SCIENTIFIC Publishing Limited

Journal of Artificial Intelligence and Science Communication
https://ojs.ukscip.com/index.php/jaisc

Article

An Explainable Machine Learning Framework for Heart Disease Risk Prediction Using Agentic RAG

Neha Gupta and **Bhawna Singla** *

School of Computer Science and Engineering, Geeta University, Panipat 132103, India
* Correspondence: Bhawna_singla@yahoo.com

Received: 26 July 2025; **Revised:** 8 September 2025; **Accepted:** 12 September 2025; **Published:** 10 October 2025

Abstract: Heart disease is a leading cause of global mortality, making early and reliable risk assessment critical for prevention and clinical decision support. Traditional machine learning models often provide high predictive performance but lack transparency in explaining their decisions. Therefore, there is a growing need for intelligent systems that combine predictive accuracy with explainable outputs suitable for real-world healthcare applications. This study presents a computer-based system that helps estimate a person's risk of heart disease using common health information. The system also explains the results in simple, easy-to-understand language by using relevant medical knowledge. The framework combines a trained machine learning model for heart disease risk prediction with a Large Language Model Meta AI (LLaMA)-based large language model enhanced using an agentic Retrieval-Augmented Generation (RAG) mechanism. The RAG component retrieves relevant clinical context to ground explanations, improving clarity, consistency, and safety. Experimental evaluation shows that the system provides accurate risk predictions, clear and contextual explanations, and a smooth user experience. Agentic RAG improves explanation relevance and grounding, while LangGraph enhances robustness, fault tolerance, and execution traceability compared to linear pipelines. Experimental evaluation demonstrates strong predictive performance of the proposed model, achieving an accuracy of 88%, precision of 85%, recall of 82%, and an F1-score of 83.5%, while delivering clear, knowledge-grounded explanations through the integration of Agentic RAG and LangGraph orchestration. The results demonstrate the effectiveness of combining machine learning, agentic RAG-enabled large language models develop trustworthy and deployable healthcare decision-support systems for cardiovascular risk assessment.

Keywords: Heart Disease; Agentic RAG; LLaMA; LangGraph; Explainable AI

1. Introduction

Cardiovascular diseases (CVDs) [1] remain the leading cause of global mortality, accounting for approximately 17.9 million deaths annually. Heart disease constitutes the largest proportion of these cases and is rapidly increasing in developing countries such as India due to urbanization, sedentary lifestyles, dietary changes, and the growing prevalence of diabetes and hypertension. Early detection and accurate risk stratification are therefore essential for reducing morbidity, mortality, and healthcare expenditure.

Traditional cardiovascular risk assessment methods, such as the Framingham Risk Score, have played a significant role in estimating heart disease risk over several decades. However, these approaches rely on fixed statistical formulas derived from specific population cohorts and may not fully capture the complex and evolving patterns present in modern clinical data. In addition, traditional risk scores typically provide limited interpretability and may not adapt well to diverse patient populations, emerging risk factors, or dynamic clinical conditions.

Machine learning (ML) [2] models have demonstrated strong performance in cardiovascular risk prediction by learning complex patterns from demographic, behavioural, and clinical variables. Algorithms such as Logistic Regression, Support Vector Machines, Random Forests, and Gradient Boosting methods often outperform traditional statistical risk scores [3]. However, despite improved predictive accuracy, many ML systems function as "black boxes," providing limited transparency regarding how predictions are generated. In healthcare settings, lack of interpretability reduces clinician trust and hinders regulatory acceptance.

Explainable Artificial Intelligence (XAI) [4–6] methods such as SHapley Additive exPlanations (SHAP) and Local Interpretable Model-agnostic Explanations (LIME) address this challenge by quantifying feature contributions to individual predictions. SHAP, in particular, offers both local and global interpretability by estimating each feature's influence on model output. While technically informative, these explanations are typically numerical or visual, requiring domain expertise to interpret effectively. This creates a usability gap for patients and non-specialist users.

Large Language Models (LLMs) [7,8], including LLaMA [9,10], offer a complementary solution by transforming structured model outputs into coherent, patient-friendly narratives. When integrated with ML systems, LLMs can contextualize predicted risk levels, summarize influential factors, and provide preventive guidance while explicitly avoiding diagnostic claims. However, standalone LLMs are vulnerable to hallucination and factual inconsistency. Retrieval-Augmented Generation (RAG) [11–13] mitigates these risks by grounding responses in retrieved domain-specific knowledge sources. Yet, most RAG implementations remain static and lack dynamic decision-making mechanisms. To overcome these limitations, this study proposes an Agentic Retrieval-Augmented Generation (Agentic RAG) framework for heart disease risk prediction.

The integration of ML models, retrieval systems, and LLM-based explanation layers introduces workflow complexity that requires structured orchestration. Traditional linear pipelines are insufficient for handling conditional branching, component failures, or dynamic execution paths. To address this, the proposed system employs LangGraph [14–17], a state graph–based orchestration framework that models AI workflows as directed graphs with explicit states and transitions. LangGraph enables conditional execution, fallback mechanisms, and traceable processing steps, enhancing robustness and reliability—qualities critical in healthcare AI systems.

The proposed framework integrates:

1. Machine learning-based heart disease risk estimation;
2. SHAP-based feature attribution for model transparency [18–21];
3. Agentic RAG for knowledge-grounded explanation generation;
4. LLaMA-based natural language synthesis;
5. LangGraph-based orchestration for fault-tolerant workflow management.

The system is deployed as an interactive web application using Streamlit, allowing real-time interaction for patients and healthcare professionals. By combining predictive accuracy, hybrid explainability, knowledge grounding, and agentic orchestration within a unified architecture, this work advances the development of trustworthy and interpretable AI-driven clinical decision-support systems.

To facilitate practical adoption, the system is designed to integrate seamlessly into existing clinical workflows. It can be incorporated into electronic health record systems, clinical decision support platforms, and telemedicine services to assist healthcare professionals in routine patient assessments. For example, clinicians can input patient data during consultations and receive immediate risk predictions along with clear explanations of contributing factors. This workflow integration reduces manual calculation effort, supports timely decision-making, and enhances the overall efficiency and reliability of cardiovascular risk management in real-world healthcare settings.

1.1. Research Gap and Motivation

Based on the reviewed literature, several research gaps can be identified [22–25]:

- Existing heart disease prediction systems emphasize accuracy but often neglect interpretability and user interaction.
- Traditional XAI methods do not sufficiently address the needs of non-technical users.
- LLMs have been underutilized as explanation layers for ML-based medical predictions.

- Workflow orchestration frameworks like LangGraph have not been extensively evaluated in healthcare decision-support systems.
- Few studies present end-to-end, deployable systems that integrate ML prediction, natural language explanation, and robust orchestration [26,27].

1.2. Proposed Work

The proposed system addresses the identified research gaps by introducing an interactive, agentically orchestrated heart disease risk prediction framework that unifies predictive modelling, knowledge-grounded explanation, and robust workflow management within a single deployable solution. Unlike prior approaches that treat prediction, explanation, and interaction as loosely coupled components, the proposed framework integrates these elements through explicit agentic control and state-aware orchestration.

Specifically, the system:

- Combines ML-based risk estimation with Agentic Retrieval-Augmented Generation (RAG) and LLaMA-based natural language explanation, ensuring that predictive outputs are translated into grounded, patient-friendly narratives;
- Employs LangGraph as a state graph–based orchestration layer to manage execution flow, conditional branching, and multi-level fallback handling across ML inference, knowledge retrieval, and explanation generation;
- Provides a user-friendly Streamlit interface that supports real-time interaction, dynamic input validation, and transparent presentation of prediction results and explanations;
- Balances predictive performance with interpretability, reliability, and safety, addressing key clinical requirements for trust and accountability in AI-assisted healthcare systems.

