</b>
<b>Figure 15</b>	Diabetes Prediction Webpage Creation	<b>15</b>
<b>Figure 16</b>	Diabetes Detection Webpage (Result Figure)	<b>16</b>
<b>Figure 17</b>	Diabetes Prediction Result. (Result Figure)	<b>17</b>
<b>Figure 18</b>	Heart Disease Detection Webpage (Result Figure)	<b>17</b>
<b>Figure 19</b>	Hypo-Thyroid Detection Webpage (Result Figure)	<b>18</b>
<b>Figure 20</b>	Hypo-Thyroid Prediction Result (Result Figure)	<b>19</b>

## LIST OF TABLES

Table. No.	Table Caption	Page No.
<b>Table 1</b>	Hardware Requirements	<b>10</b>
<b>Table 2</b>	Software Requirements	<b>10</b>

## CHAPTER 1

### Introduction

#### 1.1 Problem Statement:

Medical misdiagnosis and delayed detection of diseases remain significant challenges in healthcare, often leading to severe consequences for patients. Traditional diagnostic methods rely heavily on human expertise, which, while invaluable, can be affected by factors such as fatigue, limited access to specialists, and variations in interpretation. Additionally, the increasing volume of medical data makes it difficult for healthcare professionals to analyze information efficiently and accurately.

With advancements in artificial intelligence (AI), there is an opportunity to enhance diagnostic accuracy and efficiency. However, implementing AI-powered medical diagnostic systems comes with its own set of challenges, including data quality, ethical considerations, integration with existing medical workflows, and ensuring trust among healthcare professionals.

This project aims to address these challenges by developing an AI-powered diagnostic system that leverages machine learning and deep learning techniques to assist in disease detection and diagnosis. The system is designed to analyze medical data quickly and provide reliable insights, ultimately supporting doctors in making more informed decisions and improving patient outcomes.

#### 1.2 Motivation:

The increasing prevalence of medical misdiagnosis and delayed disease detection presents a critical challenge in healthcare. Accurate and timely diagnosis is essential for effective treatment, yet traditional diagnostic methods are often time-consuming, prone to human error, and dependent on the availability of experienced medical professionals. With the growing volume of medical data, there is an urgent need for more efficient and reliable diagnostic solutions.

Artificial intelligence (AI) has the potential to revolutionize healthcare by enhancing diagnostic accuracy, reducing the burden on medical professionals, and improving patient outcomes. AI-powered systems can quickly analyze vast amounts of medical data, identify patterns, and provide valuable insights that support doctors in making informed decisions. This not only speeds up the diagnostic process but also minimizes errors and ensures early disease detection.

The motivation behind this project is to harness the power of AI to develop a reliable, efficient, and accessible medical diagnostic system. By integrating machine learning and deep learning techniques, this system aims to bridge the gap between technology and

healthcare, ultimately improving the quality of medical diagnostics and making healthcare more accessible to all.

### **1.3 Objective:**

**Develop an AI-Powered Diagnostic System** – Design and implement a machine learning-based system that can analyze medical data and assist in disease detection.

**Improve Diagnostic Accuracy** – Enhance the precision of medical diagnoses by leveraging AI algorithms to reduce human errors and inconsistencies.

**Optimize Diagnosis Speed** – Enable faster analysis of medical records, imaging data, and lab results to support timely decision-making in healthcare.

**Ensure Data Privacy & Ethical Compliance** – Incorporate strict data security measures and adhere to ethical standards to protect patient information.

**Enhance Accessibility** – Create a system that can be used by healthcare professionals regardless of location, improving access to quality diagnostics in remote and underserved areas.

**Integrate with Existing Medical Systems** – Ensure that the AI-powered diagnostic system can seamlessly work alongside traditional medical workflows and electronic health records (EHRs).

**Evaluate Performance & Reliability** – Conduct rigorous testing and validation to assess the system's accuracy, efficiency, and usability in real-world medical settings.

### **1.4 Scope of the Project:**

The implementation of an AI-powered medical diagnostic system aims to enhance the accuracy, efficiency, and accessibility of disease diagnosis. The scope of this project includes the development, testing, and evaluation of an AI-based system capable of analyzing medical data to assist healthcare professionals in making informed decisions.

#### **In-Scope:**

1. **Disease Detection & Diagnosis** – The system will focus on identifying patterns in medical data, including patient records, imaging scans, and laboratory results, to provide diagnostic insights.
2. **Machine Learning & Deep Learning Integration** – The project will utilize AI techniques such as neural networks, decision trees, and statistical models to improve diagnostic accuracy.
3. **Data Preprocessing & Analysis** – Medical data will be cleaned, structured, and analyzed to train the AI model for better predictions.

4. **User-Friendly Interface** – A simple and intuitive interface will be designed for easy interaction with healthcare professionals.
5. **Performance Evaluation** – The system will be tested against existing diagnostic methods to assess its accuracy, reliability, and efficiency.
6. **Security & Ethical Considerations** – The project will follow data privacy regulations and ethical guidelines to ensure patient confidentiality and responsible AI use.

**Out of Scope:**

1. **Replacing Healthcare Professionals** – The AI system is designed to assist doctors, not replace them. Final medical decisions will remain with human experts.
2. **Treatment & Prescription Recommendations** – The system will focus only on diagnosis and will not suggest treatments or prescribe medications.
3. **Integration with Hospital Systems** – While the project aims for potential integration, direct implementation into existing hospital databases and EHRs is not included in this phase.
4. **Handling Rare or Uncommon Diseases** – The system will initially be trained on widely studied diseases and may not perform well on rare conditions due to data limitations.



## CHAPTER 2

# Literature Survey

### 2.1 Review relevant literature or previous work in this domain.

The use of artificial intelligence (AI) in medical diagnostics has been a growing area of research, with various studies highlighting its potential to enhance accuracy, efficiency, and accessibility in healthcare. This literature survey explores existing research and advancements in AI-powered medical diagnostic systems, focusing on key methodologies, challenges, and real-world applications.

