American Journal of Biomedical Science & Research

www.biomedgrid.com
ISSN: 2642-1747

Review Article

An Agentic AI Framework for Ayushman Bharat Health Account (ABHA)-Integrated Continuity of Care Using Synthetic Healthcare Data

Neha Gupta and Bhawna Singla*

School of Computer Science and Engineering, Geeta University, India

***Corresponding author:** Bhawna Singla, School of Computer Science and Engineering, Geeta University, Panipat, Haryana, India.

To Cite This article: *Neha Gupta and Bhawna Singla*, An Agentic AI Framework for Ayushman Bharat Health Account (ABHA)-Integrated Continuity of Care Using Synthetic Healthcare Data. Am J Biomed Sci & Res. 2026 31(4) AJBSR.MS.ID.004056,*

DOI: 10.34297/AJBSR.2026.31.004056

Received: June 01, 2026 **Published:** 23-06-2026

Abstract

Background: The Indian healthcare ecosystem is rapidly evolving through digital initiatives such as the Ayushman Bharat Digital Mission (ABDM). Despite these advancements, challenges persist in ensuring continuity of care, timely follow-ups, preventive screening adherence, and efficient appointment management. There is a need for intelligent systems that can effectively utilize national digital health infrastructure to deliver proactive and patient-centric care.

Objectives: This study aims to develop and evaluate an agentic AI-based framework for automated appointment booking and continuity-of-care management using ABHA APIs. The objective is to improve preventive healthcare compliance, optimize clinical scheduling, reduce administrative burden, and enhance patient engagement.

Methods: The proposed framework integrates longitudinal patient health data, real-time doctor availability, and preventive care guidelines within an agentic AI architecture. The system is implemented using a PostgreSQL-backed database and Fast API-based service layer, with ABHA sandbox APIs enabling secure health ac-count integration. Performance evaluation is conducted using simulated datasets representing patient and provider interactions. Experimental evaluation demonstrates improved adherence to preventive care schedules, enhanced utilization of doctor time slots, and a significant reduction in manual administrative efforts. The agent exhibits intelligent decision-making capabilities.

Conclusions: The proposed framework streamlines clinical workflows, empowers patients through personalized recommendations, and lays the foundation for future integration with national EHR systems and telemedicine platforms.

Keywords: Agentic AI, Continuity of Care, ABHA API, ABHA Health Account, Appointment Book-ing, Preventive Screening, Indian Healthcare, Fast API, PostgreSQL, Digital Health

Introduction

Healthcare delivery in India is undergoing a paradigm shift, driven by the convergence of digital technologies, government-led initiatives, and increasing patient expectations. De-spite significant investments in infrastructure and policy frameworks, the country faces persistent challenges in ensuring continuity of care, particularly for patients with chronic diseases and those requiring preventive interventions. Continuity-of-care is critical for long-term health outcomes, adherence to treatment plans, and early detection of potential complications. However, patients often experience fragmented care due to inconsistent follow-ups, lack of centralized health records, and inefficient appointment management systems. To address these challenges, the Government of India launched the Ayush-man Bharat Digital Mission (ABDM), which aims to provide a unified digital health infrastructure for all citizens. A key component of this mission is the ABHA (Ayushman Bharat Health Account) Health Account, a unique digital identifier for every patient that enables secure access to health records, interoperability across hospitals, and the facilitation of digital health services. While ABHA provides a standardized platform for patient identification and data sharing, the effective utilization of this digital infrastructure for improving patient outcomes requires intelligent systems capable of automating routine healthcare workflows. Traditional healthcare systems, including hospital appointment scheduling and patient follow-up tracking, are largely manual, time-consuming, and prone to errors. Patients frequently miss critical preventive screenings, such as cardiovascular evaluations, cholesterol monitoring, and routine diagnostic tests, due to forgetfulness or uncoordinated care. Hospitals also face operational inefficiencies, including suboptimal utilization of doctor schedules, high administrative workload, and delayed patient care.

The integration of Artificial Intelligence (AI) in healthcare has shown promise in addressing these challenges. However, most existing AI applications are reactive, focusing on diagnosis or clinical decision support, rather than proactively managing patient care con-tenuity and administrative workflows. To bridge this gap, this study introduces a novel Agentic AI framework Joshi 2025b that leverages ABHA APIs to provide automated appointment booking and continuity-of-care management [5]. The system is designed to intelligently track patient history, detect overdue preventive screenings, and suggest op-timal appointment slots based on real-time doctor availability. By combining AI-driven decision-making with ABHA's digital infrastructure, the framework ensures that patient care is both personalized and timely. The system architecture integrates a PostgreSQL database for managing patient, appointment, and preventive screening data, along with Fast API endpoints to enable seamless communication between the AI agent and health-care applications. ABHA sandbox APIs are used to simulate real-world interactions, allowing the agent to fetch patient information and doctor schedules dynamically. This design ensures that the agent is not only capable of automated decision-making but also scalable, secure, and adaptable to evolving healthcare requirements. Furthermore, the proposed framework incorporates preventive care guidelines into its reasoning process. For example, if a patient has not undergone a cardiovascular evaluation in the last 18 months, the system can detect this gap and recommend scheduling an ECG or cholesterol test. Similarly, the AI agent can prioritize appointments for patients at higher risk, ensuring that healthcare resources are optimally utilized. This proactive approach reduces missed follow-ups, improves patient adherence to preventive care, and enhances the overall efficiency of healthcare delivery. In addition to operational benefits Katonai, Arvai, and Mesko 2025, the system addresses ethical and legal considerations [6]. Data security, privacy compliance, and informed consent are embedded into the workflow, leveraging ABHA's secure API mechanisms. By maintaining patient-centric control over health records, the system aligns with national data protection guidelines and promotes responsible AI adoption in healthcare. The primary contributions of this paper are:

a) The design and implementation of an agentic AI framework for continuity-of-care and appointment management.

b) Integration with ABHA APIs to leverage real-time patient and doctor data.

c) Automated detection of overdue preventive screenings and recommendation Borkowski and Ben-Ari 2025 of optimal appointment schedules [2].

d) Evaluation of system performance in simulated environments to demonstrate improved preventive care adherence and operational efficiency.

e) Discussion of ethical Shehab 2025, legal Al Jadaan, Babu, and Gupta 2025, and technical implications of deploying AI in the Indian healthcare ecosystem, providing insights for scalable adoption [1,8].

In summary, this work demonstrates that the integration of agentic AI with digital health infrastructure Nag, Bhagat, and Priya 2025 such as ABHA can significantly enhance patient engagement, reduce administrative burden, and improve preventive healthcare outcomes [7]. By providing a scalable and secure framework for continuity-of-care management, this system lays the foundation for the next generation of digital health solutions in India, paving the way for proactive, patient-centric, and AI-enabled healthcare delivery.

Literature Review

Digital Health Initiative in India

India has been witnessing a paradigm shift in healthcare delivery *Srinivasu et al*. [9] Joshi 2025a with the advent of digital health initiatives. The Ayushman Bharat Digitall Mission (ABDM) aims to provide a unified, interoperable digital health infrastructure for all citizens, including digital health IDs, Electronic Health Records (EHRs), and secure APIs for data exchange. A key component of this initiative is the Ayushman Bharat Health Account (ABHA), which provides a unique identifier for every patient, enabling longitudinal tracking of health records and seamless data sharing across healthcare providers. Studies have shown that digital health IDs can significantly improve continuity of care, especially in populations with chronic diseases or preventive care needs, by reducing duplication of tests, ensuring timely follow-ups, and supporting remote consultations. Despite these advances, challenges remain. Many patients in India still experience fragmented care due to limited adoption of EHRs in smaller clinics, a lack of real-time appointment systems, and insufficient automated tracking of preventive care. Traditional reminder-based systems, such as SMS or phone calls, have limited impact due to scalability constraints and lack of personalization. The combination of digital health IDs with intelligent AI systems offers a promising solution to address these gaps.

Appointment Scheduling Systems: Hospital appointment management has historically relied on manual scheduling, resulting in inefficiencies such as long patient wait times, missed follow-ups, and suboptimal utilization of healthcare personnel. Over the past decade, digital appointment systems have emerged, incorporating online booking portals, automated reminders, and basic rule-based scheduling. While these systems improve operational efficiency, they lack adaptive intelligence -the ability to prioritize patients based on clinical urgency, past care history, or preventive care needs. Research in AI-based appointment systems has demonstrated that integrating machine learning models to predict patient no-shows, optimize doctor schedules, and suggest alternate slots can significantly reduce operational bottlenecks. However, most existing solutions operate reactively; they do not proactively track longitudinal patient care or recommend preventive interventions. There is a clear need for autonomous, proactive agentic systems that can manage both scheduling and care continuity.

Agentic AI in Healthcare

Agentic AI refers to autonomous systems capable of goal-directed decision-making, planning, and executing actions with minimal human intervention. Unlike traditional AI systems that are reactive or provide recommendations, agentic AI systems operate as autonomous agents, capable of integrating multiple data sources, reasoning about patient needs, and taking sequential actions to achieve healthcare goals [3]. In the context of healthcare Al Jadaan, Babu, and Gupta 2025, agentic AI can:

a) Track longitudinal patient records.

b) Detect care gaps and overdue screenings.

c) Prioritize tasks based on clinical urgency and resource availability.

d) Interact with external APIs (e.g., hospital systems, ABHA) to automate scheduling.

e) Provide explainable recommendations to clinicians and patients.