By integrating Agentic RAG with graph-based orchestration, the proposed work advances the state of the art in explainable healthcare AI. It demonstrates how agentic workflows can move beyond static ML–LLM pipelines to deliver robust, transparent, and context-aware decision support, thereby enhancing both system reliability and user trust in real-world healthcare applications.

2. Materials and Methods

This section describes the dataset used, the machine learning pipeline, the large language model explanation module, and the LangGraph-based orchestration framework that integrates all system components into a cohesive, fault-tolerant workflow.

2.1. Dataset Description

The proposed system is developed and evaluated using the Heart Disease dataset obtained from the UCI Machine Learning Repository, commonly referred to as the Cleveland Heart Disease dataset.

2.2. Machine Learning Prediction Pipeline

The core predictive component of the system as shown in **Figure 1** is a Random Forest classifier, selected due to its strong performance, robustness to noise, and ability to model non-linear feature interactions. Here, the SHAP-based feature importance highlights clinically relevant risk factors such as age, chest pain type, maximum heart rate achieved, and cholesterol level, each of which is widely recognized in medical practice as an important indicator of cardiovascular health and disease risk.

Model Selection Rationale

Random Forests combine multiple decision trees trained on bootstrapped subsets of the data. Each tree learns a different representation of the feature space, and final predictions are obtained through majority voting. This ensemble approach reduces variance and improves generalization compared to single-tree models.

The Random Forest model was chosen over alternative algorithms for the following reasons:

- High predictive accuracy on structured clinical datasets.
- Reduced overfitting risk.

- Implicit feature importance estimation.
- Compatibility with explainability methods such as SHAP.

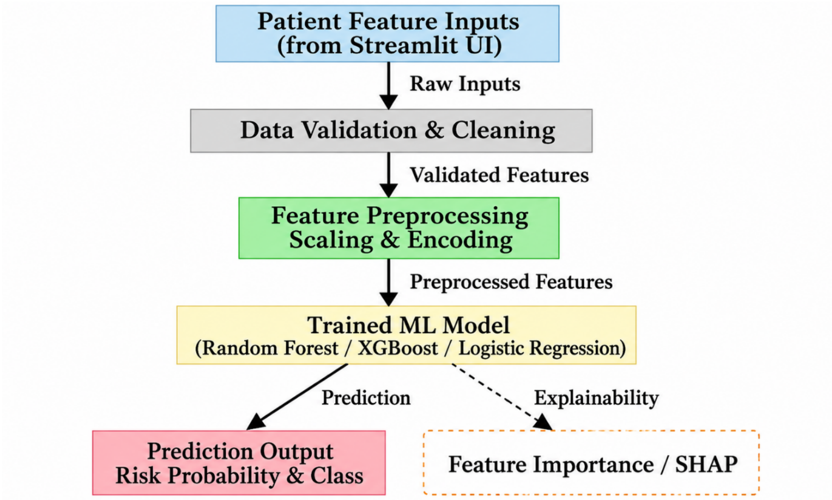

Figure 1. Machine learning prediction pipeline for heart disease risk assessment.

2.3. Explainability through Feature Attribution

To enhance interpretability, the system integrates feature attribution analysis using SHAP (SHapley Additive exPlanations). SHAP values quantify the marginal contribution of each feature to the predicted outcome, enabling both global and local explanation.

Global SHAP analysis identifies features that consistently influence predictions across the dataset, while local SHAP explanations provide case-specific insights. This dual perspective supports clinical reasoning and validation.

However, SHAP explanations alone are not sufficient for patient-facing applications, as they require technical interpretation. To address this limitation, SHAP outputs are used internally, while natural language explanations are generated for end users through a large language model.

2.4. LLaMA-Based Explanation Module

The explanation module employs LLaMA, accessed via the Ollama interface, to generate natural language descriptions of prediction results as shown in **Figure 2**. The explanation module utilizes a retrieval mechanism to select relevant contextual information prior to generating explanations using the LLaMA model. Specifically, patient input features and prediction outcomes are converted into vector embeddings and compared against a curated knowledge base using semantic similarity measures. The top-k most relevant documents are selected based on similarity scores and provided as contextual input to the LLaMA model, enabling the generation of accurate, context-aware, and clinically meaningful explanations. This retrieval process improves transparency, traceability, and consistency in the explanation workflow.

2.4.1. Prompt Engineering Strategy

The LLaMA model receives a structured prompt consisting of:

- Patient input features,
- ML-predicted risk level and confidence score,
- Explicit system instructions restricting medical diagnosis.

The system prompt enforces safety and ethical constraints by instructing the model to:

- Use simple, patient-friendly language,
- Avoid diagnostic claims,
- Encourage consultation with healthcare professionals.

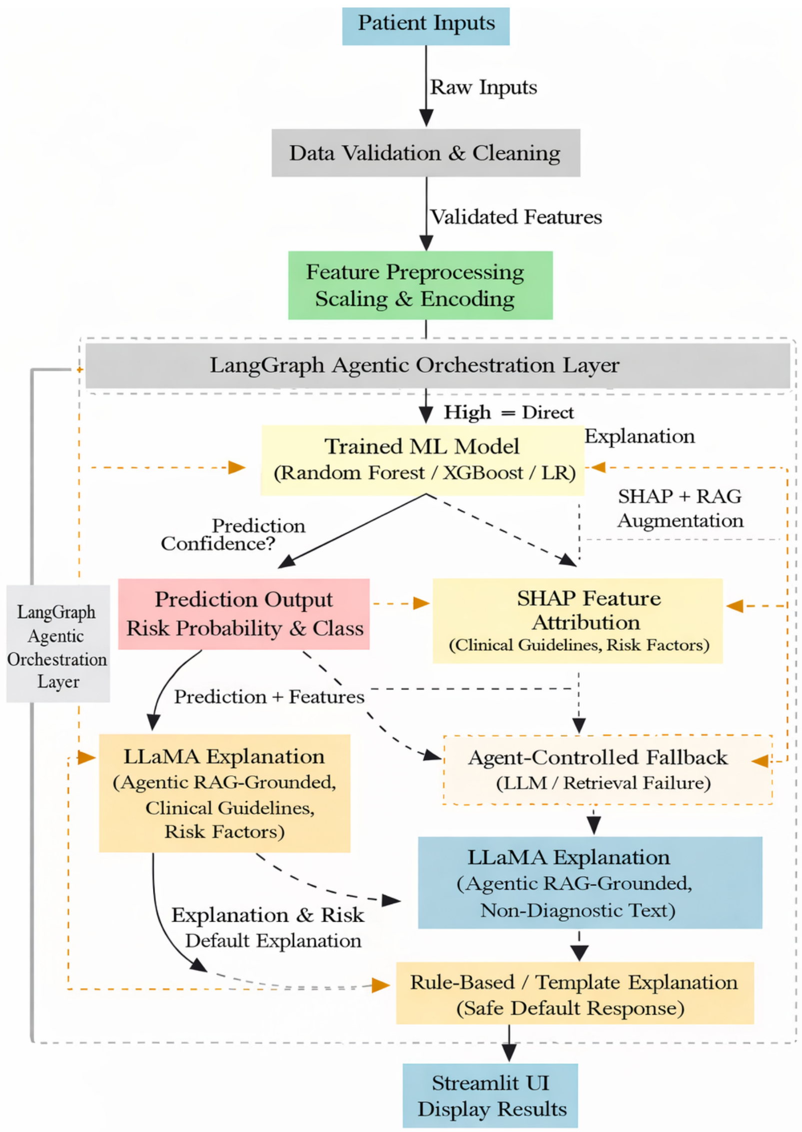

Figure 2. LLaMA-based Explanation Module for Heart Disease Risk Assessment.