#### 1. AI in Medical Diagnostics

Several studies have demonstrated the effectiveness of AI in analyzing medical images, patient records, and laboratory results. Deep learning models, such as convolutional neural networks (CNNs), have been widely used in radiology for detecting diseases like pneumonia, tumors, and fractures with accuracy comparable to human radiologists (Esteva et al., 2017). Additionally, natural language processing (NLP) techniques have been employed to extract insights from electronic health records (EHRs) to aid in diagnosis (Rajkomar et al., 2019).

#### 2. Machine Learning and Deep Learning Techniques

Research has shown that machine learning models, including decision trees, support vector machines (SVMs), and neural networks, play a significant role in medical diagnostics. CNNs have been particularly successful in image-based diagnosis, while recurrent neural networks (RNNs) and transformer models are effective in analyzing time-series medical data. Studies such as Gulshan et al. (2016) demonstrated AI's potential in detecting diabetic retinopathy with high accuracy, reducing the need for manual screening.

#### 3. AI for Early Disease Detection

Early diagnosis of diseases such as cancer, cardiovascular conditions, and neurological disorders has been a major focus of AI research. AI-powered diagnostic tools can detect abnormalities at an early stage, allowing for timely intervention. A study by Ardila et al. (2019) showed that deep learning models could detect lung cancer from CT scans more accurately than human radiologists, highlighting AI's potential in early disease detection.

#### **4. Challenges and Ethical Considerations**

Despite AI's promising advancements in medical diagnostics, several challenges remain. Issues such as data bias, model interpretability, regulatory concerns, and patient data privacy must be addressed for AI to be widely accepted in clinical practice (Amann et al., 2020). Additionally, AI systems must be continuously validated against diverse datasets to ensure their reliability across different populations.

#### **5. Future Prospects and Applications**

As AI technology continues to evolve, researchers are exploring the integration of AI with Internet of Medical Things (IoMT) and cloud-based systems to improve remote diagnostics and telemedicine. The combination of AI with wearable health devices has also shown promise in real-time health monitoring and personalized medicine (Topol, 2019).

#### **2.2Mention any existing models, techniques, or methodologies related to the problem.**

##### i. ECgMPL: AI Model for Endometrial Cancer Detection

A collaborative research team developed ECgMPL, an AI model that analyzes histopathological images to detect endometrial cancer with 99.26% accuracy. This model also shows high accuracy in diagnosing colorectal, breast, and oral cancers, aiding clinical decision-making.

##### ii. Apollo Hospitals' AI Initiatives

Apollo Hospitals in India is investing in AI to reduce the workload of medical staff by automating tasks like medical documentation. Their AI tools assist in patient diagnoses, test interpretations, treatment suggestions, and transcription of doctors' notes, aiming to improve efficiency and patient care.

##### iii. Google's AI in Medical Imaging

Google Health is advancing AI-enabled imaging research to improve disease detection and assist in developing treatment plans. Their research focuses on enhancing diagnostic accuracy and efficiency through AI applications in medical imaging.

##### iv. Spectral AI's Wound Diagnosis Tools

Spectral AI has developed tools that utilize advanced machine learning algorithms to swiftly analyze wound images, providing precise assessments. These AI-powered tools can identify

infections, evaluate burn depth, and monitor healing progress more efficiently than traditional methods.

v. Philips' AI-Enhanced Imaging Systems

Philips is integrating AI into healthcare by developing advanced medical devices like AI-powered MRI and CT scanners. These innovations aim to enhance diagnostic speed and accuracy, alleviating pressure on health systems and improving patient care.

These projects demonstrate AI's significant impact on medical diagnostics, offering promising advancements in disease detection and healthcare delivery.

## **2.3 Highlight the gaps or limitations in existing solutions and how your project will address them.**

- ECgMPL: AI Model for Endometrial Cancer Detection

While ECgMPL demonstrates high accuracy (99.26%), it primarily relies on histopathological images, which means its effectiveness depends on image quality, dataset diversity, and staining techniques used in pathology labs. Additionally, the model's real-world deployment may face challenges related to variability in medical imaging standards, potential bias in training data, and the need for expert validation before clinical use.

- Apollo Hospitals' AI Initiatives

Apollo's AI-driven automation reduces workload, but its effectiveness depends on data quality, interoperability with hospital systems, and regulatory compliance. A key limitation is the risk of misinterpretation in automated documentation, where AI may misclassify symptoms or medical conditions, leading to errors in decision-making. Moreover, AI-driven suggestions must be validated by medical experts, slowing adoption.

- Google's AI in Medical Imaging

Google Health's imaging AI is promising but faces challenges in real-world generalization. The model may struggle with data bias, where AI trained on one demographic may not perform as well for another. Integration with hospital

infrastructure and regulatory approval for clinical use also pose barriers. Additionally, AI-driven imaging requires extensive validation to avoid false positives or false negatives that could lead to misdiagnosis.

- Spectral AI's Wound Diagnosis Tools

Spectral AI's wound assessment models rely on machine learning applied to image data, which means their accuracy depends on consistent lighting, resolution, and imaging angles. Differences in skin tones, wound severity, and environmental factors may lead to misinterpretations. Additionally, clinical integration and physician trust remain hurdles, as AI-based wound assessments require manual validation.