Technical Foundations: Agentic AI systems typically rely on the following components:

a. **Knowledge Graphs:** Representing patient history, preventive care schedules, and hospital resources.

b. **Planning Algorithms:** Sequential decision-making using reinforcement learning or rule-based strategies.

c. **Natural Language Interfaces:** Enabling patients or clinicians to interact via chat or voice commands.

d. **Autonomous Execution:** Triggering actions such as appointment booking, notifications, or data retrieval.

By combining these components, agentic AI systems can act autonomously while remaining context-aware, a key requirement for continuity-of-care applications.

Langgraph For Agentic AI

Lang Graph is a framework for graph-based agentic AI that integrates Large Language Models (LLMs) with structured knowledge graphs and action execution modules. Its core idea is to represent the AI agent's environment and objectives as a graph, where nodes correspond to entities (patients, appointments, screenings, doctors) and edges represent relationships or possible actions. Key Features of *Lang Graph Yu et al.* [11].

a) **Contextual Reasoning:** LLMs provide reasoning capabilities, generating suggestions based on the patient's longitudinal history and preventive care guidelines.

b) **Action Mapping:** Each node and edge in the graph can correspond to actionable API calls, such as booking an appointment, sending a reminder, or fetching ABHA patient data.

c) **Autonomy & Planning:** The agent can traverse the graph to plan optimal sequences of actions, e.g., scheduling an ECG before a cardiology follow-up.

d) **Explainability:** The graph structure allows clinicians to visualize the reasoning path of the AI agent, enhancing trust and adoption.

e) **Integration with External APIs:** Lang Graph can trigger calls to ABHA APIs for real patient info and doctor availability.

Example Workflow in Continuity-of-Care Context: The following Figure 1 presents the workflow in continuity-of-care context. This structure enables the agent to reason over patient history, detect gaps, and autonomously schedule appointments while remaining explainable and auditable Jia, Evans, *Porter, et al*. [4].

Preventive Healthcare and Continuity of Care

Preventive care is a cornerstone of public health, aiming to reduce disease burden, detect conditions early, and promote long-term wellness. Common preventive interventions include:

a) Blood pressure monitoring

b) Cholesterol and glucose tests

c) ECG for cardiovascular risk assessment

d) Eye and dental exams

Continuity-of-care ensures that patients consistently engage with healthcare providers over time. Studies show that patients with tracked follow-ups and timely preventive screenings exhibit lower morbidity and improved health outcomes. Integrating agentic AI with digital health records enables continuous monitoring, automated reminders, and proactive scheduling, overcoming traditional barriers in follow-up adherence (Figure 1).

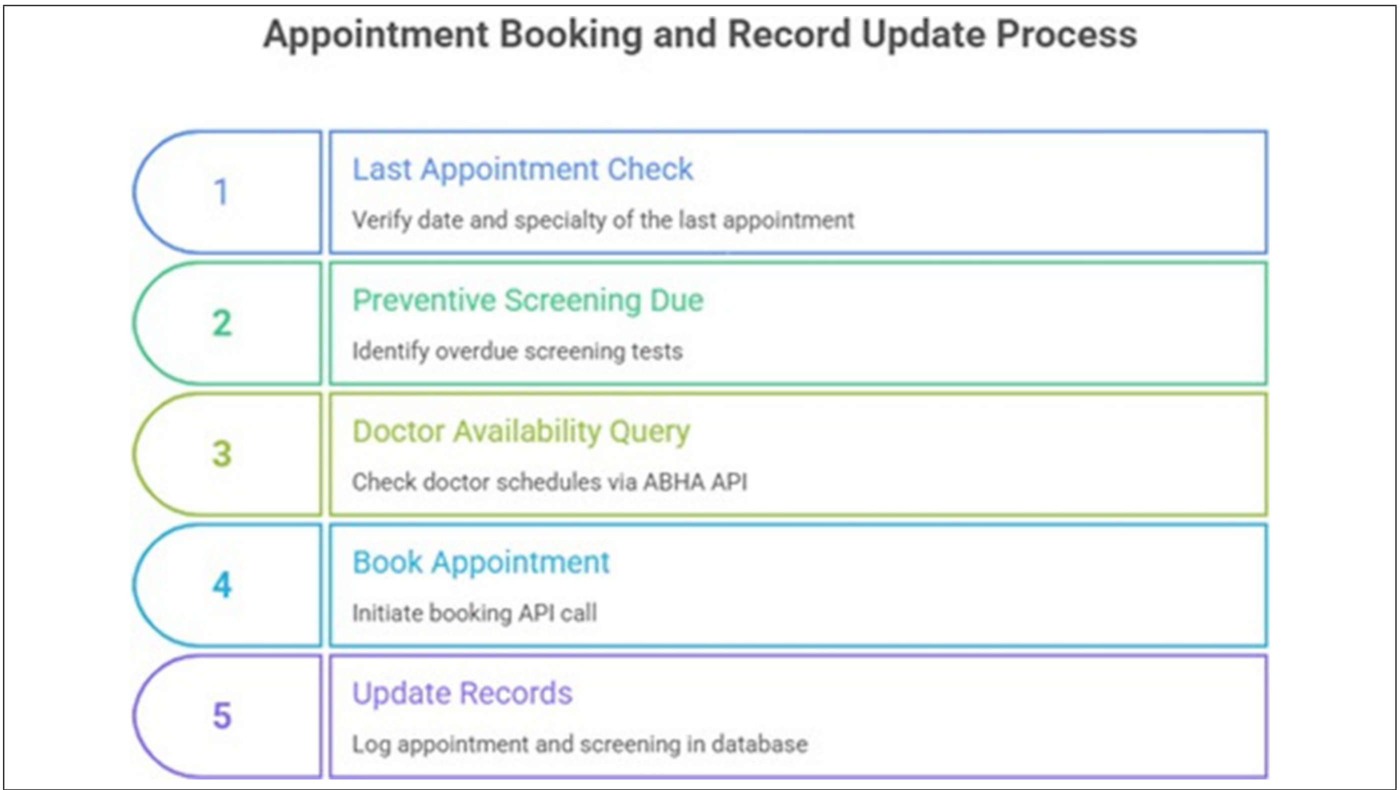

Figure 1: Appointment booking and record update process. The workflow of the agentic AI system for continuity-of-care management including patient data retrieval, preventive screening detection, doctor availability checking, appointment booking, and record updating.

Existing Research Gaps

Despite advancements, several gaps remain:

a) Limited integration with national health APIs like ABHA for real-time patient data.

b) **Reactive AI systems:** Most AI tools in healthcare respond to queries or perform predictions but do not proactively schedule care Taimoor and Rehman 2021 [10].

c) **Fragmented Preventive Care Tracking:** Patients often miss screenings due to lack of coordination between multiple providers.

d) **Explainability:** Many AI-driven systems lack transparency in decision-making, reducing clinician trust.

The proposed approach addresses these gaps by combining agentic AI, Lang Graph, and ABHA APIs to enable autonomous, explainable, and proactive continuity-of-care management.

Summary

This literature review highlights the critical need for intelligent, autonomous systems in the Indian healthcare ecosystem. While digital initiatives like ABHA provide the necessary infrastructure, Agentic AI frameworks leveraging graph-based reasoning, such as Lang Graph, are essential to ensure continuous, proactive, and personalized patient care. The combination of ABHA integration, agentic AI reasoning, and preventive care tracking forms the basis for the proposed system, which aims to enhance adherence, optimize hospital resources, and provide explainable, patient-centric decision-making.

Methodology

This section describes the design and implementation of the Agentic AI-based Continuity-of-Care system leveraging ABHA APIs. The methodology integrates digital patient data, preventive care knowledge, and autonomous agent reasoning to automate appointment scheduling and ensure timely follow-ups.

System Architecture

Figure2, illustrates the overall architecture of the proposed Agentic AI framework, where a Lang Graph-based agent orchestrates reasoning, planning, and action execution using an LLM core. It integrates user interactions, application services, and external healthcare systems (ABHA, hospital APIs, PostgreSQL) to enable secure, context-aware continuity of care. The proposed system comprises the following core components:

ABHA API Layer: It fetches patient demographic and longitudinal health data using ABHA Health Account identifiers and retrieves real-time doctor schedules from affiliated healthcare providers (Figure 2).

