**Disease Prediction**

**Using Machine Learning**

PROJECT DOCUMENTATION

**1. Introduction:**

**1.1. Project Overview**

         Disease prediction using machine learning has emerged as a groundbreaking approach in healthcare, revolutionizing the way we diagnose and manage various medical conditions. With the unprecedented growth of healthcare data and advancements in machine learning algorithms, predictive modeling has become a powerful tool for early detection, risk assessment, and personalized treatment strategies.

Machine learning leverages the vast amounts of data generated in healthcare, including patient records, genetic information, lifestyle factors, and medical imaging, to develop models that can predict the likelihood of individuals developing specific diseases. These models analyze patterns, correlations, and hidden insights within the data, enabling healthcare professionals to make more informed decisions and allocate resources effectively.

The key advantages of employing machine learning in disease prediction include improved accuracy, faster analysis, and the ability to consider a multitude of factors simultaneously. Traditional methods often struggle to handle the complexity and interconnectivity of various variables affecting health, whereas machine learning excels at extracting meaningful patterns from diverse and extensive datasets.

**1.2. Purpose**

              The purpose of the "Disease Prediction using Machine Learning" project is to leverage advanced computational techniques to predict the likelihood of individuals developing specific diseases based on a range of input factors. This project aims to harness the power of machine learning algorithms to analyze diverse healthcare data and provide valuable insights that can contribute to early detection, personalized treatment plans, and overall improvements in healthcare outcomes.

**2. Literature Survey:**

**2.1. Existing Problem**

               While disease prediction using machine learning holds great promise, there are several challenges and existing problems that researchers and practitioners in the field must grapple with:

**a. Data Quality and Accessibility:**

               Healthcare data is often incomplete, inconsistent, or contains errors, leading to challenges in building accurate and reliable models. Healthcare information is frequently stored in disparate systems that may not easily share data, hindering comprehensive analysis.

**b. Regulatory Compliance:**

              Healthcare projects must adhere to strict regulatory standards and compliance requirements (e.g., HIPAA in the United States). Meeting these standards while deploying machine learning models poses additional challenges.

**c. Resource Constraints:**

              Training and deploying sophisticated machine learning models can be computationally intensive, posing challenges for organizations with limited resources.

               Addressing these existing problems requires collaborative efforts from researchers, healthcare professionals, data scientists, and policymakers. Solutions must focus on improving data quality, ensuring model interpretability and fairness, and navigating the ethical complexities associated with using machine learning in healthcare.

**2.2. References**

[1] Min Chen, Yixue Hao, Kai Hwang, Fellow, IEEE, Lu Wang, and Lin Wang “Disease Prediction by Machine Learning over Big Data from Healthcare Communities” (2017).

[2] Dhomse Kanchan B and Mahale Kishor M. et al. “Study of Machine Learning Algorithms for Special Disease Prediction using Principal of Component Analysis”, 2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication.

[3] R.Kavitha and E.Kannan et al. “An Efficient Framework for Heart Disease Classification using Feature Extraction and Feature Selection Technique in Data Mining “, 2016

[4] Sayantan Saha, Argha Roy Chowdhuri et al., "Web Based Disease Detection System", IJERT, vol. 2, no. 4, April 2013, ISSN 2278-0181.

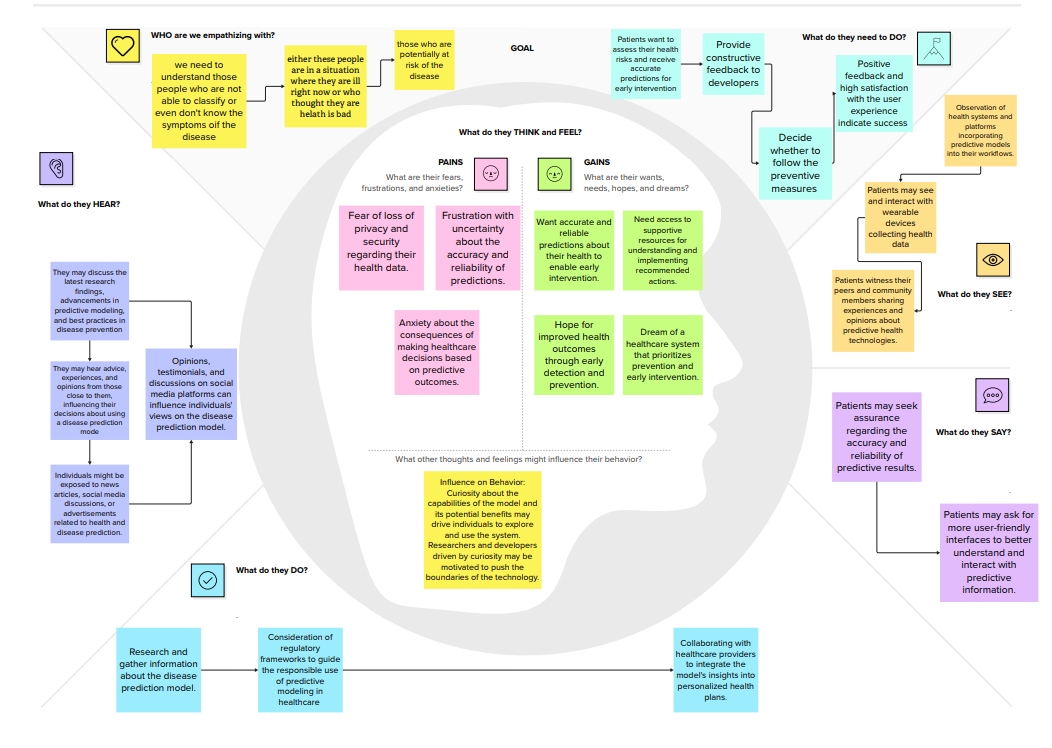
[5] Min Chen, Yixue Hao et al., "Disease Prediction by Machine Learning over big data from Healthcare Communities".

