A PROJECT REPORT

ON

Health Buddy

Submitted to:

Department of Computer Science and Multimedia

National College of Management and Technical Science

Samakhusi, Ranibari Marg

Kathmandu, Nepal

Submitted by:

Samrat Shrestha (LC00021001253)

In the partial fulfillment of the requirement for

the 2nd Semester Final Project

June 2025

Kathmandu, Nepal

ABSTRACT

The rapid advancement of artificial intelligence has led to the development of intelligent agents capable of performing complex tasks autonomously. Most modern AI systems rely heavily on graphical user interfaces; however, many enterprises and system-level applications require AI services that operate without a dedicated frontend. This project proposes the design and development of a backend-only AI agent that performs healthcare assistance, basic symptom analysis, and intelligent decision support through APIs and command-based interactions.

The proposed AI agent will accept structured or unstructured input via API requests or command-line interfaces, process the input using machine learning and natural language processing techniques, and return intelligent responses or actions. The system will be modular, scalable, and easily integrable with future frontend applications or third-party systems. The expected outcome is a flexible AI agent that can be deployed as a service for healthcare-related assistance, preliminary guidance, and decision-support tasks, reducing human effort and improving efficiency.

-

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my project supervisor for continuous guidance, encouragement, and valuable feedback throughout the preparation of this proposal. I am also thankful to the Department of Computer Science and Multimedia, National College of Management and Technical Science, for providing the academic environment and resources necessary for this work. Special thanks go to my friends and family for their constant support and motivation.

PREFACE

The motivation for selecting this project stems from a strong interest in artificial intelligence systems that operate beyond traditional user interfaces. While working with backend systems and APIs, I realized the growing demand for intelligent agents that can function independently and integrate seamlessly with other software services. This proposal reflects my learning journey in AI concepts, system design, and software architecture, and represents an effort to build a practical and industry-relevant solution.

Table of Contents

List of Figures

List of Table

List of Abbreviations

AI: Artificial intelligence

NLP: Natural Language Processor

LLM: Large Language Model

API: Application Programming Interface

1.INTRODUCTION

1.1 Background of study

Artificial Intelligence (AI) has become an important technology in modern healthcare, supporting tasks such as symptom analysis, information retrieval, and decision support. These AI-based systems help improve efficiency, reduce the workload of healthcare professionals, and increase access to basic healthcare guidance, especially in resource-limited settings.

With the growing adoption of API-driven and microservice-based architectures, software systems are increasingly designed as backend services rather than interface-heavy applications. Backend-only AI agents play a key role in this shift by providing intelligent functionalities that can be integrated into multiple platforms without requiring a dedicated frontend.

In healthcare, backend-only AI agents are particularly useful for delivering preliminary guidance and decision support in a secure and modular manner. This project focuses on designing such a backend-only AI healthcare agent that emphasizes ethical decision support rather than diagnosis or treatment.

1.2 Problem statement

Access to basic healthcare information and preliminary guidance is limited for many individuals due to time constraints, lack of awareness, or limited availability of healthcare professionals. Existing healthcare applications often require complex user interfaces and are not easily integrable with other systems. There is a need for a backend-only AI healthcare agent that can provide preliminary assistance and decision support through APIs without relying on a frontend.

1.3 Objectives of the project

- To design a backend-only AI agent with no graphical user interface
- To implement basic healthcare-related decision logic using NLP techniques
- To expose AI functionalities through secure APIs
- To ensure scalability, modularity, and easy future integration

1.4 Scope of the project

The project focuses on backend development only. It includes AI logic, APIs, data processing, and system integration. Frontend development, mobile applications, and advanced visualization dashboards are excluded from the current scope.

1.5 Signification of the project

This project demonstrates how AI agents can be deployed as independent services. It is significant for academic learning, backend-focused system design, and real-world applications such as automation tools, enterprise assistants, and developer-focused AI services.

2. LITERATURE REVIEW

2.1 Review of existing system / products

The use of artificial intelligence in healthcare has grown significantly in recent years, particularly in the areas of symptom checking, virtual health assistants, and clinical decision support systems. Many existing AI healthcare systems aim to provide users with preliminary health-related information, reduce the burden on healthcare professionals, and improve access to medical guidance.

Popular healthcare chatbots and virtual assistants such as Ada Health, Babylon Health, and Buoy Health use AI-driven questionnaires and natural language processing techniques to analyze user-reported symptoms and provide possible health insights. These systems are primarily frontend-oriented, relying on mobile or web interfaces to interact with users. While they are effective in user engagement, their backend architectures are often complex and not easily reusable or integrable into other systems.