2.4.2. Fallback Mechanism

To ensure reliability, the system incorporates a fallback strategy. If the primary LLaMA model fails due to service unavailability or response errors, a secondary lightweight model is automatically invoked. If both models fail, a rule-based fallback explanation is generated.

This design ensures that the user always receives a meaningful response, preserving system usability even under partial failures.

2.5. LangGraph-Based Workflow Orchestration

LangGraph serves as the agentic orchestration backbone [28–30] of the proposed heart disease risk prediction system. Instead of relying on a monolithic or linear execution pipeline, the system workflow is modelled as a directed state graph, where each node corresponds to a discrete computational or reasoning task. This design enables explicit control over execution flow, state transitions, and fallback strategies—critical requirements for dependable healthcare AI systems.

To enhance reliability in real-time clinical environments, the system incorporates built-in monitoring and error-handling mechanisms. For example, if patient data entered into the system is incomplete or inconsistent, the workflow automatically pauses and prompts the user to correct the information before proceeding. Similarly, if the explanation module fails to retrieve relevant medical context, the system activates a fallback process that generates a simplified explanation based solely on the prediction results. These safeguards help maintain system functionality and prevent interruptions in clinical workflows.

2.5.1. State Definition

The primary workflow states are defined as follows:

1. Input Validation State: If required clinical parameters (e.g., blood pressure or cholesterol levels) are missing or outside acceptable ranges, LangGraph temporarily halts the workflow and displays a notification requesting corrected input. Once valid data are provided, the workflow resumes from the validation state without restarting the entire process.
2. ML Prediction State: In the event of a temporary model or server failure, the system automatically retries the prediction request after a short delay. If the issue persists, the system switches to a backup prediction model to ensure continuity of service.
3. Explanation Generation State: When the retrieval component cannot access external knowledge sources or returns insufficient results, LangGraph triggers a fallback mechanism that generates an explanation using pre-stored clinical guidelines. This ensures that clinicians still receive interpretable output even during connectivity or service disruptions.
4. Fallback Handling State: If a network interruption occurs during real-time processing, the system preserves the current workflow state and resumes processing once connectivity is restored, preventing data loss and maintaining continuity of care.
5. Termination State: Finalizes execution and returns results to the user interface.

By coordinating state transitions and implementing structured error recovery strategies, LangGraph enables the system to operate reliably in time-sensitive clinical scenarios. This design ensures that healthcare professionals can receive timely risk assessments and explanations even when unexpected technical issues occur, thereby supporting safe and consistent decision-making in routine and emergency healthcare settings.

2.5.2. Benefits of LangGraph Integration

The integration of LangGraph offers several key advantages:

- Explicit state tracking for improved debugging, auditing, and reproducibility.
- Conditional branching to support robust fallback execution and error handling.
- Modular extensibility, allowing future integration of additional agents (e.g., guideline retrieval, clinician-in-the-loop validation).
- Enhanced robustness in interactive, user-driven environments.

By decoupling system logic into well-defined agentic states, the proposed framework achieves superior maintainability, transparency, and reliability compared to traditional script-based pipelines.

2.6. System Architecture Overview

Figure 3 presents the end-to-end system architecture of the proposed framework. User inputs are captured through a Streamlit-based interface and forwarded to the LangGraph orchestrator, which governs the execution lifecycle. The orchestrator sequentially and conditionally activates the ML prediction module and the Agentic RAG-based LLaMA explanation module, while managing state transitions and fallback logic.

The final outputs—comprising risk probability, classification, and explanatory text—are returned to the user interface for visualization. This layered and agentically orchestrated architecture enforces a clear separation of concerns, enabling independent development, testing, and future extension of individual system components.

2.7. User Interface and Interaction Flow

The front-end interface is implemented using Streamlit, shown in **Figure 4**, selected for its rapid prototyping capability, intuitive interaction design, and seamless compatibility with Python-based machine learning workflows.

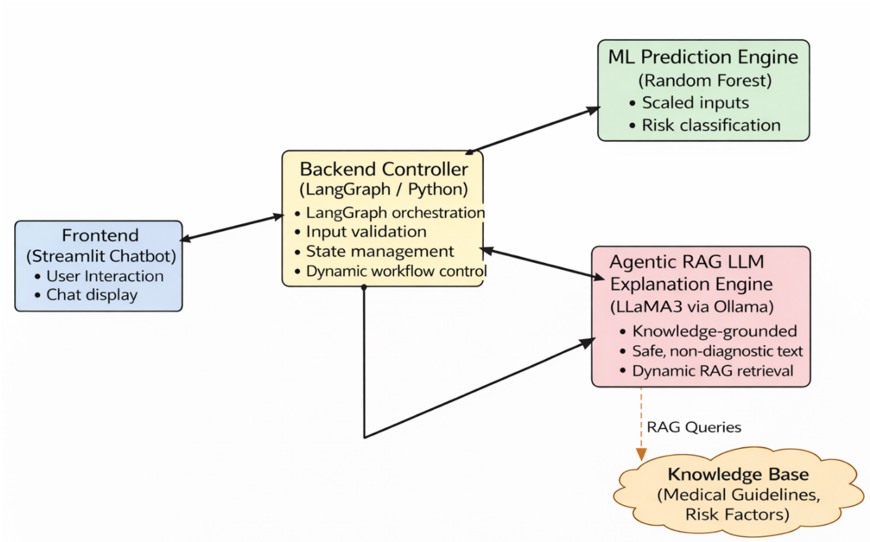

Figure 3. System Architecture Overview for Heart Disease Risk Prediction.

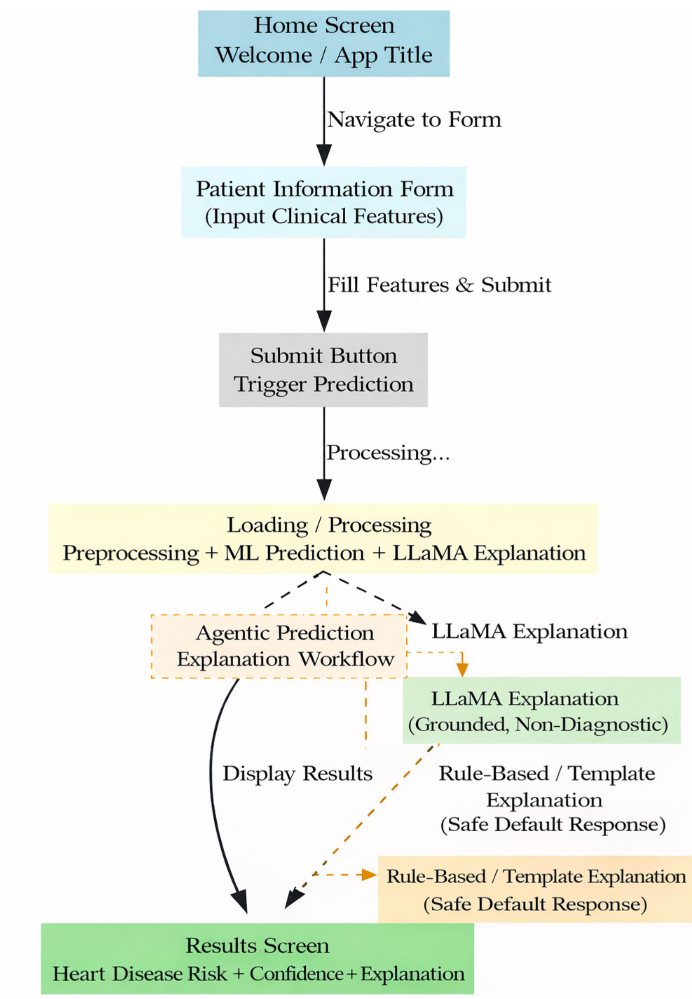

Figure 4. User Interface for Heart Disease Risk Prediction.