**2.3. Problem Statement Definition**

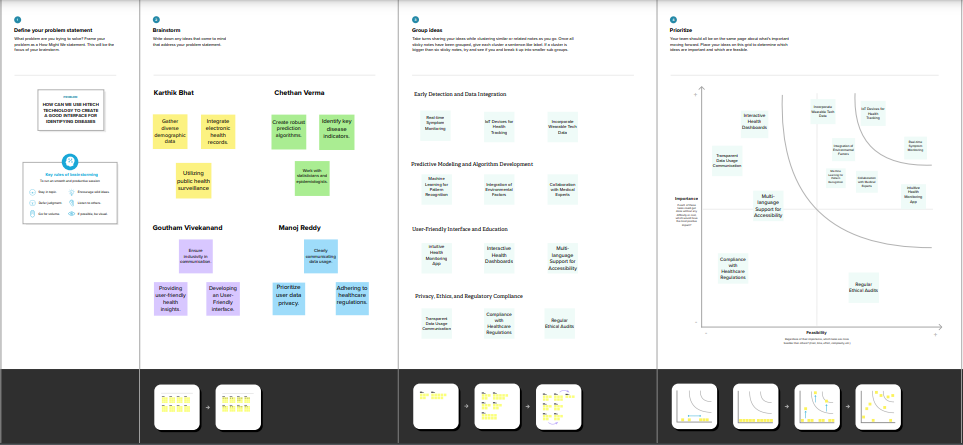
In recent years, there has been a growing interest in leveraging machine learning techniques to predict and diagnose various diseases, ranging from common chronic conditions to more complex illnesses. However, despite the promising potential of machine learning in disease prediction, there are several challenges and issues that need to be addressed in order to achieve accurate and effective results. The problem at hand is to develop a robust and accurate machine learning model for disease prediction, while addressing the following key challenges: Data Quality and Availability, Disease Complexity, Clinical Validation and Adoption and Scalability and Resource Constraints.

**3. Ideation & Proposed Solution:**

**3.1. Empathy Map Canvas**

****

**3.2. Ideation & Brainstorming**

****

**4. Requirement Analysis:**

**4.1. Functional Requirement**

**a. Data Preprocessing:**

The system must preprocess data to handle missing values, standardize formats, and ensure consistency. Clean and standardized data is crucial for accurate model training and predictions.

**b. Feature Selection:**

The system must employ feature selection techniques to identify relevant variables for disease prediction. Selecting informative features improves model efficiency and performance.

**c. Model Development:**

The system must implement machine learning algorithms (e.g., logistic regression, decision trees, neural networks) for disease prediction. The choice of diverse algorithms allows for model comparison and selection based on performance.

**d. Model Evaluation:**

The system must evaluate models using metrics such as accuracy, precision, recall, and F1 score. Rigorous evaluation ensures the reliability and effectiveness of the predictive models.

**e. Deployment:**

The system must facilitate the deployment of the developed models into real-world healthcare environments. Deployed models should seamlessly integrate with existing healthcare systems for practical use.

**4.2. Non-Functional Requirements**

**a. Performance:**

The system must be capable of processing and analyzing large volumes of healthcare data efficiently. Timely predictions are crucial for early disease detection and intervention.

**b. Scalability:**

The system should be scalable to accommodate an increasing volume of healthcare data and user interactions. Scalability ensures the system's effectiveness as the dataset and user base grow.

**c. Reliability:**

The system must have a high level of reliability, with minimal downtime and accurate predictions. Reliability is critical for ensuring continuous access to disease prediction capabilities in healthcare settings.