In addition to chatbots, hospital information systems and electronic health record (EHR) platforms incorporate AI modules for alert generation, risk prediction, and patient monitoring. However, these systems are typically large-scale, expensive, and tightly coupled with proprietary software environments, making them unsuitable for lightweight academic or small-scale implementations.

2.2 Comparative analysis:

System		Frontend Dependency	customization	Deployment Flexibility
Healthcare Chatbots		High	medium	Limited
Hospital AI systems		High	Low	Restricted
Health Buddy (simulated)		None	High	Flexible

Table 1

2.3 Research Gap / Innovation justification

Despite the availability of AI-powered healthcare applications, there is a noticeable lack of backend-only AI healthcare agents designed specifically for modular integration and academic or small-scale use. Existing systems either prioritize frontend interaction or depend heavily on external cloud services.

The research gap lies in developing a backend-focused AI healthcare agent that provides preliminary healthcare assistance through APIs without enforcing a specific user interface. The proposed project addresses this gap by designing a simple, extensible, and ethically safe AI healthcare agent that focuses on decision support rather than diagnosis or treatment. This approach allows future developers to integrate advanced AI models, mobile applications, or hospital systems without redesigning the core architecture.

3. SYSTEM ANALYSIS DESIGN

3.1 Feasibility Study

- **Technical Feasibility:**
  The system is technically feasible as it is built using Python and FastAPI, which are mature, widely used, and well-supported technologies. Basic AI and natural language processing techniques can be implemented using open-source libraries. The backend-only design reduces complexity and allows development without advanced frontend skills.

- **Operational Feasibility:**
  The proposed system is easy to operate and maintain. Since it exposes functionality through APIs, it can be accessed by developers, systems, or applications without specialized training. The modular architecture allows future enhancements without disrupting existing functionality.

- **Economic Feasibility:**
  The project is cost-effective as it relies on open-source tools and frameworks. No expensive hardware, licensed software, or paid AI services are required for the prototype, making it suitable for an academic environment.
- **Legal & Ethical Feasibility:**
  The system is designed to follow ethical AI practices by providing only preliminary healthcare guidance and decision support. It does not store sensitive personal or medical data and does not attempt diagnosis or treatment, reducing legal and ethical risks.
- **Schedule Feasibility:**
  The project can be completed within the given academic timeline. The scope is well-defined, and the use of existing frameworks allows rapid development, testing, and documentation.

3.2 Requirements specification

- **Functional Requirements:**
  1. Accept healthcare-related input via API requests
  2. Process healthcare-related text and queries intelligently
  3. Return responses or actions
- **Non-functional requirements:**
  1. High availability
  2. Secure API access
  3. Scalable architecture

