A PROJECT REPORT

ON

AI POWERED HEALTH CONNECT KIOSK

Submitted in the partial fulfillment of the requirements for the award of

BACHELOR OF TECHNOLOGY
IN
INFORMATION TECHNOLOGY

SUBMITTED BY
BELLAMKONDA RAGHU
22BK5A1203

Under the guidance of
Dr. K. LITTLE FLOWER, PhD
Associate Professor

DEPARTMENT OF INFORMATION TECHNOLOGY

St. Peter's Engineering College (UGC Autonomous)

Approved by AICTE, New Delhi, and NAAC Accredited with 'A' Grade,

Affiliated to JNTU, Hyderabad, Telangana.

2024-2025

St. PETER'S ENGINEERING COLLEGE

UGC - AUTONOMOUS

Affiliated to JNTUH, Approved by AICTE, Accredited by NAAC with "A" Grade, NBA Programe Accredited (EEE, CSE, ECE)

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that a Project entitled **"AI POWERED HEALTH CONNECT KIOSK"** is carried out by **BELLAMKONDA RAGHU (22BK5A1203)**, in partial fulfillment for the award of the degree of **Bachelor of Technology in Information Technology** is a record of bonafide work done by her/him under my supervision during the academic year 2024– 2025.

INTERNAL GUIDE
Dr. K. Little Flower, Ph. D
Associate Professor,
Department of CSE
St. Peter's Engineering College,
Hyderabad

HEAD OF THE DEPARTMENT
Department of CSE
St. Peter's Engineering College,
Hyderabad

PROJECT COORDINATOR
Mr. A. Senthil Murugan, M.E., M.B.A., (Ph. D)
Assistant Professor
Department of CSE
St. Peter's Engineering College,
Hyderabad

EXTERNALEXAMINER

DEPARTMENT OF INFORMATION TECHNOLOGY

ACKNOWLEDGEMENT

We sincerely express our deep sense of gratitude to **Dr. K. Little Flower,** for her valuable guidance, encouragement and cooperation during all phases of the project.

We are greatly indebted to our Project Coordinator **Mr. A. Senthil Murugan,** for providing valuable advice, constructive suggestions and encouragement without whom it would not been possible to complete this project.

It is a great opportunity to render our sincere thanks to Head of the Department, Computer Science and Engineering for his timely guidance and highly interactive attitude which helped us a lot in successful execution of the Project.

We are extremely thankful to our Principal **Dr. Nagu Chandra Sekhar Reddy,** who stood as an inspiration behind this project and heartfelt for his endorsement and valuable suggestions.

We respect and thank our Administrative Director **Mr. T. Anuraag Reddy**, Academic Director **Mrs. T. Saroja Reddy** and secretary **Sri. T. V. Reddy,** for providing us an opportunity to do the project work at **St. PETER'S ENGINEERING COLLEGE** and we are extremely thankful to them for providing such a nice support and guidance which made us to complete the project.

We also acknowledge with a deep sense of reverence, our gratitude towards our parents, who have always supported us morally as well as economically. We also express gratitude to all our friends' who have directly or indirectly helped us to complete this project work. We hope that we can build up on the experience and knowledge that we have gained and make a valuable contribution towards the growth of the society in coming future.

BELLAMKONDA RAGHU (22BK5A1203)

DEPARTMENT OF INFORMATION TECHNOLOGY

INSTITUTE VISION

To be a renowned Educational Institution that moulds Students into Skilled Professionals fostering Technological Development, Research and Entrepreneurship meeting the societal needs.

INSTITUTE MISSION

IM1: Making students knowledgeable in the field of core and applied areas of Engineering to innovate Technological solutions to the problems in the Society.

IM2: Training the Students to impart the skills in cutting edge technologies, with the help of relevant stake holders.

IM3: Fostering a conducive ambience that inculcates research attitude, identifying promising fields for entrepreneurship with ethical, moral and social responsibilities.

DEPARTMENT OF INFORMATION TECHNOLOGY

DEPARTMENT VISION

To be a global leader in Computer Science Education, Research and Development, Innovation through industry collaboration and cultivating future-ready technologists.

DEPARTMENT MISSION

DM1: Cultivate experiential learning through state-of-the-art technologies and algorithms to address real-world interdisciplinary challenges.

DM2: Collaborate with industry and academia to equip students with cutting-edge skills and prepare them for dynamic careers.

DM3: Strengthen theoretical and applied research in Computer Science & Engineering, fostering innovation, impactful contributions, and instilling ethical values and social responsibility in all endeavors.

DEPARTMENT OF INFORMATION TECHNOLOGY

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO1: Innovate and contribute to research, driving technological advancements in industry and society.

PEO2: Excel as leaders in software development, ensuring industry and government excellence.

PEO3: Lead entrepreneurial ventures, fostering growth in computer science-driven technologies at national and global levels.

DEPARTMENT OF INFORMATION TECHNOLOGY

PROGRAM OUTCOMES (POs)

Engineering Graduates will be able to:

PO1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9. Individual and teamwork: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12. Life-long learning: Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

DEPARTMENT OF INFORMATION TECHNOLOGY

PROGRAM SPECIFIC OBJECTIVES (PSO'S)

PSO-1: Design and develop computing subsystems for data storage, communication, information processing, and knowledge discovery.

PSO-2: Design algorithms for real-world problems, focusing on execution and complexity analysis while considering security, cost, quality, and privacy parameters in software development.

DEPARTMENT OF INFORMATION TECHNOLOGY

DECLARATION

We declare that a Major Project entitled **"AI POWERED HEALTH CONNECT KIOSK"** is an Original Work submitted by the following group members who have actively contributed and submitted in partial fulfillment for the award of degree in **"Bachelor of Technology in Information Technology",** at **St. Peter's Engineering College**, Hyderabad, and this project work has not been submitted by me to any other college or university for the award of any kind of degree.

Group No: 04

Program: B. Tech

Branch: IT

Major Project Title: AI Powered Health Connect Kiosk

Date Submitted:

Name	Roll Number	Signature
BELLAMKONDA RAGHU	**22BK5A1203**	

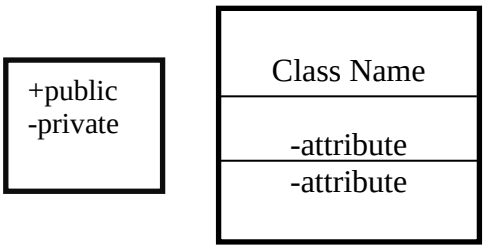
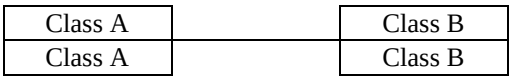
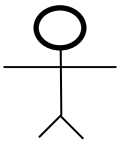
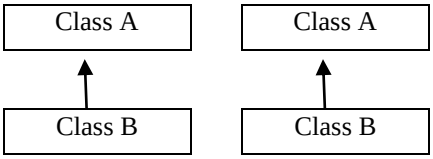
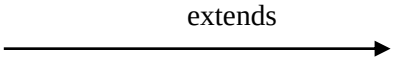
ABSTRACT

Healthcare in rural India is a serious challenge, which requires innovative yet sustainable solutions. One of the most effective solutions to this issue is the integration of AI-powered telemedicine kiosks in villages. Such kiosks, strategically placed for easy access, have the potential to revolutionize healthcare delivery by providing instant access to expert medical guidance. The process starts with a robotic system interacting with patients, gathering details about their symptoms and medical history. Based on this initial screening, the system connects the individual with specialists online for a comprehensive consultation. After the diagnosis, the prescribed medications and required services are promptly delivered by local Asha workers, ensuring timely treatment. This approach effectively integrates advanced technology with existing healthcare networks, addressing the shortage of medical professionals in rural regions while improving the efficiency and accessibility of healthcare services.

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

S.No.	NOTATION NAME	NOTATION	DESCRIPTION
1.	Class		Represents a collection of similar entities grouped together.
2.	Association		Association represents static relationships between classes. Role represents the way the two classes see each other.
3.	Actor		It aggregates several classes into a single class.
4.	Aggregation		Interaction between the system and external environment.
5.	Relation (extends)		Extends relationship is used when one use case is similar to another use case but does a bit more.

