RAMAIAH
Institute of Technology

MediMind: A Comprehensive Health Prediction and Record Keeping Platform

medimind

By

Aditya sharma(1ms20cs008)

Danish Mahajan(1ms20cs037)

Dhruv khandelwal(1ms20cs044)

Arpit gajya(1ms20cs149)

Under the Guidance of DARSHANA A .NAIK

PROBLEM STATEMENT

MediMind is a platform designed to keep comprehensive health records of individuals and provide personalized health predictions based on their data. It aims to revolutionize the healthcare industry by utilizing advanced technologies such as AI and machine learning to analyze data and predict potential health issues before they occur. MediMind aims to improve patient outcomes and reduce healthcare costs by enabling early detection and intervention of potential health problems.

INTRODUCTION

MediMind is a comprehensive record-keeping and disease prediction platform that aims to revolutionize the healthcare industry by leveraging advanced technologies to improve patient outcomes. The platform provides a range of features that enable patients and healthcare providers to manage patient data, track disease progression, and predict the likelihood of future health events.

AGENDA

- **Early diagnosis**: MediMind can be used to identify patients who are at risk of developing a particular disease, even before the onset of clinical symptoms. Early diagnosis can enable early treatment, which can improve patient outcomes and reduce healthcare costs.
- **Accurate record keeping**: The primary objective of MediMind is to maintain accurate and up-to-date records of patient health information, including medical history .This information is used by healthcare providers to make informed decisions about patient care.

SCHEDULE PLANNING

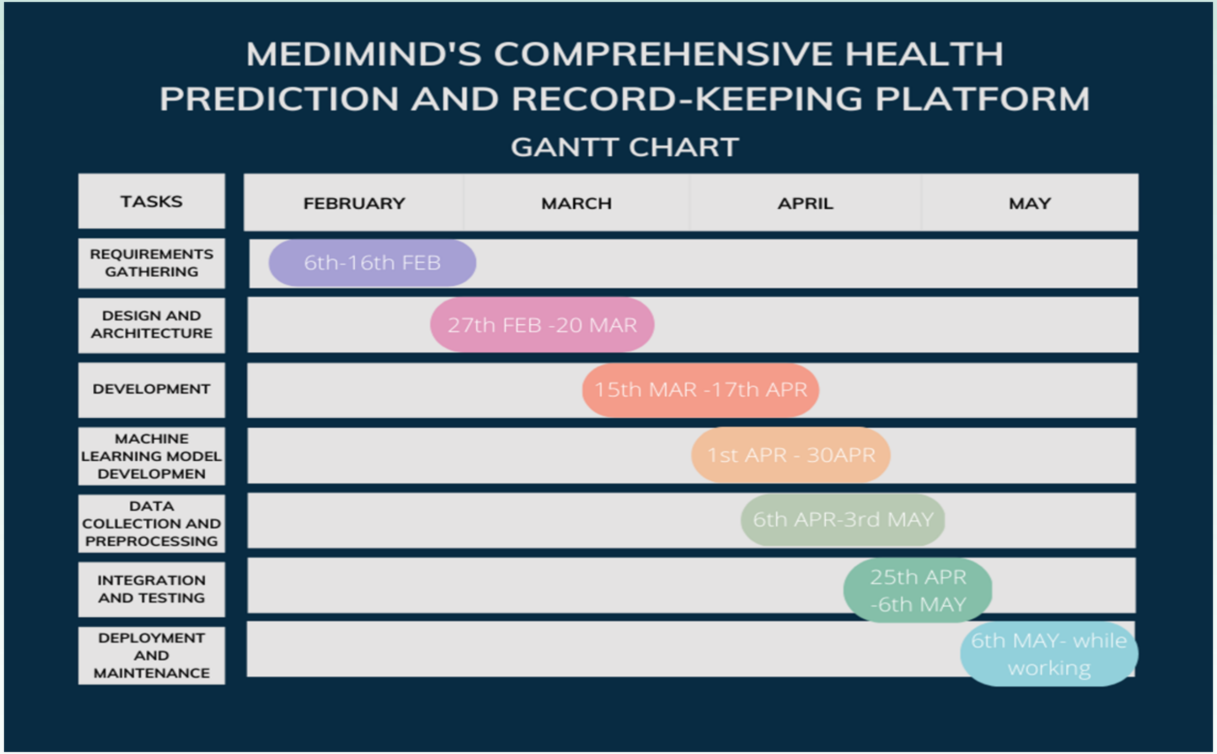


Fig 1 Gantt Chart of our Project Planning

LITERATURE SURVEY

Title, Author and Year	Technique/Methodology used	Result	Remarks
Diagnosis of skin diseases in the era of deep learning and mobile technology By Ahmed Elaziz, Moustafa M. Ghanem, Islam A. Tolba, Mohamed N. Mostafa, Mohammed A	convolutional neural networks (CNNs) and classifier is a support vector machine (SVM)	The proposed mobile application achieved an accuracy of 94.76% in identifying skin diseases using deep learning algorithms.	The research gap is the lack of studies on the integration of mobile technology and deep learning for skin disease diagnosis in low-resource settings.
A Method Of Skin Disease Detection Using Image Processing And Machine Learning Nawal Soliman ALKolifi ALEnezi	convolutional neural networks (CNNs)	The proposed method achieved an accuracy of 93.87% in detecting skin diseases using image processing and machine learning.	The proposed method includes preprocessing the image to improve its quality, segmenting the skin lesion, extracting features from the segmented image, and using a machine learning algorithm to classify the skin disease
Diagnosis of skin diseases using Convolutional Neural Networks Jainesh Rathod, Vishal Aniru Sodha, Praseniit Bhavathankar	convolutional neural networks (CNNs)	The authors achieved an accuracy of 70% in identifying 7 different skin diseases using the proposed Convolutional Neural Network (CNN) model. The CNN model also outperformed other traditional machine learning models.	the research gap lies in the limited dataset used for training and testing the model, which may not generalize well to diverse populations and skin types.