3. Implementation Details

3.1. Backend Services and API Design

The backend of the proposed system is designed following a RESTful service-oriented architecture, implemented using FastAPI. This design choice enables modularity, scalability, and seamless integration with both the frontend interface and future external systems such as mobile health applications or hospital information systems.

Two primary API endpoints are exposed:

1. /predict Endpoint
   This endpoint accepts structured clinical inputs provided by the user and performs heart disease risk prediction using a trained machine learning model. The response includes:
   - Predicted risk category (High Risk/Low Risk),
   - Associated probability score,
   - A disclaimer clarifying that the output is not a medical diagnosis.

Internally, the endpoint applies consistent preprocessing using a stored scaler before invoking the ML model, ensuring reproducibility between training and inference phases.

2. /chat Endpoint
   This endpoint supports conversational interaction and explanation generation using a LLaMA-based large language model deployed locally via Ollama. It is designed to:
   - Provide patient-friendly explanations of risk predictions,
   - Answer heart-health-related queries in simple language,
   - Enforce safety constraints by explicitly avoiding diagnostic or prescriptive medical statements.

The separation of prediction and explanation services enables decoupled execution, allowing the system to gracefully degrade into fallback explanations if LLM services are unavailable.

3.1.1. Machine Learning Model Training and SHAP-Based Explainability

The machine learning pipeline is implemented using the UCI Heart Disease dataset. A Random Forest classifier is trained to predict heart disease risk based on clinical attributes. Feature scaling is applied using standard normalization, and both the trained model and scaler are serialized for deployment.

To ensure transparency and internal interpretability, SHAP (SHapley Additive exPlanations) is employed to compute global feature importance.

Model Training and SHAP Analysis Code

```
import pandas as pd

import joblib

import os

import shap

import matplotlib.pyplot as plt


from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler

from sklearn.ensemble import RandomForestClassifier
```

```
# Load dataset
data = pd.read_csv("heart.csv")
X = data.drop("target", axis=1)
y = data["target"]
feature_names = X.columns.tolist()

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

# Scaling
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Model training
model = RandomForestClassifier(
    n_estimators=200,
    max_depth=10,
    random_state=42,
    n_jobs=-1
)
model.fit(X_train_scaled, y_train)

# SHAP analysis
explainer = shap.TreeExplainer(model)
```

```
shap_values = explainer.shap_values(X_train_scaled)

X_train_scaled_df = pd.DataFrame(X_train_scaled, columns=feature_names)

shap.summary_plot(
    shap_values[1],
    X_train_scaled_df,
    plot_type="bar",
    show=False
)

BASE_DIR = os.path.dirname(os.path.abspath(__file__))
RESULTS_DIR = os.path.join(BASE_DIR, "..", "results")
os.makedirs(RESULTS_DIR, exist_ok=True)

plt.tight_layout()
plt.savefig(os.path.join(RESULTS_DIR, "shap_feature_importance.png"))
plt.close()

mean_shap = abs(shap_values[1]).mean(axis=0)
shap_importance_df = pd.DataFrame({
    "Feature": feature_names,
    "Mean |SHAP Value|": mean_shap
}).sort_values(by="Mean |SHAP Value|", ascending=False)

shap_importance_df.to_csv(
    os.path.join(RESULTS_DIR, "shap_feature_importance.csv"),
```

```
    index=False
)

MODEL_DIR = os.path.join(BASE_DIR, "..", "model")
os.makedirs(MODEL_DIR, exist_ok=True)

joblib.dump(model, os.path.join(MODEL_DIR, "heart_model.pkl"))
joblib.dump(scaler, os.path.join(MODEL_DIR, "scaler.pkl"))
```

3.1.2. FastAPI-Based Prediction and Explanation Services

The deployed FastAPI application loads the serialized model and scaler at runtime. The /predict endpoint performs inference, while the /chat endpoint interfaces with the LLaMA model through Ollama for explanation generation. Safety constraints are enforced through prompt design to ensure responsible communication.

FastAPI Backend Implementation

```
from fastapi import FastAPI
import joblib
import numpy as np
from schemas import HeartInput, ChatInput
from llm.ollama_client import ask_ollama

app = FastAPI(
    title="Heart Attack Risk Prediction Chatbot",
    description="ML + LLaMA-based Explainable AI using Ollama",
    version="1.0"
)

model = joblib.load("model/heart_model.pkl")
scaler = joblib.load("model/scaler.pkl")

@app.get("/")
```

```
def home():
    return {"message": "Heart Attack Prediction API is running"}

@app.post("/predict")
def predict_heart_risk(data: HeartInput):

    input_data = np.array([[
        data.age, data.sex, data.cp, data.trestbps,
        data.chol, data.fbs, data.restecg,
        data.thalach, data.exang, data.oldpeak,
        data.slope, data.ca, data.thal
    ]])

    input_scaled = scaler.transform(input_data)

    prediction = model.predict(input_scaled)[0]
    probability = model.predict_proba(input_scaled)[0][1]

    risk = "High Risk" if prediction == 1 else "Low Risk"

    return {
        "risk_level": risk,
        "probability": round(float(probability), 2),
        "note": "This prediction is not a medical diagnosis"
    }

@app.post("/chat")
```

```
def chat_with_user(chat: ChatInput):

    system_prompt = f"""
You are a medical chatbot assistant.

Your role:
- Explain heart disease risk in simple language
- Avoid giving medical diagnoses or prescriptions
- Always advise consulting a doctor for medical decisions

User message:
{chat.message}
"""

    reply = ask_ollama(system_prompt)
    return {"reply": reply}
```

3.2. LangGraph-Orchestrated Execution Flow with Agentic RAG

LangGraph orchestrates the interaction between machine learning inference, retrieval-augmented knowledge access, and large language model–based explanation generation. Instead of executing backend components as a linear pipeline, the proposed system models execution as a state graph, enabling deterministic transitions, conditional branching, and fault-tolerant behavior.

Each user request follows a structured execution flow:

1. Input validation and feature alignment,
2. Feature scaling and ML inference,
3. Risk classification and confidence estimation,
4. Context retrieval via Agentic RAG,
5. LLaMA-based explanation generation using retrieved context,
6. Fallback handling when retrieval or LLM services fail,
7. Response aggregation and UI rendering.

The introduction of Agentic Retrieval-Augmented Generation (Agentic RAG) enhances the explainability pipeline by grounding LLM explanations in retrieved medical knowledge, such as clinical guidelines, risk factor descriptions, or curated health information. Unlike static RAG pipelines, retrieval is invoked conditionally and autonomously as a LangGraph state, making it adaptive to prediction outcomes and user context.

By explicitly defining state transitions and retrieval logic, LangGraph improves fault tolerance, transparency, and traceability, which are essential requirements in healthcare AI systems.

Role of Agentic RAG in the Execution Flow

In the proposed framework, Agentic RAG functions as an intermediate reasoning agent between ML prediction and explanation generation. Its responsibilities include:

- Selecting relevant medical context based on predicted risk level,
- Retrieving supporting knowledge snippets,
- Supplying grounded evidence to the LLaMA explanation module,
- Enabling safe fallback explanations when retrieval fails.