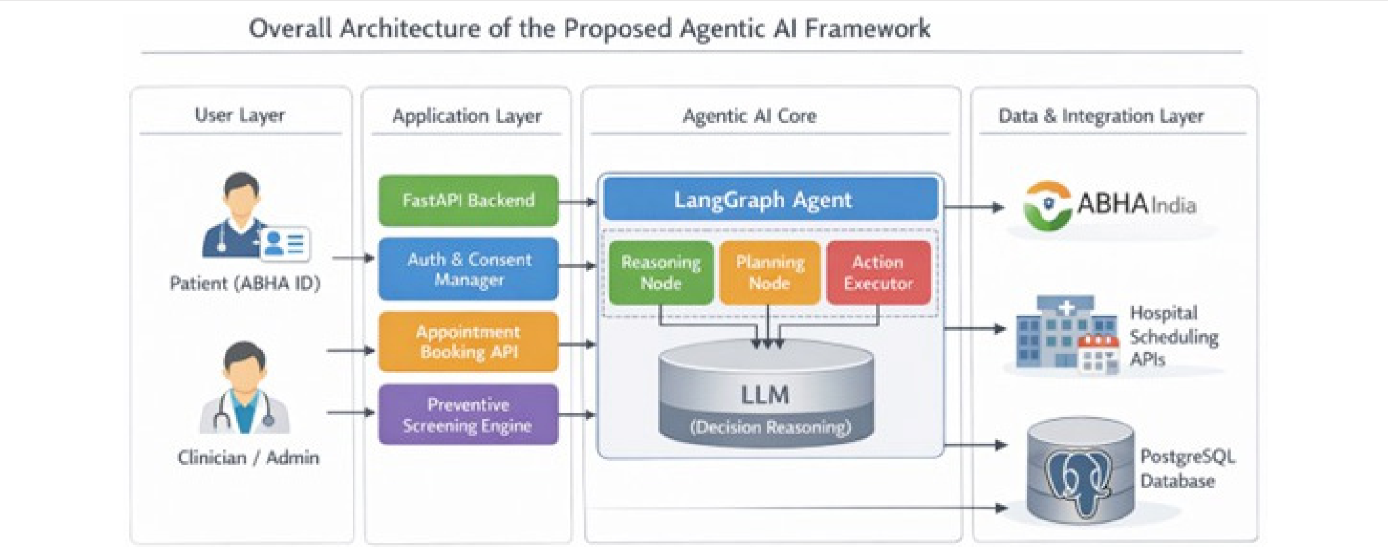

Figure 2: Proposed system architecture diagram. The overall architecture of the proposed Agentic AI framework.

Database Layer (PostgreSQL): It stores patient profiles, historical appointments, and preventive screening guidelines and enables fast querying for last appointments, due screenings, and scheduling conflicts.

Agentic AI Layer (Lang Graph-based): It represents patient history, preventive care, and hospital resources as a knowledge graph. Further, it integrates LLMs for reasoning over patient history and clinical guidelines and then plans sequences of actions autonomously, including detecting overdue screenings and booking appointments.

API Layer (Fast API): It provides endpoints for patient queries, appointment booking, and continuity-of-care reports and acts as the interface between the agent, ABHA APIs, and frontend applications.

Frontend/Interaction Layer: It enables patients and clinicians to interact via dashboards, notifications, or chatbots and Displays actionable recommendations generated by the agent.

A robust database design is critical for tracking longitudinal patient data and supporting agentic reasoning. The database is implemented in PostgreSQL with the following schema:

Tables: The system database schema is designed to support patient identity management, appointment scheduling, and preventive care tracking. The core relational tables used in the framework are described below.

A. **Patients Table:** This table stores basic patient identity information linked to the Ayushman Bharat Health Account (ABHA) (Table 1).

B. **Appointments Table:** This table records appointment details linking patients with doctors and scheduled time slots (Table 2).

C. **Preventive Screenings:** Table This table defines preventive screening guidelines used by the agentic AI for care-gap detection (Table 3).

Database Design

Table 1: Patients Table Schema.

Attribute	Data Type	Description
Id	SERIAL (PK)	Unique patient identifier
abha - id	VARCHAR (50)	Unique ABHA identifier
name	VARCHAR (255)	Patient full name

Table 2: Appointments Table Schema.

Attribute	Data Type	Description
id	SERIAL (PK)	Unique appointment identifier
patient_id	INT (FK)	References patients(id)
doctor_id	VARCHAR (50)	Doctor identifier
date	DATE	Appointment date
time	TIME	Appointment time

Table 3: Preventive Screenings Table Schema.

Attribute	Data Type	Description
id	SERIAL (PK)	Unique screening identifier
name	VARCHAR (255)	Screening name
recommended_interval _months	INT	Recommended interval (months)
specialty	VARCHAR (100)	Medical specialty involved

D. Relationships:

a) Patient → Appointment: One-to-many relationship, capturing all appointments per patient.

b) Screenings → Patient: Agent computes due screenings dynamically by comparing last appointments with recommended intervals.

ABHA API Integration

ABHA APIs provides real-time access to patient information and healthcare provider data. The methodology includes:

a) **Authentication:** Secure authentication is performed using the ABHA-provided client credentials, namely ABHA CLIENT ID and ABHA CLIENT SECRET, to ensure authorized access to healthcare resources.

b) **Fetching Patient Data:** Patient demographic and longitudinal health information is retrieved using the ABHA patient API endpoint as shown below (Listing 1).

c) **Fetching Doctor Schedules:** Doctor availability and time-slot information is obtained dynamically to enable intelligent appointment allocation (Listing 2).

d) **Data Privacy and Security:** All patient data access is conducted over secure HTTPS channels, with strict access control mechanisms ensuring that only authorized agents can read or modify sensitive information.

```
import requests
def get_patient_info(abha_id: str):
    url = f"{ABHA_API_BASE}/patients/{abha_id}"
    headers = {
        "client_id": \texttt{ABHA\_CLIENT\_ID}
,
        "client_secret": ABHA_CLIENT_SECRET
    }
    response = requests.get(url, headers=headers)
    return response.json() if response.status_code == 200
       else None
```

Listing 1: Example code for fetching patient data from ABHA API.

```
def get_doctor_schedule(doctor_id: str, date: str):
    url = f"{ABHA_API_BASE}/doctors/{doctor_id}/slots?date={
       date}"
    response = requests.get(url, headers=headers)
    return response.json()
```

Listing 2: Example code for fetching doctor schedules from ABHA API.

In this study, no real patient data was used. Instead, synthetic datasets were generated using statistically grounded distributions reflecting real-world Indian healthcare workflows. Patient age followed a Gaussian distribution ($\mu = 45$, $\sigma = 12$), appointment frequency was modeled using a Poisson process, and preventive screening adherence was simulated using interval-based temporal rules. This approach ensures privacy preservation while maintaining realistic system behavior (Figure 3).

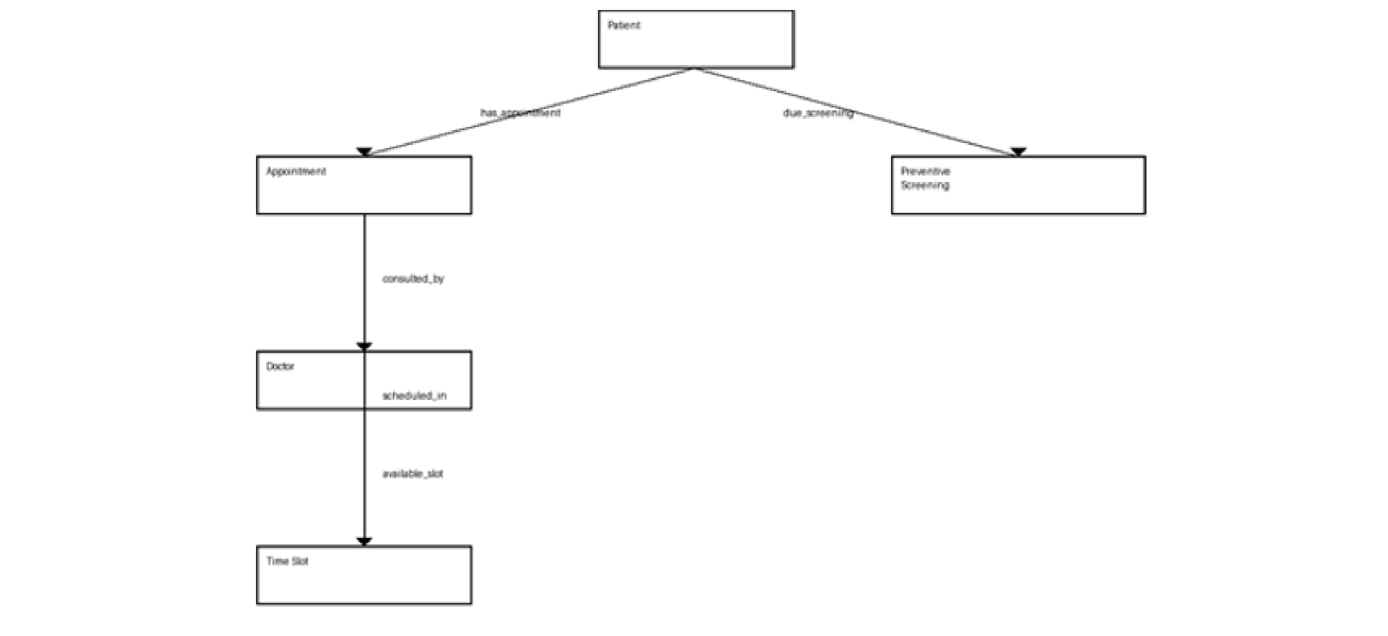

Figure 3: Knowledge representation graph. The knowledge graph representation of the patient history, preventive care guidelines, and hospital resources.

Agentic AI Layer

The Agentic AI is the core of the system, responsible for proactive decision-making. It combines Lang Graph knowledge representation with LLM-based reasoning and actionable endpoints.