Title, Author and Year	Technique/Methodology used	Result	Remarks
Prediction of Heart Disease Using a Combination of Machine Learning and Deep Learning Rohit Bharti, Aditya Khamparia, Mohammad Shabaz, Gaurav Dhiman, Sagar Pande, and Parneet Singh	Logistic Regression, KNeighborsClassifier, DecisionTreeClassifier. RandomForestClassifier.	The authors obtained good accuracy in predicting heart disease using a combination of machine learning and deep learning techniques. The proposed model outperformed several other state-of-the-art models, indicating its potential for clinical use	The paper proposes a model that combines machine learning and deep learning techniques for the prediction of heart disease. However, the study did not compare the proposed model with existing state-of-the-art models.
A machine intelligence technique for predicting cardiovascular disease (CVD) using Radiology K. Saikumar and V. Rajesh	regions with convolutional neural networks (R-CNN)	The authors achieved high accuracy in predicting CVD using radiology images with the proposed machine intelligence technique. The study suggests that the technique can be used as an effective and non-invasive tool for CVD prediction.	The paper presents a machine intelligence technique for predicting cardiovascular disease (CVD) using radiology, achieving an accuracy of 86.54%.
Predictive Classifier for Cardiovascular Disease Based on Stacking Model Fusion Jimin Liu,Xueyu DongORCID,Huiqi Zhao and Yinhua Tian	stacking model fusion, from SVM, KNN, LR, RF, ET, GBDT, XGBoost, LightGBM, CatBoost, and MLP	The authors achieved an accuracy of 90.3% in predicting cardiovascular disease using the stacking model fusion approach. The proposed method outperformed other traditional machine learning algorithms.	The paper proposes a predictive classifier for cardiovascular disease (CVD) based on stacking model fusion. The research gap is the need for accurate and efficient CVD prediction tools to help prevent and manage this common health condition.

TECHNOLOGIES USED AT EACH STAGE

PHASE I

HTML

Designing the UI/UX using Figma, and using React Framework in the Frontend. Bootstrap is used to design the components of the Frontend. We are using react to create components

In this phase, we are using the technologies such as node.js, express.js for implementing the backend and to store the information of Volunteers using mongoDB. Using JWT we implement the authentication of the user.

PHASE II

mongo DB

OVERALL VIEW OF THE PROJECT IN TERMS OF IMPLEMENTATION

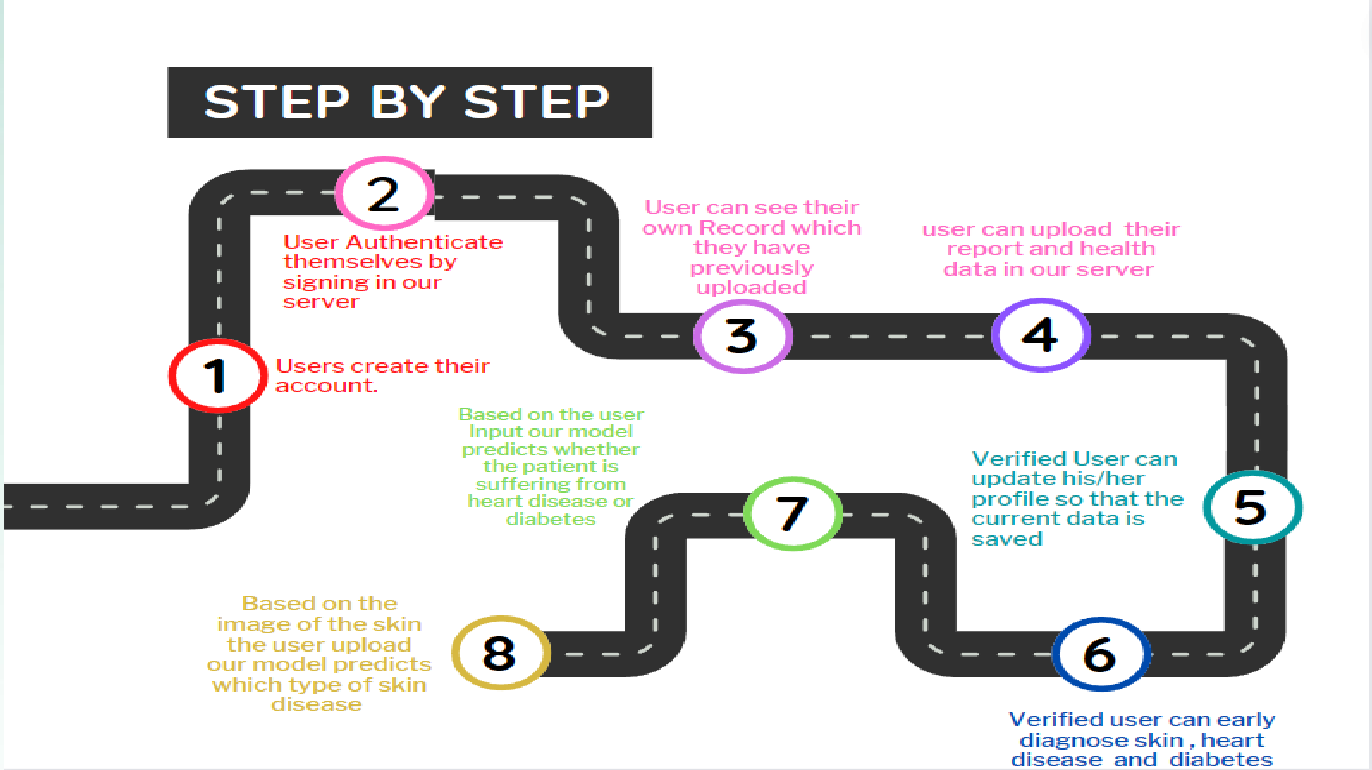


Fig 2 Step by step guide for the user

SYSTEM ARCHITECTURE

Client-server architecture, the software system is divided into two main components: the client and the server. The client is typically a user interface that allows users to interact with the system, while the server provides the core functionality and stores the data.

