A Project Report

On

MULTI DISEASE PREDICTION USING MACHINE LEARNING ALGORITHMS

Submitted in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

Y. Tulasi	**(20NN1A05C0)**
D. Shanmukhi	**(20NN1A05B0)**
M. Raveena Rai	**(20NN1A05A0)**
D. Sri Harshitha	**(20NN1A0572)**

Under the Esteemed Guidance of

Mrs. K. Chandrakala

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VIGNAN'S NIRULA INSTITUTE OF TECHNOLOGY AND SCIENCE FOR WOMEN

PEDAPALAKALURU, GUNTUR-522005

(Approved by AICTE, NEW DELHI and Affiliated to JNTUK, Kakinada)

2020-2024

VIGNAN'S NIRULA INSTITUE OF TECHNOLOGY AND SCIENCE FOR WOMEN

(Approved by AICTE, NEW DELHI and Affiliated to JNTUK, Kakinada)

PEDAPALAKALURU, GUNTUR-522005

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled **"MULTI DISEASE PREDICTION USING MACHINE LEARNING ALGORITHMS"**, is a bonafide work of **Y. Tulasi (20NN1A05C0), D. Shanmukhi (20NN1A05B0), M.Raveena Rai(20NN1A05A0) and D.Sri Harshitha(20NN1A0572)** submitted to the faculty of Computer Science And Engineering(AI &ML), in the partial fulfilment of the requirements for the award of degree of **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE AND ENGINEERING** from **VIGNAN'S NIRULA INSTITUTE OF TECHNOLOGY AND SCIENCE FOR WOMEN, GUNTUR.**

Project Guide	**Head of the Department**
Mrs. K. Chandrakala	**Dr. V Lakshman Narayana**
Assistant Professor	**Associate Professor**

EXTERNAL EXAMINE

DECLARATION

We hereby declare that the work described in this project work, entitled **"MULTI DISEASE PREDICTION USING MACHINE LEARNING ALGORITHMS"** which is submitted by us in partial fulfilment for the award of **Bachelor of Technology** in the Department of **Computer Science and Engineering** to the **Vignan's Nirula Institute of Technology and Science for women**, affiliated to Jawaharlal Nehru Technological University Kakinada, Andhra Pradesh, is the result of work done by us under the guidance of **Mrs. K. Chandrakala ,** Assistant Professor.

The work is original and has not been submitted for any Degree/Diploma of this or any other university.

Place:

Date:

Y. Tulasi	**(20NN1A05C0)**
D. Shanmukhi	**(20NN1A05B0)**
M. Raveena Rai	**(20NN1A05A0)**
D. Sri Harshitha	**(20NN1A0572)**

ACKNOWLEDGEMENT

We express our heartfelt gratitude to our beloved principal **Dr. P. Radhika** for giving a chance to study in our esteemed institution and providing us all the required resources.

We would like to thank **Dr. V Lakshman Narayana, Associate Professor, Head of the Department of Computer Science and Engineering**, for his extended and continuous support, valuable guidance and timely advices in the completion of this project thesis.

We wish to express our profound sense of sincere gratitude to our Project Guide **Mrs. K. Chandrakala, Assistant Professor, Department of Computer Science and Engineering (AI & ML)**, without whose help, guidance and motivation this project thesis could not have been completed the project successfully.

We also thank all the faculty of the Department of Computer Science and Engineering for their help and guidance of numerous occasions, which has given us the cogency to build- up adamant aspiration over the completion of our project thesis.

Finally, we thank one and all who directly or indirectly helped us to complete our project thesis successfully.

Y. Tulasi (20NN1A05C0)

D. Shanmukhi (20NN1A05B0)

M. Raveena Rai (20NN1A05A0)

D. Sri Harshitha (20NN1A0572)

MULTI DISEASE PREDICTION USING MACHINE LEARNING ALGORITHMS

ABSTRACT

Multi disease prediction is a comprehensive health prediction platform capable of forecasting multiple diseases using machine learning techniques. The platform integrates data from various sources including medical records, genetic information, and lifestyle factors to create predictive models for diseases such as diabetes, heart disease, kidney disease, breast cancer, liver disease. Through the utilization of advanced predictive analytics, the system provides personalized risk assessments and early detection alerts for individuals, empowering proactive healthcare management and preventive interventions. This represents a significant advancement in predictive health analytics, offering a holistic approach to disease prediction and prevention, ultimately contributing to improved healthcare outcomes and enhanced well-being for individuals worldwide. This uses a machine learning approach, utilizing labeled datasets containing patient health records. By employing supervised machine learning algorithms, the platform enables comprehensive disease prediction for a range of health conditions. Through meticulous data collection, preprocessing, and feature selection ensures the accuracy and reliability of its predictive models. Various algorithms such as logistic regression, random forest, support vector machines, decision tree and Gradient Descent are employed to build robust predictive models for each disease category. Feature selection and dimensionality reduction techniques are applied to enhance model performance and interpretability. This platform features a user-friendly interface accessible allowing individuals to input their health data and receive personalized disease risk assessments in real-time.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

In the realm of modern healthcare, the integration of machine learning algorithms has catalysed a paradigm shift in disease prediction and management. Early and accurate detection of these diseases is crucial for effective disease management [1]. Our project focuses on harnessing the power of supervised machine learning techniques to predict a spectrum of diseases, including Diabetes, Heart disease, Kidney disease, Breast cancer and Liver disease. There is no common system where one analysis can perform more than one disease prediction. The primary aim of our endeavour is to equip healthcare professionals with advanced predictive tools that enable early detection and proactive intervention, thereby improving patient outcomes and optimizing healthcare resource allocation. To implement multiple disease analysis used machine learning algorithms, tensorflow and Flask API. Python pickling is used to save the model behaviour and python unpickling is used to load the pickle file whenever required.

By leveraging large and diverse datasets encompassing various medical modalities such as electronic health records, medical imaging, and laboratory tests, we endeavour to develop robust predictive models capable of accurately identifying individuals at risk of developing specific diseases. Through rigorous experimentation and evaluation, we seek to elucidate the different machine learning algorithms in disease prediction, thereby paving the way for informed decision-making and personalized healthcare interventions [2].

Furthermore, to address key challenges in disease prediction, including data pre-processing, feature selection, model interpretability, and ethical considerations surrounding patient data privacy and consent [3]. As there are individual disease predictors there is no common system to perform multiple diseases. The significance lies in its potential to revolutionize healthcare delivery by enabling proactive and preventive measures, ultimately reducing the burden of disease and enhancing overall population health [4].

Moreover, each individual uses to detect based on the few parameters. Different disease predictions like diabetes prediction based on the few parameters it detects whether patient has diabetes or not [5]. And also, Breast Cancer detection used to detect based on the few parameters that whether a patient has breast cancer or not [6]. our project aims to address the pressing need for early detection and intervention in the context of chronic and acute diseases, which pose significant public health challenges worldwide. By leveraging the wealth of information contained within healthcare datasets, we strive to uncover hidden patterns, biomarkers, and risk factors that contribute to disease onset and progression [7]. The multifaceted nature of disease prediction necessitates a holistic approach that combines clinical expertise with cutting-edge computational methods. Our interdisciplinary team comprises experts in machine learning, bioinformatics, and clinical medicine, ensuring a comprehensive and nuanced understanding of the complexities involved.

Through our research, we seek to democratize access to advanced predictive analytics tools, making them accessible to healthcare providers across diverse settings and resource constraints. By democratizing access, we aim to bridge the gap between technology and healthcare delivery, ensuring that even underserved populations benefit from the latest advancements in predictive medicine [8]. Furthermore, our project emphasizes the importance of model interpretability and transparency in healthcare decision-making. We recognize that the adoption of predictive models hinges not only on their accuracy but also on their ability to provide actionable insights and facilitate meaningful dialogue between patients and healthcare providers, In addition to disease prediction, our project also explores the potential for predictive analytics to inform personalized treatment plans, medication management, and disease monitoring.

As we embark on this journey, we remain cognizant of the ethical considerations inherent in the use of healthcare data for predictive purposes. Privacy, confidentiality, and informed consent form the cornerstone of our approach, ensuring that patient rights and autonomy are upheld at every stage of the research process. In summary, our project represents a concerted effort to leverage the transformative potential of machine learning in disease prediction.

CHAPTER 2

LITERATURE SURVEY

CHAPTER 2

LITERATURE SURVEY

S Saraswat et al. [9] This project integrates machine learning techniques for smart disease prediction, featuring a user-friendly interface with three login options: user/patient, doctor, and admin. The platform's user-friendly interface allows individuals to input their symptoms easily, facilitating swift and accurate disease predictions. Furthermore, the inclusion of three distinct login options - user/patient, doctor, and admin - ensures tailored experiences for different stakeholders within the healthcare ecosystem. Through the implementation of the Naive Bayes Classifier, the system can effectively assess symptoms and generate precise disease likelihoods, thereby enabling early detection and intervention. Additionally, the incorporation of a feedback mechanism empowers users to contribute to the system's improvement by providing insights into the accuracy of predictions. This iterative learning process enhances the platform's predictive capabilities over time, making it more adept at identifying evolving disease patterns. Robust security measures, such as encryption protocols and access controls, safeguard sensitive medical data, fostering trust among users regarding the confidentiality of their health information.

J Pawar et al. [10] This project delves into the integration of machine learning techniques to develop a smart disease prediction system. Focusing on diseases like breast cancer and kidney ailments etc, we aim to harness the power of machine learning algorithms to accurately predict and analyse these health conditions. Amidst the challenges posed by the ongoing pandemic, our system provides a solution for early disease detection and personalized treatment recommendations. By leveraging machine learning algorithms, we strive to enhance healthcare accessibility and efficiency, enabling individuals to proactively manage their health from the comfort of their homes while maintaining safety protocols. Furthermore, the project may include provisions for continuous improvement and updates based on evolving medical research and user feedback, ensuring that the system remains at the forefront of disease prediction and healthcare innovation.

Dr. Nandini C et al. [11] This project aims to address the challenges posed by deadly and chronic diseases through early diagnosis, preventing complications associated with untreated conditions. By employing machine learning algorithms, our classifier system assists physicians in predicting and diagnosing diseases at an early stage. Our end-to-end data science project utilizes blood test report results to predict the likelihood of diseases such as Diabetes, Heart Disease, Kidney Disease, Liver Disease, and Breast Cancer. The system employs various machine learning algorithms to generate the most accurate predictions. Through a user-friendly web application, users can select diseases for consultation and input their blood test report details. Leveraging the most accurate model, the system prognosticates the likelihood of the selected disease in patients, facilitating early intervention and comprehensive healthcare management.

N. Shabaz Ali et al. [12] The primary focus of this paper is to explore the application of machine learning (ML) in healthcare through data mining techniques. By leveraging ML algorithms, data mining in the medical field aims to predict disease occurrences, offering insights crucial for research and education. The emergence of Smart Health Prediction Systems represents a significant advancement in medical science. Through ML and database management tools, new patterns can be extracted from large datasets, enhancing our understanding of disease patterns and improving predictive capabilities. This paper surveys the integration of data mining techniques with ML to predict diseases based on user symptoms, highlighting the potential of ML in revolutionizing smart healthcare systems.

A.N.V.K Swarupa et al. [13] This project integrates the development of a Smart Disease Prediction System using the Random Forest Algorithm, aiming to predict illnesses at an early stage due to lifestyle and environmental factors. By leveraging machine learning techniques, particularly Random Forest, the system enables accurate diagnosis based on symptoms, addressing the challenges of conventional diagnostic methods. The study utilized a dataset of 4920 patient records with 41 disorders, achieving a classification accuracy of 95% through experiments conducted with standard symptoms data. The system, implemented using machine learning and Python programming language with Tkinter interface, enhances healthcare by aiding in timely disease detection and resolution.

D Dahiwade et al. [14] This paper proposes a generalized disease prediction model based on patient symptoms, utilizing machine learning algorithms such as K-Nearest Neighbour (KNN) and Convolutional Neural Network (CNN). By incorporating factors like living habits and checkup information, the model enhances prediction accuracy. The CNN-based prediction achieves an accuracy of 84.5%, surpassing KNN. Additionally, the CNN model demonstrates lower time and memory requirements compared to KNN. Furthermore, the system provides insights into the risk associated with general diseases, aiding in effective risk management.

S.A.Siddiqui et al. [15] This project integrates the development of an IoT-based disease prediction system to address challenges faced by the healthcare sector, exacerbated by the COVID-19 pandemic. By leveraging the Internet of Things (IoT) and efficient machine learning algorithms, the system takes input from patients such as symptoms, audio recordings, and medical histories to predict diseases accurately. Additionally, it utilizes sensors like Arduino and ESP8266 to measure symptoms like fever and low blood oxygen. The system provides appropriate diagnosis and treatment based on its constantly updated database, potentially implemented as an application or website platform. This real-time capability is crucial, particularly in emergency situations or remote areas where access to healthcare facilities may be limited, thus potentially improving health outcomes and reducing healthcare disparities.

Dr. C K Gomathy et al. [16] This project integrates the development of Smart Disease Prediction using Machine Learning, a system designed to predict diseases based on symptoms provided by patients or users. The system processes user-provided symptoms as input and outputs the probability of the disease. Utilizing the Naïve Bayes classifier, a supervised machine learning algorithm, disease prediction is conducted, with the algorithm calculating the probability of the disease. With the exponential growth of biomedical and healthcare data, accurate analysis of medical data is crucial for early disease detection and patient care. With the use of linear regression and decision tree algorithms, diseases such as Diabetes, Malaria, Jaundice, Dengue, and Tuberculosis are predicted with enhanced accuracy and efficiency.

R Keniya et al. [17] This project explores the integration of machine learning techniques to develop a smart disease prediction system. The study Disease Prediction from Various symptoms using Machine Learning presents a system that employs multiple ML algorithms to forecast diseases based on symptoms, age, and gender. This project likely underscores the versatility of their smart disease prediction system, suggesting its applicability across diverse healthcare settings and populations. This versatility enhances its potential impact, contributing to more effective disease management strategies and improved public health outcomes. Utilizing a dataset comprising over 230 diseases, the system attained a remarkable accuracy of 93.5% through the weighted KNN algorithm. This methodology facilitates early disease detection, ensuring prompt intervention and potentially preserving lives.

CHAPTER 3

SYSTEM ANALYSIS

CHAPTER 3
SYSTEM ANALYSIS

SYSTEM ANALYSIS

System analysis for a Multi Disease Predictor involves a meticulous examination of requirements and operations, comprehending processes related to disease prediction, symptom input, and output generation. The intricacies such as data collection, model training, and result interpretation. The essential functionalities for accurate disease prediction, considering factors like data accuracy, model performance, and user interface design. Additionally, system operations, including data pre-processing, model selection, and result visualization. Through problem identification and solution proposition, we aim to enhance prediction accuracy and user experience, aligning solutions with stakeholder needs. At the conclusion, a comprehensive understanding of the Multi Disease Predictor and its requirements forms the basis for subsequent design and implementation phases, ensuring efficient disease prediction for users.

PROBLEM STATEMENT

The problem addressed by the Multi Disease Predictor revolves around the need for accurate and timely disease prediction, considering the rising healthcare challenges and the increasing demand for remote medical consultations. With the prevalence of various diseases and the limitations of traditional diagnostic methods, there is a critical need for a system that can effectively predict diseases based on user-provided symptoms and other relevant data. The aim is to develop a predictive model that utilizes machine learning algorithms to analyse symptoms and provide accurate disease predictions, thereby facilitating early detection, timely treatment, and improved healthcare outcomes. Additionally, the system must address issues related to data accuracy, model performance, and user interface design to ensure usability and reliability for healthcare professionals and patients alike. more information.

