A Major Project

on

MEDIFIT

(Submitted in partial fulfillment of the academic requirements of B. Tech)

In

Department of Computer Science and Engineering

by

O. Srivalli Hrushitha-19B61A05B3
V. Sai Mukesh Reddy-19B61A05F9
K. Sai Narsimha Reddy-19B61A0572
T. Venkata Giridhar-19B61A05F4

Under the Esteemed Guidance

of

Mr. S. Ramchandra Reddy
(Assistant Professor)

NALLA MALLA REDDY ENGINEERING COLLEGE
An Autonomous Institution
(Approved by AICTE, New Delhi and Affiliated to JNTUH)
Accredited by NAAC with “A” Grade
NBA Accredited B.Tech Programmes (CSE, ECE, EEE, MECH)
DIVYA NAGAR, MALKAJGIRI-MEDCHAL (DIST), HYDERABAD-500088

2022-23

Nalla Malla Reddy Engineering College

Divya Nagar Kachivanisingaram PostGhatkesar, Medchal Dist.500088

Phones: O8415-256001.02.03. Fax: 08415-256000

Email: info@nmrec.edu.in Website: www.nmrec.edu.in

CERTIFICATE

This is to certify that the project report entitled "**MEDIFIT**" being submitted

by

O. Srivalli Hrushitha-19B61A05B3
V. Sai Mukesh Reddy-19B61A05F9
K. Sai Narsimha Reddy-19B61A0572
T. Venkata Giridhar-19B61A05F4

in partial fulfillment for the award of the degree of **Bachelor of Technology in Computer Science and Engineering**, Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out under my guidance and supervision. The results embodied in this project report have not been submitted to any other University or Institute for the award of any Degree or Diploma

Internal Guide

Mr. S. Ramchandra Reddy
Assistant Professor
Department of CSE

Head of the Department

Dr. M. Raju
Associate Professor
Department of CSE

Principal

Dr. M. N. V. Ramesh

External Examiner

DECLARATION

We declare that this project Report **MEDIFIT** submitted in partial fulfillment of the degree of B. Tech in Computer Science and Engineering is a record of original work carried out by us under the supervision of **Mr.S.Ramchandra Reddy,** Assistant Professor and has not formed the basis for the award of any other degree or diploma, in this or any other Institution or University. In keeping with the ethical practice in reporting scientific information, due acknowledgments have been made wherever the findings of others have been cited.

O. Srivalli Hrushitha-19B61A05B3

V. Sai Mukesh Reddy-19B61A05F9

K. Sai Narsimha Reddy-19B61A0572

T. Venkata Giridhar-19B61A05F4

ACKNOWLEDGEMENT

Any endeavour in the field of development is a person's intensive activity. A successful project is a fruitful culmination of efforts by many people, some directly involved and some others who have quietly encouraged and supported.

Salutation to the beloved and highly esteemed Institute **NALLA MALLA REDDY ENGINEERING COLLEGE**, for grooming us into Computer Science and Engineering graduates. We wish to express our heart full thanks to the Director **Dr. Divya Nalla** whose support was indispensable to us during the course. We would also thank Principal **Dr. M. N. V. Ramesh** for providing a great learning environment.

We wish to express profound gratitude to **Dr. M. Raju, Associate Professor** and **Head of the Department**, Computer Science and Engineering, for his continuous encouragement to ensure successful results in all our endeavours.

We would like to thank **Mr. S. Ramchandra Reddy, Assistant Professor,** Department of Computer Science and Engineering, who has patiently guided and helped us throughout our project.

We take this opportunity to thank Department Project Co-Ordinator **Mrs. T. Malathi, Assistant Professor** and Project in-charges **Mr. B. Kiran Kumar, Assistant Professor and Mrs. N. Ramya, Assistant Professor** for all the review meetings, suggestions and support throughout the project development. Last but not least, we thank our family members who are the backbone and provided support in every possible way.

By

O. Srivalli Hrushitha-19B61A05B3
V. Sai Mukesh Reddy-19B61A05F9
K. Sai Narsimha Reddy-19B61A0572
T. Venkata Giridhar-19B61A05F4

ABSTRACT

There is a growing burden of lifestyle-related diseases such as diabetes and heart disease, which can often be prevented or managed through lifestyle interventions such as diet and exercise. However, many people struggle to adopt and maintain healthy habits due to factors such as lack of knowledge, motivation, and social support.

MediFit is a web application designed to detect heart disease and diabetes based on user input parameters. Logistic regression is used for heart disease prediction, and SVM is used for diabetes prediction. Additionally, the application provides personalized diet suggestions based on the user's personal details. The application also includes a feature that detects live yoga poses to correct the user's posture. By using MediFit, users can take proactive measures to identify potential health risks early on and make necessary changes to their lifestyle. This application has the potential to promote healthy habits and improve the overall health and wellbeing of its use.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

CHAPTER I
INTRODUCTION

1.1 Motivation

Maintaining physical health and wellbeing is essential for a happy and prosperous life. With the increasing prevalence of sedentary lifestyles and unhealthy dietary habits, the incidence of chronic diseases such as diabetes, heart diseases, and obesity is rising rapidly. These conditions not only affect the quality of life but also pose a significant economic burden on individuals and healthcare systems worldwide. There is a growing need for effective and accessible tools to manage these conditions and promote healthy lifestyles. This is where MediFit comes in.

MediFit is an innovative web application that combines the power of machine learning and yoga therapy to provide users with personalized disease detection, diet recommendations, and yoga poses. The motivation for this project stems from the fact that traditional healthcare systems often fail to address the root cause of chronic diseases, i.e., lifestyle factors. MediFit aims to empower users with knowledge and tools to manage their health proactively and prevent the onset of chronic conditions. By providing personalized recommendations based on user input parameters, MediFit encourages individuals to take ownership of their health and make positive changes in their daily habits.

Another motivation for the MediFit project is to make healthcare accessible and affordable to a broader population. Traditional healthcare systems are often plagued with issues such as high costs, long wait times, and limited access to specialized services, particularly in rural and remote areas. MediFit, being a web-based application, is easily accessible to anyone with an internet connection, and its services are available 24/7. This makes it an ideal solution for individuals who may not have access to traditional healthcare facilities or may be hesitant to seek medical help due to various reasons. With MediFit, users can take charge of their health and well-being from the comfort of their own homes.

1.2 Objective of the Project

The primary objective of MediFit is to help people maintain their physical health and prevent the onset of various diseases such as diabetes and heart disease. By providing users with a platform where they can input their health parameters and receive accurate disease classifications, personalized diet recommendations, and yoga poses that can improve their health, MediFit aims to help people lead a healthy and active lifestyle. By providing users with easy access to information about their health and personalized recommendations, MediFit hopes to encourage people to take control of their own health and wellbeing.

Another objective of MediFit is to make healthcare more accessible and affordable for everyone. Healthcare costs are rising globally, making it difficult for people to access the care they need. MediFit aims to bridge this gap by providing a free platform where people can access reliable health information and recommendations from the comfort of their homes. By leveraging technology and machine learning algorithms, MediFit aims to provide accurate disease classifications and recommendations, reducing the need for expensive medical tests and consultations. This can help people save both time and money, while also empowering them to take control of their health.

MediFit is designed for individuals of all ages and fitness levels who are interested in maintaining or improving their physical health. It caters to a wide range of users, from those who want to prevent chronic diseases through early detection to those who seek personalized diet recommendations and assistance with their yoga practice. The platform is user-friendly and accessible, making it suitable for both novice users and those with prior experience in health management.

This document serves as an introduction to MediFit, providing an overview of its key features, objectives, target audience, and benefits. It outlines the system architecture and highlights the scope of the document. The subsequent sections of the document will delve into further details of each feature, including their implementation, functionality, and user interaction. It aims to provide a comprehensive understanding of MediFit and serve as a reference for users and developers involved in its implementation and maintenance.

CHAPTER II

CHAPTER II
EXISTING METHODOLOGY

Advancements in software technology have transformed the way we manage our personal health and well-being. There are various existing software tools and applications available that can assist individuals in taking care of their health in a more efficient and convenient manner. These technologies leverage the power of data analysis, machine learning, and user-friendly interfaces to empower users in making informed decisions about their health. Some examples of existing software technology used in personal health management include health tracking apps, wearable devices, and telemedicine platforms.

Health tracking apps provide individuals with the ability to monitor various health parameters such as physical activity, heart rate, sleep patterns, and calorie intake. These apps utilize sensors and data analysis algorithms to gather and process information, providing users with valuable insights and personalized recommendations. Wearable devices such as fitness trackers and smartwatches have become increasingly popular as they offer continuous monitoring of vital signs, activity levels, and even stress levels. These devices sync with mobile applications, enabling users to track their progress and receive real-time feedback.

In recent years, the use of online websites and applications for maintaining physical health and receiving yoga recommendations has become increasingly popular. Some of the ways people use these resources include:

Fitness Tracking: There are several websites and applications that help users track their physical activity and monitor their progress over time. Users can input data such as their weight, height, and workout routines, and the software will track and analyze their progress.

Yoga Pose and Sequence Recommendations: Some websites and applications provide users with yoga pose and sequence recommendations based on their experience level and personal preferences. These recommendations can help users to improve their flexibility, balance, and overall well-being.

Diet Planning: Many websites and applications provide users with diet planning tools and resources. These tools can help users to track their caloric intake, monitor their nutrient levels, and plan meals that are appropriate for their personal goals and dietary restrictions.

Online Yoga Classes: There are several websites and applications that offer online yoga classes for users to practice from the comfort of their own homes. These classes are often led by experienced yoga instructors and can be accessed at any time.

2.1 Existing Systems for Disease Prediction

There are several existing systems that can help users detect diseases early.

Health monitoring apps: There are various health monitoring apps available that can track different health parameters like heart rate, blood pressure, and blood sugar levels. These apps can help users detect abnormalities in their health data, which may indicate a potential health problem.

Telemedicine platforms: Telemedicine platforms allow users to connect with healthcare providers remotely, making it easier and more convenient to seek medical advice. These platforms can also provide users with tools to monitor their health at home and detect potential health problems early.

Wearable technology: Wearable technology like smartwatches and fitness trackers can monitor various health parameters like heart rate, activity levels, and sleep patterns. This data can help users detect abnormalities in their health and take proactive measures to address them.

2.2 Existing Systems for Diet Recommendations

MyFitnessPal: MyFitnessPal is a popular calorie-counting app that can track daily food intake and provide users with personalized recommendations based on their goals and preferences.

Nutrino: Nutrino is a personalized nutrition app that uses machine learning algorithms to provide users with customized meal plans and food recommendations based on their dietary preferences, health goals, and health data.

Noom: Noom is a weight loss app that offers users personalized coaching and nutrition guidance based on their goals and preferences. The app uses a behavioural change approach to help users develop healthy habits and achieve sustainable weight loss.

Lose It!: Lose It! is a calorie-tracking app that offers users personalized meal plans, recipes, and food recommendations based on their dietary preferences and health goals.

Fooducate: Fooducate is a nutrition app that offers users personalized food recommendations based on their dietary preferences and health goals. The app also provides users with information about the nutritional value of different foods and ingredients.

2.3 Existing Systems for Yoga recommendation System

Yoga Studio: Yoga Studio is a popular app that offers users access to over 80 yoga and meditation classes, including guided pose instructions and tutorials. The app also includes a pose library that allows users to search for specific poses and learn proper form and alignment.

Pocket Yoga: Pocket Yoga is another app that offers users access to guided yoga classes and pose tutorials. The app includes a pose dictionary with detailed instructions and illustrations to help users learn proper form and alignment.

Yoga.com: Yoga.com is a website and app that offers users access to a comprehensive library of yoga poses and instructional videos. The app includes a pose tracker that allows users to track their progress and set goals for their practice.

Asana Rebel: Asana Rebel is a fitness app that combines yoga and other workouts to create personalized fitness plans for users. The app includes a pose library with detailed instructions and video demonstrations to help users learn proper form and alignment.

In the realm of personal health management, MediFit stands out as a brilliant tool that combines multiple features into a single, comprehensive platform. By integrating disease prediction, personalized diet recommendations, and yoga pose detection, MediFit streamlines the process of maintaining and improving personal health.

CHAPTER III

CHAPTER III
REVIEW OF LITERATURE

Before embarking on any project, conducting a thorough literature review is essential. A literature review involves an extensive search and analysis of existing research, publications, and scholarly articles related to the project's topic. It serves as the foundation for project development and provides several important benefits.

Literature review helps project developers gain a comprehensive understanding of the current state of knowledge in their field. It allows them to identify existing theories, methodologies, and best practices relevant to their project. By examining previous work, developers can avoid reinventing the wheel and build upon established frameworks, ensuring a solid foundation for their project. It helps identify gaps in existing research or areas that require further exploration. By understanding the limitations and shortcomings of previous work, developers can identify opportunities to contribute new insights and advancements. This process is crucial for generating innovative ideas and ensuring the project's uniqueness and relevance in the field.

The literature review played a crucial role in the development of MediFit, providing valuable insights and guiding the project's direction. Through the literature review, various machine learning algorithms commonly used in disease prediction were identified. The review helped in understanding the strengths and limitations of each algorithm and selecting the most suitable ones for MediFit. This informed the choice of logistic regression for heart disease prediction and support vector machines (SVM) for diabetes prediction, ensuring accurate and reliable disease classification.

The literature review revealed the potential of computer vision technology for tracking and detecting yoga poses. PoseNet, a deep learning model, was identified as a suitable solution. The review provided insights into the architecture and implementation of PoseNet, enabling its integration into MediFit. This feature enhanced the user experience by providing real-time feedback and guidance during yoga practice.

The literature review explored the best practices and user interface design principles for health-related applications. It provided guidance on designing an intuitive and user-friendly interface for MediFit, ensuring that users can easily navigate the system, input their parameters, and access the desired features. The review also highlighted the importance of clear and informative visualizations to present results and recommendations to users.

The literature review helped in understanding the significance of disease prediction, personalized diet recommendations, and yoga pose detection in the context of personal health management. It provided evidence of the positive impact these features can have on individuals' well-being and reinforced the importance of integrating them into MediFit.

The following section presents a collection of the significant literature papers studied during the review process, highlighting their key findings and contributions. These papers have greatly influenced the conceptualization, design, and implementation of MediFit, ensuring that it is built upon a solid foundation of scientific research and expertise.

[1] A Systematic Review of Machine Learning Techniques for Heart Disease Prediction - Shivganga Udhan, Dr. Babasaheb Ambedkar, Dr. Babasaheb Ambedkar

The paper focuses on the prediction of outcomes based on existing data is a common machine learning application. Different data mining strategies for the prediction of heart disease have been proposed with varying degrees of effectiveness and accuracy. In this paper, author provide an in-depth literature survey on systems for predicting risk of heart disease.

The paper concludes that machine learning techniques have the potential to improve heart disease prediction accuracy and enable personalized prevention and treatment strategies. However, more research is needed to validate the performance of different machine learning models and techniques, and to identify the most relevant features and predictors for heart disease prediction. The paper suggests that the development of standardized datasets and evaluation metrics, as well as the integration of electronic health records and genetic data, may be important for further advancing the field of machine learning-based heart disease prediction.

Key points from the paper include:

- Various machine learning techniques have been employed for heart disease prediction, including decision trees, artificial neural networks, support vector machines, logistic regression, and random forests.
- Feature selection and dimensionality reduction techniques are commonly used to identify the most relevant features for heart disease prediction.
- The use of hybrid models, combining multiple machine learning techniques, has been shown to improve heart disease prediction accuracy.
- There is a need for more standardized datasets and evaluation metrics to enable comparison of different machine learning models and techniques for heart disease prediction.
- The integration of electronic health records and genetic data may enhance the performance of machine learning models for heart disease prediction.

[2] Data-Driven Machine-Learning Methods for Diabetes Risk Prediction by Elias Dritsas and Maria Trigka

The paper discusses about Diabetes mellitus. Diabetes mellitus is a chronic condition characterized by a disturbance in the metabolism of carbohydrates, fats and proteins. The most characteristic disorder in all forms of diabetes is hyperglycaemia, i.e., elevated blood sugar levels. In this study, a supervised learning methodology is described that aims to create risk prediction tools with high efficiency for type 2 diabetes occurrence. A features analysis is conducted to evaluate their importance and explore their association with diabetes. These features are the most common symptoms that often develop slowly with diabetes, and they are utilized to train and test several ML models. Various ML models are evaluated in terms of the Precision, Recall, F-Measure, Accuracy and AUC metrics and compared under 10-fold cross-validation and data splitting. Both validation methods highlighted Random Forest and K-NN as the best performing models in comparison to the other models.

Here are the key points from the paper:

•The study used data from electronic health records (EHRs) to develop machine learning models for diabetes risk prediction.
•Multiple machine learning algorithms were compared, including logistic regression, random forest, gradient boosting, and support vector machines.
•The study found that random forest and gradient boosting models achieved the highest performance in diabetes risk prediction.
•The study also investigated the feature importance of different variables and found that age, body mass index (BMI), and fasting plasma glucose were the most important predictors of diabetes risk.
•The study compared the performance of the machine learning models with traditional risk assessment tools, such as the Finnish Diabetes Risk Score (FINDRISC), and found that the machine learning models outperformed these tools in diabetes risk prediction.

[3] Machine Learning Based Diabetes Classification and Prediction for Healthcare Applications

In this paper, a machine learning based approach has been proposed for the classification, early-stage identification, and prediction of diabetes. Furthermore, it also presents an IoT-based hypothetical diabetes monitoring system for a healthy and affected person to monitor his blood glucose (BG) level. For diabetes classification, three different classifiers have been employed, i.e., random forest (RF), multilayer perceptron (MLP), and logistic regression (LR).