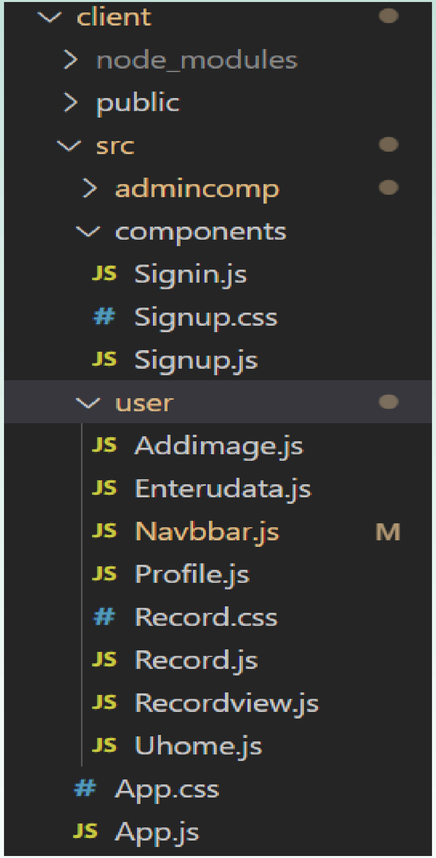


Fig 3 It shows the client File Structure

SERVER

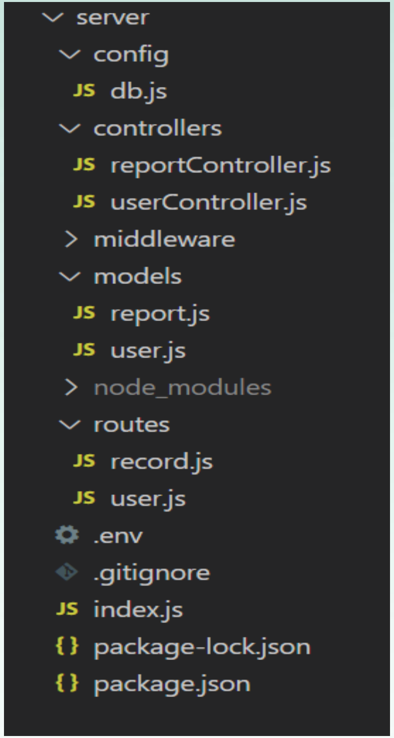


Fig 4 It shows the Server Architecture

This is a Server Architecture which includes different module to provide scalability to the project The server is divided into models which contains the schema of the user and the record How they are stored in the database

The route module shows what are the routes that our project support such as http://localhose:8000/api/user/signin for signIn of the user

Similarly the config file contains the code for connectivity with the database

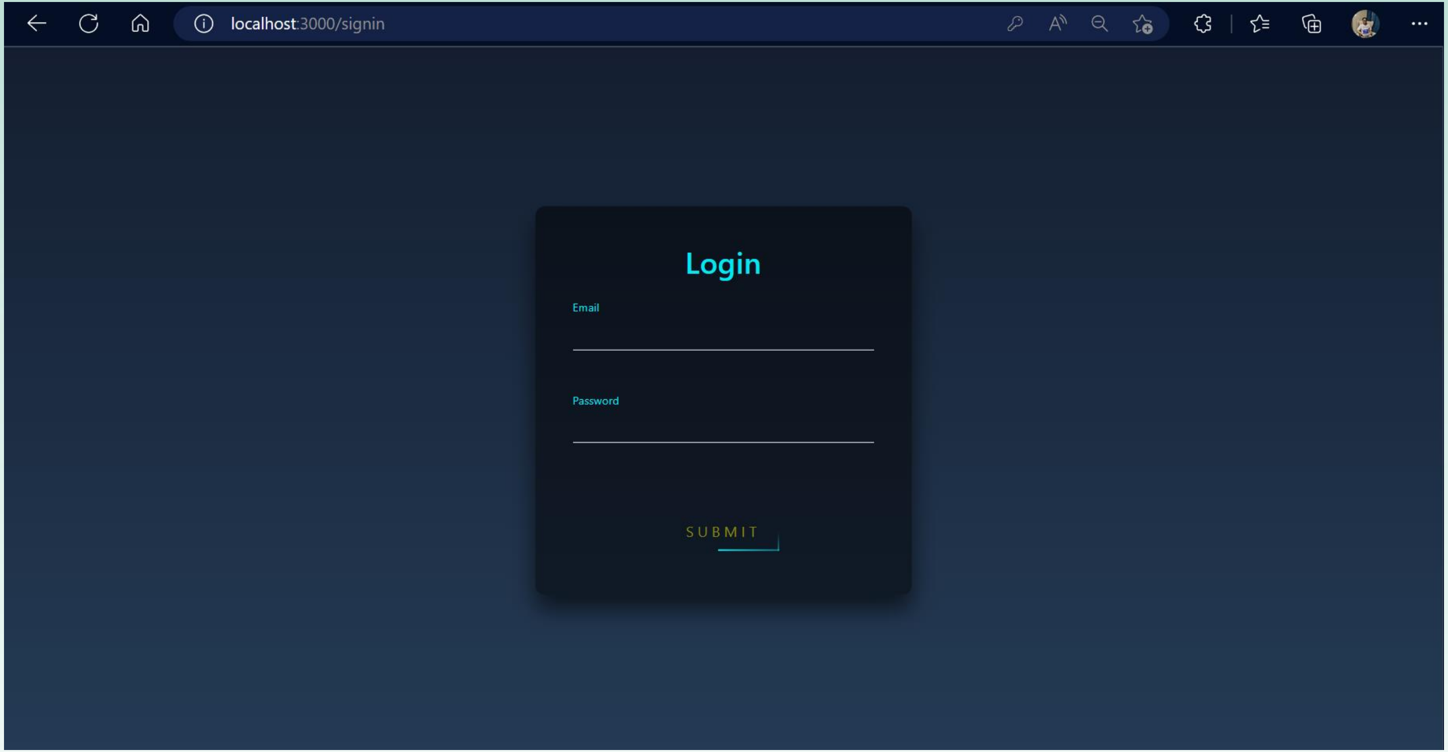
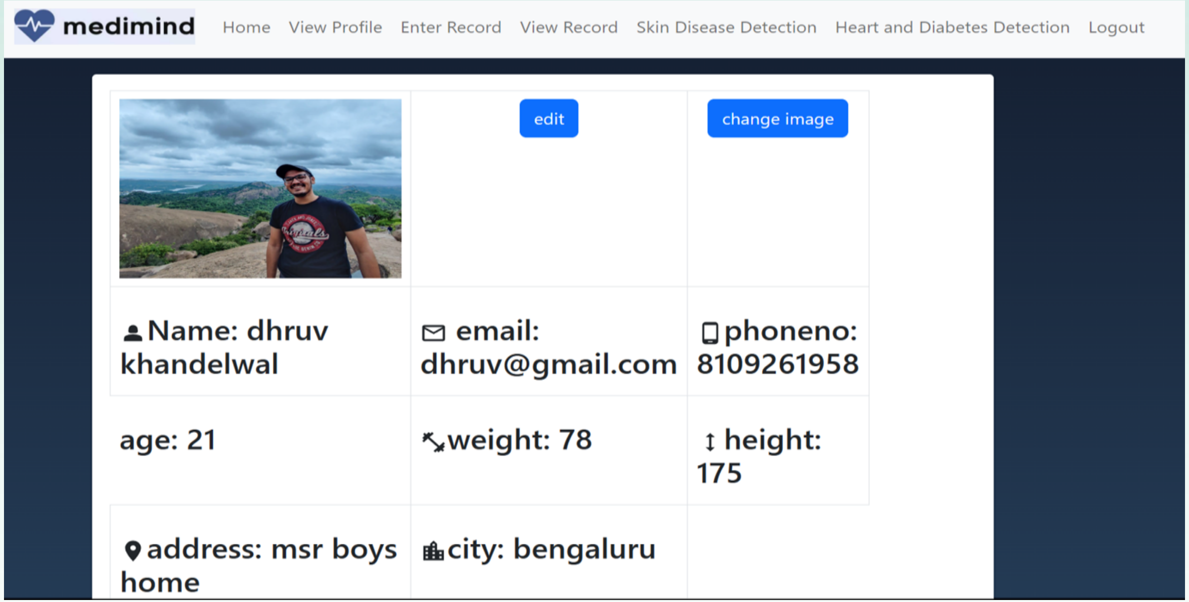