3.1 EXISTING SYSTEM

The existing system for disease prediction relies on traditional diagnostic methods, which involve manual examination, medical history review, and laboratory tests. Healthcare professionals assess patients' symptoms and medical history to make a diagnosis, a process that can be subjective and prone to errors. Accessibility to healthcare services may be limited, particularly in rural or underserved areas, leading to delays in diagnosis and treatment. Additionally, the existing system is time and resource-intensive, often requiring multiple visits to healthcare facilities and extensive medical tests. Furthermore, the lack of a Proactive approach to healthcare means that the system primarily focuses on treating disease after they have manifested clinically, rather than preventing them or detecting them early. Moreover, patient data fragmentation across different healthcare facilities and systems poses challenges in obtaining a comprehensive view of the patient's health status and medical history. These limitations highlight the need for a more advanced and integrated approach to disease prediction that leverages technology, data analytics, and machine learning to improves accuracy, accessibility, and efficiency in healthcare management. In addition to these, there is no common system where one analysis can perform more than one disease prediction.

3.1.1 LIMITATIONS OF EXISTING SYSTEM

- Subjectivity and variability in diagnostic decisions based on symptoms and medical history.
- Accessibility barriers, including geographical constraints and financial limitations, hindering timely access to diagnosis and treatment.
- Time and resource-intensive nature of traditional diagnostic methods, leading to delays and increased healthcare costs.
- Reactive approach focused on treating diseases after clinical manifestation rather than prevention or early detection.
- Fragmentation of patient data across different healthcare systems, complicating comprehensive health status assessment.

3.2 PROPOESED SYSTEM

The proposed system for the Multi Disease Predictor encompasses a comprehensive approach to disease prediction and management. At its core is the development of a robust machine learning-based prediction model capable of accurately analysing user-provided symptoms and other relevant data to predict diseases effectively. This entails seamless integration and management of diverse datasets, including symptom databases, medical records, and patient demographics, to enhance prediction accuracy. Moreover, scalability is a key consideration, ensuring the system can handle large volumes of data and user requests to accommodate future growth. Performance optimization is paramount, with a focus on maximizing prediction accuracy while minimizing processing time for timely and efficient disease prediction. A user-friendly interface will be developed to facilitate symptom input, viewing prediction results, and accessing additional resources or recommendations. Security measures will be implemented to safeguard sensitive patient data and ensure compliance with data privacy regulations. Continuous improvement mechanisms will be established to monitor prediction accuracy.

3.2.1 ADVANTAGES OF PROPOSED SYSTEM

- The use of robust machine learning algorithms ensures accurate analysis of symptoms and other relevant data, leading to precise disease predictions.
- By swiftly analyzing user-provided symptoms, the proposed model enables timely disease diagnosis, allowing for prompt medical intervention and treatment.
- The system is designed to handle large volumes of data and user requests, ensuring scalability to accommodate growing user bases and increasing data volumes without compromising performance.
- A user-friendly interface facilitates easy input of symptoms, viewing of prediction results, and access to additional resources or recommendations, enhancing user experience.
- Robust security measures safeguard sensitive patient data, ensuring compliance with data privacy regulations and maintaining confidentiality and integrity.
- By emphasizing disease prevention and early detection, the proposed model enables proactive healthcare management, reducing the burden of disease and improving patient outcomes.

3.3 DATASET SELECTION

- **DIABETES DATASET:** The dataset offers a comprehensive collection of attributes and health metrics pertinent to diabetes diagnosis. It includes information such as age, gender, body mass index (BMI), blood pressure, insulin levels, glucose levels, and diabetes pedigree function for each patient instance. With the target variable typically indicating the presence or absence of diabetes, researchers and data scientists commonly leverage this dataset for predictive modelling and classification tasks.

- **HEART DATASET:** The dataset offers a comprehensive collection of patient attributes and clinical measurements relevant to heart disease diagnosis. It encompasses factors such as age, gender, chest pain type, blood pressure, cholesterol levels, and electrocardiographic results, alongside a target variable indicating the presence or absence of heart disease.

- **CHRONIC KIDNEY DISEASE DATASET:** The dataset offers a comprehensive collection of attributes and clinical measurements related to chronic kidney disease (CKD). It includes a wide range of patient data, such as age, blood pressure, specific gravity, albumin, glucose levels, blood urea, serum creatinine, and various blood cell counts. Each instance in the dataset represents data from a single patient, with the target variable typically indicating the presence or absence of CKD.
- **LIVER DISEASE PATIENT DATASET:** The dataset available at the provided Kaggle link offers a comprehensive collection of attributes and clinical measurements related to liver disease patients. It includes a diverse range of patient data, such as age, gender, total bilirubin, direct bilirubin, liver enzyme levels (e.g., ALT and AST), total proteins, albumin, and albumin-globulin ratio (AGR).
- **BREAST CANCER DATASET:** The dataset offers a detailed array of attributes pertinent to breast cancer diagnosis, particularly focusing on characteristics of cell nuclei extracted from biopsy samples. These attributes encompass a variety of features such as radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, and fractal dimension.

3.4 FEASIBILITY STUDY

A feasibility study for a Smart Disease Predictor using Machine Learning (ML) involves assessing the technical, economic, and operational viability of the project.Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

TECHNICAL FEASIBILITY

- **Assess the availability of data:** Evaluate the availability, quality, and accessibility of datasets relevant to disease prediction. Consider factors like data volume, variety, velocity, and veracity.
- **ML model selection:** Investigate the feasibility of using ML techniques for disease prediction. Explore various algorithms and approaches suitable for the problem domain.

- **Infrastructure and technology:** Evaluate the technology stack required for model development, training, and deployment. Assess the computational resources, software tools, and frameworks needed for building the predictor.
- **Integration challenges:** Identify potential integration challenges with existing systems, databases, or APIs.

ECONOMIC FEASIBILITY

- **Cost-Benefit Analysis:** Estimate the costs associated with data acquisition, model development, infrastructure, deployment, and maintenance. Compare these costs against the potential benefits and expected returns on investment (ROI).
- **Return on Investment (ROI):** Assess the potential impact of the Smart Disease Predictor on healthcare outcomes, patient care, and resource utilization. Estimate the potential cost savings and improvements in diagnostic accuracy or early detection of diseases.

OPERATIONAL FEASIBILITY

- **User Acceptance:** Evaluate the acceptance and readiness of end-users, including healthcare professionals, patients, and other stakeholders, to adopt the Smart Disease Predictor.
- **Workflow Integration:** Assess the compatibility of the predictor with existing healthcare workflows and processes. Identify any changes or adaptations required for seamless integration.
- **Training and Support:** Determine the training needs of users and stakeholders to effectively use the predictor. Plan for ongoing support and maintenance to address technical issues and updates.

LEGAL AND ETHICAL CONSIDERATIONS

- **Data Privacy and Security:** Evaluate compliance with data protection regulations such as GDPR or HIPAA. Implement measures to ensure the privacy and security of patient data.

- **Ethical Implications:** Consider ethical concerns related to bias, fairness, and transparency in ML models, especially in healthcare applications. Ensure that the predictor's predictions are explainable and trustworthy.

PROJECT TIMELINE: A detailed project plan with defined milestones, deliverables, and timelines will ensure schedule feasibility. Agile methodologies, such as Scrum or Kanban, can facilitate iterative development, allowing for flexibility and adaptation to changing requirements.

RESOURCE AVAILABILITY: Sufficient availability of skilled developers, project managers, and other resources is essential for timely project execution. Proper resource allocation, team collaboration, and communication channels will mitigate schedule risks and delays.

CHAPTER 4

HARDWARE & SOFTWARE REQUIREMENTS

CHAPTER 4

HARDWARE & SOFTWARE REQUIREMENTS

4.1 PURPOSE, SCOPE

4.1.1 PROJECT SCOPE

The project scope for a Multi Disease Predictor with machine learning involves developing a system that utilizes machine learning algorithms to predict the likelihood of specific diseases based on input data, including patient demographics, medical history, symptoms, and diagnostic test results. Features include user registration and authentication, a user-friendly data input interface, disease prediction using machine learning models. The target audience encompasses the public seeking health assessments and healthcare professionals utilizing the system for patient diagnosis and treatment planning. Constraints such as data availability, regulatory compliance, resource availability and integration challenges must be considered. Project deliverables include a fully functional system, comprehensive documentation, and training materials for end-users and healthcare professionals. Defining the project scope ensures alignment of expectations and successful delivery of a Multi Disease Predictor system meeting user needs.

4.1.2 PROJECT PURPOSE

The purpose of developing a Multi Disease Predictor with machine learning is to provide individuals and healthcare professionals with a powerful tool for early detection, risk assessment, and personalized healthcare management. By leveraging machine learning algorithms, the system aims to analyse diverse sets of input data, including patient demographics, medical history, symptoms, and diagnostic tests, to generate accurate predictions regarding the likelihood of specific diseases. This predictive capability enables proactive intervention, timely treatment planning, and preventive measures, ultimately improving patient outcomes and reducing healthcare costs. Additionally, the Multi Disease Predictor serves as an educational resource, empowering users with knowledge about disease prevention, management strategies, and available treatment options. Overall, the system's purpose is to enhance healthcare decision-making, facilitate proactive health management, and empower individuals to lead healthier lives.

4.2 REQUIREMENT ANALYSIS

The project involves the requirement analysis serves as a foundation for the project, guiding its design and development phases to meet the defined objectives while addressing potential challenges and constraints. Requirements for ongoing system maintenance, including hardware and software updates, must be established to ensure the system remains accurate and secure over time.

- Feasibility Study
- Requirements Gathering
- Software Requirements Specification
- Software Requirements Validation

4.2.1 FUNCTIONAL REQUIREMENTS

These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed, and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements.

Examples of functional requirements:

1. Authentication of user whenever he/she logs into the system.
2. System shutdown in case of a cyber-attack.
3. A verification email is sent to the user whenever he/she registers for the first time on some software system.

4.2.2 NON-FUNCTIONAL REQUIREMENTS

These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to another. They are also called non-behavioral requirements.

4.3 SYSTEM REQUIREMENTS

By fulfilling these software and hardware requirements, developers can efficiently create, test and implement the Smart Disease Predictor

4.3.1 HARDWARE REQUIREMENTS

Hardware interfaces specifies the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements.

- **Processor** : Intel Core i3/i5 or equivalent
- **RAM** : 4GB or higher
- **Hard Disk** : 128GB

4.3.2 SOFTWARE REQUIREMENTS

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team's progress throughout the development activity.

- **Operating System** : Windows 10
- **IDE** : VS Code
- **Server-side Script** : python
- **Libraries** : Numpy, Pandas, Keras, Tensor Flow

4.4 SOFTWARE DESCRIPTION

4.4.1 PYTHON

Below are some facts about Python. Python is currently the most widely used multipurpose, high-level programming language. Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally, are smaller than other programming languages like Java. Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time. Python language is being used by almost all tech-giant companies like Google, Amazon, Facebook, Instagram, Dropbox, Uber etc. The biggest strength of Python is huge collection of standard libraries which can be used for the following:

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like OpenCV, Pillow)
- Web scraping
- Test frameworks
- Multimedia

4.4.2 ADVANTAGES OF PYTHON

4.4.2.1. EXTENSIVE LIBRARIES

Python downloads with an extensive library and it contain code for various purposes like regular expressions, documentation generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we don't have to write the complete code for that manually.

4.4.2.2. EXTENSIBLE

As we have seen earlier, Python can be extended to other languages. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

4.4.2.3. IMPROVED PRODUCTIVITY

The language's simplicity and extensive libraries render programmers more productive than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

4.4.2.4. SIMPLE AND EASY

When working with Java, you may have to create a class to print 'Hello World'. But in Python, just a print statement will do. It is also quite easy to learn, understand, and code. This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

4.4.2.5. READABLE

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and indentation is mandatory. These further aids the readability of the code.

4.4.2.6. OBJECT-ORIENTED

This language supports both the procedural and object- oriented programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the encapsulation of data and functions into one.

4.4.2.7. FREE AND OPEN-SOURCE

Like we said earlier, Python is freely available. But not only can you download Python for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

4.4.3 DISADVANTAGES OF PYTHON

So far, we've seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let's now see the downsides of choosing Python over another language.

4.4.3.1. SPEED LIMITATIONS

We have seen that Python code is executed line by line. But since Python is interpreted, it often results in slow execution. This, however, isn't a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Pythons are enough to distract us from its speed limitations.

4.4.3.2. WEAK IN MOBILE COMPUTING AND BROWSERS

While it serves as an excellent server-side language, Python is much rarely seen on the clientside. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called Carbonnelle. The reason it is not so famous despite the existence of Brython is that it isn't that secure.

4.4.3.3. DESIGN RESTRICTIONS

As you know, Python is dynamically-typed. This means that you don’t need to declare the type of variable while writing the code. It uses duck- typing. But wait, what’s that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can raise run-time errors.

4.4.4 MODULES USED IN PROJECT

NUMPY

NumPy is a general-purpose array-processing package. It provides a high- performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python.

PANDAS

Pandas is a Python library widely used for data manipulation and analysis. It provides data structures and functions for efficiently handling structured data, such as tabular data, time series, and heterogeneous data.

SCIKIT-LEARN

Scikit-learn, also known as sklearn, is a popular open-source machine learning library for Python. It provides simple and efficient tools for data analysis and machine learning tasks, making it an essential resource for both beginners and experienced developers.

TENSORFLOW

TensorFlow is an open-source machine learning framework developed by Google. It provides a comprehensive ecosystem of tools, libraries, and resources for building and deploying machine learning models. TensorFlow is designed to be flexible, scalable, and efficient, making it suitable for a wide range of applications, from research to production.

KERAS

Keras is an open-source deep learning library written in Python. It provides a high-level interface for building and training neural networks, making it easy to prototype and experiment with deep learning models.

SEABORN

Seaborn is a Python data visualization library based on Matplotlib that provides a high-level interface for creating attractive and informative statistical graphics.

MATPLOTLIB

Matplotlib is a comprehensive Python library for creating static, interactive, and animated visualizations. It offers a wide range of plotting functions and customization options, making it a versatile tool for data visualization.

4.4.5 ALGORITHMS USED

- **RANDOM FOREST:** Integrating Random Forest into the backend technology stack for a Smart Disease Predictor involves training the model on historical healthcare data using Python libraries like scikit-learn. Pre-processing steps are implemented to clean and prepare the data, including handling missing values, encoding categorical variables, and scaling numerical features.
- **K-NEAREST NEIGHBORS (KNN):** K-Nearest Neighbours (KNN) is a simple and intuitive machine learning algorithm used for classification and regression tasks. In KNN, the prediction for a new data point is made based on the majority class (for classification) or the average value (for regression) of its k nearest neighbours in the training dataset.
- **SUPPORT VECTOR MACHINE (SVM):** Support Vector Machine (SVM) is a powerful supervised machine learning algorithm used for classification and regression tasks. SVM is particularly well-suited for classification tasks in high-dimensional spaces and for datasets with complex decision boundaries.
- **GRADIENT BOOSTING CLASSIFIER:** Gradient Boosting Classifier is a popular ensemble machine learning algorithm that combines the predictions of multiple individual models, typically decision trees, to create a more powerful and accurate predictive model. It belongs to the family of boosting algorithms, which iteratively improves the performance of weak learners by focusing on the errors made by previous models.
- **XGBOOST:** XGBoost (Extreme Gradient Boosting) is an optimized and scalable implementation of the gradient boosting algorithm, designed to deliver high performance and accuracy in machine learning tasks. It builds upon the principles of gradient boosting and enhances them with several key features and optimizations.