3.3 System Design

3.3.1 System Design

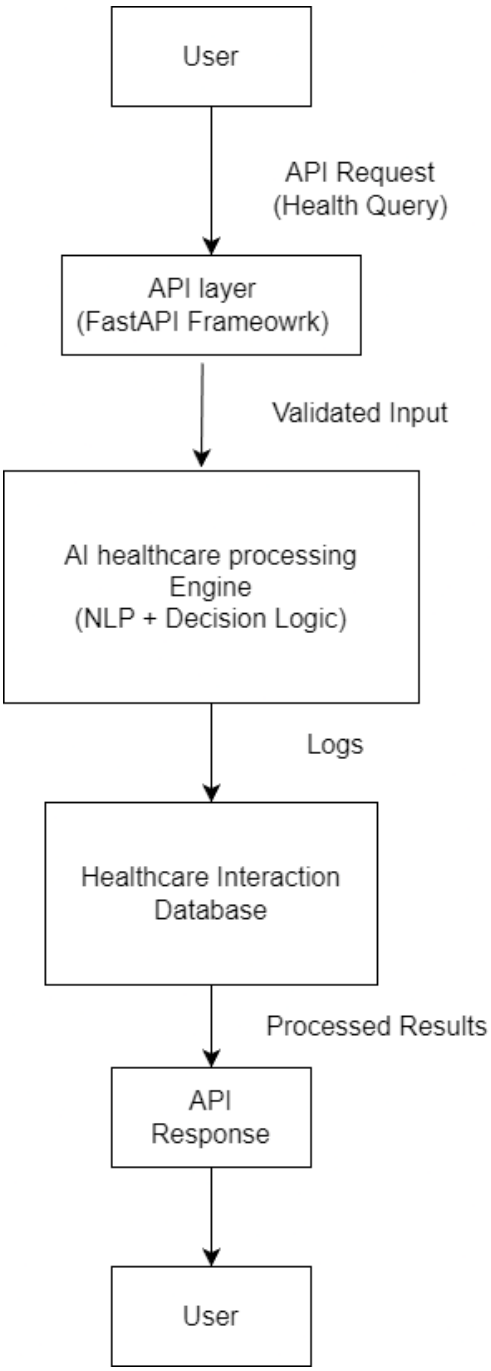

Figure 1 System Architecture

3.3.2 Data Flow Diagram

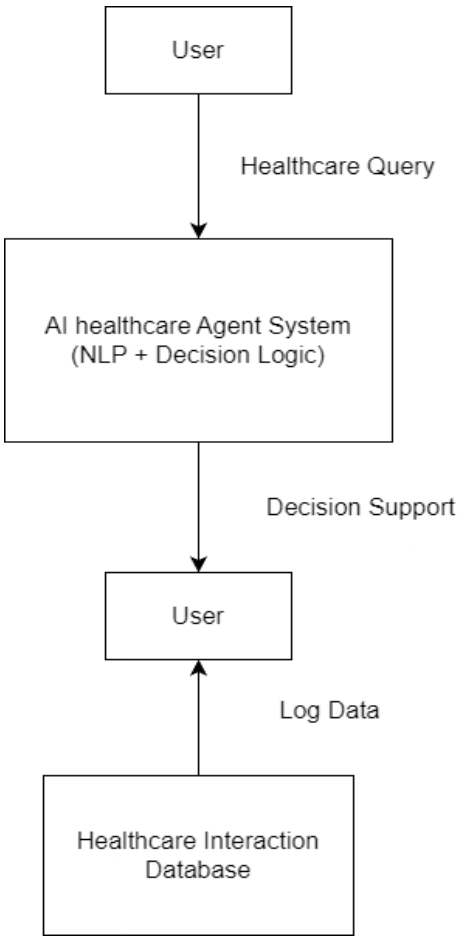

Figure 2 DFD level-0

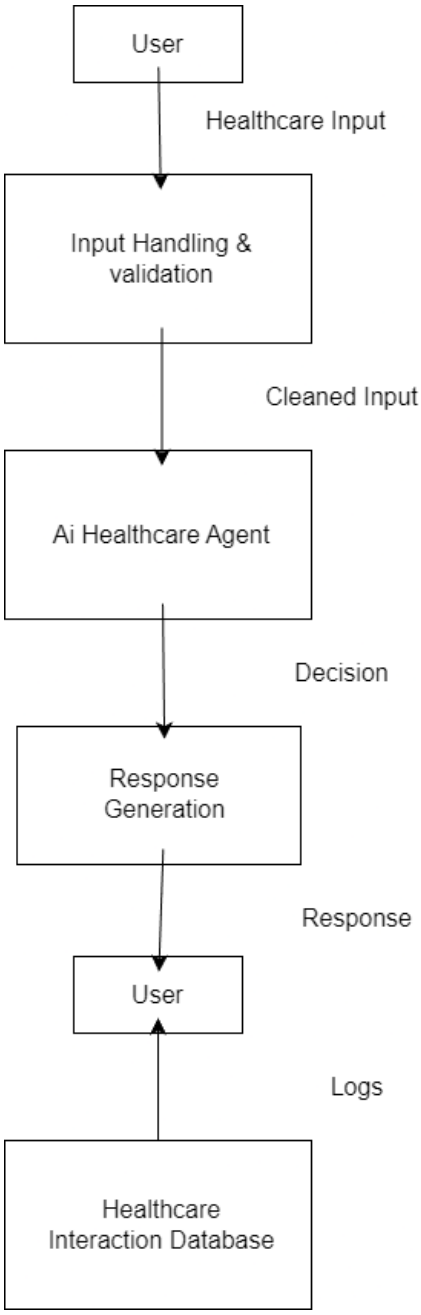

Figure 3 DFD Level-01

4.IMPLEMENTATION PLAN

4.1 Development methodology

For the development of Health Buddy, the **Agile** has been chosen. Agile Methodology is a flexible project management approach that breaks work into small, iterative cycles (sprints) for continuous delivery, adaptation, and customer feedback, prioritizing collaboration, working solutions, customer involvement, and responding to change over rigid plans and extensive documentation.