FIG 5.This is a signup / login page for the user to signup and create account

Result Snapshot

Edit Profile/View Profile-



This is a Profile Page of a user
User can update his profile and manage his/her data which get stored in database in real time

User can also upload his image which gets upload in a cloud platform called Cloudinary

Fig 6. User Interface for the patient health record

Result Snapshot

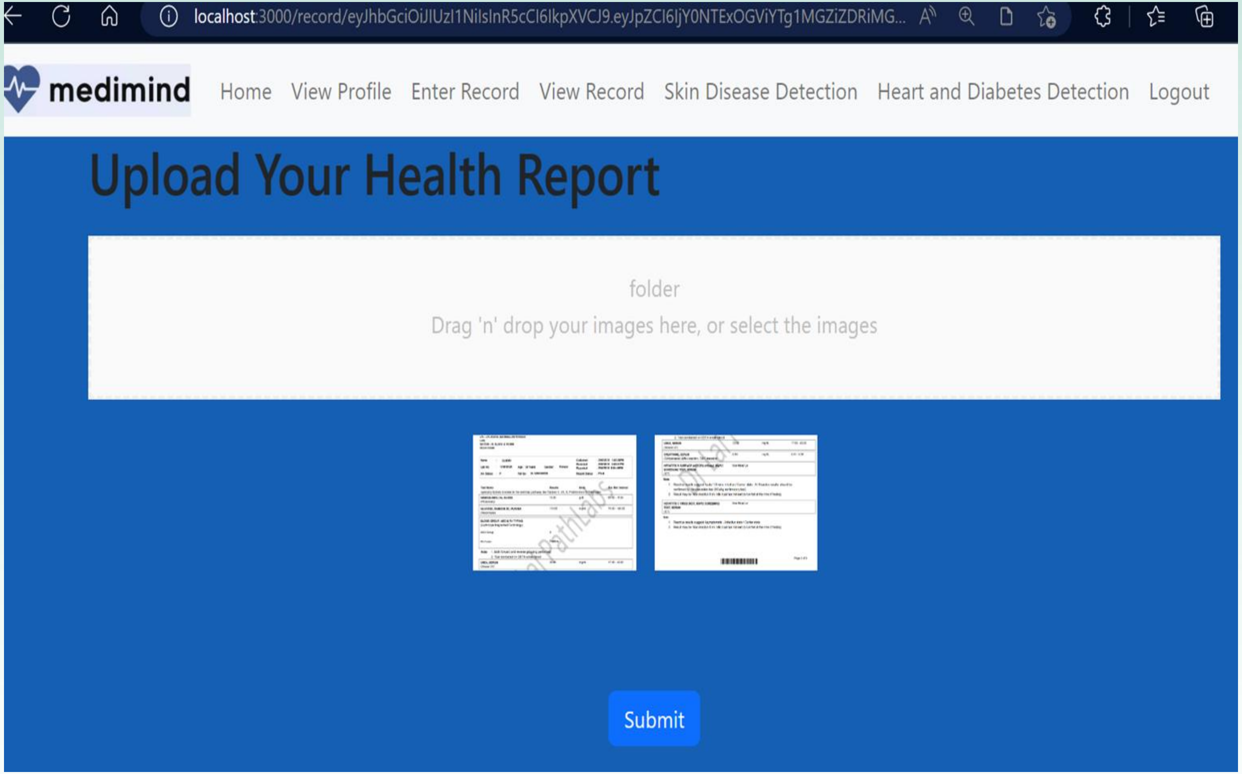


Fig 7. User entering the health report

This is a page from where the user can upload his health record

The health record is uploaded in a cloud platform named cloudinary and saved in our database