CHAPTER 5

SYSTEM DESIGN

CHAPTER 5

SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE

The proposed system collects structured and unstructured data obtained from various sources. After data collection, they are subjected to preprocessing and are split into cleaning and test data sets. Then the training dataset is trained with the machine learning algorithms such as Random forest, Decision Tree,XGBoost KNN to a number of epochs for improving the accuracy of the prediction results.

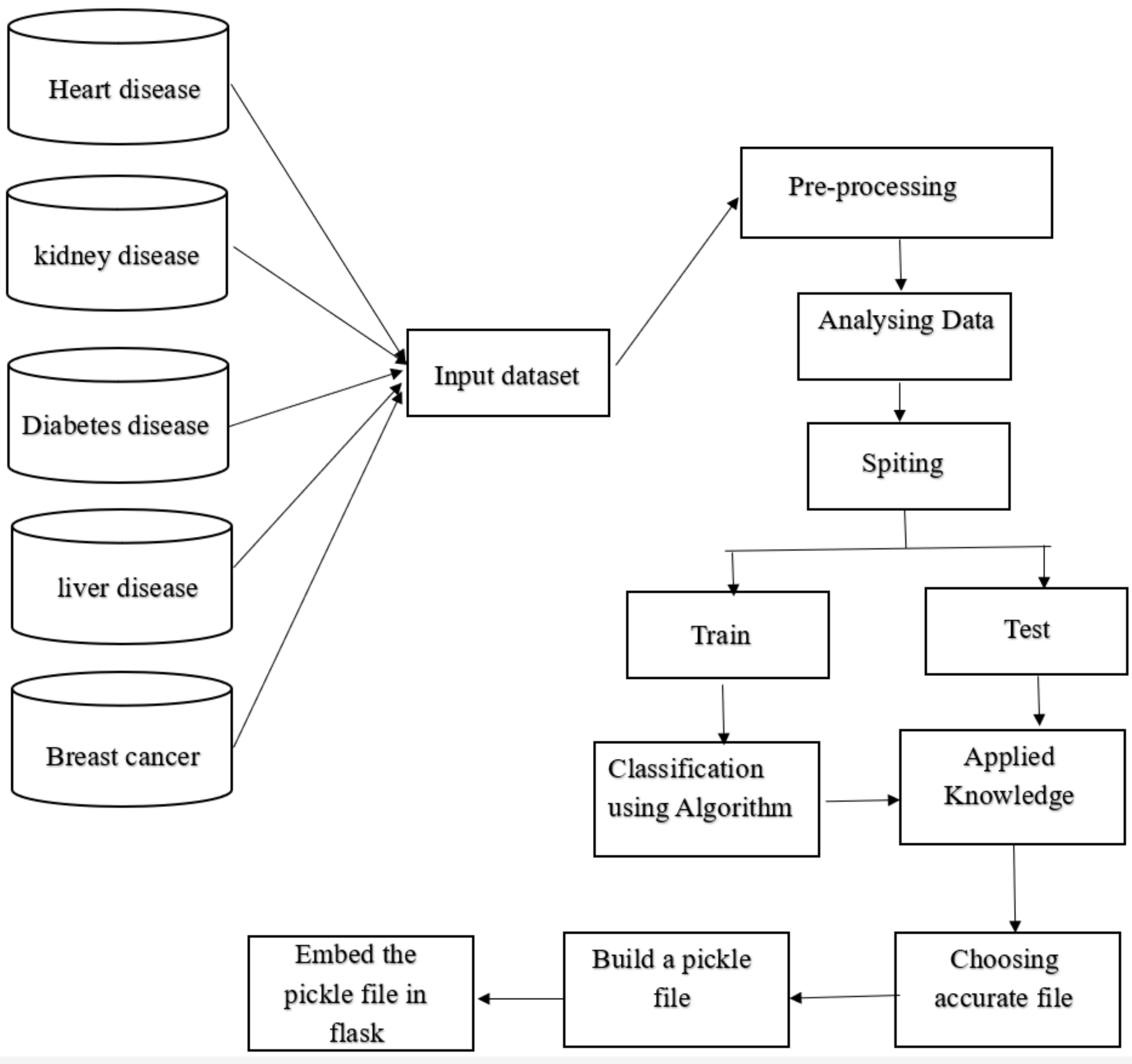


Fig 5.1: System Architecture

5.2 DESIGN OVERVIEW

UML integrates the most effective practices from data modelling, business modelling, object modelling, and component modelling, making it applicable across various stages of the software development life cycle and across different implementation technologies. By combining elements from the Brooch method, Object Modelling Technique (OMT), and Object-Oriented Software Engineering (OOSE), UML provides a unified modelling language that is widely used and understood. Its goal is to serve as a standard modelling language capable of representing concurrent and distributed systems seamlessly.

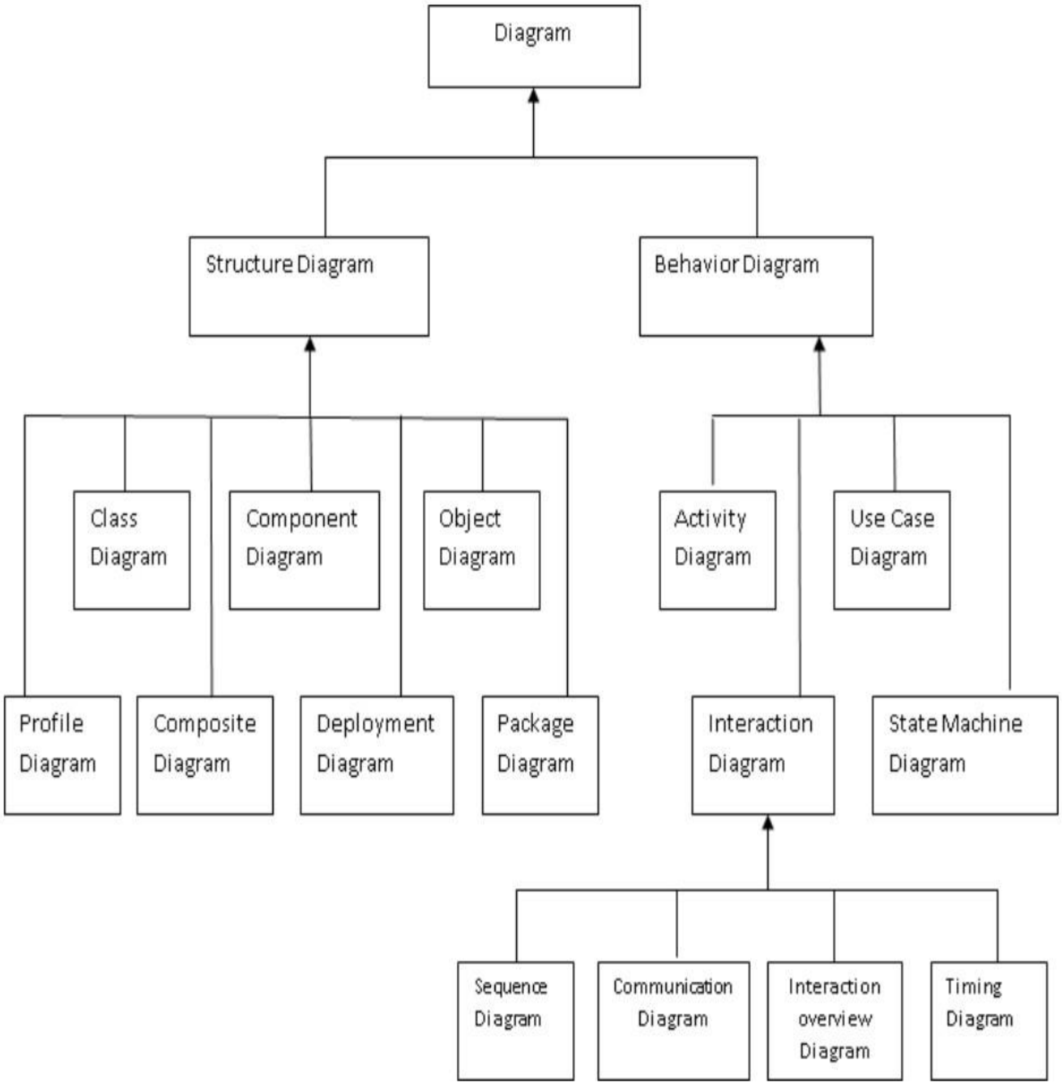


Fig 5.2: Types and categories of UML diagrams

5.2.1 UML DIAGRAMS

The Unified Modelling Language (UML) serves as a comprehensive tool for specifying, visualizing, modifying, constructing, and documenting the artifacts of object-oriented software-intensive systems during development. It provides a standardized approach to depict a system's architectural blueprints, encompassing elements such as actors, business processes, logical components, activities, programming language statements, database schemas, and reusable software components.

UML empowers software engineers to articulate an analysis model using a modelling notation governed by a set of syntactic, semantic, and pragmatic rules. A UML system is represented through five distinct views, each describing the system from unique perspectives and defined by a set of diagrams:

USER MODEL VIEW

- ❖ Portrays the system from the user's standpoint.
- ❖ Captures usage scenarios and solutions as understood by client stakeholders.

STRUCTURAL MODEL VIEW

- ❖ Focuses on internal system structures and functionalities.
- ❖ Depicts static structures within the system.

BEHAVIOURAL MODEL VIEW

- ❖ Represents dynamic behaviours of system components.
- ❖ Illustrates interactions among various elements described in the user and structural model views.

IMPLEMENTATION MODEL VIEW

- ❖ Implementation Model View is also known as the Architectural view
- ❖ Outlines subsystem enumeration, component organization, and dependencies within the implementation model.

ENVIRONMENTAL MODEL VIEW

- ❖ Describing both structural and behavioural aspects of the domain or environment
- ❖ Often referred to as the deployment or physical view.

5.2.1.1 USE CASE DIAGRAM

In the context of the Multiple Disease Prediction System, two key actors are involved: the User and the System. The User, typically a healthcare professional, interacts directly with the system, serving as the primary input provider and recipient of predictions. Healthcare professionals, including doctors, nurses, and medical practitioners, input comprehensive patient data into the system, encompassing demographic information, medical history, and diagnostic test results. Subsequently, they rely on the system-generated predictions for various diseases, empowering them to make well-informed decisions regarding patient care, treatment plans, and preventive measures.

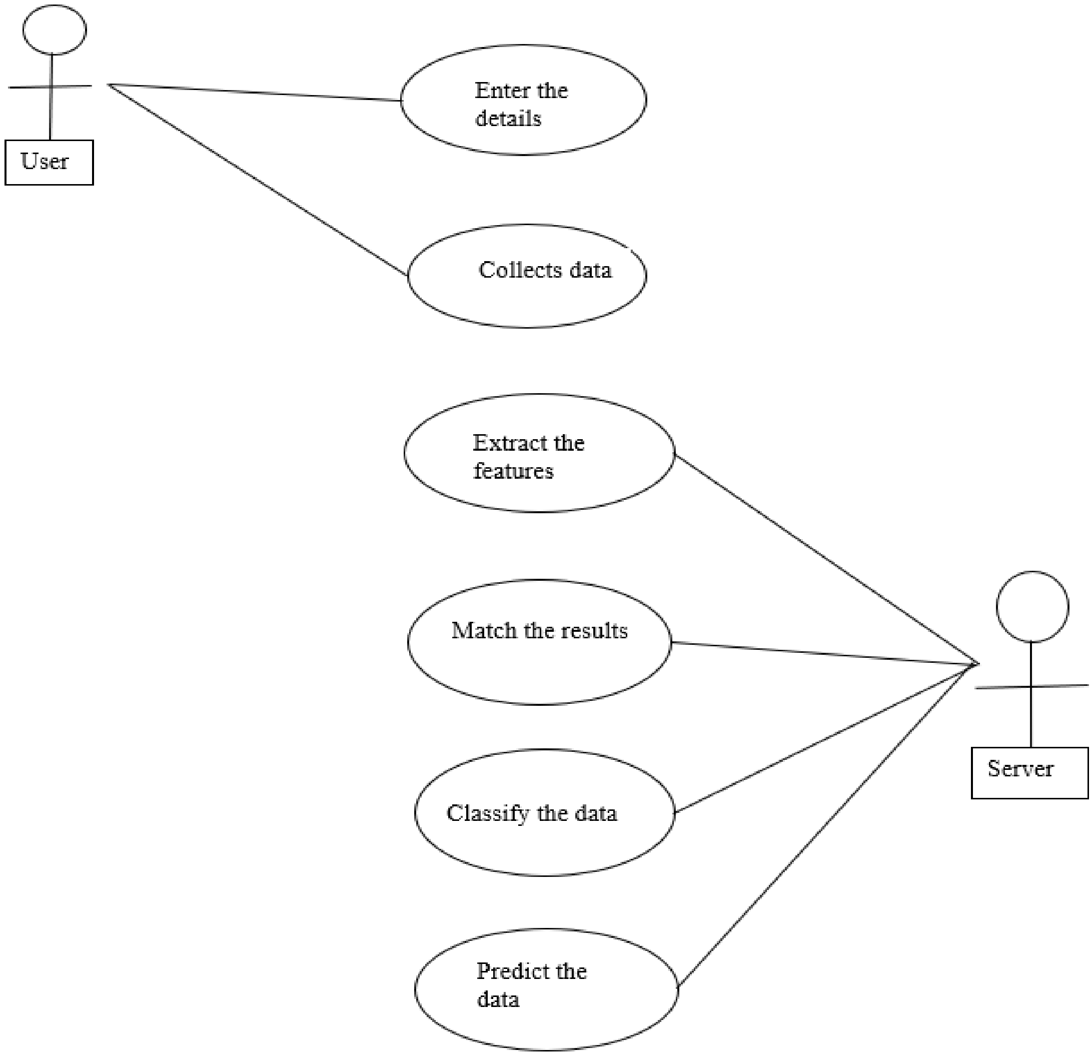


Fig 5.2.1.1: Use Case Diagram

5.2.1.2 ACTIVITY DIAGRAM

The activity diagram for the Multiple Disease Prediction System aimed at assisting healthcare professionals in making informed decisions regarding patient care. The process begins with the healthcare professional initiating the system and inputting comprehensive patient data, encompassing demographic details, medical history, and diagnostic test results. Subsequently, the system diligently processes this data using specialized machine learning and deep learning algorithms tailored for each disease prediction task. Through this activity, the system performs data preprocessing, feature extraction, and model inference to generate predictions for multiple diseases, including diabetes, heart disease, kidney disease, breast cancer, liver disease.