6.	Communication		Communication between various use cases.
7.	State		State of the processes.
8.	Initial State		Initial state of the object
9.	Final state		Final state of the object
10.	Control flow		Represents various control flow between the states

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

This project presents an innovative AI-assisted telemedicine robotic kiosk specifically aimed at rural Indian healthcare problems. Strategically positioned in villages, these kiosks allow people to connect with expert's doctors via the e-sanjeevani App, using biometric identification for secure and personalized medical interactions. For prescriptions after the consultation, local Asha workers help deliver prescribed medication and essential services, bridging the gap that healthcare shortcomings bring into isolation the distant communities.

A highlight of this system is an AI-based chatbot for disease prediction, which it achieves through state-of-the-art NLP techniques and machine learning algorithms such as KNN and Decision Tree in order to avoid false diagnosis. WordNet and tokenization increase the system's ability to read and respond according to the input of the users. The kiosks also include Optical Character Recognition (OCR) technology, which is powered by Tesseract, to extract information from scanned pathology reports, thereby allowing for easy translation and graphical analysis. By combining cutting-edge AI with existing healthcare infrastructure, this solution is expected to revolutionize healthcare delivery, bringing quality medical services to underserved rural areas.

1.1 PROBLEM STATEMENT

Healthcare accessibility in rural India is heavily constrained by the scarcity of medical professionals and insufficient infrastructure, thus causing delayed diagnosis and poor health outcomes. In order to tackle these issues, the project presents an AI-based telemedicine robotic kiosk placed strategically in villages for seamless access to expert consultations through the e-sanjeevani App. It is supported by local Asha workers to ensure that medicines are delivered on time and, thus, helps bridge the gap in healthcare for the people living in the rural areas.

1.2 OBJECTIVES

The major objectives of this AI-assisted telemedicine kiosk project are:

- To ensure accessible healthcare services for the rural Indian population with a safe, user-friendly medical consultation platform.
- Ensure timely delivery of medicines and essential services with the help of local Asha workers.

- To utilize AI-based technologies for correct diagnosis, data management, and customized care.
- To empower communities and ensure sustainable health outcomes.

1.3 EXISTING SYSTEM

Healthcare systems that exist today have mobile applications and call-based services for consultation purposes. A "Doctor Consultation" platform bridges the gap between village doctors and hospital physicians for diagnosis and treatment through phone or video calls.

Limitations of the Existing System:

- It cannot provide scope for customized patient experience.
- Expensive hardware for patient and doctor's data communication

1.5 PROPOSED SYSTEM

This proposed system seeks to automate health care procedures while reducing manual input. The introduction of an AI-powered kiosk will enable a user to upload his symptoms, along with disease severity, which will provide that user with more tailor-made support. These features are highlighted below:

- An AI-based chatbot predicts diseases and can assist the users.
- Communication is through voice-to-text method; thus, it makes easier.
- Complete connectivity with all doctors available.

This approach simplifies healthcare delivery, ensuring timely and effective medical support for rural populations.

1.6 SCOPE

The AI-assisted telemedicine project is designed for phased implementation, starting with pilot programs in rural areas, followed by regional scaling supported by robust infrastructure. Key future advancements include:

- Regular AI updates to improve diagnostic capabilities.
- Enhanced user interfaces for seamless interaction.
- Expanded diagnostic tools and feedback systems for continuous improvement.
- Training programs for the Asha workers and health care providers will be very essential in effective implementation and sustainability.

CHAPTER 2

LITERATURE SURVEY

NAME	YEAR	AUTHOR NAME	TOOLS	ADVANTAGES	LIMITATIONS
Innovation in practice: mobile phone technology in patient care.	2008	Blake H	IOT methods are used for medical services	Accurate spatial representation, visual mapping	Dependency on GIS infrastructure, limited real-time updates
The recent progress and applications of digital technologies in healthcare: a review	2020	Senbekov M, Saliev T, Bukeyeva Z, Almabayeva A, Zhanaliyeva M, Aitenova N, Toishibekov Y, Fakhradiyev I	Digital technologies in healthcare	Improved accessibility, enhanced data management	Implementation challenges, data security concerns.
AutoImpilo: smart automated health machine using IoT to improve telemedicine and telehealth	2021	Ganesh D, Seshadri G, Sokkanarayanan S, Bose P, Rajan S, Sathiyanarayanan M	IoT-based smart health machine	Real-time monitoring, automation in telehealth	High initial cost, integration complexities
Ensuring patient and public involvement in the transition to AI-assisted mental health care: A systematic scoping review and agenda for design justice	2021	Zadar T, Morrow EM, Stockley R	AI-assisted mental health care	Increased patient engagement, improved mental health diagnosis	Ethical concerns, potential bias in AI models
Health management via telemedicine: Learning from the COVID-19 experience.	2021	Sun R, Blayney DW, Hernandez-Boussard T	Telemedicine platforms	Remote consultations, reduced healthcare burden	Digital divide, patient data privacy risks

Table-2.1 Literature Survey

CHAPTER 3

SYSTEM REQUIREMENTS SPECIFICATIONS

3.1 HARDWARE REQUIREMENTS

- Processor : Intel i5/AMD Ryzen 5
- Hard Disk : 500GB SSD
- RAM : 8GB

3.2 SOFTWARE REQUIREMENTS

- Operating System : Windows 7/8/10/11
- IDE : VS Code
- Technology : PYTHON, DJANGO, HTML, CSS, JAVA SCRIPT, SQLyog

3.2.1 Windows Operating System for a Development Work Station

While one can reliably expect a compatible windows operating environment under Windows 7/8/10/11, robust development performance that takes advantage of numerous drivers and allows very familiar coding interactions, they could be equipped for any need whether it might arise from its Hyper-V functionality toward virtualizations in the new role of building microservices while ensuring the software in question interacts safely with as Linux subsystem. PowerShell commands are more often used together. Regular updates and security patches ensure a secure and efficient development platform.

3.2.2 IDE: Visual Studio Code

Microsoft offers an open-source editor called Visual Studio Code, a feature-rich coding tool that contains features like debugging, syntax highlighting, intelligent code completion, and integration with Git. Its extensibility through plugins and support for numerous programming languages make it adaptable to various projects. With such a simple interface, robust performance, and active developer community, VS Code makes it an excellent choice for developers at any level of expertise.

3.2.3 Python and Django

Python is a highly flexible and easy language to use, offering clean syntax and dynamic typing that easily facilitates rapid application development. Python has high-level data structures and a smooth approach to object-oriented programming, making it the ideal language for scripting and application development on any platform.

Django is a high-level Python web framework for rapid development and pragmatic design. It follows the Model-View-Controller (MVC) pattern and provides an Object-Relational Mapping (ORM) system for database interactions. With features like a robust admin interface, built-in security measures, and scalability, Django is well-suited for projects of any size. Its extensive documentation and large community simplify web development, providing powerful tools and resources.

3.2.4 SQLyog

SQLyog is a robust MySQL database management tool with an intuitive graphical interface. It streamlines tasks like database creation, management, and optimization. Advanced features such as data synchronization, scheduled backups, and query profiling enhance its utility for both novice and experienced developers. SQLyog is an efficient solution for managing MySQL databases, supported by extensive documentation and an active community.

3.2.5 Libraries Used

The project integrates several libraries to ensure functionality and efficiency:

Matplotlib	: For data visualization.
MySQL-connector, MySQL client, PyMySQL	: To manage database interactions.
Scikit-learn	: For machine learning and data analysis.
XLrd and Pandas	: For data manipulation and handling Excel files.
Speech Recognition and PyAudio	: To enable audio processing and speech recognition.

These libraries collectively provide a robust framework for developing advanced and feature-rich applications.