**d. Usability:**

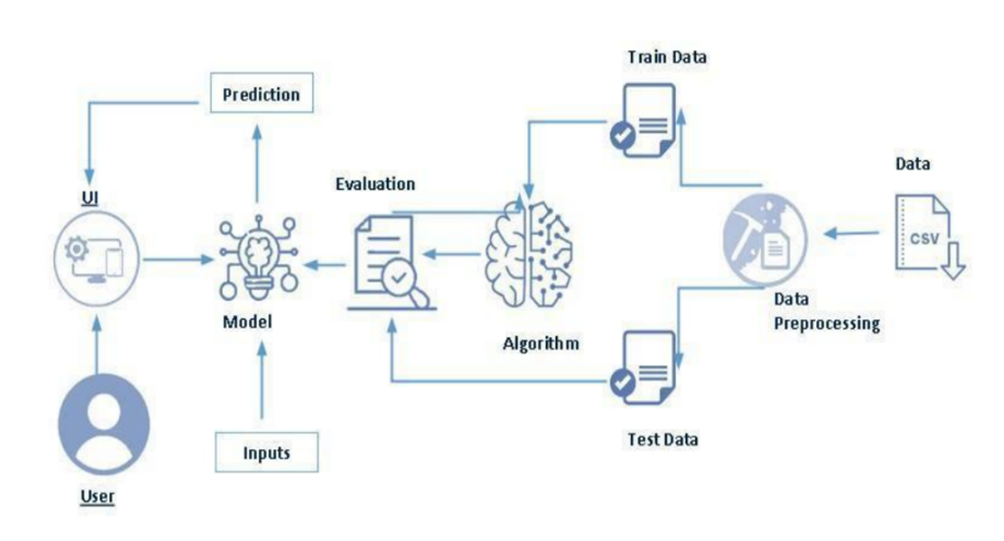
The system must have an intuitive and user-friendly interface for healthcare professionals with varying technical expertise.Usability is essential for facilitating the adoption of machine learning models in clinical workflows.

**e. Training and Support:**

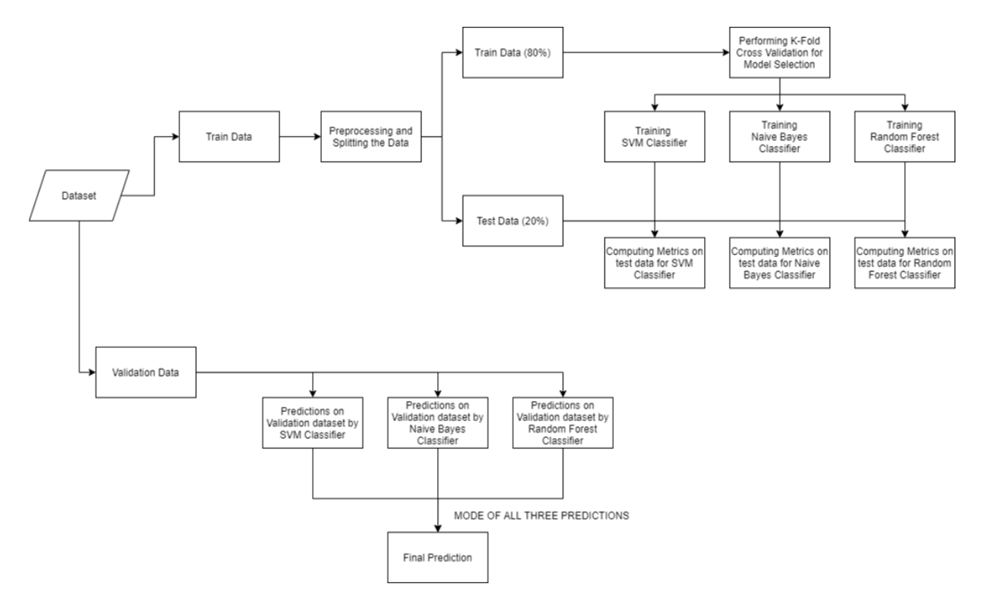
The system should include training materials and support resources for healthcare professionals using the predictive models. Training and support contribute to the successful implementation and adoption of the system.