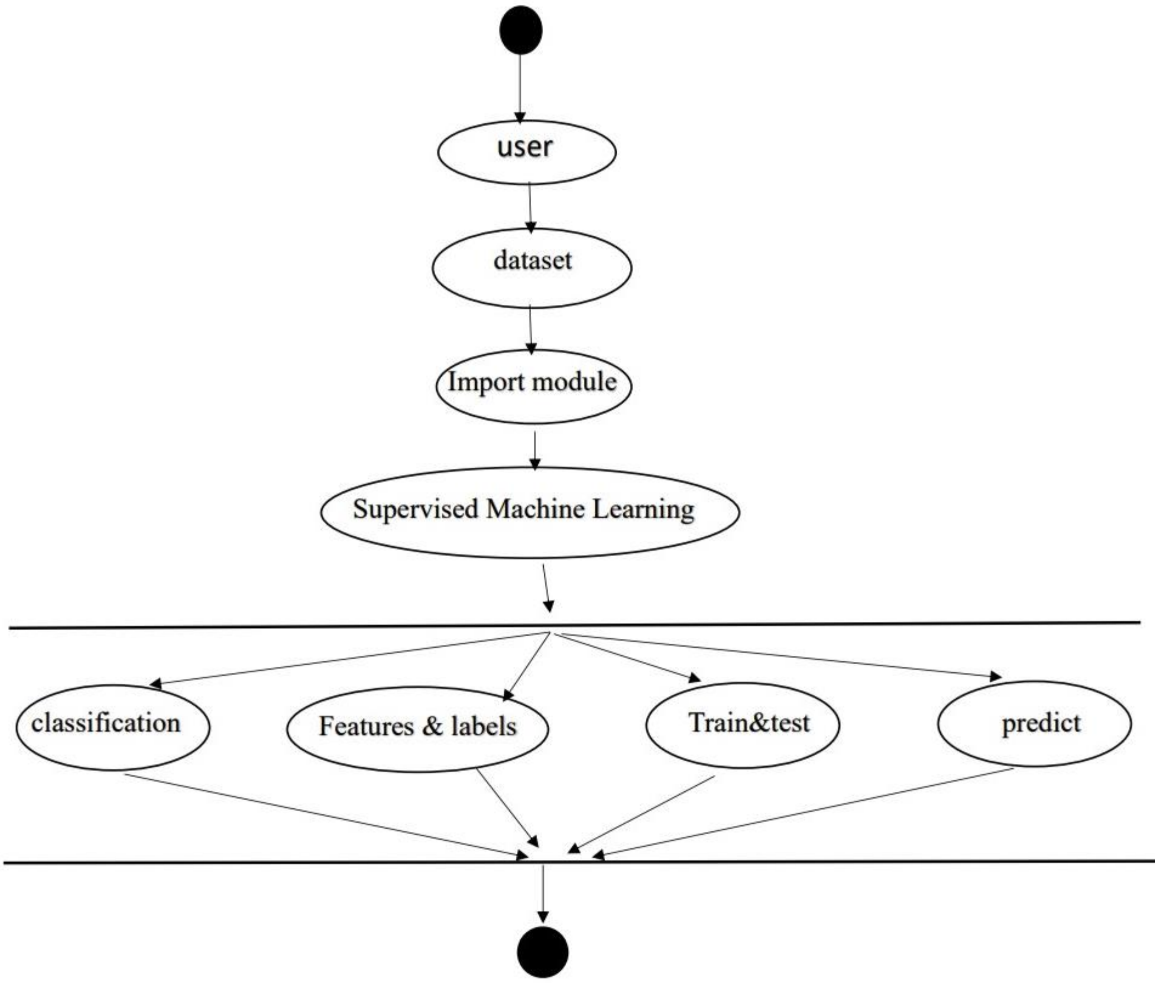


Fig 5.2.1.2: Activity Diagram

5.2.1.3 CLASS DIAGRAM

The class diagram for the Multiple Disease Prediction System encapsulates the various components and their relationships, offering a structural blueprint for the system's functionality. At its core lies the Disease Prediction System class, which orchestrates the processing of patient data and the generation of disease predictions. This class interacts with Disease Prediction Model instances, representing the machine learning models tailored for each disease prediction task. Healthcare professionals, represented by the Healthcare Professional class, utilize the system through the User Interface, inputting patient data and accessing prediction results. Patient data, stored in the Patient class, includes demographic information, medical history, and diagnostic test results.

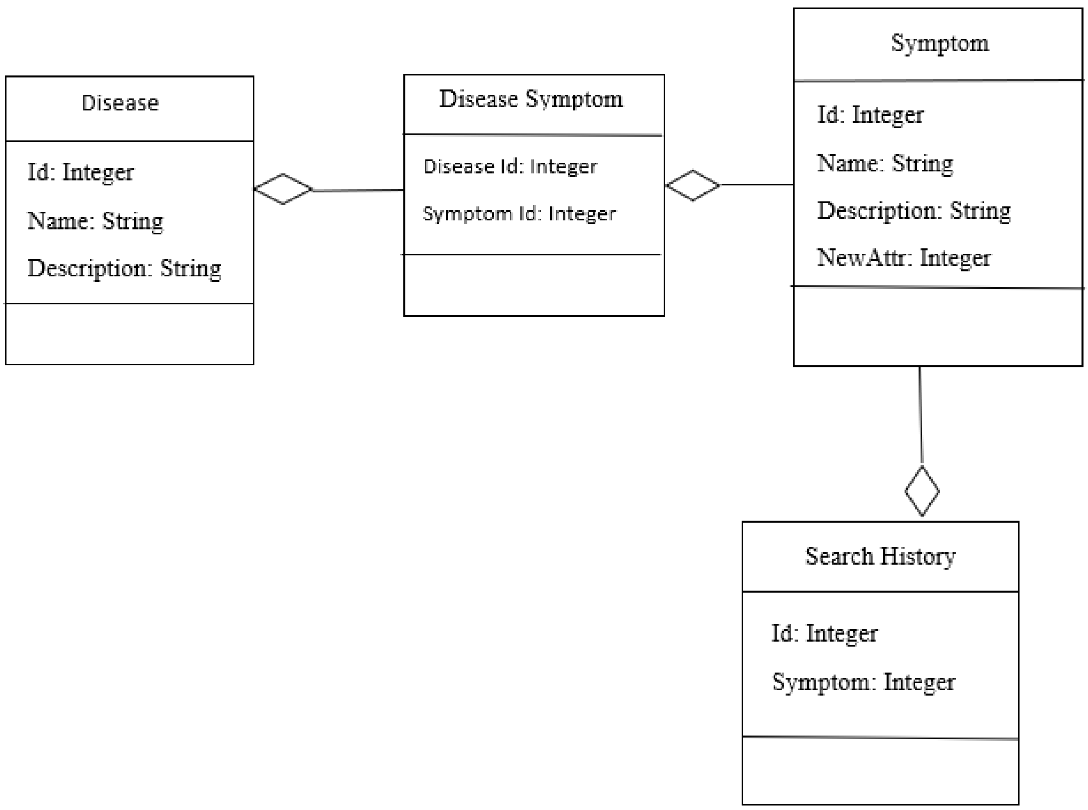


Fig 5.2.1.3: Class Diagram

5.2.1.4 COMPONENT DIAGRAM

The component diagram for the Multiple Disease Prediction System delineates the system's architecture, highlighting the key components and their interactions. At the User Interface Component, serving as the entry point for healthcare professionals to input patient data and access prediction results. This component relies on the Prediction Engine Component, which houses the core logic for processing patient data and generating disease predictions using machine learning and deep learning algorithms. The Data Processing Component plays a pivotal role in pre-processing and manipulating patient data before it is Fed into the Prediction Engine.

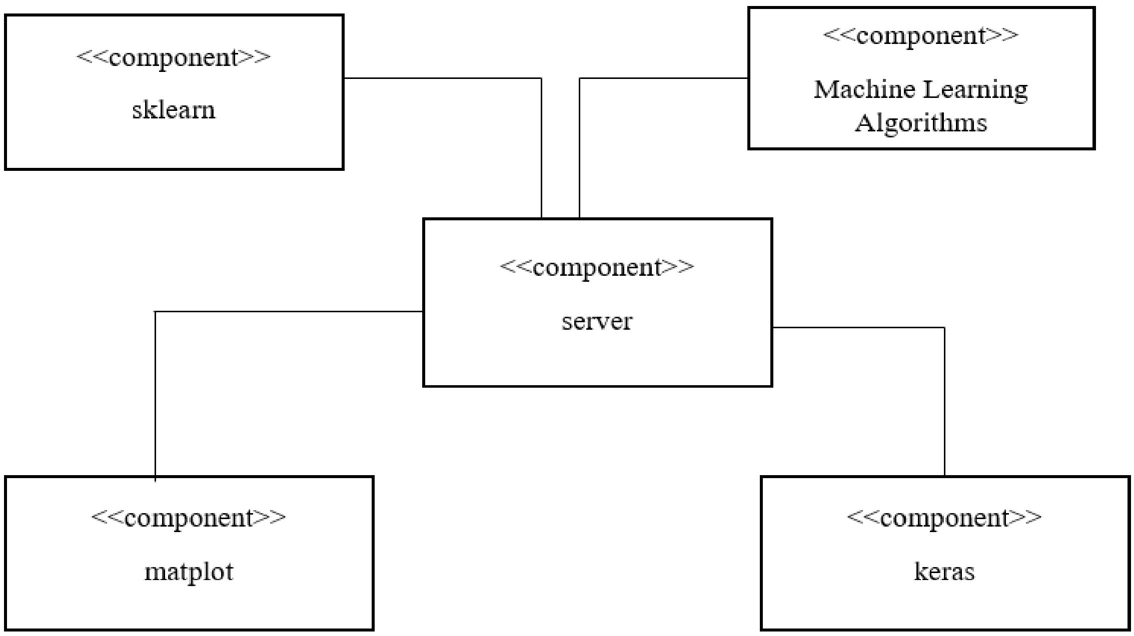


Fig-5.2.1.4: Component Diagram

5.4.1.5 Sequence Diagram

The sequence diagram for the Multiple Disease Prediction System depicts the step-by-step interactions between the system components during the disease prediction workflow. It begins with a healthcare professional initiating the prediction process through the user interface, followed by the input of patient data encompassing demographic details, medical history, and test results. Upon receiving the input data, the user interface component notifies the prediction engine to commence processing. Subsequently, the prediction engine engages the data processing component to pre-process and manipulate the patient data before utilizing disease prediction models stored within the system.

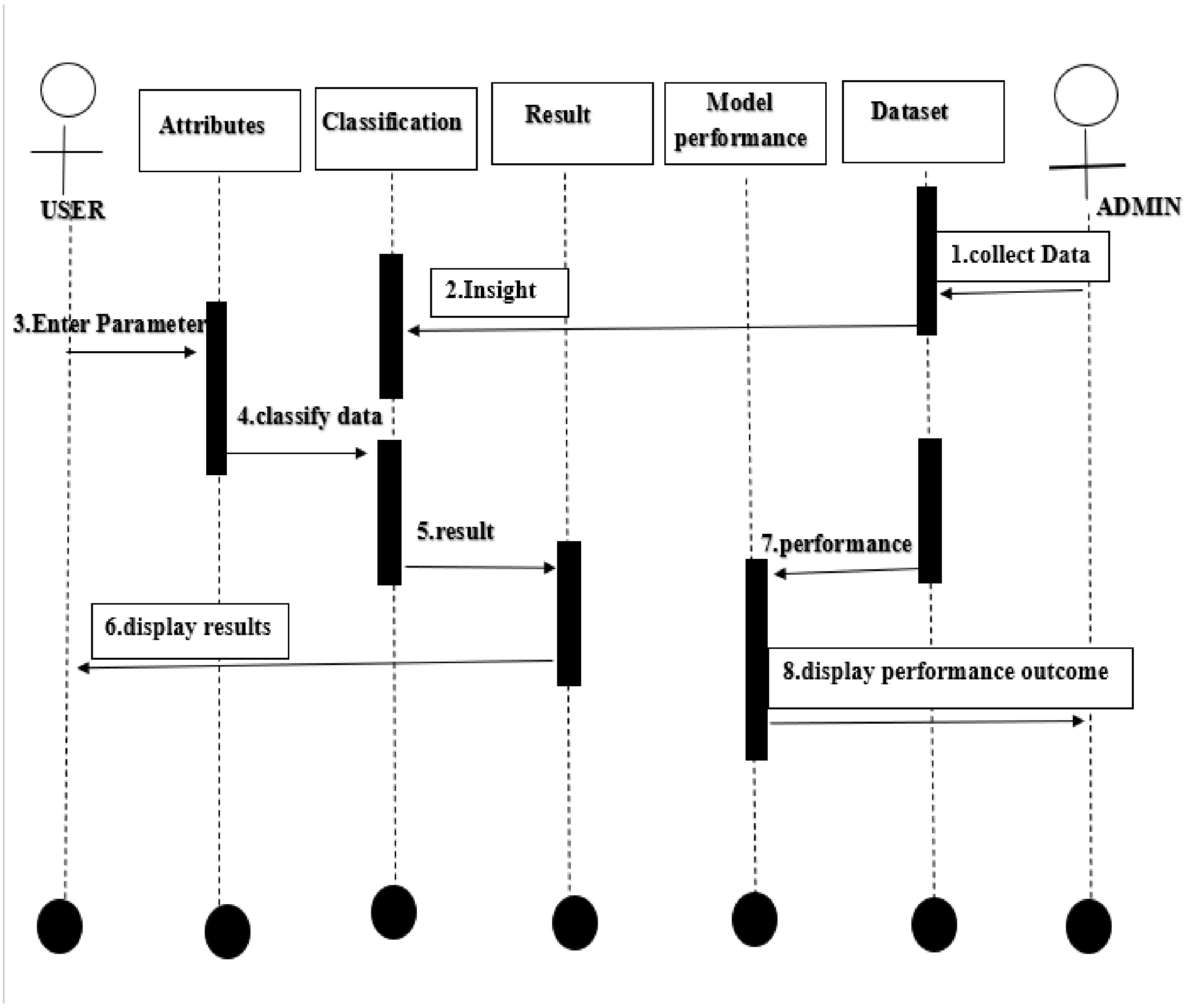


Fig-5.2.1.5: Sequence Diagram

CHAPTER 6

IMPLEMENTATION

CHAPTER 6

IMPLEMENTATION

6.1 STEPS FOR IMPLEMENTATION

Our implementation plan begins with meticulous project planning and requirement gathering, where we define clear objectives and identify the target diseases and relevant data sources. Data collection and preprocessing follow, ensuring data quality and suitability for modeling through handling missing values, normalization, and exploratory analysis. Feature selection and engineering enhance the predictive power of our models by leveraging domain knowledge and consulting with medical experts. Model development involves selecting appropriate algorithms, splitting data for training and validation, and fine-tuning model parameters for optimal performance.

In the evaluation phase, we compare model performance using standard metrics and select the best-performing models for each disease. Deployment involves integrating models into a user-friendly interface for real-time predictions while ensuring data security and compliance. Monitoring and maintenance ensure model accuracy and reliability over time through continuous performance monitoring and periodic retraining with new data. Documentation and reporting summarize the implementation process, outcomes, and findings for stakeholders and the broader community. This cohesive plan ensures the successful development and deployment of predictive models for multiple diseases, advancing healthcare outcomes.

6.1.1 IMPORTING PACKAGES

In our project, we import several essential packages to facilitate data preprocessing, model development, and deployment. These packages include scikit-learn for machine learning algorithms, TensorFlow or PyTorch for deep learning models, pandas for data manipulation, and matplotlib or seaborn for data visualization. Additionally, we utilize numpy for numerical computations, nltk for natural language processing tasks, and Flask or Django for web application development. These packages enable us to efficiently handle data, train predictive models, and create user-friendly interfaces for healthcare professionals.

6.1.2 EXPLORING DATASET

Exploring the dataset is a crucial step to gain insights into the data's characteristics and distribution. We conduct exploratory data analysis (EDA) to understand the structure of the data, identify patterns, and detect anomalies. This process involves visualizing the data using statistical plots, histograms, and heatmaps to assess feature distributions, correlations, and potential outliers. Additionally, we examine summary statistics such as mean, median, and standard deviation to understand the central tendencies and variability within the data. EDA allows us to make informed decisions regarding data preprocessing, feature selection, and modeling strategies, ensuring the robustness and reliability of our predictive models.