CHAPTER 4

SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

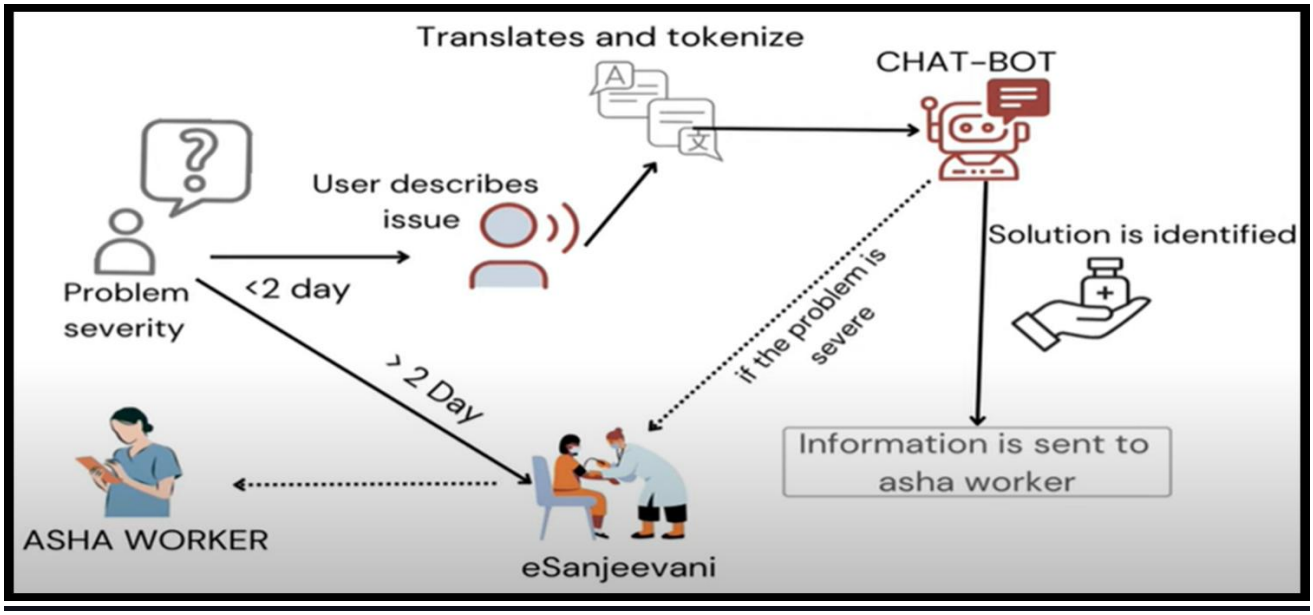


Fig-4.1: SYSTEM ARCHITECTURE

4.1.1 Architecture Description

The architecture of the project will include three primary dashboards-User, Admin, and Doctor, each with its responsibility:

User Module:

Users interact with the system either through a chatbot or by directly consulting a doctor,

depending on the severity of symptoms (it is either severe or mild, depending upon the fact whether it will take more than one day or not).

Admin Module:

Administrators will handle the dataset. They will choose the most accurate algorithm that predicts medicines through the chatbot.

Doctor Module:

Doctors can check the user's requests, chat consultation, and suggest the correct medicines according to the symptoms described.

4.2 UML Diagrams

- UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.
- The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: A Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.
- The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.
- The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.
- The UML is a very important part of developing objects-oriented software. The UML uses mostly graphical notations to express the design of software projects.

4.2.1 Use Case Diagram

- A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.

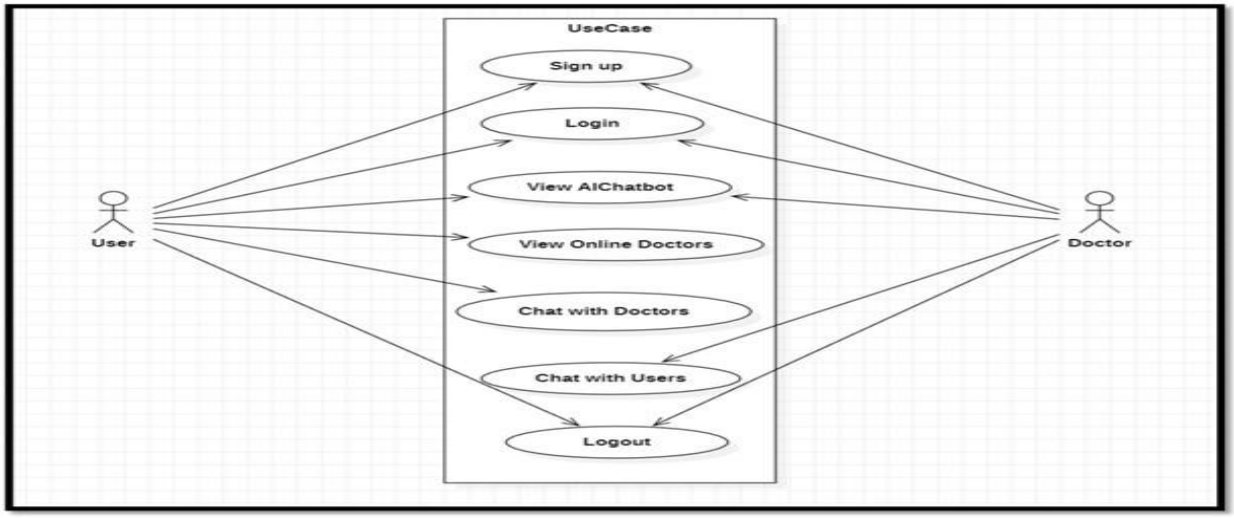


Fig-4.2.1: USE CASE DIAGRAM

4.2.2 Class Diagram

- In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains which information.