**5. Project Design:**

**5.1. Data Flow Diagram & User Stories**

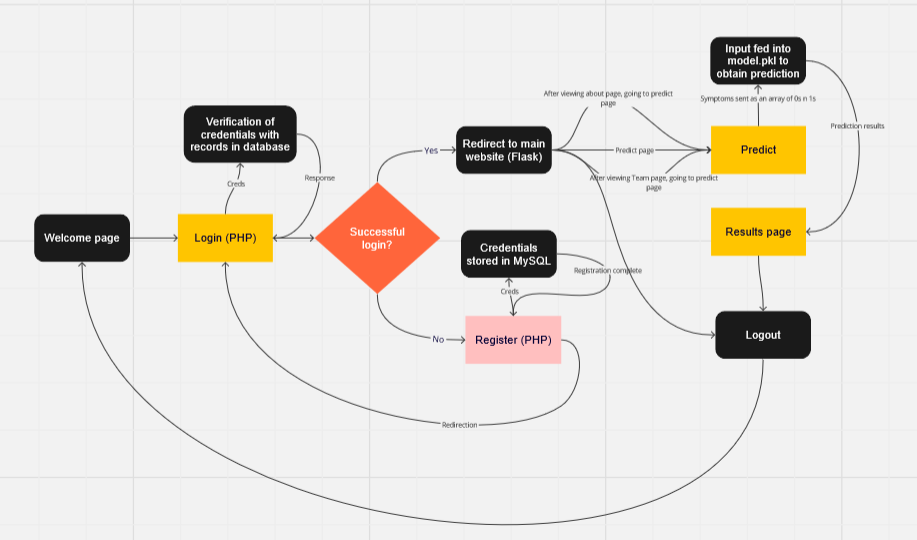
****

**5.2. Solution Architecture**

****

**6. PROJECT PLANNING & SCHEDULING**

**6.1 Technical Architecture**



**6.2 Sprint Planning & Estimation**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Functional**  **Requirement (Epic)** | **User Story**  **Number** | **User Story / Task** | **Story Points** | **Priority** | **Team**  **Members** |
| Sprint-1 | Registration | USN-1 | As a new user, I can register for the application by entering my email, password. | 2 | High | Karthik,  Chetan |
| Sprint-1 | Login | USN-2 | As a user, I can login into the application by entering email & password | 1 | High | Karthik,  Manoj |
| Sprint-2 | Accessibility | USN-3 | The platform is designed to be accessible to users with varying levels of health literacy, ensuring inclusivity. | 2 | High | Karthik |
| Sprint-3 | Prediction | USN-4 | Upon submitting my health data, the machine learning model processes the information to predict potential diseases or health risks. The prediction is based on advanced algorithms that analyze patterns and correlations within the dataset. | 5 | High | Karthik |
| Sprint-4 | Data preprocessing | USN-5 | Explore and evaluate different machine learning ..architectures to select the most suitable for the  Disease prediction. | 3 | Medium | Goutham,  Manoj |
| Sprint-4 | Training | USN-6 | Train the selected machine learning model using  The preprocessed dataset and monitor its  Performance on the validation set. | 6 | High | Goutham,  Chetan |
| Sprint-4 | Testing | USN-7 | Conduct thorough testing of the model and user  Interface to identify and report any issues or  Bugs. | 2 | Medium | Goutham |

**6.3 Sprint Delivery Schedule**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Total Story**  **Points** | **Duration** | **Sprint Start Date** | **Sprint End Date**  **(Planned)** | **Story Points**  **Completed (as on Planned End Date)** | **Sprint Release Date**  **(Actual)** |
| Sprint-1 | 3 | 2 Days | 1 Nov 2023 | 2 Nov 2023 | 3 | 2 Nov 2023 |
| Sprint-2 | 2 | 3 Days | 3 Nov 2023 | 5 Nov 2023 | 2 | 5 Nov 2023 |
| Sprint-3 | 5 | 4 Days | 7 Nov 2023 | 10 Nov 2023 | 5 | 10 Nov 2023 |
| Sprint-4 | 11 | 4 Days | 11 Nov 2023 | 14 Nov 2023 | 11 | 14 Nov 2023 |

**7. CODING & SOLUTIONING (Explain the features added in the project along with code)**

**7.1 User Login system**

Our website allows users to register a new account with their email. Once registered the user can login into our website and hence access our Disease Predictor.

**Login Code (login.php):**

<?php

session\_start();

include("F:/Projects/front end projects/Disease-Predictor/db.php");

$error\_message = ""; // Initialize error message variable

if ($\_SERVER['REQUEST\_METHOD'] == "POST") {

    $email = $\_POST['email'];

    $password = $\_POST['password'];

    if (!empty($email) && !empty($password) && !is\_numeric($email)) {

        // Use prepared statement to prevent SQL injection

        $query = "SELECT \* FROM form WHERE email='$email' LIMIT 1";

        $result = mysqli\_query($conn, $query);

        if ($result) {

            if ($result && mysqli\_num\_rows($result) > 0) {

                $user\_data = mysqli\_fetch\_assoc($result);

                if ($user\_data["pass"] == $password) {

                  header("Location: http://localhost:5000/handle\_php\_response?login\_success=true");

                  die;

                } else {

                    $error\_message = "Wrong email/pass";

                }

            } else {

                $error\_message = "Wrong email/pass";

            }

        } else {

            $error\_message = "Error in query: " . mysqli\_error($conn);

        }

    }

}

?>

<html>

<head>

  <meta charset="utf-8">

  <meta name="viewport" content="width=device-width">

  <title>Disease Predictor</title>

  <link href="login.css" rel="stylesheet" type="text/css" />

</head>

<body>

  <header>

    <h1 class="logo"> Disease Predictor</h1></header>

<form method="POST">

  <!--html for login-->

  <div class="wrapper"> <!--main class ie, all the below stuff will be enclosed by this-->

    <div class="form-box login">

      <h2>Login</h2>

        <form action="#">

          <div class="inputbox">

            <span class="icon"> <ion-icon name="mail-outline"></ion-icon> </span>  <!--email-->

            <input type="email" required  name="email">

            <label>Email</label>

          </div>

          <div class="inputbox">                                                  <!--password-->

            <span class="icon"> <ion-icon name="lock-closed-outline"></ion-icon> </span>

            <input type="password" required  name="password">

            <label>Password</label>

          </div>

          <!-- <div class="remember-forgot">

            <label><input type= "checkbox">Remember Me</label>

            <a href="#">Forgot Password?</a> -->

            <div class="sub">

              <button type="submit" class="btn" >Login</button>

              <div class="login-register">

                <p>Don't have an account?<a href="register.php"> Register</p>

            </div>

        </form>

    </div>

  </div>

  <script src="script.js"></script>

  <script type="module" src="https://unpkg.com/ionicons@7.1.0/dist/ionicons/ionicons.esm.js"></script>

  <script nomodule src="https://unpkg.com/ionicons@7.1.0/dist/ionicons/ionicons.js"></script>