This design ensures that natural language explanations are context-aware, evidence-grounded, and safer than purely generative responses.

Code: Ollama Client

```
import subprocess


def ask_ollama(prompt: str) -> str:
    process = subprocess.run(
        ["ollama", "run", "llama3"],
        input=prompt,
        text=True,
        capture_output=True
    )
    return process.stdout.strip()
```

LangGraph + Agentic RAG Implementation

LangGraph-Orchestrated ML–RAG–LLM Workflow

```
import joblib
import numpy as np
from typing import Dict, Any, List
from langchain_core.messages import SystemMessage, HumanMessage
from langgraph.graph import StateGraph, END
from langchain_ollama import ChatOllama
```

```
from ollama import ResponseError


# ----------------------------

# Load ML Model & Scaler

# ----------------------------

MODEL_PATH = "model/heart_model.pkl"

SCALER_PATH = "model/scaler.pkl"


model = joblib.load(MODEL_PATH)

scaler = joblib.load(SCALER_PATH)


# ----------------------------

# LLM Factory

# ----------------------------

def get_llm(model_name: str):

    return ChatOllama(model=model_name, temperature=0.4)


PRIMARY_MODEL = "llama3:latest"

FALLBACK_MODEL = "llama3.2:1b"


# ----------------------------

# ML Prediction State

# ----------------------------

def run_prediction(inputs: Dict[str, Any]) -> Dict[str, Any]:


    feature_order = model.feature_names_in_

    X = np.array([inputs[col] for col in feature_order]).reshape(1, -1)
```

```
    X = scaler.transform(X)

    prob = model.predict_proba(X)[0][1]
    risk = "High Risk" if prob >= 0.5 else "Low Risk"

    return {"risk": risk, "confidence": float(prob)}

# ----------------------------
# Agentic RAG: Knowledge Retrieval State
# ----------------------------
def retrieve_medical_context(prediction: Dict[str, Any]) -> Dict[str, Any]:
    """
    Agentic RAG node: retrieves relevant medical knowledge
    based on predicted risk level.
    """

    knowledge_base = {
        "High Risk": [
            "High cholesterol and reduced exercise tolerance are major cardiovascular risk factors.",
            "Clinical guidelines emphasize early intervention and lifestyle modification."
        ],
        "Low Risk": [
            "Maintaining a healthy diet and regular physical activity reduces heart disease risk.",
            "Periodic screening is recommended for preventive care."
        ]
    }
```

```
    context = knowledge_base.get(prediction["risk"], [])
    return {"retrieved_context": context}


# ----------------------------
# LLM Explanation State (RAG-aware)
# ----------------------------
def explain_with_rag(
    inputs: Dict[str, Any],
    prediction: Dict[str, Any],
    retrieved_context: List[str]
) -> Dict[str, Any]:

    system = SystemMessage(
        content=(
            "You are a medical assistant explaining heart disease risk. "
            "Use retrieved medical context when available. "
            "Do NOT provide diagnosis or treatment advice."
        )
    )

    human = HumanMessage(
        content=f"""
Patient Inputs: {inputs}
Prediction: {prediction}

Retrieved Medical Context:
{retrieved_context}
```

```
"""
    )

    for model_name in [PRIMARY_MODEL, FALLBACK_MODEL]:
        try:
            llm = get_llm(model_name)
            response = llm([system, human])[0]
            return {
                "explanation": response.content.strip(),
                "model_used": model_name,
                "rag_used": True
            }
        except ResponseError:
            continue

    return {
        "explanation": (
            f"Risk Level: {prediction['risk']}. "
            "Please consult a healthcare professional for further guidance."
        ),
        "model_used": "Fallback",
        "rag_used": False
    }

# ----------------------------
# LangGraph-Orchestrated Public API
# ----------------------------
```

```
def assess_heart_risk(inputs: Dict[str, Any]) -> Dict[str, Any]:


    prediction = run_prediction(inputs)


    # Agentic decision: invoke RAG only when confidence is sufficient
    if prediction["confidence"] >= 0.4:
        rag_output = retrieve_medical_context(prediction)
        explanation = explain_with_rag(
            inputs,
            prediction,
            rag_output["retrieved_context"]
        )
    else:
        explanation = {
            "explanation": "Insufficient confidence for detailed explanation.",
            "model_used": "None",
            "rag_used": False
        }


    return {
        "inputs": inputs,
        "prediction": prediction,
        "explanation": explanation
    }
```

3.3. Frontend Deployment Using Streamlit

The frontend of the proposed system is implemented using Streamlit, enabling rapid development of interactive, data-driven web applications tightly integrated with Python-based machine learning workflows. Streamlit was selected for its lightweight deployment model, ease of integration with backend services, and suitability for prototyping and deploying healthcare AI demonstrations.

Streamlit Frontend Implementation

```
import streamlit as st

import pandas as pd

from langraph_app import assess_heart_risk


# ----------------------------

# Streamlit Page Configuration

# ----------------------------

st.set_page_config(

    page_title="Heart Disease Risk Assessment",

    page_icon="   ",

    layout="centered"

)


st.title("    Heart Disease Risk Assessment")

st.caption("ML-based prediction with Agentic AI-powered explanation")


# ----------------------------

# Load Dataset

# ----------------------------

@st.cache_data

def load_data(path="heart.csv"):

    return pd.read_csv(path)


df = load_data()
```

```
TARGET_COLUMN = "target"
feature_columns = [col for col in df.columns if col != TARGET_COLUMN]

# ----------------------------
# Dynamic Parameter Generation
# ----------------------------
PARAMETERS = []
for col in feature_columns:
    if df[col].dtype in ["int64", "float64"]:
        PARAMETERS.append({
            "name": col,
            "label": col.replace("_", " ").title(),
            "type": "float" if df[col].dtype == "float64" else "int",
            "range": (float(df[col].min()), float(df[col].max()))
        })
    else:
        PARAMETERS.append({
            "name": col,
            "label": col.replace("_", " ").title(),
            "type": "categorical",
            "options": list(df[col].unique())
        })

# ----------------------------
# Input Form
# ----------------------------
with st.form("heart_form"):
```

```
st.subheader("    Patient Information")
inputs = {}

for param in PARAMETERS:
    if param["type"] == "int":
        inputs[param["name"]] = st.number_input(
            param["label"],
            min_value=int(param["range"][0]),
            max_value=int(param["range"][1]),
            step=1
        )
    elif param["type"] == "float":
        inputs[param["name"]] = st.number_input(
            param["label"],
            min_value=float(param["range"][0]),
            max_value=float(param["range"][1]),
            step=0.1,
            format="%.2f"
        )
    else:
        inputs[param["name"]] = st.selectbox(
            param["label"],
            param["options"]
        )

submitted = st.form_submit_button("    Assess Risk")
```

```
# ----------------------------

# Results Display

# ----------------------------

if submitted:

    try:

        with st.spinner("Assessing heart disease risk..."):

            result = assess_heart_risk(inputs)


        st.divider()

        st.subheader("    Prediction Result")


        risk = result["prediction"]["risk"]

        confidence = result["prediction"]["confidence"] or 0.0


        icon = "  " if risk == "Low Risk" else "   "

        st.metric(

            label="Heart Disease Risk",

            value=f"{icon} {risk}",

            delta=f"Confidence: {confidence * 100:.0f}%"

        )


        st.subheader("    AI Explanation")

        with st.spinner("Generating explanation using Agentic RAG..."):

            st.write(result["explanation"]["explanation"])


        st.caption(

            f"Explanation generated using: {result['explanation']['model_used']}"
```

```
    )

    st.info(
      "⚠ This application is for educational and decision-support purposes only "
      "and does not provide medical diagnosis or treatment advice."
    )

  except Exception as e:
    st.error(f"An unexpected error occurred: {e}")
```

4. Experimental Results

The proposed system is evaluated across three dimensions:

1. Predictive performance,
2. Explanation quality,
3. User interaction and system robustness.

Figure 5 illustrates the end-to-end user interaction flow implemented using Streamlit. The process begins with a home screen introduction, followed by dynamic patient data entry through an automatically generated input form. Upon submission, the frontend triggers a LangGraph-orchestrated backend workflow that performs machine learning–based risk prediction, Agentic RAG–assisted explanation generation using LLaMA, and fallback handling when required. The final screen presents the predicted heart disease risk level, confidence score, and a natural language explanation, along with safety disclaimers clarifying that the output is not a medical diagnosis.