6.1.3 DATA PREPROCESSING

Our preprocessing pipeline involves handling missing values by imputation techniques such as mean, median, or mode imputation, or using advanced methods like predictive modeling. Additionally, we normalize features to bring them to a standard scale, reducing the impact of differences in magnitude among variables.

To address class imbalance, we employ techniques such as oversampling, under sampling, or synthetic data generation to ensure that each class is adequately represented in the dataset. Exploratory data analysis is conducted to understand the distribution and characteristics of the data, guiding further preprocessing steps.

6.1.4 BUILDING THE MODEL

In building the predictive models for our project, we leverage a diverse array of supervised machine learning algorithms. Beginning with meticulous data preprocessing and feature engineering, we ensure the quality and relevance of the input data. We then select appropriate algorithms, including logistic regression, decision trees, random forests, support vector machines, and gradient boosting machines for machine learning. Finally, the best-performing models are deployed into a production environment, integrated into a user-friendly interface for real-time predictions, and monitored for ongoing performance and maintenance. This comprehensive approach ensures the development of accurate and reliable predictive models for multiple diseases, facilitating proactive healthcare management.

6.2 CODE IMPLEMENTATION

In the coding phase of our project, we implement various machine learning algorithms, using Python and relevant libraries such as scikit-learn, TensorFlow, and PyTorch. This involves writing code to pre-process the collected medical data, select and engineer features, train and validate predictive models, and deploy them into a production environment. Additionally, we develop a user-friendly interface for healthcare professionals to interact with the deployed models in real-time. Security measures are integrated into the code to protect patient data and ensure compliance with healthcare regulations. Continuous monitoring and maintenance are implemented to ensure the long-term accuracy and reliability of the deployed models.

6.2.1 APP.PY

```
import os
from flask import Flask, render_template, request
import pickle
import numpy as np
from PIL import Image
import tensorflow as tf
app = Flask(__name__)
def predict(values, dic):
    # diabetes
    if len(values) == 8:
        dic2 = {'NewBMI_Obesity 1': 0, 'NewBMI_Obesity 2': 0, 'NewBMI_Obesity 
3': 0, 'NewBMI_Overweight': 0,
                'NewBMI_Underweight': 0, 'NewInsulinScore_Normal': 0, 
'NewGlucose_Low': 0,
                'NewGlucose_Normal': 0, 'NewGlucose_Overweight': 0, 
'NewGlucose_Secret': 0}
        if dic['BMI'] <= 18.5:
            dic2['NewBMI_Underweight'] = 1
        elif 18.5 < dic['BMI'] <= 24.9:
            pass
        elif 24.9 < dic['BMI'] <= 29.9:
            dic2['NewBMI_Overweight'] = 1
        elif 29.9 < dic['BMI'] <= 34.9:
            dic2['NewBMI_Obesity 1'] = 1
        elif 34.9 < dic['BMI'] <= 39.9:
            dic2['NewBMI_Obesity 2'] = 1
        elif dic['BMI'] > 39.9:
            dic2['NewBMI_Obesity 3'] = 1
        if 16 <= dic['Insulin'] <= 166:
            dic2['NewInsulinScore_Normal'] = 1
        if dic['Glucose'] <= 70:
```

```
            dic2['NewGlucose_Low'] = 1
        elif 70 < dic['Glucose'] <= 99:
            dic2['NewGlucose_Normal'] = 1
        elif 99 < dic['Glucose'] <= 126:
     dic2['NewGlucose_Overweight'] = 1
        elif dic['Glucose'] > 126:
            dic2['NewGlucose_Secret'] = 1
        dic.update(dic2)
        values2 = list(map(float, list(dic.values())))
        model = pickle.load(open('models/diabetes.pkl','rb'))
        values = np.asarray(values2)
        return model.predict(values.reshape(1, -1))[0]
    # breast_cancer
    elif len(values) == 22:
        model = pickle.load(open('models/breast_cancer.pkl','rb'))
        values = np.asarray(values)
        return model.predict(values.reshape(1, -1))[0]
    # heart disease
    elif len(values) == 13:
        model = pickle.load(open('models/heart.pkl','rb'))
        values = np.asarray(values)
        return model.predict(values.reshape(1, -1))[0]
    # kidney disease
    elif len(values) == 24:
        model = pickle.load(open('models/kidney.pkl','rb'))
        values = np.asarray(values)
        return model.predict(values.reshape(1, -1))[0]
    # liver disease
    elif len(values) == 10:
        model = pickle.load(open('models/liver.pkl','rb'))
        values = np.asarray(values)
        return model.predict(values.reshape(1, -1))[0]
@app.route("/")
def home():
    return render_template('home.html')
@app.route("/diabetes", methods=['GET', 'POST'])
def diabetesPage():
    return render_template('diabetes.html')
@app.route("/cancer", methods=['GET', 'POST'])
def cancerPage():
    return render_template('breast_cancer.html')
@app.route("/heart", methods=['GET', 'POST'])
ef heartPage():
    return render_template('heart.html')
@app.route("/kidney", methods=['GET', 'POST'])
def kidneyPage():
    return render_template('kidney.html')
```

```
@app.route("/liver", methods=['GET', 'POST'])
def liverPage():
    return render_template('liver.html')
@app.route("/predict", methods = ['POST', 'GET'])
def predictPage():
    try:
        if request.method == 'POST':
            to_predict_dict = request.form.to_dict()
            for key, value in to_predict_dict.items():
                try:
                    to_predict_dict[key] = int(value)
                except ValueError:
                    to_predict_dict[key] = float(value)
            to_predict_list = list(map(float, list(to_predict_dict.values())))
            pred = predict(to_predict_list, to_predict_dict)
    except:
        message = "Please enter valid data"
        return render_template("home.html", message=message)
    return render_template('predict.html', pred=pred)
if __name__ == '__main__':

   app.run(debug = True)
```

6.2.2 HOME.HTML

```
% extends 'main.html' %} {% block content %} {% if message %}
<div class="alert alert-danger">{{ message }}</div>
{% endif %}
<style>
  body {
    font-family: Cambria;
    height: 100%;
    background-image: linear-gradient(#4e0374, #c37ee6);
    margin: 0;
    background-repeat: no-repeat;
    background-attachment: fixed;
  }
  h1 {
    text-align: center;
  }
</style>
<div class="container">
  <div
    class="card card-body"
    style="
      border: 1px solid black;
      box-shadow: 0 0 10px black;
```

```
      background-color: #edc8ff;
    "
  >
    <h1>Smart Disease Predictor</h1>
    <br />
    <h2>Model Accuracies:</h2>
    <ul>
      <li>Diabetes Model: <strong>92.54%</strong></li>
      <li>Heart Disease Model: <strong>98.70%</strong></li>
      <li>Breast Cancer Model: <strong>97.66%</strong></li>
      <li>Kidney Disease Model: <strong>99.16%</strong></li>
      <li>Liver Disease Model: <strong>71.18%</strong></li>
    </ul>
    <br />
    <h3>Information about the Diseases:</h3>
    <br />
    <div class="row">
      <div
        class="col-md-12 card card-body"
        style="
          box-shadow: 0 0 5px black;
          margin: 5px;
          background-color: #daa8f4;"
      >
        <h4>Diabetes</h4>
        <p>
          Diabetes mellitus refers to a group of diseases that affect how your
          body uses blood sugar (glucose). Glucose is vital to your health
          because it's an important source of energy for the cells that make
up
          your muscles and tissues. It's also your brain's main source of
fuel.
          The underlying cause of diabetes varies by type. But, no matter what
          type of diabetes you have, it can lead to excess sugar in your
blood.
          Too much sugar in your blood can lead to serious health problems.
        </p>
        <h5>Symptoms</h5>
        <ul>
          <li>Increased thirst</li>
          <li>Frequent urination</li>
          <li>Extreme hunger</li>
          <li>Unexplained weight loss</li>
          <li>Blurred vision</li>
        </ul>
      </div>
    </div>
```

```
    <br />
    <div class="row">
      <div
        class="col-md-12 card card-body"
        style="
          box-shadow: 0 0 5px black;
          margin: 5px;
          background-color: #daa8f4;"
      >
        <h4>Heart/ Cardiovascular Disease</h4>
        <p>
          Cardiovascular disease (heart disease) refers to a group of diseases
          that affect the heart and blood vessels of your body. These diseases
          can affect one or many parts of your heart and /or blood vessels. A
          person may be symptomatic (physically experience the disease) or be
          asymptomatic (not feel anything at all).
        </p>
        <h5>Symptoms</h5>
        <ul>
          <li>Pounding or racing heart (palpitations)</li>
          <li>Chest pain</li>
          <li>Sweating</li>
          <li>Lightheadedness</li>
          <li>Shortness of breath</li>
        </ul>
      </div>
    </div>
    <br />
    <div class="row">
      <div
        class="col-md-12 card card-body"
        style="
          box-shadow: 0 0 5px black;
          margin: 5px;
          background-color: #daa8f4;”>
        <h4>Breast Cancer</h4>
        <p>
          Breast cancer is a type of cancer that starts in the breast. It can
          start in one or both breasts. After skin cancer, breast cancer is
the

          most common cancer diagnosed in women in the United States. Breast
          cancer can occur in both men and women, but it's far more common in
          women.
        </p>
        <h5>Symptoms</h5>
        <ul>
          <li>New lump in the breast or underarm (armpit)</li>
```

```
          <li>Thickening or swelling of part of the breast.</li>
          <li>Irritation or dimpling of breast skin</li>
          <li>Redness or flaky skin in the nipple area or the breast</li>
          <li>Pulling in of the nipple or pain in the nipple area</li>
        </ul>
      </div>
    </div>
    <br />
    <div class="row">
      <div
        class="col-md-12 card card-body"
        style="
          box-shadow: 0 0 5px black;
          margin: 5px;
          background-color: #daa8f4;”>
        <h4>Chronic Kidney Disease</h4>
        <p>
          Chronic kidney disease (CKD) means your kidneys are damaged and
can’t
          filter blood the way they should. The disease is called “chronic”
          because the damage to your kidneys happens slowly over a long period
          of time. This damage can cause wastes to build up in your body. CKD
          can also cause other health problems.
        </p>
        <h5>Symptoms</h5>
        <ul>
          <li>Nausea</li>
          <li>Vomiting</li>
          <li>Fatigue and weakness</li>
          <li>Muscle twitches and cramps</li>
          <li>Loss of appetite</li>
        </ul>
      </div>
    </div>
    <br />
    <div class="row">
      <div
        class="col-md-12 card card-body"
        style="
          box-shadow: 0 0 5px black;
          margin: 5px;
          background-color: #daa8f4;">
        <h4>Liver Disease</h4>
        <p>
          The term “liver disease” refers to any of several conditions that
can
          affect and damage your liver. Over time, liver disease can cause
```

```
          cirrhosis (scarring). As more scar tissue replaces healthy liver
          tissue, the liver can no longer function properly. Left untreated,
          liver disease can lead to liver failure and liver cancer.
        </p>
        <h5>Symptoms</h5>
        <ul>
          <li>Abdominal (belly) pain (especially on the right side)</li>
          <li>Bruising easily</li>
          <li>Changes in the color of your urine or stool</li>
          <li>Fatigue</li>
          <li>Nausea or vomiting</li>
        </ul>
      </div>
    </div>
    <br />
  </div>
</div>
{% endblock %}
```

6.2.3 MAIN.HTML

```
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta name="og:title" content="Medibuddy: Smart Disease Predictor" />
    <meta name="author" content="KANCHI TANK" />
    <meta name="og:image" content="static/favicon.ico" />
    <title>Medibuddy</title>
    <link rel="icon" type="image/x-icon" href="static/favicon.ico?v=2" />
    <link
      rel="stylesheet"
      href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.m
in.css"
      integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
      crossorigin="anonymous"
    />
    <link
      href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css"
      rel="stylesheet"
    />
    <link
      rel="canonical"
      href="https://getbootstrap.com/docs/4.0/examples/sticky-footer/"
    />
    <script
```

```
      src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
      integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
      crossorigin="anonymous"
    ></script>
    <script
      src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.
min.js"
      integrity="sha384-
UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
      crossorigin="anonymous"
    ></script>
    <script
      src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min
.js"
      integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
      crossorigin="anonymous"
    ></script>

    <style>
      html,
      body {
        height: 100%;
        margin: 0;
      }
      header {
        height: 80px;
      }
      footer {
        height: 75px;
      }

      body {
        display: flex;
        flex-direction: column;
      }

      footer {
        padding: 10px;
        margin-top: auto;
        margin-bottom: auto;
      }
    </style>
  </head>
  <body>
    <nav
```

```
    class="navbar navbar-expand-lg navbar-dark fixed-top bg-dark"
    style="background-image: linear-gradient(#4e0374, #4e0374) !important"
  >
    <a href="{{ url_for('home') }}"
      ><img
        src="{{ url_for('static', filename = 'logo.png') }} "
        height="70px"
        width="180px"
    /></a>
    <button
      class="navbar-toggler"
      type="button"
      data-toggle="collapse"
      data-target="#navbarNav"
      aria-controls="navbarNav"
      aria-expanded="false"
      aria-label="Toggle navigation"
    >
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav ml-auto">
        <li class="nav-item active">
          <a href="{{ url_for('home') }}" class="nav-link">Home</a>
        </li>
        <li class="nav-item active">
          <a class="nav-link" href="{{ url_for('cancerPage') }}"
            >Breast Cancer</a
          >
        </li>
        <li class="nav-item active">
          <a class="nav-link" href="{{ url_for('heartPage') }}">Heart</a>
        </li>
        <li class="nav-item active">
          <a class="nav-link" href="{{ url_for('kidneyPage') }}">Kidney</a>
        </li>
        <li class="nav-item active">
          <a class="nav-link" href="{{ url_for('diabetesPage') }}"
            >Diabetes</a
          >
        </li>
        <li class="nav-item active">
          <a class="nav-link" href="{{ url_for('liverPage') }}">Liver</a>
        </li>
        </li>
      </ul>
    </div>
```

```
    </nav>
    <br />
    <br />
    <br />
    <br />
    <main>
      <div class="container-fluid" style="margin-bottom: 20px">
        {% block content %} {% endblock %}
      </div>
    </main>
    <footer
      style="background-image: linear-gradient(#4e0374, #4e0374) !important">
      <center>
        <ul style="list-style-type: none">
          <li>
            <p style="color: white">
              Made <span style="color: red"></span> by Tulasi , Shanmukhi ,
              Raveena, Harshitha
            </p>
          </li>
        </ul>
      </center>
    </footer>
  </body>
</html>
```