This model is suitable for the project because the requirements of *Health Buddy* are not well-defined and are expected to change frequently. Agile breaks this projects into short sprints to quickly test, get feedback, reducing time consumption. Hence, Agile is better suited for this project.

4.2 Tools, Platforms, and languages

Language:

- **PYTHON**: Python is a popular, high-level, general-purpose programming language known for its simple, English-like syntax and emphasis on code readability. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming, and is used across a wide variety of applications.

Framework:

- **FastAPI**: FastAPI is a modern, high-performance web framework for building APIs with Python 3.8+ based on standard Python type hints. It has quickly become one of the most popular choices for backend development due to its speed and developer-friendly design.
- **LangChain**: LangChain is an open-source orchestration framework designed to simplify the creation of applications powered by Large Language Models (LLMs). While you can call an LLM (like GPT-4) with a single request, real-world apps usually require a complex series of steps—like searching a database, remembering past conversations, or formatting data. LangChain provides the "glue" to connect these steps into a single workflow.

LLM Model:

- **Llama-3.2 3b instruct**: Llama-3.2 3B Instruct is a small but powerful open-source language model released by Meta. It belongs to the "lightweight" tier of the Llama 3.2 family, specifically designed for efficiency, high speed, and local deployment on edge devices (like mobile phones or laptops).

Tools and Platforms:

- **Code Editor:** We will use **Visual Studio Code (VS Code)**. It's a top choice among developers because it's free, cross-platform, and incredibly versatile. Its key features like syntax highlighting, intelligent code completion (IntelliSense), and a vast library of extensions will make the coding process more efficient and help in catching errors early on.

Version Control

- **GitHub:** We will use **Git** as our version control system, with **GitHub** serving as the central hosting platform for our code repository. GitHub is the industry standard for collaborative development and offers far more than just a backup. It enables us to track every single change made to the code, revert to previous versions if needed, and manage the project's history in a systematic way. This is crucial for managing project milestones and ensuring code integrity,

4.4 Project timeline (gant chart)'

Title Name	December				Jan				Feb			
	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4
Project Planning												
Project Design												
Documentation												
Implementation												
Testing												
Deployment												
Total	12 Weeks											

Table 2

4.5 Risk analysis

This table identifies and analyzes potential risks for the We Ride project, tailored to a front-end-only development with a Waterfall approach.

Risk Description	**Likelihood**	**Severity**	**Impact**	**Mitigation Strategy**
API Failure	Medium	High	High	Retry and fallback logic
Model Errors	Medium	Medium	Medium	Continuous testing

Table 3

5.EXPECTED OUTCOMES AND LIMITATIONS

5.1 Expected product features

- **Backend-only AI healthcare agent**
- **API-based interaction**
- **Modular and extensible design**

5.2 User Benefits and Impact

Developers and healthcare systems can integrate basic AI-driven healthcare assistance without building a frontend, saving time and resources.

5.3 Limitation of the system

- No Graphical Interface
- Limited to predefined tasks initially

6.CONCLUSION

Health Buddy is an intelligent backend-only AI agent designed to provide healthcare assistance, symptom analysis, and decision support. Unlike traditional systems that rely on complex graphical interfaces, this project focuses on a modular, API-driven architecture that can be seamlessly integrated into existing healthcare systems and developer workflows.

The system accepts both structured and unstructured inputs via API requests, processing them using Natural Language Processing (NLP) and Machine Learning (ML) to deliver intelligent, actionable health guidance. By utilizing a "Agile" development approach and focusing on a robust backend, Health Buddy aims to reduce human effort in preliminary medical triage and administrative documentation while ensuring high scalability for future frontend expansion. Key anticipated outcomes include improved efficiency in healthcare delivery and the democratization of AI-driven medical guidance through accessible command-based interactions.

7.REFERENCES

Russell, S., & Norvig, P. (2021). *Artificial Intelligence: A Modern Approach*. Pearson.

Topol, E. J. (2019). "High-performance medicine: the convergence of human and artificial intelligence." *Nature Medicine*, 25, 44–56.

Patel, D., et al. (2025). "AI Agents in Modern Healthcare: From Foundation to Pioneer — A Comprehensive Review and Implementation Roadmap for Impact and Integration in Clinical Settings." *Preprints.org*.

Quinn, T. P., et al. (2021). "Trust and medical AI: the challenges we face and the expertise needed to overcome them." *Journal of the American Medical Informatics Association*, 28, 890–4.