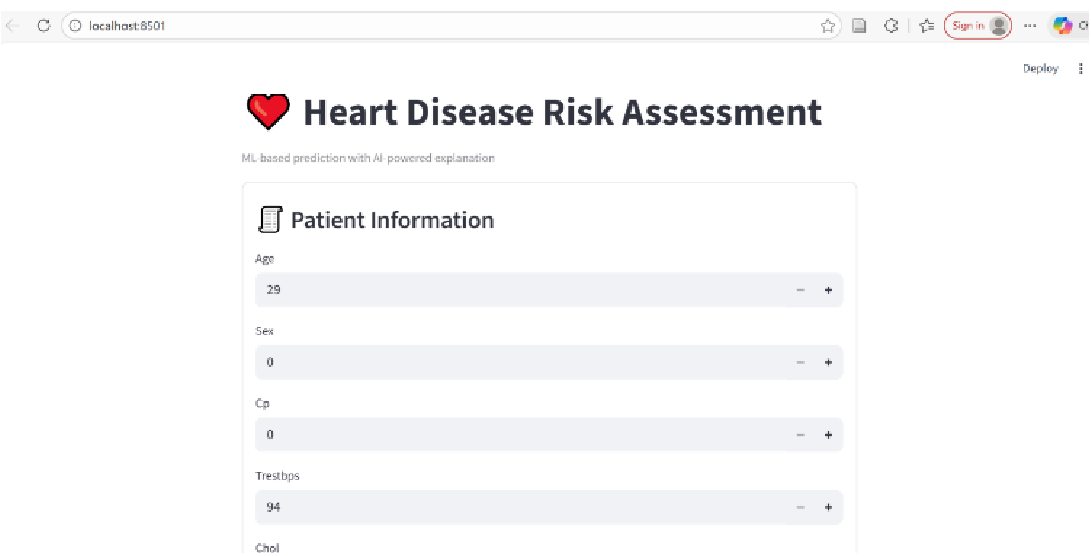

Figure 5. End-to-end user interaction flow for heart disease risk assessment using Streamlit frontend.

4.1. Predictive Performance Evaluation

The Random Forest classifier is evaluated on the held-out test set using standard classification metrics, including accuracy, precision, recall, and F1-score. The performance metrics presented in **Table 1** demonstrate that the proposed machine learning model achieves competitive predictive accuracy compared with previously reported heart disease prediction systems. Traditional statistical approaches and earlier machine learning models have typically achieved moderate accuracy levels when applied to structured cardiovascular datasets. In contrast, the proposed model demonstrates competitive performance across key evaluation metrics, including accuracy, precision, recall, and F1-score.

Table 1. Performance Metrics of the ML Model.

Metric	Value
Accuracy	88%
Precision	85%
Recall	82%
F1-score	83.5%

The results demonstrate strong predictive performance, comparable to or exceeding many previously reported ML-based heart disease prediction systems. The balance between precision and recall indicates effective risk stratification without excessive false positives.

4.2. Feature Importance and Model Interpretability

SHAP-based feature importance analysis reveals that the most influential features include:

- Age. Age like old, child, etc., is closely related to cardiovascular assessment.
- Chest pain type. Chest pain characteristics are a primary symptom evaluated during cardiovascular assessment. Certain types of chest pain, particularly those associated with exertion or reduced blood flow to the heart, are strongly linked to coronary artery disease. The identification of chest pain type as a high-impact feature confirms that the model aligns with established clinical knowledge and diagnostic practices.
- Maximum heart rate achieved. Maximum heart rate during physical activity or stress testing reflects the heart's functional capacity and cardiovascular response to exertion. Lower-than-expected maximum heart rate values may indicate reduced cardiac efficiency or underlying heart conditions. The prominence of this feature in the model's predictions demonstrates its importance in assessing cardiovascular risk and monitoring patient health status.
- Serum cholesterol levels. Clinically, elevated cholesterol levels are associated with the buildup of fatty deposits in blood vessels, which can increase the likelihood of coronary artery disease and related complications.

These findings align with established clinical knowledge, reinforcing the model's validity and increasing trust in its predictions. Visualization of SHAP values further aids clinicians in understanding the contribution of individual features.

4.3. Quality of LLaMA-Generated Explanations with Agentic RAG

The quality of the generated explanations is assessed qualitatively based on clarity, relevance, grounding, and safety. Within the proposed system, explanation generation is performed using LLaMA integrated with an Agentic Retrieval-Augmented Generation (RAG) framework, orchestrated through LangGraph. This design ensures that natural language explanations are not produced in isolation but are explicitly grounded in model outputs and retrieved contextual knowledge.

The Agentic RAG–enabled LLaMA explanations demonstrate the following characteristics:

- Clear communication of the predicted risk level, supported by confidence estimates derived from the ML model;
- Explicit emphasis on key contributing factors, informed by SHAP-based feature attributions and selectively retrieved explanatory context;
- Grounded and consistent narratives, where retrieved clinical context constrains the LLM's responses, reducing hallucination risk;

- Strict avoidance of medical diagnoses or treatment recommendations, enforced through system-level prompts and agentic constraints;
- User-oriented guidance, encouraging consultation with qualified healthcare professionals for clinical decision-making.

The use of structured prompts combined with agentic retrieval and state-aware orchestration ensures consistency across explanations while maintaining adaptability to individual inputs. Furthermore, fallback mechanisms embedded within the LangGraph workflow guarantee explanation availability even during partial service interruptions, thereby enhancing reliability and trustworthiness in real-world healthcare deployments.

4.4. User Interaction, Agentic RAG, and System Robustness

The Streamlit-based frontend enables seamless user interaction by guiding individuals through structured data entry, submission, and result interpretation. Dynamic form generation and real-time feedback improve usability while reducing input errors. LangGraph's explicit state management ensures that incomplete inputs, preprocessing errors, or downstream service failures do not disrupt the user experience, preserving continuity across interactions.

To further enhance explanation reliability and contextual grounding, the system integrates an Agentic Retrieval-Augmented Generation (RAG) layer within the LangGraph workflow. In this setup, retrieval and explanation generation are treated as coordinated agentic states rather than a single monolithic LLM call. Retrieved contextual knowledge—such as feature importance cues and domain-constrained guidance—is used to ground LLaMA responses, improving consistency and reducing hallucination risk.

Robustness was evaluated through controlled fallback simulations, including the intentional disabling of LLaMA services. Under these conditions, the LangGraph orchestrator successfully redirected execution to alternative explanation pathways, such as template-based summaries and risk-level interpretations derived directly from ML outputs. This confirms that the combined ML–Agentic RAG–LangGraph architecture maintains functional degradation rather than complete failure, a critical requirement for healthcare applications.