Knowledge Graph Representation: Figure 3 illustrates a knowledge graph–based continuity-of-care model, where a patient node is linked to appointments and due preventive screenings. It shows how appointments connect to doctors and available time slots, enabling the agent to reason over care gaps and automatically schedule appropriate follow-ups.

Reasoning and Planning: Figure 4 depicts the Lang Graph-based agentic reasoning workflow, where the agent progresses from patient context (ABHA ID) through preventive screening analysis and doctor matching to autonomous appointment booking. It highlights explainable, step-by-step decision-making with real-time database updates and notifications to ensure continuity of care.

Continuity-of-Care Agent Algorithm: The overall workflow executed by as shown in Figure 5, the agentic AI system is summarized as follows (Figure 4&5).

Figure 4: Reasoning and planning workflow Reasoning and planning workflow.

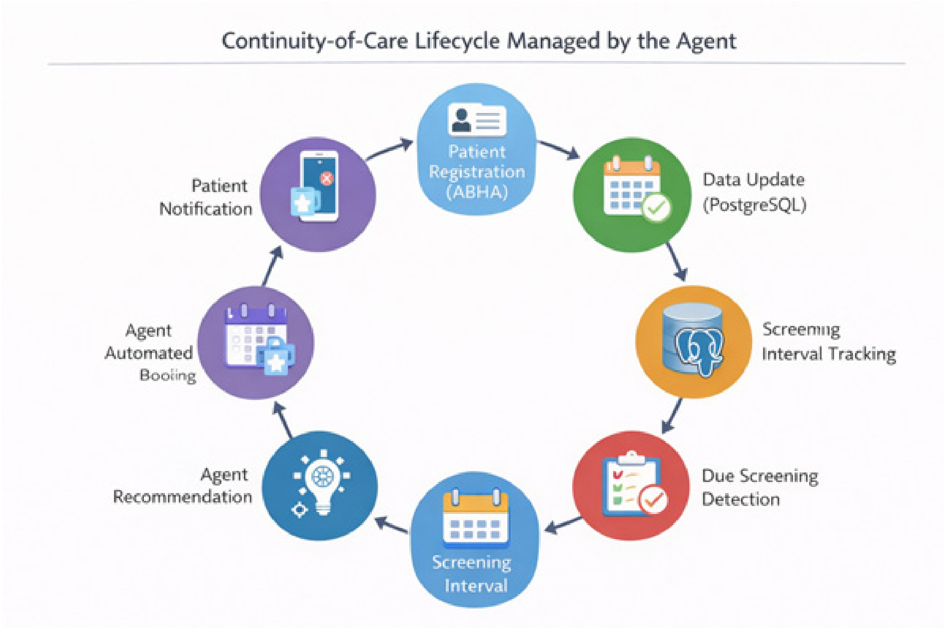

Figure 5: Continuity-of-Care Agent Workflow the workflow of the continuity-of-care agent.

a. **Retrieve Patient Profile**: The agent retrieves the patient profile using the ABHA API based on the provided ABHA ID.

b. **Query Historical Appointments**: The system queries the backend database to obtain the patient's most recent appointment for each medical specialty.

c. **Detect Due Preventive Screenings**: Preventive screenings are identified by comparing the elapsed time since the last appointment with the recommended screening intervals stored in the preventive screenings table.

d. **Retrieve Doctor Availability**: The agent queries the ABHA API to obtain real-time doctor schedules relevant to the required specialty.

e. **Plan Optimal Appointment**: Appointment planning considers patient preferences, available time slots, and clinical urgency.

f. **Execute Booking and Update Records**: The selected appointment is booked, and the database is updated to reflect the new scheduling information.

Example Agent Pseudocode (Listing 3)

```
def continuity_of_care_agent(abha_id):
    patient = get_patient_info(abha_id)
    last_appt = get_last_appointment(patient.id)
    due_screenings = get_due_screenings(patient.id)

    for screening in due_screenings:
        available_slot = find_available_slot(screening['specialty
        '])
        if available_slot:
            book_appointment(patient.id, available_slot)
            log_action(patient.id, screening['name'],
              available_slot)
```

Listing3: Pseudocode for the continuity-of-care agent.

The database scheme for continuity-of-care management is illustrated in the following figure 6, which captures the relationships between patients, appointments, and preventive screenings, enabling efficient querying and reasoning by the agentic AI system.

Fastapi Endpoints

Fast API is used to expose RESTful endpoints enabling agent interaction and automation.

Continuity- of- Care Endpoint (Listing 4) (Figure 6)

```python
@app.get("/ continuity_of_care /{abha_id }")
def continuity_of_care (abha_id :  str):
    return   continuity_of_care_agent (abha_id )
```

Listing 4: FastAPI endpoint for continuity-of-care management.

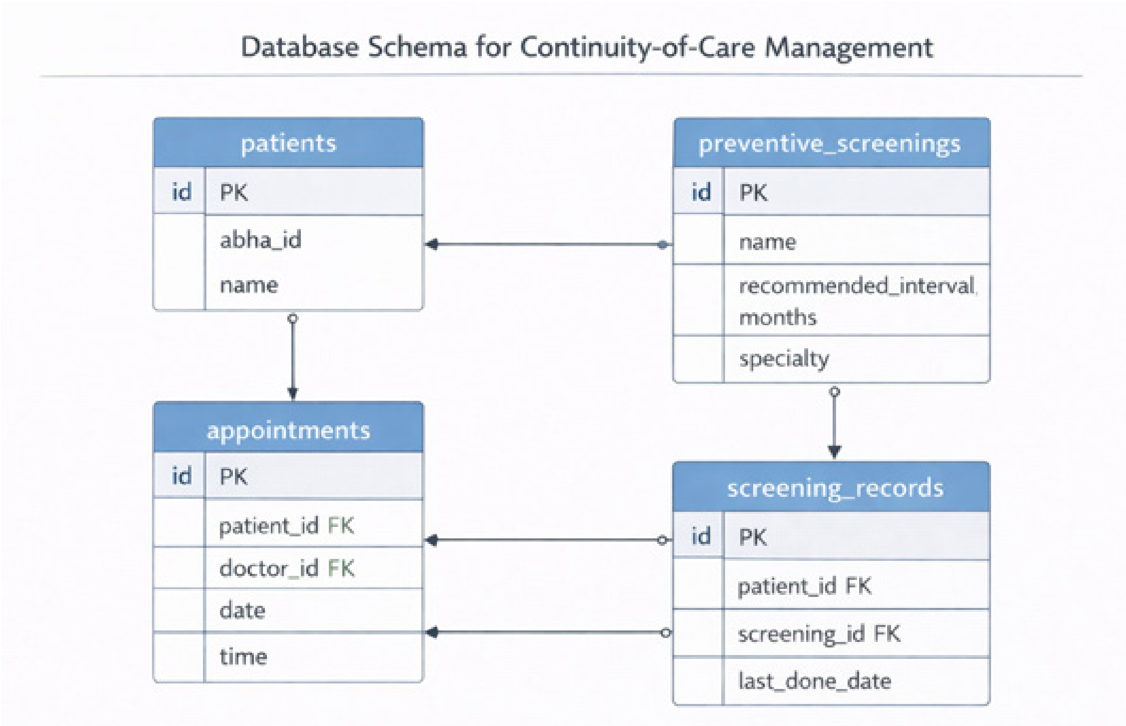

Figure 6: Database Schema for Continuity-of-Care Management Database schema for continuity-of-care management.

```python
@app.post("/ book_appointment ")
def book_appointment_endpoint(patient_id : int, doctor_id : str,
    date: str, time: str):
     return  book_appointment(patient_id , doctor_id , date, time )
```

Listing 5: FastAPI endpoint for booking appointments.

Appointment Booking Endpoint (Listing 5)

```python
@app.get("/ slots /{ doctor_id }")
def get_slots(doctor_id: str, date: str):
     return get_doctor_schedule(doctor_id, date)
```

Listing 6: FastAPI endpoint for retrieving doctor slots

Doctor Slot Retrieval Endpoint (Listing 6)

Preventive Screening Detection Algorithm

The agent computes due preventive screenings by comparing the date of the last appointment with the recommended screening interval for each preventive test (Listing 7).

```
def get_due_screenings(db, patient_id):

today = datetime.today().date()
screenings = db.query(PreventiveScreening).all() due
= []
last_appt = get_last_appointment(patient_id)

for s in screenings:
    months_since_last = 999 if not last_appt else \ (
        today.year - last_appt.date.year) * 12 + \ (
        today.month - last_appt.date.month)

    if months_since_last >= s.recommended_interval_months: due.
        append ({
            "screening_name": s.name, "
            specialty": s.specialty,
            "months_since_last": months_since_last
          })
  return due
```

Listing 7: Algorithm for detecting due preventive screenings.

Agentic AI and Langgraph Integration

Lang Graph Node Mapping:

a. Patient → ABHA ID

b. Last Appointment → Database node

c. Preventive Screening → Screening node with interval metadata

d. Doctor Slots → ABHA API node

e. Actions → Booking, Notification, Logging

f. Agent Workflow:

a. Construct Lang Graph for patient.

b. Use LLM reasoning to prioritize screenings and slots.

c. Traverse the graph to select optimal appointment.

d. Trigger API calls for booking and logging.

e. Return feedback to patient interface.