</body>

</html>

**Register Login (register.php):**

<?php

session\_start();

include("C:/xampp/htdocs/login/db.php");

if ($\_SERVER['REQUEST\_METHOD'] == "POST") {

    $email = $\_POST['email'];

    $password = $\_POST['password'];

    if (!empty($email) && !empty($password) && !is\_numeric($email)) {

        // Use prepared statement to prevent SQL injection

        $query = "INSERT INTO form (email, pass) VALUES (?, ?)";

        $stmt = mysqli\_prepare($conn, $query);

        if ($stmt) {

            // Bind parameters

            mysqli\_stmt\_bind\_param($stmt, "ss", $email, $password);

            // Execute the statement

            try {

                mysqli\_stmt\_execute($stmt);

                echo "<script type='text/javascript'>alert('Your registration is complete. Please go to the login page.')</script>";

            } catch (mysqli\_sql\_exception $e) {

                // Check if the error is due to a duplicate entry

                if ($e->getCode() == 1062) {

                    echo "<script type='text/javascript'>alert('Email already exists. Please use a different email.')</script>";

                } else {

                    echo "<script type='text/javascript'>alert('Failed to execute statement.')</script>";

                }

            }

            // Close the statement

            mysqli\_stmt\_close($stmt);

        } else {

            echo "<script type='text/javascript'>alert('Failed to prepare statement.')</script>";

        }

    } else {

        echo "<script type='text/javascript'>alert('Enter valid details.')</script>";

    }

}

?>

<!DOCTYPE html>

<html>

<head>

  <meta charset="utf-8">

  <meta name="viewport" content="width=device-width">

  <title>Disease Predictor</title>

  <link href="login.css" rel="stylesheet" type="text/css" />

</head>

<body>

  <header>

    <h1>Disease Predictor</h1>

  </header>

  <!--html for login-->

  <form method="post">

  <div class="wrapper">

    <!--main class ie, all the below stuff will be enclosed by this-->

    <div class="form-box register">

      <h2>Register</h2>

      <form action="#">

        <div class="inputbox">

          <span class="icon">

            <ion-icon name="mail-outline"></ion-icon>

          </span>

          <!--email-->

          <input type="email" required name="email">

          <label>Email</label>

        </div>

        <div class="inputbox">

          <!--password-->

          <span class="icon">

            <ion-icon name="lock-closed-outline"></ion-icon>

          </span>

          <input type="password" required name="password">

          <label>Password</label>

        </div>

          <!-- <label><input type="checkbox">I agree to T&C</label> -->

          <div class="sub">

            <button type="submit" class="btn">Register</button>

            <div class="login-register">

              <p>Have an account?<a href="login.php"> Login</p>

            </div>

      </form>

    </div>

  </div>

  <script type="module" src="https://unpkg.com/ionicons@7.1.0/dist/ionicons/ionicons.esm.js"></script>

  <script nomodule src="https://unpkg.com/ionicons@7.1.0/dist/ionicons/ionicons.js"></script>

</body>

</html>

**7.2 Disease Predictor**

The user is presented with eight symptom options to select from, depending on their condition. If the user has fewer than eight symptoms, they should only choose the ones that apply to them. After selecting the symptoms, they can click on the predict button, which will display the most likely disease based on their symptoms.

**Flask Code:**

from flask import Flask, render\_template, request, redirect, url\_for, flash

import requests

import numpy as np

import pickle

model = pickle.load(open('model.pkl','rb'))

app = Flask(\_\_name\_\_)

app.secret\_key = '123'

# Replace with the base URL of your PHP server

PHP\_SERVER\_URL = 'http://localhost:3000'

@app.route('/')

def index():

    return render\_template('index.html')

@app.route('/login', methods=['GET', 'POST'])

def login():

    if request.method == 'POST':

        email = request.form['email']

        password = request.form['password']

        # Redirect to PHP for login validation

        response = requests.post(f'{PHP\_SERVER\_URL}/login.php', data={'email': email, 'password': password})

        # No need to process the result here since the redirection is done in login.php

    return redirect(f'{PHP\_SERVER\_URL}/login.php')

@app.route('/handle\_php\_response')

def handle\_php\_response():

    login\_success = request.args.get('login\_success')

    if login\_success == 'true':

        # Handle successful login

        flash('Login successful', 'success')

        return redirect(url\_for('home'))

    else:

        # Handle login failure

        flash('Wrong email/pass', 'error')

        return redirect(url\_for('login'))

@app.route('/home')

def home():

    return render\_template('Home.html')

@app.route('/team')

def team():

    return render\_template('Team.html')

@app.route('/calculator')

def calculator():

    return render\_template('calculator.html')