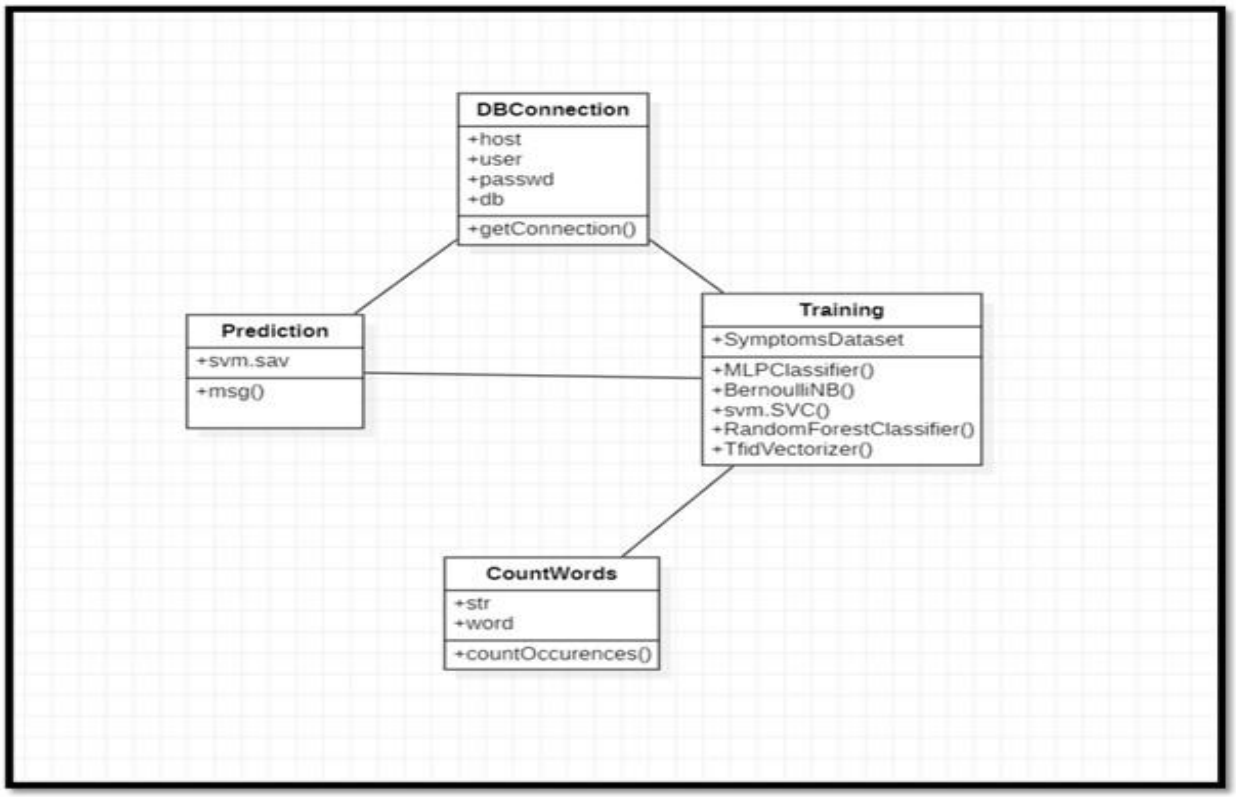


Fig-4.2.2: CLASS DIAGRAM

4.2.3 Activity Diagram

- Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

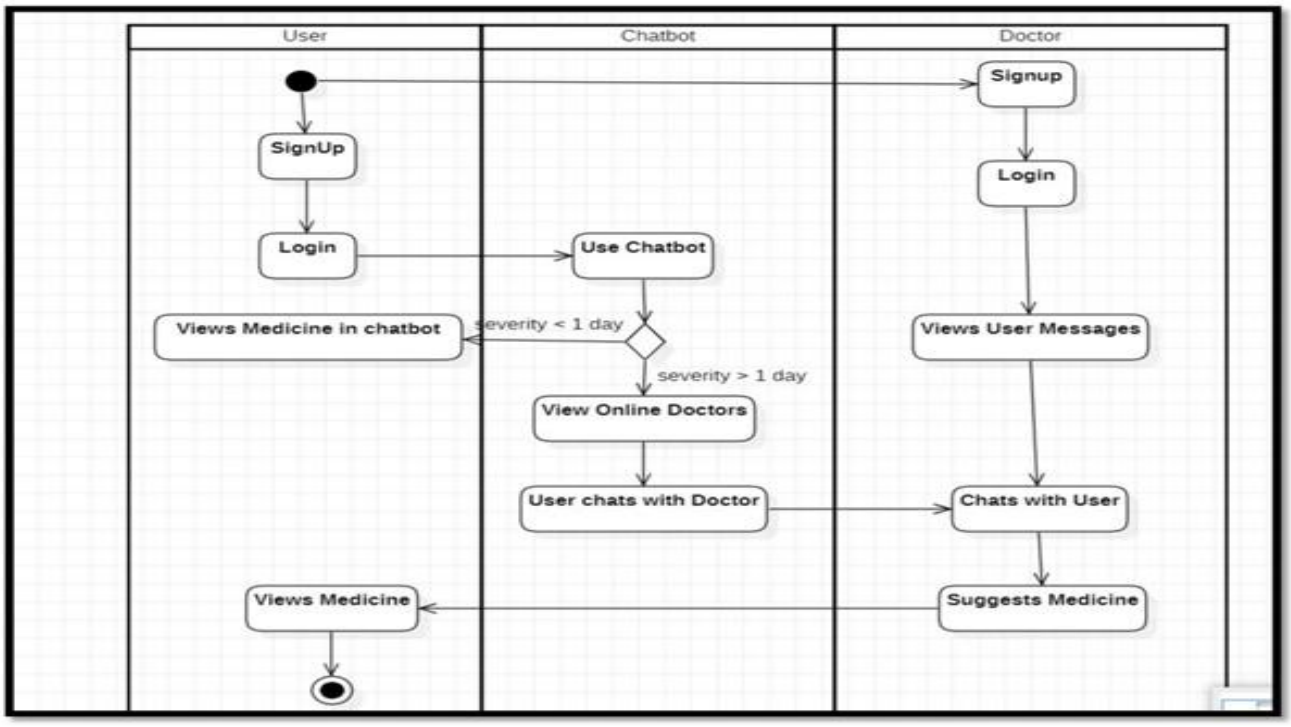


Fig-4.2.3: ACTIVITY DIAGRAM

4.2.4 SEQUENCE DIAGRAM

- Activity A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how process operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

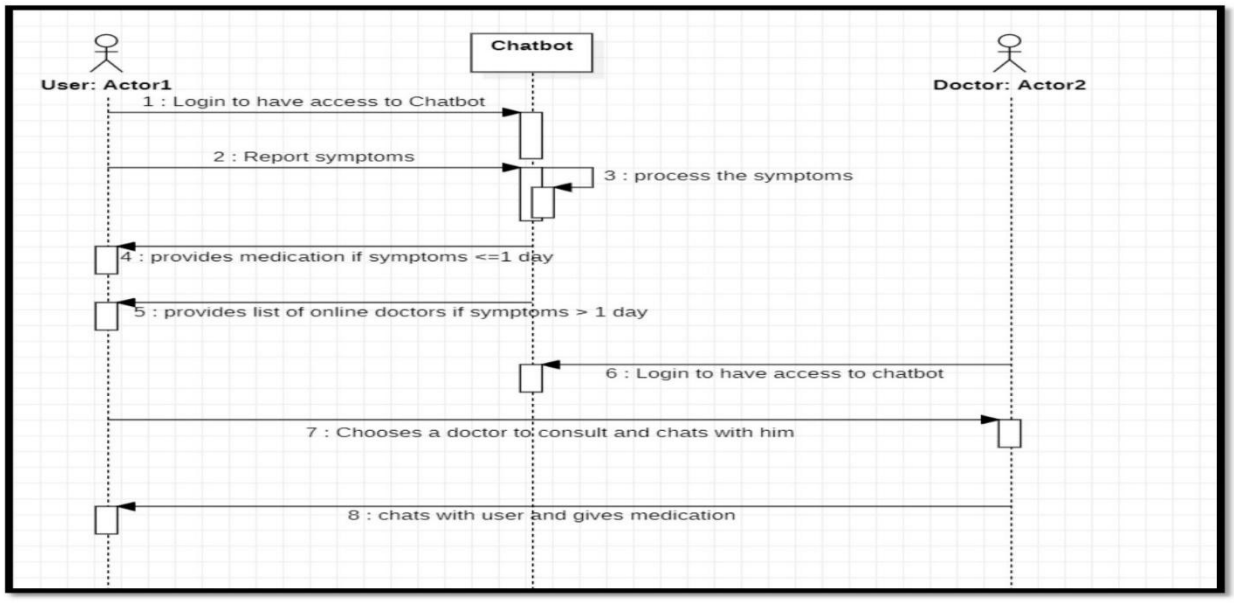


Fig-4.2.4: SEQUENCE DIAGRAM

4.2.5 DATA FLOW DIAGRAM

- A Data Flow Diagram (DFD) is a graphical representation of the flow of data through a system. It illustrates how data is processed by a system in terms of inputs and outputs. The DFD helps to understand the functionality of a system and the date flow