Result Snapshot

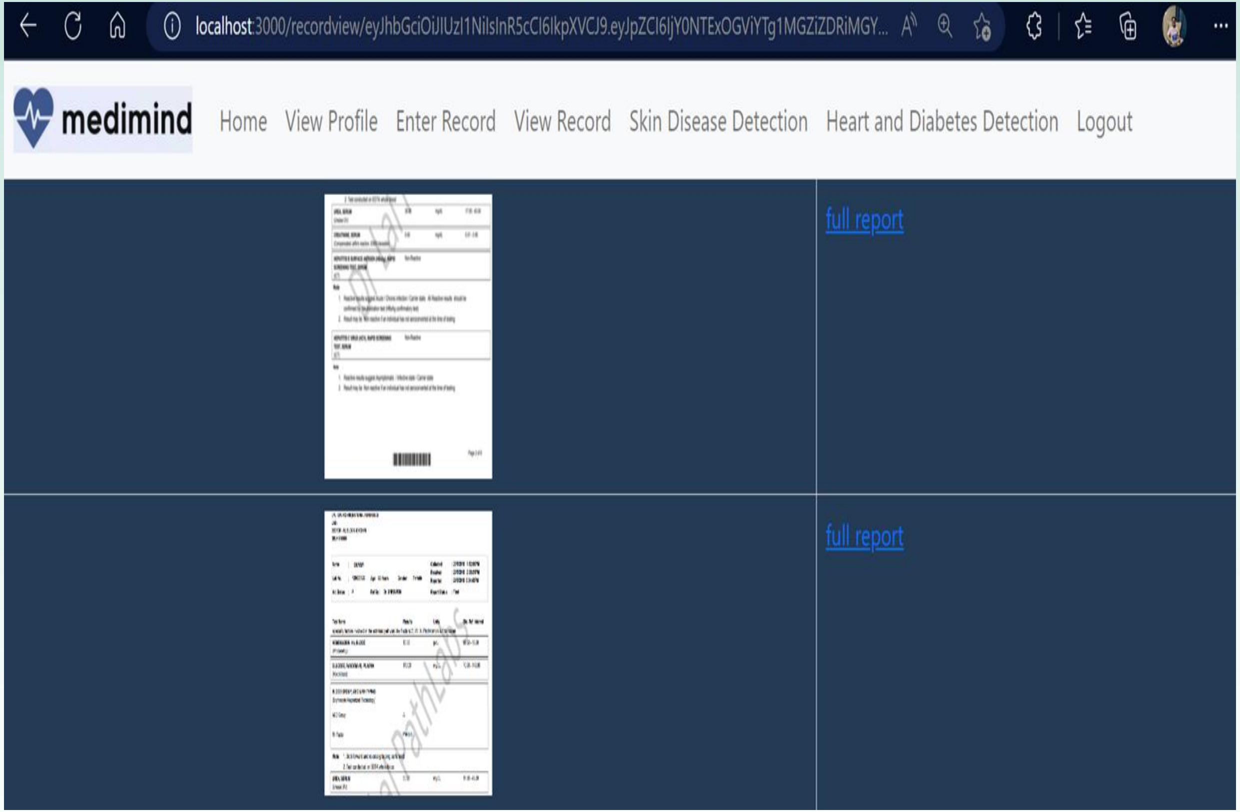


Fig 8. User viewing the health report

This is a page where the user can view all the record that he has uploaded

The data is fetch from the database

```
const mongoose=require('mongoose');
mongoose.set('strictQuery',false)

const connectWithDb=()=>{
    mongoose.connect("mongodb+srv://miniprojectmsrit2024:miniproject@cluster0.6krxmi4.mongodb.net/Health?retryWrite
    .then(console.log("Connected With The databse !!!!!!!!"))
    .catch(error=>{
        console.log("Some Error Occured")
    })
}

module.exports=connectWithDb
```

Fig 9 Code For Database Connectivity

```
const mongoose=require('mongoose');
const User=require('./user')

const reportSchema=mongoose.Schema({

    user:{
        type:mongoose.Schema.Types.ObjectId,
        ref:'User',
    },
    report:[
        {
            url:{
                type:String
            }
        }
    ]
})

module.exports=mongoose.model('Reports',reportSchema);
```

```
const mongoose=require('mongoose')
const bcrypt=require('bcryptjs')
const jwt=require('jsonwebtoken')
//name,email,phoneno,age,weight,height,address,city
const userSchema=mongoose.Schema({
    name:{
        type:String
    },
    email:{
        type:String,
        required:true
    },
    password:{
        type:String,
        required:true,
        select:false
    },
    phoneno:{
        type:Number
    },
    age:{
        type:Number
    },
    weight:{
        type:Number
    },
    height:{
        type:Number
    },
    address:{
        type:String
    },
    city:{
        type:String
    },
    url:{
        type:String
    },
    createdAt:{
        type:Date,
        default:Date.now()
    },

})
```

Fig 10 & 11 Shows the User and Record Model

```
const express=require("express");
const { addRecord,recordview } = require("../controllers/reportController");
const router=express.Router();



router.route("/addRecord/:token").post(addRecord)
router.route("/recordview/:token").get(recordview)

module.exports=router
```

```
const express=require('express')
const router=express.Router()
const {signup,signIn,profile,edit,imageadd }=require('../controllers/userController')


router.route("/signup").post(signup)

router.route("/signIn").post(signIn)

router.route("/profile/:token").get(profile)
router.route("/edit/:token").patch(edit)
router.route("/imageadd/:token").patch(imageadd)



router.route("/").get((req,res)=>{
    res.send("Hello Home Route")
})


module.exports=router
```

Fig 12 & 13 Shows The Route for User and record Saving

TECHNOLOGIES USED AT EACH STAGE

PHASE III

In this phase, we will use Tensorflow,keras and csv file with other required libraries to train the model.

PHASE IV

In this phase their is implementation of Logistic Regression and Cnn model using the phase 3 setup for Predictive Analysis of Skin Disease ,Heart and Diabetics

ALGORITHMIC DESCRIPTION

A **CNN (Convolutional Neural Network)** is a type of artificial intelligence model designed to process and analyze visual data, such as images and videos.

CNN uses a series of mathematical operations called convolutions to extract important features from an input image. These convolutions are essentially filters that slide across the image, looking for patterns in the pixel values.

The output of these convolutions is then passed through a series of layers, which help to identify increasingly complex patterns and structures in the image. This process is called feature extraction.

Finally, the output of the last layer is fed into a classification layer, which assigns a label to the input image based on the features that were extracted. This allows the CNN to recognize and classify objects within an image.

ALGORITHMIC DESCRIPTION

Logistic regression is a statistical method used to analyze the relationship between a binary (yes/no) dependent variable and one or more independent variables.This is a supervised learning technique.

The main goal of logistic regression is to predict the probability of a certain event happening based on input variables.

The logistic regression model works by fitting a curve to the data points that best describes the relationship between the input variables and the outcome. The curve is called the logistic function or sigmoid function, and it outputs a value between 0 and 1, which represents the predicted probability of the event occurring.

The model then calculates a threshold value (usually 0.5) and classifies each observation as either positive (event occurs) or negative (event does not occur) based on whether the predicted probability is above or below the threshold.