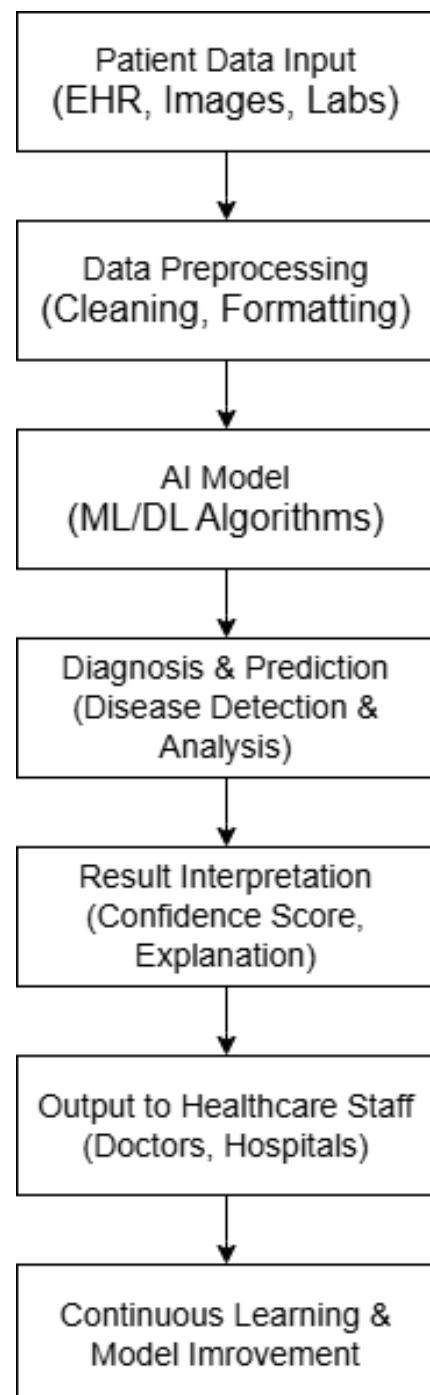
- Philips' AI-Enhanced Imaging Systems

Philips' AI-powered MRI and CT innovations improve diagnostic speed, but high costs and hardware dependency make them inaccessible in resource-limited settings. Another major challenge is data privacy and security, as AI-driven imaging collects vast amounts of sensitive patient data. Moreover, interpretability remains an issue, as radiologists may struggle to understand how AI reaches its conclusions.

## CHAPTER 3

### Proposed Methodology

#### 3.1 System Design



The above diagram represents the **workflow of our AI-driven healthcare diagnosis system**. It outlines the different stages involved in processing patient data, making predictions, and delivering insights to healthcare professionals. Here's a step-by-step explanation of each stage:

#### 1. Patient Data Input (EHR, Images, Labs)

- This is the initial stage where patient data is collected from various sources such as **Electronic Health Records (EHR)**, **medical images (X-rays, MRIs, etc.)**, and **lab test results**.

#### 2. Data Preprocessing (Cleaning, Formatting)

- The raw data is often noisy or unstructured, so it goes through **cleaning, normalization, and formatting** to make it suitable for AI models.

#### 3. AI Model (ML/DL Algorithms)

- Machine Learning (ML) or Deep Learning (DL) algorithms process the structured data to learn patterns and detect potential diseases.

#### 4. Diagnosis & Prediction (Disease Detection & Analysis)

- The trained AI model **analyzes** patient data and predicts potential diseases or health conditions.

#### 5. Result Interpretation (Confidence Score, Explanation)

- AI models often generate a **confidence score** to indicate the certainty of the prediction.
- Explanation techniques (like SHAP values or LIME) can be used to **provide insights into the prediction**.

#### 6. Output to Healthcare Staff (Doctors, Hospitals)

- The results are shared with **doctors, hospitals, or medical staff**, enabling them to make informed decisions.

#### 7. Continuous Learning & Model Improvement

- The AI system continuously **learns from new data**, improving its accuracy and performance over time.

## 3.2 Requirement Specification

### 3.2.1 Hardware Requirements:

Component	Minimum Requirement	Recommended Requirement
<b>Processor</b>	Intel i5/AMD Ryzen 5	Intel i7/AMD Ryzen 7
<b>RAM</b>	8GB	16GB
<b>Storage (Models)</b>	10GB	20GB
<b>GPU</b>	Not Needed	Not Needed
<b>OS</b>	Win 10/11, macOS, Linux	Win 10/11, macOS, Linux

### 3.2.2 Software Requirements:

Software	Purpose
<b>Python</b>	Core programming language for development
<b>Streamlit</b>	Framework for building an interactive UI
<b>Scikit-learn</b>	Machine Learning library for ranking algorithms
<b>Jupyter Notebook</b>	For Data Analysis, Model Training and Visualization

## CHAPTER 4

### Implementation and Result

Importing the Dependencies

```
[1]: import sklearn
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import StandardScaler
```



**Explanation** – Here, we load the dependencies that will be required for the model training.

Data Collection and Processing

```
[ ]: # Loading the csv data to a Pandas DataFrame
heart_data = pd.read_csv('/content/heart_disease_data.csv')

[ ]: # print first 5 rows of the dataset
heart_data.head()

200...   age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  slope  ca  thal  target
0    63    1    3     145   233    1     0    150     0     2.3    0    0    1    1
1    37    1    2     130   250    0     1    187     0     3.5    0    0    2    1
2    41    0    1     130   204    0     0    172     0     1.4    2    0    2    1
3    56    1    1     120   236    0     1    178     0     0.8    2    0    2    1
4    57    0    0     120   354    0     1    163     1     0.6    2    0    2    1

[ ]: # print last 5 rows of the dataset
heart_data.tail()

201...   age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  slope  ca  thal  target
298   57    0    0     140   241    0     1    123     1     0.2    1    0    3    0
299   45    1    3     110   264    0     1    132     0     1.2    1    0    3    0
300   68    1    0     144   193    1     1    141     0     3.4    1    2    3    0
301   57    1    0     130   131    0     1    115     1     1.2    1    1    3    0
302   57    0    1     130   236    0     0    174     0     0.0    1    1    2    0
```

**Explanation** – In this screenshot, we input the datasheets in CSV format for data reading and preprocessing.

```
[ ]: # getting some info about the data
heart_data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   age         303 non-null    int64  
 1   sex          303 non-null    int64  
 2   cp           303 non-null    int64  
 3   trestbps    303 non-null    int64  
 4   chol         303 non-null    int64  
 5   fbs          303 non-null    int64  
 6   restecg     303 non-null    int64  
 7   thalach      303 non-null    int64  
 8   exang        303 non-null    int64  
 9   oldpeak      303 non-null    float64 
 10  slope        303 non-null    int64  
 11  ca           303 non-null    int64  
 12  thal         303 non-null    int64  
 13  target       303 non-null    int64  
dtypes: float64(1), int64(13)
memory usage: 33.3 KB
```