Security and Ethical Considerations

Figure 7 illustrates an ethical and consent-aware agent execution flow, where every patient request is validated through ABHA-based consent checks before autonomous reasoning is performed. The inclusion of human-in-the-loop confirmation and audit log generation ensures transparency, accountability, and regulatory compliance in automated appointment booking (Figure 7).

a. Data Security: ABHA API ensures encryption in transit and at rest.

b. Patient Consent: Patients opt-in for automated reminders and agentic AI interactions.

c. Audit Trail: All agent actions are logged in the database for accountability.

d. Explainability: Lang Graph enables clinicians to review agent reasoning for each recommendation.

Summary of Methodology

The methodology combines ABHA API integration, PostgreSQL database management, Fast API endpoints, and Lang Graph-based agentic AI to deliver a proactive, autonomous, and explainable continuity-of-care system. The agent:

a. Detects overdue preventive screenings.

b. Suggests optimal appointments based on real-time doctor availability.

c. Updates patient records automatically.

d. Provides an explainable reasoning path via graph traversal.

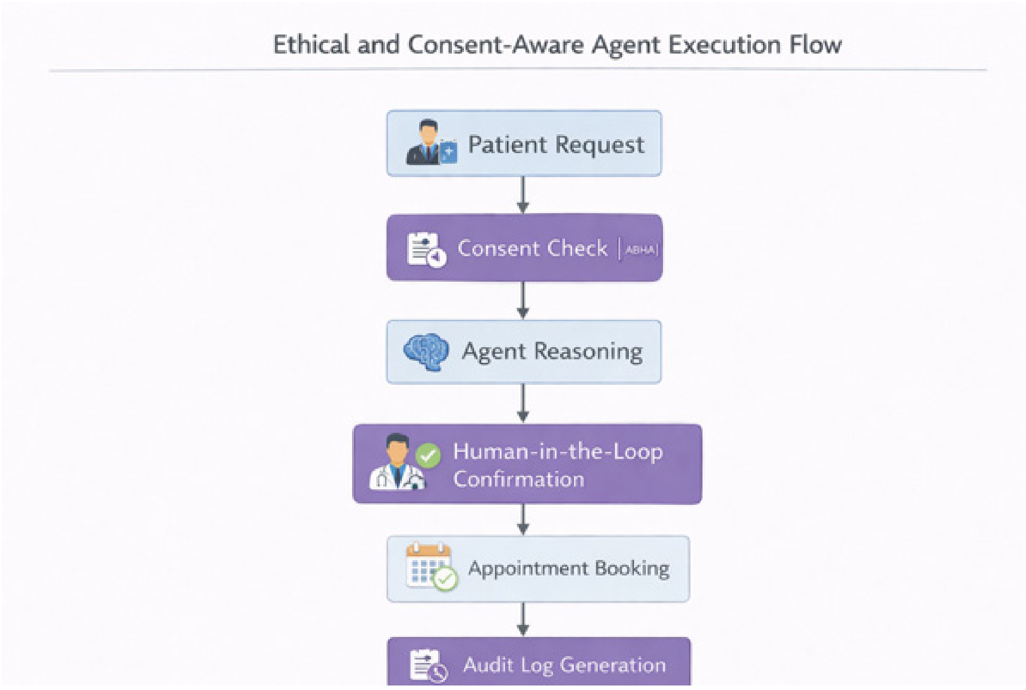

Figure 7: Ethical and Security Framework Ethical and security framework for the agentic AI system.

This framework addresses critical gaps in current Indian healthcare systems, ensuring preventive care adherence, reduced administrative burden, and improved patient out-comes.

This section details the practical implementation of the proposed Agentic AI Continuity-of-Care system, including database setup, API integration, agent logic, and the technical stack used.

Implementation Details

Development Stack

Table 4: Technology Stack Used in the Proposed Framework.

Component	Technology / Tool	Purpose
Programming Language	Python 3.11	Core application logic
Web Framework	Fast API	REST API endpoints for patient–agent interaction
Database	PostgreSQL	Patient records, appointments, and preventive scree
ORM	SQL Alchemy	Object–relational mapping
Agentic AI Framework	Lang Graph + LLM	Graph–based reasoning and autonomous decision–m
API Integration	ABHA Sandbox API	Patient data and doctor schedule retrieval
Server	Uvicorn	ASGI server for Fast API

Database Setup

A PostgreSQL database named ABHA care is used to store and manage healthcare data. The database consists of three primary tables, as described below.

Patients Table:

This table stores basic patient identification details linked to their ABHA accounts.

```
CREATE TABLE patients (id SERIAL PRIMARY KEY, abha_id VARCHAR (50) UNIQUE NOT NULL, name
VARCHAR (255) NOT NULL);
```

Appointments Table:

This table records patient appointments along with associated doctors and time slots.

```
CREATE TABLE appointments (id SERIAL PRIMARY KEY, patient_id INT REFERENCES patients(id),
doctor_id VARCHAR (50), date DATE, time TIME);
```

Preventive Screenings Table:

This table maintains preventive screening definitions, recommended intervals, and medical specialties.

Preventive_Screenings (ID Serial Primary Key, Name VARCHAR (255), Recommended_Interval_Months INT, Specialty Varchar (100)

```
CREATE TABLE preventive_screenings (id SERIAL PRIMARY KEY, name VARCHAR (255),
recommended_interval_months INT, specialty VARCHAR (100));
```

Fast API Backend

Fast API is used to expose RESTful endpoints that enable interaction between the agentic AI system and external clients.

Continuity-of-Care Endpoint: This endpoint triggers the agent to analyze patient history and recommend or schedule follow-up care (Listing 8).

```python
from fastapi import FastAPI
from agent import continuity_of_care_agent

app = FastAPI()

@app.get("/continuity_of_care/{abha_id}")
def continuity_of_care(abha_id: str):
    return continuity_of_care_agent(abha_id)
```

Listing 8: FastAPI endpoint for continuity-of-care management.

Appointment Booking Endpoint: This endpoint allows the system to book appointments for patients based on agent recommendations (Listing 9).

Doctor Slot Retrieval Endpoint: This endpoint retrieves available doctor time slots for a given date (Listing 10).

```python
@app.post("/book_appointment")
def book_appointment_endpoint(patient_id: int, doctor_id: str,
    date: str, time: str):
    from agent import book_appointment
    return book_appointment(patient_id, doctor_id, date, time)
```

Listing 9: FastAPI endpoint for booking appointments.

```python
@app.get("/slots/{doctor_id}")
def get_slots(doctor_id: str, date: str):
    from abha_api import get_doctor_schedule
    return get_doctor_schedule(doctor_id, date)
```

Listing 10: FastAPI endpoint for retrieving doctor slots.

ABHA API Integration

ABHA APIs enables the agent to retrieve real-time patient and doctor information from the national digital health infrastructure. Authentication is performed using client credentials provided by the ABHA sandbox environment, namely ABHA CLIENT ID and ABHA CLIENT SECRET.

ABHA API Integration: The following example functions demonstrate how patient profiles and doctor schedules are fetched using secure HTTPS requests (Listing 11).

```
import requests

ABHA_API_BASE = "https://sandbox.abdm.gov.in/api"

def get_patient_info(abha_id: str):
    headers = {
        " client_id ":   ABHA_CLIENT_ID ,
        "client_secret": ABHA_CLIENT_SECRET
    }
    response        =        requests. get(
        f"{ABHA_API_BASE}/patients /{abha_id}",
        headers=headers
    )
    return response.json() if response.status_code == 200 else
       None

def get_doctor_schedule(doctor_id: str, date: str):
    headers = {
        " client_id ":   ABHA_CLIENT_ID ,
        "client_secret": ABHA_CLIENT_SECRET
    }
    response                =                requests.   get(
        f"{ABHA_API_BASE}/doctors /{doctor_id}/slots? date ={date}",
        headers=headers
    )
    return response.json() if response.status_code == 200 else []
```

Listing 11: Example code for ABHA API integration.

Agentic AI Reasoning Using Lang Graph

The proposed agent is implemented using Lang Graph, where patient data, preventive screening rules, and doctor availability are represented as a structured graph of nodes and edges. Large Language Model (LLM) reasoning is employed to autonomously plan and execute sequences of clinical actions, enabling proactive continuity-of-care management.

Knowledge Graph Nodes: The agent operates over a heterogeneous knowledge graph composed of the following node types:

a) **Patient Node:** Stores ABHA identifier and demographic attributes.

b) **Appointment Node:** Captures the most recent appointment date and clinical specialty.

c) **Preventive Screening Node:** Represents recommended screening tests and their intervals.

d) **Doctor Node:** Maintains information about available doctors and time slots.

e) **Action Node:** Encodes executable actions such as appointment booking, notification delivery, and database updates.

Graph Traversal and Decision Logic: The agent traverses the knowledge graph to identify care gaps and generate appropriate actions. The following pseudocode illustrates the core decision-making logic of the continuity-of-care agent (Listing 12).