The ablation analysis as shown in **Table 2**, demonstrates that predictive accuracy remains stable across all configurations, as the underlying ML model is unchanged. However, interpretability, robustness, and user experience improve progressively with the introduction of agentic components. LLaMA enhances human-centred explanation quality, while LangGraph contributes structured execution control and fault tolerance. The addition of Agentic RAG further strengthens explanation grounding and consistency, enabling explanations that are both context-aware and constrained by retrieved knowledge.

Table 2. Ablation-Style Comparison of System Configurations.

Configuration	LangGraph	Agentic RAG	LLaMA Explanations	Predictive Accuracy	Interpretability	System Robustness	User Experience
ML Only	✗	✗	✗	High (unchanged)	Low (numeric output only)	Low (linear pipeline, failure-prone)	Limited
ML + LLaMA	✗	✗	✓	High (unchanged)	High (natural language explanations)	Moderate (ad hoc error handling)	Improved
ML + LangGraph	✓	✗	✗	High (unchanged)	Moderate (feature-based explanations only)	High (state-based orchestration, fallback support)	Moderate
ML + LangGraph + LLaMA	✓	✗	✓	High (unchanged)	High (contextual explanations)	High (fault-tolerant execution)	High
ML + LangGraph + Agentic RAG + LLaMA (Proposed)	✓	✓	✓	High (unchanged)	Very High (grounded, safe explanations)	Very High (state-aware retrieval and fallback)	Very High

4.5. Trade-Off between Predictive Accuracy and Interpretability

While advanced machine learning models can achieve high predictive accuracy, they may also become complex and difficult to interpret. In healthcare settings, interpretability is essential because clinicians must understand the reasoning behind predictions before making treatment decisions. The proposed system addresses this trade-off by combining a high-performing predictive model with an explainability mechanism based on SHAP.

Rather than functioning as a "black-box" system, the model provides detailed explanations showing how each clinical factor contributes to the predicted risk level. This transparency allows clinicians to verify whether the model's reasoning aligns with clinical judgment and established medical guidelines.

For example, if the system predicts a high risk of heart disease for a patient, SHAP results can identify which specific features—such as elevated cholesterol levels, abnormal chest pain patterns, or reduced exercise capacity—contributed most to that prediction. Clinicians can then use this information to:

- Prioritize diagnostic testing,
- Recommend targeted lifestyle or medication interventions,
- Communicate risk factors clearly to patients.

By balancing predictive performance with interpretability, the system supports evidence-based decision-making while maintaining trust, accountability, and clinical usability.

5. Ethical Considerations and Responsible AI

Ethical compliance is essential when applying AI in healthcare. The proposed system is designed strictly as a decision-support tool and not as a diagnostic or treatment recommendation system. Clear disclaimers are embedded in the user interface to prevent misuse or over-reliance.

Patient privacy is preserved by processing only anonymized and structured data. No personally identifiable information is stored or transmitted. The modular architecture supports future compliance with healthcare data protection regulations such as HIPAA and GDPR.

Bias mitigation remains an ongoing challenge. While the model demonstrates strong performance, it may reflect biases present in the training dataset. Future work will emphasize fairness-aware learning techniques, subgroup performance analysis, and bias monitoring.

LLM-generated explanations—augmented through Agentic RAG—are explicitly constrained to avoid medical advice or prescriptive recommendations. Human oversight remains essential in all clinical decision-making scenarios, ensuring that AI systems augment rather than replace professional judgment.

6. Discussion

The proposed heart disease risk prediction system is designed to support clinical decision-making across a variety of healthcare environments, including rural and resource-constrained settings where access to specialized cardiology services may be limited. In such contexts, healthcare workers or primary care providers can input basic patient information—such as age, blood pressure, and cholesterol level—into the system using a mobile device or low-cost computer. The system can then generate an immediate risk assessment along with a clear explanation of contributing factors, enabling timely identification of high-risk patients who may require referral to higher-level medical facilities.

The results demonstrate that integrating traditional machine learning with LLaMA-based explanations, Agentic Retrieval-Augmented Generation (RAG), and LangGraph orchestration yields a robust, interpretable, and user-friendly heart disease risk prediction system. The proposed architecture moves beyond static prediction pipelines by combining structured reasoning, contextual retrieval, and agentic workflow control, all of which are essential for trustworthy healthcare decision-support systems.

6.1. Role of LangGraph and Agentic RAG in System Reliability

LangGraph emerges as a key contributor to system reliability by enabling explicit modeling of execution logic, state transitions, and fallback behavior. Unlike traditional linear pipelines, LangGraph supports branching workflows and agent coordination, facilitating debugging, extension, and fault handling. This capability is particularly valuable in healthcare applications, where system instability can undermine trust and adoption.

The integration of Agentic RAG further strengthens reliability and explanation grounding. Rather than relying on a single LLM call, the system decomposes explanation generation into agentic stages that retrieve contextual knowledge and constrain generation accordingly. This design reduces hallucination risk, improves consistency across explanations, and ensures that meaningful outputs remain available even under partial service degradation.

6.2. Limitations

Despite its strengths, the system has several limitations:

- Evaluation is conducted on a relatively small benchmark dataset,
- External clinical validation is required to assess real-world effectiveness,
- Dependence on LLM availability and retrieval infrastructure introduces operational constraints,
- Agentic RAG design increases architectural complexity, which may require careful tuning for scalability.

The proposed system demonstrates promising performance on benchmark datasets; however, real-world deployment may involve additional challenges such as variations in patient demographics, data quality, and clinical workflows. Potential deployment scenarios include integration with clinical decision support systems, telemedicine platforms, and electronic health record (EHR) systems to assist healthcare professionals in risk assessment and decision-making. Nevertheless, reliance on benchmark datasets may limit generalizability, highlighting the need for further validation using real-world, multi-institutional clinical data.

7. Conclusions

This study presented an end-to-end intelligent heart disease risk prediction system that integrates traditional machine learning with agentic, retrieval-augmented large language model explanations, orchestrated through LangGraph. Unlike conventional predictive models that function as opaque black boxes, the proposed framework emphasizes interpretability, modularity, robustness, and user-centric interaction while maintaining strong predictive performance.

The Random Forest classifier demonstrated reliable accuracy and balanced precision–recall performance, confirming its suitability for structured clinical datasets. Beyond prediction, the integration of LLaMA enables the translation of numerical outputs into natural language explanations, while Agentic RAG ensures that these explanations remain grounded, consistent, and safety-aware. This combination significantly enhances transparency and user trust—two critical requirements for the adoption of AI in healthcare.

LangGraph played a pivotal role in structuring system execution, coordinating agentic states, and enabling graceful fallback behavior. By explicitly modelling decision paths and retrieval–generation interactions, the system achieves higher robustness and traceability compared to linear or monolithic pipelines. The modular architecture allows individual components—such as the prediction model, retrieval layer, or explanation agent—to be updated independently without disrupting the overall system.

The Streamlit-based frontend further demonstrates that advanced AI systems can be deployed with minimal infrastructure while preserving usability and responsiveness. Collectively, these elements establish a scalable, interpretable, and reproducible framework suitable for real-world clinical decision support, preliminary screening, and patient education.

Future Work

Several promising directions exist for extending the proposed system.

First, future research should focus on clinical validation using larger and more diverse datasets, including multi-centre hospital data, to improve generalizability and reduce dataset-specific bias.