**Explanation** – Here, we get some info about the inputted CSV file.

```
[ ]: # statistical measures about the data
heart_data.describe()

[205...    age   sex   cp  trestbps   chol   fbs  restecg  thalach  exang  oldpeak  slope   ca   thal
count  303.000000 303.000000 303.000000 303.000000 303.000000 303.000000 303.000000 303.000000 303.000000 303.000000 303.000000 303.000000 303.000000 303.000000
mean   54.366337  0.683168  0.966997 131.623762 246.264026  0.148515  0.528053 149.646865  0.326733  1.039604  1.399340  0.729373  2.313531  0
std    9.082101  0.466011  1.032052 17.538143 51.830751  0.356198  0.525860 22.905161  0.469794  1.161075  0.616226  1.022606  0.612277  0
min   29.000000  0.000000  0.000000 94.000000 126.000000  0.000000  0.000000 71.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0
25%   47.500000  0.000000  0.000000 120.000000 211.000000  0.000000  0.000000 133.500000  0.000000  0.000000  1.000000  0.000000  2.000000  0
50%   55.000000  1.000000  1.000000 130.000000 240.000000  0.000000  1.000000 153.000000  0.000000  0.800000  1.000000  0.000000  2.000000  1
75%   61.000000  1.000000  2.000000 140.000000 274.500000  0.000000  1.000000 166.000000  1.000000  1.600000  2.000000  1.000000  3.000000  1
max   77.000000  1.000000  3.000000 200.000000 564.000000  1.000000  2.000000 202.000000  1.000000  6.200000  2.000000  4.000000  3.000000  1

[ ]: # checking the distribution of Target Variable
heart_data['target'].value_counts()

[206... 1    165
0    138
Name: target, dtype: int64
1 --> Defective Heart
0 --> Healthy Heart
```

**Explanation –** In this screenshot, we check the Statistical measures of the data and distribution of target variables.

Splitting the Features and Target

```
[ ]: X = heart_data.drop(columns='target', axis=1)
Y = heart_data['target']

[ ]: print(X)

      age   sex   cp  trestbps   chol   fbs  restecg  thalach  exang  oldpeak  \
0     63     1     3     145   233     1     0     150     0     2.3
1     37     1     2     130   250     0     1     187     0     3.5
2     41     0     1     130   204     0     0     172     0     1.4
3     56     1     1     120   236     0     1     178     0     0.8
4     57     0     0     120   354     0     1     163     1     0.6
..   ...
298    57     0     0     140   241     0     1     123     1     0.2
299    45     1     3     110   264     0     1     132     0     1.2
300    68     1     0     144   193     1     1     141     0     3.4
301    57     1     0     130   131     0     1     115     1     1.2
302    57     0     1     130   236     0     0     174     0     0.0

      slope   ca   thal
0       0     0     1
1       0     0     2
2       2     0     2
3       2     0     2
4       2     0     2
..   ...
298    1     0     3
299    1     0     3
300    1     2     3
301    1     1     3
302    1     1     2

[303 rows x 13 columns]
```

**Explanation –** Splitting features and targets of the inputted data.

Splitting the Data into Training data & Test Data

```
[ ]: X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, stratify=Y, random_state=2)

[ ]: print(X.shape, X_train.shape, X_test.shape)
(303, 13) (242, 13) (61, 13)
```

**Explanation –** Here, we split the data into Training and Testing.

**Model Training****Logistic Regression**

```
model = LogisticRegression(max_iter=800)

# training the LogisticRegression model with Training data
model.fit(X_train, Y_train)
```

LogisticRegression(max\_iter=800)

**Explanation** – The model training is done in the above screenshot.

**Model Evaluation****Accuracy Score**

```
# accuracy on training data
X_train_prediction = model.predict(X_train)
training_data_accuracy = accuracy_score(X_train_prediction, Y_train)

print('Accuracy on Training data : ', training_data_accuracy)
Accuracy on Training data :  0.8553719008264463

# accuracy on test data
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)

print('Accuracy on Test data : ', test_data_accuracy)
Accuracy on Test data :  0.8032786885245902
```

**Explanation** – The Model evaluation and Accuracy scores are tested in the above screenshot.

**Building a Predictive System**

```
input_data = (57,0,0,120,354,0,1,163,1,0.6,2,0,2)

# change the input data to a numpy array
input_data_as_numpy_array= np.asarray(input_data)
print(input_data_as_numpy_array.dtype)

# reshape the numpy array as we are predicting for only one instance
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)
print(input_data_reshaped.dtype)

prediction = model.predict(input_data_reshaped)
print(prediction)

if (prediction[0]== 0):
    print('The Person does not have a Heart Disease')
else:
    print('The Person has Heart Disease')

float64
float64
[1]
The Person has Heart Disease
```

**Explanation** – A Predictive System is built in the above section.

**Saving the trained model**

```
import pickle

filename = 'heart_disease_model.sav'
pickle.dump(model, open(filename, 'wb'))

# Loading the saved model
loaded_model = pickle.load(open('heart_disease_model.sav', 'rb'))
```

**Explanation** – The Trained model is now saved in a (.sav) file.

From now on, all the screenshots will be of the main file, which I have named as **app.py**

```
import streamlit as st
import pickle
from streamlit_option_menu import option_menu
```

**Libraries Imported:** Uses streamlit, pickle, and streamlit\_option\_menu for UI.

```
# Change Name & Logo
st.set_page_config(page_title="Disease Prediction", page_icon="⚕️")

# Hiding Streamlit add-ons
hide_st_style = """
<style>
MainMenu {visibility: hidden;}
footer {visibility: hidden;}
header {visibility: hidden;}
</style>
"""
st.markdown(hide_st_style, unsafe_allow_html=True)

# Adding Background Image
background_image_url = "https://www.strategyand.pwc.com/m1/en/strategic-foresight/sector-strategies/healthcare/ai-powered-healthcare-solutions/img01-section1.jpg" # Replace with your image URL
```

- **Page Configuration:** Sets the title as "Disease Prediction" and adds a medical symbol (⚕️) as the favicon.
- **Hiding Streamlit UI Elements:** The main menu, footer, and header are hidden.
- **Adding a Background Image:** A healthcare-related image is set as the background.

```
# Load the saved models
models = {
    'diabetes': pickle.load(open('Models/diabetes_model.sav', 'rb')),
    'heart_disease': pickle.load(open('Models/heart_disease_model.sav', 'rb')),
    'parkinsons': pickle.load(open('Models/parkinsons_model.sav', 'rb')),
    'lung_cancer': pickle.load(open('Models/lungs_disease_model.sav', 'rb')),
    'thyroid': pickle.load(open('Models/Thyroid_model.sav', 'rb'))
}
```

Five **pre-trained models** are loaded from the Models/ directory using pickle.load():

- diabetes\_model.sav
- heart\_disease\_model.sav
- parkinsons\_model.sav
- lungs\_disease\_model.sav
- Thyroid\_model.sav



```
# Create a dropdown menu for disease prediction
selected = st.selectbox(
    'Select a Disease to Predict',
    ['Diabetes Prediction',
     'Heart Disease Prediction',
     'Parkinsons Prediction',
     'Lung Cancer Prediction',
     'Hypo-Thyroid Prediction']
)
```

Users can **select a disease** from a dropdown (st.selectbox).