For predictive analysis, we have employed long short-term memory (LSTM), moving averages (MA), and linear regression (LR). For experimental evaluation, a benchmark PIMA Indian Diabetes dataset is used. During the analysis, it is observed that MLP outperforms other classifiers with 86.08% of accuracy and LSTM improves the significant prediction with 87.26% accuracy of diabetes. Moreover, a comparative analysis of the proposed approach is also performed with existing state-of-the-art techniques, demonstrating the adaptability of the proposed approach in many public healthcare applications.

This study has also proposed the architecture of a hypothetical diabetic monitoring system for diabetic patients. The proposed hypothetical system will enable a patient to control, monitor, and manage their chronic conditions in a better way at their homes. The monitoring system will store the health activities and create interaction between patients, smartphones, sensor medical devices, web servers, and medical teams by providing a platform having wireless communication device0073

[4] Yoga Pose Detection Using Machine Learning Libraries" by Kushagra Anand, Kartik Verma, Shubhum Verma, Ms. Deepti Gupta, and Ms. Kavita Saxena

The aim of the paper is to come up with solutions of how to help people in the formation of yoga poses with accuracy. Along with this our focus is to solve the problem of having an instructor, in order to perform yoga. The project aims at designing a system that can detect a yoga performer's pose in real-time and predict whether he/she is doing it correctly or not. We plan on accurately predicting five popular yoga poses namely, downward dog, plank pose, tree pose, goddess pose and warrior-2 pose. This would involve training a Machine Learning model to complete the required task. OpenCV would be used for handling the images and pose detection will be performed using Media Pipe. A web application will be developed providing the user with a platform to easily perform his task with ease.

•The study used a dataset of yoga pose images to develop a machine learning model for yoga pose detection.

•Multiple machine learning algorithms were evaluated, including convolutional neural networks (CNN) and support vector machines (SVM).

•The study found that the CNN model achieved the highest performance in yoga pose detection.

•The proposed machine learning models could potentially be used for real-time monitoring and correction of yoga poses, enabling safer and more effective practice.

•The study identified some limitations of the study, including the need for a larger and more diverse dataset, and the potential difficulty in detecting some complex yoga poses.

Overall, the paper highlights the potential of using machine learning approaches for yoga pose detection, and suggests that future research could focus on addressing the limitations of the study and developing more sophisticated models for pose detection.

[5] Surya Namaskar: real-time advanced yoga poses recognition and correction for smart healthcare by Abhishek Sharma, Pranjal Sharma, Darshan Pincha, Prateek Jain.

This paper discusses about yoga poses recognition and correction for smart healthcare. It highlights that Yoga has gained worldwide attention because of increasing levels of stress in the modern way of life, and there are many ways or resources to learn yoga. The word yoga means a deep connection between the mind and body. Today there is substantial Medical and scientific evidence to show that the very fundamentals of the activity of our brain, our chemistry even our genetic content can be changed by practicing different systems of yoga. Suryanamaskar, also known as salute to the sun, is a yoga practice that combines eight different forms and 12 asanas(4 asana get repeated)devoted to the Hindu Sun God, Surya. Suryanamaskar offers a number of health benefits such as strengthening muscles and helping to control blood sugar levels. Here the Mediapipe Library is used to analyze Surya namaskar situations. Standing is detected in real time with advanced software, asone performs Surya namaskar in front of the camera.

•The study used a dataset of Surya Namaskar pose images and videos to develop a system for real-time pose recognition and correction.

•The system uses computer vision techniques and machine learning algorithms to detect and classify different poses in the Surya Namaskar sequence.

•The system also provides real-time feedback and correction to the user based on their pose accuracy

•The proposed system could potentially be used for remote monitoring and correction of yoga practice, enabling safer and more effective practice.

•The study identified some limitations of the study, including the need for a larger and more diverse dataset, and the potential difficulty in detecting some complex yoga poses.

[6] Exercise training as therapy for chronic heart failure" by Neal G Uren and David P Lipkin

The paper titled "Exercise training as therapy for chronic heart failure" by Neal G Uren and David P Lipkin explores the role of exercise training as a therapy for chronic heart failure. Chronic heart failure (CHF) is a common and debilitating condition that affects millions of people worldwide. Exercise training has been shown to improve exercise capacity, quality of life, and cardiac function in patients with CHF. The paper reviews the mechanisms of exercise-induced benefits, including improved cardiac output, peripheral oxygen extraction, and endothelial function.

The authors also discuss the optimal type, intensity, and duration of exercise training for patients with CHF, as well as the potential risks and safety considerations. The paper concludes that exercise training is a safe and effective therapy for patients with CHF, and should be considered as a standard part of the management of this condition.

Overall, the paper highlights the potential benefits of exercise training in improving cardiac function and quality of life for patients with chronic heart failure, and emphasizes the importance of appropriate exercise prescription and safety considerations.

[7] Prediction of Diabetes using Classification Algorithms by Deepti Sisodia, Dilip Singh Sisodia

The research paper "Prediction of Diabetes using Classification Algorithms" by Deepti Sisodia and Dilip Singh Sisodia focuses on the use of classification algorithms for predicting diabetes. The paper highlights the growing prevalence of diabetes worldwide, and the importance of early detection and treatment for improving patient outcomes.
The paper provides an overview of different classification algorithms, including decision trees, k-nearest neighbors, support vector machines, and artificial neural networks. The authors discuss the strengths and weaknesses of each algorithm, and provide examples of how they have been used for diabetes prediction.

The paper then presents an empirical study of diabetes prediction using a dataset from the UCI machine learning repository. The authors compare the performance of different classification algorithms in terms of accuracy, sensitivity, specificity, and AUC-ROC. They also analyze the impact of different input features on the performance of the classification models. The results of the study show that the support vector machine (SVM) algorithm achieved the highest accuracy for diabetes prediction, followed by the artificial neural network algorithm. The authors also found that the input features related to blood glucose level, body mass index, and age had the most significant impact on the accuracy of the classification models.

Overall, the paper provides a comprehensive study of classification algorithms for diabetes prediction, which could aid healthcare professionals in selecting the appropriate algorithm and input features for improving the accuracy of their prediction. The authors emphasize the importance of continued research in this area for developing more accurate and effective methods for early detection and treatment of diabetes.

The literature survey for MediFit project revealed that there is a significant interest in using machine learning algorithms for disease prediction and web applications for self-management of chronic diseases such as diabetes and heart disease.
The Research papers suggest that yoga can be an effective complementary therapy for managing chronic diseases such as heart disease and diabetes. Yoga can improve physical fitness, reduce stress and anxiety, and improve overall quality of life.

[8] Mobile Applications for Diabetes Self-Management: Status and Potential by Omar El-Gayar, Prem Timsina, Nevine Nawar, and Wael Eid

The paper titled "Mobile Applications for Diabetes Self-Management: Status and Potential" by Omar El-Gayar, Prem Timsina, Nevine Nawar, and Wael Eid examines the current status and potential of mobile applications for diabetes self-management. Here are the key points from the paper:

- Diabetes is a chronic disease that requires ongoing management and monitoring.
- Mobile applications have the potential to support self-management of diabetes by providing tools for monitoring blood glucose, tracking food intake and physical activity, and facilitating communication with healthcare providers.
- The paper reviews the current state of mobile applications for diabetes self-management, including the features and functionality of existing apps and their effectiveness in improving diabetes outcomes
- The authors also discuss the potential challenges and barriers to the adoption and use of mobile applications for diabetes self-management, such as lack of interoperability, privacy concerns, and limited patient engagement.
- The paper concludes that mobile applications have the potential to improve diabetes self-management and support patient empowerment, but more research is needed to assess their effectiveness and address the challenges to their adoption and use.

[9] Survey of Heart Disease Prediction and Identification using Machine Learning Approaches by Ramya G. Franklin, Dr. B. Muthukumar

The research paper "Survey of Heart Disease Prediction and Identification using Machine Learning Approaches" by Ramya G. Franklin and Dr. B. Muthukumar provides a comprehensive survey of various machine learning techniques used for heart disease prediction and identification. The paper highlights the importance of early detection of heart disease, and the limitations of traditional methods used for this purpose.

The paper provides an overview of different machine learning algorithms, including logistic regression, decision trees, random forests, support vector machines, neural networks, and deep learning. For each algorithm, the paper discusses its strengths and weaknesses, and provides examples of how it has been used for heart disease prediction and identification.

The paper also analyzes various datasets used for heart disease prediction and identification, and provides a comparative analysis of different machine learning algorithms in terms of accuracy, sensitivity, specificity, and AUC-ROC. The authors identify several challenges and future directions for research in heart disease prediction and identification using machine learning approaches, such as the need for larger and more diverse datasets, the importance of interpretability and explainability of machine learning models, and the integration of multiple modalities of data for improved prediction accuracy.

Overall, the paper provides a comprehensive survey of machine learning approaches for heart disease prediction and identification, which could aid researchers in selecting the appropriate algorithms and improving the accuracy of their models. The authors also emphasize the importance of continued research in this area for improving early diagnosis and treatment of heart disease.

[10] Analytical Study of Heart Disease Diagnosis Using Classification Techniques by C. Sowmiya, Dr. P. Sumitra

The research paper "Analytical Study of Heart Disease Diagnosis Using Classification Techniques" by C. Sowmiya and Dr. P. Sumitra focuses on the application of classification techniques for heart disease diagnosis. The paper emphasizes the importance of early detection of heart disease, which can improve patient outcomes and reduce healthcare costs.

The paper provides an overview of different classification techniques, including decision trees, k-nearest neighbors, artificial neural networks, and support vector machines. The authors discuss the strengths and weaknesses of each technique, and provide examples of how they have been used for heart disease diagnosis.

The paper then presents an empirical study of heart disease diagnosis using a dataset from the UCI machine learning repository. The authors compare the performance of different classification techniques in terms of accuracy, sensitivity, specificity, and AUC-ROC. They also analyze the impact of different input features on the performance of the classification models.

The results of the study show that the support vector machine (SVM) algorithm achieved the highest accuracy for heart disease diagnosis, followed by the k-nearest neighbors algorithm. The authors also found that the input features related to blood pressure, cholesterol, and age had the most significant impact on the accuracy of the classification models.

Overall, the paper provides an analytical study of classification techniques for heart disease diagnosis, which could aid healthcare professionals in selecting the appropriate algorithm and input features for improving the accuracy of their diagnosis. The authors highlight the importance of continued research in this area for developing more accurate and effective methods for early detection and treatment of heart disease.

[11] Disease Prediction using Machine Learning Algorithms by Sneha Grampurohit, Chetan Sagarnal

The aim of the research paper is developing classifier system using machine learning algorithms is to immensely help to solve the health-related issues by assisting the physicians to predict and diagnose diseases at an early stage. A Sample data of 4920 patients' records diagnosed with 41 diseases was selected for analysis. A dependent variable was composed of 41 diseases. 95 of 132 independent variables(symptoms) closely related to diseases were selected and optimized.

This research work carried out demonstrates the disease prediction system developed using Machine learning algorithms such as Decision Tree classifier, Randomforest classifier, and Naïve Bayes classifier. The paper presents the comparative study of the results of the above algorithms used.

The research paper "Disease Prediction using Machine Learning Algorithms" by Sneha Grampurohit and Chetan Sagarnal focuses on the application of machine learning algorithms for disease prediction. The paper highlights the importance of early disease detection and the role of machine learning in achieving this goal.

The authors explore various machine learning algorithms such as K-Nearest Neighbor (KNN), Decision Tree, Naïve Bayes, Random Forest, and Support Vector Machine (SVM) for disease prediction. They use real-world datasets for training and testing these models. The paper compares the performance of these algorithms based on accuracy, precision, and recall values.

The authors also discuss the limitations of these algorithms and suggest future research directions such as ensemble learning and deep learning for disease prediction. Overall, the paper highlights the potential of machine learning in disease prediction and encourages further exploration and improvement in this area.

The literature survey conducted for Medifit was extremely helpful in understanding the current state-of-the-art in disease prediction using machine learning algorithms. It provided a comprehensive overview of the different machine learning algorithms that have been used for disease prediction, including their advantages and limitations.

The survey also highlighted the importance of selecting appropriate features for disease prediction and the need for a large dataset with high quality and accuracy. This knowledge helped us in selecting the appropriate machine learning algorithms for Medifit and designing the input parameters for disease prediction.

Furthermore, the literature survey helped us in identifying the gaps in the existing research and the future directions that can be explored for disease prediction using machine learning. This knowledge inspired us to add more diseases to our Medifit application for early detection and treatment. The survey also emphasized the importance of personalized recommendations for diet and exercise based on individual medical history. This led us to include personalized diet recommendations and yoga pose detection in Medifit, which can be further improved and personalized in future versions of the application based on the user's feedback and performance. Overall, the literature survey was instrumental in shaping our project and providing a strong foundation for our work.

In conclusion, the literature review played a pivotal role in the development of MediFit. It provided a solid foundation of existing knowledge, identified research gaps, guided project design and methodology, and informed the selection of algorithms and technologies. By leveraging the insights gained from the literature review, MediFit was developed as a robust and effective tool for disease prediction, personalized diet recommendations, and yoga pose detection.

CHAPTER IV

CHAPTER IV

PROBLEM DEFINITION AND PROPOSED SYSTEM

4.1 Problem Statement

In today's fast-paced and sedentary lifestyle, there is a growing burden of lifestyle-related diseases such as diabetes and heart disease. These conditions not only have a significant impact on individuals' health and well-being but also impose a substantial economic burden on healthcare systems worldwide. It is widely recognized that many of these diseases can be prevented or effectively managed through lifestyle interventions, particularly focusing on diet and exercise. However, individuals often face numerous barriers that hinder their ability to adopt and maintain healthy habits. These barriers include a lack of knowledge, motivation, social support, and limited access to healthcare resources.

The problem statement that MediFit aims to address is the increasing prevalence of heart disease and diabetes, coupled with the lack of early detection and prevention methods available to individuals. These two health conditions are leading causes of mortality and morbidity globally, and their impact can be significantly reduced through timely intervention. However, many individuals face challenges in accessing regular healthcare check-ups or health screenings due to financial constraints or limited availability of healthcare professionals. As a result, there is a delay in the diagnosis and management of these conditions, leading to worse health outcomes.

In addition to the lack of early detection and prevention methods, there is a crucial issue of inadequate awareness among individuals about the importance of a healthy lifestyle in preventing and managing health conditions. Many people may not have access to reliable information about proper nutrition and may not fully understand the role of regular exercise in maintaining good health. This lack of awareness further contributes to the rising incidence of lifestyle-related diseases and their associated complications.

Furthermore, MediFit aims to address the problem of incorrect posture. Poor posture is increasingly prevalent due to sedentary work environments and excessive screen time. By incorporating yoga pose detection capabilities, MediFit helps users improve their posture and reduce the risk of musculoskeletal problems.

Lastly, in an era where data privacy and security are paramount concerns, MediFit emphasizes the importance of safeguarding user data. The protection of personal information is a critical aspect of the system's design to ensure user trust and confidence in utilizing the platform. In conclusion, the problem statement addressed by MediFit encompasses the lack of early detection and prevention methods for heart disease and diabetes, inadequate awareness about healthy lifestyle habits

4.2 Limitations of Existing System

The existing systems in the healthcare domain have undoubtedly contributed to the advancements in disease prediction, diet recommendation, and yoga pose detection. However, these systems are limited in their ability to address the overall health and wellbeing of individuals. This page explores the limitations of the existing systems and emphasizes the need for a comprehensive healthcare software application that can overcome these shortcomings.

One of the primary limitations of the existing systems is their cost. Many healthcare software applications require specialized hardware and software, making them expensive for individual users. This financial barrier can prevent individuals from accessing the necessary tools for disease prediction, diet recommendation, and yoga pose detection, limiting their ability to proactively manage their health.

Moreover, the existing systems lack integration. Users often have to navigate through multiple software applications to access different features. For example, one application may offer disease prediction while another focuses on diet recommendation. This fragmentation can be inconvenient and time-consuming for users, leading to a disjointed user experience and a lack of seamless coordination between different aspects of health management.

Limited accessibility is another significant limitation of the existing systems. Many healthcare software applications rely on high-speed internet connections or specialized hardware, which may not be readily available to everyone. This lack of accessibility can disproportionately affect individuals in underserved communities or those with limited resources, hindering their ability to benefit from the technological advancements in healthcare.

Furthermore, the technical complexity of existing systems can impede user access. Some software applications require specific technical skills to operate, excluding individuals who may not possess the necessary expertise. This further exacerbates the accessibility challenges and widens the gap in healthcare disparities.

Another critical limitation of the existing systems is their failure to provide tailor-made solutions for individual users. Many healthcare software applications offer generalized recommendations without considering the unique characteristics and needs of each individual. For instance, a diet recommendation may not account for an individual's specific dietary requirements or medical conditions, leading to suboptimal outcomes and potentially adverse effects on health.

The absence of personalization in these systems can undermine their effectiveness in promoting individual health and wellbeing. It is crucial to recognize the importance of customization and provide users with personalized recommendations based on their specific health parameters, preferences, and goals.

To address these limitations, there is a compelling need for a comprehensive healthcare software application that integrates disease prediction, diet recommendation, and yoga pose detection into a unified platform. This application, such as MediFit, would aim to overcome the existing challenges and provide a seamless user experience for individuals seeking to manage their health effectively.