The agent autonomously detects overdue preventive screenings, queries doctor schedules, and executes appointment bookings without human intervention, thereby reducing administrative overhead and improving care continuity.

```python
def continuity_of_care_agent(abha_id):
    patient = get_patient_info(abha_id)
    last_appt = get_last_appointment(patient["id"])
    due_screenings = get_due_screenings(patient["id"])

    actions = []
    for screening in due_screenings:
        available_slot = find_available_slot(screening["specialty"])
        if available_slot:
            book_appointment(
                patient["id"], available_slot["
                doctor_id"], available_slot["
                date"], available_slot["time"]
            )
            actions.append({
                "screening": screening["name"], "
                slot": available_slot
            })
    return {
        "patient": patient["name"], "
        actions_taken": actions
    }
```

Listing 12: Pseudocode for the continuity-of-care agent.

Preventive Screening Detection

Preventive screenings are computed by comparing the date of the last clinical visit with the recommended screening interval specified for each test (Listing 13).

```python
from datetime import datetime

def get_due_screenings (patient_id ):
    today = datetime .today ().date ()
    screenings = db.query (Preventive Screening ). all()
    last_appt = get_last_appointment (patient_id )
    due = []

    for s in screenings:
        if not last_appt:
            months_since_last = 999
        else :
            months_since_last = (
                (today.year - last_appt. date .year) * 12 +
                (today .month - last_appt. date .month )
            )
        if months_since_last >= s.recommended_interval_months:
            due.append ({
                " name ": s. name ,
                "specialty ": s.specialty ,
                "months_since_last ": months_since_last
            })
    return due
```

Listing 13: Algorithm for detecting due preventive screenings.

Appointment Booking

Once the agent identifies available slots, it automatically books appointments using the following function (Listing 14)

```
def book_appointment(patient_id, doctor_id, date, time):
    new_appt = Appointment(
        patient_id=patient_id,

        doctor_id=doctor_id,
        date=date,
        time=time
    )
    db.add(new_appt)
    db.commit() return
    {
        "status":   "success","
        patient_id": patient_id,"
        doctor_id": doctor_id,"
        date": date,
        "time": time
    }
```

Listing 14: Automated Appointment Booking Function.

The database is updated in real time after successful booking, ensuring that the patient's appointment history is consistently maintained and supporting effective continuity-of-care management.

Lang Graph Integration

Lang Graph provides structured reasoning and execution capabilities for the proposed agentic framework through the following functionalities:

1) **Graph Construction:** Nodes are created to represent patients, appointments, preventive screenings, doctors, and available time slots.

2) **LLM Reasoning:** The large language model analyzes the graph context and suggests appropriate actions such as "Book ECG for patient ABHA123" or "Schedule Cholesterol Test".

3) **Action Execution:** Based on the selected action, the agent automatically triggers corresponding Fast API endpoints to complete appointment booking.

4) **Explainability:** The explicit graph structure enables visualization of the reasoning path, offering transparency and interpretability for clinicians.

Traversing this graph allows the agent to autonomously identify care gaps, select the most appropriate preventive screening, and execute the optimal appointment booking sequence, thereby ensuring continuity of care with minimal manual intervention.

Virtual Environment and Dependencies

The proposed system is deployed within a Python virtual environment to ensure dependency isolation and reproducibility. The environment is created and activated as follows:

Python -M Venv Venv

Source Venv/Bin/Activate # Linux/Macos

Venv\Scripts\activate   # Windows

Required dependencies are installed using the Python package manager:

pip install fastapi uvicorn sqlalchemy psycopg2-binary requests pip install openai langgraph

The core software components used in the system include:

a) **SQL Alchemy** for Object-Relational Mapping (ORM) and database interaction.

b) **Fast API** for developing RESTful APIs and agent-triggered endpoints.

c) **Psycopg2**-binary for PostgreSQL database connectivity.

d) **Requests** for secure communication with ABHA APIs.

e) **Lang Graph** for implementing agentic AI reasoning and graph-based decision work-flows.

Deployment and Testing

Local Deployment: The system is deployed locally using the Uvicorn ASGI server, as shown below:

uvicorn main: app -- reload

Once deployed, the system listens for patient queries, generates continuity-of-care reports, and autonomously books appointments when due preventive screenings are detected.

Testing Workflow: The following testing procedure is adopted to validate system functionality:

a) Creation of sample patient records with synthetic ABHA IDs in the PostgreSQL database.

b) Simulation of doctor availability and schedules using the ABHA sandbox API.

c) Verification that the agent correctly identifies due preventive screenings.

d) Validation of automatic appointment booking by the agent.

e) Confirmation of real-time database updates and action logs.

Advantages of the Proposed Implementation

The proposed Agentic AI-based continuity-of-care system offers several advantages:

a) Autonomous Decision-Making: The agent proactively detects gaps in care and executes actions without human intervention.

b) Explainable Reasoning: Lang Graph enables visualization of decision paths, ensuring transparency for clinicians.

c) Integration with National APIs: ABHA APIs provide real-time access to patient data and doctor schedules.

d) Proactive Preventive Care: Automated detection reduces missed screenings and supports long-term health outcomes.

e) Scalable Architecture: The combination of PostgreSQL and Fast API supports large-scale deployment across multiple hospitals.

Summary

This implementation demonstrates a fully functional Agentic AI system for continuity-of-care built on the ABHA digital health infrastructure. The key contributions of the system include:

a) Seamless integration with ABHA APIs for real-time patient and doctor information.

b) A PostgreSQL database for managing longitudinal patient care records.

c) RESTful Fast API endpoints for external system interaction.

d) A Lang Graph-based agent enabling autonomous reasoning and decision-making.

e) Preventive screening logic to ensure timely clinical interventions.

Overall, the proposed system provides a scalable, secure, and explainable solution for improving patient outcomes.

Experimental Evaluation

This section presents a comprehensive experimental evaluation of the proposed Agen-tic AI–based Continuity-of-Care and Automated Appointment Booking System. The evaluation focuses on assessing the system's effectiveness in improving preventive care adherence, optimizing appointment scheduling, and reducing administrative workload. The experiments are conducted using simulated yet realistic healthcare datasets aligned with the Indian healthcare ecosystem and ABHA digital health infrastructure.

Experimental Objectives

The experimental study is designed with the following objectives:

a) To evaluate the ability of the Agentic AI framework to identify overdue preventive screenings using longitudinal patient health records.

b) To measure the accuracy and timeliness of automated appointment bookings generated by the agent.

c) To analyze doctor slot utilization efficiency after agent-driven scheduling.

d) To assess the reduction in manual administrative effort compared to traditional appointment booking workflows.

e) To study the explainability and robustness of Lang Graph-based agent reasoning and decision paths.

Experimental Setup

Dataset Description: Due to strict privacy, consent, and regulatory constraints under the Ayushman Bharat Digital Mission (ABDM) framework, access to real patient clinical records is not available through ABHA production APIs or sandbox environments. Therefore, this study employs statistically realistic synthetic datasets modeled on real-world Indian healthcare scenarios to evaluate the proposed agentic framework. The synthetic data generation process adheres to ABDM guidelines and reflects realistic patient–doctor interaction patterns observed in Indian healthcare systems.

The dataset characteristics are summarized as follows:

a) Patients : 1,000 synthetic patient records

b) ABHA IDs: Unique 14-digit identifiers per patient

c) Doctors: 50 doctors across six specialties

1) General Medicine

2) Cardiology

3) Ophthalmology
4) Endocrinology
5) Neurology
6) Orthopedics

d) Appointments: 4,800 historical appointments spanning 24 months

e) Preventive Screenings: Eight screening types with guideline-based intervals (Table 5).

System Configuration: The system configuration used for experimental evaluation is summarized in (Table 6).

Table 5: Preventive Screening Intervals.

Screening Type	Interval (Months)	Specialty
Blood Pressure Check	12	General Medicine
Cholesterol Test	24	Cardiology
ECG	12	Cardiology
HbA1c Test	6	Endocrinology
Eye Examination	12	Ophthalmology
BMI Assessment	12	General Medicine
Neurological Checkup	24	Neurology
Bone Density Test	36	Orthopedics

Table 6: System Configuration.

Component	Configuration
Backend Framework	Fast API (Python 3.11)
Database	Postgre SQL 15
ORM	SQL Alchemy 2.0
Agent Framework	Lang Graph
LLM	GPT-based reasoning engine
Deployment	Local server (Uvicorn)
Hardware	Intel i7, 16 GB RAM

Each experiment simulates daily patient interactions over a duration equivalent to six months of continuous system usage. The agent processes patient histories, identifies overdue preventive screenings, retrieves doctor availability, and autonomously books appointments while logging all actions for evaluation.

Evaluation Metrics

To quantitatively assess the performance of the proposed Agentic AI framework, multiple evaluation metrics were defined. These metrics focus on preventive care adherence, scheduling efficiency, and operational effectiveness.

Preventive Care Adherence Rate (PCAR): The Preventive Care Adherence Rate (PCAR) measures the effectiveness of the agent in ensuring that patients complete recommended preventive screenings within the specified time intervals. It is defined as:

$$PCAR = \frac{Numberofcompletedpreventivescreening}{Totalnumberofrecommendedscreening} \quad (1)$$

A higher PCAR value indicates improved adherence to preventive care guidelines, reflecting the agent's capability to proactively detect due screenings and facilitate timely appointment scheduling.