6.3 SCREENSHOTS

BREAST CANCER PREDICTOR

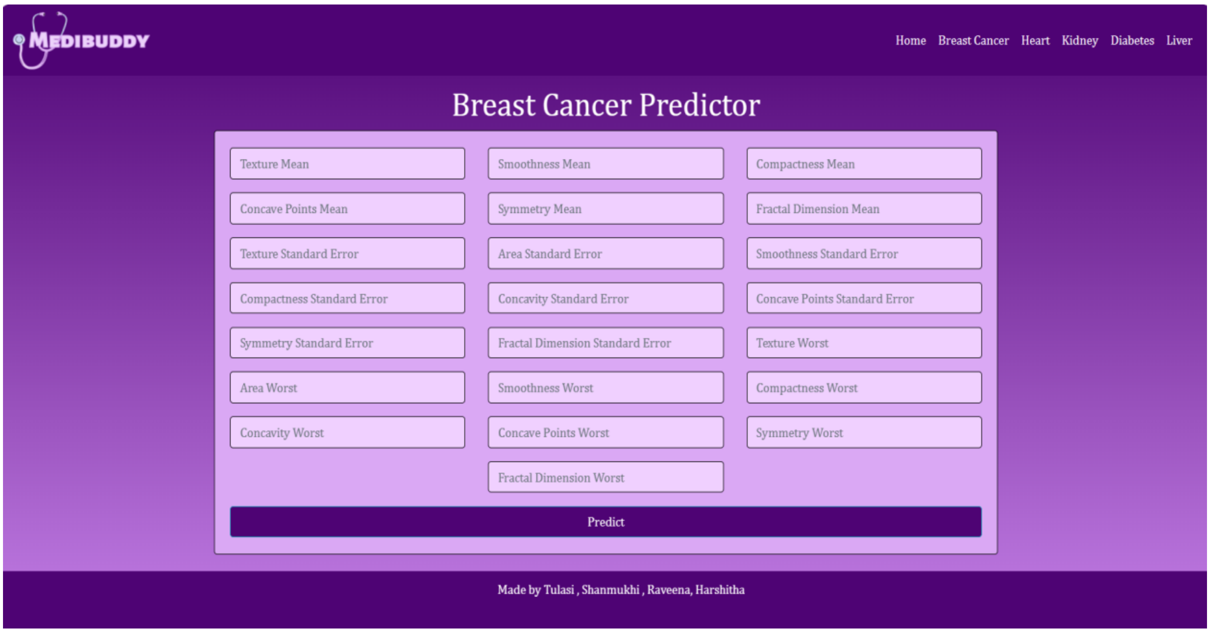


Fig-6.3.1: Breast Cancer Predictor Page

HEART DISEASE PREDICTOR

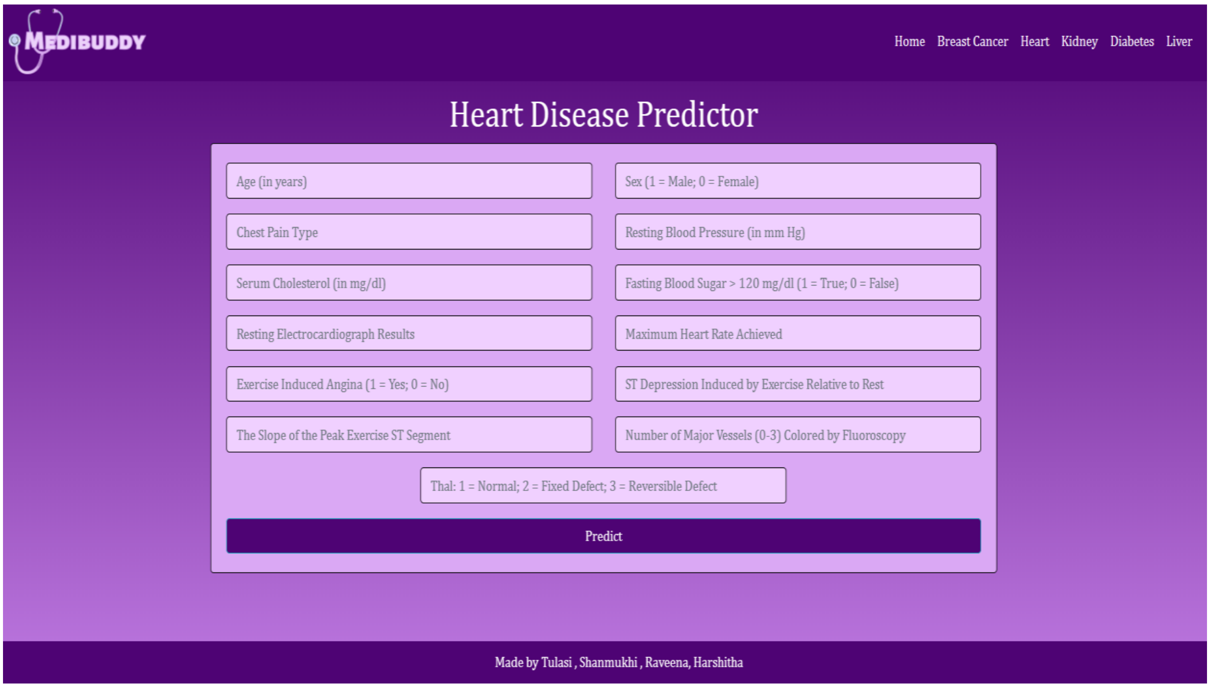


Fig-6.3.2: Heart Disease Predictor page

KIDNEY DISEASE PREDICTOR

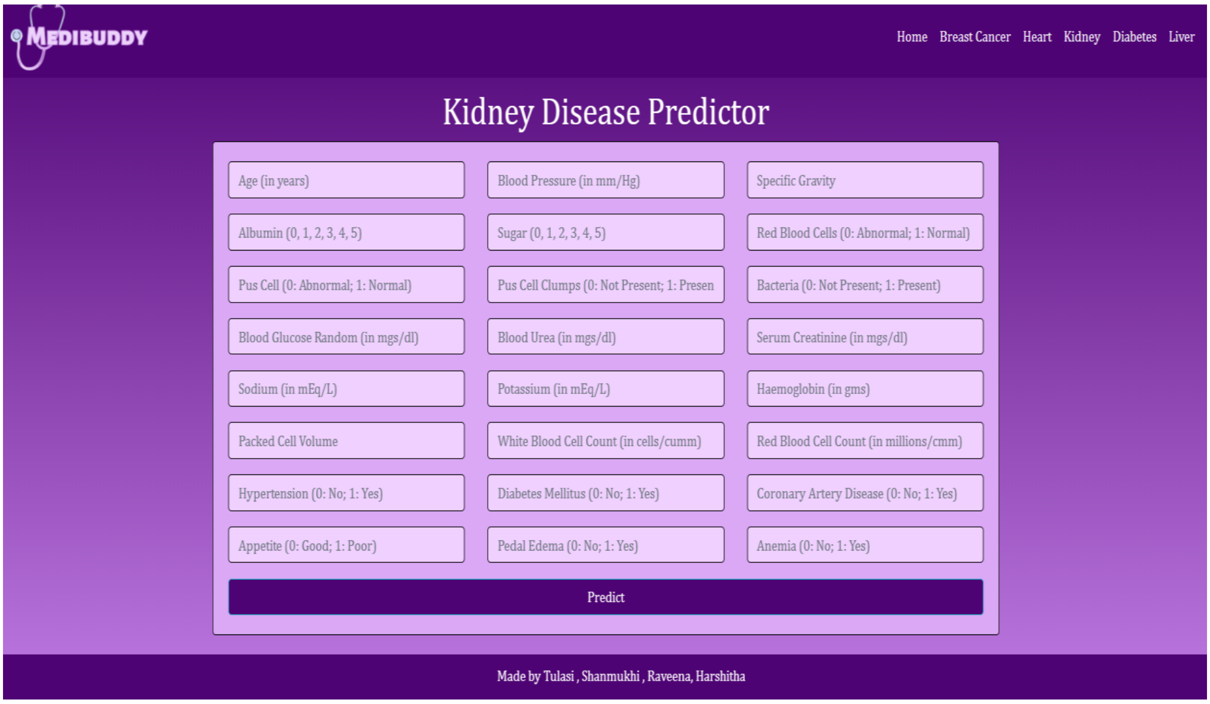


Fig-6.3.3: Kidney Disease Predictor page

DIABETES PREDICTOR

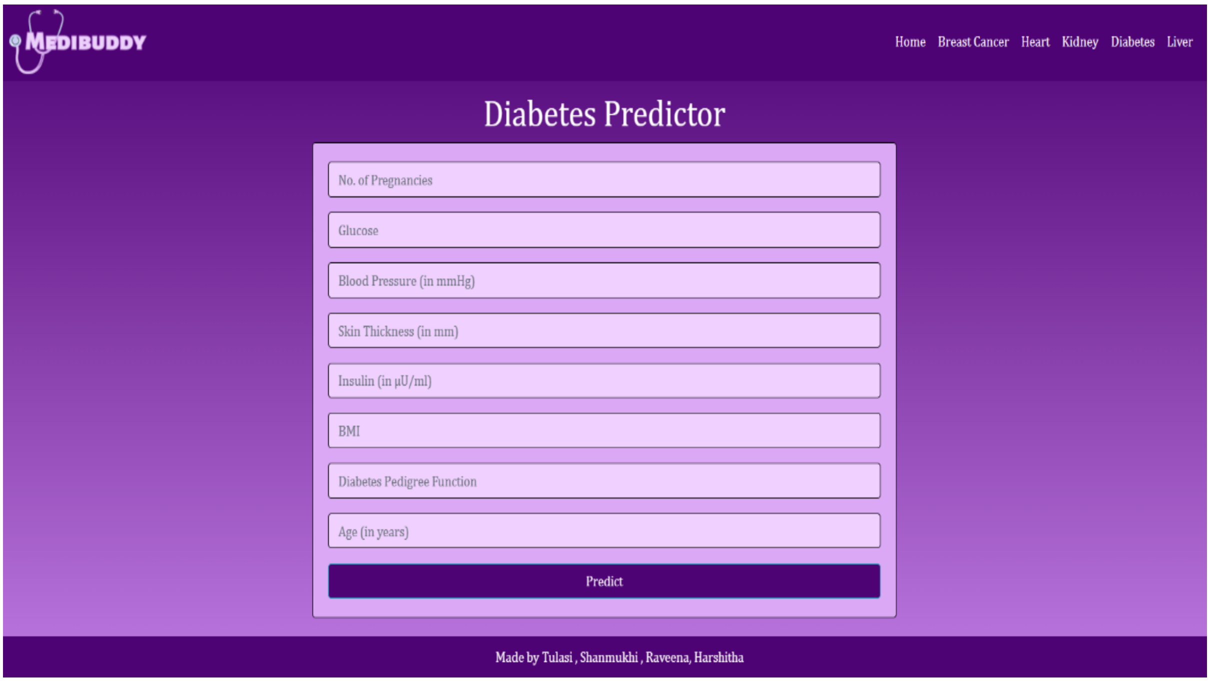


Fig-6.3.4: Diabetes predictor page

LIVER DISEASE PREDICTOR

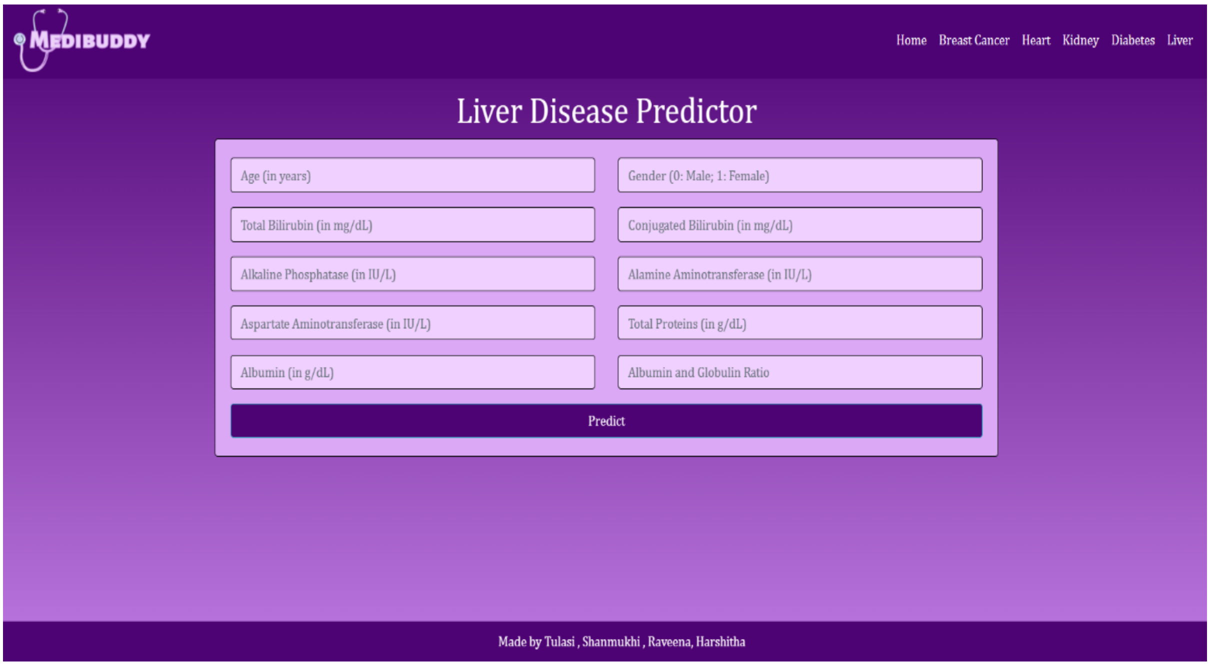
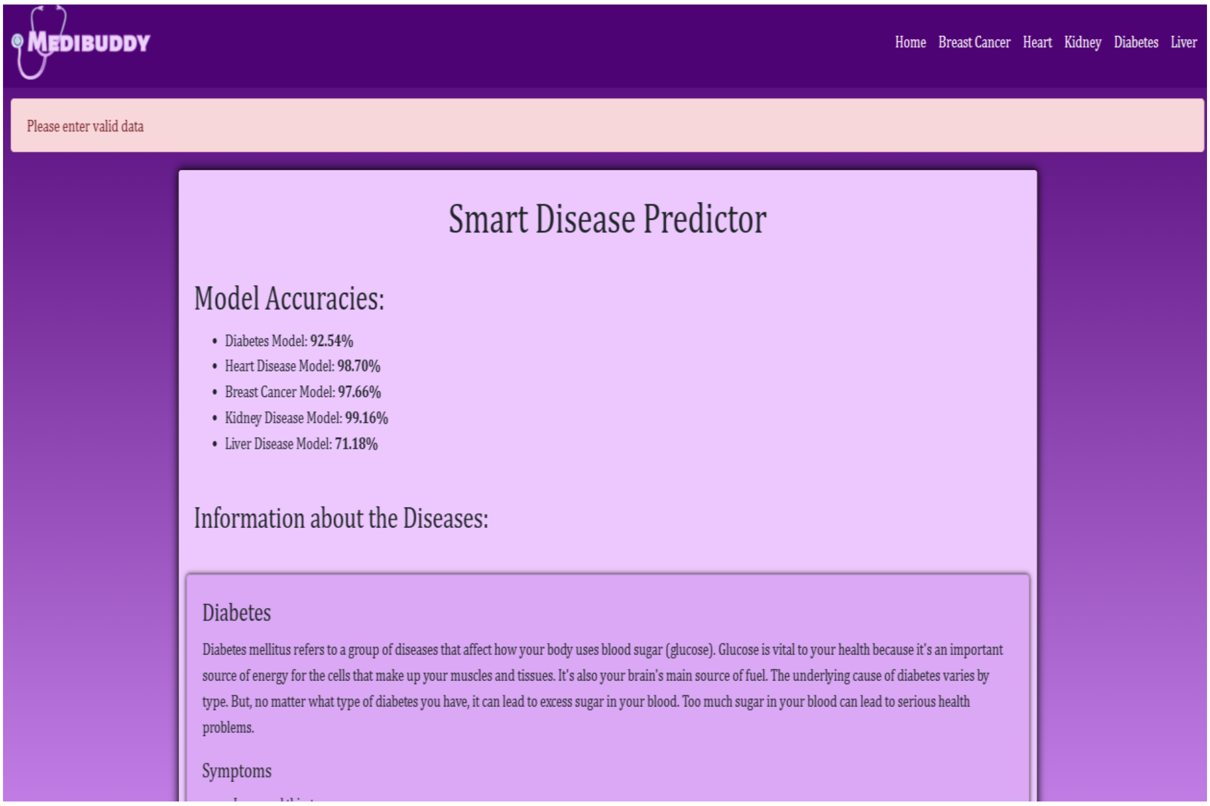