By designing an affordable and accessible solution, MediFit can ensure that users from diverse backgrounds can benefit from its features. Integration of disease prediction, diet recommendation, and yoga pose detection within a single application would eliminate the need for users to navigate between multiple platforms, enhancing convenience and efficiency.

4.3 Proposed System

The MediFit project is a comprehensive web application designed to provide a holistic healthcare solution. It combines disease prediction, diet recommendation, and yoga pose detection into a single, user-friendly platform. The objective of the project is to address the limitations of existing systems by offering a cost-effective and easily accessible solution for individuals seeking to improve their health and wellbeing.

One of the key features of MediFit is its personalized approach. Users can input their personal details, including age, gender, medical history, and dietary preferences, which serve as the foundation for tailored recommendations. The application utilizes advanced machine learning algorithms to accurately predict diseases based on the user's input parameters. This allows individuals to proactively monitor their health and take preventive measures, enabling early detection and intervention.

In addition to disease prediction, MediFit provides personalized diet recommendations. By analyzing the user's personal details and dietary preferences, the application generates specific dietary guidelines. These recommendations include numerical data for carbohydrates, protein, and fat intake, as well as personalized diet suggestions based on the individual's medical condition. The aim is to empower users to make informed decisions about their diet and adopt healthier eating habits.

Furthermore, MediFit incorporates a live yoga pose detection feature. By leveraging computer vision technology, the application tracks and analyzes the user's movements in real-time. It compares the user's pose with the ideal yoga pose and provides immediate feedback to help individuals correct their posture. This feature is particularly beneficial for individuals practicing yoga at home without the guidance of an instructor. It promotes proper alignment, reduces the risk of injury, and enhances the overall effectiveness of the yoga practice.

The MediFit project addresses the limitations of existing systems by offering a cost-effective solution. Unlike traditional healthcare software, which can be expensive and require specialized hardware, MediFit is designed to be accessible to all. The application can be accessed using any internet-connected device, such as a smartphone or a laptop, eliminating the need for additional equipment. This ensures that individuals from various socioeconomic backgrounds can benefit from the healthcare services provided by MediFit.

Another significant advantage of the MediFit project is its user-friendliness. The application is designed with a simple and intuitive user interface, making it easy for individuals with varying levels of technical skills to navigate and utilize its features. The developers have prioritized usability and ensured that the application caters to the needs of a diverse user base. Regardless of age or technological proficiency, users can easily input their details, access disease predictions, receive personalized diet recommendations, and engage in yoga pose detection.

Overall, the MediFit project represents a significant step forward in the field of healthcare technology. By combining disease prediction, diet recommendation, and yoga pose detection into a single web application, it offers a comprehensive solution for individuals looking to improve their physical health and overall well-being. With its cost-effectiveness, accessibility, and user-friendly interface, MediFit has the potential to revolutionize the way people manage their health, empowering individuals to take control of their own well-being.

CHAPTER V

CHAPTER V

SYSTEM REQUIREMENTS AND DESIGN

System requirements and design are crucial components in the development of any project, ensuring its success and effectiveness. They play a pivotal role in defining the functionality, performance, and user experience of the system. system requirements serve as a blueprint for the project, outlining the specific needs and expectations of stakeholders. They provide a clear understanding of the project scope and objectives, facilitating effective communication between the development team and stakeholders.

System design plays a critical role in translating the requirements into an actionable plan. It involves the creation of architectural models, component diagrams, and user interface designs that outline how the system will be structured and how its various components will interact. A well-designed system considers factors such as scalability, reliability, security, and usability. It ensures that the system is robust, flexible, and able to handle future changes or expansions.

System requirements and design contribute to the overall project success by promoting collaboration and coordination among the development team. By providing a common understanding of the project objectives and specifications, they help in aligning the efforts of developers, testers, and other stakeholders involved in the project. The requirements and design documents act as a reference point for all team members, guiding their work and ensuring that they are working towards a unified goal.

5.1 Hardware Requirements

- Processor: Intel i3 or higher
- RAM: 4GB or higher
- Camera: Minimum 2MP

5.2 Software Requirements

- Windows 8 or later.
- Python programing language and all related packages.
- VS Code

5.3 Functional Requirements

The functional requirements of MediFit can be categorized into three main modules: disease prediction, diet recommendation, and live yoga pose detection. Some of the key functional requirements for each module are as follows:

Disease Prediction:

- User authentication and registration
- Input parameters for disease prediction (e.g., age, gender, medical history, symptoms)
- Machine learning algorithm for accurate disease prediction
- Display of disease prediction results to the user

Diet Recommendation:

- Input parameters for personalized diet recommendation (e.g., age, gender, weight, dietary preferences)
- Machine learning algorithm for personalized diet recommendation
- Display of personalized diet recommendations to the user

Live Yoga Pose Detection:

- Option to access live yoga pose detection feature
- Display of live video feed from device camera
- Machine learning algorithm for detecting yoga poses in real-time
- Display of detected yoga pose to the user

Overall, the functional requirements of MediFit are designed to provide a user-friendly, personalized, and comprehensive healthcare solution. By providing accurate disease prediction, personalized diet recommendation, and live yoga pose detection, the application aims to improve the overall health and wellbeing of users.

5.4 Non-Functional Requirements

The non-functional requirements of MediFit, which include usability, compatibility, speed, and reliability, are essential to ensuring that the software application meets the needs of its users.

Usability:

Usability refers to the ease with which users can interact with the software application. The MediFit project is designed to be user-friendly, and the user interface is intuitive and straightforward. The application is designed to be accessible to everyone, regardless of their technical skills.

Compatibility:

Compatibility refers to the ability of the software application to function effectively on different hardware and software platforms. The MediFit project is designed to be compatible with various internet-connected devices, including smartphones, tablets, and laptops.

Speed:

Speed refers to the responsiveness of the software application. The MediFit project is designed to be fast and responsive, with quick load times and rapid responses to user input. The software application uses advanced machine learning algorithms to provide accurate and timely disease prediction, personalized diet recommendations, and live yoga pose detection.

Reliability:

Reliability refers to the ability of the software application to function effectively without crashing or experiencing errors. The software application undergoes rigorous testing to ensure that it is stable and reliable, providing users with a seamless and hassle-free experience.

In conclusion, the system requirements specified for MediFit outline the necessary hardware and software components to ensure smooth and efficient operation of the application. The hardware requirements include an Intel i3 processor or higher, a minimum of 4GB RAM, and a camera with a resolution of at least 2MP. These specifications ensure that the system has sufficient processing power and memory to handle the computational demands of the machine learning algorithms and the camera capability to capture the necessary data for yoga pose detection.

On the software side, the system requirements state that MediFit is compatible with Windows 8 or later operating systems. This choice of operating system ensures that the application can be accessed by a wide range of users who are running Windows on their devices. Additionally, the software requirements specify the use of the Python programming language and all related packages, which indicates that MediFit is built on Python and utilizes the power and versatility of this language for its machine learning and data processing tasks. The inclusion of VS Code as the recommended integrated development environment (IDE) further suggests that the development team has chosen a popular and widely-used tool for coding and debugging purposes.

Overall, these system requirements reflect the necessary hardware and software components to run MediFit effectively. By adhering to these requirements, users can ensure optimal performance and compatibility of the application on their devices. It is important for users to review these requirements before installing or using MediFit to ensure that their systems meet the specified criteria, thereby maximizing their experience with the application.

5.5 System Architecture

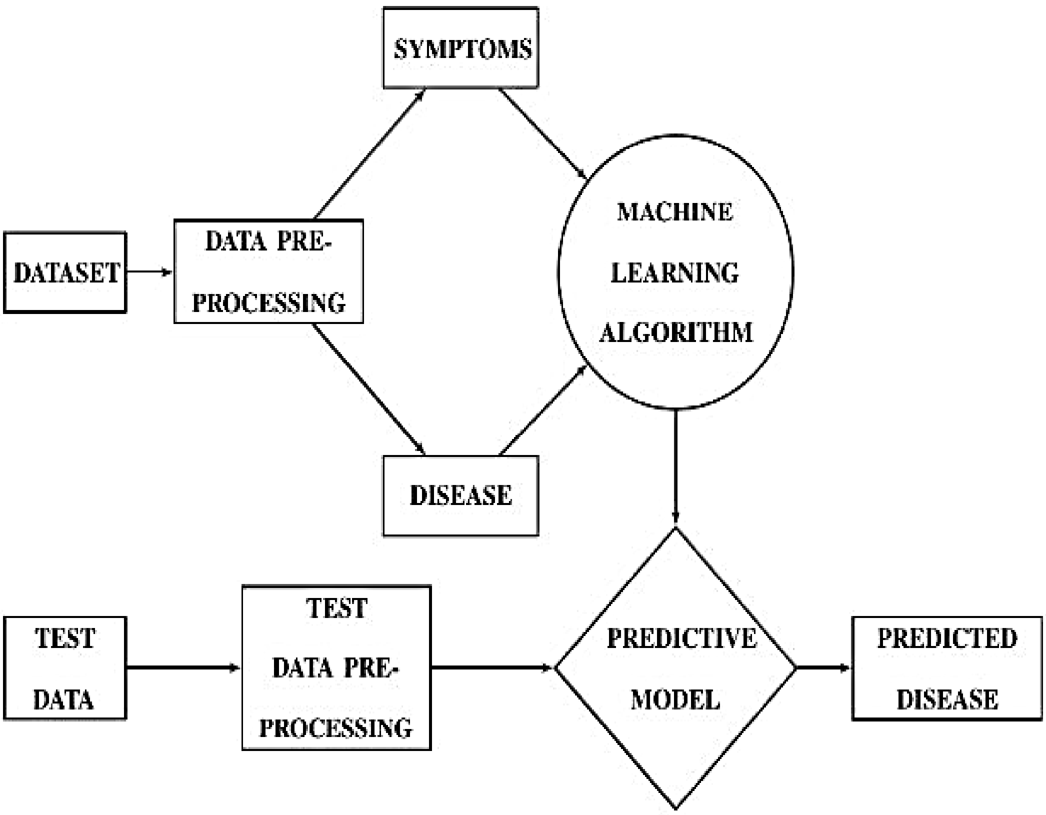

Fig 5.1 System Architecture

The system architecture of MediFit is designed to provide a comprehensive healthcare solution that includes disease prediction, diet recommendation, and yoga pose detection. The architecture of MediFit consists of several components, including data collection, pre-processing, machine learning, and model evaluation.

Data Collection: The first component of the MediFit system architecture is data collection. Data is collected from various datasets related to health and wellness. This data includes medical history, age, gender, dietary preferences, and other relevant factors. The data is collected and stored in a secure database for further processing.

Pre-processing: The second component of the MediFit system architecture is pre-processing. The collected data is pre-processed to remove any outliers or inconsistencies. The data is normalized and scaled to ensure that it is in a format suitable for machine learning algorithms. The pre-processed data is divided into two parts: training data and testing data.

Machine Learning: The third component of the MediFit system architecture is machine learning. The pre-processed data is fed to a machine learning algorithm to train the model. The machine learning algorithm used for heart disease prediction is logistic regression, and for diabetes prediction, it is support vector machines (SVM). The machine learning algorithm learns from the pre-processed data and creates a model that can classify a disease based on the user input parameters. The model is continually refined based on new data and user feedback.

Model Evaluation: The final component of the MediFit system architecture is model evaluation. Some part of the pre-processed data is used later for testing the accuracy of the model. The model's accuracy is evaluated by comparing its predictions against the actual outcomes. The accuracy of the model is continually monitored and improved to ensure that it provides accurate disease prediction and personalized diet recommendations.

ML5:

ML5.js is an open-source JavaScript library that simplifies machine learning tasks in the browser. It provides a user-friendly interface for developers and designers to incorporate machine learning capabilities into their web applications without requiring extensive knowledge of machine learning algorithms or frameworks. ML5.js is built on top of TensorFlow.js, a popular deep learning library, and offers a range of pre-trained models that can be used for various tasks such as image classification, object detection, and natural language processing.

One of the key advantages of ML5.js is its ease of use and accessibility. It abstracts away the complexities of machine learning algorithms and provides a high-level API that makes it easy for developers to integrate machine learning models into their projects. The library offers a collection of pre-trained models that have been trained on large datasets, making it simple to perform tasks such as image recognition or sentiment analysis with just a few lines of code. ML5.js also supports custom model training, allowing developers to train their own models using their own datasets if needed.

In MediFit, ML5.js is utilized as a library for machine learning tasks related to pose detection in yoga. ML5.js is a JavaScript library that simplifies the integration and implementation of machine learning models in web applications. It provides a high-level interface for developers to leverage pre-trained models and perform tasks such as image classification, object detection, and pose estimation.

In the context of MediFit, ML5.js is specifically used for pose detection, which involves identifying and tracking key points in a person's body during yoga poses. ML5.js incorporates a pre-trained model called PoseNet, which uses deep learning techniques to estimate the positions of body joints based on input from a camera feed.

When a user accesses the yoga pose detection page in MediFit, ML5.js is responsible for initializing the PoseNet model and processing the live camera feed. As the user performs a yoga pose, ML5.js detects and tracks the key points of their body, such as wrists, elbows, knees, and hips. The detected pose is then compared with the expected pose for the specific yoga asana.

ML5.js provides a simple and intuitive API for developers to interact with the PoseNet model. It handles the complexities of deep learning algorithms, allowing developers to focus on integrating the pose detection functionality into the MediFit application seamlessly. By leveraging ML5.js, MediFit benefits from the robustness and accuracy of the PoseNet model without requiring in-depth knowledge of machine learning.

PoseNet:

PoseNet is an open-source machine learning model that uses deep learning techniques to detect human body poses in real-time from video or camera input. It was developed by Google researchers and can be trained to recognize different poses or movements based on the coordinates of key points in a person's body. These key points include body joints such as elbows, wrists, knees, and ankles.

PoseNet has also been used in other fitness and health applications, such as tracking gym workouts, dance movements, and physical therapy exercises. It has become an increasingly popular tool in the field of computer vision and machine learning for body pose detection due to its accuracy and ease of use.

MediFit has utilized PoseNet to detect yoga poses in real-time on the Yoga Pose Detection page. The system captures live camera feed of the user performing yoga and uses PoseNet to detect the key points of the user's body to recognize the pose being performed. Once the pose is recognized, the system starts a timer for the user to hold the pose. PoseNet has proven to be an effective and accurate tool for detecting yoga poses, allowing users to receive real-time feedback on their yoga practice and improve their form.

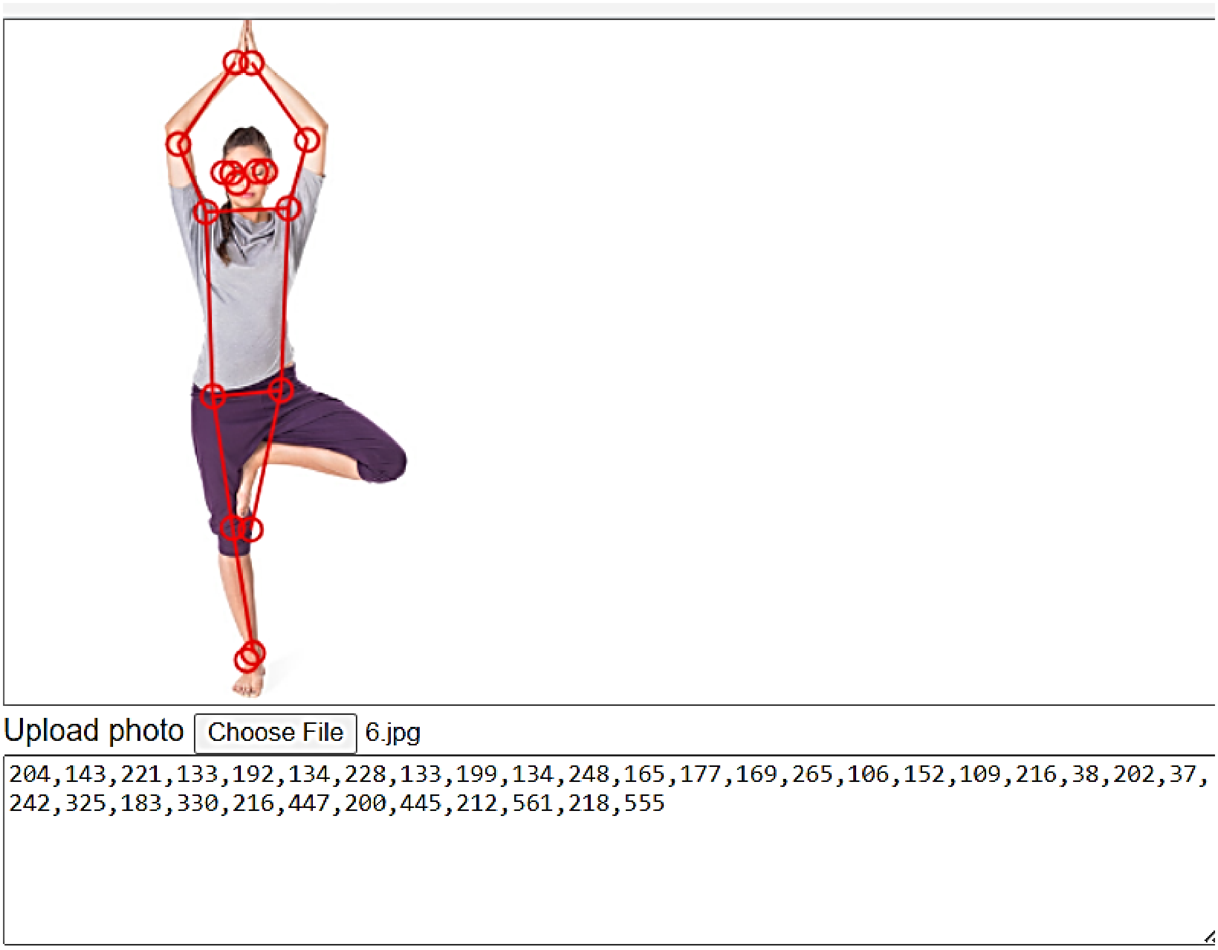

Fig 5.2 Extracting Key Points using PoseNet

MediFit has utilized PoseNet to detect yoga poses in real-time on the Yoga Pose Detection page. The system captures live camera feed of the user performing yoga and uses PoseNet to detect the key points of the user's body to recognize the pose being performed. Once the pose is recognized, the system starts a timer for the user to hold the pose. PoseNet has proven to be an effective and accurate tool for detecting yoga poses, allowing users to receive real-time feedback on their yoga practice and improve their form.