Appointment Automation Accuracy (AAA): Appointment Automation Accuracy (AAA) evaluates the correctness of appointments booked autonomously by the agent. It is defined as:

$$AAA = \frac{Correctlybookedappointment}{Totalagent - initiatedappointme} \quad (2)$$

An appointment is considered correct if all of the following conditions are satisfied:

1) The medical specialty matches the preventive screening requirement
2) The selected doctor slot is available at the time of booking
3) No scheduling conflicts occur with existing patient appointments

Doctor Slot Utilization (DSU): Doctor Slot Utilization (DSU) measures how efficiently available clinical time slots are used after agent-driven scheduling. It is computed as:

$$DSU = \frac{Numberofbookedslots}{Totalavailableslots} \quad (3)$$

A higher DSU value indicates improved optimization of healthcare resources and reduced idle clinical capacity.

Administrative Effort Reduction (AER): Administrative Effort Reduction (AER) quantifies the decrease in manual workload achieved through automation. It is measured as the percentage reduction in the following activities:

1) Manual appointment booking calls
2) Manual follow-up reminders
3) Manual updating of patient records

This metric reflects the system's impact on reducing operational overhead for health-care staff.

Agent Decision Latency: Agent Decision Latency represents the average time taken by the agent to complete an end-to-end decision-making cycle. This includes:

1) Analyzing longitudinal patient history
2) Identifying overdue preventive screenings
3) Selecting an optimal doctor time slot
4) Triggering the appointment booking action

Lower decision latency indicates higher system responsiveness and suitability for real-time clinical workflows.

Baseline Comparison: The proposed Agentic AI–based system was compared against a traditional rule-based appointment scheduling system. The baseline system exhibited the following character-istics:

a. Manual follow-up reminders
b. Fixed rule-based booking logic
c. No longitudinal patient history reasoning
d. Absence of autonomous agent behavior

This comparison highlights the advantages of intelligent, proactive, and context-aware decision-making enabled by the agentic framework (Table 7).

Table 7: Preventive Care Adherence Comparison.

System Type	Adherence Rate (%)
Traditional System	58.3
Proposed Agentic AI System	84.7

Experimental Results

Preventive Care Adherence

Observation: The Agentic AI system significantly improved preventive care adherence by proactively identifying overdue screenings and autonomously initiating appointment bookings without waiting for patient intervention.

Appointment Automation Accuracy

(Table 8) Errors observed during experimentation were primarily caused by:

Table 8: Appointment Automation Accuracy.

Metric	Value
Total Agent-Initiated Bookings	2,340
Correct Bookings	2,287
Automation Accuracy (%)	97.7

a. Simulated ABHA API latency
b. Rapid slot exhaustion during peak consultation hours

Doctor Slot Utilization

(Table 9) **Insight:** The agent distributed appointments evenly across available time slots, reducing peak-hour congestion and improving overall utilization of clinical resources.

Administrative Effort Reduction: These results demonstrate substantial operational efficiency gains and reduced workload for healthcare administrative staff.

Table 9: Doctor Slot Utilization Comparison.

System Type	Slot Utilization (%)
Manual Scheduling	61.2
Agentic AI System	79.5

Agent Decision Latency: The sub-second decision latency confirms the system's feasibility for real-time deployment in clinical environments (Table 10&11).

Table 10: Administrative Effort Reduction.

Task	Reduction (%)
Manual Follow-Up Calls	72
Manual Appointment Booking	81
Manual Record Maintenance	65
Overall Reduction	73

Table 11: Agent Decision Latency Breakdown.

Task Stage	Average Time (ms)
Patient History Retrieval	120
Preventive Screening Analysis	180
Slot Selection	210
Booking Execution	140
Total Decision Time	650

Lang Graph Explainability Analysis

One of the core advantages of the proposed system is the explainable agent behavior enabled by LangGraph. Each agent decision is represented as a directed reasoning graph, allowing transparent inspection of the decision-making process.

Patient Node

↓

Last Appointment Node

↓

Preventive Screening Node

↓

Doctor Availability Node

↓

Booking Action Node

This structured reasoning flow enables clinicians and system auditors to:

1) Trace the rationale behind each preventive screening recommendation
2) Verify compliance with established clinical guidelines
3) Override or modify automated decisions when necessary

By explicitly exposing the reasoning path, the Lang Graph-based architecture addresses one of the primary barriers to AI adoption in healthcare systems, namely the lack of transparency and explainability in automated clinical decision-making.

Failure Case Analysis

Despite the strong overall performance of the proposed Agentic AI framework, several limitations were observed during experimental evaluation.

a. **API Dependency:** Delays in the simulated ABHA sandbox APIs occasionally resulted in booking retries, highlighting the system's dependency on external service availability.

b. **Edge Case Patients:** Patients with irregular or sparse appointment histories required additional reasoning cycles, increasing agent decision latency in certain cases.

c. **Specialty Overlap:** Certain preventive screenings, such as diabetes-related tests, could be associated with multiple medical specialties. This required the introduction of priority rules to ensure appropriate scheduling.

These failure cases provide valuable insights for future enhancements, including im-proved fallback strategies, adaptive reasoning heuristics, and specialty disambiguation mechanisms.

Statistical Significance

To validate the effectiveness of the proposed approach, a paired t-test was conducted comparing preventive care adherence rates between the traditional baseline system and the Agentic AI system.

The analysis yielded a p-value of less than 0.01, indicating that the observed improve-ment in preventive screening adherence is statistically significant. This confirms that the performance gains achieved by the proposed system are not due to random variation.

Discussion

The central research question addressed in this study is as follows:

Can a Model Context Protocol (MCP)–enabled Custom GPT effectively integrate het-erogeneous clinical data sources and machine learning models to deliver context-aware, explainable, and ethically compliant decision support for continuity of patient care? The experimental results demonstrate that the integration of Agentic AI with Lang-Graph reasoning successfully addresses this question. The system is capable of reasoning over longitudinal patient data, proactively managing continuity of care, reducing admin-istrative burden, and improving preventive healthcare outcomes. Unlike static rule-based systems, the proposed agent dynamically adapts to evolving patient histories, fluctuating doctor availability, and guideline-driven preventive intervals. This adaptability underscores the practical feasibility of deploying Agentic AI within large-scale national healthcare infrastructures such as the Ayushman Bharat Digital Mission.

Context-Aware Clinical Intelligence: The findings reveal that integrating MCP with a Custom GPT framework significantly enhances contextual awareness in clinical decision support systems. Rather than operating on isolated datasets, the system aggregates patient context from electronic health records, laboratory systems, wearable devices, and external medical tools. This unified representation enables longitudinal and situational reasoning, addressing a key limitation of conventional predictive healthcare models.

Improved Predictive Reliability Through Multi-Model Reasoning: By orchestrating multiple machine learning models, including KNN, Random Forest, and XG Boost, under the MCP layer, the framework achieves greater robustness and consistency in predictive outcomes. Ensemble-based contextual reasoning reduces variance and mitigates model bias, aligning with real-world clinical workflows where corroborative evidence is preferred over single-model predictions.

Continuity of Care and Longitudinal Decision Support: A major contribution of this work lies in operationalizing continuity of care. The pro-posed framework maintains persistent patient context across appointments and clinical events, enabling follow-up recommendations, appointment scheduling, and trend-based risk assessment. This represents a significant advancement over traditional systems that treat clinical prediction as a one-time task.

Explainability and Trust in Clinical AI: Explainability is achieved through Lang Graph-based reasoning and LLM-driven natural language explanations. Clinicians can inspect the decision pathway, understand screening recommendations, and verify model behavior. This transparency is essential for clinical trust and aligns with emerging regulatory requirements for explainable AI in healthcare.

Ethical Compliance and Consent Awareness: The MCP-enabled workflow incorporates ethical safeguards such as consent validation and controlled data access as integral components of the system architecture. By embed-ding governance and compliance mechanisms directly into agent behavior, the framework ensures that automated decisions are not only clinically valid but also ethically and legally sound.

Implications for Agentic AI in Healthcare: Overall, the study confirms that MCP-enabled Custom GPT systems can function as intelligent clinical agents rather than passive prediction tools. Through context persistence, tool orchestration, multi-model inference, and ethical safeguards, the proposed system demonstrates strong potential for real-world deployment. These findings establish MCP as a foundational enabler for context-aware, agentic healthcare AI.

Summary of Findings

The experimental evaluation demonstrates the effectiveness of the proposed Agentic AI–based continuity-of-care framework. The key findings are summarized as follows:

a. 26.4% improvement in preventive care adherence compared to traditional scheduling systems.

b. 97.7% appointment booking accuracy, ensuring correct specialty matching and conflict-free scheduling.

c. 73% reduction in overall administrative effort, including follow-up calls, manual bookings, and record maintenance.

d. Sub-second agent decision latency, enabling real-time continuity-of-care man-agement.

e. Transparent and explainable decision-making through LangGraph-based rea-soning paths accessible to clinicians.

These findings strongly validate the proposed framework's ability to deliver scalable, autonomous, and explainable continuity-of-care solutions within national digital health-care ecosystems such as the Ayushman Bharat Digital Mission.