Fig-6.3.5 Liver Disease Predictor page

RESULT


MEDIBUDDY
Home Breast Cancer Heart Kidney Diabetes Liver
Please enter valid data
Smart Disease Predictor
Model Accuracies:
Diabetes Model: 92.54%
Heart Disease Model: 98.70%
Breast Cancer Model: 97.66%
Kidney Disease Model: 99.16%
Liver Disease Model: 71.18%
Information about the Diseases:
Diabetes
Diabetes mellitus refers to a group of diseases that affect how your body uses blood sugar (glucose). Glucose is vital to your health because it's an important source of energy for the cells that make up your muscles and tissues. It's also your brain's main source of fuel. The underlying cause of diabetes varies by type. But, no matter what type of diabetes you have, it can lead to excess sugar in your blood. Too much sugar in your blood can lead to serious health problems.
Symptoms

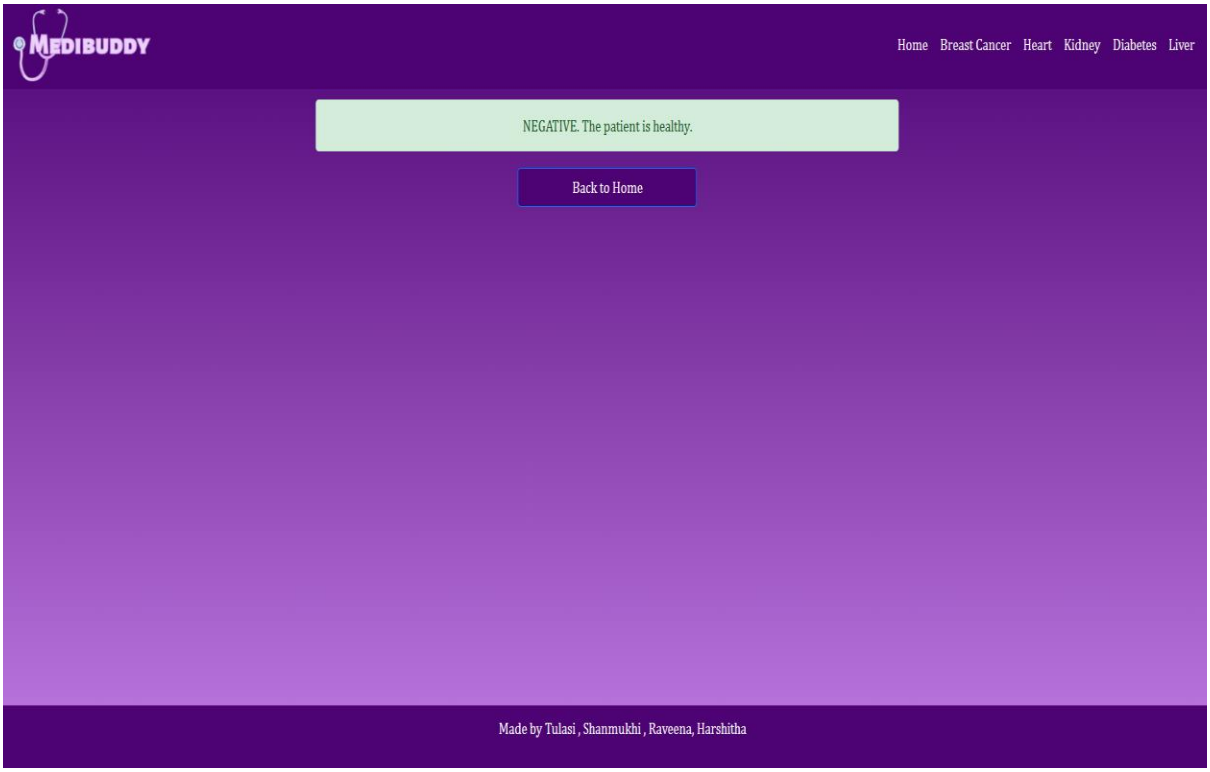

MEDIBUDDY
Home Breast Cancer Heart Kidney Diabetes Liver
NEGATIVE. The patient is healthy.
Back to Home
Made by Tulasi , Shanmukhi , Raveena, Harshitha

Fig-6.3.6: Result Pages

CHAPTER 7

SYSTEM TESTING

CHAPTER 7

SYSTEM TESTING

7.1 TESTING

Testing serves to uncover errors and weaknesses within a system. It involves systematically exploring various components and functionalities to ensure software meets its requirements and user expectations without unacceptable failure. Two primary testing methodologies, Manual Testing and Automation Testing, are employed. Manual Testing involves hands-on examination of software, mimicking end-user behaviour to identify bugs across stages like unit testing, integration testing, system testing, and user acceptance testing. In contrast, Automation Testing utilizes scripts and software tools to execute tests quickly and repetitively, especially beneficial for areas like login and registration forms or those accessed by multiple users concurrently. The decision to automate testing considers project size, frequency of testing, stability, and time availability. Automation involves selecting suitable tools, developing test suites, executing scripts, and identifying any potential issues or bugs.

7.2 TYPES OF TESTING

7.2.1 UNIT TESTING

Designing test cases for unit testing ensures that the core programmed logic is working correctly and that program inputs result in legitimate outputs. It is important to verify the internal code flow and all decision branches. It is the testing of the application's separate software components. Before integration, it is done following the completion of each individual unit. This is an intrusive structural test that depends on understanding how it was built. Unit tests carry out fundamental tests at the component level and examine a particular configuration of a system, application, or business process. Unit tests make assurance that each distinct route of a business process adheres precisely to the stated specifications and has inputs and outputs that are well-defined Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases. Test strategy and approach Field testing will be performed manually, and functional tests will be written in detail.

TEST OBJECTIVES

- All field entries must work properly.
- Pages must be activated from the identified link.
- Delays on the entering screen, messages, or answers are not acceptable.

7.2.2 INTEGRATION TESTING

Software components that have been merged are tested in integration tests to see if they genuinely operate as a single program. Testing is event-driven and focuses more on the fundamental result of screens or fields. Even if the individual components were successful in unit testing, integration tests indicate that the combination of the components is accurate and consistent. Integration testing is especially designed to highlight issues that result from combining components. Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, eg. Components in a software system or one step up software applications at the company level - interact without error.

7.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals. Functional testing is centered on the following items:

- **Valid Input:** Identified classes of valid input must be accepted.
- **Invalid Input:** Identified classes of invalid input must be rejected
- **Functions:** Identified functions must be exercised.
- **Output:** Identified classes of application outputs must be exercised.
- **Systems/Procedures:** Interfacing systems or procedures must be invoked

Functional tests are organized and prepared with a focus on requirements, important functionalities, or unique fest cases. Additionally, testing must take into account systematic coverage of data fields, established procedures, and subsequent processes as well as business Process flows. Additional tests are found, and the usefulness of the existing tests is assessed, before functional testing is finished. System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results.

An example of system testing is the configuration-oriented system integration test.

7.2.4 WHITE BOX TESTING

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

7.2.5 BLACK BOX TESTING

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

7.2.6 ACCEPTANCE TESTING

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

CHAPTER 8

CONCLUSION AND FUTURE WORK

CHAPTER 8
CONCLUSION AND FUTURE WORK

8.1 CONCLUSION

The Multiple Disease Prediction System presents a comprehensive solution for healthcare professionals to leverage advanced computational techniques in disease prediction. Through the orchestrated interactions of its components, including the user interface, prediction engine, data processing module, disease prediction models, and external database, the system streamlines the workflow of processing patient data and generating accurate predictions for multiple diseases. By integrating machine learning algorithms tailored for each disease prediction task, the system empowers healthcare professionals with valuable insights into potential health risks for their patients. This facilitates multiple disease prediction in a single system with its user-friendly interface and robust backend infrastructure. The Multiple Disease Prediction System such as diabetes prediction, breast cancer prediction, kidney disease prediction, heart disease prediction and liver disease prediction which stands as a testament to the potential of technology in revolutionizing healthcare decision-making and enhancing patient care. Furthermore, the system's adherence to data privacy regulations and ethical guidelines ensures the confidentiality and security of patient information, fostering trust and acceptance among healthcare professionals and patients alike. Overall, the Multiple Disease Prediction System represents a paradigm shift in healthcare delivery, where the synergy between human expertise and computational intelligence leads to more effective, patient-centered care, ultimately contributing to healthier communities and populations.

8.2 FURTHER ENHANCEMENT

The Multiple Disease Prediction System, already a robust tool for healthcare professionals, can undergo further enhancements to elevate its effectiveness and utility in clinical practice. Integration of real-time data sources, such as wearable devices and electronic health records, would enable the system to provide continuous updates on patient health status, enhancing its adaptability and accuracy. Interactive visualization tools within the user interface would empower healthcare professionals to explore and analyze patient data more effectively, identifying patterns and trends in disease risk factors with greater clarity. Moreover, personalized risk assessment capabilities could be expanded by incorporating additional patient-specific factors like lifestyle habits and genetic predispositions, ensuring more tailored and precise predictions. A decision support system could provide actionable recommendations based on predicted disease risks, guiding healthcare professionals in treatment decisions and preventive measures. Continuous model training and updating mechanisms would ensure that the prediction models remain accurate and up-to-date with the latest medical knowledge.

Collaboration features, mobile application support, and longitudinal health monitoring capabilities would further enhance accessibility, convenience, and proactive healthcare management. Through these enhancements, the Multiple Disease Prediction System can continue to advance healthcare delivery, facilitating more informed decision-making and improved patient outcomes.

Facilitating collaboration and knowledge sharing among healthcare professionals through networking features within the system would foster a culture of continuous learning and improvement. Discussion forums, case studies, and knowledge repositories could enable healthcare professionals to exchange insights, best practices, and experiences related to disease prediction and management, ultimately enhancing the collective expertise and effectiveness of the healthcare community. Extending the capabilities of the system to support longitudinal health monitoring would enable tracking changes in patient health status over time. By monitoring disease progression or the emergence of new health risks longitudinally, healthcare professionals could proactively intervene with personalized care plans and preventive measures, leading to better health outcomes and improved patient well-being.

BIBILOGRAPHY

BIBILOGRAPHY

[1] S. V. Topolyanskaya, T. M. Kolontai, O. N. Vakulenko, L. I., et al. (2020) explore “Diabetes Mellitus in Very Elderly Russian Patients with Coronary Artery Disease” Published in the Journal of Biosciences and Medicines, their study involves 580 CAD patients aged over 75, with 190 having diabetes.

[2] Mai Shouman, Timothy Leonard Turner, and Rob Stocker (2012) delve into "Applying k-Nearest Neighbour in Diagnosing Heart Disease Patients" in the International Journal of Information and Education Technology.

[3] Yang, Y., Miao, C., Wang, Y., He, J., et al. (2023) conduct a systematic review and meta-analysis titled "Long-term effects of bariatric/metabolic surgery vs. pharmacologic therapy in type 2 diabetes mellitus patients." They assess outcomes including diabetes remission, microvascular and macrovascular complications, and mortality over a five-year period or more.

[4] Dhafar Hamed, Jwan K. Alwan, Mohamed Ibrahim, and Mohammad B. Naeem (2017) discuss "The Utilisation of Machine Learning Approaches for Medical Data Classification" in the Annual Conference on New Trends in Information & Communications Technology Applications, March 2017.

[5] A. J. Jenkins, M. V. Joglekar, A. A. Hardikar, A. C. Keech, D. N. O'Neal, and A. S. Januszewski (2015) explore "Biomarkers in diabetes Prediction" in The Review of Diabetic Studies.

[6] M.M. Mehdy, E.E. Shair, and P.Y. Ng (2017) present "Artificial Neural Networks in Image Processing for Earlier Detection of Breast Cancer" in Hindawi's Computational and Mathematical Methods in Medicine.

[7] Parul Sinha and Poonam Sinha (2015) conduct a "Comparative study of chronic kidney disease prediction using KNN and SVM" in the International Journal of Engineering Research and Technology.

[8] Vidya Zope, Pooja Ghatge, Aaron Cherian, Piyush Mantri, and Kartik Jadhav (2017) present "Smart Health Prediction using Machine Learning" in the IJSRD International Journal for Scientific Research & Development.

[9]. Prof. Shalu Saraswat et al. (2024) created a "Smart Health Prediction Using Machine Learning" system, forecasting diseases based on user symptoms. The system, accessible to patients, doctors, and admins, utilizes the Naive Bayes Classifier for accurate predictions, aiding in early detection and understanding of diseases.

[10]. Jayshree Pawar et al. (2021) created a "Smart Disease Prediction" system at Vasantdada Patil Prathishtan College of Engineering, Mumbai, India, using AI and ML algorithms to analyse and predict diseases.

[11]. Dr. Nandini C, Antara Mukherjee, and Bhoomika M (2022) created a "Smart Health Prediction System Using Machine Learning Techniques" at Dayananda Sagar Academy of Technology and Management, Bengaluru, India. The system predicts disease risks based on user blood test reports, facilitating early diagnosis of conditions like Diabetes, Heart Disease, and Cancer through a user-friendly web interface.

[12]. N. Shabaz Ali and G. Divya (2023) created a “Prediction of Diseases in Smart Health Care System using Machine Learning "explored data mining techniques for disease prediction in Smart Health Care Systems, emphasizing the role of machine learning algorithms in enhancing prediction accuracy from user symptoms.

[13]. A.N.V.K Swarupa (2021) et al. presented a “Smart Disease Prediction System using the Random Forest Algorithm” at the IEEE International Conference. The system achieves 95% classification accuracy through machine learning and Python programming, aiding in early disease prediction based on lifestyle and environmental factors.