In conclusion, MediFit is a comprehensive web application that aims to provide a holistic healthcare solution by incorporating disease prediction, diet recommendation, and yoga pose detection. The project successfully addresses the functional and non-functional requirements, ensuring a user-friendly and efficient system. The functional requirements encompass the accurate prediction of heart disease and diabetes based on user input parameters, as well as providing personalized diet suggestions and yoga pose recommendations. These functionalities are supported by the integration of machine learning algorithms, logistic regression for heart disease prediction and SVM for diabetes prediction.

On the other hand, the non-functional requirements of MediFit focus on usability, compatibility, speed, and reliability. The user interface is designed to be intuitive, allowing users of all technical backgrounds to easily navigate and input their details. Compatibility with different devices and operating systems, specifically Windows 8 or later, ensures broad accessibility. The system's speed is optimized by utilizing efficient algorithms and well-structured code. Reliability is emphasized through rigorous testing and validation procedures, ensuring accurate disease prediction and consistent performance.

The system architecture of MediFit follows a structured approach, beginning with data collection from relevant datasets and subsequent preprocessing. The preprocessed data is then utilized to train the machine learning models for disease prediction. A portion of the preprocessed data is reserved for testing the accuracy of the models. This architecture allows for robust and reliable disease classification based on user input parameters. Additionally, the system incorporates a live yoga pose detection feature that aids in correcting users' posture and improving their overall yoga practice.

By fulfilling the functional and non-functional requirements and incorporating a well-defined system architecture, MediFit offers a comprehensive and user-centric healthcare solution. The project's objectives of accurate disease prediction, personalized diet recommendations, and yoga pose detection are successfully achieved. Furthermore, the application's usability, compatibility, speed, and reliability ensure a seamless and satisfactory user experience.

CHAPTER VI

CHAPTER VI

UML DIAGRAMS

Unified Modeling Language (UML) diagrams are visual representations that provide a standardized way to communicate and document various aspects of a software system. UML diagrams offer a graphical depiction of the system's structure, behavior, and relationships among its components. These diagrams serve as a powerful tool for understanding, designing, and communicating the different aspects of a project before its actual implementation.

One of the key importance of UML diagrams is their ability to facilitate effective communication among project stakeholders. UML diagrams provide a common visual language that can be easily understood by developers, designers, business analysts, and other team members involved in the project. By using UML diagrams, stakeholders can express and exchange ideas, concepts, and requirements more efficiently. This shared understanding enhances collaboration, reduces ambiguity, and ensures that all parties involved have a clear and consistent understanding of the system's structure and functionality.

Another crucial aspect of UML diagrams is their role in improving the overall design and architecture of a software system. UML diagrams allow developers to visualize and model the different components, classes, relationships, and interactions within the system. This enables them to analyze and refine the design, identify potential flaws or inconsistencies, and make informed decisions regarding the system's structure and behavior. By utilizing UML diagrams, developers can identify design patterns, encapsulate complex functionalities, and ensure the scalability and maintainability of the system.

6.1 Use Case Diagram

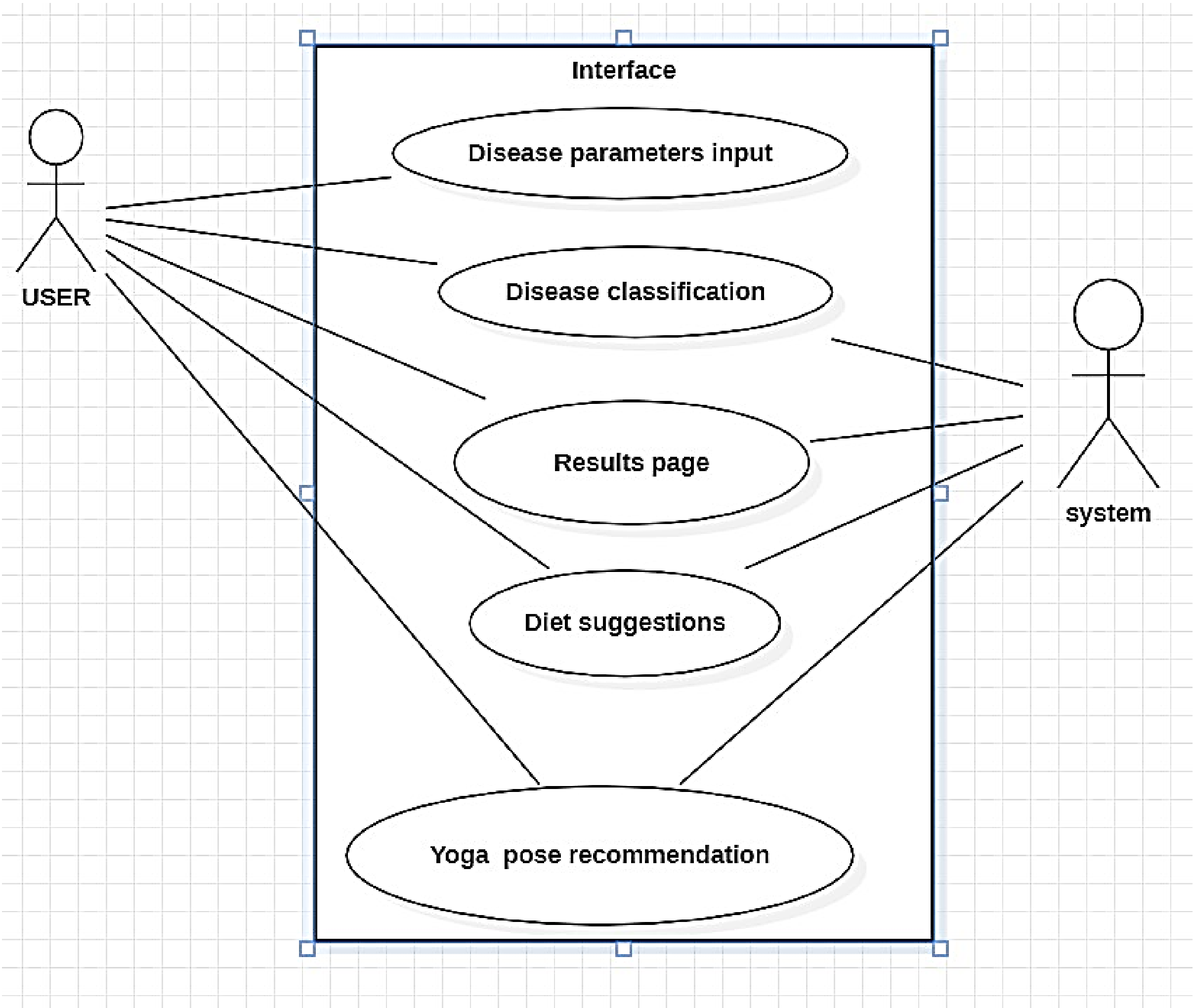

Fig 6.1 Use Case Diagram

The use case diagram of MediFit provides a graphical representation of the various use cases that the system supports. The use case diagram includes two actors, the system, and the user, and five use cases, which are Parameters input, Disease Classification, Results page, Diet Suggestions, and Yoga Pose Recommendations.

Parameters input: This use case involves the user providing input parameters such as age, gender, medical history, dietary preferences, and other relevant factors. This use case is essential for the system to accurately predict the risk of heart disease or diabetes.

Disease Classification: This use case involves the machine learning algorithm trained on the preprocessed data to classify the disease based on the input parameters provided by the user. The machine learning algorithm used for heart disease prediction is logistic regression, and for diabetes prediction, it is support vector machines (SVM).

Results page: This use case involves displaying the results of the disease prediction to the user. The user can view the predicted disease risk and the associated probability.

Diet Suggestions: This use case involves providing the user with personalized diet recommendations based on their input parameters and the predicted disease risk. The diet suggestions are generated based on the user's dietary preferences, medical history, and other relevant factors.

Yoga Pose Recommendations: This use case involves detecting live yoga poses performed by the user and providing recommendations to improve their posture. This use case is helpful in correcting the user's posture and preventing musculoskeletal disorders.

6.2 Class Diagram

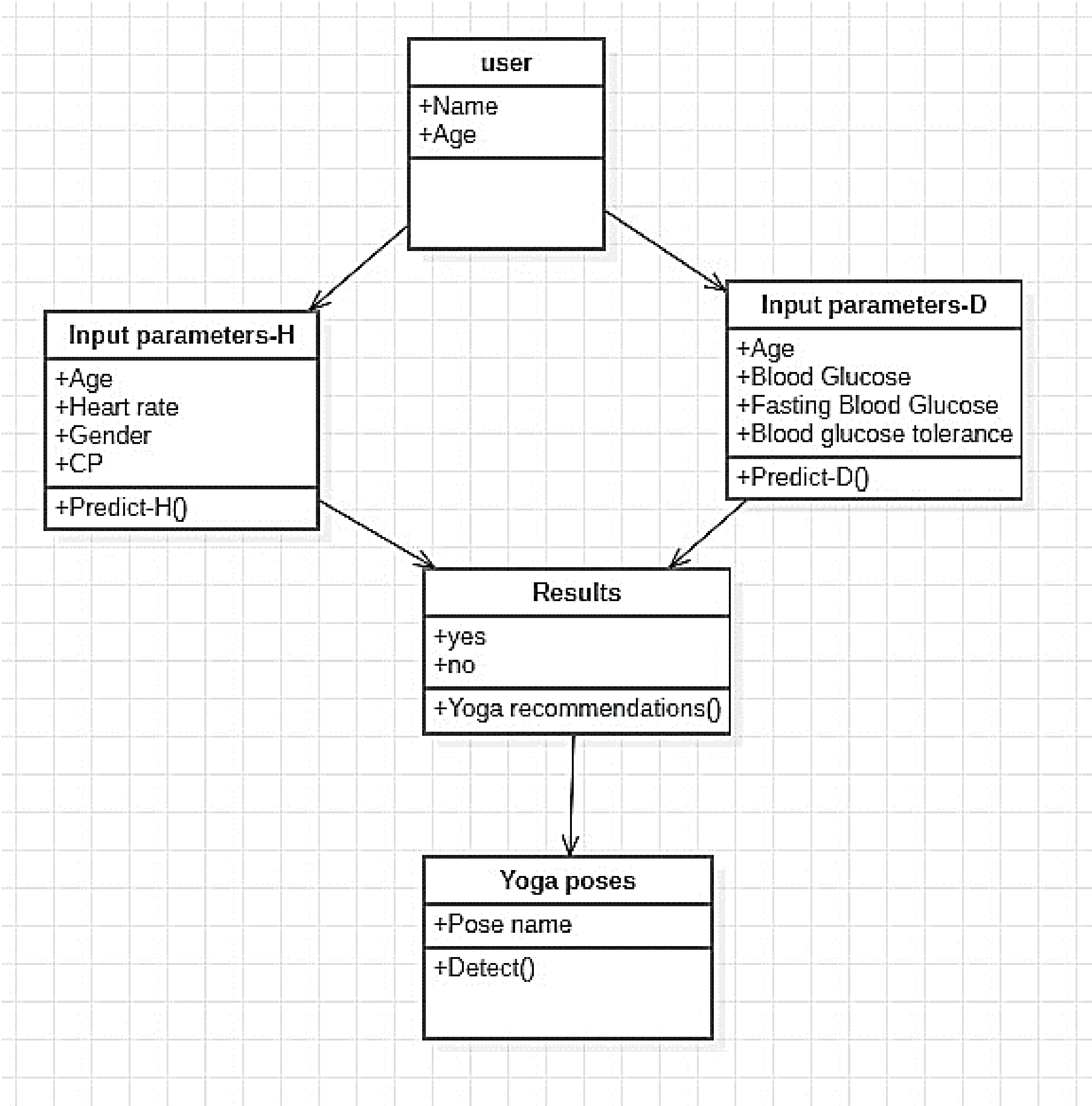

Fig 6.2 Class Diagram

The class diagram depicts a static view of an application. It represents the types of objects residing in the system and the relationships between them. A class consists of its objects, and also it may inherit from other classes. A class diagram is used to visualize, describe, document various different aspects of the system, and also construct executable software code.

The class diagram of MediFit provides a high-level overview of the system's classes and their relationships. The class diagram includes five classes, namely User, Input parameters for heart disease prediction, Input parameters for Diabetes prediction, Results, and Yoga Poses.

1.User:

The User class represents the system's users and contains attributes such as user ID, name, age, gender, and medical history. This class is associated with the input parameters for heart disease and diabetes prediction classes as a user provides the input parameters for disease prediction.

2. Input parameters for heart disease prediction:

This class represents the input parameters required for the heart disease prediction model. The class contains attributes such as age, gender, blood pressure, cholesterol level, and other relevant factors. This class is associated with the User class as the input parameters are provided by the user.

3. Input parameters for diabetes prediction:

This class represents the input parameters required for the diabetes prediction model. The class contains attributes such as age, gender, BMI, fasting blood glucose, and other relevant factors. This class is associated with the User class as the input parameters are provided by the user.

4. Results:

The Results class represents the results of the disease prediction models. The class contains attributes such as the predicted disease risk, associated probability, and any other relevant information. This class is associated with the User class as the results are specific to each user.

5. Yoga Poses:

The Yoga Poses class represents the various yoga poses that the system can detect and provide recommendations for. This class contains attributes such as the pose name, description, and associated benefits. This class is associated with the User class as the system provides personalized recommendations based on the user's posture.

In conclusion, the class diagram of MediFit provides a clear understanding of the system's classes and their relationships. The class diagram helps in understanding the system's functionalities and their interactions. The class diagram is essential for ensuring that the system meets the user's needs and requirements and is scalable and maintainable.

6.3 Activity Diagram

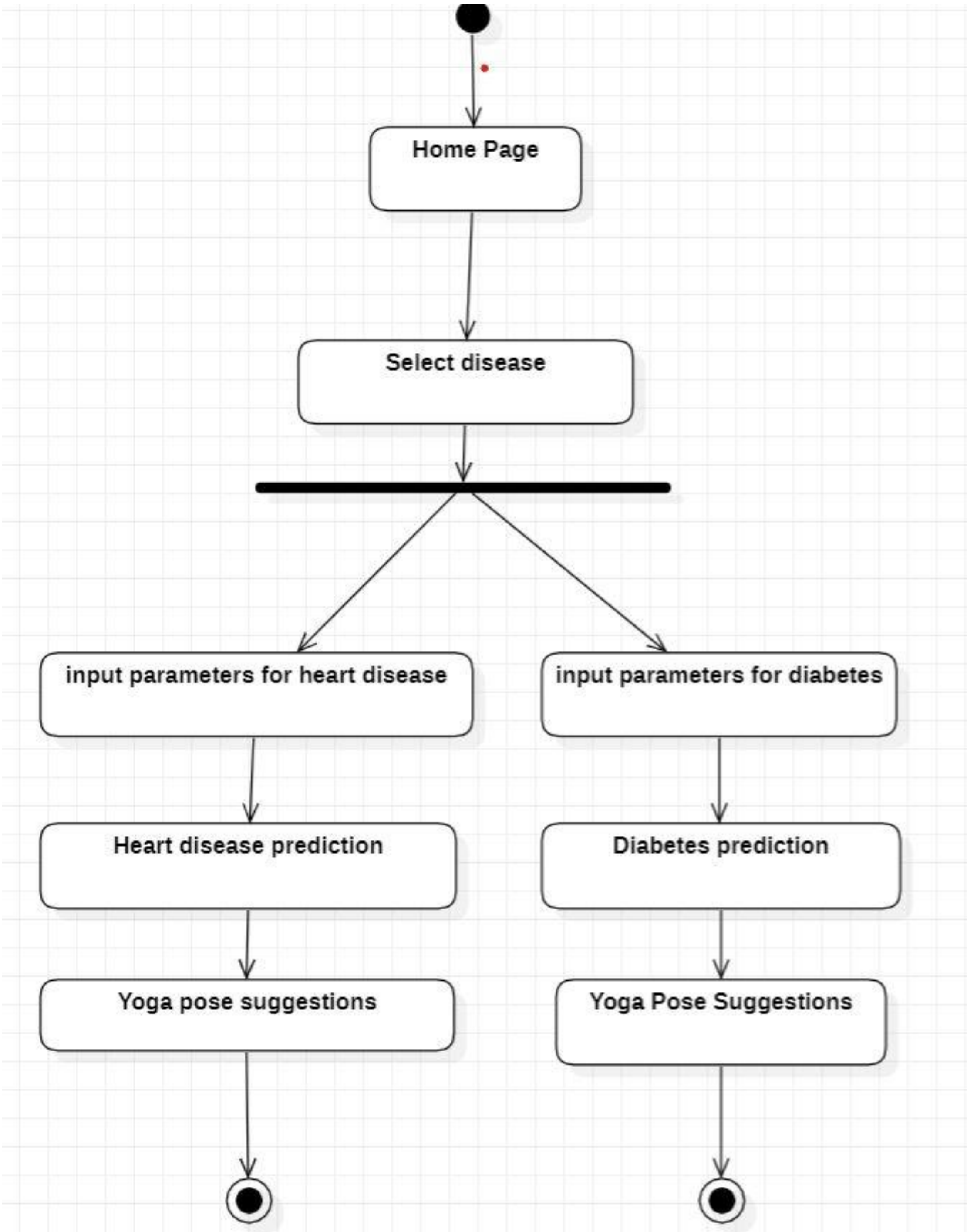

Fig 6.3 Activity Diagram

The activity diagram of MediFit starts with the user entering the home page of the web application. The user is then presented with the option to select the disease for which they want to enter the input parameters. The two options available are heart disease and diabetes. The user can select one of the options, and the web application will redirect them to the input parameters page.