The available options are:

1. Diabetes Prediction
2. Heart Disease Prediction
3. Parkinson's Prediction
4. Lung Cancer Prediction
5. Hypo-Thyroid Prediction

```
# Diabetes Prediction Page
if selected == 'Diabetes Prediction':
    st.title('Diabetes')
    st.write("Enter the following details to predict diabetes:")

    Pregnancies = display_input('Number of Pregnancies', 'Enter number of times pregnant', 'Pregnancies', 'number')
    Glucose = display_input('Glucose Level', 'Enter glucose level', 'Glucose', 'number')
    BloodPressure = display_input('Blood Pressure value', 'Enter blood pressure value', 'BloodPressure', 'number')
    SkinThickness = display_input('Skin Thickness value', 'Enter skin thickness value', 'SkinThickness', 'number')
    Insulin = display_input('Insulin Level', 'Enter insulin level', 'Insulin', 'number')
    BMI = display_input('BMI value', 'Enter Body Mass Index value', 'BMI', 'number')
    DiabetesPedigreeFunction = display_input('Diabetes Pedigree Function value', 'Enter diabetes pedigree function value',
                                              'DiabetesPedigreeFunction', 'number')
    Age = display_input('Age of the Person', 'Enter age of the person', 'Age', 'number')

    diab_diagnosis = ''
    if st.button('Diabetes Test Result'):
        diab_prediction = models['diabetes'].predict([[Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction,
                                                       Age]])
        diab_diagnosis = 'The person is diabetic' if diab_prediction[0] == 1 else 'The person is not diabetic'
        st.success(diab_diagnosis)
```

Each disease has:

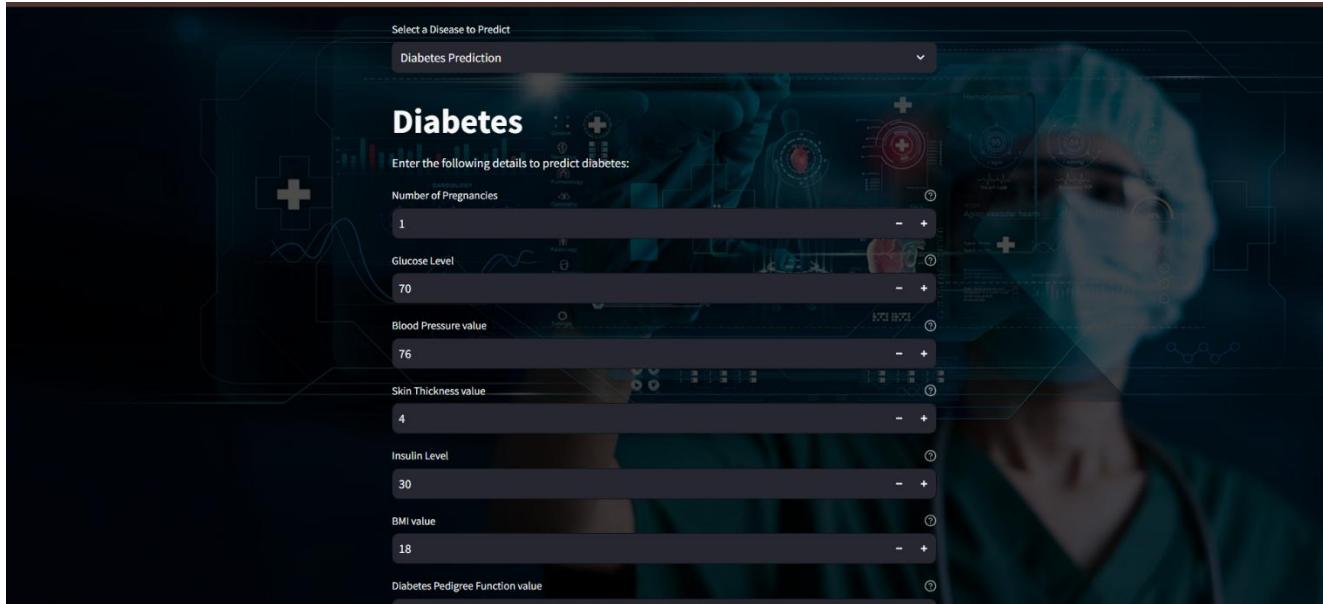
1. A title (st.title()) and input fields (st.number\_input()).
2. A **button to make a prediction** using the respective model.
3. A **diagnosis output** based on the model's prediction.

For example:

- **Diabetes Prediction**
  - Inputs: Pregnancies, Glucose, Blood Pressure, BMI, etc.
  - Model: models['diabetes']
  - Prediction Output: "The person is diabetic" or "The person is not diabetic".

Similarly, we have 4 more pages loaded like this.