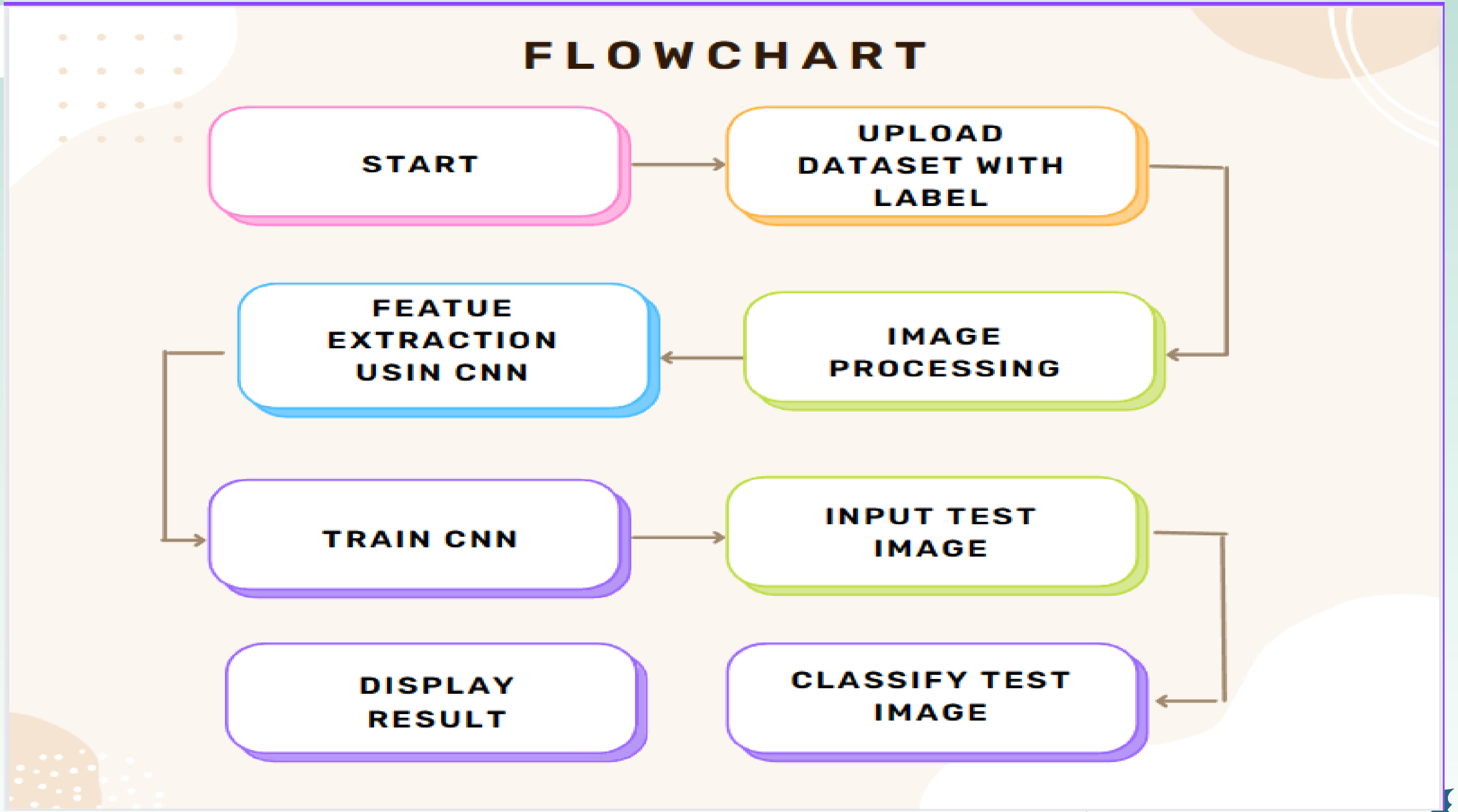


Fig 14 Flowchart for Skin Diagnosis

Result Snapshot

Fig 17. The name of the disease as output

Fig 15. User giving the image of diseased part

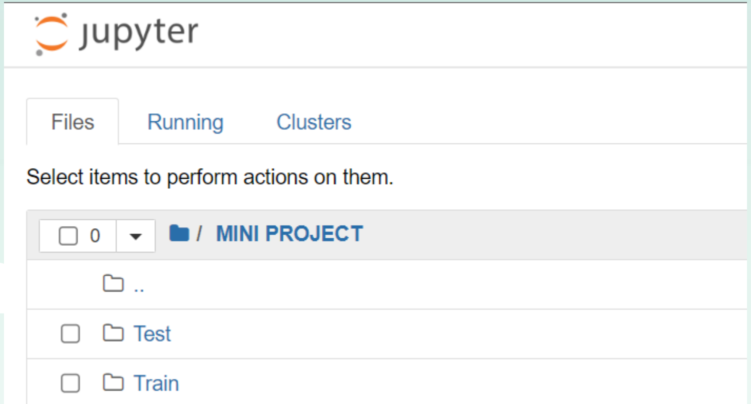


Fig 16. Test and Training Dataset

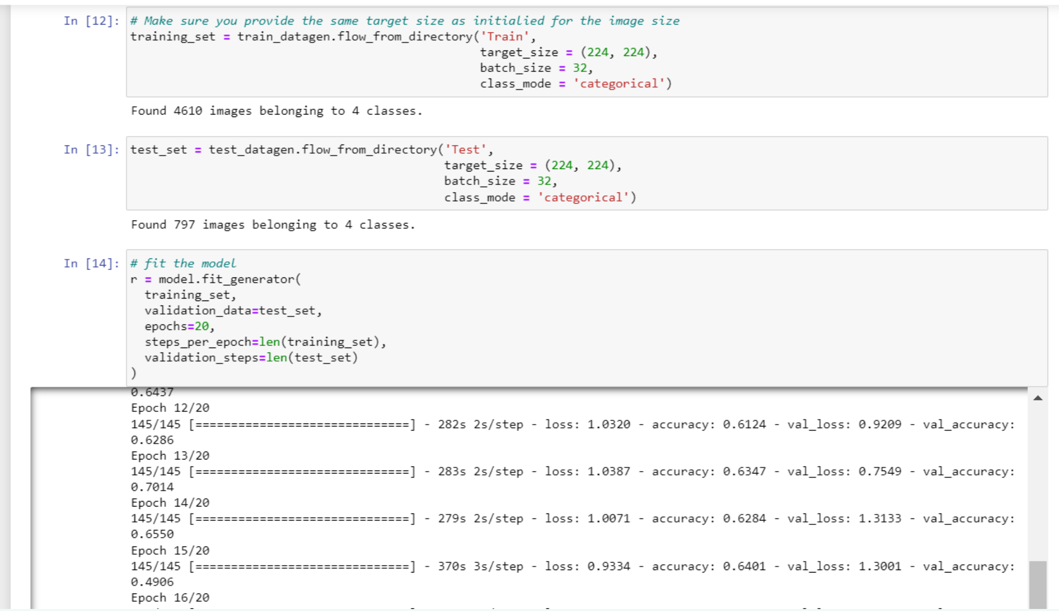


Fig 17. Training of Model

```python
diabetes_model = pickle.load(open('C:/Users/asus/OneDrive/Desktop/Multiple Disease Prediction S

heart_disease_model = pickle.load(open('C:/Users/asus/OneDrive/Desktop/Multiple Disease Predict



# sidebar for navigation
with st.sidebar:

    selected = option_menu('Multiple Disease Prediction System',

                          ['Diabetes Prediction',
                           'Heart Disease Prediction',],