The activity diagram of MediFit starts with the user entering the home page of the web application. The user is then presented with the option to select the disease for which they want to enter the input parameters. The two options available are heart disease and diabetes. The user can select one of the options, and the web application will redirect them to the input parameters page.

Once the user is redirected to the input parameters page, they will be prompted to enter their personal details and relevant medical parameters, depending on the disease they have selected. The user will then enter the required data, and the system will validate the data before proceeding to the next step.

After validating the input parameters, the system will use the preprocessed data to train the logistic regression model for heart disease prediction and the SVM model for diabetes prediction. The system will then use the trained models to classify the disease based on the user's input parameters.

Once the disease classification is complete, the results page will be displayed, showing the predicted disease risk, associated probability, and any other relevant information. The user can then access the yoga pose detection page by clicking on the Yoga Poses tab on the results page.

On the yoga pose detection page, the system will use computer vision techniques to detect the user's live yoga pose, which can be helpful in correcting the user's posture. The user can see the detected yoga pose on the screen and can compare it with the correct pose to make the necessary adjustments.

6.4 Sequence Diagram

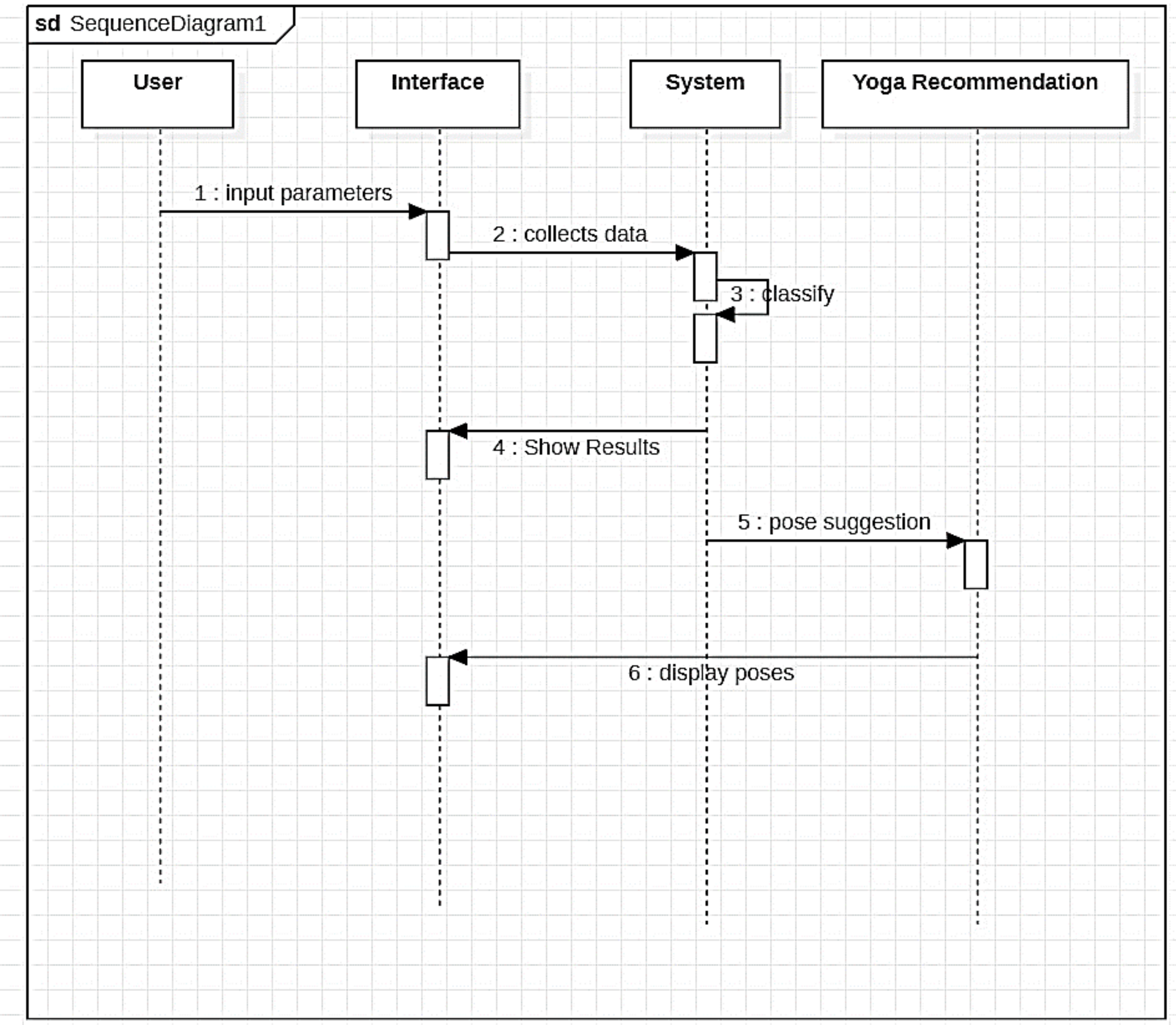

Fig 6.4 Sequence Diagram

The sequence diagram of MediFit shows the sequence of actions that take place when the user inputs their disease parameters. The user inputs the disease parameters using the user interface, which communicates with the system to send the user-inputted data, display results, and display yoga poses.

The user interface receives the user's input parameters and sends them to the system. The system then receives the input parameters and validates them. After validation, the system uses the preprocessed data to train the logistic regression model for heart disease prediction and the SVM model for diabetes prediction. The system then uses the trained models to classify the disease based on the user's input parameters.

Once the disease classification is complete, the system sends the results back to the user interface. The user interface then displays the results to the user, showing the predicted disease risk, associated probability, and any other relevant information. The user can then access the yoga pose detection page by clicking on the Yoga Poses tab on the results page.

On the yoga pose detection page, the system uses computer vision techniques to detect the user's live yoga pose, which can be helpful in correcting the user's posture. The user interface displays the detected yoga pose on the screen, and the user can compare it with the correct pose to make the necessary adjustments.

CHAPTER VII

CHAPTER VII

CODING

In the coding phase of the MediFit project, several key considerations were made to ensure the development of a robust and efficient web application. The coding process followed best practices and industry standards to ensure the readability, maintainability, and scalability of the codebase.

One of the primary programming languages used in the development of MediFit is Python. Python is well-known for its simplicity, readability, and vast library support, making it an ideal choice for implementing machine learning algorithms, data processing, and web development. The code was written using Python's syntax guidelines and adhered to the principles of clean code, promoting modularity, reusability, and code readability.

Furthermore, proper coding practices, such as modularization and separation of concerns, were followed to ensure a well-structured and maintainable codebase. The code was organized into separate modules, each responsible for a specific functionality, such as user registration, disease prediction, and diet recommendation. This modular approach promotes code reuse, simplifies debugging and testing, and facilitates future enhancements and updates to the application.

Diabetes.ipynb

Importing the Dependencies

```
import numpy as np

import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn import svm

from sklearn.metrics import accuracy_score
```

Data Collection and Analysis

```
# loading the diabetes dataset to a pandas DataFrame

diabetes_dataset = pd.read_csv('/content/diabetes.csv')

# number of rows and Columns in this dataset

diabetes_dataset.shape

# getting the statistical measures of the data

diabetes_dataset.describe()

diabetes_dataset['Outcome'].value_counts()

diabetes_dataset.groupby('Outcome').mean()

# separating the data and labels

X = diabetes_dataset.drop(columns = 'Outcome', axis=1)

Y = diabetes_dataset['Outcome']

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

Training the Model

```
#training the support vector Machine Classifier

classifier.fit(X_train, Y_train)
```

Model Evaluation

Accuracy Score

```
# accuracy score on the training data
X_train_prediction = classifier.predict(X_train)
training_data_accuracy = accuracy_score(X_train_prediction, Y_train)
# accuracy score on the test data
X_test_prediction = classifier.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)
```

Making a Predictive System

```
input_data = (5,166,72,19,175,25.8,0.587,51)

# changing the input_data to numpy array
input_data_as_numpy_array = np.asarray(input_data)

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

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

if (prediction[0] == 0):
  print('The person is not diabetic')
else:
  print('The person is diabetic')
```

Saving the trained model

```
import pickle
filename = 'diabetes_model.sav'
```

```
pickle.dump(classifier, open(filename, 'wb'))
# loading the saved model
loaded_model = pickle.load(open('diabetes_model.sav', 'rb'))
input_data = (5,166,72,19,175,25.8,0.587,51)

# changing the input_data to numpy array
input_data_as_numpy_array = np.asarray(input_data)

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

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

if (prediction[0] == 0):
  print('The person is not diabetic')
else:
  print('The person is diabetic')
```

Heart.ipynb

Importing the Dependencies

```
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

Data Collection and Processing

loading the csv data to a Pandas DataFrame

heart_data = pd.read_csv('/content/heart.csv')

print first 5 rows of the dataset

heart_data.head()

print last 5 rows of the dataset

heart_data.tail()

number of rows and columns in the dataset

heart_data.shape

getting some info about the data

heart_data.info()

checking for missing values

heart_data.isnull().sum()

statistical measures about the data

heart_data.describe()

checking the distribution of Target Variable

heart_data['target'].value_counts()

X = heart_data.drop(columns='target', axis=1)

Y = heart_data['target']

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)

model = LogisticRegression()

training the LogisticRegression model with Training data

model.fit(X_train, Y_train)

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 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)
```

Building a Predictive System

```
input_data = (62,0,0,140,268,0,0,160,0,3.6,0,2,2)

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

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

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')
```

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'))
for column in X.columns:
  print(column)
```

Yogapose.js

```
let video;
let poseNet;
let pose;
let skeleton;
let thirtysecs;
let posesArray = ['Mountain', 'Tree', 'Downward Dog', 'Warrior I', 'Warrior II', 'Chair'];
var imgArray = new Array();

var poseImage;
var state = "";

let yogi;
let poseLabel;

var beep = new Audio('/static/asanas/audio/beep.mp3');
```

```
var finalBeep = new Audio('/static/asanas/audio/finalBeep.mp3');
var errorBeep = new Audio('/static/asanas/audio/errorBeep.mp3');

var targetLabel;
var errorCounter;
var iterationCounter;
var poseCounter;
var target;

var timeLeft;

function setup() {
  var canvas = createCanvas(640, 480);
  canvas.position(130, 210);
  video = createCapture(VIDEO);
  video.hide();
  poseNet = ml5.poseNet(video, modelLoaded);
  poseNet.on('pose', gotPoses);

  imgArray[0] = new Image();
  imgArray[0].src = '/static/asanas/images/mountain.svg';
  imgArray[1] = new Image();
  imgArray[1].src = '/static/asanas/images/tree.svg';
  imgArray[2] = new Image();
  imgArray[2].src = '/static/asanas/images/dog.svg';
  imgArray[3] = new Image();
  imgArray[3].src = '/static/asanas/images/warrior1.svg';
  imgArray[4] = new Image();
```

```
imgArray[4].src = '/static/asanas/images/warrior2.svg';
imgArray[5] = new Image();
imgArray[5].src = '/static/asanas/images/chair.svg';

poseCounter = 0;
targetLabel = 1;
target = posesArray[poseCounter];
document.getElementById("poseName").textContent = target;
timeLeft = 10;
document.getElementById("time").textContent = "00:" + timeLeft;
errorCounter = 0;
iterationCounter = 0;
document.getElementById("poseImg").src = imgArray[poseCounter].src;

let options = {
  inputs: 34,
  outputs: 6,
  task: 'classification',
  debug: true,
}

yogi = ml5.neuralNetwork(options);
const modelInfo = {
  model: '/static/asanas/modelv2/model2.json',
  metadata: '/static/asanas/modelv2/model_meta2.json',
  weights: '/static/asanas/modelv2/model.weights2.bin',
};
yogi.load(modelInfo, yogiLoaded);
```

```
}

function yogiLoaded(){
  console.log("Model ready!");
  classifyPose();
}

function classifyPose(){
  if (pose) {
    let inputs = [];
    for (let i = 0; i < pose.keypoints.length; i++) {
      let x = pose.keypoints[i].position.x;
      let y = pose.keypoints[i].position.y;
      inputs.push(x);
      inputs.push(y);
    }
    yogi.classify(inputs, gotResult);
  } else {
    console.log("Pose not found");
    setTimeout(classifyPose, 100);
  }
}
```

CHAPTER VIII

CHAPTER VIII
TEST CASES

Testing is a crucial process in software development that involves evaluating a system or application to ensure it meets the desired requirements and functions as intended. It is an essential step in the software development life cycle (SDLC) that helps identify defects, errors, or issues in the software, allowing them to be addressed and resolved before the final product is released. Testing is necessary to ensure the reliability, functionality, and usability of the software, providing confidence to both the developers and end-users.

Testing is conducted to verify that the software meets the specified requirements. It ensures that all the desired features and functionalities are correctly implemented and perform as expected. By systematically testing the software against predefined test cases and scenarios, developers can identify any discrepancies or deviations from the expected behavior, allowing them to fix the issues and align the software with the intended requirements.

Testing helps to uncover defects or bugs in the software. These defects can range from syntax errors and logical flaws to issues related to performance, security, or compatibility. By thoroughly testing the software, developers can detect and rectify these defects, thereby improving the overall quality and reliability of the software. Testing also helps to identify any potential vulnerabilities or security loopholes, ensuring that the software is robust and secure. Testing is an iterative process that should be performed throughout the development lifecycle. It is not limited to a single phase but should be carried out at different stages of development.

8.1 Unit Testing

Unit testing is a type of software testing where individual units or components of a software application are tested in isolation from the rest of the system to ensure that they function correctly. In the context of MediFit, unit testing is performed by testing different modules individually, such as testing the user registration page, disease classification task, and parameters input page, among others.

The user registration page is tested to ensure that users can create an account successfully and that their information is stored correctly in the database. The testing also checks whether the user's input data is validated correctly, and error messages are displayed in case of incorrect inputs.

The disease classification task is tested by providing different input parameters to the machine learning model and checking whether the model correctly predicts the disease based on the input parameters. The testing also checks whether the model is generalizable to new data and performs accurately under various scenarios.

The parameters input page is tested to ensure that the user can input the required parameters correctly and that the parameters are validated appropriately before being sent to the machine learning model for disease classification. The testing also verifies that the user interface is easy to use and navigate and that error messages are displayed when necessary.

Other modules of MediFit are also tested using unit testing. For instance, the diet suggestions module is tested by providing different user data, such as age, weight, and height, and checking whether the system recommends the appropriate diet plan. Similarly, the yoga pose detection module is tested by providing different live yoga poses and verifying whether the system correctly detects the user's pose.

8.2 Integration Testing

Integration Testing is the process of testing the interaction between different modules of the system to ensure that they are working together correctly. In the case of MediFit, Integration Testing is done to verify that the different modules of the system are integrated correctly and the system works as a whole.

One of the most crucial aspects of MediFit is the interaction between the Disease Classification module and the Diet Suggestions module. The input parameters used for disease classification are also used for giving personalized diet suggestions to the user. Thus, Integration Testing is essential to ensure that the input parameters are being used correctly in both modules.

During Integration Testing, the focus is on the interactions between the modules and the data flow between them. For example, the integration between the User Input Parameters module and the Disease Classification module is tested to ensure that the user input parameters are being correctly processed and used for disease classification. Similarly, the integration between the Disease Classification module and the Diet Suggestions module is tested to ensure that the input parameters are being used correctly to give personalized diet suggestions.

Integration Testing in MediFit is performed after Unit Testing is completed successfully for each module. It ensures that the modules are integrated correctly and work seamlessly as a whole system. By performing Integration Testing, the MediFit team can ensure that the system functions as intended, and there are no errors or bugs due to the integration of different modules.

8.3 Usability Testing

Usability testing is a type of testing that is done to evaluate a system's ease of use, flexibility, and efficiency. It is used to identify the usability problems of a system and to provide recommendations for improving the system's usability. The goal of usability testing is to ensure that the system is intuitive and easy to use for the target audience.

In the case of MediFit, usability testing was performed to ensure that the web application is easy to use and understand for people with different levels of technical knowledge. The usability testing was done through surveys conducted with people of different age groups and technical knowledge to use our web application.

The surveys were conducted with a diverse group of people to gather feedback on the user interface, the ease of navigation, the clarity of instructions, the usefulness of the features, and the overall experience of using the application. The surveys were designed to identify any issues or problems that users might encounter when using the application.