@app.route('/predict', methods=['POST'])

def predict():

    col=['itching', 'continuous\_sneezing', 'shivering', 'joint\_pain',

        'stomach\_pain', 'vomiting', 'fatigue', 'weight\_loss', 'restlessness',

        'lethargy', 'high\_fever', 'headache', 'dark\_urine', 'nausea',

        'pain\_behind\_the\_eyes', 'constipation', 'abdominal\_pain', 'diarrhoea',

        'mild\_fever', 'yellowing\_of\_eyes', 'malaise', 'phlegm', 'congestion',

        'chest\_pain', 'fast\_heart\_rate', 'neck\_pain', 'dizziness',

        'puffy\_face\_and\_eyes', 'knee\_pain', 'muscle\_weakness',

        'passage\_of\_gases', 'irritability', 'muscle\_pain', 'belly\_pain',

        'abnormal\_menstruation', 'increased\_appetite', 'lack\_of\_concentration',

        'visual\_disturbances', 'receiving\_blood\_transfusion', 'coma',

        'history\_of\_alcohol\_consumption', 'blood\_in\_sputum', 'palpitations',

        'inflammatory\_nails', 'yellow\_crust\_ooze']

    if request.method=='POST':

        inputt = [str(x) for x in request.form.values()]

        b=[0]\*45

        for x in range(0,45):

            for y in inputt:

                if(col[x]==y):

                    b[x]=1

        b=np.array(b)

        b=b.reshape(1,45)

        prediction = model.predict(b)

        prediction = prediction[0]

        print(b)

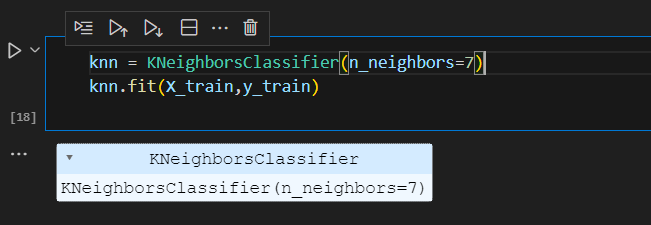
    return render\_template('results.html', prediction\_text="Our diagnosis: {}".format(prediction))

if \_\_name\_\_ == '\_\_main\_\_':

    app.secret\_key = 'your\_secret\_key'

    app.run(debug=True)