## 4.1 Snap Shots of Result:



Select a Disease to Predict  
Diabetes Prediction

**Diabetes**

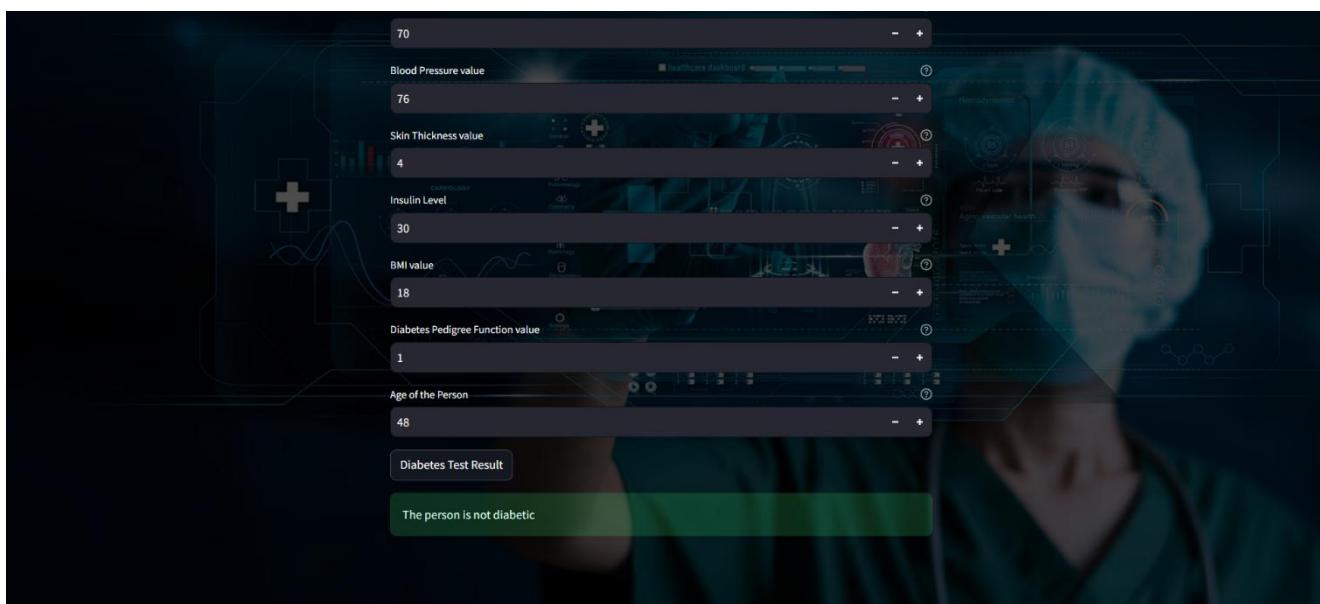
Enter the following details to predict diabetes:

Number of Pregnancies	1	- +
Glucose Level	70	- +
Blood Pressure value	76	- +
Skin Thickness value	4	- +
Insulin Level	30	- +
BMI value	18	- +
Diabetes Pedigree Function value	0	- +

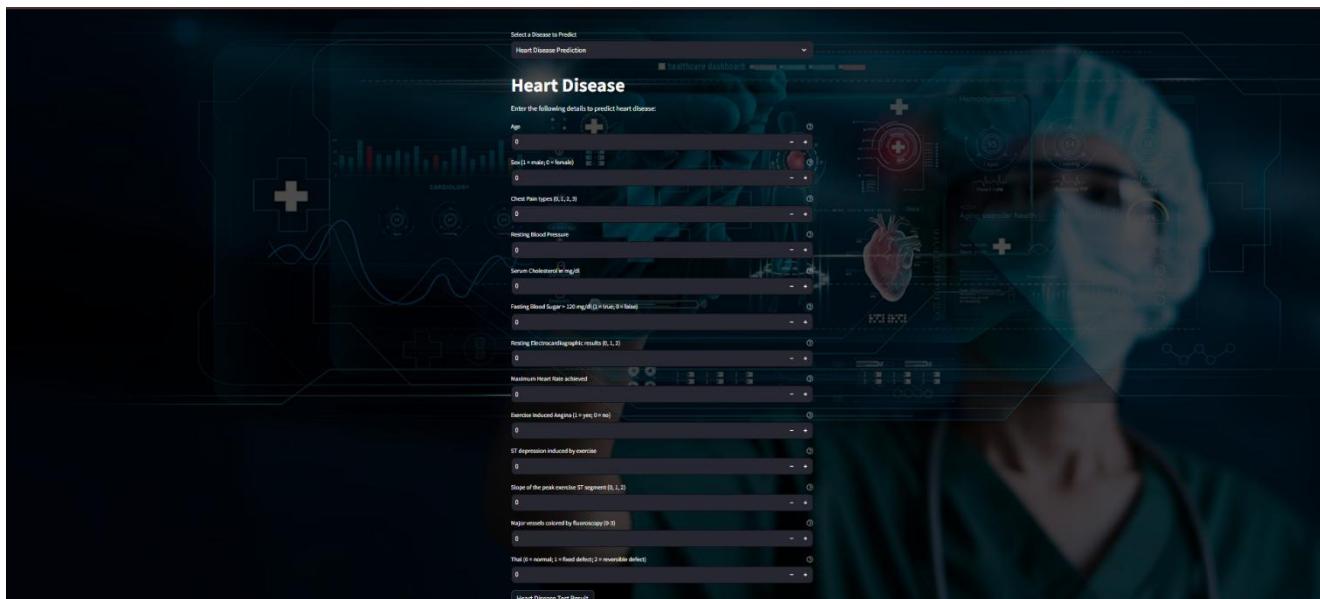
In the above screenshot, we have our webpage, and on the very top we have implemented the availability for testing different kinds of diseases a person can acquire.

For this, we are choosing to test whether a patient is Diabetic or not. Next, we have a list of input fields –

- **Number of Pregnancies**
- **Glucose Level**
- **Blood Pressure Value**
- **Skin Thickness Value**
- **Insulin Level**
- **BMI** (Body Mass Index)
- **Diabetes Pedigree Function** (which considers genetic influence)
- **Age of the person**



After we input values on the fields, we can click the button at the bottom of the fields. And then, the model will give us the result of the test based on the values inputted.



In this Screenshot, we are checking whether a patient has **Heart Disease** or not.