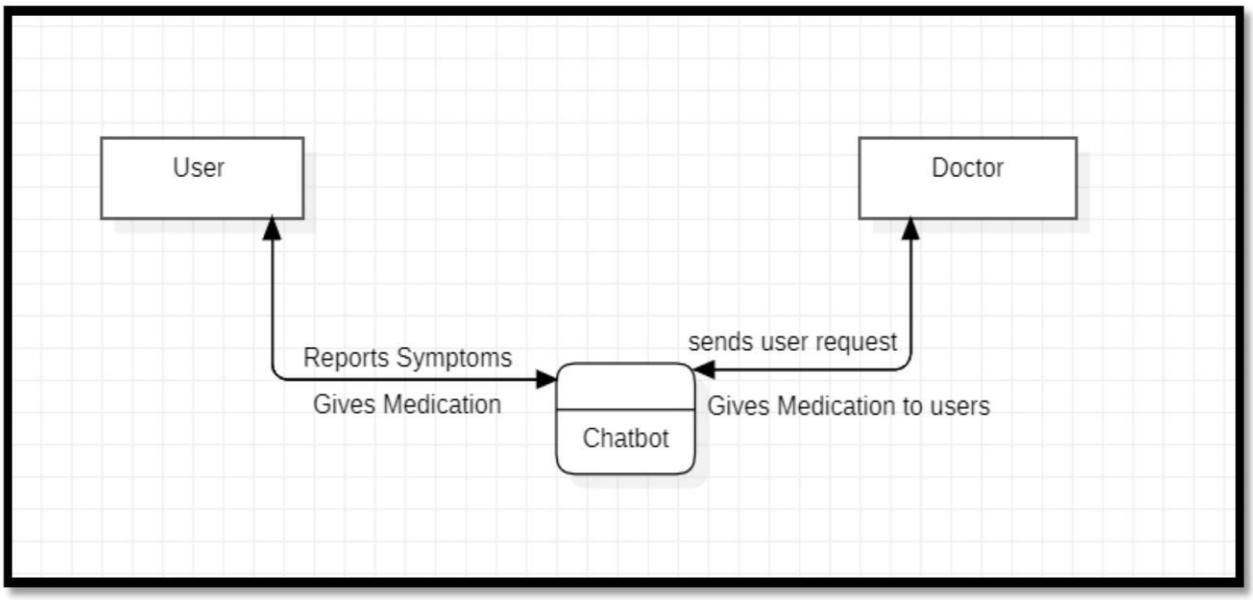


Fig-4.2.5: Level-0 DATA FLOW DIAGRAM

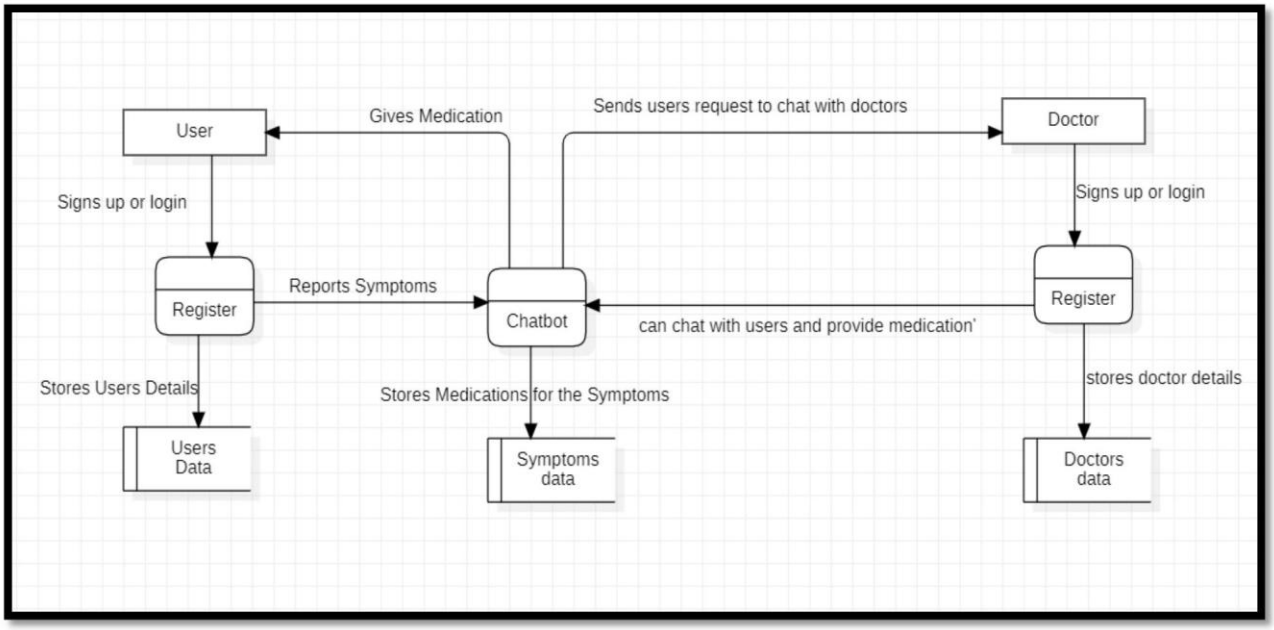


Fig-4.2.6: Level-1 DATA FLOW DIAGRAM

CHAPTER 5

METHODOLOGY

WORKING METHODOLOGY

Building an AI-supported kiosk in rural areas involves a multidisciplinary approach: integrating advanced technology with a deeper understanding of the local needs. The system provides accessible, reliable, and efficient health care that suits the rural communities. A step-by-step detailed working methodology is provided below.

USER REGISTRATION AND AUTHENTICATION

User Registration:

- Users sign up using the most important information including name, email, and password.
- Secure mechanisms, such as email verification and password hashing, ensure data protection.

Authentication:

- Upon registration, users log in using their credentials.
- Authentication tokens (e.g., JWT) maintain secure and persistent sessions, allowing seamless access without repeated logins.
- This approach ensures that only authenticated users can access their accounts and data.

INTERACTION FLOW

User Interaction:

- The user accesses the kiosk and selects a service from the main menu.
- For a telemedicine session with a doctor, symptoms are either input or selected.

Doctor Interaction:

- Doctors log into the system in a secure manner to view the health records of users and symptoms.
- They have access to diagnostic tools, update records, and prescribe medications accordingly.

Administrator Interaction:

- Administrators track the status of the system from the admin dashboard.
- Their roles involve managing user accounts, resolving issues, and providing robust data security.

DEVELOPMENT PHASES

Requirements Analysis:

- It is the phase where the purpose, functionalities, and objectives of the system are understood.
- Define input/output specifications and state clearly what the final product is supposed to be.

System Design:

- Translate the requirements into a system design. Identify hardware/software requirements and architecture definition.
- Create a detailed design blueprint to be followed in the development process.

Implementation:

- Develop the system in modular units.
- Each unit goes through exhaustive Unit Testing for the functionality of that unit.

Integration and Testing:

- Integrate all the units to make a cohesive system.
- Rigorous software testing to detect errors and correct it so that during installation, everything works smoothly.

System Deployment

- After successful functional and non-functional testing, deploy the system in the user environment or release it to the market.

Maintenance:

- Provide support and updates post-deployment.
- Customer change requests are addressed, and issues identified during live usage are resolved.
- The system is kept efficient and up-to-date.

CHAPTER 6

IMPLEMENTATION

6.1 ENVIRONMENTAL SETUP

6.1.1 Installing Visual Studio Code:

1. Visit the official website of Visual Studio (https://visualstudio.microsoft.com/) and navigate to the "Downloads" section.

2. Press the "**Download for Windows**" button on the website to start the download of the Visual Studio Code Application.

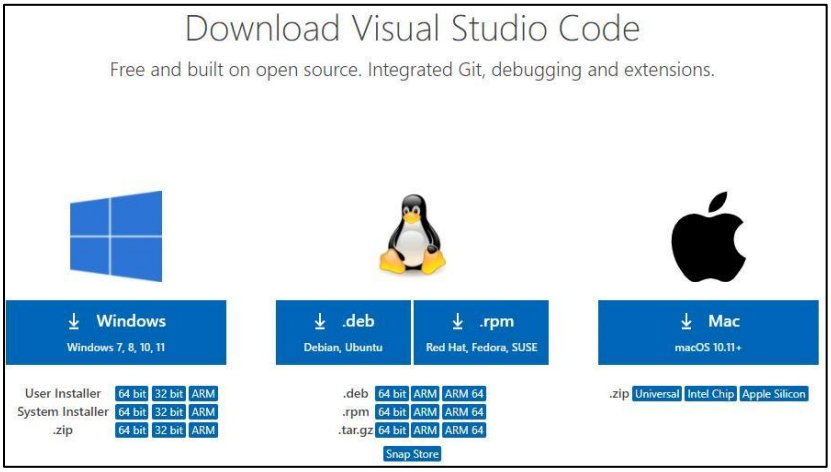


FIG-6.1 DOWNLOAD VS CODE

3. It will ask to begin the installation setup. Click on the **Install** button.

4. After the Installation setup for Visual Studio Code is finished, it will show a window. Tick the "**Launch Visual Studio Code**" checkbox and then click **Next**.

5. The **Visual Studio Code window** opens successfully. Now you can create a new file in the Visual Studio Code window and choose a language of yours.

6.1.2. INSTALLING AND IMPORTING SQLYOG:

1. The **Visual Studio Code window** opens successfully. Now you can create a new file in the Visual Studio Code window and choose a language of yours.

2. Download the installer for SQLyog Community Edition (free) or the trial version of the Professional Edition if you prefer.

3. Click Install and wait for the process to complete.

4. You can now find SQLyog in your Start Menu or desktop shortcuts. Launch it to start using SQLyog.

Fig-6.2 Download MySQL

6.2 MODULE DESCRIPTION

The AI-assisted telemedicine kiosk system is a multi-module application that is integrated to perform the specific tasks smoothly. These modules work together in order to deliver efficient healthcare services.