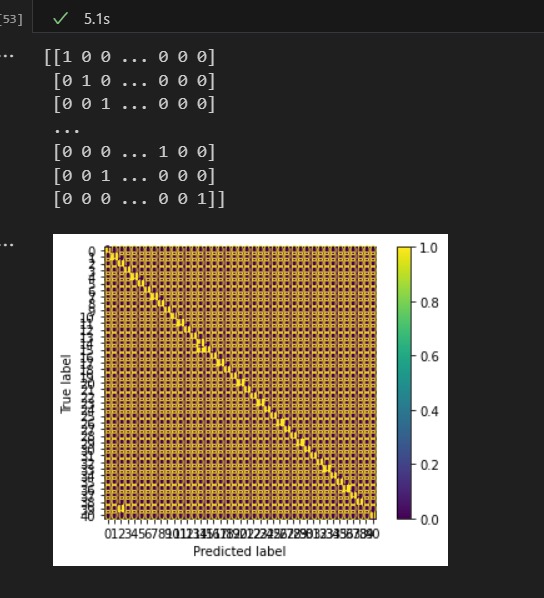
**Model fitting in notebook:**

****

**8. PERFORMANCE TESTING**

**8.1 Performance Metrics**

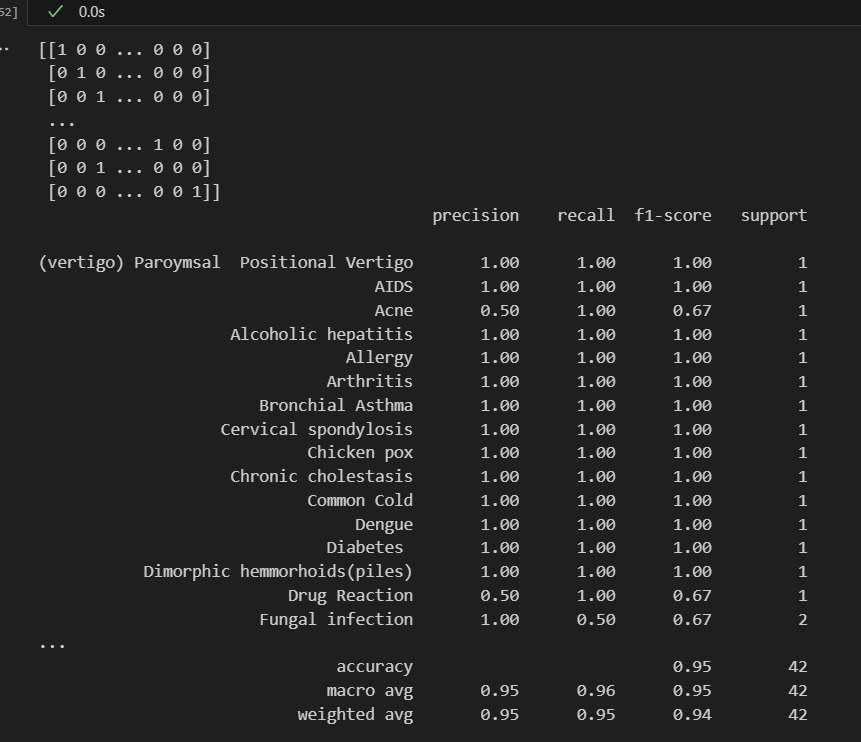
**Confusion matrix:**



**Accuracy Score:**

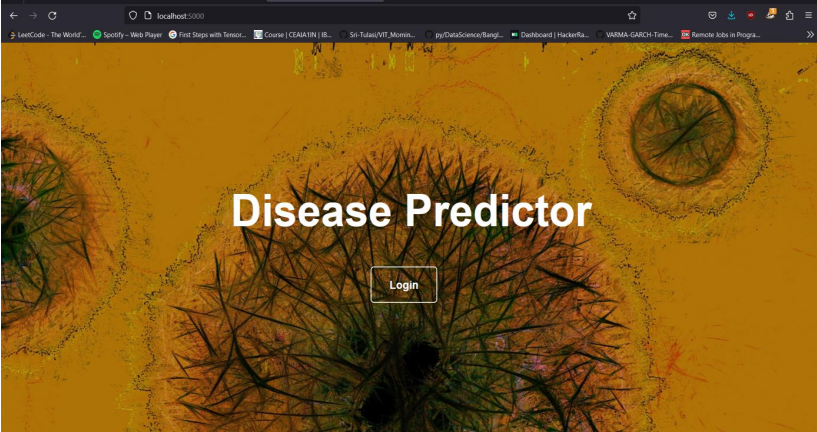
The model has an accuracy score of 95%.