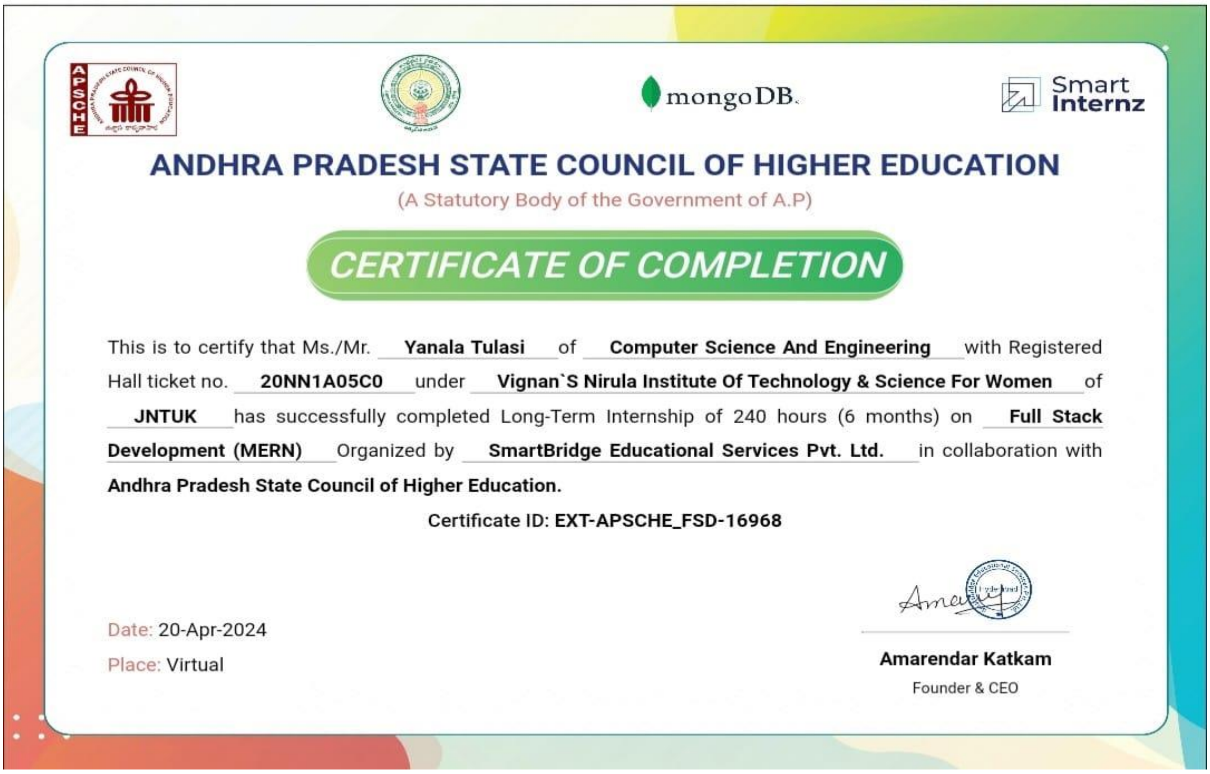
[14].D.Dahiwade, Meshram (2019) et al proposed “Designing Disease Prediction Model Using Machine Learning” with Computing Methodologies and Communication. Utilizing K-Nearest Neighbor (KNN) and Convolutional Neural Network (CNN) algorithms, the system accurately predicts diseases based on patient symptoms, considering lifestyle habits and checkup information. The CNN algorithm achieves an accuracy of 84.5%, surpassing KNN, with lower time and memory requirements.

[15]. S.A.Siddiqui, Anwar, Neda Fatima (2022) et al. Presented an IOT-based disease prediction using machine learning” The paper discusses the development of an IoT-based disease prediction system to address challenges faced by the healthcare sector, exacerbated by the COVID-19 pandemic.

[16]. Dr. C K Gomathy, Mr. A. Rohith Naidu (2023) proposed “The Prediction of Disease Using Machine Learning”. The system predicts diseases based on user-provided symptoms, utilizing the Naïve Bayes classifier, a supervised machine learning algorithm.

[17]. Rinkal Keniya (2020) et al proposed “Integrating machine learning for smart disease prediction”. Focused on diseases such as breast cancer and kidney ailments, our aim is to leverage machine learning algorithms for accurate prediction and analysis of these health conditions. The system utilizes multiple ML algorithms to predict diseases based on symptoms, age, and gender.

INTERNSHIP CERTIFICATES



mongoDB.

Smart Internz

ANDHRA PRADESH STATE COUNCIL OF HIGHER EDUCATION

(A Statutory Body of the Government of A.P)

CERTIFICATE OF COMPLETION

This is to certify that Ms./Mr. **Yanala Tulasi** of **Computer Science And Engineering** with Registered Hall ticket no. **20NN1A05C0** under **Vignan`S Nirula Institute Of Technology & Science For Women** of **JNTUK** has successfully completed Long-Term Internship of 240 hours (6 months) on **Full Stack Development (MERN)** Organized by **SmartBridge Educational Services Pvt. Ltd.** in collaboration with **Andhra Pradesh State Council of Higher Education.**

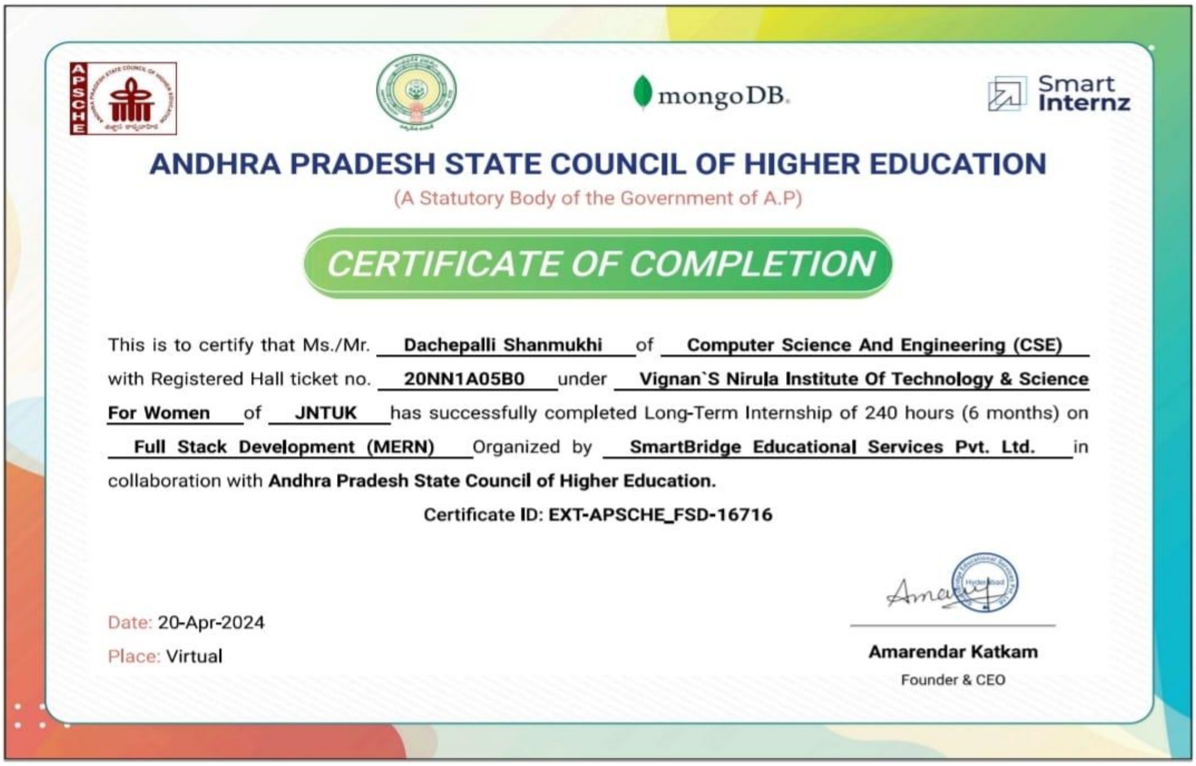
Certificate ID: **EXT-APSCHE_FSD-16968**

Date: 20-Apr-2024

Place: Virtual

Amarendar Katkam

Founder & CEO



mongoDB.

Smart Internz

ANDHRA PRADESH STATE COUNCIL OF HIGHER EDUCATION

(A Statutory Body of the Government of A.P)

CERTIFICATE OF COMPLETION

This is to certify that Ms./Mr. **Dachepalli Shanmukhi** of **Computer Science And Engineering (CSE)** with Registered Hall ticket no. **20NN1A05B0** under **Vignan`S Nirula Institute Of Technology & Science For Women** of **JNTUK** has successfully completed Long-Term Internship of 240 hours (6 months) on **Full Stack Development (MERN)** Organized by **SmartBridge Educational Services Pvt. Ltd.** in collaboration with **Andhra Pradesh State Council of Higher Education.**

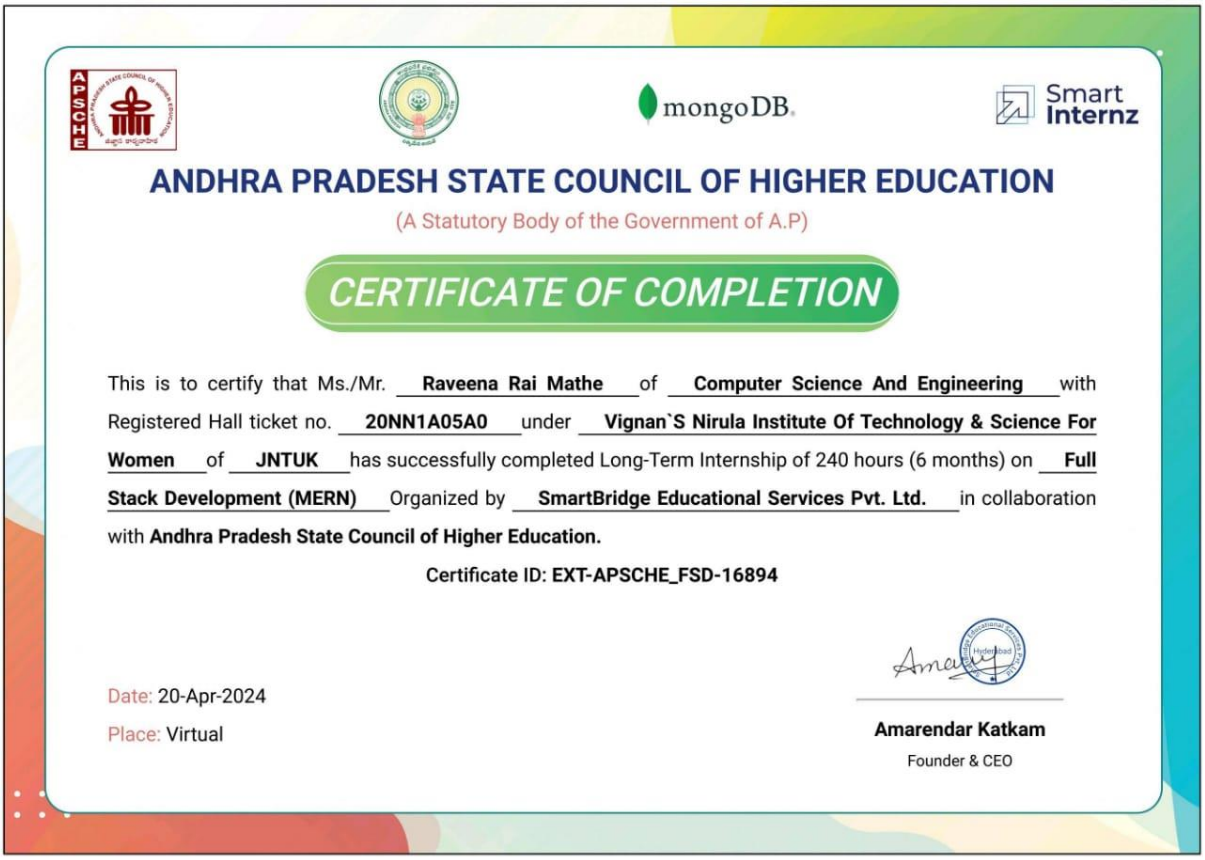
Certificate ID: **EXT-APSCHE_FSD-16716**

Date: 20-Apr-2024

Place: Virtual

Amarendar Katkam

Founder & CEO



ANDHRA PRADESH STATE COUNCIL OF HIGHER EDUCATION

(A Statutory Body of the Government of A.P)

CERTIFICATE OF COMPLETION

This is to certify that Ms./Mr. **Raveena Rai Mathe** of **Computer Science And Engineering** with Registered Hall ticket no. **20NN1A05A0** under **Vignan`S Nirula Institute Of Technology & Science For Women** of **JNTUK** has successfully completed Long-Term Internship of 240 hours (6 months) on **Full Stack Development (MERN)** Organized by **SmartBridge Educational Services Pvt. Ltd.** in collaboration with **Andhra Pradesh State Council of Higher Education.**

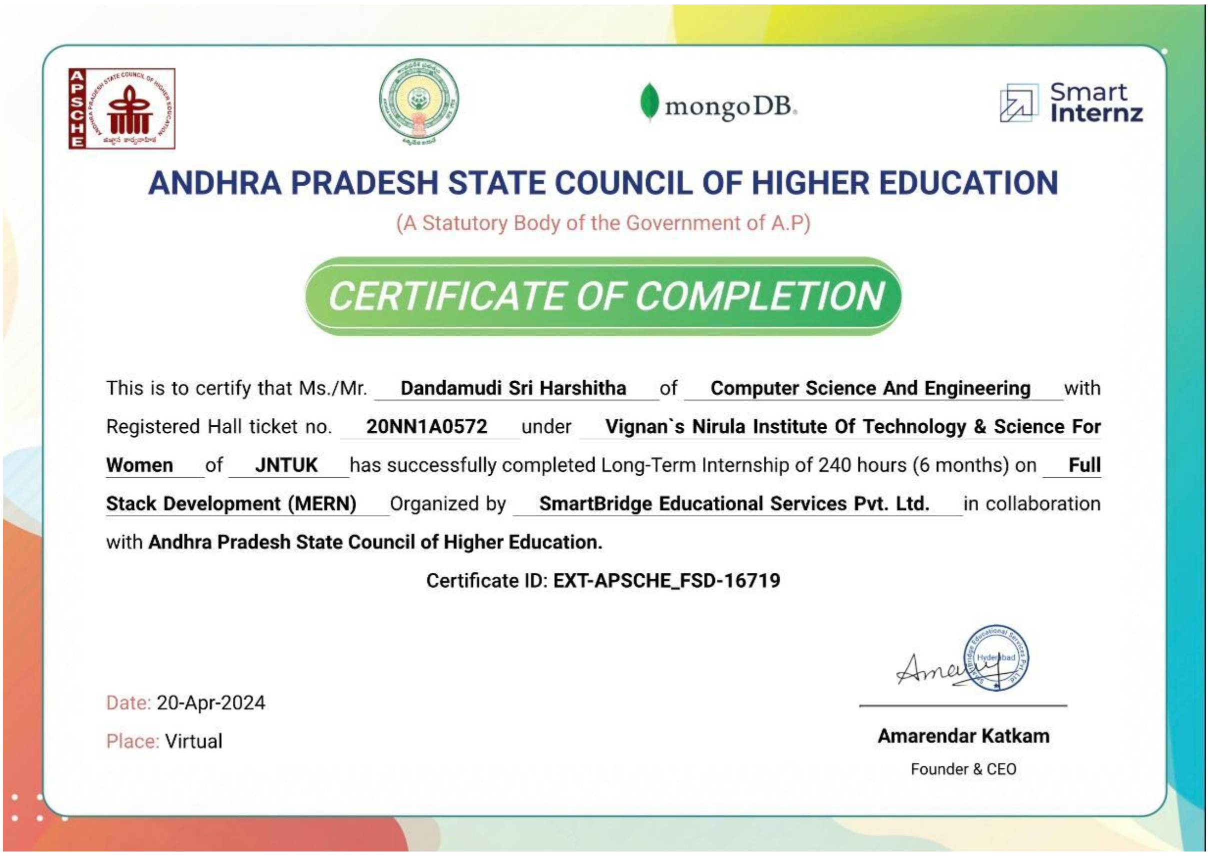
Certificate ID: **EXT-APSCHE_FSD-16894**

Date: 20-Apr-2024

Place: Virtual

Amarendar Katkam

Founder & CEO



ANDHRA PRADESH STATE COUNCIL OF HIGHER EDUCATION

(A Statutory Body of the Government of A.P)

CERTIFICATE OF COMPLETION

This is to certify that Ms./Mr. **Dandamudi Sri Harshitha** of **Computer Science And Engineering** with Registered Hall ticket no. **20NN1A0572** under **Vignan`s Nirula Institute Of Technology & Science For Women** of **JNTUK** has successfully completed Long-Term Internship of 240 hours (6 months) on **Full Stack Development (MERN)** Organized by **SmartBridge Educational Services Pvt. Ltd.** in collaboration with **Andhra Pradesh State Council of Higher Education.**

Certificate ID: **EXT-APSCHE_FSD-16719**

Date: 20-Apr-2024

Place: Virtual

Amarendar Katkam

Founder & CEO