Phase 5: Project Demonstration & Documentation

COLLEGE CODE:3105

COLLEGE NAME:Dhanalakshmi Srinivasan College of Engineering and Technology

DEPARTMENT:B.Tech Artificial Intelligence and Data Science

STUDENT NM-ID:d621ee964237b80b9564247d46d1a984

ROLL NO:310523243079

DATE: 12/05/2025

TECHNOLOGY-PROJECT NAME:

Healthcare Diagnosis and Treatment

SUBMITTED BY: Sibi Akshaya M

Your Name and team member names.

Sibi Akshaya M

Swedha V

Thajun bee Y

Shalini M

Pooja G

Title: Healthcare Diagnosis and Treatment

Abstract:

The Healthcare Diagnosis and Treatment project leverages artificial intelligence and data-driven technologies to provide accurate medical diagnoses and treatment recommendations. This system is designed to assist both patients and healthcare professionals by analyzing symptoms, predicting possible conditions, and suggesting treatments based on medical databases and real-time input. The project includes a diagnostic chatbot interface, a treatment recommendation engine, and a secure patient data management system. This report documents the final implementation, system performance, testing results, and future scalability potential.

Index (with page numbers)

(To be completed after compiling the full report)

1. Project Demonstration

Overview:A full demonstration of the Healthcare Diagnosis and Treatment platform will highlight its core functionalities: symptom analysis, diagnosis generation, treatment suggestion, and secure data handling.Demonstration Details:• System Walkthrough: User interaction flow from symptom input to diagnosis and treatment output.• Diagnosis Accuracy: Presentation of how the AI model interprets symptoms and predicts potential conditions.• Treatment Suggestion Module: Explanation of how treatment plans are generated using

evidence-based guidelines.• Performance Metrics: Focus on response times, throughput under concurrent users, and system availability.• Data Privacy & Security: Implementation of encryption and access control during data handling.Outcome:Stakeholders will observe how the system enhances healthcare services by offering reliable AI-supported diagnoses and treatments while ensuring data confidentiality.

2. Project Documentation

Overview:Complete technical documentation covering every layer of the system.Documentation Sections:• System Architecture: Visual diagrams of model flow, backend infrastructure, and treatment pipelines.• Code Documentation: Detailed descriptions of the AI models, training code, APIs, and frontend/backend logic.• User Guide: Instructions for patients and healthcare providers on using the platform.• Administrator Guide: Maintenance schedules, system monitoring procedures, and backup processes.• Testing Reports: Accuracy evaluation, stress/load test reports, and data integrity results.Outcome:This documentation will ensure ease of maintenance, future development, and user support.

3. Feedback and Final Adjustments

Overview:Feedback is gathered from testers and mentors and used to polish the system before deployment.Steps:• Collection of structured feedback during demos.• Refinements to diagnosis accuracy, user interface, and performance.• Final validation tests to confirm usability and functionality.Outcome:Refinements ensure the system is robust and deployment-ready.

4. Final Project Report Submission

Overview:A comprehensive final report summarizing objectives, results, and recommendations.Report Sections:• Executive Summary: Overview of project purpose and accomplishments.• Phase Breakdown: Description of major development milestones.• Challenges & Solutions: Issues faced during development and their resolutions.• Outcomes: Summary of current functionality and real-world deployment potential.Outcome:Final report serves as official documentation of the project's success and readiness.

5. Project Handover and Future Works

Overview:Guidance for future developers and maintainers.Handover Details:• Suggestions for improving AI diagnosis (e.g., deeper medical learning).• Adding multilingual interfaces and mobile support.• Integrating wearable health trackers for data input.Outcome:The project will be handed over with full documentation and recommendations for continuous development.

(Include Screenshots of source code, demo UI, and final project results)

Appendix: Python Implementation Code

```python
# Healthcare Diagnosis and Treatment - Core Python Implementation


import json

import random


# Sample symptom-disease mapping (you can expand this)

symptom_disease_db = {

    "fever": ["Flu", "Malaria", "COVID-19"],

    "cough": ["Common Cold", "Bronchitis", "COVID-19"],

    "headache": ["Migraine", "Tension Headache", "COVID-19"],

    "fatigue": ["Anemia", "Thyroid Issues", "COVID-19"]

}


# Sample treatment database

disease_treatment_db = {

    "Flu": "Rest, fluids, and antiviral drugs",

    "Malaria": "Antimalarial medication",

    "COVID-19": "Isolation, rest, and supportive care",

    "Common Cold": "Rest and over-the-counter medication",

    "Bronchitis": "Cough medicine and rest",

    "Migraine": "Pain relievers and rest",

    "Tension Headache": "Stress reduction and pain relief",

    "Anemia": "Iron supplements and diet changes",

    "Thyroid Issues": "Thyroid hormone therapy"

}


def get_diagnosis(symptoms):

    possible_diseases = {}

    for symptom in symptoms:
```

```python
        diseases = symptom_disease_db.get(symptom.lower(), [])

        for disease in diseases:

            possible_diseases[disease] = possible_diseases.get(disease, 0) + 1


    # Sort by likelihood

    sorted_diseases = sorted(possible_diseases.items(), key=lambda x: x[1], reverse=True)

    return [d[0] for d in sorted_diseases[:3]]  # Top 3 matches


def suggest_treatments(diseases):

    treatments = {}

    for disease in diseases:

        treatment = disease_treatment_db.get(disease, "No treatment found.")

        treatments[disease] = treatment

    return treatments


def main():

    print("Welcome to the Healthcare Diagnosis and Treatment System")

    symptoms = input("Enter your symptoms separated by commas: ").split(',')

    symptoms = [s.strip().lower() for s in symptoms]


    print("\nAnalyzing symptoms...")

    diseases = get_diagnosis(symptoms)

    print("\nPossible Diagnoses:")

    for i, disease in enumerate(diseases, 1):

        print(f"{i}. {disease}")


    treatments = suggest_treatments(diseases)

    print("\nSuggested Treatments:")

    for disease, treatment in treatments.items():

        print(f"{disease}: {treatment}")
```

```python
if __name__ == "__main__":

    main()
```