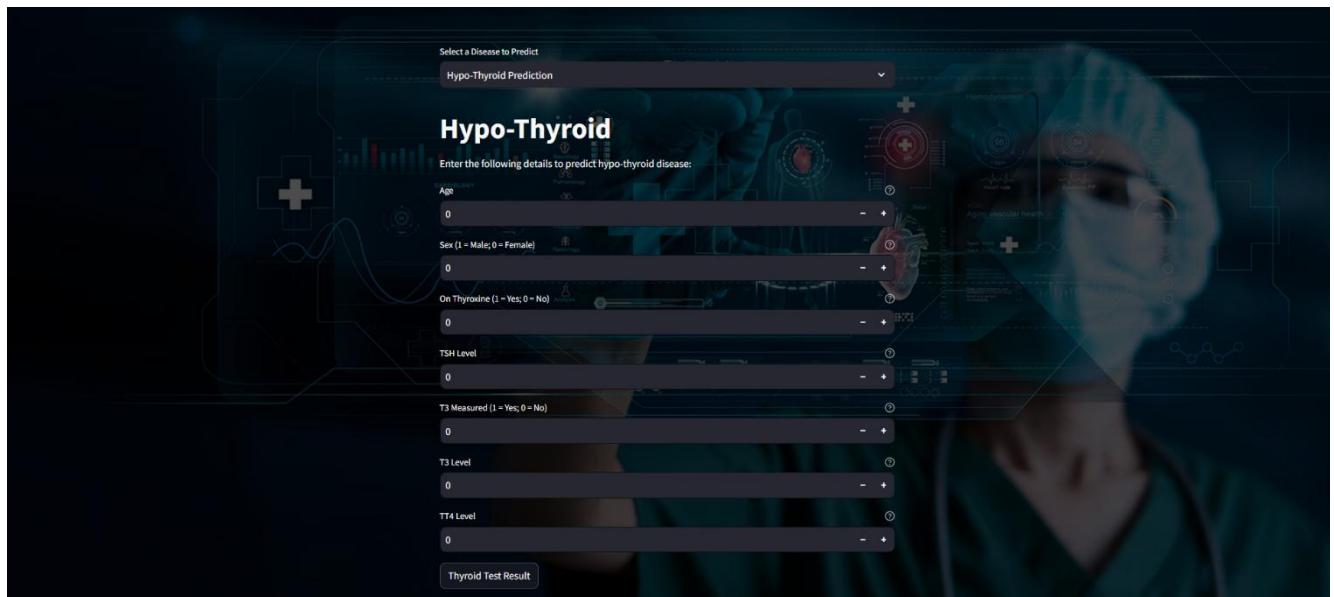
As the earlier screenshots, there are several input fields that are required to fill for the checkup.

- **Age**
- **Sex** (1 = male, 0 = female)
- **Chest Pain Type** (values 0 to 3)
- **Resting Blood Pressure**

- **Serum Cholesterol Level** (mg/dL)
- **Fasting Blood Sugar** ( $> 120 \text{ mg/dL} \rightarrow 1 = \text{true}, 0 = \text{false}$ )
- **Resting Electrocardiographic Results** (values 0, 1, 2)
- **Maximum Heart Rate Achieved**
- **Exercise-Induced Angina** (1 = yes, 0 = no)
- **ST Depression Induced by Exercise**
- **Slope of Peak Exercise ST Segment** (0, 1, 2)
- **Major Vessels Colored by Fluoroscopy** (values 0 to 3)
- **Thalassemia Test Result** (0 = normal, 1 = fixed defect, 2 = reversible defect)

After we have inputted all the fields, we have a button labeled "**Heart Disease Test Result**".

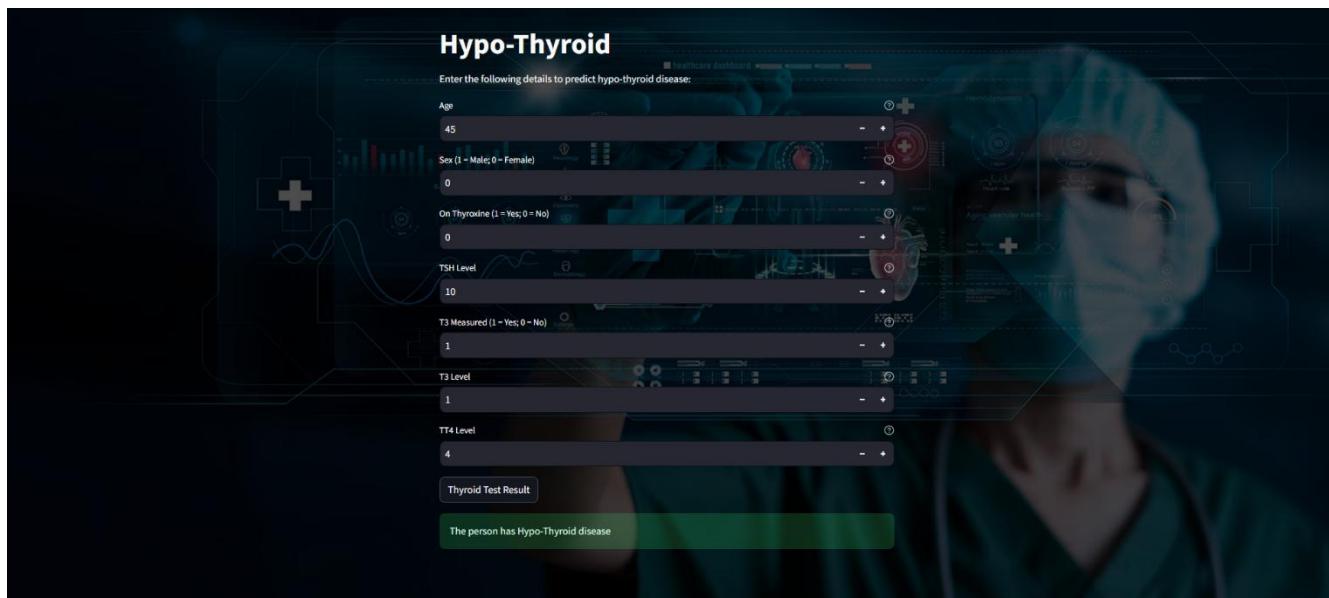
After we press the button, we will get the result whether the patient has heart disease or not.



Next, we have the Hypo-Thyroid test. Same like the earlier ones we have several input fields which will help us predict whether the patient has hypo-thyroid or not.