In conclusion, the implementation of MediFit was greatly facilitated by the various testing procedures employed throughout the development process. The tests played a crucial role in ensuring the accuracy, reliability, and functionality of the web application, leading to a robust and user-friendly healthcare solution.

Unit testing, which involved testing individual modules and components of MediFit, allowed for the identification and resolution of any defects or issues at an early stage. By isolating and testing each module separately, developers were able to verify the correctness of their implementation and ensure that the individual components of MediFit were functioning as intended. This helped to maintain code quality, reduce bugs, and ensure the overall stability of the system.

Integration testing was also instrumental in the development of MediFit, as it involved testing the interaction and compatibility between different modules and components. By combining the individual modules and assessing their collective functionality, developers could identify any potential integration issues, such as data inconsistencies or communication problems. Integration testing helped to ensure the seamless interaction between different features of MediFit, guaranteeing the smooth flow of data and information throughout the system.

Usability testing was another vital aspect of implementing MediFit, as it focused on evaluating the application's user-friendliness and intuitiveness. By conducting surveys and gathering feedback from users of different age groups and technical backgrounds, the development team gained valuable insights into the user experience and identified areas for improvement. Usability testing allowed for the refinement of the user interface, navigation, and overall user experience, ensuring that MediFit was accessible and easy to use for its target audience.

In summary, the various testing procedures, including unit testing, integration testing, and usability testing, were integral to the successful implementation of MediFit. These tests helped to identify and rectify any issues or inconsistencies in the system, ensuring its accuracy, reliability, and user-friendliness. By incorporating rigorous testing throughout the development process, MediFit was able to provide a robust and effective healthcare solution that met the functional and non-functional requirements of its users.

CHAPTER IX

CHAPTER IX

RESULTS

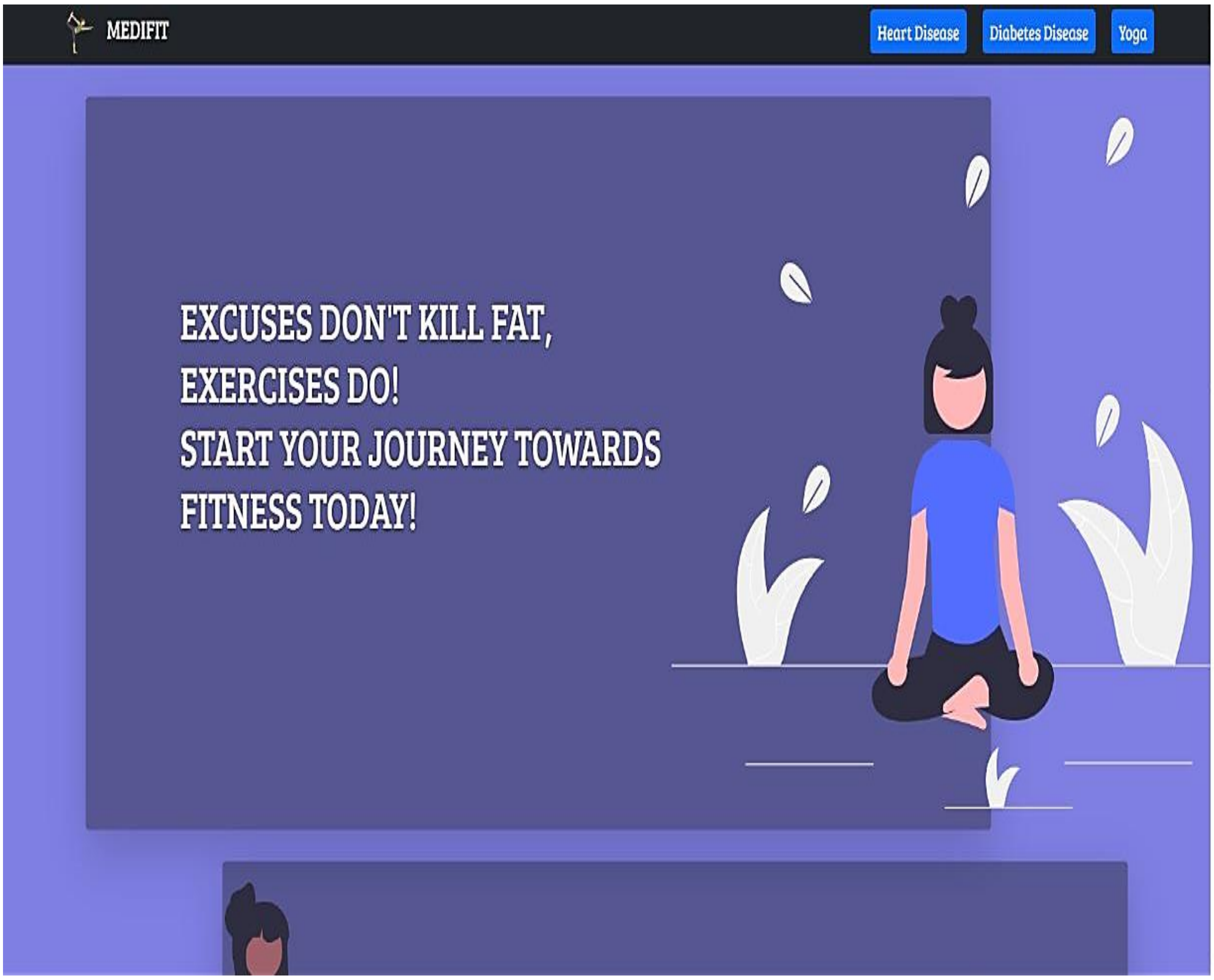

Fig 9.1 Home Page

The home page of MediFit serves as the entry point to the various functionalities provided by the web application. It contains links to the heart and diabetes input parameter pages. The home page is designed to be user-friendly, with an intuitive layout and navigation, enabling users to quickly and easily access the functionalities they need.

Fig 9.2 Heart Disease Input Parameters Page

The Heart disease parameters input page of MediFit is designed to be user-friendly, with text fields and drop-down options for easy input of details. The page includes essential parameters such as Name, Phone Number, Email, Gender, Old Peak, Max.Heart rate achieved, and type of chest pain, which are crucial in predicting heart diseases accurately. The user can quickly fill in their information and submit it for processing. With this page, we aim to make the user experience seamless and enable accurate detection of heart disease in individuals.

Fig 9.3 Diabetes Input Parameters Page

The Diabetes parameters input page of MediFit is designed to be user-friendly and intuitive, providing text fields and drop-down menus for entering the relevant information. It collects important information such as the user's name, phone number, email, gender, number of pregnancies, glucose levels, blood pressure, age, insulin levels, and other vital details. By gathering all these parameters, the system can classify the user's risk of diabetes

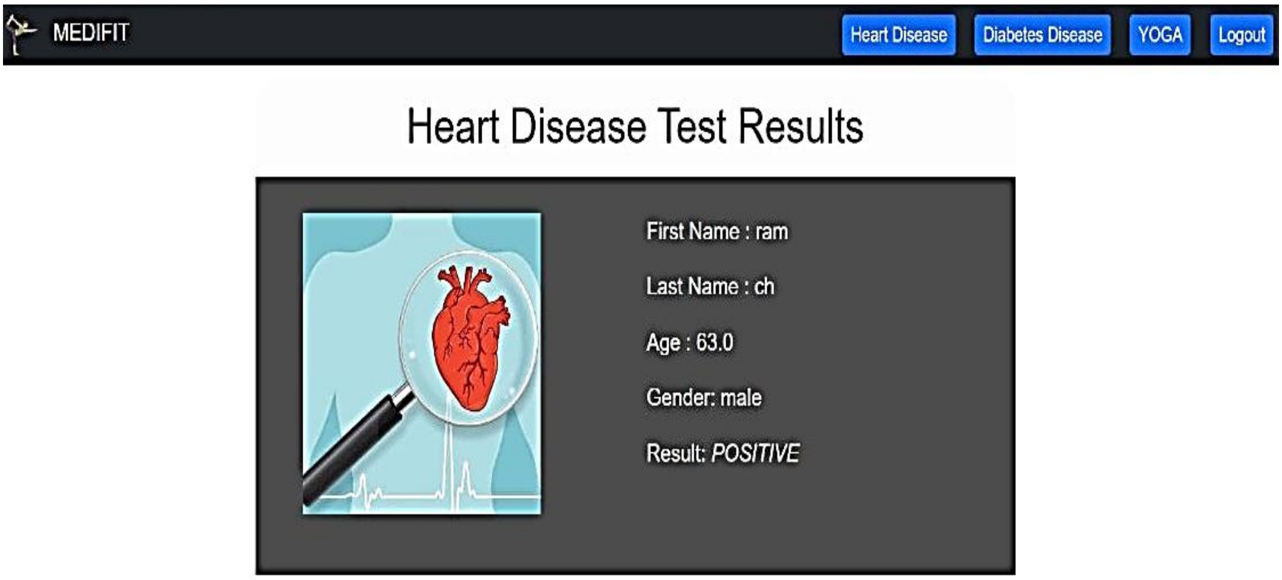

Fig 9.4 Heart Disease Test Results

The Heart disease test results page is an important component of the MediFit web application as it displays the results of the heart disease prediction test performed by the logistic regression machine learning model. After the user submits the input parameters required for the test, the results page appears with the first and last name of the user, their age and gender, and the result of the test. The result can be positive, indicating that the user is at risk of developing heart disease, or negative, indicating that the user is not at risk of developing heart disease based on the input parameters provided.

Fig 9.5 Diabetes Test Results

The Diabetes test results page in MediFit is designed to display the diagnosis of the user's diabetes status based on the input parameters provided. It displays the user's first and last name, age, gender, and the result of the test, which can either be positive or negative. The page is simple, easy to read, and provides the user with a convenient way to track their diabetes status.

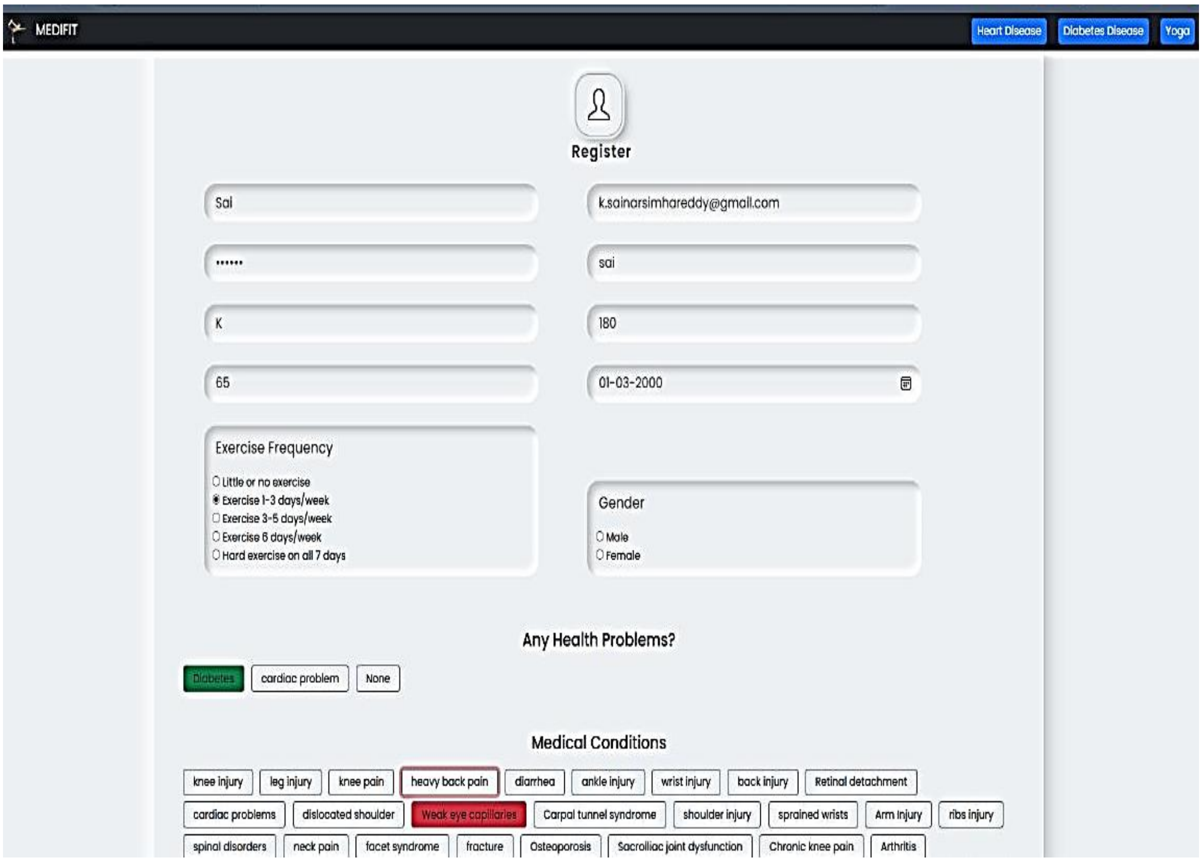

Fig 9.6 User Registration Page

The User registration page in MediFit is designed to collect basic user information for providing personalized services. It includes fields for entering the user's name, age, date of birth, and gender. Additionally, the page presents the user with a list of common health problems to choose from, which will be used to recommend diets. Once the user has filled in all the necessary information, they can click on the "Submit" button to complete the registration process.

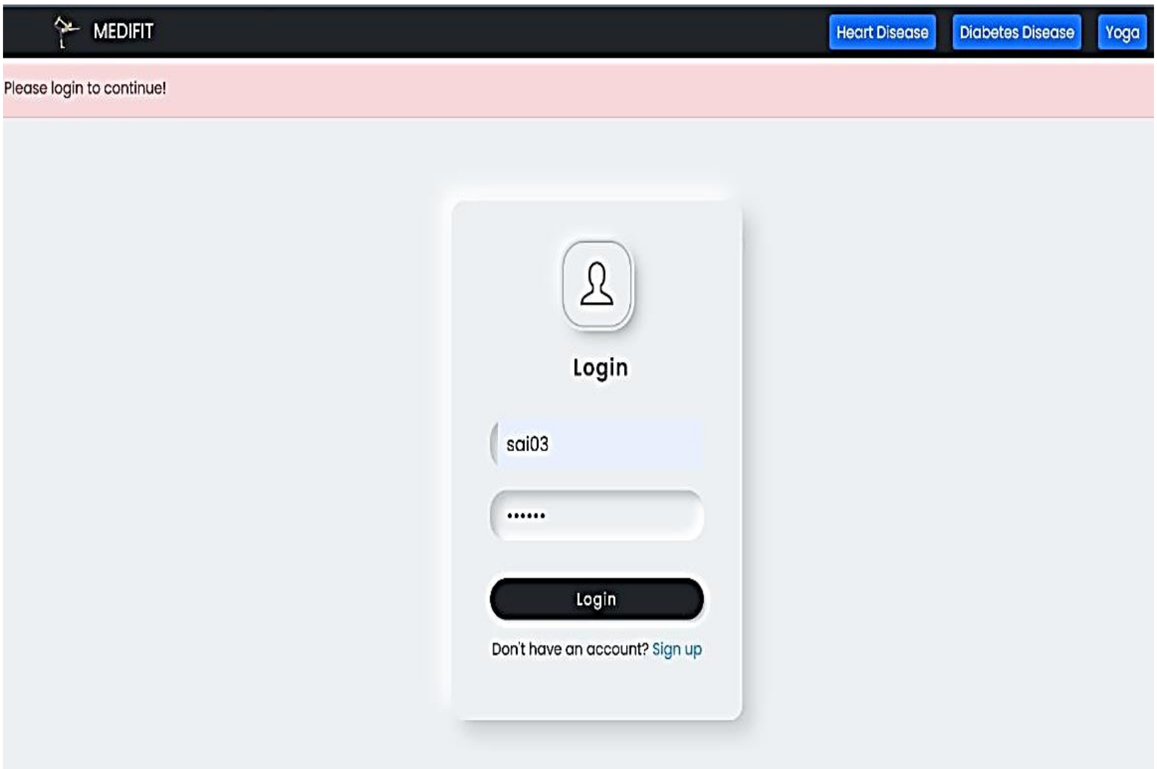

Fig 9.7 Login Page

The login page of the web application is the initial screen where a user is required to enter their login credentials such as username and password to access the main features of the platform. It serves as a gatekeeper for accessing personal information and functionalities that require a user to be authenticated.