USER MODULE

- The User Module is central to the application, enabling people to use the kiosk for healthcare services.
- Users input symptoms to receive automated solutions.
- If the symptom is less than 2, the chatbot module will be active in giving pre-trained data answers.
- For cases more than this, the module leads the users to a consultation module with the doctor.

CHATBOT MODULE

- This module makes use of Natural Language Processing (NLP) techniques for delivering automated answers to user questions.

- Users will give input via text or voice and process through NLP tools like tokenization and WordNet.
- It has pre-trained on health-related data, so it can suggest the medicines based on the symptoms.
- Answers are accurate and cater to the user's needs in order to ensure a smooth experience.

VOICE TRANSLATION MODULE

- This module increases user access by converting the user's voice into text.
- The Google Speech-to-Text API is used for efficient voice processing.
- The translated text is fed into the chatbot module where symptom analysis and solution generation will be performed.
- This is a user-friendly feature of the kiosk, especially to those who are less conversant with typing.

DOCTOR MODULE

- The Doctor module equips medical practitioners with the tools and strategies necessary to managing patient interactions appropriately.
- Doctors can create and manage their profile including information on qualifications and specializations.
- Doctors look through patients' health questions and respond through the kiosk platform.
- Depending on symptoms, doctors prescribe drugs and give additional information to them.

ADMIN MODULE

- The admin module oversees the whole system's running and performance.""".
- Monitors user activity and ensures smooth functioning of the kiosk.
- Analyzes usage data, health trends, and performance metrics to optimize the system.
- Admins ensure data security and manage user accounts, maintaining a robust and reliable system.

6.3 ALGORITHMS

6.3.1 Support Vector Machine (SVM)

As pointed out by Limei et al. (2014), SVM is a supervised learning algorithm that makes an effort to find the hyperplane with maximum distance between classes so that the minimum number of errors is made while training. Efficient with small sets of data, the SVM algorithm provides strong capabilities in machine learning. This application used the Support Vector Classifier of SVM for conducting classification analysis in the medical chatbot.

6.3.5 Natural Language Processing (NLP)

Natural Language Processing (NLP) lets machines understand and manipulate human language. NLP helps a lot in processing text and speech data because of the complexities associated with dialects, slang, and grammatical differences in everyday communication. NLP can also add value to applications that are directly customer-facing like chatbots; it can be used to process and classify requests, provide responses to frequently asked questions, and forward complex queries to customer care. This automation reduces operational costs, saves time by eliminating repetitive tasks, and improves user satisfaction

6.4 SOURCE CODE

Import Packages

```
import numpy as np
import pandas as pd
#For Plotting
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import sys
import sqlite3
#For Prediction
import warnings
import sys
import pickle
# for Speech Recognition
import speech_recognition as sr
from time import sleep
import pyaudio as pyaudio
import wave from os
import path
```

Setting Up Environment

```
import warnings
import sys
import warnings30
if not sys.warnoptions:
warnings.simplefilter("ignore")
import pickle
class Prediction:
def do(msg):
msg=[msg]
filename = 'svm.sav'
train = pickle.load(open(filename, 'rb'))
predicted_class = train.predict(msg)
return predicted_class[0]
if __name__ == "__main__":
print(Prediction.do('Fever, Cough'))
```

Setting Up Model

```
from django.db import models
# Create your models here.
class users(models.Model):
name=models.CharField(max_length=159);
email=models.CharField(max_length=159);
pass_word=models.CharField(max_length=159);
phone=models.CharField(max_length=159);
city=models.CharField(max_length=159);
gender=models.CharField(max_length=159);
```

```
age=models.CharField(max_length=159);
class queries(models.Model):
q_n=models.CharField(max_length=1000);
an_s=models.CharField(max_length=1000);
class dataset(models.Model):
Symptoms=models.CharField(max_length=500);
Causes=models.CharField(max_length=500);
Disease=models.CharField(max_length=500);
Medicine=models.CharField(max_length=500);
Specialist=models.CharField(max_length=500);
class chat(models.Model):31
name=models.CharField(max_length=100);
email=models.CharField(max_length=100);
message=models.CharField(max_length=5000);
class docchat(models.Model):
name=models.CharField(max_length=100);
email=models.CharField(max_length=100);
message=models.TextField();
chatbw = models.CharField(max_length=100);
stz = models.CharField(max_length=10);
class doctors(models.Model):
Specialist=models.CharField(max_length=1000);
email=models.CharField(max_length=100);
name=models.CharField(max_length=100);
qualification=models.CharField(max_length=100);
contact=models.CharField(max_length=100);
password=models.CharField(max_length=100);
stz=models.CharField(max_length=100);
class performance(models.Model):
alg_name = models.CharField(max_length=100)
sc1 = models.FloatField()
sc2 = models.FloatField()
sc3 = models.FloatField()
sc4 = models.FloatField()
```

Testing

```
import time
import pickle
import pandas as pd
from sklearn.metrics
import accuracy_score, precision_score, recall_score, f1_score
def Testing( model, file):
test_ = pd.read_csv(file)
test_=test_.dropna()
Y=test_['Disease']
p_model = pickle.load(open(model, 'rb'))
predicted_class = p_model.predict(test_['Symptoms'])
accuracy = model_assessment(Y, predicted_class)
```

```
return accuracy
def model_assessment(y_test, predicted_class):
accuracy = accuracy_score(y_test, predicted_class)
accuracy = round(accuracy, 4)
precision=(precision_score(y_test, 32 predicted_class, average='micro', pos_label='Allergy'))
precision = round(precision, 4)
recall=(recall_score(y_test, predicted_class, average='micro',pos_label='Allergy'))
recall = round(recall, 4)
fscore=(f1_score(y_test, predicted_class, average='micro',pos_label='Allergy'))
fscore = round(fscore, 4)
return(accuracy,precision,recall,fscore)
if __name__ == '__main__':
print(Testing('nb.sav','Testing.csv'))
```

Testing

```
import sys
import pandas as pd
import numpy as np
from sklearn.naive_bayes import BernoulliNB
from sklearn.neural_network import MLPClassifier
from sklearn import svm
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, f1_score, accuracy_score, confusion_matrix
import pickle
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.pipeline import Pipeline class Training:
def train(algo):
alg=None
if algo=='nb':
alg=BernoulliNB()
if algo=='nn':
alg=MLPClassifier()
if algo=='svm':
alg=svm.SVC()
if algo=='rf':
alg=RandomForestClassifier()
train_file="medical data.csv"
df = pd.read_csv(train_file)
df=df.drop_duplicates()
df=df.dropna()
tfidf = TfidfVectorizer(stop_words='english', use_idf=True, smooth_idf=True)
pipeline = Pipeline([('lrgTF_IDF', tfidf), ('lrg_mn', alg)])
filename = str(algo)+'.sav' pickle.dump(pipeline.fit(df['Symptoms'], df['Disease']), open(filename, 'wb'))
if __name__ == "__main__":
Training.train('nn')
Training.train('svm')
Training.train('nb')
Training.train('rf')=bannerUrl;
```

CHAPTER 7

TESTS AND RESULTS

7.1 INTRODUCTION

System testing is a critical phase in the software development lifecycle, occurring after the development process is complete. It involves evaluating the entire system to ensure it meets user requirements and functions as intended. This phase is essential for identifying defects and verifying the system's reliability, functionality, and performance under various conditions. A well-structured testing process not only guarantees quality but also prevents costly errors post-deployment.

7.2 TESTING OBJECTIVES

The main goals of testing are summarized below:

- Error Detection: Run the program to identify errors that could not be identified during development.
- Good Test Cases: Develop test cases that maximize the probability of finding undiscovered errors.
- Validation: Confirm that the software satisfies all the requirements specified in the specifications.
- Verification: Verify that the software behaves as expected in real application environments.