- **Age**
- **Sex** (1 = Male, 0 = Female)
- **On Thyroxine** (1 = Yes, 0 = No) – Indicates if the patient is on thyroid medication.
- **TSH Level** (Thyroid-Stimulating Hormone level)
- **T3 Measured** (1 = Yes, 0 = No) – Indicates if the T3 level has been measured.
- **T3 Level** (Triiodothyronine level)

- **T4 Level** (Thyroxine level)



In the above screenshot, we have inputted the values for all the fields and then pressed the final button, and our model based on the given inputs has predicted that the patient has hypo-thyroid.

#### 4.2 GitHub Link for Code:

<https://github.com/ReaveND/Medical-Diagnosis-with-AI>

## CHAPTER 5

### Discussion and Conclusion

#### 5.1 Future Work:

To improve my AI-based disease prediction model and address potential unresolved issues, I have considered the following enhancements:

##### 1. Model Performance & Accuracy

- **Feature Engineering:** Identify the most influential features using techniques like SHAP (SHapley Additive Explanations) to enhance model interpretability and reduce unnecessary inputs.
- **Hyperparameter Tuning:** Use **Grid Search** or **Bayesian Optimization** to optimize parameters for better performance.
- **Ensemble Learning:** Combine multiple models (e.g., Random Forest, XGBoost, Neural Networks) to increase predictive accuracy and reduce bias.
- **Cross-validation:** Implement **k-fold cross-validation** to ensure the model generalizes well across different datasets.

##### 2. Data Quality & Preprocessing

- **Handling Missing Data:** Instead of dropping missing values, use imputation techniques such as **KNN imputation** or **predictive modeling**.
- **Address Class Imbalance:** Use **SMOTE (Synthetic Minority Over-sampling Technique)** or weighted loss functions to handle imbalanced datasets.
- **Anomaly Detection:** Identify and remove outliers that may distort the model's learning process.

##### 3. Interpretability & Explainability

- **AI Explainability Techniques:** Implement SHAP or LIME to explain individual predictions and help medical professionals trust AI decisions.

- **Confidence Score Display:** Provide a confidence score or probability to indicate how certain the model is about a prediction.

#### 4. Expanding Disease Coverage

- **Multi-Disease Prediction:** Extend the system to predict multiple diseases at once, rather than a single disease at a time.
- **Continuous Learning:** Implement an adaptive learning approach where the model improves with newly collected real-world data.

#### 5. UI/UX Improvements

- **User-Friendly Interface:** Enhance the web UI with interactive visualizations (graphs, charts) for better result interpretation.
- **Mobile Compatibility:** Consider making the system accessible on mobile devices for wider usability.

#### 6. Deployment & Scalability

- **Cloud Integration:** Deploy the model on cloud platforms (AWS, GCP, Azure) to handle real-time predictions at scale.
- **API Development:** Expose the model as an API to allow integration with hospital management systems.

#### 7. Ethical & Security Considerations

- **Data Privacy Compliance:** Ensure compliance with **HIPAA (Health Insurance Portability and Accountability Act)** or **GDPR** for patient data protection.
- **Bias Mitigation:** Regularly evaluate the model for biases against specific demographics and adjust training data accordingly.

#### 8. Future Research Directions

- **Deep Learning Integration:** Explore CNNs or Transformers for feature extraction in complex datasets like ECG or MRI scans.

- **Federated Learning:** Implement **federated learning** to train models on distributed datasets across hospitals without sharing sensitive data.

## 5.2 Conclusion:

### Summary of Impact and Contribution of the Project

This project leverages AI and machine learning to enhance disease prediction, contributing significantly to the healthcare sector by providing **early detection, faster diagnosis, and improved decision-making**. The model enables medical professionals and patients to obtain **quick and data-driven insights**, reducing dependency on traditional, time-consuming diagnostic methods.

Key contributions include:

- **Improved Diagnostic Accuracy** – The AI model helps detect diseases like heart disease and hypothyroidism with high precision.
- **Efficient Patient Screening** – Automates preliminary screening, allowing doctors to focus on critical cases.
- **User-Friendly Interface** – Provides an intuitive and interactive dashboard for seamless data input and result interpretation.
- **Scalability & Integration** – The system can be integrated with electronic health records (EHRs) and hospital management systems for real-world deployment.
- **Ethical & Data Security Considerations** – Ensures patient data privacy while minimizing biases in medical AI predictions.

### Overall Impact:

By combining **machine learning with healthcare**, this project enhances disease prevention, reduces healthcare costs, and increases accessibility to medical diagnostics. It serves as a foundation for future AI-driven medical applications, contributing to **more efficient, data-driven, and personalized healthcare solutions**.



## REFERENCES

- [1] **Mitchell, T. (1997).** Machine Learning. McGraw Hill. – A fundamental resource on machine learning techniques.
- [2] **Bishop, C. M. (2006).** Pattern Recognition and Machine Learning. Springer. – Covers statistical techniques for pattern recognition.
- [3] **Dey, N., Ashour, A. S., & Borra, S. (Eds.). (2019).** Machine Learning in Bio-Signal Analysis and Diagnostic Imaging. Academic Press.
- [4] **Rajkomar, A., Dean, J., & Kohane, I. (2019).** Machine Learning in Medicine. New England Journal of Medicine, 380(14), 1347-1358. – Discusses AI applications in medical diagnosis.
- [5] **Esteva, A., Chou, K., Yeung, S., Naik, N., Madani, A., & Mottaghi, A. (2021).** Deep Learning-Enabled Medical Computer Vision. Nature Biomedical Engineering. – Covers how AI enhances image-based medical diagnosis.
- [6] **Shickel, B., Tighe, P. J., Bihorac, A., & Rashidi, P. (2018).** Deep EHR: A Survey of Recent Advances in Deep Learning Techniques for Electronic Health Record (EHR) Analysis. Journal of Biomedical Informatics, 83, 275-292.
- [7] WHO (World Health Organization) – AI in Healthcare – <https://www.who.int/>
- [8] National Institute of Health (NIH) – Machine Learning for Medical Diagnostics – <https://www.nih.gov/>
- [9] Kaggle Datasets – Heart Disease and Thyroid Disease Prediction – <https://www.kaggle.com/>