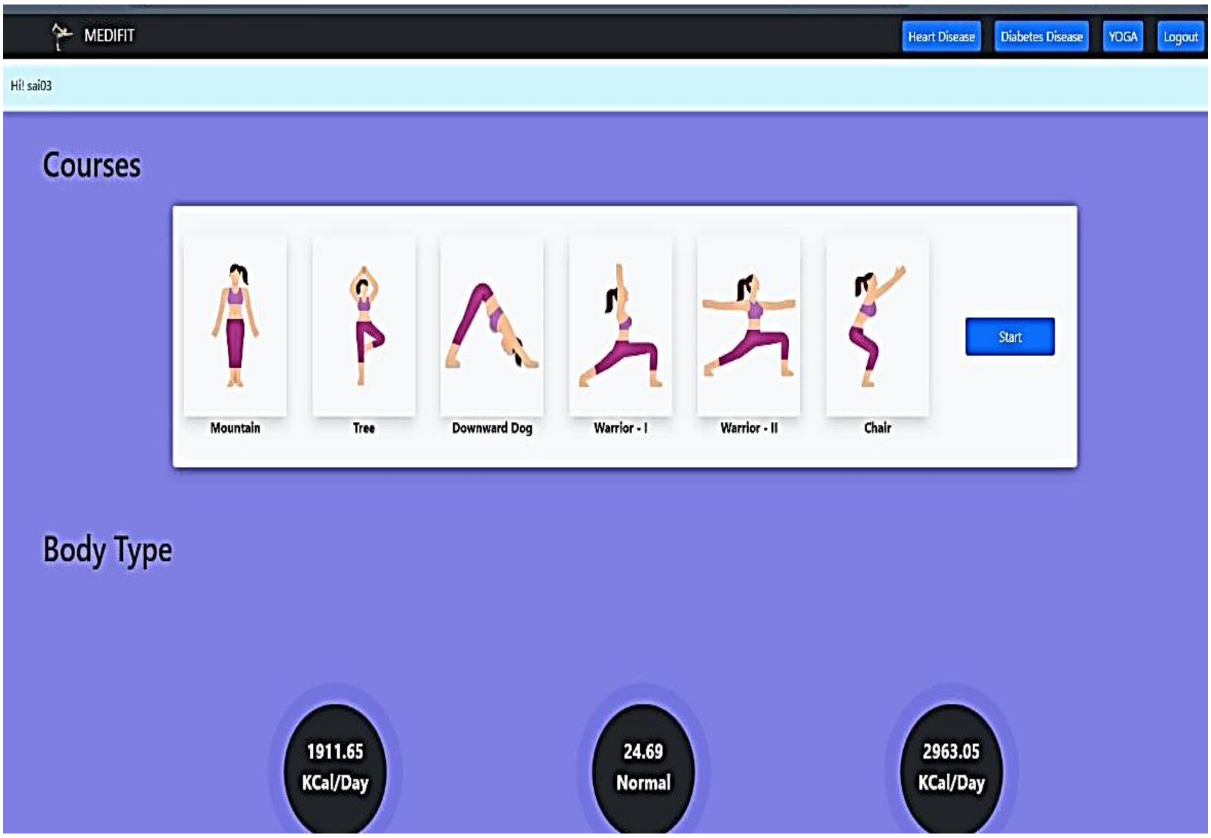

Fig 9.8 Yoga Course Page

The Yoga Course page in MediFit is designed to help users learn and practice yoga poses. The page presents a sequence of yoga poses that the user can follow. The user can click on the start button to begin the yoga session and then follow the instructions on the page to perform the poses.

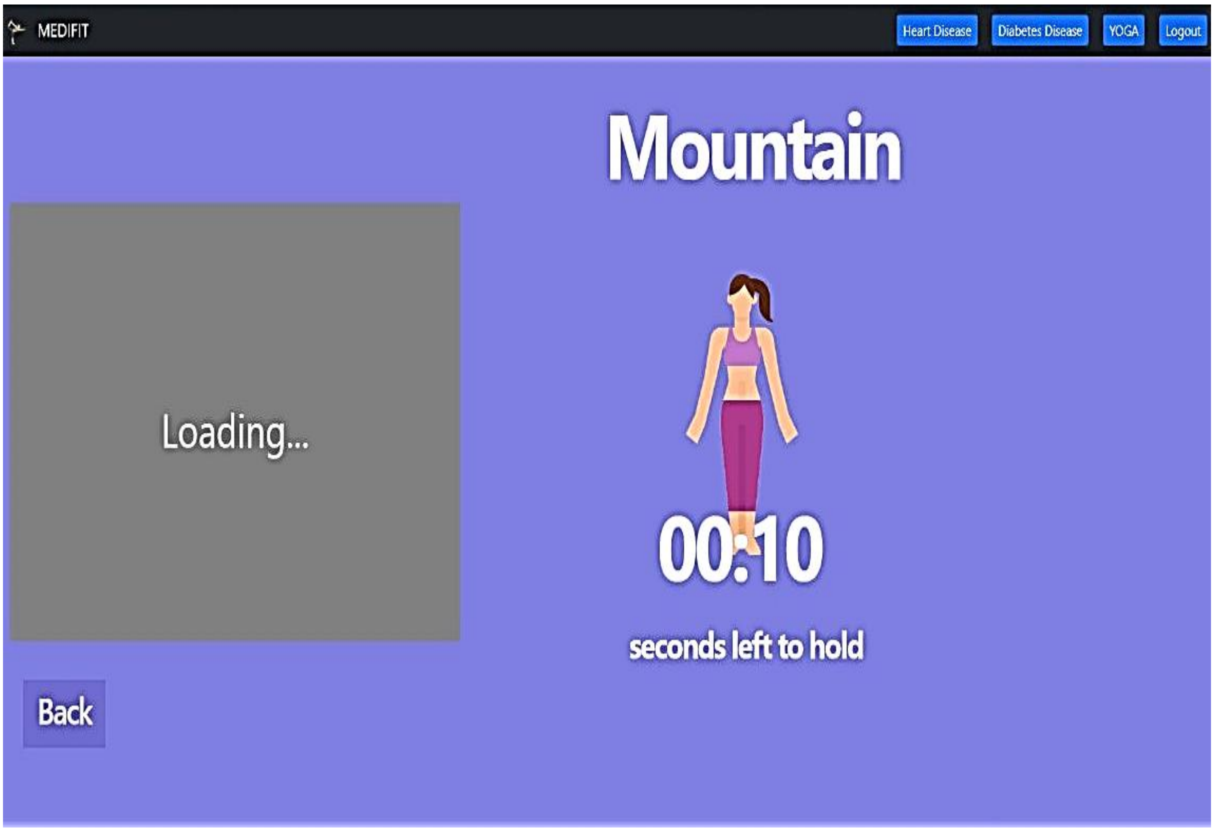

Fig 9.9 Yoga Pose Detection Page

The Yoga pose detection page in MediFit is a unique feature that leverages computer vision technology to detect and track the user's yoga pose in real-time using their device's camera. The user is provided with a yoga pose to be performed on the right side of the screen, and they are instructed to match the pose using their body movements. Once the user is in the correct position, a timer starts running, and the user must maintain the pose until the timer runs out. This page encourages users to practice yoga in a more interactive and engaging way while ensuring they perform the poses correctly.

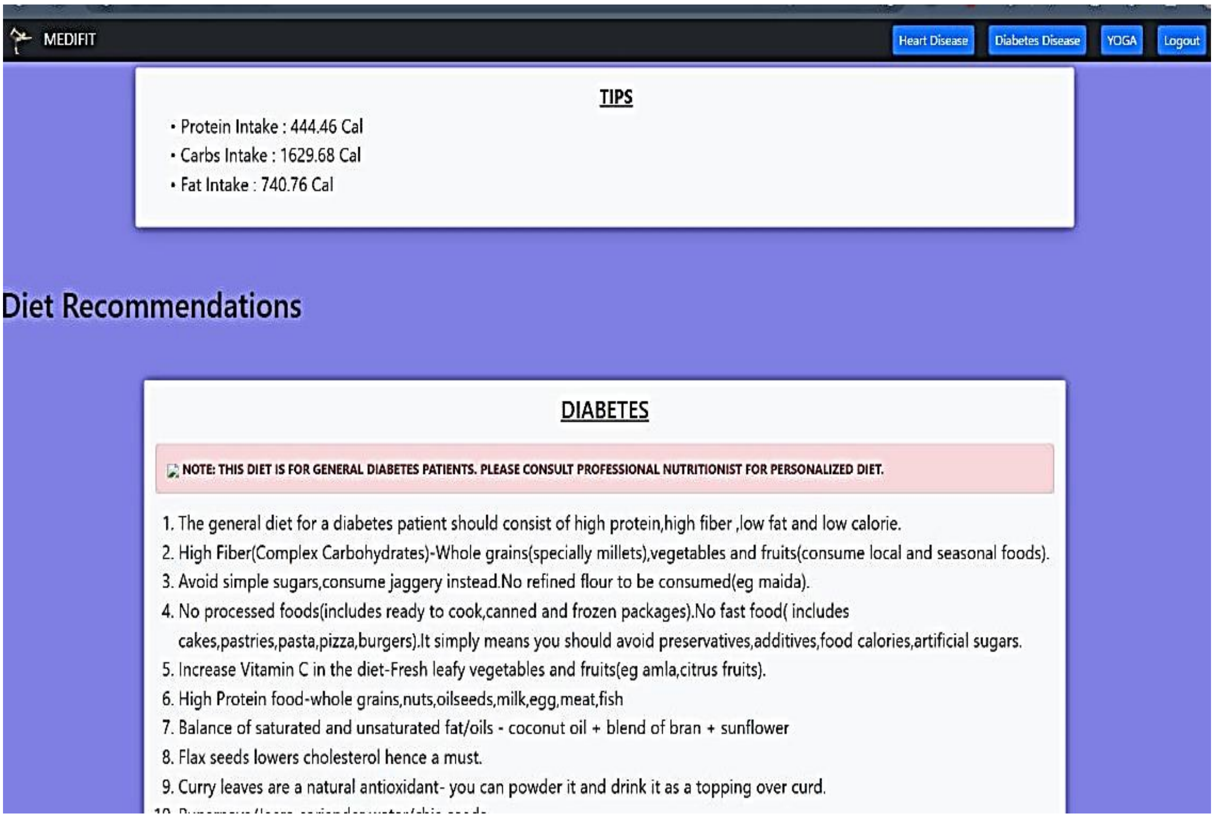

Fig 9.10 Diet Recommendations Page

The Diet recommendation page in MediFit is a very important page for users who are looking to improve their health through diet. It takes into account the user's age, gender, and health-related problems that were selected during registration to give personalized diet recommendations. The page displays the recommended amount of carbs, protein, and fat to be taken daily. Additionally, it provides diet suggestions and tips on what types of foods to consume and what to avoid based on the user's health condition.

The final results of MediFit have been highly promising, showcasing the effectiveness and value of the web application in promoting personal health and wellness. Through rigorous development, testing, and user feedback, MediFit has evolved into a comprehensive healthcare solution that provides accurate disease prediction, personalized diet recommendations, and live yoga pose detection.

In conclusion, the final results of MediFit demonstrate its effectiveness as a comprehensive healthcare web application. With accurate disease prediction, personalized diet recommendations, and live yoga pose detection, MediFit has proven to be a valuable tool in promoting personal health and wellness. The positive feedback and user experiences attest to the successful implementation and impact of the system, showcasing its potential to empower individuals in managing their health effectively.

CHAPTER X

CHAPTER X

CONCLUSION AND FUTURE WORK

10.1 Conclusion

MediFit is a comprehensive web application that offers a range of valuable features to support users in their health and wellness journey. By integrating disease prediction, personalized diet recommendations, and live yoga pose detection, MediFit provides a holistic approach to personal health management. The system's use of machine learning algorithms ensures accurate disease classification based on user input parameters, empowering individuals to take proactive measures for their well-being.

The non-functional requirements of MediFit, including usability, compatibility, speed, and reliability, have been meticulously addressed to ensure a seamless and user-friendly experience. The user interface is designed to be intuitive and accessible to individuals of varying technical expertise. Compatibility with different devices and operating systems allows users to access MediFit conveniently from their preferred platforms. The system's speed and reliability ensure quick and accurate processing of data, providing users with prompt results and recommendations.

MediFit has undergone rigorous testing to validate its performance and effectiveness. Unit testing has been conducted to evaluate the functionality of individual modules, ensuring that each component operates as intended. Integration testing has been carried out to assess the seamless integration of different modules and their interdependencies. Usability testing has played a crucial role in gathering user feedback and refining the system's interface, resulting in an intuitive and user-friendly platform that caters to individuals of all age groups and technical backgrounds.

In conclusion, MediFit is a unique and innovative solution for individuals seeking to improve their health and wellness. By combining disease prediction, personalized diet recommendations, and live yoga pose detection, MediFit offers a comprehensive and convenient platform to address various aspects of personal well-being. Whether it's early disease detection, tailored dietary guidance, or improving posture through yoga, MediFit empowers users to take proactive steps towards a healthier lifestyle. With its robust architecture, user-friendly interface, and accurate functionality, MediFit is poised to make a positive impact on the lives of individuals striving for improved health and wellness.

10.2 Future Work

In the future, MediFit aims to expand its disease prediction capabilities by adding more diseases to its repertoire. By incorporating additional machine learning algorithms and training models on diverse datasets, MediFit can enhance its ability to accurately predict a wider range of health conditions. This expansion will enable users to receive timely alerts and proactive recommendations for various diseases, enabling them to take necessary precautions and seek appropriate medical attention.

MediFit plans to incorporate personalized yoga recommendations as part of its future development. By leveraging advanced pose detection algorithms and integrating a comprehensive yoga database, the system will be able to provide tailored yoga routines based on the user's specific health conditions, flexibility levels, and goals. Personalized yoga recommendations can help users improve their physical fitness, enhance mental well-being, and address specific health concerns through targeted yoga poses and sequences.

Another important aspect of future development for MediFit is the integration of a feature that allows users to keep track of their diet and exercise history. By enabling users to log their daily meals, exercise activities, and relevant health metrics, such as weight and blood pressure, MediFit can provide a comprehensive overview of their health and progress. This feature will allow users to monitor their lifestyle choices, identify patterns, and make informed decisions about their diet and exercise routines based on the historical data.

MediFit aims to enhance user convenience and accessibility by incorporating input parameters that are more commonly available to users. This includes leveraging existing health monitoring devices, such as fitness trackers and smartwatches, to collect vital health data seamlessly. By integrating with popular wearable technologies and leveraging APIs, MediFit can retrieve and analyze real-time health data, enabling users to receive more accurate disease predictions and personalized recommendations based on their current health status.

Through these future developments, MediFit aims to provide a more comprehensive and personalized healthcare experience to its users. By expanding disease prediction capabilities, incorporating personalized yoga recommendations, enabling tracking of diet and exercise history, and leveraging commonly available input parameters, MediFit strives to continuously enhance its functionality and offer users an all-in-one platform for managing their health and well-being.

REFERENCES

1. https://ijngc.perpetualinnovation.net/index.php/ijngc/article/view/208
2. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9318204/
3. https://www.hindawi.com/journals/jhe/2021/9930985/
4. https://www.irjet.net/archives/V10/i1/IRJET-V10I171.pdf
5. https://arxiv.org/ftp/arxiv/papers/2209/2209.02492.pdf
6. https://heart.bmj.com/content/heartjnl/67/6/430.full.pdf
7. https://www.sciencedirect.com/science/article/pii/S1877050918308548
8. https://www.researchgate.net/publication/235728842_Mobile_Applications_for_Diabetes_Self-Management_Status_and_Potential
9. https://ieeexplore.ieee.org/document/9316119
10. https://ieeexplore.ieee.org/document/8303115
11. https://www.semanticscholar.org/paper/Disease-Prediction-using-Machine-Learning-Grampurohit-Sagarnal/54cd446c5eb6dc919780dc7cfd88d78132e54d73
12. https://blog.tensorflow.org/2018/05/real-time-human-pose-estimation-in.html
13. https://www.tensorflow.org/lite/examples/pose_estimation/overview
14. https://www.analyticsvidhya.com/blog/2021/09/posture-detection-using-posenet-with-real-time-deep-learning-project/
15. https://towardsdatascience.com/the-machine-learning-web-pose-and-actions-estimator-3203a0cf5f60
16. https://www.javatpoint.com/logistic-regression-in-machine-learning
17. https://www.geeksforgeeks.org/understanding-logistic-regression/
18. https://www.simplilearn.com/tutorials/machine-learning-tutorial/logistic-regression-in-python

19. https://www.kaggle.com/datasets/dileep070/heart-disease-prediction-using-logistic-regression

20. https://www.geeksforgeeks.org/ml-heart-disease-prediction-using-logistic-regression/

21. https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm

22. https://www.analyticsvidhya.com/blog/2021/10/support-vector-machinessvm-a-complete-guide-for-beginners/

23. https://www.analyticsvidhya.com/blog/2021/10/support-vector-machinessvm-a-complete-guide-for-beginners/

24. https://www.researchgate.net/publication/320395340_Classification_Of_Diabetes_Disease_Using_Support_Vector_Machine

25. https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset

26. https://www.kaggle.com/datasets/mathchi/diabetes-data-set

27. https://ai.stanford.edu/~nilsson/MLBOOK.pdf

28. https://www.cin.ufpe.br/~cavmj/Machine%20-%20Learning%20-%20Tom%20Mitchell.pdf

29. https://anjana2000-pradeep.medium.com/yoga-pose-estimation-using-posenet-vision-model-55a62977acef

LIST OF PUBLICATIONS

- International Journal of Scientific Research in Engineering and Management (IJSREM)S

MEDIFIT

Keesari Sai Narsimha Reddy[1], Vancha Sai Mukesh Reddy[2],Tirumala Venkata Giridhar[3], Ongole Srivalli Hrushitha [4], Mr. S. Ramchandra Reddy[5]

[1,2,3,4]*B.Tech. Student, Department of Computer Science and Engineering,*

sainarsimha.keesari200203@gmail.com, mukeshvancha122@gmail.com, venkatagiridhar555@gmail.com, srivallihrushithaongole99@gmail.com, rcreddy.cse@nmrec.edu.in

[5]*Assistant Professor, Department of Computer Science and Engineering,*

Nalla Malla Reddy Engineering College, Hyderabad, India

Abstract— This study uses machine learning to predict the likelihood of patients developing diabetes and heart disease. The system analyzes several input parameters, including blood pressure, insulin levels, age, and maximum heart rate achieved during exercise. Personalized nutrition recommendations and yoga courses are provided to manage the detected disease based on the patient's specific needs and preferences, medical history, and dietary restrictions. A patient dataset is collected, and machine learning algorithms are utilized to develop the disease prediction model. The model accurately forecasts the probability of a patient developing diabetes or heart disease. The study shows that yoga recommendations have therapeutic advantages for managing both illnesses, and custom-made nutritional suggestions are designed specifically for the identified disease. Incorporating machine learning, nutritional therapy, and yoga can improve patients' quality of life and promote long-lasting health and well-being. The study highlights the importance of early disease detection and personalized healthcare solutions for lifestyle-related diseases.