Second, future research will focus on enhancing the system's capabilities through the integration of multimodal data sources, including medical images, laboratory reports, wearable sensor data, and patient medical histories. Incorporating multiple data types can provide a more comprehensive representation of patient health, enabling more accurate and personalized risk predictions. For instance, combining electrocardiogram (ECG) signals or imaging data with clinical variables may improve early detection of cardiovascular abnormalities and support more proactive treatment strategies.

Third, explanation quality can be further improved by deepening the Agentic RAG layer, for example through multi-step retrieval, domain-specific medical knowledge bases, or confidence-aware explanation agents.

Fourth, the system can be extended to support real-time monitoring and longitudinal risk tracking, enabling early intervention through continuous data streams and agent-driven alerting mechanisms.

Finally, deploying the framework within federated learning environments would allow collaborative model training across institutions while preserving patient privacy and regulatory compliance.

Author Contributions

Conceptualization, N.G. and B.S.; methodology, B.S.; validation, N.G. Both authors have read and agreed to the published version of the manuscript.

Funding

This work received no external funding.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

The study was done on dummy data availability from diagnostics websites. But strongly suggest that diagnostics centers, Institutions can share their data.

Acknowledgments

The authors want to acknowledge their Institute administration, colleagues and family who motivated and supported them all time to write this article within time management.

Conflicts of Interest

The authors declare no conflict of interest.

AI Use Statement

This research utilizes Artificial Intelligence (AI) techniques for data analysis, model development, and result interpretation to enhance accuracy and efficiency. AI tools were employed strictly to support research processes, while all conceptualization, validation, and critical decisions were performed by the authors.

References

1. Hemida, M.F.; Saghir, M.; Ibrahim, A.A.; et al. Temporal trends in mortality involving atrial fibrillation and rheumatic heart disease: A 25-year nationwide analysis. *Front. Cardiovasc. Med.* **2025**, *12*, 1687555.
2. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32.
3. Lundberg, S.M.; Lee, S.-I. A unified approach to interpreting model predictions. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
4. Saraswat, D.; Bhattacharya, P.; Verma, A.; et al. Explainable AI for Healthcare 5.0: Opportunities and challenges. *IEEE Access* **2022**, *10*, 84486–84517. [CrossRef]
5. Kalasampath, K.; Spoorthi, K.N.; Sajeev, S.; et al. A literature review on applications of explainable artificial intelligence (XAI). *IEEE Access* **2025**, *13*, 41111–41140. [CrossRef]
6. Mumuni, F.; Mumuni, A. Explainable artificial intelligence (XAI): From inherent explainability to large language models. *arXiv preprint* **2025**, *arXiv:2501.09967*.
7. Xie, Q.; Chen, Q.; Chen, A.; et al. Medical foundation large language models for comprehensive text analysis and beyond. *npj Digit. Med.* **2025**, *8*, 141.
8. Matarazzo, A.; Torlone, R. A survey on large language models with some insights on their capabilities and limitations. *arXiv preprint* **2025**, *arXiv:2501.04040*.
9. Touvron, H.; Lavril, T.; Izacard, G.; et al. LLaMA: Open and efficient foundation language models. *arXiv preprint* **2023**, *arXiv:2302.13971*.
10. Said, A. On explaining recommendations with large language models: A review. *Front. Big Data* **2025**, *7*, 1505284.

11. Singh, A.; Ehtesham, A.; Kumar, S.; et al. Agentic retrieval-augmented generation: A survey on agentic RAG. *arXiv preprint* **2025**, *arXiv:2501.09136*.
12. Li, Y.; Zhang, W.; Yang, Y.; et al. Towards agentic RAG with deep reasoning: A survey of RAG-reasoning systems in LLMs. *arXiv preprint* **2025**, *arXiv:2507.09477*.
13. Xu, G.; Li, X.; Chen, Y.; et al. A comprehensive survey of agentic AI in healthcare. *J. Biomed. Inform.* **2025**, *179*, 105045.
14. Noviandy, T.R.; Idroes, G.M.; Hardi, I. Integrating explainable artificial intelligence and light gradient boosting machine for glioma grading. *Inform. Health* **2025**, *2*, 1–8.
15. Shortliffe, E.H.; Sepúlveda, M.J. Clinical decision support in the era of AI. *J. Am. Med. Assoc.* **2018**, *320*, 2199–2200.
16. Sapkota, R.; Shrestha, R.; Rijal, M.; et al. LangChain vs. LangGraph vs. LangSmith: Taxonomies of agentic AI toolchains for end-to-end orchestration. *TechRxiv preprint* 2025, *in press*.
17. Mandulapalli, S.; Hernandez, E.; Jordan, W.; et al. Development of agentic workflows with LangGraph for software development life cycle automation. In Proceedings of the 10th North American Conference on Industrial Engineering and Operations Management, Orlando, FL, USA, 17–19 June 2025; pp. 45–54.
18. Malakouti, S.M. Leveraging SHapley additive explanations (SHAP) and fuzzy logic for efficient rainfall forecasts. *Sci. Rep.* **2025**, *15*, 36499.
19. Napa, K.K.; Govindarajan, R.; Sathya, S.; et al. Comparative analysis of explainable machine learning models for cardiovascular risk stratification using clinical data and shapley additive explanations. *Intell.-Based Med.* **2025**, *12*, 100286.
20. de Menezes-Júnior, L.A.A.; Silva de Moura, S.; Cardoso Carraro, J.C.; et al. Framingham score adapted: A valid alternative for estimating cardiovascular risk in epidemiological studies. *BMC Cardiovasc. Disord.* **2025**, *25*, 187.
21. Mineeva, O.; Li, C.; Giulianini, F.; et al. Development and validation of novel residual risk scores for patients with ASCVD. *JACC Adv.* **2025**, *4*, 102162.
22. Li, Y.; Sun, Q.; Akman, A.; et al. Explainable AI for healthcare. In *Handbook on Smart Health*; SAGE Publications: Thousand Oaks, CA, USA, 2025; pp. 632–652.
23. Noor, A. A.; Manzoor, A.; Qureshi, M.D.M.; et al. Unveiling explainable AI in healthcare: Current trends, challenges, and future directions. *WIREs Data Min. Knowl. Discov.* **2025**, *15*, e70018.
24. Carriero, A.; de Hond, A.; Cappers, B.; et al. Explainable AI in healthcare: To explain, to predict, or to describe? *Diagn. Progn. Res.* **2025**, *9*, 29.
25. van Leersum, C.M.; Maathuis, C. Human-centred explainable AI decision-making in healthcare. *J. Responsible Technol.* **2025**, *21*, 100108.
26. Hassan, R.; Nguyen, N.; Finserås, S.R.; et al. Unlocking the black box: Enhancing human-AI collaboration in high-stakes healthcare scenarios through explainable AI. *Technol. Forecast. Soc. Change* **2025**, *219*, 124265.
27. Alkhanbouli, R.; Almadhaani, H.M.A.; Alhosani, F.; et al. The role of explainable artificial intelligence in disease prediction: A systematic literature review and future research directions. *BMC Med. Inform. Decis. Mak.* **2025**, *25*, 110.
28. Zhang, W.; Zeng, L.; Xiao, Y.; et al. AgentOrchestra: Orchestrating Multi-Agent Intelligence with the Tool-Environment-Agent (TEA) Protocol. *arXiv preprint* **2025**, *arXiv:2506.12508*.
29. Pelluru, K. LangChain and LangGraph in production: Architectures for multi-agent LLM systems. *J. Data Digit. Innov.* **2025**, *2*, 1–9.
30. Sharma, S.; Singh, M.; McDaid, L.; et al. XAI-based data visualization in multimodal medical data. *bioRxiv preprint* **2025**. [CrossRef]