7.3 TESTING PROCEDURES

System testing of the software checks about its functional, performance as well as reliability features. Key procedures include all of the following:

- Server Connectivity Testing-Verify the server names, machine connections, and communication between clients and executives.
- Data Validation Testing-ensure that product information provided by the company is in accordance with data existing in the centralized database
- Availability Testing-verify the availability of executives to connect with the server.
- Functionality Testing: Test the chat lines and email functionalities to ensure smooth user interactions.
- Concurrency Testing: Check the system's ability to handle multiple concurrent users without performance degradation.

7.3.1 SOURCE CODE TESTING

Purpose:

- Check the logic and flow of the program by verifying sample data updates against predefined criteria.

Procedure:

- Review the program's logic for consistency and correctness.
- Test sample files and directories for accurate data handling and processing.

7.3.2 SPECIFICATION TESTING

Purpose:

- Verify that the system operates as defined for all conditions.

Process:

- Test the system against the predefined specifications.
- Check the behaviour of the system on different scenarios and combinations of conditions, for all modules.

7.3.3 UNIT TESTING

Objective:

- To check individual modules independently, with the assumption that each and every module works correctly.

Process:

- Testing of each individual module is called module testing.
- During the coding phase, verify the expected outcome for each module.
- Integrate validation checks for input from users and data consistency.

Testing Techniques:

Black Box Testing:

- Focuses on the input-output behaviour without knowledge of the internal code structure.
- Tests the software against its requirements and specifications.
  Key Features:
  - Focuses on user views.
  - No prior knowledge of the codebase.
  - Validates functional requirements.

White Box Testing:

- Focuses on the internal logic and structure of the program.
- All paths, loops, and data flows in the code must be working.
  Key Features:
  - Knowledge of the internal code structure is required.

- Identifies possible logical errors and optimization opportunities.

Fig 7.1 Blackbox Testing

BLACK BOX TESTING

Definition:

Black Box Testing checks the software functionality without testing its internal structure or operation. The tester inputs some data and measures the outputs with no knowledge of the internal implementation of the code.

Applications:

- Black Box Testing can be used on any kind of software system, such as:
- Operating systems (Windows, Linux)
- Websites (Google, Amazon)
- Databases (Oracle, MySQL)
- Custom in-house developed applications

By considering the inputs and outputs, Black Box Testing guarantees that the software complies with user requirements without looking at the internal working mechanism.

WHITE BOX TESTING

Definition:

White Box Testing involves testing the inner code, structure, and working logic of the software. Its main objective is to enhance security, design, usability, and proper flow of inputs and outputs within the application.

Characteristics:

- Source code is needed to access.
- Internal paths, logic, and data flow.

Usually undertaken by developers or by testers who understand the architecture of the system.

Why "White Box"?

The term "White Box" represents clarity, wherein the tester can "look into" the internal workings of the application. This is in contrast to "Black Box" testing, wherein the tester views no internal architecture.

SYSTEM TESTING

Summary:

After the individual modules are tested (Unit Testing), the next step is to assemble and integrate them into a cohesive system. System Testing evaluates whether the entire system performs as expected under various conditions.

Types of System Testing:

Alpha Testing:

- Conducted by the organization's internal test team.
- Simulates real-world usage in a controlled environment.

Beta Testing:

- Performed by a selected group of end-users (friendly customers).
- Identifies real-world issues before the final release.

Acceptance Testing:

- Conducted by the customer.
- Determines if the system meets their needs and whether they should accept it.

INTEGRATION TESTING

Goal:

- Confirm that individual modules function properly when combined into the system.

Key Focus Areas:

- Data transfer between module interfaces.
- One module's effect on another.
- Overall performance of sub-functions providing the intended major function.

Procedure:

- Systematic testing with sample data.
- Validate the overall performance with smooth inter-module interactions.

OUTPUT TESTING

Goal:

The system's output format and result is verified with respect to user expectation.

Procedure:

- The output produced is reviewed with the user in terms of required format.
- Performance validation for accuracy and efficiency in producing output data.

Test Case ID #1		**Test Case Description** – Validation in Registration Form		
S#	**Prerequisites**	**S#**	**Test Data Requirement**	
1	User should be Registered	1	Data should be valid	
Test Condition				
Entering data in registration form				
Step#	**Step Details**	**Expected Results**	**Actual Results**	**Pass/Fail/Not Executed/Suspended**
1	User gives First and Last Name	Pop showing email verification message	Enter valid email/password	Fail
2	Submitting the form without entering any details	Pop showing email verification message	Enter email/password	Fail
3	User enters invalid format of email id	Pop showing email verification message	Enter valid email id	Fail
4	User enters a phone number with < 10 digits	Pop showing email verification message	Enter valid phone number	Fail
5	Entering valid username and password	Pop showing email verification message	Pop showing email verification message	Pass

Table 7.1 Registration test case

Test Case ID #2		**Test Case Description** – Validation in Login Form		
S#	**Prerequisites**	**S#**	**Test Data Requirement**	
1	User should an email id	1	Data should be valid	
Test Condition				
Entering data in login form				
Step#	**Step Details**	**Expected Results**	**Actual Results**	**Pass/Fail/Not Executed/Suspended**
1	User gives an email or password of <6characters	User logged in	Enter valid email/password	Fail
2	Submitting the form without entering any details	User logged in	Enter email/password	Fail
3	User enters wrong email (or) password	User logged in	Enter correct email/password	Fail
4	User enters a phone number with < 10 digits	Pop showing email verification message	Enter valid phone number	Fail
5	Entering valid username and password	Pop showing email verification message	Pop showing email verification message	Pass

Table 7.2 Login test case

RESULTS [SCREENSHOTS]