Keywords—Machine learning, diabetes, heart disease prediction, personalized nutrition, yoga therapy, healthcare, lifestyle-related diseases.

1. INTRODUCTION

The system analyzes various input parameters, including blood pressure, insulin levels, age, and maximum heart rate achieved during exercise, to create a disease prediction model. Additionally, personalized nutrition recommendation and yoga courses are provided to manage the detected disease, considering the patient's specific needs and preferences, medical history, and dietary restrictions.

The study also aims to explore the potential benefits of incorporating machine learning, nutritional therapy, and yoga for lifestyle-related disease management. The research collects patient data and uses machine learning algorithms to develop the prediction model. The study's findings show the model accurately forecasts disease likelihood and highlights the therapeutic advantages of yoga recommendations for managing both diabetes and heart disease. The custom-made nutritional suggestions are designed specifically for the identified disease, promoting long-term health and well-being.

Overall, this research provides a comprehensive and personalized health solution for the prevention and management of diabetes and heart disease. By incorporating machine learning, nutrition, and yoga therapy, this system aims to improve patients' quality of life and promote long-lasting health and wellness.

2.LITERATURE REVIEW

Heart disease and diabetes are two of the most common lifestyle-related illnesses that impact a large number of individuals worldwide. Various data mining techniques have been used to predict the onset of these diseases.

In a comparative study conducted by Randhawan et al. demonstrates the effectiveness of logistic regression algorithm in predicting heart disease. The study utilizes several important features such as age, gender, cholesterol levels, and blood pressure to develop a logistic regression model that predicts the presence of heart disease with high accuracy.

In another study by Farooqui and Ahmad (2020), support vector machines (SVM) and multilinear regression techniques were used to predict diabetes. The results showed that the SVM model performed better than the multilinear regression model, and the features selected for the SVM model played a significant role in improving the model's performance.

In the study conducted by Sundar et al. (2019), machine learning and neural networks were utilized to classify and detect yoga asanas. The study aims to create a system that can recognize yoga poses in real-time using video inputs. The researchers used the PoseNet model, which is a convolutional neural network, to detect human body key points in the video frames. The study found that the PoseNet model accurately detected human body key points and extracted features that were useful for classification. Overall, the study suggests that machine learning algorithms can be used to classify and detect yoga poses accurately, which can have numerous applications, including personalized yoga recommendations, feedback for practitioners, and monitoring of correct pose execution.

3. EXISTING METHODOLOGY

The current systems are primarily individual software applications that focus on specific areas such as disease prediction, diet recommendation, and yoga pose detection. These systems suffer from several limitations that make them less effective in addressing the overall health and wellbeing of individuals.

One significant limitation of the existing systems is their high cost, as they often require specialized hardware and software. Additionally, all the features are not available at one place, which can be inconvenient and time-consuming for users, and may result in a lack of integration between different applications.

Another limitation of the existing systems is limited accessibility, as many healthcare software applications require high-speed internet connections or specialized hardware. Moreover, the software may require specific technical skills to operate, which can limit accessibility to individuals who do not have the required expertise.

Finally, the existing systems are not tailor-made for individual users. Many healthcare software applications provide generalized recommendations that may not be suitable for everyone, resulting in suboptimal outcomes and even being harmful to the individual's health.

In conclusion, a comprehensive healthcare software application is required to address these limitations and provide a one-stop solution for disease prediction, diet recommendation, and yoga pose detection. The methodology for such an application could include incorporating machine learning algorithms and neural networks for accurate disease prediction and personalized diet recommendations. Additionally, computer vision techniques such as PoseNet could be utilized for accurate and real-time yoga pose detection. The application should be designed to be accessible and user-friendly for individuals with varying technical expertise and available resources.

4. PROPOSED METHODOLOGY

The proposed methodology for this project involves the use of support vector machines (SVM) and logistic regression for diabetes and heart disease prediction, respectively. For the yoga classification part, the mj5.js library, which includes the PoseNet model and neural networks, will be utilized. The proposed system will be represented as a web application that integrates all the features, providing a one-stop solution for users.

First, data related to heart disease and diabetes will be collected from various sources, including medical records and surveys. The data will then be preprocessed, which involves cleaning, feature selection, and normalization. The preprocessed data will be used to train the SVM and logistic regression models.

Next, for yoga pose detection and classification, the mj5.js library will be used to detect human body key points and extract features. The extracted features will be used to train neural networks for classification.

Finally, the heart disease, diabetes, and yoga classification models will be integrated into a web application. Users can input their data, and the system will provide personalized recommendations based on the input. The web application will also include features such as visualization of the results, feedback for practitioners, and monitoring of correct pose execution.

In summary, the proposed methodology involves the use of logistic regression and SVM for heart disease and diabetes prediction, respectively, and mj5.js with neural networks for yoga pose detection and classification. The proposed system will be represented as a web application that integrates all the features and provides personalized recommendations for users. This approach provides a comprehensive and accessible solution for individuals to monitor their health and wellbeing.

5. LIMITATIONS

1) The accuracy of the yoga classification models may be affected by factors such as clothing, lighting, and camera quality. This can result in inaccurate pose detection and classification, which may affect the recommendations provided to users.

2) The use of machine learning algorithms such as MJ5.js (PoseNet, Neural Network) for yoga classification also has limitations. While these algorithms can accurately detect and classify yoga poses, they require significant amounts of computational resources, including processing power and memory. This can be challenging to achieve in a web application, especially for users with limited hardware resources.

3) Another limitation is the availability of data. For accurate disease prediction, a large amount of data is required, including medical history, lifestyle habits, and biometric data. It may be challenging to obtain such data from individuals, especially those who are not regularly monitored by healthcare professionals.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 04 | April - 2023 Impact Factor: 8.176 ISSN: 2582-3930

6. SYSTEM ARCHITECTURE AND PERFORMANCE

.

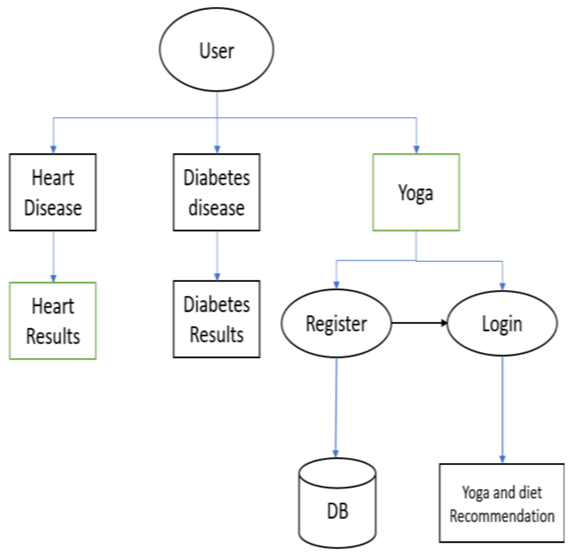

Fig1: System architecture

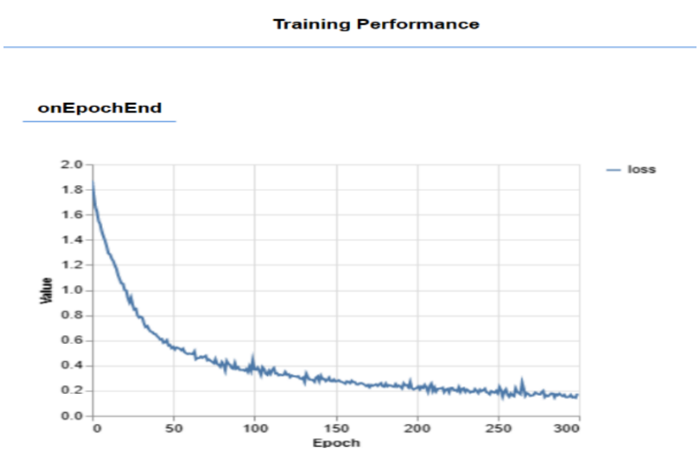

Fig2: Yoga Training performance

7. MODULES

1) Heart disease prediction module: This module will utilize logistic regression to predict the likelihood of an individual developing heart disease based on input data such as age, gender, blood pressure, cholesterol levels, and other relevant factors.

2) Diabetes prediction module: This module will use SVM to predict the likelihood of an individual developing diabetes based on input data such as age, gender, body mass index, blood sugar levels, and other relevant factors.

3) Yoga classification module: This module will utilize the PoseNet model, a convolutional neural network, to detect human body key points in video inputs and classify them into different yoga poses. This module can be used to provide personalized yoga recommendations, feedback for practitioners, and monitoring of correct pose execution.

Overall, these modules can be integrated into a web application that provides users with a comprehensive solution for disease prediction and yoga classification.

8. RESULTS AND CONCLUSION

The proposed system demonstrated the feasibility of using machine learning and neural networks for the prediction of heart disease and diabetes as well as the classification of yoga asanas. The system achieved high accuracy rates for heart disease and diabetes prediction using logistic regression and SVM techniques. The PoseNet model was effective in detecting human body key points and

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 04 | April - 2023 Impact Factor: 8.176 ISSN: 2582-3930

classifying yoga asanas accurately. The system was implemented as a web application, which provides easy access to users.

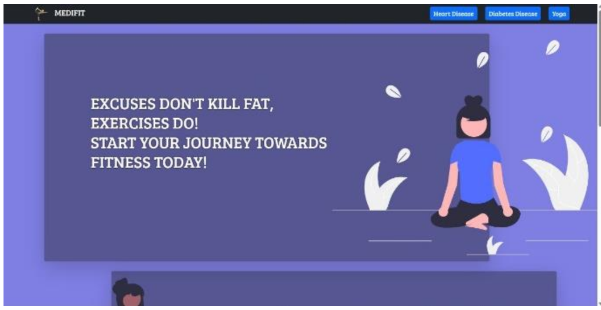

Fig3: Home page

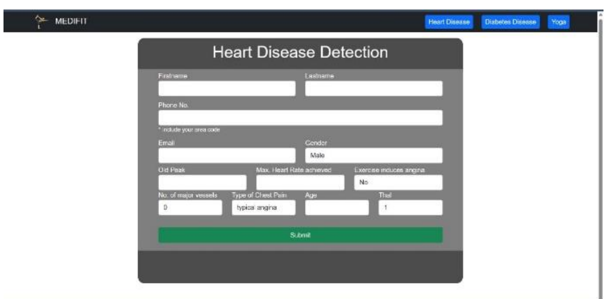

Fig 4 Diabetes Input Parameters Page

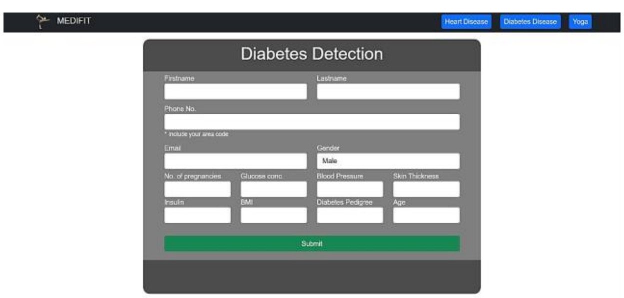

Fig 5 Diabetes Input Parameters Page

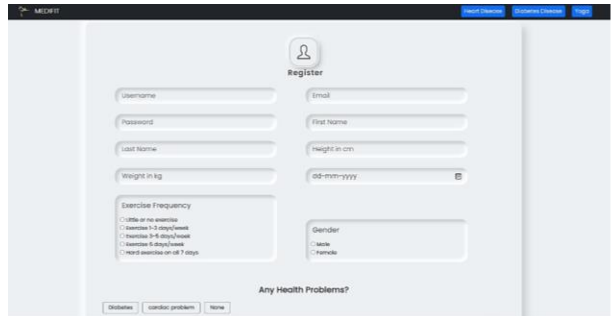

Fig 9.6 User Registration Page

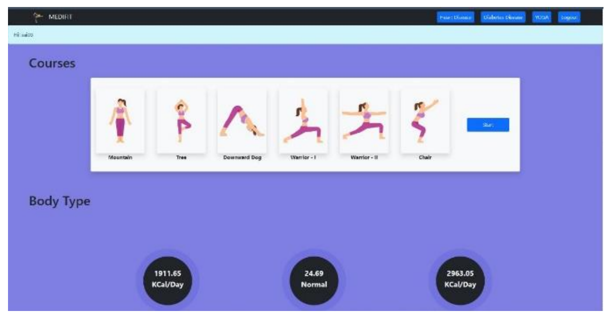

Fig 6 Yoga Course Page

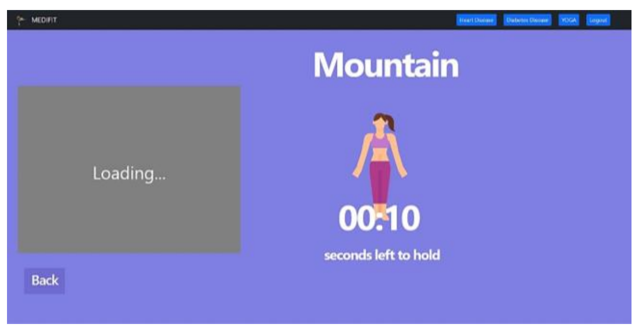

Fig7: Yoga Pose Detection

9. FUTURE SCOPE

1) Personalized recommendations: The system can be improved to provide more personalized recommendations for users, taking into account their specific medical conditions, dietary requirements, and physical abilities.

2) Expansion of yoga pose detection: The system can be expanded to detect more yoga poses, including advanced poses, which can benefit advanced yoga practitioners.

3) Integration with wearable devices: Integration with wearable devices such as smartwatches, fitness trackers, and other health monitoring devices can provide more accurate data for analysis, leading to better predictions and recommendations.

REFERENCES

[1]. Farooqui, Md. Ehtisham and Ahmad, Dr. Jameel, Disease Prediction System using Support Vector Machine and Multilinear Regression (August 13, 2020). International Journal of Innovative Research in Computer Science & Technology (IJIRCST)

[2] Tasnim, F., & Habiba, S. U. (2021). A Comparative Study on Heart Disease Prediction Using Data Mining Techniques and Feature Selection. 2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST).

[3] Yoga Asana Detection and Classification using Machine Learning and Neural Networks by Sundar V, Bhargav Reddy V, Shruti Srivatsan and Sree Vishal R

[4] Heart Disease Prediction Using Logistic Regression Algorithm Authors: Bhagyesh Randhawan, Ritesh Jagtap, Amruta Bhilawade, Durgesh Chaure

IJSREM CERTIFICATES

DOI: 10.55041/IJSREM20070

ISSN: 2582-3930

Impact Factor: 8.176

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING & MANAGEMENT

An Open Access Scholarly Journal || Index in major Databases & Metadata

CERTIFICATE OF PUBLICATION

International Journal of Scientific Research in Engineering & Management is hereby awarding this certificate to

Ongole Srivalli Hrushitha

in recognizaton to the publication of paper titled

MEDIFIT

published in IJSREM Journal on Volume 07 Issue 04 April, 2023

Editor-in-Chief
IJSREM Journal

www.ijsrem.com

ijsremjournal@gmail.com

DOI: 10.55041/IJSREM20070

ISSN: 2582-3930

Impact Factor: 8.176

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING & MANAGEMENT

An Open Access Scholarly Journal || Index in major Databases & Metadata

CERTIFICATE OF PUBLICATION

International Journal of Scientific Research in Engineering & Management is hereby awarding this certificate to

Vancha Sai Mukesh Reddy

in recognizaton to the publication of paper titled

MEDIFIT

published in IJSREM Journal on Volume 07 Issue 04 April, 2023

Editor-in-Chief
IJSREM Journal

www.ijsrem.com

ijsremjournal@gmail.com

DOI: 10.55041/IJSREM20070

ISSN: 2582-3930

Impact Factor: 8.176

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING & MANAGEMENT

An Open Access Scholarly Journal || Index in major Databases & Metadata

CERTIFICATE OF PUBLICATION

International Journal of Scientific Research in Engineering & Management is hereby awarding this certificate to

Keesari Sai Narsimha Reddy

in recognizaton to the publication of paper titled

MEDIFIT

published in IJSREM Journal on Volume 07 Issue 04 April, 2023

Editor-in-Chief
IJSREM Journal

www.ijsrem.com

ijsremjournal@gmail.com

DOI: 10.55041/IJSREM20070

ISSN: 2582-3930

Impact Factor: 8.176

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING & MANAGEMENT

An Open Access Scholarly Journal || Index in major Databases & Metadata

CERTIFICATE OF PUBLICATION

International Journal of Scientific Research in Engineering & Management is hereby awarding this certificate to

Tirumala Venkata Giridhar

in recognizaton to the publication of paper titled

MEDIFIT

published in IJSREM Journal on Volume 07 Issue 04 April, 2023

Editor-in-Chief
IJSREM Journal

www.ijsrem.com

ijsremjournal@gmail.com