                          icons=['activity','heart'],
                          default_index=0)


# Diabetes Prediction Page
if (selected == 'Diabetes Prediction'):

    # page title
    st.title('Diabetes Prediction using ML')


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

    with col1:
        Pregnancies = st.text_input('Number of Pregnancies')
```

```python
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, res

    if (heart_prediction[0] == 1):
      heart_diagnosis = 'The person is having heart disease'
    else:
      heart_diagnosis = 'The person does not have any heart disease'

st.success(heart_diagnosis)
```

Fig 18 & 19 Backend Code snippet for Heart and Diabetes Diagnoses

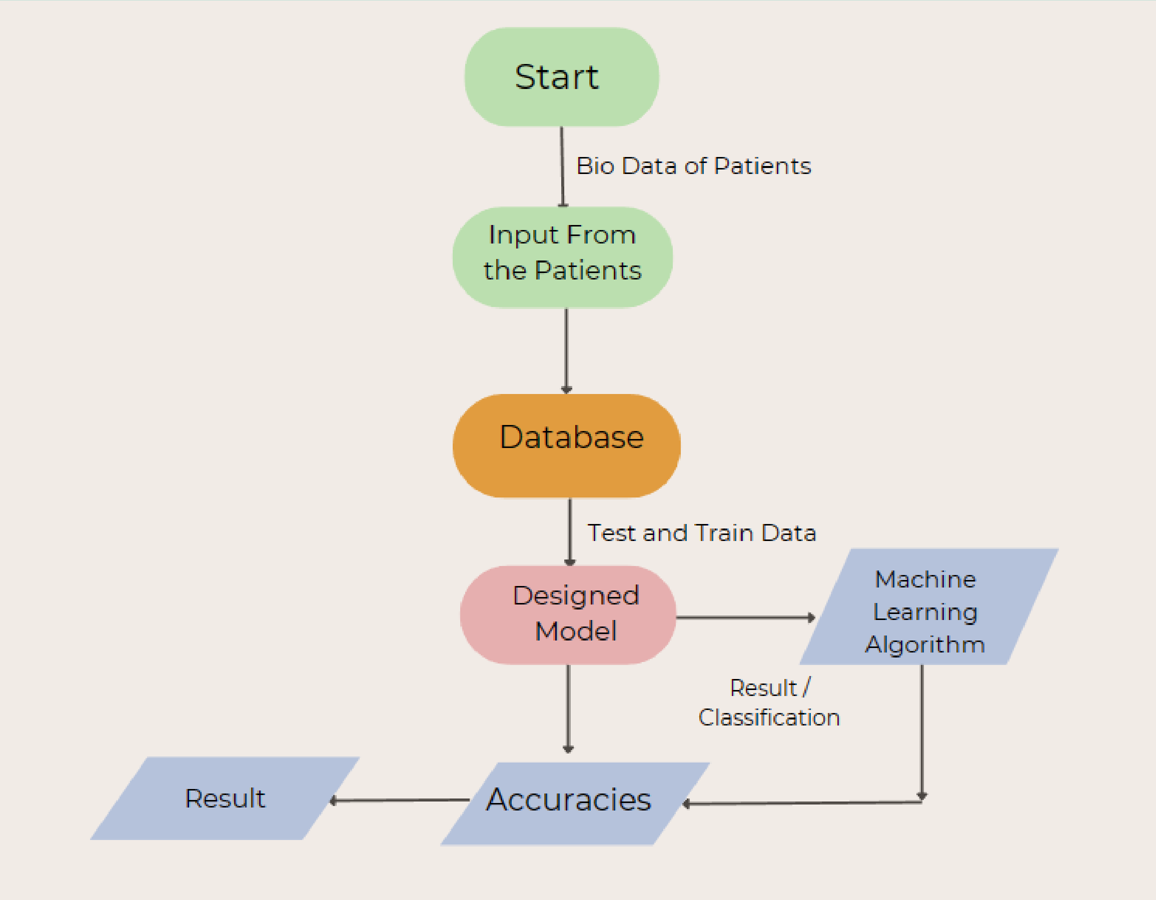
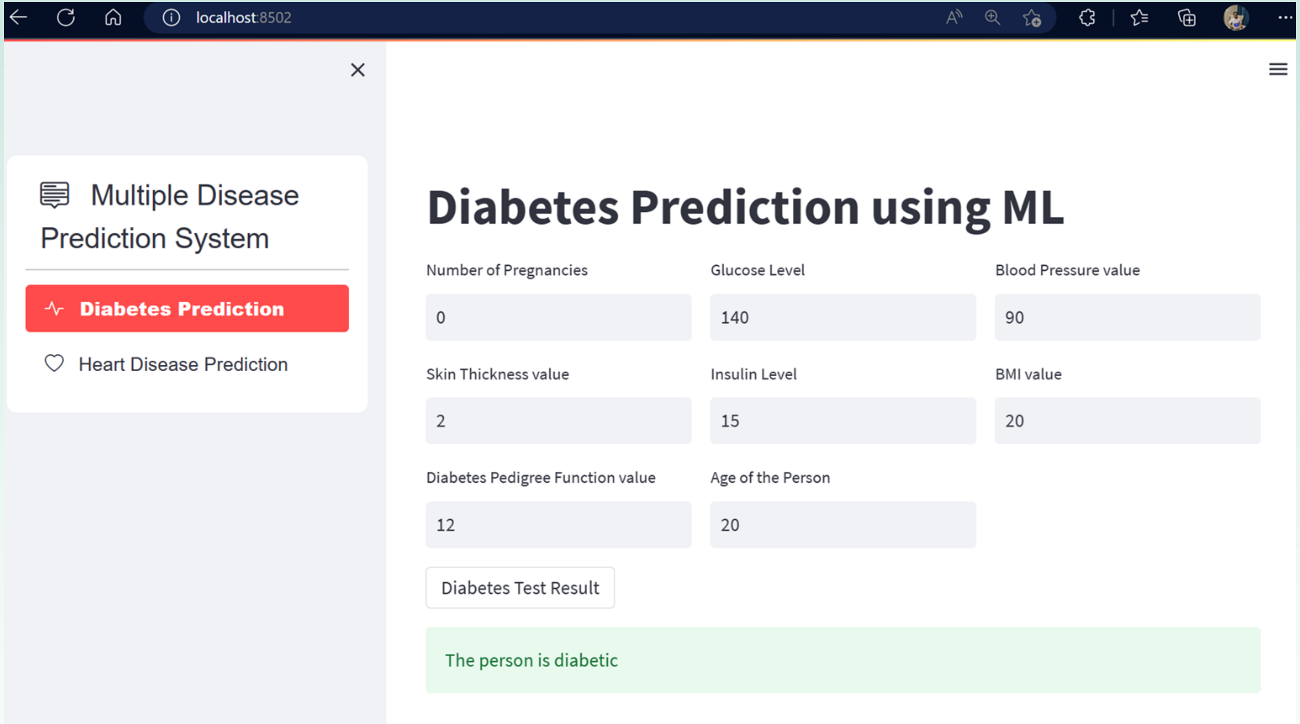