FIG – 7.2: HOME PAGE

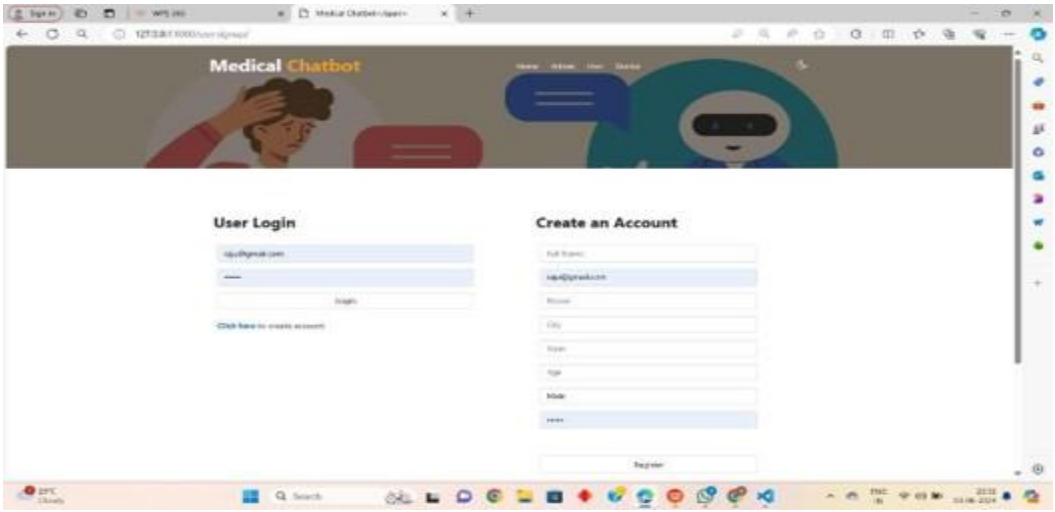


FIG – 7.3: USER LOGIN PAGE

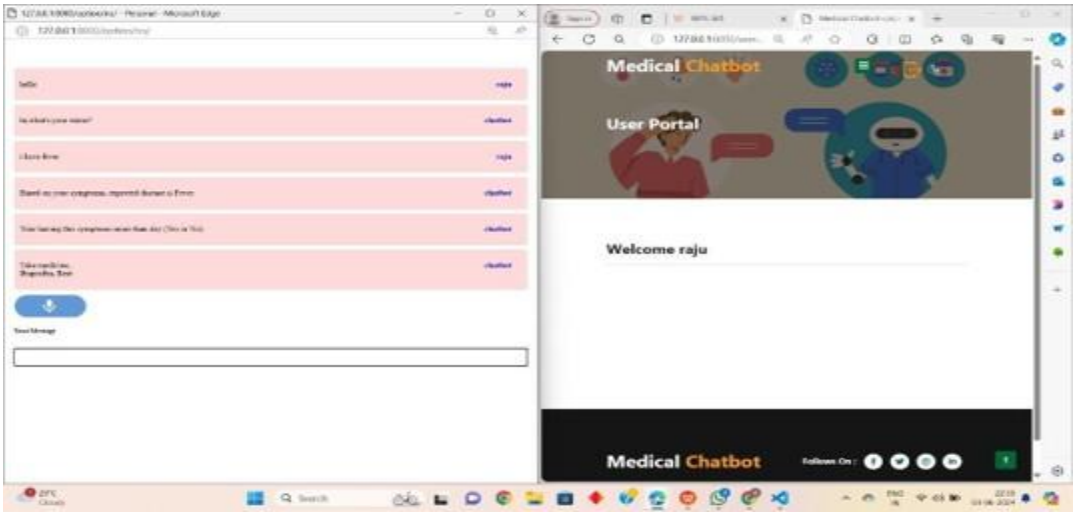


FIG -7.4: USER INTERACTION WITH CHATBOT

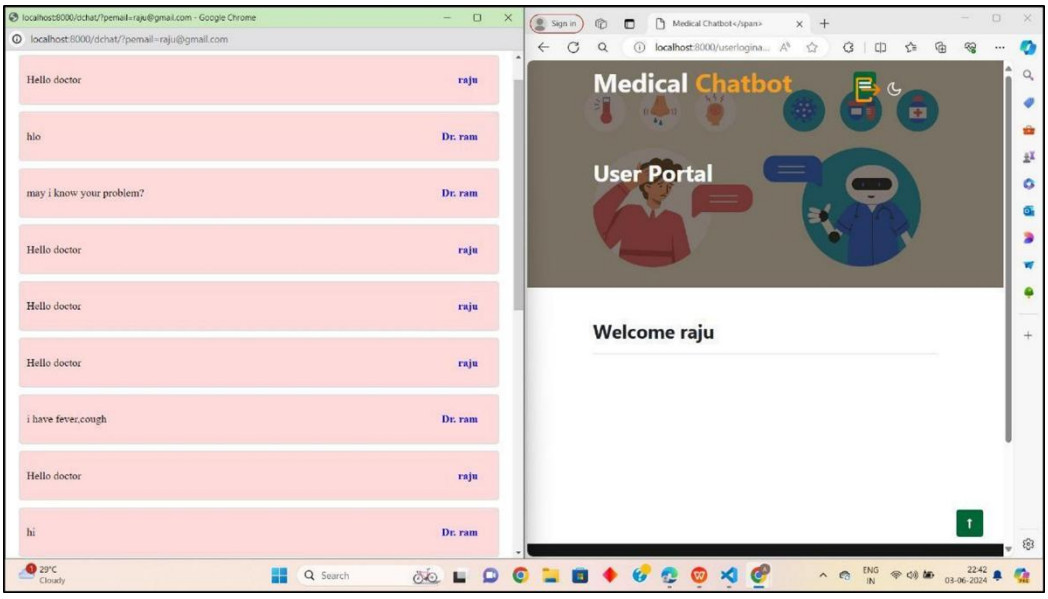


FIG -7.5: USER INTERACTION WITH DOCTOR

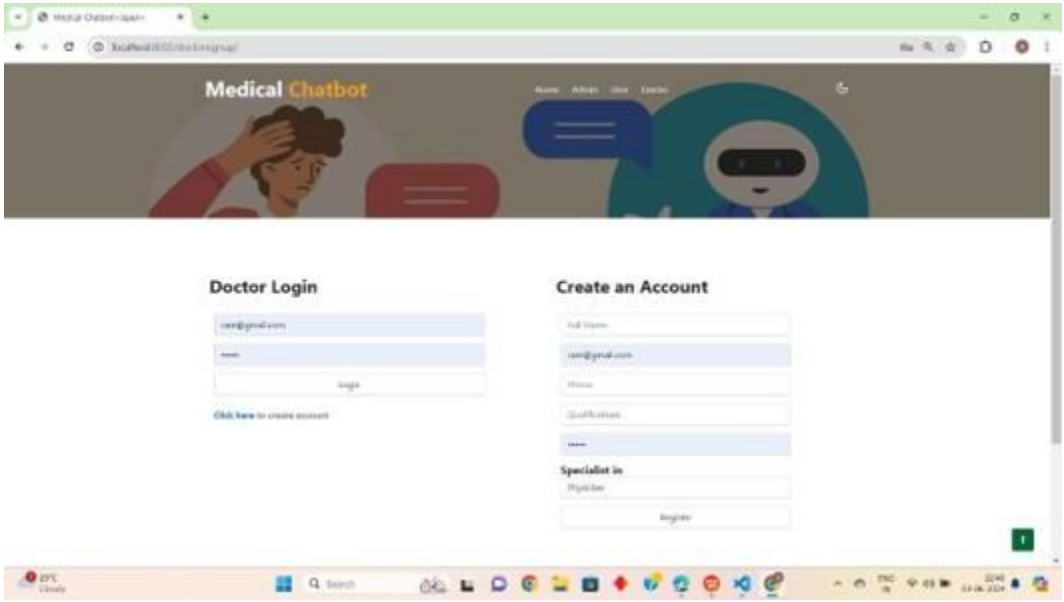


FIG -7.6: DOCTOR LOGIN PAGE

CHAPTER 8

CONCLUSION

The telemedicine robotic Kiosk project, aided by artificial intelligence, shall transform the way healthcare is delivered in India's rural regions. By installing the kiosks at villages, we directly connect villagers with medical experts through the e-Sanjeevani application without the constraints of distance or a scarcity of healthcare facilities. The prescribed drugs and follow-up services are handled by local Asha workers to connect gaps in accessibility with healthcare.

The continuous data collection and analysis of the Kiosk will help evolve with the particular health needs of the community and give insight to guide public health initiatives. It is not only enhancing access to healthcare but is also a precedent for integrating technology into rural health care systems and makes quality medical care more accessible to underserved populations.

FUTURE ENHANCEMENTS

Currently, the system uses a conversational chatbot model that is used to process both text and audio inputs, but it doesn't support media data like images or PDF files for medical reports and case studies. Future development would include the following:

- Uploading the capacity of the chatbot to handle image and PDF data for processing and analysis
- Including AI-based image recognition in medical report interpretation, X-rays, and other diagnostic images.
- The system will enable kiosks to have multi-modal inputs to allow them to enhance the user experience, making it possible to expand more or broader medical services.
- As the ability of the chatbot to handle diverse data types is continually enhanced, then even more comprehensive and accurate healthcare to rural communities will be offered through the system.

REFERENCES

1. Blake H. Innovation in practice: mobile phone technology in patient care. British journal of community nursing. 2008 Apr;13(4):160-5. https://doi.org/10.12968/bjcn.2008.13.4.29024
2. Senbekov M, Saliev T, Bukeyeva Z, Almabayeva A, Zhanaliyeva M, Aitenova N, Toishibekov Y, Fakhradiyev I. The recent progress and applications of digital technologies in healthcare: a review. International journal of telemedicine and applications. 2020;2020(1):8830200. https://doi.org/10.1155/2020/8830200
3. Ganesh D, Seshadri G, Sokkanarayanan S, Bose P, Rajan S, Sathiyanarayanan M. AutoImpilo: smart automated health machine using IoT to improve telemedicine and telehealth. In2020 International Conference on Smart Technologies in Computing, (ICSTCEE) 2020 Oct 9 (pp. 487-493). https://doi.org/10.1109/ICSTCEE49637.2020.9277223
4. Zadar T, Morrow EM, Stockley R. Ensuring patient and public involvement in the transition to AI-assisted mental health care: A systematic scoping review and agenda for design justice. Health Expectations. 2021 Aug;24(4):1072-124. https://doi.org/10.1111/hex.13299
5. Sun R, Blayney DW, Hernandez-Boussard T. Health management via telemedicine: Learning from the COVID-19 experience. Journal of the American Medical Informatics Association. 2021 Nov 1;28(11):2536-40. https://doi.org/10.1093/jamia/ocab145