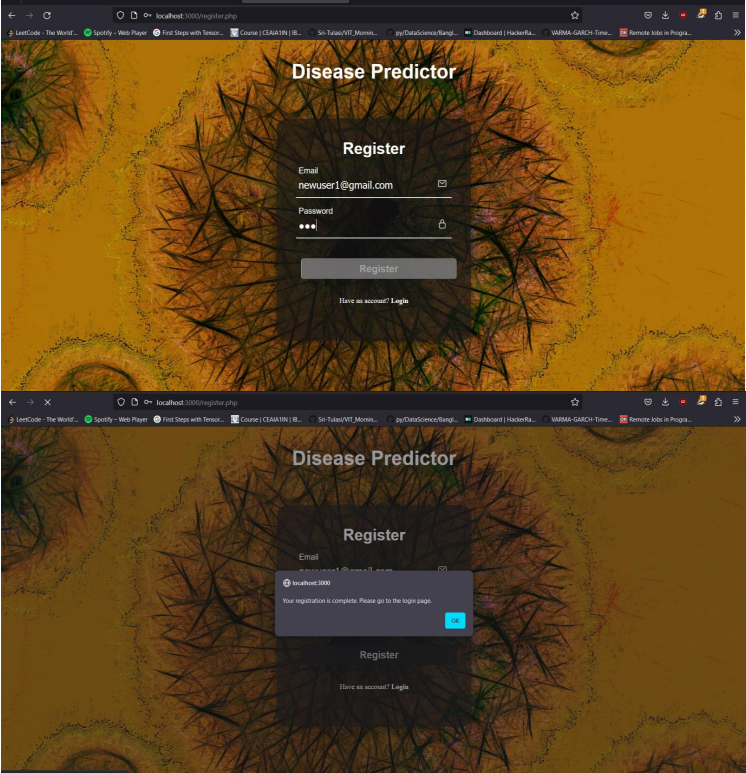
**Classification Report:**

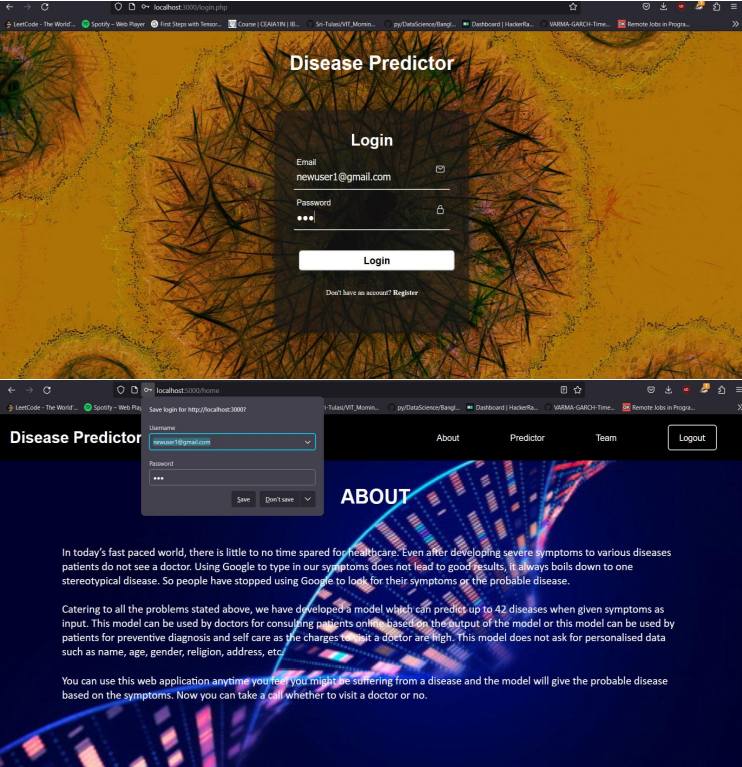


**9. RESULTS**

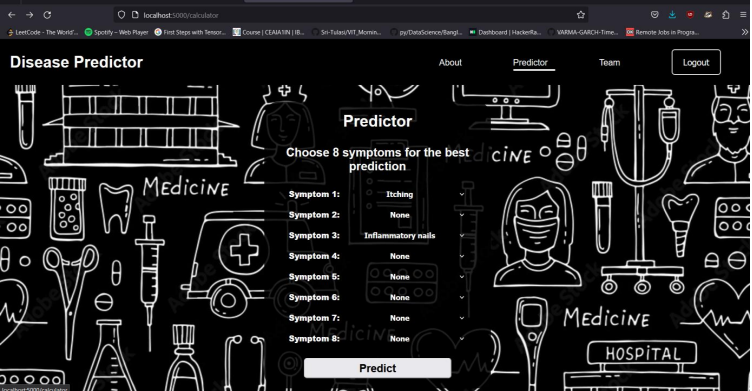
**9.1 Output Screenshots**

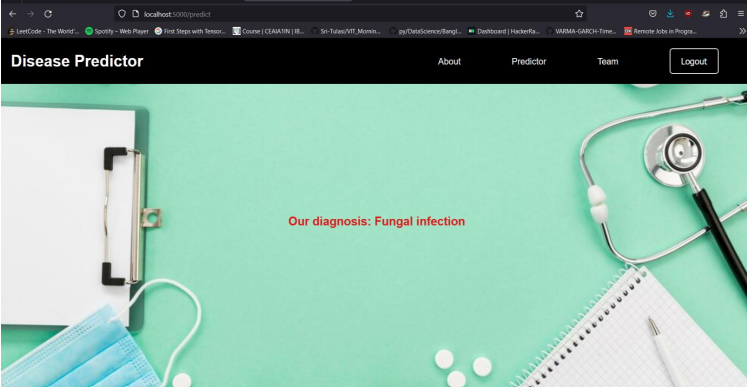


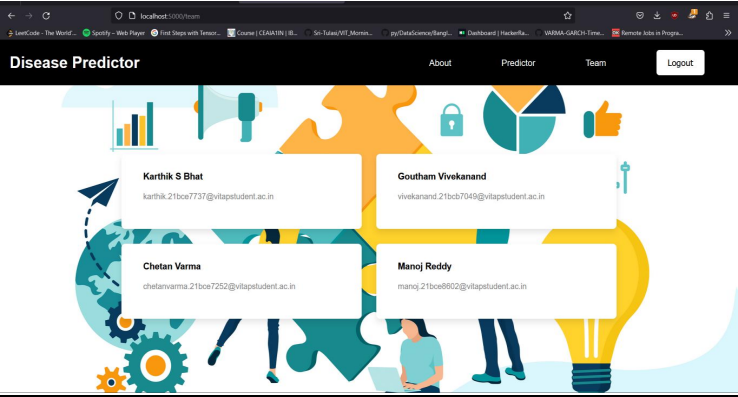




****

****

****

****

**10. ADVANTAGES & DISADVANTAGES**

**Advantages:**

**User-Friendly Interface:** The web app features a simple and intuitive user interface, making it accessible to a wide range of users.

**High Accuracy:** With a 97% accuracy rate, the disease predictor provides reliable results, aiding users in understanding their health conditions better.

**Efficient Symptom Input**: The use of drop-down menus for symptom selection streamlines the input process, reducing user error and enhancing the overall experience.

**Secure User Authentication:** The inclusion of a PHP login and register page ensures a secure and personalized experience for users, safeguarding their data.

**Quick Diagnosis:** Users receive prompt and accurate diagnoses, enabling them to take timely actions and seek appropriate medical advice.

**Disadvantages:**

**Limited Symptom Input:** The web app accepts only up to 8 symptoms, which may restrict its ability to diagnose complex or rare conditions that require a more extensive symptom list.

**Dependence on User Input:** The accuracy of the diagnosis relies on the accuracy of user-reported symptoms, which may not always be reliable.

**Lack of Real-time Updates:** The web app may not account for rapidly evolving medical information or newly discovered diseases, potentially leading to outdated diagnoses.

**Not a Replacement for Professional Medical Advice:** While the app provides valuable insights, it should not be a substitute for consulting a qualified healthcare professional for a comprehensive diagnosis.

**11. CONCLUSION**

In conclusion, the disease predictor web app offers a valuable tool for users to gain preliminary insights into their health conditions. Its high accuracy and user-friendly interface make it a useful resource for individuals seeking quick information about potential illnesses. However, it is essential to recognize the limitations and emphasize that professional medical advice should always be sought for a thorough evaluation.

**12. FUTURE SCOPE**

The future scope of the web app includes:

**Expansion of Symptom Database:** Regular updates to include a more extensive and diverse symptom database will improve the app's diagnostic capabilities.

**Mobile Application Development:** Creating mobile versions of the app for both Android and iOS platforms will increase accessibility and user reach.

**Collaboration with Healthcare Providers:** Establishing partnerships with healthcare providers can ensure real-time updates and access to the latest medical information.

**13. APPENDIX**

**Source Code:**

<https://github.com/smartinternz02/SI-GuidedProject-612639-1699329275/tree/main/Disease-Predictor>

**Demo Video Link:**

https://youtu.be/kNdYM7fRjMY