Fig 20 Process Flow for Diabetes and Heart

Result Snapshot



This is a page where the user can test based on the data he enter whether he has suffer from diabetes or not Based on the user input

Fig 21. The User Interface for Diabetes prediction

In [6]:
```python
# loading the diabetes dataset to a pandas DataFrame
diabetes_dataset = pd.read_csv('diabetes.csv')
```

In [7]:
```python
diabetes_dataset.head()
```

Out[7]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

In [8]:
```python
# number of rows and Columns in this dataset
diabetes_dataset.shape
```

Out[8]:
```
(768, 9)
```

In [9]:
```python
# getting the statistical measures of the data
diabetes_dataset.describe()
```

Out[9]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

In [9]:
```python
# statistical measures about the data
heart_data.describe()
```

Out[9]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	
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	303.0
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	2.31
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	0.61
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	0.0
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	2.0
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	2.0
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	3.0
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	3.0

In [10]:
```python
# checking the distribution of Target Variable
heart_data['target'].value_counts()
```

Out[10]:
```
1    165
0    138
Name: target, dtype: int64
```

In [11]:
```python
X = heart_data.drop(columns='target', axis=1)
Y = heart_data['target']
```

In [12]:
```python
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
```

Fig 22&23. Training model for Diabetes and Heart

Reason For Specific Algorithm

For diabetes and heart disease prediction, we are using logistic regression model because it is a supervised learning technique and easy to use. In this technique, we feed the model with the data and the what will be the output for the certain set of input data.This gives us the output in the form of 0/1.This model is easy to use and easy to implement as it works on the prefeeded input data.Therefore, it has good accuracy.

For skin disease prediction, we are using CNN model. This uses the properties of feature extraction by creating patterns out of the images . Then, it divides the data/images on the basis of feature. Also, CNN model is best if we are working on the visual data such as images/videos. The accuracy of CNN model is good.

PROPOSED METHODOLOGY

1. Data Privacy and Security(Integrating Blockchain)
2. Continuous Monitoring and Feedback
3. Deployment and Scalability
4. Continuous Development and Updating
5. Increasing accuracy of our model

CONCLUSION

- In conclusion, the development of MediMind, a comprehensive health prediction and record-keeping platform, has the potential to significantly improve healthcare outcomes by leveraging machine learning algorithms to predict and prevent diseases, while also providing a centralized platform for managing patient health records.

- Through the use of advanced deep learning algorithms, such as neural networks and logistic regression models, MediMind can accurately predict and diagnose diseases, leading to earlier interventions and improved patient outcomes.

REFERENCES

1.Diagnosis of skin diseases in the era of deep learning and mobile technology Ahmed Elaziz, Moustafa M. Ghanem, Islam A. Tolba, Mohamed N. Mostafa, Mohammed A. https://www.sciencedirect.com/science/article/pii/S0010482521002523

2.A Method Of Skin Disease Detection Using Image Processing And Machine Learning Nawal Soliman ALKolifi ALEnezi https://www.sciencedirect.com/science/article/pii/S1877050919321295

3.Diagnosis of skin diseases using Convolutional Neural Networks Jainesh Rathod, Vishal Aniru Sodha, Praseniit Bhavathankar https://ieeexplore.ieee.org/abstract/document/8474593

4.Deep Learning and Machine Learning Techniques of Diagnosis Dermoscopy Images for Early Detection of Skin Diseases Ebrahim Mohammed Senan, Ibrahim Abunadi https://www.mdpi.com/2079-9292/10/24/315

5.Early Skin Disease Identification Using eep Neural Network Vinay Gautam, Naresh Kumar Trivedi, Abhineet Anand, Rajeev Tiwari, Atef Zaguia,Deepika Koundal 4 , Sachin Jain https://www.techscience.com/csse/v44n3/49136

6.Heart disease prediction using machine learning algorithms Harshit Jindal, Sarthak Agrawal, Rishabh Khera, Rachna Jain2and Preeti Nagrath https://iopscience.iop.org/article/10.1088/1757-899X/1022/1/012072/met

7.Prediction of Heart Disease Using a Combination of Machine Learning and Deep Learning Rohit Bharti, Aditya Khamparia, Mohammad Shabaz, Gaurav Dhiman, Sagar Pande, and Parneet Singh https://www.hindawi.com/journals/cin/2021/8387680/