COLLEGE CODE:2109

COLLEGE NAME:LOYOLA INSTITUTE OF TECHNOLOGY

DEPARTMENT:CSE

STUDENT NM-ID:F5F38F997B7421A9E36428D5AC98D28F

ROLL NO:210923104005

DATE

TECHNOLOGY-PROJECT NAME:AUTONOMOUS VEHICLES AND ROBOTICS

SUBMITTED BY

ANGEL.Y

AARAARTHIKA DEVI M

ABARJITHRAJ A

ADHITYAN S

ABISHEK D

PHASE:5:PROJECT DEMONSTRATION &DOCUMENTATION

TITLE:AUTONOMOUS VEHICLES AND ROBOTICS

ABSTRACT

The AI-Powered Healthcare Assistant project aims to revolutionize healthcare accessibility by leveraging artificial intelligence, natural language processing, and IoT (Internet of Things) technologies. In its final phase, the system integrates advanced AI models to diagnose symptoms, real-time health data collection from IoT devices, and secure data management, while ensuring scalability and seamless integration with Enterprise Resource Planning (ERP) systems. This document provides a comprehensive report of the project's completion, covering the system demonstration, technical documentation, performance metrics, source code, and testing reports. The project is designed to handle large-scale operations with robust data security measures, providing accurate health recommendations in real-time. Screenshots, ERP diagrams, and codebase snapshots will be included for a full understanding of the system's architecture and functionality.

1.Project Demonstration

Overview:

The AI-Powered Healthcare Assistant system will be demonstrated to stakeholders, showcasing its Features, performance improvements, and functionality. This demonstration highlights the system's Real-time responses, IoT data integration, security measures, and performance scalability.

Demonstration Details:

- System Walkthrough: A live walkthrough of the platform, from user interaction to the output Of health recommendations, showcasing the chatbot's responses to user health queries.

- AI Diagnosis Accuracy: The demonstration will show how the AI model provides accurate Health suggestions based on real-time user inputs and IoT device data.

- IoT Integration: Real-time metrics like heart rate, oxygen level, and body temperature Collected from IoT devices will be displayed and analyzed.
- Performance Metrics: Response time, system scalability, and load handling under multiple Users will be highlighted to show improved system capacity.
- Security & Privacy: Encryption protocols and privacy measures will be explained and Demonstrated as the system handles user health data.

Outcome:

By the end of the demonstration, the system's ability to handle real-world scenarios, ensure data Security, and deliver health insights through IoT integration will be showcased to the stakeholders.

1.Project Documentation

Overview:

Comprehensive documentation for the AI-Powered Healthcare Assistant is provided to detail every Aspect of the project. This includes system architecture, AI model details, code explanations, and Usage guidelines for both users and administrators.

Documentation Sections:

- **System Architecture**: Diagrams illustrating the complete system, including AI algorithms, workflows, and IoT device integrations.
- **Code Documentation:** Source code and explanations for all code modules, including AI Training scripts, API integrations for IoT devices, and chatbot interactions.

- **User Guide**: A manual for end users explaining how to interact with the AI assistant and how To interpret health data and recommendations.
- **Administrator Guide**: Instructions for system maintenance, monitoring, and performance Testing procedures.
- **Testing Reports**: Detailed reports on performance metrics, load testing, and data security Evaluations.

Outcome:

All critical components of the system will be well-documented, providing a clear guide for future Development, deployment, or system scaling.

1.Feedback and Final Adjustments

Overview:

Feedback from the project demonstration will be collected from instructors, stakeholders, and a Broader group of test users. This feedback will be used to make final refinements before project Handover.

Steps:

- **Feedback Collection**: Feedback from mentors, stakeholders, and test users will be gathered Via surveys and observation during the demonstration.
- **Refinement**: Based on the feedback, any performance bottlenecks, inaccuracies in AI Diagnosis, or usability issues will be addressed.
- **Final Testing**: After adjustments, the system will undergo final testing to ensure full Functionality, usability, and scalability.

Outcome:

Final adjustments will optimize the system for a broader rollout, ensuring that it is fully ready for Real-world deployment.

1.Final Project Report Submission

Overview:

The final project report provides a comprehensive summary of all phases, key achievements, Challenges faced, and outcomes of the AI-Powered Healthcare Assistant project. This report will Include testing results, performance improvements, and future recommendations.

Report Sections:

• **Executive Summary:** A concise overview of the project, outlining its objectives and major Achievements.

• **Phase Breakdown**: A detailed breakdown of each phase, covering AI model development, Chatbot improvements, IoT integration, and data security.

• **Challenges & Solutions:** A section documenting the key challenges encountered, such as AI Misdiagnosis or security under load, and how they were resolved.

• **Outcomes**: A summary of the system's current capabilities and readiness for deployment.

Outcome:

A detailed project report will be submitted, outlining the entire journey from concept to completion.

1.Project Handover and Future Works

Overview:

The projects intro for future development.

Handover Details:

• **Next Steps**: Suggestions for future work, including scaling the system to support more users, Expanding AI capabilities, and implementing multilingual support, will be provided.

Outcome:

The AI-Powered Healthcare Assistant will be officially handed over, along with recommendations for Future enhancements and guidelines for system maintenance.

SCREENSHOT OF SOURCE CODE AND FINAL PROJECT

```python
#!/usr/bin/env python
from __future__ import print_function

import roslib
roslib.load_manifest('my_package')
import sys
import rospy
import cv2
from std_msgs.msg import String
from sensor_msgs.msg import Image
from cv_bridge import CvBridge, CvBridgeError

class image_converter:

  def __init__(self):
    self.image_pub = rospy.Publisher("image_topic_2",Image)

    self.bridge = CvBridge()
    self.image_sub = rospy.Subscriber("image_topic",Image,self.callback)

  def callback(self,data):
    try:
      cv_image = self.bridge.imgmsg_to_cv2(data, "bgr8")
    except CvBridgeError as e:
      print(e)

    (rows,cols,channels) = cv_image.shape
    if cols > 60 and rows > 60 :
      cv2.circle(cv_image, (50,50), 10, 255)

    cv2.imshow("Image window", cv_image)
    cv2.waitKey(3)

    try:
      self.image_pub.publish(self.bridge.cv2_to_imgmsg(cv_image, "bgr8"))
    except CvBridgeError as e:
      print(e)

def main(args):
  ic = image_converter()
  rospy.init_node('image_converter', anonymous=True)
  try:
    rospy.spin()
  except KeyboardInterrupt:
    print("Shutting down")
  cv2.destroyAllWindows()

if __name__ == '__main__':
    main(sys.argv)
```