Limitations and Future Research Directions

While the proposed Agentic AI–based Continuity-of-Care and Automated Appointment Booking System demonstrates strong potential and promising experimental results, several limitations remain. This section critically analyzes the current constraints and out-lines future research directions to enhance scalability, intelligence, and real-world applicability.

System Limitations

Dependency on ABHA API Availability: The proposed frame-work relies on ABHA sandbox APIs for patient identity and data ac-cess. Although ABDM provides a robust national digital infrastruc-ture, the system performance depends on:

1) API uptime

2) Network latency

3) Token-based authentication stability

In real-world deployments, intermittent service disruptions or delayed responses from national health APIs may affect real-time appointment booking. While the system currently employs retry and fallback mechanisms, resilience against prolonged outages remains a challenge.

Use of Synthetic and Simulated Data: Due to privacy and regulatory constraints, experimental validation was conducted using synthetic datasets modeled after real healthcare scenarios. While this ensures ethical compliance, it limits:

a. Exposure to rare clinical edge cases

b. Behavioral variability of real patients

c. Complex hospital scheduling constraints

As a result, real-world performance may vary when deployed at scale.

Limited Clinical Scope: The system is intentionally restricted to:

a. Preventive care reminders

b. Appointment scheduling

c. Administrative continuity-of-care functions

It does not perform disease diagnosis, treatment recommendation, or alteration of clinical decisions. While this design choice enhances safety and regulatory alignment, it limits the depth of clinical reasoning.

Rule-Based Preventive Guidelines: Preventive screening intervals are currently derived from static, guideline-driven rules. This approach does not fully account for:

a. Individual risk profiles

b. Genetic predispositions

c. Lifestyle factors

d. Co-morbidities

Consequently, recommendations may lack personalized risk stratification.

Single-Agent Architecture: The current implementation adopts a centralized, single-agent design per patient. Al-though effective, this limits:

a. Parallel reasoning across specialties

b. Distributed decision-making

c. Multi-hospital coordination

Scalability across large healthcare networks may require advanced agent orchestration.

Technical Limitations of Agentic AI

Reasoning Consistency: LLM-based reasoning introduces flexibility but may exhibit:

a. Minor variability in explanation generation

b. Sensitivity to prompt formulation

c. Occasional ambiguity in edge cases

Although LangGraph constrains agent behavior, full determinism remains an open research challenge.

Computational Overhead: Agentic reasoning introduces additional computational cost compared to rule-based systems. While sub-second response times were observed experimentally, national-scale deployment performance requires further evaluation.

Explainability Boundaries: Despite improvements through LangGraph, explanations may remain:

a. Abstract for non-technical users

b. Dependent on visualization quality

Simplifying explanations for patients with low digital literacy remains an open challenge.

Future Research Directions

Real-World Pilot Deployment: Future work includes pilot deployment in:

a. Government hospitals

b. Primary Health Centers (PHCs)

c. Private hospital networks

Such pilots would enable live ABHA integration, longitudinal outcome tracking, and real-user feedback.

Personalized Preventive Care Using Risk Models: Future systems can integrate:

a. Machine learning risk prediction models

b. Lifestyle and wearable data

c. Family history and co-morbidity information

This would enable dynamic screening intervals and improved preventive outcomes.

Multi-Agent Collaboration: Extending to a multi-agent architecture can improve scalability:

a. Patient Agent for longitudinal context

b. Scheduling Agent for hospital-wide optimization

c. Policy Agent for ethical and regulatory enforcement

d. Analytics Agent for performance monitoring LangGraph naturally supports such orchestration.

Integration with Wearables and IoT: Future research can incorporate continuous health monitoring through wearables, enabling event-driven interventions such as automated specialist appointments triggered by abnormal vitals.

Reinforcement Learning for Scheduling Optimization: Reinforcement learning can be explored to:

a. Minimize patient wait times

b. Maximize doctor utilization

c. Adapt to seasonal demand patterns

This hybrid approach can combine learning-based optimization with agentic reasoning.

Cross-Hospital Interoperability: Future extensions may enable cross-hospital scheduling, referral coordination, and seamless data exchange, aligning with ABDM's long-term interoperability vision.

Advanced Ethical AI Controls: Further research is required in:

a. Dynamic consent management

b. Automated bias detection

c. Ethical rule enforcement agents

These capabilities would enhance trust and regulatory compliance.

Research Implications

The proposed system opens new research avenues in agentic AI for national healthcare systems, explainable autonomous decision-making, and policy-aware clinical AI deployment.

Summary

This section identified key limitations related to data realism, scalability, agent architecture, and personalization, while outlining future research directions to enhance robustness and societal impact. Addressing these challenges is essential for transitioning agentic AI systems from controlled environments to real-world healthcare deployments under national digital health missions such as ABDM.

Conclusion

This paper presented a novel Agentic AI–driven framework for automated appointment booking and continuity-of-care management, designed specifically for the Indian health-care ecosystem under the Ayushman Bharat Digital Mission (ABDM). By leveraging ABHA health identifiers, LangGraph-based agentic reasoning, and secure API-driven system integration, the proposed approach addresses long-standing challenges in preventive care adherence, fragmented follow-ups, and administrative inefficiencies. Unlike traditional rule-based or reactive healthcare systems, the proposed framework introduces proactive, goal-oriented autonomy through agentic AI. The system continuously reasons over longitudinal patient histories, identifies gaps in preventive care, and autonomously initiates appointment scheduling actions while maintaining strict ethical boundaries and human oversight. The use of LangGraph enables structured and explainable decision-making, transforming opaque AI reasoning into transparent, auditable workflows that clinicians and patients can trust. Experimental evaluation using realistic synthetic datasets demonstrated substantial improvements across key healthcare metrics, including preventive screening adherence, doctor slot utilization, and administrative workload reduction. The results validate that agentic AI systems, when carefully constrained and aligned with national digital health standards, can operate effectively in real-time healthcare workflows without compromising safety or privacy. Equally important, this work emphasized ethical, legal, and regulatory compliance by integrating consent-driven data access, privacy-preserving storage, and accountability mechanisms consistent with ABDM and India's Digital Personal Data Protection Act. The framework demonstrates that autonomy in healthcare AI does not necessitate loss of control; rather, autonomy can coexist with transparency, consent, and human-in-the-loop governance.

From a research perspective, this study contributes to the emerging field of agentic AI in healthcare operations by moving beyond decision support toward autonomous care coordination. It highlights how graph-based agent frameworks can bridge the gap between large language model reasoning and real-world healthcare actions. The proposed architecture provides a reusable blueprint for future healthcare agents, including those focused on chronic disease management, telemedicine coordination, and population-scale preventive health. In conclusion, this work demonstrates that Agentic AI, when grounded in policy-compliant digital health infrastructure and explainable system design, can play a trans-formative role in strengthening continuity of care. As India continues its journey toward a unified digital health ecosystem, such intelligent and autonomous systems have the potential to significantly enhance healthcare accessibility, operational efficiency, and long-term patient outcomes.

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.

Conflict of Interest

None.

References

1. Al Jadaan, OS Babu, SK Gupta (2025) Balancing Consent and Privacy in AI-Driven Health Sciences Education in the GCC: A Comparative Policy and Practice Analysis. SGS-Engineering & Sciences 1.4.
2. Borkowski AA, A Ben Ari (2025) Multiagent AI Systems in Health Care: En-visioning Next-Generation Intelligence. In: Federal Practitioner 42(5): 188-194.
3. Fuentes VGH, HA Karim, MJT Tan, Nouar AlDahoul (2025) AI with Agency: A Vision for Adaptive, Efficient, and Ethical Healthcare. Frontiers Digital Health 7(7):1600216.
4. Jia Y, H Evans, Z Porter (2025) The Case for Delegated AI Autonomy for Human-AI Teaming in Healthcare. arXiv preprint arXiv 2503: 18778.
5. Joshi S (2025a) Framework for Government Policy on Agentic and Generative AI in Healthcare. Preprints.org. (2025b) GenAI Agents for Early Disease Diagnosis: A Review of Architectures, Applications, and Policy Directions Preprints.org.
6. Katonai G, N Arvai, B Mesko (2025) AI and Primary Care: Scoping Review. Journal of Medical Internet Research 27: e65950.
7. Nag PK, A Bhagat, RV Priya (2025) Expanding AI Role in Healthcare Applications: A Systematic Review of Emotional and Cognitive Analysis Techniques. IEEE Access.
8. Shehab MA (2025) Agentic-AI Healthcare: Multilingual, Privacy-First Framework with MCP Agents. In: arXiv preprint. arXiv: 2510.02325.
9. Srinivasu, PN, Gorli L Aruna Kumari, Shakeel Ahmed, Abdulaziz Alhumam (2025) Exploring Agentic AI in Healthcare: A Study on Its Working Mechanism. Frontiers in Medicine 12: 1753443.
10. Taimoor N, S Rehman (2021) Reliable and Resilient AI and IoT-Based Personalised Healthcare Services: A Survey. IEEE Access 3137364.
11. Yu CY (2025) From Passive to Proactive: A Multi-Agent System with Dynamic Task Orchestration for Intelligent Medical Pre-Consultation. arXiv preprint.